



**AN AUTOML TOOL FOR SOFTWARE VULNERABILITY PREDICTION**

**TARIK GÜRCAN**

**AUGUST 2023**

**ÇANKAYA UNIVERSITY**

**GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**

**DEPARTMENT OF COMPUTER ENGINEERING**

**M.Sc. Thesis in**

**COMPUTER ENGINEERING**

**AN AUTOML TOOL FOR SOFTWARE VULNERABILITY PREDICTION**

**TARIK GÜRCAN**

**AUGUST 2023**

## **ABSTRACT**

### **AN AUTOML TOOL FOR SOFTWARE VULNERABILITY PREDICTION**

**GÜRCAN, TARIK**

**M.Sc. in Computer Engineering**

Supervisor: Assist. Prof. Dr. Gül TOKDEMİR

August 2023, 209 pages

The popularization of machine learning has been inevitable, especially in recent years, as large amounts of data have become more accessible and provide significant benefits for industries such as finance, healthcare, logistics, and technology. The need to implement machine learning applications, which have started to be used in different sectors, has increased rapidly and efficiently. However, the deep technical knowledge and coding skills required to build a machine-learning model using traditional methods may not be accessible to everyone. Therefore, no-code or low-code platforms have started to attract more attention. These platforms allow users to load datasets, define features for building the model, and even train and evaluate the model with non-technical interfaces. In this way, it eliminates the need to write complex code, making it possible for a wider audience to be involved in machine learning projects. In this thesis, a platform that enables performing no-code machine learning prediction models for software vulnerability analysis is developed, which offers important features such as run-time limiting and model degradation analysis in addition to other no-code platforms. This platform is named PPDaaS (Prediction Platform as a Service) and is used to train prediction models for “Software Vulnerability” and “OpenML” benchmarking datasets. Initially, the platform is tested with benchmarking data that is used for AutoML platforms. The performance of the classification algorithms is evaluated by placing a time limit on the "OpenML adult" dataset, to make a fair comparison between the machine learning algorithms. Once the

platform is tested for its intended AutoML functionalities, it is used to find model degradation on Software Vulnerability data. Finally, the feedback about the developed PPDaaS platform is gathered from end users as well. The results indicate that the developed PPDaaS platform can find a place for itself in the software industry.

**Keywords:** Machine Learning, Runtime Limit in Machine Learning, Model Degradation, No Code/Low Code Machine Learning, Software Vulnerability, Data Analysis



## ÖZET

### YAZILIM GÜVENLİK AÇIĞI TAHMİNİ İÇİN AUTOML ARACI

**GÜRCAN, TARIK**

**Bilgisayar Mühendisliği Yüksek Lisans**

Danışman: Dr. Öğr. Üyesi Gül TOKDEMİR

Ağustos 2023, 209 sayfa

Özellikle son yıllarda büyük miktarda verinin daha erişilebilir hale gelmesi ve finans, sağlık, lojistik ve teknoloji gibi sektörlerde önemli faydalar sağlanması nedeniyle makine öğreniminin yaygınlaşması kaçınılmaz olmuştur. Farklı sektörlerde kullanılmaya başlayan makine öğrenmesi uygulamalarının hayata geçirilmesi ihtiyacı hızlı ve verimli bir şekilde artmaktadır. Ancak geleneksel yöntemleri kullanarak bir makine öğrenimi modeli oluşturmak için gereken derin teknik bilgi ve kodlama becerilerine herkes yetkin olmayabilir. Bu nedenden dolayı kodsuz veya az kodlu platformlar daha fazla ilgi görmeye başlamıştır. Bu platformlar, kullanıcıların veri kümelerini yüklemesine, modeli oluşturmaya yönelik özellikleri tanımlamasına ve hatta teknik olmayan arayüzlerle modeli eğitmesine ve değerlendirmesine olanak tanır. Bu sayede karmaşık kod yazma ihtiyacını ortadan kaldırarak daha geniş bir kitlenin makine öğrenimi projelerine dahil olmasını mümkün kılar. Bu tezde, diğer kodsuz platformlara ek olarak çalışma zamanı sınırlama ve model bozulma analizi gibi temel özellikleri sunan, yazılım zafiyet analizi için kodsuz makine öğrenimi tahmin modellerinin gerçekleştirilmesine olanak sağlayan bir platform geliştirilmiştir. Bu platform, PPDaaS (Prediction Platform as a Service) olarak adlandırılmıştır ve "Yazılım Güvenlik Açığı" ve "OpenML" veri kümeleriyle tahmin modellerini eğitmek için kullanılmıştır. Başlangıçta platform, AutoML platformları için kullanılan kıyaslama verileriyle test edildi. Sınıflandırma algoritmalarının performansı, makine öğrenimi algoritmaları arasında adil bir karşılaştırma yapmak için "OpenML yetişkin"

veri kümesine bir zaman sınırı konularak değerlendirildi. Platform, amaçlanan AutoML işlevleri açısından test edildikten sonra Yazılım Güvenlik Açığı verilerindeki model bozulmasını bulmak için kullanıldı. Son olarak geliştirilen PPDaaS platformuna ilişkin geri bildirimler son kullanıcılardan da toplandı. Sonuçlar geliştirilen PPDaaS platformunun yazılım sektöründe kendine yer bulabileceğini göstermektedir.

**Anahtar Kelimeler:** Otomatik Makine Öğrenimi, Makine Öğreniminde Zaman Sınırı, Model Bozunumu, Kodsuz/Düşük Kodlu Makine Öğrenimi, Yazılım Açığı, Veri Analizi



## ACKNOWLEDGEMENT

First and foremost, I would like to extend my heartfelt gratitude to my advisor, Assist. Prof. Dr. Gül TOKDEMİR, for her exceptional guidance, ideas, and patience. Your guidance played a significant role in enhancing the quality of this thesis. Similarly, I would like to thank my family and girlfriend, Ms. Pelin Kùlah, who has been by my side throughout this process. Your patience, understanding, and unwavering support made me never feel alone during this challenging journey. Knowing that you were always there for me has been my most significant source of strength.

## TABLE OF CONTENTS

<b>STATEMENT OF NONPLAGIARISM .....</b>	<b>III</b>
<b>ABSTRACT .....</b>	<b>IV</b>
<b>ÖZET .....</b>	<b>VI</b>
<b>ACKNOWLEDGEMENT .....</b>	<b>VIII</b>
<b>LIST OF TABLES .....</b>	<b>XII</b>
<b>LIST OF FIGURES .....</b>	<b>XIV</b>
<b>LIST OF SYMBOLS AND ABBREVIATIONS .....</b>	<b>XV</b>
<b>CHAPTER I .....</b>	<b>1</b>
<b>INTRODUCTION.....</b>	<b>1</b>
1.1 THESIS STRUCTURE.....	3
1.2 THESIS OBJECTIVE.....	3
<b>CHAPTER II.....</b>	<b>4</b>
<b>BACKGROUND .....</b>	<b>4</b>
2.1 EXPLORING THE SIGNIFICANCE OF MACHINE LEARNING .....	4
2.2 MACHINE LEARNING INTEGRATION IN NO-CODE AND LOW- CODE PLATFORMS.....	5
2.3 SOFTWARE VULNERABILITY DETECTION.....	6
<b>CHAPTER III .....</b>	<b>12</b>
<b>RESEARCH METHODOLOGY .....</b>	<b>12</b>
3.1 PLATFORM PREDICTION AS A SERVICE PLATFORM .....	12
3.2 RUNTIME LIMIT FEATURE .....	17
3.3 MODEL DEGRADATION FEATURE .....	18
3.4 INTERFACES OF THE PPDAS .....	19
3.4.1 Import Data.....	19
3.4.2 Specifying the Main Options.....	20
3.4.3 Target Feature Selection.....	21
3.4.4 Drop Features .....	22
3.4.5 Feature Selection Algorithms .....	23



3.4.6	Label Encoding.....	23
3.4.7	Building the Machine Learning Process .....	24
3.4.8	Result.....	24
3.4.9	Download Output .....	24
3.4.10	Model Degradation Analysis.....	25
3.4.11	Model Performance Metrics Provided by PPDaaS Platform .....	26
3.4.11.1	Accuracy .....	26
3.4.11.2	Mean Squared Error (MSE) .....	26
3.4.11.3	Precision.....	26
3.4.11.4	Root Mean Squared Error (RMSE).....	26
3.4.11.5	Recall.....	26
3.4.11.6	Mean Absolute Error (MAE) .....	27
3.4.11.7	F1 Score .....	27
3.4.11.8	R-Squared( $R^2$ ) .....	27
3.4.12	Survey About the Usability of PPDaaS.....	27
<b>CHAPTER IV</b>	<b>.....</b>	<b>28</b>
<b>EXPERIMENTS AND RESULTS</b>	<b>.....</b>	<b>28</b>
4.1	DATA OVERVIEW .....	28
4.1.1	Benchmarking AutoML Data.....	28
4.1.2	Vulnerability Experiment Data .....	29
4.2	VULNERABILITY DATASET EXPERIMENT .....	31
4.2.1	Functions Code Level.....	31
4.2.2	Class Code Level and File Code Level .....	32
4.3	BECHMARKING AUTOML EXPERIMENT.....	33
4.4	VULNERABILITY EXPERIMENT RESULTS.....	35
4.4.1	Function Code Level .....	35
4.4.1.1	Results Between the Years 2002 to 2014 .....	35
4.4.1.2	Results Between the Years 2002 to 2016 .....	37
4.4.1.3	Results Between the Years 2002 to 2019 .....	39
4.4.2	Results of the Class Level .....	41
4.4.3	File Code Level Results .....	42
4.5	BENCHMARKING AUTOML EXPERIMENT RESULTS.....	43
4.6	SURVEY RESULTS OF THE USABILITY OF PPDAAAS.....	45
<b>CHAPTER V</b>	<b>.....</b>	<b>47</b>

<b>DISCUSSION .....</b>	<b>47</b>
5.1 DISCUSSION ON OPENML ADULT DATA EXPERIMENT.....	47
5.2 SOFTWARE VULNERABILITY EXPERIMENT.....	48
5.2.1 Function Code Level .....	49
5.2.2 Class Code Level.....	53
5.2.3 File Code Level .....	54
5.2.4 Usability Analysis .....	55
<b>CHAPTER VI.....</b>	<b>57</b>
<b>CONCLUSION .....</b>	<b>57</b>
6.1 FUTURE WORK.....	58
<b>REFERENCES.....</b>	<b>59</b>
<b>APPENDICES .....</b>	<b>66</b>

## LIST OF TABLES

<b>Table 1:</b> Comparison Table of the AutoML Tools.....	6
<b>Table 2:</b> Classification Algorithms Provided by the PPDaaS Platform.....	14
<b>Table 3:</b> Regression Algorithms Provided by the PPDaaS Platform .....	15
<b>Table 4:</b> All Feature Selection Methods Provided by The Platform.....	16
<b>Table 5:</b> Vulnerability Count in Each Period.....	32
<b>Table 6:</b> Results of the 2002-2014 Period Sorted by Accuracy .....	35
<b>Table 7:</b> Results of the 2002-2014 Period Sorted by Precision .....	35
<b>Table 8:</b> Results of the 2002-2014 Period Sorted by Recall .....	36
<b>Table 9:</b> Results of the 2002-2014 Period Sorted by F1 Score.....	36
<b>Table 10:</b> Results of the 2002-2016 Period Sorted by Accuracy .....	37
<b>Table 11:</b> Results of the 2002-2016 Period Sorted by Precision .....	37
<b>Table 12:</b> Results of the 2002-2016 Period Sorted by Recall .....	38
<b>Table 13:</b> Results of the 2002-2016 Period Sorted by F1 Score.....	38
<b>Table 14:</b> Results of the 2002-2019 Period Sorted by Accuracy .....	39
<b>Table 15:</b> Results of the 2002-2019 Period Sorted by Precision .....	39
<b>Table 16:</b> Results of the 2002-2019 Period Sorted by Recall .....	40
<b>Table 17:</b> Results of the 2002-2019 Period Sorted by F1 Score.....	40
<b>Table 18:</b> Results Sorted by Accuracy .....	41
<b>Table 19:</b> Results Sorted by Precision .....	41
<b>Table 20:</b> Results Sorted by Recall .....	41
<b>Table 21:</b> Results Sorted by F1 Score .....	42
<b>Table 22:</b> File Code Level Results Sorted by Accuracy .....	42
<b>Table 23:</b> File Code Level Results Sorted by Precision.....	42
<b>Table 24:</b> File Code Level Results Sorted by Recall .....	43
<b>Table 25:</b> File Code Level Results Sorted by F1 Score .....	43
<b>Table 26:</b> Results of the OpenML Data Experiment Sorted by ROC AUC .....	43
<b>Table 27:</b> Results of the OpenML Data Experiment Sorted by Precision .....	44
<b>Table 28:</b> Results of the OpenML Data Experiment Sorted by Recall.....	44

<b>Table 29:</b> Results of the OpenML Data Experiment Sorted by F1 Score.....	44
<b>Table 30:</b> Results of the Evaluation Survey.....	46
<b>Table 31:</b> Highest Scores Concerning Score Parameters.....	47
<b>Table 32:</b> Comparison Table with AutoML vs. PPDaaS .....	48
<b>Table 33:</b> Comparison between our results versus Pereira and colleague's results ..	50
<b>Table 34:</b> 2002-2014 to 2002-2016 Model Degradation Top 4 Model Improvement Results.....	51
<b>Table 35:</b> 2002-2014 to 2002-2016 Model Degradation Top 4 Model Deterioration Results.....	51
<b>Table 36:</b> 2002-2016 to 2002-2019 Model Degradation Top 4 Model Improvement Results.....	52
<b>Table 37:</b> 2002-2016 to 2002-2019 Model Degradation Top 4 Model Deterioration Results.....	53
<b>Table 38:</b> Best ML Algorithm and Feature Selection according to Metrics in all Granularity .....	55
<b>Table 39:</b> Survey Results with Average Score.....	56

## LIST OF FIGURES

<b>Figure 1:</b> Runtime Limit Cycle .....	17
<b>Figure 2:</b> Model Degradation Cycle.....	18
<b>Figure 3:</b> Main Page of The Platform .....	19
<b>Figure 4:</b> Options Section of The Platform.....	20
<b>Figure 5:</b> Target Feature Selection Section .....	21
<b>Figure 6:</b> Drop Feature Section.....	22
<b>Figure 7:</b> Feature Selection Section .....	23
<b>Figure 8:</b> Model Performance Modal of the Platform.....	24
<b>Figure 9:</b> Model Degradation Tab.....	25
<b>Figure 10:</b> OpenML Data Overview .....	29
<b>Figure 11:</b> Reported Vulnerabilities Concerning Year .....	31
<b>Figure 12:</b> Class Code Level Settings Display in Platform .....	32
<b>Figure 13:</b> Optimized Values Display in Platform.....	33
<b>Figure 14:</b> Overview of The OpenML Adult Data .....	34
<b>Figure 15:</b> Model Degradation Tool Output .....	50

## LIST OF SYMBOLS AND ABBREVIATIONS

### SYMBOLS

$\Sigma$  : Uppercase Sigma (Summation Notation)

### ABBREVIATIONS

PPDaaS : Prediction Platform as a Service

Gaussian NB : Gaussian Naïve Bayes

MLP : Multi-Layer Perceptron

Bernoulli NB : Bernoulli Naive Bayes

SVC : Support Vector Classifier

SGD : Stochastic Gradient Descent

XGB : XG Boost

LGBM : Light GBM

Ridge CV : Ridge Cross-Validated

SVR : Support Vector Regressor

LGBM : Light GBM

XGB : XG Boost

Nu SVR : Nu Support Vector Regressor

RBF : Radial Basis Function

PCA : Principal Component Analysis

SVD : Singular Value Decomposition

RFE : Recursive Feature Elimination

RFECV : Recursive Feature Elimination with Cross-Validation

ICA : Independent Component Analysis

CSV : Comma-Separated Values

## **CHAPTER I**

### **INTRODUCTION**

The ability of computers to learn from the data and to improve their performance without any programming has enhanced machine learning and artificial intelligence which has a significant effect on our lives [1]. In today's world, artificial intelligence, and machine learning change [2] many domains ranging from medical [3][4][5], finance [6][7][8], business [9], empirical sciences [1], technology [10], consumer services [1], identification of bugs [1] and logistics [1]. Therefore, applying machine learning (ML) techniques and extracting valuable insights from the data has become more important. It led to no and low code machine learning platforms [11][12][13][14] to emerge in the response to growing diversity of machine learning applications and cause the demand for machine learning solutions. This resulted in challenges associated with traditional manual model development. These platforms (e.g., Weka[15], H2O.ai[16] ) enabled users to apply any model to his/her data without writing any code which increased the population of the data-science workers. Thus, automated machine learning tools (AutoML) have increasingly popular in today's world. However, there are very few AutoML platforms that provide runtime limits for experimentation with fair comparisons of different models. These tools work as applied machine learning systems that run ML algorithms for as much time as needed and evaluate the results. The absence of a runtime constraint in automatic machine learning tools gives rise to apprehensions regarding fairness and the feasibility of conducting unbiased comparisons between various ML algorithms. In today's world, the popularity, and the diversity in domains of machine learning applications have accelerated data accumulation which might affect the performance of the algorithms. As data is generated from the systems continuously, the previously created prediction models might not work properly on the newly generated datasets. Hence, the performance of the ML prediction models might degrade as new data arrives from resources. This might have a significant impact on the performance and reliability of the prediction models. Thus, the recognition of the potential for data degradation is

important for data scientists and software practitioners who heavily depend on predictive models to make well-informed decisions. Accordingly, in the software world, various machine learning models are utilized to enhance the quality of systems by detecting errors[17], defects[18], or vulnerabilities[19][21]. The software vulnerabilities are critical weaknesses that can expose systems to threats [22] and risks [23]. Particularly, detecting software vulnerability has become vital [24][25] when developing software systems [26] as it may cause attackers might use it to their advantage, which could compromise the system[27]. Accordingly, several software metrics were collected for the prediction of the vulnerabilities in C/C++ and Java-based systems[28]. Although metrics are utilized to identify vulnerabilities[20] through machine learning, prediction models built using data at different granularity levels (class and method level) have not been studied before [29] for the recently generated Software Vulnerability Datasets data.

The absence of the runtime limit option and model degradation analysis in AutoML systems shows that there is a need for an innovative and comprehensive tool including these features as well. Hence, the primary objective of the thesis is to develop a no-code platform named Prediction Platform as a Service (PPDaaS) that enables the user to apply machine learning models automatically with or without a given run time limit for fair comparison; apply feature selection models and observe model degradation over a period on software vulnerability data. Therefore, after the development of the platform, the software vulnerability data provided by Pereira and colleagues [28] in 2022 with all granularity levels namely, function, classes, and files are utilized for building prediction models. Studies of software vulnerability prediction are simulated by separating this dataset at the function granularity level and conducting experiments on it. In addition to that, a benchmark dataset from the openML platform is used in validating the AutoML platform. OpenML [30] is an open-source platform that provides machine learning datasets, algorithms, and experiments that allow researchers and data scientists to compare their works. Finally, the developed AutoML platform is tested by software professionals and their feedback is gathered through a survey instrument.



## **1.1 THESIS STRUCTURE**

The rest of this thesis is organized as follows: Chapter 2 will include an in-depth literature review on the topics of machine learning and its diversity, software vulnerability, and automated machine learning tools. Chapter 3 will present the design of the architecture of the Platform named Prediction Platform as a Service (PPDaaS) and the features of this platform that show how a user can utilize and interact with it. Chapter 4 will explain the experimental setup and results of the experiments performed on the PPDaaS platform using the Software Vulnerability dataset and OpenML adult dataset. Chapter 5 will discuss the results of the experiments and draw an insight into the tool's strengths and limitations. Finally, Chapter 6 will conclude the thesis, and summarize the contributions and recommendations for future study to improve the effectiveness and usefulness of PPDaaS in practical settings.

## **1.2 THESIS OBJECTIVE**

This thesis aims to bridge the gap in the field of automated machine learning tools by introducing the PPDaaS. By developing this tool, we strive to enable the researchers to apply a large set of machine learning models without writing any code use a large set of feature selection techniques with or without running specific time limits, and easily analyze model degradation over time on software vulnerability data. Accordingly, this thesis addresses four main research questions:

1. What are the missing features in AutoML tools?
2. How effective PPDaaS (Platform for Prediction as a Service) is?
3. How does software vulnerability prediction model performance change at different granularity levels?
4. How does prediction model performance change as new data arrive?

## **CHAPTER II**

### **BACKGROUND**

In this chapter, we will explore the importance of machine learning, its diverse applications, its integration into no-code/low-code platforms, and its role in addressing software vulnerabilities, thereby aiming to fill the existing gap in the literature.

#### **2.1 EXPLORING THE SIGNIFICANCE OF MACHINE LEARNING**

In the field of machine learning, many studies highlight the importance of machine learning. For example, Anute and colleagues [9] investigated the effects of machine learning on business organizations and operations. Furthermore, Sarker [10] provided a thorough overview of the machine-learning techniques that may be used to improve the functionality and intelligence of an application.

Jordan and colleagues [1], discuss the dramatic progression of machine learning applications over the past decades. It has become the approach of choice in AI for creating useful software for robot control, computer vision, and speech recognition which has a significant impact on computer science, and also on sectors that deal with data such as consumer services, identification of bugs, medicine, complex systems, and logistics. In addition to these, Jordan and colleagues [1] state that it has a big impact on empirical sciences including biology cosmology, and social science. Accordingly,

Gong and colleagues [2] state that the methods of machine learning have reached high performance and have been widely used in a variety of applications that are used in the real world. For example, Lee and colleagues [3] compare Gradient boosting, Support vector machine, Random Forest, Cat Boost, multilayered perceptron, and AdaBoost machine learning models to predict malaria by using patient data. They state that machine learning can have promising results. Gogas and Papadimitriou evaluate the machine learning effects on finance and economics[8]. The authors of [4] have reviewed current cardiology literature and reported on how doctors in the area are utilizing artificial intelligence tools, notably machine learning

approaches. The study of [5] has aimed to describe the usage of machine learning techniques on biological data to biologists. According to Goodell and colleagues [6] the rise of machine learning and artificial intelligence significantly affected the financial industry. Similarly, the study of [7] focuses on the effects of machine learning techniques on the financial sector and they evaluate techniques with traditional methods.

## **2.2 MACHINE LEARNING INTEGRATION IN NO-CODE AND LOW-CODE PLATFORMS**

A study by Lebens and colleagues [11] explores the utilization of low-code and no-code development platforms to enhance an organization's development capabilities. The study revealed that companies of various sizes were utilizing these platforms, frequently in combination with workflow automation.

AutoML is an acronym for Automated Machine Learning that refers to the process of automating the development of machine learning algorithms with various features such as feature selection, data preprocessing, and model training. The study conducted by Hanussek M. and colleagues [12] discusses that automated machine learning can surpass humans. In addition, they study a comparative analysis of four AutoML frameworks using a selection of 12 widely recognized datasets sourced from OpenML. The datasets encompass a balanced distribution of both supervised classification and supervised regression tasks, with six datasets assigned to each category.

Musigmann M and colleagues [13] examine the viability, as well as the merits and drawbacks, of constructing AutoML models in comparison to constructing traditional machine learning models. They use a total of 20 distinct machine-learning algorithms that were concurrently trained and evaluated. To facilitate comparison, both a neural network and various conventional machine learning algorithms were trained and subsequently tested. The findings presented in the article demonstrate the promising potential of AutoML.

Eggenberger K et al. [14] provide a new solution for automated machine learning tools (AutoML). They provide Auto-Sklearn 2.0 as an AutoML system to run under the time limits named budget allocation. They evaluate their systems by datasets located on the openML website.

In Table 1, we observe key differences between various AutoML platforms and our platform PPDaaS. Generally, all platforms provide preprocessing, automatic execution of machine learning algorithms, and feature selection methods. Specifically, Auto-Sklearn 1.0 and Auto-Sklearn 2.0 provide runtime limits but are different than ours. Their approach differs from ours, we simulate all algorithms at the same time, and they eliminate algorithms that exceed a certain runtime. However, none of them provides a model degradation analysis. Therefore, our provided PPDaaS platform aims to fill these gaps, offering an enhanced runtime limit and model degradation feature.

**Table 1:** Comparison Table of the AutoML Tools

Platform	Preprocessing Capabilities	Machine Learning Algorithms	Feature Selection	Runtime Limit	Model Degradation Feature
Auto-Sklearn 1.0	Yes	Yes	Yes	Yes	No
Auto-Sklearn 2.0	Yes	Yes	Yes	Yes	No
H2O	Yes	Yes	Yes	No	No
Weka	Yes	Yes	Yes	No	No
PPDaaS	Yes	Yes	Yes	Yes (Improved Customized)	Yes

### 2.3 SOFTWARE VULNERABILITY DETECTION

Software vulnerability refers to flaws or weaknesses within a software system within a software system [19][21]. These vulnerabilities stem from errors in coding, design, or system architecture. A software vulnerability is vital because it makes the platform vulnerable to attacks or other unauthorized actions. A system's vulnerability may be defined as a weakness because it tends to be susceptibility to attack from external factors such as viruses, hackers, or others [22]. Hence, software security and software vulnerability have long become seen as vital when developing software [26], as they pose serious risks to security [23].

In recent years, vulnerability exposure has attracted significant attention [24] and machine learning approaches are crucial in accurate vulnerability predictions[25]. Accordingly, several researchers studied various machine-learning techniques for detecting software vulnerabilities [22][23].

Pereira [31] studied the integration of different ML techniques to enhance the performance of vulnerability detection. The article states that Static Code Analysis has a good detection ratio, but it may give many false positives. Another study by Pereira and colleagues [32] examined the automatic detection of vulnerabilities. They are focused on static techniques to detect vulnerabilities in C functions. They also processed all features by Deep Graph Convolution Neural Network (DGCNN). They found this technique enhances the recall but decreases the precision. Similarly, Abdulrahman and colleagues [33] study information about the effect of code smells on software security and evaluate open-source integration under the Tomcat Software developed by Apache. They concluded that the relation between vulnerability and smells has not significantly impacted each other.

A study by Alves H and colleagues [34] focuses on the relationship between the vulnerabilities and the software metrics. Their analysis depends on 2875 patches to build a metric and vulnerability dataset for the classes, functions, and files. The researchers performed calculations on software metrics derived from their data sources. They then applied a correlation algorithm and conducted statistical tests on these metrics to determine any potential relationships between them and the preexisting vulnerabilities. The results show that software metrics discriminate the non-vulnerability and vulnerable functions. However, it is not possible to state a significant relationship between the number of vulnerabilities and metrics.

An article by Mederios N and colleagues [35] conducted two experiments. In the first experiment, multiple vulnerability prediction models are built using software metrics. In the second experiment, a consensus decision-making strategy is proposed for classifying the code levels, based on the output of different classification models. The results indicate that the rate of false alarms does not allow us to use machine learning algorithms.

The authors of [28] provide a dataset comprising vulnerabilities obtained from five prominent open-source C/C++ projects, namely Mozilla, Linux Kernel, Xen, httpd, and Glibc. They developed a method to collect necessary data that relies on an automated process utilizing vulnerabilities sourced from the Common Vulnerability and Exposures (CVE) Details website. The researchers obtained the relevant source code units for each vulnerability from the project repository, including both the vulnerable and patched versions. Following this, the researchers performed a comprehensive analysis of Software Metrics (SMs) on the code levels, and they

conducted Static Analysis Tools to get security alerts. They specifically targeted potential vulnerabilities and weaknesses. They gathered the dataset consisting of a total of 5214 vulnerabilities. To demonstrate its practicality, the researchers investigated the application of the dataset for training machine learning models designed to identify susceptible C/C++ functions. The findings of the study indicate that the dataset has practical applications and is a significant contribution to the field of software security, particularly for researchers in this area.

The study of [36] focuses on the alternations in software metrics after the resolution of a specific vulnerability by the developer. They applied comparative analysis on a total of 250 files obtained from Apache Tomcat and Apache CXF. Similarly, Siavvas's [37] thesis introduces software measures (such as complexity metrics) for distinguishing between vulnerabilities and investigating the possibility of linkages between vulnerabilities. While the analysis showed that there may be major interdependencies between different types of vulnerabilities, it also showed that software metrics are only useful as weak indications of specific security risks.

The researchers of [38] study the impact of data balancing techniques on the efficacy of software vulnerability prediction models. The experimental design comprises four distinct balancing methods and seven different classification algorithms. In addition to these, they used text mining-based features. Their results show that to obtain accurate estimations of software components that are prone to vulnerabilities, it is necessary to balance highly unbalanced datasets used for software vulnerability prediction. The utilization of software metric features proves to be advantageous in accurately characterizing vulnerable samples, while the implementation of ensemble-based classifiers yields superior outcomes.

The study in [39] investigates various code patterns utilized in the identification of vulnerabilities and assesses their efficacy in comparison to conventional software metrics. The researchers have developed an adaptive vulnerability prediction model through the analysis of security vulnerabilities in Java web applications, Apache CXF, and Tomcat. Machine learning and statistical techniques are employed to predict vulnerabilities by utilizing metrics and traceable trends as features.

Gupta A. and colleagues [40] used a collection of software measures that were converted into understandable regulations through the application of machine learning. 32 supervised techniques for machine learning have been evaluated for the three most common software system vulnerabilities, such as the *LawofDemeter*,

*BeanMemberShouldSerialize*, and *LocalVariablecouldBeFinal*. In this study, the J48 machine learning method was applied, and the vulnerability detection accuracy was up to 96%. They stated that the results are verified using tenfold cross-validation, and they have also been analyzed using statistical metrics like the ROC curve, Kappa statistics, Recall, Precision, etc.

The authors [41] put forth a methodology for the automated assessment of the efficacy of static code analyzers, utilizing Common Vulnerabilities and Exposures (CVE) reports as a basis for evaluation. The researchers evaluated five open-source static C code analyzers, as well as one commercial alternative. The findings indicate that while static C analyzers have demonstrated strong performance in benchmark tests involving artificial bugs, contemporary tools fail to detect a significant portion, ranging from 47% to 80% of vulnerabilities present in a benchmark dataset consisting of real-world programs.

The researchers in [42] proposed a method to evaluate the performance of machine learning (ML) based solutions for detecting software vulnerabilities. They conducted three experiments, focusing on ML techniques for detecting software vulnerabilities in different scenarios. The first experiment examined ML techniques for detecting a single type of vulnerability, while the second explored ML techniques for detecting multiple types of vulnerabilities within a dataset. The third experiment investigated ML techniques for detecting software vulnerabilities when faced with class imbalance, using varying an imbalance ratio. The empirical findings demonstrate the feasibility of utilizing machine learning techniques for software vulnerability detection.

Shukla and colleagues [43] proposed a model that statistically evaluating in the software vulnerability area. They considered with severity of vulnerability. Their experiment results showed that estimating the necessary information becomes more accurate than now.

The study done by Bahaa and colleagues [21] considered how to use techniques of learning such as machine learning and deep learning to detect vulnerabilities. They achieved results that indicate that deep learning and machine learning obtained positive outcomes in the field of detecting software vulnerability. Similarly, Bassi and colleagues [44] observed the effect of hyperparameter optimization on Software Vulnerability Models. They were focused on evaluating the effect of dual hyperparameter optimizations on Software Vulnerability Models. Their results showed

that increasing the productivity of Software Vulnerability models with dual HPO is more than 64% effective.

Pereira and colleagues [45] proposed a technique for gathering software vulnerabilities from databases and improving them with static data which are SAT alarms and SMs. The article studied by Ugrenovic and colleagues [46] evaluated the pros and cons of the one-class Support Vector Machine, Local Outlier Factor, and Isolation Forest to detect the distribution of the data. Their results showed that Isolation Forest is a superior method to the others.

Ugrenovic and colleagues [47] studied different approaches based on a collection of one-class autoencoders. This framework can be capable of both picture classification and image detection. Experimental results demonstrate its robustness to unknown pictures while maintaining accuracy on par with the state of the neural networks for datasets.

The correlation between software metrics and the existence of vulnerabilities is studied by Fonseca and colleagues [48]. The analysis is based on 2875 security patches, used to build a dataset with metrics and vulnerabilities for all the functions, classes, and files of 5750 versions of five widely used projects that are exposed to attacks: Linux Kernel, Mozilla, Xen Hypervisor, httpd, and glibc. the software metrics from their sources are calculated and correlation algorithms and statistical tests are used on these metrics to identify relations between them and the existing vulnerabilities. In another study, Fonseca et al, [49] used a representative and large dataset to examine the state-of-the-art vulnerability prediction techniques. Their obtained results showed that the dataset can be used to distinguish.

Using data accessible in the National Vulnerability Database and other open-source repositories, Bhandari and colleagues [50] proposed the CVEfixes dataset and an automated collecting method to compile it.

In the study of [29], researchers focused on metrics at various granularity levels (methods and java classes). They applied machine learning techniques to predict vulnerability in classes or methods. Their results show that the class-level metrics suppress within the 70 higher in recall and 75% higher in a prediction of vulnerability.

Clemente and colleagues [51] compare the traditional machine learning methods of naive Bayes, decision trees, support vector machines, and random forests with the deep learning technique. Their findings indicate that deep learning methods generate higher performance in predicting software security bugs.



The article by Salimi and colleagues [52] focused on studies of software metrics to predict software vulnerability. They focused on if they change the metrics, will the model's performance increase or decrease. They introduced new metrics named 'Vulnerable Slices' and tested and compared them with metrics that predict the vulnerability. Their results show that the accuracy of the model had been increased with the new metrics that they provide.

In the study of Medeiros and colleagues [20], the authors examined the effects of software metrics in detecting software vulnerability. Their results state the software metrics assist in identifying and predicting vulnerability.

Siavvas and colleagues [53] shed light on the ability of the software metrics to predict software vulnerability among different types. Their results indicate that software metrics are not adequate for predicting specific types of vulnerability. Campos and colleagues [54] provide a framework named Propheticus. Their opinion was that tools can be used to decrease the complexity of using machine learning techniques.

Kaya et al [55] examined the impacts of the classifiers, feature types, and data balancing techniques on the performance of the predicting software vulnerability model. Their results state that, smaller datasets have good performance at Random Forest on the opposite larger datasets have good performance at Rusboost Tree. As seen from the literature, many studies were conducted on software vulnerability detection that incorporates various ML techniques. However, none of the studies utilize no/low code platforms that enable fair comparison of ML algorithms and provide analysis of model degradation when new data is obtained from resources.

## **CHAPTER III**

### **RESEARCH METHODOLOGY**

During this research, existing literature in the field of Software Vulnerability, Machine Learning, No Code Platforms like AutoML are explored. Through this exploration, we identified several features that are missing in AutoML platforms. Hence, a platform that offers Automatic No Code Machine Learning Model Building named Prediction Platform as a Service (PPDaaS) is developed that includes run-time limit and model degradation analysis features. In the section below, we will discuss the details of the platform.

#### **3.1 PLATFORM PREDICTION AS A SERVICE PLATFORM**

In this section, the details of the platform that provides an automatic machine-learning solution that can be utilized without writing any code are explained. This platform allows users to apply various machine learning techniques without knowledge of coding or details of algorithms.

For the development of the platform and execution of the experiments, the following system configurations and libraries were used:

Hardware Specifications:

- **Processor:** Intel Core i5-8250U 1.60GHz
- **Memory:** 8GB RAM

Software and Libraries:

- **Programming Languages:** Python 3.7 for backend and Vue 2.6 for frontend
- **Backend Libraries:**
  - Flask for creating the web service
  - Scikit-learn (sklearn) for machine learning algorithms
  - XGBoost for gradient boosting framework
  - NumPy for numerical operations
  - Pandas for data manipulation

- **Frontend Libraries:**
  - XLSX for handling Excel files
  - Axios for making HTTP requests

The users can utilize the platform by dragging/dropping a CSV file into the interface and then the journey begins. Users can specify the random state and test size of their model by using the interface. The following sections will explain the details of the platform features.

**Random State:** The random state parameter is used for the control of the level of randomness in actions or algorithms. Specifying the random state in the machine learning algorithm enables users to gain more reliable outcomes. In our platform, the default value of random state is set to 40. Users can change the random state in a text box. If the user has not set the random state, it will use the default value.

**Test Size:** The test size refers to the subset of the dataset for evaluating the performance measurements of the trained model. In the machine learning process, users should create two data sets named trained and test sets. The test set is used for the evaluation of the trained model performance without applying any training process to it. Test size parameters are commonly stated as percentages. For example, a 40% test size means that 40% of the data is reserved for testing and the remaining is reserved for the training process. In our platform, the default value has been set to 0.4. Users can enter the values in the range of 0 to 1 that set the test size of a train test split.

PPDaaS provides a large set of classification and regression algorithms including Decision Tree Classifier, Random Forest Classifier, Gaussian Naïve Bayes, Linear Regression, Decision Tree Regressor, Random Forest Regressor, etc. as shown in Table 2 and Table 3. PPDaaS enables users to select these algorithms for the uploaded dataset. The platform automatically calculates all algorithm's results at once and gives the result to the user. It makes the user easily see which algorithm is more preferable for the dataset.

**Table 2: Classification Algorithms Provided by the PPDaaS Platform.**

Classifiers			
Decision Tree Classifier	Gaussian NB (Gaussian Naive Bayes)	Bagging Classifier	Linear Discriminant Analysis
Random Forest Classifier	Gradient Boosting Classifier	Extra Tree Classifier	Quadratic Discriminant Analysis
KNeighbors Classifier	MLP Classifier (Multi-Layer Perceptron Classifier)	Bernoulli NB (Bernoulli Naive Bayes)	Hist Gradient Boosting Classifier
SVC (Support Vector Classifier)	Ada Boost Classifier	SGD Classifier (Stochastic Gradient Descent Classifier)	Stacking Classifier
Voting Classifier	Label Spreading	XGB Classifier (XG Boost Classifier)	LGBM Classifier (Light GBM Classifier)
Self Training Classifier	Categorical NB	Nearest Centroid	Label Propagation

**Table 3: Regression Algorithms Provided by the PPDaaS Platform**

Regressors			
Linear Regression	KNeighbors Regressor	Bayesian Ridge	Ridge CV (Ridge Cross-Validated)
Ridge	SVR (Support Vector Regressor)	Extra Tree Regressor	LGBM Regressor (Light GBM Regressor)
Lasso	Gradient Boosting Regressor	Hist Gradient Boosting Regressor	XGB Regressor (XGBoost Regressor)
Elastic Net	MLP Regressor (Multi-Layer Perceptron Regressor)	Multi-Output Regressor	Stacking Regressor
Decision Tree Regressor	Ada Boost Regressor	NuSVR (Nu Support Vector Regressor)	Bagging Regressor
Random Forest Regressor			

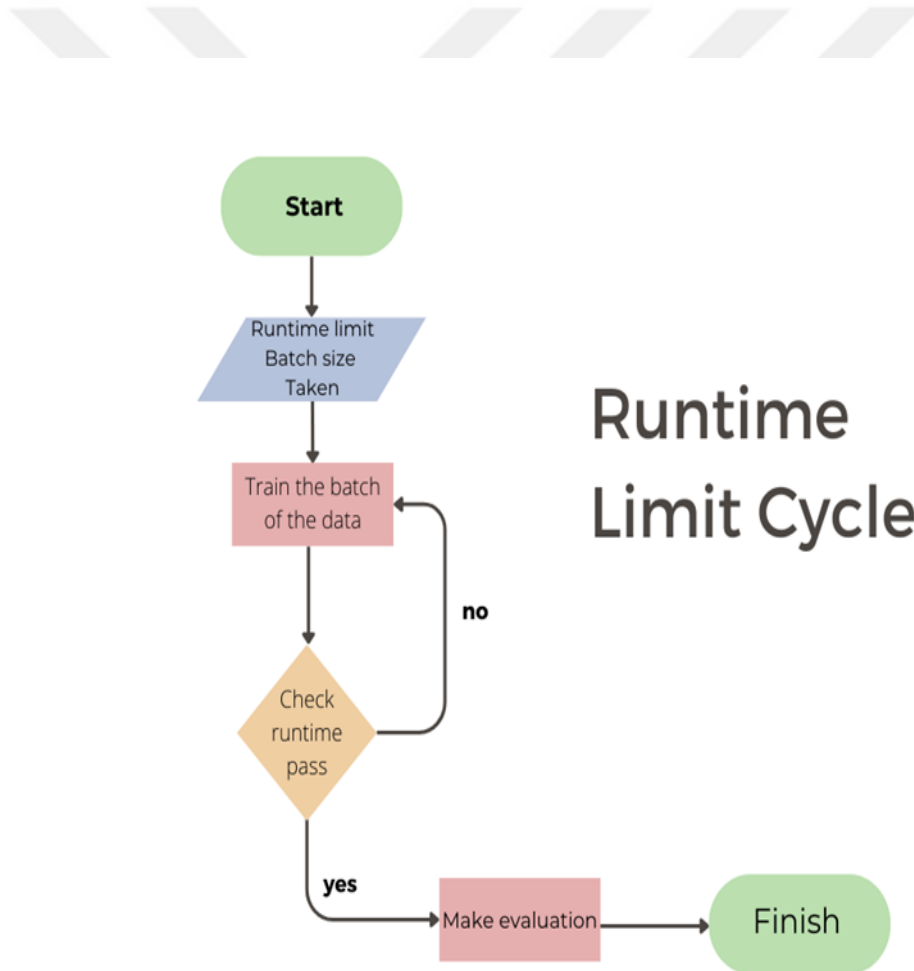
PPDaaS platform provides a very large set of feature selection techniques to enhance users selecting the feature selection technique to their own needs. Moreover, it enables the application of all feature selection models to data separately. The results of feature selection with any of the machine learning algorithm is listed. The list of feature selection techniques includes PCA, SelectKBest, RFE, SparsePCA, etc. as shown in Table 4.

**Table 4:** All Feature Selection Methods Provided by The Platform

Feature Selection Methods			
SelectKBest with f_classif	Select From Model with Lasso	RFE with Logistic Regression	Sparse PCA
SelectKBest with mutual_info_classif	SelectFromModel with Ridge	RFE with SVC	Mini Batch Sparse PCA
SelectKBest with chi2	SelectFromModel with Elastic Net	RFECV with estimator Random Forest Classifier	Truncated SVD
SelectKBest with f_regression	RFE with Decision Tree Classifier	RFECV with estimator Logistic Regression	Fast ICA
Sequential Feature Selector (SVC),	RFE with Random Forest Classifier	PCA	RFECV with estimator SVC
Sequential Feature Selector with Random Forest Classifier,	RFE with estimator MLP Classifier	Kernel PCA	RFECV with estimator MLP Classifier
Sequential Feature Selector with Gradient Boosting Classifier,	RFECV with estimator Decision Tree Classifier		

### 3.2 RUNTIME LIMIT FEATURE

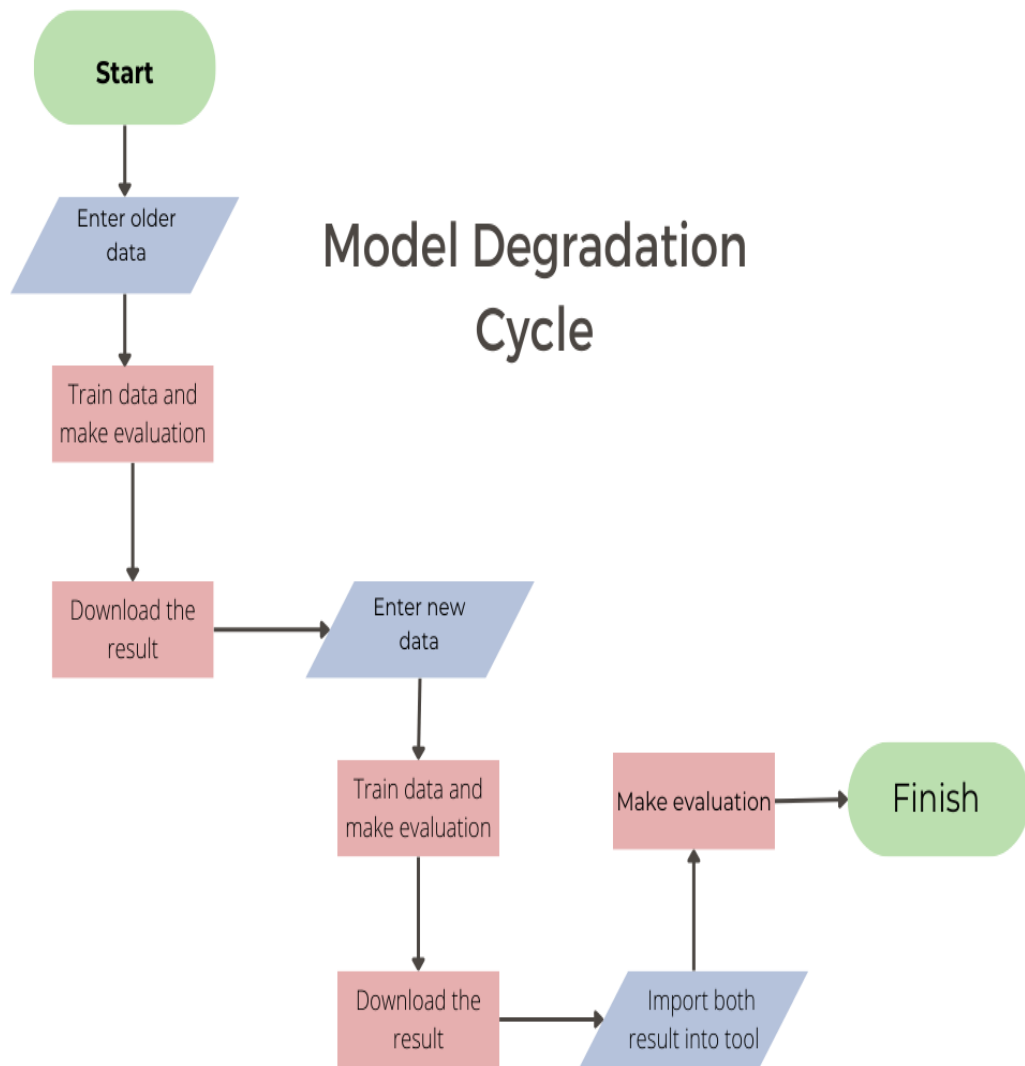
The PPDaaS platform allows users to set a limit on the duration of the machine learning process through the Runtime Limit feature. This functionality is based on two parameters which are time and batch size. Users can set the maximum time for the runtime and batch size. In the process, first, all data rows are randomized by the platform. The runtime limit works by fitting the model with the first batch size and controlling the time if it exceeds or not. If not, the second batch size will be operated, and the cycle will repeatedly proceed on other batches. This feature allows a user can make reliable comparisons between different machine learning algorithms. This makes users easily compare each machine-learning algorithm results within the same execution time.



**Figure 1:** Runtime Limit Cycle

### 3.3 MODEL DEGRADATION FEATURE

PPDaaS platform allows users to analyze their machine-learning results over a period as their data accumulates. For example, a user has data about COVID-19 in 2021-2022. By using PPDaaS, performance score results are taken and after a year, new data has been added to the previous data. After that, the new results can be taken from the platform. In that use case, the model degradation feature can help users compare the results of two machine-learning applications. The tool shows the model performance differences between the datasets.



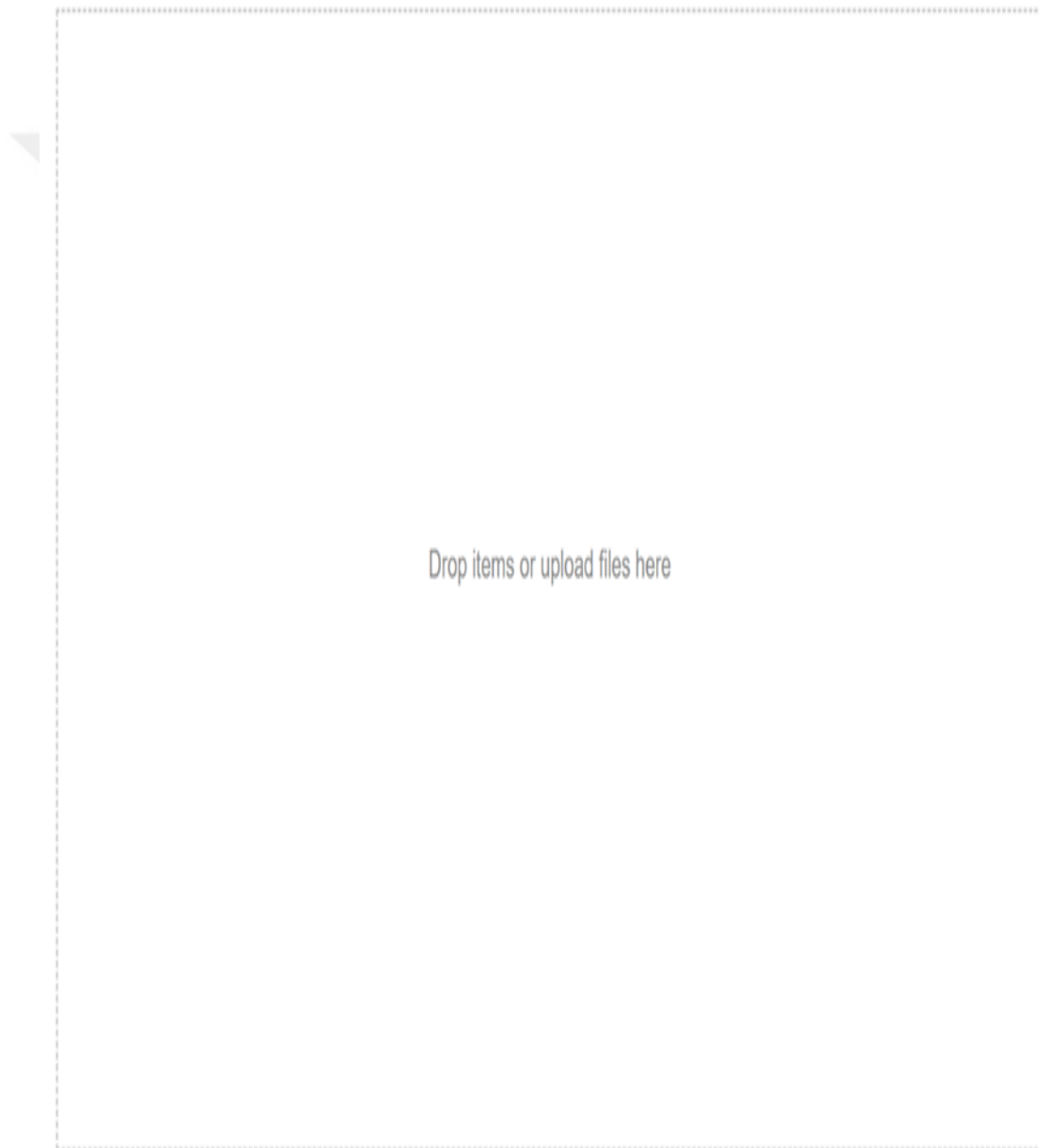
**Figure 2:** Model Degradation Cycle



### 3.4 INTERFACES OF THE PPDAS

#### 3.4.1 Import Data

The dataset to be used for the analysis can be uploaded by dragging/dropping the CSV file into the box located on the platform. The drag-drop feature reduces the importing data complexities and makes the platform more user-friendly. The main page of our platform is shown in Figure 3.



**Figure 3:** Main Page of The Platform

### 3.4.2 Specifying the Main Options

After the data import stage is completed, the tool displays a drag-and-drop options window. This page allows the user to specify the options by dragging items into a page or entering values into a text box. That step includes the user to customize parameters related to machine learning algorithms including random state, test size, and a runtime limit.

For The Analysis Drag Drop Items From The Below Menu To The Dropzone On Right

Main Options

Please Set The Experiment Parameters Below (Default values will be selected if no data is entered)

Random State:	Test Size:	<input type="checkbox"/> Runtime Limit:	Runtime Limit Batch Size:
40	0.4	5	10000

Drag Drop Options By Selecting Items From The Menu Below

- Select Target Feature
- Select Features to Drop
- Select Feature Selection Modal
- Model Check for Degradation

**Figure 4:** Options Section of The Platform

### 3.4.3 Target Feature Selection

By clicking the select target feature tab located on the right pane, the accordion menu opens with the feature names of the data. Using the drag/drop feature, all features of the dataset are shown on the left pane of the platform below the Select Target Feature Tab. Users can specify the target feature by drag/ drop feature.

Select Target Feature	
loc	v(g)
ev(g)	iv(g)
n	v
l	d
i	e
b	t
IOCode	IOComment
IOBlank	locCodeAndComment
uniq_Op	uniq_Opnd
total_Op	total_Opnd
branchCount	defects

**Figure 5:** Target Feature Selection Section

### 3.4.4 Drop Features

During the preprocessing step, the user may want to drop some irrelevant features from the dataset. With the drag/drop feature, the user can easily drop features by dragging the feature onto the drop box of the platform located on the right side of the screen.

Select Features to Drop	
loc	v(g)
ev(g)	iv(g)
n	v
l	d
i	e
b	t
IOCode	IOComment
IOBlank	locCodeAndComment
uniq_Op	uniq_Opnd
total_Op	total_Opnd
branchCount	defects

**Figure 6:** Drop Feature Section

### 3.4.5 Feature Selection Algorithms

At this step, the user has already selected the main options, target feature, and features to be dropped. Hence, the user can select the feature selection method and select the feature count for the machine learning algorithm. Users can select just one or multiple feature selection methods. In addition to that, users can select all feature selection methods at once. Each feature selection method is used by each machine learning algorithm as a result.

Select Feature Selection Modal	
<input type="checkbox"/> Select All	SelectKBest(f_classif)
SelectKBest(mutual_info_classif)	SelectKBest(chi2)
SelectKBest(f_regression)	SequentialFeatureSelector(SVC)
SequentialFeatureSelector(RandomFore	SequentialFeatureSelector(GradientBoo
SelectFromModel(estimator=Lasso)	SelectFromModel(estimator=Ridge)
SelectFromModel(estimator=ElasticNet)	RFE(estimator=DecisionTreeClassifier)
RFE(estimator=RandomForestClassifier)	RFE(estimator=LogisticRegression)
RFE(estimator=SVC)	RFE(estimator=MLPClassifier)
RFECV(estimator=DecisionTreeClassifie	RFECV(estimator=RandomForestClassi
RFECV(estimator=LogisticRegression)	RFECV(estimator=SVC)
RFECV(estimator=MLPClassifier)	PCA()
KernelPCA(kernel="linear")	SparsePCA()
MiniBatchSparsePCA()	TruncatedSVD()
FastICA()	

**Figure 7:** Feature Selection Section

### 3.4.6 Label Encoding

If the dataset contains categorical data, the label encoder comes into play. Label encoder turns categorical data such as words or names, it turns these words into numbers. For example, when confronted with categories, say, "Small", "Medium", or "Large", this tool gracefully translates them into a numerical format think 0, 1, and 2.

### 3.4.7 Building the Machine Learning Process

Once the previous options are selected, users can submit the dataset and the options to the platform. Platform starts to evaluate machine learning and feature selection algorithms at the backend side. The user will see the loading screen at the end of the calculation.

### 3.4.8 Result

After the machine learning process is completed, the tool provides an analysis result as a downloadable table. The table consists of columns of model type, accuracy/mean squared error, precision/root mean squared error, recall/mean absolute error, F1 score/R-squared, test size, random state, batch size, runtime limit, drop columns, selected column, and feature selection model.

### 3.4.9 Download Output

PPDaaS tool provides users with the capability to download the analysis result as a table. This feature allows for further examination of the data or presentation of the findings to relevant stakeholders.

Index	Model	Type	Accuracy/Mean Squared Error	Precision/Root Mean Squared Error	Recall/Mean Absolute Error	F1 Score/R-squared	Test Size	Random State	Batch Size	Runtime Limit	Drop Columns	Selected Column	Feature Selection Model	Selected Feature Variable
14	StackingClassifier	Classifier	0.9515340565	0.9371434941	0.9515340565	0.9367494091	0.25	42	-	-	Description COMMIT_HASH COMMIT_DATE COMMIT_YEAR VULNERABILITY_CVE VULNERABILITY_YEAR VULNERABILITY_CWE VULNERABILITY_CATEGORY ID_Function P_ID Visibility Complement BeginLine EndLine NameMethod ID_Class ID_File FilePath Patched Occurrence R_ID Affected grouped_multiclass_label binary_label	multiclass_label	RFE(CV(estimator=DecisionTreeClassifier(), max_features_to_select=20))	28
16	StackingClassifier	Classifier	0.9515340565	0.930994417	0.9515340565	0.9364293599	0.25	42	-	-	Description COMMIT_HASH COMMIT_DATE COMMIT_YEAR VULNERABILITY_CVE VULNERABILITY_YEAR VULNERABILITY_CWE VULNERABILITY_CATEGORY ID_Function P_ID Visibility Complement BeginLine EndLine NameMethod ID_Class ID_File FilePath Patched Occurrence R_ID Affected grouped_multiclass_label binary_label	multiclass_label	SelectKBest(mutual_info_func=k=20)	28
17	StackingClassifier	Classifier	0.9514074994	0.9369521134	0.9514074994	0.9365043879	0.25	42	-	-	Description COMMIT_HASH COMMIT_DATE COMMIT_YEAR VULNERABILITY_CVE VULNERABILITY_YEAR VULNERABILITY_CWE VULNERABILITY_CATEGORY ID_Function P_ID Visibility Complement BeginLine EndLine NameMethod ID_Class ID_File FilePath Patched Occurrence R_ID Affected grouped_multiclass_label	multiclass_label	RFE(CV(estimator=LogisticRegression(), max_features_to_select=20))	28

Figure 8: Model Performance Modal of the Platform

### 3.4.10 Model Degradation Analysis

After all the processing is finished, the tool provides a feature that enables comparison of the analysis results. The task can be done by comparing two machine learning analysis results which are previously taken from the platform. After a while new data might be added to the previously analyzed dataset and the results obtained for the new dataset and the previously obtained results could be compared for analyzing model degradation. This feature shows the stability and reliability of the machine learning models as data gets larger in time.

For The Analysis Drag Drop Items From The Below Menu To The Dropzone On Right

Main Options

Please Set The Experiment Parameters Below (Default values will be selected if no data is entered)

Random State:	Test Size:	<input type="checkbox"/> Runtime Limit:	Runtime Limit Batch Size:
42	0.25	5	10000

Drag Drop Options By Selecting Items From The Menu Below

Select Target Feature

Select Features to Drop

Select Feature Selection Modal

Model Check for Degradation

**Figure 9:** Model Degradation Tab

### **3.4.11 Model Performance Metrics Provided by PPDaaS Platform**

The applied machine learning algorithms are evaluated based on various performance metrics such as Accuracy, Mean Squared Error, Precision Root Mean Squared Error, Recall, Mean Absolute Error, F1 Score, and R-squared. All evaluation metrics are explained below.

#### **3.4.11.1 Accuracy**

Accuracy is a metric used for evaluating classification models that measure the proportion of classified data correctly over the amount of data in the dataset.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

#### **3.4.11.2 Mean Squared Error (MSE)**

Mean Squared Error is used for evaluating regression models that measure the average squared difference between the predicted and actual values.

$$\text{MSE} = \frac{1}{n} * \sum (y_{\text{pred}} - y_{\text{actual}})^2$$

#### **3.4.11.3 Precision**

Mean Squared Error is used for evaluating regression models that measure the average squared difference between the predicted and actual values.

$$\text{MSE} = \frac{1}{n} * \sum (y_{\text{pred}} - y_{\text{actual}})^2$$

#### **3.4.11.4 Root Mean Squared Error (RMSE)**

The RMSE quantifies the typical difference between the model's predictions and the observed data. In mathematical terms, it is the residuals' standard deviation. The residuals are the gaps between the data points and the regression line.

$$\text{RMSE} = \sqrt{\frac{1}{n} * \sum (y_{\text{actual}} - y_{\text{predicted}})^2}$$

#### **3.4.11.5 Recall**

Recall, also known as True Positive Rate or Sensitivity, is a metric used in statistics and machine learning to assess how well a model can accurately identify all relevant occurrences within a dataset.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$



#### **3.4.11.6 Mean Absolute Error (MAE)**

Absolute error is the difference between a prediction and its true value. MAE measures group errors by averaging absolute errors from predictions and observations. The L1 loss function is MAE.

$$\text{MAE} = (1/n) * \sum |y_{\text{pred}} - y_{\text{actual}}|$$

#### **3.4.11.7 F1 Score**

The F1 score is a metric for the evaluation of models in machine learning that indicates how accurate they are. It brings together a model's precision and recall scores in a single value.

$$\text{F1 Score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$$

#### **3.4.11.8 R-Squared(R<sup>2</sup>)**

R<sup>2</sup> is a statistical measure to assess how well a regression model fits the data. R-squared can take on any value from zero to one. If the data are a perfect fit for the model, then the projected value and the actual value will be identical, and the R-squared value will equal 1.

$$\text{R-squared} = 1 - (\text{Sum of Squares of Residuals}) / (\text{Total Sum of Squares})$$

### **3.4.12 Survey About the Usability of PPDaaS**

To collect the opinions of professionals about the PPDaaS platform, a usability study is performed. Several demo videos were prepared that show how the platform works, so everyone can understand it. The videos were shown with both verbal and multichoice survey options to software professionals and their opinions were gathered. Afterward, to create the survey questions, an initial draft was generated using zero-shot prompting [56] within Baize LLM [57], with version baize-v2-7b running as a demo under HuggingFace[58]. Then, the AI-generated survey questions and choices were manually edited for improvement. The details of the prompts and the survey questions used are given in Appendix 1.

Then the videos and the survey are shared with software professionals and their feedback is analyzed to gain insights into the usefulness and effectiveness of the PPDaaS platform.

## **CHAPTER IV**

### **EXPERIMENTS AND RESULTS**

The developed platform is used to analyze two datasets. Initially, the dataset used for benchmarking AutoML platforms is used to test the features of the system. Afterward, software vulnerability data is analyzed to see how model degradation can be analyzed. After the experiments, the platform is tested by potential users to gather feedback about its usability and effectiveness. Below the details of the experiments are explained.

#### **4.1 DATA OVERVIEW**

##### **4.1.1 Benchmarking AutoML Data**

PPDaaS (Prediction Platform as a Service) platform is a powerful platform that enables the application of a large set of machine learning algorithms and feature selection techniques at once efficiently. Thus, to test the capabilities of this platform, we used a benchmarking dataset used for AutoM platforms from the Open Machine Learning service for the first experiment.

OpenML is a service named Open Machine Learning, a free public software service for machine learning research. Service provides sharing datasets, experiments, and algorithms that are done by researchers and practitioners. In addition to these, OpenML is a platform where researchers and practitioners can compare their results with others.

The dataset obtained from the OpenML platform is named an adult dataset, which is used to analyze how an individual's annual income is influenced by a multitude of factors. The factors that influence it include the individual's education level, age, gender, occupation, and other relevant variables. This dataset is frequently referenced in K-Nearest Neighbors (KNN) literature. The dataset features can be seen in Figure 10.

id	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	class	
0	1	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40	United-States	<=50K
1	2	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	50	United-States	<=50K
2	3	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	40	United-States	>50K
3	4	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	40	United-States	>50K
4	5	18	?	103497	Some-college	10	Never-married	?	Own-child	White	Female	0	0	30	United-States	<=50K
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
48837	48838	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Female	0	0	38	United-States	<=50K
48838	48839	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	40	United-States	>50K
48839	48840	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female	0	0	40	United-States	<=50K
48840	48841	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male	0	0	20	United-States	<=50K
48841	48842	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Female	15024	0	40	United-States	>50K

**Figure 10: OpenML Data Overview**

The attribute income has two category levels: Individuals with an annual income of less than or equal to \$50,000 and those with an annual income exceeding \$50,000. The number of attributes is 14. The other features of the dataset such as age, education, or occupation are the demographic characteristics and additional attributes utilized to delineate an individual. The potential for predicting an individual's income level based on their personal information can be investigated.

#### 4.1.2 Vulnerability Experiment Data

The software vulnerability dataset is "The dataset, which was created and made openly available on GitHub by Pereira and colleagues [28]. For this dataset, they created an automated process for building a dataset build with vulnerability metadata, including SMs and SAT Alerts. The steps for retrieving data as listed below.

The first step is to retrieve reported vulnerabilities from the online platforms. They started the first step by querying the CVE Details website to be able to take the list of published vulnerabilities for chosen projects. For the Vulnerability data, they considered and saved the metadata consisting of.

- Common Vulnerabilities and Exposures (CVE) identifier: A unique identifier that is assigned to each vulnerability.
- Common Vulnerability Scoring System (CVSS) score: The severity of the vulnerability based on its environment and characteristics. •impact on confidentiality, integrity, and availability
- Exploit difficulty
- Authentication Requirement
- Zero or more vulnerability types
- Common Weakness Enumeration (CWE) maps a vulnerability to a specific weakness.

The second step was, that they used commit hashes which are gained from collecting the metadata of the vulnerabilities, to be able to collect the vulnerable version and the normal version for each. They focused on Git repositories to be able to find the code version of the vulnerability and the correct version before patching. In short, they have two types of affected files, which are vulnerable and non-vulnerable (after patching).

In the third step, Pereira, and colleagues [28] used a third-party tool to perform SCA in C++ code levels which are function file classes, and they extracted SAT alerts and Software Metrics. They used two security analysis tools (SATs) named Flawfinder3 v2.0.10 and Cppcheck4 v1.82 to produce security alerts. These tools are used to take the files as input and give the output as a CSV file.

After all these in the last step, they collected all the vulnerability metadata and generated the SAT alerts and software metrics for the vulnerable and non-vulnerable data.

The creation of a large dataset includes Mozilla (i.e., the source code for the Firefox browser), the Linux Kernel, the Xen Hypervisor, the Apache HTTP Server (httpd), and the GNU C Library (Glibc). That dataset contains projects with different complexities, security needs, and sizes. For instance, 21.6 million lines of code came from Linux Kernal, and 24.6 million lines of code came from Mozilla. In contrast, others were much smaller than comparing xen, httpd, and glibc.

To be able to apply machine learning algorithms they applied simple SQL queries that slice and dice the data. So that the dataset that we use in our thesis has been created. It consists of two features as follows.

- Various levels of code units which means the different granularity levels correspond to different organizational units of the software. For example, functions files and classes.
- The number of alert types per function in the SAT system varies, as each SAT can generate zero or more alerts for each specific alert type. The number of alerts per type was extracted for each function to identify potential security issues associated with the code.

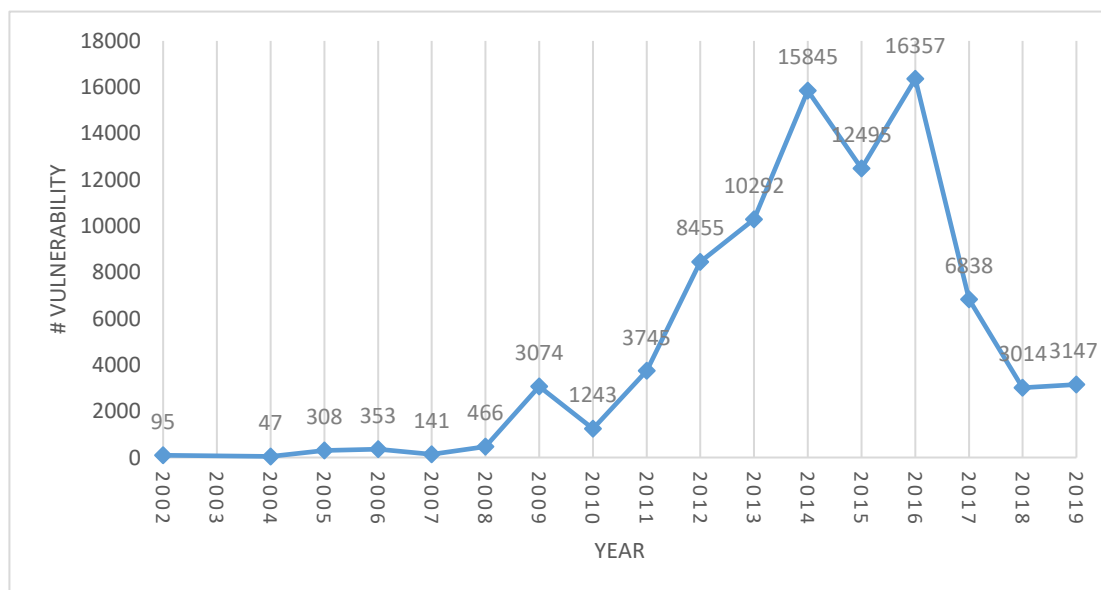
The dataset comprised a total of 126 different SAT alert kinds from Flawfinder and a collection of 28 software metrics for each function class or file. All features have been shown in the Appendix 2. The target class is considered as multiclass classification. This represents each function, file or class is vulnerable with a non-defined category or vulnerable in any category.

## 4.2 VULNERABILITY DATASET EXPERIMENT

The PPDaaS is configured for the Vulnerability dataset that we discussed in the previous section. We applied experiments on various levels of data granularity namely functions, class, and files.

### 4.2.1 Functions Code Level

To be able to observe the model degradation of the dataset in a period, we separated the data according to vulnerability years as shown in Figure 11.



**Figure 11:** Reported Vulnerabilities Concerning Year

Figure 11 shows that the reported vulnerabilities started low in 2002, before a significant increase in 2005, and decreased in 2004. Until 2014, it caught an increasing trend and reached 15845. Before reaching the highest point, it slightly decreases before 2016. After that, it continues the alternating periods of decrease and slight increase. Thus, we separated the data into 3 parts based on the vulnerability count. in the years 2002-2014, 2002-2016, and 2002-2019 as shown in Table 5.

**Table 5: Vulnerability Count in Each Period**

2002-2014	2002-2016	2002-2019
44064	72916	85915

The random state is set to 42 and the test size is set to 0.25. Target (Class) Feature set to multiclass label similar to the Pereira and colleagues [28] and dropped columns are listed in Appendix 2. We drop these features because of they are some of these are unique identifiers that do not provide any meaningful information and some of these are metadata that do not contribute to the target prediction.

To observe the effect of various feature selection methods on this dataset and to reduce its dimensions, I applied different combinations of feature selection. After evaluating these combinations, I determined that selecting 28 features on the platform struck an optimal balance between training efficiency and model accuracy.

#### 4.2.2 Class Code Level and File Code Level

We applied the same experiment settings as the Function Code Level settings on the PPDaaS. To be described briefly, we set the random state value as 42 and the test size as 0.25. Our evaluation has been conducted on the multiclass\_label feature. The feature selection count option on the platform is set to 28, which is used to determine which feature selection approach is best for the data and to apply feature selection methods for lowering the dataset's dimensions.

<b>Random State:</b> 42	<b>Drop Columns:</b> Description COMMIT_HASH COMMIT_DATE COMMIT_YEAR	<b>Feature Selection Model:</b>
<b>Test Size:</b> 0.25	VULNERABILITY_CVE VULNERABILITY_YEAR VULNERABILITY_CWE	SelectKBest(f_classif)
<b>Processed Column:</b> multiclass_label	VULNERABILITY_CATEGORY ID_File P_ID FilePath Patched Occurrence R_ID Affected binary_label	SelectKBest(mutual_info_classif)
	grouped_multiclass_label	SelectKBest(chi2)
		SelectKBest(f_regression)
		<b>Feature Selection Model Threshold Value:</b> 28

**Figure 12: Class Code Level Settings Display in Platform**

### 4.3 BECHMARKING AUTO ML EXPERIMENT

The PPDaaS platform is configured for the OpenML adult dataset that we discussed in the previous section in the Dataset part. The platform is configured with a random state of 40.

---

#### No Code Platform for Prediction Model Degradation(PMDaaS)

Random State: 40	Drop Columns:	Feature Selection Model:
Test Size: 0.4	relationship workclass id	SelectKBest(f_classif)
Runtime Limit: 600	marital-status	SelectKBest(mutual_info_classif)
Runtime Limit Batch Size: 1000		SelectKBest(chi2)
Processed Column: class		SelectKBest(f_regression)
		Feature Selection Model Threshold Value: 4

---

**Figure 13:** Optimized Values Display in Platform

To be able to split data, the platform configured the test size with 0.4. In addition to these, the platform serves runtime limit and runtime batch size to make fair comparisons. We set the runtime limit as 10 minutes same as [14]. Therefore, we can do a benchmarking analysis on our system by comparing other autoML tools. Drop columns are selected as id, relationship, marital status, and work class because id values do not carry any meaningful information, relationship, and marital status might have multicollinearity and occupation might give a more detailed or relevant insight about the work rather than work class. Lastly, we selected all feature selection models to be applied to the data, and to observe the effect of various feature selection methods on this dataset and to reduce its dimensions, I applied different combinations of feature selection. After evaluating these combinations, I determined that selecting feature selection counts 4 and the target feature as a class. Figure 14 shows that the dataset has a total of nine categorical variables work class, relationship, marital status, education, race, sex, occupation, native country, and class.

	id	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	class	
	0	1	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40	United-States	<=50K
	1	2	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	50	United-States	<=50K
	2	3	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	40	United-States	>50K
	3	4	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	40	United-States	>50K
	4	5	18	?	103497	Some-college	10	Never-married	?	Own-child	White	Female	0	0	30	United-States	<=50K
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	48837	48838	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Female	0	0	38	United-States	<=50K
	48838	48839	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	40	United-States	>50K
	48839	48840	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female	0	0	40	United-States	<=50K
	48840	48841	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male	0	0	20	United-States	<=50K
	48841	48842	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Female	15024	0	40	United-States	>50K

**Figure 14:** Overview of The OpenML Adult Data

In this experiment, we integrate the label encoder into our platform which uses the label encoder for encoding categorical data to numerical data. This applied to all categorical data of the dataset which are education, race, sex, occupation, and native country. Label encoder is a data preprocessing technique that is used for converting categorical data to unique numerical values. After the machine learning process has been finished, the testing dataset is used to evaluate the model to test its performance. To determine how well the model can predict the class attribute, evaluation metrics such as precision, recall, F1-score, and ROC-AUC are computed.



## 4.4 VULNERABILITY EXPERIMENT RESULTS

### 4.4.1 Function Code Level

#### 4.4.1.1 Results Between the Years 2002 to 2014

The top results of the function code level experiment between 2002 to 2014 according sorted by Accuracy, Precision, Recall, and F1 Score are shown in Table 6, Table 7, Table 8, and Table 9. All the results are listed in Appendix 3.

**Table 6:** Results of the 2002-2014 Period Sorted by Accuracy

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Stacking Classifier	0.950436	0.938478	0.950436	0.939923	Sparse PCA
Random Forest Classifier	0.950436	0.938336	0.950436	0.939579	PCA
Random Forest Classifier	0.950254	0.938236	0.950254	0.939873	Mini Batch Sparse PCA
Stacking Classifier	0.950254	0.938117	0.950254	0.939617	Mini Batch Sparse PCA
Stacking Classifier	0.950163	0.937936	0.950163	0.939464	PCA

**Table 7:** Results of the 2002-2014 Period Sorted by Precision

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
SVC	0.947712	0.940082	0.947712	0.923982	SelectKBest with f_classif
SVC	0.947712	0.940082	0.947712	0.923982	SelectKBest with mutual_info_classif
SVC	0.947712	0.940082	0.947712	0.923982	SelectKBest with f_regression
SVC	0.947712	0.940082	0.947712	0.923982	RFE with estimator Decision Tree Classifier
SVC	0.947712	0.940082	0.947712	0.923982	RFE with estimator Random Forest Classifier

**Table 8: Results of the 2002-2014 Period Sorted by Recall**

<b>Model</b>	<b>Accuracy</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>	<b>Feature Selection Model</b>
Stacking Classifier	0.950436	0.938478	0.950436	0.939923	Sparse PCA
Random Forest Classifier	0.950436	0.938336	0.950436	0.939579	PCA
Random Forest Classifier	0.950254	0.938236	0.950254	0.939873	Mini Batch Sparse PCA
Stacking Classifier	0.950254	0.938117	0.950254	0.939617	Mini Batch Sparse PCA
Stacking Classifier	0.950163	0.937936	0.950163	0.939464	PCA

**Table 9: Results of the 2002-2014 Period Sorted by F1 Score**

<b>Model</b>	<b>Accuracy</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>	<b>Feature Selection Model</b>
Stacking Classifier	0.950436	0.938478	0.950436	0.939923	Sparse PCA
Random Forest Classifier	0.950254	0.938236	0.950254	0.939873	Mini Batch Sparse PCA
Stacking Classifier	0.950254	0.938117	0.950254	0.939617	Mini Batch Sparse PCA
Random Forest Classifier	0.950436	0.938336	0.950436	0.939579	PCA
Stacking Classifier	0.950163	0.937936	0.950163	0.939464	PCA

#### 4.4.1.2 Results Between the Years 2002 to 2016

The top results of the function code level experiment between 2002 to 2016 according sorted by Accuracy, Precision, Recall, and F1 Score are shown in Table 10, Table 11, Table 12, and Table 13. All the results are listed in Appendix 4.

**Table 10:** Results of the 2002-2016 Period Sorted by Accuracy

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Random Forest Classifier	0.957143	0.94753	0.957143	0.947079	PCA
Random Forest Classifier	0.957086	0.94745	0.957086	0.947523	RFE with estimator MLP Classifier
Random Forest Classifier	0.957029	0.94733	0.957029	0.94732	SelectKBest with mutual_info_classif
Random Forest Classifier	0.957029	0.94733	0.957029	0.94732	RFE with estimator SVC
Stacking Classifier	0.956914	0.947139	0.956914	0.946003	RFE with estimator SVC

**Table 11:** Results of the 2002-2016 Period Sorted by Precision

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Random Forest Classifier	0.957143	0.94753	0.957143	0.947079	PCA
Random Forest Classifier	0.957086	0.94745	0.957086	0.947523	RFE with estimator MLP Classifier
Random Forest Classifier	0.957029	0.94733	0.957029	0.94732	SelectKBestwith mutual_info_classif
RandomForestClassifier	0.957029	0.94733	0.957029	0.94732	RFE with estimator SVC
StackingClassifier	0.956914	0.947139	0.956914	0.946003	RFE with estimator SVC

**Table 12: Results of the 2002-2016 Period Sorted by Recall**

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Random Forest Classifier	0.957143	0.94753	0.957143	0.947079	PCA
Random Forest Classifier	0.957086	0.94745	0.957086	0.947523	RFE with estimator MLP Classifier
Random Forest Classifier	0.957029	0.94733	0.957029	0.94732	SelectKBest with mutual_info_classif
Random Forest Classifier	0.957029	0.94733	0.957029	0.94732	RFE with estimator SVC
Stacking Classifier	0.956914	0.947139	0.956914	0.946003	RFE with estimator SVC

**Table 13: Results of the 2002-2016 Period Sorted by F1 Score**

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Random Forest Classifier	0.957086	0.94745	0.957086	0.947523	RFE with estimator MLP Classifier
Random Forest Classifier	0.957029	0.94733	0.957029	0.94732	SelectKBest with mutual_info_classif
Random Forest Classifier	0.957029	0.94733	0.957029	0.94732	RFE with estimator SVC
Random Forest Classifier	0.956629	0.946666	0.956629	0.947231	SelectKBest with f_regression
Random Forest Classifier	0.957143	0.94753	0.957143	0.947079	PCA

#### 4.4.1.3 Results Between the Years 2002 to 2019

The top results of the function code level experiment between 2002 to 2019 sorted by Accuracy, Precision, Recall, and F1 Score are shown in Table 14, Table 15, Table 16, and Table 17. All the results are listed in Appendix 5.

**Table 14:** Results of the 2002-2019 Period Sorted by Accuracy

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Stacking Classifier	0.951068	0.936839	0.951068	0.937856	Sparse PCA
Stacking Classifier	0.951068	0.936765	0.951068	0.937641	Mini Batch Sparse PCA
Stacking Classifier	0.951022	0.936595	0.951022	0.937392	RFECV with estimator SVC
Random Forest Classifier	0.950975	0.937009	0.950975	0.938623	PCA
Stacking Classifier	0.950929	0.936359	0.950929	0.937219	RFE with estimator Random Forest Classifier
Stacking Classifier	0.950882	0.936118	0.950882	0.936743	RFECV with estimator Decision Tree Classifier

**Table 15:** Results of the 2002-2019 Period Sorted by Precision

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
SGD Classifier	0.949253	0.951828	0.949253	0.924633	RFECV with estimator Random Forest Classifier
Random Forest Classifier	0.950975	0.937009	0.950975	0.938623	PCA
Stacking Classifier	0.951068	0.936839	0.951068	0.937856	Sparse PCA
Random Forest Classifier	0.950743	0.936812	0.950743	0.93886	SelectKBest with mutual_info_classif
Stacking Classifier	0.951068	0.936765	0.951068	0.937641	Mini Batch Sparse PCA
Random Forest Classifier	0.950696	0.936706	0.950696	0.938778	SelectKBest with f_regression

**Table 16:** Results of the 2002-2019 Period Sorted by Recall

<b>Model</b>	<b>Accuracy</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>	<b>Feature Selection Model</b>
Stacking Classifier	0.951068	0.936839	0.951068	0.937856	Sparse PCA
Stacking Classifier	0.951068	0.936765	0.951068	0.937641	Mini Batch Sparse PCA
Stacking Classifier	0.951022	0.936595	0.951022	0.937392	RFECV with estimator SVC
Random Forest Classifier	0.950975	0.937009	0.950975	0.938623	PCA
Stacking Classifier	0.950929	0.936359	0.950929	0.937219	RFE with estimator Random Forest Classifier
Stacking Classifier	0.950882	0.936118	0.950882	0.936743	RFECV with estimator Decision Tree Classifier

**Table 17:** Results of the 2002-2019 Period Sorted by F1 Score

<b>Model</b>	<b>Accuracy</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>	<b>Feature Selection Model</b>
Random Forest Classifier	0.950743	0.936812	0.950743	0.93886	SelectKBest with mutual_info_classif
Random Forest Classifier	0.950696	0.936706	0.950696	0.938778	SelectKBest with f_regression
Random Forest Classifier	0.950975	0.937009	0.950975	0.938623	PCA
Random Forest Classifier	0.950556	0.936413	0.950556	0.938579	RFE with estimator SVC
Bagging Classifier	0.949718	0.935522	0.949718	0.938573	RFECV with estimator Decision Tree Classifier
Random Forest Classifier	0.950649	0.936513	0.950649	0.938545	RFECV with estimator Random Forest Classifier

#### 4.4.2 Results of the Class Level

The top results of the class code level experiment sorted by Accuracy, Precision, Recall, and F1 Score are shown in Table 18, Table 19, Table 20, and Table 21. All the results are listed in Appendix 6.

**Table 18: Results Sorted by Accuracy**

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Ridge Classifier CV	0.948179	0.932614	0.948179	0.928172	SelectKBest with f_classif
Ridge Classifier CV	0.948179	0.932614	0.948179	0.928172	SelectKBest with mutual_info_classif
Ridge Classifier CV	0.948179	0.932614	0.948179	0.928172	SelectKBest with f_regression
Ridge Classifier CV	0.948179	0.932614	0.948179	0.928172	RFE with estimator Decision Tree Classifier
RidgeClassifier CV	0.948179	0.932614	0.948179	0.928172	RFE Random Forest Classifier

**Table 19: Results Sorted by Precision**

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Random Forest Classifier	0.947347	0.936937	0.947347	0.940376	RFECV with estimator SVC
Bagging Classifier	0.946722	0.936175	0.946722	0.939751	RFE with estimator Random Forest Classifier
Bagging Classifier	0.947138	0.936016	0.947138	0.939561	SelectKBest with mutual_info_classif
Random Forest Classifier	0.946722	0.935988	0.946722	0.939587	RFECV with estimator Decision Tree Classifier
Bagging Classifier	0.946514	0.935986	0.946514	0.939597	RFECV with estimator SVC

**Table 20: Results Sorted by Recall**

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Ridge Classifier CV	0.948179	0.932614	0.948179	0.928172	SelectKBest with f_classif
Ridge Classifier CV	0.948179	0.932614	0.948179	0.928172	SelectKBest mutual_info_classif
Ridge Classifier CV	0.948179	0.932614	0.948179	0.928172	SelectKBest with f_regression, k=28
Ridge Classifier CV	0.948179	0.932614	0.948179	0.928172	RFE with estimator Decision Tree Classifier
Ridge Classifier CV	0.948179	0.932614	0.948179	0.928172	RFE with estimator Random Forest Classifier

**Table 21: Results Sorted by F1 Score**

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Random Forest Classifier	0.9473 47	0.9369 37	0.947 347	0.940 376	RFECV with estimator SVC
Bagging Classifier	0.9467 22	0.9361 75	0.946 722	0.939 751	RFE with estimator Random Forest Classifier
Bagging Classifier	0.9465 14	0.9359 86	0.946 514	0.939 597	RFECV with estimator SVC
Random Forest Classifier	0.9467 22	0.9359 88	0.946 722	0.939 587	RFECV with estimator Decision Tree Classifier
Bagging Classifier	0.9471 38	0.9360 16	0.947 138	0.939 561	SelectKBest with mutual_info_classif

#### 4.4.3 File Code Level Results

The top results of the file code level experiment according to sorted by Accuracy, Precision, Recall, and F1 Score are shown in Table 22, Table 23, Table 24, and Table 25. All the results are listed in Appendix 7.

**Table 22: File Code Level Results Sorted by Accuracy**

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
XGB Classifier	0.9547 95	0.9448 24	0.9547 95	0.9364 33	SelectKBest with f_classif
XGB Classifier	0.9547 95	0.9448 24	0.9547 95	0.9364 33	SelectKBest with mutual_info_classif
XGB Classifier	0.9547 95	0.9448 24	0.9547 95	0.9364 33	SelectKBest with f_regression
XGB Classifier	0.9547 95	0.9448 24	0.9547 95	0.9364 33	RFE with estimator Decision Tree Classifier
XGB Classifier	0.9547 95	0.9448 24	0.9547 95	0.9364 33	RFE with estimator Random Forest Classifier

**Table 23: File Code Level Results Sorted by Precision**

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Hist Gradient Boosting Classifier	0.953 683	0.950 056	0.953 683	0.932 078	RFECV with estimator Random Forest Classifier
Hist Gradient Boosting Classifier	0.954 599	0.949 567	0.954 599	0.934 694	RFE with estimator MLP Classifier
Hist Gradient Boosting Classifier	0.953 749	0.948 622	0.953 749	0.932 355	SparsePCA
Hist Gradient Boosting Classifier	0.954 337	0.947 664	0.954 337	0.934 204	SelectKBest with f_regression
Voting Classifier	0.953 814	0.947 612	0.953 814	0.932 631	RFECV with estimator Random Forest Classifier



**Table 24: File Code Level Results Sorted by Recall**

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
XGB Classifier	0.9547 95	0.9448 24	0.9547 95	0.9364 33	SelectKBest with f_classif
XGB Classifier	0.9547 95	0.9448 24	0.9547 95	0.9364 33	SelectKBest with mutual_info_classif
XGB Classifier	0.9547 95	0.9448 24	0.9547 95	0.9364 33	SelectKBest with f_regression
XGB Classifier	0.9547 95	0.9448 24	0.9547 95	0.9364 33	RFE with estimator Decision Tree Classifier
XGB Classifier	0.9547 95	0.9448 24	0.9547 95	0.9364 33	RFE with estimator Random Forest Classifier

**Table 25: File Code Level Results Sorted by F1 Score**

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Random Forest Classifier	0.9533 56	0.9366 99	0.953 356	0.937 493	RFECV with estimator SVC
Random Forest Classifier	0.9529 63	0.9356 17	0.952 963	0.937 335	RFE estimator SVC
Random Forest Classifier	0.9531 6	0.9360 26	0.953 16	0.937 184	SelectKBest with f_regression
Random Forest Classifier	0.9531 6	0.9359 81	0.953 16	0.937 091	RFE with estimator Random Forest Classifier
Random Forest Classifier	0.9529 63	0.9353 44	0.952 963	0.936 874	RFE with estimator Decision Tree Classifier

#### 4.5 BENCHMARKING AUTOML EXPERIMENT RESULTS

The top results of the benchmarking autoML experiment according to sorted by ROC AUC, Precision, Recall, and F1 Score are shown in Table 26, Table 27, Table 28, and Table 29. All the results are listed in Appendix 8.

**Table 26: Results of the OpenML Data Experiment Sorted by ROC AUC**

Model	ROC AUC	Precision	Recall	F1 Score	Feature Selection Model
Passive Aggressive Classifier	0.7364 46	0.464 739	0.749 367	0.573 69	RFECV with estimator SVC
SGD Classifier	0.7235 37	0.524 789	0.629 747	0.572 497	RFE with estimator MLP Classifier
SGD Classifier	0.7177 09	0.490 028	0.653 165	0.559 957	RFECV with estimator Random Forest Classifier
SGD Classifier	0.7154 04	0.386 526	0.876 371	0.536 45	RFECV with estimator Logistic Regression
SGD Classifier	0.7115 69	0.497 151	0.625 949	0.554 165	RFECV estimator Decision Tree Classifier

**Table 27: Results of the OpenML Data Experiment Sorted by Precision**

Model	ROC AUC	Precision	Recall	F1 Score	Feature Selection Model
Passive Aggressive Classifier	0.592519	0.80084	0.201055	0.321417	RFECV with estimator MLP Classifier
Passive Aggressive Classifier	0.638139	0.684023	0.324262	0.43996	SelectKBest with mutual_info_classif
Gaussian NB	0.610679	0.671706	0.262447	0.377427	SelectKBest with f_classif
Gaussian NB	0.610679	0.671706	0.262447	0.377427	SelectKBest with mutual_info_classif
Gaussian NB	0.610679	0.671706	0.262447	0.377427	SelectKBest with f_regression

**Table 28: Results of the OpenML Data Experiment Sorted by Recall**

Model	ROC AUC	Precision	Recall	F1 Score	Feature Selection Model
SGD Classifier	0.715404	0.386526	0.876371	0.53645	RFECV with estimator Logistic Regression
Passive Aggressive Classifier	0.736446	0.464739	0.749367	0.57369	RFECV with estimator SVC
Passive Aggressive Classifier	0.682597	0.414557	0.666878	0.511282	Mini Batch Sparse PCA
SGD Classifier	0.717709	0.490028	0.653165	0.559957	RFECV with estimator Random Forest Classifier
SGD Classifier	0.699235	0.462627	0.634599	0.535136	PCA

**Table 29: Results of the OpenML Data Experiment Sorted by F1 Score**

Model	ROC AUC	Precision	Recall	F1 Score	Feature Selection Model
Passive Aggressive Classifier	0.736446	0.464739	0.749367	0.57369	RFECV with estimator SVC
SGD Classifier	0.723537	0.524789	0.629747	0.572497	RFE with estimator MLP Classifier
SGD Classifier	0.717709	0.490028	0.653165	0.559957	RFECV with estimator Random Forest Classifier
MLP Classifier	0.704278	0.610123	0.513713	0.557783	RFE with estimator Random Forest Classifier
SGD Classifier	0.711569	0.497151	0.625949	0.554165	RFECV with estimator Decision Tree Classifier

#### **4.6 SURVEY RESULTS OF THE USABILITY OF PPDAAS**

The platform is tested by users to get insights about its applicability and usage in real-world scenarios.

We created a video demonstrating the capabilities of the platform and asked software professionals with machine-learning backgrounds to watch the video and evaluate the system. We wanted to collect comments about the effectiveness and challenges of our platform. Therefore, initially, 10 software professionals were reached to get initial views on the platform. Then, an initial draft of the survey instrument using zero-shot prompting [57] within Baize LLM [57], with version baize-v2-7). Then, the AI-generated survey questions and choices were manually edited for improvement. The survey questions, accompanied by their respective answers, are presented below alongside their corresponding in Table 30. The participation in the survey has been diverse, ranging from academic individuals to those with long-term experience in the industry. Based on the survey, the age of the participants predominantly falls between 25 to 44 years. Specifically, four of them are aged between 25 and 34, while five are between 35 and 44. In terms of sectoral experience, most of the participants have over 5 years of experience, though there is also a notable number with experience ranging from 2 to 4 years. Considering this information, it can be deduced that the demographic primarily consists of mid-aged professionals with significant experience in the sector.

**Table 30:** Results of the Evaluation

<b>Experience In Sector</b>	5+	1-2	2-4	5+	2-4	2-4	5+	2-4	5+	5+	2-4
<b>Age</b>	35-44	25-34	25-34	45-54	25-34	25-34	35-44	25-34	35-44	35-44	25-34
<b>Satisfaction Level</b>	5	4	5	3	4	4	4	5	5	5	5
<b>Understandability</b>	4	4	4	3	3	3	3	3	5	5	4
<b>User-friendliness</b>	5	5	4	3	4	4	3	3	5	4	4
<b>Effectiveness</b>	3	5	5	4	3	4	4	4	5	4	5
<b>Utility</b>	3	4	3	4	4	4	4	5	5	4	5
<b>Recommendation Likelihood</b>	5	5	5	4	4	4	4	5	5	3	4
<b>Future Usage Likelihood</b>	4	4	5	3	4	4	4	5	5	3	4
<b>Enjoyment Level</b>	5	5	3	3	3	3	3	5	5	4	3
<b>Learning Level from Video</b>	4	4	5	2	3	3	2	5	4	5	5
<b>Time-saving Potential</b>	5	5	4	3	4	4	5	4	4	5	5
<b>Quality Improvement Potential</b>	4	4	4	3	4	4	4	4	4	5	5
<b>Cost-saving Potential</b>	5	5	5	4	4	4	5	4	5	5	5

## CHAPTER V DISCUSSION

The results of the experiments which are stated in chapter IV will be discussed below.

### 5.1 DISCUSSION ON OPENML ADULT DATA EXPERIMENT

The highest performance scores according to the evaluation metric have been shown below in Table 31.

**Table 31:** Highest Scores Concerning Score Parameters

Score Parameter	Score Value	Model	Feature Selection Method
ROC AUC	0.736446	Passive Aggressive Classifier	RFECV with estimator SVC
Precision	0.80084	Passive Aggressive Classifier	RFECV with estimator MLP
Recall	0.876371	SGD Classifier	RFECV with estimator Logistic Regression
F1 Score	0.57369	Passive Aggressive Classifier	RFECV with estimator SVC

The Passive Aggressive Classifier achieved the highest ROC\_AUC value, it achieved around 0.736 with the feature selection technique RFECV with an estimator SVC with use as hyperparameter n components equals 4. The result shows that the model has a strong discriminatory capacity in differentiating positive and negative instances. In addition to this SGD Classifier obtained promising results with 0.723 which uses the RFE feature selection technique with an estimator MLP Classifier and hyperparameter tuning with set feature selection count 4.

The Passive Aggressive Classifier shows the highest precision among the classifiers with a value of around 0.80 with the feature selection RFECV with an estimator MLP Classifier and hyperparameter tuning as features to select as four. If we look closer at the top 5 precision values, it can be easily seen that Passive Aggressive Classifier and Gaussian Naïve Bayes have the highest values with different feature selection techniques. The SGD Classifier demonstrates the highest recall value

around 0.876 with the feature selection RFECV estimator Logistic Regression and used hyperparameter tuning technique with features to select as 4. This suggests that the model demonstrated efficacy in identifying a substantial portion of true positive instances, although it may have also generated a higher number of false positives, resulting in a reduced precision score.

The Passive Aggressive Classifier with RFECV with an estimator SVC and tuning as a feature to select equals 4 feature selection method obtained the highest value of F1 score with the value of 0.5736. This shows that the data have better performance metrics on Recall AUC and Precision. Therefore, it shows that the data is imbalanced.

By comparing the results on [14] their results based on 1-AUC and ours are ROC AUC, Precision, Recall, and F1 Score. Our lowest 1-AUC results equal around 0.263. With comparing the [14] results. Our results tend to approach their results. Their provided autoML solution platform obtained 0.1547 and ours was 0.263 shown in Table 32.

**Table 32:** Comparison Table with AutoML vs. PPDaaS

<b>Platform</b>	<b>1-AUC</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>
PPDaaS	0.2635	0.80084	0.876371	0.57369
Feurer and colleagues	0.1547	Not Provided	Not Provided	Not Provided

To sum up, we wanted to examine the performance of classifiers within a specific time to make a fair comparison by limiting the iterations by setting runtime limits. With the analysis results, Although, the Passive Aggressive Classifiers perform better than the others with a feature selection method such as RFECV with different estimators, The study of Thapa S [59] found the Random Forest Classifier to be their top-performing algorithm.

## 5.2 SOFTWARE VULNERABILITY EXPERIMENT

The results of the performance of machine learning algorithms on the software vulnerability data by using PPDaaS (Prediction Platform as a Service) show the efficacy of various classifiers. Each of the classification methods tested with a different

feature selection method as shown in Table 4. The performance evaluation of these models is based on F1 Score, Accuracy, and Precision.

### 5.2.1 Function Code Level

Firstly, for the whole dataset where year range 2002-2019, the Random Forest Classifier gives the best results with feature selection models with SelectKBest, PCA, and Recursive Feature Elimination with Cross-Validation (RFECV). The top accuracy is nearly 95.08 and precision recall and F1 score is nearly close to 0.94.

Between the years 2002 to 2014, the Stacking Classifier and Random Forest Classifier showed commendable performance achieving high results. Accuracy hovered around 0.95 and Precision hovered around 0.939. By that time Bernoulli NB Passive Aggressive Classifier gave the worst performance at around 0.73 and 0.55 respectively.

The best performance for that year, optimizing model performance for this data is the Stacking Classifier with and Sparse PCA Feature Selection Method and the second is the Random Forest Classifier with Mini Batch Sparse PCA.

In the period between 2002 to 2016, the Random Forest Classifier algorithm once again gave the best results. However, this time it was accompanied by different feature selection models, such as Recursive Feature Elimination (RFE) and SelectKBest. The highest model performance is taken with evolution metrics with an accuracy of about 95.71% and the prediction, recall and F1 score is nearly the same around 0.95.

The authors Pereira and colleagues [9] applied several classification methods to the Software Vulnerability dataset. They used several classifiers and hyperparameter settings, they provided their best precision, recall, and F1-score results. They obtained the best Precision score with the Random Forest algorithm using the data balancing technique in addition to this, they obtained the best Recall score on the Bagging algorithm, and with not using the data balancing technique and dimension reduction used variance. Lastly, they took the best F1-score with the Random Forest algorithm without using the Data balancing technique and using dimension reduction as Variance. Their best Precision is 0.9362, their best Recall is 0.9505, and best F1-score is 0.9379.

**Table 33:** Comparison between our results versus Pereira and colleague's results

Platform	Precision	Recall	F1 Score
Pereira and colleagues	0.9362	0.9505	0.9379
PPDaaS	0.951828	0.951068	0.93886

Our outcomes were superior to their highest precision, recall, and F1-score performances. The SGD Classifier model in our analysis had the highest precision with 0.951828. Similarly, the Stacking Classifier outperformed the Bagging No Variance bootstrap model from the last trial, achieving our best recall with a value of 0.951068. Our top-performing model in terms of the F1-score measure was the Random Forest Classifier, which outperformed their RF No Variance bootstrap modal with an F1-score of 0.93886.

We used various feature selection methods to select the best method for the data, in that way we observed best feature selection methods are for the evaluation metrics SelectKBest with mutual\_info\_classif for best F1-Score, Sparse PCA for best Recall, RFECV with an estimator Random Forest Classifier and hyperparameter tuning as number of features set to 28 for best Precision and Sparse PCA for the best accuracy.

We used the model degradation feature (Fig 15) to analyze which model is degraded or improved in a specific period and evaluated them.

Model	Feature Selection File 1	Feature Selection File 2	Metric	File 1 Value	File 2 Value	Difference	Status
RandomForestClassifier	PCA(n_components=28)	PCA(n_components=28)	Accuracy	0.9383355931	0.9475297473	0.009194154199999915	Positive
RandomForestClassifier	RFE(estimator=MLPClassifier(), n_features_to_select=28)	RFE(estimator=MLPClassifier(), n_features_to_select=28)	Accuracy	0.9350577625	0.9474503525	0.012392590000000037	Positive
RandomForestClassifier	SelectKBest(mutual_info_classif, k=28)	SelectKBest(mutual_info_classif, k=28)	Accuracy	0.9357382239	0.9473301687	0.011591944799999992	Positive
RandomForestClassifier	RFE(estimator=SVC(), n_features_to_select=28)	RFE(estimator=SVC(), n_features_to_select=28)	Accuracy	0.935864876	0.9473301687	0.011465226099999963	Positive
StackingClassifier	RFE(estimator=SVC(), n_features_to_select=28)	RFE(estimator=SVC(), n_features_to_select=28)	Accuracy	0.9356931526	0.9471390524	0.011445899799999992	Positive
StackingClassifier	RFECV(estimator=SVC(), min_features_to_select=28)	RFECV(estimator=SVC(), min_features_to_select=28)	Accuracy	0.9363537628	0.9469010989	0.010547336099999916	Positive
StackingClassifier	MiniBatchSparsePCA(n_components=28)	MiniBatchSparsePCA(n_components=28)	Accuracy	0.9381173256	0.9468807932	0.008763467599999908	Positive
RandomForestClassifier	RFECV(estimator=LogisticRegression(), min_features_to_select=28)	RFECV(estimator=LogisticRegression(), min_features_to_select=28)	Accuracy	0.9359925648	0.9468799813	0.010887416499999913	Positive
StackingClassifier	SelectKBest(f_regression, k=28)	SelectKBest(f_regression, k=28)	Accuracy	0.9359744893	0.9468137219	0.01083923259999997	Positive
RandomForestClassifier	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)	Accuracy	0.9347404046	0.9468018279	0.012061423300000004	Positive
RandomForestClassifier	SelectKBest(f_regression, k=28)	SelectKBest(f_regression, k=28)	Accuracy	0.9362250955	0.9466663851	0.010441289600000014	Positive
RandomForestClassifier	RFE(estimator=LogisticRegression(), n_features_to_select=28)	RFE(estimator=LogisticRegression(), n_features_to_select=28)	Accuracy	0.9349913433	0.9466192342	0.011627890000000049	Positive
StackingClassifier	RFE(estimator=MLPClassifier(), n_features_to_select=28)	RFE(estimator=MLPClassifier(), n_features_to_select=28)	Accuracy	0.9362615468	0.9465542113	0.010292664499999993	Positive
StackingClassifier	SparsePCA(n_components=28)	SparsePCA(n_components=28)	Accuracy	0.9384780947	0.9465542113	0.008076116599999996	Positive
RandomForestClassifier	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)	Accuracy	0.9349920969	0.9464880872	0.011495990300000036	Positive
StackingClassifier	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)	Accuracy	0.9363109095	0.9464536977	0.010142788200000003	Positive
StackingClassifier	RFECV(estimator=LogisticRegression(), min_features_to_select=28)	RFECV(estimator=LogisticRegression(), min_features_to_select=28)	Accuracy	0.9368274107	0.9464376284	0.009610217700000034	Positive
StackingClassifier	RFE(estimator=LogisticRegression(), n_features_to_select=28)	RFE(estimator=LogisticRegression(), n_features_to_select=28)	Accuracy	0.936164601	0.9463039991	0.010139390999999974	Positive

**Figure 15:** Model Degradation Tool Output



Table 34 shows that the Passive Aggressive Classifier takes the lead for model improvement with a value of 0.047 between 2002-2014 to 2002-2016 period.

**Table 34:** 2002-2014 to 2002-2016 Model Degradation Top 4 Model Improvement Results

Model	Feature Selection	Metric	2002-2014	2002-2016	Difference	Status
Passive Aggressive Classifier	RFE estimator Random Forest Classifier	Accuracy	0.869909508	0.9173598174	0.0474503094	Negative
Passive Aggressive Classifier	Mini Batch Sparse PCA	Accuracy	0.881254073	0.9192524087	0.0379983357	Negative
Passive Aggressive Classifier	RFECV estimator SVC	Accuracy	0.9011516057	0.9211184253	0.0199668196	Negative
Passive Aggressive Classifier	RFE estimator SVC	Accuracy	0.9076376841	0.924677541	0.0170398569	Negative

**Table 35:** 2002-2014 to 2002-2016 Model Degradation Top 4 Model Deterioration Results

Model	Feature Selection	Metric	2002-2014	2002-2016	Difference	Status
GaussianNB	SelectKBest with f_classif	Accuracy	0.915628	0.241069	-0.67456	Positive
GaussianNB	SelectKBest with mutual_info_classif	Accuracy	0.915628	0.241069	-0.67456	Positive
GaussianNB	SelectKBest with f_regression	Accuracy	0.915628	0.241069	-0.67456	Positive
GaussianNB	RFE with estimator Decision Tree Classifier	Accuracy	0.915628	0.241069	-0.67456	Positive

Passive Aggressive Classifier with RFE using Random Forest Classifier is an estimator and selecting 28 features shows us the Accuracy in 2002-2014 as 0.869909508. On the other hand, Accuracy in 2002-2016 is 0.9173598174. The difference is 0.0474503094 which means the accuracy of the model has shown a

significant improvement of 0.0474503094 between in 2016. This suggests that the RFE feature selection technique, coupled with the Random Forest Classifier as the estimator, has positively improved over time. On the other hand, the GaussianNB model seems to degrade significantly in performance (in terms of accuracy) when trained on data from the period 2002-2014 compared to the period 2002-2016 shown in Table 35.

Passive Aggressive Classifier with SelectKBest feature selection using f\_classif scoring and selecting 28 best features shows us the accuracy in 2002-2016 as 0.8765474036. On the other hand, accuracy in 2002-2019 is 0.917257212. The difference is 0.0407098084 which means the model's performance has improved by 0.0407098084 between 2002 and 2019, indicating the added data improved the model prediction capability. Table 36 shows the top four model improvement results compared with the data from 2002-2016 to 2002-2019. On the other hand, The GaussianNB model, when evaluated using different feature selection techniques, shows a consistent degradation in performance when transitioning from data of the period 2002-2016 to 2002-2019 shown in Table 37.

**Table 36:** 2002-2016 to 2002-2019 Model Degradation Top 4 Model Improvement Results

Model	Feature Selection	Metric	2002-2016	2002-2019	Difference	Status
Passive Aggressive Classifier	SelectKBest with f_classif	Accuracy	0.8765474036	0.917257212	0.0407098084	Negative
SGD Classifier	RFECV with estimator Random Forest Classifier	Accuracy	0.9285697598	0.9518282808	0.023258521	Negative
SGD Classifier	RFECV with estimator Decision Tree Classifier	Accuracy	0.9061690468	0.9179257369	0.0117566901	Negative

Ada Boost Classifier	Mini Batch Sparse PCA	Accuracy	0.9218097851	0.926350702	0.0045409169	Negative
----------------------	-----------------------	----------	--------------	-------------	--------------	----------

**Table 37:** 2002-2016 to 2002-2019 Model Degradation Top 4 Model Deterioration Results

Model	Feature Selection	Metric	2002-2016	2002-2019	Difference	Status
GaussianNB	SelectKBest with f_classif	Accuracy	0.92492	0.251167	-0.67375	Positive
GaussianNB	SelectKBest with mutual_info_classif	Accuracy	0.92492	0.251167	-0.67375	Positive
GaussianNB	SelectKBest with f_regression	Accuracy	0.92492	0.251167	-0.67375	Positive
GaussianNB	RFE with estimator Decision Tree Classifier	Accuracy	0.92492	0.251167	-0.67375	Positive

In summary, the results from the analyzed models and feature selection techniques in the years 2002-2014 to 2002-2016 and 2002-2016 to 2002-2019 periods indicate a generally positive trend in model performance over time. These findings are valuable in the context of model maintenance and deployment, as they highlight the importance of choosing appropriate feature selection methods to ensure model robustness over extended periods. On the other hand, the GaussianNB model exhibits a notable decrease in accuracy when expanding the data timeframe, irrespective of the feature selection method used.

### 5.2.2 Class Code Level

The most significant observation is the consistency of Ridge Classifier CV around all metrics at class code level granularity. We obtained the highest accuracy, precision, recall, and F1 score from it. Moreover, these evaluation metrics are nearly close to each other around 94%. This shows us the class-level dataset tends to be consistent with Ridge Classifier CV and this classifier is robust and reliable for this dataset. In addition to these, we obtained a Random Forest Classifier and Bagging Classifier gave promising results. Although Ridge Classifier CV, on the other hand, performs extraordinarily well in terms of accuracy, precision, recall, and F1 score, the

Random Forest Classifier had the greatest precision score, which was around 93.69%. With the comparison of best case to worst case, SGDClassifier shows the worst results in every metric. This shows that Ridge Classifier CV was capable of accurately detecting vulnerabilities and SGDClassifier was the worst. The dataset used is gathered from the study [28] for this data granularity. Pereira and colleagues [28] did not perform an analysis of class-level vulnerabilities; however, we performed an analysis of this gap.

### 5.2.3 File Code Level

The top 5 results of the file-level vulnerability dataset results show that feature selection methods including SelectKBest with `f_classif` and `mutual_info_classif` and RFE with Decision Tree Classifier and Random Forest Classifier have no significant effect on accuracy, as the highest accuracy is approximately 95.48%. In addition to that, precision differs from it. Hist Gradient Boosting Classifier achieved the highest precision of approximately 95%. However, precision differs with different feature selection methods, the highest score was achieved with RFECV with estimator Random Forest Classifier. By looking at the Recall, the same results are seen with Accuracy. Lastly, the highest F1 Score is achieved by the Random Forest Classifier, and the Top 5 is the same with the highest result. The top 5 results differ because of the feature selection methods. The best result of the F1 Score was achieved with the feature selection method the RFECV with the estimator SVC. Pereira and colleagues [28] did not perform an analysis of file-level vulnerabilities; however, we performed an analysis of this gap.

Overall, the analysis indicates that the models achieved high accuracy and balanced performance in terms of Precision, Recall, and F1 Score. Different machine learning algorithms were employed with varying feature selection methods, showcasing the potential of these models in detecting vulnerabilities in the software dataset.

In conclusion, the optimal choice of machine learning algorithms and feature selection techniques can significantly vary based on the granularity level and the performance metric in focus as shown in Table 38.

**Table 38:** Best ML Algorithm and Feature Selection according to Metrics in all Granularity

Metric	Function		Class		File	
	Algorithm	Feature Selection Technique	Algorithm	Feature Selection Technique	Algorithm	Feature Selection Technique
Accuracy	Stacking Classifier	Sparse PCA	Ridge Classifier CV	SelectKBest with f_classif	XGB Classifier	SelectKBest with f_classif
Precision	SGD Classifier	RFECV with estimator Random Forest Classifier	Random Forest Classifier	RFECV with estimator SVC	Hist Gradient Boosting Classifier	RFECV with estimator Random Forest Classifier
Recall	Stacking Classifier	Sparse PCA	Ridge Classifier CV	SelectKBest with f_classif	XGB Classifier	SelectKBest with f_classif
F1 Score	Random Forest Classifier	SelectKBest with mutual_info_classif	Random Forest Classifier	RFECV with estimator SVC	Random Forest Classifier	RFECV with estimator SVC

#### 5.2.4 Usability Analysis

The PPDaaS software platform received predominantly positive feedback from the survey respondents. Most users, across varied experience levels and age groups, were satisfied with its capabilities and found it user-friendly and effective for automated machine learning. The survey questions and answers with an average score are shown in Table 39 which indicates a generally positive perception of the developed platform PPDaaS especially, among those in the age group of 25-44 years with significant experience in the sector. Most participants believe the platform is effective and the average score is 4.3; user-friendly with an average score of 3.8, and offers considerable utility. On average, the platform's cost-saving potential stood out the most with a rating of 4.3. The overall satisfaction level and time-saving potential also received high averages, ranging from 4.0 to 4.5. These averages show that feedback from the user is generally positive.

**Table 39: Survey Results with Average Score**

<b>Questions</b>	<b>Average of the Answers</b>
Overall, how satisfied are you with the PPDaaS software platform? (1=not at all satisfied, 5=very satisfied)	4.4
How easy was it to understand how to use the PPDaaS software platform? (1=very difficult, 5=very easy)	3.8
How user-friendly is the PPDaaS software platform? (1=not at all user-friendly, 5=very user-friendly)	3.8
How effective is the PPDaaS software platform at helping you build an automated machine learning model? (1=not effective at all, 5=very effective)	4.3
How useful is the PPDaaS software platform in terms of its capabilities and features? (1=not at all useful, 5=very useful)	4.2
How likely are you to recommend the PPDaaS software platform to a colleague or friend? (1=not at all likely, 5=very likely)	4.3
How likely are you to use the PPDaaS software platform in the future for building automated machine learning models? (1=not at all likely, 5=very likely)	4.1
How much did you enjoy using the PPDaaS software platform? (1=not at all enjoyable, 5=very enjoyable)	3.8
How much did you learn about automated machine learning through this video on the PPDaaS software platform? (1=not at all, 5=very much)	3.9
How much will the PPDaaS software platform help you save time and effort in building automated machine learning models? (1=not at all, 5=very much)	4.3
How much will the PPDaaS software platform help you improve the quality of your automated machine learning models? (1=not at all, 5=very much)	4.1
How much will the PPDaaS software platform help you reduce the cost of building automated machine learning models? (1=not at all, 5=very much)	4.7

## CHAPTER VI

### CONCLUSION

In conclusion, in this thesis, an AutoML platform named PPDaaS is developed that incorporates runtime limit and model degradation features. Initially, we explored and highlighted the absence of features in the available platforms. PPDaaS enables users to apply and use feature selection techniques and machine learning models without writing any code automatically. Users can compare their model performance results around 45 Machine Learning Algorithms (Table 2, Table 3), each run with different feature selection methods (Table 4). Because software vulnerability has a critical role in the software industry [1], we use the software vulnerability data created by [28] to build prediction models using PPDaaS. In addition to this, we conducted tests on the OpenML adult dataset to make comparisons with other automatic machine-learning tools such as Weka[15], and H2O.ai[16].

Firstly, we analyzed the performance of all granularity levels of the vulnerability dataset which are functions, classes, and files. Our study showed that the levels of granularity of the vulnerability dataset performed well nearly the same in predicting the vulnerability. Especially, class-based level has the highest performance in classifying the vulnerability based on the F1 Score. In addition to that, machine learning algorithms also showed differences in performance for each granularity.

Furthermore, an experiment is conducted on OpenML adult data to make a benchmark analysis and compare it with the other AutoML tools. The results show that the PPDaaS platform evaluation value as 1-ROC is very close to the platform that was created by [14]. However, we evaluate the model performance with different metrics such as Recall, Precision, and F1-Score. The model has the best performance in the SDGClassifier and Passive Aggressive Classifier.

Afterward, because the vulnerability data may change over time, we simulated model degradation analysis on the function level data by splitting it into three chunks (2002 to 2014, 2002 to 201, and 2002 to 2019) and developed models for those datasets. The comprehensive analysis of the model's performance for the periods

2002-2014 to 2002-2016 and 2002-2016 to 2002-2019 reveal a consistent trend of improved in model performance over time. Our results revealed that Aggressive Classifiers have a positive trend in model performance evaluation over time. On the other hand, the GaussianNB model demonstrates a comparable and significant degradation in performance when transitioning from the earlier dataset to the later one. This consistency in performance drop is observed irrespective of the feature selection method applied. The observed positive outcomes have a vital role in selecting suitable feature selection methods and machine learning algorithms to protect resilience and accuracy over an extended duration.

Overall, we have successfully addressed the research questions by providing new solutions for the "Runtime Limit" and Model Degradation features for the PPDaaS platform that fills the gap in the AutoML literature. When applied to software vulnerability data, our experiment results outperform Pereira and colleagues [28] results for software vulnerability prediction. Furthermore, our data indicates that software vulnerability prediction is optimal at the class granularity level. Additionally, the model performance has displayed a generally positive trend over the specific periods examined.

Lastly, to get feedback about the capabilities of this platform we conducted a study to gather users' views on the platform. The platform is evaluated by out of 10 software professionals. Overall, their answers highlight that our platform can find a place in the software industry because of its capabilities ease of use, and applicability to real-world scenarios.

## **6.1 FUTURE WORK**

In the next phase of this research, the AutoML prediction results can be visually enhanced by introducing complex visualization techniques. Not only will these visual aids help specialists in the field to quickly understand the results, but they can also make the research more comprehensible for non-experts. Additionally, automatic interpretations of the machine learning results could be added to the tool.



## REFERENCES

- [1] Jordan M. I. and Mitchell T. M. (2015), “Machine learning: Trends, perspectives, and prospects”, *Science*, vol. 349, no. 6245, pp. 253–255, doi: 10.1126/science.aac4520.
- [2] Gong Z., Zhong P., and Hu W. (2019), “Diversity in Machine Learning”, *IEEE Access*, vol. 7, pp. 64323–64350, doi: 10.1109/ACCESS.2019.2917620.
- [3] Lee Y. W., Choi J. W., and Shin E. H. (2021), “Machine learning model for predicting malaria using clinical information”, *Comput Biol Med*, vol. 129, pp. 1-7, doi: 10.1016/j.compbimed.2020.104151.
- [4] Arfat Y., Mittone G., Esposito R., Cantalupo B., Deferrari G. M., and Aldinucci M. (2022), “Machine learning for cardiology”, *Minerva Cardiology and Angiology*, vol. 70, no. 1, pp. 75–91, doi: 10.23736/S2724-5683.21.05709-4.
- [5] Greener J. G., Kandathil S. M., Moffat L., and Jones D. T. (2022), “A guide to machine learning for biologists”, *Nature Reviews Molecular Cell Biology*, vol. 23, no. 1, pp. 40–55, doi: 10.1038/s41580-021-00407-0.
- [6] Goodell J. W., Kumar S., Lim W. M., and Pattnaik D. (2021), “Artificial intelligence and machine learning in finance: Identifying foundations, themes, and research clusters from bibliometric analysis”, *Journal of Behavioral and Experimental Finance*, vol. 32, pp. 1-19, doi: 10.1016/j.jbef.2021.100577.
- [7] Rundo F., Trenta F., di Stallo A. L., and Battiato S. (2019), “Machine learning for quantitative finance applications: A survey”, *Applied Sciences (Switzerland)*, vol. 9, no. 24, pp. 1-20, doi: 10.3390/app9245574.
- [8] Gogas P. and Papadimitriou T. (2021), “Machine Learning in Economics and Finance”, *Computational Economics*, vol. 57, no. 1, pp. 1-4, doi: 10.1007/s10614-021-10094-w.

- [9] Anute N., Paliwal M., Patel M., and Kandale N. (2021), “Impact of artificial intelligence and machine learning on business operations”, *Journal of Management Research and Analysis*, vol. 8, no. 2, pp. 69–74, doi: 10.18231/j.jmra.2021.015.
- [10] Sarker I. H. (2021), “Machine Learning: Algorithms, Real-World Applications and Research Directions”, *SN Computer Science*, vol. 2, no. 3, pp. 1-21, doi: 10.1007/s42979-021-00592-x.
- [11] Lebens M., Finnegan R., Sorsen S., and Shah J. (2021), “Rise of the citizen developer”, *Muma Business Review*, vol. 5, no. 12, pp. 101-111, doi:10.28945/4885.
- [12] Hanussek M., Blohm M., and Kintz M. (2021), “Can AutoML outperform humans? An evaluation on popular OpenML datasets using AutoML Benchmark”, *In 2020 2nd International Conference on Artificial Intelligence, Robotics and Control*, pp. 29–32, New York, USA, doi: 10.1145/3448326.3448353.
- [13] Musigmann M., Akkurt B.H., Krähling H., Nacul N.G., Remonda L., Sartoretti T., Henssen D., Brokinkel B., Stummer W., Heindel W., and Mannil M. (2022), “Testing the applicability and performance of Auto ML for potential applications in diagnostic neuroradiology”, *Sci Rep*, vol. 12, no. 1, pp. 1-21, doi: 10.1038/s41598-022-18028-8.
- [14] Feurer M., Eggenberger K., Falkner S., Lindauer M., and Hutter F. (2022), “Auto-Sklearn 2.0: Hands-free AutoML via Meta-Learning”, *Journal of Machine Learning Research*, vol. 23, pp. 1-61, doi: 10.48550/arXiv.2007.04074.
- [15] Witten I. H., Frank E., Trigg L. E., Hall M. A., Holmes G., and Cunningham S. J. (1999), “Weka: Practical machine learning tools and techniques with Java implementations”, *Computer Science Working Papers*, pp. 1-5.
- [16] Cook D. (2016), *Practical Machine Learning with H2O*, 1st Edition, O'Reilly Media Inc., Sebastopol.
- [17] Krishnan S., Franklin M. J., Goldberg K., and Wu E. (2017), “BoostClean: Automated Error Detection and Repair for Machine Learning”, *arXiv*, pp. 1-15, doi: 10.48550/arXiv.1711.01299.

- [18] Krummenacher G., Ong C. S., Koller S., Kobayashi S., and Buhmann J. M. (2018), "Wheel Defect Detection with Machine Learning", *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1176–1187, doi: 10.1109/TITS.2017.2720721.
- [19] Harer J. A., Kim L. Y., Russell R. L., Ozdemir O., Kosta L. R., Rangamani A., Hamilton L. H., Centeno G. I., Key J. R., Ellingwood P. M., McConley M. W., Opper J. M., Chin P., and Lazovich T. (2018). "Automated software vulnerability detection with machine learning", *arXiv*, pp. 1-8, doi: 10.48550/arXiv.1803.04497.
- [20] Medeiros N., Ivaki N., Costa P., and Vieira M. (2020), "Vulnerable Code Detection Using Software Metrics and Machine Learning", *IEEE Access*, vol. 8, pp. 219174–219198, doi: 10.1109/ACCESS.2020.3041181.
- [21] Eberendu A. C., Udegbe V. I., Ezennorom E. O., Ibegbulam A. C., and Chinebu T. I. (2022), "A Systematic Literature Review of Software Vulnerability Detection", *European Journal of Computer Science and Information Technology*, vol. 10, no. 1, pp. 23-37.
- [22] Shar L. K., Briand L. C., and Tan H. B. K. (2015), "Web Application Vulnerability Prediction Using Hybrid Program Analysis and Machine Learning", *IEEE Trans Dependable Secure Comput*, vol. 12, no. 6, pp. 688–707, doi: 10.1109/TDSC.2014.2373377.
- [23] Hydera I., Sultan A. B. M., Zulzalil H., and Admodisastro N. (2015), "Current state of research on cross-site scripting (XSS) - A systematic literature review", *Information and Software Technology*, vol. 58, pp. 170–186, doi: 10.1016/j.infsof.2014.07.010.
- [24] Zeng P., Lin G., Pan L., Tai Y., and Zhang J. (2020), "Software vulnerability analysis and discovery using deep learning techniques: A survey", *IEEE Access*, vol. 8, pp. 197158–197172, doi: 10.1109/ACCESS.2020.3034766.
- [25] Shah I. A., Rajper S., and ZamanJhanjhi N. (2021), "Using ML and Data-Mining Techniques in Automatic Vulnerability Software Discovery", *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 10, no. 3, pp. 2109–2126, doi: 10.30534/ijatcse/2021/871032021.

- [26] Dowd M., McDonald J., and Schuh J. (2006), *The Art of Software Security Assessment-Identifying and Preventing Software Vulnerabilities*, 1st Edition, O'Reilly Media Inc., Ann Arbor.
- [27] Bagnato A. (2009), "Security in Model-Driven Architecture", *CTIT Workshop Proceedings Series*, pp. 5-42, Enschede, Netherlands.
- [28] Pereira J. D., Antunes J. H., and Vieira M. (2022), "A Software Vulnerability Dataset of Large Open Source C/C++ Projects", *Proceedings of IEEE Pacific Rim International Symposium on Dependable Computing*, pp. 152–163, Beijing, China, doi: 10.1109/PRDC55274.2022.00029.
- [29] Sultana K. Z., Anu V., and Chong T. Y. (2021), "Using software metrics for predicting vulnerable classes and methods in Java projects: A machine learning approach", *Journal of Software: Evolution and Process*, vol. 33, no. 3, pp. 1-20, doi: 10.1002/smr.2303.
- [30] Vanschoren J., van Rijn J. N., Bischl B., and Torgo L. (2013), "OpenML: networked science in machine learning", *SIGKDD Explor. Newsl.*, vol. 15, no. 2, pp. 49–60, doi: 10.1145/2641190.2641198.
- [31] Pereira J. D. A. (2020), "Techniques and Tools for Advanced Software Vulnerability Detection", *Proceedings - 2020 IEEE 31st International Symposium on Software Reliability Engineering Workshops, ISSREW 2020*, pp. 123–126, Coimbra, Portugal, doi: 10.1109/ISSREW51248.2020.00049.
- [32] Pereira J. D. A., Lourenço N., and Vieira M. (2023), "On the Use of Deep Graph CNN to Detect Vulnerable C Functions", *ACM International Conference Proceeding Series*, pp. 45–50, New York, USA, doi: 10.1145/3569902.3569913.
- [33] Elkhail A. A. and Cerny T. (2019), "On Relating Code Smells to Security Vulnerabilities", *2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity)*, pp. 7–12, Washington, USA, doi: 10.1109/BigDataSecurity-HPSC-IDS.2019.00013.
- [34] Alves H., Fonseca B., and Antunes N. (2016), "Software Metrics and Security Vulnerabilities: Dataset and Exploratory Study", *Proceedings - 2016 12th European Dependable Computing Conference, EDCC 2016*, pp. 37–44, Gothenburg, Sweden, doi: 10.1109/EDCC.2016.34.

- [35] Medeiros N., Ivaki N., Costa P., and Vieira M. (2021), “An Empirical Study on Software Metrics and Machine Learning to Identify Untrustworthy Code”, *Proceedings - 2021 17th European Dependable Computing Conference, EDCC 2021*, pp. 87–94, Munich, Germany, doi: 10.1109/EDCC53658.2021.00020.
- [36] Zhou A., Sultana K. Z., and Samanthula B. K. (2021), “Investigating the Changes in Software Metrics after Vulnerability is Fixed”, *Proceedings - 2021 IEEE International Conference on Big Data, Big Data 2021*, pp. 5658–5663, Orlando, USA, doi: 10.1109/BigData52589.2021.9671334.
- [37] Siavvas M. (2019), *Static Analysis for Facilitating Secure and Reliable Software* (Doctoral Dissertation), Imperial College London, London.
- [38] Kaya A., Keceli A. S., Catal C., and Tekinerdogan B. (2019). “The impact of feature types, classifiers, and data balancing techniques on software vulnerability prediction models”, *J. Softw. Evol. Process*, pp. 1–25, doi: 10.1002/smr.2164.
- [39] Sultana K.Z., Boyd C.B., and Williams B.J. (2023), "A Software Vulnerability Prediction Model Using Traceable Code Patterns and Software Metrics", *SN COMPUT. SCI.*, vol. 4, pp. 1-23, doi: 10.1007/s42979-023-02077-5.
- [40] Gupta A., Suri B., Kumar V., and Jain P. (2021), “Extracting rules for vulnerabilities detection with static metrics using machine learning”, *International Journal of System Assurance Engineering and Management*, vol. 12, no. 1, pp. 65–76, doi: 10.1007/s13198-020-01036-0.
- [41] S. Lipp, S. Banescu, and A. Pretschner (2022), “An Empirical Study on the Effectiveness of Static C Code Analyzers for Vulnerability Detection”, *ISSTA 22*, vol. 12, pp. 1-12, doi: 10.1145/3533767.
- [42] Ban X., Liu S., Chen C., and Chua C. (2019), “A performance evaluation of deep-learnt features for software vulnerability detection”, *Concurrency and Computation: Practice and Experience*, vol. 31, no. 19, pp. 1-10, doi: 10.1002/cpe.5103.
- [43] Shukla A., Katt B., and Nweke L.O. (2019), "Vulnerability Discovery Modelling With Vulnerability Severity", *2019 IEEE Conference on Information and Communication Technology*, pp. 1-6, Allahabad, India, doi: 10.1109/CICT48419.2019.9066187.

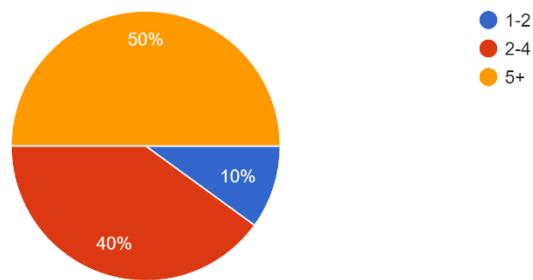
- [44] Bassi D. and Singh H. (2023), "The Effect of Dual Hyperparameter Optimization on Software Vulnerability Prediction Models", *E-Informatica Software Engineering Journal*, vol. 17, no. 1, pp. 230102, doi: 10.37190/e-Inf230102.
- [45] Pereira J.D., Antunes J.H., and Vieira M. (2021), "On Building a Vulnerability Dataset with Static Information from the Source Code", *2021 10th Latin-American Symposium on Dependable Computing (LADC)*, pp. 1-2, Florianópolis, Brazil, doi: 10.1109/LADC53747.2021.9672589.
- [46] Ugrenovic D., Vankeirsbilck J., Pissoort D., Holvoet T., and Boydens J. (2020), "Designing Out-of-distribution Data Detection using Anomaly Detectors: Single Model vs. Ensemble", *2020 XXIX International Scientific Conference Electronics (ET)*, pp. 1-4, Sozopol, Bulgaria, doi: 10.1109/ET50336.2020.9238227.
- [47] Ugrenovic D. et al. (2021), "Towards classification trustworthiness: one-class classifier ensemble", *2021 XXX International Scientific Conference Electronics (ET)*, pp. 1-6, Sozopol, Bulgaria, doi: 10.1109/ET52713.2021.9580016.
- [48] Alves H., Fonseca B., and Antunes N. (2016), "Software Metrics and Security Vulnerabilities: Dataset and Exploratory Study", *Proceedings - 2016 12th European Dependable Computing Conference, EDCC 2016*, pp. 37–44, Gothenburg, Sweden, doi: 10.1109/EDCC.2016.34.
- [49] Alves H., Fonseca B., and Antunes N. (2016), "Experimenting machine learning techniques to predict vulnerabilities", *Proceedings - 7th Latin-American Symposium on Dependable Computing, LADC 2016*, pp. 151–156, Cali, Colombia, doi: 10.1109/LADC.2016.32.
- [50] Bhandari G., Naseer A., and Moonen L. (2021), "CVEfixes: Automated collection of vulnerabilities and their fixes from open-source software", *PROMISE 2021 - Proceedings of the 17th International Conference on Predictive Models and Data Analytics in Software Engineering, co-located with ESEC/FSE 2021*, pp. 30–39, Athens, Greece, doi: 10.1145/3475960.3475985.
- [51] Clemente C. J., Jaafar F., and Malik Y. (2018), "Is predicting software security bugs using deep learning better than the traditional machine learning algorithms?", *Proceedings - 2018 IEEE 18th International Conference on Software Quality, Reliability, and Security, QRS 2018*, pp. 95–102, Lisbon, Portugal, doi: 10.1109/QRS.2018.00023.

- [52] Salimi S., Ebrahimzadeh M., and Kharrazi M. (2020), "Improving real-world vulnerability characterization with vulnerable slices", *PROMISE 2020 - Proceedings of the 16th ACM International Conference on Predictive Models and Data Analytics in Software Engineering, Co-located with ESEC/FSE 2020*, pp. 11–20, New York, USA, doi: 10.1145/3416508.3417120.
- [53] Siavvas M., Kehagias Di., and Tzovaras Di. (2018), "A Preliminary Study on the Relationship among Software Metrics and Specific Vulnerability Types", *Proceedings - 2017 International Conference on Computational Science and Computational Intelligence, CSCI 2017*, pp. 916–921, Las Vegas, USA, doi: 10.1109/CSCI.2017.159.
- [54] Campos J.R., Vieira M., and Costa E. (2019), "Propheticus: Machine Learning Framework for the Development of Predictive Models for Reliable and Secure Software", *2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE)*, pp. 173-182, Berlin, Germany, doi: 10.1109/ISSRE.2019.00026.
- [55] Kaya A., Keceli A.S., Catal C., and Tekinerdogan B. (2019), "The impact of feature types, classifiers, and data balancing techniques on software vulnerability prediction models", *Journal of Software: Evolution and Process*, vol. 31, no. 9, pp. 1-25, doi: 10.1002/smr.2164.
- [56] Brown T., Mann B., Ryder N., Subbiah M., Kaplan J.D., Dhariwal P., Neelakantan A., Shyam P., Sastry G., Askell A., Agarwal S., Herbert-Voss A., Krueger G., Henighan T., Child R., Ramesh A., Ziegler D., Wu J., Winter C., Hesse C., Chen M., Sigler E., Litwin M., Gray S., Chess B., Clark J., Berner C., McCandlish S., Radford A., Sutskever I., and Amodei D. (2020), "Language Models are Few-Shot Learners", *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877-1901.
- [57] Xu C., Guo D., Duan N., and McAuley J. (2023), "Baize: An Open-Source Chat Model with Parameter-Efficient Tuning on Self-Chat Data", *arXiv*, pp. 1-10, doi:10.48550/arXiv:2304.01196.
- [58] huggingface (2023), *Chat with Baize - a hugging face space by Project-Baize*, <https://huggingface.co/spaces/project-baize/chat-with-baize>, DoA. 24.8.2023.
- [59] Thapa S. (2023), "Adult Income Prediction Using various ML Algorithms", *SSRN*, vol. 1, no. 1, pp. 1-4, doi: dx.doi.org/10.2139/ssrn.4325813.

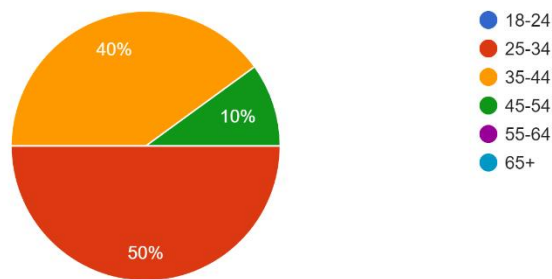
## APPENDICES

### APPENDIX 1: EVALUATION SURVEY RESULTS

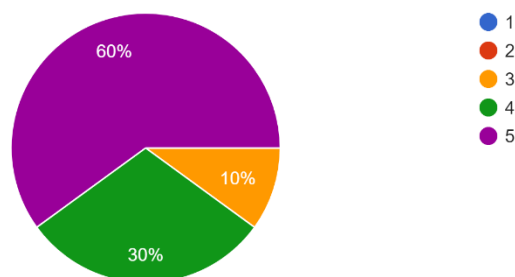
Can you please provide your years of experience in the industry?  
10 yanit



How old are you?  
10 yanit



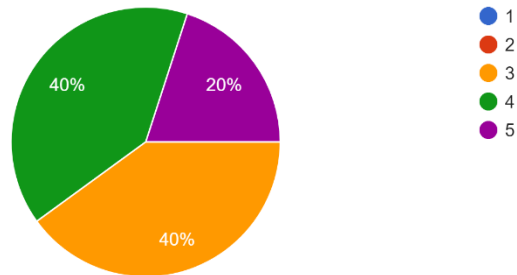
Overall, how satisfied are you with the PPDaaS software platform? (1=not at all satisfied, 5=very satisfied)  
10 yanit





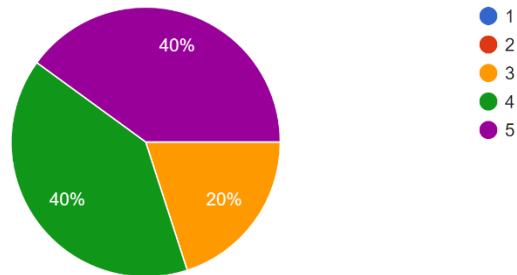
How easy was it to understand how to use the PPDaaS software platform? (1=very difficult, 5=very easy)

10 yanıt



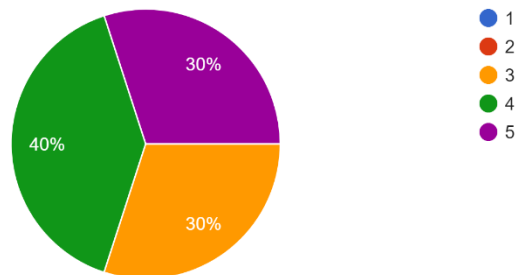
How effective is the PPDaaS software platform at helping you build an automated machine learning model? (1=not effective at all, 5=very effective)

10 yanıt



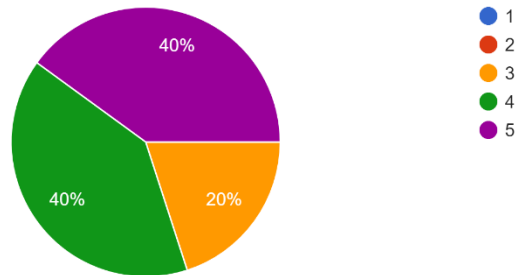
How user-friendly is the PPDaaS software platform? (1=not at all user-friendly, 5=very user-friendly)

10 yanıt



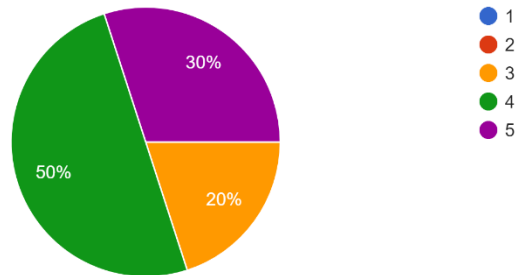
How effective is the PPDaaS software platform at helping you build an automated machine learning model? (1=not effective at all, 5=very effective)

10 yanıt



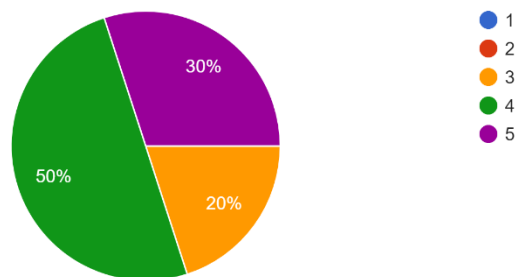
How useful is the PPDaaS software platform in terms of its capabilities and features? (1=not at all useful, 5=very useful)

10 yanıt



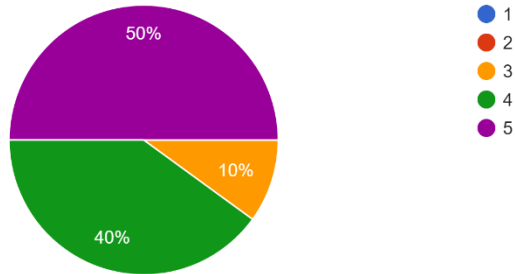
How useful is the PPDaaS software platform in terms of its capabilities and features? (1=not at all useful, 5=very useful)

10 yanıt



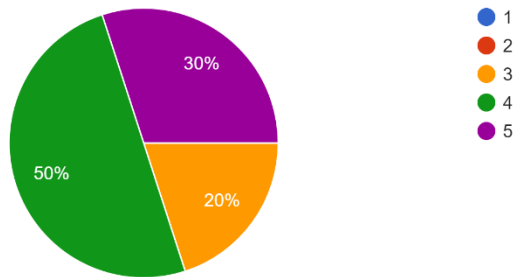
How likely are you to recommend the PPDaaS software platform to a colleague or friend? (1=not at all likely, 5=very likely)

10 yant



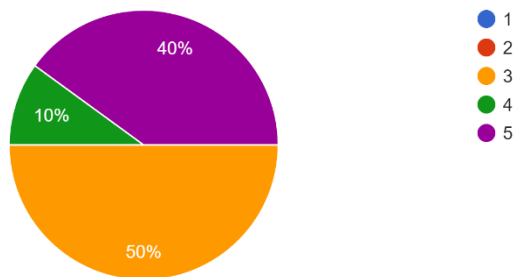
How likely are you to use the PPDaaS software platform in the future for building automated machine learning models? (1=not at all likely, 5=very likely)

10 yant



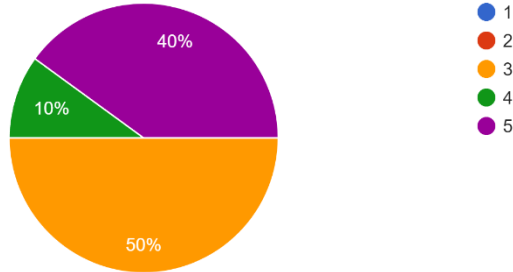
How much did you enjoy using the PPDaaS software platform? (1=not at all enjoyable, 5=very enjoyable)

10 yant



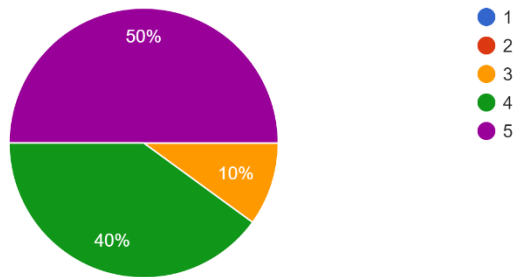
How much did you enjoy using the PPDaaS software platform? (1=not at all enjoyable, 5=very enjoyable)

10 yanıt



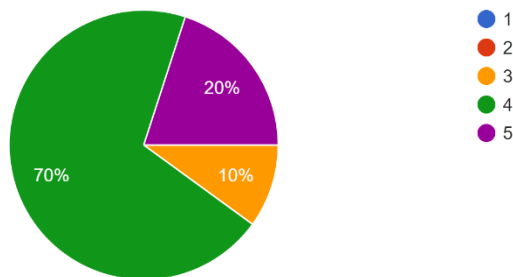
How much will the PPDaaS software platform help you save time and effort in building automated machine learning models? (1=not at all, 5=very much)

10 yanıt



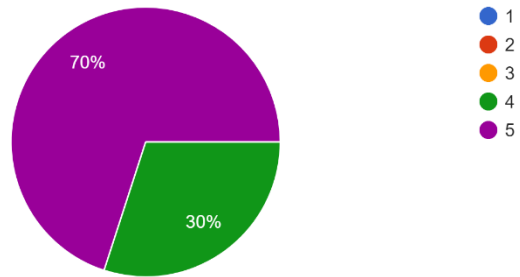
How much will the PPDaaS software platform help you improve the quality of your automated machine learning models? (1=not at all, 5=very much)

10 yanıt



How much will the PPDaaS software platform help you reduce the cost of building automated machine learning models? (1=not at all, 5=very much)

10 yanıt



## **APPENDIX 2: VULNERABILITY EXPERIMENT FEATURES AND DROP COLUMNS**

### **DROP COLUMNS OF THE VULNERABILITY EXPERIMENT**

Description, COMMIT\_HASH, COMMIT\_DATE, COMMIT\_YEAR, VULNERABILITY\_CVE, VULNERABILITY\_YEAR, VULNERABILITY\_CWE, VULNERABILITY\_CATEGORY, ID\_Function, P\_ID, Complement, BeginLine, EndLine, NameMethod, ID\_Class, ID\_File, FilePath, Patched, Occurrence, R\_ID, Affected, binary\_label, grouped\_multiclass\_label, and Visibility.

### **ALL FEATURES OF THE VULNERABILITY EXPERIMENT**

Description, COMMIT\_HASH, COMMIT\_DATE, COMMIT\_YEAR, VULNERABILITY\_CVE, VULNERABILITY\_YEAR, VULNERABILITY\_CWE, VULNERABILITY\_CATEGORY, ID\_Function, P\_ID, Visibility, Complement, BeginLine, EndLine, NameMethod, ID\_Class, ID\_File, FilePath, Patched, Occurrence, R\_ID, Affected, AltCountLineBlank, AltCountLineCode, AltCountLineComment, CountInput, CountLine, CountLineBlank, CountLineCode, CountLineCodeDecl, CountLineCodeExe, CountLineComment, CountLineInactive, CountLinePreprocessor, CountOutput, CountPath, CountSemicolon, CountStmt, CountStmtDecl, CountStmtEmpty, CountStmtExe, Cyclomatic, CyclomaticModified, CyclomaticStrict, Essential, Knots, MaxEssentialKnots, MaxNesting, MinEssentialKnots, RatioCommentToCode, ELIGIBLE\_FOR\_ALERTS, COMMIT\_HAS\_ALERTS, TOTAL\_ALERTS, Flawfinder\_chown, Flawfinder\_gets, Flawfinder\_printf, Flawfinder\_strcpy, Flawfinder\_access, Flawfinder\_sscanf, Flawfinder\_syslog, Flawfinder\_crypt, Flawfinder\_strcat, Flawfinder\_fprintf, Flawfinder\_execl, Flawfinder\_vfprintf, Flawfinder\_sprintf, Flawfinder\_execv, Flawfinder\_ShellExecute, Flawfinder\_\_sntprintf, Flawfinder\_\_tcscopy, Flawfinder\_\_tscat, Flawfinder\_WinExec, Flawfinder\_random, Flawfinder\_getenv, Flawfinder\_chroot, Flawfinder\_srand, Flawfinder\_LoadLibrary, Flawfinder\_EnterCriticalSection, Flawfinder\_InitializeCriticalSection, Flawfinder\_getopt, Flawfinder\_char, Flawfinder\_memcpy, Flawfinder\_atoi, Flawfinder\_atol, Flawfinder\_open, Flawfinder\_fopen, Flawfinder\_tmpfile, Flawfinder\_wscopy, Flawfinder\_TCHAR, Flawfinder\_strlen, Flawfinder\_read, Flawfinder\_strncat, Flawfinder\_mismatch, Flawfinder\_strncpy, Flawfinder\_umask,

Flawfinder\_wcslen, Flawfinder\_getc, Flawfinder\_\_tcslen, Flawfinder\_readlink,  
Flawfinder\_chmod, Flawfinder\_vsnprintf, Flawfinder\_snprintf, Flawfinder\_vsprintf,  
Flawfinder\_vsscanf, Flawfinder\_getlogin, Flawfinder\_execvp, Flawfinder\_execlp,  
Flawfinder\_fscanf, Flawfinder\_popen, Flawfinder\_vprintf, Flawfinder\_system,  
Flawfinder\_wprintf, Flawfinder\_scanf, Flawfinder\_realpath, Flawfinder\_srandom,  
Flawfinder\_getopt\_long, Flawfinder\_bcopy, Flawfinder\_mkstemp, Flawfinder\_vfork,  
Flawfinder\_memalign, Flawfinder\_usleep, Flawfinder\_getchar, Flawfinder\_fgetc,  
Flawfinder\_equal, Flawfinder\_vfscanf, Flawfinder\_setstate, Flawfinder\_tmpnam,  
Flawfinder\_wchar\_t, Flawfinder\_ulimit, Flawfinder\_signal,  
Flawfinder\_CopyMemory, Flawfinder\_wcscat, Flawfinder\_\_snprintf,  
Flawfinder\_vswprintf, Flawfinder\_swprintf, Flawfinder\_\_ftprintf,  
Flawfinder\_lstrcpyW, Flawfinder\_crypt\_r, Flawfinder\_lstrcpy, Flawfinder\_lstrcpyA,  
Flawfinder\_fwprintf, Flawfinder\_\_stprintf, Flawfinder\_\_vftprintf,  
Flawfinder\_vfwprintf, Flawfinder\_\_vsntprintf, Flawfinder\_\_vstprintf,  
Flawfinder\_\_tscanf, Flawfinder\_wscanf, Flawfinder\_SetThreadToken,  
Flawfinder\_lstrcatW, Flawfinder\_getwd, Flawfinder\_LoadLibraryEx,  
Flawfinder\_GetTempFileName, Flawfinder\_CreateProcess, Flawfinder\_lrand48,  
Flawfinder\_AddAccessAllowedAce, Flawfinder\_drand48, Flawfinder\_curl\_getenv,  
Flawfinder\_MultiByteToWideChar, Flawfinder\_\_wtoi, Flawfinder\_\_wtoi64,  
Flawfinder\_wcsncpy, Flawfinder\_lstrcpy, Flawfinder\_tcsncpy,  
Flawfinder\_wcsncat, Flawfinder\_execl, Flawfinder\_nrand48, Flawfinder\_lstrcpyA,  
Flawfinder\_getpass, Flawfinder\_fwscanf, Flawfinder\_vwprintf,  
Flawfinder\_CreateProcessAsUser, Flawfinder\_g\_get\_home\_dir,  
Flawfinder\_mktemp, Flawfinder\_lstrcat, Flawfinder\_g\_get\_tmp\_dir,  
Flawfinder\_\_mbcat, Flawfinder\_lstrcatA, Flawfinder\_tmpnam, multiclass\_label,  
binary\_label, grouped\_multiclass\_label

**APPENDIX 3: VULNERABILITY FUNCTION LEVEL DATA RESULTS IN 2002 TO 2014**

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
StackingClassifier	0.950 43573	0.938 47809	0.950 43573	0.939 92294	SparsePCA(n_components=28)
RandomForestClassifier	0.950 25417	0.938 23629	0.950 25417	0.939 87313	MiniBatchSparsePCA(n_components=28)
StackingClassifier	0.950 25417	0.938 11732	0.950 25417	0.939 61711	MiniBatchSparsePCA(n_components=28)
RandomForestClassifier	0.950 43573	0.938 33559	0.950 43573	0.939 57851	PCA(n_components=28)
StackingClassifier	0.950 16339	0.937 93568	0.950 16339	0.939 46380	PCA(n_components=28)
RandomForestClassifier	0.950 07262	0.937 71369	0.950 07262	0.939 22326	SparsePCA(n_components=28)
StackingClassifier	0.949 98184	0.937 49022	0.949 98184	0.938 98125	SelectKBest (mutual_info_classif, k=28)
StackingClassifier	0.949 70951	0.937 10116	0.949 70951	0.938 86884	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
StackingClassifier	0.949 61873	0.936 82741	0.949 61873	0.938 53828	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
BaggingClassifier	0.948 43863	0.935 61565	0.948 43863	0.938 45136	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
RandomForestClassifier	0.948 89252	0.935 99256	0.948 89252	0.938 44350	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
RandomForestClassifier	0.949 16485	0.936 22509	0.949 16485	0.938 38250	SelectKBest (f_regression, k=28)
RandomForestClassifier	0.948 80174	0.935 86487	0.948 80174	0.938 37716	RFE (estimator=SVC(), n_features_to_select=28)
StackingClassifier	0.949 25562	0.936 31091	0.949 25562	0.938 36113	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
RandomForestClassifier	0.948 71096	0.935 73822	0.948 71096	0.938 31086	SelectKBest (mutual_info_classif, k=28)
StackingClassifier	0.949 25562	0.936 26154	0.949 25562	0.938 27298	RFE (estimator=MLPClassifier(), n_features_to_select=28)



StackingClassifier	0.949 34640 5	0.936 35376 3	0.949 34640 5	0.938 25056 6	RFECV (estimator=SVC(), min_features_to_select=28)
RandomForestClassifier	0.948 62018 9	0.935 61259 6	0.948 62018 9	0.938 24460 5	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
BaggingClassifier	0.947 98475	0.935 10033 5	0.947 98475	0.938 20345 9	SelectKBest (f_classif, k=28)
StackingClassifier	0.948 98329 7	0.935 90323 9	0.948 98329 7	0.938 16244 7	SelectKBest (f_classif, k=28)
StackingClassifier	0.949 25562 8	0.936 16460 1	0.949 25562 8	0.938 09531 9	RFE (estimator=LogisticRegression(), n_features_to_select=28)
RandomForestClassifier	0.948 52941 2	0.935 42705 2	0.948 52941 2	0.938 09249 2	RFECV (estimator=SVC(), min_features_to_select=28)
StackingClassifier	0.949 07407 4	0.935 93448 1	0.949 07407 4	0.938 05213 2	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
StackingClassifier	0.949 16485 1	0.935 97448 9	0.949 16485 1	0.937 93979 9	SelectKBest (f_regression, k=28)
RandomForestClassifier	0.948 34785 8	0.935 11588 1	0.948 34785 8	0.937 87412 4	SelectKBest (f_classif, k=28)
RandomForestClassifier	0.948 25708 1	0.934 99209 7	0.948 25708 1	0.937 80817 1	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
StackingClassifier	0.948 98329 7	0.935 69315 3	0.948 98329 7	0.937 80791 8	RFE (estimator=SVC(), n_features_to_select=28)
BaggingClassifier	0.948 16630 4	0.934 86930 9	0.948 16630 4	0.937 74226 2	PCA(n_components=28)
RandomForestClassifier	0.948 34785 8	0.934 99134 3	0.948 34785 8	0.937 70042 2	RFE (estimator=LogisticRegression(), n_features_to_select=28)
RandomForestClassifier	0.948 43863 5	0.935 05776 3	0.948 43863 5	0.937 67872 2	RFE (estimator=MLPClassifier(), n_features_to_select=28)
RandomForestClassifier	0.948 16630 4	0.934 74040 5	0.948 16630 4	0.937 56881 6	RFECV (estimator=RandomForestClassifier( ) , min_features_to_select=28)
BaggingClassifier	0.947 98475	0.934 55997 4	0.947 98475	0.937 52419 7	RFE (estimator=LogisticRegression(), n_features_to_select=28)
RandomForestClassifier	0.948 16630 4	0.934 67640 3	0.948 16630 4	0.937 48144 2	RFE (estimator=RandomForestClassifier( ) , n_features_to_select=28)
RandomForestClassifier	0.948 16630 4	0.934 61271 1	0.948 16630 4	0.937 39363 1	RFECV (estimator=MLPClassifier(), min_features_to_select=28)

XGBClassifier	0.949 52795 9	0.936 19940 5	0.949 52795 9	0.937 37552 3	SparsePCA(n_components=28)
BaggingClassifier	0.948 07552 7	0.934 48656 6	0.948 07552 7	0.937 32802 5	MiniBatchSparsePCA(n_components=28)
BaggingClassifier	0.947 53086 4	0.933 96500 2	0.947 53086 4	0.937 19606 2	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
XGBClassifier	0.948 98329 7	0.935 34732 8	0.948 98329 7	0.937 16982 6	PCA(n_components=28)
BaggingClassifier	0.946 89542 5	0.933 40437 8	0.946 89542 5	0.936 99408	SelectKBest (mutual_info_classif, k=28)
BaggingClassifier	0.946 98620 2	0.933 35830 1	0.946 98620 2	0.936 88939 5	RFE (estimator=SVC(), n_features_to_select=28)
BaggingClassifier	0.946 35076 3	0.933 01918 5	0.946 35076 3	0.936 85318 8	RFE (estimator=RandomForestClassifier( ) , n_features_to_select=28)
BaggingClassifier	0.946 53231 7	0.932 89749 5	0.946 53231 7	0.936 64822 4	SelectKBest (f_regression, k=28)
BaggingClassifier	0.946 71387 1	0.932 95095 1	0.946 71387 1	0.936 60810 1	RFE (estimator=MLPClassifier(), n_features_to_select=28)
BaggingClassifier	0.946 44154	0.932 46398 6	0.946 44154	0.936 24099 8	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
XGBClassifier	0.948 07552 7	0.933 67385	0.948 07552 7	0.936 14606 6	MiniBatchSparsePCA(n_components=28)
BaggingClassifier	0.946 89542 5	0.932 62650 7	0.946 89542 5	0.936 12717 5	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
BaggingClassifier	0.946 44154	0.932 21781 3	0.946 44154	0.935 98002 4	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
BaggingClassifier	0.946 35076 3	0.931 0285	0.946 35076 3	0.934 73941 2	SparsePCA(n_components=28)
BaggingClassifier	0.945 17066 1	0.930 28362 9	0.945 17066 1	0.934 63945 4	RFECV (estimator=SVC(), min_features_to_select=28)
XGBClassifier	0.948 16630 4	0.932 15920 3	0.948 16630 4	0.932 59281 2	SelectKBest (f_classif, k=28)
XGBClassifier	0.948 16630 4	0.932 15920 3	0.948 16630 4	0.932 59281 2	SelectKBest (mutual_info_classif, k=28)
XGBClassifier	0.948 16630 4	0.932 15920 3	0.948 16630 4	0.932 59281 2	SelectKBest (f_regression, k=28)

XGBClassifier	0.948 16630 4	0.932 15920 3	0.948 16630 4	0.932 59281 2	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
XGBClassifier	0.948 16630 4	0.932 15920 3	0.948 16630 4	0.932 59281 2	RFE (estimator=RandomForestClassifier( ) , n_features_to_select=28)
XGBClassifier	0.948 16630 4	0.932 15920 3	0.948 16630 4	0.932 59281 2	RFE (estimator=LogisticRegression(), n_features_to_select=28)
XGBClassifier	0.948 16630 4	0.932 15920 3	0.948 16630 4	0.932 59281 2	RFE (estimator=SVC(), n_features_to_select=28)
XGBClassifier	0.948 16630 4	0.932 15920 3	0.948 16630 4	0.932 59281 2	RFE (estimator=MLPClassifier(), n_features_to_select=28)
XGBClassifier	0.948 16630 4	0.932 15920 3	0.948 16630 4	0.932 59281 2	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
XGBClassifier	0.948 16630 4	0.932 15920 3	0.948 16630 4	0.932 59281 2	RFECV (estimator=RandomForestClassifier( ) , min_features_to_select=28)
XGBClassifier	0.948 16630 4	0.932 15920 3	0.948 16630 4	0.932 59281 2	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
XGBClassifier	0.948 16630 4	0.932 15920 3	0.948 16630 4	0.932 59281 2	RFECV (estimator=SVC(), min_features_to_select=28)
XGBClassifier	0.948 16630 4	0.932 15920 3	0.948 16630 4	0.932 59281 2	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
ExtraTreeClassifier	0.934 7313	0.930 16355 1	0.934 7313	0.932 32121 6	MiniBatchSparsePCA(n_components=28)
ExtraTreeClassifier	0.934 64052 3	0.929 45204 5	0.934 64052 3	0.931 88539 3	RFE (estimator=RandomForestClassifier( ) , n_features_to_select=28)
LGBMClassifier	0.948 89252	0.934 25087 4	0.948 89252	0.931 78318 4	MiniBatchSparsePCA(n_components=28)
HistGradientBoostingClassifier	0.949 34640 5	0.936 14096 3	0.949 34640 5	0.931 71258 2	PCA(n_components=28)
ExtraTreeClassifier	0.934 00508 4	0.929 04075 1	0.934 00508 4	0.931 38090 6	RFE (estimator=LogisticRegression(), n_features_to_select=28)
LGBMClassifier	0.948 80174 3	0.934 15502 6	0.948 80174 3	0.931 10428 2	PCA(n_components=28)
ExtraTreeClassifier	0.933 64197 5	0.927 49598 2	0.933 64197 5	0.930 35953 2	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
ExtraTreeClassifier	0.932 46187 4	0.928 23060 7	0.932 46187 4	0.930 25134 7	SparsePCA(n_components=28)

KNeighborsClassifier	0.945 8061	0.926 47585	0.945 8061	0.930 10903 1	SparsePCA(n_components=28)
SelfTrainingClassifier	0.945 8061	0.926 47585	0.945 8061	0.930 10903 1	SparsePCA(n_components=28)
ExtraTreeClassifier	0.932 46187 4	0.927 88507 2	0.932 46187 4	0.930 06298 8	SelectKBest (f_classif, k=28)
ExtraTreeClassifier	0.932 46187 4	0.927 88507 2	0.932 46187 4	0.930 06298 8	SelectKBest (f_regression, k=28)
ExtraTreeClassifier	0.931 73565 7	0.927 91695 5	0.931 73565 7	0.929 75182 8	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.947 80319 5	0.930 67989 6	0.947 80319 5	0.929 69645 2	SparsePCA(n_components=28)
ExtraTreeClassifier	0.930 37400 2	0.928 96449 3	0.930 37400 2	0.929 65953 5	PCA(n_components=28)
ExtraTreeClassifier	0.932 64342 8	0.926 91787	0.932 64342 8	0.929 60883 9	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
HistGradientBoostingClassifier	0.947 25853 3	0.928 98025 7	0.947 25853 3	0.929 60218 4	MiniBatchSparsePCA(n_components=28)
LGBMClassifier	0.947 62164 1	0.930 04292 3	0.947 62164 1	0.929 45191 8	SparsePCA(n_components=28)
ExtraTreeClassifier	0.931 82643 4	0.927 25904 9	0.931 82643 4	0.929 43681 9	RFECV (estimator=SVC(), min_features_to_select=28)
KNeighborsClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	SelectKBest (mutual_info_classif, k=28)
SelfTrainingClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	SelectKBest (mutual_info_classif, k=28)
KNeighborsClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	SelectKBest (f_regression, k=28)
SelfTrainingClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	SelectKBest (f_regression, k=28)
KNeighborsClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
SelfTrainingClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)

SelfTrainingClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	RFE (estimator=LogisticRegression(), n_features_to_select=28)
SelfTrainingClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	RFE (estimator=LogisticRegression(), n_features_to_select=28)
KNeighborsClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	RFE (estimator=SVC(), n_features_to_select=28)
SelfTrainingClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	RFE (estimator=SVC(), n_features_to_select=28)
KNeighborsClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	RFE (estimator=MLPClassifier(), n_features_to_select=28)
SelfTrainingClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	RFE (estimator=MLPClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
SelfTrainingClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
KNeighborsClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	RFECV (estimator=SVC(), min_features_to_select=28)
SelfTrainingClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	RFECV (estimator=SVC(), min_features_to_select=28)
KNeighborsClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	PCA(n_components=28)

SelfTrainingClassifier	0.944 98910 7	0.924 26383 4	0.944 98910 7	0.928 87882 8	PCA(n_components=28)
DecisionTreeClassifier	0.930 46477 9	0.927 03786	0.930 46477 9	0.928 69528 7	SelectKBest (f_regression, k=28)
DecisionTreeClassifier	0.930 46477 9	0.927 03786	0.930 46477 9	0.928 69528 7	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.944 26289	0.923 27563 7	0.944 26289	0.928 64668 9	MiniBatchSparsePCA(n_components=28)
SelfTrainingClassifier	0.944 26289	0.923 27563 7	0.944 26289	0.928 64668 9	MiniBatchSparsePCA(n_components=28)
DecisionTreeClassifier	0.930 28322 4	0.927 08124 6	0.930 28322 4	0.928 63365 2	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
ExtraTreeClassifier	0.930 46477 9	0.926 92051 6	0.930 46477 9	0.928 63282 6	RFE (estimator=SVC(), n_features_to_select=28)
HistGradientBoostingClassifier	0.947 53086 4	0.929 64665 1	0.947 53086 4	0.928 60779	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.944 89833	0.923 84209 3	0.944 89833	0.928 58311 7	SelectKBest (f_classif, k=28)
SelfTrainingClassifier	0.944 89833	0.923 84209 3	0.944 89833	0.928 58311 7	SelectKBest (f_classif, k=28)
ExtraTreeClassifier	0.930 82788 7	0.926 48125 9	0.930 82788 7	0.928 56388	RFE (estimator=MLPClassifier(), n_features_to_select=28)
LGBMClassifier	0.948 07552 7	0.932 10557 1	0.948 07552 7	0.928 54199	SelectKBest (f_classif, k=28)
LGBMClassifier	0.948 07552 7	0.932 10557 1	0.948 07552 7	0.928 54199	SelectKBest (mutual_info_classif, k=28)
LGBMClassifier	0.948 07552 7	0.932 10557 1	0.948 07552 7	0.928 54199	SelectKBest (f_regression, k=28)
LGBMClassifier	0.948 07552 7	0.932 10557 1	0.948 07552 7	0.928 54199	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
LGBMClassifier	0.948 07552 7	0.932 10557 1	0.948 07552 7	0.928 54199	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
LGBMClassifier	0.948 07552 7	0.932 10557 1	0.948 07552 7	0.928 54199	RFE (estimator=LogisticRegression(), n_features_to_select=28)
LGBMClassifier	0.948 07552 7	0.932 10557 1	0.948 07552 7	0.928 54199	RFE (estimator=SVC(), n_features_to_select=28)

LGBMClassifier	0.948 07552 7	0.932 10557 1	0.948 07552 7	0.928 54199	RFE (estimator=MLPClassifier(), n_features_to_select=28)
LGBMClassifier	0.948 07552 7	0.932 10557 1	0.948 07552 7	0.928 54199	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
LGBMClassifier	0.948 07552 7	0.932 10557 1	0.948 07552 7	0.928 54199	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
LGBMClassifier	0.948 07552 7	0.932 10557 1	0.948 07552 7	0.928 54199	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
LGBMClassifier	0.948 07552 7	0.932 10557 1	0.948 07552 7	0.928 54199	RFECV (estimator=SVC(), min_features_to_select=28)
LGBMClassifier	0.948 07552 7	0.932 10557 1	0.948 07552 7	0.928 54199	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.929 92011 6	0.927 16964 4	0.929 92011 6	0.928 50951 4	RFE (estimator=SVC(), n_features_to_select=28)
DecisionTreeClassifier	0.930 19244 7	0.926 80969 5	0.930 19244 7	0.928 44729	RFE (estimator=LogisticRegression(), n_features_to_select=28)
ExtraTreeClassifier	0.930 64633 3	0.926 40551 3	0.930 64633 3	0.928 44036 4	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.930 10167	0.926 77300 5	0.930 10167	0.928 38548 8	SelectKBest (mutual_info_classif, k=28)
ExtraTreeClassifier	0.931 00944 1	0.925 96333 7	0.931 00944 1	0.928 36432	SelectKBest (mutual_info_classif, k=28)
DecisionTreeClassifier	0.928 46768 3	0.928 00706 1	0.928 46768 3	0.928 23642 4	MiniBatchSparsePCA(n_components=28)
HistGradientBoostingClassifier	0.947 53086 4	0.929 65296 8	0.947 53086 4	0.928 06754 2	SelectKBest (f_classif, k=28)
DecisionTreeClassifier	0.929 37545 4	0.926 71992 6	0.929 37545 4	0.928 01567 7	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.929 64778 5	0.926 47323 1	0.929 64778 5	0.928 01444 8	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
MLPClassifier	0.945 89687 7	0.924 75984	0.945 89687 7	0.927 97770 1	SparsePCA(n_components=28)
ExtraTreeClassifier	0.930 37400 2	0.925 57608 1	0.930 37400 2	0.927 86837 4	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.929 1939	0.926 53106 6	0.929 1939	0.927 83062 7	SelectKBest (f_classif, k=28)

DecisionTreeClassifier	0.929 01234 6	0.926 69664 2	0.929 01234 6	0.927 83050 7	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.947 53086 4	0.929 68210 4	0.947 53086 4	0.927 79243 1	RFE (estimator=LogisticRegression(), n_features_to_select=28)
LabelPropagation	0.945 71532 3	0.924 18779 7	0.945 71532 3	0.927 73601 6	SelectKBest (mutual_info_classif, k=28)
LabelPropagation	0.945 71532 3	0.924 18779 7	0.945 71532 3	0.927 73601 6	SelectKBest (f_regression, k=28)
LabelPropagation	0.945 71532 3	0.924 18779 7	0.945 71532 3	0.927 73601 6	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
LabelPropagation	0.945 71532 3	0.924 18779 7	0.945 71532 3	0.927 73601 6	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
LabelPropagation	0.945 71532 3	0.924 18779 7	0.945 71532 3	0.927 73601 6	RFE (estimator=LogisticRegression(), n_features_to_select=28)
LabelPropagation	0.945 71532 3	0.924 18779 7	0.945 71532 3	0.927 73601 6	RFE (estimator=SVC(), n_features_to_select=28)
LabelPropagation	0.945 71532 3	0.924 18779 7	0.945 71532 3	0.927 73601 6	RFE (estimator=MLPClassifier(), n_features_to_select=28)
LabelPropagation	0.945 71532 3	0.924 18779 7	0.945 71532 3	0.927 73601 6	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
LabelPropagation	0.945 71532 3	0.924 18779 7	0.945 71532 3	0.927 73601 6	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
LabelPropagation	0.945 71532 3	0.924 18779 7	0.945 71532 3	0.927 73601 6	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
LabelPropagation	0.945 71532 3	0.924 18779 7	0.945 71532 3	0.927 73601 6	RFECV (estimator=SVC(), min_features_to_select=28)
LabelPropagation	0.945 71532 3	0.924 18779 7	0.945 71532 3	0.927 73601 6	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
MLPClassifier	0.946 89542 5	0.927 14316 7	0.946 89542 5	0.927 67555 4	SelectKBest (mutual_info_classif, k=28)
MLPClassifier	0.947 25853 3	0.928 47564 1	0.947 25853 3	0.927 48710 4	SelectKBest (f_regression, k=28)
LabelSpreading	0.945 89687 7	0.924 29483 4	0.945 89687 7	0.927 45918 7	SelectKBest (mutual_info_classif, k=28)
LabelSpreading	0.945 89687 7	0.924 29483 4	0.945 89687 7	0.927 45918 7	SelectKBest (f_regression, k=28)



LabelSpreading	0.945 89687 7	0.924 29483 4	0.945 89687 7	0.927 45918 7	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
LabelSpreading	0.945 89687 7	0.924 29483 4	0.945 89687 7	0.927 45918 7	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
LabelSpreading	0.945 89687 7	0.924 29483 4	0.945 89687 7	0.927 45918 7	RFE (estimator=LogisticRegression(), n_features_to_select=28)
LabelSpreading	0.945 89687 7	0.924 29483 4	0.945 89687 7	0.927 45918 7	RFE (estimator=SVC(), n_features_to_select=28)
LabelSpreading	0.945 89687 7	0.924 29483 4	0.945 89687 7	0.927 45918 7	RFE (estimator=MLPClassifier(), n_features_to_select=28)
LabelSpreading	0.945 89687 7	0.924 29483 4	0.945 89687 7	0.927 45918 7	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
LabelSpreading	0.945 89687 7	0.924 29483 4	0.945 89687 7	0.927 45918 7	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
LabelSpreading	0.945 89687 7	0.924 29483 4	0.945 89687 7	0.927 45918 7	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
LabelSpreading	0.945 89687 7	0.924 29483 4	0.945 89687 7	0.927 45918 7	RFECV (estimator=SVC(), min_features_to_select=28)
LabelSpreading	0.945 89687 7	0.924 29483 4	0.945 89687 7	0.927 45918 7	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.929 1939	0.925 69840 7	0.929 1939	0.927 39206 8	SparsePCA(n_components=28)
MLPClassifier	0.947 07697 9	0.927 70908 2	0.947 07697 9	0.927 37623 4	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
MLPClassifier	0.947 07697 9	0.927 70908 2	0.947 07697 9	0.927 37623 4	PCA(n_components=28)
LabelPropagation	0.946 25998 6	0.925 01066 6	0.946 25998 6	0.927 28486 8	SparsePCA(n_components=28)
HistGradientBoostingClassifier	0.947 80319 5	0.931 38726 2	0.947 80319 5	0.927 25402 5	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
LabelPropagation	0.945 53376 9	0.923 34950 7	0.945 53376 9	0.927 23486 7	SelectKBest (f_classif, k=28)
LabelPropagation	0.945 53376 9	0.923 34950 7	0.945 53376 9	0.927 23486 7	PCA(n_components=28)
HistGradientBoostingClassifier	0.947 53086 4	0.929 80797 3	0.947 53086 4	0.927 23193 3	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)

LabelSpreading	0.946 53231 7	0.925 71404	0.946 53231 7	0.927 18102 8	SparsePCA(n_components=28)
HistGradientBoostingClassifier	0.947 34931	0.928 8954	0.947 34931	0.927 12205 8	SelectKBest (mutual_info_classif, k=28)
MLPClassifier	0.946 16920 8	0.924 62230 3	0.946 16920 8	0.927 09471 6	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
HistGradientBoostingClassifier	0.947 16775 6	0.928 03447 5	0.947 16775 6	0.927 01228 2	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
LabelSpreading	0.945 71532 3	0.923 42181 6	0.945 71532 3	0.926 95099 4	SelectKBest (f_classif, k=28)
LabelSpreading	0.945 71532 3	0.923 42181 6	0.945 71532 3	0.926 95099 4	PCA(n_components=28)
DecisionTreeClassifier	0.928 10457 5	0.925 63785 6	0.928 10457 5	0.926 84542 5	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.927 83224 4	0.925 77501 8	0.927 83224 4	0.926 78583	RFECV (estimator=SVC(), min_features_to_select=28)
LabelSpreading	0.945 44299 2	0.922 69701 7	0.945 44299 2	0.926 78439 5	MiniBatchSparsePCA(n_components=28)
MLPClassifier	0.947 98475	0.933 24720 1	0.947 98475	0.926 78141 1	SelectKBest (f_classif, k=28)
MLPClassifier	0.946 25998 6	0.924 47939 1	0.946 25998 6	0.926 60358 2	MiniBatchSparsePCA(n_components=28)
HistGradientBoostingClassifier	0.947 16775 6	0.928 01569 2	0.947 16775 6	0.926 58505 1	RFE (estimator=MLPClassifier(), n_features_to_select=28)
HistGradientBoostingClassifier	0.946 62309 4	0.925 64299 2	0.946 62309 4	0.926 54316 5	RFECV (estimator=SVC(), min_features_to_select=28)
DecisionTreeClassifier	0.927 65069	0.925 46814 7	0.927 65069	0.926 53963 2	RFE (estimator=MLPClassifier(), n_features_to_select=28)
MLPClassifier	0.945 89687 7	0.923 41557 1	0.945 89687 7	0.926 52167 4	RFECV (estimator=SVC(), min_features_to_select=28)
VotingClassifier	0.947 16775 6	0.928 01871 2	0.947 16775 6	0.926 44085 7	SparsePCA(n_components=28)
MLPClassifier	0.945 89687 7	0.923 28122 6	0.945 89687 7	0.926 38450 4	RFE (estimator=LogisticRegression(), n_features_to_select=28)
LabelPropagation	0.944 35366 7	0.920 35657 3	0.944 35366 7	0.926 25040 3	MiniBatchSparsePCA(n_components=28)

HistGradientBoostingClassifier	0.946 98620 2	0.927 07137 6	0.946 98620 2	0.926 18807 7	SelectKBest (f_regression, k=28)
MLPClassifier	0.947 44008 7	0.929 84838 9	0.947 44008 7	0.926 01472 4	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.947 34931	0.929 24018 4	0.947 34931	0.925 96144 4	RFE (estimator=SVC(), n_features_to_select=28)
MLPClassifier	0.946 89542 5	0.926 54692 4	0.946 89542 5	0.925 84257 2	RFE (estimator=MLPClassifier(), n_features_to_select=28)
MLPClassifier	0.947 07697 9	0.927 54730 5	0.947 07697 9	0.925 80172 5	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.947 44008 7	0.930 09418 4	0.947 44008 7	0.925 71485 2	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.927 46913 6	0.923 81891 3	0.927 46913 6	0.925 59086 8	PCA(n_components=28)
VotingClassifier	0.946 89542 5	0.926 49045 3	0.946 89542 5	0.925 54718 5	PCA(n_components=28)
LinearDiscriminantAnalysis	0.940 81336 2	0.917 33658 2	0.940 81336 2	0.925 54414 4	SelectKBest (f_classif, k=28)
LinearDiscriminantAnalysis	0.940 81336 2	0.917 33658 2	0.940 81336 2	0.925 54414 4	SelectKBest (mutual_info_classif, k=28)
LinearDiscriminantAnalysis	0.940 81336 2	0.917 33658 2	0.940 81336 2	0.925 54414 4	SelectKBest (f_regression, k=28)
LinearDiscriminantAnalysis	0.940 81336 2	0.917 33658 2	0.940 81336 2	0.925 54414 4	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.940 81336 2	0.917 33658 2	0.940 81336 2	0.925 54414 4	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.940 81336 2	0.917 33658 2	0.940 81336 2	0.925 54414 4	RFE (estimator=LogisticRegression(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.940 81336 2	0.917 33658 2	0.940 81336 2	0.925 54414 4	RFE (estimator=SVC(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.940 81336 2	0.917 33658 2	0.940 81336 2	0.925 54414 4	RFE (estimator=MLPClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.940 81336 2	0.917 33658 2	0.940 81336 2	0.925 54414 4	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.940 81336 2	0.917 33658 2	0.940 81336 2	0.925 54414 4	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)

LinearDiscriminantAnalysis	0.940 81336 2	0.917 33658 2	0.940 81336 2	0.925 54414 4	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.940 81336 2	0.917 33658 2	0.940 81336 2	0.925 54414 4	RFECV (estimator=SVC(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.940 81336 2	0.917 33658 2	0.940 81336 2	0.925 54414 4	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.940 81336 2	0.917 33658 2	0.940 81336 2	0.925 54414 4	PCA(n_components=28)
LinearDiscriminantAnalysis	0.940 81336 2	0.917 33658 2	0.940 81336 2	0.925 54414 4	SparsePCA(n_components=28)
LinearDiscriminantAnalysis	0.940 81336 2	0.917 33658 2	0.940 81336 2	0.925 54414 4	MiniBatchSparsePCA(n_components=28)
VotingClassifier	0.947 62164 1	0.931 99786 1	0.947 62164 1	0.925 51595 3	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
VotingClassifier	0.946 98620 2	0.927 01454 7	0.946 98620 2	0.925 45083 3	MiniBatchSparsePCA(n_components=28)
VotingClassifier	0.947 44008 7	0.930 41466 4	0.947 44008 7	0.925 41111 6	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
VotingClassifier	0.947 62164 1	0.932 28765 2	0.947 62164 1	0.925 36215 2	RFE (estimator=LogisticRegression(), n_features_to_select=28)
VotingClassifier	0.947 62164 1	0.932 28765 2	0.947 62164 1	0.925 36215 2	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
SVC	0.947 62164 1	0.932 28765 2	0.947 62164 1	0.925 36215 2	SparsePCA(n_components=28)
GradientBoostingClassifier	0.947 80319 5	0.934 60447 3	0.947 80319 5	0.925 31132 6	SparsePCA(n_components=28)
VotingClassifier	0.947 53086 4	0.931 42782 9	0.947 53086 4	0.925 30995 5	SelectKBest (f_regression, k=28)
VotingClassifier	0.947 44008 7	0.930 60973 3	0.947 44008 7	0.925 25777 6	SelectKBest (mutual_info_classif, k=28)
VotingClassifier	0.947 44008 7	0.930 60973 3	0.947 44008 7	0.925 25777 6	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
VotingClassifier	0.947 34931	0.929 83038 1	0.947 34931	0.925 20561 4	SelectKBest (f_classif, k=28)
MLPClassifier	0.947 34931	0.929 83038 1	0.947 34931	0.925 20561 4	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)

VotingClassifier	0.947 34931	0.929 83038 1	0.947 34931	0.925 20561 4	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
VotingClassifier	0.947 53086 4	0.931 70161	0.947 53086 4	0.925 15538 8	RFECV (estimator=SVC(), min_features_to_select=28)
HistGradientBoostingClassifier	0.947 25853 3	0.929 08707 3	0.947 25853 3	0.925 15346 9	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
VotingClassifier	0.947 44008 7	0.930 83277 6	0.947 44008 7	0.925 10344 2	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.947 62164 1	0.932 99321 3	0.947 62164 1	0.925 05154	PCA(n_components=28)
VotingClassifier	0.947 34931	0.930 00716 2	0.947 34931	0.925 05151 2	RFE (estimator=SVC(), n_features_to_select=28)
VotingClassifier	0.947 34931	0.930 00716 2	0.947 34931	0.925 05151 2	RFE (estimator=MLPClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.947 25853 3	0.929 37790 1	0.947 25853 3	0.924 84473 2	MiniBatchSparsePCA(n_components=28)
MLPClassifier	0.946 71387 1	0.925 15409 5	0.946 71387 1	0.924 53498 8	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
MLPClassifier	0.946 98620 2	0.927 22895 6	0.946 98620 2	0.924 37845 4	RFE (estimator=SVC(), n_features_to_select=28)
GradientBoostingClassifier	0.947 34931	0.931 83096 2	0.947 34931	0.924 10555 1	RFE (estimator=MLPClassifier(), n_features_to_select=28)
SVC	0.947 71241 8	0.940 08231 2	0.947 71241 8	0.923 98174 6	SelectKBest (f_classif, k=28)
SVC	0.947 71241 8	0.940 08231 2	0.947 71241 8	0.923 98174 6	SelectKBest (mutual_info_classif, k=28)
SVC	0.947 71241 8	0.940 08231 2	0.947 71241 8	0.923 98174 6	SelectKBest (f_regression, k=28)
SVC	0.947 71241 8	0.940 08231 2	0.947 71241 8	0.923 98174 6	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
SVC	0.947 71241 8	0.940 08231 2	0.947 71241 8	0.923 98174 6	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
SVC	0.947 71241 8	0.940 08231 2	0.947 71241 8	0.923 98174 6	RFE (estimator=LogisticRegression(), n_features_to_select=28)
SVC	0.947 71241 8	0.940 08231 2	0.947 71241 8	0.923 98174 6	RFE (estimator=SVC(), n_features_to_select=28)

SVC	0.947 71241 8	0.940 08231 2	0.947 71241 8	0.923 98174 6	RFE (estimator=MLPClassifier(), n_features_to_select=28)
SVC	0.947 71241 8	0.940 08231 2	0.947 71241 8	0.923 98174 6	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
SVC	0.947 71241 8	0.940 08231 2	0.947 71241 8	0.923 98174 6	RFECV (estimator=RandomForestClassifier( ) , min_features_to_select=28)
SVC	0.947 71241 8	0.940 08231 2	0.947 71241 8	0.923 98174 6	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
SVC	0.947 71241 8	0.940 08231 2	0.947 71241 8	0.923 98174 6	RFECV (estimator=SVC(), min_features_to_select=28)
SVC	0.947 71241 8	0.940 08231 2	0.947 71241 8	0.923 98174 6	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
SVC	0.947 71241 8	0.940 08231 2	0.947 71241 8	0.923 98174 6	PCA(n_components=28)
GradientBoostingClassifier	0.947 25853 3	0.931 04324 4	0.947 25853 3	0.923 89403 7	SelectKBest (f_classif, k=28)
GradientBoostingClassifier	0.947 25853 3	0.931 04324 4	0.947 25853 3	0.923 89403 7	SelectKBest (mutual_info_classif, k=28)
GradientBoostingClassifier	0.947 25853 3	0.931 04324 4	0.947 25853 3	0.923 89403 7	SelectKBest (f_regression, k=28)
GradientBoostingClassifier	0.947 25853 3	0.931 04324 4	0.947 25853 3	0.923 89403 7	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.947 25853 3	0.931 04324 4	0.947 25853 3	0.923 89403 7	RFE (estimator=RandomForestClassifier( ) , n_features_to_select=28)
GradientBoostingClassifier	0.947 25853 3	0.931 04324 4	0.947 25853 3	0.923 89403 7	RFE (estimator=LogisticRegression(), n_features_to_select=28)
GradientBoostingClassifier	0.947 25853 3	0.931 04324 4	0.947 25853 3	0.923 89403 7	RFE (estimator=SVC(), n_features_to_select=28)
GradientBoostingClassifier	0.947 25853 3	0.931 04324 4	0.947 25853 3	0.923 89403 7	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
GradientBoostingClassifier	0.947 25853 3	0.931 04324 4	0.947 25853 3	0.923 89403 7	RFECV (estimator=RandomForestClassifier( ) , min_features_to_select=28)
GradientBoostingClassifier	0.947 25853 3	0.931 04324 4	0.947 25853 3	0.923 89403 7	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
GradientBoostingClassifier	0.947 25853 3	0.931 04324 4	0.947 25853 3	0.923 89403 7	RFECV (estimator=SVC(), min_features_to_select=28)

GradientBoostingClassifier	0.947 25853 3	0.931 04324 4	0.947 25853 3	0.923 89403 7	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.946 98620 2	0.927 66904 4	0.946 98620 2	0.923 74349 8	PCA(n_components=28)
AdaBoostClassifier	0.946 98620 2	0.927 66904 4	0.946 98620 2	0.923 74349 8	SparsePCA(n_components=28)
SVC	0.947 16775 6	0.930 20123	0.947 16775 6	0.923 68197 9	MiniBatchSparsePCA(n_components=28)
SGDClassifier	0.945 07988 4	0.916 96564 2	0.945 07988 4	0.923 45858 3	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
PassiveAggressiveClassifier	0.937 00072 6	0.914 08793 7	0.937 00072 6	0.923 31080 8	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
GaussianNB	0.933 27886 7	0.915 62767 6	0.933 27886 7	0.923 24827	SelectKBest (f_classif, k=28)
GaussianNB	0.933 27886 7	0.915 62767 6	0.933 27886 7	0.923 24827	SelectKBest (mutual_info_classif, k=28)
GaussianNB	0.933 27886 7	0.915 62767 6	0.933 27886 7	0.923 24827	SelectKBest (f_regression, k=28)
GaussianNB	0.933 27886 7	0.915 62767 6	0.933 27886 7	0.923 24827	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
GaussianNB	0.933 27886 7	0.915 62767 6	0.933 27886 7	0.923 24827	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
GaussianNB	0.933 27886 7	0.915 62767 6	0.933 27886 7	0.923 24827	RFE (estimator=LogisticRegression(), n_features_to_select=28)
GaussianNB	0.933 27886 7	0.915 62767 6	0.933 27886 7	0.923 24827	RFE (estimator=SVC(), n_features_to_select=28)
GaussianNB	0.933 27886 7	0.915 62767 6	0.933 27886 7	0.923 24827	RFE (estimator=MLPClassifier(), n_features_to_select=28)
GaussianNB	0.933 27886 7	0.915 62767 6	0.933 27886 7	0.923 24827	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
GaussianNB	0.933 27886 7	0.915 62767 6	0.933 27886 7	0.923 24827	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
GaussianNB	0.933 27886 7	0.915 62767 6	0.933 27886 7	0.923 24827	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
GaussianNB	0.933 27886 7	0.915 62767 6	0.933 27886 7	0.923 24827	RFECV (estimator=SVC(), min_features_to_select=28)

GaussianNB	0.933 27886 7	0.915 62767 6	0.933 27886 7	0.923 24827	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.946 62309 4	0.923 93825 1	0.946 62309 4	0.923 22119 3	MiniBatchSparsePCA(n_components=28)
LogisticRegression	0.945 98765 4	0.919 50470 5	0.945 98765 4	0.923 19236	SelectKBest (f_classif, k=28)
LogisticRegression	0.945 98765 4	0.919 50470 5	0.945 98765 4	0.923 19236	SelectKBest (mutual_info_classif, k=28)
LogisticRegression	0.945 98765 4	0.919 50470 5	0.945 98765 4	0.923 19236	SelectKBest (f_regression, k=28)
LogisticRegression	0.945 98765 4	0.919 50470 5	0.945 98765 4	0.923 19236	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
LogisticRegression	0.945 98765 4	0.919 50470 5	0.945 98765 4	0.923 19236	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
LogisticRegression	0.945 98765 4	0.919 50470 5	0.945 98765 4	0.923 19236	RFE (estimator=LogisticRegression(), n_features_to_select=28)
LogisticRegression	0.945 98765 4	0.919 50470 5	0.945 98765 4	0.923 19236	RFE (estimator=SVC(), n_features_to_select=28)
LogisticRegression	0.945 98765 4	0.919 50470 5	0.945 98765 4	0.923 19236	RFE (estimator=MLPClassifier(), n_features_to_select=28)
LogisticRegression	0.945 98765 4	0.919 50470 5	0.945 98765 4	0.923 19236	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
LogisticRegression	0.945 98765 4	0.919 50470 5	0.945 98765 4	0.923 19236	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
LogisticRegression	0.945 98765 4	0.919 50470 5	0.945 98765 4	0.923 19236	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
LogisticRegression	0.945 98765 4	0.919 50470 5	0.945 98765 4	0.923 19236	RFECV (estimator=SVC(), min_features_to_select=28)
LogisticRegression	0.945 98765 4	0.919 50470 5	0.945 98765 4	0.923 19236	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
LogisticRegression	0.945 98765 4	0.919 50470 5	0.945 98765 4	0.923 19236	PCA(n_components=28)
PassiveAggressiveClassifier	0.934 36819 2	0.914 90485 6	0.934 36819 2	0.923 15165 6	SelectKBest (f_regression, k=28)
LogisticRegression	0.945 89687 7	0.918 97326 7	0.945 89687 7	0.923 14232	SparsePCA(n_components=28)



LogisticRegression	0.945 89687 7	0.918 97326 7	0.945 89687 7	0.923 14232	MiniBatchSparsePCA(n_components=28)
SGDClassifier	0.947 07697 9	0.930 16086 7	0.947 07697 9	0.923 14075 4	RFE (estimator=LogisticRegression(), n_features_to_select=28)
SGDClassifier	0.946 44154	0.922 26123 2	0.946 44154	0.923 12198 7	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
PassiveAggressiveClassifier	0.944 17211 3	0.913 83788 4	0.944 17211 3	0.922 79914 3	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
SGDClassifier	0.944 98910 7	0.914 77354 8	0.944 98910 7	0.922 64247 4	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.946 62309 4	0.923 62976 4	0.946 62309 4	0.922 56485 7	SelectKBest (f_classif, k=28)
AdaBoostClassifier	0.946 62309 4	0.923 62976 4	0.946 62309 4	0.922 56485 7	SelectKBest (mutual_info_classif, k=28)
AdaBoostClassifier	0.946 62309 4	0.923 62976 4	0.946 62309 4	0.922 56485 7	SelectKBest (f_regression, k=28)
AdaBoostClassifier	0.946 62309 4	0.923 62976 4	0.946 62309 4	0.922 56485 7	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.946 62309 4	0.923 62976 4	0.946 62309 4	0.922 56485 7	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.946 62309 4	0.923 62976 4	0.946 62309 4	0.922 56485 7	RFE (estimator=LogisticRegression(), n_features_to_select=28)
AdaBoostClassifier	0.946 62309 4	0.923 62976 4	0.946 62309 4	0.922 56485 7	RFE (estimator=SVC(), n_features_to_select=28)
AdaBoostClassifier	0.946 62309 4	0.923 62976 4	0.946 62309 4	0.922 56485 7	RFE (estimator=MLPClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.946 62309 4	0.923 62976 4	0.946 62309 4	0.922 56485 7	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.946 62309 4	0.923 62976 4	0.946 62309 4	0.922 56485 7	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.946 62309 4	0.923 62976 4	0.946 62309 4	0.922 56485 7	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
AdaBoostClassifier	0.946 62309 4	0.923 62976 4	0.946 62309 4	0.922 56485 7	RFECV (estimator=SVC(), min_features_to_select=28)
AdaBoostClassifier	0.946 62309 4	0.923 62976 4	0.946 62309 4	0.922 56485 7	RFECV (estimator=MLPClassifier(), min_features_to_select=28)

RidgeClassifier	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	SelectKBest (f_classif, k=28)
RidgeClassifier CV	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	SelectKBest (f_classif, k=28)
RidgeClassifier	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	SelectKBest (mutual_info_classif, k=28)
RidgeClassifier CV	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	SelectKBest (mutual_info_classif, k=28)
RidgeClassifier	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	SelectKBest (f_regression, k=28)
RidgeClassifier CV	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	SelectKBest (f_regression, k=28)
RidgeClassifier	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
RidgeClassifier CV	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
RidgeClassifier	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	RFE (estimator=RandomForestClassifier( ) , n_features_to_select=28)
RidgeClassifier CV	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	RFE (estimator=RandomForestClassifier( ) , n_features_to_select=28)
RidgeClassifier	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	RFE (estimator=LogisticRegression(), n_features_to_select=28)
RidgeClassifier CV	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	RFE (estimator=LogisticRegression(), n_features_to_select=28)
RidgeClassifier	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	RFE (estimator=SVC(), n_features_to_select=28)
RidgeClassifier CV	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	RFE (estimator=SVC(), n_features_to_select=28)
RidgeClassifier	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	RFE (estimator=MLPClassifier(), n_features_to_select=28)
RidgeClassifier CV	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	RFE (estimator=MLPClassifier(), n_features_to_select=28)
RidgeClassifier	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
RidgeClassifier CV	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)

RidgeClassifier	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	RFECV (estimator=RandomForestClassifier( , min_features_to_select=28)
RidgeClassifier CV	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	RFECV (estimator=RandomForestClassifier( , min_features_to_select=28)
RidgeClassifier	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
RidgeClassifier CV	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
RidgeClassifier	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	RFECV (estimator=SVC(), min_features_to_select=28)
RidgeClassifier CV	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	RFECV (estimator=SVC(), min_features_to_select=28)
RidgeClassifier	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
RidgeClassifier CV	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
RidgeClassifier	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	PCA(n_components=28)
RidgeClassifier CV	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	PCA(n_components=28)
RidgeClassifier	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	SparsePCA(n_components=28)
RidgeClassifier CV	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	SparsePCA(n_components=28)
RidgeClassifier	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	MiniBatchSparsePCA(n_component s=28)
RidgeClassifier CV	0.946 53231 7	0.922 46487 7	0.946 53231 7	0.922 51630 2	MiniBatchSparsePCA(n_component s=28)
SGDClassifier	0.939 99636 9	0.911 80832 9	0.939 99636 9	0.922 48831 4	MiniBatchSparsePCA(n_component s=28)
SGDClassifier	0.946 98620 2	0.931 11908	0.946 98620 2	0.922 42216 6	RFE (estimator=RandomForestClassifier( , n_features_to_select=28)
PassiveAggres siveClassifier	0.945 98765 4	0.917 64880 9	0.945 98765 4	0.922 38895 3	SelectKBest (mutual_info_classif, k=28)
SGDClassifier	0.944 89833	0.912 98648 2	0.944 89833	0.922 12247 9	SparsePCA(n_components=28)

SGDClassifier	0.946 35076 3	0.919 64945 4	0.946 35076 3	0.922 08608 2	PCA(n_components=28)
SGDClassifier	0.946 98620 2	0.934 01527 5	0.946 98620 2	0.922 08062	RFECV (estimator=SVC(), min_features_to_select=28)
SGDClassifier	0.946 53231 7	0.921 82424 6	0.946 53231 7	0.922 01329 3	RFE (estimator=SVC(), n_features_to_select=28)
SGDClassifier	0.946 80464 8	0.927 77888 7	0.946 80464 8	0.921 98571	SelectKBest (f_classif, k=28)
SGDClassifier	0.946 80464 8	0.927 77888 7	0.946 80464 8	0.921 98571	RFE (estimator=MLPClassifier(), n_features_to_select=28)
SGDClassifier	0.946 44154	0.920 42423 2	0.946 44154	0.921 96556 8	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
SGDClassifier	0.946 71387 1	0.925 37922 6	0.946 71387 1	0.921 93825 9	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
SGDClassifier	0.946 89542 5	0.932 15458 3	0.946 89542 5	0.921 86093 6	SelectKBest (f_regression, k=28)
PassiveAggressiveClassifier	0.934 27741 5	0.912 49094 6	0.934 27741 5	0.921 76014	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
SGDClassifier	0.946 25998 6	0.914 17658 4	0.946 25998 6	0.921 18692 9	SelectKBest (mutual_info_classif, k=28)
PassiveAggressiveClassifier	0.943 99055 9	0.908 32257 9	0.943 99055 9	0.921 15904 1	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
PassiveAggressiveClassifier	0.941 17647 1	0.907 63768 4	0.941 17647 1	0.920 84321 8	RFE (estimator=SVC(), n_features_to_select=28)
PassiveAggressiveClassifier	0.933 46042 1	0.909 16622	0.933 46042 1	0.919 74412 8	SelectKBest (f_classif, k=28)
QuadraticDiscriminantAnalysis	0.924 47349 3	0.915 09617 2	0.924 47349 3	0.919 54281 3	SelectKBest (f_classif, k=28)
QuadraticDiscriminantAnalysis	0.924 47349 3	0.915 09617 2	0.924 47349 3	0.919 54281 3	SelectKBest (mutual_info_classif, k=28)
QuadraticDiscriminantAnalysis	0.924 47349 3	0.915 09617 2	0.924 47349 3	0.919 54281 3	SelectKBest (f_regression, k=28)
QuadraticDiscriminantAnalysis	0.924 47349 3	0.915 09617 2	0.924 47349 3	0.919 54281 3	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.924 47349 3	0.915 09617 2	0.924 47349 3	0.919 54281 3	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)

QuadraticDiscriminantAnalyses	0.924 47349 3	0.915 09617 2	0.924 47349 3	0.919 54281 3	RFE (estimator=LogisticRegression(), n_features_to_select=28)
QuadraticDiscriminantAnalyses	0.924 47349 3	0.915 09617 2	0.924 47349 3	0.919 54281 3	RFE (estimator=SVC(), n_features_to_select=28)
QuadraticDiscriminantAnalyses	0.924 47349 3	0.915 09617 2	0.924 47349 3	0.919 54281 3	RFE (estimator=MLPClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalyses	0.924 47349 3	0.915 09617 2	0.924 47349 3	0.919 54281 3	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalyses	0.924 47349 3	0.915 09617 2	0.924 47349 3	0.919 54281 3	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalyses	0.924 47349 3	0.915 09617 2	0.924 47349 3	0.919 54281 3	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
QuadraticDiscriminantAnalyses	0.924 47349 3	0.915 09617 2	0.924 47349 3	0.919 54281 3	RFECV (estimator=SVC(), min_features_to_select=28)
QuadraticDiscriminantAnalyses	0.924 47349 3	0.915 09617 2	0.924 47349 3	0.919 54281 3	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalyses	0.924 47349 3	0.915 09617 2	0.924 47349 3	0.919 54281 3	PCA(n_components=28)
QuadraticDiscriminantAnalyses	0.924 47349 3	0.915 09617 2	0.924 47349 3	0.919 54281 3	SparsePCA(n_components=28)
QuadraticDiscriminantAnalyses	0.924 47349 3	0.915 09617 2	0.924 47349 3	0.919 54281 3	MiniBatchSparsePCA(n_components=28)
GaussianNB	0.923 47494 6	0.915 19705 2	0.923 47494 6	0.919 15217 6	PCA(n_components=28)
GaussianNB	0.923 47494 6	0.915 19705 2	0.923 47494 6	0.919 15217 6	SparsePCA(n_components=28)
PassiveAggressiveClassifier	0.928 01379 8	0.909 17020 8	0.928 01379 8	0.917 79721 4	RFE (estimator=MLPClassifier(), n_features_to_select=28)
PassiveAggressiveClassifier	0.930 01089 3	0.906 27760 7	0.930 01089 3	0.917 07263 3	SparsePCA(n_components=28)
GaussianNB	0.917 39288 3	0.916 32224 1	0.917 39288 3	0.916 85484 1	MiniBatchSparsePCA(n_components=28)
PassiveAggressiveClassifier	0.897 78504	0.917 80480 8	0.897 78504	0.907 12334 6	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
PassiveAggressiveClassifier	0.869 82570 8	0.919 01260 2	0.869 82570 8	0.891 51635 5	RFE (estimator=LogisticRegression(), n_features_to_select=28)

PassiveAggressiveClassifier	0.874 18300 7	0.901 15160 6	0.874 18300 7	0.887 19099 5	RFECV (estimator=SVC(), min_features_to_select=28)
PassiveAggressiveClassifier	0.798 20261 4	0.918 25811 5	0.798 20261 4	0.847 68679 8	PCA(n_components=28)
BernoulliNB	0.770 24328 3	0.925 44953 4	0.770 24328 3	0.830 61467 8	SelectKBest (f_classif, k=28)
BernoulliNB	0.770 24328 3	0.925 44953 4	0.770 24328 3	0.830 61467 8	SelectKBest (mutual_info_classif, k=28)
BernoulliNB	0.770 24328 3	0.925 44953 4	0.770 24328 3	0.830 61467 8	SelectKBest (f_regression, k=28)
BernoulliNB	0.770 24328 3	0.925 44953 4	0.770 24328 3	0.830 61467 8	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
BernoulliNB	0.770 24328 3	0.925 44953 4	0.770 24328 3	0.830 61467 8	RFE (estimator=RandomForestClassifier( ) , n_features_to_select=28)
BernoulliNB	0.770 24328 3	0.925 44953 4	0.770 24328 3	0.830 61467 8	RFE (estimator=LogisticRegression(), n_features_to_select=28)
BernoulliNB	0.770 24328 3	0.925 44953 4	0.770 24328 3	0.830 61467 8	RFE (estimator=SVC(), n_features_to_select=28)
BernoulliNB	0.770 24328 3	0.925 44953 4	0.770 24328 3	0.830 61467 8	RFE (estimator=MLPClassifier(), n_features_to_select=28)
BernoulliNB	0.770 24328 3	0.925 44953 4	0.770 24328 3	0.830 61467 8	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
BernoulliNB	0.770 24328 3	0.925 44953 4	0.770 24328 3	0.830 61467 8	RFECV (estimator=RandomForestClassifier( ) , min_features_to_select=28)
BernoulliNB	0.770 24328 3	0.925 44953 4	0.770 24328 3	0.830 61467 8	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
BernoulliNB	0.770 24328 3	0.925 44953 4	0.770 24328 3	0.830 61467 8	RFECV (estimator=SVC(), min_features_to_select=28)
BernoulliNB	0.770 24328 3	0.925 44953 4	0.770 24328 3	0.830 61467 8	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
BernoulliNB	0.738 10820 6	0.915 99726 4	0.738 10820 6	0.808 35852 8	MiniBatchSparsePCA(n_component s=28)
BernoulliNB	0.735 92955 7	0.922 01219 2	0.735 92955 7	0.807 33723 7	PCA(n_components=28)
BernoulliNB	0.735 92955 7	0.922 01219 2	0.735 92955 7	0.807 33723 7	SparsePCA(n_components=28)

PassiveAggressiveClassifier	0.551 56136 5	0.881 25407 3	0.551 56136 5	0.671 38739 9	MiniBatchSparsePCA(n_components=28)
PassiveAggressiveClassifier	0.447 07697 9	0.869 90950 8	0.447 07697 9	0.580 20789 3	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)



**APPENDIX 4: VULNERABILITY FUNCTION LEVEL DATA RESULTS IN 2002 TO 2016**

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
RandomForestClassifier	0.957143	0.94753	0.957143	0.947079	PCA(n_components=28)
RandomForestClassifier	0.957086	0.94745	0.957086	0.947523	RFE (estimator=MLPClassifier(), n_features_to_select=28)
RandomForestClassifier	0.957029	0.94733	0.957029	0.94732	SelectKBest (mutual_info_classif, k=28)
RandomForestClassifier	0.957029	0.94733	0.957029	0.94732	RFE (estimator=SVC(), n_features_to_select=28)
StackingClassifier	0.956914	0.947139	0.956914	0.946003	RFE (estimator=SVC(), n_features_to_select=28)
StackingClassifier	0.9568	0.946901	0.9568	0.94586	RFECV (estimator=SVC(), min_features_to_select=28)
StackingClassifier	0.9568	0.946881	0.9568	0.946034	MiniBatchSparsePCA(n_components=28)
RandomForestClassifier	0.9568	0.94688	0.9568	0.946822	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
StackingClassifier	0.956743	0.946814	0.956743	0.945582	SelectKBest (f_regression, k=28)
RandomForestClassifier	0.956743	0.946802	0.956743	0.946996	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
RandomForestClassifier	0.956629	0.946666	0.956629	0.947231	SelectKBest (f_regression, k=28)
RandomForestClassifier	0.956629	0.946619	0.956629	0.947018	RFE (estimator=LogisticRegression(), n_features_to_select=28)
StackingClassifier	0.956629	0.946554	0.956629	0.945557	RFE (estimator=MLPClassifier(), n_features_to_select=28)
StackingClassifier	0.956629	0.946554	0.956629	0.945557	SparsePCA(n_components=28)
RandomForestClassifier	0.956571	0.946488	0.956571	0.946813	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
StackingClassifier	0.956571	0.946454	0.956571	0.945337	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
StackingClassifier	0.956571	0.946438	0.956571	0.945455	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
StackingClassifier	0.956514	0.946304	0.956514	0.945531	RFE (estimator=LogisticRegression(), n_features_to_select=28)



StackingClassifier	0.956 514	0.946 297	0.956 514	0.945 648	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
StackingClassifier	0.956 514	0.946 294	0.956 514	0.945 764	PCA(n_components=28)
RandomForestClassifier	0.956 457	0.946 255	0.956 457	0.946 564	SelectKBest (f_classif, k=28)
RandomForestClassifier	0.956 343	0.946 064	0.956 343	0.946 533	RFECV (estimator=SVC(), min_features_to_select=28)
RandomForestClassifier	0.956 343	0.946 041	0.956 343	0.946 424	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
StackingClassifier	0.956 343	0.945 952	0.956 343	0.945 228	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
RandomForestClassifier	0.956 286	0.945 872	0.956 286	0.945 993	MiniBatchSparsePCA(n_components=28)
StackingClassifier	0.956 286	0.945 829	0.956 286	0.945 245	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
RandomForestClassifier	0.956 114	0.945 733	0.956 114	0.946 578	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
StackingClassifier	0.956 229	0.945 711	0.956 229	0.945 203	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
StackingClassifier	0.956 171	0.945 596	0.956 171	0.944 923	SelectKBest (f_classif, k=28)
RandomForestClassifier	0.956	0.945 509	0.956	0.946 386	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
StackingClassifier	0.956 114	0.945 479	0.956 114	0.944 761	SelectKBest (mutual_info_classif, k=28)
RandomForestClassifier	0.955 943	0.945 209	0.955 943	0.945 569	SparsePCA(n_components=28)
BaggingClassifier	0.955 086	0.944 321	0.955 086	0.946 18	RFECV (estimator=SVC(), min_features_to_select=28)
BaggingClassifier	0.955 143	0.944 287	0.955 143	0.946 013	RFE (estimator=SVC(), n_features_to_select=28)
BaggingClassifier	0.954 857	0.943 874	0.954 857	0.945 801	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
BaggingClassifier	0.954 857	0.943 843	0.954 857	0.945 749	RFE (estimator=LogisticRegression(), n_features_to_select=28)
BaggingClassifier	0.954 743	0.943 524	0.954 743	0.945 399	RFE (estimator=MLPClassifier(), n_features_to_select=28)

BaggingClassifier	0.954 743	0.943 434	0.954 743	0.945 237	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
BaggingClassifier	0.954 914	0.943 403	0.954 914	0.944 756	SelectKBest (mutual_info_classif, k=28)
XGBClassifier	0.955 086	0.943 326	0.955 086	0.941 647	PCA(n_components=28)
BaggingClassifier	0.954 629	0.943 054	0.954 629	0.944 77	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
BaggingClassifier	0.954 514	0.942 939	0.954 514	0.944 797	SelectKBest (f_classif, k=28)
BaggingClassifier	0.954 343	0.942 715	0.954 343	0.944 726	SelectKBest (f_regression, k=28)
BaggingClassifier	0.954 343	0.942 65	0.954 343	0.944 616	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
BaggingClassifier	0.954 229	0.942 581	0.954 229	0.944 697	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
BaggingClassifier	0.954 514	0.942 564	0.954 514	0.944 066	MiniBatchSparsePCA(n_components=28)
BaggingClassifier	0.954 171	0.942 465	0.954 171	0.944 6	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
BaggingClassifier	0.954 286	0.942 404	0.954 286	0.944 297	SparsePCA(n_components=28)
HistGradientBoostingClassifier	0.954 114	0.942 39	0.954 114	0.936 887	PCA(n_components=28)
XGBClassifier	0.954 571	0.941 992	0.954 571	0.941 692	SparsePCA(n_components=28)
HistGradientBoostingClassifier	0.953 6	0.941 967	0.953 6	0.934 812	SelectKBest (f_regression, k=28)
XGBClassifier	0.954 514	0.941 931	0.954 514	0.940 269	SelectKBest(f_classif, k=28)
XGBClassifier	0.954 514	0.941 931	0.954 514	0.940 269	SelectKBest(mutual_info_classif, k=28)
XGBClassifier	0.954 514	0.941 931	0.954 514	0.940 269	SelectKBest(f_regression, k=28)
XGBClassifier	0.954 514	0.941 931	0.954 514	0.940 269	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
XGBClassifier	0.954 514	0.941 931	0.954 514	0.940 269	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
XGBClassifier	0.954 514	0.941 931	0.954 514	0.940 269	RFE(estimator=LogisticRegression(), n_features_to_select=28)
XGBClassifier	0.954 514	0.941 931	0.954 514	0.940 269	RFE(estimator=SVC(), n_features_to_select=28)

XGBClassifier	0.954 514	0.941 931	0.954 514	0.940 269	RFE(estimator=MLPClassifier(), n_features_to_select=28)
XGBClassifier	0.954 514	0.941 931	0.954 514	0.940 269	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
XGBClassifier	0.954 514	0.941 931	0.954 514	0.940 269	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
XGBClassifier	0.954 514	0.941 931	0.954 514	0.940 269	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
XGBClassifier	0.954 514	0.941 931	0.954 514	0.940 269	RFECV (estimator=SVC(), min_features_to_select=28)
XGBClassifier	0.954 514	0.941 931	0.954 514	0.940 269	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
XGBClassifier	0.954 457	0.941 702	0.954 457	0.941 344	MiniBatchSparsePCA(n_components=28)
BaggingClassifier	0.953 2	0.940 54	0.953 2	0.943 116	PCA(n_components=28)
LGBMClassifier	0.953 543	0.940 371	0.953 543	0.935 582	SelectKBest (f_classif, k=28)
LGBMClassifier	0.953 543	0.940 371	0.953 543	0.935 582	SelectKBest (mutual_info_classif, k=28)
LGBMClassifier	0.953 543	0.940 371	0.953 543	0.935 582	SelectKBest (f_regression, k=28)
LGBMClassifier	0.953 543	0.940 371	0.953 543	0.935 582	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
LGBMClassifier	0.953 543	0.940 371	0.953 543	0.935 582	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
LGBMClassifier	0.953 543	0.940 371	0.953 543	0.935 582	RFE (estimator=LogisticRegression(), n_features_to_select=28)
LGBMClassifier	0.953 543	0.940 371	0.953 543	0.935 582	RFE (estimator=SVC(), n_features_to_select=28)
LGBMClassifier	0.953 543	0.940 371	0.953 543	0.935 582	RFE (estimator=MLPClassifier(), n_features_to_select=28)
LGBMClassifier	0.953 543	0.940 371	0.953 543	0.935 582	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
LGBMClassifier	0.953 543	0.940 371	0.953 543	0.935 582	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)

LGBMClassifier	0.953 543	0.940 371	0.953 543	0.935 582	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
LGBMClassifier	0.953 543	0.940 371	0.953 543	0.935 582	RFECV (estimator=SVC(), min_features_to_select=28)
LGBMClassifier	0.953 543	0.940 371	0.953 543	0.935 582	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
SVC	0.952 686	0.939 932	0.952 686	0.931 58	MiniBatchSparsePCA(n_components=28)
SVC	0.952 571	0.939 813	0.952 571	0.931 106	SelectKBest (f_classif, k=28)
SVC	0.952 571	0.939 813	0.952 571	0.931 106	SelectKBest (mutual_info_classif, k=28)
SVC	0.952 571	0.939 813	0.952 571	0.931 106	SelectKBest(f_regression, k=28)
SVC	0.952 571	0.939 813	0.952 571	0.931 106	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
SVC	0.952 571	0.939 813	0.952 571	0.931 106	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SVC	0.952 571	0.939 813	0.952 571	0.931 106	RFE(estimator=LogisticRegression(), n_features_to_select=28)
SVC	0.952 571	0.939 813	0.952 571	0.931 106	RFE(estimator=SVC(), n_features_to_select=28)
SVC	0.952 571	0.939 813	0.952 571	0.931 106	RFE(estimator=MLPClassifier(), n_features_to_select=28)
SVC	0.952 571	0.939 813	0.952 571	0.931 106	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
SVC	0.952 571	0.939 813	0.952 571	0.931 106	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
SVC	0.952 571	0.939 813	0.952 571	0.931 106	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
SVC	0.952 571	0.939 813	0.952 571	0.931 106	RFECV(estimator=SVC(), min_features_to_select=28)
SVC	0.952 571	0.939 813	0.952 571	0.931 106	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
SVC	0.952 571	0.939 813	0.952 571	0.931 106	PCA(n_components=28)
HistGradientBoostingClassifier	0.953 2	0.939 795	0.953 2	0.934 111	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.953 314	0.939 71	0.953 314	0.934 82	MiniBatchSparsePCA(n_components=28)
SVC	0.952 686	0.939 549	0.952 686	0.931 682	SparsePCA(n_components=28)
HistGradientBoostingClassifier	0.953 257	0.939 094	0.953 257	0.935 053	SelectKBest(mutual_info_classif, k=28)

HistGradientBoostingClassifier	0.953086	0.93897	0.953086	0.934043	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
HistGradientBoostingClassifier	0.953143	0.938834	0.953143	0.934535	RFE(estimator=SVC(), n_features_to_select=28)
RidgeClassifier	0.952457	0.93864	0.952457	0.930836	MiniBatchSparsePCA(n_components=28)
RidgeClassifierCV	0.952457	0.93864	0.952457	0.930836	MiniBatchSparsePCA(n_components=28)
HistGradientBoostingClassifier	0.952914	0.938194	0.952914	0.933569	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.953086	0.937964	0.953086	0.935299	SparsePCA(n_components=28)
LGBMClassifier	0.952971	0.937805	0.952971	0.934432	PCA(n_components=28)
HistGradientBoostingClassifier	0.952857	0.937778	0.952857	0.933535	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
GradientBoostingClassifier	0.952571	0.93773	0.952571	0.931719	SelectKBest(mutual_info_classif, k=28)
GradientBoostingClassifier	0.952571	0.93773	0.952571	0.931719	SelectKBest(f_regression, k=28)
GradientBoostingClassifier	0.952571	0.93773	0.952571	0.931719	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.952571	0.93773	0.952571	0.931719	RFE(estimator=SVC(), n_features_to_select=28)
GradientBoostingClassifier	0.952571	0.93773	0.952571	0.931719	RFECV(estimator=SVC(), min_features_to_select=28)
LGBMClassifier	0.952971	0.937595	0.952971	0.93479	SparsePCA(n_components=28)
VotingClassifier	0.952743	0.937441	0.952743	0.932995	RFE(estimator=SVC(), n_features_to_select=28)
VotingClassifier	0.952743	0.937441	0.952743	0.932995	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.9528	0.937373	0.9528	0.933502	RFE(estimator=LogisticRegression(), n_features_to_select=28)
VotingClassifier	0.952743	0.937332	0.952743	0.93309	SelectKBest(mutual_info_classif, k=28)
RidgeClassifier	0.9524	0.937237	0.9524	0.930909	SelectKBest(f_classif, k=28)
RidgeClassifierCV	0.9524	0.937237	0.9524	0.930909	SelectKBest(f_classif, k=28)
RidgeClassifier	0.9524	0.937237	0.9524	0.930909	SelectKBest(mutual_info_classif, k=28)
RidgeClassifierCV	0.9524	0.937237	0.9524	0.930909	SelectKBest(mutual_info_classif, k=28)
RidgeClassifier	0.9524	0.937237	0.9524	0.930909	SelectKBest(f_regression, k=28)
RidgeClassifierCV	0.9524	0.937237	0.9524	0.930909	SelectKBest(f_regression, k=28)

RidgeClassifier	0.952 4	0.937 237	0.952 4	0.930 909	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.952 4	0.937 237	0.952 4	0.930 909	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
RidgeClassifier	0.952 4	0.937 237	0.952 4	0.930 909	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.952 4	0.937 237	0.952 4	0.930 909	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
RidgeClassifier	0.952 4	0.937 237	0.952 4	0.930 909	RFE(estimator=LogisticRegression(), n_features_to_select=28)
RidgeClassifierCV	0.952 4	0.937 237	0.952 4	0.930 909	RFE(estimator=LogisticRegression(), n_features_to_select=28)
RidgeClassifier	0.952 4	0.937 237	0.952 4	0.930 909	RFE(estimator=SVC(), n_features_to_select=28)
RidgeClassifierCV	0.952 4	0.937 237	0.952 4	0.930 909	RFE(estimator=SVC(), n_features_to_select=28)
RidgeClassifier	0.952 4	0.937 237	0.952 4	0.930 909	RFE(estimator=MLPClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.952 4	0.937 237	0.952 4	0.930 909	RFE(estimator=MLPClassifier(), n_features_to_select=28)
RidgeClassifier	0.952 4	0.937 237	0.952 4	0.930 909	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.952 4	0.937 237	0.952 4	0.930 909	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
RidgeClassifier	0.952 4	0.937 237	0.952 4	0.930 909	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.952 4	0.937 237	0.952 4	0.930 909	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
RidgeClassifier	0.952 4	0.937 237	0.952 4	0.930 909	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
RidgeClassifierCV	0.952 4	0.937 237	0.952 4	0.930 909	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
RidgeClassifier	0.952 4	0.937 237	0.952 4	0.930 909	RFECV(estimator=SVC(), min_features_to_select=28)
RidgeClassifierCV	0.952 4	0.937 237	0.952 4	0.930 909	RFECV(estimator=SVC(), min_features_to_select=28)
RidgeClassifier	0.952 4	0.937 237	0.952 4	0.930 909	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.952 4	0.937 237	0.952 4	0.930 909	RFECV(estimator=MLPClassifier(), min_features_to_select=28)

RidgeClassifier	0.952 4	0.937 237	0.952 4	0.930 909	PCA(n_components=28)
RidgeClassifierCV	0.952 4	0.937 237	0.952 4	0.930 909	PCA(n_components=28)
RidgeClassifier	0.952 4	0.937 237	0.952 4	0.930 909	SparsePCA(n_components=28)
RidgeClassifierCV	0.952 4	0.937 237	0.952 4	0.930 909	SparsePCA(n_components=28)
HistGradientBoostingClassifier	0.952 743	0.937 14	0.952 743	0.933 28	SelectKBest(f_classif, k=28)
GradientBoostingClassifier	0.952 514	0.937 042	0.952 514	0.931 687	SelectKBest(f_classif, k=28)
GradientBoostingClassifier	0.952 514	0.937 042	0.952 514	0.931 687	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.952 514	0.937 042	0.952 514	0.931 687	RFE(estimator=LogisticRegression(), n_features_to_select=28)
GradientBoostingClassifier	0.952 514	0.937 042	0.952 514	0.931 687	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
GradientBoostingClassifier	0.952 514	0.937 042	0.952 514	0.931 687	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
GradientBoostingClassifier	0.952 514	0.937 042	0.952 514	0.931 687	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
GradientBoostingClassifier	0.952 514	0.937 042	0.952 514	0.931 687	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
GradientBoostingClassifier	0.952 514	0.937 042	0.952 514	0.931 687	MiniBatchSparsePCA(n_components=28)
VotingClassifier	0.952 686	0.936 988	0.952 686	0.932 962	SelectKBest(f_regression, k=28)
VotingClassifier	0.952 686	0.936 988	0.952 686	0.932 962	RFE(estimator=LogisticRegression(), n_features_to_select=28)
VotingClassifier	0.952 686	0.936 988	0.952 686	0.932 962	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LGBMClassifier	0.952 8	0.936 843	0.952 8	0.934 328	MiniBatchSparsePCA(n_components=28)
HistGradientBoostingClassifier	0.952 686	0.936 809	0.952 686	0.933 152	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
ExtraTreeClassifier	0.940 743	0.936 796	0.940 743	0.938 67	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.952 8	0.936 776	0.952 8	0.934 507	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
HistGradientBoostingClassifier	0.952 686	0.936 731	0.952 686	0.933 246	RFE(estimator=MLPClassifier(), n_features_to_select=28)

VotingClassifier	0.952 629	0.936 636	0.952 629	0.932 833	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
VotingClassifier	0.952 629	0.936 549	0.952 629	0.932 928	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
VotingClassifier	0.952 629	0.936 549	0.952 629	0.932 928	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
ExtraTreeClassifier	0.941 543	0.936 463	0.941 543	0.938 832	RFE(estimator=LogisticRegression(), n_features_to_select=28)
GradientBoostingClassifier	0.952 457	0.936 384	0.952 457	0.931 655	RFE(estimator=MLPClassifier(), n_features_to_select=28)
HistGradientBoostingClassifier	0.952 571	0.936 362	0.952 571	0.932 607	RFECV(estimator=SVC(), min_features_to_select=28)
DecisionTreeClassifier	0.940 057	0.936 344	0.940 057	0.938 117	MiniBatchSparsePCA(n_components=28)
ExtraTreeClassifier	0.941 714	0.936 333	0.941 714	0.938 83	RFECV(estimator=SVC(), min_features_to_select=28)
ExtraTreeClassifier	0.941 886	0.936 204	0.941 886	0.938 828	SelectKBest(f_classif, k=28)
VotingClassifier	0.952 571	0.936 194	0.952 571	0.932 8	RFECV(estimator=SVC(), min_features_to_select=28)
VotingClassifier	0.952 571	0.936 123	0.952 571	0.932 895	SelectKBest(f_classif, k=28)
VotingClassifier	0.952 571	0.936 123	0.952 571	0.932 895	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
VotingClassifier	0.952 571	0.936 123	0.952 571	0.932 895	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
ExtraTreeClassifier	0.940 286	0.936 089	0.940 286	0.938 079	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
ExtraTreeClassifier	0.939 371	0.935 972	0.939 371	0.937 604	SparsePCA(n_components=28)
ExtraTreeClassifier	0.941 543	0.935 964	0.941 543	0.938 549	PCA(n_components=28)
ExtraTreeClassifier	0.941 429	0.935 836	0.941 429	0.938 429	SelectKBest(f_regression, k=28)
ExtraTreeClassifier	0.941 2	0.935 798	0.941 2	0.938 311	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
VotingClassifier	0.952 514	0.935 766	0.952 514	0.932 767	MiniBatchSparsePCA(n_components=28)
ExtraTreeClassifier	0.941 257	0.935 753	0.941 257	0.938 31	RFE(estimator=SVC(), n_features_to_select=28)
DecisionTreeClassifier	0.938 514	0.935 753	0.938 514	0.937 091	RFECV(estimator=SVC(), min_features_to_select=28)
GradientBoostingClassifier	0.952 4	0.935 63	0.952 4	0.931 723	PCA(n_components=28)
DecisionTreeClassifier	0.938 343	0.935 609	0.938 343	0.936 935	SelectKBest(f_regression, k=28)



VotingClassifier	0.952 514	0.935 504	0.952 514	0.933 333	SparsePCA(n_components=28)
DecisionTreeClassifier	0.938 229	0.935 417	0.938 229	0.936 78	SelectKBest(f_classif, k=28)
DecisionTreeClassifier	0.939 029	0.935 39	0.939 029	0.937 134	PCA(n_components=28)
DecisionTreeClassifier	0.937 486	0.935 334	0.937 486	0.936 386	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
ExtraTreeClassifier	0.939 2	0.935 246	0.939 2	0.937 133	SelectKBest(mutual_info_classif, k=28)
ExtraTreeClassifier	0.938 514	0.935 244	0.938 514	0.936 82	RFE(estimator=MLPClassifier(), n_features_to_select=28)
ExtraTreeClassifier	0.940 4	0.935 202	0.940 4	0.937 636	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.938 286	0.935 149	0.938 286	0.936 663	RFE(estimator=LogisticRegression(), n_features_to_select=28)
DecisionTreeClassifier	0.938 229	0.935 125	0.938 229	0.936 624	RFE(estimator=MLPClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.937 429	0.935 092	0.937 429	0.936 232	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.937 714	0.935 061	0.937 714	0.936 35	SparsePCA(n_components=28)
DecisionTreeClassifier	0.937 371	0.934 996	0.937 371	0.936 154	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.937 314	0.934 974	0.937 314	0.936 115	RFE(estimator=SVC(), n_features_to_select=28)
MLPClassifier	0.952 4	0.934 971	0.952 4	0.934 877	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
VotingClassifier	0.952 4	0.934 947	0.952 4	0.932 7	PCA(n_components=28)
ExtraTreeClassifier	0.94	0.934 944	0.94	0.937 32	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.937 657	0.934 891	0.937 657	0.936 234	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.937 143	0.934 758	0.937 143	0.935 921	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LabelSpreading	0.952 286	0.934 56	0.952 286	0.934 892	MiniBatchSparsePCA(n_components=28)
MLPClassifier	0.951 714	0.934 552	0.951 714	0.937 23	SparsePCA(n_components=28)
MLPClassifier	0.952 343	0.934 521	0.952 343	0.932 857	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.937 6	0.934 425	0.937 6	0.935 96	SelectKBest(mutual_info_classif, k=28)

LabelPropagation	0.952 171	0.934 41	0.952 171	0.935 41	MiniBatchSparsePCA(n_comp onents=28)
DecisionTreeClassifi er	0.937 086	0.934 366	0.937 086	0.935 688	RFE(estimator=RandomForest Classifier(), n_features_to_select=28)
MLPClassifier	0.952 229	0.934 22	0.952 229	0.934 512	RFECV(estimator=SVC(), min_features_to_select=28)
ExtraTreeClassifier	0.938 114	0.934 192	0.938 114	0.936 071	MiniBatchSparsePCA(n_comp onents=28)
MLPClassifier	0.952 229	0.934 13	0.952 229	0.934 25	SelectKBest(f_classif, k=28)
ExtraTreeClassifier	0.938 286	0.933 966	0.938 286	0.936 026	RFECV(estimator=LogisticRe gression(), min_features_to_select=28)
MLPClassifier	0.952 057	0.933 868	0.952 057	0.935 005	RFECV(estimator=LogisticRe gression(), min_features_to_select=28)
MLPClassifier	0.952 114	0.933 74	0.952 114	0.934 355	RFECV(estimator=MLPClassi fier(), min_features_to_select=28)
MLPClassifier	0.952 057	0.933 649	0.952 057	0.934 58	RFE(estimator=DecisionTreeC lassifier(), n_features_to_select=28)
GradientBoostingCla ssifier	0.952 171	0.933 463	0.952 171	0.931 595	SparsePCA(n_components=28 )
KNeighborsClassifie r	0.950 629	0.933 442	0.950 629	0.937 515	SparsePCA(n_components=28 )
SelfTrainingClassifie r	0.950 629	0.933 442	0.950 629	0.937 515	SparsePCA(n_components=28 )
LabelSpreading	0.951 829	0.933 405	0.951 829	0.935 279	SparsePCA(n_components=28 )
MLPClassifier	0.952	0.933 352	0.952	0.934 373	SelectKBest(f_regression, k=28)
MLPClassifier	0.952	0.933 26	0.952	0.934 199	SelectKBest(mutual_info_clas sif, k=28)
KNeighborsClassifie r	0.951 086	0.933 152	0.951 086	0.936 673	MiniBatchSparsePCA(n_comp onents=28)
SelfTrainingClassifie r	0.951 086	0.933 152	0.951 086	0.936 673	MiniBatchSparsePCA(n_comp onents=28)
KNeighborsClassifie r	0.950 743	0.933 146	0.950 743	0.937 1	RFE(estimator=RandomForest Classifier(), n_features_to_select=28)
SelfTrainingClassifie r	0.950 743	0.933 146	0.950 743	0.937 1	RFE(estimator=RandomForest Classifier(), n_features_to_select=28)
KNeighborsClassifie r	0.950 743	0.933 146	0.950 743	0.937 1	RFE(estimator=SVC(), n_features_to_select=28)
SelfTrainingClassifie r	0.950 743	0.933 146	0.950 743	0.937 1	RFE(estimator=SVC(), n_features_to_select=28)
KNeighborsClassifie r	0.950 743	0.933 146	0.950 743	0.937 1	RFECV(estimator=DecisionTr eeClassifier(), min_features_to_select=28)

SelfTrainingClassifier	0.950 743	0.933 146	0.950 743	0.937 1	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.950 743	0.933 146	0.950 743	0.937 1	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
SelfTrainingClassifier	0.950 743	0.933 146	0.950 743	0.937 1	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
KNeighborsClassifier	0.950 743	0.933 146	0.950 743	0.937 1	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.950 743	0.933 146	0.950 743	0.937 1	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.950 743	0.933 146	0.950 743	0.937 1	PCA(n_components=28)
SelfTrainingClassifier	0.950 743	0.933 146	0.950 743	0.937 1	PCA(n_components=28)
MLPClassifier	0.951 829	0.933 13	0.951 829	0.934 864	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LabelSpreading	0.951 6	0.933 066	0.951 6	0.935 541	SelectKBest(f_classif, k=28)
KNeighborsClassifier	0.950 629	0.932 952	0.950 629	0.937 025	SelectKBest(mutual_info_classif, k=28)
SelfTrainingClassifier	0.950 629	0.932 952	0.950 629	0.937 025	SelectKBest(mutual_info_classif, k=28)
BernoulliNB	0.771 714	0.932 946	0.771 714	0.834 776	SelectKBest(f_classif, k=28)
BernoulliNB	0.771 714	0.932 946	0.771 714	0.834 776	SelectKBest(mutual_info_classif, k=28)
BernoulliNB	0.771 714	0.932 946	0.771 714	0.834 776	SelectKBest(f_regression, k=28)
BernoulliNB	0.771 714	0.932 946	0.771 714	0.834 776	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
BernoulliNB	0.771 714	0.932 946	0.771 714	0.834 776	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
BernoulliNB	0.771 714	0.932 946	0.771 714	0.834 776	RFE(estimator=LogisticRegression(), n_features_to_select=28)
BernoulliNB	0.771 714	0.932 946	0.771 714	0.834 776	RFE(estimator=SVC(), n_features_to_select=28)
BernoulliNB	0.771 714	0.932 946	0.771 714	0.834 776	RFE(estimator=MLPClassifier(), n_features_to_select=28)
BernoulliNB	0.771 714	0.932 946	0.771 714	0.834 776	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
BernoulliNB	0.771 714	0.932 946	0.771 714	0.834 776	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)

BernoulliNB	0.771 714	0.932 946	0.771 714	0.834 776	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
BernoulliNB	0.771 714	0.932 946	0.771 714	0.834 776	RFECV(estimator=SVC(), min_features_to_select=28)
BernoulliNB	0.771 714	0.932 946	0.771 714	0.834 776	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LabelPropagation	0.951 543	0.932 919	0.951 543	0.935 505	SparsePCA(n_components=28)
KNeighborsClassifier	0.950 571	0.932 857	0.950 571	0.936 988	SelectKBest(f_regression, k=28)
SelfTrainingClassifier	0.950 571	0.932 857	0.950 571	0.936 988	SelectKBest(f_regression, k=28)
LabelSpreading	0.951 486	0.932 773	0.951 486	0.935 469	SelectKBest(f_regression, k=28)
LabelSpreading	0.951 486	0.932 773	0.951 486	0.935 469	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LabelSpreading	0.951 486	0.932 773	0.951 486	0.935 469	RFE(estimator=SVC(), n_features_to_select=28)
LabelSpreading	0.951 486	0.932 773	0.951 486	0.935 469	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LabelSpreading	0.951 486	0.932 773	0.951 486	0.935 469	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LabelSpreading	0.951 486	0.932 773	0.951 486	0.935 469	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LabelSpreading	0.951 486	0.932 773	0.951 486	0.935 469	PCA(n_components=28)
LabelPropagation	0.951 429	0.932 695	0.951 429	0.935 513	SelectKBest(f_classif, k=28)
LabelSpreading	0.951 257	0.932 488	0.951 257	0.935 642	SelectKBest(mutual_info_classif, k=28)
LabelSpreading	0.951 257	0.932 488	0.951 257	0.935 642	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.952 057	0.932 485	0.952 057	0.931 432	SelectKBest(f_classif, k=28)
AdaBoostClassifier	0.952 057	0.932 485	0.952 057	0.931 432	SelectKBest(mutual_info_classif, k=28)
AdaBoostClassifier	0.952 057	0.932 485	0.952 057	0.931 432	SelectKBest(f_regression, k=28)
AdaBoostClassifier	0.952 057	0.932 485	0.952 057	0.931 432	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.952 057	0.932 485	0.952 057	0.931 432	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.952 057	0.932 485	0.952 057	0.931 432	RFE(estimator=LogisticRegression(), n_features_to_select=28)

AdaBoostClassifier	0.952 057	0.932 485	0.952 057	0.931 432	RFE(estimator=SVC(), n_features_to_select=28)
AdaBoostClassifier	0.952 057	0.932 485	0.952 057	0.931 432	RFE(estimator=MLPClassifier (), n_features_to_select=28)
AdaBoostClassifier	0.952 057	0.932 485	0.952 057	0.931 432	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.952 057	0.932 485	0.952 057	0.931 432	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.952 057	0.932 485	0.952 057	0.931 432	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
AdaBoostClassifier	0.952 057	0.932 485	0.952 057	0.931 432	RFECV(estimator=SVC(), min_features_to_select=28)
AdaBoostClassifier	0.952 057	0.932 485	0.952 057	0.931 432	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.950 343	0.932 482	0.950 343	0.936 838	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
SelfTrainingClassifier	0.950 343	0.932 482	0.950 343	0.936 838	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.950 343	0.932 482	0.950 343	0.936 838	RFE(estimator=LogisticRegression(), n_features_to_select=28)
SelfTrainingClassifier	0.950 343	0.932 482	0.950 343	0.936 838	RFE(estimator=LogisticRegression(), n_features_to_select=28)
KNeighborsClassifier	0.950 343	0.932 482	0.950 343	0.936 838	RFE(estimator=MLPClassifier (), n_features_to_select=28)
SelfTrainingClassifier	0.950 343	0.932 482	0.950 343	0.936 838	RFE(estimator=MLPClassifier (), n_features_to_select=28)
KNeighborsClassifier	0.950 343	0.932 482	0.950 343	0.936 838	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.950 343	0.932 482	0.950 343	0.936 838	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.950 343	0.932 482	0.950 343	0.936 838	RFECV(estimator=SVC(), min_features_to_select=28)
SelfTrainingClassifier	0.950 343	0.932 482	0.950 343	0.936 838	RFECV(estimator=SVC(), min_features_to_select=28)
LabelPropagation	0.951 314	0.932 415	0.951 314	0.935 441	SelectKBest(f_regression, k=28)
LabelPropagation	0.951 314	0.932 415	0.951 314	0.935 441	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LabelPropagation	0.951 314	0.932 415	0.951 314	0.935 441	RFE(estimator=SVC(), n_features_to_select=28)
LabelPropagation	0.951 314	0.932 415	0.951 314	0.935 441	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)

LabelPropagation	0.951 314	0.932 415	0.951 314	0.935 441	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LabelPropagation	0.951 314	0.932 415	0.951 314	0.935 441	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LabelPropagation	0.951 314	0.932 415	0.951 314	0.935 441	PCA(n_components=28)
LabelPropagation	0.951 086	0.932 171	0.951 086	0.935 611	SelectKBest(mutual_info_classifier, k=28)
LabelPropagation	0.951 086	0.932 171	0.951 086	0.935 611	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
MLPClassifier	0.951 429	0.932 009	0.951 429	0.934 7	MiniBatchSparsePCA(n_components=28)
KNeighborsClassifier	0.949 943	0.931 934	0.949 943	0.936 647	SelectKBest(f_classifier, k=28)
SelfTrainingClassifier	0.949 943	0.931 934	0.949 943	0.936 647	SelectKBest(f_classifier, k=28)
MLPClassifier	0.951 543	0.931 49	0.951 543	0.933 748	PCA(n_components=28)
MLPClassifier	0.951 6	0.931 321	0.951 6	0.933 34	RFE(estimator=SVC(), n_features_to_select=28)
LabelSpreading	0.950 743	0.931 109	0.950 743	0.935 081	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LabelSpreading	0.950 743	0.931 109	0.950 743	0.935 081	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LabelSpreading	0.950 743	0.931 109	0.950 743	0.935 081	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LabelSpreading	0.950 743	0.931 109	0.950 743	0.935 081	RFECV(estimator=SVC(), min_features_to_select=28)
LabelPropagation	0.950 571	0.930 832	0.950 571	0.935 052	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LabelPropagation	0.950 571	0.930 832	0.950 571	0.935 052	RFE(estimator=MLPClassifier(), n_features_to_select=28)
MLPClassifier	0.950 571	0.930 832	0.950 571	0.935 052	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LabelPropagation	0.950 571	0.930 832	0.950 571	0.935 052	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LabelPropagation	0.950 571	0.930 832	0.950 571	0.935 052	RFECV(estimator=SVC(), min_features_to_select=28)
PassiveAggressiveClassifier	0.858 743	0.930 025	0.858 743	0.889 042	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LogisticRegression	0.951 657	0.929 623	0.951 657	0.931 308	MiniBatchSparsePCA(n_components=28)
MLPClassifier	0.951 2	0.929 595	0.951 2	0.932 834	RFE(estimator=MLPClassifier(), n_features_to_select=28)

SGDClassifier	0.951 6	0.928 57	0.951 6	0.930 778	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LogisticRegression	0.951 486	0.928 435	0.951 486	0.931 114	SparsePCA(n_components=28)
LogisticRegression	0.951 429	0.928 255	0.951 429	0.931 18	SelectKBest(f_classif, k=28)
LogisticRegression	0.951 429	0.928 255	0.951 429	0.931 18	SelectKBest(mutual_info_classifier, k=28)
LogisticRegression	0.951 429	0.928 255	0.951 429	0.931 18	SelectKBest(f_regression, k=28)
LogisticRegression	0.951 429	0.928 255	0.951 429	0.931 18	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LogisticRegression	0.951 429	0.928 255	0.951 429	0.931 18	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LogisticRegression	0.951 429	0.928 255	0.951 429	0.931 18	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LogisticRegression	0.951 429	0.928 255	0.951 429	0.931 18	RFE(estimator=SVC(), n_features_to_select=28)
LogisticRegression	0.951 429	0.928 255	0.951 429	0.931 18	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LogisticRegression	0.951 429	0.928 255	0.951 429	0.931 18	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LogisticRegression	0.951 429	0.928 255	0.951 429	0.931 18	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LogisticRegression	0.951 429	0.928 255	0.951 429	0.931 18	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LogisticRegression	0.951 429	0.928 255	0.951 429	0.931 18	RFECV(estimator=SVC(), min_features_to_select=28)
LogisticRegression	0.951 429	0.928 255	0.951 429	0.931 18	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LogisticRegression	0.951 429	0.928 255	0.951 429	0.931 18	PCA(n_components=28)
SGDClassifier	0.950 971	0.927 946	0.950 971	0.932 063	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
SGDClassifier	0.951 829	0.927 942	0.951 829	0.929 333	RFE(estimator=LogisticRegression(), n_features_to_select=28)
AdaBoostClassifier	0.951 429	0.927 768	0.951 429	0.930 884	SparsePCA(n_components=28)
PassiveAggressiveClassifier	0.951 714	0.927 65	0.951 714	0.929 808	PCA(n_components=28)
SGDClassifier	0.951 486	0.926 721	0.951 486	0.930 207	SelectKBest(f_regression, k=28)
LinearDiscriminantAnalysis	0.947 257	0.926 54	0.947 257	0.933 577	SelectKBest(f_classif, k=28)

LinearDiscriminantAnalysis	0.947 257	0.926 54	0.947 257	0.933 577	SelectKBest(mutual_info_classifier, k=28)
LinearDiscriminantAnalysis	0.947 257	0.926 54	0.947 257	0.933 577	SelectKBest(f_regression, k=28)
LinearDiscriminantAnalysis	0.947 257	0.926 54	0.947 257	0.933 577	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.947 257	0.926 54	0.947 257	0.933 577	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.947 257	0.926 54	0.947 257	0.933 577	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.947 257	0.926 54	0.947 257	0.933 577	RFE(estimator=SVC(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.947 257	0.926 54	0.947 257	0.933 577	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.947 257	0.926 54	0.947 257	0.933 577	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.947 257	0.926 54	0.947 257	0.933 577	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.947 257	0.926 54	0.947 257	0.933 577	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.947 257	0.926 54	0.947 257	0.933 577	RFECV(estimator=SVC(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.947 257	0.926 54	0.947 257	0.933 577	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.947 257	0.926 54	0.947 257	0.933 577	PCA(n_components=28)
LinearDiscriminantAnalysis	0.947 257	0.926 54	0.947 257	0.933 577	SparsePCA(n_components=28)
AdaBoostClassifier	0.951 2	0.926 253	0.951 2	0.930 658	PCA(n_components=28)
PassiveAggressiveClassifier	0.886 571	0.926 181	0.886 571	0.904 347	RFE(estimator=LogisticRegression(), n_features_to_select=28)
SGDClassifier	0.949 6	0.925 9	0.949 6	0.932 327	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.948 743	0.925 546	0.948 743	0.932 568	MiniBatchSparsePCA(n_components=28)
PassiveAggressiveClassifier	0.913 029	0.925 521	0.913 029	0.918 972	SelectKBest(mutual_info_classifier, k=28)
SGDClassifier	0.949 657	0.925 274	0.949 657	0.931 928	PCA(n_components=28)
GaussianNB	0.941 886	0.924 92	0.941 886	0.932 001	SelectKBest(f_classifier, k=28)
GaussianNB	0.941 886	0.924 92	0.941 886	0.932 001	SelectKBest(mutual_info_classifier, k=28)



GaussianNB	0.941 886	0.924 92	0.941 886	0.932 001	SelectKBest(f_regression, k=28)
GaussianNB	0.941 886	0.924 92	0.941 886	0.932 001	RFE(estimator=DecisionTreeC lassifier(), n_features_to_select=28)
GaussianNB	0.941 886	0.924 92	0.941 886	0.932 001	RFE(estimator=RandomForest Classifier(), n_features_to_select=28)
GaussianNB	0.941 886	0.924 92	0.941 886	0.932 001	RFE(estimator=LogisticRegres sion(), n_features_to_select=28)
GaussianNB	0.941 886	0.924 92	0.941 886	0.932 001	RFE(estimator=SVC(), n_features_to_select=28)
GaussianNB	0.941 886	0.924 92	0.941 886	0.932 001	RFE(estimator=MLPClassifier ( ), n_features_to_select=28)
GaussianNB	0.941 886	0.924 92	0.941 886	0.932 001	RFECV(estimator=DecisionTr eeClassifier(), min_features_to_select=28)
GaussianNB	0.941 886	0.924 92	0.941 886	0.932 001	RFECV(estimator=RandomFo restClassifier(), min_features_to_select=28)
GaussianNB	0.941 886	0.924 92	0.941 886	0.932 001	RFECV(estimator=LogisticRe gression(), min_features_to_select=28)
GaussianNB	0.941 886	0.924 92	0.941 886	0.932 001	RFECV(estimator=SVC(), min_features_to_select=28)
GaussianNB	0.941 886	0.924 92	0.941 886	0.932 001	RFECV(estimator=MLPClassi fier(), min_features_to_select=28)
PassiveAggressiveCl assifier	0.917 371	0.924 678	0.917 371	0.920 914	RFE(estimator=SVC(), n_features_to_select=28)
BernoulliNB	0.783 029	0.924 646	0.783 029	0.841 178	PCA(n_components=28)
PassiveAggressiveCl assifier	0.948 514	0.924 464	0.948 514	0.932 02	RFECV(estimator=DecisionTr eeClassifier(), min_features_to_select=28)
QuadraticDiscrimina ntAnalysis	0.925 657	0.924 425	0.925 657	0.925 037	SparsePCA(n_components=28 )
PassiveAggressiveCl assifier	0.950 571	0.924 335	0.950 571	0.930 702	RFECV(estimator=MLPClassi fier(), min_features_to_select=28)
QuadraticDiscrimina ntAnalysis	0.926 457	0.924 294	0.926 457	0.925 364	RFECV(estimator=RandomFo restClassifier(), min_features_to_select=28)
QuadraticDiscrimina ntAnalysis	0.926 4	0.924 279	0.926 4	0.925 328	SelectKBest(f_regression, k=28)
QuadraticDiscrimina ntAnalysis	0.926 457	0.924 207	0.926 457	0.925 319	RFECV(estimator=SVC(), min_features_to_select=28)
QuadraticDiscrimina ntAnalysis	0.926 4	0.924 105	0.926 4	0.925 239	RFECV(estimator=DecisionTr eeClassifier(), min_features_to_select=28)
QuadraticDiscrimina ntAnalysis	0.927 029	0.924 015	0.927 029	0.925 498	PCA(n_components=28)

QuadraticDiscriminantAnalysis	0.926686	0.924008	0.926686	0.925328	RFE(estimator=MLPClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.9256	0.923978	0.9256	0.924783	MiniBatchSparsePCA(n_components=28)
PassiveAggressiveClassifier	0.926686	0.92392	0.926686	0.925284	RFE(estimator=MLPClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.925657	0.923906	0.925657	0.924774	SelectKBest(mutual_info_classif, k=28)
QuadraticDiscriminantAnalysis	0.926286	0.923899	0.926286	0.925078	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926229	0.923884	0.926229	0.925042	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926229	0.923884	0.926229	0.925042	RFE(estimator=LogisticRegression(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.925886	0.923792	0.925886	0.924828	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
BernoulliNB	0.782057	0.923763	0.782057	0.840442	SparsePCA(n_components=28)
GaussianNB	0.926	0.923735	0.926	0.924855	SparsePCA(n_components=28)
QuadraticDiscriminantAnalysis	0.925314	0.92373	0.925314	0.924516	SelectKBest(f_classif, k=28)
GaussianNB	0.927543	0.923718	0.927543	0.925592	PCA(n_components=28)
SGDClassifier	0.950971	0.923689	0.950971	0.930032	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.925486	0.923687	0.925486	0.924578	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.925543	0.923615	0.925543	0.92457	RFE(estimator=SVC(), n_features_to_select=28)
PassiveAggressiveClassifier	0.937371	0.923433	0.937371	0.929697	SelectKBest(f_regression, k=28)
SGDClassifier	0.948914	0.923409	0.948914	0.93132	RFECV(estimator=SVC(), min_features_to_select=28)
GaussianNB	0.934171	0.92329	0.934171	0.928355	MiniBatchSparsePCA(n_components=28)
BernoulliNB	0.8412	0.923129	0.8412	0.876797	MiniBatchSparsePCA(n_components=28)
AdaBoostClassifier	0.951486	0.92181	0.951486	0.929047	MiniBatchSparsePCA(n_components=28)
SGDClassifier	0.951543	0.921648	0.951543	0.928968	SelectKBest(f_classif, k=28)
SGDClassifier	0.938057	0.921436	0.938057	0.928823	RFE(estimator=MLPClassifier(), n_features_to_select=28)
SGDClassifier	0.949086	0.921368	0.949086	0.930342	SelectKBest(mutual_info_classif, k=28)
PassiveAggressiveClassifier	0.901714	0.921118	0.901714	0.910906	RFECV(estimator=SVC(), min_features_to_select=28)

SGDClassifier	0.9508	0.919841	0.9508	0.929221	RFE(estimator=SVC(), n_features_to_select=28)
PassiveAggressiveClassifier	0.9468	0.919834	0.9468	0.929927	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
SGDClassifier	0.950114	0.919329	0.950114	0.929368	SparsePCA(n_components=28)
PassiveAggressiveClassifier	0.946686	0.919312	0.946686	0.929691	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
PassiveAggressiveClassifier	0.838743	0.919252	0.838743	0.874354	MiniBatchSparsePCA(n_components=28)
PassiveAggressiveClassifier	0.943086	0.919116	0.943086	0.929053	SparsePCA(n_components=28)
PassiveAggressiveClassifier	0.941543	0.91736	0.941543	0.927779	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SGDClassifier	0.949714	0.915302	0.949714	0.928547	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SGDClassifier	0.950686	0.914914	0.950686	0.928528	MiniBatchSparsePCA(n_components=28)
SGDClassifier	0.951371	0.906169	0.951371	0.92822	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
PassiveAggressiveClassifier	0.447314	0.876547	0.447314	0.585029	SelectKBest(f_classif, k=28)

**APPENDIX 5: VULNERABILITY FUNCTION LEVEL DATA RESULTS IN 2002 TO 2019**

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
RandomForestClassifier	0.950743	0.936812	0.950743	0.93886	SelectKBest(mutual_info_classifier, k=28)
RandomForestClassifier	0.950696	0.936706	0.950696	0.938778	SelectKBest(f_regression, k=28)
RandomForestClassifier	0.950975	0.937009	0.950975	0.938623	PCA(n_components=28)
RandomForestClassifier	0.950556	0.936413	0.950556	0.938579	RFE(estimator=SVC(), n_features_to_select=28)
BaggingClassifier	0.949718	0.935522	0.949718	0.938573	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
RandomForestClassifier	0.950649	0.936513	0.950649	0.938545	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
RandomForestClassifier	0.950463	0.936258	0.950463	0.938514	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
RandomForestClassifier	0.950743	0.936539	0.950743	0.938357	MiniBatchSparsePCA(n_components=28)
RandomForestClassifier	0.950417	0.936089	0.950417	0.938331	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
RandomForestClassifier	0.950324	0.935966	0.950324	0.938315	SelectKBest(f_classifier, k=28)
RandomForestClassifier	0.95051	0.936188	0.95051	0.938295	RFE(estimator=MLPClassifier(), n_features_to_select=28)
RandomForestClassifier	0.950137	0.935629	0.950137	0.938134	RFECV(estimator=SVC(), min_features_to_select=28)
RandomForestClassifier	0.950277	0.935794	0.950277	0.938132	RFE(estimator=LogisticRegression(), n_features_to_select=28)
RandomForestClassifier	0.950184	0.935672	0.950184	0.938117	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
RandomForestClassifier	0.950044	0.935444	0.950044	0.938019	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
RandomForestClassifier	0.950696	0.936246	0.950696	0.937912	SparsePCA(n_components=28)
StackingClassifier	0.951068	0.936839	0.951068	0.937856	SparsePCA(n_components=28)
RandomForestClassifier	0.949951	0.935226	0.949951	0.937853	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
BaggingClassifier	0.94902	0.934304	0.94902	0.937793	RFE(estimator=LogisticRegression(), n_features_to_select=28)
BaggingClassifier	0.949858	0.935041	0.949858	0.937737	PCA(n_components=28)

StackingClassifier	0.951068	0.936765	0.951068	0.937641	MiniBatchSparsePCA(n_components=28)
BaggingClassifier	0.949299	0.934331	0.949299	0.937547	RFECV(estimator=SVC(), min_features_to_select=28)
StackingClassifier	0.950836	0.936283	0.950836	0.93748	PCA(n_components=28)
StackingClassifier	0.951022	0.936595	0.951022	0.937392	RFECV(estimator=SVC(), min_features_to_select=28)
BaggingClassifier	0.94888	0.933812	0.94888	0.937354	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
BaggingClassifier	0.949532	0.934336	0.949532	0.937256	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
StackingClassifier	0.950929	0.936359	0.950929	0.937219	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
StackingClassifier	0.950743	0.936006	0.950743	0.937199	RFE(estimator=LogisticRegression(), n_features_to_select=28)
BaggingClassifier	0.94902	0.933629	0.94902	0.937002	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
StackingClassifier	0.950743	0.935901	0.950743	0.936925	SelectKBest(f_classif, k=28)
StackingClassifier	0.950789	0.935982	0.950789	0.936902	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
BaggingClassifier	0.948694	0.933242	0.948694	0.936877	RFE(estimator=MLPClassifier(), n_features_to_select=28)
BaggingClassifier	0.948322	0.932912	0.948322	0.936818	RFE(estimator=SVC(), n_features_to_select=28)
BaggingClassifier	0.949392	0.933803	0.949392	0.936746	SparsePCA(n_components=28)
StackingClassifier	0.950882	0.936118	0.950882	0.936743	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
BaggingClassifier	0.948834	0.933224	0.948834	0.936721	MiniBatchSparsePCA(n_components=28)
BaggingClassifier	0.94888	0.933206	0.94888	0.936651	SelectKBest(f_classif, k=28)
StackingClassifier	0.95037	0.935071	0.95037	0.936559	SelectKBest(f_regression, k=28)
StackingClassifier	0.950603	0.935495	0.950603	0.936552	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
StackingClassifier	0.95051	0.935316	0.95051	0.936544	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
StackingClassifier	0.95037	0.934995	0.95037	0.936393	SelectKBest(mutual_info_classif, k=28)
BaggingClassifier	0.948229	0.932407	0.948229	0.936355	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)

StackingClassifier	0.950 463	0.935 147	0.950 463	0.936 344	RFE(estimator=SVC(), n_features_to_select=28)
BaggingClassifier	0.947 716	0.932 025	0.947 716	0.936 301	RFECV(estimator=RandomFo restClassifier(), min_features_to_select=28)
BaggingClassifier	0.947 903	0.932 066	0.947 903	0.936 231	SelectKBest(f_regression, k=28)
StackingClassifier	0.950 417	0.934 999	0.950 417	0.936 2	RFE(estimator=MLPClassifier ( ), n_features_to_select=28)
StackingClassifier	0.950 184	0.934 526	0.950 184	0.936 098	RFECV(estimator=RandomFo restClassifier(), min_features_to_select=28)
BaggingClassifier	0.948 042	0.931 907	0.948 042	0.935 973	SelectKBest(mutual_info_clas sif, k=28)
XGBClassifier	0.949 672	0.932 097	0.949 672	0.933 126	SparsePCA(n_components=28 )
XGBClassifier	0.949 718	0.932 175	0.949 718	0.933 028	PCA(n_components=28)
XGBClassifier	0.949 765	0.932 104	0.949 765	0.932 472	SelectKBest(f_classif, k=28)
XGBClassifier	0.949 765	0.932 104	0.949 765	0.932 472	SelectKBest(mutual_info_clas sif, k=28)
XGBClassifier	0.949 765	0.932 104	0.949 765	0.932 472	SelectKBest(f_regression, k=28)
XGBClassifier	0.949 765	0.932 104	0.949 765	0.932 472	RFE(estimator=DecisionTreeC lassifier(), n_features_to_select=28)
XGBClassifier	0.949 765	0.932 104	0.949 765	0.932 472	RFE(estimator=RandomForest Classifier(), n_features_to_select=28)
XGBClassifier	0.949 765	0.932 104	0.949 765	0.932 472	RFE(estimator=LogisticRegres sion(), n_features_to_select=28)
XGBClassifier	0.949 765	0.932 104	0.949 765	0.932 472	RFE(estimator=SVC(), n_features_to_select=28)
XGBClassifier	0.949 765	0.932 104	0.949 765	0.932 472	RFE(estimator=MLPClassifier ( ), n_features_to_select=28)
XGBClassifier	0.949 765	0.932 104	0.949 765	0.932 472	RFECV(estimator=DecisionTr eeClassifier(), min_features_to_select=28)
XGBClassifier	0.949 765	0.932 104	0.949 765	0.932 472	RFECV(estimator=RandomFo restClassifier(), min_features_to_select=28)
XGBClassifier	0.949 765	0.932 104	0.949 765	0.932 472	RFECV(estimator=LogisticRe gression(), min_features_to_select=28)
XGBClassifier	0.949 765	0.932 104	0.949 765	0.932 472	RFECV(estimator=SVC(), min_features_to_select=28)
XGBClassifier	0.949 765	0.932 104	0.949 765	0.932 472	RFECV(estimator=MLPClassi fier(), min_features_to_select=28)
XGBClassifier	0.948 927	0.929 978	0.948 927	0.932 393	MiniBatchSparsePCA(n_comp onents=28)

KNeighborsClassifier	0.948 601	0.929 24	0.948 601	0.932 249	SparsePCA(n_components=28)
SelfTrainingClassifier	0.948 601	0.929 24	0.948 601	0.932 249	SparsePCA(n_components=28)
ExtraTreeClassifier	0.934 261	0.928 801	0.934 261	0.931 385	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LabelPropagation	0.949 718	0.931 637	0.949 718	0.931 288	SparsePCA(n_components=28)
ExtraTreeClassifier	0.934 448	0.928 443	0.934 448	0.931 269	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.947 996	0.927 121	0.947 996	0.931 096	MiniBatchSparsePCA(n_components=28)
SelfTrainingClassifier	0.947 996	0.927 121	0.947 996	0.931 096	MiniBatchSparsePCA(n_components=28)
ExtraTreeClassifier	0.933 889	0.928 267	0.933 889	0.930 927	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
ExtraTreeClassifier	0.933 377	0.928 052	0.933 377	0.930 583	RFE(estimator=MLPClassifier(), n_features_to_select=28)
ExtraTreeClassifier	0.933 144	0.927 83	0.933 144	0.930 358	RFE(estimator=LogisticRegression(), n_features_to_select=28)
ExtraTreeClassifier	0.932 399	0.928 411	0.932 399	0.930 334	SelectKBest(f_regression, k=28)
ExtraTreeClassifier	0.933 004	0.927 645	0.933 004	0.930 195	RFE(estimator=SVC(), n_features_to_select=28)
ExtraTreeClassifier	0.933 144	0.927 448	0.933 144	0.930 149	SelectKBest(mutual_info_classifier, k=28)
KNeighborsClassifier	0.945 761	0.923 561	0.945 761	0.930 147	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SelfTrainingClassifier	0.945 761	0.923 561	0.945 761	0.930 147	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.945 761	0.923 561	0.945 761	0.930 147	RFE(estimator=SVC(), n_features_to_select=28)
SelfTrainingClassifier	0.945 761	0.923 561	0.945 761	0.930 147	RFE(estimator=SVC(), n_features_to_select=28)
KNeighborsClassifier	0.945 761	0.923 561	0.945 761	0.930 147	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.945 761	0.923 561	0.945 761	0.930 147	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.945 761	0.923 561	0.945 761	0.930 147	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
SelfTrainingClassifier	0.945 761	0.923 561	0.945 761	0.930 147	RFECV(estimator=LogisticRegression(), min_features_to_select=28)

KNeighborsClassifier	0.945 761	0.923 561	0.945 761	0.930 147	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.945 761	0.923 561	0.945 761	0.930 147	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.945 854	0.923 613	0.945 854	0.930 143	SelectKBest(f_classif, k=28)
SelfTrainingClassifier	0.945 854	0.923 613	0.945 854	0.930 143	SelectKBest(f_classif, k=28)
KNeighborsClassifier	0.945 854	0.923 613	0.945 854	0.930 143	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.945 854	0.923 613	0.945 854	0.930 143	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LabelSpreading	0.949 346	0.929 848	0.949 346	0.930 137	SparsePCA(n_components=28)
KNeighborsClassifier	0.945 808	0.923 542	0.945 808	0.930 114	RFE(estimator=LogisticRegression(), n_features_to_select=28)
SelfTrainingClassifier	0.945 808	0.923 542	0.945 808	0.930 114	RFE(estimator=LogisticRegression(), n_features_to_select=28)
KNeighborsClassifier	0.945 808	0.923 542	0.945 808	0.930 114	RFE(estimator=MLPClassifier(), n_features_to_select=28)
SelfTrainingClassifier	0.945 808	0.923 542	0.945 808	0.930 114	RFE(estimator=MLPClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.945 808	0.923 542	0.945 808	0.930 114	RFECV(estimator=SVC(), min_features_to_select=28)
SelfTrainingClassifier	0.945 808	0.923 542	0.945 808	0.930 114	RFECV(estimator=SVC(), min_features_to_select=28)
KNeighborsClassifier	0.945 761	0.923 472	0.945 761	0.930 085	SelectKBest(f_regression, k=28)
SelfTrainingClassifier	0.945 761	0.923 472	0.945 761	0.930 085	SelectKBest(f_regression, k=28)
KNeighborsClassifier	0.945 761	0.923 472	0.945 761	0.930 085	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
SelfTrainingClassifier	0.945 761	0.923 472	0.945 761	0.930 085	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.945 621	0.923 354	0.945 621	0.930 061	SelectKBest(mutual_info_classif, k=28)
SelfTrainingClassifier	0.945 621	0.923 354	0.945 621	0.930 061	SelectKBest(mutual_info_classif, k=28)
LGBMClassifier	0.950 417	0.936 409	0.950 417	0.929 957	SparsePCA(n_components=28)
ExtraTreeClassifier	0.933 097	0.927 108	0.933 097	0.929 942	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.944 923	0.922 474	0.944 923	0.929 688	PCA(n_components=28)



SelfTrainingClassifier	0.944 923	0.922 474	0.944 923	0.929 688	PCA(n_components=28)
LabelPropagation	0.948 415	0.926 331	0.948 415	0.929 506	MiniBatchSparsePCA(n_components=28)
LGBMClassifier	0.950 137	0.934 66	0.950 137	0.929 485	PCA(n_components=28)
HistGradientBoostingClassifier	0.949 998	0.933 602	0.949 998	0.929 403	SparsePCA(n_components=28)
ExtraTreeClassifier	0.932 166	0.926 794	0.932 166	0.929 357	PCA(n_components=28)
ExtraTreeClassifier	0.932 539	0.926 425	0.932 539	0.929 321	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.931 84	0.926 924	0.931 84	0.929 28	RFECV(estimator=SVC(), min_features_to_select=28)
ExtraTreeClassifier	0.932 213	0.926 489	0.932 213	0.929 212	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LGBMClassifier	0.949 672	0.931 406	0.949 672	0.929 136	SelectKBest(f_classif, k=28)
LGBMClassifier	0.949 672	0.931 406	0.949 672	0.929 136	SelectKBest(mutual_info_classif, k=28)
LGBMClassifier	0.949 672	0.931 406	0.949 672	0.929 136	SelectKBest(f_regression, k=28)
LGBMClassifier	0.949 672	0.931 406	0.949 672	0.929 136	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LGBMClassifier	0.949 672	0.931 406	0.949 672	0.929 136	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LGBMClassifier	0.949 672	0.931 406	0.949 672	0.929 136	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LGBMClassifier	0.949 672	0.931 406	0.949 672	0.929 136	RFE(estimator=SVC(), n_features_to_select=28)
LGBMClassifier	0.949 672	0.931 406	0.949 672	0.929 136	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LGBMClassifier	0.949 672	0.931 406	0.949 672	0.929 136	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LGBMClassifier	0.949 672	0.931 406	0.949 672	0.929 136	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LGBMClassifier	0.949 672	0.931 406	0.949 672	0.929 136	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LGBMClassifier	0.949 672	0.931 406	0.949 672	0.929 136	RFECV(estimator=SVC(), min_features_to_select=28)
LGBMClassifier	0.949 672	0.931 406	0.949 672	0.929 136	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
ExtraTreeClassifier	0.932 585	0.926 052	0.932 585	0.929 136	RFECV(estimator=SVC(), min_features_to_select=28)

MLPClassifier	0.948 973	0.927 674	0.948 973	0.928 956	SelectKBest(f_classif, k=28)
DecisionTreeClassifier	0.931 468	0.926 521	0.931 468	0.928 893	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
MLPClassifier	0.948 229	0.925 136	0.948 229	0.928 89	SelectKBest(f_regression, k=28)
LabelSpreading	0.948 368	0.925 442	0.948 368	0.928 825	MiniBatchSparsePCA(n_components=28)
LabelPropagation	0.947 111	0.922 841	0.947 111	0.928 8	SelectKBest(f_classif, k=28)
LabelPropagation	0.947 111	0.922 841	0.947 111	0.928 8	SelectKBest(f_regression, k=28)
LabelPropagation	0.947 111	0.922 841	0.947 111	0.928 8	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LabelPropagation	0.947 111	0.922 841	0.947 111	0.928 8	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LabelPropagation	0.947 111	0.922 841	0.947 111	0.928 8	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LabelPropagation	0.947 111	0.922 841	0.947 111	0.928 8	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LabelPropagation	0.947 111	0.922 841	0.947 111	0.928 8	RFECV(estimator=SVC(), min_features_to_select=28)
MLPClassifier	0.948 415	0.925 523	0.948 415	0.928 779	SparsePCA(n_components=28)
MLPClassifier	0.947 996	0.924 383	0.947 996	0.928 753	SelectKBest(mutual_info_classifier, k=28)
DecisionTreeClassifier	0.931 328	0.926 337	0.931 328	0.928 731	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LabelPropagation	0.946 785	0.922 225	0.946 785	0.928 675	SelectKBest(mutual_info_classifier, k=28)
LabelPropagation	0.946 785	0.922 225	0.946 785	0.928 675	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LabelSpreading	0.947 484	0.923 223	0.947 484	0.928 667	SelectKBest(f_classif, k=28)
LabelSpreading	0.947 484	0.923 223	0.947 484	0.928 667	SelectKBest(f_regression, k=28)
LabelSpreading	0.947 484	0.923 223	0.947 484	0.928 667	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LabelSpreading	0.947 484	0.923 223	0.947 484	0.928 667	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LabelSpreading	0.947 484	0.923 223	0.947 484	0.928 667	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LabelSpreading	0.947 484	0.923 223	0.947 484	0.928 667	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)

LabelSpreading	0.947 484	0.923 223	0.947 484	0.928 667	RFECV(estimator=SVC(), min_features_to_select=28)
LabelPropagation	0.946 739	0.922 126	0.946 739	0.928 647	RFE(estimator=SVC(), n_features_to_select=28)
LabelPropagation	0.946 739	0.922 126	0.946 739	0.928 647	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LabelPropagation	0.946 739	0.922 126	0.946 739	0.928 647	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LabelPropagation	0.946 739	0.922 126	0.946 739	0.928 647	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.931 282	0.926 189	0.931 282	0.928 63	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
DecisionTreeClassifier	0.931 142	0.926 266	0.931 142	0.928 607	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
MLPClassifier	0.948 461	0.925 436	0.948 461	0.928 583	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LabelSpreading	0.947 158	0.922 51	0.947 158	0.928 546	SelectKBest(mutual_info_classifier, k=28)
LabelSpreading	0.947 158	0.922 51	0.947 158	0.928 546	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LabelSpreading	0.947 111	0.922 398	0.947 111	0.928 518	RFE(estimator=SVC(), n_features_to_select=28)
LabelSpreading	0.947 111	0.922 398	0.947 111	0.928 518	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LabelSpreading	0.947 111	0.922 398	0.947 111	0.928 518	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LabelSpreading	0.947 111	0.922 398	0.947 111	0.928 518	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LabelPropagation	0.946 413	0.921 459	0.946 413	0.928 454	PCA(n_components=28)
MLPClassifier	0.948 461	0.925 255	0.948 461	0.928 432	RFE(estimator=MLPClassifier(), n_features_to_select=28)
HistGradientBoostingClassifier	0.949 486	0.930 277	0.949 486	0.928 406	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.930 723	0.926 237	0.930 723	0.928 4	SparsePCA(n_components=28)
DecisionTreeClassifier	0.930 863	0.926 029	0.930 863	0.928 353	SelectKBest(mutual_info_classifier, k=28)
DecisionTreeClassifier	0.931 095	0.925 79	0.931 095	0.928 33	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LabelSpreading	0.946 785	0.921 648	0.946 785	0.928 326	PCA(n_components=28)

HistGradientBoostin gClassifier	0.949 206	0.928 406	0.949 206	0.928 247	SelectKBest(f_classif, k=28)
HistGradientBoostin gClassifier	0.949 067	0.927 613	0.949 067	0.928 245	RFE(estimator=RandomForest Classifier(), n_features_to_select=28)
MLPClassifier	0.947 251	0.922 158	0.947 251	0.928 243	PCA(n_components=28)
MLPClassifier	0.948 508	0.925 166	0.948 508	0.928 232	RFECV(estimator=LogisticRe gression(), min_features_to_select=28)
HistGradientBoostin gClassifier	0.949 532	0.930 691	0.949 532	0.928 196	SelectKBest(mutual_info_clas sif, k=28)
HistGradientBoostin gClassifier	0.949 905	0.934 473	0.949 905	0.928 165	PCA(n_components=28)
ExtraTreeClassifier	0.931 421	0.925 122	0.931 421	0.928 114	SparsePCA(n_components=28 )
DecisionTreeClassifi er	0.930 397	0.925 92	0.930 397	0.928 08	SelectKBest(f_classif, k=28)
MLPClassifier	0.948 694	0.925 644	0.948 694	0.928 032	RFECV(estimator=RandomFo restClassifier(), min_features_to_select=28)
ExtraTreeClassifier	0.930 304	0.925 886	0.930 304	0.928 019	SelectKBest(f_classif, k=28)
HistGradientBoostin gClassifier	0.949 206	0.928 306	0.949 206	0.928 011	RFECV(estimator=RandomFo restClassifier(), min_features_to_select=28)
LGBMClassifier	0.949 253	0.928 598	0.949 253	0.927 958	MiniBatchSparsePCA(n_comp onents=28)
HistGradientBoostin gClassifier	0.949 206	0.928 274	0.949 206	0.927 931	MiniBatchSparsePCA(n_comp onents=28)
MLPClassifier	0.948 787	0.925 942	0.948 787	0.927 93	RFECV(estimator=MLPClassi fier(), min_features_to_select=28)
DecisionTreeClassifi er	0.929 885	0.926 058	0.929 885	0.927 915	PCA(n_components=28)
DecisionTreeClassifi er	0.930 397	0.925 593	0.930 397	0.927 906	SelectKBest(f_regression, k=28)
LinearDiscriminantA nalysis	0.943 154	0.919 152	0.943 154	0.927 866	MiniBatchSparsePCA(n_comp onents=28)
LinearDiscriminantA nalysis	0.943 107	0.919 102	0.943 107	0.927 838	SelectKBest(f_classif, k=28)
LinearDiscriminantA nalysis	0.943 107	0.919 102	0.943 107	0.927 838	SelectKBest(mutual_info_clas sif, k=28)
LinearDiscriminantA nalysis	0.943 107	0.919 102	0.943 107	0.927 838	SelectKBest(f_regression, k=28)
LinearDiscriminantA nalysis	0.943 107	0.919 102	0.943 107	0.927 838	RFE(estimator=DecisionTreeC lassifier(), n_features_to_select=28)
LinearDiscriminantA nalysis	0.943 107	0.919 102	0.943 107	0.927 838	RFE(estimator=RandomForest Classifier(), n_features_to_select=28)
LinearDiscriminantA nalysis	0.943 107	0.919 102	0.943 107	0.927 838	RFE(estimator=LogisticRegres sion(), n_features_to_select=28)

LinearDiscriminantAnalysis	0.943 107	0.919 102	0.943 107	0.927 838	RFE(estimator=SVC(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.943 107	0.919 102	0.943 107	0.927 838	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.943 107	0.919 102	0.943 107	0.927 838	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.943 107	0.919 102	0.943 107	0.927 838	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.943 107	0.919 102	0.943 107	0.927 838	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.943 107	0.919 102	0.943 107	0.927 838	RFECV(estimator=SVC(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.943 107	0.919 102	0.943 107	0.927 838	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.943 107	0.919 102	0.943 107	0.927 838	PCA(n_components=28)
LinearDiscriminantAnalysis	0.943 107	0.919 102	0.943 107	0.927 838	SparsePCA(n_components=28)
MLPClassifier	0.948 834	0.926 027	0.948 834	0.927 799	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.930 025	0.925 587	0.930 025	0.927 731	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
MLPClassifier	0.949 113	0.927 554	0.949 113	0.927 719	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
ExtraTreeClassifier	0.929 978	0.925 438	0.929 978	0.927 63	MiniBatchSparsePCA(n_components=28)
HistGradientBoostingClassifier	0.949 067	0.927 182	0.949 067	0.927 613	RFECV(estimator=SVC(), min_features_to_select=28)
HistGradientBoostingClassifier	0.949 346	0.929 262	0.949 346	0.927 609	RFE(estimator=LogisticRegression(), n_features_to_select=28)
DecisionTreeClassifier	0.929 838	0.925 519	0.929 838	0.927 608	RFE(estimator=MLPClassifier(), n_features_to_select=28)
MLPClassifier	0.949 02	0.926 866	0.949 02	0.927 587	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.929 606	0.925 5	0.929 606	0.927 49	RFE(estimator=SVC(), n_features_to_select=28)
MLPClassifier	0.948 694	0.924 968	0.948 694	0.927 484	RFECV(estimator=SVC(), min_features_to_select=28)
DecisionTreeClassifier	0.929 559	0.925 418	0.929 559	0.927 425	MiniBatchSparsePCA(n_components=28)
DecisionTreeClassifier	0.929 699	0.925 205	0.929 699	0.927 376	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.949 02	0.926 667	0.949 02	0.927 346	RFE(estimator=SVC(), n_features_to_select=28)

PassiveAggressiveClassifier	0.94081	0.918775	0.94081	0.927335	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.948601	0.923839	0.948601	0.927032	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
MLPClassifier	0.94888	0.925406	0.94888	0.927025	RFE(estimator=SVC(), n_features_to_select=28)
HistGradientBoostingClassifier	0.949299	0.928872	0.949299	0.927009	RFE(estimator=MLPClassifier(), n_features_to_select=28)
SGDClassifier	0.946832	0.918759	0.946832	0.926821	SparsePCA(n_components=28)
HistGradientBoostingClassifier	0.94902	0.926139	0.94902	0.926773	SelectKBest(f_regression, k=28)
HistGradientBoostingClassifier	0.948927	0.925344	0.948927	0.926722	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
VotingClassifier	0.948694	0.923565	0.948694	0.926595	PCA(n_components=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	SelectKBest(f_classif, k=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	SelectKBest(mutual_info_classif, k=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	SelectKBest(f_regression, k=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	RFE(estimator=LogisticRegression(), n_features_to_select=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	RFE(estimator=SVC(), n_features_to_select=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	RFE(estimator=MLPClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	RFECV(estimator=SVC(), min_features_to_select=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
VotingClassifier	0.948741	0.923747	0.948741	0.926538	SparsePCA(n_components=28)

MLPClassifier	0.948 322	0.921 418	0.948 322	0.926 473	MiniBatchSparsePCA(n_comp onents=28)
SGDClassifier	0.945 389	0.916 767	0.945 389	0.926 444	MiniBatchSparsePCA(n_comp onents=28)
VotingClassifier	0.948 834	0.924 182	0.948 834	0.926 422	RFE(estimator=RandomForest Classifier(), n_features_to_select=28)
VotingClassifier	0.948 834	0.924 182	0.948 834	0.926 422	RFE(estimator=LogisticRegres sion(), n_features_to_select=28)
GradientBoostingCla ssifier	0.949 439	0.931 409	0.949 439	0.926 41	RFE(estimator=DecisionTreeC lassifier(), n_features_to_select=28)
GradientBoostingCla ssifier	0.949 439	0.931 409	0.949 439	0.926 41	RFE(estimator=RandomForest Classifier(), n_features_to_select=28)
GradientBoostingCla ssifier	0.949 439	0.931 409	0.949 439	0.926 41	RFE(estimator=SVC(), n_features_to_select=28)
GradientBoostingCla ssifier	0.949 439	0.931 409	0.949 439	0.926 41	RFE(estimator=MLPClassifier (), n_features_to_select=28)
GradientBoostingCla ssifier	0.949 439	0.931 409	0.949 439	0.926 41	RFECV(estimator=LogisticRe gression(), min_features_to_select=28)
GradientBoostingCla ssifier	0.949 439	0.931 409	0.949 439	0.926 41	RFECV(estimator=SVC(), min_features_to_select=28)
VotingClassifier	0.948 787	0.923 795	0.948 787	0.926 397	RFECV(estimator=RandomFo restClassifier(), min_features_to_select=28)
GradientBoostingCla ssifier	0.949 392	0.930 606	0.949 392	0.926 385	SelectKBest(f_regression, k=28)
GradientBoostingCla ssifier	0.949 392	0.930 606	0.949 392	0.926 385	RFE(estimator=LogisticRegres sion(), n_features_to_select=28)
VotingClassifier	0.948 741	0.923 421	0.948 741	0.926 372	SelectKBest(f_regression, k=28)
VotingClassifier	0.948 741	0.923 421	0.948 741	0.926 372	RFECV(estimator=DecisionTr eeClassifier(), min_features_to_select=28)
VotingClassifier	0.948 694	0.923 06	0.948 694	0.926 347	RFE(estimator=SVC(), n_features_to_select=28)
VotingClassifier	0.948 694	0.923 06	0.948 694	0.926 347	RFE(estimator=MLPClassifier (), n_features_to_select=28)
VotingClassifier	0.948 694	0.923 06	0.948 694	0.926 347	RFECV(estimator=LogisticRe gression(), min_features_to_select=28)
VotingClassifier	0.948 834	0.924 028	0.948 834	0.926 339	RFECV(estimator=SVC(), min_features_to_select=28)
GradientBoostingCla ssifier	0.949 392	0.930 762	0.949 392	0.926 3	SelectKBest(f_classif, k=28)
GradientBoostingCla ssifier	0.949 392	0.930 762	0.949 392	0.926 3	SelectKBest(mutual_info_clas sif, k=28)

GradientBoostingClassifier	0.949 392	0.930 762	0.949 392	0.926 3	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
GradientBoostingClassifier	0.949 346	0.929 953	0.949 346	0.926 275	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
GradientBoostingClassifier	0.949 346	0.929 953	0.949 346	0.926 275	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
VotingClassifier	0.948 694	0.922 874	0.948 694	0.926 263	SelectKBest(f_classif, k=28)
VotingClassifier	0.948 648	0.922 515	0.948 648	0.926 238	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
VotingClassifier	0.948 648	0.922 515	0.948 648	0.926 238	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.949 113	0.926 497	0.949 113	0.926 15	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
SVC	0.948 927	0.924 444	0.948 927	0.926 136	SparsePCA(n_components=28)
VotingClassifier	0.948 601	0.921 953	0.948 601	0.926 129	SelectKBest(mutual_info_classif, k=28)
VotingClassifier	0.948 694	0.922 249	0.948 694	0.926 011	MiniBatchSparsePCA(n_components=28)
AdaBoostClassifier	0.949 113	0.926 351	0.949 113	0.925 979	MiniBatchSparsePCA(n_components=28)
GradientBoostingClassifier	0.949 253	0.928 593	0.949 253	0.925 967	SparsePCA(n_components=28)
GradientBoostingClassifier	0.949 299	0.929 597	0.949 299	0.925 905	PCA(n_components=28)
LogisticRegression	0.948 554	0.920 856	0.948 554	0.925 852	SparsePCA(n_components=28)
GaussianNB	0.936 915	0.917 399	0.936 915	0.925 769	SelectKBest(f_classif, k=28)
GaussianNB	0.936 915	0.917 399	0.936 915	0.925 769	SelectKBest(mutual_info_classif, k=28)
GaussianNB	0.936 915	0.917 399	0.936 915	0.925 769	SelectKBest(f_regression, k=28)
GaussianNB	0.936 915	0.917 399	0.936 915	0.925 769	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
GaussianNB	0.936 915	0.917 399	0.936 915	0.925 769	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
GaussianNB	0.936 915	0.917 399	0.936 915	0.925 769	RFE(estimator=LogisticRegression(), n_features_to_select=28)
GaussianNB	0.936 915	0.917 399	0.936 915	0.925 769	RFE(estimator=SVC(), n_features_to_select=28)
GaussianNB	0.936 915	0.917 399	0.936 915	0.925 769	RFE(estimator=MLPClassifier(), n_features_to_select=28)



GaussianNB	0.936 915	0.917 399	0.936 915	0.925 769	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
GaussianNB	0.936 915	0.917 399	0.936 915	0.925 769	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
GaussianNB	0.936 915	0.917 399	0.936 915	0.925 769	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
GaussianNB	0.936 915	0.917 399	0.936 915	0.925 769	RFECV(estimator=SVC(), min_features_to_select=28)
GaussianNB	0.936 915	0.917 399	0.936 915	0.925 769	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LogisticRegression	0.948 554	0.920 571	0.948 554	0.925 767	MiniBatchSparsePCA(n_components=28)
SGDClassifier	0.946 133	0.915 112	0.946 133	0.925 667	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.948 834	0.922 382	0.948 834	0.925 659	SparsePCA(n_components=28)
SGDClassifier	0.944 597	0.914 653	0.944 597	0.925 634	SelectKBest(mutual_info_classifier, k=28)
PassiveAggressiveClassifier	0.943 2	0.914 724	0.943 2	0.925 616	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
GradientBoostingClassifier	0.949 113	0.925 897	0.949 113	0.925 545	MiniBatchSparsePCA(n_components=28)
LogisticRegression	0.948 415	0.918 83	0.948 415	0.925 523	SelectKBest(f_classifier, k=28)
LogisticRegression	0.948 415	0.918 83	0.948 415	0.925 523	SelectKBest(mutual_info_classifier, k=28)
LogisticRegression	0.948 415	0.918 83	0.948 415	0.925 523	SelectKBest(f_regression, k=28)
LogisticRegression	0.948 415	0.918 83	0.948 415	0.925 523	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LogisticRegression	0.948 415	0.918 83	0.948 415	0.925 523	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LogisticRegression	0.948 415	0.918 83	0.948 415	0.925 523	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LogisticRegression	0.948 415	0.918 83	0.948 415	0.925 523	RFE(estimator=SVC(), n_features_to_select=28)
LogisticRegression	0.948 415	0.918 83	0.948 415	0.925 523	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LogisticRegression	0.948 415	0.918 83	0.948 415	0.925 523	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LogisticRegression	0.948 415	0.918 83	0.948 415	0.925 523	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)

LogisticRegression	0.948 415	0.918 83	0.948 415	0.925 523	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LogisticRegression	0.948 415	0.918 83	0.948 415	0.925 523	RFECV(estimator=SVC(), min_features_to_select=28)
LogisticRegression	0.948 415	0.918 83	0.948 415	0.925 523	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LogisticRegression	0.948 415	0.918 83	0.948 415	0.925 523	PCA(n_components=28)
SVC	0.949 206	0.927 909	0.949 206	0.925 506	SelectKBest(f_classif, k=28)
SVC	0.949 206	0.927 909	0.949 206	0.925 506	SelectKBest(mutual_info_classif, k=28)
SVC	0.949 206	0.927 909	0.949 206	0.925 506	SelectKBest(f_regression, k=28)
SVC	0.949 206	0.927 909	0.949 206	0.925 506	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
SVC	0.949 206	0.927 909	0.949 206	0.925 506	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SVC	0.949 206	0.927 909	0.949 206	0.925 506	RFE(estimator=LogisticRegression(), n_features_to_select=28)
SVC	0.949 206	0.927 909	0.949 206	0.925 506	RFE(estimator=SVC(), n_features_to_select=28)
SVC	0.949 206	0.927 909	0.949 206	0.925 506	RFE(estimator=MLPClassifier(), n_features_to_select=28)
SVC	0.949 206	0.927 909	0.949 206	0.925 506	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
SVC	0.949 206	0.927 909	0.949 206	0.925 506	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
SVC	0.949 206	0.927 909	0.949 206	0.925 506	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
SVC	0.949 206	0.927 909	0.949 206	0.925 506	RFECV(estimator=SVC(), min_features_to_select=28)
SVC	0.949 206	0.927 909	0.949 206	0.925 506	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
SVC	0.949 206	0.927 909	0.949 206	0.925 506	PCA(n_components=28)
SGDClassifier	0.948 508	0.918 801	0.948 508	0.925 401	PCA(n_components=28)
SGDClassifier	0.948 368	0.917 688	0.948 368	0.925 328	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
PassiveAggressiveClassifier	0.941 524	0.914 523	0.941 524	0.925 317	RFECV(estimator=SVC(), min_features_to_select=28)
SVC	0.948 834	0.921 045	0.948 834	0.925 312	MiniBatchSparsePCA(n_components=28)

RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	SelectKBest(f_classif, k=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	SelectKBest(f_classif, k=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	SelectKBest(mutual_info_classif, k=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	SelectKBest(mutual_info_classif, k=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	SelectKBest(f_regression, k=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	SelectKBest(f_regression, k=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	RFE(estimator=LogisticRegression(), n_features_to_select=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	RFE(estimator=LogisticRegression(), n_features_to_select=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	RFE(estimator=SVC(), n_features_to_select=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	RFE(estimator=SVC(), n_features_to_select=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	RFE(estimator=MLPClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	RFE(estimator=MLPClassifier(), n_features_to_select=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	RFECV(estimator=LogisticRegression(), min_features_to_select=28)

RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	RFECV(estimator=SVC(), min_features_to_select=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	RFECV(estimator=SVC(), min_features_to_select=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	PCA(n_components=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	PCA(n_components=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	SparsePCA(n_components=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	SparsePCA(n_components=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	MiniBatchSparsePCA(n_components=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	MiniBatchSparsePCA(n_components=28)
PassiveAggressiveClassifier	0.937 986	0.915 961	0.937 986	0.925 278	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.948 508	0.917 913	0.948 508	0.925 229	PCA(n_components=28)
PassiveAggressiveClassifier	0.943 992	0.913 656	0.943 992	0.925 221	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SGDClassifier	0.946 832	0.913 567	0.946 832	0.925 097	RFE(estimator=SVC(), n_features_to_select=28)
SGDClassifier	0.948 554	0.916 679	0.948 554	0.924 992	SelectKBest(f_regression, k=28)
SGDClassifier	0.948 415	0.915 388	0.948 415	0.924 92	RFE(estimator=LogisticRegression(), n_features_to_select=28)
PassiveAggressiveClassifier	0.948 834	0.918 113	0.948 834	0.924 871	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
SGDClassifier	0.949 113	0.924 21	0.949 113	0.924 834	RFECV(estimator=SVC(), min_features_to_select=28)
SGDClassifier	0.948 694	0.914 986	0.948 694	0.924 71	SelectKBest(f_classif, k=28)
SGDClassifier	0.949 253	0.951 828	0.949 253	0.924 633	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
SGDClassifier	0.948 508	0.912 843	0.948 508	0.924 615	RFE(estimator=MLPClassifier(), n_features_to_select=28)

SGDClassifier	0.948 648	0.913 183	0.948 648	0.924 597	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
SGDClassifier	0.949 067	0.917 926	0.949 067	0.924 539	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
SGDClassifier	0.948 089	0.910 749	0.948 089	0.924 489	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
PassiveAggressiveClassifier	0.946 971	0.910 37	0.946 971	0.924 426	RFE(estimator=LogisticRegression(), n_features_to_select=28)
PassiveAggressiveClassifier	0.947 344	0.909 202	0.947 344	0.924 28	PCA(n_components=28)
PassiveAggressiveClassifier	0.940 407	0.911 924	0.940 407	0.923 89	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
PassiveAggressiveClassifier	0.937 986	0.912 549	0.937 986	0.923 52	RFE(estimator=SVC(), n_features_to_select=28)
PassiveAggressiveClassifier	0.931 421	0.915 563	0.931 421	0.922 825	SelectKBest(mutual_info_classif, k=28)
PassiveAggressiveClassifier	0.944 597	0.904 14	0.944 597	0.922 7	SelectKBest(f_regression, k=28)
PassiveAggressiveClassifier	0.939 15	0.909 569	0.939 15	0.922 519	SparsePCA(n_components=28)
PassiveAggressiveClassifier	0.940 267	0.906 113	0.940 267	0.921 594	RFE(estimator=MLPClassifier(), n_features_to_select=28)
GaussianNB	0.925 741	0.916 848	0.925 741	0.921 106	PCA(n_components=28)
QuadraticDiscriminantAnalysis	0.925 788	0.916 627	0.925 788	0.921 009	PCA(n_components=28)
QuadraticDiscriminantAnalysis	0.924 903	0.917 216	0.924 903	0.920 919	MiniBatchSparsePCA(n_components=28)
GaussianNB	0.924 95	0.916 767	0.924 95	0.920 702	SparsePCA(n_components=28)
QuadraticDiscriminantAnalysis	0.924 95	0.916 14	0.924 95	0.920 37	SelectKBest(f_regression, k=28)
QuadraticDiscriminantAnalysis	0.924 531	0.916 489	0.924 531	0.920 363	RFECV(estimator=SVC(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.924 205	0.916 628	0.924 205	0.920 286	RFE(estimator=MLPClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.924 065	0.916 588	0.924 065	0.920 2	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.924 019	0.916 575	0.924 019	0.920 172	SparsePCA(n_components=28)
QuadraticDiscriminantAnalysis	0.924 112	0.916 369	0.924 112	0.920 106	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.923 553	0.916 597	0.923 553	0.919 968	RFE(estimator=LogisticRegression(), n_features_to_select=28)

QuadraticDiscriminantAnalysis	0.923 6	0.916 533	0.923 6	0.919 956	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.923 414	0.916 635	0.923 414	0.919 922	SelectKBest(mutual_info_classifier, k=28)
QuadraticDiscriminantAnalysis	0.923 274	0.916 596	0.923 274	0.919 837	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.923 134	0.916 711	0.923 134	0.919 831	SelectKBest(f_classifier, k=28)
QuadraticDiscriminantAnalysis	0.923 227	0.916 584	0.923 227	0.919 808	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.923 274	0.916 52	0.923 274	0.919 796	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.923 227	0.916 507	0.923 227	0.919 768	RFE(estimator=SVC(), n_features_to_select=28)
GaussianNB	0.923 553	0.915 428	0.923 553	0.919 352	MiniBatchSparsePCA(n_components=28)
CategoricalNB	0.916 104	0.920 773	0.916 104	0.918 391	SelectKBest(f_classifier, k=28)
CategoricalNB	0.916 104	0.920 773	0.916 104	0.918 391	SelectKBest(mutual_info_classifier, k=28)
CategoricalNB	0.916 104	0.920 773	0.916 104	0.918 391	SelectKBest(f_regression, k=28)
CategoricalNB	0.916 104	0.920 773	0.916 104	0.918 391	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
CategoricalNB	0.916 104	0.920 773	0.916 104	0.918 391	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
CategoricalNB	0.916 104	0.920 773	0.916 104	0.918 391	RFE(estimator=LogisticRegression(), n_features_to_select=28)
CategoricalNB	0.916 104	0.920 773	0.916 104	0.918 391	RFE(estimator=SVC(), n_features_to_select=28)
CategoricalNB	0.916 104	0.920 773	0.916 104	0.918 391	RFE(estimator=MLPClassifier(), n_features_to_select=28)
CategoricalNB	0.916 104	0.920 773	0.916 104	0.918 391	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
CategoricalNB	0.916 104	0.920 773	0.916 104	0.918 391	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
CategoricalNB	0.916 104	0.920 773	0.916 104	0.918 391	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
CategoricalNB	0.916 104	0.920 773	0.916 104	0.918 391	RFECV(estimator=SVC(), min_features_to_select=28)
CategoricalNB	0.916 104	0.920 773	0.916 104	0.918 391	RFECV(estimator=MLPClassifier(), min_features_to_select=28)

PassiveAggressiveClassifier	0.830 392	0.917 257	0.830 392	0.868 12	SelectKBest(f_classif, k=28)
PassiveAggressiveClassifier	0.828 344	0.912 276	0.828 344	0.865 637	MiniBatchSparsePCA(n_components=28)
BernoulliNB	0.820 802	0.916 849	0.820 802	0.862 238	SparsePCA(n_components=28)
BernoulliNB	0.823 735	0.908 522	0.823 735	0.861 923	MiniBatchSparsePCA(n_components=28)
BernoulliNB	0.819 871	0.917 136	0.819 871	0.861 737	PCA(n_components=28)
BernoulliNB	0.764 375	0.927 738	0.764 375	0.828 126	SelectKBest(f_classif, k=28)
BernoulliNB	0.764 375	0.927 738	0.764 375	0.828 126	SelectKBest(mutual_info_classifier, k=28)
BernoulliNB	0.764 375	0.927 738	0.764 375	0.828 126	SelectKBest(f_regression, k=28)
BernoulliNB	0.764 375	0.927 738	0.764 375	0.828 126	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
BernoulliNB	0.764 375	0.927 738	0.764 375	0.828 126	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
BernoulliNB	0.764 375	0.927 738	0.764 375	0.828 126	RFE(estimator=LogisticRegression(), n_features_to_select=28)
BernoulliNB	0.764 375	0.927 738	0.764 375	0.828 126	RFE(estimator=SVC(), n_features_to_select=28)
BernoulliNB	0.764 375	0.927 738	0.764 375	0.828 126	RFE(estimator=MLPClassifier(), n_features_to_select=28)
BernoulliNB	0.764 375	0.927 738	0.764 375	0.828 126	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
BernoulliNB	0.764 375	0.927 738	0.764 375	0.828 126	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
BernoulliNB	0.764 375	0.927 738	0.764 375	0.828 126	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
BernoulliNB	0.764 375	0.927 738	0.764 375	0.828 126	RFECV(estimator=SVC(), min_features_to_select=28)
BernoulliNB	0.764 375	0.927 738	0.764 375	0.828 126	RFECV(estimator=MLPClassifier(), min_features_to_select=28)

## APPENDIX 6: VULNERABILITY CLASS LEVEL DATA RESULTS

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
RandomForestClassifier	0.947347	0.936937	0.947347	0.940376	RFECV(estimator=SVC(), min_features_to_select=28)
BaggingClassifier	0.946722	0.936175	0.946722	0.939751	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
BaggingClassifier	0.947138	0.936016	0.947138	0.939561	SelectKBest(mutual_info_classifier, k=28)
RandomForestClassifier	0.946722	0.935988	0.946722	0.939587	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
BaggingClassifier	0.946514	0.935986	0.946514	0.939597	RFECV(estimator=SVC(), min_features_to_select=28)
RandomForestClassifier	0.946514	0.935796	0.946514	0.939434	RFE(estimator=MLPClassifier(), n_features_to_select=28)
RandomForestClassifier	0.946722	0.935614	0.946722	0.939254	SelectKBest(f_regression, k=28)
RandomForestClassifier	0.946722	0.935427	0.946722	0.939085	RFE(estimator=LogisticRegression(), n_features_to_select=28)
RandomForestClassifier	0.946098	0.935225	0.946098	0.938962	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
RandomForestClassifier	0.946306	0.935223	0.946306	0.938948	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
RandomForestClassifier	0.945473	0.935075	0.945473	0.938832	RFE(estimator=SVC(), n_features_to_select=28)
RandomForestClassifier	0.947138	0.934936	0.947138	0.938516	MiniBatchSparsePCA(n_components=28)
LGBMClassifier	0.946722	0.934868	0.946722	0.938567	PCA(n_components=28)
RandomForestClassifier	0.945682	0.934854	0.945682	0.938657	SelectKBest(mutual_info_classifier, k=28)
RandomForestClassifier	0.94589	0.934842	0.94589	0.938643	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
BaggingClassifier	0.946098	0.934836	0.946098	0.938627	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
XGBClassifier	0.946514	0.93466	0.946514	0.938415	MiniBatchSparsePCA(n_components=28)
XGBClassifier	0.946306	0.934646	0.946306	0.938437	PCA(n_components=28)
BaggingClassifier	0.946722	0.934496	0.946722	0.938212	MiniBatchSparsePCA(n_components=28)
RandomForestClassifier	0.945682	0.934255	0.945682	0.938153	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LGBMClassifier	0.946306	0.93407	0.946306	0.937909	SparsePCA(n_components=28)



RandomForestClassifier	0.945682	0.934055	0.945682	0.937982	SelectKBest(f_classif, k=28)
BaggingClassifier	0.94589	0.934052	0.94589	0.93796	SelectKBest(f_regression, k=28)
LGBMClassifier	0.947138	0.933692	0.947138	0.937208	KernelPCA(kernel="linear", n_components=28)
RandomForestClassifier	0.946306	0.933685	0.946306	0.937548	KernelPCA(kernel="linear", n_components=28)
BaggingClassifier	0.945682	0.933654	0.945682	0.937633	RFE(estimator=LogisticRegression(), n_features_to_select=28)
BaggingClassifier	0.945682	0.933252	0.945682	0.937277	RFE(estimator=SVC(), n_features_to_select=28)
BaggingClassifier	0.945057	0.933072	0.945057	0.937182	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
BaggingClassifier	0.945265	0.933057	0.945265	0.937155	KernelPCA(kernel="linear", n_components=28)
DecisionTreeClassifier	0.937357	0.932999	0.937357	0.935044	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LGBMClassifier	0.946306	0.932916	0.946306	0.936801	MiniBatchSparsePCA(n_components=28)
RandomForestClassifier	0.945265	0.93285	0.945265	0.936977	PCA(n_components=28)
HistGradientBoostingClassifier	0.946098	0.932687	0.946098	0.936651	SparsePCA(n_components=28)
BaggingClassifier	0.945682	0.932646	0.945682	0.936729	PCA(n_components=28)
RidgeClassifierCV	0.948179	0.932614	0.948179	0.928172	SelectKBest(f_classif, k=28)
RidgeClassifierCV	0.948179	0.932614	0.948179	0.928172	SelectKBest(mutual_info_classif, k=28)
RidgeClassifierCV	0.948179	0.932614	0.948179	0.928172	SelectKBest(f_regression, k=28)
RidgeClassifierCV	0.948179	0.932614	0.948179	0.928172	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.948179	0.932614	0.948179	0.928172	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.948179	0.932614	0.948179	0.928172	RFE(estimator=LogisticRegression(), n_features_to_select=28)
RidgeClassifierCV	0.948179	0.932614	0.948179	0.928172	RFE(estimator=SVC(), n_features_to_select=28)
RidgeClassifierCV	0.948179	0.932614	0.948179	0.928172	RFE(estimator=MLPClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.948179	0.932614	0.948179	0.928172	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.948179	0.932614	0.948179	0.928172	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)

RidgeClassifierCV	0.948 179	0.932 614	0.948 179	0.928 172	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
RidgeClassifierCV	0.948 179	0.932 614	0.948 179	0.928 172	RFECV(estimator=SVC(), min_features_to_select=28)
RidgeClassifierCV	0.948 179	0.932 614	0.948 179	0.928 172	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.948 179	0.932 614	0.948 179	0.928 172	PCA(n_components=28)
RandomForestClassifier	0.945 057	0.932 233	0.945 057	0.936 465	SparsePCA(n_components=28)
RandomForestClassifier	0.944 017	0.932 153	0.944 017	0.936 433	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
StackingClassifier	0.947 138	0.932 128	0.947 138	0.935 367	KernelPCA(kernel="linear", n_components=28)
ExtraTreeClassifier	0.938 398	0.932 093	0.938 398	0.934 948	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
BaggingClassifier	0.944 641	0.932 053	0.944 641	0.936 349	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
DecisionTreeClassifier	0.934 443	0.931 87	0.934 443	0.933 117	SelectKBest(f_classif, k=28)
XGBClassifier	0.944 433	0.931 862	0.944 433	0.936 2	SparsePCA(n_components=28)
DecisionTreeClassifier	0.934 651	0.931 714	0.934 651	0.933 13	RFE(estimator=SVC(), n_features_to_select=28)
StackingClassifier	0.946 93	0.931 656	0.946 93	0.935 004	SparsePCA(n_components=28)
HistGradientBoostingClassifier	0.946 514	0.931 638	0.946 514	0.935 351	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
ExtraTreeClassifier	0.936 941	0.931 57	0.936 941	0.934 058	SelectKBest(f_regression, k=28)
ExtraTreeClassifier	0.939 438	0.931 474	0.939 438	0.934 95	RFE(estimator=SVC(), n_features_to_select=28)
HistGradientBoostingClassifier	0.946 514	0.931 449	0.946 514	0.935 141	RFE(estimator=MLPClassifier(), n_features_to_select=28)
BaggingClassifier	0.943 6	0.931 351	0.943 6	0.935 784	RFE(estimator=MLPClassifier(), n_features_to_select=28)
XGBClassifier	0.944 225	0.931 231	0.944 225	0.935 687	KernelPCA(kernel="linear", n_components=28)
PassiveAggressiveClassifier	0.920 708	0.931 183	0.920 708	0.925 552	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.934 86	0.931 059	0.934 86	0.932 871	SelectKBest(mutual_info_classif, k=28)
BaggingClassifier	0.944 017	0.931 04	0.944 017	0.935 539	SelectKBest(f_classif, k=28)
DecisionTreeClassifier	0.934 651	0.930 963	0.934 651	0.932 725	RFE(estimator=LogisticRegression(), n_features_to_select=28)

BernoulliNB	0.867 222	0.930 958	0.867 222	0.893 352	SelectKBest(f_classif, k=28)
BernoulliNB	0.867 222	0.930 958	0.867 222	0.893 352	SelectKBest(mutual_info_classif, k=28)
BernoulliNB	0.867 222	0.930 958	0.867 222	0.893 352	SelectKBest(f_regression, k=28)
BernoulliNB	0.867 222	0.930 958	0.867 222	0.893 352	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
BernoulliNB	0.867 222	0.930 958	0.867 222	0.893 352	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
BernoulliNB	0.867 222	0.930 958	0.867 222	0.893 352	RFE(estimator=LogisticRegression(), n_features_to_select=28)
BernoulliNB	0.867 222	0.930 958	0.867 222	0.893 352	RFE(estimator=SVC(), n_features_to_select=28)
BernoulliNB	0.867 222	0.930 958	0.867 222	0.893 352	RFE(estimator=MLPClassifier(), n_features_to_select=28)
BernoulliNB	0.867 222	0.930 958	0.867 222	0.893 352	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
BernoulliNB	0.867 222	0.930 958	0.867 222	0.893 352	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
BernoulliNB	0.867 222	0.930 958	0.867 222	0.893 352	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
BernoulliNB	0.867 222	0.930 958	0.867 222	0.893 352	RFECV(estimator=SVC(), min_features_to_select=28)
BernoulliNB	0.867 222	0.930 958	0.867 222	0.893 352	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.946 93	0.930 952	0.946 93	0.934 113	KernelPCA(kernel="linear", n_components=28)
MLPClassifier	0.946 514	0.930 882	0.946 514	0.934 494	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.936 524	0.930 862	0.936 524	0.933 48	SparsePCA(n_components=28)
StackingClassifier	0.946 306	0.930 788	0.946 306	0.934 567	SelectKBest(f_regression, k=28)
DecisionTreeClassifier	0.934 651	0.930 711	0.934 651	0.932 588	RFE(estimator=MLPClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.935 692	0.930 699	0.935 692	0.933 037	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
StackingClassifier	0.946 722	0.930 629	0.946 722	0.933 969	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.945 473	0.930 562	0.945 473	0.934 825	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)

ExtraTreeClassifier	0.937 357	0.930 545	0.937 357	0.933 628	RFE(estimator=MLPClassifier(), n_features_to_select=28)
StackingClassifier	0.946 514	0.930 503	0.946 514	0.934 051	RFECV(estimator=SVC(), min_features_to_select=28)
ExtraTreeClassifier	0.934 027	0.930 427	0.934 027	0.932 151	SelectKBest(mutual_info_classifier, k=28)
DecisionTreeClassifier	0.934 027	0.930 427	0.934 027	0.932 151	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
BaggingClassifier	0.942 768	0.930 426	0.942 768	0.935 015	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
StackingClassifier	0.946 306	0.930 397	0.946 306	0.934 129	SelectKBest(mutual_info_classifier, k=28)
DecisionTreeClassifier	0.933 819	0.930 333	0.933 819	0.932 006	SelectKBest(f_regression, k=28)
DecisionTreeClassifier	0.933 819	0.930 333	0.933 819	0.932 006	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.933 819	0.930 333	0.933 819	0.932 006	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LGBMClassifier	0.945 265	0.930 322	0.945 265	0.934 679	SelectKBest(f_classifier, k=28)
LGBMClassifier	0.945 265	0.930 322	0.945 265	0.934 679	SelectKBest(mutual_info_classifier, k=28)
LGBMClassifier	0.945 265	0.930 322	0.945 265	0.934 679	SelectKBest(f_regression, k=28)
LGBMClassifier	0.945 265	0.930 322	0.945 265	0.934 679	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LGBMClassifier	0.945 265	0.930 322	0.945 265	0.934 679	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LGBMClassifier	0.945 265	0.930 322	0.945 265	0.934 679	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LGBMClassifier	0.945 265	0.930 322	0.945 265	0.934 679	RFE(estimator=SVC(), n_features_to_select=28)
LGBMClassifier	0.945 265	0.930 322	0.945 265	0.934 679	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LGBMClassifier	0.945 265	0.930 322	0.945 265	0.934 679	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LGBMClassifier	0.945 265	0.930 322	0.945 265	0.934 679	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LGBMClassifier	0.945 265	0.930 322	0.945 265	0.934 679	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LGBMClassifier	0.945 265	0.930 322	0.945 265	0.934 679	RFECV(estimator=SVC(), min_features_to_select=28)

LGBMClassifier	0.945 265	0.930 322	0.945 265	0.934 679	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
StackingClassifier	0.946 098	0.930 309	0.946 098	0.934 204	RFE(estimator=SVC(), n_features_to_select=28)
BaggingClassifier	0.943 6	0.930 206	0.943 6	0.934 874	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
StackingClassifier	0.946 306	0.930 202	0.946 306	0.933 907	PCA(n_components=28)
DecisionTreeClassifier	0.934 027	0.930 172	0.934 027	0.932 013	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
ExtraTreeClassifier	0.933 403	0.930 149	0.933 403	0.931 716	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.934 443	0.930 107	0.934 443	0.932 164	KernelPCA(kernel="linear", n_components=28)
BaggingClassifier	0.943 809	0.929 934	0.943 809	0.934 643	SparsePCA(n_components=28)
VotingClassifier	0.946 93	0.929 911	0.946 93	0.932 694	KernelPCA(kernel="linear", n_components=28)
MLPClassifier	0.946 098	0.929 906	0.946 098	0.933 763	RFE(estimator=SVC(), n_features_to_select=28)
ExtraTreeClassifier	0.935 9	0.929 789	0.935 9	0.932 607	RFECV(estimator=SVC(), min_features_to_select=28)
StackingClassifier	0.946 098	0.929 704	0.946 098	0.933 538	RFE(estimator=LogisticRegression(), n_features_to_select=28)
ExtraTreeClassifier	0.935 692	0.929 685	0.935 692	0.932 461	SelectKBest(f_classif, k=28)
DecisionTreeClassifier	0.934 027	0.929 659	0.934 027	0.931 733	RFECV(estimator=SVC(), min_features_to_select=28)
PassiveAggressiveClassifier	0.912 175	0.929 585	0.912 175	0.920 028	PCA(n_components=28)
ExtraTreeClassifier	0.934 235	0.929 498	0.934 235	0.931 736	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
StackingClassifier	0.945 473	0.929 493	0.945 473	0.933 77	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
RidgeClassifier	0.947 555	0.929 471	0.947 555	0.928 417	KernelPCA(kernel="linear", n_components=28)
RidgeClassifierCV	0.947 555	0.929 471	0.947 555	0.928 417	KernelPCA(kernel="linear", n_components=28)
RidgeClassifier	0.947 555	0.929 471	0.947 555	0.928 417	SparsePCA(n_components=28)
RidgeClassifierCV	0.947 555	0.929 471	0.947 555	0.928 417	SparsePCA(n_components=28)
RidgeClassifier	0.947 555	0.929 471	0.947 555	0.928 417	MiniBatchSparsePCA(n_components=28)
RidgeClassifierCV	0.947 555	0.929 471	0.947 555	0.928 417	MiniBatchSparsePCA(n_components=28)

ExtraTreeClassifier	0.936 733	0.929 453	0.936 733	0.932 741	PCA(n_components=28)
DecisionTreeClassifier	0.935 068	0.929 377	0.935 068	0.932 026	MiniBatchSparsePCA(n_components=28)
StackingClassifier	0.945 89	0.929 2	0.945 89	0.933 167	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
ExtraTreeClassifier	0.935 692	0.929 171	0.935 692	0.932 165	RFE(estimator=LogisticRegression(), n_features_to_select=28)
ExtraTreeClassifier	0.935 484	0.929 066	0.935 484	0.932 021	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
StackingClassifier	0.945 473	0.929 057	0.945 473	0.933 331	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
MLPClassifier	0.943 6	0.929 029	0.943 6	0.933 915	SelectKBest(f_regression, k=28)
DecisionTreeClassifier	0.932 986	0.928 933	0.932 986	0.930 87	PCA(n_components=28)
ExtraTreeClassifier	0.932 986	0.928 933	0.932 986	0.930 87	SparsePCA(n_components=28)
StackingClassifier	0.945 682	0.928 905	0.945 682	0.933 024	RFE(estimator=MLPClassifier(), n_features_to_select=28)
StackingClassifier	0.945 473	0.928 837	0.945 473	0.933 108	SelectKBest(f_classif, k=28)
StackingClassifier	0.945 473	0.928 837	0.945 473	0.933 108	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
VotingClassifier	0.946 306	0.928 817	0.946 306	0.932 271	SparsePCA(n_components=28)
SGDClassifier	0.927 992	0.928 784	0.927 992	0.928 385	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
MLPClassifier	0.945 89	0.928 777	0.945 89	0.932 705	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
ExtraTreeClassifier	0.931 738	0.928 658	0.931 738	0.930 149	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
MLPClassifier	0.943 184	0.928 632	0.943 184	0.933 624	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
RidgeClassifier	0.947 347	0.928 566	0.947 347	0.927 662	SelectKBest(f_classif, k=28)
RidgeClassifier	0.947 347	0.928 566	0.947 347	0.927 662	SelectKBest(mutual_info_classif, k=28)
RidgeClassifier	0.947 347	0.928 566	0.947 347	0.927 662	SelectKBest(f_regression, k=28)
RidgeClassifier	0.947 347	0.928 566	0.947 347	0.927 662	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)

RidgeClassifier	0.947 347	0.928 566	0.947 347	0.927 662	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
RidgeClassifier	0.947 347	0.928 566	0.947 347	0.927 662	RFE(estimator=LogisticRegression(), n_features_to_select=28)
RidgeClassifier	0.947 347	0.928 566	0.947 347	0.927 662	RFE(estimator=SVC(), n_features_to_select=28)
RidgeClassifier	0.947 347	0.928 566	0.947 347	0.927 662	RFE(estimator=MLPClassifier(), n_features_to_select=28)
RidgeClassifier	0.947 347	0.928 566	0.947 347	0.927 662	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
RidgeClassifier	0.947 347	0.928 566	0.947 347	0.927 662	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
RidgeClassifier	0.947 347	0.928 566	0.947 347	0.927 662	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
RidgeClassifier	0.947 347	0.928 566	0.947 347	0.927 662	RFECV(estimator=SVC(), min_features_to_select=28)
RidgeClassifier	0.947 347	0.928 566	0.947 347	0.927 662	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
RidgeClassifier	0.947 347	0.928 566	0.947 347	0.927 662	PCA(n_components=28)
GradientBoostingClassifier	0.945 89	0.928 564	0.945 89	0.932 47	KernelPCA(kernel="linear", n_components=28)
AdaBoostClassifier	0.945 89	0.928 564	0.945 89	0.932 47	MiniBatchSparsePCA(n_components=28)
MLPClassifier	0.945 057	0.928 52	0.945 057	0.933 045	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.944 849	0.928 492	0.944 849	0.933 122	SelectKBest(f_regression, k=28)
MLPClassifier	0.946 098	0.928 471	0.946 098	0.932 131	SelectKBest(mutual_info_classif, k=28)
PassiveAggressiveClassifier	0.946 514	0.928 405	0.946 514	0.931 41	SelectKBest(f_regression, k=28)
MLPClassifier	0.945 473	0.928 393	0.945 473	0.932 653	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LabelSpreading	0.947 138	0.928 367	0.947 138	0.929 356	MiniBatchSparsePCA(n_components=28)
PassiveAggressiveClassifier	0.925 078	0.928 318	0.925 078	0.926 656	RFE(estimator=LogisticRegression(), n_features_to_select=28)
ExtraTreeClassifier	0.931 53	0.928 307	0.931 53	0.929 866	KernelPCA(kernel="linear", n_components=28)
GradientBoostingClassifier	0.945 682	0.928 25	0.945 682	0.932 328	SparsePCA(n_components=28)
XGBClassifier	0.942 352	0.928 127	0.942 352	0.933 24	SelectKBest(f_classif, k=28)

XGBClassifier	0.942 352	0.928 127	0.942 352	0.933 24	SelectKBest(mutual_info_classifier, k=28)
XGBClassifier	0.942 352	0.928 127	0.942 352	0.933 24	SelectKBest(f_regression, k=28)
XGBClassifier	0.942 352	0.928 127	0.942 352	0.933 24	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
XGBClassifier	0.942 352	0.928 127	0.942 352	0.933 24	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
XGBClassifier	0.942 352	0.928 127	0.942 352	0.933 24	RFE(estimator=LogisticRegression(), n_features_to_select=28)
XGBClassifier	0.942 352	0.928 127	0.942 352	0.933 24	RFE(estimator=SVC(), n_features_to_select=28)
XGBClassifier	0.942 352	0.928 127	0.942 352	0.933 24	RFE(estimator=MLPClassifier(), n_features_to_select=28)
XGBClassifier	0.942 352	0.928 127	0.942 352	0.933 24	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
XGBClassifier	0.942 352	0.928 127	0.942 352	0.933 24	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
XGBClassifier	0.942 352	0.928 127	0.942 352	0.933 24	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
XGBClassifier	0.942 352	0.928 127	0.942 352	0.933 24	RFECV(estimator=SVC(), min_features_to_select=28)
XGBClassifier	0.942 352	0.928 127	0.942 352	0.933 24	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.945 265	0.928 105	0.945 265	0.932 511	PCA(n_components=28)
HistGradientBoostingClassifier	0.945 057	0.928 057	0.945 057	0.932 597	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
StackingClassifier	0.944 849	0.928 024	0.944 849	0.932 68	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
HistGradientBoostingClassifier	0.945 473	0.927 942	0.945 473	0.932 187	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
ExtraTreeClassifier	0.934 027	0.927 832	0.934 027	0.930 713	MiniBatchSparsePCA(n_components=28)
StackingClassifier	0.945 057	0.927 823	0.945 057	0.932 37	MiniBatchSparsePCA(n_components=28)
HistGradientBoostingClassifier	0.944 849	0.927 787	0.944 849	0.932 455	SelectKBest(mutual_info_classifier, k=28)
MLPClassifier	0.943 809	0.927 782	0.943 809	0.932 835	PCA(n_components=28)
GradientBoostingClassifier	0.945 473	0.927 714	0.945 473	0.931 95	MiniBatchSparsePCA(n_components=28)



MLPClassifier	0.946 514	0.927 628	0.946 514	0.930 356	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
PassiveAggressiveClassifier	0.946 098	0.927 625	0.946 098	0.931 133	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.945 682	0.927 579	0.945 682	0.931 607	MiniBatchSparsePCA(n_components=28)
SelfTrainingClassifier	0.945 682	0.927 579	0.945 682	0.931 607	MiniBatchSparsePCA(n_components=28)
GradientBoostingClassifier	0.944 849	0.927 548	0.944 849	0.932 228	PCA(n_components=28)
MLPClassifier	0.945 89	0.927 475	0.945 89	0.931 249	SelectKBest(f_classif, k=28)
MLPClassifier	0.945 682	0.927 351	0.945 682	0.931 36	RFE(estimator=MLPClassifier(), n_features_to_select=28)
SGDClassifier	0.941 727	0.927 334	0.941 727	0.932 615	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.944 017	0.927 265	0.944 017	0.932 332	RFE(estimator=SVC(), n_features_to_select=28)
PassiveAggressiveClassifier	0.925 078	0.927 253	0.925 078	0.926 147	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
HistGradientBoostingClassifier	0.944 641	0.927 031	0.944 641	0.931 857	MiniBatchSparsePCA(n_components=28)
SGDClassifier	0.941 935	0.926 99	0.941 935	0.932 365	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
SGDClassifier	0.940 895	0.926 918	0.940 895	0.932 233	RFE(estimator=MLPClassifier(), n_features_to_select=28)
GaussianNB	0.936 316	0.926 864	0.936 316	0.931 03	SelectKBest(f_classif, k=28)
GaussianNB	0.936 316	0.926 864	0.936 316	0.931 03	SelectKBest(mutual_info_classif, k=28)
GaussianNB	0.936 316	0.926 864	0.936 316	0.931 03	SelectKBest(f_regression, k=28)
GaussianNB	0.936 316	0.926 864	0.936 316	0.931 03	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
GaussianNB	0.936 316	0.926 864	0.936 316	0.931 03	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
GaussianNB	0.936 316	0.926 864	0.936 316	0.931 03	RFE(estimator=LogisticRegression(), n_features_to_select=28)
GaussianNB	0.936 316	0.926 864	0.936 316	0.931 03	RFE(estimator=SVC(), n_features_to_select=28)
GaussianNB	0.936 316	0.926 864	0.936 316	0.931 03	RFE(estimator=MLPClassifier(), n_features_to_select=28)
GaussianNB	0.936 316	0.926 864	0.936 316	0.931 03	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)

GaussianNB	0.936 316	0.926 864	0.936 316	0.931 03	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
GaussianNB	0.936 316	0.926 864	0.936 316	0.931 03	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
GaussianNB	0.936 316	0.926 864	0.936 316	0.931 03	RFECV(estimator=SVC(), min_features_to_select=28)
GaussianNB	0.936 316	0.926 864	0.936 316	0.931 03	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.939 23	0.926 766	0.939 23	0.931 828	SelectKBest(f_classif, k=28)
LinearDiscriminantAnalysis	0.939 23	0.926 766	0.939 23	0.931 828	SelectKBest(mutual_info_classif, k=28)
LinearDiscriminantAnalysis	0.939 23	0.926 766	0.939 23	0.931 828	SelectKBest(f_regression, k=28)
LinearDiscriminantAnalysis	0.939 23	0.926 766	0.939 23	0.931 828	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.939 23	0.926 766	0.939 23	0.931 828	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.939 23	0.926 766	0.939 23	0.931 828	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.939 23	0.926 766	0.939 23	0.931 828	RFE(estimator=SVC(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.939 23	0.926 766	0.939 23	0.931 828	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.939 23	0.926 766	0.939 23	0.931 828	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.939 23	0.926 766	0.939 23	0.931 828	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.939 23	0.926 766	0.939 23	0.931 828	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.939 23	0.926 766	0.939 23	0.931 828	RFECV(estimator=SVC(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.939 23	0.926 766	0.939 23	0.931 828	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.939 23	0.926 766	0.939 23	0.931 828	PCA(n_components=28)
LinearDiscriminantAnalysis	0.939 23	0.926 766	0.939 23	0.931 828	KernelPCA(kernel="linear", n_components=28)
LinearDiscriminantAnalysis	0.939 23	0.926 766	0.939 23	0.931 828	SparsePCA(n_components=28)
LinearDiscriminantAnalysis	0.939 23	0.926 766	0.939 23	0.931 828	MiniBatchSparsePCA(n_components=28)
MLPClassifier	0.944 433	0.926 761	0.944 433	0.931 717	KernelPCA(kernel="linear", n_components=28)

SGDClassifier	0.946 93	0.926 753	0.946 93	0.927 09	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LabelSpreading	0.946 722	0.926 651	0.946 722	0.928 204	KernelPCA(kernel="linear", n_components=28)
ExtraTreeClassifier	0.930 697	0.926 624	0.930 697	0.928 582	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LogisticRegression	0.946 306	0.926 543	0.946 306	0.929 395	KernelPCA(kernel="linear", n_components=28)
LogisticRegression	0.946 306	0.926 543	0.946 306	0.929 395	SparsePCA(n_components=28)
LogisticRegression	0.946 306	0.926 543	0.946 306	0.929 395	MiniBatchSparsePCA(n_components=28)
LabelPropagation	0.946 306	0.926 543	0.946 306	0.929 395	MiniBatchSparsePCA(n_components=28)
HistGradientBoostingClassifier	0.943 809	0.926 514	0.943 809	0.931 748	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
PassiveAggressiveClassifier	0.937 981	0.926 49	0.937 981	0.931 325	SelectKBest(mutual_info_classif, k=28)
KNeighborsClassifier	0.941 935	0.926 459	0.941 935	0.931 964	SelectKBest(f_classif, k=28)
SelfTrainingClassifier	0.941 935	0.926 459	0.941 935	0.931 964	SelectKBest(f_classif, k=28)
KNeighborsClassifier	0.941 935	0.926 459	0.941 935	0.931 964	SelectKBest(mutual_info_classif, k=28)
SelfTrainingClassifier	0.941 935	0.926 459	0.941 935	0.931 964	SelectKBest(mutual_info_classif, k=28)
KNeighborsClassifier	0.941 935	0.926 459	0.941 935	0.931 964	SelectKBest(f_regression, k=28)
SelfTrainingClassifier	0.941 935	0.926 459	0.941 935	0.931 964	SelectKBest(f_regression, k=28)
KNeighborsClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
SelfTrainingClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SelfTrainingClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFE(estimator=LogisticRegression(), n_features_to_select=28)
SelfTrainingClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFE(estimator=LogisticRegression(), n_features_to_select=28)
KNeighborsClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFE(estimator=SVC(), n_features_to_select=28)

SelfTrainingClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFE(estimator=SVC(), n_features_to_select=28)
KNeighborsClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFE(estimator=MLPClassifier() , n_features_to_select=28)
SelfTrainingClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFE(estimator=MLPClassifier() , n_features_to_select=28)
KNeighborsClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
SelfTrainingClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
KNeighborsClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFECV(estimator=SVC(), min_features_to_select=28)
SelfTrainingClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFECV(estimator=SVC(), min_features_to_select=28)
KNeighborsClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.941 935	0.926 459	0.941 935	0.931 964	PCA(n_components=28)
SelfTrainingClassifier	0.941 935	0.926 459	0.941 935	0.931 964	PCA(n_components=28)
HistGradientBoostingClassifier	0.945 265	0.926 441	0.945 265	0.930 834	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SGDClassifier	0.945 89	0.926 332	0.945 89	0.929 949	SelectKBest(f_classif, k=28)
MLPClassifier	0.945 473	0.926 291	0.945 473	0.930 465	RFECV(estimator=SVC(), min_features_to_select=28)
HistGradientBoostingClassifier	0.944 641	0.926 273	0.944 641	0.931 152	PCA(n_components=28)
MLPClassifier	0.944 433	0.926 249	0.944 433	0.931 25	MiniBatchSparsePCA(n_components=28)
GaussianNB	0.928 824	0.926 149	0.928 824	0.927 456	PCA(n_components=28)
HistGradientBoostingClassifier	0.943 392	0.926 039	0.943 392	0.931 467	RFE(estimator=LogisticRegression(), n_features_to_select=28)

LabelSpreading	0.946 514	0.926 022	0.946 514	0.928 075	SparsePCA(n_components=28)
HistGradientBoostingClassifier	0.943 809	0.925 986	0.943 809	0.931 296	RFECV(estimator=SVC(), min_features_to_select=28)
MLPClassifier	0.944 225	0.925 975	0.944 225	0.931 111	SparsePCA(n_components=28)
KNeighborsClassifier	0.945 057	0.925 857	0.945 057	0.930 445	KernelPCA(kernel="linear", n_components=28)
SelfTrainingClassifier	0.945 057	0.925 857	0.945 057	0.930 445	KernelPCA(kernel="linear", n_components=28)
KNeighborsClassifier	0.945 057	0.925 857	0.945 057	0.930 445	SparsePCA(n_components=28)
SelfTrainingClassifier	0.945 057	0.925 857	0.945 057	0.930 445	SparsePCA(n_components=28)
PassiveAggressiveClassifier	0.919 459	0.925 824	0.919 459	0.922 512	RFE(estimator=SVC(), n_features_to_select=28)
GaussianNB	0.928 2	0.925 638	0.928 2	0.926 891	KernelPCA(kernel="linear", n_components=28)
GaussianNB	0.928 2	0.925 638	0.928 2	0.926 891	SparsePCA(n_components=28)
PassiveAggressiveClassifier	0.936 524	0.925 612	0.936 524	0.930 328	RFECV(estimator=SVC(), min_features_to_select=28)
SGDClassifier	0.939 438	0.925 543	0.939 438	0.931 045	RFECV(estimator=SVC(), min_features_to_select=28)
SGDClassifier	0.939 23	0.925 393	0.939 23	0.930 903	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LabelPropagation	0.945 89	0.925 357	0.945 89	0.928 848	KernelPCA(kernel="linear", n_components=28)
LabelPropagation	0.945 89	0.925 357	0.945 89	0.928 848	SparsePCA(n_components=28)
SGDClassifier	0.941 311	0.925 356	0.941 311	0.931 129	PCA(n_components=28)
PassiveAggressiveClassifier	0.937 357	0.925 271	0.937 357	0.930 366	SelectKBest(f_classif, k=28)
MLPClassifier	0.943 6	0.925 189	0.943 6	0.930 694	RFE(estimator=LogisticRegression(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926 951	0.925 177	0.926 951	0.926 052	SelectKBest(f_classif, k=28)
QuadraticDiscriminantAnalysis	0.926 951	0.925 177	0.926 951	0.926 052	SelectKBest(mutual_info_classifier, k=28)
QuadraticDiscriminantAnalysis	0.926 951	0.925 177	0.926 951	0.926 052	SelectKBest(f_regression, k=28)
QuadraticDiscriminantAnalysis	0.926 951	0.925 177	0.926 951	0.926 052	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926 951	0.925 177	0.926 951	0.926 052	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926 951	0.925 177	0.926 951	0.926 052	RFE(estimator=LogisticRegression(), n_features_to_select=28)

QuadraticDiscriminantAnalysis	0.926 951	0.925 177	0.926 951	0.926 052	RFE(estimator=SVC(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926 951	0.925 177	0.926 951	0.926 052	RFE(estimator=MLPClassifier() , n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926 951	0.925 177	0.926 951	0.926 052	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926 951	0.925 177	0.926 951	0.926 052	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926 951	0.925 177	0.926 951	0.926 052	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926 951	0.925 177	0.926 951	0.926 052	RFECV(estimator=SVC(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926 951	0.925 177	0.926 951	0.926 052	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926 951	0.925 177	0.926 951	0.926 052	PCA(n_components=28)
QuadraticDiscriminantAnalysis	0.926 951	0.925 177	0.926 951	0.926 052	KernelPCA(kernel="linear", n_components=28)
QuadraticDiscriminantAnalysis	0.926 951	0.925 177	0.926 951	0.926 052	SparsePCA(n_components=28 )
QuadraticDiscriminantAnalysis	0.926 951	0.925 177	0.926 951	0.926 052	MiniBatchSparsePCA(n_comp onents=28)
VotingClassifier	0.945 473	0.925 007	0.945 473	0.929 138	MiniBatchSparsePCA(n_comp onents=28)
LabelSpreading	0.942 976	0.924 739	0.942 976	0.930 509	SelectKBest(f_classif, k=28)
LabelSpreading	0.942 976	0.924 739	0.942 976	0.930 509	SelectKBest(mutual_info_clas sif, k=28)
LabelSpreading	0.942 976	0.924 739	0.942 976	0.930 509	SelectKBest(f_regression, k=28)
LabelSpreading	0.942 976	0.924 739	0.942 976	0.930 509	RFE(estimator=DecisionTreeC lassifier(), n_features_to_select=28)
LabelSpreading	0.942 976	0.924 739	0.942 976	0.930 509	RFE(estimator=RandomForest Classifier(), n_features_to_select=28)
LabelSpreading	0.942 976	0.924 739	0.942 976	0.930 509	RFE(estimator=LogisticRegres sion(), n_features_to_select=28)
LabelSpreading	0.942 976	0.924 739	0.942 976	0.930 509	RFE(estimator=SVC(), n_features_to_select=28)
LabelSpreading	0.942 976	0.924 739	0.942 976	0.930 509	RFE(estimator=MLPClassifier() , n_features_to_select=28)
LabelSpreading	0.942 976	0.924 739	0.942 976	0.930 509	RFECV(estimator=DecisionTr eeClassifier(), min_features_to_select=28)
LabelSpreading	0.942 976	0.924 739	0.942 976	0.930 509	RFECV(estimator=RandomFo restClassifier(), min_features_to_select=28)

LabelSpreading	0.942 976	0.924 739	0.942 976	0.930 509	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LabelSpreading	0.942 976	0.924 739	0.942 976	0.930 509	RFECV(estimator=SVC(), min_features_to_select=28)
LabelSpreading	0.942 976	0.924 739	0.942 976	0.930 509	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LabelSpreading	0.942 976	0.924 739	0.942 976	0.930 509	PCA(n_components=28)
GradientBoostingClassifier	0.944 641	0.924 655	0.944 641	0.929 662	SelectKBest(f_classif, k=28)
GradientBoostingClassifier	0.944 641	0.924 655	0.944 641	0.929 662	SelectKBest(mutual_info_classif, k=28)
GradientBoostingClassifier	0.944 641	0.924 655	0.944 641	0.929 662	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.944 641	0.924 655	0.944 641	0.929 662	RFE(estimator=LogisticRegression(), n_features_to_select=28)
GradientBoostingClassifier	0.944 641	0.924 655	0.944 641	0.929 662	RFE(estimator=SVC(), n_features_to_select=28)
GradientBoostingClassifier	0.944 641	0.924 655	0.944 641	0.929 662	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
GradientBoostingClassifier	0.944 641	0.924 655	0.944 641	0.929 662	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
GradientBoostingClassifier	0.944 641	0.924 655	0.944 641	0.929 662	RFECV(estimator=SVC(), min_features_to_select=28)
GradientBoostingClassifier	0.944 641	0.924 655	0.944 641	0.929 662	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
SGDClassifier	0.943 392	0.924 652	0.943 392	0.930 32	SelectKBest(mutual_info_classif, k=28)
SGDClassifier	0.943 392	0.924 652	0.943 392	0.930 32	SelectKBest(f_regression, k=28)
VotingClassifier	0.944 225	0.924 6	0.944 225	0.929 898	PCA(n_components=28)
LabelPropagation	0.942 56	0.924 571	0.942 56	0.930 459	SelectKBest(f_classif, k=28)
LabelPropagation	0.942 56	0.924 571	0.942 56	0.930 459	SelectKBest(mutual_info_classif, k=28)
LabelPropagation	0.942 56	0.924 571	0.942 56	0.930 459	SelectKBest(f_regression, k=28)
LabelPropagation	0.942 56	0.924 571	0.942 56	0.930 459	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LabelPropagation	0.942 56	0.924 571	0.942 56	0.930 459	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LabelPropagation	0.942 56	0.924 571	0.942 56	0.930 459	RFE(estimator=LogisticRegression(), n_features_to_select=28)

LabelPropagation	0.942 56	0.924 571	0.942 56	0.930 459	RFE(estimator=SVC(), n_features_to_select=28)
LabelPropagation	0.942 56	0.924 571	0.942 56	0.930 459	RFE(estimator=MLPClassifier ( ), n_features_to_select=28)
LabelPropagation	0.942 56	0.924 571	0.942 56	0.930 459	RFECV(estimator=DecisionTr eeClassifier(), min_features_to_select=28)
LabelPropagation	0.942 56	0.924 571	0.942 56	0.930 459	RFECV(estimator=RandomFo restClassifier(), min_features_to_select=28)
LabelPropagation	0.942 56	0.924 571	0.942 56	0.930 459	RFECV(estimator=LogisticRe gression(), min_features_to_select=28)
LabelPropagation	0.942 56	0.924 571	0.942 56	0.930 459	RFECV(estimator=SVC(), min_features_to_select=28)
LabelPropagation	0.942 56	0.924 571	0.942 56	0.930 459	RFECV(estimator=MLPClassi fier(), min_features_to_select=28)
LabelPropagation	0.942 56	0.924 571	0.942 56	0.930 459	PCA(n_components=28)
PassiveAggressiveCl assifier	0.942 144	0.924 428	0.942 144	0.930 405	RFE(estimator=DecisionTreeC lassifier(), n_features_to_select=28)
PassiveAggressiveCl assifier	0.908 012	0.924 42	0.908 012	0.915 584	RFE(estimator=MLPClassifier ( ), n_features_to_select=28)
GradientBoostingCla ssifier	0.944 433	0.924 332	0.944 433	0.929 527	SelectKBest(f_regression, k=28)
GradientBoostingCla ssifier	0.944 433	0.924 332	0.944 433	0.929 527	RFE(estimator=RandomForest Classifier(), n_features_to_select=28)
GradientBoostingCla ssifier	0.944 433	0.924 332	0.944 433	0.929 527	RFE(estimator=MLPClassifier ( ), n_features_to_select=28)
GradientBoostingCla ssifier	0.944 433	0.924 332	0.944 433	0.929 527	RFECV(estimator=DecisionTr eeClassifier(), min_features_to_select=28)
HistGradientBoostin gClassifier	0.943 392	0.924 068	0.943 392	0.929 842	SelectKBest(f_classif, k=28)
PassiveAggressiveCl assifier	0.928 408	0.924 043	0.928 408	0.926 147	MiniBatchSparsePCA(n_comp onents=28)
VotingClassifier	0.944 225	0.924 016	0.944 225	0.929 391	SelectKBest(f_regression, k=28)
VotingClassifier	0.944 225	0.924 016	0.944 225	0.929 391	RFE(estimator=LogisticRegres sion(), n_features_to_select=28)
VotingClassifier	0.944 225	0.924 016	0.944 225	0.929 391	RFE(estimator=MLPClassifier ( ), n_features_to_select=28)
VotingClassifier	0.944 225	0.924 016	0.944 225	0.929 391	RFECV(estimator=DecisionTr eeClassifier(), min_features_to_select=28)
VotingClassifier	0.944 225	0.924 016	0.944 225	0.929 391	RFECV(estimator=LogisticRe gression(), min_features_to_select=28)
VotingClassifier	0.944 225	0.924 016	0.944 225	0.929 391	RFECV(estimator=SVC(), min_features_to_select=28)



VotingClassifier	0.944 017	0.924 01	0.944 017	0.929 51	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
VotingClassifier	0.944 017	0.924 01	0.944 017	0.929 51	RFE(estimator=SVC(), n_features_to_select=28)
VotingClassifier	0.944 017	0.924 01	0.944 017	0.929 51	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LogisticRegression	0.945 473	0.923 877	0.945 473	0.928 014	SelectKBest(f_classif, k=28)
AdaBoostClassifier	0.945 473	0.923 877	0.945 473	0.928 014	SelectKBest(f_classif, k=28)
LogisticRegression	0.945 473	0.923 877	0.945 473	0.928 014	SelectKBest(mutual_info_classif, k=28)
AdaBoostClassifier	0.945 473	0.923 877	0.945 473	0.928 014	SelectKBest(mutual_info_classif, k=28)
LogisticRegression	0.945 473	0.923 877	0.945 473	0.928 014	SelectKBest(f_regression, k=28)
AdaBoostClassifier	0.945 473	0.923 877	0.945 473	0.928 014	SelectKBest(f_regression, k=28)
LogisticRegression	0.945 473	0.923 877	0.945 473	0.928 014	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.945 473	0.923 877	0.945 473	0.928 014	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LogisticRegression	0.945 473	0.923 877	0.945 473	0.928 014	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.945 473	0.923 877	0.945 473	0.928 014	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LogisticRegression	0.945 473	0.923 877	0.945 473	0.928 014	RFE(estimator=LogisticRegression(), n_features_to_select=28)
AdaBoostClassifier	0.945 473	0.923 877	0.945 473	0.928 014	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LogisticRegression	0.945 473	0.923 877	0.945 473	0.928 014	RFE(estimator=SVC(), n_features_to_select=28)
AdaBoostClassifier	0.945 473	0.923 877	0.945 473	0.928 014	RFE(estimator=SVC(), n_features_to_select=28)
LogisticRegression	0.945 473	0.923 877	0.945 473	0.928 014	RFE(estimator=MLPClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.945 473	0.923 877	0.945 473	0.928 014	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LogisticRegression	0.945 473	0.923 877	0.945 473	0.928 014	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.945 473	0.923 877	0.945 473	0.928 014	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)

LogisticRegression	0.945 473	0.923 877	0.945 473	0.928 014	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.945 473	0.923 877	0.945 473	0.928 014	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LogisticRegression	0.945 473	0.923 877	0.945 473	0.928 014	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
AdaBoostClassifier	0.945 473	0.923 877	0.945 473	0.928 014	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LogisticRegression	0.945 473	0.923 877	0.945 473	0.928 014	RFECV(estimator=SVC(), min_features_to_select=28)
AdaBoostClassifier	0.945 473	0.923 877	0.945 473	0.928 014	RFECV(estimator=SVC(), min_features_to_select=28)
LogisticRegression	0.945 473	0.923 877	0.945 473	0.928 014	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.945 473	0.923 877	0.945 473	0.928 014	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LogisticRegression	0.945 473	0.923 877	0.945 473	0.928 014	PCA(n_components=28)
SGDClassifier	0.943 6	0.923 737	0.943 6	0.929 489	SparsePCA(n_components=28)
VotingClassifier	0.944 017	0.923 709	0.944 017	0.929 256	SelectKBest(f_classif, k=28)
VotingClassifier	0.944 017	0.923 709	0.944 017	0.929 256	SelectKBest(mutual_info_classif, k=28)
VotingClassifier	0.944 017	0.923 709	0.944 017	0.929 256	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
VotingClassifier	0.944 017	0.923 709	0.944 017	0.929 256	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
GaussianNB	0.926 951	0.923 494	0.926 951	0.925 176	MiniBatchSparsePCA(n_components=28)
SGDClassifier	0.941 311	0.923 309	0.941 311	0.929 628	KernelPCA(kernel="linear", n_components=28)
SGDClassifier	0.946 722	0.923 098	0.946 722	0.921 22	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
SGDClassifier	0.942 976	0.922 934	0.942 976	0.929 081	MiniBatchSparsePCA(n_components=28)
AdaBoostClassifier	0.942 56	0.921 763	0.942 56	0.928 314	KernelPCA(kernel="linear", n_components=28)
AdaBoostClassifier	0.942 56	0.921 763	0.942 56	0.928 314	SparsePCA(n_components=28)
PassiveAggressiveClassifier	0.936 941	0.921 745	0.936 941	0.928 002	SparsePCA(n_components=28)
PassiveAggressiveClassifier	0.920 083	0.921 539	0.920 083	0.920 805	KernelPCA(kernel="linear", n_components=28)

SGDClassifier	0.946 098	0.921 184	0.946 098	0.924 263	RFE(estimator=SVC(), n_features_to_select=28)
BernoulliNB	0.792 3	0.920 367	0.792 3	0.844 321	MiniBatchSparsePCA(n_comp onents=28)
SVC	0.943 809	0.919 854	0.943 809	0.926 403	SelectKBest(f_classif, k=28)
SVC	0.943 809	0.919 854	0.943 809	0.926 403	SelectKBest(mutual_info_clas sif, k=28)
SVC	0.943 809	0.919 854	0.943 809	0.926 403	SelectKBest(f_regression, k=28)
SVC	0.943 809	0.919 854	0.943 809	0.926 403	RFE(estimator=DecisionTreeC lassifier(), n_features_to_select=28)
SVC	0.943 809	0.919 854	0.943 809	0.926 403	RFE(estimator=RandomForest Classifier(), n_features_to_select=28)
SVC	0.943 809	0.919 854	0.943 809	0.926 403	RFE(estimator=LogisticRegres sion(), n_features_to_select=28)
SVC	0.943 809	0.919 854	0.943 809	0.926 403	RFE(estimator=SVC(), n_features_to_select=28)
SVC	0.943 809	0.919 854	0.943 809	0.926 403	RFE(estimator=MLPClassifier (), n_features_to_select=28)
SVC	0.943 809	0.919 854	0.943 809	0.926 403	RFECV(estimator=DecisionTr eeClassifier(), min_features_to_select=28)
SVC	0.943 809	0.919 854	0.943 809	0.926 403	RFECV(estimator=RandomFo restClassifier(), min_features_to_select=28)
SVC	0.943 809	0.919 854	0.943 809	0.926 403	RFECV(estimator=LogisticRe gression(), min_features_to_select=28)
SVC	0.943 809	0.919 854	0.943 809	0.926 403	RFECV(estimator=SVC(), min_features_to_select=28)
SVC	0.943 809	0.919 854	0.943 809	0.926 403	RFECV(estimator=MLPClassi fier(), min_features_to_select=28)
SVC	0.943 809	0.919 854	0.943 809	0.926 403	PCA(n_components=28)
SGDClassifier	0.945 89	0.919 628	0.945 89	0.923 795	RFE(estimator=LogisticRegres sion(), n_features_to_select=28)
BernoulliNB	0.867 638	0.919 611	0.867 638	0.890 431	PCA(n_components=28)
PassiveAggressiveCl assifier	0.942 768	0.919 45	0.942 768	0.926 606	RFE(estimator=RandomForest Classifier(), n_features_to_select=28)
SVC	0.944 017	0.919 374	0.944 017	0.925 942	KernelPCA(kernel="linear", n_components=28)
SVC	0.944 017	0.919 374	0.944 017	0.925 942	SparsePCA(n_components=28 )
BernoulliNB	0.776 483	0.917 857	0.776 483	0.833 817	KernelPCA(kernel="linear", n_components=28)

BernoulliNB	0.776 483	0.917 857	0.776 483	0.833 817	SparsePCA(n_components=28)
SVC	0.943 6	0.917 105	0.943 6	0.924 782	MiniBatchSparsePCA(n_components=28)
PassiveAggressiveClassifier	0.945 265	0.915 206	0.945 265	0.922 737	RFECV(estimator=MLPClassifier(), min_features_to_select=28)



## APPENDIX 7: VULNERABILITY FILE LEVEL RESULTS

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
RandomForestClassifier	0.952 767	0.933 185	0.952 767	0.934 382	MiniBatchSparsePCA(n_components=28)
MLPClassifier	0.952 702	0.932 658	0.952 702	0.934 027	MiniBatchSparsePCA(n_components=28)
KNeighborsClassifier	0.952 178	0.930 456	0.952 178	0.933 722	MiniBatchSparsePCA(n_components=28)
SelfTrainingClassifier	0.952 178	0.930 456	0.952 178	0.933 722	MiniBatchSparsePCA(n_components=28)
BaggingClassifier	0.951 721	0.929 007	0.951 721	0.933 56	MiniBatchSparsePCA(n_components=28)
XGBClassifier	0.953 16	0.934 976	0.953 16	0.933 42	MiniBatchSparsePCA(n_components=28)
StackingClassifier	0.953 749	0.941 71	0.953 749	0.933 298	MiniBatchSparsePCA(n_components=28)
LabelPropagation	0.953 094	0.934 575	0.953 094	0.932 587	MiniBatchSparsePCA(n_components=28)
LabelSpreading	0.953 291	0.937 074	0.953 291	0.932 462	MiniBatchSparsePCA(n_components=28)
LGBMClassifier	0.953 552	0.942 369	0.953 552	0.932 369	MiniBatchSparsePCA(n_components=28)
HistGradientBoostingClassifier	0.953 552	0.947 049	0.953 552	0.931 886	MiniBatchSparsePCA(n_components=28)
VotingClassifier	0.953 094	0.936 793	0.953 094	0.931 153	MiniBatchSparsePCA(n_components=28)
SVC	0.952 898	0.931 679	0.952 898	0.930 172	MiniBatchSparsePCA(n_components=28)
BernoulliNB	0.952 898	0.908 015	0.952 898	0.929 915	MiniBatchSparsePCA(n_components=28)
RidgeClassifier	0.952 898	0.908 015	0.952 898	0.929 915	MiniBatchSparsePCA(n_components=28)
RidgeClassifierCV	0.952 898	0.908 015	0.952 898	0.929 915	MiniBatchSparsePCA(n_components=28)
LogisticRegression	0.952 833	0.908 012	0.952 833	0.929 882	MiniBatchSparsePCA(n_components=28)
LinearDiscriminantAnalysis	0.951 917	0.914 869	0.951 917	0.929 801	MiniBatchSparsePCA(n_components=28)
AdaBoostClassifier	0.952 571	0.908	0.952 571	0.929 752	MiniBatchSparsePCA(n_components=28)
GradientBoostingClassifier	0.952 571	0.908	0.952 571	0.929 752	MiniBatchSparsePCA(n_components=28)
SGDClassifier	0.951 197	0.909 677	0.951 197	0.929 189	MiniBatchSparsePCA(n_components=28)
PassiveAggressiveClassifier	0.931 113	0.910 178	0.931 113	0.920 34	MiniBatchSparsePCA(n_components=28)
QuadraticDiscriminantAnalysis	0.917 441	0.916 56	0.917 441	0.917	MiniBatchSparsePCA(n_components=28)
ExtraTreeClassifier	0.909 59	0.919 416	0.909 59	0.914 379	MiniBatchSparsePCA(n_components=28)
DecisionTreeClassifier	0.904 88	0.917 969	0.904 88	0.911 234	MiniBatchSparsePCA(n_components=28)

GaussianNB	0.900 432	0.914 744	0.900 432	0.907 412	MiniBatchSparsePCA(n_comp onents=28)
XGBClassifier	0.954 076	0.940 06	0.954 076	0.935 788	PCA(n_components=28)
MLPClassifier	0.953 16	0.935 364	0.953 16	0.935 641	PCA(n_components=28)
StackingClassifier	0.953 487	0.936 785	0.953 487	0.935 226	PCA(n_components=28)
BaggingClassifier	0.952 048	0.931 448	0.952 048	0.934 869	PCA(n_components=28)
KNeighborsClassifier	0.953 225	0.935 437	0.953 225	0.934 862	PCA(n_components=28)
SelfTrainingClassifier	0.953 225	0.935 437	0.953 225	0.934 862	PCA(n_components=28)
RandomForestClassifier	0.952 506	0.932 545	0.952 506	0.934 743	PCA(n_components=28)
LGBMClassifier	0.954 337	0.946 339	0.954 337	0.934 43	PCA(n_components=28)
HistGradientBoostingClassifier	0.953 879	0.943 118	0.953 879	0.933 487	PCA(n_components=28)
LabelPropagation	0.952 833	0.932 442	0.952 833	0.932 785	PCA(n_components=28)
LabelSpreading	0.952 898	0.932 812	0.952 898	0.932 594	PCA(n_components=28)
VotingClassifier	0.953 356	0.940 998	0.953 356	0.931 781	PCA(n_components=28)
GradientBoostingClassifier	0.953 094	0.937 461	0.953 094	0.931 029	PCA(n_components=28)
SVC	0.953 094	0.941 95	0.953 094	0.930 653	PCA(n_components=28)
AdaBoostClassifier	0.952 898	0.931 735	0.952 898	0.930 3	PCA(n_components=28)
BernoulliNB	0.952 898	0.908 015	0.952 898	0.929 915	PCA(n_components=28)
RidgeClassifier	0.952 898	0.908 015	0.952 898	0.929 915	PCA(n_components=28)
RidgeClassifierCV	0.952 898	0.908 015	0.952 898	0.929 915	PCA(n_components=28)
LogisticRegression	0.952 767	0.908 009	0.952 767	0.929 85	PCA(n_components=28)
LinearDiscriminantAnalysis	0.951 851	0.914 56	0.951 851	0.929 768	PCA(n_components=28)
SGDClassifier	0.945 44	0.910 652	0.945 44	0.926 939	PCA(n_components=28)
QuadraticDiscriminantAnalysis	0.917 833	0.917 068	0.917 833	0.917 45	PCA(n_components=28)
ExtraTreeClassifier	0.911 946	0.918 971	0.911 946	0.915 394	PCA(n_components=28)
DecisionTreeClassifier	0.910 114	0.918 996	0.910 114	0.914 455	PCA(n_components=28)
GaussianNB	0.901 086	0.914 615	0.901 086	0.907 693	PCA(n_components=28)
PassiveAggressiveClassifier	0.851 106	0.915 312	0.851 106	0.880 618	PCA(n_components=28)

RandomForestClassifier	0.952 963	0.935 344	0.952 963	0.936 874	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
BaggingClassifier	0.952 178	0.933 483	0.952 178	0.936 656	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
XGBClassifier	0.954 795	0.944 824	0.954 795	0.936 433	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
StackingClassifier	0.954 337	0.941 642	0.954 337	0.936 05	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.953 225	0.935 437	0.953 225	0.934 862	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
SelfTrainingClassifier	0.953 225	0.935 437	0.953 225	0.934 862	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LGBMClassifier	0.954 272	0.946 636	0.954 272	0.934 167	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
HistGradientBoostingClassifier	0.953 945	0.943 551	0.953 945	0.933 639	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LabelPropagation	0.952 833	0.932 442	0.952 833	0.932 785	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
VotingClassifier	0.953 749	0.945 933	0.953 749	0.932 595	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LabelSpreading	0.952 898	0.932 812	0.952 898	0.932 594	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
MLPClassifier	0.953 094	0.934 575	0.953 094	0.932 587	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.952 963	0.933 776	0.952 963	0.930 837	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
SVC	0.953 094	0.941 95	0.953 094	0.930 653	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.952 898	0.908 015	0.952 898	0.929 915	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
RidgeClassifier	0.952 898	0.908 015	0.952 898	0.929 915	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.952 898	0.908 015	0.952 898	0.929 915	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
SGDClassifier	0.952 636	0.915 91	0.952 636	0.929 913	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)

LogisticRegression	0.952 767	0.908 009	0.952 767	0.929 85	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.951 851	0.914 56	0.951 851	0.929 768	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
PassiveAggressiveClassifier	0.924 572	0.914 815	0.924 572	0.919 579	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.917 833	0.917 068	0.917 833	0.917 45	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
ExtraTreeClassifier	0.914 366	0.920 57	0.914 366	0.917 411	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.908 282	0.920 494	0.908 282	0.914 189	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
GaussianNB	0.890 684	0.916 518	0.890 684	0.903 02	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
BernoulliNB	0.742 117	0.920 026	0.742 117	0.814 678	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
XGBClassifier	0.954 795	0.944 824	0.954 795	0.936 433	RFE(estimator=LogisticRegression(), n_features_to_select=28)
RandomForestClassifier	0.952 571	0.934 025	0.952 571	0.936 347	RFE(estimator=LogisticRegression(), n_features_to_select=28)
StackingClassifier	0.953 618	0.937 408	0.953 618	0.935 615	RFE(estimator=LogisticRegression(), n_features_to_select=28)
BaggingClassifier	0.951 328	0.930 165	0.951 328	0.934 919	RFE(estimator=LogisticRegression(), n_features_to_select=28)
KNeighborsClassifier	0.953 225	0.935 437	0.953 225	0.934 862	RFE(estimator=LogisticRegression(), n_features_to_select=28)
SelfTrainingClassifier	0.953 225	0.935 437	0.953 225	0.934 862	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LGBMClassifier	0.954 272	0.946 636	0.954 272	0.934 167	RFE(estimator=LogisticRegression(), n_features_to_select=28)
HistGradientBoostingClassifier	0.953 814	0.943 842	0.953 814	0.933 103	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LabelPropagation	0.952 833	0.932 442	0.952 833	0.932 785	RFE(estimator=LogisticRegression(), n_features_to_select=28)
VotingClassifier	0.953 749	0.945 933	0.953 749	0.932 595	RFE(estimator=LogisticRegression(), n_features_to_select=28)



LabelSpreading	0.952 898	0.932 812	0.952 898	0.932 594	RFE(estimator=LogisticRegression(), n_features_to_select=28)
MLPClassifier	0.952 506	0.929 794	0.952 506	0.932 262	RFE(estimator=LogisticRegression(), n_features_to_select=28)
GradientBoostingClassifier	0.952 833	0.930 388	0.952 833	0.930 769	RFE(estimator=LogisticRegression(), n_features_to_select=28)
SVC	0.953 094	0.941 95	0.953 094	0.930 653	RFE(estimator=LogisticRegression(), n_features_to_select=28)
AdaBoostClassifier	0.952 898	0.908 015	0.952 898	0.929 915	RFE(estimator=LogisticRegression(), n_features_to_select=28)
RidgeClassifier	0.952 898	0.908 015	0.952 898	0.929 915	RFE(estimator=LogisticRegression(), n_features_to_select=28)
RidgeClassifierCV	0.952 898	0.908 015	0.952 898	0.929 915	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LogisticRegression	0.952 767	0.908 009	0.952 767	0.929 85	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.951 851	0.914 56	0.951 851	0.929 768	RFE(estimator=LogisticRegression(), n_features_to_select=28)
SGDClassifier	0.952 178	0.911 662	0.952 178	0.929 682	RFE(estimator=LogisticRegression(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.917 833	0.917 068	0.917 833	0.917 45	RFE(estimator=LogisticRegression(), n_features_to_select=28)
ExtraTreeClassifier	0.913 319	0.921 777	0.913 319	0.917 437	RFE(estimator=LogisticRegression(), n_features_to_select=28)
DecisionTreeClassifier	0.907 89	0.920 533	0.907 89	0.913 999	RFE(estimator=LogisticRegression(), n_features_to_select=28)
PassiveAggressiveClassifier	0.911 88	0.914 316	0.911 88	0.913 093	RFE(estimator=LogisticRegression(), n_features_to_select=28)
GaussianNB	0.890 684	0.916 518	0.890 684	0.903 02	RFE(estimator=LogisticRegression(), n_features_to_select=28)
BernoulliNB	0.742 117	0.920 026	0.742 117	0.814 678	RFE(estimator=LogisticRegression(), n_features_to_select=28)
RandomForestClassifier	0.952 636	0.934 485	0.952 636	0.936 761	RFE(estimator=MLPClassifier(), n_features_to_select=28)
XGBClassifier	0.954 795	0.944 824	0.954 795	0.936 433	RFE(estimator=MLPClassifier(), n_features_to_select=28)
StackingClassifier	0.954 403	0.941 777	0.954 403	0.936 297	RFE(estimator=MLPClassifier(), n_features_to_select=28)

BaggingClassifier	0.952 178	0.932 796	0.952 178	0.935 915	RFE(estimator=MLPClassifier (), n_features_to_select=28)
KNeighborsClassifier	0.953 225	0.935 437	0.953 225	0.934 862	RFE(estimator=MLPClassifier (), n_features_to_select=28)
SelfTrainingClassifier	0.953 225	0.935 437	0.953 225	0.934 862	RFE(estimator=MLPClassifier (), n_features_to_select=28)
HistGradientBoostingClassifier	0.954 599	0.949 567	0.954 599	0.934 694	RFE(estimator=MLPClassifier (), n_features_to_select=28)
LGBMClassifier	0.954 272	0.946 636	0.954 272	0.934 167	RFE(estimator=MLPClassifier (), n_features_to_select=28)
LabelPropagation	0.952 833	0.932 442	0.952 833	0.932 785	RFE(estimator=MLPClassifier (), n_features_to_select=28)
LabelSpreading	0.952 898	0.932 812	0.952 898	0.932 594	RFE(estimator=MLPClassifier (), n_features_to_select=28)
VotingClassifier	0.953 683	0.946 636	0.953 683	0.932 32	RFE(estimator=MLPClassifier (), n_features_to_select=28)
GradientBoostingClassifier	0.953 029	0.935 892	0.953 029	0.930 871	RFE(estimator=MLPClassifier (), n_features_to_select=28)
SVC	0.953 094	0.941 95	0.953 094	0.930 653	RFE(estimator=MLPClassifier (), n_features_to_select=28)
MLPClassifier	0.952 767	0.928 535	0.952 767	0.930 611	RFE(estimator=MLPClassifier (), n_features_to_select=28)
AdaBoostClassifier	0.952 898	0.908 015	0.952 898	0.929 915	RFE(estimator=MLPClassifier (), n_features_to_select=28)
RidgeClassifier	0.952 898	0.908 015	0.952 898	0.929 915	RFE(estimator=MLPClassifier (), n_features_to_select=28)
RidgeClassifierCV	0.952 898	0.908 015	0.952 898	0.929 915	RFE(estimator=MLPClassifier (), n_features_to_select=28)
LogisticRegression	0.952 767	0.908 009	0.952 767	0.929 85	RFE(estimator=MLPClassifier (), n_features_to_select=28)
LinearDiscriminantAnalysis	0.951 851	0.914 56	0.951 851	0.929 768	RFE(estimator=MLPClassifier (), n_features_to_select=28)
SGDClassifier	0.952 506	0.907 997	0.952 506	0.929 719	RFE(estimator=MLPClassifier (), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.917 833	0.917 068	0.917 833	0.917 45	RFE(estimator=MLPClassifier (), n_features_to_select=28)
ExtraTreeClassifier	0.909 721	0.919 732	0.909 721	0.914 595	RFE(estimator=MLPClassifier (), n_features_to_select=28)
DecisionTreeClassifier	0.908 413	0.920 9	0.908 413	0.914 443	RFE(estimator=MLPClassifier (), n_features_to_select=28)
GaussianNB	0.890 684	0.916 518	0.890 684	0.903 02	RFE(estimator=MLPClassifier (), n_features_to_select=28)
PassiveAggressiveClassifier	0.802 564	0.917 364	0.802 564	0.852 567	RFE(estimator=MLPClassifier (), n_features_to_select=28)
BernoulliNB	0.742 117	0.920 026	0.742 117	0.814 678	RFE(estimator=MLPClassifier (), n_features_to_select=28)
RandomForestClassifier	0.953 16	0.935 981	0.953 16	0.937 091	RFE(estimator=RandomForest Classifier(), n_features_to_select=28)
XGBClassifier	0.954 795	0.944 824	0.954 795	0.936 433	RFE(estimator=RandomForest Classifier(), n_features_to_select=28)

StackingClassifier	0.954 206	0.940 351	0.954 206	0.936 381	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
BaggingClassifier	0.951 393	0.931 13	0.951 393	0.935 616	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.953 225	0.935 437	0.953 225	0.934 862	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SelfTrainingClassifier	0.953 225	0.935 437	0.953 225	0.934 862	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LGBMClassifier	0.954 272	0.946 636	0.954 272	0.934 167	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
HistGradientBoostingClassifier	0.954 01	0.943 467	0.954 01	0.933 904	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LabelPropagation	0.952 833	0.932 442	0.952 833	0.932 785	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
MLPClassifier	0.951 59	0.927 114	0.951 59	0.932 638	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LabelSpreading	0.952 898	0.932 812	0.952 898	0.932 594	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
VotingClassifier	0.953 683	0.945 408	0.953 683	0.932 44	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.952 963	0.933 776	0.952 963	0.930 837	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SVC	0.953 094	0.941 95	0.953 094	0.930 653	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.952 898	0.908 015	0.952 898	0.929 915	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
RidgeClassifier	0.952 898	0.908 015	0.952 898	0.929 915	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.952 898	0.908 015	0.952 898	0.929 915	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SGDClassifier	0.952 898	0.908 015	0.952 898	0.929 915	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LogisticRegression	0.952 767	0.908 009	0.952 767	0.929 85	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.951 851	0.914 56	0.951 851	0.929 768	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)

PassiveAggressiveClassifier	0.945 244	0.915 889	0.945 244	0.928 587	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.917 833	0.917 068	0.917 833	0.917 45	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
ExtraTreeClassifier	0.912 6	0.920 477	0.912 6	0.916 449	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.906 843	0.920 381	0.906 843	0.913 374	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
GaussianNB	0.890 684	0.916 518	0.890 684	0.903 02	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
BernoulliNB	0.742 117	0.920 026	0.742 117	0.814 678	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
RandomForestClassifier	0.952 963	0.935 617	0.952 963	0.937 335	RFE(estimator=SVC(), n_features_to_select=28)
BaggingClassifier	0.952 636	0.934 485	0.952 636	0.936 761	RFE(estimator=SVC(), n_features_to_select=28)
StackingClassifier	0.954 272	0.940 539	0.954 272	0.936 624	RFE(estimator=SVC(), n_features_to_select=28)
XGBClassifier	0.954 795	0.944 824	0.954 795	0.936 433	RFE(estimator=SVC(), n_features_to_select=28)
KNeighborsClassifier	0.953 225	0.935 437	0.953 225	0.934 862	RFE(estimator=SVC(), n_features_to_select=28)
SelfTrainingClassifier	0.953 225	0.935 437	0.953 225	0.934 862	RFE(estimator=SVC(), n_features_to_select=28)
MLPClassifier	0.953 291	0.935 76	0.953 291	0.934 478	RFE(estimator=SVC(), n_features_to_select=28)
LGBMClassifier	0.954 272	0.946 636	0.954 272	0.934 167	RFE(estimator=SVC(), n_features_to_select=28)
LabelPropagation	0.952 833	0.932 442	0.952 833	0.932 785	RFE(estimator=SVC(), n_features_to_select=28)
LabelSpreading	0.952 898	0.932 812	0.952 898	0.932 594	RFE(estimator=SVC(), n_features_to_select=28)
VotingClassifier	0.953 749	0.947 144	0.953 749	0.932 475	RFE(estimator=SVC(), n_features_to_select=28)
HistGradientBoostingClassifier	0.953 291	0.939 384	0.953 291	0.931 746	RFE(estimator=SVC(), n_features_to_select=28)
GradientBoostingClassifier	0.953 029	0.935 892	0.953 029	0.930 871	RFE(estimator=SVC(), n_features_to_select=28)
SVC	0.953 094	0.941 95	0.953 094	0.930 653	RFE(estimator=SVC(), n_features_to_select=28)
AdaBoostClassifier	0.952 898	0.908 015	0.952 898	0.929 915	RFE(estimator=SVC(), n_features_to_select=28)
RidgeClassifier	0.952 898	0.908 015	0.952 898	0.929 915	RFE(estimator=SVC(), n_features_to_select=28)
RidgeClassifierCV	0.952 898	0.908 015	0.952 898	0.929 915	RFE(estimator=SVC(), n_features_to_select=28)
SGDClassifier	0.952 898	0.908 015	0.952 898	0.929 915	RFE(estimator=SVC(), n_features_to_select=28)

LogisticRegression	0.952 767	0.908 009	0.952 767	0.929 85	RFE(estimator=SVC(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.951 851	0.914 56	0.951 851	0.929 768	RFE(estimator=SVC(), n_features_to_select=28)
ExtraTreeClassifier	0.914 955	0.921 764	0.914 955	0.918 285	RFE(estimator=SVC(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.917 833	0.917 068	0.917 833	0.917 45	RFE(estimator=SVC(), n_features_to_select=28)
DecisionTreeClassifier	0.908 02	0.920 262	0.908 02	0.913 944	RFE(estimator=SVC(), n_features_to_select=28)
PassiveAggressiveClassifier	0.902 787	0.918 382	0.902 787	0.910 315	RFE(estimator=SVC(), n_features_to_select=28)
GaussianNB	0.890 684	0.916 518	0.890 684	0.903 02	RFE(estimator=SVC(), n_features_to_select=28)
BernoulliNB	0.742 117	0.920 026	0.742 117	0.814 678	RFE(estimator=SVC(), n_features_to_select=28)
RandomForestClassifier	0.952 702	0.934 531	0.952 702	0.936 616	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
BaggingClassifier	0.952 178	0.933 316	0.952 178	0.936 474	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
XGBClassifier	0.954 795	0.944 824	0.954 795	0.936 433	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
StackingClassifier	0.953 749	0.937 988	0.953 749	0.936	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.953 225	0.935 437	0.953 225	0.934 862	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.953 225	0.935 437	0.953 225	0.934 862	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LGBMClassifier	0.954 272	0.946 636	0.954 272	0.934 167	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LabelPropagation	0.952 833	0.932 442	0.952 833	0.932 785	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
MLPClassifier	0.953 16	0.935 178	0.953 16	0.932 739	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LabelSpreading	0.952 898	0.932 812	0.952 898	0.932 594	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
VotingClassifier	0.953 683	0.945 408	0.953 683	0.932 44	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.953 552	0.943 205	0.953 552	0.932 249	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)

GradientBoostingClassifier	0.953029	0.935892	0.953029	0.930871	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
SVC	0.953094	0.94195	0.953094	0.930653	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.952898	0.908015	0.952898	0.929915	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
RidgeClassifier	0.952898	0.908015	0.952898	0.929915	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.952898	0.908015	0.952898	0.929915	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
SGDClassifier	0.952833	0.908012	0.952833	0.929882	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LogisticRegression	0.952767	0.908009	0.952767	0.92985	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.951851	0.91456	0.951851	0.929768	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
PassiveAggressiveClassifier	0.94616	0.910079	0.94616	0.927093	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.917833	0.917068	0.917833	0.91745	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
ExtraTreeClassifier	0.910506	0.920442	0.910506	0.915337	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.909329	0.91987	0.909329	0.914453	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
GaussianNB	0.890684	0.916518	0.890684	0.90302	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
BernoulliNB	0.742117	0.920026	0.742117	0.814678	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
RandomForestClassifier	0.952702	0.934597	0.952702	0.93671	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
XGBClassifier	0.954795	0.944824	0.954795	0.936433	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
StackingClassifier	0.954076	0.939864	0.954076	0.935995	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
BaggingClassifier	0.951982	0.932482	0.951982	0.935981	RFECV(estimator=LogisticRegression(), min_features_to_select=28)

KNeighborsClassifier	0.953 225	0.935 437	0.953 225	0.934 862	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
SelfTrainingClassifier	0.953 225	0.935 437	0.953 225	0.934 862	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LGBMClassifier	0.954 272	0.946 636	0.954 272	0.934 167	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
HistGradientBoostingClassifier	0.954 076	0.945 558	0.954 076	0.933 713	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LabelPropagation	0.952 833	0.932 442	0.952 833	0.932 785	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LabelSpreading	0.952 898	0.932 812	0.952 898	0.932 594	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
VotingClassifier	0.953 618	0.944 839	0.953 618	0.932 284	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
MLPClassifier	0.952 963	0.933 233	0.952 963	0.931 692	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
GradientBoostingClassifier	0.952 898	0.931 961	0.952 898	0.930 803	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
SVC	0.953 094	0.941 95	0.953 094	0.930 653	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
AdaBoostClassifier	0.952 898	0.908 015	0.952 898	0.929 915	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
RidgeClassifier	0.952 898	0.908 015	0.952 898	0.929 915	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
RidgeClassifierCV	0.952 898	0.908 015	0.952 898	0.929 915	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LogisticRegression	0.952 767	0.908 009	0.952 767	0.929 85	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.951 851	0.914 56	0.951 851	0.929 768	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
SGDClassifier	0.947 992	0.911 895	0.947 992	0.928 26	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.917 833	0.917 068	0.917 833	0.917 45	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
ExtraTreeClassifier	0.909 721	0.920 224	0.909 721	0.914 823	RFECV(estimator=LogisticRegression(), min_features_to_select=28)

DecisionTreeClassifier	0.908675	0.921036	0.908675	0.914644	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
GaussianNB	0.890684	0.916518	0.890684	0.90302	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
PassiveAggressiveClassifier	0.88931	0.914008	0.88931	0.901208	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
BernoulliNB	0.742117	0.920026	0.742117	0.814678	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
RandomForestClassifier	0.952833	0.934783	0.952833	0.93651	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
XGBClassifier	0.954795	0.944824	0.954795	0.936433	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
StackingClassifier	0.953879	0.938808	0.953879	0.935773	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
BaggingClassifier	0.95087	0.929688	0.95087	0.935016	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.953225	0.935437	0.953225	0.934862	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.953225	0.935437	0.953225	0.934862	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LGBMClassifier	0.954272	0.946636	0.954272	0.934167	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.953945	0.947293	0.953945	0.933058	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LabelPropagation	0.952833	0.932442	0.952833	0.932785	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LabelSpreading	0.952898	0.932812	0.952898	0.932594	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
VotingClassifier	0.953749	0.947144	0.953749	0.932475	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
MLPClassifier	0.953225	0.936224	0.953225	0.932426	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
GradientBoostingClassifier	0.952963	0.933776	0.952963	0.930837	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
SVC	0.953094	0.94195	0.953094	0.930653	RFECV(estimator=MLPClassifier(), min_features_to_select=28)



AdaBoostClassifier	0.952 898	0.908 015	0.952 898	0.929 915	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
RidgeClassifier	0.952 898	0.908 015	0.952 898	0.929 915	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.952 898	0.908 015	0.952 898	0.929 915	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
SGDClassifier	0.952 375	0.915 955	0.952 375	0.929 908	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LogisticRegression	0.952 767	0.908 009	0.952 767	0.929 85	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.951 851	0.914 56	0.951 851	0.929 768	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
PassiveAggressiveClassifier	0.938 44	0.916 595	0.938 44	0.926 575	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
ExtraTreeClassifier	0.914 759	0.922 609	0.914 759	0.918 582	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.917 833	0.917 068	0.917 833	0.917 45	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.908 086	0.920 078	0.908 086	0.913 894	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
GaussianNB	0.890 684	0.916 518	0.890 684	0.903 02	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
BernoulliNB	0.742 117	0.920 026	0.742 117	0.814 678	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
RandomForestClassifier	0.952 506	0.934 139	0.952 506	0.936 679	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
XGBClassifier	0.954 795	0.944 824	0.954 795	0.936 433	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
BaggingClassifier	0.952 244	0.932 882	0.952 244	0.935 861	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
StackingClassifier	0.953 618	0.937 432	0.953 618	0.935 512	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.953 225	0.935 437	0.953 225	0.934 862	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.953 225	0.935 437	0.953 225	0.934 862	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)

LGBMClassifier	0.954 272	0.946 636	0.954 272	0.934 167	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LabelPropagation	0.952 833	0.932 442	0.952 833	0.932 785	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
VotingClassifier	0.953 814	0.947 612	0.953 814	0.932 631	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LabelSpreading	0.952 898	0.932 812	0.952 898	0.932 594	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.953 683	0.950 056	0.953 683	0.932 078	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
MLPClassifier	0.952 898	0.932 074	0.952 898	0.931 05	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
GradientBoostingClassifier	0.952 963	0.933 776	0.952 963	0.930 837	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
SVC	0.953 094	0.941 95	0.953 094	0.930 653	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.952 898	0.908 015	0.952 898	0.929 915	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
RidgeClassifier	0.952 898	0.908 015	0.952 898	0.929 915	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.952 898	0.908 015	0.952 898	0.929 915	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
SGDClassifier	0.952 898	0.908 015	0.952 898	0.929 915	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LogisticRegression	0.952 767	0.908 009	0.952 767	0.929 85	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.951 851	0.914 56	0.951 851	0.929 768	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.917 833	0.917 068	0.917 833	0.917 45	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
ExtraTreeClassifier	0.912 404	0.920 048	0.912 404	0.916 144	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.909 46	0.920 962	0.909 46	0.915 026	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
GaussianNB	0.890 684	0.916 518	0.890 684	0.903 02	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)

PassiveAggressiveClassifier	0.845022	0.918362	0.845022	0.878035	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
BernoulliNB	0.742117	0.920026	0.742117	0.814678	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
RandomForestClassifier	0.953356	0.936699	0.953356	0.937493	RFECV(estimator=SVC(), min_features_to_select=28)
BaggingClassifier	0.952636	0.934417	0.952636	0.936669	RFECV(estimator=SVC(), min_features_to_select=28)
XGBClassifier	0.954795	0.944824	0.954795	0.936433	RFECV(estimator=SVC(), min_features_to_select=28)
StackingClassifier	0.954076	0.940171	0.954076	0.935684	RFECV(estimator=SVC(), min_features_to_select=28)
KNeighborsClassifier	0.953225	0.935437	0.953225	0.934862	RFECV(estimator=SVC(), min_features_to_select=28)
SelfTrainingClassifier	0.953225	0.935437	0.953225	0.934862	RFECV(estimator=SVC(), min_features_to_select=28)
HistGradientBoostingClassifier	0.954206	0.944595	0.954206	0.934355	RFECV(estimator=SVC(), min_features_to_select=28)
LGBMClassifier	0.954272	0.946636	0.954272	0.934167	RFECV(estimator=SVC(), min_features_to_select=28)
LabelPropagation	0.952833	0.932442	0.952833	0.932785	RFECV(estimator=SVC(), min_features_to_select=28)
LabelSpreading	0.952898	0.932812	0.952898	0.932594	RFECV(estimator=SVC(), min_features_to_select=28)
MLPClassifier	0.953225	0.936077	0.953225	0.932543	RFECV(estimator=SVC(), min_features_to_select=28)
VotingClassifier	0.953618	0.944839	0.953618	0.932284	RFECV(estimator=SVC(), min_features_to_select=28)
GradientBoostingClassifier	0.953029	0.935892	0.953029	0.930871	RFECV(estimator=SVC(), min_features_to_select=28)
SVC	0.953094	0.94195	0.953094	0.930653	RFECV(estimator=SVC(), min_features_to_select=28)
AdaBoostClassifier	0.952898	0.908015	0.952898	0.929915	RFECV(estimator=SVC(), min_features_to_select=28)
RidgeClassifier	0.952898	0.908015	0.952898	0.929915	RFECV(estimator=SVC(), min_features_to_select=28)
RidgeClassifierCV	0.952898	0.908015	0.952898	0.929915	RFECV(estimator=SVC(), min_features_to_select=28)
LogisticRegression	0.952767	0.908009	0.952767	0.92985	RFECV(estimator=SVC(), min_features_to_select=28)
SGDClassifier	0.952375	0.912758	0.952375	0.929781	RFECV(estimator=SVC(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.951851	0.91456	0.951851	0.929768	RFECV(estimator=SVC(), min_features_to_select=28)
PassiveAggressiveClassifier	0.948515	0.910556	0.948515	0.928187	RFECV(estimator=SVC(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.917833	0.917068	0.917833	0.91745	RFECV(estimator=SVC(), min_features_to_select=28)
DecisionTreeClassifier	0.908936	0.921075	0.908936	0.914801	RFECV(estimator=SVC(), min_features_to_select=28)
ExtraTreeClassifier	0.909329	0.91898	0.909329	0.914038	RFECV(estimator=SVC(), min_features_to_select=28)

GaussianNB	0.890 684	0.916 518	0.890 684	0.903 02	RFECV(estimator=SVC(), min_features_to_select=28)
BernoulliNB	0.742 117	0.920 026	0.742 117	0.814 678	RFECV(estimator=SVC(), min_features_to_select=28)
RandomForestClassifier	0.952 898	0.935 153	0.952 898	0.936 833	SelectKBest(f_classif, k=28)
XGBClassifier	0.954 795	0.944 824	0.954 795	0.936 433	SelectKBest(f_classif, k=28)
BaggingClassifier	0.951 786	0.931 911	0.951 786	0.935 766	SelectKBest(f_classif, k=28)
StackingClassifier	0.953 094	0.934 867	0.953 094	0.934 992	SelectKBest(f_classif, k=28)
KNeighborsClassifier	0.953 225	0.935 437	0.953 225	0.934 862	SelectKBest(f_classif, k=28)
SelfTrainingClassifier	0.953 225	0.935 437	0.953 225	0.934 862	SelectKBest(f_classif, k=28)
LGBMClassifier	0.954 272	0.946 636	0.954 272	0.934 167	SelectKBest(f_classif, k=28)
MLPClassifier	0.953 356	0.936 966	0.953 356	0.933 078	SelectKBest(f_classif, k=28)
LabelPropagation	0.952 833	0.932 442	0.952 833	0.932 785	SelectKBest(f_classif, k=28)
HistGradientBoostingClassifier	0.953 749	0.944 925	0.953 749	0.932 714	SelectKBest(f_classif, k=28)
VotingClassifier	0.953 749	0.945 933	0.953 749	0.932 595	SelectKBest(f_classif, k=28)
LabelSpreading	0.952 898	0.932 812	0.952 898	0.932 594	SelectKBest(f_classif, k=28)
GradientBoostingClassifier	0.953 029	0.935 892	0.953 029	0.930 871	SelectKBest(f_classif, k=28)
SVC	0.953 094	0.941 95	0.953 094	0.930 653	SelectKBest(f_classif, k=28)
AdaBoostClassifier	0.952 898	0.908 015	0.952 898	0.929 915	SelectKBest(f_classif, k=28)
RidgeClassifier	0.952 898	0.908 015	0.952 898	0.929 915	SelectKBest(f_classif, k=28)
RidgeClassifierCV	0.952 898	0.908 015	0.952 898	0.929 915	SelectKBest(f_classif, k=28)
LogisticRegression	0.952 767	0.908 009	0.952 767	0.929 85	SelectKBest(f_classif, k=28)
LinearDiscriminantAnalysis	0.951 851	0.914 56	0.951 851	0.929 768	SelectKBest(f_classif, k=28)
PassiveAggressiveClassifier	0.949 169	0.916 399	0.949 169	0.929 663	SelectKBest(f_classif, k=28)
SGDClassifier	0.943 609	0.913 114	0.943 609	0.927 007	SelectKBest(f_classif, k=28)
QuadraticDiscriminantAnalysis	0.917 833	0.917 068	0.917 833	0.917 45	SelectKBest(f_classif, k=28)
ExtraTreeClassifier	0.912 273	0.921 505	0.912 273	0.916 761	SelectKBest(f_classif, k=28)
DecisionTreeClassifier	0.906 647	0.920 16	0.906 647	0.913 169	SelectKBest(f_classif, k=28)
GaussianNB	0.890 684	0.916 518	0.890 684	0.903 02	SelectKBest(f_classif, k=28)

BernoulliNB	0.742 117	0.920 026	0.742 117	0.814 678	SelectKBest(f_classif, k=28)
RandomForestClassifier	0.953 16	0.936 026	0.953 16	0.937 184	SelectKBest(f_regression, k=28)
XGBClassifier	0.954 795	0.944 824	0.954 795	0.936 433	SelectKBest(f_regression, k=28)
StackingClassifier	0.953 945	0.938 99	0.953 945	0.936 12	SelectKBest(f_regression, k=28)
BaggingClassifier	0.952 178	0.932 706	0.952 178	0.935 82	SelectKBest(f_regression, k=28)
KNeighborsClassifier	0.953 225	0.935 437	0.953 225	0.934 862	SelectKBest(f_regression, k=28)
SelfTrainingClassifier	0.953 225	0.935 437	0.953 225	0.934 862	SelectKBest(f_regression, k=28)
HistGradientBoostingClassifier	0.954 337	0.947 664	0.954 337	0.934 204	SelectKBest(f_regression, k=28)
LGBMClassifier	0.954 272	0.946 636	0.954 272	0.934 167	SelectKBest(f_regression, k=28)
LabelPropagation	0.952 833	0.932 442	0.952 833	0.932 785	SelectKBest(f_regression, k=28)
MLPClassifier	0.953 487	0.939 561	0.953 487	0.932 688	SelectKBest(f_regression, k=28)
LabelSpreading	0.952 898	0.932 812	0.952 898	0.932 594	SelectKBest(f_regression, k=28)
VotingClassifier	0.953 683	0.945 408	0.953 683	0.932 44	SelectKBest(f_regression, k=28)
GradientBoostingClassifier	0.952 963	0.933 776	0.952 963	0.930 837	SelectKBest(f_regression, k=28)
SVC	0.953 094	0.941 95	0.953 094	0.930 653	SelectKBest(f_regression, k=28)
AdaBoostClassifier	0.952 898	0.908 015	0.952 898	0.929 915	SelectKBest(f_regression, k=28)
RidgeClassifier	0.952 898	0.908 015	0.952 898	0.929 915	SelectKBest(f_regression, k=28)
RidgeClassifierCV	0.952 898	0.908 015	0.952 898	0.929 915	SelectKBest(f_regression, k=28)
SGDClassifier	0.952 898	0.908 015	0.952 898	0.929 915	SelectKBest(f_regression, k=28)
LogisticRegression	0.952 767	0.908 009	0.952 767	0.929 85	SelectKBest(f_regression, k=28)
LinearDiscriminantAnalysis	0.951 851	0.914 56	0.951 851	0.929 768	SelectKBest(f_regression, k=28)
PassiveAggressiveClassifier	0.935 824	0.912 46	0.935 824	0.923 538	SelectKBest(f_regression, k=28)
QuadraticDiscriminantAnalysis	0.917 833	0.917 068	0.917 833	0.917 45	SelectKBest(f_regression, k=28)
ExtraTreeClassifier	0.908 805	0.919 891	0.908 805	0.914 187	SelectKBest(f_regression, k=28)
DecisionTreeClassifier	0.908 544	0.920 047	0.908 544	0.914 122	SelectKBest(f_regression, k=28)
GaussianNB	0.890 684	0.916 518	0.890 684	0.903 02	SelectKBest(f_regression, k=28)
BernoulliNB	0.742 117	0.920 026	0.742 117	0.814 678	SelectKBest(f_regression, k=28)

BaggingClassifier	0.952 178	0.933 4	0.952 178	0.936 565	SelectKBest(mutual_info_classification, k=28)
RandomForestClassifier	0.952 767	0.934 652	0.952 767	0.936 564	SelectKBest(mutual_info_classification, k=28)
XGBClassifier	0.954 795	0.944 824	0.954 795	0.936 433	SelectKBest(mutual_info_classification, k=28)
StackingClassifier	0.953 356	0.936 111	0.953 356	0.935 044	SelectKBest(mutual_info_classification, k=28)
KNeighborsClassifier	0.953 225	0.935 437	0.953 225	0.934 862	SelectKBest(mutual_info_classification, k=28)
SelfTrainingClassifier	0.953 225	0.935 437	0.953 225	0.934 862	SelectKBest(mutual_info_classification, k=28)
LGBMClassifier	0.954 272	0.946 636	0.954 272	0.934 167	SelectKBest(mutual_info_classification, k=28)
LabelPropagation	0.952 833	0.932 442	0.952 833	0.932 785	SelectKBest(mutual_info_classification, k=28)
HistGradientBoostingClassifier	0.953 421	0.938 609	0.953 421	0.932 652	SelectKBest(mutual_info_classification, k=28)
VotingClassifier	0.953 749	0.945 933	0.953 749	0.932 595	SelectKBest(mutual_info_classification, k=28)
LabelSpreading	0.952 898	0.932 812	0.952 898	0.932 594	SelectKBest(mutual_info_classification, k=28)
MLPClassifier	0.953 291	0.937 074	0.953 291	0.932 462	SelectKBest(mutual_info_classification, k=28)
GradientBoostingClassifier	0.953 029	0.935 892	0.953 029	0.930 871	SelectKBest(mutual_info_classification, k=28)
SVC	0.953 094	0.941 95	0.953 094	0.930 653	SelectKBest(mutual_info_classification, k=28)
AdaBoostClassifier	0.952 898	0.908 015	0.952 898	0.929 915	SelectKBest(mutual_info_classification, k=28)
RidgeClassifier	0.952 898	0.908 015	0.952 898	0.929 915	SelectKBest(mutual_info_classification, k=28)
RidgeClassifierCV	0.952 898	0.908 015	0.952 898	0.929 915	SelectKBest(mutual_info_classification, k=28)
LogisticRegression	0.952 767	0.908 009	0.952 767	0.929 85	SelectKBest(mutual_info_classification, k=28)
SGDClassifier	0.952 702	0.908 006	0.952 702	0.929 817	SelectKBest(mutual_info_classification, k=28)
LinearDiscriminantAnalysis	0.951 851	0.914 56	0.951 851	0.929 768	SelectKBest(mutual_info_classification, k=28)
PassiveAggressiveClassifier	0.947 403	0.911 051	0.947 403	0.927 844	SelectKBest(mutual_info_classification, k=28)
QuadraticDiscriminantAnalysis	0.917 833	0.917 068	0.917 833	0.917 45	SelectKBest(mutual_info_classification, k=28)
ExtraTreeClassifier	0.911 88	0.922 118	0.911 88	0.916 838	SelectKBest(mutual_info_classification, k=28)
DecisionTreeClassifier	0.907 105	0.920 226	0.907 105	0.913 442	SelectKBest(mutual_info_classification, k=28)
GaussianNB	0.890 684	0.916 518	0.890 684	0.903 02	SelectKBest(mutual_info_classification, k=28)
BernoulliNB	0.742 117	0.920 026	0.742 117	0.814 678	SelectKBest(mutual_info_classification, k=28)
MLPClassifier	0.950 87	0.931 093	0.950 87	0.936 101	SparsePCA(n_components=28)

XGBClassifier	0.954 076	0.939 864	0.954 076	0.935 995	SparsePCA(n_components=28 )
RandomForestClassifier	0.952 571	0.933 113	0.952 571	0.935 185	SparsePCA(n_components=28 )
StackingClassifier	0.953 487	0.936 911	0.953 487	0.934 7	SparsePCA(n_components=28 )
BaggingClassifier	0.951 786	0.930 582	0.951 786	0.934 613	SparsePCA(n_components=28 )
KNeighborsClassifier	0.952 702	0.933 054	0.952 702	0.934 551	SparsePCA(n_components=28 )
SelfTrainingClassifier	0.952 702	0.933 054	0.952 702	0.934 551	SparsePCA(n_components=28 )
LGBMClassifier	0.954 272	0.946 636	0.954 272	0.934 167	SparsePCA(n_components=28 )
LabelPropagation	0.953 291	0.936 211	0.953 291	0.933 156	SparsePCA(n_components=28 )
LabelSpreading	0.953 356	0.937 485	0.953 356	0.932 732	SparsePCA(n_components=28 )
HistGradientBoostingClassifier	0.953 749	0.948 622	0.953 749	0.932 355	SparsePCA(n_components=28 )
VotingClassifier	0.953 421	0.940 344	0.953 421	0.932 178	SparsePCA(n_components=28 )
SVC	0.953 356	0.943 198	0.953 356	0.931 537	SparsePCA(n_components=28 )
GradientBoostingClassifier	0.953 16	0.938 03	0.953 16	0.931 311	SparsePCA(n_components=28 )
AdaBoostClassifier	0.952 833	0.923 769	0.952 833	0.930 011	SparsePCA(n_components=28 )
BernoulliNB	0.952 898	0.908 015	0.952 898	0.929 915	SparsePCA(n_components=28 )
RidgeClassifier	0.952 898	0.908 015	0.952 898	0.929 915	SparsePCA(n_components=28 )
RidgeClassifierCV	0.952 898	0.908 015	0.952 898	0.929 915	SparsePCA(n_components=28 )
LogisticRegression	0.952 833	0.908 012	0.952 833	0.929 882	SparsePCA(n_components=28 )
LinearDiscriminantAnalysis	0.951 851	0.914 56	0.951 851	0.929 768	SparsePCA(n_components=28 )
SGDClassifier	0.952 309	0.912 327	0.952 309	0.929 748	SparsePCA(n_components=28 )
QuadraticDiscriminantAnalysis	0.917 833	0.917 068	0.917 833	0.917 45	SparsePCA(n_components=28 )
ExtraTreeClassifier	0.910 31	0.918 925	0.910 31	0.914 523	SparsePCA(n_components=28 )
DecisionTreeClassifier	0.908 871	0.918 415	0.908 871	0.913 533	SparsePCA(n_components=28 )
PassiveAggressiveClassifier	0.906 058	0.911 219	0.906 058	0.908 621	SparsePCA(n_components=28 )
GaussianNB	0.901 086	0.914 615	0.901 086	0.907 693	SparsePCA(n_components=28 )

## APPENDIX 8: AUTOML EXPERIMENT RESULTS

Model	ROC AUC	Precision	Recall	F1 Score	Feature Selection Model
PassiveAggressiveClassifier	0.592519	0.80084	0.201055	0.321417	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
PassiveAggressiveClassifier	0.638139	0.684023	0.324262	0.43996	SelectKBest(mutual_info_classif, k=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	SelectKBest(f_classif, k=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	SelectKBest(mutual_info_classif, k=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	SelectKBest(f_regression, k=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	RFE(estimator=LogisticRegression(), n_features_to_select=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	RFE(estimator=SVC(), n_features_to_select=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	RFE(estimator=MLPClassifier(), n_features_to_select=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	RFECV(estimator=SVC(), min_features_to_select=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
SVC	0.669785	0.670056	0.403165	0.503425	SparsePCA(n_components=4)
QuadraticDiscriminantAnalysis	0.601693	0.668611	0.241772	0.355129	SelectKBest(f_classif, k=4)
QuadraticDiscriminantAnalysis	0.601693	0.668611	0.241772	0.355129	SelectKBest(mutual_info_classif, k=4)
QuadraticDiscriminantAnalysis	0.601693	0.668611	0.241772	0.355129	SelectKBest(f_regression, k=4)
QuadraticDiscriminantAnalysis	0.601693	0.668611	0.241772	0.355129	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
QuadraticDiscriminantAnalysis	0.601693	0.668611	0.241772	0.355129	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
QuadraticDiscriminantAnalysis	0.601693	0.668611	0.241772	0.355129	RFE(estimator=LogisticRegression(), n_features_to_select=4)
QuadraticDiscriminantAnalysis	0.601693	0.668611	0.241772	0.355129	RFE(estimator=SVC(), n_features_to_select=4)
QuadraticDiscriminantAnalysis	0.601693	0.668611	0.241772	0.355129	RFE(estimator=MLPClassifier(), n_features_to_select=4)



QuadraticDiscriminantAnalysis	0.601 693	0.66 8611	0.24 1772	0.35 5129	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
QuadraticDiscriminantAnalysis	0.601 693	0.66 8611	0.24 1772	0.35 5129	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
QuadraticDiscriminantAnalysis	0.601 693	0.66 8611	0.24 1772	0.35 5129	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
QuadraticDiscriminantAnalysis	0.601 693	0.66 8611	0.24 1772	0.35 5129	RFECV(estimator=SVC(), min_features_to_select=4)
QuadraticDiscriminantAnalysis	0.601 693	0.66 8611	0.24 1772	0.35 5129	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
QuadraticDiscriminantAnalysis	0.601 693	0.66 8611	0.24 1772	0.35 5129	PCA(n_components=4)
QuadraticDiscriminantAnalysis	0.601 693	0.66 8611	0.24 1772	0.35 5129	SparsePCA(n_components=4)
QuadraticDiscriminantAnalysis	0.601 693	0.66 8611	0.24 1772	0.35 5129	MiniBatchSparsePCA(n_components=4)
PassiveAggressiveClassifier	0.632 999	0.66 5339	0.31 7089	0.42 949	SelectKBest(f_classif, k=4)
SVC	0.672 37	0.66 5301	0.41 097	0.50 8086	SelectKBest(f_classif, k=4)
SVC	0.672 37	0.66 5301	0.41 097	0.50 8086	SelectKBest(mutual_info_classif, k=4)
SVC	0.672 37	0.66 5301	0.41 097	0.50 8086	SelectKBest(f_regression, k=4)
SVC	0.672 37	0.66 5301	0.41 097	0.50 8086	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
SVC	0.672 37	0.66 5301	0.41 097	0.50 8086	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
SVC	0.672 37	0.66 5301	0.41 097	0.50 8086	RFE(estimator=LogisticRegression(), n_features_to_select=4)
SVC	0.672 37	0.66 5301	0.41 097	0.50 8086	RFE(estimator=SVC(), n_features_to_select=4)
SVC	0.672 37	0.66 5301	0.41 097	0.50 8086	RFE(estimator=MLPClassifier(), n_features_to_select=4)
SVC	0.672 37	0.66 5301	0.41 097	0.50 8086	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
SVC	0.672 37	0.66 5301	0.41 097	0.50 8086	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
SVC	0.672 37	0.66 5301	0.41 097	0.50 8086	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
SVC	0.672 37	0.66 5301	0.41 097	0.50 8086	RFECV(estimator=SVC(), min_features_to_select=4)
SVC	0.672 37	0.66 5301	0.41 097	0.50 8086	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
SVC	0.672 37	0.66 5301	0.41 097	0.50 8086	PCA(n_components=4)
SVC	0.671 417	0.66 4837	0.40 8861	0.50 6336	MiniBatchSparsePCA(n_components=4)
VotingClassifier	0.677 701	0.65 8624	0.42 616	0.51 7484	RFE(estimator=LogisticRegression(), n_features_to_select=4)
VotingClassifier	0.677 351	0.65 8075	0.42 5527	0.51 6848	RFE(estimator=SVC(), n_features_to_select=4)
VotingClassifier	0.677 6	0.65 798	0.42 616	0.51 7286	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)

VotingClassifier	0.677 815	0.65 7672	0.42 6793	0.51 7656	SelectKBest(f_regression, k=4)
VotingClassifier	0.675 938	0.65 7471	0.42 2363	0.51 4322	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
VotingClassifier	0.675 942	0.65 6937	0.42 2574	0.51 4315	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
VotingClassifier	0.676 191	0.65 6843	0.42 3207	0.51 4755	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
VotingClassifier	0.677 68	0.65 6818	0.42 6793	0.51 7391	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
VotingClassifier	0.676 937	0.65 6566	0.42 5105	0.51 6071	RFECV(estimator=SVC(), min_features_to_select=4)
VotingClassifier	0.677 718	0.65 6503	0.42 7004	0.51 7449	RFE(estimator=MLPClassifier(), n_features_to_select=4)
VotingClassifier	0.677 363	0.65 6484	0.42 616	0.51 6822	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
VotingClassifier	0.677 507	0.65 628	0.42 6582	0.51 7069	SelectKBest(mutual_info_classif, k=4)
BernoulliNB	0.639 789	0.65 5825	0.33 6076	0.44 4413	SelectKBest(f_classif, k=4)
BernoulliNB	0.639 789	0.65 5825	0.33 6076	0.44 4413	SelectKBest(mutual_info_classif, k=4)
BernoulliNB	0.639 789	0.65 5825	0.33 6076	0.44 4413	SelectKBest(f_regression, k=4)
BernoulliNB	0.639 789	0.65 5825	0.33 6076	0.44 4413	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
BernoulliNB	0.639 789	0.65 5825	0.33 6076	0.44 4413	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
BernoulliNB	0.639 789	0.65 5825	0.33 6076	0.44 4413	RFE(estimator=LogisticRegression(), n_features_to_select=4)
BernoulliNB	0.639 789	0.65 5825	0.33 6076	0.44 4413	RFE(estimator=SVC(), n_features_to_select=4)
BernoulliNB	0.639 789	0.65 5825	0.33 6076	0.44 4413	RFE(estimator=MLPClassifier(), n_features_to_select=4)
BernoulliNB	0.639 789	0.65 5825	0.33 6076	0.44 4413	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
BernoulliNB	0.639 789	0.65 5825	0.33 6076	0.44 4413	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
BernoulliNB	0.639 789	0.65 5825	0.33 6076	0.44 4413	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
BernoulliNB	0.639 789	0.65 5825	0.33 6076	0.44 4413	RFECV(estimator=SVC(), min_features_to_select=4)
BernoulliNB	0.639 789	0.65 5825	0.33 6076	0.44 4413	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
VotingClassifier	0.677 334	0.65 5743	0.42 6371	0.51 6748	SelectKBest(f_classif, k=4)
VotingClassifier	0.680 688	0.63 6226	0.44 2405	0.52 1901	MiniBatchSparsePCA(n_components=4)
VotingClassifier	0.677 172	0.62 845	0.43 7131	0.51 5615	SparsePCA(n_components=4)
VotingClassifier	0.677 075	0.62 7421	0.43 7342	0.51 5415	PCA(n_components=4)
StackingClassifier	0.645 378	0.62 6794	0.35 9283	0.45 6752	RFE(estimator=LogisticRegression(), n_features_to_select=4)

StackingClassifier	0.642 188	0.62 5986	0.35 1688	0.45 0358	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
StackingClassifier	0.649 182	0.62 5984	0.36 8987	0.46 4295	SelectKBest(mutual_info_classif, k=4)
StackingClassifier	0.642 863	0.62 5934	0.35 3376	0.45 1726	SparsePCA(n_components=4)
StackingClassifier	0.642 724	0.62 5561	0.35 3165	0.45 1456	RFE(estimator=MLPClassifier(), n_features_to_select=4)
StackingClassifier	0.648 55	0.62 5179	0.36 7722	0.46 3071	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
SGDClassifier	0.603 531	0.62 5	0.25 6329	0.36 3555	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
RidgeClassifierCV	0.648 09	0.62 4955	0.36 6667	0.46 2173	SparsePCA(n_components=4)
RidgeClassifierCV	0.649 191	0.62 4911	0.36 9409	0.46 4333	SelectKBest(f_classif, k=4)
RidgeClassifierCV	0.649 191	0.62 4911	0.36 9409	0.46 4333	SelectKBest(mutual_info_classif, k=4)
RidgeClassifierCV	0.649 191	0.62 4911	0.36 9409	0.46 4333	SelectKBest(f_regression, k=4)
RidgeClassifierCV	0.649 191	0.62 4911	0.36 9409	0.46 4333	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
RidgeClassifierCV	0.649 191	0.62 4911	0.36 9409	0.46 4333	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
RidgeClassifierCV	0.649 191	0.62 4911	0.36 9409	0.46 4333	RFE(estimator=LogisticRegression(), n_features_to_select=4)
RidgeClassifierCV	0.649 191	0.62 4911	0.36 9409	0.46 4333	RFE(estimator=SVC(), n_features_to_select=4)
RidgeClassifierCV	0.649 191	0.62 4911	0.36 9409	0.46 4333	RFE(estimator=MLPClassifier(), n_features_to_select=4)
RidgeClassifierCV	0.649 191	0.62 4911	0.36 9409	0.46 4333	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
RidgeClassifierCV	0.649 191	0.62 4911	0.36 9409	0.46 4333	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
RidgeClassifierCV	0.649 191	0.62 4911	0.36 9409	0.46 4333	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
RidgeClassifierCV	0.649 191	0.62 4911	0.36 9409	0.46 4333	RFECV(estimator=SVC(), min_features_to_select=4)
RidgeClassifierCV	0.649 191	0.62 4911	0.36 9409	0.46 4333	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
RidgeClassifierCV	0.649 191	0.62 4911	0.36 9409	0.46 4333	PCA(n_components=4)
StackingClassifier	0.648 161	0.62 4865	0.36 6878	0.46 2316	SelectKBest(f_classif, k=4)
StackingClassifier	0.646 891	0.62 378	0.36 4135	0.45 9837	SelectKBest(f_regression, k=4)
StackingClassifier	0.642 631	0.62 3744	0.35 3586	0.45 1326	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
RidgeClassifierCV	0.648 136	0.62 3567	0.36 73	0.46 2294	MiniBatchSparsePCA(n_components=4)
StackingClassifier	0.639 189	0.62 3476	0.34 5148	0.44 4324	PCA(n_components=4)
StackingClassifier	0.644 883	0.62 172	0.35 9916	0.45 5906	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)

StackingClassifier	0.645 672	0.62 0752	0.36 2236	0.45 7501	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
StackingClassifier	0.652 961	0.61 9636	0.38 0802	0.47 171	RFE(estimator=SVC(), n_features_to_select=4)
StackingClassifier	0.647 566	0.61 9268	0.36 7511	0.46 1274	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
StackingClassifier	0.646 895	0.61 8759	0.36 6034	0.45 9968	RFECV(estimator=SVC(), min_features_to_select=4)
RidgeClassifier	0.652 298	0.61 8132	0.37 9747	0.47 0465	SparsePCA(n_components=4)
RidgeClassifier	0.652 37	0.61 8051	0.37 9958	0.47 0604	SelectKBest(f_classif, k=4)
RidgeClassifier	0.652 37	0.61 8051	0.37 9958	0.47 0604	SelectKBest(mutual_info_classif, k=4)
RidgeClassifier	0.652 37	0.61 8051	0.37 9958	0.47 0604	SelectKBest(f_regression, k=4)
RidgeClassifier	0.652 37	0.61 8051	0.37 9958	0.47 0604	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
RidgeClassifier	0.652 37	0.61 8051	0.37 9958	0.47 0604	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
RidgeClassifier	0.652 37	0.61 8051	0.37 9958	0.47 0604	RFE(estimator=LogisticRegression(), n_features_to_select=4)
RidgeClassifier	0.652 37	0.61 8051	0.37 9958	0.47 0604	RFE(estimator=SVC(), n_features_to_select=4)
RidgeClassifier	0.652 37	0.61 8051	0.37 9958	0.47 0604	RFE(estimator=MLPClassifier(), n_features_to_select=4)
RidgeClassifier	0.652 37	0.61 8051	0.37 9958	0.47 0604	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
RidgeClassifier	0.652 37	0.61 8051	0.37 9958	0.47 0604	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
RidgeClassifier	0.652 37	0.61 8051	0.37 9958	0.47 0604	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
RidgeClassifier	0.652 37	0.61 8051	0.37 9958	0.47 0604	RFECV(estimator=SVC(), min_features_to_select=4)
RidgeClassifier	0.652 37	0.61 8051	0.37 9958	0.47 0604	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
RidgeClassifier	0.652 37	0.61 8051	0.37 9958	0.47 0604	PCA(n_components=4)
RidgeClassifier	0.652 336	0.61 7839	0.37 9958	0.47 0542	MiniBatchSparsePCA(n_components=4)
MLPClassifier	0.699 653	0.61 7194	0.49 8312	0.55 1418	MiniBatchSparsePCA(n_components=4)
MLPClassifier	0.694 506	0.61 6827	0.48 5654	0.54 3437	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
StackingClassifier	0.651 568	0.61 6203	0.37 8692	0.46 9097	MiniBatchSparsePCA(n_components=4)
MLPClassifier	0.699 243	0.61 5645	0.49 8101	0.55 0671	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
PassiveAggressiveClassifier	0.624 402	0.61 5096	0.31 1181	0.41 3281	RFE(estimator=MLPClassifier(), n_features_to_select=4)
MLPClassifier	0.698 754	0.61 4923	0.49 7257	0.54 9866	RFE(estimator=LogisticRegression(), n_features_to_select=4)
PassiveAggressiveClassifier	0.671 368	0.61 4571	0.42 8903	0.50 5219	RFE(estimator=SVC(), n_features_to_select=4)

MLPClassifier	0.697 382	0.61 3613	0.49 4515	0.54 7664	RFE(estimator=MLPClassifier(), n_features_to_select=4)
MLPClassifier	0.696 353	0.61 3523	0.49 1983	0.54 6072	RFE(estimator=SVC(), n_features_to_select=4)
MLPClassifier	0.697 956	0.61 3139	0.49 6203	0.54 8507	RFECV(estimator=SVC(), min_features_to_select=4)
MLPClassifier	0.696 471	0.61 248	0.49 2827	0.54 6177	PCA(n_components=4)
MLPClassifier	0.704 278	0.61 0123	0.51 3713	0.55 7783	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
MLPClassifier	0.693 619	0.61 0098	0.48 692	0.54 1593	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
MLPClassifier	0.697 689	0.60 8348	0.49 8101	0.54 7732	SparsePCA(n_components=4)
MLPClassifier	0.697 157	0.60 8213	0.49 6835	0.54 6911	SelectKBest(f_regression, k=4)
MLPClassifier	0.695 774	0.60 8006	0.49 346	0.54 4777	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
MLPClassifier	0.698 014	0.60 7914	0.49 9156	0.54 8193	SelectKBest(f_classif, k=4)
MLPClassifier	0.692 758	0.60 7595	0.48 6076	0.54 0084	SelectKBest(mutual_info_classif, k=4)
MLPClassifier	0.701 546	0.60 0544	0.51 2236	0.55 2886	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
PassiveAggressiveClassifier	0.616 886	0.59 6876	0.29 8312	0.39 7806	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
LinearDiscriminantAnalysis	0.668 638	0.59 2261	0.43 27	0.50 0061	SelectKBest(f_classif, k=4)
LinearDiscriminantAnalysis	0.668 638	0.59 2261	0.43 27	0.50 0061	SelectKBest(mutual_info_classif, k=4)
LinearDiscriminantAnalysis	0.668 638	0.59 2261	0.43 27	0.50 0061	SelectKBest(f_regression, k=4)
LinearDiscriminantAnalysis	0.668 638	0.59 2261	0.43 27	0.50 0061	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
LinearDiscriminantAnalysis	0.668 638	0.59 2261	0.43 27	0.50 0061	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
LinearDiscriminantAnalysis	0.668 638	0.59 2261	0.43 27	0.50 0061	RFE(estimator=LogisticRegression(), n_features_to_select=4)
LinearDiscriminantAnalysis	0.668 638	0.59 2261	0.43 27	0.50 0061	RFE(estimator=SVC(), n_features_to_select=4)
LinearDiscriminantAnalysis	0.668 638	0.59 2261	0.43 27	0.50 0061	RFE(estimator=MLPClassifier(), n_features_to_select=4)
LinearDiscriminantAnalysis	0.668 638	0.59 2261	0.43 27	0.50 0061	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
LinearDiscriminantAnalysis	0.668 638	0.59 2261	0.43 27	0.50 0061	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
LinearDiscriminantAnalysis	0.668 638	0.59 2261	0.43 27	0.50 0061	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
LinearDiscriminantAnalysis	0.668 638	0.59 2261	0.43 27	0.50 0061	RFECV(estimator=SVC(), min_features_to_select=4)
LinearDiscriminantAnalysis	0.668 638	0.59 2261	0.43 27	0.50 0061	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
LinearDiscriminantAnalysis	0.668 638	0.59 2261	0.43 27	0.50 0061	PCA(n_components=4)

LinearDiscriminantAnalysis	0.668 638	0.59 2261	0.43 27	0.50 0061	SparsePCA(n_components=4)
LinearDiscriminantAnalysis	0.668 638	0.59 2261	0.43 27	0.50 0061	MiniBatchSparsePCA(n_components=4)
LogisticRegression	0.673 658	0.57 9061	0.45 2743	0.50 817	SparsePCA(n_components=4)
LogisticRegression	0.673 696	0.57 8862	0.45 2954	0.50 8226	MiniBatchSparsePCA(n_components=4)
LogisticRegression	0.673 885	0.57 7873	0.45 4008	0.50 8507	SelectKBest(f_classif, k=4)
LogisticRegression	0.673 885	0.57 7873	0.45 4008	0.50 8507	SelectKBest(mutual_info_classif, k=4)
LogisticRegression	0.673 885	0.57 7873	0.45 4008	0.50 8507	SelectKBest(f_regression, k=4)
LogisticRegression	0.673 885	0.57 7873	0.45 4008	0.50 8507	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
LogisticRegression	0.673 885	0.57 7873	0.45 4008	0.50 8507	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
LogisticRegression	0.673 885	0.57 7873	0.45 4008	0.50 8507	RFE(estimator=LogisticRegression(), n_features_to_select=4)
LogisticRegression	0.673 885	0.57 7873	0.45 4008	0.50 8507	RFE(estimator=SVC(), n_features_to_select=4)
LogisticRegression	0.673 885	0.57 7873	0.45 4008	0.50 8507	RFE(estimator=MLPClassifier(), n_features_to_select=4)
LogisticRegression	0.673 885	0.57 7873	0.45 4008	0.50 8507	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
LogisticRegression	0.673 885	0.57 7873	0.45 4008	0.50 8507	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
LogisticRegression	0.673 885	0.57 7873	0.45 4008	0.50 8507	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
LogisticRegression	0.673 885	0.57 7873	0.45 4008	0.50 8507	RFECV(estimator=SVC(), min_features_to_select=4)
LogisticRegression	0.673 885	0.57 7873	0.45 4008	0.50 8507	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
LogisticRegression	0.673 885	0.57 7873	0.45 4008	0.50 8507	PCA(n_components=4)
PassiveAggressiveClassifier	0.575 128	0.57 6828	0.19 6414	0.29 3044	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
GaussianNB	0.651 09	0.57 5284	0.39 5781	0.46 8941	PCA(n_components=4)
GaussianNB	0.651 09	0.57 5284	0.39 5781	0.46 8941	SparsePCA(n_components=4)
LabelPropagation	0.688 89	0.56 8477	0.49 9156	0.53 1566	SparsePCA(n_components=4)
LabelSpreading	0.688 89	0.56 8477	0.49 9156	0.53 1566	SparsePCA(n_components=4)
LabelSpreading	0.689 641	0.56 8285	0.50 1266	0.53 2676	MiniBatchSparsePCA(n_components=4)
LabelPropagation	0.689 607	0.56 8149	0.50 1266	0.53 2616	MiniBatchSparsePCA(n_components=4)
LabelPropagation	0.696 232	0.56 3268	0.52 2152	0.54 1931	SelectKBest(mutual_info_classif, k=4)
LabelSpreading	0.696 232	0.56 3268	0.52 2152	0.54 1931	SelectKBest(mutual_info_classif, k=4)

LabelPropagation	0.695 27	0.56 3229	0.51 962	0.54 0546	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
LabelSpreading	0.695 27	0.56 3229	0.51 962	0.54 0546	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
LabelPropagation	0.695 27	0.56 3229	0.51 962	0.54 0546	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
LabelSpreading	0.695 27	0.56 3229	0.51 962	0.54 0546	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
LabelPropagation	0.695 27	0.56 3229	0.51 962	0.54 0546	RFE(estimator=LogisticRegression(), n_features_to_select=4)
LabelSpreading	0.695 27	0.56 3229	0.51 962	0.54 0546	RFE(estimator=LogisticRegression(), n_features_to_select=4)
LabelPropagation	0.695 342	0.56 32	0.51 9831	0.54 0647	SelectKBest(f_regression, k=4)
LabelSpreading	0.695 342	0.56 32	0.51 9831	0.54 0647	SelectKBest(f_regression, k=4)
LabelPropagation	0.695 455	0.56 2728	0.52 0464	0.54 0772	PCA(n_components=4)
LabelSpreading	0.695 455	0.56 2728	0.52 0464	0.54 0772	PCA(n_components=4)
LabelSpreading	0.696 505	0.56 161	0.52 4051	0.54 2181	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
LabelSpreading	0.696 505	0.56 161	0.52 4051	0.54 2181	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
LabelSpreading	0.696 505	0.56 161	0.52 4051	0.54 2181	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
LabelSpreading	0.696 505	0.56 161	0.52 4051	0.54 2181	RFECV(estimator=SVC(), min_features_to_select=4)
LabelSpreading	0.696 505	0.56 161	0.52 4051	0.54 2181	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
LabelPropagation	0.696 471	0.56 1483	0.52 4051	0.54 2121	RFE(estimator=SVC(), n_features_to_select=4)
LabelSpreading	0.696 471	0.56 1483	0.52 4051	0.54 2121	RFE(estimator=SVC(), n_features_to_select=4)
LabelPropagation	0.696 471	0.56 1483	0.52 4051	0.54 2121	RFE(estimator=MLPClassifier(), n_features_to_select=4)
LabelSpreading	0.696 471	0.56 1483	0.52 4051	0.54 2121	RFE(estimator=MLPClassifier(), n_features_to_select=4)
LabelPropagation	0.696 471	0.56 1483	0.52 4051	0.54 2121	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
LabelPropagation	0.696 471	0.56 1483	0.52 4051	0.54 2121	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
LabelPropagation	0.696 471	0.56 1483	0.52 4051	0.54 2121	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
LabelPropagation	0.696 471	0.56 1483	0.52 4051	0.54 2121	RFECV(estimator=SVC(), min_features_to_select=4)
LabelPropagation	0.696 471	0.56 1483	0.52 4051	0.54 2121	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
LabelPropagation	0.697 437	0.56 125	0.52 6793	0.54 3476	SelectKBest(f_classif, k=4)
LabelSpreading	0.697 437	0.56 125	0.52 6793	0.54 3476	SelectKBest(f_classif, k=4)
GradientBoostingClassifier	0.686 464	0.55 3389	0.50 2954	0.52 6967	PCA(n_components=4)

GradientBoostingClassifier	0.685 835	0.55 2485	0.50 1899	0.52 5978	SparsePCA(n_components=4)
SGDClassifier	0.682 923	0.54 9125	0.49 6414	0.52 144	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
GaussianNB	0.641 652	0.54 8406	0.38 481	0.45 2269	MiniBatchSparsePCA(n_components=4)
KNeighborsClassifier	0.687 832	0.54 7738	0.51 0759	0.52 8603	MiniBatchSparsePCA(n_components=4)
SelfTrainingClassifier	0.687 832	0.54 7738	0.51 0759	0.52 8603	MiniBatchSparsePCA(n_components=4)
SGDClassifier	0.641 951	0.54 7221	0.38 6287	0.45 2882	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
GradientBoostingClassifier	0.686 65	0.54 3434	0.51 0759	0.52 6591	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
GradientBoostingClassifier	0.686 616	0.54 3312	0.51 0759	0.52 6533	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
GradientBoostingClassifier	0.686 582	0.54 319	0.51 0759	0.52 6476	SelectKBest(f_classif, k=4)
GradientBoostingClassifier	0.686 654	0.54 3171	0.51 097	0.52 6579	SelectKBest(mutual_info_classif, k=4)
GradientBoostingClassifier	0.686 62	0.54 3049	0.51 097	0.52 6522	RFE(estimator=LogisticRegression(), n_features_to_select=4)
GradientBoostingClassifier	0.686 514	0.54 2947	0.51 0759	0.52 6362	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
GradientBoostingClassifier	0.686 514	0.54 2947	0.51 0759	0.52 6362	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
GradientBoostingClassifier	0.686 586	0.54 2928	0.51 097	0.52 6465	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
GradientBoostingClassifier	0.686 409	0.54 2844	0.51 0549	0.52 6201	RFE(estimator=MLPClassifier(), n_features_to_select=4)
GradientBoostingClassifier	0.686 375	0.54 2723	0.51 0549	0.52 6144	RFE(estimator=SVC(), n_features_to_select=4)
GradientBoostingClassifier	0.686 375	0.54 2723	0.51 0549	0.52 6144	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
GradientBoostingClassifier	0.686 447	0.54 2703	0.51 0759	0.52 6247	SelectKBest(f_regression, k=4)
GradientBoostingClassifier	0.686 413	0.54 2582	0.51 0759	0.52 619	RFECV(estimator=SVC(), min_features_to_select=4)
AdaBoostClassifier	0.679 289	0.54 157	0.49 1983	0.51 5587	PCA(n_components=4)
AdaBoostClassifier	0.679 289	0.54 157	0.49 1983	0.51 5587	SparsePCA(n_components=4)
KNeighborsClassifier	0.693 587	0.54 0506	0.53 2068	0.53 6253	RFE(estimator=SVC(), n_features_to_select=4)
SelfTrainingClassifier	0.693 587	0.54 0506	0.53 2068	0.53 6253	RFE(estimator=SVC(), n_features_to_select=4)
KNeighborsClassifier	0.693 587	0.54 0506	0.53 2068	0.53 6253	RFE(estimator=MLPClassifier(), n_features_to_select=4)
SelfTrainingClassifier	0.693 587	0.54 0506	0.53 2068	0.53 6253	RFE(estimator=MLPClassifier(), n_features_to_select=4)
KNeighborsClassifier	0.693 587	0.54 0506	0.53 2068	0.53 6253	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
SelfTrainingClassifier	0.693 587	0.54 0506	0.53 2068	0.53 6253	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)



KNeighborsClassifier	0.693 553	0.54 039	0.53 2068	0.53 6196	SelectKBest(f_classif, k=4)
SelfTrainingClassifier	0.693 553	0.54 039	0.53 2068	0.53 6196	SelectKBest(f_classif, k=4)
KNeighborsClassifier	0.693 553	0.54 039	0.53 2068	0.53 6196	SelectKBest(f_regression, k=4)
SelfTrainingClassifier	0.693 553	0.54 039	0.53 2068	0.53 6196	SelectKBest(f_regression, k=4)
KNeighborsClassifier	0.693 519	0.54 0274	0.53 2068	0.53 6139	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
SelfTrainingClassifier	0.693 519	0.54 0274	0.53 2068	0.53 6139	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
KNeighborsClassifier	0.693 519	0.54 0274	0.53 2068	0.53 6139	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
SelfTrainingClassifier	0.693 519	0.54 0274	0.53 2068	0.53 6139	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
KNeighborsClassifier	0.693 519	0.54 0274	0.53 2068	0.53 6139	RFE(estimator=LogisticRegression(), n_features_to_select=4)
SelfTrainingClassifier	0.693 519	0.54 0274	0.53 2068	0.53 6139	RFE(estimator=LogisticRegression(), n_features_to_select=4)
KNeighborsClassifier	0.694 32	0.53 9345	0.53 5021	0.53 7174	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
SelfTrainingClassifier	0.694 32	0.53 9345	0.53 5021	0.53 7174	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
KNeighborsClassifier	0.694 32	0.53 9345	0.53 5021	0.53 7174	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
SelfTrainingClassifier	0.694 32	0.53 9345	0.53 5021	0.53 7174	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
KNeighborsClassifier	0.694 32	0.53 9345	0.53 5021	0.53 7174	RFECV(estimator=SVC(), min_features_to_select=4)
SelfTrainingClassifier	0.694 32	0.53 9345	0.53 5021	0.53 7174	RFECV(estimator=SVC(), min_features_to_select=4)
KNeighborsClassifier	0.694 32	0.53 9345	0.53 5021	0.53 7174	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
SelfTrainingClassifier	0.694 32	0.53 9345	0.53 5021	0.53 7174	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
KNeighborsClassifier	0.694 286	0.53 923	0.53 5021	0.53 7117	SelectKBest(mutual_info_classif, k=4)
SelfTrainingClassifier	0.694 286	0.53 923	0.53 5021	0.53 7117	SelectKBest(mutual_info_classif, k=4)
KNeighborsClassifier	0.687 752	0.53 7017	0.51 8776	0.52 7739	SparsePCA(n_components=4)
SelfTrainingClassifier	0.687 752	0.53 7017	0.51 8776	0.52 7739	SparsePCA(n_components=4)
SGDClassifier	0.695 259	0.53 659	0.53 9873	0.53 8227	SelectKBest(f_classif, k=4)
RandomForestClassifier	0.682 939	0.53 6465	0.50 5907	0.52 0738	PCA(n_components=4)
RandomForestClassifier	0.680 885	0.53 5392	0.50 1055	0.51 7655	SparsePCA(n_components=4)
KNeighborsClassifier	0.692 997	0.53 3124	0.53 6498	0.53 4805	PCA(n_components=4)
SelfTrainingClassifier	0.692 997	0.53 3124	0.53 6498	0.53 4805	PCA(n_components=4)

AdaBoostClassifier	0.695 745	0.52 735	0.54 9156	0.53 8032	SelectKBest(f_classif, k=4)
AdaBoostClassifier	0.695 745	0.52 735	0.54 9156	0.53 8032	SelectKBest(mutual_info_classif, k=4)
AdaBoostClassifier	0.695 745	0.52 735	0.54 9156	0.53 8032	SelectKBest(f_regression, k=4)
AdaBoostClassifier	0.695 745	0.52 735	0.54 9156	0.53 8032	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
AdaBoostClassifier	0.695 745	0.52 735	0.54 9156	0.53 8032	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
AdaBoostClassifier	0.695 745	0.52 735	0.54 9156	0.53 8032	RFE(estimator=LogisticRegression(), n_features_to_select=4)
AdaBoostClassifier	0.695 745	0.52 735	0.54 9156	0.53 8032	RFE(estimator=SVC(), n_features_to_select=4)
AdaBoostClassifier	0.695 745	0.52 735	0.54 9156	0.53 8032	RFE(estimator=MLPClassifier(), n_features_to_select=4)
AdaBoostClassifier	0.695 745	0.52 735	0.54 9156	0.53 8032	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
AdaBoostClassifier	0.695 745	0.52 735	0.54 9156	0.53 8032	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
AdaBoostClassifier	0.695 745	0.52 735	0.54 9156	0.53 8032	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
AdaBoostClassifier	0.695 745	0.52 735	0.54 9156	0.53 8032	RFECV(estimator=SVC(), min_features_to_select=4)
AdaBoostClassifier	0.695 745	0.52 735	0.54 9156	0.53 8032	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
SGDClassifier	0.723 537	0.52 4789	0.62 9747	0.57 2497	RFE(estimator=MLPClassifier(), n_features_to_select=4)
BaggingClassifier	0.672 95	0.52 2297	0.48 9241	0.50 5229	RFECV(estimator=SVC(), min_features_to_select=4)
RandomForestClassifier	0.683 7	0.52 1115	0.52 0675	0.52 0895	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
RandomForestClassifier	0.675 248	0.51 9709	0.49 789	0.50 8566	RFECV(estimator=SVC(), min_features_to_select=4)
RandomForestClassifier	0.676 525	0.51 8269	0.50 2743	0.51 0388	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
RandomForestClassifier	0.678 684	0.51 7574	0.50 9494	0.51 3502	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
LGBMClassifier	0.666 74	0.51 7202	0.47 5738	0.49 5604	SelectKBest(f_classif, k=4)
LGBMClassifier	0.666 74	0.51 7202	0.47 5738	0.49 5604	SelectKBest(mutual_info_classif, k=4)
LGBMClassifier	0.666 74	0.51 7202	0.47 5738	0.49 5604	SelectKBest(f_regression, k=4)
LGBMClassifier	0.666 74	0.51 7202	0.47 5738	0.49 5604	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
LGBMClassifier	0.666 74	0.51 7202	0.47 5738	0.49 5604	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
LGBMClassifier	0.666 74	0.51 7202	0.47 5738	0.49 5604	RFE(estimator=LogisticRegression(), n_features_to_select=4)
LGBMClassifier	0.666 74	0.51 7202	0.47 5738	0.49 5604	RFE(estimator=SVC(), n_features_to_select=4)
LGBMClassifier	0.666 74	0.51 7202	0.47 5738	0.49 5604	RFE(estimator=MLPClassifier(), n_features_to_select=4)

LGBMClassifier	0.66674	0.517202	0.475738	0.495604	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
LGBMClassifier	0.66674	0.517202	0.475738	0.495604	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
LGBMClassifier	0.66674	0.517202	0.475738	0.495604	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
LGBMClassifier	0.66674	0.517202	0.475738	0.495604	RFECV(estimator=SVC(), min_features_to_select=4)
LGBMClassifier	0.66674	0.517202	0.475738	0.495604	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
RandomForestClassifier	0.686027	0.517163	0.530802	0.523894	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
RandomForestClassifier	0.674761	0.516827	0.498945	0.507729	SelectKBest(f_regression, k=4)
SGDClassifier	0.595329	0.5164	0.272363	0.35663	MiniBatchSparsePCA(n_components=4)
RandomForestClassifier	0.675242	0.515968	0.501055	0.508402	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
BaggingClassifier	0.676608	0.515608	0.505274	0.510389	MiniBatchSparsePCA(n_components=4)
LGBMClassifier	0.675714	0.515578	0.502743	0.509079	SparsePCA(n_components=4)
PassiveAggressiveClassifier	0.599883	0.515405	0.285865	0.367757	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
RandomForestClassifier	0.677186	0.515327	0.507173	0.511217	SelectKBest(f_classif, k=4)
RandomForestClassifier	0.676916	0.514445	0.507173	0.510783	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
RandomForestClassifier	0.680588	0.513254	0.518776	0.516	MiniBatchSparsePCA(n_components=4)
BaggingClassifier	0.677674	0.512149	0.511392	0.51177	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
RandomForestClassifier	0.681039	0.512129	0.521097	0.516574	SelectKBest(mutual_info_classif, k=4)
RandomForestClassifier	0.679896	0.511992	0.517932	0.514945	RFE(estimator=MLPClassifier(), n_features_to_select=4)
BaggingClassifier	0.666817	0.510629	0.481435	0.495602	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
RandomForestClassifier	0.670333	0.508254	0.493671	0.500856	RFE(estimator=LogisticRegression(), n_features_to_select=4)
LGBMClassifier	0.674685	0.50728	0.507173	0.507227	PCA(n_components=4)
HistGradientBoostingClassifier	0.673884	0.506121	0.505907	0.506014	SparsePCA(n_components=4)
HistGradientBoostingClassifier	0.672352	0.505632	0.501899	0.503759	MiniBatchSparsePCA(n_components=4)
HistGradientBoostingClassifier	0.668121	0.503896	0.491139	0.497436	SelectKBest(f_classif, k=4)
HistGradientBoostingClassifier	0.668121	0.503896	0.491139	0.497436	SelectKBest(mutual_info_classif, k=4)
HistGradientBoostingClassifier	0.668121	0.503896	0.491139	0.497436	SelectKBest(f_regression, k=4)
HistGradientBoostingClassifier	0.668121	0.503896	0.491139	0.497436	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)

HistGradientBoo stingClassifier	0.668 121	0.50 3896	0.49 1139	0.49 7436	RFE(estimator=RandomForestClassifie r(), n_features_to_select=4)
HistGradientBoo stingClassifier	0.668 121	0.50 3896	0.49 1139	0.49 7436	RFE(estimator=LogisticRegression(), n_features_to_select=4)
HistGradientBoo stingClassifier	0.668 121	0.50 3896	0.49 1139	0.49 7436	RFE(estimator=SVC(), n_features_to_select=4)
HistGradientBoo stingClassifier	0.668 121	0.50 3896	0.49 1139	0.49 7436	RFE(estimator=MLPClassifier(), n_features_to_select=4)
HistGradientBoo stingClassifier	0.668 121	0.50 3896	0.49 1139	0.49 7436	RFECV(estimator=DecisionTreeClassi fier(), min_features_to_select=4)
HistGradientBoo stingClassifier	0.668 121	0.50 3896	0.49 1139	0.49 7436	RFECV(estimator=RandomForestClass ifier(), min_features_to_select=4)
HistGradientBoo stingClassifier	0.668 121	0.50 3896	0.49 1139	0.49 7436	RFECV(estimator=LogisticRegression( ) , min_features_to_select=4)
HistGradientBoo stingClassifier	0.668 121	0.50 3896	0.49 1139	0.49 7436	RFECV(estimator=SVC(), min_features_to_select=4)
HistGradientBoo stingClassifier	0.668 121	0.50 3896	0.49 1139	0.49 7436	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
BaggingClassifie r	0.675 725	0.50 3515	0.51 3713	0.50 8563	RFE(estimator=LogisticRegression(), n_features_to_select=4)
BaggingClassifie r	0.654 6	0.50 2576	0.45 2743	0.47 636	RFE(estimator=SVC(), n_features_to_select=4)
HistGradientBoo stingClassifier	0.675 872	0.50 1538	0.51 6034	0.50 8683	PCA(n_components=4)
GradientBoostin gClassifier	0.671 435	0.50 1155	0.50 3376	0.50 2263	MiniBatchSparsePCA(n_components= 4)
ExtraTreeClassif ier	0.662 907	0.50 0994	0.47 8481	0.48 9479	RFE(estimator=LogisticRegression(), n_features_to_select=4)
BaggingClassifie r	0.664 776	0.50 0764	0.48 4177	0.49 2331	SelectKBest(f_regression, k=4)
SGDClassifier	0.701 601	0.50 0178	0.59 3038	0.54 2664	SelectKBest(mutual_info_classif, k=4)
XGBClassifier	0.659 507	0.49 877	0.47 0464	0.48 4204	PCA(n_components=4)
BaggingClassifie r	0.666 5	0.49 8715	0.49 1139	0.49 4898	SelectKBest(mutual_info_classif, k=4)
BaggingClassifie r	0.656 272	0.49 7276	0.46 2236	0.47 9117	SelectKBest(f_classif, k=4)
SGDClassifier	0.711 569	0.49 7151	0.62 5949	0.55 4165	RFECV(estimator=DecisionTreeClassi fier(), min_features_to_select=4)
XGBClassifier	0.659 435	0.49 689	0.47 1941	0.48 4094	SparsePCA(n_components=4)
LGBMClassifier	0.668 868	0.49 5004	0.50 1688	0.49 8324	MiniBatchSparsePCA(n_components= 4)
BaggingClassifie r	0.649 383	0.49 3452	0.44 5148	0.46 8057	PCA(n_components=4)
RandomForestCl assifier	0.665 91	0.49 2246	0.49 557	0.49 3902	RFE(estimator=SVC(), n_features_to_select=4)
XGBClassifier	0.664 648	0.49 0764	0.49 3249	0.49 2003	MiniBatchSparsePCA(n_components= 4)
SGDClassifier	0.717 709	0.49 0028	0.65 3165	0.55 9957	RFECV(estimator=RandomForestClass ifier(), min_features_to_select=4)
BaggingClassifie r	0.668 409	0.48 9388	0.50 5907	0.49 751	RFECV(estimator=DecisionTreeClassi fier(), min_features_to_select=4)

BaggingClassifier	0.653 159	0.48 7316	0.46 2025	0.47 4334	RFE(estimator=MLPClassifier(), n_features_to_select=4)
PassiveAggressiveClassifier	0.669 294	0.48 6649	0.51 1392	0.49 8714	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
XGBClassifier	0.660 76	0.48 2637	0.48 9662	0.48 6124	SelectKBest(f_classif, k=4)
XGBClassifier	0.660 76	0.48 2637	0.48 9662	0.48 6124	SelectKBest(mutual_info_classif, k=4)
XGBClassifier	0.660 76	0.48 2637	0.48 9662	0.48 6124	SelectKBest(f_regression, k=4)
XGBClassifier	0.660 76	0.48 2637	0.48 9662	0.48 6124	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
XGBClassifier	0.660 76	0.48 2637	0.48 9662	0.48 6124	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
XGBClassifier	0.660 76	0.48 2637	0.48 9662	0.48 6124	RFE(estimator=LogisticRegression(), n_features_to_select=4)
XGBClassifier	0.660 76	0.48 2637	0.48 9662	0.48 6124	RFE(estimator=SVC(), n_features_to_select=4)
XGBClassifier	0.660 76	0.48 2637	0.48 9662	0.48 6124	RFE(estimator=MLPClassifier(), n_features_to_select=4)
XGBClassifier	0.660 76	0.48 2637	0.48 9662	0.48 6124	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
XGBClassifier	0.660 76	0.48 2637	0.48 9662	0.48 6124	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
XGBClassifier	0.660 76	0.48 2637	0.48 9662	0.48 6124	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
XGBClassifier	0.660 76	0.48 2637	0.48 9662	0.48 6124	RFECV(estimator=SVC(), min_features_to_select=4)
XGBClassifier	0.660 76	0.48 2637	0.48 9662	0.48 6124	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
PassiveAggressiveClassifier	0.690 191	0.48 2274	0.57 9747	0.52 6538	RFE(estimator=LogisticRegression(), n_features_to_select=4)
BaggingClassifier	0.650 93	0.48 1997	0.46 0338	0.47 0918	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
BaggingClassifier	0.655 728	0.48 1023	0.47 5949	0.47 8473	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
BaggingClassifier	0.649 005	0.47 9416	0.45 6962	0.46 792	SparsePCA(n_components=4)
AdaBoostClassifier	0.645 028	0.47 8793	0.44 5359	0.46 1471	MiniBatchSparsePCA(n_components=4)
BaggingClassifier	0.660 555	0.47 7208	0.49 4726	0.48 5809	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
SGDClassifier	0.578 616	0.47 1423	0.24 5359	0.32 2742	RFECV(estimator=SVC(), min_features_to_select=4)
SGDClassifier	0.689 922	0.46 5653	0.60 0633	0.52 4599	SparsePCA(n_components=4)
PassiveAggressiveClassifier	0.736 446	0.46 4739	0.74 9367	0.57 369	RFECV(estimator=SVC(), min_features_to_select=4)
SGDClassifier	0.699 235	0.46 2627	0.63 4599	0.53 5136	PCA(n_components=4)
DecisionTreeClassifier	0.655 117	0.45 9086	0.49 8312	0.47 7896	MiniBatchSparsePCA(n_components=4)
ExtraTreeClassifier	0.645 447	0.45 7938	0.46 8565	0.46 3191	RFECV(estimator=SVC(), min_features_to_select=4)

DecisionTreeClassifier	0.65493	0.457906	0.499156	0.477642	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
DecisionTreeClassifier	0.650111	0.457445	0.484177	0.470431	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
ExtraTreeClassifier	0.635472	0.454209	0.440506	0.447253	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
DecisionTreeClassifier	0.651466	0.453786	0.493038	0.472599	SelectKBest(f_regression, k=4)
DecisionTreeClassifier	0.652971	0.452823	0.499156	0.474862	RFE(estimator=LogisticRegression(), n_features_to_select=4)
DecisionTreeClassifier	0.65225	0.452798	0.496835	0.473795	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
ExtraTreeClassifier	0.659253	0.452692	0.519831	0.483944	PCA(n_components=4)
DecisionTreeClassifier	0.658161	0.45175	0.517511	0.482399	RFE(estimator=MLPClassifier(), n_features_to_select=4)
DecisionTreeClassifier	0.651089	0.451638	0.494515	0.472105	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
DecisionTreeClassifier	0.642554	0.450865	0.467511	0.459037	PCA(n_components=4)
DecisionTreeClassifier	0.650484	0.448618	0.496414	0.471307	RFECV(estimator=SVC(), min_features_to_select=4)
PassiveAggressiveClassifier	0.665584	0.448485	0.546414	0.49263	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
DecisionTreeClassifier	0.654435	0.44811	0.510127	0.477111	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
DecisionTreeClassifier	0.655417	0.447857	0.513713	0.47853	RFE(estimator=SVC(), n_features_to_select=4)
ExtraTreeClassifier	0.630028	0.446238	0.431646	0.43882	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
ExtraTreeClassifier	0.636294	0.445643	0.453165	0.449372	SelectKBest(mutual_info_classif, k=4)
DecisionTreeClassifier	0.654149	0.444181	0.514557	0.476786	SelectKBest(f_classif, k=4)
ExtraTreeClassifier	0.624946	0.44417	0.417089	0.430203	SparsePCA(n_components=4)
DecisionTreeClassifier	0.649385	0.443965	0.498945	0.469852	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
DecisionTreeClassifier	0.647917	0.443457	0.494726	0.46769	SelectKBest(mutual_info_classif, k=4)
SGDClassifier	0.658097	0.440341	0.533333	0.482397	RFE(estimator=LogisticRegression(), n_features_to_select=4)
ExtraTreeClassifier	0.650133	0.435764	0.51308	0.471272	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
ExtraTreeClassifier	0.650857	0.43348	0.518987	0.472396	MiniBatchSparsePCA(n_components=4)
DecisionTreeClassifier	0.629059	0.433244	0.444304	0.438704	SparsePCA(n_components=4)
ExtraTreeClassifier	0.639881	0.426428	0.491561	0.456684	SelectKBest(f_classif, k=4)
ExtraTreeClassifier	0.62958	0.425152	0.457173	0.440581	RFE(estimator=MLPClassifier(), n_features_to_select=4)
ExtraTreeClassifier	0.63923	0.42411	0.492827	0.455894	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)

ExtraTreeClassifier	0.623 144	0.42 1853	0.43 903	0.43 027	SelectKBest(f_regression, k=4)
ExtraTreeClassifier	0.629 722	0.41 8254	0.46 7932	0.44 1701	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
ExtraTreeClassifier	0.616 138	0.41 791	0.41 9409	0.41 8659	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
ExtraTreeClassifier	0.611 507	0.41 4608	0.40 7173	0.41 0857	RFE(estimator=SVC(), n_features_to_select=4)
PassiveAggressiveClassifier	0.682 597	0.41 4557	0.66 6878	0.51 1282	MiniBatchSparsePCA(n_components=4)
PassiveAggressiveClassifier	0.633 97	0.40 2561	0.51 0759	0.45 0251	PCA(n_components=4)
SGDClassifier	0.715 404	0.38 6526	0.87 6371	0.53 645	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
PassiveAggressiveClassifier	0.505 663	0.35 5556	0.02 7004	0.05 0196	SelectKBest(f_regression, k=4)
PassiveAggressiveClassifier	0.616 375	0.34 6182	0.58 9241	0.43 6134	SparsePCA(n_components=4)
BernoulliNB	0.536 973	0.33 288	0.20 654	0.25 4915	PCA(n_components=4)
BernoulliNB	0.536 973	0.33 288	0.20 654	0.25 4915	SparsePCA(n_components=4)
BernoulliNB	0.530 646	0.30 1609	0.23 7342	0.26 5643	MiniBatchSparsePCA(n_components=4)
SGDClassifier	0.549 54	0.29 1753	0.44 557	0.35 2617	SelectKBest(f_regression, k=4)
SGD Classifier	0.503 94	0.25 9615	0.09 1139	0.13 4916	RFE(estimator=SVC(), n_features_to_select=4)