ANALYZING THE IMPACT OF FEATURE SELECTION ON MACHINE LEARNING ALGORITHMS: A MULTI-DATASET INVESTIGATION

Nahush Agrawal

A dissertation submitted to
The School of Computing Sciences of the University of East Anglia
in partial fulfilment of the requirements for the degree of
MASTER OF SCIENCE.
10 OCTOBER

SUPERVISOR(S), MARKERS/CHECKER AND ORGANISER

The undersigned hereby certify that the markers have independently marked the dissertation entitled "**ANALYZING THE IMPACT OF FEATURE SELECTION ON MACHINE LEARNING ALGORITHMS: A MULTI-DATASET INVESTIGATION**" by **Nahush Agrawal**, and the external examiner has checked the marking, in accordance with the marking criteria and the requirements for the degree of **Master of Science**.

Supervisor: _____
Dr. Gavin Cawley

Markers: _____
Marker 1: Dr. Gavin Cawley

_____
Marker 2: Dr. Taoyang Wu

External Examiner: _____
Checker/Moderator

Moderator: _____
Dr. Wenjia Wang

DISSERTATION INFORMATION AND STATEMENT

Dissertation Submission Date: **10 October**

Student:	**Nahush Agrawal**
Title:	**ANALYZING THE IMPACT OF FEATURE SELECTION ON MACHINE LEARNING ALGORITHMS: A MULTI-DATASET INVESTIGATION**
School:	**Computing Sciences**
Course:	**Computing Science**
Degree:	**MSc.**
Duration:	**2022–2023**
Organiser:	**Dr. Wenjia Wang**

STATEMENT:

Unless explicitly stated or cited, the research presented in this dissertation is, to the best of my understanding and conviction, solely my own contribution. The work in question has not been previously presented, either in its entirety or in any partial form, for the purpose of obtaining a degree at this or any other educational or professional institution.

Subject to confidentiality restriction if stated, permission is herewith granted to the University of East Anglia to circulate and to have copied for non-commercial purposes, at its discretion, the above title upon the request of individuals or institutions.

Signature of Student

Abstract

This study investigates the impact of feature selection on machine learning algorithms using a collection of 20 diverse datasets. In order to improve the performance of algorithms, the method of feature selection becomes a crucial approach. The objectives of this study are the evaluation of various selection techniques, the analysis of their impacts on classifiers using different metrics, the establishment of robustness through K-fold cross-validation, the comparison with alternative classifiers such as logistic regression, SVM , Random forest etc and the investigation of algorithm-specific hyperparameters. The methodology encompasses many key steps, including preprocessing of the dataset, application of feature selection techniques, training of classifiers both with and without feature selection, and a thorough assessment of performance. The process of conducting systematic experiments reveals the intricate relationship among selection techniques, algorithm selection, and hyperparameters. The expected contributions of this study encompass the provision of empirical insights pertaining to the optimisation of algorithms through the process of feature selection methods including Hybrid Filter-Wrapper and Wrapper RFE,SFS and SBS. Additionally, this research aims to enhance the understanding of metrics that are sensitive to the selection process. Furthermore, a comparative analysis will be conducted based on both the cases. Lastly, the study aims to examine the impact of feature selection on ml algorithms.

In conclusion, this study contributes to the advancement of knowledge about the significance of feature selection in the optimisation of machine learning processes. It provides valuable insights that may guide the selection, setup, and improvement of algorithms.

Acknowledgements

I wish to express my sincere thanks to my exceptional supervisor, Dr. Gavin Cawley, for his helpful advice, steadfast support, and smart ideas during this research journey.His profound knowledge and continuous encouragement have laid the foundation for this investigation. In addition to the foregoing, I also want to express my sincere thanks to all of my lecturers who have guided me, my friends who accompanied with me during this quest, and my family who have unwaveringly provided me with financial as well as emotional assistance. My drive has continuously been fueled by their unshakable faith in my abilities throughout all of my academic endeavours.

In closing, I would want to say that the support, assistance, and love that I have gotten from the aforementioned folks have been extremely helpful in turning my study into a practical endeavour. Their contributions, no matter how big or how small, have left an indelible stamp on this trip, and for that, I am appreciative in the deepest and most honest way possible.

Nahush Agrawal

Norwich, UK.

Table of Contents

Abstract iv

Acknowledgements v

Table of Contents vi

List of Tables ix

List of Figures x

List of Abbreviations xiv

1 Introduction 1
- 1.1 Background . . . 2
- 1.2 Essence of Feature Selection in Machine Learning Models . . . 3
- 1.3 The Aims of the Study . . . 4
  - 1.3.1 Evaluation of Feature Selection Techniques: . . . 4
  - 1.3.2 Analysis of Classifier Performance under Feature Selection: . . . 4
  - 1.3.3 Robustness through K-fold Cross-Validation: . . . 4
  - 1.3.4 Comparative Analysis of Different Classifiers: . . . 5
  - 1.3.5 Hyperparameter Investigation Specific to Algorithms: . . . 7
- 1.4 Expected Outcomes . . . 8
- 1.5 Summary . . . 8

2 Literature Review 9
- 2.1 Introduction . . . 9
- 2.2 Feature Selection Methods . . . 10
  - 2.2.1 Feature ranking . . . 10
  - 2.2.2 Feature Subset Selection . . . 10
- 2.3 Models Based On Feature Selection . . . 12
  - 2.3.1 Filter . . . 12
  - 2.3.2 Wrapper Methods . . . 14
  - 2.3.3 Embedded Methods . . . 17
  - 2.3.4 Hybrid Based Filter-Wrapper . . . 18
- 2.4 Limitations of Feature Selection . . . 20
- 2.5 Strategies for Addressing Feature Selection Limitations . . . 21
- 2.6 Summary . . . 21

3 Design of the Methodology **22**
3.1 Research Design . . . 22
3.1.1 Research Objective . . . 22
3.1.2 Research Hypotheses . . . 23
3.2 Data Collection . . . 24
3.2.1 Data Preprocessing . . . 29
3.2.2 Feature Engineering . . . 30
3.3 Feature Selection Techniques . . . 30
3.3.1 Overview . . . 31
3.3.2 Technique Selection . . . 32
3.3.3 Implementation . . . 33
3.3.4 Model Training . . . 34
3.3.5 Hyperparameter Tuning . . . 36
3.3.6 Evaluation Metrics . . . 37
3.3.7 Experimental Setup and Design . . . 39
3.4 Data Analysis . . . 39
3.5 Summary . . . 40

4 Research and Experiments **41**
4.1 Introduction . . . 41
4.1.1 Adjustment to Data Presentation in the Appendix . . . 42
4.2 Hybrid Method[Filter-Wrapper] . . . 42
4.2.1 Case 1 - Video Game Dataset . . . 43
4.2.2 Case 2: Sonar Dataset . . . 47
4.2.3 Case 3: Student Dataset . . . 50
4.2.4 Case 4:Loan Prediction Dataset . . . 53
4.2.5 Case 5: Social Media Ads . . . 56
4.2.6 Case 6:Water Quality . . . 60
4.2.7 Case 7:Bank Authentication Dataset . . . 61
4.2.8 Aggregate Results . . . 63
4.3 Wrapper Methods . . . 64
4.3.1 Case1:Glass Dataset . . . 66
4.3.2 Case 2:Heart Dataset . . . 68
4.3.3 Case 3:Titanic Dataset . . . 71
4.3.4 Case 4: Breast Dataset . . . 72
4.3.5 Case 5: Adult Dataset . . . 76
4.3.6 Case 6:Diabetes Dataset . . . 77
4.3.7 Case 7:Wine Dataset . . . 79
4.3.8 Case 8:Weather Dataset . . . 80
4.3.9 Aggregate Results . . . 81

5 Conclusion **83**
5.1 Evaluation and Discussion . . . 83
5.2 Conclusion . . . 84
5.3 Suggestion for Further Work . . . 85

References **87**

A A Sample Appendix on Invariant Subspaces **92**
A.1 Evaluating Performance metrics 92
A.2 Data Cleaning and Preprocessing 94
A.2.1 Data Summarization 94
A.2.2 Handling Missing Values: 94
A.2.3 Handling Duplicate value 95
A.2.4 Handling Outliers 95
A.2.5 Data Transformation 96
A.2.6 Data Stratification 96
A.3 Feature Selection 96
A.3.1 Hybrid(Filter-Wrapper) 96
A.3.2 Wrapper Method 97
A.3.3 Model Training 98
A.4 Hyperparameter Tuning For other datasets 99
A.4.1 Hybrid Filter Wrapper Datasets 100
A.4.2 Case 1: Indian Diabetes 100
A.4.3 Case 2:Stroke Prediction Dataset 103
A.4.4 Case 3:Irish Dataset 104
A.4.5 Wrapper based Dataset 105
A.4.6 Case 2:Ionosphere Dataset 108
A.4.7 Limitation of Hybrid (Filter-Wrapper) 110
A.4.8 Limitation of Wrapper Method 110
A.5 110

List of Tables

List of Figures

2.1 Schematic diagram based on Wrapper 14
2.2 Schematic Diagram of Proposed Hybrid[Filter-wrapper] method 18

4.1 Hybrid Method Dataset Table 42
4.2 Videogame Class Distribution 44
4.3 Sigmoid Distribution of Selected Features 44
4.4 Performance Metrics Comparison of All Features 45
4.5 Performance Metrics Comparison of Selected Features 45
4.6 Bar Plot Comparison of Selected Feature 46
4.7 Class Distribution Diagram 47
4.8 Sigmoid Diagram of Selected Features 48
4.9 Performance metrics based on All features 48
4.10 Performance metrics based on All Features 49
4.11 Performance metrics based on Selected Features 49
4.12 Performance metrics based on Selected Features 50
4.13 Class Distribution Diagram 50
4.14 Sigmoid Diagram of Selected Features 51
4.15 Performance metrics based on All features 51
4.16 Performance metrics based on All Features 51
4.17 Performance metrics based on All Features 51
4.18 Performance metrics based on All Features 52
4.19 Performance metrics based on Selected Features 52
4.20 Performance metrics based on Selected Features 52
4.21 Class Distribution Diagram 53
4.22 Sigmoid Diagram of Selected Features 54
4.23 Performance metrics based on All Features 54
4.24 Performance metrics based on All Features 55
4.25 Performance metrics based on Selected Features 55
4.26 Performance metrics based on Selected Features 56
4.27 Class Distribution Diagram 57

4.28 Sigmoid Diagram of Selected Features 57
4.29 Sigmoid Diagram of Selected Features 57
4.30 Sigmoid Diagram of Selected Features 58
4.31 Sigmoid Diagram of Selected Features 58
4.32 Sigmoid Diagram of Selected Features 58
4.33 Performance metrics based on All features 58
4.34 Performance metrics based on All features 59
4.35 Performance metrics based on Selected Features 59
4.36 Performance metrics based on Selected Features 59
4.37 Class Distribution Diagram 60
4.38 Performance metrics based on All Features 60
4.39 Performance metrics based on Selected Features 61
4.40 Class Distribution Diagram 61
4.41 Performance metrics based on All Features 62
4.42 Performance metrics based on All Features 62
4.43 Performance metrics based on selected Features 62
4.44 Performance metrics based on selected Features 63
4.45 Aggregated Outcomes According to Performance Indicators 64
4.46 Wrapper Classifiers List 65
4.47 Performance metrics After Feature Selection [RFE] 66
4.48 Performance metrics After Feature Selection [RFE] 66
4.49 Performance metrics After Feature Selection [SFS] 66
4.50 Performance metrics After Feature Selection [SFS] 67
4.51 Performance metrics After Feature Selection [SBS] 67
4.52 Performance metrics After Feature Selection [SBS] 67
4.53 Performance metrics based on All Features 67
4.54 Performance metrics based on All Features 68
4.55 Performance metrics based on RFE 68
4.56 Performance metrics based on RFE 69
4.57 Performance metrics based on Forward Selection 69
4.58 Performance metrics based on Forward Selection 69
4.59 Performance metrics based on Backward Selection 69
4.60 Performance metrics based on Backward Features 70
4.61 Performance metrics based on All Features 70
4.62 Performance metrics based on All Features 70
4.63 Performance metrics based on RFE 71
4.64 Performance metrics based on SFS 71

4.65 Performance metrics based on SBS 71
4.66 Performance metrics based on All Features 72
4.67 Performance metrics based on Recursive Feature selection 73
4.68 Performance metrics based on Forward Selection 73
4.69 Performance metrics based on Backward Selection 74
4.70 Performance metrics based on All Feature 74
4.71 Performance metrics based on RFE 76
4.72 Performance metrics based on SFS 76
4.73 Performance metrics based on SBS 76
4.74 Performance metrics based on All Features 77
4.75 Performance metrics based on [RFE,SFE,SBS,All] 77
4.76 Performance metrics based on All Features 78
4.77 Performance metrics based on [RFE,SFE,SBS] 79
4.78 Performance metrics based on [RFE,SFE,SBS] 79
4.79 Performance metrics based on RFE 80
4.80 Performance metrics based on Forward Selection 80
4.81 Performance metrics based on Backward Selection 80
4.82 Performance metrics based on All Features 81
4.83 82
4.84 Aggregated Outcomes According to Performance Indicators 82

A.1 Describing Dataset 94
A.2 Describing Dataset 94
A.3 Describing Dataset 95
A.4 Describing Dataset 95
A.5 Scaling 96
A.6 Stratification(train-test split) 96
A.7 Filter Method(Pearson Corelation,IG,and Fisher) 97
A.8 Wrapper With RFE 97
A.9 Wrapper Methods SFS and SBS 98
A.10 Machine Learning Models 98
A.11 Target Variable Class Distribution 100
A.12 Sigmoid diagram based on feature selection 101
A.13 SIgmoid diagram based on feature selection 101
A.14 Bar Plot All Features 102
A.15 Performance metrics based on all Features 102
A.16 Performance metrics based on Selected Features 103
A.17 Class Distribution Diagram 103

A.18 Performance metrics based on All Features 104
A.19 Performance metrics based on Selected Features 104
A.20 Performance metrics based on Selected Features 105
A.21 Performance metrics based on All 105
A.22 Performance metrics based on (RFE,SFS,SBS) 106
A.23 Performance metrics based on RFE 108
A.24 Performance metrics based on Forward Selection 108
A.25 Performance metrics based on Backward Selection 108
A.26 Performance metrics based on All Features 109

Chapter 1

Introduction

This Chapter describes the research scope and aims of this study are presented, focusing on the examination of the complex interplay between feature selection and machine learning algorithms. In the current era characterised by an enormous amount of data and growing complexity of algorithms, the careful selection and organisation of pertinent aspects have emerged as crucial factors in optimising algorithms. This section help to provides an explanation of the underlying purpose for doing the study, which arises from the significant impact that feature selection has on improving the efficiency and performance of algorithms. The research aims to address several key objectives and one of the goals is to look at various feature selection methods across different algorithms and data sets, assessing the influence of these techniques on classifier performance using various metrics, ensuring the reliability of results through K-fold cross validation, conducting a thorough comparative analysis of classifiers with and without feature selection, and investigating the impact of algorithm-specific hyper parameters. The chapter presents a comprehensive examination of the structured technique, which includes many stages such as data preparation, feature selection, classifier training, performance assessment, robustness validation, and hyper parameter exploration. Finally, the chapter emphasises the anticipated contributions of the study, which encompass empirical observations on the effectiveness of feature selection, a comprehensive examination of performance metrics using sensitivity analysis, educated decision-making in classifier selection, and Understanding of hyper parameters with detail. In summary,Chapter 1 focused on the foundation for a thorough investigation of the crucial significance of the role of feature selection in optimising machine learning algorithms

1.1 Background

The background of this study is based on how quickly machine learning is changing and how it can be used in many different fields and businesses. As per Iniesta et al. (2016)as the amount of data keeps growing at a rate that has never been seen before, it has become clear that it is getting harder and harder to get useful information from complicated and high-dimensional datasets. Machine learning algorithms are a useful way to turn these huge datasets into knowledge that can be used, but their success depends on the quality and importance of the features they are fed . The "garbage in, garbage out" concept by Canbek (2022) demonstrates the significance of feature selection for machine learning.The performance of a model can be negatively impacted by components that are incorrect, redundant, or unnecessary, which can lead to less-than-ideal outcomes and perhaps erroneous conclusions. On the other hand, models may be made simpler to comprehend, their processing complexity can be reduced, and it can become much simpler to apply previously acquired patterns to new data if the appropriate features are used. As per Zhang et al. (2018) the emergence of deep learning and other sophisticated approaches has brought feature selection back into the limelight, despite the fact that it has long been recognised that feature selection is a crucial aspect of machine learning. The number of dimensions that make up the feature space increases as algorithms get more complex and are able to process more datasets. Because of this, it becomes more difficult to make effective use of computing resources and to avoid overfitting. Because of this, it is essential to have a solid understanding of the characteristics that contribute the most to the precision and adaptability of forecasts.

Cai et al. (2018) aims to fill the gap between the exponential growth of data and the need for better machine-learning methods. The study aims to find out how well different methods work by carefully looking at how different feature selection techniques affect different machine learning algorithms and datasets. The study also tries to find out how feature selection affects algorithm behaviour in subtle ways. This will lead to better model performance and better decisions. This study is primarily a response to the pressing need to get a deeper understanding of the function of feature selection in

the context of contemporary machine learning. It aims to contribute to the continuing conversation in the area by casting light on best practises, demonstrating the linkages between feature selection and algorithm optimisation, and providing practitioners with the information they require to deal with the complexities of dealing with feature-rich datasets in a data-driven world.

1.2 Essence of Feature Selection in Machine Learning Models

The exponential development of technology in the modern period has led to the massive data creation from several directions, including films, photographs, texts, and other types of media. The availability of high-dimensional data, which includes a great number of characteristics or variables, poses difficulties for data analysis and decision-making due to the complexity of the data and the possibility of overfitting. In response to these obstacles, a new strategy known as feature selection has been developed (Cai et al. 2018). Choosing a subset of the most important characteristics from a dataset is the process known as feature selection. The goals of feature selection are to increase learning accuracy, minimise processing time, and simplify output. This approach is distinct from feature extraction, which generates new features by using the original data as a starting point. Applications of feature selection may be found in a broad variety of domains, including image recognition and bioinformatics.As per Li et al. (2014)There is a wide variety of methodology that may be utilised for the process of feature selection. Some of these methodologies include statistics, information theory, and manifold learning. They can also be categorised in a variety of other ways, such as by the type of training data that they use (supervised, unsupervised, or semi-supervised), by their relationship to learning methods (filter, wrapper, or embedded), by their evaluation criteria (such as correlation or Euclidean distance), by their search strategies, and by the nature of their output (ranking features or selecting subsets). In short feature selection is a crucial instrument that must be utilised in order to properly harness the potential of data in light of its continuing growth in volume and complexity.

1.3 The Aims of the Study

The research goals of this study have been meticulously formulated to deeply examine the synergy between feature selection and machine learning methodologies. The study aims to provide tangible real-world evidence and insights that can be leveraged to enhance algorithmic performance in the context of large and intricate datasets. The specific research objectives are as follows:

1.3.1 Evaluation of Feature Selection Techniques:

The purpose of this research is to evaluate in depth a wide range of different feature selection approaches, applying them to a number of different data sets. Methods like as Recursive Feature Elimination, Sequential forward selection,Filter methods such as the Mutual Information, and Pearson co relation etc are going to be used in a methodical manner so that their efficacy may be evaluated in determining which characteristics are the most important contributors to an improvement in algorithmic performance.

1.3.2 Analysis of Classifier Performance under Feature Selection:

In this work, a significant amount of focus is placed on determining how the incorporation of feature selection methods impact the performance of Machine learning models. The purpose of this study is to quantify the multifarious effect that feature selection has on many aspects of classification by using a broad spectrum of performance measurements such as accuracy, precision, recall, F1-score, and AUC-ROC (Forman et al. 2003).

1.3.3 Robustness through K-fold Cross-Validation:

The approach of k-fold cross-validation will be used into the research project to assure the reliability and validity of the findings. This technique will ensure the stability of findings across multiple data splits by iteratively splitting the dataset and testing on unique subsets. This will enhance the applicability and dependability of the results.The experimental results on 20 data sets indicate that the accuracy estimates derived from multiple replications of k-fold cross validation are highly correlated, with the correlation

increasing as the number of folds increases. For evaluating the efficacy of classification algorithms, k-fold cross validation with a large number of folds and a small number of replications should be used (Wong & Yeh 2019).

1.3.4 Comparative Analysis of Different Classifiers:

In this work, a comparison and contrast is made between well-known machine learning classifiers such as Support Vector Machines and logistic regression, as well as other options for classifiers. The study sheds light on the benefits and restrictions associated with various algorithmic options by analyzing the performance of classifiers with feature selection and including all features (Jalali et al. 2017).

Machine Learning Models

Our study involves the evaluation of four prominent machine learning models, each selected for its relevance to classification tasks and varying levels of complexity. Below, we provide detailed explanations of these models and their suitability for our research:

Logistic Regression

Regression modelling is a widely employed and valuable statistical technique that is utilised to examine and elucidate the association between a dependent or response variable and a collection of independent predictors (Das 2021). Logistic Regression is a fundamental classification algorithm widely used for binary and multi-class classification tasks. It models the probability of a binary outcome based on one or more predictor variables. In our study, we employ Logistic Regression due to its simplicity, interpretability, and effectiveness in linearly separable classification problems. It serves as a baseline model to evaluate the impact of feature selection techniques.

Support Vector Machine (SVM)

As per Awad et al. (2015) in classification problems, the objective of a discriminant machine learning approach is to identify a function with discriminant properties that can accurately predict labels for newly obtained instances. This is achieved by utilising an independent and identically distributed (iid) training dataset.Support Vector

Machine (SVM) is a versatile and powerful classification algorithm capable of handling both linear and non-linear classification tasks. SVM aims to find a hyperplane that maximizes the margin between classes, making it robust against overfitting. Including SVM in this study to assess feature selection's impact on a model known for its ability to handle complex classification scenarios.

Naive Bayes

In the context of binary classification problems, logistic Regression and support vector machines are commonly used. However, it is important to note that these methods are specifically designed to best work on scenarios when there are only two target classes. This constraint poses a hindrance to the ability of these models to generalise effectively to real-world activities that include numerous classifications.The Naive Bayes Classifier has demonstrated its versatility in multiple classification tasks, such as weather forecasting and health classification. After the tabulation of data, the process of computing classification validity takes place, whereby the most appropriate category is selected (Yang 2018). Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. De- spite its simplicity and the assumption of feature independence (hence the "naive" part), Naive Bayes has been surprisingly effective in various classification tasks. We include Naive Bayes to explore how feature selection affects probabilistic classification models.

Decision Trees

Decision trees are widely used in the field of data mining as tools for categorization or prediction, relying on the consideration of many criteria. Populations are partitioned into branch-like structures consisting of root, internal, and leaf nodes.Its Non-parametric nature are capable of effectively handling datasets that lack a predetermined framework due to their inherent flexibility (Song & Ying 2015). A supervised machine learning approach known as a decision tree is utilised for both classification and regression problems. The value of the input features is used to divide the data into subsets. Each split is a decision node that branches off into leaf nodes. The final result or choice is represented by the leaf nodes. A succession of decisions are represented graphically by a tree, which facilitates interpretation and understanding.

Random Forest

Random forests consist of several classification and regression trees that employ binary splits on variables in order to generate predictions. In contrast to an individual decision tree, a random forest algorithm generates several decision trees by employing random subsets of data and predictors.. Predictions from each tree are then combined for a final outcome. The use of ensemble methods frequently yields better precision compared to individual decision trees, while yet maintaining the interpretability that comes with models based on decision trees. (Brasil et al. 2019). One notable feature of random forests is their ability to effectively handle enormous predictor datasets. In order to optimise efficiency, it is advisable to restrict the quantity of predictors, particularly in domains such as medical records, where just essential variables may be necessary for precise predictions (Speiser et al. 2019).

Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. It is known for its robustness, ability to handle high-dimensional data, and resistance to overfitting. Random Forest is included in our study to assess feature selection's impact on ensemble models, which often benefit from reduced feature sets.

These machine learning models were chosen based on their relevance to classification tasks and their varying complexities. This selection allows us to investigate the impact of feature selection techniques across a spectrum of algorithms, from simple and interpretable models like Logistic Regression to more complex and powerful mod- els like Random Forest. By assessing these models' performance with and without feature selection, we aim to provide insights into the effectiveness of feature selection techniques in enhancing classification model outcomes.

1.3.5 Hyperparameter Investigation Specific to Algorithms:

This work investigates, in a methodical manner, the interaction that exists between hyperparameters and feature selection, bearing in mind the significant influence that algorithm-specific hyperparameters have on the effectiveness of classifiers.(Tran et al. 2020)The purpose of the study is to determine the ideal configurations of hyperparameters for a variety of feature selection techniques and models. This will be accomplished

by carefully modifying hyperparameter configurations.

1.4 Expected Outcomes

The study findings and observations underscore the significance of feature selection techniques in influencing the behaviour and performance of machine learning models.

Empirical Insights: The purpose of the study is to work on the effectiveness of hybrid Filter -Wrapper and Wrapper [RFE,SFS and SBS]feature selection approaches over a broad range of datasets and machine learning algorithms .

Metric Sensitivity: This study will helps to understand how feature selection affects a number of different aspects of classifier performance by analysing a comprehensive range of performance metrics.

Comparative Analysis: Through a comparative analysis of different classifiers, the study will helps understanding the benefits and limitations of various algorithmic options under diverse feature selection circumstances.

1.5 Summary

In summary, the purpose of this investigation is to broaden our understanding of the ways in which feature selection might affect the performance of machine learning approaches. The study aims to provide practitioners with relevant insights that will allow them to make educated decisions about feature selection and algorithm optimisation by achieving the outlined research objectives and applying a structured approach.

Chapter 2

Literature Review

2.1 Introduction

Machine learning techniques are frequently used in modern scientific research to solve problems with predictive modeling (Li et al. (2017)). Building a predictive model is difficult since these problems are frequently characterized by enormous amounts of data and an overwhelming number of features (Bolón-Canedo et al. (2015)). Frequently, a sizable fraction of these characteristics are either unnecessary or unrelated to the model.An excessive amount of features can lead to poorer accuracy, more computing needs, more memory consumption, and slower processing rates for common applications like supervised and unsupervised classification or regression. As a result, it's critical to choose an ideal—and preferably small—set of characteristics that produce the greatest results for classification or regression tasks. With regard to overcoming these difficulties, feature selection is a crucial process that provides numerous advantages Saeys et al. (2011). When dealing with datasets that encompass thousands or even hundreds of thousands of characteristics, however, the process of selecting the most important features becomes more difficult (Bolón-Canedo et al. (2015)).In academic discussions, these features are often referred to as variables or dimensions Saeys et al. (2011). If a feature gives redundant information or is obviously unrelated to the desired result, it may be regarded superfluous.In the realm of machine learning, feature selection has significant importance for both classification and regression objectives, particularly within specialist disciplines such as bioinformatics. Saeys et al. (2011). The main objective is to identify a certain group of important features that enhance the functionality of classifiers or regressors Li et al. (2017).

2.2 Feature Selection Methods

2.2.1 Feature ranking

The process of selecting important characteristics features can be broadly divided into two main methodologies: the method of ranking specific features based on their importance, frequently referred to as "feature ranking," and the more thorough approach of selecting a subset of features that collectively contribute to the performance of the model, known as "features subset selection." [reference].

Feature ranking stands as a cornerston in machine larning and data science, designd to prioritize faturs based on their relevance to a specific objective. Unlike feature selection, which isolates a subset of features,

Yet, feature ranking is not without its challenges. A significant constraint is its presumption that features operate independently. This can manifest in two primary complications:

• Certain features, while individually deemed non-essential, may gain relevance when considered in conjunction with others.

• Features that are recognized as individually impactful might inadvertently introduce redundancies.

There's a plethora of techniques available for feature ranking, ranging from correlation coefficients and mutual information to machine learning models like decision trees and SVM classifiers , as discussed by Zhang et al. Chang & Lin (2011). Furthermore, in the realm of gene expression data, Tan et al. emphasize the efficacy of feature ranking in pinpointing crucial genes that play a role in microbe-host interactions Tan et al. (2016).

2.2.2 Feature Subset Selection

Feature subset selection involves picking the most effective features to enhance prediction or classification accuracy. It's grounded in the principle of parsimony(Bell & Wang (2007)), also known as Occam's razor, which advocates for the simplest model that sufficiently represents data. Einstein emphasized this, suggesting models should be as simple as necessary, but no simpler ((Parzen 1982)). However, implementing this

principle in feature selection is challenging. The challenge of selecting the optimal subset of features is classified as NP-complete. ((Gheyas & Smith 2010)). Some features, though individually irrelevant, become significant when combined with others, leading to complex interactions. Additionally, while some features are essential, others add noise and complexity, making them redundant. High feature correlation doesn't negate feature complementarity. Lastly, multicollinearity can mistakenly exclude significant predictors from models.Performing an extensive search of all potential subsets of characteristics ensures the identification of the optimal subset of features. Regrettably, the computational feasibility of this task is limited, even when considering a database of moderate size. (for n features, the number of possible feature subsets is , too large to be evaluated even for modest n).

There exist two basic categories of optimum feature subset selection: filter and wrapper. According to (Jović et al. 2015), filter methods us statistical criteria to rank features before choosing the ones with the highest rankings. The t test, chi square test, Wil coxon Mann-Whitney test, mutual information, Pearson correlation, and principal component analysis are widely employed filtering techniques in academic research.

Although affective, filter methods may miss feature interaction and redundancy. It is unclear what precise cut-off for feature importance should b usd to distinguish btwn important features and nois.According to (Panthong & Srivihok 2015), Wrapper approaches, which offer a more complete approach but frequently come with a higher computing cost, evaluate features based on how well they perform in a predictive model.The learning algorithm is applied to specific sets of features, and on a different set, it is evaluated. The feature set's quality is assessed using its ability to predict outcomes accurately. Wrapper methods generally perform better than filter methods. However, In order to mitigate the computational intensity associated with exhaustive searches, it is important for wrapper techniques to employ a search algorithm to effectively discover the optimal collection of characteristics.

2.3 Models Based On Feature Selection

2.3.1 Filter

According to (Yildirim 2015), the filter method of feature selection acts independently of any particular learning algorithm, in contrast to its peer methods. Instead, it evaluates features based on the inherent characteristics of the data, especially the statistical correlation between each feature and the targeted variable. High correlation features are thought to be essential for predicting the target.In a situation when there is a need to avoid the requirement for repetitive model training and instead concentrate exclusively on data analysis, filter methods are noticeably more effective and scalable than wrapper approaches. They are suitable for processing massive datasets because of their inherent design, which guarantees lower computing complexity. In essence, wrapper approaches use feature omission and iterative model training to determine the relevance of a feature makes him computationally expensive.By considering into account the illustration provided by(Wang et al. 2015). Huang and his team, they were able to estimate the appropriateness of antidepressant drug duration by using a filter-based feature selection method on a database of medical claims. To choose important features, they employed discretization and the inconsistency rate measure. For predictions, decision tree and logistic regression techniques were used. According to their research, the filter-based technique effectively decreased the complexity of healthcare databases, speeding data processing by concentrating on the most important features in the vast set of records.

Pearson Corelation

The Pearson correlation coefficient, sometimes known as Pearson's r, is a statistical measure that quantifies the linear relationship between two variables. The Pearson correlation coefficient is utilised to ascertain the magnitude and direction of the linear association between two continuous variables. It returns a value ranging from -1 to 1.

$$r = \frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i-\bar{X})^2 \sum_{i=1}^{n}(Y_i-\bar{Y})^2}}$$

n is the number of data points. XiXi and YiYi are the individual data points. XX and YY are the means of the XX and YY variables, respectively.

•The linear relationship between each attribute and the target variable is measured using Pearson correlation.

•Features with a high absolute correlation value (close to 1 or -1) are considered more relevant for prediction since they have a strong linear relationship with the target.

•Features with value close to 0 will be considered for removal, as they show no or very little linear association with the target.

Mutual Information

Mutual Information (MI) is a measurement derived from information theory that expresses how much knowledge about one variable is gained by monitoring another variable. In simpler terms, it measures how much uncertainty regarding the value of another variable is reduced when one variable's value is known.

Advantages of using Mutual Information in filter method:

•Non-linear Relationships: MI can capture both linear and non-linear dependencies, as contrast to Pearson correlation, which only records linear interactions.

• No Assumption: MI doesn't assume any particular type of relationship (e.g., linear) between variables.

Fisher Score

The Fisher Score is a statistical metric used to rank features as per their discrimination ability. It calculates the proportion of dispersion within class scatter to between class dispersion. it helps us to understands how features distinguishes between data points from various classifications. Feature with a higher Fisher Score is more capable of greater sepration.

$F(x) = \frac{\sigma_1^2+\sigma_2^2}{(\mu_1-\mu_2)^2}$

where mu1 and mu2 are the feature means value with respect to two classes, and sigma1 and sigma2 are the variances.

Univariate Feature selection

Univariate evaluates each feature based on the relation to the target variable.In this process one feature test against the target variable, while ignoring any potential combined effects of several characteristics.

Depending on the type of data, various methods can be applied for univariate feature selection:

- For continuous features and a continuous target: Pearson correlation coefficient.

- For categorical features and a categorical target: Chi-squared test.
- For continuous features and a categorical target: ANOVA F-test.

2.3.2 Wrapper Methods

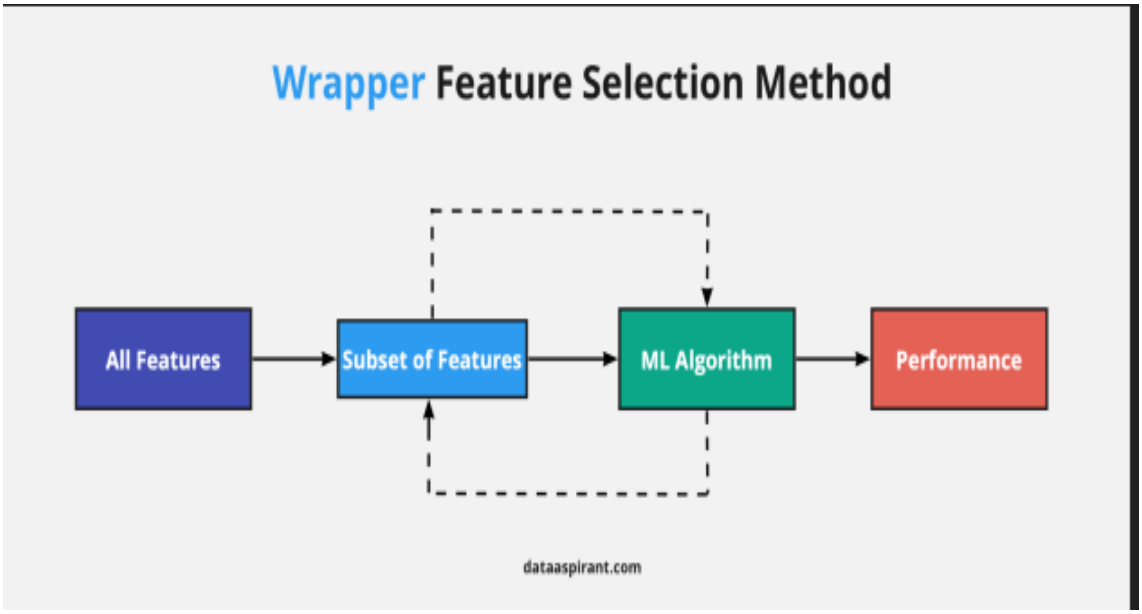


Figure 2.1: Schematic diagram based on Wrapper

Wrappers examine feature subsets based on classifier overall performance for responsibilities like Naive Bayes or SVM and clustering obligations the use of clustering algorithms like K-way and modelling algorithms as black box evaluators. As per Jović et al. (2015)For each subset, the evaluation procedure is repeated, mimicking the dependence on search method seen in filters. Wrappers rely on the requirements of the modelling process, hence they are computationally slower than filters.However,

because they compare the usage of actual modelling strategies, they frequently outperform filters in terms of performance. Their bias in favour of the particular modelling approach used for assessment is a flaw that makes it important to employ an independent validation sample and a separate modelling approach on the way to acquire a sincere generalisation blunders estimate. In spite of the truth that wrappers can be used with any seek strategy and modelling algorithm, they performed well while mixed with greedy search techniques and short algorithms like Naive Bayes, linear SVM, and Extreme Learning Machines (Benoît et al. 2013).

SFS and SBS

As per Fahmiin & Lim (2020)Sequential forward selection (SFS) and sequential backward selection (SBS) are two methods commonly employed for feature selection in machine learning. These techniques utilise the performance of a classifier to identify the most optimal subset of features that yield the highest quality outcome..An empty feature set is used as the starting point for the feature addition process known as sequential forward selection (SFS). The feature that best enhances the performance of the model is chosen for each iteration. Until the target number of features is attained or performance stops advancing, this process is continued.The search comes to an end when the evaluation function is not improved by the addition of additional features.

```
let complete dataset: D={d1,d2,d3,...,dn}

let new subset: S = { }

for k iterations do

    sadd=bestF(S+s)

, where s  D { S

S=S+sadd

k=k+1
```

In SBS, the process operates in reverse, where a feature that, upon removal, contributes to the best outcome for the classifier performance is eliminated from the feature subset. The ANN is then used to calculate the score metrics for the final optimal subgroup.

```
let complete dataset: D={d1,d2,d3,...,dn}

let new subset: S = D

for k iterations do

    sminus=bestF(Ss)

, where s  S

S=Ssminus

k=k+1
```

Recursive Feature Selection (RFE)

The goal of RFE is to analyse the model, determine which features are most important to its performance, and then recursively eliminate features from it. RFE reduces the most important features for the model by performing this iteratively.This approach involves reconstructing the model and iteratively removing features of least importance until the desired number of inputs is obtained.The relevance of each feature is then assessed by the absolute values of its coefficients in models like logistic regression or SVM for classification based problem, Considering the case suggested by Tosin et al. (2017) RFE used the SVM initially designed for the binary classification problem which is used for gene detection in cancer detection initially RFE trains a model utilising all of the available features then retrained using the smaller feature set after the least significant feature is eliminated. This procedure iterates until a predetermined stopping criterion, such as a predetermined amount of features or a performance threshold, is fulfilled, continuously deleting the least significant feature and retraining the model. The final features are thought to be the most important one for the model after the

RFE procedure.

What makes RFE seprate from traditional SBS

SBS and RFE are different in that SBS is frequently based on a criterion like model performance. However, once a feature is eliminated, its significance is not taken into account in further revisions.SBS might not capture changing feature interactions as efficiently like RFE because it does not re-evaluate the model after removing each feature whereasRFE was created especially for algorithms like support vector machines (SVM) that weigh or prioritise features. Based on these weights, the features are sorted, and the one with the lowest weight is eliminated.RFE can better account for the interactions between features and their overall relevance by retraining the model after each feature removal make them more efficient.Which also leads to increases in the computational cost of RFE relative to SBS.

2.3.3 Embedded Methods

Embedded and wrapper share some similarities . The choice of features and classification methods are related.Compared to wrapper methods, this connection is stronger with embedded methods. Embedded methods may be formed by combining filter and wrapper methods.Embedded approaches utilise classification algorithms that include inherent feature selection capabilities..As per Maldonado & López (2018) Random Forest, and Regression Tree (CART) are three decision tree algorithm-based embedded feature selection approaches (Liu et al. 2019).The least absolute shrinkage and selection operator (LASSO) regression l1 regularisation is another widely used method that imposes a penalty equal to the absolute value of the magnitude of the coefficients.

The difference between Hybrid Filter-wrapper and embedded methods is that hybrid use filter to reduce feature space then apply wrapper methods on this reduced set whereas embedded is an integrated feature selection process support model training process and select features while training using lasso regression by adding penalty for non zero coefficients.Since in this thesis we mainly focus on the hybrid filter-wrapper and wrapper for feature selection and emdedded is outside of this scope.

2.3.4 Hybrid Based Filter-Wrapper

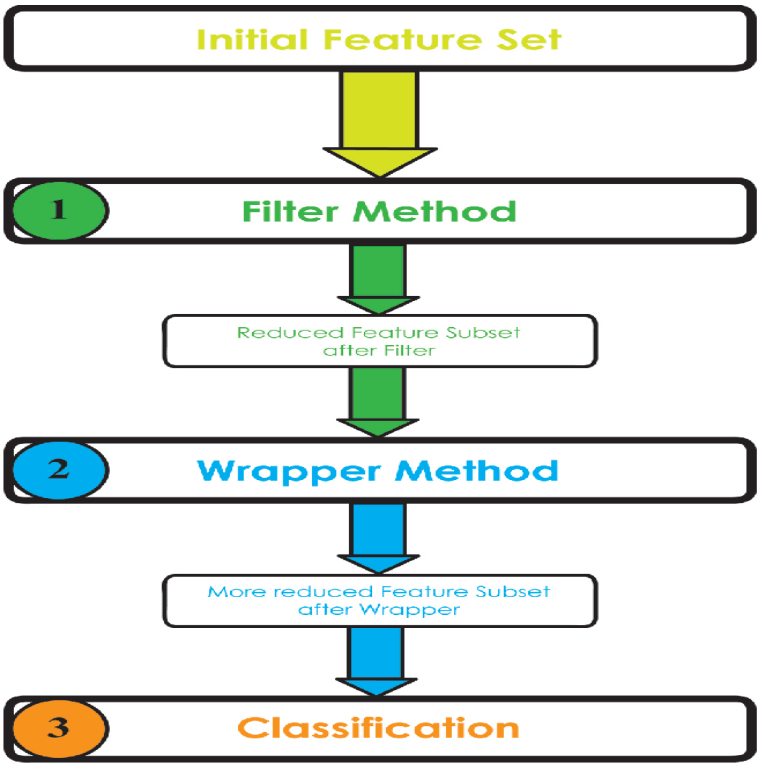


Figure 2.2: Schematic Diagram of Proposed Hybrid[Filter-wrapper] method

There are four primary classifications of feature selection strategies, including filter, wrapper, embedding, and hybrid approaches.(Singh & Singh 2021) By evaluating and ranking features based on statistical analysis using pearson correlation, information gain, fisher score, uni variate, etc., the filter approach is applied as classifier independent, utilising the intrinsic qualities of data. Filter approaches are more advantageous for handling high-dimensional data set issues since they are quick, computationally straightforward, and affordable. While employing a specific classifier as the important measure, wrapper approaches, on the other hand, produce candidate feature subsets by iterative scanning the entire feature space. Although these algorithms yield encouraging results, one drawback is the growing computing complexity of the feature space.Wrappers typically offer better results than filters because the FS process is tailored for a specific learning algorithm. In order to improve the Feature selection technique, hybrid methods combine the use of filter and wrapper methods in a single context(Ansari et al. 2019).The hybrid method proposed by (Agarwal & Mittal 2013) suggested the initial step involves the elimination of unnecessary and noisy features using the Information Gain (IG) technique. Subsequently, the reduced feature set is

subjected to rough set attribute reduction, followed by a wrapper approach for fine-tuning in order to get an ideal feature set.

In their respective studies,Xie & Wang (2011) aimed to increase classification accuracy by combining a two-step feature selection process with the improved F-score as a filter technique and sequential forward search as a wrapper method. In contrast, Peng et al. Peng et al. (2010) introduced a pre-selection step involving random sampling as a filter method to reduce computational costs in subsequent wrapper methods. The hybrid method proposed by (Agarwal & Mittal 2013) bringing together Information Gain (IG) and Rough Set Attribute Reduction (RSAR), using a two-tiered filtering procedure to remove unnecessary and noisy features, with the ultimate goal of producing an ideal feature set.. Lastly, Yousefpour et al. (Yousefpour et al. 2017) With the aim of utilising the strengths of multiple filter methods while enhancing feature selection for greater accuracy, integrated feature sets from various filter-based methods using Frequency-Based Integration of Feature Subsets and Ordinal Based Integration of Feature Vectors also knows as OIFV, followed by wrapper-based selection of the best feature subset. Collectively, these methods show many methods for feature selection in machine learning applications.

Recursive Feature Elimination (RFE), the second suggested strategy, is used to improve the performance of our classifier by removing redundant features that have been chosen via filter methods. In contrast to more straightforward univariate methods, RFE is a multivariate feature selection technique that makes use of a wrapper model that is based on the classifier itself. Although RFE is strong, its iterative nature makes it computationally more expensive.The RFE process operates iteratively, employing a backward feature elimination strategy.Features are sorted according to the absolute values of their classifier weights during each iteration. Then, a specific amount of features with the lightest weights are eliminated. The problem's requirements dictate how many features should remain after repeating this process.The fact that basic RFE implies knowledge of the ideal feature set size in beforehand, which is sometimes an unrealistic assumption, presents a considerable barrier. Recursive Feature Elimination with Cross-Validation (RFECV), which combines cross-validation with the RFE procedure to overcome this constraint, is the result. By fitting the model several times,

RFECV assesses several feature subsets, with each step entailing the elimination of the weakest features as determined by their feature significance scores inside the fitted model.RFECV reduces the need for prior knowledge of the ideal feature set size by including cross-validation. The algorithm adapts in real time to the available data, aiding in the identification of the most advantageous collection of attributes that maximizes the classifier's performance. The use of the RFE (Recursive Feature Elimination) and cross-validation methodology is a robust technique that holds significant potential in the realm of feature selection, classifier performance optimization, and prediction accuracy enhancement. This methodology may be utilized in many situations when the precise size of the feature collection remains uncertain.

2.4 Limitations of Feature Selection

Due to a number of reasons, feature selection approaches frequently show a bias towards large-scale and computational consumption. First off, datasets have drastically increased in size and feature count in many real-world applications. Despite the fact that these traits could contain important information, processing and analysing them can be laborious and time-consuming. In order to evaluate the relevance of each feature, feature selection approaches have a propensity to favour large computational resources, which might be problematic in systems with limited resources.Second, some sophisticated feature selection methods use iterative model training and evaluation procedures, such as recursive feature elimination with cross-validation (RFECV). When working with huge datasets, these iterations can considerably increase processing needs. Such methods may be chosen by researchers and practitioners in the hope of achieving optimal model performance by carefully analysing various feature subsets, even though they have higher processing costs.Furthermore, the computational overhead of feature selection increases as machine learning models get more complex and require larger amounts of data for training. This preference for computational consumption over efficiency in feature selection reflects efforts to increase model generalisation and accuracy, which are frequently given preference over efficiency, especially in situations where computer resources are relatively available.

2.5 Strategies for Addressing Feature Selection Limitations

Hybrid filter-wrapper and wrapper feature selection methods provide distinct strategies for addressing challenges arising from large datasets, overfitting risks, and computing resource utilisation. The advantage of large-scale datasets is effectively countered by hybrid filter-wrapper methods, which start with a filter-based strategy to compress the feature space. The ensuing wrapper stage, though potentially computationally demanding, makes cautious feature subset selections based on iterative model evaluations, which helps reduce the danger of overfitting and justifies the computing investment by putting a focus on actual model performance.Wrapper approaches, on the other hand, are very skilled at handling overfitting issues. They continually develop and test models using different feature subsets, successfully identifying feature subsets that improve generalisation and reduce overfitting hazards. Wrapper approaches need additional optimisation work, such as parallelization or sampling strategies, when working with huge datasets, nevertheless, in order to effectively manage computing resource usage.

2.6 Summary

In conclusion we can say by apply both strategies hybrid and wrapper provide useful solutions to deal with the problems that large datasets, overfitting, and computational demands present. Hybrid strategies balance filtering effectiveness and wrapper precision, while wrapper strategies excel at mitigating overfitting when computational effectiveness is properly optimised. The decision between them is based on the particular dataset characteristics and research objectives.

Chapter 3

Design of the Methodology

This chapter outlines the methods employed in the dissertation project.

3.1 Research Design

The overarching goal of this research is to investigate and quantify the impact of feature selection techniques, specifically focusing on the Hybrid based Filter-Wrapper RFE and Wrapper method, on the performance of prominent machine learning algorithms. Our primary research objective can be stated as follows:

3.1.1 Research Objective

To assess the effectiveness of feature selection methods Hybrid based Filter-Wrapper approach including statistical analysis, pearson correlation ,RFE based on classifiers and Wrapper method feature se-lection techniques, such forward selection, backward elimination, and RFE, in improving the predictive performance of machine learning algorithms, Specifically, this study focuses on the use of Logistic Regression, Support Vector Machines (SVM), Naive Bayes, and Random Forest algorithms for classification tasks using various datasets.

This research objective aligns with the broader aim of advancing our understanding of feature selection's role in machine learning, particularly in classification tasks. By conducting this study, we aim to contribute valuable insights into whether and how Wrapper-based feature selection methods can enhance the overall performance of machine learning algorithms when applied to real-world datasets.

3.1.2 Research Hypotheses

To guide our investigation and provide a basis for empirical testing, we formulate the following testable hypotheses:

Hypothesis 1 (H1):

The incorporation of Hybrid based Filter-Wrapper approach will lead to a statistically significant improvement in the performance metrics (precision, F1 score, accuracy, recall, log loss, ROC, and AUC) of machine learning classifiers on classification datasets compared to the baseline model without feature selection.

Hypothesis 2 (H2):

The inclusion of Wrapper method using SFS,SBS and RFE , , will result in a statistically significant enhancement in the performance metrics (precision, F1 score, accuracy, recall, log loss, ROC, and AUC) of machine learning classifiers on classification datasets in com- parison to the baseline model without feature selection

Hypothesis 3 (H3):

The utilization of Hybrid based Filter-Wrapper and Wrapper feature selection, particularly recursive Feature Elimination also known as rfe , will produce a statistically significant boost in the performance metrics (precision, F1 score, accuracy, recall, log loss, ROC, and AUC) of machine learning models on classification datasets when contrasted with the baseline model without feature selection.

These hypotheses will serve as the foundation for our empirical investigation, allowing us to rigorously assess whether the application of Wrapper-based feature selection techniques contributes to improved machine learning model performance across multiple classification datasets. Through this research, we aim to provide practical insights for practitioners in the field of machine learning and data science regarding the appropriate use of feature selection methods to enhance model outcomes.

3.2 Data Collection

In this section, we provide a comprehensive overview of the datasets used in our study, explaining their sources, data preprocessing steps, and any feature engineering techniques applied.

1. Titanic Dataset: A frequently used benchmark in data analysis and machine learning is the UCI Titanic dataset. Details on the passengers from the 1912 Titanic shipwreck are included. Age, sex, class, fare, and whether the passenger survived the accident are among the features. Predicting survival outcomes using these characteristics is frequently the main objective of this dataset analysis. This dataset provides insightful information on the socioeconomic aspects affecting marine catastrophe survivorship.

2. Heart Dataset: In medical data analysis and machine learning, the heart dataset is a well-liked resource for predicting the existence of heart disease. It includes a number of characteristics, such as age, sex, cholesterol readings, and electrocardiogram data. Understanding important risk variables linked to cardiac illnesses is made easier by the dataset. Researchers and medical professionals are trying to anticipate and identify probable cardiac problems by looking at these characteristics. Its use contributes to improving cardiology's diagnostic precision and preventative treatment.

3.Breast Cancer Dataset: The breast cancer dataset is a well recognised collection that is commonly employed for the purpose of classification in the fields of machine learning and medical research. It has characteristics that were taken from breast cancer biopsy cell nuclei. Cell nuclei's texture, radius, and smoothness are among its characteristics. The main goal of the dataset is to use these characteristics to categorise tumours as benign or malignant. Understanding breast cancer-related tumour features and improving early detection are both aided by the analysis of this dataset.

4.Iris Dataset: A classic in data science and machine learning is the Iris dataset, sometimes known as Fisher's Iris dataset. The dataset comprises 150 samples derived from three distinct species of iris blooms, namely virginica,setosa, and versicolor. Each

sample is characterised by four attributes: sepal length, sepal width, petal length, and petal width. The primary objective of this dataset is to employ the aforementioned metrics in order to classify the iris flowers into one of the three distinct species. It serves as a foundational dataset for data visualization and classification systems.

5.Glass identification dataset Machine learning classification tasks are performed using the Glass Identification dataset. It includes chemical examinations of glass samples, including information on the contents of sodium, magnesium, aluminium, and refractive index. The main objective of the dataset is to classify glass samples into one of three types W, such window glass or container glass. The source of glass fragments found at crime scenes can be identified with the use of this categorization in forensic examinations. The collection provides information about the patterns in chemical composition that identify different varieties of glass.

6:Ionosphere Dataset Machine learning experts frequently utilise the Ionosphere dataset for binary classification problems. It contains properties deduced from radar returns and includes radar data collected from the ionosphere. Determine whether or if there is evidence of a structure in the ionosphere in the signals picked up by the radar is the main objective of this dataset. The properties record data regarding the amplitude and phase of the returning signals. This dataset's analysis assists in comprehending and foretelling ionospheric structures from radar signals.

7:Brain Stroke For stroke prediction research and machine learning, the Brain Stroke dataset is an essential tool. It includes demographic information and health indicators including age, hypertension status, heart condition, and glucose levels, among others, for each individual patient. The major objective of the dataset is to categorise people according to their risk of having a stroke. Researchers can find important risk factors and patterns related to strokes by analysing these characteristics. Its study helps with early diagnosis, preventative actions, and figuring out what causes strokes.

8: Adult Census Income Dataset For categorization tasks in data science, and especially for forecasting income levels, the Adult Census Income dataset is frequently used. Age, education, marital status, and work information of adults, among other things, are included. The primary objective is to use these characteristics as a predictor of whether or not a person earns more than 50,000 USD yearly. The income-influencing

social and economic elements are illuminated by this data collection. Its analysis helps shed light on the causes of pay gaps and the role of numerous demographic factors in determining income.

9:Diabetes Dataset The Diabetes dataset is a large data set used for predicting diabetes risk that has use in both the medical and data science communities. Glucose levels, insulin levels, age, and body mass index are only few of the indications of health included. The primary goal of this dataset is to determine whether or not a patient has diabetes based on these characteristics. Important diabetes-related risk variables can be determined with the use of this data analysis. It's a helpful resource for finding out what causes diabetes early on, providing better treatment to patients, and learning about the full scope of the disease.

10:Wine Dataset When it comes to classification problems, the Wine dataset is a machine learning classic. Included are the findings of a chemical examination of wines from a particular region of Italy, including information about their alcohol percentage, malic acid concentration, and chroma. The major goal of this dataset is to use these characteristics to categorise wines into one of three cultivar groups. Researchers can learn about the chemical features that differentiate apart wine varietals by analysing the dataset. It's a starting point for learning about feature significance and categorization methods.

11:Weather Dataset Data scientists frequently use the Weather dataset to make forecasts about the weather. On a daily basis, meteorological parameters such as temperature, humidity, precipitation, and wind speed are consistently monitored and documented. Predicting meteorological phenomena like precipitation, cloud cover, and sunshine is a common use case for this dataset. Understanding climate trends and patterns across time requires analysis of this dataset. It's a useful tool for scientists studying and forecasting the weather, as well as for researchers and environmentalists.

12:Videogame Dataset Each video game in the videogame is represented by a distinct ID and name. Genre, platform, and year of release are only some of the data provided. The popularity of each game is reflected in its rating. Features like as multiplayer support, total playtime, and visual quality are also present. This data collection is useful for analysing platform-specific performance and seeing patterns

based on the game difficulties which cateogorised as easy , medium and hard.

13:Indian Diabetes Dataset The Indian Diabetes dataset is a popular resource for predicting diabetes in Pima Indians used in scientific studies. Some of the metrics included include blood sugar, insulin, age, and body mass index. The major goal of this data collection is to use these diagnostic characteristics to predict whether a patient will acquire diabetes. This data collection enables the identification of significant risk factors associated with diabetes. This information can be useful for spotting diabetes at an early stage and learning more about the prevalence of the disease in this community.

14:Sonar Dataset The Sonar dataset is a well-known machine learning collection, typically put to use for binary classification projects. It contains sonar sounds collected from various surfaces, such as rocks and metal cylinders. There are 60 properties associated with each record, which describe the relative intensity of sonar echoes at various angles. The primary focus of this dataset is to separate signals reflected from a metal cylinder from those reflected from a rock. Patterns and unique properties of sonar signal returns can be better understood with the help of this dataset's analysis.

15:Student Dataset Insights on student demographics, academic achievement, and behaviours may be obtained from the Student dataset, which sees heavy usage in educational research and machine learning. Age, gender, amount of time spent studying, and previous grades are all typical components. The primary purpose of this dataset is to aid in the prediction of student performance. Researchers and teachers used this information to pinpoint the factors that are most influential in a student's eventual achievement in school. This data collection may be used to better understand student behaviour and develop more effective teaching methods.

16:Loan Dataset Financial analytics and machine learning frequently use the Loan dataset to evaluate creditworthiness. The borrower's income, credit, loan amount, and employment status are all factors. The main goal is to foretell whether or not a borrower will not repay on their debt. By analysing this data, banks can reduce risk in their loan decisions. It sheds light on the variables that determine a person's creditworthiness and their likelihood of repaying a loan.

17:Social Media Advertisement Data from online advertising campaigns, including interactions and results, are collected in the Social Media Advertisement dataset. User

demographics, click-through rates, engagement duration, and purchase history are all examples of the kinds of data that make up a user profile. The major purpose of this dataset is to evaluate ad performance and forecast user purchase behaviour [whether or not users will buy this specific item]. Marketers may use this information to fine-tune ad targeting and identify what motivates users to take action. To improve advertising methods and get the most for the money spent, this data collection is crucial.

18:Stroke Prediction The Stroke Prediction dataset is a vital resource for medical researchers interested in predicting stroke occurrences. Patient characteristics including age, hypertension status, cardiac condition, and glucose levels are common components. The major objective of the dataset is to categorise people according to their risk of having a stroke. Researchers can identify major risk factors for stroke by examining these characteristics. This data set is useful for spotting problems early and learning more about the causes of stroke.

19:Water Dataset When it comes to environmental research and analytics, the Water Quality dataset is essential because of its focus on the evaluation of water's potability. pH, turbidity, dissolved oxygen, and pollutant concentrations are common examples. The primary purpose of this data collection is to ascertain whether or not a certain water supply is drinkable. Researchers can pinpoint the origins of the contamination and evaluate the efficacy of the treatments by looking at these characteristics. For the sake of public health and to direct water treatment initiatives, this dataset is crucial.

20:Bank Authentication Dataset For the sake of both financial security and machine learning, the Bank Authentication dataset is essential. Variance, skewness, and curtosis, all determined by examining bills, are typically included. The primary goal of the collection is to identify fake[0] or authentic[1] currency. Financial organisations can improve their currency verification procedures by analysing this data. The integrity of monetary transactions and the prevention of cash fraud are both

3.2.1 Data Preprocessing

A standardised data pretreatment pipeline was implemented for each dataset in order to assure data quality and consistency.

Data Cleaning

Data integrity remains a significant concern, and soiled data can result in inaccurate decisions and unreliable analysis. Frequent mistakes encompass the absence of values, incorrect wording, inconsistent formats, replicated entries pertaining to same real-world entities, and breaches of business rules (Simões et al. 2013).As per Chu et al. (2016) Before making decisions, experts must consider the effects of corrupted data, as a result, data cleansing has become an important area of database research .Performed data cleaning to identify and address any inconsistencies, outliers, or errors in the datasets. This step involved removing duplicate records and handling any anomalies that could adversely affect the modeling process.

Handling Missing Values

Preprocessing is the primary step in dataset processing.. As per Saar-Tsechansky & Provost (2007) Filling in the missing values in the missing data is a part of the preprocessing stage. A circumstance known as missing values occurs when there is no value in the observation, which causes information to be lost. To handle missing values, we employed various strategies such as imputation, removal of rows or columns with excessive missing data, or using domain-specific methods where applicable. The selection of a technique was contingent upon the characteristics of the dataset and the type of missing values included.

Data Transformation

Azimi et al. (2017) suggested data transformation enables better comprehension of the data as well as the identification of new and fascinating links between its many properties. Data transformation removes noise from data and also summarizes data. Additionally, data processing techniques like normalisation and aggregation are data

transformation processes that might benefit the mining process..Applied data transformation techniques to normalize or scale the data as necessary. This step aimed to ensure that all features were on a consistent scale, which is essential for many machine learning algorithms to perform effectively.

3.2.2 Feature Engineering

Feature engineering is a crucial component in the data preparation process for machine learning. The act of deriving suitable features from given attributes enhances the accuracy of predictions. Feature engineering is the application of various mathematical operations on existing features, resulting in the creation of new features through transformation. Such as Change in a non-linear relationship or scale a feature with the aid of transformations (Nargesian et al. 2017).Feature engineering is a critical aspect of our study as it can significantly impact the performance of machine learning models. For each dataset, i have tried to considered their domain knowledge and applied the following feature engineering techniques:

• Feature scaling (e.g., Min-Max scaling, Standardization).

• One-hot encoding for categorical variables.

• Creation of new features through mathematical transformations when relevant.

• Feature selection techniques other than Wrapper methods, or filter methods by developing logics based on problem.

These steps were carried out meticulously to ensure that the datasets were well-prepared for our experiments, facilitating a fair assessment of the impact of feature selection techniques on machine learning model performance.

By following these data collection, preprocessing, and feature engineering procedures, My aim is to create a robust foundation for our empirical investigation into the efficacy of feature selection across a diverse set of datasets.

3.3 Feature Selection Techniques

Classification in the real world frequently employs supervised learning with indeterminate class probability, and each instance has a class label. Classification in the real

world frequently employs supervised learning with indeterminate class probability, and each instance has a class label. Numerous Features are presented, but many are unnecessary or redundant with respect to the target concept (Tang et al. 2014). Data preprocessing is crucially dependent on feature selection, which encompasses multiple objectives. The idealised method seeks the minimal subset of features that are both necessary and sufficient to define the objective concept. In the classical perspective suggested by Liu & Setiono (2022) the aim is to choose a subset of MM features from NN features (M¡NM¡N) that best optimizes a specific criterion function.While many methodologies place emphasis on improving the accuracy of predictions, others prioritise the approximation of the original class distribution by utilising a small number of characteristics, therefore guaranteeing that the class distribution stays mostly unaltered.

In this section, we delve into the various feature selection techniques that we will evaluate in our study, with a primary emphasis on the Hybrid Filter-wrapper and Wrapper method, including forward selection, backward elimination, and Recursive Feature Elimination (RFE).

3.3.1 Overview

The process of selecting features is a critical component in the construction of efficient machine learning models. Feature selection is a process that involves the careful selection of a subset of pertinent features from the initial feature set. This is done with the aim of enhancing the performance of a model, decreasing dimensionality, and mitigating the potential for overfitting. In our study, we will explore the following feature selection techniques:

Wrapper Method:

The Wrapper method evaluates feature subsets using a specific machine learning algorithm's performance as a criterion. It includes forward selection, backward elimination, and RFE. This method is considered more computationally expensive but often leads to better feature subsets. diagram

Hybrid (Filter-wrapper Method) :

The Hybrid Filter-Wrapper technique is a feature selection approach that integrates both filter and wrapper procedures. Initially, statistical measurements ¡IG,Fisher,Univariate¿ and like Pearson correlation are used to rank and filter out features. A wrapper method is used to further improve these selected parameters while evaluating predicted performance. The efficiency of filter methods and the accuracy of wrapper techniques are integrated in this strategy. Considered to be less computationally expensive leads to better subsets. Diagram

3.3.2 Technique Selection

Statistical techniques and RFE with a classifier will be primarily used to evaluate the Hybrid-based Filter-Wrapper strategy.The Wrapper technique, which incorporates RFE, SFS, and SBS, will also be assessed along with different classifiers.

Rationale

• Analysis : By ensuring thorough evaluation of the Hybrid-based Filter-Wrapper strategy by using both statistical methods and RFE with a classifier. By combining these two approaches, will take advantage of the statistical techniques as well as the classifier's capacity to determine the significance of features.

• Versatility of the Wrapper Technique: Forward selection explores feature subsets by incrementally adding features, while backward elimination does the opposite. RFE systematically ranks and removes features based on the assigned weights. Together, these techniques provide a comprehensive evaluation of feature subsets.

• Classifier Diversity: Ensuring findings are not biased towards any one classifier type by testing the Wrapper approach with a variety of them. By utilizing a variety of classifiers, we may gain a more comprehensive knowledge of the relative relevance of the various features (Parvin et al. 2015).

• Research Focus: Our study aims to assess The influence of feature selection on the performance of machine learning algorithms. Based on two approaches, I can thoroughly investigate the relationship between feature selection and model performance.

Strengths and Weakness

Both Hybrid-based and Wrapper method, as the chosen techniques, has its strengths and weaknesses

• Strengths: - Using a combination method and a variety of classifiers assures that the impact of feature selection on machine learning performance is well-understood, as well as the value of features in general.

• Limitations:-

Computational Intensity: Can be computationally expensive, especially when evaluating multiple feature subsets. Nested models and inappropriate parameters leads to wrong feature selection by algorithm such as SBS.

Overfitting:There is a risk of overfitting to the performance metric used for evaluation. - Algorithm Dependency: Effectiveness may vary with different machine learning algorithms.¡Explain hybrid overfitting and wrapper overfitting¿

3.3.3 Implementation

For Hybrid Filter-Wrapper

Data Initialization: Load the cleaned and encoded dataset for feature subset selection after data preprocessing and label encoding of cateogorical features.

Hybrid Filter-Wrapper Feature Selection: Using statistical analysis and methods like pearson correlation, chi-squared tests, information gain, univariate analysis, etc., choose features depending on their correlation with the goal variable. Use RFE to rank and systematically reduce features based on importance until achieving the desired subset size

Performance Assessment: The model should be evaluated using metrics such as precision, F1 score, accuracy, and AUC for the chosen feature subset.

Full Feature Set Benchmarking : Compare the model's performance using all features with the same metrics.

Iterative Evaluation: Repeat the process for ML algorithms and remaining datasets datasets.

For Wrapper Method

To implement the Wrapper method and its components (forward selection, backward elimination, and RFE), we will follow these steps for each machine learning algorithm and dataset:

Initialize Subset: After the completion of data preprocessing load the encoded dataset for feature subset.

Iterative Process:¡reduce this ¿ - For forward selection: Add one feature at a time, eval- uating model performance at each step until reaching an optimal subset. - For backward elimination: Begin with all features and remove one at a time, evaluating model performance at each step until reaching an optimal subset. - For RFE: Rank features and iteratively remove the least important feature until reaching the desired subset size. ¡mention about classifiers used¿

Performance Evaluation: : Evaluate the performance of the algorithm on the chosen feature subset using appropriate metrics (e.g., precision, F1 score, accu- racy, recall, log loss, ROC, and AUC).

All features: Evaluate the performance of the algorithms including all features using appropriate metrics (e.g., Accuracy,precision, F1 score, ROC, and AUC).

Repeat: the process for each algorithm and datasets.

This implementation approach allows us to systematically assess the impact of the Wrapper-based feature selection techniques on model performance across various datasets and machine learning algorithms, providing valuable insights into their effectiveness.

3.3.4 Model Training

For each machine learning model, we follow a standardized procedure for model training:

Data Splitting:

The dataset is divided into training and testing subsets for both the full feature set and the selected feature subset. This divide makes it possible to validate models on one area while training them on another, ensuring an objective evaluation.

Model Initialization:

Standard hyperparameters are used to initialise the designated machine learning model, ensuring a consistent starting point for all trials, whether they use all features or only a subset.

Training:

All Features: The complete collection of features is utilised in the training set, as the model is calibrated to capture the inherent associations between the features and the target variables.

Selected Features: Simultaneously, the model undergoes training on the training set, only utilising the characteristics that have been selected through the process of feature selection.

Cross-Validation:

Cross-validation is a widely used method in data resampling that is applied to estimate the precise prediction error of models and optimise their parameters (Berrar et al. 2019). The dataset is partitioned into 'k' subsets using K-fold cross-validation. In each cycle, one subset is designated as the test set, while the remaining 'k-1' subsets are employed as the training set. The model undergoes training and testing 'k' times throughout each examination of the test set. The ultimate performance is determined by calculating the average of the 'k' ratings. This methodology facilitates the assessment of the model's efficacy and robustness across diverse datasets.

Performance Testing:

All features: Classification metrics are used to evaluate the model's performance after it has been trained with all of the features Selected Features: Similarly, the performance of the model trained with the selected features is evaluated, allowing for a comparative analysis.

This standardized approach to model training enables us to maintain consistency and fairness in evaluating the impact of feature selection across different datasets and models.

3.3.5 Hyperparameter Tuning

Hyperparameter optimization is crucial since it allows us to decide what parameters to use for each machine learning method with all features and after getting the features . To tailoring datasets including Social Media Ads, Video Games, Indian Diabetes, Sonar, and Loan Prediction for better understanding. Below are the key aspects of hyperparameter tuning in our study:

Importance of Hyperparameter Tuning:

Hyperparameters significantly in- fluence model performance and the effectiveness of feature selection techniques. Proper tuning can lead to improved model accuracy and generalization.

Tuning Hyperparameters Following Feature selection:

Once features are acquired using the hybrid filter-wrapper approach, i have performed hyperparameter tuning In order to enhance the efficacy of machine learning models, many strategies may be employed.. applying hybrid method helps removing unnecessary dimensions, this method streamlines the feature set while simultaneously saving computational time and resources.

Hyperparameter Tuning Techniques:

Various techniques, including grid search with reasonable base size and random search to find optimal hyperparameter configurations for each model and selected datasets. These methods systematically search the hyperparameter space to identify the best settings.

Hyperparameter Selection:

I evaluate the relevant hyperparameters for each model and the chosen dataset while considering how they affect the model's performance. For instance, in SVM, the appropriate regularization intensity and kernel selection are key hyperparameters, while in decision trees, the appropriate regularization intensity and the height of the tree at

its maximum allowing the model to learn relationships that are extremely specific to a given sample, higher depth prevents over-fitting.

Implementation of Hyperparameter Tuning:

This includes defining hyperparameter search spaces, selecting optimization algorithms, and evaluating the model's performance for each configuration.

Evaluation Metrics:

As per Hossin & Sulaiman (2015) Evaluation measures were employed to choose the most suitable model from a range of trained classifiers, with a focus on achieving optimal performance in future scenarios including unknown data. These measurements were also utilised as discriminators throughout the training process, enabling the selection of the most optimal solution from a pool of created solutions. Model performance during hyperparameter tuning is evaluated using accuracy , f1score, roc,auc considering the impact of feature selection on these metrics. This step helps us identify the best hyperparameter configurations for feature-selected and non-feature-selected scenarios. By systematically tuning hyperparameters for each machine learning model, my aim to create optimal conditions for evaluating feature selection's effectiveness. Hyperparameter tuning allows me to find configurations that maximize model performance, providing valuable insights into the impact of feature selection in the most favorable circumstances.

3.3.6 Evaluation Metrics

To quantitatively assess the impact of feature selection techniques on machine learning model performance, we employ a comprehensive set of evaluation metrics tailored to the nature of the datasets (classification or regression). These metrics include:

Classification Metrics

For classification datasets, we evaluate model performance using the following metrics:
Precision: This metric quantifies the precision of positive predictions., i.e., the ratio of true positives to the total predicted positives.

$$\text{Precision} = \frac{TP}{TP + FP}$$

F1 Score: A harmonic mean of precision and recall is a mathematical measure that provides a balanced assessment of both precision and recall. The metric offers a full evaluation of the performance of a given model..

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Accuracy: Measures the ratio of correctly predicted instances to the total instances. It provides a straightforward assessment of overall model correctness.

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

Recall: Measures the proportion of true positives among all actual positives, quantifying a model's ability to identify positive instances.

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

ROC: A graphical representation of the model's ability to distinguish between positive and negative classes across var- ious thresholds. It allows us to assess model trade-offs between sensitivity and specificity.

AUC (Area Under the ROC Curve): Quantifies the ROC curve's overall performance. An AUC of 1 indicates perfect discrimination, while 0.5 suggests random guessing.

These classification metrics provide a comprehensive view of model performance, considering aspects like precision, recall, and the ability to handle class imbalances. By assessing these metrics, we can determine how feature selection techniques affect a model's ability to correctly classify instances.

3.3.7 Experimental Setup and Design

To ensure the validity and reproducibility of our experiments, we establish a rigorous experimental setup:

• **Randomization:** To minimize bias, we employ randomization techniques in data splitting, hyperparameter tuning, and cross-validation. This helps ensure that the results are not influenced by the order of data instances or parameter choices.

• **Control Groups:** Establishing control groups for each machine learning model, consisting of experiments without feature selection. These control groups serve as benchmarks against which we can compare the performance of feature-selected models.

• **Cross-Dataset Analysis:** To assess the generalizability of our findings, we perform a cross-dataset analysis. This involves applying the same feature selection techniques and models across multiple datasets, enabling us to identify patterns and trends.

• **Reproducibility:** I have documented every step of experiments, including dataset sources, preprocessing, hyperparameter configurations, and code. This documentation ensures the reproducibility of results and allows for future validation.

This well-structured experimental setup provides a robust foundation for empirical investigation. It allows us to draw meaningful conclusions about the impact of feature selection techniques on machine learning model performance while accounting for potential sources of bias and variability

3.4 Data Analysis

The data analysexperimental results and their alignment with our research objectives and hypotheses. This phase includes:

•**Visulization** Visual aids, including as charts, graphs, and plots, are employed to enhance the effectiveness of result presentations. These aids facilitate the identification of trends and differences pertaining to chosen variables and all features. .

• **Comparative Analysis:** In this study, we assess the performance of machine learning models with and without feature selection on various datasets and hyperparameter combinations..

• **Statistical Tests:** We conduct statistical tests, including t-tests and ANOVA, to determine whether the observed performance differences are statistically sig- nificant after getting the subset of features .¡modify this

• **Pattern Identification:** We analyze the patterns in the data to identify scenarios where feature selection techniques consistently improve or degrade model performance

• **Version Control:** Using GitHub private to store and keep track of any modification .based on applying various approaches.

The implications of the results will be discussed in the context of our research objectives and hypotheses. Through this data analysis process, we aim to draw meaningful conclusions about the efficacy of feature selection in improving machine learning model performance.

3.5 Summary

The present chapter has provided a detailed description of the thorough methodology that has been developed for our empirical study, which aims to examine the effectiveness of feature selection strategies in enhancing the performance of machine learning models.. We have discussed the research objectives, hypotheses, data collection, feature engineering, feature selection techniques, machine learning models, evaluation metrics, experimental setup, and data analysis procedures. This methodology provides a structured framework for conducting our research and evaluating the impact of feature selection techniques systematically.

The next chapter will present the experimental results and their analysis, shedding light on whether the hypotheses formulated in this chapter are supported by empirical evidence.

Chapter 4

Research and Experiments

4.1 Introduction

The complex environment of machine learning consistently emphasizes the essentiality of feature selection. Ensuring model interpretability is of utmost importance, as it not only enhances the understanding of the model but also helps in reducing computational expenses, mitigating overfitting, and often improving model performance. In this study chapter, we undertake a comprehensive investigation of two separate feature selection paradigms.

The initial methodology employed is a hybrid filter-wrapper strategy. Within the realm of filtering, we leverage the capabilities of many statistical techniques, including Information Gain (IG), Fisher Score, and Univariate feature selection. The prioritisation of features in these techniques is based on their inherent statistical qualities and their link with the target variable. In addition to the aforementioned methods, the Pearson Correlation is employed to assess the presence of linear associations among variables, while the Recursive Feature Elimination (RFE) is utilised to iteratively eliminate the least significant features. The integration of these filter techniques with wrappers guarantees a resilient and all-encompassing feature selection procedure.Moreover, after performing feature selection on certain datasets .oreover, after performing feature selection on certain datasets, i had perform the hyperparameter tuning to improve the model accuracy

.

The other approach primarily centres around the utilisation of wrapper methods, such as Sequential Forward Selection (SFS), Sequential Backward Selection (SBS), and Recursive Feature Elimination (RFE). The aforementioned approaches provide a distinct

capacity to assess subsets of features by considering their performance in conjunction with a certain model, so ensuring that the selected subset is compatible with the designated algorithm.

Our evaluation was conducted with the utmost rigour and included the use of 20 datasets that were diverse. This study aimed to evaluate and compare the classification performance metrics of models that were trained using the whole feature set, as opposed to models that were trained using the selected subsets of features after the process of feature selection.

4.1.1 Adjustment to Data Presentation in the Appendix

Certain topics and dataset findings associated to feature selection have been moved to the Appendix portion of this dissertation in the sake of clarity and emphasis on essential conclusions. With this change, the main body will have a more concise presentation while the supplemental part will still have all relevant information.

4.2 Hybrid Method[Filter-Wrapper]

Datasets	Wrapper RFE classifier	Filter method used	Hyperparameter tuning[selected features	Status
Video games	Random Forest	Pearson Coorelation	YES	Done
Indian Diabetes	Univariate RMSE	Statistial method <IG,Fisher,Univariate>	Yes	Done
Sonar	SVM	Statistial method <IG,Fisher,Univariate>	Yes	Done
Student Dataset	Random Forest	Pearson Coorelation	No	Done
Loan Prediction	Decision Tree	Statistial method <IG,Fisher,Univariate>	Yes	Done
Social Media Ads	Random Forest	Pearson Coorelation	Yes	Done
Stroke Prediction dataset	Random Forest	Pearson Coorelation	No	Done
Water quality	Logistic Regression	Chi square Test <Coorelation >	No	Done
Bank Authentication dataset	Logistic Regression	Pearson Coorelation	No	Done
IRISH Dataset	Logistic Regression	Pearson Coorelation	No	Done

Figure 4.1: Hybrid Method Dataset Table

The hybrid filter-wrapper approach effectively combines filter approaches, such as Pearson Correlation, with statistical methods, especially Information Gain (IG), Fisher Score, and Univariate. This integration is achieved by including these techniques into

the wrapper method, which utilises the Recursive Feature Elimination (RFE) classifier. This technique was subjected to thorough testing on 10 datasets, demonstrating its effectiveness in identifying crucial traits. It is worth mentioning that in the case of datasets such as Video Games, Indian Diabetes, Sonar, Loan Prediction, and Social Media Ads, the process of hyperparameter tuning was executed post feature selection. This thorough approach resulted in a further improvement of the model's performance on the top features.

Analyzing Selected Features Probability Using Sigmoid The sigmoid curve illustrates the functional connection between a certain trait and the likelihood of a given result. For feature values that are smaller in magnitude, the curve initiates in close proximity to zero, hence signalling a diminished probability of the desired outcome. As the value of the characteristic grows, the likelihood exhibits a significant increase, eventually reaching a point of inflection. Once a certain threshold is reached, the rate of rise levels off. The decision boundary, commonly set at a probability threshold of 0.5, is used to classify the result. The curve presented in this analysis serves to emphasise the influence of the characteristic in question on the predictive capacity of the result.Evaluating the extent to which a certain feature may serve as a predictor for the target variable and ascertaining if the model has a tendency towards favouring a specific class of the target variable.considering example based on Indian diabetes dataset checking glucose variable with respect to target variable Using a logarithmic equation, logistic regression converts inputs like glucose levels to probabilities between 0 and 1. This equation's result is a projected chance of diabetes for each value of glucose that has been input. For instance, a glucose output of 0.8 at 150 datapoint suggests an 80 percentage likelihood of diabetes.

4.2.1 Case 1 - Video Game Dataset

Class Distribution The target variables may be classified into three separate categories: Easy, Medium, and Hard. This observation suggests that the problem at hand pertains to multi-class categorization. It is worth noting that the "Easy" class exhibits a higher degree of dominance compared to the other two classes.

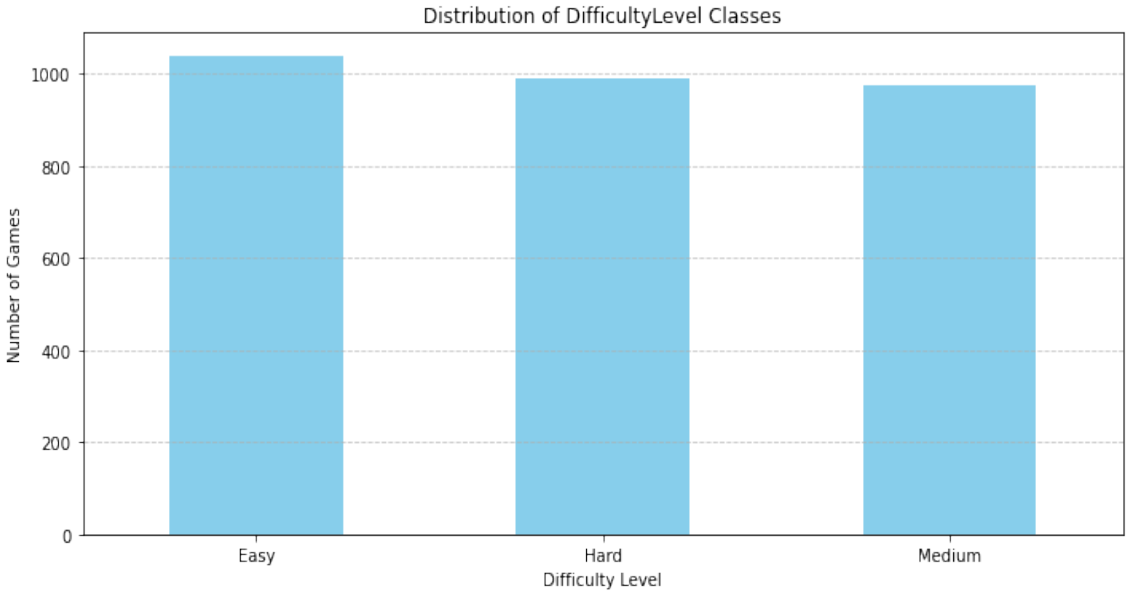


Figure 4.2: Videogame Class Distribution

In summary, the following are three distinct machine learning models accompanied with their respective selected parameters and the corresponding computational time. The calculation time denotes the duration required to train or assess each model using the specified parameters.

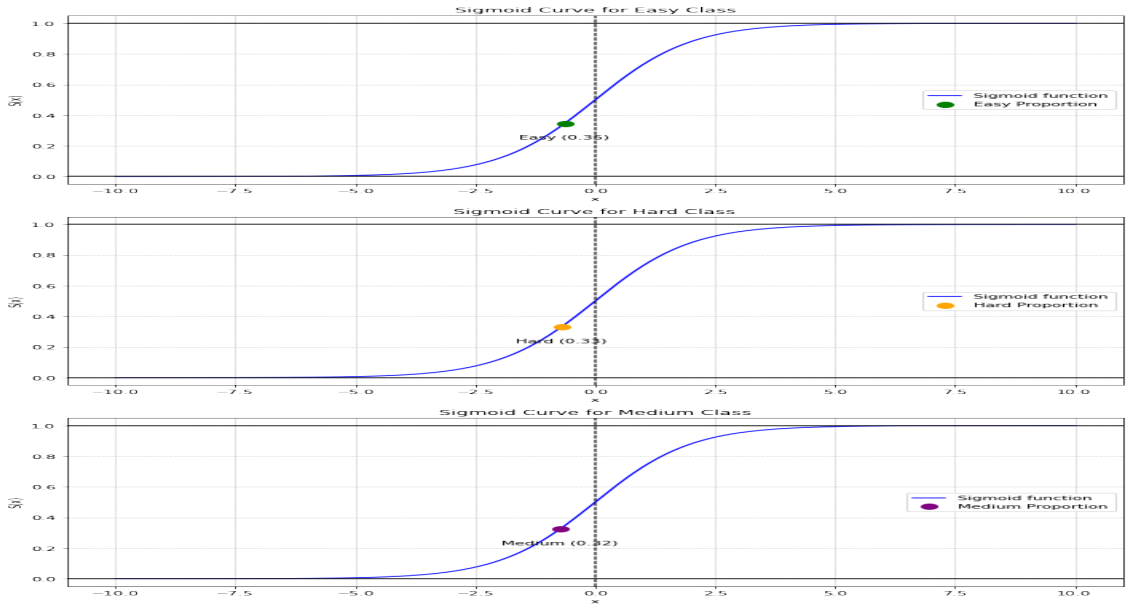
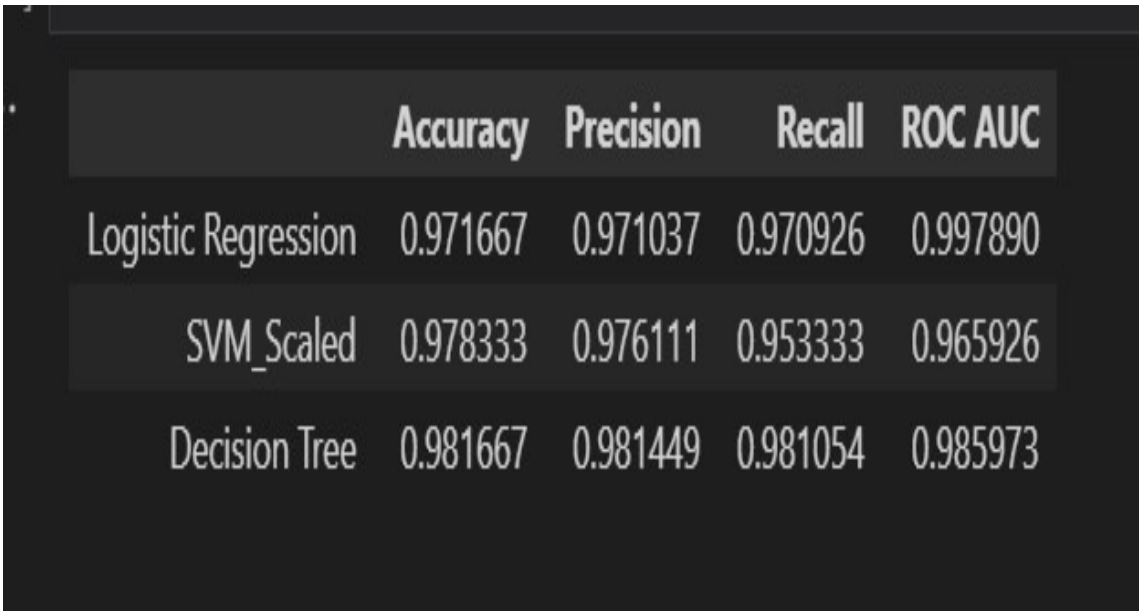


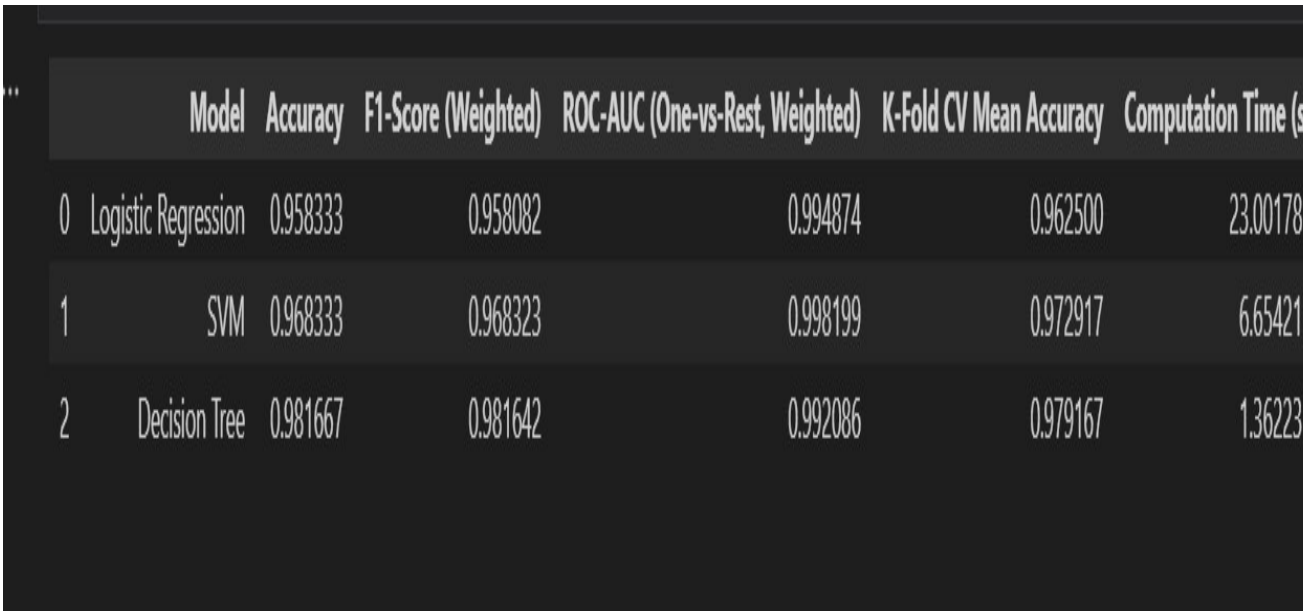
Figure 4.3: Sigmoid Distribution of Selected Features

Conclusions In the context of Logistic Regression, it was observed that the model incorporating all features had a slightly superior performance compared to the model incorporating only selected features, as measure by metrics such as Accuracy, Precision, and Recall. Nevertheless, the receiver operating characteristic (ROC) area under



	Accuracy	Precision	Recall	ROC AUC
Logistic Regression	0.971667	0.971037	0.970926	0.997890
SVM_Scaled	0.978333	0.976111	0.953333	0.965926
Decision Tree	0.981667	0.981449	0.981054	0.985973

Figure 4.4: Performance Metrics Comparison of All Features



	Model	Accuracy	F1-Score (Weighted)	ROC-AUC (One-vs-Rest, Weighted)	K-Fold CV Mean Accuracy	Computation Time (s
0	Logistic Regression	0.958333	0.958082	0.994874	0.962500	23.00178
1	SVM	0.968333	0.968323	0.998199	0.972917	6.65421
2	Decision Tree	0.981667	0.981642	0.992086	0.979167	1.36223

Figure 4.5: Performance Metrics Comparison of Selected Features

Figure 4.6: Bar Plot Comparison of Selected Feature

the curve (AUC) values show minimal variation.

For SVM, the model with all features had a higher Accuracy but a slightly lower ROC AUC compared to the model with selected features.The Decision Tree had identical accuracy for both sets of features, but the model with all features had a slightly lower ROC AUC compared to the one with selected features.

The models trained on all features generally performed slightly better or comparably to the models trained on selected features. However, if computation efficiency and simplicity are considered, using selected features might be advantageous, especially if the drop in performance is minimal. Given that the computation time for models with selected features is provided, one could also weigh the benefits of reduced computation time against the slight drop in some performance metrics.

4.2.2 Case 2: Sonar Dataset

Class Distribution The target variables may be classified into two separate categories: metal cylinder 1 and cylindrical rock which is zero. This observation suggests that the problem at hand pertains to binary class categorization. It is worth noting

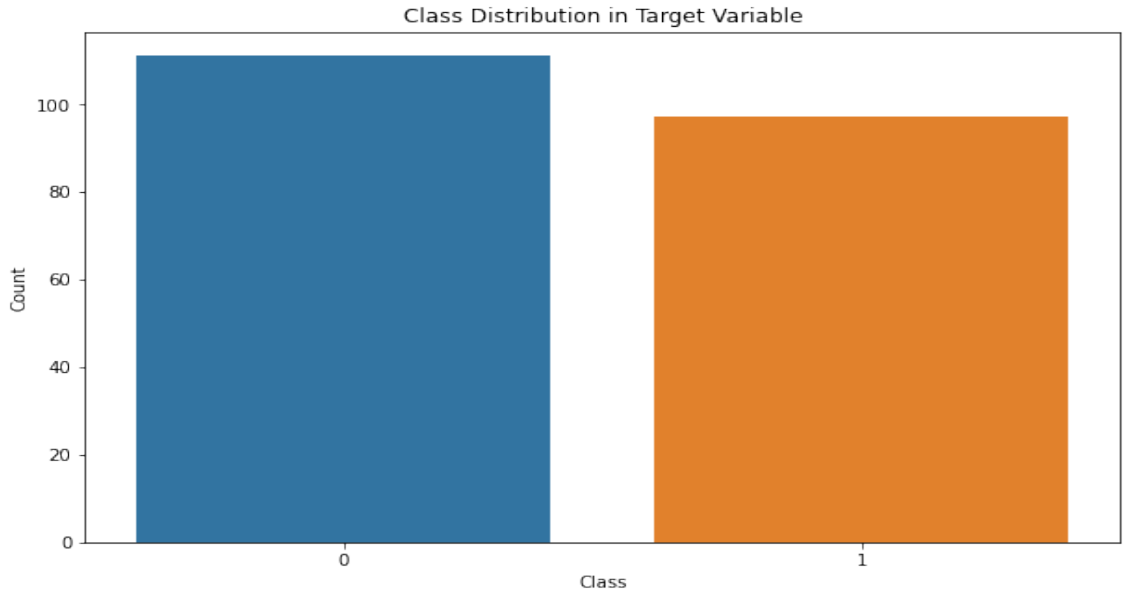


Figure 4.7: Class Distribution Diagram

that the cylindrical rockclass have a slighter dominance compared to the metal cylinder class. But still class 0 don't cause any bias because difference is not much.

Hyperparameter Tuning Outcome: Logistic Regression Model: Regularization Strength (C): 10 -Penalty Type: L1 Solver Used: liblinear Computational time: 4.2554

Support Vector Machine (SVM) Model: Regularization Strength (C): 10 Gamma: scale Kernel: rbf Computational time:: 0.3477

Decision Tree Model: Criterion: gini Maximum Depth of the Tree: None Minimum Samples at a Leaf Node: 2 Minimum Samples Required to Split: 5 Splitting Strategy: random Computational time:: 0.7565

Results based on All features SVM displayed high accuracy but required more time to compute than the Decision Tree. The ROC-AUC results for Logistic Regression showed its superior class distinction capabilities. Although Decision Tree calculated the fastest, it did poorly on other measures.

Results based on Selected Features Accuracy and F1 Score were best for SVM and Logistic Regression, with the latter marginally outperforming SVM in ROC AUC.

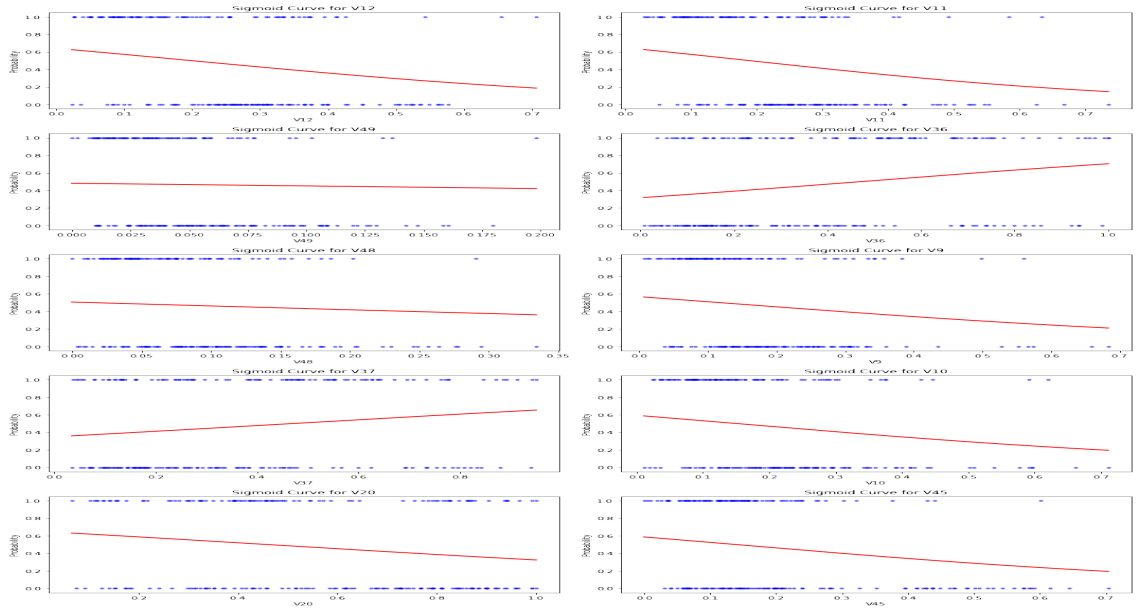


Figure 4.8: Sigmoid Diagram of Selected Features

	Logistic Regression	SVM	Decision Tree
CV Score (Mean)	0.753476	0.741176	0.746881
Accuracy	0.785714	0.833333	0.714286
Precision	0.666667	0.736842	0.611111
Recall	0.875000	0.875000	0.687500
ROC-AUC	0.937500	0.915865	0.709135
Computational Time (s)	0.073877	0.058514	0.037999

Figure 4.9: Performance metrics based on All features

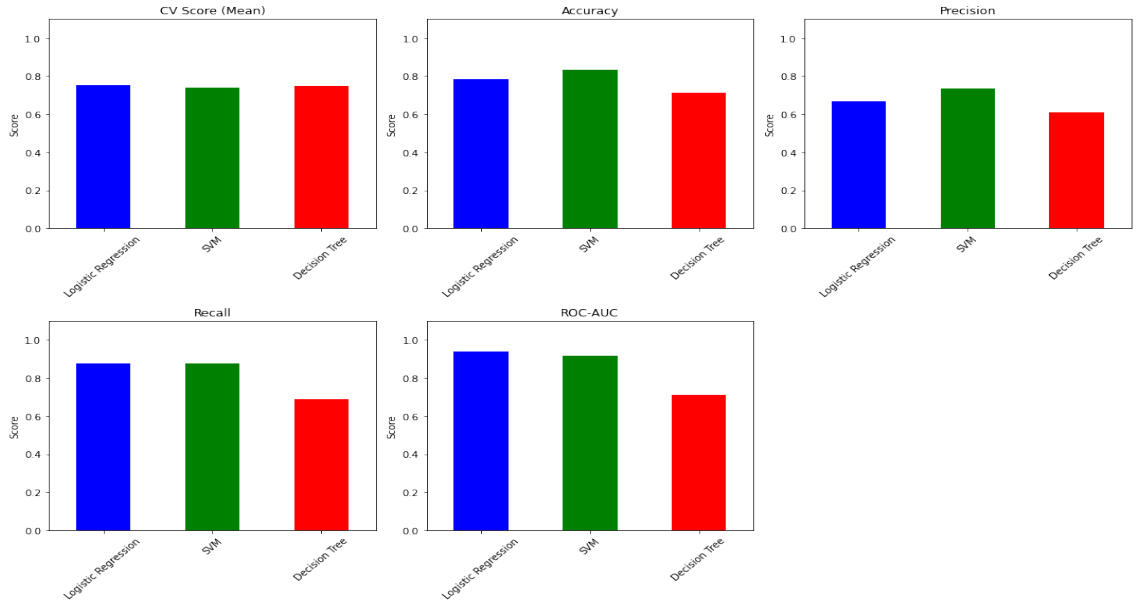


Figure 4.10: Performance metrics based on All Features

	Model	Accuracy	F1 Score	ROC AUC
0	Logistic Regression	0.809524	0.777778	0.930288
1	SVM	0.809524	0.777778	0.923077
2	Decision Tree	0.738095	0.717949	0.745192

Figure 4.11: Performance metrics based on Selected Features

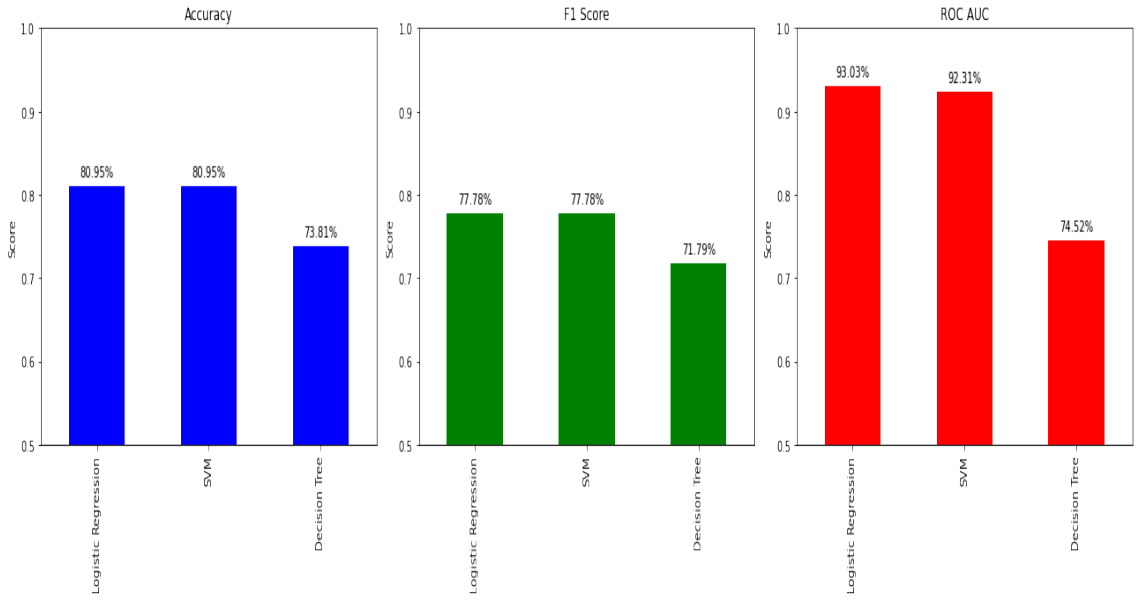


Figure 4.12: Performance metrics based on Selected Features

Decision Tree was computationally moderate but had the lowest scores.

Conclusion

Model Performance had been improved after the feature selection using hybrid based approach by comparing with all features .

4.2.3 Case 3: Student Dataset

Class Distribution The target variables may be classified into three separate groups, namely Low, Medium, and High. This observation suggests that the problem at hand pertains to the domain of multi-class categorization. Significantly, the group labelled as "Medium" appears to have a higher prevalence compared to the remaining two classifications.

Result Summary: The model utilizing the selected features for the Support Vector Machine (SVM) demonstrates evident superiority in terms of both performance and efficiency. In the context of Logistic Regression, it has been shown that the all-features model has superior performance in terms of accuracy and recall. However, it is worth noting that this model requires a longer execution time. In terms of performance metrics, the Decision Tree model with the specified characteristics exhibits a minor improvement.

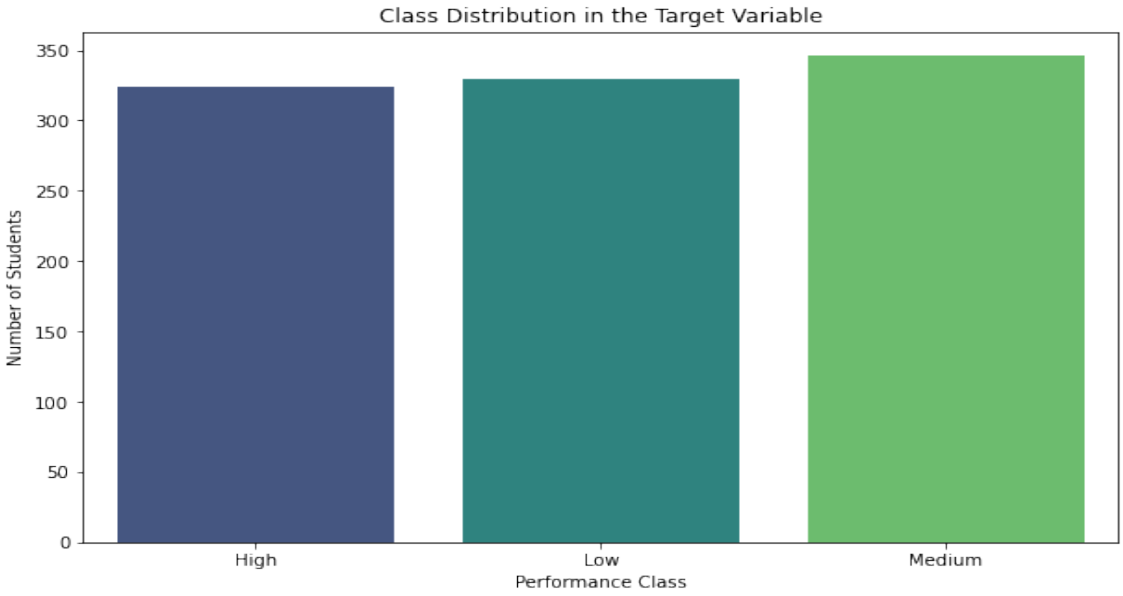


Figure 4.13: Class Distribution Diagram

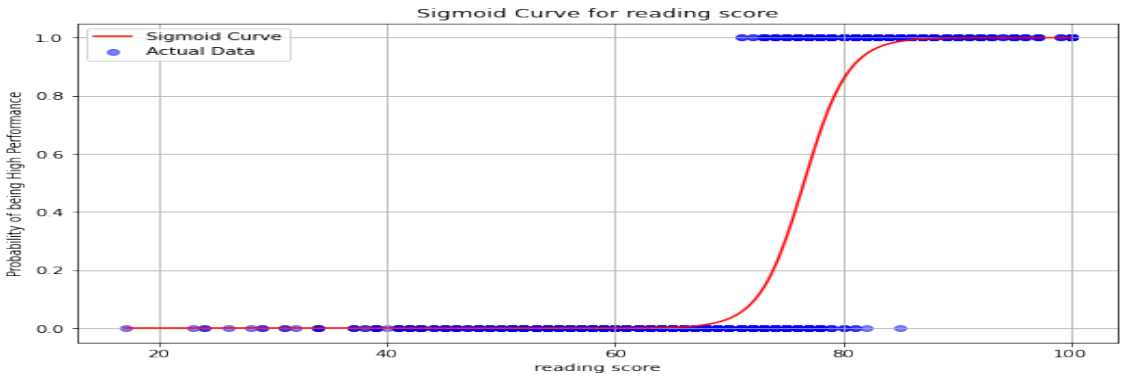


Figure 4.14: Sigmoid Diagram of Selected Features

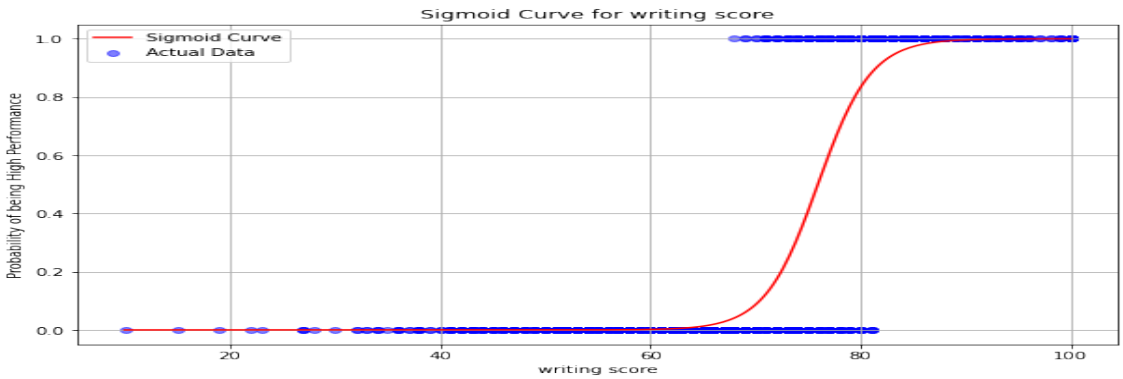


Figure 4.15: Performance metrics based on All features

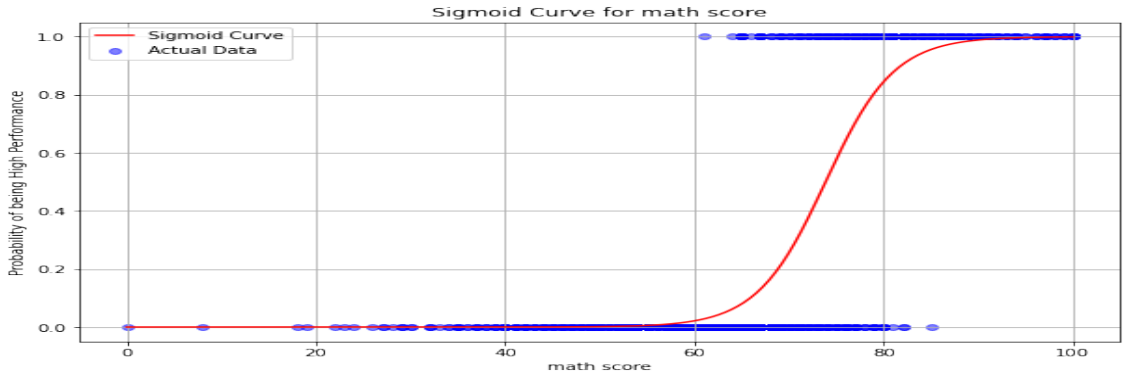


Figure 4.16: Performance metrics based on All Features

	Model	Accuracy	Recall (macro)	ROC AUC (macro)	Execution Time (s)
0	Logistic Regression	0.99	0.989415	0.999884	1.758399
1	SVM	0.96	0.961149	0.999884	0.169787
2	Decision Tree	0.93	0.928891	0.947139	0.028597

Figure 4.17: Performance metrics based on All Features

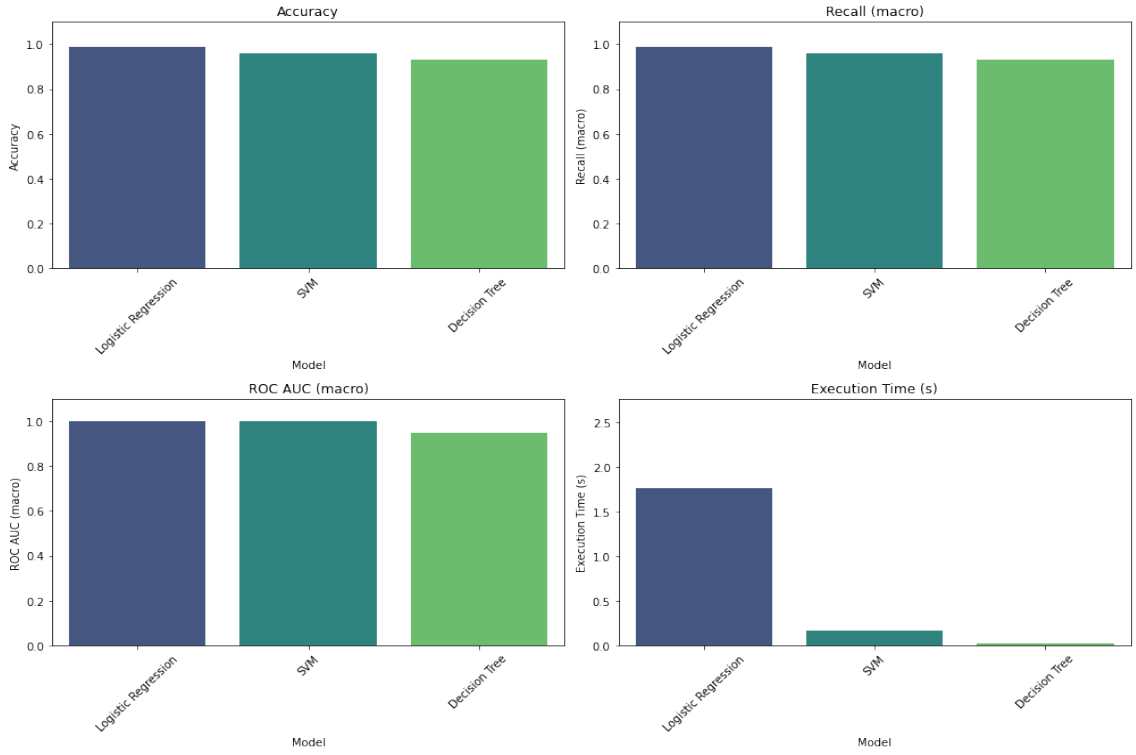


Figure 4.18: Performance metrics based on All Features

	Model	Accuracy	F1-Score	ROC AUC
0	Logistic Regression	91.0%	93.94%	99.988%
1	SVM	100%	100%	100%
2	Decision Tree	94.5%	94.39%	95.85%

Figure 4.19: Performance metrics based on Selected Features

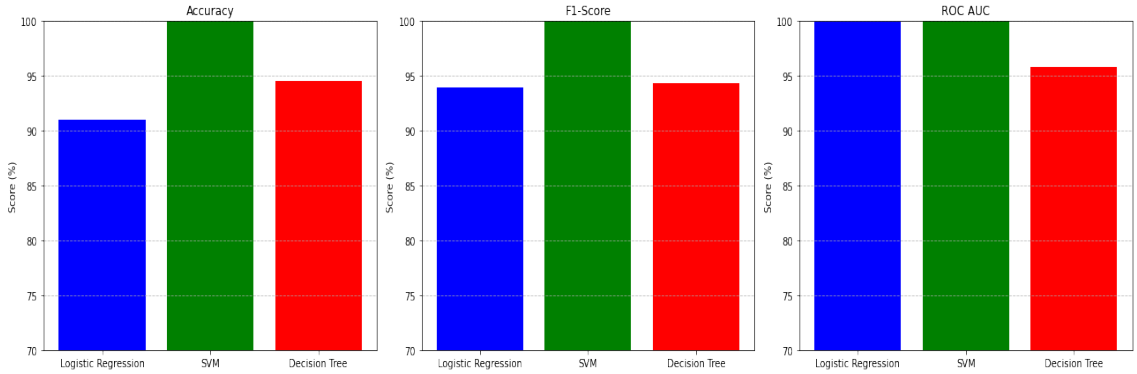


Figure 4.20: Performance metrics based on Selected Features

4.2.4 Case 4:Loan Prediction Dataset

Class Distribution: The target variables may be classified into two unique categories: "Yes" indicating sanctioned loans, which is later encoded as 1, and "No" indicating disapprovals, encoded as 0. This observation underscores the fact that the topic under consideration may be classified into two distinct categories. It is worth noting that the dataset exhibits a greater frequency of "Yes" approvals in comparison to disapprovals. The dataset has a predominant prevalence of "Yes" responses for the goal variable.

Conclusion: In terms of the primary assessment metrics, models trained on all features often exhibit superior performance or achieve comparable results compared to models trained on chosen characteristics. Nevertheless, it is worth noting that models that are trained on specifically chosen features have a tendency to exhibit quicker computational performance, a characteristic that is particularly prominent in the Support Vector Machine (SVM) model. If the main focus is on achieving optimal performance, then utilising all available features is advantageous. If prioritising computing efficiency,

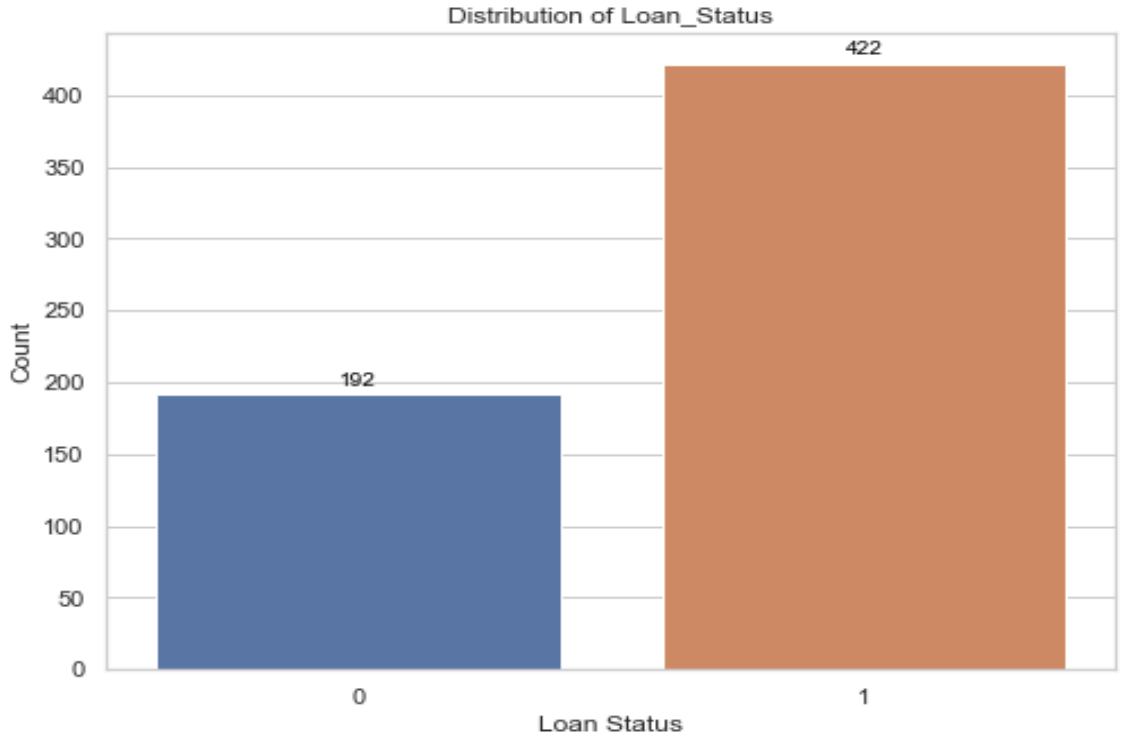


Figure 4.21: Class Distribution Diagram

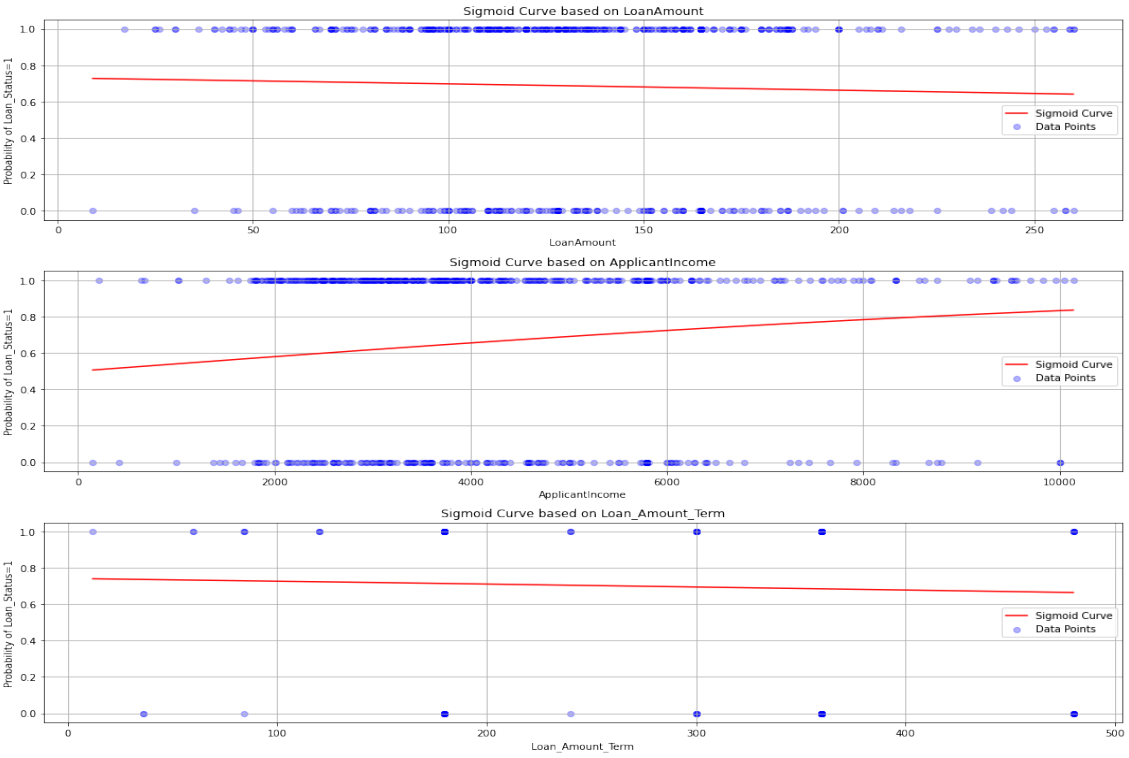


Figure 4.22: Sigmoid Diagram of Selected Features

	Model	Cross-Validation Mean Score	Computational Time (s)	Test Accuracy	Test Recall	Test ROC AUC
0	Logistic Regression	0.820487	0.216993	0.783784	0.983333	0.699359
1	SVM	0.703987	0.094825	0.648649	1.000000	0.500000
2	Decision Tree	0.689867	0.040564	0.724324	0.825000	0.681731

Figure 4.23: Performance metrics based on All Features

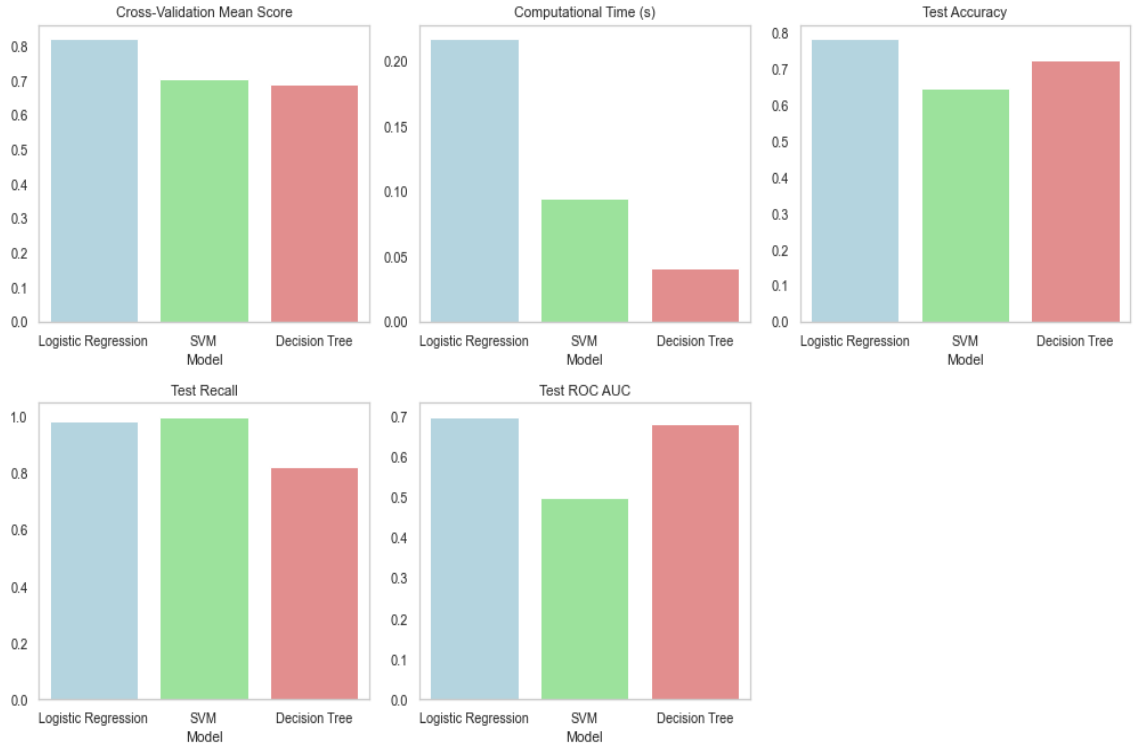
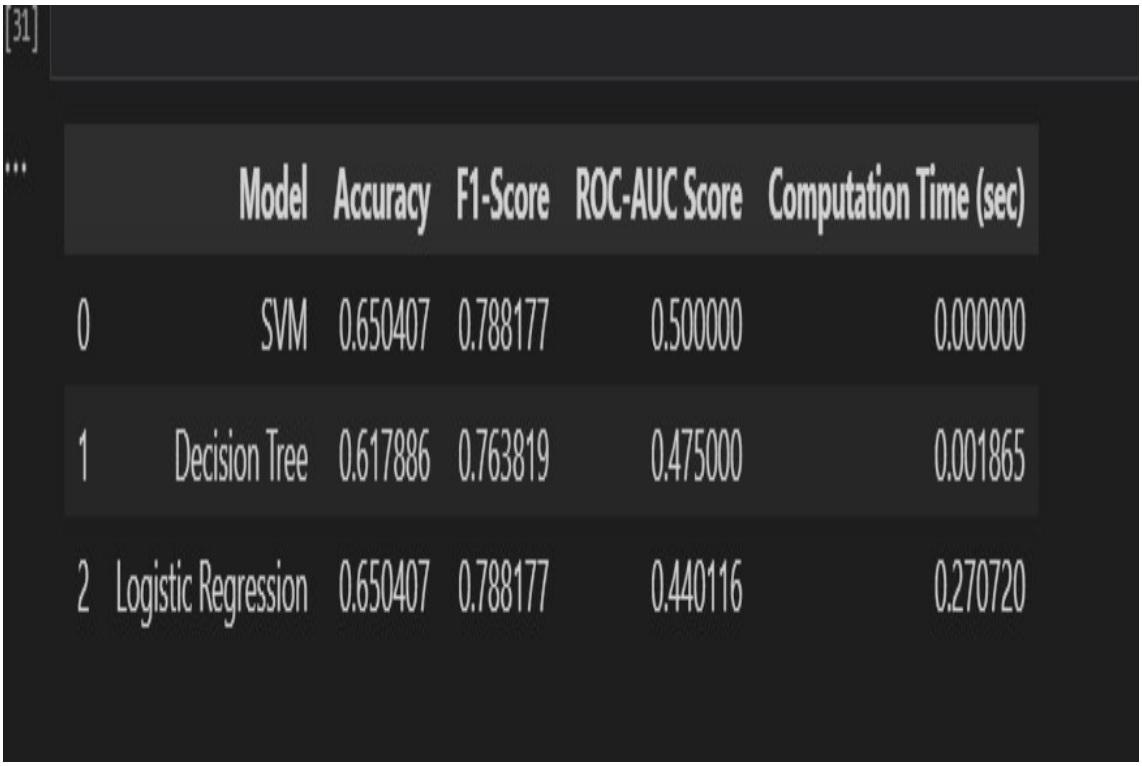


Figure 4.24: Performance metrics based on All Features



	Model	Accuracy	F1-Score	ROC-AUC Score	Computation Time (sec)
0	SVM	0.650407	0.788177	0.500000	0.000000
1	Decision Tree	0.617886	0.763819	0.475000	0.001865
2	Logistic Regression	0.650407	0.788177	0.440116	0.270720

Figure 4.25: Performance metrics based on Selected Features

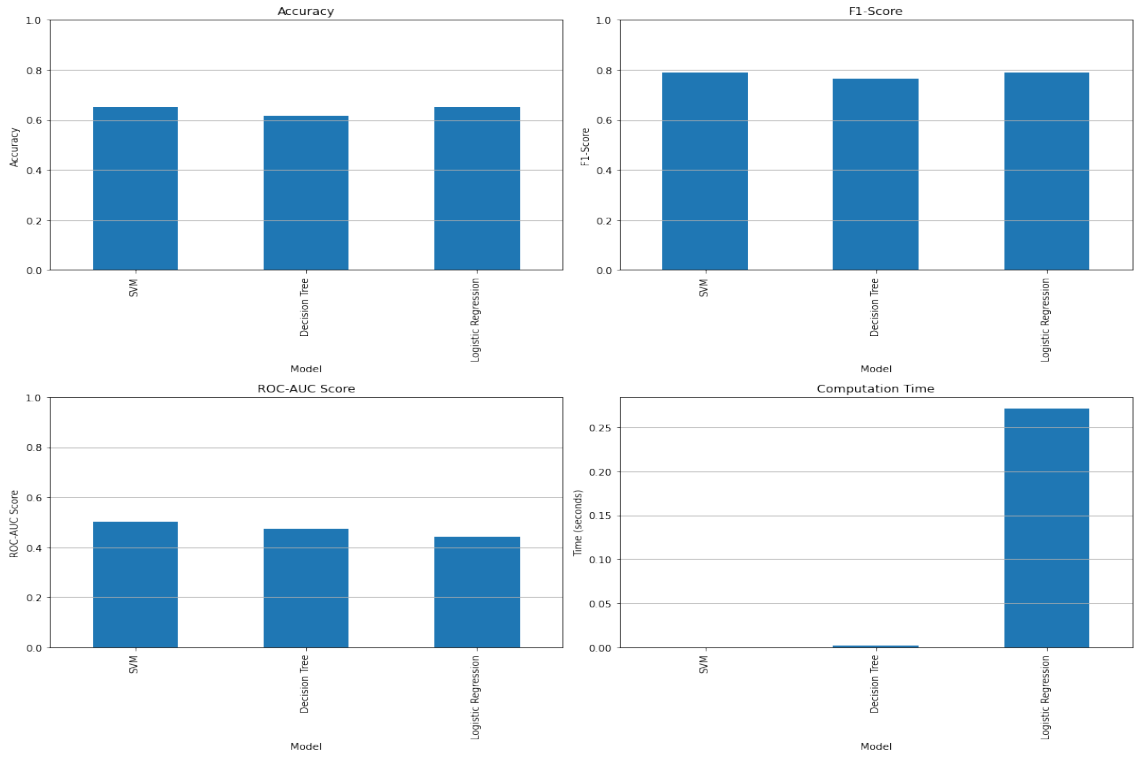


Figure 4.26: Performance metrics based on Selected Features

selecting features may provide more advantages, especially for the Support Vector Machine (SVM). In terms of performance measurements only, the all-features strategy appears to be superior.

4.2.5 Case 5: Social Media Ads

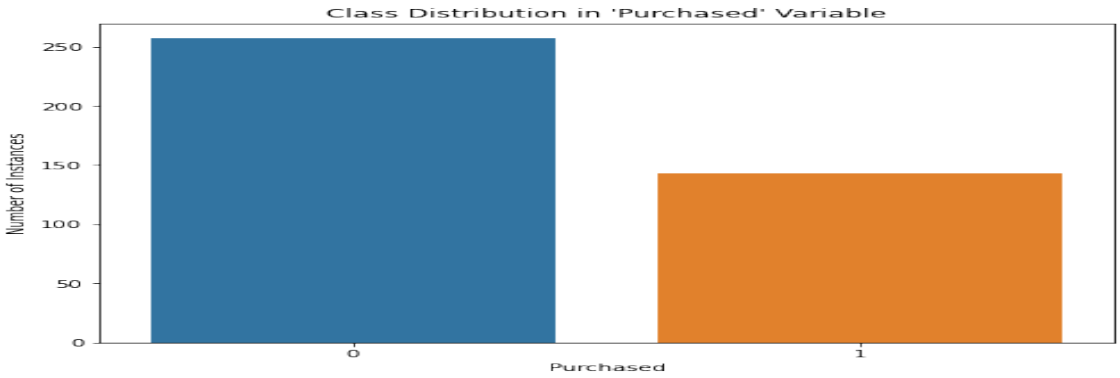


Figure 4.27: Class Distribution Diagram

Class Distribution The target variables may be classified into two unique categories: "Yes" indicating purchased, which is later encoded as 0, and "No" indicating not purchased item, encoded as 1. This observation underscores the fact that the topic under consideration may be classified into two distinct categories. It is worth noting that the dataset exhibits a greater frequency of "Yes" approvals in comparison to disapprovals. The dataset has a predominant prevalence of "Yes" responses for the goal variable.

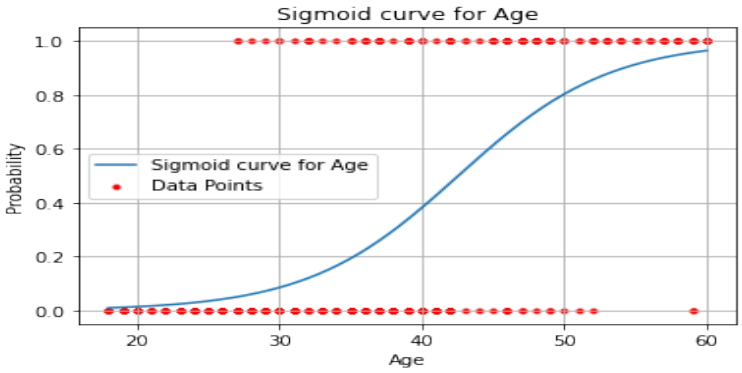


Figure 4.28: Sigmoid Diagram of Selected Features

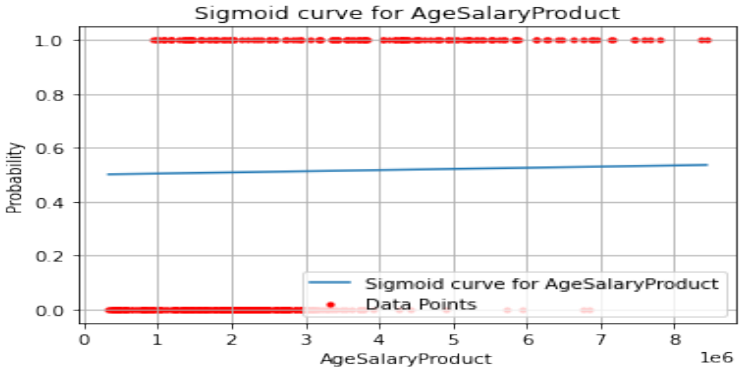


Figure 4.29: Sigmoid Diagram of Selected Features

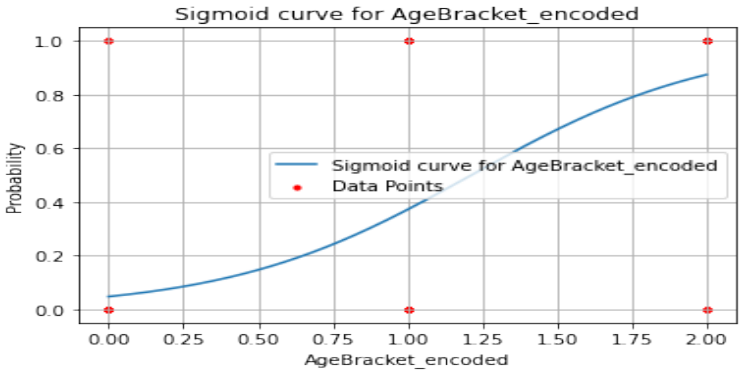


Figure 4.30: Sigmoid Diagram of Selected Features

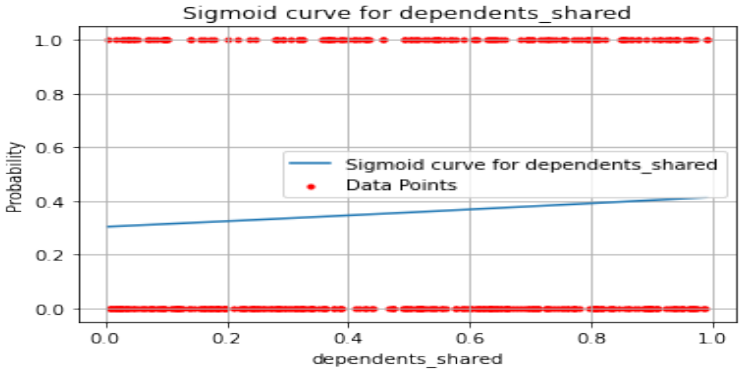


Figure 4.31: Sigmoid Diagram of Selected Features

Results Summary: In the case of Logistic Regression, the model trained on the selected features is more accurate, making it a superior option if accuracy is the main

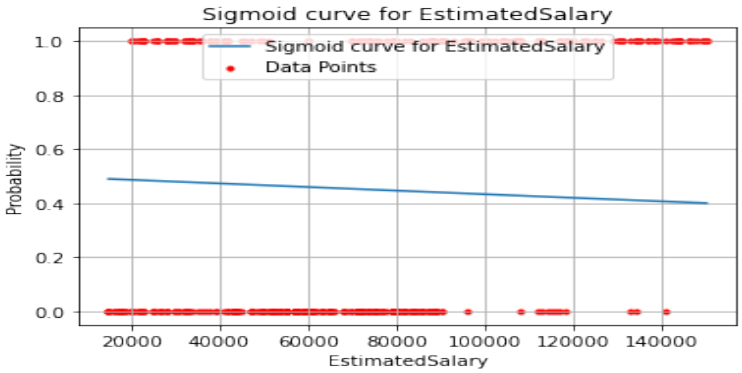


Figure 4.32: Sigmoid Diagram of Selected Features

[11]

	Test Accuracy	Test Recall	Test ROC AUC
Logistic Regression	0.9125	0.928571	0.984203
SVM	0.9375	0.928571	0.975275
Decision Tree	0.9000	0.928571	0.907967

Figure 4.33: Performance metrics based on All features

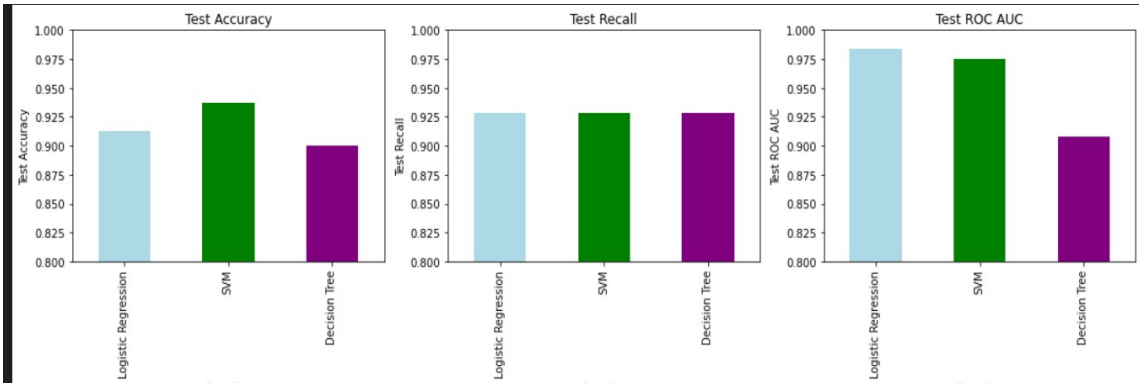


Figure 4.34: Performance metrics based on All features

	Model	Accuracy	F1 Score	ROC AUC Score	Time Minutes
0	Logistic Regression	0.9500	0.931034	0.978022	0.54
1	SVM	0.9400	0.923077	0.979408	1.20
2	Decision Tree	0.9125	0.885200	0.924500	0.28

Figure 4.35: Performance metrics based on Selected Features

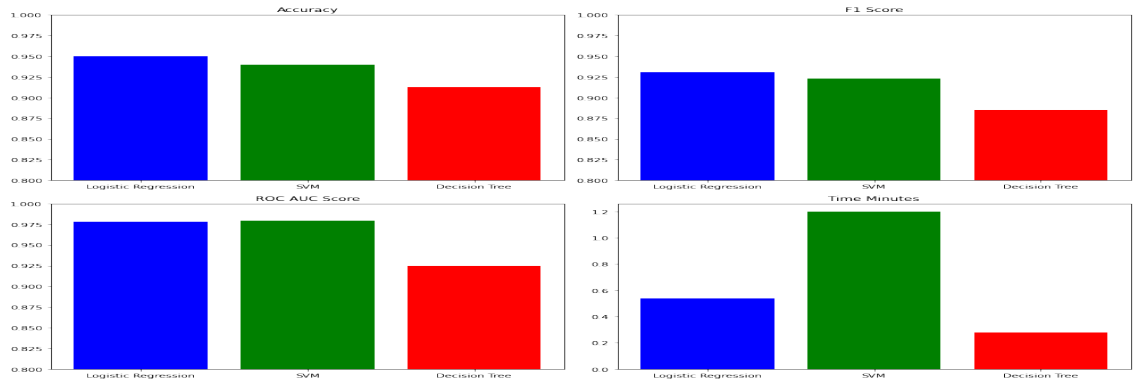


Figure 4.36: Performance metrics based on Selected Features

factor to be taken into account. With just little variations in performance measures, both models for SVM are extremely comparable. Recall and ROC AUC favour the Decision Tree model when it is trained on all features. Overall, the models developed using selected features are competitive and occasionally outperform their counterparts, although the gap isn't large. The minor performance improvement with selected features for Logistic Regression may be advantageous if computing time and resources are constrained. But combining all characteristics could be more beneficial if your main goal is to maximise your ROC AUC or recall, particularly for the Decision Tree.

4.2.6 Case 6:Water Quality

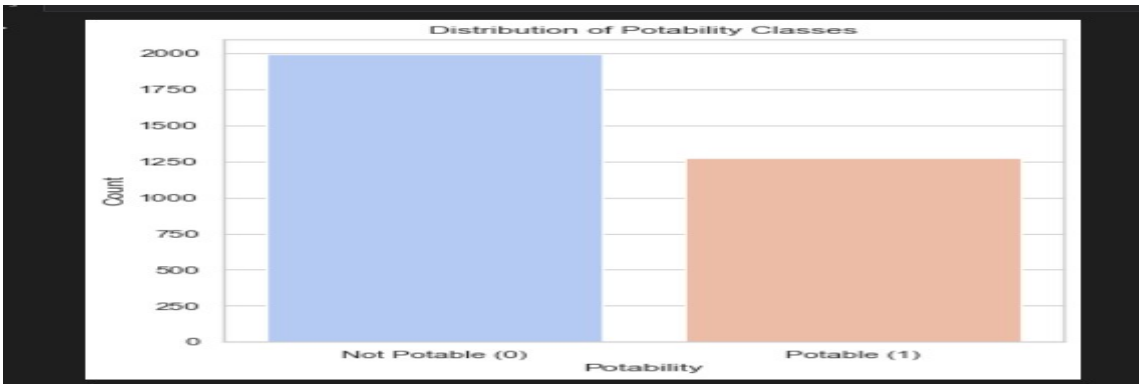


Figure 4.37: Class Distribution Diagram

Class Distribution A total of 3,276 samples make up the dataset, 1,998 of which are classified as "not potable" and 1,278 as "potable." This demonstrates an unequal distribution of the classes, with "not potable" samples predominating over "potable" ones. When modelling, appropriate methods should be used to correct this imbalance.

	Model	Accuracy	Precision	Recall	F1 Score	ROC AUC	Cross-Validation Mean	Computational Time (s)
0	Logistic Regression	0.628049	0.000000	0.000000	0.000000	0.519925	0.609890	0.172407
1	SVM	0.628049	0.000000	0.000000	0.000000	0.486690	0.609890	7.871785
2	Random Forest	0.673780	0.597403	0.377049	0.462312	0.687162	0.640728	5.289919
3	Naive Bayes	0.628049	0.500000	0.213115	0.298851	0.609422	0.611417	0.021727

Figure 4.38: Performance metrics based on All Features

	Model	Accuracy	Precision	Recall	F1 Score	ROC AUC	Cross-Validation Mean	Computational Time (s)
0	Logistic Regression	0.628049	0.000000	0.000000	0.000000	0.524371	0.609890	0.083184
1	SVM	0.628049	0.500000	0.004098	0.008130	0.583330	0.608363	7.979287
2	Random Forest	0.571646	0.407960	0.336066	0.368539	0.535856	0.554944	2.878756
3	Naive Bayes	0.641768	0.561644	0.168033	0.258675	0.579659	0.615379	0.022724

Figure 4.39: Performance metrics based on Selected Features

Result Summary With regard to feature selection, Logistic Regression and Naive Bayes both have mixed outcomes, with the former showing improvement in ROC AUC and the latter in accuracy and precision. The accuracy and ROC AUC gains from feature selection for SVM are greatest, while recall and F1 score suffer. Surprisingly, Random Forest outperforms in every statistic, demonstrating its durability. Importantly, feature selection always speeds up calculation. Overall, Random Forest outperforms competing models, and computations may be sped up by feature selection.

4.2.7 Case 7:Bank Authentication Dataset

Class Distribution The dataset has two classes that correspond to different types of banknotes: Class 1 (authentic) contains 610 examples, whereas Class 0 (inauthentic) has 738 instances. This suggests a fairly even distribution of classes, with fake banknotes being somewhat more common than real ones. Given this distribution, proper modelling methods should be used.

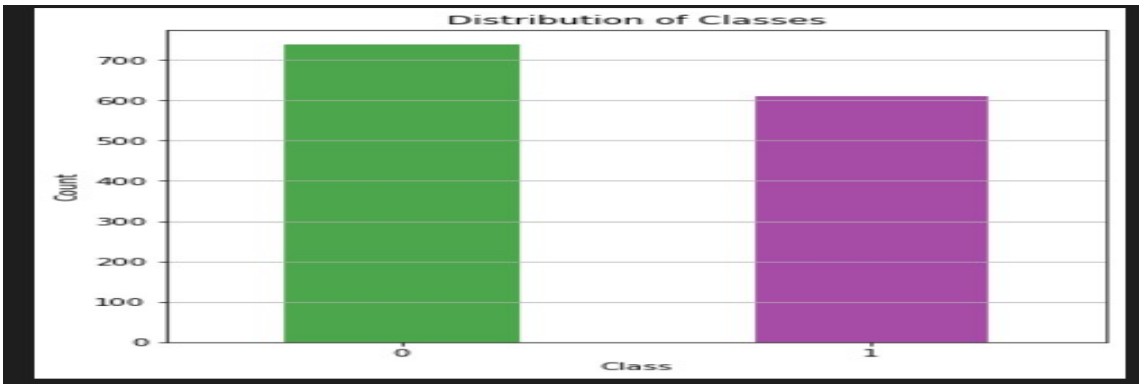


Figure 4.40: Class Distribution Diagram

	Model	CV Mean Accuracy	Computational Time (s)	Test Accuracy	Precision	Recall	F1-Score	ROC-AUC
0	Logistic Regression	0.992567	0.125999	0.992593	0.984000	1.0	0.991935	1.000000
1	SVM	0.998139	0.259264	0.992593	0.984000	1.0	0.991935	1.000000
2	Decision Tree	0.982364	0.061029	0.981481	0.960938	1.0	0.980080	0.982993

Figure 4.41: Performance metrics based on All Features

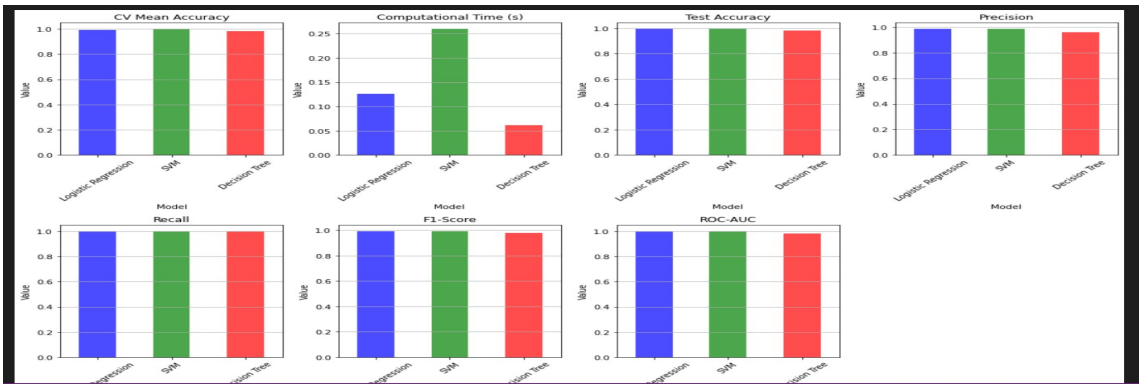


Figure 4.42: Performance metrics based on All Features

Result Summary Logistic Regression: Results from both feature sets were essentially the same. However, after employing a few features, the computing time was cut in half roughly.

	Training Time	CV Accuracy Mean	CV Accuracy Std	Test Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression	0.067849	0.993493	0.007264	0.992593	0.98400	1.0	0.991935	1.000000
SVM	0.237050	0.993493	0.005968	0.988889	0.97619	1.0	0.987952	1.000000
Decision Tree	0.040324	0.985151	0.010339	0.992593	0.98400	1.0	0.991935	0.993197

Figure 4.43: Performance metrics based on selected Features

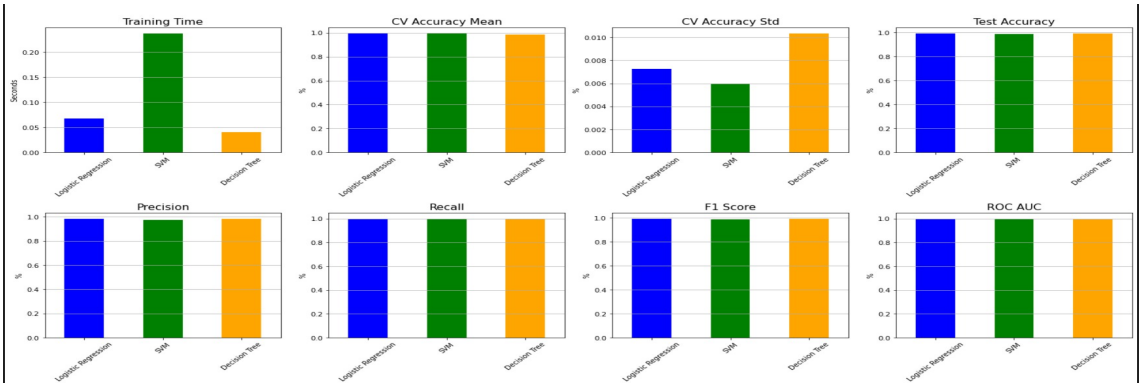


Figure 4.44: Performance metrics based on selected Features

SVM: While the precision reduced marginally while employing particular characteristics, the CV mean accuracy stayed constant. Even with less characteristics, the computational and training time did not vary considerably.

Decision Tree: The performance metrics of the model were same for both the ases . However, it became marginally more effective since the computational/training time for the feature selection was more feasible.

Overall, all three models achieved almost flawless metrics with both feature sets and performed quite well. Due to the shorter processing time without any discernible performance loss, the decision between all features and chosen features may favour selected features. The precise choice would, however, be based on the task's particular restrictions and objectives.

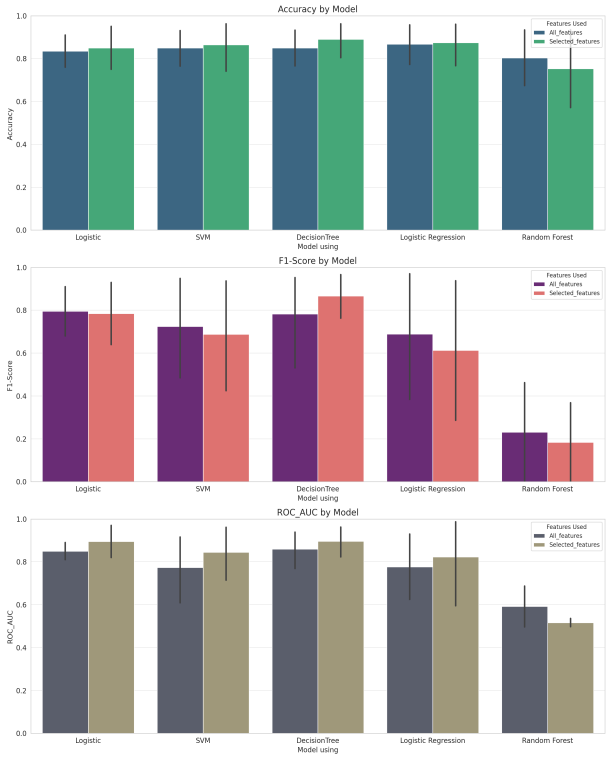


Figure 4.45: Aggregated Outcomes According to Performance Indicators

4.2.8 Aggregate Results

Analysis of the overall performance of three models (Logistic Regression, Support Vector Machine, and Decision Tree) on datasets that used two alternative feature sets: all features and those chosen using a hybrid approach. During the accuracy evaluation, the performance of the Logistic model was nearly equivalent for both the hybrid based selected feature and all features , with a little preference towards the former. The accuracy of the Support Vector Machine (SVM) remained consistent for both cases, however the Decision Tree algorithm demonstrated slightly improved results when using the selected features. Upon examination of the F1-Score metric, it was seen that the Logistic model shown a preference for the Hybrid approach. In contrast, the SVM model demonstrated similar scores for both approaches, while the DecisionTree model some what inclined towards the selected features. In terms of ROC AUC, the Logistic and Decision Tree models exhibited a predilection for the hybrid approach whereas the SVM model maintained a state of equilibrium. In conclusion, the performance of the all features approach is giving slight edge with minimal difference in terms of Accuracy. However, when considering the F1 Score and ROC AUC metric, the Hybrid approach demonstrates notable strength, particularly when utilizing Logistic and Decision tree

models. The selection of the most suitable collection of features is contingent upon the specific measure being sought and the choice of model.

4.3 Wrapper Methods

Wrapper methods are a class of feature selection strategies that depend on the effectiveness of a particular algorithm to assess the value of features. Recursive Feature Elimination (RFE), Sequential Forward Selection (SFS), and Sequential Backward Selection (SBS) are a few of the well-liked wrapper approaches. I used these three submethods in my investigation on 10 different datasets. I applied each technique in an effort to determine the best collection of characteristics for the selected machine-learning model. I contrasted the outcomes of the selected characteristics against the model's performance utilising all features to validate the effectiveness of each technique. I was able to identify the relative benefits and drawbacks of RFE, SFS, and SBS in the context of the various datasets according to this thorough evaluation.

DataSets	RFE Classifier Using	SFS Classifier Using	SBS Classifier Using	Hyperparameter Tuning	Notes
Titanic Dataset	logistic regression	SVM	Random Forest	Default parameters	Done
Heart Dataset	logistic regression	SVM	Logistic Regression	Default parameters	Done
Breast Cancer	Random forest	Naïve Bayes	SVM	Default parameters	Done
Irish Dataset	logistic regression	SVM	Random Forest	yes	Done
Diabetes Dataset	Random forest	Naïve Bayes	SVM issue hence Random Forest	Default parameters	Done
Glass identification	Random Forest	logistic regression	SVM	Default parameters	Done
IONOSPHERE	Decision tree	SVM	KNN	Default parameters	Done
Wine Dataset	Logistic	Random Forest	Decision Tree	Default parameters	Done
Water quality	Random Forest	Logtistic Regression	SVM	yes	Done
Brain Stroke	Random Forest	logistic regression	Logistic Regression	Default parameters	Done
Adult Census Income	Decision Tree	SVM	Logistic regression	Default parameters	Done

Figure 4.46: Wrapper Classifiers List

4.3.1 Case1:Glass Dataset

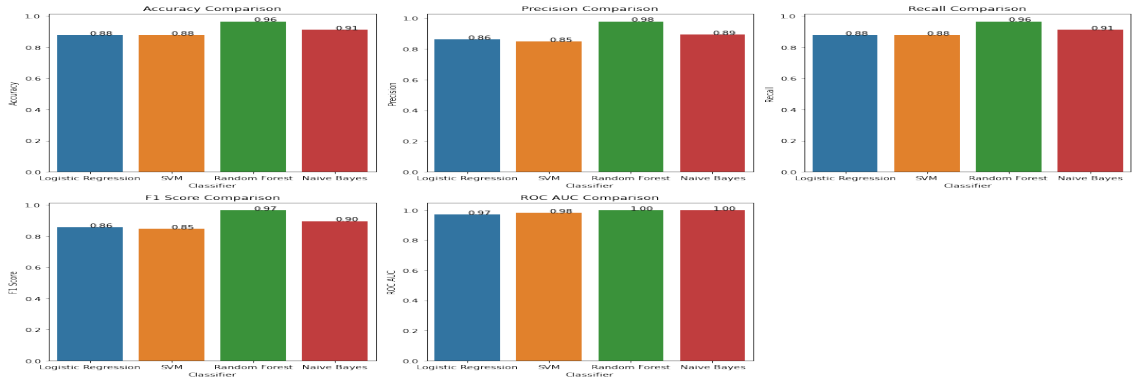


Figure 4.47: Performance metrics After Feature Selection [RFE]

	Classifier	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	Logistic Regression	0.877143	0.862630	0.877143	0.856384	0.971533
1	SVM	0.877143	0.849401	0.877143	0.847532	0.982374
2	Random Forest	0.964874	0.978517	0.964874	0.967023	0.998515
3	Naive Bayes	0.912101	0.893484	0.912101	0.896726	0.997348

Figure 4.48: Performance metrics After Feature Selection [RFE]

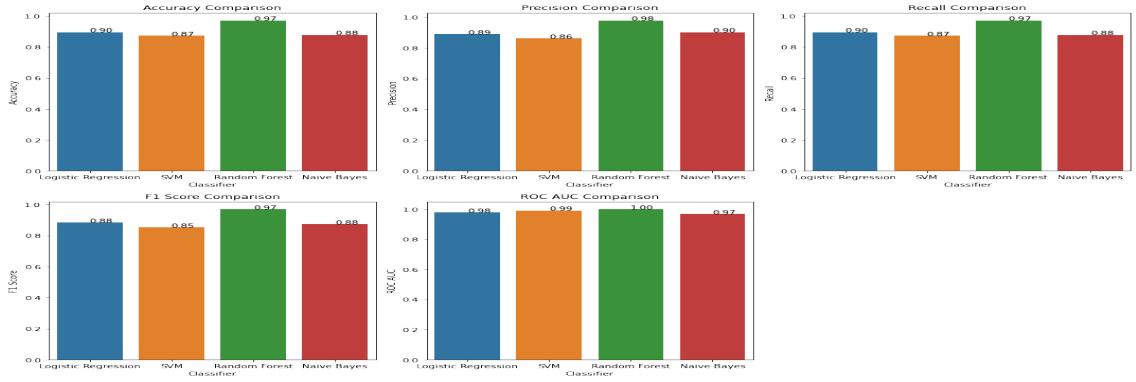


Figure 4.49: Performance metrics After Feature Selection [SFS]

Result Summary

With the greatest Accuracy, Precision, F1 Score, and ROC AUC, RFE has the best performance. Particularly in terms of Precision and ROC AUC, SFS follows closely.The performances of All Features and SBS are extremely comparable, although they fall just short of RFE and SFS. The Recursive Feature Elimination (RFE) approach outperforms other assessment measures, such as Accuracy, Precision, F1 Score, and ROC

	Classifier	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	Logistic Regression	0.895294	0.888739	0.895294	0.883899	0.976590
1	SVM	0.871597	0.861040	0.871597	0.852539	0.989993
2	Random Forest	0.970924	0.976893	0.970924	0.970018	0.998712
3	Naive Bayes	0.877647	0.900687	0.877647	0.875057	0.965831

Figure 4.50: Performance metrics After Feature Selection [SFS]

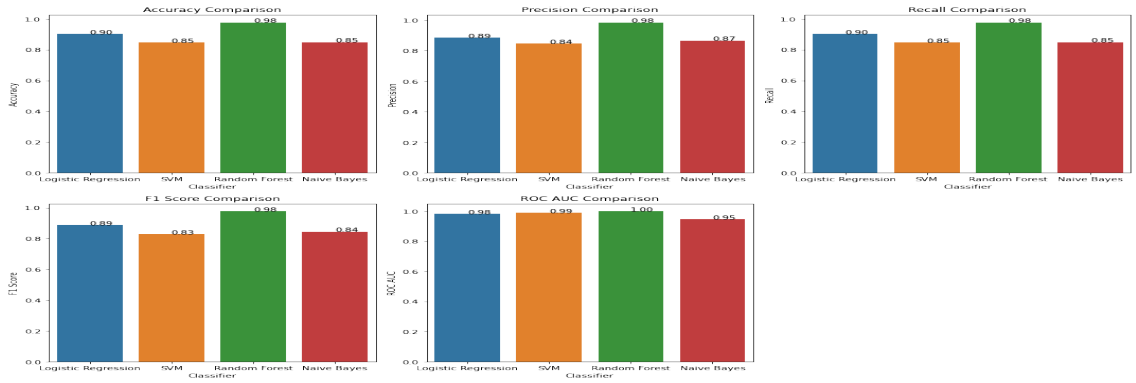


Figure 4.51: Performance metrics After Feature Selection [SBS]

	Classifier	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	Logistic Regression	0.901008	0.885404	0.901008	0.887620	0.980561
1	SVM	0.847899	0.844963	0.847899	0.828052	0.989511
2	Random Forest	0.976807	0.983503	0.976807	0.977247	0.999356
3	Naive Bayes	0.848403	0.865284	0.848403	0.844522	0.945340

Figure 4.52: Performance metrics After Feature Selection [SBS]

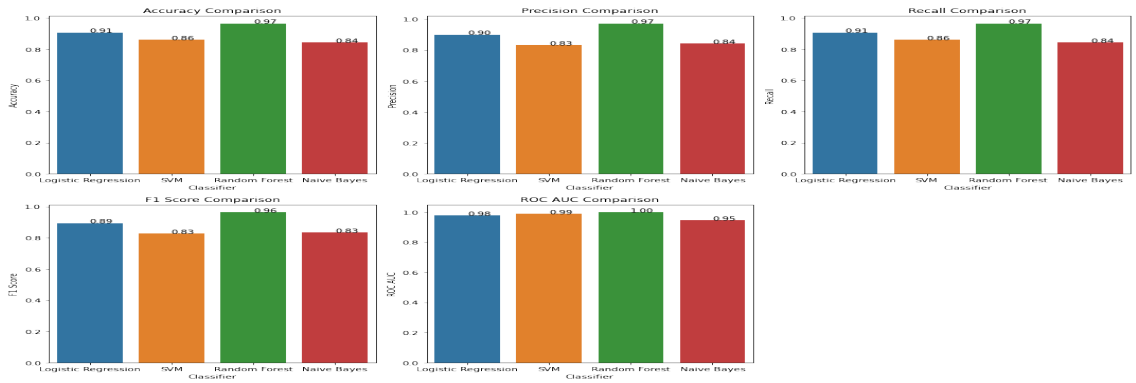


Figure 4.53: Performance metrics based on All Features

	Classifier	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	Logistic Regression	0.906723	0.899686	0.906723	0.894273	0.976368
1	SVM	0.859664	0.831447	0.859664	0.827458	0.988484
2	Random Forest	0.965042	0.972247	0.965042	0.964095	0.998859
3	Naive Bayes	0.842521	0.843575	0.842521	0.833963	0.946337

Figure 4.54: Performance metrics based on All Features

AUC, according to the empirical findings. This indicates that the most pertinent feature subset for the given classification tasks is effectively captured by RFE's iterative method of ranking and removing the least significant features based on the selected estimate.The performance differences between RFE and the other approaches, such as Sequential Backward Selection (SBS) and Sequential Forward Selection (SFS), are, despite the fact that RFE tops the charts in our assessments, rather slight.

4.3.2 Case 2:Heart Dataset

	Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression	0.819512	0.784483	0.883495	0.83105	0.877879
SVM	0.853659	0.823009	0.902913	0.861111	0.919855
Random Forest	0.97561	0.962264	0.990291	0.976077	0.996954
Naive Bayes	0.790244	0.754237	0.864078	0.80543	0.87331

Figure 4.55: Performance metrics based on RFE

Result Summary

SFS (Sequential Forward Selection) appears to have an advantage, especially when taking into account SVM's better performance and Random Forest's persistent high scores.

Additionally, All Features has considerable promise, particularly in light of the Random Forest classifier's flawless Precision and ROC AUC scores.However, it's worth noting that while Random Forest's performance is stellar across all feature selection

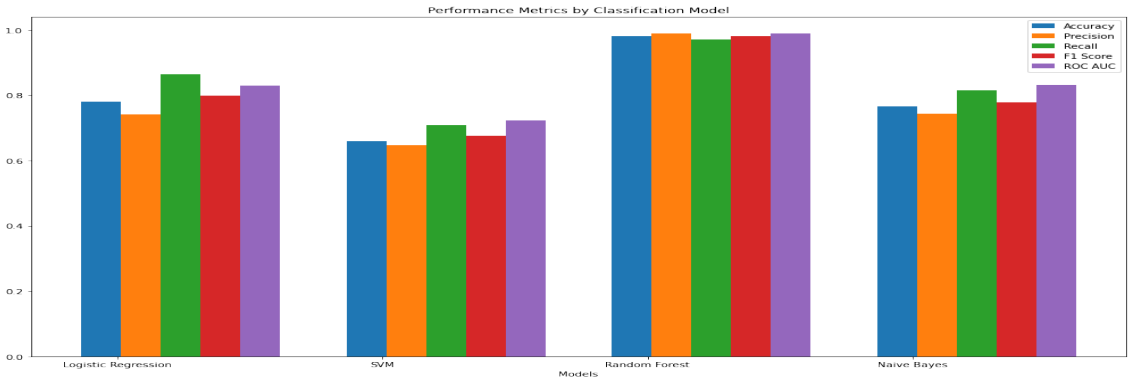


Figure 4.56: Performance metrics based on RFE

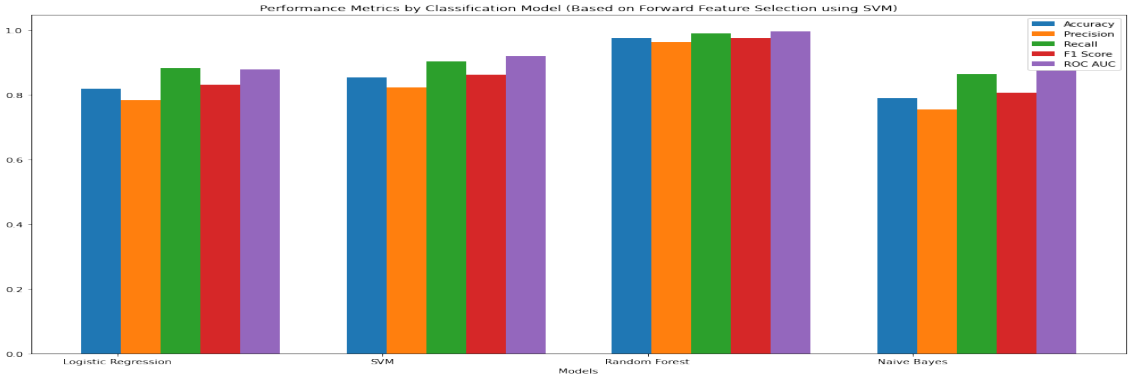


Figure 4.57: Performance metrics based on Forward Selection

	Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression	0.819512	0.784483	0.883495	0.83105	0.877879
SVM	0.853659	0.823009	0.902913	0.861111	0.919855
Random Forest	0.97561	0.962264	0.990291	0.976077	0.996954
Naive Bayes	0.790244	0.754237	0.864078	0.80543	0.87331

Figure 4.58: Performance metrics based on Forward Selection

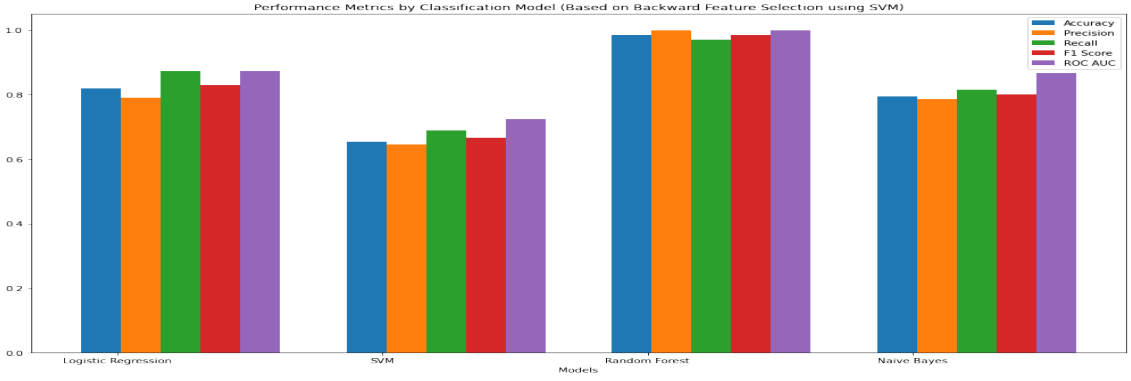


Figure 4.59: Performance metrics based on Backward Selection

	Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression	0.819512	0.789474	0.873786	0.829493	0.872549
SVM	0.653659	0.645455	0.68932	0.666667	0.724824
Random Forest	0.985366	1.0	0.970874	0.985222	1.0
Naive Bayes	0.795122	0.785047	0.815534	0.8	0.866077

Figure 4.60: Performance metrics based on Backward Features

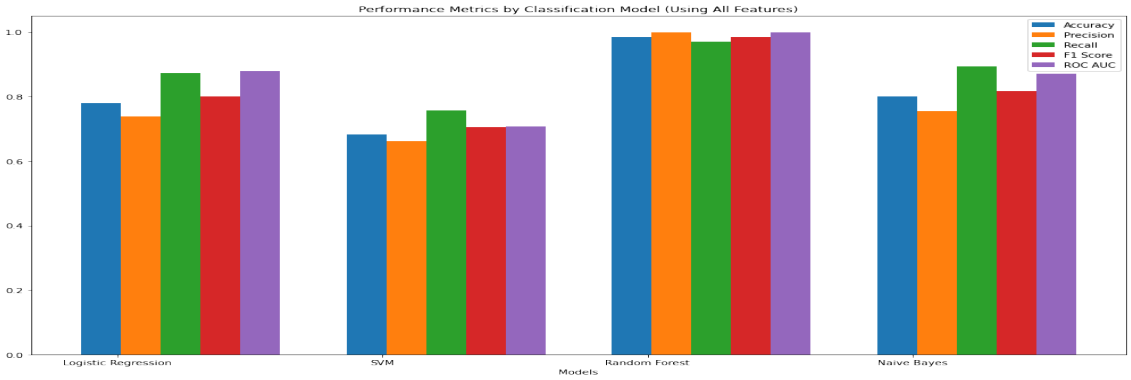


Figure 4.61: Performance metrics based on All Features

	Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression	0.780488	0.737705	0.873786	0.8	0.879117
SVM	0.682927	0.661017	0.757282	0.705882	0.70712
Random Forest	0.985366	1.0	0.970874	0.985222	1.0
Naive Bayes	0.8	0.754098	0.893204	0.817778	0.87055

Figure 4.62: Performance metrics based on All Features

methods, the other classifiers show more variability. The optimal feature selection method may depend on the classifier in use and the specific goals of the analysis.

4.3.3 Case 3:Titanic Dataset

	Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
0	Logistic Regression	0.821229	0.800000	0.756757	0.777778	0.881081
1	SVM	0.608939	0.833333	0.067568	0.125000	0.872844
2	Random Forest	0.793296	0.793651	0.675676	0.729927	0.856628
3	Naive Bayes	0.776536	0.688889	0.837838	0.756098	0.850064

Figure 4.63: Performance metrics based on RFE

	Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
0	Logistic Regression	0.770950	0.708861	0.756757	0.732026	0.857014
1	SVM	0.798883	0.796875	0.689189	0.739130	0.822136
2	Random Forest	0.787709	0.810345	0.635135	0.712121	0.850901
3	Naive Bayes	0.770950	0.708861	0.756757	0.732026	0.865894

Figure 4.64: Performance metrics based on SFS

	Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
0	Logistic Regression	0.804469	0.774648	0.743243	0.758621	0.872458
1	SVM	0.603352	0.800000	0.054054	0.101266	0.761905
2	Random Forest	0.815642	0.797101	0.743243	0.769231	0.896976
3	Naive Bayes	0.759777	0.706667	0.716216	0.711409	0.844144

Figure 4.65: Performance metrics based on SBS

	Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
0	Logistic Regression	0.815642	0.797101	0.743243	0.769231	0.886358
1	SVM	0.664804	0.791667	0.256757	0.387755	0.730888
2	Random Forest	0.826816	0.794521	0.783784	0.789116	0.912934
3	Naive Bayes	0.798883	0.743590	0.783784	0.763158	0.849292

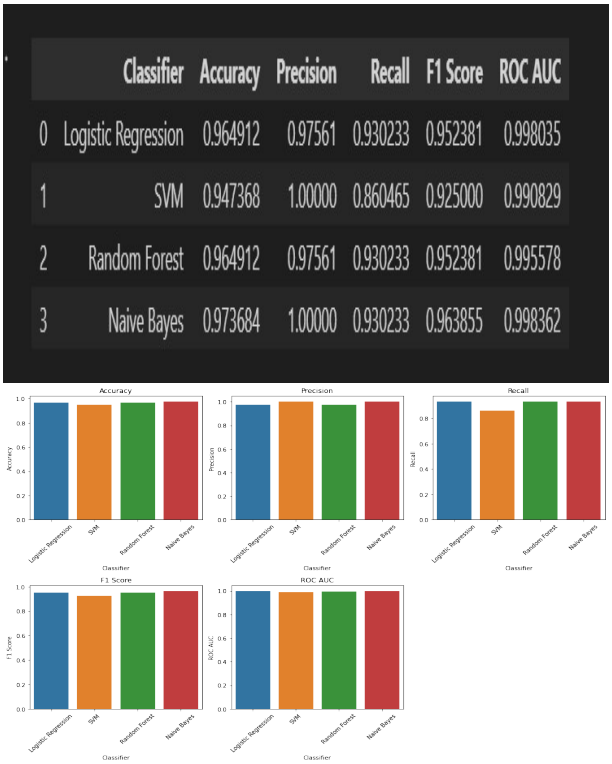
Figure 4.66: Performance metrics based on All Features

Result Summary

Overall, RFE and SBS perform the best, followed by SFS and All features. . However, for the Logistic Regression, Random Forest, and Naive Bayes models in this situation, RFE and SBS perform the best. SFS, however, performs optimally for the SVM model. Here are some additional things to consider when choosing a feature selection method: Computational complexity: Some feature selection methods, such as RFE and SFS, can be computationally expensive for large datasets.

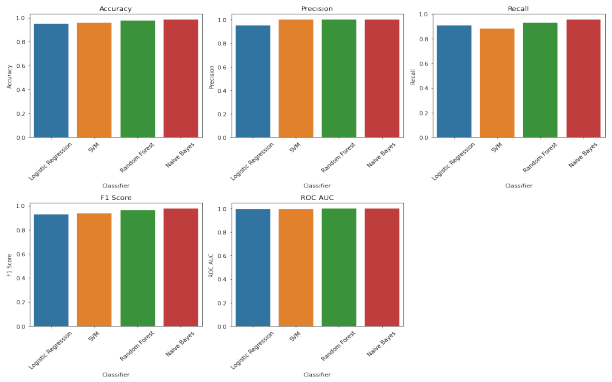
Accuracy: When it comes to accuracy, RFE and SBS typically exceed SFS and All features. Accuracy is a crucial factor to take into account when selecting a feature selection technique, but it is not the only one.

4.3.4 Case 4: Breast Dataset



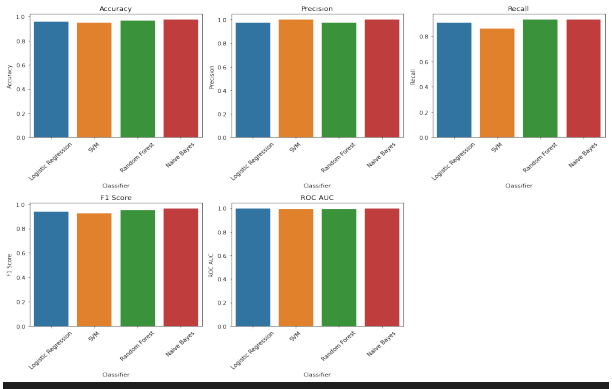
	Classifier	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	Logistic Regression	0.964912	0.97561	0.930233	0.952381	0.998035
1	SVM	0.947368	1.00000	0.860465	0.925000	0.990829
2	Random Forest	0.964912	0.97561	0.930233	0.952381	0.995578
3	Naive Bayes	0.973684	1.00000	0.930233	0.963855	0.998362

Figure 4.67: Performance metrics based on Recursive Feature selection



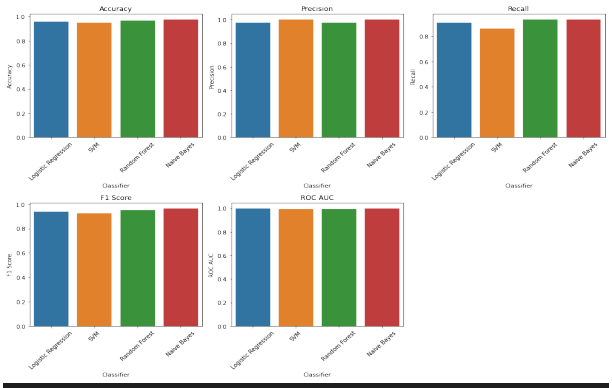
	Classifier	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	Logistic Regression	0.947368	0.95122	0.906977	0.928571	0.995414
1	SVM	0.956140	1.00000	0.883721	0.938272	0.996069
2	Random Forest	0.973684	1.00000	0.930233	0.963855	0.999345
3	Naive Bayes	0.982456	1.00000	0.953488	0.976190	0.998690

Figure 4.68: Performance metrics based on Forward Selection



	Classifier	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	Logistic Regression	0.956140	0.97500	0.906977	0.939759	0.998035
1	SVM	0.947368	1.00000	0.860465	0.925000	0.993449
2	Random Forest	0.964912	0.97561	0.930233	0.952381	0.995414
3	Naive Bayes	0.973684	1.00000	0.930233	0.963855	0.998035

Figure 4.69: Performance metrics based on Backward Selection



	Classifier	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	Logistic Regression	0.956140	0.97500	0.906977	0.939759	0.998362
1	SVM	0.947368	1.00000	0.860465	0.925000	0.993449
2	Random Forest	0.964912	0.97561	0.930233	0.952381	0.995251
3	Naive Bayes	0.973684	1.00000	0.930233	0.963855	0.998362

Figure 4.70: Performance metrics based on All Feature

Result Summary Recursive Feature Elimination (RFE): This technique appears to perform well across all classifiers, especially in ROC AUC, where all classifiers score extremely near to 1.

All Features: Once more, all classifiers function well when given a full feature set, especially in terms of ROC AUC values.

SFS (Sequential Forward Selection): This technique appears to perform exceptionally well, particularly when used with the Naive Bayes classifier, which has the best accuracy of all techniques or classifiers.

SBS (Sequential Backward Selection): The performance is similar to the 'All Features' instance, with exceptionally high ROC AUC values.

RFE, All Features, and SFS appear to perform well and consistently among classifiers from a broad perspective. SFS excels in accuracy, especially with the Naive Bayes classifier.

If accuracy is the primary priority given the data, SFS with Naive Bayes may be the best option. But for performance that holds up across multiple classifiers, RFE and All Features both deliver impressive outcomes.

4.3.5 Case 5: Adult Dataset

	Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	Logistic Regression	0.799017	0.786486	0.191447	0.307937	0.538370
1	SVM	0.798709	0.964602	0.143421	0.249714	0.597586
2	Random Forest	0.845882	0.688274	0.621711	0.653301	0.895188
3	Naive Bayes	0.795329	0.709821	0.209211	0.323171	0.824424

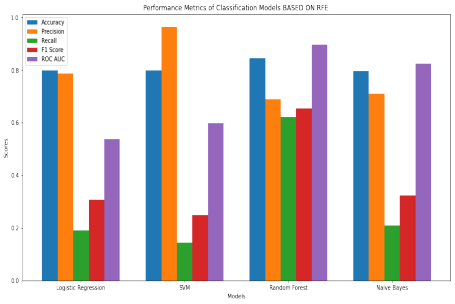


Figure 4.71: Performance metrics based on RFE

	Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	Logistic Regression	0.824985	0.751651	0.374342	0.499780	0.805413
1	SVM	0.805624	0.731397	0.265132	0.389184	0.811829
2	Random Forest	0.861862	0.775510	0.575000	0.660370	0.903529
3	Naive Bayes	0.798095	0.645892	0.300000	0.409704	0.811964

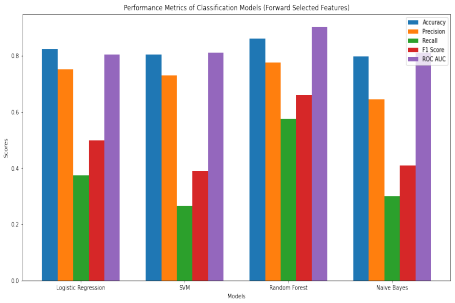


Figure 4.72: Performance metrics based on SFS

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output ...

	Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	Logistic Regression	0.812999	0.662032	0.407237	0.504277	0.809048
1	SVM	0.805624	0.731397	0.265132	0.389184	0.826036
2	Random Forest	0.845114	0.682596	0.629605	0.655031	0.889904
3	Naive Bayes	0.800092	0.649795	0.312500	0.422035	0.847243

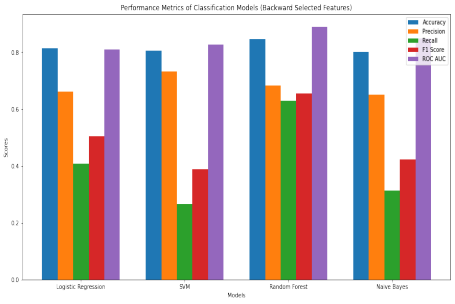
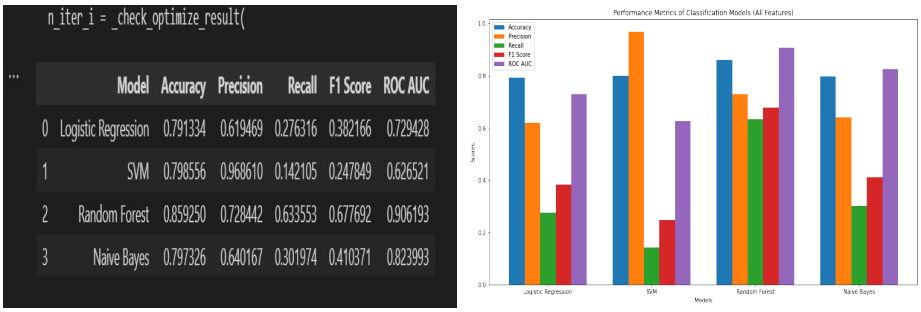


Figure 4.73: Performance metrics based on SBS



	Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	Logistic Regression	0.791334	0.619469	0.276316	0.382166	0.729428
1	SVM	0.798556	0.968610	0.142105	0.247849	0.626521
2	Random Forest	0.859250	0.728442	0.633553	0.677692	0.906193
3	Naive Bayes	0.797326	0.640167	0.301974	0.410371	0.823993

Figure 4.74: Performance metrics based on All Features

Result Summary Based on a comprehensive evaluation of several metrics, it can be concluded that the SFS (Sequential Forward Selection) technique has superior performance in terms of accuracy. Nevertheless, Sequential Backward Selection (SBS) demonstrates optimal equilibrium between accuracy and recall, as evidenced by its F1 score and the maximum Receiver Operating Characteristic Area Under the Curve (ROC AUC). If one places importance on achieving a balance between precision and recall, as shown by the F1 score, as well as the capacity to effectively differentiate across classes, then the Sequential Backward Selection (SBS) method would be the most suitable option. If prioritising accuracy is a more relevant criterion for your specific use-case, then the usage of SFS (Sequential Forward Selection) is recommended

4.3.6 Case 6:Diabetes Dataset

	Classifier	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	Logistic Regression	0.758750	0.721667	0.580000	0.624899	0.719444
1	SVM	0.765833	0.731667	0.596667	0.641061	0.727778
2	Random Forest	0.741667	0.758810	0.520000	0.579655	0.695000
3	Naive Bayes	0.780000	0.732143	0.660000	0.682222	0.754444

	Classifier	Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression		0.819167	0.838929	0.656667	0.709210	0.783333
SVM		0.772500	0.751667	0.600000	0.652273	0.735000
Random Forest		0.723333	0.660000	0.550000	0.590571	0.685000
Naive Bayes		0.812500	0.778571	0.696667	0.717599	0.788333

	Classifier	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	Logistic Regression	0.766250	0.726190	0.643333	0.651896	0.741111
1	SVM	0.733750	0.688333	0.496667	0.558232	0.682778
2	Random Forest	0.741667	0.672143	0.573333	0.600831	0.706111
3	Naive Bayes	0.805417	0.735476	0.710000	0.717498	0.784444

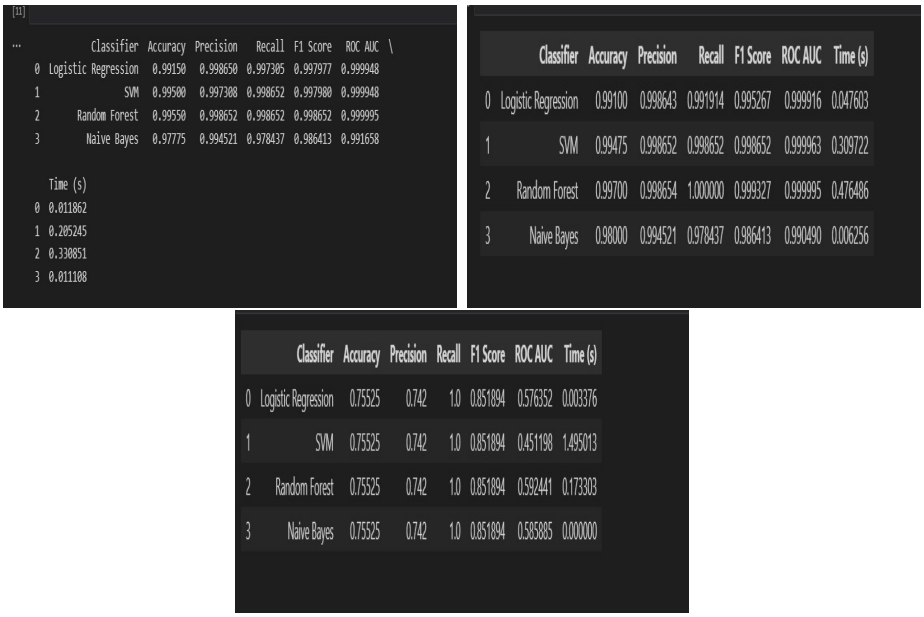
	Classifier	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	Logistic Regression	0.766250	0.726190	0.643333	0.651896	0.741111
1	SVM	0.733750	0.688333	0.496667	0.558232	0.682778
2	Random Forest	0.741667	0.672143	0.573333	0.600831	0.706111
3	Naive Bayes	0.805417	0.735476	0.710000	0.717498	0.784444

Figure 4.75: Performance metrics based on [RFE,SFE,SBS,All]

Figure 4.76: Performance metrics based on All Features

Result Summary Considering the balance across metrics, the SFS (Sequential Forward Selection) method seems to perform the best overall. It leads in most of the metrics and provides a good balance between precision and recall, as indicated by its F1 score.The SFS (Sequential Forward Selection) approach appears to perform the best overall when taking into account the balance across measures. It performs well in the majority of parameters and offers a decent mix between recall and accuracy, as seen by its F1 score.

4.3.7 Case 7:Wine Dataset



	Classifier	Accuracy	Precision	Recall	F1 Score	ROC AUC	Time (s)
0	Logistic Regression	0.99150	0.998650	0.997305	0.997977	0.999948	0.011862
1	SVM	0.99500	0.997308	0.998652	0.997980	0.999948	0.205245
2	Random Forest	0.99550	0.998652	0.998652	0.998652	0.999995	0.330851
3	Naive Bayes	0.97775	0.994521	0.978437	0.986413	0.991658	0.011108

	Classifier	Accuracy	Precision	Recall	F1 Score	ROC AUC	Time (s)
0	Logistic Regression	0.99100	0.998643	0.991914	0.995267	0.999916	0.047603
1	SVM	0.99475	0.998652	0.998652	0.998652	0.999963	0.309722
2	Random Forest	0.99700	0.998654	1.000000	0.999327	0.999995	0.476486
3	Naive Bayes	0.98000	0.994521	0.978437	0.986413	0.990490	0.006256

	Classifier	Accuracy	Precision	Recall	F1 Score	ROC AUC	Time (s)
0	Logistic Regression	0.75525	0.742	1.0	0.851894	0.576352	0.003376
1	SVM	0.75525	0.742	1.0	0.851894	0.451198	1.495013
2	Random Forest	0.75525	0.742	1.0	0.851894	0.592441	0.173303
3	Naive Bayes	0.75525	0.742	1.0	0.851894	0.585885	0.000000

Figure 4.77: Performance metrics based on [RFE,SFE,SBS]

	Classifier	Accuracy	Precision	Recall	F1 Score	ROC AUC	Time (s)
0	Logistic Regression	0.99275	0.997308	0.998652	0.997980	0.999943	0.013542
1	SVM	0.99575	1.000000	0.998652	0.999326	0.999963	0.350745
2	Random Forest	0.99575	0.997312	1.000000	0.998654	0.999948	0.402747
3	Naive Bayes	0.98025	0.994513	0.977089	0.985724	0.992170	0.010872

Figure 4.78: Performance metrics based on [RFE,SFE,SBS]

Result Summary Accuracy: All Features has the highest average accuracy. Precision: SFS edges in the precision. Recall: All Features has the highest recall compare to other cases. F1 Score: All Features perform best on F1 score. ROC AUC: All Features perform best on ROC AUC. Time (s): RFE is the fastest on average, followed closely by All Features.

With the greatest scores in Accuracy, Recall, F1 Score, and ROC AUC, All Features appears to perform the best overall when all metrics are balanced. However, RFE is a little bit faster while still providing competitive performance metrics if computing time is a major factor.

4.3.8 Case 8:Weather Dataset

	Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
0	Logistic Regression	0.821229	0.800000	0.756757	0.777778	0.881081
1	SVM	0.608939	0.833333	0.067568	0.125000	0.872844
2	Random Forest	0.793296	0.793651	0.675676	0.729927	0.856628
3	Naive Bayes	0.776536	0.688889	0.837838	0.756098	0.850064

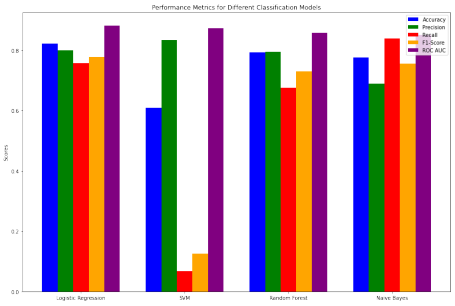


Figure 4.79: Performance metrics based on RFE

	Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
0	Logistic Regression	0.770950	0.708861	0.756757	0.732026	0.857014
1	SVM	0.798883	0.796875	0.689189	0.739130	0.822136
2	Random Forest	0.787709	0.810345	0.635135	0.712121	0.850901
3	Naive Bayes	0.770950	0.708861	0.756757	0.732026	0.865894

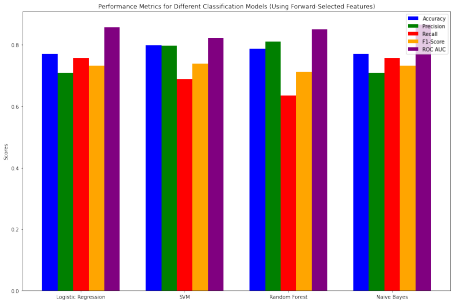


Figure 4.80: Performance metrics based on Forward Selection

	Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
0	Logistic Regression	0.804469	0.774648	0.743243	0.758621	0.872458
1	SVM	0.603352	0.800000	0.054054	0.101266	0.761905
2	Random Forest	0.815642	0.797101	0.743243	0.769231	0.896976
3	Naive Bayes	0.759777	0.706667	0.716216	0.711409	0.844144

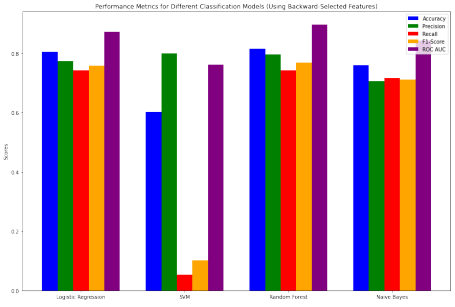


Figure 4.81: Performance metrics based on Backward Selection

	Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
0	Logistic Regression	0.815642	0.797101	0.743243	0.769231	0.886358
1	SVM	0.664804	0.791667	0.256757	0.387755	0.730888
2	Random Forest	0.826816	0.794521	0.783784	0.789116	0.912934
3	Naive Bayes	0.798883	0.743590	0.783784	0.763158	0.849292

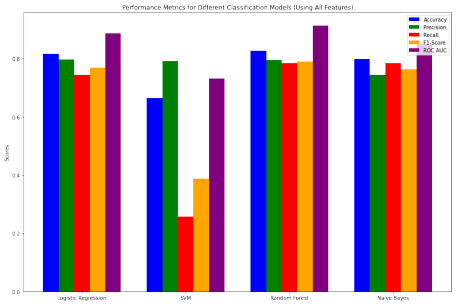


Figure 4.82: Performance metrics based on All Features

Result Summary Accuracy: The best accuracy is provided by SFS with Naive Bayes.SVM consistently provides great accuracy across all methodologies, however it is typically accompanied by relatively low recall.

Precision: SVM consistently offers high precision across all methods, but it's complemented by a very low recall in most cases.

Recall: The best recall is achieved using the Naive Bayes with RFE and All Features approaches.

F1 Score: Of all the approaches, Random Forest and Naive Bayes often provide the best F1 scores when balancing accuracy and recall. The best ROC AUC is found in the Logistic Regression using the RFE approach, demonstrating a potent capacity to differentiate between the positive and negative classes.

Based on the research done on the weather dataset, RFE approach may be the best option overall, particularly when paired with Random Forest or Logistic Regression.

4.3.9 Aggregate Results

When exploring other performance criteria, it is routinely seen that Random Forest with Recursive Feature Elimination (RFE) consistently emerges as a leading performer. The method exhibits a high level of precision and is only comparable to the performance of Support Vector Machines (SVM) when using All Features. When considering recall and F1-Score metrics, it can be observed that both Recursive Feature Elimination (RFE) and Sequential Forward Selection (SFS) techniques, when combined with the Random Forest algorithm, provide greater performance. Moreover, in the assessment of the area under the receiver operating characteristic (ROC) curve, the Random Forest algorithm with Recursive Feature Elimination (RFE) exhibits a clear superiority

over other combinations, thereby emphasising its effectiveness.The results of this study emphasise the significance of careful feature selection in improving the reliability and precision of models. Among many data circumstances, the wrapper technique shows great potential as an effective strategy.

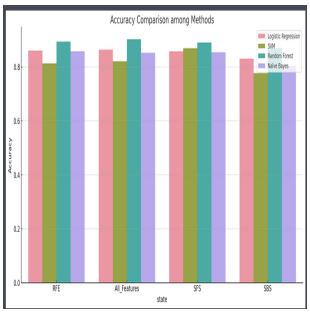


Figure 4.83:

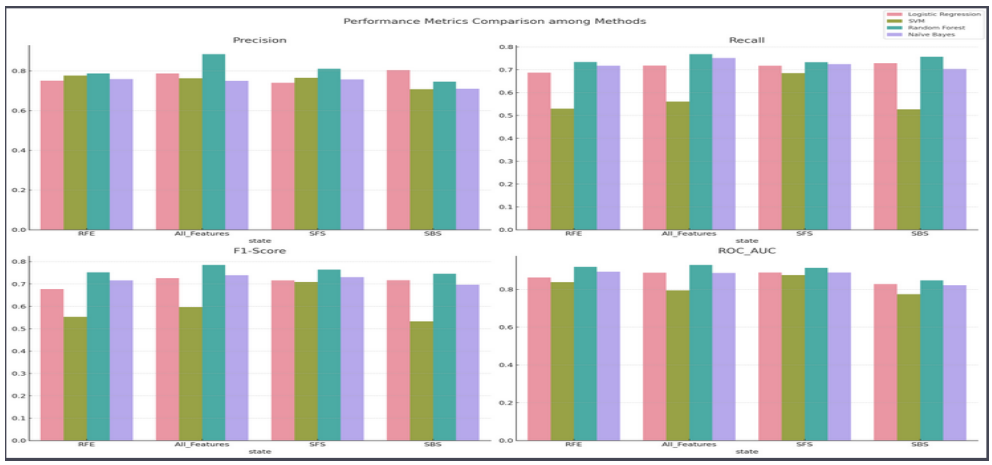


Figure 4.84: Aggregated Outcomes According to Performance Indicators

Chapter 5

Conclusion

Feature selection is a crucial phase in the machine learning process, as it is used to improve the performance of models, mitigate overfitting, and accelerate training. The dissertation focused on investigating the effectiveness of feature selection in enhancing the results of machine learning. This was achieved by employing a hybrid filter-wrapper strategy by applying statistical measures with a combination of RFE methods with different classifiers for getting best features with low computational power on a total of 10 datasets. Additionally, three wrapper approaches, including Sequential Forward Selection (SFS), Sequential Backward Selection (SBS), and RFE, were applied to another set of 10 datasets.

5.1 Evaluation and Discussion

Upon analysing the data, it becomes apparent that the utilisation of feature selection often leads in enhancements in model performance. The hybrid filter-wrapper strategy, which integrates the advantages of both filtering and wrapping techniques, has demonstrated its promise in selecting a subset of characteristics that positively impacting the performance metrics.Despite problems in some dataset like complex interaction overlooked and causing redundancy by choosing high corelated features. This strategy demonstrates a remarkable ability to effectively strike a balance between eliminating unimportant or redundant characteristics while preserving the most influential ones, hence highlighting the strong synergy between filter and wrapper approaches.

When considering the three wrapper approaches (SFS, SBS, RFE), it is evident that each method possesses distinct strengths. While the Sequential Forward Selection (SFS) and Sequential Backward Selection (SBS) techniques demonstrate proficiency in the incremental addition and removal of features, respectively, Recursive Feature Elimination (RFE) has shown robustness across many datasets due to its recursive elimination approach based on model performance. Sometimes having problem with approach like in the case of SBS in wrapper where its not able to create proper subset of features causing lack of performance .Also SFS is creating large subset containing highly correlated features consuming extra computing power.despite this Wrapper approaches provide customised feature selection, which maximises features for a particular machine learning algorithm to assure top performance. They are especially important in datasets where complex feature interactions have a major impact on prediction results. Wrapper approaches frequently uncover feature combinations that maximise performance by assessing subsets within the context of the model, making them perfect for situations where predictive precision overrides computational economy.

5.2 Conclusion

The hybrid [filter-wrapper] approach provides a combination of computational efficiency and equitable feature selection. By commencing with a filter method, this methodology effectively minimises the feature space, hence simplifying the subsequent wrapper step. This is particularly significant when dealing with datasets that encompass an extensive number of features. Furthermore, this approach combines the holistic viewpoint of filter methods with the preciseness of wrapper techniques, guaranteeing a thorough feature selection process. The presence of duality is advantageous in cases when a dataset contains traits that hold independent significance as well as when they are combined. Furthermore, commencing the analysis with a filter can effectively mitigate the potential problem of overfitting, particularly in situations when there is a scarcity of data. In contrast, the wrapper technique offers a customised approach to feature selection, wherein the selection process is optimised specifically for a certain machine learning model. This approach demonstrates a high level of proficiency in

capturing complex feature interactions and frequently has the ability to select feature sets that optimise model accuracy. Nevertheless, achieving such accuracy may incur a greater processing burden. Fundamentally, although the hybrid approach achieves a compromise between efficiency and comprehensive feature selection, the wrapper approach may exhibit higher performance in some models and datasets, contingent upon the objectives of the research.

The results from this research underline the importance of feature selection not just as a dimensionality reduction tool, but as a strategic measure to enhance model performance. While the improvements were more pronounced in some datasets than others, the overarching theme was clear: judicious feature selection has the potential to bolster machine learning outcomes.

Nevertheless, it is essential to acknowledge that there is no universally applicable approach to feature selection. The effectiveness of different strategies may vary depending on the qualities of the dataset and the context of the problem at hand. The ongoing development of machine learning necessitates a persistent focus on enhancing and advancing feature selection methodologies in order to generate models that are characterised by high efficiency and effectiveness.

5.3 Suggestion for Further Work

Advanced Hybrid Techniques: Explore combinations of multiple filter and wrapper methods to create more advanced hybrid techniques. This could potentially address the recall challenge observed in the current research.

Incorporate Embedded Methods: Extend the research to incorporate embedded methods like LASSO and feature importance from tree-based algorithms. Evaluating these in conjunction with wrapper and filter methods could provide a more holistic view of feature selection.

Feature Engineering: Before feature selection: consider diving deeper into feature engineering. Creating new features or transforming existing ones might enhance

the quality of datasets and improve the performance of machine learning models.

Alternative Evaluation Metrics: Given the observed challenge with recall, it might be beneficial to employ alternative metrics like the Matthews correlation coefficient (MCC) or F2 score, which provide more weight to recall.

:Deep Learning Approaches: Evaluate the impact of feature selection techniques on deep learning models. Neural networks often handle high-dimensional data differently, and understanding the interplay between feature selection and deep learning could be insightful.

References

Agarwal, B. & Mittal, N. (2013), Sentiment classification using rough set based hybrid feature selection, *in* 'Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis', pp. 115–119.

Ansari, G., Ahmad, T. & Doja, M. N. (2019), 'Hybrid filter–wrapper feature selection method for sentiment classification', *Arabian Journal for Science and Engineering* **44**, 9191–9208.

Awad, M., Khanna, R., Awad, M. & Khanna, R. (2015), 'Support vector machines for classification', *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers* pp. 39–66.

Azimi, R., Ghayekhloo, M., Ghofrani, M. & Sajedi, H. (2017), 'A novel clustering algorithm based on data transformation approaches', *Expert Systems with Applications* **76**, 59–70.

Bell, D. A. & Wang, H. (2007), 'A formalism for relevance and its application in feature subset selection', *Machine learning* **41**, 175–195.

Benoît, F., Van Heeswijk, M., Miche, Y., Verleysen, M. & Lendasse, A. (2013), 'Feature selection for nonlinear models with extreme learning machines', *Neurocomputing* **102**, 111–124.

Berrar, D. et al. (2019), 'Cross-validation.'.

Bolón-Canedo, V., Sánchez-Maroño, N. & Alonso-Betanzos, A. (2015), 'Recent advances and emerging challenges of feature selection in the context of big data', *Knowledge-based systems* **86**, 33–45.

Brasil, S., Pascoal, C., Francisco, R., dos Reis Ferreira, V., A. Videira, P. & Valadão, G. (2019), 'Artificial intelligence (ai) in rare diseases: is the future brighter?', *Genes* **10**(12), 978.

Cai, J., Luo, J., Wang, S. & Yang, S. (2018), 'Feature selection in machine learning: A new perspective', *Neurocomputing* **300**, 70–79.

Canbek, G. (2022), 'Gaining insights in datasets in the shade of "garbage in, garbage out" rationale: Feature space distribution fitting', *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **12**(3), e1456.

Chang, Y.-W. & Lin, C.-J. (2011), Feature ranking using linear svm, *in* 'Causation and prediction challenge', PMLR, pp. 53–64.

Chu, X., Ilyas, I. F., Krishnan, S. & Wang, J. (2016), Data cleaning: Overview and emerging challenges, *in* 'Proceedings of the 2016 international conference on management of data', pp. 2201–2206.

Das, A. (2021), Logistic regression, *in* 'Encyclopedia of Quality of Life and Well-Being Research', Springer, pp. 1–2.

Fahmiin, M. & Lim, T. (2020), 'Evaluating the effectiveness of wrapper feature selection methods with artificial neural network classifier for diabetes prediction', pp. 3–17.

Forman, G. et al. (2003), 'An extensive empirical study of feature selection metrics for text classification.', *J. Mach. Learn. Res.* **3**(Mar), 1289–1305.

Gheyas, I. A. & Smith, L. S. (2010), 'Feature subset selection in large dimensionality domains', *Pattern recognition* **43**(1), 5–13.

Hossin, M. & Sulaiman, M. N. (2015), 'A review on evaluation metrics for data classification evaluations', *International journal of data mining & knowledge management process* **5**(2), 1.

Iniesta, R., Stahl, D. & McGuffin, P. (2016), 'Machine learning, statistical learning and the future of biological research in psychiatry', *Psychological medicine* **46**(12), 2455–2465.

Jalali, S. M. J., Moro, S., Mahmoudi, M. R., Ghaffary, K. A., Maleki, M. & Alidoostan, A. (2017), 'A comparative analysis of classifiers in cancer prediction using multiple data mining techniques', *International Journal of Business Intelligence and Systems Engineering* **1**(2), 166–178.

Jović, A., Brkić, K. & Bogunović, N. (2015), 'A review of feature selection methods with applications', pp. 1200–1205.

Li, H., Li, C.-J., Wu, X.-J. & Sun, J. (2014), 'Statistics-based wrapper for feature selection: An implementation on financial distress identification with support vector machine', *Applied soft computing* **19**, 57–67.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J. & Liu, H. (2017), 'Feature selection: A data perspective', *ACM computing surveys (CSUR)* **50**(6), 1–45.

Liu, H. & Setiono, R. (2022), Feature selection and classification–a probabilistic wrapper approach, *in* 'Industrial and Engineering Applications or Artificial Intelligence and Expert Systems', CRC Press, pp. 419–424.

Liu, H., Zhou, M. & Liu, Q. (2019), 'An embedded feature selection method for imbalanced data classification', *IEEE/CAA Journal of Automatica Sinica* **6**(3), 703–715.

Maldonado, S. & López, J. (2018), 'Dealing with high-dimensional class-imbalanced datasets: Embedded feature selection for svm classification', *Applied Soft Computing* **67**, 94–105.

Nargesian, F., Samulowitz, H., Khurana, U., Khalil, E. B. & Turaga, D. S. (2017), Learning feature engineering for classification., *in* 'Ijcai', Vol. 17, pp. 2529–2535.

Panthong, R. & Srivihok, A. (2015), 'Wrapper feature subset selection for dimension reduction based on ensemble learning algorithm', *Procedia Computer Science* **72**, 162–169.

Parvin, H., MirnabiBaboli, M. & Alinejad-Rokny, H. (2015), 'Proposing a classifier ensemble framework based on classifier selection and decision tree', *Engineering Applications of Artificial Intelligence* **37**, 34–42.

Parzen, E. (1982), 'Ararma models for time series analysis and forecasting', *Journal of Forecasting* **1**(1), 67–82.

Peng, Y., Wu, Z. & Jiang, J. (2010), 'A novel feature selection approach for biomedical data classification', *Journal of Biomedical Informatics* **43**(1), 15–23.

Saar-Tsechansky, M. & Provost, F. (2007), 'Handling missing values when applying classification models'.

Saeys, Y., Inza, I. & Larranaga, P. (2011), 'A review of feature selection techniques in bioinformatics', *bioinformatics* **23**(19), 2507–2517.

Simões, G., Galhardas, H. & Gravano, L. (2013), 'When speed has a price: Fast information extraction using approximate algorithms', *Proceedings of the VLDB Endowment* **6**(13), 1462–1473.

Singh, N. & Singh, P. (2021), 'A hybrid ensemble-filter wrapper feature selection approach for medical data classification', *Chemometrics and Intelligent Laboratory Systems* **217**, 104396.

Song, Y.-Y. & Ying, L. (2015), 'Decision tree methods: applications for classification and prediction', *Shanghai archives of psychiatry* **27**(2), 130.

Speiser, J. L., Miller, M. E., Tooze, J. & Ip, E. (2019), 'A comparison of random forest variable selection methods for classification prediction modeling', *Expert systems with applications* **134**, 93–101.

Tan, J., Hammond, J. H., Hogan, D. A. & Greene, C. S. (2016), 'Adage-based integration of publicly available pseudomonas aeruginosa gene expression data with denoising autoencoders illuminates microbe-host interactions', *MSystems* **1**(1), e00025–15.

Tang, J., Alelyani, S. & Liu, H. (2014), 'Feature selection for classification: A review', *Data classification: Algorithms and applications* p. 37.

Tosin, M. C., Majolo, M., Chedid, R., Cene, V. H. & Balbinot, A. (2017), 'semg feature selection and classification using svm-rfe', pp. 390–393.

Tran, N., Schneider, J.-G., Weber, I. & Qin, A. K. (2020), 'Hyper-parameter optimization in classification: To-do or not-to-do', *Pattern Recognition* **103**, 107245.

Wang, D., Nie, F. & Huang, H. (2015), 'Feature selection via global redundancy minimization', *IEEE transactions on Knowledge and data engineering* **27**(10), 2743–2755.

Wong, T.-T. & Yeh, P.-Y. (2019), 'Reliable accuracy estimates from k-fold cross validation', *IEEE Transactions on Knowledge and Data Engineering* **32**(8), 1586–1594.

Xie, J. & Wang, C. (2011), 'Using support vector machines with a novel hybrid feature selection method for diagnosis of erythemato-squamous diseases', *Expert Systems with Applications* **38**(5), 5809–5815.

Yang, F.-J. (2018), An implementation of naive bayes classifier, *in* '2018 International conference on computational science and computational intelligence (CSCI)', IEEE, pp. 301–306.

Yildirim, P. (2015), 'Filter based feature selection methods for prediction of risks in hepatitis disease', *International Journal of Machine Learning and Computing* **5**(4), 258.

Yousefpour, A., Ibrahim, R. & Hamed, H. N. A. (2017), 'Ordinal-based and frequency-based integration of feature selection methods for sentiment analysis', *Expert Systems with Applications* **75**, 80–93.

Zhang, D., Zou, L., Zhou, X. & He, F. (2018), 'Integrating feature selection and feature extraction methods with deep learning to predict clinical outcome of breast cancer', *Ieee Access* **6**, 28936–28944.

Appendix A

A Sample Appendix on Invariant Subspaces

A.1 Evaluating Performance metrics

When evaluating the effectiveness of categorization models, performance metrics are essential. They assess a model's predicted efficacy and accuracy as well as identify its strengths and shortcomings. For example, they contrast possible strengths like high precision with potential drawbacks like limited recall. By comparing and optimising several models, these measures help to ensure the optimal match for a particular job. They also give stakeholders a clear understanding of the dependability of a model. Model improvement and refinement are ultimately driven by this feedback loop.

In the subsequent illustration, we shall examine the performance metrics utilising a target variable which is derived from a video game. The dependent variable, referred to as the "difficulty level" is divided into three distinct categories: easy, medium, and hard.

Formulas:

1. Accuracy: Accuracy = (TP + TN) / (TP + TN + FP + FN)

2. Precision: Precision = TP / (TP + FP)

3. Recall (or True Positive Rate): Recall = TP / (TP + FN)

4. F1 Score: F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

Difficulty Level Class Easy(0): - Accuracy: (TP + TN) / (TP + TN + FP + FN) = (TP + TN) / Total Predictions = 600/600 = 1.0 or 100

- Precision: TP / (TP + FP) = TP / All Positive Predictions = TP/TP = 1.0 or 100

- Recall: TP / (TP + FN) = TP / All Actual Positives = TP/TP = 1.0 or 100 percentage

- F1 Score: Harmonic mean of precision and recall = 2 * (1 * 1) / (1 + 1) = 1.0 or 100per

DifficultyLevel Class Hard (1): Using the provided metrics results:

- Accuracy: (TP + TN) / (TP + TN + FP + FN) = 593/600 = 0.9883 or 98.83

- Precision: TP / (TP + FP) = 230/231 = 0.9946 or 99.46

- Recall: TP / (TP + FN) = 230/237 = 0.9683 or 96.83

- F1 Score: Harmonic mean of 0.9946 and 0.9683 = 2 * (0.9946 * 0.9683) / (0.9946 + 0.9683) = 0.9812 or 98.12

DifficultyLevel Class Medium(2): Using the provided metrics results:

- Accuracy: (TP + TN) / (TP + TN + FP + FN) = 593/600 = 0.9883 or 98.83

- Precision: TP / (TP + FP) = 334/344 = 0.9710 or 97.10

- Recall: TP / (TP + FN) = 334/336 = 0.9950 or 99.50

- F1 Score: Harmonic mean of 0.9710 and 0.9950 = 2 * (0.9710 * 0.9950) / (0.9710 + 0.9950) = 0.9829 or 98.29

In summary, these calculations aid in quantifying the performance of the classifier for each degree of complexity. According to the results based on decision tree classifier, it does fairly well for levels hard and medium of difficulty while doing very well for level easy..

A.2 Data Cleaning and Preprocessing

A.2.1 Data Summarization

A summary of the central tendency, dispersion, and shape of the distribution of a dataset is provided by the pandas method df.describe(). Key data like mean, median, and standard deviation are provided in a brief manner. This approach streamlines the initial data analysis stage of data science by helping to comprehend data distributions, spot outliers, and assess data quality.

```
# Generate basic statistical summary
statistical_summary = df.describe()
statistical_summary
```

	Pregnancies	Glucose	BloodPressure	Ski
count	768.000000	768.000000	768.000000	
mean	3.845052	120.894531	69.105469	
std	3.369578	31.972618	19.355807	
min	0.000000	0.000000	0.000000	
25%	1.000000	99.000000	62.000000	
50%	3.000000	117.000000	72.000000	
75%	6.000000	140.250000	80.000000	
max	17.000000	199.000000	122.000000	

Figure A.1: Describing Dataset

A.2.2 Handling Missing Values:

When handling missing data in pandas, df.isnull().sum() calculates the proportion of null entries in each column. By closing these gaps, data integrity is ensured, model accuracy is increased, and false outcomes are avoided. Data scientists can use imputation approaches by quantifying missing variables to improve the robustness and reliability of their findings. Preventing possible biases in models improves prediction accuracy by filling in data gaps.

```
# Check for missing values in each column
missing_values = df.isnull().sum()
missing_values

[2]
```

```
Pregnancies          0
Glucose              0
BloodPressure        0
```

Figure A.2: Describing Dataset

A.2.3 Handling Duplicate value

Duplicate rows in datasets may be found by using the pandas function 'df.duplicated().sum()'. Eliminating them ensures unique records, which improves data quality. With this optimisation, processing performance is increased and computational burden is decreased. Additionally, removing duplicates guards against biassed outcomes, guaranteeing reliable statistical analyses and models.

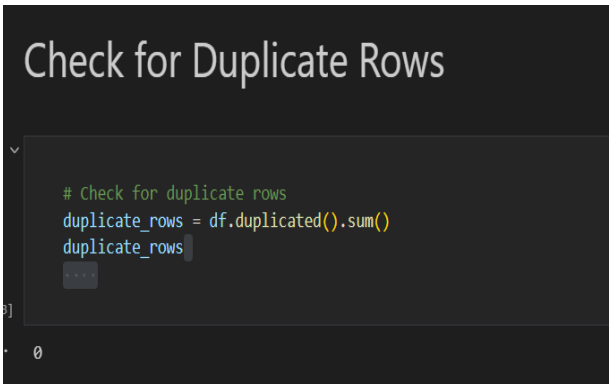


Figure A.3: Describing Dataset

A.2.4 Handling Outliers

Researchers in data science have to discover how to handle outliers for a variety of reasons. Misinterpretations may result from outliers' tendency to distort statistical metrics. Models and insights are guaranteed to be more accurate by locating and dealing with these abnormalities. Better forecasts, more understandable visualisations, and better decision-making are produced by well managed data that doesn't include extreme values.

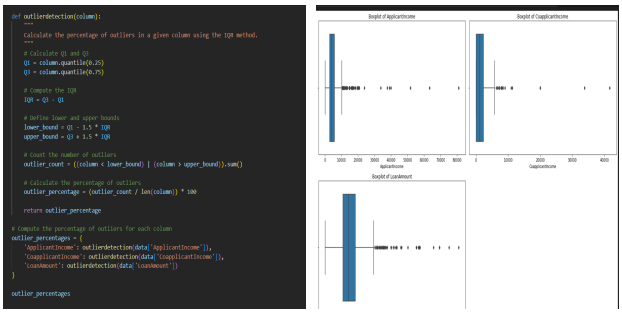


Figure A.4: Describing Dataset

A.2.5 Data Transformation

Data science datasets are refined by data transformation, which includes normalisation and standardisation. Data is scaled between 0 and 1 during normalisation to ensure uniformity. While standardisation increases algorithm efficiency by rescaling data to have a mean of 0 and a standard deviation of 1. Both methods improve the speed and accuracy of model convergence.

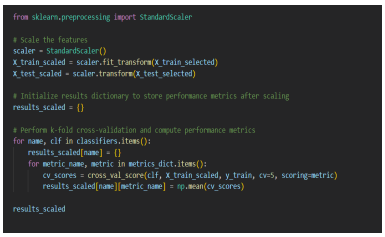


Figure A.5: Scaling

A.2.6 Data Stratification

When datasets are divided into training and testing sets (Xtrain, Xtest, ytrain, ytest), data stratification makes sure that the distribution of classes remains representative. By stratifying the data, we make sure that the proportions of each category in the training and testing sets are comparable, resulting in more precise and broadly applicable model assessments.

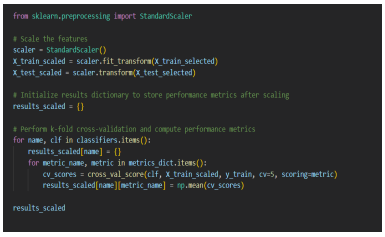


Figure A.6: Stratification(train-test split)

A.3 Feature Selection

A.3.1 Hybrid(Filter-Wrapper)

A advanced feature selection strategy that incorporates the benefits of both the filter and wrapper techniques is the hybrid filter-wrapper method. In the beginning, a filter technique is used to preliminary rank and filter out irrelevant characteristics based on

their statistical qualities, such as Pearson Correlation, Chi-square test, Information Gain (IG), Fisher score, or univariate tests. These techniques assess feature relevance without taking into account any particular learning process. A wrapper technique, such as Recursive Feature Elimination (RFE), is then used. Recursively removing features using RFE and classifiers identifies the best subset for model performance. The statistically significant characteristics that are guaranteed by this hybrid technique also improve model correctness.

The'mlxtend' package is used to implement the feature selection methods, incorporating both logical and statistical approaches, for various datasets based on the aforementioned principles.

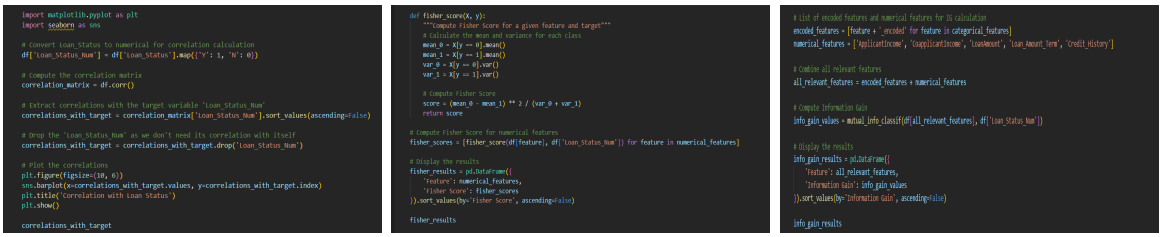


Figure A.7: Filter Method(Pearson Corelation,IG,and Fisher)

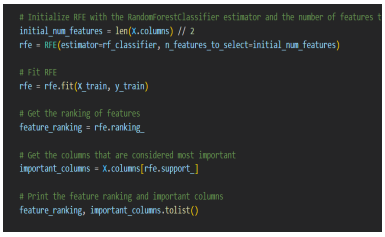


Figure A.8: Wrapper With RFE

A.3.2 Wrapper Method

Utilising a particular machine learning model, the wrapper approach of feature selection assesses subsets of features by directly assessing their performance. Sequential Backward Selection (SBS), Sequential Forward Selection (SFS), and Recursive Feature Elimination (RFE) are important strategies. While SBS starts with all features and gradually removes the least important ones, SFS starts with no features and gradually adds them based on performance improvement. On the other hand, RFE recursively eliminates features by taking feature significance into account. To find the best feature subset for model accuracy and performance, these techniques are frequently paired

with classifiers like random forest, decision tree, logistic regression, SVM, and Naive Bayes.

Figure A.9: Wrapper Methods SFS and SBS

A.3.3 Model Training

In order to develop predictive models, algorithms must be trained. Calculating the likelihood of categorical outcomes is done using logistic regression, which is appropriate for binary categorization. Decision trees divide data hierarchically based on feature values and are excellent for interpretability. Multiple trees are used by random forests, which profit from ensemble learning, to provide reliable predictions, particularly for complicated datasets. SVM identifies hyperplanes dividing classes in high-dimensional spaces. Naive Bayes determines class probabilities using feature independence and is effective for text data because of its probabilistic foundation built on Bayes' theorem. Each model uses training data to identify patterns that may be used to solve certain issues and forecast the future.

Figure A.10: Machine Learning Models

A.4 Hyperparameter Tuning For other datasets

Case 1:Video Game Dataset Hyperparameter Results Certainly! Here's a more understandable breakdown:

1. Logistic Regression: - Parameters: - Regularization Strength (C): 0.1 - Penalty: L1 (Lasso) - Solver: liblinear - Computation Time: 23.00 seconds

2. Support Vector Machine (SVM): - Parameters: - Regularization Strength (C): 1 - Gamma: auto - Kernel: linear - Computation Time: 6.65 seconds

3. Decision Tree: - Parameters: Criterion (measure of quality of a split): gini, Maximum Depth of the Tree: 5, Minimum Number of Samples Required at a Leaf Node: 2, Minimum Number of Samples Required to Split an Internal Node: 2, Computation Time: 1.36 seconds,

Case 2:Indian Diabetes Hyperparameter Tuning results Logistic Regression: Regularization Strength (CC): 10 Penalty Type: L1 (Lasso regularization) Solver Used: liblinear Computational time: 3.5896

Support Vector Machine (SVM): Regularization Strength (CC): 10 Gamma: 0.1 (It defines how far the influence of a single training example reaches) Kernel: Linear Computational time: 157.3582

Decision Tree: Criterion (Measure of Quality of a Split): Entropy Maximum Depth of the Tree: 3 Minimum Number of Samples Required at a Leaf Node: 4 Minimum Number of Samples Required to Split an Internal Node: 2 Computational time: 2.2984

Case 3:Sonar Dataset Hyperparameter Tuning Outcome: Logistic Regression Model: Regularization Strength (C): 10 -Penalty Type: L1 Solver Used: liblinear Computational time: 4.2554

Support Vector Machine (SVM) Model: Regularization Strength (C): 10 Gamma: scale Kernel: rbf Computational time:: 0.3477

Decision Tree Model: Criterion: gini Maximum Depth of the Tree: None Minimum Samples at a Leaf Node: 2 Minimum Samples Required to Split: 5 Splitting Strategy: random Computational time:: 0.7565

Loan Dataset Hyperparameter Tuning Results Logistic Regression:** Regularization Strength: 0.001 Penalty Type: L1

Support Vector Machine (SVM):** Regularization Strength: 0.001 Kernel: linear

Gamma: auto

Decision Tree: Quality Measure: entropy Maximum Tree Depth: 3 Minimum Samples per Leaf: 4 Minimum Samples to Split: 2

case 4:Social Advertisement Dataset Hyperparameter Tuning Results

Logistic Regression: Regularization Strength: 10 Penalty Type: L1 Solver: liblinear

Support Vector Machine (SVM): Parameters: Default parameters

Decision Tree: Quality Measure: entropy Maximum Tree Depth: 5 Minimum Samples per Leaf: 1 Minimum Samples to Split: 2

A.4.1 Hybrid Filter Wrapper Datasets

A.4.2 Case 1: Indian Diabetes

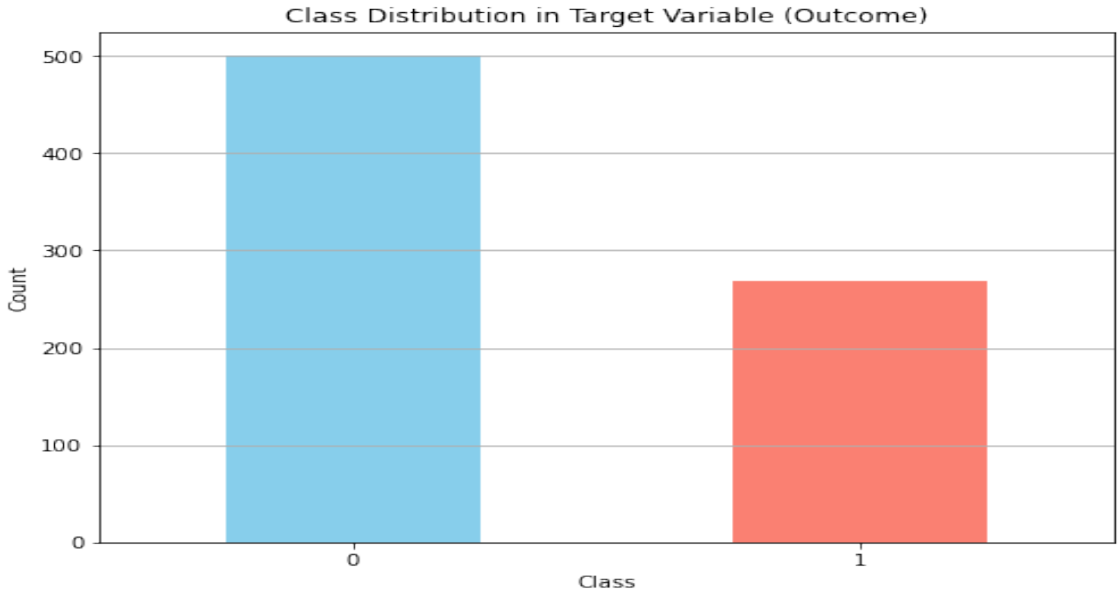


Figure A.11: Target Variable Class Distribution

Class Distribution The target variables may be classified into two separate categories: dianosed 1 and not diagonosed which is zero. This observation suggests that the problem at hand pertains to binary class categorization. It is worth noting that the not diagonosed class exhibits a higher degree of dominance compared to the diagnosed class. As per the analysis we can say that class in the target variabe is not equally distributed.

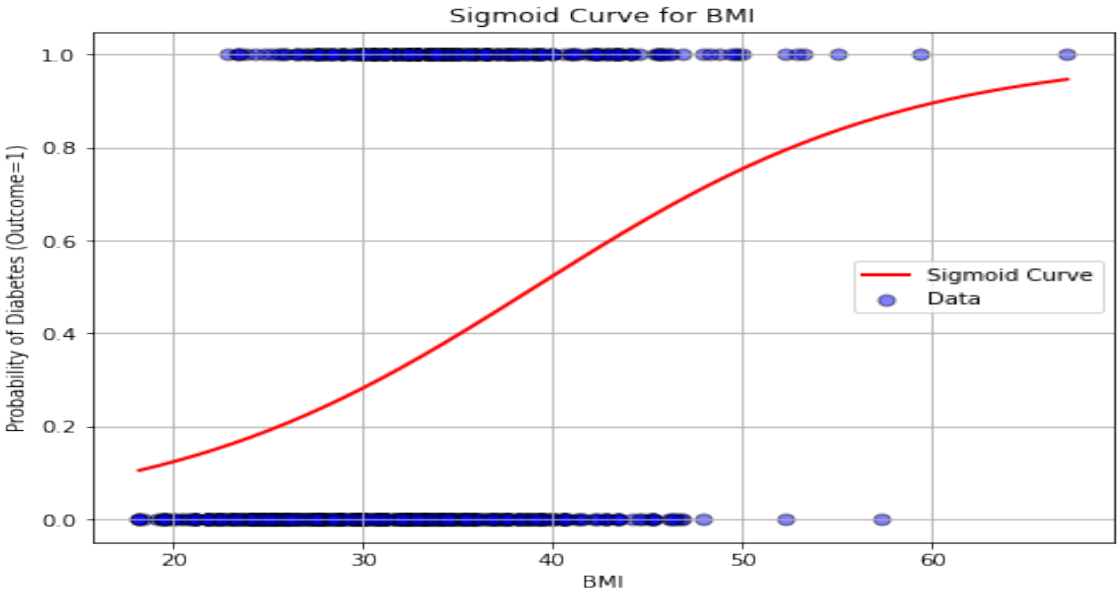


Figure A.12: Sigmoid diagram based on feature selection

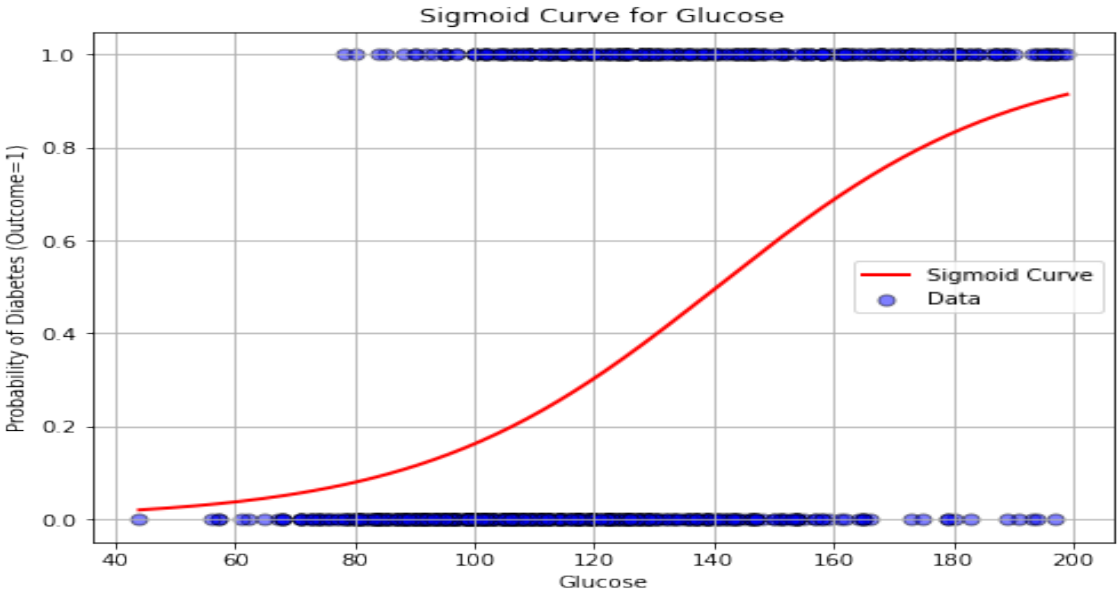


Figure A.13: SIgmoid diagram based on feature selection

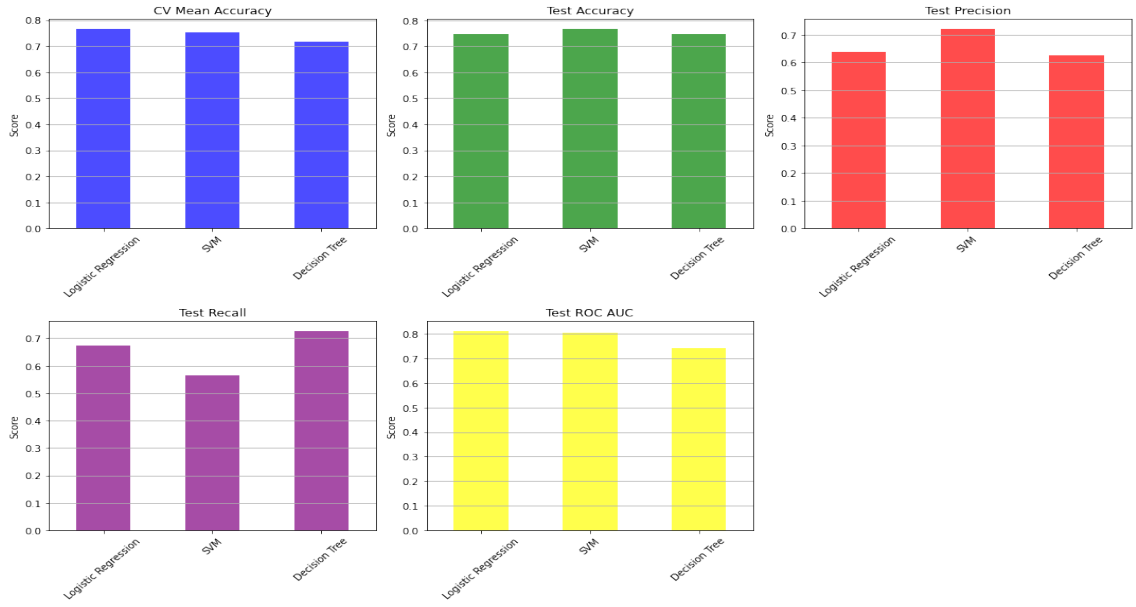


Figure A.14: Bar Plot All Features

	CV Mean Accuracy	CV Std Accuracy	Training Time (s)	Test Accuracy	Test Precision	Test Recall	Test ROC AUC
Logistic Regression	0.765507	0.027605	0.107051	0.746753	0.637931	0.672727	0.812856
SVM	0.752446	0.026483	0.183628	0.766234	0.720930	0.563636	0.806612
Decision Tree	0.721458	0.044615	0.021229	0.759740	0.640625	0.745455	0.756566

Figure A.15: Performance metrics based on all Features

	Model	Accuracy (%)	F1 Score (%)	ROC-AUC
0	Logistic Regression	75.324675	64.150943	0.822222
1	SVM	76.623377	66.037736	0.824426
2	Decision Tree	77.272727	66.019417	0.808815

Figure A.16: Performance metrics based on Selected Features

Result Summary The models trained on chosen features typically outperform those trained on all features when comparing the two sets of data, particularly in terms of ROC AUC. In terms of accuracy, the Decision Tree model in particular appears to profit the most from feature selection. Using a few features may be useful if computing efficiency and simplicity are taken into account, especially if performance is increased or barely affected.

A.4.3 Case 2:Stroke Prediction Dataset

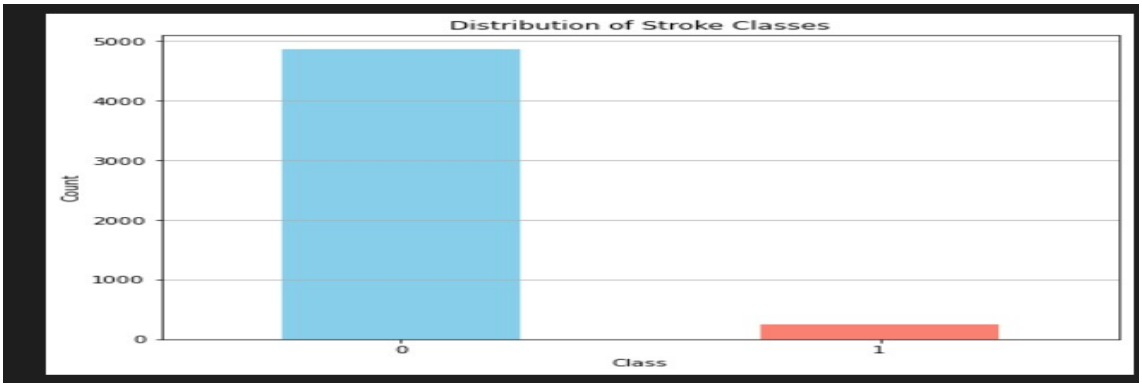


Figure A.17: Class Distribution Diagram

Class Distribution The target features is cateogrised in two class possibility of having stroke which is 1 and no possibility which is encoded by 0.as per the class diagram its clearly stated that there is a hugh difference between classes where 0 dominated more then 75 percent of the entries . this enequal distribution leads to bias results.

	Accuracy	Precision	Recall	F1 Score	ROC AUC	Training Time (seconds)
Logistic Regression	0.936408	0.000000	0.000000	0.000000	0.498437	0.903900
SVM	0.939340	0.000000	0.000000	0.000000	0.500000	0.227855
Random Forest	0.934448	0.000000	0.000000	0.000000	0.497396	0.682536
Naive Bayes	0.388417	0.088036	0.969231	0.161349	0.660136	0.028670

Figure A.18: Performance metrics based on All Features

	Model	Accuracy	Precision	Recall	F1-score	ROC-AUC
0	Logistic Regression	0.937379	0.100000	0.014286	0.025000	0.505580
1	SVM	0.939349	0.000000	0.000000	0.000000	0.500000
2	Random Forest	0.933486	0.000000	0.000000	0.000000	0.496875
3	Naive Bayes	0.384542	0.087544	0.969048	0.160496	0.657961

Figure A.19: Performance metrics based on Selected Features

Result Summary Overall, it appears that using specific characteristics for Logistic Regression has considerable advantages, particularly in terms of accuracy. The benefits of the other models, however, are less obvious. It's also crucial to keep in mind that (as observed in Logistic Regression and Random Forest) employing all characteristics tends to lengthen training duration.The decision on which feature set to use should be based on the specific goals of the analysis. If reducing training time is a priority, selected features might be preferred. If maximizing performance metrics is the goal, then the choice will depend on the specific metric of interest. The unique objectives of the study should be taken into consideration when choosing which feature set to utilise. Selected features could be favoured if cutting training time is a top objective. The decision will be determined by the particular statistic of interest if increasing performance metrics is the objective.

A.4.4 Case 3:Irish Dataset

Result Summary Except for Naive Bayes, all models after hyperparameter adjustment obtained flawless accuracy, precision, recall, F1 score, and ROC AUC of 1 when

	Model	Features	Accuracy	F1 Score	Precision	Recall	ROC AUC
0	Logistic Regression	All Features	1	1	1	1	1
1	Logistic Regression	Column3 & Column4	1	1	1	1	1
2	SVM	All Features	1	1	1	1	1
3	SVM	Column3 & Column4	1	1	1	1	1
4	Random Forest	All Features	1	1	1	1	1
5	Random Forest	Column3 & Column4	1	1	1	1	1
6	Naive Bayes	All Features	0.933	0.933	0.933	0.933	0.99
7	Naive Bayes	Column3 & Column4	0.767	0.776	0.802	0.767	0.83

Figure A.20: Performance metrics based on Selected Features

utilising both all features and a subset (Columns 3 and 4). With all characteristics, Naive Bayes got 99percent ROC AUC and 93.3percent across the majority of measures. Its performance decreased to 76.7percent accuracy and recall, 80.2percent precision, 77.6percent F1 score, and 83percent ROC AUC when Columns 3 and 4 were used. This highlights Naive Bayes' sensitivity to feature selection..

A.4.5 Wrapper based Dataset

Case 1: Brain Stroke Dataset

	Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression	0.9425	0.500000	0.090909	0.153846	0.934584
SVM	0.9450	0.000000	0.000000	0.000000	0.496392
Random Forest	0.9425	1.000000	0.090909	0.166667	0.884079
Naive Bayes	0.8700	0.190476	0.363636	0.250000	0.820587

Figure A.21: Performance metrics based on All

Result Summary Across all methods, the accuracy consistently remains at a high level with few fluctuations. The "All Features" approach has notable accuracy, particularly when utilising Random Forest, and achieves high ROC AUC values. However, it is important to note that the recall metric consistently exhibits low performance across all scenarios. The suboptimal recall adversely impacts the F1 score, but with a somewhat improved outcome shown in the case of Naive Bayes. The performance of "All Features" is commendable; nonetheless, the persistent poor recall seen across

	Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression	0.94500	0.000	0.000000	0.000000	0.943723
SVM	0.94500	0.000	0.000000	0.000000	0.658009
Random Forest	0.94125	0.000	0.000000	0.000000	0.930255
Naive Bayes	0.91750	0.125	0.090909	0.105263	0.891294

	Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression	0.94375	0.000000	0.000000	0.000000	0.916787
SVM	0.94500	0.000000	0.000000	0.000000	0.702742
Random Forest	0.91500	0.200000	0.090909	0.125000	0.891294
Naive Bayes	0.87875	0.153846	0.181818	0.166667	0.789322

```
_warn_prf(average, modifier, msg_start, len(result))
```

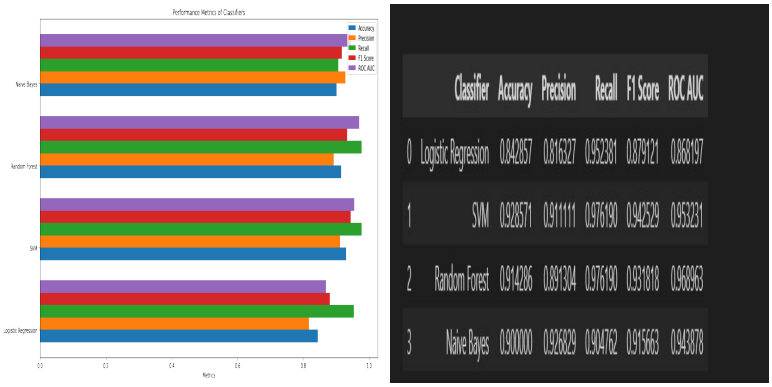
	Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression	0.94625	1.000000	0.090909	0.166667	0.760462
SVM	0.94500	0.000000	0.000000	0.000000	0.560847
Random Forest	0.94000	0.000000	0.000000	0.000000	0.521645
Naive Bayes	0.90375	0.214286	0.272727	0.240000	0.765272

	Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression	0.9425	0.500000	0.090909	0.153846	0.934584
SVM	0.9450	0.000000	0.000000	0.000000	0.496392
Random Forest	0.9425	1.000000	0.090909	0.166667	0.884079
Naive Bayes	0.8700	0.190476	0.363636	0.250000	0.820587

Figure A.22: Performance metrics based on (RFE,SFS,SBS)

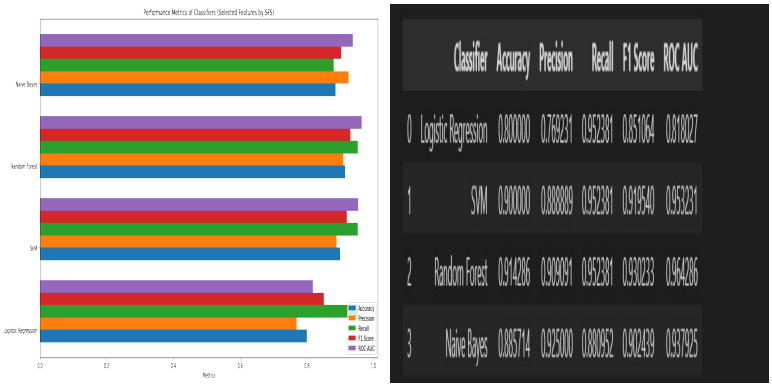
many approaches and models raises concerns. If the prioritisation of recall is of utmost importance, Naive Bayes presents itself as a promising choice that warrants more investigation.

A.4.6 Case 2:Ionosphere Dataset



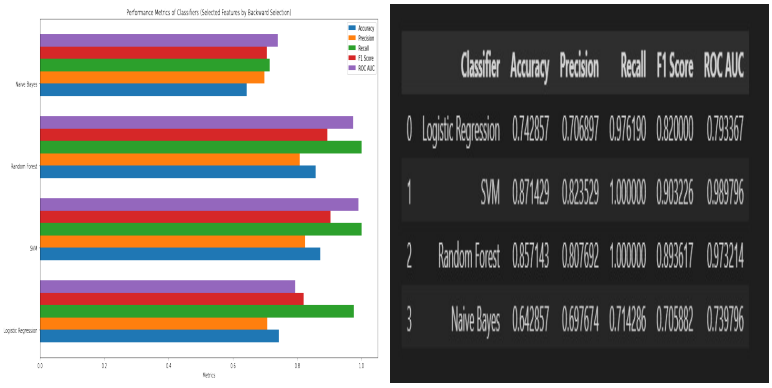
	Classifier	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	Logistic Regression	0.842857	0.816327	0.952381	0.879121	0.868197
1	SVM	0.928571	0.911111	0.976190	0.942529	0.953231
2	Random Forest	0.914286	0.891304	0.976190	0.931818	0.968963
3	Naive Bayes	0.900000	0.926829	0.904762	0.915663	0.943878

Figure A.23: Performance metrics based on RFE



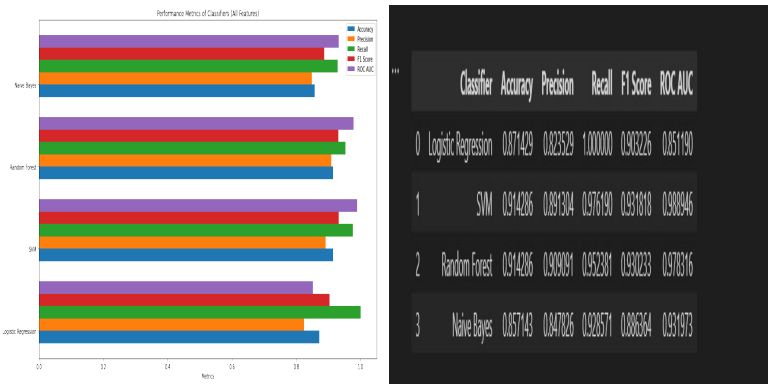
	Classifier	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	Logistic Regression	0.800000	0.769231	0.952381	0.851064	0.818027
1	SVM	0.900000	0.888889	0.952381	0.919540	0.953231
2	Random Forest	0.914286	0.908091	0.952381	0.930233	0.964286
3	Naive Bayes	0.885714	0.925000	0.880952	0.902439	0.937925

Figure A.24: Performance metrics based on Forward Selection



	Classifier	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	Logistic Regression	0.742857	0.706897	0.976190	0.820000	0.793367
1	SVM	0.871429	0.823529	1.000000	0.903226	0.988796
2	Random Forest	0.857143	0.807692	1.000000	0.893617	0.973214
3	Naive Bayes	0.642857	0.687674	0.714286	0.705882	0.739796

Figure A.25: Performance metrics based on Backward Selection



	Classifier	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	Logistic Regression	0.871429	0.823529	1.000000	0.903226	0.851190
1	SVM	0.914286	0.891304	0.976190	0.931818	0.988946
2	Random Forest	0.914286	0.909091	0.952381	0.930233	0.978316
3	Naive Bayes	0.857143	0.847826	0.928571	0.886364	0.931973

Figure A.26: Performance metrics based on All Features

Result Summary

Accuracy:RFE has the highest average accuracy.

Precision: RFE also leads in precision.

Recall: All features slightly edge out RFE in recall.

F1 Score: RFE has the highest F1 score.

ROC AUC: All features have the highest ROC AUC, but RFE is a close second.

In terms of the majority of measures (Accuracy, Precision, and F1 Score), RFE appears to do the best. However, the Recall and ROC AUC are somewhat improved when all features are used. while making decision based on the necessity to prioritise a specific measure. RFE seems to be the better option for a balanced performance across all measures.

A.4.7 Limitation of Hybrid (Filter-Wrapper)

Numerous reasons during research can cause reduction in efficiency of hybrid filter-wrapper techniques. Aggressive initial filtering might leave out key components, impeding future wrapping processes. Particularly when there is a lack of data, the approach may overlook complex feature interactions or overfit a trimmed set. Pure filter or wrapper techniques may be preferred in some sectors, such as Indian diabetes due to particular feature connections. Considering the example of loan dataset While keeping correlated features might increase redundancy and reduce the model's accuracy, choosing the wrong model can overlook the best feature combinations.

A.4.8 Limitation of Wrapper Method

Wrapper methods entail retraining a model for several feature subsets. Particularly for SFS beginning from scratch or SBS starting from the whole set, this can be computationally costly and time-consuming.These approaches are greedy because they always choose the best option without reviewing past decisions. This may result in local optima where causing missing optimal feature set which is encounter by using random initlization considering example of Ionosphere . Wrapper techniques can overfitting because they are continually optimised for the best-performing feature subset on the training data, Particularly if they are not tested on a separate set, this renders the model useless for freshly obtained data, a problem that is addressed by KFold validation using the adult income dataset,brain stroke etc as an example..

R————————————————————————————————-

A.5

.