ANALYZING THE EFFECTIVENESS OF CORRELATION-BASED FEATURE SELECTION COMPARED TO TRADITIONAL METHODS FOR CHURN PREDICTION

By

ELDHOSE VARGHESE
URN: 6793260

A dissertation submitted in partial fulfillment of the requirements for the award of

MASTER OF SCIENCE IN BUSINESS ANALYTICS

September 2024

Faculty of Arts and Social Sciences

University of Surrey

September 2024

Word Count: 14,603

EXECUTIVE SUMMARY

The telecom business faces a pivotal moment where retaining customers is just as crucial as acquiring new ones. With rising competition and growing customer demands, predicting churn has become essential for existence and expansion. This dissertation examines the application of predictive analytics to tackle customer churn, a major challenge for telecom companies, by utilizing well-established feature selection methods to identify potential churners.

The research primarily aims to identify the key factors contributing to churn and to compare the efficacy of correlation-based feature selection with traditional methods like LASSO, backward and forward stepwise selection, and optimal selection in predicting churn. Through rigorous analysis, the study pinpoints Contract Type, Tenure, Monthly Charges, Internet Service, Online Security, Online Backup, Device Protection, Tech Support, and Payment Method as crucial predictors of customer churn.

A thorough analysis was conducted using four machine learning models: Random Forest, Decision Tree, Logistic Regression, and XGBoost. These models were evaluated based on accuracy, precision, recall, F1 score, and ROC AUC, with the results being translated into actionable business insights.

From a business perspective, accurate churn prediction models are instrumental in guiding strategic decisions, allowing telecom companies to deploy focused retention initiatives, enhance marketing strategies, and boost overall customer satisfaction. The study's findings reveal that while correlation-based feature selection is computationally efficient, it generally results in lower prediction quality compared to traditional methods. Specifically, models that used traditional methods like LASSO and stepwise selection outperformed others in terms of accuracy, precision, recall, F1 score, and ROC-AUC score. XGBoost consistently delivered superior results, particularly when paired with traditional feature selection techniques. Overall, the study highlights the trade-offs between computational efficiency and prediction quality, suggesting that although correlation-based feature selection is an efficient alternative, traditional methods are more effective for predicting churn in the telecom industry. Among the models, XGBoost combined with the best subset selection exhibited the highest performance, providing superior prediction quality that is essential for business stakeholders to trust and act on the generated insights.

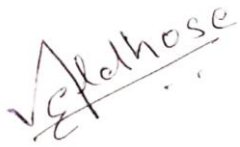
Despite its comprehensive nature, this research has certain limitations. Firstly, the dataset used was sourced from Kaggle and is anonymized and hypothetical, which may limit the applicability of the findings to real-world telecom scenarios. Additionally, since the dataset focuses specifically on customer attrition in the telecommunications sector, the results may not be directly transferable to other industries or domains without further validation. Moreover, the analysis was limited to a fixed set of machine learning algorithms, which means that the findings might differ if other algorithms not included in this study were applied. Lastly, the study's cross-sectional approach, which examines data at a single point in time, does not account for temporal changes in customer behavior or churn patterns, which restricts the ability to generalize the findings across different periods.

Future research could address these limitations by employing real-world datasets from the telecommunications industry to validate the findings and improve the generalizability of the results. Broadening the analysis to include a wider variety of machine learning algorithms and hybrid models could further strengthen the findings and offer a more comprehensive view of the effectiveness of different feature selection techniques. Additionally, longitudinal studies that track customer behavior and churn over time could provide deeper insights into the dynamics of customer attrition and help identify long-term trends that are not captured in a cross-sectional analysis.

In conclusion, this dissertation highlights the significant role that machine learning can play in tackling churn in the telecom sector. By leveraging the predictive power of XGBoost combined with best subset selection and other traditional methods, telecom companies can gain valuable insights into customer behavior, enabling them to take a proactive approach in engaging customers at risk, improving their services, and sustaining a competitive edge in an evolving market.

Declaration of Originality

I hereby declare that this thesis has been composed by myself and has not been presented or accepted in any previous application for a degree. The work, of which this is a record, has been carried out by me unless otherwise stated and where the work is mine, it reflects personal views and values. All quotations have been distinguished by quotation marks and all sources of information have been acknowledged by means of references including those of the Internet**. I agree that the University has the right to submit my work to the plagiarism detection sources for originality checks.**



Eldhose Varghese

01-09-2024

1. TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

CRISP-DM - Cross-Industry Standard Process for Data Mining

LASSO - Least Absolute Shrinkage and Selection Operator

CFS – Correlation-based feature selection

EDA – Exploratory data analysis

KNN - K-nearest neighbors

BIC - Bayesian Information Criterion

ACKNOWLEDGEMENT

First and foremost, I extend my deepest gratitude to Dr. Wolfgang Garn, my dissertation supervisor, whose invaluable guidance and mentorship were crucial throughout this research.

I would also like to express my thanks to Dr. Colin Fu, Dr. Soumya Prakash Rana, and all the other faculty members for their expert knowledge and insights in Machine Learning and Data Analytics, which significantly influenced my dissertation.

Additionally, I acknowledge the unwavering support of my family and friends during my entire academic journey.

1 INTRODUCTION

1.1 Background

Feature selection is crucial for constructing efficient machine-learning models while simultaneously reducing computational complexity and mitigating overfitting. Feature selection is the procedure of pinpointing and choosing key features that significantly enhance the model's predictive capabilities and accuracy (Kiptoon, 2023). Traditional methods such as backward and forward stepwise selection, LASSO, and optimal selection have been extensively employed to identify relevant features, albeit at a significant computational cost. These techniques aim to enhance model accuracy by selecting a subset of features that contribute the most to the prediction task. However, the growing complexity of data and the need for efficient processing have spurred interest in simpler, more computationally feasible methods. One such method is correlation-based feature selection, which ranks features based on their correlation with the target variable, offering a more streamlined approach. This study explores the viability of correlation-based feature selection compared to traditional methods across various machine learning algorithms.

The LASSO method enhances the predictive accuracy and interpretability of statistical models by applying an L1 penalty to the coefficients, which can reduce some of them to zero, thereby enabling variable selection (Tibshirani, 1996). This feature makes it particularly useful for high-dimensional datasets where the number of predictors exceeds the number of observations. On the other hand, correlation-based feature selection ranks variables based on their correlation with the target variable, making it computationally efficient and empirically successful in various studies (Guyon & Elisseeff, 2003). In supervised learning, the Forward-Backward Selection (FBS) algorithm is widely used for its simplicity and broad applicability to different data types. The FBS algorithm includes a forward phase to add variables that improve model performance and a backward phase to remove variables that do not contribute to model improvement. Despite its utility, FBS is challenged by high computational costs and a tendency toward multiple testing problems, which can result in overfitting and the selection of irrelevant variables (Borboudakis & Tsamardinos, 2019). However, there is a lack of comprehensive studies comparing the effectiveness of correlation-based feature selection with traditional methods across different types of machine learning algorithms. This research seeks to address this gap by evaluating

the performance of correlation-based feature selection against methods such as stepwise selection, LASSO, and optimal selection across various algorithms including random forests, decision trees, logistic regression, and XGBoost.

1.2 Academic Motivation

The academic motivation behind this research stems from the need to explore less complex yet effective techniques for selecting features in machine learning. Existing literature highlights the success of traditional methods but also points out their limitations, particularly regarding computational efficiency and complexity. The potential of correlation-based methods, which rank features based on their correlation with the target variable, offers a promising area for exploration. By empirically comparing these methods across various algorithms and datasets, this research seeks to add to the current body of knowledge and offer practical insights for the academic community.

1.3 Problem Statement

The traditional methods of feature selection, while effective, are often computationally intensive and complex, particularly when dealing with high-dimensional datasets (Hastie et al., 2017; Borboudakis & Tsamardinos, 2019). This complexity can pose significant challenges in practical applications where computational resources and time are limited. The correlation-based approach to feature selection is efficient in detecting relevant features and eliminating irrelevant and redundant ones. It enhances the performance of learning algorithms and reduces computational costs (Hall, 1999). However, there is a lack of comprehensive studies that compare its effectiveness with traditional methods across different types of data and machine learning algorithms. This gap in the literature necessitates an empirical investigation to determine whether correlation-based feature selection can serve as a viable alternative without compromising prediction quality.

1.4 Research Aim

This research aims to empirically evaluate the efficacy of correlation-based feature selection compared to traditional methods (backward and forward stepwise selection, LASSO, and optimal selection) across various machine learning algorithms. This evaluation will focus on

prediction quality, and computational efficiency, aiming to determine if simpler methods can offer comparable or superior performance.

1.5 Research Objectives

1. To compare the prediction quality of models using correlation-based feature selection against those using traditional methods.
2. To evaluate and compare the computational efficiency of feature selection based on correlation with the target variable against traditional techniques.
3. To analyze the impact of selecting different numbers of top-ranked features based on correlation on prediction quality.

1.6 Research Questions

1. Is feature selection based on correlations to the target variable as effective as traditional methods in terms of prediction quality?
2. Is feature selection based on correlations to the target variable as effective as traditional methods in terms of computational efficiency?
3. How does the prediction quality of models vary with the number of top-ranked features selected based on correlations?

1.7 Significance of Study

This study holds significant implications for the field of machine learning and its application in business analytics. By potentially validating a more computationally efficient method for feature selection, the research could facilitate quicker and more cost-effective model development. This efficiency is particularly valuable in industries that rely heavily on data analytics, such as finance, healthcare, and marketing, where rapid and accurate predictions can provide a competitive edge. Moreover, the findings could spur further exploration into hybrid feature selection methods that combine the simplicity of correlation-based approaches with the robustness of traditional techniques.

1.8 Scope of Study

This study is confined to evaluating the efficacy of correlation-based feature selection compared with traditional methods across multiple machine learning algorithms, including random forests, decision trees, logistic regression, and XGBoost. The research will focus on developing predictive models and comparing their performance based on various metrics such as prediction quality, runtime, and ease of use. The study will employ quantitative research methods, utilizing secondary datasets to derive insights (Sardana et al., 2023)

1.9 Solution Approach

The research methodology involves several key steps: first, secondary datasets will be collected for analysis. Features will then be ranked by computing their correlation with the target variable. Predictive models will be developed using various machine learning algorithms, and their performance will be compared based on prediction quality, runtime, and ease of use, considering both correlation-based and traditional feature selection methods. Finally, quality curves will be generated to evaluate model performance with incrementally selected top-ranked features.

1.10 Contributions

This research will contribute to the field of machine learning and predictive modeling by:

1. Providing empirical evidence on the effectiveness of correlation-based feature selection.
2. Offering insights into the trade-offs between prediction quality and computational efficiency.
3. Enhancing the understanding of correlation-based methods through theoretical and empirical analysis.
4. Proposing practical guidelines for selecting features in various machine learning applications.

1.11 Dissertation Structure

The dissertation is structured as follows: Chapter 1 provides an introduction, outlining the background, motivation, problem statement, research aim, objectives, questions, significance, scope, solution approach, contributions, and structure of the dissertation.

Chapter 2 reviews existing literature on feature selection methods, highlighting their advantages, limitations, and research gaps. Chapter 3 details the research design, data collection methods, feature selection, model building, and evaluation criteria. Chapter 4 presents the empirical findings, and compares the performance of different feature selection methods. Chapter 5 discusses the results, addressing the research questions. Chapter 6 concludes the dissertation by summarizing the research findings, discussing their significance, and suggesting directions for future research.

2 LITERATURE REVIEW

2.1 Introduction

Feature selection is a crucial component of machine learning that entails selecting the most pertinent features from a dataset. It has a crucial impact on strengthening the performance of the model, mitigating overfitting, and improving interpretability (Jha, 2024). This literature review aims to explore and compare various feature selection techniques, focusing on correlation-based approach, LASSO, forward stepwise selection, backward stepwise selection, and optimal feature selection. The review is organized as follows: each method will be defined, compared with other methods, and analysed to identify research gaps.

2.2 Related work

Correlation-Based Feature Selection (CFS)

Correlation-based Feature Selection (CFS) is a method for selecting subsets of features that exhibit a strong correlation with the target variable while maintaining a low correlation among themselves. The concept of CFS is to choose a collection of features that can yield the highest level of information about the target variable while minimizing redundancy among the features (Sahazada, 2024). The fundamental principle of CFS is grounded in statistical correlation metrics, such as Pearson's correlation coefficient, which measures the linear relationship between two variables. In this context, the coefficient ranges from -1 to 1, where values closer to 1 or -1 indicate stronger relationships, either positive or negative, respectively. The rationale behind using correlation coefficients is that features with higher absolute correlation values with the target variable are considered more relevant, as they potentially explain more variance in the target variable (Banerjee, 2023).

Empirical studies conducted on both discrete and continuous class data sets demonstrate that CFS has the ability to significantly decrease the dimensionality of datasets, while simultaneously preserving or enhancing the performance of learning algorithms. When compared to RELIEF (A machine learning algorithm used for feature selection. It helps identify which features (or attributes) in a dataset are most relevant for predicting the outcome. It does this by examining how well each feature differentiates between instances

(data points) of different classes, using the concept of "nearest neighbors" to compare similar and dissimilar instances.), CFS achieves a higher level of dimensionality reduction for both discrete and numeric class issues. Additionally, CFS performs similarly to RELIEF in terms of the accuracy of learning methods when employing the smaller feature sets. These results indicate that CFS has the potential as a useful feature selector for popular machine-learning techniques (Hall, 2000).

The paper titled "Correlation-Based Feature Selection (CFS) Technique to Predict Student Performance" by Mital Doshi and Dr. Setu K Chaturvedi (Doshi & Chaturvedi, 2014) explores the use of feature selection techniques within the context of predicting student success in engineering college admissions. The authors address the challenge of identifying relevant features from academic and socio-demographic data to enhance the accuracy of predictive models. The study employs several feature selection algorithms, including Chi-square, Information Gain, and Gain Ratio, to identify relevant features. The Fast Correlation-Based Filter (FCBF) technique is used to further refine the selection by removing redundant features. The paper focuses on evaluating the effectiveness of the Fast Correlation-Based Filter (FCBF) algorithm in removing redundant and irrelevant features, thereby improving computation time and predictive accuracy. The study evaluates the following classifiers: NBTree, Naive Bayes, Instance-based k-nearest neighbor (IBK) (A non-parametric method that classifies a new instance based on the majority class of its k-nearest neighbors.), and Multilayer Perceptron (MLP). The dataset comprises 380 instances with 32 attributes related to students' academic and socio-demographic backgrounds. The authors reduced the number of attributes to 17, which were deemed relevant for the study, using the aforementioned feature selection techniques. The results demonstrate that the FCBF algorithm effectively reduces computation time and increases predictive accuracy by selecting the most relevant features. Among all classifiers, IBK is found to be the best in terms of both accuracy and computation time. The study found that the IBK classifier provided the highest accuracy (75%) with four features before applying the FCBF technique. After applying FCBF, IBK achieved a perfect accuracy of 100% with three selected features. The FCBF algorithm is particularly effective in identifying relevant features, with family pressure and student interest being the most important factors for predicting engineering admissions.

The paper titled "Improving Effectiveness of Intrusion Detection by Correlation Feature Selection" by (Nguyen, Franke, and Petrovic, 2010) addresses the importance of feature selection in enhancing the performance of Intrusion Detection Systems (IDS). The authors

emphasize that reducing the number of features while maintaining or improving classification accuracy is crucial for the effectiveness of IDS. The study introduces an automatic feature selection procedure based on the filter method, focusing specifically on the Correlation Feature Selection (CFS) technique. The experiments are conducted using the KDD CUP'99 IDS benchmarking dataset, which includes normal traffic and four types of attacks: Denial of Service (DoS), Probe, User to Root (U2R), and Remote to Local (R2L).

The cut-off or selection of features was based on the following process:

Correlation Feature Selection (CFS) Measure: The authors used the CFS measure, which evaluates subsets of features based on their correlation with the class (e.g., normal or attack) and their inter-correlation with other features.

Optimization Problem: They formulated the feature selection as an optimization problem, converting it into a polynomial mixed 0−1 fractional programming (P01FP) problem. This was further reduced to a mixed 0−1 linear programming (M01LP) problem, which was solved using the branch-and-bound algorithm to find the globally optimal subset of features.

The authors compared the performance of intrusion detection systems (IDS) using all available features against using a reduced set of features selected through their proposed Correlation Feature Selection (CFS)-based method and other feature selection strategies. Classification accuracies are evaluated using C4.5 and BayesNet machine learning algorithms with 5-fold cross-validation. The proposed method improves classification performance, with an average accuracy of 99.41% for C4.5 and 99.52% for BayesNet across the datasets.

The article titled "A Novel Feature Selection Method Based on CFS in Cancer Recognition" by Lu et al. (2012) tackles the challenge of identifying key genes for cancer diagnosis through gene expression data derived from microarray technology. The authors introduce an advanced feature selection method, combining Correlation-based Feature Selection (CFS) with a stratified sampling strategy (CFS-SS) to enhance the accuracy and efficiency of cancer recognition. The study utilized datasets including Leukemia (72 samples with 7129 gene expression levels), Colon Cancer (62 samples with 2000 gene expression levels), and Prostate Tumor (136 samples with 12600 gene expression levels). The classifiers used in the study were K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Multilayer Perceptron (MLP), Naive Bayes (NB), and Decision Tree (J48). The CFS-SS method consistently reduced the number of features while maintaining or improving classification

accuracy. Specifically, for the Leukemia dataset, KNN and SVM achieved 100% accuracy with CFS-SS. For the Colon Cancer dataset, KNN and J48 performed best with accuracies of 93.55% and 91.94% respectively, using CFS-SS. In the Prostate Tumor dataset, MLP reached the highest accuracy of 96.32% with CFS-SS. The CFS-SS method outperformed other feature selection methods (IG, PCA, and CFS) in terms of both the number of features selected and classification accuracy. The study concludes that the CFS-SS method effectively reduces the dimensionality of gene expression data and enhances the performance of cancer recognition models. By integrating filter and wrapper approaches, CFS-SS achieves a balance between computational efficiency and predictive accuracy.

Table 1: Comparison of correlation-based approaches

Aspect	**Hall (2000)**	**Doshi & Chaturvedi (2014)**	**Nguyen, Franke & Petrovic (2010)**	**Lu et al. (2012)**
Focus	CFS's ability to reduce dimensionality and maintain accuracy	Application of CFS and FCBF in predicting student performance	Application of CFS in Intrusion Detection Systems (IDS)	Application of CFS-SS in cancer recognition
Dataset	Empirical studies with discrete and continuous data	380 instances with 32 attributes related to student academic and socio-demographic data	KDD CUP'99 IDS benchmarking dataset	Gene expression data (Leukemia, Colon Cancer, Prostate Tumor)
Feature Selection Technique	CFS, focusing on reducing redundancy and preserving accuracy	CFS combined with FCBF to further refine feature selection	CFS optimized via polynomial mixed 0−1 fractional programming (P01FP)	CFS combined with stratified sampling strategy (CFS-SS)

Classifier(s) Used	Not specified	NBTree, Naive Bayes, IBK, Multilayer Perceptron (MLP)	C4.5, BayesNet	KNN, SVM, MLP, Naive Bayes, Decision Tree
Key Metrics for Feature Selection	Dimensionality reduction, accuracy	Computation time, accuracy	accuracy, number of features selected	accuracy, number of features selected
Results	CFS reduces dimensionality while maintaining accuracy	FCBF improves predictive accuracy and reduces computation time	Proposed CFS method improves classification accuracy and reduces redundancy	CFS-SS reduces features while maintaining or improving accuracy
Comparison Methods	Compared to RELIEF	Chi-square, Information Gain, Gain Ratio	Best-first search, Genetic algorithm	IG, PCA, traditional CFS
Conclusions	CFS is effective for both discrete and continuous data	FCBF and CFS effectively identify key features, with IBK achieving 100% accuracy	The proposed method outperforms other strategies in feature selection for IDS	CFS-SS effectively reduces dimensionality and enhances performance in cancer recognition models

LASSO Feature Selection

The LASSO method was initially developed by Robert Tibshirani in 1996. Lasso Regression, a type of linear regression technique using L1 regularization, is particularly useful for feature selection. By adding the absolute values of the coefficients as a penalty term to the cost function, Lasso encourages the model to reduce some coefficients to zero. This means that

features with non-zero coefficients are considered important, while those with zero coefficients are effectively ignored. This capability of Lasso to shrink some feature coefficients to zero simplifies the model, making it more interpretable and reducing the risk of overfitting. Consequently, Lasso not only regularizes the model but also aids in identifying the most relevant features, making it highly effective for high-dimensional datasets (Verma, 2023).

The paper "Feature Selection using LASSO" by Valeria Fonti and Eduard Belitser (Fonti and Belitser, 2017) effectively demonstrates the use of LASSO for feature selection in both linear models and Generalized Linear Models (GLMs). In linear models, LASSO is applied to predict a response variable using several explanatory variables. This approach is illustrated with the "mtcars" dataset, which includes data on fuel consumption and various automobile design aspects for 32 car models, with the response variable being miles per gallon (mpg). The analysis reveals that weight (wt), number of cylinders (cyl), and transmission type (am) are the most significant predictors of fuel efficiency. GLMs extend linear models to accommodate response variables with distributions other than normal. The paper particularly focuses on logistic regression for binary outcomes, using a high-dimensional dataset related to prostate cancer that includes 6033 gene expressions for 102 samples. The findings emphasize LASSO's effectiveness in selecting the most relevant genes for predicting prostate cancer, identifying key genes like V610 and V1720 as significant predictors. The study concludes that LASSO is a powerful feature selection tool for both linear models and GLMs, as it reduces dataset dimensionality, simplifies models, and enhances prediction accuracy.

The article titled "LASSO: A Feature Selection Technique in Predictive Modeling for Machine Learning" by Muthukrishnan R and Rohini R (Muthukrishnan & Rohini, 2016) examines the significance of feature selection in machine learning, with a particular focus on the LASSO method. The study highlights the necessity of choosing a relevant subset of features to develop models that are both interpretable and generalizable, especially when dealing with a large number of features and a limited number of observations. The authors compare traditional regression methods such as Ordinary Least Squares (OLS) and Ridge regression with LASSO, outlining the benefits and drawbacks of each. Experiments were carried out using both real and simulated data to evaluate the performance of OLS, Ridge regression, and LASSO. The real dataset consists of 422 observations with ten baseline variables, including age, sex, BMI, blood pressure, and six blood serum measurements. The response variable measures the progression of diabetes one year after baseline. Additionally,

synthetic data were generated from a model with predefined coefficients and normal errors to assess the methods under multicollinearity conditions. OLS and Ridge Regression methods included all variables, even those that were not significant, while LASSO selected only the most significant variables (BMI, BP, S3, S5), demonstrating its efficacy in feature selection. LASSO had the lowest median MSE, indicating superior prediction accuracy. Unlike OLS and Ridge regression, LASSO was able to shrink the non-significant coefficients exactly to zero. The study concludes that LASSO surpasses traditional methods like OLS and Ridge regression in feature selection and prediction accuracy. By reducing some coefficients to zero, LASSO effectively decreases model complexity and enhances interpretability. Results from both real and simulated data affirm that LASSO is a robust method for managing high-dimensional datasets.

Forward Stepwise Selection

The forward selection method is a straightforward data-driven approach to model building. It begins with no predictors and incrementally adds significant predictors until a specified statistical stopping criterion is met. In this process, variables are added to the model one at a time. At each step, every variable not yet included in the model is evaluated for potential inclusion. The variable with the highest significance, as long as its p-value is below a predetermined threshold, is then added to the model (Kipsang, 2023).

An essential benefit of forward stepwise selection is its capacity to systematically and methodically build a model, beginning with the most significant elements. This method is particularly valuable when the dataset comprises numerous predictors, as it aids in finding the most crucial aspects that should be incorporated in the model without the need to assess all potential subsets of features, which would be computationally impractical. Moreover, forward selection is especially efficient in datasets where the most prominent traits are relatively limited in number and have substantial individual impacts. In such situations, the approach may rapidly detect and include these crucial characteristics, resulting in a model that effectively captures the essential elements of the data while keeping complexity to a minimum. This particular component of the strategy is advantageous in fields such as biological research, where it is more useful to discover a small number of crucial biomarkers rather than evaluating a wide collection of less significant variables (Draper & Smith, 1998).

The article "Extended Comparisons of Best Subset Selection, Forward Stepwise Selection, and the Lasso" by Trevor Hastie, Robert Tibshirani, and Ryan J. Tibshirani (Hastie et al., 2017) examines various methods for variable selection in regression modeling. This research extends previous studies by conducting a comprehensive set of simulations to compare best subset selection with the lasso and forward stepwise selection techniques. The authors ran extensive simulations, adjusting parameters such as sample size (n), ranging from small (n=100) to large (n=500); number of predictors (p), including high-dimensional data scenarios (p=1000); signal-to-noise ratio (SNR), covering low to high values to evaluate performance under different noise conditions; and predictor autocorrelation (ρ), assessing the impact of correlated predictors on the methods. The simulation results indicated that, in low SNR settings, the lasso outperformed both best subset selection and forward stepwise selection in terms of prediction accuracy. Conversely, in high SNR settings, best subset selection and forward stepwise selection generally surpassed the lasso. Regarding computational efficiency, the lasso method was found to be fast and efficient, while forward stepwise selection was also efficient but slightly slower. Best subset selection, although optimal in certain situations, proved to be computationally intensive and slower, particularly with large datasets. The study concludes that no single method is universally the best. The optimal choice between best subset selection, forward stepwise selection, and the lasso depends on the specific statistical context, particularly the SNR and the correlation among predictors.

The article "Feature Selection in a Credit Scoring Model" by Juan Laborda and Seyong Ryoo (Laborda & Ryoo, 2021) explores improving credit risk management by enhancing the accuracy and interpretability of credit scoring models through feature selection. The study evaluates various classification algorithms: logistic regression, support vector machine (SVM), K-nearest neighbors (KNN), and random forest, assesses the effectiveness of different feature selection methods: Chi-squared test, correlation coefficients (filter method), forward stepwise selection, and backward stepwise selection (wrapper methods). The performance is measured using two primary metrics: mean absolute error (MAE) for prediction accuracy and the number of selected features to gauge model simplicity. The research utilizes a dataset from Taiwan, provided by Chung Hua University, containing 30,000 observations and 25 features, including demographic details, economic conditions, and past repayment behaviors. Empirical results indicate that both forward and backward stepwise selection methods significantly reduced the number of features, with logistic

regression and KNN showing the greatest reduction. Forward stepwise selection consistently improved the MAE across all classifiers, with logistic regression and random forest showing the highest performance gains. In contrast, the Chi-squared test and correlation coefficients method provided moderate improvements in MAE but were less effective than stepwise methods. The study concludes that feature selection is vital for developing effective credit-scoring models. Forward stepwise selection emerged as the most effective method, simplifying models and enhancing prediction accuracy across all classifiers. These findings highlight the importance of incorporating feature selection in credit risk analysis to improve model interpretability and performance.

Backward Stepwise Selection

Backward stepwise selection is commonly used in statistical modeling and machine learning to identify a subset of significant predictors from a larger set of available features. Unlike forward selection, which begins with an empty model and adds features one by one, backward stepwise selection starts with a model that includes all available features and iteratively removes the least significant ones. This process continues until a predetermined stopping criterion is met, such as a threshold p-value or a maximum allowable number of features (An, 2021).

One of the primary advantages of backward stepwise selection is its holistic consideration of the feature set. Unlike forward selection, which evaluates features individually as they are added, backward selection assesses each feature in the context of all other features in the model. This approach allows the method to account for interactions and multicollinearity among predictors, potentially leading to a more accurate and robust final model (Miller, 2002).

Despite its strengths, backward stepwise selection has several notable limitations. A major drawback is the method's computational intensity, especially when dealing with large datasets. Fitting and refitting the model multiple times as features are removed can be time-consuming and resource-intensive, making the method impractical for very large datasets with many predictors. In such cases, the computational cost can become prohibitive, limiting the method's applicability in big data scenarios (Miller, 2002).

The article “Stepwise Feature Selection by Cross Validation for EEG-based Brain Computer Interface” by Tanaka et al. (2006) explores how brain-computer interfaces (BCIs) can aid

paralyzed patients through improved brain wave classification. The study introduces a backward stepwise feature selection method paired with cross-validation to boost classifier performance. Utilizing a dataset of 700 samples (500 for training and 200 for testing), the optimal feature set attained a 91.4% recognition rate, compared to 84.0% using all features. On the test set, the best recognition rate was 82.5% with 72 selected features, versus 76.0% using all features. This method significantly enhanced the generalization performance and computational efficiency of EEG-based classifiers.

Optimal Feature Selection

Best subset selection identifies the optimal model for each subset size, ranging from one variable to p variables when p predictors are available. For instance, with variables a, b, and c, the method evaluates each variable separately for a one-variable model, pairs (ab, ac, bc) for two-variable models, and the combination abc for a three-variable model. The best models are selected based on criteria such as Mallows' Cp, adjusted R^2, or Bayesian Information Criterion (BIC). Mallows' Cp assesses the trade-off between precision and bias, with lower values indicating more precise models with lower test error. BIC evaluates model performance on new data, favoring models with lower test error by assigning them lower BIC values. Adjusted R^2 measures the explanatory power of the model for the response variable, where higher adjusted R^2 values correspond to lower test error. The main benefit of best subset selection is its ability to identify the most optimal models. However, it quickly becomes computationally intensive as the number of predictors increases, due to the need to evaluate 2^p variable combinations. Forward and backward selection methods address this issue by not evaluating every possible combination, thus being more computationally efficient. However, this efficiency comes at the cost of potentially not identifying the absolute best models, unlike the best subset selection (An, 2021).

The paper titled "Variable Selection with Stepwise and Best Subset Approaches" by Zhongheng Zhang (Zhang, 2016) explores techniques for selecting significant variables in regression models, with a focus on stepwise and best subset selection methods. Zhang provides an extensive overview of these methods, highlighting their implementation using the R programming language. The primary aim is to improve model accuracy and interpretability by including only essential variables. The study compares various stepwise selection methods (forward, backward, and both-direction) and best subset selection in regression modeling, using practical examples in R to illustrate their benefits and potential

drawbacks. The "bwt" dataset from the MASS package in R, containing birth weights and various maternal factors, is used for demonstration. The logistic regression models predict low birth weight as the response variable. Using the stepAIC function in R for stepwise selection, the final model retained the variables lwt (Mother's weight), race (Mother's race), smoke (Smoking status during pregnancy), ptd (History of premature labor), ht (History of hypertension), and ui (Uterine irritability). Both forward selection and backward elimination yielded similar models, indicating the robustness of the selected features. For best subset selection, the bestglm function in R identified a model with the variables lwt, ptd, and ht as the best subset based on BIC. Both methods significantly reduced the number of predictors while maintaining model accuracy, with best subset selection providing a more parsimonious model compared to stepwise selection, which included a slightly larger set of variables. The paper concludes that stepwise and best subset selection are both valuable tools for variable selection in regression models. The choice between these methods depends on the analysis context, including the number of predictors, available computational resources, and the need for model interpretability.

Other feature selection techniques

The research paper titled "Churn prediction in telecom using Random Forest and PSO based data balancing in combination with various feature selection strategies" by Adnan Idris, Muhammad Rizwan, and Asifullah Khan (Idris, Rizwan, and Khan, 2012) addresses the challenge of forecasting customer churn in the telecom industry. The authors introduce a hybrid approach called Chr-PmRF, which combines Particle Swarm Optimization (PSO) for data balancing with feature selection techniques and the Random Forest (RF) classifier. This method aims to optimize the selection of instances and features to enhance the performance of the churn prediction model. The study uses a dataset provided by a French telecom company, Orange, consisting of 50,000 instances and 260 features. The following feature selection techniques are used:

Principal Component Analysis (PCA): A dimensionality reduction technique that transforms the data into principal components that capture the most variance.

Fisher’s Ratio: Measures the discriminative power of features by comparing the means and variances between classes.

F-score: Evaluates the discrimination ability of features based on the ratio of between-class variance to within-class variance.

Minimum Redundancy Maximum Relevance (mRMR): Selects features that are highly correlated with the target class while being minimally redundant with each other.

The study uses two classifiers: Random Forest (RF) and KNN. The performance of the proposed Chr-PmRF approach is evaluated using AUC, sensitivity, and specificity. The results show that the combination of PSO-based undersampling, mRMR for feature selection, and RF for classification (Chr-PmRF) outperforms other combinations of techniques. Chr-PmRF achieved an AUC of 0.7511, which was higher than other methods evaluated, demonstrating its effectiveness in predicting churn. The paper concludes that the Chr-PmRF approach is a robust and efficient method for predicting customer churn in the telecommunications industry. By combining PSO-based data balancing with mRMR feature selection and the Random Forest classifier, the proposed method addresses the key challenges of high dimensionality and class imbalance, leading to improved prediction accuracy and model performance.

The article titled "Automated Feature Selection and Churn Prediction using Deep Learning Models" by V. Umayaparvathi and K. Iyakutti (Umayaparvathi and Iyakutti, 2017) explores the application of deep learning techniques for predicting customer churn in the telecommunications industry. The paper addresses the challenges posed by traditional churn prediction models, which rely heavily on manual feature selection—a process that is time-consuming, error-prone, and often tailored to specific datasets. The paper proposes using deep learning models to automate the feature selection process. Deep learning algorithms are capable of automatically learning useful features and representations from raw data, potentially eliminating the need for manual feature selection. The authors developed three different deep neural network architectures to predict customer churn. These models include small and large Feedforward Neural Networks (FNNs) and a Convolutional Neural Network (CNN). The models were evaluated on two real-world telecom datasets:

Cell2Cell dataset: Collected from a large U.S. wireless company with over 70,000 customer records, including 74 predictor variables.

CrowdAnalytix dataset: A smaller dataset provided by the CrowdAnalytix community, consisting of 3,333 records with 18 predictor variables.

The models were evaluated using stratified 10-fold cross-validation to ensure robustness against imbalanced data.

For the CrowdAnalytix dataset, the large FNN achieved better accuracy than both the baseline model and the small FNN.

For the Cell2Cell dataset, the large FNN also outperformed the baseline model, though the small FNN performed worse than the baseline.

Overall, the deep learning models demonstrated performance comparable to traditional classifiers like SVM and random forests, with the added benefit of eliminating the need for manual feature selection. The study concludes that deep learning models are a viable alternative to traditional churn prediction models, offering similar accuracy without the necessity for extensive manual feature engineering.

The research paper titled "Customer Churn Prediction in Telecom Sector with Machine Learning and Information Gain Filter Feature Selection Algorithms" by Yakub K. Saheed and Moshood A. Hambali (Saheed and Hambali, 2021) explores the development of a customer churn prediction model for the telecommunications industry using machine learning (ML) techniques and an innovative feature selection method combining Information Gain (IG) and the Ranker algorithm. The authors propose a combination of the Information Gain (IG) technique and the Ranker method for feature selection. IG evaluates the relevance of features by measuring their entropy (how much information they provide about the target variable). The Ranker algorithm ranks the features based on their IG scores, allowing for the selection of the most important features. The model was tested using a telecom churn dataset with 5000 instances and 21 attributes, and tenfold cross-validation was applied to ensure robust results. The churn prediction model utilized several machine learning classifiers including SVM, Multi-Layer Perceptron (MLP), Random Forest, and Naive Bayes. The performance of these models was evaluated with and without feature selection, using metrics such as accuracy, sensitivity, precision, and F-measure. With Feature Selection (IG+Ranker): The Random Forest (RF) classifier outperformed the other models, achieving an accuracy of 95.02%, a sensitivity of 95%, precision of 94.90%, and an F-measure of 94.70%. The least effective model was Naive Bayes (NB), which still achieved solid results with an accuracy of 88.46%. Without Feature Selection: The performance of the models without feature selection was lower, with the RF model achieving an accuracy of 92.92% and the SVM model achieving 85.86% accuracy. The proposed models were compared with existing

models from previous studies, showing that the Information Gain-based feature selection combined with Random Forest provided superior results in terms of accuracy, precision, and F-measure, making it competitive with state-of-the-art churn prediction models. The paper concludes that using Information Gain combined with the Ranker algorithm for feature selection significantly improves the performance of machine learning models in customer churn prediction.

The journal titled "Intelligent Churn Prediction in Telecom: Employing mRMR Feature Selection and RotBoost Based Ensemble Classification" by Adnan Idris, Asifullah Khan, and Yeon Soo Lee (Idris, Khan, and Lee, 2013), focuses on developing an intelligent churn prediction system for the telecom industry. The authors propose using Minimum Redundancy Maximum Relevance (mRMR) for feature selection. This method selects features that are most relevant to the prediction target (churn or non-churn) while minimizing redundancy between the features themselves. The paper introduces RotBoost, a hybrid ensemble classification method combining Rotation Forest and AdaBoost. RotBoost works by rotating the feature space through Principal Component Analysis (PCA) and applying boosting to handle difficult-to-classify instances. The authors evaluate several ensemble classifiers, including Random Forest, Rotation Forest, DECORATE, and RotBoost, on two standard telecom datasets: the Orange and Cell2Cell datasets. The performance of these models is evaluated using AUC (Area Under the Curve), sensitivity, and specificity. The results show that RotBoost, in combination with mRMR, achieves the best performance, outperforming the other classifiers. Without feature selection, all ensemble models, including RotBoost, perform poorly due to the large number of irrelevant features. However, when mRMR is applied, the models demonstrate significant improvements in performance. RotBoost with mRMR achieves the highest AUC scores of 0.816 on the Cell2Cell dataset and 0.761 on the Orange dataset, demonstrating superior prediction accuracy compared to other models. The study compares mRMR with other feature selection techniques, such as Fisher's Ratio and F-score. The results show that mRMR consistently provides a more informative feature set, leading to better prediction performance across all classifiers. The paper concludes that the combination of mRMR for feature selection and RotBoost for classification offers a highly effective solution for predicting customer churn in telecom datasets. The CP-MRB (Churn Prediction using mRMR and RotBoost) model outperforms other existing approaches and shows great promise for telecom churn prediction, especially in handling high-dimensional and imbalanced datasets.

The research paper titled "Sequential Feature Selection in Customer Churn Prediction Based on Naive Bayes" by Y. Yulianti and A. Saifudin (Yulianti and Saifudin, 2020) focuses on predicting customer attrition in the telecom sector using data mining techniques and sequential feature selection methods to improve model performance. The study uses a dataset consisting of 7,043 records and 20 features, with both categorical and numerical data types. Five different Sequential Feature Selection techniques are proposed to improve model performance: Sequential Forward Selection (SFS), Sequential Backward Selection (SBS), Sequential Forward Floating Selection (SFFS), and Sequential Backward Floating Selection (SBFS). Naive Bayes algorithm is used as the classification model in this study due to its simplicity and effectiveness. The model was implemented with and without feature selection to compare the impact of different feature selection methods on model accuracy and performance. Model performance was evaluated using accuracy and ROC AUC metrics. The results showed that models implementing feature selection performed significantly better than the baseline Naive Bayes model without feature selection. The models that used SBS and SBFS feature selection techniques achieved the highest accuracy and AUC, with 19 features providing the best results.

2.3 Framework

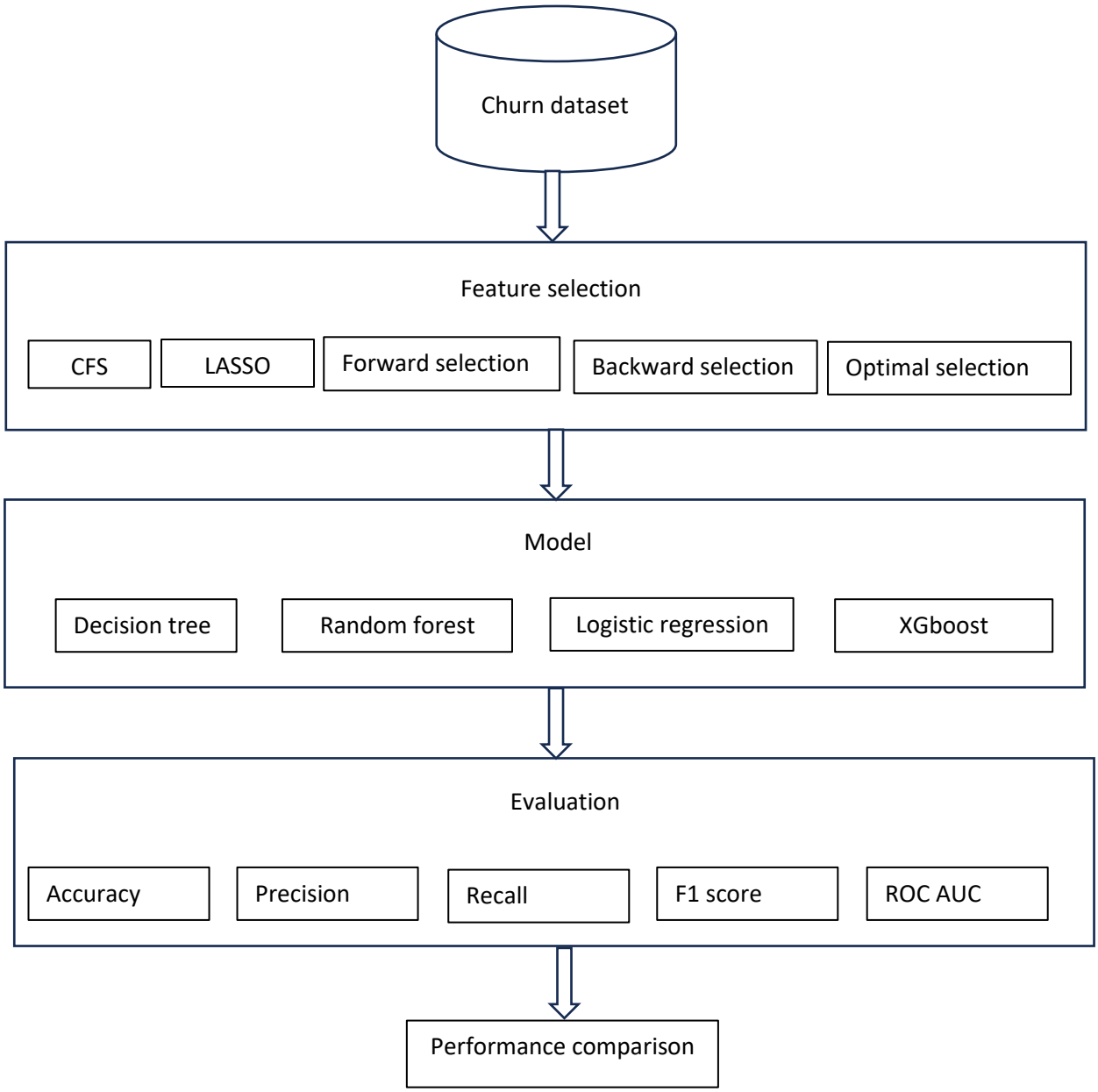


Figure 1 - Churn Prediction Model Framework

2.4 Summary

The literature review explored various feature selection techniques crucial for enhancing machine learning models' performance by identifying the most relevant features from datasets. Correlation-based feature Selection (CFS) focuses on selecting features highly correlated with the target variable and minimally correlated among themselves, proving effective in domains such as student performance prediction and intrusion detection. The LASSO method, employing L1 regularization, is highlighted for its ability to shrink some coefficients to zero, thus simplifying models and enhancing interpretability, particularly in high-dimensional datasets. Forward stepwise selection, a data-driven approach that adds significant predictors incrementally, and backward stepwise selection, which starts with all predictors and removes the least significant, are both effective but vary in computational efficiency and suitability depending on dataset characteristics. Finally, best subset selection identifies optimal models by evaluating all possible predictor combinations, though it becomes computationally intensive as the number of predictors increases, highlighting a trade-off between thoroughness and computational efficiency. Nonetheless, there is a notable absence of comprehensive research comparing the efficacy of correlation-based feature selection with traditional methods across various data types and machine learning algorithms. This gap highlights the need for empirical studies to assess whether correlation-based feature selection can be a practical alternative without sacrificing prediction quality.

3 METHODOLOGY

3.1 Saunder's Research Onion

The Research Onion model, created by Mark Saunders, Philip Lewis, and Adrian Thornhill, as depicted in Figure 1, serves as an extensive framework to assist researchers through the different phases of a research project. This model offers a structured method for developing a research methodology and formulating a research design, which is crucial for ensuring that a study is organized and logically sound (Saunders et al., 2019).

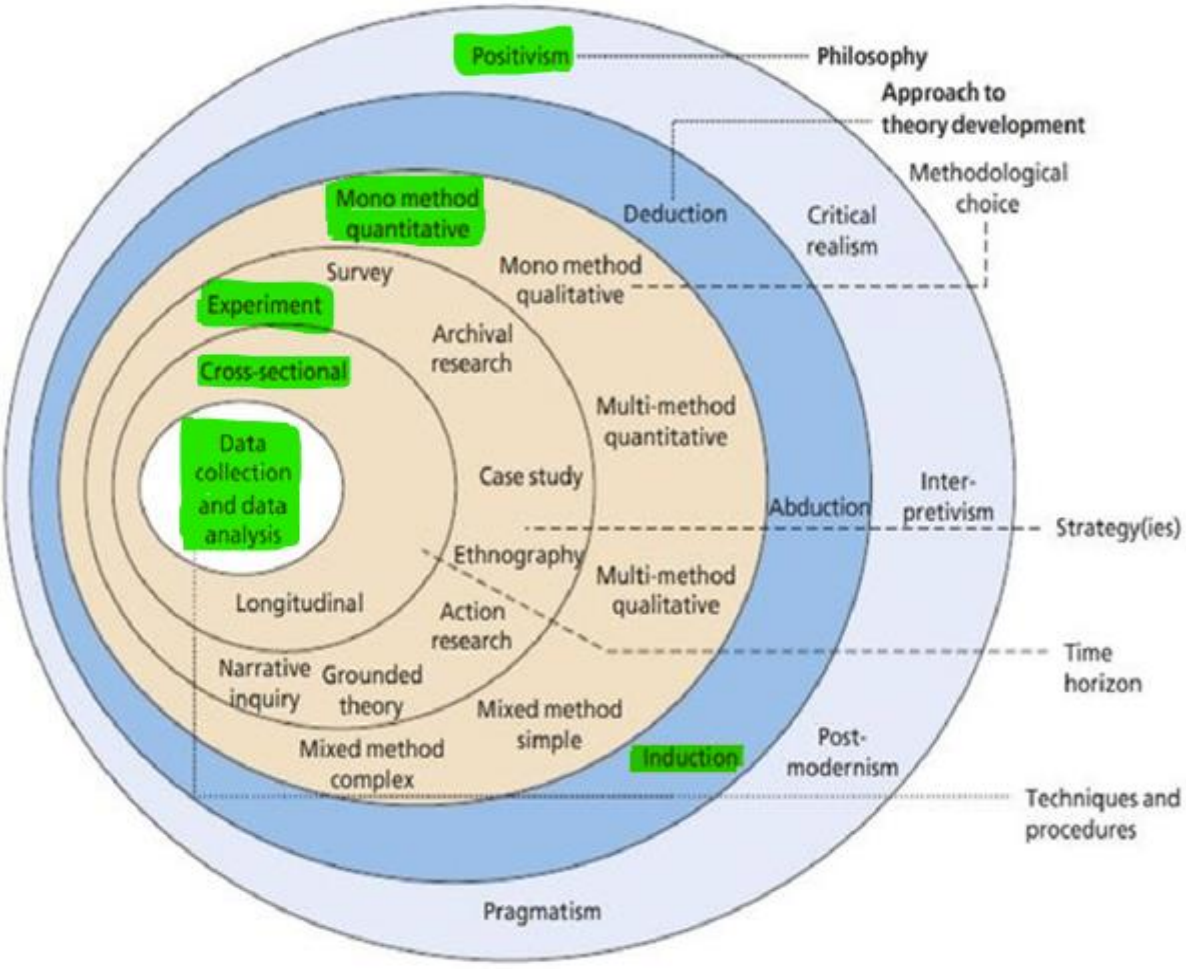


Figure 2- Saunder's Research Onion Model as given in the book RESEARCH METHODS FOR BUSINESS STUDENTS Eighth Edition (Saunders et al., 2019)

3.1.1 Research Philosophy

The research philosophy is located at the outermost layer of the research onion and represents our core convictions regarding the nature of knowledge and reality. It includes your viewpoint on how you see the world and your convictions regarding the appropriate methods

for conducting research. These ideologies, namely positivism, realism, interpretivism, and pragmatism, have a profound influence on the direction and framework of your research.

Three major philosophical paradigms are:

Positivism – Positivism encompasses the idea that an observable, measurable, and analyzable reality exists. Researchers who follow this paradigm use quantitative methods to investigate hypotheses and establish causation. Positivist research prioritizes objectivity and aims to attain generalizability and replicability.

Interpretivism – Interpretivism asserts that reality is subjective and shaped by social factors, highlighting the importance of personal experiences and social environment. Researchers who embrace this paradigm employ qualitative methodologies, such as conducting interviews, making observations, and analyzing documents, to thoroughly investigate and analyze social phenomena.

Realism – Realism adopts a moderate stance, recognizing the presence of an independent reality and the impact of societal frameworks and personal interpretations. Researchers adhering to this paradigm aim to discover both the objective reality and the subjective interpretations of research participants by utilizing qualitative and quantitative approaches.

In this dissertation, we are adhering to the research philosophy of 'Positivism.'

3.1.2 Research Methodology Approach to theory development

The research approach is an integral part of the research onion framework, providing guidance to researchers in choosing the most suitable methodological path for their study.

Deductive Approach - The deductive approach entails the examination of specific hypotheses that are taken from pre-existing ideas or frameworks. Researchers begin by formulating a theory and then gathering empirical data to evaluate hypotheses, frequently utilizing quantitative research methodologies.

Inductive Approach - The inductive technique involves the generation of novel theories or concepts by analyzing observed patterns and facts. Researchers initiate the process by gathering qualitative data and then proceed to construct theoretical frameworks to elucidate the observed events.

In this dissertation we are following an Inductive Approach.

3.1.3 Methodological Choice

In this stage, decisions are made about the research approach, including selecting between mono-method, mixed-method, or multi-method options.

mono-method - Utilizes a singular approach for research, either by collecting qualitative or quantitative data only.

Quantitative research is a method of gathering and analyzing numerical data, typically through the use of surveys, experiments, or statistical analysis. This methodology allows researchers to discern patterns, connections, and associations between variables, offering the chance to make generalizations and draw statistical inferences.

Qualitative research focuses on comprehending the intricacy and abundance of human experiences and societal processes. Researchers collect qualitative data through interviews, observations, and content analysis to investigate meanings, settings, and subjective perspectives.

Mixed method - refers to the utilization of many research methods, typically using both qualitative and quantitative methodologies, in order to accomplish diverse objectives and overcome the limitations associated with relying solely on a single method.

Multi-method - employs a diverse range of methodologies that extends beyond the traditional qualitative and quantitative approaches.

In this dissertation we are employing mono method by using quantitative data.

3.1.4 Research Strategy

The subsequent stage entails formulating a strategic plan and trajectory for your research, typically guided by your research questions and objectives. Strategies can include experimental investigations, surveys, case studies, action research, and other methods.

For this project we are using '**experimental investigation**' strategy to check the effectiveness of correlation-based feature selection against traditional methods for churn prediction considering a telecom churn dataset.

3.1.5 Time Horizon

It pertains to the duration and time frame of the research study

Cross-Sectional Approach - Cross-sectional studies gather data concurrently to analyze variables and relationships at a particular point in time. This method offers a static representation of the research subject, without taking into account any modifications or advancements that may have occurred over time. Survey research frequently employs cross-sectional studies.

Longitudinal Approach - Longitudinal studies entail the gathering of data from the same participants over a prolonged duration. Researchers can analyze alterations, patterns, and progressions over some time. Longitudinal studies can be categorized as either prospective, where data is collected in the future, or retrospective, where data is collected from the past. This technique is beneficial for examining processes, behaviors, and enduring consequences.

In this project we follow '**Cross-Sectional Approach**' where the data is collected in the past at a certain point of time.

3.1.6 Techniques and Procedures

In the innermost layer, the emphasis is on the techniques and methods used for data collection and analysis.

Data Collection Methods

Interviews

Interviews consist of individual or collective discussions between the researcher and participants. Surveys can be organized in an organized manner, where a fixed set of questions is used, or in an unstructured manner, which allows for free-flowing discussions. Interviews yield abundant qualitative data and enable researchers to delve into participants' viewpoints, encounters, and viewpoints.

Observations

Observations entail the methodical act of closely observing and meticulously documenting behaviors, events, or phenomena as they occur in their authentic environment. Researchers can adopt the role of participant observers, where they actively engage in the observed situation, or non-participant observers, where they observe without direct involvement.

Observations yield significant qualitative data regarding behaviors, interactions, and contextual elements.

Questionnaires

Questionnaires are formal tools that have a set of predetermined questions. They are usually given to participants to collect quantitative or qualitative data. Questionnaires are an effective method for gathering data from a large number of participants, and they can be completed by the participants themselves or by an interviewer.

Focus Groups

Focus groups consist of researcher-facilitated group conversations. Participants express their perspectives, viewpoints, and personal encounters about a particular subject. Focus groups facilitate participatory and dynamic discussions among participants, offering valuable insights into shared viewpoints, group dynamics, and collective attitudes. This approach is valuable for investigating intricate social matters and obtaining a wide range of perspectives.

Document Analysis

Document analysis involves the examination of written or recorded resources, such as texts, reports, archive documents, and media sources. Researchers scrutinize these materials to extract pertinent information and discerning insights on the research topic. Document analysis can offer historical background, policy analysis, or textual evidence to substantiate study conclusions (Ellul, 2023).

data analysis approaches

Thematic Analysis

It is the identification of patterns, themes, and significance within qualitative data. Researchers methodically analyze and classify the data to provide themes or concepts that accurately represent the core of the material. Thematic analysis enables a thorough investigation of the data and the recognition of repetitive patterns.

Statistical Analysis

It is the process of applying statistical approaches to numerical data. Researchers utilize statistical approaches to compress, examine, and interpret numerical data. This investigation

includes detailed descriptions, logical conclusions, relationships between different elements, regression analysis, and testing of hypotheses.

Content Analysis

It is a methodical strategy used to examine textual or visual data. Researchers analyze and classify distinct aspects or themes in the content to obtain significant insights. It can be conducted using either quantitative or qualitative methods. Quantitative analysis involves examining the frequency and distribution of specific words or concepts, while qualitative analysis focuses on interpreting and understanding the meaning of the content.

For this project, we utilized secondary data that may have been gathered through the use of questionnaires. We applied both thematic and statistical analysis methods to analyze the data.

Together, these layers comprise the Research Onion, with each stage playing a role in shaping and carrying out a research project.

3.2 CRISP-DM

CRISP-DM, which stands for Cross-Industry Standard Process for Data Mining, as shown in Figure 2, emerged in the late 1990s as a systematic methodology for developing projects related to data mining and knowledge discovery (Talaviya, 2023). The CRISP-DM framework consists of six distinct phases:

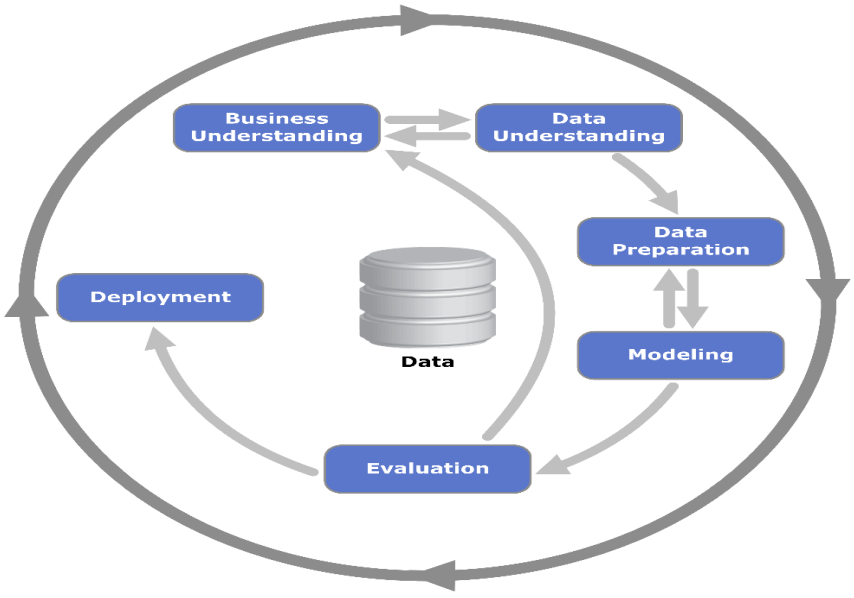


Figure 3 - CRISP-DM process (Chumbar, 2023)

3.2.1 Business understanding

The primary objective of this initial phase is to comprehensively comprehend the project's goals, objectives, and requirements from a business standpoint. Subsequently, this understanding is translated into a data mining problem definition and an initial plan consisting of a specified set of tasks and expected outcomes. The purpose of this plan is to effectively accomplish the project-level objectives.

The primary objective of this research is to evaluate whether feature selection based on correlations is as effective as traditional methods, such as backward and forward stepwise selection, LASSO, and optimal selection, in developing predictive models. The business relevance lies in enhancing the efficiency and effectiveness of feature selection processes in machine learning, which could lead to faster and more accessible model development. By simplifying the feature selection process without compromising model performance, businesses can achieve quicker insights, reduce computational costs, and maintain a competitive edge across industries such as finance, healthcare, and marketing.

3.2.2 Data Understanding

This phase begins with the collection of initial data and continues with tasks such as describing features, conducting primary data analysis, and performing EDA. These activities help you become acquainted with the data, identify issues with data quality such as missing values and inconsistent entries, and identify interesting subsets of data to form hypotheses about confidential information.

Data Collection

The dataset for this research was obtained from Kaggle (BlastChar, 2018). The data focuses on customer attrition in the telecommunications sector. However, in order to adhere to industry standards and protect the confidentiality, specific information about the company has been anonymized. This method safeguards confidential data while also guaranteeing that the conclusions can be applied to the wider telecommunications industry. The dataset has 7,043 customer records, each distinguished by different demographic, account, and service consumption factors. Although the data is hypothetical, the qualities and patterns identified closely correspond to the dynamics of the real-world telecom business. This dataset is not

only suitable for academic purposes, but also beneficial for extracting insights that may be applied to current and future real-world telecom operations.

Attributes Description

The dataset includes 21 attributes, offering a detailed view of customer profiles.

*Demographic Information***:** Includes attributes like 'Gender', 'Partner', and 'Dependents'

Account Information: Comprises 'Customer ID', 'Tenure in Months', 'Contract Type', 'Payment Method', 'Monthly Charges', 'Total Charges', and the pivotal 'Churn' which indicates customer status.

Service Usage: Includes details on the customer's usage of various services like 'Phone Service', 'Multiple Lines', 'Internet Service', and additional offerings including 'Online Security', 'Online Backup', 'Device Protection', 'Tech Support', etc.

Data Dictionary

The dataset contains 7,043 customer records with 21 attributes that provide a detailed overview of customer demographics, service usage, and financial aspects within the telecom sector. The target variable in the dataset is ‘Churn’ with a binary classification of yes or no. The data dictionary below provides details on each attribute and its corresponding description.

Table 2: Data Dictionary (BlastChar, 2018)

Attribute	Description
Customer ID	A unique ID that identifies each customer.
Gender	Whether the customer is a male or a female
Senior Citizen	Whether the customer is a senior citizen or not (1, 0)
Partner	Whether the customer has a partner or not (Yes, No)
Dependents	Whether the customer has dependents or not (Yes, No)
Tenure	Number of months the customer has stayed with the company.
Phone Service	Whether the customer has phone service or not (Yes, No).

Multiple Lines	Whether the customer has multiple lines or not (Yes, No, No phone service)
Internet Service	Customer's internet service provider (DSL, Fiber optic, No).
Online Security	Whether the customer has online security (Yes, No, No Internet Service).
Online Backup	Whether the customer has online backup (Yes, No, No Internet Service).
Device Protection	Subscription to a device protection plan: Yes or No.
Tech Support	Subscription to a technical support plan: Yes or No.
Streaming TV	Usage of Internet service to stream TV: Yes or No.
Streaming Movies	Usage of Internet service to stream movies: Yes or No.
Contract	The customer's current contract type.
Paperless Billing	Status of paperless billing choice: Yes or No.
Payment Method	How the customer pays their bill (bank transfer, credit card, etc).
Monthly Charge	The customer's current total monthly charge for all their services.
Total Charge	The customer's total charge for a specified quarter.
Churn	The status of the customer at the end of the quarter.

3.2.3 Data Preparation

The data preparation step encompasses all necessary operations to produce the ultimate dataset, inputted into the modeling tools, starting from the first raw data. Data preparation tasks are prone to being executed repeatedly and without a specific sequence. The tasks involve working with data tables, handling missing values, variable encoding, selecting relevant features, performing feature engineering, data balancing, and transforming and cleaning the data to prepare it for modeling tools and procedures. Finally, the dataset is divided into separate training and testing sets to evaluate the model's predictive power ensuring that the data is well-prepared, relevant, and correctly formatted for the upcoming modeling stage.

Handling Missing Values

The ‘TotalCharges’ column in the dataset has 0.16% missing values and the corresponding rows are deleted from the dataset.

Removing Irrelevant Columns

The Customer ID, which serves as a unique identifier for each customer, does not influence churn prediction and is therefore excluded.

Data Balancing

Datasets where more than 50% of the entries belong to one class are considered imbalanced. Most machine learning algorithms perform better with balanced datasets, as they aim to maximize overall classification accuracy or similar metrics. When faced with imbalanced data, these algorithms often establish decision boundaries that favor the majority class, resulting in incorrect classification of the minority class (Santiago, 2023).

A balance check was performed to evaluate the distribution of our target variable, ‘Churn,’ which is a crucial step before moving forward with feature selection and modeling. Recognizing the risk of model bias due to imbalanced classes, we calculated the imbalance ratio, revealing a significant disparity, with the minority class (Churned) accounting for only 26.5% compared to the majority class (Stayed). This calculation provided a quantitative insight into the level of imbalance.

To address this issue, we employed the SMOTE-ENN technique to balance the dataset. The primary aim of this approach was to increase the representation of the minority class (Churned) while also eliminating potentially mislabeled or noisy data points. This process led to a more balanced distribution between the classes. The main goal was to reduce the bias introduced by the majority class, allowing for a more accurate and comprehensive capture of the unique features of each class.

Variable Encoding

Categorical variables are converted into numerical ones using one-hot encoding, as many machine learning algorithms necessitate numerical input and cannot directly process

categorical data. To maintain consistency and accuracy in our analysis, this conversion was applied uniformly to all categorical variables.

Feature selection methods

Various methods such as CFS, backward and forward stepwise selection, LASSO, and best subset selection are utilized to find the best feature subset. For traditional methods such as forward, backward, and best subset selection, the best feature subset is selected based on the BIC score.

3.2.4 Modeling

During this phase, a range of modeling approaches are chosen and implemented based on the problem statement. The parameters of these techniques are adjusted to provide the best possible modeling performance. Generally, there are multiple methodologies available for addressing the same data mining problem category. Certain approaches have distinct prerequisites regarding the data format it necessitates. Thus, it is often required to return to the data preparation phase at this level.

Dataset Splitting

Before applying the predictive models, the dataset was divided into separate subsets designated for training and testing. This partitioning is essential for assessing the models' predictive performance in a reliable and impartial way. A commonly used split ratio of 70:30 was utilized, with 70% of the data assigned for training and 30% reserved for testing (Sree, 2023).

Evaluation

The models are evaluated within the framework of the business objectives set at the initial phase. This evaluation has the potential to uncover novel business insights, allowing for the refinement of the model accordingly.

Evaluation Metrics

The performance of each predictive model in this study was evaluated using five essential metrics: Accuracy, Precision, Recall, F1-Score, and the Area Under the Curve

(AUC) of the Receiver Operating Characteristic (ROC) curve. These metrics are computed as follows:

- Accuracy: It refers to the proportion of correct predictions in the total prediction made.

$$Accuracy = \frac{\text{Number of Correct Prediction}}{\text{Total Number of Prediction}}$$

- Precision: It is defined as the ratio of true positive predictions to the total positive predictions.

$$Precision = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

- *Recall (Sensitivity):* It is defined as the ratio of true positive predictions to the actual positive cases in the dataset.

$$Recall = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

- *F1-Score:* It is defined as the harmonic mean of Precision and Recall

$$F1\ Score = 2 * \left(\frac{Precison * Recall)}{(Precision + Recall)}\right)$$

- *AUC-ROC:* The Area Under the Receiver Operating Characteristic Curve represents the model's ability to discriminate between the classes.

Algorithms used

Logistic Regression

Logistic Regression is a widely used technique in predictive modeling, especially for churn prediction in the telecom industry, due to its ease of use, clarity, and effectiveness in handling binary classification tasks. This model calculates the likelihood of binary outcomes, such as churn or no churn, making it particularly well-suited for binary classification scenarios. Logistic Regression operates on the principle of utilizing a logistic function, also known as a sigmoid function, to capture the relationship between the independent variables and the predicted outcome. The logistic function transforms any real-valued input into a probability, compressing it into a range between 0 and 1.

Probability of an instance belonging to one class is given by the formula:

$$P(class = 1) = 1/(1 + e\text{^}(-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n)))$$

where x_1 to x_n are the independent variables associated with the instance, and β_0 to β_n represent the regression coefficients (Franco, 2023)

Decision Tree Classifier

The Decision Tree Classifier algorithm is highly regarded in machine learning due to its intuitive decision trees and rule sets, making it particularly well-suited for industries like telecom, which deal with complex and large-scale data. It transforms intricate data into straightforward visual models, making it easier to identify the key factors influencing customer churn. The algorithm excels at selecting the most significant variables, efficiently trimming away irrelevant data to prevent overfitting, ensuring that the models are not only accurate but also generalize effectively to new data. One of the algorithm's main strengths lies in its advanced pruning techniques, where it autonomously handles multiple decisions using reasonable defaults. Its core method involves post-pruning the tree by first building a large, potentially overfitted tree and then systematically removing nodes and branches that contribute little to classification accuracy. The decision to apply this algorithm to the customer churn dataset was motivated by its interpretability, ability to capture complex decision paths, and effectiveness in dealing with mixed data types, including categorical variables. Its transparency also facilitates the extraction of actionable insights (RPubs,2018).

Random Forest Classifier

The Random Forest algorithm, an ensemble learning method that combines predictions from multiple decision trees, greatly improves prediction accuracy. In the context of churn forecasting, precision is crucial. By minimizing the likelihood of errors from individual trees, Random Forest delivers more dependable and consistent predictions. This is especially beneficial in pinpointing customers at risk of churning, where high accuracy is vital for successful retention strategies (Omila, 2023).

XGBoost

XGBoost (Extreme Gradient Boosting) algorithm is a popular choice for churn prediction due to its high precision and accuracy in model creation. XGBoost has a significant potential for improvement through adjustments in its hyperparameters compared to other algorithms which provides a higher probability of enhancing its performance (Loffredo, 2023).

3.2.5 Deployment

The completion of the model typically does not mark the conclusion of the project. Although the primary objective of the model is to analyse and enhance comprehension of the data, it is essential to convey the knowledge or insights derived from the modelling in a manner that allows the end users of the model to utilize it. End users include operational-level staff, business executives, and customers.

The optimal feature selection method, combined with the best-performing machine learning model, will be implemented on new customer data to accurately predict potential churners, enabling the application of personalized retention strategies such as offering free data, free SIM cards, six-month free subscriptions, etc, with the ultimate goal of retaining customers and driving revenue growth.

4 ANALYSIS AND FINDINGS

4.1 Exploratory data analysis (EDA)

Distribution of target variable

The original distribution of the 'Churn' variable in the dataset is as follows:

- Not Churned: 73.5%
- Churned: 26.5%

The distribution suggests that the dataset is imbalanced, with many more non-churned customers than churned ones.

Univariate Analysis

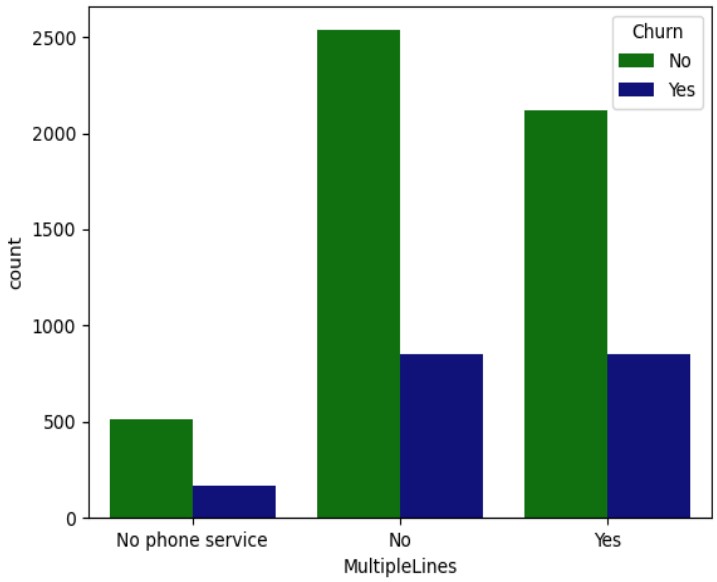


Figure 4: Multiple Lines vs Churn

Multiple phone lines do not have any impact on churn

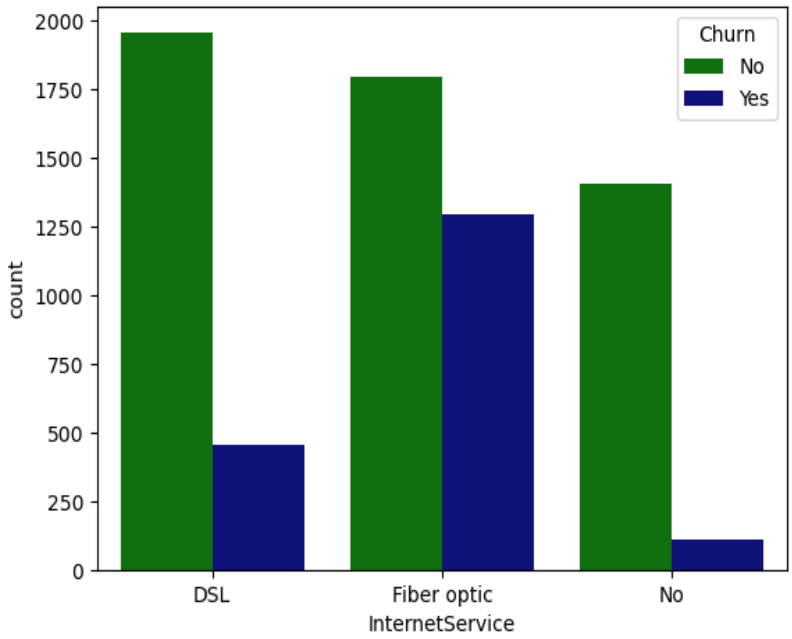


Figure 5: Internet Service vs Churn

Higher customer churn is observed with fiber optic type

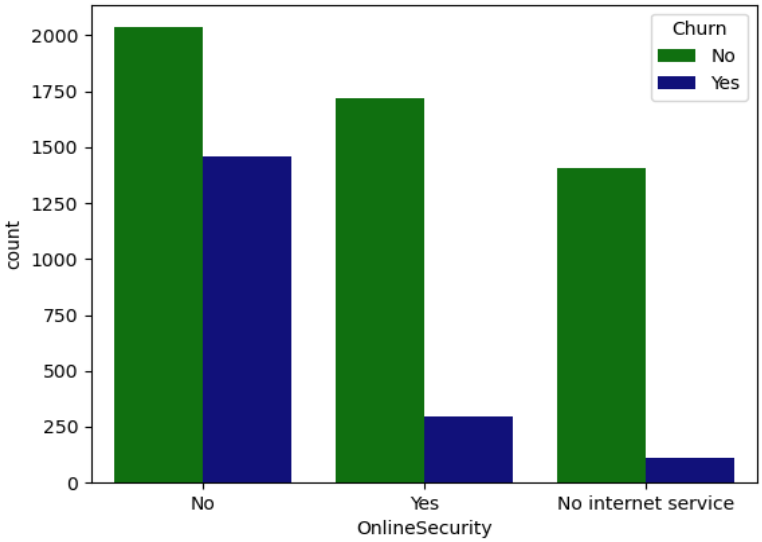


Figure 6: Online Security vs Churn

Customers with online security are less likely to churn

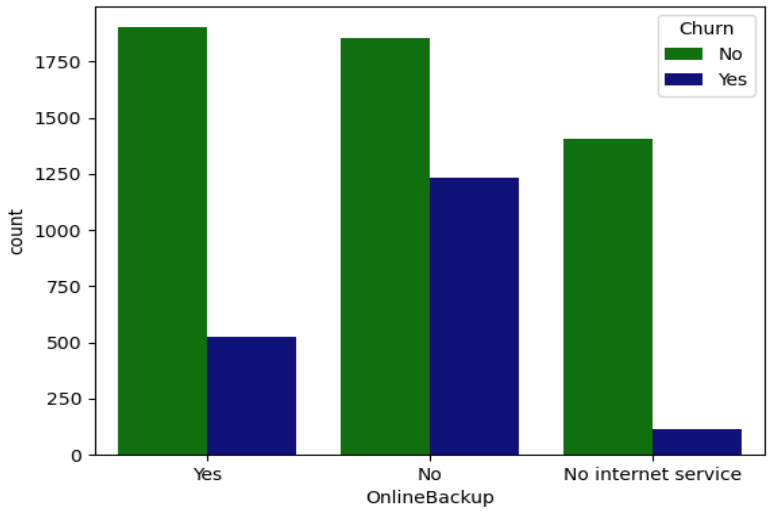


Figure 7: Online Backup vs Churn

Customers using backup services are less prone to churn

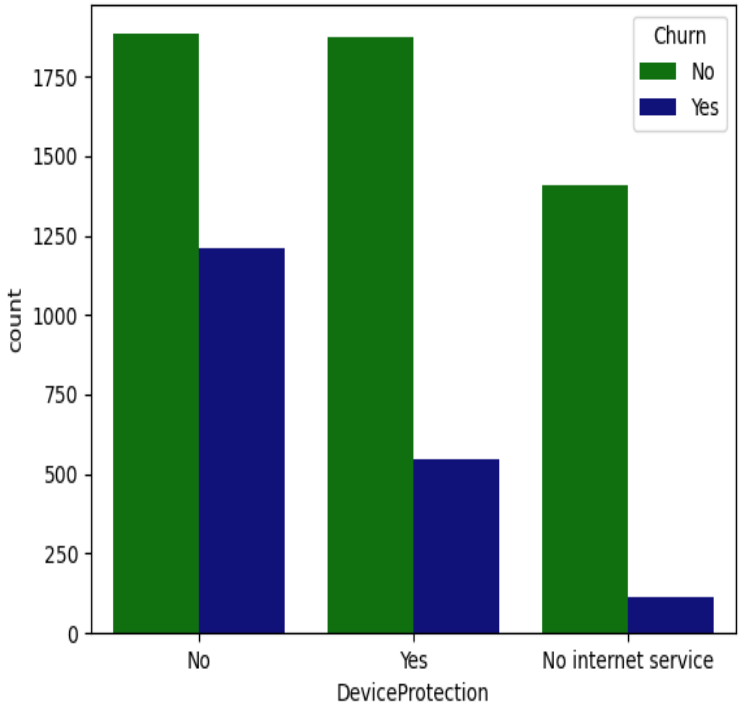


Figure 8: Device Protection vs Churn

Customers without device protection services are more prone to churn

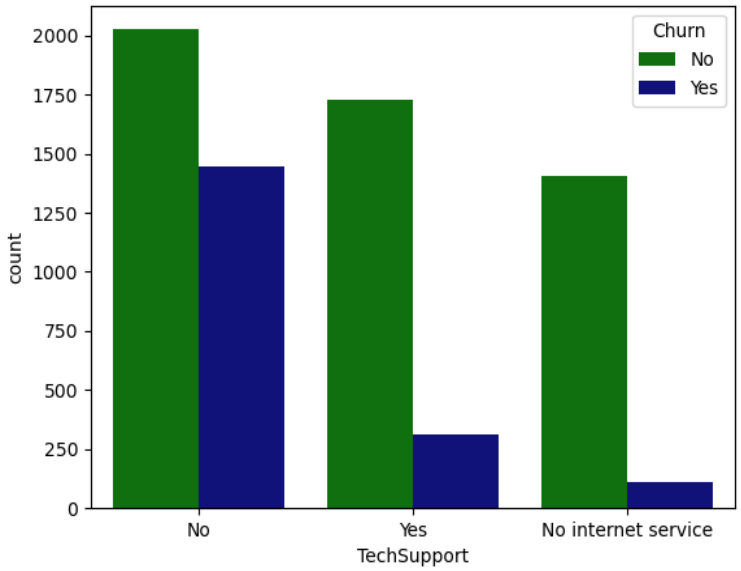


Figure 9: Tech Support vs Churn

Customers without tech support are more likely to churn

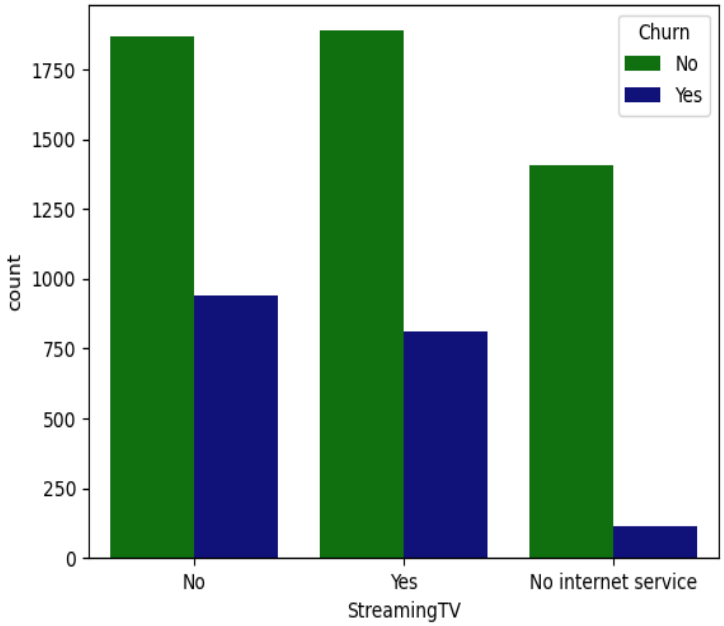


Figure 10: Streaming TV vs Churn

Streaming TV services does not have a considerable impact on churn

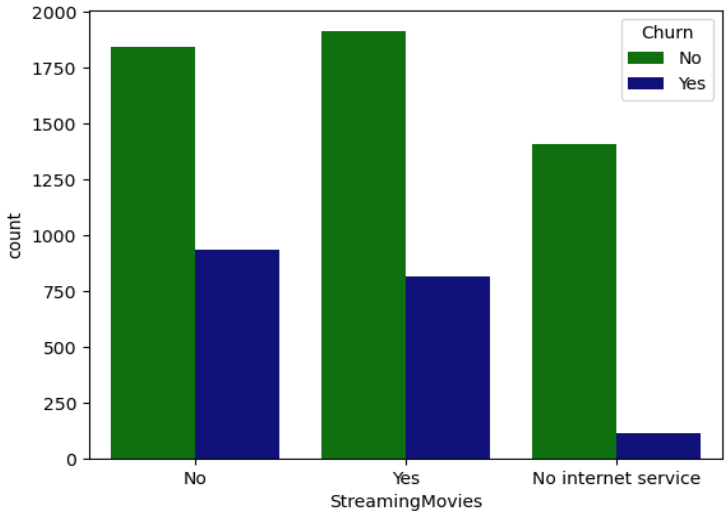


Figure 11: Streaming Movies vs Churn

Streaming movie services does not have a significant impact on churn

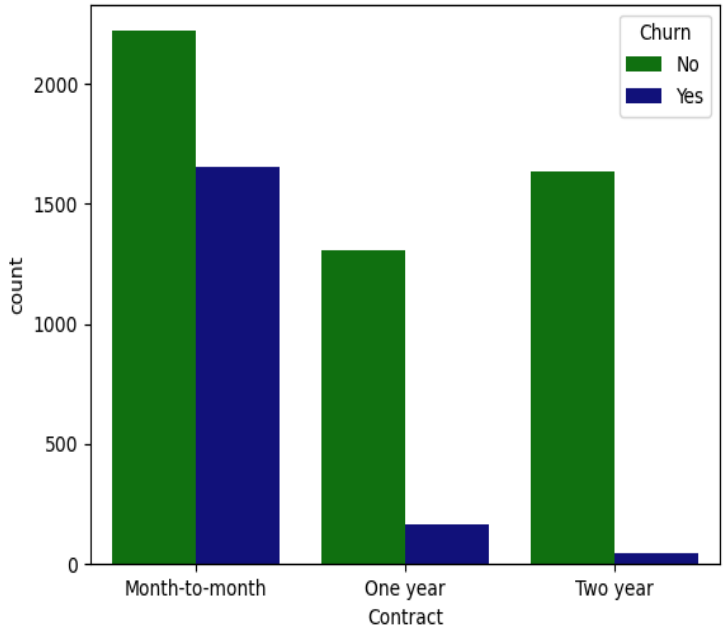


Figure 12: Contract vs Churn

Customers with longer contracts churn less due to contractual obligations, while month-to-month customers may churn more freely.

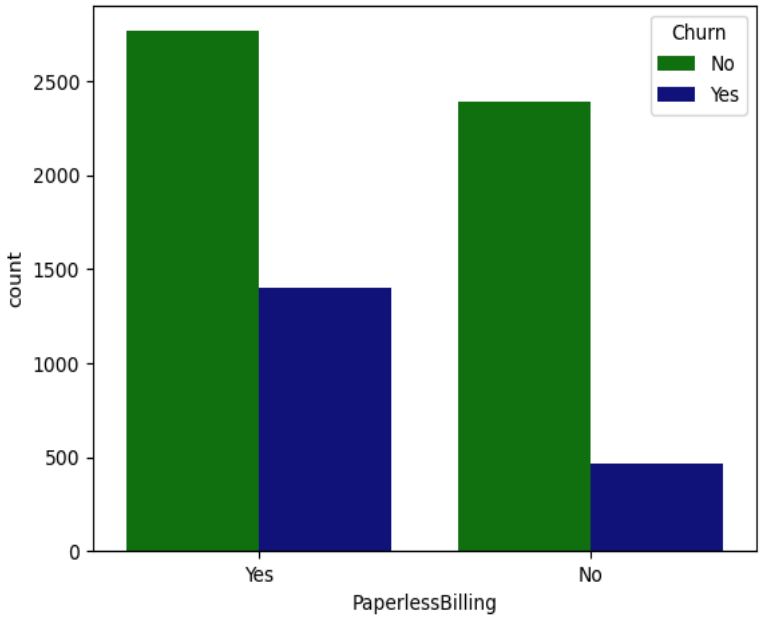


Figure 13: Paperless Billing vs Churn

Customers who choose paperless billing churn more.

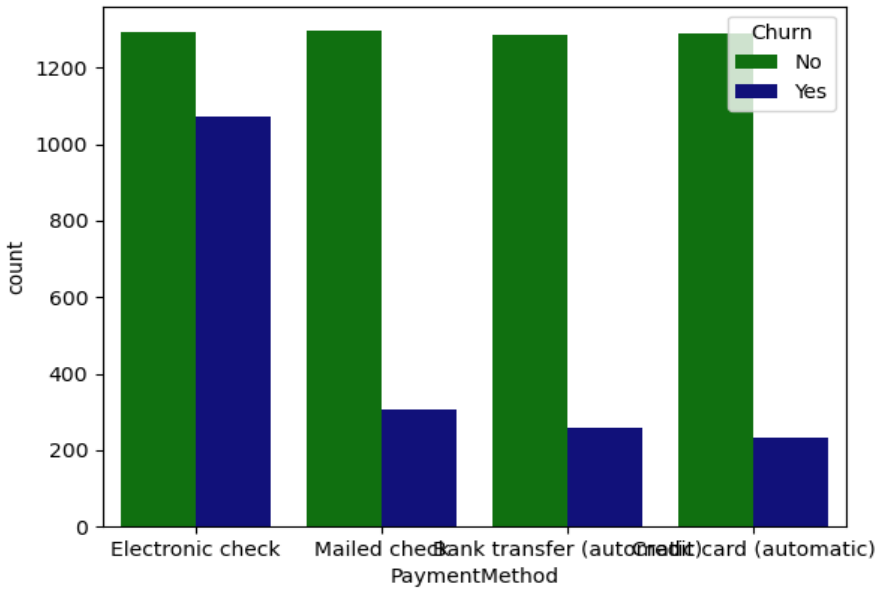


Figure 14: Payment method vs Churn

Customers utilizing the electronic check payment method exhibit the highest rate of churn.

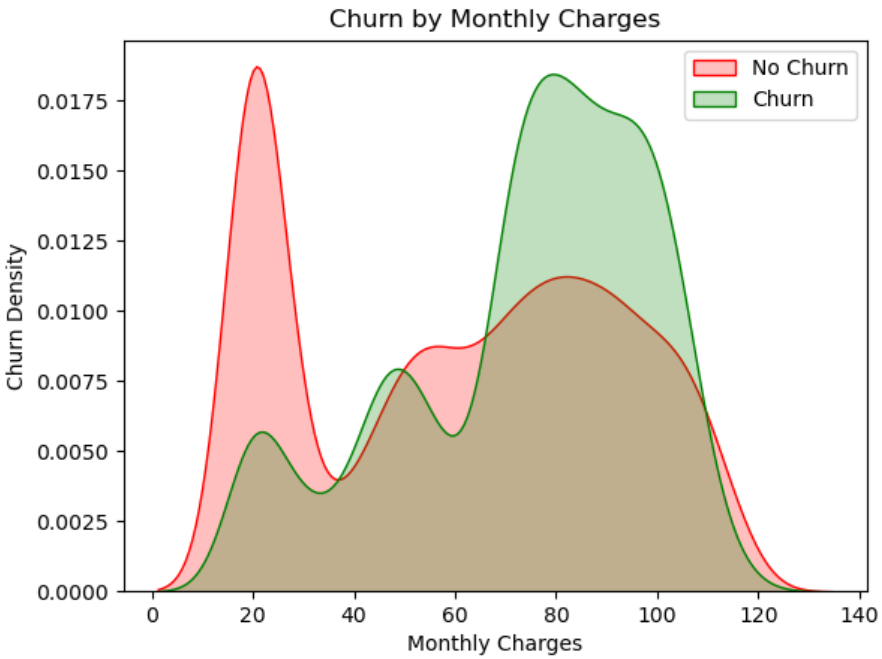


Figure 15: Churn by monthly charges ($ per month)

The above plot depicts a positive relationship between the amount customers are charged monthly and their likelihood of churning. higher charges are associated with higher churn rates.

4.2 Prediction Quality Analysis of Feature Selection Methods

Accuracy

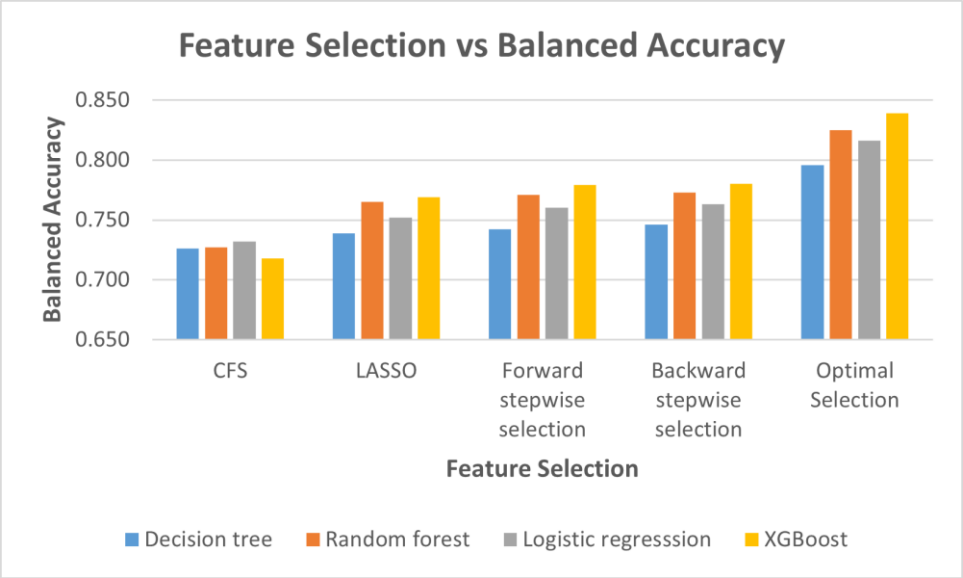


Figure 16: Feature Selection vs Accuracy

Figure 16 illustrates accuracy scores for various feature selection methods across four machine learning models: Decision Tree, Random Forest, Logistic Regression, and XGBoost. Among these methods, Correlation-based Feature Selection (CFS) demonstrates the lowest accuracy, with scores ranging from 0.718 to 0.732 across all models, indicating that it is the least effective. In contrast, the LASSO method shows a noticeable improvement over CFS, with accuracy scores between 0.739 and 0.769, suggesting a better performance. Forward Stepwise Selection further enhances accuracy, building on LASSO's performance, as it achieves scores ranging from 0.742 to 0.779. Additionally, Backward Stepwise Selection performs in a manner similar to Forward Stepwise Selection, reinforcing the benefits of these methods. Ultimately, Optimal Selection stands out by delivering the highest accuracy across all models, culminating in a top score of 0.839 for the XGBoost model.

Precision

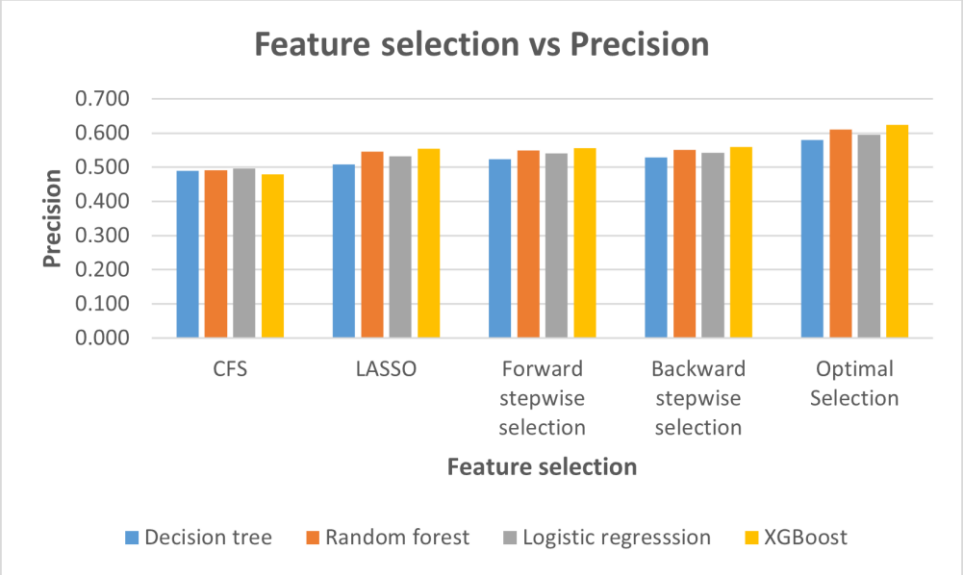


Figure 17: Feature selection vs precision

The graph indicates a clear trend where precision improves progressively as more advanced feature selection methods are used. With CFS, all models exhibit relatively low precision, with values ranging from 0.479 for XGBoost to 0.497 for Logistic Regression, suggesting that solely relying on correlations between features and the target variable is not sufficient to enhance the precision of the predictions. When LASSO is applied, which introduces regularization by shrinking less important feature coefficients, precision improves notably, particularly for Random Forest and XGBoost, where their values rise to 0.546 and 0.554, respectively, indicating that LASSO helps reduce overfitting and improve the precision of these models. Forward Stepwise Selection, which incrementally adds features based on their impact on model performance, further boosts precision, with Random Forest and XGBoost benefiting the most, reaching 0.549 and 0.556, respectively. The slight increase in precision seen with Backward Stepwise Selection, where less significant features are systematically removed, suggests that this method refines feature selection more

effectively, as seen in the precision scores of 0.551 for Random Forest and 0.559 for XGBoost, which slightly outperform Forward Stepwise Selection. Optimal Selection, which evaluates all possible feature combinations, yields the highest precision across all models, with XGBoost achieving 0.624, Random Forest at 0.610, and Logistic Regression at 0.595, reflecting the advantages of selecting the most optimal subset of features for enhancing precision.

Recall

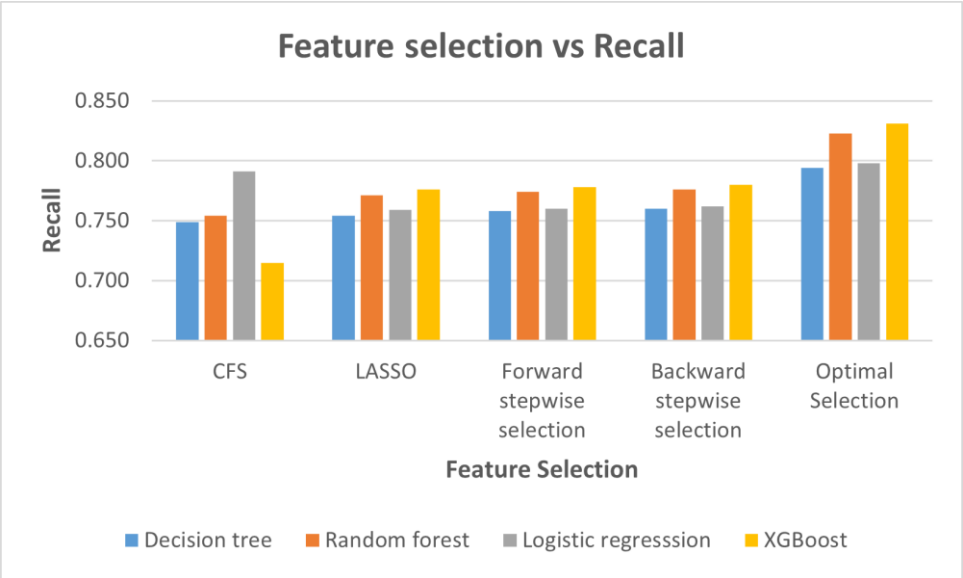


Figure 18: Feature Selection vs Recall

The results show that as the feature selection method becomes more advanced, recall tends to improve across all models. Starting with CFS, the recall values are moderate, with Logistic Regression achieving the highest recall at 0.791, while XGBoost records the lowest recall at 0.715. When LASSO is applied, recall improves slightly for all models, with Random Forest and XGBoost showing notable gains, reaching recall values of 0.771 and 0.776, respectively, suggesting that LASSO enhances the models' ability to correctly identify positive instances. Moving to Forward Stepwise Selection, recall continues to improve slightly, with Random Forest reaching 0.774 and XGBoost improving to 0.778, demonstrating that this method effectively enhances recall by selecting the most impactful features. Backward Stepwise Selection, which eliminates less significant features, yields similar but slightly better recall results than Forward Stepwise Selection, with XGBoost reaching 0.780 and Random Forest achieving 0.776, indicating that feature elimination helps refine the model's ability to detect true positives. Finally, Optimal Selection consistently delivers the highest recall across all models, with XGBoost reaching an impressive 0.831, followed closely by Random Forest at 0.823 and Logistic Regression at 0.798, demonstrating that an exhaustive search for the best combination of features maximizes the models' ability to correctly identify positive cases.

F1 Score

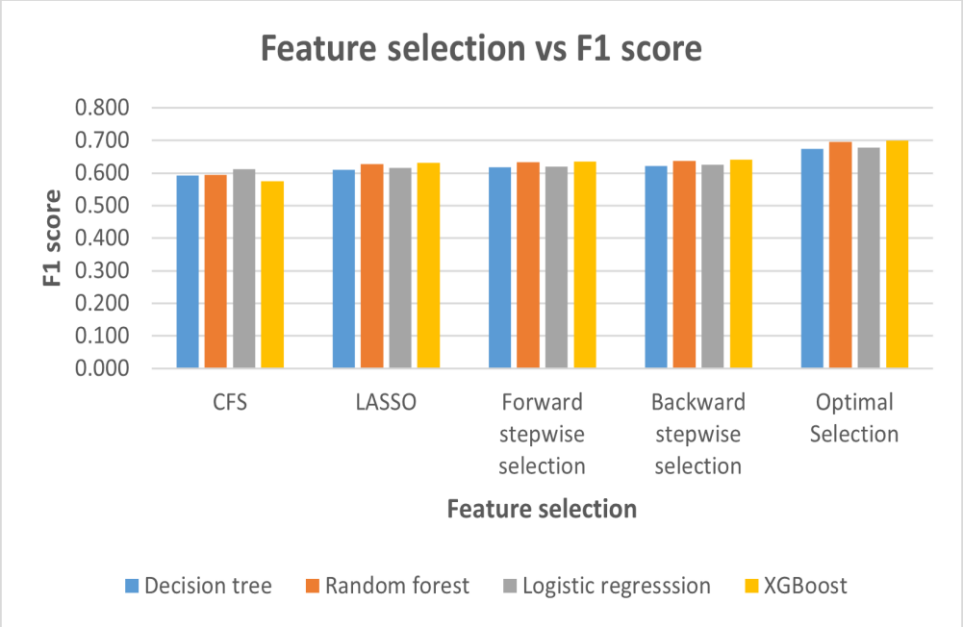


Figure 19: Feature Selection vs F1 Score

The CFS method consistently delivers the lowest performance across all models, with F1 scores ranging from 0.574 for XGBoost to 0.611 for Logistic Regression, indicating that CFS struggles to achieve an optimal balance between precision and recall, which hampers its overall effectiveness. As LASSO is introduced, there is a noticeable improvement in F1 scores, particularly for Random Forest and XGBoost, which achieve scores of 0.628 and 0.632, respectively, demonstrating that LASSO helps in selecting more relevant features. Forward and Backward Stepwise Selection further enhance the F1 scores, with Backward Stepwise Selection slightly outperforming Forward Stepwise Selection across all models, showing that removing less important features can fine-tune the model's performance. Finally, Optimal Selection yields the highest F1 scores for all models, with XGBoost reaching 0.700 and Random Forest achieving 0.695, indicating that selecting the best combination of features results in the most balanced performance between precision and recall. Even the Decision Tree, which consistently records the lowest scores, sees significant improvement with Optimal Selection, reaching an F1 score of 0.674, underscoring the importance of using advanced feature selection techniques to enhance overall model performance.

ROC AUC Score

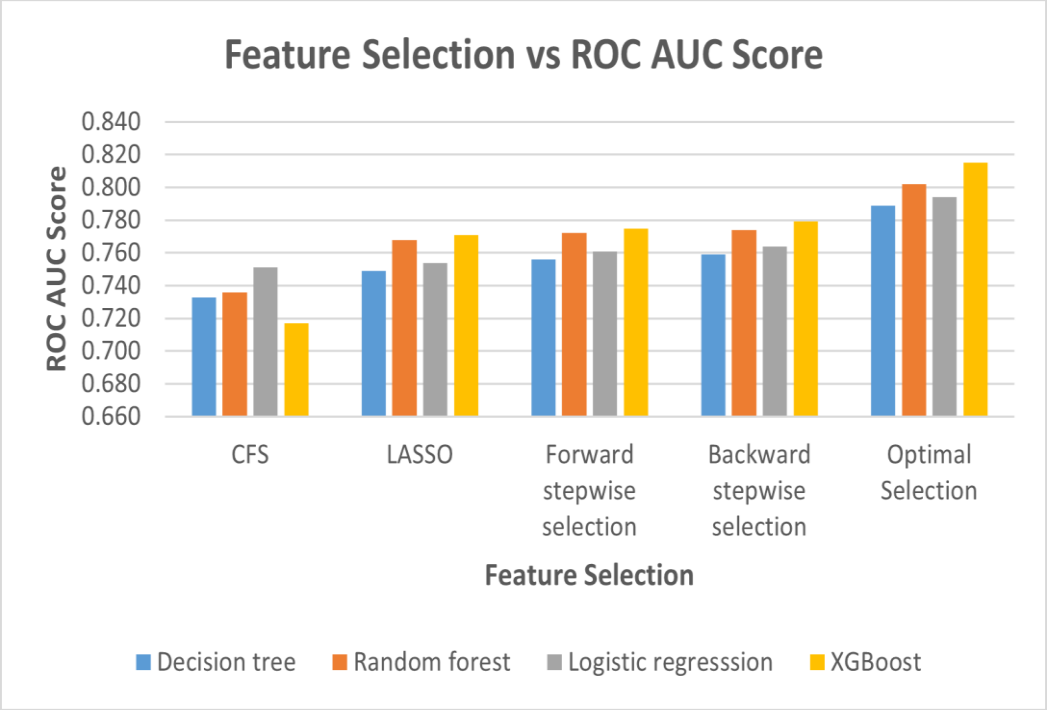


Figure 20: Feature Selection vs ROC AUC Score

The ROC AUC score shows a consistent improvement as more advanced feature selection techniques are applied. Starting with CFS, the scores are relatively low, with XGBoost achieving the lowest at 0.717 and Logistic Regression the highest at 0.751. The introduction of LASSO leads to noticeable improvements, particularly for XGBoost and Random Forest, which achieve scores of 0.771 and 0.768, respectively, reflecting LASSO's strength in enhancing model performance by reducing overfitting through regularization. Forward Stepwise Selection provides further gains in ROC AUC, especially for XGBoost, which increases to 0.775, while Random Forest and Logistic Regression also see steady improvements, reaching 0.772 and 0.761, respectively, indicating that incrementally adding relevant features enhances the models' ability to classify instances. Backward Stepwise Selection yields marginally higher scores than Forward Stepwise, with XGBoost reaching 0.779 and Logistic Regression achieving 0.764, demonstrating that selectively removing less important features can fine-tune model performance. Finally, Optimal Selection consistently delivers the highest ROC AUC scores across all models, with XGBoost reaching an impressive 0.815 and Random Forest closely following at 0.802, indicating that an exhaustive search for the best feature combinations maximizes the models' class discrimination ability.

4.3 Computational efficiency analysis of feature selection methods

Table 3: Feature Selection vs Runtime

Feature Selection Method	Average Runtime (ms)
Correlation-based Feature Selection (CFS)	46.48
LASSO	190.88
Forward Stepwise Selection	8704.37
Backward Stepwise Selection	17247.03
Optimal Selection	236216.20

The computational efficiency of feature selection methods for the churn dataset shows substantial variation. CFS stands out as the most efficient, with an average runtime of 46.48 milliseconds. LASSO is close behind, averaging 190.88 milliseconds. In contrast, forward and backward stepwise selection methods take significantly longer, with runtimes of 8704.37 and 17247.03 milliseconds, respectively. Backward stepwise selection is more resource-intensive than forward selection because it starts with all features and removes them one at a time, necessitating an initial full model fit. Lastly, optimal selection, with an exceptionally high average runtime of 236216.20 milliseconds, proves to be the most computationally demanding of all the methods.

4.4 Quality Curve Analysis for Correlation-based Feature Selection

Accuracy

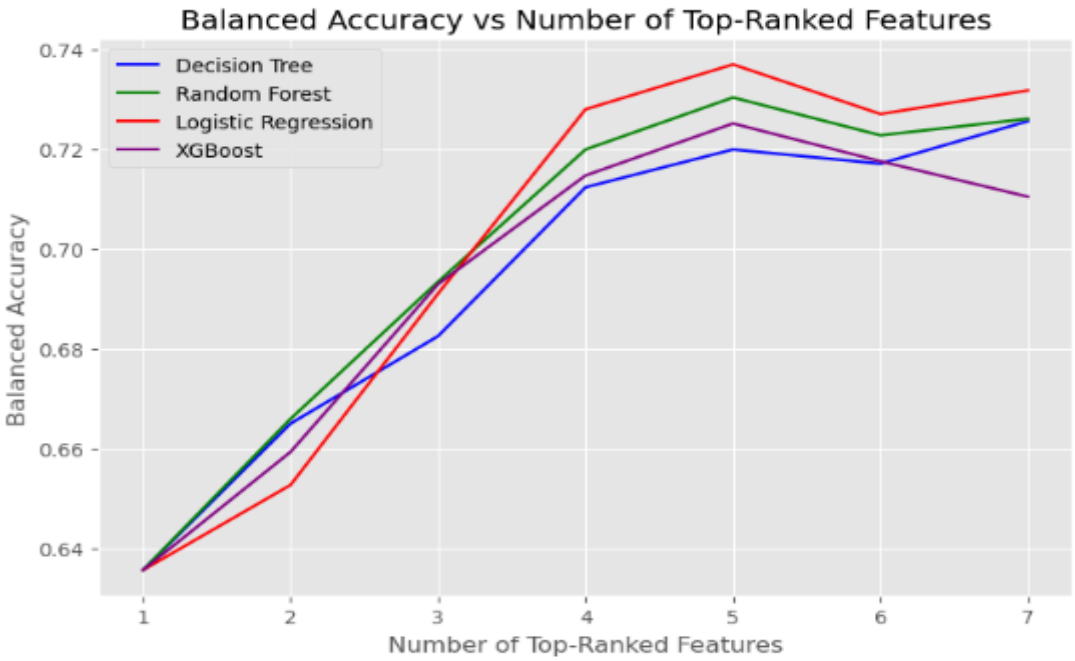


Figure 21: Number of top-ranked features vs Accuracy

Figure 20 illustrates the relationship between the number of top-ranked features and balanced accuracy across four machine learning models: Decision Tree, Random Forest, Logistic Regression, and XGBoost. As the number of top-ranked features increases from 1 to 7, all models exhibit a general upward trend in balanced accuracy, indicating that including more significant features enhances the models' ability to make accurate predictions. Initially, with only one top-ranked feature, all models show relatively low balanced accuracy, with values starting around 0.64. As more features are added, the accuracy of all models increases significantly, peaking around five top-ranked features. Logistic regression experiences the most notable improvement, reaching its highest accuracy of approximately 0.74 with five top-ranked features, while Random Forest and Decision Tree also demonstrate strong performance, with balanced accuracy stabilizing around 0.73 after six features. Beyond five features, the performance of most models either stabilizes or slightly declines, indicating a point of diminishing returns where adding more features no longer improves prediction accuracy and may even slightly degrade performance. This suggests that five top-ranked features provide an optimal balance for maximizing balanced accuracy across all models.

Precision

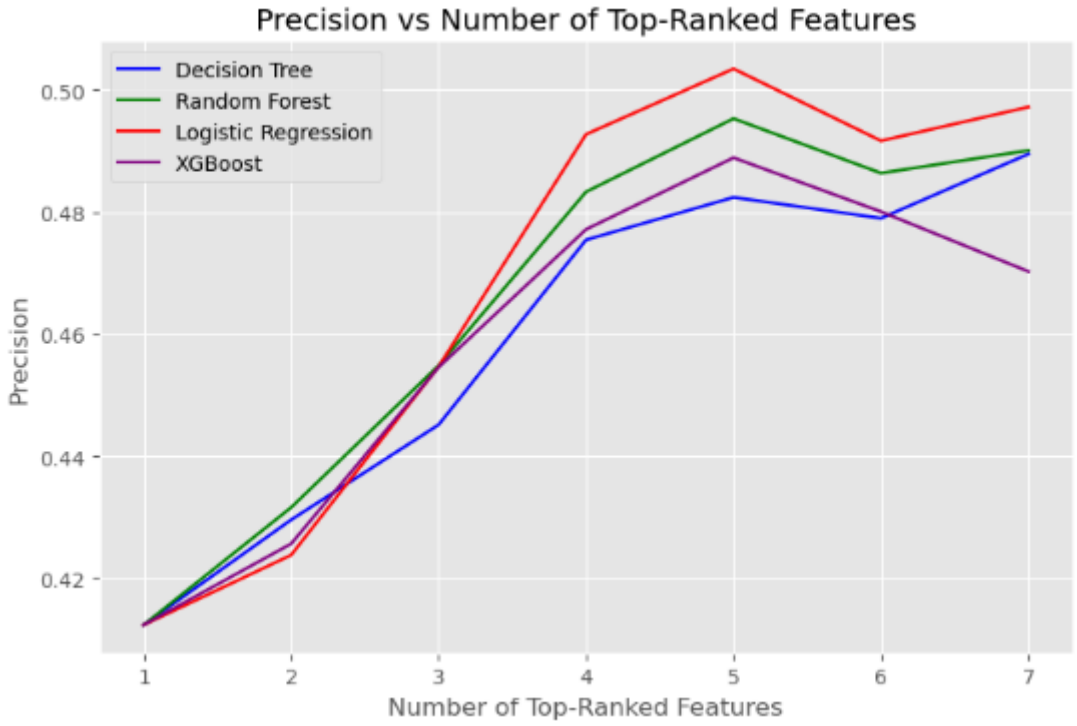


Figure 22: Number of top-ranked features vs Precision

As the number of top-ranked features increases from 1 to 7, all models experience an initial improvement in precision, with the most significant gains occurring around five features. Initially, with only one top-ranked feature, the precision values are relatively low, hovering around 0.42 for all models. As more top-ranked features are added, the precision of all models begins to rise, with Logistic Regression showing

the most pronounced increase, peaking at approximately 0.51 with five features. Random Forest and XGBoost also exhibit strong improvements, peaking at 0.50 and 0.49, respectively, at around five top-ranked features. The Decision Tree, while showing consistent improvement, lags behind the other models, peaking at around 0.48 with five features. Beyond five features, there is a slight decline in precision for most models, particularly Logistic Regression, which drops below 0.50 with six features, while others experience minor dips before stabilizing. This suggests that including more than five top-ranked features may not contribute to further improvement and could even slightly reduce precision. Overall, the graph highlights that while adding more top-ranked features initially enhances model precision, there is an optimal number—around five features—beyond which performance gains plateau or slightly decrease.

Recall

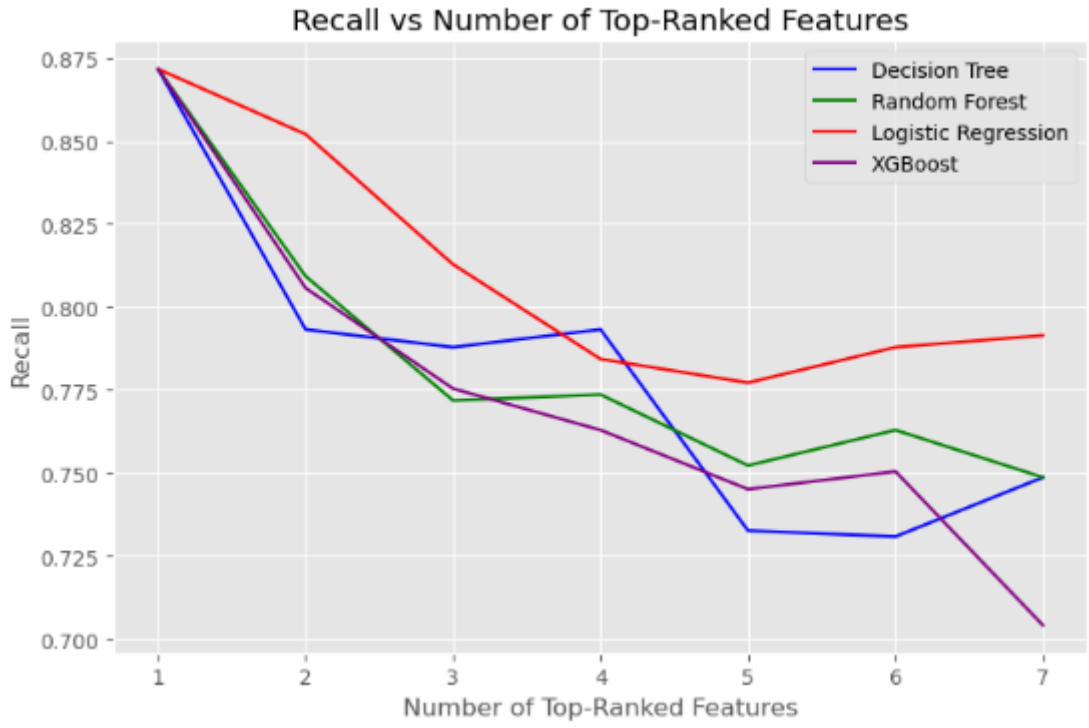


Figure 23: Number of top-ranked features vs Recall

Unlike precision and accuracy, recall exhibits a downward trend as more top-ranked features are added, indicating that including more features tends to reduce the models' ability to correctly identify positive instances. Initially, with only one top-ranked feature, recall is highest for all models, with Logistic Regression (red line) achieving a maximum value of approximately 0.875, while XGBoost (purple line) and Random Forest (green line) start at around 0.85. However, as the number of features increases, recall begins to decline rapidly for all models, with XGBoost dropping significantly and stabilizing at around 0.7 with seven features, indicating that additional features may not contribute positively to identifying true positives. Logistic Regression maintains a higher recall compared to the other models but still decreases to around

0.775 after five features, with slight stabilization after adding more features. Random Forest and Decision Tree (blue line) follow a similar pattern, both starting strong but steadily declining to around 0.75, with some fluctuations beyond five features. This general decline suggests that while fewer top-ranked features might enhance recall by focusing on the most relevant variables, adding too many dilutes the models' ability to maintain high recall, potentially due to overfitting or noise introduced by less important features. Overall, the graph highlights that while initial top-ranked features positively impact recall, there is a clear threshold beyond which adding more features diminishes performance across all models.

F1 Score

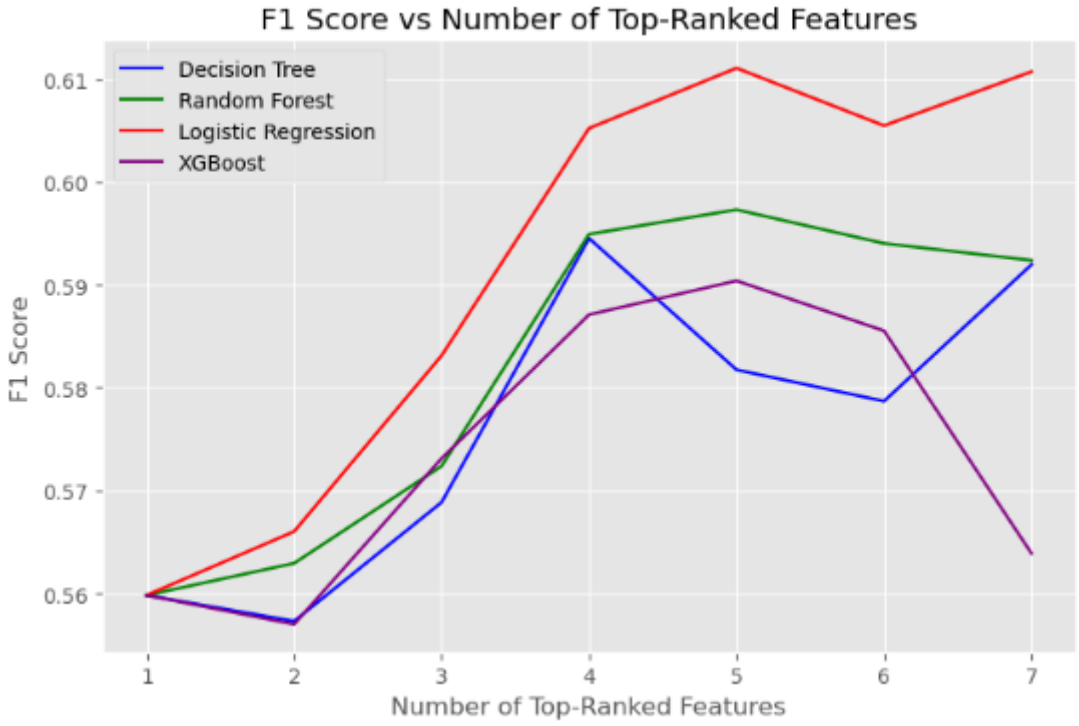


Figure 24: Number of top-ranked features vs F1 Score

A general upward trend in F1 score is observed across all models as the number of top-ranked features grows, until about four or five features, beyond which performance starts to vary. Initially, when presented with only one highly ranked feature, all models exhibit somewhat modest F1 scores of approximately 0.56. As the number of characteristics increases, Logistic Regression (red line) demonstrates the most notable enhancement, reaching its highest point at an F1 score of nearly 0.61 when five features are included. Random Forest (green line) exhibits consistent improvement, reaching its highest point around five features with an F1 score of around 0.60. XGBoost (purple line) and Decision Tree (blue line) also demonstrate a similar trend, but their progressions are less significant, as both models stabilise or decrease after four or five features. Following their peaks, both Logistic Regression and Random Forest demonstrate modest decreases, while XGBoost experiences a more significant downward trend as additional features are added. The prevailing

pattern indicates that incorporating four to five highly ranked variables yields an ideal equilibrium for enhancing the F1 score. Excessive addition of features may bring spurious or less pertinent data, therefore diminishing the models' capacity to successfully balance precision and recall.

ROC AUC Score

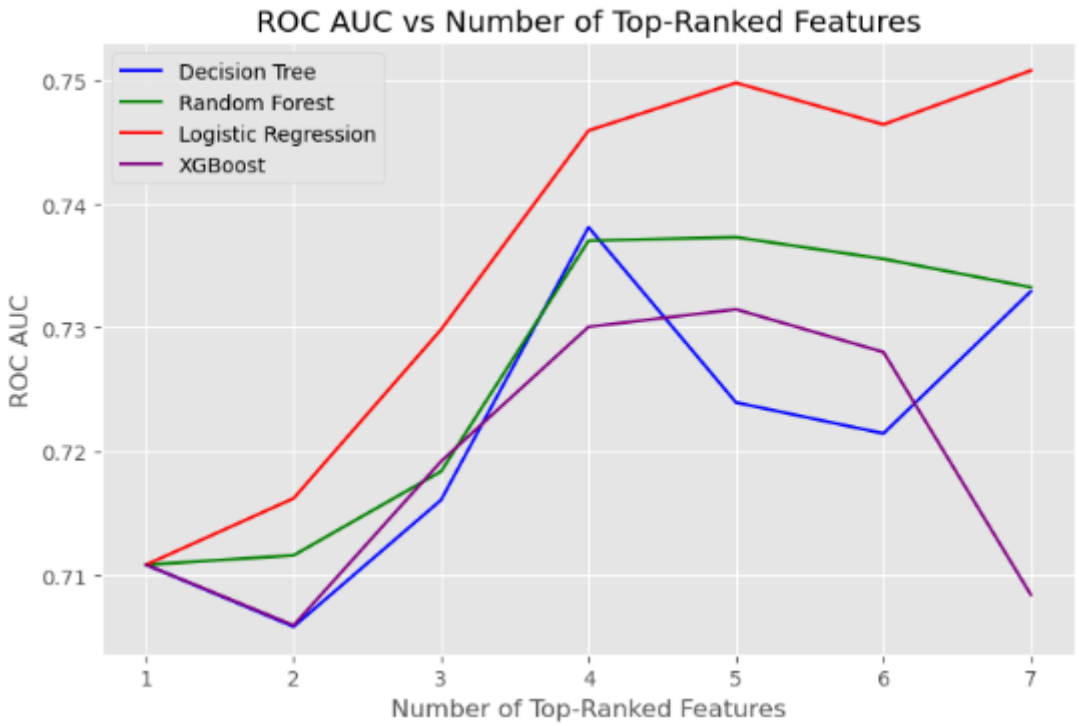


Figure 25: Number of top-ranked features vs ROC AUC Score

At first, all models demonstrate consistently low ROC AUC scores, hovering around 0.71, when using a single top-ranked feature. However, the ROC AUC values increase when additional features are included, with Logistic Regression (red line) demonstrating the most significant enhancement, reaching a peak of roughly 0.75 when five top-ranked features are adopted. The Random Forest model (green line) likewise exhibits a consistent upward trend, reaching its maximum value of approximately 0.74 with four to five features, and thereafter stabilising. XGBoost reaches its highest point at five features and then decreases, whereas Decision Tree has a significant rise up to four features, followed by some volatility. Incorporating four to five top-ranked features consistently produces the greatest ROC AUC scores across models, indicating that this range is the most effective for feature selection. Moving beyond this threshold, the inclusion of additional characteristics may result in a deterioration of performance, especially for XGBoost, which undergoes a significant decrease after five features. These findings indicate that meticulous choice of prominent characteristics is essential for optimising a model's capacity to distinguish between different categories.

5 RESULTS AND DISCUSSIONS

5.1 Overview

This chapter presents a comprehensive discussion of the analysis results in chapter 4 and thoroughly addresses each research question, with detailed explanations and insights.

5.2 Addressing the Research Questions

1. **Is feature selection based on correlations to the target variable as effective as traditional methods in terms of prediction quality?**

   Based on the analysis results presented in Chapter 4 for the churn dataset, it is evident that feature selection based on correlations to the target variable, as implemented by CFS, is less effective in terms of prediction quality for telecom churn prediction compared to traditional methods like LASSO, Forward Stepwise Selection, Backward Stepwise Selection, and Optimal Selection. CFS consistently delivers the lowest scores across all evaluation metrics—accuracy, precision, recall, F1 score, and ROC AUC—indicating that relying solely on correlations between features and the target variable is insufficient for capturing the complex relationships needed for high-quality churn prediction. For instance, CFS produces accuracy scores ranging from 0.718 to 0.732 across models, which are significantly lower than those achieved by traditional methods, such as LASSO, which yields accuracy scores between 0.739 and 0.769, and Optimal Selection, which produces the highest scores, reaching 0.839 with XGBoost. Similarly, in terms of precision, CFS performs poorly, with values between 0.479 and 0.497, while traditional methods like Optimal Selection significantly outperform it, achieving precision values as high as 0.624 for XGBoost. Recall and F1 scores follow a similar trend, with CFS consistently lagging behind traditional methods. For instance, CFS achieves recall values ranging from 0.715 to 0.791, while Optimal Selection pushes recall as high as 0.831. Furthermore, the F1 scores produced by CFS range from 0.574 to 0.611, whereas traditional methods like Optimal Selection lead to much higher F1 scores, reaching 0.700 and 0.695 for XGBoost and Random Forest, respectively. Finally, in terms of ROC AUC, CFS again falls short, producing values between 0.717 and 0.751, while traditional methods, particularly Optimal Selection, generate significantly higher scores, reaching 0.815 for XGBoost. Overall, these results demonstrate that feature selection methods based on correlations to the target variable are not as effective as traditional methods in terms of prediction quality, as they fail to maximize the models' ability to capture complex relationships, differentiate between classes, and balance precision and recall. Therefore, traditional methods like LASSO, Forward Stepwise Selection, Backward Stepwise Selection, and Optimal Selection are more suitable for telecom churn prediction, as they consistently deliver higher prediction quality across all metrics.

2. **Is feature selection based on correlations to the target variable as effective as traditional methods in terms of computational efficiency?**

In addressing the research question—whether feature selection based on correlations to the target variable is as effective as traditional methods in terms of computational efficiency—the answer is clearly yes in terms of raw speed. CFS is highly efficient compared to traditional methods like stepwise selection, which are orders of magnitude slower. However, the effectiveness of CFS in selecting the most informative features depends on the assumption that features with the highest correlation to the target variable are the most predictive. This assumption may not always hold, particularly in datasets with complex interactions or where multicollinearity is present, which traditional methods or regularization-based approaches like LASSO can handle better. Thus, while CFS is far more computationally efficient, it may sacrifice some accuracy or robustness in more complex scenarios, where methods like LASSO, though slightly slower, might offer a better balance between efficiency and effectiveness. Traditional stepwise methods, despite their thoroughness, are often too slow, making them less desirable in terms of computational efficiency. Optimal Selection, although theoretically ideal, is rarely feasible due to its excessive computational demands. Therefore, feature selection based on correlations to the target variable is indeed more computationally efficient than traditional methods, but the choice of method ultimately depends on the trade-offs between speed, complexity, and the specific characteristics of the dataset at hand.

3. **How does the prediction quality of models vary with the number of top-ranked features selected based on correlations?**

Based on the analysis results, the prediction quality of models varies significantly with the number of top-ranked features selected based on correlations, revealing distinct patterns across accuracy, precision, recall, F1 score, and ROC AUC score. Initially, as the number of top-ranked features increases from one to four or five, there is a general upward trend in prediction quality across all models, suggesting that the inclusion of a limited set of highly ranked features improves the models' ability to capture meaningful relationships between variables and the target outcome. For instance, accuracy consistently improves up to five features, with Logistic Regression reaching a peak accuracy of 0.74, while Random Forest and Decision Tree stabilize around 0.73. Similarly, precision exhibits its most significant gains up to five features, with Logistic Regression and Random Forest peaking at around 0.51 and 0.50, respectively, reinforcing the idea that these top-ranked features contribute effectively to the models' ability to make precise predictions. However, beyond five features, the performance of most models either stabilizes or slightly declines, indicating a point of diminishing returns where adding more features no longer enhances prediction quality and may even introduce noise, reducing the models' ability to distinguish between classes. This trend is particularly evident in recall, where all models experience a decline as more features are added, with XGBoost's recall dropping significantly to 0.7 when seven features are used, suggesting that more features dilute the model's ability to identify true positives. A similar pattern is observed in F1 score, where Logistic Regression reaches a peak of 0.61 with five features but experiences a modest decrease thereafter, while XGBoost shows a more pronounced decline. Furthermore, the ROC AUC score follows a similar trajectory,

with Logistic Regression and Random Forest reaching their highest values around five features, but XGBoost's performance deteriorates beyond this threshold, emphasizing that careful feature selection is crucial to maintaining model performance. The overall trend suggests that while adding top-ranked features based on correlations initially improves the models' predictive abilities by focusing on the most relevant variables, there is a clear threshold—typically around five features—beyond which the addition of more features may degrade performance by introducing irrelevant or redundant information.

In conclusion, the prediction quality of models generally improves as the number of top-ranked features increases, but the extent of improvement and the point at which diminishing returns occur vary across models and performance metrics. The optimal number of top-ranked features for most models appears to be around five, as performance gains level off or even decline beyond this point.

6 CONCLUSION

In this dissertation, we undertook an exploratory investigation into the effectiveness of correlation-based feature selection in comparison to traditional methods for predicting churn in the telecom industry through the application of machine learning. Our approach has been systematic and constructive, adhering to the principles outlined in the CRISP-DM framework.

By carefully preprocessing the data, we converted the raw dataset into a polished version suitable for analysis. We applied various feature selection techniques, including correlation analysis, LASSO, backward and forward stepwise selection, and best subset selection (optimal selection). Following this, we assessed the performance of four machine learning models: Random Forest, Decision Tree, Logistic Regression, and XGBoost. The models were evaluated using a range of metrics, such as accuracy, precision, recall, F1 score, and ROC-AUC score.

The findings reveal that while CFS is computationally efficient, it generally yields lower prediction quality compared to traditional methods. Specifically, models employing traditional methods like LASSO and stepwise selection demonstrated superior performance in terms of accuracy, precision, recall, F1 score, and ROC-AUC score. XGBoost consistently outperformed other models, particularly when traditional feature selection methods were employed. Overall, the study highlights the trade-offs between computational simplicity and prediction quality, suggesting that while CFS offers an efficient alternative, traditional methods remain more effective for churn prediction in the telecom industry.

Limitations

Firstly, the dataset used for analysis, while comprehensive, may not fully represent the diversity of customer behavior across different telecom companies. The results could be influenced by the specific characteristics of the dataset, such as geographic focus, customer demographics, and service offerings, which may limit the generalizability of the findings. Secondly, the study predominantly focused on evaluating feature selection methods using a limited number of machine learning models. Although models like Logistic Regression and XGBoost were used, incorporating a wider variety of algorithms could provide a more thorough understanding of the interaction between feature selection methods and model performance. Additionally, the study's reliance on secondary data implies that potential biases in the data collection process were not fully addressed, which could have impacted the results. Finally, the study's cross-sectional approach, which analyzes data at a single point in time, does not account for temporal changes in customer behavior or churn patterns, limiting the ability to generalize findings across different periods.

Future Research

Building upon the findings and limitations of this dissertation, future research can explore several avenues. Firstly, incorporating additional machine learning models, such as deep learning algorithms or ensemble methods like stacking, could provide a more holistic assessment of feature selection methods. Secondly, applying the models and feature selection techniques to datasets from different telecom companies and regions would enhance the generalizability of the results and allow for the identification of common churn predictors across diverse markets. Additionally, future studies could extend the analysis to longitudinal

data, enabling researchers to explore how customer behavior changes over time and how feature importance may evolve. This would be particularly valuable in identifying early indicators of churn, allowing for more proactive customer retention strategies. Finally, exploring hybrid feature selection methods that combine the efficiency of correlation-based approaches with the robustness of traditional techniques could yield a new class of feature selection strategies that balance computational efficiency and prediction quality, offering practical solutions for real-world applications.

7 REFERENCES

An, S. (2021) What are three approaches for variable selection and when to use which, Medium. Available at: https://medium.com/codex/what-are-three-approaches-for-variable-selection-and-when-to-use-which-54de12f32464 (Accessed: 07 August 2024).

Banerjee, C. (2023) All about feature selec, Medium. Available at: https://medium.com/@chandradip93/all-about-feature-selec-e6e88e8ccd46 (Accessed: 05 August 2024).

BlastChar (2018) Telco customer churn, Kaggle. Available at: https://www.kaggle.com/datasets/blastchar/telco-customer-churn (Accessed: 11 August 2024).

Borboudakis, G. and Tsamardinos, I., 2019. Forward-backward selection with early dropping. Journal of Machine Learning Research, 20(8), pp.1-39.

Chumbar, S. (2023) The CRISP-DM process: A comprehensive guide, Medium. Available at: https://medium.com/@shawn.chumbar/the-crisp-dm-process-a-comprehensive-guide-4d893aecb151 (Accessed: 11 August 2024).

Doshi, M. and Chaturvedi, S.K. (2014) 'Correlation based feature selection (CFS) technique to predict student perfromance', International journal of Computer Networks & Communications, 6(3), pp. 197–206. doi:10.5121/ijcnc.2014.6315.

Draper, N.R. and Smith, H., 1998. Applied regression analysis (Vol. 326). John Wiley & Sons.

Ellul, D.D.B. (2023) Exploring the depths of research design: Revealing the layers of the research onion, LinkedIn. Available at: https://www.linkedin.com/pulse/exploring-depths-research-design-revealing-layers-onion-borg-ellul/ (Accessed: 11 August 2024).

Fonti, V. and Belitser, E., 2017. Feature selection using lasso. VU Amsterdam research paper in business analytics, 30, pp.1-25.

Franco, F. (2023) Logistic regression algorithm, Medium. Available at: https://medium.com/@francescofranco_39234/logistic-regression-algorithm-6451c7928375 (Accessed: 13 August 2024).

Guyon, I. and Elisseeff, A., 2003. An introduction to variable and feature selection. Journal of machine learning research, 3(Mar), pp.1157-1182.

Hall, M.A., 1999. Correlation-based feature selection for machine learning (Doctoral dissertation, The University of Waikato).

Hall, M.A., 2000. Correlation-based feature selection of discrete and numeric class machine learning.

Hastie, T., Tibshirani, R. and Tibshirani, R.J. (2017) Extended comparisons of best subset selection, forward stepwise selection, and the lasso, arXiv.org. Available at: https://arxiv.org/abs/1707.08692 (Accessed: 07 August 2024).

Idris, A., Khan, A. and Lee, Y.S., 2013. Intelligent churn prediction in telecom: employing mRMR feature selection and RotBoost based ensemble classification. Applied intelligence, 39, pp.659-672.

Idris, A., Rizwan, M. and Khan, A., 2012. Churn prediction in telecom using Random Forest and PSO based data balancing in combination with various feature selection strategies. Computers & Electrical Engineering, 38(6), pp.1808-1819.

Jha, N. (2024) Understanding feature selection techniques in machine learning, Medium. Available at: https://medium.com/@nirajan_DataAnalyst/understanding-feature-selection-techniques-in-machine-learning-02e2642ef63e (Accessed: 05 August 2024).

Kipsang, K. (2023) Feature selection; stepwise regression (forward selection and backward elimination) with python, Medium. Available at: https://medium.com/@kelvinsang97/feature-selection-stepwise-regression-forward-selection-and-backward-elimination-with-python-d53230be995c (Accessed: 07 August 2024).

Kiptoon, D. (2023) Feature selection in machine learning, Medium. Available at: https://medium.com/@jdkiptoon/feature-selection-in-machine-learning-20417d052b80 (Accessed: 03 August 2024).

Laborda, J. and Ryoo, S. (2021) 'Feature selection in a credit scoring model', Mathematics, 9(7), p. 746. doi:10.3390/math9070746.

Loffredo, R. (2023) Construction of predictive churn model using xgboost, Medium. Available at: https://medium.com/@loffredo.ds/construction-of-predictive-churn-model-using-xgboost-899f206a52b8 (Accessed: 13 August 2024).

Lu, X. et al. (2012) 'A novel feature selection method based on CFS in cancer recognition', 2012 IEEE 6th International Conference on Systems Biology (ISB) [Preprint]. doi:10.1109/isb.2012.6314141.

Miller, A. (2002). Subset Selection in Regression. 2nd ed. New York: Chapman & Hall/CRC.

Muthukrishnan, R. and Rohini, R. (2016) 'Lasso: A feature selection technique in predictive modeling for Machine Learning', 2016 IEEE International Conference on Advances in Computer Applications (ICACA) [Preprint]. doi:10.1109/icaca.2016.7887916.

Nguyen, H., Franke, K. and Petrovic, S., 2010, February. Improving effectiveness of intrusion detection by correlation feature selection. In 2010 International conference on availability, reliability and security (pp. 17-24). IEEE.

Omila, D. (2023) Random Forest Algorithm - forecasting and predicting churn, Medium. Available at: https://medium.com/@dave.omila/random-forest-algorithm-forecasting-and-predicting-churn-c46fd886ca84 (Accessed: 13 August 2024).

RPubs - C5.0 Decision Tree Algorithm (2018) Rpubs.com. Available at: https://rpubs.com/cyobero/C50.

Sahazada, S. (2024) Correlation-based feature selection in a data science project, Medium. Available at: https://medium.com/@sariq16/correlation-based-feature-selection-in-a-data-

science-project-3ca08d2af5c6#:~:text=In%20a%20data%20science%20project%2C%20feature%20selection%20is%20essential%20to,%2C%20and%20decrease%20over%2Dfitting. (Accessed: 05 August 2024).

Saheed, Y.K. and Hambali, M.A., 2021, October. Customer churn prediction in telecom sector with machine learning and information gain filter feature selection algorithms. In 2021 International Conference on Data Analytics for Business and Industry (ICDABI) (pp. 208-213). IEEE.

Santiago, D. (2023) Balancing imbalanced data: Undersampling and oversampling techniques in Python, Medium. Available at: https://medium.com/@daniele.santiago/balancing-imbalanced-data-undersampling-and-oversampling-techniques-in-python-7c5378282290 (Accessed: 13 August 2024).

Sardana, N., Shekoohi, S., Cornett, E.M. and Kaye, A.D. (2023). Chapter 6 - Qualitative and quantitative research methods. [online] ScienceDirect. Available at: https://www.sciencedirect.com/science/article/abs/pii/B9780323988148000081.

Saunders, M., Lewis, P., Thornhill, A. and Bristow, A. (2019). 'Research Methods for Business students' Chapter 4: Understanding Research Philosophy and Approaches to Theory Development. [online] researchgate. Available at: https://www.researchgate.net/publication/330760964_Research_Methods_for_Business_Students_Chapter_4_Understanding_research_philosophy_and_approaches_to_theory_development.

Sree, K. (2023) Train test split and its importance, Medium. Available at: https://medium.com/@kavyasree42/train-test-split-and-its-importance-f2022472382d (Accessed: 13 August 2024).

Talaviya, A. (2023) CRISP-DM Framework: A foundational data mining process model, Medium. Available at: https://medium.com/@avikumart_/crisp-dm-framework-a-foundational-data-mining-process-model-86fe642da18c (Accessed: 11 August 2024).

Tanaka, K. et al. (2006) 'Stepwise feature selection by cross validation for EEG-based brain computer interface', The 2006 IEEE International Joint Conference on Neural Network Proceedings [Preprint]. doi:10.1109/ijcnn.2006.247119.

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society Series B: Statistical Methodology, 58(1), pp.267-288.

Umayaparvathi, V. and Iyakutti, K., 2017. Automated feature selection and churn prediction using deep learning models. International Research Journal of Engineering and Technology (IRJET), 4(3), pp.1846-1854.

Verma, N. (2023) Comprehensive guide to lasso regression: Feature selection, regularization, and use cases, LinkedIn. Available at: https://www.linkedin.com/pulse/comprehensive-guide-lasso-regression-feature-selection-nandini-verma-5smpf/ (Accessed: 06 August 2024).

Yulianti, Y. and Saifudin, A., 2020, July. Sequential feature selection in customer churn prediction based on Naive Bayes. In IOP conference series: materials science and engineering (Vol. 879, No. 1, p. 012090). IOP Publishing.

Zhang, Z. (2016) ‘Variable selection with stepwise and best subset approaches’, Annals of Translational Medicine, 4(7), pp. 136–136. doi:10.21037/atm.2016.03.35.

8 APPENDIX

8.1 Python code

```
# Import necessary libraries
import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('ggplot')
import pandas as pd
import numpy as np
import itertools
import time
import statsmodels.api as sma
import seaborn as sns

from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split
from sklearn.feature_selection import SelectFromModel, RFE
from sklearn.preprocessing import StandardScaler

from sklearn.metrics import precision_score, recall_score, f1_score, roc_auc_score,
accuracy_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
#!pip install xgboost
import xgboost as xgb
from xgboost import XGBClassifier
from sklearn.linear_model import Lasso, LinearRegression, LogisticRegression
from sklearn import linear_model
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import GridSearchCV, KFold
```

```
np.random.seed(42)


# Remove Warnings

import warnings

warnings.filterwarnings('ignore')


telco = pd.read_csv(r'C:/Users/ev00246/OneDrive - University of
Surrey/Dissertation/dataset/Churn.csv')

telco.head()


# EDA

# Check the various attributes of data like shape (no:of rows and cols), Column names,
datatypes

telco.shape

telco.columns.values

telco.dtypes

# SeniorCitizen is actually a categorical variable, so lets convert it to category

telco['SeniorCitizen'] = telco['SeniorCitizen'].astype('category')

# Total Charges should be numeric. Let's convert it to numerical data type

telco.TotalCharges = pd.to_numeric(telco.TotalCharges, errors='coerce')

# Check the descriptive statistics of numeric variables

telco.describe()

telco.info(verbose = True)

100*telco['Churn'].value_counts()/len(telco['Churn']) # Data is imbalanced, ratio = 73:27

telco.isna().sum() # checking for missing values

# droppinng rows with missing values

telco.dropna(how = 'any', inplace = True)

telco.info(verbose = True)

for i, predictor in enumerate(telco.drop(columns=['Churn', 'TotalCharges',
'MonthlyCharges','customerID', 'tenure'])):

    plt.figure(i)
```

```
    sns.countplot(data=telco, x=predictor, hue='Churn', palette=['green', 'darkblue'])
    plt.show()

telco.drop(columns= ['customerID'], axis=1, inplace=True)
telco['Churn'] = np.where(telco.Churn == 'Yes',1,0)
telco_dummies = pd.get_dummies(telco)
telco_dummies = telco_dummies.astype(int)
telco_dummies.head()

# churn by monthly charges
Mth = sns.kdeplot(telco_dummies.MonthlyCharges[(telco_dummies["Churn"] == 0) ],
                  color="Red", shade = True)
Mth = sns.kdeplot(telco_dummies.MonthlyCharges[(telco_dummies["Churn"] == 1) ],
                  ax =Mth, color="Green", shade= True)
Mth.legend(["No Churn","Churn"],loc='upper right')
Mth.set_ylabel('Churn Density')
Mth.set_xlabel('Monthly Charges')
Mth.set_title('churn by monthly charges')
telco_1 = telco_dummies
telco_1.head()

# Build models and evaluate

models = {
    'Decision Tree': DecisionTreeClassifier(
        random_state=42,
        max_depth=10,        # Limit the depth of the tree
        min_samples_split=10, # Minimum samples required to split an internal node
        min_samples_leaf=5    # Minimum samples required to be at a leaf node
    ),
```

```
    'Random Forest': RandomForestClassifier(
        random_state=42,
        n_estimators=100,    # Number of trees in the forest
        max_depth=10,        # Maximum depth of each tree
        min_samples_split=10, # Minimum samples required to split an internal node
        min_samples_leaf=5,   # Minimum samples required to be at a leaf node
        n_jobs=-1            # Use all available cores
    ),

    'Logistic Regression': LogisticRegression(
        random_state=42,
        max_iter=1000,       # Maximum number of iterations
        solver='liblinear',  # Solver for optimization (use 'liblinear' for smaller datasets)
        penalty='l2',        # Regularization term
        C=1.0,               # Inverse of regularization strength
        n_jobs=-1            # Use all available cores (works with 'liblinear' solver)
    ),

    'XGBoost': xgb.XGBClassifier(
        random_state=42,
        n_estimators=100,    # Number of boosting rounds
        max_depth=10,        # Maximum depth of a tree
        learning_rate=0.1,   # Step size shrinkage used to prevent overfitting
        subsample=0.8,       # Subsample ratio of the training instances
        colsample_bytree=0.8,# Subsample ratio of columns when constructing each tree
        n_jobs=-1            # Use all available cores
    )
}
```

```
# Correlation-based feature selection (CFS)

runtime_cfs_start = time.time() # starting time
correlations = telco_1.corr()['Churn'].abs()
selected_features = correlations[correlations > 0.3].index.tolist()
selected_features.remove('Churn')
runtime_cfs_end = time.time() # ending time
X_cfs = telco_1[selected_features]
runtime_cfs = (runtime_cfs_end - runtime_cfs_start) * 1000  # Runtime in milliseconds
print(f"Runtime for CFS: {runtime_cfs:.2f} ms")

X_train_cfs, X_test_cfs, y_train, y_test = train_test_split(X_cfs, telco_1['Churn'],
test_size=0.3, random_state=42)

smote_enn = SMOTE(random_state=42)
X_train_cfs_resampled, y_train_resampled = smote_enn.fit_resample(X_train_cfs,
y_train)

results_cfs = {}

for model_name, model in models.items():
    model.fit(X_train_cfs_resampled, y_train_resampled)
    y_pred = model.predict(X_test_cfs)
    results_cfs[model_name] = {
        'Precision': precision_score(y_test, y_pred),
        'Recall': recall_score(y_test, y_pred),
        'F1 Score': f1_score(y_test, y_pred),
        'ROC AUC': roc_auc_score(y_test, y_pred),
        'Balanced Accuracy': accuracy_score(y_test, y_pred)
    }
```

```
ranked_features = correlations[selected_features].sort_values(ascending=False)
print(f"Ranked Features: \n{ranked_features}")

# Display the results
for model_name, metrics in results_cfs.items():
    print(f"Results for {model_name}:")
    for metric_name, value in metrics.items():
        print(f"{metric_name}: {value:.3f}")
    print("\n")

# Quality curves

# Define the metrics and colors
metrics = ['Balanced Accuracy', 'Precision', 'Recall', 'F1 Score', 'ROC AUC']
colors = ['blue', 'green', 'red', 'purple']
n_features = len(ranked_features)

# Create subplots
fig, axes = plt.subplots(3, 2, figsize=(15, 15))

# Flatten the axes array for easy indexing
axes = axes.flatten()

# Iterate over the metrics to plot each one in a subplot
for i, metric in enumerate(metrics[:]):
    ax = axes[i]
    for model_name, model in models.items():
        metric_values = []
        for j in range(1, n_features + 1):
            top_features = ranked_features.index[:j]
```

```
            X_train_top = X_train_cfs_resampled[top_features]
            X_test_top = X_test_cfs[top_features]
            model.fit(X_train_top, y_train_resampled)
            y_pred_top = model.predict(X_test_top)
            if metric == 'Balanced Accuracy':
                score = accuracy_score(y_test, y_pred_top)
            elif metric == 'Precision':
                score = precision_score(y_test, y_pred_top)
            elif metric == 'Recall':
                score = recall_score(y_test, y_pred_top)
            elif metric == 'F1 Score':
                score = f1_score(y_test, y_pred_top)
            elif metric == 'ROC AUC':
                score = roc_auc_score(y_test, y_pred_top)
            metric_values.append(score)
        ax.plot(range(1, n_features + 1), metric_values, label=model_name,
color=colors[list(models.keys()).index(model_name)])
    ax.set_title(f"{metric} vs Number of Top-Ranked Features")
    ax.set_xlabel("Number of Top-Ranked Features")
    ax.set_ylabel(f"{metric}")
    ax.legend()
    ax.grid(True)
plt.tight_layout()
plt.show()


# LASSO Feature selection


X = telco_1.drop(columns=['Churn'])
y = telco_1['Churn']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=42, stratify=y)

params = {"alpha":np.arange(0.00001, 10, 500)}

kf=KFold(n_splits=5,shuffle=True, random_state=42)

lasso = Lasso()

lasso_cv=GridSearchCV(lasso, param_grid=params, cv=kf)

lasso_cv.fit(X, y)

print("Best Params {}".format(lasso_cv.best_params_))

names=telco_1.drop("Churn", axis=1).columns


runtime_lasso_start = time.time() # starting time


lasso1 = Lasso(alpha=0.00001)

lasso1.fit(X_train, y_train)


lasso1_coef = np.abs(lasso1.coef_)

feature_subset=np.array(names)[lasso1_coef>0.001]

print("Selected Feature Columns: {}".format(feature_subset))


runtime_lasso_end = time.time() # ending time


runtime_lasso = (runtime_lasso_end - runtime_lasso_start) * 1000  # Runtime in milliseconds

print(f"Runtime for Lasso feature selection: {runtime_lasso:.2f} ms")

X_lasso = X[list(feature_subset)]

X_train_lasso, X_test_lasso, y_train, y_test = train_test_split(X_lasso, y, test_size=0.3, random_state=42)

X_train_lasso_resampled, y_train_resampled = smote_enn.fit_resample(X_train_lasso, y_train)


# Build models and evaluate

results_lasso = {}
```

```
for model_name, model in models.items():
    model.fit(X_train_lasso_resampled, y_train_resampled)
    y_pred = model.predict(X_test_lasso)
    results_lasso[model_name] = {
        'Precision': precision_score(y_test, y_pred),
        'Recall': recall_score(y_test, y_pred),
        'F1 Score': f1_score(y_test, y_pred),
        'ROC AUC': roc_auc_score(y_test, y_pred),
        'Balanced Accuracy': accuracy_score(y_test, y_pred)
    }

for model_name, metrics in results_lasso.items():
    print(f"Results for {model_name}:\n")
    for metric_name, value in metrics.items():
        print(f"{metric_name}: {value:.3f}")
    print("\n")

# Forward stepwise selection

# Function to process a subset of features and calculate BIC
def processSubset(feature_set):
    model = sma.OLS(y, X[list(feature_set)])  # Fit the model on the selected feature set
    regr = model.fit()
    BIC = regr.bic
    return {"model": regr, "BIC": BIC, "features": feature_set}

# Function to perform forward stepwise selection based on BIC
def forward(predictors):
```

```
    remaining_predictors = [p for p in X.columns if p not in predictors]

    tic = time.time()

    results = []

    for p in remaining_predictors:
        results.append(processSubset(predictors + [p]))

    # Wrap everything up in a nice dataframe
    models = pd.DataFrame(results)

    # Choose the model with the lowest BIC
    best_model = models.loc[models['BIC'].argmin()]

    toc = time.time()
    print("Processed ", models.shape[0], "models on", len(predictors) + 1, "predictors in",
(toc - tic), "seconds.")

    return best_model

# Initialize an empty dataframe to store the models
models_fwd = pd.DataFrame(columns=["BIC", "features", "model"])

tic = time.time()
predictors = []

# Forward stepwise selection
for i in range(1, len(X.columns) + 1):
    models_fwd.loc[i] = forward(predictors)
    predictors = models_fwd.loc[i]["model"].model.exog_names
```

```
toc = time.time()
print("Total elapsed time:", (toc - tic), "seconds.")
models_fwd
list(models_fwd.iloc[10, 1])
runtime_forward = (toc-tic) * 1000
X_forward = X[list(models_fwd.iloc[10, 1])]
X_train_forward, X_test_forward, y_train, y_test = train_test_split(X_forward, y,
test_size=0.3, random_state=42)
X_train_forward_resampled, y_train_resampled =
smote_enn.fit_resample(X_train_forward, y_train)
print(f"Runtime for Forward stepwise selection: {runtime_forward:.2f} ms")
results_forward = {}


for model_name, model in models.items():
    model.fit(X_train_forward_resampled, y_train_resampled)
    y_pred = model.predict(X_test_forward)
    results_forward[model_name] = {
        'Precision': precision_score(y_test, y_pred),
        'Recall': recall_score(y_test, y_pred),
        'F1 Score': f1_score(y_test, y_pred),
        'ROC AUC': roc_auc_score(y_test, y_pred),
        'Balanced Accuracy': accuracy_score(y_test, y_pred)
    }


for model_name, metrics in results_forward.items():
    print(f"Results for {model_name}:\n")
    for metric_name, value in metrics.items():
        print(f"{metric_name}: {value:.3f}")
    print("\n")
```

```
# Backward stepwise selection

# Function to perform backward stepwise selection based on BIC
def backward(predictors):

    tic = time.time()

    results = []

    # Test all combinations of predictors where one predictor is removed
    for combo in itertools.combinations(predictors, len(predictors) - 1):
        results.append(processSubset(combo))

    models = pd.DataFrame(results)

    # Choose the model with the lowest BIC
    best_model = models.loc[models['BIC'].argmin()]

    toc = time.time()
    print("Processed ", models.shape[0], "models on", len(predictors) - 1, "predictors in",
(toc - tic), "seconds.")

    return best_model

# Initialize an empty dataframe to store the models
models_bwd = pd.DataFrame(columns=["BIC", "features", "model"], index=range(1,
len(X.columns)))

tic = time.time()
predictors = X.columns  # Start with all predictors
```

```
# Backward stepwise selection
while len(predictors) > 1:
    models_bwd.loc[len(predictors) - 1] = backward(predictors)
    predictors = models_bwd.loc[len(predictors) - 1]["model"].model.exog_names


toc = time.time()
print("Total elapsed time:", (toc - tic), "seconds.")
models_bwd
list(models_bwd.iloc[16, 1])
runtime_backward = (toc-tic) * 1000
X_backward = X[list(models_bwd.iloc[16, 1])]
X_train_backward, X_test_backward, y_train, y_test = train_test_split(X_backward, y,
test_size=0.3, random_state=42)
X_train_backward_resampled, y_train_resampled =
smote_enn.fit_resample(X_train_backward, y_train)
print(f"Runtime for Backward stepwise selection: {runtime_backward:.2f} ms")


results_backward = {}


for model_name, model in models.items():
    model.fit(X_train_backward_resampled, y_train_resampled)
    y_pred = model.predict(X_test_backward)
    results_backward[model_name] = {
        'Precision': precision_score(y_test, y_pred),
        'Recall': recall_score(y_test, y_pred),
        'F1 Score': f1_score(y_test, y_pred),
        'ROC AUC': roc_auc_score(y_test, y_pred),
        'Balanced Accuracy': accuracy_score(y_test, y_pred)
    }


for model_name, metrics in results_backward.items():
```

```
    print(f"Results for {model_name}:\n")
    for metric_name, value in metrics.items():
        print(f"{metric_name}: {value:.3f}")
    print("\n")

# Best Subset Selection

# Function to find the best model for a given number of features based on BIC
def getBest(k):
    tic = time.time()

    results = []
    # Loop through all combinations of k features
    for combo in itertools.combinations(X.columns, k):
        results.append(processSubset(combo))

    models = pd.DataFrame(results)

    # Choose the model with the lowest BIC
    best_model = models.loc[models['BIC'].argmin()]

    toc = time.time()
    print("Processed", models.shape[0], "models on", k, "predictors in", (toc - tic),
"seconds.")

    # Return the best model with BIC and the list of features used
    return best_model

# Initialize the dataframe to store the best models
models_best = pd.DataFrame(columns=["BIC", "features", "model"])
```

```
tic = time.time()


# Loop through different numbers of predictors (e.g., 1 to 3 predictors)
for i in range(1, 4):
    best_model = getBest(i)
    # Store the BIC, list of features, and model in the dataframe
    models_best.loc[i] = [best_model["BIC"], best_model["features"],
best_model["model"]]


toc = time.time()
runtime_optimal = (toc - tic) * 1000
print("Total elapsed time:", runtime_optimal, "ms.")
# Display the resulting dataframe
models_best
plt.figure(figsize=(20,10))
plt.rcParams.update({'font.size': 18, 'lines.markersize': 10})


bic = models_best.apply(lambda row: row[2].bic, axis=1)


plt.plot(bic)
plt.plot(bic.argmin(), bic.min(), "or")
plt.xlabel('# Predictors')
plt.ylabel('BIC')


X_optimal = X[["MonthlyCharges", "TotalCharges","Contract_Month-to-month"]]
X_train_optimal, X_test_optimal, y_train, y_test = train_test_split(X_optimal, y,
test_size=0.3, random_state=42)
X_train_optimal_resampled, y_train_resampled =
smote_enn.fit_resample(X_train_optimal, y_train)


results_optimal = {}
```

```
for model_name, model in models.items():
    model.fit(X_train_optimal_resampled, y_train_resampled)
    y_pred = model.predict(X_test_optimal)
    results_optimal[model_name] = {
        'Precision': precision_score(y_test, y_pred),
        'Recall': recall_score(y_test, y_pred),
        'F1 Score': f1_score(y_test, y_pred),
        'ROC AUC': roc_auc_score(y_test, y_pred),
        'Balanced Accuracy': accuracy_score(y_test, y_pred)
    }

for model_name, metrics in results_optimal.items():
    print(f"Results for {model_name}:\n")
    for metric_name, value in metrics.items():
        print(f"{metric_name}: {value:.3f}")
    print("\n")
```