2024

A Customer churn prediction and corrective action suggestion model for the telecommunications industry using predictive analytics.

Wanda, Rit – wane Kim
School of Computing and Engineering Sciences
Strathmore University

A Customer Churn Prediction and Corrective Action Suggestion Model for the Telecommunications Industry Using Predictive Analytics

By

Kim Rit-wane Wanda

071300

Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computing and Information Systems at Strathmore University

School of Computing and Engineering Sciences

Strathmore University

Nairobi, Kenya

April 2024

Declaration

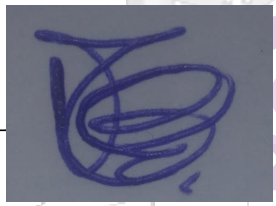
I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the dissertation contains no material previously published or written by another person except where due reference is made in the dissertation itself.

Student's Name: Kim Rit-wane Wanda

Signature:  Date: 03/04/2024

Approval

The dissertation of Kim Rit-wane Wanda was reviewed and approved (for examination) by the following:

Sign: 

Digitally signed by Dr. Kennedy Ronoh
DN: C=KE, OU=SCES, O=Strathmore University, CN=Dr. Kennedy Ronoh, E=kronoh@strathmore.edu
Reason: I am approving this document
Location: Nairobi
Date: 2024.04.04 08:11:42+03'00'
Foxit PDF Reader Version: 11.2.1

Dr Kennedy Ronoh

Lecturer, School of Computing & Engineering Sciences,

Strathmore University

Abstract

The telecommunications industry is significantly susceptible to customer churn. Customer churn leads to loss of customer base which leads to reduction in revenue, reduced profit margins, increased customer acquisition costs and loss of brand value. Mitigating the effects of customer churn has proved to be a tall order for many organizations in the telecommunications industry. Most companies employ a reactive approach to customer churn and thus do not take any corrective actions until the customer has left. This approach does not enable organizations to know and prevent potential churn before it occurs. Alternatively, some organizations employ a more proactive approach to mitigate customer churn through predictive analytics. Although this approach is more effective, it only predicts which customers will churn without recommending the appropriate corrective action.

In this dissertation, a customer churn prediction and corrective action suggestion model using predictive analytics was implemented to predict churn and suggest appropriate corrective actions. The IBM telco customer churn dataset accessed via API from the open machine learning.org website was used for this study. The dataset was subjected to pre-processing and exploratory data analysis to gain valuable insights into the data. To enhance the reliability of the developed model, an 80/20 train/test split was applied to the dataset. The training dataset was then divided into 5 folds before model fitting. Several classification algorithms; Logistic Regression, Gaussian Naive Bayes, Complement Naïve Bayes, K-NN, Random Forest and CatBoost were then fit with the training data and their performance was evaluated. Logistic Regression achieved a recall of 80% and was selected for system implementation. Logistic regression feature coefficients were then used to determine the appropriate corrective actions. A locally hosted web interface was then developed using the Python Streamlit library to enable users to feed input into the model and get churn predictions and corrective action suggestions. The developed model demonstrated ease of use and high performance and will enable telecommunication companies to accurately predict customer attrition and take appropriate corrective actions, reducing customer attrition's impact on the companies' bottom line.

Keywords: churn, machine learning, predictive analytics, telecommunications industry

Table of Contents

Declaration ii

Abstract iii

Table of Contents iv

List of Figures viii

List of Tables x

List of Abbreviations/Acronyms xi

Operational Definition of Terms xii

Acknowledgements xiii

Dedication xiv

Chapter 1: Introduction 1

1.1 Background 1

1.2 Problem Statement 2

1.3 Research Objectives 2

1.4 Research Questions 3

1.5 Justification 3

1.6 Scope 3

1.7 Limitations and Delimitations 4

Chapter 2: Literature Review 5

2.1 Overview 5

2.2 Theoretical Framework 5

2.2.1 Customer Churn 5

2.2.2 Effects of Customer Churn 6

2.2.3 Approaches to Mitigating Customer Churn 7

2.2.4 Difficulties Encountered in Customer Churn Mitigation 8

2.3 Empirical Framework 9

2.3.1 Feature Engineering....9
2.3.2 Feature Engineering Process....10
2.3.3 Feature Engineering Methods....11
2.3.4 Feature Importance Measurement....14
2.4 Predictive Analytics....16
2.4.1 Predictive Analytics in Industry....18
2.5 Classification Algorithms....19
2.5.1 Decision Trees....20
2.5.2 Logistic Regression....21
2.5.3 Naïve Bayes....22
2.5.4 K-Nearest Neighbour....24
2.5.5 Random Forest....25
2.5.6 CatBoost....26
2.6 Exploratory Data Analysis....27
2.6.1 Univariate Analysis....27
2.6.2 Bivariate Analysis....29
2.6.3 Multivariate Analysis....30
2.7 Imbalanced Data....31
2.7.1 Data-level Methods....32
2.7.2 Algorithm-level Methods....32
2.7.3 Hybrid Methods....33
2.8 Hyperparameter Tuning....33
2.6 Related Works....34
2.7 Conceptual Framework....35
Chapter 3: Research Methodology....38
3.1 Overview....38
3.2 Research Design....38

3.3 Population and Sample Size 39
3.4 Customer Churn Prediction Model Development 39
3.4.1 Data Acquisition 39
3.4.2 Data Preprocessing and Feature Engineering 39
3.4.3 Prominent Feature Identification and Selection 41
3.4.4 Model Training and Evaluation 41
3.4.5 Feature Importance Measurement and Corrective Action Suggestion. 42
3.5 System Development Methodology 42
3.6 Research Quality 43
3.7 Ethical Considerations 44
Chapter 4: System Analysis and Design 45
4.1 Introduction 45
4.2 Requirements Analysis 45
4.2.1 Functional Requirements 46
4.2.2 Non-Functional Requirements 46
4.3 System Architecture 46
4.4 Use Case Diagram 47
4.4.1 Use Case Descriptions 48
4.5 Sequence Diagrams 49
4.6 Wireframes 50
4.6.2 Home Page Wireframe 50
4.6.3 Churn Prediction Wireframe 51
Chapter 5: System Implementation and Testing 52
5.1 System Overview 52
5.1.1 Models 52
5.1.2 Web interface 52
5.2 System Implementation 54

5.2.1 Development Environment 54
5.2.2 Dataset Collection and Pre-processing 55
5.2.3 Exploratory Data Analysis and Feature Engineering 55
5.2.4 Model Training and Evaluation 61
5.2.5 Corrective Action Suggestion 62
5.3 System Testing 65
Chapter: 6 Discussions 66
6.1 Introduction 66
6.2 Exploratory Data Analysis 66
6.3 Model Evaluation Metrics 67
6.4 Corrective Actions 71
6.5 Advantages of the Tool 72
6.6 Limitations of the Tool 73
6.7 Challenges Encountered 73
Chapter 7: Conclusion and Recommendation 75
7.1 Conclusion 75
7.2 Recommendations 76
7.3 Future Work 76
References 77
Appendices 88
Appendix A: Turnitin Report 88
Appendix B: Ethical Approval 89
Appendix C: Code 90

List of Figures

Figure 2. 1: Median Churn Rates by Industry 6
Figure 2. 2 Feature Selection Techniques 11
Figure 2. 3 Application of Predictive Analytics in Various Industries 19
Figure 2. 4 Decision Trees 21
Figure 2. 5 Logistic Regression Sigmoid Curve 22
Figure 2. 6 Random Forest 26
Figure 3. 1 K-Fold Cross Validation 42
Figure 3. 2 Throwaway Prototyping Phases 43
Figure 4. 1 System Architecture 47
Figure 4. 2 Use Case Diagram 48
Figure 4. 3 Sequence Diagram 50
Figure 4. 4 Home Page Wireframe 51
Figure 4. 5 Churn Prediction Page Wireframe 51
Figure 5. 1 Home Page 53
Figure 5. 2 Churn Prediction Page 53
Figure 5. 3 Churn Prediction Page Continuation 54
Figure 5. 4 Summary Statistics 56
Figure 5. 5 Anderson-Darling Test Results 57
Figure 5. 6 Tenure Histogram 57
Figure 5. 7 Monthly Charges Histogram 57
Figure 5. 8Total Charges Histogram 58
Figure 5. 9 Churn Distribution 58
Figure 5. 10 Spearman Rank Correlation Coefficients 59
Figure 5. 11 Mann Whitney Results 59
Figure 5. 12 Chi-Square Results 60
Figure 5. 13 Cramer's V Score 61
Figure 5. 14 Churn by Payment Method 63
Figure 5. 15 Logistic Regression Sample Coefficients 64
Figure 5. 16 Corrective Action Suggestion 64
Figure 6. 1 CatBoost Metrics 68
Figure 6. 2 Complement Naive Bayes Metrics 68
Figure 6. 3 K-NN Metrics 69

Figure 6. 4 Logistic Regression Metrics 69
Figure 6. 5 Gaussian Naive Bayes Metrics 69
Figure 6. 6 Random Forest Classifier Metrics 70
Figure 6. 7 Streaming Coefficients 72
Figure 6. 8 Random Forest Grid Search Duration 74

List of Tables

Table 4. 1 Use Case Descriptions 49
Table 6. 1 Definition of Terms.....67
Table 6. 2 Model Performance Metrics.....70

List of Abbreviations/Acronyms

API	Application Programming Interface
CatBoost	Categorical Boosting
CSV	Comma Separated Values
DOF	Degree of Freedom
ECDF	Empirical Cumulative Distribution Function
EDA	Exploratory Data Analysis
ELI5	Explain Like I'm 5
GDP	Gross Domestic Product
GDPR	General Data Protection Regulation
HIPAA	Health Insurance Portability and Accountability Act of 1996
IBM	International Business Machines Corporation
IQR	Interquartile Range
LIME	Local Independent Model-agnostic Explanations
NFLT	No Free Lunch Theorem
PLC	Public Limited Company
RAD	Rapid Application Development
ROC	Receiver Operating Characteristic
SHAP	Shapley Additive exPlanations
SME	Small to Medium Enterprises
SMOTE	Synthetic Minority Oversampling Technique
SOHO	Small Office Home Office
SWQS	Symmetric Weighted Quantile Sketch Algorithm
TNR	True Negative Rate
TPR	True Positive Rate
UML	Unified Modelling Language
XGBoost	Extreme Gradient Boosting

Operational Definition of Terms

Artificial Intelligence Artificial intelligence refers to the implementation of computer systems or machines that can perform tasks that typically require human intelligence (Mohamed, 2020).

Big Data Refers to high-velocity, voluminous, and complex datasets encompassing a wide variety of data types that exceed the capabilities of traditional data processing methods and tools (Arena & Pau, 2020; Gandomi & Haider, 2015).

Customer Churn Customer churn is the phenomenon of customers discontinuing their association with a particular organization and discarding its services (Umayaparvathi & Iyakutti, 2016).

Machine Learning Machine learning is a subset of artificial intelligence involving the development of algorithms and models that enable computers to learn, analyze, and make inferences from data without being specifically programmed by humans (Alpaydin, 2020; Alzubi et al., 2018).

Predictive Analytics Predictive analytics is a branch of data analytics that employs statistical and machine learning techniques on historical data to predict the likelihood of future events (McCarthy et al., 2022; Bradlow et al., 2017).

Acknowledgements

I am sincerely grateful for the blessings of Providence that have guided me throughout this journey. I extend my heartfelt gratitude to Dr. Kennedy Ronoh, my supervisor, whose mentorship, expertise, and constructive feedback have been invaluable. Your willingness to generously invest your time and knowledge in my academic growth has been instrumental in shaping the outcome of this project. To my family, classmates, and friends, your unconditional support has been immensely valuable during the entire duration of this project. I am extremely grateful for your support, in a way that words simply cannot fully convey.

Dedication

I dedicate this dissertation project to my dad Wanda Dume, mum Emma, and sisters Wendy and Amelia. I am extremely grateful for your unconditional support throughout this study.

Chapter 1: Introduction

1.1 Background

The telecommunications industry represents a significant percentage of global GDP. The industry is highly competitive due to the existence of several national and multinational companies offering varied services (Meena & Geng, 2022). Additionally, advancements in technology have led to more effective customer service further increasing the competition amongst industry players. These factors have led to market saturation making it difficult for telecommunication companies to acquire new customers (Frost, 2023). Therefore, for telecommunication companies to remain profitable, they must maintain their existing customers by preventing customer churn.

Customer churn also referred to as customer attrition, defection, or turnover, refers to the number of customers paying for a particular product or service who fail to become repeat customers. Customer churn in the telecommunications industry is the process of pre-paid or post-paid subscribers switching from one telecommunications company to another (Fujo & Khder, 2022). Customer churn can either be voluntary or involuntary (Johny & Mathai, 2017). Voluntary churn occurs when a customer no longer purchases the products of a particular provider while involuntary churn occurs when a seller discontinues a business relationship with a customer (Park & Ahn, 2022; Johny & Mathai, 2017). Voluntary churn can be either incidental churn or deliberate churn. Deliberate churn occurs when a customer purposely ends a business relationship with a seller. Conversely, incidental churn occurs when reasons beyond the customer's control cause the business relationship to end.

Customer churn is the most significant issue affecting the wireless telecommunications industry today (Hwang et al., 2004). The average customer attrition rate in the telecommunications industry ranges from 20% to 40% (Ly & Son, 2022). This implies that telecommunication companies lose on average a third of their customers annually. In Kenya, Safaricom PLC lost eighty-nine thousand (30%) of its SOHO customers and one-hundred and ten thousand (10%) of its SME customers in the year ending March 2022 (Safaricom, 2019). According to Airtel Africa's 2019 report, its monthly churn rate was 5.7%, way above the global average for telecommunication companies (Airtel, 2019).

Customer churn leads to revenue loss which lowers profit margins negatively affecting the long-term prospects of a company. It also leads to increased costs as the costs of acquiring

a new customer are quintuple the costs of retaining an existing customer (Gallo, 2014). Presently, most companies deal with customer churn after it occurs. This approach is not ideal as it does not provide any avenue to address the churn before it happens. Additionally, some telecommunication companies apply machine learning techniques to predict customer churn. Although this approach is more efficient, it only provides information on which customers are likely to churn without proposing the most optimal corrective action. A more complete approach would be using models that can predict churners while also suggesting the most appropriate course of corrective action.

1.2 Problem Statement

Heightened competition in the telecommunications industry has led to increased difficulty in acquiring new customers. Additionally, it has led to higher customer churn rates due to the availability of numerous product offerings. Higher churn rates lead to revenue loss while increasing customer acquisition costs hence lowering profit margins. Presently, some companies in the telecommunications sector take corrective actions to remedy customer churn after it occurs. This reactive approach is not effective as it does not prevent customer churn from happening until after it occurs. The dawn of big data analytics has led to some companies predicting customer churn before it occurs (Shirazi & Mohammadi, 2019). Although this approach is more proactive, it only predicts the churn probability but does not suggest the most optimal corrective action.

The above solutions do not address the churn issue in its entirety hence the gap addressed by this study; a model that not only predicts the likelihood of churning but also the appropriate mitigating actions to be taken. By employing the proposed model, telecommunication companies will accurately predict potential churners, and take the appropriate corrective action thus reducing revenue loss while improving their financial position.

1.3 Research Objectives

Main Objective

To develop an application based on predictive analytics that can determine the probability of customer churn in the telecommunications industry and suggest the appropriate corrective actions.

Specific Objectives

i. To investigate the existing approaches to mitigating the effects of customer churn in the telecommunications industry.

ii. To determine the most important model training features in churn prediction.
iii. To develop a model for customer attrition prediction and corrective action suggestions using classification algorithms.
iv. To evaluate the performance of the developed customer attrition prediction model.

1.4 Research Questions

i. How do telecommunication companies currently mitigate the effects of customer churn?
ii. How can the most significant model training features in churn prediction be determined?
iii. How can customer churn prediction and corrective action suggestion models be developed using classification algorithms?
iv. How can the performance of the developed customer attrition prediction model be evaluated?

1.5 Justification

The key to telecommunication companies remaining as going concerns lies in their ability to mitigate the effects of customer churn (Baker et al., 2023). The proposed model will enable telecommunication firms to predict customer churn before it occurs and take the optimal corrective action. This will in turn improve customer retention. Improved customer retention will lead to lower revenue loss hence improved profit margins, with a 5% increase in customer retention rates increasing profits by 25 to 95% (Zhao et al., 2021). Through improved customer retention, the organizational brand value also increases, thereby improving the overall financial position of the organization. Additionally, customer acquisition costs will also reduce further increasing profit margins. Higher profit margins ensure that the companies can meet their short-term and long-term obligations ensuring the economy as a whole remains healthy. Finally, the outcome of this research can be used by future researchers to improve existing techniques used in customer churn prediction further narrowing the existing gaps.

1.6 Scope

Customer churn prediction involves various stages; data acquisition, data preprocessing, variable identification, univariate analysis, multivariate analysis, missing value treatment, outlier handling, variable transformation, feature engineering, model development and training, hyperparameter tuning, model evaluation, and model deployment. This research aims to cover all the stages from data acquisition to model deployment. A customer churn prediction and corrective action suggestion model based on predictive analytics is proposed. Feature

importance measurement methods will be employed to enable the model to give the appropriate corrective actions.

1.7 Limitations and Delimitations

This research is limited by financial constraints. The costs of acquiring and working with extremely large datasets are prohibitive and thus limit the amount of data used in the research. Additionally, technological constraints also limit the size of the datasets used as working with large datasets requires a lot of computational power and storage. Moreover, since the model will be trained on data from a particular country, its application may be limited, since different countries have data with different attributes. However, the data required to develop the model is available to the researcher and will significantly improve the chances of success.

Chapter 2: Literature Review

2.1 Overview

In subsequent sections of this chapter, customer churn, its effects, mitigating actions, and the difficulties encountered in mitigating customer churn are outlined and discussed. Predictive analytics and its industrial applications are also highlighted. Additionally, various machine learning algorithms extensively used in predictive analytics are discussed, highlighting their advantages and disadvantages. Feature engineering and feature importance measurement techniques are outlined within this chapter. Previous research papers on the applications of machine learning techniques in the telecommunications industry are reviewed to identify potential areas of improvement. Lastly, a conceptual framework of the proposed solution is outlined.

2.2 Theoretical Framework

2.2.1 Customer Churn

Customer churn occurs when customers cease doing business with a particular organization (Tamaddoni et al., 2016; Celik & Osmanoglu, 2019). Customer churn is one of the most significant issues facing entities today (Amin et al., 2017; Abbasimehr et al., 2012). Customer attrition is a complex multidimensional problem that can take various forms depending on the context and industry. These forms include; contract cancellations in subscription-based companies such as SaaS, cable television, and telecommunication companies; and non-renewals in industries offering renewable services such as insurance. For retail and e-commerce platforms, churn occurs when customers stop buying products from the platforms. Lastly, for websites, customer churn manifests through decreased visits leading to fewer interactions with the organization's website. In summary, though customer churn may take different forms depending on industry and context, at its core it involves the departure of customers from an organization.

Customer churn can be grouped into voluntary churn and involuntary churn (Restum, 2021). Voluntary or active churn occurs when a customer willingly and purposefully terminates their usage of a particular entity's products. Voluntary churn is caused by several factors including; superior product offerings by competitors, changing customer needs, customer dissatisfaction, high prices, and poor customer service. In contrast, involuntary/passive churn occurs when a customer stop using the products and services of a particular organization due

to reasons beyond their control. Such reasons are; tough economic conditions, relocation, and death. Involuntary churn is more difficult to mitigate since it occurs due to reasons beyond the customers' and organizations' control.

The telecommunications industry is increasingly susceptible to customer churn. In most countries, there are several national and multinational companies offering various telecommunication products and services. Furthermore, advancements in technology and regulatory restructuring have led to market coalescing, increasing the competition for customers within the industry (PWC, 2020). Consequently, customers have a wide variety of products and services to choose from, exponentially increasing the churn risk (Gordini & Veglio, 2017). As displayed in Figure 2.1 below, the median churn rate for the telecommunications industry is thirty-one per cent. The extremely high churn rate has led to increased efforts by various industry players to devise ways to mitigate the effects of customer churn.

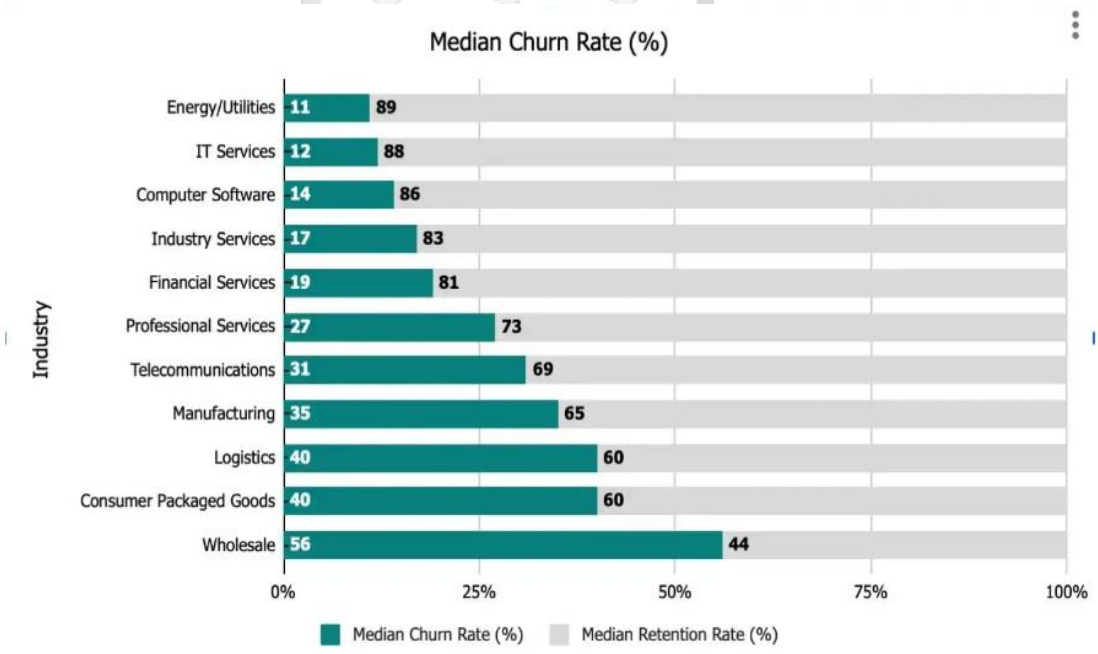


Figure 2. 1: Median Churn Rates by Industry (Tessitore, 2023)

2.2.2 Effects of Customer Churn

Understanding the effects of customer attrition is integral in developing strategies to mitigate customer attrition and improve retention. Customer churn causes revenue loss. Customer turnover leads to a decrease in the customer base, thereby reducing purchases. This leads to a reduction in sales revenue due to lower purchases and lost potential future income. Secondly, customer attrition increases the cost of customer acquisition. The costs associated with acquiring a new customer are five times higher than the cost incurred in retaining an

existing customer (Agha et al., 2021; Khan, 2013). When customer turnover occurs, organizations incur additional costs when recruiting new customers to offset the loss of old customers. Significant resources will be devoted to marketing and advertising to attract new customers. Moreover, onboarding costs will be incurred to integrate the new customers with the organization's products and services.

Customer churn reduces profit margins due to the decrease in sales revenue coupled with the increased customer acquisition costs. This in turn affects the long-term outlook of organizations preventing them from achieving their strategic goals (Mulyadi & Sinaga, 2020). Additionally, the decreased profit margins lead to reduced investment in innovation, increased competitive pressure to lower prices, implementation of austerity measures, reduced market capitalization, and increased risk of insolvency. Thirdly, customer churn makes revenue forecasting difficult as it introduces uncertainty in revenue projections hindering an organization's ability to adequately plan and budget for its resources. This in turn makes the organization susceptible to risk, both external and internal risk.

Customer attrition erodes brand value and equity through reputational damage. Customers who churn due to poor customer service, poor product quality, or other negative customer experiences may leave negative reviews painting the organization in a bad light. This in turn will dissuade future customers from conducting any business with the organization. The reputational damage occasioned by dissatisfied customers leads to loss of brand value and market capitalization (Beneke et al., 2016). In conclusion, customer churn negatively affects organizations primarily through revenue reduction and increased costs thereby increasing uncertainty that the organization may cease to be a going concern in the foreseeable future.

2.2.3 Approaches to Mitigating Customer Churn

As outlined in the previous section, customer churn has several detrimental effects on the financial position of an organization. Therefore, for entities to maintain sustained financial success, they must take corrective actions to lessen the impact of customer churn. Summarily, organizations take five primary approaches to reduce the effects of customer attrition; product improvement, engagement campaigns, one-on-one customer interactions, price rightsizing, and target acquisitions (Gold & Tzuo, 2020). The approach employed is determined by the cost, effectiveness, and the reason for customer churn.

The most straightforward method of increasing customer retention is through product improvement (Lina, 2022). Product creators enhance product features by adding functionality thus improving their utility and user enjoyment. Moreover, a product or service can be modified

to include features and functionality that would make it prohibitively expensive for a customer to shift to a different product since the features are difficult to replicate or transfer. This improves customer loyalty as the cost of churn would be too high for the customers. Secondly, organizations reduce churn by carrying out marketing campaigns through mass communication media to familiarize customers with their most popular products and services. When implementing this strategy, a more educational marketing approach is applied since customers are mostly familiar with the organization's products. By increasing their customers' knowledge of their products, they are less likely to churn.

Thirdly, organizations engage customer support and customer success representatives to facilitate smoother one-on-one customer-employee interactions thereby reducing churn. Poor customer experience is one of the leading causes of voluntary churn, and thus by improving the customer experience the risk of attrition is lessened (Kumar et al., 2014). Traditionally, organizations engaged customer support departments to assist customers when in need. Presently, organizations not only engage customer support departments but also customer success departments tasked with proactively identifying and providing assistance to customers in need, before they ask for any help. The keen attention customers receive from these departments significantly aids in reducing customer churn. Fourthly, organizations implement price rightsizing to reduce customer churn. Sales and customer support representatives reduce product prices, down-selling customers to more affordable product offerings (Wong, 2020). Although this approach may lead to a temporary reduction in sales revenue, the customer churn rates will decrease eventually leading to an increase in sales revenue while improving customer lifetime value.

Lastly, companies reduce customer churn through targeted customer acquisition. Organizations implement strategies to acquire customers from channels that yield high retention rates. This approach to churn reduction is the least effective especially for the telecommunications industry due to market saturation and increased competition. In conclusion, due to the multi-faceted nature of customer churn, organizations apply various approaches to mitigate its effects to ensure they remain going-concerns. However, all the approaches discussed in this section are reactive and do not address customer churn before it occurs, significantly reducing their efficacy.

2.2.4 Difficulties Encountered in Customer Churn Mitigation

The previous section of this dissertation outlines several approaches to customer churn mitigation. Although these measures are somewhat impactful in improving customer retention,

they are not very effective in reducing customer churn. Customer churn is a complex phenomenon that is difficult to mitigate due to the factors outlined herein. The multidimensional nature of customer churn makes it difficult to mitigate (Zhao et al., 2021) Customer churn can be caused by various reasons including price sensitivity, customer dissatisfaction, and competitor offerings, and thus it is difficult to implement a single approach to mitigate churn. Furthermore, customer churn is often a gradual process that occurs over time. Due to the time lag involved in customer churn, it is difficult to identify the exact moment churn will occur, thus hindering the optimal application of corrective measures.

In certain industries, the churn rate varies significantly throughout the year due to seasonal factors making it difficult to analyze churn patterns and take corrective actions. Additionally, the data quality and availability are sometimes poor, greatly hampering the efforts by organizations to understand customer churn thus limiting their effectiveness in addressing its effects (Vilhomaa, 2022). The cost implications of implementing some of the mitigating measures are prohibitive reducing organizational effectiveness in improving customer retention. Finally, the complexity of human behaviour leads to constantly evolving customer preferences severely limiting the effectiveness of churn mitigation measures unless they are continuously reviewed to keep up with customer behaviour.

2.3 Empirical Framework

2.3.1 Feature Engineering

Feature engineering refers to the preprocessing step of machine learning that involves the selection, extraction, manipulation, and transformation of raw data into appropriate and relevant attributes used to develop effective machine learning models (Kuhn & Johnson, 2019). A feature is an individually quantifiable attribute or characteristic of a particular instance of data used by machine learning models as the input. Features are categorized as;

i. Numerical features, which refer to quantitative and continuous variables, integers, or real numbers representing quantities or counts e.g., temperature, age, and weight. They can be directly used as inputs in machine learning models.
ii. Categorical features. These are attributes that can be classified into separate distinct categories e.g., gender and income level. Nominal categorical features consist of unordered separate distinct categories e.g., single/married while ordinal categorical features consist of separate distinct categories that can be ranked e.g., low/medium/high (Dahouda & Joe., 2021). Dichotomous categorical features

consist of two distinct categories while polytomous categorical features contain more than two distinct categories. Categorical features need to be converted into numerical features using encoding techniques before being fed into machine learning models as input.

iii. Text features. These are pieces of information represented as a sequence of characters or words. They are quantitative and thus used in machine learning models constructed for natural language processing, sentiment analysis, and text classification.

iv. Time series features. These refer to measurements of attributes taken over a specific period e.g., population growth statistics and weather patterns. The data points are measured and recorded over a sequence of time intervals and often display patterns, trends, and seasonal variations.

Feature engineering is an integral step in the development of all machine learning models. It aids in improving the performance of models by ensuring the input data is valid, relevant, and high quality. Feature engineering enables variable creation and transformation capturing vital relationships and patterns within the dataset drastically improving the model's performance. Additionally, through feature engineering, outliers, missing values, and noise are mitigated further improving the model's performance. In cases involving highly dimensional datasets, feature engineering enables attribute reduction ensuring only the relevant features are used as model inputs (Reddy et al., 2020). It helps improve generalization by reducing overfitting or underfitting. Lastly, feature engineering improves the computational efficiency of machine learning models thus reducing the number of resources required to develop the model.

2.3.2 Feature Engineering Process

Feature engineering is an iterative process that is repeated severally until the best selection of attributes is identified (Kuhn & Johnson, 2019). It consists of five main processes;

i. Feature creation. Involves developing new attributes anchored on industry-specific domain knowledge. New features can also be created after analyzing and observing the existing patterns within a dataset or by combining two or more existing features into one relevant feature.

ii. Feature transformation. This is the conversion of attributes into more fitting formats, easily input into machine learning models (Kuhn & Johnson, 2019). It is achieved through normalization, encoding, scaling, and transformation.

iii. Feature extraction involves creating new relevant attributes from the existing attributes through feature combination, feature aggregation, feature transformation, and dimensionality reduction.

iv. Feature selection is the choosing of a subset of relevant attributes to be used as the input in machine learning models (Cai et al., 2018). It helps reduce the number of input variables by selecting those with the highest correlation with the output variable. Feature selection methods include; the filter method, wrapper method, and embedded method. The filter method employs statistical measures to determine the correlation between the attribute and the output variable. The wrapper method uses a specific algorithm to evaluate each feature before selection while the embedded method implements feature selection as part of the model's training process.

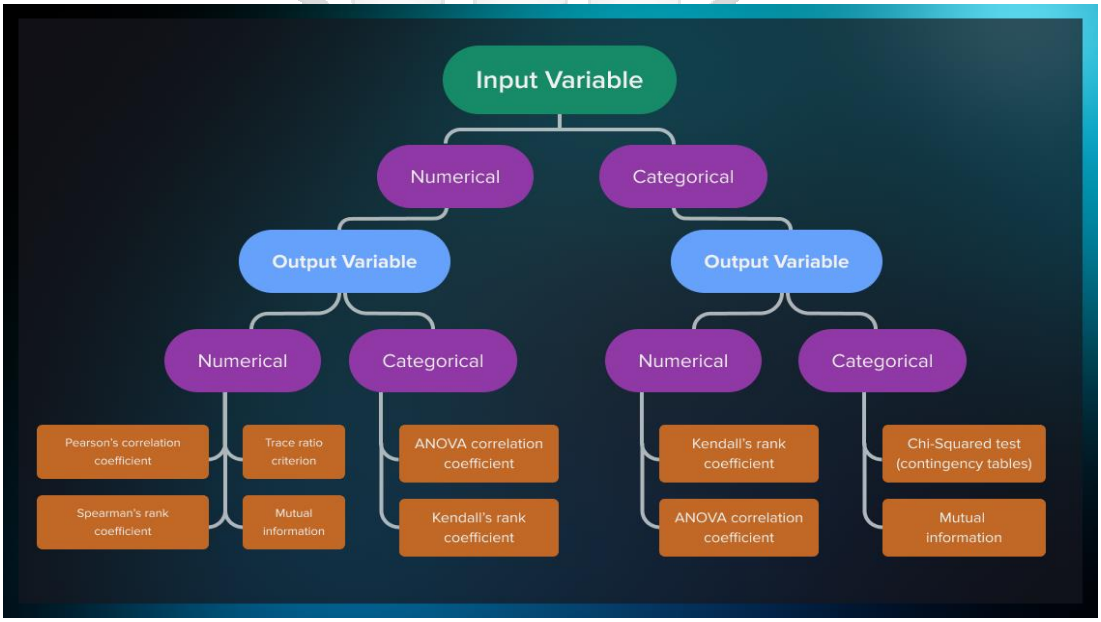


Figure 2. 2 Feature Selection Techniques (Logunova, 2022)

2.3.3 Feature Engineering Methods

There are various feature engineering methods and techniques. These are;

i. Imputation. In most machine learning applications, the datasets employed contain missing values. Training models using datasets containing a lot of missing values reduces the model's power of fit. Imputation entails the identification and replacement of missing values in a dataset (Lin & Tsai, 2020). It can be achieved through mean, mode, or median imputation or K-NN imputation. Mean, mode or median imputation replaces missing values with their respective mean, mode, or median values while K-NN imputation replaces missing values using the given

number of attributes most similar to the missing value as determined by a specific distance metric.

ii. Outlier treatment. Outliers refer to data points significantly outside the range of other data points (Kwak & Kim, 2017). Statistical measures such as Z scores and the interquartile range or visualizations such as the box plot can be used to identify outliers. The outliers may then be handled through normalization, scaling, or deletion.

iii. Feature scaling is the process of standardizing the range of input variables thus ensuring they have comparable ranges (Raschka, 2015). It ensures that attributes with larger ranges do not dominate those with smaller scales during the learning process thus enabling the machine learning algorithm to converge faster while improving its accuracy. Feature scaling algorithms include the MinMax Scaler, Standard Scaler (Z-score normalization) and Robust Scaler.

The MinMax Scaler rescales attributes to a specific range, usually between 0 and 1, the minimum value for each feature is 0 while their maximum value is 1. It subtracts the mean of each attribute from each attribute entry and divides by the range. The equation below represents the MinMax Scaler algorithm (Raschka, 2015).

Let,

xi be a single column entry,

min (x) be the mean of all entries in the column,

max(x) – min(x) is the range of x

Then,

$$\text{x scaled} = \frac{xi - \min(x)}{\max(x) - \min(x)}$$

The Standard Scaler assumes each attribute has a Gaussian distribution and centres the distribution of the respective attributes around zero, with a standard deviation of 1. Additionally, it ensures each feature has a mean of zero. The equation below represents the Standard Scaler algorithm (Raschka, 2015).

Let,

xi be a single column entry,

mean (x) be the mean of all entries in the column,

std (x) is the standard deviation of

Then,

$$x_{scaled} = \frac{xi - \text{mean}\,(x)}{std\,(x)}$$

The Robust Scaler algorithm rescales attributes using the interquartile range after eliminating the median. It computes the median for each attribute across all entries. The median value is then subtracted from each value in the attribute thereby centering the entries around the median. Subsequently, each value is divided by the interquartile range. The equation below represents the Robust Scaler algorithm (Raschka, 2015).

Let,

xi be a single column entry,

median (x) be the median of all entries in the column,

IQR is the interquartile range

Then,

$$x_{scaled} = \frac{xi - \text{median}\,(x)}{IQR}$$

iv. Encoding. This refers to the process of converting categorical variables into numerical variables. One-hot encoding develops binary columns for each class, assigning a 1 if the class is present and a 0 if it is absent. It extends the original attribute into a multidimensional matrix with each matrix dimension representing a particular state of the attribute and the matrix dimension signifying the number of states the attribute can have (Yu et al., 2022). Label encoding assigns a unique integer to each class the attribute exemplifies. Binary encoding assigns a unique binary code to each category while ordinal encoding assigns numerical variables to ordinal categorical variables.

v. Transformation. This refers to the application of logarithmic, square root, exponential, reciprocal, Box-Cox or Yeo-Johnson transformations to skewed datasets to achieve a normal distribution. In log transformation, each entry in a column is replaced with its log, and square root transformation replaces x with the square root of x. Exponential transformation replaces x with the exponent of x while reciprocal transformation replaces x with the inverse of x. The Box-Cox transformation encompasses the reciprocal, square root and logarithmic transformations. However, the Box-Cox transformation only works with positive entries (Atkinson et al., 2021) The Yeo-Johnson transformation is similar to the Box-Cox transformation but works with both positive and negative data.

2.3.4 Feature Importance Measurement

Feature importance is a machine learning concept that measures the contribution of each input in a model to the prediction or classification output thus highlighting which features significantly influence the model's decision-making (Li et al., 2017). Understanding feature importance improves the interpretability of the model enabling finetuning to increase the model's accuracy. Additionally, understanding which features contributed the most to a particular outcome influences the course of action taken, ensuring only the most appropriate actions are recommended. This in turn can be employed to construct prediction models that not only predict the outcome but also suggest the appropriate action based on how much each feature contributed to a particular outcome.

Feature importance can be measured using several techniques depending on the type of machine learning model. In decision tree-based models, feature importance can be computed by measuring how much each attribute contributes to impurity reduction or information gain at each split (Scornet, 2023). The attributes that cause significant decreases in impurity are more integral to the model's outcome. Additionally, measures like the average gain which quantifies the average improvement in model fit each time an attribute at a node is used in decision making can determine the importance of a particular feature (Gomes et al., 2019). Algorithms such as XGBoost compute the improvement in model fit resulting from a particular attribute at each node resulting in a "gain "value. This gain is then averaged for each time the attribute is used to split datasets in the tree resulting in an average gain that can be used to determine feature importance.

Moreover, tree-based algorithms may employ the split count to determine how many times a particular attribute has been used to split the dataset at the tree's nodes. The features that have been employed more frequently in splitting the datasets are taken to be more important. Feature importance may also be measured by determining how much a model's performance deteriorates when a specific feature is excluded. This is the dropout loss method and if excluding a feature severely impacts the model's performance, that particular feature is considered to be vital. In models that apply linear techniques like linear and logistic regression, feature importance may be assessed by analyzing the coefficients of each attribute, with larger coefficients implying a greater influence on the model's output (Osborne, 2014). For binary classification problems e.g. Churn/No churn, a negative coefficient implies that the respective attribute contributes more towards no churn classification. Conversely, a positive coefficient means that the respective attribute contributes more towards a churn classification. Logistic regression coefficients represent the impact of each feature on the predicted log odds of

belonging to a certain class. They capture the average effect of each feature across all instances in the dataset. They also reveal the overall relationship between features and the target variable. Therefore, as the sample size increases, the coefficients approach the true underlying relationships in the population and thus provide valid estimates of feature effects even for individual instances.

Novel approaches to feature importance measurement are through the Shapley values and LIME. SHAP values are based on game theory and measure how each attribute impacts the outcome, the importance of each attribute compared to other attributes, and the model's dependence on the interactions between the various attributes (Tallon-Ballesteros & Chen, 2020). Features with positive SHAP values positively influence the output while features with negative SHAP values negatively impact the output. Shapley value is the average of all marginal contributions to all possible conditions (Lundberg & Lee, 2017). The equation below denotes how Shap values are computed (Sundararajan & Najmi, 2020; Lundberg & Lee, 2017). Let,

S refers to a subset of features that excludes the feature the SHAP value is being computed

S ∪ i refers to the subset features in S and those in i.

s ⊆ M \ i refers to all sets S that are subsets of the full set of attributes M excluding feature i.

Then, the SHAP values are computed using;

$$\Phi_i = \sum_{S \subseteq M \setminus i} \frac{|S|!(|M|-|S|-1)!}{|M|!} [f(S \cup i) - f(S)]$$

SHAP values are additive and independently compute the contribution of each feature to the final prediction before adding them. SHAP values are robust to missing values as the SHAP values for missing or irrelevant features are zero. Additionally, SHAP values provide measures of local importance and are model agnostic. Finally, local feature importance measurement using SHAP values can easily be implemented using Python's SHAP library. Despite the above advantages, SHAP values have several drawbacks including;

i. SHAP values often require extensive computational resources.
ii. Interpretation of SHAP values requires some technical expertise, it is not as straightforward as logistic regression coefficient interpretation.
iii. In cases where the model's decision boundary is relatively flat, especially around the vicinity of prediction, small changes in the input attributes may not significantly affect the target variable leading to zero SHAP values.

Feature importance measurement can also be done through Local Interpretable Model-agnostic Explanations (LIME). LIME determines local interpretability by approximating the model's behaviour around a s specific instance in the attribute space using interpretable models (Garreau & Luxburg, 2020). Initially, LIME selects the instance for which the explanation is desired. Subsequently, LIME generates a local neighbourhood of instances similar to the original instance but with slightly different values through random perturbation of attributes. Consequently, LIME fits an interpretable model such as linear/logistic regression or decision trees to infer the local behaviour of the model. Lastly, LIME assigns weights to the perturbed instances based on their similarity to the original data and generates an explanation after analyzing the coefficients or decision rules of the interpretable model. Similar to SHAP values, LIME is model agnostic and provides local interpretable explanations. However, LIME is computationally expensive, and sample dependent i.e. generated local approximations are sensitive to the choice of samples.

2.4 Predictive Analytics

The advent of big data platforms has revolutionized how organizations analyze and work with data to achieve their strategic goals. Study areas encompassing business intelligence, data analytics, data science, and artificial intelligence have become extremely important to all organizations globally, especially those with voluminous data. Business intelligence refers to a collection of software, infrastructure, tools, and applications that ingest business data and generate user-friendly insights that inform the business' decision-making process (Bharatiya, 2023; Sharda & Turban, 2014). Business intelligence encompasses data mining, data analytics, data visualization, and data infrastructure to enable organizations to make data-driven decisions. There are four major business intelligence analyses, each covering a different use case. They are;

i. Descriptive analytics, a branch of data analytics concerned with providing a retrospective view of what has happened in an organization (Yalcin et al., 2022). Descriptive analytics provides answers to questions such as "What happened?" or "How did we perform in the past?"
ii. Diagnostic analytics, a technique that explores historical data to understand the reasons for past outcomes by identifying the underlying patterns and correlations that led to the specific outcome (Silva et al., 2021).

iii. Predictive analytics which employ statistical methods and machine learning techniques to past data to forecast future occurrences.

iv. Prescriptive Analytics, the most advanced branch of data analytics, incorporates insights from descriptive, diagnostic, and predictive analytics to recommend the most optimal course of action.

Predictive analytics encompasses the use of data, statistical techniques, and machine learning algorithms to determine the probability of upcoming events/trends from past data (McCarthy et al., 2022). Predictive analytics can be achieved through four major techniques;

i. Classification refers to supervised machine learning models that group data into predefined classes or labels (Ahuja et al., 2020). Classification techniques include; logistic regression, decision trees, Naïve Bayes, Random Forest, support vector machines, K- Nearest Neighbor, and neural networks.

ii. Regression refers to techniques that use regression equations such as the Sigmoid Function to predict a dependent variable from an independent variable (Maulud & Abdulazeez, 2020).

iii. Time series analysis, a technique that focuses on data points collected over a specific period or frequency to uncover unique patterns/trends within the temporal data (Chaurasia & Pal, 2020).

iv. Clustering, an unsupervised machine learning technique that bands data points with similar characteristics (Ahuja et al., 2020). Common clustering algorithms include; K-means clustering, mean-shift clustering, density-based spatial clustering of applications with noise, and hierarchical clustering.

Predictive analytics offers several advantages including; enabling organizations to make informed decisions by providing data-driven insights and enabling risk mitigation by identifying potential risks beforehand allowing for preemptive risk reduction measures to be initiated. In certain industries such as e-commerce and marketing, predictive analytics allows for personalized marketing campaigns and recommendations significantly improving customer retention. Additionally, through predictive analytics, organizations gain competitive advantages over their competitors by staying ahead of market trends (McCarthy et al., 2022). Although predictive analytics has clear advantages, it has several drawbacks. These are; the accuracy and effectiveness of predictive models are heavily dependent on the standard of the input, and some of the data used is sensitive raising issues of data privacy and compliance with regulations such as GDPR and HIPAA. In some instances, predictive analytics may unintentionally push biases inherent in historical data leading to prejudiced outcomes

(Favaretto et al., 2019). Lastly, some predictive models such as deep learning models are very complex and thus difficult to interpret the models' decision-making process, a major drawback in situations where interpretability is crucial.

2.4.1 Predictive Analytics in Industry

Presently, organizations collect enormous volumes of various types of data, from numerous internal and external sources. Traditional data analysis methods are not effective in analyzing such voluminous datasets. Conversely, predictive analytics augmented with artificial intelligence technologies can quickly analyze large datasets, revealing patterns and information vital for decision-making. Exponential increases in the amount in the amount of data being generated and computer processing power have made it imperative for most organizations to implement predictive analytics to remain competitive. The global predictive analytics market size is projected to be worth $34.5 billion by 2027, with its compound annual growth rate being 21.9% (Allied Market Research, 2019).

Various industries apply predictive analytics to improve operational efficiency including;

i. Human resources and personnel. Predictive analytics is used in human resources to optimize various aspects of management and decision-making. It is applied in resume screening, employee turnover prediction, performance evaluation, demand forecasting, and compensation analysis (Kakulapati et al., 2020).

ii. Healthcare. Predictive analytics is used in healthcare to ensure optimal resource allocation and utilization. It is employed in clinical care to detect early signs of disease or patient deterioration thus improving patient diagnosis and treatment.

iii. Retail and marketing. In the retail industry, predictive analytics has been leveraged for revenue projection, churn management, fraud detection, and inventory management (Rangari, 2021). Predictive marketing analytics have been implemented to aid in market segmentation, product recommendations, and personalization.

iv. Banking and Finance. Revenue projection, fraud detection, loan default risk analysis, loss projection, and customer churn in the banking and finance sector are achieved through predictive analytics.

v. Manufacturing industry. Using historical data, predictive models have been trained to accurately predict when a particular machine may malfunction, allowing for the necessary steps to be taken beforehand.

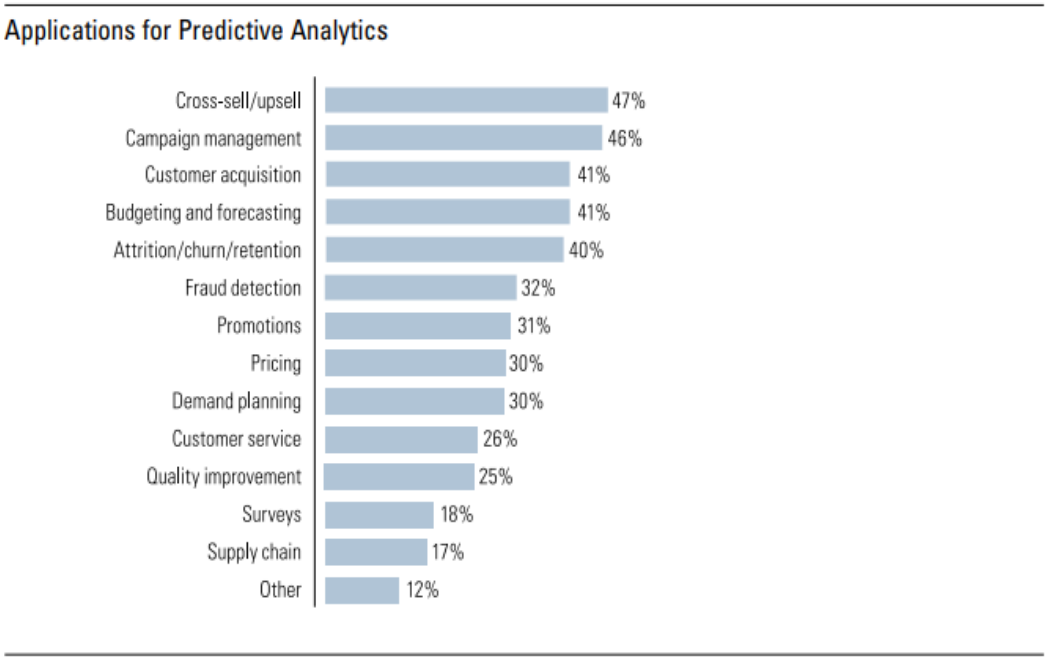


Figure 2. 3 Application of Predictive Analytics in Various Industries (Eckerson, 2007)

As discussed above, numerous industries apply predictive analytics in their operations for various reasons. A survey by Eckerson showed that predictive analytics was implemented in fourteen major areas across various industries as displayed in Figure 2.1 above. Customer churn prediction features prominently, with 40% of respondents having implemented predictive analytics to improve customer retention. In the telecommunications industry, predictive analytics is employed to analyze large datasets encompassing call details records, customer records charging information, and service usage trends to identify potential churners. Based on the insights generated by the models, the organizations may take corrective action to prevent customer attrition.

2.5 Classification Algorithms

Classification refers to a supervised machine learning technique that categorizes instances into predefined classes/labels based on their inherent features (Kesavaraj & Sukumaran 2013). Predictive models are trained on labelled datasets to learn the relationships and patterns within the data and thus make predictions using new data without labels. The objective of classification algorithms is to assign each instance the most appropriate label. Classification models are used in various tasks such as medical diagnosis, fraud detection, sentiment analysis, spam filtering, market segmentation, and churn prediction among other use cases. There are numerous classification algorithms including; logistic regression, decision trees, support vector machines, Naïve Bayes, K-Nearest Neighbor, Random Forest, and

gradient boosting. Although there are numerous classification algorithms, the No Free Lunch Theorem suggests that there is no universally superior model for solving all kinds of problems (Shwartz-Ziv & Armon, 2022). NFLT suggests that if an algorithm excels in one domain, it may not excel in a different domain. Therefore, when building a model for a particular task, it is advisable to evaluate a variety of techniques before settling on one.

2.5.1 Decision Trees

Decision trees are a supervised machine learning algorithm, that builds a tree-like model of decisions and potential consequences used for both classification and regression tasks (Kubat & Miroslav, 2021; Priyanka & Kumar, 2020). A decision tree consists of a root node, branches, and leaf nodes. Each internal node of the decision tree denotes a test on a particular feature, each branch represents the outcome of an examination of an attribute and each leaf node denotes a class label. Decision trees begin by selecting the root node using measures of attribute importance such as GINI impurity, entropy, or information gain (Daniya & Kumar, 2020). Subsequently, the algorithm divides the datasets into subsets based on the previously selected attribute. The splitting then continues recursively for each child node until a stopping criterion is met. Once the tree is constructed, the leaf nodes are assigned class labels.

Advantages of decision trees include;

i. Highly interpretable since their decision-making process is easily understood
ii. Fast learning and prediction.
iii. Can work with both categorical and numerical data without extensive preprocessing.
iv. Captures non-linear relationships between variables.

The disadvantages of decision trees are;

i. Unstable as minor changes in data lead to different tree structures.
ii. Attributes in decision trees are assumed to be locally independent.
iii. Resource-intensive when used with voluminous datasets.

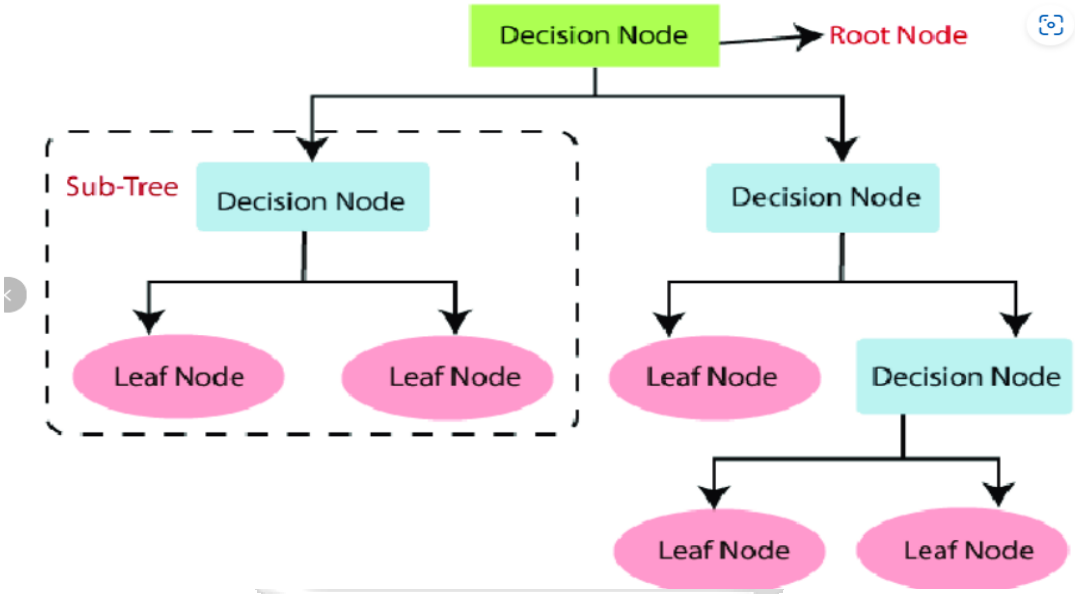


Figure 2. 4 Decision Trees (Hafeez et al., 2021)

2.5.2 Logistic Regression

Logistic regression refers to a supervised machine learning algorithm used for classification tasks, modelling the relationship between a target variable and features, by estimating the likelihood of the target variable belonging to a particular class (Bisong, 2019; Singh & Sharma, 2016). Logistic regression can be binary regression, multinomial regression, or ordinal regression. It uses the sigmoid function to determine the likelihood of the dependent variable belonging to a particular category. It is an S-shaped curve that maps any real value number to a range between 0 and 1. The equation below represents the Sigmoid Function (Dombi & Jonas, 2022).

Let,

x be the input value,

y be the output value,

b_o is the bias/intercept term, and

b_1 is the input coefficient.

Then,

$$y = \frac{e^{(bo+b1x)}}{1+e^{(bo+b1x)}}$$

The logistic regression equation, like the linear regression equation, accepts an input value to predict the output value. However, unlike linear regression where the output value is any numerical value, the output in logistic regression is a binary value. Additionally, logistic regression employs the concept of a threshold value, that defines the probability of 0 or 1. Any value above the value is assigned a 1, while all values below the value are designated 0. Logistic

regression assumes that the dependent variables are categorical, the independent variable has little to no multicollinearity and extreme outliers are absent.

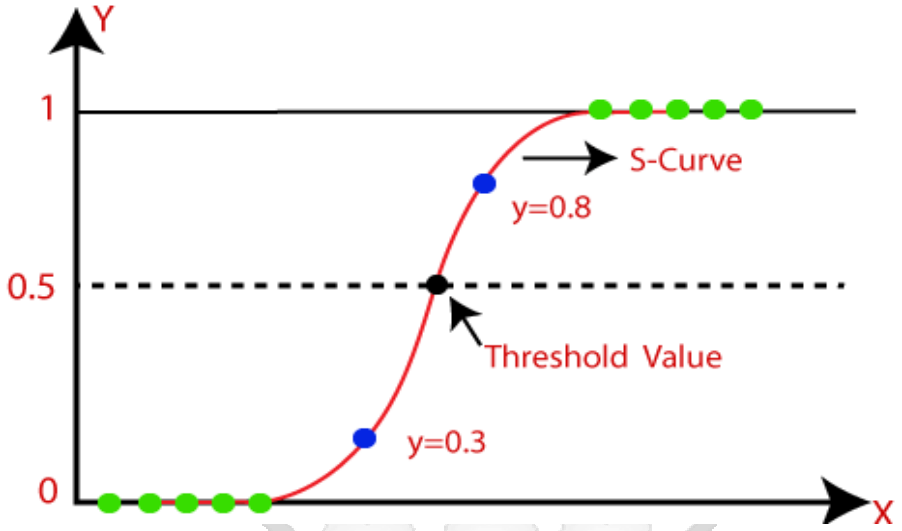


Figure 2. 5 Logistic Regression Sigmoid Curve (Javatpoint, 2021)

The advantages of logistic regression include;

i. Easy to understand, thus highly interpretable.
ii. Computationally efficient, and can be used with large datasets.
iii. Accurate, especially when the data is linearly separable.
iv. Less prone to overfitting, unless when used with highly dimensional datasets.
v. Interprets model coefficients as feature importance indicators.

Disadvantages include;

i. Ineffective when applied to non-linear problems
ii. Little to no multicollinearity for the independent variable.
iii. Assumes linearity between the dependent and independent variables.
iv. Applicable in predicting discrete functions only.
v. Susceptible to outliers.

2.5.3 Naïve Bayes

Naïve Bayes refers to a supervised probabilistic machine learning algorithm for classification tasks (Ray, 2019; Yang, 2018). The algorithm assumes feature independence, simplifying calculations. It calculates the conditional likelihood of an occurrence based on prior knowledge of conditions related to that occurrence. In doing so, it assumes that the given class label and all attributes are conditionally independent, the presence or absence of one attribute has no impact on the presence or absence of another feature. Naïve Bayes makes predictions by computing the probability of observing the attributes of each class multiplied by the prior

probability of that class, eventually choosing the class with the highest calculated probability as the predicted class. There are several variants of the Naïve Bayes algorithm; Gaussian Naïve Bayes, Complement Naïve Bayes, Multinomial Naïve Bayes, Bernoulli Naïve Bayes, and Categorical Naïve Bayes.

Gaussian Naïve Bayes works well with continuous attributes and assumes that the attributes have a normal distribution. It calculates the probability of observing a given attribute value given a class label by fitting a Gaussian distribution to each attribute within each category. The assumption of feature independence still holds for the Gaussian Naïve Bayes algorithm. Given a new instance, Gaussian Naïve Bayes computes the posterior probability of each category using the observed attribute values using Bayes' theorem. It then multiplies the class conditional probabilities with the prior probabilities of the classes and finally selects the class with the highest probability as the prediction. Complement Naïve Bayes is a version of the Naïve Bayes algorithm designed to handle imbalanced datasets (Anagaw, 2019). It focuses on identifying the least likely class, unlike the standard Naïve Bayes algorithm which predicts the class with the highest probability.

The formula for Gaussian Naïve Bayes is expressed below (Berrar, 2019).

Let,

$P_{(y|x1, x2, \ldots\ldots\ldots, xn)}$ be the posterior probability of class y given the observed attributes x

$P(y)$ be the prior probability of class y

$P(x_i|y)$ be the probability of observing attribute x_i given class y

$P(x_i)$ be the marginal probability of observing feature x_i

Then,

$$P_{(y|x1, x2, \ldots\ldots\ldots, xn)} = \frac{P(y).P\,(x1|y).P\,(x2|y).P\,(xn|y)}{P(x1).P(x2).P(xn)}$$

Naïve Bayes advantages are;

i. Simple to understand and implement.
ii. High computational efficiency.
iii. Works well with highly dimensional data.
iv. Fast
v. Effective in text classification tasks.

Disadvantages include;

i. Feature independence assumption may not hold
ii. Susceptible to noise
iii. May fail to capture complex relationships within data.

2.5.4 K-Nearest Neighbour

K-NN is a nonparametric supervised machine learning algorithm used in classification and regression and predicts an object's value or class based on the k-closest training examples (Taunk et al., 2019; Jadhav & Channe, 2016). It is especially useful in instances where the dataset has no obvious patterns or distributions. The initial step in K-NN is selecting the value of K, the number of nearest neighbours to be considered when making a prediction. A small value of K means that only the closest neighbors will be considered while a large value of K takes more neighbors into account. Subsequently, K-NN calculates the distance between a specific data point and all the other data points involved using distance metrics like Euclidian Distance (L2 Norm), Manhattan Distance (L1 Norm), Minkowski Distance, and Hamming Distance. Once the distances have been calculated, the algorithm selects the data points with the shortest distances as the K-Nearest Neighbors. During classification, the number of neighbours in each group is counted and then assigned to the group with the majority vote. The Euclidian distance equation is given below (Chomboon et al., 2015).

Let,

(x_1, y_1) be coordinates of one point,

(x_2, y_2) be coordinates of another point, and

d the distance between (x_1, y_1) and (x_2, y_2)

Then the Euclidian Distance between the two points is,

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

The Minkowski distance equation is given below (Mailagaha & Luukka, 2022).

Let,

d be the Minkowski distance between any two points

n be the number of attributes

p be the order of Minkowski distance

Then,

$$d = (\sum_{i=1}^{n} |x_{i2} - x_{i1}|^p)^{1/p}$$

The Manhattan distance equation is given below (Gao & Li, 2020).

Let,

d be the Manhattan distance between any two points

n be the number of attributes

Then,

$$d = \sum_{i=1}^{n} |x_{i2} - x_{i1}|$$

The advantages of K-NN include;

i. Easy to understand and implement
ii. Can be used for both classification and regression tasks
iii. Not susceptible to outliers
iv. Nonparametric as it does not make any underlying assumptions on data distribution.

Disadvantages of K-NN are;

i. Increased computational cost as the size of the dataset grows.
ii. The choice of K greatly influences model performance.
iii. Ineffective with highly dimensional data.
iv. The choice of distance metric greatly affects model performance.

2.5.5 Random Forest

Random Forest is an ensemble machine learning algorithm used in both classification and regression tasks. It involves multiple decision trees trained on different sets of the training data significantly improving their accuracy (Ahmad et al., 2018; Rigatti, 2017). The model assembles a collection of decision trees, each trained on a random subset of the training data and a random subset of features. Each tree then votes for a class and the class with the most votes becomes the prediction. It creates bootstrapped samples by randomly selecting data points from the original dataset with replacements.

The advantages of Random Forest are;

i. Highly accurate predictions due to the combination of decision trees.
ii. The ensemble nature of Random Forest reduces overfitting.
iii. Provides a measure of feature importance.
iv. Versatile, as it can work with structured and unstructured data.
v. Handles missing values well.
vi. Does not require encoding of the target variable
vii. Works well with imbalanced data

Disadvantages of Random Forest include;

i. Computationally expensive especially as the quantity of attributes and trees increase.
ii. The resulting trees may take up a lot of memory.

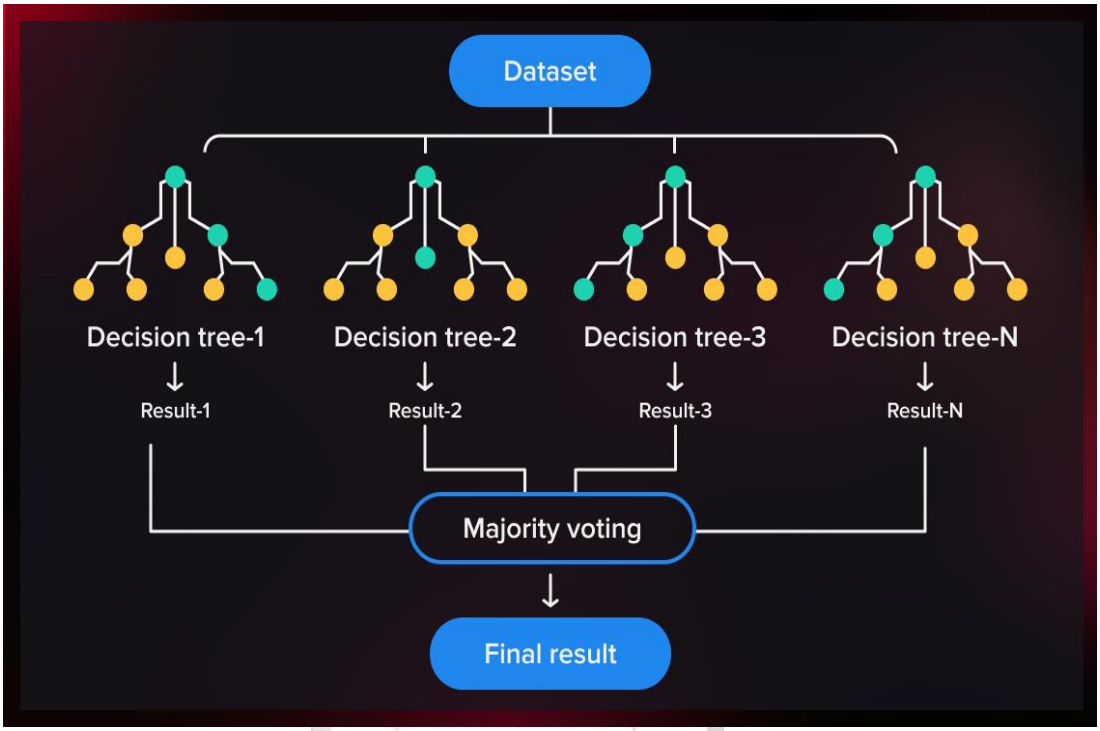


Figure 2. 6 Random Forest (Logunova, 2022)

2.5.6 CatBoost

CatBoost, categorical boosting is a type of supervised gradient boosting machine learning algorithm developed to efficiently handle categorical variables (Hancock & Khoshgoftaar, 2020; Prokhorenkova et al., 2018). It operates by combining several weak learners usually decision trees to create a more robust model. It constructs an ensemble of decision trees, with each subsequent tree learning from the errors of its predecessor. CatBoost begins the boosting process by creating an initial model e.g. one decision tree. Each instance in the dataset is assigned a weight, then the decision tree is fitted to minimize the error between the model's predictions and the actual values. Subsequently, CatBoost iteratively improves the initial decision tree by adding more trees. Each iteration focuses on learning from the errors made by previous models by computing the slope of the loss function concerning the prediction of the former model to update the current model's parameters. In constructing trees, CatBoost selects the optimal slit points for each node based on the gradient of the loss function.

CatBoost natively supports categorical features without the need for extensive preprocessing by applying an algorithm that inherently encodes the categorical features, improving the training speed. Additionally, it automatically handles missing values using the symmetric weighted quantile sketch algorithm (SWQS) (Yau et al., 2023). The advantages of CatBoost include;

i. No need to encode categorical variables as CatBoost natively handles the encoding.
ii. CatBoost is robust to missing values and overfitting.
iii. CatBoost can be optimized for speed and memory usage through parallelisation.
iv. CatBoost has built-in tools for model interpretation.
v. CatBoost handles imbalanced datasets.

Disadvantages of CatBoost are;

i. CatBoost tends to be computationally expensive, especially during the hyperparameter tuning process and model training.
ii. CatBoost performance is extremely sensitive to hyperparameters.
iii. CatBoost models are complex and thus not easily interpretable

2.6 Exploratory Data Analysis

Exploratory data analysis (EDA) refers to the initial process of analyzing a dataset to test assumptions, and hypotheses, and discover hidden patterns, relationships and associations within the dataset through the use of visualizations and summary statistics (Pearson, 2018; Martinez et al., 2017). The main objectives of EDA include; gaining a deeper understanding of the dataset by observing visualizations, distributions and summary statistics, identification of anomalies and outliers, formulation of hypotheses and research questions, assessing the quality of the data, attribute selection and engineering, data segmentation and the discovery of relationships and patterns within the data. EDA can be broadly categorized into three main categories; univariate analysis, bivariate analysis and multivariate analysis.

2.6.1 Univariate Analysis

Univariate analysis refers to a statistical method used to analyze and explore each variable in a dataset separately with a focus on understanding its distribution, central tendency and variability (Canova et al., 2017; Cleff, 2019). It is usually the first step in EDA, often used to gain insights into individual variables' properties and characteristics. The technique used during univariate analysis depends on the type of variable, whether it is continuous or categorical. For categorical variables, univariate analysis can be performed by; plotting a frequency table to understand the distribution in each category, measuring the percentage of values under each category using two metrics (count and count percentage against each category) or through visualizations such as bar plots. For continuous variables, the main objective of EDA is to understand the central tendency and the dispersion of the variable. Measures of central tendency such as the mean, mode, and median can be calculated to better

understand the variables' typical value. Additionally, measures of dispersion such as the standard deviation, variance, range, quartiles, skewness, kurtosis and the interquartile range (IQR) can be computed to determine the variables' dispersion.

Furthermore, statistical visualizations such as bar plots, histograms, boxplots, and density plots provide graphical representations of the variable's distribution and characteristics. For both numerical and categorical variables, various statistical tests can be performed to assess the relationships between the various classes within the categorical variable and the characteristics of the continuous variable. Testing for normality is an essential step in carrying out univariate analysis for continuous variables. Normality refers to the property of a continuous variable to follow a Gaussian distribution (Mishra et al., 2019). A Gaussian/normal distribution is characterized by a bell-shaped curve symmetric around the mean, with both tails being equal in length and shape and most observations clustering around the mean (Krithikadatta, 2014). Normality is an essential assumption in many statistical tests and methods such as the t-test, analysis of variance (ANOVA), linear regression and the Pearson Correlation Coefficient (Mishra et al., 2019). Normality in continuous variables can be examined using visualizations and statistical tests.

Visualizations such as histograms may be plotted and if the variable distribution resembles a bell-shaped curve, then the variable is normally distributed. Common statistical tests for normality include; the Shapiro-Wilk test, and the Anderson-Darling test. The equation for the Shapiro-Wilk test statistic W is given below (Gonzalez & Cosmes, 2019).

Let,

$x_{(i)}$ be the ordered continuous variable values

$\bar{x}$ be the variable mean

a_i be constants calculated from the covariance matrix of the ordered variable values

W be the test statistic for the Shapiro-Wilks test

Then,

$$W = \frac{\left[\sum_{i=1}^{n} a_i x_{(i)}\right]^2}{\left[\sum_{i=1}^{n} \{x_{(i)} - \bar{x}\}\right]^2}$$

The equation for the Anderson-Darling test statistic is given below (Gonzalez & Cosmes, 2019).

Let,

n be the sample size

$X_{(i)}$ be the ordered sample pairs

F ($X_{(i)}$) be the empirical cumulative distribution (ECDF) of the sample at $X_{(i)}$

Then,

$$A^2 = -n - \frac{1}{n}\sum_{i=1}^{n}(2i-1)[ln(F(X_{(i)})) + ln(1-F(X_{(n+1-i)}))]$$

2.6.2 Bivariate Analysis

Bivariate analysis is a statistical method used to analyze the relationship between two variables in a dataset. It focuses on the analysis of the association/disassociation between variables at a predetermined significance level (Denis, 2018). The nature of the bivariate analysis performed depends on the type of variables being examined. If both variables are categorical, a two-way table of count and count percentage can be created, with the rows representing the category of one variable while the columns represent the category of the other variable. Moreover, a stacked column chart, a visual representation of the two-way table can also be created. Statistical tests such as the Chi-Square, Fisher's Exact test, Cramer's V, and Kendall Tau may be performed to determine the relationship between the categorical variables. For categorical and continuous variables, statistical techniques such as the T-test, Z-test, ANOVA, and the Mann-Whitney U test may be applied to determine the relationship between the variables.

Lastly, to determine the relationship between two continuous variables, statistical methods such as the Pearson Correlation Coefficient, Spearman's Rank Correlation Coefficient and linear regression may be applied. The formula for the Chi-Square statistic is highlighted below (Verma, 2017).

Let,

O be the observed frequency in the i^{th} row and j^{th} column

E be the expected frequency in the i^{th} row and j^{th} column

r be the number of rows

c be the number of columns

X^2 be the Chi-Square statistic

DOF be the degree of freedom

Then,

$$X^2 = \sum \frac{(0-E)^2}{E}$$

$$\text{Expected frequency} = \frac{Row\ Total\ x\ Column\ Total}{Total\ (Sample\ size)}, \quad \text{DOF} = (r-1).(c-1)$$

The equation below represents the computation of Cramer's V (Akoglu, 2018).

Let,

V be Cramer's V

x^2 be the Chi-Square statistic

n be the total number of observations

r be the number of rows

c be the number of columns

Then,

$$V = \sqrt{\frac{x^2}{n \cdot min(r-1, c-1)}}$$

The equation below showcases how to compute the Kendall Tau coefficient (Bergsma & Dassios, 2014).

$$\tau = \frac{Number\ of\ concordant\ pairs - Number\ of\ discordant\ pairs}{Total\ number\ of\ pairs}$$

The equation for the Mann-Whitney U statistic is given below (Wall, 2023). The U statistic is the smaller value of U.

Let,

U be the Mann-Whitney U statistic

R_1 be the sum of the ranks of the first sample

n_1 be the number of observations in the first sample

Then,

$$U = R_1 - \frac{n_1 (n_1+1)}{2}$$

The formula for the Spearman rank correlation coefficient is highlighted below (Xiao et al., 2016)

Let,

ρ be the Spearman's rank correlation coefficient

d_i be the difference between the ranks of corresponding observations in the two variables

n be the number of pair observations

Then,

$$\rho = 1 - \frac{6\Sigma\, d_i^2}{n(n^2-1)}$$

2.6.3 Multivariate Analysis

Multivariate analysis refers to a statistical method employed to analyse data involving more than two variables. It entails examining the relationship between multiple variables to

discover complex patterns, associations, and interactions within the data (Chatfield, 2018). There are numerous multivariate analysis techniques including; multiple regression analysis, discriminant analysis, factor analysis, cluster analysis, correspondence analysis, multidimensional scaling (MDS), canonical correlation, and multivariate analysis of variance (MANOVA). Additionally, multivariate analysis may be carried out using various statistical visualizations such as the scatterplot matrix, heatmaps, 3D scatterplots, correlation matrix plots and the dendrogram.

2.7 Imbalanced Data

Imbalanced datasets refer to datasets with an unequal distribution of data points in the target variable, with one category, the majority class significantly overshadowing the other category, the minority class (Weiss, 2013). The imbalance can either be intrinsic imbalance due to naturally skewed data distribution e.g. medical diagnosis or extrinsic imbalance introduced by external factors such as data collection techniques and procedures. The degree of imbalance varies from mild to moderate then severe imbalance. Imbalanced data distributions are fairly common in numerous real-world situations including disease diagnosis, fraud detection, image recognition, and churn prediction (Herland et al., 2018; Wei et al., 2013). Dataset imbalance presents a significant challenge to most machine learning and data science tasks. If a dataset exhibits class imbalance in the training data, learners tend to overclassify the majority class while misclassifying the minority class due to the increased prior probability of the dominant class (Johnson & Khoshgoftaar, 2019). This bias towards the majority class causes the model to perform poorly when classifying the minority class.

Furthermore, imbalanced data may lead to models performing poorly with unseen data, especially when classifying the minority class i.e. poor generalization. It could also lead to the minority class being overlooked during the training process causing the model to miss important patterns and associations within the data. Lastly, imbalanced datasets may lead to misleading evaluation metrics especially if the model's performance is evaluated using traditional performance metrics such as accuracy and error rate (Johnson & Khoshgoftaar, 2019). When evaluating the performance of a model trained using imbalanced data, both accuracy and error rate are not the most optimal evaluation metrics due to the dominance of the majority class e.g. if eighty-five per cent of the target variable in a dataset consists of the positive class, the model achieves an 85% accuracy score by simply labelling all instances as the positive class.

Imbalanced datasets have negative effects on the performance of machine learning models as highlighted above. Several methods have been developed to mitigate the effects of imbalanced data on machine learning algorithms. These methods include; data-level methods, algorithm-level methods, and hybrid methods.

2.7.1 Data-level Methods

Data-level methods of mitigating the effects of imbalanced data on machine learning models incorporate resampling techniques, under-sampling and oversampling, to address the class imbalance. Random under-sampling of the majority class may be employed to discard random samples from the dominant category while random oversampling may be employed to duplicate random samples from the minority class. Although both random under-sampling and random oversampling aid in addressing the effects of class imbalance, under-sampling leads to loss of data reducing the amount of information available for model training while random oversampling increases model training time due to additional data and often leads to overfitting (Simpson, 2015). Several intelligent under-sampling and oversampling techniques have been developed to address the above concerns. The Near-Miss algorithms employ K-NN classifiers to identify majority class instances to be discarded based on their distance from the minority class. The Synthetic Minority Oversampling Technique (SMOTE) artificially generates instances of the minority class by interpolating between existing minority instances and their nearest minority neighbours (Fernandez et al., 2018). Borderline-SMOTE restricts the oversampling of the minority instances to instances near class borders while Safe-Level-SMOTE denotes safe boundaries to limit oversampling within overlapping regions (Sun et al., 2022; Daud et al., 2023).

2.7.2 Algorithm-level Methods

Algorithm-level methods unlike data-level methods, do not alter the distribution of the training data but they adjust the learning and decision process such that more emphasis is given to the underrepresented class. In cost-sensitive learning, penalties/weights are assigned to each category through a cost matrix. The cost of the minority class is raised increasing its importance thereby decreasing the probability of the model incorrectly classifying instances from the minority class. In threshold moving, the decision threshold of binary classifiers is adjusted to optimize the model's performance for specific objectives. The default threshold in classification tasks is usually 0.5. However, based on domain knowledge, precision-recall curves, ROC (Receiver Operating Characteristic) curves, and the F1 score, the classification threshold may be adjusted to a desired value. In certain machine learning algorithms such as

the Naive Bayes Classifier, class imbalance may be addressed by manually setting prior probabilities for each class based on business requirements or domain knowledge (Yang, 2018). Higher prior probabilities may be assigned to the minority classes to give them more weight during model fitting.

2.7.3 Hybrid Methods

Another approach to addressing the effects of imbalanced datasets is combining data-level methods and algorithm-level methods. Data-level methods such as SMOTE may first be applied to reduce class imbalance and noise. Subsequently, cost-sensitive learning or threshold moving may be implemented to reduce the bias towards the majority class. Additionally, several ensemble algorithms that combine cost-sensitive learning with sampling methods such as EasyEnsemble, BalanceCascade, SMOTEBoost, DataBoost-IM, JOUS-Boost, AdaC1, AdaC2, AdaC3 have been developed.

2.8 Hyperparameter Tuning

Hyperparameters refer to external configuration variables and settings set by data scientists or machine learning engineers before model training to control the model's learning process, complexity and generalization ability. Hyperparameter tuning is the process of selecting the optimal hyperparameter values to optimize the performance of a machine-learning model on unseen data (Joy et al., 2016). The main objectives of hyperparameter tuning include; improving and optimising model performance, preventing model over/underfitting, enhancing model generalization, and model customisation to meet specific user needs. Hyperparameter tuning can be achieved through manual tuning, random search, grid search, and automated hyperparameter optimisation libraries. Grid search exhaustively searches the entire predetermined hyperparameter space iteratively combining different hyperparameter values while evaluating model performance for every single iteration until it determines the most optimal combination of hyperparameter values (Shekar & Dagnew, 2019).

Random search randomly samples hyperparameter values from a predetermined space and evaluates model performance for a defined number of iterations (Hossain & Timmer, 2021). Random search offers a more computationally efficient alternative to the grid search. Manual tuning involves manually optimizing hyperparameters based on domain knowledge, prior experience or intuition while evaluating model performance until the most optimal set of hyperparameter values are determined. Although manual tuning is relatively straightforward, it is rather subjective and time-consuming. Lastly, hyperparameter tuning can be achieved by using automated hyperparameter optimization libraries such as Optuna and Hyperopt. The

advantages of hyperparameter tuning include; significantly improved model performance, allows for model customization, and ensuring that the model is more robust to over/underfitting. The main disadvantage of hyperparameter tuning is the computational cost involved especially for exhaustive search methods such as the grid search which evaluates every single combination of hyperparameters in the hyperparameter space.

2.6 Related Works

In 2020, Martin Mathu carried out research titled *Reducing Customer Churn in the Telecommunications Industry by use of Predictive Analytics.* The study used three categories of datasets; demographic data, behavioral data, and engagement data. Missing values were handled through mean imputation while outliers were identified using Z-scores and deleted. Chi-Square statistics were used to gauge feature importance for feature engineering. Subsequently, four classification models, Naïve Bayes, K-NN, neural networks, and Random Forest were developed, trained, and evaluated. The models achieved F1 scores of 0.69, 0.65, 0.63, and 0.71 respectively. This research provided a guideline on which features could be used in churn prediction. However, the main gap of this study is that it only predicts churn probability, and thus does not offer any sort of corrective action. Additionally, the dataset used was highly imbalanced, with a 49:1 non-churner to churners ratio. This imbalance was not addressed using any technique save for the application of K-fold cross-validation and using F1 scores as the key evaluation metric in place of traditional performance metrics such as accuracy and error rate.

Omonge Jevans undertook a research study *titled A Customer Segmentation Model Using Logistic Regression, a Telkom Kenya Case Study* in 2021. The data used in the study was obtained from Telkom Kenya, a major telecommunications services provider operating in Kenya. Missing values in the dataset were handled through list-wise deletion. Categorical variables were encoded to convert them into forms easily understood by the models. Scikit-Learn's MinMax scaler was used to ensure the data followed a normal distribution handling outliers. Five classifiers; logistic regression, linear discriminant analysis, K-NN, decision trees, and Naïve Bayes were developed, trained, and tested. The study achieved precision, recall, F1 and accuracy scores of 0.56, 0.54, 0.55 and 0.71 respectively. Although this study employs the same techniques used in churn prediction, it was implemented to perform customer segmentation and not churn prediction and corrective action suggestions. List-wise deletion of outliers may lead to significant data loss affecting the performance of the model. Additionally, model performance was not fully optimized as hyperparameter tuning was not carried out.

Lastly, this study implemented binomial logistic regression, even though it had three categories, which caused the precision and recall of the last category to be 0.

In 2015, Joseph Halim conducted a research, *The Causes of Churn in the Telecommunications Industry: A Case Study of Various Kenyan Service Providers*. The study aimed to investigate the behavioural patterns of churners, the economic patterns leading to customer churn, and strategies and regulations that influence customer turnover. Interviews were conducted with at least one manager from each service provider and with members of the public with handsets. This study was very informative, outlining various causes of customer churn in the telecommunications industry, vital for feature engineering. However, this study only focuses on the causes of customer churn and provides no solution in terms of customer churn prediction and corrective actions.

2.7 Conceptual Framework

The conceptual framework below links the literature review with the research problem and the research objectives. To realize the model, the following steps will be followed;

i. The initial stage of this dissertation will comprise data acquisition.

ii. This will be followed by data preprocessing, an iterative step involving variable identification, exploratory data analysis, missing value treatment, outlier handling, and categorical variable encoding.

iii. Prominent feature identification will then be done to select the desired input features. The feature identification process will be dynamic and will involve experimentation with various feature combinations.

iv. The models will then be trained using the cleaned dataset before being evaluated. Stratified K-fold cross-validation (stratified shuffle split) will be employed to train the models, with a K value of 5, with 80% of the dataset used for training and the remaining 20% for testing. SMOTE will be applied to the training dataset only before model fitting.

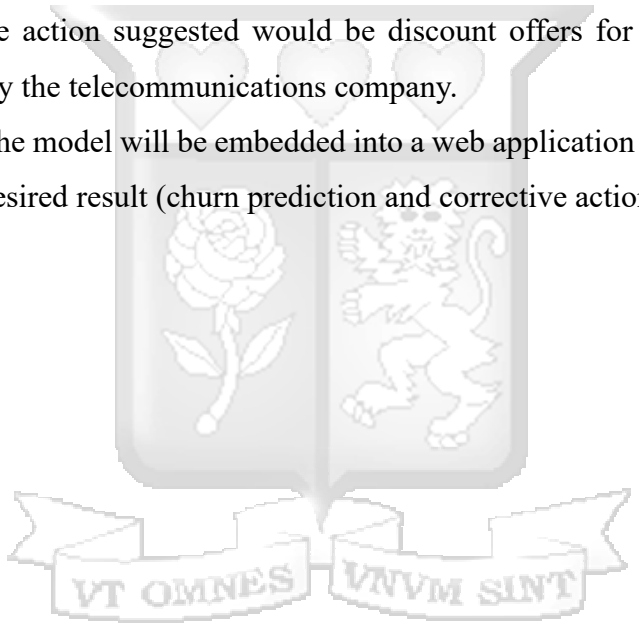
v. Consequently, the dataset will then be split into 5 folds. Various machine learning models; logistic regression, Gaussian Naïve Bayes, Complement Naïve Bayes, CatBoost, random forest and K-NN will then be trained using their default hyperparameter values.

vi. The performance of the various models will then be evaluated using the appropriate evaluation metrics. Depending on each model's performance, feature reengineering and hyperparameter tuning may be carried out to further improve

model performance. The models will then be retrained and step (f) above repeated until the optimal performance for each model is achieved. Subsequently, the models will then be evaluated and the model with the highest score will be selected.

vii. The selected model will be used for churn prediction. Feature importance measurement techniques such as SHAP values, LIME, and attribute coefficients will be computed to determine how much each input feature contributes to the outcome. The method chosen for feature importance measurement will depend on the selected model. The most significant local features for a particular outcome will then be used to determine the most appropriate corrective action e.g. if the feature that contributes the most towards the churn prediction is total charges the corrective action suggested would be discount offers for the various products offered by the telecommunications company.

viii. Finally, the model will be embedded into a web application that can be queried to get the desired result (churn prediction and corrective action suggestion).



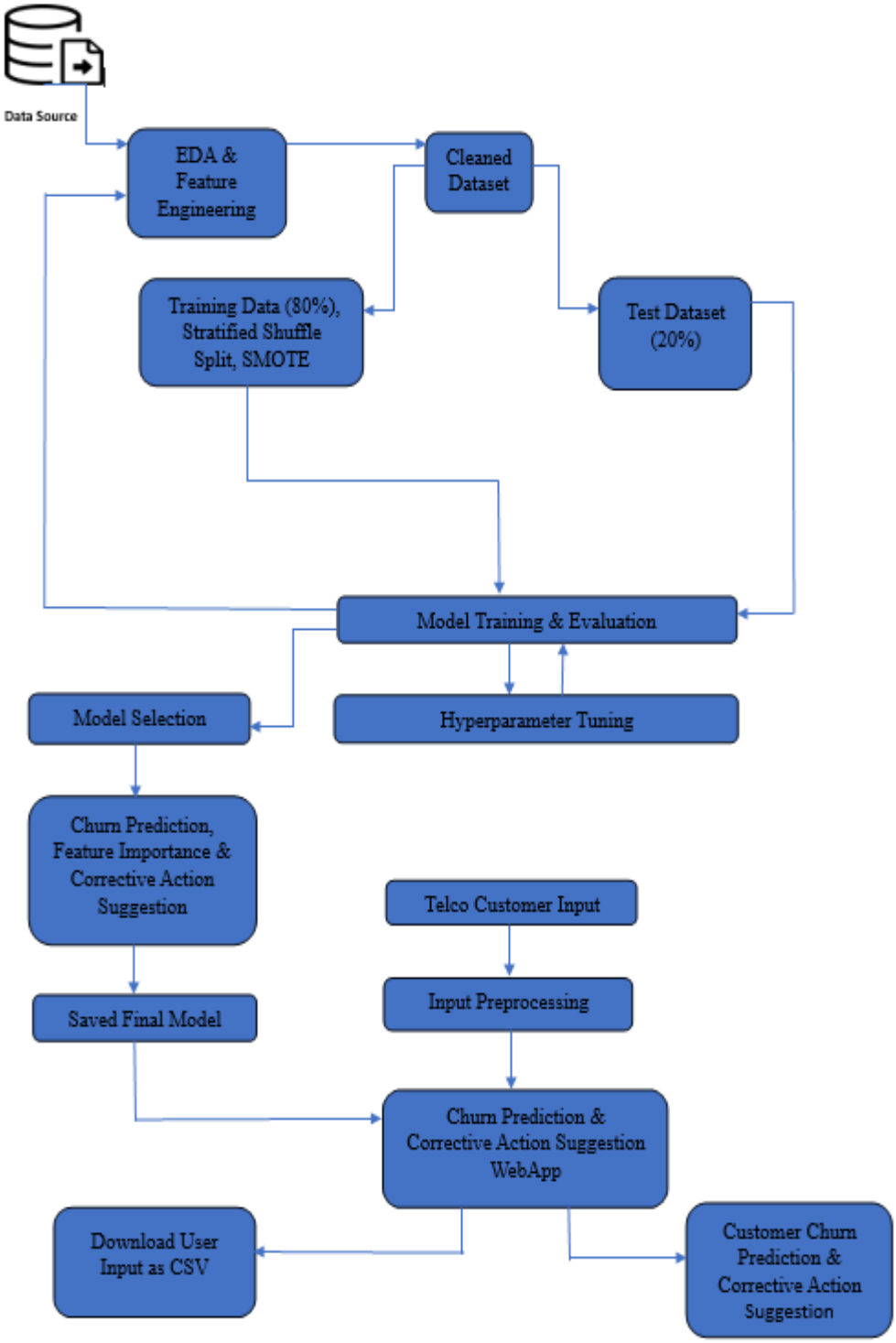


Figure 2. 7 Conceptual Framework

Chapter 3: Research Methodology

3.1 Overview

This chapter highlights the research undertaken and all the steps involved in the development of the customer churn prediction and corrective action suggestion model. Additionally, this chapter covers the system development methodology employed in implementing the web application the model will be embedded. Lastly, the research quality, ethical issues arising from this research, and the risk-benefit analysis, were addressed herein.

3.2 Research Design

This research employed a quantitative research design to develop a predictive and corrective action suggestion model for customer churn in the telecommunications industry. Quantitative research design approach involves the systematic collection and analysis of numerical data to identify patterns, test relationships, or make predictions (Fischer et al., 2014). The research was guided by the problem statement outlined in Chapter 1, the objectives and research questions highlighted in Chapter 1, and the related works reviewed in Chapter 2 of this dissertation. The IBM telco customer churn dataset served as the primary data source for this study. Firstly, exploratory data analysis was conducted to gain a deeper understanding of the dataset. This process helped identify trends, patterns, and anomalies in the data and informed the subsequent feature selection process. Following the EDA, relevant attributes were selected to serve as inputs for the machine learning models. These models were then subjected to a rigorous training and evaluation process, culminating in the selection of the best-performing model.

The final output of this research was a locally hosted web application, with the selected model embedded in the application, that accepts specific telecommunications customer data as input. The application processes the data through the embedded prediction model and outputs a prediction of whether the customer will churn. In the event of predicted churn, the application suggests appropriate corrective actions. This research design, with its focus on quantitative data analysis, enabled the acquisition and analysis of customer churn data to make accurate predictions on churn probability. By integrating this design with machine learning techniques, the study aimed to provide valuable insights that can help telecommunications companies reduce customer churn and improve customer retention strategies.

3.3 Population and Sample Size

The target population for this dissertation was the IBM telco customer churn dataset, accessed from the openml.org website via an application programming interface (API). It has 7047 entries and 21 attributes, with each row representing a customer and each column representing a customer attribute stored in a .csv format. The attributes include; Customer ID, Gender, Senior Citizen, Partner, Dependents, Tenure, Phone Service, Multiple Lines, Internet Service, Online Security, Online Backup, Device Protection, Tech Support, Streaming TV, Streaming Movies, Contract, Paperless Billing, Payment Method, Monthly Charges, Total Charges and Churn. The dataset's churners to non-churners ratio is 23:77. The dataset does not overrepresent any gender, age group, marital status or tenure (how long a particular subscriber has been with the telco). Additionally, compared to the related works reviewed in Chapter 2 of this dissertation, the dataset is particularly balanced and thus well suited for this study.

Sample refers to a collection of individuals or observations chosen from a wider group of individuals or observations, subsequently subjected to various studies, observations and statistical analyses to draw conclusions and inferences that can be applied to the global population (Hennink & Kaiser, 2022). Since the dataset is not voluminous, (7047 rows and 21 columns) the entire dataset was used. Stratified K-fold cross-validation was implemented, with a train-test split of 80/20, and a K value of 5.

3.4 Customer Churn Prediction Model Development

3.4.1 Data Acquisition

Data acquisition is the initial step of any data science or machine learning project. It involved the acquisition of the dataset that was used in the training, testing, and evaluation of the customer churn prediction models. This research used the IBM telco customer churn raw dataset to develop the models. The dataset is open source under a Creative Commons Public domain license and was accessed via an API from the openml.org website (Vanschoren, 2019).

3.4.2 Data Preprocessing and Feature Engineering

Data preprocessing involves the cleaning and transformation of raw data into well-structured datasets suitable for data mining and analytics. It improves the accuracy of machine learning models by ensuring data suitability (Garcia & Ferrera, 2015). It involves data quality assessment, data cleaning, data transformation, and data reduction. This research employed various Python libraries primarily NumPy, Pandas, Matplotlib, Seaborn, and Scipy to clean and process the data. Univariate, bivariate and multivariate analyses were carried out to gain

insights into the trends and relationships between variables in the dataset. Univariate analysis of numerical attributes involved the computation of summary statistics, identification and imputation of missing values, binning, skewness and normality testing and variable transformation.

Various statistical tests were performed to determine the relationship between and among variables in the dataset. Several statistical visualizations were plotted to determine variable trends and relationships between/among variables. Since 18/21 attributes in the dataset were categorical, one-hot encoding of model inputs was implemented. The target variable, churn, was encoded using Scikit-learn's label encoder. However, for machine learning algorithms such as the Random Forest Classifier and CatBoost, the target variable was not encoded as they inherently encoded the target variable. Numerical features varied significantly in absolute value (single digits such as tenure to five digits in total charges), thus Sci-kit learn's robust scaler was used to scale numerical features before model fitting. The equations for variance and standard deviation are highlighted below (Thurkal et al., 2019).

Let,

σ^2 be the population variance

σ be the population standard deviation

N be the number of observations in the population

x_i be the ith observation in the population

μ be the population mean

Then

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

The formula for skewness is outlined below (Shapi et al., 2021).

$$Skewness = \frac{Mean - Mode}{Standard\ Deviation}$$

The equation below represents Pearson's Correlation Coefficient (Obilor & Amadi, 2018).

$$\Upsilon = \frac{cov\,(x,y)}{S_x\,S_y} = \frac{\{\frac{1}{n}[\sum_{i=1}^{n} x_i\ y_i\,] - \bar{x}\bar{y}\}}{S_x S_y}$$

3.4.3 Prominent Feature Identification and Selection

Statistical methods such as the Chi-Square test, and Spearman ranks correlation coefficient, were performed to determine the statistical significance of the relationships between variables. Model input features were selected based on the outcome of the statistical tests performed. Attributes with high correlation were eliminated and model performance was gauged. Additionally, during one-hot encoding, the first category in each categorical attribute was dropped to serve as the reference category and avoid multicollinearity. Feature selection was done repeatedly while evaluating the impact of each combination of input features on the performance of the models before the final input features were chosen.

3.4.4 Model Training and Evaluation

The prediction models were trained using stratified K-fold cross-validation (StratifiedShuffleSplit). For optimal model performance, the value for K in K-fold cross-validation should be between 5 and 10. (Wong & Yeh, 2019). The dataset was split into five equal-sized folds, four folds were used to train the models while the last fold was used to evaluate the models' efficacy. This was repeated until each of the five folds had served as both the training set and the testing set. Sci-kit learn's StratifiedShuffleSplit method was employed in splitting the dataset into five folds. StratifiedShuffleSplit ensures stratification to maintain the original dataset's class balance within each fold and randomly shuffles instances before splitting the data to avoid bias (Bradshaw et al., 2023).

Since the dataset is imbalanced, SMOTE was only applied to the training dataset but not the test set, before model fitting. This was done to prevent information leakage from the test set to the training set which leads to model overfitting due to instances from the test set influencing the creation of synthetic samples. Subsequently, the model's performance was evaluated using Sci-Kit Learn's inbuilt performance measurement metrics including accuracy, AUC, and recall. Since the dataset is imbalanced, less emphasis was placed on accuracy. Furthermore, since the cost of false negatives is significantly higher than the cost of false positives, recall was given more weight when evaluating model performance. The equations below represent the formulas for precision, recall, F1 score, accuracy, selectivity, balanced accuracy and error rate (Derczynski, 2016; Johnson & Khoshgoftaar, 2019).

$$Precision = \frac{(True\ Positives)}{(True\ Positives + False\ Positives)}$$

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

$$F1\ Score = 2\left[\frac{(Recall\ X\ Precision)}{(Recall + Precision)}\right]$$

$$Accuracy = \frac{(True\ Positives + True\ Negatives)}{(True\ Positives + False\ Positives + True\ Negatives + False\ Negatives)}$$

$$Selectivity = TNR\ (True\ Negative\ Rate) = \frac{True\ Negative}{True\ Negative + False\ Positive}$$

$$Balanced\ accuracy = \frac{1}{2}\{True\ Positive\ Rate * True\ Negative\ Rate\}$$

$$Error\ rate = 1\text{-}accuracy$$

Depending on the models' performance, hyperparameter tuning, and threshold moving were carried out to further improve model performance. The models were then retrained, following the steps outlined in the conceptual framework in chapter 2 of this dissertation until the best performance was achieved. The models were then evaluated and the best-performing model was selected.

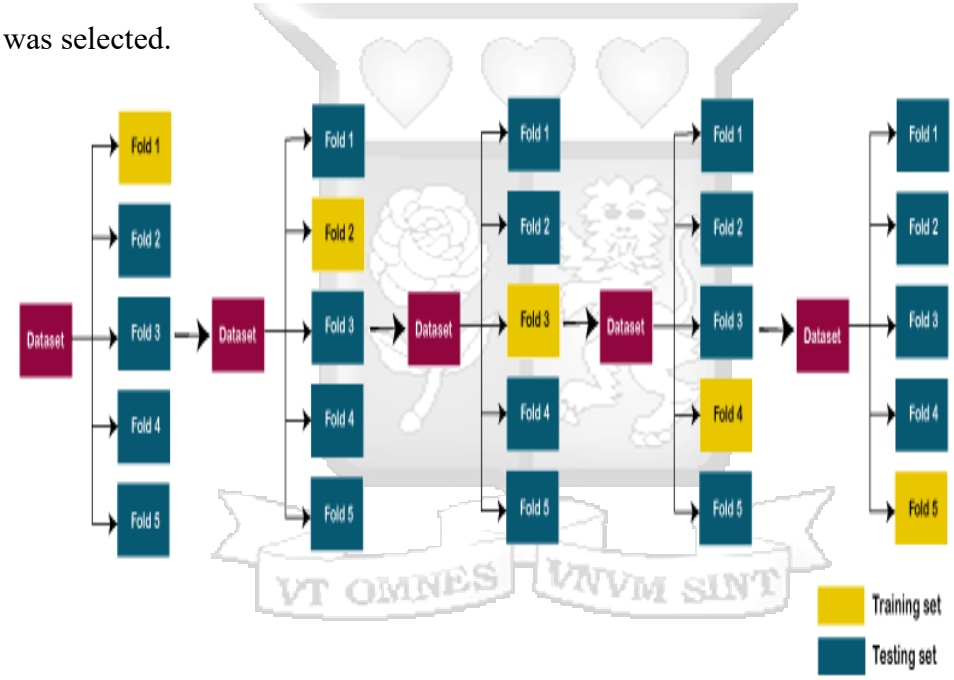


Figure 3. 1 K-Fold Cross Validation (Jaiswal, 2021)

3.4.5 Feature Importance Measurement and Corrective Action Suggestion.

After the best-performing model was selected, its functionality was tested on sample input data. Feature importance measurement to determine how much each input feature contributed to a particular classification coupled with the EDA results were applied to determine the appropriate corrective actions to be suggested.

3.5 System Development Methodology

This dissertation implemented rapid application development to reduce system development time. RAD methodologies refine the system development lifecycle process

ensuring some part of the system is rapidly developed and delivered to the users (Olorunshola & Ogwueleka, 2022). They embrace the dynamic nature of client needs, emphasizing continuous iterative development rather than structural development. RAD methodologies include; phased development, prototyping, and throwaway prototyping. In throwaway prototyping, the analysis, design, and implementation phases are amalgamated before the final design is implemented. Unlike prototyping where the final design prototype evolves into the final system, the final design prototype is discarded and the eventual system design is implemented.

Throwaway prototyping was implemented in developing the customer churn prediction and corrective action suggestion system in this study. A detailed analysis phase was carried out to gather the requisite information needed for the system concept development. Subsequently, multiple design prototypes representing various parts of the final system were developed. The prototypes contained enough information to enable users to comprehend the design issues being worked on. The prototypes were continuously and repeatedly worked on until all the design issues were resolved. Both the web application and the prediction model were developed using throwaway prototyping as some of the parameters involved such as the number of training steps are experimental.

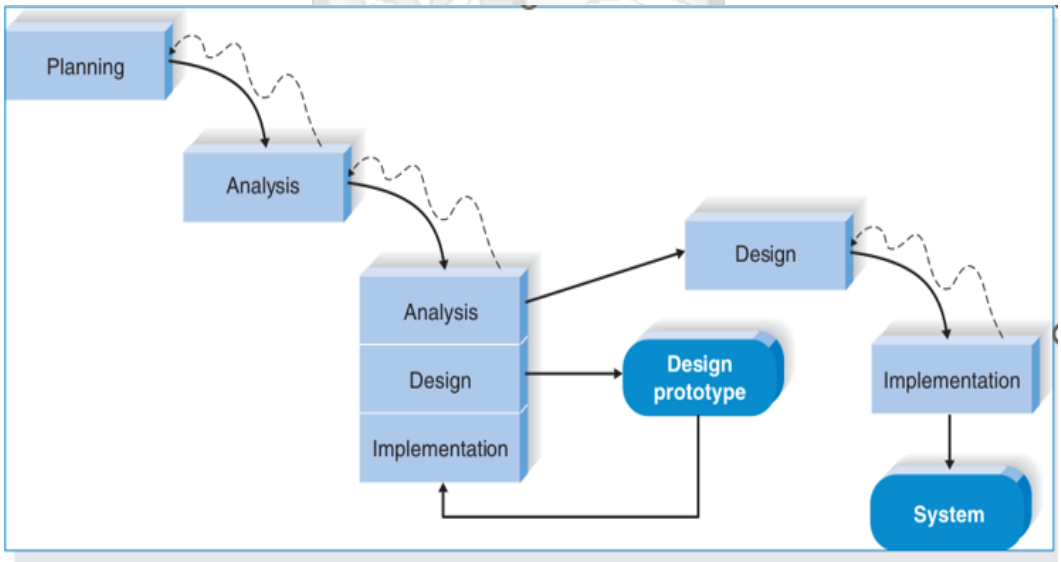


Figure 3. 2 Throwaway Prototyping Phases (PadaKuu, 2021)

3.6 Research Quality

To ensure research quality, the research must be valid, objective, and reliable. All the processes that were undertaken during this research were geared towards the objectives of this study. Clear, precise, and well-defined research objectives and questions outlined in Chapter 1

of this research guided the research process. To ensure quality, only data valid to customer churn in the telecommunications industry from the IBM telco customer churn dataset was used. The IBM dataset is open source and readily available to anyone and thus the results of this research can be independently verified. Additionally, during feature engineering, only the most suitable techniques were applied. Any assumptions made were clearly outlined and discussed during model implementation. Relevant performance metrics were used to evaluate the performance of each classification algorithm employed before the selection of the most appropriate algorithm. Extensive EDA and feature importance measurement was carried out before determining the corrective actions to be suggested. Lastly, this dissertation was repeatedly submitted to experts in the machine learning domain including the dissertation supervisor, mentors, and peers to aid in uncovering potential blind spots or areas of improvement.

3.7 Ethical Considerations

The IBM telco customer churn dataset that was used in this study was obtained from public sources with permission to be used for educational research. The dataset was only used for this dissertation and not for any other purposes. The authors and owners of previous works and images that were referenced in this study were all appropriately cited. Additionally, before commencing this study, ethical clearance from the Strathmore University Institutional Scientific and Ethical Review Committee and the National Commission for Science, Technology and Innovation were obtained.

Chapter 4: System Analysis and Design

4.1 Introduction

System analysis is a critical process that involves a thorough examination and analysis of the various components, interactions, and processes within a system (Dennis et al., 2015). The primary objective of system analysis is to gain a comprehensive understanding of the system's functionality, requirements, and constraints. This process encompasses several key stages, including problem identification, requirements gathering, feasibility studies, system modelling, and scope definition. By engaging in system analysis, stakeholders can ensure that the developed system effectively addresses their needs and effectively resolves any identified issues. Conversely, system design creates a blueprint for a system that fulfils the requirements and needs captured during the system analysis phase (Kendall, 2014). System design entails architecture design, components specifications, user interface design and database design. This chapter delves into the details of the requirements analysis, system architecture, and system design implemented to realize the customer churn prediction and corrective action suggestion model. Furthermore, it explores various Unified Modeling Language (UML) diagrams that elucidate the interactions between the users of the developed system and its components and processes.

4.2 Requirements Analysis

Requirements analysis is a critical phase in the system development process, involving the systematic collection of information regarding the objectives, features, characteristics, and expected behaviour of the system. (Curumsing et al., 2019). The main purpose of requirements elicitation is to gather stakeholder needs, define the scope of the system, and establish a foundation for system design and development thus ensuring the efficiency and effectiveness of the resulting system. To ensure the robustness and comprehensiveness of the developed system, various techniques can be employed in the requirements-gathering process including interviews, surveys, workshops, observations, prototyping and benchmarking. Diverse techniques were followed in identifying the requirements integral to the development of the model. Existing customer churn prediction tools were extensively reviewed providing valuable insights into the core functionalities a churn prediction tool should possess. Additionally, various research papers on customer churn prediction highlighted in Chapter 2 of this dissertation, were analyzed providing vital information on the functional and non-functional requirements essential to effective churn prediction.

4.2.1 Functional Requirements

The customer churn prediction and corrective action suggestion model should;

i. Provide a user interface that enables a user to input customer data to get a prediction.
ii. Accurately predict and display the likelihood of a customer churning based on the customer data input by the user.
iii. In the event of a positive churn prediction, suggest and display corrective actions.

4.2.2 Non-Functional Requirements

The non-functional requirements include;

i. The system should provide customer churn predictions and corrective action suggestions with minimal latency ensuring a responsive user experience.
ii. The user interface should be intuitive and user friendly making it easy for users to interact with the system without the need for any prior training.
iii. The user interface should be compatible with all major web browsers.
iv. The developed model should be easy to maintain.
v. The tool should be reliable and fault-tolerant.
vi. The developed system should be secure.

4.3 System Architecture

System architecture serves as a foundational blueprint that outlines the structure and behaviour of a system, delineating the interaction between its components and their roles in fulfilling system requirements. (Sangwan, 2014). The main components of the developed system are the web interface and the machine learning model. The web interface serves as the primary point of interaction between users and the system. It enables interactions between the users and the developed model. It facilitates the input of telco customer data into the system through various select boxes, providing a user-friendly interface for data entry. Once the user inputs customer data, the web interface directs this information to the machine learning model for further processing. On the other hand, the machine learning model acts as the analytical engine of the system. It receives the pre-processed customer data from the web interface, performs churn prediction analysis, and generates corrective action suggestions based on the input data. This predictive functionality is the core feature of the system, enabling it to anticipate customer churn and provide actionable insights.

The web interface offers users two primary options upon entering customer data; the ability to display and download the entered information as a CSV file, and the option to initiate

churn prediction and receive corrective action suggestions. If the user selects the prediction option, the web interface seamlessly integrates with the machine learning model, feeding the input data into the trained model for analysis. Subsequently, the model processes the input data and generates predictions regarding customer churn risk, along with recommended corrective actions. These outcomes are then communicated back to the user through the web interface, providing actionable insights and enabling informed decision-making. Overall, the system architecture ensures seamless integration between the web interface and the machine learning model, facilitating efficient data flow and enabling users to leverage predictive analytics for addressing customer churn effectively. The figure below outlines the developed system's architecture.

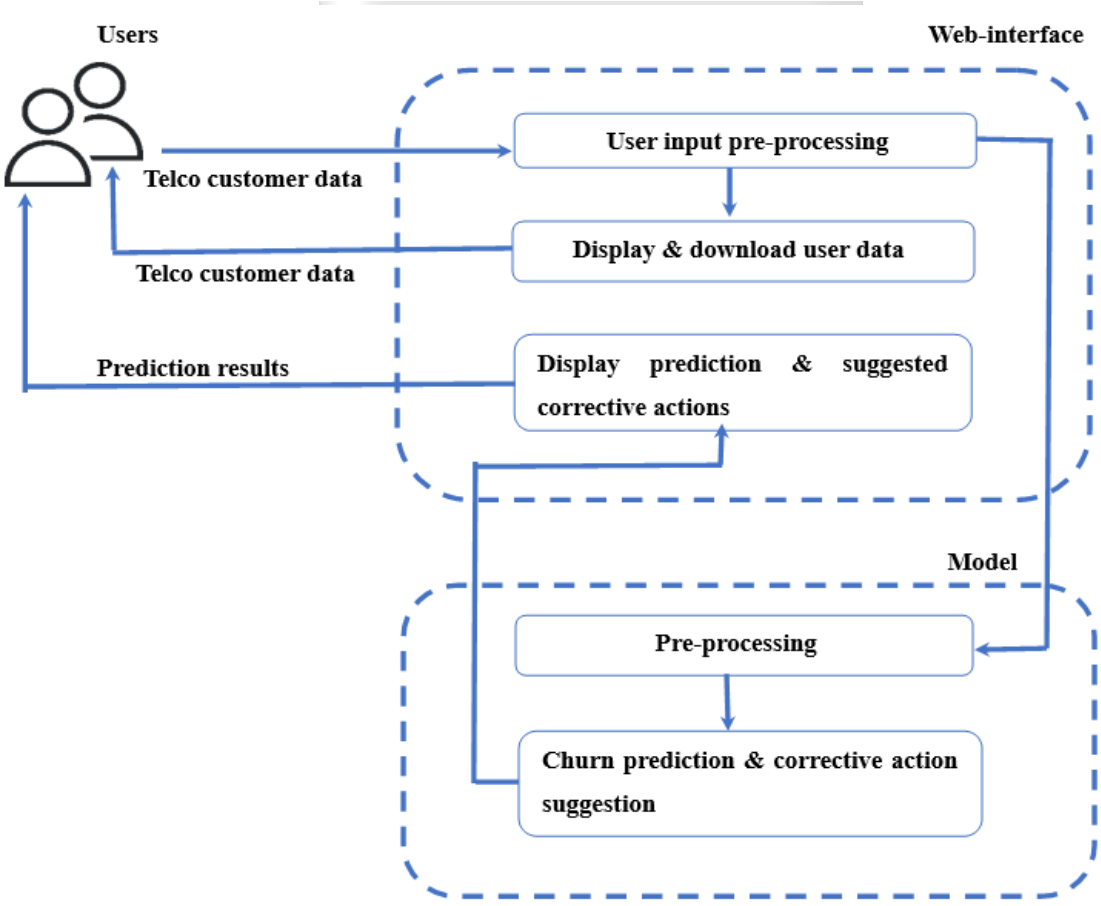


Figure 4. 1 System Architecture

4.4 Use Case Diagram

A use case diagram serves as a visual representation illustrating how users interact with a system and how the system responds to those interactions (Fauzan et al., 2019). It provides a high-level overview of user actions and system behaviour, offering valuable insights into the

functional requirements of the system. The main components of a use case diagram include the actors, use cases, relationships and the system boundary. Use case diagrams enhance the understanding of the functional requirements of the system, identify and organize the use cases the system must support, expound on the roles of the various actors, and highlight system behaviour. The figure below illustrates the customer churn prediction and corrective action suggestion use case diagram.

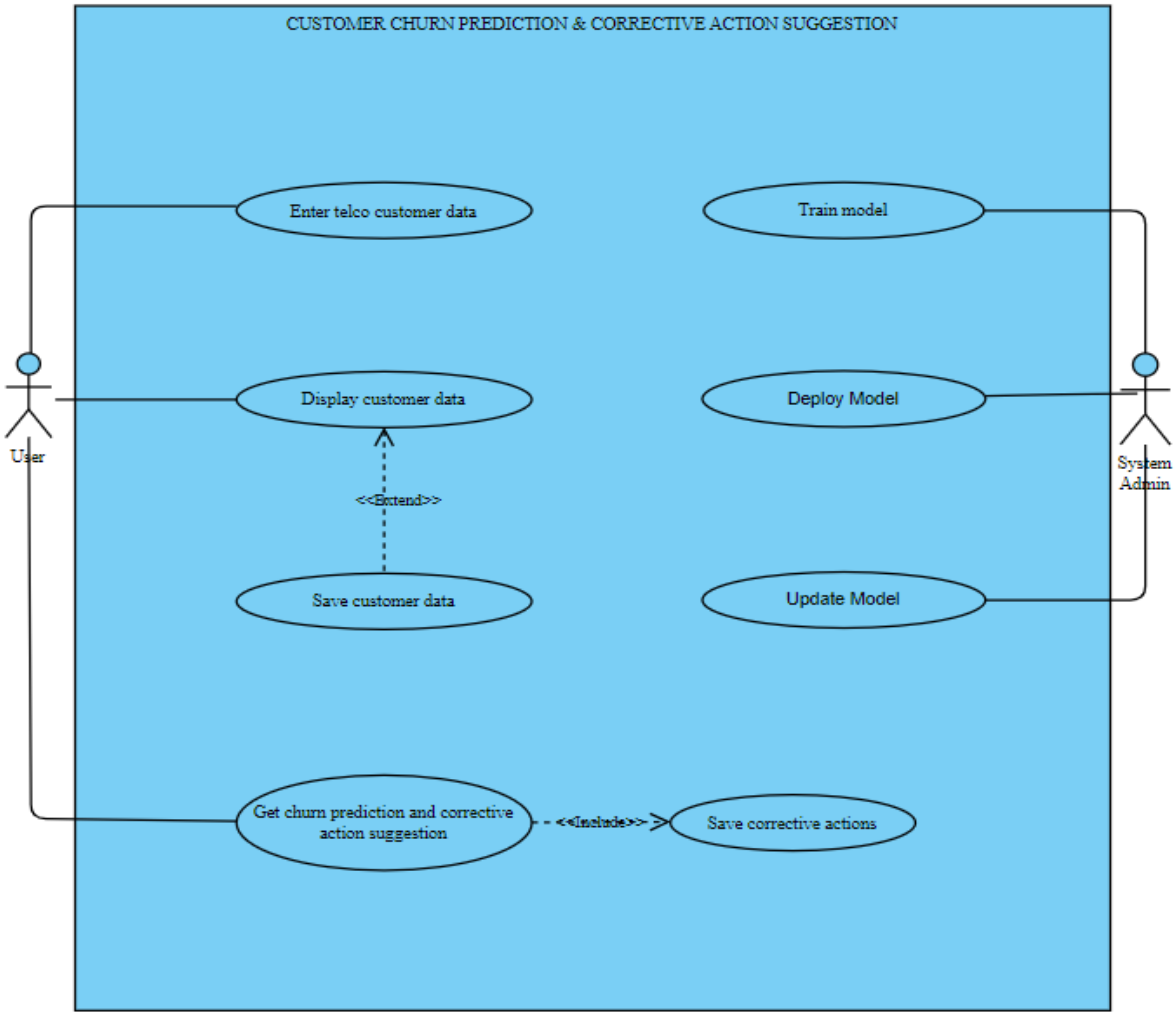


Figure 4. 2 Use Case Diagram

4.4.1 Use Case Descriptions

Table 4.1 below highlights the various use cases, necessary preconditions, success scenarios and the post conditions after the use case is executed.

Table 4. 1 Use Case Descriptions

Actor	Use Case	Preconditions	Primary success scenario	Postconditions
User	Enter telco customer data.	None	The user enters the telco customer data in the respective fields	Customer data is fed into the tool
User	Display customer data	Customer data must have been previously entered by the user	Users can view the customer data they entered	Customer data is displayed to the user via the web interface
User	Save customer data.	Customer data must have been previously entered by the user The customer data must already be displayed to the user	Users can save the customer data they entered into the system as a CSV file	CSV file of input customer data available.
User	Get churn prediction & corrective action suggestions	Customer data must have been previously entered by the user.	Users can view the churn prediction and the suggested corrective actions.	Prediction and suggested corrective actions displayed.
User	Save corrective actions	Customer data must have been previously entered by the user. Prediction and corrective action suggestions must have been obtained.	Users can save the recommended corrective actions.	A file containing the suggested corrective actions.
System Admin	Train model	None	Model fit with training data	Trained classification model
System Admin	Deploy model	The model must have been trained	Model functionalities accessible to users	Live model
System Admin	Update model	The model must have been deployed first	The model is updated to the desired version	Users have access to the most recent model version

4.5 Sequence Diagrams

A sequence diagram is a visual representation of the communication and interactions between various components of a system (Jacobson & Booch, 2021). Sequence diagrams represent the chronological order of events and messages exchanged between various components within the system. They make it easier for both the system designers and users to understand the flow of data and controls within the system. The figure below illustrates the sequence diagram for the developed system.

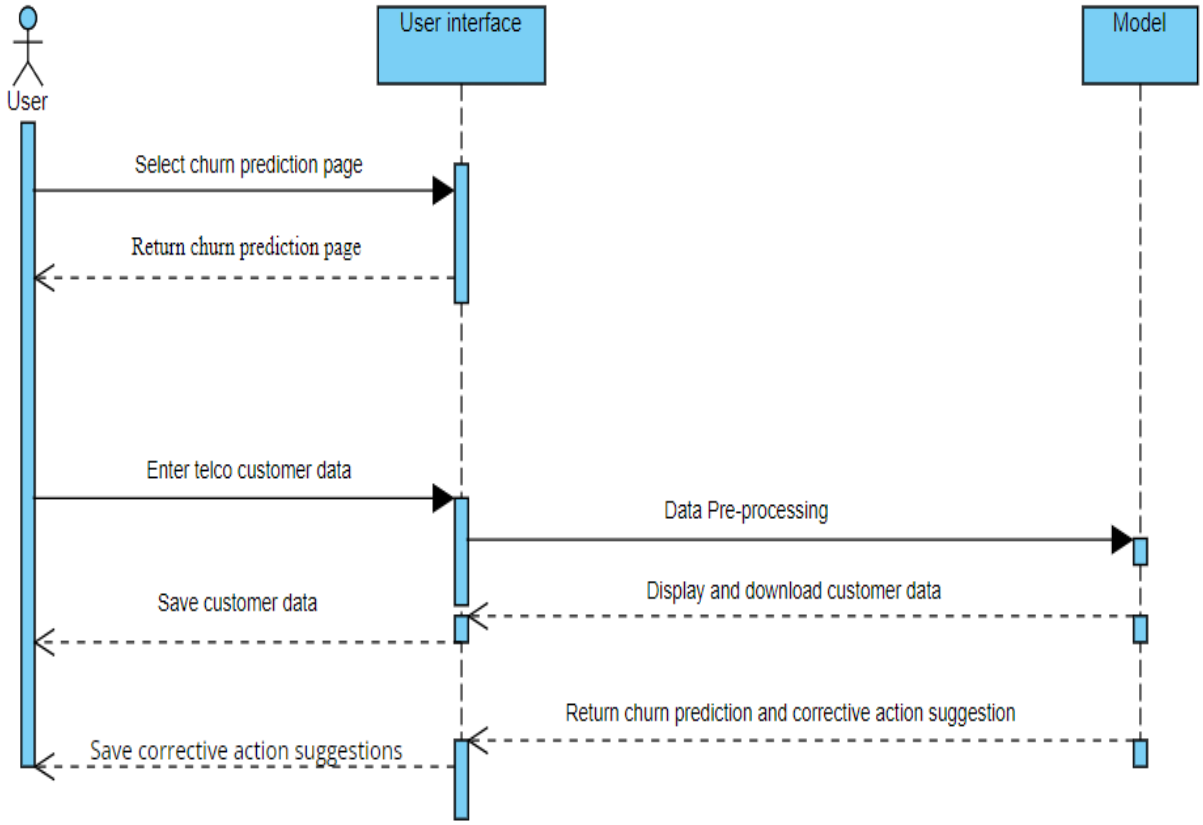


Figure 4. 3 Sequence Diagram

4.6 Wireframes

A wireframe is a two-dimensional skeletal blueprint outlining the basic structure, components, layout and navigation of a user interface (Beaird et al., 2020). Wireframes exhibit low fidelity and thus lack many aesthetic visual design elements such as images and colour. They outline the arrangement of elements on the user interface highlighting the hierarchy of information and the sequence of user interactions.

4.6.2 Home Page Wireframe

The figure below shows the home page wireframe. This will be the initial page the user will see once the web application is launched. It contains a select box to navigate between the home page and the churn prediction page.

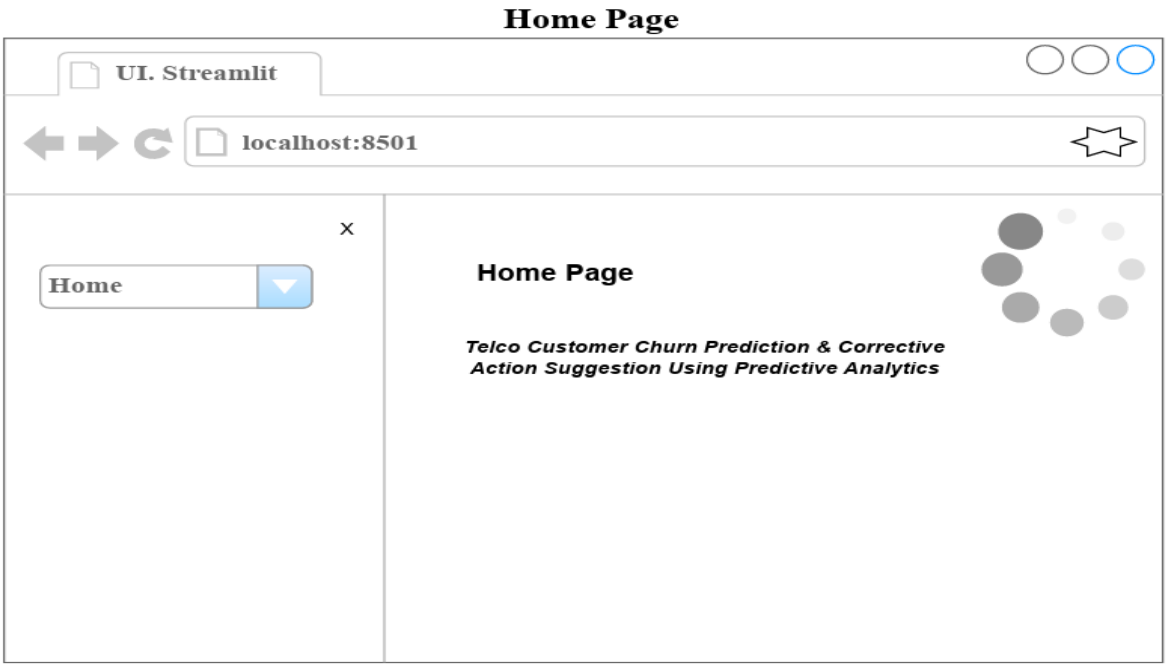


Figure 4. 4 Home Page Wireframe

4.6.3 Churn Prediction Wireframe

The churn prediction wireframe is illustrated below. It contains a select box to navigate between the home page and the churn prediction page, various select boxes to enable data input, display customer data and predict buttons.

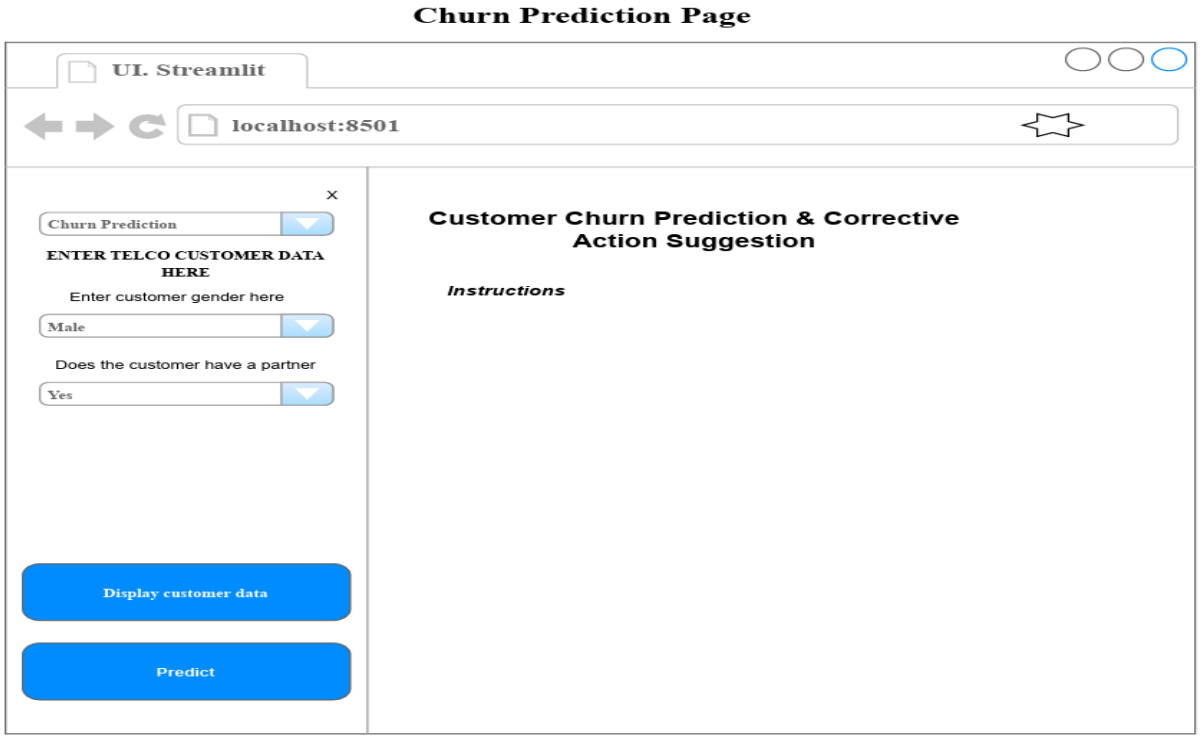


Figure 4. 5 Churn Prediction Page Wireframe

Chapter 5: System Implementation and Testing

The outcome of this research was a locally hosted web application that can predict the likelihood of a telco customer churning and suggest corrective actions. This section outlines the processes followed in developing the churn prediction and corrective action suggestion system. The system was developed by training several classification algorithms. Subsequently, the trained algorithms were then evaluated and the best-performing one was selected to be used for churn prediction. The selected model was then embedded into a web application to enable easier user interaction with the model. System testing encompassed both functionality and usability assessment to verify whether the developed system achieved the objectives outlined at the outset of this dissertation.

5.1 System Overview

The developed customer churn prediction and corrective action suggestion system comprised two main components; the trained model and a web interface.

5.1.1 Models

The No Free Lunch Theorem highlighted in Chapter 2 of this dissertation asserts that there is no single machine learning algorithm that works best for every use case. It stresses the importance of understanding the problem at hand and comprehensively experimenting and evaluating several algorithms before selecting one for a particular task. Therefore, several classification algorithms including Logistic Regression, Gaussian Naïve Bayes, Complement Naïve Bayes, K-Nearest Neighbours, Random Forest Classifier and Categorical Boosting were developed and evaluated before the best-performing one was selected. The choice of classification algorithms was heavily influenced by the nature of the dataset and the computational cost associated with each model.

5.1.2 Web interface

The web application was developed using Streamlit, a free and open-source Python library that enables the transformation of Python scripts into interactive web applications. Streamlit streamlines the creation of interactive web applications through its simple syntax and smooth integration with Python libraries. Additionally, HTML and CSS were also employed to enhance the appearance of the created web interface. The developed web interface has two pages; a home page and a churn prediction page. The home page briefly describes the application and its functionality. It contains a select box that can be used to navigate between the home page and the churn prediction page. The churn prediction page is the primary interface of the application. It contains several select boxes that enable the user to enter telco customer

data required for customer churn prediction. Additionally, it highlights instructions to be followed when using the tool. The page contains two buttons, a display customer data button and a predict button. The display customer data button enables the user to view and download the captured customer data while the predict button enables the user to view the prediction and suggested corrective actions. Figures 5.1, 5.2 and 5.3 below illustrate the home page and the churn prediction page.

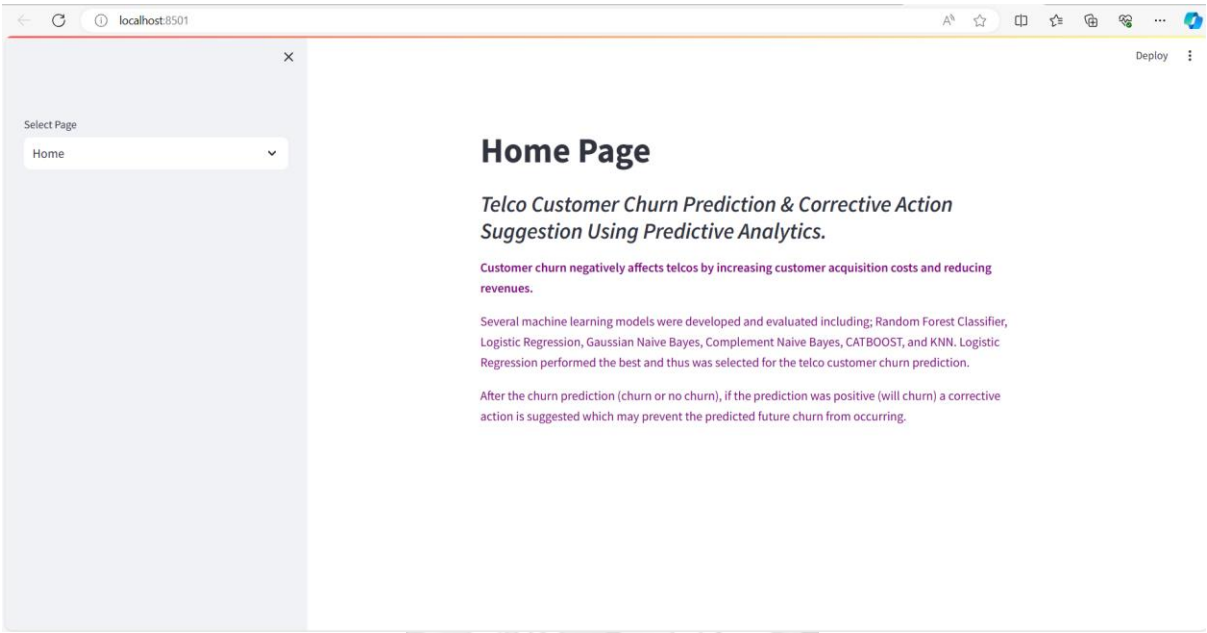


Figure 5. 1 Home Page

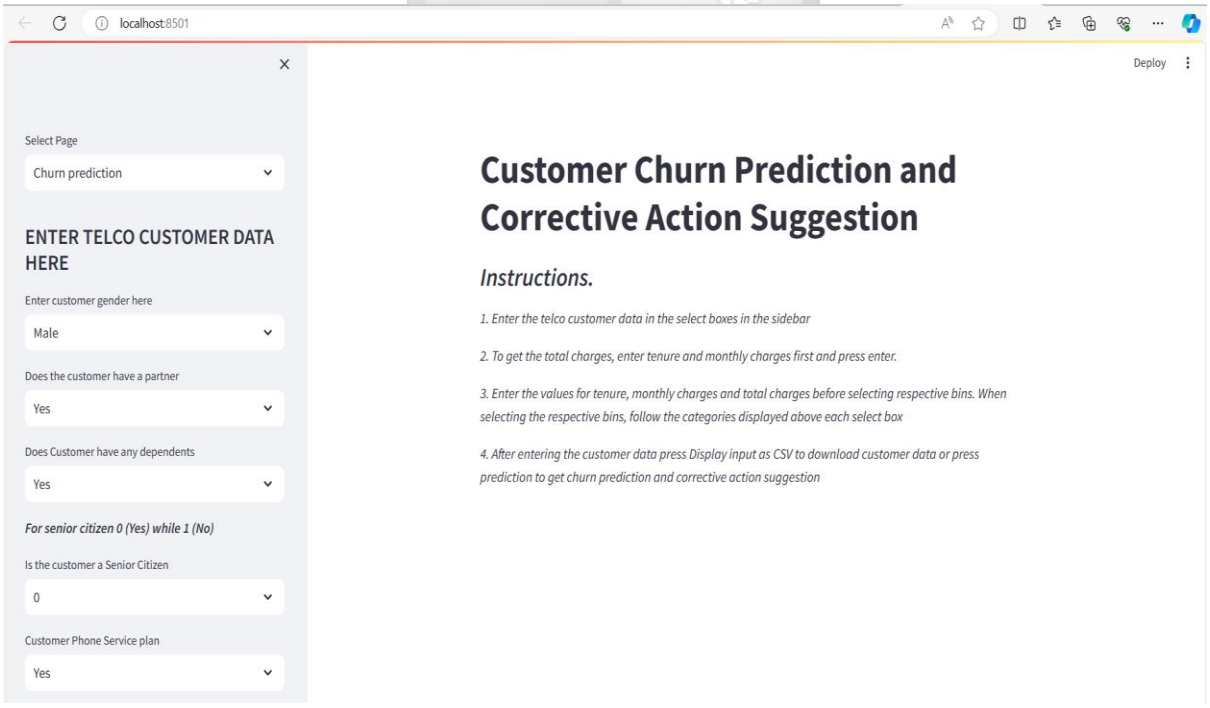


Figure 5. 2 Churn Prediction Page

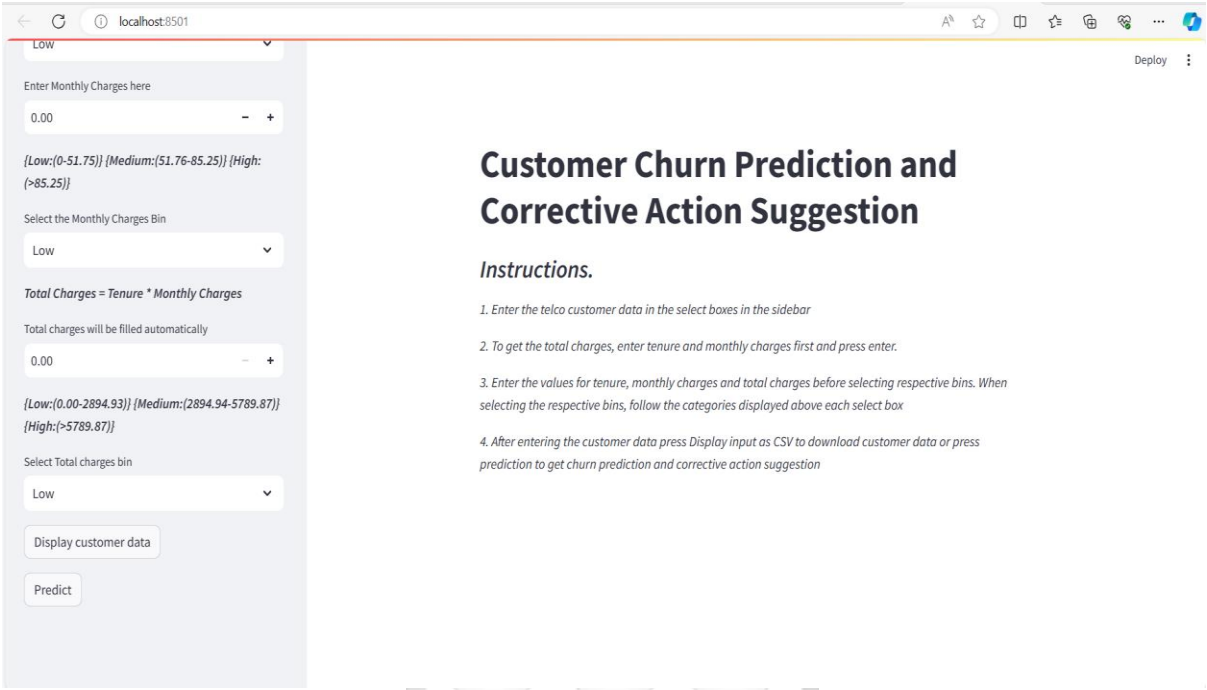


Figure 5. 3 Churn Prediction Page Continuation

5.2 System Implementation

Throwaway prototyping, a type of rapid application development was implemented in developing the churn prediction and corrective action suggestion system as highlighted in Chapter 3 of this dissertation. The front end of the system is a locally hosted web application developed using HTML, CSS and Python (Streamlit). The back end of the system is a trained classification model that provides churn prediction and corrective action suggestions. The system implementation process was iterative with several prototypes designed and implemented before arriving at the final system design.

5.2.1 Development Environment

The customer churn prediction and corrective action suggestion system was developed in the following environment;

i. Asus TUF Dash F15 FX516PR_FX516PR
ii. 11th Gen Intel(R) Core (TM) i7-11370H @ 3.30GHz, 3302 MHz, 4 Core(s), 8 Logical Processor(s)
iii. 24GB System RAM
iv. 2TB Solid-state drive
v. Microsoft Windows 11 Pro version 10.0.22631
vi. Python 3.12.1
vii. Pycharm Community Edition version 2022.3.1
viii. Catboost 1.2.3

ix. Imbalanced-learn 0.12.0
x. Joblib 1.3.2
xi. Lime 0.2.0.1
xii. Matplotlib 3.8.2
xiii. Numpy 1.26.3
xiv. Pandas 2.2.0
xv. Streamlit 1.32.0
xvi. Scikit-learn 1.4.1
xvii. Scipy 1.12.0
xviii. Seaborn 0.13.2
xix. Shap 0.44.1

5.2.2 Dataset Collection and Pre-processing

The dataset used in this study was retrieved from the OpenML website via API as highlighted in Chapter 3 of this dissertation. The read_csv function of the Pandas library was used to load the retrieved dataset for pre-processing. The Pandas info() method was used to inspect the datatypes of each column in the dataset. The astype() method was then applied to convert each column's datatype to the appropriate datatype. Subsequently, the duplicated() and sum () methods from the Pandas library were applied to check for duplicated entries within the dataset. The dataset did not contain any duplicated entries. The isna () method was then used to detect missing values in our dataset. None of the columns in the dataset had any missing values. However, upon further pre-processing of the dataset, it was discovered that the total charges column contained eleven missing values. Further analysis of the total charges column revealed that when tenure was one, the value of total charges was the same as the value of monthly charges implying that the total charge was the product of tenure and monthly charge. In all the eleven instances in which the value of the total charge was missing, the tenure was zero. Therefore, the missing total charges were imputed by zero.

5.2.3 Exploratory Data Analysis and Feature Engineering

Upon completion of the data cleaning process, an exploratory data analysis was carried out to gain a better understanding of the dataset. Exploratory data analysis involved univariate, bivariate and multivariate analyses of the dataset's attributes. Exploratory data analysis was primarily carried out using NumPy, Pandas, Scipy and Matplotlib Python libraries. The initial step of EDA entailed a univariate analysis of the dataset's numerical columns; tenure, monthly charges and total charges. The describe() method was used to compute the summary statistics

of the numerical columns as highlighted in Figure 5.4 below. The standard deviation of the three numerical columns was particularly important as it highlighted the dispersion of data around the mean. Tenure had a standard deviation of 24.56 months indicating that most tenure values lie within 24.56 months of the mean tenure value while monthly charges had a standard deviation of 30.09, thus monthly charges values were expected to be within 30 dollars of the mean monthly charges. Total charges had the largest dispersion from its mean value.

```
The summary statistics of the telco dataset are:
                 count         mean          std    min     25%     50%      75%      max
tenure          7043.0    32.371149    24.559481   0.00    9.00   29.00    55.00    72.00
MonthlyCharges  7043.0    64.761692    30.090047  18.25   35.50   70.35    89.85   118.75
TotalCharges    7043.0  2279.734304  2266.794470   0.00  398.55  1394.55  3786.60  8684.80
```

Figure 5. 4 Summary Statistics

Normality testing was then carried out on the numerical columns via statistical methods and visualizations. The Anderson-Darling normality test was selected over the Shapiro-Wilks test due to the dataset size. The Shapiro-Wilks test is sensitive to large sample sizes, especially if the sample size is greater than 5000 since as the sample size increases, it becomes more susceptible to deviations from normality (Ishak et al., 2024). The Anderson-Darling test was performed at a 5% significance level. During the normality testing, the null hypothesis was the sample had a Gaussian distribution while the alternative hypothesis was the sample did not have a Gaussian distribution. All three numerical columns were not normally distributed as indicated in Figure 5.5 below. This was further confirmed by the histograms of the numerical columns shown in Figures 5.6, 5.7 and 5.8 below, none simulated a bell curve. The numerical columns were thus subjected to log, square root, inverse and exponential transformations. However, none of the transformations yielded a Gaussian distribution. The numerical columns were binned creating three new features; tenure-binned, MonthlyCharges-binned and TotalCharges-binned.

```
Statistic: 203.23547
Critical Values: [0.576 0.656 0.787 0.917 1.091]
Significance Levels: [15.  10.   5.   2.5  1. ]
Tenure Sample does not look Gaussian (reject H0)
Statistic: 346.63803
Critical Values: [0.576 0.656 0.787 0.917 1.091]
Significance Levels: [15.  10.   5.   2.5  1. ]
Totalcharges Sample does not look Gaussian (reject H0)
Statistic: 170.55524
Critical Values: [0.576 0.656 0.787 0.917 1.091]
Significance Levels: [15.  10.   5.   2.5  1. ]
Monthlycharges Sample does not look Gaussian (reject H0)
```

Figure 5. 5 Anderson-Darling Test Results

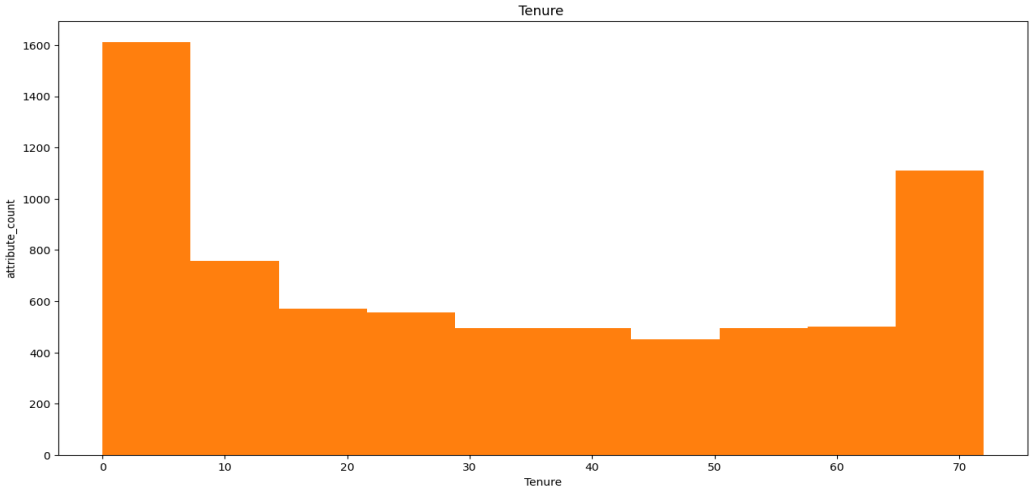


Figure 5. 6 Tenure Histogram

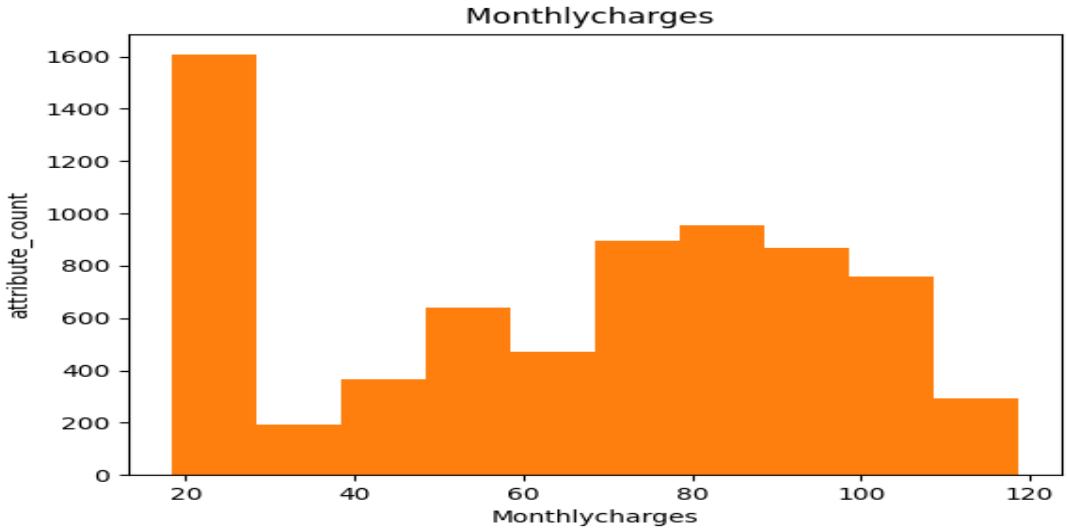


Figure 5. 7 Monthly Charges Histogram

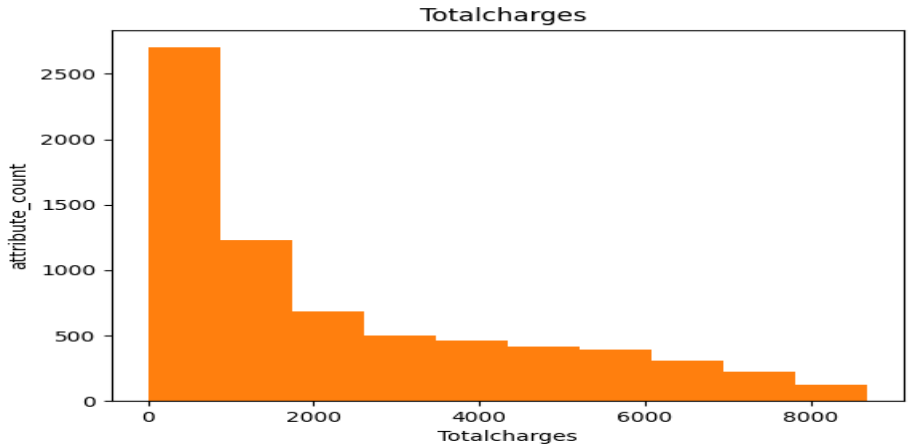


Figure 5. 8Total Charges Histogram

Lastly, for the univariate analysis, statistical visualizations (pie charts and bar plots) were created to provide insights into the frequency distribution within selected categorical attributes. Figure 5.9 below highlights the imbalance in our dataset, with 73.5% of the target variable being non-churners while 26.5% represented churners.

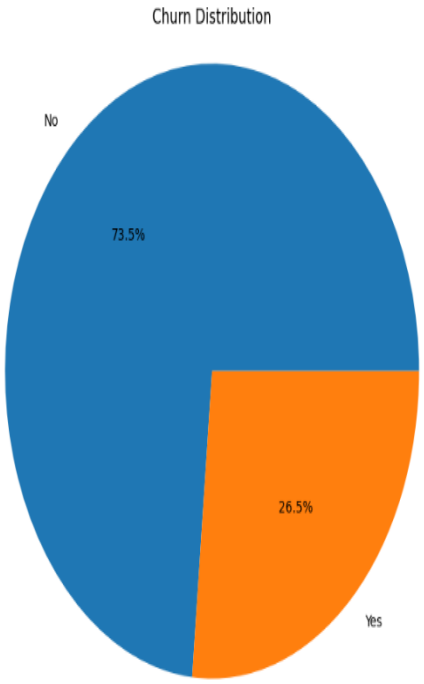


Figure 5. 9 Churn Distribution

The second stage of EDA involved bivariate analysis between numerical and numerical, numerical and categorical and categorical and categorical variables. Statistical methods and

visualizations were implemented to carry out the bivariate analysis with the type of method/visualization chosen depending on the pair variables datatypes. For numerical and numerical variables, the Spearman Rank Correlation Coefficient was selected over the Pearson Correlation Coefficient as none of the numerical variables was normally distributed, a requisite assumption when applying Pearson correlation. The significance level used was 0.05. The null hypothesis during the test was that the two samples did not have a monotonic relationship. In contrast, the alternative hypothesis was the existence of a monotonic relationship between the two variables. The Spearman correlation coefficient between tenure and monthly charges was 0.2764, tenure and total charges was 0.8897 and monthly charges and total charges was 0.6380.

```
tenure , MonthlyCharges spearman correlation coefficient: 0.27641678933130215
tenure and MonthlyCharges have a monotonic relationship. ALTERNATIVE HYPOTHESIS ACCEPTED!!
tenure , TotalCharges spearman correlation coefficient: 0.8896957900597577
tenure and TotalCharges have a monotonic relationship. ALTERNATIVE HYPOTHESIS ACCEPTED!!
MonthlyCharges , TotalCharges spearman correlation coefficient: 0.638028390201301
MonthlyCharges and TotalCharges have a monotonic relationship. ALTERNATIVE HYPOTHESIS ACCEPTED!!
```

Figure 5. 10 Spearman Rank Correlation Coefficients

The Mann-Whitney U test was then applied to the numerical columns and the target variable to determine if they were drawn from the same population. The null hypothesis applied was that the samples came from the same population while the alternative hypothesis was that they came from different populations. The test was performed at a 0.05 significance level, yielding none of the samples came from the same distribution. Therefore, all the numerical columns were retained as features since coming from different distributions suggests that each feature may provide unique information that may affect predictive performance.

```
Correlation with **tenure**
Correlation between tenure and Churn
Statistics = 48981984.50000, p = 0.00000
Different distribution (reject H0)
----

Correlation with **MonthlyCharges**
Correlation between MonthlyCharges and Churn
Statistics = 49603849.00000, p = 0.00000
Different distribution (reject H0)
----

Correlation with **TotalCharges**
Correlation between TotalCharges and Churn
Statistics = 49554833.00000, p = 0.00000
Different distribution (reject H0)
----
```

Figure 5. 11 Mann Whitney Results

To determine the relationship between dichotomous categorical variables and the target variable churn, also a categorical variable, the Chi-Square statistic was applied. The null hypothesis was that the two variables were independent of each other while the alternative hypothesis was the two variables were not independent of each other. Only gender and phone service were independent of the target variable churn as shown in Figure 5.12 below. Since variables that are not significantly associated with the target variable may not provide vital predictive information, both gender and phone service were dropped as input features.

```
**Chi-Squre Correlation Between Dichotomous Features with Target : Churn**
Correlation between **gender** and **Churn**
p-value : 0.48657873605618596, degree of freedom: 1
probability=0.950, critical=3.841, stat=0.484
Independent (fail to reject H0)
-----------------------------------

Correlation between **Partner** and **Churn**
p-value : 2.1399113440759935e-36, degree of freedom: 1
probability=0.950, critical=3.841, stat=158.733
Dependent (reject H0)
-----------------------------------

Correlation between **Dependents** and **Churn**
p-value : 4.92492166121541 96e-43, degree of freedom: 1
probability=0.950, critical=3.841, stat=189.129
Dependent (reject H0)
-----------------------------------

Correlation between **PaperlessBilling** and **Churn**
p-value : 4.073354668665985e-58, degree of freedom: 1
probability=0.950, critical=3.841, stat=258.278
Dependent (reject H0)
-----------------------------------

Correlation between **PhoneService** and **Churn**
p-value : 0.3387825358066928, degree of freedom: 1
probability=0.950, critical=3.841, stat=0.915
Independent (fail to reject H0)
```

Figure 5. 12 Chi-Square Results

Cramér's V was utilized to assess the relationship between polytomous variables and the target variable churn. Scores for Cramér's V range from 0 to 1, with 0 indicating no association and 1 indicating a perfect association between the categorical variables. The results are outlined in Figure 5.13 below. The contract column had the highest score of 0.41 while the multiple lines column had the lowest score of 0.04. The multiple lines column was thus dropped due to its low Cramer's V score as low scores imply a weak association with the target variable churn and may not provide much predictive value. Lastly, for bivariate analysis, several visualizations were plotted to gain further understanding of the relationship between the various input features and the target variable churn. These visualizations are highlighted in the corrective action suggestion subsection below.

```
**Correlation Between Polytomous Features with Target : Churn**
Contract 0.40979839182553446
OnlineSecurity 0.34701606688272874
TechSupport 0.3425261587493695
InternetService 0.3220367323307425
PaymentMethod 0.3026771381187204
OnlineBackup 0.291850036724674
DeviceProtection 0.28109492388964397
StreamingMovies 0.2303514728244421
StreamingTV 0.22990176915403476
MultipleLines 0.03639958908232507
```

Figure 5. 13 Cramer's V Score

5.2.4 Model Training and Evaluation

After cleaning the dataset, exploratory data analysis and feature engineering, the processed dataset was saved and used to train the classification algorithms. Six classification algorithms; Logistic Regression, Gaussian Naïve Bayes, Complement Naïve Bayes, K-NN, Random Forest Classifier and CatBoost were trained using the saved dataset. The initial step in model training entailed the separation of the dataset into input features and the target variable. The new features created through binning were then converted into categorical datatypes. Subsequently, the categorical columns were encoded using Sci-kit learn's OneHotEncoder since the majority of the categorical columns did not exhibit any inherent ranking within their subcategories. The first category of each input feature was dropped to avoid multicollinearity. Depending on the classification algorithm, the target variable was then encoded using Scikit-learn's LabelEncoder() method. The target variable was encoded for all the classification algorithms implemented except for CatBoost and Random Forest Classifier as they inherently handle the encoding of the target variable. The numerical columns were then scaled using Sci-kit learn's RobustScaler() method.

Initially, the pre-processed data was split using Scikit-learn's train_test_split method into a training set (80% of the dataset) and a testing set (20% of the dataset). The random_state was set to 42 to ensure that the same random split of data was generated each time the code was run. The six classification algorithms were then fitted with the training data and their performance was evaluated. The training process was then repeated but instead of using the train_test_split() method, Scikit-learn's StratifiedShuffleSplit() method was used to split the dataset into 5 folds while maintaining the same train/test split and random state. The models were then fit with the training data and their performance on the test data was evaluated. To further improve model performance by addressing the class imbalance, SMOTE was solely applied to the training dataset after the StratifiedShuffleSplit method had been used to split the dataset. The models' performance was then reevaluated. A grid search was then performed to

obtain the optimal combination of hyperparameters, which were then used to retrain and evaluate the models. In addition to applying SMOTE to address the class imbalance, threshold moving and custom priors (Bayesian Classifiers only) were applied before the final model evaluation. In determining the optimal threshold value for classification, the precision-recall curves for each classification algorithm were used.

Model performance evaluation was done using Scikit-learn's inbuilt performance evaluation metrics including accuracy, precision, recall, and F1 Scores. Moreover, the confusion matrices and classification reports for each iteration of model training and evaluation were determined. Due to the nature of the problem i.e. classification of churners and the imbalance in the dataset, recall was given more weight over the other performance metrics. Additionally, when evaluating the models, the execution time, computational cost, model interpretation and ease of feature importance measurement were also considered. The classification algorithm with the best metrics, logistic regression was selected from the six algorithms and used to develop the churn prediction and corrective action suggestion system. Python's Joblib library was then used to save the logistic regression model and its encoder.

5.2.5 Corrective Action Suggestion

The saved logistic regression model was then presented with unseen test data and used to make predictions. In the event of a positive churn prediction, the model needed to suggest the appropriate corrective actions. Two approaches were followed in determining the appropriate corrective actions to be suggested. Firstly, the bivariate analysis conducted between the input features and the target variable was used to determine the corrective actions to be suggested. For each attribute subcategory, the churn count was determined and plotted against the respective subcategories as shown in Figure 5.14 below. The attribute subcategory with the lowest churn count, e.g. in the case of payment method it was credit card (automatic), was the appropriate corrective action i.e. if the customer churned and their payment method was not credit card automatic, the appropriate corrective action suggestion would be to recommend credit card automatic as a payment option to the churned customer. A similar approach was used in determining the appropriate corrective action suggestions for selected input attributes.

Bivariate analyses of input features and the target variable churn provide a simple straightforward way of determining the corrective actions to be suggested. However, they do not capture the complex interplay of features contributing to churn. To address this, the logistic regression model coefficients were also used to determine the appropriate corrective actions. The coefficient polarity and absolute value determine the effect of each input attribute on churn

prediction. Positive coefficients indicate the input feature contributes towards a positive churn prediction(churner) while negative coefficients indicate that the input feature contributes towards a negative churn prediction(non-churner). The higher the absolute value the greater the contribution towards the respective classification. Logistic regression feature coefficients were given more weight over the bivariate analysis in determining the appropriate corrective actions. Figure 5.16 illustrates the implementation of the corrective actions determined based on the approaches discussed above.

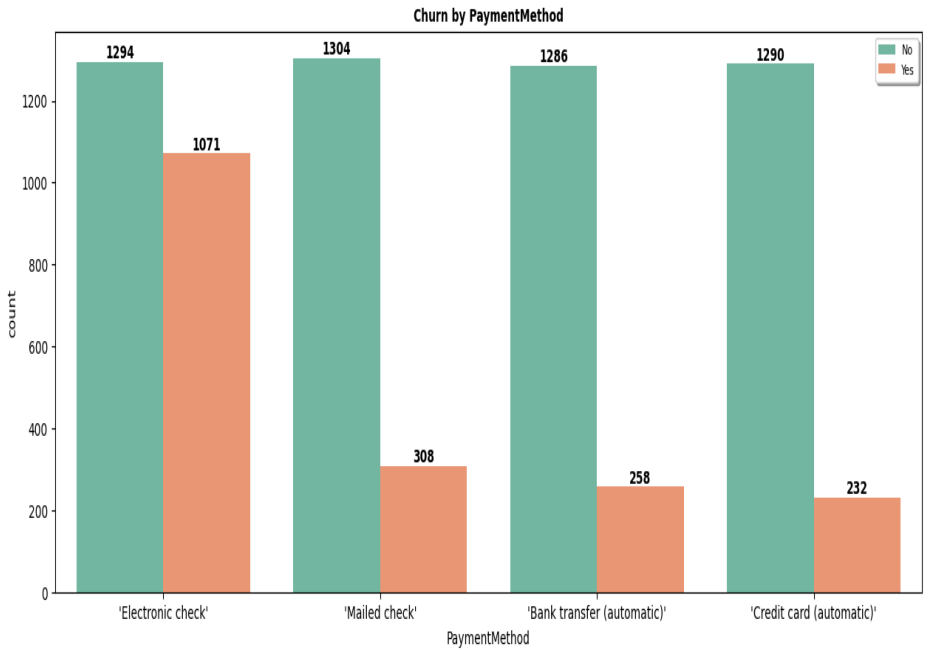


Figure 5. 14 Churn by Payment Method

```
Feature Coefficients:
gender_Male: 0.003808814540884713
PaperlessBilling_Yes: 0.41830332779666257
TotalCharges-binned_Low: -0.43970363348653935
TotalCharges-binned_Medium: -0.2825786678793455
StreamingMovies_No: 0.041251100975146074
StreamingMovies_Yes: 0.25003544983190984
Dependents_Yes: -0.2678467915483641
DeviceProtection_No: 0.20393740515098044
DeviceProtection_Yes: 0.08734914565586964
StreamingTV_No: 0.015038551899257149
StreamingTV_Yes: 0.27624799890789936
Partner_Yes: -0.012441829813327506
OnlineBackup_No: 0.20978456494947673
OnlineBackup_Yes: 0.08150198585728485
MultipleLines_No: -0.21337180753491586
MultipleLines_Yes: -0.020988596021220315
PhoneService_Yes: -0.23436040355584264
TechSupport_No: 0.3575839582458469
TechSupport_Yes: -0.06629740743889724
InternetService_DSL: -0.7733175892108458
InternetService_No: -0.2964619840549278
OnlineSecurity_No: 0.3677706626352971
OnlineSecurity_Yes: -0.07648411182812534
Contract_'Two year': -0.6610988836320197
Contract_Month-to-month: 0.7886307396342601
MonthlyCharges-binned_Low: 0.5061058298633423
MonthlyCharges-binned_Medium: 0.1675511850499269
PaymentMethod_'Credit card (automatic)': -0.043607337477940926
```

Figure 5. 15 Logistic Regression Sample Coefficients

```
def corrective_action_suggestion():
    print('RECOMMEND DISCOUNT PACKAGES CURRENTLY BEING OFFERED BY THE TELCO TO THE CUSTOMER')
    if prediction == 0:
        print('THE CUSTOMER DID NOT CHURN')
    if prediction == 1 and instance['DeviceProtection'] == 'No':
        print('RECOMMEND DEVICE PROTECTION PACKAGES OFFERED BY THE TELCO TO THE CUSTOMER')
    if prediction == 1 and instance['OnlineBackup'] == 'No':
        print('RECOMMEND CLOUD/SYNC UP DRIVE OPTIONS OFFERED BY THE TELCO TO THE CUSTOMER')
    if prediction == 1 and instance['MultipleLines'] == 'Yes' and instance['Dependents'] == 'No' \
            and instance['Partner'] == 'No':
        print('RECOMMEND SINGLE LINE PLANS CURRENTLY OFFERED BY THE TELCO TO THE CUSTOMER')
    if prediction == 1 and instance['PhoneService'] == 'No':
        print('RECOMMEND PHONE SERVICE PACKAGES (VOIP) OFFERED BY THE TELCO')
    if prediction == 1 and instance['TechSupport'] == 'No':
        print('RECOMMEND TECH SUPPORT PACKAGES OFFERED BY THE TELCO TO THE CUSTOMER')
    if prediction == 1 and instance['InternetService'] == 'No':
        print('RECOMMEND INTERNET SERVICE PLANS (FIBRE OPTIC/DSL) OFFERED BY THE TELCO TO THE CUSTOMER')
    if prediction == 1 and instance['OnlineSecurity'] == 'No':
        print('RECOMMEND ONLINE SECURITY OPTIONS OFFERED BY THE TELCO TO THE CUSTOMER')
    if prediction == 1 and instance['Contract'] == 'Month-to-month':
        print('RECOMMEND LONGER CONTRACT PERIOD (ONE YEAR/ TWO YEAR) TO THE CUSTOMER')
    if prediction == 1 and instance['PaymentMethod'] == 'Electronic check' \
            or instance['PaymentMethod'] == 'Mailed check':
        print('RECOMMEND CREDIT CARD AS AN ALTERNATIVE PAYMENT METHOD TO THE CUSTOMER')


corrective_action_suggestion()
```

Figure 5. 16 Corrective Action Suggestion

5.3 System Testing

The final customer churn prediction and corrective action suggestion model was embedded into a locally hosted web application. The web application allows users to enter customer data and receive churn predictions and corrective action suggestions. The developed system was subjected to testing to ensure it satisfied all the functional requirements highlighted in Chapter 4 of this dissertation. The system allowed the user to seamlessly enter customer data via select boxes, view and download entered customer data, and get churn predictions and corrective action suggestions seamlessly with minimal latency. Additionally, the system was subjected to numerous input scenarios with varying entries to test its corrective action suggestions functionality.

Chapter: 6 Discussions

6.1 Introduction

This chapter discusses the study objectives, reviews the research findings and the developed customer churn prediction and corrective action suggestion system. Additionally, the advantages and disadvantages of the developed system along with the challenges encountered during system development are addressed.

6.2 Exploratory Data Analysis

The data pre-processing and exploratory data analysis carried out before the dataset was used in training the classification algorithms was quite extensive as highlighted in Chapter 5 of this dissertation. The data cleaning process revealed that the IBM telco customer churn dataset was of high quality as only one attribute, total charges had missing values. Imputation of the missing values in the total charges column was also quite straightforward as it was easy to derive the relationship between the total charges column, the tenure and the monthly charges column as highlighted in Chapter 5 subsection 5.2.2 of this text. The data cleaning process revealed that most of the dataset attributes were unordered categorical variables which influenced the choice to apply one-hot encoding over ordinal encoding to the categorical variables. Normality testing on the numerical variables showed that none of the numerical variables was normally distributed. All variable transformation techniques that were applied to the numerical variables could not transform their distributions into Gaussian distributions. This greatly influenced the choice of statistical tests employed during the EDA as statistical methods that assumed normal distribution of variables were inapplicable as they could lead to inaccurate inferences from the data.

Additionally, through EDA, the size of the dataset, (7042 rows * 21 columns) was determined. The dataset also influenced the choice of statistical methods since certain statistical tests such as the Shapiro- Wilks test, are sensitive to sample size. Univariate analysis of the target variable showed that the dataset was imbalanced, thus influencing the choice to apply techniques such as stratified shuffle split and SMOTE to mitigate the effects of the class imbalance on model performance. Numerical attributes also varied greatly in absolute value thus they were rescaled to prevent attribute dominance during the learning process. Moreover, discretization was also applied to the three numerical attributes creating additional model inputs. Bivariate analysis of the input features and the target variable revealed that all the input features except gender, phone service and multiple lines had an associative relationship. This influenced the feature engineering process as all the dataset attributes excluding gender, phone

service and multiple lines provided significant information towards the models' classification. However, it was noted that excluding gender, phone service and multiple lines from the models' inputs had no significant impact on model performance with recall, precision and F1 scores largely remaining stagnant. Summarily, EDA proved to be vital in gaining insights into the patterns and relationships within the dataset and played a pivotal role in the feature engineering and selection process.

6.3 Model Evaluation Metrics

The development of the customer churn prediction and corrective action suggestion model entailed training several classification algorithms before selecting the best-performing algorithm for final system implementation. It was necessary to employ the appropriate evaluation metrics in selecting the best-performing model. There are several model evaluation metrics as highlighted in Chapter 3 subsection 3.4.4 of this text. The choice of evaluation metric is primarily influenced by the dataset quality and model purpose. Accuracy gauges model performance based on the total number of correct predictions (true positives and true negatives) against the total number of predictions. Although accuracy is quite straightforward, it is not particularly informative, especially in cases where the dataset is imbalanced. For imbalanced datasets, the model's accuracy will be high as long as it correctly classifies the majority class. This may not always be ideal, especially in instances in which the costs of a false negative in the minority class are high. Conversely, precision gives more weight to false positives while recall places more emphasis on false negatives. The harmonic mean between recall and precision is the F1 score, which provides a balance between precision and recall as generally as precision increases, recall decreases and vice versa. Table 6.1 below highlights the meanings of various keywords in the context of the developed churn prediction model.

Table 6. 1 Definition of Terms

Keyword	Meaning
True Positive	The model correctly predicts that a customer will churn and the customer churns
False Positive	The model incorrectly predicts that a customer will churn but the customer does not churn
True Negative	The model correctly predicts that a customer will not churn and the customer does not churn
False Negative	The model incorrectly predicts that a customer will not churn but the customer churns

The developed system aims to predict churners and suggest the appropriate corrective actions. Therefore, the system must effectively identify actual churners to enhance its usability. The cost of false negatives is significantly high as the primary goal of the system is to identify potential churners. The primary performance metric considered when evaluating the

classification algorithms' performance before selection for model implementation was their recall. Additional considerations included the computational efficiency(model execution time and grid search time), and the ease of model interpretation. The performance metrics for the various classification algorithms are highlighted in the figures below.

```
Mean Accuracy: 0.78708303761533
Mean Precision: 0.5837927170038243
Mean Recall: 0.6935828877005348
Mean F1-score: 0.6337904627096498
Mean AUC: 0.757226226458963

Final Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.82      0.85      1035
           1       0.58      0.67      0.62       374

    accuracy                           0.78      1409
   macro avg       0.73      0.75      0.73      1409
weighted avg       0.80      0.78      0.79      1409


Final Confusion Matrix:
[[849.6 185.4]
 [114.6 259.4]]
CatBoost execution time in seconds: 54.58241081237793 seconds
```

Figure 6. 1 CatBoost Metrics

```
Mean Accuracy: 0.7349893541518808
Mean Precision: 0.5008120165689204
Mean Recall: 0.7802139037433155
Mean F1-score: 0.6099073850523825
Mean AUC: 0.7494306233692424

Final Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.73      0.80      1035
           1       0.51      0.77      0.61       374

    accuracy                           0.74      1409
   macro avg       0.70      0.75      0.71      1409
weighted avg       0.79      0.74      0.75      1409


Final Confusion Matrix:
[[743.8 291.2]
 [ 82.2 291.8]]
Complement Naive Bayes execution time in seconds: 0.2375946044921875 seconds

Process finished with exit code 0
```

Figure 6. 2 Complement Naive Bayes Metrics

```
Mean Accuracy: 0.6983676366217175
Mean Precision: 0.4627682724382101
Mean Recall: 0.8449197860962567
Mean F1-score: 0.597980298655116
Mean AUC: 0.745165207057790 2

Final Classification Report:
                precision    recall  f1-score   support

             0       0.92      0.63      0.75      1035
             1       0.45      0.84      0.59       374

    accuracy                           0.69      1409
   macro avg       0.69      0.74      0.67      1409
weighted avg       0.79      0.69      0.71      1409


Final Confusion Matrix:
[[668.  367.]
 [ 58.  316.]]
K-NN execution time in seconds: 0.4312264919281006 seconds
```

Figure 6. 3 K-NN Metrics

```
Mean Accuracy: 0.7481902058197303
Mean Precision: 0.5170823362419312
Mean Recall: 0.7893048128342246
Mean F1-score: 0.6247166225956774
Mean AUC: 0.7613190730837789

Final Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.73      0.81      1035
           1       0.51      0.80      0.63       374

    accuracy                           0.75      1409
   macro avg       0.71      0.76      0.72      1409
weighted avg       0.81      0.75      0.76      1409


Final Confusion Matrix:
[[759.  276. ]
 [ 78.8 295.2]]
Logistic Regression execution time in seconds: 1.0281493663787842 seconds
```

Figure 6. 4 Logistic Regression Metrics

```
Mean Accuracy: 0.7403832505322924
Mean Precision: 0.5075828615701719
Mean Recall: 0.7727272727272727
Mean F1-score: 0.6125376562388261
Mean AUC: 0.750711462450593

Final Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.72      0.80      1035
           1       0.49      0.76      0.60       374

    accuracy                           0.73      1409
   macro avg       0.69      0.74      0.70      1409
weighted avg       0.79      0.73      0.74      1409


Final Confusion Matrix:
[[754.2 280.8]
 [ 85.  289. ]]
Gaussian Naive Bayes execution time in seconds: 0.25042271614074707 seconds
```

Figure 6. 5 Gaussian Naive Bayes Metrics

```
Mean Accuracy: 0.7506032647267565
Mean Precision: 0.5220804625235113
Mean Recall: 0.7262032085561498
Mean F1-score: 0.6073080461640785
Mean AUC: 0.7428117492056111

Final Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.75      0.82      1035
           1       0.53      0.75      0.62       374

    accuracy                           0.75      1409
   macro avg       0.71      0.75      0.72      1409
weighted avg       0.80      0.75      0.77      1409


Final Confusion Matrix:
[[786.  249. ]
 [102.4 271.6]]
Random Forest Classifier execution time in seconds: 9.931707382202148 seconds
```

Figure 6. 6 Random Forest Classifier Metrics

Figures 6.1 to 6.6 above highlight the performance metrics of the models evaluated for implementing the churn prediction and corrective action suggestion system. As discussed earlier, due to the high cost of false negatives, recall was given the most weight when evaluating the models. Table 6.2 below outlines the various performance metrics of the implemented classification algorithms. Although K-NN had the highest recall, it had the lowest precision, accuracy, F1 score and AUC score. Conversely, the Logistic Regression classifier had the second-highest recall of 0.80. Additionally, it achieved the highest precision and AUC scores and the second-highest accuracy. Generally, the logistic regression classifier achieved better metrics across board compared to the other classification algorithms. Furthermore, Logistic Regression classifiers are easily interpretable especially compared to ensemble algorithms such as the Random Forest Classifier or gradient boosting algorithms such as the CatBoost classifier. Logistic regression feature coefficients provide a clear, precise and concise way for model interpretation and feature importance as highlighted in Chapter 4 of this dissertation.

Table 6. 2 Model Performance Metrics

Model	Recall	Accuracy	Precision	F1 Score	AUC	Execution Time
Logistic Regression	**0.80**	**0.75**	**0.51**	**0.63**	**0.761**	**1.02s**
K-NN	**0.84**	**0.69**	**0.41**	**0.59**	**0.7451**	**0.43s**
CatBoost	**0.67**	**0.78**	**0.58**	**0.62**	**0.757**	**54.585s**
Gaussian Naive Bayes	**0.76**	**0.73**	**0.49**	**0.60**	**0.7507**	**0.25s**
Complement Naive Bayes	**0.77**	**0.74**	**0.51**	**0.61**	**0.749**	**0.23s**
Random Forest	**0.75**	**0.75**	**0.53**	**0.62**	**0.7428**	**9.93s**

Logistic Regression models produce linear decision boundaries, making it straightforward to understand how each feature contributes to the classification outcome. Additionally, they provide probabilistic interpretations of the classification outcomes through the logistic function. The predicted probabilities represent the likelihood of each class, allowing for a more intuitive understanding of the model's predictions. Unlike Bayesian classifiers, Logistic Regression does not assume independence between features. This allows Logistic Regression to capture more complex relationships between variables, providing a more accurate representation of the data. K-NN classifiers are based on distance metrics relying on the nearest neighbours class labels and do not provide clear insights into how the decision boundary was formed. Both K-NN and Bayesian classifiers do not inherently provide feature importance measurement techniques. The execution time was not given a lot of significance in model selection since it could be greatly influenced by other factors such as the number of concurrently running tasks during the model execution and CPU temperatures. However, the hyperparameter tuning for the complex models (Random Forest Classifier and CatBoost) was computationally expensive due to the large hyperparameter space. Therefore, Logistic Regression's superior performance metrics coupled with its interpretability made it the best choice for the development of the telco customer churn prediction and corrective action suggestion model.

Theoretically, ensemble and gradient boosting models were expected to perform better than the classical machine learning algorithms. However, Logistic Regression outperformed both Random Forest and CatBoost as highlighted in Table 6.2 above. This is attributed to logistic regression's simplicity which required less computational resources and thus was optimally tuned. Hyperparameter tuning for both the Random Forest Classifier and the CatBoost Classifier was extremely costly, requiring a lot of computational resources due to their large hyperparameter spaces. For the CatBoost Classifier, due to the high computational cost, hyperparameter tuning was done through manual tuning. Additionally, to fasten the grid search during Random Forest tuning, the hyperparameter space was reduced. Thus, both classifiers were implemented without the optimal set of hyperparameters which explains their poor performance compared to the computationally cheap classical machine learning algorithms.

6.4 Corrective Actions

The procedure for corrective action suggestions was highlighted in Chapter 5 subsection 5.2.5. Logistic Regression classifiers provide model coefficients for all the input

features. However, when determining the corrective actions to be suggested, not all the feature coefficients were taken into account. Input features such as gender, partner, dependents and senior citizen were not considered when developing the corrective actions since if the listed inputs contributed towards a churn classification, it was not viable to recommend the converse as a corrective action. For example, based on the coefficients, having a partner contributes more towards a negative churn classification (non-churner), thus if the model classified a customer as a potential churner, the appropriate corrective action suggestions would entail recommending that the customer get a partner if they did not have one. However, this is not a logical recommendation as it is not easily enforceable and thus cannot be suggested. Hence gender, partner, dependents and senior citizen were excluded from the corrective action suggestions.

Furthermore, input features streaming movies and streaming TV were also excluded from the corrective action suggestions. This is because both StreamingMovies_No and StreamingTV_No had lower coefficients than their alternative options as shown in Figure 6.7 below. The coefficients indicate that customers with no streaming services are less likely to churn. However, it is not logical for a telecommunications company to recommend to a customer to subscribe to fewer of its services as this goes against its primary bottom line.

```
StreamingMovies_No: 0.04125110097514607 4
StreamingMovies_Yes: 0.25003544983190984
Dependents_Yes: -0.267846791548364 1
DeviceProtection_No: 0.20393740515098044
DeviceProtection_Yes: 0.08734914565586964
StreamingTV_No: 0.015038551899257149
StreamingTV_Yes: 0.27624799890789936
```

Figure 6. 7 Streaming Coefficients

6.5 Advantages of the Tool

The advantages of the developed customer churn prediction and corrective action suggestion system include;

i. The tool not only predicts the likelihood of customer churn but also suggests an appropriate corrective action which if implemented may aid in preventing the churn from occurring. Existing churn prediction models only predict the churn but do not offer any steps that may prevent the churn from occurring.

ii. Unlike the works reviewed in Chapter 2, the system provides an interface for users to seamlessly enter customer data and get churn predictions and corrective action suggestions.

6.6 Limitations of the Tool

The shortcomings of the developed tool include;

i. The developed system is locally hosted, meaning it cannot be readily accessed via the internet limiting the number of people who may potentially use it.
ii. The dataset used in training the models was not locally sourced. Although most of the dataset's features are transferrable to a local scenario, some features are not applicable in the Kenyan context. This limits the local use of the developed tool.
iii. It is also not possible to test the effectiveness of the suggested corrective actions without additional data.

6.7 Challenges Encountered

The primary challenge encountered during the implementation of the system was insufficient computational resources. Hyperparameter tuning is a necessary step in optimizing the performance of machine learning algorithms. However, it is computationally expensive and thus the hyperparameter spaces had to be reduced. For the Random Forest Classifier, even with the reduced hyperparameter space, the grid search took 54062 seconds, roughly 15 hours as shown in Figure 6.8 below. Additionally, for the CatBoost classifier, the grid search was too computationally demanding that a manual search was implemented instead. This means that the optimal performance for both classifiers was not achieved. Secondly, due to the nature of the problem, churn prediction, most of the available datasets will always be imbalanced i.e. more non-churners compared to churners due to the inherent nature of successful businesses. Although several techniques were applied to address the class imbalance, it was still evident from the classification reports highlighted in Figures 6.1 to 6.6 of this chapter that all the models achieved significantly better scores with the majority class (0, non-churners). Working with imbalanced datasets is more complex and demands extra steps and techniques not required with balanced data (Krawczyk, 2016).

```
Best Hyperparameters:
{'class_weight': None, 'criterion': 'entropy', 'max_depth': 32, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 250}
THE EXECUTION TIME FOR THE GRID SEARCH IN SECONDS : 54062.413281202316


Process finished with exit code 0
```

Figure 6. 8 Random Forest Grid Search Duration

Chapter 7: Conclusion and Recommendation

7.1 Conclusion

The advent of big data has greatly revolutionized the machine learning and data science space. Machine learning algorithms are applied in numerous day-to-day tasks to simplify them thus making work easier while improving efficiency and effectiveness. The main aim of this study, applying predictive analytics to develop a customer churn prediction and corrective action suggestion system for the telecommunication industry addresses a significant problem in the telecommunications industry. The study commenced by reviewing several approaches to customer churn prediction and mitigation in the telecommunications industry. Several research papers highlighting diverse approaches to predicting churn were reviewed and areas of improvement identified. A conceptual framework was then presented on how to achieve the main objective while filling the gaps identified. Subsequently, various statistical methods were employed to determine relevant model training features for churn prediction and corrective action suggestions. Several classification algorithms were then implemented and the best-performing one was selected for system implementation. The implemented system performed satisfactorily well achieving a recall of 80% and was able to smoothly predict customer churn and recommend appropriate corrective actions. Furthermore, despite recall being prioritised over accuracy during the model training process, the developed system achieved an accuracy score of 75%, four points higher than the accuracy score achieved by the logistic regression model developed in Omonge Jevans' research study *titled A Customer Segmentation Model Using Logistic Regression, a Telkom Kenya Case Study* reviewed in Chapter 2 of this dissertation.

In conclusion, the development of a customer churn prediction and corrective action suggestion tool utilizing predictive analytics presents a promising avenue for improving customer retention within the telecommunications sector. This tool has the potential to offer valuable insights to telecom service providers and marketers operating within the industry's rapidly evolving landscape. In recent years, there has been a growing interest in leveraging artificial intelligence (AI) to anticipate customer churn in various industries, including the telecommunications industry. The high churn rates prevalent in the telecom sector underscore the need for robust predictive models to identify and mitigate customer attrition effectively. The customer churn prediction and corrective action suggestion model developed in this study analysed a range of features, including customer demographics, usage patterns, and service preferences, to accurately forecast churn probabilities, achieving a recall of 80%. This study

lays the groundwork for further research into corrective action suggestions to effectively mitigate the effects of customer churn as merely predicting the churn is not adequate.

7.2 Recommendations

Upon conclusion of the study, the researcher recommends;

i. The application of automatic hyperparameter optimization libraries such as Optuna and Hyperopt. Hyperparameter tuning using grid search proved to be too computationally expensive for ensemble and boosting algorithms. Optimization libraries leverage advanced algorithms such as Bayesian Optimization to intelligently search the hyperparameter space thereby identifying optimal hyperparameter combinations more efficiently.

ii. The utilization of algorithm-level methods such as SMOTEBoost and AdaBoost to address the class imbalance within the data.

7.3 Future Work

This study was implemented using the IBM telco customer churn dataset. Although some of the insights gained apply to all telecommunications companies generally, future research may focus on using a local dataset to make the work directly transferrable to a Kenyan context. Furthermore, in determining the corrective action suggestions, more robust model-agnostic feature importance measurement techniques such as Shapley values, LIME and ELI5 (explain like I'm 5) may be used in place of coefficients which work only with logistic regression.

References

Abbasimehr, H., Setak, M., & Soroor, J. (2013). A framework for identification of high-value customers by including social network-based variables for churn prediction using neuro-fuzzy techniques. *International Journal of Production Research*, *51*(4), 1279-1294.

Agha, A. A., Rashid, A., Rasheed, R., Khan, S., & Khan, U. (2021). Antecedents of Customer Loyalty in Telecomm Sector. *Turkish Online Journal of Qualitative Inquiry*, *12*(9).

Ahmad, I., Basheri, M., Iqbal, M. J., & Rahim, A. (2018). Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. *IEEE Access*, *6*, 33789-33795.

Ahuja, R., Chug, A., Gupta, S., Ahuja, P., & Kohli, S. (2020). Classification and clustering algorithms of machine learning with their applications. *Nature-inspired computation in data mining and machine learning*, 225-248.

Akoglu, H. (2018). User's guide to correlation coefficients. *Turkish journal of emergency medicine*, *18*(3), 91-93.

Alpaydin, E. (2020). *Introduction to machine learning*. MIT Press.

Alzubi, J., Nayyar, A., & Kumar, A. (2018, November). Machine learning from theory to algorithms: an overview. In *Journal of Physics: conference series* (Vol. 1142, p. 012012). IOP Publishing

Airtel Africa (2019). Quarterly report on the results for the fourth quarter and year ended March 31, 2019. Retrieved from https://airtel.africa/assets/pdf/pdf1.pdf

Amin, A., Anwar, S., Adnan, A., Nawaz, M., Alawfi, K., Hussain, A., & Huang, K. (2017). Customer churn prediction in the telecommunication sector using a rough set approach. *Neurocomputing*, *237*, 242-254.

Anagaw, A., & Chang, Y. L. (2019). A new complement naïve Bayesian approach for biomedical data classification. *Journal of Ambient Intelligence and Humanized Computing*, *10*, 3889-3897.

Arena, F., & Pau, G. (2020). An overview of big data analysis. *Bulletin of Electrical Engineering and Informatics*, *9*(4), 1646-1653.

Atkinson, A. C., Riani, M., & Corbellini, A. (2021). The box–cox transformation: Review and extensions.

Beaird, J., Walker, A., & George, J. (2020). *The principles of beautiful web design*. SitePoint Pty Ltd.

Baker, S. R., Baugh, B., & Sammon, M. (2023). Customer churn and intangible capital. *Journal of Political Economy Macroeconomics*, *1*(3), 447-505.

Beneke, J., de Sousa, S., Mbuyu, M., & Wickham, B. (2016). The effect of negative online customer reviews on brand equity and purchase intention of consumer electronics in South Africa. *The international review of retail, distribution and consumer research*, *26*(2), 171-201.

Bergsma, W., & Dassios, A. (2014). A consistent test of independence based on a sign covariance related to Kendall's tau. *Bernoulli*, 1006-1028.

Berrar, D. (2019). Bayes' Theorem and Naive Bayes Classifier.

Bharadiya, J. P. (2023). A Comparative Study of Business Intelligence and Artificial Intelligence with Big Data Analytics. *American Journal of Artificial Intelligence*, *7*(1), 24.

Bisong, E. (2019). Logistic regression. *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, 243-250.

Bloomfield, J., & Fisher, M. J. (2019). Quantitative research design. *Journal of the Australasian Rehabilitation Nurses Association*, *22*(2), 27-30.

Bradlow, E. T., Gangwar, M., Kopalle, P., & Voleti, S. (2017). The role of big data and predictive analytics in retailing. *Journal of Retailing*, *93*(1), 79-95.

Bradshaw, T. J., Huemann, Z., Hu, J., & Rahmim, A. (2023). A Guide to Cross-Validation for Artificial Intelligence in Medical Imaging. *Radiology: Artificial Intelligence*, e220232.

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

Canova, S., Cortinovis, D. L., & Ambrogi, F. (2017). How to describe univariate data. *Journal of thoracic disease*, *9*(6), 1741.

Çelik, O., & Osmanoglu, U. O. (2019). Compared to techniques used in customer churn analysis. *Journal of Multidisciplinary Developments*, *4*(1), 30-38.

Chatfield, C. (2018). *Introduction to multivariate analysis*. Routledge.

Chaurasia, V., & Pal, S. (2020). Application of machine learning time series analysis for prediction of COVID-19 pandemic. *Research on Biomedical Engineering*, 1-13.

Chomboon, K., Chujai, P., Teerarassamee, P., Kerdprasop, K., & Kerdprasop, N. (2015, March). An empirical study of distance metrics for k-nearest neighbour algorithm. In *Proceedings of the 3rd international conference on industrial application engineering* (Vol. 2).

Cleff, T., (2019). Univariate Data Analysis. *Applied Statistics and Multivariate Data Analysis for Business and Economics: A Modern Approach Using SPSS, Stata, and Excel*, 27-70.

Curumsing, M. K., Fernando, N., Abdelrazek, M., Vasa, R., Mouzakis, K., & Grundy, J. (2019). Understanding the impact of emotions on software: A case study in requirements gathering and evaluation. *Journal of Systems and Software*, *147*, 215-229.

Dahouda, M. K., & Joe, I. (2021). A deep-learned embedding technique for categorical feature encoding. *IEEE Access*, *9*, 114381-114391.

Daniya, T., Geetha, M., & Kumar, K. S. (2020). Classification and regression trees with Gini index. *Advances in Mathematics: Scientific Journal*, *9*(10), 8237-8247.

Daud, S. N. S. S., Sudirman, R., & Shing, T. W. (2023). Safe-level SMOTE method for handling the class imbalanced problem in electroencephalography dataset of adult anxious state. *Biomedical Signal Processing and Control*, *83*, 104649.

Dennis, A., Wixom, B., & Tegarden, D. (2015). *Systems analysis and design: An object-oriented approach with UML*. John Wiley & sons.

Denis, D. J. (2018). *SPSS data analysis for univariate, bivariate, and multivariate statistics*. John Wiley & Sons.

Derczynski, L. (2016, May). Complementarity, F-score, and NLP Evaluation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* (pp. 261-266).

Dombi, J., & Jónás, T. (2022). Generalizing the sigmoid function using continuous-valued logic. *Fuzzy Sets and Systems*, *449*, 79-99.

Eckerson, W. W. (2007). Predictive analytics. *Extending the Value of Your Data Warehousing Investment. TDWI Best Practices Report*, *1*, 1-36.

Fauzan, R., Siahaan, D., Rochimah, S., & Triandini, E. (2019, July). Use case diagram similarity measurement: A new approach. In *2019 12th International Conference on Information & Communication Technology and Systems (ICTS)* (pp. 3-7). IEEE.

Favaretto, M., De Clercq, E., & Elger, B. S. (2019). Big Data and discrimination: perils, promises, and solutions. A systematic review. *Journal of Big Data*, *6*(1), 1-27.

Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, *61*, 863-905.

Fischer, H. E., Boone, W. J., & Neumann, K. (2014). Quantitative research designs and approaches. In *Handbook of Research on Science Education, Volume II* (pp. 18-37). Routledge.

Frost, S. (2023). *Telecom Industry Growth: Telecom Industry Market Report: Telecom Market*. Store.Frost.com. https://store.frost.com/industries/telecom.html

Fujo, S. W., Subramanian, S., & Khder, M. A. (2022). Customer churn prediction in the telecommunication industry using deep learning. *Information Sciences Letters*, *11*(1), 24

Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International journal of information management*, *35*(2), 137-144.

Gao, X., & Li, G. (2020). A KNN model based on Manhattan distance to identify the SNARE proteins. *Ieee Access*, *8*, 112922-112931.

García, S., Luengo, J., & Herrera, F. (2015). *Data preprocessing in data mining* (Vol. 72, pp. 59-139). Cham, Switzerland: Springer International Publishing.

Garreau, D., & Luxburg, U. (2020, June). Explaining the explainer: A first theoretical analysis of LIME. In *International conference on artificial intelligence and statistics* (pp. 1287-1296). PMLR.

Gold, C., & Tzuo, T. (2020). Interventions that reduce churn. In *Fighting churn with data: The science and strategy of customer retention* (pp. 7–9). essay, Manning Publications Co.

Gomes, H. M., de Mello, R. F., Pfahringer, B., & Bifet, A. (2019, December). Feature scoring using tree-based ensembles for evolving data streams. In *2019 IEEE International Conference on Big Data (Big Data)* (pp. 761-769). IEEE.

González-Estrada, E., & Cosmes, W. (2019). Shapiro–Wilk test for skew normal distributions based on data transformations. *Journal of Statistical Computation and Simulation*, *89*(17), 3258-3272.

Gordini, N., & Veglio, V. (2017). Customers churn prediction and marketing retention strategies. An application of support vector machines based on the AUC parameter-selection technique in the B2B e-commerce industry. *Industrial Marketing Management*, *62*, 100-107.

Hafeez, M. A., Rashid, M., Tariq, H., Abideen, Z. U., Alotaibi, S. S., & Sinky, M. H. (2021). Performance improvement of the decision tree: A robust classifier using a tabu search algorithm. *Applied Sciences*, *11*(15), 6728.

Hamilton, H. (2009). *Overview of decision trees*. Machine Learning/Inductive Inference/DecisionTrees/Overview.https://www2.cs.uregina.ca/~dbd/cs831/notes/ml/dtrees/4_dtrees1.html

Hancock, J. T., & Khoshgoftaar, T. M. (2020). CatBoost for big data: an interdisciplinary review. *Journal of big data*, *7*(1), 94.

Hennink, M., & Kaiser, B. N. (2022). Sample sizes for saturation in qualitative research: A systematic review of empirical tests. *Social science & medicine*, *292*, 114523.

Herland, M., Khoshgoftaar, T. M., & Bauder, R. A. (2018). Big data fraud detection using multiple medicare data sources. *Journal of Big Data*, *5*(1), 1-21.

Hossain, M. R., & Timmer, D. (2021). Machine learning model optimization with hyper parameter tuning approach. *Global Journal of Computer Science and Technology*, *21*(D2), 7-13.

Hwang, H., Jung, T., & Suh, E. (2004). An LTV model and customer segmentation based on customer value: a case study on the wireless telecommunication industry. *Expert systems with applications*, *26*(2), 181-188.

Ishak Latfi, N. S., & Abdullah, S. (2024). The Implementation of The Alexander-Govern Test in Factorial Design Analysis. *Journal of Computational Innovation and Analytics (JCIA)*, *3*(1), 19-36.

Jacobson, L., & Booch, J. R. G. (2021). The unified modelling language reference manual.

Jadhav, S. D., & Channe, H. P. (2016). Comparative study of K-NN, naive Bayes, and decision tree classification techniques. *International Journal of Science and Research (IJSR)*, *5*(1), 1842-1845.

Jaiswal, S. (2021). Retrieved from https://www.javatpoint.com/logistic-regression-in-machine-learning.

Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, *6*(1), 1-54.

Johny, C. P., & Mathai, P. P. (2017). Customer churn prediction: A survey. *International Journal of Advanced Research in Computer Science*, *8*(5), 2178-2181

Joy, T. T., Rana, S., Gupta, S., & Venkatesh, S. (2016, December). Hyperparameter tuning for big data using Bayesian optimisation. In *2016 23rd International Conference on Pattern Recognition (ICPR)* (pp. 2574-2579). IEEE.

Kakulapati, V., Chaitanya, K. K., Chaitanya, K. V. G., & Akshay, P. (2020). Predictive analytics of HR-A machine learning approach. *Journal of Statistics and Management Systems*, *23*(6), 959-969.

Kendall, K. E., & Kendall, J. E. (2014). *Systems analysis and design*. Pearson.

Kesavaraj, G., & Sukumaran, S. (2013, July). A study on classification techniques in data mining. In *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)* (pp. 1-7). IEEE.

Khan, M. T. (2013). Customer loyalty: Concept & definition (a review). *International Journal of Information, Business and Management*, *5*(3), 168-191.

Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, *5*(4), 221-232.

Krithikadatta, J. (2014). Normal distribution. *Journal of Conservative Dentistry and Endodontics*, *17*(1), 96-97.

Kubat, M., & Miroslav, K. (2021). Decision trees. *An Introduction to Machine Learning*, 91-115.

Kuhn, M., & Johnson, K. (2019). *Feature engineering and selection: A practical approach for predictive models*. Chapman and Hall/CRC.

Kumar, P. (2019). Predictive analytics market size, share: Industry report - 2027. Retrieved from https://www.alliedmarketresearch.com/predictive-analytics-market

Kumar, V., Umashankar, N., Kim, K. H., & Bhagwat, Y. (2014). Assessing the influence of economic and customer experience factors on service purchase behaviors. *Marketing Science*, *33*(5), 673-692.

Kwak, S. K., & Kim, J. H. (2017). Statistical data preparation: management of missing values and outliers. *Korean Journal of Anesthesiology*, *70*(4), 407-411.

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

Lin, W. C., & Tsai, C. F. (2020). Missing value imputation: a review and analysis of the literature (2006–2017). *Artificial Intelligence Review*, *53*, 1487-1509.

Lina, R. (2022). Improving Product Quality and Satisfaction as Fundamental Strategies in Strengthening Customer Loyalty. *AKADEMIK: Jurnal Mahasiswa Ekonomi & Bisnis*, *2*(1), 19-26.

Logunova, I. (2022). Retrieved from https://serokell.io/blog/feature-engineering-for-machine-learning

Logunova, I. (2022, June 23). *Random forest classifier: Basic principles and applications*. Guide to Random Forest Classification and Regression Algorithms. https://serokell.io/blog/random-forest-classification

Lu, J. (2002). Predicting customer churn in the telecommunications industry—An application of survival analysis modelling using SAS. *SAS User Group International (SUGI27) Online Proceedings*, *114*, 27.

Ly, T. V., & Son, D. V. T. (2022). Churn prediction in the telecommunication industry using kernel Support Vector Machines. *Plos one*, *17*(5), e0267935.

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, *30*.

Mailagaha Kumbure, M., & Luukka, P. (2022). A generalized fuzzy k-nearest neighbour regression model based on Minkowski distance. *Granular Computing*, *7*(3), 657-671.

Martinez, W. L., Martinez, A. R., & Solka, J. (2017). *Exploratory data analysis with MATLAB*. Chapman and Hall/CRC.

Maulud, D., & Abdulazeez, A. M. (2020). A review on linear regression comprehensive in machine learning. *Journal of Applied Science and Technology Trends*, *1*(4), 140-147.

McCarthy, R. V., McCarthy, M. M., Ceccucci, W., Halawi, L. (2022). *Applying predictive analytics* (pp. 89-121). Springer International Publishing.

McConnell, S. (1996). Rapid Development Microsoft Press. *Redmond, WA*.

Meena, M. E., & Geng, J. (2022). Dynamic competition in Telecommunications: A Systematic Literature Review. *SAGE Open*, *12*(2), 215824402210946. https://doi.org/10.1177/21582440221094609

Mishra, P., Pandey, C. M., Singh, U., Gupta, A., Sahu, C., & Keshri, A. (2019). Descriptive statistics and normality tests for statistical data. *Annals of cardiac anaesthesia*, *22*(1), 67-72.

Mohamed, E. (2020). The relation of artificial intelligence with internet of things: A survey. *Journal of Cybersecurity and Information Management*, *1*(1), 30-24.

Mulyadi, D., & Sinaga, O. (2020). Analysis of Current Ratio, Net Profit Margin, and Good Corporate Governance against Company Value. *Systematic Reviews in Pharmacy*, *11*(1).

Neslin, Scott A., et al. "Defection detection: Measuring and understanding the predictive accuracy of customer churn models." *Journal of marketing research* 43.2 (2006): 204-211.

Obilor, E. I., & Amadi, E. C. (2018). Test for significance of Pearson's correlation coefficient. *International Journal of Innovative Mathematics, Statistics & Energy Policies*, *6*(1), 11-23.

Olorunshola, O. E., & Ogwueleka, F. N. (2022). Review of system development life cycle (SDLC) models for effective application delivery. In *Information and Communication Technology for Competitive Strategies (ICTCS 2020) ICT: Applications and Social Interfaces* (pp. 281-289). Springer Singapore.

Osborne, J. W. (2014). *Best practices in logistic regression*. Sage Publications.

PadaKuu, P. (2023). Retrieved from https://padakuu.com/prototyping-model-598-article

Park, W., & Ahn, H. (2022). Not All Churn Customers Are the Same: Investigating the Effect of Customer Churn Heterogeneity on Customer Value in the Financial Sector: Sustainability, *14*(19), 12328.

Pearson, R. K. (2018). *Exploratory data analysis using R*. Chapman and Hall/CRC.

Pfeifer, P. E. (2005). The optimal ratio of acquisition and retention costs. *Journal of Targeting, measurement, and analysis for marketing*, *13*, 179-188.

PricewaterhouseCoopers. (n.d.). Telecommunications. Retrieved from https://www.pwc.com/ke/en/industries/telecommunications.html

Priyanka, & Kumar, D. (2020). Decision tree classifier: a detailed survey. *International Journal of Information and Decision Sciences*, *12*(3), 246-269.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. *Advances in neural information processing systems*, *31*.

Rangari, K. T. (2021). *Predictive Analysis for Retail Industry Using Big Data Technology* (Doctoral dissertation).

Raschka, S. (2015). *Python machine learning*. Packt publishing ltd.

Ray, S. (2019, February). A quick review of machine learning algorithms. In *2019 International Conference on Machine Learning, Big Data, cloud and Parallel Computing (COMITCon)* (pp. 35-39). IEEE.

Reddy, G. T., Reddy, M. P. K., Lakshmanna, K., Kaluri, R., Rajput, D. S., Srivastava, G., & Baker, T. (2020). Analysis of dimensionality reduction techniques on big data. *Ieee Access*, *8*, 54776-54788.

Reichheld, F. F., & Teal, T. (1996). The loyalty effect: The hidden force behind growth, profits, and lasting. *Harvard Business School Publications, Boston*.

Restum, Y. (2021). Types of churn: Understand the difference. Retrieved from https://blog.zooxsmart.com/types-of-churn-understand-the-difference

Rigatti, S. J. (2017). Random forest. *Journal of Insurance Medicine*, *47*(1), 31-39.

Sangwan, R. S. (2014). *Software and systems architecture in action*. CRC Press.

Safaricom PLC (2019). News Release. Retrieved from https://www.safaricom.co.ke/images/Downloads/Resources_Downloads/FY2019/FY2019_Press_Commentary.pdf

Scornet, E. (2023, February). Trees, forests, and impurity-based variable importance in regression. In *Annales de l'Institut Henri Poincare (B) Probabilites et statistiques* (Vol. 59, No. 1, pp. 21-52). Institut Henri Poincaré.

Shapi, M. K. M., Ramli, N. A., & Awalin, L. J. (2021). Energy consumption prediction by using machine learning for smart building: Case study in Malaysia. *Developments in the Built Environment*, *5*, 100037.

Sharda, R., Delen, D., & Turban, E. (2014). *Business intelligence and analytics: systems for decision support*. Pearson.

Shirazi, F., & Mohammadi, M. (2019). A big data analytics model for customer churn prediction in the retiree segment. *International Journal of Information Management*, *48*, 238-253.

Shwartz-Ziv, R., & Armon, A. (2022). Tabular data: Deep learning is not all you need. *Information Fusion*, *81*, 84-90.

Shekar, B. H., & Dagnew, G. (2019, February). Grid search-based hyperparameter tuning and classification of microarray cancer data. In *2019 second international conference on advanced computational and communication paradigms (ICACCP)* (pp. 1-8). IEEE.

Silva, A. J., Cortez, P., Pereira, C., & Pilastri, A. (2021). Business analytics in Industry 4.0: A systematic review. *Expert systems*, *38*(7), e12741.

Simpson, A. J. (2015). Over-sampling in a deep neural network. *arXiv preprint arXiv:1502.03648*.

Singh, A., Thakur, N., & Sharma, A. (2016, March). A review of supervised machine learning algorithms. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 1310-1315). Ieee.

Sun, Y., Que, H., Cai, Q., Zhao, J., Li, J., Kong, Z., & Wang, S. (2022). Borderline smote algorithm and feature selection-based network anomalies detection strategy. *Energies*, *15*(13), 4751.

Sundararajan, M., & Najmi, A. (2020, November). The many Shapley values for model explanation. In *International conference on machine learning* (pp. 9269-9278). PMLR.

Tallón-Ballesteros, A., & Chen, C. (2020). Explainable AI: Using Shapley value to explain complex anomaly detection ML-based systems. *Machine learning and artificial intelligence*, *332*, 152.

Tamaddoni Jahromi, A., Stakhovych, S., & Ewing, M. (2016). Comparing churn prediction techniques and assessing their performance: A contingent perspective. Journal of Service Research, 19(2), 123–141. http://dx.doi.org/10.1177/1094670515616376.

Taunk, K., De, S., Verma, S., & Swetapadma, A. (2019, May). A brief review of nearest neighbour algorithm for learning and classification. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)* (pp. 1255-1260). IEEE.

Tessitore, S. (2023, May 16). *What's the average churn rate by industry?* CustomerGauge. https://customergauge.com/blog/average-churn-rate-by-industry

Thukral, A. K., Bhardwaj, R., Kumar, V., & Sharma, A. (2019). New indices regarding the dominance and diversity of communities, derived from sample variance and standard deviation. *Heliyon*, *5*(10).

Umayaparvathi, V., & Iyakutti, K. (2016). A survey on customer churn prediction in telecom industry: Datasets, methods, and metrics. *International Research Journal of Engineering and Technology (IRJET)*, *3*(04).

Vanschoren, J. (2019, October 15). *Telco-customer-churn*. OpenML. https://api.openml.org/d/42178

Verma, B. (2017). Chi-Square Test for Independence.

Vilhomaa, L. (2022). *Assessing effects of data quality on newly proposed scoring methods for churn management: a simulation study* (Master's thesis, Hanken School of Economics).

Wall Emerson, R. (2023). Mann-Whitney U test and t-test. *Journal of Visual Impairment & Blindness*, *117*(1), 99-100.

Wei, W., Li, J., Cao, L., Ou, Y., & Chen, J. (2013). Effective detection of sophisticated online banking fraud on extremely imbalanced data. *World Wide Web*, *16*, 449-475.

Weiss, G. M. (2013). Foundations of imbalanced learning. *Imbalanced learning: Foundations, algorithms, and applications*, 13-41.

Wiley, J. (2010). Retrieved from https://catalogimages.wiley.com/images/db/pdf/0471073229.ch1.pdf

Winter, E. (2002). The Shapley value. *Handbook of game theory with economic applications*, *3*, 2025-2054.

Wong, K. K. K. (2010). Fighting churn with rate plan right-sizing: A customer retention strategy for the wireless telecommunications industry. *The Service Industries Journal*, *30*(13), 2261-2271.

Wong, T. T., & Yeh, P. Y. (2019). Reliable accuracy estimates from k-fold cross-validation. *IEEE Transactions on Knowledge and Data Engineering*, *32*(8), 1586-1594.

Xiao, C., Ye, J., Esteves, R. M., & Rong, C. (2016). Using Spearman's correlation coefficients for exploratory data analysis on a big dataset. *Concurrency and Computation: Practice and Experience*, *28*(14), 3866-3878.

Yalcin, A. S., Kilic, H. S., & Delen, D. (2022). The use of multi-criteria decision-making methods in business analytics: A comprehensive literature review. *Technological forecasting and social change*, *174*, 121193.

Yang, F. J. (2018, December). An implementation of naive Bayes classifier. In *2018 International Conference on Computational Science and Computational Intelligence (CSCI)* (pp. 301-306). IEEE.

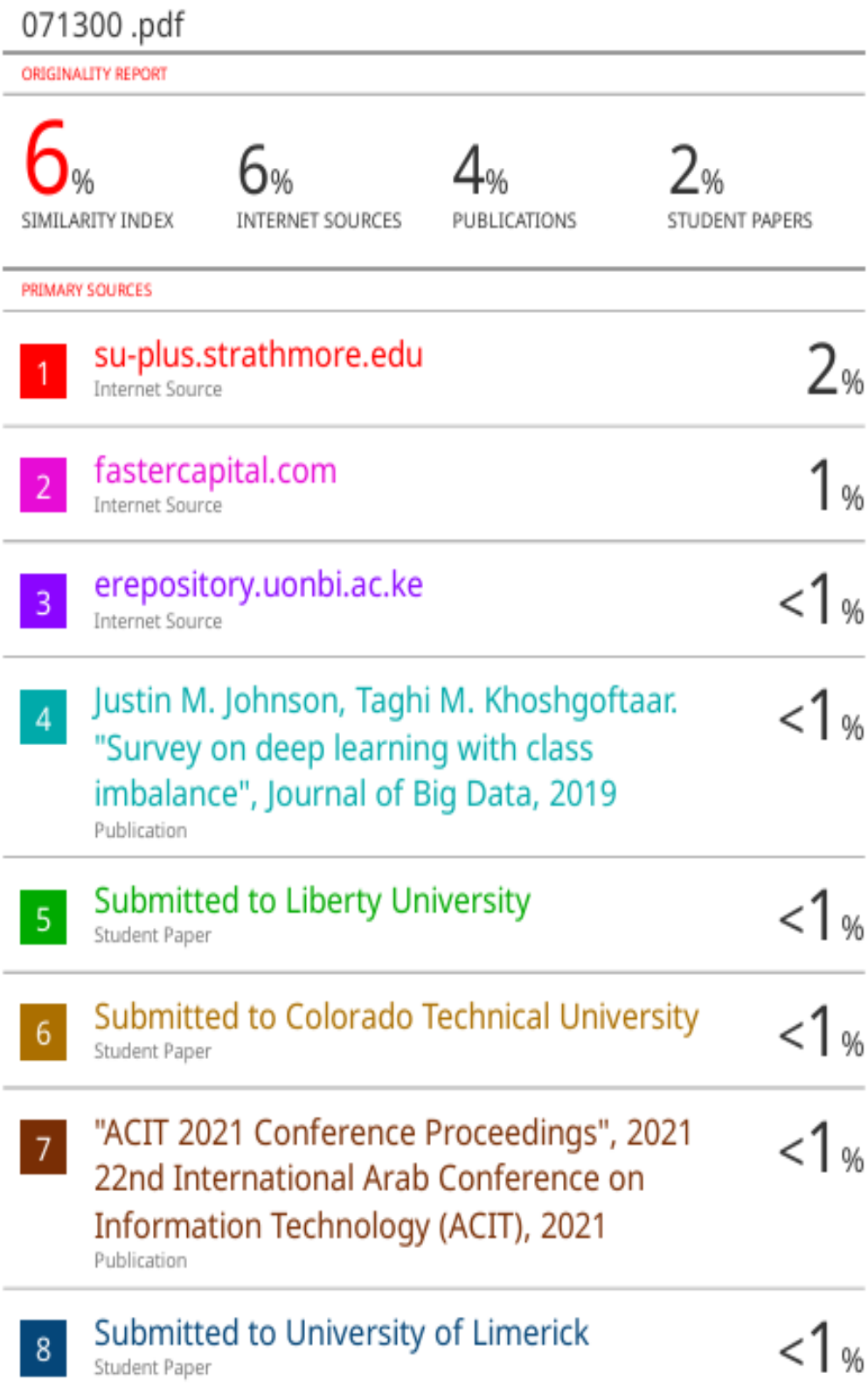
Yau, L. Q., Shah, M. H., Ang, E. J. Y., Rameshkumar, D., Chee, J. S., Lam, Y. T., & Gupta, P. (2023). MALPRED: Predictive Modeling for Malware Detection in Windows Systems using Ensemble Learning.

Yu, L., Zhou, R., Chen, R., & Lai, K. K. (2022). Missing data preprocessing in credit classification: One-hot encoding or imputation? *Emerging Markets Finance and Trade*, *58*(2), 472-482

Zhao, M., Zeng, Q., Chang, M., Tong, Q., & Su, J. (2021). A prediction model of customer churn considering customer value: an empirical research of telecom industry in China. *Discrete Dynamics in Nature and Society*, *2021*, 1-12.

Appendices

Appendix A: Turnitin Report



071300 .pdf

ORIGINALITY REPORT

6%	6%	4%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	su-plus.strathmore.edu Internet Source	2%
2	fastercapital.com Internet Source	1%
3	erepository.uonbi.ac.ke Internet Source	<1%
4	Justin M. Johnson, Taghi M. Khoshgoftaar. "Survey on deep learning with class imbalance", Journal of Big Data, 2019 Publication	<1%
5	Submitted to Liberty University Student Paper	<1%
6	Submitted to Colorado Technical University Student Paper	<1%
7	"ACIT 2021 Conference Proceedings", 2021 22nd International Arab Conference on Information Technology (ACIT), 2021 Publication	<1%
8	Submitted to University of Limerick Student Paper	<1%

Appendix B: Ethical Approval

Strathmore UNIVERSITY

28th November 2023

Mr Wanda Kim Rit-wane,
Ritwane.Wanda@strathmore.edu

Dear Mr Wanda,

RE: A Customer Churn Prediction and Corrective Action Suggestion Model for the Telecommunications Industry Using Predictive Analytics

This is to inform you that SU-ISERC has reviewed and **approved** your above SU-masters research proposal. Your application reference number is **SU-ISERC1917/23**. The approval period is from **28th November 2023 to 27th November 2024.**

This approval is subject to compliance with the following requirements:

i. Only approved documents including (informed consents, study instruments, MTA) will be used.
ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by SU-ISERC.
iii. Death and life-threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to SU-ISERC within 72 hours of notification.
iv. Any changes anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to SU-ISERC within 72 hours.
v. Clearance for the export of biological specimens must be obtained from relevant institutions.
vi. Submission of a request for renewal of approval at least 60 days prior to the expiry of the approval period. Attach a comprehensive progress report to support the renewal.
vii. Submission of an executive summary report within 90 days of completion of the study to SU-ISERC.

Before commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology, and Innovation (NACOSTI) https://research-portal.nacosti.go.ke/ and obtain other clearances needed.

Yours sincerely,

Mr Ambrose Rachier,
Chairperson; SU-ISERC

STRATHMORE UNIVERSITY INSTITUTIONAL SCIENTIFIC AND ETHICAL REVIEW COMMITTEE
(SU-ISERC)
28-Nov-2023
Email:ethicsreview@strathmore.edu
P.O BOX 59857-00200
NAIROBI-KENYA

Ole Sangale Rd, Madaraka Estate. PO Box 59857-00200, Nairobi, Kenya. Tel +254 (0)703 034000
Email admissions@strathmore.edu www.strathmore.edu

Appendix C: Code

```
"""
CUSTOMER CHURN PREDICTION AND CORRECTIVE ACTION SUGGESTION MODEL BY KIM
RIT-WANE WANDA.
SCHOOL OF ENGINEERING & COMPUTING SCIENCES, STRATHMORE UNIVERSITY.
"""

# IMPORTATION OF LIBRARIES
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from scipy.stats import anderson

#  READING THE ORIGINAL OPENML.ORG TELCO CHURN DATASET
telco_dataset =
pd.read_csv("C:\\Users\\ASUS\\Desktop\\Project\\datasets\\Original
openml.org telco churn dataset.csv")
pd.options.display.width = 0  # makes the run window cover the entire page
pd.set_option('display.max_columns', None)  # displays all the columns from
the dataset
print(telco_dataset.head(5))
print(telco_dataset.info())

#  CHECKING FOR DUPLICATE ROWS
duplicate_rows = [telco_dataset.duplicated()]
print(duplicate_rows)  # returns boolean True if there are duplicated rows,
False if none

#  DROPPING THE CUSTOMER ID COLUMN
del telco_dataset['customerID']

#  FINDING ALL THE ATTRIBUTES IN OUR DATASET OF TYPE OBJECT
for attribute in telco_dataset.columns:
    if telco_dataset[attribute].dtype == 'object':
        print(attribute)

#  FINDING THE NUMBER OF UNIQUE ENTRIES IN EACH COLUMN
unique_counts = telco_dataset.nunique()
for col, count in unique_counts.items():
    print(f"{col}: {count}")

#  FINDING MISSING VALUES IN OUR DATASET
#  missing/undefined values in Python are represented using NaN(not a
number).
#  The isnull() or isna() methods find missing values in our dataset and
produce a table, with the entry true is
#  NaN or False otherwise.
# missing_values = telco_dataset.isnull()
missing_values = telco_dataset.isna().sum()
#  The sum function sums the number of missing values for each attribute in
our dataset.
print(missing_values)
#  percentage of missing values
missing_values_percentage = telco_dataset.isnull().sum() /
len(telco_dataset) * 100
print(missing_values_percentage)

#  FINDING MISSING VALUES IN THE TOTAL CHARGES COLUMN
```

```
#  finding the missing values in a particular column.
column_missing = 'TotalCharges'
missing_values = telco_dataset['TotalCharges'].isnull()
print(f'the telco customer churn {column_missing} column has
{missing_values} missing values')

#  GETTING THE CATEGORICAL COLUMNS ONLY FROM OUR DATASET
categorical_attributes = set(telco_dataset.columns) -
set(telco_dataset._get_numeric_data().columns)
# the _get_numeric_data method returns columns of numeric datatype only.
print(categorical_attributes)

#  GETTING THE MISSING VALUES IN THE TOTAL CHARGES COLUMN
total_charge = telco_dataset['TotalCharges']
missing = total_charge[~total_charge.str.replace(".", "").str.isdigit()]
print(missing)

#  CONVERTING THE DATATYPE OF THE TOTAL CHARGES COLUMN TO NUMERIC
telco_dataset['TotalCharges'] =
pd.to_numeric(telco_dataset['TotalCharges'], errors='coerce')
# print(telco_dataset.info())
# Printing the columns with the missing total charges dataset
indexes_to_print = [488, 753, 936, 1082, 1340, 3331, 3826, 4380, 5218,
6670, 6754]
print(telco_dataset.iloc[indexes_to_print])

#  FILLING THE MISSING VALUES(NaN) IN TOTAL CHARGES COLUMN WITH ZERO
telco_dataset.loc[missing.index, 'TotalCharges'] = 0
#
telco_dataset.to_csv("C:\\Users\\ASUS\\Desktop\\Project\\datasets\\telco_da
taset_TotalCharges_Filled.csv",
# index=False)
print(telco_dataset['TotalCharges'].isnull().sum())
print(telco_dataset.info())

# Printing all the columns where tenure == 1
rows_with_tenure_one = telco_dataset[telco_dataset['tenure'] == 1]
print(rows_with_tenure_one)

#  CONVERTING THE DATATYPE OF THE TOTAL CHARGES COLUMN TO NUMERIC
telco_dataset['TotalCharges'] =
pd.to_numeric(telco_dataset['TotalCharges'], errors='coerce')
print(telco_dataset.info())

# SUMMARY STATISTICS
print('The summary statistics of the telco dataset are: \n',
      telco_dataset[['tenure', 'MonthlyCharges',
'TotalCharges']].describe().T)


#  BINNING (TOTAL CHARGES, MONTHLY CHARGES, TENURE)
def discretization_attribute(attribute):
    plt.hist(telco_dataset[attribute])
    plt.xlabel(f'{attribute.title()}')
    plt.ylabel('attribute_count')
    plt.title(f'{attribute.title()} Bins')
    plt.show()
    bins = np.linspace(telco_dataset[attribute].min(),
telco_dataset[attribute].max(), 4)

    print('** VALUE RANGE**')
```

```
    print(f"Low ({bins[0] : .2f} - {bins[1]: .2f})")
    print(f"Medium ({bins[1]: .2f} - {bins[2]: .2f})")
    print(f"High ({bins[2]: .2f} - {bins[3]: .2f})")

    group_names = ['Low', 'Medium', 'High']
    telco_dataset.insert(telco_dataset.shape[1] - 1, f'{attribute}-binned',
                         pd.cut(telco_dataset[attribute], bins,
labels=group_names, include_lowest=True))
    print(telco_dataset[[attribute, f'{attribute}-binned']].head(10))

    # count values
    print('BINNING DISTRIBUTION')
    print(telco_dataset[f'{attribute}-binned'].value_counts())

    #  plotting the distribution of each bin
    plt.bar(group_names, telco_dataset[f'{attribute}-
binned'].value_counts())

    # set x/y labels and plot title
    plt.xlabel(f"{attribute.title()}")
    plt.ylabel("Count")
    plt.title(f"{attribute.title()} Bins")
    plt.show()


discretization_attribute('tenure')
discretization_attribute('MonthlyCharges')
discretization_attribute('TotalCharges')

#  CONVERTING THE BINNED COLUMNS INTO CATEGORICAL DATA TYPES
telco_dataset[['tenure-binned', 'MonthlyCharges-binned', 'TotalCharges-
binned']] = \
    telco_dataset[['tenure-binned', 'MonthlyCharges-binned', 'TotalCharges-
binned']].astype('category')
telco_dataset[['gender', 'Partner', 'Dependents', 'PhoneService',
'MultipleLines', 'InternetService']] = \
    telco_dataset[['gender', 'Partner', 'Dependents', 'PhoneService',
'MultipleLines', 'InternetService']].astype \
        ('category')
telco_dataset[['OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
'TechSupport', 'StreamingTV', 'StreamingMovies']] \
    = telco_dataset[['OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
'TechSupport', 'StreamingTV',
                     'StreamingMovies']].astype('category')
telco_dataset[['Contract', 'PaperlessBilling', 'PaymentMethod', 'Churn']] =
\
    telco_dataset[['Contract', 'PaperlessBilling', 'PaymentMethod',
'Churn']].astype('category')
print(telco_dataset.info())
#
telco_dataset.to_csv("C:\\Users\\ASUS\\Desktop\\Project\\datasets\\Final_da
taset_Index.csv", index=True)
telco_dataset.to_csv("C:\\Users\\ASUS\\Desktop\\Project\\datasets\\Final_da
taset.csv", index=False)


# NORMALITY TESTING
# H0: the sample has a Gaussian distribution.
# H1: the sample does not have a Gaussian distribution.
# NB: we can not perform the Shapiro-Wilk Test because the sample size is>
5000 and for this test p-value may not be
```

```
# accurate for N > 5000
def anderson_darling(attribute):
    result = anderson(telco_dataset[attribute], dist='norm')
    print(f"Statistic: {result.statistic: .5f}")
    print(f"Critical Values: {result.critical_values}")
    print(f"Significance Levels: {result.significance_level}")
    # Interpret
    alpha = 0.05
    if result.statistic < result.critical_values[2]:
        print(f"{attribute.title()} Sample looks Gaussian (fail to reject
H0)")
    else:
        print(f"{attribute.title()} Sample does not look Gaussian (reject
H0)")


anderson_darling('tenure')
anderson_darling('TotalCharges')
anderson_darling('MonthlyCharges')


# Function to plot the distribution for each categorical column in the
dataset
def plot_count_distribution(df):
    for column in df.columns:
        # Skip columns that are not categorical
        if df[column].dtype != 'object':
            continue

        # Create a catplot for each categorical column
        sns.set_style("whitegrid")
        g = sns.catplot(x=column, kind="count", data=df, hue=column,
palette='Set2', legend=True)

        # Add count labels on top of each bar
        for ax in g.axes.flat:
            for p in ax.patches:
                x = p.get_x() + p.get_width() / 2
                y = p.get_height()
                ax.annotate(f'{int(y)}', (x, y), ha='center', va='bottom',
fontsize=10)

        # Set title and labels
        plt.title(f'{column} Distribution', fontweight='bold')
        plt.xlabel(column, fontweight='bold')
        plt.ylabel('Count', fontweight='bold')
        plt.show()


# Call the function with telco_dataset
plot_count_distribution(telco_dataset)


# BI VARIATE ANALYSIS
# Numerical and Numerical attributes
# H0 The two samples do not have a monotonic relationship
# H1 The two samples have a monotonic relationship
def spearman_correlation_coefficient(attribute1, attribute2):
    alpha = 0.05
    correlation_coefficient, p_value =
stats.spearmanr(telco_dataset[attribute1], telco_dataset[attribute2])
```

```
    print(f'{attribute1} , {attribute2} spearman correlation coefficient:
{correlation_coefficient}')
    if p_value > alpha:
        print(f'{attribute1} and {attribute2} do not have a monotonic
relationship. NULL HYPOTHESIS ACCEPTED!!')
    else:
        print(f'{attribute1} and {attribute2} have a monotonic
relationship. ALTERNATIVE HYPOTHESIS ACCEPTED!!')


spearman_correlation_coefficient('tenure', 'MonthlyCharges')
spearman_correlation_coefficient('tenure', 'TotalCharges')
spearman_correlation_coefficient('MonthlyCharges', 'TotalCharges')


def mannwhitney_correlation(feature1):
    stat, p_value = stats.mannwhitneyu(telco_dataset[feature1],
(telco_dataset['Churn'] == 'Yes').astype(int))
    print(f"Correlation between {feature1} and Churn")
    print('Statistics = %.5f, p = %.5f' % (stat, p_value))

    # interpret the significance
    alpha = 0.05
    if p_value > alpha:
        print('Same distribution (fail to reject H0)')
    else:
        print('Different distribution (reject H0)')
    print('----\n')


numerical_features = ['tenure', 'MonthlyCharges', 'TotalCharges']

for num in numerical_features:
    print(f"Correlation with **{num}**")
    mannwhitney_correlation(num)


# alpha/significance = 0.05
# If p-value <= alpha: significant result, reject null hypothesis (H0),
dependent
# If p-value > alpha: not significant result, fail to reject null
hypothesis (H0), independent

def chi_square_statistic(attribute1, attribute2='Churn'):
    print(f"Correlation between **{attribute1}** and **{attribute2}**")
    crosstab = pd.crosstab(telco_dataset[attribute1],
telco_dataset[attribute2])

    stat, p, dof, expected = stats.chi2_contingency(crosstab,
correction=True)

    print(f'p-value: {p}, degree of freedom: {dof}')
    # print("expected frequencies :\n", expected)

    # interpret test-statistic
    prob = 0.95
    critical = stats.chi2.ppf(prob, dof)
    print('probability=%.3f, critical=%.3f, stat=%.3f' % (prob, critical,
stat))

    if abs(stat) >= critical:
```

```
        print('Dependent (reject H0)')
    else:
        print('Independent (fail to reject H0)')

    # interpret p-value
    # alpha = 1.0 - prob
    #
    # print('significance=%.3f, p=%.3f' % (alpha, p))
    # if p <= alpha:
    #     print('Dependent (reject H0)')
    # else:
    #     print('Independent (fail to reject H0)')
    print('-------------------------------------\n')


# credit: https://machinelearningmastery.com/chi-squared-test-for-machine-learning
print("**Chi-Square Correlation Between Dichotomous Features with Target : Churn**")
dichotomous_cols = telco_dataset[['gender', 'Partner', 'Dependents', 'PaperlessBilling', 'PhoneService']]
for col in dichotomous_cols:
    chi_square_statistic(col)


def cramers_v(x, y):
    """ calculate Cramers V statistic for categorial-categorial association.
      uses correction from Bergsma and Wicher,
      Journal of the Korean Statistical Society 42 (2013): 323-328
    """
    confusion_matrix = pd.crosstab(x, y)
    chi2 = stats.chi2_contingency(confusion_matrix)[0]
    n = confusion_matrix.sum().sum()
    phi2 = chi2 / n
    r, k = confusion_matrix.shape
    phi2corr = max(0, phi2 - ((k - 1) * (r - 1)) / (n - 1))
    rcorr = r - ((r - 1) ** 2) / (n - 1)
    kcorr = k - ((k - 1) ** 2) / (n - 1)
    return np.sqrt(phi2corr / min((kcorr - 1), (rcorr - 1)))


# credit: https://stackoverflow.com/a/46498792/11105356
print("**Correlation Between Polytomous Features with Target : Churn**")
cramer_v_val_dict = {}
polytomous_cols = telco_dataset[
    ['MultipleLines', 'InternetService', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport',
     'StreamingTV', 'StreamingMovies', 'Contract', 'PaymentMethod']]
for col in polytomous_cols:
    cramer_v_val_dict[col] = cramers_v(telco_dataset[col], telco_dataset['Churn'])

cramer_v_val_dict_sorted = sorted(cramer_v_val_dict.items(), key=lambda x: x[1], reverse=True)

for k, v in cramer_v_val_dict_sorted:
    print(k, v)
```

```
import time
import joblib
import pandas as pd
import numpy as np
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import StratifiedShuffleSplit
from sklearn.preprocessing import OneHotEncoder, RobustScaler, LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score,
f1_score, roc_auc_score, confusion_matrix, \
    classification_report

start_time = time.time()
# Load the dataset
telco_dataset =
pd.read_csv("C:\\Users\\ASUS\\Desktop\\Project\\datasets\\telco_dataset_cle
aned.csv")

# Separate features and target variable
X = telco_dataset.drop(columns=['Churn'])
# X = telco_dataset.drop(columns=['Churn', 'gender', 'PhoneService',
'MultipleLines'])
y = telco_dataset['Churn'].values

# One-hot encode categorical columns
cat_cols = list(set(X.columns) - set(X._get_numeric_data().columns))
ohe = OneHotEncoder(drop='first')  # Drop the first category to avoid
multicollinearity
X_encoded_cat = ohe.fit_transform(X[cat_cols]).toarray()

# Scale numerical features
num_cols = ['tenure', 'MonthlyCharges', 'TotalCharges']
transformer = RobustScaler()
X_scaled = transformer.fit_transform(X[num_cols])

# Combine encoded categorical features and scaled numerical features
X_encoded = np.hstack((X_encoded_cat, X_scaled))

# Encode the target variable using LabelEncoder
le = LabelEncoder()
y_encoded = le.fit_transform(y)

# Save the preprocessed dataset features
preprocessed_data = pd.DataFrame(X_encoded,
columns=list(ohe.get_feature_names_out()) + num_cols)
preprocessed_data['Churn'] = y
preprocessed_data.to_csv("C:\\Users\\ASUS\\Desktop\\Project\\datasets\\telc
o_dataset_preprocessed2.csv", index=False)

# Print feature names and their order
print("Feature names and their order as passed during fit:")
feature_names = list(ohe.get_feature_names_out()) + num_cols
print(feature_names)

# Applying Stratified ShuffleSplit
sss = StratifiedShuffleSplit(n_splits=5, test_size=0.2, random_state=42)

# Initialize lists to store evaluation metrics
accuracy = []
precision = []
recall = []
```

```
f1 = []
auc = []
conf_matrix_list = []

# Model training and evaluation
model = LogisticRegression(C=1, max_iter=1000, solver='sag')
for train_index, test_index in sss.split(X_encoded, y_encoded):
    X_train, X_test = X_encoded[train_index], X_encoded[test_index]
    y_train, y_test = y_encoded[train_index], y_encoded[test_index]

    # Applying SMOTE to the training dataset only
    sm = SMOTE(random_state=42)
    X_train_res, y_train_res = sm.fit_resample(X_train, y_train)
# param_grid = {
#         'C': [0.1, 1, 10],
#         'max_iter': [1000, 2000],  # Increased number of iterations
further
#         'solver': ['lbfgs', 'liblinear', 'newton-cg', 'sag', 'saga']
# }
#
#  # Create the grid search object
# grid_search = GridSearchCV(LogisticRegression(), param_grid=param_grid)
#
# # Fit the grid search to the data
# grid_search.fit(X_train_res, y_train_res)
#
# # Get the best hyperparameters
# best_params = grid_search.best_params_
#
# # Print the best hyperparameters
# print("Best Hyperparameters:")
# print(best_params)


    model.fit (X_train_res, y_train_res)
#     # Get probabilities of churn (positive class) for each sample in the
test set
#     proba = model.predict_proba(X_test)[:, 1]
#
#     # Calculate precision-recall curve
#     precision, recall, thresholds = precision_recall_curve(y_test, proba)
#
#     # Convert to F-score
#     fscore = (2 * precision * recall) / (precision + recall)
#
#     # Locate the index of the largest f score
#     ix = np.argmax(fscore)
#
#     # Print best threshold and F1 score for the current fold
#     print("Best Threshold for Fold {}: {:.3f}, F1-Score:
{:.3f}".format(i, thresholds[ix], fscore[ix]))
#
#     # Plot the precision-recall curve
#     plt.plot(recall, precision, marker='.', label='Fold {}'.format(i))
#
#     # Plot threshold for the best f1-score
#     plt.scatter(recall[ix], precision[ix], marker='o', color='black',
#                 label='Best Threshold for Fold {}: {:.3f}'.format(i,
thresholds[ix]))
#
# # Plotting the average precision-recall curve
```

```
# plt.xlabel('Recall')
# plt.ylabel('Precision')
# plt.title('Precision-Recall Curve with Best Thresholds')
# plt.legend()
# plt.show()

    pred = model.predict(X_test)

    # Evaluation metrics
    accuracy_fold = accuracy_score(y_test, pred)
    precision_fold = precision_score(y_test, pred)
    recall_fold = recall_score(y_test, pred)
    f1_fold = f1_score(y_test, pred)
    auc_fold = roc_auc_score(y_test, pred)
    conf_matrix_fold = confusion_matrix(y_test, pred)

    accuracy.append(accuracy_fold)
    precision.append(precision_fold)
    recall.append(recall_fold)
    f1.append(f1_fold)
    auc.append(auc_fold)
    conf_matrix_list.append(conf_matrix_fold)

# Calculate mean metrics across all folds
mean_accuracy = np.mean(accuracy)
mean_precision = np.mean(precision)
mean_recall = np.mean(recall)
mean_f1 = np.mean(f1)
mean_auc = np.mean(auc)
mean_conf_matrix = np.mean(conf_matrix_list, axis=0)  # Average confusion
matrices

# Print mean scores
print("\nMean Accuracy:", mean_accuracy)
print("Mean Precision:", mean_precision)
print("Mean Recall:", mean_recall)
print("Mean F1-score:", mean_f1)
print("Mean AUC:", mean_auc)

# Print final classification report and confusion matrix
print("\nFinal Classification Report:")
print(classification_report(y_test, pred))
print("\nFinal Confusion Matrix:")
print(mean_conf_matrix)

end_time = time.time()
execution_time = end_time - start_time
print(f'Logistic Regression execution time in seconds: {execution_time}
seconds')

# Save the trained model
model_path = "C:\\Users\\ASUS\\PycharmProjects\\Saved Models\\LR_model.pkl"
joblib.dump(model, model_path)

# Save the encoding mechanism
encoding_path = "C:\\Users\\ASUS\\PycharmProjects\\Saved
Models\\LREncoder.pkl"
joblib.dump((ohe, le, transformer), encoding_path)

print("Preprocessed dataset features, trained model, and encoding mechanism
saved successfully.")
```