University of Essex

Department of Mathematical Sciences

MA981 DISSERTATION

CUSTOMER CHURN PREDICTION IN TELECOMMUNICATION

ANMOL MAHAJAN

Registeration Number: 2200735

Supervisor: **DR SAIDEH FERDOWSI**

CHAPTER 1

Abstract

The swift growth of the technological industry has transformed the course of doing business. The companies are creating products around the customer needs and focusing on maintaining a long term relationship with their services. The subscription based model is a result of the ongoing technical advancements in the field of commercial use of artificial intelligence. The consumer has now ability to choose from a variety of products and services which makes the organisations uncertain about their long term survival in the consumer space. The study proposes a machine learning based churn prediction model for a subscription based internet services provider in the business-to-customer(B2C) setting focusing on profit maximisation. The dataset used for the analysis is contributed by IBM and is imbalanced with most non-churners. The dataset is resampled using sampling methods like SMOTE and BorderLine resampling method. The random forest classifier showed promising results with accuracy of nearly 96% stating that algorithm can be applied in the required setting to the dataset. The models are evaluated against each other based on their feature importance, confusion matrix, F1-score, AUC score, Precision-Recall and Accuracy Score.

Contents

1 Abstract 2

2 Introduction 5

2.1 Research Question 6

2.2 Scope and Limitation 6

2.3 Overview of Dataset 6

3 Literature Review 8

4 Methodology 13

4.1 Data Preprocessing 13

4.1.1 Missing Value Imputation 15

4.1.2 Statistical Analysis of Customer Churn 15

4.2 Feature Engineering 18

4.3 Model Evaluation Metrics 19

4.4 Model Building 21

4.4.1 Logistic Regression 22

4.4.2 K-Nearest Neighbor Classifier 26

4.4.3 Random Forest Classifier 28

4.4.4 Support Vector Classifier 31

5 A Neural Network Approach 35

6 Profit Maximisation in Churn 37

7 Results 39

7.1 Traditional Classification Algorithms Comparison 39

7.2 Tuning Traditional Machine Learning Models 40

7.2.1 Tuned Logistic Regression 40

7.2.2 Tuned Support Vector Classifier 40

7.2.3 Tuned Random Forest Classifier 41

7.3 Analysis of Neural Network Approach 41

7.4 Analysis of Profit Maximisation in Churn 44

8 Conclusion and Future Work 46

Bibliography 50

A Code for Churn Prediction Analysis 51

A.1 Steps to run code in Google Colab 51

CHAPTER 2

Introduction

Since 2020, I have been working for prominent product based companies where I closely observed the importance of customer feedback for the growth of the business and product development. Given that there are multiple minds working together in an organisation to build services for the market, the consumer feedback is equally important to maintain a long term relationship to achieve profitability and recognition in the industry. Thus customer retention is a major concern for the companies around the world.

The exponential growth of available data has underscored the necessity to automate the extraction of pertinent insights and uncover valuable knowledge from it. Decision-makers have historically relied on their domain expertise and intuition to derive information from diverse data repositories due to the absence of effective tools for automating the extraction of valuable information [1]. One of the research works in this field shares that information fetched from the data stores can be utilised to make suitable decisions in order to expand the business. The expansion has further caused the implementation of data mining techniques and machine learning algorithms to handle abundance of data with efficiency [2].

Taking a reference from the telecommunication industry, smart phones, Internet plans, video games, video calls and normal calls have become a crucial part of life. Having high call rates and Internet plans are detrimental for price-sensitive customers, and therefore, they look out for cheaper alternatives, resulting in customers switching the services. Switching one service provider to another is called Churn [3].

This dissertation implements a machine learning framework on indicators such as customer-type, tenure, contract, device type, churn value, customer support and customer reviews to predict customer churn and to select a model that maximises profitability. The study will compare models implemented using Logistic Regression, KNN, Tree-Based Classifiers, SVC and Neural Networks to predict customer churn in the telecommunication industry.

2.1 Research Question

The following research questions will be investigated through the study:

1. Which traditional algorithm performed relatively well for churn prediction?

2. Which features prompt churn in the dataset?

3. How does the Neural Network algorithm perform against traditional algorithms?

4. Which churning customers are retained?

2.2 Scope and Limitation

The study is limited to the telecommunication industry and the factors studied will be focused around the business to customer churn prediction. The study does not account for the business to business churn. The dataset does not account for customer satisfaction ratings which can be a major factor to decide the company goals at large for customer retention and allow better prediction.

2.3 Overview of Dataset

The IBM Telco Customer Churn [4] sample dataset consists of 33 attributes and 7043 observations. The dataset consists of a combination of numerical and categorical features. The Churn Value feature indicates whether the customer has stopped using the services of the company or not. There are other important features in the dataset that help to understand the churn in detail. The other features include CustomerID, Count, Country, State, City, Zip Code, Lat Long, Latitude, Longitude, Gender, Senior

Citizen, Partner, Dependents, Tenure Months, 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, Churn Label, CLTV, Churn Reason. The Customer Lifetime Value(CLTV) gives a measure of how much business a customer can give to the company in its lifetime.

CHAPTER 3

Literature Review

The ease of access to the services to businesses and consumers around the world has become possible due to the widespread network of various telecommunication service providers. The number of operators in the marketplace has significantly multiplied looking at the advancements in the telecommunication industry. This widespread network of telecommunication services motivated businesses to understand their customers better by collecting relevant data for their product and services in turn helping them understand the customer churn in the long run. The companies can understand customer sentiment for their services based on the analysis of customer clickstream data log files, web analytics tools such as IBM Cognos Analytics can provide a sequence of the user's online activities and reveal the user's surfing and purchasing behaviour. Website design, product placement optimisation, customer transaction analysis, market structure analysis, and product recommendations can be accomplished through web analytics [5]. The ability to effectively access the customer information can help infer customer's long term trust in the company's product and services and steer clear of customer churn.

The telecommunication sector has become a competitive market and customer retention is a major concern for the broadband service providers but they can forecast the customers who will move to another provider previously by analysing their behaviour. They can retain them by providing offers and their preferred services according to their historical records [6].

A novel algorithm using the profit variance as the basis for selecting features is proposed that assigns an appropriate coefficient to customer loss, whereby the data imbalance problem is solved. At the same time, the most profitable solution is included in the process of building the algorithm. Profits are taken into account in the construction of classifiers, which would improve the ability of the model to maximise profit. The substantial experiment is conducted and the results show that the models outperform conventional techniques in terms of profit, whereas the other performance metrics are not reduced [7]. Churn models aim to predict a customer's churning tendency by using behavioural and historical information. Yet, these models often focus on achieving maximum prediction accuracy rather than aiming their attention to the most important business requirement: profit maximisation [8]. It becomes equally important to not only focus on predicting the customers that are leaving the services but the ones that are staying to boost the profits. Verbraken et al. put forward a profit maximisation metric termed the expected maximum profit measure for customer churn (EMPC), which enabled the identification of the most profitable churn model [9].

There are multiple studies that are carried out in the aspect for churn prediction using traditional supervised learning and deep learning models.

There is a study that focuses on using hybridisation of two supervised learning algorithms namely logistic regression and decision trees. The dataset is split using the decision tree and the prediction is carried out using the logistic regression. The dataset splits were individually modelled for prediction using logistic regression which resulted in desired prediction [10].

Another study showed a refined one-class SVM model for churn prediction. This improved SVM detects anomalies that are close to the origin. On comparison of different kernel functions, Gaussian kernel gave the highest accuracy of 87.15% on the Teradata Duke Dataset [11].

A research study implemented a Neural Network application on the UCI dataset. They utilised the Feed Forward Backpropagation (FFBP) Neural Network, to get an accuracy of 92.35%. The best performance was achieved with a generated model composed of a NN having one neuron per numeric attribute for input layer, one hidden layer with three neurons and the output layer with two neurons for churners and non churners. This study was carried out without preprocessing or sampling of data to balance the churners and non churners classes [12].

A study of the customer churn by effectively identifying discrete groups of former customers and create a customer churn retention plan, sorting the various lost customer types using the RFM(Recency, Frequency, Monetary) principle. The principle can be useful to understand the customer behaviour and in turn help the organisation plan offers on their product and services. The study employed the SVC model to predict prospective churners by analysing the feature vectors of present customers to infer the likelihood of each customer churning in the future. This allows businesses to proactively identify and target those customers who are at higher risk of leaving the services. To ensure the accuracy of the model, it promptly requires the company to run the prediction on the same customers from time to time [13].

One paper implemented the Fuzzy Logic algorithm for churn prediction analysis on the data of higher education. The approach involved using the Fuzzy method C-means for clustering the customers imported from the data preprocessing. This allowed for obtaining normalisation results from each data. Subsequently, the membership degrees of each information were calculated using the Fuzzy method C-means [14].

In this research, the efficiency of four distinct machine learning algorithms was examined on the dataset. The algorithms included Logistic Regression, Random Forest, Decision Tree, and Gradient Boosting. To evaluate their performance, the F1 score and ROC-AUC were utilised. By comparing the outcomes with existing models,the findings revealed that Gradient Boosting, coupled with the feature selection technique, outperformed all other algorithms in this study, achieving an impressive 99% F1-score and 99% AUC [15].

A study showed that the application of CNN in conjunction with a variational autoencoder can assist in churn prediction with good accuracy. The test is conducted on three telecom datasets namely Cell2Cell, Telco and Orange. The model demonstrated superior performance on all three datasets, achieving a high level of accuracy. The integrating all string attributes in the dataset (ISAD) model played a significant role in improving the performance by providing additional feature options, enabling more accurate prediction of customer behaviour [3].

There is a proposed a model using Cell2Cell dataset and improved performance outcomes using a grid for hyper parameter optimisation. The model surpassed old models and accomplished an accuracy of 95% using Random Forest and 97% accuracy using gradient boosted trees [16].

The dataset consists of multiple features and thus it makes it important to reduce the dimensionality space such that the model complexity can be decreased. Researchers proposed a merging platform as a transitional step to combine feature selection algorithms both in classification and clustering [17]. For feature selection, another study computed AUC for every single feature. All the features with AUC larger than 0.5 were selected and unnecessary features were removed repeatedly [18].

There is another research that puts forward comparison of different feature selection algorithms in combination with the classification techniques. The study experimented with the Ant Search Method, Cuckoo Search Method, PSO Search Method, Linear Forward Selection and Principal Component Analysis. The results indicate that combining Ant search with the Decision Tree algorithm accomplished better accuracy then other feature selection algorithms [19].

Another feature selection study explores the use of an ensemble of Multilayer perceptrons (MLP) with negative correlation learning (NCL) to predict customer churn in a telecommunication company. The experimental results demonstrate that the NCL-based MLP ensemble outperforms both the flat ensemble of MLP without NCL and other conventional data mining techniques commonly used for churn analysis, exhibiting better generalisation performance with a higher churn rate [20].

The K-nearest neighbour algorithm based on the intuition that customers tend to be influenced by recent past events. The changes that occurred long in the past are unlikely to affect the future churning customers. Additionally, handling sequential data methods tends to be more tangled compared to dealing with uncomplicated non-temporal data, which further emphasises the advantage of representing recent information in a non-sequential format [21].

Despite being studied across various fields and yielding numerous algorithms, the exploration of feature selection encounters a triad of factors that have constrained its advancement. Primarily, each algorithm exhibits effectiveness limited to specific data types, rendering them less versatile. The performance in terms of efficiency and accuracy tends to deteriorate significantly when applied to inappropriate data types. Secondly, feature selection constitutes an NP-hard problem, signifying the absence of a universally optimal algorithm. For instance, while exhaustive search algorithms can yield optimal solutions, they often come at the expense of considerable time, whereas heuristic algorithms risk foregoing optimality while conserving resources [22].

Customer segmentation is a frequently employed marketing strategy wherein a company divides its clientele into more manageable subsets, each of which can be addressed with tailored content. This stratagem involves scrutinising customer behavioral data, which, in turn, furnishes the company with profound insights into the diverse customer profiles within its ecosystem [23].

CHAPTER 4

Methodology

The study will implement steps for efficient data analysis which includes data preprocessing, exploratory data analysis, feature engineering, model building and evaluation techniques. The traditional machine learning techniques for churn prediction like Logistic Regression, K Nearest Neighbors, Random Forest, Support Vector Machines and will utilise the hyperparameterisation techniques to compare the accuracy for prediction. The study also implements the churn prediction based on Feed Forward Neural Networks using TensorFlow.

4.1 Data Preprocessing

The data preprocessing is the first stage for data analysis which ensures the data quality and gives initial understanding of the data. The data is checked for consistency and missing values are replaced before employing the analysis. The dataset has 9 numerical features and 24 categorical features as shown in the Table 4.1. There are many unwanted features that do not contribute towards churn prediction such as **CustomerID**, **Count**, **Country**, **State**, **City**, **Zip Code**, **Lat Long**, **Latitude**, **Longitude**, **Churn Label** and are hence dropped from the analysis in the preprocessing stage of the study.

Index	**Data Type**
CustomerID	object
Count	int64
Country	object
State	object
City	object
Zip Code	int64
Lat Long	object
Latitude	float64
Longitude	float64
Gender	object
Senior Citizen	object
Partner	object
Dependents	object
Tenure Months	int64
Phone Service	object
Multiple Lines	object
Internet Service	object
Online Security	object
Online Backup	object
Device Protection	object
Tech Support	object
Streaming TV	object
Streaming Movies	object
Contract	object
Paperless Billing	object
Payment Method	object
Monthly Charges	float64
Total Charges	object
Churn Label	object
Churn Value	int64
Churn Score	int64
CLTV	int64
Churn Reason	object

Table 4.1: Data type of features

4.1.1 Missing Value Imputation

The data consists of less than 5% missing values. The missing values in the feature are replaced using KNNImputer.The function uses the parameter **n_neighbors** to select the number of similar rows of observations to estimate the value of the missing datapoint. The number of neighbors based on the square root of the number of observations to balance the trade-off between accuracy and computational efficiency.

$$n_{\text{neighbors}} = \text{int}\left(\text{round}\left(\text{data.shape[0]}^{0.5}\right)\right) \tag{4.1}$$

4.1.2 Statistical Analysis of Customer Churn

Exploratory data analysis (EDA) is a technique employed to examine and explore datasets, summarising their fundamental attributes through the use of visualisations. It aids in identifying the most effective strategies for processing data sources to acquire desired insights, facilitating the detection of patterns, identification of irregularities, validation of hypotheses, and verification of assumptions in a more accessible manner for data scientists. The study initially understands the data statistically and look for various factors that can affect the model results. The factors can be checking for outliers which can result in bias results. It can affect the model performance and can lead to unwanted relationships among features. The numerical features in the dataset as shown in the Figure 4.1 used in the study does not have any outliers.

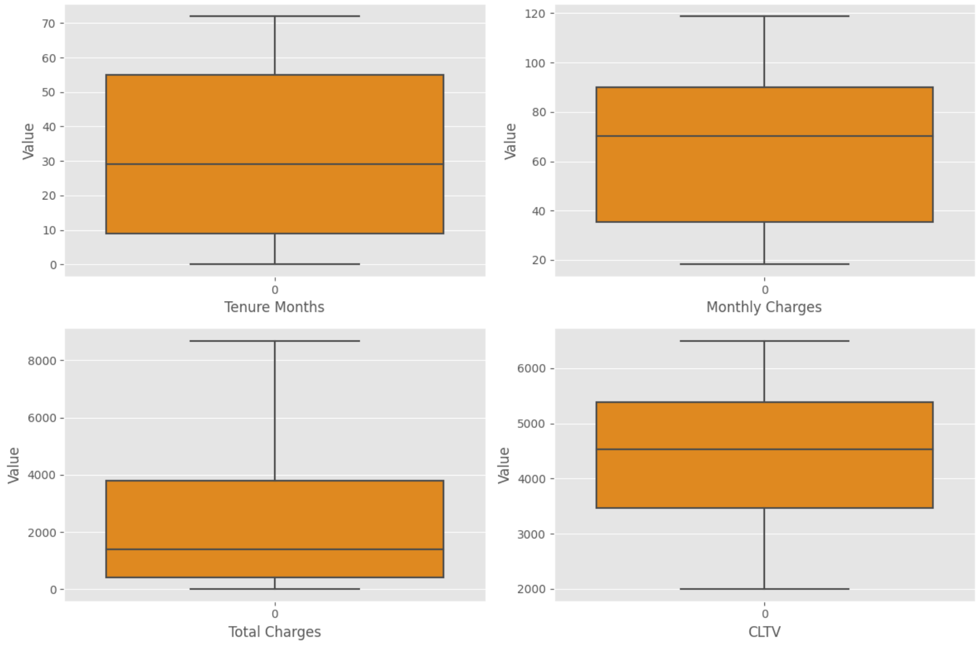

Figure 4.1: Checking outliers in data

The Figure 4.2 exhibit class distribution of prediction class used in the study. There are about three times more non-churners than churners in the dataset. The imbalance dataset can lead to biased results when trained against machine learning algorithms. The study deals with such scenario using SMOTE(Synthetic Minority Over-sampling Technique). In the study, it is applied to synthetically increase the instances of the churners in the dataset.

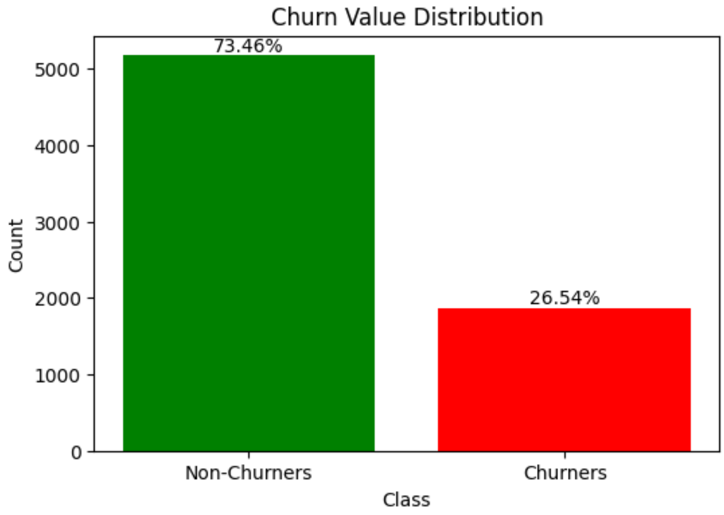

Figure 4.2: Churn class distribution

As a statistical measure of a customer churn data, understanding customer behaviour overtime gives a better understanding of churning at a broader level. The Figure 4.3 clearly depicts that the number of churners in the Month-to-month contract are higher in number in comparison to the customers on One-year and Two-year contracts. This statistically indicate that tenure is an important feature for consideration. Further looking at the customer demographics, it can be seen different features such as customer gender, whether a senior citizen or not, is in a relationship and whether dependent on someone also have some significant level of impact on the churn value. The countplots as depicted in the Figure 4.4 show the relation relation between customer type and the churn value. The plot shows that senior citizens are less likely to churn, meanwhile the customers with no dependents are more likely to churn.

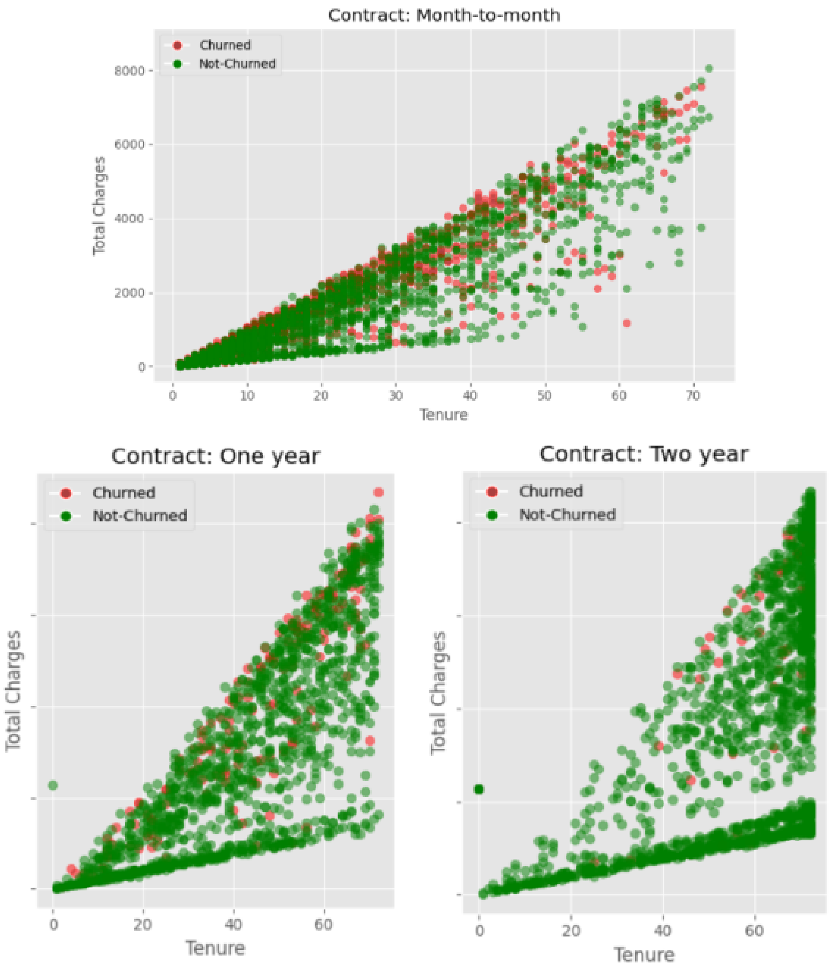

Figure 4.3: Customer churn by Tenure, Total Charges, and Contract Type

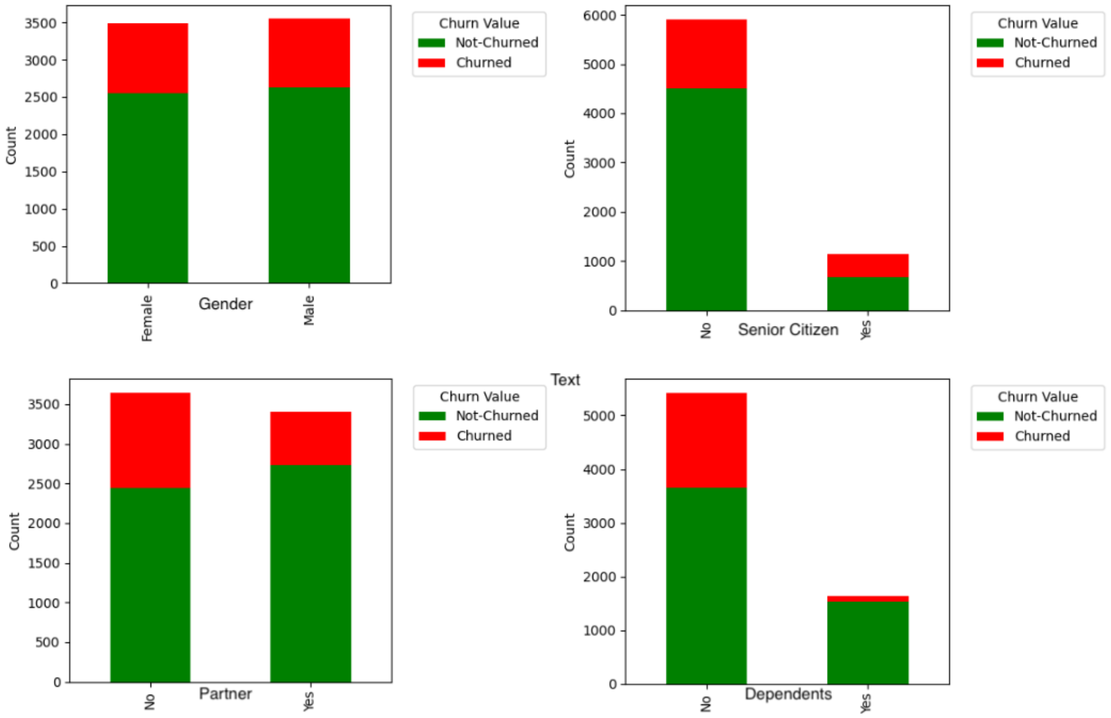

Figure 4.4: Churn based on customer demographics

4.2 Feature Engineering

In the study, there are various steps undertaken to improve the features for better analysis. It involved feature creation, feature transformation, feature encoding and feature selection.

There are a number of features in the dataset that act as an additional service for the customer. To better understand them, they are clubbed together in an umbrella and counted if the customer has churned and subscribed to the additional paying services. These services are **Multiple Lines**, **Online Security**, **Device Protection**, **Streaming Movies**, **Tech Support**, **Streaming TV** and **Online Backup**. The Figure 4.5 illustrates count of churned and non-churned customers based on the number of services opted by them.

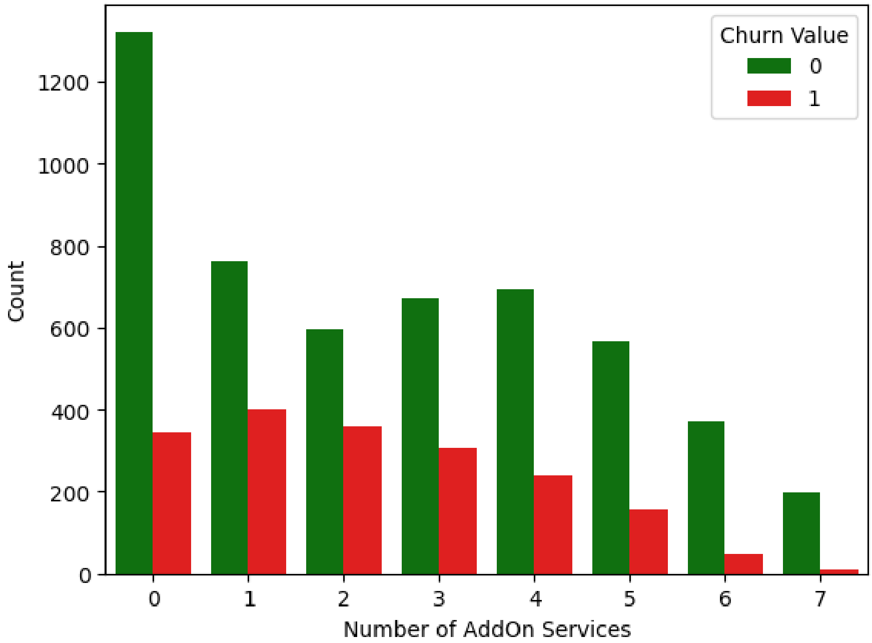

Figure 4.5: Churn based on additional services

The Table 4.2 provides a comprehensive overview of the statistical analysis conducted on various features within the dataset. The "Feature" column enumerates the attributes under evaluation. The "T-Statistic" signifies the difference between sample and population means, with positive values indicating higher sample means. The "P-Value" gauges evidence against the null hypothesis, with lower values indicating significant feature impact. The "Significant" column distinguishes statistically significant relationships; features with P-Values below a predetermined threshold (e.g., 0.05) are influential for outcome prediction.

Feature	T-Statistic	P-Value	Significant
Tenure Months	34.8238	1.1955×10^{-232}	true
Monthly Charges	-18.4075	8.5924×10^{-73}	true
Total Charges	18.8042	1.1005×10^{-75}	true
CLTV	10.6908	3.0489×10^{-26}	true

Table 4.2: Numerical Feature Significance Analysis

The Chi-square analysis post categorical feature encoding provides key insights into feature-outcome relationships. Features with p-values < 0.05 are significant, e.g., "Senior Citizen", "Partner", "Dependents", "Paperless Billing", "Churn Value", "Internet Services", "Security", "Backup", "Support", "Streaming", "Contract", "Payment Methods", and "Add-on count". These aid predicting churn. Conversely, p >= 0.05 makes features, like "Gender", "Phone Service", "Multiple Lines_No phone service" and "AddOns Count_4" less impactful for churn prediction and are removed. The Figure 4.6 refers to the highly correlated features with the "Churn Value".

4.3 Model Evaluation Metrics

The choice of an evaluation metric significantly impacts the attainment of an optimal classifier during classification training. Therefore, selecting an appropriate evaluation metric is a pivotal factor in distinguishing and achieving the best classifier. This study comprehensively examines relevant evaluation metrics tailored to act as discriminators for enhancing generative classifiers [24].

The models in the study are evaluated based on Accuracy Score, AUC Score, Confusion Matrix, F1-Score and Precision-Recall. For comparing the machine learning algorithms among each other, ROC AUC Mean and ROC AUC STD, Train Accuracy Mean and Train Accuracy STD, and Test Accuracy Mean and Test Accuracy STD is calculated for both non parameterised and parameterised machine learning algorithms applied in the study. In the context of evaluating machine learning models, several key metrics play a pivotal role in assessing their performance.

1. **Confusion Matrix**: A confusion matrix provides insights into the model's performance by showing how many instances were correctly or incorrectly classified

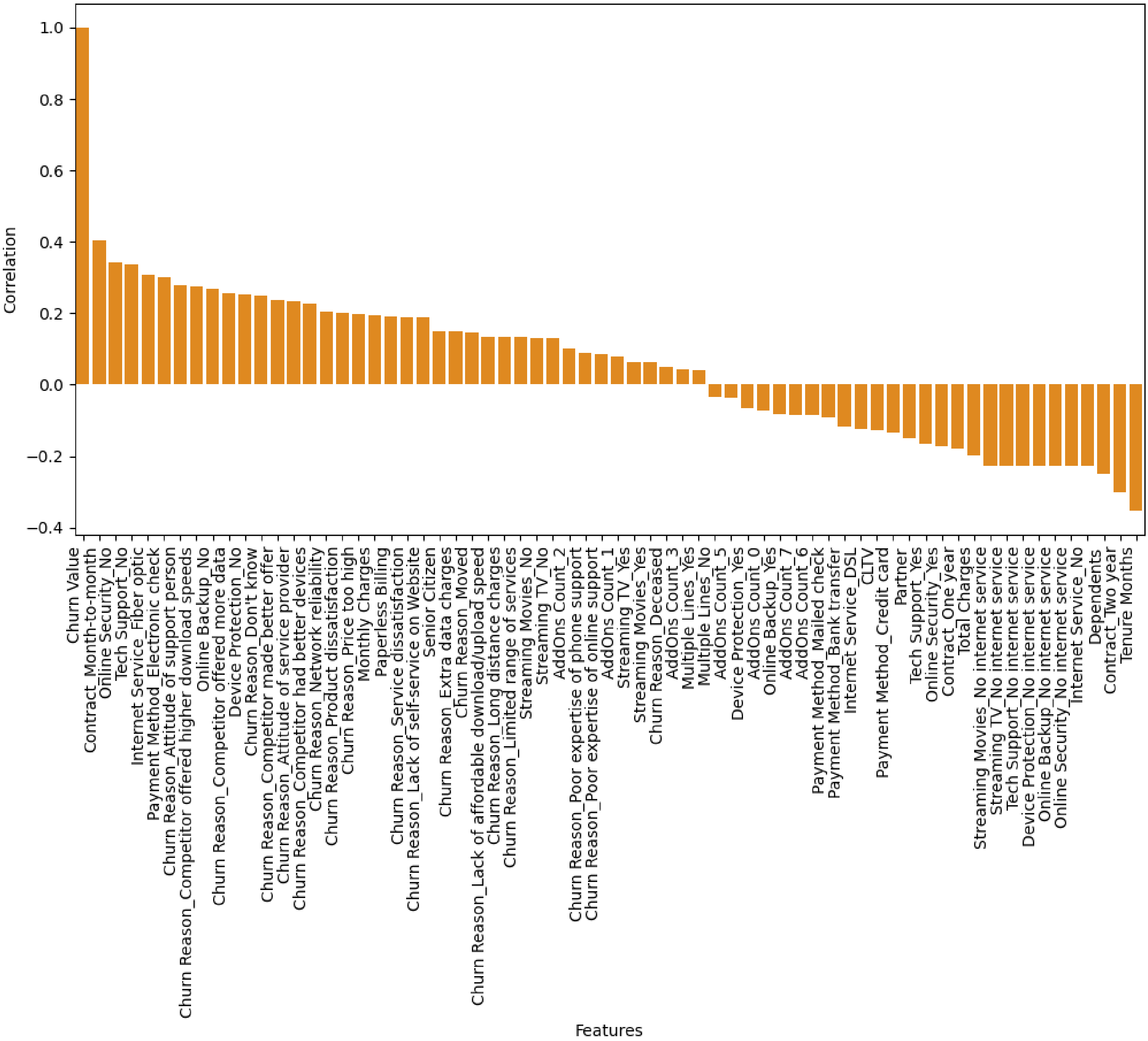

Figure 4.6: Correlation of features with churn value

for each class. From these values, various metrics can be calculated, such as accuracy, precision, recall (sensitivity), specificity, and F1-score, which give a more comprehensive picture of the model's effectiveness in different scenarios [25].

In a confusion matrix:

- True Positive (TP): The model correctly predicted a positive outcome when the actual outcome was positive.

- True Negative (TN): The model correctly predicted a negative outcome when the actual outcome was negative.
- False Positive (FP): The model predicted a positive outcome when the actual outcome was negative (also known as a Type I error).
- False Negative (FN): The model predicted a negative outcome when the actual outcome was positive (also known as a Type II error).

2. **ROC AUC Mean and ROC AUC STD**: The Receiver Operating Characteristic Area Under the Curve (ROC AUC) is a measure that quantifies a model's ability to distinguish between classes. The ROC AUC Mean reflects the average performance across different runs or folds of cross-validation, while the ROC AUC STD indicates the variability or spread of these values. Higher ROC AUC values indicate better model discrimination [26].

3. **Train Accuracy Mean and Train Accuracy STD**: These metrics measure the accuracy of a model's predictions on the training dataset. The Train Accuracy Mean is the average accuracy across various iterations or cross-validation folds, and the Train Accuracy STD represents the variation in accuracy. Higher Train Accuracy indicates that the model performs well on the training data.

4. **Test Accuracy Mean and Test Accuracy STD**: Similar to train accuracy, Test Accuracy Mean gauges the average accuracy of a model's predictions on unseen test data, while Test Accuracy STD quantifies the variability in these accuracies across different test sets. A higher Test Accuracy Mean suggests that the model generalises well to new data.

These metrics collectively provide insights into the performance, consistency, and generalisation capabilities of machine learning models. Researchers and practitioners use them to compare and select models that best suit the desired classification task.

4.4 Model Building

In this phase of the study, the preprocessed data is split in train and test data. subsequent machine learning analysis. The primary objective was to appropriately configure the dataset for training and testing while addressing the challenge of class imbalance. The process involved several key steps orchestrated to ensure the integrity of the analysis. The dataset was divided into an 80% training subset and a 20% testing subset. To ensure the reproducibility of results.

In the context of dealing with imbalanced datasets, the application of the Borderline-SMOTE technique involves creating synthetic data instances to address the discrimination between the classes. This technique is designed to tackle situations where the minority class instances lie near the decision boundary that separates it from the majority class. By identifying these borderline instances, Borderline-SMOTE generates synthetic samples that are more relevant to the complex decision boundary, thus enhancing the representation of the minority class.

The study compared four traditional machine learning algorithms namely, Logistic Regression, k-Nearest Neighbor, Random Forest and Support Vector Machines.

4.4.1 Logistic Regression

Logistic Regression[15] is a binary classification algorithm that uses probability of occurrence of the categorical value which is related to more than one feature. It statistically determines the relationship of features with each other. Unlike a linear regression model, logistic regression model is used where response variable takes only two values like 1/0, Yes/No or True/False. The probability of occurrence of response variable is given by the logistic function logit(p).

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \epsilon \tag{4.2}$$

Where:

- p represents probability of the positive class.
- $x_1, x_2, ..., x_n$ represents the predictor features.
- $\beta_1, \beta_2, ..., \beta_n$ represents the coefficients of predictor features.
- ϵ represents the error term in prediction of the outcome class.

In the study, an instance of the logistic regression model is created with the Logistic Regression class, where **max_iter** specifies the maximum number of iterations for convergence. The model is then trained using the oversampled training data using Borderline-SMOTE, which has been obtained to address class imbalance.

The **feature_importance** function is designed to plot the most important and least important features based on their weights in a logistic regression model. The weights associated with the features in a logistic regression model play a significant role in determining the impact of each feature on the prediction outcome. The Figure 4.7 shows the most important and the least important features contributing to the model's prediction accuracy.

The function calculates the coefficients (weights) associated with each feature in the logistic regression model using $classifier.coef\_[0]$. These coefficients represent the change in the log-odds of the response variable for a one-unit change in the feature while keeping other features constant. The calculated weights are stored in a Pandas Series called weights.

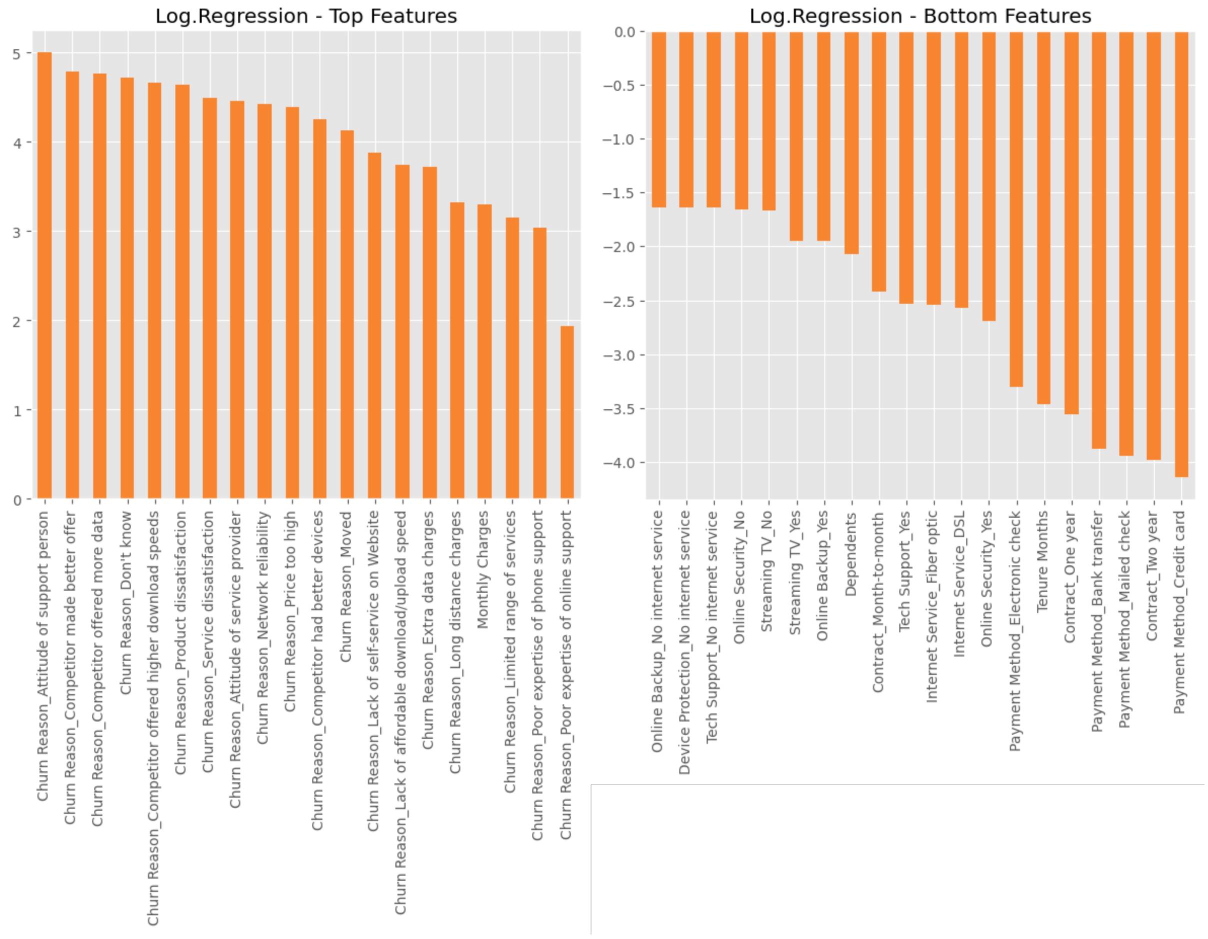

Figure 4.7: Feature importance of logistic regression model

Confusion matrix provides insights into the true positive, true negative, false positive, and false negative predictions of the model. It aids in assessing the model's accuracy and its ability to correctly classify instances. The Figure 4.8 shows a clear visual representation of the model predictions.

- True Positive (TP): The model correctly predicted 392 instances as "Churn."
- True Negative (TN): The model accurately predicted 905 instances as "Non-churn."
- False Positive (FP): The model incorrectly classified 104 instances as "Churn" when they were actually "Non-churn."
- False Negative (FN): The model misclassified 8 instances as "Non-churn" when they were actually "Churn."

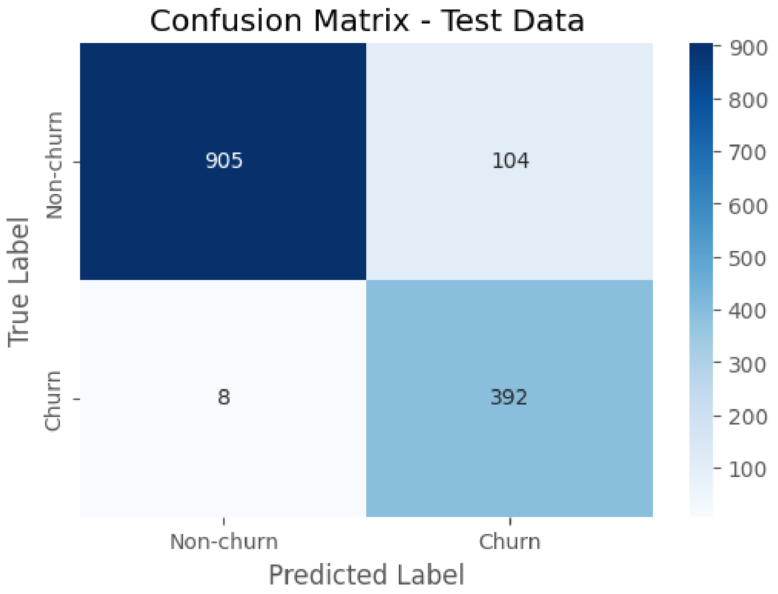

Figure 4.8: Confusion matrix of logistic regression model

It accurately predicted approximately 91.78% of the instances in the training set and achieved an accuracy of 92% on the separate test set. This suggests that the model is consistent in its predictions and generalises well to new, unseen data. The relatively high accuracy on both the training and test sets indicates that the model's predictions are reliable and effective in distinguishing between different classes. The provided evaluation metrics offer insights into the performance of the logistic regression model:

- AUC Score (ROC): 0.9945

  The Area Under the Receiver Operating Characteristic (ROC) Curve (AUC-ROC) score as shown in Figure 4.9 is a measure of the model's ability to distinguish between the positive and negative classes. A value close to 1 indicates that the model has a high true positive rate and a low false positive rate, suggesting that it is effective in classifying instances.

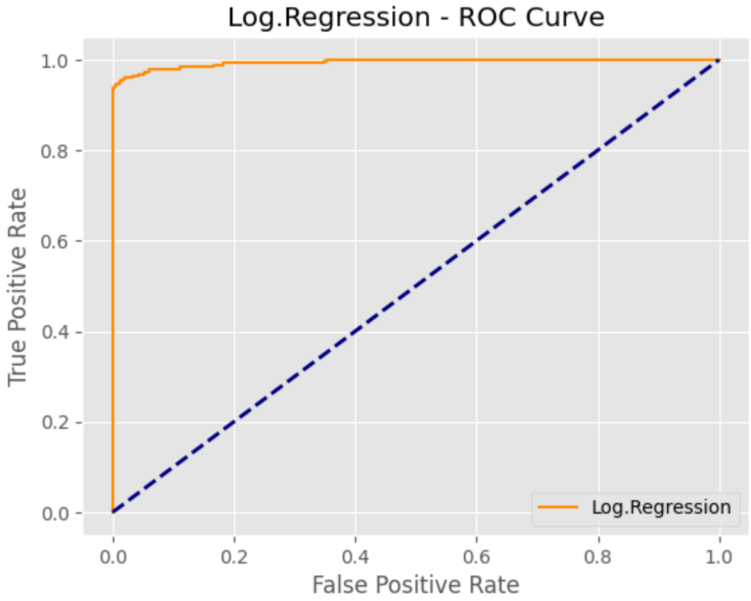

Figure 4.9: ROC curve of logistic regression model

- F1 Score: 0.875

  The F1 score is a metric that considers both precision and recall. It provides a balanced measure of the model's accuracy, particularly in situations with imbalanced class distribution. An F1 score of 0.875 indicates that the model achieves a good balance between precision and recall.

- AUC Score (PR): 0.9903

  The Area Under the Precision-Recall (PR) Curve (AUC-PR) score assesses the model's performance in terms of precision and recall across different probability thresholds. A value close to 1 as shown in Figure 4.10 indicates that the model has high precision and recall, demonstrating its ability to correctly classify positive instances while minimising false positives.

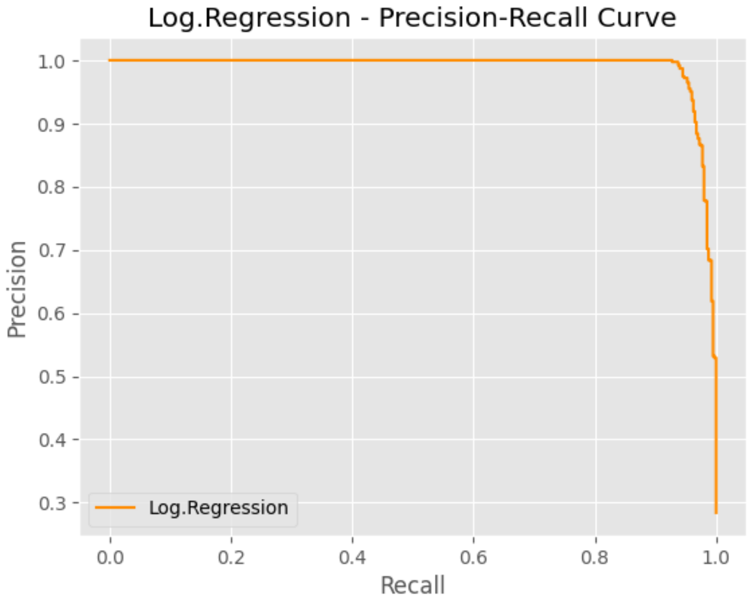

Figure 4.10: Precision-Recall curve of logistic regression model

4.4.2 K-Nearest Neighbor Classifier

Neighbors-based classification is a form of instance-based or non-generalising learning, wherein the objective is not to create an overarching internal model, but rather to preserve instances from the training dataset. Classification is determined through a straightforward majority vote involving the closest neighbors of each data point. Essentially, a query point is attributed the class that appears most frequently within its nearest neighbor set.Neighbors-based classification is a form of instance-based or non-generalising learning, wherein the objective is not to create an overarching internal model, but rather to preserve instances from the training dataset. Classification is determined through a straightforward majority vote involving the closest neighbors of each data point. Essentially, a query point is attributed the class that appears most frequently within its nearest neighbor set [27]. The KNN algorithm can be mathematically represented as follows:

$$\hat{y}(x) = \text{mode}\{y_i\}_{i \in \mathcal{N}_k(x)} \tag{4.3}$$

Where:

- $\hat{y}(x)$ is the predicted class label for the query instance of x.
- $\mathcal{N}_k(x)$ represents the set of k nearest neighbors of the query instance x.
- y_i is the class lavel of the i^{th} nearest neighbor.

The confusion matrix illustrated in the Figure 4.11 shows:

- True Positives (TP): 255 instances were accurately predicted as "Churn."
- True Negatives (TN): 814 instances were correctly identified as "Non-churn."
- False Positives (FP): 195 instances were wrongly classified as "Churn" when they were actually "Non-churn."
- False Negatives (FN): 145 instances were mistakenly categorized as "Non-churn" when they were truly "Churn."

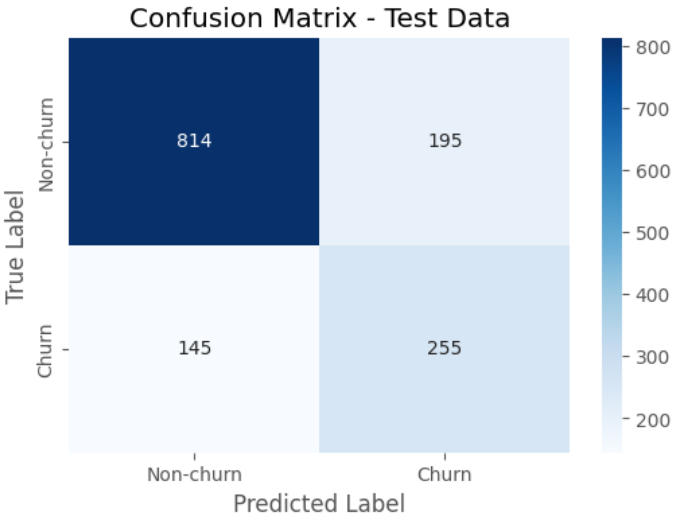

Figure 4.11: Confusion matrix of KNN model

The provided evaluation metrics offer insights into the performance of the K-Nearest Neighbors (KNN) classifier:

- **Train Set Accuracy:** 0.9002
  This metric indicates that the K-Nearest Neighbors (KNN) model achieved an accuracy of approximately 90.02% on the training dataset. It reflects the percentage of correctly predicted instances in the training set.
- **Test Set Accuracy:** 0.76
  The test set accuracy of 0.76 implies that the KNN model accurately predicted around 76% of the instances in the test dataset. It provides an indication of how well the model generalises to unseen data.

- **AUC Score (ROC):** 0.8003

  The Area Under the Receiver Operating Characteristic (ROC) Curve (AUC-ROC) score measures the KNN model's ability to discriminate between positive and negative classes. An AUC-ROC score of 0.8003 indicates a relatively good separation of classes.

- **F1 Score:** 0.6

  The F1 score balances precision and recall, providing an overall measure of model accuracy. A value of 0.6 suggests a moderate balance between correct positive predictions and minimising false negatives.

- **AUC Score (PR):** 0.6101

  The Area Under the Precision-Recall (PR) Curve (AUC-PR) score evaluates the model's precision-recall trade-off. A score of 0.6101 indicates that the model has moderate precision and recall performance. The Figure 4.12 depicts the false positive rate and the recall of the algorithm.

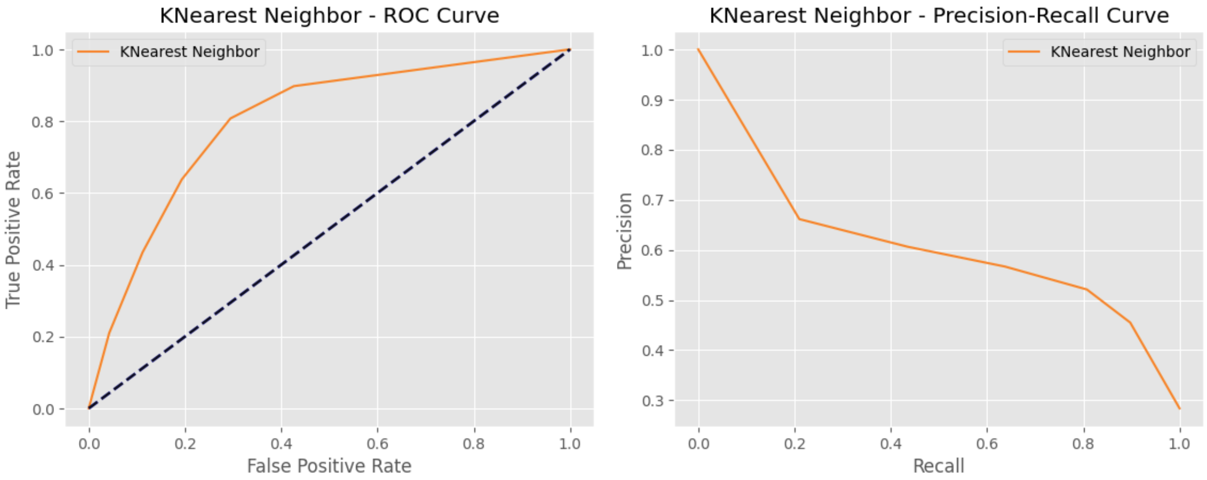

Figure 4.12: ROC and Precision-Recall curve of KNN model

The KNN classifier demonstrates reasonable performance with respect to accuracy, AUC scores, and the F1 score. However, further optimisation may be needed to enhance its predictive capabilities, particularly in terms of precision and recall.

4.4.3 Random Forest Classifier

A random forest is a composite model that constructs multiple decision tree classifiers on different subsets of the dataset. It leverages averaging to enhance predictive accuracy and mitigate overfitting [28]. The random forest is mathematically represented as:

$$F(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x) \tag{4.4}$$

Where:

- N is the number of decision trees in the forest.
- $f_i(x)$ is the prediction of the i^{th} tree.

The traditional random forest classifier has ability to deal with imbalanced dataset. Therefore the training data in the study is supplied without the SMOTE sampling method for prediction. As portrayed in the Figure 4.13, it clearly shows how it has different features with more importance as compared to logistic regression model. The Figure 4.14 shows the confusion matrix obtained after model creation.

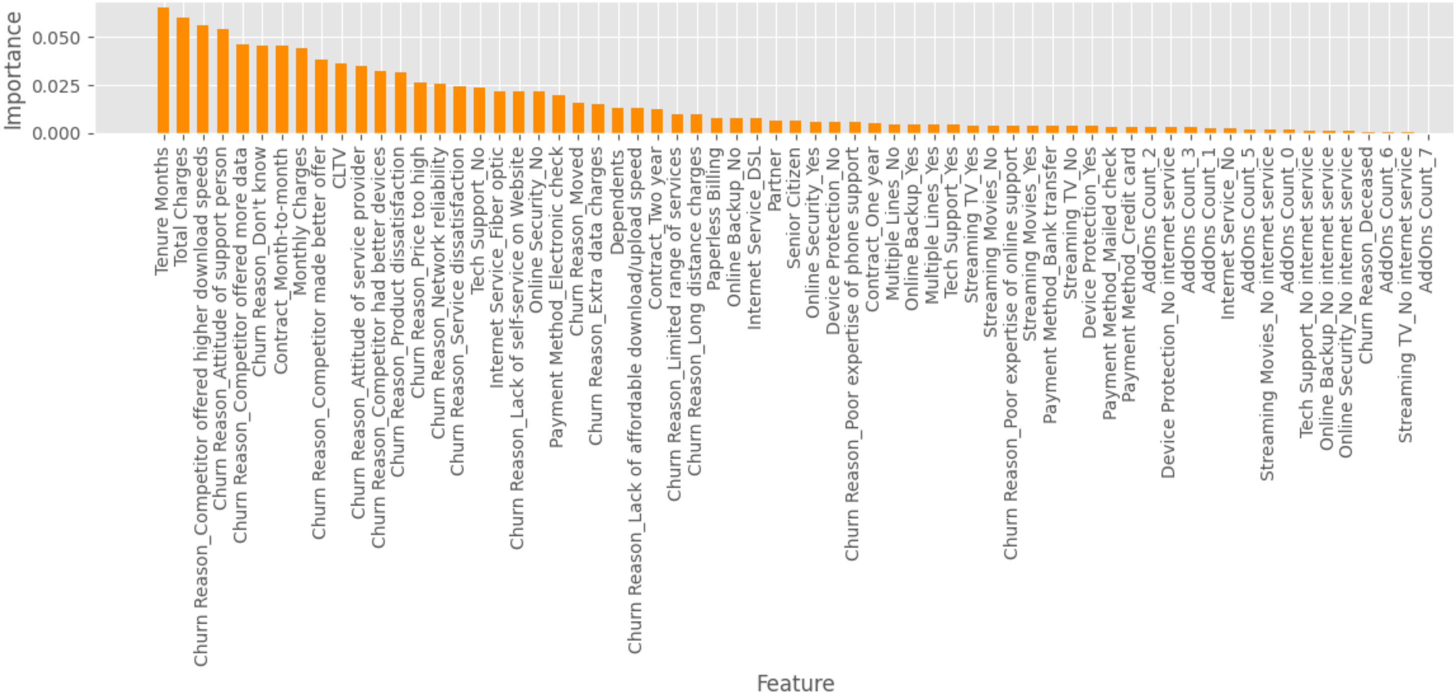

Figure 4.13: Feature importance in random forest model

- True Positives (TP): 366 instances were correctly predicted as "Churn."
- True Negatives (TN): 1001 instances were accurately predicted as "Non-churn."
- False Positives (FP): 8 instances were incorrectly classified as "Churn" when they were actually "Non-churn."

- False Negatives (FN): 34 instances were misclassified as "Non-churn" when they were actually "Churn."

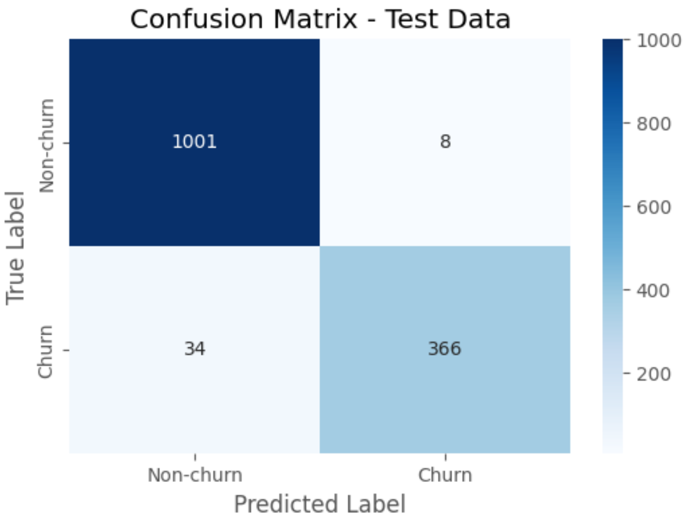

Figure 4.14: Confusion matrix of random forest classifier model

The provided evaluation metrics offer insights into the performance of the Random Forest classifier:

- **Train Set Accuracy**: 1.0
  The model achieves a perfect accuracy of 100% on the training set. This suggests that the model has learned the training data extremely well and can predict the training instances with high accuracy. However, this perfect accuracy could also indicate overfitting, where the model might not generalise well to new, unseen data.

- **Test Set Accuracy**: 0.97
  The model demonstrates a high accuracy of 97% on the test set. This indicates that the model is performing well on unseen data, and it can generalise its predictions effectively to new instances.

- **AUC Score (ROC)**: 0.9948
  The Area Under the Receiver Operating Characteristic (ROC) Curve (AUC-ROC) score is very close to 1 as illustrated in the Figure 4.15, indicating that the model has a high true positive rate and a low false positive rate. This suggests that the model's ability to distinguish between positive and negative classes is excellent.

- **F1 Score**: 0.9457

  The F1 score is a balanced measure of precision and recall. An F1 score of 0.9457 indicates that the model is achieving a good balance between correctly identifying positive instances and minimising false positives.

- **AUC Score (PR)**: 0.9844

  The Area Under the Precision-Recall (PR) Curve (AUC-PR) score shown in Figure 4.15 is close to 1, suggesting that the model has high precision and recall across different probability thresholds. This indicates that the model is performing well in classifying positive instances while controlling false positives.

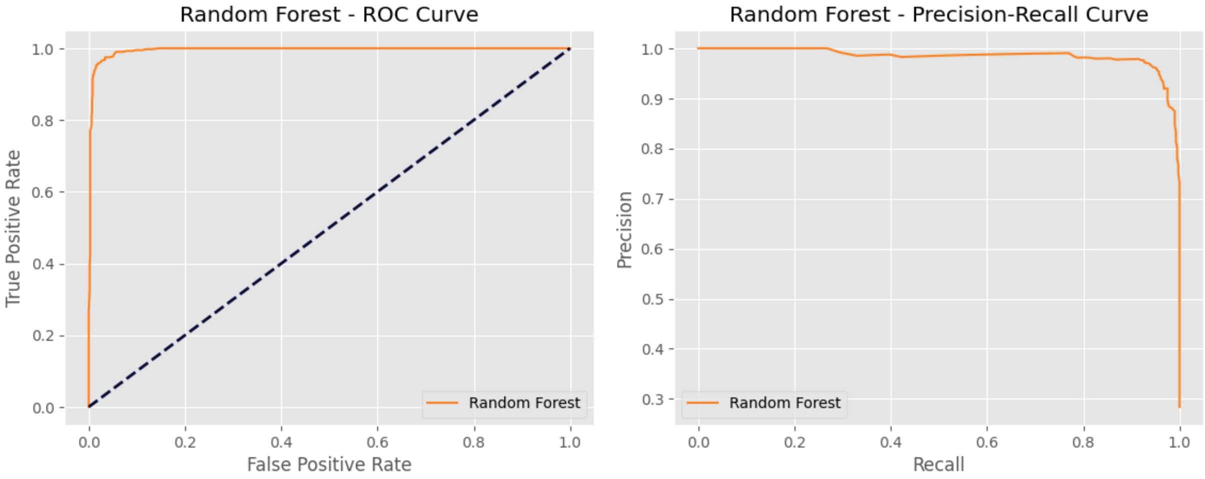

Figure 4.15: ROC and Precision-Recall curve of random forest classifier model

4.4.4 Support Vector Classifier

The SVM algorithm, pioneered by Vapnik, is grounded in the principles of statistical learning theory. In certain classification scenarios, the goal is to identify an optimal hyperplane that effectively separates two distinct classes. When the points belonging to these classes in the training dataset can be distinguishedsing a linear hyperplane, it becomes intuitive to employ the hyperplane that creates the largest margin between the two sets of points [11]. The optimal hyperplane is found by solving the following optimisation problem:

$$\text{minimise:} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to} \quad y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

Where:

- $f(x)$ is the decision function that assigns an instance x to a class label.
- w is the weight vector orthogonal to the hyperplane.
- x is the feature vector of the instance.
- b is the bias term that shifts the hyperplane from the origin.
- C is the regularisation parameter that controls the trade-off between maximizing the margin and minimising the classification error.
- x_i are slack variables that allow for some instances to be within the margin or misclassified.

  In the study for churn prediction, SVC in combination with the RBF kernel. When utilising the Radial Basis Function (RBF) kernel for training a Support Vector Machine (SVM), it's essential to account for two crucial parameters: C and gamma. The parameter C, shared across various SVM kernels, represents a trade-off between the misclassification of training instances and the simplicity of the decision boundary. A lower C value results in a smoother decision boundary, whereas a higher C value aims to classify all training examples accurately. On the other hand, gamma determines the extent of influence that a single training example holds. A higher gamma value requires other examples to be closer in proximity to be impacted by it [29].

After the model is fit, the confusion matrix is generated as shown in Figure 4.16.

- True Positives (TP): 393 instances were correctly predicted as "Churn."
- True Negatives (TN): 885 instances were accurately predicted as "Non-churn."
- False Positives (FP): 124 instances were incorrectly classified as "Churn" when they were actually "Non-churn."
- False Negatives (FN): 7 instances were misclassified as "Non-churn" when they were actually "Churn."

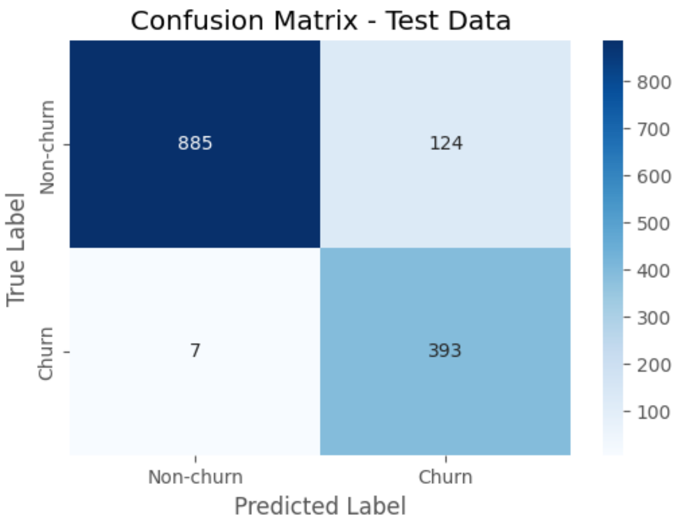

Figure 4.16: Confusion matrix of SVC model

The provided evaluation metrics offer insights into the performance of the Support Vector classifier:

- **Train Set Accuracy:** 0.930
  The model achieved an accuracy of approximately 93% on the training set. This indicates that the model is performing well on the data it was trained on.

- **Test Set Accuracy:** 0.910
  The model exhibited an accuracy of around 91% on the test set. This suggests that the model is also performing well on new, unseen data, indicating its generalisation capability.

- **AUC Score (ROC):** 0.977
  The Area Under the Receiver Operating Characteristic (ROC) Curve (AUC-ROC) score is close to 1, indicating that the model's ability to discriminate between positive and negative classes is excellent. The model is effective in distinguishing between the two classes.

- **F1 Score:** 0.857
  The F1 score is a balanced measure of precision and recall. An F1 score of approximately 0.857 indicates that the model is achieving a good balance between correctly identifying positive instances and minimising false positives.

- **AUC Score (PR):** 0.943

  The Area Under the Precision-Recall (PR) Curve (AUC-PR) score shown in Figure 4.17 is close to 1, suggesting that the model has high precision and recall across different probability thresholds. This indicates that the model is performing well in classifying positive instances while controlling false positives.

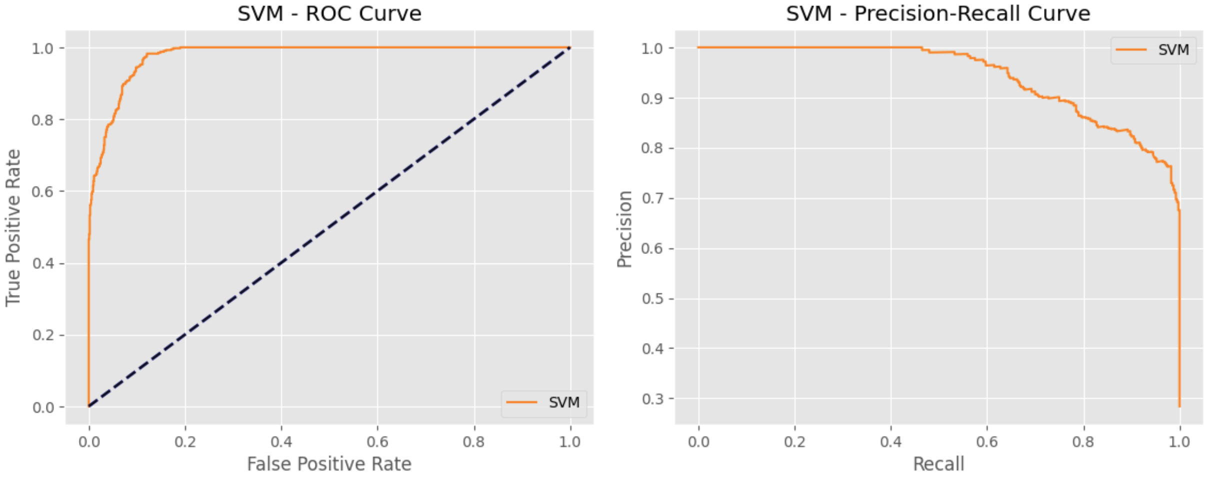

Figure 4.17: ROC and Precision-Recall curve of SVC model

CHAPTER 5

A Neural Network Approach

An Artificial Neural Network (ANN) is a sophisticated system consisting of a multitude of basic units called neural cells. The concept of ANN drew inspiration from intricate studies in biological research related to human brain tissue and neural systems. This network architecture is designed to replicate the neural processes of information handling observed within the human brain [30]. A standard multi-layer perceptron neural network [12] of the feed-forward type includes an input layer, output layer, and hidden layer as illustrated in the Figure 5.1.

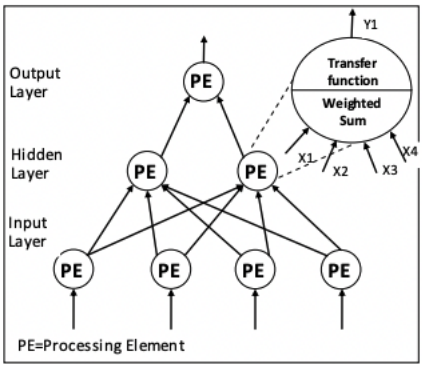

Figure 5.1: Neural network with one hidden layer

In the study for churn prediction, the construction and training of a neural network model for binary classification using TensorFlow and Keras is taken into consideration. It incorporates techniques such as ReLU activation, dropout, L2 regularisation, learning rate scheduling, and early stopping to improve model performance and prevent overfitting. The model implements a sequential keras approach. The parameters taken into factor for the neural network are discussed as follows:

1. ReLU activation is used to commence non-linearity to the model, helping it to learn complex relationships in the data.

2. Dropout is a regularisation technique where randomly selected neurons are ignored during training.

3. L2 regularisation adds a penalty to the loss function based on the magnitude of the weights and enables to prevent overfitting.

4. Learning rate scheduling adjusts the learning rate during training. A higher learning rate at the start helps the model converge faster.

The optimisation technique employed is the Adam method, which stands for Adaptive Moment Estimation. This approach incorporates historical gradients along with their coordinate-wise squared gradients. One of the notable benefits of this algorithm lies in its implicit adjustment of the learning rate. Specifically, the algorithm adapts the learning rate proportionally to the steepness of the error landscape. Consequently, it facilitates faster progress in regions of higher gradient, thereby leading to more direct convergence towards the optimal solution within the error space.

CHAPTER 6

Profit Maximisation in Churn

Client segmentation serves as a potent marketing strategy extensively embraced by businesses with the overarching goal of optimising profits. This entails partitioning the customer base into more compact groups, driven by distinguishing features encompassing demographics, behavioral traits, and historical purchasing patterns [23].

In the pursuit of effective customer segmentation, the study employs random forest model based on the new features selected as shown in Figure 4.13 to categorise customers based on their probability of churning. Through this approach, the aim is to address the challenge of retaining customers by identifying those who are more likely to churn.

The heart of the process lies in categorising customers into distinct churn rate categories based on their predicted probabilities. This is achieved by iteratively assessing each churn prediction value against predetermined thresholds. Depending on the prediction's magnitude, customers are assigned to categories such as "high" and "low" churn rate segments. This categorisation is executed systematically, leading to a list of churn rate categories associated with each customer as shown in the Figure 6.1.

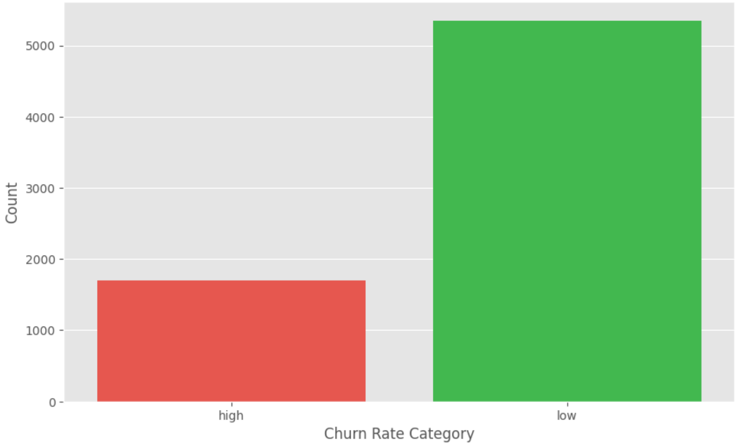

Figure 6.1: Churn Categories

As shown in the Figure 6.2, the study checks the density distribution of Customer Lifetime Value(CLTV). A high Customer Lifetime Value (CLTV) signifies that a customer generates substantial sales and this tendency is likely to continue in the future. On average, approximately 20% of a company's customer base tends to be unprofitable, while 60% can be classified as profitable customers, leaving only a 20% subset that significantly contributes to a company's profitability.

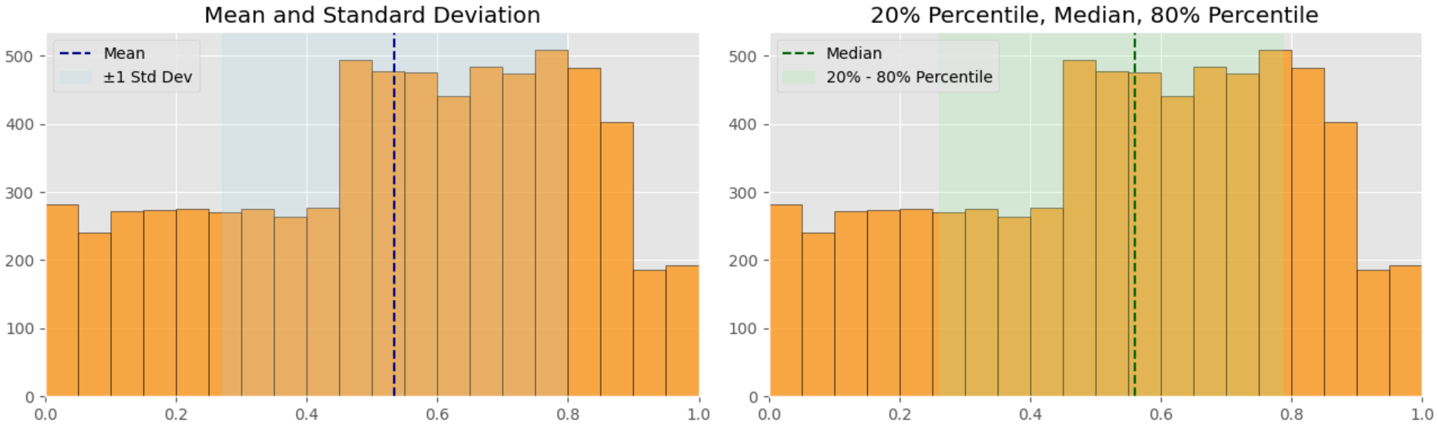

Figure 6.2: CLTV Distribution

CHAPTER 7

Results

The results of the study gives a clear understanding of the problem set in the section 2.1. The algorithms applied for the analysis dealt with unbalanced dataset during cross-validation and performance is compared among models. The research questions that were discussed earlier in the study are appropriately presented with implemented solutions as follows:

7.1 Traditional Classification Algorithms Comparison

After studying the traditional algorithms, it is evident that Logistic Regression, SVC and Random Forest provide substantial results. Though Random Forest Classifier performed 100% on the training set, it is still validated using cross-validation techniques along with the other traditional algorithms. One solution to tackle this challenge involves a technique known as cross-validation (CV). While a test set is still reserved for final evaluation, cross-validation eliminates the need for a separate validation set. The fundamental approach, often referred to as k-fold CV, involves partitioning the training set into k smaller subsets [31]. For each of the k "folds", the following procedure is executed:

1. Train a model using k-1 of the folds as training data.

2. Validate the resultant model on the remaining portion of the data. This subset is utilised as a test set to compute performance metrics such as accuracy.

The performance measure presented by k-fold cross-validation is the average of the values calculated during each iteration. While this approach might demand computational resources, it offers the advantage of efficient data utilisation. This characteristic is particularly valuable in scenarios like inverse inference, where the dataset size is notably limited.The comparison can be seen in the Table 7.1.

Index	Algorithm	ROC AUC Mean	ROC AUC STD	Train Accuracy Mean	Train Accuracy STD	Test Accuracy Mean	Test Accuracy STD
0	Logistic Regression	100.0	0.0	99.88	0.18	98.37	1.31
3	SVC	100.0	0.0	99.95	0.08	96.81	2.18
2	Random Forest	99.49	0.2	96.66	0.83	88.43	3.04
1	K Nearest Neigbhors	78.57	3.04	77.83	1.55	76.16	3.6

Table 7.1: Summary of evaluation metrics for different algorithms

The ROC AUC Mean and ROC AUC STD values for logistic regression and SVC indicate data leakage or overfitting. It might suggest that the model has unknowingly learned something from the test data or features that it shouldn't have access to, leading to an artificially high performance on unseen data. Thus hyperparameter tuning is applied to logistic regression, SVC and Random Forest for further analysis in the study.

7.2 Tuning Traditional Machine Learning Models

7.2.1 Tuned Logistic Regression

These results suggest that the logistic regression model, with the specified hyperparameters, achieved an accuracy of around 93.47% on the unseen test data. The choice of L1 regularisation ("l1") can result in a sparse model, meaning some features may have coefficients set to zero, effectively performing feature selection. The "saga" solver is known for its effectiveness with L1 regularisation. The "saga" solver is preferred for large datasets [32].

7.2.2 Tuned Support Vector Classifier

These results indicate that the SVM model, using the specified hyperparameters, achieved an accuracy of around 93.04% on the unseen test data. The hyperparameters suggest a relatively high value of C (1000.0), which means the model is more tolerant of misclassified points in the training data. The choice of the radial basis function kernel ("rbf") is common for SVMs due to its flexibility in mapping data to higher dimensions. The "gamma" parameter being set to "auto" suggests that the algorithm determines the value automatically, likely based on the input data.

7.2.3 Tuned Random Forest Classifier

Prior to this hyperparameter tuning, an initial feature importance analysis as shown in the Figure 4.13 was conducted in the study. Features with importance scores greater than 0.02 were selected for training the model again. This strategic feature selection aimed to retain the most relevant variables and discard those with limited predictive value.

The hyperparameters optimisation process using RandomizedSearchCV has led to the identification of the optimal configuration for the given model. The best hyperparameters obtained are as follows: n_estimators set to 200, max_features utilising the "log2" value, max_depth assigned as 20, criterion employing the "gini" criterion for impurity reduction, and bootstrap enabled. The selected parameters collectively define the characteristics of the random forest model, indicating a higher number of estimators, specific features to consider for each tree, a limited maximum depth for trees to prevent overfitting, the Gini impurity criterion for node splitting, and the use of bootstrapped samples for training.

After the tuning, **Random Forest Classifier** outperformed Logistic Regression and SVC with a test accuracy of 96%. It can be seen in the Table 7.2. The evaluation metrics for the best random forest classifier are illustrated in the Figure 7.1.

Model	Train Set Accuracy	Test Set Accuracy	AUC Score (ROC)	F1 Score
Tuned Logistic Regression	0.9236	0.9300	0.99997	0.8969
Tuned Random Forest Classifier	0.9695	0.9600	0.9713	0.9170
Tuned SVC	0.9510	0.9300	1.0000	0.8889

Table 7.2: Performance metrics of tuned models

7.3 Analysis of Neural Network Approach

In the implementation, the Rectified Linear Unit (ReLU) activation function is employed for all dense layers. This selection is a prevailing practice due to its computational efficiency and its ability to avert vanishing gradient issues for positive inputs. This activation function is widely embraced in deep learning models.

The utilisation of dropout is evident with rates of 0.2 and 0.1 applied after the initial and subsequent dense layers, respectively. These rates signify the proportion of neurons

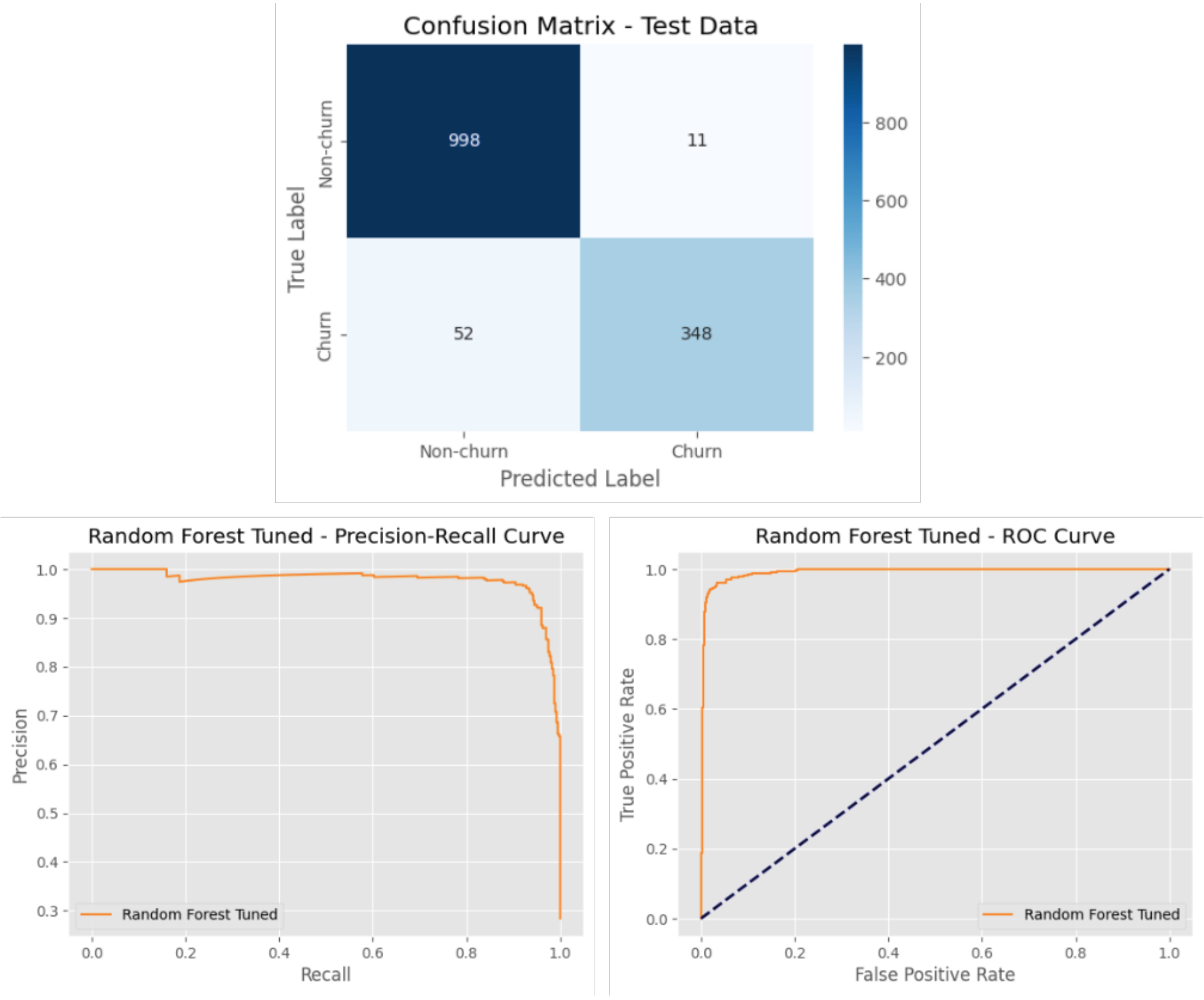

Figure 7.1: Evaluation metrics of tuned random forest model

randomly disregarded during training. This strategic choice of rates is grounded in their established effectiveness in mitigating overfitting. By employing relatively lower dropout rates, a more robust signal is retained while effectively counteracting overfitting.

The inclusion of L2 regularisation involves a strength of 0.0001 for both dense layers. This calibrated value is relatively modest, striking a balance between regularisation and the weight penalty. The value's determination likely amalgamates empirical exploration and domain knowledge, aligning with effective regularisation practices.

Furthermore, a learning rate schedule is instantiated utilising the InverseTimeDecay approach. Commencing with an initial learning rate of 0.0001, this schedule orchestrates a gradual decrease in the learning rate over time. The values chosen for decay steps and rate are contingent on factors such as dataset size and the desired pace of learning rate reduction, encapsulating pertinent considerations.

The results as presented in the Table 7.3 reflects the evaluation of the trained neural network model's performance on both the training and testing datasets. The "Training Score" of 0.2215 indicates the average loss incurred during the model's predictions on the training data. In conjunction, the "Training Accuracy" of 0.9017 signifies the proportion of correctly predicted outcomes in the training set.

Metric	Value
Training Loss	0.2215
Training Accuracy	0.9017
Testing Loss	0.1567
Testing Accuracy	0.9283

Table 7.3: Neural network evaluation metrics

On the other hand, the "Testing Score" of 0.1567 denotes the average loss when the model is applied to the unseen testing data. This is indicative of the discrepancy between predicted and actual values. The "Testing Accuracy" of 0.9283 reflects the accuracy of the model's predictions on the testing set. Figure 7.2 clearly shows the accuracy and loss of the model.

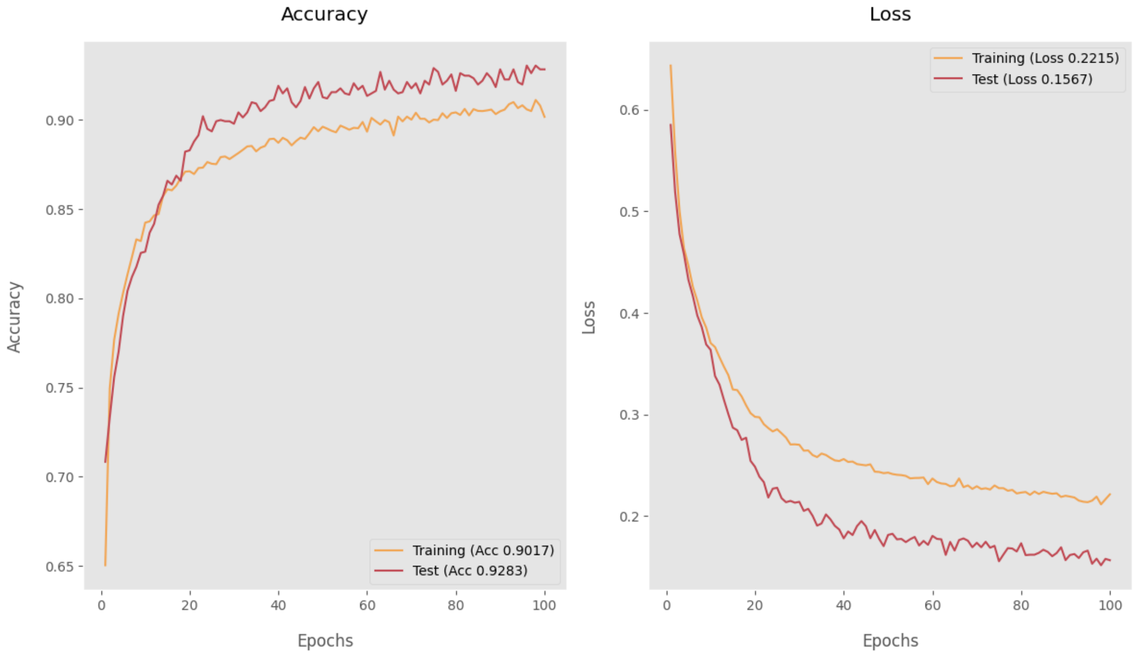

Figure 7.2: Accuracy and loss score of neural network

Additionally, the impressive AUC Score as shown in the Figure 7.3 with value of

0.999 underscores the model's robust discriminatory power in distinguishing between classes. A high AUC value, close to 1, signifies an exceptional ability to distinguish between positive and negative instances, further validating the model's efficacy.

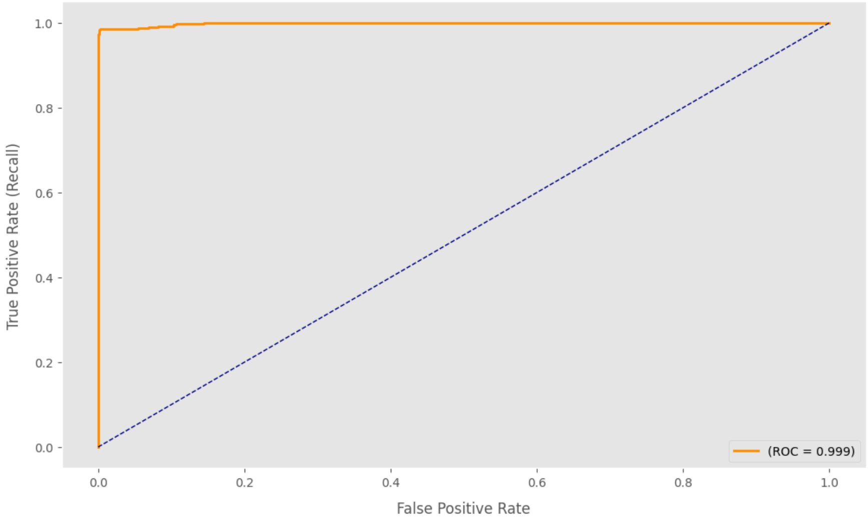

Figure 7.3: ROC curve of neural network model

7.4 Analysis of Profit Maximisation in Churn

The focus of the study is to determine the appropriate strategy to react to customers based on their predicted churn rates and their Customer Lifetime Value (CLTV). The goal is to identify customers who need attention and targeted efforts for retention, ultimately aiming to maximise profitability and customer loyalty. The customers with a high CLTV are collected and compared with predicted high churn.

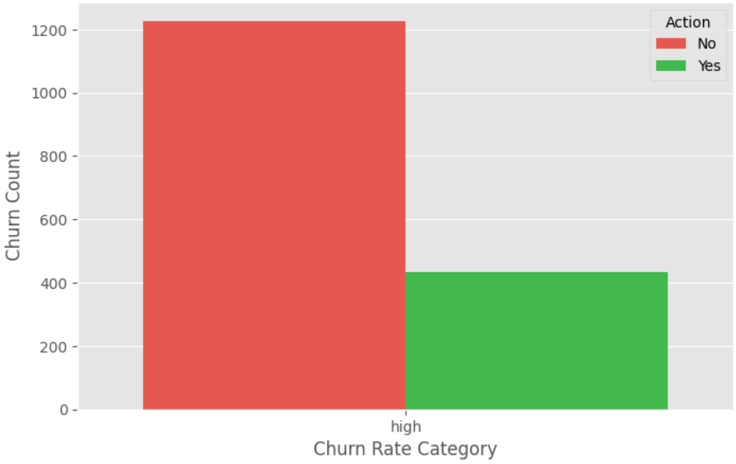

Figure 7.4: Churn rate category (high) vs. action

The customers that shows high in both cases are the ones that need to be given attention for retaining them in the business. Based on the analysis, it can be seen in the Figure 7.4 the amount of churned customers that need to be given recognition else they will likely leave the business.

CHAPTER 8

Conclusion and Future Work

In summary, the research employed a selection of five algorithms to forecast customer churn. The outcomes indicated that the Random Forest Classification model, particularly when coupled with feature selection tasks, exhibited superior performance when compared to the remaining four algorithms and prior models. Moreover, the study underscored the pivotal role and efficacy of feature engineering and selection in augmenting and enhancing model performance. Additionally, once customer churn is identified, it becomes imperative to present alternative strategies for addressing the churn rate and prioritising actions accordingly. Opportunities for further refinement remain, such as exploring the potential of neural network models in contrast to conventional approaches. Given the ongoing daily generation of data, continued experimentation remains essential to unearth additional insights and advancements.

Bibliography

[1] Predicting customer churn at a swedish crm-system company. https://www.diva-portal.org/smash/get/diva2:727881/FULLTEXT01.pdf. Accessed: 2023-07-14.

[2] Narendra Singh, Pushpa Singh, and Mukul Gupta. An inclusive survey on machine learning for crm: a paradigm shift. 2020.

[3] Hemlata Jain, Ajay Khunteta, and Sumit Srivastava. Telecom churn prediction using cnn with variational autoencoder. 2021.

[4] Telco customer churn. https://www.ibm.com/docs/en/cognos-analytics/11.1.0?topic=samples-telco-customer-churn. Accessed: 2023-06-28.

[5] Business intelligence and analytics: From big data to big impact. https://doi.org/10.2307/41703503. Accessed: 2023-07-15.

[6] Khulood Ebrah and Selma Elnasir. Churn prediction using machine learning and recommendations plans for telecoms. 2019.

[7] Xiaokai Zhang, Zhaojing Zhang, Dong Liang, and Hao Jin. A novel decision tree based on profit variance maximization criterion for customer churn problem. 2018.

[8] Sebastiaan Höppner, Eugen Stripling, Bart Baesens, Seppe vanden Broucke, and Tim Verdonck. Profit driven decision trees for churn prediction. 2017.

[9] Thomas Verbraken, Wouter Verbeke, and Bart Baesens. A novel profit maximizing metric for measuring classification performance of customer churn prediction models. 2012.

[10] Arno De Caigny, Kristof Coussement, and Koen W. De Bock. A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. 2018.

[11] Yu Zhao, Bing Li, Xiu Li, Wenhuang Liu, and Shouju Ren. Customer churn prediction using improved one-class support vector machine. 2005.

[12] Anuj Sharma and Dr. Prabin Kumar Panigrahi. A neural network based approach for predicting customer churn in cellular network services. 2013.

[13] Shobana J, Ch. Gangadhar, Rakesh Kumar Arora, J. Bamini P.N. Renjith, and Yugendra devidas Chincholkar. E-commerce customer churn prevention using machine learning-based business intelligence strategy. 2023.

[14] Supangat Supangat, Mohd Zainuri Bin Saringat, Geri Kusnanto, and Anang Andrianto. Churn prediction on higher education data with fuzzy logic algorithm. 2021.

[15] Alanoud Moraya Aldalan and Abdulaziz Almaleh. Customer churn prediction using four machine learning algorithms integrating feature selection and normalization in the telecom sector. 2023.

[16] Hemlata Jain, Ajay Khunteta, and Sumit Srivastava. International journal of intelligent systems and applications in engineering telecom churn prediction using an ensemble approach with feature engineering and importance. 2022.

[17] Huan Liu and Lei Yu. Toward integrating feature selection algorithms for classification and clustering. 2021.

[18] Jiayin Qi, Li Zhang, Yanping Liu, Ling Li, Yongpin Zhou, Yao Shen, Liang Li, and Huaizu Li. Adtreeslogit model for customer churn prediction. 2008.

[19] Sandhya Gangadharan. Exploring effective feature selection methods for telecom churn prediction. 2020.

[20] Ali Rodan, Ayham Fayyoumi, Hossam Faris, Jamal Alsakran, and Omar S. Al-Kadi. Negative correlation learning for customer churn prediction: A comparison study. 2015.

[21] Mark Eastwood and Bogdan Gabrys. A non-sequential representation of sequential data for churn prediction. 2009.

[22] Yin Wu, Jiayin Qi, and Chen Wang. The study on feature selection in customer churn prediction modeling. 2009.

[23] Mahmoud SalahEldin Kasema, Mohamed Hamadab, and Islam Taj-Eddinc. Customer profiling, segmentation, and sales prediction using ai in direct marketing. 2023.

[24] Mohammad Hossin and Sulaiman M.N. A review on evaluation metrics for data classification evaluations. 2015.

[25] Sofia Visa, Brian Ramsay, Anca Ralescu, and Esther van der Knaap. Confusion matrix-based feature selection. 2011.

[26] Andrew P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. 1997.

[27] 1.6. nearest neighbors. https://scikit-learn.org/stable/modules/neighbors.html#classification:~:text=1.6.2.%20Nearest%20Neighbors%20Classification%C2%B6. Accessed: 2023-07-18.

[28] sklearn.ensemble.randomforestclassifier. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. Accessed: 2023-07-20.

[29] Parameters of the rbf kernel. https://scikit-learn.org/stable/modules/svm.html#parameters-of-the-rbf-kernel:~:text=the%20RBF%20Kernel-,%C2%B6,-When%20training%20an. Accessed: 2023-07-27.

[30] Angelos P. Markopoulos, Dimitrios E. Manolakos, and Nikolaos M. Vaxevanidis. Artificial neural network models for the prediction of surface roughness in electrical discharge machining. 2007.

[31] 3.1. cross-validation: evaluating estimator performance. https://scikit-learn.org/stable/modules/cross_validation.html#:

~:text=validation%3A%20evaluating%20estimator-,performance,-%C2%B6. Accessed: 2023-08-02.

[32] sklearn.linear_model.logisticregression. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html. Accessed: 2023-08-05.

APPENDIX A

Code for Churn Prediction Analysis

A.1 Steps to run code in Google Colab

1. Download "200735_churn_pred.ipynb" code file and dataset from Faser.
2. Upload dataset on the google drive.
3. Click here to open google colab.
4. Upload the ".ipynb" code file on google colab.
5. Set the path of the dataset location in the code file.
6. Run the code cell by cell.