University of Essex
Department of Mathematical Sciences

MA981: DISSERTATION

A Comparative Analysis of Customer Churn Prediction in Banking Industry

Asish Kumar Pattnaik

Supervisor: Dongjiao Ge

November 26, 2021
Colchester

Contents

CHAPTER 1

Abstract

Due to rapid digitization and growth of internet, now a days consumers have much more choices than before. With information being easily available, consumers do not hesitate to switch their services as per their required needs. Customer churn happens, when the customer is not pleased with the current company product and services and found a better service provider for their needs. Customer churn is a prominent problem faced by many industries and banking industry is no exception. The cost involved in gaining a new customer being significantly high, it is more important for the banks to retain their existing customers by implementing a churn prediction models. Machine learning classification algorithms can be used to predict whether the customer will leave the company or not. This paper gives a comparative analysis of 6 customer churn prediction models that are Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbor (K-NN), Decision Tree (CART), Random Forest, and Gradient Boosting algorithm with a banking domain dataset consist of 10000 records of customers with 12 features and their effectiveness in predicting the customer churn. The result of the experiment showed that Gradient Boost algorithm gave the overall best performance in terms of cross validation accuracy, precision and recall score.Gradient Boost algorithm achieved a cross validation accuracy of 86.08%, precision score of 75.57% and recall score of 48.06% and can be used as a customer churn prediction model for the banks to get better result.

List of Figures

List of Tables

CHAPTER 2

Introduction

The growth of any organization depends upon their customer base. Organizations are aware of this fact that to sustain and grow in a competitive market they need to focus on customer and their needs. Every organization have a customer relationship management (CRM) team which solely focus on the customer satisfaction and to develop strategies to retain and grow their customer base. Churn is when an existing customer of a company refuse to use the product and services of the company and switch to a different company for their needs. In banking industry prospective, when an existing customer closes all the bank accounts and stops doing banking with the current bank in a given time period. Though the churn rate is low in banking industry, but the losses occurred due to the churn is huge. According to Ling Xie et al. [1] the cost of gaining a new customer is 5-6 times higher than retaining an existing customer. Therefore, it is crucial for the customer relationship management (CRM) of the bank to take appropriate action to prevent the customer churn. To retain the existing customers, it is not possible for any bank to give promotions and discounts to all the customers as it will attract more loss than profit. So, CRM team of the organizations should implement some strategies and techniques to identify only those customers who has a higher chance of leaving. And design a retention program, which will ultimately lead to profit for the organizations. Customer churn is a continuous process, no organization can stop this completely, but what they can do is to implement some good machine learning predictive algorithms which can detect churn customer earlier, so that they can be retained.

Customer retention plays a very big part in revenue generation for a financial institution.

Unlike e-commerce industry where new customers can come and buy product and services more frequently. Banking industry product and services are limited. Most of the customers do not change banks frequently, so most of the profit drives from the existing customers. Existing customers are more likely to avail other new experimental services provided by the bank that will again add to the profit and growth. Having a loyal customer base add advantages to the reputation of the bank which will again attract more new customers. So, it is very crucial for the bank to set up strong customer retention plan and avoid customer churn.

In this dissertation, we regard customer churn prediction as a binary classification problem. A number of machine learning classification methods are applied to solve this binary classification problem. There already exist many research works on the customer churn prediction. Some of the algorithms used in those studies are, Logistic regression [2], decision tree [3], random forest [4] [5], boosting algorithms [6] [7], transfer learning based algorithms [1], Naive Bayes [8], support Vector Machine (SVM) [9], Neural network based algorithm [2]. Few of the research have done comparative analysis of different machine learning techniques to find the best algorithm for the customer churn problem [10] [11] [12] [13] [4].

With all these existing works for machine learning methods on customer churn, this dissertation aims at investigating the problem "Which is the best classification algorithms for customer churn prediction in banking industry?". To address this problem, we have compared six different classification models. Those are Logistic regression, K-Nearest Neighbor (K-NN), Decision Tree (CART), Support vector Machine (SVM), Random Forest and Gradient Boosting algorithm on a banking dataset.

To achieve the objective of finding the best classification algorithm for the customer churn problem, the following steps need to be followed. The first step is to analysis the banking dataset to understand different features and their impact on the target variable by performing an exploratory data analysis and drawing insight from the data. Second step is to pre-process the dataset using different techniques like data cleansing, feature transformation, feature scaling to make it ready for training and prediction. Third step is to split the dataset into training and test set and training and testing all six models using cross validation techniques to stop over-fitting and to get a better accuracy. Fourth step is to search for best hyper parameter for all the selected models using *RandomizedSearchCV()* and perform hyper parameter tuning for all the models to achieve better accuracy. Fifth step is to calculate different confusion matrix like accuracy, precision, recall for model evaluation. Finally, the last step is to compare all the model performance based on evaluation metrics and obtaining the best algorithm for the customer churn prediction.

The key contribution of this paper is to analyse commonly used machine learning classifiers to find out which will be a better classifier to implement customer churn in a banking sector. We found out that ensemble learning based classifiers are giving better accuracy than the traditional classifiers.

The remaining part of the paper organised as follows. Section 2 Literature Review, which will address different work done by different researchers and proposed models. Section 3 Data Description gives insight of the dataset, features and their relationship with the target variable. Section 4 Methodology gives details about the data pre-processing, different algorithms used for the experimentation, model tuning and predictions. Section 5 Numerical Results shows the comparative results of the study. Section 6 Conclusion gives the conclusion about the study and future scope of work for this paper.

CHAPTER 3

Literature Review

Customer churn problem is one of the most common but challenging issues faced by many industries like Banking, Retail, Telecommunication Industries. Many research have been taken place over the years. In 2008, Xie, Yaya, and Xiu Li.[6], proposed a hybrid classification algorithm which combines Leaner discriminant analysis (LDA) with boosting framework, called Leaner discriminant boosting (LD-Boosting) which perform better than most conventional algorithms for churn prediction. The dataset used for this study is taken from a major Chinese bank data warehouse. It consists of 20,000 customer records. Then a random sample of 1524 taken for experimentation. The train, validation and test split were same for all experiments. To evaluate the model accuracy, top decile lift and lift curve criteria were chosen. The LD-Boosting algorithm was compared with Artificial neural network, Decision tree, SVM and AdaBoost algorithms. The result of the experiment shows that LD-boosting algorithm outperforms all other conventional classification models.

In 2011, Ling Xie, et al.[1] in their paper has used feature selection-based transfer ensemble (FSTE) model to predict the customer churn. The model used the power of the transfer learning along with the combination of different classifiers which exert in their preponderant space to improve the performance of the model. Two datasets were used for this experiment, one from the UCI dataset for telephone customers and second one is from Chongqing commercial bank credit card customer dataset. Both the datasets were highly imbalanced. The FSTE model was compared with other transfer learning models such as traditional method by utilizing all data (TMA), traditional method by utilizing target domain data only (TMOT), transferred feature selection (TFS), instance-based transfer learning strategy (TrBagg) and instance-based transfer learning strategy (TrAdaboost). Accuracy and AUC curve were used to evaluate the models. From the experiment, researchers concluded that, FSTE model performs better than other models for customer churn prediction.

In 2012, Lu, Ning, et al.[7] in their paper, tried to use boosting algorithm to create two clusters, one for high-risk customers for churn and another for low-risk customers based on the weights assigned by boosting. As a base classifier logistic regression was used for building models for both clusters. The dataset taken from a tele communication company customer data. 7190 random sample was used for training the model where number of churn customer were 678. And for testing the model next 6 months customer data were taken. The dataset has 700+ variables. The variables were reduced to 70 by using chi-square automatic interaction detection (CHAID)analysis and further with the help of decision tree and forward selection method, the variables were reduced to 21 for modelling. The gentle AdaBoost algorithm and logistic regression were compared in this study and ROC curve was used to evaluate the model. From the experiment, the result shows that using boosting algorithm with logistic regression gave better result than logistic regression alone.

In 2013, Sun, Peng, et al.[14] proposed a customer churn model based on Bayesian network. The dataset was taken from a post office company in China. The Bayesian network model was able to produce a predictive accuracy of 69.8%.

In 2015, Ismail, M. R. et al.[2] in their paper proposed a multi layered perceptron (MLP) based neural network algorithm for customer churn prediction by comparing it with logistic regression and multiple regression models. The dataset was taken from a Malaysian based telecommunication company, which consist of customer demographics, relationship data, billing, and usage data. Accuracy, specificity, and sensitivity were used for the evaluation of the models. The MLP based neural network model was compared with logistic regression and multiple regression models. The experimental result showed that, MLP based neural network algorithm outperformed logistic regression and multiple regression model with prediction accuracy of 91.28%.

In 2015, F. Guo and H. Qin[3] proposed a decision tree-based algorithm for customer churn prediction in e-commerce industry. The dataset was taken from a food distribution site and the records contains customer trading record of 2012. The dataset contains 31000 customers of which 8627 records are of churn customer. Decision tree algorithm was used for model building. The model was able to achieve an accuracy score of 88.53%. The results of the experiment showed that decision tree-based model for customer churn is proven to be a good and effective model.

In 2016, Dalvi, P.K et al.[15] proposed datamining-based algorithms for predicting customer churn in telecommunication industry. In their study, they compared both logistic regression and decision tree algorithm, but the result of the experiment was open ended. They let the user decide which model to choose based on the analysis they obtained.

In 2017, Farshid Keynia et al.[10] in their paper gave a comparative analysis of different data mining techniques for customer and employee churn prediction in a large organization. The dataset was taken from a large organization for a period of one and half years. The employee dataset consists of 9237 records of which 2235 were churn employees and for customer dataset consist of 9239 records of customer of which 1415 were churn customer. The datasets were highly imbalanced. For comparison, decision tree, naive bayes, SVM and neural network-based models were selected. To measure the model performance accuracy was used as an evaluation metrics. The result showed that for employee churn Naive bayes algorithm and for customer churn SVM model gave the highest accuracy. So, the researchers concluded that, for both employee and customer churn prediction SVM based model should be preferred.

In 2017, Sisodia, Dilip Singh et al.[11] in their paper, gave a comparative analysis of different machine learning algorithms to find the best algorithm for employee churn prediction. The dataset used for this experiment was taken from a HR analytics dataset from Kaggle. SVM, Decision Tree, Random Forest, k-nearest neighbor, and NaÃ¯ve Bayes classifier were used for churn prediction. The confusion matrix was used for the evaluation of the algorithms. The result of the experiment showed that Random Forest based classifiers outperformed all other classifiers.

In 2018, Alamsyah, Andry et al.[12] in their paper, performed a comparative analysis of different machine learning algorithms for employee churn prediction in telecommunication industry. The dataset was taken from Human Resource Information System (HRMS) of an Indonesian telecommunication company. Decision Tree, Random Forest and Naive Bayes algorithms were put into test for the prediction. Confusion Matrix was taken as evaluation metrics. The result from the experiment showed that, Random Forest based classifier performed best with an accuracy of 97.5% compared to other two algorithms.

In 2018, Asthana, Praveen[16] in his paper, demonstrated that, by applying boosting algorithm on conventional classifiers like decision tree, SVM and Artificial neural network algorithms can increase the accuracy by 1% to 4% and F-measure between 4.5% to 15%. The dataset was taken from UCI Machine Learning repository. The classification algorithms were tested 2 times. One without any boosting and second one with boosting. AdaBoost algorithm was used for boosting. For evaluation, Accuracy, Precision, Recall and F-measure was used. The experimental result showed that, by implementing boosting on the conventional algorithms the model performance can be increased. The best model with highest accuracy and F-measure was found to be boosted SVM model with 97% accuracy and 84% f-measure.

In 2018, Sabbeh, Sahar F[13] performed a comparative analysis of all the available state of the art classification techniques for customer churn problem in telecommunication industry. The dataset used for this study was taken from a telecommunication company. Decision Tree, Support vector machine (SVM), K-nearest Neighbor, AdaBoost, Random Forest, Stochastic gradient boost, ANN, Naive Bayes, Leaner Discriminant Analysis were used form comparison. Accuracy was used for the evaluation of the models. The result of the experiment showed that, ensembled learning based techniques like random forest and Ada Boost algorithms produces highest accuracy.

In 2018, M. Spiteri and G. Azzopardi[4] in their paper, tried to find the best algorithm for the customer churn prediction using different conventional algorithm for an auto insurance company. The dataset was taken from a Malta based auto insurance company. The data collected over the period of one year and the dataset consist of 72445 policy holder details including, policy details, coverage details, premium details. Decision Tree, Logistic Regression, Naive Bayes, Random Forest, support Vector Machine (SVM) and survival analysis algorithms were put into test. Accuracy was taken as the evaluation measure. From the experimental result, researchers have concluded that Random Forest algorithm was able to achieve the highest accuracy of 91.18% and can be used as a better churn predictor for the insurance company.

In 2018, K. Kim and J. Lee[8] in their paper, tried to improve the performance of the deep learning-based churn prediction model using Bayesian optimization. In their experimentation, they were able to improve the model performance by 9.4% from the manually optimized model.

In 2019, K.G.M. Karvana et al.[9] in their study, utilized the data mining techniques to predict the customer churn in banking industry. The dataset has been taken from a private bank in Indonesia which has 57 attributes. The dataset consists of demographics data, transnational data, and average monthly balance data. To analysis furthermore, they have taken 3 samples of data with a ratio of 50:50, 30:70 and stratified sampling of 1% to find out which sample class produces highest value of accuracy and sensitivity. Decision tree, neural network, support vector machine (SVM), naive bayes and logistic regression classification models were used for this research. Based on the results, researchers concluded that support vector machine (SVM) with 50:50 sampling data gave the best result for customer churn prediction.

In 2019, I. Ullah et al.[5] in their research paper proposed a random forest and clustering based classifier for churn prediction for telecom industry. The dataset used for this study was taken from a south Asian telecom company. The dataset consists of usage data, marketing data and financial data. The random forest-based model was compared with other conventional as well as hybrid algorithms for churn prediction. The models were evaluated using model accuracy, recall, precision, F-measure, and ROC curve. The result of the experiment showed that random forest-based classifiers was performed better than other algorithms with 88.63% accuracy.

In 2020, X. Hu et al.[17] in their paper, proposed a hybrid model for the customer churn problem. They have taken a decision tree and artificial neural network-based hybrid model to predict the customer churn. The dataset was taken from a supermarket database of 2681 customers between June 2018 to April 2019. The hybrid model was compared with the individual decision tree model and artificial neural network model. Accuracy was taken as the evaluation measure. From the result of the experiment, researchers have concluded that, decision tree and artificial neural network-based hybrid model is able to achieve a prediction accuracy of 98.87% compared to decision tree based and artificial neural network models.

CHAPTER 4

Data and Methodology

4.1 Data Description

The dataset considered for this experiment contains 10000 records of customers data of a retail bank. The dataset consists of 13 features variable and one target variable.The dataset contains customer demographics data, geographic data, and account related data. The detailed about the columns are mentioned in Table 4.1.The dataset can be found in Kaggle.[1]

The summary statistics of the dataset has given in Table 4.2. From Table 4.2, following points can be drawn:

- The dataset consists of 10000 records.
- Customer credit score ranges from 350 to 850.
- Customers are in the age range of 18 to 92.
- The tenure considered for this dataset is 10 years.
- The customers are using 1 to 4 products of the bank.
- The dataset is quite clean dataset with no null values.

There were 2 columns **CustomerId** and **Surname** which are not significant for the prediction of target variable, so those columns can be removed. Now the dataset consists of 6 categorical

[1] Dataset: https://www.kaggle.com/mathchi/churn-for-bank-customers

Column Name	Description
CustomerId	Contains unique values related to customer
Surname	Surname of the customers
CreditScore	Credit score of the customer
Geography	Customer geographic location
Gender	Gender of the customer
Age	Age of the customer
Tenure	Number of years customer associated with the bank
Balance	Account balance of customer
NumOfProducts	Number of banking product used by the customer
HasCrCard	Whether a customer has a bank provided credit card or not
IsActiveMember	Is customer an active member or not
EstimatedSalary	Salary of customer
Exited	Whether or not the customer has left bank

Table 4.1: Dataset Column Description

	Count	Mean	Std	Min	Max
CreditScore	10000	650.528800	96.653299	350	850
Age	10000	38.921800	10.487806	18	92
Tenure	10000	5.012800	2.8921740	0	10
Balance	10000	76484.889288	62397.405202	0	250898.09
NumOfProducts	10000	1.530200	0.581654	1	4
HasCrCard	10000	0.70550	0.45584	0	1
IsActiveMember	10000	0.515100	0.499797	0	1
EstimatedSalary	10000	100090.239881	57510.492818	11.58	199992.48
Exited	10000	0.203700	0.402769	0	1

Table 4.2: Summary Statistics

variables, 4 numerical variables and 1 target variable. Details about Categorical variables, target variable and their classes are mentioned in Table 4.3.

Categorical variable	**Class**
Geography	Germany, France, Spain
Gender	Male, Female
Tenure	1 to 10 years
NumOfProducts	1, 2, 3, 4
HasCrCard	1(Yes), 0(No)
IsActiveMember	1(Yes), 0(No)
Exited	1(Churn), 0(Non-churn)

Table 4.3: Categorical Variable and Class

4.1.1 Exploratory Data Analysis

To study the distribution of churn and non-churn customers in this dataset, We have plotted a pie chart in Figure 4.1, we can clearly see that out of 10000 customers about 20.4% of customer were churned.

To check the class distribution of different categorical variable, we have plotted bar plots to get a clear picture of different categorical variables. The bar plots can be seen in Figure 4.2. Below points can be drawn about the categorical variables.

- **Geography**: About 50% of the customer are from France and rest equal number of customers are from Spain and Germany.
- **Gender**: Male customer count is greater than that of female customer count.
- **Tenure**: This is a balanced graph. Almost equal number of customers fall under different tenure bracket.
- **NumOfProduct**: Around 50% customers have at least used 1 product and followed by around 45% of the customer have used 2 products and less than 5% customers have used more than 2 products.
- **HasCrCard**: Around 70% customers have credit card provided by the bank.

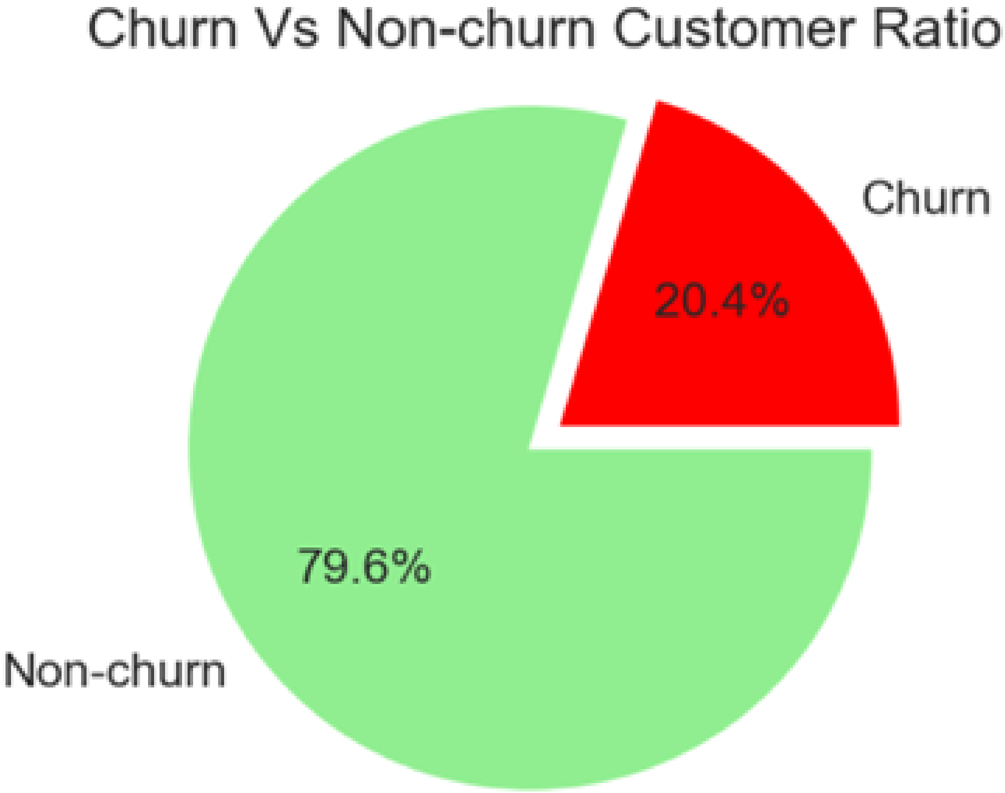


Figure 4.1: Churn Vs Non-Churn Customer Ratio

- **IsActiveMember**: Almost equal number of customers are active and non-active users.

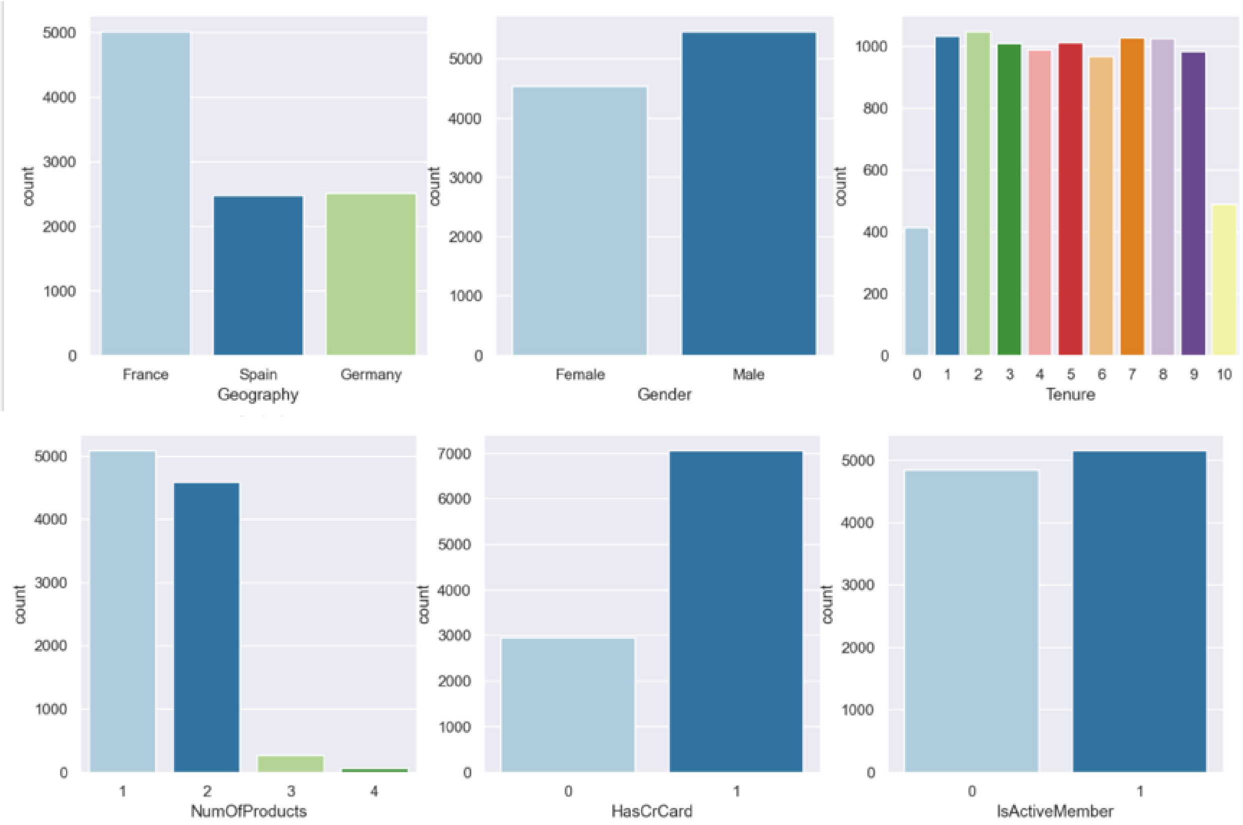


Figure 4.2: Categorical Variable Class Distribution

To check the distribution of numerical variable, we have plotted histograms to get a clear picture of different numerical variables. The histograms can be seen in Figure 4.3. Below insights can be drawn from Figure 4.3 about numerical variables.

- **CreditScore**: The credit Score plot is a bell-shaped curve with most of the customer has less than 700 credit score.

- **Age**: Age graph is right skewed. That means most of the customers are young and in the age range of 29-40 age. and very few customers are after the age of 60 years.

- **Balance**: The account balance is a normally distributed curve, but more than 3500 customers have a bank balance of 0.

- **EstimatedSalary**: The estimated salary of customers is evenly distributed among all the category of salary range.

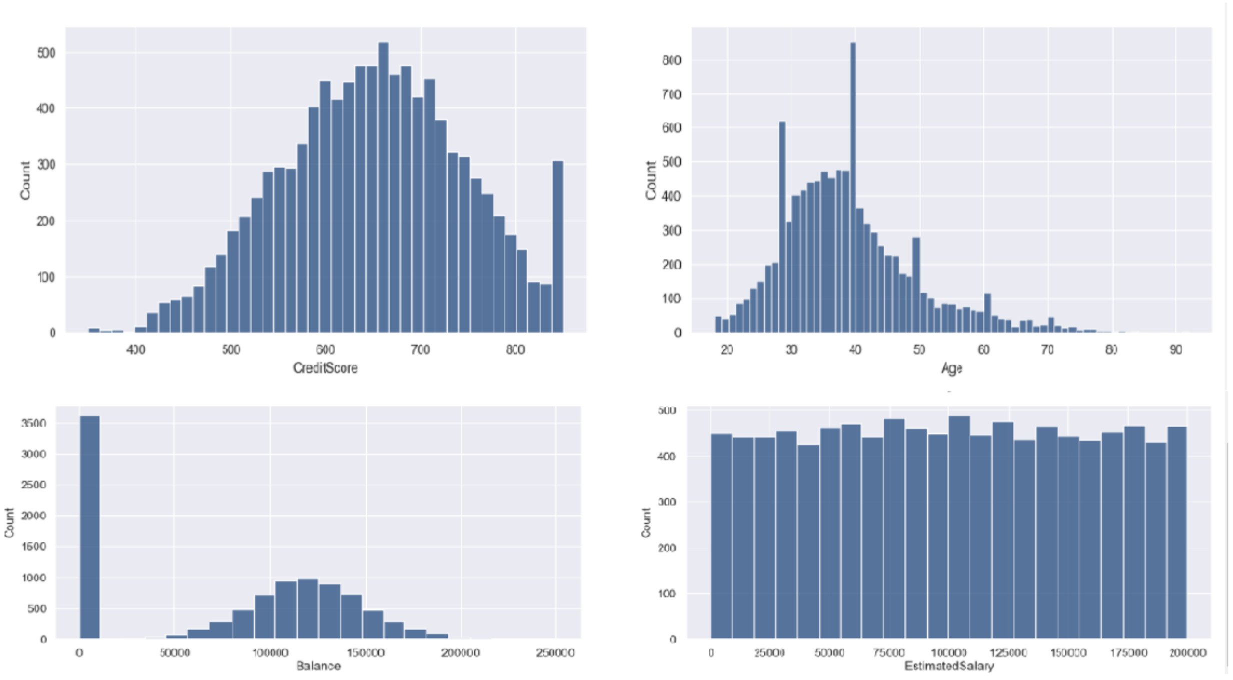
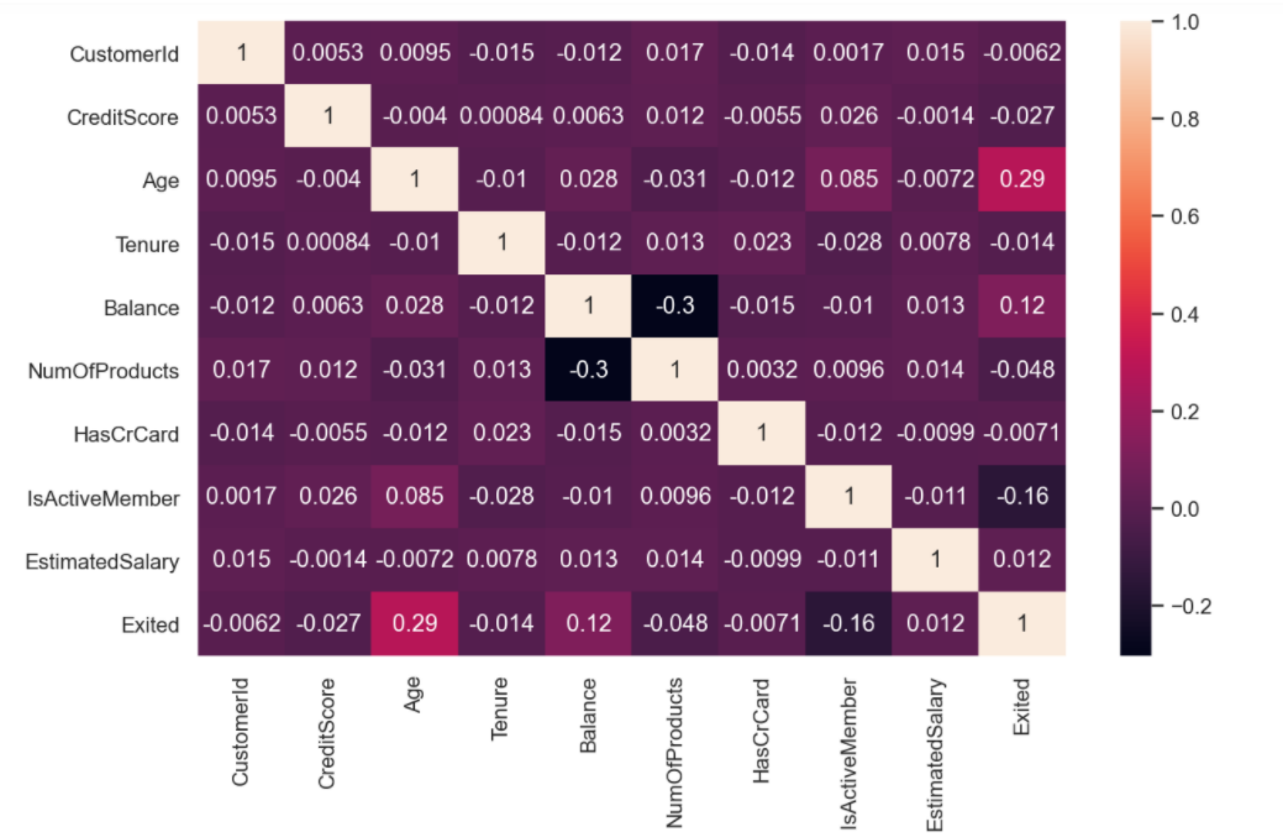


Figure 4.3: Numerical Variable Distribution

To check the correlation between all variable and target variable, we have plotted a correlation plot which can be seen in Figure 4.4. The correlation plot shows that **Age** has a positive correlation with **Exited** column. **Balance** has negative correlation with **NumOfProducts**. There is no other correlation found in between all the feature variables and with respect to target variable. A detailed exploration about the correlation of different input features with target variable (Exited) is covered in following section.

To understand the trend of the target variable with respect to all the feature variables where 1 means exited and 0 means non-exited, histogram plots were plotted in Figure 4.5. Below insights can be drawn.

- **CreditScore** (**Non-exited** vs **Exited**): Most of the customers having credit score in between 600 to 700 exited the bank more than the other credit score range.
- **Geography** (**Non-exited** vs **Exited**): Though France has most customers but the churn rate for Germany customer is the highest whereas the churn rate is lowest in France.
- **Gender** (**Non-exited** vs **Exited**): Female customers tends to churn more than male counterparts.
- **Age** (**Non-exited** vs **Exited**): Customers in the age bracket of 40-60 tends to churn more



	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
CustomerId	1	0.0053	0.0095	-0.015	-0.012	0.017	-0.014	0.0017	0.015	-0.0062
CreditScore	0.0053	1	-0.004	0.00084	0.0063	0.012	-0.0055	0.026	-0.0014	-0.027
Age	0.0095	-0.004	1	-0.01	0.028	-0.031	-0.012	0.085	-0.0072	0.29
Tenure	-0.015	0.00084	-0.01	1	-0.012	0.013	0.023	-0.028	0.0078	-0.014
Balance	-0.012	0.0063	0.028	-0.012	1	-0.3	-0.015	-0.01	0.013	0.12
NumOfProducts	0.017	0.012	-0.031	0.013	-0.3	1	0.0032	0.0096	0.014	-0.048
HasCrCard	-0.014	-0.0055	-0.012	0.023	-0.015	0.0032	1	-0.012	-0.0099	-0.0071
IsActiveMember	0.0017	0.026	0.085	-0.028	-0.01	0.0096	-0.012	1	-0.011	-0.16
EstimatedSalary	0.015	-0.0014	-0.0072	0.0078	0.013	0.014	-0.0099	-0.011	1	0.012
Exited	-0.0062	-0.027	0.29	-0.014	0.12	-0.048	-0.0071	-0.16	0.012	1

Figure 4.4: Correlation Plot

than the younger customers.

- **Tenure** (**Non-exited** vs **Exited**): Almost all the tenure category has around 15-20% churn rate.
- **Balance** (**Non-exited** vs **Exited**): Surprisingly customers with zero balance tends to churn less.
- **NumOfProduct** (**Non-exited** vs **Exited**): Customers who has used only one product tends to churn more and customers who has used more than 2 products tends to churn more. Customers using 2 products are less prone to churn.
- **HasCrCard** (**Non-exited** vs **Exited**): Customers having a credit card and not having a credit card tends to churn equally.
- **IsActiveMember** (**Non-exited** vs **Exited**): Customers who are not active members churn more.

- **EstimatedSalary** (**Non-exited** vs **Exited**): Customers in all the salary bracket are equally churned.

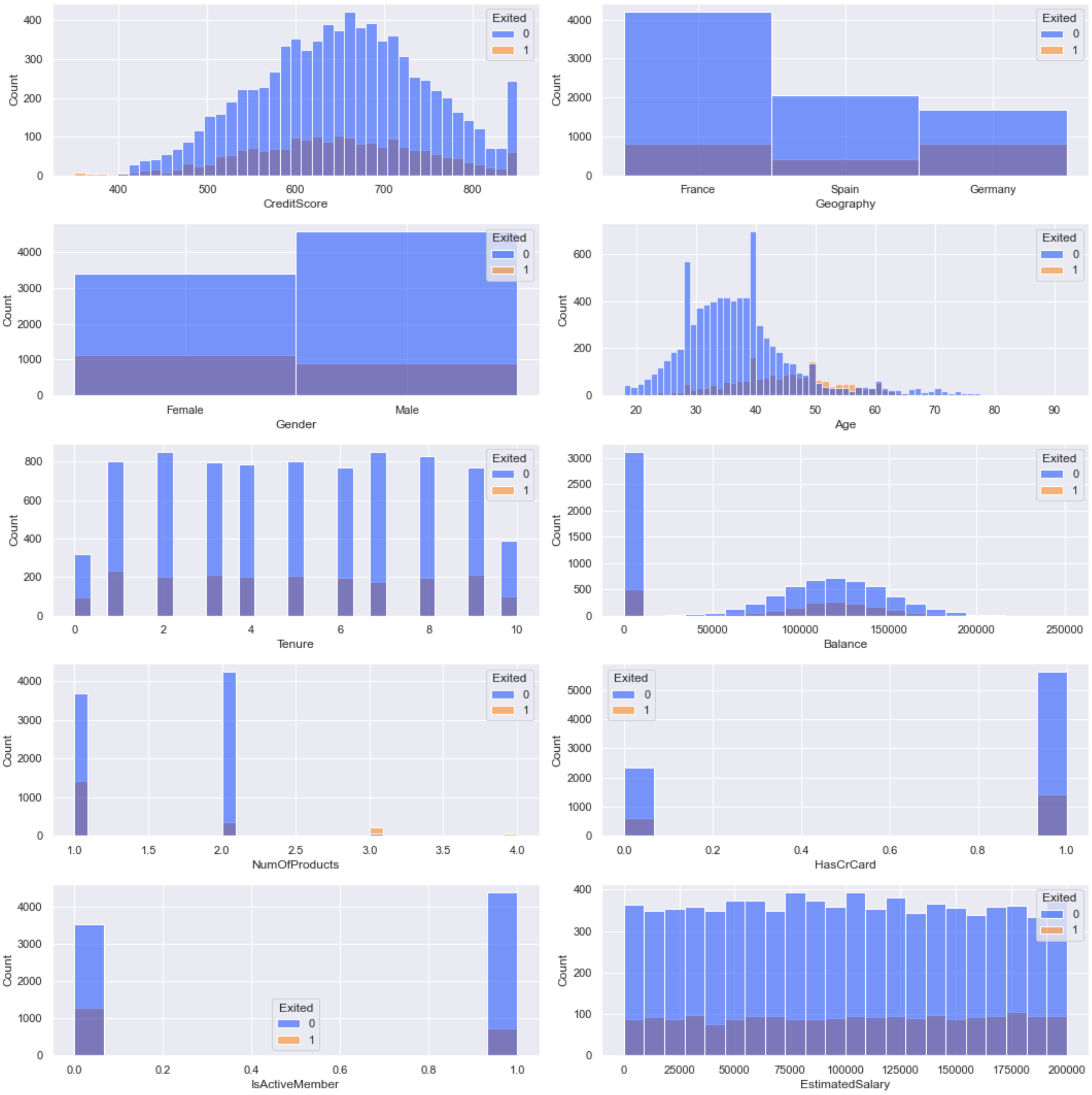


Figure 4.5: Feature Variables Vs Target Variable

4.2 Methodology

Customer churn is a binary classification problem which we want to solve. The motivation of this project is to find a best algorithm for customer churn prediction by comparing different classification models. In a summary, there are 4 modules which we need to perform in order to find the best algorithms for customer churn prediction. The first module is data preprocessing, the second module is model training and prediction. The third module is hyper-parameter tuning for better performance and finally model comparison based on evaluation metrics. The framework for this experiment can be seen in Figure 4.6. The whole process is done using programming language Python.

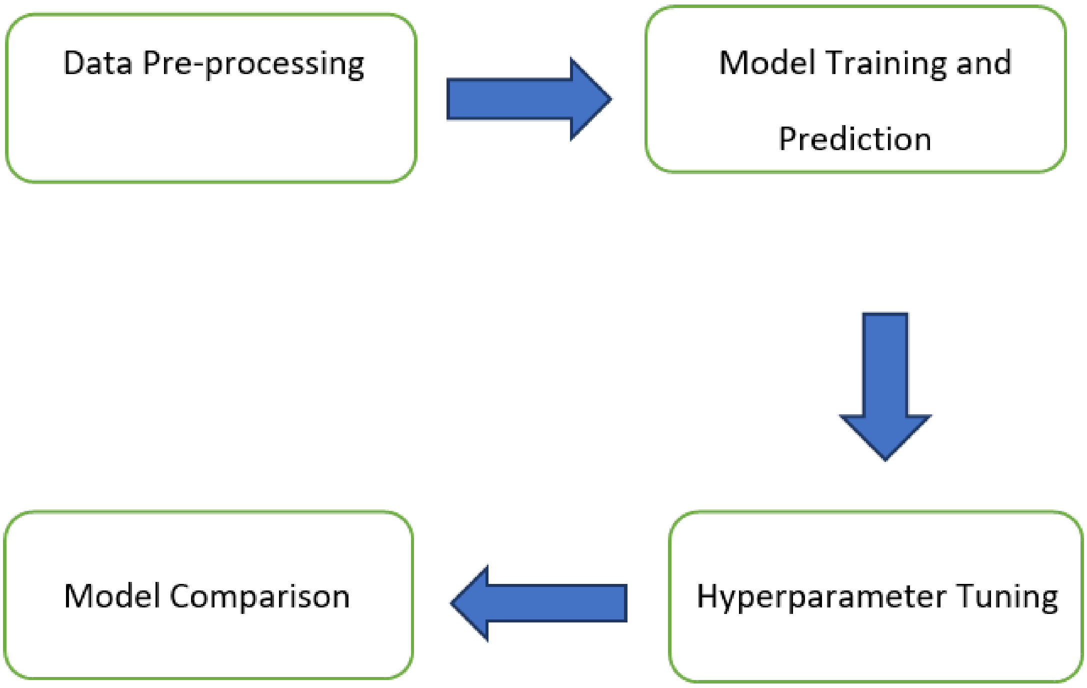


Figure 4.6: Experimental Framework

4.2.1 Data Preprocessing

Data preprocessing is one of the crucial processes in machine learning. A good and well processed dataset can give better prediction accuracy. To preprocess the data 4 steps were performed,

- Data cleansing
- Outlier detection and removal
- Feature Transformation

- Feature Scaling

Data cleansing process involves handling of missing data. As algorithms like Support Vector machine can not handle missing data. The missing values can be replaced with mean, median or zeros. The dataset was checked for missing values and duplicate values with the help of "*isnull()*" function and "*duplicate()*" function in python. But we found that the dataset is free from missing and duplicate values. This dataset is a clean dataset.

Outliers are the data points which are very unusual from the rest data points. Outliers have significant impact on the model performance. So, detection and removal of the outliers should be done before training the models. To detect the outliers in numerical features we have used capping method by setting an upper limit and lower limit at 95% and 5% as a threshold limit. After that we have created another function called "*check_outliers()*" to check the outliers in all the numerical columns. After applying the function in the numerical columns, the output shows that there are no outliers in all the numerical column. The dataset chosen for this study is found to be a very clean dataset.

Feature transformation involves converting the data into machine learning suitable format.To train machine learning models, first we need to convert the categorical data to indicator variables like 0 and 1. The categorical features like **Geography**, **Gender**, **NumberOfProduct**, **HasCrCard** and **IsActiveMember** is converted into indicator variables with the help of pandas "*get_dummies()*" function. The dataset is only one step behind the model training process. The next step is feature scaling.

Feature scaling is an important process to perform when different features are in different value range. And it is a good practice to bring all the numerical variable into a one scale range because there are few distance-based algorithms like K-Nearest Neighbor(K-NN) and Support Vector Machine(SVM) which need scaled input for the model training. To name a few, features like **CreditScore** is in the range of 350-850, **Age** is in the range of 18-92, **Tenure** in the range of 1-10. In this paper, as we have taken K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) for our experimentation, we have applied min, max normalization techniques so that all the numerical features will be in a standard range of 0 and 1.

After applying all the steps on the dataset, Our dataset is ready for the model training and prediction.

4.2.2 Machine Learning Classifiers

The customer churn and retention process are one of the main areas of Customer Relationship Management (CRM). To predict which customer is going to churn in future, organizations use different machine learning algorithms. The most widely used classification algorithms are Logistic Regression, K-Nearest Neighbor, Decision Tree based algorithms, Ensemble learning based algorithms and Neural Network based algorithms. The classifiers used for this study is given in this section.

Logistic Regression: Logistic regression [18] is a binary classification algorithm, which is used in machine learning and statistics to solve binary classification problem. Logistic regression is used to analyze the relationship between one dependent variable with binary output data and one or more independent variables. Suppose $X_1, X_2, ..., X_k$ are independent variables where k is the number of variables considered and α and $\beta_1, \beta_2, ..., \beta_k$ are unknown parameters with constant values which we need to estimate. then

$$\boldsymbol{Z} = \alpha + \beta_1 \boldsymbol{X_1} + \beta_2 \boldsymbol{X_2} + \ldots + \beta_k \boldsymbol{X_k} \tag{4.2.1}$$

Where Z is the leaner combination of the constants and the independent variable. And Y is the dependent variable. Logistic regression uses logistic function which can be defined as

$$\boldsymbol{P}(\boldsymbol{X}) = \frac{1}{(1 + e^{-Z})} \tag{4.2.2}$$

Where $\boldsymbol{P}(\boldsymbol{X})$ is the conditional probability of the dependent variable Y which can be either 1 or 0.

The output of the logistic function is always come between 1 and 0, which makes the logistic regression so unique. When a classification model has to find pass or fail, win or lose, healthy or sick type of output, logistic regression will be the first choice. As in this study we want to find whether the customer will churn or not, so logistic regression model is our first pick.

K-Nearest Neighbor: K-Nearest Neighbor (K-NN) [19][13] is the simplest classifier which uses memory-based learning techniques. New instances are labeled based on the previous instances. This model retains the whole training set instances during training and labeled the next instances based on the majority voting of its neighbors. Instances are labeled based on the neighbor class. The neighbors are determined by either Euclidian distance or Manhattan distance or Murkowski distance. For churn prediction K-NN will determine how close the features of one customer matches with the other customer, if the matching probability is greater than 50% then it will classify the customer either churn or non-churn as per the nearest customer. If the previous customer is churn and the next customer features are matching with a 50% or more probability, then the KNN model will labeled the next customer as churn. The advantage of this model is, its simple and immune to noisy training data.

[19] Suppose $TS = \{(x_j, y_j)\}_{j=1}^{N}$ where TS is training set and x_j is the training vector and y_j is the respective class label.Then for a x' data point and its unknown class y' can be found by selecting a set of k similar labelled target neighbor for the data point x' , then the dataset $TS^{'=\{(x_j{}^{NN},y_j{}^{NN})\}_{j=1}^{k}}$ is arranged with respect to Euclidean distance $d(x', x_j{}^{NN})$ in increasing order between x' and $x_j{}^{NN}$

$$\boldsymbol{d(x', x_j{}^{NN})} = \sqrt{(x' - x_j{}^{NN})^T(x' - x_j{}^{NN})} \tag{4.2.3}$$

then based on majority vote of the x' class label is predicted.

$$\boldsymbol{y'} = \arg max_y \sum_{(x_j{}^{NN},y_j{}^{NN})\epsilon T'} \delta(y = y_j{}^{NN}) \tag{4.2.4}$$

where y is the class label and $y_j{}^{NN}$ is the class label for the jth nearest neighbor among its k nearest neighbor. $\delta(y = y_j{}^{NN})$ equals to 1 if $y = y_j{}^{NN}$ or zero otherwise.

The K-NN model can be seen in figure 4.7.

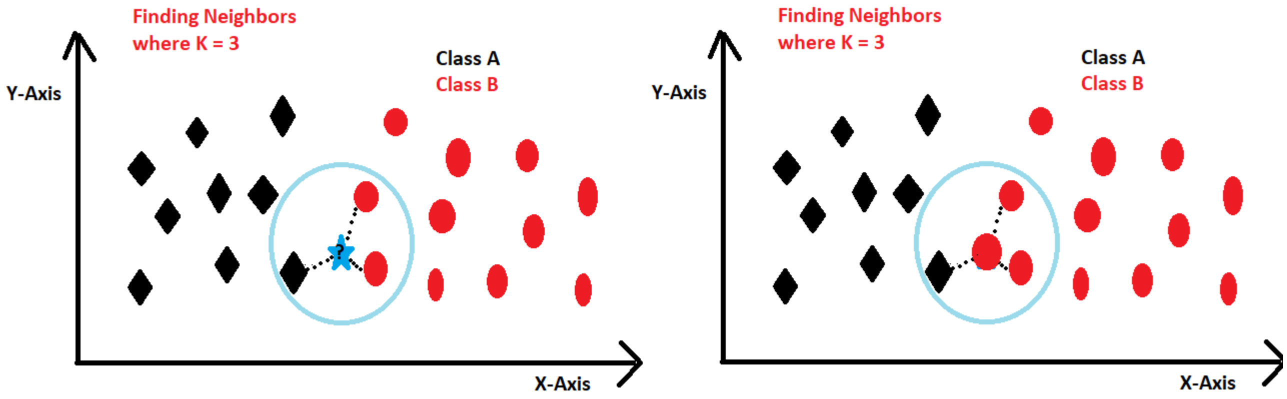


Figure 4.7: K-NN Classification Algorithm

Decision Tree (CART): CART [20][13] is a decision Tree (DT) based classifier which generate tree like structure based on the decisions and rules. CART follows binary recursive partitioning. The decision tree starts with training set as root node. Each node can be split into only 2 child nodes based on the Gini Index. The resultant child node is assigned a predicted class based on the rules and distribution and Gini index. The split node is called internal node. With Tree pruning CART creates sequence of simpler trees. This recursive process goes on until it is impossible to split again or when Gini index is 0, like each node have only 1 observation or each child node has identical distribution of predictor node, or an external tree depth was set. As no prior assumption are made in CART, it can easily handle all kind of numerical and categorical data. In CART Tree diagram each child node represent a predictor variable and each branch represent the outcome of the split and each child node represents the class labels. The Gini Index can be found for each node as.

$$\boldsymbol{GI} = \sum_{j=0}^{c} P_j(1 - P_j) \tag{4.2.5}$$

or

$$\boldsymbol{GI} = 1 - \sum_{j=0}^{c} P_j^2 \tag{4.2.6}$$

where P is the probability of class j and c is the total number of classes.

A simple decision tree algorithm is given in Figure 4.8 for better understanding.

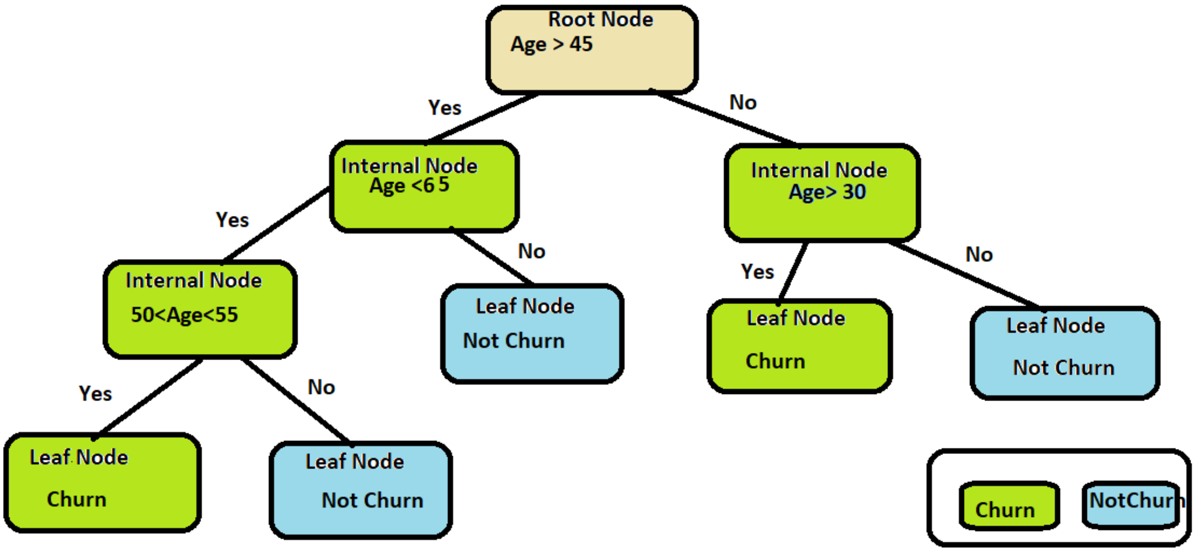


Figure 4.8: Decision Tree Algorithm

Random Forest: Random Forest [21] is an ensemble learning algorithm in which the forest is made with number of decision tree classifiers which are trained by bagging method.

Definition 4.2.1 *In 2001, L.Breiman published an article [21] in Machine learning, vol. 45 titled "Random Forests" "A random forest is a classifier consisting of a collection of tree-structured classifiers* $\{h(x, \Theta_k), k = 1, ...\}$ *where the* $\{\Theta_k\}$ *are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input* x*".*

Each decision tree gets trained by a subset of randomly selected features and gave a response, based on the response of all the trees, the model chose the response based on majority of the votes by the individual decision tree. As the response is selected based o majority voting, it protects against overfitting and the performance of the algorithm improves significantly. The only drawback of random forest is, if large number of trees are involved, it makes the algorithm slow. Due to its robustness to overfitting and simple and quick to build and use, it is one of the first choice for classification and regression problems. A basic random forest algorithm is shown in Figure 4.9.

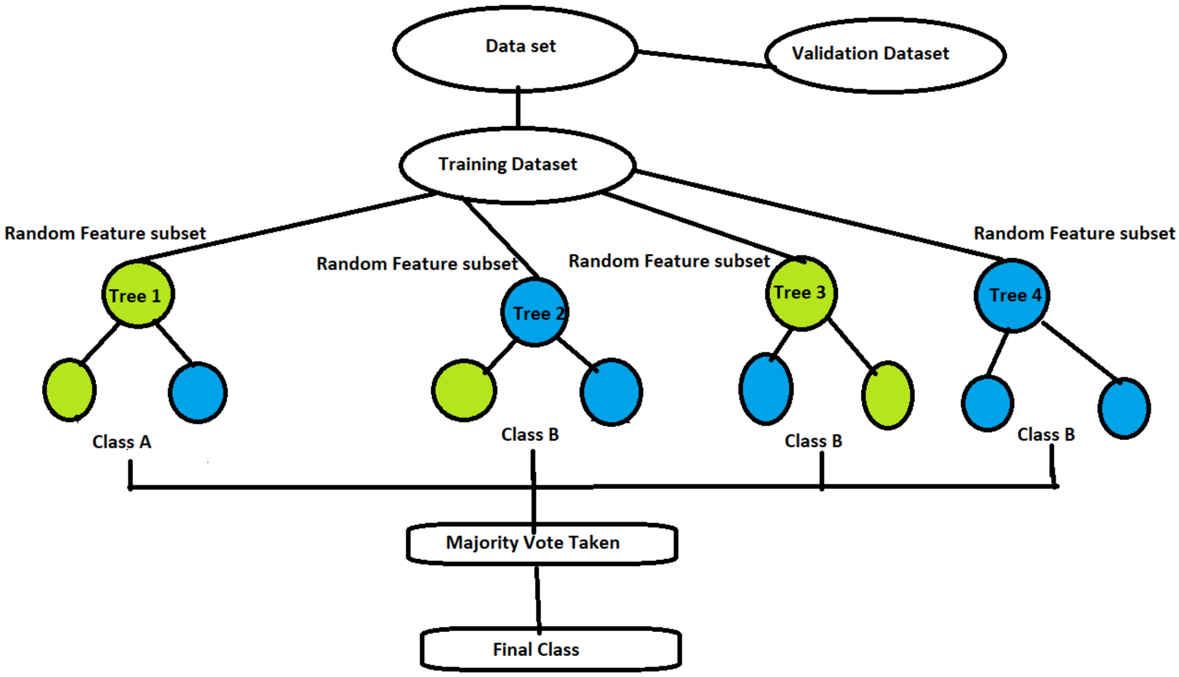


Figure 4.9: Random Forest Algorithm

Support Vector Machine: Support vector machine is a supervised learning algorithm which can be used for both classification and regression problem. The algorithm tried to separate the data points present in the training dataset based on classes by finding an optimal hyperplane which maximizes the margin between the classes. The datapoints which are present on the margin are called support vectors. The decision boundary was decided by the support vectors. This is the reason why the name is support vector machine. The algorithm tris to maximize the margin between the datapoints to achieve higher accuracy. If only two features are involved, then the decision boundary is straight-line. If 3 features are involved, then the decision boundary is a hyperplane. The graphical representation of Support vector machine (SVM) classifier algorithm can be seen in Figure 4.10.

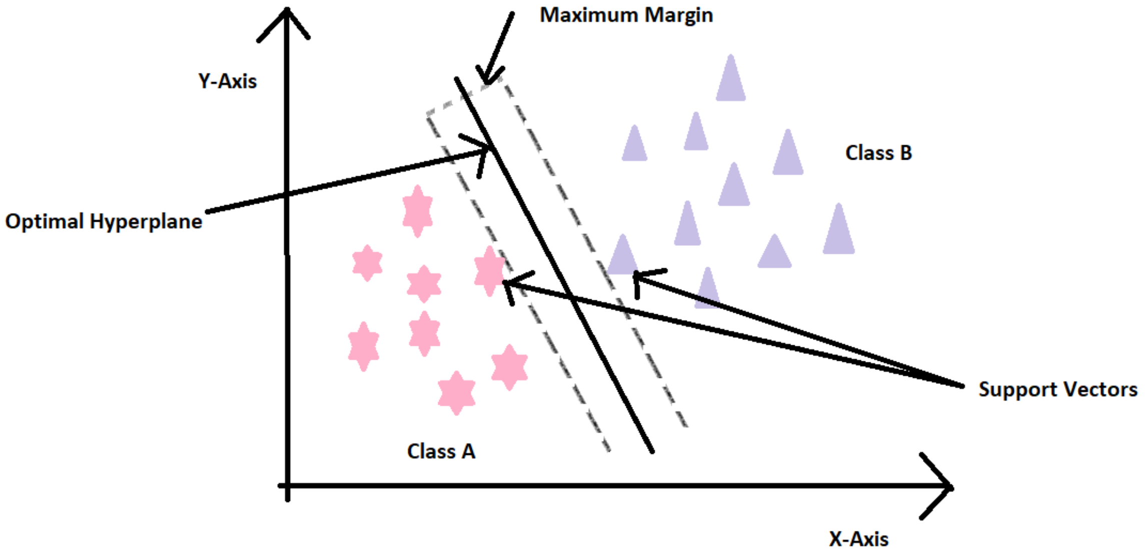


Figure 4.10: Support Vector Machine Algorithm

Gradient Boost: Gradient boosting algorithm is an ensemble learning algorithms in which the algorithm tries to predict the output with the help of weak learners. Three steps are involved for predicting the output. Firstly, the loss function to optimize. Secondly the weak learner to make prediction. The weak learn used in gradient boosting is decision tree. Trees are added to the model 1 at a time and once added no changes made to the existing tree. Gradient descent process is done to minimize the loss function. The whole process is done in sequential manner the next week learner decision tree learns from the previous weak learner and tries to minimize the error of the previous weak learner. The whole process of minimizing the loss function led to better accuracy. But as the model learns sequentially, the training process is slow. The gradient boosting algorithm gives better accuracy with default parameters also.

4.2.3 Model Training and Prediction

To train the selected classifiers, the dataset splits into training and test set by using *"train_test_split()"* with a splitting ratio of 80:20. The models are fitted with the help of *"fit()"* function. The fitted classifiers used for prediction and both training and test accuracy was calculated using the *"score()"* method. Initially the classifiers were trained and tested on their default parameter setting, just to see how well it predicts. The results of this experiment shown in Table 4.4

Model	Hyperparameter Setting	Training Accuracy	Test Accuracy
Logistic Regression	Default	84.04%	84.9%
K-Nearest Neighbor	Default	87.21%	82.4%
Support Vector Machine	Default	81.91%	82.2%
Decision Tree (CART)	Default	100%	79.15%
Random Forest	Default	100%	85.4%
Gradient Boost	Default	87.28%	85.95%

Table 4.4: Initial Training and Test Accuracy

Decision Tree and Random Forest classifiers got a training accuracy of 100% and a test accuracy of 79.15% and 85.4%, which seems like overfitting the training set. To avoid over fitting in training and test set, we have used K-fold cross validation techniques with K=10. The models were evaluated for cross validation accuracy with the help of *"cross_val_score()"* method which will prevent the over fitting due to same type of test and train set. The results shown in Table 4.5

Model	Hyperparameter Setting	CV Accuracy
Logistic Regression	Default	84.06%
K-Nearest Neighbor	Default	82.51%
Support Vector Machine	Default	82.01%
Decision Tree (CART)	Default	79.13%
Random Forest	Default	86.19%
Gradient Boost	Default	86.34%

Table 4.5: Initial Cross Validation Accuracy

By comparing the Table 4.4 and 4.5, we can see that there is slight difference between the test accuracy and cross validation accuracy. From confusion matrix, we have calculated Precision and Recall score to check the model performance. The details are shown in Table 4.6.

Model	**Precision**	**Recall**
Logistic Regression	76.44%	38.59%
K-Nearest Neighbor	62.50%	36.41%
Support Vector Machine	85.90%	16.26%
Decision Tree (CART)	49.40%	36.41%
Random Forest	74.00%	44.90%
Gradient Boost	77.41%	44.90%

Table 4.6: Initial Precision and Recall Score

For predicting the customer churn correctly, precision and recall are 2 important metrics, as we can see in the table 4.6 the Precision for Support vector machine classifier is 85.90% which is very good but the recall is just 16.26%, which is very bad score. The highest recall score achieved is 44.90% and we need to achieve a good precision score as well as good recall score. So now the classifiers need to tuned to get better result.

Hyperparameter tuning is a computational expensive and time-consuming process. But with best hyper parameters the classifiers can produce better results in prediction. The method used for the hyperparameter selection is called *"RandomizedSearchCV()"*. This method randomly selects the best hyper parameter from a set of inputs. There are other hyperparameter selection methods like Grid Search which is an exhaustive search approach. For Logistic Regression, we have not used any hyperparameter tuning as there are very limited hyperparameter to tune. For K-NN classifier the most important hyper parameter is *n_neighbors*, we have selected *n_neighbors*, *leaf_size*, *p* to tune. The range taken for *n_neighbors* is 1 to 30, *leaf_size* 1 to 50, and p is 1,2. For Support vector classifier, we have selected regularisation hyperparameter *C* which is a penalty parameter. Normally when *C* is higher, the classification accuracy is high. So, we have taken *C* in the range of 10 to 50 with increment of 5. For decision tree CART classifier, we have selected *max_depth*, *min_samples_leaf* hyperparameters, the range selected for *max_depth* is 2 to 16 with increment of 2 and for *min_samples_leaf* 1,2,4,6,8. For Random Forest classifier *n_estimators*, *max_depth*, *max_features*, *min_samples_split*, *min_samples_leaf* hyperparameters are selected to tune. For *n_estimators* the range used is

10, 100, 1000, 1200, 1500. For *max_depth* 5, 10, 15, 20, 25, 30, 35, 40 used. For *max_features* "auto" and "sqrt" used. For *min_samples_split* 2, 4, 6, 8, 10, 12 used and for *min_samples_leaf* 1, 2, 4, 6, 8, 10, 12 used. For gradient boosting algorithm 6 hyperparameters are used those are *min_samples_split*, *min_samples_leaf*, *max_depth*, *learning_rate*, *n_estimators*, *max_features*. For *min_sample_split* 2, 4, 6, 8, 10, 12, 14 used, for *min_samples_leaf* 1, 2, 4, 6, 8, 10, 12, 14 used, for *max_depth* 5, 10, 15, 20, 25, 30, 35, 40 used, for *n_estimators* 10, 100, 1000, 1200, 1500 used, for *max_features* "auto", "sqrt" used, for *learning_rate* 0.05, 0.06, 0.07, 0.08, 0.09, 0.10, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18 used. *"RandomizedSearchCV()"* method was initiated for all the classifiers with selected hyperparameter range, with 10-fold cross validation. The output for the best hyperparameter for each model is given below

- **K-NN**: *'p'* : 1, *'n_neighbors'* : 15, *'leaf_size'* : 20
- **SVC**: *'C'* : 20
- **CART**: *'min_samples_split'* : 2, *'min_samples_leaf'* : 6, *'max_depth'* : 6
- **Random Forest**: *'n_estimators'* : 1000, *'min_samples_split'*: 6, *'min_samples_leaf'* : 2, *'max_features'* : 'sqrt', *'max_depth'* : 30
- **Gradient Boost**: *'n_estimators'* : 100, *'min_samples_split'*: 4, *'min_samples_leaf'* : 14, *'max_features'* : 'sqrt', *'max_depth'* : 15, *'learning_rate'* : 0.06

The obtained best hyperparameters are used for the model prediction.

CHAPTER 5

Numerical Results

To compare all the model performance, cross validation accuracy, precision and recall taken as evaluation measure. Accuracy, Precision and Recall can be derived from the confusion matrix. The confusion matrix consists of 4 components,

- True Positive (TP): when classifier classifies actual churner as churner.
- True Negative (TN): when classifier classifies actual non-churner as non-churner.
- False Positive (FP): when classifier classifies actual non-churner as churner.
- False Negative (FN): when classifier classifies actual churner as non-churners.

From these four components several evaluation metrics can be drawn.

Accuracy: It is the summation of true positive and true negative to the total number of predictions.

$$\boldsymbol{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{5.0.1}$$

Precision: It is the ratio of actual true positive to the total number of positive predictions. Precision gives us insight about how many correctly predicted churners are actually churners.

$$\boldsymbol{Precision} = \frac{TP}{(TP + FP)} \tag{5.0.2}$$

Recall: It is the ratio of True positive prediction to the total number of True positive and false negative. Recall gives an insight about how many numbers of actual churner we are able to predict correctly with the model.

$$\boldsymbol{Recall} = \frac{TP}{(TP + FN)} \tag{5.0.3}$$

For this comparative analysis of customer churn, above three evaluation metrics were used for evaluation. The result of this experiment is shown in Table 5.1. (Highest scores are in dark bold).

Model	**Hyperparameter Setting**	**CV Accuracy**	**Precision**	**Recall**
Logistic Regression	Default	84.06%	76.44%	38.59%
K-Nearest Neighbor	Tuned	83.37%	70.41%	28.88%
Support Vector Machine	Tuned	85.30%	**79.70%**	39.08%
Decision Tree (CART)	Tuned	85.51%	77.97%	42.96%
Random Forest	Tuned	**86.50%**	77.97%	44.66%
Gradient Boost	Tuned	86.08%	75.57%	**48.06%**

Table 5.1: Results

From Results table we can see that Random Forest classifiers has achieved highest cv accuracy of 86.50% and Precision score of 77.97% which is 2% less than highest precision score and recall score of 44.66% which is 4% less than highest score. Gradient boost classifier has achieved highest recall score of 48.06% and cv accuracy of 86.08% which is 0.5% less than random forest and a precision score of 75.57% which is 2.4% less than random forest and 4% less than highest score. The lowest performer on the basis of accuracy, precision and recall score is K-nearest neighbor followed by logistic regression, support vector machine and then decision tree CART algorithm.

CHAPTER 6

Conclusions

Customer churn prevention is and will be the top priority for most of the product and service-based organizations. In this paper we have performed an experiment on finding the best classification algorithm among the most commonly used state of the art classification models. The classifiers were evaluated based on the cv accuracy, precision and recall score on a banking dataset. The results which we have achieved after tuning the models are not distinctive. Almost all classifiers have achieved a good cv accuracy and precision score but only one evaluation metrics which differentiates the results is recall score. As customer churn prediction problem is to predict the actual churn customer, in this case the recall score becomes first priority to consider. By considering the recall score as top priority then precision and then cv score, gradient boost algorithm will be yielding the best performances among all models followed by random forest classifier. The gradient boost algorithm can be used in banking industry for customer churn prediction. From this experiment, we concluded that ensemble learning based classifiers should be the best choice for customer churn prediction problem.

In future, we can further extend this study to implement some advance boosting algorithms like XGBoost, CatBoost and deep learning based and transfer learning-based models to find the best algorithm for the customer churn problem. We can consider use of different evaluation metrics such as F1-score, ROC curve for model evaluation and can take different datasets from different domains.

Bibliography

[1] L. Xie, D. Li, and J. Xiao, "Feature selection based transfer ensemble model for customer churn prediction," in *2011 International Conference on System science, Engineering design and Manufacturing informatization*, vol. 2, 2011, pp. 134–137.

[2] M. R. Ismail, M. K. Awang, M. N. A. Rahman, and M. Makhtar, "A multi-layer perceptron approach for customer churn prediction," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 10, no. 7, pp. 213–222, 2015.

[3] F. Guo and H.-L. Qin, "The analysis of customer churns in e-commerce based on decision tree," in *2015 International Conference on Computer Science and Applications (CSA)*, 2015, pp. 199–203.

[4] M. Spiteri and G. Azzopardi, "Customer churn prediction for a motor insurance company," in *2018 Thirteenth International Conference on Digital Information Management (ICDIM)*, 2018, pp. 173–178.

[5] I. Ullah, B. Raza, A. K. Malik, M. Imran, S. U. Islam, and S. W. Kim, "A churn prediction model using random forest: Analysis of machine learning techniques for churn prediction and factor identification in telecom sector," *IEEE Access*, vol. 7, pp. 60 134–60 149, 2019.

[6] Y. Xie and X. Li, "Churn prediction with linear discriminant boosting algorithm," in *2008 International Conference on Machine Learning and Cybernetics*, vol. 1. IEEE, 2008, pp. 228–233.

[7] N. Lu, H. Lin, J. Lu, and G. Zhang, "A customer churn prediction model in telecom industry using boosting," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1659–1665, 2012.

[8] K. Kim and J.-H. Lee, “Bayesian optimization of customer churn predictive model,” in *2018 Joint 10th International Conference on Soft Computing and Intelligent Systems (SCIS) and 19th International Symposium on Advanced Intelligent Systems (ISIS)*, 2018, pp. 85–88.

[9] K. G. M. Karvana, S. Yazid, A. Syalim, and P. Mursanto, “Customer churn analysis and prediction using data mining models in banking industry,” in *2019 International Workshop on Big Data and Information Security (IWBIS)*, 2019, pp. 33–38.

[10] S. H. Dolatabadi and F. Keynia, “Designing of customer and employee churn prediction model based on data mining method and neural predictor,” in *2017 2nd International Conference on Computer and Communication Systems (ICCCS)*. IEEE, 2017, pp. 74–77.

[11] D. S. Sisodia, S. Vishwakarma, and A. Pujahari, “Evaluation of machine learning models for employee churn prediction,” in *2017 International Conference on Inventive Computing and Informatics (ICICI)*. IEEE, 2017, pp. 1016–1020.

[12] A. Alamsyah and N. Salma, “A comparative study of employee churn prediction model,” in *2018 4th International Conference on Science and Technology (ICST)*. IEEE, 2018, pp. 1–4.

[13] S. F. Sabbeh, “Machine-learning techniques for customer retention: A comparative study,” *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 2, 2018.

[14] P. Sun, X. Guo, Y. Zhang, and Z. Wu, “Analytical model of customer churn based on bayesian network,” in *2013 Ninth International Conference on Computational Intelligence and Security*. IEEE, 2013, pp. 269–271.

[15] P. K. Dalvi, S. K. Khandge, A. Deomore, A. Bankar, and V. Kanade, “Analysis of customer churn prediction in telecom industry using decision trees and logistic regression,” in *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*. IEEE, 2016, pp. 1–4.

[16] P. Asthana, “A comparison of machine learning techniques for customer churn prediction,” *International Journal of Pure and Applied Mathematics*, vol. 119, no. 10, pp. 1149–1169, 2018.

[17] X. Hu, Y. Yang, L. Chen, and S. Zhu, “Research on a customer churn combination prediction model based on decision tree and neural network,” in *2020 IEEE 5th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*, 2020, pp. 129–132.

[18] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, and M. Klein, *Logistic regression*. Springer, 2002.

[19] J. Gou, L. Du, Y. Zhang, T. Xiong *et al.*, "A new distance-weighted k-nearest neighbor classifier," *J. Inf. Comput. Sci*, vol. 9, no. 6, pp. 1429–1436, 2012.

[20] R. J. Lewis, "An introduction to classification and regression tree (cart) analysis," in *Annual meeting of the society for academic emergency medicine in San Francisco, California*, vol. 14. Citeseer, 2000.

[21] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

APPENDIX A

Experimental Source Code

The source code of this experiment will be given in a separate zip file along with the dataset file.