**Customer Churn Prediction**

**for Bank in the U.S. Market**

Naveen Ajay karasu

MS in Cybersecurity

UNT

Varun Simha Reddy

MS in Artificial Intelligence

UNT

Santhoshi Kareddy

MS in computer Science

UNT

Venkata Sai Karthik

MS in computer Science

UNT

Naga sive sankar

MS in computer Science

UN

**Abstract**

In this project we will address the problem of classifying bank customers into two categories. Those who will churn or those who will remain loyal. As we are dividing the data into two different output, we can call it as a binary classification problem, making machine learning algorithms like decision trees, Random Forest classifier, and neural networks as suitable candidates. Our primary goal in doing this project is to build efficient model which will have high accuracy, precision, and recall in predicting

churn, it allows the banks to identify at-risk customers proactively. Customer churns also known as the loss of customers to other competitors in this case it is to other banks, as it presents a significant challenge to the banks. Our project aim is to study different things, behavior from day to day activities of the user to understand more clearly that why the customer churn is happening and after that we will we try to feed the behavior of the customer as data to the to a machine learning model by employing various machine learning model techniques on a bank customer dataset. The analyzing done on different factors like demographics, account activity, product usage etc..,. We think this predictive capability can empower banks to implement targeted different strategies, to reducing churn and enhancing customer loyalty.

**Introduction**

Before knowing about our project let’s understand the problem we are trying to solve here which is customer churn, it can be defined as the losing clients to banks, which is a very important challenge to any bank as it can significantly affect the working of the banks as it will cause decline in the sustainability and growth of the bank. In today’s modern world there are more and more banks are present for a customer to choose from in the competitive market, keeping the customer retained as a existing customers becomes a big challenge for any bank. As having more customer very importantly essential for maintaining a steady revenue for the bank to opereate. So, Understanding the patterns and trying to predict the customer churn has a very important necessity for every bank.

So, In our project we want to harnesses the power of machine learning models to analyze and try to predict reason for the churn using a dataset containing data on customer demographics, financial details, account activity etc..,. We think this approach allows us to move beyond traditional techiniques and strategies to deploy proactive create measures that can mitigate or reduce the risk of churn before it really happens.

The predictive model will help the banks to separate or divide the it’s customer’s database into distinct different groups based on their level of risk of churning. This segmentation can then help inform targeted intervention strategies which can be tailored to specific particular customer needs and preferences which thereby trying to increase the customer satisfaction and loyalty to the bank. By integrating these insights given by model into the bank’s strategic planning, we think banks can not only reduce churn rates but also enhance the overall customer engagement and profitability in the bank.

**Data Description**

The important thing in our analysis is a non-other than the data itself [1]. So, for our project we are using the dataset provided by non-other than the U.S. banks itself. The data contains a wide range of customer characteristics carefully labeled with their churn status. This dataset plays an important role in developing our machine learning models to predict customer churn accurately.

**Data Composition**

The dataset consists of different varity of variable that makes a customer profile which can be used in creating a model for predicting the customer churn in the bank. The data is created based on the U.S. banks data. So, we can assume that the model we are going to develop is suitable for banks which are located and operated inside the U.S. itself. Knowing where the data came from and the targeted users of our model can help us greatly create a good model. For example, even if we develop a good model with high accuracy and prediction it can still fail if it is used on data from regions as we solely developed our model using data only from U.S. region.

**Data Quality and Preprocessing**

The data consists of 10000 rows of customer data in the dataset we are using and have 14 columns like customer ID, Surname, credit score, gender, age, tenure, balance, salary, does he credit card and many other. Having 10000 rows of data doesn’t mean we have data on 10000 unique customer, because same person can have multiple accounts in the same bank or in a different banks.

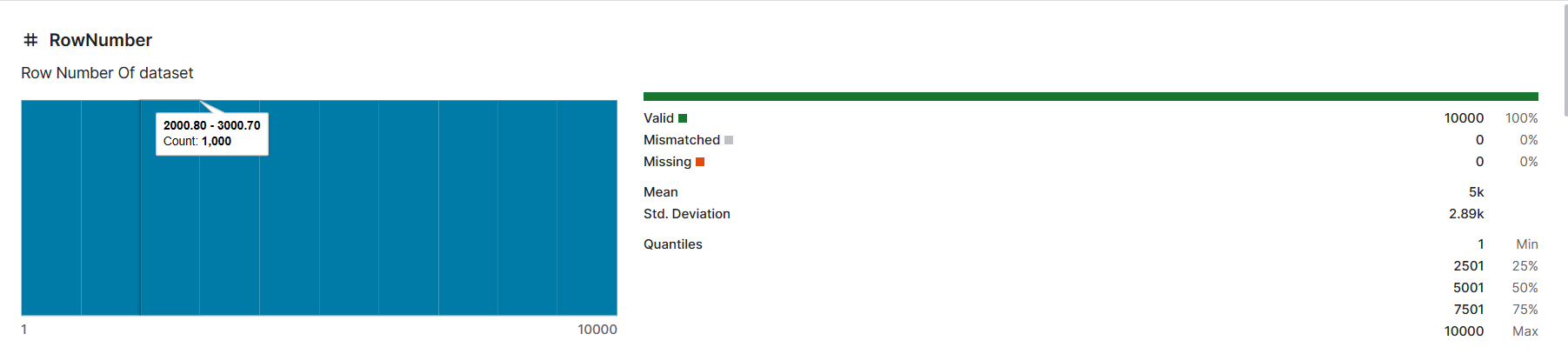


Figure 1: Showing no of rows in the dataset.

We cannot use the raw data directly as it may have anomalies or missing data for some customers, we need to preprocess the data before using it to build the model. As it will increase the accuracy and precdiction of our model. We can do the preprocessing of the data using data normalization and scaling process by using techniques like min-max scaling, z-score normalization which helps in ensuring that each column/ feature contributes equally to the prediction of the model. Which helps in preventing the model’s decision being solely depended on a single feature/column.

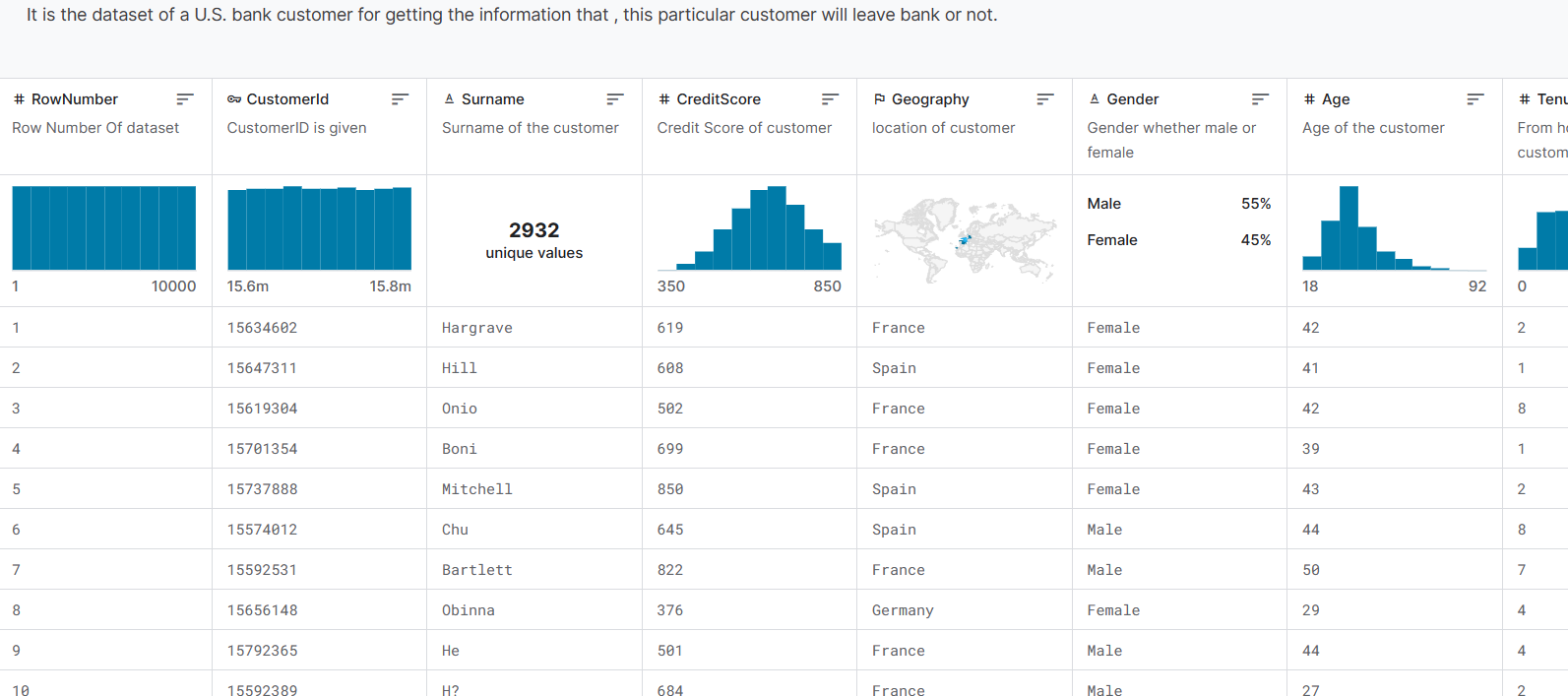


Figure 2: Showing different features in the dataset

We can further process the data by making the data simple and easier for the machine learning model to understand like converting the alphabetic word into numeric or binary representation as 1 or 0. For example, we can use this to represent gender in this way so for male as ‘1’ and female as ‘0’. We can do the same thing for marital status, has creditcard etc..,. So, it makes easier to feed the data to the model and reduces the need for computation.

**Methodology**

**Data Exploration**

The data exploration process requires an in-depth analysis of the dataset's features to differentiate their distributions and inter-relationships between each other, mainly with the customer churn feature. So, this phase is very important for understanding and getting to know the insights into the dataset which can help us form hypotheses regarding the potential predictive factors.

Below is the list of library we have used in our project.

Code:

import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

from sklearn.metrics import accuracy\_score, confusion\_matrix, roc\_curve, auc

import xgboost as xgb

from xgboost import XGBClassifier

from sklearn.model\_selection import RandomizedSearchCV

from sklearn.model\_selection import train\_test\_split

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.preprocessing import LabelEncoder

import plotly.express as px

warnings.simplefilter(action='ignore', category=FutureWarning)

import os

We have used google colab for doing our project. The reason for using it is because, it’s easier to share the progess with other team-mates in real time and each and every one of the team can contribute in writing the code at there own time without needing to wait for getting the updated code from the other team mates. We crearted a new notebook in the colab and shared the access along with other teammates to see the changes in real time. Below is the code for adding files and access it for developing the model.

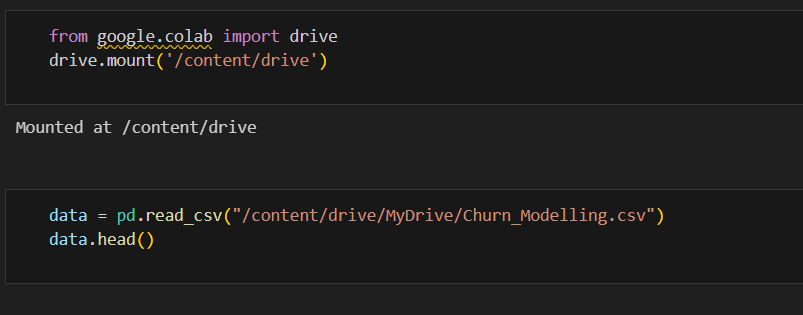


Figure 3: code for accessing the dataset

**Techniques**

We used three different machine learning models namely XGBoost, Random forest, Decision tree. We need to use the technique known as hyper parameter [7] optimization because it is a important step in building any machine learning model. The Hyper parameters are known as configuration settings used to model architecture, they are trained or learned from the data but they are used to set before the learning process start for the model. It is also important step just like data preprocessing. Optimizing the parameters is very important because they can directly control the behavior of the training algorithm model which can have a significant impact on the performance of the model being trained on.

A screenshot of a computer

Description automatically generated

Figure 4: Heat map of the features in the data set

The first algortim we used is XGBoost, which is an advanced implementation [6] of gradient boosting that is said to be highly effective in different machine learning algorithms as it is known for its performance and speed. The benefits of using XGBoost include handling sparse data to have regularisation which can help us avoid overfitting. Hyper parameters used for XGBoost are learning\_rate, max\_depth, min\_child\_weight, colsample\_bytree [5].

The second algorithm we used is Random Forest[4], which can defined as model that operates by constructing different types of decision trees during training and getting the class result that is defined has classification mean prediction using regression. The benefits of using it is its robustness against overfitting and the ability to handle huge amounts of datasets with more features. Hyper parameters used for Random forest are n\_estimators, max\_depth, min\_samples\_split, min\_samples\_leaf, max\_features

The third algorithm we used is Decision Tree [3] which can be defined as a tree structure where each internal node represents a feature and the branch represents the decision rule where leaf node represents the outcome from the branch. It can be easily visualized and understood even by people who have little to no knowledge. Hyper parameters used for decision tree are criterion, splitter, max\_depth, min\_samples\_split, min\_samples\_leaf and max\_features.

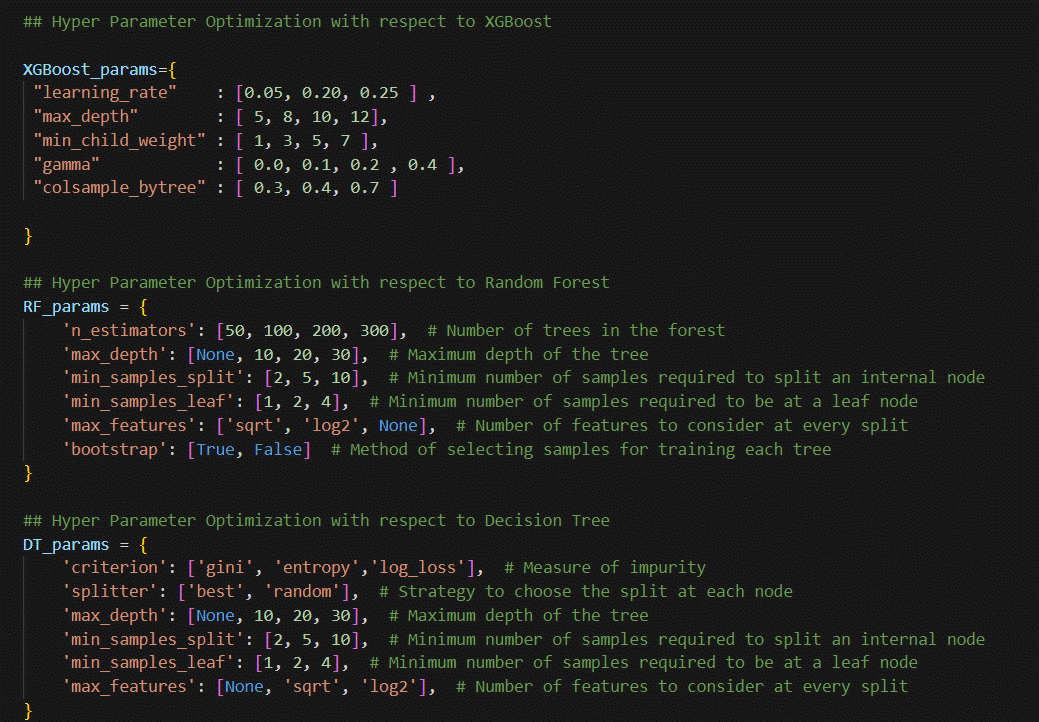


Figure 5: Showing the hyper parameters setting

We tried using the neural networks on the dataset to train the data. But the result we got from it has much less accuracy when compared the above three techniques. The reason for getting less accuracy might be due to several reasons but we think having less small data to train the network as the main reason for the slow accuracy. So, we removed as the accuracy is much less compared to others.

**Results**

Below is the kde plot known as violin plot, we used it to show the age distribution on the customers of different bank and Exited status. So, we represented the customers who stayed loyal to the bank and statyed are shown using red color and for those who left the bank are shown in blue color. From the graph we can identify that red area as more density when age is less which can help us understand that customers who are younger stay with the bank. As for the blue area which represents customers who left the bank, is more almost evenly spread over different ages which indicates that they come from a different ages. This information can help also help bank understand that age also has some degree of influence whether a customer stays with the bank or leave the bank.

A graph showing different colored shapes

Description automatically generated with medium confidence

Figure 6: Kde plot Age v/s Exited

We used KDE plot as a data visualization tool to estimate the probability of density function to the continuous variable like age. It's help to have a look at the distribution of a data set, smoothed by a kernel. Which can help us in understanding the shape of the distribution. As it's useful for seeing the underlying trend between the data points distribution.

A graph of a normal distribution

Description automatically generated with medium confidence

Figure 7: Kde plot of Density v/s Age

We have plotted ROC curves for three different algorithms used in our project: XGBoost, Random Forest, and Decision Tree. The ROC curve can be defined as a graphical representation that for the ability of a binary classifier system is varied. It plots the True Positive Rate against the False Positive Rate at different threshold values.

From the graph we can understand that each model's performance is quantified by the area which is under the ROC curve, so higher the area which indicates the higher the model performance. The XGBoost model has an accuracy of 86.20% next the Random Forest model has an accuracy of 86.35% and finally the Decision Tree has an accuracy of 81.70%.

By reading the where the curve is nearer to the top-left corner. We can know where each has more balance between sensitivity and specificity. From looking at the cureves we can say that the Random Forest and XGBoost curves have a higher discriminatory power to differeinaite between the positive class and the negative class compared to the Decision Tree model.

A graph of a curve

Description automatically generated with medium confidence

Figure 8:Roc curve of the three algortihms

**Conclusion and Future Work**

We think Random Forest is the best model that can be used has a best tool for predicting customer churn for improving customer retention strategies in banks. The Future work for this will be collecting more data and train the model more so that we can increase the accuracy of our model. Once we have enough data, we are planning to train again training the neural network with that data. So, we can use the power of deep learning fully to have better accuracy which can integrating real-time data feeds to continually refine the model's predictions on the go.

**References**

1. <https://www.kaggle.com/code/pjmathematician/pii-external-data-creation-loading>
2. <https://www.ibm.com/topics/decision-trees>
3. <https://builtin.com/machine-learning/decision-tree>
4. <https://careerfoundry.com/en/blog/data-analytics/what-is-random-forest/>
5. <https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/>
6. <https://www.geeksforgeeks.org/ml-xgboost-extreme-gradient-boosting/>
7. <https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac>