CUSTOMER CHURN PREDICTION

Team Members:
19BDS0006(Rashi Maheshwari)
19BDS0048 (Nishkarsh Gautam)
19BDS0059 (Pragati Singh)

Report submitted for the
Final Project Review of

Course Code: CSE3045
Predictive Analysis

Slot: A1 Slot

Professor: Dr. Ilanthenral Kandasamy

1. Introduction:

Undoubtedly, the financial industry is evolving at a rapid pace with growth in line with recognizable changes in consumer choices and expectations due to emerging technologies and the critical availability of various products and services. As a result, the banking industry is highly competitive due to all the threats and turmoil posed by all new and innovative market entrants such as Apple, Google, new start-ups, as well as direct competitors, i.e. It has become very competitive. Therefore, maintaining a competitive advantage, maintaining a point of differentiation (POD) and maintaining a customer's financial path is in terms of customer acquisition and more importantly the retention within the retail banking sector. It is considered one of the top priorities in strategic planning. After the business community has shifted its marketing focus from a product-centric strategy to a customer-centric strategy, despite the fact that different types of customer relationship management (CRM) strategies have existed for decades. Customers have become the focus of many researchers and practitioners. As a result, customer-company relationships have developed and many new marketing opportunities have been created. In addition, customer retention has become one of the key areas that most CRM strategies focus on. One of the foundations of CRM is customer churn forecasting. It is a prediction if the customer will leave the company. In addition, churn is defined as an APR where a customer unsubscribes from a service or terminates a business relationship. Reducing churn and retaining existing customers is the most cost-effective marketing approach to maximizing shareholder value. With so much competition, companies need to focus on retaining existing customers by effectively meeting the needs of their existing customers. Otherwise, you run the risk of losing your customers. And losing customers gives competitors the opportunity to attract them.

2. Literature Review Summary Table

Authors and Year (Reference)	Title (Study)	Concept / Theoretical model/ Framework	Methodology used/ Implementation	Dataset details/ Analysis	Relevant Finding	Limitations/ Future Research/ Gaps identified
Farid Shirazi, Mahbobeh Mohammadi (2018)	*A big data analytics model for customer churn prediction in the retiree segment.*	*The primary objective of this paper is to construct a predictive churn model by utilizing big data, including the structured archival data, integrated with unstructured data from sources such as online web pages, the number of website visits and phone conversation logs, for the first time in the financial industry.*	*a)Decision Tree with the growing method known as Classification and Regression Tree (CRT).* *b)deploying the classical General Linear Model (GLM) analysis using SAS system on a selected group based on CRT.*	*Analytic Window from November 1, 2011, to September 30, 2015.* *Age 50 is identified as the retirement age for most of the personal clients.* *The Analytic Universe is segmented, based on following three groups of clients: Existing Clients(92%), New to the bank (~3%) and also those clients who are new to the system(~5%).*	*Out of 2,813,276 clients, approximately 10.45% (294,036) fall under the retired group, while the remaining 89.55% of clients are part of the non-retirees sub-group. Second, by utilizing attrition cues, each group is further divided into two sub-groups of "Churners" and "Non-Churners." The result of this analysis for retired clients revealed that 8.7% of those clients who retired*	*the available behavioral data related to clients' online research did not satisfy the result. The results of this study will be different in the next few years when the younger generation reaches their retirement stage, as the Internet and social media usage, in particular, is inevitable among this group.*

					with the target bank have already attrited. *17% of retired clients have already churned to other financial institutes and kept a low-level relationship with the target bank; 16% of clients fall under committed risk who may or may not churn, depending on the future retention strategies.*	
Iris Figalist , Christoph Elsner, Jan Bosch , and Helena Holmstr¨om Olsson (2022)	*Customer Churn Prediction in B2B Contexts*	*Researchers have implemented a prediction model for customer churn within a B2B software product and derived a model based on the results.*	*a) mapping previous decision outcomes (churn or non-churn) to the respective customer ID b) cleaning, standardizing, and resampling the data; and c) applying appropriate feature selection techniques to identify the*	*NA*	*Single customers of B2B businesses are often of greater importance compared to B2C businesses since their number is typically much lower but their*	*the limitations and threat to validity of this study is the number of investigated cases. However, after working with multiple other B2B product providers prior to this study, and comparing the*

			relevant feature set.		transaction al value is much higher. Losing even one might have a significant impact on the provider of B2B products. While this reinforces the importance of customer churn prediction in B2B contexts, there is a lack of research on how to achieve this.	B2B-specific characteristics to the ones identified in existing literature.
Nurul Izzati Mohammad, Saiful Adli Ismail, Mohd Nazri Kama, Othman Mohd Yusop & Azri Azmi (2019)	*Customer Churn Prediction In Telecommunication Industry Using Machine Learning Classifiers*	*Identify the factors that influence customer churn and develop an effective churn prediction model as well as provide best analysis of data visualization results*	*The proposed methodology for analysis of churn prediction covers several phases: data pre-processing (data cleaning, data transformation and feature selection), analysis, implementing machine learning algorithms (Logistic Regression, Artificial Neural Network and*	*Dataset has been collected from Kaggle open data website. The dataset consists of 7043 records and each record is described by the following 21 attributes w. The attributes*	*Based on the experimental result, every classifier produces good results with high accuracy over 85%. The output shows that logistic regression outperforms compared to artificial neural network and*	*Throughout the analysis, fiber optic (attribute) provides fast internet would make customer stay, but it is listed on top of a positive impact on churn. Hence, there is a need to explore more for better understanding*

			Random Forest), evaluation of the classifiers by using performance measurement (accuracy, precision, recall and error rate) and choose the best one for prediction	include customer demographic information, billing information, product services and customer relationship variables. The target attribute is the churn where the customer is going to churn or not.	random forest. The classifier obtained by logistic regression shows the best results, but the disadvantage is the computational time.	and get some context of data.
Nadeem Ahmad Naz, Umar Shoaib & M. Shahzad Sarfraz (2018)	A Review on Customer Churn Prediction Data Mining Modeling Techniques [6]	Find one of the best data mining techniques in telecommunication especially in customer churn prediction.	Appropriate modeling techniques such as LR, NNM, DT, FL, CMC, SVM and DME are discussed for the churning purpose.	A large number of attributes such as segmentation, account info, billing info, call dialup types, line-info, and payment info, and complaint info, service provider info, and services info, etc. are used to predict customer churn.	DT and SVM with a low ratio used to find true churn rate and false churn rate. The LR might be used if looking for the churn probability. DMEL modeling technique is impractical and ineffective on a large dataset with high dimension. The high	CRT, NNM, LR, DT, SVM and fuzzy logic are most frequently used techniques for Churn prediction. The paper concludes that which one is the best technique under what condition and also a literature review of these techniques

					dimensional data for NB modeling technique is necessarily transformed into the low dimension.	
Adnan Amina, , Feras Al-Obeidatb , Babar Shahb Awais Adnana , Jonathan Looc , Sajid Anwar 2018 (https://www.sciencedirect.com/science/article/abs/pii/S0148296318301231)	*Customer churn prediction in telecommunication industry using data certainty*	*the proposed CCP approach will not only predict the customer churns but can also calculate the level of the certainty of the prediction by evaluating the classifier's decision into the following categories, (i) customer churn and non-churn with high certainty, (ii) customer churn and non-churn with low certainty. The low certainty can be considered as uncertain classification for predicting the customer churns. The*	*An empirical study is designed to evaluate the proposed CCP model where they have focused on distance factors using different distance zones (i.e., Upper and Lower zones) in the given TCI datasets. a benchmarking framework is setup to present and evaluate the performance of the proposed study. These experiments were carried out using MATLAB toolkit to fulfill the objectives of the proposed study by addressing a set of research questions.*	*For this study, they have selected arbitrary four publicly available datasets. The dataset-1 consists of 3333 samples and each sample represent individual customer; whereas, the ratio of churn and non-churn customers are 85.5% and 14.49%, respectively. Similarly, datasets-2, 3 and 4 contain 7043, 18,000 and 100,000 samples, respectively.*	*the upper distance zone has not shown more effect on the performance of CCP model in TCI datasets because it has obtained the performance in term of differences in the accuracy is 0.30%, 0.80%, 0.81% and 1.00% in datasets 1, 2, 3, and 4, respectively. On the other hand, lower distance zone achieved dramatic changes in the performance when the*	*Future studies might be able to provide empirical results on the balanced dataset with multiple base-classifiers. Observe the effect on the CCP model if we apply the feature selection method by assigning weights to the features. Test more comprehensive study with other types of models would offer the possibility to compare their results and eventually help to evaluate this effect statistically. Since the proposed model predicts*

		distance factors in term of upper and lower zones has not been considered for CCP in TCI yet. The proposed approach towards the target industry, exploring the discussed unexplored factors, can play a pivotal role in CCP models.			*zone size increases, it obtained differences in the accuracy such as 5.91%, 5.60%, 4.20% and 7.00% in datasets 1, 2, 3 and 4, respectively. Therefore, it is concluded that the upper zone is highly certain for classification because the classifier provided no big change in the results while the lower zone is highly uncertain for the classification due to drastic change in the classifier's results.*	*level of certainty that leads to expected level of accuracy. This can be used to select good cases for training the classifier efficiently and more accurately. This can also be used to predict outliers in training data that can have negative effect on the classification. This technique can also be used on priority sampling. With minor modifications in this techniques it can be applied in social media for critical node identification.*

3. Objective of the project: Customer churn is a significant issue and one of the most pressing challenges for large businesses. Companies are working to create methods to predict prospective customer churn because it has such a direct impact on their revenues, particularly in the telecom industry. As a result, identifying factors that contribute to customer churn is critical in order to take the required steps to reduce churn. Our work's key contribution is the development of a churn prediction model that helps telecom carriers estimate which customers are most likely to churn. We will use various algorithms from basic to advanced to predict customer churn like random forest classifier,XGBoost classifier and several hybrid models to achieve the highest accuracy.

4. Innovation component in the project:

In this project, we will be using various predictive analysis models to predict whether a customer will change telecommunications providers or not in the near future based on various parameters. The training dataset contains 4250 samples. Each sample contains 19 features and 1 boolean variable "churn" which indicates the class of the sample. The test dataset contains 750 samples. Each sample contains an index number and the 19 features. The proposed methodology for analysis of churn prediction covers several phases: data visualization and analysis, data pre-processing, implementing various models, evaluation of the classifiers and choosing the best one for prediction. We will be visualizing and analyzing univariate (both categorical and numerical) and bivariate variables. Data pre-processing phase will include – detecting and removing outliers, handling categorical variables, handling imbalanced dataset and scaling the dataset. We will be using and comparing three models – Support Vector Classification, Random Forest Classifier and XGBClassifier. The performance of the model will be measured by finding accuracy, classification report, confusion matrix and cohen kappa score. The model that gives the best performance will be chosen for the prediction of customer churn.

5. Work done and implementation

a. Methodology adapted:

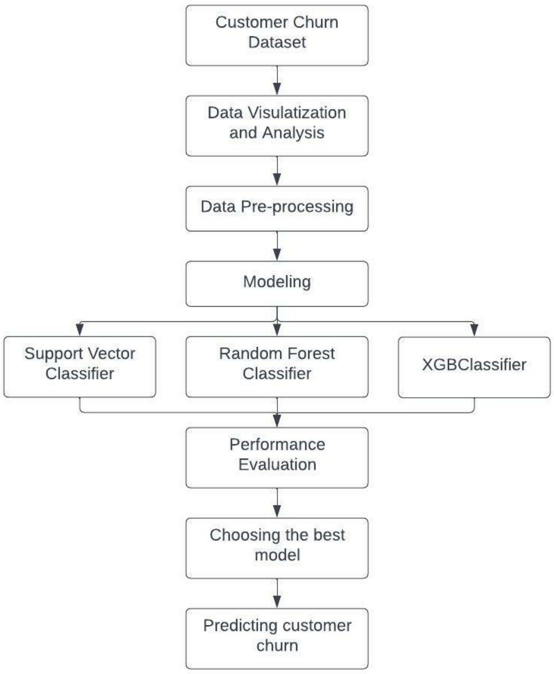


Fig: Our approach/architecture flow diagram

b. Dataset used:

a. The training set Contains 4250 lines with 20 columns. 3652 samples (85.93%) belong to class churn=no and 598 samples (14.07%) belong to class churn=yes.

Data fields

- state, *string*. 2-letter code of the US state of customer residence
- account_length, *numerical*. Number of months the customer has been with the current telco provider
- area_code, *string*="area_code_AAA" where AAA = 3 digit area code.
- international_plan, *(yes/no)*. The customer has international plan.
- voice_mail_plan, *(yes/no)*. The customer has voice mail plan.
- number_vmail_messages, *numerical*. Number of voice-mail messages.
- total_day_minutes, *numerical*. Total minutes of day calls.
- total_day_calls, *numerical*. Total number of day calls.
- total_day_charge, *numerical*. Total charge of day calls.
- total_eve_minutes, *numerical*. Total minutes of evening calls.
- total_eve_calls, *numerical*. Total number of evening calls.
- total_eve_charge, *numerical*. Total charge of evening calls.
- total_night_minutes, *numerical*. Total minutes of night calls.
- total_night_calls, *numerical*. Total number of night calls.
- total_night_charge, *numerical*. Total charge of night calls.
- total_intl_minutes, *numerical*. Total minutes of international calls.
- total_intl_calls, *numerical*. Total number of international calls.
- total_intl_charge, *numerical*. Total charge of international calls
- number_customer_service_calls, *numerical*. Number of calls to customer service
- churn, *(yes/no)*. Customer churn - target variable.

b. The project is not based on any previous projects.

c. Tools used: The analysis is done in one single machine that enables the code to run on Google Colab notebook and perform the simulation of the churn prediction models and visualize the analysis of customer behavior. Colab notebooks are Jupyter notebooks that are hosted by Colab. Various libraries and modules in python such as numpy, pandas, matplotlib, seaborn, scikit-learn, imblearn, xgboost, etc. are used for the visualization and prediction of customer churn.

d. Screenshot and Demo along with Visualization: (Preprocessing)

- Loading the necessary libraries and loading the dataset.

```
# Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import category_encoders as ce
from sklearn.preprocessing import OneHotEncoder
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from sklearn.metrics import cohen_kappa_score
from xgboost import XGBClassifier
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.
  import pandas.util.testing as tm
```

```
[13] # load the dataset
train = pd.read_csv('/content/train.csv')
test = pd.read_csv('/content/test.csv')
print('Train shape {}'.format(train.shape))
print('Test shape {}'.format(test.shape))
```

```
Train shape (4250, 20)
Test shape (750, 20)
```

```
# Checking the missing values
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4250 entries, 0 to 4249
Data columns (total 20 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   state                          4250 non-null   object
 1   account_length                 4250 non-null   int64
 2   area_code                      4250 non-null   object
 3   international_plan             4250 non-null   object
 4   voice_mail_plan                4250 non-null   object
 5   number_vmail_messages          4250 non-null   int64
 6   total_day_minutes              4250 non-null   float64
 7   total_day_calls                4250 non-null   int64
 8   total_day_charge               4250 non-null   float64
 9   total_eve_minutes              4250 non-null   float64
 10  total_eve_calls                4250 non-null   int64
 11  total_eve_charge               4250 non-null   float64
 12  total_night_minutes            4250 non-null   float64
 13  total_night_calls              4250 non-null   int64
 14  total_night_charge             4250 non-null   float64
 15  total_intl_minutes             4250 non-null   float64
 16  total_intl_calls               4250 non-null   int64
 17  total_intl_charge              4250 non-null   float64
 18  number_customer_service_calls  4250 non-null   int64
 19  churn                          4250 non-null   object
dtypes: float64(8), int64(7), object(5)
memory usage: 664.2+ KB
```

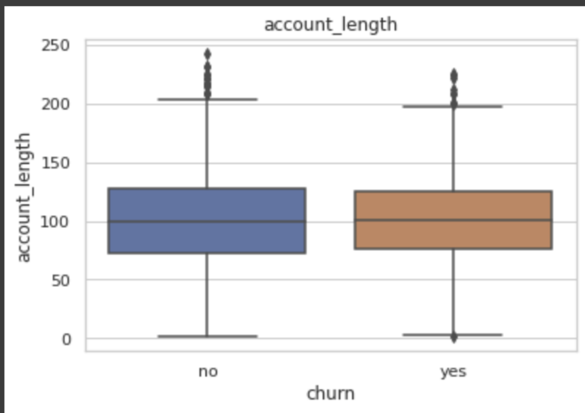
- Statistical analysis

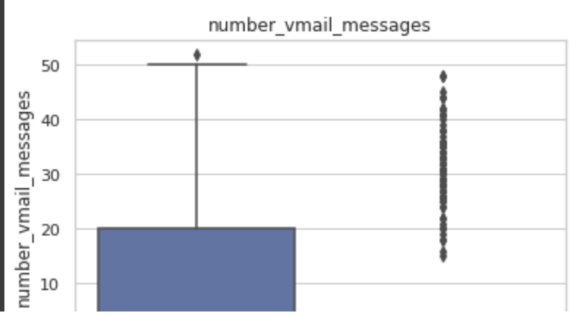
```
# describe the five points of statistics of numericals data
train.describe()
```

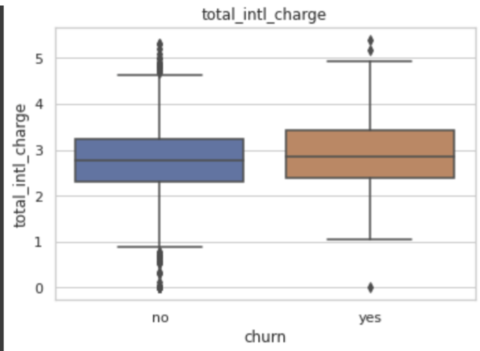
	account_length	number_vmail_messages	total_day_minutes	total_day_calls	total_day_charge	total_eve_minutes	total_eve_calls	total_eve_charge	total_
count	4250.000000	4250.000000	4250.000000	4250.000000	4250.000000	4250.000000	4250.000000	4250.000000	
mean	100.236235	7.631765	180.259600	99.907294	30.644682	200.173906	100.176471	17.015012	
std	39.698401	13.439882	54.012373	19.850817	9.182096	50.249518	19.908591	4.271212	
min	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	73.000000	0.000000	143.325000	87.000000	24.365000	165.925000	87.000000	14.102500	
50%	100.000000	0.000000	180.450000	100.000000	30.680000	200.700000	100.000000	17.060000	
75%	127.000000	16.000000	216.200000	113.000000	36.750000	233.775000	114.000000	19.867500	
max	243.000000	52.000000	351.500000	165.000000	59.760000	359.300000	170.000000	30.540000	

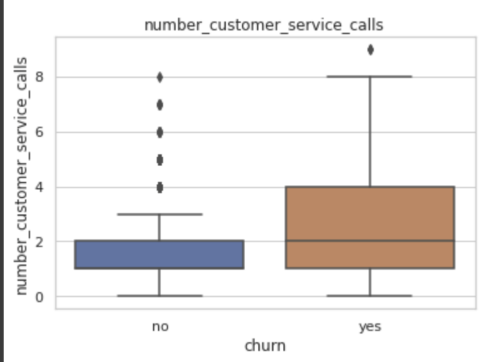
Outlier Detection

```python
for feature in num_var:
    if feature != 'churn':
        sns.boxplot(x ='churn', y = feature, data = train)
        plt.title(feature)
        plt.show()
```









- every features has a outliers so we need to remove the outliers.
- outlies contains the some usefull information.
- so we have to replace the outliers with some meaning full values. so we should replace the outliers with median values

Removing the outliers

```
#functions for removing outliers
def remove_outliers(train,labels):
    for label in labels:
        q1 = train[label].quantile(0.25)
        q3 = train[label].quantile(0.75)
        iqr = q3 - q1
        upper_bound = q3 + 1.5 * iqr
        lower_bound = q1 - 1.5 * iqr
        train[label] = train[label].mask(train[label]< lower_bound, train[label].median(),axis=0)
        train[label] = train[label].mask(train[label]> upper_bound, train[label].median(),axis=0)

    return train
```

```
[41] train = remove_outliers(train, num_var)
```

- Handling categorical data

Handling the Categorical Variable

- state feature has 51 different category so we can't converted into onehot encoder that is it create 51 different features so it leads to overfitting so I will use the hashing encoding for state featuer.

```
[43] hash_state = ce.HashingEncoder(cols = 'state')
     train = hash_state.fit_transform(train)
     test = hash_state.transform(test)
     train.head()
```

	col_0	col_1	col_2	col_3	col_4	col_5	col_6	col_7	account_length	area_code	international_plan	voice_mail_plan	number_vmail_messages	total_day_mi
0	0	0	0	0	1	0	0	0	107.0	area_code_415	no	yes	26.0	
1	0	1	0	0	0	0	0	0	137.0	area_code_415	no	no	0.0	
2	0	0	0	0	1	0	0	0	84.0	area_code_408	yes	no	0.0	
3	0	0	0	0	1	0	0	0	75.0	area_code_415	yes	no	0.0	
4	0	0	0	0	1	0	0	0	121.0	area_code_510	no	yes	24.0	

```
test.head()
```

	col_0	col_1	col_2	col_3	col_4	col_5	col_6	col_7	id	account_length	area_code	international_plan	voice_mail_plan	number_vmail_messages	total_day
0	0	1	0	0	0	0	0	0	1	128	area_code_415	no	yes	25	
1	0	1	0	0	0	0	0	0	2	118	area_code_510	yes	no	0	
2	0	0	0	0	0	0	0	1	3	62	area_code_415	no	no	0	
3	0	0	0	0	1	0	0	0	4	93	area_code_510	no	no	0	
4	1	0	0	0	0	0	0	0	5	174	area_code_415	no	no	0	

```
[45] # replace no to 0 and yes to 1
     train.international_plan.replace(['no','yes'],[0,1],inplace = True)
     train.voice_mail_plan.replace(['no','yes'],[0,1],inplace=True)
     train.churn.replace(['no','yes'],[0,1],inplace = True)
     test.international_plan.replace(['no','yes'],[0,1],inplace = True)
     test.voice_mail_plan.replace(['no','yes'],[0,1],inplace = True)
     train.head()
```

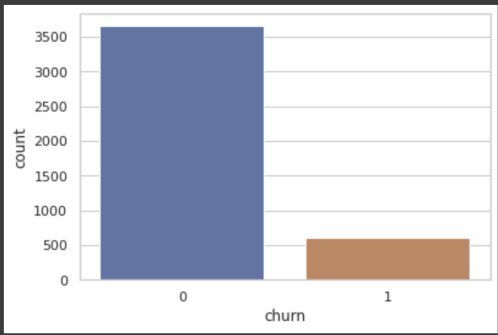
	col_0	col_1	col_2	col_3	col_4	col_5	col_6	col_7	account_length	area_code	international_plan	voice_mail_plan	number_vmail_messages	total_day_mi
0	0	0	0	0	1	0	0	0	107.0	area_code_415	0	1	26.0	
1	0	1	0	0	0	0	0	0	137.0	area_code_415	0	0	0.0	
2	0	0	0	0	1	0	0	0	84.0	area_code_408	1	0	0.0	
3	0	0	0	0	1	0	0	0	75.0	area_code_415	1	0	0.0	
4	0	0	0	0	1	0	0	0	121.0	area_code_510	0	1	24.0	

```
# converting the area_code to numerical variable using one-hot encoder
onehot_area = OneHotEncoder()
onehot_area.fit(train[['area_code']])

# Train
encoded_values = onehot_area.transform(train[['area_code']])
train[onehot_area.categories_[0]] = encoded_values.toarray()
train = train.drop('area_code', axis=1)

# Test
encoded_values = onehot_area.transform(test[['area_code']])
test[onehot_area.categories_[0]] = encoded_values.toarray()
test = test.drop('area_code', axis=1)
```

- Handling imbalanced dataset

Handling the Imbalanced dataset

```
# showing the imbalanced class
sns.countplot(x = 'churn', data = train)
plt.show()
```



- 0 represent the no churn and 1 represent the churn so there are huge difference in the class. so we need to balanced the dataset
- We have to use upsampling for handling the dataset

```
[50] x = train.drop('churn',axis=1).values
     y = train.churn.values
     id_submission = test.id
     test = test.drop('id', axis=1)
     # spliting the data into test and train
     x_train, x_test , y_train, y_test = train_test_split(x, y , test_size=0.3, random_state=0)
```

```
[51] print('Before upsampling count of label 0 {}'.format(sum(y_train==0)))
```

```
print('Before upsampling count of label 0 {}'.format(sum(y_train==0)))
print('Before upsampling count of label 1 {}'.format(sum(y_train==1)))
# Minority Over Sampling Technique
sm = SMOTE(sampling_strategy = 1, random_state=1)
x_train_s, y_train_s = sm.fit_resample(x_train, y_train.ravel())

print('After upsampling count of label 0 {}'.format(sum(y_train_s==0)))
print('After upsampling count of label 1 {}'.format(sum(y_train_s==1)))
```

```
Before upsampling count of label 0 2550
Before upsampling count of label 1 425
After upsampling count of label 0 2550
After upsampling count of label 1 2550
```

- Scaling the dataset

Scaling the dataset

after apply the upsampling technique the number of samples of both classes are same

```
# creating the object of minmax scaler
scaler = MinMaxScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)
test = scaler.transform(test)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:444: UserWarning: X has feature names, but MinMaxScaler was fitted without feature names
  f"X has feature names, but {self.__class__.__name__} was fitted without"
```

e. Models used:

(a) Support Vector Machine

- Support Vector Machine(SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges. However, it is mostly used in classification problems.
- In the SVM algorithm, each data item is plotted as a point in n-dimensional space (where n is a number of features) with the value of each feature being the value of a particular coordinate. Then, classification is performed by finding the hyper-plane that differentiates the two classes very well.
- It uses a subset of training points in the decision function called support vectors which makes it memory efficient.
- Different kernel functions can be specified for the decision function.

(b) Random forest classifier

- Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.
- The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.
- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

(c) XGBoost classifier

- XGBoost stands for Extreme Gradient Boosting. It is an implementation of Gradient Boosted decision trees.
- In this algorithm, decision trees are created in sequential form. Weights play an important role in XGBoost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results. The weight of variables predicted wrong by the tree is increased and these variables are then fed to the second decision tree. These individual classifiers/predictors then ensemble to give a strong and more precise model. It can work

on regression, classification, ranking, and user-defined prediction problems.

- It is a scalable and highly accurate implementation of gradient boosting that pushes the limits of computing power for boosted tree algorithms, being built largely for energizing machine learning model performance and computational speed.

- **Handling imbalanced dataset:**
  (1) The dataset imbalance is handled using Synthetic Minority Oversampling Technique (SMOTE).
  (2) SMOTE works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line.
  (3) The approach is effective because new synthetic examples from the minority class are created that are plausible, that is, are relatively close in feature space to existing examples from the minority class.

f. Screenshot and Demo along with Visualization (For results):

(a) SVM

```
svc = SVC(kernel='rbf', decision_function_shape='ovr')

svc.fit(x_train, y_train)

y_pred = svc.predict(x_test)

print('Accuracy: ')

print('{}'.format(accuracy_score(y_test, y_pred)))

print('Classification report: ')

print('{}'.format(classification_report(y_test, y_pred)))

print('Confusion Matrix')

print('{}'.format(confusion_matrix(y_test, y_pred)))

print('Cohen kappa score: ')

print('{}'.format(cohen_kappa_score(y_test, y_pred)))
```

```
Accuracy:
0.8690196078431373
Classification report:
              precision    recall  f1-score   support

           0       0.87      1.00      0.93      1102
           1       0.69      0.06      0.12       173

    accuracy                           0.87      1275
   macro avg       0.78      0.53      0.52      1275
weighted avg       0.85      0.87      0.82      1275

Confusion Matrix
[[1097    5]
 [ 162   11]]
Cohen kappa score:
0.09562561852539309
```

(b) Random Forest

```
rfc = RandomForestClassifier()

rfc.fit(x_train, y_train)

y_pred = rfc.predict(x_test)

print('Accuracy: ')

print('{}'.format(accuracy_score(y_test, y_pred)))

print('Classification report: ')

print('{}'.format(classification_report(y_test, y_pred)))

print('Confusion Matrix')

print('{}'.format(confusion_matrix(y_test, y_pred)))

print('Cohen kappa score: ')

print('{}'.format(cohen_kappa_score(y_test, y_pred)))
```

```
Accuracy:
0.9333333333333333
Classification report:
              precision    recall  f1-score   support

           0       0.93      1.00      0.96      1102
           1       0.98      0.52      0.68       173

    accuracy                           0.93      1275
   macro avg       0.95      0.76      0.82      1275
weighted avg       0.94      0.93      0.92      1275

Confusion Matrix
[[1100    2]
 [  83   90]]
Cohen kappa score:
0.6458830948592191
```

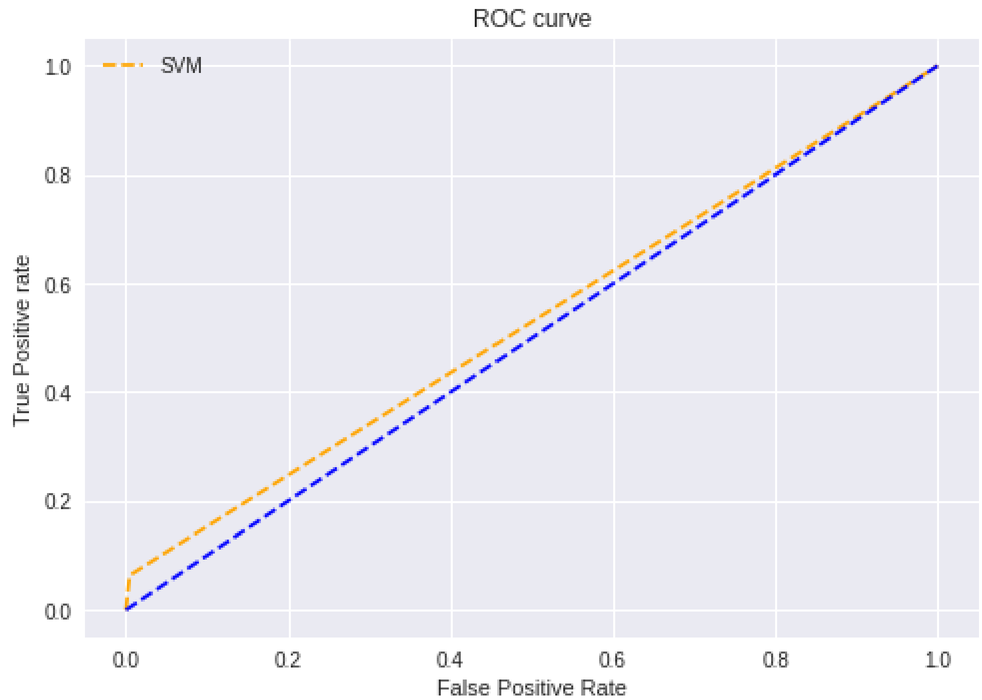
(c) XG boost

```
clf = XGBClassifier(max_depth=7, n_estimators=200, colsample_bytree=0.7,
                    subsample=0.8, nthread=10, learning_rate=0.01)

clf.fit(x_train, y_train)

y_pred = clf.predict(x_test)

print('Accuracy: ')

print('{}'.format(accuracy_score(y_test, y_pred)))

print('Classification report: ')

print('{}'.format(classification_report(y_test, y_pred)))

print('Confusion Matrix')

print('{}'.format(confusion_matrix(y_test, y_pred)))

print('Cohen kappa score: ')

print('{}'.format(cohen_kappa_score(y_test, y_pred)))
```

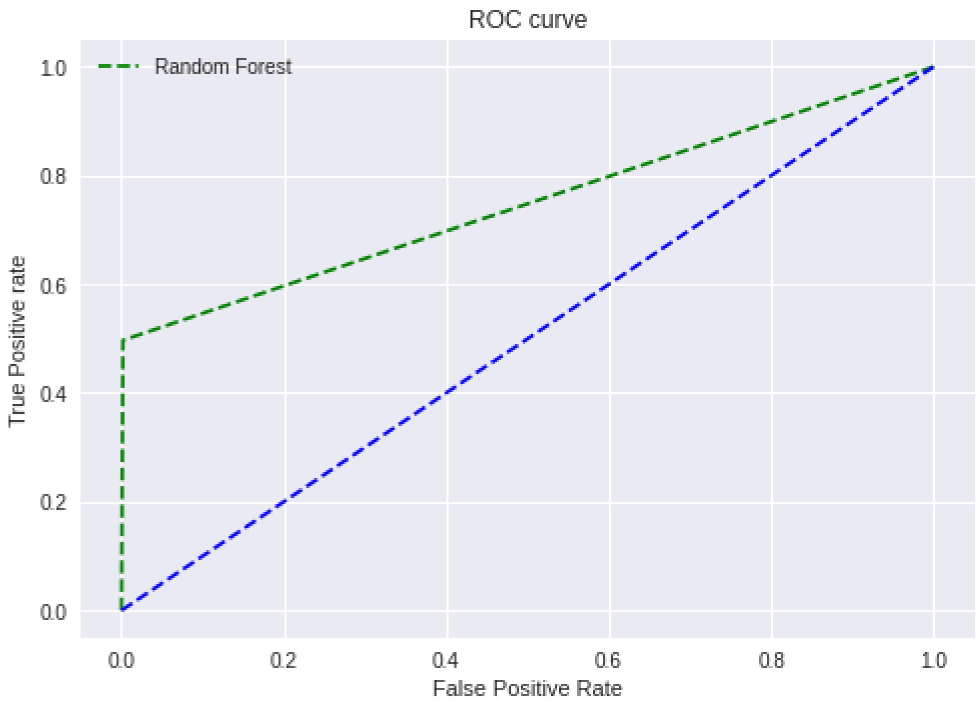
```
Accuracy:
0.9325490196078431
Classification report:
              precision    recall  f1-score   support

           0       0.93      1.00      0.96      1102
           1       0.98      0.51      0.67       173

    accuracy                           0.93      1275
   macro avg       0.95      0.76      0.82      1275
weighted avg       0.94      0.93      0.92      1275

Confusion Matrix
[[1100    2]
 [  84   89]]
Cohen kappa score:
0.6406261266280799
```

6. Results:

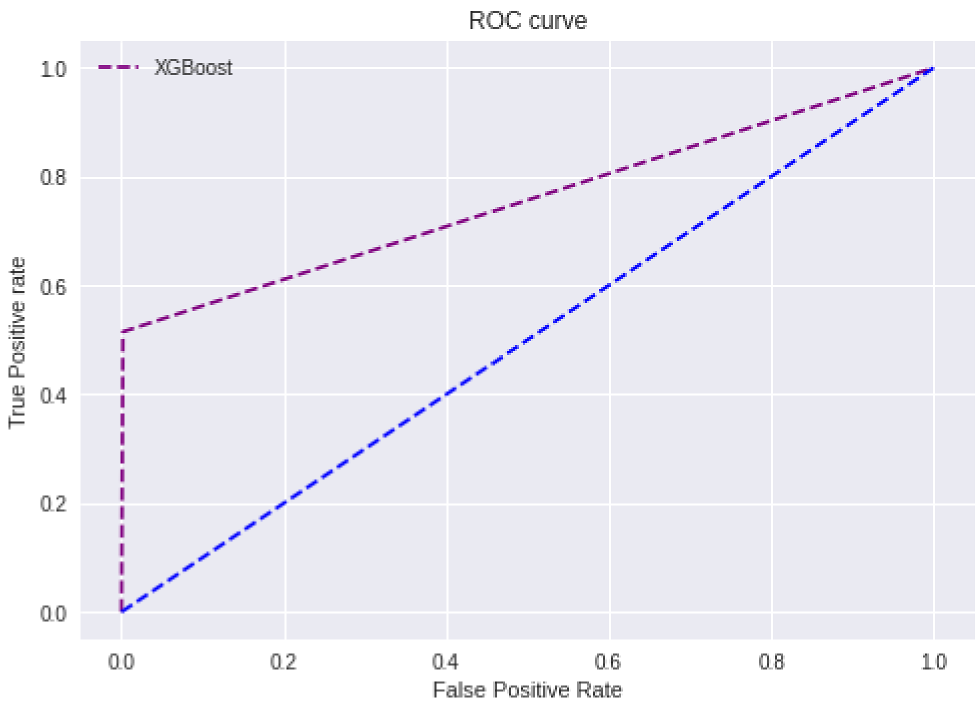
Model Name	Accuracy	AUC Score
1. SupportVector Machine	0.8690	0.5295
2. Random forest classifier	0.9333	0.7476
3. XGBoost classifier	0.9325	0.7563



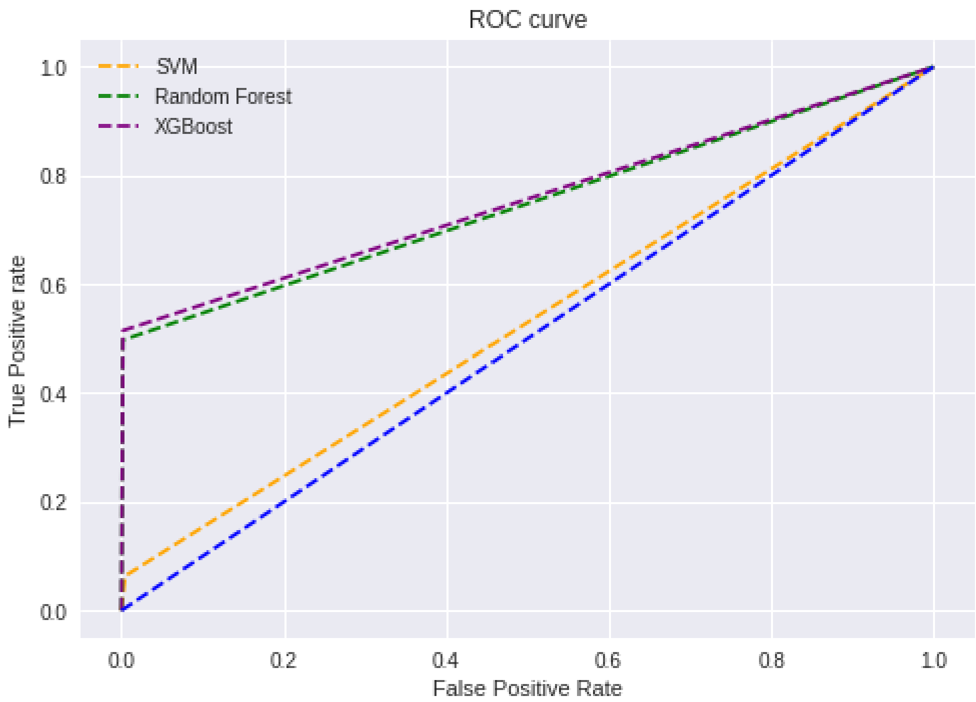
ROC curve for Support Vector Machine



ROC curve for Random Forest classifier



ROC curve for XGBoost classifier



ROC curve of all the models

- Random forest classifier has the best performance,compared to the other two models. It has an accuracy of 0.9333 **and an auc score of** 0.7476**.**

7. References

[1] Adnan Amina, Feras Al-Obeidatb, Babar Shahb, Awais Adnana, Jonathan Looc, Sajid Anwara, "Customer churn prediction in telecommunication industry using data certainty," Journal of Business Research, vol. 94, pp. 290-301, Jan 2019. https://doi.org/10.1016/j.jbusres.2018.03.003

[2] Abdelrahim Kasem Ahmad, Assef Jafar, Kadan Aljoumaa, "Customer churn prediction in telecom using machine learning in big data platform", Journal of Big Data, vol. 6, 2019, https://doi.org/10.1186/s40537-019-0191-6

[3] Iris Figalist, Christoph Elsner, Jan Bosch, Helena Holmström Olsson, "Customer Churn Prediction in B2B Contexts", International Conference on Software Business, ICSOB 2019: Software Business, pp. 378-386, https://doi.org/10.1007/978-3-030-33742-1_30

[4] Farid Shirazi, Mahbobeh Mohammadi, "A big data analytics model for customer churn prediction in the retiree segment", International Journal of Information Management, vol. 48, pp. 238-253, Oct 2019, https://doi.org/10.1016/j.ijinfomgt.2018.10.005

[5] Nurul Izzati Mohammad, Saiful Adli Ismail, Mohd Nazri Kama, Othman Mohd Yusop, Azri Azmi, "Customer Churn Prediction In Telecommunication Industry Using Machine Learning Classifiers", ICVISP 2019: Proceedings of the 3rd International Conference on Vision, Image and Signal Processing, article 34, pp. 1-7, Aug 2019, https://doi.org/10.1145/3387168.3387219

[6] Nadeem Ahmad Naz, Umar Shoaib, M. Shahzad Sarfraz, "A Review on Customer Churn Prediction Data Mining Modeling Techniques", Indian Journal of Science and Technology, vol. 11, 27, pp. 1-7, July 2018, https://doi.org/10.17485/ijst/2018/v11i27/121478