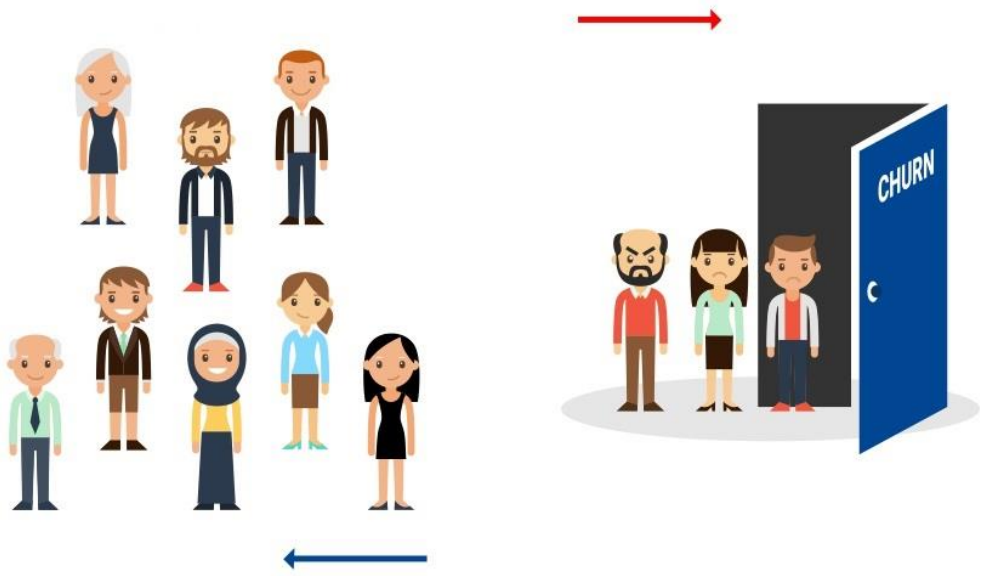
CHURN PREDICTION PROJECT ASSESSMENT 2



Understanding Problem Statement

Decreasing the Customer Churn is a key goal for any business. Predicting Customer Churn (also known as Customer Attrition) represents an additional potential revenue source for any business. Customer Churn impacts the cost to the business. Higher Customer Churn leads to loss in revenue and the additional marketing costs involved with replacing those customers with new ones.

In this challenge, as a data scientist of a bank, you are asked to analyze the past data and predict whether the customer will churn or not in the next 6 months. This would help the bank to have the right engagement with customers at the right time.

Objective

Our objective is to build a machine learning model to predict whether the customer will churn or not in the next six months.

Data Dictionary

You are provided with 3 files - train.csv, test.csv and sample_submission.csv

Training set

train.csv contains the customer demographics and past activity with the bank. And also, the target label representing whether the customer will churn or not.

Variable	Description
ID	Unique Identifier of a row
Age	Age of the customer
Gender	Gender of the customer (Male and Female)
Income	Yearly income of the customer
Balance	Average quarterly balance of the customer
Vintage	No. of years the customer is associated with bank
Transaction_Status	Whether the customer has done any transaction in the past 3 months or not
Product_Holdings	No. of product holdings with the bank
Credit_Card	Whether the customer has a credit card or not
Credit_Category	Category of a customer based on the credit score
Is_Churn	Whether the customer will churn in next 6 months or not

Test set

test.csv contains the customer demographics and past activity with the bank. And you need to predict whether the customer will churn or not.

Variable	Description
ID	Unique Identifier of a row
Age	Age of the customer
Gender	Gender of the customer (Male and Female)
Income	Yearly income of the customer
Balance	Average quarterly balance of the customer
Vintage	No. of years the customer is associated with bank
Transaction_Status	Whether the customer has done any transaction in the past 3 months or not
Product_Holdings	No. of product holdings with the bank
Credit_Card	Whether the customer has a credit card or not
Credit_Category	Category of a customer based on the credit score

Submission File Format

sample_submission.csv contains only 2 variables - row_id and engagement_score

Variable	Description
ID	Unique identifier of the row
Is_Churn	Whether the customer will churn in next 6 months or not

Evaluation metric

The evaluation metric for this hackathon is macro f1 score.

Approach used to solve the problem:

First, I imported all the required libraries/packages along with the data file provided in the assessment 2 section. Understood the problem statement and the feature target details present in our data files. Had to perform various Exploratory Data Analysis to check through the shape of the training data file, bifurcate the numeric and object datatype columns, check the description and statistical details provided by the numeric data columns.

Checked for missing value information and confirmed non null details with a visual on it. Took help of pandas-profiling to analyze the initial raw training dataset. Then with the help of unique values function was able to take a closer look in each and every low value categorical details.

Used visualizations to perform univariate, bivariate and multi variate analysis noting down all the observations that would be needed to perform any data preprocessing or feature engineering steps before we began building our classification machine learning models using various classification algorithms.

Data preprocessing/Feature Engineering undertaken:

Then we encoded the categorical data columns with the help of Label Encoder and Ordinal Encoder techniques. Using box plot we were able to check through various outlier details and then checked for skewness information to deal with columns that did not show a bel shape curve or normal distribution. I used Z score and IQR methods to try and check what worked best in dealing with the outlier issues. I observed that Z score method performed better than the IQR mechanism also we were losing less data percentage in it.

The feature columns Age and Balance were the only continuous data type columns and were almost normally distributed. When I tried using Log Transformation to details with the slight skewness it did not improve the model building algorithm and rather degraded the score so I chose not to deal with the skewness as it was not a challenge that needed to be dealt with. Finally, I checked for correlation details using a heatmap and bar plot comparing the feature columns with the target label.

I was then bale to spit the feature and target columns in 2 different variables namely X and Y. I observed that there was a huge class imbalance in our target label so made use of SMOTE function to handle the imbalance issue. I did try to use the SMOTEENN function option to deal with the class imbalance concern but that up sampling option did not yield any better result for me. Used the feature scaling technique with the help of Standard Scaler function to ensure that there was no biasness over the unit of numeric data present in the feature columns. After splitting the feature and target label we then used the entire data set to separate them in training feature-label and testing feature-label using the 75:25 ratio and checked for the best random state number to be used.

Final model details:

Using the Logistic Regression algorithm, we were able to obtain the best random state number for our classification model. And with the help of Random Forest Classifier algorithm, we check the feature importance details along with a visual on the most important feature column range contributing towards the machine learning model building.

I created a user defined function called “classify” and used it to train a given model, predict the test values, checked multiple evaluation metrics starting with classification report, accuracy score, f1 score, cross validation score and the difference between the accuracy score and validation scores. Then one by one I called our user defined function to enter our classification algorithms starting with Logistic Regression, Support Vector Classifier, Decision Tree Classifier, Random Forest Classifier, KNN Classifier, Extra Tree Classifier, XGB Classifier, LGBM Classifier. I noticed that the algorithms which did well were SVC, Random Forest, XGB and LGBM. So, I tried using hyper parameter tuning along with Grid Search CV mechanism to come up with a better parameter list and boost the model confidence score.

I even used the Neural Network model to check if I am able to get a better score however it did not show any promising results even after the tweaking the data, the parameters and the layers for it. Therefore, I finally compared through the best models and chose Support Vector Classifier as my final model that provided me with the highest score and decided to submit it for the final scoring solution. Using this I was then able to contemplate the testing dataset perform the same data preprocessing steps on it as the training dataset and then predicted the values removing the ID column each time and then clubbed it into our final CSV file during submission process using the dataframe methods.

Shortcoming of this approach:

This approach would have worked better if we had more data points that had the churn labels giving us more samples while making the model understand better from feature point of view in predicting the customers who would leave the bank in next 6 months’ time frame. The reason being that human behavior regress to it’s mean so when we try to predict human behavior, we actually are trying to predict the mean of the overall human behavior. Due to less data points, this approach tends to overfit for algorithms like Neural Network or Random Forest Classifier therefore Support Vector Classifier showed better results during this hackathon.

Learnings from this assessment:

The best approach must be decided taking the availability of overall data into consideration. Simple algorithms tend to suit better when we have less information and a smaller number of labelled data records, as they are more robust in nature. Whatever gets measured improves eventually so widening the range of the measurements was something that could have been taken into consideration with a higher time requirement.