# Credit Card customer Attrition rate

Niveditha Channapatna Raju

**Abstract**

With the advance of digital technology, people are increasingly resorting to credit cards online for their transactions. In this competitive market space, it is paramount for banks to retain customers to maintain the profit margin. We aim to predict credit card customer churn using machine learning models to deal with customer churn problems. We have applied four models including Logistic regression, Decision tree, Random Forest, and Light GBM to our dataset which contains more than 10000 pieces and 20 features.

## 1 Introduction

Given the fierce market competition, credit cards are a crucial part of a bank's profit. Customers are therefore crucial from a business perspective. As a result, customer turnover is a key area of attention for many banks. If we take a look at the AARRR or HEART frameworks utilized by several institutions, it is accurate. According to studies, a bank's profits might rise by 85% when the retention rate goes up 5%. The purpose of this study is to forecast customer churn. Once forecasted, banks would have enough time to take proactive steps to keep clients by providing better services or more alluring discounts. Therefore, it is very important, particularly in today's world where there is a wealth of customer-related data, and with the widespread usage of big data, large users' data have become priceless jewels for businesses. Large volumes of data may be processed and analyzed using machine learning. While using models to predict the outcome, some other articles primarily concentrate on unsupervised learning, which is generally unreliable and has relatively low interpretability. In this study, we use Kaggle, which has over 10,000 data points and 21 different attributes, to get credit card holders' information. To determine its distribution and display correlations between attributes, we perform exploratory data analysis. The dataset was then divided into training and testing, and standardization came next. To evaluate the performance of the models, Logistic Regression, Decision Tree, Random Forest, and Light GBM are employed.

## 2 Method

### 2.1 Logistic regression

The method of modeling the likelihood of a discrete result given an input variable is known as logistic regression. The most common algorithm models binary outcome by classifying a sample to the class if the estimated probability is greater than 50%. The probability estimated by the model in vector form is given by:

$$p = h\theta x = \sigma x T \theta$$

Interestingly, because it uses a non-linear log transformation of the linear regression, logistic regression can handle non-linear correlations between the dependent and independent variables. A Logistic function or a logistic curve is a common s-shaped curve (sigmoid curve) with the equation

$$f(x) = \frac{L}{1 + e^{-k(x - x_0)}}$$

where,
$e$ = The natural logarithm base,
$x_0$ = The x value of the sigmoid's midpoint,
$L$ = The supremum of the values of the function,
$k$ = The logistic growth rate or steepness of the curve.

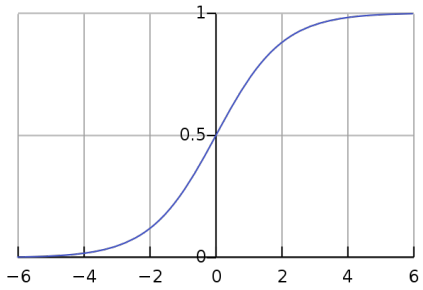



Figure 1: Standard logistic function where $L = 0, k = 1, x_0 = 0$

## 2.2 Decision tree

Decision trees classify data by utilizing a tree structure that is built by segmenting the dataset into several subsets. The outcome is a tree containing leaf nodes and decision nodes. It has a tree-based structure to show the predictions that result from a series of feature-based splits which starts with a root node and ends with a decision made by leaves. Some terms used in the Decision tree:

- Root Node: It is the starting point of the decision tree. The population starts to get split from this particular node based on the features.
- Decision Node: The nodes we get after splitting the root node.
- Leaf Nodes: The last node in the decision tree after which no further splitting of the node is possible.
- Sub-tree: A part of the decision tree.
- Pruning: Process of cutting down some nodes to stop overfitting.

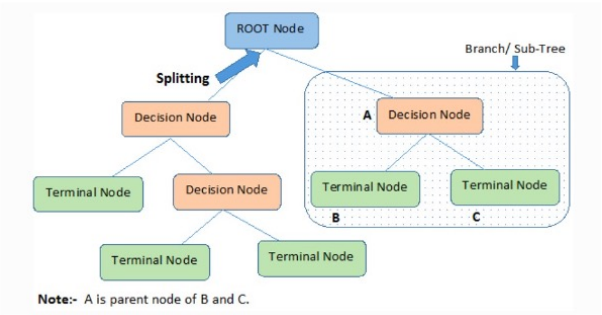



Figure 2: Decision Tree

The Decision Tree Algorithm also tries to deal with the uncertainty in the dataset using ENTROPY. A Pure sub-split is the split in which the entropy of the subset becomes zero indicating that features are perfectly separated.
Formula of Entropy:

$$-p_{+ve}.log(p_{+ve}) - p_{-ve}.log(p_{ve})$$

Entropy reveals a node's impurity, but information gain, which chooses the feature for the root node, can be used to assess how an impurity changed after splitting.

$$\text{Information Gain} = \text{E(Parent)} - \text{E(Parent|Decisive feature)}$$

E(Parent)→The entropy of the Parent Node.
E(Parent|Decisive Feature)→The weighted average of Entropies of each decisive node split out of a particular feature.
The feature which has the largest information gain after the split is picked for root node
In real-world data sets, there are a lot more features and it takes a lot of time to process the Decision tree algorithm as the tree gets more and more complicated. One of the drawbacks of the Decision tree algorithm is, this algorithm won't stop until the entropy reaches 0 and tries to fit to each and every data point, even the noise in the data set. Thus, it leads to overfitting of the model on the training data set.

## 2.3 Random Forest

A Random Forest is a bagging ensemble learning technique which incorporates two fundamental ideas that give it the term random rather than just average the predictions of trees:

- A Random sampling of training observations:
  In a random forest, each tree gains knowledge from a random selection of the training observations. As a result of the samples being drawn using replacement, or bootstrapping, certain samples will be utilized more than once in a single tree. The concept is that by training each tree on many samples, even though each tree may have a large variance relative to a specific set of training data, the overall variance of the forest will be reduced without increasing the bias.

- Random Subsets of features for splitting nodes:
  The random forest's second fundamental idea is that, while deciding whether to divide a node, each tree considers only a portion of all the features. This may be done in Sklearn by setting $max\_features = \sqrt{(n\_features)}$ which means that if a node in a tree has 16 characteristics, only 4 random features will be taken into account when splitting the node.

A Random Forest achieves low variance and low bias by merging predictions from multiple decision trees into a single model, resulting in forecasts that are more accurate on average than predictions from individual decision trees.

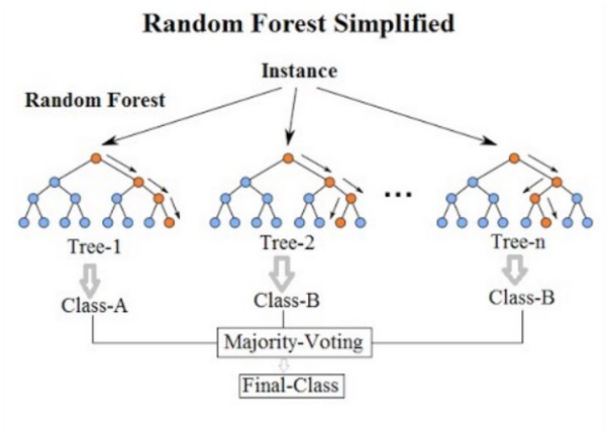



Figure 3: Random Forest

## 2.4 Light GBM

Gradient Boosting Decision Tree has a variant called Light GBM that is very effective. In terms of computation speed and memory use, it can perform noticeably better than XGBoost and SGB. Gradient boosted decision trees are implemented using the Light GBM framework. Because of Light GBM's many benefits, including its quicker training speed, good accuracy with default parameters, parallel and GPU learning, small memory footprint, and capacity to handle huge datasets, we chose to employ it. In Light GBM, the train() technique is used to generate an estimator. It accepts dictionary and training dataset as estimator parameter inputs. The estimator is then trained, and a returned object of type Booster has a trained estimator that can be used to forecast the future.

## 2.5 Evaluation criteria

The model predictions of churn of the customer are evaluated using confusion matrix which is a table with 2 rows and two columns which that reports the number of true positives, false negatives, false positives, and true negatives.


Figure 4: Confusion Matrix

The metrics used to evaluate the models are:

- Accuracy: $\frac{TP+TN}{TP+TN+FP+FN}$
- Precision: $\frac{TP}{TP+FP}$
- Recall: $\frac{TP}{TP+FN}$
- F1 score: $\frac{2(Precision*Recall)}{(Precision+Recall)}$

# 3 Data

The Data used for modeling is originally from the Leaps Analytica website `https://leaps.analyttica.com/home`. We have retrieved the data from Kaagle's Credit card customers - Predict churning customers project from the website `https://www.kaggle.com/datasets/sakshigoyal7/credit-card-customers`. The dataset comprises details on 10000 clients with 21 variables, including their age, gender, level of education, and income, among others.

| Feature | Type | Description |
|---|---|---|
| CLIENTNUM | Categorical | Client Number. Unique identifier for the customer holding the account |
| Attrition_Flag | Categorical | Internal event (customer activity) variable: Attrited Customer or Existing Customer |
| Customer_Age | Quantitative | Age in years |
| Gender | Categorical | Gender of the Account Holder |
| Dependent_count | Quantitative | Number of Dependents |
| Education_Level | Categorical | Educational Qualification of the Account Holder |
| Marital_Status | Categorical | Marital Status of the Account Holder |
| Income_Category | Categorical | Annual Income Category of the account holder |
| Card_Category | Categorical | Type of Card |
| Months_on_book | Quantitative | Period of Relationship with the bank |
| Total_Relationship_Count | Quantitative | Total number of products held by the customer |
| Months_Inactive_12_mon | Quantitative | Number of Months inactive in the last 12 months |
| Contacts_Count_12_mon | Quantitative | The number of times the customer gets contacted by the bank in the last 12 months |
| Credit_Limit | Continuous | Credit Limit on the credit card |
| Total_Revolving_Bal | Continuous (Quantitative) | The balance that carries over from one month to the next is the revolving balance |
| Avg_Open_To_Buy | Continuous (Quantitative) | Open to buy refers to the amount left on the credit card to use |
| Total_Trans_Amt | Continuous (Quantitative) | Total transaction amount (Last 12 months) |
| Total_Trans_Ct | Quantitative | Total number of transactions in the last 12 months |
| Total_Ct_Chng_Q4_Q1 | Quantitative | The ratio of the total transaction count in the 4th quarter and the total transaction count in the 1st quarter. |
| Total_Amt_Chng_Q4_Q1 | Quantitative | The ratio of the total transaction amount in the 4th quarter and the total transaction amount in the 1st quarter. |
| Avg_Utilization_Ratio | Quantitative | Represents how much of the available credit the customer spent |

# 4 Exploratory Data Analysis

We conduct exploratory data analysis (EDA) to have a better understanding of the data by checking for missing and duplicated values, handling outliers, visualizing distributions and plotting graphs to see the relationships between features and our target, which is whether the customer get churned. Here are some important features that needed to illustrate.

## 4.1 Data Understanding

The data set consists of 21 attributes of 10000 customers of a credit card service.
The qualitative features include "Attrition_Flag", "Gender", "Education_Level", "Marital_Status", "Income_Category" and "Card_Category" . The quantitative features include "Customer_Age", "Dependent_count", "Months_on_book", "Total_Relationship_Count", "Months_Inactive_12_mon", "Contacts_Count_12_mon", "Credit_Limit", "Total_Revolving_Bal", "Avg_Open_To_Buy", "Total_Amt_Chng_Q4_Q1", "Total_Trans_Amt", "Total_Trans_Ct", "Total_Ct_Chng_Q4_Q1" and "Avg_Utilization_Ratio".
Our target feature is Attrition_Flag.

## 4.2 Univariate Analysis

### 4.2.1 Categorical Features

1. Attrition_Flag: The categories in the feature are unevenly distributed with Existing Customers accounting for 83.9% of all the customers and Attrited Customer accounting for only 16.1%. we can observe that the data is biased. To overcome this, we will use resampling techniques to make the data unbiased towards any particular class label.

2. Card_category: We observed that blue card are the most commonly used card category and accounts for 93.2% of the credit cards used by customers.

3. There are no null values in the data, but some of the entries in the columns for"Education level","Marital Status," and"Income Category" have "Unknown" values.

### 4.2.2 Numerical Features

Each numerical feature is analyzed by extracting a five-number summary consisting of five values namely the most extreme values in the data set, the lower and upper quartiles, and the median. These values are presented together and ordered from lowest to highest: minimum value, lower quartile (Q1), median value (Q2), upper quartile (Q3), maximum value.
We observe considerable number of outliers in the following quantitative features:

1. Credit Limit
2. Avg_Open_To_Buy
3. Total_Amt_Chng_Q4_Q1
4. Total_Trans_Amt
5. Total_Ct_Chng_Q4_Q1

## 4.3 Bivariate Analysis

### 4.3.1 Numerical-Categorical (Attrition_Flag) analysis

1. We can observe that the average of the total transaction amount of Existing customers is more than the Attrited customers.

2. The average number of Total transactions made by Existing Customers is more than the Attrited customers.

3. Avg Utilization ratio of the Credit Card by the Existing Customers is more than the Attrited Customers.

4. Average Total Revolving Balance of the Existing Customers is more than the Attrited customers.

5. Average number of months of inactivity (in months out of the past 12 months) of Existing Customers is less than Attrited Customers.

#### 4.3.2 Numerical-Numerical Bivariate Analysis

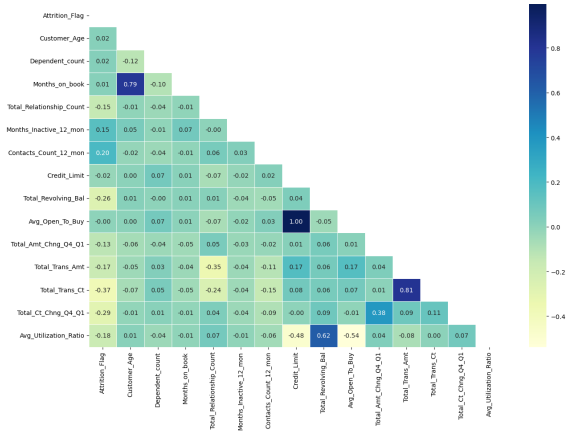



Figure 5: Correlation Graph for Numerical features

After observing the heatmap we can infer that:

1. There is no independent quantitative feature in our data set that is strongly correlated to our Target feature "Attrition_Flag."
2. We can observe that features like Customer_Age, Dependent_count, Months_on_book, Credit_Limit, Avg_Open_to_buy are not correlated to the target variable at all. Thus, we will drop these columns.
3. We can observe multicollinearity among the independent features like Avg_Open_to_Buy and Credit_Limit are perfectly correlated; therefore, we will drop these columns.

# 5 Data Cleaning

## 5.1 Missing values

We noticed that a few of the values in columns 'Education level', 'Marital Status', and 'Income Category' have 'Unknown' as one of the labels. We replace these values and treat them as null values by using Forward Fill (" ffill ") method.

## 5.2 Dropping unwanted columns

We will drop the following list of columns from the data set: CLIENTNUM, Unnamed: 21, Customer_Age, Dependent_count, Months_on_book, Credit_Limit, Avg_Open_to_buy.

# 6 Data Preperation

## 6.1 Encoding the Categorical Features

### 6.1.1 Label Encoding

Label encoding is a process of converting the string values of a categorical variable into numerical value to enhance the performance of machine learning algorithms during training and testing. This method is used when the categories in the feature being encoded is ordinal in nature.
The columns "Education_Level", "Gender", "Attrition_Flag" and "Income_Category" are encoded using this method.

### 6.1.2 One Hot Encoding

One hot encoding is one method of converting data to prepare it for an algorithm and get a better prediction. This method is used when the categories in the feature being encoded are may or may not be ordinal in nature. With one-hot, we create a new category column for each categorical value and give it a binary value of 1 or 0. A binary

vector is used to represent each integer value. The index is denoted by a 1 and all values are zero.
The columns "Marital_Status" and "Card_Category" are encoded using this method.

### 6.2 Dividing the Data set into train, validation, and test data set

We divided our data into train, validation and test data sets using Stratified Random Sampling technique also known as proportionate random sampling or quota random sampling. A sampling strategy employed in this method is the stratification of a population into smaller subgroups known as strata that are generated based on the shared attributes or traits of the individuals. This generated sample population best depicts customer behavior. We used proportional stratified random sampling, which, in contrast to disproportionate sampling, entails selecting random samples from stratified groups in proportion to the target variable.

### 6.3 Feature Scaling

We use Robust Scaling method to scale the independent features in the train, validation, and test data sets. The statistical distribution of the characteristics is quantified using the robust scaler restricting outliers. To scale the characteristics "Total Relationship Count," "Months Inactive," "Contacts Count," "Total Revolving Bal," "Total Amt Change Q4 Q1," "Total Trans Amt," "Total Trans Ct," "Total Ct Change Q4 Q1," and "Avg Utilization Ratio," we utilize the interquartile range robust statistic. The data are scaled using the quantile range after the median is removed by this Scaler.

### 6.4 Resampling the Data Sets

#### 6.4.1 Over-sampling

To balance the data set and make it unbiased, we resample the data using SMOTE (Synthetic Minority Oversampling Technique) algorithm using 'minority' Strategy. Our target variable is highly imbalanced in the dataset. In order to preserve the information on the majority class we upsample the attrited customers using SMOTE. This method randomly duplicates the observations of the minority class.

The SMOTE algorithm can be summarized as follows:

- A random number between 0 and 1 is multiplied by the distance between a sample and its nearest neighbor.
- To create a new synthetic sample in feature space, this is added to the sample.
- The user-defined number of samples is produced by repeating this method.

#### 6.4.2 Under-sampling

To balance the data using under-sampling we used RandomUnderSampler and used the strategy 'majority' to drop the records with majority label class, to make the proportions of the two classes in the target variable equal. According to our sample approach, RandomUnderSampler randomly deletes the rows of the majority class to even the count with minority class.

## 7 Modelling and Explanation

We tested Logistic Regression, Decision Tree, Random Forest Classifier, and Light GBM Classifier models to validate their accuracy on normal, over-sampled, and under-sampled training data set. According to our problem statement, we can formulate a hypothesis:
$H_0$: The customer will not attrite $\rightarrow y = 0$
$H_a$: The customer will attrite $\rightarrow y > 0 => y = 1$
There can be two distinct kinds of errors that could occur while predicting the value for y label:

- Type 1 error: To predict that the customer will get Attrited, but the customer still utilizes the credit card services from the bank.
- Type 2 error: To predict that the customer will not get attrited, but the customer stops using the Credit card services from the bank.

From the above two types of errors, we can observe that the bank would incur more losses due to Type 2 error as they will continue to lose their existing customers and will not be able to take any suitable action to resolve the issue.
Thus, to measure the effectiveness of the model we need to reduce False Negative.
Therefore, to compare the performance of the Classification models we will use Recall Score.
The higher the Recall Score, the lesser the proportion of False Negatives.
We will also use the ROC_AUC Score to compare the performance of the classification models.

| | Model_Name | Train_Acc | Test_Acc | Train_Prec | Test_Prec | Train_Recall | Test_Recall | Train_F1 | Test_F1 | Train_roc_auc | Test_roc_auc | Train_cv_score |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 | Light GBM (Upsampled) | 0.999054 | 0.963390 | 0.998531 | 0.905914 | 0.999580 | 0.861893 | 0.999055 | 0.883355 | 0.999997 | 0.988927 | 0.979197 |
| 6 | Random Forest (Upsampled) | 1.000000 | 0.951460 | 1.000000 | 0.840399 | 1.000000 | 0.861893 | 1.000000 | 0.851010 | 1.000000 | 0.984026 | 0.978777 |
| 11 | Light GBM (Down-Sampled) | 1.000000 | 0.946935 | 1.000000 | 0.786026 | 1.000000 | 0.920716 | 1.000000 | 0.848057 | 1.000000 | 0.984679 | 0.953906 |
| 10 | Random Forest (Down-Sampled) | 1.000000 | 0.935418 | 1.000000 | 0.734940 | 1.000000 | 0.936061 | 1.000000 | 0.823397 | 1.000000 | 0.980508 | 0.949505 |
| 5 | Decision Tree (Upsampled) | 1.000000 | 0.925545 | 1.000000 | 0.745327 | 1.000000 | 0.815857 | 1.000000 | 0.778999 | 1.000000 | 0.881213 | 0.942847 |
| 4 | Logistic Regression (upsampled) | 0.880752 | 0.851501 | 0.873300 | 0.525862 | 0.890733 | 0.780051 | 0.881931 | 0.628218 | 0.946075 | 0.915424 | 0.890313 |
| 9 | Decision Treee (Down-Sampled) | 1.000000 | 0.897162 | 1.000000 | 0.630798 | 1.000000 | 0.869565 | 1.000000 | 0.731183 | 1.000000 | 0.886008 | 0.875944 |
| 3 | Light GBM | 0.999647 | 0.962567 | 0.997809 | 0.926136 | 1.000000 | 0.833760 | 0.998904 | 0.877524 | 0.999999 | 0.990803 | 0.871537 |
| 8 | Logistic Regression (Down-sampled) | 0.853458 | 0.839161 | 0.853070 | 0.500000 | 0.854007 | 0.826087 | 0.853538 | 0.622951 | 0.925458 | 0.920523 | 0.846376 |
| 1 | Decision Tree | 1.000000 | 0.930070 | 1.000000 | 0.804408 | 1.000000 | 0.746803 | 1.000000 | 0.774536 | 1.000000 | 0.856000 | 0.809007 |
| 2 | Random Forest | 1.000000 | 0.952694 | 1.000000 | 0.920732 | 1.000000 | 0.772379 | 1.000000 | 0.840056 | 1.000000 | 0.984858 | 0.800234 |
| 0 | Logistic Regression | 0.905115 | 0.901275 | 0.765292 | 0.764912 | 0.590560 | 0.557545 | 0.666667 | 0.644970 | 0.925584 | 0.925001 | 0.582898 |

Figure 6: Model Comparision

**Observation:** From the above Table we could observe that the Test Recall ratio of the Random Forest Classifier trained on under-sampled train data gives the best recall proportion followed by Light GBM Classifier trained on under-sampled train data.
To reduce the overfitting of the Light GBM and Random Forest Classifier Models and increase the model's accuracy, we will tune the model by introducing the best Hyperparameters for each model.
To identify the best parameters to tune our models we will use the RandomizedSearchCV method. The RandomizedSearchCV method trains the model using different combinations of specified parameters and compares the performance of the estimator (model) with each distinct combination of parameters. The combination of parameters with the best performance is stored.
Below are the set of parameters after tuning the Random Forest Classifier and Light GBM Classifier models:

| **Random Forest Classifier (Tuned) Hyperparameters** |
|---|
| Max_depth = 12 |
| Max_features = 5 |
| Min_samples_leaf = 11 |
| Min_samples_split = 12 |
| N_estimators = 207 |

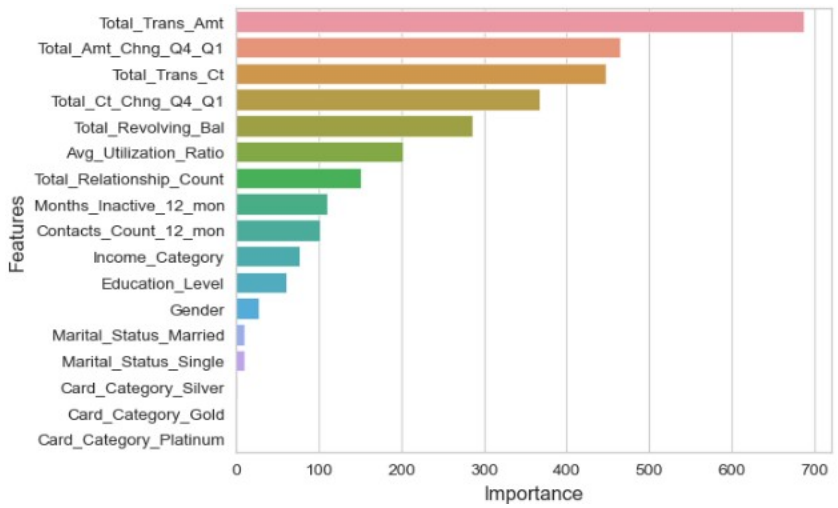
| **Light GBM Classifier (Tuned) Hyperparameters** |
|---|
| Objective = 'binary' |
| Min_gain_to_split = 0.01 |
| Min_data_in_leaf = 25 |
| Metric = 'binary_logloss' |
| Max_depth = 45 |
| Learning_rate = 0.1367 |
| Is_unbalance = True |
| Feature_fraction = 1.0 |
| Extra_trees = False |
| Boosting_type = 'gbdt' |

After tuning the best two models we trained, validated and tested the models and compared the performance of the tuned models.

| | Test Recall Score | Test ROC-AUC Score |
|---|---|---|
| Random Forest Classifier (Tuned) trained on under-sampled data | 0.89 | 0.965 |
| Light GBM Classifier (Tuned) trained on under-sampled data | 0.91 | 0.98 |

**Observation:** From the above results we could easily conclude that after tuning the model Light GBM classification model has better Recall score and ROC-AUC score, thus the best model to predict customer attrition for the bank.

**Feature Importance**




| | Features | Importance |
|---|---|---|
| 8 | Total_Trans_Amt | 687 |
| 7 | Total_Amt_Chng_Q4_Q1 | 465 |
| 9 | Total_Trans_Ct | 447 |
| 10 | Total_Ct_Chng_Q4_Q1 | 367 |
| 6 | Total_Revolving_Bal | 286 |
| 11 | Avg_Utilization_Ratio | 201 |
| 3 | Total_Relationship_Count | 151 |
| 4 | Months_Inactive_12_mon | 110 |
| 5 | Contacts_Count_12_mon | 101 |
| 2 | Income_Category | 76 |
| 1 | Education_Level | 61 |
| 0 | Gender | 27 |
| 12 | Marital_Status_Married | 10 |
| 13 | Marital_Status_Single | 10 |
| 16 | Card_Category_Silver | 1 |
| 14 | Card_Category_Gold | 0 |
| 15 | Card_Category_Platinum | 0 |

Figure 7: Feature Importance

# 8 Conclusion

From the above results we could conclude that:

1. Light GBM model trained on the under-sampled training data set gives the best performance.
2. The most important features to determine the Attrition Rate among the customers of the bank are:
   (a) Total Transaction Amount
   (b) Total Transaction Count
   (c) Total Amount Change Q4 to Q1
   (d) Total Count Change Q4 to Q1
   (e) Total Revolving Balance
   (f) Average Utilization Ratio
   (g) Total Relationship Count
3. All the above features are negatively correlated to the target feature Attrition_Flag, that is, the higher the values of the above features, the lower the chances of the customers getting churned.
4. The bank must engage with their customers more frequently and increase the relationship count with each customer.
5. Bank must come up with some offers or policies where they could decrease the inactivity of the Existing customers and promote the usage of their credit card services.

# 9 Future Scope

1. The attrition rate is an extremely popular and widely researched problem statement in the industry. Each business in the industry wants to retain their customers and reduce Customer Acquisition costs to increase their profit margin.

2. In our project, we used only 4 different Machine Learning algorithms to predict the customers' Attrition rate namely: Logistic regression, Decision Tree Classifier, Random Forest Classifier, and Light GBM Classifiers. In the industry other classification algorithms like Neural network algorithms, genetic programming approaches using the AdaBoost model, and many other models have been utilized.

3. We were provided with the data set for a few customers to train our model on; it would be more accurate and generalized if we supplied more data to the training model.

4. We can use more sophisticated ensembling techniques by combining two or more base classifiers.

# References


[1] Gustafsson, A., Johnson, M.D., Roos, I.: The effects of customer satisfaction, relationship commitment dimensions, and triggers on customer retention. Journal of Marketing 69(4), 210–218 (2005)

[2] Roberts, J.H.: Developing New Rules for New Markets. Journal of the Academy of Mar- keting Science 28(1), 31–44 (2000)

[3] Slater, S.F., Narver, J.C.: Intelligence Generation and Superior Customer Value. Journal of the Academy of Marketing Science 28(1), 120–127 (2000)

[4] Kotler, P.: Marketing Management. Prentice-Hall, NJ (2000)

[5] Lu, J.: Predicting Customer Churn in the Telecommunications Industry -— An Application of Survival Analysis Modeling Using SAS. Sprint Communications Company

[6] Glady, N., Baesens, B., Croux, C.: Modeling Churn Using Customer Lifetime Value. European Journal of Operational Research (2008), doi:10.1016/j.ejor.2008.06.027

[7] Van den Poel, D., Larivi‘ere, B.: Customer attrition analysis for financial services using proportional hazard models. European Journal of Operational Research 157(1), 196–217 (2004)

[8] Buckinx, W., Van den Poel, D.: Customer base analysis: Partial defection of behaviorally-loyal clients in a non-contractual fmcg retail setting. European Journal of Operational Re-search 164(1), 252–268 (2005)

[9] Glady, N., Baesens, B., Croux, C.: Modeling Churn Using Customer Lifetime Value. European Journal of Operational Research (2008), doi:10.1016/j.ejor.2008.06.027

[10] Neslin, S.A., Gupta, S., et al.: Defection Detection: Improving Predictive Accuracy of Customer Churn Models (2004)

[11] G. L. Nie, W. Rowe, L. L. Zhang, Y. J. Tian, and Y.Shi, "Credit card churn forecasting by logistic regression and decision tree," Expert Systems with Applications, vol. 38, pp. 15273-15285, 2011.

[12] Q. F. Bi, K. E. Goodman, J. Kaminsky, and J. Lessler, "What is Machine Learning? A Primer for the Epidemiologist," American Journal of Epidemiology, vol. 188, pp. 2222-2239, October 2019.

[13] An Introduction to Logistic Regression Analysis and Reporting CHAO-YING JOANNE PENG KUK LIDA LEE GARY M. INGERSOLL