**Reducing Customer Churn in Banking Industry by classifying Attrition using Clustering and other Classification Models. (EC784P)**

**Abstract**

This project will represent the analysis made on a data set which represents the customer churn in a Banking Industry and means to classify customers based on certain info about them (demographic, banking habits, etc) using clustering and multiple classification algorithms and thus expressing which method is superior for the application and the use case along with observing the difference in their performance.

The methodologies we have used for classification are k-NN (k Nearest Neighbors) and Logistic Regression and for clustering we use k-Means.

The data set we are using is a .csv file which was procured from kaggle and has 23 features (columns) . Out of which there are 6 categorical data, 16 numeric data and our target variable here will be the Attrition\_Flag.

In this report, we shall be applying Exploratory Data Analysis on this data set, trying to understand the data better specifically what each feature signify. We shall process the data by cleaning the data, and then apply feature engineering to find out the best feature that can help us solve our problem and then apply the machine learning algorithms mentioned along with giving a brief explanation on how these algorithms work for our data set and what conclusions can be drawn from them and then finally compare these models to better understand the problem and choose which solution works best.

**Introduction**

The aim of the project is to analyze a data set of bank customers and classify them based on their attrition flag – whether they are an “Attrited Customer” or an “Existing Customer” .

To understand the problem, we first need to understand “***Churn***” and “***Attrition***”, Churn and attrition are similar concepts used to describe customers who have quit using a service. The main distinction between the two terms is that attrition may also include customers who have decreased their engagement or activity with the bank but have not necessarily terminated their relationship, whereas churn typically refers to customers who have terminated their relationship with a bank.

For example, in the banking business, a customer who shuts their account is termed as a churned customer, where as a consumer who reduces their account activity, such as completing fewer transactions, may be labeled an attrited customer.

Attrition can have a significant impact on the effectiveness of retention campaigns for banks. A bank may see a drop in income and profitability as well as a loss of market share to rivals if it is unable to recognize and keep customers who are more likely to leave. A bank's retention effort may be impacted by attrition in a number of ways, including *Decreased effectiveness of retention efforts*, *Reduced customer loyalty*, *Negative impact on brand image* and *Increased competition*.

Machine learning classification algorithms can be used to solve the problem of classifying bank customers as either "Attrited" or "Existing". Machine learning models can use customer data to analyze patterns and links between various attributes and customer churn, and then use these associations to predict which customers are more likely to leave.

Banks may utilize machine learning algorithms to pinpoint the essential characteristics—like account balance, transaction history, age, and income—that are most indicative of client churn. Banks can create tailored retention strategies that are more likely to be successful by concentrating on these essential characteristics. Banks may also identify which customers are more likely to quit the bank by analyzing past customer data and machine learning models, and they can take proactive steps to keep those customers before they go. In order to improve the entire customer experience and lower customer turnover, banks can use machine learning algorithms to evaluate client interactions and find behavioral patterns linked to increased customer satisfaction and loyalty. Banks can lower the cost of recruiting new customers, which is typically more expensive than keeping existing ones, by utilizing machine learning to predict customer churn and enhance client retention

**Data Cleaning an EDA**

The data is obtained from the zhyli. (2020). Prediction of Churning Credit Card Customers [Data set]. Zenodo.https://doi.org/10.5281/zenodo.4322342. We can now start by exploring the data to understand what each columns hold. On first glance by printing the head of the data frame, we can see that the data set has a combination of categorical and numerical data. Along with that we can observe that the Attrition\_Flag displays the flags for the customers indicating churn. This is our target variable (Y) and we shall drop this column when we will be applying our machine learning models. The data set has features, “Naive\_Bayes\_Classifier\_Attrition\_Flag\_Card\_Category\_Contacts\_Count\_12\_mon\_Dependent\_count\_Education\_Level\_Months\_Inactive\_12\_mon\_1” and “Naive\_Bayes\_Classifier\_Attrition\_Flag\_Card\_Category\_Contacts\_Count\_12\_mon\_Dependent\_count\_Education\_Level\_Months\_Inactive\_12\_mon\_2”. As per the source these columns are classification result for this problem using Naive Bayes Classification. As applying any classification methodology to the data set with these two features as a part of out selected features could lead to model giving is in accurate results, it will be best to drop them.

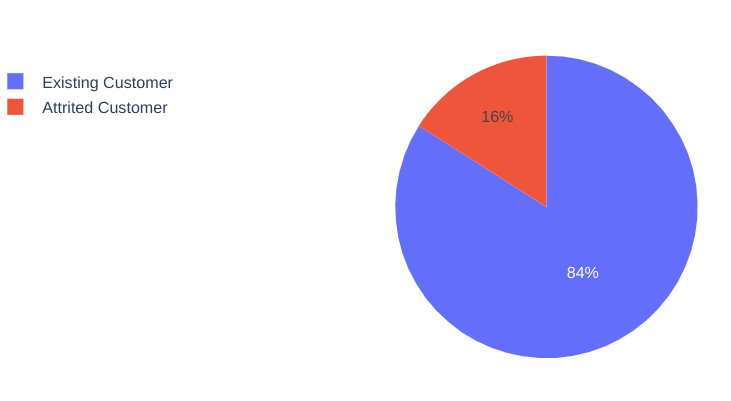
Before beginning any exploratory analysis, it is best to check for missing data, Unknown vales and null or zero values. After applying the isnull().sum() function on our data frame, we can see that there are no null values present in any of our columns.

|  |  |
| --- | --- |
| Feature Name | Description |
| CLIENTNUM | Unique identifier (Int) |
| Attrition\_Flag | Flag indicating churn (Boolean) |
| Customer\_Age | Age (Int) |
| Gender | Gender (String) |
| Dependent\_count | Number of dependents (Int) |
| Education\_level | Education level (String) |
| Marital\_Status | Marital Status (String) |
| Income\_Category | Income Category (String) |
| Card\_Category | Type of card (String) |
| Months\_on\_book | Months on book (Int) |
| Total\_Relationship\_Count | No. of relationships with credit card provider (Int) |
| Months\_Inactive\_12\_mon | Inactivity in 12 months (Int) |
| Contacts\_Count\_12\_mon | Contacts in 12 months (Int) |
| Credit\_Limit | Credit limit (Int) |
| Total\_Revolving\_Bal | Total Revolving balance (Int) |
| Avg\_Open\_To\_Buy | Average open to buy ratio (Int) |
| Total\_Amt\_Chng\_Q4\_Q1 | Total Amount change fro Q4 to Q1 (Int) |
| Total\_Trans\_Amt | Total Transaction Amount (Int) |
| Total\_Trans\_Ct | Total Transaction Count (Int) |
| Total\_Ct\_Chng\_Q4\_Q1 | Total count change from Q4 to Q1 (Int) |
| Avg\_Utilization\_Ratio | Average utilization ratio (Int |
| NaiveBayes\_Attrition\_1\_Classification | Naive Bayes classifier using one set of parameters (Int) |
| NaiveBaues\_Attrition\_2\_Classification | Naive Bayes classifier using one set of parameters (Int) |

**Table 1: Feature Table**

In addition to checking for null values, it also necessary to check for any duplicates in the data set by using the CLIENTNUM column in the data. We can also check for Unknown and 0 values in the data set by using count() and unique() functions as per the requirement. Duplicates and other such discrepancy can cause some issues when we try to classify the churn. Missing values can cause the model to be biased and thus not perform well. Duplicated data will harm the test - train split as the duplicates can be split as well leading to more bias.

Continuing with our EDA, we can check the data split between Existing Customer and Attrited Customer using a pyplot.



As you can see, the 84% of the customers in the data set are existing customers and 16% are Attrited Customers. We can print the count of these flags based on the client ID. As per the data, 1627 customers are attrited and 8500 are existing customers. We have some categorical data. By applying chi\_2\_contingency we can see the difference between the categories. This will help us determine if these categories are the features that can be considered for our model. The following table represents the chi2 values of our categorical data. Here we shall focus on the p-values as they can be considered as evidence of a statistically significant result. If p-values > 0,05, we can consider that the null hypothesis for that feature is **True**.

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Chi\_2 | P\_value | DOF |
| Gender | 13.86 | 0.00019 | 1 |
| Income | 12.83 | 0.02500 | 5 |
| Marital\_Status | 6.056 | 0.10891 | 3 |
| Card\_Category | 2.234 | 0.52524 | 3 |
| Education\_Level | 12.51 | 0.051489 | 6 |

**Table 2: Chi Squared Statistics**

After observing the table, we can deduce that only Gender and Income have a relationship with the Attrition Flag since their p-values are 0.00019 and 0.02500 respectively which are < 0.05.

Moving on to numeric features, we can check their significance using box plot.

|  |  |
| --- | --- |
| Dependent\_Count | Total\_Relationship\_Count |
| Months\_Inactive\_12\_mon | Contacts\_Count\_12\_mon |
| Total\_Revolving\_Bal | Total\_Trans\_Amt |
| Total\_trans\_Ct | Avg\_Utilization\_Ratio |

The above feature are the only ones that have some sort of relation with the Attrition. Here Blue plots represent Existing Customers– and the orange plots resemble Attrited Customers. The other features show no or very small difference in their plots signifying very very slight to no relationship with our target variable.

**Feature Engineering.**

To ensure that all data is accounted for in respect to the attrition flag and to better understand the relationship with the attrition, categorical data should be converted into numerical data. For this purpose we can use the ***LabelEncoder()*** function from sklearn.preprocessing. Label Encoder essentially assigns each categorical value to an integer value based on alphabetical order. Thus we shall be left with new data frame, ***df\_main*** which comprises of t––––he numerical features and the converted features. We can verify if the operation was successful by printing the first 20 rows of the data frame. Since now our data is ***normalized***, we can create a heatmap based on the correlation of each feature. We have used the seaborn library for this.

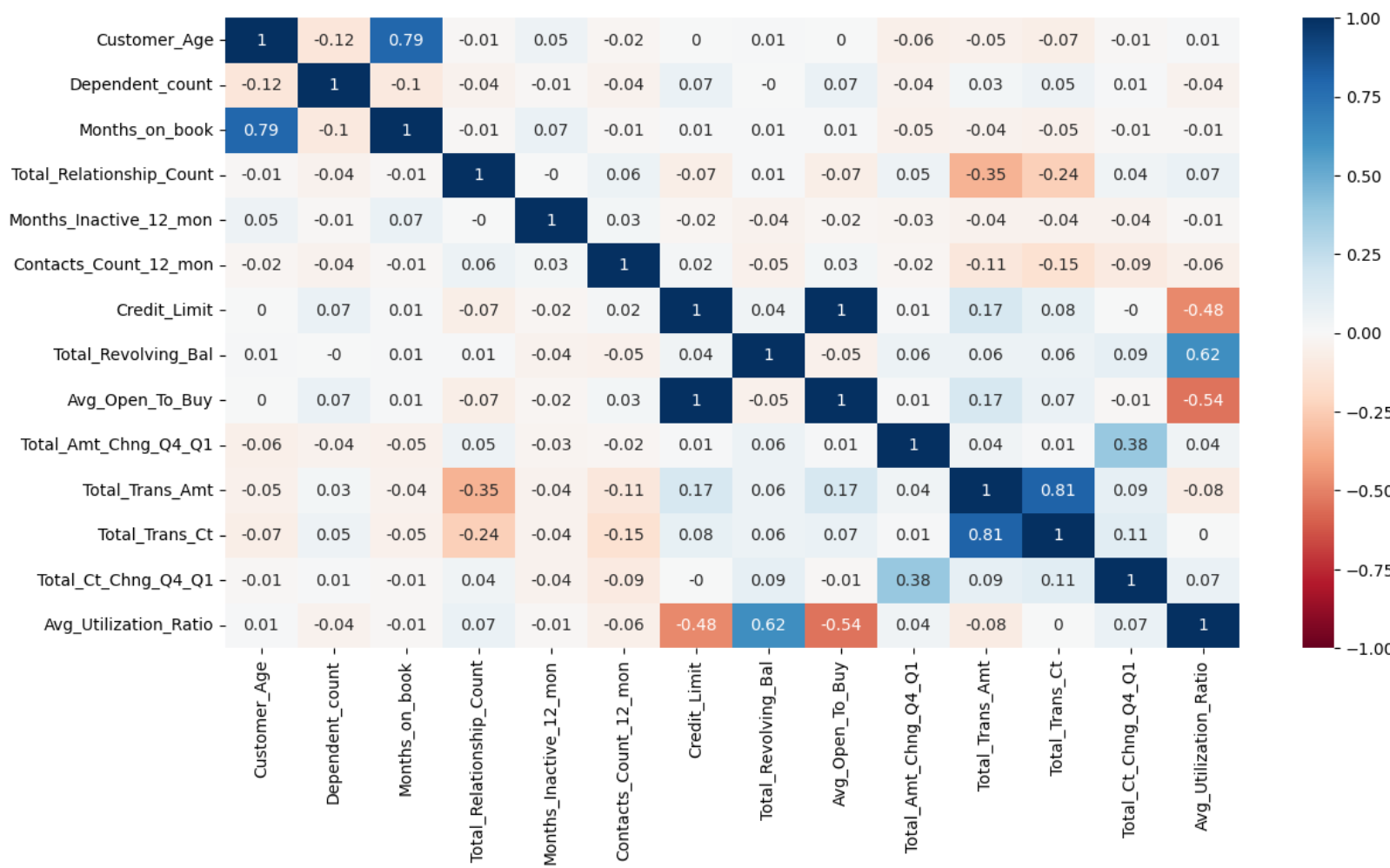


Figure : Correlation Matrix

Observing the correlation matrix we can conclude the following:

* Total\_Revolving\_bal and Avg\_Utilization\_Ratio are positively correlated
* Avg\_Utilization\_Ratio and Avg\_Open\_To\_Buy are negatively correlated
* Avg\_Utilization\_Ratio and Credit\_Limit are negatively correlated.

**Feature Selection.**

For the machine learning models to perform well, it needs best features to obtain the target variable To find this, we can consider the features as a hyperparameter. To perform hyperparameter tuninig, a combination of Random Fores, GridSearchCV and Stratified k-fold Cross Validation. This process aims to identify the optimal combination of hyperparameters that maximizes the F1 score, which is used as the evaluation metric. The params dictionary specifies the hyperparameters to be tuned, which are max\_depth, min\_samples\_split, and min\_samples\_leaf. The s\_kfold object is employed to partition the data into training and validation sets.

GridSearchCV is used to perform an exhaustive search over the specified hyperparameters, utilizing cross-validation to avoid overfitting. The n\_jobs parameter is set to -1 to utilize all available CPU cores for faster processing. Once the best hyperparameters have been identified, the best\_params\_ and best\_score\_ attributes of the GridSearchCV object are used to display the best hyperparameters and corresponding F1 score.

Additionally, the code uses the feature\_importances\_ attribute of the best estimator to calculate the relative importance of each feature in predicting the target variable. These feature importances are then displayed using a Pandas Series.