A

Project Report

On

**“CREDIT CARD USERS CHURN PREDICTION”**

**ACKNOWLEDGEMENT**

**ABSTRACT**

In the current trading environment, banking and financial guests have a large number of consumers. Banks provide aids through different channels, like ATMs, entry cards, credit cards, internet banking, etc. The number of customers has increased significantly, and customers' awareness of the impact of quality has also increased. This intensifies the overwhelming competition among the various banks, which greatly increases the reliability and value of the banks. Also, consumers shift loyalties from one bank to another by way of different reasons, as much as the availability of new electronics, customer-friendly bank stick, depressed-interest rates, the closeness of the geographic region, the differing aids offered.Hence, skilled is an urgent need to evolve a model which can conclude that consumer is likely to beat out established the consumers’ demographic, psychographic and variable dossier.

An direct Customer Relationship Management (CRM) decision group providing support to members increases the kind of consumer relationships, through growing memory in several habits: It supports predicting shaping to help banks identify the one is inclined leave and reason and what to do about It allows a new level of embodiment functional offers and marketing approaches, that promotes dependability. ItAs customer satisfaction increases, it brings better richness to consumer interactions as consistent messages are shared across all customer touchpoints (CRM in Banking, SAS White Paper, 2001). The miracle of consumer churn is not limited to the investment and treasury industries. In the excess of additional service activities, it has performed extremely well in mobile electronics, radio ratings, etc. The aim of this investigation search to amplify a prescient model for customer stir in the Debit Card's customers' beat that can recognize customers. One undoubtedly disturbs the fate of intimacy, while the other accompanies the arrangement.

The commitment of a particular model for the organization is that it would forestall the misuse of cash because of the mass promoting approaches and it empowers the organizations to focus on the genuine churners by removing the clients with a high likelihood of agitating. Plus, as talked about in the introduction the expense of procuring another client is multiple times more than holding a current one, subsequently since the churn prescient model is fit for demonstrating the future churners, the organizations that are expected to keep up their client base can zero in on maintenance approaches rather than obtaining approaches which are less exorbitant. Summarizing, for the finding of this exploration, we can recommend the organizations to use the data mining strategies to change the current client data in their databases to exploitable information that can help them in their showcasing plans.

Besides, it would VIII be valuable for them to construct a prescient churn model by the utilization of data mining which assumes the part of a cautioning framework for the organizations, and furthermore, it can assist them with spending their maintenance financial plan proficiently.

Every year, many customers buy credit cards, and many customers leave so we need to forecast future churn in the credit card company.

Credit card churn prediction can assist a bank to know which of his customers is going to retain and can also assist the bank to know on which customer he needs to emphasize more so that he can earn the highest gains.

Future churn can be estimated by evaluating recent industry patterns and churn rate, as well as potential innovations.

Total transaction, Total transaction count, and total relationship count matters a lot in retaining the customer. The functionality is based on a website that recognizes customer requirements and then incorporates the use of data mining's multiple regression algorithm and Auto-ML. This study aims to use classification analysis and Auto-ML to predict Customer Churn based.

**CONTENTS**

CHAPTER 1: INTRODUCTION TO THE PROJECT

CHAPTER 2: LITERATURE REVIEW & ANALYSIS

CHAPTER 3: EDA AND DATA PREPROCESSING

CHAPTER 4: MODEL EVALUATION CRITERION

CHAPTER 5: APPLYING DIFFERENT ML MODELS TO THE ORIGINAL

DATA

CHAPTER 6: CONCLUSION AND FUTURE SCOPE

CHAPTER 1: INTRODUCTION

****

Customer Churn (Customer attrition) is the most challenging problem for businesses such as credit cards or telecommunication companies etc. It would be nice to build models to predict who churn or retain for their service. We will be looking into the churn of customers for using credit card services.

**Problem Statement**

Thera bank recently saw a steep decline in the number of users of their credit card, credit cards are a good source of income for banks because of different kinds of fees charged by the banks like annual fees, balance transfer fees, and cash advance fees, late payment fees, foreign transaction fees, and others. Some fees are charged to every user irrespective of usage, while others are charged under specified circumstances.

Customers’ leaving credit card services would lead banks to loss, so the bank wants to analyze the data of customers and identify the customers who will leave their credit card services and the reason for same – so that the bank could improve upon those areas.

You as a Data scientist at Thera bank need to come up with a classification model that will help the bank improve its services so that customers do not renounce their credit cards.

**We need to identify the best possible model that will give the required performance**

**Data Description:**

* CLIENTNUM: Client number. Unique identifier for the customer holding the account
* Attrition\_Flag: Internal event (customer activity) variable - if the account is closed then "Attrited Customer" else "Existing Customer"
* Customer\_Age: Age in Years
* Gender: Gender of the account holder
* Dependent\_count: Number of dependents
* Education\_Level: Educational Qualification of the account holder - Graduate, High School, Unknown, Uneducated, College (refers to a college student), Post-Graduate, Doctorate
* Marital\_Status: Marital Status of the account holder
* Income\_Category: Annual Income Category of the account holder
* Card\_Category: Type of Card
* Months\_on\_book: Period of relationship with the bank (in months)
* Total\_Relationship\_Count: Total no. of products held by the customer
* Months\_Inactive\_12\_mon: No. of months inactive in the last 12 months
* Contacts\_Count\_12\_mon: No. of Contacts in the last 12 months
* Credit\_Limit: Credit Limit on the Credit Card
* Total\_Revolving\_Bal: Total Revolving Balance on the Credit Card
* Avg\_Open\_To\_Buy: Open to Buy Credit Line (Average of last 12 months)
* Total\_Amt\_Chng\_Q4\_Q1: Change in Transaction Amount (Q4 over Q1)
* Total\_Trans\_Amt: Total Transaction Amount (Last 12 months)
* Total\_Trans\_Ct: Total Transaction Count (Last 12 months)
* Total\_Ct\_Chng\_Q4\_Q1: Change in Transaction Count (Q4 over Q1)
* Avg\_Utilization\_Ratio: Average Card Utilization Ratio

**What Is a Revolving Balance?**

If we don't pay the balance of the revolving credit account in full every month, the unpaid portion carries over to the next month. That's called a revolving balance

**What is the Average Open to buy?**

Open to Buy' means the amount left on your credit card to use. Now, this column represents the average of this value for the last 12 months.

**What is the Average utilization Ratio?**

The Avg\_Utilization\_Ratio represents how much of the available credit the customer spent. This is useful for calculating credit scores.

**Relation Between Avg\_Open\_To\_Buy, Credit\_Limit and Avg\_Utilization\_Ratio:**

(Avg\_Open\_To\_Buy / Credit\_Limit) + Avg\_Utilization\_Ratio = 1

**Research Questions to be answered:**

* What are the main reasons for a customer to churn from the usage of a credit card?
* Can the important features be obtained from the bank and customer information stored which drives the churn?
* What are the things that can be done to retain customers to a bank?
* Can promotional or other offers be provided to a set of customers who are on verge of being churn?

**Why are these questions important?**

* Banks generate a lot of revenue from customers using credit cards. It is evident that 30% of total revenue comes from the usage of cards so it is important for the banks to maintain a healthy relationship with the customers who use credit cards.
* Also customers tend to churn from the credit card usage when there is a decrease in the offers that they receive from their cards so it is important to decide over the required promotional offers that can be given to customers.
* Also the offers over credit cards seem to decrease the rate of churn and increase its usage over various online shopping or purchases so this can also be a trigger for banks to maintain good relationships with ecommerce companies.

CHAPTER 2: Literature Review

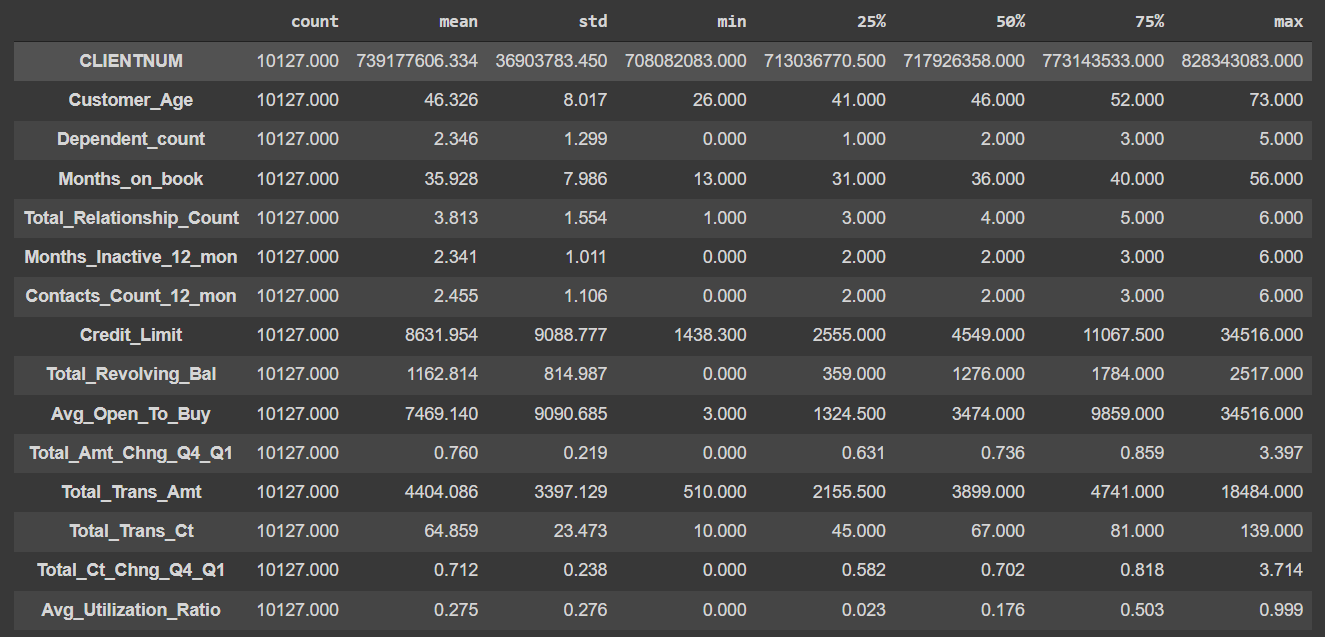
A researcher(Dr Lucas) focused on the credit card transactions that were focused on both fraud and non-fraud. And as known already, no payments company would make its data publicly available when it comes to transactions so there is less data related to credit card fraud detection in the real-world. This researcher has used a PCA transformed form of credit transactions, He concentrated mainly on the process of treating the data imbalance problem and used several undersampling and oversampling approaches and finally made it thoughtful using SMOTE. He then tried to encode target variables and then use a deep learning approach called sequential approach which used RNNs as their backbone.

Similarly in other research, it is also cited of the usage of SVM to predict churn of users, Here the researcher has received about 91% accuracy with the SVM model by just using a linear kernel.

## **Exploratory Data Analysis**

As we have looked into several of the different attributes of a customer that we would likely use in predicting the reason for churn in the usage of credit cards.

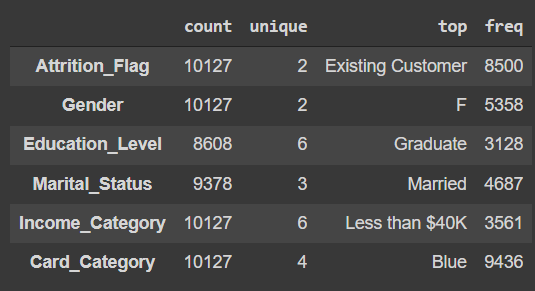
First, we will try to look into the statistical and numerical analysis of the different columns present in the data and will try to gain some analysis from this statistical information.



We can have several observations from the above chart:

* **CLIENTNUM**: It is a unique identifier for customers and can be dropped as it wouldn't add any information to our analysis.
* **Customer\_Age**: Average age of clients is 46 age; the age of consumers has a wide range from 26 to 73 age.
* **Dependent\_count**: On average the customers in the data have 2 dependents and a maximum of 5 dependents.
* **Months\_on\_book**: All the customers of the bank have at least been with them for a year and 50% of the customers for at least 3 years.
* **Total\_Relationship\_Count**: All consumers use not completely individual products of the bank, in as much as 75% of clients use 5 or minor commodities of the bank.
* **Months\_Inactive\_12\_mon**: On average customers were inactive for two months in the past 12 months - this shows that the bank customers are active in transactions or usage of cards it would be interesting to see if high inactivity leads to churning of a customer.
* **Contacts\_Count\_12\_mon**: On average banks and customers interacted twice in the past 12 months.
* **Credit\_Limit**: There's a huge difference between the third quartile and maximum value. The range of credit limits is very wide from 1438 to 34516, customers with high credit limit might be outliers.
* **Total\_Revolving\_Bal**: Average revolving balance of customers is 1162, there's not much difference in the third quartile and maximum value.
* **Avg\_Open\_To\_Buy**: Average amount that goes unused by the customers is 7469, the range is very wide for this variable and the extreme values (min and max) might be outliers.
* **Total\_Amt\_Chng\_Q4\_Q1**: For 75% of the customers the transaction amount in Q4 was less than the transaction amount in Q1 (as value is equal to ~0.9).
* **Total\_Trans\_Amt**: Average transaction amount of last 12 months is 4404, some customers spent as little as 510 while some customers made the transaction of more than 18k.
* **Total\_Trans\_Ct**: On average customers made 64 or fewer transactions while 75% of the customers made 81 transactions.
* **Total\_Ct\_Chng\_Q4\_Q1**: For 75% of the customers the number of transactions in Q4 was less than the transactions made in Q1.
* **Avg\_Utilization\_Ratio**: On average customers used ~27% of the available credit amount of their card, with 75% of the customers utilizing 50% or less of their available credit amount.

Now when we try to have a look into the unique values present in the object category



We can observe that:

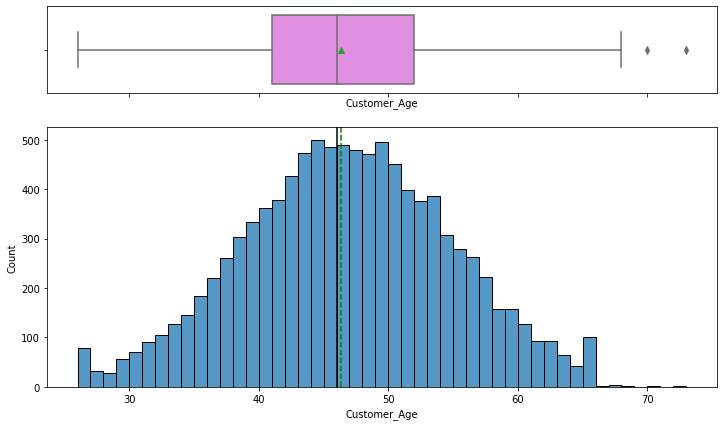
* Most of the records are for existing customers.
* Most of the bank's customers are female
* Most customers are graduates.
* Most customers are married.
* Most customers lie in the income group of less than $40k
* Most customers have a blue card.
* 'abc' value of Income\_Category can be considered and treated as missing values.

Now we will try to look into the univariate analysis on each attribute in the data.

The univariate analysis is performed using the histogram\_box() function and it can be seen as:

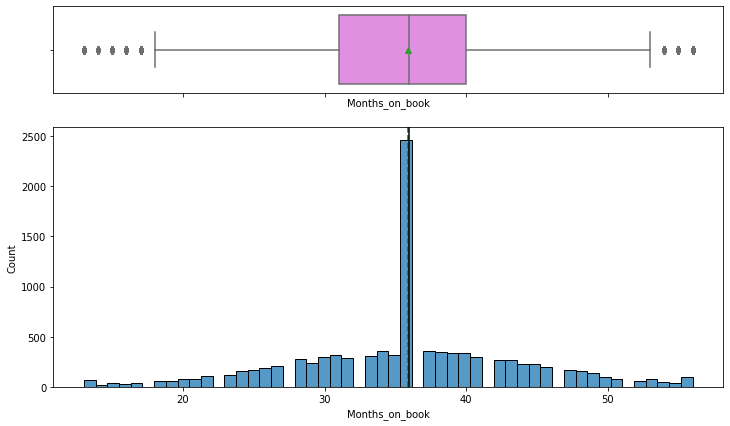
| *# writing a function to plot a boxplot and a histogram along the same scale.*  def histogram\_box(data, feature, figsize=(10, 8), kde=False, bins=None):  """  Boxplot and histogram combined   data: dataframe  feature: dataframe column  figsize: size of figure (default (12,7))  kde: whether to the show density curve (default False)  bins: number of bins for histogram (default None)  """  f2, (ax\_box2, ax\_hist2) = plt.subplots(  nrows=2, # Number of rows of the subplot grid= 2  sharex=True, # x-axis will be shared among all subplots  gridspec\_kw={"height\_ratios": (0.25, 0.75)},  figsize=figsize,  ) # creating the 2 subplots  sns.boxplot(  data=data, x=feature, ax=ax\_box2, showmeans=True, color="violet"  ) # boxplot will be created and a star will indicate the mean value of the column  sns.histplot(  data=data, x=feature, kde=kde, ax=ax\_hist2, bins=bins, palette="winter"  ) if bins else sns.histplot(  data=data, x=feature, kde=kde, ax=ax\_hist2  ) # For histogram  ax\_hist2.axvline(  data[feature].mean(), color="green", linestyle="--"  ) # Add mean to the histogram  ax\_hist2.axvline(  data[feature].median(), color="black", linestyle="-"  ) # Add median to the histogram |
| --- |

**Observations on Customer\_age:**

****

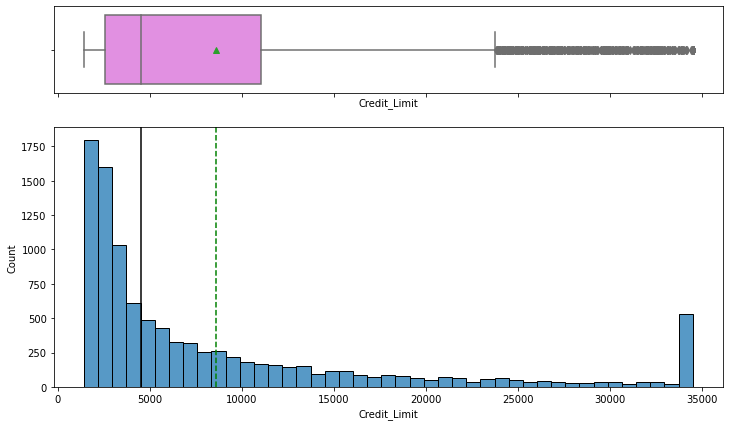
* The distribution of Customer\_Age is normally distributed with mean and median at 46 years.
* From the boxplot, we can see that there are a few outliers.

**Observations on Months\_on\_book:**

****

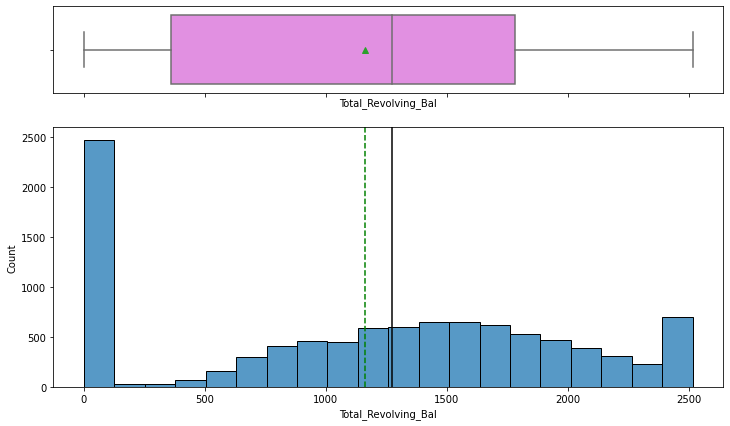
* Most customers are with the bank for 3 years.
* From the boxplot, we can see that there are outliers on both sides of the whiskers.

**Observations on Credit\_Limit:**



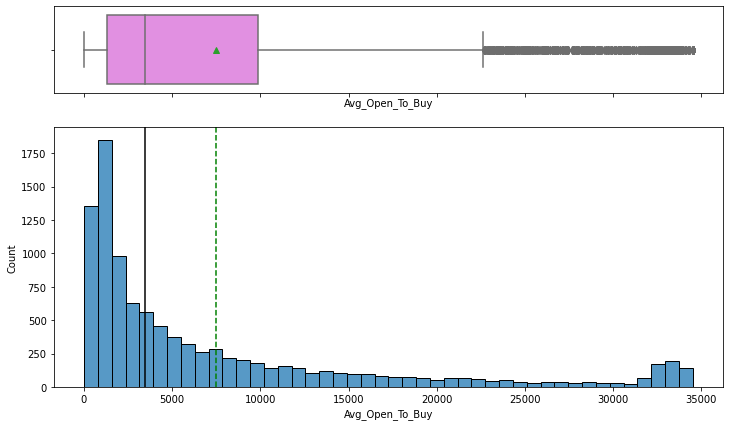
* The distribution of the Credit\_Limit is skewed to the right.
* There are quite a few customers with a maximum Credit Limit of 35000.
* 50% of the customers of the bank have a credit limit of less than <5000.

**Observations on Total\_Revolving\_Bal:**



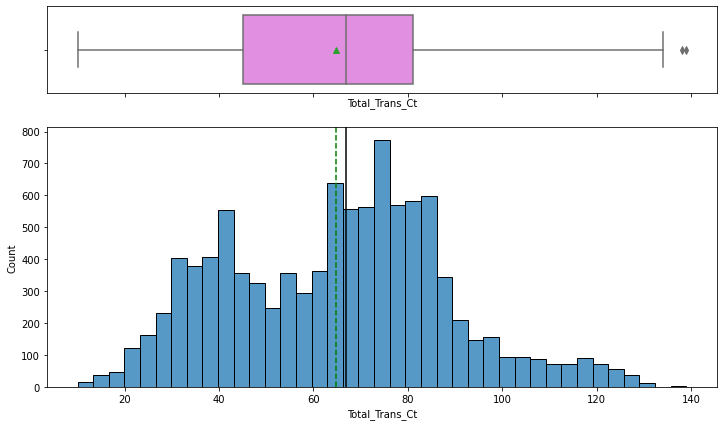
* Most customers pay the complete dues of credit card and have 0 revolving balance.
* There are quite a few customers with a revolving balance of 2500.

**Observations on Avg\_Open\_To\_Buy:**



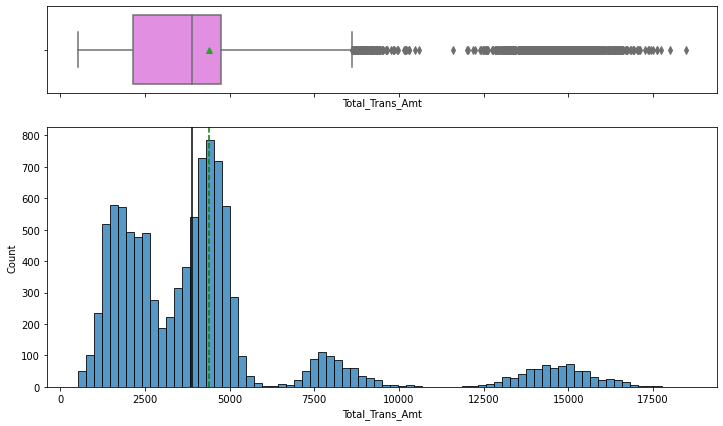
* The distribution of the Avg\_Open\_To\_Buy column is right-skewed.
* A right-skewed distribution indicates that most customers used a big part of their limit while only a few customers (on the right tail) were left with a majority of their credit amount

**Observations on Total\_Trans\_Ct:**



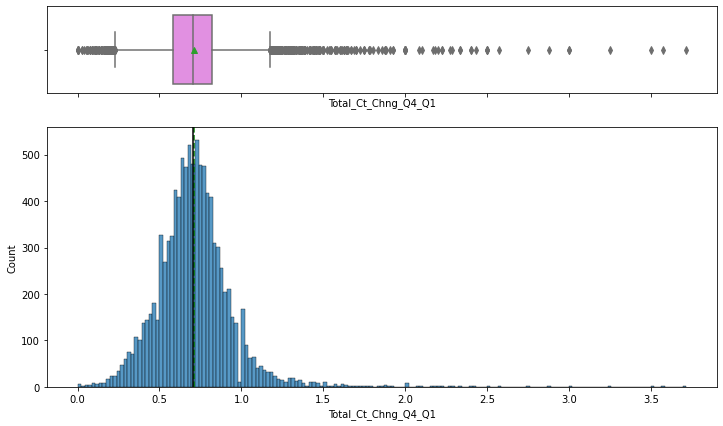
* The distribution of Total\_Trans\_Ct shows two peaks on 40 and 80 transactions in a year which indicates that customers used credit cards 3 to 6 times a month to make transactions.

**Observations on Total\_Trans\_Amt:**



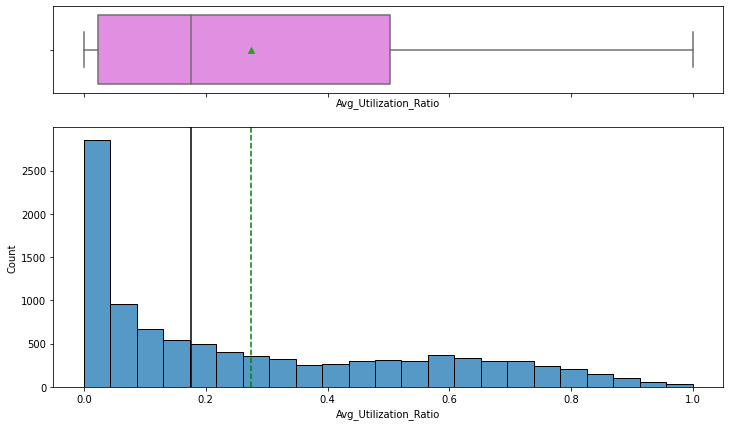
* The distribution of Total\_Trans\_Amt is skewed to the right.
* There are two peaks in data at total transaction amounts of one around 2500 and the second around the mean value of ~4500.
* From the boxplot, we can see that there are outliers - customers with more than ~8000 total transaction amounts are being considered as outliers.
* It would be interesting to check if the customers spending less with the card are the ones churning or the ones spending more are churning, if the latter is the case, then there is a problem for the bank as it is losing valuable customers.

**Observations on Total\_Ct\_Chng\_Q4\_Q1:**



* The distribution of Total\_Ct\_Chng\_Q4\_Q1 looks normally distributed but there's a slight skew towards the right.
* From the boxplot, we can see that there are outliers on both sides of the whiskers.

**Observations on Avg\_Utilization\_Ratio:**



* The distribution of Avg\_Utilization\_Ratio is skewed to the right.
* This distribution is not a positive sign for the bank as most of the customers are not utilizing their credit amount.

**Some of the important observations from the univariate analysis**

**Credit limit, Average open to buy, and Average utilization ratio is right-skewed**

1. Open to buy means how much credit a customer is left with
   * Low values of Open to buy could represent either

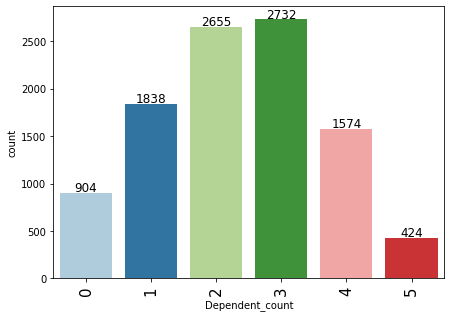
* Customers have low credit limits
* Customers are spending a lot so they are left less open to buy

1. Average utilization ratio = (1 - (open to buy/credit limit))
   * Low values of the Average utilization ratio represent
   * (Open to buy/credit limit) is nearly equal to 1
     + Open to buy is nearly equal to the credit limit
     + Customers are spending less using their credit cards
2. Credit limit is also right-skewed which represents - most of the customers have low credit limits.

Looking at the 3 variables, we can conclude that most of the customers have low credit limits and are not utilizing their credit cards much

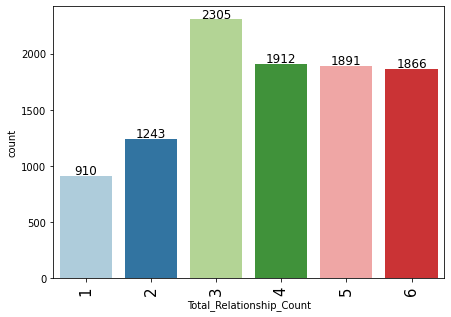
Now this statement justifies the right skewness for all 3 variables.

**Observations on Dependent\_count:**



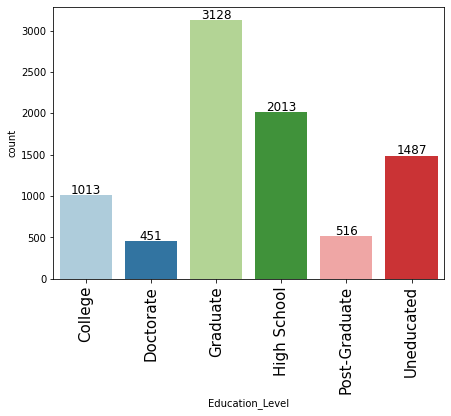
* The distribution of dependents is very realistic with most customers having 2 or 3 dependents.

**Observations on Total\_Relationship\_Count:**



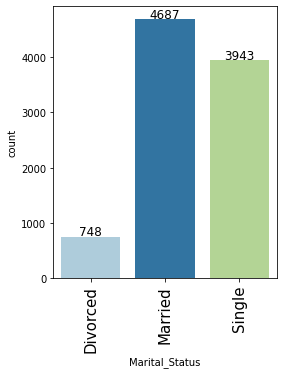
* 22.8% of the customers are using 3 products offered by the bank while an equal percentage ~19% of customers use 4 or more than 4 products.
* Equal percentage of customers using 4,5 and 6 products might suggest that customers who opt or buy the 4th product will also be ready to buy more products.

**Observations on Education\_Level:**

****

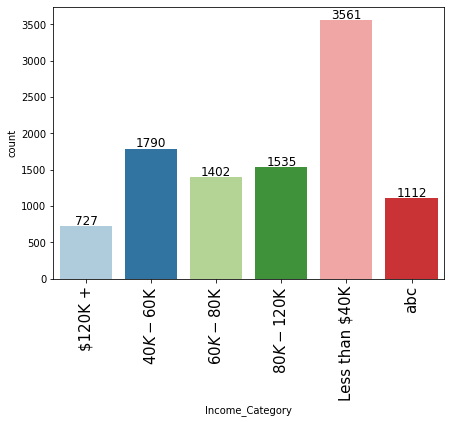
* 30.9% of the customers are graduates, followed by 19.9% of the customers who completed high school.
* Percentage of missing value in Education\_Level column - 15%.

**Observations on Marital\_Status:**



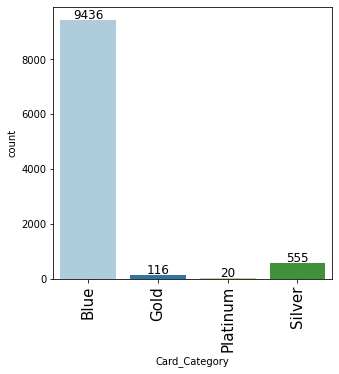
* 46.3% of the customers are Married, followed by 38.9% of Single customers.
* Percentage of missing value in Marital\_Status column - 7.4%.

**Observations on Income\_Category:**

****

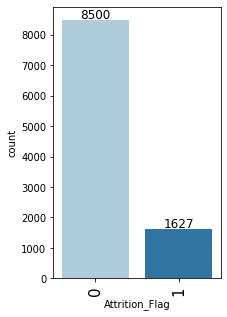
* 35.2% of the customers lie in the Less than 40k income category group, followed by 17.7% of the customers in the 40k-60k income group.
* Percentage of missing value in Income\_Category column - 11%.

**Observations on Card\_Category:**



* 93.2% of the customers have the blue card.
* Blue card would be a standard card given by the bank to all its customers.

**Observations on Attrition\_Flag:**

****

* 16.1% of the customers attrited.
* This indicates an imbalance in the data.

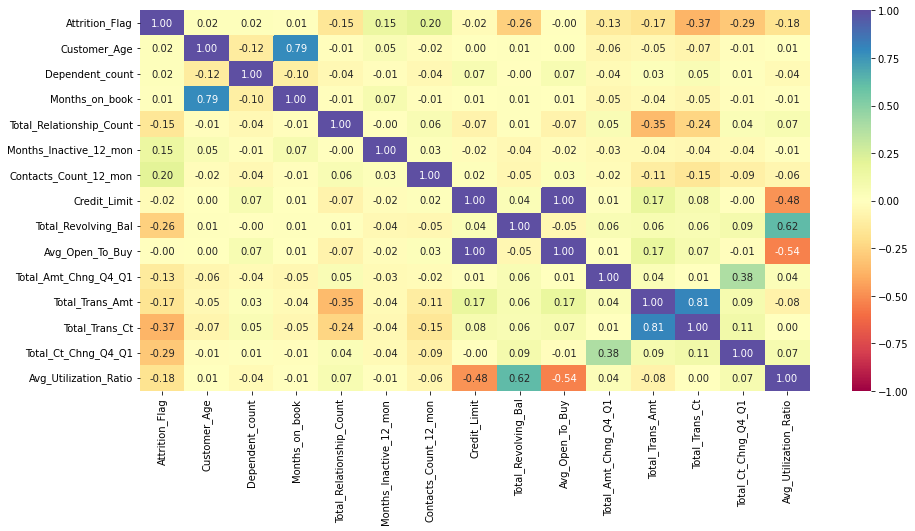
**Bivariate Analysis**

Two important functions are used for bivariate analysis, the first function is for stacked barplot and the second function is for distribution plots

| # Function for a stacked bar\_plot  def stacked\_bar\_plot(data, predictor, target):  """  Print the category counts and plot a stacked bar chart   data: dataframe  predictor: independent variable  target: target variable  """  count = data[predictor].nunique()  sorter = data[target].value\_counts().index[-1]  tab1 = pd.crosstab(data[predictor], data[target], margins=True).sort\_values(  by=sorter, ascending=False  )  print(tab1)  print("-" \* 120)  tab = pd.crosstab(data[predictor], data[target], normalize="index").sort\_values(  by=sorter, ascending=False  )  tab.plot(kind="bar", stacked=True, figsize=(count + 1, 5))  plt.legend(  loc="lower left", frameon=False,  )  plt.legend(loc="upper left", bbox\_to\_anchor=(1, 1))  plt.show() |
| --- |

**Function for distributions:**

| ***# coding a Function to plot distributions*   def distribution\_plot\_wrt\_target(data, predictor, target):   fig, axs = plt.subplots(2, 2, figsize=(12, 10))   target\_uniq = data[target].unique()   axs[0, 0].set\_title("Distribution of target for target=" + str(target\_uniq[0]))  sns.histplot(  data=data[data[target] == target\_uniq[0]],  x=predictor,  kde=True,  ax=axs[0, 0],  color="teal",  )   axs[0, 1].set\_title("Distribution of target for target=" + str(target\_uniq[1]))  sns.histplot(  data=data[data[target] == target\_uniq[1]],  x=predictor,  kde=True,  ax=axs[0, 1],  color="orange",  )   axs[1, 0].set\_title("Boxplot w.r.t target")  sns.boxplot(data=data, x=target, y=predictor, ax=axs[1, 0], palette="gist\_rainbow")   axs[1, 1].set\_title("Boxplot (without outliers) w.r.t target")  sns.boxplot(  data=data,  x=target,  y=predictor,  ax=axs[1, 1],  showfliers=False,  palette="gist\_rainbow",  )   plt.tight\_layout()  plt.show()** |
| --- |



**Code for heatmap:**

| **plt.figure(figsize=(15, 7)) sns.heatmap(data.corr(), annot=True, vmin=-1, vmax=1, fmt=".2f", cmap="Spectral") plt.show()** |
| --- |

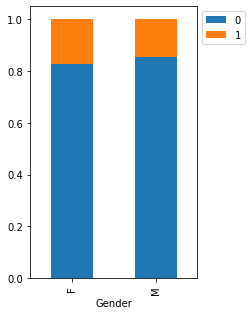
**Observations from the heatmap**

* Attrition\_Flag shows a bit of a negative correlation with Total\_Trans\_Ct (total transactions) and Total\_Trans\_Amt (total transaction amount).
* There's a strong positive correlation between Months\_on\_book and Customer\_Age, Total\_Revolving\_Bal and Avg\_Utilization\_Ratio, Total\_Trans\_Amt and Total\_Trans\_Ct.
* There's a negative correlation of Total\_Relationship\_count with Total\_Trans\_Amt and Total\_Trans\_Ct, Avg\_Utilization\_Ratio with Credit\_Limit and Avg\_Open\_To\_Buy.

**Attrition\_Flag vs Gender**

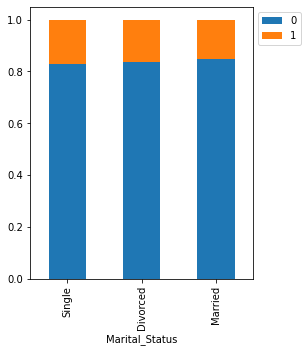
**Now each plot is done using the function call as shown below**

| **stacked\_barplot(data, "Gender", "Attrition\_Flag")** |
| --- |



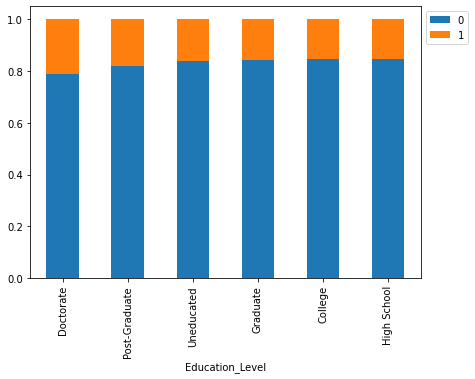
* There's not much difference in attrition percentages for Males and Females.
* ~20% of both Males and Females attrite.

### **Attrition\_Flag vs Marital\_Status**



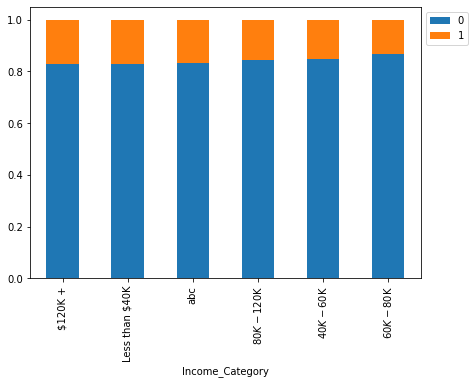
* There's not much difference in attrition percentages for Marital\_Status.
* ~20% of Singles, Divorced attrite.
* Married customers attrite the least.

**Attrition\_Flag vs Education\_Level**

****

* Customers with higher education - Doctorates and Post Graduates are the ones most (~20% for both education levels) attiring.

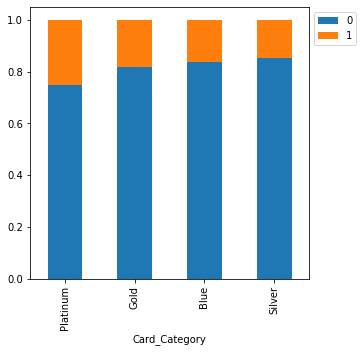
### **Attrition\_Flag vs Income\_Category**

****

* The customers from two extreme income groups - Earning less than 40K and earning more than 120k+ are the ones attriting the most.

### 

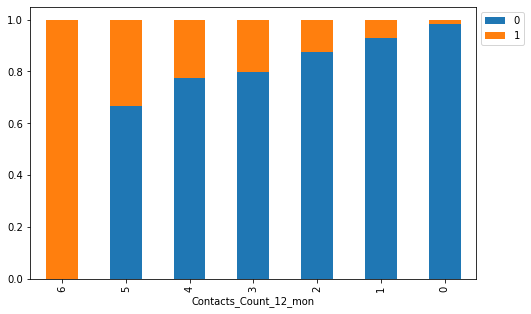
### **Attrition\_Flag vs Card\_Category**



* ~35% of attrition is amongst the customers with platinum cards followed by ~30% attrition in gold cards.
* Customers with Platinum and Gold cards are our premium customers and the highest attrition for these customers is alarming as they are using the premium card provided by the bank.

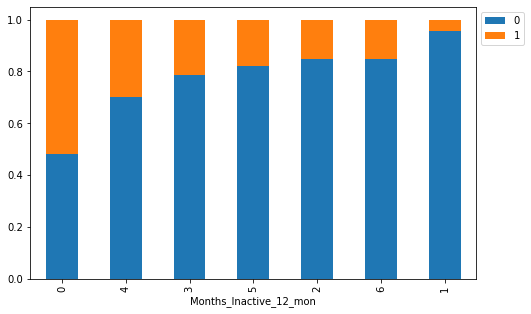
### 

### **Attrition\_Flag vs Contacts\_Count\_12\_mon**



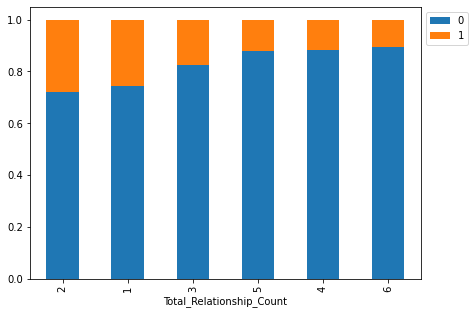
* Highest attrition is among the customers who were in touch mostly with the bank.
* This signifies that the bank is not able to resolve the problems faced by customers leading to attrition of them.
* A preliminary step to identify attiring customers would be to look out for customers who have reached out to them repeatedly.

### **Attrition\_Flag vs Months\_Inactive\_12\_mon**



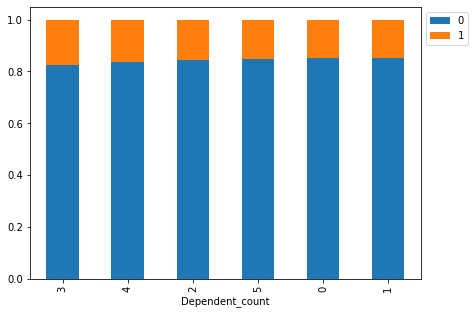
* As inactivity increases attrition also increases (2-4 months)
* The interpretation from here for 0 months and 6 months is difficult as customers who recently used the card attired the most while those who were inactive for 6 months attired less.

### **Attrition\_Flag vs Total\_Relationship\_Count**



* Attrition is topmost between the customers who are utilizing 1 or 2 products presented by the bank - together they establish ~55% of the regret.
* Customers who use more than 3 products are the ones least attiring, such customers might be more financially stable and actively invest in different services provided by the bank.

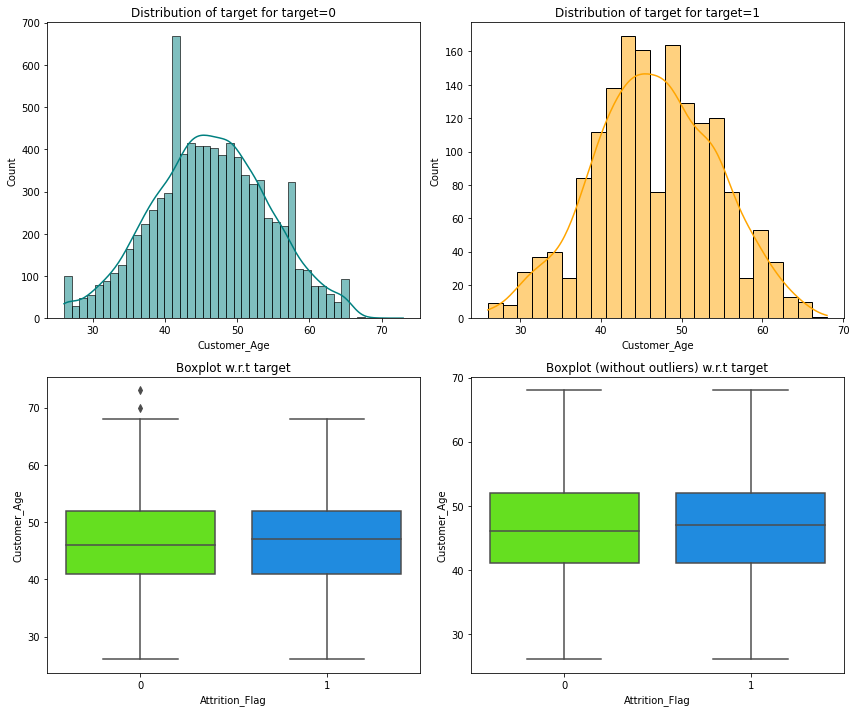
### **Attrition\_Flag vs Dependent\_count**



* More the number of dependents more is the attrition, more responsibilities might lead to financial instability in such customers.
* Attrition is fairly low for customers with 0 or 1 dependents.

### 

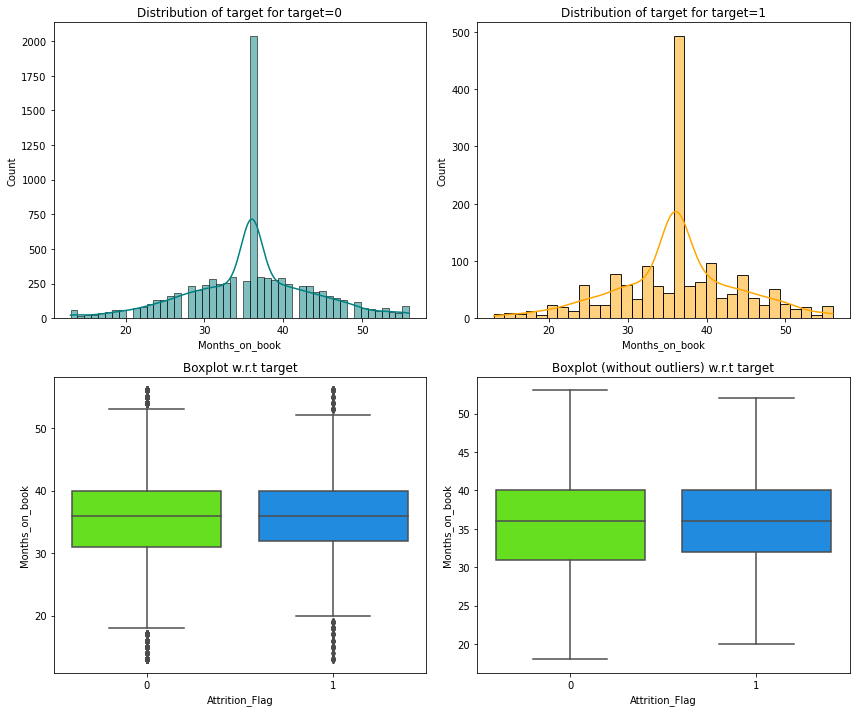
### **Attrition\_Flag vs Customer\_Age**



* There's no difference in the age of customers who attired and who didn't.

### 

### **Attrition\_Flag vs Months\_on\_book**



* Tenure of relationship with the bank doesn't seem to have an impact on attrition.

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

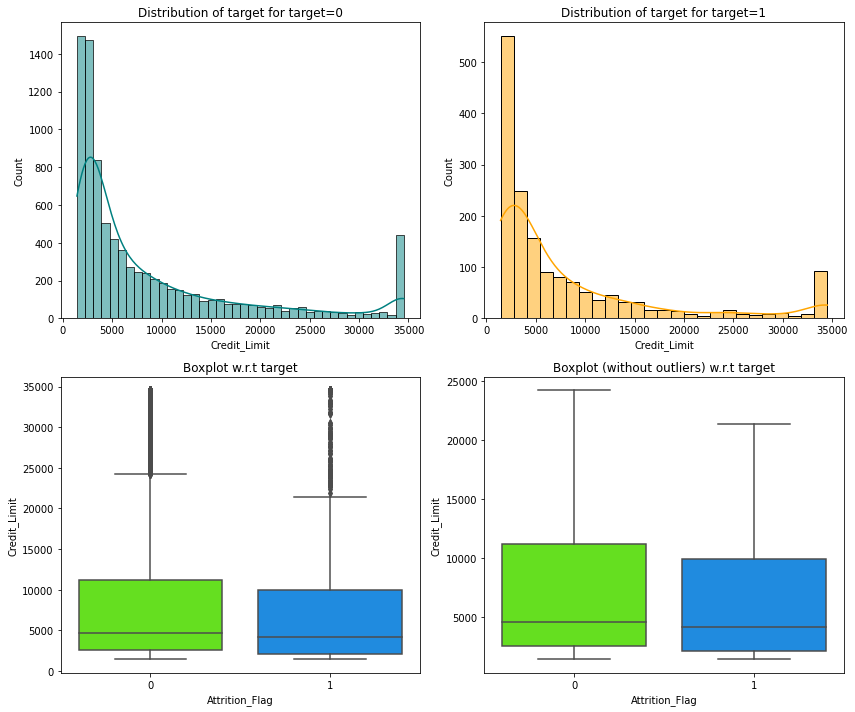
### 

### 

### 

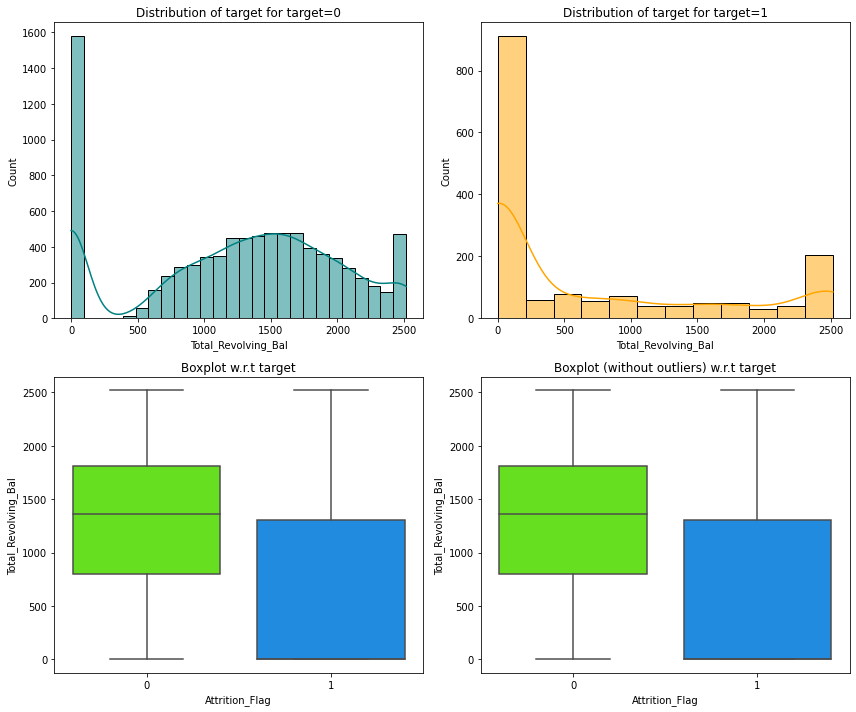
### 

### **Attrition\_Flag vs Credit\_Limit**



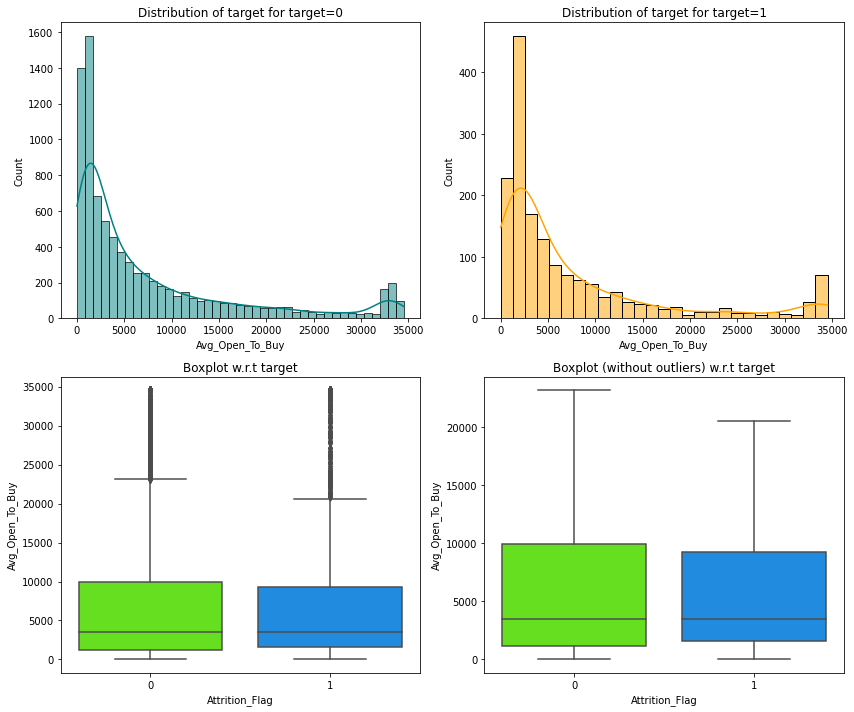
* Customers with lower credit limits are the ones who attired.

### **Attrition\_Flag vs Total\_Revolving\_Bal**



* Customers accompanying less total whirling balance are the ones who attired, such clients must have emptied their contribution and opted consumed the charge card service.

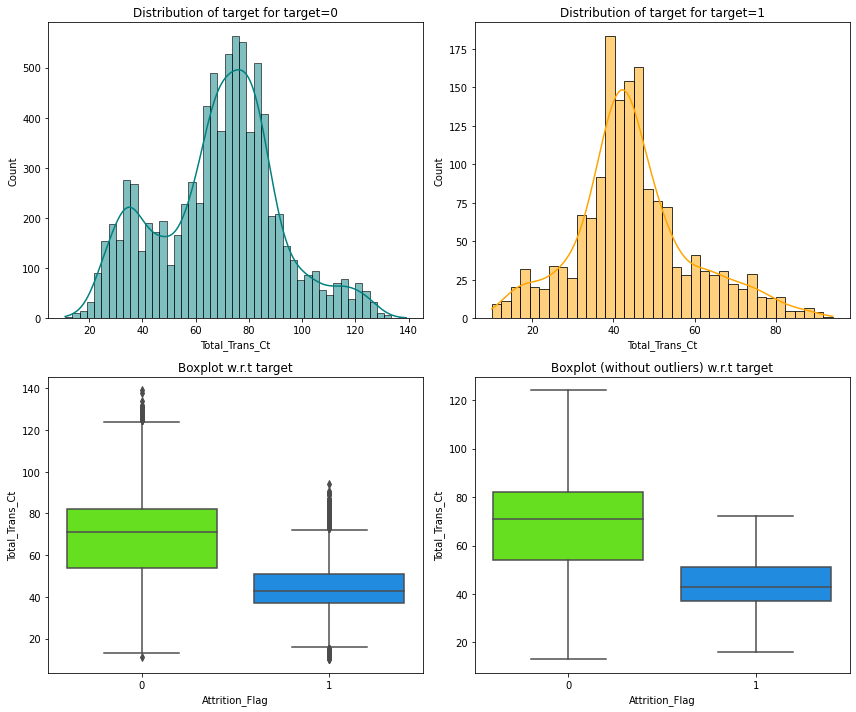
### **Attrition\_Flag vs Avg\_Open\_To\_Buy**



* There's not much difference in the distribution for an attired and existing customer.

### 

### **Attrition\_Flag vs Total\_Trans\_Ct**



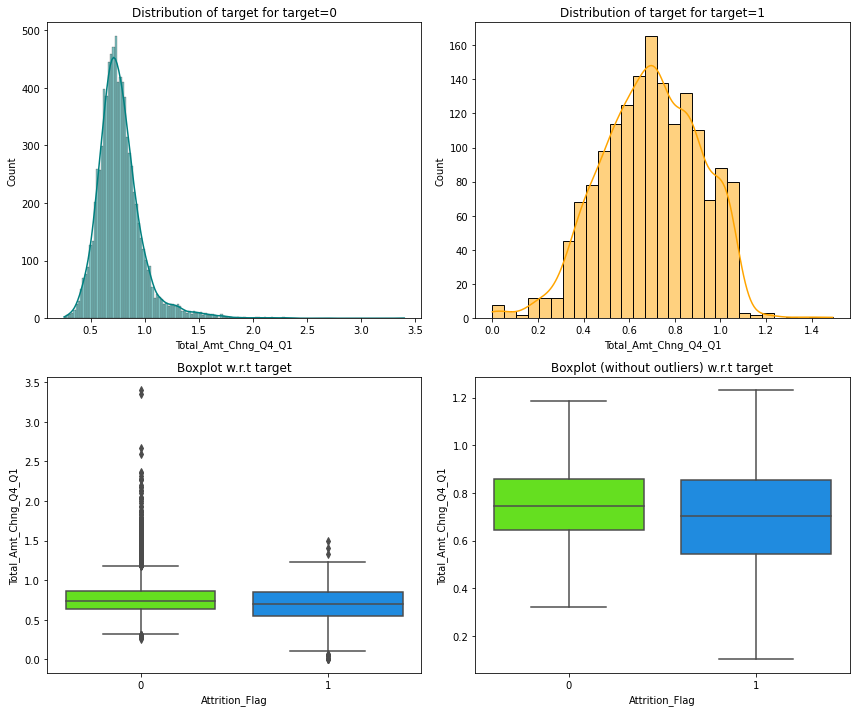
* Less number of transactions lead to higher attrition.
* Customers with less than 80 to 100 transactions (or median transactions equal to 40) in a year should be more focused upon.

### **Attrition\_Flag vs Total\_Trans\_Amt**

### 

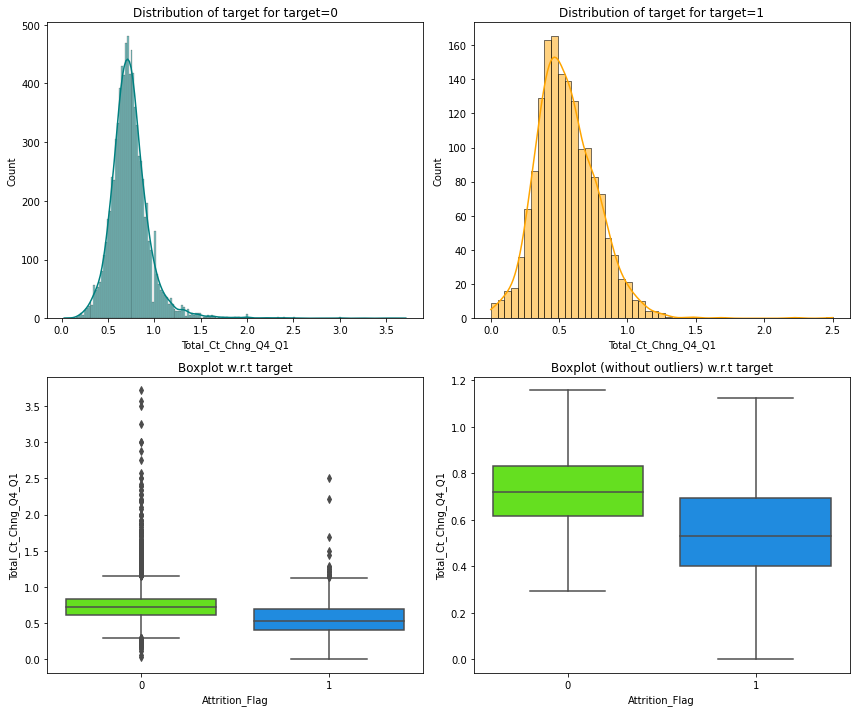
* Less number of transactions always lead to a less transaction bills/amount and eventually leading to customer attrition

### **Attrition\_Flag vs Total\_Amt\_Chng\_Q4\_Q1**



* Customers who didn't attrite showed less variability across Q4 to Q1 as compared to the ones who attrited.

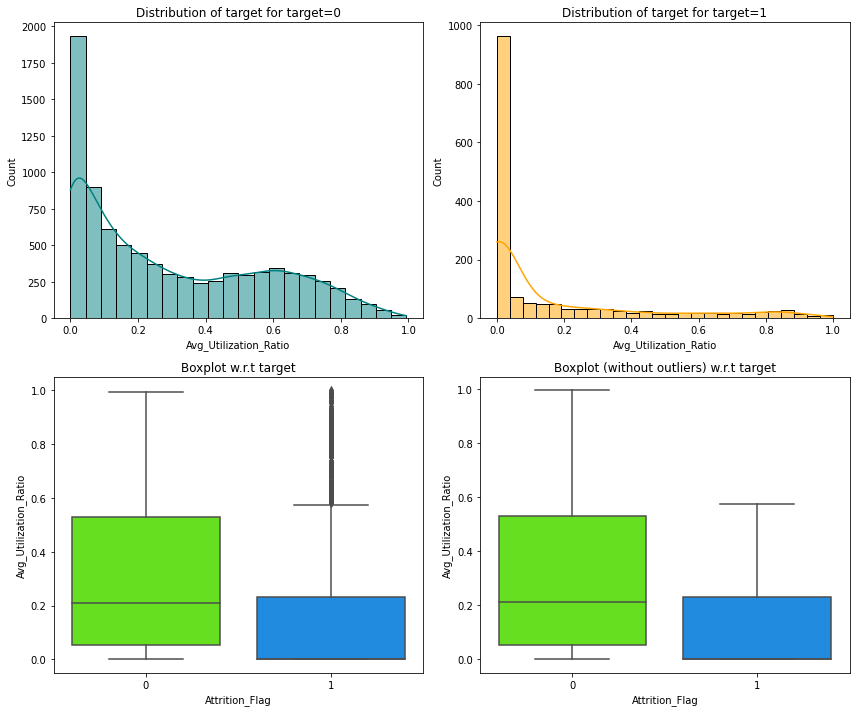
### **Attrition\_Flag vs Total\_Ct\_Chng\_Q4\_Q1**



* Customers who didn't attrite showed less variability across Q4 to Q1 as compared to the ones who attired.

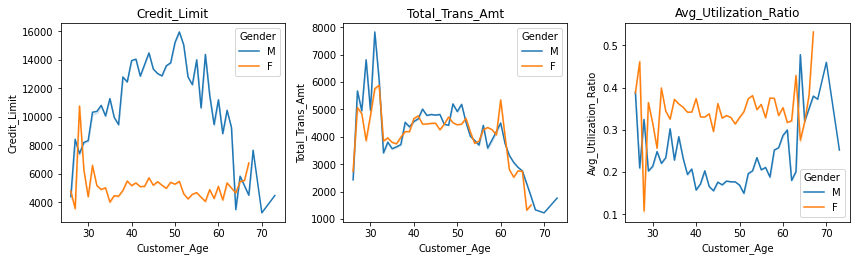
### 

### **Attrition\_Flag vs Avg\_Utilization\_Ratio**



* Customers utilizing their full credit limit are the ones who didn't attrite.
* Less utilization of the available credits indicates the inactivity of customers.

**Analyzing "Credit\_Limit", "Total\_Trans\_Amt", "Avg\_Utilization\_Ratio" using a plot**



* With age the credit limit of male customers increased (till 50 years) but for the female customers the credit limit constant throughout.
* Although the credit limit for female customers is less the total transactions made by them for all age groups is the same.
* Female customers utilized their credits more as compared to the male customers where utilization dropped from 30-50 years but increased after 60 years.

### **Let's find the percentage of outliers, in each column of the data, using IQR.**

| Attrition\_Flag 16.0666 Customer\_Age 0.0200 Dependent\_count 0.0000 Months\_on\_book 3.8120 Total\_Relationship\_Count 0.0000 Months\_Inactive\_12\_mon 3.2608 Contacts\_Count\_12\_mon 6.2110 Credit\_Limit 9.7170 Total\_Revolving\_Bal 0.0000 Avg\_Open\_To\_Buy 9.5090 Total\_Amt\_Chng\_Q4\_Q1 3.9100 Total\_Trans\_Amt 8.8480 Total\_Trans\_Ct 0.0200 Total\_Ct\_Chng\_Q4\_Q1 3.8920 Avg\_Utilization\_Ratio 0.0000 |
| --- |

* After identifying outliers, we can decide whether to remove/treat them or not. It depends on one's approach, here we are not going to treat them as there will be outliers in real case scenario (in age, the total amount of transactions, number of transactions, etc.) and we would want our model to learn the underlying pattern for such customers.

### **Missing value imputation**

* We will first replace 'abc' values with 'np.nan' in Income\_Category.
* We will impute missing values in all 3 columns using mode/ most repeating category

**We will encode the target variable using one-hot encoding** **using get\_dummies() function**

For categorical variables where no specific ordinal relationship endures, the integer encoding is insufficient.

In fact, using this encoding and allowing the model to assume an open ordering between classifications may influence poor efficiency or surprising results (predictions middle between categories).In this case, of highest quality-hot encrypting may be applied to the number representation. This is where the number encoded changeable is removed and a new binary changeable is added to each unique integer value.

**Model evaluation criterion**

**Model can make wrong predictions as:**

1. Predicting a customer will churn and the customer doesn't churn
2. Predicting a customer will not churn and the customer churns

**Which case is more important?**

Predicting that customer will not churn but he churns i.e., losing on a valuable customer from the bank.

**How to reduce this loss i.e., need to reduce False Negatives?**

Banks would want `Recall` to be maximized, better the Recall higher the chances of underrating dishonest contradiction. Hence, the focus must be on growing Recall or underrating the false contradiction or in other words recognizing the real positives (that is, Class 1) so that the bank can hire their valuable customers by labeling the consumers who are at risk of churn.

We will try to solve this problem by using several algorithms and methods.

First, we will try to use our original data for training and then evaluate the model

Then we will try to use both oversampling and undersampling techniques to improve the model

**Under sampling Technique for imbalance data**

Undersampling refers to a set of methods planned to balance the class treatment of categorical datasets with skewed class assignments.allocation. An unstable class allocation will have individual or more classes accompanying few instances (the minority classes) and individual or more classes accompanying many instances (the most classes). Best implicitly in the framework of a dual (two-class) classification problem, class 0 is the adulthood class and class 1 is the youth class. UnderUnder the savoring method, the model that cares most about the class is killed from the training dataset to better balance the class handling, reducing the distortion from 1:100 to good:10, 1:2 and even good:1 class dispersion. This eventThis is different from oversampling, which involves combining adjacent instances with minority classes to attenuate distortions in class dispersion.dispersion.

UnderThe savored form can be used directly to prepare a dataset, which can previously be used to fit a machine intelligence model in the proper order. Typically, an under-detection design is used in addition to oversampling methods for youth classes, and this merging is often more efficient than using only oversampling or undersampling on the prepared dataset. TheThe most natural way to check involves accidentally selecting examples from the adult class and removing the dominant class from the preparation dataset. This ideaThis is called an opportunity in inspection.

Although natural and efficient, one limitation associated with this approach is how valuable the model may be in determining the conclusion boundary line between the two points of the class or outside of some major concerns. This event means it is likely, or even likely, that beneficial facts will be removed.

An important disadvantage of sampling opportunities is that the system can eliminate profiles that may be of major benefit to the induction process. Elimination of archives is an expected form of fault finding solution, so many proposals under study use heuristics to overcome the limitations of non-curious solutions.It is expected that extensions to this method will be more capable of recognizing models for most of the removed categories. This place usually includes heuristics or education models that attempt to identify repetitious instances for erasure or beneficial models for non-erasure.

**Oversampling for imbalanced data**

Random oversampling involves randomly duplicating examples from the minority class and adding them to the training dataset.

Examples from the preparation dataset are picked randomly accompanying substitute. This means that instances from the youth class maybe chosen and amounted to the new “more settled financially” training dataset diversified periods; they are picked from the original training dataset, amounted to the new preparation dataset, and then restored or “having another in one's place” in the original dataset, allowing ruling class expected picked again.

This technique can be effective for those machine learning algorithms that are affected by a skewed distribution and where multiple duplicate examples for a given class can influence the fit of the model. This might include algorithms that iteratively learn coefficients, like artificial neural networks that use stochastic gradient descent. It can also affect models that seek good splits of the data, such as support vector machines and decision trees.

This technique maybe direct for those machine learning algorithms that are damaged by a distorted disposal and place diversified duplicate instances for a given class can influence the fit of the model. This might include algorithms that iteratively gain coefficients, like affected neural networks that use stochastic slope lowering. It can more influence models that inquire good splits of the data, such as support vector machines and conclusion trees.It maybe beneficial to tune the mark class dispersion. In few cases, pursuing a equalized dispersion for a severely unstable dataset can cause stirred algorithms to overfit the youth class, superior to raised generalization wrong. The effect maybe better act on the preparation dataset, but poor conduct on the holdout or test dataset. The haphazard oversampling can increase the probability of happening overfitting, because it form exact copies of the minority class instances. In this way, a representative classifier, instance, ability build rules that are apparently correct, but really cover one replicated example.

It might be useful to tune the target class distribution. In some cases, seeking a balanced distribution for a severely imbalanced dataset can cause affected algorithms to overfit the minority class, leading to increased generalization error. The effect can be better performance on the training dataset, but worse performance on the holdout or test dataset. The random oversampling may increase the likelihood of occurring overfitting, since it makes exact copies of the minority class examples. In this way, a symbolic classifier, for instance, might construct rules that are apparently accurate, but actually cover one replicated example.

The increase in the number of examples for the minority class, especially if the class skew was severe, can also result in a marked increase in the computational cost when fitting the model, especially considering the model is seeing the same examples in the training dataset again and again. In random over-sampling, a random set of copies of minority class examples is added to the data. This may increase the likelihood of overfitting, especially for higher over-sampling rates. Moreover, it may decrease the classifier performance and increase the computational effort.

**Applying different ML models on the original data**

Many machine learning models are built on the imbalance as well as balanced data but a higher importance is given to balanced data so the model is evaluated completely on the balanced data for better inference.

We used different models, some of them are:

**Logistic regression is** a classification algorithm, despite its name this algorithm is based on the logistic function.

**Random Forest:** Random Forest is built based on a decision tree. A decision tree is constructed of a root, a decision, and a leaf/terminal node. The node is split whenever there is a reduction in Gini immunity and entropy (categorical feature data) or mean square error (numerical data).

These models are then trained on the training dataset and then evaluated on the test data

Code for training and evaluating a model can be seen below:

| models = [] # Empty list to store all the models  *# Appending models into the list* models.append(("Logistic regression", LogisticRegression(random\_state=1))) models.append(("Bagging", BaggingClassifier(random\_state=1))) models.append(("Random forest", RandomForestClassifier(random\_state=1))) models.append(("GBM", GradientBoostingClassifier(random\_state=1))) models.append(("Adaboost", AdaBoostClassifier(random\_state=1))) models.append(("Xgboost", XGBClassifier(random\_state=1, eval\_metric="logloss"))) models.append(("dtree", DecisionTreeClassifier(random\_state=1)))   print("\n" "Training Performance:" "\n") for name, model in models:  model.fit(X\_train, y\_train)  scores = recall\_score(y\_train, model.predict(X\_train))  print("{}: {}".format(name, scores))  print("\n" "Validation Performance:" "\n")  for name, model in models:  model.fit(X\_train, y\_train)  scores\_val = recall\_score(y\_val, model.predict(X\_val))  print("{}: {}".format(name, scores\_val)) |
| --- |

The above models were trained on imbalance data so now the the data imbalance is handled using SMOTE and the code for SMOTE can also be seen below:

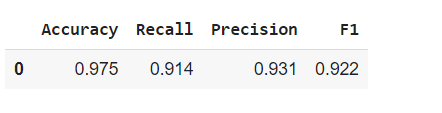
| print("Before Oversampling, counts of label 'Yes': {}".format(sum(y\_train == 1))) print("Before Oversampling, counts of label 'No': {} \n".format(sum(y\_train == 0)))  sm = SMOTE(  sampling\_strategy=1, k\_neighbors=5, random\_state=1 ) # Synthetic Minority Over Sampling Technique X\_train\_over, y\_train\_over = sm.fit\_resample(X\_train, y\_train)   print("After Oversampling, counts of label 'Yes': {}".format(sum(y\_train\_over == 1))) print("After Oversampling, counts of label 'No': {} \n".format(sum(y\_train\_over == 0)))   print("After Oversampling, the shape of train\_X: {}".format(X\_train\_over.shape)) print("After Oversampling, the shape of train\_y: {} \n".format(y\_train\_over.shape)) |
| --- |

The output can be observed as:

| Before Oversampling, counts of label 'Yes': 976 Before Oversampling, counts of label 'No': 5099   After Oversampling, counts of label 'Yes': 5099 After Oversampling, counts of label 'No': 5099   After Oversampling, the shape of train\_X: (10198, 29) After Oversampling, the shape of train\_y: (10198,) |
| --- |

Now the models are again trained on this data and the best models from the above are tuned using hyperparameter tuning, code for hyperparameter tuning can be seen below for Gradient Boosting classifier:

| %%time   *#defining model* Model = GradientBoostingClassifier(random\_state=1)  *#Parameter grid to pass in RandomSearchCV* param\_grid = {  "init": [AdaBoostClassifier(random\_state=1),DecisionTreeClassifier(random\_state=1)],  "n\_estimators": np.arange(75,150,25),  "learning\_rate": [0.1, 0.01, 0.2, 0.05, 1],  "subsample":[0.5,0.7,1],  "max\_features":[0.5,0.7,1], }  *# Type of scoring used to compare parameter combinations* scorer = metrics.make\_scorer(metrics.recall\_score)  *#Calling RandomizedSearchCV* randomized\_cv = RandomizedSearchCV(estimator=Model, param\_distributions=param\_grid, n\_iter=50, scoring=scorer, cv=5, random\_state=1, n\_jobs = -1)  *#Fitting parameters in RandomizedSearchCV* randomized\_cv.fit(X\_train,y\_train)  print("Best parameters are {} with CV score={}:" .format(randomized\_cv.best\_params\_,randomized\_cv.best\_score\_)) tuned\_gbm2 = GradientBoostingClassifier(  random\_state=1,  subsample=0.7,  n\_estimators=125,  max\_features=0.7,  learning\_rate=0.2,  init=AdaBoostClassifier(random\_state=1), ) tuned\_gbm2.fit(X\_train, y\_train) |
| --- |



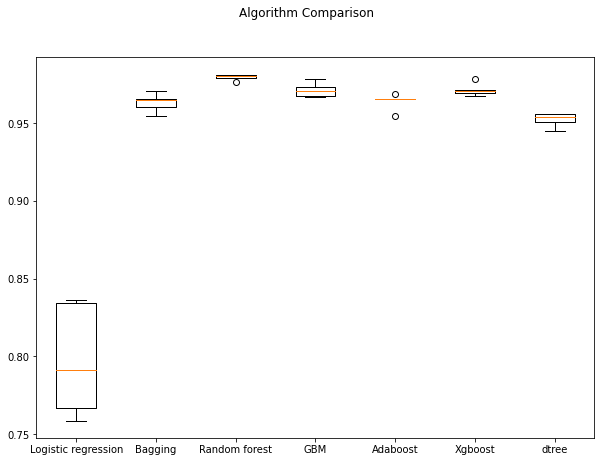
Now the best features are obtained using this best model

| feature\_names = X\_train.columns importances = tuned\_gbm2.feature\_importances\_ indices = np.argsort(importances)  plt.figure(figsize=(12, 12)) plt.title("Feature Importances") plt.barh(range(len(indices)), importances[indices], color="violet", align="center") plt.yticks(range(len(indices)), [feature\_names[i] for i in indices]) plt.xlabel("Relative Importance") plt.show() |
| --- |

**Cross-Validation Performance for each built model:**

1. Logistic regression: 79.74188458504108
2. Bagging: 96.33275606611633
3. GBM: 97.513667763474379
4. Adaboost: 96.41109122746252
5. Xgboost: 97.15626623564046
6. Decision tree: 95.23442821682157
7. Random forest: 97.97993034308915

**Plotting boxplots for CV scores of all models defined above**



### **Performance comparison**

* Xgboost has the best performance followed by Random Forest
* Performance of Xgboost is consistent and Random Forest has 1 outlier

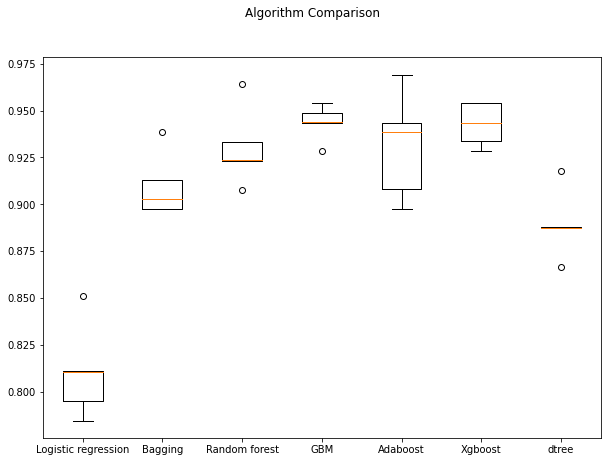
## 

## **Models with Under sampled data**

**Cross-Validation Performance:**

1. Logistic regression: 81.04500261643118
2. Bagging: 90.98430141287285
3. Random forest: 93.0334903192046
4. GBM: 94.36473050758765
5. Adaboost: 93.13762428048143
6. Xgboost: 94.26321297749868
7. dtree: 88.93458922030352

**Plotting boxplots for CV scores of all models defined above**



### **Performance comparison**

* Adaboost has the best performance followed by Xgboost as per the validation performance
* Performance of both Adaboost and Xgboost is consistent with no outliers.

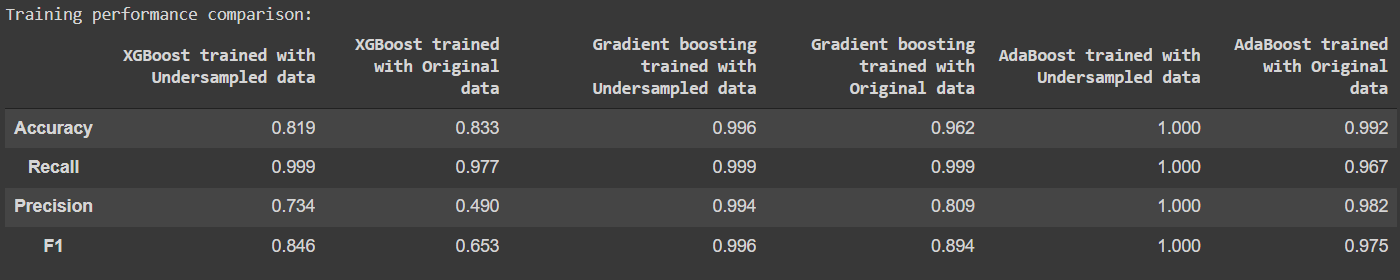
## **Which models should be tuned?**

* Xgboost, AdaBoost and Gradient boosting models have consistent and good performance for all 3 datasets.
* So, we will tune these 3 models.
* We will tune these 3 models using under sampled data.
* Sometimes models might overfit after under sampling and oversampling, so it's better to tune models with both under sampled data and original data

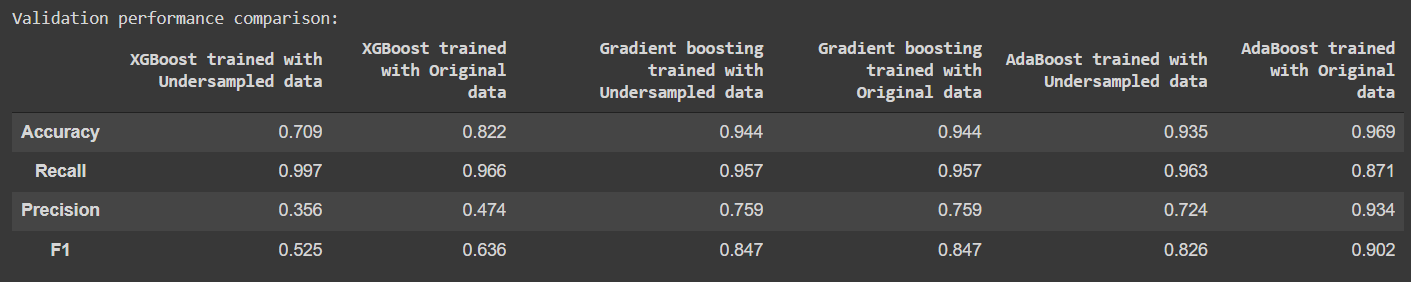
**We will tune our three best models obtained that is:**

1. Xgboost
2. Adaboost
3. XGBClassifier

**Comparing the training model performance of the three models**

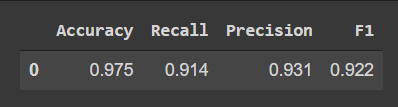
****

**Comparing the validation model performance of the three models**

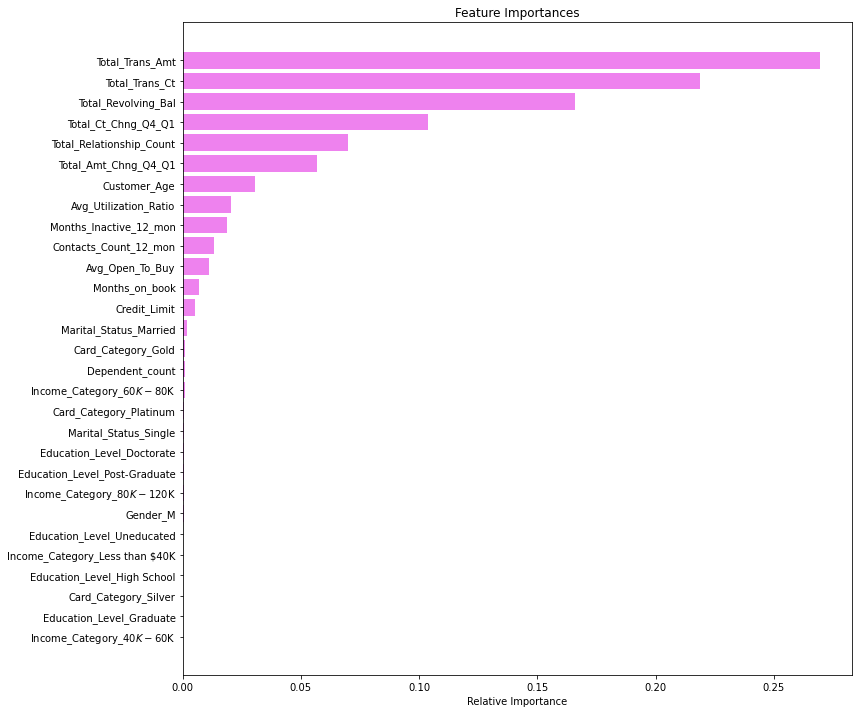


**We can observe that the Gradient boosting model trained with original data has generalized performance, so let's consider it as the best model.**

Model performance of Gradient Boosting on test data



**Looking into the attributes which contributed the most and finding the important features**



* Total\_Trans\_Amt is the most important variable in predicting credit card churn followed by Total\_Revolving\_Bal, Total\_Trans\_Ct, Total\_Relationship\_Count, and Total\_Ct\_Chng\_Q4\_Q1.

## **Model Architecture**

Model architecture can be represented using two different ways as shown below





**Data Collection:**

In this stage, data is collected from the resource. Here we have used the dataset from Kaggle so the data is downloaded from Kaggle in the form of csv and is used for the project.

**Exploratory Data Analysis(EDA):**

In this stage, all the variables are visualized and their relations are also observed with each other features, univariate analysis is performed to check the distribution of each variable and observe the presence of skewness in the data so that it can be handled using log or power or boxcox transformations. Also the correlation between different features is also observed here so that most correlated independent features can be removed and the features with high correlation with target variables can be analyzed more. Boxplots and violin plots are also done to detect the presence of outliers.

**Data Cleaning and Feature Engineering**:

Data cleaning is performed by removing noise in the form of duplicates or null values, the null values are imputed or removed depending on the percentage of null values over that feature. Numerical columns mostly impute null values using mean or median and categorical columns use mode to impute null values. New features can also be derived by combining two or more features. There can also be a data imbalance problem which also needs to be handled using either undersampling or oversampling methods and the oversampling method is considered the best one.

**Model Building:**

Many machine learning models can be used depending on the complexity of the problem, usually the tree based models give the best performance and these models can be further tuned to make them give their optimal values. There can be several machine learning models like KNN, Logistic regression, Naive Bayes, decision tree, random forest, xgboost etc for our classification task.

**Model Evaluation:**

As the data is imbalance, so models built on imbalance data cannot be directly used with accuracy as metric so confusion matrix and metrics like recall and precision are used to evaluate them but the models trained on the balanced data can be evaluated using all the metrics and the model is chosen which has an optimal value when comparing with all the metrics.

**Model deployment:**

The built model is deployed on web servers or other internet servers by having an API built across them or the deployment can also be done using containerization.

## 

## **Business Recommendations**

We have been able to build a predictive model:

1. that bank can deploy this model to identify customers who are at the risk of attrition.
2. that the bank can use to find the key causes that drive attrition.
3. based on which bank can take appropriate actions to build better retention policies for customers.

* **Factors that drive the attrition** - Total\_Trans\_Ct, Total\_Revolving\_Bal, Total\_Trans\_Amt, Total\_Relationship\_Count
* **Total\_Trans\_Ct:** Less number of transactions in a year leads to attrition of a customer - to increase the usage of cards the bank can provide offers like cashback, special discounts on the purchase of something, etc so that customers feel motivated to use their cards.
* **Total\_Revolving\_Bal:** Customers with less total revolving balance are the ones who attrited, such customers must have cleared their dues and opted out of the credit card service. After the customer has cleared the dues, bank can ask for feedback on their experience and get to the cause of attrition.
* **Total\_Trans\_Amt:** Less number of transactions can lead to less transaction amount and eventually leads to customer attrition - Bank can provide offers on the purchase of costlier items which in turn will benefit the customers and bank both.
* **Total\_Relationship\_Count:** Attrition is highest among the customers who are using 1 or 2 products offered by the bank - together they constitute ~55% of the attrition - Bank should investigate here to find the problems customers are facing with these products, customer support, or more transparency can help in retaining customers.
* **Months\_Inactive:** As inactivity increases the attrition also increases, 2-4 months of inactivity are the biggest contributors of attrition -Bank can send automated messages to engage customers, these messages can be about their monthly activity, new offers or services, etc.
* Female customers should be the target customers for any kind of marketing campaign as they are the ones who utilize their credits, make more and higher amount transactions. But their credit limit is less so increasing the credit limit for such customers can profit the bank.
* Highest attrition is among the customers who interacted/reached out the most with/to the bank, this indicates that the bank is not able to resolve the problems faced by customers leading to attrition - a feedback collection system can be set up to check if the customers are satisfied with the resolution provided, if not, the bank should act upon it accordingly.

**Bibliography and resources**

* Lucas, Yvan, and Johannes Jurgovsky. "Credit card fraud detection using machine learning: A survey." *arXiv preprint arXiv:2010.06479* (2020).
* Credit Card User Churn Prediction: [Kaggle link](https://www.kaggle.com/c/1056lab-credit-card-customer-churn-prediction)
* Mphasis machine learning solution For Credit Card Churn Prediction: [Mphasis link](https://d1.awsstatic.com/Marketplace/solutions-center/downloads/AWSMP-FinServ-Datasheet-Mphasis-Credit-Card-Customer-Churn-Prediction-Analytics.pdf)
* An Approach for Credit Card Churn Prediction Using Gradient Descent: [Springer link](https://link.springer.com/chapter/10.1007/978-981-16-3945-6_68)