MODEL TUNING

CREDIT CARD CUSTOMER CHURN

NAEEM SUFI

BACKGROUND AND CONTEXT

- THE 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 CARDS SERVICES WOULD LEAD BANK TO LOSS, SO THE BANK WANTS TO ANALYZE THE DATA OF CUSTOMERS AND IDENTIFY THE CUSTOMERS WHO WILL LEAVE THEIR CREDIT CARD SERVICES AND REASON FOR SAME – SO THAT 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
- YOU NEED TO IDENTIFY THE BEST POSSIBLE MODEL THAT WILL GIVE THE REQUIRED PERFORMANCE

OBJECTIVE & ROADMAP

- TO DEFINE THE SUCCESS OF THE SOLUTION THAT WE WILL DELIVER LET'S DEFINE THE METRICS AS: F1 SCORE, PRECISION AND RECALL. THIS METRICS WERE CHOSEN SINCE NORMALLY CHURN PROBLEMS ARE IMBALANCED, BUT ALL DEPENDS ON THE DEFINITION OF CHURN AND THE COST DRIVEN BY EACH SCENARIO.
- THE FIRST GOAL OF THIS PROJECT IS TO PROVIDE AN ANALYSIS WHICH SHOWS THE **DIFFERENCE** BETWEEN A **NON-CHURNING AND CHURNING CUSTOMER**. THIS WILL PROVIDE US INSIGHT INTO WHICH CUSTOMERS ARE EAGER TO CHURN.
- THE TOP PRIORITY OF THIS CASE IS TO IDENTIFY IF A CUSTOMER WILL CHURN OR WON'T. IT'S IMPORTANT THAT WE DON'T **PREDICT** CHURNING AS NON-CHURNING CUSTOMERS. THAT'S WHY THE MODEL NEEDS TO BE EVALUATED ON THE **"RECALL"**- METRIC (GOAL > 62%).

- **OBJECTIVES**
- EXPLORE AND VISUALIZE THE DATASET.
- BUILD A CLASSIFICATION MODEL TO PREDICT IF THE CUSTOMER IS GOING TO CHURN OR NOT
- OPTIMIZE THE MODEL USING APPROPRIATE TECHNIQUES
- GENERATE A SET OF INSIGHTS AND RECOMMENDATIONS THAT WILL HELP THE BANK

PROCESS TO ACHIEVE THE OBJECTIVES OF THE PROJECT?

- WE SELECT 7 MODELS FOR OUR CLASSIFICATION PROBLEM BECAUSE IN OUR DATASET TARGET VARIABLE IS IN CATEGORICAL FORMAT SO, WHEN CLASS LABEL IS IN CATEGORICAL THEN THIS PROBLEM IS RELATED TO CLASSIFICATION. WE WILL SELECT THE FOLLOWING MODEL FOR PREDICTION
- DECISION TREE
- LOGISTIC REGRESSION
- BAGGING CLASSIFIER
- RANDOM FOREST CLASSIFIER
- ADA BOOST CLASSIFIER
- GRADIENT BOOSTING CLASSIFIER
- XGBOOST CLASSIFIER

METRICS TO MEASURE PERFORMANCE OF EACH MODEL

- **ACCURACY**

MEASURE TO EVALUATE HOW ACCURATE MODEL'S PERFORMANCE IS:

$$\frac{TP + TN}{TP + FP + FN + FP}$$

- **PRECISION**

MEASURE TO EVALUATE HOW ACCURATE MODEL'S PERFORMANCE IS:

$$\frac{TP}{TP + FP}$$

- **RECALL**

MEASURE TO EVALUATE HOW ACCURATE MODEL'S PERFORMANCE IS:

$$\frac{TP}{TP + FN}$$

- **F_1**

PROVIDES INFORMATION OF BOTH SIDES TN AND TP

$$2 * \frac{Precision * Recall}{Precision + Recall}$$

- $where\ TP = True\ Positive$
- $FP = False\ Positive$
- $TN = True\ Negative$
- $FN = False\ Negetive$
- **CONFUSION MATRIX**

EXPLORATORY DATA ANALYSIS(EDA)

- WE HAVE A DATA SET OF BANK CHURNERS. THIS IS NOT A HUGE DATASET SO, WE WILL EXPLORE THE DATASET TO KNOW MORE ABOUT THE DATA SHAPE, DESCRIPTIVE ANALYSES, STATISTICAL ANALYSIS, UNIVARIATE ,BIVARIATE ANALYSIS, CORRELATION, MISSING VALUES, DTYPES, COLUMN NAMES ETC.
- LET'S TRY TO LOOK THE SHAPE OF DATA:

```
# lets try to check the shape of data
data.shape
```

```
(10127, 20)
```

• DATA DESCRIPTION

```
Index(['Attrition_Flag', 'Customer_Age', 'Gender', 'Dependent_count',
       'Education_Level', 'Marital_Status', 'Income_Category', 'Card_Category',
       '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', 'Avg_Utilization_Ratio'],
      dtype='object')
```

Exploring column names is an important aspect of EDA. We can see that columns are not null. The data types of all columns are int, float and object data type. By closely observing the data and description given about each column attribute we can say that:

- Numeric data columns are **Customer Age, Dependent_count and Months_on_book, Total_Relationship_count, Credit_Limit,Total_Revolving_Bal, Total Trans_Amt and Total_Trans_Ct.**
- Categorical columns are **Attrition_Flag, Gender, Education_Level, Matrial_Status, Income_Category, Card_Category.**
- Floating columns are **Credit_Limit,Avg_Open_Buy,Total_Amt_Chung_Q4_Q1, Total_Amt_Chung_CT_Q, Avg_Utilization_Ratio**

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10127 entries, 768805383 to 714337233
Data columns (total 20 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Attrition_Flag            10127 non-null  object
 1   Customer_Age              10127 non-null  int64
 2   Gender                    10127 non-null  object
 3   Dependent_count           10127 non-null  int64
 4   Education_Level           8608 non-null   object
 5   Marital_Status            9378 non-null   object
 6   Income_Category           10127 non-null  object
 7   Card_Category             10127 non-null  object
 8   Months_on_book            10127 non-null  int64
 9   Total_Relationship_Count  10127 non-null  int64
 10  Months_Inactive_12_mon    10127 non-null  int64
 11  Contacts_Count_12_mon     10127 non-null  int64
 12  Credit_Limit              10127 non-null  float64
 13  Total_Revolving_Bal       10127 non-null  int64
 14  Avg_Open_To_Buy           10127 non-null  float64
 15  Total_Amt_Chng_Q4_Q1      10127 non-null  float64
 16  Total_Trans_Amt           10127 non-null  int64
 17  Total_Trans_Ct            10127 non-null  int64
 18  Total_Ct_Chng_Q4_Q1       10127 non-null  float64
 19  Avg_Utilization_Ratio     10127 non-null  float64
dtypes: float64(5), int64(9), object(6)
memory usage: 1.6+ MB
```

From this we can see that there are No missing values in every columns. So, we don't need to handle these missing values per the case we have.

Attrition_Flag	0
Customer_Age	0
Gender	0
Dependent_count	0
Education_Level	1519
Marital_Status	749
Income_Category	0
Card_Category	0
Months_on_book	0
Total_Relationship_Count	0
Months_Inactive_12_mon	0
Contacts_Count_12_mon	0
Credit_Limit	0
Total_Revolving_Bal	0
Avg_Open_To_Buy	0
Total_Amt_Chng_Q4_Q1	0
Total_Trans_Amt	0
Total_Trans_Ct	0
Total_Ct_Chng_Q4_Q1	0
Avg_Utilization_Ratio	0

dtype: int64

DESCRIPTIVE ANALYSIS

- THE EXPLORATORY DATA ANALYSIS IS MORE IMPORTANT METHOD WHICH DESCRIBE METHOD SHOWS BASIC STATISTICAL CHARACTERISTICS OF EACH NUMERICAL FEATURE (INT64 AND FLOAT64 TYPES):
- NUMBER OF NON-MISSING VALUES, MEAN, STANDARD DEVIATION, RANGE, MEDIAN, 0.25 AND 0.
- WE CAN SEE THE MIN, MAX, MEAN AND STANDARD DEVIATION FOR ALL KEY ATTRIBUTES OF THE DATASE75 QUARTILES.

	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category	Months_on_book	Tota
count	10127.000000	10127.000000	10127.000000	10127.000000	10127.000000	10127.000000	10127.000000	10127.000000	10127.000000	
mean	0.160660	46.325960	0.529081	2.346203	3.096574	1.463415	2.863928	0.179816	35.928409	
std	0.367235	8.016814	0.499178	1.298908	1.834812	0.737808	1.504700	0.693039	7.986416	
min	0.000000	26.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	13.000000	
25%	0.000000	41.000000	0.000000	1.000000	2.000000	1.000000	2.000000	0.000000	31.000000	
50%	0.000000	46.000000	1.000000	2.000000	3.000000	1.000000	3.000000	0.000000	36.000000	
75%	0.000000	52.000000	1.000000	3.000000	5.000000	2.000000	4.000000	0.000000	40.000000	
max	1.000000	73.000000	1.000000	5.000000	6.000000	3.000000	5.000000	3.000000	56.000000	

DESCRIPTIVE ANALYSIS ON NON-NUMERIC

- PREVIOUSLY WE SAW THAT DESCRIPTIVE ANALYSIS ON INT OR FLOAT TYPE BUT NOT IN NUMERIC.
- IN ORDER TO SEE STATISTICS ON NON-NUMERICAL FEATURES, ONE MUST EXPLICITLY INDICATE DATA TYPES OF INTEREST.

	Attrition_Flag	Gender	Education_Level	Marital_Status	Income_Category	Card_Category
count	10127	10127	8608	9378	10127	10127
unique	2	2	6	3	6	4
top	Existing Customer	F	Graduate	Married	Less than $40K	Blue
freq	8500	5358	3128	4687	3561	9436

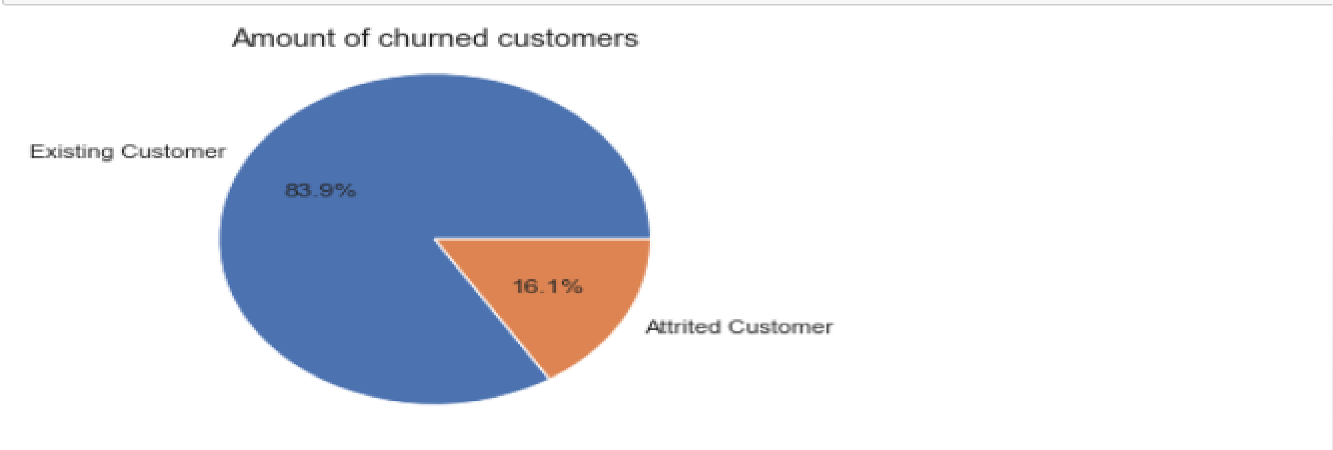
- THIS GIVE US COUNT OF VALUES, UNIQUE VALUES, TOP AND FREQUENCY OF VALUES

DATA VISUALIZATION(UNIVARIATE)

In this section, we will visualize all the columns and check the distribution and relationship with other columns and, we will get more analysis from the graph.

Univariate stands for "one," meaning that, there is just one sort of variable in the data. Univariate analysis' main purpose is to characterize the data. The information will be collected, analyzed, and a pattern will be identified

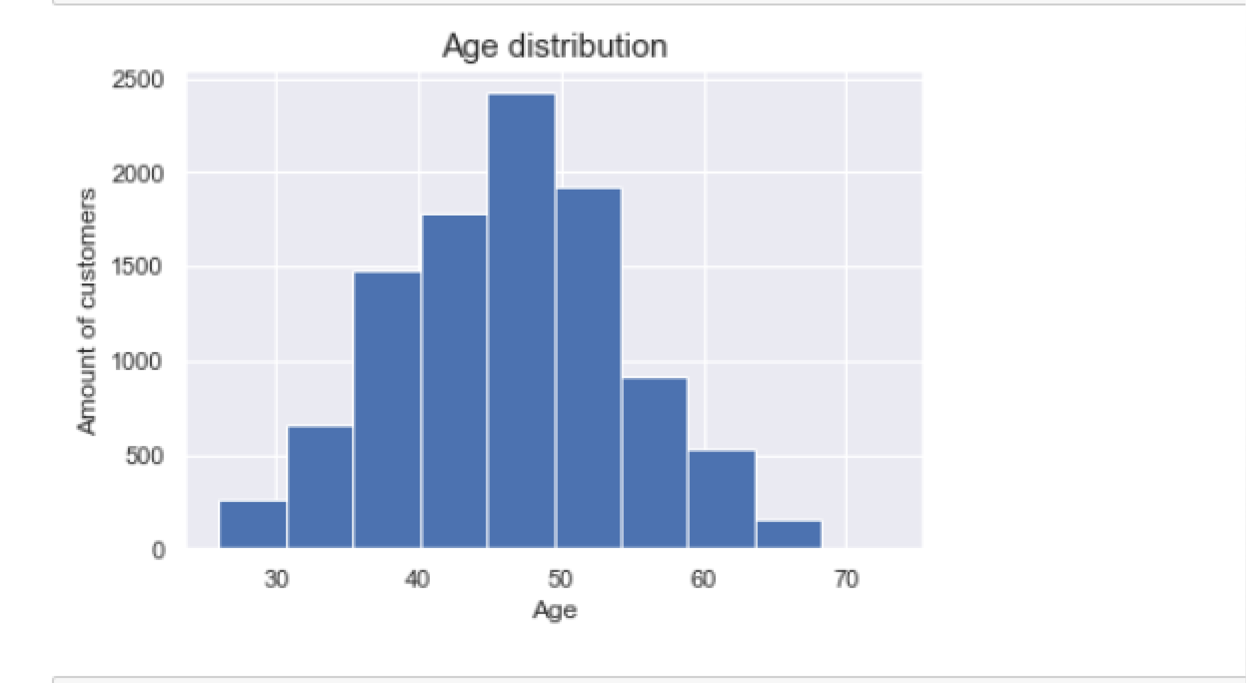
- Now let us check the target variable amount of churned customers
- How many customers have churned?.



It's clear that most of the customers (83.9 %) stays. Since "attrited" or "churned" label is less then 20% of the total all customers. We can say that we have an imbalanced Data. Upsampling will be required to receive a better results.

- Internal event (customer activity) variable - if the account is closed
- Then "attrited customer" else "existing customer“
- Here is 16 person are those people who closed their account and remaning re those are existing customers

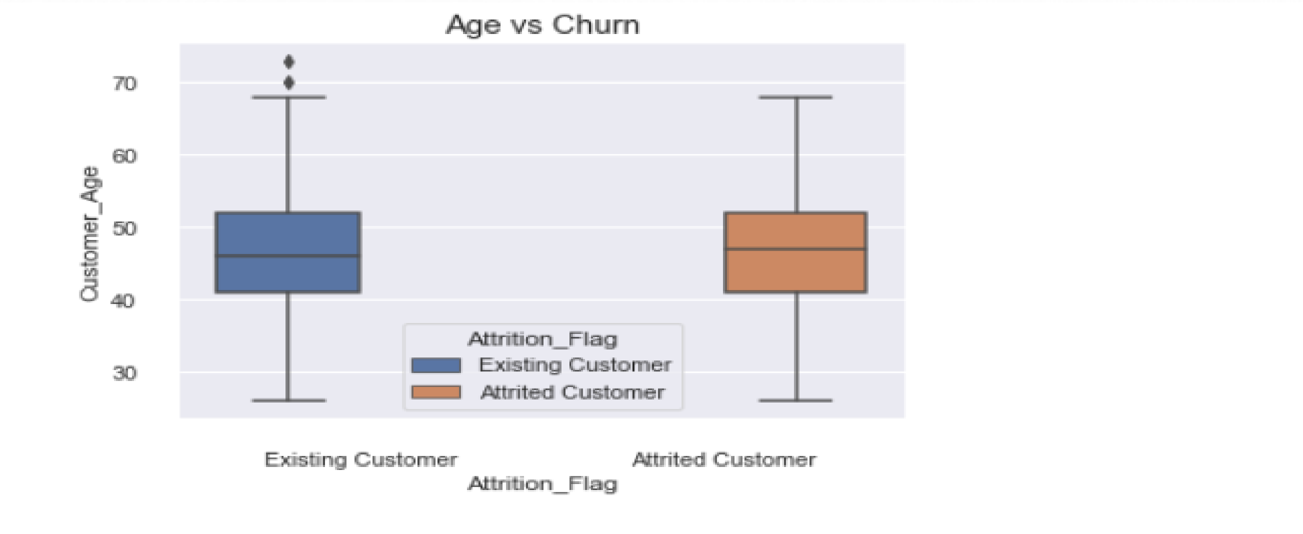
Let’s now try to look that checking the overall distribution between age and amount of customers



- High number of customers are between 45 to 55 and then under 30 and lowest under 60.

Comparing the age distribution vs the target

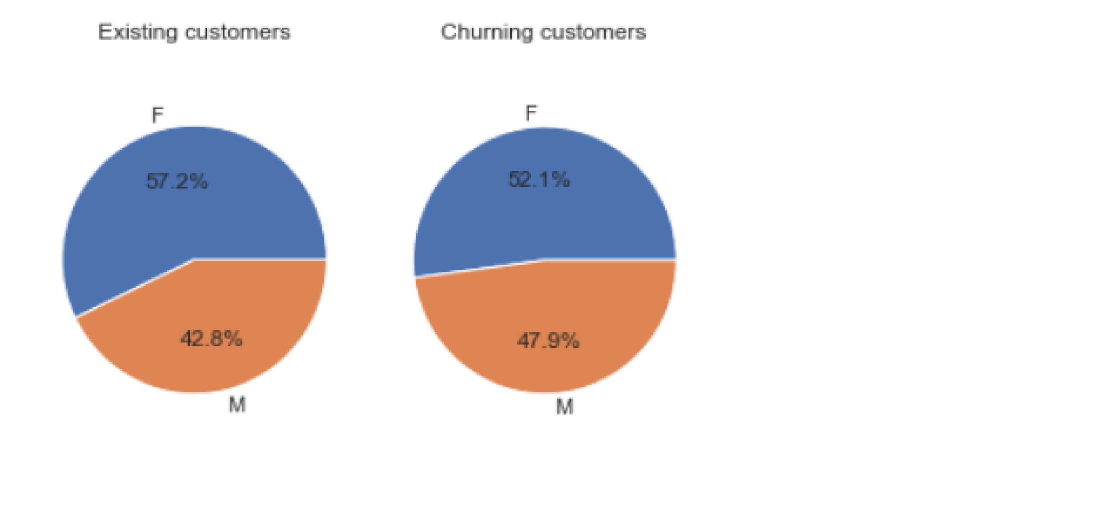
The age is normally distributed. As we seen there is no clear difference between the age distribution.



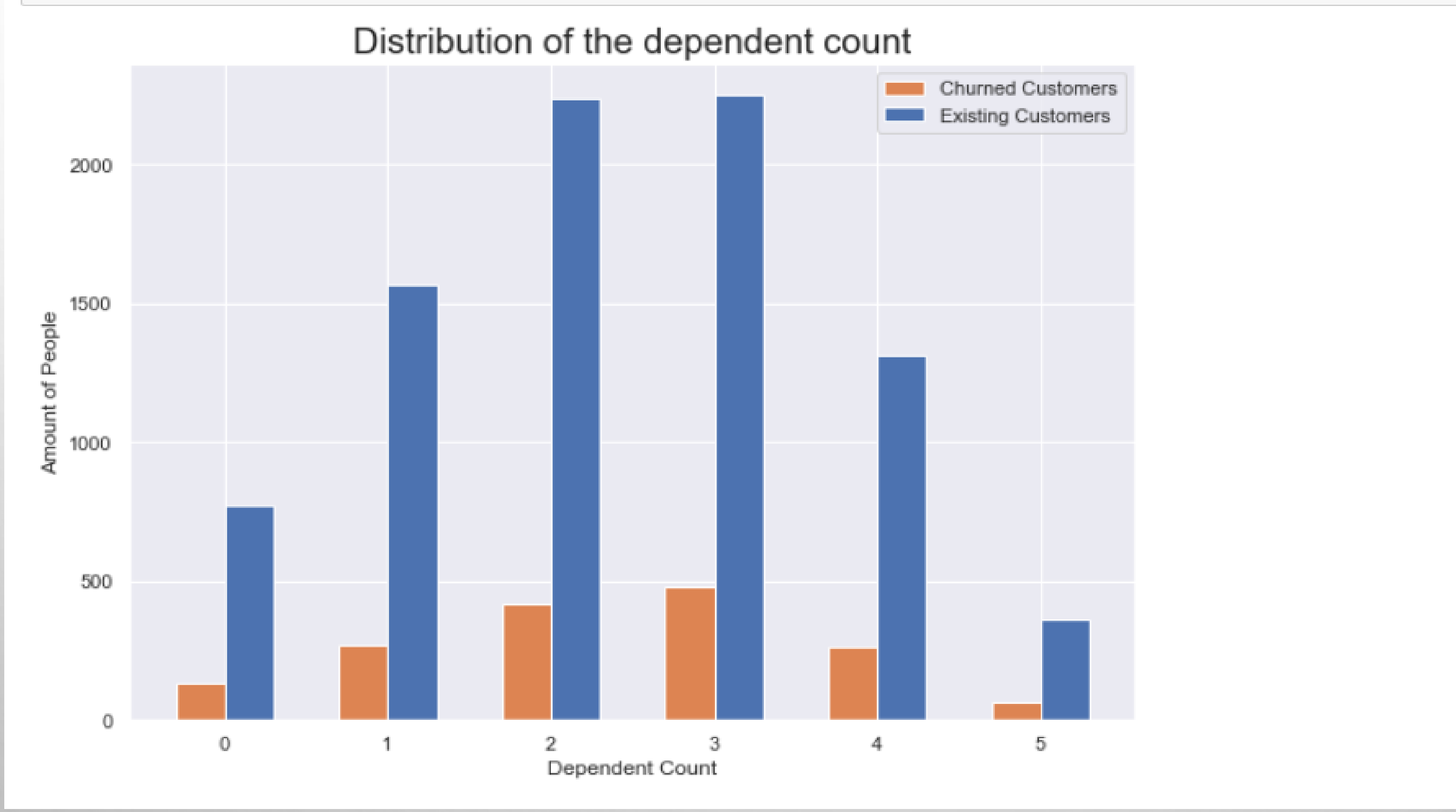
Gender vs churn

Are males of females more eager to churn?

The difference is too small to say that one gender is more eager to churn.

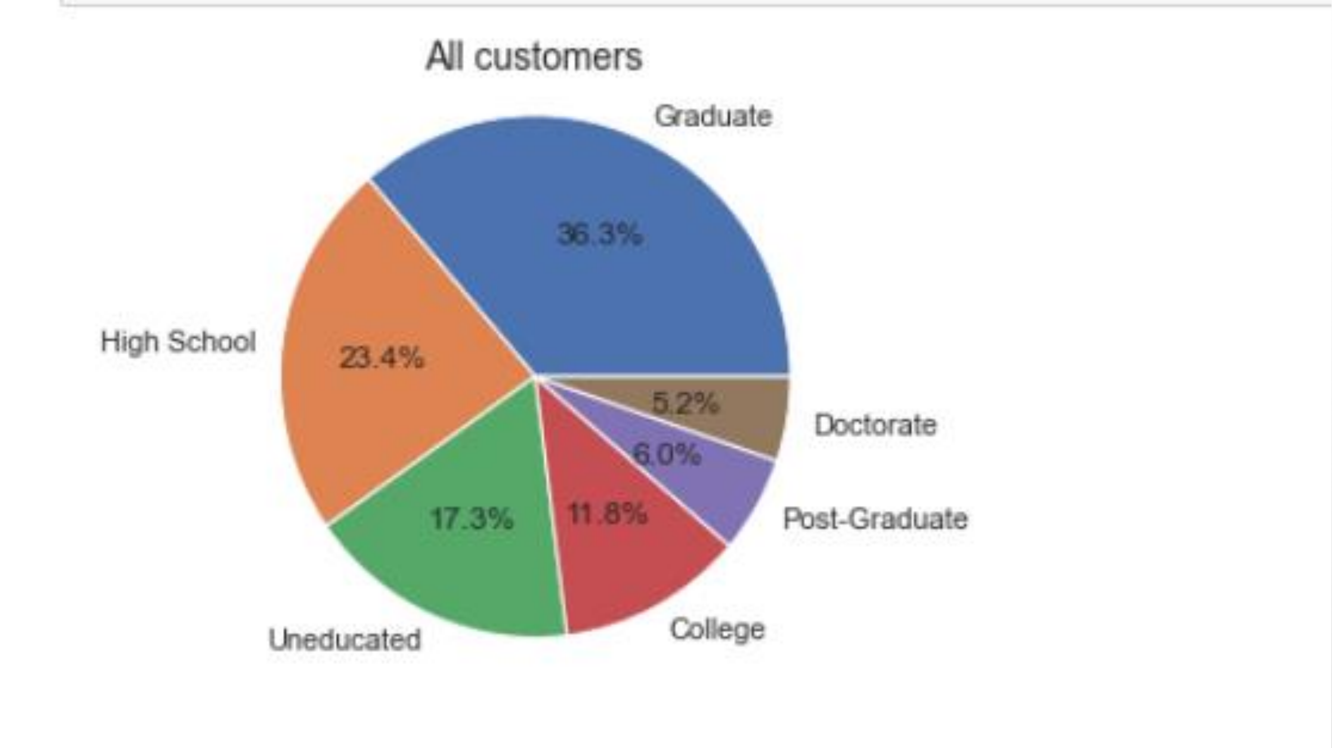


Number of dependents vs churn

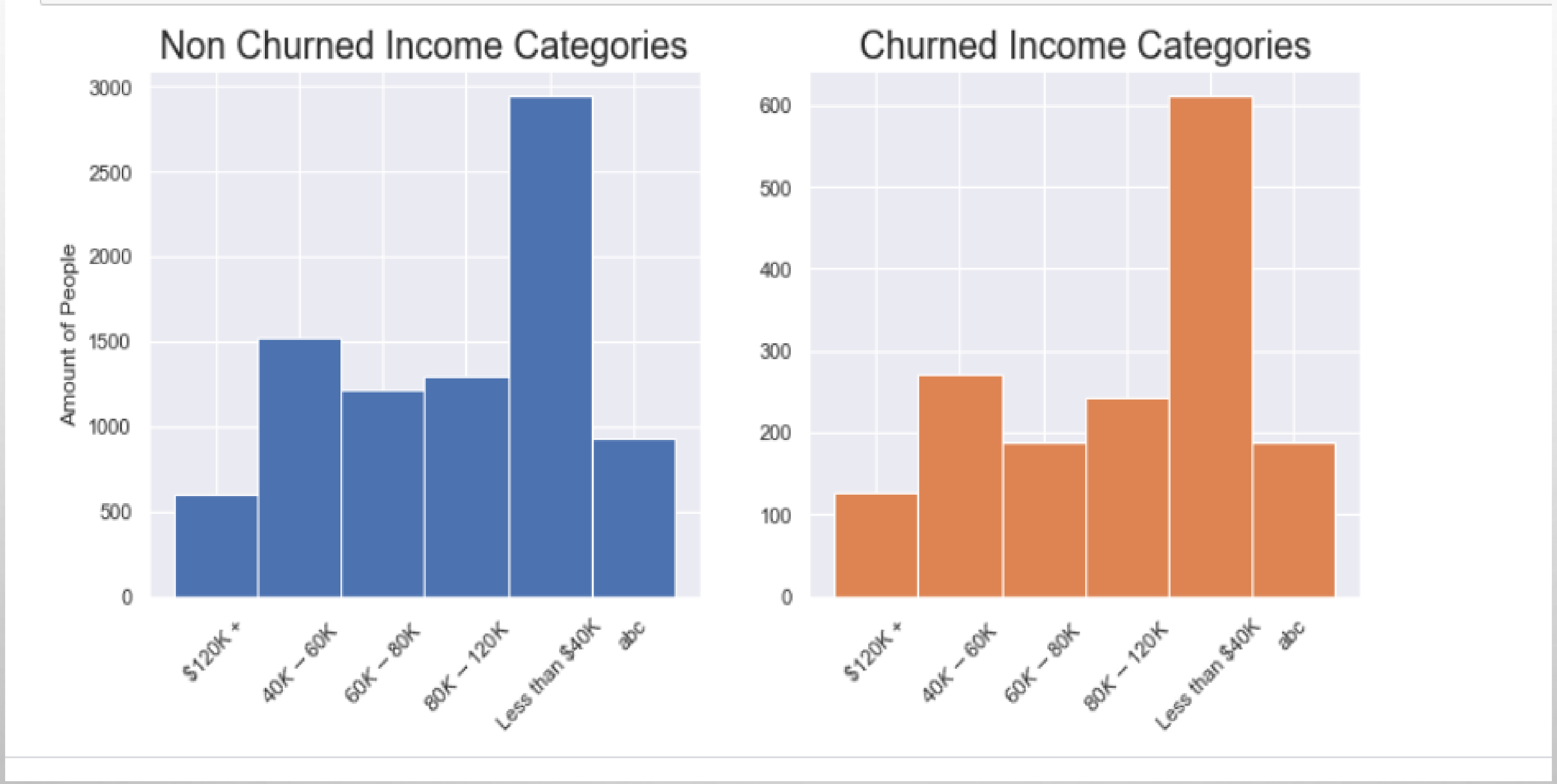


The dependent count shows us a normal distribution. No clear shift is visible when comparing the churned- and non churned distribution.

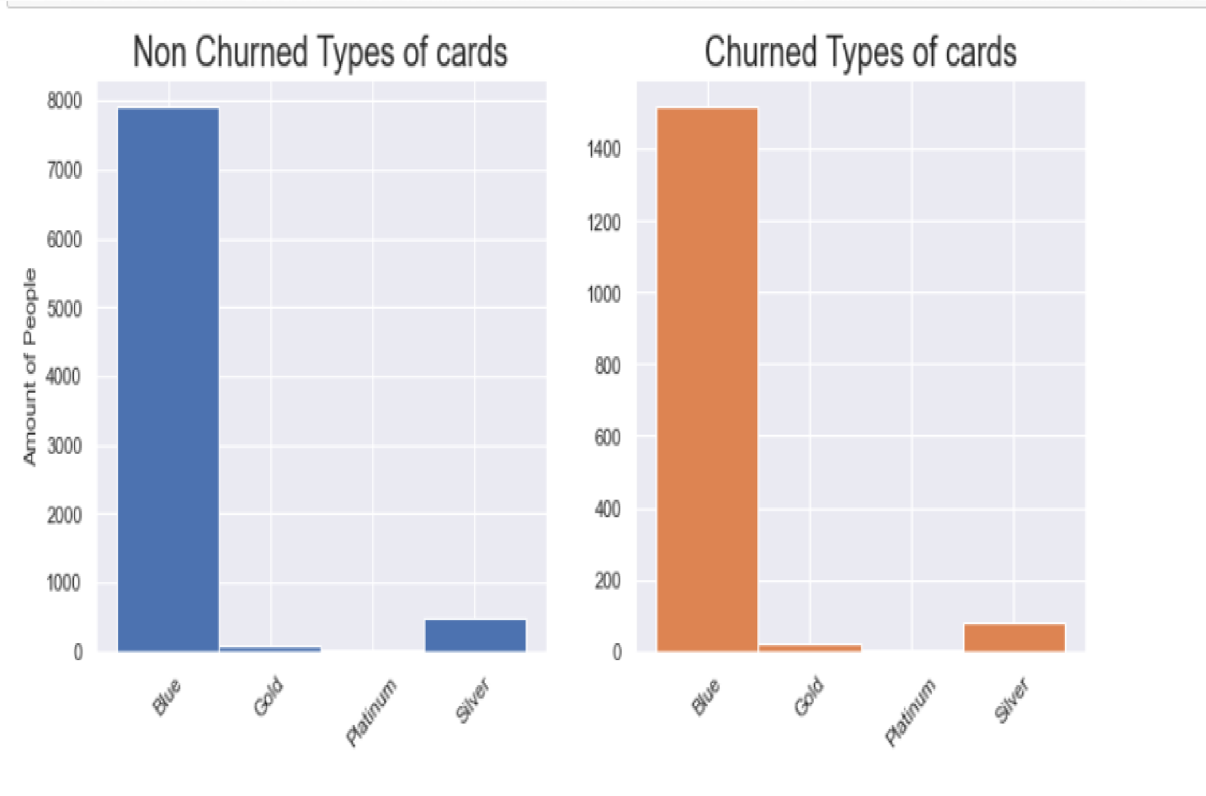
Most people have a graduate education level followed by high school. 15% of the population has an unknown education level.



- We notice that the largest number of customers earn less then $40K per year. Like the other demographic variables, no clear shift in the distributions can be noticed.



- We can clearly see that most of the customers have the "blue" card. The distribution of churned/not churned is the same.

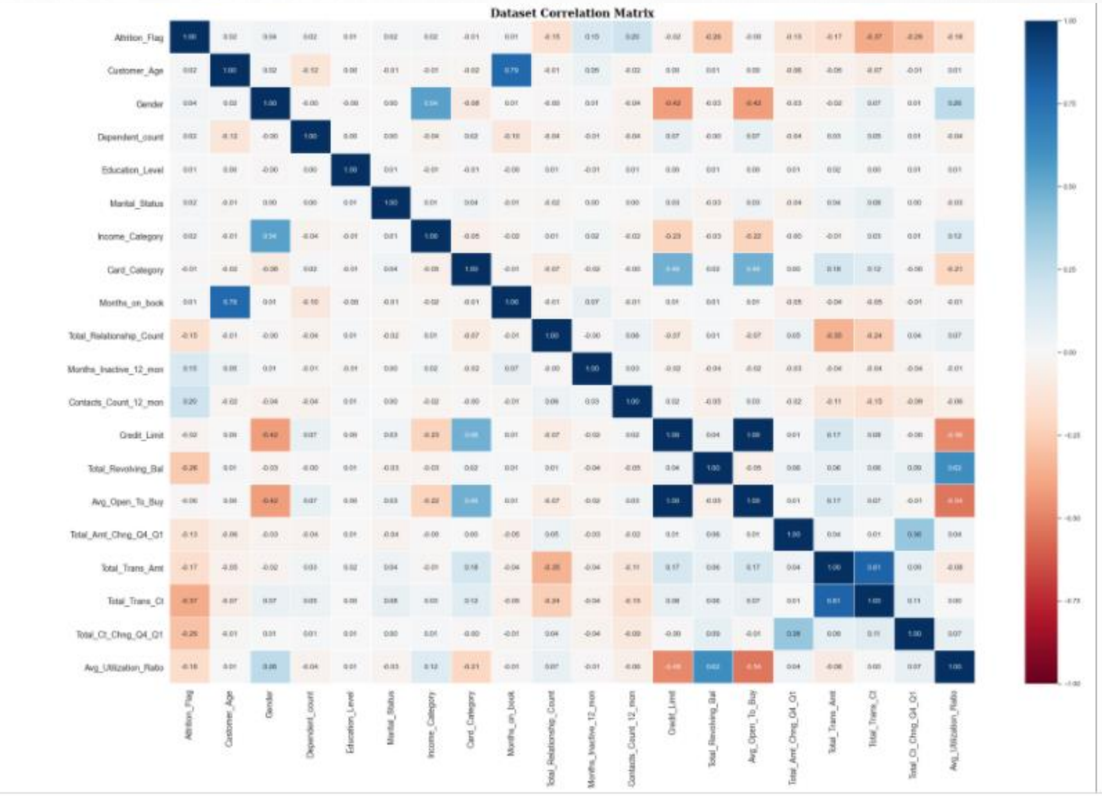


INFRENCE FROM UNIVARIATE ANALYSIS

- The customer age mean is ~46 this means that the **data isn't from a nonbank and seems to be more a traditional bank**, also the min customer age is 26 which is good (there isn't any underage data) and the max age is 70 which is a rational number.
- Dependent count is between 0 and 5 with a mean of 2.3, which seems fine.
- The least relationship_count is 1, which seems fine at least 1 product customers. For the max product that a user got is 6, we will check if there is a relationship between the total relationship and the churn, *hypothetically users that acquire more products are less likely to churn.*
- Months inactive the least value is 0 the most is 6. *Hypothetically when a user gets more inactive will be most likely to churn.*
- Total_trans_ct on average there are 65 tx, approx. 5.4 per month. The max value is 139 more than the double of the mean, indicating a possible outlier. *The more the transaction could indicate that they are users less likely to churn.*
- Total_ct_chng_q4_q1 seem to have a similar behavior from the total amount changed.
- Avg. Utilization of the credit limit is 0.27 but the median is 0.18 which could indicate a positive skew, also there seem to be outliers on the left side. **The percentile 25 is 0.02 indicates that 25% are hardly using the product.**

EXPLORATORY DATA ANALYSIS (BIVARIATE)

- This graph is of correlation between variables in the form of heat map.
- There is a high correlation between customer age ,monthly on book and aug utilization ratio and total_trains_amt. If we use those features which have high correlation between variables to make our model, then model performance will increase.
- By using the most important features yields high accuracy as compared to using all features for Model building.



EXPLORATORY DATA ANALYSIS (BIVARIATE)

Comparing the age distribution vs the target

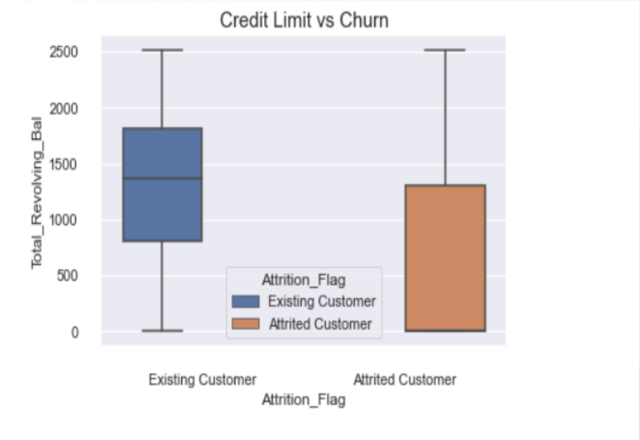
Means comparing which customers have left more balance in the credit card

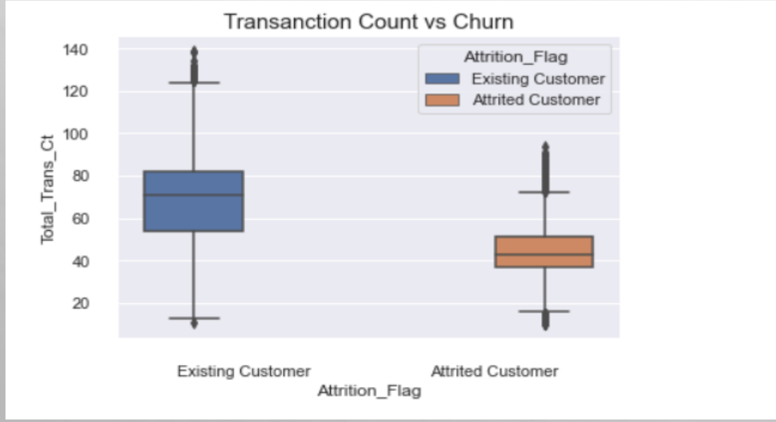
By seeing this boxplot, we conclude that existing customer have more amount left on credit card to use else the Attired customer have low amount on credit card as compared to existing customer.

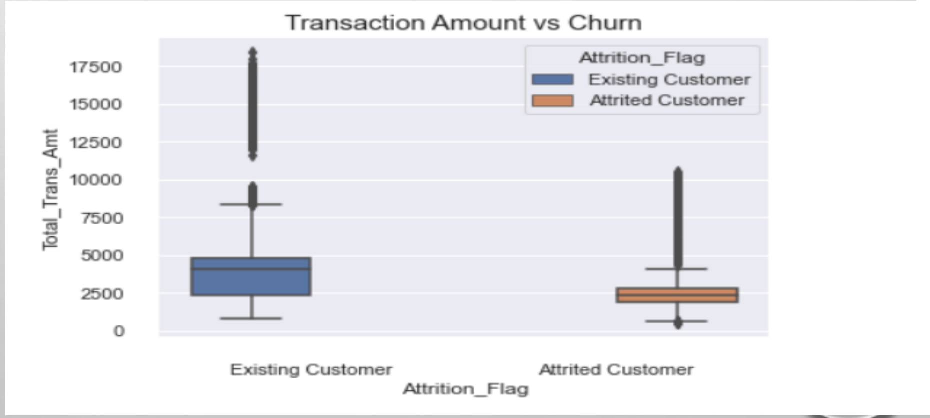
But there is no distinctive difference.
As we see there is not much noticeable difference between both.

INFERENCE FROM BIVARIATE ANALYSIS

- Let's check the relationship of two variables concerning each other. This is done because it helps us to detect anomalies, understand de dependence of two variables on each other, the impact of each variable within the target variable (giving us good insights).
- From these box plots matrix for each continuous variable, there seem to be differences between attrited vs non-attrited, for the following variables: total_trans_ct, total_trans_amount, total_revolving_bal.
- From these three box plots we understand that there seem to be difference between Existing customer and Attired customer.







OBSERVATION IS THAT EITHER CHURN OR NON-CHURN PROFILE WHICH ARE LESS ACTIVE

- According to the EDA above, the profiles underneath can be made. It's clear that the main difference lays in the "product variables" of the customers. And those product variables are TYPE OF CARD, LENGTH OF RELATIONSHIP, PRODUCTS BOUGHT, INACTIVE MONTHS, NUMBER OF CONTACT.
- A churning customer tends to be less active than an existing customer. It's clear that the most influential parameters are features related to the activity of the customer. And those feature are OPEN TO BUY CREDIT LINE, TRANSACTION AMOUNT CHANGE, TOTAL TRANSACTION AMOUNT, TOTAL TRANSACTION COUNT

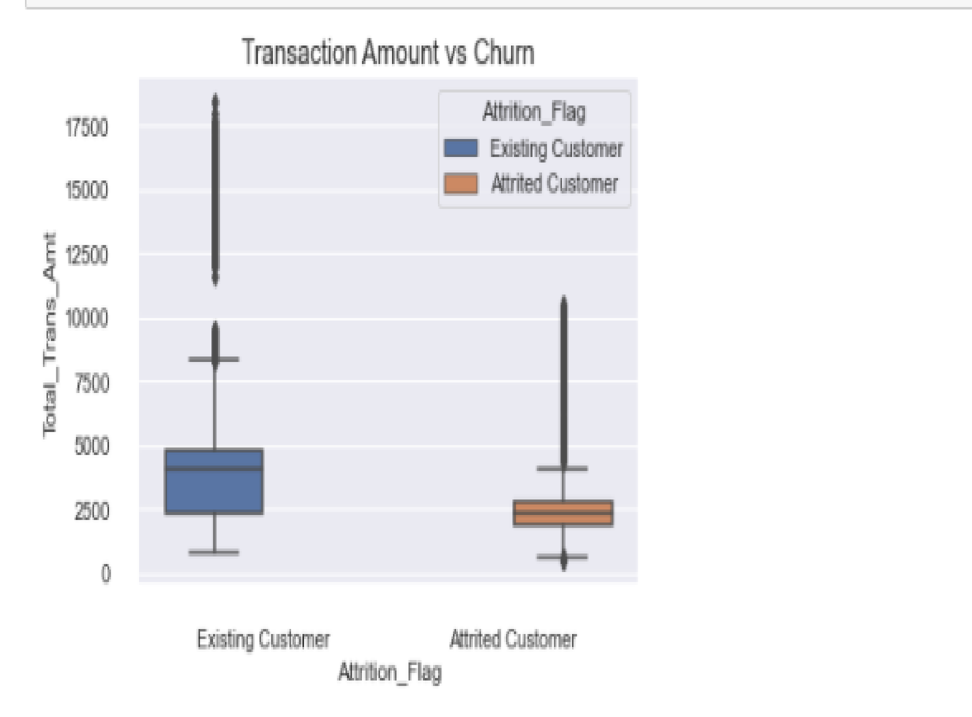
	Non Churning Customer	Churning Customer
Demographic variables		
Age	47	46
Gender	F/M	F/M
Dependents	2	2
Education Level	Graduate	Graduate
Marital Level	Married/Single	Married/Single
Income Category	Less then $40K	Less then $40K
Product variables		
Type Of Card	Blue	Blue
Length Of Relationship	36 months	36 months
Products Bought	4	3
Inactive Months	2	3
Number Of Contact	2	3
Credit Limit	$8726	$8136
Revolving Balance	1256	672
Open To Buy Credit Line	7470	7463
Transaction Amount Change	0.77	0.69
Total Transaction Amount	4650	3095
Total Transaction Count	69	45
Transaction Count Change	0.74	0.55
Card Utilization Ratio	0.3	0.16

DATA PREPARATION

- Before we start training a model, we must prepare our data. Different steps that we can undertake:
- Encode all categorical data (watch out with one hot encoding and tree-based models…).
- Scale data
- Check correlation matrix to extract the most influential features.
- Generate new columns from data.
- Up sample the imbalanced dataset (smote/adasyn).
- In this notebook we shall focus on the upsampling method. The data wrangling performed is to make sure that the upsampling is performed in a correct manner.

CHECKING THE RELATIONSHIP BETWEEN ATTRITION_FLAG AND TOTAL_TRANS_AMT

- It's clear that the transaction amount is lower for the churned customers compared to the existing customers.

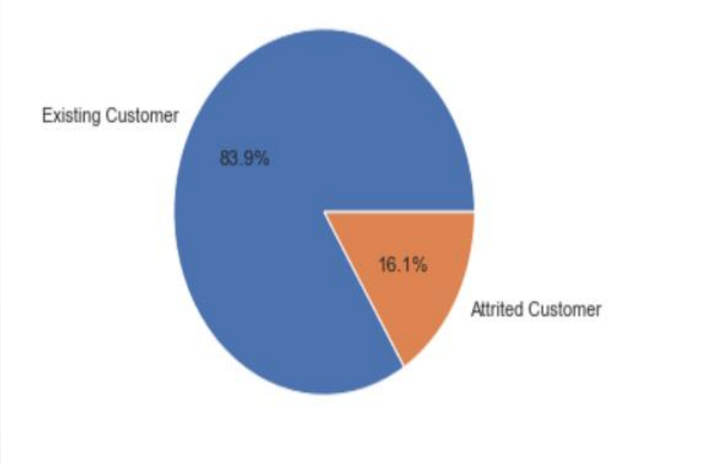


IMBALANCED DATASET

- It's clear that most of the customers (83.9 %) stays. Since "attrited" or "churned" label is less then 20% of the total all customers. We can say that we have an imbalanced data.

- Upsampling or Undersampling will be required to receive a better results.

- Existing customers are in majority numbers and Attrited customers are in minority numbers. As we see in graph as well.



CATEGORICAL FEATURES WILL NEED ENCODING

When we need to use SMOTE(Sampling) we'll need to encode our categorical features. Features we need encode are **Attrition_Flag, Customer_Age, Gender, Dependent_count, Education_Level, Marital_Status, Income_Category, Card_Category, Months_on_book, Total_Relationship_count, Months_Inactive_12_mon etc.**

	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category	Months_on_book	Tota
CLIENTNUM										
768805383	0	45	0	3	3	1	2	0	39	
818770008	0	49	1	5	2	2	4	0	44	
713982108	0	51	0	3	2	1	3	0	36	
769911858	0	40	1	4	3	3	4	0	34	
709106358	0	40	0	3	5	1	2	0	21	
...	...	...	...	...	...	...	...	...	...	
772366833	0	50	0	2	2	2	1	0	40	
710638233	1	41	0	2	6	0	1	0	25	
716506083	1	44	1	1	3	1	4	0	36	
717406983	1	30	0	2	2	3	1	0	36	
714337233	1	43	1	2	2	1	4	3	25	

FEATURE ENGINEERING OR SELECTION

- Let's try to remove columns with percentage of high one category values and high missing values.
- Removing columns with 90% features with one category only and 90% features with missing values.
- Those features achieve through feature engineering.

	Feature	Unique_values	Percentage of missing values	percentage high one category values	type
0	Attrition_Flag	2	0.0	83.934038	object
1	Customer_Age	45	0.0	4.937296	int64
18	Total_Ct_Chng_Q4_Q1	830	0.0	1.688555	float64
17	Total_Trans_Ct	126	0.0	2.053915	int64
16	Total_Trans_Amt	5033	0.0	0.108621	int64
15	Total_Amt_Chng_Q4_Q1	1158	0.0	0.355485	float64
14	Avg_Open_To_Buy	6813	0.0	3.199368	float64
13	Total_Revolving_Bal	1974	0.0	24.390244	int64
12	Credit_Limit	6205	0.0	5.016293	float64
11	Contacts_Count_12_mon	7	0.0	33.376123	int64
10	Months_Inactive_12_mon	7	0.0	37.977683	int64
9	Total_Relationship_Count	6	0.0	22.760936	int64
8	Months_on_book	44	0.0	24.321122	int64
7	Card_Category	4	0.0	93.176656	object
6	Income_Category	6	0.0	35.163425	object
5	Marital_Status	4	0.0	46.282216	object
4	Education_Level	7	0.0	30.887726	object
3	Dependent_count	6	0.0	26.977387	int64
2	Gender	2	0.0	52.908068	object
19	Avg_Utilization_Ratio	964	0.0	24.390244	float64

AFTER FEATURE ENGINEERING

After feature engineering we select those columns and by selecting those columns, model will provide higher accuracy score. Those columns are Attrition_Flag, CUSTOMER_AGE, GENDER, DEPENDENT_COUNT,EDUCATION_LEVEL, MATRIAL_STATUS, INCOME_CATEGORY,CARD _CATEGORY

	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category	Months_on_book
CLIENTNUM									
768805383	Existing Customer	45	M	3	High School	Married	60K−80K	Blue	39
818770008	Existing Customer	49	F	5	Graduate	Single	Less than $40K	Blue	44
713982108	Existing Customer	51	M	3	Graduate	Married	80K−120K	Blue	36
769911858	Existing Customer	40	F	4	High School	Unknown	Less than $40K	Blue	34
709106358	Existing Customer	40	M	3	Uneducated	Married	60K−80K	Blue	21

From this we can see that we have 2 columns (columns Education_Level, Marital_Status) in which have missing values. So, we will need to handle these missing values.

```
Attrition_Flag                 0
Customer_Age                   0
Gender                         0
Dependent_count                0
Education_Level             1519
Marital_Status               749
Income_Category                0
Card_Category                  0
Months_on_book                 0
Total_Relationship_Count       0
Months_Inactive_12_mon         0
Contacts_Count_12_mon          0
Credit_Limit                   0
Total_Revolving_Bal            0
Avg_Open_To_Buy                0
Total_Amt_Chng_Q4_Q1           0
Total_Trans_Amt                0
Total_Trans_Ct                 0
Total_Ct_Chng_Q4_Q1            0
Avg_Utilization_Ratio          0
dtype: int64
```

HANDLING WITH MISSING VALUES

- After putting 'unknown' where we have missing values in object.
- We have two columns Education_Level, Marital_Status.
- So, we put 'unknown' and fill it
- After that we are checking the column again and we see nothing null value on it.

CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category	Months_on_book	To
768805383	Existing Customer	45	M	3	High School	Married	60K−80K	Blue	39	
818770008	Existing Customer	49	F	5	Graduate	Single	Less than $40K	Blue	44	
713982108	Existing Customer	51	M	3	Graduate	Married	80K−120K	Blue	36	
769911858	Existing Customer	40	F	4	High School	Unknown	Less than $40K	Blue	34	
709106358	Existing Customer	40	M	3	Uneducated	Married	60K−80K	Blue	21	

```
Attrition_Flag              0
Customer_Age                0
Gender                      0
Dependent_count             0
Education_Level             0
Marital_Status              0
Income_Category             0
Card_Category               0
Months_on_book              0
Total_Relationship_Count    0
Months_Inactive_12_mon      0
Contacts_Count_12_mon       0
Credit_Limit                0
Total_Revolving_Bal         0
Avg_Open_To_Buy             0
Total_Amt_Chng_Q4_Q1        0
Total_Trans_Amt             0
Total_Trans_Ct              0
Total_Ct_Chng_Q4_Q1         0
Avg_Utilization_Ratio       0
dtype: int64
```

PREDICTION OF RELIABLE 7 MODELS' ACCURACY SIMPLY MODEL BUILDING

- Decision Tree classification model

Decision tree accuracy on the testing data was at about 94% accuracy which is REASONABLE ACCURACY. The recall is low, at around 81%. And precision accuracy is 80 and f1_score is around 81.

```
********************Confusion Matrix**********************
[[2502   89]
 [  81  367]]
********************Classification Report**********************
              precision    recall  f1-score   support

           0       0.97      0.97      0.97      2591
           1       0.80      0.82      0.81       448

    accuracy                           0.94      3039
   macro avg       0.89      0.89      0.89      3039
weighted avg       0.94      0.94      0.94      3039
```

]:

	accuracy	precision	recall	f1_score
Decision Tree	94.406055	80.482456	81.919643	81.19469

- Bagging classifier

Accuracy of the bagging classifier model was at about 95% which was what was predicted as a reasonable accuracy for the weighted model. Recall saw an improvement over the decision tree model, but it is still good(90%). Latency of the prediction is much greater than the decision tree, but it is not clear whether latency is important here. Fi score is 85% and precision is around 81.

```
********************Confusion Matrix**********************
[[2546   86]
 [  37  370]]
********************CLassification Report*********************
              precision    recall  f1-score   support

           0       0.99      0.97      0.98      2632
           1       0.81      0.91      0.86       407

    accuracy                           0.96      3039
   macro avg       0.90      0.94      0.92      3039
weighted avg       0.96      0.96      0.96      3039
```

]:

	accuracy	precision	recall	f1_score
Bagging Boosting classifier	95.952616	81.140351	90.909091	85.747393

- Random Forest

The random forest model has a around 95% accuracy which is good, like bagging. Recall is better than random at 92%, though not by much. Precision is around 79 percent and f1_score is around 85%.

```
********************Confusion Matrix**********************
[[2553   95]
 [  30  361]]
********************CLassification Report*********************
              precision    recall  f1-score   support

           0       0.99      0.96      0.98      2648
           1       0.79      0.92      0.85       391

    accuracy                           0.96      3039
   macro avg       0.89      0.94      0.91      3039
weighted avg       0.96      0.96      0.96      3039
```

5]:

	accuracy	precision	recall	f1_score
Random Forest	95.886805	79.166667	92.327366	85.242031

- Ada boost

Model performance is great. Again, the accuracy is near to 95% which is quite good but as we say recall is 88% and precision is around 82 and f1_score is 85%.

```
********************Confusion Matrix*********************
[[2532   78]
 [  51  378]]
********************CLassification Report*********************
              precision    recall  f1-score   support

           0       0.98      0.97      0.98      2610
           1       0.83      0.88      0.85       429

    accuracy                           0.96      3039
   macro avg       0.90      0.93      0.91      3039
weighted avg       0.96      0.96      0.96      3039
```

]:

	accuracy	precision	recall	f1_score
Ada BoostClassifier classifier	95.755183	82.894737	88.111888	85.423729

- Gradient boosting classifier

This model has slightly higher than predict accuracy around 96% . recall at 93% is an. The model has precision of 82 and f1_score is 87%.

```
********************Confusion Matrix**********************
[[2555   80]
 [  28  376]]
********************CLassification Report*********************
              precision    recall  f1-score   support

           0       0.99      0.97      0.98      2635
           1       0.82      0.93      0.87       404

    accuracy                           0.96      3039
   macro avg       0.91      0.95      0.93      3039
weighted avg       0.97      0.96      0.97      3039
```

	accuracy	precision	recall	f1_score
Gradient Boosting	96.446199	82.45614	93.069307	87.44186

- Xgboost classifier

Accuracy near prediction 97% but recall lower than the gradient boosting model. Recall is around 93 and precision is almost 87 and f1_score near 90.

```
********************Confusion Matrix**********************
[[2556   57]
 [  27  399]]
********************CLassification Report**********************
              precision    recall  f1-score   support

           0       0.99      0.98      0.98      2613
           1       0.88      0.94      0.90       426

    accuracy                           0.97      3039
   macro avg       0.93      0.96      0.94      3039
weighted avg       0.97      0.97      0.97      3039
```

	accuracy	precision	recall	f1_score
XGBoost classifier	97.235933	87.5	93.661972	90.47619

- Logistic regression

Logistic regression accuracy is lower from all others model which is near 88 and recall is near 65 and f1_score is around 51 and precision is 42.

```
********************Confusion Matrix**********************
[[2480  261]
 [ 103  195]]
********************CLassification Report**********************
              precision    recall  f1-score   support

           0       0.96      0.90      0.93      2741
           1       0.43      0.65      0.52       298

    accuracy                           0.88      3039
   macro avg       0.69      0.78      0.72      3039
weighted avg       0.91      0.88      0.89      3039
```

	accuracy	precision	recall	f1_score
Logistic Regression	88.022376	42.763158	65.436242	51.724138

MODELING ON OVERSAMPLED DATA

- Notice the large discrepency between the amount of data for existing customers and churned customers. This difference in data points is what we call bias error. This will lead our model to underfit the data related to the churned customers. In the end this results in the model miss the small relevent relationships in the data. This will lead to predictions that are not accurate towards deciding who is considered a churned customer. In order to remedy this situation, we will introduce some synthetically generated data points in order to balance the samples in order to fit the model better.
- This will be done using the smote or synthetic minority oversampling technique. How this technique works in laymans terms is that SMOTE will select similar data points from the data we are trying to over sample.

MODEL BULIDING WITH OVERSAMPLED

- Logistic regression, Decision Tree Classifer, Random Forest, Gradient Boosting, Adaboost Classifier, Xgboost, Bagging Classifer.
- Now we will only see the confusion matrix accuracy precision recall and compare them that based on over sampling model building and check which classifier performance is good.

```
********************Confusion Matrix**********************
[[2551   88]
 [  32  368]]
********************CLassification Report*********************
              precision    recall  f1-score   support

           0       0.99      0.97      0.98      2639
           1       0.81      0.92      0.86       400

    accuracy                           0.96      3039
   macro avg       0.90      0.94      0.92      3039
weighted avg       0.96      0.96      0.96      3039
```

]:

	accuracy	precision	recall	f1_score
Random Forest	96.051333	80.701754	92.0	85.981308

```
*******************Confusion Matrix**********************
[[2493   95]
 [  90  361]]
*******************CLassification Report*********************
              precision    recall  f1-score   support

           0       0.97      0.96      0.96      2588
           1       0.79      0.80      0.80       451

    accuracy                           0.94      3039
   macro avg       0.88      0.88      0.88      3039
weighted avg       0.94      0.94      0.94      3039
```

	accuracy	precision	recall	f1_score
Decision Tree	93.912471	79.166667	80.044346	79.603087

```
*******************Confusion Matrix*********************
[[2480  261]
 [ 103  195]]
*******************CLassification Report*********************
              precision    recall  f1-score   support

           0       0.96      0.90      0.93      2741
           1       0.43      0.65      0.52       298

    accuracy                           0.88      3039
   macro avg       0.69      0.78      0.72      3039
weighted avg       0.91      0.88      0.89      3039
```

]:

	accuracy	precision	recall	f1_score
Logistic Regression	88.022376	42.763158	65.436242	51.724138

```
********************Confusion Matrix**********************
[[2546   86]
 [  37  370]]
********************CLassification Report**********************
              precision    recall  f1-score   support

           0       0.99      0.97      0.98      2632
           1       0.81      0.91      0.86       407

    accuracy                           0.96      3039
   macro avg       0.90      0.94      0.92      3039
weighted avg       0.96      0.96      0.96      3039
```

	accuracy	precision	recall	f1_score
Bagging Boosting classifier	95.952616	81.140351	90.909091	85.747393

```
********************Confusion Matrix**********************
[[2532   78]
 [  51  378]]
********************CLassification Report**********************
              precision    recall  f1-score   support

           0       0.98      0.97      0.98      2610
           1       0.83      0.88      0.85       429

    accuracy                           0.96      3039
   macro avg       0.90      0.93      0.91      3039
weighted avg       0.96      0.96      0.96      3039
```

	accuracy	precision	recall	f1_score
Ada BoostClassifier classifier	95.755183	82.894737	88.111888	85.423729

```
*******************Confusion Matrix*********************
[[2556   57]
 [  27  399]]
*******************CLassification Report*********************
              precision    recall  f1-score   support

           0       0.99      0.98      0.98      2613
           1       0.88      0.94      0.90       426

    accuracy                           0.97      3039
   macro avg       0.93      0.96      0.94      3039
weighted avg       0.97      0.97      0.97      3039
```

	accuracy	precision	recall	f1_score
XGBoost classifier	97.235933	87.5	93.661972	90.47619

```
*******************Confusion Matrix*********************
[[2555   80]
 [  28  376]]
*******************CLassification Report*********************
              precision    recall  f1-score   support

           0       0.99      0.97      0.98      2635
           1       0.82      0.93      0.87       404

    accuracy                           0.96      3039
   macro avg       0.91      0.95      0.93      3039
weighted avg       0.97      0.96      0.97      3039
```

	accuracy	precision	recall	f1_score
Gradient Boosting	96.446199	82.45614	93.069307	87.44186

- We see that logistic regression have accuracy low from other. Others are decision tree, adaboost, gradient boost and xgboost.
- Decision tree have accuracy close to 93 and other like adaboost or xgboost or gradient boost those all have accuracy more then 95 which is reasonable and, they give good recall and precision.
- We build this model by doing over sampling on it and in next phase we check our model by doing under sampling on it.

MODEL BUILDING - UNDER SAMPLED DATA

- Similar to OVERSAMPLING but key difference is that we delete examples from the majority class. In oversampling to equal both categories **existing customer** and **attrired** but in undersampling we delete examples from majority class so that they are equal to minority class.
- Like existing customer are almost 85% and only 15% are attrited now we delete existing customer examples randomlly so that they equal to attired and then we build our model.

MODEL BULIDING WITH UNDERSAMPLING

- Logistic Regression, Decision Tree Classifer, Random Forest, Gradient Boosting, Adaboost Classifier, Xgboost, Bagging Classifer.
- Now we will only see confusion matrix accuracy precision recall and compare them based on oversampling model building and then we will see which classifier is performing better.

```
********************Confusion Matrix**********************
[[2498   89]
 [  85  367]]
********************CLassification Report*********************
              precision    recall  f1-score   support

           0       0.97      0.97      0.97      2587
           1       0.80      0.81      0.81       452

    accuracy                           0.94      3039
   macro avg       0.89      0.89      0.89      3039
weighted avg       0.94      0.94      0.94      3039
```

	accuracy	precision	recall	f1_score
Decision Tree	94.274432	80.482456	81.19469	80.837004

```
********************Confusion Matrix**********************
[[2480  261]
 [ 103  195]]
********************CLassification Report**********************
              precision    recall  f1-score   support

           0       0.96      0.90      0.93      2741
           1       0.43      0.65      0.52       298

    accuracy                           0.88      3039
   macro avg       0.69      0.78      0.72      3039
weighted avg       0.91      0.88      0.89      3039
```

	accuracy	precision	recall	f1_score
Logistic Regression	88.022376	42.763158	65.436242	51.724138

```
********************Confusion Matrix**********************
[[2555   80]
 [  28  376]]
********************CLassification Report**********************
              precision    recall  f1-score   support

           0       0.99      0.97      0.98      2635
           1       0.82      0.93      0.87       404

    accuracy                           0.96      3039
   macro avg       0.91      0.95      0.93      3039
weighted avg       0.97      0.96      0.97      3039
```

	accuracy	precision	recall	f1_score
Gradient Boosting	96.446199	82.45614	93.069307	87.44186

```
*******************Confusion Matrix**********************
[[2546   86]
 [  37  370]]
*******************CLassification Report**********************
              precision    recall  f1-score   support

           0       0.99      0.97      0.98      2632
           1       0.81      0.91      0.86       407

    accuracy                           0.96      3039
   macro avg       0.90      0.94      0.92      3039
weighted avg       0.96      0.96      0.96      3039
```

	accuracy	precision	recall	f1_score
Bagging Boosting classifier	95.952616	81.140351	90.909091	85.747393

```
*******************Confusion Matrix**********************
[[2532   78]
 [  51  378]]
*******************CLassification Report**********************
              precision    recall  f1-score   support

           0       0.98      0.97      0.98      2610
           1       0.83      0.88      0.85       429

    accuracy                           0.96      3039
   macro avg       0.90      0.93      0.91      3039
weighted avg       0.96      0.96      0.96      3039
```

	accuracy	precision	recall	f1_score
Ada BoostClassifier classifier	95.755183	82.894737	88.111888	85.423729

```
********************Confusion Matrix**********************
[[2556   57]
 [  27  399]]
********************CLassification Report*********************
              precision    recall  f1-score   support

           0       0.99      0.98      0.98      2613
           1       0.88      0.94      0.90       426

    accuracy                           0.97      3039
   macro avg       0.93      0.96      0.94      3039
weighted avg       0.97      0.97      0.97      3039
```

	accuracy	precision	recall	f1_score
XGBoost classifier	97.235933	87.5	93.661972	90.47619

We see that logistic regression have accuracy low from others. Others are decision tree, adaboost, gradient boost and xgboost.

Decision tree have accuracy close to 94 and other like Adaboost or Xgboost or gradient boost those all have accuracy more then 95 which is reasonable and, they give good recall and precision.

In next section we will select 3 models which have low accuracy for tuning so that after performing tunning they perform well.

So, we select Logistic model and two more and tunning those model parameter so that we can get good performance. Due their low accuracy we will tune parameters so that performance can improve.

CHECK PERFORMANCE

Below we see results one by one like first we see results of logistic regression over data and then on oversampling data and after that in under sampling data. And all others like random_forest,gradient_boost, bagging and xgboost.

```
logistic_result

[                      accuracy  precision     recall   f1_score
 Logistic Regression  88.022376  42.763158  65.436242  51.724138,
                                  accuracy  precision     recall  \
 Logistic Regression Over sample data  88.022376  42.763158  65.436242

                                  f1_score
 Logistic Regression Over sample data  51.724138  ,
                                accuracy  precision     recall   f1_score
 Logistic Regression Under Sampling  88.022376  42.763158  65.436242  51.724138]
```

```
random_forest_result

[                accuracy  precision  recall   f1_score
 Random Forest  95.985522  80.482456   91.75  85.747664,
                             accuracy  precision    recall  f1_score
 Random Forest Oversample data  95.788088  79.605263  91.20603  85.01171]
```

```
decision_tree_result
```

```
[                 accuracy  precision     recall   f1_score
 Decision Tree  93.978282  79.166667  80.400891  79.779006,
                                accuracy  precision     recall   f1_score
 Decision Tree Over sample data  93.879566   80.04386  79.347826  79.694323,
                                 accuracy  precision     recall   f1_score
 Decision Tree Undersample data  94.109905  80.921053  80.043384  80.479826]
```

```
bagging_result
```

```
[                              accuracy  precision     recall   f1_score
 Bagging Boosting classifier  95.952616  81.140351  90.909091  85.747393,
                                              accuracy  precision     recall  \
 Bagging Boosting Over sample data classifier  95.952616  81.140351  90.909091

                                              f1_score
 Bagging Boosting Over sample data classifier  85.747393  ,
                                          accuracy  precision     recall  \
 Bagging Boosting classifier Under sample  95.952616  81.140351  90.909091

                                          f1_score
 Bagging Boosting classifier Under sample  85.747393  ]
```

```
gradient_result
```

```
[                    accuracy  precision     recall  f1_score
 Gradient Boosting  96.446199   82.45614  93.069307  87.44186,
                                    accuracy  precision     recall  f1_score
 Gradient Boosting Oversample data  96.446199   82.45614  93.069307  87.44186,
                                        accuracy  precision     recall  \
 Gradient Boosting Under sampling data  96.446199   82.45614  93.069307

                                        f1_score
 Gradient Boosting Under sampling data  87.44186  ]
```

```
adaboost_result
```

```
[                                   accuracy  precision     recall   f1_score
 Ada BoostClassifier classifier  95.755183  82.894737  88.111888  85.423729,
                                                accuracy  precision  \
 Ada BoostClassifier Oversample data classifier  95.755183  82.894737

                                                   recall   f1_score
 Ada BoostClassifier Oversample data classifier  88.111888  85.423729  ,
                                                 accuracy  precision  \
 Ada BoostClassifier classifier Undersample data  95.755183  82.894737

                                                   recall   f1_score
 Ada BoostClassifier classifier Undersample data  88.111888  85.423729  ]
```

```
xgboost_result
```

```
[                     accuracy  precision     recall  f1_score
 XGBoost classifier  97.235933       87.5  93.661972  90.47619,
                                     accuracy  precision     recall  f1_score
 XGBoost classifier Oversample data  97.235933       87.5  93.661972  90.47619,
                                       accuracy  precision     recall  \
 XGBoost classifier Under sample data  97.235933       87.5  93.661972

                                       f1_score
 XGBoost classifier Under sample data  90.47619  ]
```

MODEL TUNING

- Now we select three models and tune their parameter those are logistic regression bagging boosting and gradient boosting classifer.
- And we check their performance.
- Now we check by seeing accuracy recall and precision and see if there is an improvement in accuracy.

```
********************Confusion Matrix**********************
[[2483  191]
 [ 100  265]]
********************CLassification Report**********************
              precision    recall  f1-score   support

           0       0.96      0.93      0.94      2674
           1       0.58      0.73      0.65       365

    accuracy                           0.90      3039
   macro avg       0.77      0.83      0.80      3039
weighted avg       0.92      0.90      0.91      3039
```

	accuracy	precision	recall	f1_score
Logistic Regression	90.424482	58.114035	72.60274	64.55542

```
*******************Confusion Matrix**********************
[[2545   69]
 [  38  387]]
*******************CLassification Report*********************
              precision    recall  f1-score   support

           0       0.99      0.97      0.98      2614
           1       0.85      0.91      0.88       425

    accuracy                           0.96      3039
   macro avg       0.92      0.94      0.93      3039
weighted avg       0.97      0.96      0.97      3039
```

	accuracy	precision	recall	f1_score
Bagging Boosting classifier	96.479105	84.868421	91.058824	87.854711

```
*******************Confusion Matrix**********************
[[2553  115]
 [  30  341]]
*******************CLassification Report*********************
              precision    recall  f1-score   support

           0       0.99      0.96      0.97      2668
           1       0.75      0.92      0.82       371

    accuracy                           0.95      3039
   macro avg       0.87      0.94      0.90      3039
weighted avg       0.96      0.95      0.95      3039
```

	accuracy	precision	recall	f1_score
Gradient Boosting	95.228694	74.780702	91.913747	82.466747

- First, we see from the chart that logistic regression improved up to 5% which is great. Accuracy improved from 85% to 90% now, which is reasonably good.
- Second, we see that bagging and gradient improved accuracy from 1% to 2 % more which is another good indicator. Both models are giving accuracy more than 86% now and recall and precision is also good.

After performing tunning, we see that the Logistic Regression, Bagging and Gradient Boosting improved in accuracy and recall. Bagging Boosting classifier is also giving good accuracy score.

```
tunne_result

[                        accuracy  precision    recall  f1_score
 Logistic Regression  90.424482  58.114035  72.60274  64.55542,
                              accuracy  precision     recall   f1_score
 Bagging Boosting classifier  96.479105  84.868421  91.058824  87.854711,
                    accuracy  precision     recall   f1_score
 Gradient Boosting  95.228694  74.780702  91.913747  82.466747]
```

HYPER PARAMETER TUNING USING GRID SEARCH

Grid-search is used **to find the optimal hyperparameters of a model which results in the most 'accurate' predictions.**
That's why we are using grid search by giving different types of parameters.
This method will pick the best combination of parameters which will provide high accuracy.

After performing hyperparameter we see that logistic regression accuracy is 88% while before hyperparameter tunning accuracy was under 84%. So, we do see the accuracy improved.

```
********************Confusion Matrix**********************
[[2481  262]
 [ 102  194]]
********************CLassification Report**********************
              precision    recall  f1-score   support

           0       0.96      0.90      0.93      2743
           1       0.43      0.66      0.52       296

    accuracy                           0.88      3039
   macro avg       0.69      0.78      0.72      3039
weighted avg       0.91      0.88      0.89      3039
```

]:

	accuracy	precision	recall	f1_score
Logistic Regression	88.022376	42.54386	65.540541	51.595745

Accuracy score for Gradient boosting classifier and bagging boosting classifier also improved after performing hyperparameter tunning By using grid search it increased almost 2% more accuracy which is quite good.

```
********************Confusion Matrix**********************
[[2562  160]
 [  21  296]]
********************CLassification Report*********************
              precision    recall  f1-score   support

           0       0.99      0.94      0.97      2722
           1       0.65      0.93      0.77       317

    accuracy                           0.94      3039
   macro avg       0.82      0.94      0.87      3039
weighted avg       0.96      0.94      0.95      3039
```

	accuracy	precision	recall	f1_score
Gradient Boosting classifier	94.044093	64.912281	93.375394	76.584735

```
********************Confusion Matrix**********************
[[2560   89]
 [  23  367]]
********************CLassification Report*********************
              precision    recall  f1-score   support

           0       0.99      0.97      0.98      2649
           1       0.80      0.94      0.87       390

    accuracy                           0.96      3039
   macro avg       0.90      0.95      0.92      3039
weighted avg       0.97      0.96      0.96      3039
```

	accuracy	precision	recall	f1_score
Bagging classifier	96.314577	80.482456	94.102564	86.761229

We conclude that bagging classifier model provides good accuracy as compared to gradient boosting classifier and logistic regression.

```
grid_results
```

```
[                       accuracy  precision     recall   f1_score
 Bagging classifier  96.314577  80.482456  94.102564  86.761229,
                                  accuracy  precision     recall   f1_score
 Gradient Boosting classifier  94.044093  64.912281  93.375394  76.584735,
                         accuracy  precision     recall   f1_score
 Logistic Regression  88.022376   42.54386  65.540541  51.595745]
```

MODEL PERFORMANCES

- Compare the model performance of tuned models - choose the best model - recall on the test set is expected to be > 0.95, and precision and accuracy is expected to be > 0.70

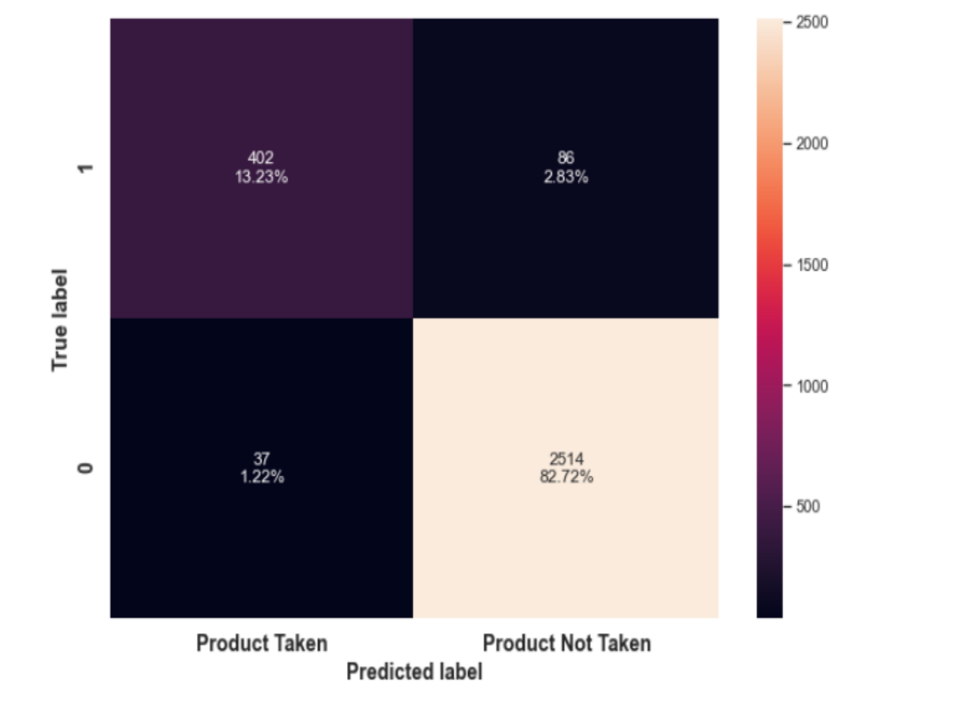
	Model_name	Accuracy	Precision	Recall	F1	Classification Report	Confusion Matrix
0	Random Forest	0.957223	0.957223	0.957223	0.957223	precision recall f1-score ...	[[2552, 31], [99, 357]]
1	Gradient Boosting	0.964462	0.964462	0.964462	0.964462	precision recall f1-score ...	[[2555, 28], [80, 376]]
2	DecisionTreeClassifier	0.941099	0.941099	0.941099	0.941099	precision recall f1-score ...	[[2500, 83], [96, 360]]
3	Logistic Regression	0.880224	0.880224	0.880224	0.880224	precision recall f1-score ...	[[2479, 104], [260, 196]]
4	Bagging Boosting classifier	0.959526	0.959526	0.959526	0.959526	precision recall f1-score ...	[[2546, 37], [86, 370]]
5	Ada BoostClassifier classifier	0.957552	0.957552	0.957552	0.957552	precision recall f1-score ...	[[2532, 51], [78, 378]]
6	XGBoost classifier	0.972359	0.972359	0.972359	0.972359	precision recall f1-score ...	[[2556, 27], [57, 399]]
7	Random Forest Oversample data	0.957552	0.957552	0.957552	0.957552	precision recall f1-score ...	[[2503, 80], [49, 407]]
8	Decision Tree Over sample data	0.936163	0.936163	0.936163	0.936163	precision recall f1-score ...	[[2464, 119], [75, 381]]
9	Logistic Regression Over sample data	0.797302	0.797302	0.797302	0.797302	precision recall f1-score ...	[[2067, 516], [100, 356]]
10	Bagging Boosting Over sample data classifier	0.951629	0.951629	0.951629	0.951629	precision recall f1-score ...	[[2491, 92], [55, 401]]
11	Ada BoostClassifier Oversample data classifier	0.936163	0.936163	0.936163	0.936163	precision recall f1-score ...	[[2430, 153], [41, 415]]
12	XGBoost classifier Oversample data	0.964462	0.964462	0.964462	0.964462	precision recall f1-score ...	[[2516, 67], [41, 415]]
13	Gradient Boosting Oversample data	0.951300	0.951300	0.951300	0.951300	precision recall f1-score ...	[[2466, 117], [31, 425]]
14	Decision Tree Undersample data	0.931227	0.931227	0.931227	0.931227	precision recall f1-score ...	[[2440, 143], [66, 390]]
15	Logistic Regression Under Sampling	0.802896	0.802896	0.802896	0.802896	precision recall f1-score ...	[[2080, 503], [96, 360]]
16	Gradient Boosting Under sampling data	0.950971	0.950971	0.950971	0.950971	precision recall f1-score ...	[[2466, 117], [32, 424]]
17	Bagging Boosting classifier Under sample	0.950642	0.950642	0.950642	0.950642	precision recall f1-score ...	[[2492, 91], [59, 397]]
18	Ada BoostClassifier classifier Undersample data	0.937150	0.937150	0.937150	0.937150	precision recall f1-score ...	[[2436, 147], [44, 412]]
19	XGBoost classifier Under sample data	0.966107	0.966107	0.966107	0.966107	precision recall f1-score ...	[[2513, 70], [33, 423]]
20	Tunne Model Logistic Regression	0.905232	0.905232	0.905232	0.905232	precision recall f1-score ...	[[2480, 103], [185, 271]]
21	Tunne Model Bagging Boosting classifier	0.964791	0.964791	0.964791	0.964791	precision recall f1-score ...	[[2545, 38], [69, 387]]
22	Tunne Model Gradient Boosting	0.952287	0.952287	0.952287	0.952287	precision recall f1-score ...	[[2553, 30], [115, 341]]

Summary of Model Performances

- We are comparing model performances here we see that only logistic regression give 76% recall and other are Gradient and Bagging classifiers give more than 90% recall.
- And in terms of accuracy that Bagging Classifier has the highest accuracy which is 96% and Gradient has 94% accuracy but Logistic has the lowest accuracy.
- In conclusion Bagging Classifier has good performance and now we make the final model through the Bagging Classifer.

FINAL MODEL

- We are using BAGGING CLASSIFIER as our final model as it is giving good accuracy recall and f1_score as we can seen and easily understand through the confusion matrix and by checking the classification report and ACCURACY score.
- **The model predicts 402 true negatives, 86 false positives, 37 false negatives, 2514 true positives**




```
'              precision    recall  f1-score   support\n\n           0       0.97      0.99      0.98      2551\n           1
0.92      0.82      0.87       488\n\n    accuracy                           0.96      3039\n   macro avg       0.94      0.90
0.92      3039\nweighted avg       0.96      0.96      0.96      3039\n'
```

A Confusion matrix is **an N x N matrix used for evaluating the performance of a classification model,** where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model.
The rows represent the predicted values of the target variable.
Now we understand from confusion matrix that almost 97% they give accuracy answer and only 3% it’s predicted answer is wrong.

SUMMARY AND RECOMMENDATIONS FOR THE FUTURE BUSINESS

- **Future improvements**
- Use correlation matrix in EDA to find the most influential features.
- Use iterative imputer to get rid of the "unknown" values?
- Use pca for feature selection.
- Create a training and inferencing pipeline.
- Data upsampling with adasyn instead of smote

- With the existing consumer insights through data, companies can predict customers' possible needs and issues, define proper strategies and solutions against them, meet their expectations and retain their business. Based on the predictive analysis and modeling, businesses can focus their attention with targeted approach by segmenting and offering them customized solutions. Analyzing how and when the churn is happening in customer's lifecycle with the services will allow the company to produce more preemptive measures.

CONCLUSION

- We can conclude that the top 3 most influential features are the product variables: "total_trans_ct", "total_trans_amt", "total_amt_chng_q4_q1".
- Using the existing data, we managed to train a model with upsampled data which reaches a recall score of 92%.
- As shown in confusion matrix previously by the TP/(TP+FN) accuracy output above we have a ~96% accuracy in predicting which users will be churned customers out of all users who were marked as churned. The next steps following this analysis will be to apply the trained model to the user base to see if the model marks any existing users as churned. Following you should monitor and record data of the status of the marked clients without intervention to understand if the model is accurately classifying potentially churned customers. Following a successful monitoring period, intervention techniques should be discussed, and which methods would prove beneficial. Then create sample groups for each type of intervention technique and record data to analyze which intervention technique is the most effective for which groups of individuals for effective targeted intervention in the future.
- Out of all the users who are predicted to be churned. Roughly 3% of them will be classified as existing users. This could be due to the margin of error because with any prediction.
- A presentation is formed with all the relevant information and explanations about the overall financial impacts and costs associated with the transition to the new model. Then executive leadership would decide on whether the provided information justifies the transition to the new model, or the executive leadership will request more research to be conducted about this aspect of the financial impact.
- With the existing consumer insights through data, companies can predict customers' possible needs and issues, define proper strategies and solutions against them, meet their expectations and retain their business. Based on the predictive analysis and modeling, businesses can focus their attention with targeted approach by segmenting and offering them customized solutions. Analyzing how and when the churn is happening in customer's lifecycle with the services will allow the company to produce more preemptive measures.
- **The model predicts 402 true negatives, 86 false positives, 37 false negatives, 2514 true positives.**