CAPSTONE PROJECT

# TELCOM CUSTOMER CHURN

11 APRIL 2021

Jotinder Singh Matta

PGPDSBA Online Mar20_A

# CONTENTS

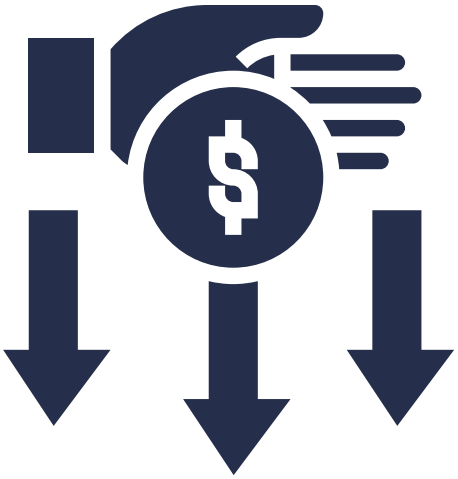
# 1. Introduction

## CUSTOMER CHURN

Telecom operator is concerned about the rising customer attrition rates. Given the highly competitive nature of the market and the fact that the entire telecom market is facing increasing churn rates, the operator wants to minimize the churn rate by recognizing the factors impacting it and formulating a proactive retention strategy.

## DROPPING ARPU

Along with the increasing customer churn, telecom as an industry is also facing the challenge of decreasing ARPU i.e. Average Revenue Per Unit. "Raves Saves" plays more important of a role in the current market, than it has ever before. Hence, even though predicting Churn is the primary objective, coming out with a strategy to arrest the drop in ARPU, might be a secondary objective, but an important one at that.



## REACTIVE ACTIONS SO FAR

All the actions by the telecom operator so far have been reactive in nature. Until a customer gives call to the customer care number, the telecom operator is oblivious to the possibility of that customer eventually churning. Telecom operator is looking for a way to proactively predict the customer that are likely to churn, at the same time also have a strategy to ,maximize ARPU.

# WHAT WE NEED TO DO?

Below are the customer expectations from this exercise.

**01**

### DETERMINE FACTORS

Determine top factors contributing towards customer churn.

**02**

### RETENTION STRATEGY

Recommend retention strategy to the telecom operator

**03**

### CUSTOMER SEGEMENTATION

Segement customers to identify high revenue generators from the low revenue generators.

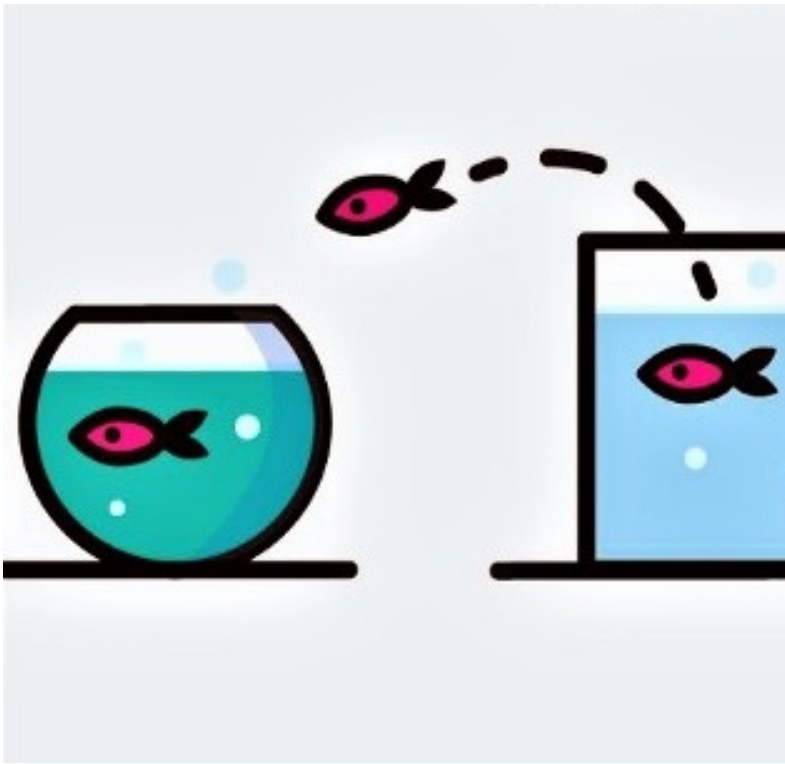
**04**

### REVENUE SAVES

Given the budget constraint, prioritize revenue saves apart from customer churn.

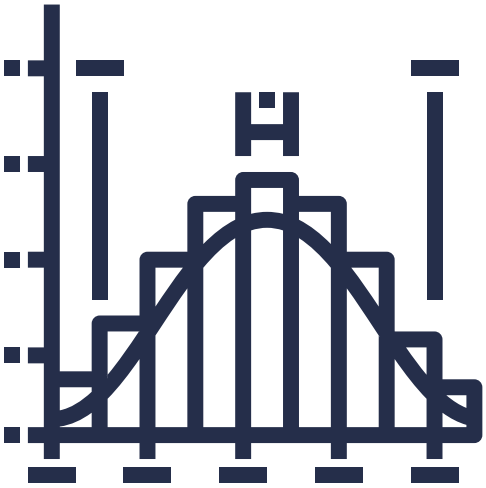
# APPROACH

We started with data analysis (EDA) and data cleaning, followed by building various classification models to predict customer churn and also clustering models to get insights of the customer segmentation. We then compared all the different classification models, keeping AUC / F1 Score as the criteria instead of accuracy due to the imbalanced nature of the given data set. Important factors were extracted from the most optimal model and the results were also correlated with the different customer segments. With various business insights all along the way, we finally end with a proactive retention strategy.



# 2. EDA and Business Implication



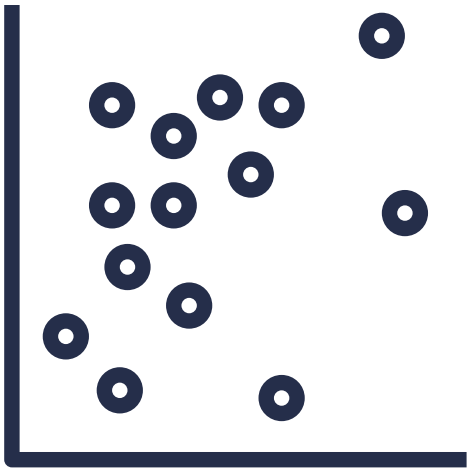


## UNI / BI - VARIATE ANALYSIS

Univariate analysis of individual column was done. Bar plot for categorical variables and distplot for continuous. Also a Bi-Variate analysis of all the variables was done. Correlation heatmap and Pairplots can be found later in the report. This was done to understand how the different real world factors (features) interact with each other.

## MULTI VARIATE ANALYSIS

Multivariate analysis was also done to understand these interactions. Strategies like VIF were employed to understand if a feature can already be derived by a combination of other features, and thus if required be removed all together. Chi-square test was done for categorical feature selection, to see if we could remove the features which were independent of the Churn variable.



## BUSINESS IMPLICATIONS

Multiple insights were generated during data analysis phase of the project. These have been mentioned throughout the report. Business insights like understanding if a customer from a particular credit class are likely to churn, amongst many others , can have huge business implications from altering the customer retention strategy, to an overhaul of the companies marketing and PR strategies.

# UNI / BI - VARIATE ANALYSIS

## UNI-VARIATE ANALYSIS

Univariate analysis was done on all the variables, categorical as well as continuous. Bar plots for categorical variables, and histograms for the continuous variable were plotted to understand their distribution. Below are graphs for some of the variables. For a complete list of graphs for Univariate analysis refer to the **Appendix 2.1** at the end of this report.

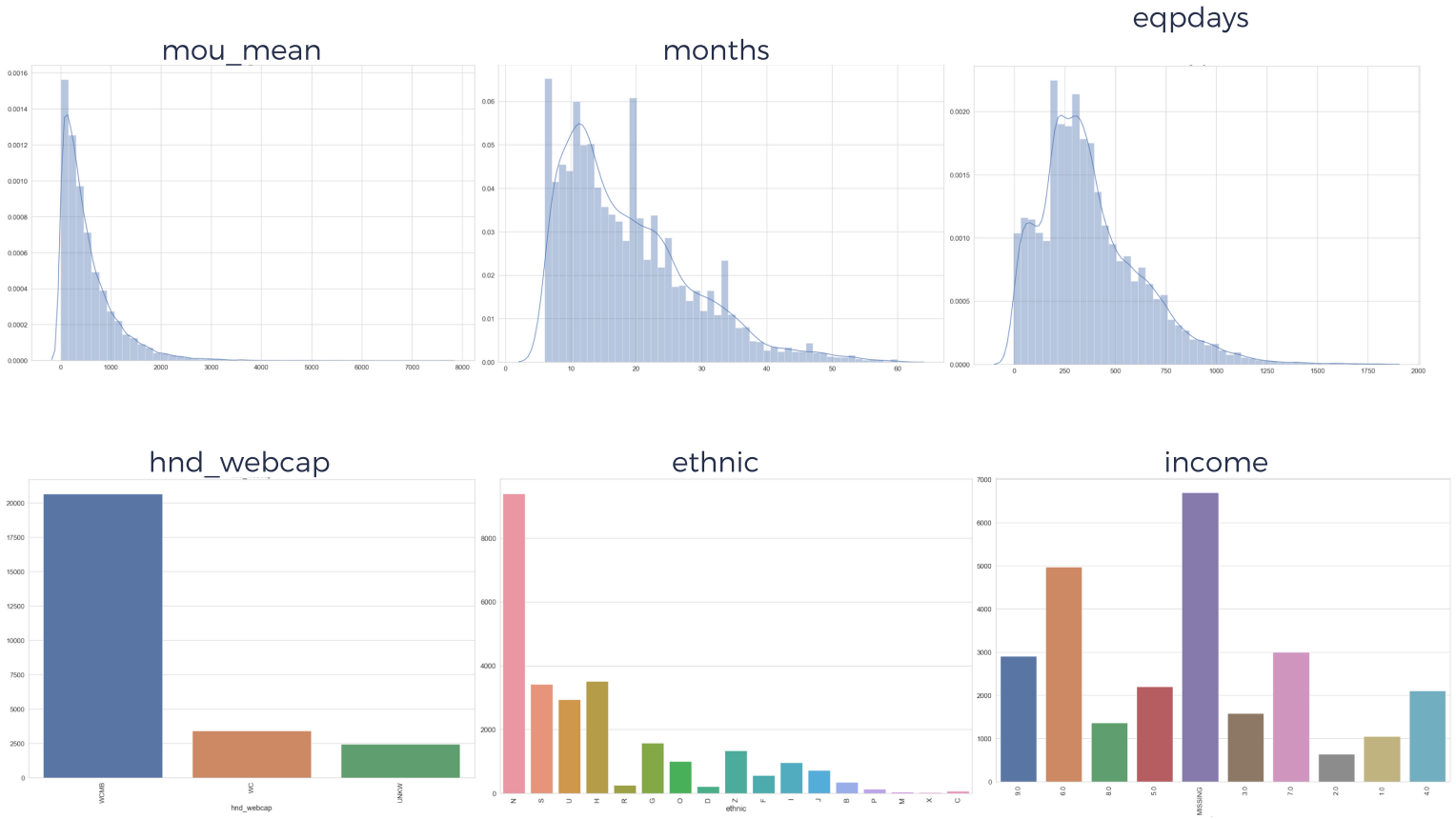



**Figure 2.1 Univariate Analysis**

We could see that most continuous variables are highly left skewed. For revenue variables we could see most of the customers generate very low revenue, while it is only a few who are the high revenue generator for the operator. Hence the importance of such customers is even so higher.

Also we could see a lot of customers are pretty new, many of them having been with the telecom operator for less than a year. However there are still a big chunk of customer who have stuck with the telecom operator even after 2 years and hence a loyalty program might be apt for such customers, or at least for the ones out of these who have high ARPU.

# UNI-VARIATE ANALYSIS

We also found that majority of the users have a handset that is between 1 - 2 years old . Keeping up to date with the network technologies that have been relevant in the past 1 - 2 years e.g. 4G and 5G, might be a good strategy for the telecom operator to follow. However at the same time there might be an incentive to still maintain older networking technologies like 2G/ 3G as we would see later in the report.

Looking at the variable hnd_webcap we found that most of the customers do have a web compatible handsets. Hence enabling features like Wi-Fi calling make more sense than ever. A simple software solution, if enabled can resolve the indoor network coverage issues, without much effort or investment.

The operator has a mix bag of customers, with many of them belonging to different ethnic groups. There were multiple variables having high dimensions in the data set. These were transformed to tackle the problem of high dimensionality as highlighted later in the report.

We also found a big portion of the customers belong to the middle class. Only a small section of the customers belong to the rich category, while there is a big amount of subscribers who belong to the lowest class as per the income variable.

Multiple other insights were drawn from this exercise which came in handy understanding the overall distribution of the data and were used in the rest of the exercise.

For a full list of graphs for all the variables, refer to **Appendix 2.1.**

# BI-VARIATE ANALYSIS

Bi-variate analysis was done on the data set, to find the correlations between various variables. It was done on both categorical as well as continuous variables.

Bivariate analysis of some of the categorical and continuous variables have been put here. However to see the entire analysis, refer to the **Appendix 2.2**.

Here we are going to cover few categorical and few continuous variables. We have done bivariate analysis of all the variables with the churn variable to understand the split along various columns. Also correlation heatmap and pair plot have been done to analyze the correlations between various columns.

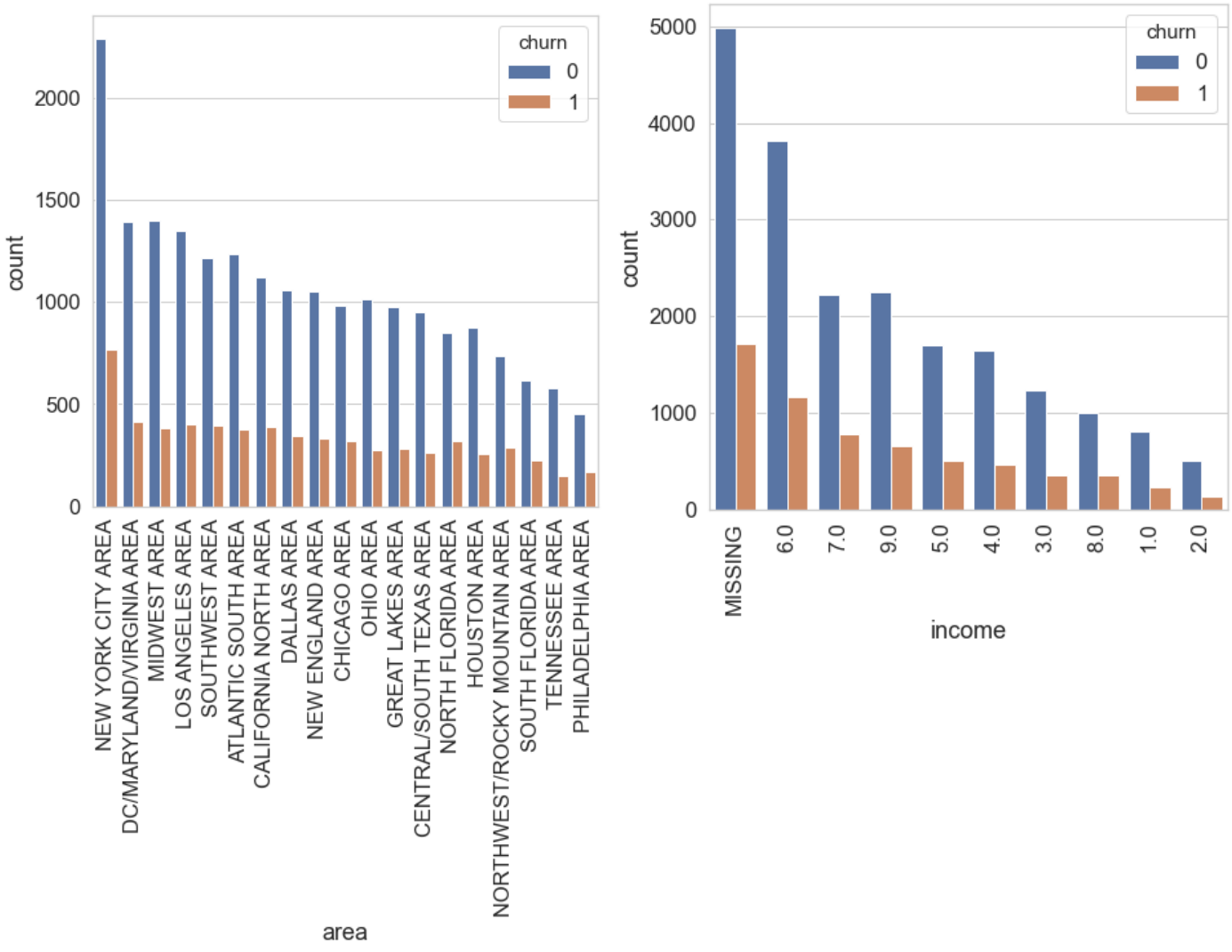



**Figure 2.2 Bivariate Analysis - Categorical Variables**

# BI-VARIATE ANALYSIS

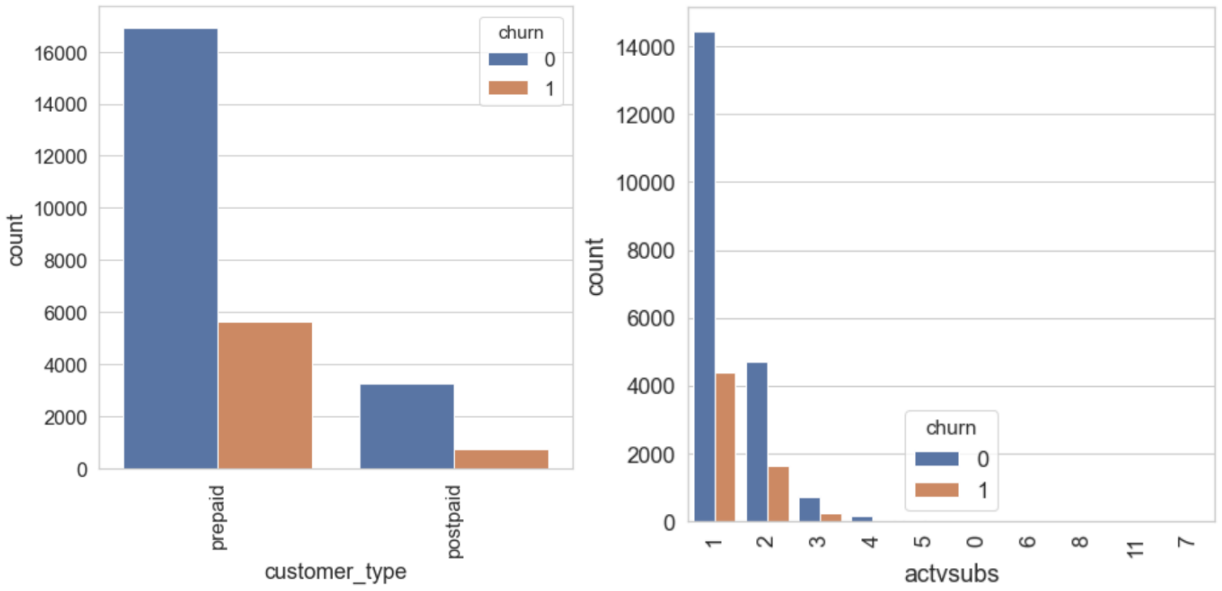



**Figure 2.3 Bivariate Analysis - Categorical Variables (contd.)**

We had many insights from the bi-variate analysis, few of them being.

We could see in **Figure 2.2** customers from New York City Area are churning the most when compared with other areas. This might be in part due to the higher number of customers in that area. But it could also mean the higher number of customers means more dense network coverage might be required to cater to consumer demands.

We identified a lot of missing values for many columns, one of them being income. Missing values were therefore treated before we moved to the model building. How these missing values were treated has been detailed later in the report.

We transformed the asl_flag column to customer_type (**Figure 2.3**), since account spending limit is set only for postpaid customers. This gave us the insight that a higher percentage of prepaid customers are likely to attrite compared to their postpaid counterparts.

Also we found from **Figure 2.3** that most of the households are having a single active subscriber and these are the ones who attrite the most. Where customers have more than one active subscriber in the household they are less likely to churn. This can have a big significance for the operator. People having multiple household members as active subscribers might be offered family plans in order to further decrease their chance of churning.

# BI-VARIATE ANALYSIS

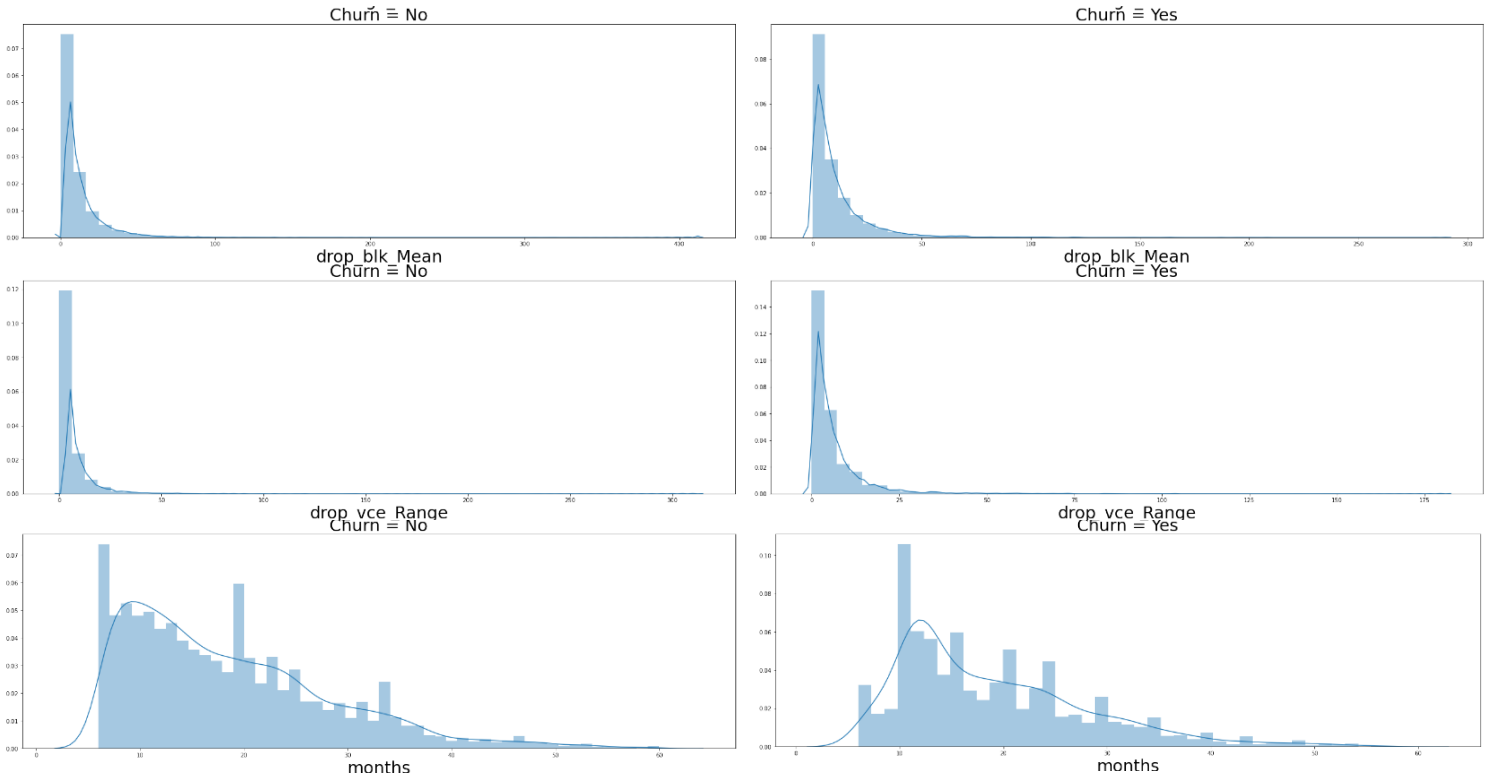



**Figure 2.4 Bivariate Analysis - Continuous Variables**

We also compared all the continuous variables with the response variable i.e. Churn. Some of them can be found in **Figure 2.4**

Although we did not find any significance difference between the customers who would churn vs the those who would not. However there still a few findings that might come in handy.

One of them being that the customers who churn are facing slightly more dropped calls compared to the non churning customers. Network quality improvement might be one of the areas that the operator might want to concentrate on. We will look deeper into this aspect later.

We also figured that the customers who churn, do so after having spent almost a year with the telecom operator. Most customers attrite between 1 to 2 years. These are the customers that the operator would want to stop from churning.

A complete list of graphs for both categorical as well as continuous variables can be found in **Appendix 2.2.**

# BI-VARIATE ANALYSIS

## CORRELATION

Correlation heatmap was drawn. We could see many of the variables within the data set were correlated. This can present the problem of multi-collinearity when we try to build the models. Hence we performed tried to drop the columns which are highly correlated and then built the models. We also tried building the models without dropping these columns. As you would see in further in the report, even though a lot of columns were correlated there was an incentive to include these columns in the model building as was clearly evident from the model performance evaluation using various metrics like recall, precision, F1 score etc.

A heatmap of the correlation can be found in **Appendix 2.3.**

Below are some of the columns for which correlation was found to be above 90%, shown in **Table 2.1.**

| Column1 | Column2 |
|---|---|
| mou_Mean | avg3mou |
| mou_Mean | avg6mou |
| totcalls | adjqty |
| totcalls | adjmou |
| adjqty | totcalls |
| adjqty | adjmou |
| ovrrev_Mean | ovrmou_Mean |
| ovrmou_Mean | ovrrev_Mean |
| comp_vce_Mean | plcd_vce_Mean |
| plcd_vce_Mean | comp_vce_Mean |
| avg3mou | mou_Mean |
| avg3mou | avg6mou |
| avgmou | avg6mou |
| avg3qty | avgqty |
| avg3qty | avg6qty |

**Table 2.1 Columns with greater than 90% correlation.**

For complete list refer to **Appendix 2.3**

# BI-VARIATE ANALYSIS

## PAIRPLOT

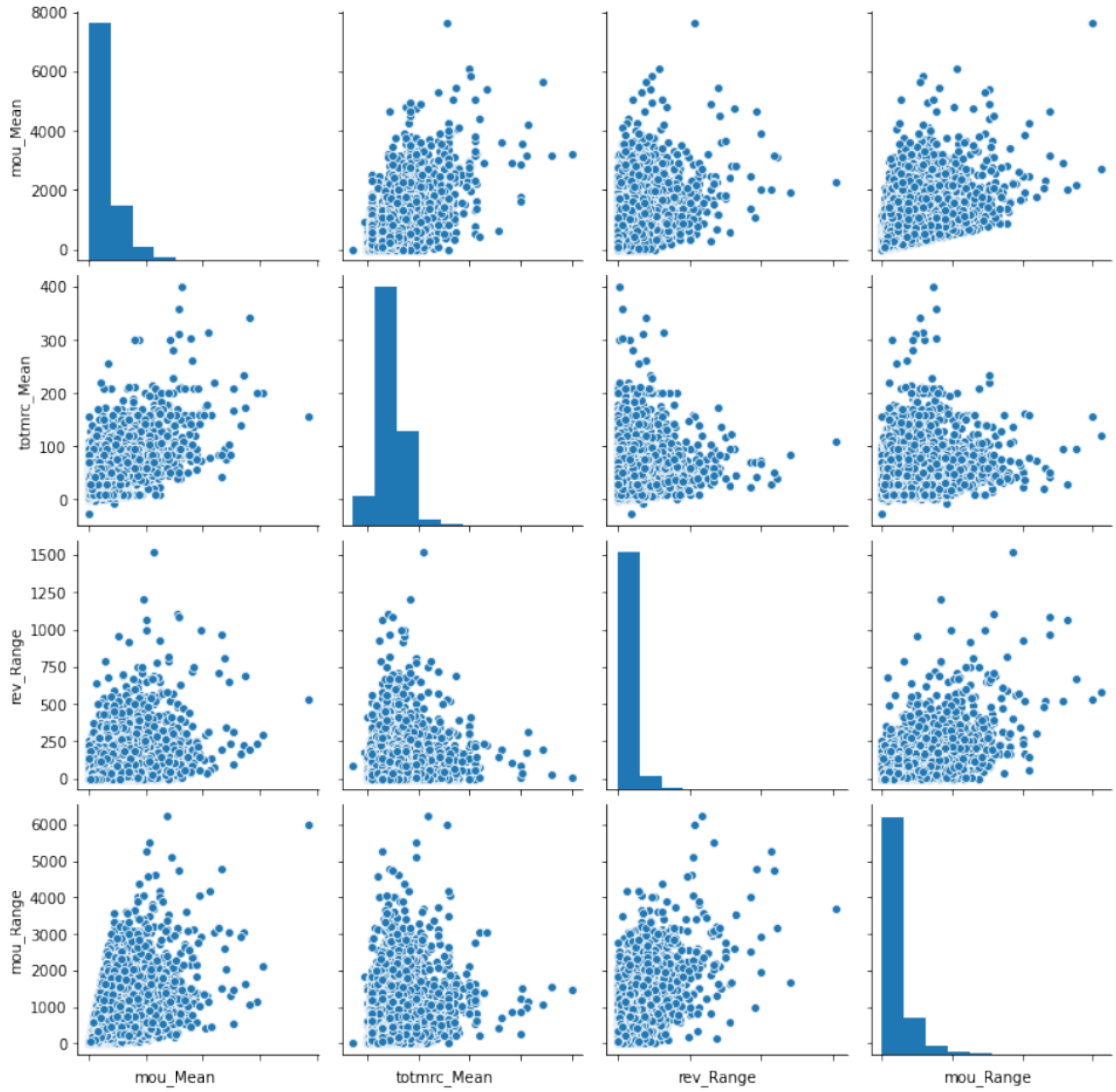



**Figure 2.5 Pairplot for continuous variables (Subset)**

Scatter plot of each continuous variable with the other was plotted as well.

In the **Figure 2.5** above we have a subset of the same, refer to **Appendix 2.4** for all the scatter plots.

Scatter plot helped us understand the interactions between a variable with all the other variables individually. We did find linear patters for a lot of them. Linear patterns were observed between various revenue related columns, which was expected, given the nature of the data set. Linear relations were high between columns who were found to be highly correlated as per the heatmap earlier, thus solidifying our understanding.

# MULTI-VARIATE ANALYSIS

Apart from doing the uni-variate and bi-variate analysis, we also performed multi-variate analysis. Multivariate analysis was done to understand the interactions between various combination of columns.

Strategies like VIF were employed to understand if a feature can already be derived by a combination of other features, and thus if required be removed all together.

Chi-square test was done for categorical feature selection, to see if we could remove the features which were independent of the Churn variable.

## VIF (VARIATION INFLATION FACTOR)

VIF analysis was done on the data to see if there are any continuous variables that can be dropped if they can be derived from a linear combination of other variables. A VIF threshold of 5 was used for this. As a result we were able to drop below columns, which could have been derived using a linear combination of other columns in the data set.

Below columns were initially dropped as a result of the VIF analysis.

**['rev_Mean','avgrev','avgmou','avgqty','mou_Mean','totrev','plcd_vce_Mean','months','mou_Range', 'drop_vce_Mean']**

However after building a model with the dropped columns due to high VIF, we found that the overall recall and precision of the model dropped. Hence there was an incentive to retain these columns. Hence during the final model building exercise this step was not taken into consideration.

Refer to the code used for VIF analysis in **Appendix 2.5.**

# MULTI-VARIATE ANALYSIS

## CHI-SQUARE TEST

Chi Square test was done for categorical variables and it was found that a few variables were independent of the churn variable and hence would not have contributed to the model building process. Hence these were removed. An hypothesis test with alpha = 0.05 was done where the hypothesis was as below.

H0 --> cat_variable and churn are independent
Ha --> cat_variable and churn are not independent

| ColumnName | Chi_Value | p_Value |
|---|---|---|
| forgntvl | 0.100936 | 0.750710 |
| mtrcycle | 0.574078 | 0.448643 |
| truck | 0.515877 | 0.472606 |
| car_buy | 3.333988 | 0.067862 |

**Table 2.2 Columns that are independent of the churn variable.**

Above columns in **Table 2.2** had p value greater than alpha, hence we failed to reject the null hypothesis for them and they were indeed independent of the target variable i.e. churn.

We then tried to drop these columns and compare the resulting model with the base model (with all columns) and we found that even though some of the variables are independent of the response variable i.e. Churn, they still add value when it comes to predicting the churn as was evident from the model performance evaluation.

Find the chi-square test code and results for all columns in **Appendix 2.6.**

# BUSINESS IMPLICATIONS

Various business implications have been discussed in the report while the data analysis was being discussed. However for the sake of ease and clarity, these have been mentioned briefly below as well.

Data was found to be highly skewed, indicating the distribution of customers hence allowing us to separate high revenue generators from the rest.

Customer's relationship with telecom operator was found to be relatively new. It was observed that a lot of customers are churning between their 1st and the 2nd year with the telecom operator. Loyalty programs can be offered in order for a customer to have an inventive to stay longer with the operator.

We found customers generally have a handset between 1 - 2 years old. Hence keeping up to date with network technologies was critical for the operator, while also keeping in mind older technologies for some of the customers.

Most customers have new handsets, hence having solutions like Wi-Fi calling might be a boon to resolve indoor network coverage issues.

It was found that customers from New York City region are churning the most, which might be due to the customer density and the operator might want to tweak its network coverage accordingly.

Percentage of prepaid customers churning is much higher than their postpaid counterparts.

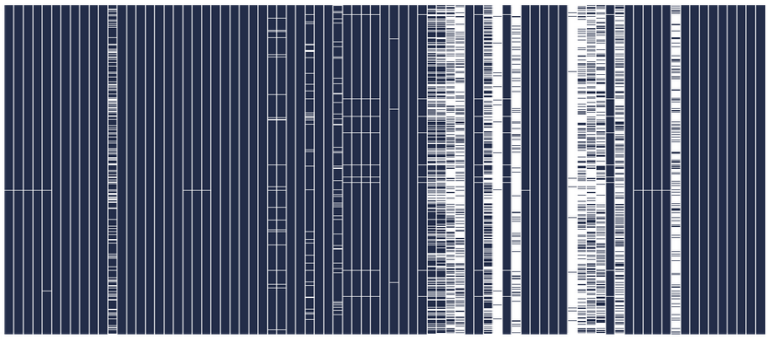
Family plan could be an option to reduce customer churn, since subscribers having more than 1 active subscriber in the household are less likely to churn.

We could see that the customers who had churned had higher number of dropped voice calls, hence another indicator for the operator to work on its network coverage.

We found high correlation between various columns and could hence conclude that making a single change in strategy can have multi fold effects on the customer behavior.

# 3. Data Cleaning and Pre-processing



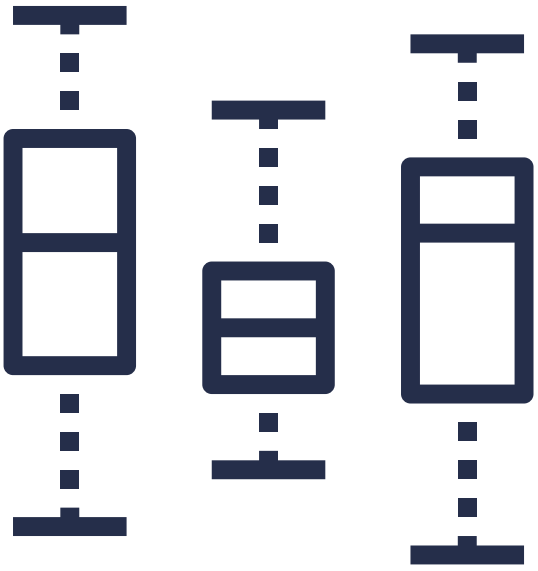


## MISSING VALUE ANALYSIS

A large chunk of values were missing for multiple columns in the data set. Columns with more than 30% missing value were dropped, while for the columns with less than 3% missing values, we dropped the corresponding rows. Data was imputed for the rest.

## OUTLIER ANALYSIS

We observed the data set to be highly skewed Most of the continuous variables were found to be highly left skewed. In order to reduce the skewness of the data set, we performed Outlier treatment. Values that had more than +/- 3 standard deviation were dropped. We noticed a remarkable reduction in overall skewness after this step. Also the model performance compared against the base model increased after performing outlier treatment.



## VARIABLE TRANSFORMATION

We transformed various variables over the course of this exercise. E.g. asl_flag variable was converted to customer_type, since account spending limit is set only for postpaid customer. We also encoded multiple categorical variables using OneHot and Label encoding techniques. Scaling of the variables was done wherever necessary.

# MISSING VALUE TREATMENT

A total of 42 columns out of the total 81 had missing values. We even had certain columns with barely any information like solflag which had just 2% of the values populated or retdays which had less than 4% of the values populated in the data set.

In the light of these findings we split or missing value treatment strategy into three parts

**> 30%**

### Missing Values Greater than 30%

For the columns having missing values greater than 30%, we went ahead and dropped these columns, as there was too much of a missing information to be used in the modelling or to be even imputed. 13 columns were dropped. List of columns can be found in **Appendix 3.1.**

**< 3%**

### Missing Values Less than 3%

For the columns which had less than 3% missing value, since they were within the industry accepted limits, we dropped their corresponding rows. A total of 875 rows were dropped in this exercise. For the list of these 45 columns, refer to **Appendix 3.1.**

**BETWEEN 3 & 30%**

### Missing Values Between 3 & 30%

For the columns where missing values were between 3 & 30% we went for imputation

- Categorical
  - MISSING Keyword
- Continuous
  - Linear Interpolation (both directions).

# MISSING VALUE TREATMENT

Below in **Table 3.1** is the list of columns that were imputed using the previously mentioned strategy for columns having between 3 & 30% missing values.

| | Missing % |
|---|---|
| income | 25.25 |
| hnd_webcap | 8.98 |
| prizm_social_one | 7.09 |
| avg6qty | 3.07 |
| avg6mou | 3.07 |

**Table 3.1 Columns having between 3 & 30% missing values.**

Below is the aerial view of data before and after missing value treatment.

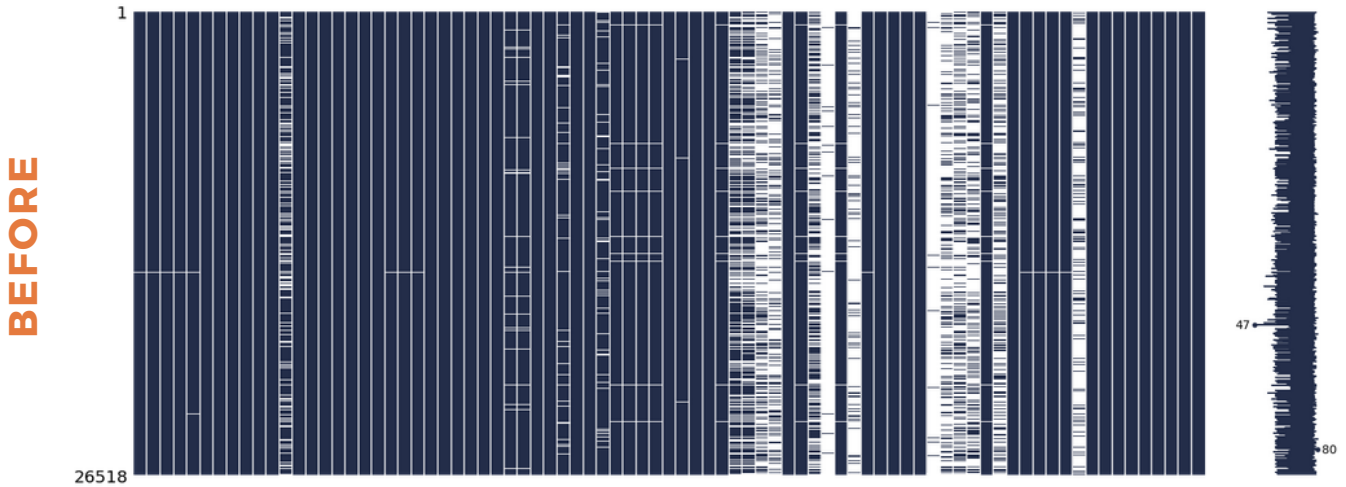



**Figure 3.1 Data before Missing values treatment**

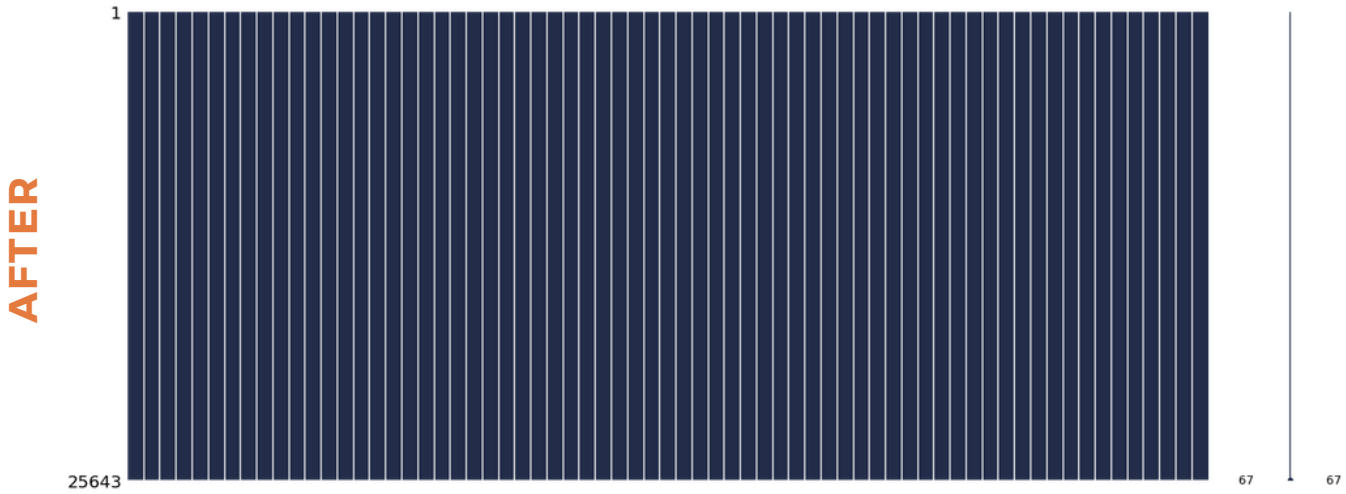



**Figure 3.2 Data after Missing values treatment**

# OUTLIER TREATMENT

We observed the data set to be highly skewed Most of the continuous variables were found to be highly left skewed. In order to reduce the skewness of the data set, we performed Outlier treatment.

Values that had more than +/- 3 standard deviation were dropped. We noticed a remarkable reduction in overall skewness after this step. Also the model performance compared against the base model increased after performing outlier treatment.

Some columns had higher number of outliers compared to the others. We did the analysis and found roam_Mean to have the highest number of outliers at 4705, which was 18.34% of the entire data set, while recv_sms_mean had the lowest number of outliers at 215, which is just under 1% of the data set.

In **Table 3.2** are the top 10 columns having outliers. For the entire list of columns with their outlier percentage as well as the code used to generate this data, refer to **Appendix 3.2.**

| | Outlier Count | Percent Outlier |
|---|---|---|
| roam_Mean | 4705 | 18.348087 |
| plcd_dat_Mean | 3990 | 15.559802 |
| callwait_Mean | 3831 | 14.939750 |
| datovr_Mean | 3595 | 14.019421 |
| datovr_Range | 3594 | 14.015521 |
| change_mou | 3557 | 13.871232 |
| comp_dat_Mean | 3556 | 13.867332 |
| custcare_Mean | 3461 | 13.496861 |
| ccrndmou_Range | 3244 | 12.650626 |
| ovrmou_Mean | 2983 | 11.632804 |

**Table 3.2 Top 10 columns having highest outlier count**

# OUTLIER TREATMENT

Below are a few of the columns and their distribution plot before and after outlier treatment. We can see the skewness is clearly reduced and it is easier to make inferences from the distribution.

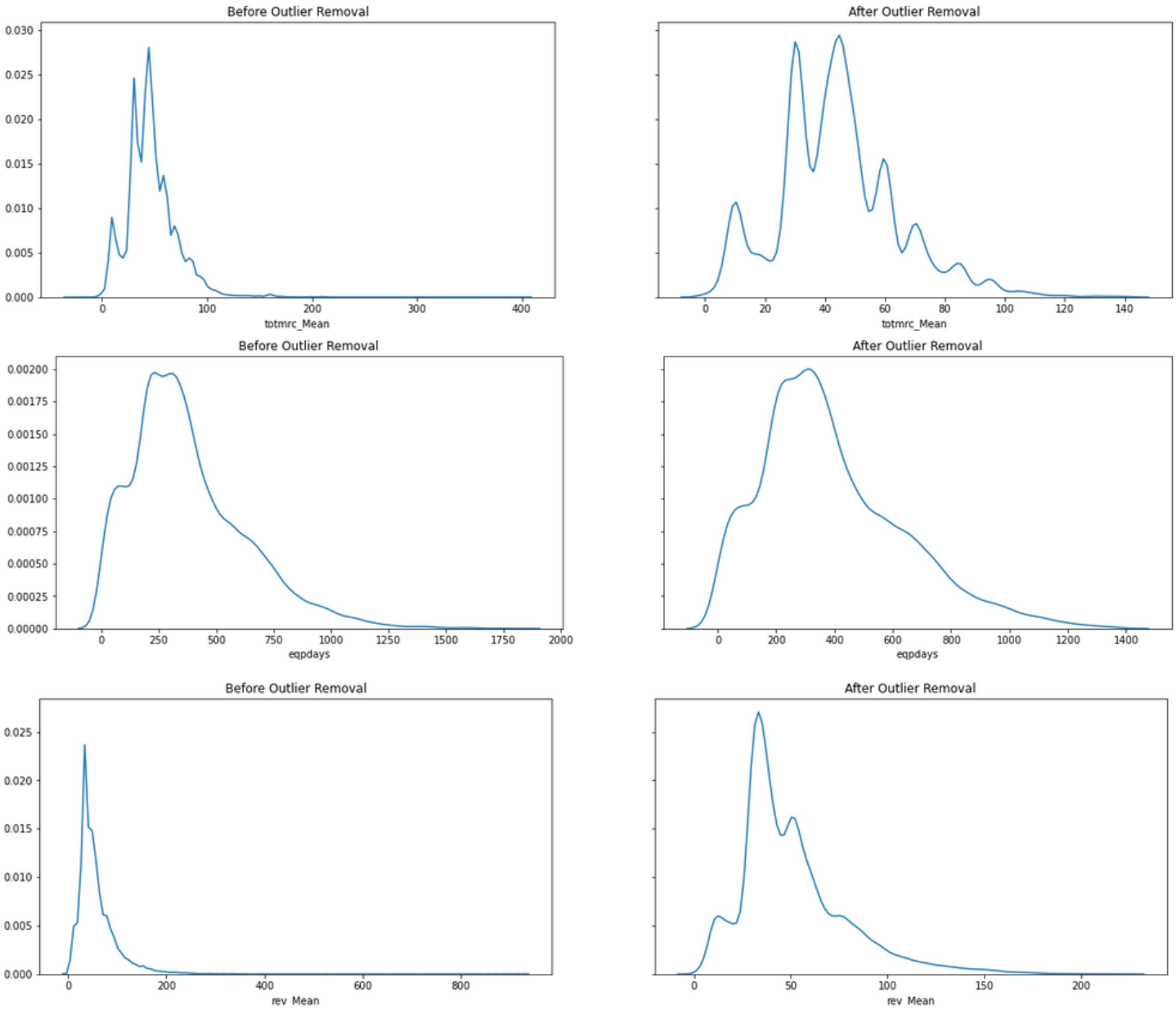



**Figure 3.3 Distribution of data before & after outlier treatment**

Refer to the complete list of graphs in **Appendix3.2.**

# VARIABLE TRANSFORMATION

Multiple variables were transformed as part of this exercise. Some of the categorical variables were bucketed, while some variables had to be encoded using techniques like OneHot Encoding and Label Encoding.

Variable asl_flag was renamed to customer_type, since account spending limit flag is applicable only for postpaid customers, hence this gave us the split of postpaid vs. prepaid customers.

Various columns were bucket to manage the curse of dimensionality. There were many categorical columns which had to be bucketed in order to reduce the over all dimension of the data set once they were encoded using OneHot Encoding.

Some of the columns which were bucketed include.

01 CRCLSCOD

02 CSA

03 ETHNIC

04 ACTVSUBS

05 UNIQSUBS

Analyzing individually for each of the columns, we decided on the top N number of values to retain, while all the values below the threshold were clubbed together into a new category named OTHERS.

We were able to reduce dimensionality of the data set significantly using this technique. For some of the columns like csa the cardinality was reduced from 679 to just 61. A massive drop!

For some of the other columns like actvsubs and uniqsubs, the reduction was subtle but still helpful.

Find the list of all the columns that were bucketed along with the counts for each of the unique values in **Appendix 3.3.**

# VARIABLE TRANSFORMATION

We created two data sets, one for tree based models and other for distance based model. Variable transformation was done differently for both of them. Variables were identified to be boolean, ordinal or nominal in nature and accordingly a transformation strategy was chosen.

## Tree Based

- Boolean Variables
  - Label Encoded
- Ordinal Variables
  - Label Encoded
- Nominal Variables
  - Label Encoded
- Total Columns
  - 67 Columns

## Distance Based

- Boolean Variables
  - Label Encoded
- Ordinal Variables
  - Label Encoded
- Nominal Variables
  - OneHot Encoded
- Total Columns
  - 177 columns

Nominal variables were OneHot encoded for distance based algorithms. E.g. 9 will be treated as a higher value than 1 when it comes to distance based algorithms, which is not the case with nominal data, where there is no inherent ordering, hence we chose to go with OneHot encoding using pandas dummies function.

# VARIABLE TRANSFORMATION

Apart from the above, we also created separate data sets using sklearn's standard scaler. This was done so that standardized data sets can be used for some of the models which are sensitive to scaling e.g. SVM, KMeans etc.

On the other hand for scaling agnostic models like Logit and Random Forest etc. we used unscaled data set.

SMOTE data set was also created since this was a imbalanced data set. Again, smote data set were also standardized and used for the algorithms accordingly.

We created multiple models with various permutation and combinations of different data sets and compared them with each other to evaluate their performance.

# 4. Model building

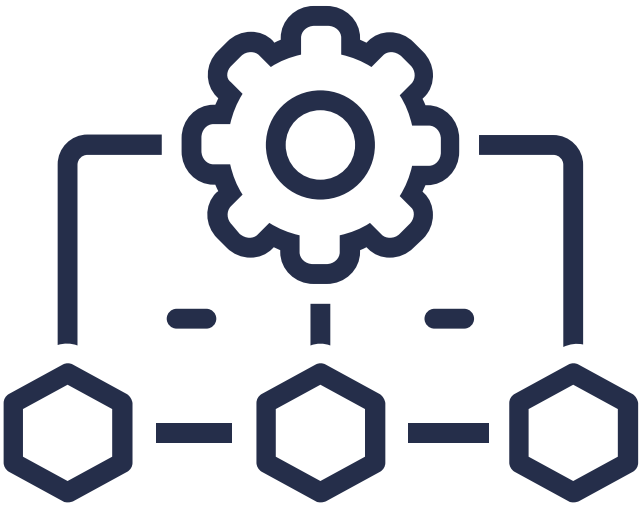


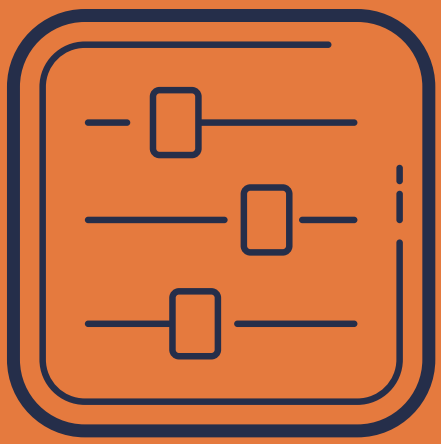
## APPROACH

We have created multiple models and applied them on different sets of data as required. All the different models which were created, were then evaluated using the AUC / F1 score at the end for the testing data set. Based on this an optimal model was chosen. Eventually we also found the feature importance for the most optimal model.

## MODELS BUILT

Various tree based as well as distance based models were built as part of this exercise using the different data sets as elaborated earlier. These models were built using sklearn and statsmodel libraries. There were various constraints, biggest one being the Type 2 error, which we had to minimize, as churn variable was the main objective of this exercise. This will discussed in detail later in the report.





## MODEL TUNING METHODS

Various model tuning approaches were followed. Primarily we made use of GridSearchCV function with cv = 3 for model hyper parameter tuning. Also we had to tweak the threshold values to maximize the recall values. Threshold tweaking was required as we had a typical problem of recall precision trade off.

# APPROACH

Various different approaches were followed to create multiple models. As mentioned earlier we had created multiple data sets like tree, tree_scaled, tree_smote, tree_smote_scaled, linear, linear_scaled,linear_smote, linear_smote_scaled etc.

We have also creating two generic functions which will be used to evaluate various models and also to tweak their threshold to maximize the recall.

## APPLY_EVAL

1

### USAGE

This is used to train the model, apply the model on test set and then output all the performance metrics like confusion matrix, Classification report, AUC curve etc.

2

### LOGIC

X_train, X_test,y_train & y_test are input to the function along with the model and param grid for GCV. model is trained, tuned then validated against the test set and performance metrics are generated.

## TWEAK_THRESHOLD

1

### USAGE

This is used to tweak the threshold, once the best model has been selected after hyper parameter tuning. Threshold is tweaked to maximize the recall.

2

### LOGIC

Threshold tweaking is done by calculating performance metrics like recall for all the values of probabilities between 0 and 1, and a step size of 0.1. Threshold with best AUC score is selected.

Refer **Appendix 4.1** for full code for both these functions.

# APPROACH

## TREE / LINEAR

- We used two data sets.
  - Tree
  - Linear
- Tree - For Tree based models like CART, Random Forest etc.
  - 67 Columns
- Linear - For distance based models like Kmeans, LDA etc.
  - 177 Columns

## ENSEMBLE MODELLING

- Various ensemble models were also used apart from regular models.
- Both Bagging and Boosting approaches were tried, evaluated and compared to determine the best model for our purpose.

## SCALED / UNSCALED

- Some of the models were sensitive to scaling e.g. SVM, KMeans etc.
- On the other hand we had models like Logit and other tree based models which are scaling agnostic, we used unscaled data set there.

## UNBALANCED DATA SET

- Since the data set was highly in favor of non churners.
  - 76% Non Churners.
  - 24% Churners
- SMOTE data set was also scaled wherever required for a given model.
- Helped with better recalls and accuracy.

## RECALL - PRECISION TRADE OFF

- There was a typical problem of recall-precision trade off with this data set.
- Increase in Precision would lead to loss of accuracy.
- And vice versa.

## REDUCE TYPE II ERROR

- Since the focus of this study was to emphasize on the customers who would churn i.e. class 1.
- The focus was to reduce Type II error, i.e. to Avoid False Negatives.
- We chose to consider recall as the primary evaluation criterion.

# MODELS BUILT

We have created multiple models as part of the telecom churn prediction project. The models include descriptive models like KMeans where we try to segment the customers and gain insights and also predictive classification models like Random Forest, Gradient Boosting model, Logistic regression in order to predict customer churn.

Combined they can provide prescriptive analysis to the telecom operator and help them with the retention strategies.

Various permutation and combinations were tried for various models.
**Table 4.1** contains the list of all the models that were created as part of this exercise.

| | Model_Name | | Model_Name |
|---|---|---|---|
| 0 | best_model_ADA_tree_scaled | 13 | best_model_XGB_tree_scaled |
| 1 | best_model_ADA_tree_smote_unscaled | 14 | best_model_XGB_tree_unscaled |
| 2 | best_model_ADA_tree_unscaled | 15 | best_model_lda_linear_unscaled |
| 3 | best_model_GBM_tree_scaled | 16 | best_model_logit_linear_scaled |
| 4 | best_model_GBM_tree_smote_scaled | 17 | best_model_logit_linear_smote_scaled |
| 5 | best_model_GBM_tree_smote_unscaled | 18 | best_model_logit_linear_smote_unscaled |
| 6 | best_model_GBM_tree_unscaled | 19 | best_model_logit_linear_unscaled |
| 7 | best_model_NB_linear_scaled | 20 | best_model_smote_lr |
| 8 | best_model_NB_linear_smote_scaled | 21 | best_model_smote_rf |
| 9 | best_model_NB_linear_smote_unscaled | 22 | best_model_smote_scaled_lr |
| 10 | best_model_NB_linear_unscaled | 23 | best_model_smote_scaled_lr_saga |
| 11 | best_model_RF_tree_smote_unscaled | 24 | best_model_svm_linear_scaled |
| 12 | best_model_RF_tree_unscaled | 25 | best_model_svm_linear_unscaled |

**Table 4.1 List of all the models that were built**

All the models were created using the apply_eval generic function and their thresholds were tweaked using tweak_threshold function. Code for these is present under **Appendix 4.1**

# MODELS BUILT

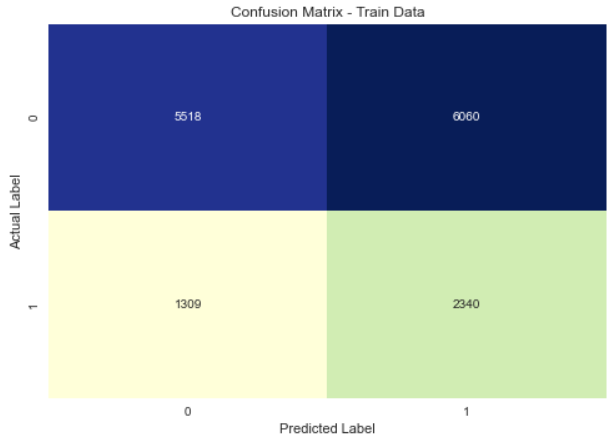
On the following pages you will find the performance metrics of some of the models that we have built. Performance metrics will be discussed in detail shortly under model validation section..

Below are some of the performance metrics that we calculated for **Gaussian Navie Bayes** ML model.

Gaussian Navie Bayes - Performance Metrics

**Train Data**

Accuracy Score (Train Data) - 52 %




```
--------------------Classification Report - Train Data-------
              precision    recall  f1-score   support

           0       0.81      0.48      0.60     11578
           1       0.28      0.64      0.39      3649

    accuracy                           0.52     15227
   macro avg       0.54      0.56      0.49     15227
weighted avg       0.68      0.52      0.55     15227

------------------------------------------------------------
AUC: 0.580
```

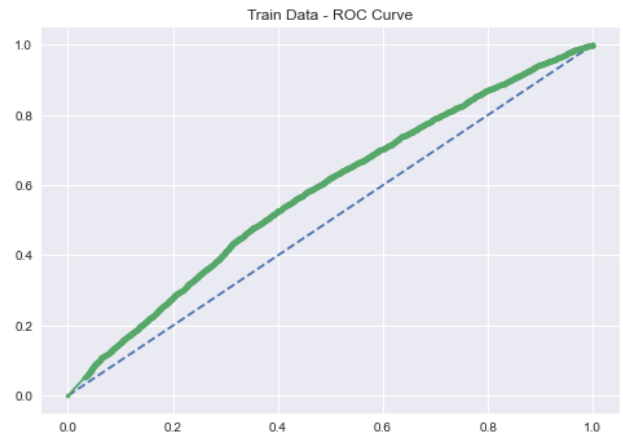
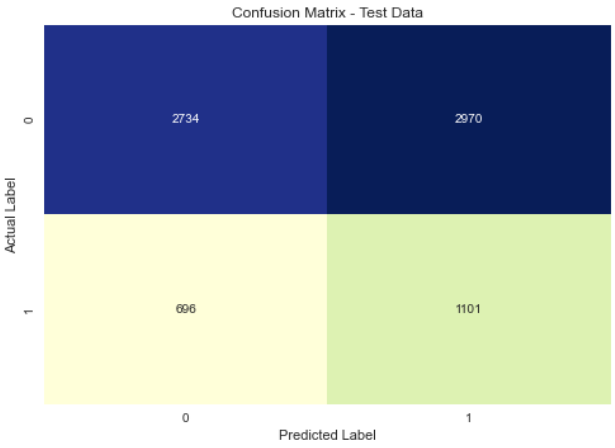



**Figure 4.1 Performance metrics for GNB (Train)**

**Test Data**

**Accuracy Score (Test Data) - 51 %**




```
--------------------Classification Report - Test Data-------
              precision    recall  f1-score   support

           0       0.80      0.48      0.60      5704
           1       0.27      0.61      0.38      1797

    accuracy                           0.51      7501
   macro avg       0.53      0.55      0.49      7501
weighted avg       0.67      0.51      0.55      7501

------------------------------------------------------------
AUC: 0.551
```

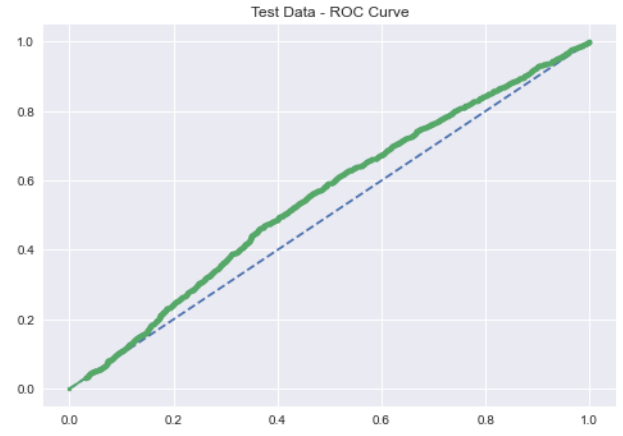



**Figure 4.2 Performance metrics for GNB (Test)**

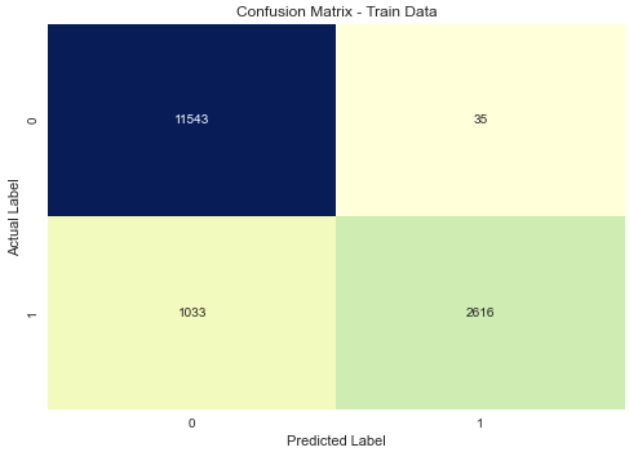
# MODELS BUILT

Below are some of the performance metrics that we calculated for **XGBoost** ML model.

XGBoost - Performance Metrics

## Train Data

Accuracy Score (Train Data) - 93 %




```
--------------------Classification Report - Train Data------
              precision    recall  f1-score   support

           0       0.92      1.00      0.96     11578
           1       0.99      0.72      0.83      3649

    accuracy                           0.93     15227
   macro avg       0.95      0.86      0.89     15227
weighted avg       0.93      0.93      0.93     15227

------------------------------------------------------------
```

AUC: 0.989

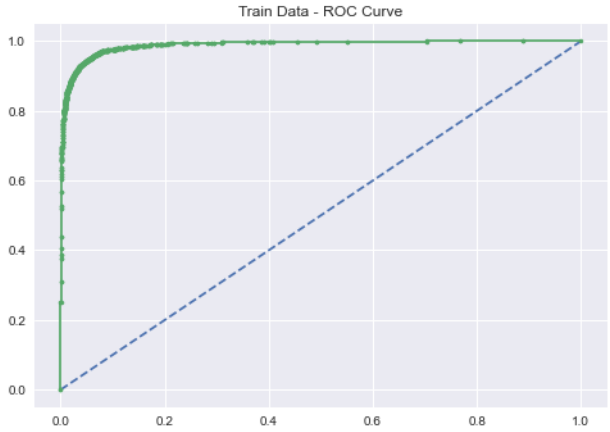
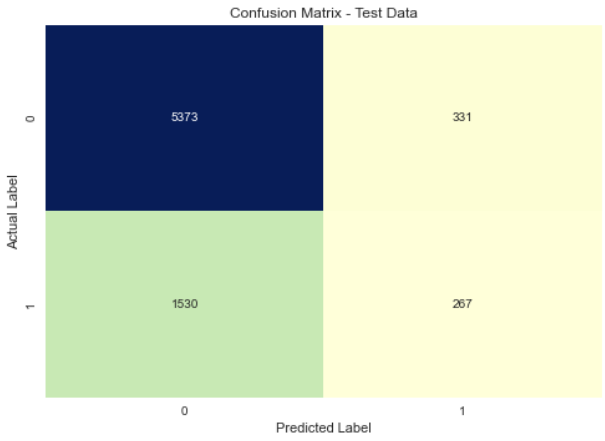



**Figure 4.3 Performance metrics for XGB (Train)**

## Test Data

**Accuracy Score (Test Data) - 75 %**




```
--------------------Classification Report - Test Data-------
              precision    recall  f1-score   support

           0       0.78      0.94      0.85      5704
           1       0.45      0.15      0.22      1797

    accuracy                           0.75      7501
   macro avg       0.61      0.55      0.54      7501
weighted avg       0.70      0.75      0.70      7501

------------------------------------------------------------
```

AUC: 0.642


**Figure 4.4 Performance metrics for XGB (Test)**

# MODELS BUILT

We had created KMeans as our descriptive model. Hierarchical model was also created as part of this exercise. For KMeans we have choose 4 clusters based on the silhouette scores and 2d pca rendition of the data. Below were the clusters that were created.

Entire code for this model can be found in **Appendix 4.2.**

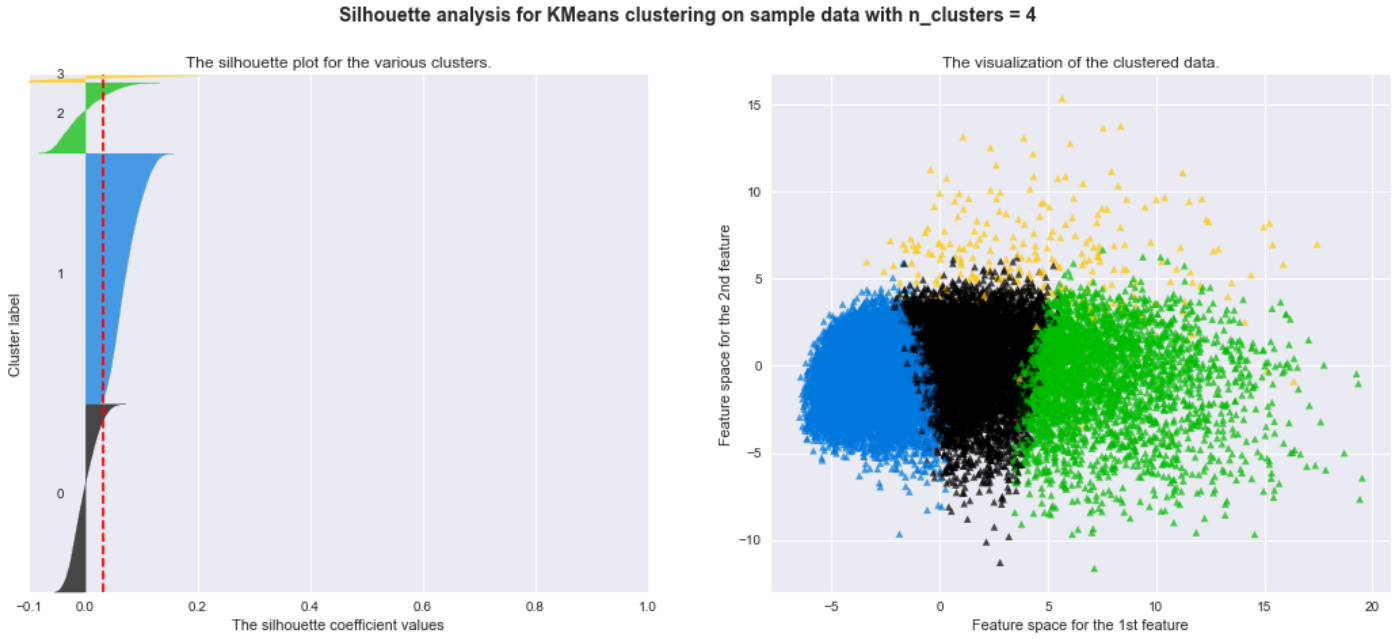



**Figure 4.5 KMeans model clustering output**

| kmeans_clus | mou_Mean | totmrc_Mean | rev_Range | mou_Range | change_mou | drop_blk_Mean | drop_vce_Range | owylis_vce_Range | mou_opkv_Range | months |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | -0.706247 | -0.499828 | -0.440138 | -0.580773 | 0.049378 | -0.514163 | -0.427896 | -0.463529 | -0.460723 | 0.112729 |
| 1 | 0.084829 | 0.098761 | 0.124341 | 0.198969 | 0.037454 | 0.086046 | 0.136379 | 0.129637 | 0.175148 | -0.549267 |
| 2 | 1.802938 | 0.916181 | 0.993910 | 1.254910 | -0.235973 | 1.356323 | 0.965626 | 1.075373 | 1.091527 | -0.200559 |
| 3 | 0.314689 | 0.460167 | 0.155764 | 0.179792 | -0.018555 | 0.136733 | 0.118073 | 0.141263 | 0.018735 | 1.044745 |

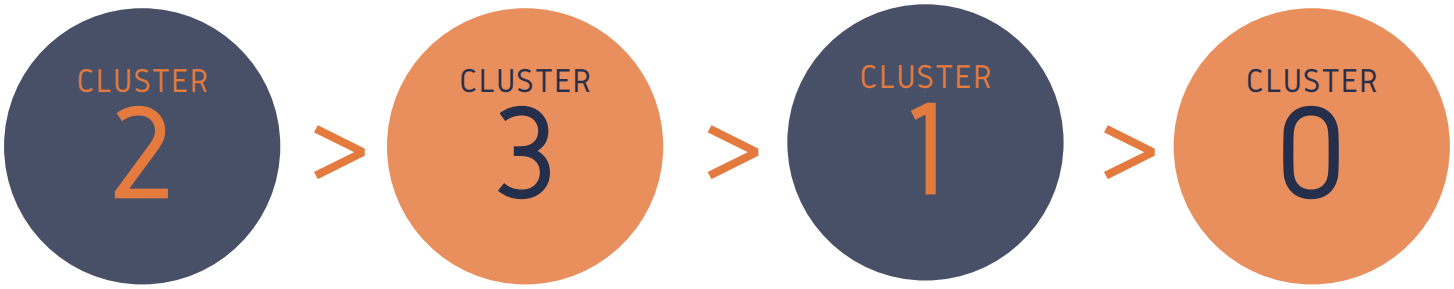
**Table 4.2 KMeans Cluster profile**

From the above cluster profile we can segment the customers into premium and regular categories. Clearly customers belonging to cluster 2 are the most premium customers for the telecom company and hence a good target audience to retain. As these help a lot in increasing the ARPU for the company. **Customers in cluster 2 have a rev_Mean of 1.39,**

While customers in cluster 0 are the lest paying users with old handsets and minimum revenue generation for the company. There is not much incentive in retaining these customers as company would not be able to recover the cost of retention from these customers for a long time. **Customers in cluster 0 have a rev_Mean of -0.59**

# MODELS BUILT

Below is the sequence of these clusters from most important customer segment to the least important.



# MODEL TUNING METHODS



## Ensemble Modelling

### Ensemble Modelling

We have used various ensemble models in our analysis. Some of them being Random Forest, XGBBoost, Gradient Boosting Model, ADABoost etc.

Ensemble model use multiple weak learners to come up with a better prediction than a single strong learner in a stand-alone manner.

Ensembling was chosen to increase the overall accuracy and recall values as even the strong models were suffering to get decent figures.

## Threshold Tuning

### Threshold Tuning

This is a measure by which we tweaked the threshold value of probabilities based on which the records are classified as 0 and 1.

By tuning the threshold we can greatly impact the recall, precision and accuracy values of the model.

This was done to reduce the type 2 error as our focus here is to predict as much as possible the people who are going to churn, even if that means we label a few non churning customers as "will churn".

## GridSearchCV

### GridSearchCV

GridSearchCV was used for hyper tuning the model parameters.

Sklearns GridSearchCV functionality was used for this.

Parameter Grid was supplied as an argument to the apply_evl function along with the model itself.

cv=3 was used in this exercise.

# MODEL TUNING METHODS

SMOTE

### SMOTE

A smote data set was also created as the dataset was highly imbalanced in the favor of non churning customers.

To overcome this challenge we create a smote dataset which had synthetically created records so that the churn and non churn classes could be balanced.

This data set was also used to evaluate various models.

Stratify

### Stratify

Stratify was used while splitting the data sets so that the ratio of churning customers is similar in train as well as test set.

This ensured impartial training as well as performance evaluation.

This was mainly required for non smote data sets where data imbalance problem prevailed.

# 5. Model validation

## VALIDATION CRITERIA

Due to the imbalanced nature of the data set, accuracy was not a good choice for validation criteria. We chose AUC Score as the validation criteria based on which we tried to achieve a balance between recall and precision. Primary goal was to maximize the the recall.

## MODEL COMPARISON

All the models which were build were then compared in a tabular as well as visual manner for both test as well as training data sets. Although both train and data set were evaluation, but for for the purpose of choosing the most optimal model, we have only made use of test data set, as that is the data which is unseen to the model and emulates a real world scenario. Hence a model that performs well on test set is likely to perform well in real life.

## MOST OPTIMAL MODEL

After building various models, evaluating performance metrics for each of them and comparing them using the test data set, we finally found our optimal model.

In our case we found **Gradient Boosting Model (GBM)** to be the best performing model.

# VALIDATION CRITERIA

## Maximize Recall

### Maximize Recall

Since the data set is highly imbalanced in favor of non churning customers, it did not made sense to keep accuracy as the primary goal.

Instead we used recall for class 1 i.e. churn, for a better picture. Since recall can tell us out of all the customers who were predicted to churn, how many were correctly classified.

Our focus is to correctly predict as many churning customers as possible, even if that comes at the cost incorrectly predicted non churners.

## Type II Error

### Type II Error

We had to reduce type II error as much as possible. The goal was to maximize True Positives even if that comes at the cost of False Positives.

The primary goal was to minimize False Negatives. We do not want to incorrectly classify a customer as non churner and then loose the customer because of this mistake.

However we also had to strike a balance, as incorrectly predicting too many False Positives also had a financial bearing of the retention budget being used sub optimally.

## AUC Score

### AUC Score

Eventually we settled with AUC score as our ultimate validation criterion. We were able to ward of the recall precision trade off problem to the best possible extent by aiming for the highest AUC score.

AUC Score and F1 Score complemented each other pretty well, hence we could have chosen either of the two.

However AUC Score was chosen at the end as the final contender.

# MODEL COMPARISON

We created multiple models as elaborated earlier, and we also did some model tuning on each of those models. Below are the model performance metrics before and after the threshold tweaking.

Below are the comparison of models before the threshold tweaking.

| Model_Name | Accuracy | Precision | Recall | F1 | AUC |
|---|---|---|---|---|---|
| df_metrics_GBM_tree_unscaled | 0.762298 | 0.550725 | 0.0422927 | 0.078553 | 0.6683 |
| df_metrics_GBM_tree_smote_scaled | 0.762165 | 0.547445 | 0.0417362 | 0.0775595 | 0.668126 |
| df_metrics_GBM_tree_scaled | 0.762298 | 0.551471 | 0.0417362 | 0.0775996 | 0.668028 |
| df_metrics_ADA_tree_scaled | 0.758832 | 0.473451 | 0.0595437 | 0.105783 | 0.653594 |
| df_metrics_ADA_tree_unscaled | 0.758832 | 0.473451 | 0.0595437 | 0.105783 | 0.653594 |
| df_metrics_RF_tree_unscaled | 0.762298 | 0.549296 | 0.0434057 | 0.0804538 | 0.651252 |
| df_metrics_XGB_tree_unscaled | 0.7519 | 0.446488 | 0.148581 | 0.222965 | 0.642457 |
| df_metrics_XGB_tree_scaled | 0.751633 | 0.444816 | 0.148024 | 0.222129 | 0.642343 |
| df_metrics_RF_tree_smote_unscaled | 0.72857 | 0.362486 | 0.175292 | 0.236309 | 0.629299 |
| df_metrics_GBM_tree_smote_unscaled | 0.726303 | 0.365828 | 0.194213 | 0.253726 | 0.623447 |
| df_metrics_logit_linear_scaled | 0.761365 | 0.537634 | 0.0278242 | 0.0529101 | 0.614036 |
| df_metrics_logit_linear_smote_scaled | 0.761099 | 0.525773 | 0.0283806 | 0.0538543 | 0.613638 |
| df_metrics_logit_linear_smote_unscaled | 0.579256 | 0.300558 | 0.569839 | 0.393543 | 0.604523 |
| df_metrics_NB_linear_unscaled | 0.546061 | 0.289859 | 0.61714 | 0.394451 | 0.586783 |
| df_metrics_ADA_tree_smote_unscaled | 0.657512 | 0.300414 | 0.323317 | 0.311445 | 0.582059 |
| df_metrics_svm_linear_scaled | 0.760432 | 0 | 0 | 0 | 0.580036 |
| df_metrics_logit_linear_unscaled | 0.760299 | 0.486486 | 0.0100167 | 0.0196292 | 0.569646 |
| df_metrics_NB_linear_smote_scaled | 0.511265 | 0.27045 | 0.612688 | 0.375256 | 0.551299 |
| df_metrics_NB_linear_scaled | 0.511265 | 0.27045 | 0.612688 | 0.375256 | 0.551299 |
| df_metrics_lda_linear_unscaled | 0.760432 | 0 | 0 | 0 | 0.551063 |
| df_metrics_svm_linear_unscaled | 0.760432 | 0 | 0 | 0 | 0.551063 |
| df_metrics_NB_linear_smote_unscaled | 0.364085 | 0.249621 | 0.824708 | 0.383243 | 0.531791 |

**Table 5.1 Model Performance comparison (Before threshold tweaking)**

We can even though the models have decent enough accuracy, they have really poor recall values and are hence not suitable to be used in our case.

# MODEL COMPARISON

Below are the model performance metrics after doing threshold tweaking.

| Model_Name | AUC | Accuracy | Recall | Precision | F1_Score | Threshold |
|---|---|---|---|---|---|---|
| df_tweak_GBM_tree_scaled | 0.6110 | 0.5186 | 0.7885 | 0.3049 | 0.4397 | 0.2 |
| df_tweak_GBM_tree_unscaled | 0.6110 | 0.5185 | 0.7885 | 0.3048 | 0.4397 | 0.2 |
| df_tweak_GBM_tree_smote_scaled | 0.6108 | 0.5185 | 0.7880 | 0.3047 | 0.4395 | 0.2 |
| df_tweak_XGB_tree_unscaled | 0.6034 | 0.6081 | 0.5943 | 0.3257 | 0.4208 | 0.2 |
| df_tweak_XGB_tree_scaled | 0.6032 | 0.6078 | 0.5943 | 0.3255 | 0.4206 | 0.2 |
| df_tweak_RF_tree_unscaled | 0.6019 | 0.6783 | 0.4552 | 0.3632 | 0.4041 | 0.3 |
| df_tweak_RF_tree_smote_unscaled | 0.5909 | 0.5091 | 0.7479 | 0.2939 | 0.4220 | 0.3 |
| df_tweak_GBM_tree_smote_unscaled | 0.5874 | 0.5490 | 0.6611 | 0.2998 | 0.4126 | 0.3 |
| df_tweak_logit_linear_smote_scaled | 0.5831 | 0.5022 | 0.7385 | 0.2890 | 0.4155 | 0.2 |
| df_tweak_logit_linear_scaled | 0.5824 | 0.5002 | 0.7401 | 0.2884 | 0.4150 | 0.2 |
| df_tweak_logit_linear_smote_unscaled | 0.5760 | 0.5793 | 0.5698 | 0.3006 | 0.3935 | 0.5 |
| df_tweak_NB_linear_unscaled | 0.5724 | 0.5906 | 0.5376 | 0.3013 | 0.3862 | 0.6 |
| df_tweak_NB_linear_smote_scaled | 0.5479 | 0.5939 | 0.4597 | 0.2847 | 0.3516 | 0.9 |
| df_tweak_NB_linear_scaled | 0.5479 | 0.5939 | 0.4597 | 0.2847 | 0.3516 | 0.9 |
| df_tweak_ADA_tree_smote_unscaled | 0.5431 | 0.6575 | 0.3233 | 0.3004 | 0.3114 | 0.5 |
| df_tweak_svm_linear_scaled | 0.5421 | 0.3803 | 0.8525 | 0.2590 | 0.3973 | 0.2 |
| df_tweak_logit_linear_unscaled | 0.5385 | 0.6483 | 0.3278 | 0.2917 | 0.3087 | 0.3 |
| df_tweak_NB_linear_smote_unscaled | 0.5280 | 0.4001 | 0.7735 | 0.2535 | 0.3819 | 0.9 |
| df_tweak_ADA_tree_scaled | 0.5193 | 0.7588 | 0.0595 | 0.4735 | 0.1058 | 0.5 |
| df_tweak_ADA_tree_unscaled | 0.5193 | 0.7588 | 0.0595 | 0.4735 | 0.1058 | 0.5 |
| df_tweak_lda_linear_unscaled | 0.5003 | 0.7606 | 0.0006 | 1.0000 | 0.0011 | 0.3 |
| df_tweak_svm_linear_unscaled | 0.5003 | 0.7606 | 0.0006 | 1.0000 | 0.0011 | 0.5 |

**Table 5.2 Model Performance comparison (After threshold tweaking)**

We can now clearly see much better recall values once the thresholds have been tweaked. The **Table 5.2** is sorted on the AUC column.

The best performing models comes out to be **df_tweak_GBM_tree_scaled i.e.** Gradient Boosting model with Label encoded scaled data. There is not much difference in terms of accuracy and recall between the scaled and unscaled GBM models since it is a tree based model. On the following page you would find the comparison of all the models in a visual manner.

# MODEL COMPARISON

Below is a visual representation of comparison of all the models for all the performance metrics like accuracy, recall, precision, F1 Score and AUC Score.

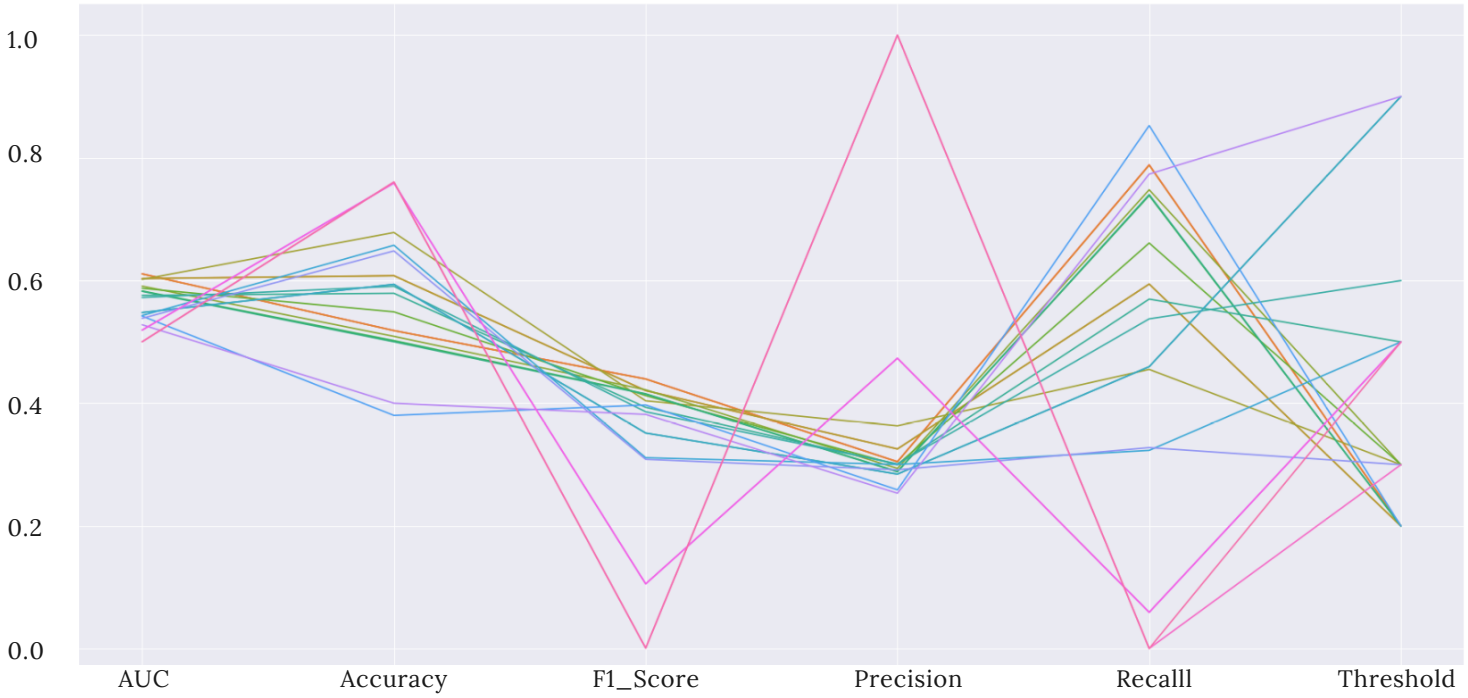
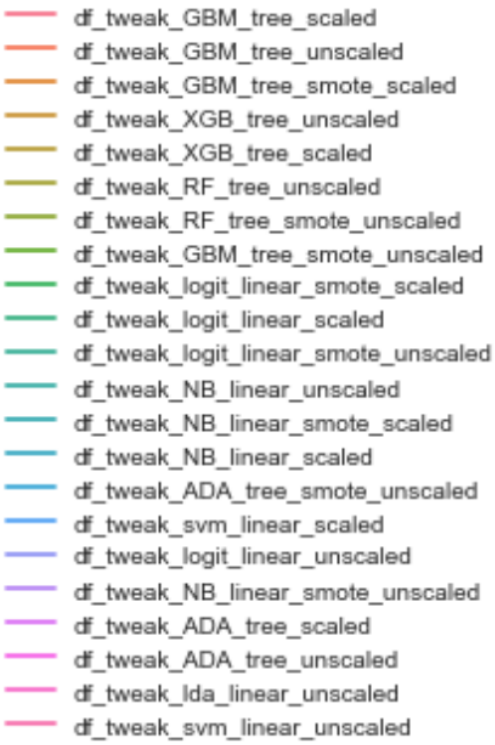



**Figure 5.1 Visual Comparison of all the models (After threshold tweaking)**




**Legend 5.1 Legend for visual comparison of all models.**

# MODEL COMPARISON

We can see the GBM model i.e. **df_tweak_GBM_tree_scaled** indicated by the orange line has the highest AUC score, highest F1 score, almost the highest recall amongst all.

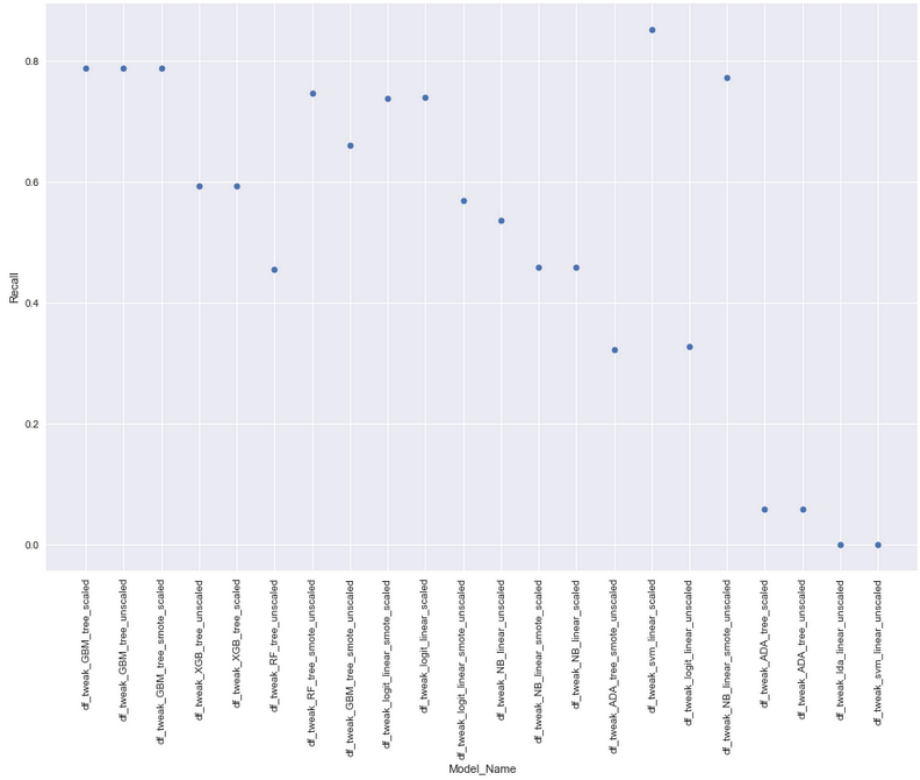



**Figure 5.2 Plot of recall for all the models**

Purely on the basis of recall **df_tweak_svm_linear_scaled** comes out to be the best model with 85.25% recall, however we have tried to maintain a balance between recall and precision. Although False positives are not that important but we still do not want too many of false positives too, where the company will waste its resources to retain customers who are not leaving in the first place.

This is a typical precision - recall trade off problem. We tried to find the sweet spot with GBM model.

# MOST OPTIMAL MODEL

Final model has been chosen as the **df_tweak_GBM_tree_scaled,** this is considering a good balance of recall, precision and accuracy. This model also has the best AUC score.

Data was scaled using Standard Scaler for this model.
Also OneHot encoding was used for the categorical variables while this was built.

Below are the performance metrics of this model.

**Accuracy** - 0.52
**Precision** - 0.30
**Recall** - 0.79
**F1 Score** - 0.44
**AUC Score** - 0.61

This model will help the business predict the customers who are going to churn and will thus lead to customer retention. Which will in turn affect companies ARPU.

It is able to predict 1417 correct customer churns out of 1797 actual churns in the test data. Which means out of all the customers who will actually churn this model is able to predict 79% of them correctly.

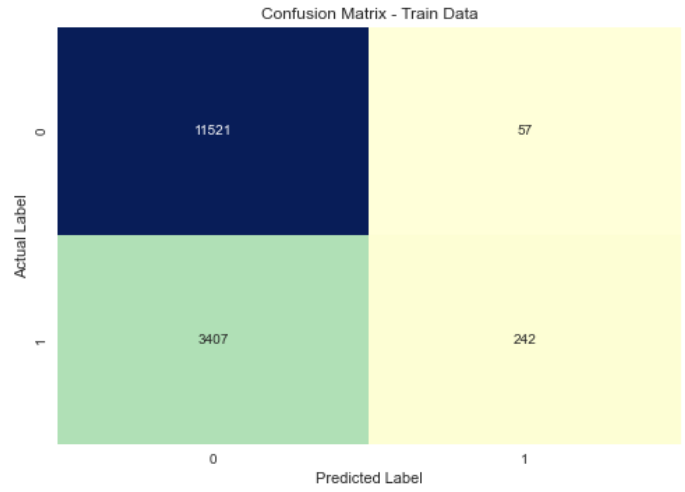
# MOST OPTIMAL MODEL

GBM - Performance Metrics (Threshold = 0.5)

Below are some of the performance metrics that we calculated for GBM ML model. Please note these are performance metrics with threshold 0.5. Actual metrics with 0.2 threshold which was selected to be optimum are on the next page.

## Train Data

Accuracy Score (Train Data) - 77 %




```
--------------------Classification Report - Train Data-------
              precision    recall  f1-score   support

           0       0.77      1.00      0.87     11578
           1       0.81      0.07      0.12      3649

    accuracy                           0.77     15227
   macro avg       0.79      0.53      0.50     15227
weighted avg       0.78      0.77      0.69     15227

------------------------------------------------------------
```

AUC: 0.756

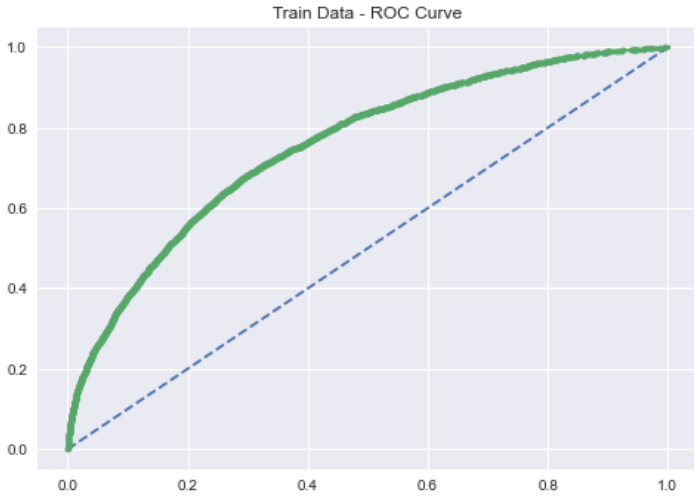
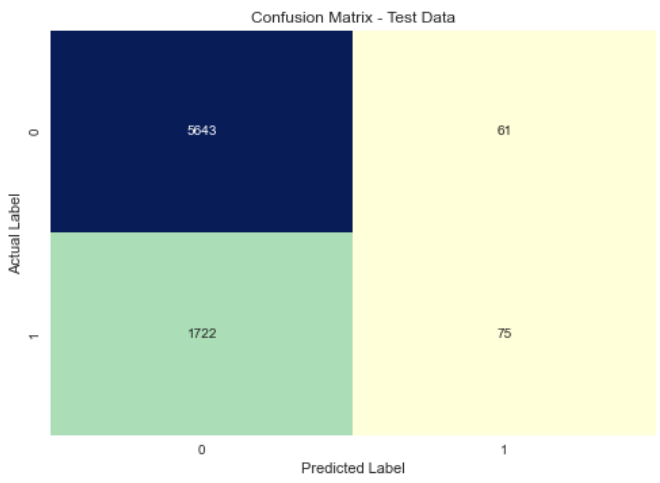



**Figure 5.3 Performance metrics for GBM (Train)**

## Test Data

**Accuracy Score (Test Data) - 76 %**



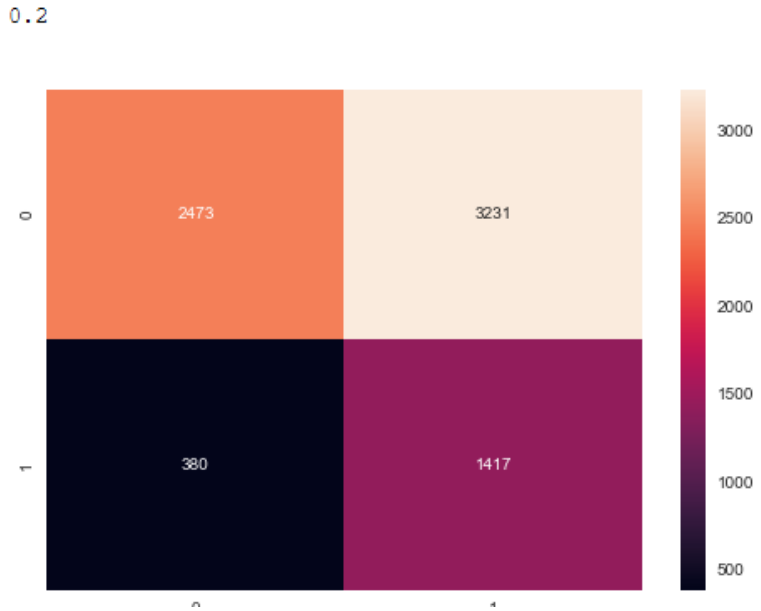

```
--------------------Classification Report - Test Data---------
              precision    recall  f1-score   support

           0       0.77      0.99      0.86      5704
           1       0.55      0.04      0.08      1797

    accuracy                           0.76      7501
   macro avg       0.66      0.52      0.47      7501
weighted avg       0.71      0.76      0.68      7501

------------------------------------------------------------
```

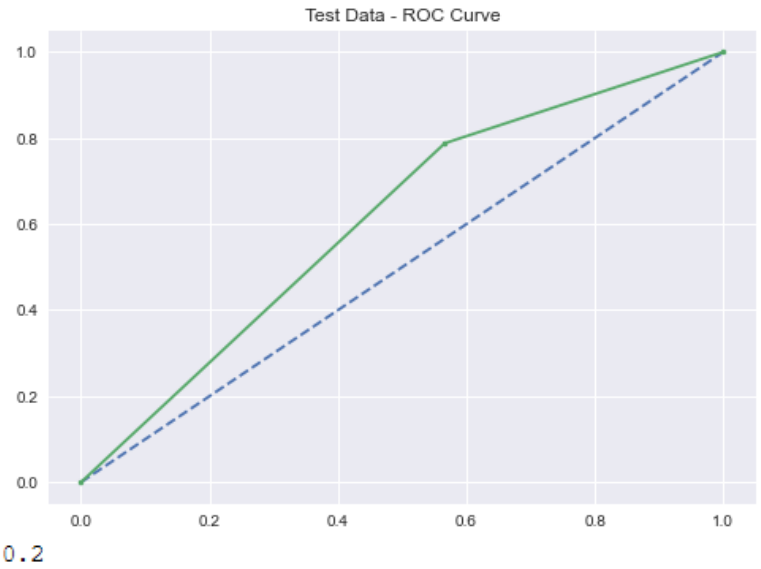
AUC: 0.668


**Figure 5.4 Performance metrics for GBM (Test)**

# MOST OPTIMAL MODEL

## GBM - Performance Metrics (Threshold = 0.2)




----------Classification Report - Test Data----

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.87 | 0.43 | 0.58 | 5704 |
| 1 | 0.30 | 0.79 | 0.44 | 1797 |
| accuracy | | | 0.52 | 7501 |
| macro avg | 0.59 | 0.61 | 0.51 | 7501 |
| weighted avg | 0.73 | 0.52 | 0.54 | 7501 |

**Figure 5.5 Performance metrics for GBM (Test) (Threshold Tweaked)**

Above are the performance metrics with tweaked threshold of 0.2 for this model.

Performance metrics for all the other models can be found in **Appendix 5.1**

# 6. Final interpretation / recommendations



## TOP 5 FACTORS

From the most optimal model we have determined the top 5 factors that influence customer churn. Eqpdays, mou_mean, months, change_mou and age1 were found to be most influential factors in customers decision to churn. Each of these factors have been discussed in details in the next sections.

## REVENUE SAVES

We had done customer segmentation using KMeans model. However when we combine the learning from the classification model to that of the clsutering model, it generates a few more business insights which might be of interest to the telecom operator, as these can directly lead to "Revenue Saves" and hence help increase the overall ARPU for the telecom operator.

## PROACTIVE RETENTION STRATEGY

As a final step of this project, we are proposing a comprehensive proactive retention strategy for the telecom operator.

Following this strategy will help the operator reduce customer churn and will also help arrest the dropping ARPU.

# TOP 5 FACTORS

Feature importance was calculated for the most optimal model i.e. GBM. Below is the graph showing the relative importance of each of the features in the data set.

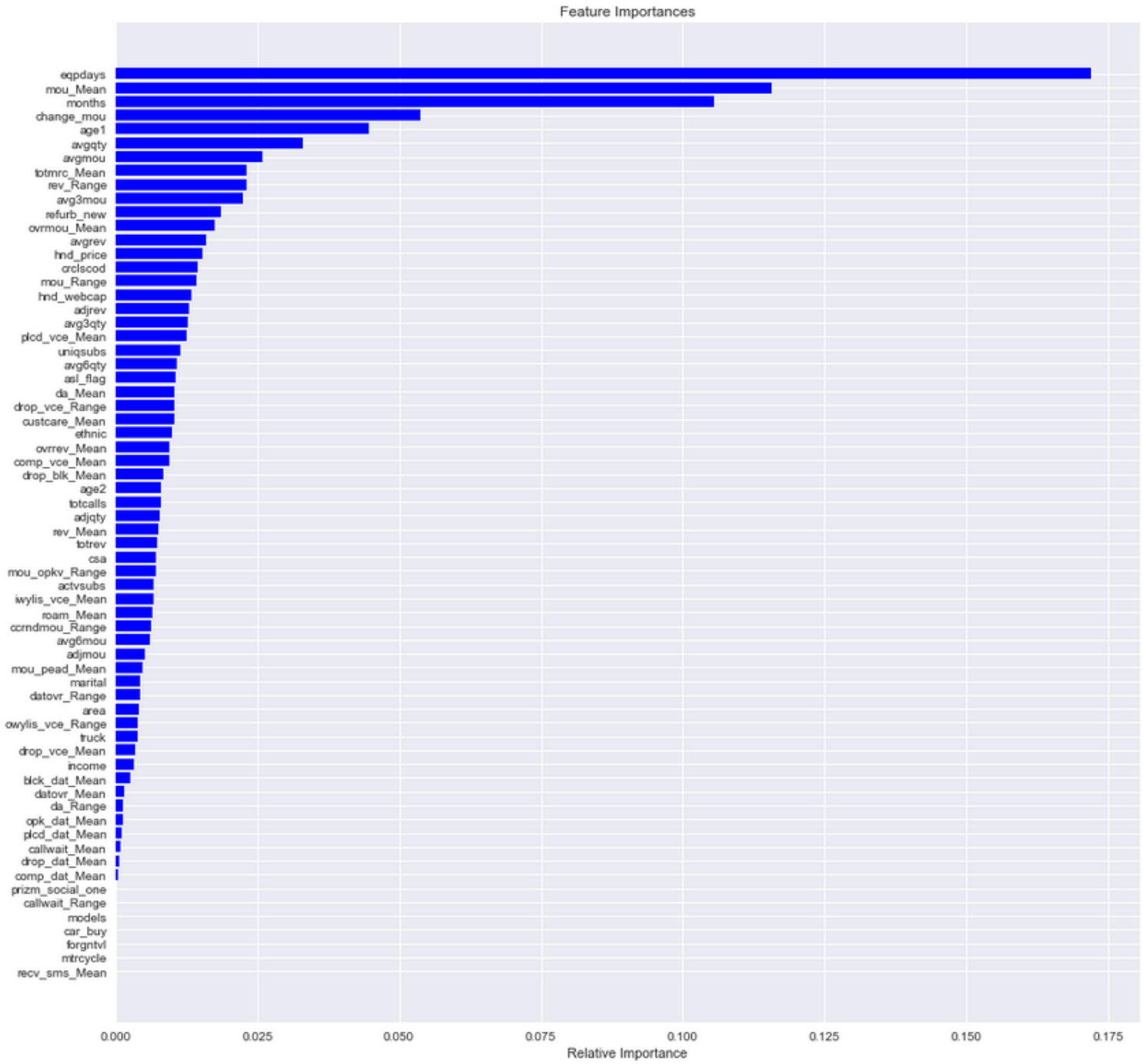



**Figure 6.1 Feature Importance for the most optimal model (GBM)**

# TOP 5 FACTORS

## EQPDAYS

**EQPDAYS**

It was found that eqpdays days plays the most important role in customer churn. Which can indicate that older a customers mobile equipment, higher is the chance to churn.

This might be because older equipment do not have the latest features like 4G, 5G and while the company might be focusing to update to new technologies, customers who do not have compatible handsets and are left with older weaker network coverage like 2G and 3G might be likely to churn.

## MOU_MEAN

**MOU_MEAN**

Another important factor for customer churn is mou_mean i.e. Mean number of monthly minutes of use.

This means the users who are calling more often are more likely to churn compared to those who are calling for lesser number of minutes on an average.

This points to a cost and biling problem.

## MONTHS

**MONTHS**

Also factors like how long the customer has been with the telecom operator plays a major role.

Older a customer, more likely is he / she to churn. This is indicated by the months column appearing very high on our feature importance list.

## CHANGE_MOU

**CHANGE_MOU**

Change in the monthly minutes of usage compared to last 3 months is also an important factor.

Higher the change, more is the chance to churn.

## AGE1

**AGE1**

Age of customers first household member also has a bearing on the churn rate.

We assume the age of the first household member might be the age of the customer himself / herself.

This would mean that the age of the customer has a bearing on the customer churn rate.

# REVENUE SAVES

Using the information available after doing exploratory data analysis (EDA), prediction model building as well as the customer segmentation analysis. We can combine all this information to extract a few useful business insights.

From the customer segmentation we can see that cluster 2 churns the least followed by Cluster 1, Cluster 3 and Cluster 0.

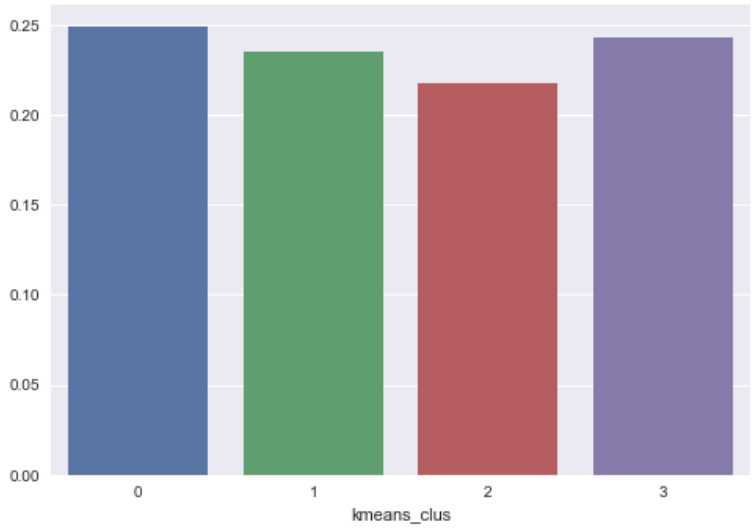



**Figure 6.2 Cluster Profile**

Looking at the above, one might be tempted to jump to the conclusion and pour all the efforts to arrest customer churn in Cluster 0. However when we combine the customer segmentation information with the some of the top factors influencing churn, we see a different picture.

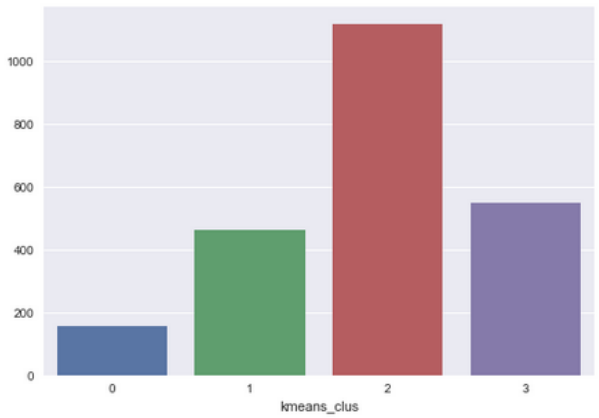



**Figure 6.3 Cluster Profile (mou_mean)**

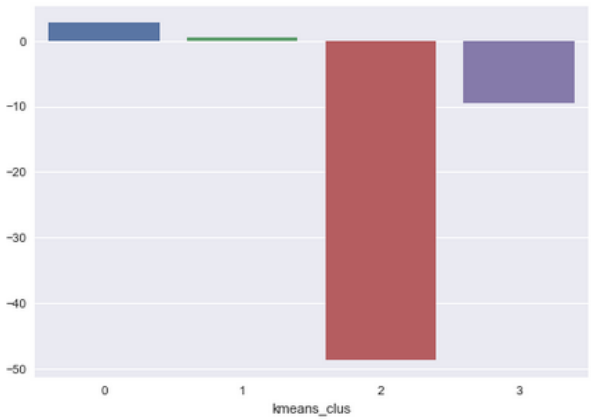



**Figure 6.4 Cluster Profile (change_mou)**

# REVENUE SAVES

Looking at **mou_mean**, we can clearly see, even though cluster 0 customers are more likely to churn, however the average revenue generated by these customers is the lowest. On the other hand, even though customers in cluster 2 are least likely to churn, however they are the highest ARPU generators for the telecom operator.

Hence it makes sense to focus on these high revenue generators rather than pouring money to retain customers who will not even generate revenue for the company in future.

Another important factor is **change_mou.** We can see that the change in monthly minutes of usage is the highest amongst customers in cluster 2, which is alarming and a grave cause of concern for the operator. We do not want our most premium customer to be using lesser and lesser of our services.

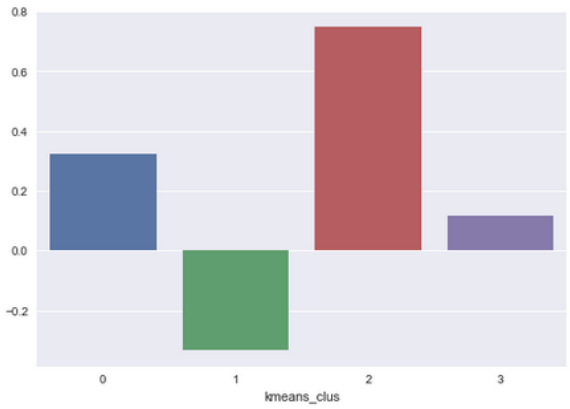



Figure 6.5 Cluster Profile (custcare_mean)

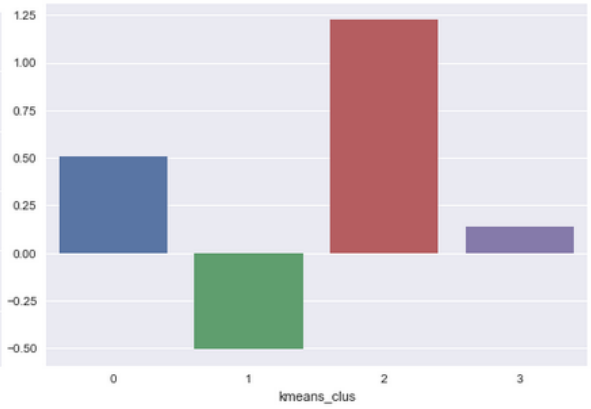



Figure 6.6 Cluster Profile (drop_vce_mean)

Also we can customers in cluster 2 have called customer care the most number of times and have also faced the most dropped voice calls as is evident from **Figure 6.5** and **Figure 6.6.**

Hence it suffices to say that the telecom operator should not look at the just the top factors nor should they simply look at the customer segmentation data independently.

It would make sense for the operator to combine all the information that has been derived as part of this exercise and then extract meaningful business insights which not only help with customer retention but also help with maximizing the "Revenue Saves".

# PROACTIVE RETENTION STRATEGY

As a final step we are proposing a proactive customer retention strategy. However before that there were a few questions related to "Cost and Billing Issues" & "Network & Service Quality" that the telecom operator wanted to be answered.

## Cost & Billing Issues

- Cost & Billing related issue seem to be one of the driving factors for customer churn.
- mou_mean, change_mou, avgmou, rev_range, totmrc_mean, avgrev and ovrmou_Mean - all of them being in the top 15 factors influencing churn, clearly points to the cost and billing related issues.
- Customers having higher usage, hence higher bill amount are more likely to churn compared to the others.

## Network & Service Quality

- Network & Service Quality when looked at directly does not seem to be a very important factor.
- Although it is an important factor, but not as much as the cost & billing issues.
- drop_vce_Range, drop_blk_Mean feature somewhere in the middle when we look at all the features and their respective relative importance.
- However eqpdays can be directly related to network, as the old equipments might not be compatible with new network technologies like 4G and 5G, hence we cannot completely ignore network and service quality issues.

# PROACTIVE RETENTION STRATEGY

- Since **cost and billing issues** weight heavily towards a customers decision to leave the telecom operator .
  - We would suggest a **rate plan migration** for customers who have the highest usage i.e. customers in cluster 2.

- **EQPDAYS** is the most important factor driving churn. Which significantly impacts customers in clusters 0 & 3. Even if we ignore cluster 0 customers, it makes sense to improve on this for the sake of customers in cluster 3, who have high ARPU.
  - Continuing support for erstwhile network technologies like 2G & 3G might pay off in long term, if the cost of maintaining older networks is lower than the overall revenue generated by these customers.

- Customers in Cluster 3 have the oldest relationship (**months**) with the telecom operator on an average, and they are also very high ARPU generators (if not the highest). Hence we would suggest to retain this segment of the customers as well.
  - Loyalty programs can be offered to these customers.

- In terms of "**revenue saves**" , customer retention budget should be focused on customers in Cluster 2 followed by Cluster 3 > Cluster 1 > Cluster 0

- **Some other actions which might be taken if the retention budget allows.**

- **crclscod**
  - crclscod is within the top 15 factors influencing churn.
  - we can see customers in credit class 2 are more likely to churn.
  - Telecom operator may choose to retain these customers to reduce the churn rate.

- **asl_flag**
  - Account Spending Limit Flag is usually only set for postpaid customers.
  - Upon analysis we saw, prepaid customers are the ones who are churning more than the postpaid customers.
  - Even though ARPU for prepaid customers is lower than postpaid customers, there is still a vast difference in the churn for prepaid vs. postpaid, that we cannot simply ignore prepaid customers.


THANK
YOU!