Telecom Churn Machine Learning Case Study

By:
Srikanth Reddy Somireddy
Shubham Rustagi

Business Problem Overview

- ❖ In the telecom industry, customers are able to choose from multiple service providers and actively switch from one operator to another. In this highly competitive market, the telecommunications industry experiences an average of 15-25% annual churn rate. Given the fact that it costs 5-10 times more to acquire a new customer than to retain an existing one, customer retention has now become even more important than customer acquisition.
- ❖ For many incumbent operators, retaining high profitable customers is the number one business goal.
- ❖ To reduce customer churn, telecom companies need to predict which customers are at high risk of churn.
- ❖ In this project, you will analyze customer-level data of a leading telecom firm, build predictive models to identify customers at high risk of churn and identify the main indicators of churn.

Business objective:

The business objective is to predict the churn in the last (i.e. the ninth) month using the data (features) from the first three months. To do this task well, understanding the typical customer behavior during churn will be helpful.

Understanding Customer Behavior During Churn

Customers usually do not decide to switch to another competitor instantly, but rather over a period of time (this is especially applicable to high-value customers). In churn prediction, we assume that there are three phases of customer lifecycle :

- **The good phase:** In this phase, the customer is happy with the service and behaves as usual.
- **The action phase:** The customer experience starts to sore in this phase, for e.g. he/she gets a compelling offer from a competitor, faces unjust charges, becomes unhappy with service quality etc. In this phase, the customer usually shows different behavior than the 'good' months. Also, it is crucial to identify high-churn-risk customers in this phase, since some corrective actions can be taken at this point (such as matching the competitor's offer/improving the service quality etc.)
- **The churn phase:** In this phase, the customer is said to have churned. You define churn based on this phase. Also, it is important to note that at the time of prediction (i.e. the action months), this data is not available to you for prediction. Thus, after tagging churn as 1/0 based on this phase, you discard all data corresponding to this phase.

In this case, since you are working over a four-month window, the first two months are the good phase, the third month is the action phase, while the fourth month is the churn phase.

Analyzing the “telecom_churn_data.csv”

The dataset contains customer-level information for a span of four consecutive months - June, July, August and September. The months are encoded as 6, 7, 8 and 9, respectively.

Checking the Shape, Info & Statistics of the data set

In [3]:
```
print (churn.shape)
print('-'*50)
print (churn.info())
print('-'*50)
churn.describe()
```

```
(99999, 226)
--------------------------------------------------
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99999 entries, 0 to 99998
Columns: 226 entries, mobile_number to sep_vbc_3g
dtypes: float64(179), int64(35), object(12)
memory usage: 172.4+ MB
None
--------------------------------------------------
```

Out[3]:

	mobile_number	circle_id	loc_og_t2o_mou	std_og_t2o_mou	loc_ic_t2o_mou	arpu_6	arpu_7	arpu_8	arpu_9	onnet_mou_6	on
count	9.999900e+04	99999.0	98981.0	98981.0	98981.0	99999.000000	99999.000000	99999.000000	99999.000000	96062.000000	96
mean	7.001207e+09	109.0	0.0	0.0	0.0	282.987358	278.536648	279.154731	261.645069	132.395875	
std	6.956694e+05	0.0	0.0	0.0	0.0	328.439770	338.156291	344.474791	341.998630	297.207406	
min	7.000000e+09	109.0	0.0	0.0	0.0	-2258.709000	-2014.045000	-945.808000	-1899.505000	0.000000	
25%	7.000606e+09	109.0	0.0	0.0	0.0	93.411500	86.980500	84.126000	62.685000	7.380000	
50%	7.001205e+09	109.0	0.0	0.0	0.0	197.704000	191.640000	192.080000	176.849000	34.310000	
75%	7.001812e+09	109.0	0.0	0.0	0.0	371.060000	365.344500	369.370500	353.466500	118.740000	
max	7.002411e+09	109.0	0.0	0.0	0.0	27731.088000	35145.834000	33543.624000	38805.617000	7376.710000	8

Based on the results provided above, we can conclude that our dataset comprises 99,999 records with 226 features.

After Performing the Exploratory Data Analysis on the telecom churn data set, we Were left out with 99618 records with 211 features

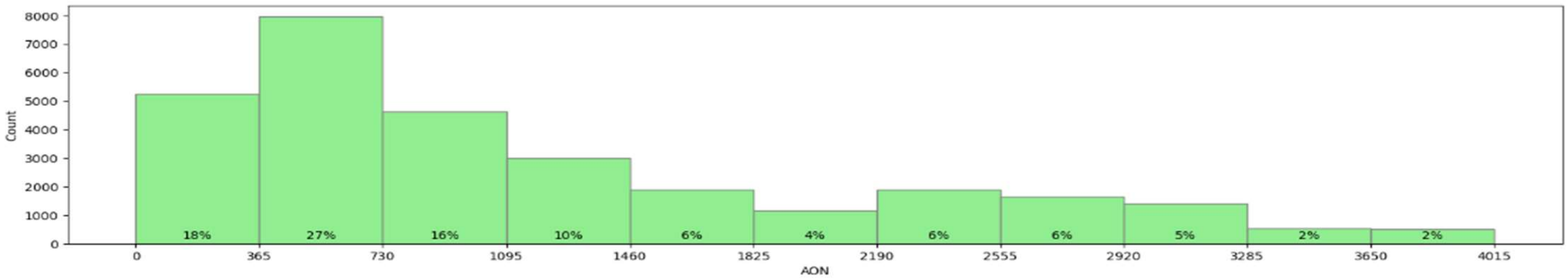
```
In [33]: # Percentage of data left after removing the missing values.
         print("Percentage of data remaining after treating missing values: {}%".format(round(churn.shape[0]/99999 *100,2)))
         print ("Number of customers: {}".format(churn.shape[0]))
         print ("Number of features: {}".format(churn.shape[1]))

         Percentage of data remaining after treating missing values: 99.62%
         Number of customers: 99618
         Number of features: 211
```

Data Analysis :

Customers Distribution of the Age on Network

```
# Customers distribution of the age on network
print(hv_users.aon.describe())
plot_hist(hv_users,'aon',365)
```

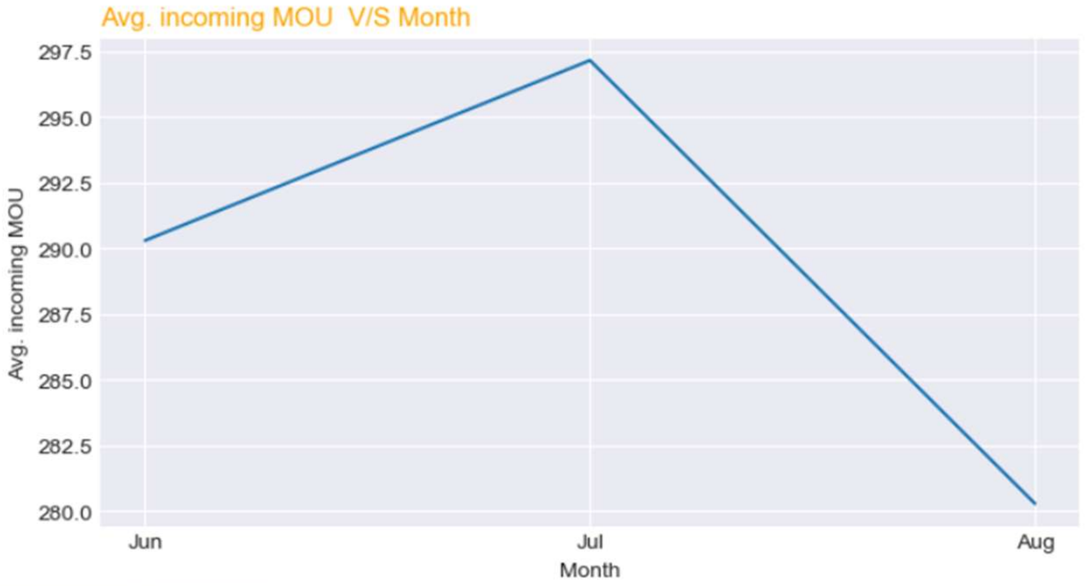
```
count     29906.000000
mean       1209.062396
std         957.342718
min         180.000000
25%         460.000000
50%         846.000000
75%        1755.000000
max        4321.000000
Name: aon, dtype: float64
```

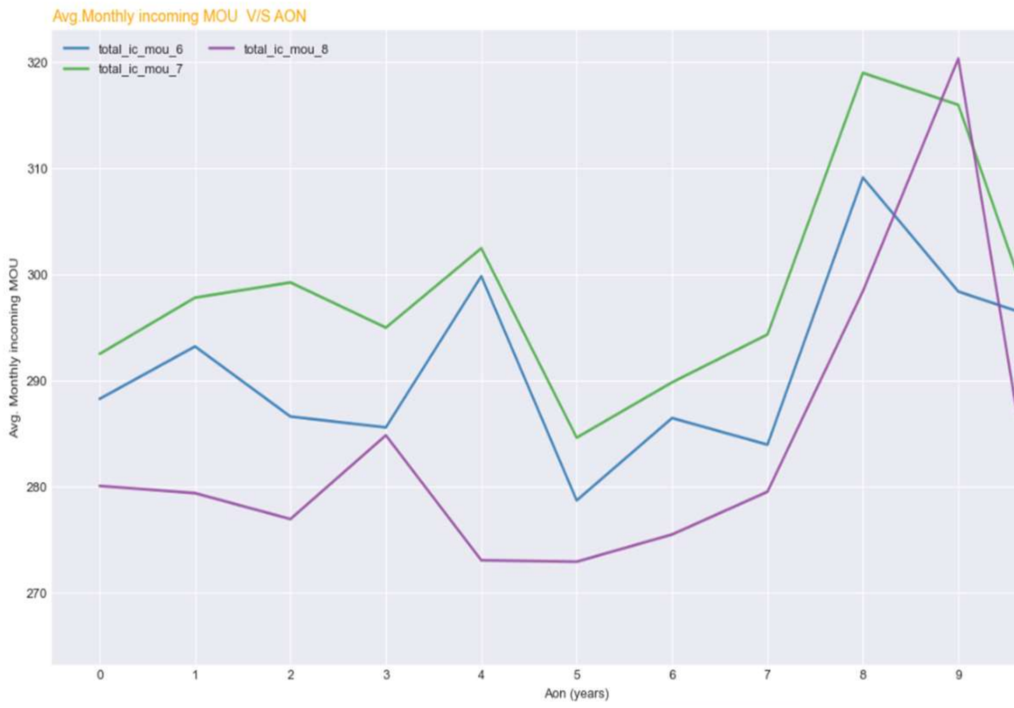


- Minimun Age on network is 180 days.
- Average age on network for customers is 1200 days (3.2 years).
- 27% of the HV users are in their 2nd year with the network.
- Almost 71% users have Age on network less than 4 years.
- 15% users are with the network from over 7 years.

Incoming VS month VS AON

```
# Plotting Avg. total monthly incoming MOU vs AON
ic_col = hv_users.filter(regex ='total_ic_mou').columns
plot_avgMonthlyCalls('single',hv_users,calltype='incoming',colList=ic_col)
plot_avgMonthlyCalls('multi',hv_users,calltype='incoming',colList=ic_col)
```



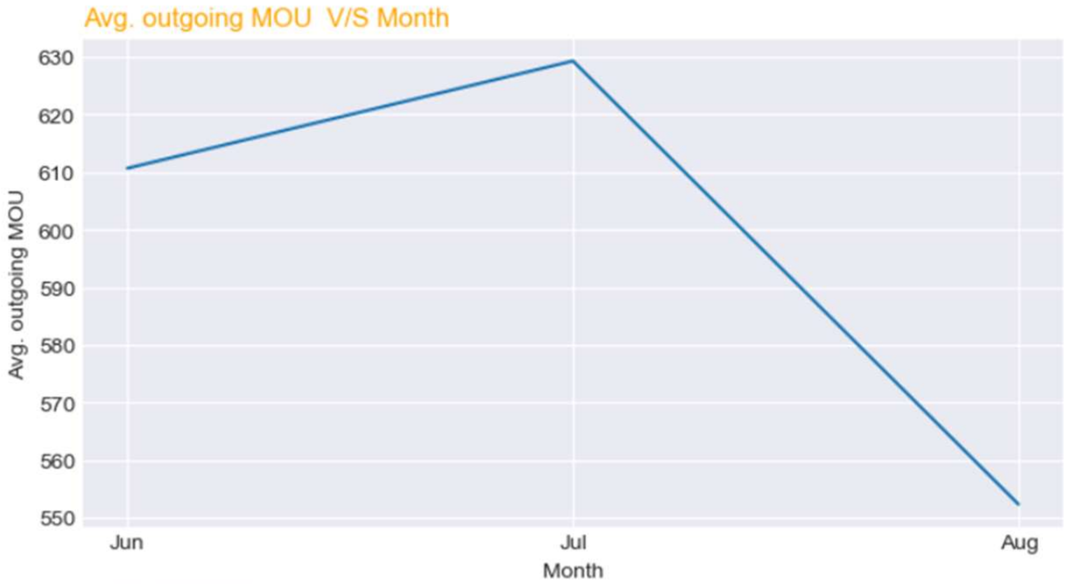


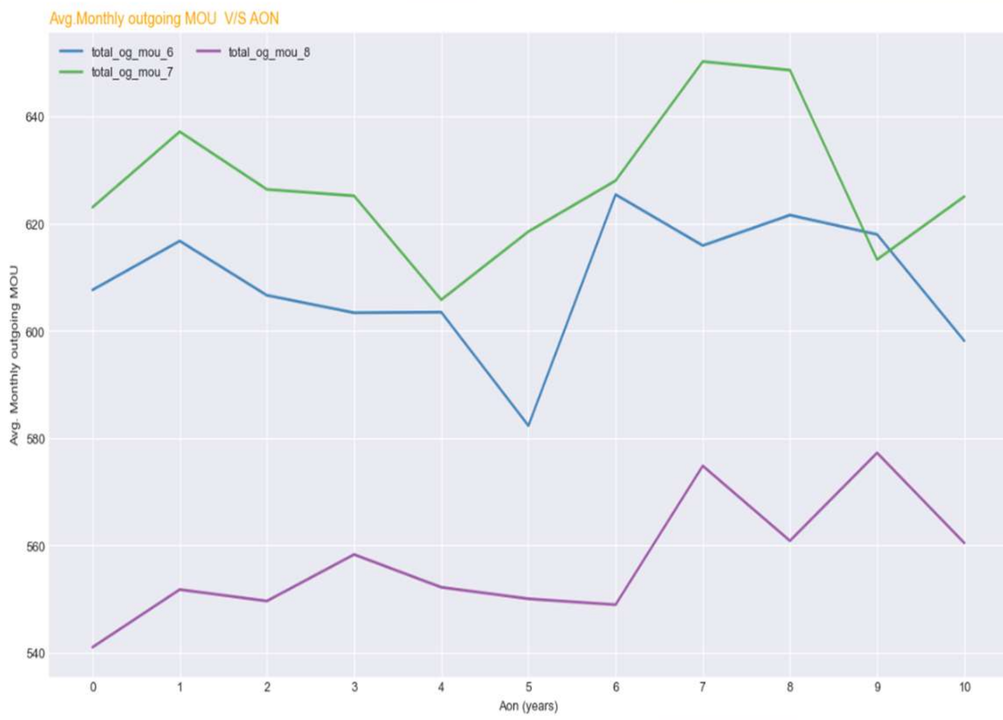
It is evident from the plot that,

- The more a customer stays on with the operator(AON), more are the total monthly incoming MOU.
- Total Incoming MOU avg. for Jul(_7) are more than the previous Jun(_6) for customers in all AON bands.
- Total Incoming MOU avg. for Aug(_8) cease to increace, infact it shows a decline compared to Jul(_7).
- Total Incoming MOU avg. for Sep(_9) is well below the first months(jun _6) avg.
- Althought the Total incoming mou avg inceases from jun to july, it drop little from aug and reduces lower than that for jun.

Outgoing VS month VS AON

```
# Plotting Avg. total monthly outgoing MOU vs AON
og_col = hv_users.filter(regex ='total_og_mou').columns
plot_avgMonthlyCalls('single',hv_users,calltype='outgoing',colList=og_col)
plot_avgMonthlyCalls('multi',hv_users,calltype='outgoing',colList=og_col)
```



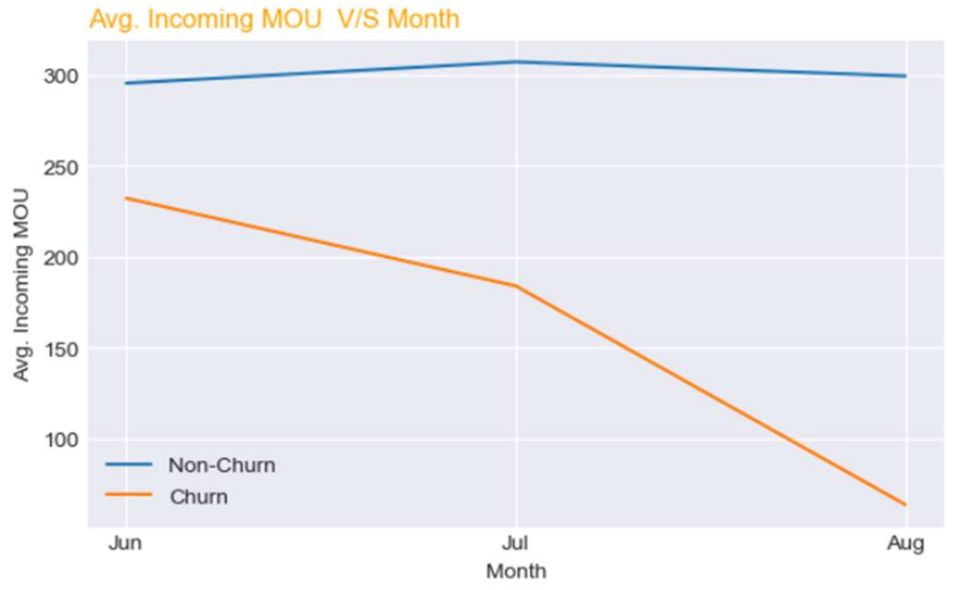


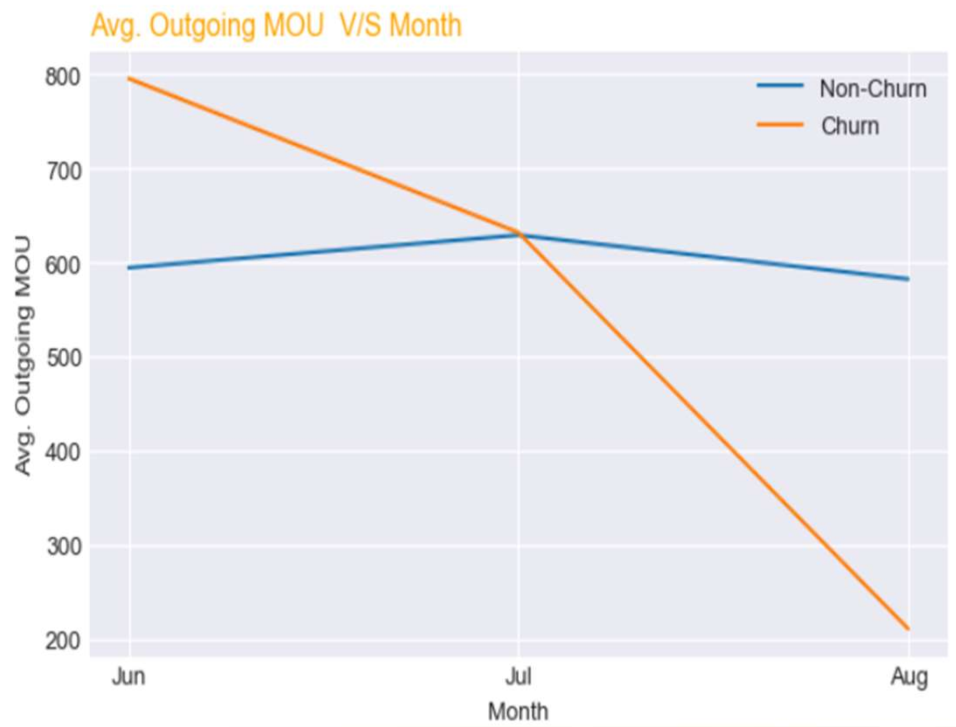
What is the above plot saying ?

- Overall, the Avg. total outgoing usage reduces with the increasing age on network.
- Total Outgoing MOU avg. for Jul(_7) are more than the previous Jun(_6) for customers in all AON bands, except in the AON band between 7 - 8 years where it is almost simillar.
- Total outgoing MOU avg. for Aug(_8) cease to increace, infact it shows a significant decline compared to Jul(_7).
- Total outgoing MOU avg. for Sep(_9) is the lowest of all 4 months.
- The Avg. outgoing usage reduces drastically for customers in the AON band between 7 - 8 years.

Incoming/Outgoing MOU VS Churn

```
ic_col = ['total_ic_mou_6','total_ic_mou_7','total_ic_mou_8']
og_col = ['total_og_mou_6','total_og_mou_7','total_og_mou_8']
plot_byChurnMou(ic_col,'Incoming')
plot_byChurnMou(og_col,'Outgoing')
```





It can be observed,

- Churners Avg. Incoming/Outgoing MOU's drops drastically after the 2nd month,Jul.
- While the non-churners Avg. MOU's remains consistant and stable with each month.
- Therefore, users MOU is a key feature to predict churn.

Let's also see this trend in terms of actual numbers.

```
# Avg.Incoming MOU per month churn vs Non-Churn
hv_users.groupby(['churn'])['total_ic_mou_6','total_ic_mou_7','total_ic_mou_8'].mean()
```

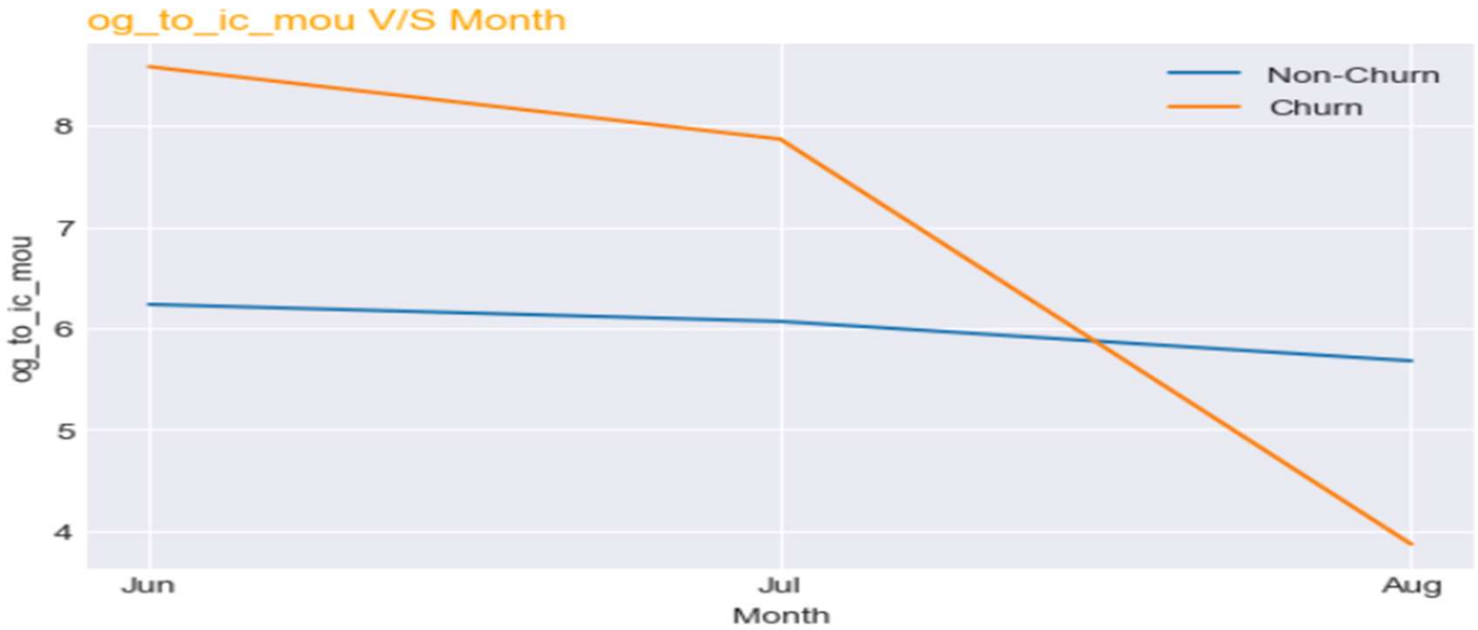
churn	total_ic_mou_6	total_ic_mou_7	total_ic_mou_8
0	295.401726	307.108317	299.319664
1	232.221162	183.978888	63.813168

```
# Avg. Outgoing MOU per month churn vs Non-Churn
hv_users.groupby(['churn'])['total_og_mou_6','total_og_mou_7','total_og_mou_8'].mean()
```

churn	total_og_mou_6	total_og_mou_7	total_og_mou_8
0	594.414582	629.096568	582.380539
1	795.591038	631.859433	210.659326

Og_to_Ic_mou V/s Month

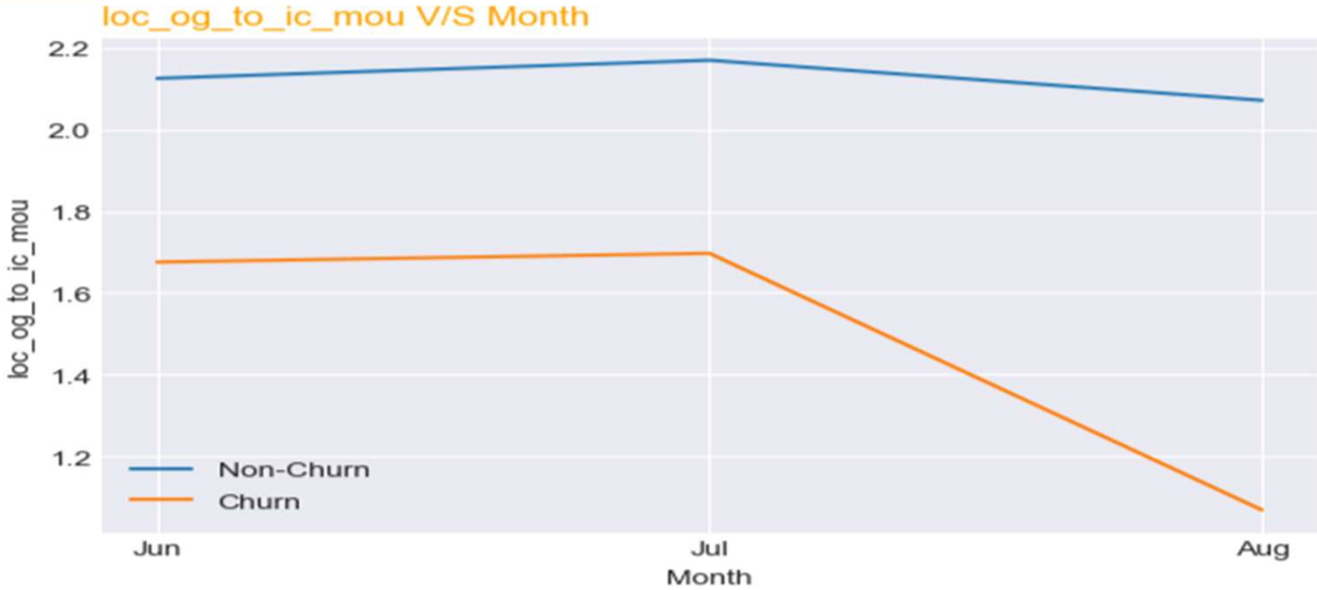
```
plot_byChurn(hv_users,'og_to_ic_mou')
```



churn	og_to_ic_mou_6	og_to_ic_mou_7	og_to_ic_mou_8
0	6.235602	6.067952	5.678424
1	8.580257	7.865938	3.870145

- Outgoing to incoming mou remains drops significantly for churners from month Jul(6) to Aug(7).
- While it remains almost consistent for the non-churners.

Loc_Og_to_Ic_Mou V/s Month

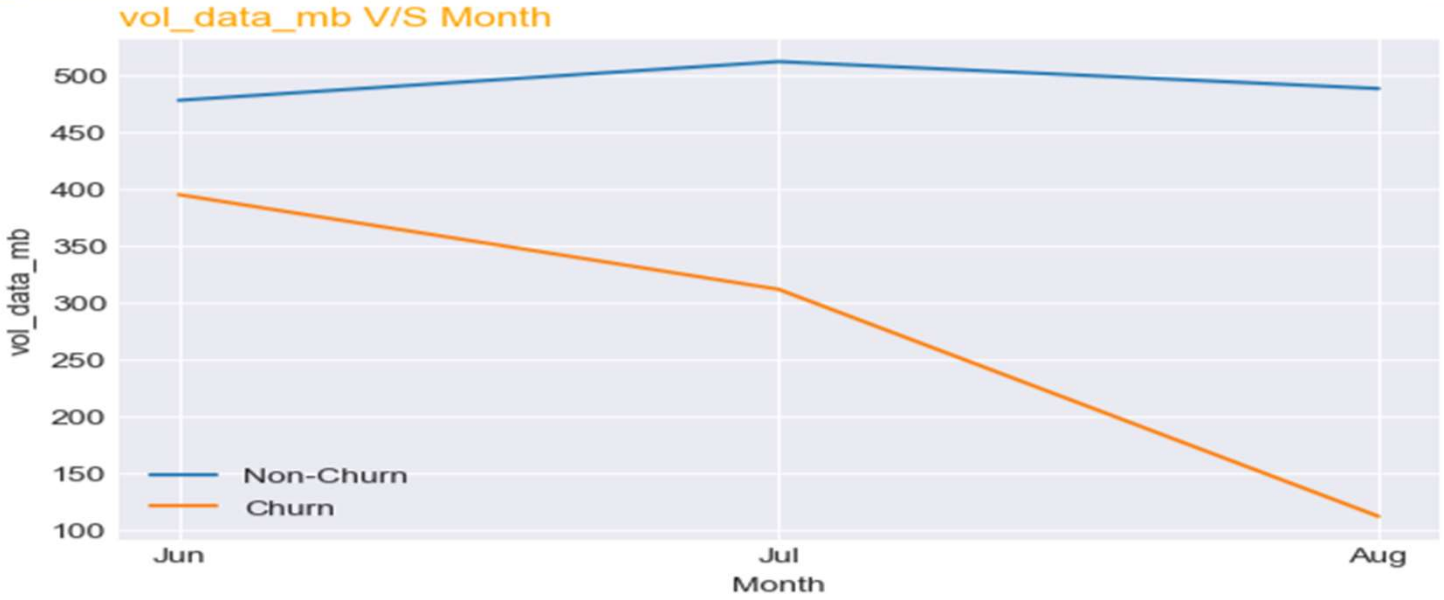


churn	loc_og_to_ic_mou_6	loc_og_to_ic_mou_7	loc_og_to_ic_mou_8
0	2.124471	2.168763	2.070806
1	1.675413	1.696809	1.069765

It can be observed that,

- The local outgoing to incoming call mou ratio is genrally low for churners right from the begining of the good phase.
- local mou pattern for the non-churners remains almost constant through out the 3 months.
- The churners genrally show a low loc mou ratio but it drops dramatically after the 2nd month.
- This might suggest that people who are not making/reciving much local calls during their tenure are more likely to churn.

Total Data Volume V/s Churn

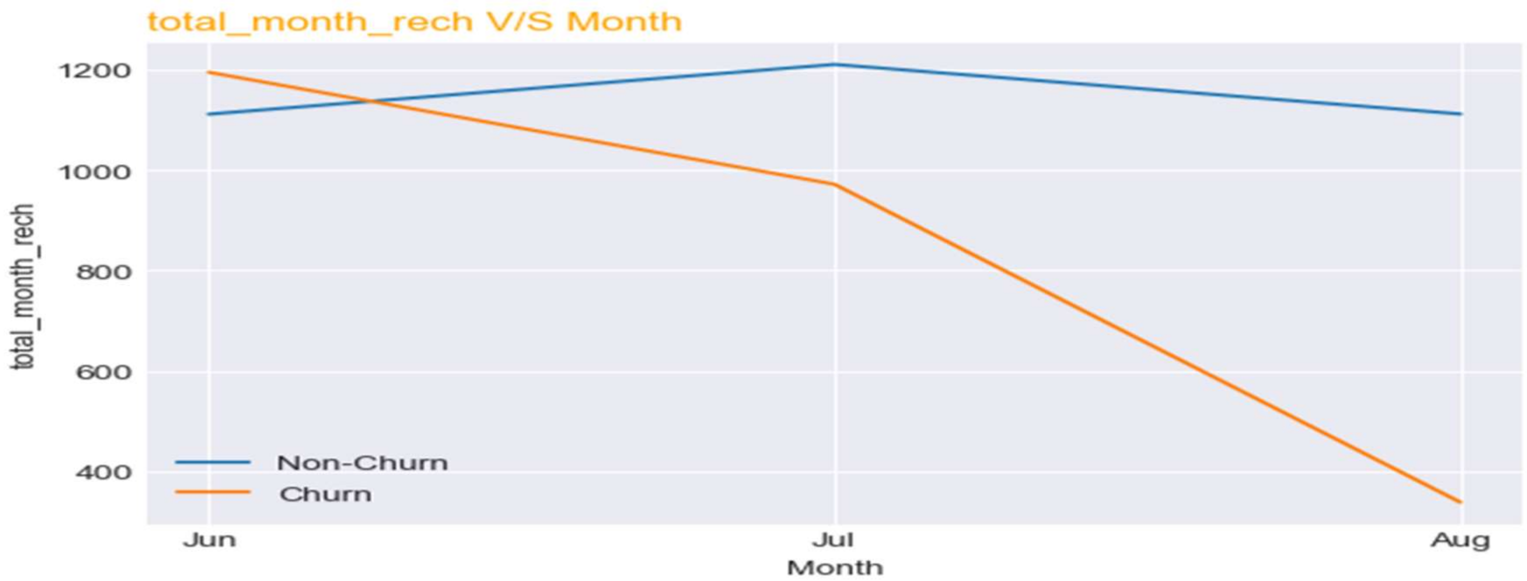


churn	vol_data_mb_6	vol_data_mb_7	vol_data_mb_8
0	478.037762	512.164072	488.389661
1	394.949545	311.507444	111.469396

- The volume of data mb used drops significantly for churners from month Jul(6) to Aug(7).
- While it remains almost consistent for the non-churners.

Total Monthly rech V/s Churn

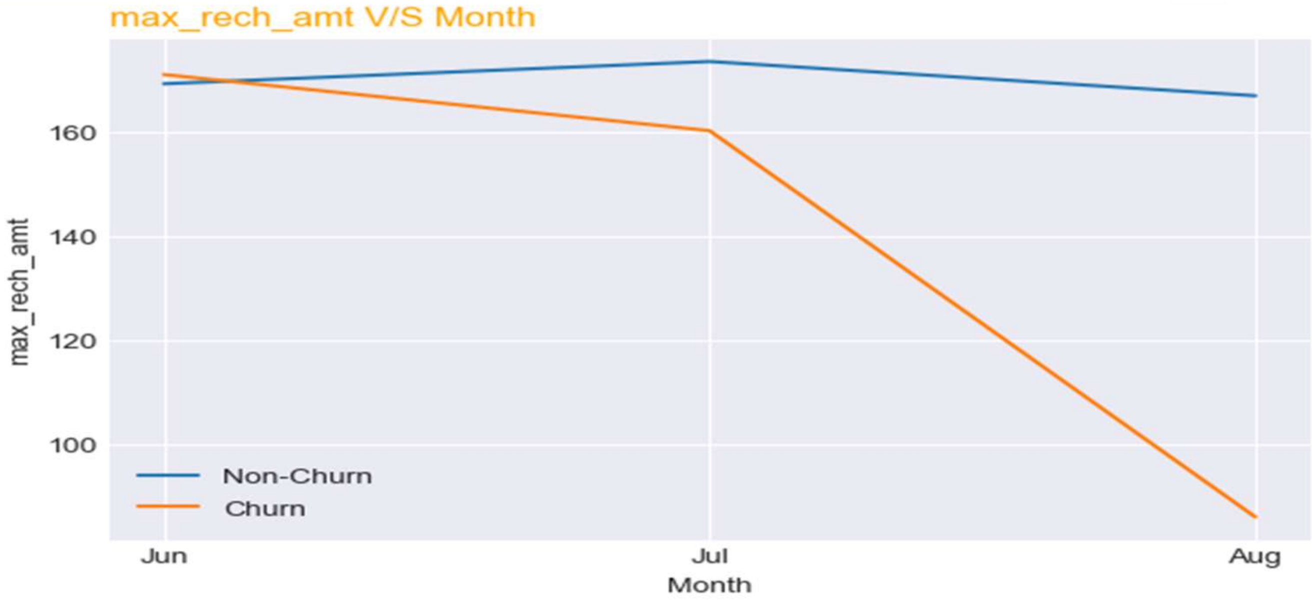
```
plot_byChurn(hv_users,'total_month_rech')
```



churn	total_month_rech_6	total_month_rech_7	total_month_rech_8
0	1111.439977	1210.362853	1111.756912
1	1194.747593	971.802758	339.278974

- Total monthly rech amount also drops significantly for churners from month Jul(6) to Aug(7).
- While it remains almost consistent for the non-churners.

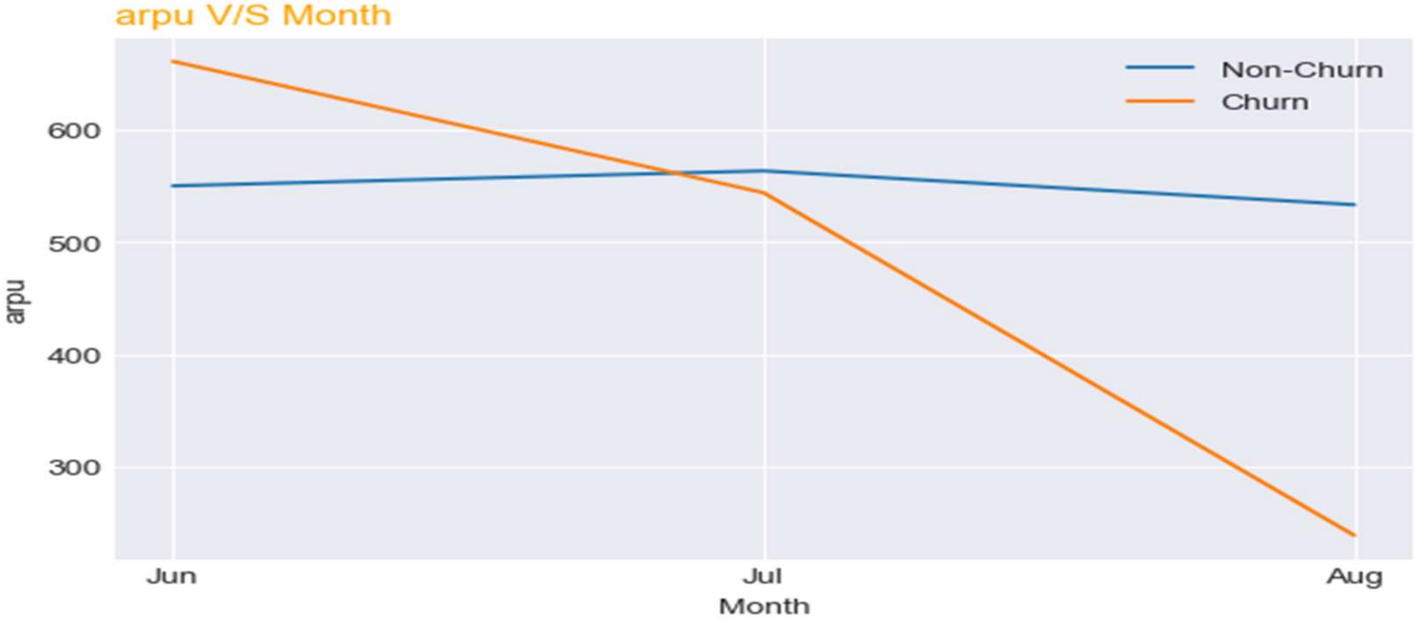
Max Rech Amount V/s Churn



churn	max_rech_amt_6	max_rech_amt_7	max_rech_amt_8
0	169.160943	173.437282	166.865250
1	170.930108	160.152192	86.026468

- Maximum recharge amount also drops significantly for churners from month Jul(6) to Aug(7).
- While it remains almost consistent for the non-churners.

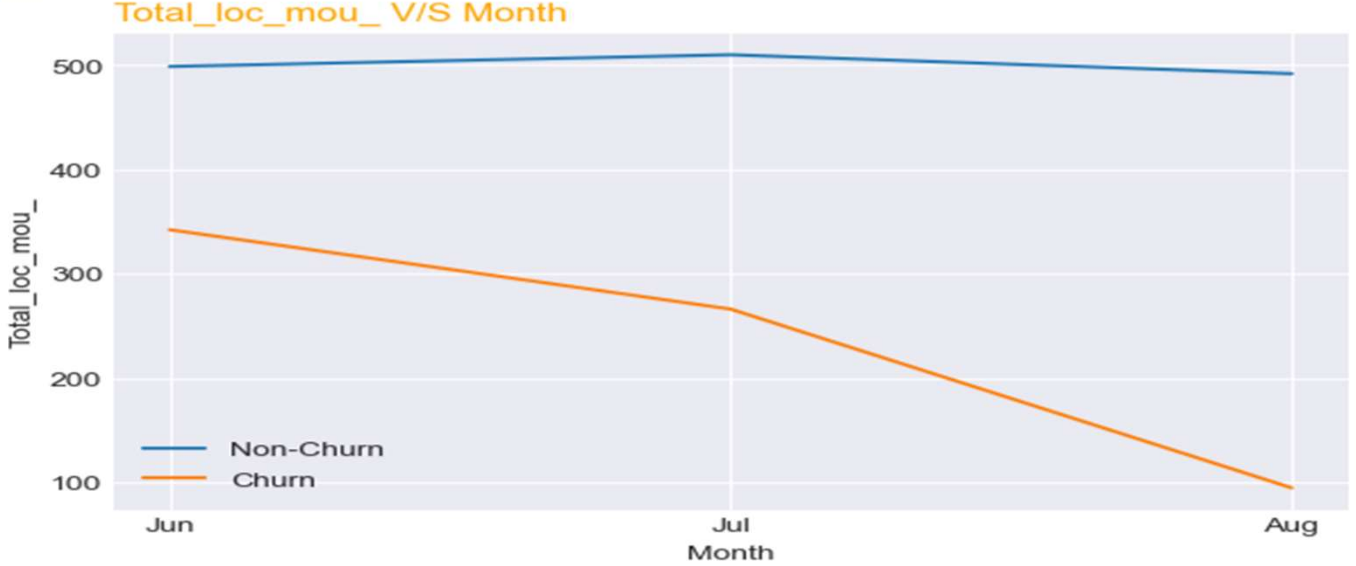
Arpu V/s Churn



	arpu_6	arpu_7	arpu_8
churn			
0	549.843524	563.190828	533.052496
1	660.695411	543.722952	238.631887

- Average revenue per user,arpu also drops significantly for churners from month Jul(6) to Aug(7).
- While it remains almost consistent for the non-churners.

Total_loc_mou V/s Month

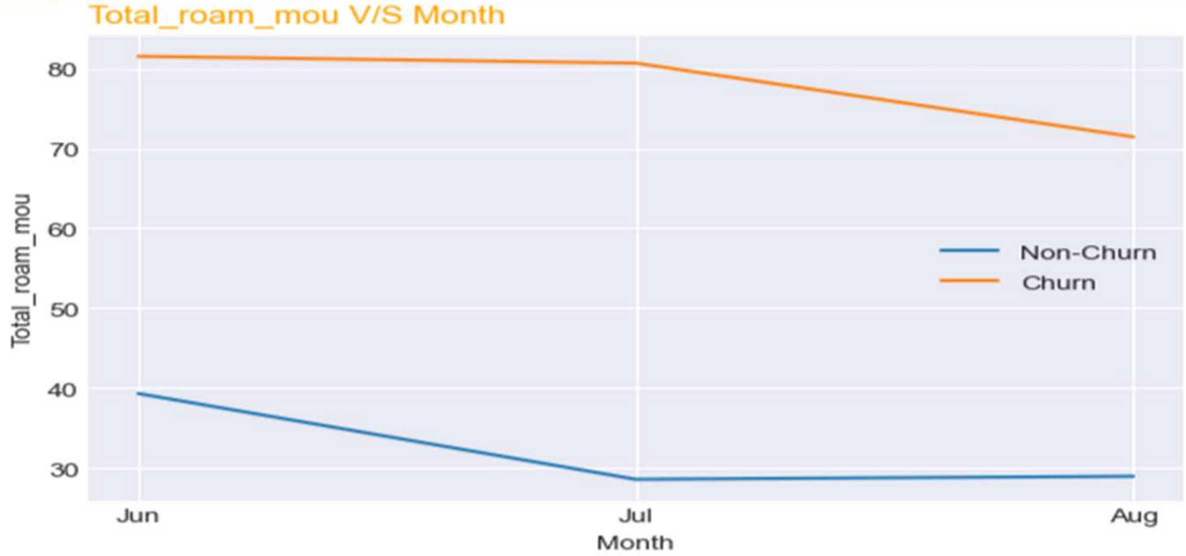


churn	Total_loc_mou_6	Total_loc_mou_7	Total_loc_mou_8
0	498.548969	509.835211	491.705600
1	342.113462	266.025666	94.701154

It can be observed that,

- The Total local call mou is genrally low for churners right from the begining of the good phase.
- local mou pattern for the non-churners remains almost constant through out the 3 months.
- The churners genrally show a low total loc mou but it drops dramatically after the 2nd month.
- This might suggest that people who are not making/reciving much local calls during their tenure are more likely to churn.

Total_roam_mou V/s Month

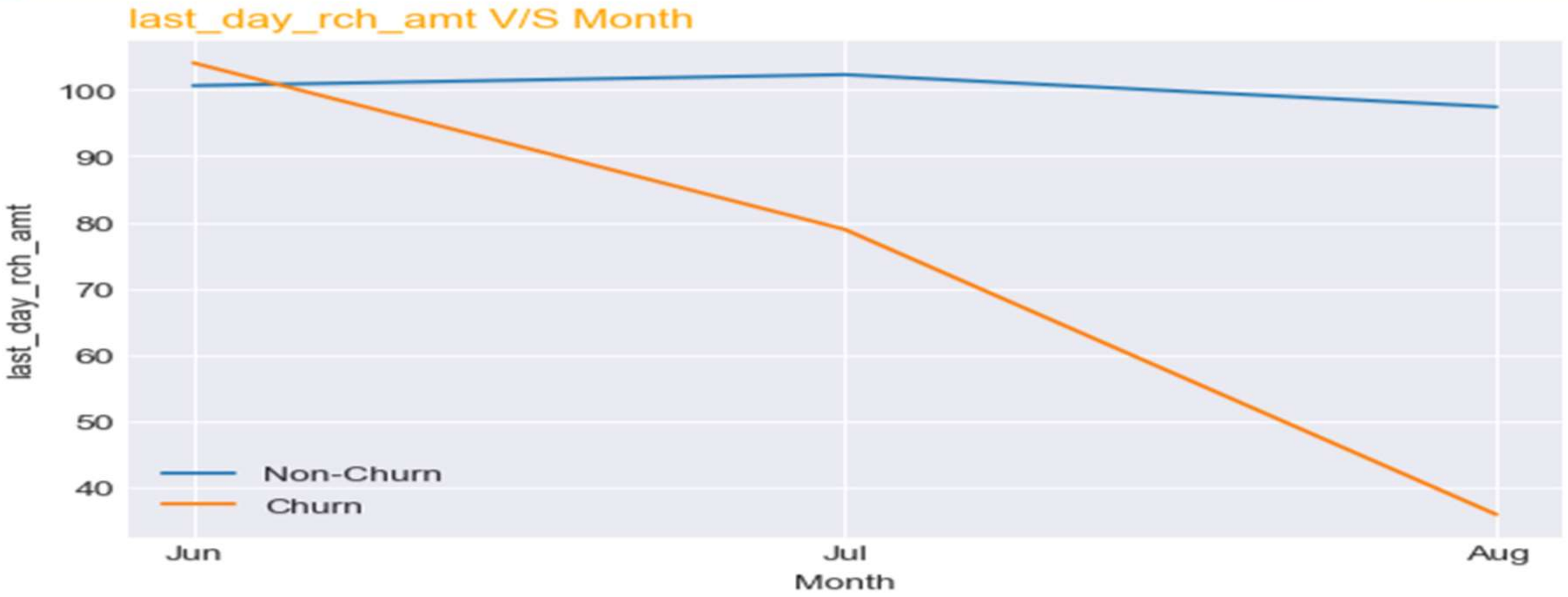


churn	Total_roam_mou_6	Total_roam_mou_7	Total_roam_mou_8
0	39.360033	28.643301	29.016734
1	81.504156	80.651973	71.443623

It can be observed that,

- Surprisingly, the roaming usage of churners is way higher than those of non-churners across all months
- People who are making/reciving more roaming calls during their tenure are more likely to churn.
- This might suggest that the operators roaming tariffs are higher than what are offered by its competitor, thus forming one of the reasons of churn.

Last_day_rch_Amt V/s Month



churn	last_day_rch_amt_6	last_day_rch_amt_7	last_day_rch_amt_8
0	100.657232	102.318284	97.451724
1	104.085194	78.956989	35.955749

- The avg. last recharge amount for churners is less than half the amount of that of the non-churners.
- Suggesting, as the recharge amount reduces for a customer its chances to churn increases.

Modelling

List of Required Libraries for the Modelling as below:

```
import sklearn.preprocessing
from sklearn import metrics
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV

from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
```

Handling Class Imbalance

Standard classifier algorithms like Decision Tree and Logistic Regression have a bias towards classes which have number of instances. They tend to only predict the majority class data. The features of the minority class are treated as noise and are often ignored. Thus, there is a high probability of misclassification of the minority class as compared to the majority class.

Informed Over Sampling: Synthetic Minority Over-sampling Technique

This technique is followed to avoid overfitting which occurs when exact replicas of minority instances are added to the main dataset. A subset of data is taken from the minority class as an example and then new synthetic similar instances are created. These synthetic instances are then added to the original dataset. The new dataset is used as a sample to train the classification models.

Advantages

- Mitigates the problem of overfitting caused by random oversampling as synthetic examples are generated rather than replication of instances
- No loss of useful information

Handling Class Imbalance using SMOTE library

```
print("Before OverSampling, counts of label '1': {}".format(sum(y_train==1)))
print("Before OverSampling, counts of label '0': {} \n".format(sum(y_train==0)))
print("Before OverSampling, churn event rate : {}% \n".format(round(sum(y_train==1)/len(y_train)*100,2)))
```

```
Before OverSampling, counts of label '1': 1700
Before OverSampling, counts of label '0': 19234

Before OverSampling, churn event rate : 8.12%
```

```
from imblearn.over_sampling import SMOTE
sm = SMOTE(random_state=12)
X_train_res, y_train_res = sm.fit_resample(X_train, y_train)
```

```
print('After OverSampling, the shape of train_X: {}'.format(X_train_res.shape))
print('After OverSampling, the shape of train_y: {} \n'.format(y_train_res.shape))

print("After OverSampling, counts of label '1': {}".format(sum(y_train_res==1)))
print("After OverSampling, counts of label '0': {}".format(sum(y_train_res==0)))
print("After OverSampling, churn event rate : {}% \n".format(round(sum(y_train_res==1)/len(y_train_res)*100,2)))
```

```
After OverSampling, the shape of train_X: (38468, 178)
After OverSampling, the shape of train_y: (38468,)

After OverSampling, counts of label '1': 19234
After OverSampling, counts of label '0': 19234
After OverSampling, churn event rate : 50.0%
```

For the prediction of churn customers we will be fitting variety of models and select one which is the best predictor of churn. Models trained are:

- Logistic Regression
- Decision Tree
- Random Forest
- Boosting Models – Gradient Boosting Classifier and XGBoost Classifier
- SVM

1. Logistic Regression:

The resulting model, after PCA and logistic regression (with optimal cutoff setting) has a right balance of different metrics score for sensitivity, specificity and Roc Accuracy on the train and test set.

- `train sensitivity` : 86.47%, `train roc auc score` : 82.1%
- `test sensitivity` : 84.40%, `test roc auc score` : 81.21%

2. Decision Tree:

- Decision tree after selecting optimal cut-off also is resulting in a model with
- `Train Recall` : 89.78% and `Train Roc_auc_score` : 82.40
- `Test Recall` : 78.13% and `Test Roc_auc_score` : 76.56

3. Random Forest:

- Random Forest after selecting optimal cut-off also is resulting in a model with
- `Train Recall` : 88.70% and `Train Roc_auc_score` : 85.60
- `Test Recall` : 77.57% and `Test Roc_auc_score` : 79.65

4. Boosting Models:

4.1 Gradient Boosting Classifier

```
Roc_auc_score : 0.8087663076161942
Sensitivity/Recall : 0.7952646239554317
Specificity: 0.8222679912769566
False Positive Rate: 0.17773200872304337
Positive predictive value: 0.2801766437684004
Negative Predictive value: 0.9788001153735217
sklearn precision score value: 0.2801766437684004
```

	churn	churn_Prob	final_predicted
4265	0	0.307174	1
29221	0	0.056469	0
974	0	0.447631	1
1602	0	0.023201	0
10225	0	0.016619	0

This model is litrally over-fitting the Training data with a lower performance on the Test data.

4.2 XGBoost Classifier:

```
Roc_auc_score : 0.806017408289591
Sensitivity/Recall : 0.754874651810585
Specificity: 0.857160164768597
False Positive Rate: 0.14283983523140295
Positive predictive value: 0.31493317838466006
Negative Predictive value: 0.9757274858640188
sklearn precision score value: 0.31493317838466006
```

5. SVM

5.1 Using Linear Kernel

```
Roc_auc_score : 0.7431706953259094
Sensitivity/Recall : 0.5793871866295265
Specificity: 0.9069542040222922
False Positive Rate: 0.09304579597770778
Positive predictive value: 0.35135135135135137
Negative Predictive value: 0.9612223934257833
sklearn precision score value: 0.35135135135135137
```

5.2 Using Non Linear Kernal:

```
Roc_auc_score : 0.7431706953259094
Sensitivity/Recall : 0.5793871866295265
Specificity: 0.9069542040222922
False Positive Rate: 0.09304579597770778
Positive predictive value: 0.35135135135135137
Negative Predictive value: 0.9612223934257833
sklearn precision score value: 0.35135135135135137
```

Final Choice of Model

Recall is the most important business metric for the telecom churn problem. The company would like to identify most customers at risk of churning, even if there are many customers that are misclassified as churn. The cost to the company of churning is much higher than having a few false positives.

Overall, the `Logistic Regression` model with probability cut-off = 0.45, performs best. It achieved the best recall accuracy of 84.4% for test data. Also the overall accuracy and specificity is consistent for Test and train data, thus avoiding overfitting. The precision is compromised in this effort but the business objective to predict Churn customers is most accuratety captured by it.

Some of the top main predictors of churns for the action phase(3rd Month August) are Identified using Random Forest are as follows:

1. `total_ic_mou_8` -- Total incoming minutes of usage in month 8
2. `loc_ic_mou_8` -- local incoming minutes of usage in month 8
3. `total_month_rech_8` -- Total month recharge amount in month 8
4. `total_roam_mou_8` -- Total incoming+outgoing roaming minutes of usage in month 8
5. `loc_ic_t2m_mou_8` -- local incoming calls to another operator minutes of usage in month 8
6. `roam_og_mou_8` -- outgoing roaming calls minutes of usage in month 8
7. `Total_loc_mou_8` -- Total local minutes of usage in month 8
8. `roam_ic_mou_8` -- incoming roaming calls minutes of usage in month 8
9. `total_rech_amt_8` -- total recharge amount in month 8
10. `loc_ic_t2t_mou_8` -- local incoming calls from same operator minutes of usage in month 8
11. `max_rech_amt_8` -- maximum recharge amount in month 8
12. `last_day_rch_amt_8` -- last (most recent) recharge amount in month 8
13. `arpu_8` -- average revenue per user in month 8
14. `loc_og_mou_8` -- local outgoing calls minutes of usage in month 8
15. `loc_og_t2n_mou_8` -- local outgoing calls minutes of usage to other operator mobile in month 8
16. `av_rech_amt_data_8` -- average recharge amount for mobile data in month 8
17. `total_rech_data_8` -- total data recharge (MB) in month 8
18. `total_og_t2t_mou_8` -- total outgoing calls from same operator minutes of usage in month 8
19. `total_rech_num_8` -- total number of recharges done in the month 8
20. `total_rech_amt_data_8` -- total recharge amount for data in month 8
21. `max_rech_data_8` -- maximum data recharge (MB) in month 8
22. `avg_rech_amt_8` -- average recharge amount in month 8
23. `fb_user_8` -- services of Facebook and similar social networking sites for month 8
24. `vol_data_mb_8` -- volume of data (MB) consumed for month 8
25. `count_rech_2g_8` -- Number of 2g data recharge in month 8
26. `loc_og_to_ic_mou_8` -- local outgoing to incoming mou ratio for month of 8
27. `spl_og_mou_7` -- Special outgoing call for the month of 7

Local calls Mou's be it incoming or outgoing have a very important role for churn predictions. Reduction in these KPI's forms a clear indicator of churn.

Overall, drop in any of these indicator KPI is a signal that the customer is not actively engaging in the services offered by the Network operator and thus may choose to churn in the near future.

Next, we will look at some of the stratergic steps which can be taken to retain these predicted churners.

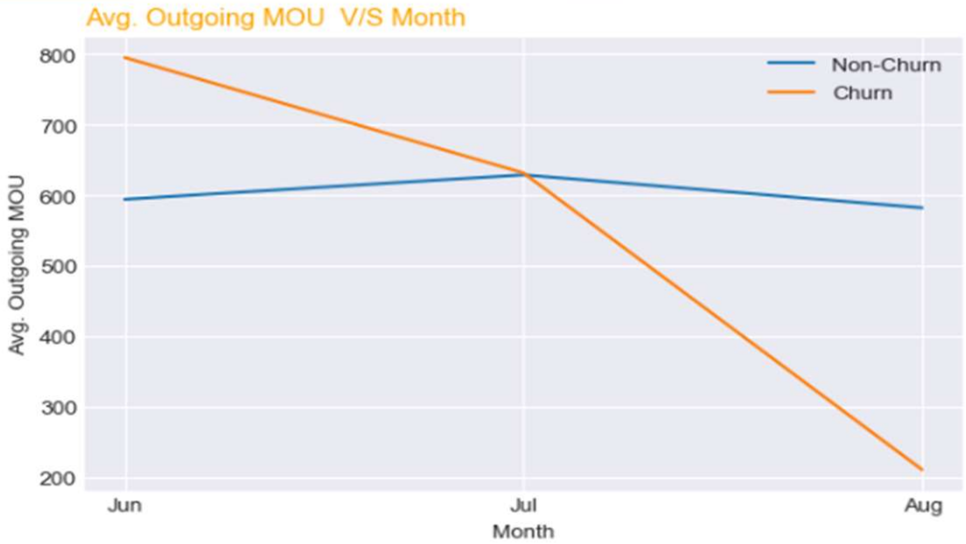
Strategies to manage customer churn

- It is a fact that it costs 5-10 times more to acquire a new customer than to retain an existing one, customer retention has now become even more important than customer acquisition.
- For many incumbent operators, retaining high profitable customers is the number one business goal.

Monitoring Drop in usage

1. Customer churn seems to be well predicted by drop in usage.
2. Aside from using the Machine Learning model for predicting churn, the telecom company should pay close attention to drop in MoU, ARPU and data usage (2g and 3g) month over month. If feasible, the company should track these numbers week over week. Since billing cycles are typically monthly, a drop in usage numbers will give the company time to react when tracked at weekly level.
3. Contact these customers proactively to find out what's affecting their experience. Perhaps, offer them coupons or other incentives to continue to use the services, while the company fixes the issues reported.
4. Marketing team must come up with campaigns which targets these high-value to-be churner.

Improving Outgoing Services:

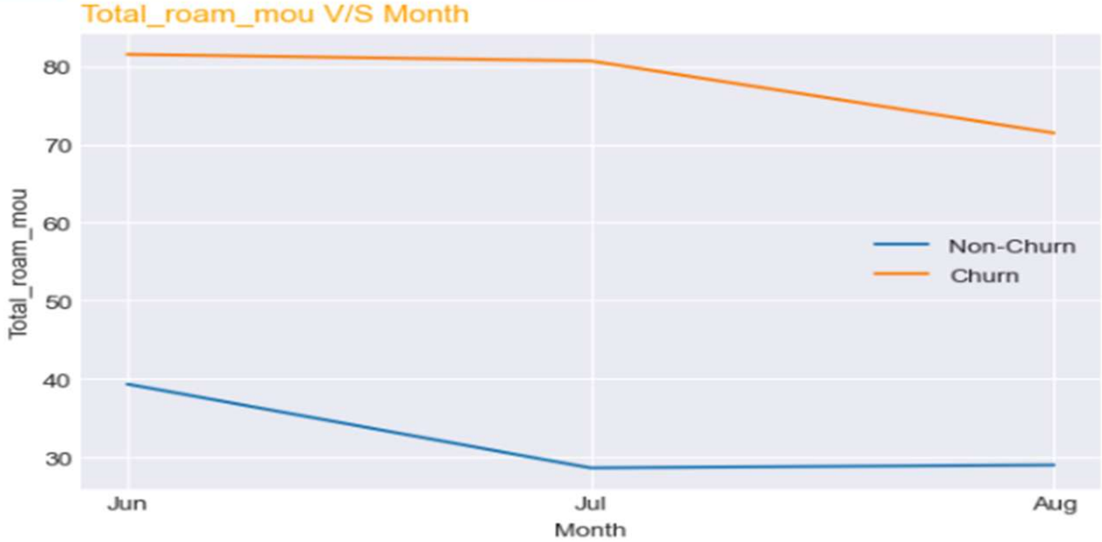


- Initially, churner's outgoing usage was more than that of non-churners. Gradually they dropped there outgoing usage. May be these customers din't like the outgoing services offered to them or may be the call tariffs seemed expensive to them or may be the overall call quality, network coverage was not liked my them. This could be further investigated by the network service provider.

Stratergy suggestions,

- The Network operators must futher investigate their outgoing tariffs, plans and campaigns.
- Might be that the outgoing tariffs offered to it's customer are less competitive to the outgoing tariffs of their competitor.
- New campaigns which targets the customers with high outgoing usage be rolled out.Like,
  - Discounted outgoing rates during particular hours of the day for these customers.
  - For every X mou, grant customer with some % of X free mou.
  - Investigate and if need be revise the outgoing tarrifs to make it competitive.
  - Free monthly outgoing mou's depending on the users past roaming mou usage.

Improving Roaming Services:



churn	Total_roam_mou_6	Total_roam_mou_7	Total_roam_mou_8
0	39.360033	28.643301	29.016734
1	81.504156	80.651973	71.443623

Stratergy suggestions,

- Churners show higher roaming usage than non-churners.
- The Network operators must futher investigate their roaming tariffs, and quality of service.
- Might be that the roaming tariffs offered are less competitive than their competitor.
- It might be that the customer is not getting good quality of service while roaming. In this case, quality of service guarantees with roaming partners and network quality need to be investigated.
- New campaigns which targets the roaming customers can be rolled out. Like,
  - Discounted roaming rates during particular hours of the day.
  - Free monthly roaming mou's depending on the users past roaming mou usage.