

Telecom Churn Case Study

Team members - Sanjeebani Swain

PROBLEM STATEMENT

The 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. Customer retention is better than new acquisition

To reduce customer churn, telecom companies need to **predict which customers are at high risk of churn**. In this project, you will analyse 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 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.
- 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 behaviour during churn will be helpful.

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:

- 1.The 'good' phase: In this phase, the customer is happy with the service and behaves as usual.
- 2.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 behaviour 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.)
- 3.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 fourth month is the 'churn' phase.

APPROACH

- Introduction
- Problem Statement
- Data Understanding
- Exploratory Data Analysis
- Feature Engineering
- Preprocess data (convert columns to appropriate formats, handle missing values, etc.)
- Conduct appropriate exploratory analysis to extract useful insights (whether directly useful for business or for eventual modeling/feature engineering).
- Derive new features if you need.
- Train a variety of models, tune model hyperparameters, etc. (handle class imbalance using appropriate techniques).
- Evaluate the models using appropriate evaluation metrics.
- Finally, choose a model based on an evaluation metric with proper justification.
- Recommendation for business strategy

DATA MANIPULATION

- There are 226 attributes and 99,999 records in the telecom data churn csv file for analysis purpose
- Attributes that has a lot of missing values and single value in it doesn't make sense for analysis , hence dropped
- Impute roam_ic, loc_ic, std_ic, spl_ic, isd_ic, ic_others as 0 as total_ic_mou is 0
- Attributes with >40% nulls in it were removed, as they may not account more into the EDA & models
- Fill null values for data pack recharge amount and count to 0
- Even after doing the previous steps , there were huge number of attributes which had null values in it but removing them was not logical at this stage .However they will be removed from model if found insignificant
- The total amount of recharge for talktime is missing and has to be calculated from the average and number of recharges
- total_rech_6/_7/_8/_9 were calculated as summation of the total_data_rech_amt_* & total_rech_amt_* fields
- High value customer were tagged from the top 70 percentile from the data set to proceed further
- Also the churn indicator was tagged /derived for each customer based on the fields (total_ic_mou_9, total_og_mou_9, vol_2g_mb_9, vol_3g_mb_9) i.e. Those who have not made any calls (either incoming or outgoing) AND have not used mobile internet even once in the churn phase

EDA Summary -Correlation Metrix

data_corr

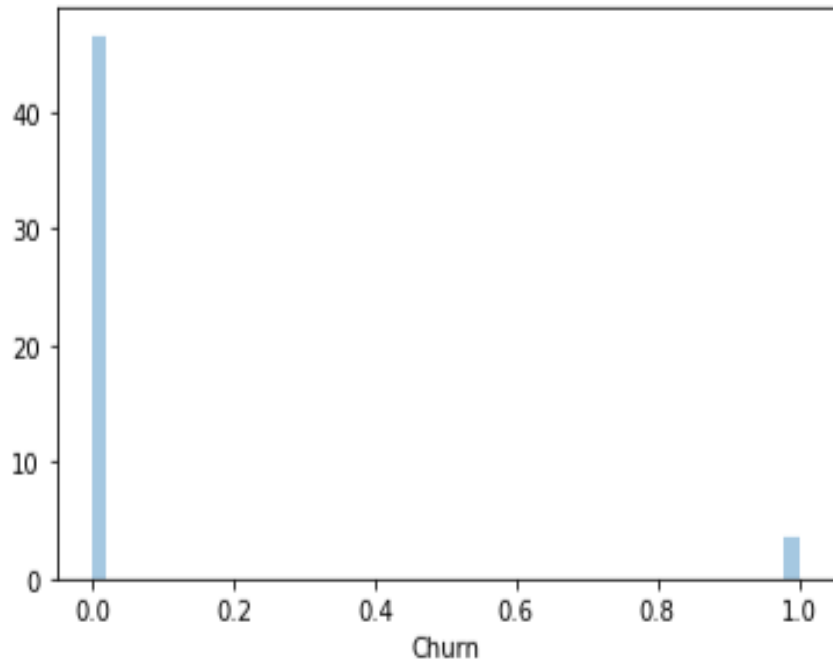
	mobile_number	arpu_6	arpu_7	arpu_8	onnet_mou_6	onnet_mou_7	onnet_mou_8	offnet_mou_6	offnet_mou_7	offnet_mou_8	Churn
mobile_number	1.000000	0.034817	0.030741	0.034163	0.010159	0.005386	0.007863	0.023784	0.013254	0.019429	0.000000
arpu_6	0.034817	1.000000	0.703421	0.645781	0.347725	0.225080	0.196474	0.522732	0.356846	0.306076	0.000000
arpu_7	0.030741	0.703421	1.000000	0.771531	0.212763	0.317806	0.268496	0.356104	0.488919	0.395434	0.000000
arpu_8	0.034163	0.645781	0.771531	1.000000	0.152120	0.229522	0.340762	0.283522	0.375256	0.519533	0.000000
onnet_mou_6	0.010159	0.347725	0.212763	0.152120	1.000000	0.758100	0.628326	0.083029	0.034522	0.031210	0.000000
onnet_mou_7	0.005386	0.225080	0.317806	0.229522	0.758100	1.000000	0.628326	0.083029	0.034522	0.031210	0.000000
onnet_mou_8	0.007863	0.196474	0.268496	0.340762	0.628326	0.628326	1.000000	0.083029	0.034522	0.031210	0.000000
offnet_mou_6	0.023784	0.522732	0.356104	0.283522	0.083029	0.034522	0.031210	1.000000	0.034522	0.031210	0.000000
offnet_mou_7	0.013254	0.356846	0.488919	0.375256	0.034522	0.031210	0.031210	0.034522	1.000000	0.031210	0.000000
offnet_mou_8	0.019429	0.306076	0.395434	0.519533	0.031210	0.031210	0.031210	0.031210	0.031210	1.000000	0.000000
total_data_rech_amt_8	-0.008026	0.005011	0.030687	0.127511	-0.077633	-0.075334	-0.057268	-0.090106	-0.089527	-0.052715	0.000000
total_rech_6	-0.004484	0.420108	0.237058	0.223716	0.029399	-0.015630	-0.012257	0.078710	0.015844	0.017005	0.000000
total_rech_7	-0.004667	0.221932	0.420822	0.295395	-0.021851	0.017800	0.023930	0.002850	0.058047	0.048254	0.000000
total_rech_8	0.003750	0.218514	0.283556	0.438694	-0.020381	0.006633	0.059343	0.011315	0.041022	0.122140	0.000000
Churn	-0.024566	0.056565	0.006278	-0.132059	0.081520	0.033259	-0.060836	0.067195	0.020928	-0.100166	1.000000

- Highly correlated fields have been dropped from the data set before moving ahead for model building

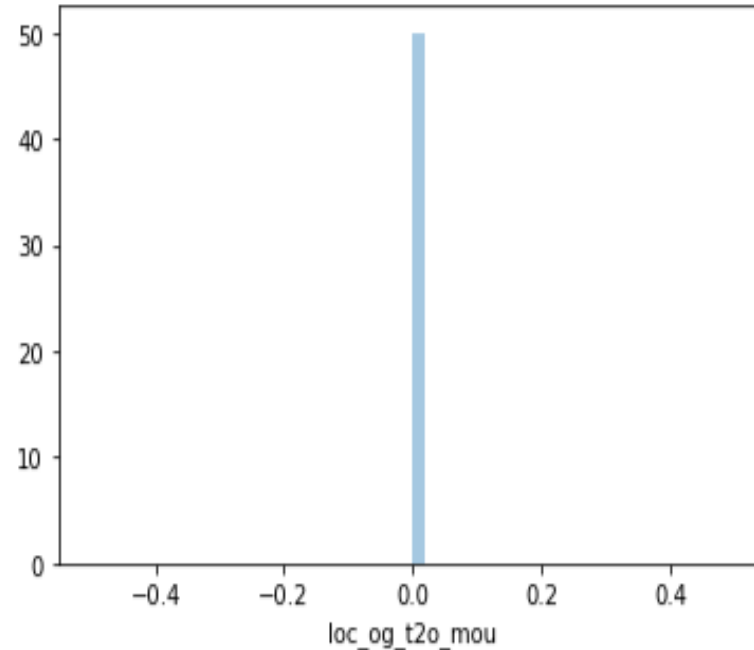
EDA Summary -contd.

```
sns.distplot(data1["Churn"],kde=True)
```

<matplotlib.axes._subplots.AxesSubplot at 0x22869837d30:



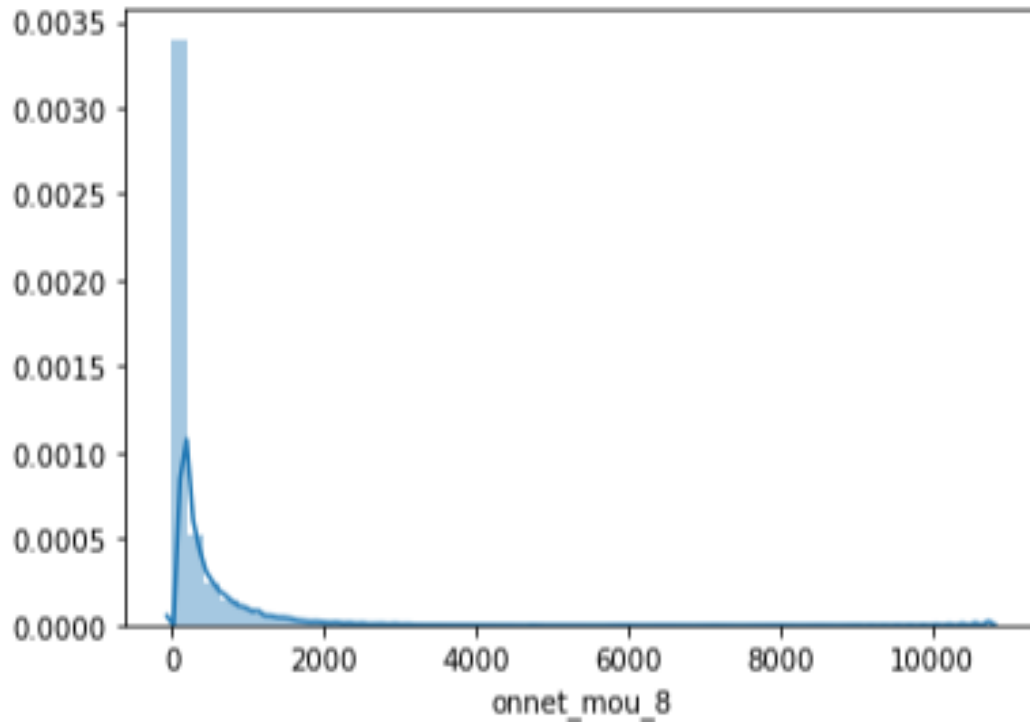
```
: sns.distplot(data2['loc_og_t2o_mou'])  
plt.show()
```



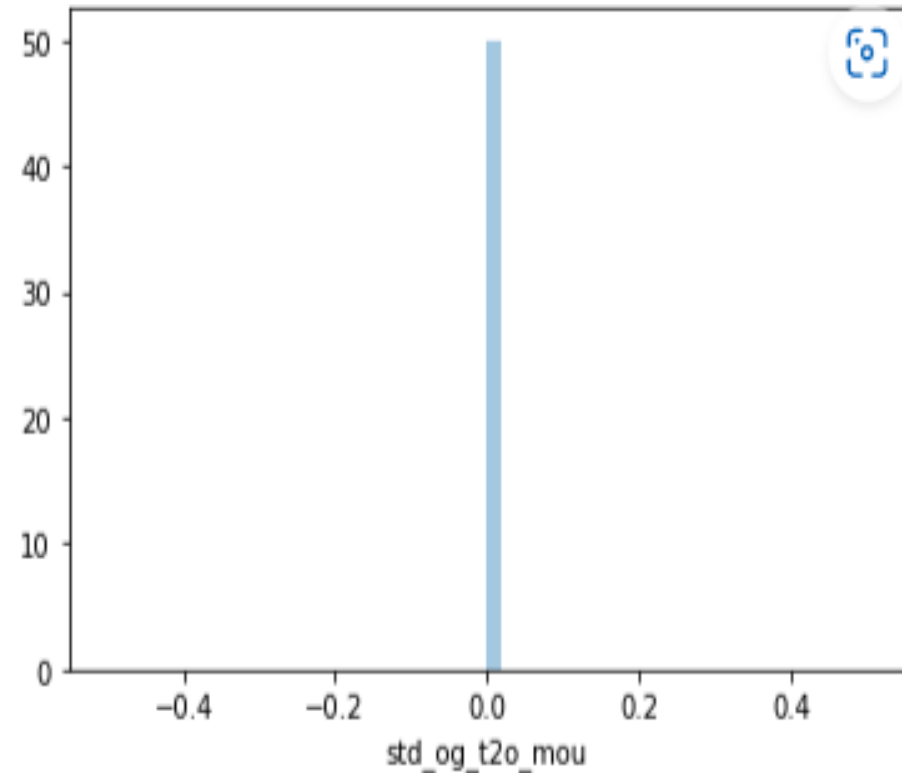
- Plot for the churn is highly imbalanced and should be balanced before proceeding for modelling

EDA Summary -contd

```
sns.distplot(data2['onnet_mou_8'])  
plt.show()
```



```
: sns.distplot(data2['std_og_t2o_mou'])  
plt.show()
```



- For omnet_month of usage (August) is skewed mostly towards left

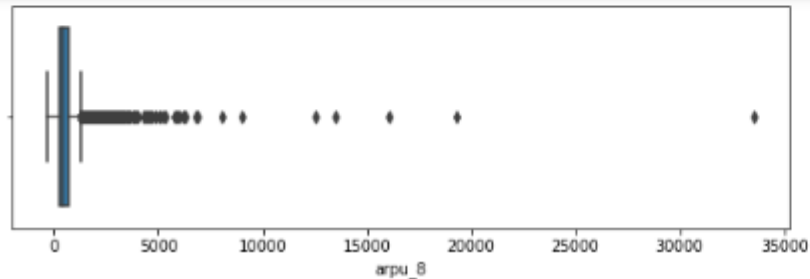
EDA - Numerical variables

```
In [58]: # List of features to be analyzed
col_boxplot = ['arpu_8', 'loc_og_mou_8', 'max_rech_amt_8', 'last_day_rch_amt_8', 'aon', 'total_mou_8',
               'gd_ph_loc_ic_mou', 'gd_ph_last_day_rch_amt', 'gd_ph_std_og_mou', 'gd_ph_max_rech_amt',
               'gd_ph_loc_og_mou', 'gd_ph_arpu']

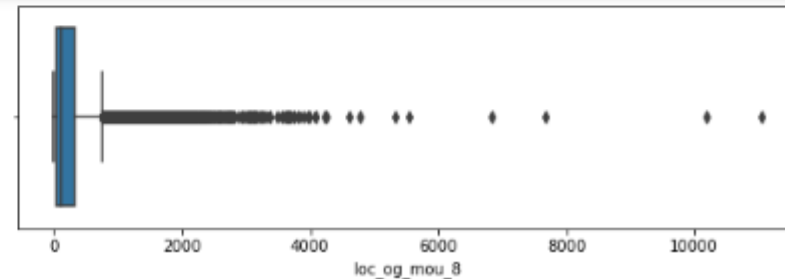
# Plot boxplots for each variable
fig, axes = plt.subplots(6, 2, figsize=(20, 20))

for index, col in enumerate(col_boxplot):
    i, j = divmod(index, 2)
    sns.boxplot(data_churn[col], ax=axes[i, j])

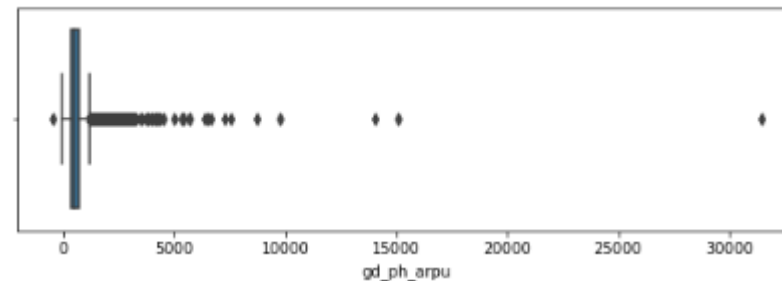
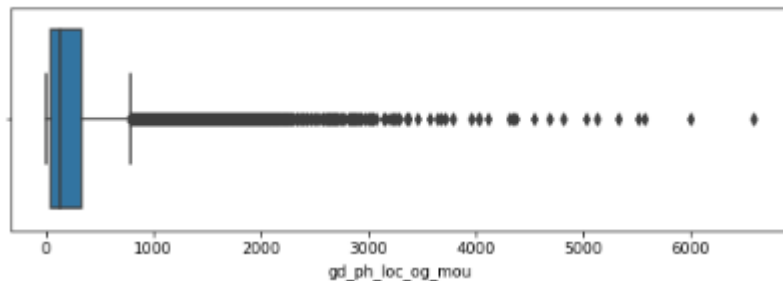
plt.subplots_adjust(hspace=0.3)
plt.show()
```



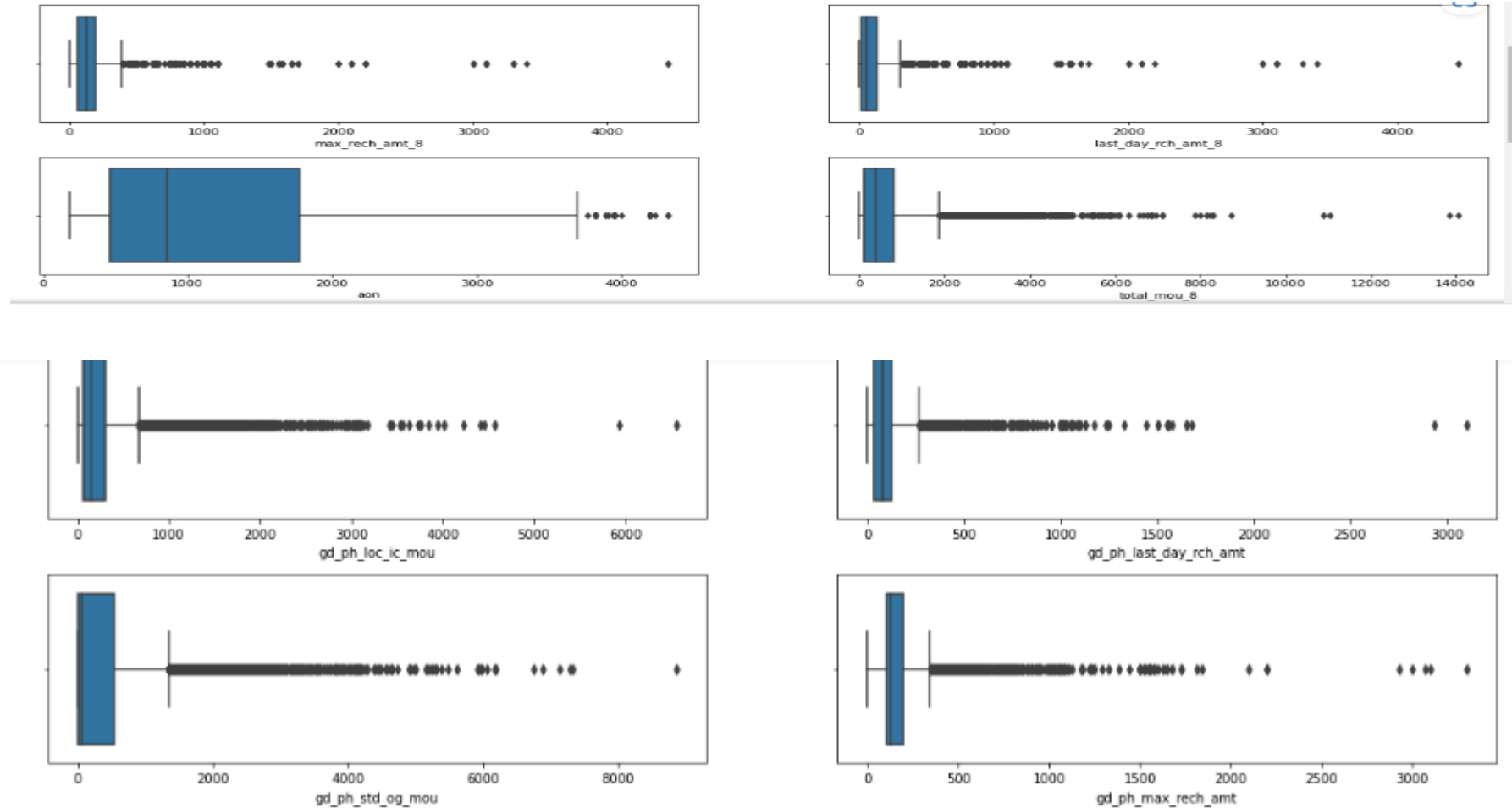
gd_ph_std_og_mou



gd_ph_max_rech_amt



EDA - Numerical variables contd..

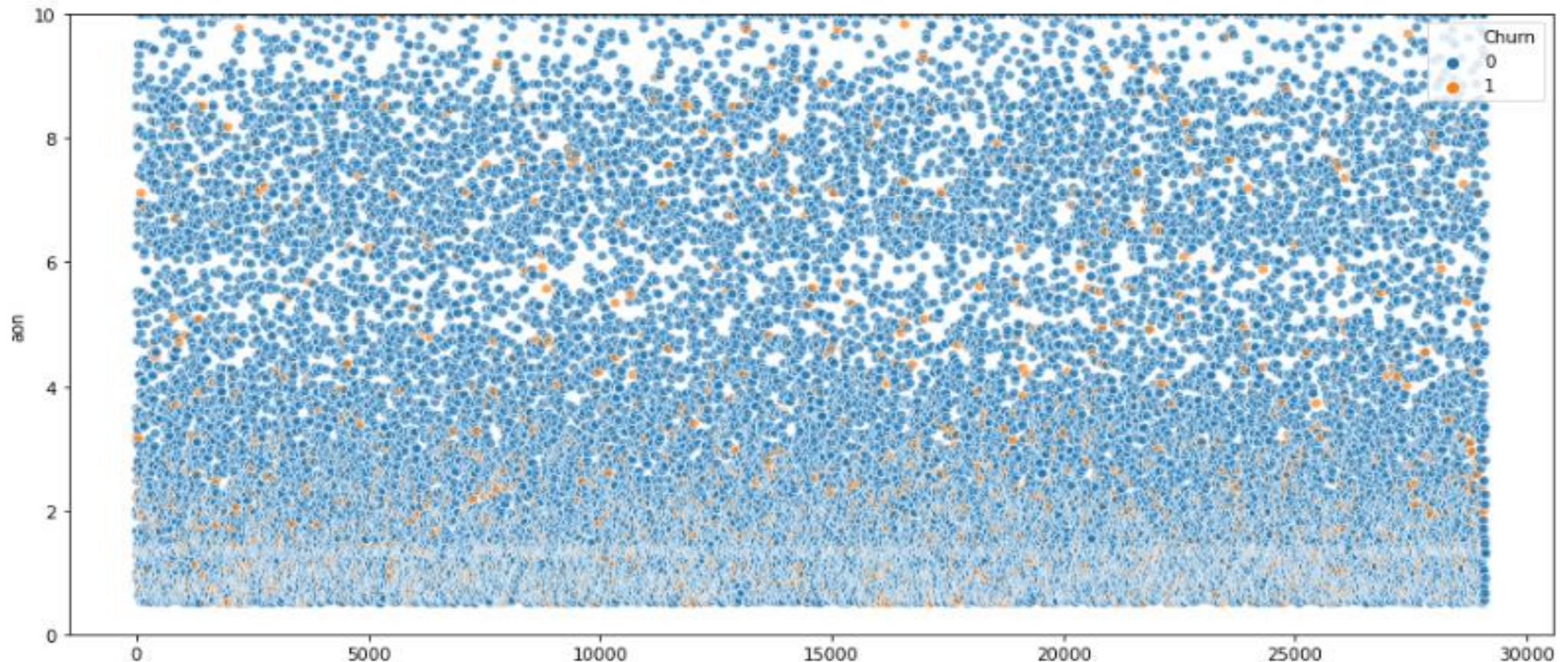


- We can see almost every column has some outliers, while most of them are because there are 0.0 as the service was not used some are actual outliers
- Since we don't have actual business people to check the factfulness of the data, we will cap those features

EDA -contd..

```
In [59]: plt.figure(figsize=(15,7))  
sns.scatterplot(y=data_churn['aon'] / 365, x=data_churn.index, hue=data_churn.Churn, alpha=0.7)  
plt.ylim(0,10)
```

Out[59]: (0.0, 10.0)



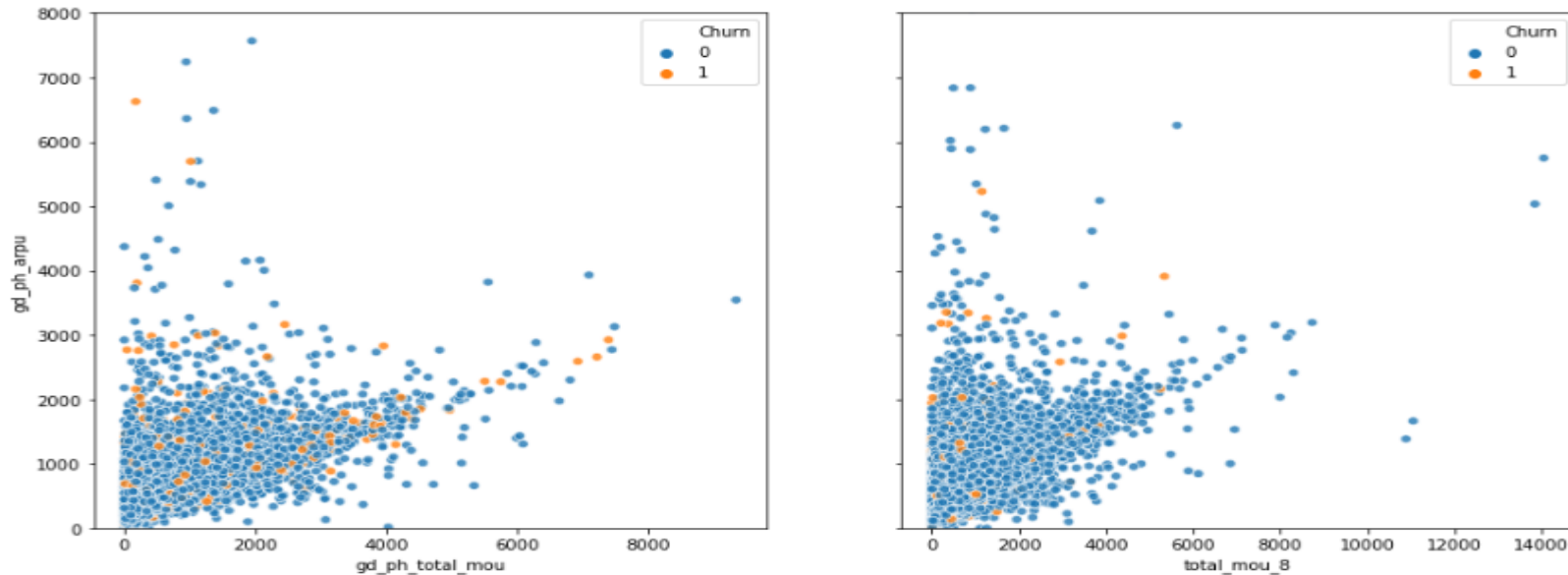
- Scatterplot above shows like most of the customer would churn some where around 4 months tenure

EDA -contd..

```
## Analyzing Volume based cost vs revenue impact
```

```
fig, axes = plt.subplots(1, 2, sharey=True, figsize=(15, 7))
sns.scatterplot(y='gd_ph_arpu', x='gd_ph_total_mou', data=data_churn, ax=axes[0], hue='Churn', alpha=0.8)
sns.scatterplot(y='arpu_8', x='total_mou_8', data=data_churn, ax=axes[1], hue='Churn', alpha=0.8)

plt.ylim(0,8000)
plt.show()
```



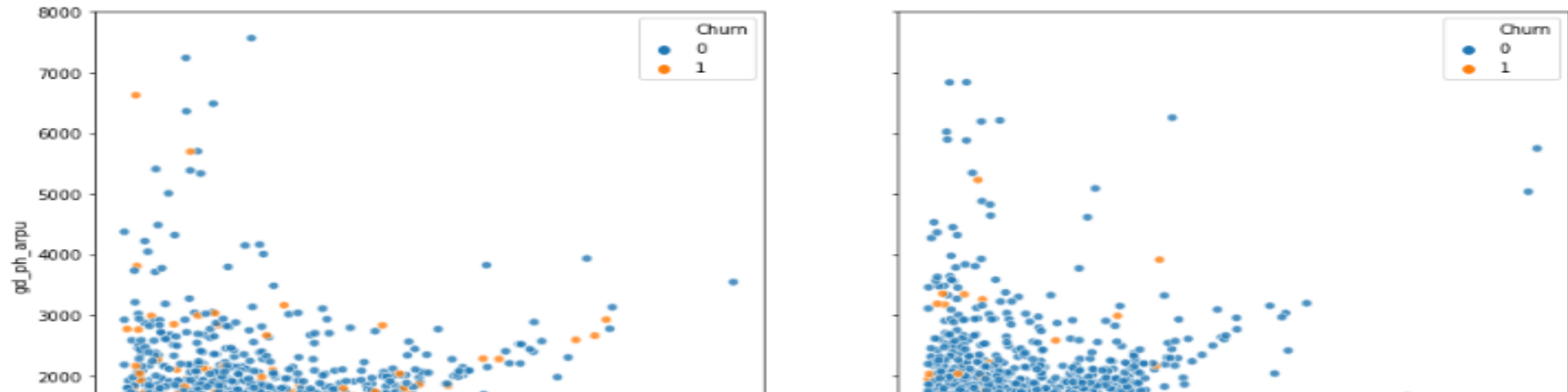
- We can clearly see that MOU have dropped significantly for the churners in the action pahse i.e 8th month, thus hitting the revenue generated from them
- It is also interesting that though the MOU is between 0-2000, the revenue is highest in that region that tells us these users had other services that were boosting the revenue

EDA -contd..

In [60]: *## Analyzing Volume based cost vs revenue impact*

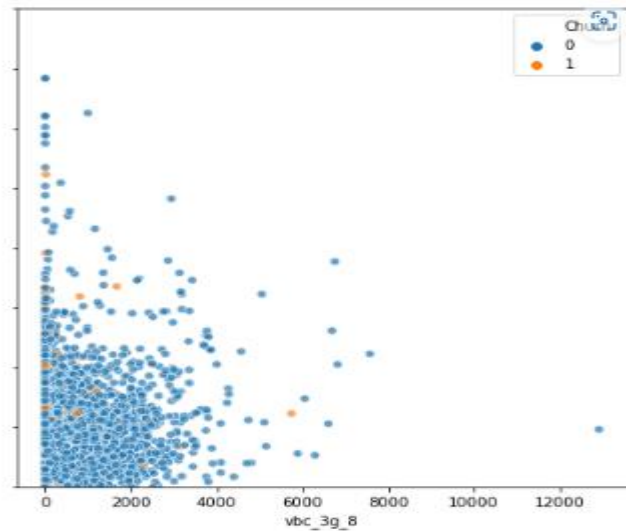
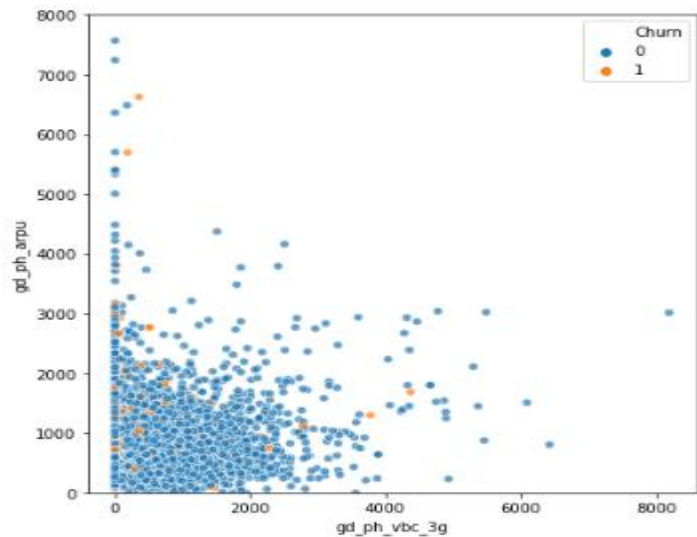
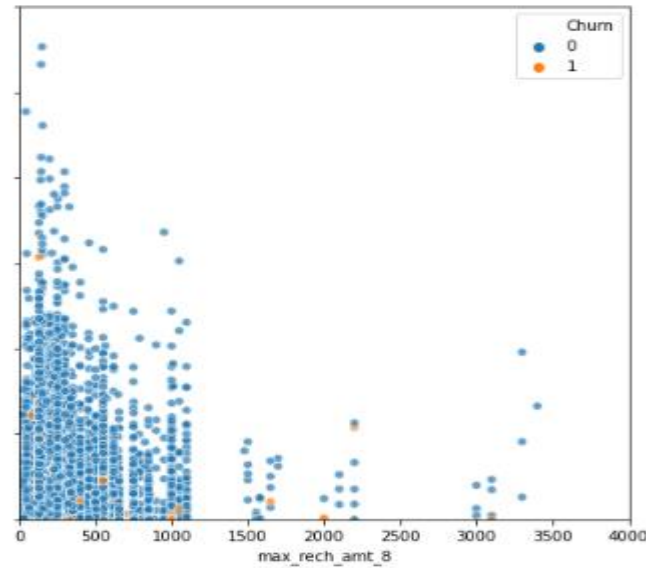
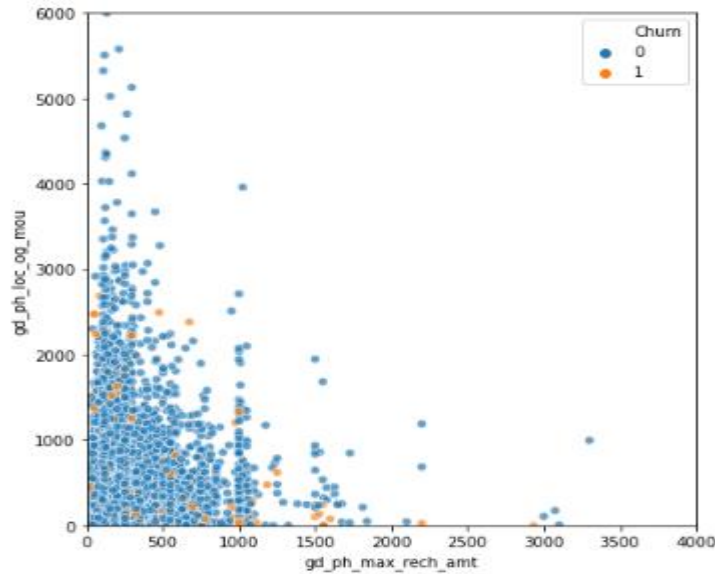
```
fig, axes = plt.subplots(1, 2, sharey=True, figsize=(15, 7))
sns.scatterplot(y='gd_ph_arpu', x='gd_ph_total_mou', data=data_churn, ax=axes[0], hue='Churn', alpha=0.8)
sns.scatterplot(y='arpu_8', x='total_mou_8', data=data_churn, ax=axes[1], hue='Churn', alpha=0.8)

plt.ylim(0,8000)
plt.show()
```



- We can see that the users who were using very less amount of VBC data and yet were generating high revenue churned
- Yet again we see that the revenue is higher towards the lesser consumption side

EDA -contd..



- Users who were recharging with high amounts were using the service for local uses less as compared to user who did lesser amounts of recharge
- Intuitively people whose max recharge amount as well as local out going were very less even in the good phase churned more
- We can see that the users who were using very less amount of VBC data and yet were generating high revenue churned
- Yet again we see that the revenue is higher towards the lesser consumption side
- Also from scatter plot users who had the max recharge amount less than 250 churned more

Data Preparation

- The outliers were capped based on the box plot results
- Importing the necessary python libraries to perform logistic regression
- Splitting the data set into Train and Test (30-70 ratio) using the train_test_split() function
- Handling the class imbalance using SMOTE
 - Note- I encountered problem and tried installing the imblearn ,but failed

```
In [71]: # Use SMOTE to take care of class imbalance
from imblearn.over_sampling import SMOTE

sm = SMOTE(random_state=42)
X_res, y_res = sm.fit_resample(X, y)
```

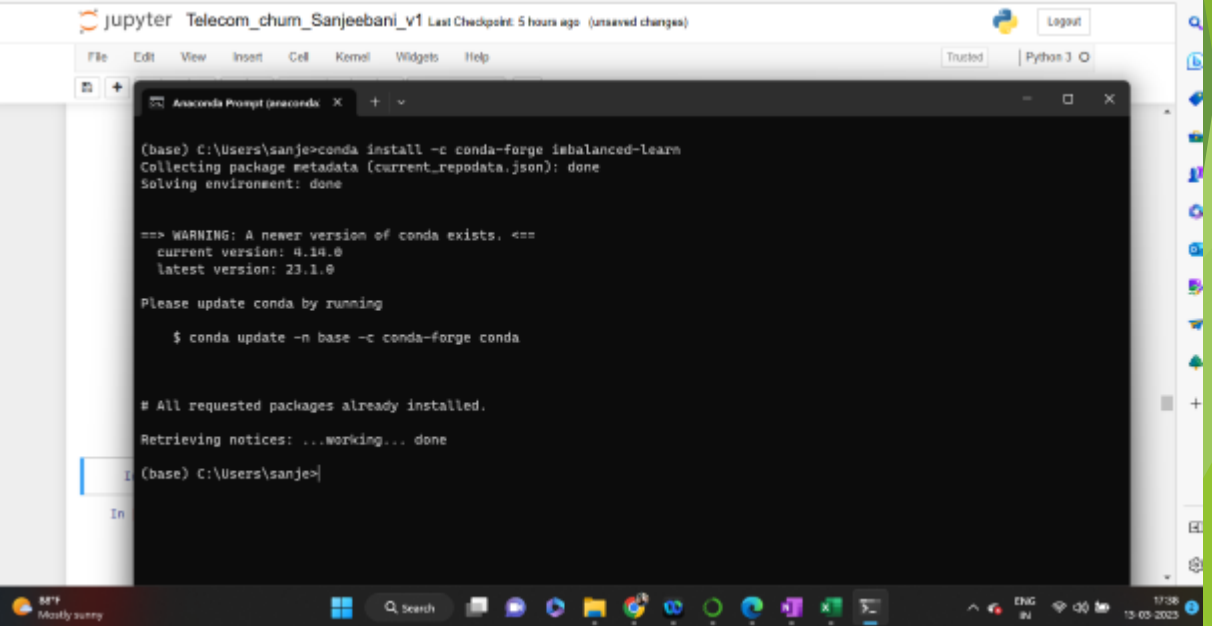
```
-----
ImportError                                Traceback (most recent call last)
<ipython-input-71-ab825f682736> in <module>
----> 1 from imblearn.over_sampling import SMOTE
      2
      3 sm = SMOTE(random_state=42)
      4 X_res, y_res = sm.fit_resample(X, y)

~\anaconda3\lib\site-packages\imblearn\__init__.py in <module>
    35 import types
    36
----> 37 from . import combine
    38 from . import ensemble
    39 from . import exceptions

~\anaconda3\lib\site-packages\imblearn\combine\__init__.py in <module>
      3 """
      4
----> 5 from ._smote_enn import SMOTEENN
      6 from ._smote_tomek import SMOTETomek
      7

~\anaconda3\lib\site-packages\imblearn\combine\_smote_enn.py in <module>
      9
     10 from ..base import BaseSampler
----> 11 from ..over_sampling import SMOTE
     12 from ..over_sampling.base import BaseOverSampler
     13 from ..under_sampling import EditedNearestNeighbours

~\anaconda3\lib\site-packages\imblearn\over_sampling\__init__.py in <module>
      6 from ._adasyn import ADASYN
      7 from .random_over_sampler import RandomOverSampler
```



The screenshot shows a Jupyter Notebook window titled "jupyter Telecom_churn_Sanjeebani_v1". Below the notebook interface, a terminal window is open, displaying the following text:

```
(base) C:\Users\sanje>conda install -c conda-forge imbalanced-learn
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
current version: 4.14.0
latest version: 23.1.0

Please update conda by running

    $ conda update -n base -c conda-forge conda

# All requested packages already installed.

Retrieving notices: ...working... done
(base) C:\Users\sanje>
```

Model Building

- Importing the necessary python libraries to perform logistic regression ,Decision Tree and other models run
- Feature Scaling through Standard scaler
- These ML algorithms were run and tested
 - PCA modelling using PCA module
 - Logistic Regression
 - Decision Tree
 - Random Forest
 - Adaboost
- Finding the most relevant features for further analysis through RFE using sklearn with 15 variables as output

```
rfe_col = X.columns[rfe.support_]
rfe_col
```

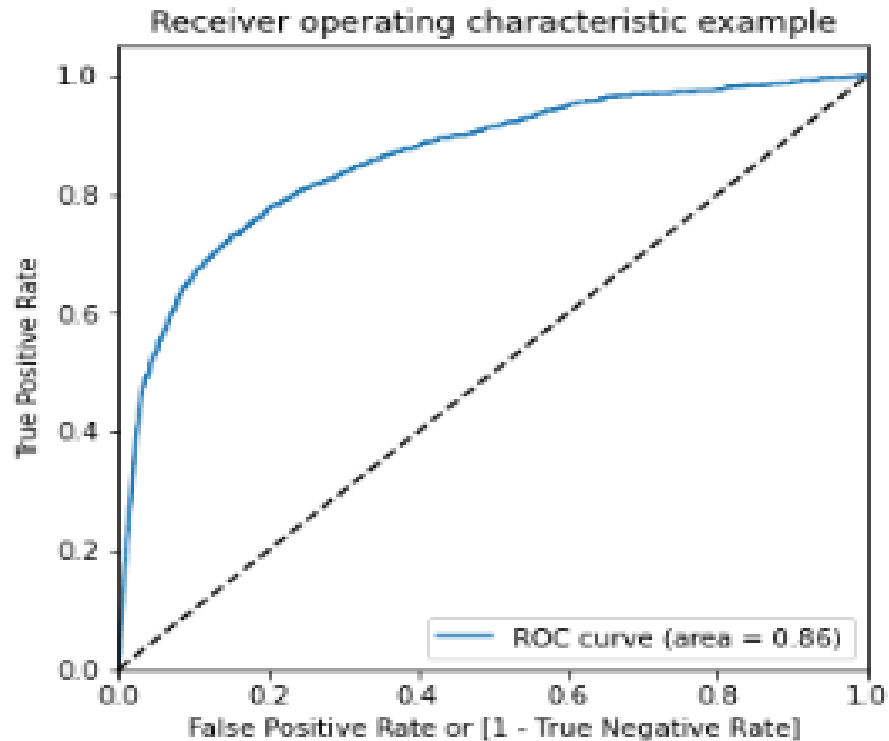
```
: Index(['loc_og_t2c_mou_8', 'std_og_t2f_mou_8', 'spl_og_mou_8', 'og_others_8',
        'loc_ic_mou_8', 'std_ic_t2f_mou_8', 'spl_ic_mou_8', 'total_rech_num_8',
        'total_rech_data_8', 'count_rech_2g_8', 'count_rech_3g_8',
        'monthly_2g_8', 'sachet_2g_8', 'sachet_3g_8', 'sep_vbc_3g',
        'gd_ph_sachet_2g', 'gd_ph_spl_og_mou', 'gd_ph_total_rech_data',
        'gd_ph_monthly_3g', 'gd_ph_sachet_3g', 'gd_ph_count_rech_3g',
        'gd_ph_count_rech_2g', 'gd_ph_monthly_2g', 'gd_ph_spl_ic_mou',
        'gd_ph_loc_og_t2c_mou'],
        dtype='object')
```

- Running the logistic model on the test data set for the iteratively and checking the pvalue & VIF in each step until optimal is reached (variables are iteratively removed whose p value>0.05 or VIF >5)
- Predictions are made on the test data set . With the current cut off as 0.5 we have around 79% accuracy.
- But this may not be optimal .hence use the ROC curve to determine the optimal threshold for predictions

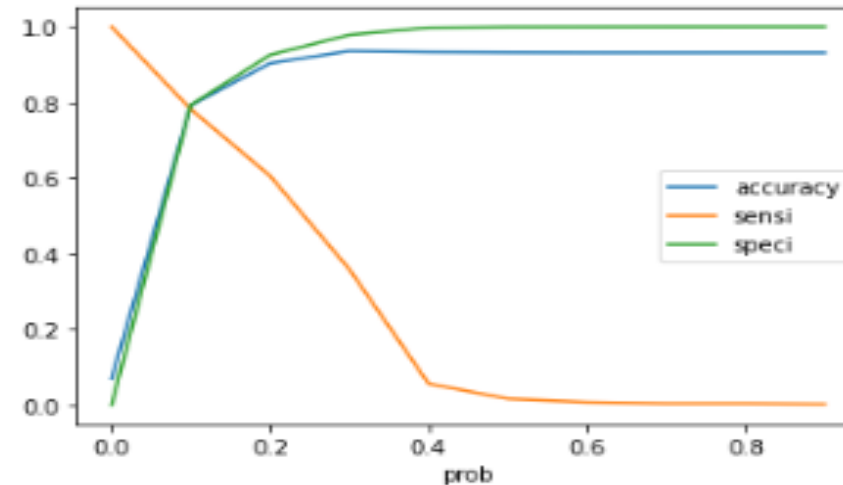
Confusion Matrix and Accuracy

- **'Confusion Matrix'** - shows a comparison of the predicted and actual labels
- Metrics used to evaluate the performance of the logistic regression -
 - ✓ Accuracy
 - ✓ Sensitivity, specificity and the ROC curve
 - ✓ Precision and Recall
- $TP = \text{confusion}[1,1]$ # true positive
- $TN = \text{confusion}[0,0]$ # true negatives
- $FP = \text{confusion}[0,1]$ # false positives
- $FN = \text{confusion}[1,0]$ # false negatives
- Sensitivity = $TP / \text{float}(TP + FN)$
- Specificity = $TN / \text{float}(TN + FP)$
- Precision = $TP / TP + FP$
- Recall = $TP / TP + FN$
- **ROC Curve** -
 - ✓ It shows the tradeoff between sensitivity and specificity (any increase in sensitivity will be accompanied by a decrease in specificity).
 - ✓ The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test.
 - ✓ The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test.
- Optimal cutoff probability is that prob where we get balanced sensitivity and specificity

ROC Curve- Find Optimal Probabilities



```
cutoff_df.plot.line(x='prob', y=['accuracy', 'sensi', 'speci'])  
plt.show()
```

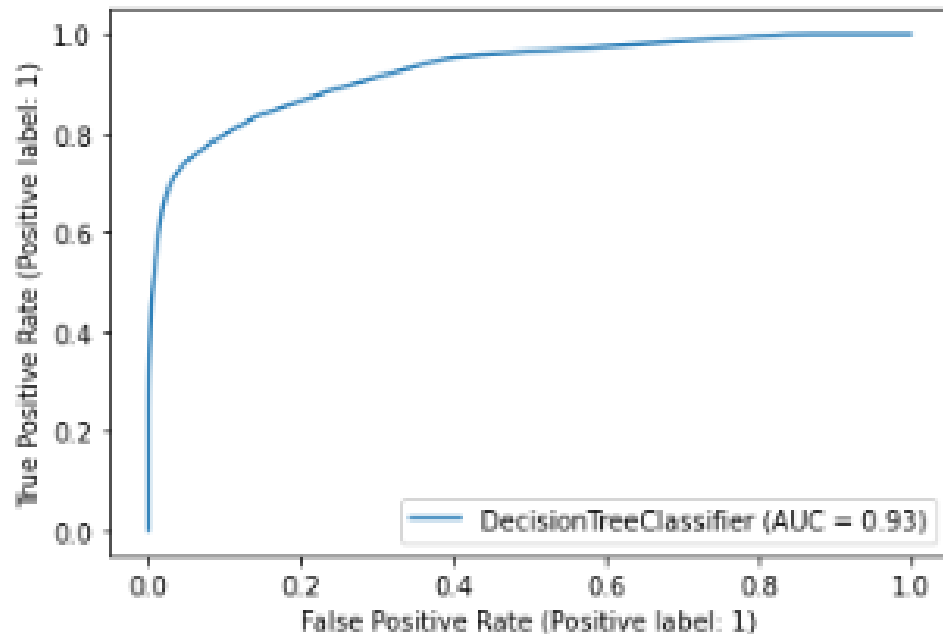


Optimal value is 0.1

- Optimal cutoff for our problem statement was 0.1
- Now the same cutoff is used and the model is run against the train data set to find the optimal results
- On running the model on train data set , with 0.1 as probability cut off , we observed a accuracy of 79 % and Recall around 78% on the test dataset
- We can clearly see most of the critical features are form the action phase, which is inline with the bussiness understanding that action phase needs more attention
- Probably getting the class imbalance sorted could have given better model

Model Building -Decision Tree

- Importing the necessary python libraries to perform decision tree
- Hyperparameter tuning was done
- ROC curve was had AUC =0.93 and the classification metrix was published too

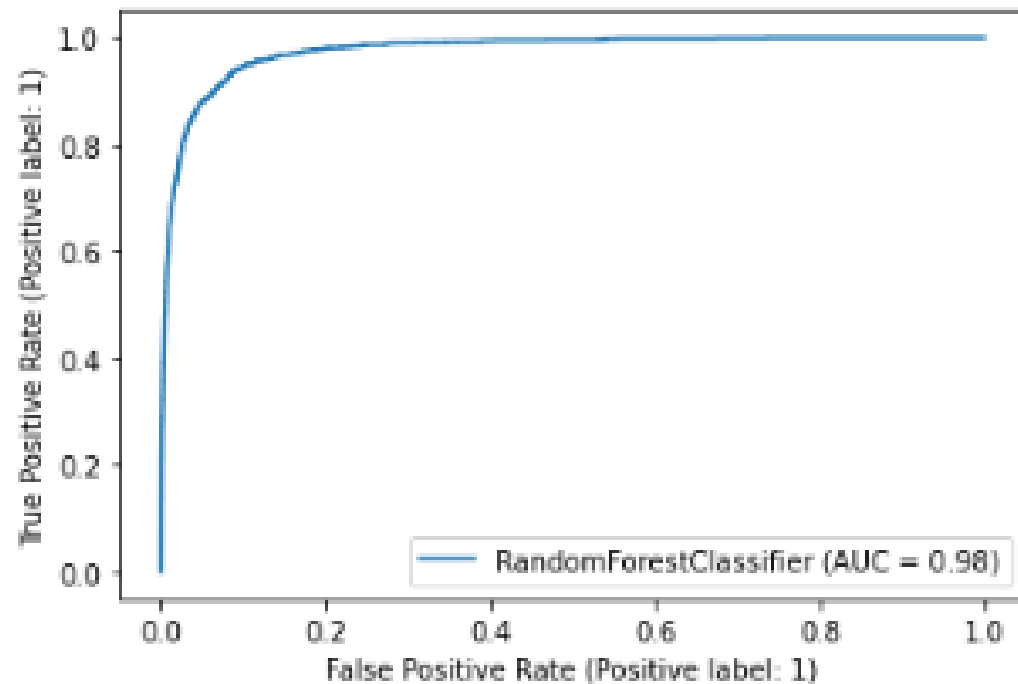


	precision	recall	f1-score	support
0	0.94	0.99	0.96	8121
1	0.52	0.18	0.27	612
accuracy			0.93	8733
macro avg	0.73	0.58	0.62	8733
weighted avg	0.91	0.93	0.92	8733

We are getting an accuracy of 93% on test data, with decision tree

Model Building -Random Forest

- Importing the necessary python libraries to perform random forest algorithms (RandomForestClassifier from ens
- Hyperparameter tuning was done
- ROC curve was had AUC =0.98 and the classification metrix was published too



```
: y_train_pred = rf_best.predict(X_train)
y_test_pred = rf_best.predict(X_test)

# Print the report
print(metrics.classification_report(y_test, y_test_pred))
```

	precision	recall	f1-score	support
0	0.94	0.99	0.97	8121
1	0.60	0.17	0.26	612
accuracy			0.93	8733
macro avg	0.77	0.58	0.61	8733
weighted avg	0.92	0.93	0.92	8733

We are getting an accuracy of 93% on test data, with decision tree

Key Take Aways

- Given our business problem, to retain their customers, we need higher recall.
- As giving an offer to a user not going to churn will cost less as compared to losing a customer and bring a new customer,
- we need to have a high rate of correctly identifying the true positives, hence recall.
- When we compare the models trained we can see the tuned random forest and AdaBoost are performing the best,
- which is highest accuracy along with highest recall
- So, we will go with random forest instead of AdaBoost as that is comparatively a simpler model.

Recommendations -

- ✓ Users whose maximum recharge amount is less than 250 even in the good phase, should have a tag and be re-evaluated time to time as they are more likely to churn
- ✓ Users that have been with the network less than 4 years, should be monitored time to time, as from data we can see that users who have been associated with the network for less than 4 years tend to churn more
- ✓ MOU is one of the major factors, but data especially VBC if the user is not using a data pack is another factor to look out
- ✓ Telecom company needs to pay attention to the roaming rates. They need to provide good offers to the customers who are using services from a roaming zone.
- ✓ 2. The company needs to focus on the STD and ISD rates. Perhaps, the rates are too high. Provide them with some kind of STD and ISD packages.
- ✓ 3. To look into both of the issues stated above, it is desired that the telecom company collect customer query and complaint data and work on their services according to the needs of customers.