TELECOM Churn Case Study

Presentation by

Pranavu G
Abhirame Pillai
Vanita Ishwarlal Ratnani

Case Study – Breakdown

- We have to Predict the Churn Rate in a Telecom Industry.
- We encounter 15-25 % Annual Churn Rate.
- Prepaid Customers Churn Prediction.
- Data Classification into Phases and Encodings.
- Understanding Customer Behavior
- Usage - Based Churn.
- High Value Customers.
- Build Models.
- Select and Evaluate The Models.
- Understand Through Visualization of the Data.
- Recommend Strategies To Manage Customer Churn.
- Recommend Strategies for Customer Retention.

Steps involved in solving the Business Problem

- Step 1. Importing Required Libraries
- Step 2. Importing Data and Data Reading
- Step 3. Data Cleaning and Data Operations
- Step 4. Treating Missing Values and Null Values
- Step 5. Exploratory Data Analysis and Data Visualization
- Step 6. Statistical Operations and Handling Class Imbalance
- Step 7. Numerical and Categorical Data Analysis
- Step 8. Understanding Data and Data Patterns
- Step 9. Separating Train and Test Data
- Step 10. Model Building and Model Selection
- Step 11. Model Evaluation
- Step 12. Inference and Insights Generation
- Step 13. Business Recommendation

Step 1. Importing Required Libraries

Importing the Python Libraries:

In [2]:

```
# Data Analysis:
import numpy as np
import pandas as pd
from collections import Counter
from math import sqrt

# Visualization
import matplotlib.pyplot as plt
import seaborn as sns

# Stats libraries:
from scipy import stats
from scipy.stats import skew, norm
from scipy.special import boxcox1p
from scipy.stats import boxcox_normmax

# Machine Learning Libraries
import statsmodels.api as sm

#Sci-kit learn libraries
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split,GridSearchCV,KFold,cross_val_score
from sklearn import metrics
from sklearn.feature_selection import RFE
from sklearn.metrics import recall_score, accuracy_score, confusion_matrix, f1_score,classification_report
from sklearn.metrics import precision_score, auc, roc_auc_score, roc_curve, precision_recall_curve,plot_roc_curve
from sklearn.linear_model import LogisticRegression
from sklearn.decomposition import PCA, IncrementalPCA
from sklearn.ensemble import RandomForestClassifier
```

Step 2. Importing Data and Data Reading

Reading & Understanding the data

In [4]:
```
# Importing the datasets:

Telecom_churn = pd.read_csv("telecom_churn_data.csv")

Telecom_churn.head()
```

Out[4]:

	mobile_number	circle_id	loc_og_t2o_mou	std_og_t2o_mou	loc_ic_t2o_mou	last_date_of_month_6	last_date_of_month_7	last_date_of_month_8	last_date_of
0	7000842753	109	0.0	0.0	0.0	6/30/2014	7/31/2014	8/31/2014	
1	7001865778	109	0.0	0.0	0.0	6/30/2014	7/31/2014	8/31/2014	
2	7001625959	109	0.0	0.0	0.0	6/30/2014	7/31/2014	8/31/2014	
3	7001204172	109	0.0	0.0	0.0	6/30/2014	7/31/2014	8/31/2014	
4	7000142493	109	0.0	0.0	0.0	6/30/2014	7/31/2014	8/31/2014	

In [5]:
```
# checking the final 5 rows:
Telecom_churn.tail()
```

Out[5]:

	mobile_number	circle_id	loc_og_t2o_mou	std_og_t2o_mou	loc_ic_t2o_mou	last_date_of_month_6	last_date_of_month_7	last_date_of_month_8	last_da
99994	7001548952	109	0.0	0.0	0.0	6/30/2014	7/31/2014	8/31/2014	
99995	7000607688	109	0.0	0.0	0.0	6/30/2014	7/31/2014	8/31/2014	
99996	7000087541	109	0.0	0.0	0.0	6/30/2014	7/31/2014	8/31/2014	
99997	7000498689	109	0.0	0.0	0.0	6/30/2014	7/31/2014	8/31/2014	
99998	7001905007	109	0.0	0.0	0.0	6/30/2014	7/31/2014	8/31/2014	

Step 3. Data Cleaning

Note:

Dropping the data for the 9th month as they will not be used in prediction purpose as these values are not available for the model. They are only considered to calculate whether customer has churned or not.

```
# Dropping the 9 month data. Let's check how many column data we are about to drop
print("Number of columns to be dropped : ", len(cols_with_9))
```

```
Number of columns to be dropped :  56
```

```
Tel_high_val = Tel_high_val.drop(cols_with_9, axis=1)

print("Number of Columns remaining:", len(Tel_high_val.columns))
```

```
Number of Columns remaining: 175
```

Step 4. Treating Missing Values and Null Values

```
# Checking the null values
Telecom_null = null_calc(Telecom_churn)
Telecom_null
```

	Column Name	Null Values	Null Values Percentage
0	count_rech_2g_6	74846	74.85
1	max_rech_data_6	74846	74.85
2	arpu_3g_6	74846	74.85
3	av_rech_amt_data_6	74846	74.85
4	count_rech_3g_6	74846	74.85
5	night_pck_user_6	74846	74.85
6	arpu_2g_6	74846	74.85
7	fb_user_6	74846	74.85
8	total_rech_data_6	74846	74.85
9	date_of_last_rech_data_6	74846	74.85
10	count_rech_3g_7	74428	74.43
11	arpu_3g_7	74428	74.43

columns to consider:

date_of_last_rech_data_6 - 74.85 % missing values

date_of_last_rech_data_7 - 74.43 % missing values

av_rech_amt_data_6 - 74.85 % missing values

av_rech_amt_data_7 - 74.43 % missing values

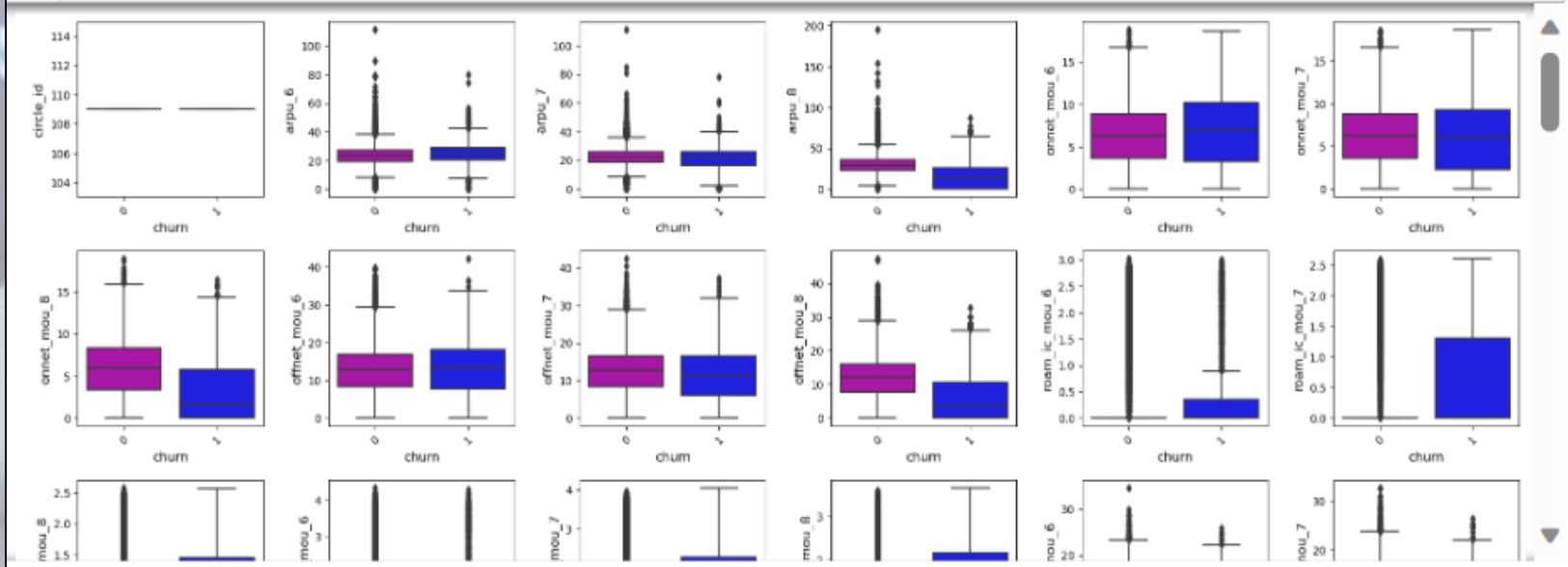
total_rech_data_6 - 74.85 % missing values

total_rech_data_7 - 74.43 % missing values

Step 5. Exploratory Data Analysis and Data Visualization

Boxplot visualization after Box-cox transformation

```
# Create box plots for all numeric features

fig, axes = plt.subplots(round(len(num_cols) / 6), 6, figsize=(20, 40))

for i, ax in enumerate(fig.axes):
    if i < len(num_cols):
        ax.set_xticklabels(ax.xaxis.get_majorticklabels(), rotation=45)
        sns.boxplot(y=num_cols[i], data=Tel_high_val, x='churn', ax=ax,palette = ['m','b'])
        ax.set_ylabel(num_cols[i],fontsize=12)
        ax.set_xlabel('churn',fontsize=12)

fig.tight_layout()
plt.show()
```



Step 6. Statistical Operations and Handling Class Imbalance

Insights:

Maximum recharge value found out to be 1555 and minimum recharge value is 1. We can consider that null value means the customer has not recharged that month. We will impute 0 for the null values for these three columns.

In [45]:
```
for col in ['max_rech_data_6', 'max_rech_data_7', 'max_rech_data_8']:
    Tel_high_val[col].fillna(0, inplace=True)
```

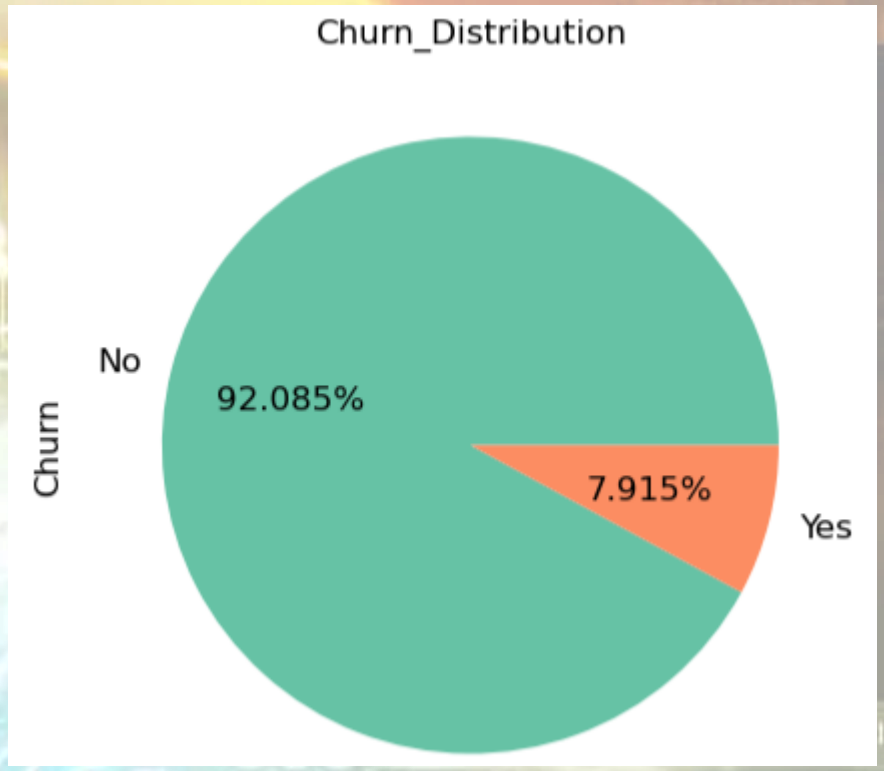
In [46]:
```
# check the remaining null values for rest of the columns
Telecom_null = null_calc(Tel_high_val)

Telecom_null
```

Out[46]:

	Column Name	Null Values	Null Values Percentage
0	loc_og_t2m_mou_8	1088	3.68
1	std_ic_t2m_mou_8	1088	3.68
2	std_ic_t2t_mou_8	1088	3.68
3	std_og_mou_8	1088	3.68
4	loc_ic_mou_8	1088	3.68
5	loc_og_t2f_mou_8	1088	3.68
6	loc_ic_t2f_mou_8	1088	3.68
7	loc_og_t2c_mou_8	1088	3.68
8	loc_ic_t2m_mou_8	1088	3.68
9	loc_og_mou_8	1088	3.68
10	loc_ic_t2t_mou_8	1088	3.68
11	std_og_t2t_mou_8	1088	3.68

In [47]:
```
# Dropping the date columns in which imputation is not possible:

Tel_high_val.drop(['date_of_last_rech_6','date_of_last_rech_7','date_of_last_rech_8',
                   'last_date_of_month_6','last_date_of_month_7','last_date_of_month_8'], axis = 1, inplace = True)
Tel_high_val.shape
```

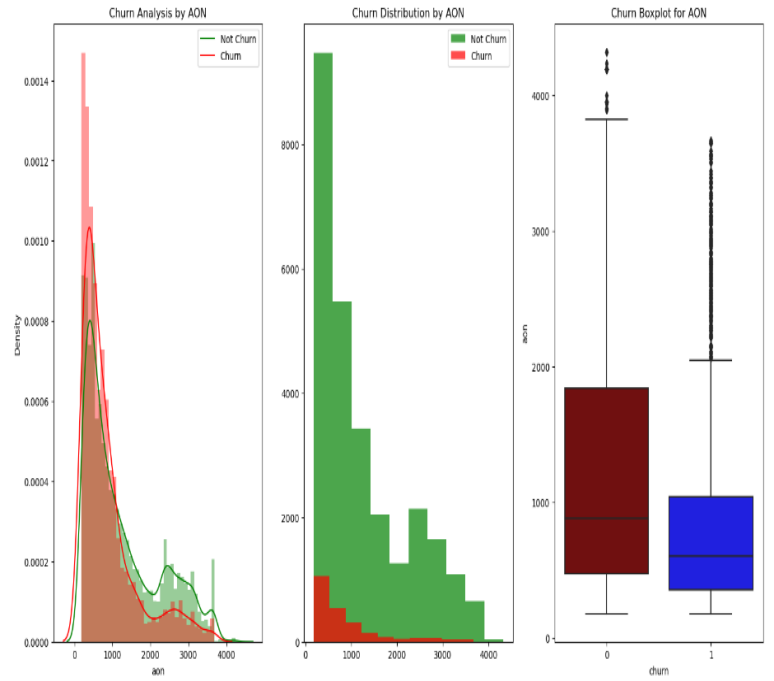
Out[47]: (29591, 166)

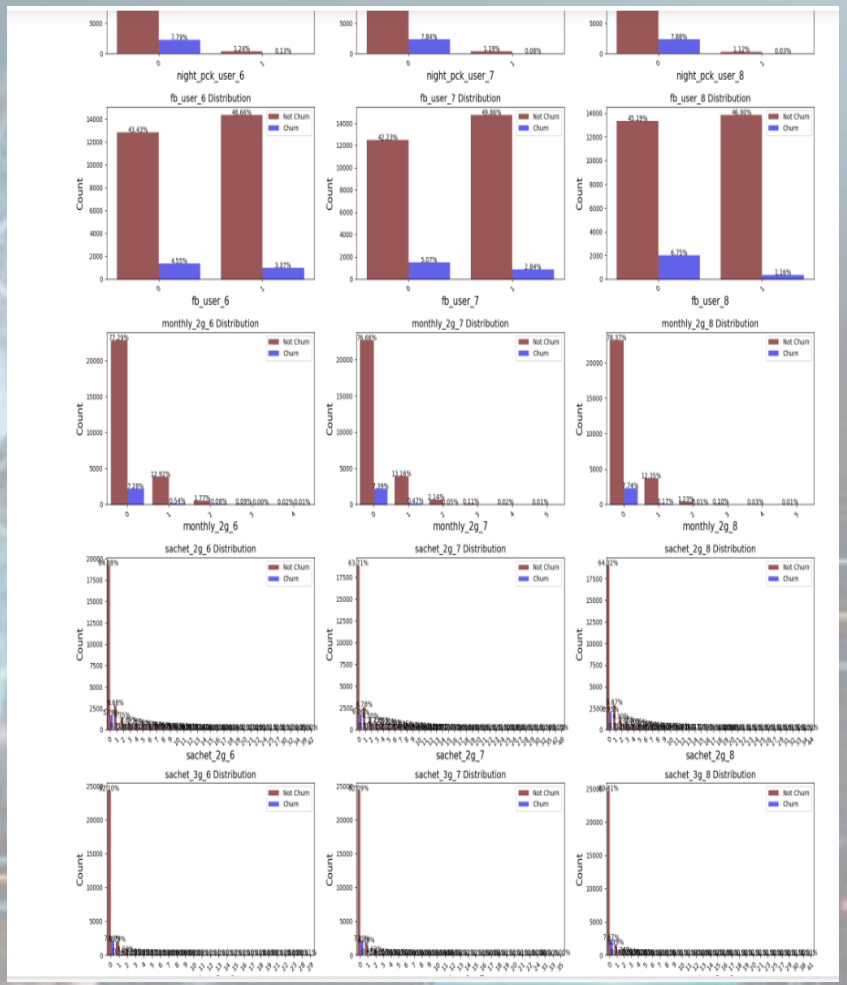


Dataset is heavily Imbalanced we used PCA and sampling techniques to handle class-Imbalance

Step 7. Numerical and Categorical Data Analysis

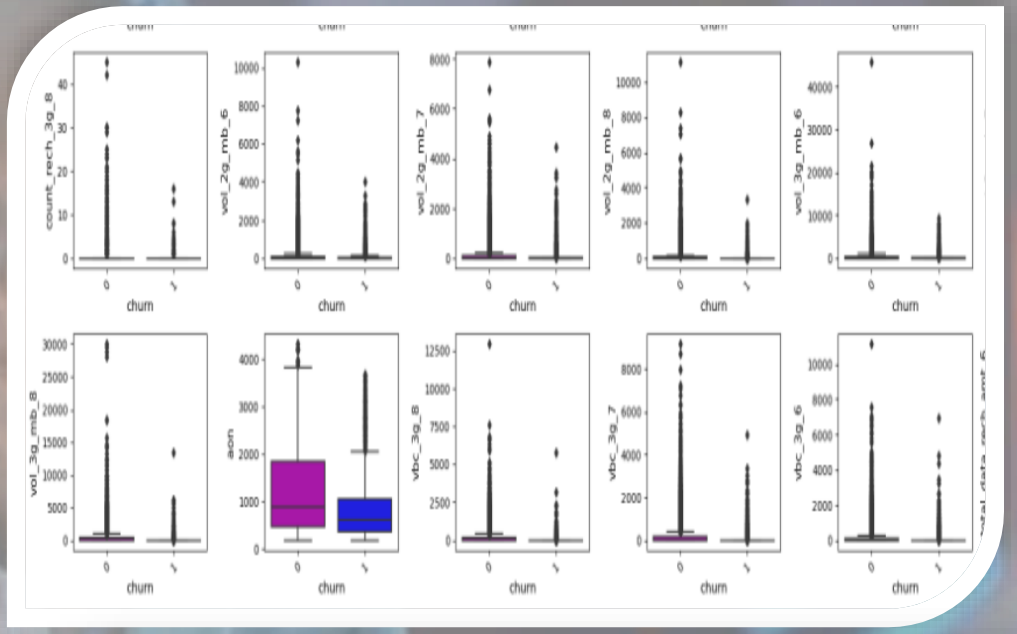
```
plt.subplot(1, 3, 2)
plt.hist(Tel_high_val.loc[Tel_high_val['churn'] == 0, 'aon'], color='g', alpha=0.7)
plt.hist(Tel_high_val.loc[Tel_high_val['churn'] == 1, 'aon'], color='r', alpha=0.7)
plt.legend(['Not Churn', 'Churn'])
plt.title('Churn Distribution by AON')

# Plot 3
plt.subplot(1, 3, 3)
sns.boxplot(y='aon', data=Tel_high_val, x='churn', palette=['maroon', 'blue'])
plt.title('Churn Boxplot for AON')

plt.show()
```

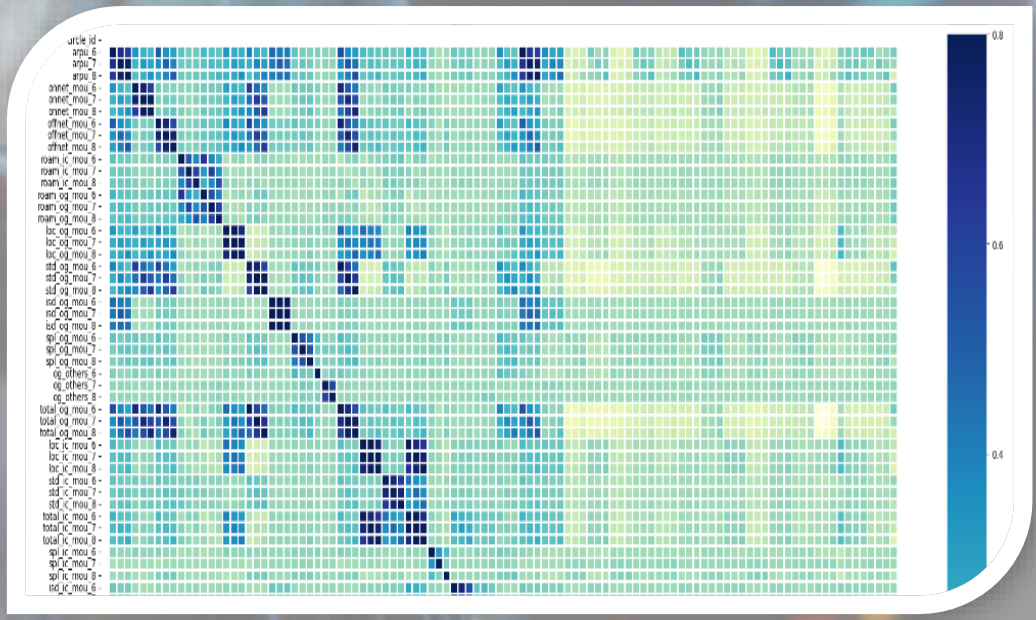




Step 8. Outliers Handling and Correlation Analysis

- Analyzed the data patterns using different plots:
- Box-plot, Count-plot and Heat-maps (for correlation analysis among different features





Step 9: Train and Test Split

Train- Test Split

In [75]:

```
#Target variable

X = Tel_high_val.drop('churn', axis = 1)
y = Tel_high_val[['churn']]

# Splitting the data into train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 100)

#Checkiong the shape of the created Train & Test DFs
print(" Shape of X_train is : ",X_train.shape)
print(" Shape of y_train is : ",y_train.shape)
print(" Shape of X_test is  : ",X_test.shape)
print(" Shape of y_test is  : ",y_test.shape)
```

```
Shape of X_train is :  (20713, 72)
Shape of y_train is :  (20713, 1)
Shape of X_test is  :  (8878, 72)
Shape of y_test is  :  (8878, 1)
```

Step 10. Model Building

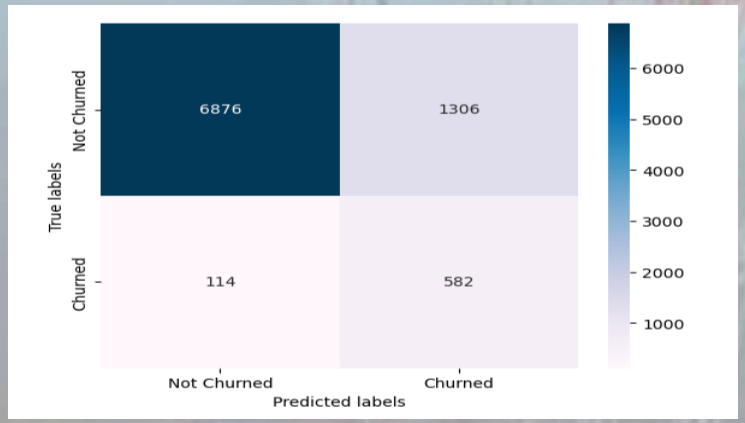
- We have used Logistic Regression model with sampling techniques and PCA and also random forest models.
- Finally, we have chosen the best model for this case study and based on certain Evaluation metrics

```
# Step 2: Build the logistic model using RFE selected columns with StatsModels

logistic_model_imbalanced, X_train_sm_imbalanced = build_logistic_model(X_train_rfe_imbalanced, y_train)
logistic_model_imbalanced.summary()
```

Generalized Linear Model Regression Results

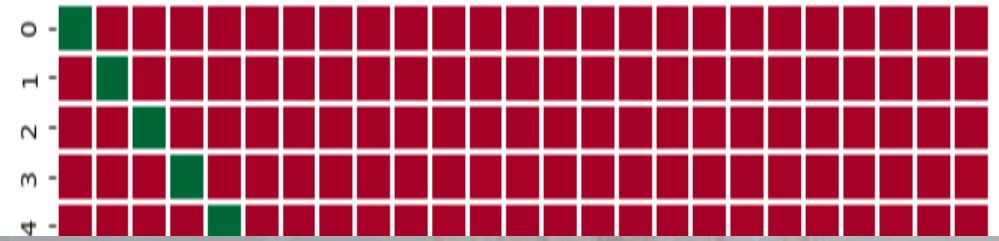
Dep. Variable:	churn	No. Observations:	20713
Model:	GLM	Df Residuals:	20687
Model Family:	Binomial	Df Model:	25
Link Function:	Logit	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-3468.4
Date:	Sun, 03 Dec 2023	Deviance:	6936.8
Time:	07:07:25	Pearson chi2:	3.23e+04
No. Iterations:	8	Pseudo R-squ. (CS):	0.1975
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-3.8240	0.058	-65.512	0.000	-3.938	-3.710
night_pck_user_8	0.8465	0.392	2.162	0.031	0.079	1.614
arpu_good_phase	0.1839	0.126	1.462	0.144	-0.063	0.431

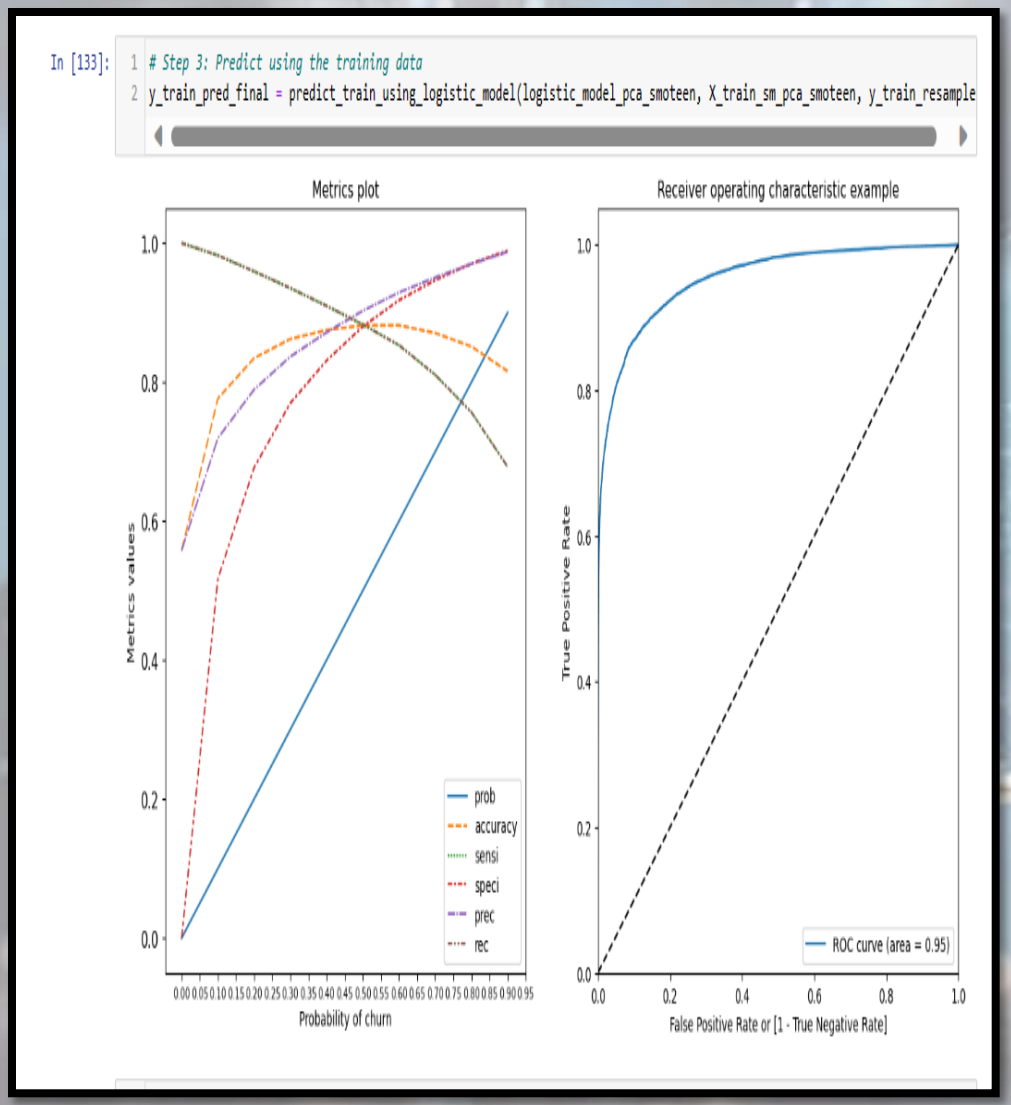



```
# Step 1.2 : Create 25 Principal components
X_train_pca_smoteen, X_test_pca_smoteen = perform_incremental_PCA(X_train_resampled, y_
```

```
Shape of X train PCA :  (34146, 25)
Shape of Y train PCA :  (34146, 1)
Shape of X test PCA :  (8878, 25)
Shape of Y test PCA :  (8878, 1)
```



Step 11. Model Evaluation



- We have used Evaluation metrics like:

- Precision
- Recall
- F1-Score
- AUC-ROC Curve

- Considered the Trade-off b/w different metrics and Interpretability and use case which we are dealing with it to achieve best possible results

Step 10. Model Building and Model Selection

- One of the best Performing Model: Random Forest with SMOTEENN & Hyper-parameter tuning (Model 7)
- Balances interpretability and prediction performance.
- Achieves a ***high recall of 0.8218***, critical for identifying positive cases in churn prediction.
- Offers a competitive ***AUC score of 0.8424***, indicating good discrimination power.
- ***Precision (0.3379) and F1 score (0.4789)*** strike a reasonable balance.
- Considering the trade-off between interpretability and performance, this model stands out for its overall effectiveness in capturing churn instances while maintaining reasonable interpretability.
- **We can also choose other models like:**
- Logistic Regression with PCA and SMOTE
- Logistic Regression with PCA & SMOTEENN (highest recall value - best prediction)
- In case of logisitic regression, make sure to handle multi-collinearity among the features

- Note: Achieving the balance in terms results (i.e trade-off b/w recall, discrimination power and interpretability needs to considered) Rather than focusing only on recall as a sole-determining factor of evaluation metric.

	Model	Accuracy	Precision	Recall	F1 score	Area under ROC curve
0	Logistic Regression with PCA & SMOTEENN	0.7799	0.2470	0.8822	0.3859	0.8267
1	Logistic Regression with PCA, SMOTEEN, L2 regu...	0.8029	0.2674	0.8707	0.4092	0.8339
2	Logistic Regression with PCA & Random over sam...	0.8319	0.2974	0.8391	0.4391	0.8352
3	Logistic Regression with SMOTE & without PCA	0.8401	0.3083	0.8362	0.4505	0.8383
4	Logistic Regression with PCA & Random under sa...	0.8312	0.2957	0.8348	0.4367	0.8328
5	Logistic Regression with PCA & SMOTE	0.8361	0.3020	0.8319	0.4432	0.8342
6	Random Forest with SMOTEENN & Hyperparameter t...	0.8598	0.3379	0.8218	0.4789	0.8424
7	Random Forest with SMOTEENN	0.9097	0.4535	0.7428	0.5632	0.8333
8	Logistic Regression on imbalanced data without...	0.9401	0.6565	0.4943	0.5639	0.7361
9	Random Forest with class_weight	0.9459	0.7798	0.4325	0.5564	0.7110

Step 12. Feature selection and Insights

- High loc_ic_mou_action_phase indicates significant local incoming call activity, influencing retention.
- Total_ic_mou_action_phase, encompassing both local and STD incoming call minutes, strongly impacts customer retention.
- loc_og_mou_action_phase, representing local outgoing call activity, is a pivotal factor in customer decisions.
- Roam_og_mou_action_phase highlights the role of roaming outgoing minutes in understanding churn risk.
- Total_rech_amt_action_phase reveals that higher recharge investments correlate with increased customer retention.
- Total_og_mou_action_phase, combining local and STD outgoing call minutes, significantly influences retention.
- Roam_ic_mou_action_phase emphasizes the importance of incoming call activity during roaming for churn prediction.
- Arpu_action_phase, reflecting Average Revenue per User, contributes to understanding customer value and potential retention.
- Max_rech_amt_action_phase shows that customers with higher maximum recharges are more likely to be retained.
- Fb_user_8, indicating Facebook usage in the eighth month, contributes to churn prediction.

	Value	Feature
31	0.111361	loc_ic_mou_action_phase
41	0.109216	total_ic_mou_action_phase
19	0.072808	loc_og_mou_action_phase
17	0.068389	roam_og_mou_action_phase
45	0.065768	total_rech_amt_action_phase
29	0.063555	total_og_mou_action_phase
15	0.060821	roam_ic_mou_action_phase
9	0.039440	arpu_action_phase
47	0.035405	max_rech_amt_action_phase
6	0.035241	fb_user_8

Step 13. Business Recommendation

- Call usage, Recharge behavior, and roaming activity during the action phase.
- These insights suggest that customer communication patterns and recharge behaviors during the action phase heavily influence churn decisions.
- Emphasizing targeted promotions, improving network quality, and addressing customer needs during this critical phase can contribute to better customer retention.
- In summary, by strategically implementing targeted promotions, improving network quality, and addressing customer needs during the action phase, businesses can create a more customer-centric approach that positively impacts retention rates and overall customer satisfaction.


THANK YOU