TELECOM CHURN PREDICTION

- Churn is the measure of how many customers stop using a product. This can be measured based on actual usage or failure to renew (when the product is sold using a subscription model). Often evaluated for a specific period of time, there can be a monthly, quarterly, or annual churn rate.\

- For Telco companies it is key to attract new customers and at the same time avoid contract terminations (=churn) to grow their revenue generating base. Looking at churn, different reasons trigger customers to terminate their contracts, for example better price offers, more interesting packages, bad service experiences or change of customers' personal situations..

- Step 1: Problem Definition
- Step 2: Data Collection
- Step 3: Exploratory Data Analysis (EDA)
- Step 4: Feature Engineering
- Step 5: Train/Test Split
- Step 6: Model Evaluation Metrics Definition
- Step 7: Model Selection, Training, Prediction and Assessment
- Step 8: Hyperparameter Tuning/Model Improvement

PROBLEM DEFINITION

- 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.

DATA COLLECTION

```
1]: import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns
    import warnings
    warnings.filterwarnings('ignore')
    import matplotlib.pyplot as plt
    import matplotlib.style as style
    import seaborn as sns

    import warnings
    warnings.filterwarnings('ignore')
```

Data preparation

```
2]: a=pd.read_csv("telecom_churn_data.csv")
    a.head()
```

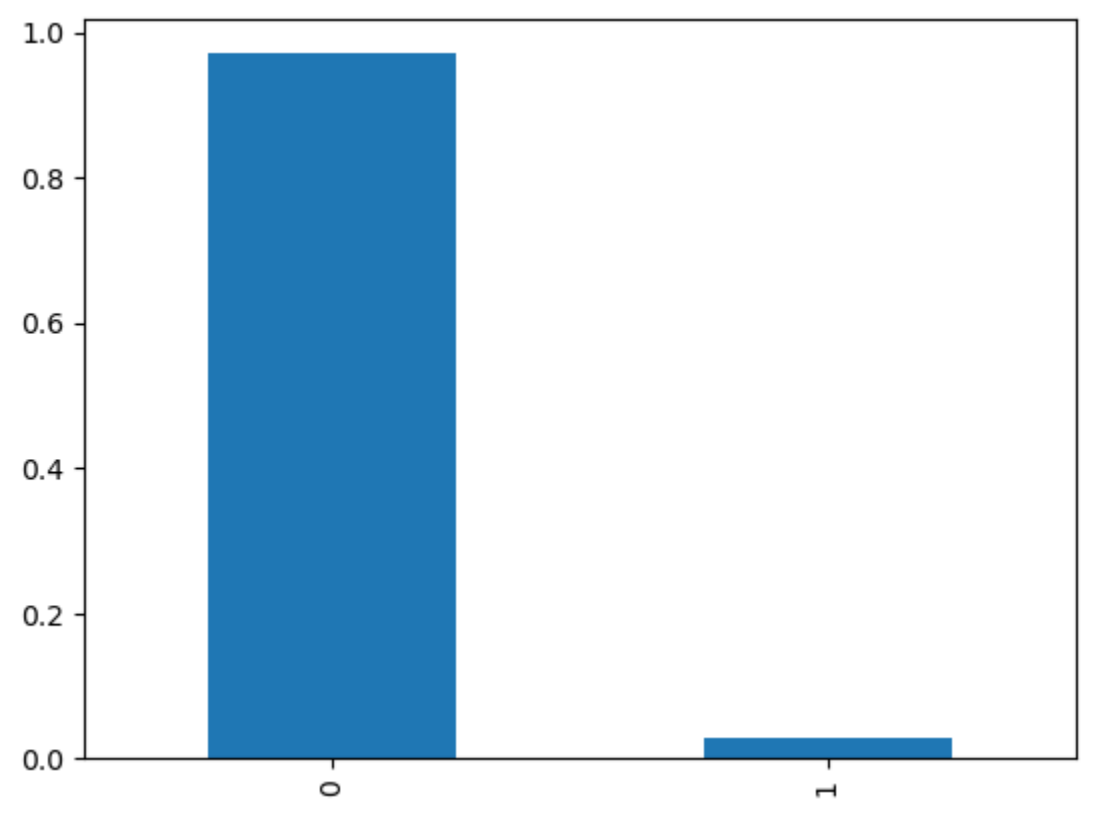
2]:

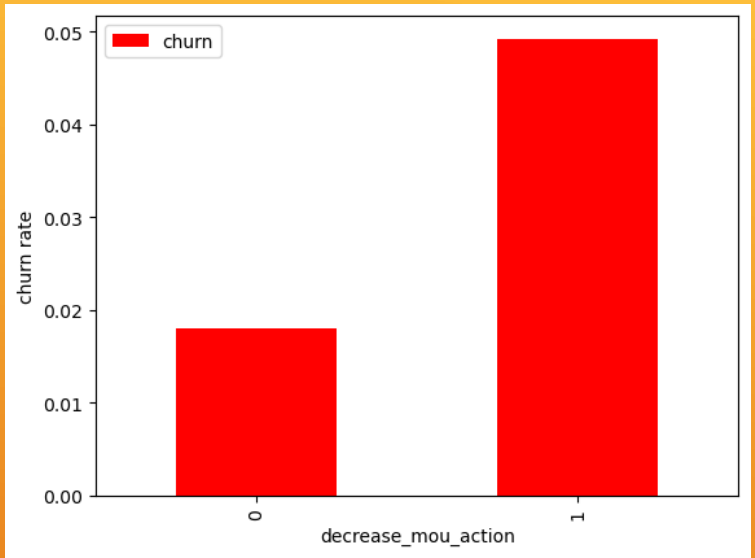
	mobile_number	circle_id	loc_og_t2o_mou	std_og_t2o_mou	loc_ic_t2o_mou	last_date_of_month_6	last_date_of_month_7	last_date_of_month_8	last_date_of
0	7000842753	109	0.0	0.0	0.0	6/30/2014	7/31/2014	8/31/2014	
1	7001865778	109	0.0	0.0	0.0	6/30/2014	7/31/2014	8/31/2014	
2	7001625959	109	0.0	0.0	0.0	6/30/2014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

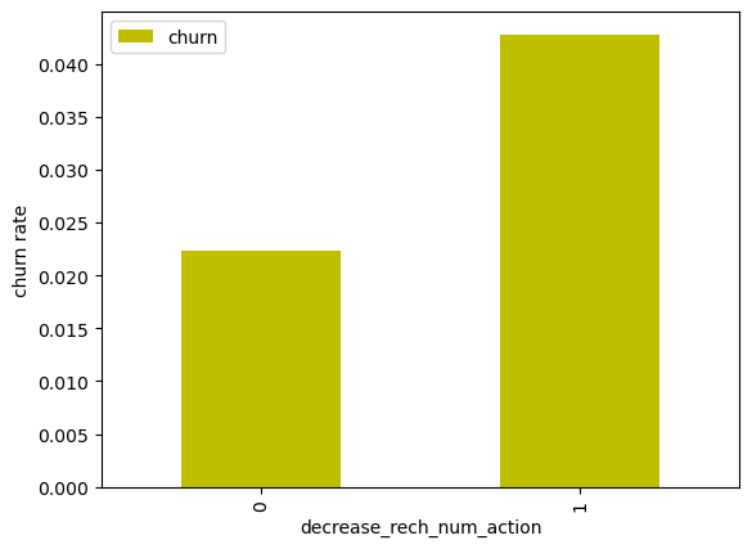
5 rows × 226 columns

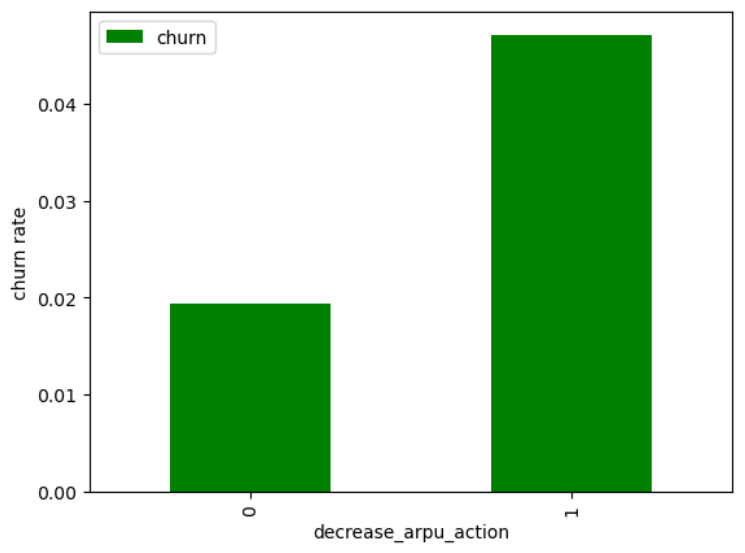
EXPLORATORY DATA ANALYSIS (EDA)

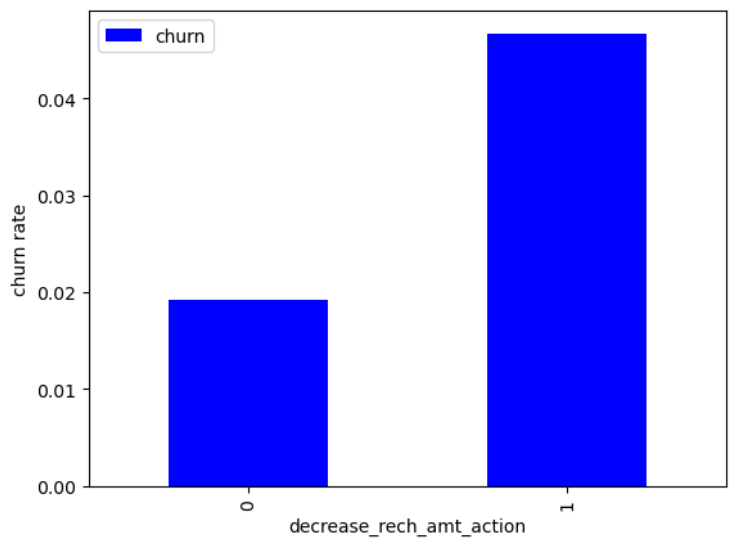
Churn count as 1 and Not Churn as 0

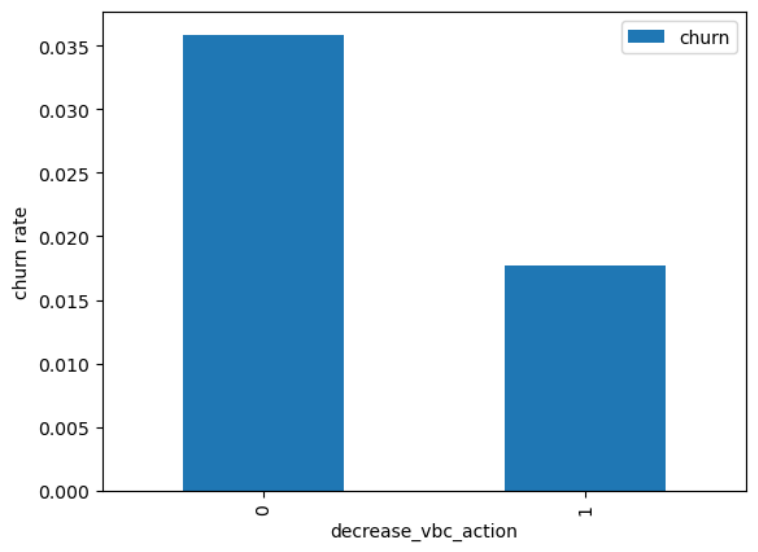


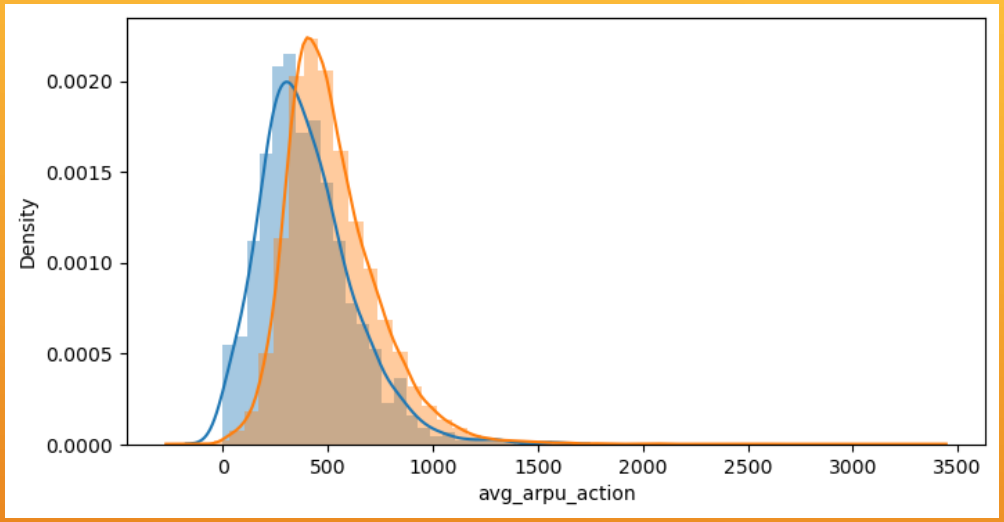

churn
churn rate
0.05
0.04
0.03
0.02
0.01
0.00
0
1
decrease_mou_action

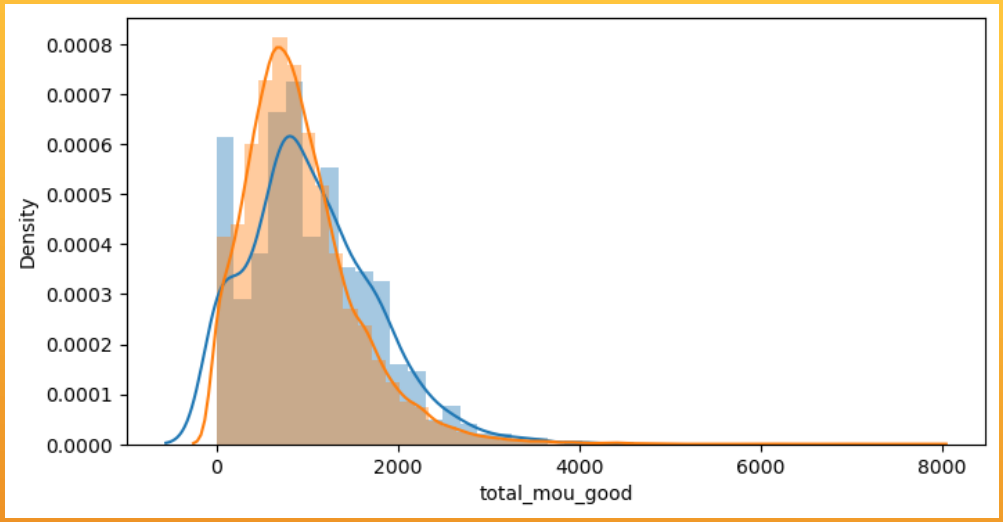

churn
churn rate
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
1
decrease_rech_num_action


churn
churn rate
0.04
0.03
0.02
0.01
0.00
0
1
decrease_arpu_action

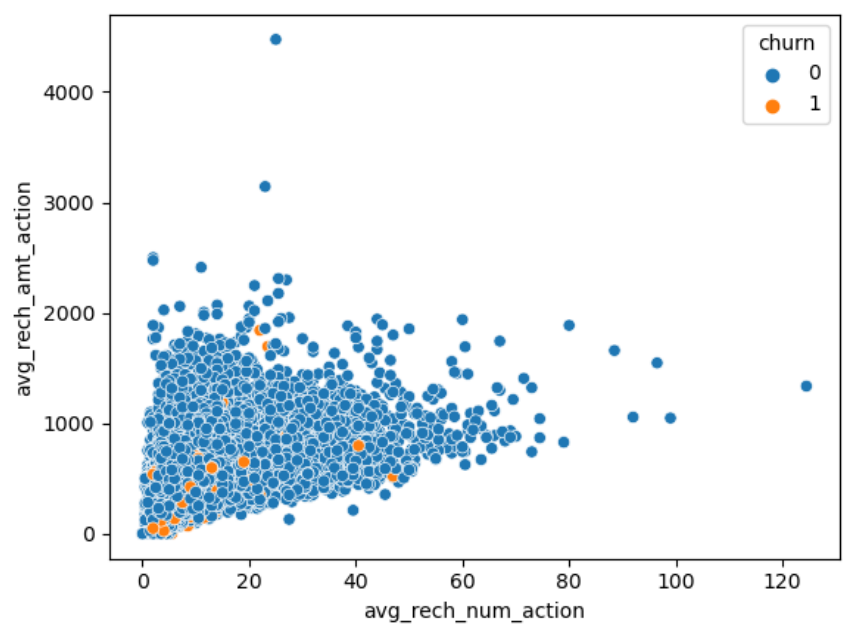

churn
churn rate
0.04
0.03
0.02
0.01
0.00
0
1
decrease_rech_amt_action


churn
churn rate
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
1
decrease_vbc_action

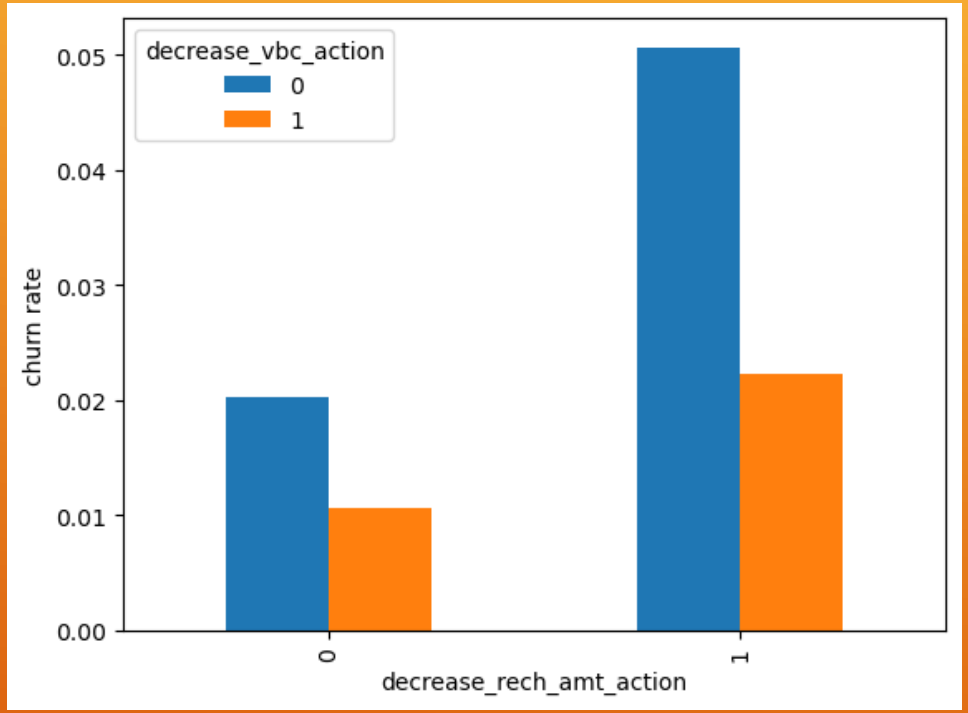




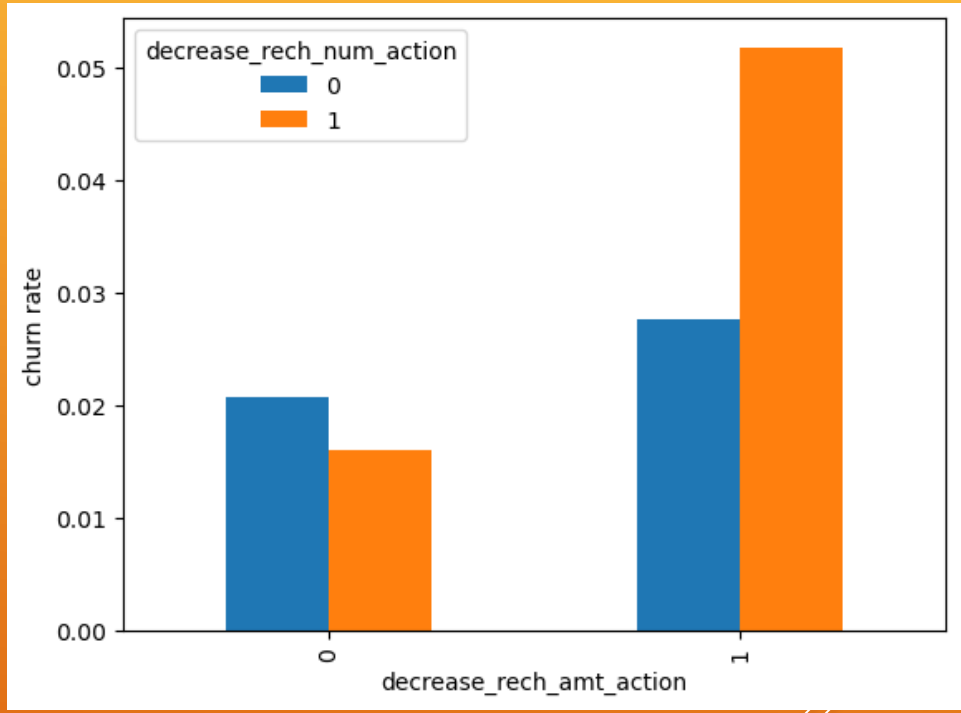
#Recharge amount and number of recharge in action month



#churn rate by the decreasing recharge amount and volume based cost in the action phase



#churn rate by the decreasing recharge amount and number of recharge



TRAIN TEST SPLIT

```
1]: from sklearn.model_selection import train_test_split

2]: y=a.pop("churn")
    x=a

3]: X_train,x_test,y_train,y_test = train_test_split(x, y, train_size=0.7, random_state=100)

4]: print(X_train.shape, x_test.shape)

    (15635, 138) (6702, 138)
```

TRAIN/TEST SPLIT

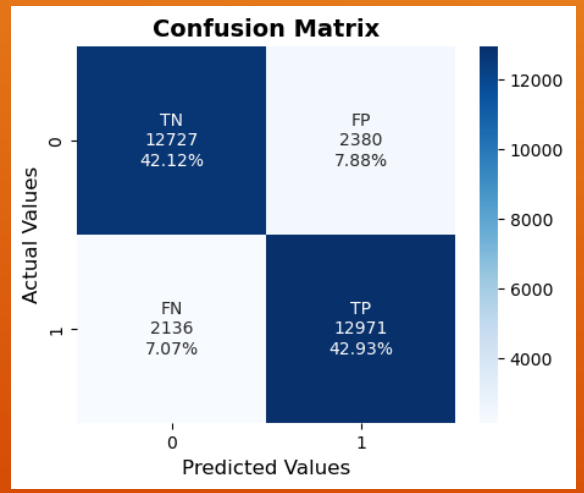
MODEL PREPARATION

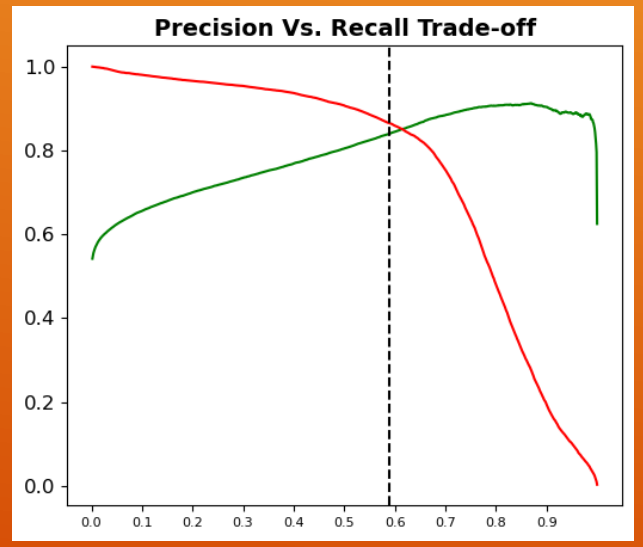
Logistic Regression:

In [93]:
```
# importing the required libraries
import statsmodels.api as sm
from sklearn.linear_model import LogisticRegression

logreg = LogisticRegression()

from sklearn.feature_selection import RFE
rfe = RFE(logreg, n_features_to_select = 15)
rfe = rfe.fit(x_train_sm, y_train_sm)
```

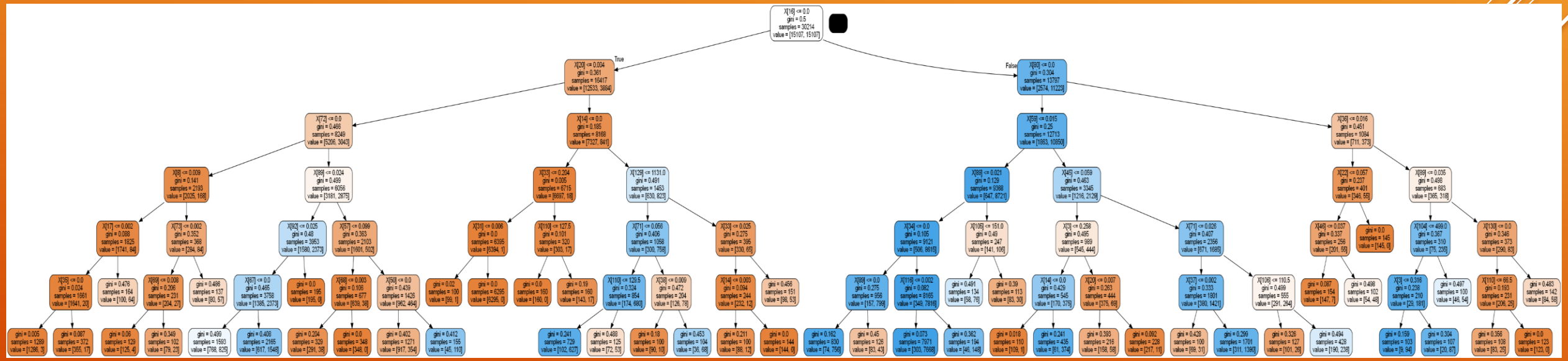




Decision Trees:

In [152]:
```
from sklearn.tree import DecisionTreeClassifier,plot_tree
```

In [153]:
```
decision_model = DecisionTreeClassifier(class_weight='balanced',
                                        max_features='auto',
                                        min_samples_split=100,
                                        min_samples_leaf=100,
                                        max_depth=6,
                                        random_state=100).fit(x_train_sm, y_train_sm)
```



- ALL 3 MODEL HAVE SHOWED A GREAT ACCURACY VALUE.
- Important features on which models were build:

```
Index(['offnet_mou_8', 'roam_og_mou_7', 'roam_og_mou_8', 'loc_og_t2m_mou_8
       'loc_og_t2f_mou_6', 'loc_og_t2f_mou_8', 'loc_og_mou_8', 'std_og_mou
       'loc_ic_t2t_mou_8', 'loc_ic_t2f_mou_8', 'total_ic_mou_8', 'aug_vbc
```