

```
In [22]: import pandas as pd
import numpy as np
from imblearn.combine import SMOTEENN
from sklearn.metrics import recall_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn import tree
import matplotlib.pyplot as plt
```

```
In [2]: df = pd.read_csv("telChurn.csv")
```

```
In [3]: obj_col = df.select_dtypes(include='object').columns
obj_col
```

```
Out[3]: Index(['last_date_of_month_6', 'last_date_of_month_7', 'last_date_of_month_8',
            'date_of_last_rech_6', 'date_of_last_rech_7', 'date_of_last_rech_8',
            'date_of_last_rech_data_6', 'date_of_last_rech_data_7',
            'date_of_last_rech_data_8'],
            dtype='object')
```

```
In [4]: df.drop(obj_col,axis=1,inplace=True)
df.head()
```

```
Out[4]:
```

| | loc Og t2o mou | std Og t2o mou | loc ic t2o mou | arpu_6 | arpu_7 | arpu_8 | onnet_mou |
|---|----------------|----------------|----------------|---------|---------|---------|-----------|
| 0 | 0.0 | 0.0 | 0.0 | 197.385 | 214.816 | 213.803 | 0 |
| 1 | 0.0 | 0.0 | 0.0 | 34.047 | 355.074 | 268.321 | 24 |
| 2 | 0.0 | 0.0 | 0.0 | 261.636 | 309.876 | 238.174 | 50 |
| 3 | 0.0 | 0.0 | 0.0 | 378.721 | 492.223 | 137.362 | 413 |
| 4 | 0.0 | 0.0 | 0.0 | 119.518 | 247.435 | 170.231 | 33 |

5 rows × 165 columns

```
In [5]: X = df.drop('Churn', axis=1)
y = df.Churn
y.value_counts()
```

```
Out[5]: 0    27879
1     2569
Name: Churn, dtype: int64
```

```
In [6]: sm = SMOTEENN()
xrs, yrs = sm.fit_resample(X, y)
```

```
In [7]: from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(xrs,yrs,train_size
```

```
In [8]: df = DecisionTreeClassifier(criterion='gini', random_state=100, ma
```

```
In [9]: df.fit(X_train,y_train)
```

```
Out[9]:
```

```
DecisionTreeClassifier(max_depth=6, min_samples_leaf=8, random_state=100)
```

```
In [10]: y_pred4=df.predict(X_test)
         v_pred4
```

```
Out[10]: array([0, 0, 0, ..., 1, 1, 0])
```

```
In [11]: df_score(X_test, v_test)
```

```
Out[11]: 0.9187365198636332
```

```
In [12]: print(classification_report(v_test, v_pred4, labels=[0, 1]))
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.89 | 0.93 | 0.91 | 6053 |
| 1 | 0.94 | 0.91 | 0.93 | 8320 |
| accuracy | | | 0.92 | 14373 |
| macro avg | 0.92 | 0.92 | 0.92 | 14373 |
| weighted avg | 0.92 | 0.92 | 0.92 | 14373 |

Random_Forest

```
In [15]: rf=RandomForestClassifier(n_estimators=100, criterion='gini', random_state=100)
```

```
In [16]: rf.fit(X_train, v_train)
```

```
Out[16]: RandomForestClassifier(max_depth=6, min_samples_leaf=8, random_state=100)
```

```
In [17]: y_pred5=df.predict(X_test)
         v_pred5
```

```
Out[17]: array([0, 0, 0, ..., 1, 1, 0])
```

```
In [18]: rf_score(X_test, v_test)
```

```
Out[18]: 0.9244416614485493
```

```
In [19]: print(classification_report(v_test, v_pred5, labels=[0, 1]))
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.89 | 0.93 | 0.91 | 6053 |
| 1 | 0.94 | 0.91 | 0.93 | 8320 |
| accuracy | | | 0.92 | 14373 |
| macro avg | 0.92 | 0.92 | 0.92 | 14373 |
| weighted avg | 0.92 | 0.92 | 0.92 | 14373 |

```
In [ ]:
```

