In [1]:
```
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
```

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 [8]:
```
df.drop(['loc_og_t2o_mou', 'std_og_t2o_mou', 'loc_ic_t2o_mou',
                           'std_og_t2c_mou_6', 'std_og_t2c_mou_7',
                          'std_og_t2c_mou_8',  'std_ic_t2o_mou_6',
                          'std_ic_t2o_mou_7', 'std_ic_t2o_mou_8'], 1
        )
```

In [9]:
```
X = df.drop('Churn', axis=1)
y = df.Churn
y.value_counts()
```

Out[9]:
```
0    27879
1     2569
Name: Churn, dtype: int64
```

In [10]:
```
sm = SMOTEENN()
xrs, yrs = sm.fit_resample(X, y)
```

In [11]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(xrs,yrs,train_size
```

In [12]:
```
from sklearn.linear_model import LogisticRegression
log_clf = LogisticRegression(solver='saga', random_state=42,max_ite
```

In [13]:
```
log_clf.fit(X_train,y_train)
```

Out[13]:
```
LogisticRegression(class_weight='balanced', max_iter=100000, rando
m_state=42,
                   solver='saga')
```

In [14]:
```
log_clf.score(X_test,y_test)
```

Out[14]:
```
0.8728448275862069
```

In [15]:
```
y_pred=log_clf.predict(X_test)
y_pred
```

Out[15]:
```
array([1, 0, 0, ..., 1, 1, 0])
```

In [16]:
```
print(classification_report(y_test, y_pred, labels=[0,1]))
```

```
              precision    recall  f1-score   support

           0       0.82      0.90      0.86      6080
           1       0.92      0.86      0.89      8304

    accuracy                           0.87     14384
   macro avg       0.87      0.88      0.87     14384
weighted avg       0.88      0.87      0.87     14384
```

In [ ]:
```

```