Customer Churn Analysis project with Machine learning

Problem Statement: -

Customer churn is when a company's customers stop doing business with that company. Businesses are very keen on measuring churn because keeping an existing customer is far less expensive than acquiring a new customer. New business involves working leads through a sales funnel, using marketing and sales budgets to gain additional customers. Existing customers will often have a higher volume of service consumption and can generate additional customer referrals.

Customer retention can be achieved with good customer service and products. But the most effective way for a company to prevent attrition of customers is to truly know them. The vast volumes of data collected about customers can be used to build churn prediction models. Knowing who is most likely to defect means that a company can priorities focused marketing efforts on that subset of their customer base.

Preventing customer churn is critically important to the telecommunications sector, as the barriers to entry for switching services are so low.

You will examine customer data from IBM Sample Data Sets with the aim of building and comparing several customer churn prediction models.

Problem Definition: -

As we all know that churn prediction is a common use case in machine learning domain and here, we will analysis the churn prediction in telecommunication sector. Churn simply means "leaving the company". In any business or company, it is far less expensive to keep their existing customer to get new customer. We will use different methods to analysis and observe the churn prediction using machine learning model.

Source of dataset: -

The dataset is given by "Datatrained".

Link: - https://github.com/dsrscientist/DSData/blob/master/Telecom_customer_churn.csv

Customer Churn Analysis project with Machine learning

Data Analysis: -

First, we need to import the important libraries and required dataset from the above given link.

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
print('Libraries imported')
```
```
Libraries imported
```

In [2]:
```
#storing the file path/ url path in a variable
url = "https://raw.githubusercontent.com/dsrscientist/DSData/master/Telecom_customer_churn.csv"

#make dataframe of the data
df0 = pd.read_csv(url)
print('Dataset imported')
```
```
Dataset imported
```

In [35]:
```
print(df0.shape)
print(' ')
print(df0.dtypes)
```
```
(7043, 21)

customerID           object
gender               object
SeniorCitizen         int64
Partner              object
Dependents           object
tenure                int64
PhoneService         object
MultipleLines        object
InternetService      object
OnlineSecurity       object
OnlineBackup         object
DeviceProtection     object
TechSupport          object
StreamingTV          object
StreamingMovies      object
Contract             object
PaperlessBilling     object
PaymentMethod        object
MonthlyCharges      float64
TotalCharges        float64
Churn                object
dtype: object
```

The given dataset contains 21 columns and 7043 rows. In which 20 features are independent variables and 1 is our target (dependent variable). Target variable indicates if a customer has left the company (i.e., churn=yes) or not (i.e., churn==no).

There are different types of variables in the given dataset. customerID, gender, Partner, Dependents, PhoneService, MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, Contract, PaperlessBilling, PaymentMethod, TotalCharges and Churn contains categorical datatypes and remaining i.e., SeniorCitizen, tenure, MonthlyCharges are of continuous datatype.

Customer Churn Analysis project with Machine learning

At first glance, only customerID seems irrelevant to customer churn. Other variables may or may not have an effect on customer churn. We will figure out.

```
In [4]:  df0.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7043 non-null   object
 1   gender            7043 non-null   object
 2   SeniorCitizen     7043 non-null   int64
 3   Partner           7043 non-null   object
 4   Dependents        7043 non-null   object
 5   tenure            7043 non-null   int64
 6   PhoneService      7043 non-null   object
 7   MultipleLines     7043 non-null   object
 8   InternetService   7043 non-null   object
 9   OnlineSecurity    7043 non-null   object
 10  OnlineBackup      7043 non-null   object
 11  DeviceProtection  7043 non-null   object
 12  TechSupport       7043 non-null   object
 13  StreamingTV       7043 non-null   object
 14  StreamingMovies   7043 non-null   object
 15  Contract          7043 non-null   object
 16  PaperlessBilling  7043 non-null   object
 17  PaymentMethod     7043 non-null   object
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object
 20  Churn             7043 non-null   object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

Let's start with checking the missing values in our given dataset and there is no missing value.

```
n [7]:  df0.isna().sum()

Out[7]: customerID          0
        gender              0
        SeniorCitizen       0
        Partner             0
        Dependents          0
        tenure              0
        PhoneService        0
        MultipleLines       0
        InternetService     0
        OnlineSecurity      0
        OnlineBackup        0
        DeviceProtection    0
        TechSupport         0
        StreamingTV         0
        StreamingMovies     0
        Contract            0
        PaperlessBilling    0
        PaymentMethod       0
        MonthlyCharges      0
        TotalCharges        0
        Churn               0
        dtype: int64
```

Customer Churn Analysis project with Machine learning

Now, let's check for the duplicate values

```
In [6]:  df0.duplicated().sum()
Out[6]:  0
```

There is no duplicate value.

Checking the variables that contain zero

```
In [8]:  df0[df0 ==0].count()
Out[8]:  customerID             0
         gender                 0
         SeniorCitizen       5901
         Partner                0
         Dependents             0
         tenure                11
         PhoneService           0
         MultipleLines          0
         InternetService        0
         OnlineSecurity         0
         OnlineBackup           0
         DeviceProtection       0
         TechSupport            0
         StreamingTV            0
         StreamingMovies        0
         Contract               0
         PaperlessBilling       0
         PaymentMethod          0
         MonthlyCharges         0
         TotalCharges           0
         Churn                  0
         dtype: int64
```

Target variable i.e., "Churn" has two value "yes" and "no". in the given dataset the count of yes and no are as below

```
In [8]:  df0.Churn.value_counts()
Out[8]:  No     5174
         Yes    1869
         Name: Churn, dtype: int64
```

Yes has 1869 counts and No has 5174

Customer Churn Analysis project with Machine learning

Check the variables that contain "zero" as its value.

```
In [8]:  df0[df0 ==0].count()

Out[8]:  customerID           0
         gender               0
         SeniorCitizen     5901
         Partner              0
         Dependents           0
         tenure              11
         PhoneService         0
         MultipleLines        0
         InternetService      0
         OnlineSecurity       0
         OnlineBackup         0
         DeviceProtection     0
         TechSupport          0
         StreamingTV          0
         StreamingMovies      0
         Contract             0
         PaperlessBilling     0
         PaymentMethod        0
         MonthlyCharges       0
         TotalCharges         0
         Churn                0
         dtype: int64
```

As we can clearly see only Seniorcitizen and tenure have "0" as its value. Seniorcitizen column contains two values '0' and '1'. '0' indicates customer who are not senior citizen or not aged and '1' indicates customer who are senior citizen or aged. Tenure columns also contains numerical values and shows the time period of customers associated with this company.

Customer Churn Analysis project with Machine learning

Check the columns that contains only categorical data.

```
In [9]: df0.select_dtypes(np.number)
```

There are only three columns(variables) of numerical datatype.

Out[9]:

	SeniorCitizen	tenure	MonthlyCharges
0	0	1	29.85
1	0	34	56.95
2	0	2	53.85
3	0	45	42.30
4	0	2	70.70
...	...	...	...
7038	0	24	84.80
7039	0	72	103.20
7040	0	11	29.60
7041	1	4	74.40
7042	0	66	105.65

7043 rows × 3 columns

Now, let's check the variables that contains categorical data i.e., datapes= 'object'.

```
df0.select_dtypes('object')
```

	customerID	gender	Partner	Dependents	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	TotalCharges	Churn
0	7590-VHVEG	Female	Yes	No	No	No phone service	DSL	No	Yes	No	No	No	No	Month-to-month	Yes	Electronic check	29.85	No
1	5575-GNVDE	Male	No	No	Yes	No	DSL	Yes	No	Yes	No	No	No	One year	No	Mailed check	1889.5	No
2	3668-QPYBK	Male	No	No	Yes	No	DSL	Yes	Yes	No	No	No	No	Month-to-month	Yes	Mailed check	108.15	Yes
3	7795-CFOCW	Male	No	No	No	No phone service	DSL	Yes	No	Yes	Yes	No	No	One year	No	Bank transfer (automatic)	1840.75	No
4	9237-HQITU	Female	No	No	Yes	No	Fiber optic	No	No	No	No	No	No	Month-to-month	Yes	Electronic check	151.65	Yes
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
7038	6840-RESVB	Male	Yes	Yes	Yes	Yes	DSL	Yes	No	Yes	Yes	Yes	Yes	One year	Yes	Mailed check	1990.5	No
7039	2234-XADUH	Female	Yes	Yes	Yes	Yes	Fiber optic	No	Yes	Yes	No	Yes	Yes	One year	Yes	Credit card (automatic)	7362.9	No
7040	4801-JZAZL	Female	Yes	Yes	No	No phone service	DSL	Yes	No	No	No	No	No	Month-to-month	Yes	Electronic check	346.45	No
7041	8361-LTMKD	Male	Yes	No	Yes	Yes	Fiber optic	No	No	No	No	No	No	Month-to-month	Yes	Mailed check	306.6	Yes
7042	3186-AJIEK	Male	No	No	Yes	No	Fiber optic	Yes	No	Yes	Yes	Yes	Yes	Two year	Yes	Bank transfer (automatic)	6844.5	No

7043 rows × 18 columns

TotalCharges variable is containing continuous datatype but here it is shown as a categorical datatype. we will change it's datatype.

Customer Churn Analysis project with Machine learning

```
df0["TotalCharges"]=pd.to_numeric(df0['TotalCharges'], errors='coerce')
```

It has been changed now. Let’s check it once again.

```
df0.select_dtypes(np.number)
```

Now ,it’s showing as numerical data.

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
0	0	1	29.85	29.85
1	0	34	56.95	1889.50
2	0	2	53.85	108.15
3	0	45	42.30	1840.75
4	0	2	70.70	151.65
...	...	...	...	...
7038	0	24	84.80	1990.50
7039	0	72	103.20	7362.90
7040	0	11	29.60	346.45
7041	1	4	74.40	306.60
7042	0	66	105.65	6844.50

7043 rows × 4 columns

Checking null values once again

```
df0.isnull().sum().sum()
17]: 11
```

It’s showing null values, we have fix it.

Filling the null values

```
df1["TotalCharges"].fillna(df1["TotalCharges"].mean(), inplace=True)
```

```
df1['TotalCharges'].isnull().sum()
]: 0
```

we have fill the null values with mean(). Now, there is no null values present in our dataset.

Dropping the irrelevent column i.e., CustomerID

```
df1.drop(columns='customerID', inplace=True)
```

Customer Churn Analysis project with Machine learning

After this, we done statistical analysis and wrote some observation.

```
n [37]:  df1.describe()
```

Out[37]:

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
count	7043.000000	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692	2283.300441
std	0.368612	24.559481	30.090047	2265.000258
min	0.000000	0.000000	18.250000	18.800000
25%	0.000000	9.000000	35.500000	402.225000
50%	0.000000	29.000000	70.350000	1400.550000
75%	0.000000	55.000000	89.850000	3786.600000
max	1.000000	72.000000	118.750000	8684.800000

.describe() method is used to get descreptive analysis of numerical data with indexes like count, mean, standard deviation, min and max.

The observation from this table is written below.

Observation

1. SeniorCitizen contains only 0 and 1 data.
2. Tenure also has it minimum value as 0 and maximum as 72.
3. Monthly charges varies from 18 to 118 units, with an average of 65(approx).
4. Total charges contain its maximum unit as 8684 and minimum as 18 unit only.

Now data visualization

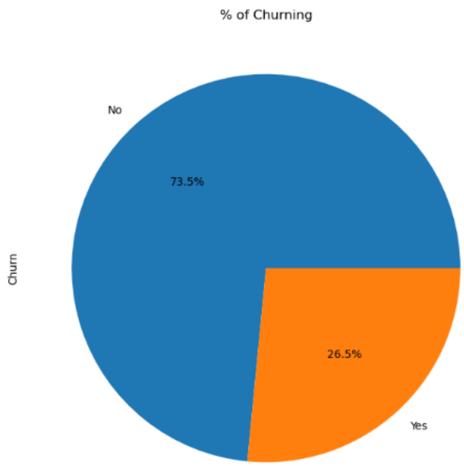
First, import all the necessary libraries to do datavisualization.

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('darkgrid')
%matplotlib inline
import scipy.stats
from scipy.stats import skew
```

Other libraries will be imported when it is required.

Our target variable “Churn” has two values. So , we have created a pie plot of it.

Customer Churn Analysis project with Machine learning

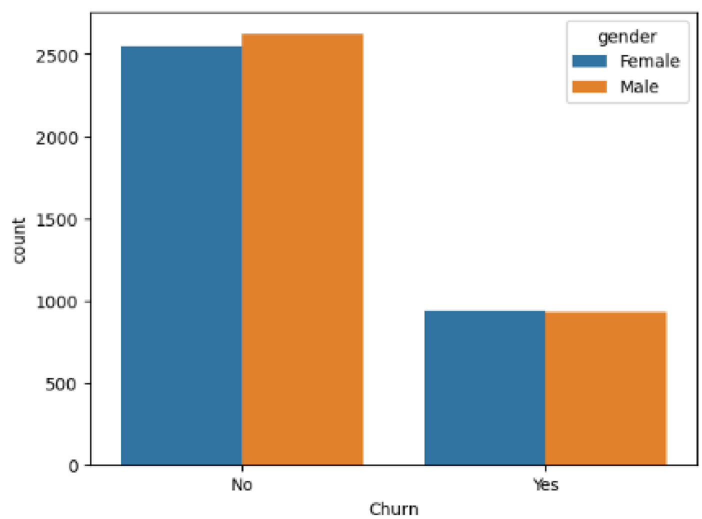


As we already know that in this given data set number of customer who has churned the company is relatively very low and this pie plot is showing it perfectly.

Let's see the churning percentage in different gender.

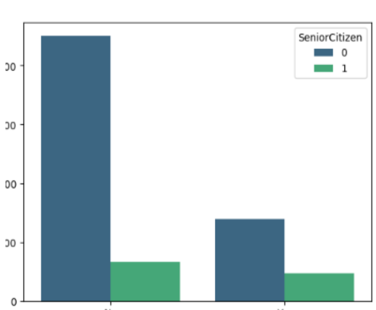
```
sns.countplot("Churn", data=df1,hue="gender")
```

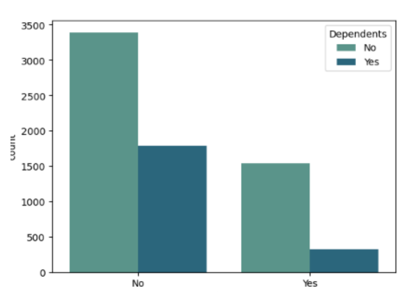
```
<AxesSubplot:xlabel='Churn', ylabel='count'>
```

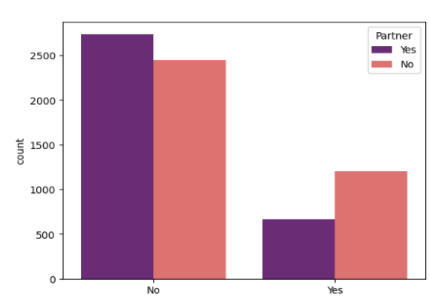


In both female and male have nearly equal percentage of churning.

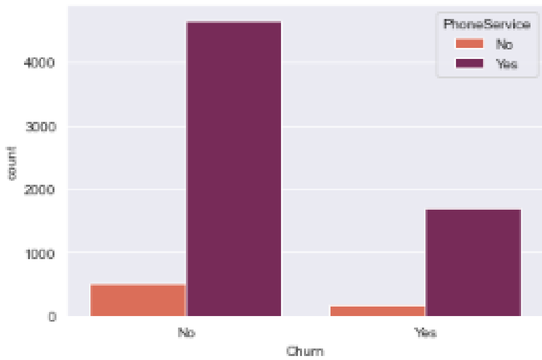
Now, let's check churn percentage different binary features

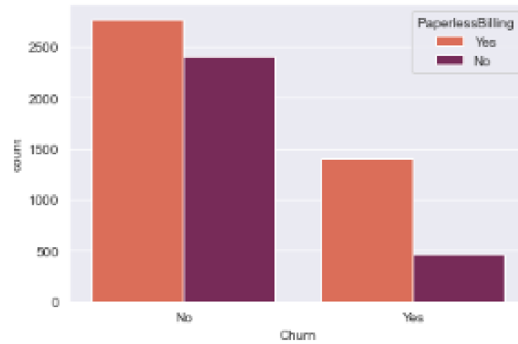






Customer Churn Analysis project with Machine learning





In above all the features churning number of customer is different in each feature.

Done some changes in features value.

- MultipleLines column have 3 value :'No phone service','No' and 'Yes', in which 'no phone service' and 'no' indicates the same thing, so we try to replace 'no phone service' to 'no'.
- OnlineSecurity,OnlineBackup,DeviceProtection,TechSupport,StreamingTV,StreamingMovies, in these columns 'No internet service' and 'no' refers to the same point so we will replace one of these to other.

```
df1['MultipleLines'] = df1['MultipleLines'].replace('No phone service', 'No')

df1[['OnlineSecurity','OnlineBackup',
      'DeviceProtection','TechSupport', 'StreamingTV',
      'StreamingMovies']] = df1[['OnlineSecurity',
                                  'OnlineBackup', 'DeviceProtection',
                                  'TechSupport', 'StreamingTV', 'StreamingMovies']].replace('No internet service', 'No')
```

Using .replace() method.Now, let’s check it out.

Customer Churn Analysis project with Machine learning

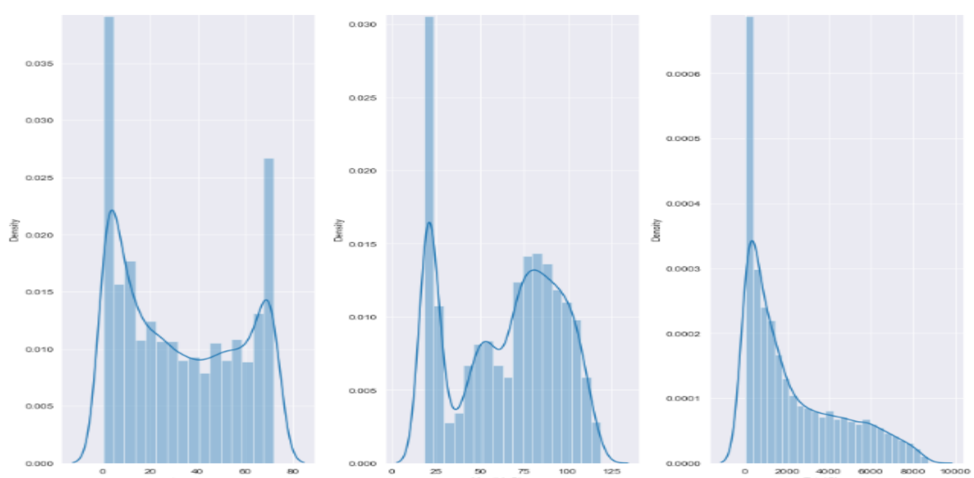
```
get_uniques(df1, get_categorical_columns(df1))
```

```
{'gender': ['Female', 'Male'],
 'Partner': ['Yes', 'No'],
 'Dependents': ['No', 'Yes'],
 'PhoneService': ['No', 'Yes'],
 'MultipleLines': ['No', 'Yes'],
 'InternetService': ['DSL', 'Fiber optic', 'No'],
 'OnlineSecurity': ['No', 'Yes'],
 'OnlineBackup': ['Yes', 'No'],
 'DeviceProtection': ['No', 'Yes'],
 'TechSupport': ['No', 'Yes'],
 'StreamingTV': ['No', 'Yes'],
 'StreamingMovies': ['No', 'Yes'],
 'Contract': ['Month-to-month', 'One year', 'Two year'],
 'PaperlessBilling': ['Yes', 'No'],
 'PaymentMethod': ['Electronic check',
  'Mailed check',
  'Bank transfer (automatic)',
  'Credit card (automatic)'],
 'Churn': ['No', 'Yes']}
```

All the above features values has been changed to correct manner.

Now, let's check the distribution of numerical features using distplot.

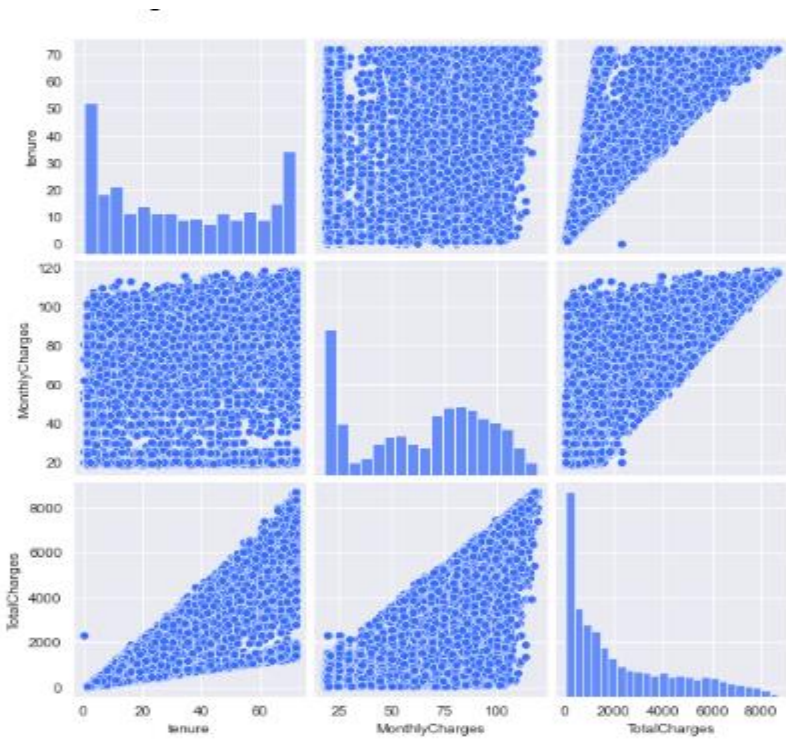
```
i=0
plt.figure(figsize=(12,10))
for column in df1[['tenure','MonthlyCharges','TotalCharges']].columns:
    plt.subplot(1,3,i+1)
    sns.distplot(df1[column],kde=True)
    plt.xlabel(column,fontsize=10)
    i+=1

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



The following features is not distributed well. We will check it later and will try to correct it .

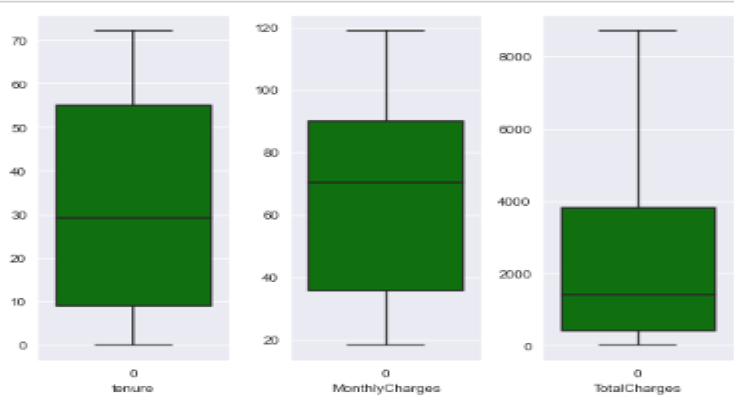
Pairplot:

Customer Churn Analysis project with Machine learning



Checking the outlier:

```
collist=df1[['tenure','MonthlyCharges','TotalCharges']].columns.values

plt.figure(figsize=(7,5))
for i in range(0,3):
    plt.subplot(1,3,i+1)
    sns.boxplot(data=df1[collist[i]],color='green',orient='v').set(xlabel=collist[i])
    plt.tight_layout()
```



There is no outlier detected here. Now, let's move and check the skewness of the following features.

Customer Churn Analysis project with Machine learning

1. First, we have to import required libraries to check the skewness of features.
2. Now, we will check it using .skew
3. As we know skewness values should lies between -0.5 to +0.5, if any feature's skewness is not in between this range will transform the particular feature.
4. Let's do it now

```
import scipy.stats
from scipy.stats import skew
```

Libraries imported.

```
df2[['tenure','MonthlyCharges','TotalCharges']].skew()
```

```
tenure            0.239540
MonthlyCharges   -0.220524
TotalCharges      0.962394
dtype: float64
```

Only totalcharges's skew value is not between above range. Let's try to transform it with np.log1p method.

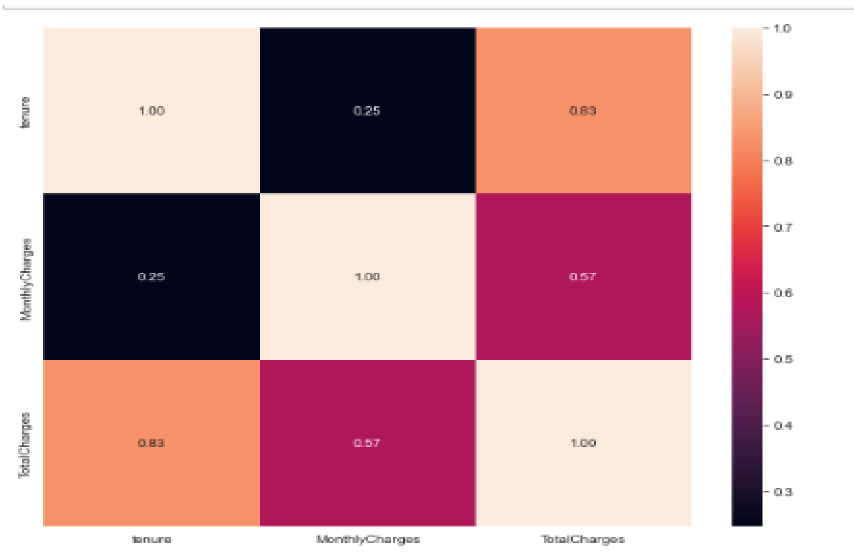
```
df2['TotalCharges']=np.log1p(df2['TotalCharges'])
```

```
df2[['tenure','MonthlyCharges','TotalCharges']].skew()
```

```
tenure            0.239540
MonthlyCharges   -0.220524
TotalCharges     -0.745402
dtype: float64
```

It skew value is decreased little bit. We will leave it as this and move ahead.

Heatmap:



There is not much collinearity. So, we will move to preprocessing now.

Customer Churn Analysis project with Machine learning

In data analysis, we have checked the all null values , duplicate, variables containg zero as it's value. Also changed the datatype of totalcharges variable and filled it null values.Dropped the uneccessary column. For visualization we have different visualization graph like-distplot,countplot,scatterplot,pairplot, pieplot and etc. we have also checked and corrected the distribution of all numerical features. Also used the outlier detection technique. Skew values is reduced by using log transformation. For multicollinearity, we have ploted heatmap. Now, moving to next step i.e. data encoding…

Data encoding:

Import the libraries.

```
from sklearn import preprocessing
```

Now, using label encoder, we will encode the following features:

```
columns=['gender','Partner','Dependents','PhoneService','MultipleLines',
         'OnlineSecurity','OnlineBackup','DeviceProtection','TechSupport',
         'StreamingTV','StreamingMovies','PaperlessBilling','Churn']
for x in columns:
    df3[x]=preprocessing.LabelEncoder().fit_transform(df3[x])

df3
```

Using ordinal encoder, we will encode internetservices and contract features.

```
columns=[['InternetService','Contract']]
for x in columns:
    df4[x]=preprocessing. OrdinalEncoder().fit_transform(df4[x])

df4
```

The remaining features will be encoded using one hot encoder.

```
df5=pd.get_dummies(df4)
```

Building the machine learning model:

For machine learning model, first we have separate the target and feature variables.

Customer Churn Analysis project with Machine learning

```
X = df6.drop('Churn', axis=1)
X


y = df6['Churn']
y
```

Now, we will scale the data using "**standardscaler**"

scaling the data using StandardScaler

```
from sklearn.preprocessing import StandardScaler
```

```
#data scaling
scaler=StandardScaler()
X=scaler.fit_transform(X)
```

Now, we will split the train and test data using train_test_split method of from sklearn.model_selection.

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25)
```

in the practise project, we were told to use only 25% for testing.

Important libraries for building the different machine learning model:

Customer Churn Analysis project with Machine learning

```python
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.dummy import DummyClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score,recall_score,f1_score,precision_score
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_auc_score
from sklearn.metrics import plot_roc_curve
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import RepeatedStratifiedKFold
from sklearn.metrics import precision_recall_curve
```

Now, we will use the required algorithm to get cross validation score, roc auc score , confusion metrix and classification report.

```python
def model(classifier,x_train,y_train,x_test,y_test):

    classifier.fit(x_train,y_train)
    prediction = classifier.predict(x_test)
    cv = RepeatedStratifiedKFold(n_splits = 10,n_repeats = 3,random_state = 1)
    print("Cross Validation Score : ",'{0:.2%}'.format(cross_val_score(classifier,x_train,y_train,cv = cv,scoring = 'roc_auc'
    print("ROC_AUC Score : ",'{0:.2%}'.format(roc_auc_score(y_test,prediction)))
    plot_roc_curve(classifier, x_test,y_test)
    plt.title('ROC_AUC_Plot')
    plt.show()

def model_evaluation(classifier,x_test,y_test):

    # Confusion Matrix
    cm = confusion_matrix(y_test,classifier.predict(x_test))
    names = ['True Neg','False Pos','False Neg','True Pos']
    counts = [value for value in cm.flatten()]
    percentages = ['{0:.2%}'.format(value) for value in cm.flatten()/np.sum(cm)]
    labels = [f'{v1}\n{v2}\n{v3}' for v1, v2, v3 in zip(names,counts,percentages)]
    labels = np.asarray(labels).reshape(2,2)
    sns.heatmap(cm,annot = labels,cmap = 'Blues',fmt ='')

    # Classification Report
    print(classification_report(y_test,classifier.predict(x_test)))
```

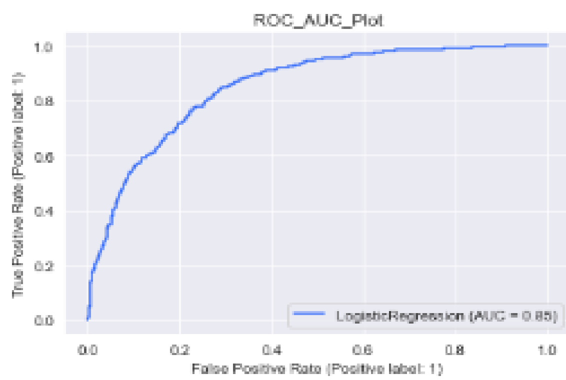
Customer Churn Analysis project with Machine learning

cross validation score, roc auc score, roc auc curve, classification report and confusion metrix of different classification machine learning model.

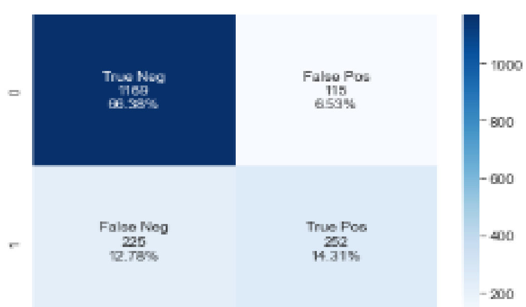
1. Logistic regression.

```
model_lr= LogisticRegression()
model(model_lr,X_train,y_train,X_test,y_test)
model_evaluation(model_lr,X_test,y_test)
```

```
Cross Validation Score :  84.97%
ROC_AUC Score :  71.94%
```




```
              precision    recall  f1-score   support

           0       0.84      0.91      0.87      1284
           1       0.69      0.53      0.60       477

    accuracy                           0.81      1761
   macro avg       0.76      0.72      0.74      1761
weighted avg       0.80      0.81      0.80      1761
```



Cross validation score is 84.97%

Roc auc score is 71.94%

Accuracy is 81%.

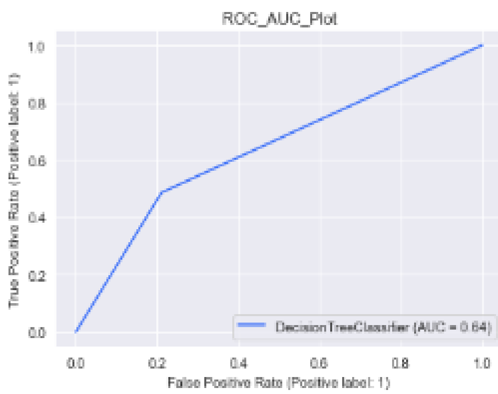
This model has a good accuracy of 81% but we will try to built some other model and check their performance.

Customer Churn Analysis project with Machine learning

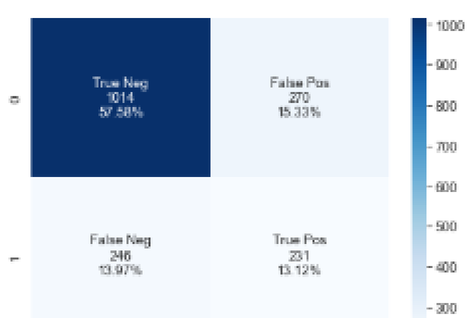
2. Decision tree classifier

```
model_dtc = DecisionTreeClassifier()
model(model_dtc,X_train,y_train,X_test,y_test)
model_evaluation(model_dtc,X_test,y_test)
```

```
Cross Validation Score :  65.27%
ROC_AUC Score :  63.70%
```




```
              precision    recall  f1-score   support

           0       0.80      0.79      0.80      1284
           1       0.46      0.48      0.47       477

    accuracy                           0.71      1761
   macro avg       0.63      0.64      0.63      1761
weighted avg       0.71      0.71      0.71      1761
```



It’s accuracy is even less than logistic regression model. We will try someother model.

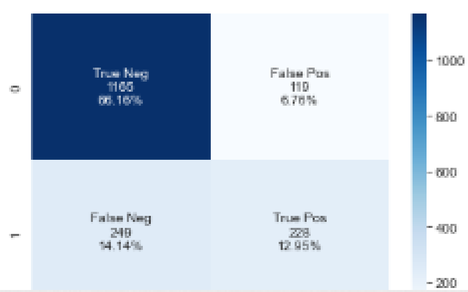
Customer Churn Analysis project with Machine learning

3. Random forest classifier.

```
model_rfc=RandomForestClassifier()
model(model_rfc,X_train,y_train,X_test,y_test)
model_evaluation(model_rfc,X_test,y_test)
```

```
Cross Validation Score :  81.96%
ROC_AUC Score :  69.27%
```


```
              precision    recall  f1-score   support

           0       0.82      0.91      0.86      1284
           1       0.66      0.48      0.55       477

    accuracy                           0.79      1761
   macro avg       0.74      0.69      0.71      1761
weighted avg       0.78      0.79      0.78      1761
```



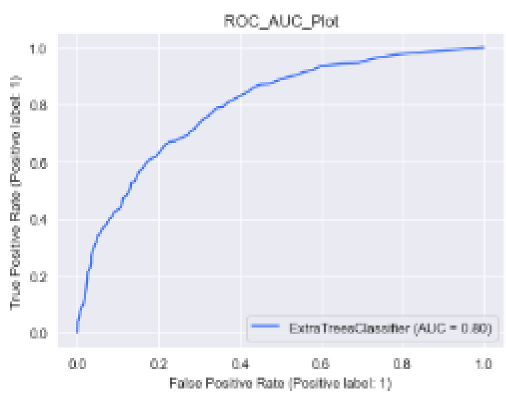
It’s accuracy is better than decision tree classifier but lower than logistic regression. Will try someother model and check their performance.

Customer Churn Analysis project with Machine learning

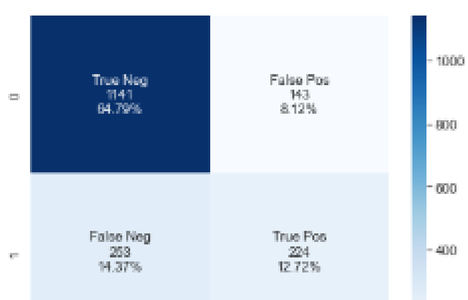
4. Extra tree classifier.

```
model_etc=ExtraTreesClassifier()
model(model_etc,X_train,y_train,X_test,y_test)
model_evaluation(model_etc,X_test,y_test)
```

```
Cross Validation Score :  79.11%
ROC_AUC Score :  67.91%
```



	precision	recall	f1-score	support
0	0.82	0.89	0.85	1284
1	0.61	0.47	0.53	477
accuracy			0.78	1761
macro avg	0.71	0.68	0.69	1761
weighted avg	0.76	0.78	0.77	1761



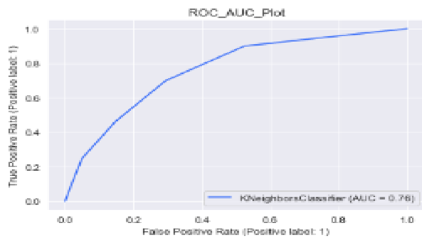
It accuracy is in between logistic regression and random forest classifier, we will see some more model and will compare it with others.

Customer Churn Analysis project with Machine learning

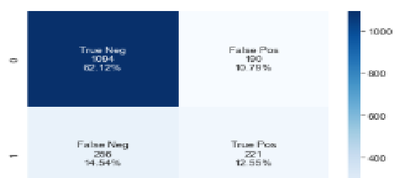
5. KNeighbors classifier.

```
model_knc=KNeighborsClassifier()
model(model_knc,X_train,y_train,X_test,y_test)
model_evaluation(model_knc,X_test,y_test)
```

```
Cross Validation Score :  76.36%
ROC_AUC Score :  65.77%
```




```
              precision    recall  f1-score   support

           0       0.81      0.85      0.83      1284
           1       0.54      0.46      0.50       477

    accuracy                           0.75      1761
   macro avg       0.67      0.66      0.66      1761
weighted avg       0.74      0.75      0.74      1761
```



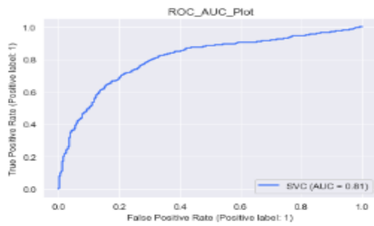
It’s accuracy is also not that good. So, we will try other model.

6. SVC.

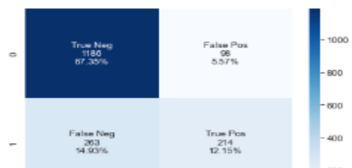
Customer Churn Analysis project with Machine learning

```
model_svc=SVC()
model(model_svc,X_train,y_train,X_test,y_test)
model_evaluation(model_svc,X_test,y_test)
```

```
Cross Validation Score :  80.83%
ROC_AUC Score :  68.62%
```




```
              precision    recall  f1-score   support

           0       0.82      0.92      0.87      1284
           1       0.69      0.45      0.54       477

    accuracy                           0.80      1761
   macro avg       0.75      0.69      0.71      1761
weighted avg       0.78      0.80      0.78      1761
```

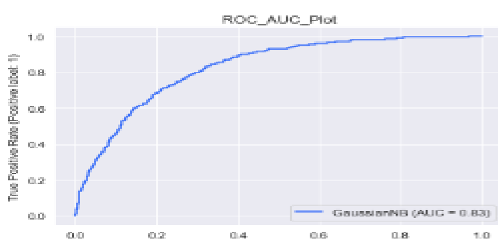


It’s accuracy is better than the above classifier but lesser than logistic regression model.

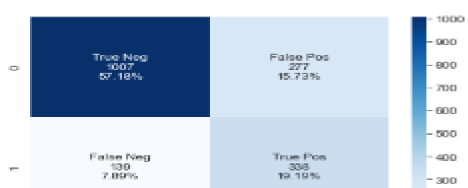
7. GausianNB.

```
model_gnb=GaussianNB()
model(model_gnb,X_train,y_train,X_test,y_test)
model_evaluation(model_gnb,X_test,y_test)
```

```
Cross Validation Score :  82.96%
ROC_AUC Score :  74.64%
```




```
              precision    recall  f1-score   support

           0       0.88      0.78      0.83      1284
           1       0.55      0.71      0.62       477

    accuracy                           0.76      1761
   macro avg       0.71      0.75      0.72      1761
weighted avg       0.79      0.76      0.77      1761
```



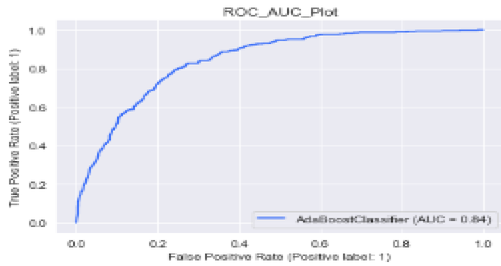
It’s accuracy is also not that good.

8. AdaBoost classifier.

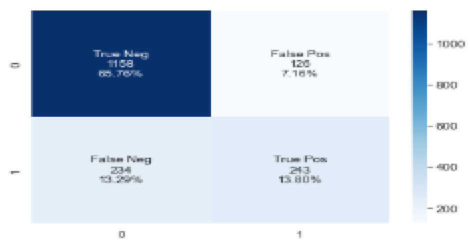
Customer Churn Analysis project with Machine learning

```
model(model_abc,X_train,y_train,X_test,y_test)
model_evaluation(model_abc,X_test,y_test)
```

```
Cross Validation Score :  84.34%
ROC_AUC Score :  70.57%
```




```
              precision    recall  f1-score   support

           0       0.83      0.90      0.87      1284
           1       0.66      0.51      0.57       477

    accuracy                           0.80      1761
   macro avg       0.75      0.71      0.72      1761
weighted avg       0.78      0.80      0.79      1761
```

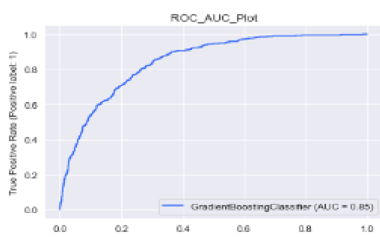


It has better accuracy than someothers model and also lesser than some.

9. Gradient Boosting classifier.

```
model(model_gbc,X_train,y_train,X_test,y_test)
model_evaluation(model_gbc,X_test,y_test)
```

```
Cross Validation Score :  84.25%
ROC_AUC Score :  71.32%
```




```
              precision    recall  f1-score   support

           0       0.84      0.91      0.87      1284
           1       0.67      0.52      0.59       477

    accuracy                           0.80      1761
   macro avg       0.75      0.71      0.73      1761
weighted avg       0.79      0.80      0.79      1761
```

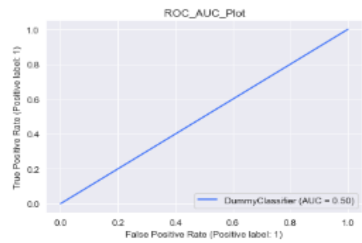
It has pretty much similar accuracy than above model.

Customer Churn Analysis project with Machine learning

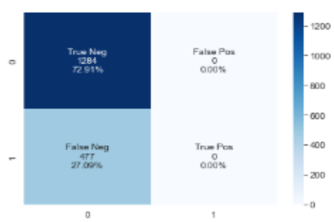
10. Dummy classifier.

```
model(model_dc,X_train,y_train,X_test,y_test)
model_evaluation(model_dc,X_test,y_test)
```

```
Cross Validation Score :  50.00%
ROC_AUC Score :  50.00%
```




```
              precision    recall  f1-score   support

           0       0.73      1.00      0.84      1284
           1       0.00      0.00      0.00       477

    accuracy                           0.73      1761
   macro avg       0.36      0.50      0.42      1761
weighted avg       0.53      0.73      0.61      1761
```



It has worst accuracy than others.

Out of all above different models, Logistic regression has highest accuracy and better cross validation score with roc auc score. So, we will tune this using gridsearchcv library with required parameters and check, if the accuracy of the model is increased or decreased or have same as it was before the tuning.

Library: from sklearn.model_selection import GridSearchCV

Customer Churn Analysis project with Machine learning

```
model_lr=LogisticRegression(solver='liblinear')
param={'C':(0.2,0.3,0.35,0.45,0.55),'fit_intercept':('True','False')}
clf=GridSearchCV(model_lr,param)
clf.fit(X_train,y_train)
```

```
]: GridSearchCV(estimator=LogisticRegression(solver='liblinear'),
                param_grid={'C': (0.2, 0.3, 0.35, 0.45, 0.55),
                            'fit_intercept': ('True', 'False')})
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
#find the best params
clf.best_params_
```

```
]: {'C': 0.2, 'fit_intercept': 'True'}
```

```
#print the best score
clf.best_score_
```

```
]: 0.804427374501878
```

```
grid_search=LogisticRegression(C= 0.2, penalty= 'l1', solver= 'liblinear')
grid_search.fit(X_train,y_train)
```

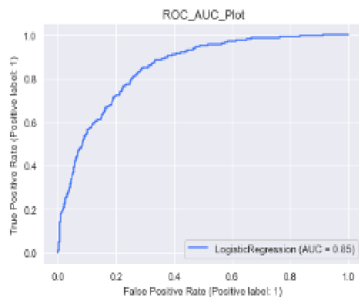
In above algorithm we have used the hyperparameter of logistic regression and than fit it with X_train and y_train data. After this, we will get our best parameter for this model, using this “best_params_” we will again fit the X_train and y_train, and will predict the X_test. After all this , we will again built the logistic regression model with these parameter and these data.

```
grid_search=LogisticRegression(C= 0.2, penalty= 'l1', solver= 'liblinear')
grid_search.fit(X_train,y_train)
y_pred1=grid_search.predict(X_test)

cv = RepeatedStratifiedKFold(n_splits = 10,n_repeats = 3,random_state = 1)
print("Cross Validation Score : ",'{0:.2%}'.format(cross_val_score(grid_search,X_train,y_train,cv = cv,scoring = 'roc_auc').r
print("ROC_AUC Score : ",'{0:.2%}'.format(roc_auc_score(y_test,y_pred1)))
plot_roc_curve(grid_search, X_test,y_test)
plt.title('ROC_AUC_Plot')
plt.show()

cm = confusion_matrix(y_test,grid_search.predict(X_test))
names = ['True Neg','False Pos','False Neg','True Pos']
counts = [value for value in cm.flatten()]
percentages = ['{0:.2%}'.format(value) for value in cm.flatten()/np.sum(cm)]
labels = [f'{v1}\n{v2}\n{v3}' for v1, v2, v3 in zip(names,counts,percentages)]
labels = np.asarray(labels).reshape(2,2)
sns.heatmap(cm,annot = labels,cmap = 'Blues',fmt ='')

# Classification Report
print(classification_report(y_test,grid_search.predict(X_test)))
```

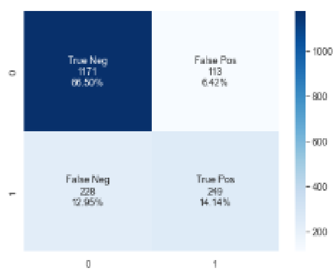
This is the required algorithm.Now, will check the performance of this model and will compare it with the model before tuning.

Customer Churn Analysis project with Machine learning

```
Cross Validation Score :  85.00%
ROC_AUC Score :  71.70%
```



	precision	recall	f1-score	support
0	0.84	0.91	0.87	1284
1	0.69	0.52	0.59	477
accuracy			0.81	1761
macro avg	0.76	0.72	0.73	1761
weighted avg	0.80	0.81	0.80	1761



If we see both model (before tuning and after tuning) closely, we can clearly observe that the accuracy and other performance parameter have increased by very small margin. It's cross validation score and roc auc score has also increased.

Now, we will save the model using pickle library.

```
import pickle
filename='customerchurn.pkl'
pickle.dump(model_lr,open(filename,'wb'))
```

Conclusion:-

From the above modelling technique, we can reduce the customer churn percentage by analysis all features individualy and can get the exact reasons of why the existing customer is churning or likely to churn. This model will also help the company to retain the existing customer by providing slight changes in the different services(mentioned as features in this model). Companies very well know the the expenses of keeping an existing customer is far less than to get new customer and they can also make the required changes to keep their existing customer as if they know the exact reasons of churning. These machine learning model is vastly used to get the exact reasons of churning and also help in lowering the churning rate of any company.