In [2]:
```
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
```

In [4]:
```
data=pd.read_csv("/home/placement/Downloads/TelecomCustomerChurn.csv")
```

In [3]:
```
data.describe()
```

Out[3]:

	SeniorCitizen	tenure	MonthlyCharges
count	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692
std	0.368612	24.559481	30.090047
min	0.000000	0.000000	18.250000
25%	0.000000	9.000000	35.500000
50%	0.000000	29.000000	70.350000
75%	0.000000	55.000000	89.850000
max	1.000000	72.000000	118.750000

In [4]:
```
data.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
```

In [5]:
```
data.head()
```

Out[5]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtec
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	

5 rows × 21 columns

In [6]:
```
data.dtypes
```

Out[6]:
```
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         object
```

In [7]:
```
data['TotalCharges'] = pd.to_numeric(data['TotalCharges'], errors='coerce')
data.dtypes
```

Out[7]:
```
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
```

In [8]:
```
data.isna().sum()
```

Out[8]:
```
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       11
Churn               0
dtype: int64
```

In [9]:
```
data['TotalCharges']=data['TotalCharges'].fillna(data['TotalCharges'].median())
```

In [10]:
```
data.isna().sum()
```

Out[10]:
```
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
```

In [11]:
```
data['SeniorCitizen']=data['SeniorCitizen'].map({0:'No',1:'Yes'})
```

In [25]:
```
data.head()
```

Out[25]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtec
0	7590-VHVEG	Female	No	Yes	No	1	No	No phone service	DSL	No	...	
1	5575-GNVDE	Male	No	No	No	34	Yes	No	DSL	Yes	...	
2	3668-QPYBK	Male	No	No	No	2	Yes	No	DSL	Yes	...	
3	7795-CFOCW	Male	No	No	No	45	No	No phone service	DSL	Yes	...	
4	9237-HQITU	Female	No	No	No	2	Yes	No	Fiber optic	No	...	

5 rows × 21 columns

In [26]:
```
y=data['Churn']
x=data.drop(['customerID','Churn'],axis=1)
```

In [27]:
```
x=pd.get_dummies(x)
```

In [28]:
```
x
```

Out[28]:

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

7043 rows × 46 columns

In [30]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.33,random_state=42)
```

In [31]:
```
x_train.isna().sum()
```

Out[31]:
```
tenure                                    0
MonthlyCharges                            0
TotalCharges                              0
gender_Female                             0
gender_Male                               0
SeniorCitizen_No                          0
SeniorCitizen_Yes                         0
Partner_No                                0
Partner_Yes                               0
Dependents_No                             0
Dependents_Yes                            0
PhoneService_No                           0
PhoneService_Yes                          0
MultipleLines_No                          0
MultipleLines_No phone service            0
MultipleLines_Yes                         0
InternetService_DSL                       0
InternetService_Fiber optic               0
InternetService_No                        0
OnlineSecurity_No                         0
OnlineSecurity_No internet service        0
OnlineSecurity_Yes                        0
OnlineBackup_No                           0
OnlineBackup_No internet service          0
OnlineBackup_Yes                          0
DeviceProtection_No                       0
DeviceProtection_No internet service      0
DeviceProtection_Yes                      0
TechSupport_No                            0
TechSupport_No internet service           0
TechSupport_Yes                           0
StreamingTV_No                            0
StreamingTV_No internet service           0
StreamingTV_Yes                           0
StreamingMovies_No                        0
StreamingMovies_No internet service       0
StreamingMovies_Yes                       0
Contract_Month-to-month                   0
Contract_One year                         0
```

```
Contract_Two year                          0
PaperlessBilling_No                        0
PaperlessBilling_Yes                       0
PaymentMethod_Bank transfer (automatic)    0
PaymentMethod_Credit card (automatic)      0
PaymentMethod_Electronic check             0
PaymentMethod_Mailed check                 0
dtype: int64
```

In [32]:
```
%time
from sklearn.model_selection import GridSearchCV #GridSearchCV is for parameter tuning
from sklearn.ensemble import RandomForestClassifier
cls=RandomForestClassifier()
n_estimators=[25,50,75,100,125,150,175,200] #number of decision trees in the forest, default = 100
criterion=['gini','entropy'] #criteria for choosing nodes default = 'gini'
max_depth=[3,5,10] #maximum number of nodes in a tree default = None (it will go till all possible nodes)
parameters={'n_estimators': n_estimators,'criterion':criterion,'max_depth':max_depth} #this will undergo 8*2
RFC_cls = GridSearchCV(cls, parameters)
RFC_cls.fit(x_train,y_train)
```

```
CPU times: user 7 µs, sys: 1 µs, total: 8 µs
Wall time: 14.1 µs
```

Out[32]:
```
GridSearchCV(estimator=RandomForestClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [3, 5, 10],
                         'n_estimators': [25, 50, 75, 100, 125, 150, 175, 200]})
```

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.

In [33]:
```
RFC_cls.best_params_
```

Out[33]:
```
{'criterion': 'entropy', 'max_depth': 10, 'n_estimators': 200}
```

In [36]:
```
cls=RandomForestClassifier(n_estimators=200,criterion='entropy',max_depth=10)
```

In [37]:
```
cls.fit(x_train,y_train)
```

Out[37]:
```
RandomForestClassifier(criterion='entropy', max_depth=10, n_estimators=200)
```
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.

In [39]:
```
rfy_pred=cls.predict(x_test)
```

In [40]:
```
rfy_pred
```

Out[40]:
```
array(['Yes', 'No', 'No', ..., 'Yes', 'No', 'No'], dtype=object)
```

In [41]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,rfy_pred)
```

Out[41]:
```
array([[1556,  141],
       [ 296,  332]])
```

In [43]:
```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,rfy_pred)
```

Out[43]:
```
0.8120430107526881
```

In [ ]:
```
#logistic regression
```

In [44]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.33,random_state=42)
```

In [45]:
```
from sklearn.linear_model import LogisticRegression
classifier= LogisticRegression()
classifier.fit(x_train,y_train)
```

Out[45]:
```
LogisticRegression()
```
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.

In [46]:
```
y_pred=classifier.predict(x_test)
y_pred
```

Out[46]:
```
array(['Yes', 'No', 'No', ..., 'Yes', 'No', 'No'], dtype=object)
```

In [47]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred)
```

Out[47]:
```
array([[1538,  159],
       [ 279,  349]])
```

In [48]:
```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

Out[48]:
```
0.8116129032258065
```

In [ ]:
```

```