In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

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

In [3]:
```
data=pd.read_csv("/home/placement/Downloads/TelecomCustomerChurn.csv")
```

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

Out[4]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceP
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	...	
...	...	...	...	...	...	...	...	...	...	...	...	
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes	...	
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No	...	
7040	4801-JZAZL	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes	...	
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	Yes	Fiber optic	No	...	
7042	3186-AJIEK	Male	0	No	No	66	Yes	No	Fiber optic	Yes	...	

7043 rows × 21 columns

In [5]:
```
print(data)
```

```
      customerID  gender  SeniorCitizen Partner Dependents  tenure  \
0     7590-VHVEG  Female              0     Yes         No       1   
1     5575-GNVDE    Male              0      No         No      34   
2     3668-QPYBK    Male              0      No         No       2   
3     7795-CFOCW    Male              0      No         No      45   
4     9237-HQITU  Female              0      No         No       2   
...          ...     ...            ...     ...        ...     ...   
7038  6840-RESVB    Male              0     Yes        Yes      24   
7039  2234-XADUH  Female              0     Yes        Yes      72   
7040  4801-JZAZL  Female              0     Yes        Yes      11   
7041  8361-LTMKD    Male              1     Yes         No       4   
7042  3186-AJIEK    Male              0      No         No      66   

     PhoneService     MultipleLines InternetService OnlineSecurity  ...  \
0              No  No phone service             DSL             No  ...   
1             Yes                No             DSL            Yes  ...   
2             Yes                No             DSL            Yes  ...   
3              No  No phone service             DSL            Yes  ...   
4             Yes                No     Fiber optic             No  ...   
...           ...               ...             ...            ...  ...   
7038          Yes               Yes             DSL            Yes  ...   
7039          Yes               Yes     Fiber optic             No  ...   
7040           No  No phone service             DSL            Yes  ...   
7041          Yes               Yes     Fiber optic             No  ...   
7042          Yes                No     Fiber optic            Yes  ...   

     DeviceProtection TechSupport StreamingTV StreamingMovies        Contract  \
0                  No          No          No              No  Month-to-month   
1                 Yes          No          No              No        One year   
2                  No          No          No              No  Month-to-month   
3                 Yes         Yes          No              No        One year   
4                  No          No          No              No  Month-to-month   
...               ...         ...         ...             ...             ...   
7038              Yes         Yes         Yes             Yes        One year   
7039              Yes          No         Yes             Yes        One year   
7040               No          No          No              No  Month-to-month   
7041               No          No          No              No  Month-to-month   
7042              Yes         Yes         Yes             Yes        Two year   
```

```
     PaperlessBilling              PaymentMethod MonthlyCharges  TotalCharges  \
0                 Yes           Electronic check          29.85         29.85
1                  No               Mailed check          56.95        1889.5
2                 Yes               Mailed check          53.85        108.15
3                  No  Bank transfer (automatic)          42.30       1840.75
4                 Yes           Electronic check          70.70        151.65
...               ...                        ...            ...           ...
7038              Yes               Mailed check          84.80        1990.5
7039              Yes    Credit card (automatic)         103.20        7362.9
7040              Yes           Electronic check          29.60        346.45
7041              Yes               Mailed check          74.40         306.6
7042              Yes  Bank transfer (automatic)         105.65        6844.5

     Churn
0       No
1       No
2      Yes
3       No
4      Yes
...    ...
7038    No
7039    No
7040    No
7041   Yes
7042    No

[7043 rows x 21 columns]
```

In [6]:
```
data.head()
```

Out[6]:

	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 [7]:
```
data.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
```

In [8]:
```
data.dtypes
```

Out[8]:
```
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
Churn               object
dtype: object
```

In [9]:
```
data['TotalCharges']=pd.to_numeric(data['TotalCharges'],errors='coerce')
data.dtypes
```

Out[9]:
```
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 [10]:
```
data['SeniorCitizen']=data['SeniorCitizen'].map({0:'No',1:'Yes'})
```

In [11]:
```
data
```

Out[11]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DevicePro
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	...	
...	...	...	...	...	...	...	...	...	...	...	...	
7038	6840-RESVB	Male	No	Yes	Yes	24	Yes	Yes	DSL	Yes	...	
7039	2234-XADUH	Female	No	Yes	Yes	72	Yes	Yes	Fiber optic	No	...	
7040	4801-JZAZL	Female	No	Yes	Yes	11	No	No phone service	DSL	Yes	...	
7041	8361-LTMKD	Male	Yes	Yes	No	4	Yes	Yes	Fiber optic	No	...	
7042	3186-AJIEK	Male	No	No	No	66	Yes	No	Fiber optic	Yes	...	

7043 rows × 21 columns

In [12]:
```
data['TotalCharges']=data['TotalCharges'].fillna(data['TotalCharges'].median())
```

In [13]:
```
data.isna().sum()
```

Out[13]:
```
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 [14]:
```
y=data['Churn']
x=data.drop(['customerID','Churn'],axis=1)
```

In [15]:
```
x
```

Out[15]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProte
0	Female	No	Yes	No	1	No	No phone service	DSL	No	Yes	
1	Male	No	No	No	34	Yes	No	DSL	Yes	No	
2	Male	No	No	No	2	Yes	No	DSL	Yes	Yes	
3	Male	No	No	No	45	No	No phone service	DSL	Yes	No	
4	Female	No	No	No	2	Yes	No	Fiber optic	No	No	
...	...	...	...	...	...	...	...	...	...	...	...
7038	Male	No	Yes	Yes	24	Yes	Yes	DSL	Yes	No	
7039	Female	No	Yes	Yes	72	Yes	Yes	Fiber optic	No	Yes	
7040	Female	No	Yes	Yes	11	No	No phone service	DSL	Yes	No	
7041	Male	Yes	Yes	No	4	Yes	Yes	Fiber optic	No	No	
7042	Male	No	No	No	66	Yes	No	Fiber optic	Yes	No	

7043 rows × 19 columns

In [16]:
```
y
```

Out[16]:
```
0        No
1        No
2       Yes
3        No
4       Yes
       ...
7038     No
7039     No
7040     No
7041    Yes
7042     No
Name: Churn, Length: 7043, dtype: object
```

In [17]:
```
x=pd.get_dummies(x)
x
```

Out[17]:

	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 [18]:
```
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 [19]:
```
x_train.isna().sum()
```

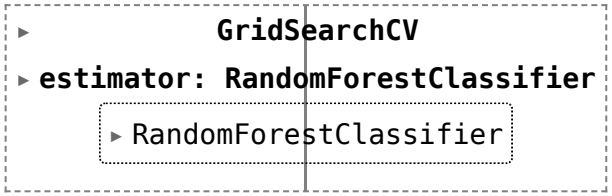
Out[19]:
```
tenure                                    0
MonthlyCharges                            0
TotalCharges                              0
gender_Female                             0
gender_Male                               0
SeniorCitizen_No                          0
SeniorCitizen_Yes                         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 [20]:
```
#importing Randaom Forest Classifier from sklearn.ensemble
%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 6 µs, sys: 0 ns, total: 6 µs
Wall time: 14.8 µs
```

Out[20]:



In [21]:
```
RFC_cls.best_params_
```

Out[21]:
```
{'criterion': 'entropy', 'max_depth': 10, 'n_estimators': 100}
```

In [23]:
```
cls=RandomForestClassifier(n_estimators=200,criterion='entropy',max_depth=10)
```

In [24]:
```
cls.fit(x_train,y_train)
```

Out[24]:
```
RandomForestClassifier
RandomForestClassifier(criterion='entropy', max_depth=10, n_estimators=200)
```

In [25]:
```
rfy_pred=cls.predict(x_test)
```

In [26]:
```
rfy_pred
```

Out[26]:
```
array(['Yes', 'No', 'No', ..., 'Yes', 'No', 'No'], dtype=object)
```

In [28]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,rfy_pred)
```

Out[28]:
```
array([[1551,  146],
       [ 297,  331]])
```

In [29]:
```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,rfy_pred)
```

Out[29]:
```
0.8094623655913978
```

In [30]:
```
from sklearn.linear_model import LogisticRegression
classifier=LogisticRegression()
classifier.fit(x_train,y_train)# command for traning / fitting the mode
```

Out[30]:
```
LogisticRegression
LogisticRegression()
```

In [32]:
```
y_pred=classifier.predict(x_test)
```

In [33]:
```
y_pred
```

Out[33]:
```
array(['Yes', 'No', 'No', ..., 'Yes', 'No', 'No'], dtype=object)
```

In [34]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred)
```

Out[34]:
```
array([[1538,  159],
       [ 279,  349]])
```

In [35]:
```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

Out[35]:
```
0.8116129032258065
```

In [ ]:
```

```