

In [1]:

```
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
```

In [2]:

```
data1=pd.read_csv("/home/placement/Downloads/TelecomCustomerChurn.csv")
```

In [3]:

```
data1.isna().sum()
```

Out[3]:

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 [4]:

```
data1.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]:

```
data1['TotalCharges']=pd.to_numeric(data1['TotalCharges'],errors='coerce')
```

In [6]:

```
data1.isna().sum()
```

Out[6]:

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

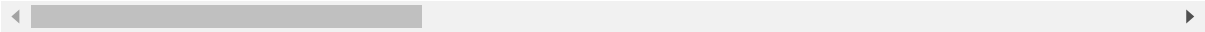
In [7]:

```
data1=data1.fillna(data1.median())
data1
```

Out[7]:

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

7043 rows × 21 columns



In [8]:

```
data1.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   float64
20   Churn                 7043 non-null   object
dtypes: float64(2), int64(2), object(17)
memory usage: 1.1+ MB
```

In [9]:

```
y=data1['Churn']
x=data1.drop(['customerID', 'Churn'],axis=1)
```

In [10]:

```
x=pd.get_dummies(x)  
x.isna().sum()
```

Out[10]:

SeniorCitizen	0
tenure	0
MonthlyCharges	0
TotalCharges	0
gender_Female	0
gender_Male	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 []:

In [11]:

x.head()

Out[11]:

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges	gender_Female	gender_Male	Partner_I
0	0	1	29.85	29.85	1	0	
1	0	34	56.95	1889.50	0	1	
2	0	2	53.85	108.15	0	1	
3	0	45	42.30	1840.75	0	1	
4	0	2	70.70	151.65	1	0	

5 rows × 45 columns

In [12]:

```
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 [13]:

```
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
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 leaf nodes)
parameters={'n_estimators': n_estimators,'criterion':criterion,'max_depth':max_depth}
RFC_cls = GridSearchCV(cls, parameters)
RFC_cls.fit(x_train,y_train)
```

Out[13]:

```
GridSearchCV
  estimator: RandomForestClassifier
    RandomForestClassifier
```

In [14]:

RFC_cls.best_params_

Out[14]:

{'criterion': 'gini', 'max_depth': 10, 'n_estimators': 150}

In [15]:

```
cls=RandomForestClassifier(n_estimators=200,criterion='entropy',max_depth=10)
```

In [16]:

```
cls.fit(x_train,y_train)
```

Out[16]:

```
▼ RandomForestClassifier  
RandomForestClassifier(criterion='entropy', max_depth=10, n_estimators=200)
```

In [17]:

```
p=cls.predict(x_test)
```

In [18]:

```
p  
array(['Yes', 'No', 'No', ..., 'Yes', 'No', 'No'], dtype=object)
```

In [19]:

```
from sklearn.metrics import confusion_matrix  
confusion_matrix(y_test,p)
```

Out[19]:

```
array([[1552, 145],  
       [ 297, 331]])
```

In [20]:

```
from sklearn.metrics import accuracy_score  
accuracy_score(y_test,p)
```

Out[20]:

```
0.8098924731182796
```

In [21]:

```
from sklearn.linear_model import LogisticRegression  
clas=LogisticRegression()  
clas.fit(x_train,y_train)
```

Out[21]:

```
▼ LogisticRegression  
LogisticRegression()
```

In [22]:

```
y_pred=clas.predict(x_test)
```


In [23]:

```
y_pred
```

Out[23]:

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

In [24]:

```
from sklearn.metrics import confusion_matrix  
confusion_matrix(y_test,y_pred)
```

Out[24]:

```
array([[1526,  171],  
       [ 266,  362]])
```

In [25]:

```
from sklearn.metrics import accuracy_score  
accuracy_score(y_test,y_pred)
```

Out[25]:

```
0.8120430107526881
```

In []: