In [1]:

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

In [2]:

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

In [3]:

```
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
```

In [4]:
```
data.isna().sum()
```

Out[4]:
```
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 [5]:
```
data["TotalCharges"]=pd.to_numeric(data["TotalCharges"],errors='coerce')
```

```
In [6]: data.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=data.fillna(data.median())
data1
```

Out[7]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DevicePro
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 [8]:

```
data1.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        0
Churn               0
dtype: int64
```

In [9]:

```
data2=data1.drop(['customerID','gender','tenure','PaymentMethod','Dependents','MultipleLines','StreamingMovi
data2
```

Out[9]:

	SeniorCitizen	Partner	PhoneService	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	Contract	MonthlyCharg
0	0	Yes	No	DSL	No	Yes	No	No	Month-to-month	29
1	0	No	Yes	DSL	Yes	No	Yes	No	One year	56
2	0	No	Yes	DSL	Yes	Yes	No	No	Month-to-month	53
3	0	No	No	DSL	Yes	No	Yes	Yes	One year	42
4	0	No	Yes	Fiber optic	No	No	No	No	Month-to-month	70
...	...	...	...	...	...	...	...	...	...	
7038	0	Yes	Yes	DSL	Yes	No	Yes	Yes	One year	84
7039	0	Yes	Yes	Fiber optic	No	Yes	Yes	No	One year	103
7040	0	Yes	No	DSL	Yes	No	No	No	Month-to-month	29
7041	1	Yes	Yes	Fiber optic	No	No	No	No	Month-to-month	74
7042	0	No	Yes	Fiber optic	Yes	No	Yes	Yes	Two year	105

7043 rows × 12 columns

```
In [10]: data2['Churn']=data2['Churn'].map({'No':0,'Yes':1})
         data2
```

Out[10]:

	SeniorCitizen	Partner	PhoneService	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	Contract	MonthlyCharg
0	0	Yes	No	DSL	No	Yes	No	No	Month-to-month	29
1	0	No	Yes	DSL	Yes	No	Yes	No	One year	56
2	0	No	Yes	DSL	Yes	Yes	No	No	Month-to-month	53
3	0	No	No	DSL	Yes	No	Yes	Yes	One year	42
4	0	No	Yes	Fiber optic	No	No	No	No	Month-to-month	70
...	...	...	...	...	...	...	...	...	...	
7038	0	Yes	Yes	DSL	Yes	No	Yes	Yes	One year	84
7039	0	Yes	Yes	Fiber optic	No	Yes	Yes	No	One year	103
7040	0	Yes	No	DSL	Yes	No	No	No	Month-to-month	29
7041	1	Yes	Yes	Fiber optic	No	No	No	No	Month-to-month	74
7042	0	No	Yes	Fiber optic	Yes	No	Yes	Yes	Two year	105

7043 rows × 12 columns

In [11]:
```
data3=pd.get_dummies(data2)
data3
```

Out[11]:

	SeniorCitizen	MonthlyCharges	TotalCharges	Churn	Partner_No	Partner_Yes	PhoneService_No	PhoneService_Yes	InternetService_DSL	Ir
0	0	29.85	29.85	0	0	1	1	0	1	
1	0	56.95	1889.50	0	1	0	0	1	1	
2	0	53.85	108.15	1	1	0	0	1	1	
3	0	42.30	1840.75	0	1	0	1	0	1	
4	0	70.70	151.65	1	1	0	0	1	0	
...	...	...	...	...	...	...	...	...	...	
7038	0	84.80	1990.50	0	0	1	0	1	1	
7039	0	103.20	7362.90	0	0	1	0	1	0	
7040	0	29.60	346.45	0	0	1	1	0	1	
7041	1	74.40	306.60	1	0	1	0	1	0	
7042	0	105.65	6844.50	0	1	0	0	1	0	

7043 rows × 26 columns

In [12]:
```
y=data3['Churn']
x=data3.drop('Churn',axis=1)
```

In [13]:
```
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 [14]: from sklearn.linear_model import LogisticRegression
         classifier=LogisticRegression()
         classifier.fit(x_train,y_train)

Out[14]: ▾ LogisticRegression

         LogisticRegression()
```

```
In [15]: y_pred=classifier.predict(x_test)
         y_pred

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

```
In [16]: from sklearn.metrics import confusion_matrix
         confusion_matrix(y_test,y_pred)

Out[16]: array([[1510,  187],
                [ 274,  354]])
```

```
In [17]: from sklearn.metrics import accuracy_score
         accuracy_score(y_test,y_pred)

Out[17]: 0.8017204301075269
```

```
In [ ]:
```