In [15]:

```
import pandas as pd
```

In [16]:

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

In [17]:

```
data.describe()
```

Out[17]:

	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 [18]:

```
list(data)
```

Out[18]:

```
['customerID',
 'gender',
 'SeniorCitizen',
 'Partner',
 'Dependents',
 'tenure',
 'PhoneService',
 'MultipleLines',
 'InternetService',
 'OnlineSecurity',
 'OnlineBackup',
 'DeviceProtection',
 'TechSupport',
 'StreamingTV',
 'StreamingMovies',
 'Contract',
 'PaperlessBilling',
 'PaymentMethod',
 'MonthlyCharges',
 'TotalCharges',
 'Churn']
```

In [19]:

```
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 [ ]:

```

```

In [20]:

```
data.head()
```

Out[20]:

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

5 rows × 21 columns

In [21]:

```
data1=data.drop(['customerID','gender','SeniorCitizen','PaymentMethod','PaperlessBi
 'TechSupport','StreamingTV',
 'StreamingMovies','InternetService'],axis=1)
```

In [22]:

```
data1
```

Out[22]:

	Partner	tenure	PhoneService	Contract	MonthlyCharges	TotalCharges	Churn
0	Yes	1	No	Month-to-month	29.85	29.85	No
1	No	34	Yes	One year	56.95	1889.5	No
2	No	2	Yes	Month-to-month	53.85	108.15	Yes
3	No	45	No	One year	42.30	1840.75	No
4	No	2	Yes	Month-to-month	70.70	151.65	Yes
...	...	...	...	...	...	...	...
7038	Yes	24	Yes	One year	84.80	1990.5	No
7039	Yes	72	Yes	One year	103.20	7362.9	No
7040	Yes	11	No	Month-to-month	29.60	346.45	No
7041	Yes	4	Yes	Month-to-month	74.40	306.6	Yes
7042	No	66	Yes	Two year	105.65	6844.5	No

7043 rows × 7 columns

In [23]:

```
data1['Partner']=data['Partner'].map({'Yes':1,'No':0})
```

In [24]:

```
data1['PhoneService']=data['PhoneService'].map({'Yes':1,'No':0})
```

In [25]:

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

In [26]:

```
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Partner         7043 non-null   int64  
 1   tenure          7043 non-null   int64  
 2   PhoneService    7043 non-null   int64  
 3   Contract        7043 non-null   object 
 4   MonthlyCharges  7043 non-null   float64
 5   TotalCharges    7032 non-null   float64
 6   Churn           7043 non-null   object 
dtypes: float64(2), int64(3), object(2)
memory usage: 385.3+ KB
```

In [27]:

```
data1
```

Out[27]:

	Partner	tenure	PhoneService	Contract	MonthlyCharges	TotalCharges	Churn
0	1	1	0	Month-to-month	29.85	29.85	No
1	0	34	1	One year	56.95	1889.50	No
2	0	2	1	Month-to-month	53.85	108.15	Yes
3	0	45	0	One year	42.30	1840.75	No
4	0	2	1	Month-to-month	70.70	151.65	Yes
...	...	...	...	...	...	...	...
7038	1	24	1	One year	84.80	1990.50	No
7039	1	72	1	One year	103.20	7362.90	No
7040	1	11	0	Month-to-month	29.60	346.45	No
7041	1	4	1	Month-to-month	74.40	306.60	Yes
7042	0	66	1	Two year	105.65	6844.50	No

7043 rows × 7 columns

In [30]:

```
data1['Churn']=data['Churn'].map({'Yes':1,'No':0})
```

In [31]:

```
data1
```

Out[31]:

	Partner	tenure	PhoneService	Contract	MonthlyCharges	TotalCharges	Churn
0	1	1	0	Month-to-month	29.85	29.85	0
1	0	34	1	One year	56.95	1889.50	0
2	0	2	1	Month-to-month	53.85	108.15	1
3	0	45	0	One year	42.30	1840.75	0
4	0	2	1	Month-to-month	70.70	151.65	1
...	...	...	...	...	...	...	...
7038	1	24	1	One year	84.80	1990.50	0
7039	1	72	1	One year	103.20	7362.90	0
7040	1	11	0	Month-to-month	29.60	346.45	0
7041	1	4	1	Month-to-month	74.40	306.60	1
7042	0	66	1	Two year	105.65	6844.50	0

7043 rows × 7 columns

In [32]:

```
data1=pd.get_dummies(data1)
```

In [33]:

```
data1
```

Out[33]:

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

7043 rows × 9 columns

In [37]:

```
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 9 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Partner                  7043 non-null   int64  
 1   tenure                   7043 non-null   int64  
 2   PhoneService             7043 non-null   int64  
 3   MonthlyCharges           7043 non-null   float64
 4   TotalCharges             7032 non-null   float64
 5   Churn                    7043 non-null   int64  
 6   Contract_Month-to-month  7043 non-null   uint8  
 7   Contract_One year        7043 non-null   uint8  
 8   Contract_Two year        7043 non-null   uint8  
dtypes: float64(2), int64(4), uint8(3)
memory usage: 350.9 KB
```

In [ ]:

```

```

In [52]:

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

In [53]:

```
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 [54]:

```
x_test
```

Out[54]:

	Partner	tenure	PhoneService	MonthlyCharges	TotalCharges	Contract_Month-to-month	Contract
185	1	1	0	24.80	24.80	1	
2715	0	41	1	25.25	996.45	1	
3825	1	52	1	19.35	1031.70	0	
1807	0	1	1	76.35	76.35	1	
132	0	67	1	50.55	3260.10	0	
...	...	...	...	...	...	...	
4147	1	71	1	24.85	1901.00	0	
3542	1	29	0	55.35	1636.95	0	
3759	0	7	1	89.35	631.85	1	
1114	1	32	1	98.85	3145.90	1	
4958	1	59	1	94.75	5597.65	1	

2325 rows × 8 columns

In [46]:

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

Out[46]:

```
Partner                    0
tenure                     0
PhoneService               0
MonthlyCharges             0
TotalCharges               0
Churn                      0
Contract_Month-to-month    0
Contract_One year          0
Contract_Two year          0
dtype: int64
```

In [51]:

```
data1=data1.fillna(data1.mean())
```

In [60]:

```
import warnings
warnings.filterwarnings("ignore")
```

In [61]:

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

Out[61]:

▾ LogisticRegression

LogisticRegression()

In [56]:

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

In [57]:

```
y_pred
```

Out[57]:

```
array([1, 0, 0, ..., 1, 1, 0])
```

In [58]:

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

Out[58]:

```
array([[1508,  189],
       [ 272,  356]])
```

In [59]:

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

Out[59]:

```
0.8017204301075269
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: