In [59]:
```
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
```

In [60]:
```
data=pd.read_csv("/home/placement/Downloads/TelecomCustomerChurn.csv")
```

In [61]:
```
data['TotalCharges']=pd.to_numeric(data['TotalCharges'],errors='coerce')
```

In [62]:
```
data.describe()
```

Out[62]:

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
count	7043.000000	7043.000000	7043.000000	7032.000000
mean	0.162147	32.371149	64.761692	2283.300441
std	0.368612	24.559481	30.090047	2266.771362
min	0.000000	0.000000	18.250000	18.800000
25%	0.000000	9.000000	35.500000	401.450000
50%	0.000000	29.000000	70.350000	1397.475000
75%	0.000000	55.000000	89.850000	3794.737500
max	1.000000	72.000000	118.750000	8684.800000

In [63]:
```
#data.info()
data.dtypes
```

Out[63]:
```
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 [64]:
```
list(data)
```

Out[64]:
```
['customerID',
 'gender',
 'SeniorCitizen',
 'Partner',
 'Dependents',
 'tenure',
 'PhoneService',
 'MultipleLines',
 'InternetService',
 'OnlineSecurity',
 'OnlineBackup',
 'DeviceProtection',
 'TechSupport',
 'StreamingTV',
 'StreamingMovies',
 'Contract',
 'PaperlessBilling',
 'PaymentMethod',
 'MonthlyCharges',
 'TotalCharges',
 'Churn']
```

In [65]:
```
data.shape
```

Out[65]:
```
(7043, 21)
```

In [66]:
```
data.isna().sum()
```

Out[66]:
```
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 [67]:
```
#data['Totalcharges']=pd.to_numerical(data['TotalCharges'],errors='coerce')
data.PhoneService.unique()
```

Out[67]:
```
array(['No', 'Yes'], dtype=object)
```

In [68]:
```
eamingTV','StreamingMovies','OnlineSecurity','TechSupport','PhoneService','OnlineBackup','PaperlessBilling']
```

In [69]:
```
data
```

Out[69]:

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

7043 rows × 9 columns

In [70]:
```
data=data.fillna(data.median())
```

In [71]:
```
data
```

Out[71]:

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

7043 rows × 9 columns

In [72]:
```
data['Churn']=data['Churn'].map({'Yes':1,'No':0})
```

In [84]:
```
data=pd.get_dummies(data)
```

In [85]:
```
data
```

Out[85]:

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

7043 rows × 17 columns

In [86]:
```
y=data['Churn'] #predicted value removed from data frame
x=data.drop('Churn',axis=1)
```

In [87]:
```
y
```

Out[87]:
```
0       0
1       0
2       1
3       0
4       1
       ..
7038    0
7039    0
7040    0
7041    1
7042    0
Name: Churn, Length: 7043, dtype: int64
```

In [88]:
```
x
```

Out[88]:

	tenure	MonthlyCharges	TotalCharges	gender_Female	gender_Male	Partner_No	Partner_Yes	MultipleLines_No	MultipleLines_No phone service	MultipleL
0	1	29.85	29.85	1	0	0	1	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	0	1	
4	2	70.70	151.65	1	0	1	0	1	0	
...	...	...	...	...	...	...	...	...	...	
7038	24	84.80	1990.50	0	1	0	1	0	0	
7039	72	103.20	7362.90	1	0	0	1	0	0	
7040	11	29.60	346.45	1	0	0	1	0	1	
7041	4	74.40	306.60	0	1	0	1	0	0	
7042	66	105.65	6844.50	0	1	1	0	1	0	

7043 rows × 16 columns

In [89]:
```
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 [90]:
```
x_test.head(5)
```

Out[90]:

	tenure	MonthlyCharges	TotalCharges	gender_Female	gender_Male	Partner_No	Partner_Yes	MultipleLines_No	MultipleLines_No phone service	MultipleL
185	1	24.80	24.80	1	0	0	1	0	1	
2715	41	25.25	996.45	0	1	1	0	0	0	
3825	52	19.35	1031.70	1	0	0	1	1	0	
1807	1	76.35	76.35	1	0	1	0	1	0	
132	67	50.55	3260.10	0	1	1	0	1	0	

In [91]:
```
y_test.head(5)
```

Out[91]:
```
185     1
2715    0
3825    0
1807    1
132     0
Name: Churn, dtype: int64
```

In [92]:
```
x_train.head(5)
```

Out[92]:

	tenure	MonthlyCharges	TotalCharges	gender_Female	gender_Male	Partner_No	Partner_Yes	MultipleLines_No	MultipleLines_No phone service	MultipleL
298	40	74.55	3015.75	0	1	0	1	0	0	
3318	10	29.50	255.25	0	1	1	0	0	1	
5586	27	19.15	501.35	1	0	1	0	1	0	
6654	7	86.50	582.50	1	0	0	1	0	0	
5362	65	24.75	1715.10	0	1	0	1	0	0	

In [93]:
```
y_train.head(5)
```

Out[93]:
```
298     0
3318    1
5586    0
6654    1
5362    0
Name: Churn, dtype: int64
```

In [94]:
```
from sklearn.linear_model import LogisticRegression
classifier=LogisticRegression()
classifier.fit(x_train,y_train)
```

Out[94]:
```
▾ LogisticRegression

LogisticRegression()
```

In [95]:
```
y_pred=classifier.predict(x_test)
y_pred
```

Out[95]:
```
array([1, 0, 0, ..., 1, 1, 0])
```

In [96]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred)
```

Out[96]:
```
array([[1507,  190],
       [ 269,  359]])
```

In [97]:
```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

Out[97]:
```
0.8025806451612904
```

In [ ]:
```

```