```
In [1]: import pandas as pd
        import warnings
        warnings.filterwarnings('ignore')
        data=pd.read_csv("/home/placement/Downloads/TelecomCustomerChurn.csv")
```

```
In [2]: data['TotalCharges']=pd.to_numeric(data['TotalCharges'],errors='coerce')
```

```
In [3]: data.describe()
```

Out[3]:

	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 [4]:
```python
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        11
Churn                0
dtype: int64
```

In [5]:
```python
data1=data.fillna(data.median())
```

```
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        0
        Churn               0
        dtype: int64
```

In [7]: `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 [8]: list(data1)
```

```
Out[8]: ['customerID',
         'gender',
         'SeniorCitizen',
         'Partner',
         'Dependents',
         'tenure',
         'PhoneService',
         'MultipleLines',
         'InternetService',
         'OnlineSecurity',
         'OnlineBackup',
         'DeviceProtection',
         'TechSupport',
         'StreamingTV',
         'StreamingMovies',
         'Contract',
         'PaperlessBilling',
         'PaymentMethod',
         'MonthlyCharges',
         'TotalCharges',
         'Churn']
```

```
In [9]: data1.shape
```

```
Out[9]: (7043, 21)
```

```
In [10]: data2=data1.drop(['customerID','SeniorCitizen','OnlineSecurity','OnlineBackup','DeviceProtection','TechSupp
```

In [11]: `data2`

Out[11]:

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

7043 rows × 11 columns

In [12]: `data2['Churn']=data2['Churn'].map({'Yes':1,'No':0})`

In [13]: `data2`

Out[13]:

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

7043 rows × 11 columns

In [14]: `data3=pd.get_dummies(data2)`

In [15]: `data3`

Out[15]:

	tenure	MonthlyCharges	TotalCharges	Churn	gender_Female	gender_Male	Partner_No	Partner_Yes	Dependents_No	Dependents_Yes	...
0	1	29.85	29.85	0	1	0	0	1	1	0	...
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	1	0	...
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	1	...
7039	72	103.20	7362.90	0	1	0	0	1	0	1	...
7040	11	29.60	346.45	0	1	0	0	1	0	1	...
7041	4	74.40	306.60	1	0	1	0	1	1	0	...
7042	66	105.65	6844.50	0	0	1	1	0	1	0	...

7043 rows × 21 columns

In [16]: `data3.shape`

Out[16]: (7043, 21)

In [17]:
```python
y=data3['Churn']
x=data3.drop('Churn',axis=1)
```

In [18]:
```python
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]:
```python
from sklearn.linear_model import LogisticRegression
reg=LogisticRegression()
reg.fit(x_train,y_train)
```

Out[19]:
```
▾ LogisticRegression

LogisticRegression()
```

In [20]:
```python
y_pred=reg.predict(x_test)
```

In [22]:
```python
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred)
```

Out[22]:
```
array([[1514,  183],
       [ 272,  356]])
```

In [23]:
```python
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

Out[23]:
```
0.8043010752688172
```

In [ ]: