In [59]:
```
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
data=pd.read_csv("/home/placement/Downloads/TelecomCustomerChurn.csv")
```

In [61]:
```
data.dtypes
```

Out[61]:
```
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         object
Churn                object
dtype: object
```

In [63]:
```
data['TotalCharges'] = pd.to_numeric(data['TotalCharges'], errors='coerce')
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]:
```
data.describe()
```

Out[64]:

	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 [65]:
```
data.isna().sum()
```

Out[65]:
```
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 [66]:
```
data1=data.fillna(data.median())
```

In [67]:
```
data1.isna().sum()
```

Out[67]:
```
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 [68]:
```
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 [69]:
```
list(data1)
```

Out[69]:
```
['customerID',
 'gender',
 'SeniorCitizen',
 'Partner',
 'Dependents',
 'tenure',
 'PhoneService',
 'MultipleLines',
 'InternetService',
 'OnlineSecurity',
 'OnlineBackup',
 'DeviceProtection',
 'TechSupport',
 'StreamingTV',
 'StreamingMovies',
 'Contract',
 'PaperlessBilling',
 'PaymentMethod',
 'MonthlyCharges',
 'TotalCharges',
 'Churn']
```

In [70]:
```
data1.shape
```

Out[70]:
```
(7043, 21)
```

In [72]:
```
data2=data1.drop(['customerID','Dependents','StreamingTV','StreamingMovies','Partner','OnlineSecurity','Devi
data2
```

Out[72]:

	gender	SeniorCitizen	tenure	PhoneService	MultipleLines	InternetService	OnlineBackup	TechSupport	Contract	MonthlyCharges	TotalCh
0	Female	0	1	No	No phone service	DSL	Yes	No	Month-to-month	29.85	
1	Male	0	34	Yes	No	DSL	No	No	One year	56.95	18
2	Male	0	2	Yes	No	DSL	Yes	No	Month-to-month	53.85	1
3	Male	0	45	No	No phone service	DSL	No	Yes	One year	42.30	18
4	Female	0	2	Yes	No	Fiber optic	No	No	Month-to-month	70.70	1
...	...	...	...	...	...	...	...	...	...	...	
7038	Male	0	24	Yes	Yes	DSL	No	Yes	One year	84.80	19
7039	Female	0	72	Yes	Yes	Fiber optic	Yes	No	One year	103.20	73
7040	Female	0	11	No	No phone service	DSL	No	No	Month-to-month	29.60	3
7041	Male	1	4	Yes	Yes	Fiber optic	No	No	Month-to-month	74.40	3
7042	Male	0	66	Yes	No	Fiber optic	No	Yes	Two year	105.65	68

7043 rows × 12 columns

In [73]:
```
data2['Churn']=data2['Churn'].map({'Yes':1,'No':0})
```

In [74]:
```
data2
```

Out[74]:

	gender	SeniorCitizen	tenure	PhoneService	MultipleLines	InternetService	OnlineBackup	TechSupport	Contract	MonthlyCharges	TotalCh
0	Female	0	1	No	No phone service	DSL	Yes	No	Month-to-month	29.85	
1	Male	0	34	Yes	No	DSL	No	No	One year	56.95	18
2	Male	0	2	Yes	No	DSL	Yes	No	Month-to-month	53.85	1
3	Male	0	45	No	No phone service	DSL	No	Yes	One year	42.30	18
4	Female	0	2	Yes	No	Fiber optic	No	No	Month-to-month	70.70	1
...	...	...	...	...	...	...	...	...	...	...	
7038	Male	0	24	Yes	Yes	DSL	No	Yes	One year	84.80	19
7039	Female	0	72	Yes	Yes	Fiber optic	Yes	No	One year	103.20	73
7040	Female	0	11	No	No phone service	DSL	No	No	Month-to-month	29.60	3
7041	Male	1	4	Yes	Yes	Fiber optic	No	No	Month-to-month	74.40	3
7042	Male	0	66	Yes	No	Fiber optic	No	Yes	Two year	105.65	68

7043 rows × 12 columns

In [75]:
```
data3=pd.get_dummies(data2)
```

In [76]:
```
data3
```

Out[76]:

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

7043 rows × 24 columns

In [77]:
```
data3.shape
```

Out[77]: (7043, 24)

In [78]:
```
y=data3['Churn']
x=data3.drop('Churn',axis=1)
```

In [79]:
```
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 [80]:
```
from sklearn.linear_model import LogisticRegression
classifier= LogisticRegression()
classifier.fit(x_train,y_train)
```

Out[80]:
```
LogisticRegression()
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [83]:
```
y_pred=classifier.predict(x_test)
y_pred
```

Out[83]:
```
array([1, 0, 0, ..., 1, 0, 0])
```

In [84]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred)
```

Out[84]:
```
array([[1526,  171],
       [ 276,  352]])
```

In [85]:
```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

Out[85]:
```
0.807741935483871
```

In [ ]:
```

```