In [1]:
```
1 import pandas as pd
2 import warnings
3 warnings.filterwarnings("ignore")
```

In [2]:
```
1 data=pd.read_csv("/home/placement/Downloads/TelecomCustomerChurn.csv")
```

In [3]:
```
1 data
```

Out[3]:

	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 [4]:
```
1 data.describe()
```

Out[4]:

	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 [5]:
```
1 data.isna().sum()
```

Out[5]:
```
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 [6]:
```
1 data['TotalCharges']=pd.to_numeric(data['TotalCharges'],errors='coerce')
```

In [7]:
```
1 data.dtypes
```

Out[7]:
```
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
```

In [8]:
```
data=data.drop(['OnlineSecurity','OnlineBackup','DeviceProtection','PaymentMethod','PhoneService','Depe
```

In [9]:
```
1 data
```

Out[9]:

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

In [10]:
```
1 data=data.fillna(data.median())
2 data
```

Out[10]:

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

7043 rows × 9 columns

In [11]:
```
1 data['Churn']=data['Churn'].map({'Yes':1,'No':0})
```

In [12]:
```
data=pd.get_dummies(data)
data
```

Out[12]:

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

In [18]:
```
y=data['Churn']
x=data.drop('Churn',axis=1)
```

In [19]:
```
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 [20]:
```
1 x_test
```

Out[20]:

	tenure	MonthlyCharges	TotalCharges	gender_Female	gender_Male	MultipleLines_No	MultipleLines_No phone service	MultipleLines_Yes	InternetService_
185	1	24.80	24.80	1	0	0	1	0	
2715	41	25.25	996.45	0	1	0	0	1	
3825	52	19.35	1031.70	1	0	1	0	0	
1807	1	76.35	76.35	1	0	1	0	0	
132	67	50.55	3260.10	0	1	1	0	0	
...	...	...	...	...	...	...	...	...	
4147	71	24.85	1901.00	0	1	0	0	1	
3542	29	55.35	1636.95	0	1	0	1	0	
3759	7	89.35	631.85	0	1	0	0	1	
1114	32	98.85	3145.90	0	1	0	0	1	
4958	59	94.75	5597.65	1	0	0	0	1	

2325 rows × 17 columns

In [16]:
```
1 x_train
```

Out[16]:

	tenure	MonthlyCharges	TotalCharges	gender_Female	gender_Male	MultipleLines_No	MultipleLines_No phone service	MultipleLines_Yes	InternetService_
298	40	74.55	3015.75	0	1	0	0	1	
3318	10	29.50	255.25	0	1	0	1	0	
5586	27	19.15	501.35	1	0	1	0	0	
6654	7	86.50	582.50	1	0	0	0	1	
5362	65	24.75	1715.10	0	1	0	0	1	
...	...	...	...	...	...	...	...	...	...
3772	1	95.00	95.00	0	1	1	0	0	
5191	23	91.10	2198.30	1	0	0	0	1	
5226	12	21.15	306.05	0	1	1	0	0	
5390	12	99.45	1200.15	0	1	0	0	1	
860	26	19.80	457.30	0	1	1	0	0	

4718 rows × 17 columns

In [21]:
```
1 from sklearn.linear_model import LogisticRegression
2 classifier=LogisticRegression()
3 classifier.fit(x_train,y_train)
```

Out[21]:
```
▾ LogisticRegression

LogisticRegression()
```

In [23]:
```
1 y_pred=classifier.predict(x_test)
```

In [24]:
```
1 y_pred
```

Out[24]:
```
array([1, 0, 0, ..., 1, 1, 0])
```

In [25]:
```
1 from sklearn.metrics import confusion_matrix
2 confusion_matrix(y_test,y_pred)
```

Out[25]:
```
array([[1519,  178],
       [ 273,  355]])
```

In [26]:
```
1 from sklearn.metrics import accuracy_score
2 accuracy_score(y_test,y_pred)
```

Out[26]:
```
0.8060215053763441
```

In [ ]:
```
1
```