

```
In [1]: import pandas as pd
```

```
In [2]: import warnings  
warnings.filterwarnings('ignore')
```

```
In [3]: data=pd.read_csv("/home/placement/Downloads/TelecomCustomerChurn.csv")
```

```
In [4]: 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]: 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]: data1=data.fillna(data.median())
```

```
In [7]: data1.isna().sum()
```

```
Out[7]: 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 [8]: 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   object
20  Churn                 7043 non-null   object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

```
In [9]: list(data1)
```

```
Out[9]: ['customerID',  
         'gender',  
         'SeniorCitizen',  
         'Partner',  
         'Dependents',  
         'tenure',  
         'PhoneService',  
         'MultipleLines',  
         'InternetService',  
         'OnlineSecurity',  
         'OnlineBackup',  
         'DeviceProtection',  
         'TechSupport',  
         'StreamingTV',  
         'StreamingMovies',  
         'Contract',  
         'PaperlessBilling',  
         'PaymentMethod',  
         'MonthlyCharges',  
         'TotalCharges']
```

```
In [10]: data1.shape
```

```
Out[10]: (7043, 21)
```

```
In [11]: data2=data.drop(['customerID', 'SeniorCitizen', 'Dependents', 'PhoneService', 'InternetService', 'OnlineBackup', 'TotalCharges'])
```

In [12]: data2

Out[12]:

	gender	Partner	tenure	MultipleLines	OnlineSecurity	DeviceProtection	StreamingTV	Contract	PaymentMethod	TotalCharges	Churn
0	Female	Yes	1	No phone service	No	No	No	Month-to-month	Electronic check	29.85	No
1	Male	No	34	No	Yes	Yes	No	One year	Mailed check	1889.5	No
2	Male	No	2	No	Yes	No	No	Month-to-month	Mailed check	108.15	Yes
3	Male	No	45	No phone service	Yes	Yes	No	One year	Bank transfer (automatic)	1840.75	No
4	Female	No	2	No	No	No	No	Month-to-month	Electronic check	151.65	Yes
...	...	...	...	...	...	...	...	...	...	...	...
7038	Male	Yes	24	Yes	Yes	Yes	Yes	One year	Mailed check	1990.5	No
7039	Female	Yes	72	Yes	No	Yes	Yes	One year	Credit card (automatic)	7362.9	No
7040	Female	Yes	11	No phone service	Yes	No	No	Month-to-month	Electronic check	346.45	No
7041	Male	Yes	4	Yes	No	No	No	Month-to-month	Mailed check	306.6	Yes
7042	Male	No	66	No	Yes	Yes	Yes	Two year	Bank transfer (automatic)	6844.5	No

7043 rows × 11 columns

In [13]: data2['Churn']=data2['Churn'].map({'Yes':1, 'No':0})

In [14]: data2

Out[14]:

	gender	Partner	tenure	MultipleLines	OnlineSecurity	DeviceProtection	StreamingTV	Contract	PaymentMethod	TotalCharges	Churn
0	Female	Yes	1	No phone service	No	No	No	Month-to-month	Electronic check	29.85	0
1	Male	No	34	No	Yes	Yes	No	One year	Mailed check	1889.5	0
2	Male	No	2	No	Yes	No	No	Month-to-month	Mailed check	108.15	1
3	Male	No	45	No phone service	Yes	Yes	No	One year	Bank transfer (automatic)	1840.75	0
4	Female	No	2	No	No	No	No	Month-to-month	Electronic check	151.65	1
...	...	...	...	...	...	...	...	...	...	...	...
7038	Male	Yes	24	Yes	Yes	Yes	Yes	One year	Mailed check	1990.5	0
7039	Female	Yes	72	Yes	No	Yes	Yes	One year	Credit card (automatic)	7362.9	0
7040	Female	Yes	11	No phone service	Yes	No	No	Month-to-month	Electronic check	346.45	0
7041	Male	Yes	4	Yes	No	No	No	Month-to-month	Mailed check	306.6	1
7042	Male	No	66	No	Yes	Yes	Yes	Two year	Bank transfer (automatic)	6844.5	0

7043 rows × 11 columns

In [15]: data3=pd.get\_dummies(data2)

In [16]: data3

Out[16]:

Charges_996.85	TotalCharges_996.95	TotalCharges_997.65	TotalCharges_997.75	TotalCharges_998.1	TotalCharges_999.45	TotalCharges_999.8	TotalCh
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

In [17]: data3.shape

Out[17]: (7043, 6556)

In [18]: y=data3['Churn']  
x=data3.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]: from sklearn.linear_model import LogisticRegression
reg=LogisticRegression()
reg.fit(x_train,y_train)
```

```
Out[20]: ▾ LogisticRegression
LogisticRegression()
```

```
In [21]: y_pred=reg.predict(x_test)
```

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

```
Out[23]: array([[1509, 188],
               [ 299, 329]])
```

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

```
Out[26]: 0.7905376344086021
```

```
In [ ]:
```