

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

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

```
In [3]: data=pd.read_csv("/home/placement/Desktop/BhanuSiva4K8/TelecomCustomerChurn.csv")
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-JJAZL	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]: data['TotalCharges'] = pd.to_numeric(data['TotalCharges'], errors='coerce')
```

```
In [5]: data.describe()
```

Out[5]:

	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 [6]: data.shape
```

Out[6]: (7043, 21)

```
In [7]: list(data)
```

```
Out[7]: ['customerID',  
        'gender',  
        'SeniorCitizen',  
        'Partner',  
        'Dependents',  
        'tenure',  
        'PhoneService',  
        'MultipleLines',  
        'InternetService',  
        'OnlineSecurity',  
        'OnlineBackup',  
        'DeviceProtection',  
        'TechSupport',  
        'StreamingTV',  
        'StreamingMovies',  
        'Contract',  
        'PaperlessBilling',  
        'PaymentMethod',  
        'MonthlyCharges',  
        'TotalCharges',  
        'Churn']
```

In [8]: `data.head(5)`

Out[8]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtec
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	...	

5 rows × 21 columns



```
In [9]: data.isna().sum()
```

```
Out[9]: 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 [10]: data1=data.fillna(data.median())
```

```
In [11]: data1.isna().sum()
```

```
Out[11]: 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 [12]: data2=data1.drop(['customerID','SeniorCitizen','Partner','Dependents','PhoneService','OnlineBackup','PaymentMethod'])
data2
```

Out[12]:

	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 [13]: data3=data2.fillna(data2.median())
```

```
In [14]: data3.isna().sum()
```

```
Out[14]: gender          0  
         tenure         0  
         MultipleLines  0  
         InternetService 0  
         TechSupport    0  
         Contract       0  
         MonthlyCharges 0  
         TotalCharges   0  
         Churn          0  
         dtype: int64
```

```
In [15]: data3['Churn']=data3['Churn'].map({'Yes':1,'No':0})
```

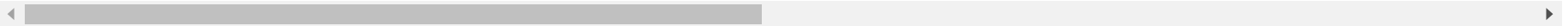


```
In [16]: data4=pd.get_dummies(data3)
data4
```

Out[16]:

	tenure	MonthlyCharges	TotalCharges	Churn	gender_Female	gender_Male	MultipleLines_No	MultipleLines_No phone service	MultipleLines_Yes	Internet
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	
7042	66	105.65	6844.50	0	0	1	1	0	0	

7043 rows × 18 columns



```
In [19]: y=data4['Churn']
x=data4.drop('Churn',axis=1)
```

```
In [20]: #data['TotalChargesn']=data(['TotalCharges'].apply(pd.to_numeric)
data['TotalCharges']= pd.to_numeric(data['TotalCharges'],errors='coerce')
```

## Logistic Regression

```
In [21]: 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 [22]: from sklearn.linear_model import LogisticRegression
reg=LogisticRegression()#creating object of LogisticRegression
reg.fit(x_train,y_train)#training and fitting LR object using training data
```

```
Out[22]: ▾ LogisticRegression
LogisticRegression()
```

```
In [24]: y_pred=reg.predict(x_test)
y_pred
```

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

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

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

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

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

```
In [ ]:
```