

```
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.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 [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.isna().sum()
```

```
Out[8]: 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 [9]: data.head(5)
```

```
Out[9]:
```

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

In [26]: data.dtypes

```
Out[26]: 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 [27]: `chod', 'DeviceProtection', 'StreamingTV', 'StreamingMovies', 'PaperlessBilling', 'OnlineSecurity', 'OnlineBackup']`,

In [28]: data

Out[28]:

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

Out[29]:

	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 [30]: data1.isna().sum()
```

```
Out[30]: gender          0
tenure          0
MultipleLines     0
InternetService   0
TechSupport       0
Contract          0
MonthlyCharges    0
TotalCharges      0
Churn             0
dtype: int64
```



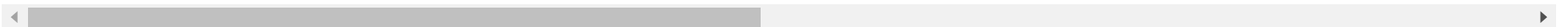
```
In [31]: data1['Churn']=data1['Churn'].map({'Yes':1, 'No':0})
```

```
In [32]: data2=pd.get_dummies(data1)
data2
```

Out[32]:

	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 [33]: y=data2['Churn']
x=data2.drop('Churn',axis=1)
```

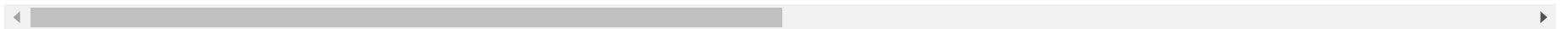
```
In [34]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=42)
```

In [35]: x_test

Out[35]:

	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

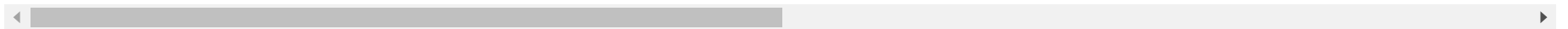


```
In [36]: x_train
```

```
Out[36]:
```

	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 [37]: y_train
```

```
Out[37]: 298      0
          3318    1
          5586    0
          6654    1
          5362    0
          ..
          3772    1
          5191    0
          5226    0
          5390    1
          860     0
          Name: Churn, Length: 4718, dtype: int64
```

logisticregression

```
In [38]: from sklearn.linear_model import LogisticRegression
          classifier=LogisticRegression()
          classifier.fit(x_train,y_train)
```

```
Out[38]: ▾ LogisticRegression
          LogisticRegression()
```

```
In [39]: y_pred=classifier.predict(x_test)
```

```
In [40]: y_pred
```

```
Out[40]: array([1, 0, 0, ..., 1, 1, 0])
```

```
In [41]: from sklearn.metrics import confusion_matrix
          confusion_matrix(y_test,y_pred)
```

```
Out[41]: array([[1519, 178],
                [ 273, 355]])
```

```
In [42]: from sklearn.metrics import accuracy_score  
accuracy_score(y_test,y_pred)
```

```
Out[42]: 0.8060215053763441
```

```
In [ ]:
```