

```
In [264]: import pandas as pd  
import numpy as np  
import warnings  
warnings.filterwarnings("ignore")
```

```
In [265]: data=pd.read_csv("/home/placement/Downloads/TelecomCustomerChurn.csv")
```

```
In [ ]:
```

```
In [ ]:
```

```
In [266]: 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 [267]: list(data)
```

```
Out[267]: ['customerID',  
          'gender',  
          'SeniorCitizen',  
          'Partner',  
          'Dependents',  
          'tenure',  
          'PhoneService',  
          'MultipleLines',  
          'InternetService',  
          'OnlineSecurity',  
          'OnlineBackup',  
          'DeviceProtection',  
          'TechSupport',  
          'StreamingTV',  
          'StreamingMovies',  
          'Contract',  
          'PaperlessBilling',  
          'PaymentMethod',  
          'MonthlyCharges',  
          'TotalCharges',  
          'Churn']
```

In [268]: data.head()

Out[268]:

	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 [269]: data.describe()

Out[269]:

	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

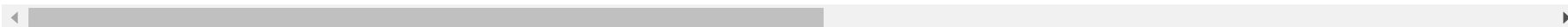
In [270]: data=data.drop("customerID",axis=1)

In [271]: data

Out[271]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProte
0	Female	0	Yes	No	1	No	No phone service	DSL	No	Yes	
1	Male	0	No	No	34	Yes	No	DSL	Yes	No	
2	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	
3	Male	0	No	No	45	No	No phone service	DSL	Yes	No	
4	Female	0	No	No	2	Yes	No	Fiber optic	No	No	
...	...	...	...	...	...	...	...	...	...	...	
7038	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes	No	
7039	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No	Yes	
7040	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes	No	
7041	Male	1	Yes	No	4	Yes	Yes	Fiber optic	No	No	
7042	Male	0	No	No	66	Yes	No	Fiber optic	Yes	No	

7043 rows × 20 columns



In [ ]:

In [272]: data1=data.drop(["PaperlessBilling","PaymentMethod","Dependents","SeniorCitizen","Partner","gender"],axis=1)

```
In [273]: data1["Churn"]=data1["Churn"].map({"Yes":1,"No":0})
data1
```

```
Out[273]:
```

	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	StreamingTV	Streaming
0	1	No	No phone service	DSL	No	Yes	No	No	No	
1	34	Yes	No	DSL	Yes	No	Yes	No	No	
2	2	Yes	No	DSL	Yes	Yes	No	No	No	
3	45	No	No phone service	DSL	Yes	No	Yes	Yes	No	
4	2	Yes	No	Fiber optic	No	No	No	No	No	
...	...	...	...	...	...	...	...	...	...	...
7038	24	Yes	Yes	DSL	Yes	No	Yes	Yes	Yes	
7039	72	Yes	Yes	Fiber optic	No	Yes	Yes	No	Yes	
7040	11	No	No phone service	DSL	Yes	No	No	No	No	
7041	4	Yes	Yes	Fiber optic	No	No	No	No	No	
7042	66	Yes	No	Fiber optic	Yes	No	Yes	Yes	Yes	

7043 rows × 14 columns

```
In [276]: data1['TotalCharges']=pd.to_numeric(data1['TotalCharges'],errors='coerce')
```

```
In [277]: data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   tenure                 7043 non-null  int64  
1   PhoneService           7043 non-null  object  
2   MultipleLines          7043 non-null  object  
3   InternetService        7043 non-null  object  
4   OnlineSecurity          7043 non-null  object  
5   OnlineBackup           7043 non-null  object  
6   DeviceProtection       7043 non-null  object  
7   TechSupport            7043 non-null  object  
8   StreamingTV            7043 non-null  object  
9   StreamingMovies        7043 non-null  object  
10  Contract               7043 non-null  object  
11  MonthlyCharges         7043 non-null  float64 
12  TotalCharges           7032 non-null  float64 
13  Churn                  7043 non-null  int64  
dtypes: float64(2), int64(2), object(10)
memory usage: 770.5+ KB
```

```
In [278]: data1=data1.fillna(data1.median())
```

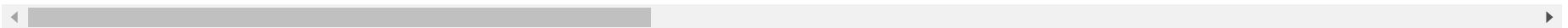
```
In [279]: data1=pd.get_dummies(data1)
```

In [280]: data1

Out[280]:

	tenure	MonthlyCharges	TotalCharges	Churn	PhoneService_No	PhoneService_Yes	MultipleLines_No	MultipleLines_No phone service	MultipleLines_Yes
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	1	0	0	1	0
4	2	70.70	151.65	1	0	1	1	0	0
...	...	...	...	...	...	...	...	...	...
7038	24	84.80	1990.50	0	0	1	0	0	1
7039	72	103.20	7362.90	0	0	1	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 × 33 columns



In [281]: `y=data1['Churn']`  
`x=data1.drop('Churn',axis=1)`

In [282]: `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 [ ]:



```
In [283]: from sklearn.linear_model import LogisticRegression
reg=LogisticRegression()
reg.fit(x_train,y_train)
#importing logistic regression
```

Out[283]: 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 [ ]:

```
In [284]: y_pred=reg.predict(x_test)
```

```
In [285]: from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred)# CONFUSIO MATRIX OF TRUE POSITIVE&NEGATIVE , FASLE POSITIVE & NEGATIVE
```

Out[285]: array([[1516, 181],  
[ 265, 363]])

```
In [286]: from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)#EFFICENCY OF THE CONFUSION MATRIX
```

Out[286]: 0.8081720430107527

In [ ]:

In [ ]:

In [ ]:

