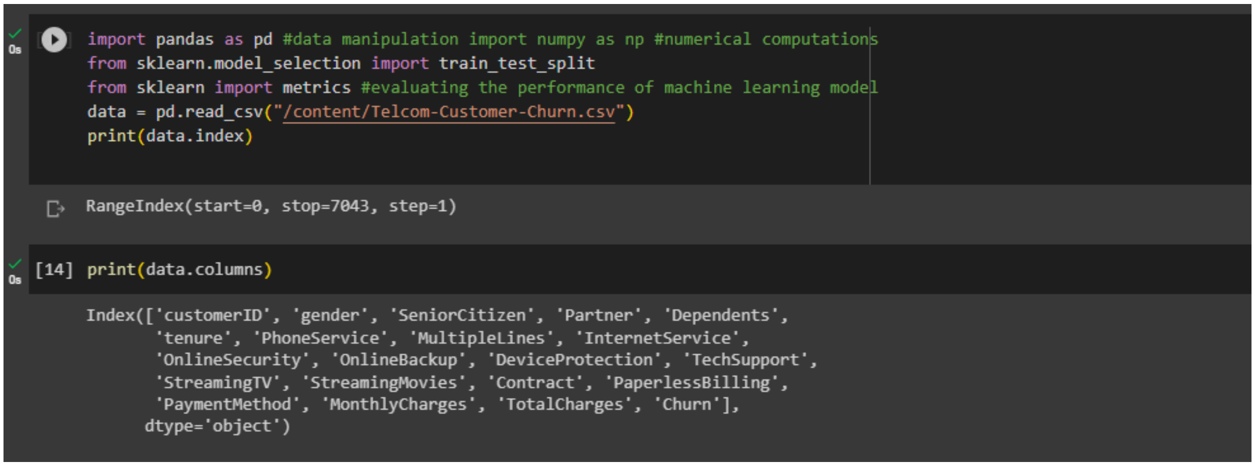
Assignment No. 08

Title – Data Cleaning and Preparation.

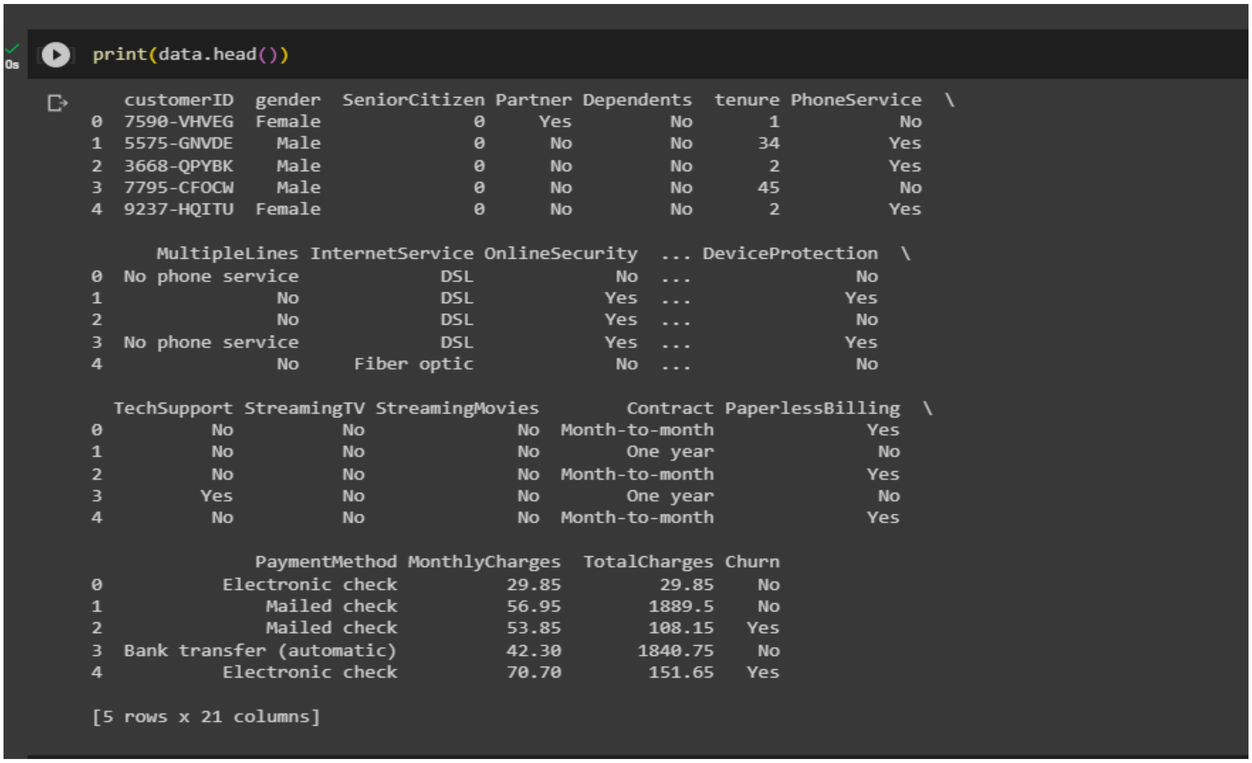


```
import pandas as pd #data manipulation import numpy as np #numerical computations
from sklearn.model_selection import train_test_split
from sklearn import metrics #evaluating the performance of machine learning model
data = pd.read_csv("/content/Telcom-Customer-Churn.csv")
print(data.index)
```

```
RangeIndex(start=0, stop=7043, step=1)
```

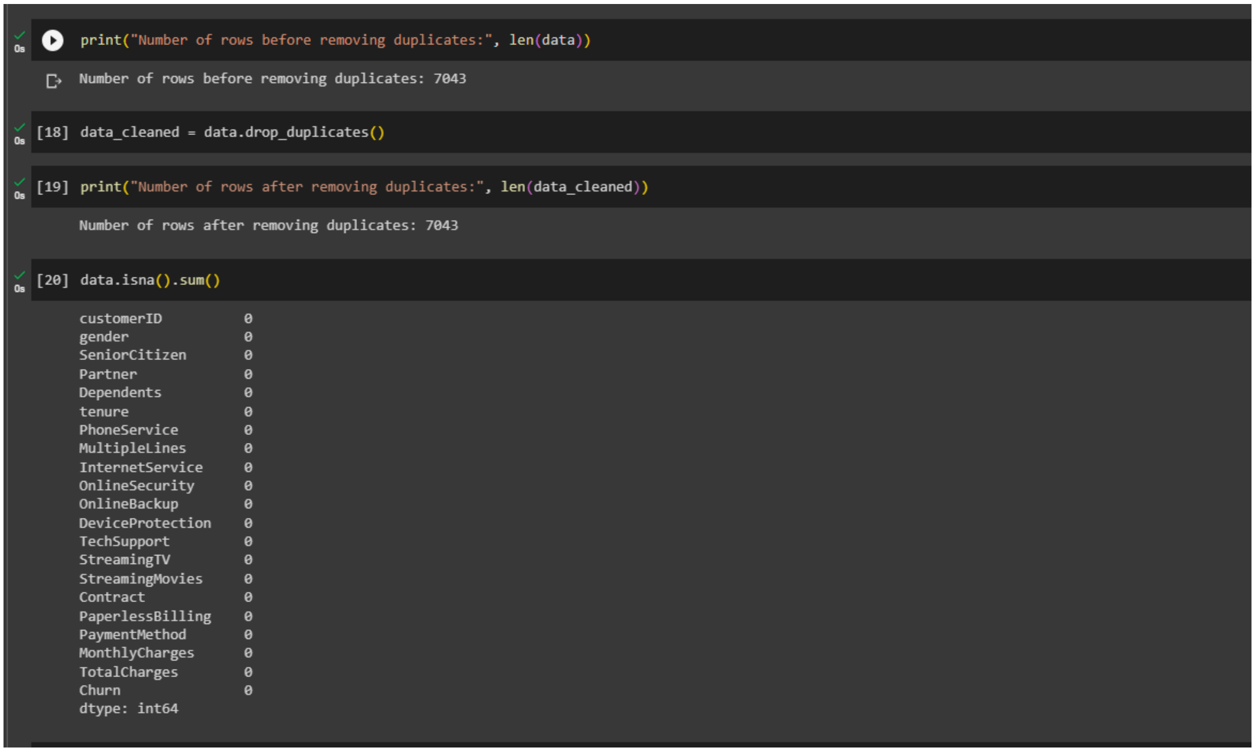
```
[14] print(data.columns)
```

```
Index(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport',
       'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling',
       'PaymentMethod', 'MonthlyCharges', 'TotalCharges', 'Churn'],
      dtype='object')
```

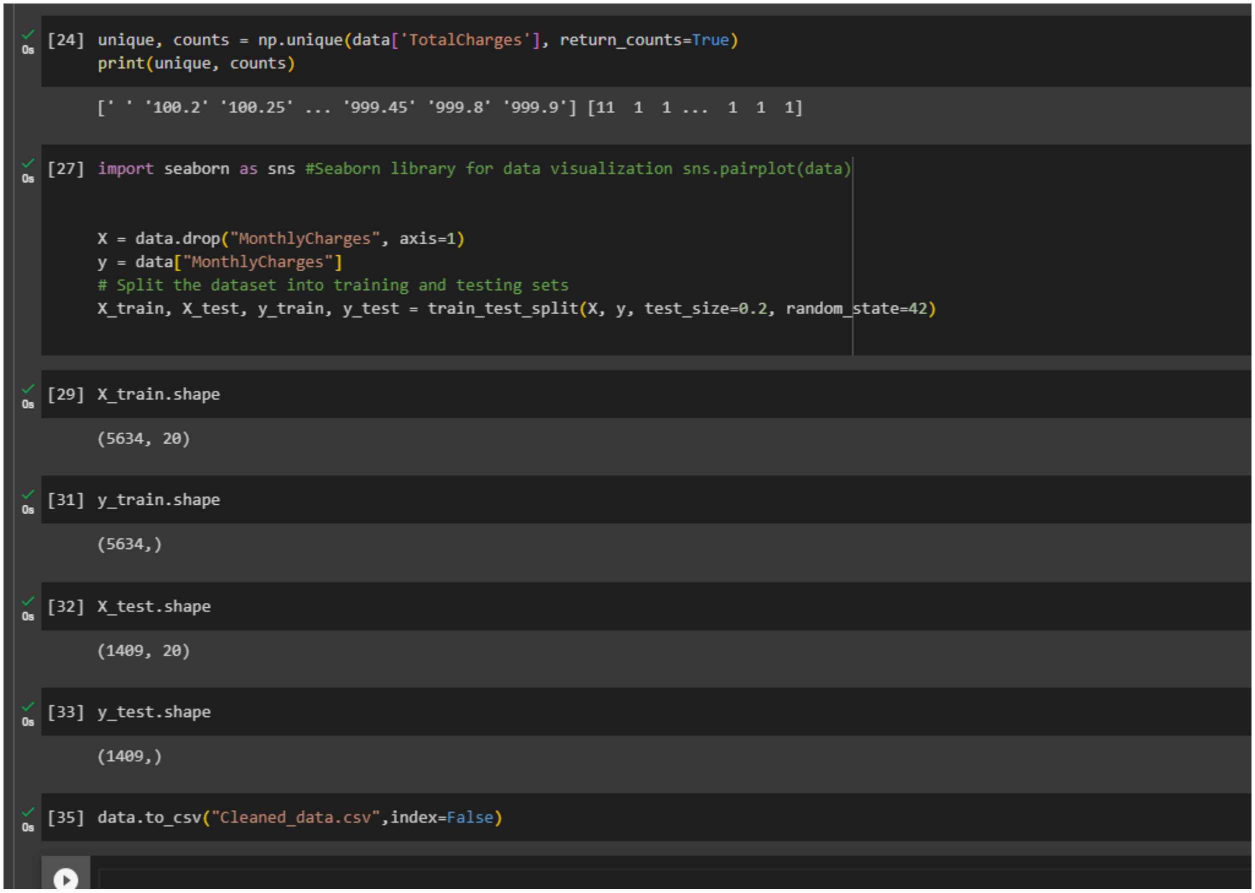


```
print(data.head())
```

```
   customerID  gender  SeniorCitizen Partner Dependents  tenure PhoneService  \
0  7590-VHVEG  Female              0     Yes         No       1           No
1  5575-GNVDE    Male              0      No         No      34          Yes
2  3668-QPYBK    Male              0      No         No       2          Yes
3  7795-CFOCW    Male              0      No         No      45           No
4  9237-HQITU  Female              0      No         No       2          Yes

      MultipleLines InternetService OnlineSecurity  ... DeviceProtection  \
0  No phone service             DSL             No  ...               No
1                No             DSL            Yes  ...              Yes
2                No             DSL            Yes  ...               No
3  No phone service             DSL            Yes  ...              Yes
4                No     Fiber optic             No  ...               No

  TechSupport StreamingTV StreamingMovies        Contract PaperlessBilling  \
0          No          No              No  Month-to-month              Yes
1          No          No              No        One year               No
2          No          No              No  Month-to-month              Yes
3         Yes          No              No        One year               No
4          No          No              No  Month-to-month              Yes

               PaymentMethod MonthlyCharges  TotalCharges Churn
0           Electronic check          29.85         29.85    No
1               Mailed check          56.95        1889.5    No
2               Mailed check          53.85        108.15   Yes
3  Bank transfer (automatic)          42.30       1840.75    No
4           Electronic check          70.70        151.65   Yes

[5 rows x 21 columns]
```



```
print("Number of rows before removing duplicates:", len(data))

Number of rows before removing duplicates: 7043

[18] data_cleaned = data.drop_duplicates()

[19] print("Number of rows after removing duplicates:", len(data_cleaned))

Number of rows after removing duplicates: 7043

[20] data.isna().sum()

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
```



```
[24] unique, counts = np.unique(data['TotalCharges'], return_counts=True)
     print(unique, counts)

     [' ' '100.2' '100.25' ... '999.45' '999.8' '999.9'] [11  1  1 ...  1  1  1]

[27] import seaborn as sns #Seaborn library for data visualization sns.pairplot(data)


     X = data.drop("MonthlyCharges", axis=1)
     y = data["MonthlyCharges"]
     # Split the dataset into training and testing sets
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

[29] X_train.shape

     (5634, 20)

[31] y_train.shape

     (5634,)

[32] X_test.shape

     (1409, 20)

[33] y_test.shape

     (1409,)

[35] data.to_csv("Cleaned_data.csv",index=False)
```