**Assignment -2**

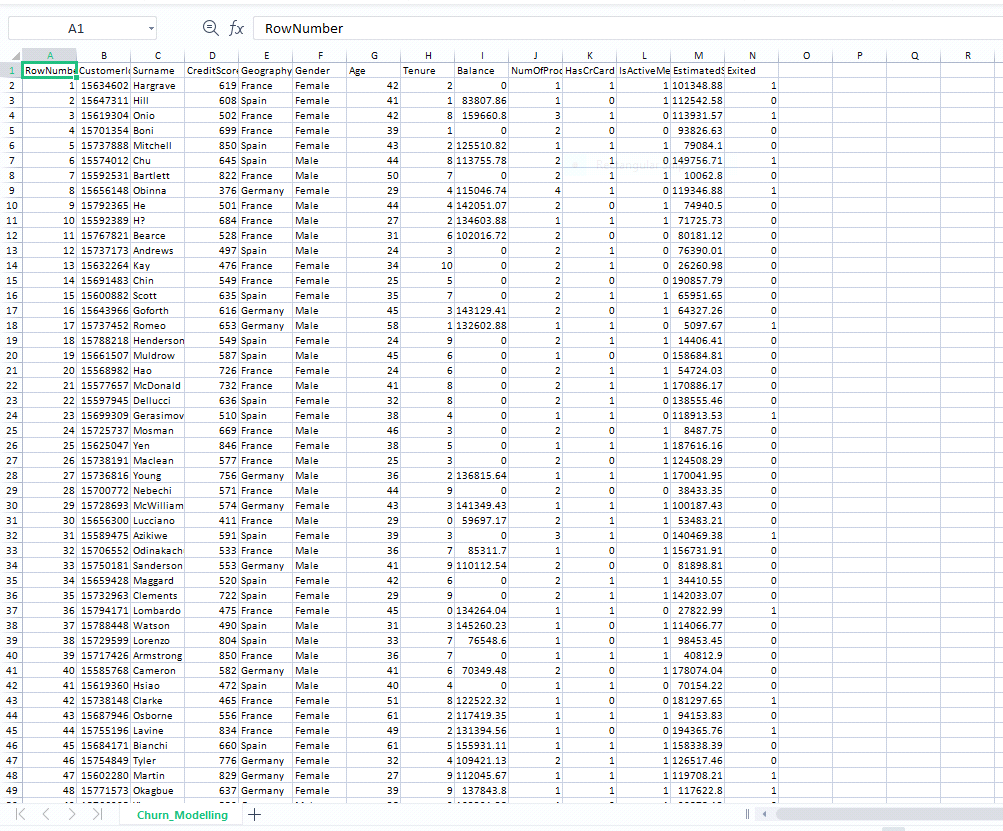
**DATA VISUALISATION AND PRE-PROCESSING**

|  |  |
| --- | --- |
| **ASSIGNMENT DATE** | 28 September 2022 |
| **STUDENT NAME** | THANGA PANDI.P |
| **STUDENT ROLLNUMBER** | 612419104022 |
| **MAXIMUM MARKS** | 2 Marks |

**Question-1:**

Download the Dataset

**SOLUTION:**



**Question-2:**

Loading dataset

**SOLUTION:**

import pandas as pd

import seaborn as sns

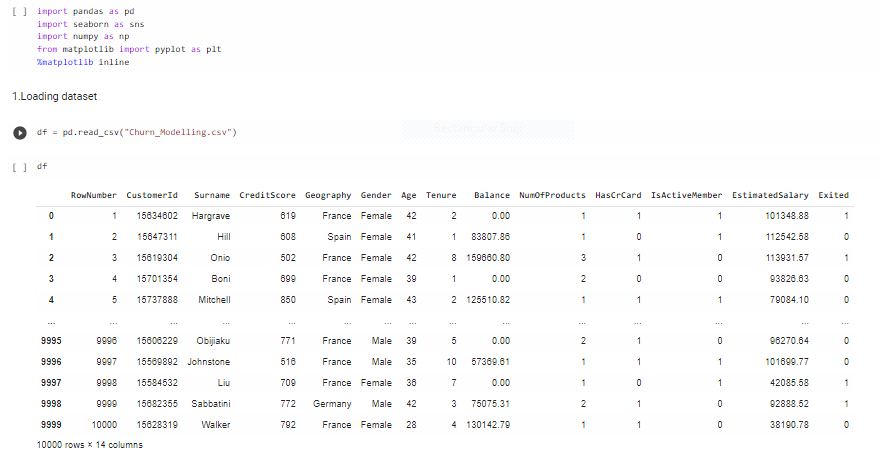
import numpy as np

from matplotlib import pyplot as plt

%matplotlib inline

df = pd.read\_csv("Churn\_Modelling.csv")

df



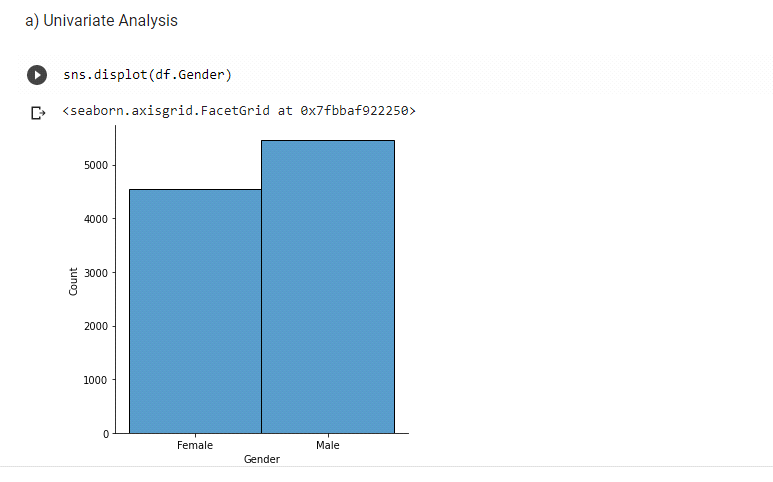
**Question-3:**

Visualizations

a) Univariate Analysis

**SOLUTION:**

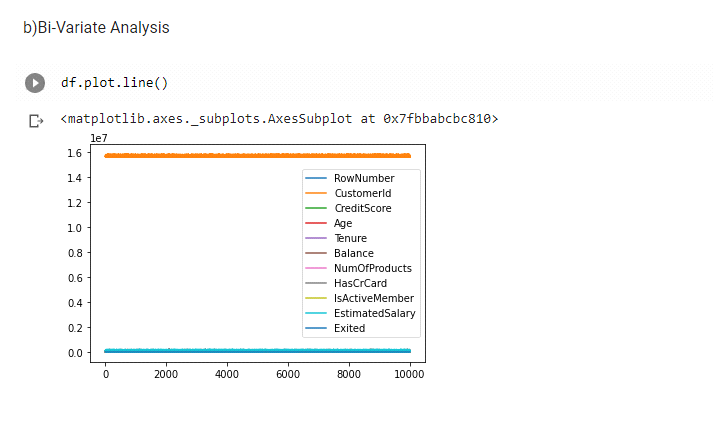
sns.displot(df.Gender)



b) Bi-Variate Analysis

**SOLUTION:**

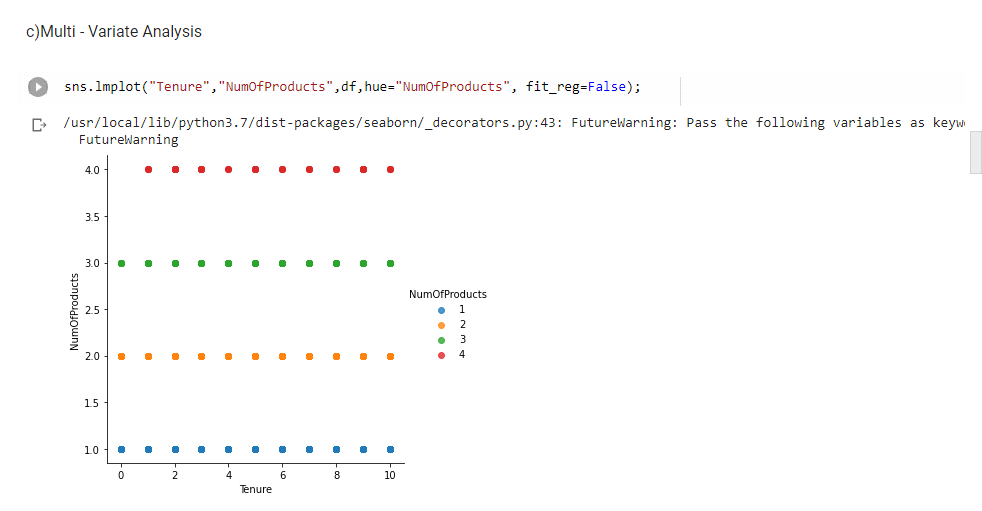
df.plot.line()



c) Multi - Variate Analysis

**SOLUTION:**

sns.lmplot("Tenure","NumOfProducts",df,hue="NumOfProducts",fit\_reg=False);

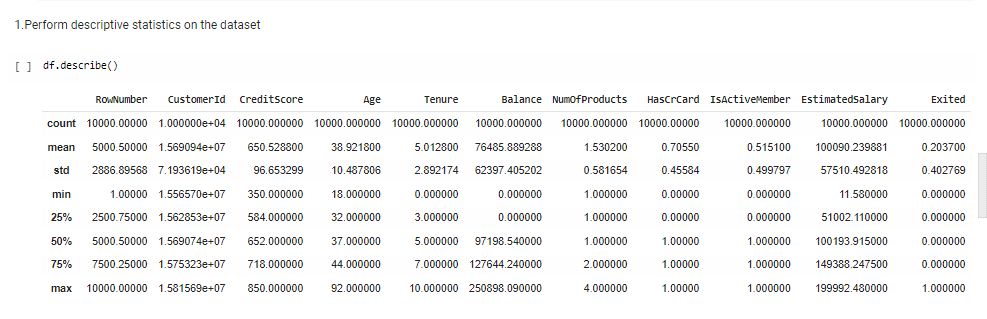


**Question-4:**

Perform descriptive statistics on the dataset.

**SOLUTION:**

df.describe()



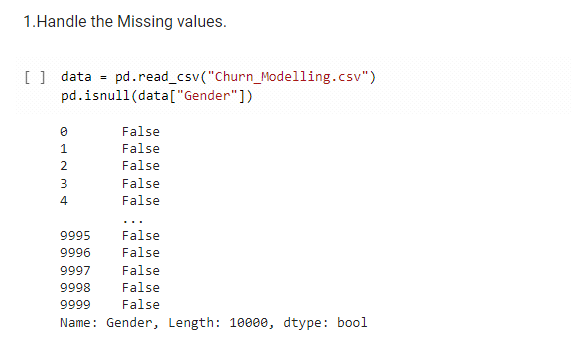
**Question-5:**

Handle the Missing values.

**SOLUTION:**

data = pd.read\_csv("Churn\_Modelling.csv")

pd.isnull(data["Gender"])

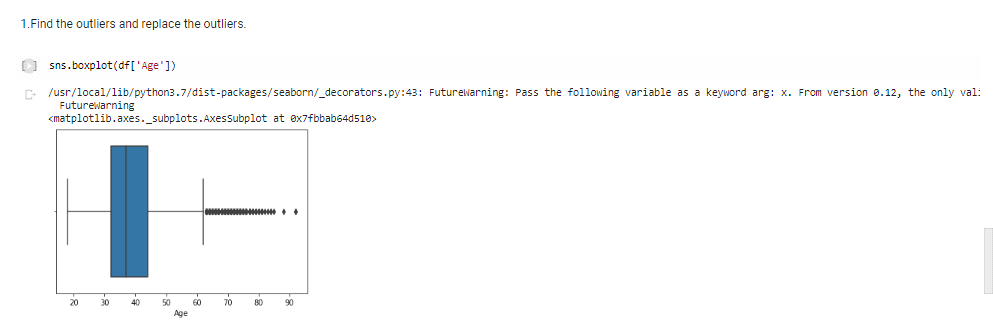


**Question-6:**

Find the outliers and replace the outliers.

**SOLUTION:**

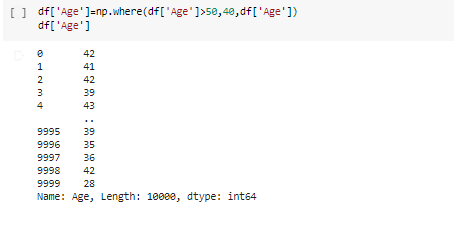
sns.boxplot(df['Age'])



**SOLUTION:**

df['Age']=np.where(df['Age']>50,40,df['Age'])

df['Age']

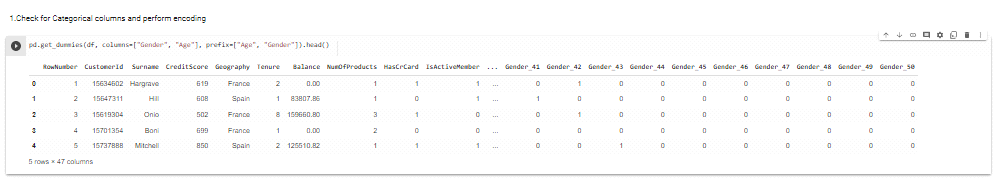


**Question-7:**

Check for Categorical columns and perform encoding.

**SOLUTION:**

pd.get\_dummies(df, columns=["Gender", "Age"], prefix=["Age", "Gender"]).hea



**Question-8:**

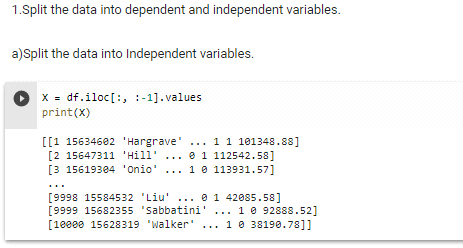
. Split the data into dependent and independent variables.

(a) Split the data into Independent variables.

**SOLUTION:**

X = df.iloc[:, :-1].values

print(X)

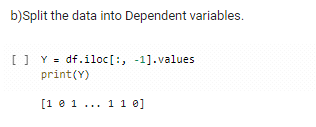


(b) Split the data into Dependent variables

**SOLUTION:**

Y = df.iloc[:, -1].values

print(Y)



**Question-9:**

Scale the independent variables

**SOLUTION:**

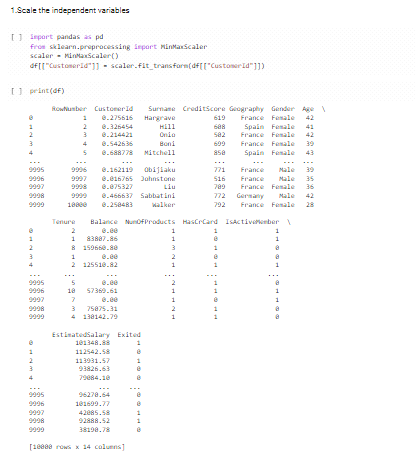
import pandas as pd

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

df[["CustomerId"]] = scaler.fit\_transform(df[["CustomerId"]])

print(df)



**Question-10:**

Split the data into training and testing

**SOLUTION:**

from sklearn.model\_selection import train\_test\_split

train\_size=0.8

X = df.drop(columns = ['Tenure']).copy()

y = df['Tenure']

X\_train, X\_rem, y\_train, y\_rem = train\_test\_split(X,y, train\_size=0.8)

test\_size = 0.5

X\_valid, X\_test, y\_valid, y\_test = train\_test\_split(X\_rem,y\_rem, test\_size=0.5)

print(X\_train.shape), print(y\_train.shape)

print(X\_valid.shape), print(y\_valid.shape)

print(X\_test.shape), print(y\_test.shape)

