# Assignment -2

Data Visualization and Preprocessing

|  |  |
| --- | --- |
| Assignment Date | 19 September 2022 |
| Student Name | Ishani. S |
| Student Roll Number | 211419104107 |
| Maximum Marks | 2 Marks |

# Question-1:

Download the dataset:

# Question-2:

Load the dataset.

|  |
| --- |
| **Solution:** |

import pandas as pd

import matplotlib.pyplot as plt

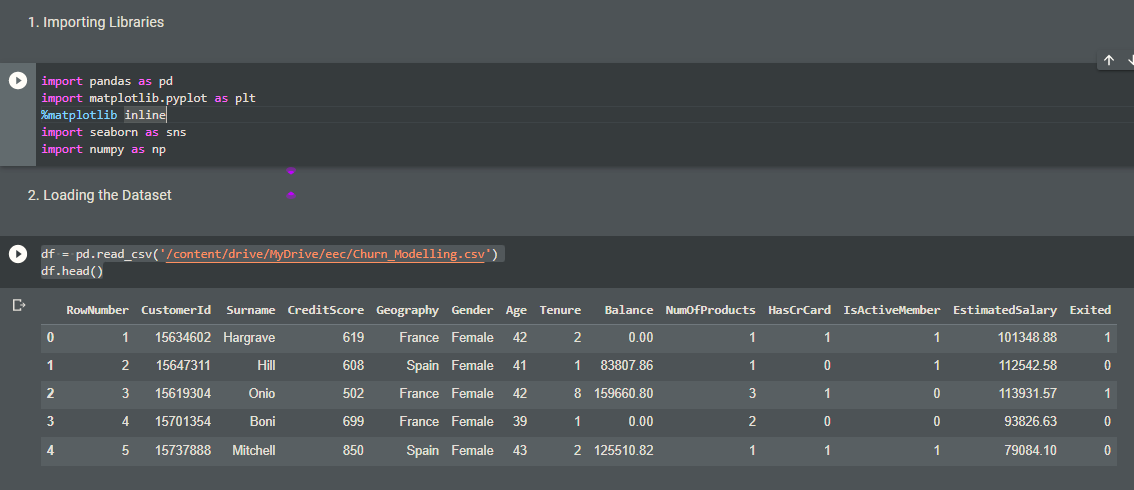
%matplotlib inline

import seaborn as sns

import numpy as np

df = pd.read\_csv('/content/drive/MyDrive/eec/Churn\_Modelling.csv')

df.head()

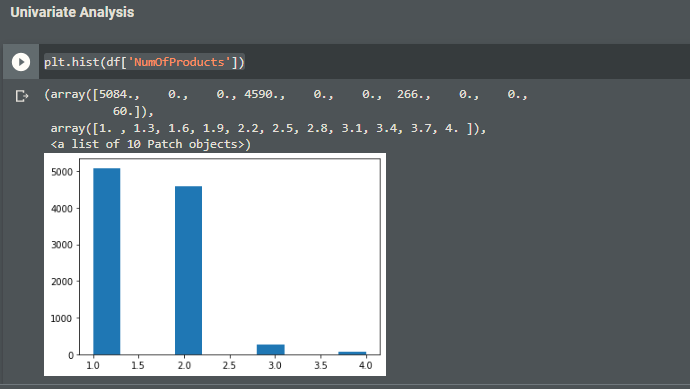


# Question-3:

Perform Below Visualizations. 1)Univariate Analysis

# Solution:

plt.hist(df['NumOfProducts'])



2)Bi - Variate Analysis

# Solution:

sns.barplot(x=df.NumOfProducts,y=df.Tenure)



1)Multivariate Analysis

# Solution:

sns.lineplot(

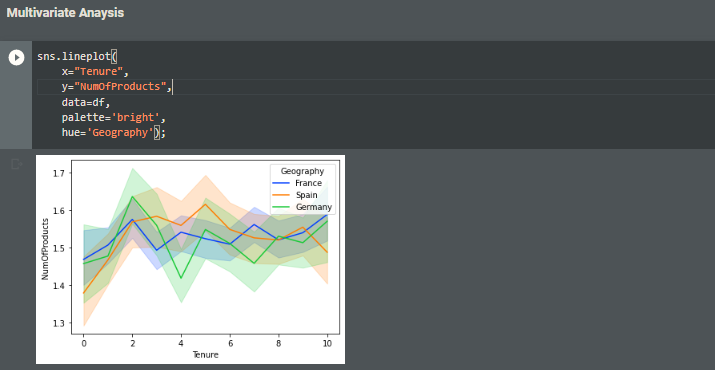
x="Tenure",

y="NumOfProducts",

data=df,

palette='bright',

hue='Geography');

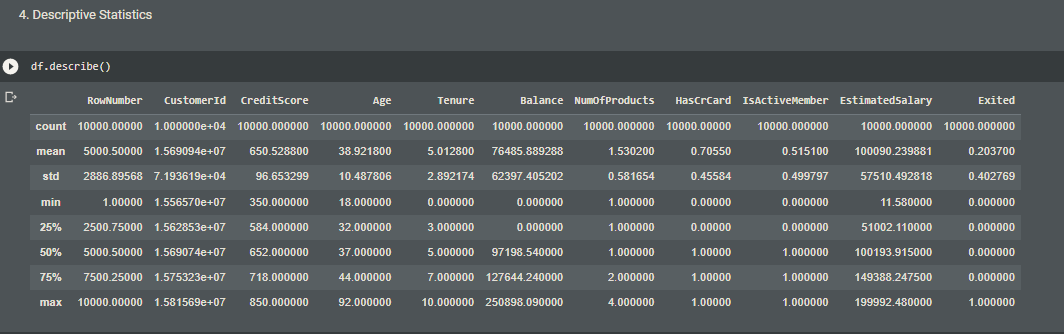


# Question-4:

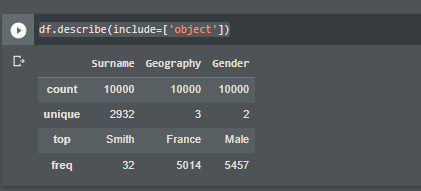
Perform descriptive statistics on the dataset.

# Solution:

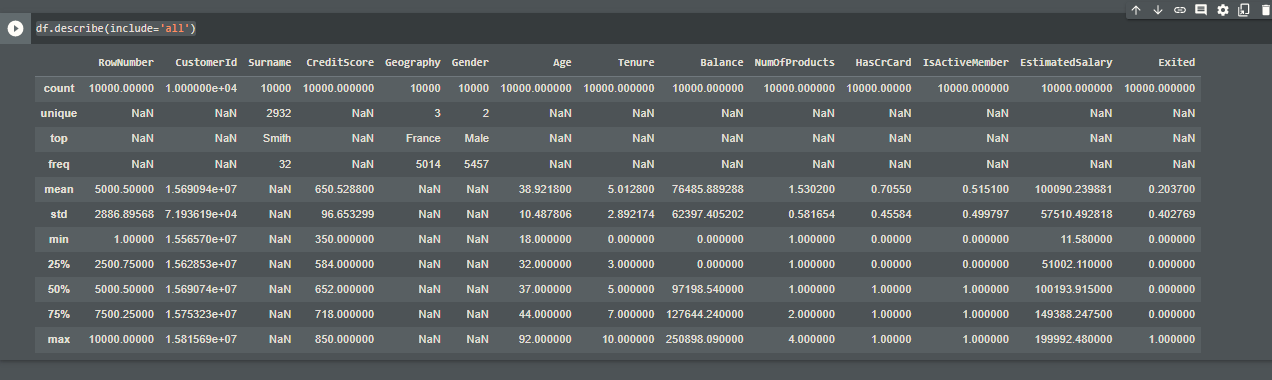
df**.**describe()



df.describe(include=['object'])



df.describe(include='all')

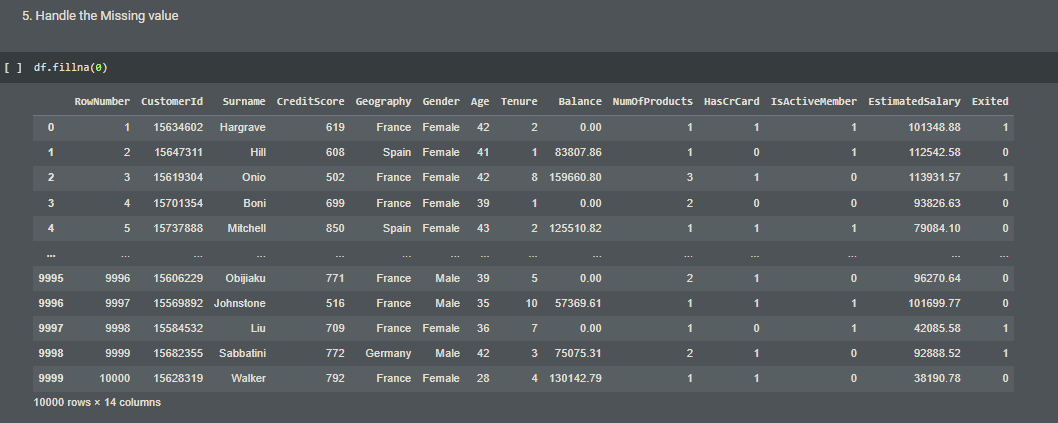


# Question-5:

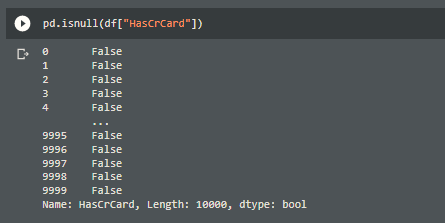
Handle the Missing values.

# Solution:

df**.**fillna(0)



d.isnull(df["HasCrCard"])



# Question-6:

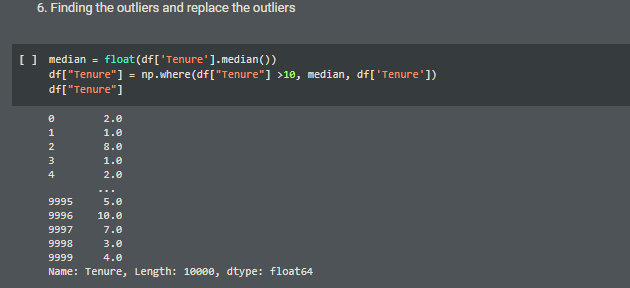
Find the outliers and replace the outliers

# Solution:

median = float(df['Tenure'].median())

df["Tenure"] = np.where(df["Tenure"] >10, median, df['Tenure'])

df["Tenure"]

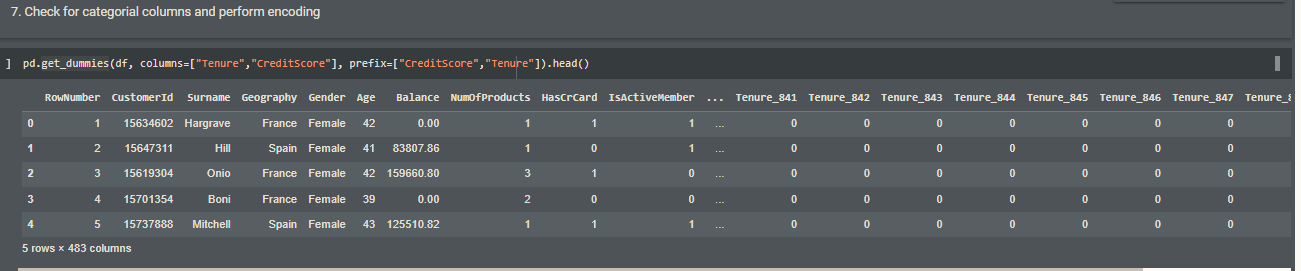


# Question-7:

Check for Categorical columns and perform encoding.

# Solution:

pd.get\_dummies(df, columns=["Tenure","CreditScore"], prefix=["CreditScore","Tenure"]).head()



# Question-8:

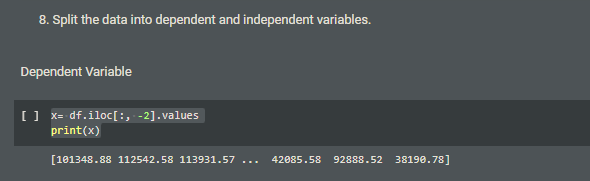
Split the data into dependent and independent variables.

# Solution:

Dependent Variable

x= df.iloc[:, -2].values

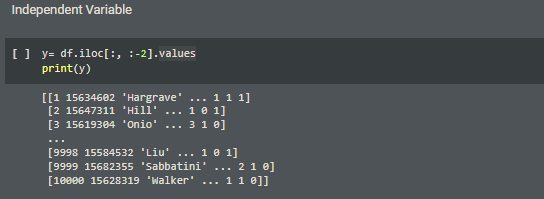
print(x)



Independent Variable

y= df.iloc[:, :-2].values

print(y)



# Question-9:

Scale the independent variables

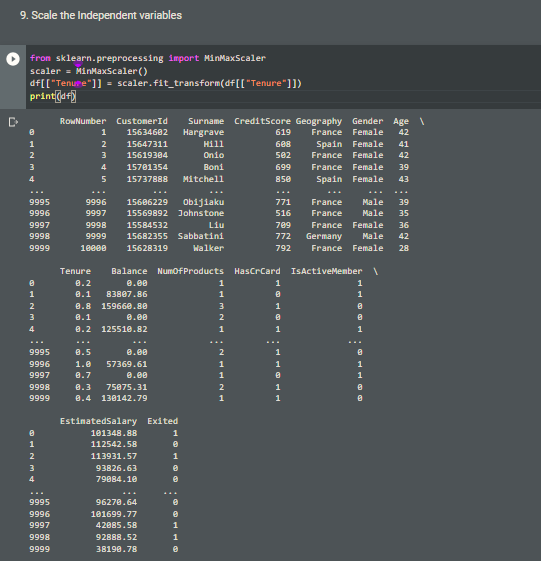
# Solution:

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

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

print(df)



# Question-10:

Testing and training data

# Solution:

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

# train\_size=0.7

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

# y = df['CreditScore']

# X\_train, X\_rem, y\_train, y\_rem = train\_test\_split(X,y, train\_size=0.7)

# test\_size = 0.4

# X\_valid, X\_test, y\_valid, y\_test = train\_test\_split(X\_rem,y\_rem, test\_size=0.4)

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

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

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

