Assignment -2

Data Visualization and Pre-Processing

|  |  |
| --- | --- |
| Assignment Date | 27 September 2022 |
| Student Name | S. Alagammai |
| Student Roll Number | 9517201903007 |
| Maximum Marks | 2 Marks |

# Question 1 - Load the dataset.

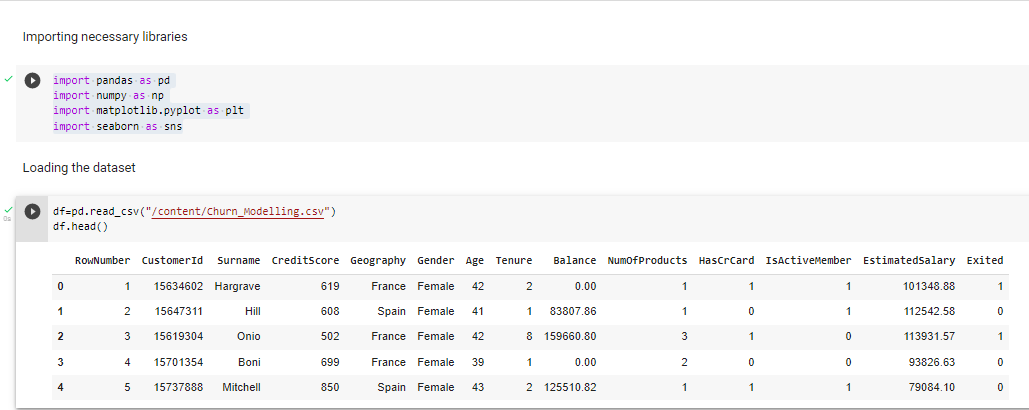
SOLUTION:

import pandas as pd import numpy as np

import matplotlib.pyplot as plt import seaborn as sns

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

OUTPUT:

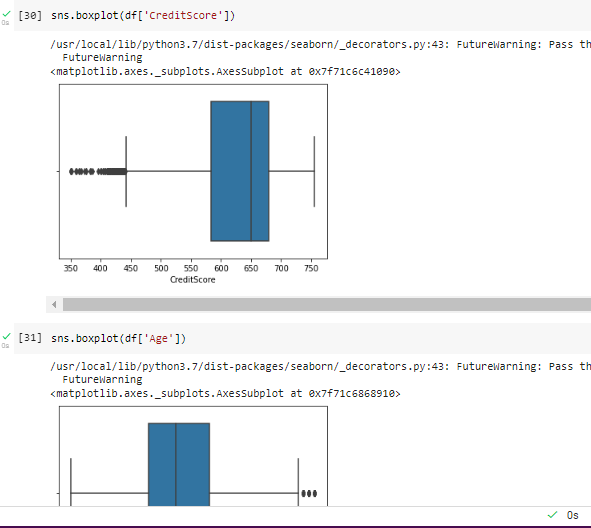


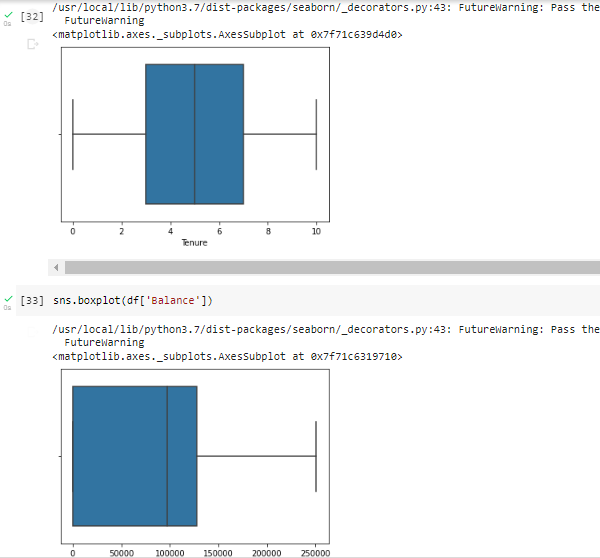
# Question 2 - Perform Univariate, Bivariate and Multivariate Analysis

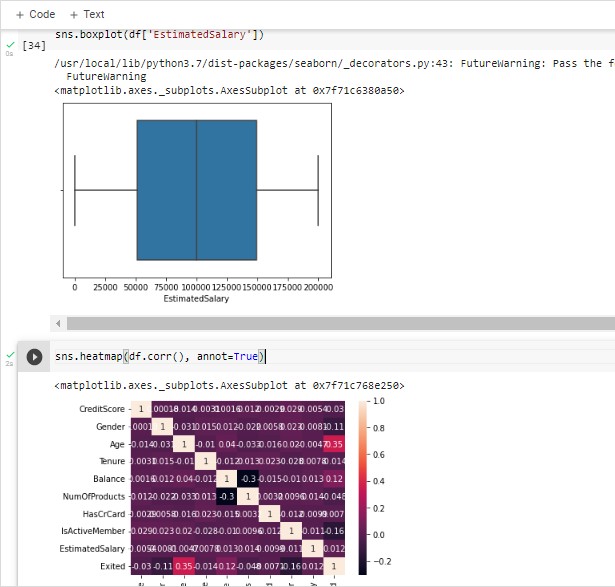
SOLUTION:

sns.boxplot(df['CreditScore']) sns.boxplot(df['Age']) sns.boxplot(df['Tenure']) sns.boxplot(df['Balance']) sns.boxplot(df['EstimatedSalary']) sns.heatmap(df.corr(), annot=True)

OUTPUT:



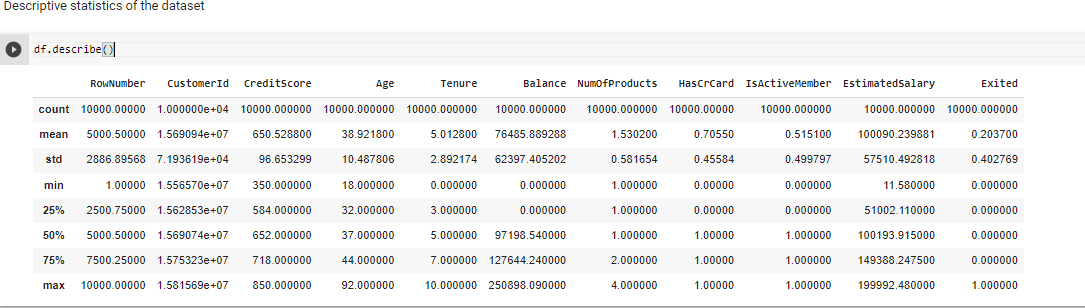




# Question 3 - Perform descriptive statistics on the dataset.

SOLUTION:

df.describe() OUTPUT:

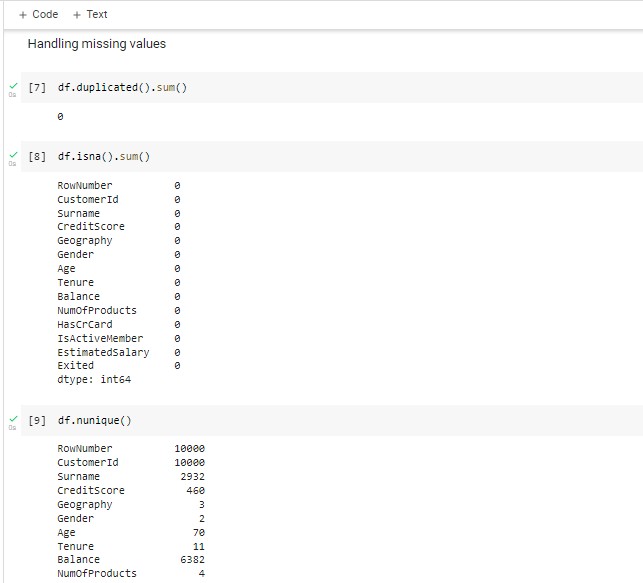


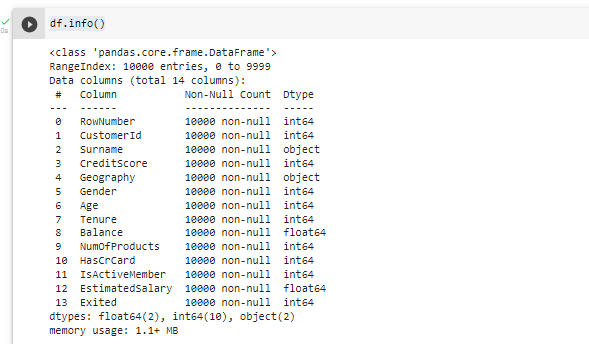
# Question 4 – Handle the missing values

SOLUTION:

df.duplicated().sum() df.nunique()

df.info() OUTPUT:

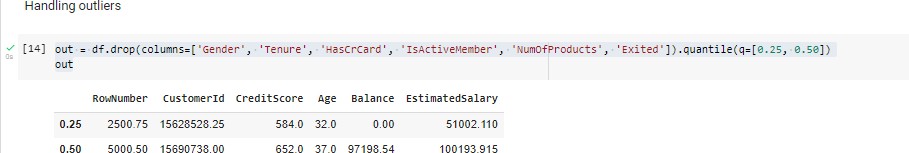




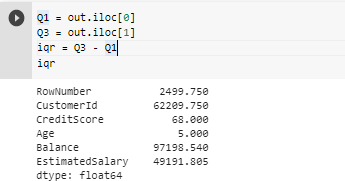
# Question 5 - Find the outliers and replace the outliers SOLUTION:

out = df.drop(columns=['Gender', 'Tenure', 'HasCrCard', 'IsActiveMember', 'NumOfProducts', 'Exi ted']).quantile(q=[0.25, 0.50])

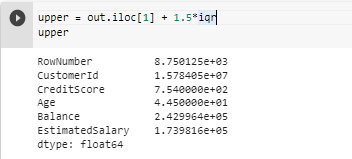
out



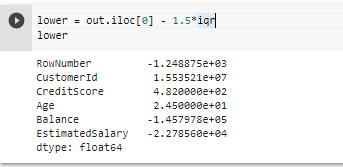
Q1 = out.iloc[0] Q3 = out.iloc[1] iqr = Q3 - Q1 iqr



upper = out.iloc[1] + 1.5\*iqr upper



lower = out.iloc[0] - 1.5\*iqr lower



# Replace outliers

SOLUTION:

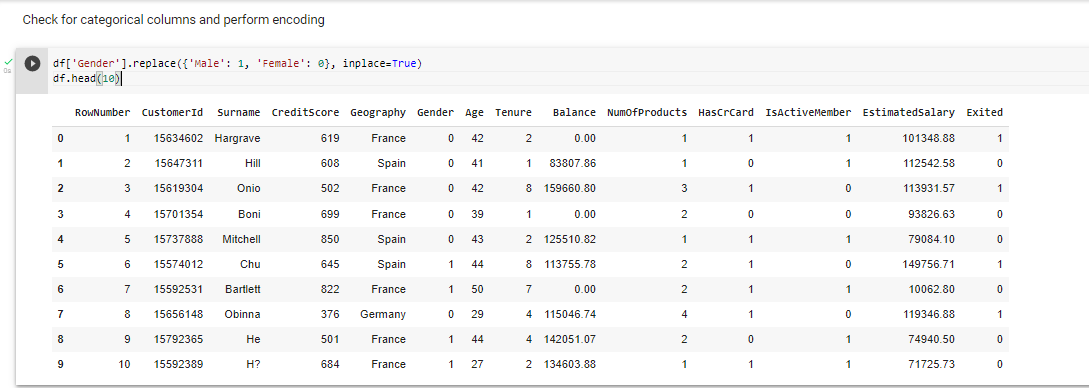
df['CreditScore'] = np.where(df['CreditScore']>756, 650.5288, df['CreditScore']) df['Age'] = np.where(df['Age']>62, 38.9218, df['Age'])

# Question 6 - Check for Categorical columns and perform encoding.

SOLUTION:

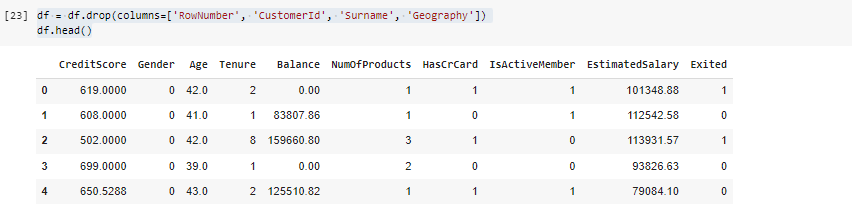
df['Gender'].replace({'Male': 1, 'Female': 0}, inplace=True) df.head(10)

OUTPUT:

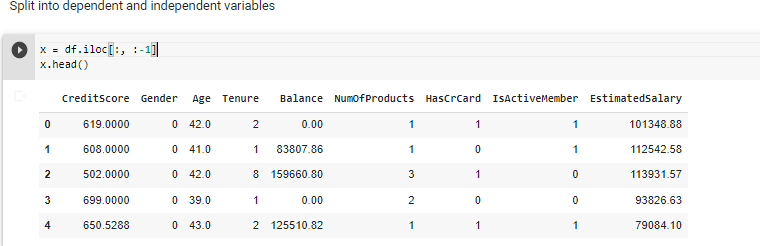


# Question 7 – Split the data into dependent and independent variables.

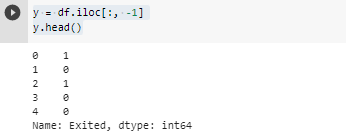
SOLUTION:

df = df.drop(columns=['RowNumber', 'CustomerId', 'Surname', 'Geography']) df.head()

x = df.iloc[:, :-1] x.head()



y = df.iloc[:, -1] y.head()



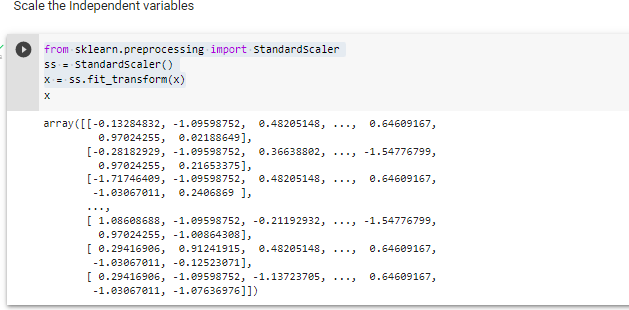
# Question 8 – Scale the independent variables

SOLUTION:

from sklearn.preprocessing import StandardScaler ss = StandardScaler()

x = ss.fit\_transform(x) x

OUTPUT:



# Question 9 - Split the data into training and testing

SOLUTION:

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

x\_train, y\_test = train\_test\_split(x, y, test\_size=0.2, random\_state=0) print(x\_train.shape)

print(x\_test.shape) print(y\_train.shape) print(y\_test.shape)

OUTPUT:

