Assignment -2

Data Visualization and Pre-processing

|  |  |
| --- | --- |
| Assignment Date | 24 September 2022 |
| Student Name | MANIKANDAN R |
| Student Roll Number | 611719104014 |
| Maximum Marks | 2 Marks |

**To Perform Below Tasks to complete the assignment:-**

# Step 1. Download the dataset: Dataset

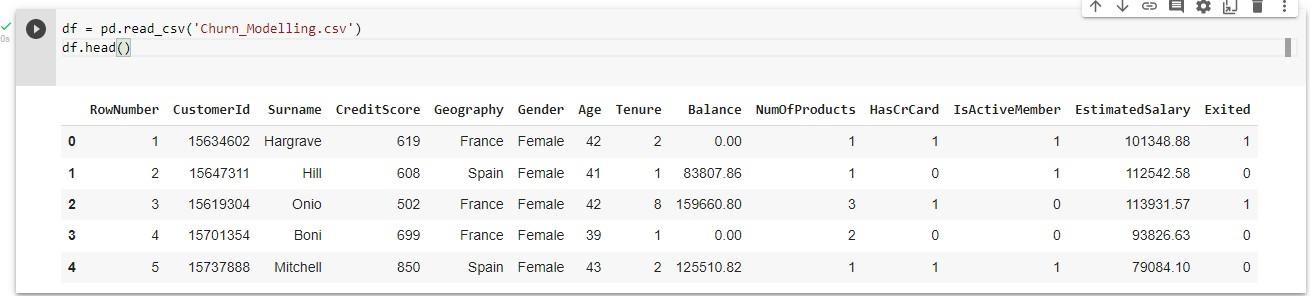
Step 2. Load the dataset.

import pandas as pd import numpy as np

import matplotlib.pyplot as plt import seaborn as sns

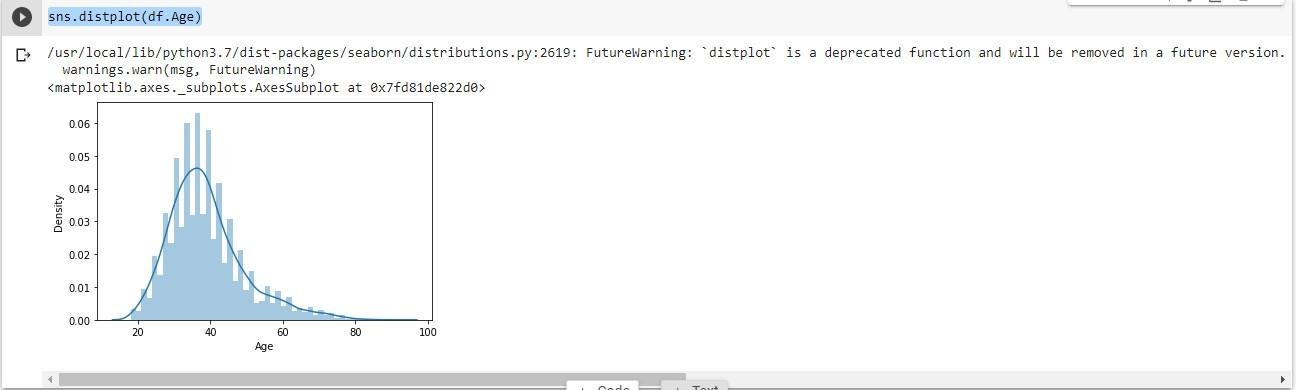
df = pd.read\_csv('Churn\_Modelling.csv') df.head()

# Output :



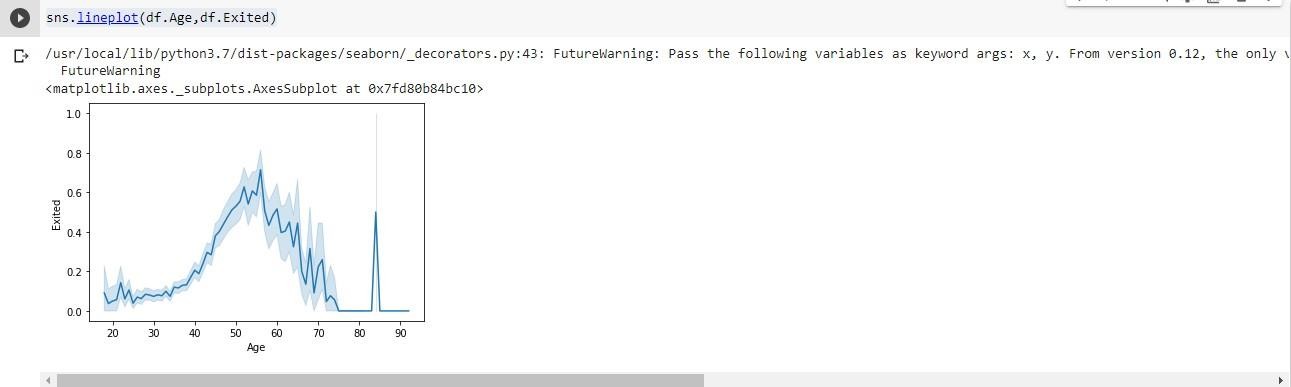
Step 3. Perform Below Visualizations.

* Univariate Analysis sns.distplot(df.Age) Output :



sns.lineplot(df.Age,df.Exited)

# Output :



plt.pie(df.Gender.value\_counts(),[0.2,0],colors=['red','green'],labels=['Male','Female'],autopct='%1.1f%%') plt.title('GENDER')

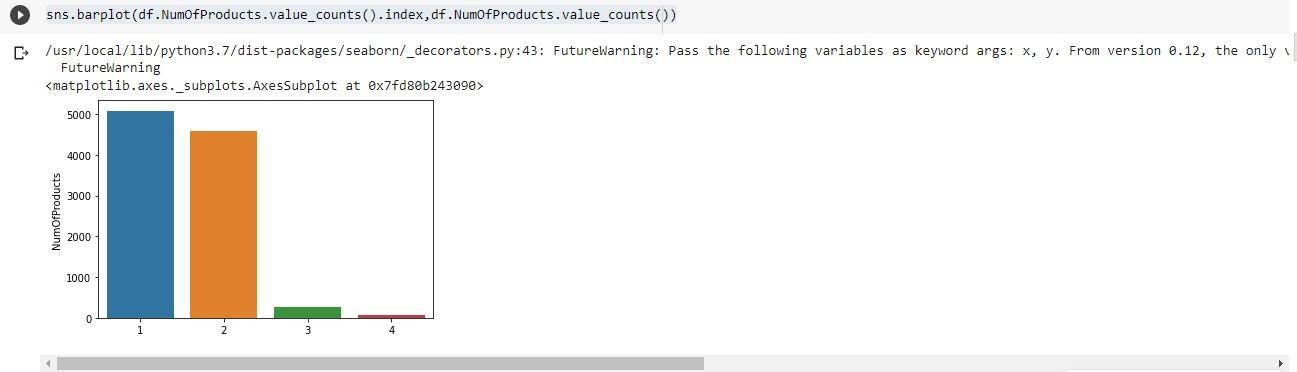
plt.show()

# Output :



sns.barplot(df.NumOfProducts.value\_counts().index,df.NumOfProducts.value\_counts())

# Output :



* Bi - Variate Analysis

def countplot\_2(x,hue,title=None,figsize=(6,5)): plt.figure(figsize=figsize) sns.countplot(data=df[[x,hue]],x=x,hue=hue) plt.title(title)

plt.show()

countplot\_2('IsActiveMember','NumOfProducts','Credit Card Holders Product Details')

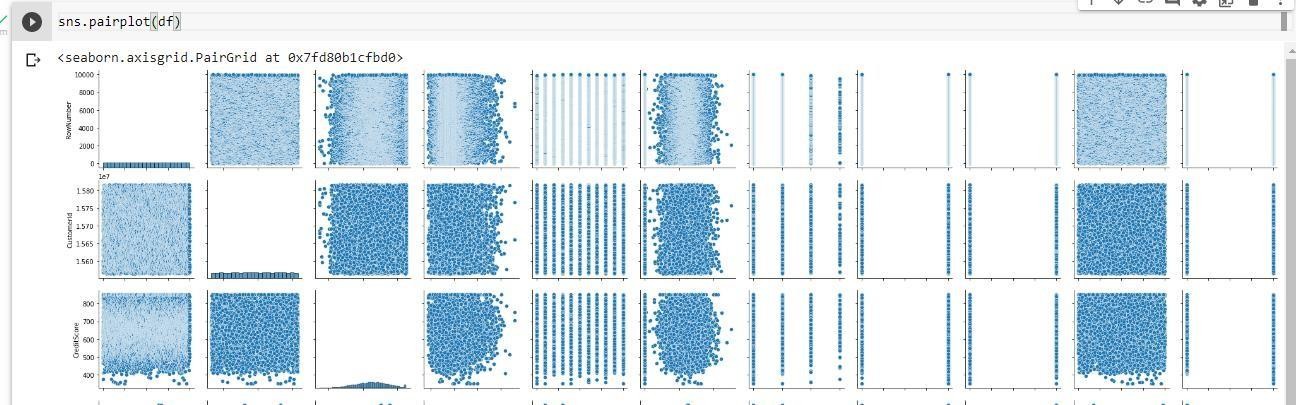
# Output :



* Multi - Variate Analysis

sns.pairplot(df)

# Output :



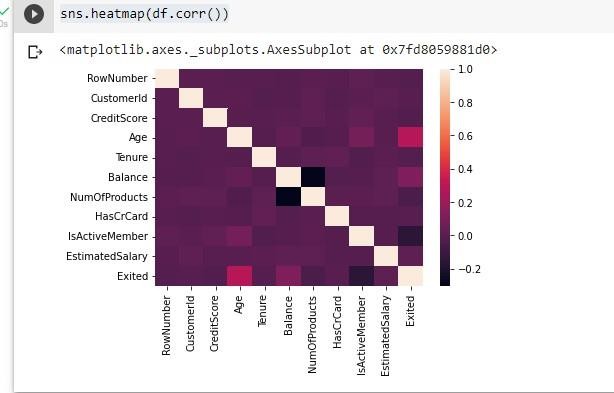
df.corr()

# Output :



sns.heatmap(df.corr())

# Output :



plt.figure(figsize=(16,15)) sns.heatmap(df.corr(),annot=True) plt.show()

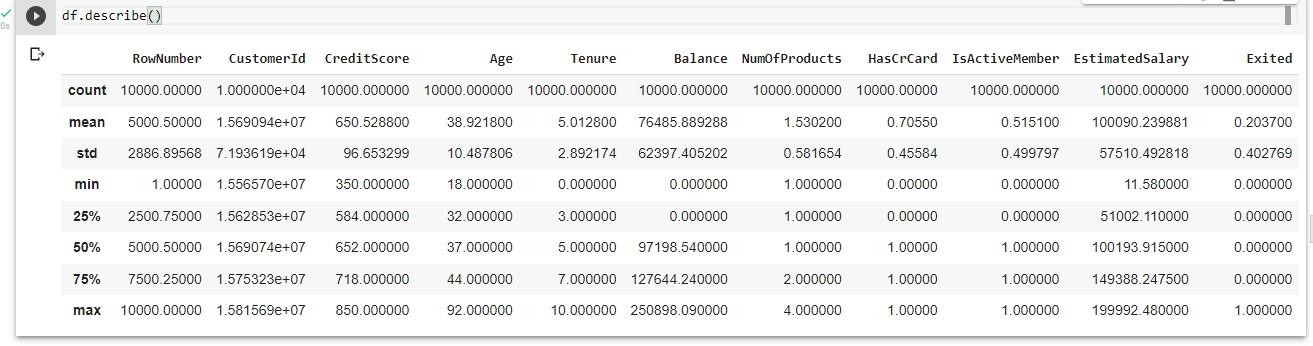
# Output :



Step 4. Perform descriptive statistics on the dataset.

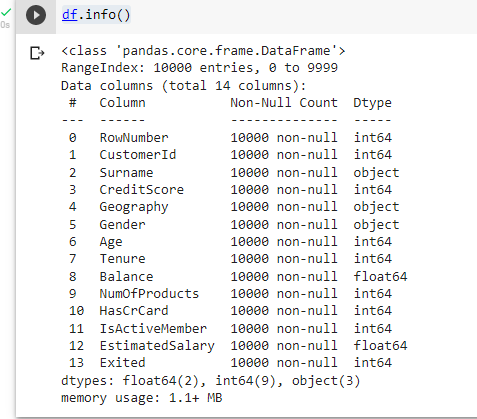
df.describe()

# Output :



df.info()

# Output :

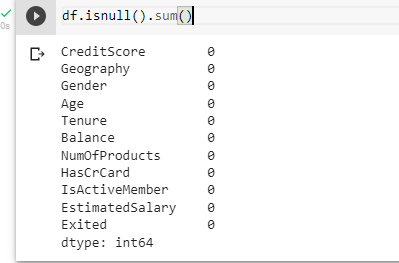


Step 5. Handle the Missing values.

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

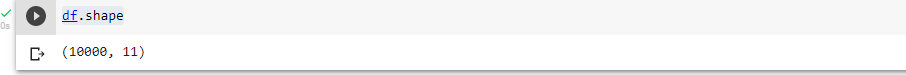
df.isnull().sum()

# Output :



df.shape

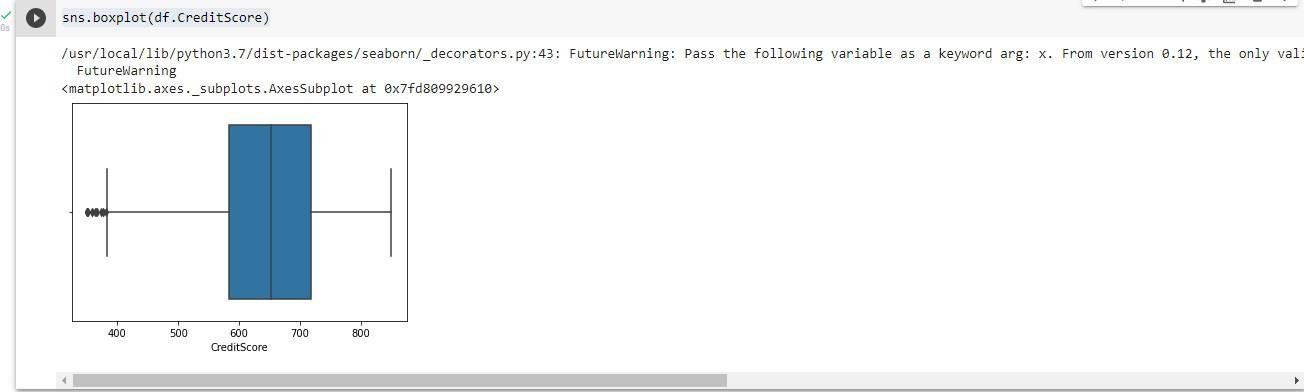
# Output :



Step 6. Find the outliers and replace the outliers

sns.boxplot(df.CreditScore)

# Output :



Q1 = df.CreditScore.quantile(0.25) Q3 = df.CreditScore.quantile(0.75)

IQR = Q3-Q1

upper\_limit = Q3 + (1.5\*IQR) lower\_limit = Q1 - (1.5\*IQR)

df['CreditScore'] = np.where(df['CreditScore']<lower\_limit,650,df['CreditScore']) sns.boxplot(df.CreditScore)

# Output :



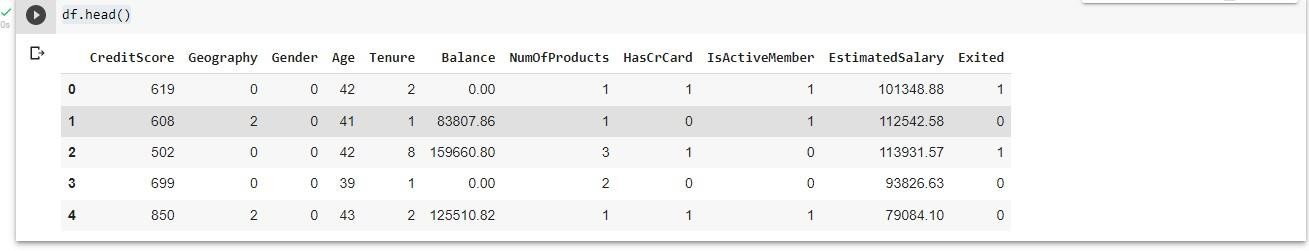
Step 7. Check for Categorical columns and perform encoding. from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

df.Geography = le.fit\_transform(df.Geography) df.Gender = le.fit\_transform(df.Gender)

df.head()

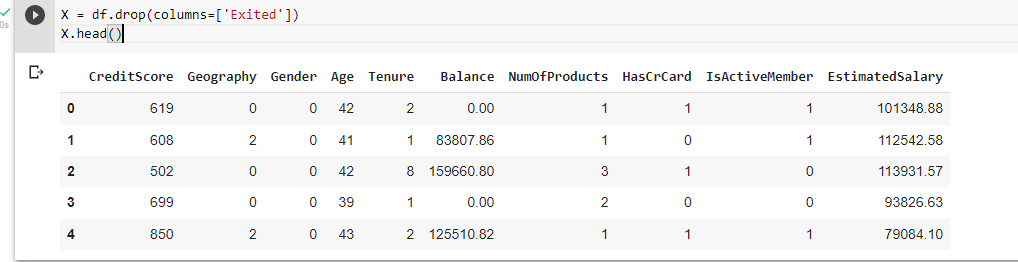
# Output :



Step 8. Split the data into dependent and independent variables.

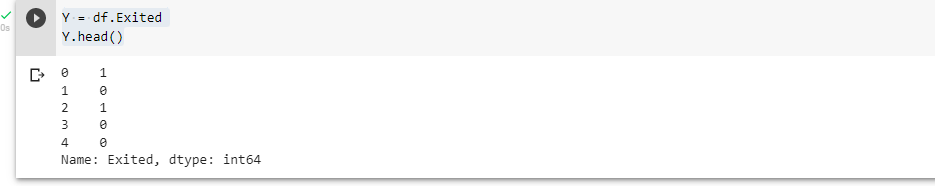
X = df.drop(columns=['Exited']) X.head()

# Output :



Y = df.Exited Y.head()

# Output :



Step 9. Scale the independent variables

from sklearn.preprocessing import MinMaxScaler scale = MinMaxScaler()

X\_scaled = pd.DataFrame(scale.fit\_transform(X),columns=X.columns) Step 10. Split the data into training and testing

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

x\_train , y\_train , x\_test , y\_test = train\_test\_split(X\_scaled,Y,test\_size=0.2,random\_state=0)

# Output :

