ASSIGNMENT–2

PythonProgramming

|  |  |
| --- | --- |
| AssignmentDate | 29-09-2022 |
| StudentName | Nagam Pragathi |
| StudentRollNumber | 410719106068 |
| MaximumMarks | 2 Mark |

# Question-1:

1 . Importing Required PackageSolution:

import pandas as pdimport seaborn as snsimportnumpyasnp

frommatplotlibimportpyplotasplt

%matplotlibinline

Question-2:

1. Loading the DatasetSolution:

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

Output:



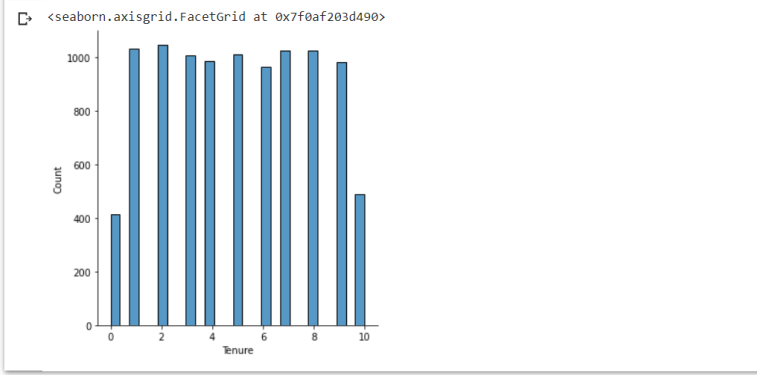
1. Visualizations

# Question-3:

* 1. UnivariateAnalysisSolution:

sns.displot(df.Tenure)

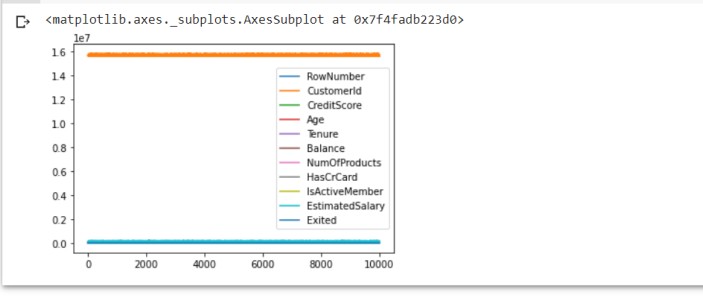
Output:



* 1. Bi-VariateAnalysisSolution:

df.plot.line()

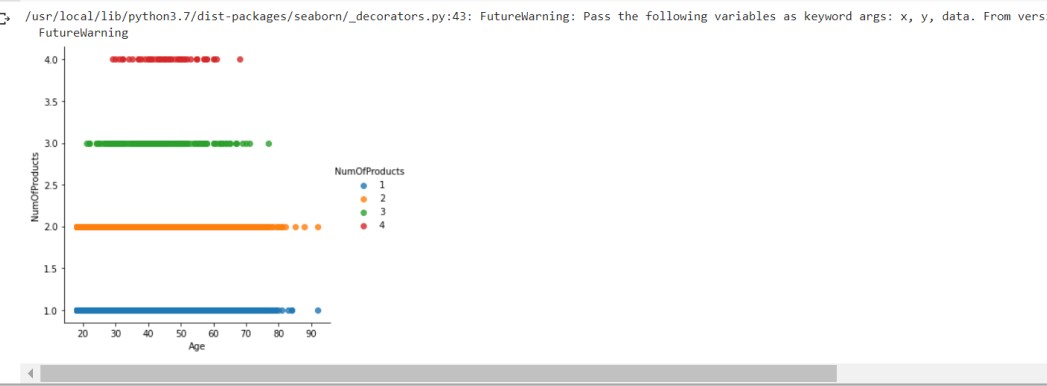
Output:



* 1. Multi-VariateAnalysisSolution:

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

Output:



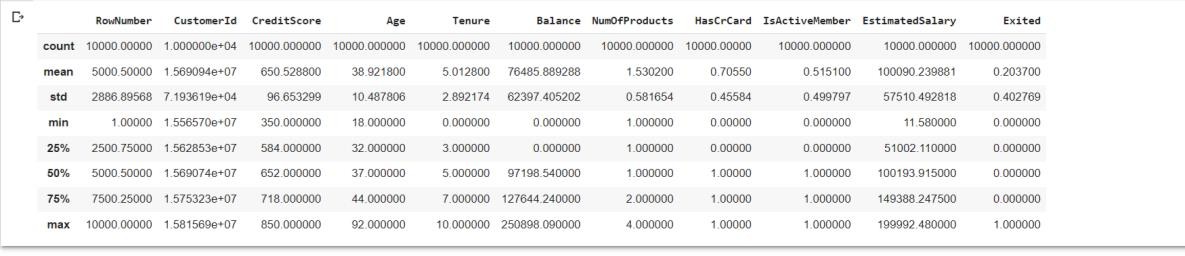
1. Performdescriptivestatisticsonthedataset.

# Question-4:

Solution:

df.describe()

Output:



1. HandletheMissingvalues.

# Question-5:

Solution:

data = pd.read\_csv("Churn\_Modelling.csv")pd.isnull(data["Gender"])

Output:



# Question-6:

1. Findtheoutliersandreplacetheoutliers.Solution:

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

Output:



# Question-7:

1. CheckforCategoricalcolumnsandperformencoding.Solution:

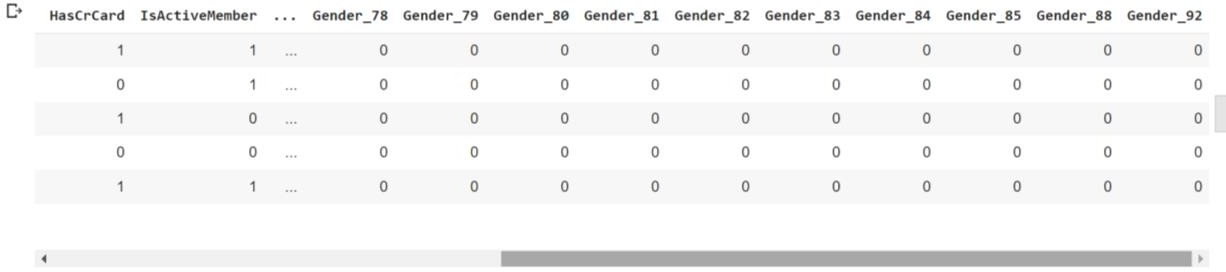
pd.get\_dummies(df,columns=["Gender","Age"],prefix=["Age","Gender"]

).head()

Output:



Output:

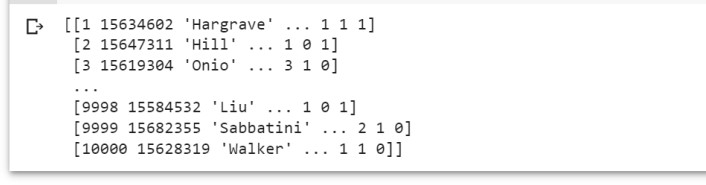


# Question-8:

1. Splitthedataintodependentandindependentvariables
   1. SplitthedataintoIndependentvariables.Solution:

X=df.iloc[:,:-2].valuesprint(X)

Output:



* 1. SplitthedataintoDependentvariables.Solution:

Y = df.iloc[:, -1].valuesprint(Y)

Output:



# Question-9:

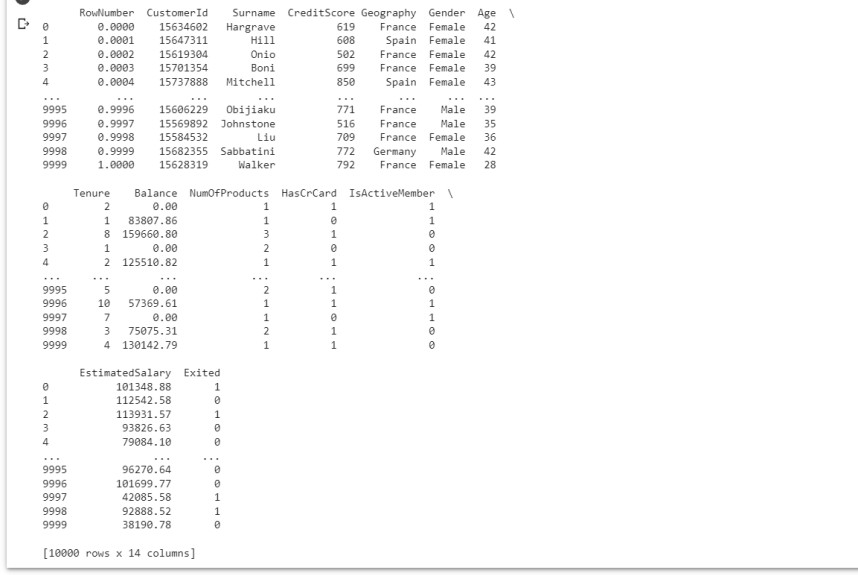
1. Scale the independent variablesSolution:

importpandasaspd

fromsklearn.preprocessingimportMinMaxScalerscaler=MinMaxScaler()

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

Output:



# Question-10:

1. SplitthedataintotrainingandtestingSolution:

fromsklearn.model\_selectionimporttrain\_test\_splittrain\_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)

Output:

