|  |  |
| --- | --- |
| Assignment Date | 24-09-2022 |
| Student Name | Ms . Amethyst Sardonyx P |
| Student Roll Number | 923319104005 |
| Maximum Marks | 2 Mark |

Question-1 :

1 . Importing Required Package

Solution :

import pandas as pd

import seaborn as sns

import numpy as np

from matplotlib import pyplot as plt

%matplotlib inline

Question-2 :

**2. Loading the Dataset**

Solution :

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

df

Output:



# ****3. Visualizations****

Question-3 :

# ****3.1 Univariate Analysis****

# ****Solution:****

sns.displot(df.Tenure)

Output:



# ****3.2 Bi-Variate Analysis****

Solution:

df.plot.line()

Output:



# ****3.3 Multi - Variate Analysis****

Solution:

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

Output:

# 

# ****4. Perform descriptive statistics on the dataset.****

Question-4 :

Solution:

df.describe()

Output:

# 

# ****5. Handle the Missing values.****

Question-5 :

Solution:

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

pd.isnull(data["Gender"])

Output:

# 

Question-6:

**6. Find the outliers and replace the outliers.**

Solution:

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

df["Tenure"]

Output:



Question-7 :

# ****7. Check for Categorical columns and perform encoding.****

Solution:

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

Output:

# 

Output:

# 

Question-8:

# ****8. Split the data into dependent and independent variables****

# ****8.1 Split the data into Independent variables.****

Solution:

X = df.iloc[:, :-2].values

print(X)

Output:

# 

# ****8.2 Split the data into Dependent variables.****

Solution:

Y = df.iloc[:, -1].values

print(Y)

Output:



Question-9 :

# ****9. Scale the independent variables****

Solution:

import pandas as pd

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

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

print(df)

Output:

# 

Question-10 :

# ****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)

Output:

