*ASSIGNMENT – 3*

*Python Programming*

# Question-1 :

*1 . Importing Required Package Solution :*

*import pandas as pd import seaborn as sns import numpy as np*

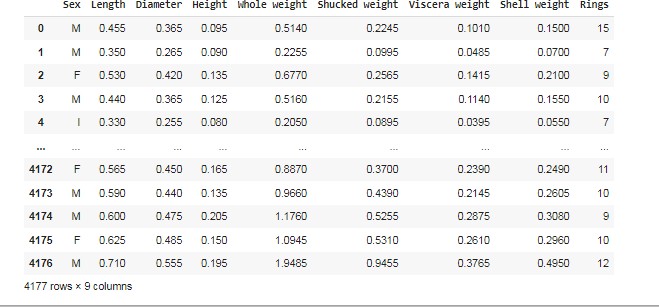
*from matplotlib import pyplot as plt*

*%matplotlib inline*

*Question-2 :*

1. ***Loading the Dataset*** *Solution :*

*df = pd.read\_csv("abalone.csv") df*

*Output:*

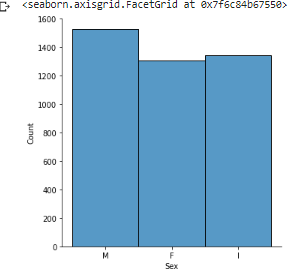
1. ***Visualizations***

# Question-3 :

* 1. ***Univariate Analysis Solution:***

*sns.displot(df.Sex)*

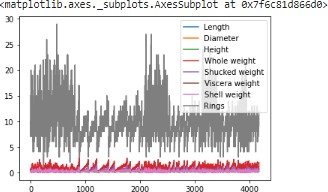
*Output:*



* 1. ***Bi-Variate Analysis*** *Solution:*

*df.plot.line()*

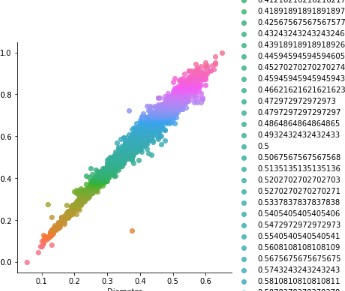
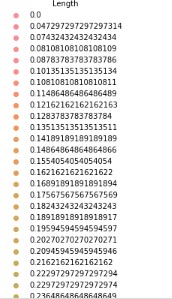
*Output:*



* 1. ***Multi - Variate Analysis*** *Solution:*

sns.lmplot("Diameter","Length",df,hue="Length", fit\_reg=False);

*Output:*



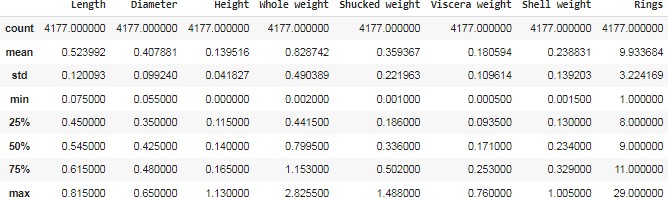
1. ***Perform descriptive statistics on the dataset.***

# Question-4 :

*Solution:*

*df.describe()*

*Output:*



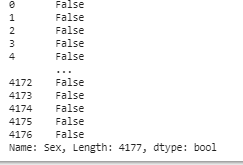
1. ***Handle the Missing values.***

# Question-5 :

*Solution:*

data = pd.read\_csv("abalone.csv") pd.isnull(data["Sex"])

*Output:*

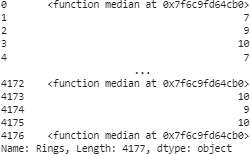


# Question-6:

1. *Find the outliers and replace the outliers. Solution:*

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

*Output:*

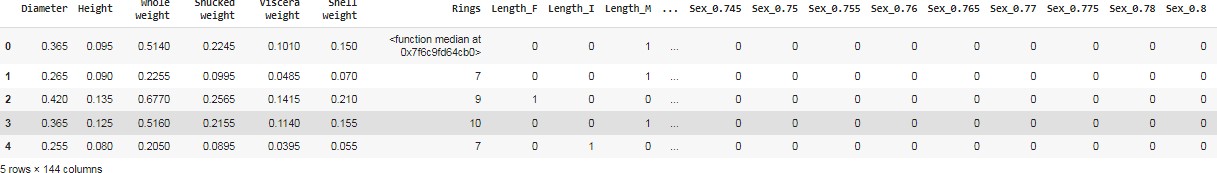


# Question-7 :

1. ***Check for Categorical columns and perform encoding.*** *Solution:*

*pd.get\_dummies(df, columns=["Sex", "Length"], prefix=["Length", "Sex"]).head()*

*Output:*

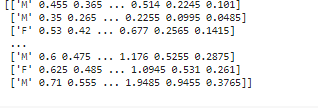


# Question-8:

1. ***Split the data into dependent and independent variables***
   1. ***Split the data into Independent variables.*** *Solution:*

*X = df.iloc[:, :-2].values print(X)*

*Output:*



* 1. ***Split the data into Dependent variables.*** *Solution:*

*Y = df.iloc[:, -1].values print(Y)*

*Output:*



# Question-9 :

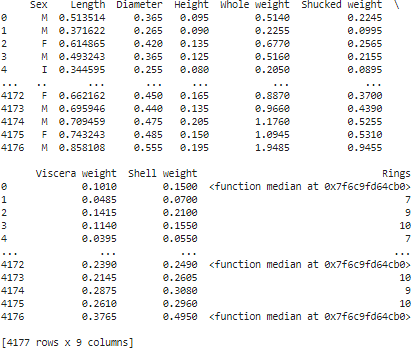
1. ***Scale the independent variables*** *Solution:*

*import pandas as pd*

*from sklearn.preprocessing import MinMaxScaler scaler = MinMaxScaler()*

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

*Output:*



# Question-10 :

1. ***Split the data into training and testing*** *Solution:*

*from sklearn.model\_selection import train\_test\_split train\_size=0.8*

*X = df.drop(columns = ['Sex']).copy() y = df['Sex']*

*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:*



1. *Build the Model*

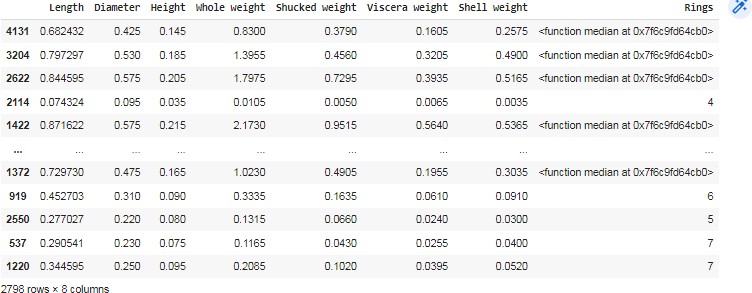
test\_size = 0.33

seed = 7

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=test\_size, random\_state=seed)

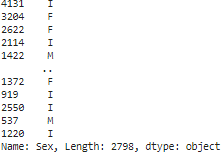
1. *Train the model*

X\_train

*Output:*

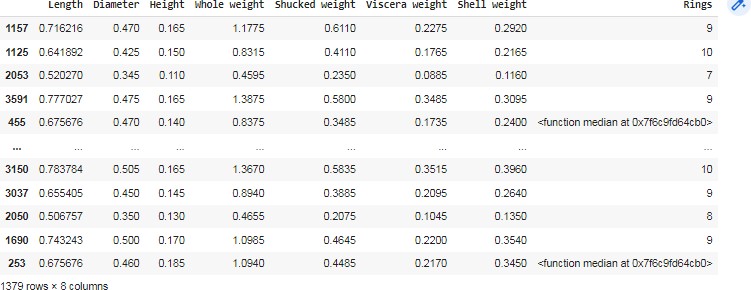
y\_train

*Output:*



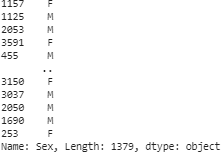
1. *Test the model: X\_test*

*Output:*



y\_test

*Output:*



1. *Measure the performance using Metrics*

from sklearn.metrics import r2\_score

from sklearn.metrics import mean\_absolute\_error from sklearn.metrics import mean\_squared\_error X\_train = [5, -1, 2, 10]

y\_test = [3.5, -0.9, 2, 9.9]

print ('R Squared =',r2\_score(X\_train, y\_test))

print ('MAE =',mean\_absolute\_error(X\_train, y\_test)) print ('MSE =',mean\_squared\_error(X\_train, y\_test))

Output:

