Assignment #2

Task-1
Perform linear or Regression is a dataset to predict brain weight from head size

Source Code

C:\Users\mhza\Documents\Machine learning online class 2020 NED\Assignment 2\Assignment 2 code .py

temp.py | Assignment 2 code .py

```
# Muhammad Huzaifa Abbasi
# Simple Linear Regression

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing dataset.csv (Gender,  Age, brain weight, head size)
dataset = pd.read_csv('dataset.csv')
X = dataset.iloc[:, 2:3].values
y = dataset.iloc[:, 3].values

# Splitting the dataset into the Training set and Test set (using 1:4 as Test Size)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 1/4, random_state = 0)

# Fitting Simple Linear Regression to the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

# Predicting the Test set results
y_pred = regressor.predict(X_test)

# Visualising the Training set results
plt.scatter(X_train, y_train, color = 'salmon')
plt.plot(X_train, regressor.predict(X_train), color = 'navy')
plt.title('Plot-1 \n brain weight from head size (Training set)')
plt.xlabel('Head Size')
plt.ylabel('Brain weight')
plt.show()

# Visualising the Test set results
plt.scatter(X_test, y_test, color = 'b')
plt.plot(X_train, regressor.predict(X_train), color = 'peru')
plt.title('Plot-2 \n brain weight from head size (Test set)')
plt.xlabel('Head Size')
plt.ylabel('Brain weight')
plt.show()
```

Picture- 1 (Spyder- Assignment 2 code.py)

Plot-1 Training Set

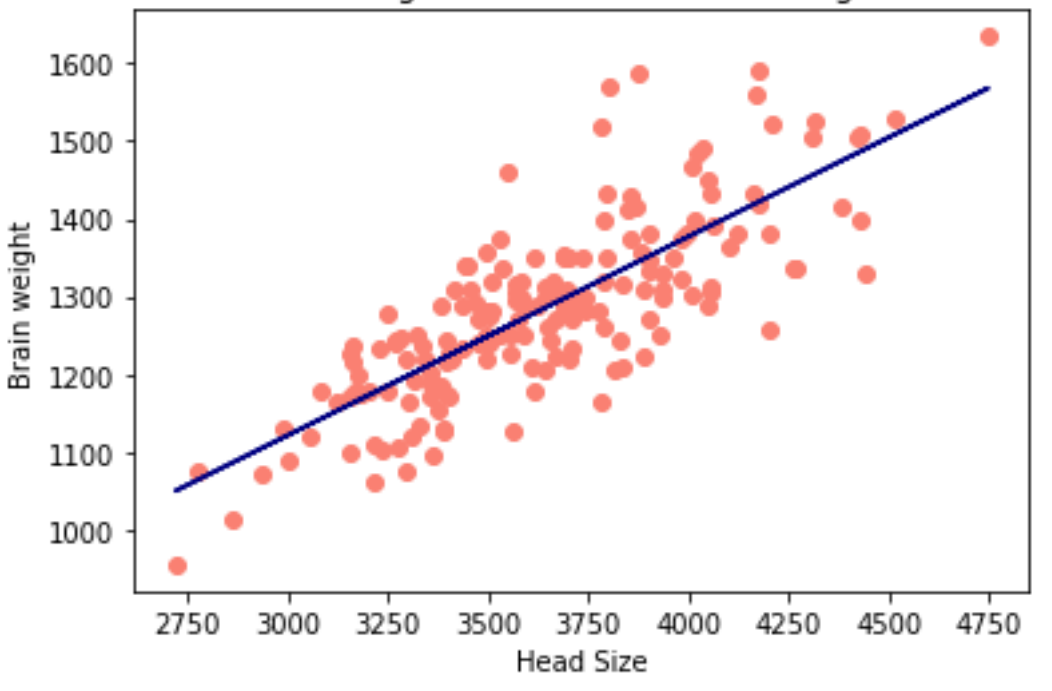

Figure-I (Assignmet 2 plot-1 Training Set.png)

Plot-2 Test Set

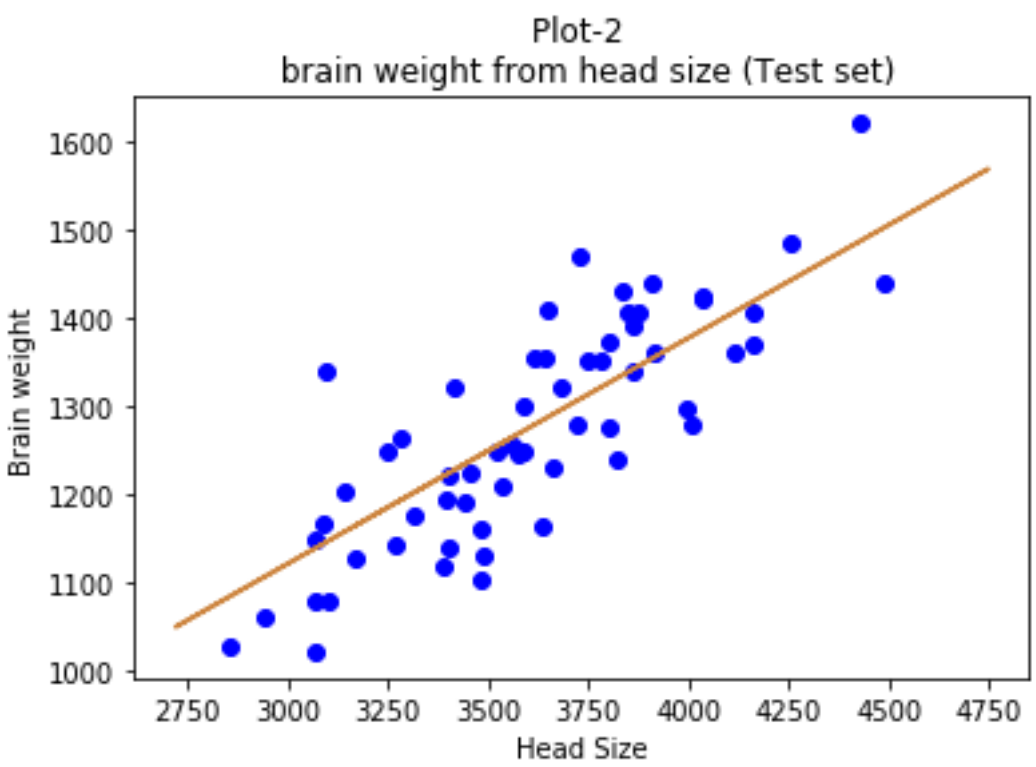

Figure II (Assignmet 2 plot-2 Test Set.png)

Task-2

Create two random arrays A and B, and multiply them. Get their result in C and add 1 to every element of C.

Source Code

```
#Muhammad Huzaifa Abbasi
import numpy as np

#Random Arrays
A = np.random.randn(4,4)
B = np.random.randn(4,4)

#Multiplying Arrays A&B
C = np.multiply(A,B)

#Adding 1 to each element of C
C1= np.add(np.ones((4,4)),C)

```

Picture- 2 (Spyder- Classwork2.py)

Variable Explorer

A - NumPy array

	0	1	2	3
0	-1.52588	-0.968134	-0.757421	1.694
1	0.697085	2.10152	0.292981	-1.0051
2	0.0898488	1.40613	-0.0253143	-0.404309
3	1.44191	-0.0191031	1.44583	0.524145

B - NumPy array

	0	1	2	3
0	1.31847	0.888611	-0.00344965	0.382584
1	-0.599887	0.538224	-0.979128	0.968576
2	0.143624	-1.18156	0.764674	0.249514
3	1.20352	1.64944	0.0816186	-0.596897

C - NumPy array

	0	1	2	3
0	-2.01183	-0.860295	0.00261284	0.648098
1	-0.418172	1.13109	-0.286865	-0.973517
2	0.0129044	-1.66143	-0.0193572	-0.100881
3	1.73537	-0.0315093	0.118006	-0.31286

C1 - NumPy array

	0	1	2	3
0	-1.01183	0.139705	1.00261	1.6481
1	0.581828	2.13109	0.713135	0.0264825
2	1.0129	-0.661431	0.980643	0.899119
3	2.73537	0.968491	1.11801	0.68714

Picture- 3 (Spyder- Classwork2.py)