Jisha Chacko

S3RMCA:A

ROLLNO:44

REG NO:AJC20MCA-2044

DATA SCIENCE LAB

1.GIVEN 3 MATRICES A,B,C WRITE A PROGRAM TO PERFORM THE FOLLOWING OPERATION

A.B+4(B.B)-C/4

CODE

```
import numpy as np
A=np.array([[1,2],[3,4]])
B=np.array([[2,2],[5,4]])
C=np.array([[1,3],[4,4]])
print('matrix operation')
print('matrix multiplication (A*B)')
x=np.multiply(A,B)
print(x)
print('second part 4(B*B)')
y=np.multiply(4,(B,B))
print(y)
print('matrix division(C/4)')
z=np.divide(C,4)
print(z)
print('A.B+4BB-C/4 =')
print((x+y)-z)
```

OUTPUT

```
C:\Users\ajcemca\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/ajcemca/PycharmProjects/pythonProject/1.py
matrix operation
matrix multiplication (A*B)
[[ 2  4]
 [15 16]]
second part 4(B*B)
[[[ 8  8]
  [20 16]]

 [[ 8  8]
  [20 16]]]
matrix division(C/4)
[[0.25 0.75]
 [1.   1.  ]]
A.B+4BB-C/4 =
[[[ 9.75 11.25]
  [34.   31.  ]]

 [[ 9.75 11.25]
  [34.   31.  ]]]

Process finished with exit code 0
```

2.PROGRAM TO IMPLEMENT MULIPLE REGRESSION USING BOSTON DATASET AVAILABLE IN PUBLIC DOMAIN AND EVALUATE PERFORMANCE DISPLAY THE COEFFICIENT VALUES

```
import pandas as pd
import numpy as np

from sklearn.model_selection import train_test_split
from sklearn.datasets import load_boston

from sklearn import linear_model
iboston  =load_boston()
x = iboston.data
y = iboston.target
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_sta
te=42)
reg=linear_model.LinearRegression()
reg.fit(x_train,y_train)
r_sq=reg.score(x,y)
y_predict=reg.predict(x_test)
print('predicted response',y_predict)
print('coefficent of dertermination',r_sq)
print("maximum varience{}".format(reg.predict(y_predict, y_test)))
```

OUTPUT

```
predicted response [28.64896005 36.49501384 15.4111932  25.40321303 18.85527988 23.14668944
 17.3921241  14.07859899 23.03692679 20.59943345 24.82286159 18.53057049
 -6.86543527 21.80172334 19.22571177 26.19191985 20.27733882  5.61596432
 40.44887974 17.57695918 27.44319095 30.1715964  10.94055823 24.02083139
 18.07693812 15.934748   23.12614028 14.56052142 22.33482544 19.3257627
 22.16564973 25.19476081 25.31372473 18.51345025 16.6223286  17.50268505
 30.94992991 20.19201752 23.90440431 24.86975466 13.93767876 31.82504715
 42.56978796 17.62323805 27.01963242 17.19006621 13.80594006 26.10356557
 20.31516118 30.08649576 21.3124053  34.15739602 15.60444981 26.11247588
 39.31613646 22.99282065 18.95764781 33.05555669 24.85114223 12.91729352
 22.68101452 30.80336295 31.63522027 16.29833689 21.07379993 16.57699669
 20.36362023 26.15615896 31.06833034 11.98679953 20.42550472 27.55676301
 10.94316981 16.82660609 23.92909733  5.28065815 21.43504661 41.33684993
 18.22211675  9.48269245 21.19857446 12.95001331 21.64822797  9.3845568
 23.06060014 31.95762512 19.16662892 25.59942257 29.35043558 20.13138581
 25.57297369  5.42970803 20.23169356 15.1949595  14.03241742 20.91078077
 24.82249135 -0.47712079 13.70520524 15.69525576 22.06972676 24.64152943
 10.7382866  19.68622564 23.63678009 12.07974981 18.47894211 25.52713393
 20.93461307 24.6955941   7.59054562 19.01046053 21.9444339  27.22319977
 32.18608828 15.27826455 34.39190421 12.96314168 21.01681316 28.57880911
 15.86300844 24.85124135  3.37937111 23.90465773 25.81792146 23.11020547
 25.33489201 33.35545176 20.60724498 38.4772665  13.97398533 25.21923987
 17.80946626 20.63437371  9.80267398 21.07953576 22.3378417  32.32381854
```

```
 15.86300844 24.85124135  3.37937111 23.90465773 25.81792146 23.11020547
 25.33489201 33.35545176 20.60724498 38.4772665  13.97398533 25.21923987
 17.80946626 20.63437371  9.80267398 21.07953576 22.3378417  32.32381854
 31.48694863 15.46621287 16.86242766 28.99330526 24.95467894 16.73633557
  6.12858395 26.65990044 23.34007187 17.40367164 13.38594123 39.98342478
 16.68286302 18.28561759]
coefficent of derermination 0.7365934508044946
```