

Name:Saurabh singh

Roll no: 583

PRN: 202201090160

Div: E-4

Sub: EDS Assignment 3

Code:

```
f1 = open("D:\\Assignment_564\\testmarks1.csv",'r')
f2 = open("D:\\Assignment_564\\testmarks2.csv",'r')
import numpy as np
array=
np.loadtxt('D:\\Assignment_564\\testmarks1.csv',dtype=str,delimiter=',')
print(array)
import numpy as np
array1=
np.loadtxt('D:\\Assignment_564\\testmarks1.csv',dtype=float,delimiter=',',skip
rows = 1)
print(array1)
array2 = array1.astype(float)
print(array1)
RollNo = array1[:,0]
EDS = array1[:,1]
SON = array1[:,2]
DT = array1[:,3]
ET = array1[:,4]
print(RollNo)
print(EDS)
print(SON)
print(DT)
print(ET)
#mean marks of students in EDS course
mean_EDS = np.mean(EDS)
print(mean_EDS)
#standard deviation of marks in SON course
std_deviation_SON = np.std(SON)
print(std_deviation_SON)
#correlation between two courses marks {DT and ET}
corr_DT_ET = np.corrcoef(DT,ET)
print(corr_DT_ET)
#to print the sum of marks in the row
print(np.sum(SON,axis = 0))
#to print maximum marks scored in DT
print(max(DT))
#to print minimum marks scored in EDS
```

```

print(min(EDS))
#to stack the two arrays vertically
Vstack = np.vstack((EDS,DT))
print(Vstack)
#to copy the transpose of an array in another array
sample_array = np.fastCopyAndTranspose(Vstack)
print(sample_array)
#to check whether the performance of a student is better in one course
compared to another course
for i in range(0,10):
    performance_checker = np.greater_equal(ET[i],SON[i])
    print(RollNo[i])
    if(performance_checker == False):
        print("not good performance than SON")
    else:
        print("good performance than SON")
Increasing_marks_SON = np.sort(SON)
print(Increasing_marks_SON)

```

SCREENSHOT:

```

1 f1 = open("D:\\Assignment_564\\testmarks1.csv","r")
26
Python
1 F2 = open("D:\\Assignment_564\\testmarks2.csv","r")
27
Python
1 import numpy as np
2 array= np.loadtxt('D:\\Assignment_564\\testmarks1.csv',dtype=str,delimiter=',')
3 print(array)
28
Python
... [['RollNo' 'EDS' 'SON' 'DT' 'ET']
[['001' '43.05' '27.79' '28.7' '27.79']
[['002' '43.47' '28.52' '28.98' '27.89']
[['003' '42.24' '28.16' '28.16' '25.63']
[['004' '39.24' '26.16' '26.16' '26.16']
[['005' '40.9' '26.03' '27.27' '25.65']
[['006' '39.47' '26.31' '26.31' '25.21']
[['007' '41.68' '25.63' '27.79' '25.46']
[['008' '42.19' '27.61' '28.13' '26.21']
[['009' '44.75' '28.35' '29.83' '28.21']
[['010' '46.95' '28.88' '31.3' '28.53']]
Python
1 import numpy as np
2 array1= np.loadtxt('D:\\Assignment_564\\testmarks1.csv',dtype=float,delimiter=',',skiprows = 1)
3 print(array1)
29
Python
... [[001. 43.05 27.79 28.7 27.79]
[002. 43.47 28.52 28.98 27.89]
[003. 42.24 28.16 28.16 25.63]
[004. 39.24 26.16 26.16 26.16]
[005. 40.9 26.03 27.27 25.65]
[006. 39.47 26.31 26.31 25.21]
[007. 41.68 25.63 27.79 25.46]
[008. 42.19 27.61 28.13 26.21]
[009. 44.75 28.35 29.83 28.21]
[010. 46.95 28.88 31.3 28.53]]

```

```
File Edit Selection View Go Run Terminal Help Assignment3_564official.ipynb - Visual Studio Code
Assignment3_564official.ipynb X
+ Code + Markdown + Run All + Clear All Outputs + Outline ... Python 3.11.3

1 import numpy as np
2 array1= np.loadtxt('D:\\Assignment_564\\testmarks1.csv',dtype=float,delimiter=',',skiprows = 1)
3 print(array1)

[29]
... [[801. 43.05 27.79 28.7 27.79]
[802. 43.47 28.52 28.98 27.89]
[803. 42.24 28.16 28.16 25.63]
[804. 39.24 26.16 26.16 26.16]
[805. 40.9 26.03 27.27 25.65]
[806. 39.47 26.31 26.31 25.21]
[807. 41.68 25.63 27.79 25.46]
[808. 42.19 27.61 28.13 26.21]
[809. 44.75 28.35 29.83 28.21]
[810. 46.95 28.88 31.3 28.53]]

1 array2 = array1.astype(float)
2 int(array1)

[30]
... [[801. 43.05 27.79 28.7 27.79]
[802. 43.47 28.52 28.98 27.89]
[803. 42.24 28.16 28.16 25.63]
[804. 39.24 26.16 26.16 26.16]
[805. 40.9 26.03 27.27 25.65]
[806. 39.47 26.31 26.31 25.21]
[807. 41.68 25.63 27.79 25.46]
[808. 42.19 27.61 28.13 26.21]
[809. 44.75 28.35 29.83 28.21]
[810. 46.95 28.88 31.3 28.53]]

1 RollNo = array1[:,0]
2 EDS = array1[:,1]
3 SON = array1[:,2]
4 DT = array1[:,3]
5 ET = array1[:,4]
```

```
File Edit Selection View Go Run Terminal Help Assignment3_564official.ipynb - Visual Studio Code
Assignment3_564official.ipynb X
+ Code + Markdown + Run All + Clear All Outputs + Outline ... Python 3.11.3

1 RollNo = array1[:,0]
2 EDS = array1[:,1]
3 SON = array1[:,2]
4 DT = array1[:,3]
5 ET = array1[:,4]

[38]

1 print(RollNo)
2 print(EDS)
3 print(SON)
4 print(DT)
5 print(ET)

[39]
... [801. 802. 803. 804. 805. 806. 807. 808. 809. 810.]
[43.05 43.47 42.24 39.24 40.9 39.47 41.68 42.19 44.75 46.95]
[27.79 28.52 28.16 26.16 26.03 26.31 25.63 27.61 28.35 28.88]
[28.7 28.98 28.16 26.16 27.27 26.31 27.79 28.13 29.83 31.3 ]
[27.79 27.89 25.63 26.16 25.65 25.21 25.46 26.21 28.21 28.53]

1 #mean marks of students in EDS course
2 mean_EDS = np.mean(EDS)
3 print(mean_EDS)

[47]
... 42.394

1 #standard deviation of marks in SON course
2 std_deviation_SON = np.std(SON)
3 print(std_deviation_SON)

[48]
... 1.1324857614998962
```

```
File Edit Selection View Go Run Terminal Help
Assignment3_564official.ipynb - Visual Studio Code

+ Code + Markdown + Run All + Clear All Outputs + Outline ...

Python 3.11.3

1 #correlation between two courses marks {DT and ET}
2 corr_DT_ET = np.corrcoef(DT,ET)
3 print(corr_DT_ET)

[42]
... [[1.          0.84076728]
      [0.84076728 1.        ]]

1 #to print the sum of marks in the row
2 print(np.sum(SON,axis = 0))

[50]
... 273.44

1 #to print maximum marks scored in DT
2 print(max(DT))

[51]
... 31.3

1 #to print minimum marks scored in EDS
2 print(min(EDS))

[52]
... 39.24

+ Code + Markdown

1 #to stack the two arrays vertically
2 Vstack = np.vstack((EDS,DT))
3 print(Vstack)

[57]
... [[43.05 43.47 42.24 39.24 40.9 39.47 41.68 42.19 44.75 46.95]
```

```
File Edit Selection View Go Run Terminal Help
Assignment3_564official.ipynb - Visual Studio Code

+ Code + Markdown + Run All + Clear All Outputs + Outline ...

Python 3.11.3

1 #to copy the transpose of an array in another array
2 sample_array = np.fastCopyAndTranspose(Vstack)
3 print(sample_array)

[58]
... [[43.05 28.7 ]
      [43.47 28.98]
      [42.24 28.16]
      [39.24 26.16]
      [40.9 27.27]
      [39.47 26.31]
      [41.68 27.79]
      [42.19 28.13]
      [44.75 29.83]
      [46.95 31.3 ]]

1 #to check whether the performance of a student is better in one course compared to another course
2 for i in range(0,10):
3     performance_checker = np.greater_equal(ET[i],SON[i])
4     print(RolNo[i])
5     if(performance_checker == False):
6         print("not good performance than SON")
7     else:
8         print("good performance than SON")

[59]
... 881.0
good performance than SON
882.0
not good performance than SON
883.0
not good performance than SON
884.0
good performance than SON
885.0
not good performance than SON
886.0
```

File Edit Selection View Go Run Terminal Help Assignment3_564official.ipynb - Visual Studio Code

Untitled-2.ipynb • assign3.ipynb Assignment3_564official.ipynb X

+ Code + Markdown | Run All | Clear All Outputs | Outline ... Python 3.11.3

```
good performance than SON
885.0
not good performance than SON
886.0
not good performance than SON
887.0
not good performance than SON
888.0
not good performance than SON
889.0
not good performance than SON
810.0
not good performance than SON
```

```
1 Increasing_marks_SON = np.sort(SON)
2 print(Increasing_marks_SON)
```

[71] Python

... [25.63 26.03 26.16 26.31 27.61 27.79 28.16 28.35 28.52 28.88]

```
1
```

[] Python

Ln 3, Col 18 (102 selected) Cell 2 of 18 R

