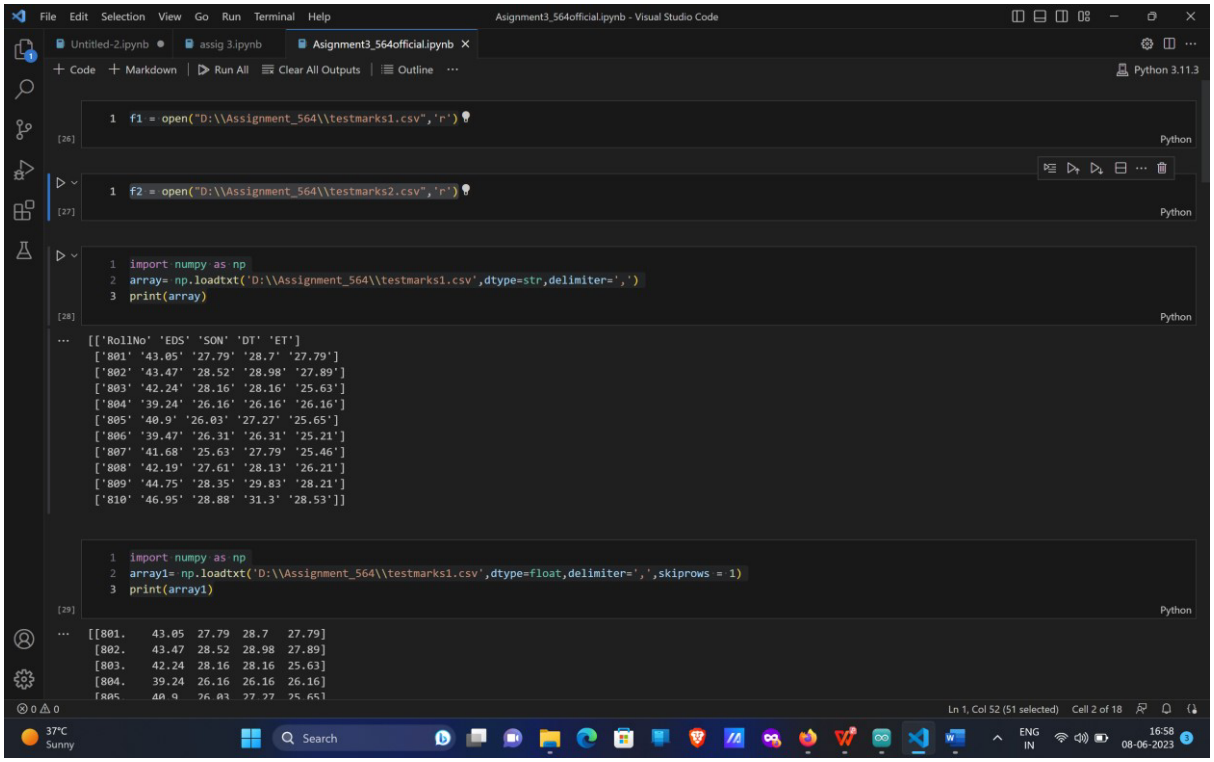
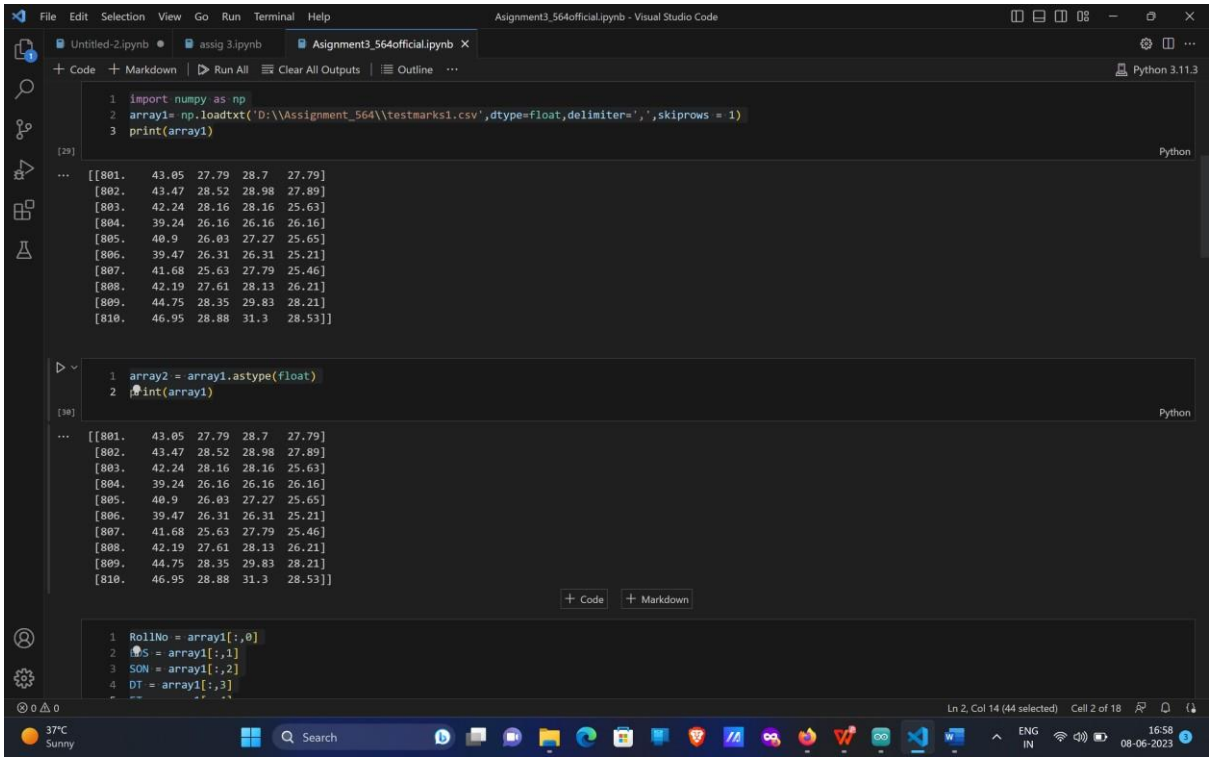
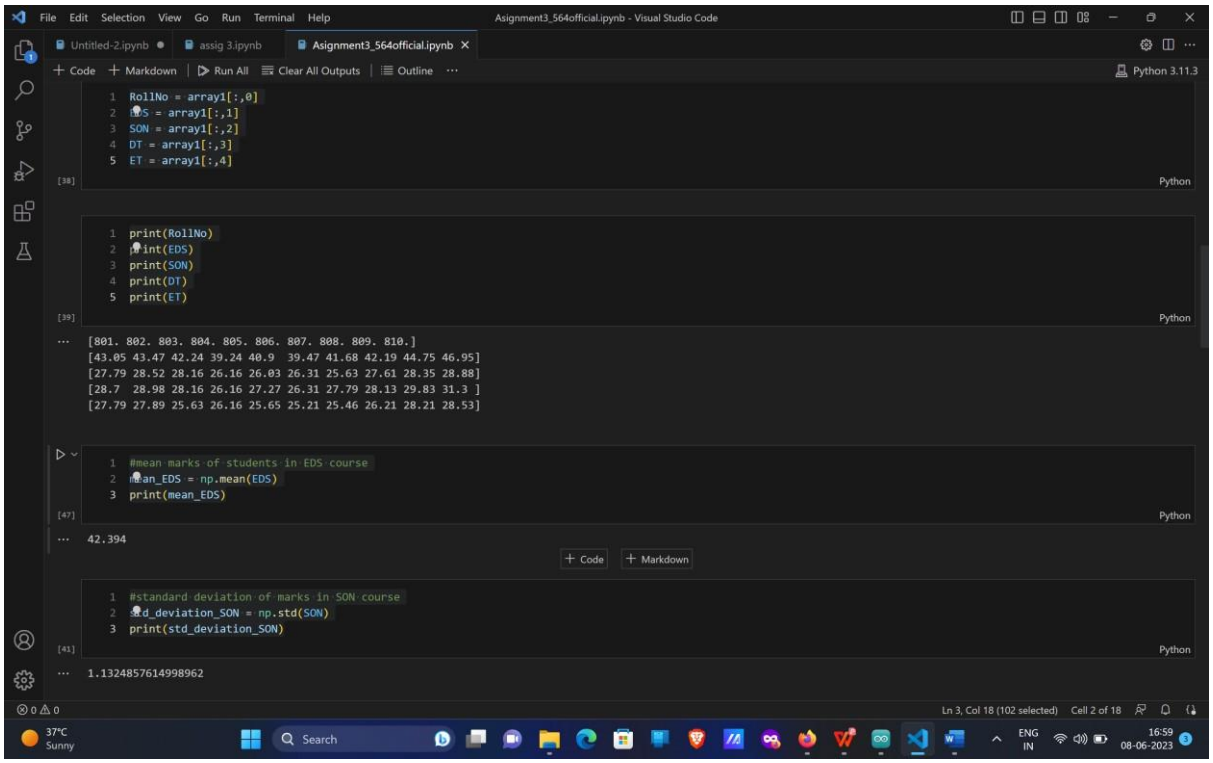
```
f1 = open("D:\\Assignment_575\\testmarks1.csv",'r') f2 =
open("D:\\Assignment_575\\testmarks2.csv",'r') import numpy as np array=
np.loadtxt('D:\\Assignment_575\\testmarks1.csv',dtype=str,delimiter=',')
print(array) import numpy as np array1=
np.loadtxt('D:\\Assignment_575\\testmarks1.csv',dtype=float,delimiter=',',skip
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
```

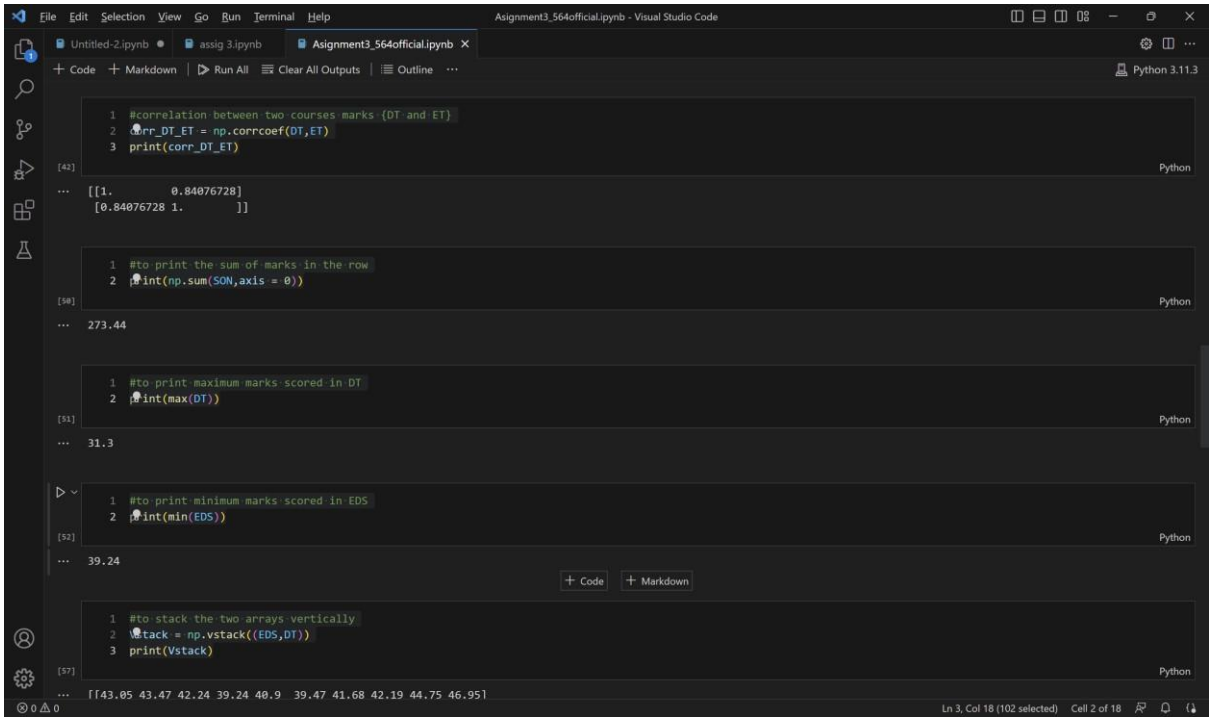
```
print(min(EDS))
#to stack the two arrays vertically Vstack
= np.vstack((EDS,DT)) print(Vstack)
#to copy the transpose of an array in another array  sample_array
= np.fastCopyAndTranspose(Vstack) print(sample_array)
#to check whether the performance of a student is better in one course
compared to 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)
```

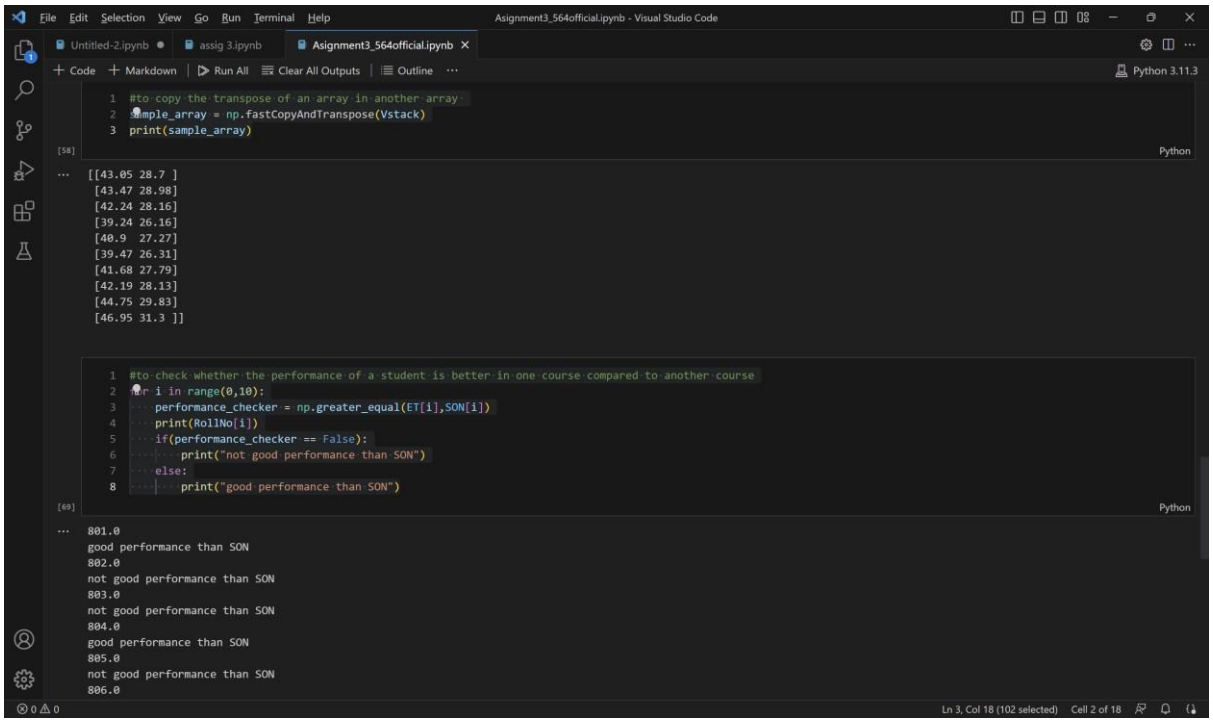
SCREENSHOT:

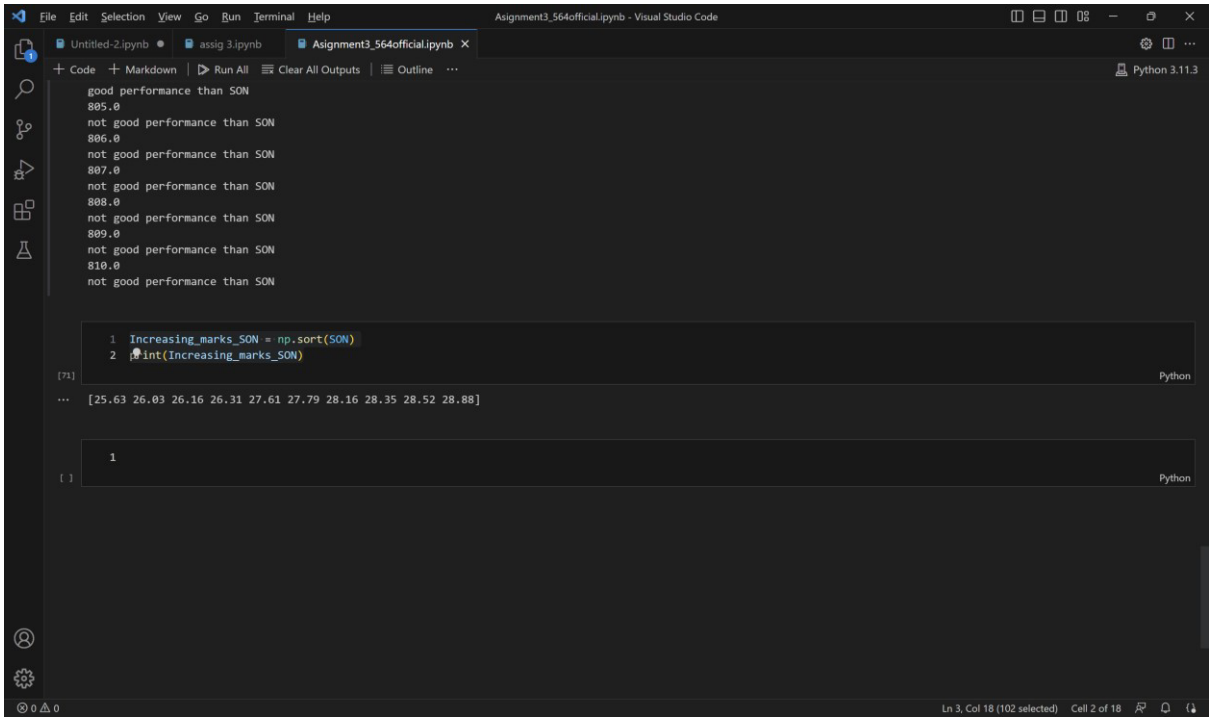



File Edit Selection View Go Run Terminal Help
Asignment3_564official.ipynb - Visual Studio Code
Untitled-2.ipynb
assig 3.ipynb
Asignment3_564official.ipynb
Code Markdown Run All Clear All Outputs Outline
Python 3.11.3
import numpy as np
array1= np.loadtxt('D:\\Assignment_564\\testmarks1.csv',dtype=float,delimiter=',',skiprows = 1)
print(array1)
[29]
Python
[[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]]
array2 = array1.astype(float)
print(array1)
[30]
Python
[[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]]
Code Markdown
RollNo = array1[:,0]
EDS = array1[:,1]
SON = array1[:,2]
DT = array1[:,3]
Ln 2, Col 14 (44 selected) Cell 2 of 18
37°C Sunny
Search
ENG IN
16:58 08-06-2023


File Edit Selection View Go Run Terminal Help
Asignment3_564official.ipynb - Visual Studio Code
Untitled-2.ipynb
assig 3.ipynb
Asignment3_564official.ipynb
Code Markdown Run All Clear All Outputs Outline
Python 3.11.3
RollNo = array1[:,0]
EDS = array1[:,1]
SON = array1[:,2]
DT = array1[:,3]
ET = array1[:,4]
[38]
Python
print(RollNo)
print(EDS)
print(SON)
print(DT)
print(ET)
[39]
Python
[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]
#mean marks of students in EDS course
mean_EDS = np.mean(EDS)
print(mean_EDS)
[47]
Python
42.394
Code Markdown
#standard deviation of marks in SON course
std_deviation_SON = np.std(SON)
print(std_deviation_SON)
[41]
Python
1.1324857614998962
Ln 3, Col 18 (102 selected) Cell 2 of 18
37°C Sunny
Search
ENG IN
16:59 08-06-2023


File Edit Selection View Go Run Terminal Help
Asignment3_564official.ipynb - Visual Studio Code
Untitled-2.ipynb
assig 3.ipynb
Asignment3_564official.ipynb
Code Markdown Run All Clear All Outputs Outline
Python 3.11.3
#correlation between two courses marks (DT and ET)
corr_DT_ET = np.corrcoef(DT,ET)
print(corr_DT_ET)
[42]
Python
[[1. 0.84076728]
[0.84076728 1.]]
#to print the sum of marks in the row
print(np.sum(SON,axis = 0))
[50]
Python
273.44
#to print maximum marks scored in DT
print(max(DT))
[51]
Python
31.3
#to print minimum marks scored in EDS
print(min(EDS))
[52]
Python
39.24
Code Markdown
#to stack the two arrays vertically
Vstack = np.vstack((EDS,DT))
print(Vstack)
[57]
Python
[[43.05 43.47 42.24 39.24 40.9 39.47 41.68 42.19 44.75 46.95]
Ln 3, Col 18 (102 selected) Cell 2 of 18


File Edit Selection View Go Run Terminal Help
Asignment3_564official.ipynb - Visual Studio Code
Untitled-2.ipynb
assig 3.ipynb
Asignment3_564official.ipynb
Code Markdown Run All Clear All Outputs Outline
Python 3.11.3
#to copy the transpose of an array in another array
sample_array = np.fastCopyAndTranspose(Vstack)
print(sample_array)
[58]
Python
[[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]]
#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")
[69]
Python
801.0
good performance than SON
802.0
not good performance than SON
803.0
not good performance than SON
804.0
good performance than SON
805.0
not good performance than SON
806.0
Ln 3, Col 18 (102 selected) Cell 2 of 18


File Edit Selection View Go Run Terminal Help
Asignment3_564official.ipynb - Visual Studio Code
Untitled-2.ipynb
assig 3.ipynb
Asignment3_564official.ipynb
Code Markdown Run All Clear All Outputs Outline
Python 3.11.3
good performance than SON
805.0
not good performance than SON
806.0
not good performance than SON
807.0
not good performance than SON
808.0
not good performance than SON
809.0
not good performance than SON
810.0
not good performance than SON
Increasing_marks_SON = np.sort(SON)
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