

Image-Clustering-Visualisation Project



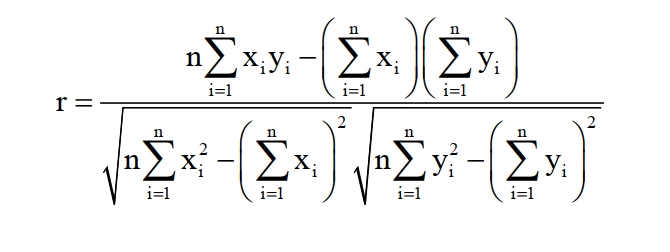
****

Made by Shehryar Ahmad

First, we input the file provided with the IRIS data set and stored the data in an np array.

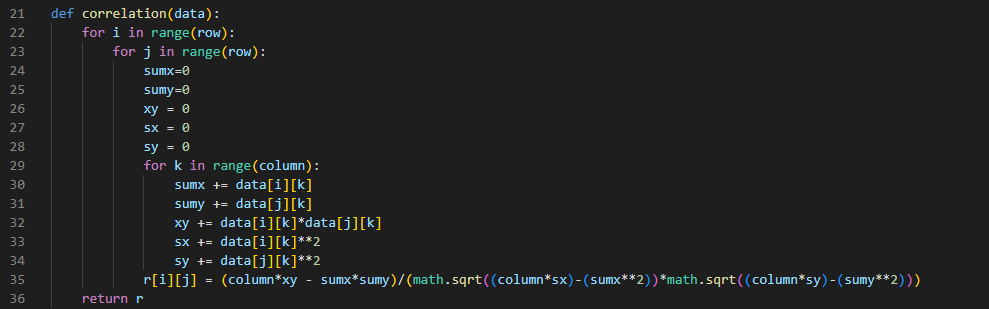
**Task 1:**

In Task-1 we had to find the correlation matrix by using Pearson’s Correlation Coefficient. The Pearson’s Correlation Formula I used in my code is as follows:



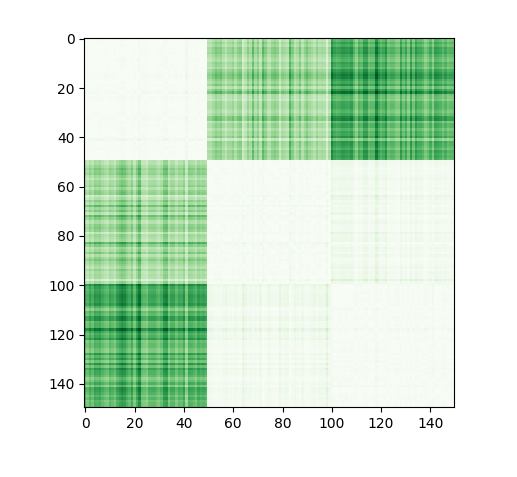
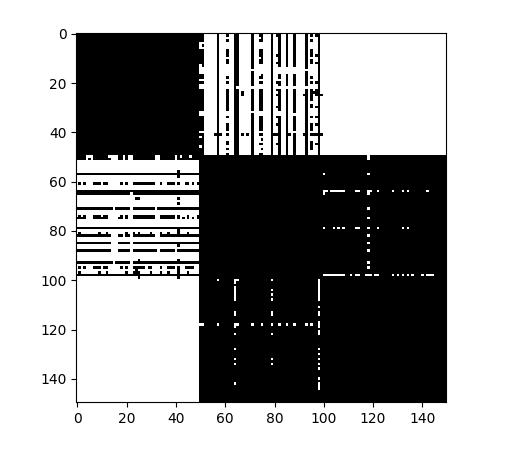
Where x and y are the rows.

This formula is used in code as:



The second step was to Discretize the data. I calculated it by finding the mean of each column and compared it with the whole column. The values in columns above the respective mean are changed to 1 while the rest to 0. Hence, we get our desired discretized matrix.   
I visualized the discretized matrix by converting it to a bitmap using Matplotlib. Now for the color-coded part of the similarity Matrix I did the following steps:

1. For each column I derived the max value in that column.
2. Then I divided the whole column by that value and multiplied it by 255.
3. The resulting values ranged between 0 and 255.
4. Then I used these values to shade the pixels.



Color Coded Image of Similarity matrix

Bitmap of discretized matrix

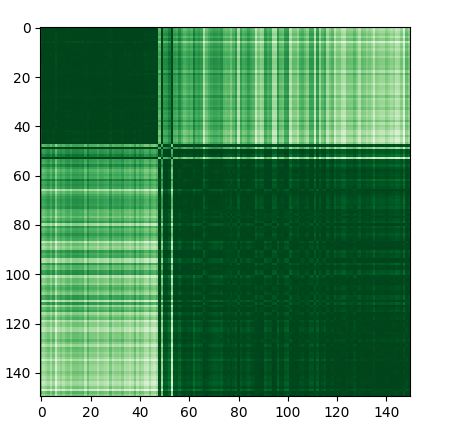
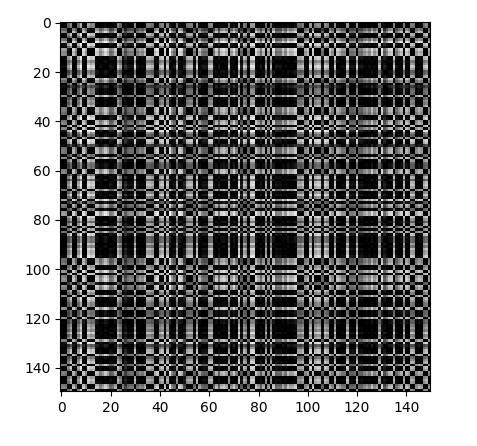
**Task 2:**

In this Task we had to first permute the data matrix. This was done by shuffling the individual rows in the data sheet. We then showed the color-coded image of the permuted data matrix. Now we used the Signature technique to recover the image clusters.

Signature Method includes:

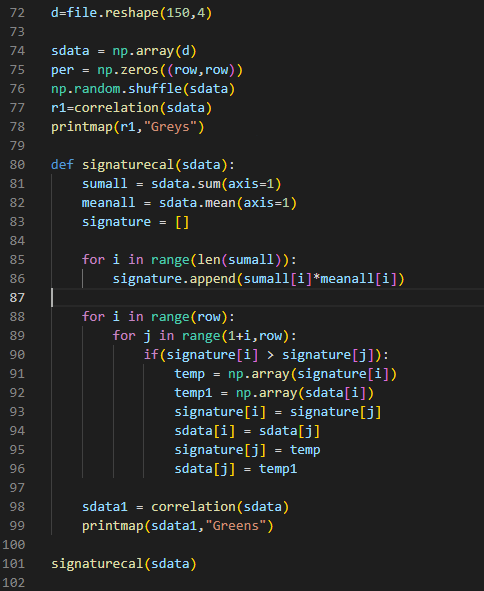
1. Summing all the values in a row.
2. Calculating mean of the row.
3. Multiplying the sum of the row with the mean.
4. This produces the signature of the respective row.

Now we sort the similarity matrix by the signature value of each row. Now we again apply task 1 to the Sorted Matrix that is applying correlation coefficient followed by calculating the mean and discretizing.

The code and output pictures are attached.

Bitmap of iris data Correlation Matrix after Signature Generation and Arrangement

Bitmap of IRIS data after Permutation



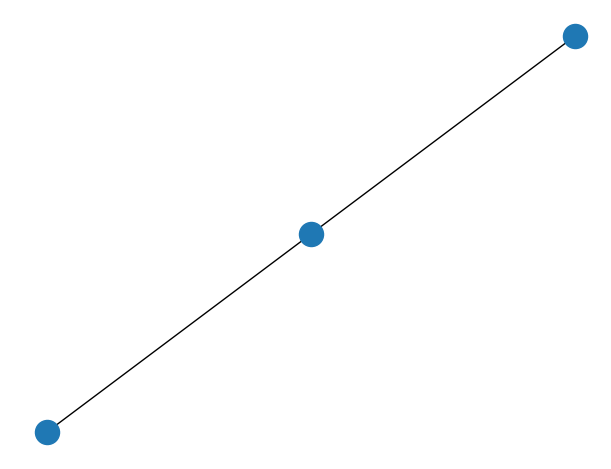
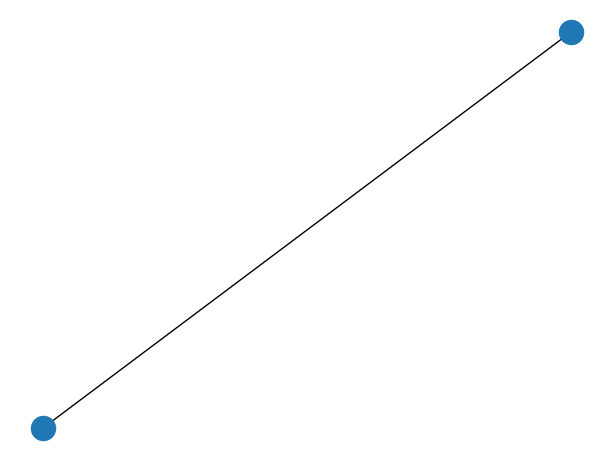
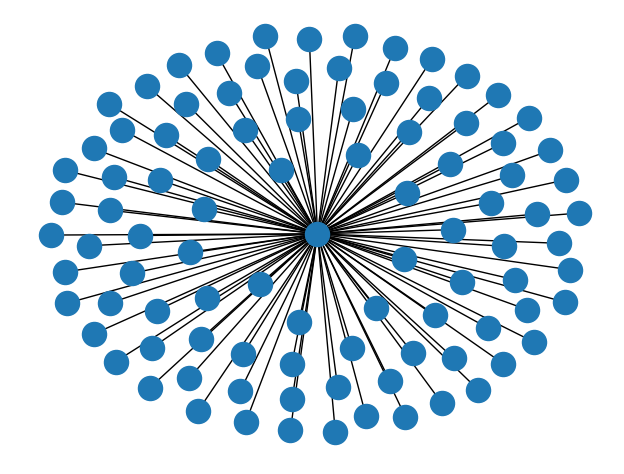
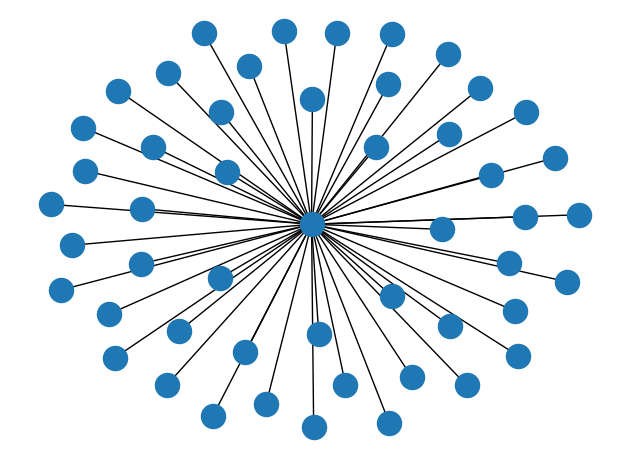
Driver Code for Task 2

**Task 3:**

This task included creating a weighted graph for the permuted data set. The steps taken were as follows:

1. Calculating the correlation matrix and considering it as a graph.
2. Removing weights that were below a certain threshold, which was provided by the user.
3. Finding the node with the highest weight and making its neighboring nodes as cluster.
4. Repeating this process until no clusters are being left.

When threshold is equal to 0.98, 4 clusters are obtained. Visualization and code of this process is attached:

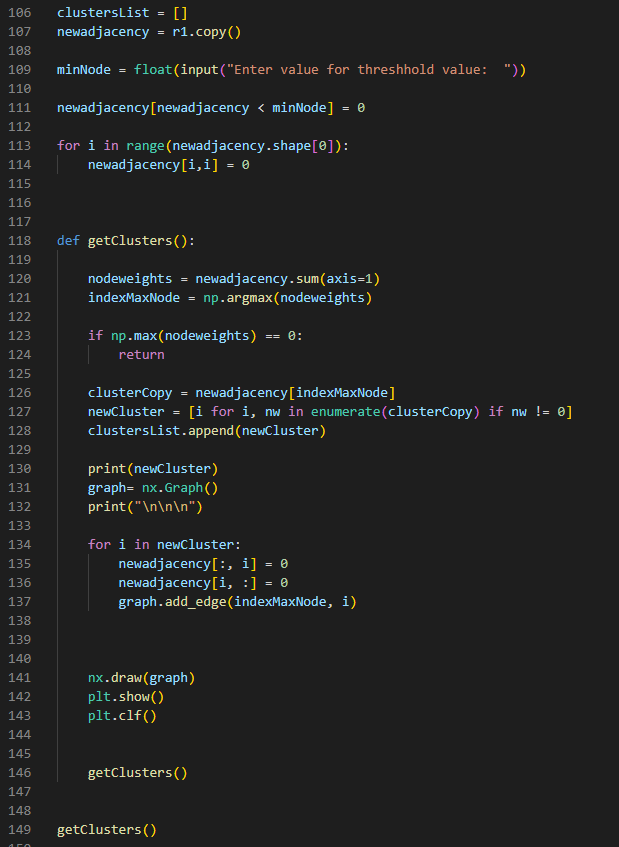


Cluster 4

Cluster 3

Cluster 2

Cluster 1



Driver Code for Task 3

**Comparison between Task 2 and Task 3:**

Task 2 and Task 3 both include the recovery of image but just the process we’re doing is different. In Task 2 we are implementing Signature Technique whose details is already given in Task 2, while in Task 3 we’re implementing Clustering technique whose details is given in Task 3.  
This is the main difference between the two tasks.

**Work Distribution:**

This project is wholly and solely made by Shehryar Ahmad.