Homework 3 **ISYE 6740**

Shrikanth Mahale (smahale6)

1. Order of faces using ISOMAP (50 points)

The objective of this question is to reproduce the ISOMAP algorithm results that we have seen discussed in lecture as an excercise. The file isomap.mat (or isomap.dat) contains 698 images, corresponding to different poses of the same face. Each image is given as a 64 _ 64 luminosity maps, hence represented as a vector in R_{4096}. This vector is stored as a row in the file. [This is one of the datasets used in the original paper for ISOMAP, J.B. Tenenbaum, V. de Silva, and J.C. Langford, Science 290 (2000) 2319-2323.]

Q1a: (20 points) Choose the Euclidean distance between images (i.e., in this case a distance in R4096). Construct a similarity graph with vertices corresponding to the images, and tune the threshold so that each node has at least 100 neighbors. Visualize the similarity graph (e.g., plot the adjacency matrix, or visualize the graph and illustrate a few images corresponds to nodes at different parts of the graph; you can be a bit creative here).

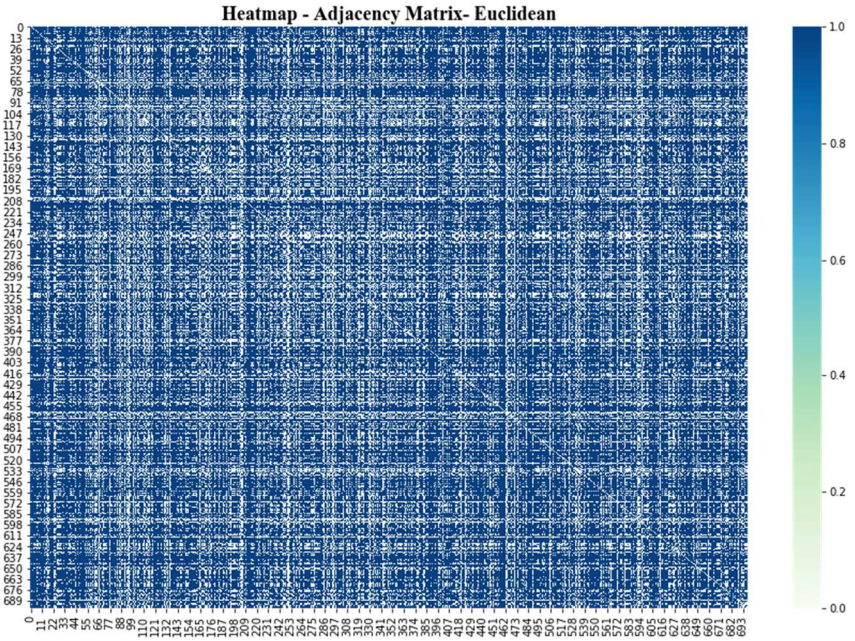
Solution 1(a)

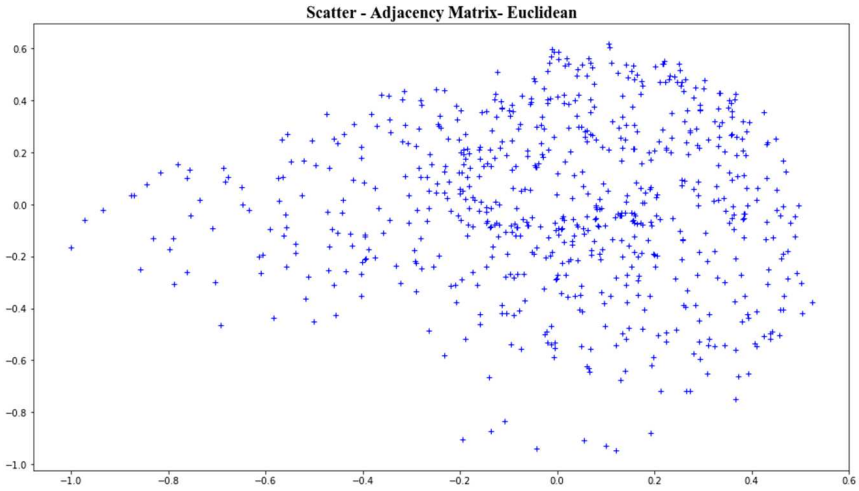
Isomap.mat file was used. The file was populated into a dictionary using the scipy.io module and the images key was used. There were 698 images with 4096 dimensions making it into 698*4096 matrix.

As requested, a Distance matrix D was created by calculating the Euclidean distances between the images and this distance matrix D was used to calculate the Adjacency Matrix A.

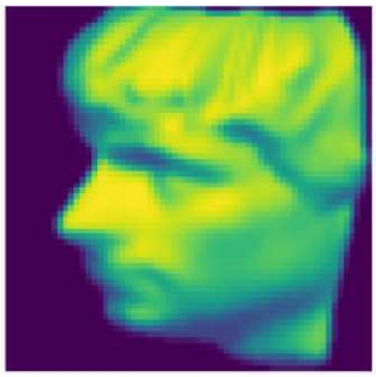
The goal was to calculate the lowest value of ϵ such that each node has at least 100 neighbors. A for loop was used to calculate the lowest value of ϵ and it was found that for ϵ**= 23**, each node had had at least 100 neighbors (an image with lowest number of neighbors was 151 neighbors for Epsilon = 23).

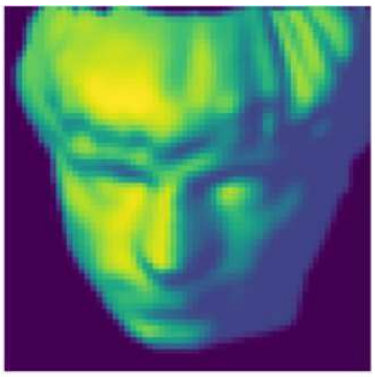
As requested, a similarity graph in form of heatmap and scatter plot was constructed using the Adjacency Matric. The heatmap is shown below

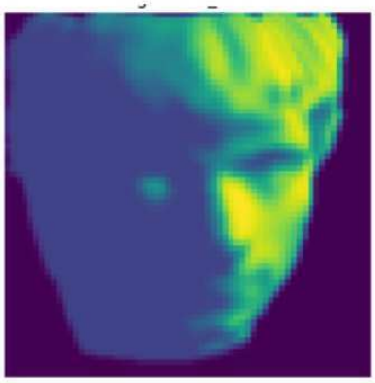




Three images were selected from different corners and visualized. The pictures are visually very different from each other as shown below.







Q1b: (20 points) Implement the ISOMAP algorithm and apply it to this graph to obtain a d = 2-dimensional embedding. Present a plot of this embedding. Find three points that are close to each other in the embedding space, and show what they look like.
Do you see any visual similarity among them?

Solution 1(b)

In order to implement ISOMAP algorithm, the distance matrix was used in the calculation of Centering Matrix H using the formula below

$$H = I - \frac{1}{m} 1 1^{\top}$$

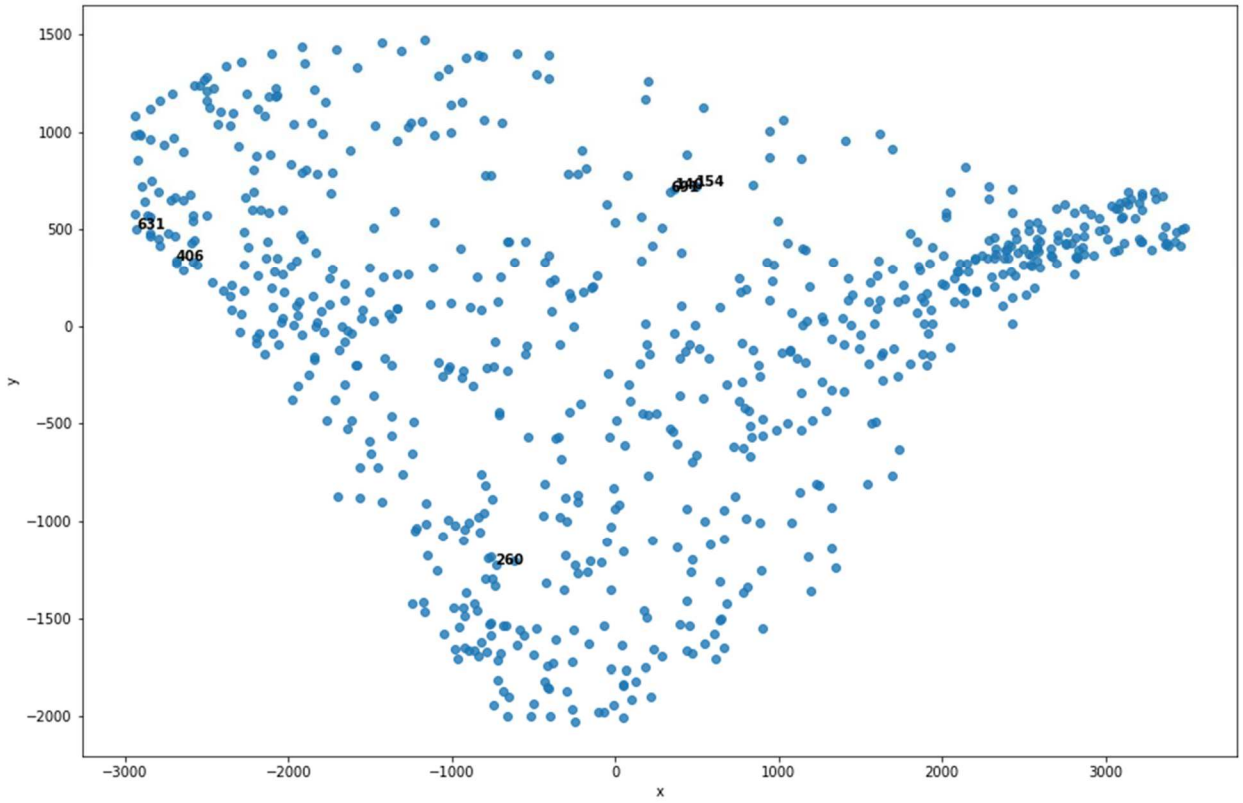
Here value of m = 698 since there are 698 images.

After calculation of H, C matrix was calculated using the centering matrix H using the formula below

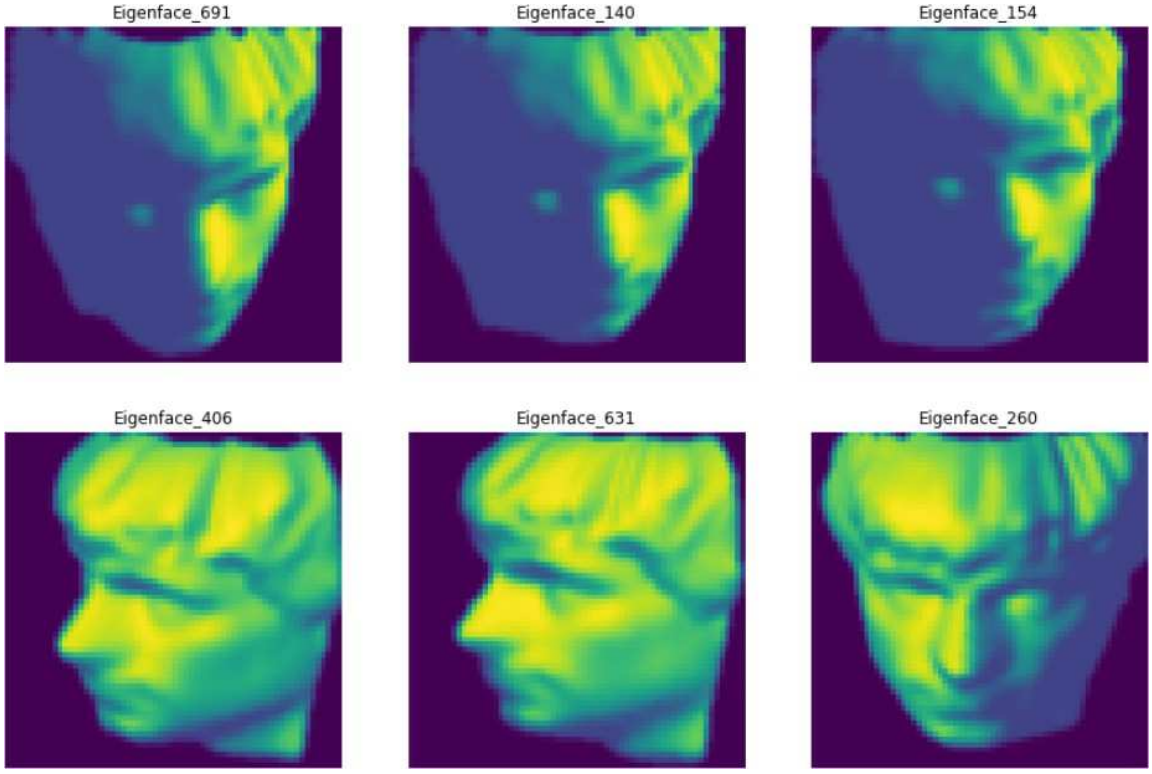
$$C = -\frac{1}{2m} H(D)^2 H$$

Leading Eigen values and eigen vectors were derived and sorted. 2 leading eigen vectors of used for 2 Dimensional embedding

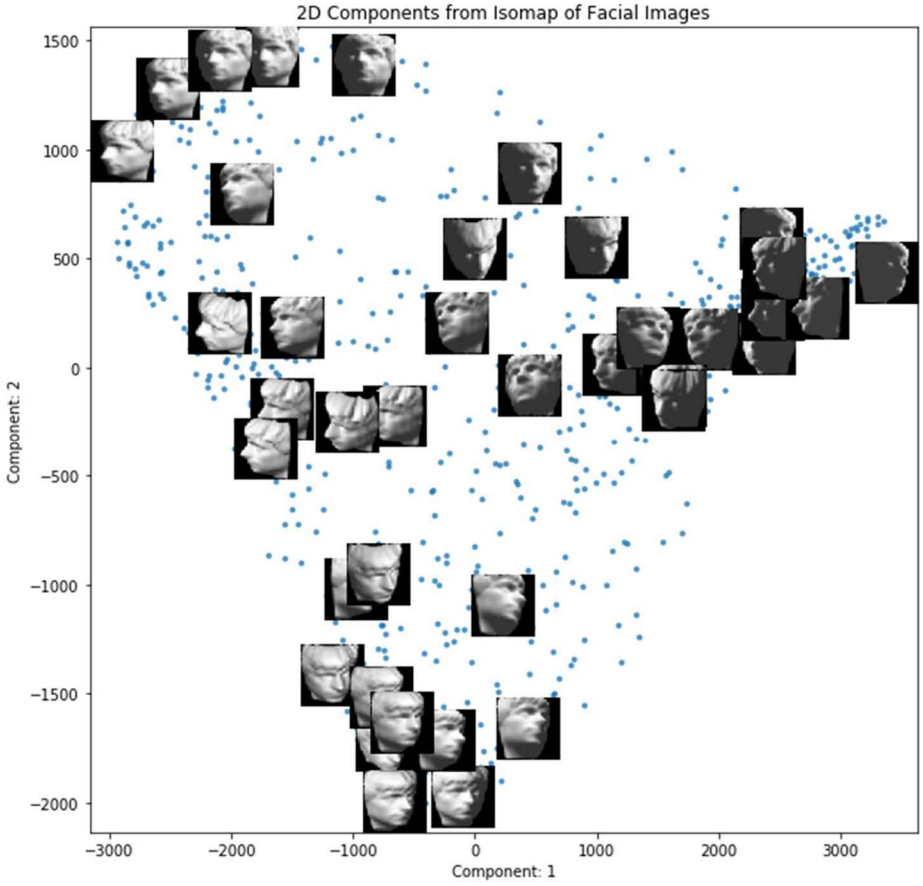
Isomap was plotted with labels for selected 6 points in 2 sets that are close to each other



The images are visually very similar. Image 75 being a bit from a higher elevation from Image 363 and 530. Similarly images 406 and 631 are similar.



The ISOMAP was plotted and the images were also plotted on the graph as shown below.



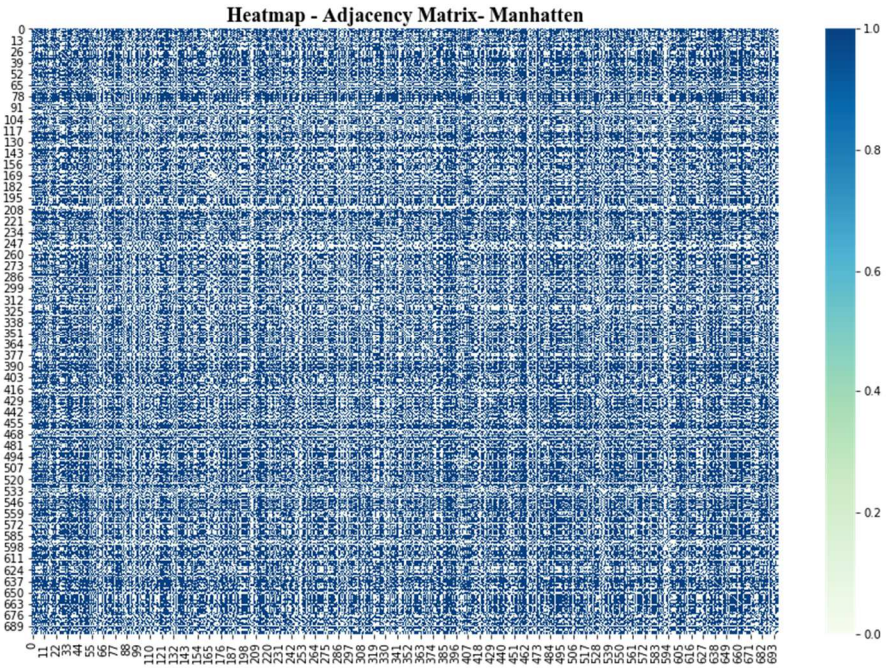
Q1c: (10 points) Now choose `1 distance (or Manhattan distance) between images (recall the definition from \Clustering" lecture)). Repeat the steps above. Again construct a similarity graph with vertices corresponding to the images, and tune the threshold so that each node has at least 100 neighbors. Implement the ISOMAP algorithm andapply it to this graph to obtain a d = 2-dimensional embedding. Present a plot of this embedding. Do you see any difference by choosing a difference similarity measure?

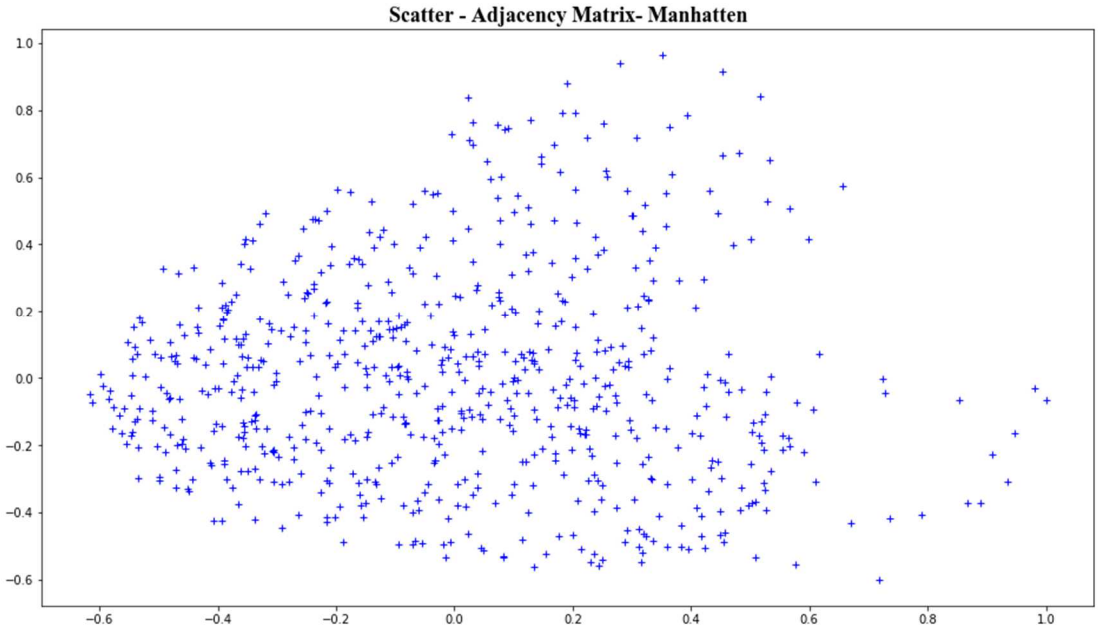
Solution 1(c)

Same steps of Solution 1a and Solution 1b were repeated, except Manhattan distance was used between images instead of Euclidean distance. Given below are the images

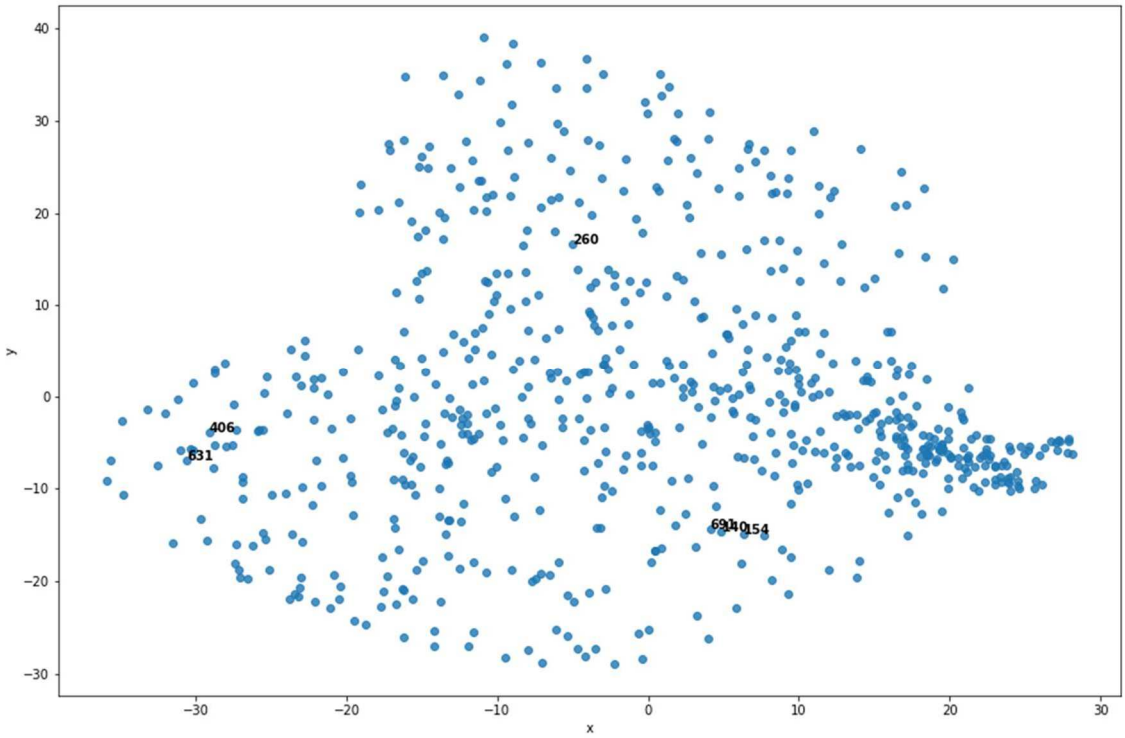
The goal was to calculate the lowest value of $\in$ such that each node has at least 100 neighbors. A for loop was used to calculate the lowest value of $\in$ and it was found that for $\in$**= 1010**, each node had had at least 100 neighbors.

Similarity Graph

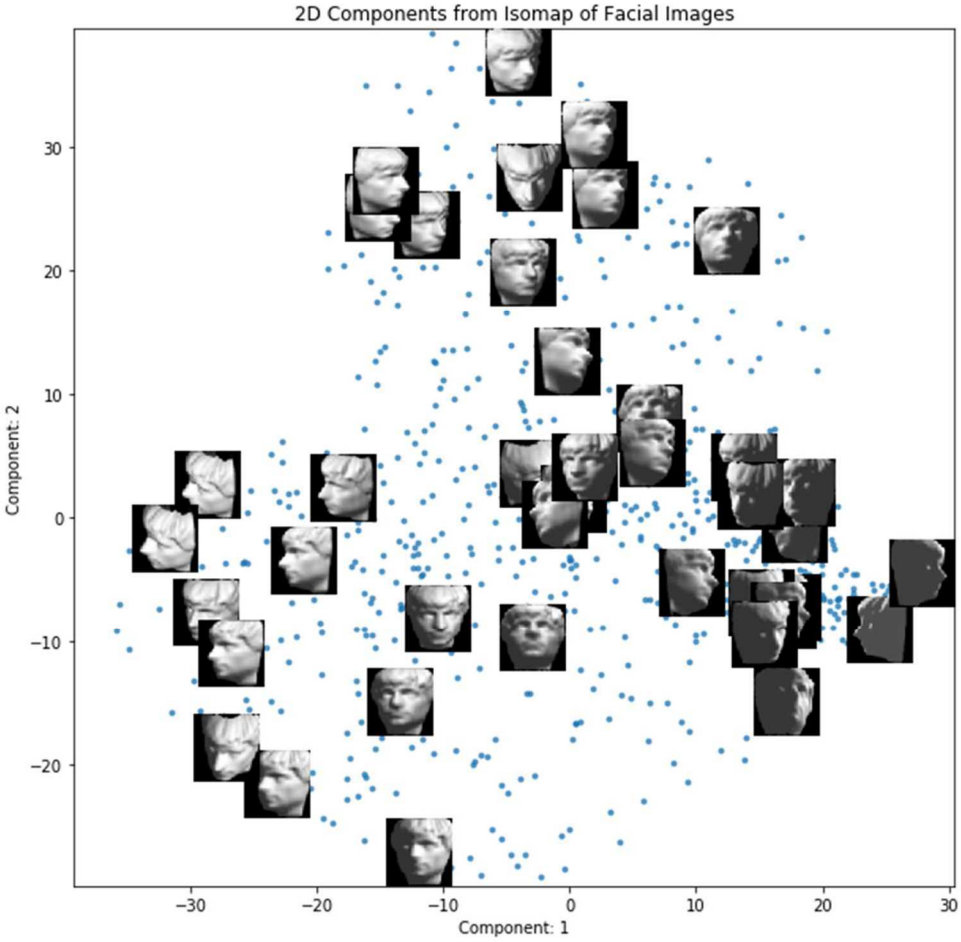


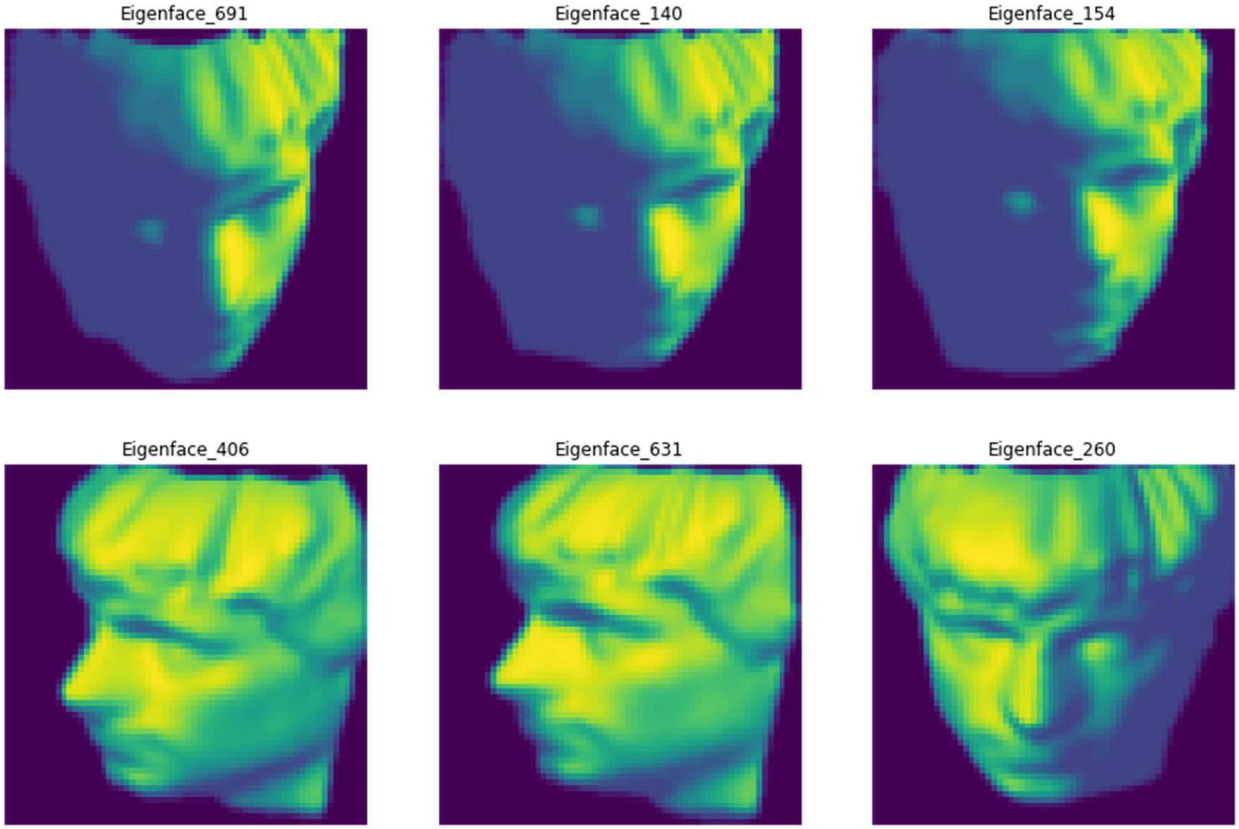


ISOMAP highlighting selected images



The ISOMAP was plotted and the images were also plotted on the graph as shown below.





The distribution is different in this case, but the final output is again very visually similar. Image 140 and 154 are exactly similar

2. Density estimation: Psychological experiments. (50 points)

Q2a: The data set n90pol.csv contains information on 90 university students who participated in a psychological experiment designed to look for relationships between the size of different regions of the brain and political views. The variables amygdala and acc indicate the volume of two particular brain regions known to be involved in emotions and decision-making, the amygdala and the anterior cingulate cortex; more exactly, these are residuals from the predicted volume, after adjusting for height, sex, and similar body-type variables. The variable orientation gives the students' locations on a five-point scale from 1 (very conservative) to 5 (very liberal).

Solution 2(a)

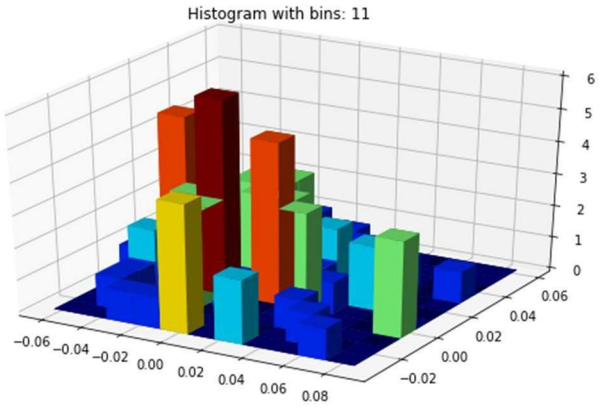
For the calculation of the number of bins, the general formula below was used assuming the underlying density being estimated is Gaussian.

General Formula (aka Freedman–Diaconis):

$$h = 2\mathrm{IQR}(x)N^{-1/3}$$

Where h is the bandwidth that minimizes the mean integrated squared error
You then calculate the number of bins M along each dimension as being equal to: $M = \lceil (\max(x) - \min(x)) / h \rceil$

Using the formula above, the optimal value of number of Bins was found to be 11.



Q2b:

(20 points) Now implement kernel-density-estimation (KDE) to estimate the 2-dimensional with a two-dimensional density function of (amygdala, acc). Use a simple multi-dimensional Gaussian kernel, for

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^2,$$

where x_1 and x_2 are the two dimensions respectively

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x_1^2 + x_2^2)}{2}}.$$

Recall in this case, the kernel density estimator (KDE) for a density is given by

$$p(x) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{h} K\left(\frac{x^i - x}{h}\right),$$

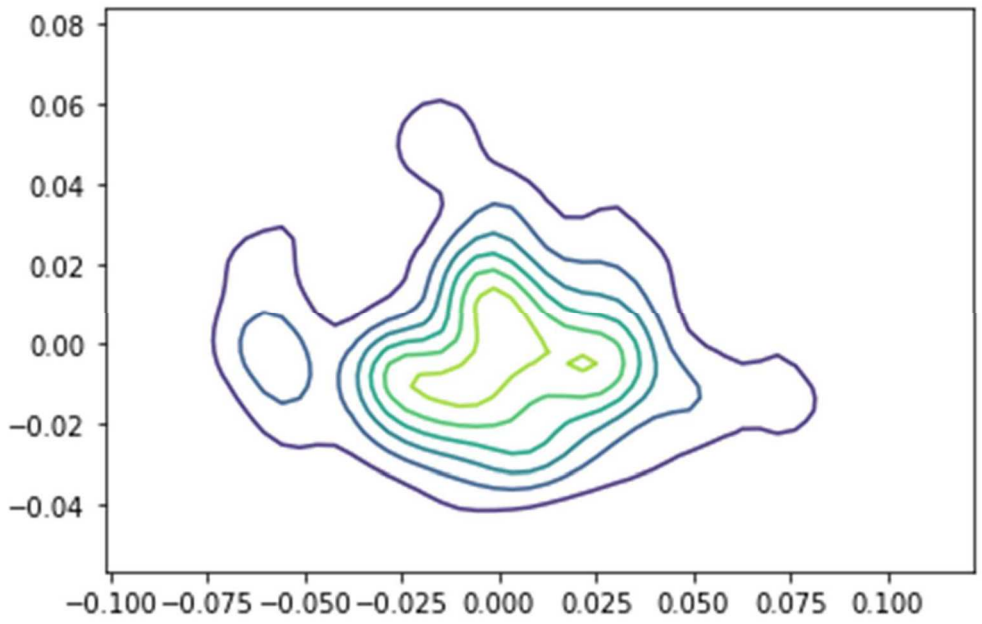
where x^i are two-dimensional vectors, $h > 0$ is the kernel bandwidth. Set an appropriate h so you can see the shape of the distribution clearly. Plot of contour plot (like the ones in slides) for your estimated density.

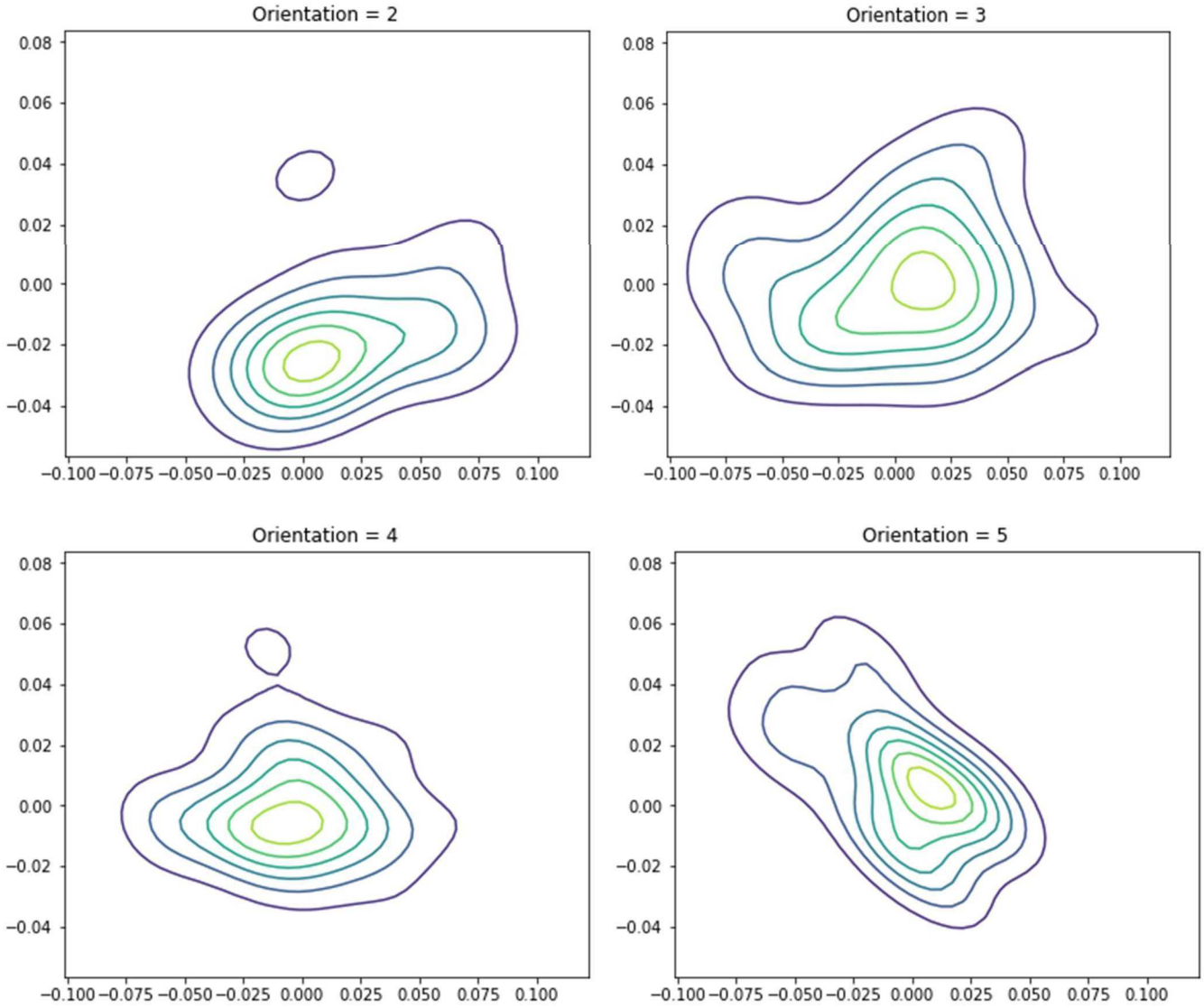
Solution 2(b)

The gaussian kernel $K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x^2+x_2^2)}{2}}$ is calculated and then the KDE is calculated using $e(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{h} K\left(\frac{s^i - s}{h}\right)$

As requested, the Kernel Density Estimation function was implemented using the Gaussian Kernel for amydgala and acc.

Contours have been plotted with different orientations shown below.




Orientation = 2
Orientation = 3
Orientation = 4
Orientation = 5
0.08
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.100 −0.075 −0.050 −0.025 0.000 0.025 0.050 0.075 0.100

Q2c: (10 points) Plot the condition distribution of the volume of the amygdala as a function of political orientation: p(amygdala | orientation = a), a = 1; : : : ; 5. Do the same for the volume of the acc. Plot p(acc | orientation = a), a = 1; : : : ; 5. You may either use histogram or KDE to achieve the goal.

Solution 2(c)

Line plots were plotted for the 2 variables amydgala and acc for multiple orientations using seaborn kde function.