20W-COMSCIM146 ps4

JACOB KAUFMAN

TOTAL POINTS

32 / 32

QUESTION 1

PCA and Image Reconstruction 4 pts

1.1 average face 1 / 1

✓ **- 0 pts Correct**

- 0.5 pts Need to plot average face

- 1 pts Cannot find solution

1.2 eigenfaces 1 / 1

✓ **- 0 pts Correct**

- 0.5 pts Need to plot eigenfaces

- 1 pts Could not find solution

1.3 facial reconstruction 2 / 2

✓ **- 0 pts Correct**

- 1 pts Need to also plot eigenfaces

- 2 pts Could not find solution

QUESTION 2

K-means and K-medoid 16 pts

2.1 a. J(c, \mu, k) 4 / 4

✓ **- 0 pts Correct/minor problem**

- 4 pts No Ans

- 2 pts this answer also asks the values of J and others?

- 1 pts What is the min values of J

2.2 d. plot s 4 / 4

✓ **- 0 pts Correct**

- 4 pts No Ans

- 2 pts Looks very wrong

2.3 e. plots 4 / 4

✓ **- 0 pts Correct**

- 2 pts looks very wrong

- 4 pts No Ans

2.4 f. plots 4 / 4

✓ **- 0 pts Correct**

- 2 pts Looks very wrong

- 4 pts No Ans

- 1 pts looks wrong with the cheat sheet

QUESTION 3

12 pts

3.1 a. comparison 4 / 4

✓ **- 0 pts Correct**

- 1 pts kmeans scores too low

- 1 pts k-medoids scores too low

- 4 pts not attempted / not found

- 2 pts k-medoids scores inaccurate

3.2 b. plot clustering score vs. l. 4 / 4

✓ **- 0 pts Correct**

- 1 pts performance of k-means abnormal

- 1 pts performance of k-medoids abnormal

- 4 pts not attempted / not found

3.3 c. find pairs 4 / 4

✓ **- 0 pts Correct**

- 4 pts Not attempted / not found

gradescope

CM146, Winter 2020
Problem Set 2: SVM and Kernels
Due March 1, 2020 at 11:59 pm

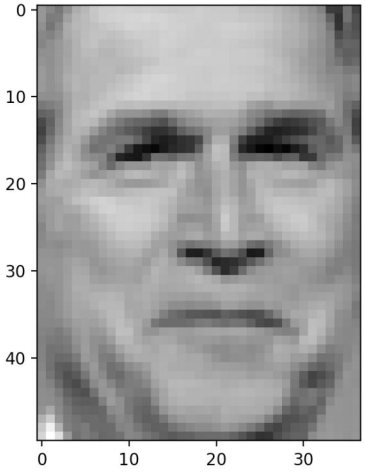
Jacob Kaufman

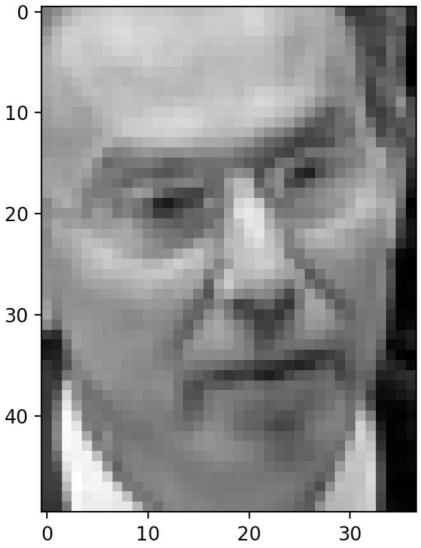
03/01/2020

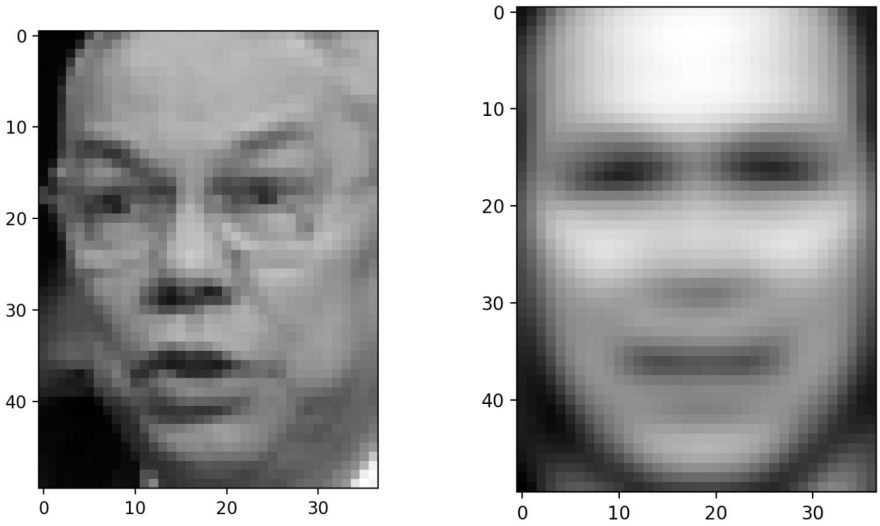
1 PCA and Image Reconstruction [4pts]

Solution:

(a) The three faces, along with their average, are shown below.







The average face is a blurry face with few to none discerning features. It is clear to represent a human face though, as it shows clear eyes, nose, lips, cheeks, and head shape.

(b) The top twelve eigenfaces are shown below.

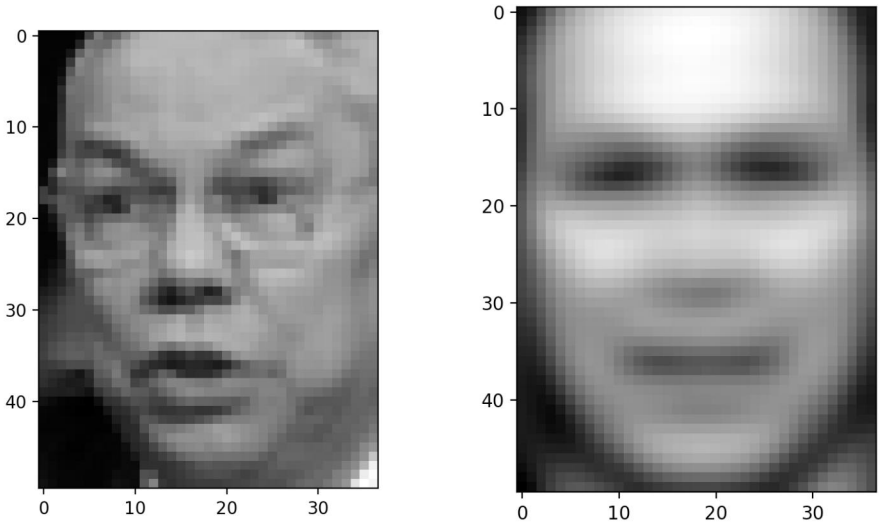
1.1 average face 1 / 1

✓ **- 0 pts** **Correct**

- 0.5 pts Need to plot average face

- 1 pts Cannot find solution

gradescope



The average face is a blurry face with few to none discerning features. It is clear to represent a human face though, as it shows clear eyes, nose, lips, cheeks, and head shape.

(b) The top twelve eigenfaces are shown below.

These eigenfaces capture the maximum variance in the dataset, in variables such as lighting. These are selected as the top eigenfaces because they most resemble the real faces after the number of features is reduced.

(c) The twelve faces' reconstructions are shown below, for $l = 1, 10, 50, 100, 500, 1288$:

1.2 eigenfaces 1 / 1

✓ **- 0 pts** **Correct**

- 0.5 pts Need to plot eigenfaces

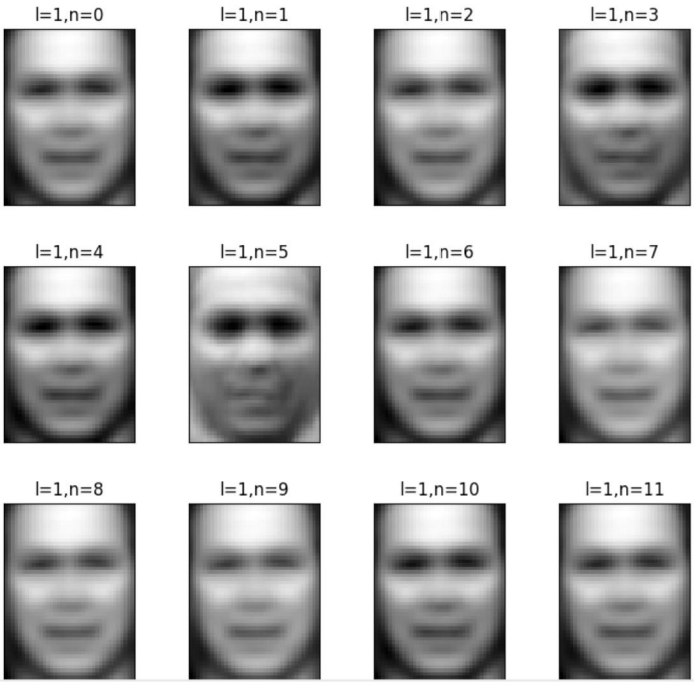
- 1 pts Could not find solution

gradescope

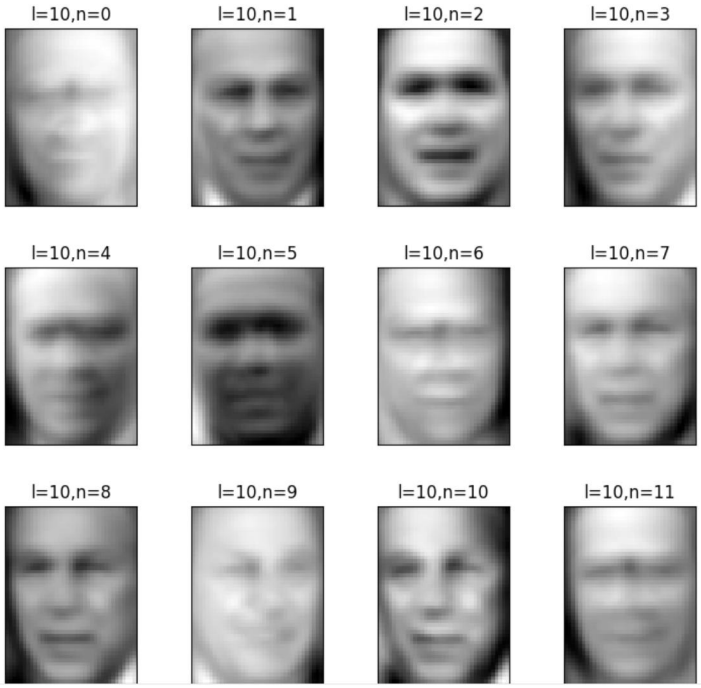
These eigenfaces capture the maximum variance in the dataset, in variables such as lighting. These are selected as the top eigenfaces because they most resemble the real faces after the number of features is reduced.

(c) The twelve faces' reconstructions are shown below, for $l = 1, 10, 50, 100, 500, 1288$:

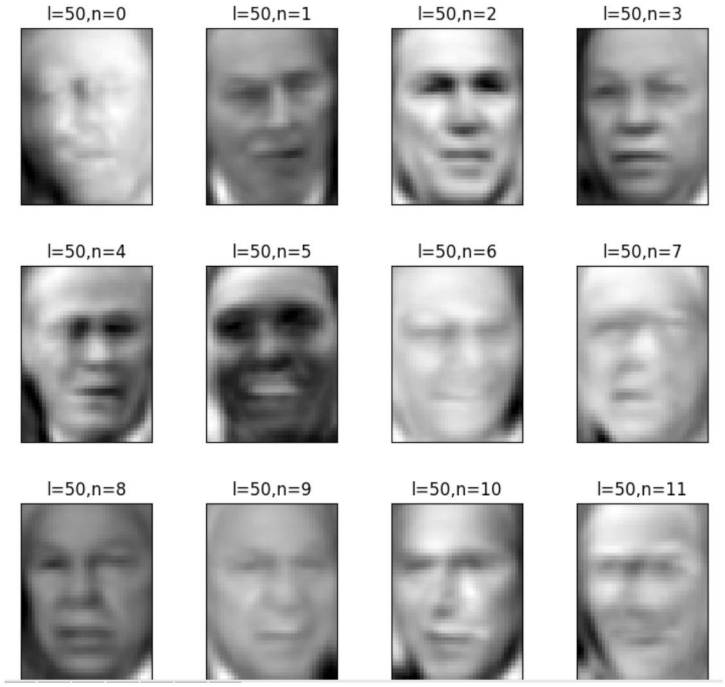
$l = 1$


l=1,n=0
l=1,n=1
l=1,n=2
l=1,n=3
l=1,n=4
l=1,n=5
l=1,n=6
l=1,n=7
l=1,n=8
l=1,n=9
l=1,n=10
l=1,n=11

$l = 10$


l=10,n=0
l=10,n=1
l=10,n=2
l=10,n=3
l=10,n=4
l=10,n=5
l=10,n=6
l=10,n=7
l=10,n=8
l=10,n=9
l=10,n=10
l=10,n=11

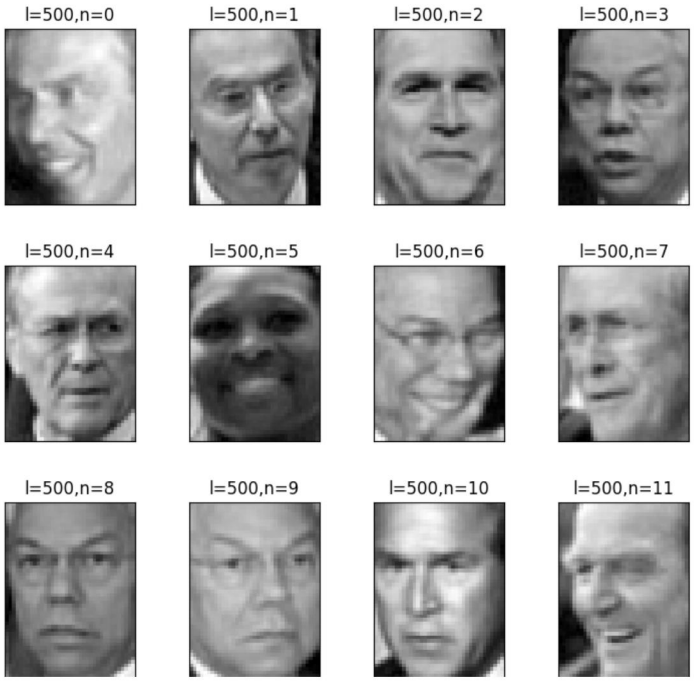
$l = 50$


l=50,n=0
l=50,n=1
l=50,n=2
l=50,n=3
l=50,n=4
l=50,n=5
l=50,n=6
l=50,n=7
l=50,n=8
l=50,n=9
l=50,n=10
l=50,n=11

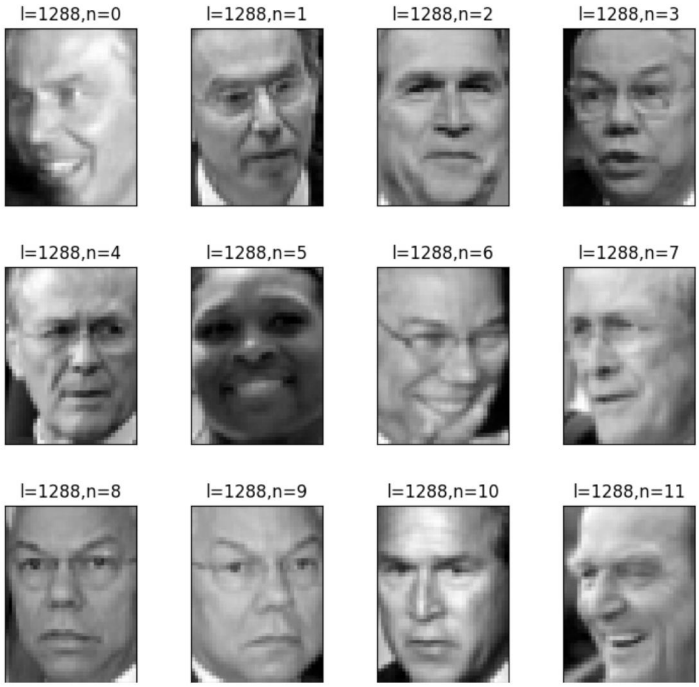
$l = 100$


l=100,n=0
l=100,n=1
l=100,n=2
l=100,n=3
l=100,n=4
l=100,n=5
l=100,n=6
l=100,n=7
l=100,n=8
l=100,n=9
l=100,n=10
l=100,n=11

$l = 500$


l=500,n=0
l=500,n=1
l=500,n=2
l=500,n=3
l=500,n=4
l=500,n=5
l=500,n=6
l=500,n=7
l=500,n=8
l=500,n=9
l=500,n=10
l=500,n=11

$l = 1288$



The photos seem to get much less blurry as the l goes up. This is because the faces are being reconstructed with more features as l increases. The faces become slightly distinguishable at $l = 100$ and very clear at $l = 500$.

2 K-Means and K-Medioids [16pts]

Solution:

(a) Minimizing the objective function over μ, c, and k is a bad idea because the objective function will always minimize to $J = 0$, with values for the corresponding variables as $\mu^*_{(i)} = x^{(i)}$, $k_* =$ #training examples, $c^* =$ some unique value for i. In other words, each training example will become its own cluster and there will be as many as means as training examples.

(b) Implemented in `cluster.py`

(c) Implemented in `faces.py`

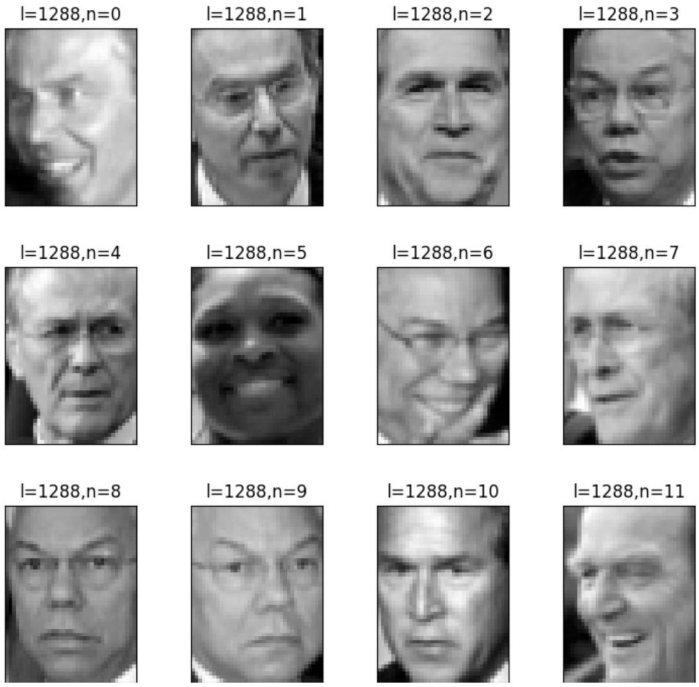
1.3 facial reconstruction **2 / 2**

✓ **- 0 pts** **Correct**

- 1 pts Need to also plot eigenfaces

- 2 pts Could not find solution

gradescope



The photos seem to get much less blurry as the l goes up. This is because the faces are being reconstructed with more features as l increases. The faces become slightly distinguishable at $l = 100$ and very clear at $l = 500$.

2 K-Means and K-Medioids [16pts]

Solution:

(a) Minimizing the objective function over μ, c, and k is a bad idea because the objective function will always minimize to $J = 0$, with values for the corresponding variables as $\mu^*_{(i)} = x^{(i)}$, $k_* =$ #training examples, $c^* =$ some unique value for i. In other words, each training example will become its own cluster and there will be as many as means as training examples.

(b) Implemented in `cluster.py`

(c) Implemented in `faces.py`

2.1 a. J(c, \mu, k) **4 / 4**

✓ **- 0 pts** **Correct/minor problem**

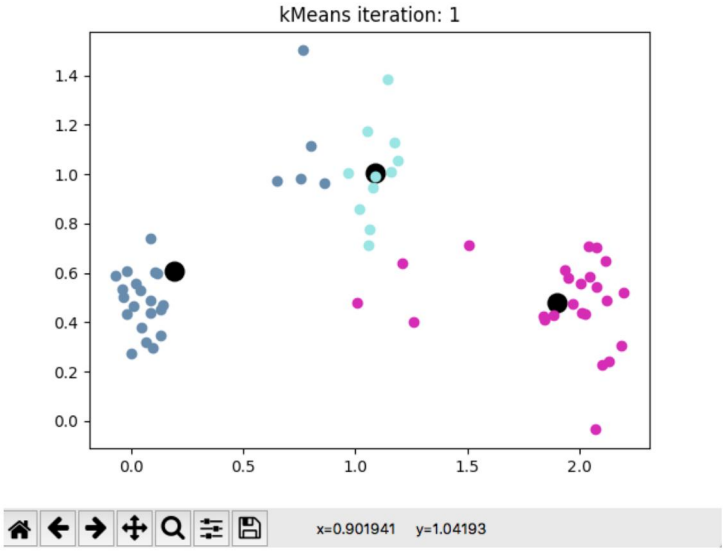
- 4 pts No Ans

- 2 pts this answer also asks the values of J and others?

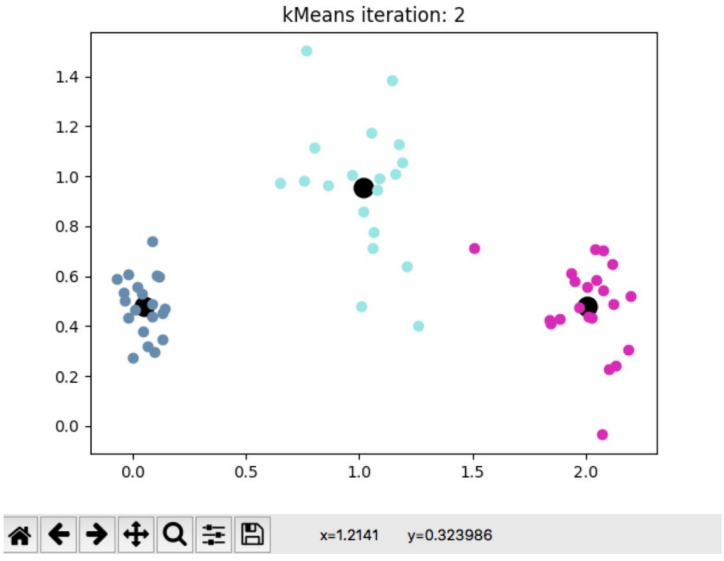
- 1 pts What is the min values of J

gradescope

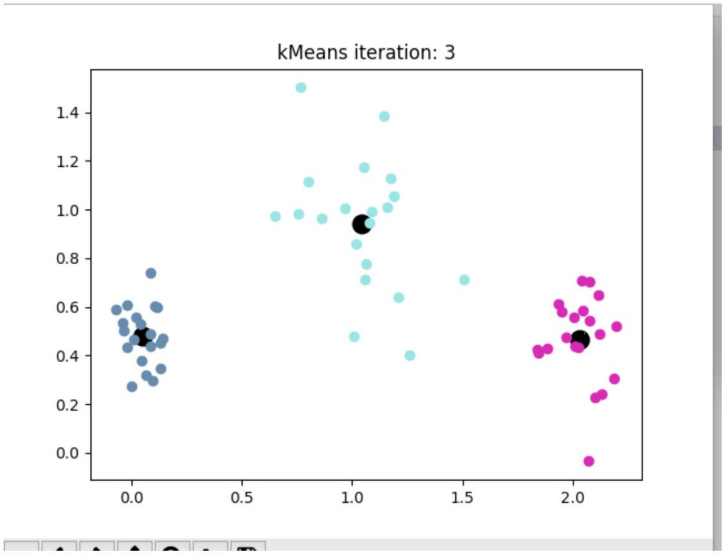
(d) The three clusters are assigned three separate colors. The centroids are shown as black dots.
kMeans Iteration 1:



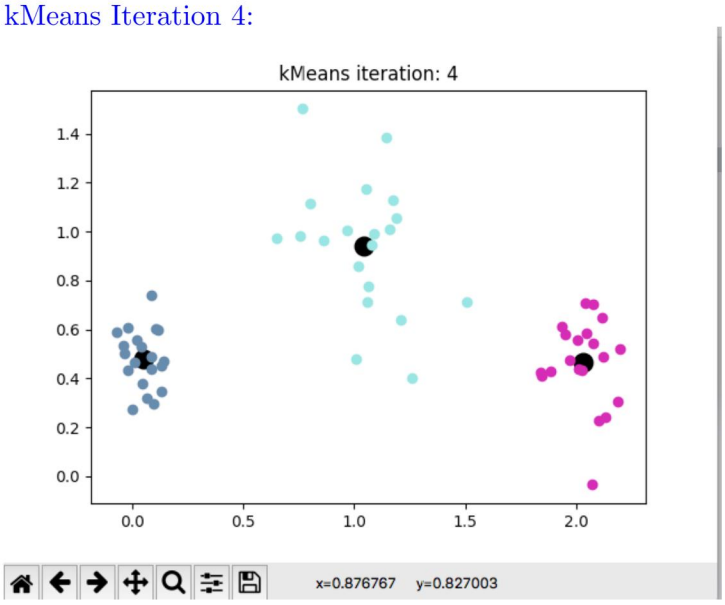
kMeans Iteration 2:



kMeans Iteration 3:



kMeans Iteration 4:



(e) The three clusters are assigned three separate colors.
kMedoids Iteration 1:

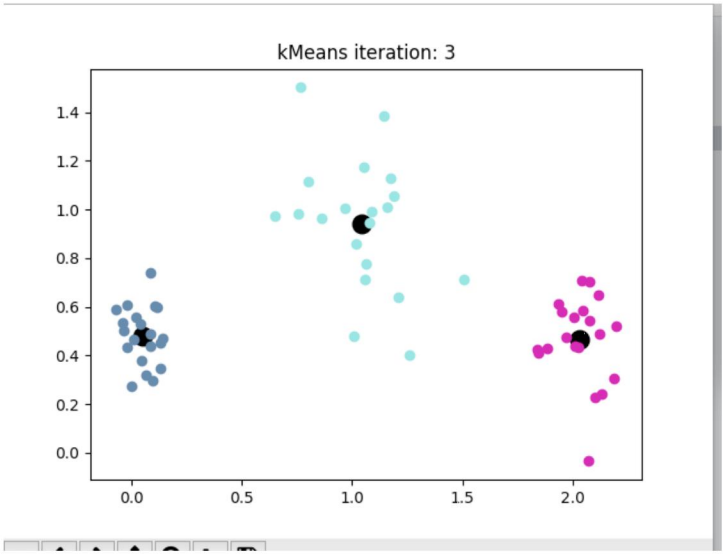
2.2 d. plot s **4 / 4**

✓ **- 0 pts** **Correct**

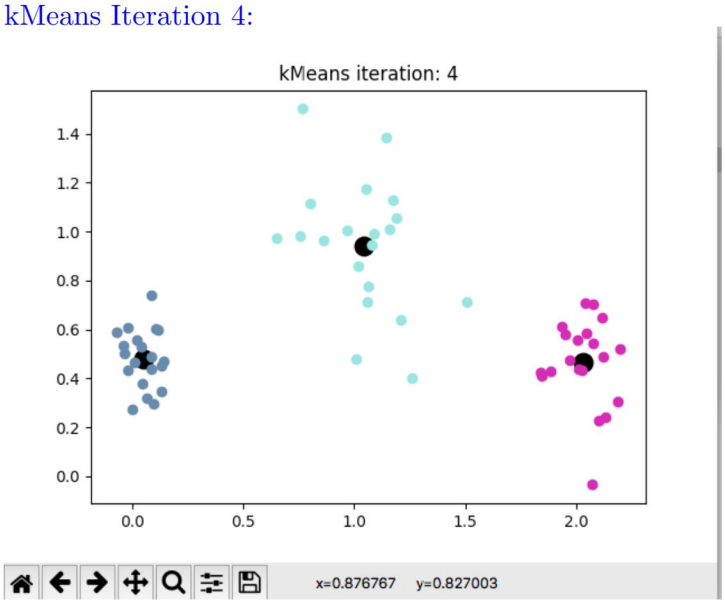
- 4 pts No Ans

- 2 pts Looks very wrong

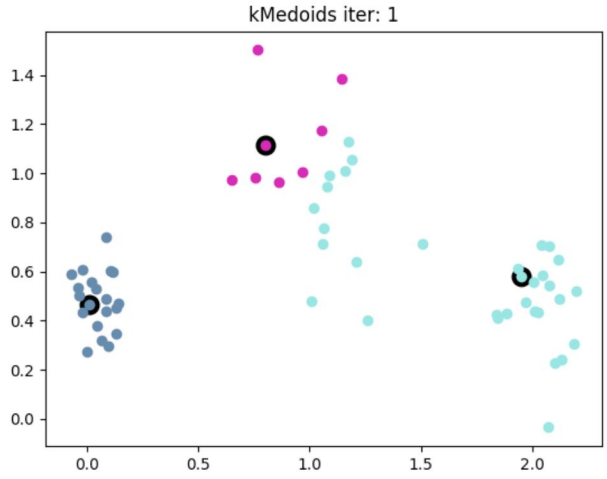
gradescope



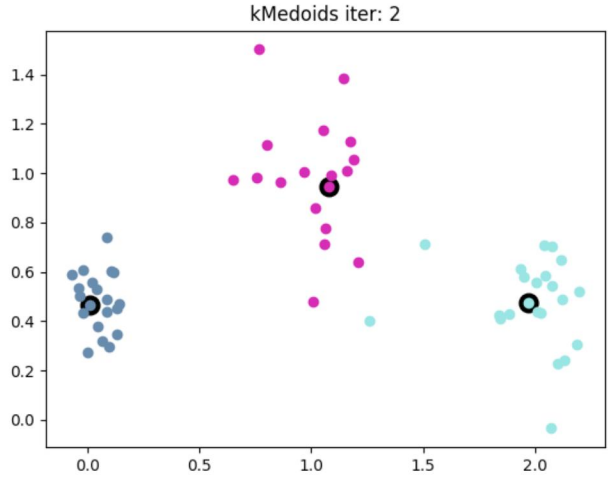
kMeans Iteration 4:



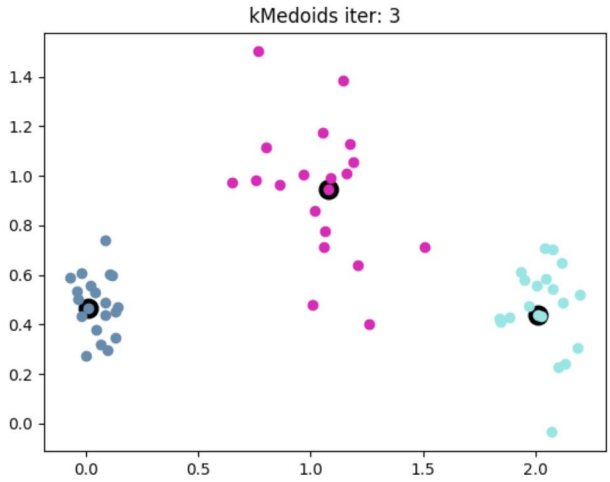
(e) The three clusters are assigned three separate colors.
kMedoids Iteration 1:



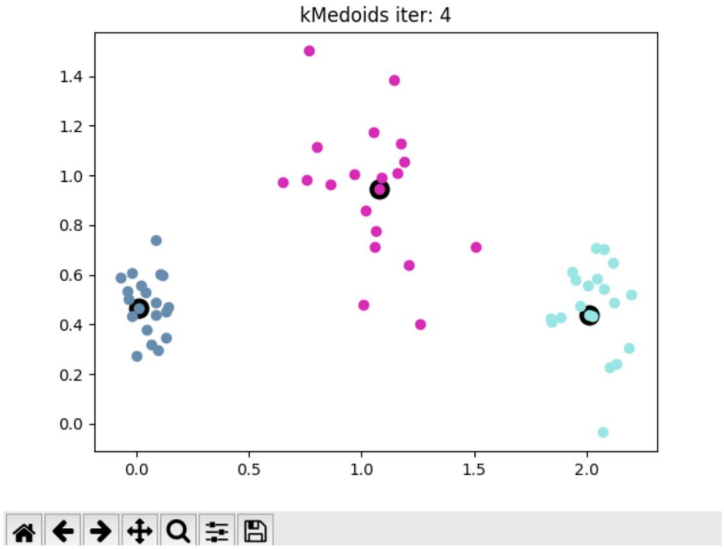
kMedoids Iteration 2:



kMedoids Iteration 3:



kMedoids Iteration 4:



(f) Clustering using `cheat_init` is shown below.

kMeans Iteration 1:

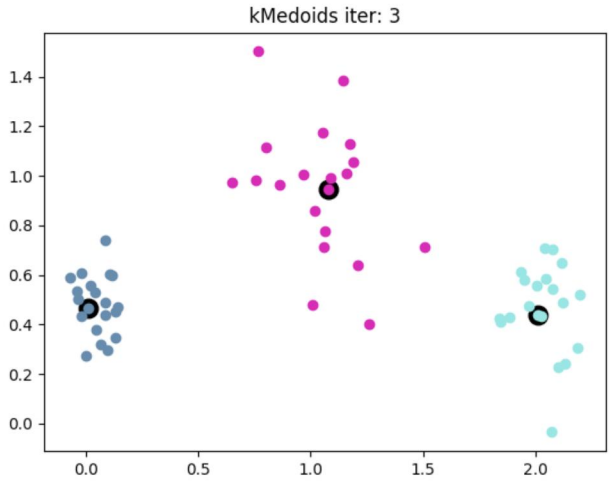
2.3 e. plots **4 / 4**

✓ **- 0 pts** **Correct**

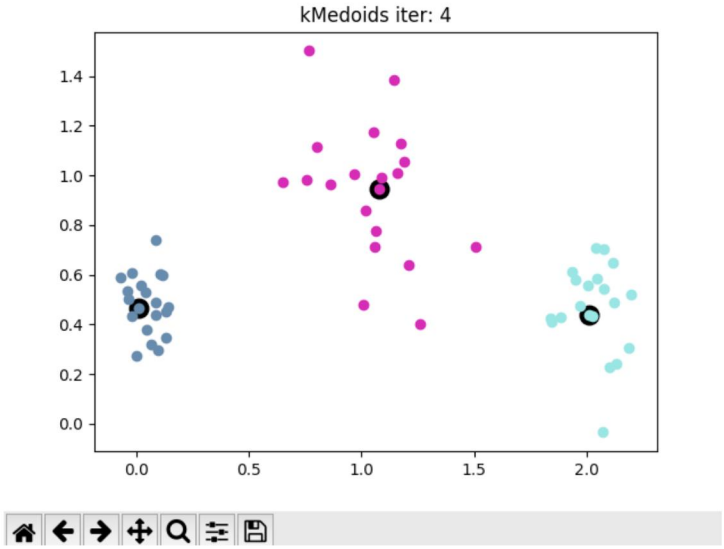
- 2 pts looks very wrong

- 4 pts No Ans

gradescope

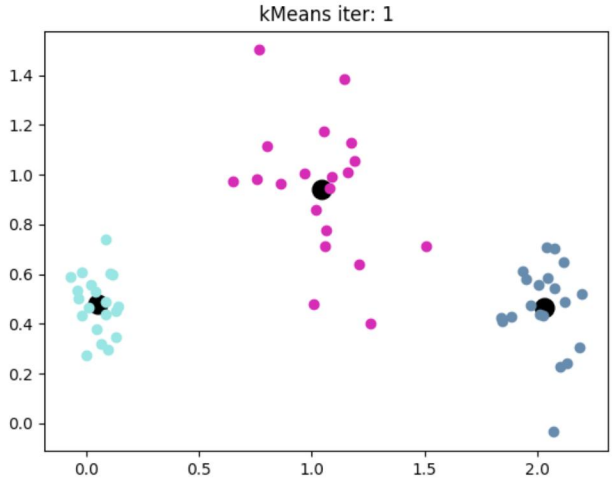


kMedoids Iteration 4:

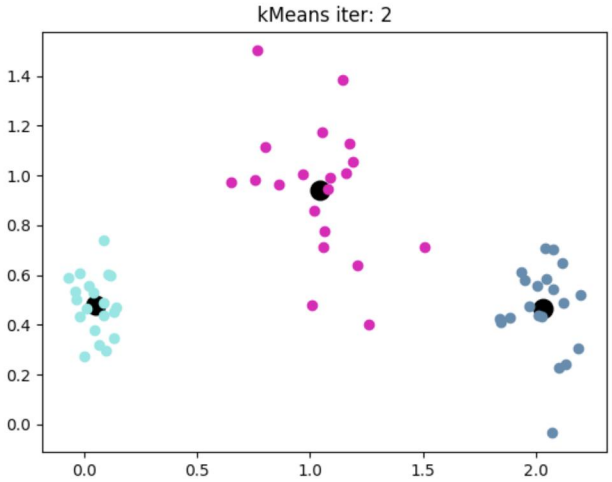


(f) Clustering using `cheat_init` is shown below.

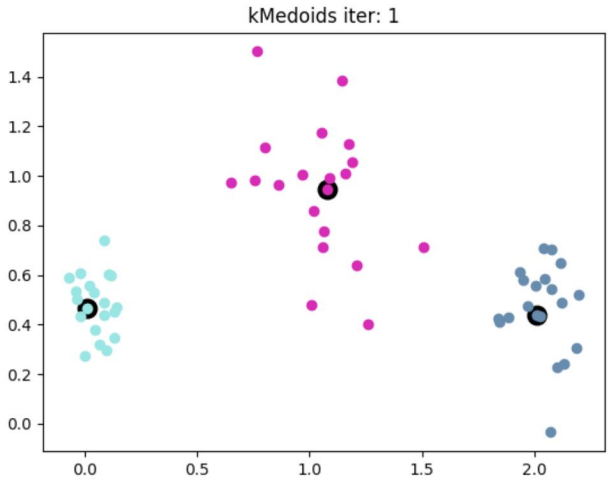
kMeans Iteration 1:



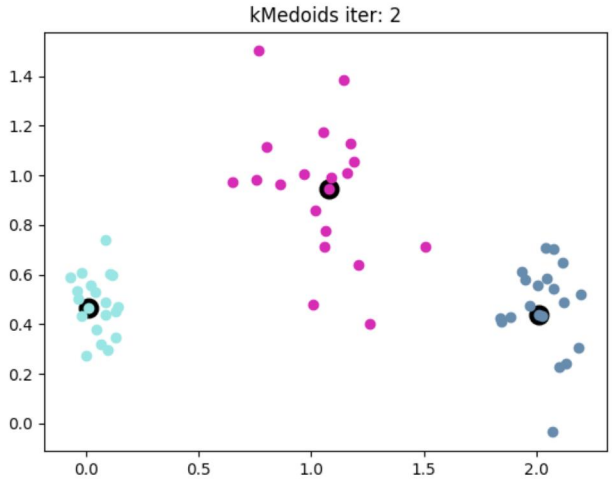
kMeans Iteration 2:



kMedoids Iteration 1:



kMedoids Iteration 2:



3 Clustering Faces [12pts]

Solution:

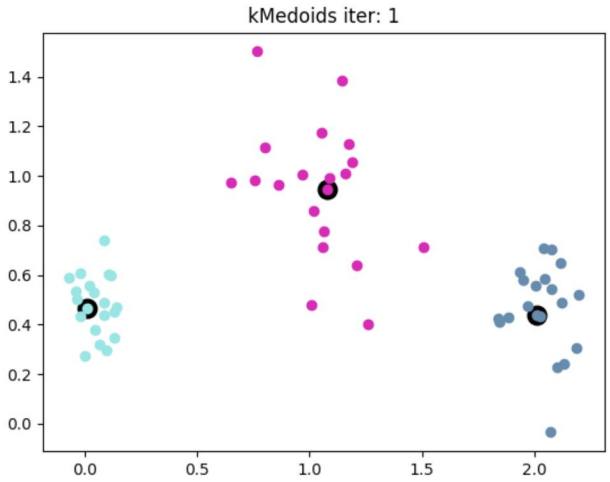
(a) The table for kMeans is shown below.

2.4 f. plots 4 / 4

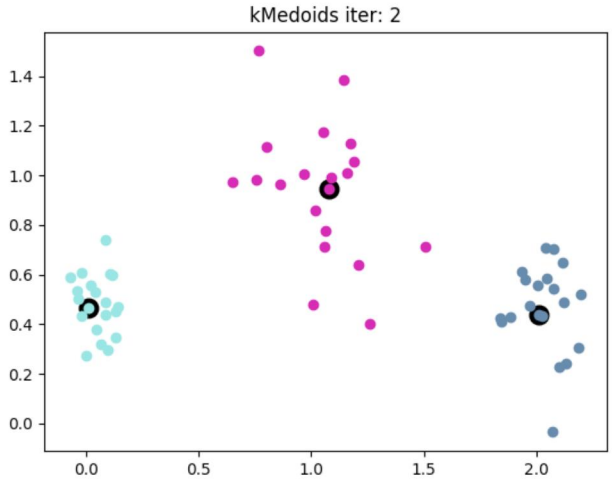
✓ - **0 pts** Correct

- **2 pts** Looks very wrong
- **4 pts** No Ans
- **1 pts** looks wrong with the cheat sheet

gradescope



kMedoids Iteration 2:



3 Clustering Faces [12pts]

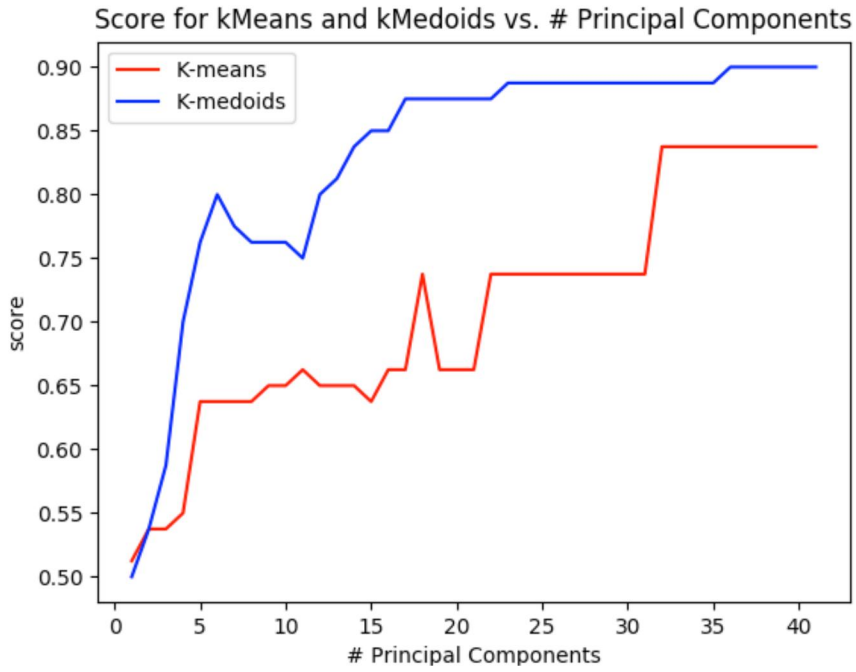
Solution:

(a) The table for kMeans is shown below.

	average	min	max	runtime
k-means	0.55	0.6175	0.775	0.8612
k-medoids	0.575	0.6325	0.725	0.3711

K-medoids performs better than k-means, as the min and max are closer together in k-medoids than k-means. The runtime was also better for k-medoids, as seen in the table.

(b) The plot is shown below



As the number of principal components increases, the clustering score for each K-means and K-medoids increases. This is because the reconstructed dataset more accurately approximates the original dataset as the number of principal components increases, and it is easier to partition the faces into separate categories as the number of features increases. Additionally, K-medoids consistently outperforms K-means for the entire range of principal component values.

(c) I chose K-medoids because it consistently outperformed K-means. I looped over all pairs of people and computed the k-medoids clustering score for each pair. Every class has over 40 faces each. The results of this algorithm are summarized below.

3.1 a. comparison 4 / 4

✓ **- 0 pts Correct**

- 1 pts kmeans scores too low

- 1 pts k-medoids scores too low

- 4 pts not attempted / not found

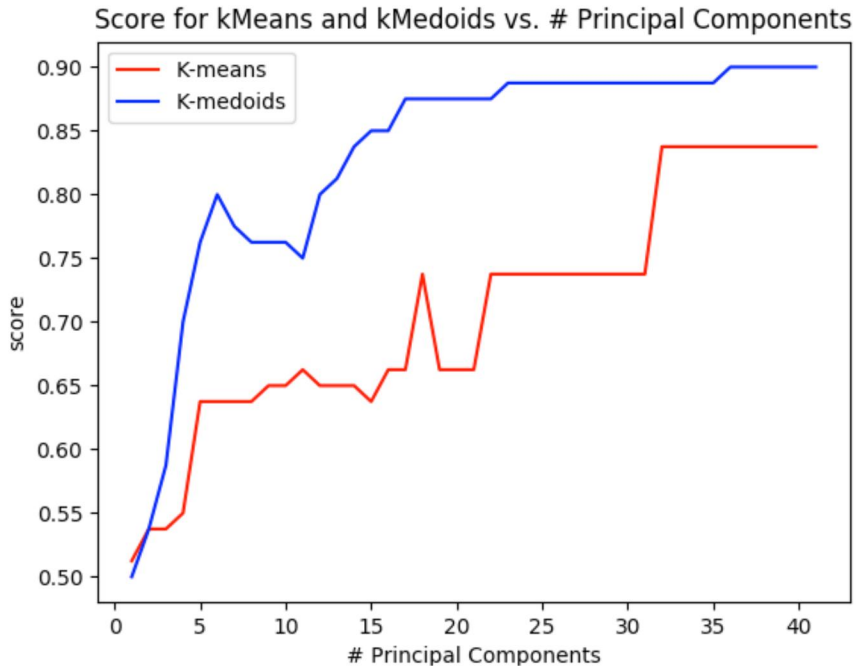
- 2 pts k-medoids scores inaccurate

gradescope

	average	min	max	runtime
k-means	0.55	0.6175	0.775	0.8612
k-medoids	0.575	0.6325	0.725	0.3711

K-medoids performs better than k-means, as the min and max are closer together in k-medoids than k-means. The runtime was also better for k-medoids, as seen in the table.

(b) The plot is shown below



As the number of principal components increases, the clustering score for each K-means and K-medoids increases. This is because the reconstructed dataset more accurately approximates the original dataset as the number of principal components increases, and it is easier to partition the faces into separate categories as the number of features increases. Additionally, K-medoids consistently outperforms K-means for the entire range of principal component values.

(c) I chose K-medoids because it consistently outperformed K-means. I looped over all pairs of people and computed the k-medoids clustering score for each pair. Every class has over 40 faces each. The results of this algorithm are summarized below.

3.2 b. plot clustering score vs. l. **4 / 4**

✓ **- 0 pts** **Correct**

- 1 pts performance of k-means abnormal

- 1 pts performance of k-medoids abnormal

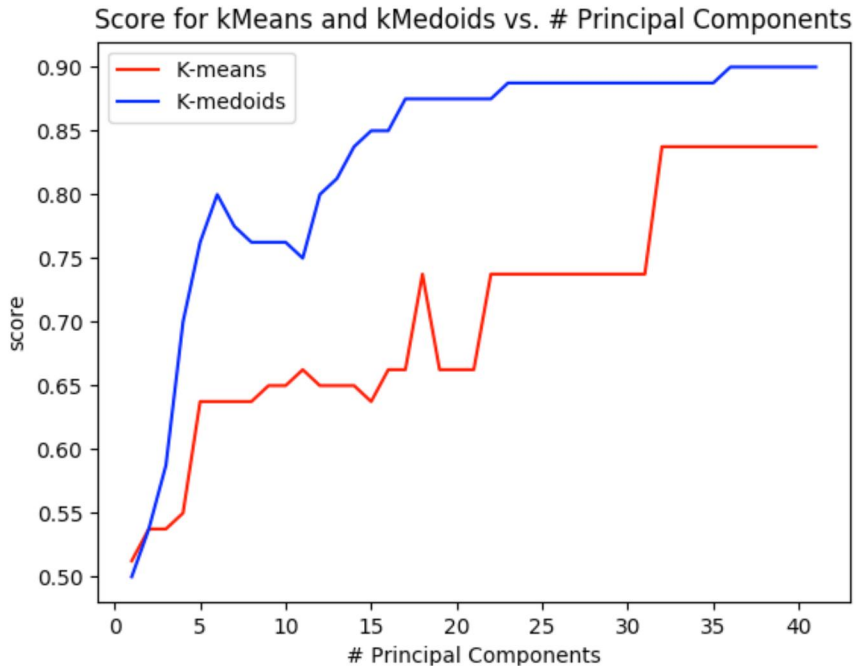
- 4 pts not attempted / not found

gradescope

	average	min	max	runtime
k-means	0.55	0.6175	0.775	0.8612
k-medoids	0.575	0.6325	0.725	0.3711

K-medoids performs better than k-means, as the min and max are closer together in k-medoids than k-means. The runtime was also better for k-medoids, as seen in the table.

(b) The plot is shown below



As the number of principal components increases, the clustering score for each K-means and K-medoids increases. This is because the reconstructed dataset more accurately approximates the original dataset as the number of principal components increases, and it is easier to partition the faces into separate categories as the number of features increases. Additionally, K-medoids consistently outperforms K-means for the entire range of principal component values.

(c) I chose K-medoids because it consistently outperformed K-means. I looped over all pairs of people and computed the k-medoids clustering score for each pair. Every class has over 40 faces each. The results of this algorithm are summarized below.

Above are the two faces that are the most similar to each other.

Above are the two faces that are the most different from each other.

This algorithm gave a satisfactory output. The two faces that the algorithm found to be the most similar are indeed very similar in lighting, face structure, etc. Additionally, the two faces that the algorithm found to be the most different are indeed very different, in facial structure, lighting, color, etc. The clustering score for the first set is 0.9875, representing very similar faces, and the clustering score for the second set is 0.5125, representing very different faces.

3.3 c. find pairs 4 / 4

✓ **- 0 pts** **Correct**

- 4 pts Not attempted / not found

gradescope