INTRODUCTION TO MACHINE LEARNING

Assignment 2-2021AMA2090(Rahman Alam)

Contents

`

NON-AIP701-PART 1_A

1. K-means library

I will be using scikit learn library as I am familiar with this library.

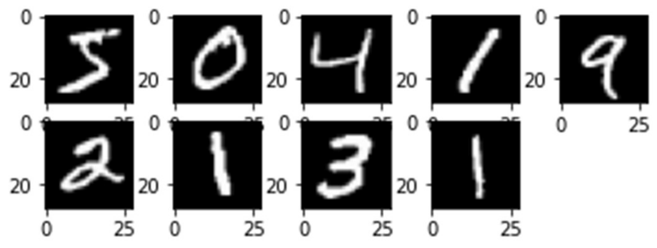
```
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
```

class sklearn.cluster.**KMeans**(*n_clusters=8*, ***, *init='k-means++'*, *n_init=10*, *max_iter=300*, *tol=0.0001*, *verbose=0*, *random_state=None*, *copy_x=True*, *algorithm='lloyd'*)[source]

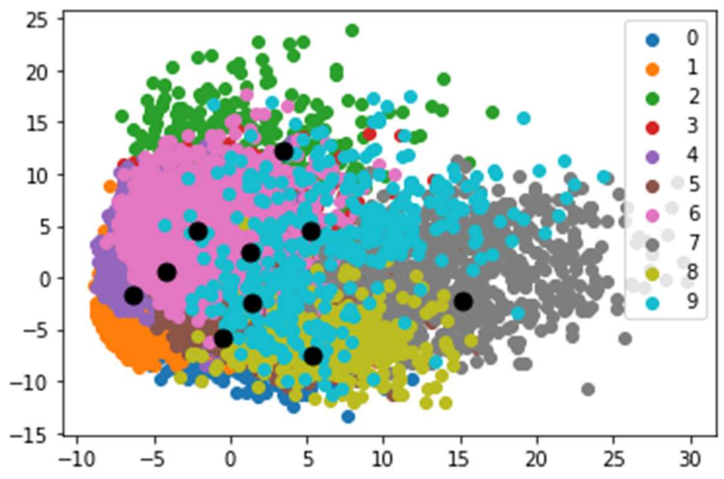
2. visualizing data set

Train: X=(60000, 28, 28), y=(60000,)

Test: X=(10000, 28, 28), y=(10000,)



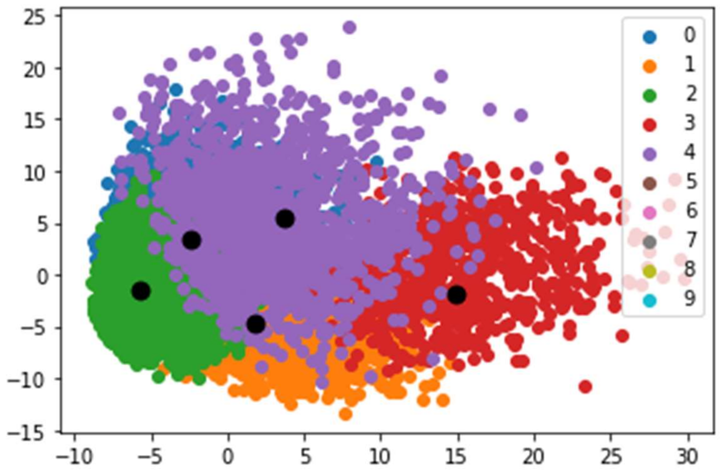
3. K-means with K = 10 on your handwritten digits(60000)



	0	1	2	3	4	5	6	7	8	9
0	19	13	0	140	11	50	33	622	63	29
1	72	1052	0	6	0	3	0	0	1	1
2	205	170	3	44	1	209	28	20	190	162
3	257	193	3	489	12	20	3	1	18	14
4	3	84	74	34	81	19	650	9	0	28
5	155	125	1	339	167	12	53	4	25	11
6	10	62	0	7	0	683	7	32	60	97
7	29	147	151	37	550	1	110	0	0	3
8	350	126	2	302	100	6	51	9	11	17
9	11	57	19	58	366	1	473	4	0	20

Accuracy Score of Kmeans testing set: 17%

4. K-means with K = 5 on your handwritten digits(60000)

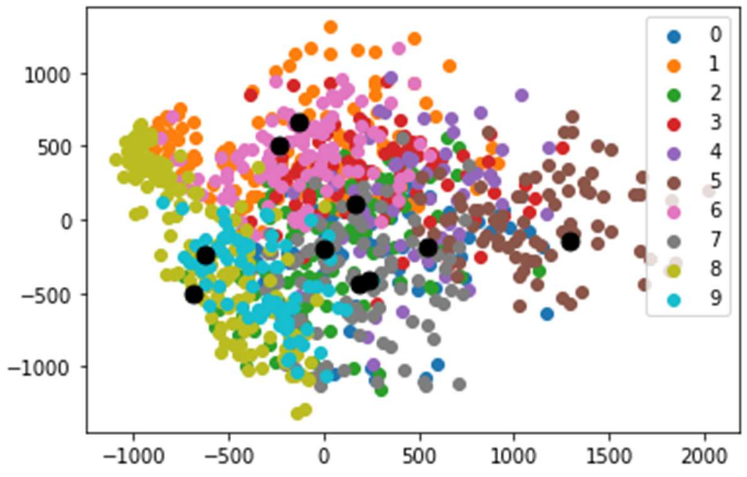


	0	1	2	3	4	5	6	7	8	9
0	14	268	17	614	67	0	0	0	0	0
1	0	17	1115	0	3	0	0	0	0	0
2	24	590	206	26	186	0	0	0	0	0
3	29	403	359	1	218	0	0	0	0	0
4	657	19	123	11	172	0	0	0	0	0
5	257	282	214	6	133	0	0	0	0	0
6	6	678	70	86	118	0	0	0	0	0

7	695	11	196	1	125	0	0	0	0	0
8	226	279	295	13	161	0	0	0	0	0
9	775	9	95	5	125	0	0	0	0	0

Accuracy Score of Kmeans testing set: 4%

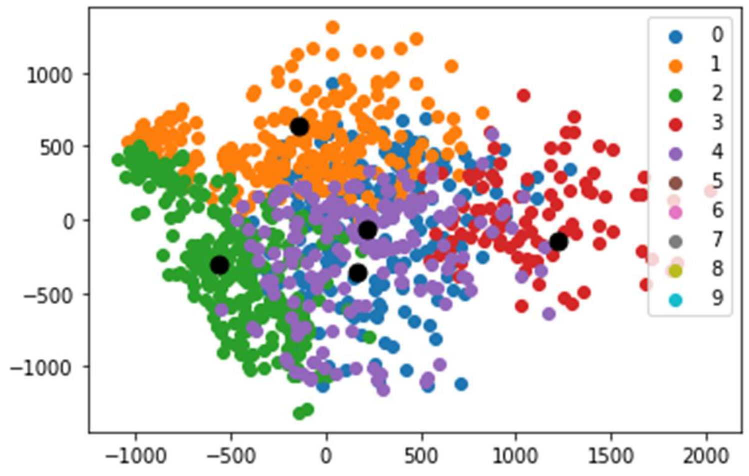
5. K-means with K = 10 on your handwritten digits(3000)



	0	1	2	3	4	5	6	7	8	9
0	12	1	0	8	10	66	3	2	0	0
1	0	30	0	0	0	0	10	0	66	1
2	20	38	0	39	1	2	1	5	2	3
3	4	14	9	6	18	0	44	1	2	1
4	7	1	16	0	1	0	0	31	16	30
5	11	8	28	2	14	1	18	2	15	6
6	5	25	3	13	0	10	9	23	0	4
7	7	1	0	1	6	0	2	12	59	9
8	12	13	25	15	2	1	16	2	6	1
9	2	1	25	3	4	0	1	16	20	20

Accuracy Score of K-means testing set: 10%

6. K-means with K = 5 on your handwritten digits(3000)



0	1	2	3	4	5	6	7	8	9	
0	8	3	1	76	14	0	0	0	0	0
1	0	53	53	0	1	0	0	0	0	0
2	46	31	5	4	25	0	0	0	0	0
3	13	66	6	0	14	0	0	0	0	0
4	32	1	52	0	17	0	0	0	0	0
5	13	24	25	4	39	0	0	0	0	0
6	30	33	7	7	15	0	0	0	0	0
7	10	1	71	1	14	0	0	0	0	0
8	20	27	11	1	34	0	0	0	0	0
9	27	1	51	0	13	0	0	0	0	0

Accuracy Score of K-means testing set: 8%

7. Conclusion

Using an implementation of your choice, run K-means with K = 10 on your handwritten digits data set. Assess the clusters obtained: do they correspond to the digits 0–9? If you label each cluster with the digit that occurs most frequently within it, then what is your classification accuracy with this unsupervised method? What kinds of misclassifications are happening, and why? Now try re-running K-means with K = 5. Do your clusters make any sense in this case? Why or why not?

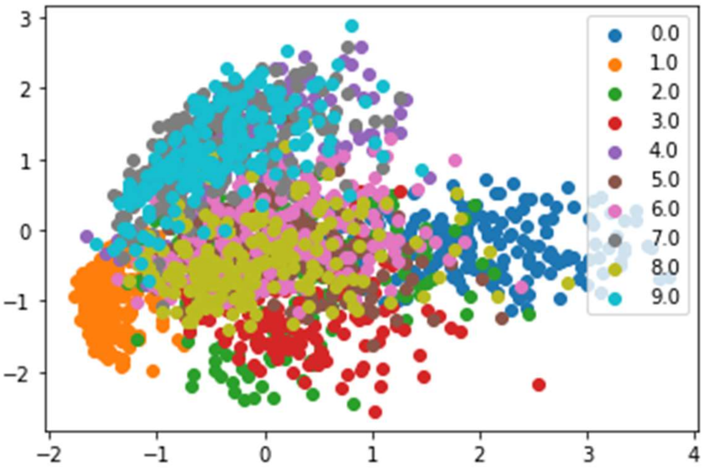
I have run K-means on original data set containing 60000 train data and 10000 test data set. Also, I have ran the k-mean on the dataset available for the data in Assignment 3 with K=10 and 5 both. So, we can see above for K=10 the clusters obtained are 10 but they are not much corresponding as so many overlapping is observed resulting very poor accuracy of **14%(Most observed).**Other cases of K=5 and data with lesser input are giving worse results as mentioned in the table. Misclassification is happening with all the digits with minimum misclassification with1 and 3.

Running with K=5 are resulting worse results than the K=10 as they can't even recognize digits after 4 as shown in graph.

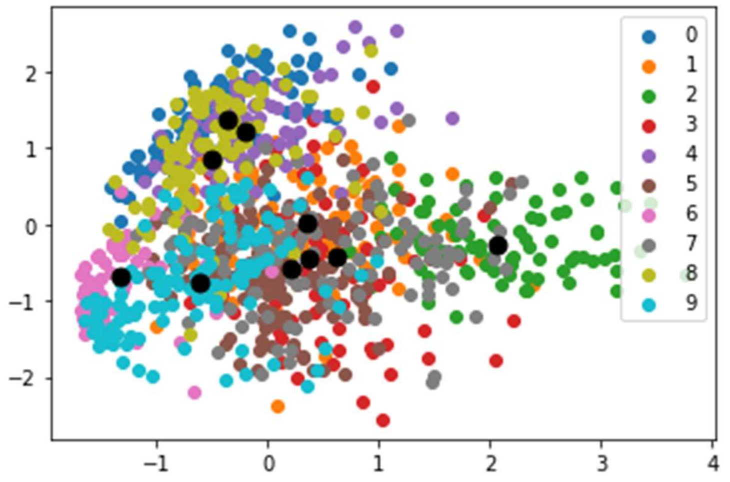
	Model	Data/Features	accuracy(%)
Question-1	**K=10**	**60000/784**	14
	K=5	**60000/784**	4
	K=10	**3000/784**	10
	K=5	**3000/784**	8
Question-2(PCA+Kmeans)	**K=10**	**3000/25**	12
	K=5	**3000/25**	14

NON-AIP701-PART 1_B

8. Visualizing PCA data(Data from assignment 2)



9. K-means with K = 10 on your handwritten digits(PCA=25(features))

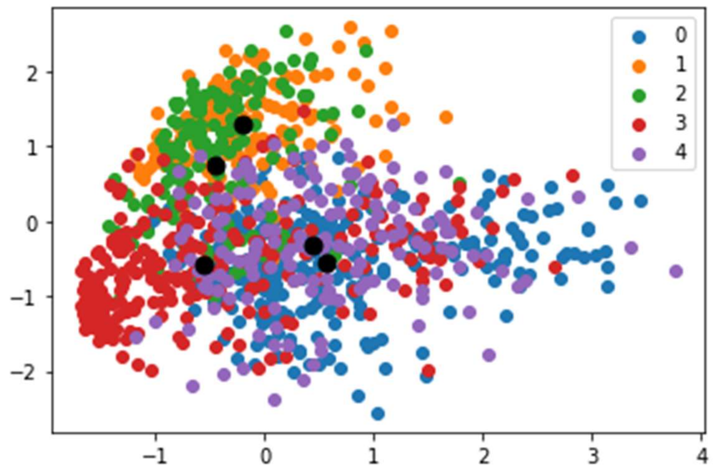


0	1	2	3	4	5	6	7	8	9	
0	0	5	76	0	0	6	0	16	0	1
1	0	0	0	0	0	0	53	0	0	41
2	1	8	1	0	2	2	6	56	5	12
3	0	2	0	34	3	48	4	13	0	2
4	34	2	0	4	31	0	1	1	19	4
5	3	0	0	20	2	47	0	10	9	7
6	0	86	5	0	0	5	2	10	0	1

7	**25**	**0**	**2**	**2**	**20**	**0**	**3**	**1**	**44**	**4**
8	**0**	**2**	**1**	**5**	**2**	**24**	**3**	**5**	**2**	**48**
9	**40**	**0**	**0**	**5**	**37**	**0**	**1**	**2**	**21**	**1**

Accuracy Score of Kmeans testing set: 12%

10. K-means with K = 5 on your handwritten digits(PCA=25(features))



0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	
0	**64**	**0**	**0**	**22**	**18**	**0**	**0**	**0**	**0**	**0**
1	**1**	**0**	**0**	**93**	**0**	**0**	**0**	**0**	**0**	**0**
2	**4**	**2**	**3**	**49**	**35**	**0**	**0**	**0**	**0**	**0**
3	**76**	**5**	**0**	**19**	**6**	**0**	**0**	**0**	**0**	**0**
4	**1**	**56**	**33**	**4**	**2**	**0**	**0**	**0**	**0**	**0**
5	**64**	**4**	**11**	**13**	**6**	**0**	**0**	**0**	**0**	**0**
6	**1**	**0**	**0**	**23**	**85**	**0**	**0**	**0**	**0**	**0**
7	**0**	**39**	**54**	**7**	**1**	**0**	**0**	**0**	**0**	**0**
8	**51**	**2**	**19**	**15**	**5**	**0**	**0**	**0**	**0**	**0**
9	**2**	**66**	**32**	**7**	**0**	**0**	**0**	**0**	**0**	**0**

Accuracy Score of K-means testing set: 14%

11. Conclusion

b) Now we will try to use a lower-dimensional representation of our images for clustering them. In Assignment 2 you were already given, for your personalised data set, the top 25 principal components obtained by using PCA for dimensionality reduction. Visualise at least the top few components and try to interpret what kind of variation in the data they're capturing. Now run K-means with K = 10 on your PCA representations. Repeat the analysis as in (a): how does this clustering in PCA space compare to that in raw pixel space? Is it better or worse? Why?

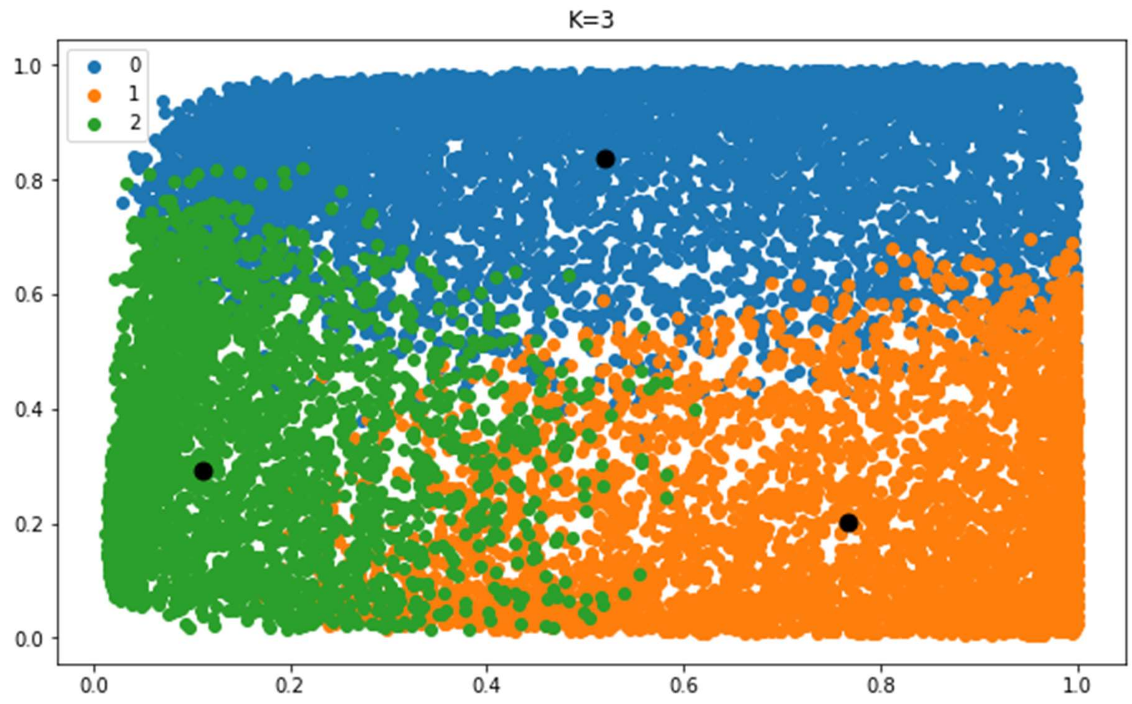
Overall PCA captured 25 dimensions are working equivalently compared to full 784 features as we are getting 12% accuracy with K=10 and 14% accuracy with K=5. Compare to raw pixel this representation is working well, Because the data is very sparse in full representation, so the PCA dimensions are working very well.

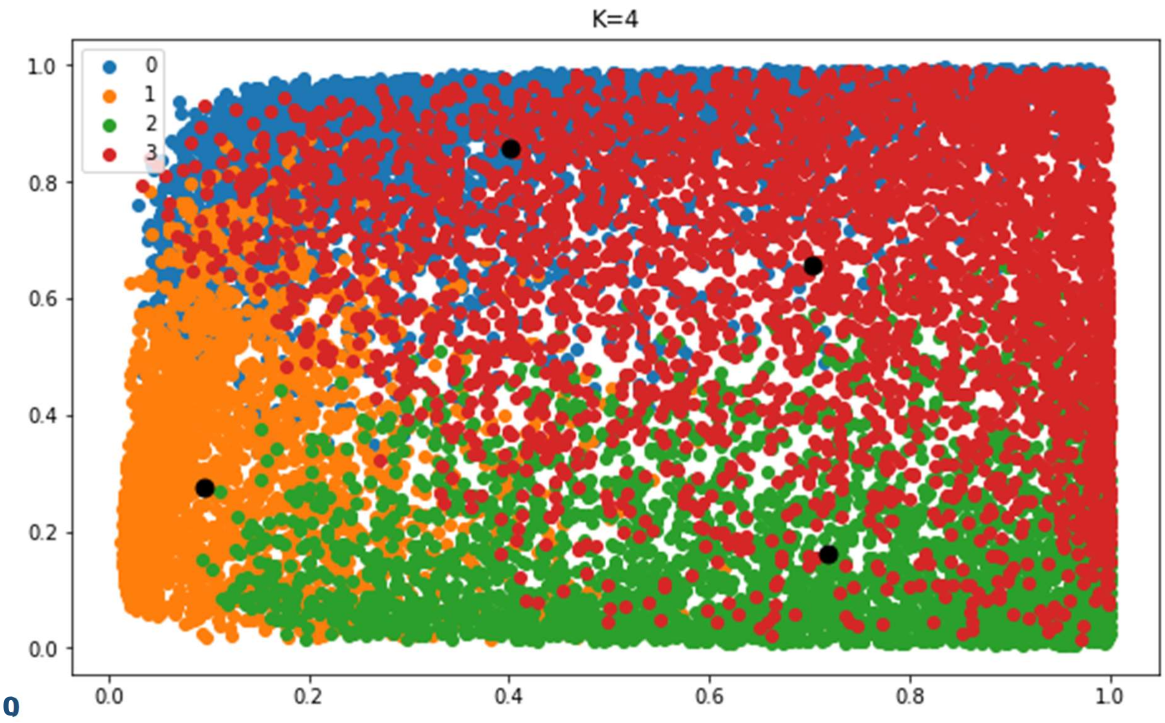
NON-AIP701-PART 2

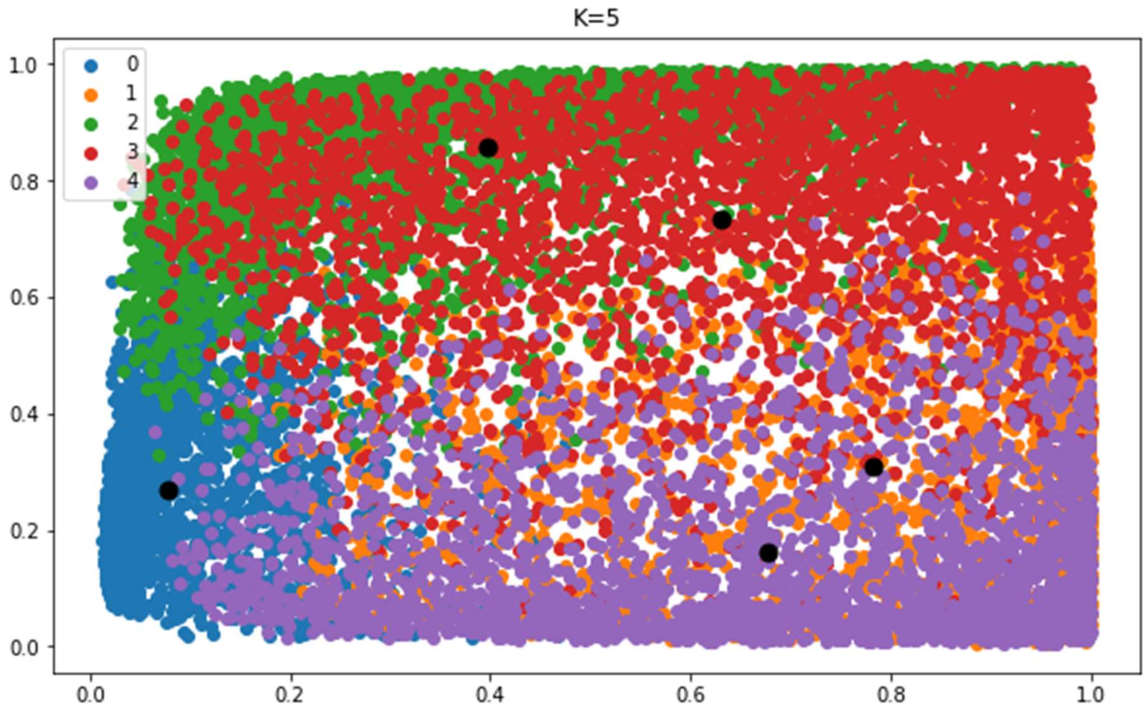
12. K-means for 127 features

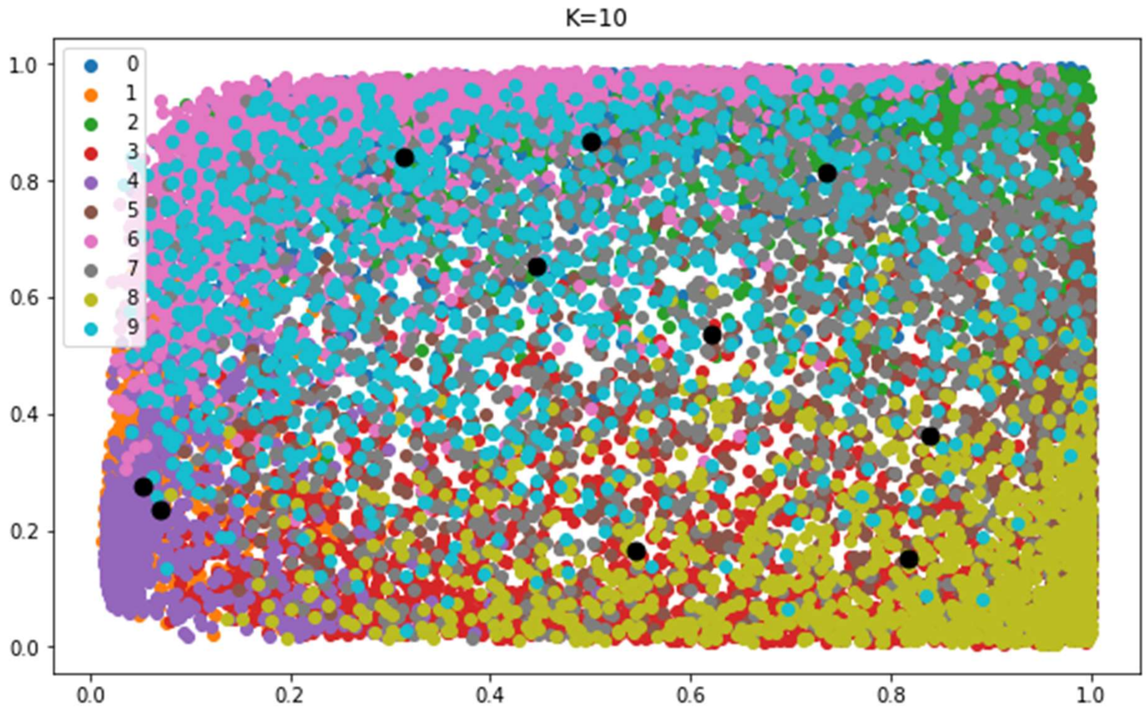
Best result is at K=15

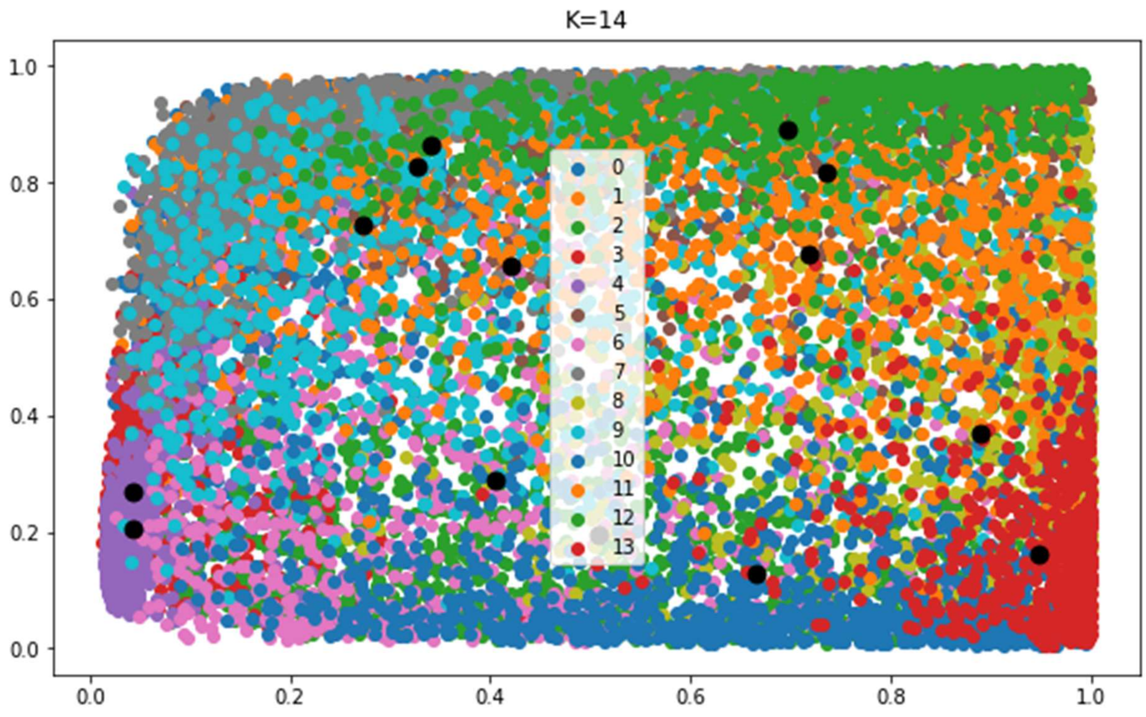
Few results are as below.

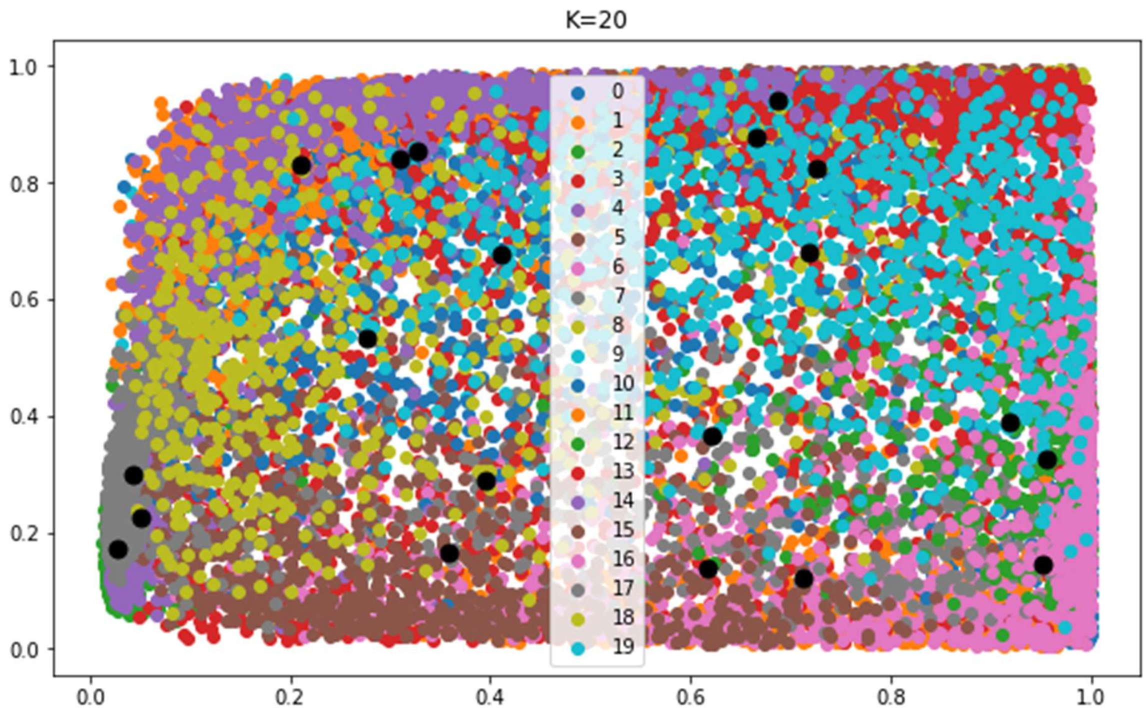





K=5
0
1
2
3
4
0.0
0.2
0.4
0.6
0.8
1.0


K=10
0
1
2
3
4
5
6
7
8
9
0.0
0.2
0.4
0.6
0.8
1.0

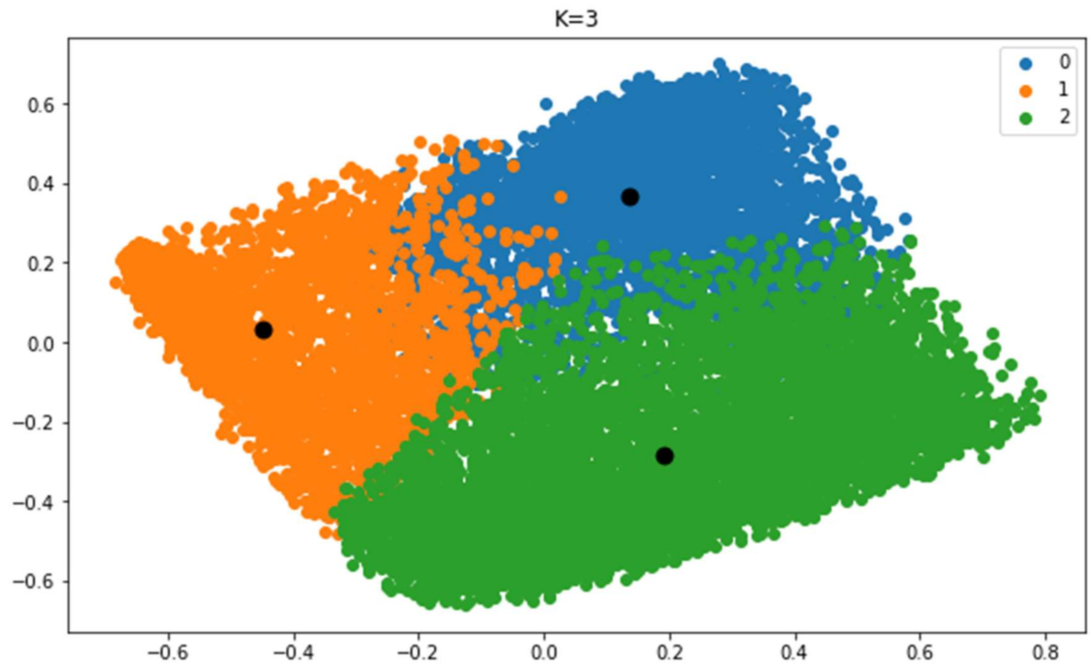


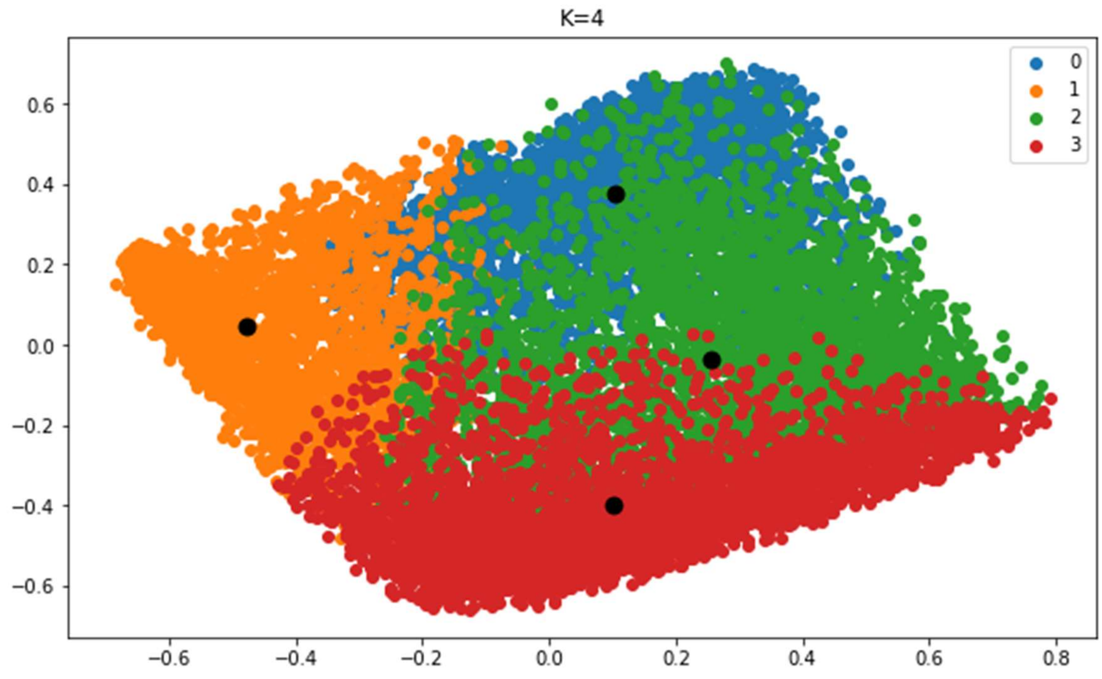


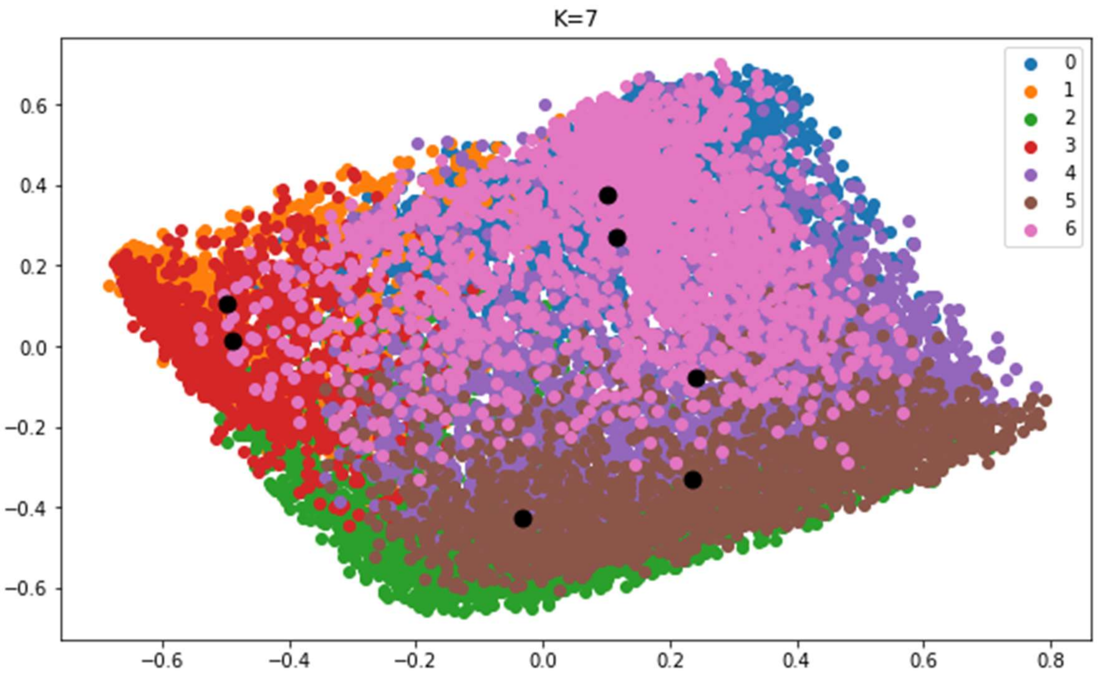
13. K-means for 25 PCA features

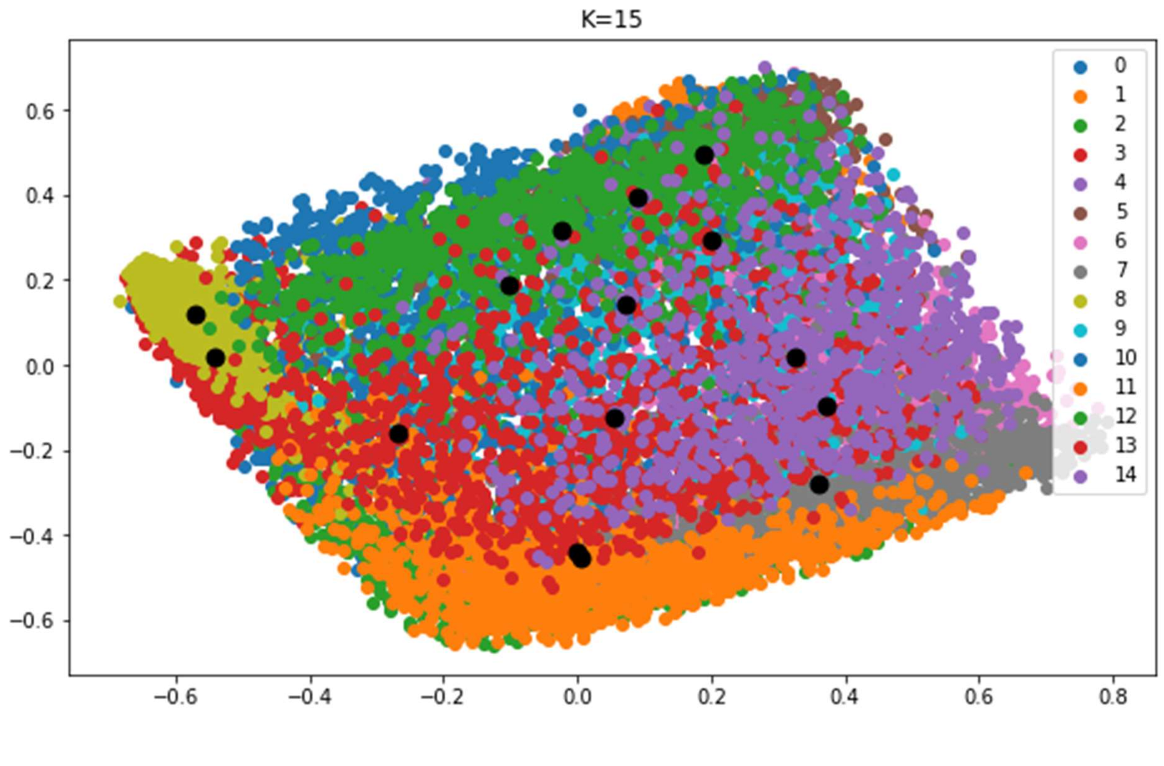
Best result is at K=15 again

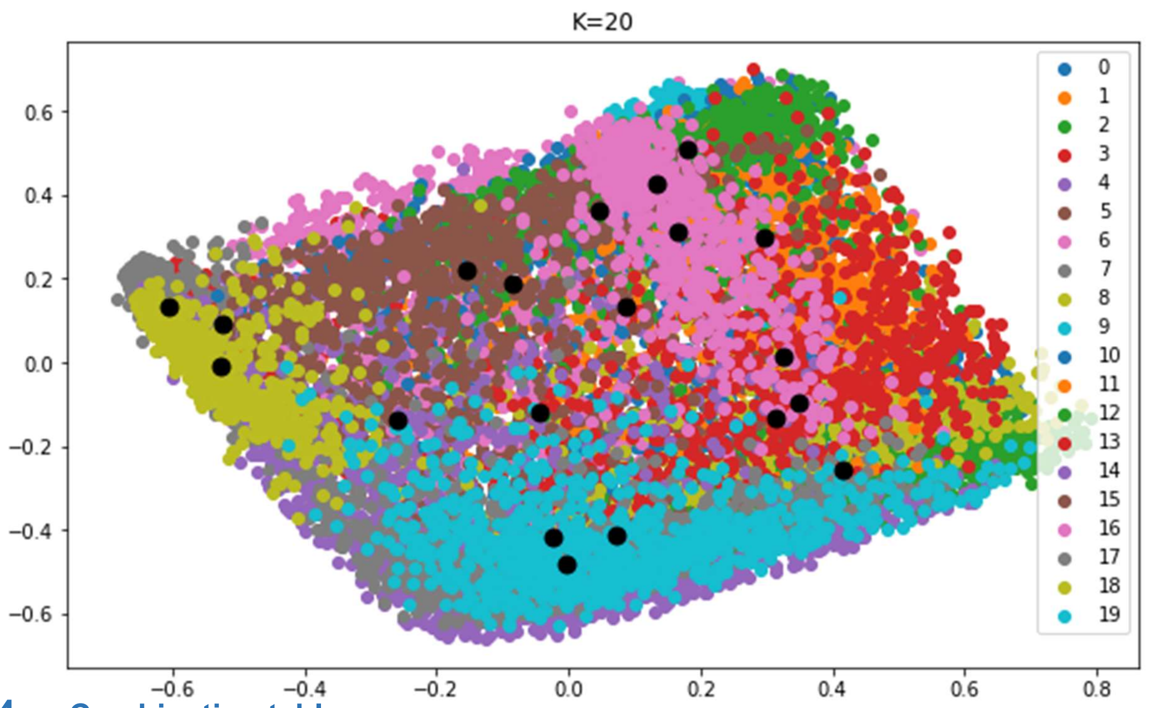
Few results are as below.


K=3
0
1
2


K=4
0
1
2
3


K=7
0
1
2
3
4
5
6
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8


K=15
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8



14. Combination table

Model	Data/Features
K=0	15000/127
K=5	15000/127
K=10	15000/127
K=15	15000/127
K=20	15000/127
K=5	15000/127
K=0	15000/25
K=5	15000/25
K=10	15000/25
K=15	15000/25
K=20	15000/25
K=5	15000/25

15. Conclusion

In both raw and PCA dimensions the best model seems with K=15.

The score on Kaggle for this model is around=

16. Reference

- training set images (9912422 bytes)--->http://yann.lecun.com/exdb/mnist/train-images-idx3-ubyte.gz
- training set labels (28881 bytes)----->http://yann.lecun.com/exdb/mnist/train-labels-idx1-ubyte.gz
- test set images (1648877 bytes)------->http://yann.lecun.com/exdb/mnist/t10k-images-idx3-ubyte.gz
- test set labels (4542 bytes)---------->http://yann.lecun.com/exdb/mnist/t10k-labels-idx1-ubyte.gz
- https://www.youtube.com/watch?v=xp-WhryKhas
- https://github.com/SimpleAI2022/MNIST_FC1/blob/main/MNIST_FC1_Complete.ipynb
- https://bytepawn.com/building-a-pytorch-autoencoder-for-mnist-digits.html
- https://medium.com/analytics-vidhya/training-mnist-handwritten-digit-data-using-pytorch-5513bf4614fb
- https://stackoverflow.com/questions/69975400/pytorch-cnn-expected-input-to-have-1-channel-but-got-60000-channels-instead
- https://www.youtube.com/watch?v=Y_hSBwucDjg