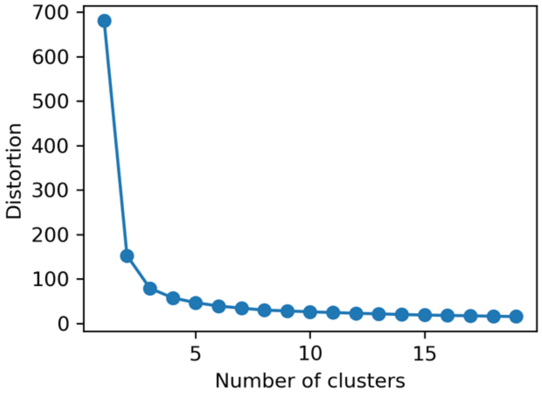
Report

Task 1:

All clusters are included in the folder and implemented in the main.py.

Task 2:

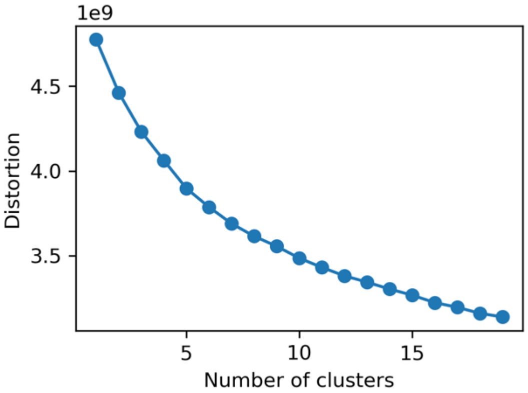
- [Iris dataset]



The distortion decreases significantly at the cluster number of 1 and 2. Then a very stable curve is achieved when cluster number is larger than 3. Such result is consistent with the fact that the iris dataset has 3 labels as we know before.

Conclusion: k=3.

- [MNIST dataset]

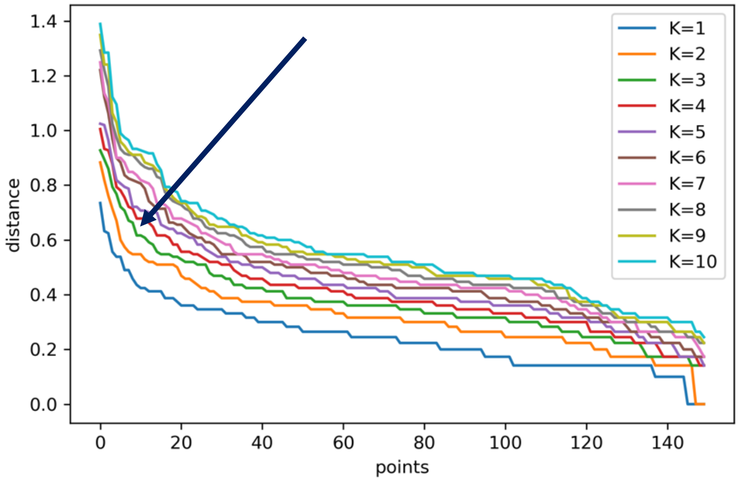


The distortion curve shows a continuously decreasing trend. No obvious decrease can be observed. As we know, the MNIST dataset has 10 labels. However, the distortion at 10 clusters doesn't show an obvious decrease, which means the instance between each label may have similar features. The possible reason may result from the sparse matrix, in which a lot of features are 0 for each instance.

Conclusion: k=10.

Task 3

- [Iris dataset]

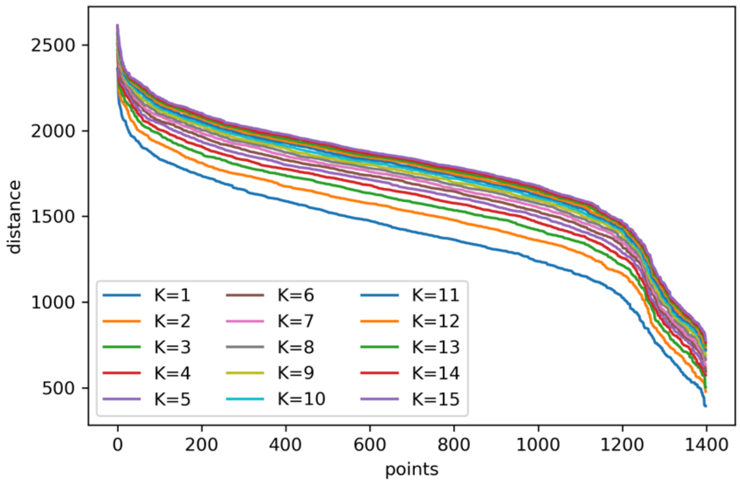


In order to get reasonable MinPts and eps, a best_values method is developed.

K from 1 to 10 are set to get the above curves. The threshold point is shown with a blue arrow. Corresponding to this point, K=3, MinPts in x label is about 10, and the eps in y axis is 0.6.

The result cannot match well with the iris dataset since it have three labels and 50 instances for each label.

- [MNIST dataset]



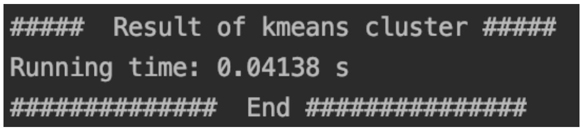
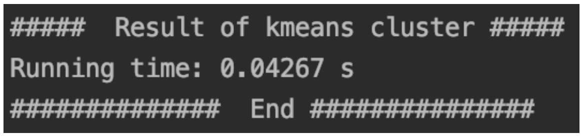
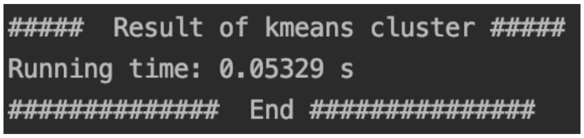
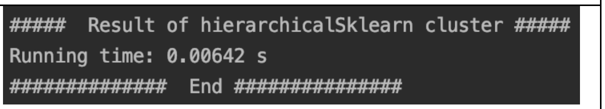
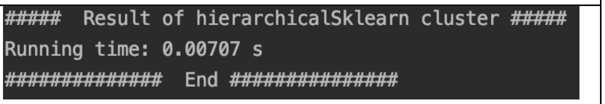
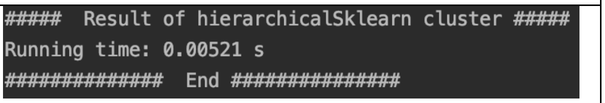
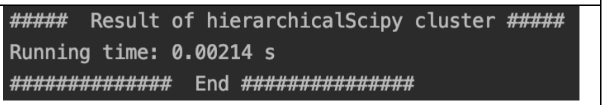
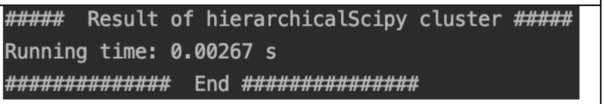
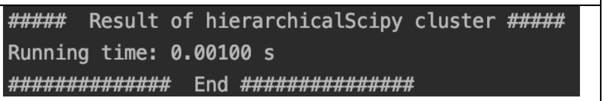
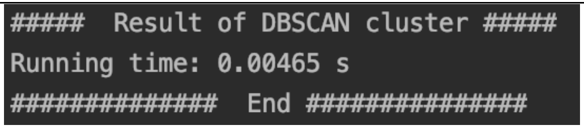
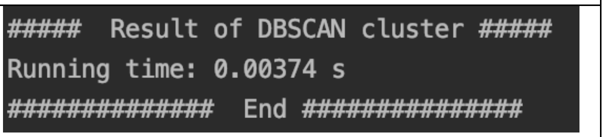
The same procedure is applied to sub MNIST dataset. We have already know the K should be 10. However, the curves in the above figure doesn't show an obvious changing point on the curve with k=10. The instance in MNIST dataset may be very similar and hard to differentiated.

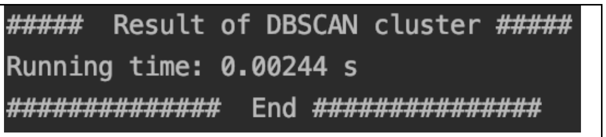
Task 4

The Iris (Table 1) and MNIST (Table 2) datasets are both processed by different cluster methods.

- Iris data
  Read the data from an input file.
- MNIST
  The original MNIST data contains 70000 instances, which is too many to run in laptop. 2% of the dataset is selected as a sub dataset and save as 'MNIST.csv' in the same folder. In the main code, only this sub dataset is used for training and testing, including 1400 instances. A data preprocess is used to achieve the above process, and the corresponding code is placed outside the main code.

Table 1 Performance of different clusters [Iris dataset]

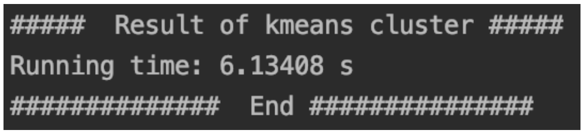
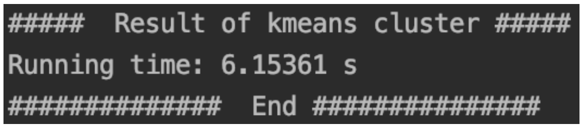
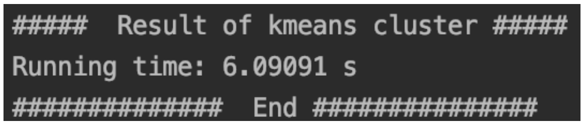
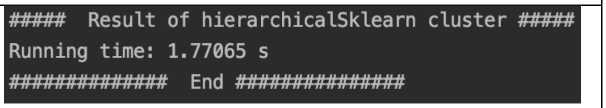
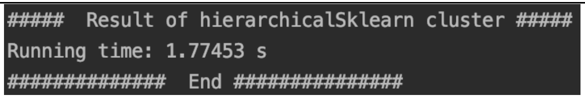
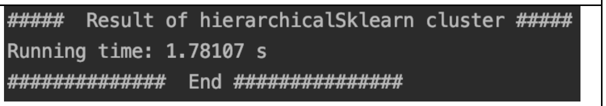
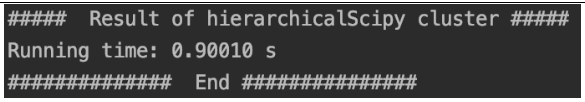
Clusters	Parameters	Run Time and Errors
K-means	--n_clusters1 3 --init random --n_init 200	SSE = 78.941  ##### Result of kmeans cluster ##### Running time: 0.04138 s ############## End ###############
	--n_clusters1 3 --init k-means++ --n_init 200	SSE = 78.941  ##### Result of kmeans cluster ##### Running time: 0.04267 s ############## End ###############
	--n_clusters1 3 --init k-means++ --n_init 500	SSE = 78.941  ##### Result of kmeans cluster ##### Running time: 0.05329 s ############## End ###############
hierarchical_Sklearn	--n_clusters2 3 --affinity euclidean --linkage complete	 ##### Result of hierarchicalSklearn cluster ##### Running time: 0.00642 s ############# End ##############
	--n_clusters2 3 --affinity manhattan --linkage complete	 ##### Result of hierarchicalSklearn cluster ##### Running time: 0.00707 s ############# End ##############
	--n_clusters2 3 --affinity euclidean --linkage single	 ##### Result of hierarchicalSklearn cluster ##### Running time: 0.00521 s ############# End ##############
hierarchical_Scipy	--method complete --criterion maxclust	 ##### Result of hierarchicalScipy cluster ##### Running time: 0.00214 s ############# End ##############
	--method single --criterion maxclust	 ##### Result of hierarchicalScipy cluster ##### Running time: 0.00267 s ############# End ##############
	--method single --criterion distance	 ##### Result of hierarchicalScipy cluster ##### Running time: 0.00100 s ############# End ##############
DBSCAN	--eps 0.2 --min_samples 5 --metric euclidean	 ##### Result of DBSCAN cluster ##### Running time: 0.00465 s ############## End ###############
	--eps 0.5 --min_samples 5 --metric euclidean	 ##### Result of DBSCAN cluster ##### Running time: 0.00374 s ############## End ###############

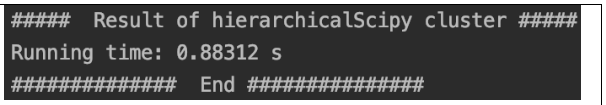
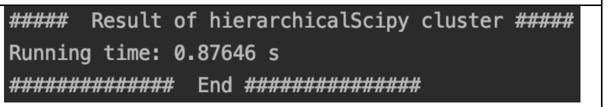
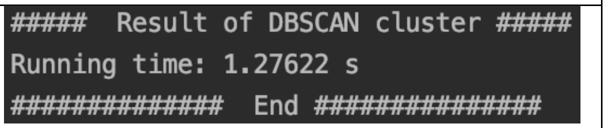
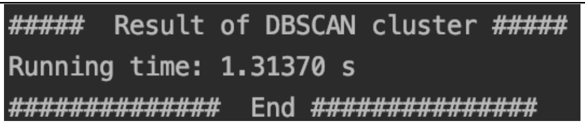
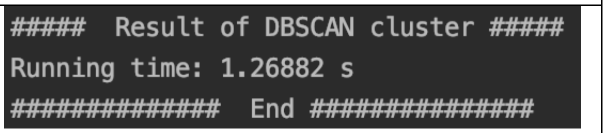
	--eps 0.2 --min_samples 10 --metric euclidean	 ##### Result of DBSCAN cluster ##### Running time: 0.00244 s ############## End ###############

Neglecting the initial labels in the dataset, only the four features are used to do the clustering since it is an unsupervised task.

- The SSE in kmeans method is 78.941.
- Overall running time: due to the smaller number of instances in the Iris dataset, the difference among various clustering methods is not obvious. The performance of each method cannot be properly compared.

Table 2 Performance of different clusters [MNIST dataset]

Clusters	Parameters	Run Time and Errors
K-means	--n_clusters1 3 --init random --n_init 200	SSE = 4231401191.474  ##### Result of kmeans cluster ##### Running time: 6.13408 s ############## End ###############
	--n_clusters1 3 --init k-means++ --n_init 200	SSE = 4231401191.474  ##### Result of kmeans cluster ##### Running time: 6.15361 s ############## End ###############
	--n_clusters1 3 --init k-means++ --n_init 500	SSE = 4231401191.474  ##### Result of kmeans cluster ##### Running time: 6.09091 s ############## End ###############
hierarchical_Sklearn	--n_clusters2 3 --affinity euclidean --linkage complete	 ##### Result of hierarchicalSklearn cluster ##### Running time: 1.77065 s ############# End ##############
	--n_clusters2 3 --affinity manhattan --linkage complete	 ##### Result of hierarchicalSklearn cluster ##### Running time: 1.77453 s ############# End ##############
	--n_clusters2 3 --affinity euclidean --linkage single	 ##### Result of hierarchicalSklearn cluster ##### Running time: 1.78107 s ############# End ##############
hierarchical_Scipy	--method complete --criterion maxclust	 ##### Result of hierarchicalScipy cluster ##### Running time: 0.90010 s ############# End ##############

	--method single --criterion maxclust	 ##### Result of hierarchicalScipy cluster ##### Running time: 0.88312 s ############# End ##############
	--method single --criterion distance	 ##### Result of hierarchicalScipy cluster ##### Running time: 0.87646 s ############# End ##############
DBSCAN	--eps 0.2 --min_samples 5 --metric euclidean	 ##### Result of DBSCAN cluster ##### Running time: 1.27622 s ############## End ###############
	--eps 0.5 --min_samples 5 --metric euclidean	 ##### Result of DBSCAN cluster ##### Running time: 1.31370 s ############## End ###############
	--eps 0.2 --min_samples 10 --metric euclidean	 ##### Result of DBSCAN cluster ##### Running time: 1.26882 s ############## End ###############

For the MNIST dataset, different cluster methods are run with different parameters considered. In order to achieve clustering purpose, the original labels are also not considered as Iris dataset.

- The SSE in kmeans method is 4231401191.474.
- Overall running time: Kmeans > hierarchical_Sklearn > DBSCAN > hierarchical_Scipy