Pattern Recognition and Biometrics - ENSIIE

Clustering Project

24th May 2024

Professor:

GARCIA Sonia

By:

CHEAM Richard

NOUV Ratanakmuny

Table of Contents

1. Introduction

- *Optical Recognition of Handwritten Digits* (UC Irvine ML Repository) retrieved from
  - https://archive.ics.uci.edu/ml/datasets/optical+recognition+de+chiffres+manuscrits

- Preprocessing programs made by *NIST*: 32x32 bitmaps -> 8x8 matrix values in [0,16]

- ***5620*** instances, ***64*** features (*integer*): ***3823*** and **1797** instances for training and testing

- Classification by exploiting ***K-means*** and ***Hierarchical*** **clustering** algorithms

2. *K*-means

- An approach for partitioning a data set into *K* distinct group.

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

- Pseudo algorithm:
  - Specify the desired number of clusters or groups *K*
  - Assign each observation to exactly one of the *K* clusters (initial clusters)
  - Iterate until the cluster assignments stop changing:
    - For each cluster *K*, compute the cluster *centroid*
    - Assign each data point to the cluster whose *centroid* is *closest*

K-means for *K* = 10

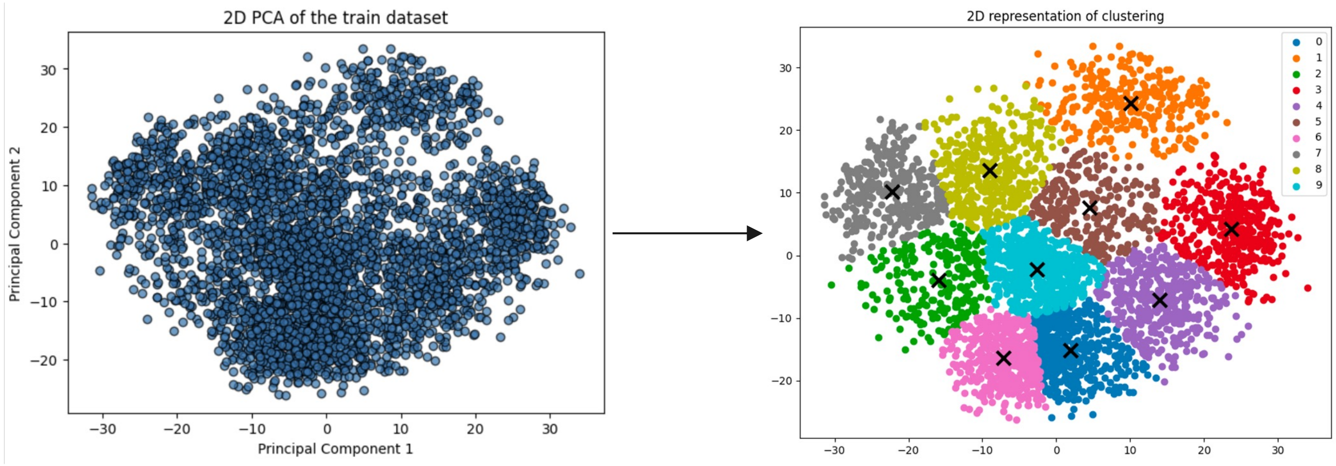


Figure 1: data points *before* and *after* clustering for K = 10

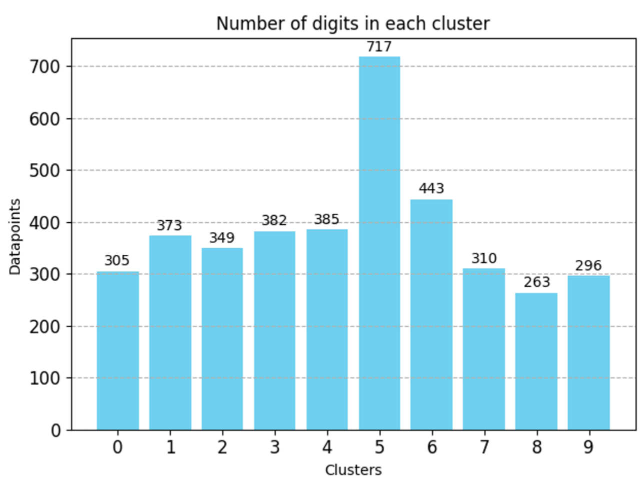
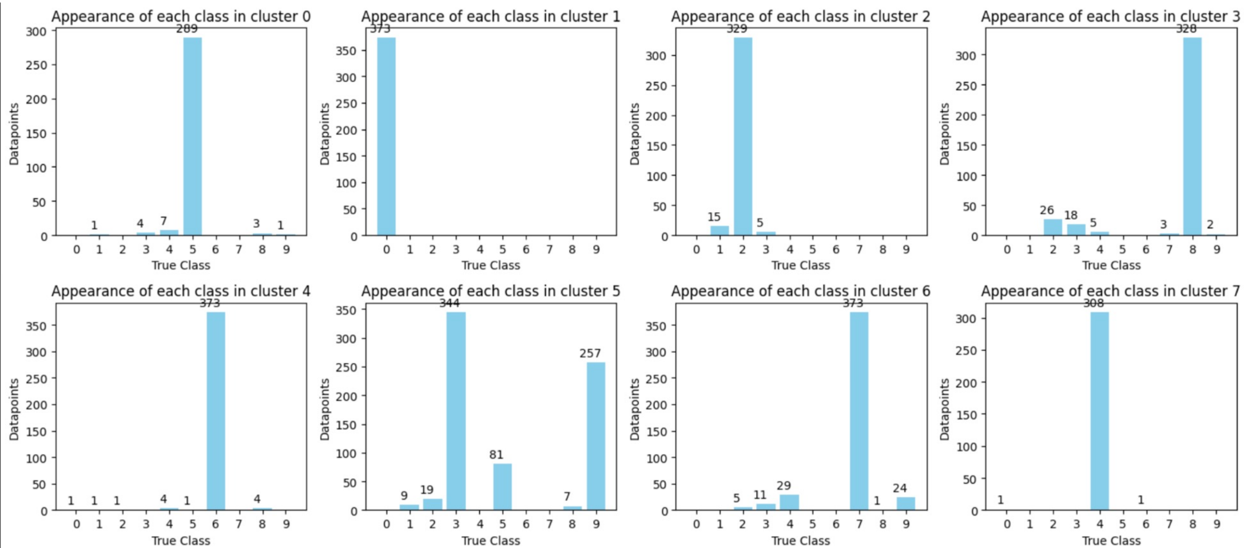


Figure 2: number of data points in each cluster

- Cluster 5 possesses the highest data points
- While others are fairly clustered

K-means for *K* = 10



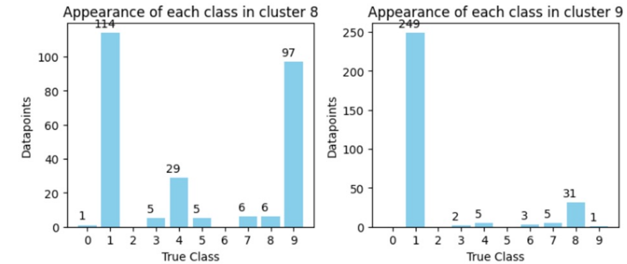


Figure 3: occurrence of classes in each cluster

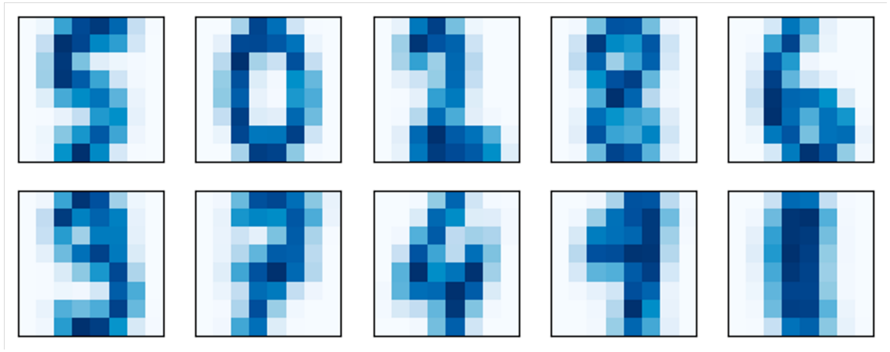


Figure 4: centroid plot for each cluster

- Cluster 5 and 8 seem to have more than 1 class
- For cluster 5, there is a dilemma mostly between 3 and 9
- Whereas, for cluster 8, it is for digit 1 and 9

Silhouette score = 0.19150284317979774 (mean)
Sample silhouette score > 0.5: 8

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

Selecting best *K*

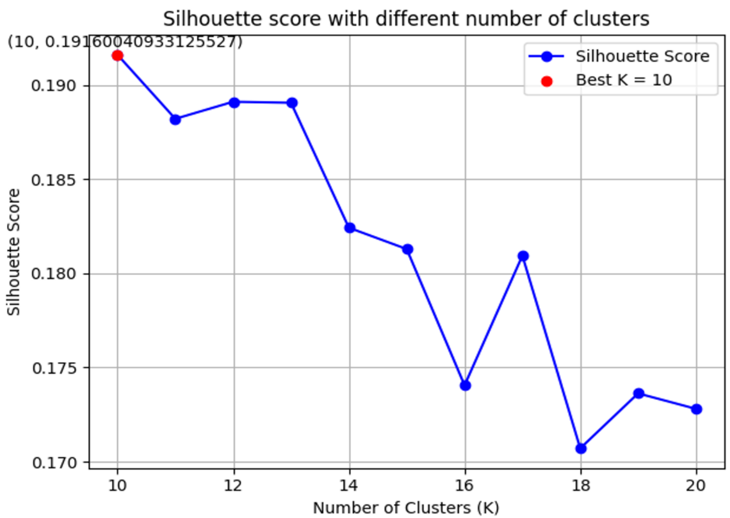


Figure 5: Performance of each *K* in closed set [10,20]

- Label each cluster by ***Majority Vote***

	Digit
0	5
1	0
2	2
3	8
4	6
5	3
6	7
7	4
8	1
9	1

Figure 6: Class for each cluster

3. Hierarchical clustering

Steps:

1. Start with Each Data Point as a Cluster: Initially, each data point is treated as its own cluster.
2. Compute Distances: Calculate the linkage distance for all pairs of clusters using the provided formula.
1. Merge Clusters: Merge the pair of clusters with the smallest linkage distance.
2. Update Distances: Recompute distances between the new cluster and all other clusters.
3. Repeat: Continue merging clusters until all data points are in a single cluster or the desired number of clusters is reached.

Linkage Methods:

1. Single Linkage: Minimum distance between points in different clusters.
2. Complete Linkage: Maximum distance between points in different clusters.
3. Average Linkage: Average distance between all pairs of points in different clusters.
4. Ward's Method: Minimizes the total within-cluster variance.

$$[d(A,B) = \frac{W_A W_B}{W_A + W_B} d^2(G_A, G_B)]$$

(W_A) and (W_B) are the cardinalities (sizes) of clusters (A) and (B), respectively.
(G_A) and (G_B) are the centroids (centers of gravity) of clusters (A) and (B), respectively.
$(d(G_A, G_B))$ is the Euclidean distance between the centroids of clusters (A) and (B).

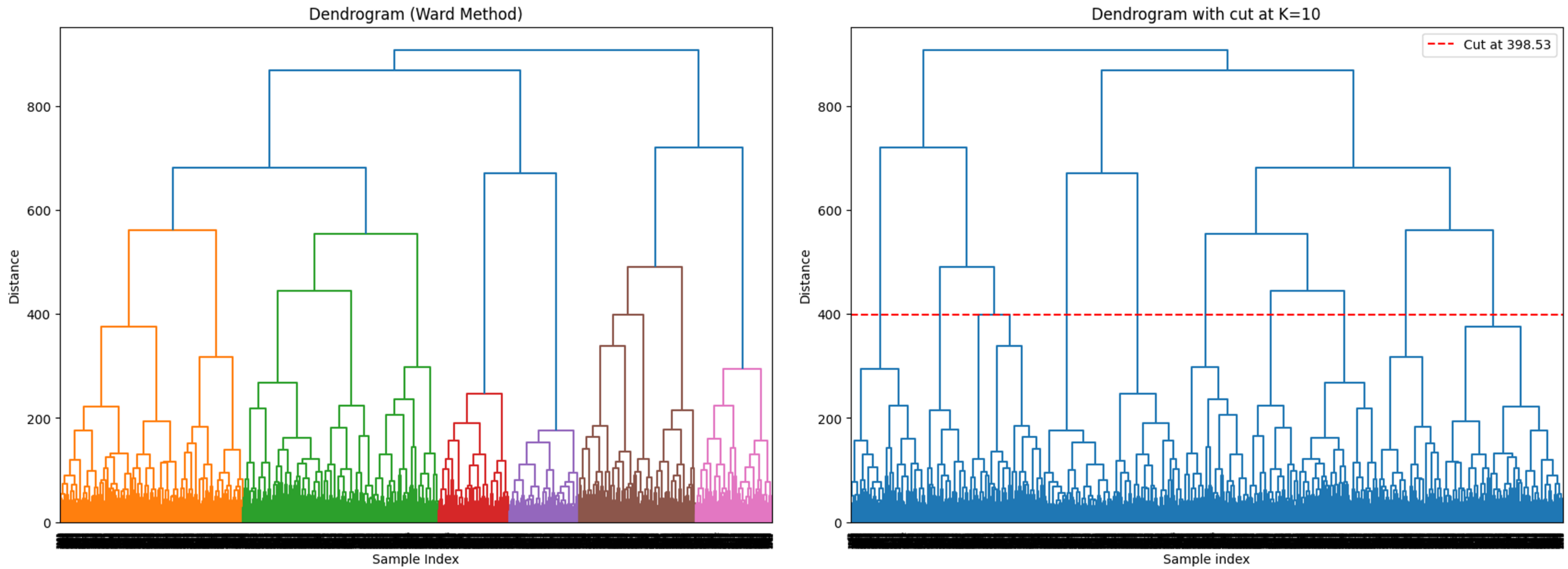


Figure 7: Dendrogram

Hierarchical Clustering for *K* = 10

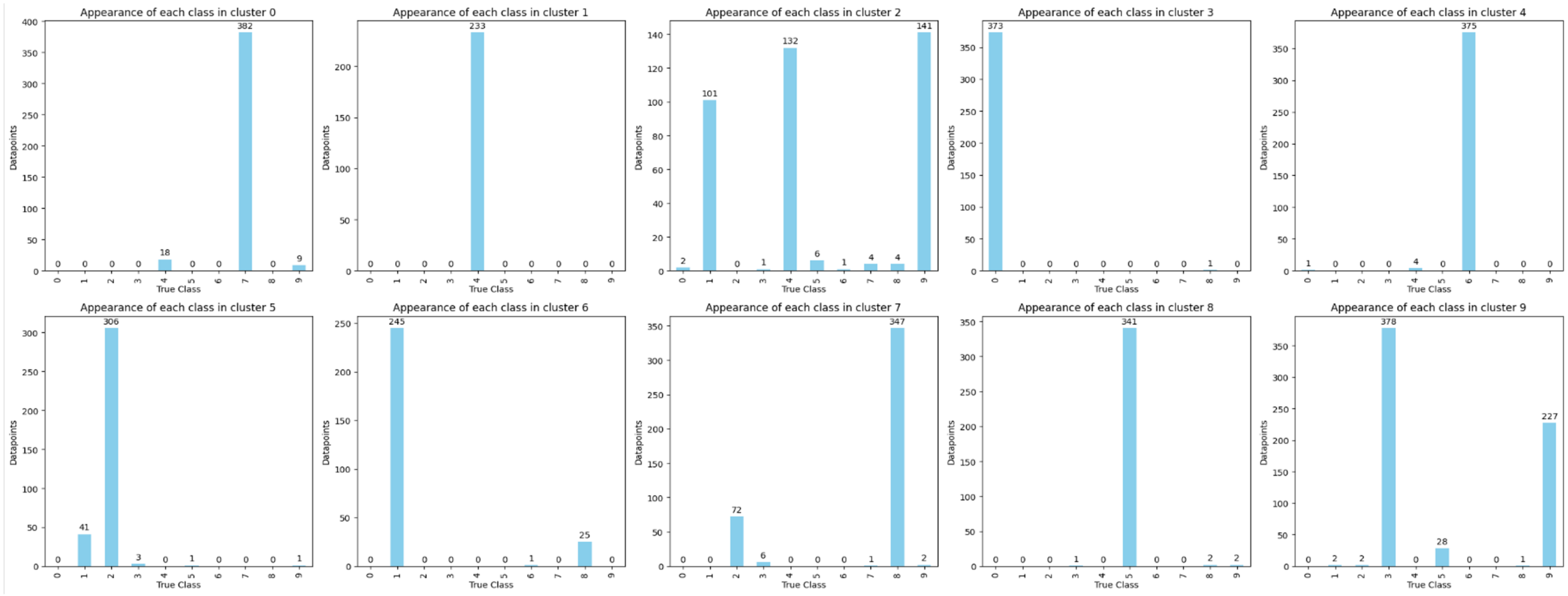


Figure 8: occurrence of classes in each cluster

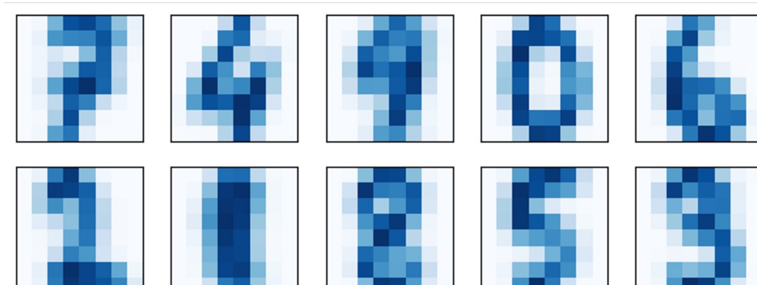


Figure 9: centroid plot for each cluster

Hierarchical Clustering performance from *K=10* to *K=20*

Silhouette score = 0.1645628407814536

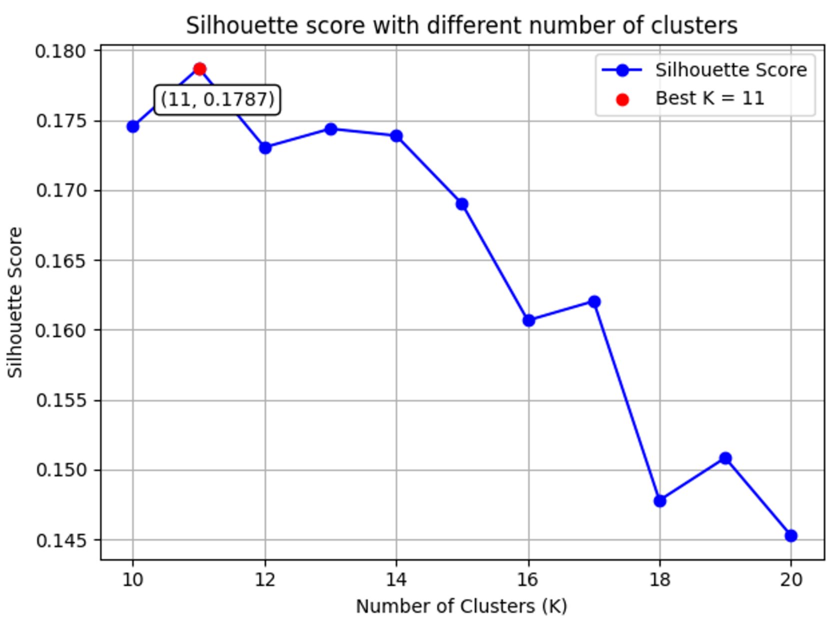


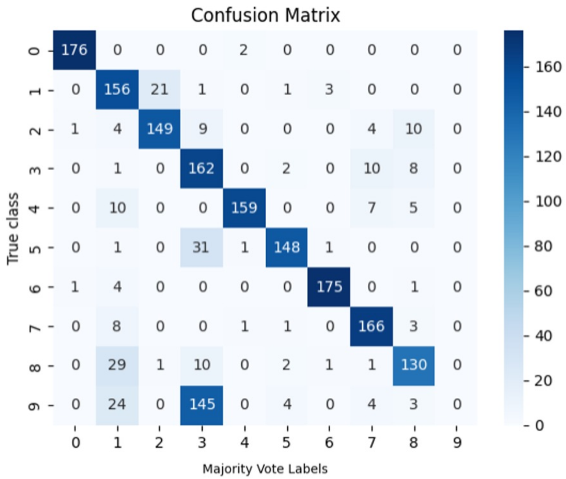
Figure 10: Performance of each *K* in closed set [10,20]

- Label each cluster by ***Majority Vote***

	Digit
0	7
1	4
2	1
3	9
4	0
5	6
6	2
7	1
8	8
9	5
10	3

Figure 11: Class for each cluster with *K=11*

4. Comparison



	precision	recall	f1-score	support
0	0.99	0.99	0.99	178
1	0.66	0.86	0.74	182
2	0.87	0.84	0.86	177
3	0.45	0.89	0.60	183
4	0.98	0.88	0.92	181
5	0.94	0.81	0.87	182
6	0.97	0.97	0.97	181
7	0.86	0.93	0.89	179
8	0.81	0.75	0.78	174
9	0.00	0.00	0.00	180
accuracy			0.79	1797
macro avg	0.75	0.79	0.76	1797
weighted avg	0.75	0.79	0.76	1797

----- Overall Performance -----

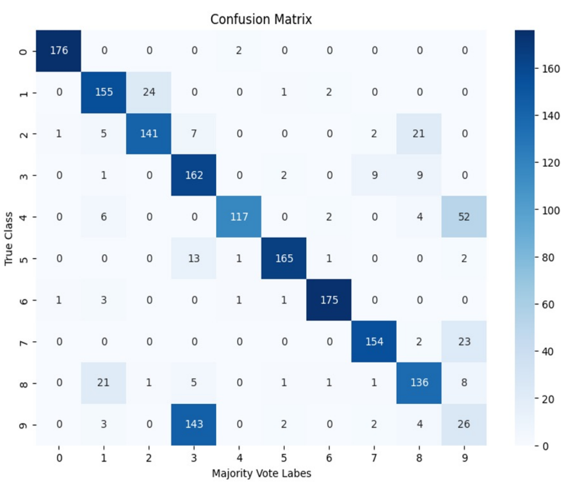
Accuracy: 0.7908

Precision: 0.7532

Recall: 0.7906

F1-score: 0.7626

Figure 12: Global performance of *K*-means



	precision	recall	f1-score	support
0	0.99	0.99	0.99	178
1	0.80	0.85	0.82	182
2	0.85	0.80	0.82	177
3	0.49	0.89	0.63	183
4	0.97	0.65	0.77	181
5	0.96	0.91	0.93	182
6	0.97	0.97	0.97	181
7	0.92	0.86	0.89	179
8	0.77	0.78	0.78	174
9	0.23	0.14	0.18	180
accuracy			0.78	1797
macro avg	0.79	0.78	0.78	1797
weighted avg	0.79	0.78	0.78	1797

----- Overall Performance -----

Accuracy: 0.7830

Precision: 0.7945

Recall: 0.7829

F1-score: 0.7784

Figure 13: Global performance of *Hierarchical Clustering*

5. Experiment

- Transformations that were tried on the dataset:
  - Polynomial Features
  - Exponential / Logarithmic transformation
  - Principal Component Analysis (with random number of components: 17, 20, 28, ...)
- The best result obtained is single polynomial features (degree = 2) or X^2.
- By varying number of clusters *(K)* in set [10,20] and evaluating each *K* using silhouette score, the chosen *K* are always range from 10 to 14 with performances describe as below:

Number of clusters (K)	Silhouette score	Accuracy
K = 10	0.15951794567609953	0.8553
K = 11	0.1583355174143404	0.8542
K = 12	0.16032746002597156	0.8486
K = 13	0.15877647222943458	0.8553
K = 14	0.16145098846048	0.8614

Table 1: Performance of *K*-means for *K* in [10,14] with X^2 transformation

- Based on **Table 1**, *K = 14* has the best accuracy score, but, out of 15 times, the algorithm chose "*K = 10*" 7 times.

So, even if *K = 14* has better accuracy than *K = 10,* it is not a robust model since it will not be able to generalize on unseen data like *K = 10*. Hence, we will choose ***K = 10*** in this project and we will see the difference between X and X^2 features.

- Why X^2 performs better than X?

-> Each feature or predictor ($X_{i=1,2,\ldots,64}$) takes value in set [0,16], so K-means will not be able to separate well between, for example, 14 and 16. However, $12^2 = 144$ and $16^2 = 256$ will make a big difference since the algorithm is based on the euclidean distance. More precisely, the difference between the two values turned from "16-12 = 4" to "256-144 = 112".

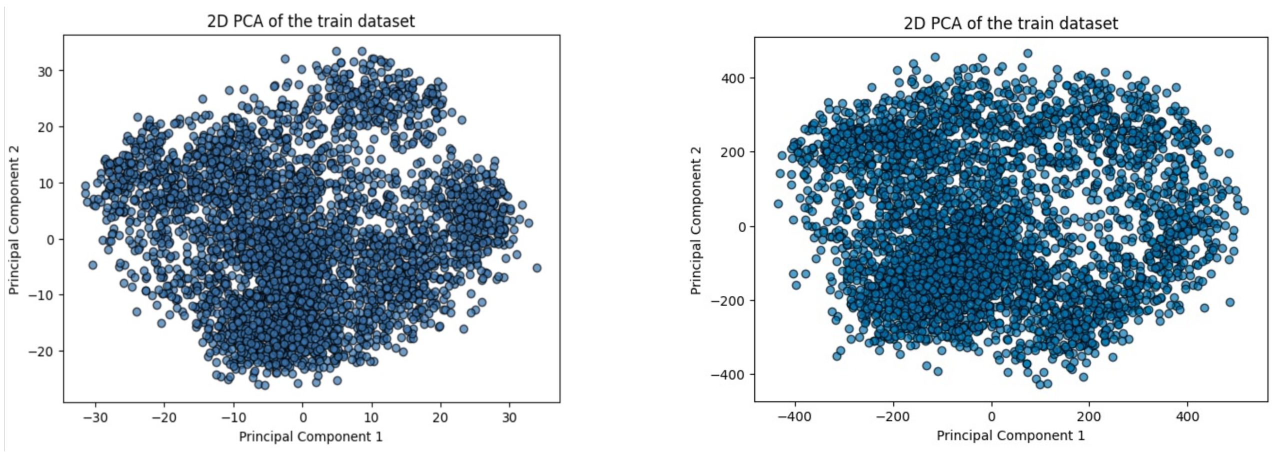


Figure 14: 2D representation of X (left) X^2 (right) features using PCA

- We can see that the X^2 features' data points are more spread and have bigger values.

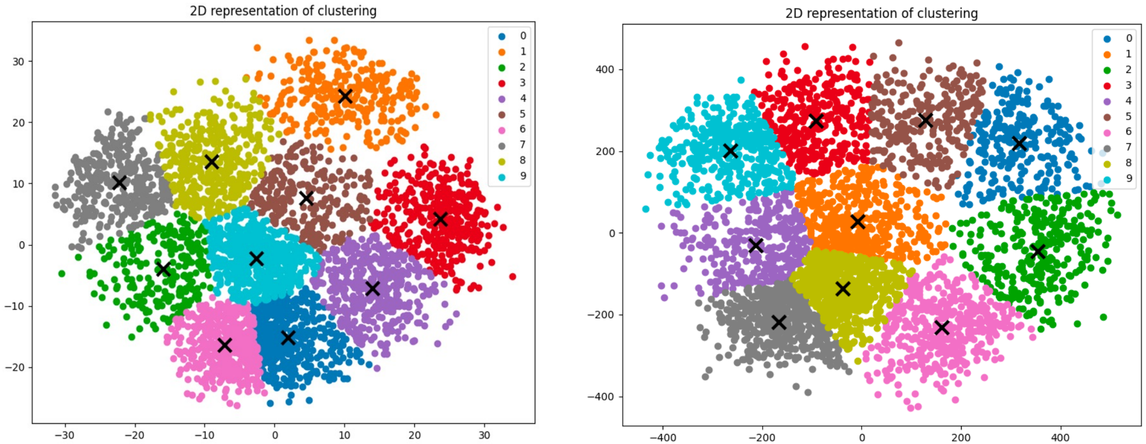


Figure 15: Clustering of X (left) X^2 (right) features

- Again, we can see that the X^2 features' data points are widespread (well clustered).

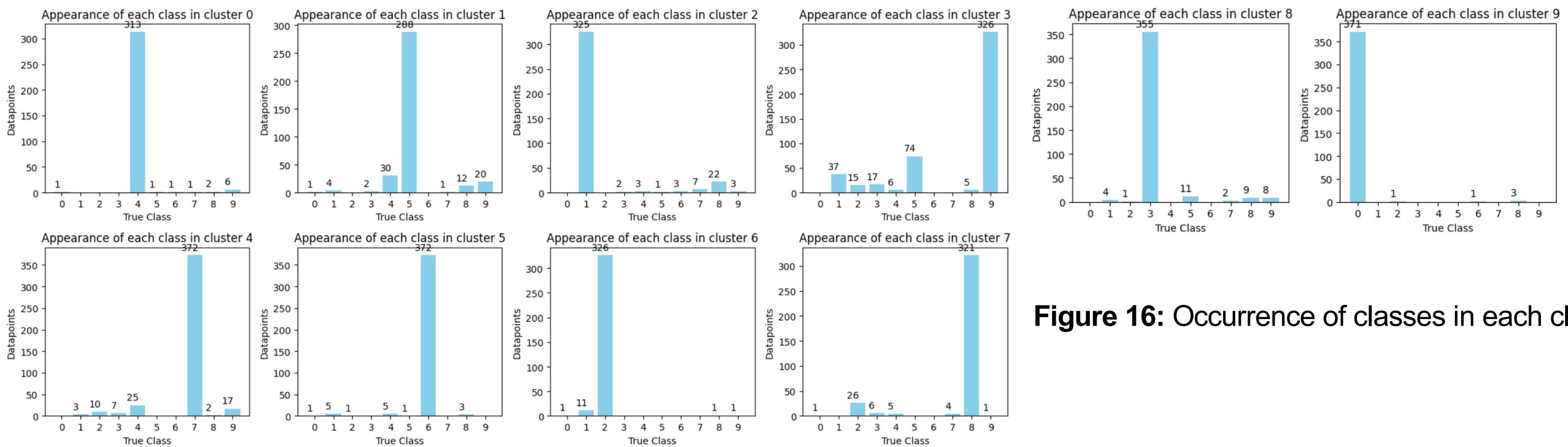


Figure 16: Occurrence of classes in each cluster for (X^2 predictors)

- Even though there are many classes appeared in each cluster, noticeably *cluster 1, 2, 3, 4, 7, and 8,* we can clearly see that there is only one dominant class (outnumbered the other classes) unlike before *(K*-means without transformation*)* in which it implies that the algorithm is now capable of differentiate different handwritten digits.

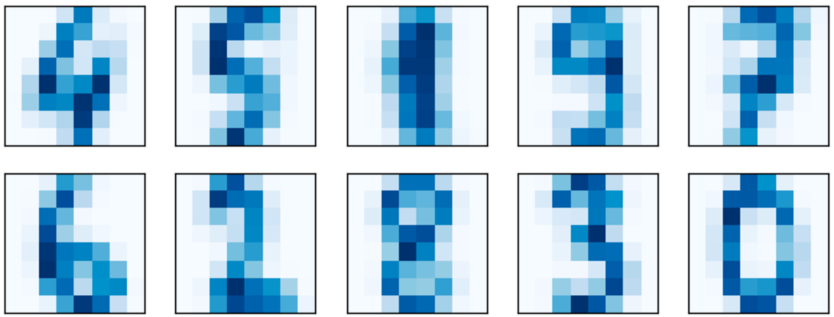
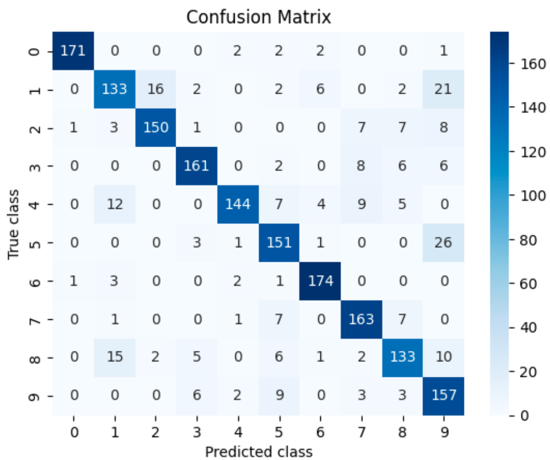


Figure 18: Centroid plot for each cluster (X^2 features)

	Digit
0	4
1	5
2	1
3	9
4	7
5	6
6	2
7	8
8	3
9	0

Figure 17: Class for each cluster by ***Majority Vote*** for K = 10

- Based on **Figure 18**, we can see that the digits are now more visible than before since there is no mixture between 3 and 9 for example.




```
              precision    recall  f1-score   support

           0       0.99      0.96      0.97       178
           1       0.80      0.73      0.76       182
           2       0.89      0.85      0.87       177
           3       0.90      0.88      0.89       183
           4       0.95      0.80      0.86       181
           5       0.81      0.83      0.82       182
           6       0.93      0.96      0.94       181
           7       0.85      0.91      0.88       179
           8       0.82      0.76      0.79       174
           9       0.69      0.87      0.77       180

    accuracy                           0.86      1797
   macro avg       0.86      0.86      0.86      1797
weighted avg       0.86      0.86      0.86      1797
```

```
----- Overall Performance -----
Accuracy: 0.8553
Precision: 0.8613
Recall: 0.8552
F1-score: 0.8560
```

Figure 19: Global performance of X^2 transformation (*K-means* for *K* = 10)

- The model is now capable of predicting digit 9.

6. Conclusion

- Understanding more about the **clustering.**
- Without feature engineering both models perform are inconsistent.
- There is still some mistakes for the classification, but 85% accuracy is not bad.
- Improvements:
  - Try combination of **different features engineering.**
  - Play around with other features engineering like ***Fourier*** transformation, etc.
  - Consider other types of distance than ***Euclidean*** distance such as ***Mahalanobis*** distance, etc.

Thank you!