# Applying K-Means Clustering on Hotness Classification

bdss lab _ undergraduated intern 정준혁

25. 2. 6.Thu.

# Contents

1. **Goal**

2. **Data Classification**

3. **K-means Clustering Algorithm**

4. **Review & Future Plan**

Details exist in each Chapters !

# 1. Goal

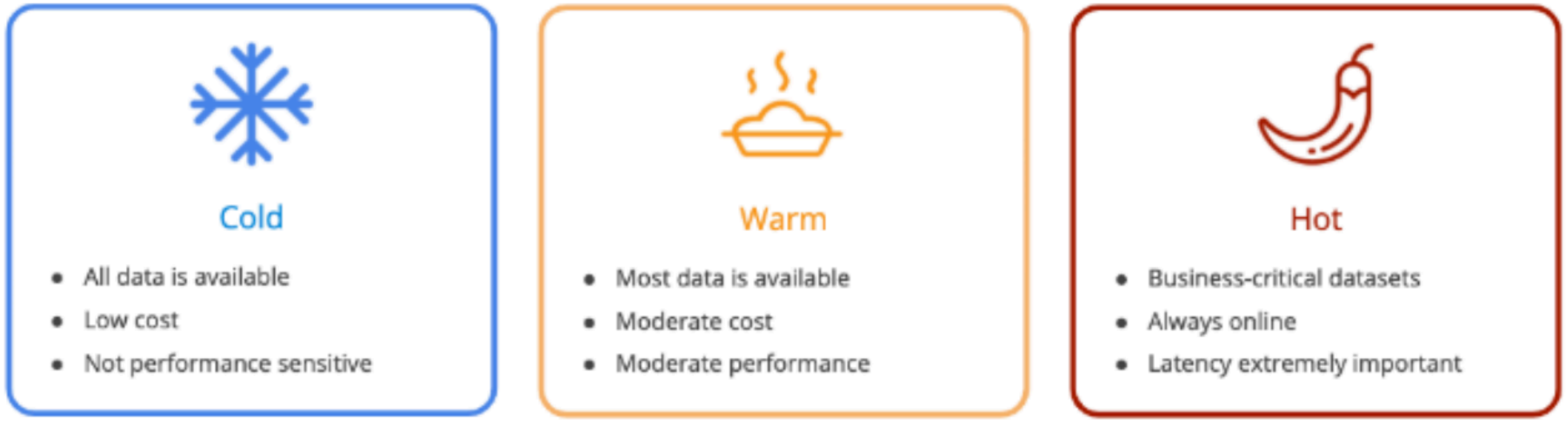
- Demand of SSD is improving nowadays, and it has a particular feature, which is called ‘garbage collection’. It occurs ‘write amplification’, and it makes life-cycle to down.

- Many researchers go ahead the study(classify data according to I/O access pattern) to reduce GC overhead.

- And we expect to find more sophisticated method.

# 2. Data Classification

## 2-1. Hot & Cold & Warm



Cold

- All data is available
- Low cost
- Not performance sensitive

Warm

- Most data is available
- Moderate cost
- Moderate performance

Hot

- Business-critical datasets
- Always online
- Latency extremely important

# 2. Data Classification

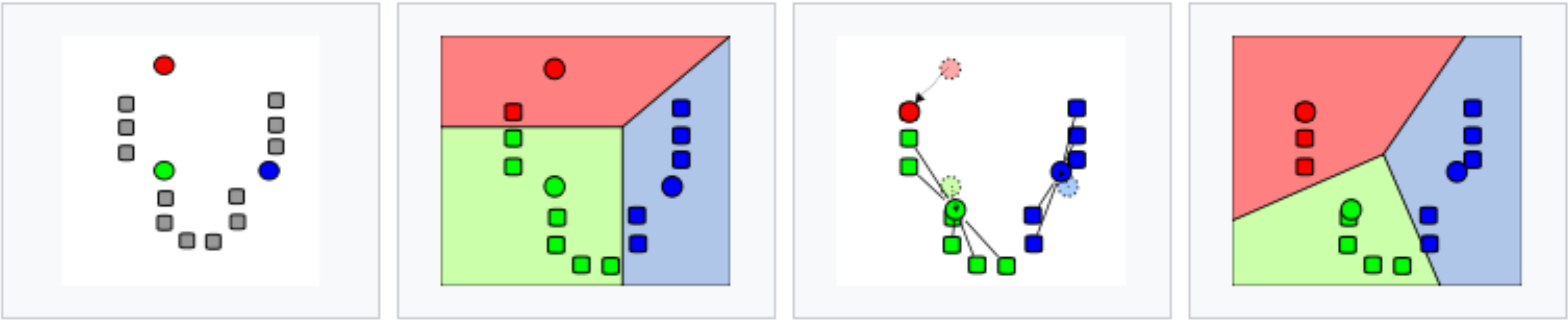
## 2-2. Purpose of Data Classification

- The arrangement of similar-state data helps to avoid the write amplification.
- Criteria of classification is ‘Hotness’

# 3. K-means Clustering Algorithm

## 3-1. Concepts

Demonstration of the standard algorithm

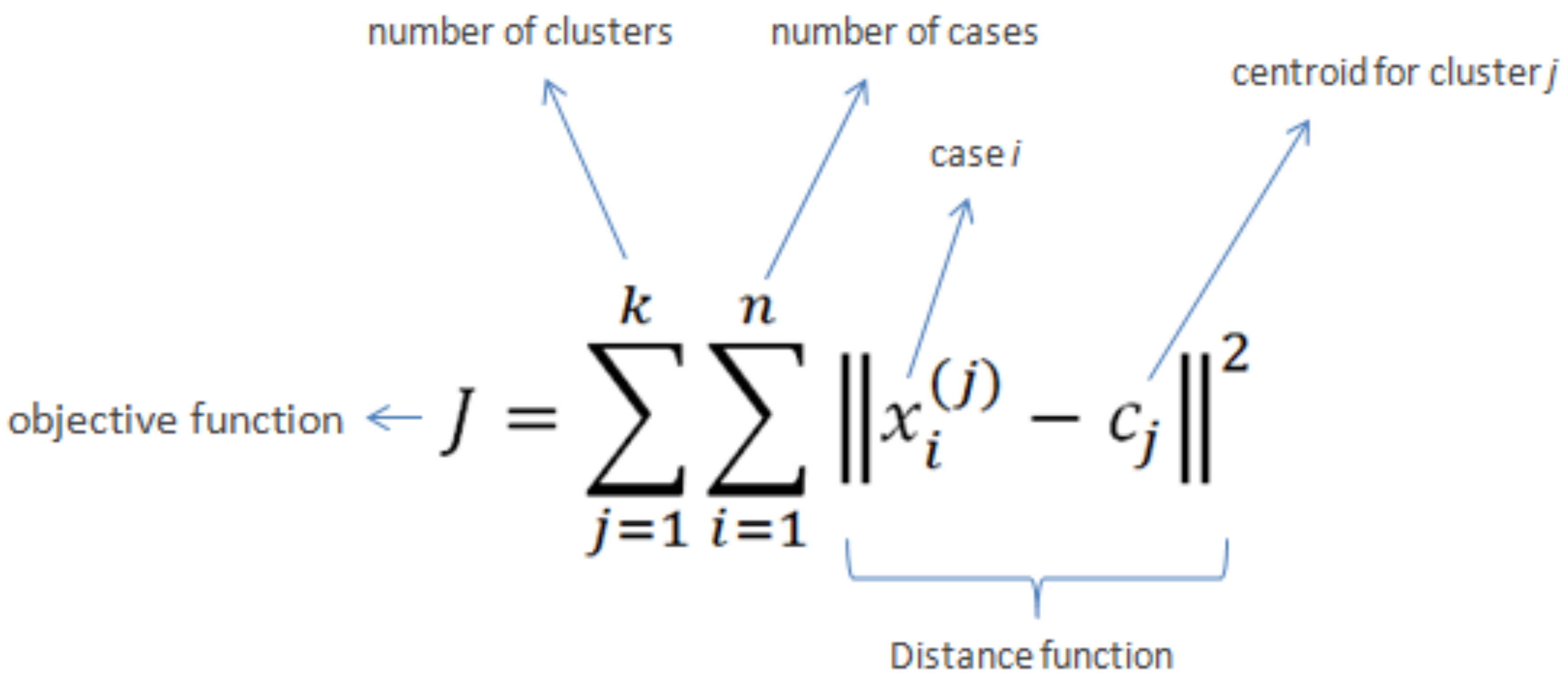


1. *k* initial "means" (in this case *k*=3) are randomly generated within the data domain (shown in color).

2. *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.

3. The centroid of each of the *k* clusters becomes the new mean.

4. Steps 2 and 3 are repeated until convergence has been reached.

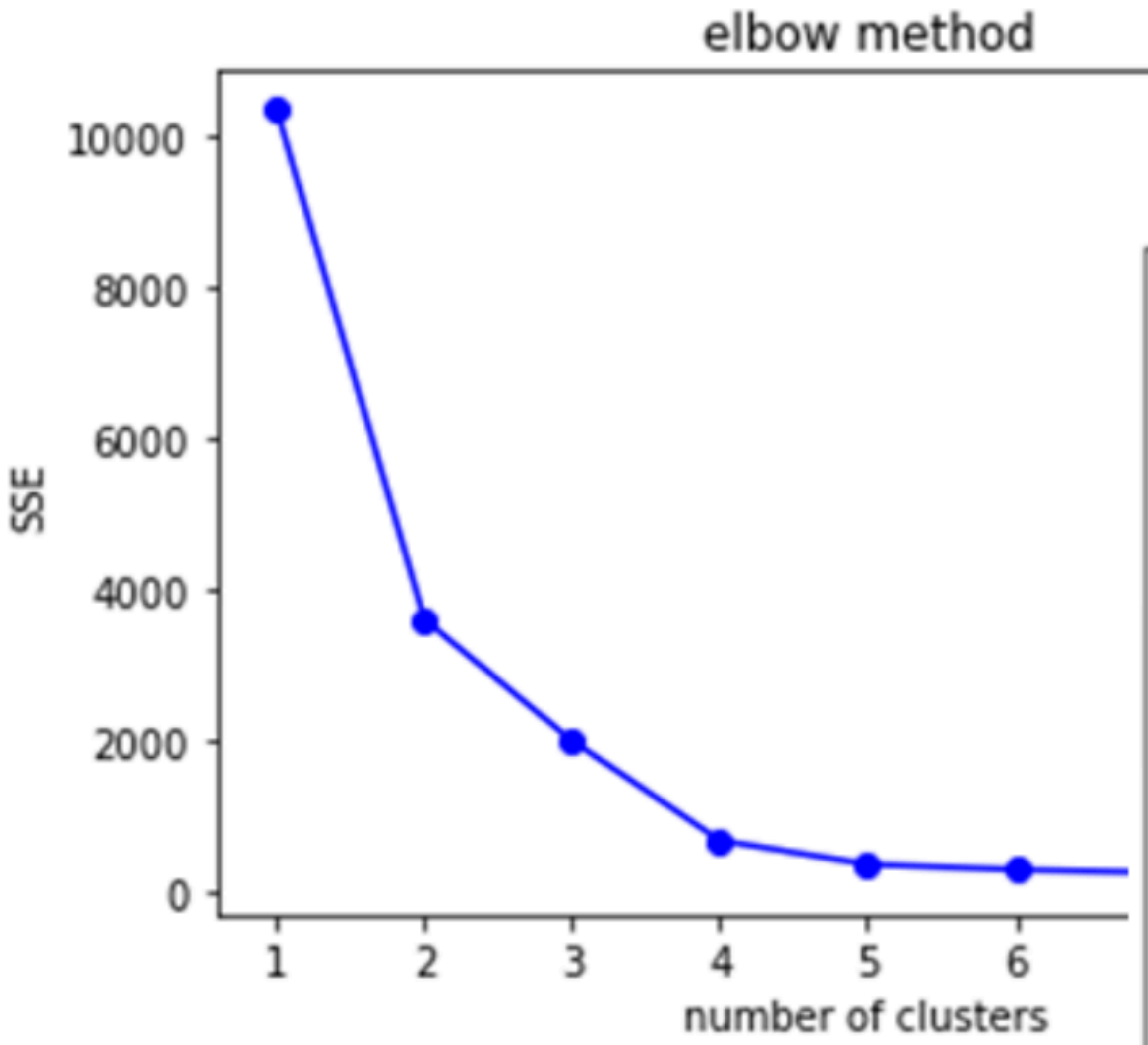


$$\text{objective function} \leftarrow J = \sum_{j=1}^{k}\sum_{i=1}^{n} \underbrace{\left\| x_i^{(j)} - c_j \right\|^2}_{\text{Distance function}}$$

Core goal : Maximize the cohesion of data within each cluster & maximize the separation of clusters.

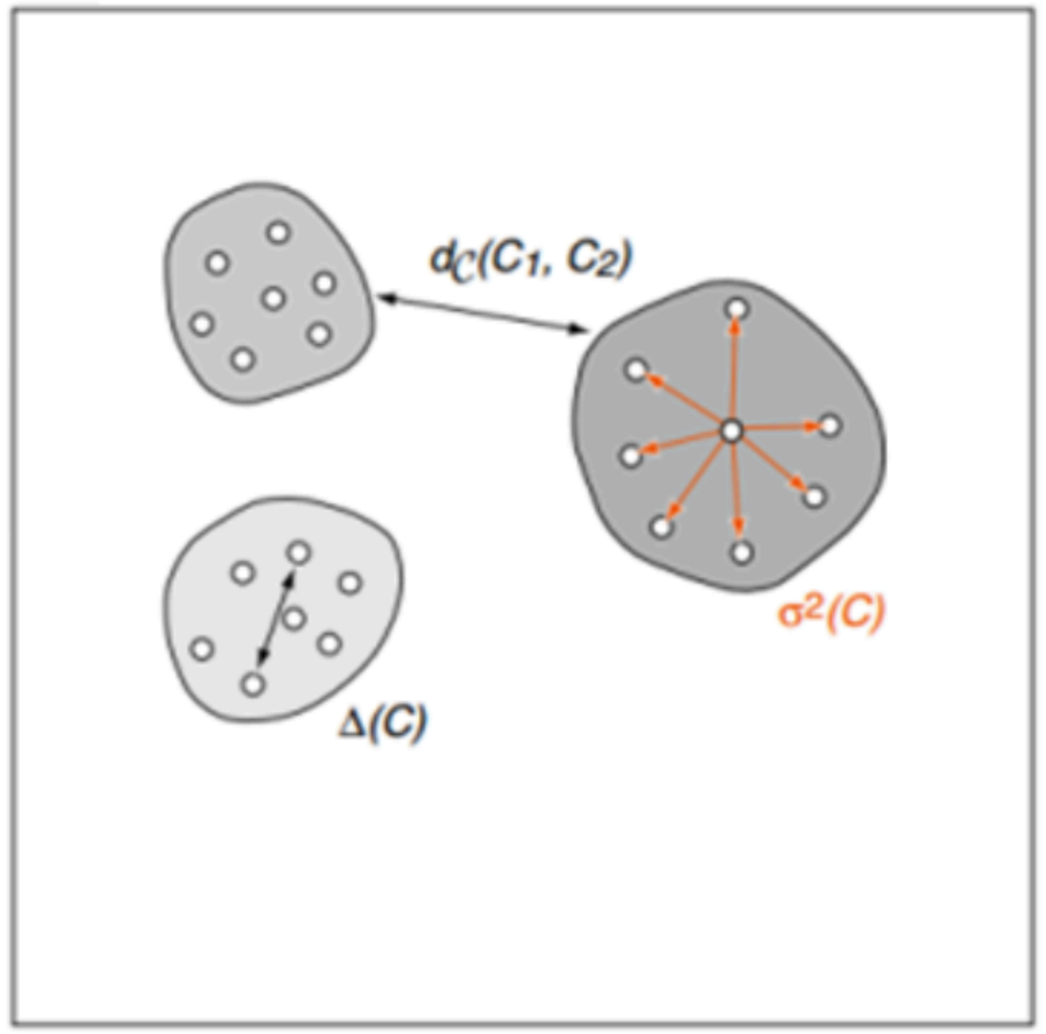
# 3. K-means Clustering Algorithm

## 3-2. Elbow Method : Find the optimized number(k) of Clusters



$$SSE = \sum_{i=1}^{k}\sum_{x\in c_i} dist(x,c_i)^2$$

SSE 공식

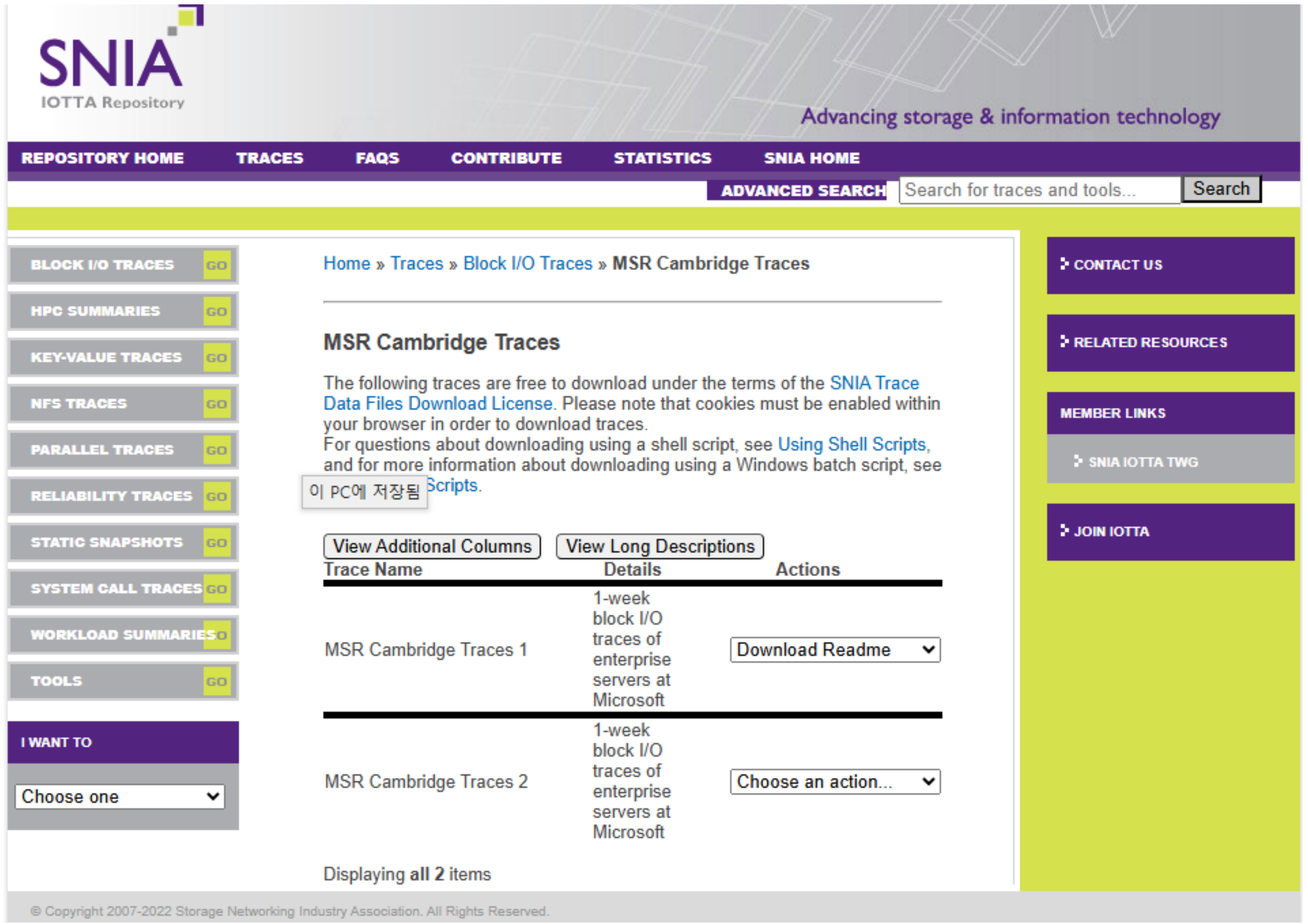


- Find the moment When SSE decreases the most

- Easy to implement

# 3. K-means Clustering Algorithm

## 3-3. Prepare files and Example implementation



Home » Traces » Block I/O Traces » **MSR Cambridge Traces**

**MSR Cambridge Traces**

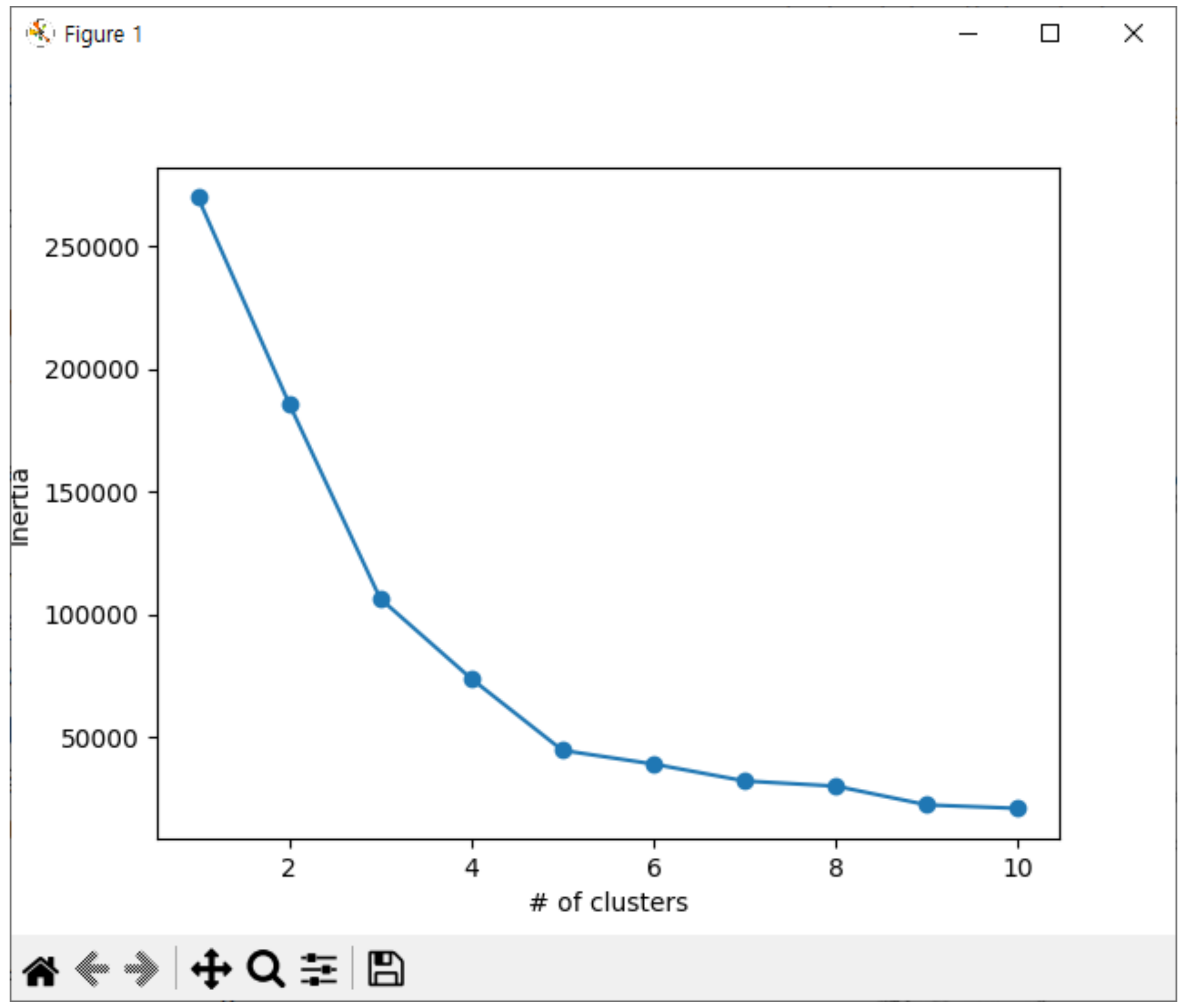
The following traces are free to download under the terms of the SNIA Trace Data Files Download License. Please note that cookies must be enabled within your browser in order to download traces.
For questions about downloading using a shell script, see Using Shell Scripts, and for more information about downloading using a Windows batch script, see [illegible] Scripts.

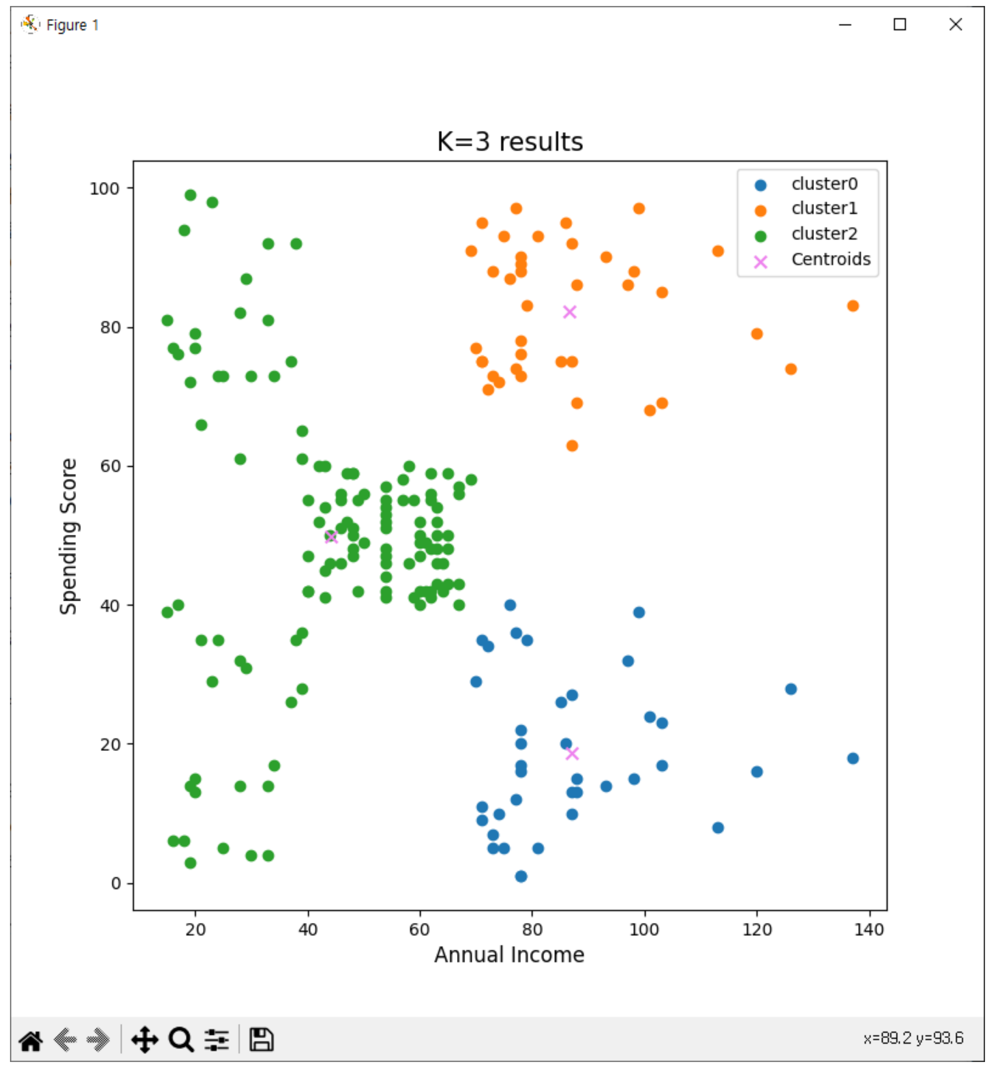
| Trace Name | Details | Actions |
|---|---|---|
| MSR Cambridge Traces 1 | 1-week block I/O traces of enterprise servers at Microsoft | Download Readme |
| MSR Cambridge Traces 2 | 1-week block I/O traces of enterprise servers at Microsoft | Choose an action... |

Displaying **all 2** items

© Copyright 2007-2022 Storage Networking Industry Association. All Rights Reserved.

mscambridge trace

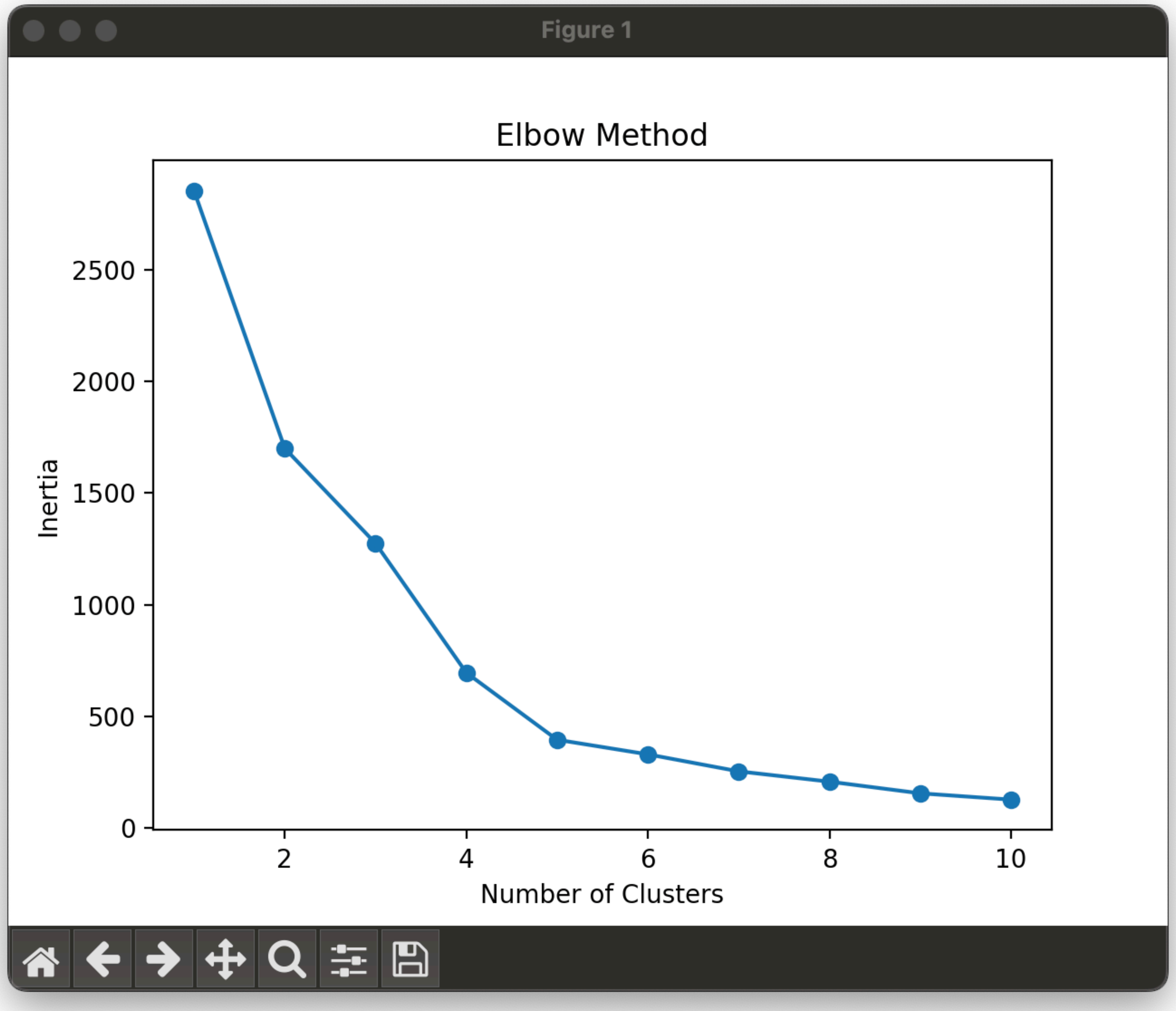


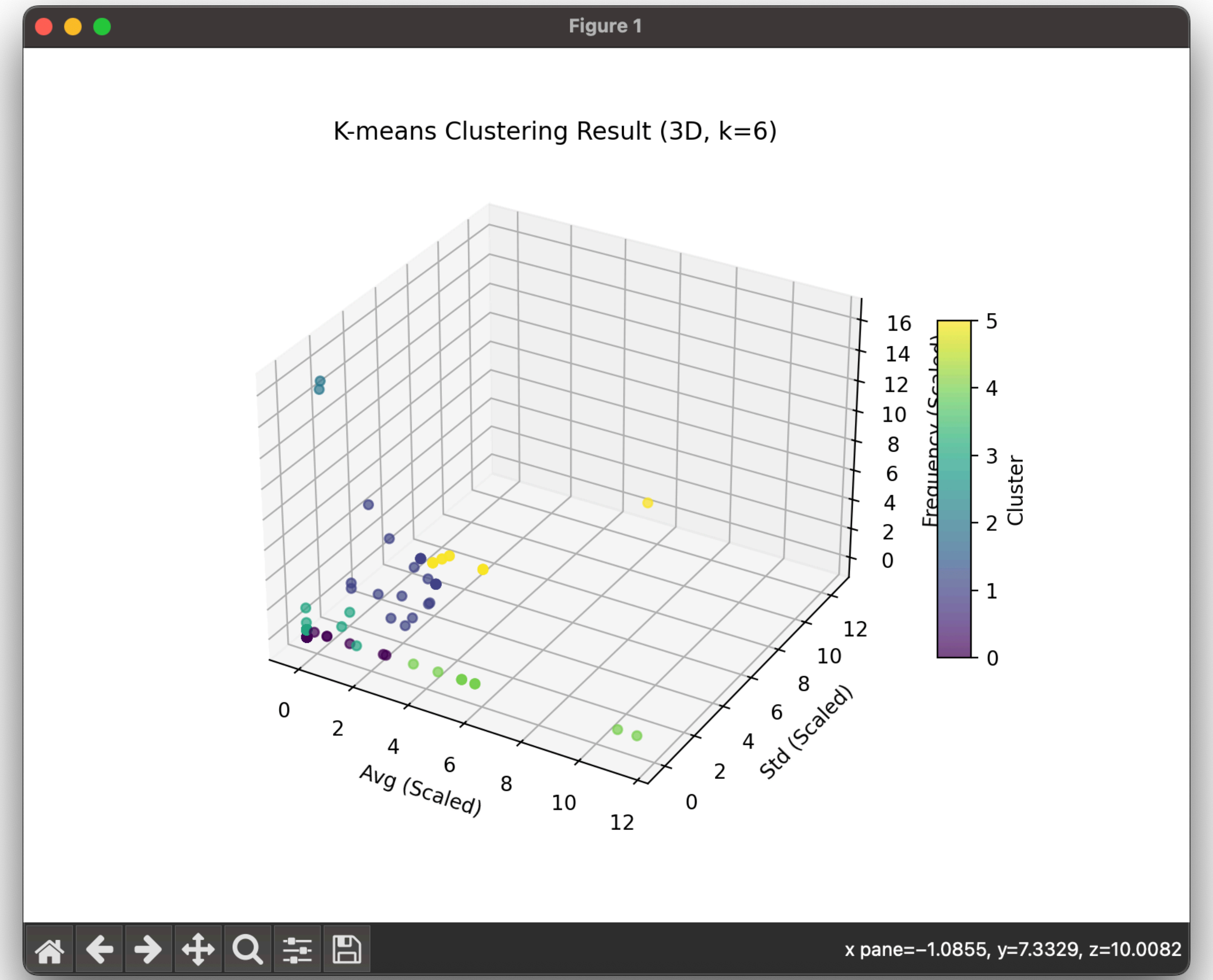


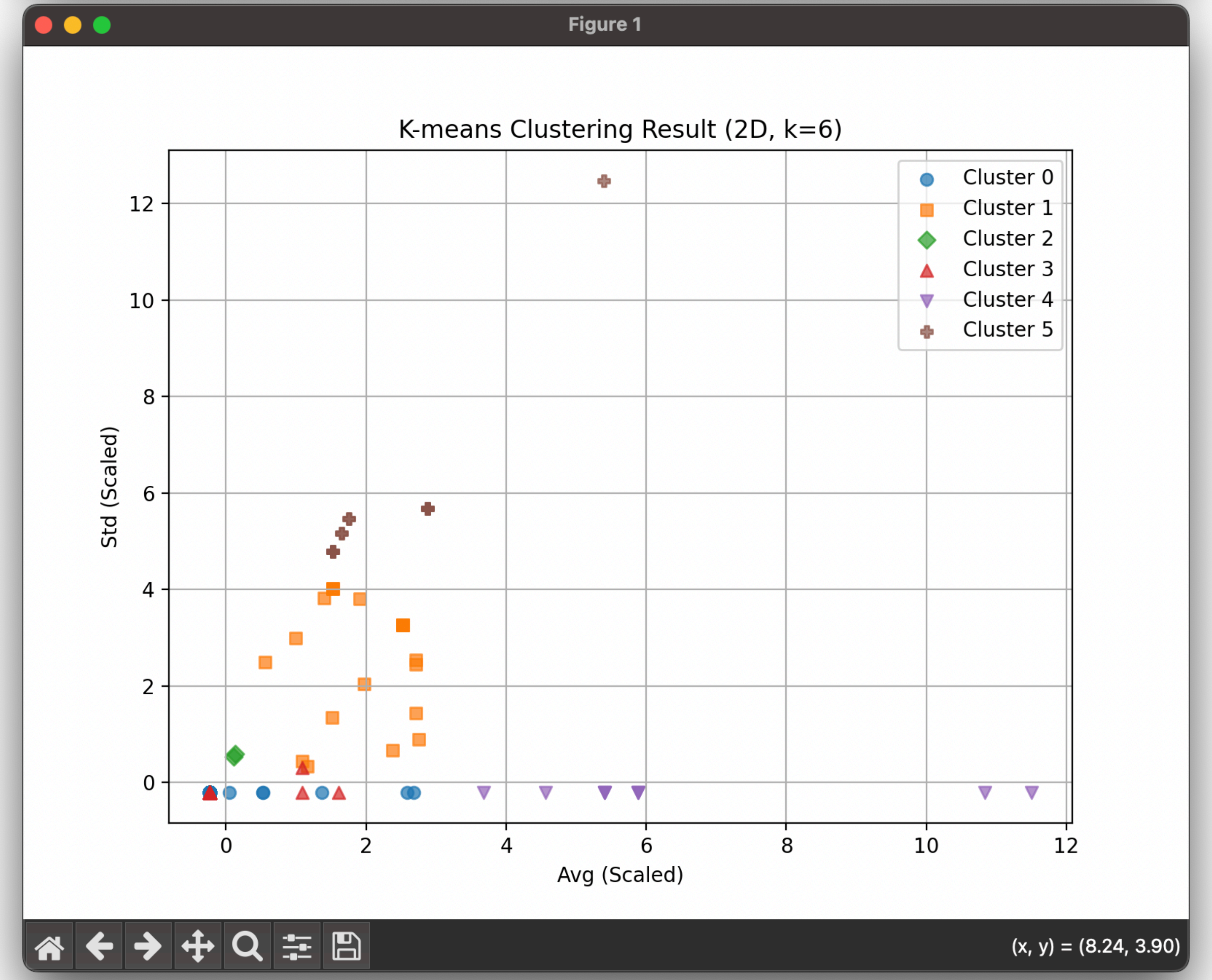
Ex file : ‘Mall_Customer.csv’

# 3. K-means Clustering Algorithm

## 3-4. Result(..?)

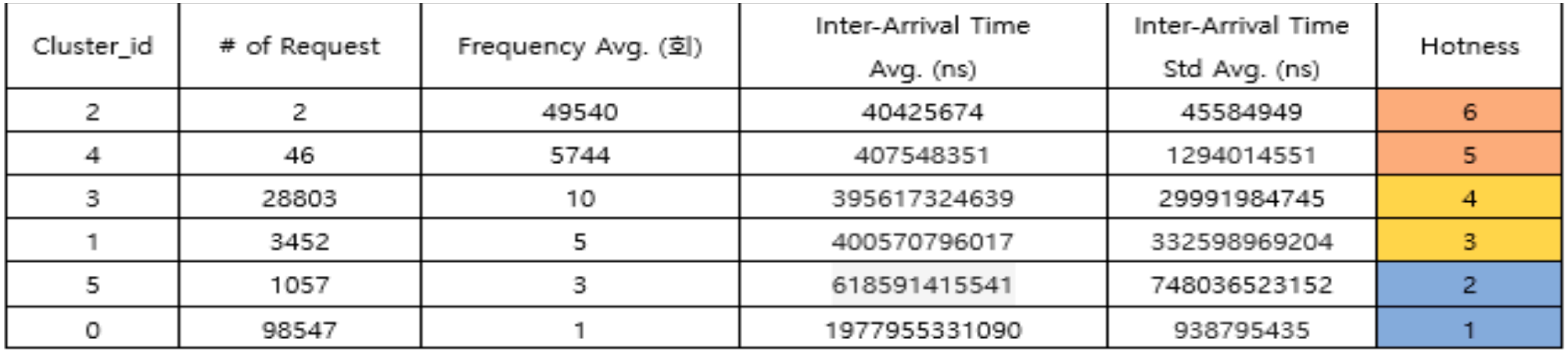






# 3. K-means Clustering Algorithm

## 3-4. Result(..?)



| Cluster_id | # of Request | Frequency Avg. (회) | Inter-Arrival Time Avg. (ns) | Inter-Arrival Time Std Avg. (ns) | Hotness |
|---|---|---|---|---|---|
| 2 | 2 | 49540 | 40425674 | 45584949 | 6 |
| 4 | 46 | 5744 | 407548351 | 1294014551 | 5 |
| 3 | 28803 | 10 | 395617324639 | 29991984745 | 4 |
| 1 | 3452 | 5 | 400570796017 | 332598969204 | 3 |
| 5 | 1057 | 3 | 618591415541 | 748036523152 | 2 |
| 0 | 98547 | 1 | 1977955331090 | 938795435 | 1 |

Classification results when progressed normally

# 4. Review & Future Plan

**Review**

- What caused the inaccurate results ?
- > Suspected candidates : Scaling, **Data missing**
- We can use a variety of learning methods for classification.

**What I learned**

- Main Idea : Efficient lifespan management through classification according to number of updates.

- Unsupervised Learning: K-Means Clustering

# 4. Review & Future Plan

**Review**

- First application after learning ML concept
- First study of ML paper
- Necessity to study more about Python

**Future Plan**

- Read the paper, ‘Reliable Storage Study with ML’
- Language review(C, Py) for smooth implementation
- OS ..?

# References

- SSD 쓰기 증폭을 줄이기 위한 머신러닝 기반 정교한 Hotness 분류 방안 : https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE11036160\

- K-Means Clustering : en.wikipedia.org/wiki/K-means_clustering

- K-Menas 알고리즘 : velog.io/@eogns1208/K-Means-알고리즘