2022년 가을학기

Clustering

Lecture 09

Clustering

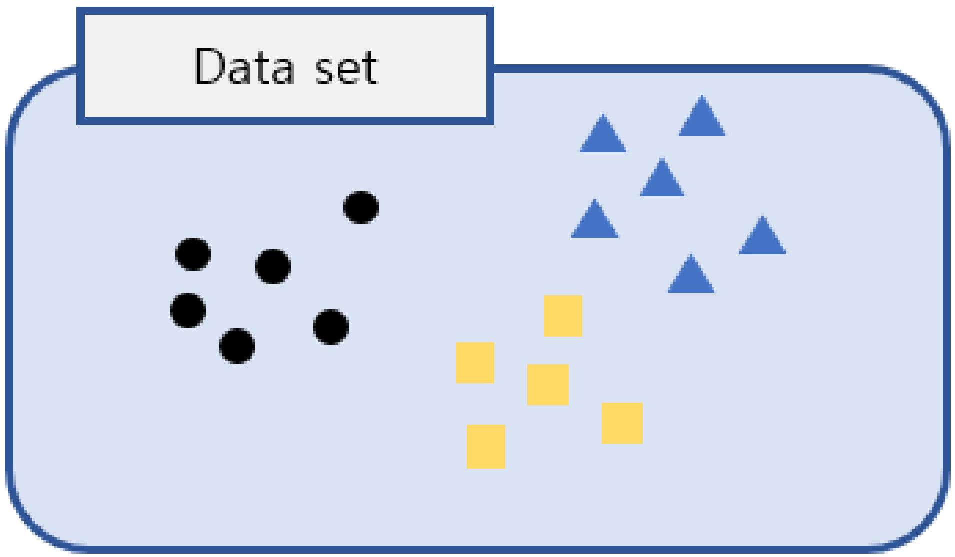
CONTENTS

What is Clustering?

Clustering Concept

❖ What is Clustering?

- The process of partitioning a set of data objects into subsets
  - A subset is called cluster

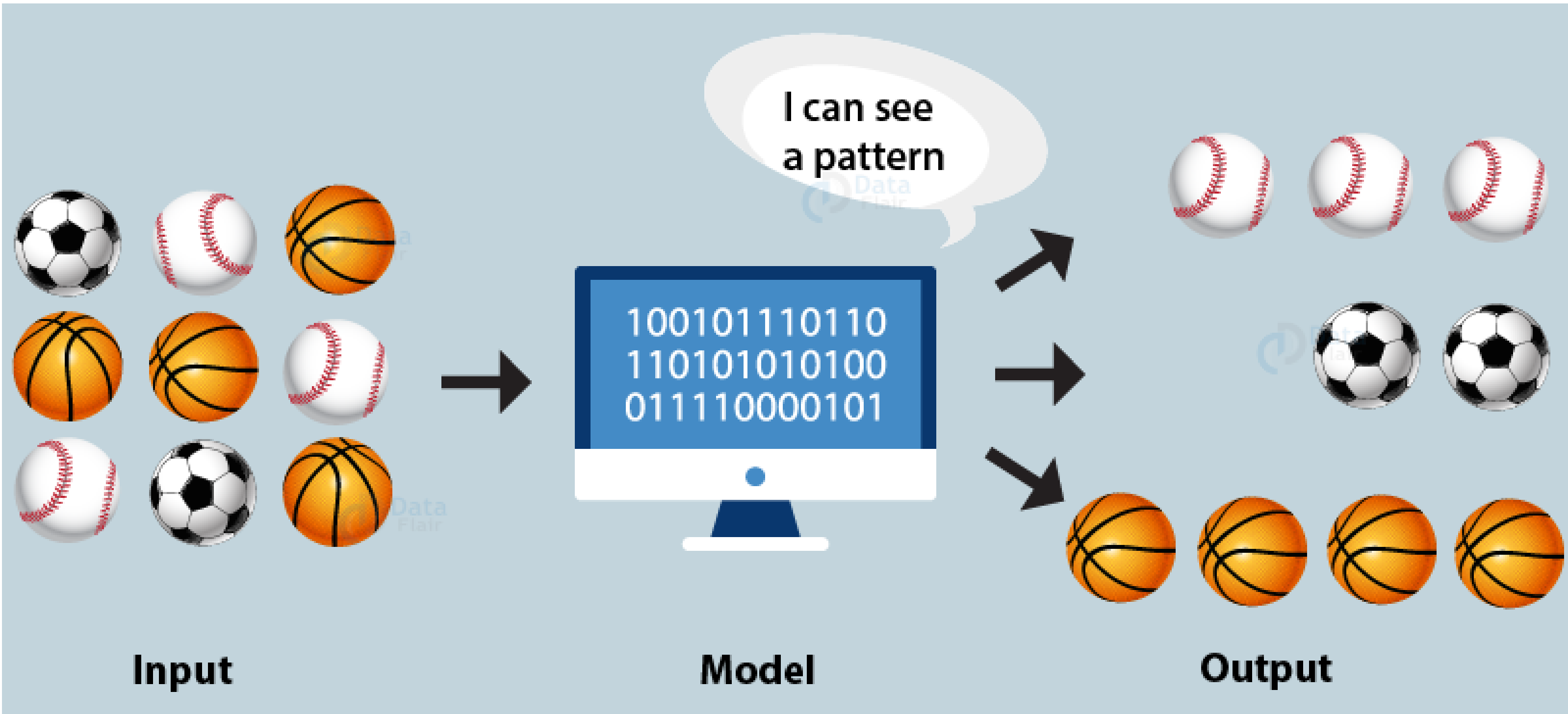


  - Clustering data objects that are similar to each other

Clustering Concept

❖ Clustering example

un supervi.

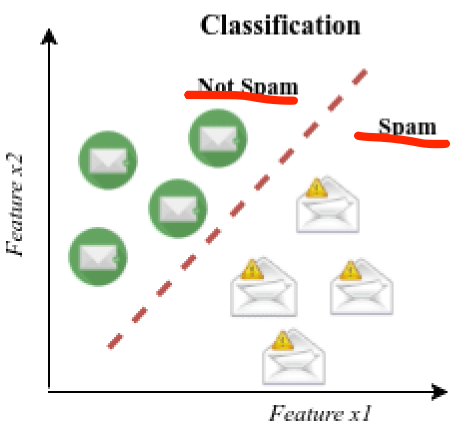


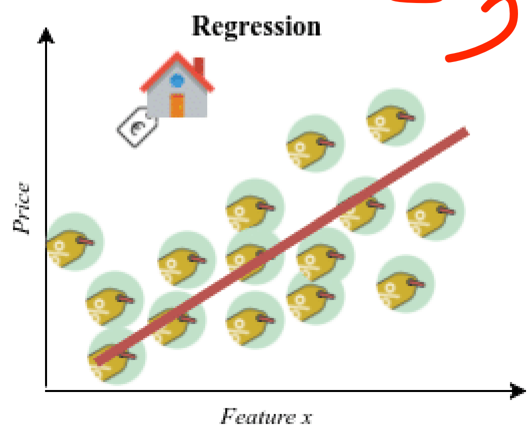
Clustering Concept

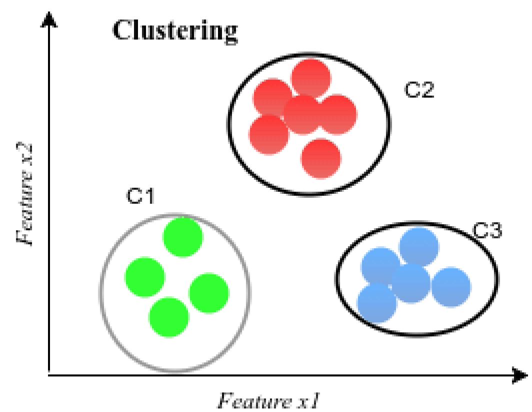
❖ Difference

- Classification: KNN, Decision Tree
- Regression: Linear and Logistic regression
- Clustering: K-Means, Agglomerative Filtering, DBSCAN

회귀분석







Clustering Concept

❖ Applications

- Segmenting customers into groups with similar demographics or buying patterns for targeted marketing campaigns

- Detecting anomalous behavior
  - Unauthorized network intrusions, by identifying patterns of use falling outside the known clusters

- Simplifying extremely large datasets 비슷한 성향:
  - Group features with similar values into a smaller number of homogeneous categories

B

Clustering Techniques

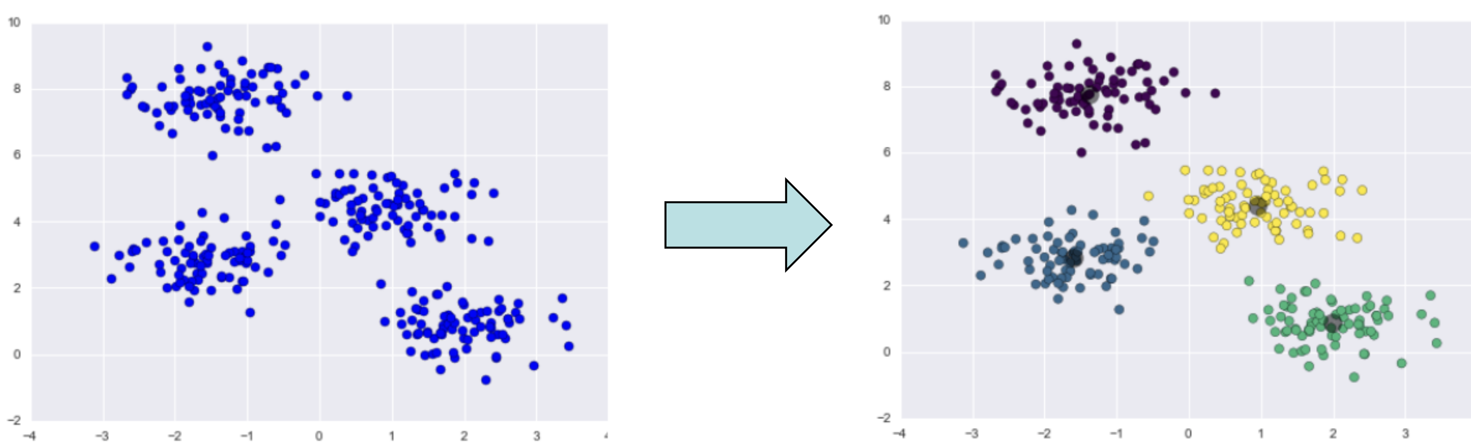
Clustering Techniques

❖ K-Means

- K is number of clusters

- A centroid-based technique
  - Centroid is the average of objects belonging to each cluster

- Groups each object with the closest centroid



Clustering Techniques

❖ K-Means Procedure

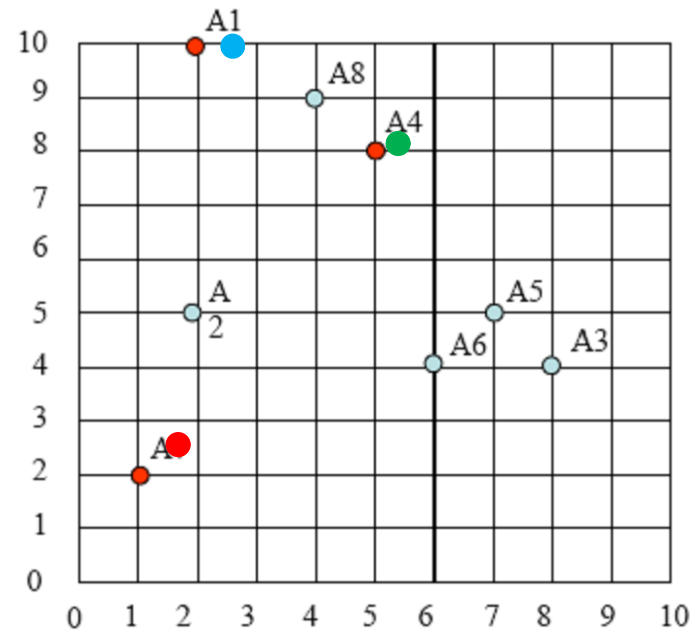
1. Step 1: Determine parameter k (k > 0)
2. Step 2: Randomly choose k points for starting centroids.
3. Step 3: Form k clusters by assigning all points to the closest centroid
4. Step 4: Recompute the centroid of each cluster (Calculate the mean of each cluster)
5. Step 5: repeat Step 3 until the centroid don’t change

Understanding clustering

❖ k-means example

- Step 1: Determine parameter *k* (*k* > 0)
- Step 2: Randomly choose *k* points for starting centroids.

		Point
C1	A1	(2, 10)
	A2	(2, 5)
	A3	(8, 4)
C2	A4	(5, 8)
	A5	(7, 5)
	A6	(6, 4)
C3	A7	(1, 2)
	A8	(4, 9)



Understanding clustering

❖ k-means example

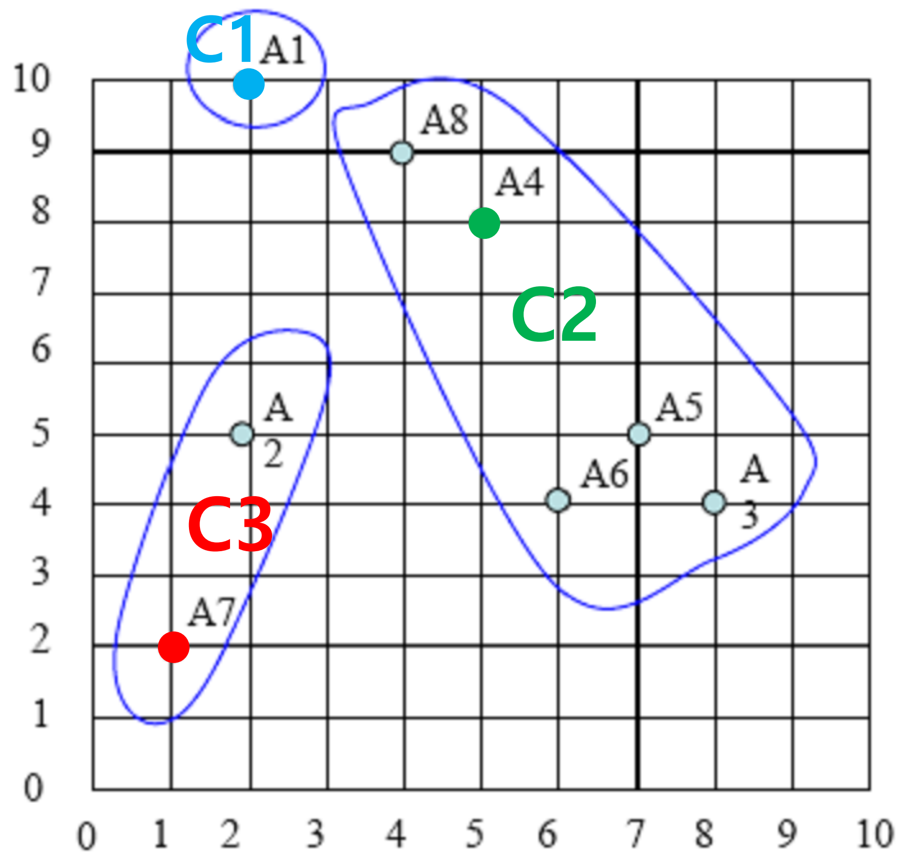
- Step 3: Form k clusters by assigning all points to the closest centroid

	Point	C1(2, 10)	C2(5, 8)	C3(1, 2)	Cluster
A1	(2, 10)	0	3.60	8.06	C1
A2	(2, 5)	5	4.24	3.16	C3
A3	(8, 4)	8.48	5	7.28	C2
A4	(5, 8)	3.60	0	7.21	C2
A5	(7, 5)	7.07	3.60	6.70	C2
A6	(6, 4)	7.21	4.12	5.38	C2
A7	(1, 2)	8.06	7.21	0	C3
A8	(4, 9)	2.23	1.41	7.61	C2

Understanding clustering

❖ k-means example

- Step 3: Form k clusters by assigning all points to the closest centroid



Understanding clustering

❖ k-means example

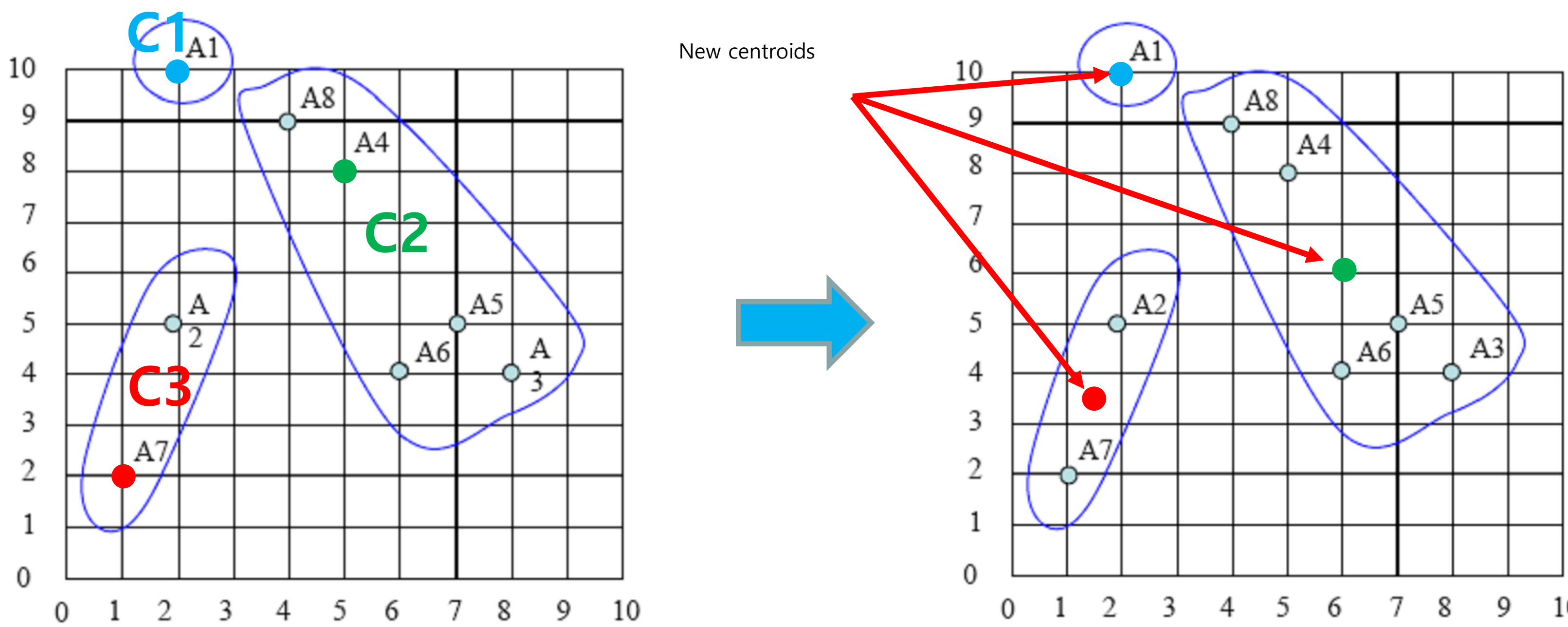
- Step 4: Recompute the centroid of each cluster (Calculate the mean of each cluster)

	Point	Cluster	Mean of C#
A1	(2, 10)	C1	(2, 10)
A3	(8, 4)	C2	(6, 6)
A4	(5, 8)	C2	
A5	(7, 5)	C2	
A6	(6, 4)	C2	
A8	(4, 9)	C2	
A7	(1, 2)	C3	(1.5, 3.5)
A2	(2, 5)	C3	

Understanding clustering

❖ k-means example

- Step 4: Recompute the centroid of each cluster (Calculate the mean of each cluster)



Understanding clustering

❖ k-means example

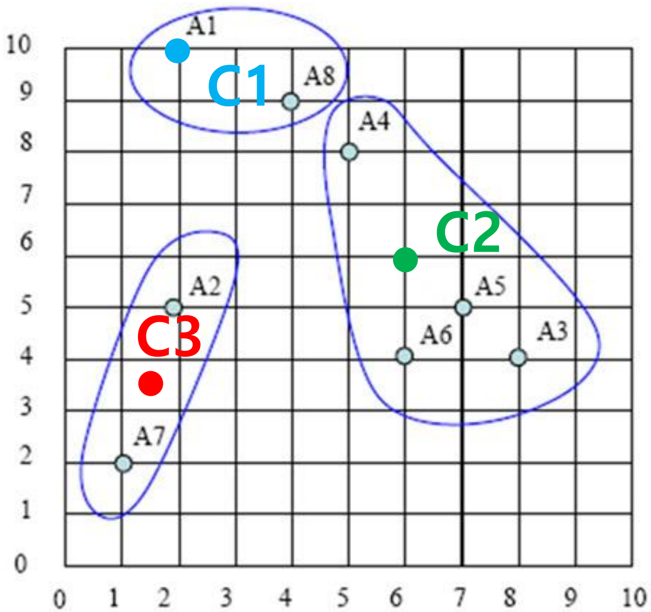
- Step 5: repeat Step 3 until the centroid don't change
- Step 3: Form *k* clusters by assigning all points to the closest centroid

	Point	C1(2, 10)	C2(6, 6)	C3(1.5, 3.5)	Cluster
A1	(2, 10)	0	5.65	6.51	C1
A2	(2, 5)	5	4.12	1.58	C3
A3	(8, 4)	8.48	2.82	6.51	C2
A4	(5, 8)	3.60	2.23	5.70	C2
A5	(7, 5)	7.07	1.41	5.70	C2
A6	(6, 4)	7.21	2	4.52	C2
A7	(1, 2)	8.06	6.40	1.58	C3
A8	(4, 9)	2.23	3.60	6.04	C1

Understanding clustering

❖ k-means example

- Step 5: repeat Step 3 until the centroid don't change
- Step 3: Form *k* clusters by assigning all points to the closest centroid



Understanding clustering

❖ k-means example

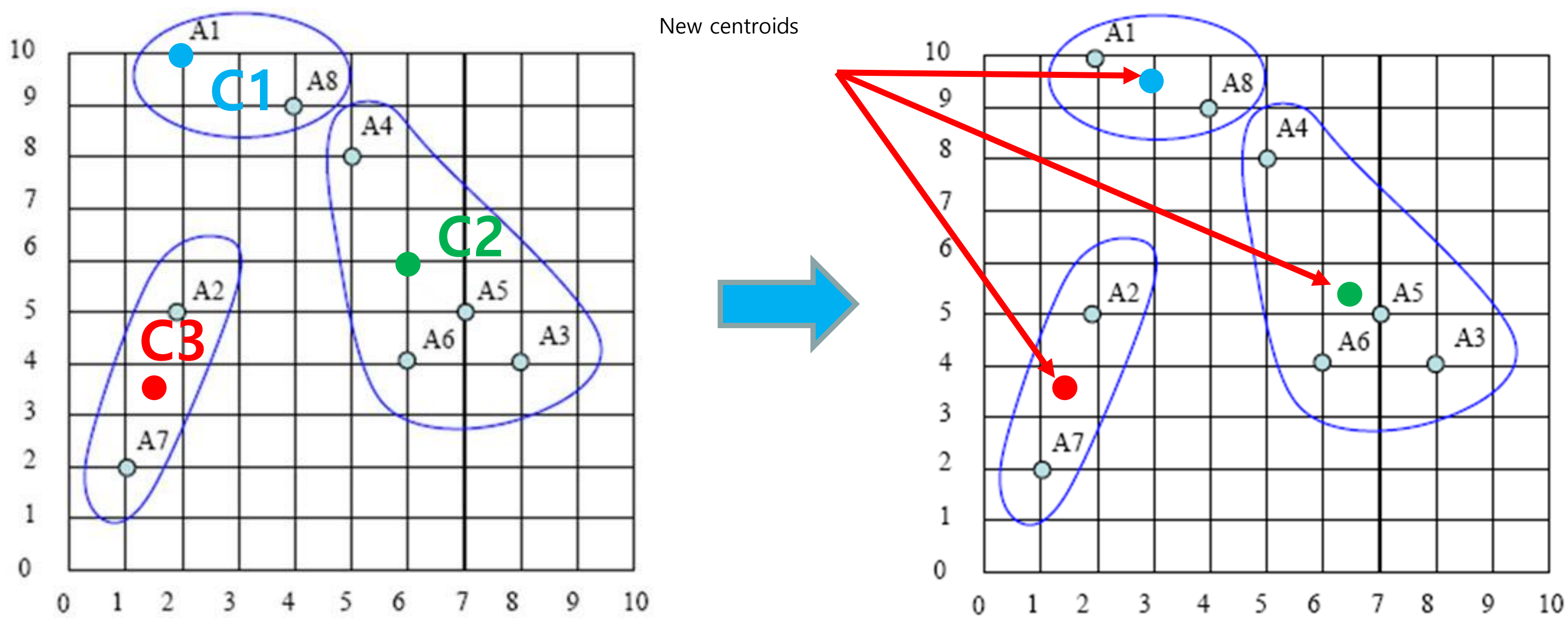
- Step 4: Recompute the centroid of each cluster (Calculate the mean of each cluster)

	Point	Cluster	Mean of C#
A1	(2, 10)	C1	(3, 9.5)
A8	(4, 9)	C1	
A3	(8, 4)	C2	(6.5, 5.25)
A4	(5, 8)	C2	
A5	(7, 5)	C2	
A6	(6, 4)	C2	
A7	(1, 2)	C3	(1.5, 3.5)
A2	(2, 5)	C3	

Understanding clustering

❖ k-means example

- Step 4: Recompute the centroid of each cluster (Calculate the mean of each cluster)



Understanding clustering

❖ k-means example

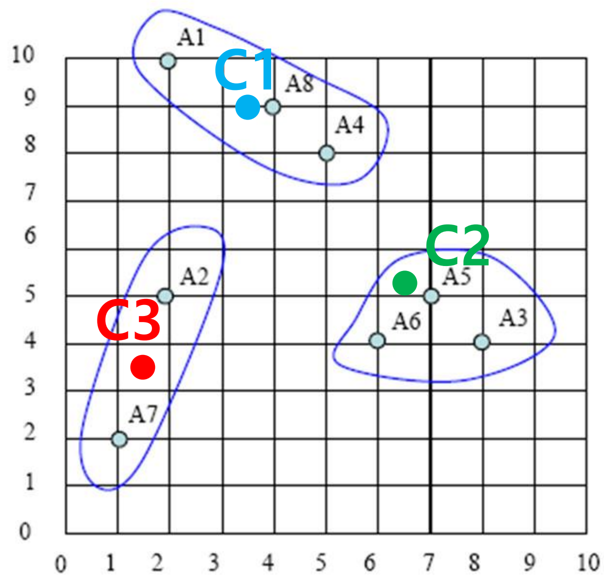
- Step 5: repeat Step 3 until the centroid don't change
- Step 3: Form *k* clusters by assigning all points to the closest centroid

	Point	C1(3, 9.5)	C2(6.5, 5.25)	C3(1.5, 3.5)	Cluster
A1	(2, 10)	1.11	6.54	6.51	C1
A2	(2, 5)	4.60	4.5	1.58	C3
A3	(8, 4)	7.43	1.95	6.51	C2
A4	(5, 8)	2.5	3.13	5.70	C1
A5	(7, 5)	6.02	0.55	5.70	C2
A6	(6, 4)	6.26	1.34	4.52	C2
A7	(1, 2)	7.76	6.38	1.58	C3
A8	(4, 9)	1.11	4.5	6.04	C1

Understanding clustering

❖ k-means example

- Step 5: repeat Step 3 until the centroid don't change
- Step 3: Form *k* clusters by assigning all points to the closest centroid



Understanding clustering

❖ k-means example

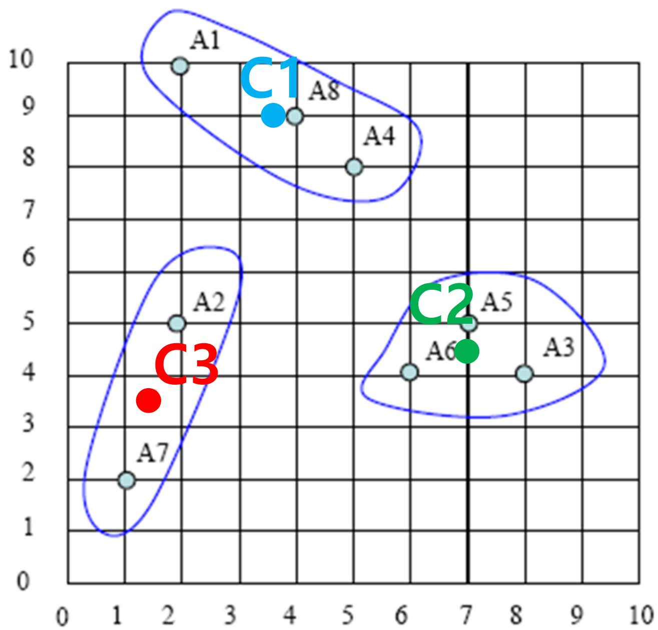
- Step 4: Recompute the centroid of each cluster (Calculate the mean of each cluster)

	Point	Cluster	Mean of C#
A1	(2, 10)	C1	(3.66, 9)
A8	(4, 9)	C1	
A4	(5, 8)	C1	
A3	(8, 4)	C2	(7, 4.33)
A5	(7, 5)	C2	
A6	(6, 4)	C2	
A7	(1, 2)	C3	(1.5, 3.5)
A2	(2, 5)	C3	

Understanding clustering

❖ k-means example

- Final cluster result



Clustering Techniques

❖ K-Means in Python

```
from sklearn.cluster import KMeans
import numpy as np

X = np.array([[2, 10], [2, 5], [8, 4],[5, 8], [7, 5], [6, 4], [1, 2], [4, 9]])

kmeans = KMeans(n_clusters=3).fit(X)

print("Labels: ", kmeans.labels_)

print("Cluster Centers: ", kmeans.cluster_centers_)

print("Predict Values: ", kmeans.predict([[1, 1]]))
```

```
Labels:            [2 1 0 2 0 0 1 2]
Cluster Centers:   [[7.         4.33333333]
                    [1.5        3.5       ]
                    [3.66666667 9.        ]]
Predict Values:    [1]
```

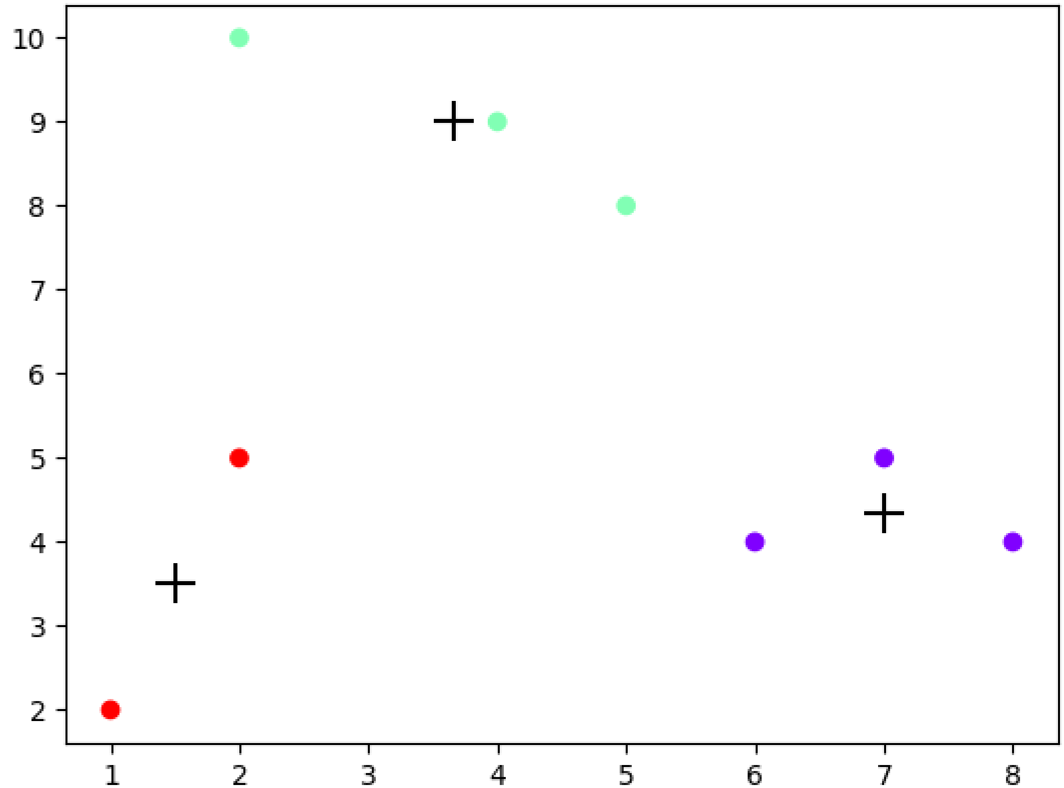
Clustering Techniques

❖ K-Means visualization in Python

```
from sklearn.cluster import KMeans
import numpy as np
import matplotlib.pyplot as plt

X = np.array([[2, 10], [2, 5], [8, 4],[5, 8], [7, 5], [6, 4], [1, 2], [4, 9]])

kmeans = KMeans(n_clusters=3).fit(X)

plt.scatter(X[:,0], X[:,1], c=kmeans.labels_, cmap='rainbow')

plt.scatter(kmeans.cluster_centers_[:,0] ,kmeans.cluster_centers_[:,1], color='black',
                    marker="+", s=200)

plt.show()
```

Clustering Techniques

❖ K-Means visualization in Python



Clustering Techniques

❖ K-Means visualization in Python

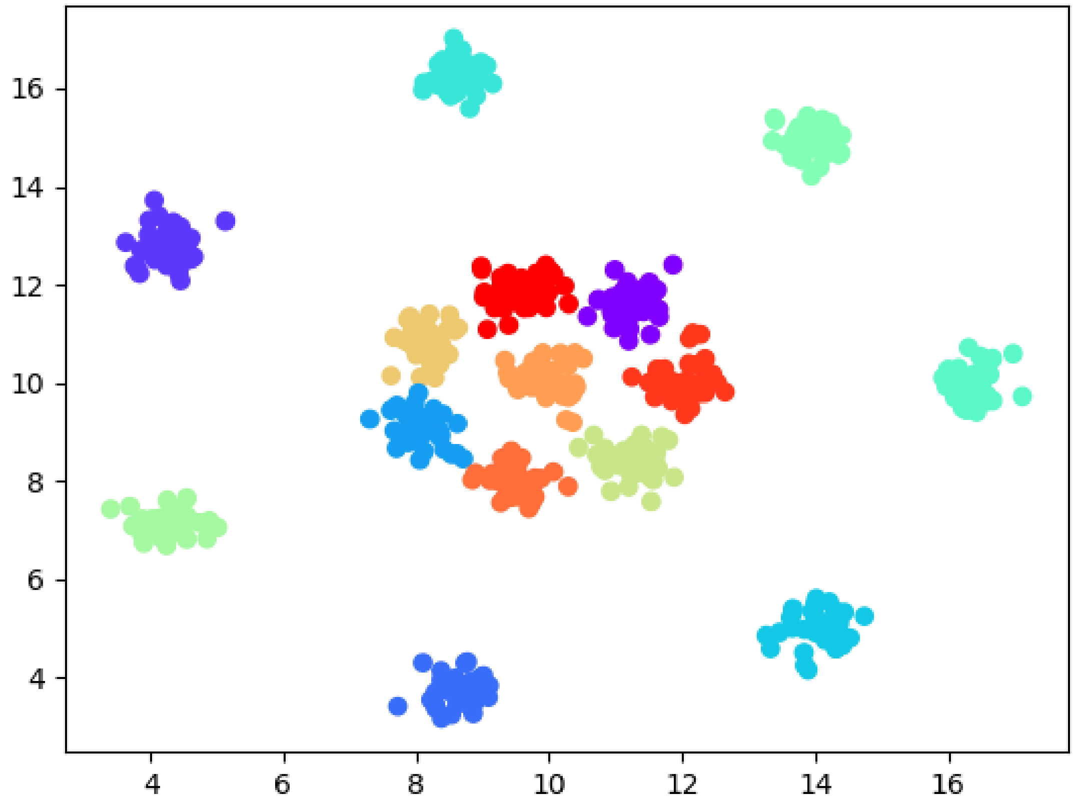
- r15 dataset (r15.csv) -> n_clusters=15

```
from sklearn.cluster import KMeans
import pandas as pd
import matplotlib.pyplot as plt

sample_df = pd.read_csv("D:/r15.csv")

training_points = sample_df[["col1", "col2"]]
training_labels = sample_df["target"]

kmeans = KMeans(n_clusters=15).fit(training_points)

plt.scatter(training_points["col1"], training_points["col2"], c=kmeans.labels_,
                    cmap='rainbow')

plt.show()
```

Clustering Techniques

❖ K-Means visualization in Python

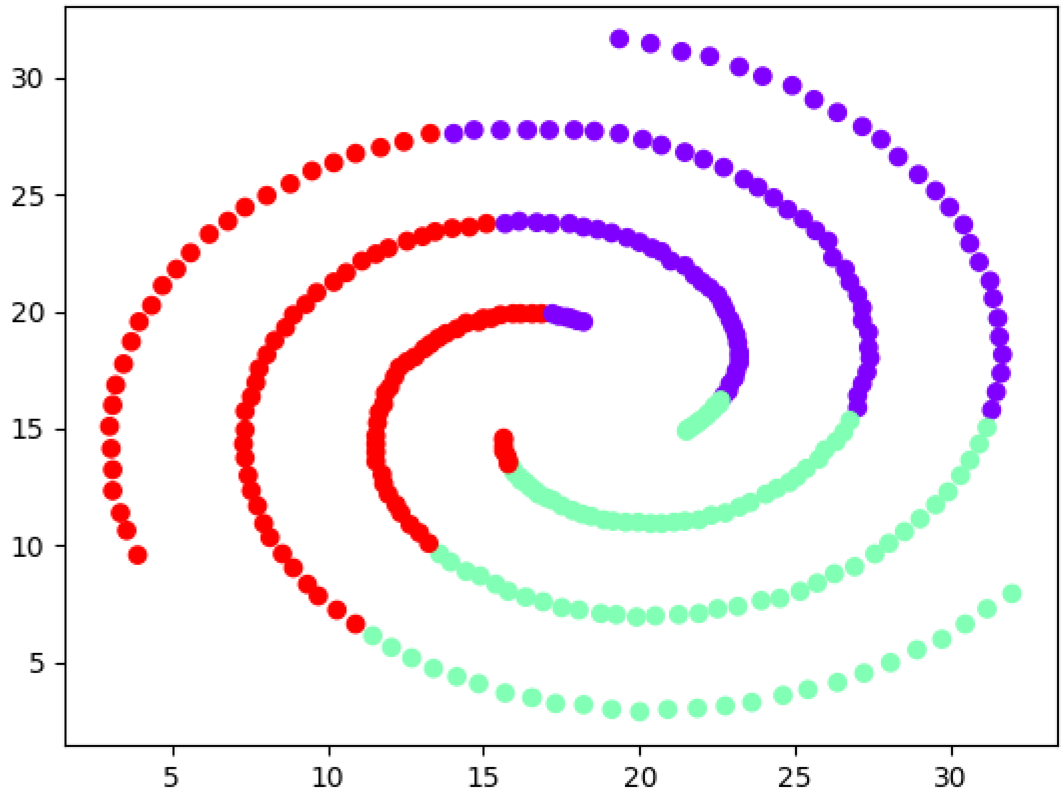
- r15 dataset (r15.csv) -> n_clusters=15



Clustering Techniques

❖ K-Means visualization in Python

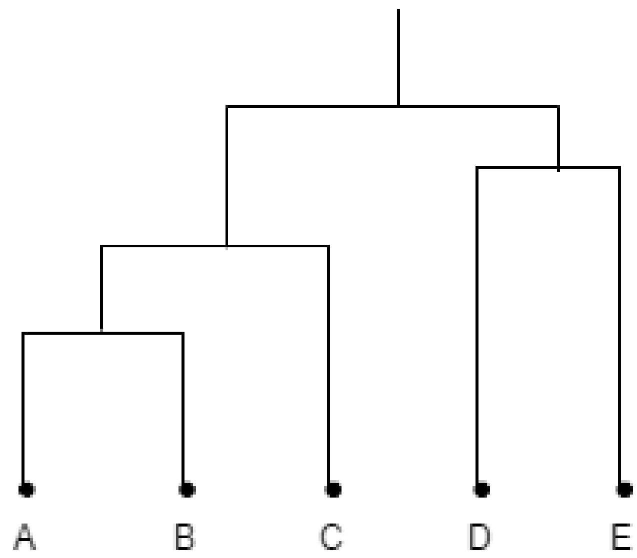
- spiral dataset (spiral.csv) -> n_clusters=3



Clustering Techniques

❖ Agglomerative clustering

- Grouping data objects into a hierarchy or “tree” of clusters
- Dendrogram is used to represent the process of hierarchical clustering



Clustering Techniques

❖ Agglomerative clustering

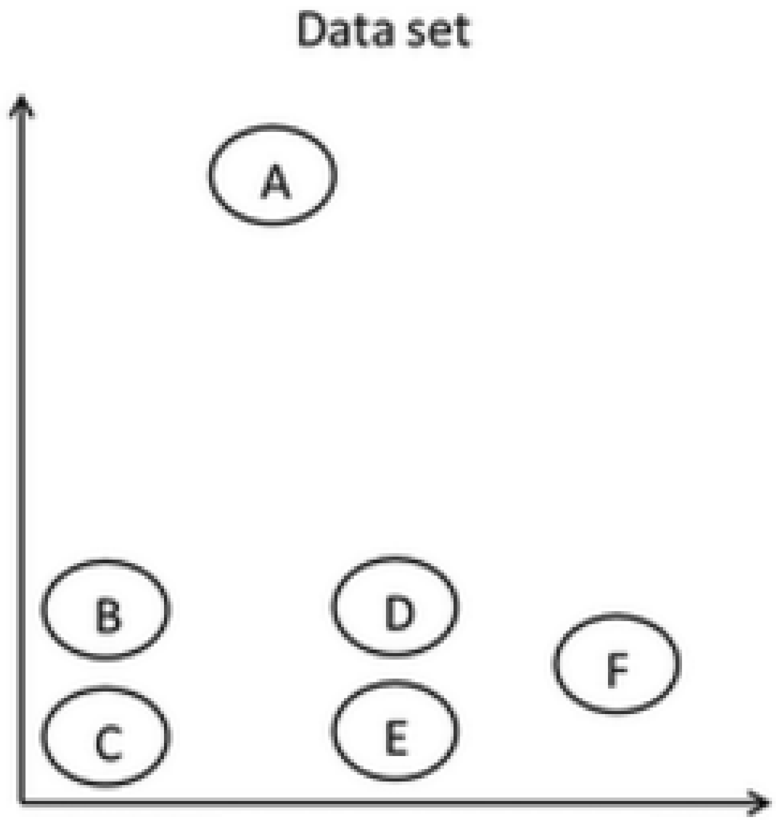
- Bottom-up strategy
- Starts by letting each object form its own cluster
- Iteratively merges clusters into larger and larger clusters
  - Until all the objects are in a single cluster

❖ Agglomerative hierarchical clustering : Procedure

1. Each object forms one cluster
2. Merges the two closest (similar) clusters at the lowest level int o one cluster
3. Repeat step 2 until it becomes a single cluster

Clustering Techniques

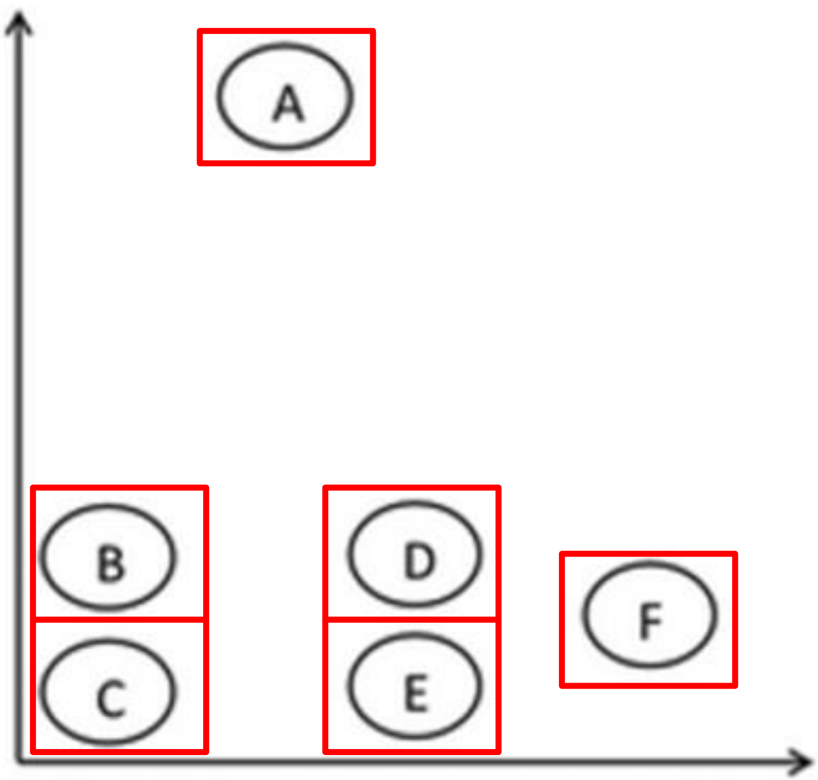
❖ Agglomerative clustering example



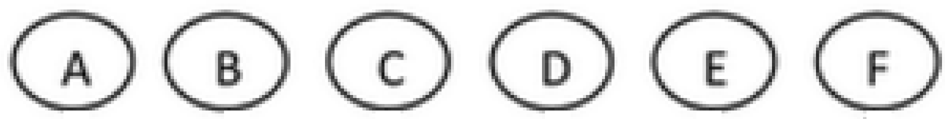
Clustering Techniques

❖ Agglomerative clustering example

Data set



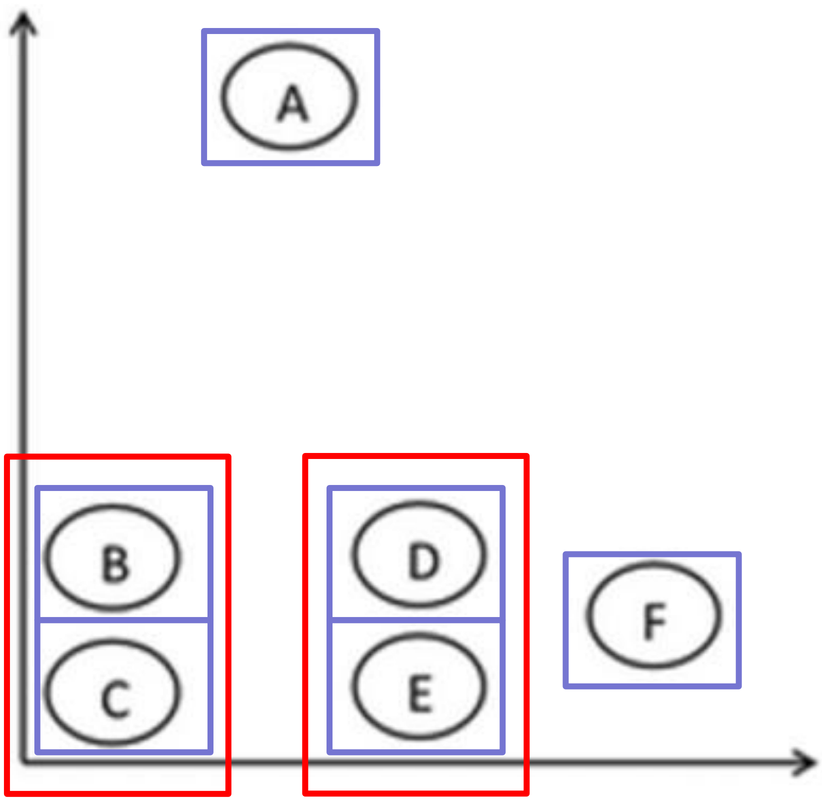
Agglomerative clustering of data (tree)



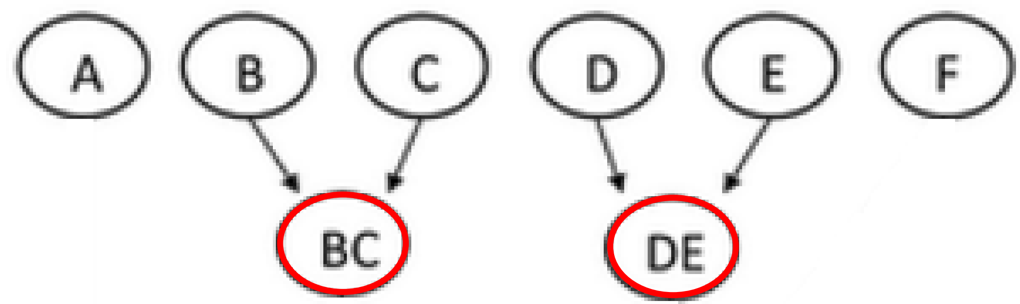
Clustering Techniques

❖ Agglomerative clustering example

Data set



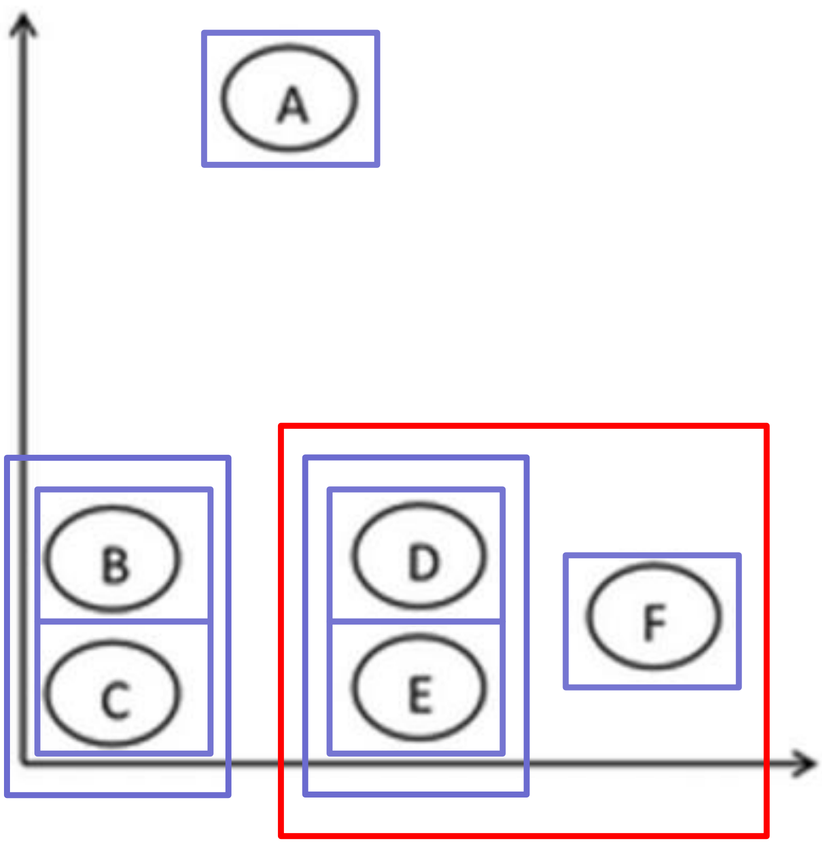
Agglomerative clustering of data (tree)



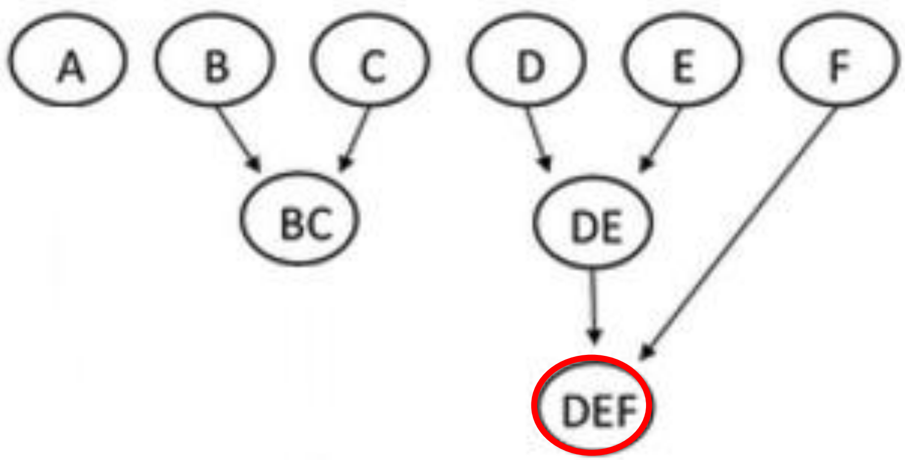
Clustering Techniques

❖ Agglomerative clustering example

Data set



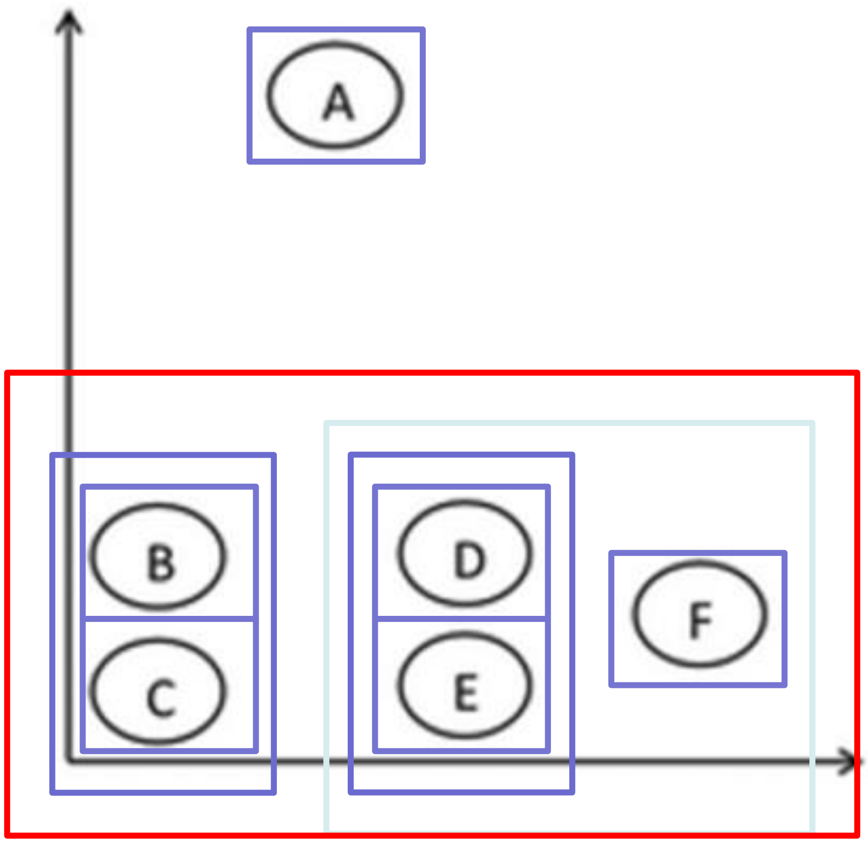
Agglomerative clustering of data (tree)



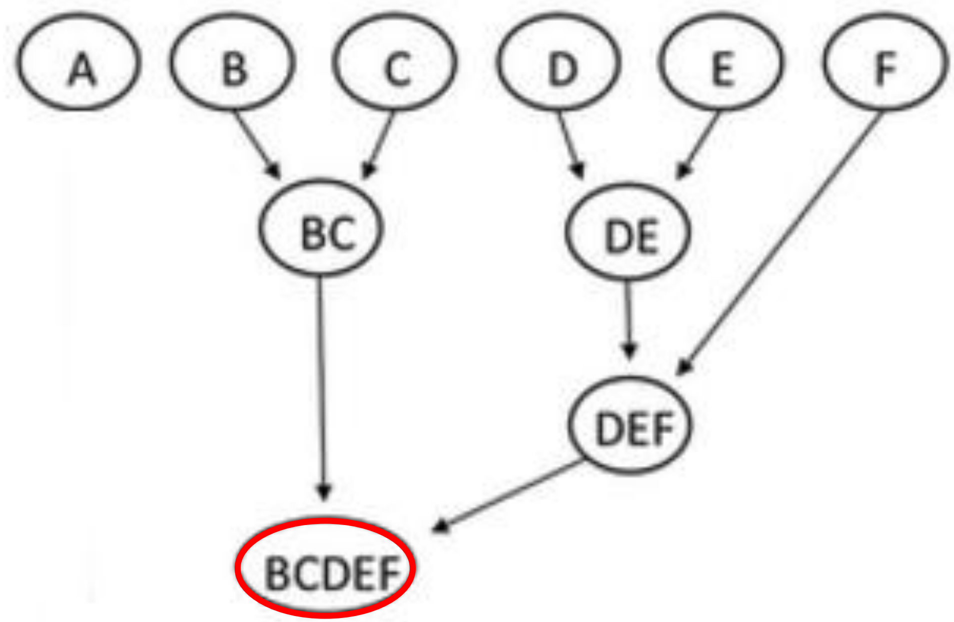
Clustering Techniques

❖ Agglomerative clustering example

Data set



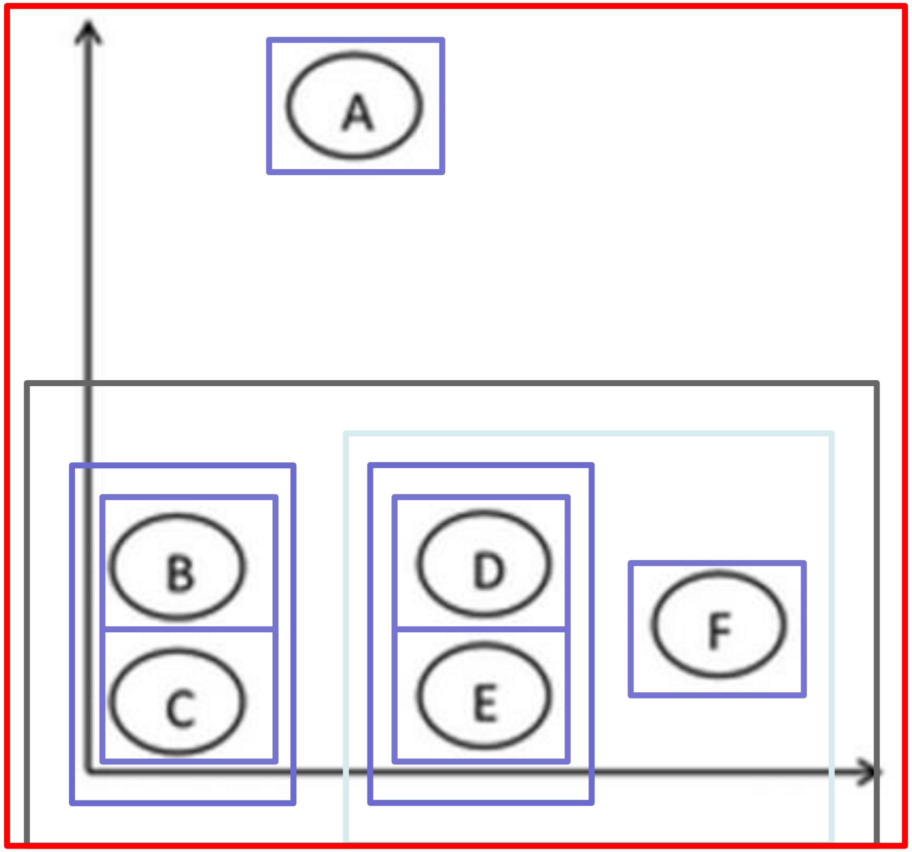
Agglomerative clustering of data (tree)



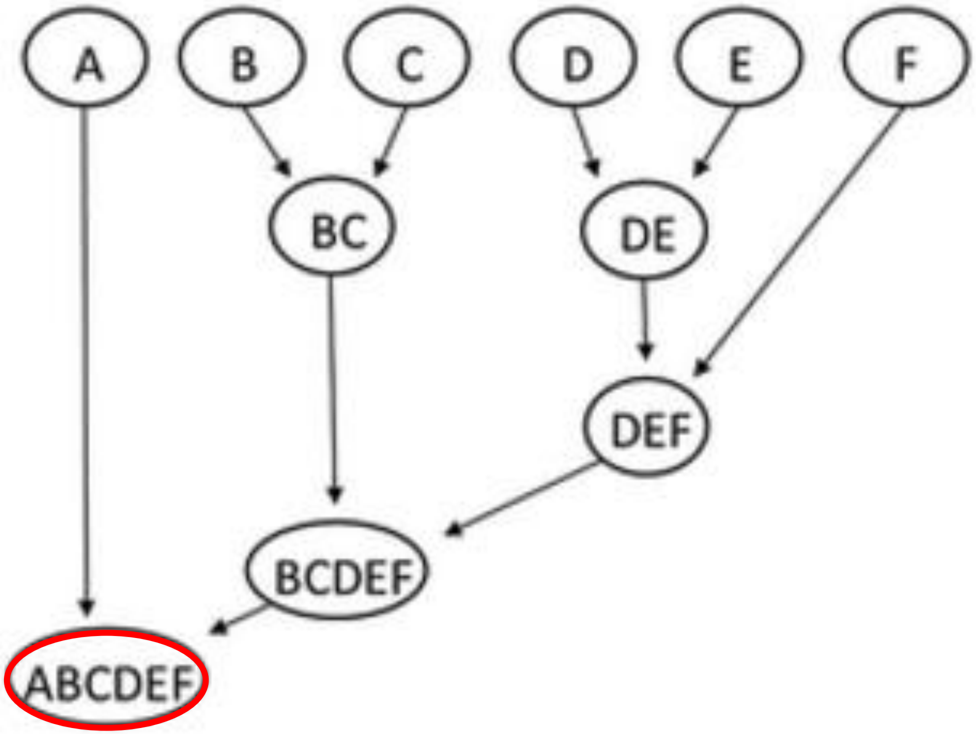
Clustering Techniques

❖ Agglomerative clustering example

Data set



Agglomerative clustering of data (tree)



Clustering Techniques

❖ Agglomerative clustering in Python

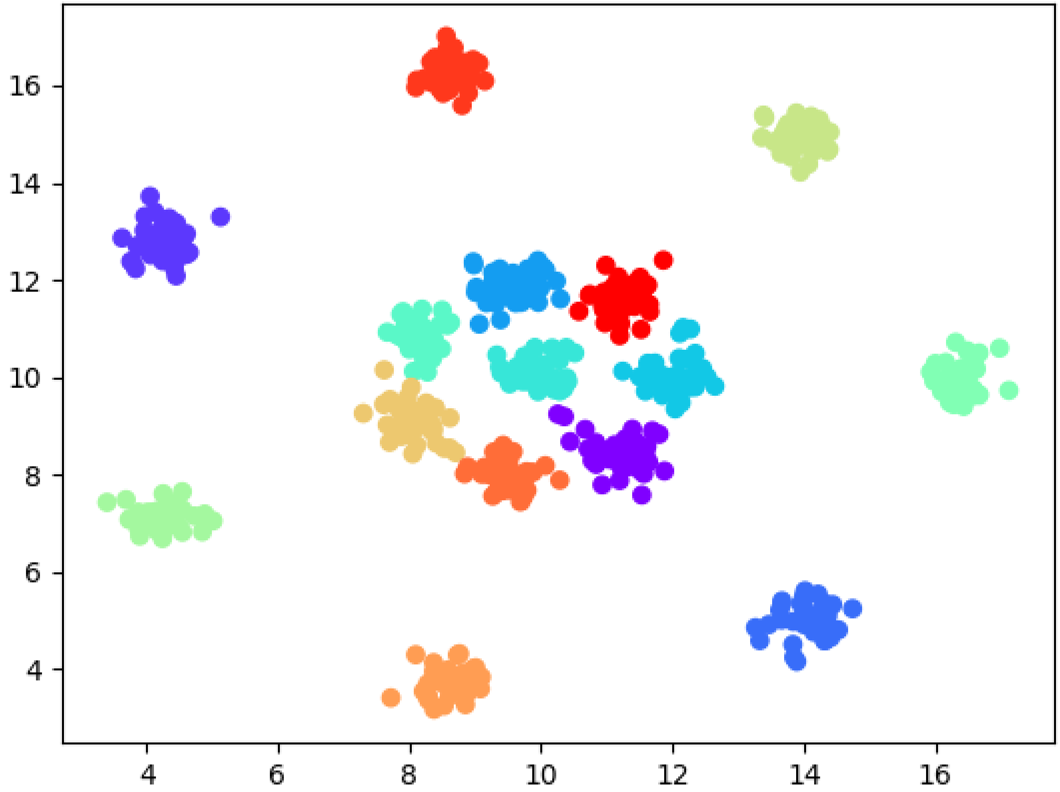
- r15 dataset (r15.csv) -> n_clusters=15

```
from sklearn.cluster import AgglomerativeClustering
import pandas as pd
import matplotlib.pyplot as plt

sample_df = pd.read_csv("D:/r15.csv")

training_points = sample_df[["col1", "col2"]]
training_labels = sample_df["target"]

agglo = AgglomerativeClustering(n_clusters=15).fit(training_points)

plt.scatter(training_points["col1"], training_points["col2"], c=agglo.labels_,
                    cmap='rainbow')

plt.show()
```

Clustering Techniques

❖ Agglomerative clustering in Python

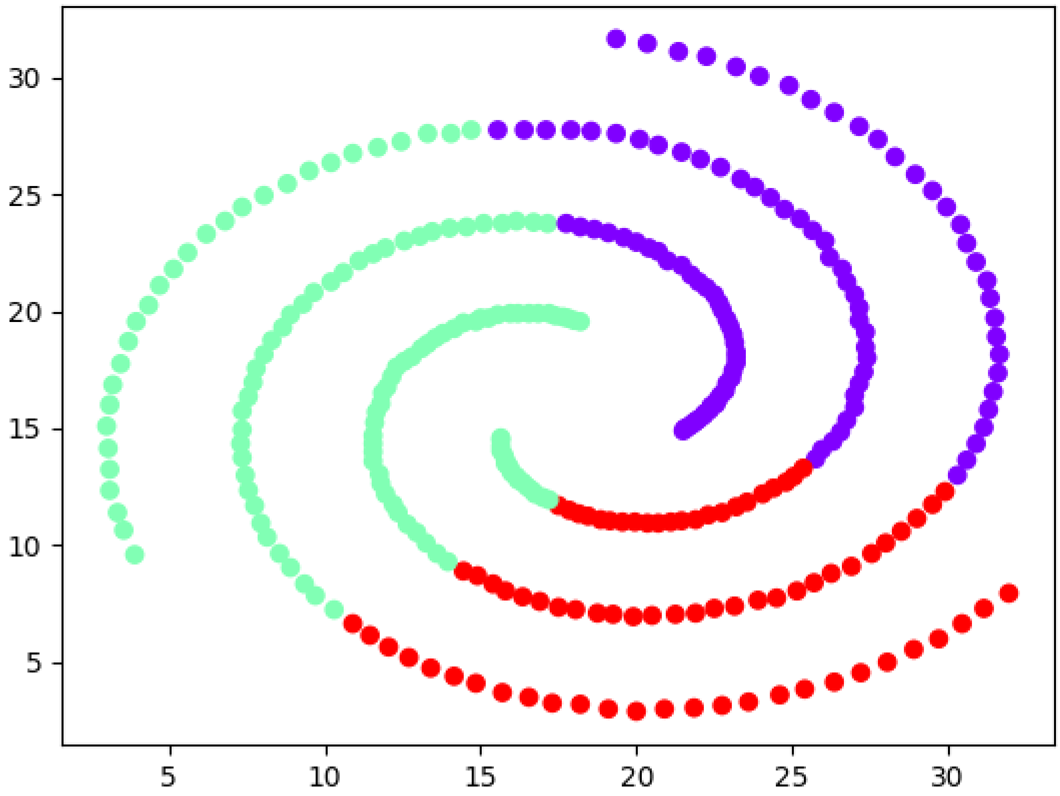
- r15 dataset (r15.csv) -> n_clusters=15



Clustering Techniques

❖ Agglomerative clustering in Python

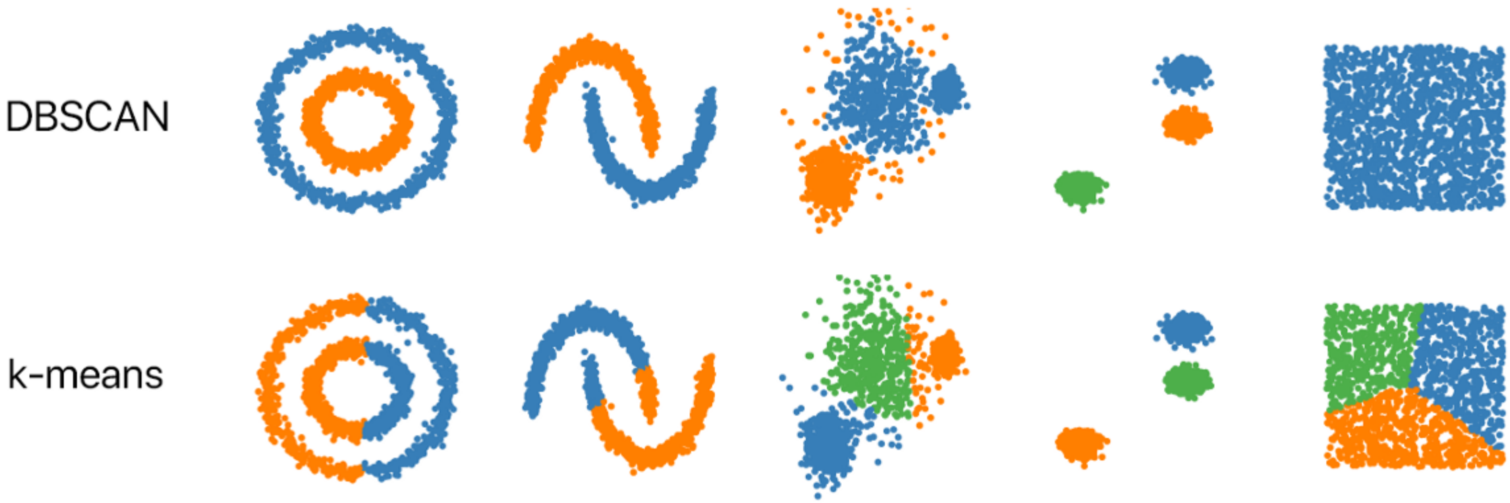
- spiral dataset (spiral.csv) -> n_clusters=3



Clustering Techniques

❖ **What is DBSCAN?**

- Clustering based on density-connected points
- Continues growing a given cluster as long as the density in the "neighborhood" exceeds some threshold

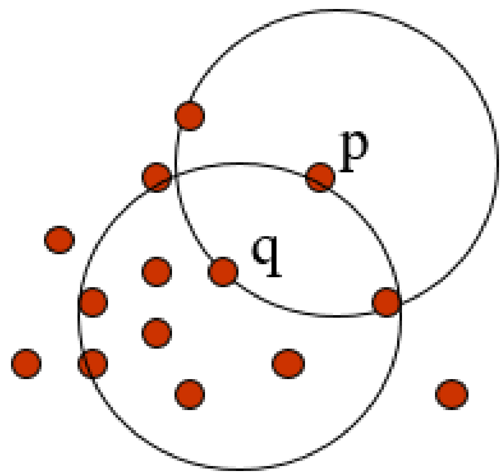


Clustering Techniques

❖ DBSCAN parameters

- Epsilon (ε)
  - Maximum radius of the neighborhood

- minPts
  - Minimum number of points in an Eps-neighborhood of that point



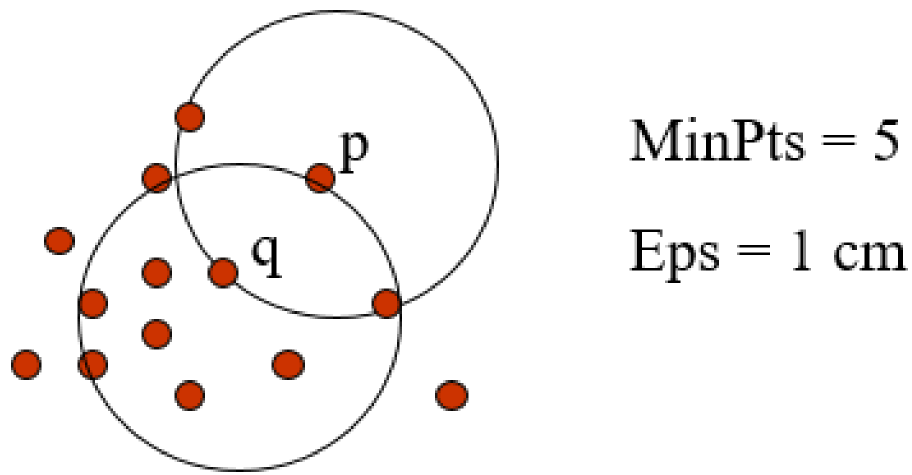
MinPts = 5

Eps = 1 cm

Clustering Techniques

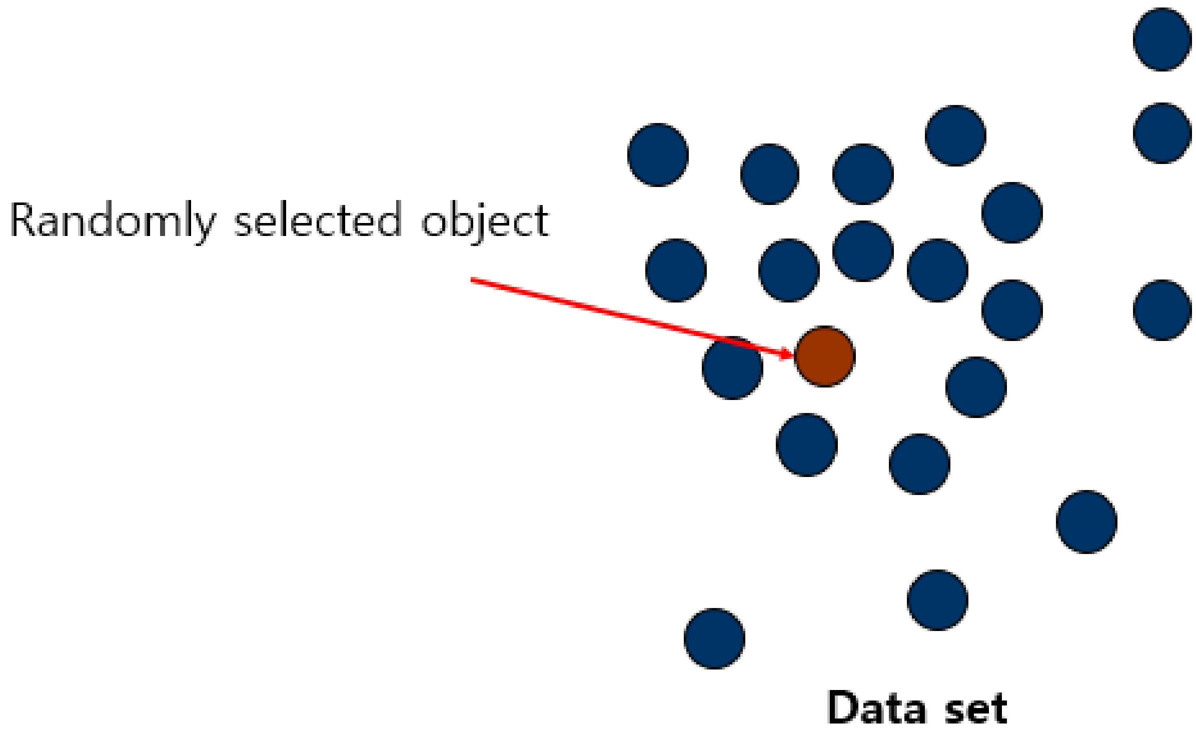
❖ DBSCAN parameters

- A Core object is an object that meets the minimum number of points in the ε-neighborhood
- The object q is a core object



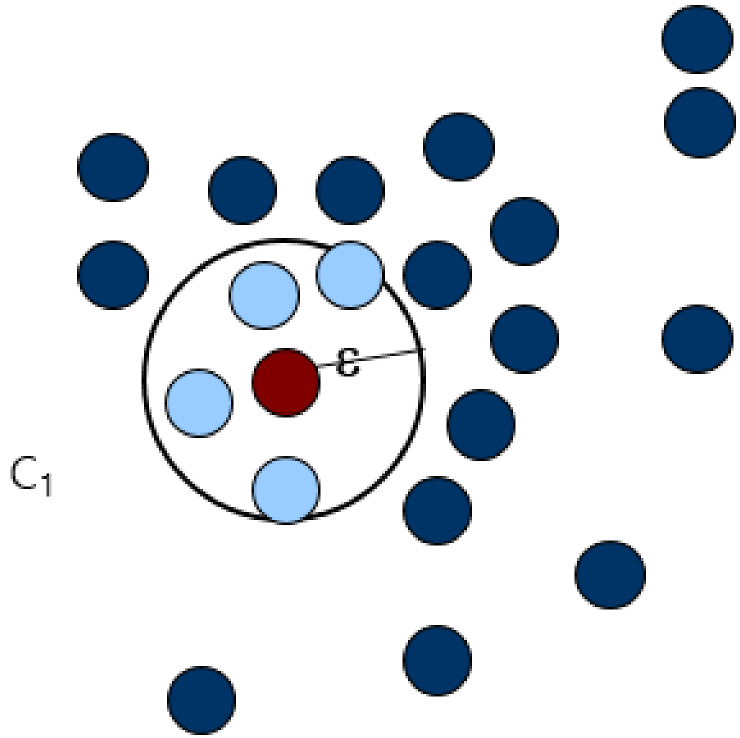
Clustering Techniques

❖ **DBSCAN example**



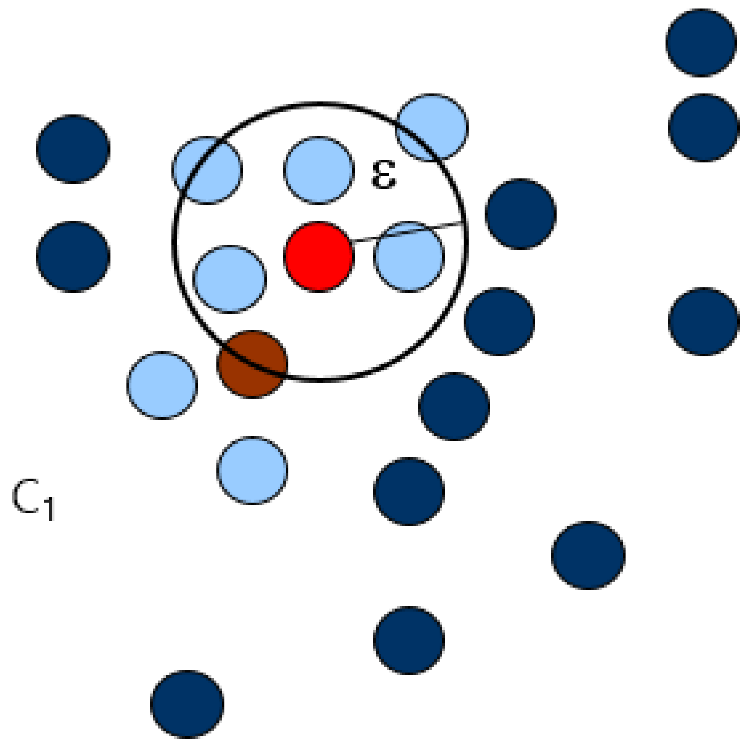
Clustering Techniques

❖ **DBSCAN example**



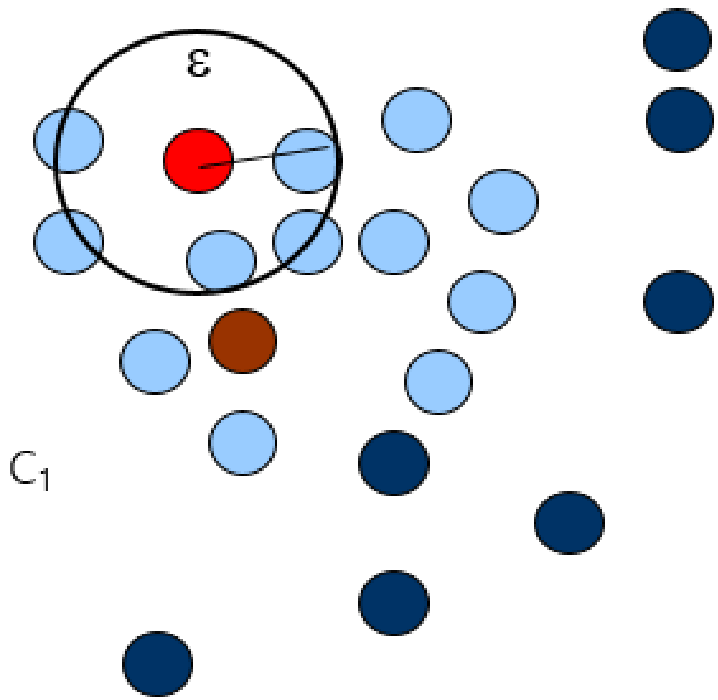
Clustering Techniques

❖ DBSCAN example



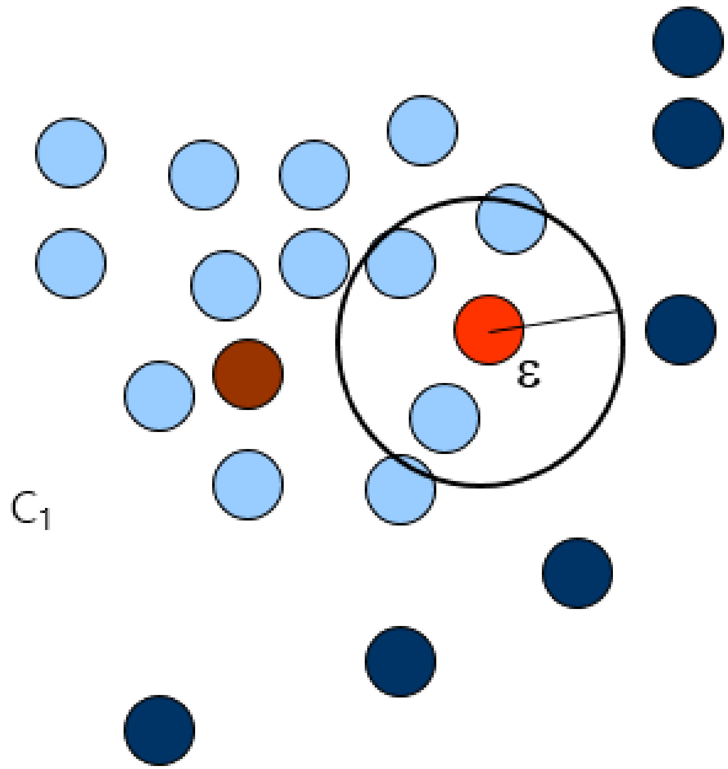
Clustering Techniques

❖ **DBSCAN example**



Clustering Techniques

❖ DBSCAN example



Clustering Techniques

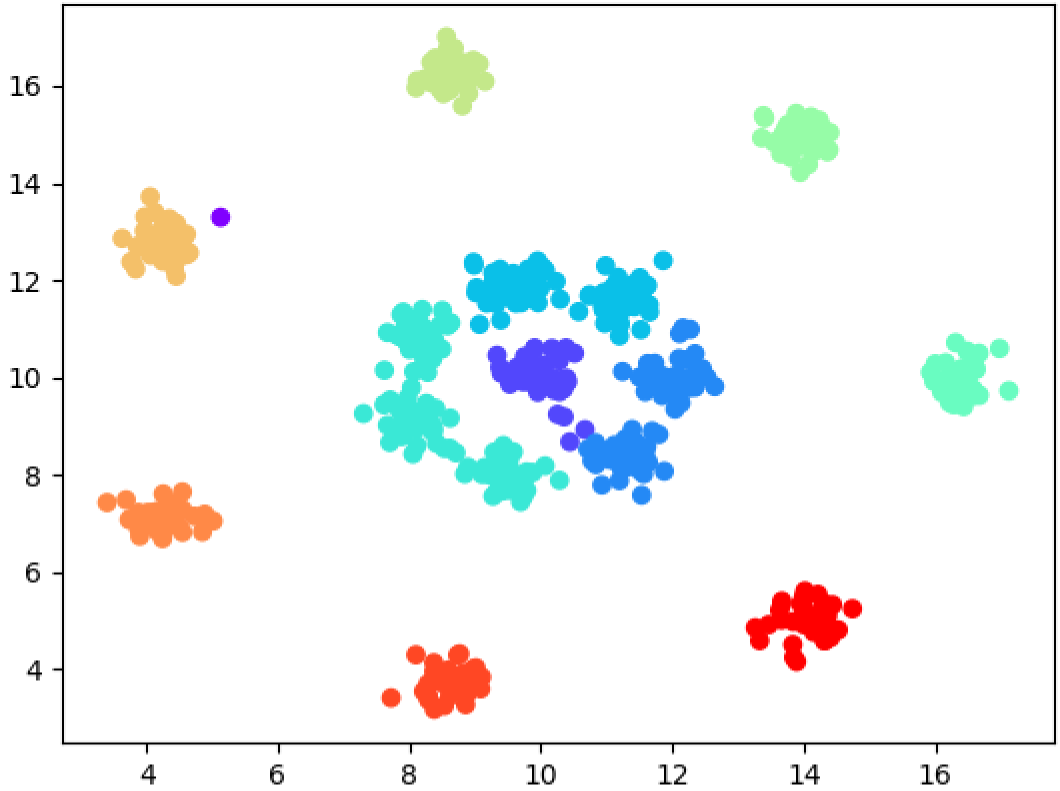
❖ DBSCAN in Python

- r15 dataset (r15.csv)

```
from sklearn.cluster import DBSCAN
import pandas as pd
import matplotlib.pyplot as plt

sample_df = pd.read_csv("D:/r15.csv")

training_points = sample_df[["col1", "col2"]]
training_labels = sample_df["target"]

dbscan = DBSCAN(eps=0.6, min_samples=10).fit(training_points)

plt.scatter(training_points["col1"], training_points["col2"], c=dbscan.labels_,
            cmap='rainbow')

plt.show()
```

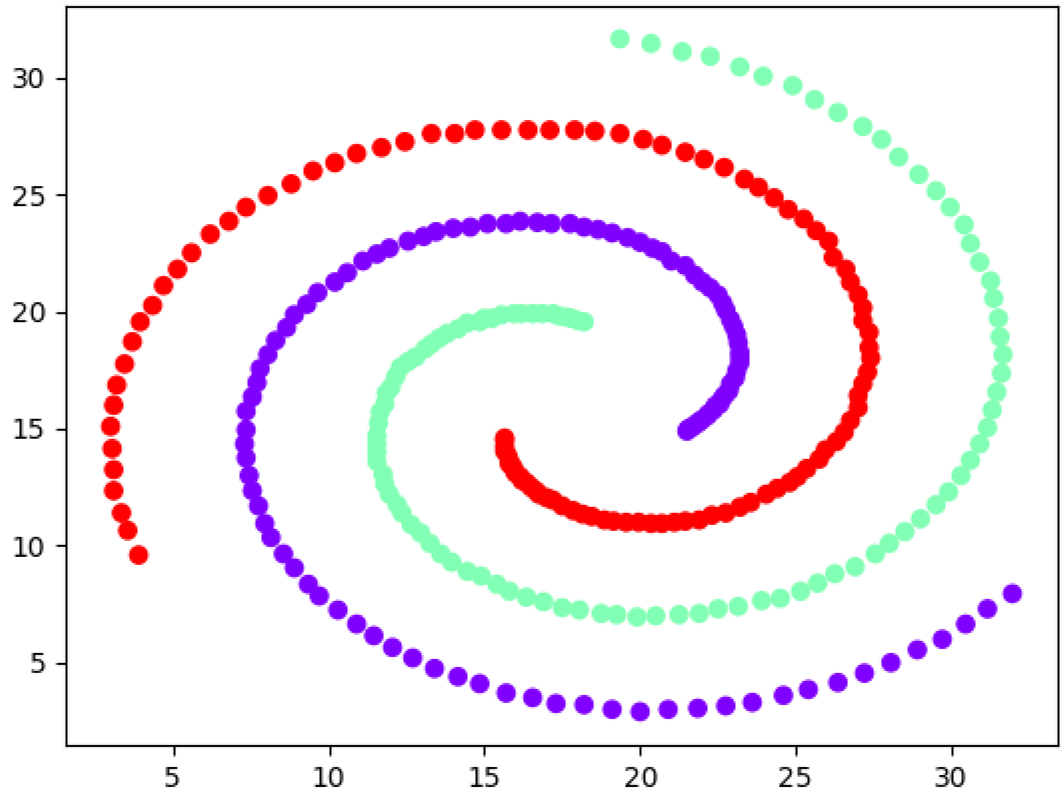
Clustering Techniques

❖ DBSCAN in Python



Clustering Techniques

❖ DBSCAN in Python



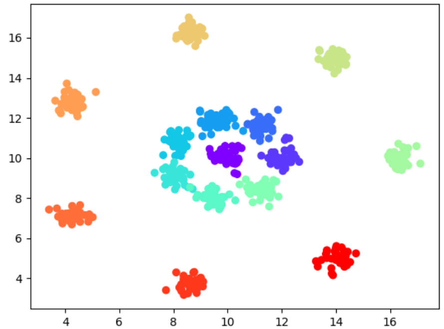
C

Evaluation of Clustering

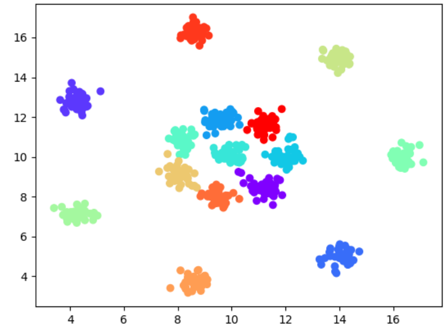
Evaluation of Clustering

❖ Why evaluate?

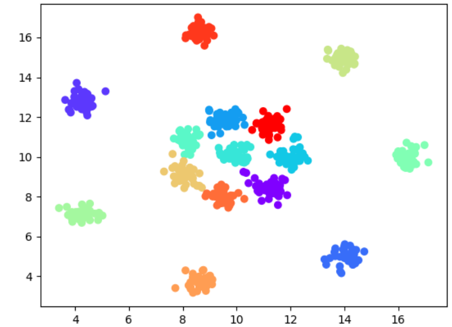
- Comparison on r15



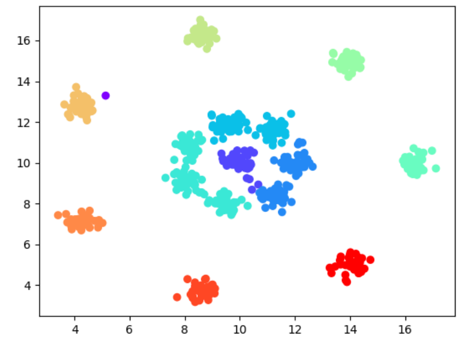
<Ground Truth>



<K-Means>

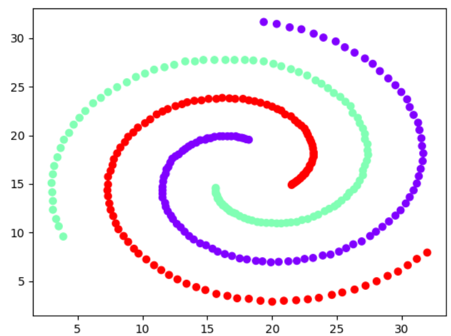


<Agglomerative>

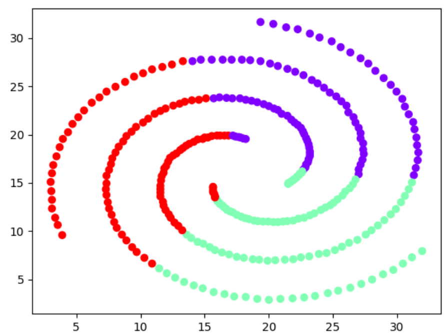


<DBSCAN>

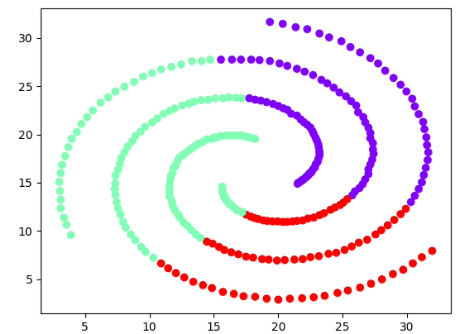
- Comparison on spiral



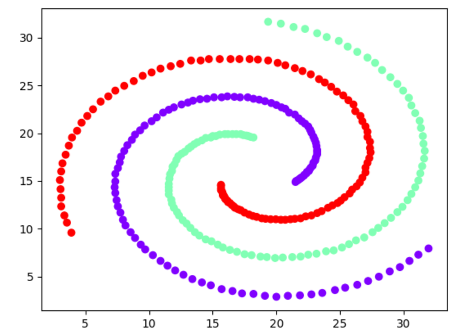
<Ground Truth>



<K-Means>



<Agglomerative>



<DBSCAN>

Evaluation of Clustering

❖ Adjusted Rand Index

- Computes a similarity measure between two clustering

❖ Calculated used the following formula:

$$Adjusted\ RI = (RI - Expected\_RI) \,/\, (max\,(RI) - Expected\_RI)$$

❖ It has two parameters

- labels_true
  - Ground truth class labels
- labels_pred
  - Clusters label to evaluate

Clustering Techniques

❖ Adjusted Rand Index for K-Means

- r15 dataset (r15.csv)

```
from sklearn.cluster import KMeans
import pandas as pd
from sklearn.metrics.cluster import adjusted_rand_score

sample_df = pd.read_csv("D:/r15.csv")

training_points = sample_df[["col1", "col2"]]
training_labels = sample_df["target"]

kmeans = KMeans(n_clusters=15).fit(training_points)

arc = adjusted_rand_score(training_labels, kmeans.labels_)

print(arc)
```

```
0.9927781994136302
```

Clustering Techniques

❖ Adjusted Rand Index for DBSCAN

- spiral dataset (spiral.csv)

```
from sklearn.cluster import DBSCAN
import pandas as pd
from sklearn.metrics.cluster import adjusted_rand_score

sample_df = pd.read_csv("D:/spiral.csv")

training_points = sample_df[["col1", "col2"]]
training_labels = sample_df["target"]

dbscan = DBSCAN(eps=3, min_samples=2).fit(training_points)

arc = adjusted_rand_score(training_labels, dbscan.labels_)

print(arc)
```

```
1.0
```

D

Use Case

Useful

- Customer Segmentation
  - https://www.kaggle.com/karnikakapoor/customer-segmentation-clustering
- SaaS Data Analysis
  - https://medium.com/@sygong/k-means-clustering-for-customer-segmentations-a-practical-real-world-example-196a10323b9f
- Anime recommendation based on user clustering
  - https://www.kaggle.com/karnikakapoor/customer-segmentation-clustering
- Segmenting and Clustering Airbnb Listings in Zurich, Switzerland
  - https://www.linkedin.com/pulse/segmenting-clustering-airbnb-listings-zurich-georgios-chatzis/

Final Task

- Submit your source code for the following task:
  1. Try all source code in the lecture

- Submission: source code, result screenshots and result explanation

감사합니다!