Introduction to Data Science

Dr. Irfan Yousuf

Department of Computer Science (New Campus)

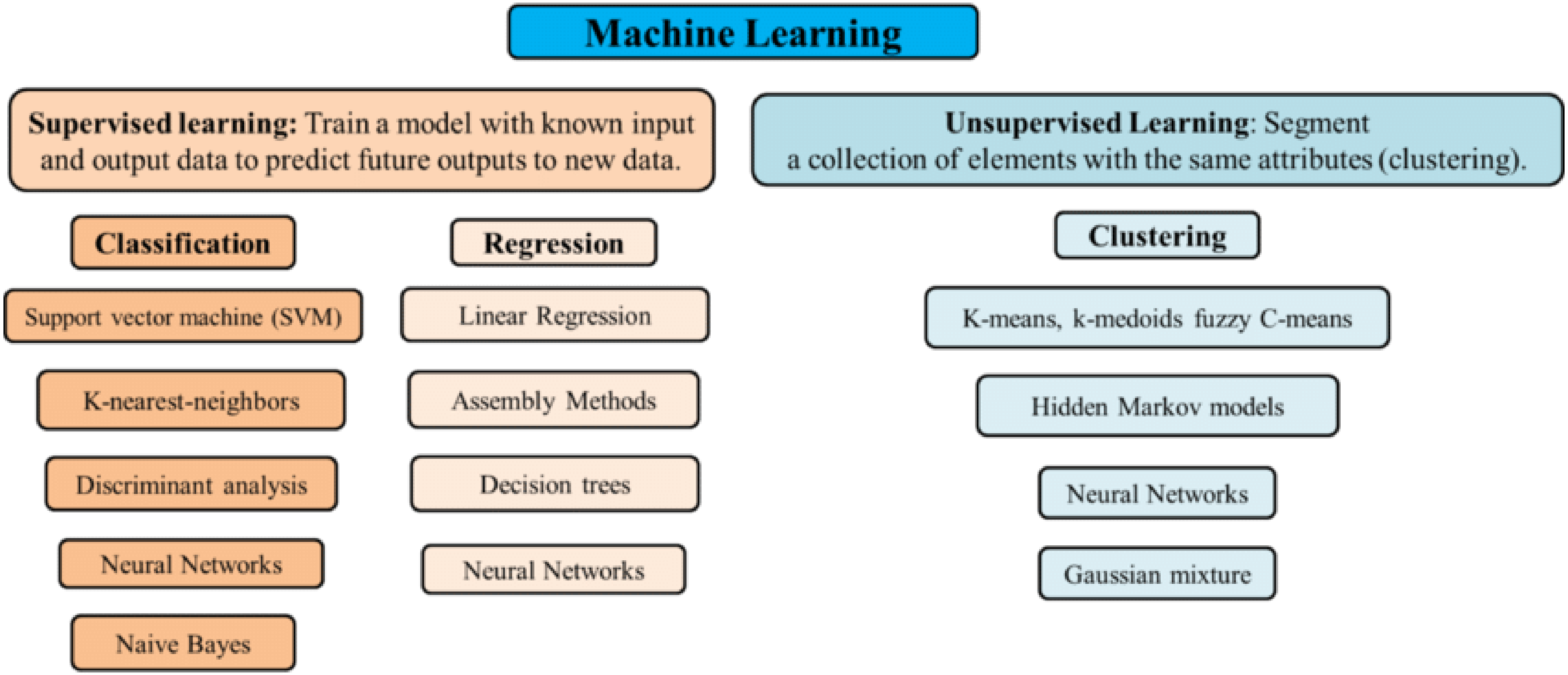
UET, Lahore

(Week # 10; March 18 - 22, 2024)

Outline

- K Nearest Neighbors
- K-means

Machine Learning Algorithms



Machine Learning

Supervised learning: Train a model with known input and output data to predict future outputs to new data.

Unsupervised Learning: Segment a collection of elements with the same attributes (clustering).

Classification

- Support vector machine (SVM)
- K-nearest-neighbors
- Discriminant analysis
- Neural Networks
- Naive Bayes

Regression

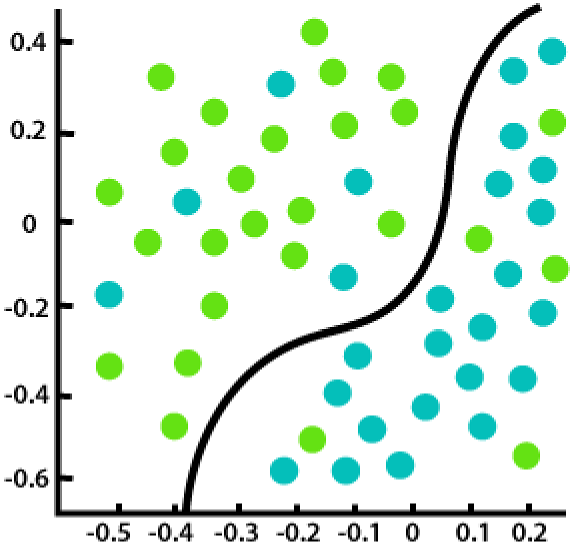
- Linear Regression
- Assembly Methods
- Decision trees
- Neural Networks

Clustering

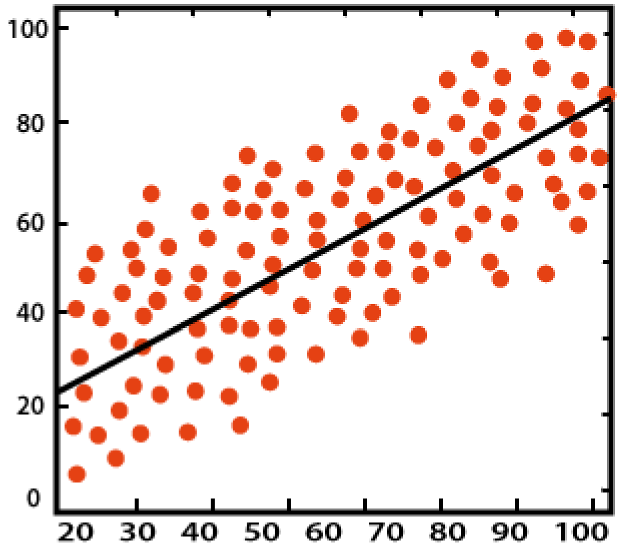
- K-means, k-medoids fuzzy C-means
- Hidden Markov models
- Neural Networks
- Gaussian mixture

Supervised Machine Learning Algorithms

- **Classification:** A classification problem is when the output variable is a **category**, such as “red” or “blue” or “disease” and “no disease”.
- Regression: A regression problem is when the output variable is **numeric**, such as “age” or “weight”.

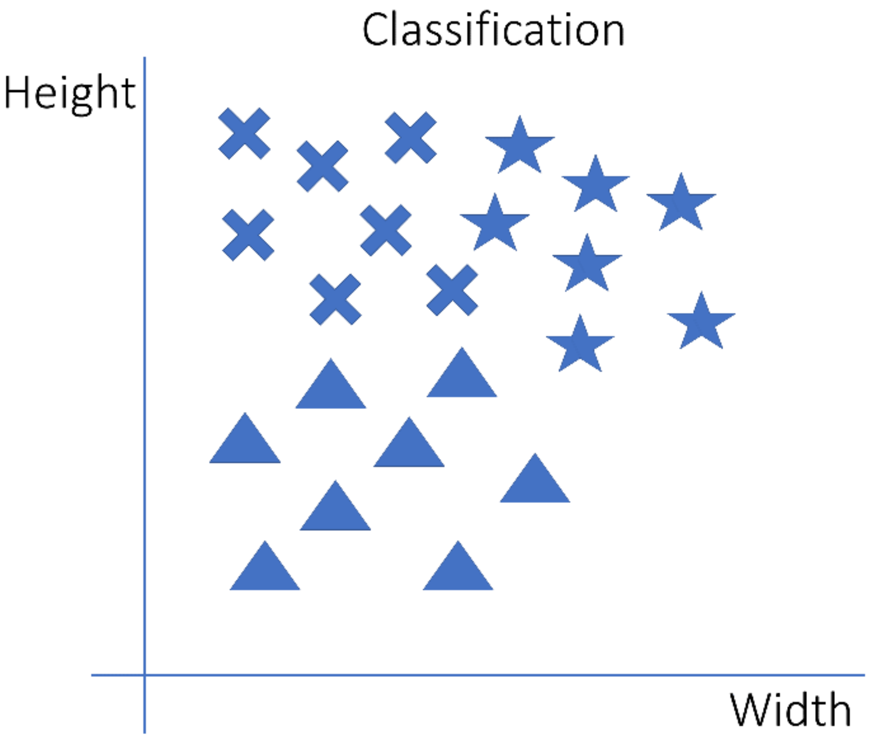


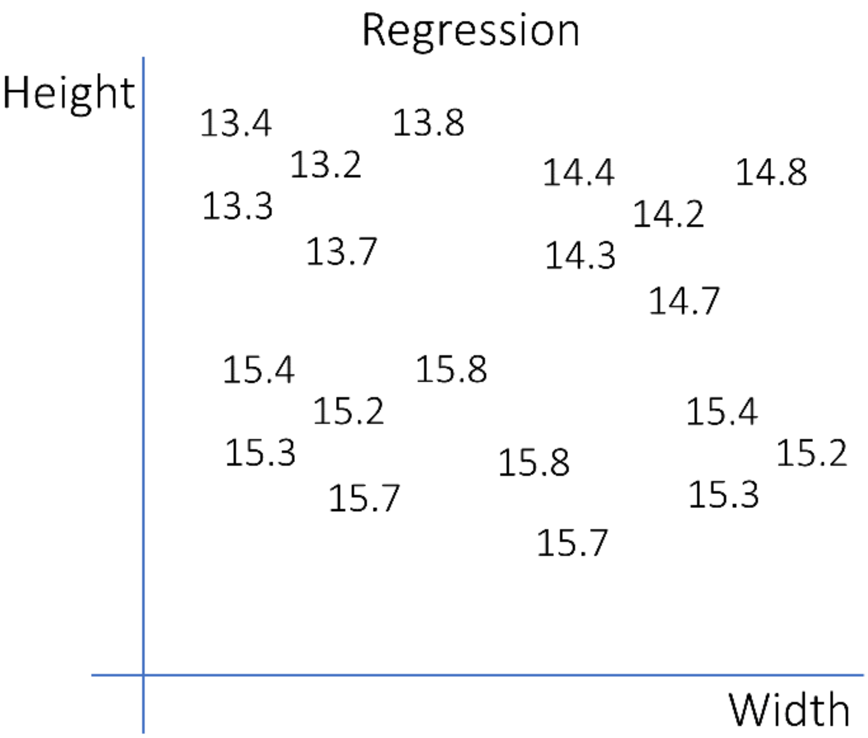
Classification



Regression

Supervised Machine Learning Algorithms


Classification


Regression

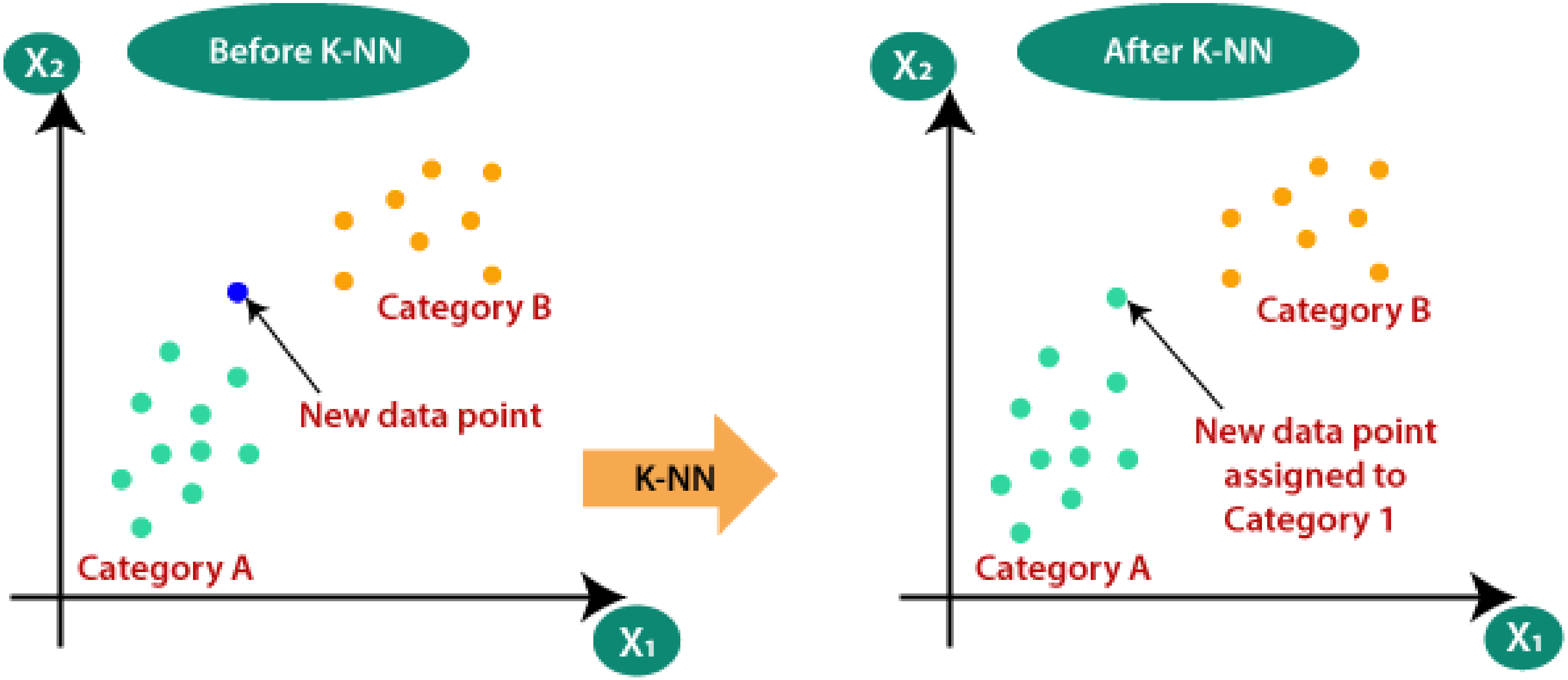
k Nearest Neighbors

- K-Nearest Neighbors (kNN) is one of the simplest Machine Learning algorithms based on Supervised Machine Learning technique.
- A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function.
- The algorithm assumes the similarity between the new case/data and available cases and put the new case into the most suitable category.
- The algorithm stores all the available data and classifies a new data point based on the similarity.

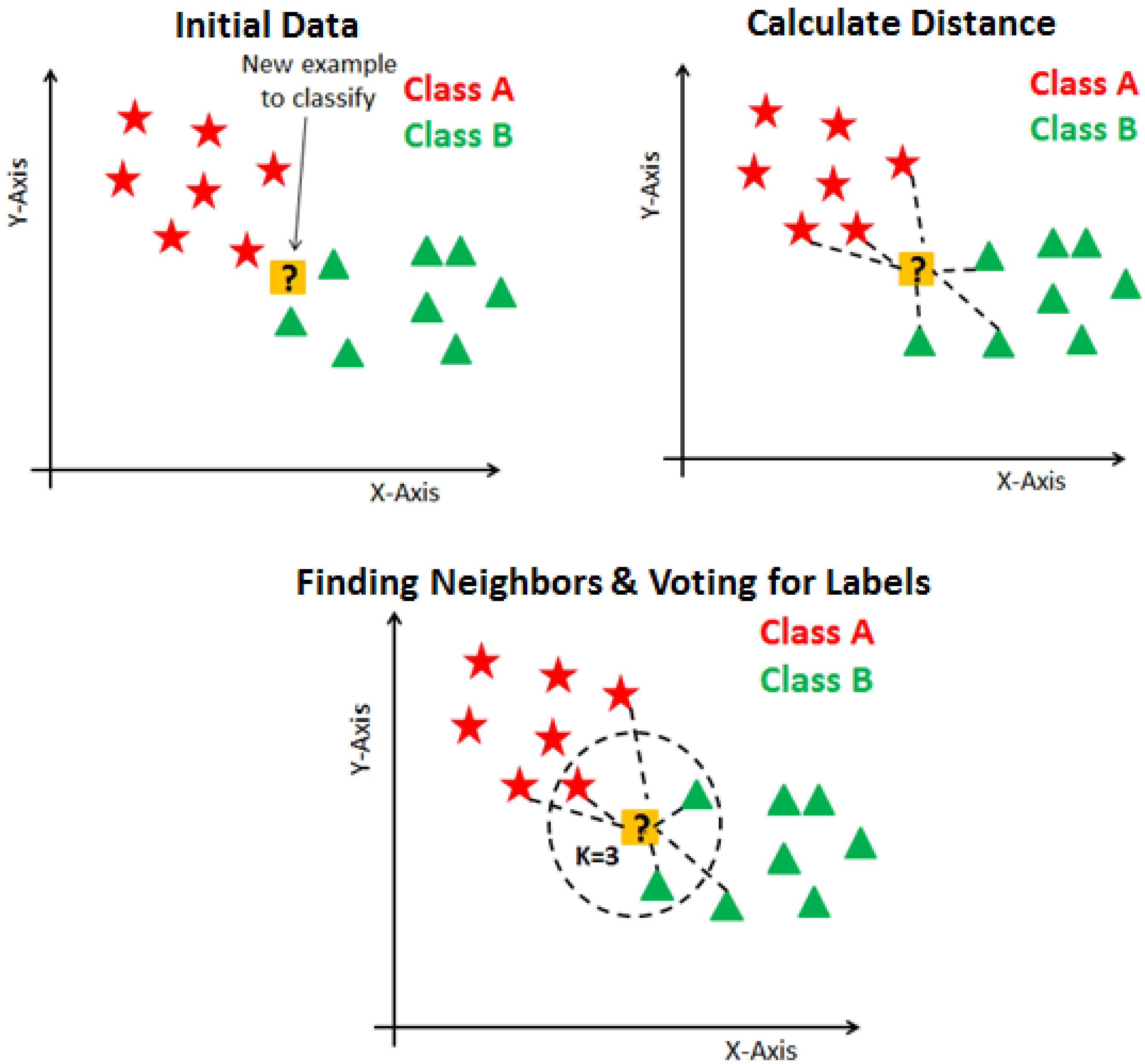
k Nearest Neighbors

- It can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- KNN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
- It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

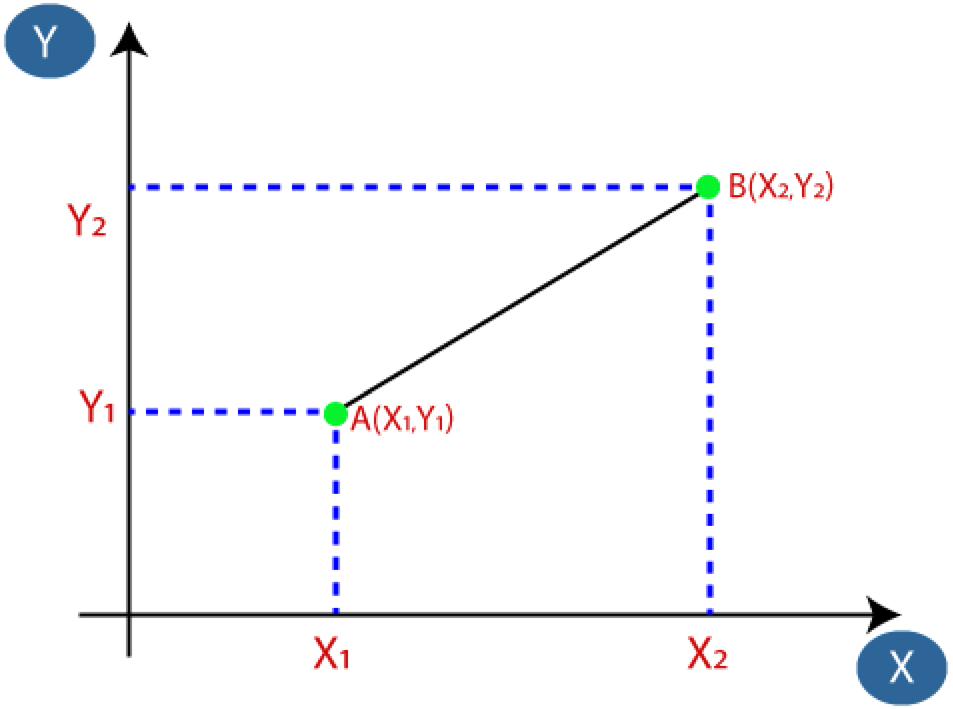
Why kNN?


Before K-NN
X₂
Category B
New data point
Category A
X₁
K-NN
After K-NN
X₂
Category B
New data point assigned to Category 1
Category A
X₁

How kNN Works?



How kNN Works?



Euclidean Distance between A_1 and B_2 = $\sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

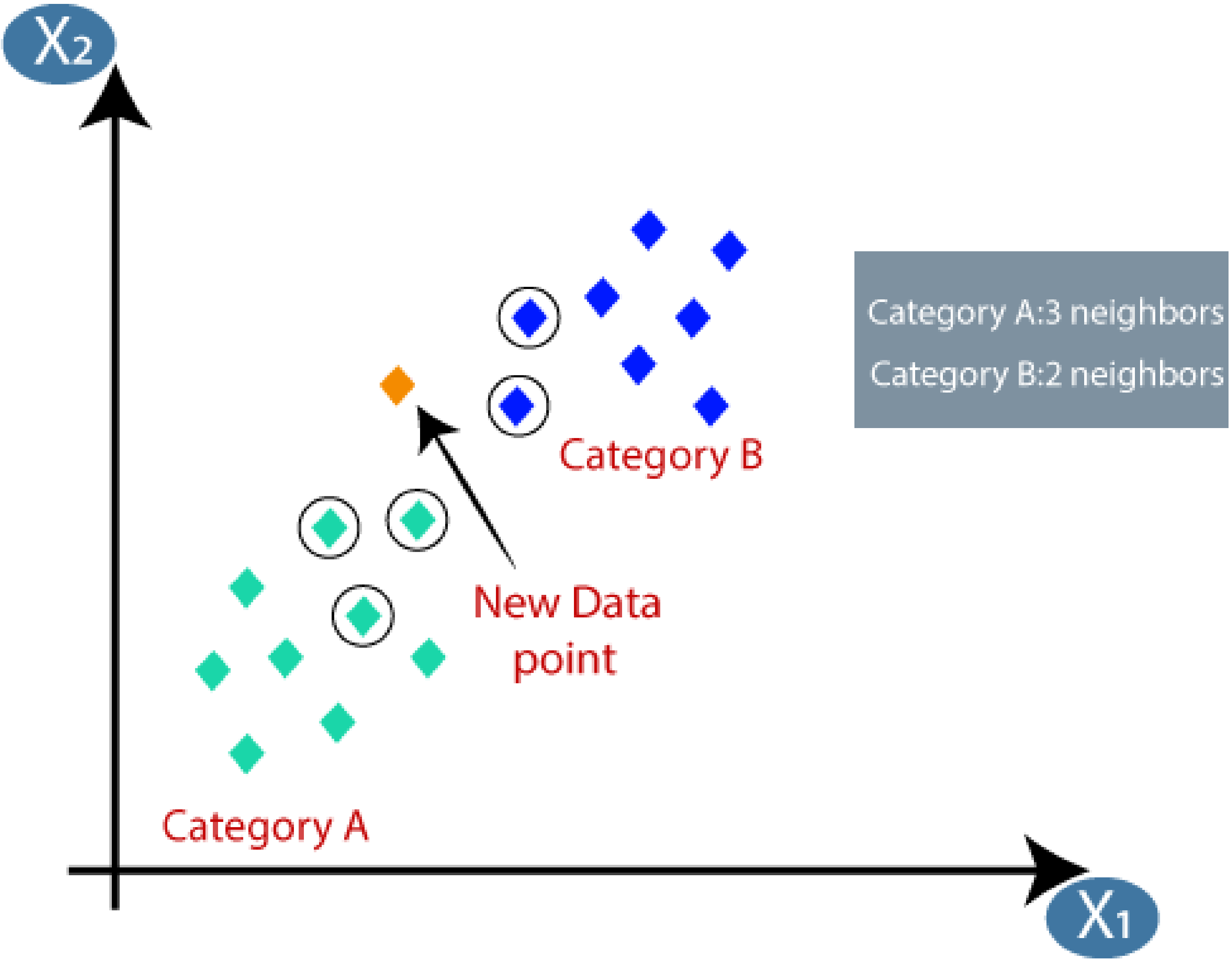
Distance functions

Euclidean $$\sqrt{\sum_{i=1}^{k}(x_i-y_i)^2}$$

Manhattan $$\sum_{i=1}^{k}|x_i-y_i|$$

Minkowski $$\left(\sum_{i=1}^{k}(|x_i-y_i|)^q\right)^{1/q}$$

How kNN Works?



How kNN Works?

The K-NN working can be explained on the basis of the below algorithm:

Step-1: Select the number K.

Step-2: Calculate the Euclidean distance of all the data points from the point in question.

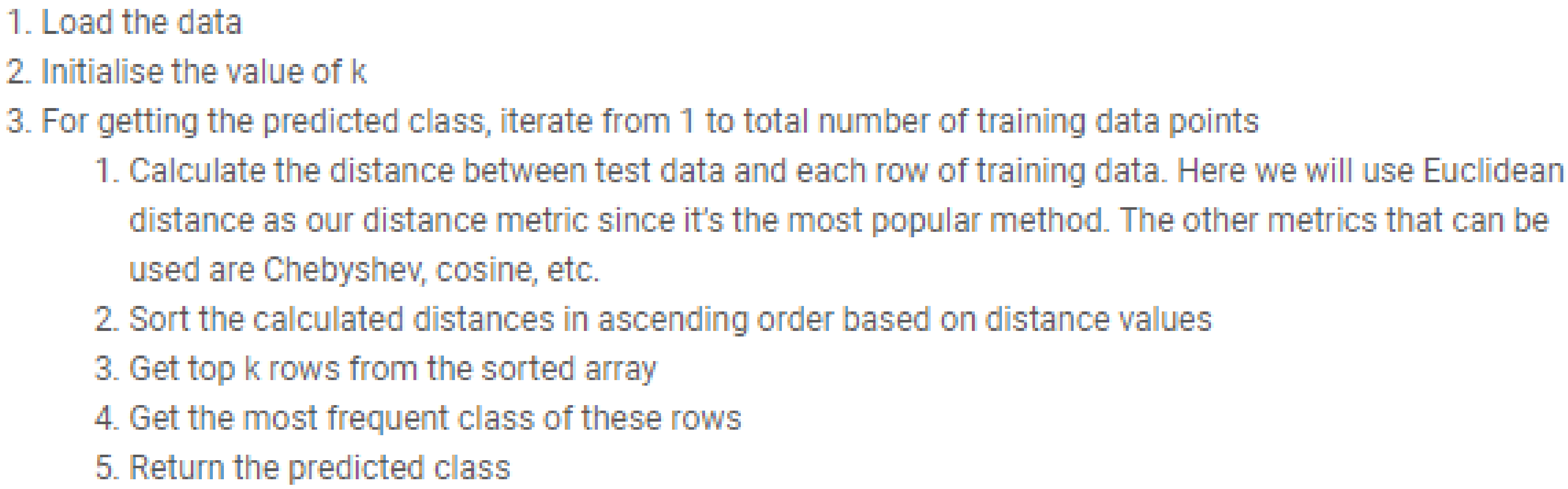
Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

Step-6: Our model is ready.

How kNN Works?



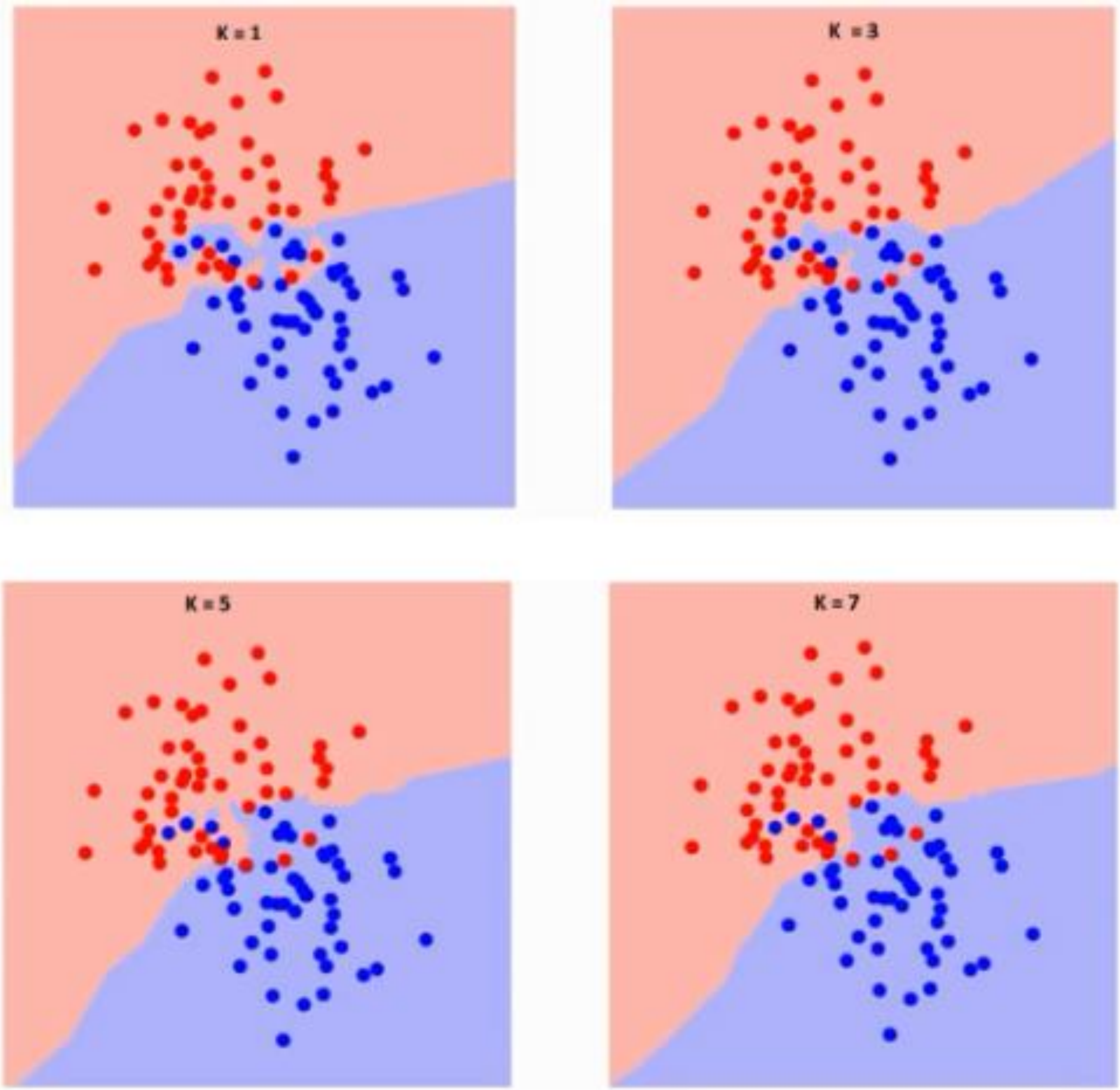
1. Load the data
2. Initialise the value of k
3. For getting the predicted class, iterate from 1 to total number of training data points
    1. Calculate the distance between test data and each row of training data. Here we will use Euclidean distance as our distance metric since it's the most popular method. The other metrics that can be used are Chebyshev, cosine, etc.
    2. Sort the calculated distances in ascending order based on distance values
    3. Get top k rows from the sorted array
    4. Get the most frequent class of these rows
    5. Return the predicted class

How to select the value of K in kNN?

Below are some points to remember while selecting the value of K in the K-NN algorithm:

- There is **no way to determine** the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
- A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
- Large values for K are good, but it may find some difficulties.

How to select the value of K in kNN?



kNN Example

X	Y	Label
7	7	A
7	4	A
3	4	B
1	4	B

New Point = (3, 7)

kNN Example

New Point = (3, 7)

X	Y	Label	
7	7	A	$(3-7)^2 + (7-7)^2$
7	4	A	$(3-7)^2 + (4-7)^2$
3	4	B	$(3-3)^2 + (4-7)^2$
1	4	B	$(3-1)^2 + (4-7)^2$

X	Y	Label	
3	4	B	9
1	4	B	13
7	7	A	16
7	4	A	25

Advantages and Disadvantages of kNN?

Advantages:

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

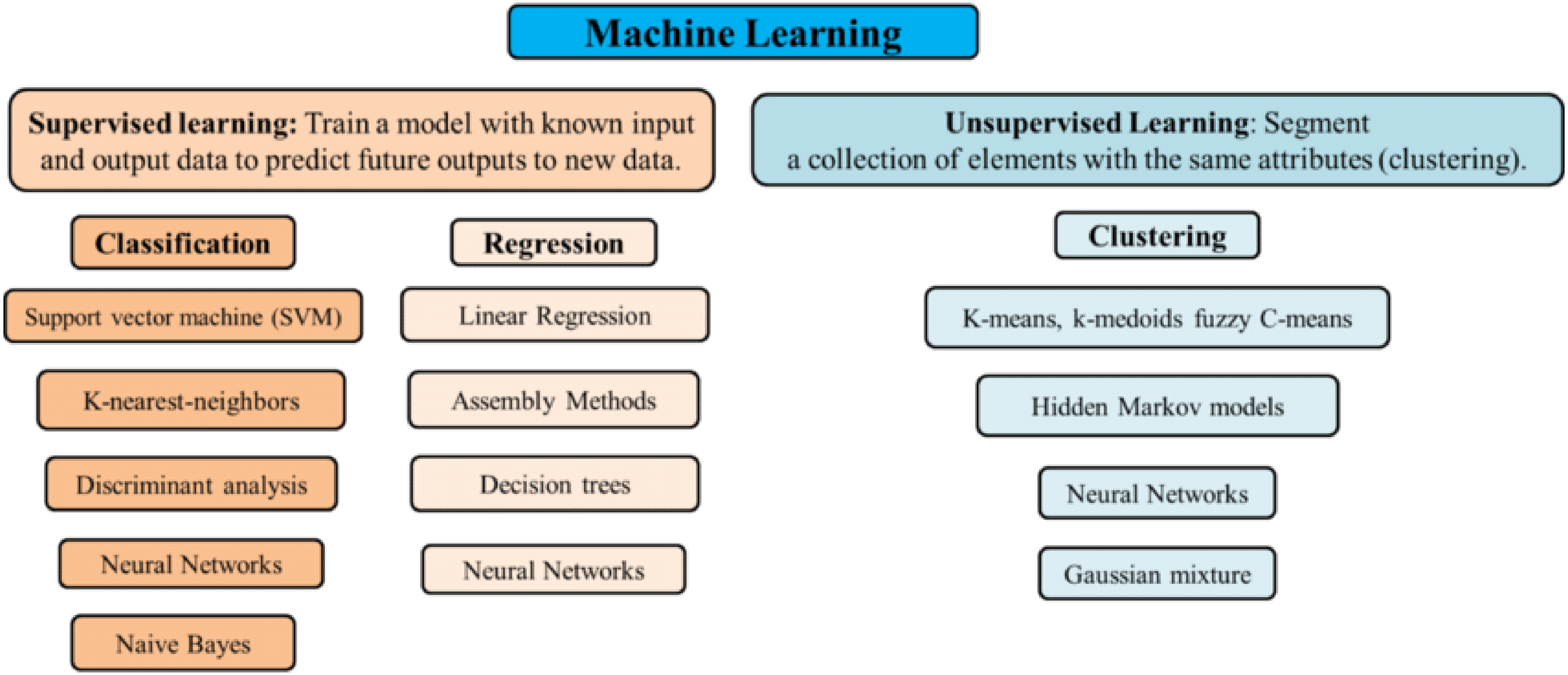
Disadvantages:

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

kNN Implementation

Implement kNN Algorithm

Machine Learning Algorithms



Machine Learning

Supervised learning: Train a model with known input and output data to predict future outputs to new data.

Classification

- Support vector machine (SVM)
- K-nearest-neighbors
- Discriminant analysis
- Neural Networks
- Naive Bayes

Regression

- Linear Regression
- Assembly Methods
- Decision trees
- Neural Networks

Unsupervised Learning: Segment a collection of elements with the same attributes (clustering).

Clustering

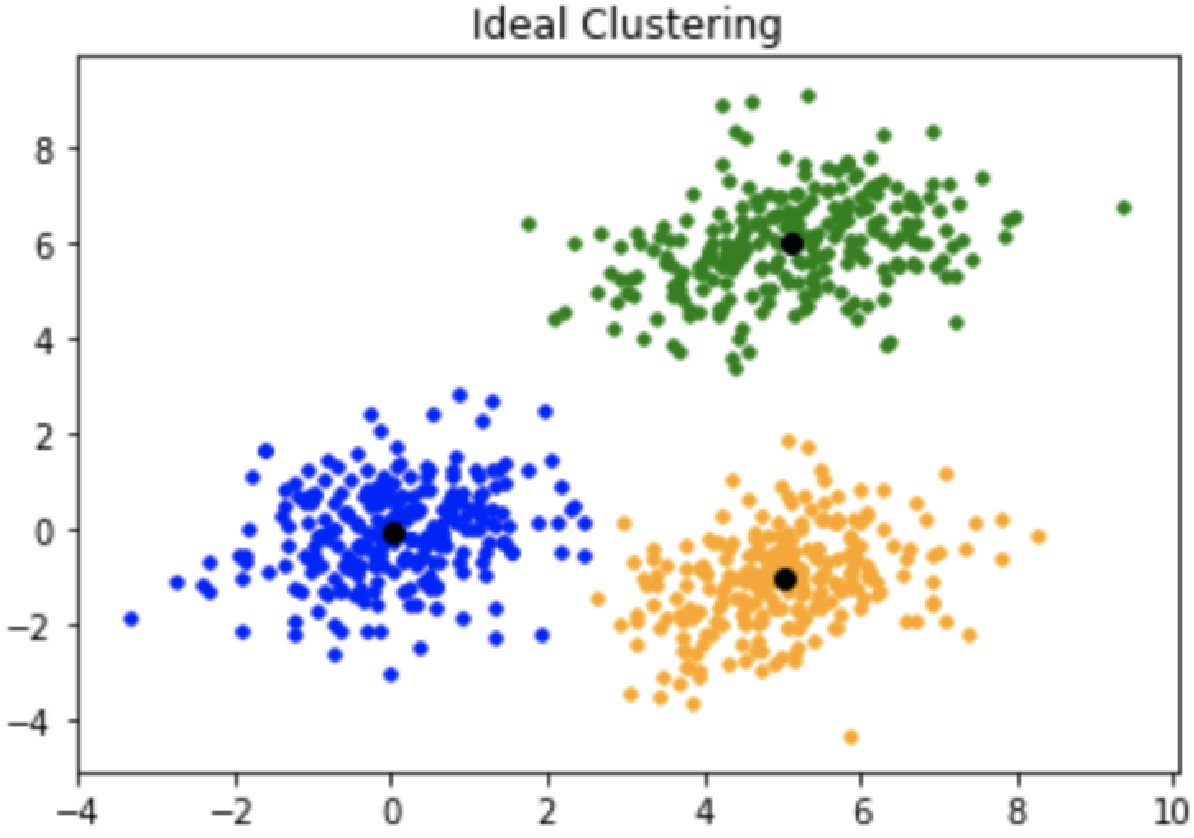
- K-means, k-medoids fuzzy C-means
- Hidden Markov models
- Neural Networks
- Gaussian mixture

Clustering

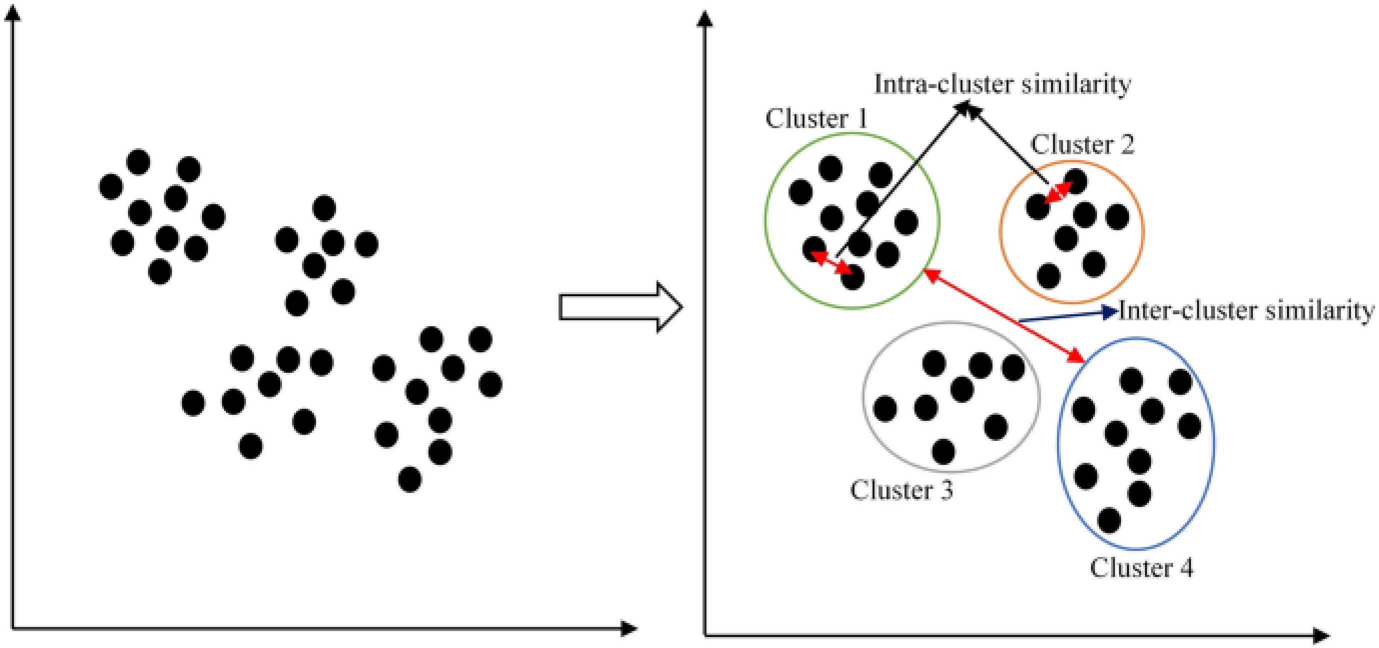
- Clustering is one of the most common exploratory data analysis technique used to get an intuition about the structure of the data.
- It can be defined as the task of identifying subgroups in the data such that data points in the same subgroup (cluster) are very similar while data points in different clusters are very different.
- we try to find homogeneous subgroups within the data such that data points in each cluster are as similar as possible according to a similarity measure such as Euclidean-based distance or correlation-based distance.

Clustering

- Clustering is considered an **unsupervised learning** method since we don’t have the ground truth to compare the output of the clustering algorithm to the true labels to evaluate its performance.



Clustering



(a) Data objects

(b) Clustered data objects

K-Means

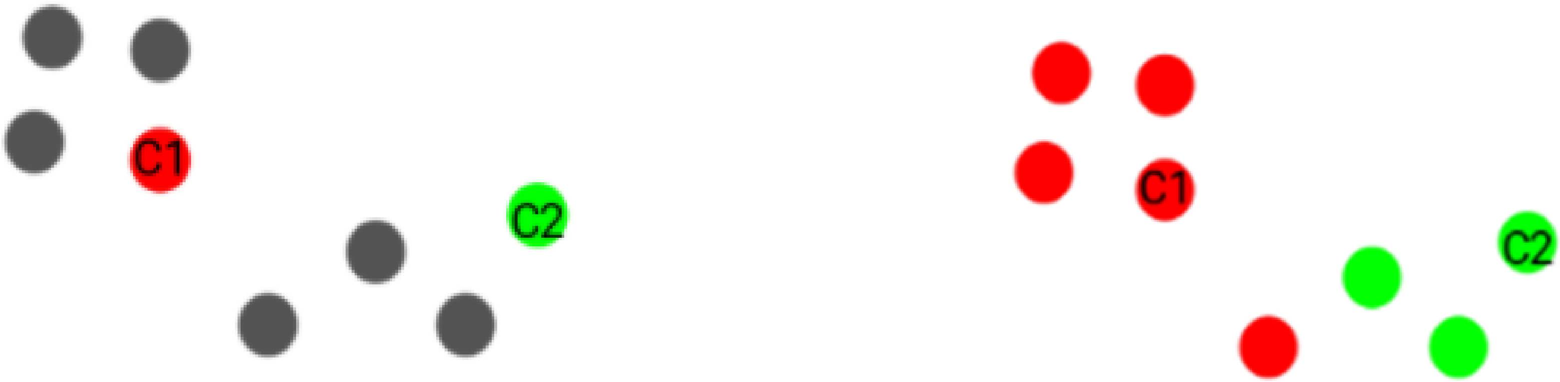
- K-means clustering is one of the simplest and popular unsupervised machine learning algorithms.
- Unsupervised algorithms make inferences from datasets using **only input vectors** without referring to known, or labelled, outcomes.
- The objective of K-means is to group similar data points together and discover underlying patterns. To achieve this objective, K-means looks for a fixed number (k) of clusters in a dataset.
  - A cluster refers to a collection of data points aggregated together because of certain similarities.

K-Means

- We define a target number k, which refers to **the number of centroids we need in the dataset**. A centroid is the imaginary or real location representing the center of the cluster.
- Every data point is allocated to each of the clusters through reducing the **within-cluster sum of squares**.
- K-means algorithm is an **iterative algorithm** that tries to partition the dataset into K pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group

K-Means Algorithm

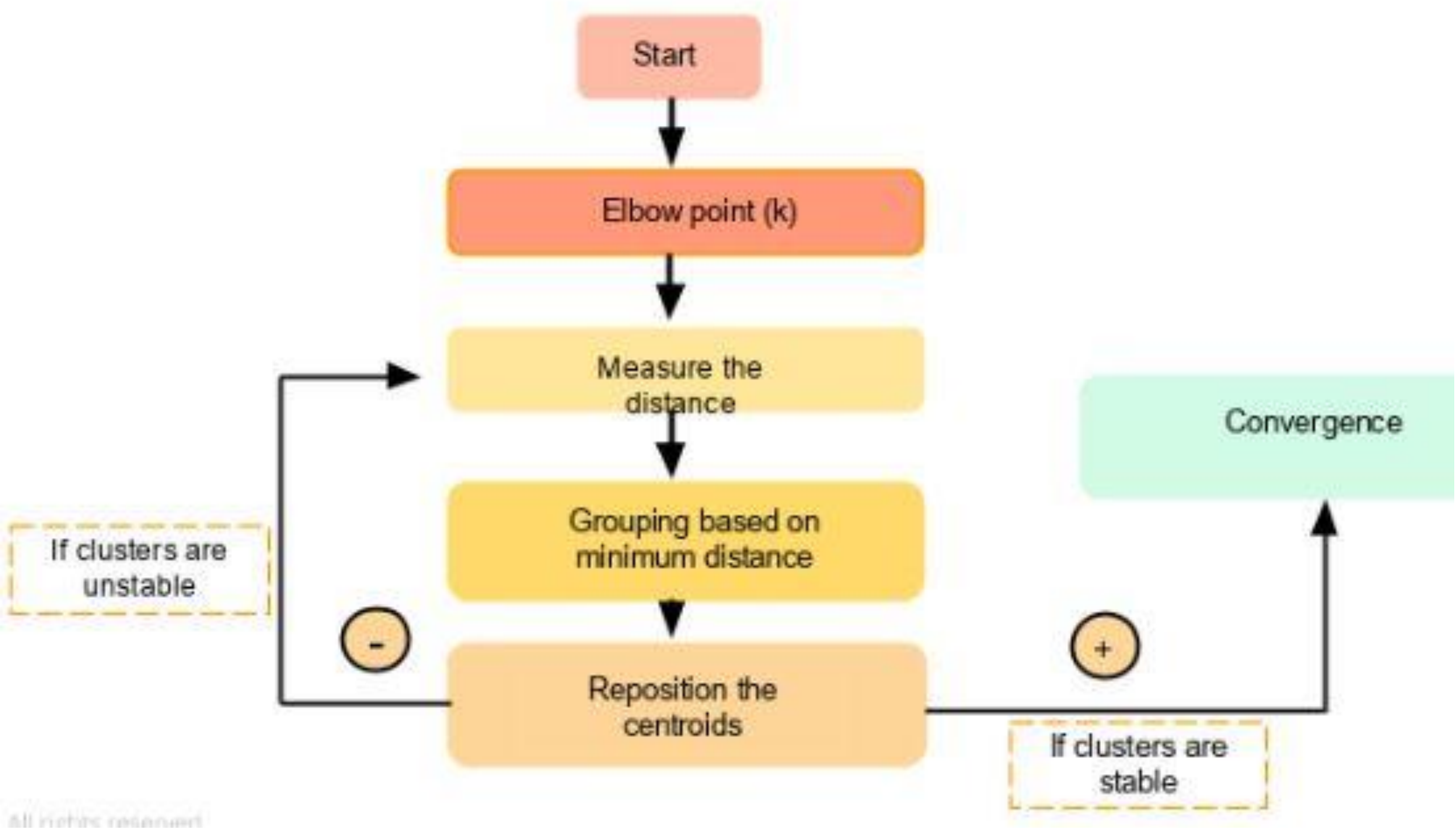
- The first step in k-means is to pick the number of clusters, K.
- Next, we randomly select the centroid for each cluster. Let’s say we want to have 2 clusters, so k is equal to 2 here. We then randomly select the centroid.
- Once we have initialized the centroids, we assign each point to the closest cluster centroid:



K-Means Algorithm

- Specify number of clusters K.
- Initialize centroids by first shuffling the dataset and then randomly selecting K data points for the centroids without replacement.
- Compute the sum of the squared distance between data points and all centroids.
- Assign each data point to the closest cluster (centroid).
- Compute the centroids for the clusters by taking the average of the all-data points that belong to each cluster.
- Keep iterating until there is no change to the centroids, i.e., assignment of data points to clusters isn’t changing.

K-Means Algorithm



Start

Elbow point (k)

Measure the distance

Grouping based on minimum distance

Reposition the centroids

If clusters are unstable

-

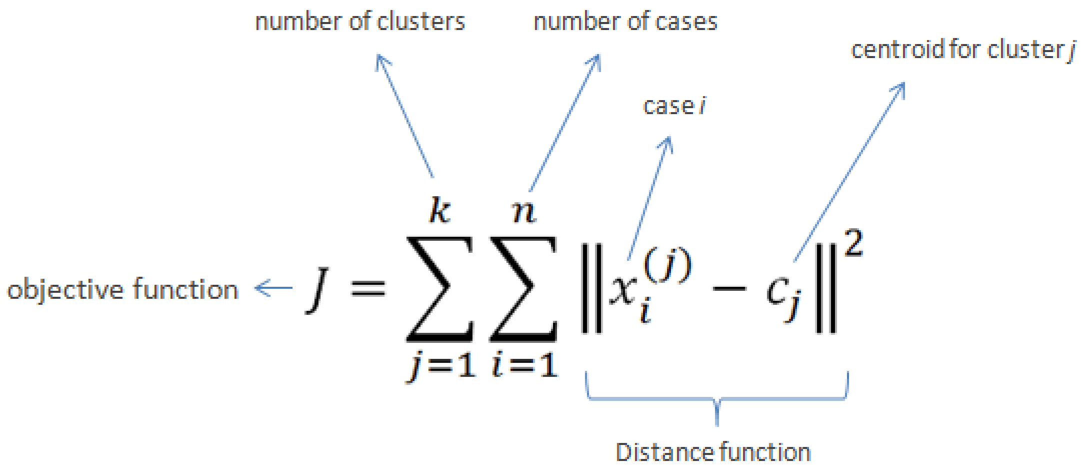
+

If clusters are stable

Convergence

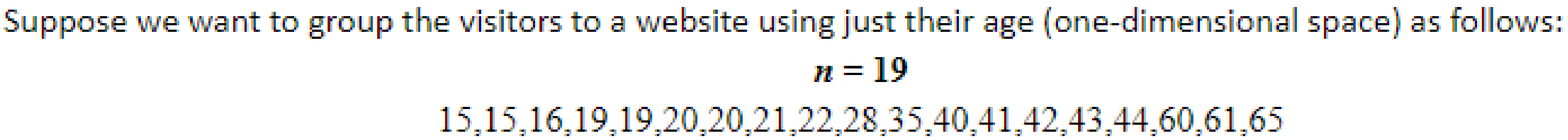
Expectation Maximization

- The approach k-means follows to solve the problem is called **Expectation-Maximization**.
- The E-step is assigning the data points to the closest cluster. The M-step is computing the centroid of each cluster.



$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

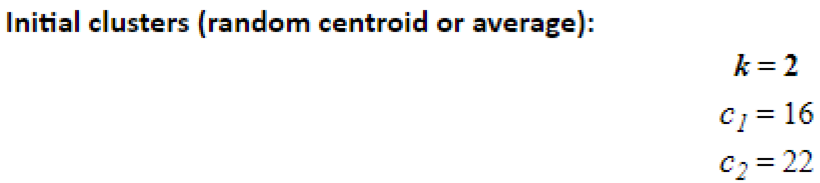
K-Means Example



Suppose we want to group the visitors to a website using just their age (one-dimensional space) as follows:

$$n = 19$$

15,15,16,19,19,20,20,21,22,28,35,40,41,42,43,44,60,61,65

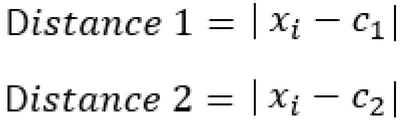


Initial clusters (random centroid or average):

$$k = 2$$

$$c_1 = 16$$

$$c_2 = 22$$

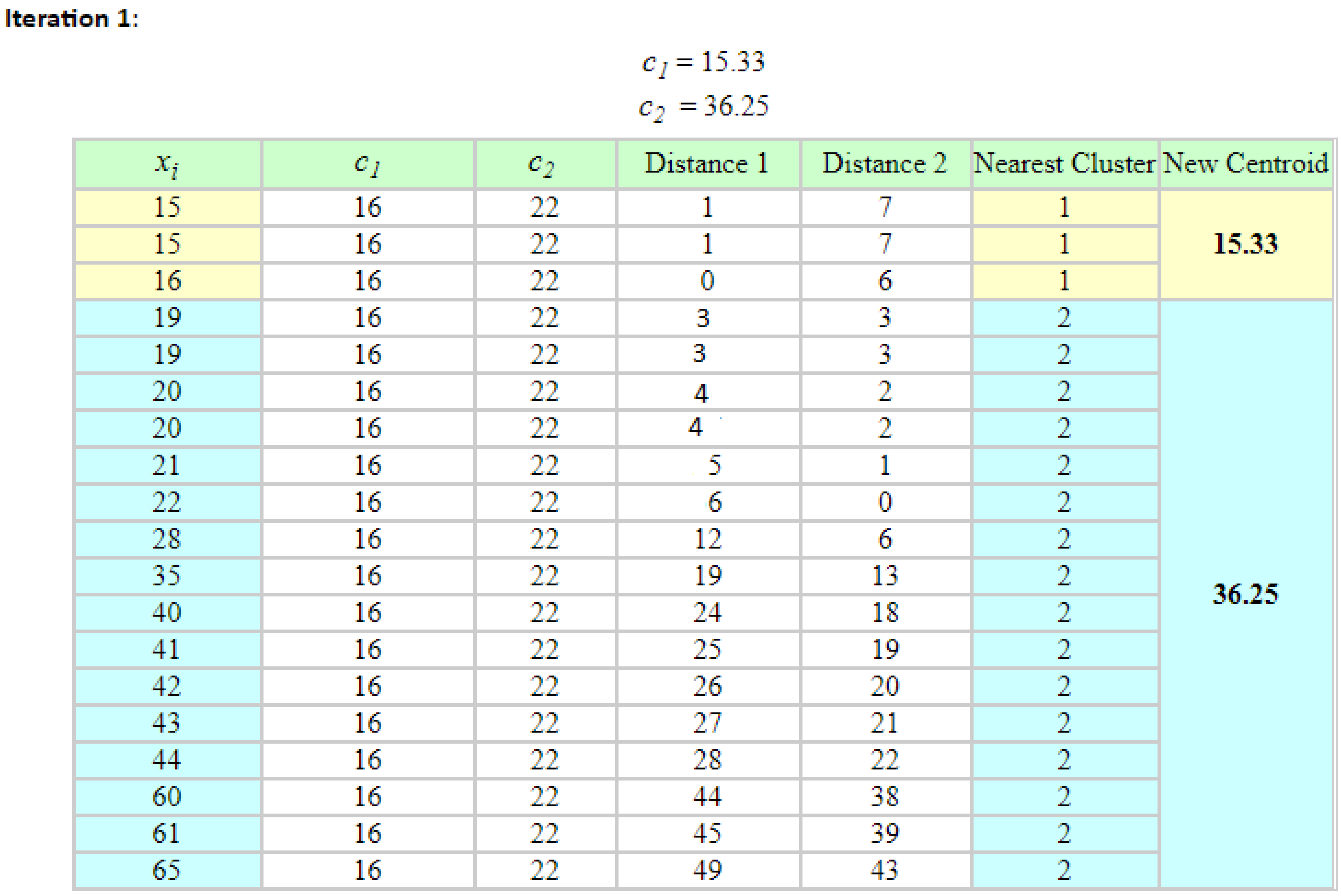


$$Distance\ 1 = |x_i - c_1|$$

$$Distance\ 2 = |x_i - c_2|$$

Source: https://www.saedsayad.com/clustering_kmeans.htm

K-Means Example



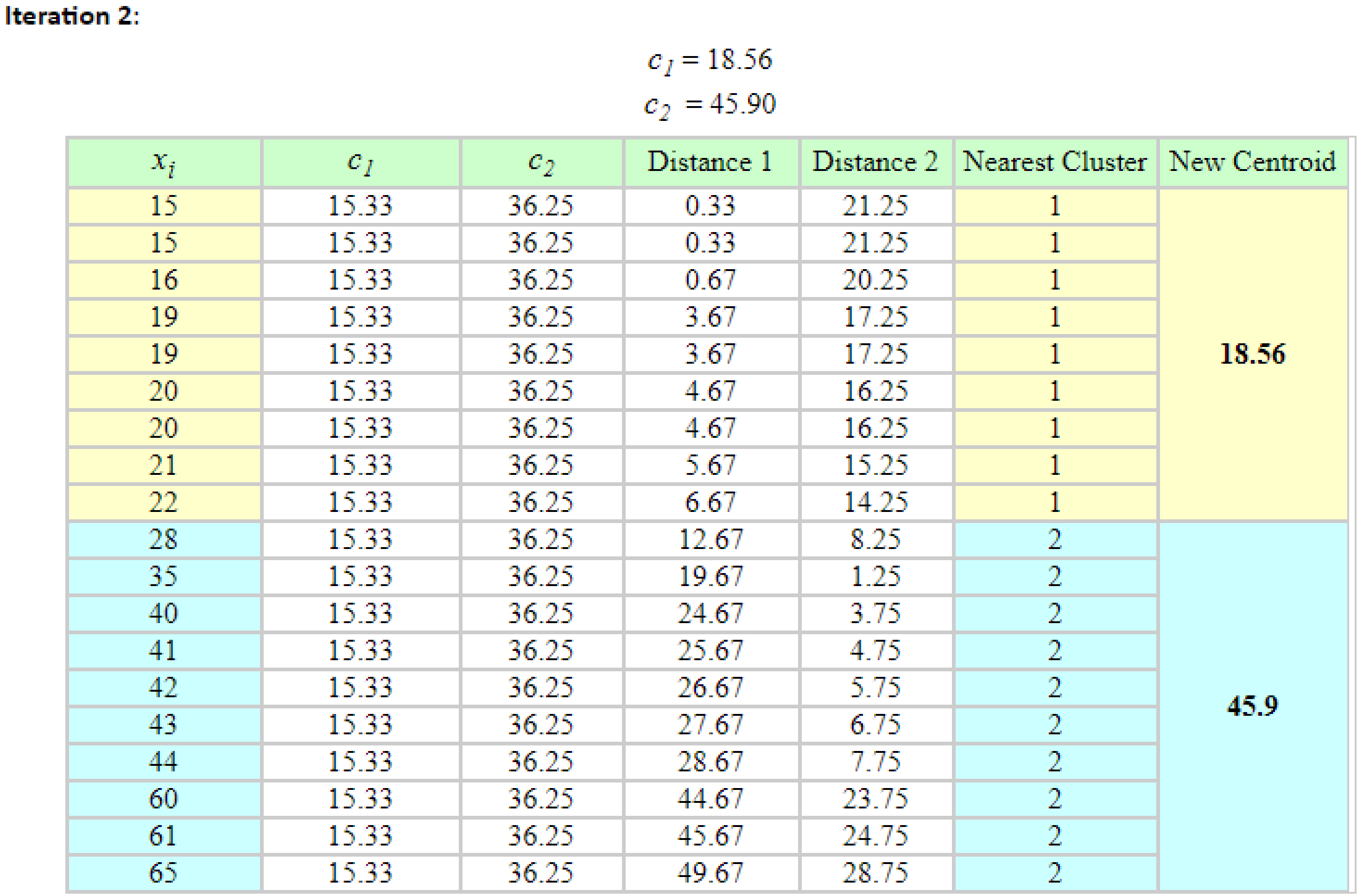
Iteration 1:

$c_1 = 15.33$

$c_2 = 36.25$

x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	16	22	1	7	1	**15.33**
15	16	22	1	7	1	
16	16	22	0	6	1	
19	16	22	3	3	2	**36.25**
19	16	22	3	3	2	
20	16	22	4	2	2	
20	16	22	4	2	2	
21	16	22	5	1	2	
22	16	22	6	0	2	
28	16	22	12	6	2	
35	16	22	19	13	2	
40	16	22	24	18	2	
41	16	22	25	19	2	
42	16	22	26	20	2	
43	16	22	27	21	2	
44	16	22	28	22	2	
60	16	22	44	38	2	
61	16	22	45	39	2	
65	16	22	49	43	2	

K-Means Example



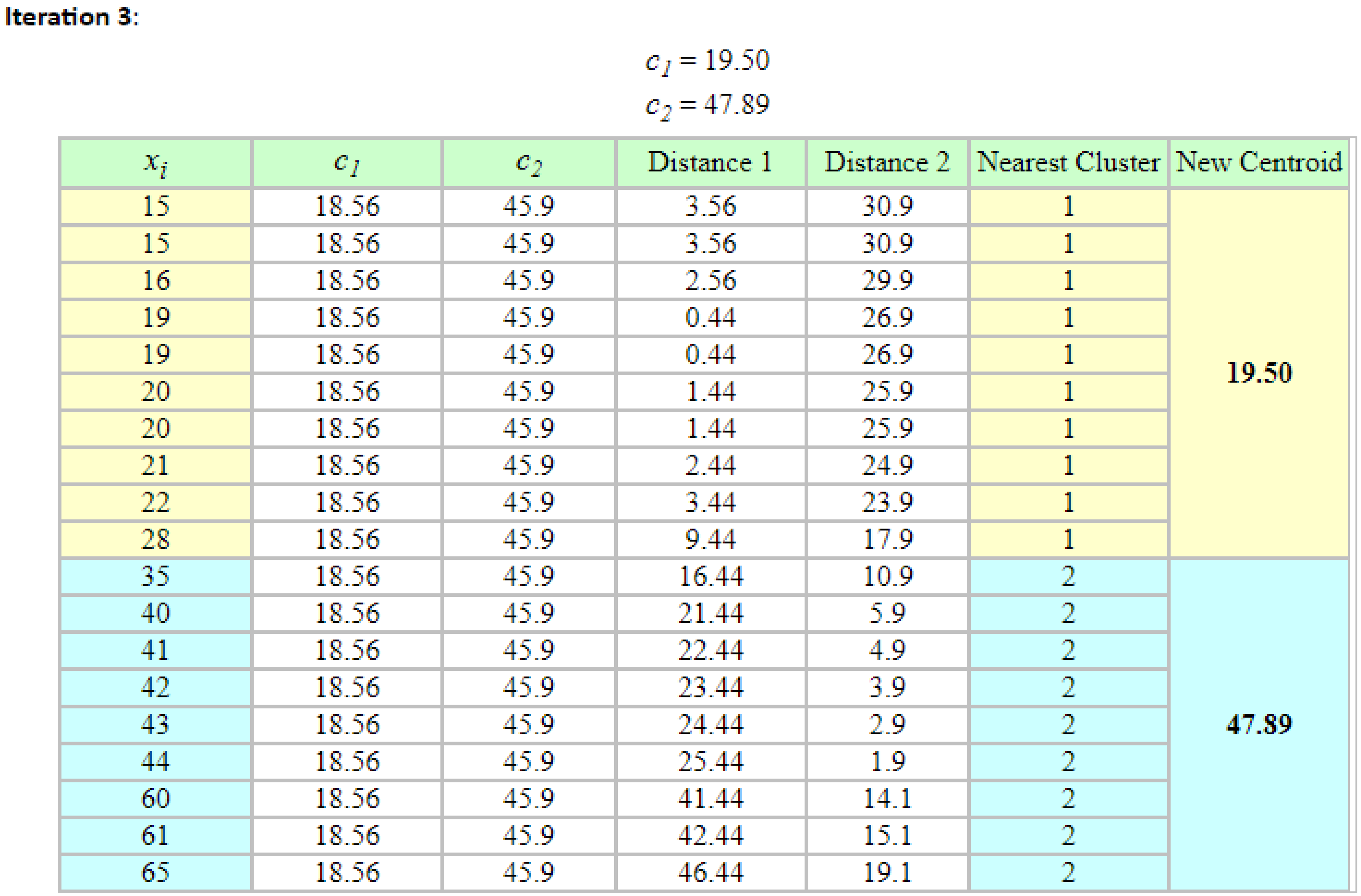
Iteration 2:

$c_1 = 18.56$

$c_2 = 45.90$

x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	15.33	36.25	0.33	21.25	1	**18.56**
15	15.33	36.25	0.33	21.25	1	
16	15.33	36.25	0.67	20.25	1	
19	15.33	36.25	3.67	17.25	1	
19	15.33	36.25	3.67	17.25	1	
20	15.33	36.25	4.67	16.25	1	
20	15.33	36.25	4.67	16.25	1	
21	15.33	36.25	5.67	15.25	1	
22	15.33	36.25	6.67	14.25	1	
28	15.33	36.25	12.67	8.25	2	**45.9**
35	15.33	36.25	19.67	1.25	2	
40	15.33	36.25	24.67	3.75	2	
41	15.33	36.25	25.67	4.75	2	
42	15.33	36.25	26.67	5.75	2	
43	15.33	36.25	27.67	6.75	2	
44	15.33	36.25	28.67	7.75	2	
60	15.33	36.25	44.67	23.75	2	
61	15.33	36.25	45.67	24.75	2	
65	15.33	36.25	49.67	28.75	2	

K-Means Example



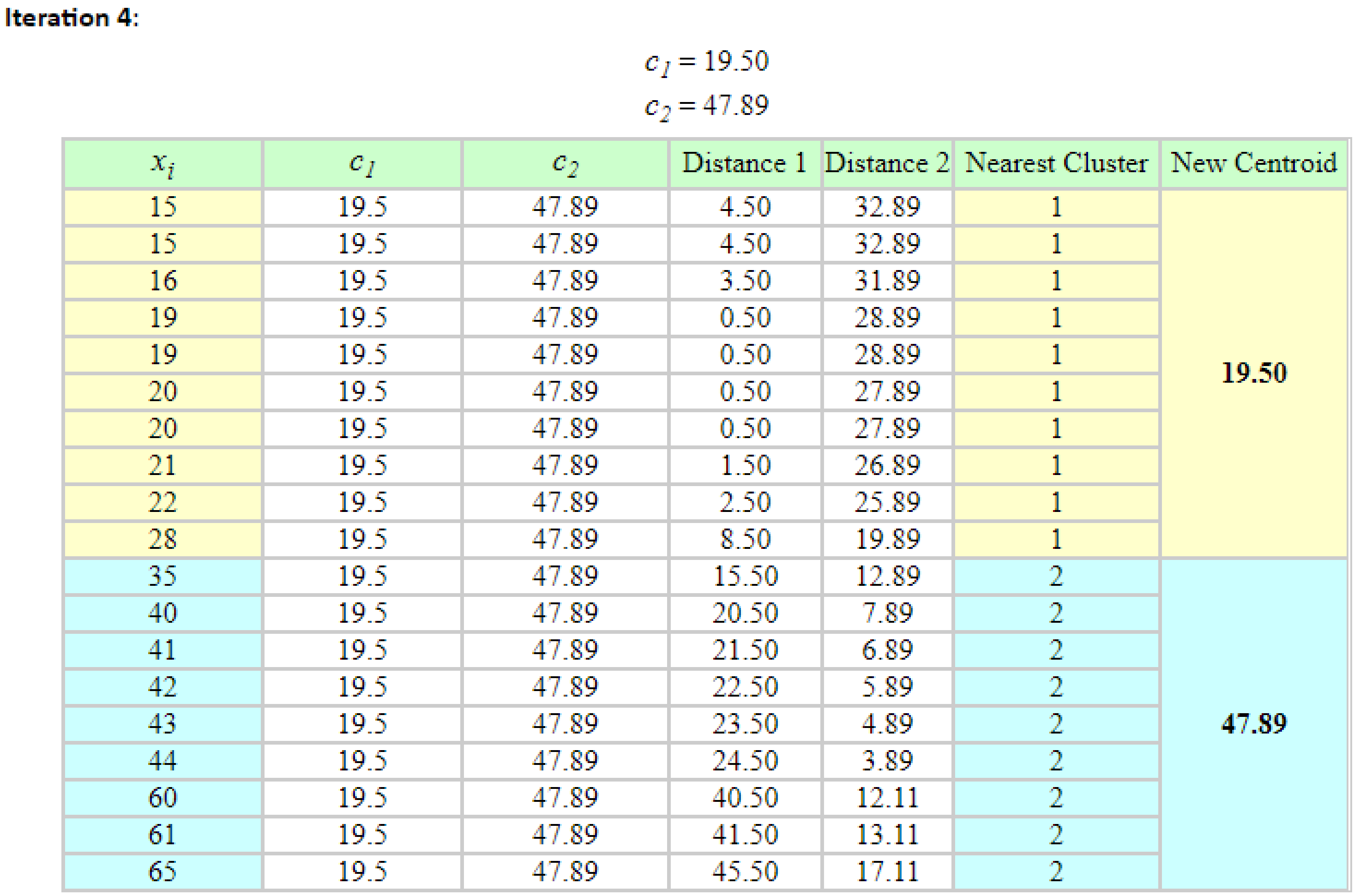
Iteration 3:

$c_1 = 19.50$

$c_2 = 47.89$

x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	18.56	45.9	3.56	30.9	1	**19.50**
15	18.56	45.9	3.56	30.9	1	
16	18.56	45.9	2.56	29.9	1	
19	18.56	45.9	0.44	26.9	1	
19	18.56	45.9	0.44	26.9	1	
20	18.56	45.9	1.44	25.9	1	
20	18.56	45.9	1.44	25.9	1	
21	18.56	45.9	2.44	24.9	1	
22	18.56	45.9	3.44	23.9	1	
28	18.56	45.9	9.44	17.9	1	
35	18.56	45.9	16.44	10.9	2	**47.89**
40	18.56	45.9	21.44	5.9	2	
41	18.56	45.9	22.44	4.9	2	
42	18.56	45.9	23.44	3.9	2	
43	18.56	45.9	24.44	2.9	2	
44	18.56	45.9	25.44	1.9	2	
60	18.56	45.9	41.44	14.1	2	
61	18.56	45.9	42.44	15.1	2	
65	18.56	45.9	46.44	19.1	2	

K-Means Example



Iteration 4:

$c_1 = 19.50$

$c_2 = 47.89$

x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	19.5	47.89	4.50	32.89	1	**19.50**
15	19.5	47.89	4.50	32.89	1	
16	19.5	47.89	3.50	31.89	1	
19	19.5	47.89	0.50	28.89	1	
19	19.5	47.89	0.50	28.89	1	
20	19.5	47.89	0.50	27.89	1	
20	19.5	47.89	0.50	27.89	1	
21	19.5	47.89	1.50	26.89	1	
22	19.5	47.89	2.50	25.89	1	
28	19.5	47.89	8.50	19.89	1	
35	19.5	47.89	15.50	12.89	2	**47.89**
40	19.5	47.89	20.50	7.89	2	
41	19.5	47.89	21.50	6.89	2	
42	19.5	47.89	22.50	5.89	2	
43	19.5	47.89	23.50	4.89	2	
44	19.5	47.89	24.50	3.89	2	
60	19.5	47.89	40.50	12.11	2	
61	19.5	47.89	41.50	13.11	2	
65	19.5	47.89	45.50	17.11	2	

K-Means Implementation

Implement k-Means Algorithm

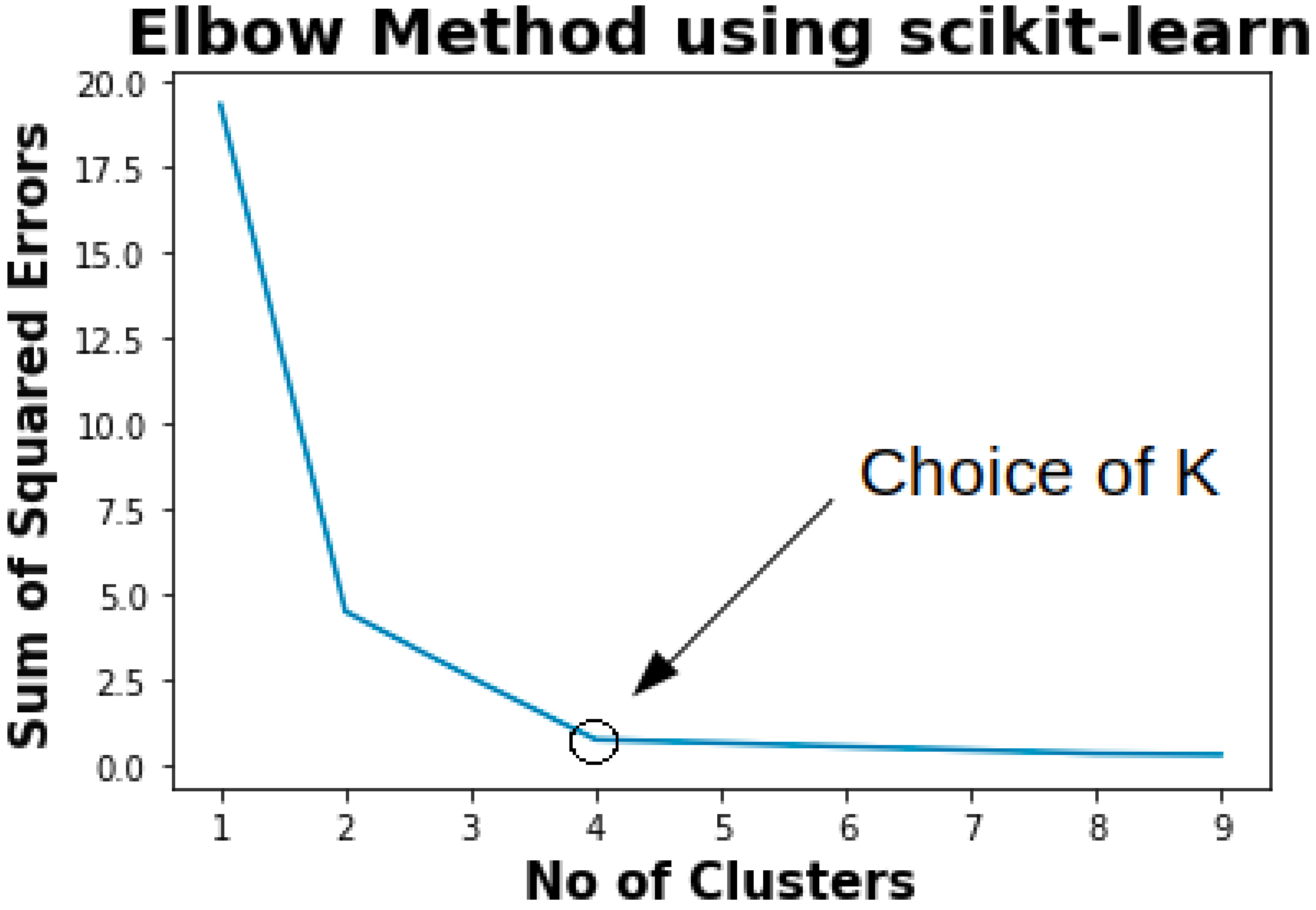
Optimal Value of K in K-means

- Elbow Method
- Silhouette Method

Elbow Method

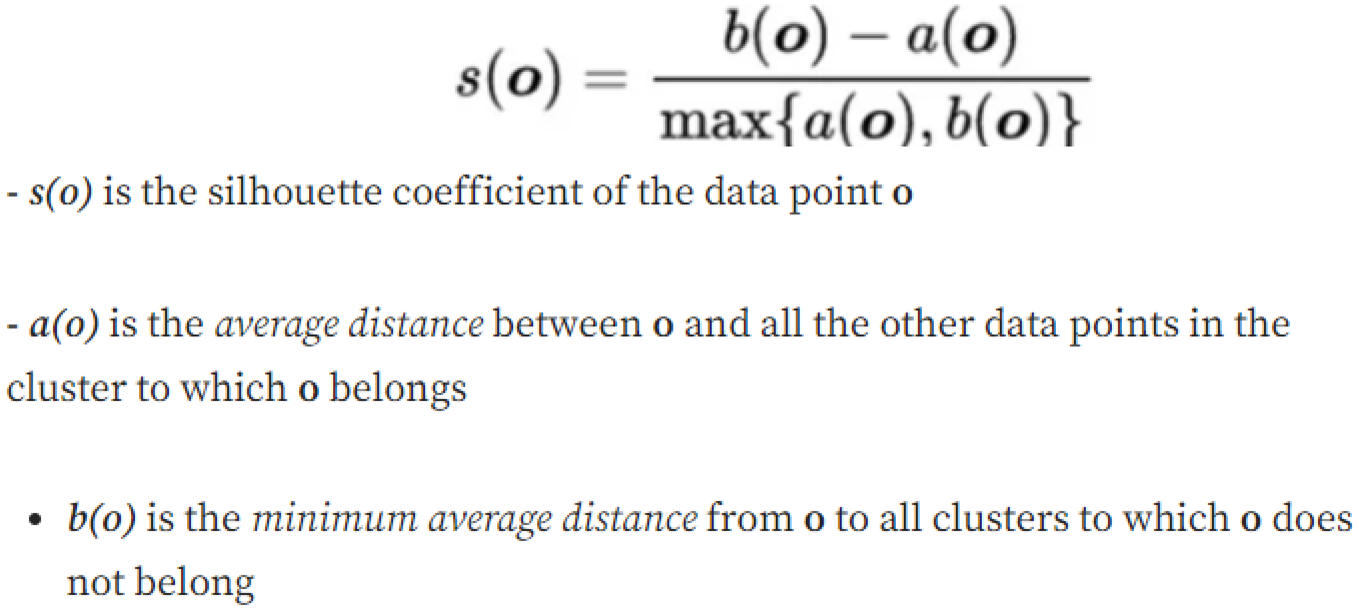
- The elbow method runs k-means clustering on the dataset for a range of values for k (say from 1-10) and then for each value of k computes an average score for all clusters.
- We can compute Within-Cluster Sum of Squares (WCSS), the sum of square distances from each point to its assigned center.
- We then draw k vs. WCSS.

Elbow Method



Silhouette Method

- The equation for calculating the silhouette coefficient for a particular data point:



$$s(o) = \frac{b(o) - a(o)}{\max\{a(o), b(o)\}}$$

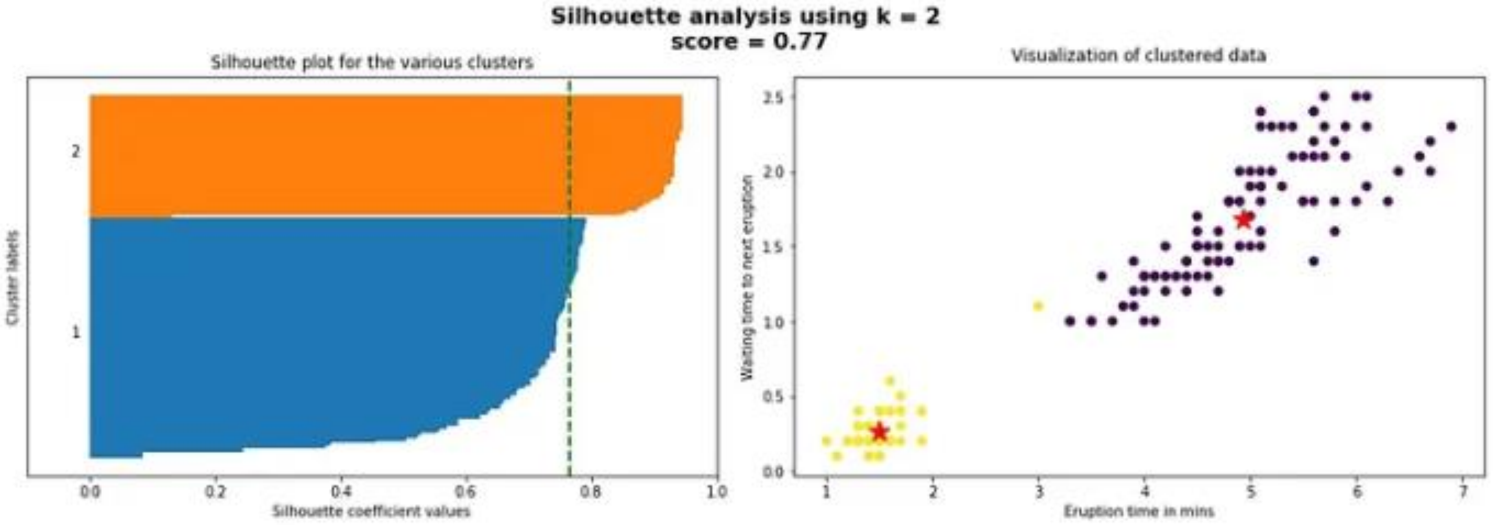
- *s(o)* is the silhouette coefficient of the data point **o**

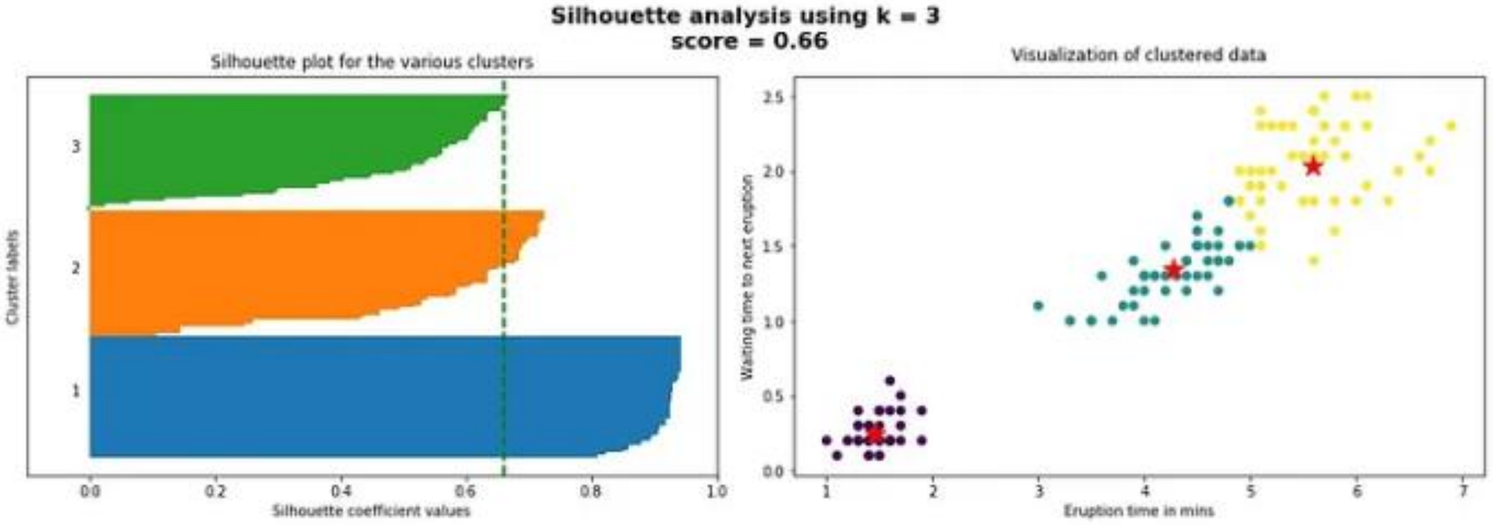
- *a(o)* is the *average distance* between **o** and all the other data points in the cluster to which **o** belongs

- *b(o)* is the *minimum average distance* from **o** to all clusters to which **o** does not belong

The value of the silhouette coefficient is between [-1, 1]. A score of 1 denotes the best meaning that the data point o is very compact within the cluster to which it belongs and far away from the other clusters. The worst value is -1. Values near 0 denote overlapping clusters.

Silhouette Method





Summary

- K Nearest Neighbors
- K-means