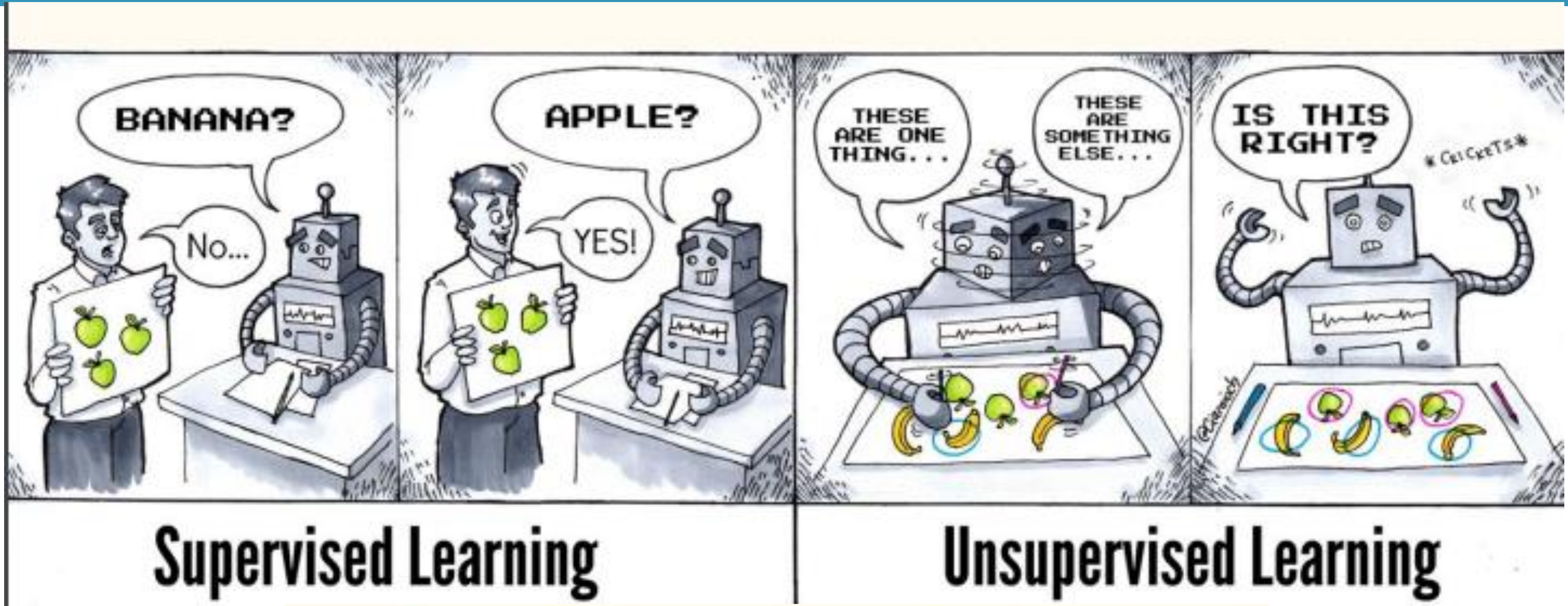
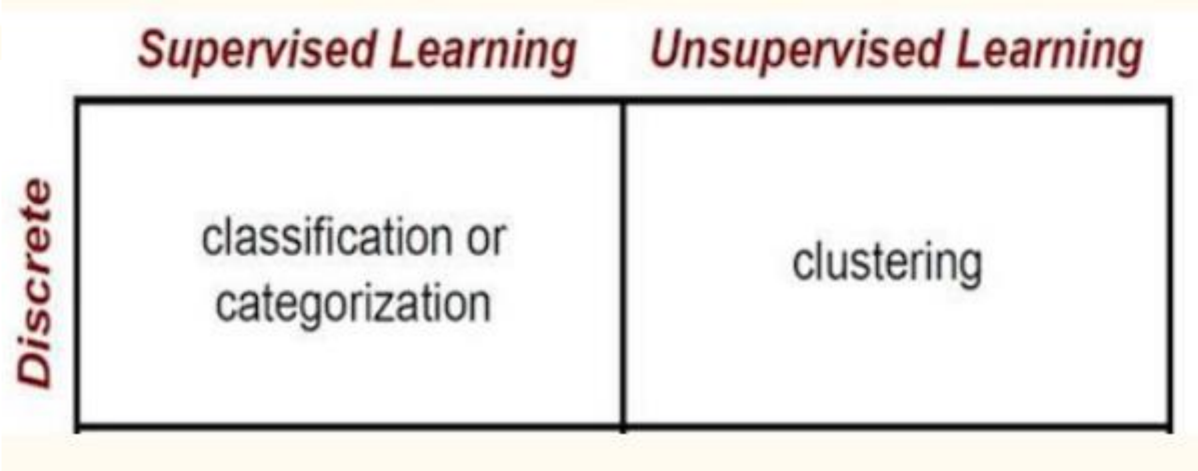
Unit 4





	Supervised Learning	Unsupervised Learning
Discrete	classification or categorization	clustering

Overview

1. Choosing distance metrics
2. Different clustering approaches
3. hierarchical agglomerative clustering
4. k-means, DBSCAN
5. Relative merits of each method
6. clustering tendency and quality.
7. Computer science and engineering applications Data mining, Network protocols, analysis of Web traffic.

Common Distance Metrics

Euclidean Distance: $d(p, q) = sqrt(\Sigma(pi - qi)^2)$

- Measures the straight-line distance between two points in Euclidean space.
- Most commonly used distance metric due to its simplicity and intuitive interpretation.
- Suitable for data with continuous numerical features.

Manhattan Distance:$d(p, q) = \Sigma|pi - qi|$

- Measures the distance between two points by summing the absolute differences of their Cartesian coordinates.
- Also known as L1 norm or Taxicab distance.
- More robust to outliers compared to Euclidean distance.
- Useful for data with categorical or ordinal features.

Minkowski Distance: $d(p, q) = (\Sigma|pi - qi|^p)^{(1/p)}$

- Generalization of Euclidean and Manhattan distances.
- Defined as the p-th root of the sum of the p-th powers of the differences between the coordinates of the two points.
- Euclidean distance is a special case of Minkowski distance with $p=2$.
- Manhattan distance is a special case of Minkowski distance with $p=1$.

Different Clustering techniques

1. Hierarchical Clustering

- **Agglomerative Hierarchical Clustering:**
  - Starts with each data point as a separate cluster.
- **Divisive Hierarchical Clustering:**
  - Starts with all data points in a single cluster.

2. Partitional Clustering

- **K-Means Clustering:**
  - Divides data into K clusters, where K is specified beforehand.
  - Assigns each data point to the nearest cluster centroid.
  - Recomputes the centroids based on the assigned points.
  - Iterates until the cluster assignments stabilize.
- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):**
  - Groups together points that are closely packed together (high density).
  - Identifies clusters of arbitrary shape.
  - Can handle noise and outliers effectively.

3. Fuzzy Clustering

- **Fuzzy C-Means Clustering:**
  - Assigns each data point to multiple clusters with a degree of membership.
  - The sum of membership degrees for a data point across all clusters equals 1.

4. Model-Based Clustering

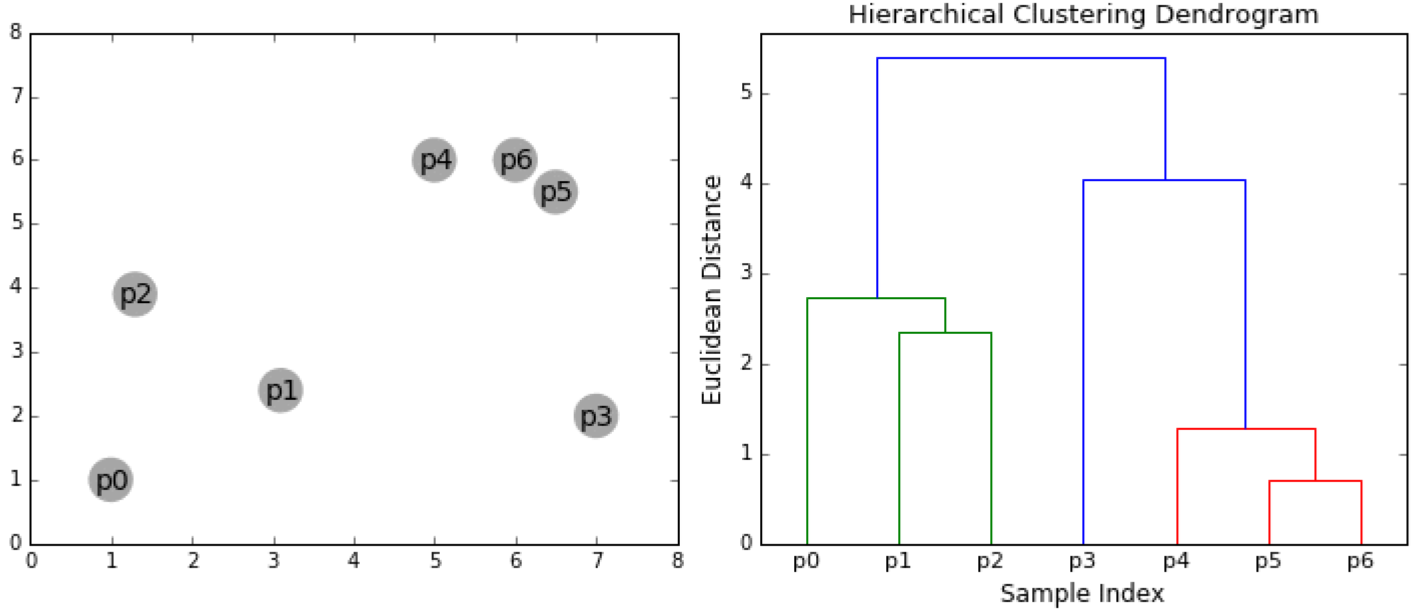
- **Gaussian Mixture Models (GMM):**
  - Assumes that the data is generated from a mixture of Gaussian distributions.
  - Identifies the number of clusters and their parameters.
- **Evaluation Metrics:**
  - Metrics like silhouette score, Calinski-Harabasz index, and Davies-Bouldin index can be used to evaluate the quality of clustering.

hierarchical agglomerative clustering

It creates a hierarchy of clusters

How it Works:

1. **Initial Step:** Each data point is considered an individual cluster.
2. **Merge Closest Clusters:** In each iteration, the two closest clusters are merged into a single cluster.
3. **Update Distances:** The distances between the newly formed cluster and other clusters are recalculated.
4. **Repeat:** Steps 2 and 3 are repeated until all data points belong to a single cluster



Visualizing Hierarchical Clustering:

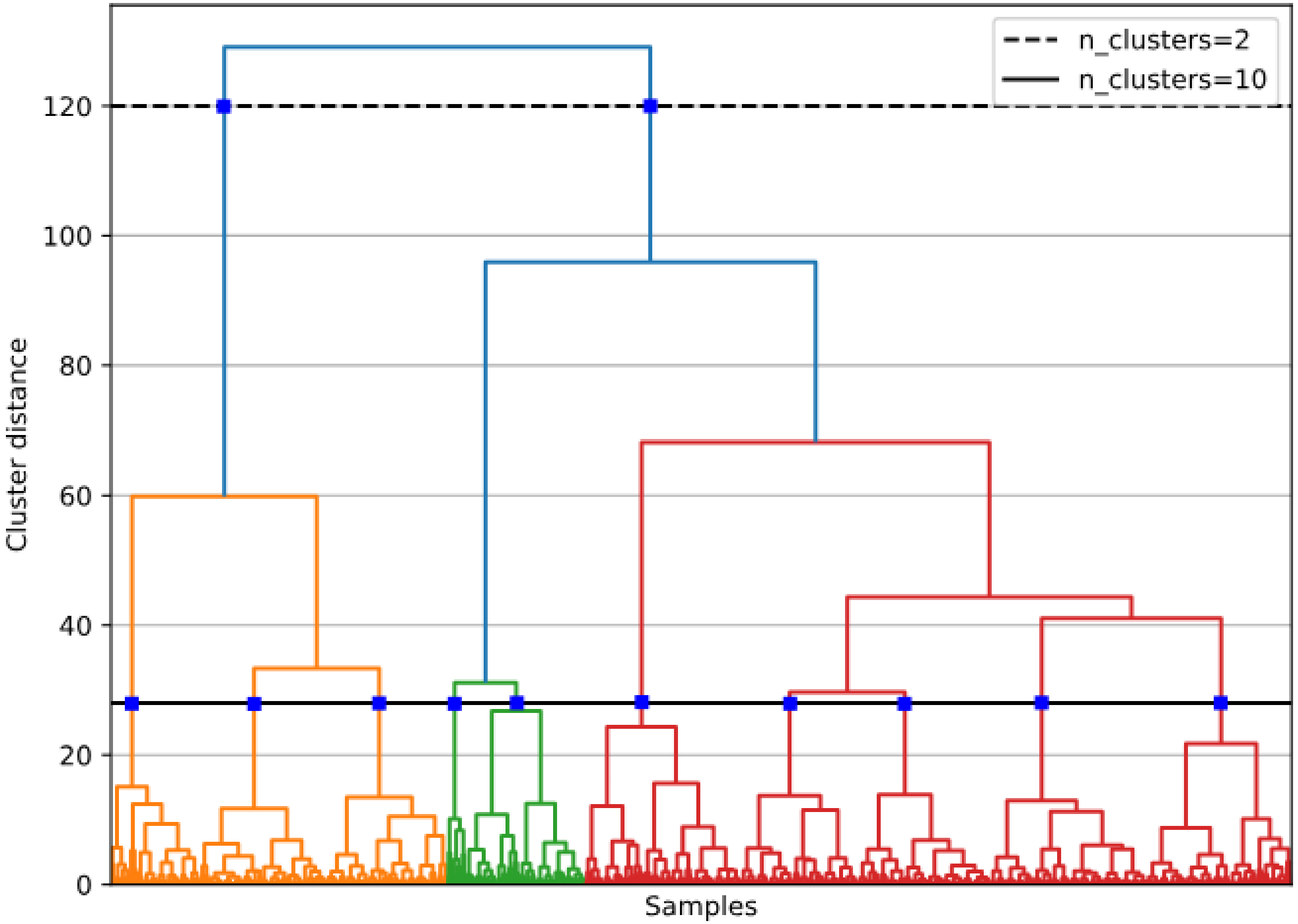
Hierarchical clustering can be visualized using a dendrogram. A dendrogram is a tree-like diagram that shows the hierarchical structure of the clusters.

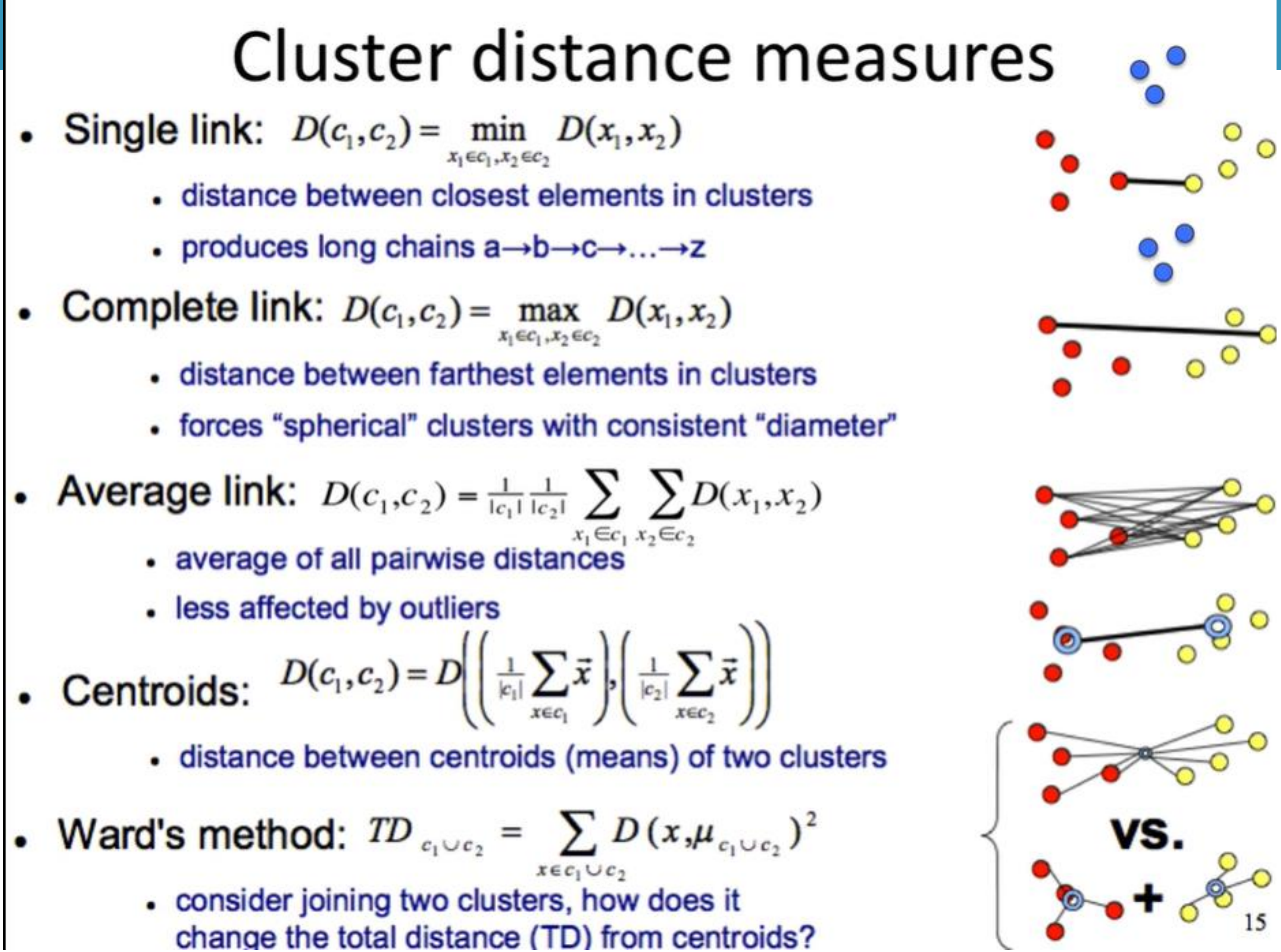
The height of the branches in the dendrogram represents the distance between the clusters.

hierarchical agglomerative clustering

Key Concepts:

- **Proximity Matrix:** A matrix that stores the pairwise distances between all data points.
- **Linkage Criteria:** The method used to determine the distance between clusters.
  - **Single Linkage:**
  - **Complete Linkage:**
  - **Average Linkage:**
  - **Ward's Linkage:**





Cluster distance measures

- Single link: $D(c_1,c_2) = \min_{x_1\in c_1, x_2\in c_2} D(x_1,x_2)$
  - distance between closest elements in clusters
  - produces long chains a→b→c→…→z
- Complete link: $D(c_1,c_2) = \max_{x_1\in c_1, x_2\in c_2} D(x_1,x_2)$
  - distance between farthest elements in clusters
  - forces “spherical” clusters with consistent “diameter”
- Average link: $D(c_1,c_2) = \frac{1}{|c_1|}\frac{1}{|c_2|}\sum_{x_1\in c_1}\sum_{x_2\in c_2} D(x_1,x_2)$
  - average of all pairwise distances
  - less affected by outliers
- Centroids: $D(c_1,c_2) = D\left(\left(\frac{1}{|c_1|}\sum_{x\in c_1}\vec{x}\right),\left(\frac{1}{|c_2|}\sum_{x\in c_2}\vec{x}\right)\right)$
  - distance between centroids (means) of two clusters
- Ward's method: $TD_{c_1\cup c_2} = \sum_{x\in c_1\cup c_2} D(x,\mu_{c_1\cup c_2})^2$
  - consider joining two clusters, how does it change the total distance (TD) from centroids?

Single Linkage

- **Sensitivity to Outliers:** Highly sensitive to outliers, as a single outlier can significantly influence the distance between two clusters.
- **Chain Effect:** Can lead to the formation of long, chain-like clusters, especially in noisy data.
- **Best for:** Identifying loose clusters or groups with a few similar objects.

Complete Linkage

- **Robust to Outliers:** Less sensitive to outliers compared to single linkage.
- **Tendency to Form Compact Clusters:** Encourages the formation of tight, compact clusters.
- **Best for:** Identifying tight, well-defined clusters.

Average Linkage

- **Balance Between Single and Complete Linkage:** Offers a balance between the sensitivity of single linkage and the robustness of complete linkage.
- **Less Sensitive to Outliers:** Less affected by outliers compared to single linkage.
- **Best for:** Identifying clusters with a moderate level of similarity.

Ward's Linkage

- **Minimizes Variance:** Aims to minimize the total within-cluster variance.
- **Tendency to Form Spherical Clusters:** Encourages the formation of spherical clusters.
- **Best for:** Identifying clusters of similar size and shape.

Single Linkage: Suitable for data with well-separated clusters, but can be sensitive to noise.

Complete Linkage: Ideal for data with compact, well-defined clusters, but can be sensitive to outliers.

Average Linkage: A good compromise between single and complete linkage, often a good default choice.

Ward's Linkage: Suitable for data with spherical clusters and a focus on minimizing within-cluster variance.

hierarchical agglomerative clustering

Advantages:

- Does not require specifying the number of clusters beforehand.
- Can handle non-spherical clusters.
- Provides a hierarchical structure that can be useful for visualization and interpretation.

Disadvantages:

- Sensitive to noise and outliers.
- Computational complexity can be high for large datasets.

Applications:

- **Biology:** Analyzing gene expression data and protein sequences.
- **Document Clustering:** Grouping similar documents together.
- **Image Segmentation:** Dividing images into meaningful regions.

K-mean

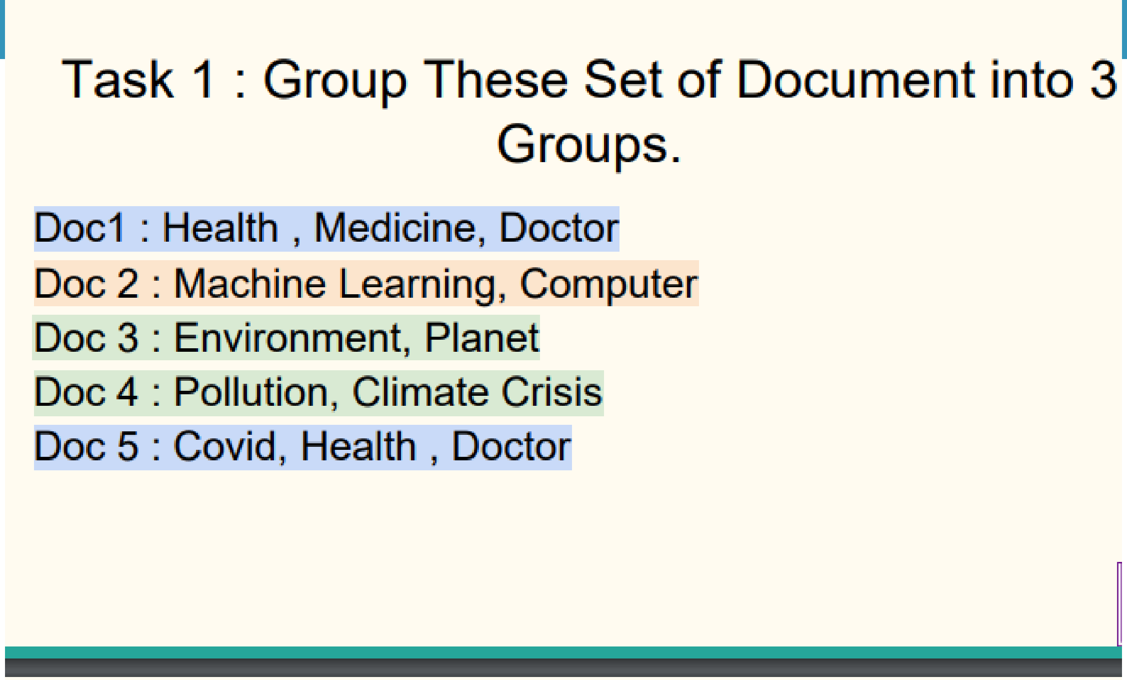
K-Means Working

Basic Steps

- Assign Cluster Centroids
- Until Convergence :
    - Cluster Assignment Step
    - Re-assigning Centroid Step

Initialization:

- **Random Initialization:** K data points are randomly selected as initial cluster centroids.
- **K-Means++:** A more sophisticated method that selects initial centroids to improve the quality of the clustering.



Task 1 : Group These Set of Document into 3 Groups.

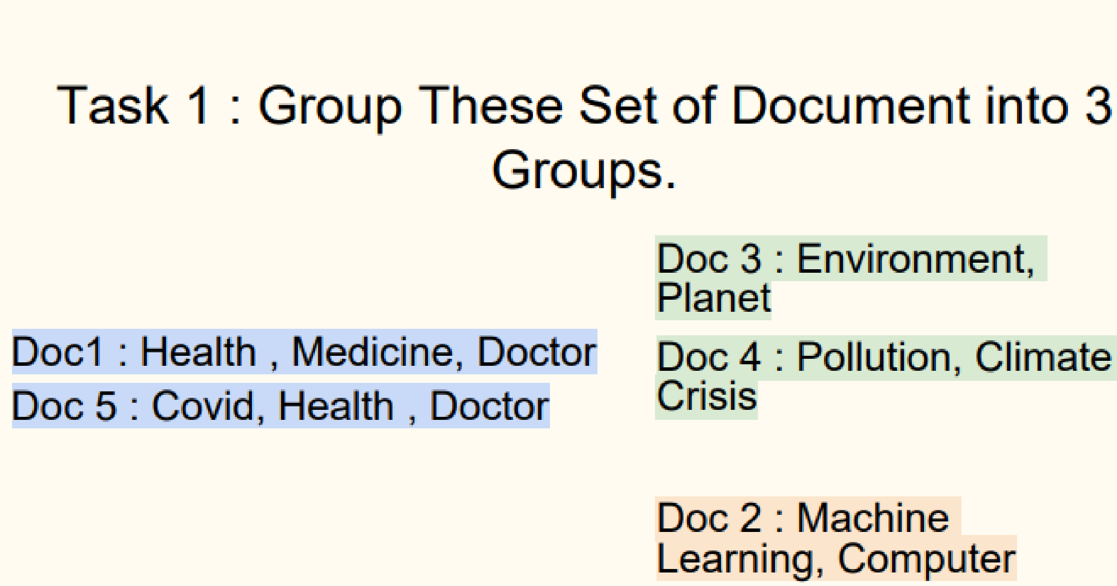
Doc1 : Health , Medicine, Doctor

Doc 2 : Machine Learning, Computer

Doc 3 : Environment, Planet

Doc 4 : Pollution, Climate Crisis

Doc 5 : Covid, Health , Doctor



Task 1 : Group These Set of Document into 3 Groups.

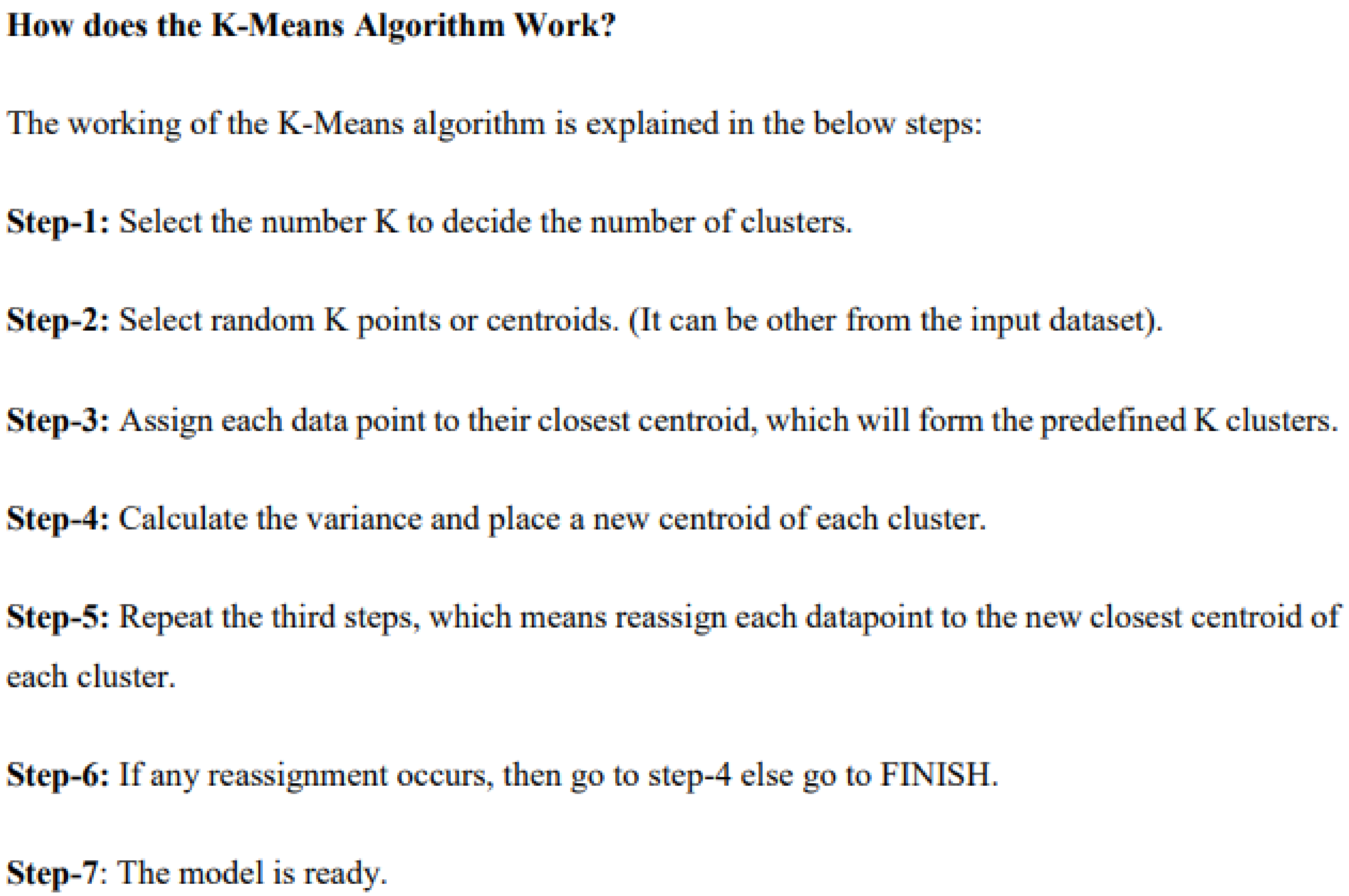
Doc 3 : Environment, Planet

Doc1 : Health , Medicine, Doctor

Doc 5 : Covid, Health , Doctor

Doc 4 : Pollution, Climate Crisis

Doc 2 : Machine Learning, Computer



How does the K-Means Algorithm Work?

The working of the K-Means algorithm is explained in the below steps:

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

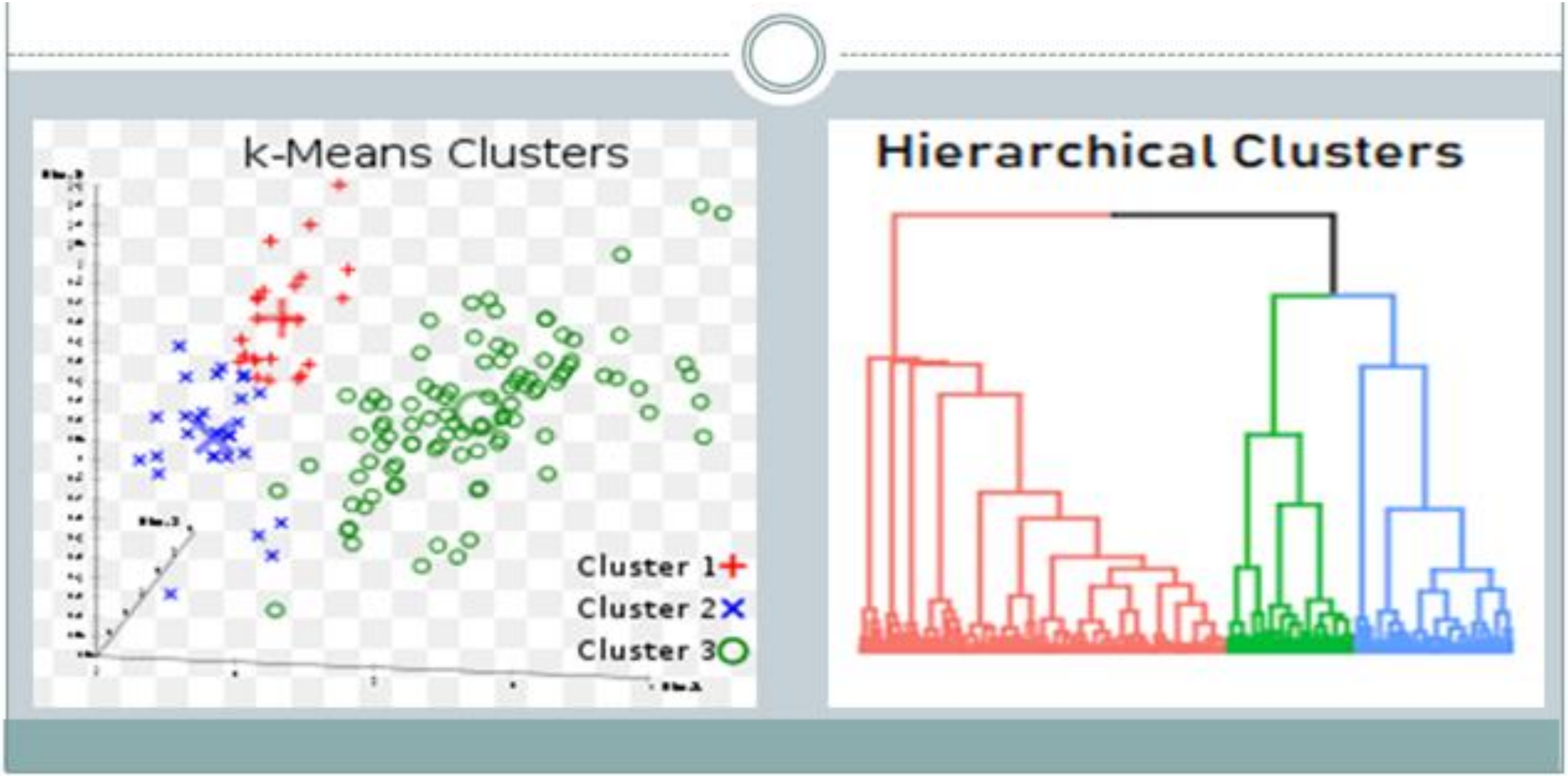
Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

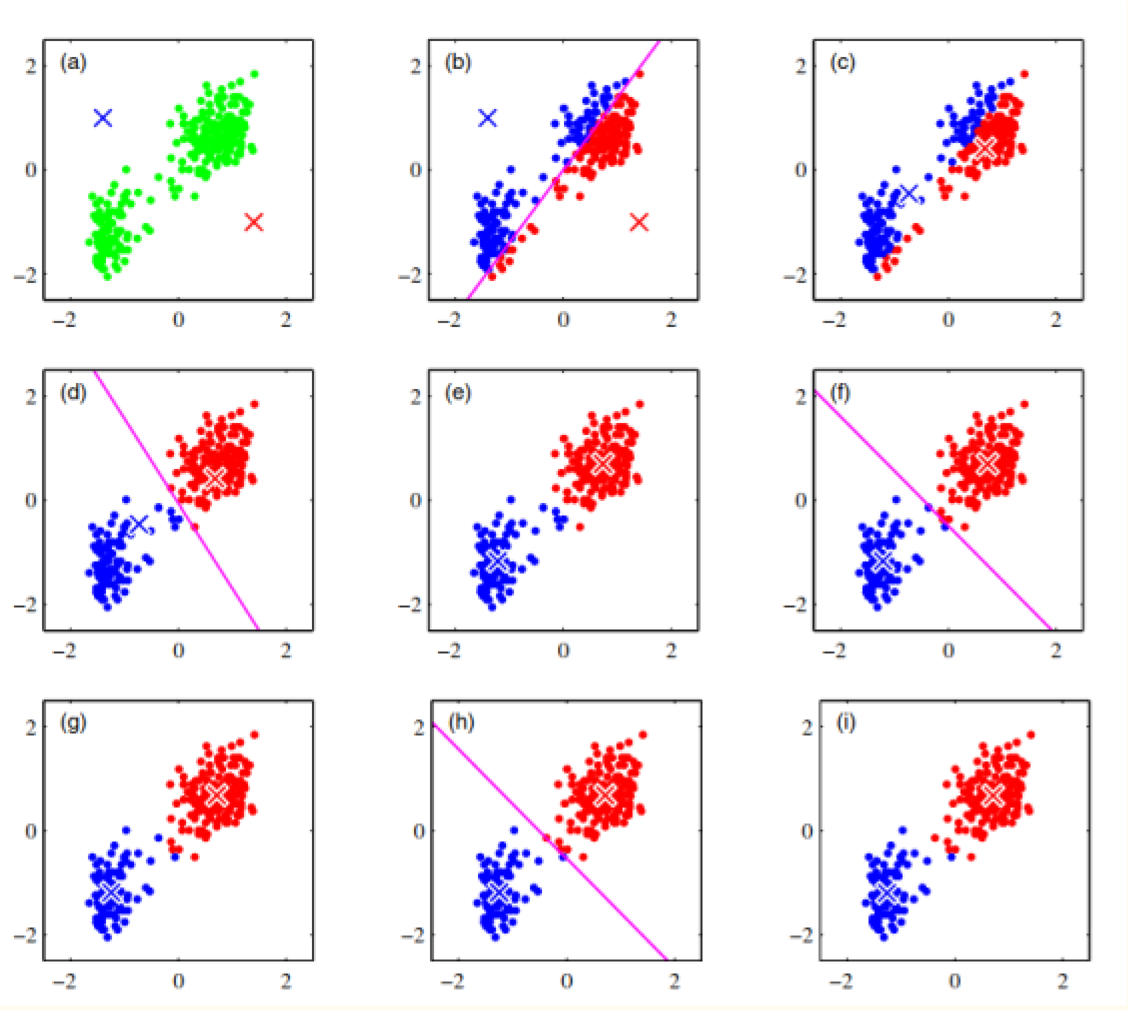
Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

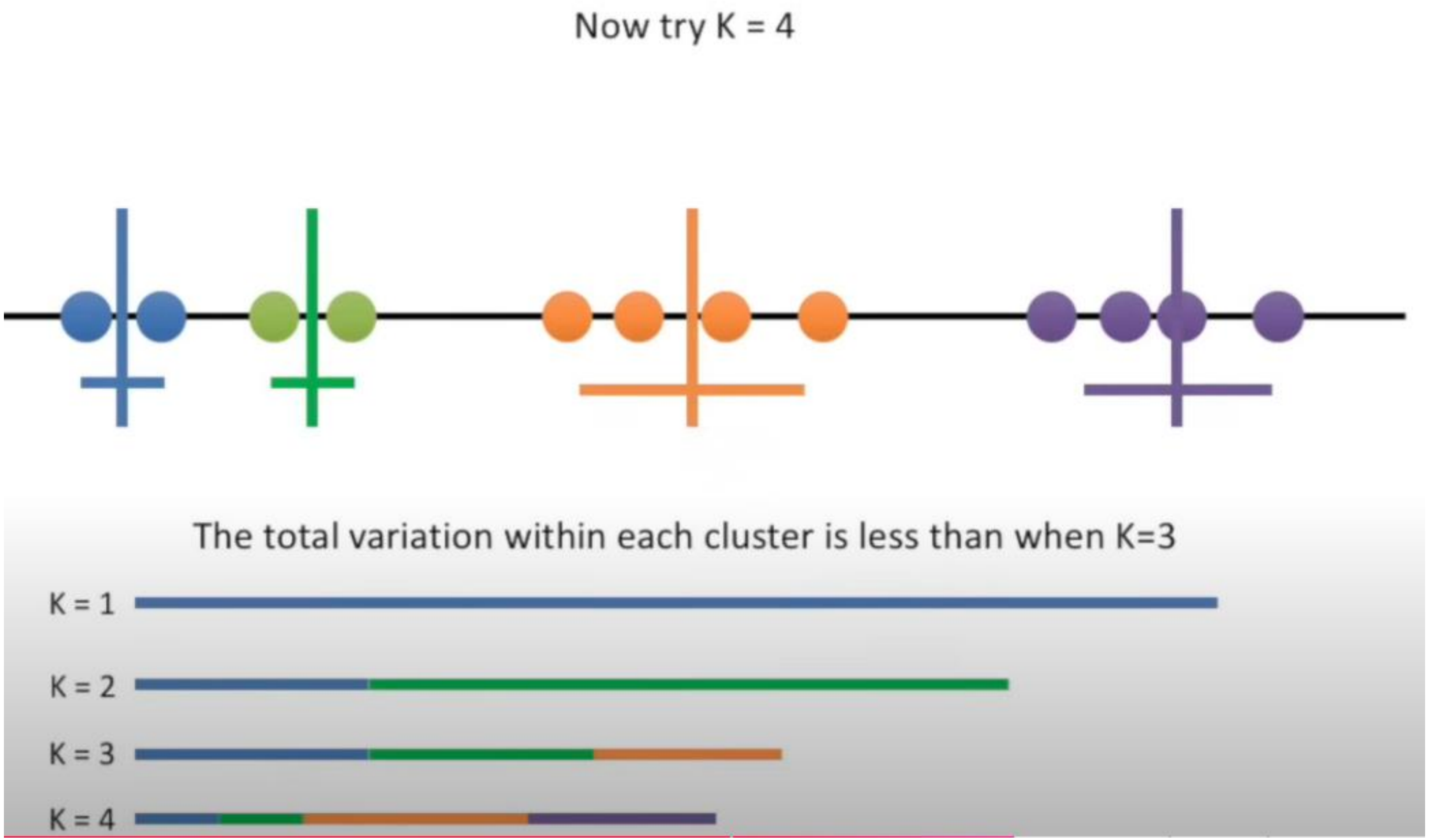
Step-7: The model is ready.


k-Means Clusters
Cluster 1
Cluster 2
Cluster 3
Hierarchical Clusters

k-means



k-means

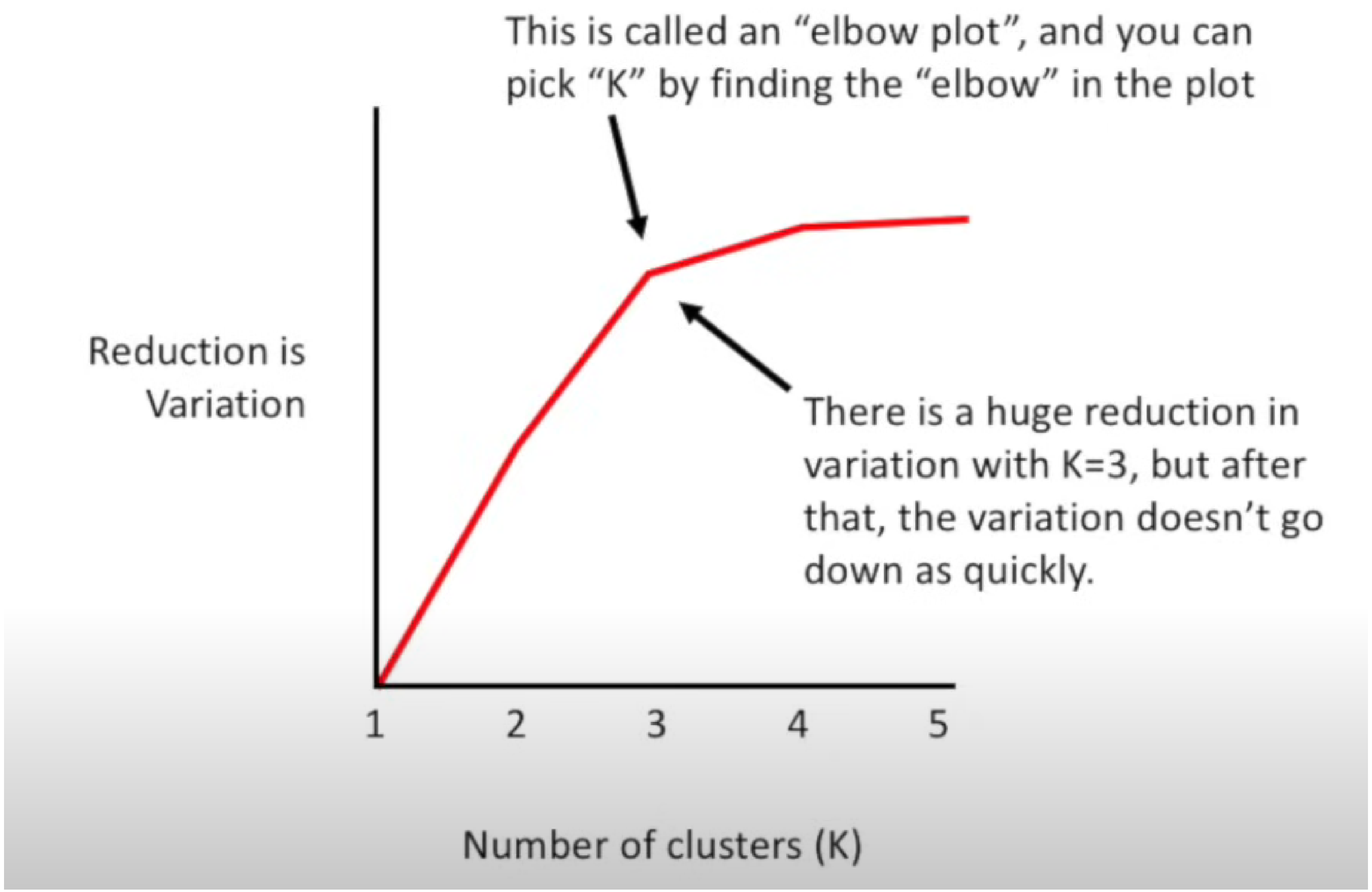


Now try K = 4

The total variation within each cluster is less than when K=3

from : **StatQuest: K-means clustering**

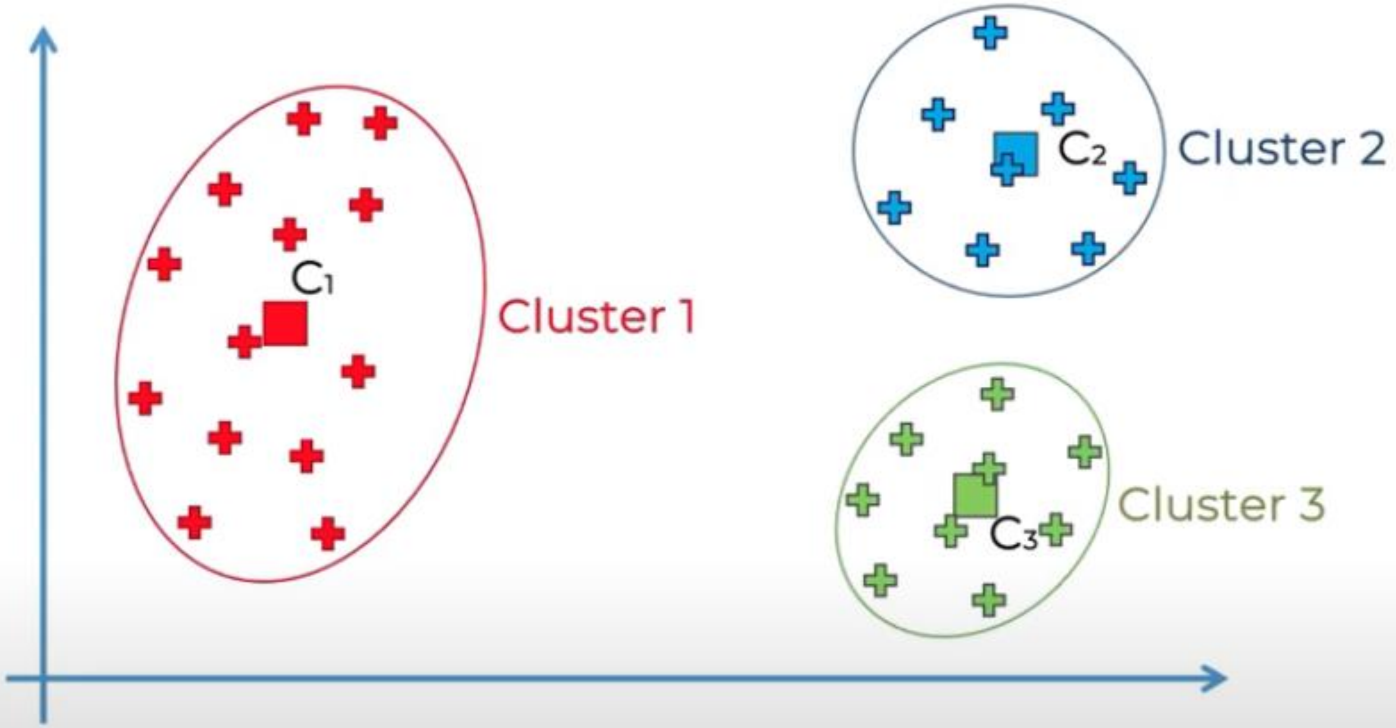
k-means



This is called an "elbow plot", and you can pick "K" by finding the "elbow" in the plot

There is a huge reduction in variation with K=3, but after that, the variation doesn't go down as quickly.

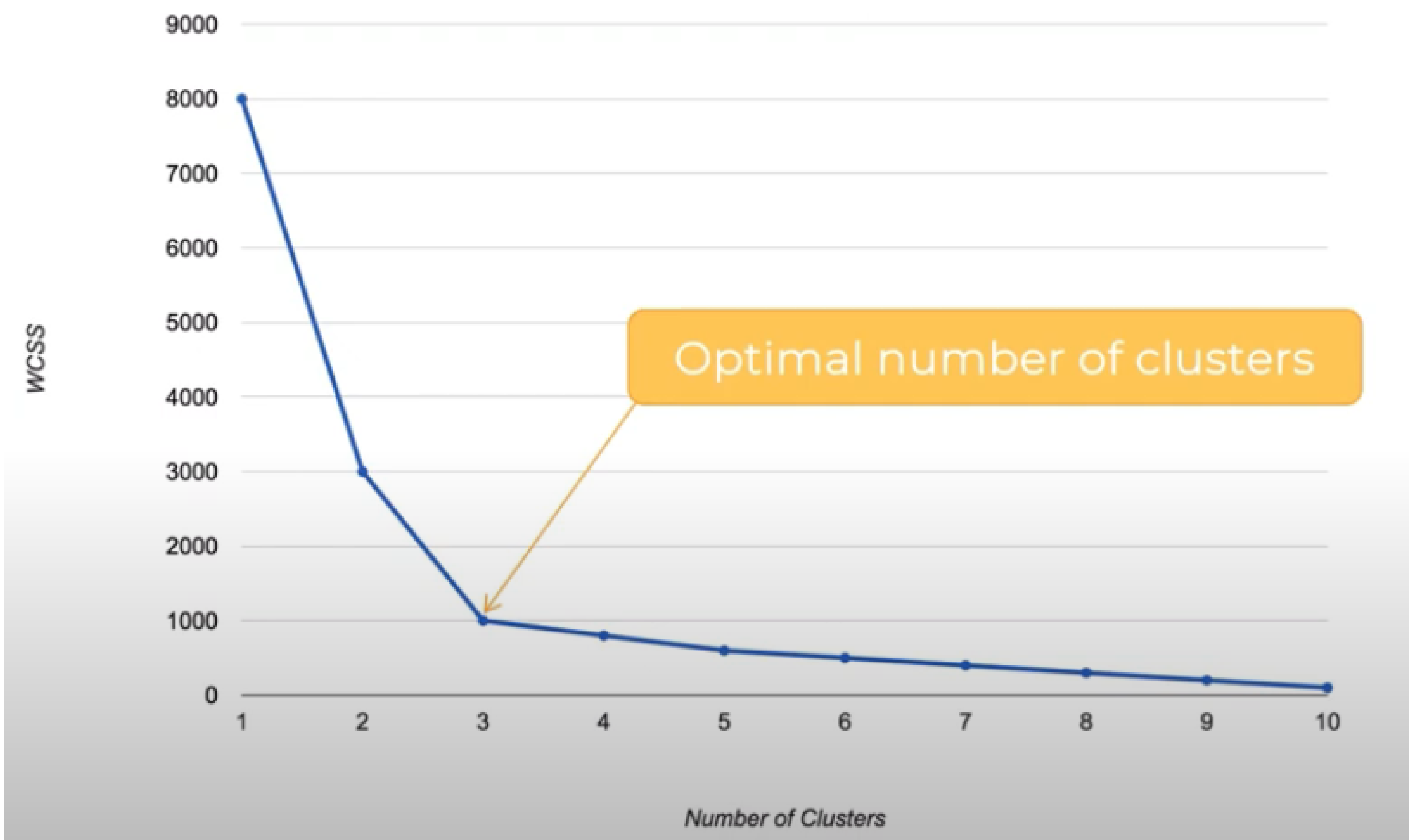
Elbow Method



Within Cluster Sum of Squares:

$$\mathrm{WCSS} = \sum_{\mathrm{P_i\ in\ Cluster\ 1}} \mathrm{distance}(\mathrm{P_i}, \mathrm{C_1})^2 + \sum_{\mathrm{P_i\ in\ Cluster\ 2}} \mathrm{distance}(\mathrm{P_i}, \mathrm{C_2})^2 + \sum_{\mathrm{P_i\ in\ Cluster\ 3}} \mathrm{distance}(\mathrm{P_i}, \mathrm{C_3})^2$$

The Elbow Method



Optimal number of clusters

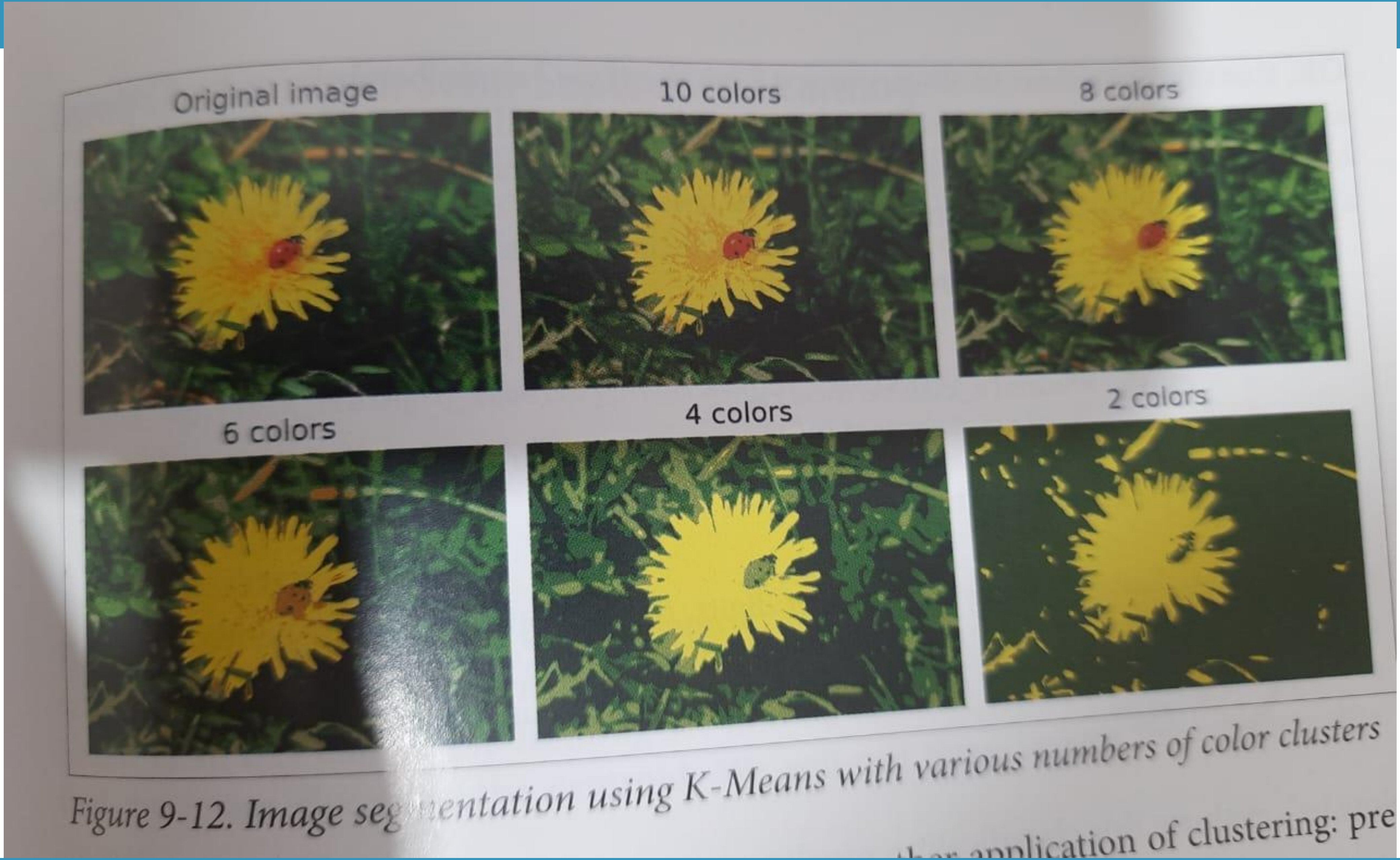
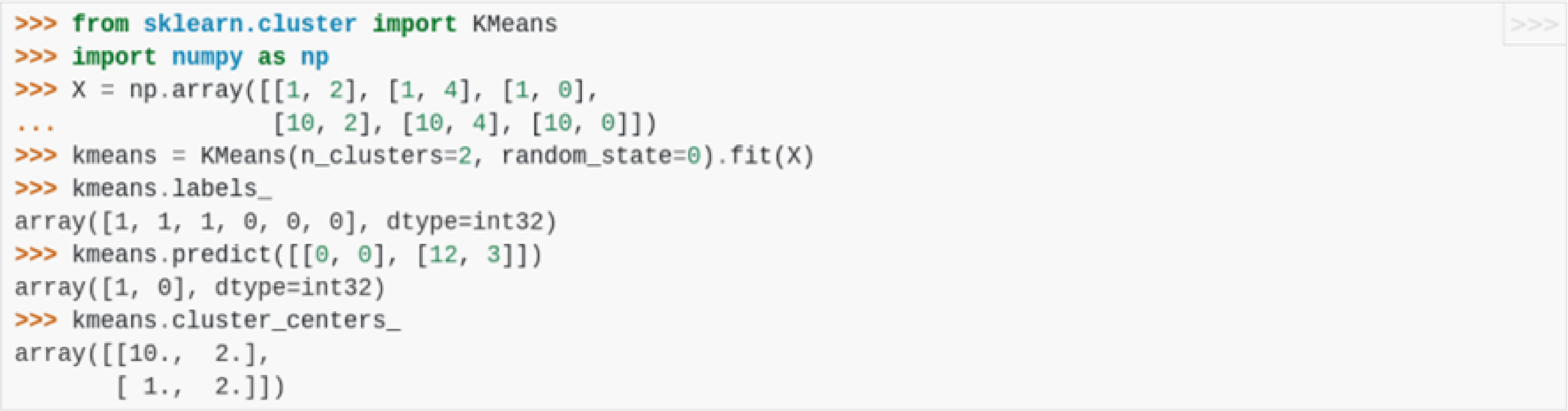


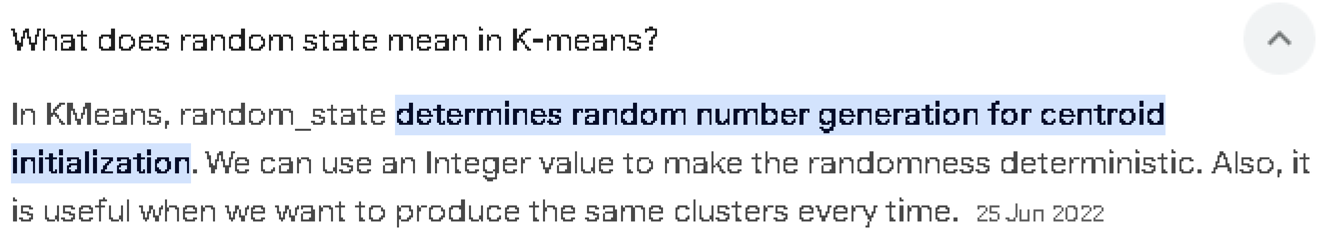
Figure 9-12. Image segmentation using K-Means with various numbers of color clusters

ther application of clustering: pre

k-mean



```
>>> from sklearn.cluster import KMeans
>>> import numpy as np
>>> X = np.array([[1, 2], [1, 4], [1, 0],
...               [10, 2], [10, 4], [10, 0]])
>>> kmeans = KMeans(n_clusters=2, random_state=0).fit(X)
>>> kmeans.labels_
array([1, 1, 1, 0, 0, 0], dtype=int32)
>>> kmeans.predict([[0, 0], [12, 3]])
array([1, 0], dtype=int32)
>>> kmeans.cluster_centers_
array([[10.,  2.],
       [ 1.,  2.]])
```



What does random state mean in K-means?

In KMeans, random_state determines random number generation for centroid initialization. We can use an Integer value to make the randomness deterministic. Also, it is useful when we want to produce the same clusters every time. 25 Jun 2022

k-mean

Key Concepts:

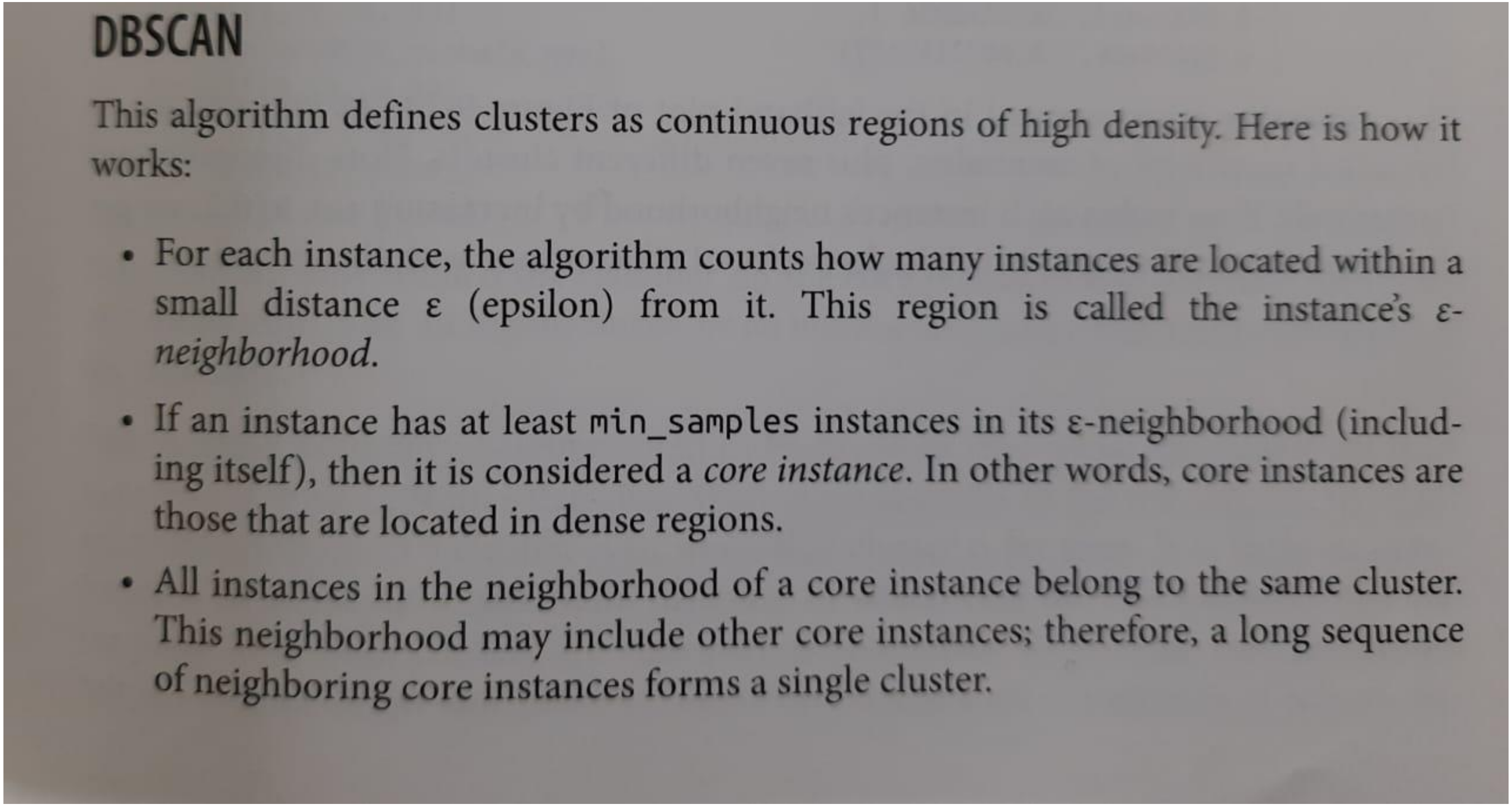
- **Cluster Centroid:** The mean of all data points assigned to a cluster.
- **Euclidean Distance:** The most commonly used distance metric to measure the distance between data points and centroids.
- **Convergence:** The algorithm converges when the cluster assignments stabilize.

Strengths of K-Means:

- **Simple and Efficient:** Relatively easy to implement and computationally efficient.
- **Scalable:** Can handle large datasets.

Weaknesses of K-Means:

- **Sensitive to Initial Centroids:** The initial choice of centroids can significantly impact the final clustering.
- **Assumes Spherical Clusters:** K-Means tends to work best with clusters that are spherical and of similar size.
- **Requires Specifying K:** The number of clusters, K, must be specified beforehand.



DBSCAN

This algorithm defines clusters as continuous regions of high density. Here is how it works:

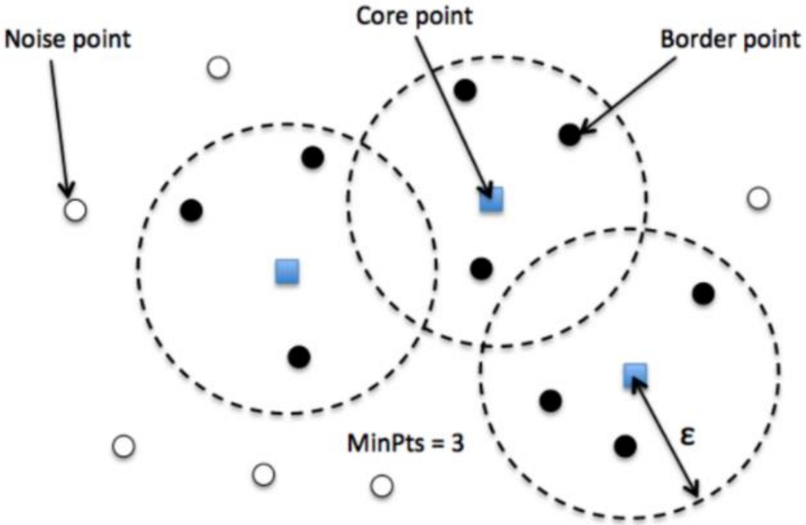
- For each instance, the algorithm counts how many instances are located within a small distance ε (epsilon) from it. This region is called the instance's ε-*neighborhood*.
- If an instance has at least `min_samples` instances in its ε-neighborhood (including itself), then it is considered a *core instance*. In other words, core instances are those that are located in dense regions.
- All instances in the neighborhood of a core instance belong to the same cluster. This neighborhood may include other core instances; therefore, a long sequence of neighboring core instances forms a single cluster.

DBSCAN

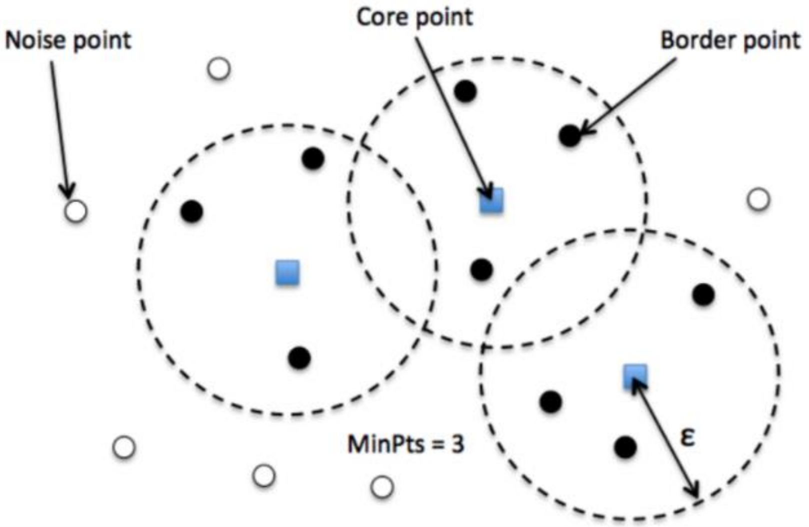
DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a clustering algorithm that groups together points that are closely packed together (high-density regions) and separates low-density regions as noise.

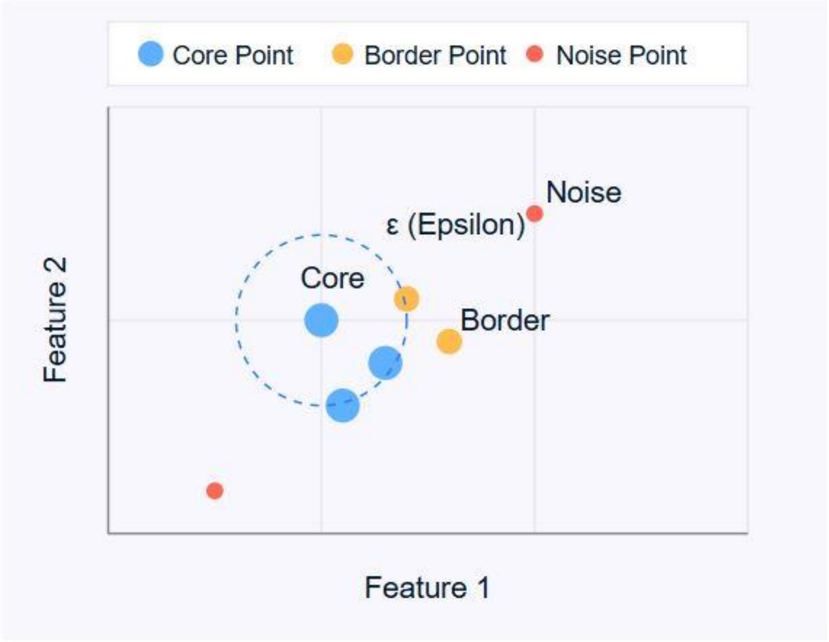
Key Concepts:

- **Core Point:** A point that has at least *MinPts* points within its *Eps* neighborhood.
- **Border Point:** A point that is not a core point but is within the *Eps* neighborhood of a core point.
- **Noise Point:** A point that is neither a core point nor a border point.



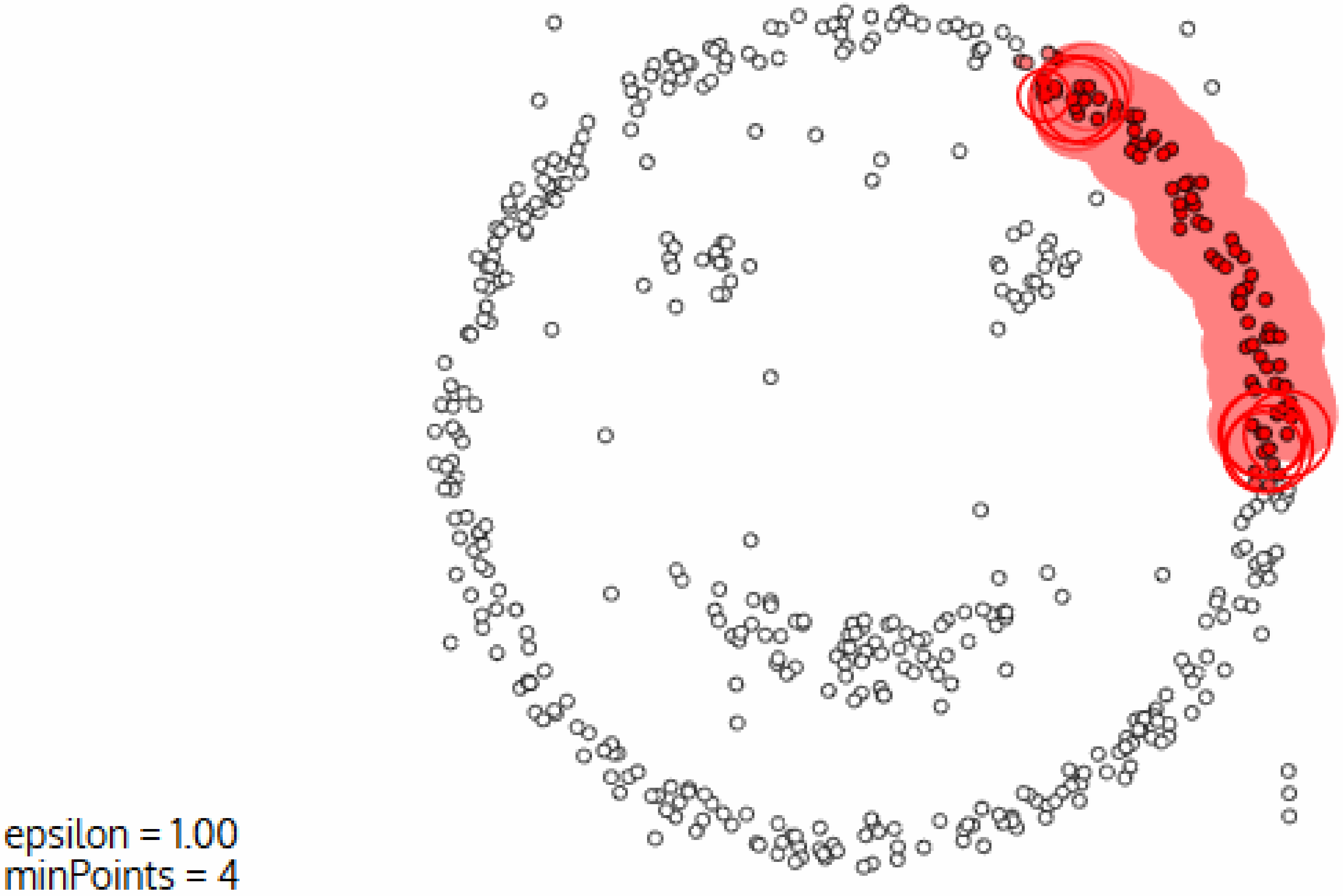
DBSCAN





How DBSCAN Works:

1. **Core Point Identification:**
   - For each point, count the number of points within its *Eps* neighborhood.
   - If the count is greater than or equal to *MinPts*, the point is a core point.
2. **Cluster Formation:**
   - Start with a core point and recursively add its *Eps*-neighborhood points to the cluster.
   - Continue adding points to the cluster until no more core points are found in the neighborhood.
3. **Noise Point Identification:**
   - Any point that is not assigned to a cluster is considered noise.


epsilon = 1.00
minPoints = 4

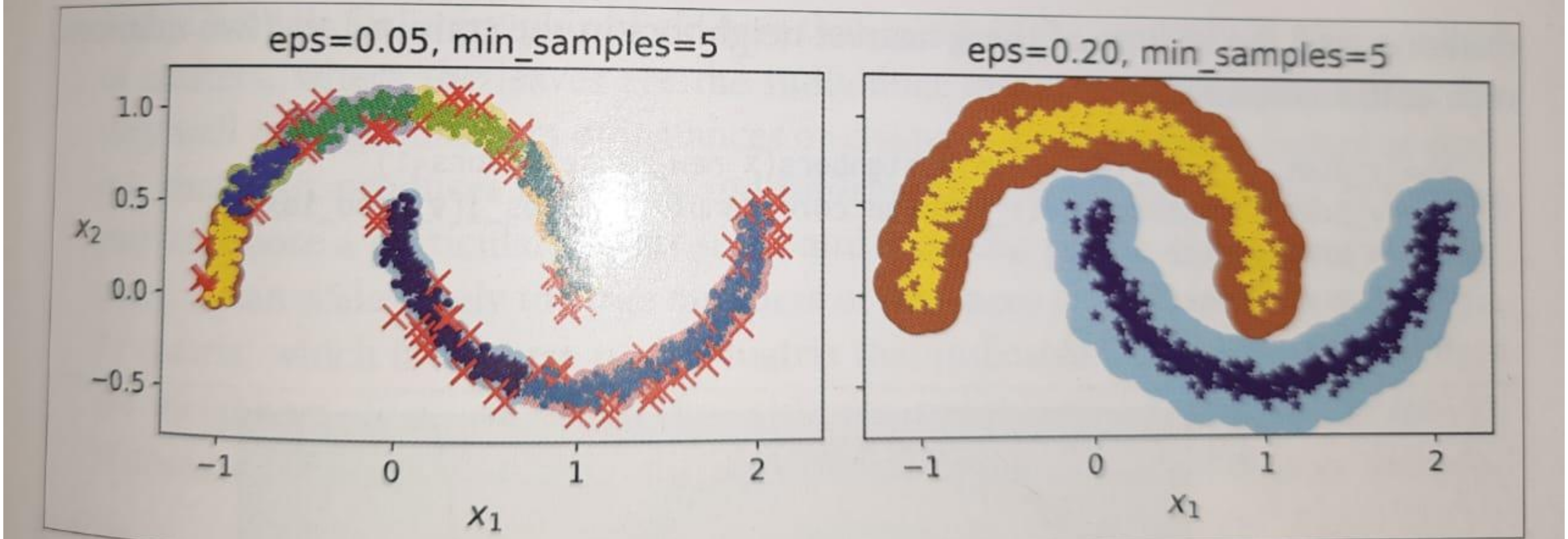


Figure 9-14. DBSCAN clustering using two different neighborhood radiuses

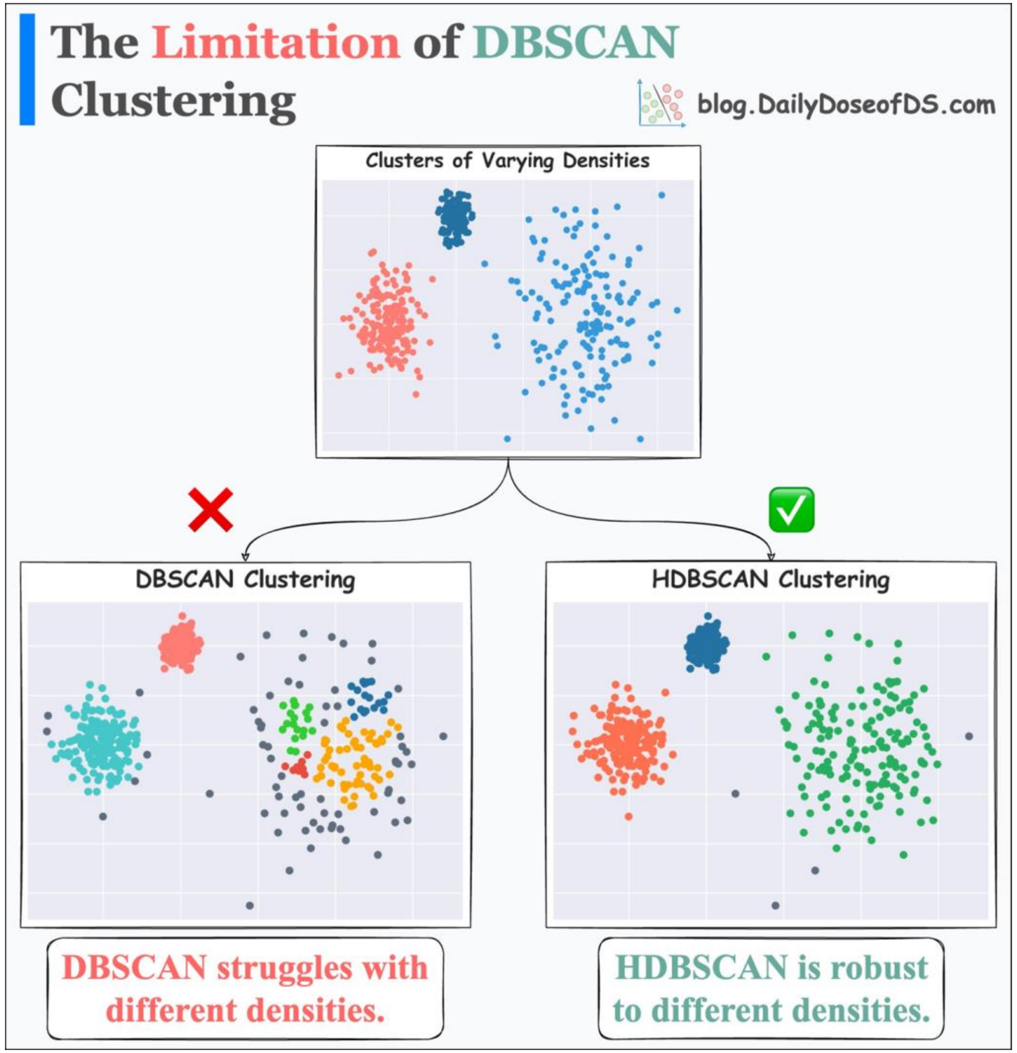
DBSCAN

Strengths of DBSCAN:

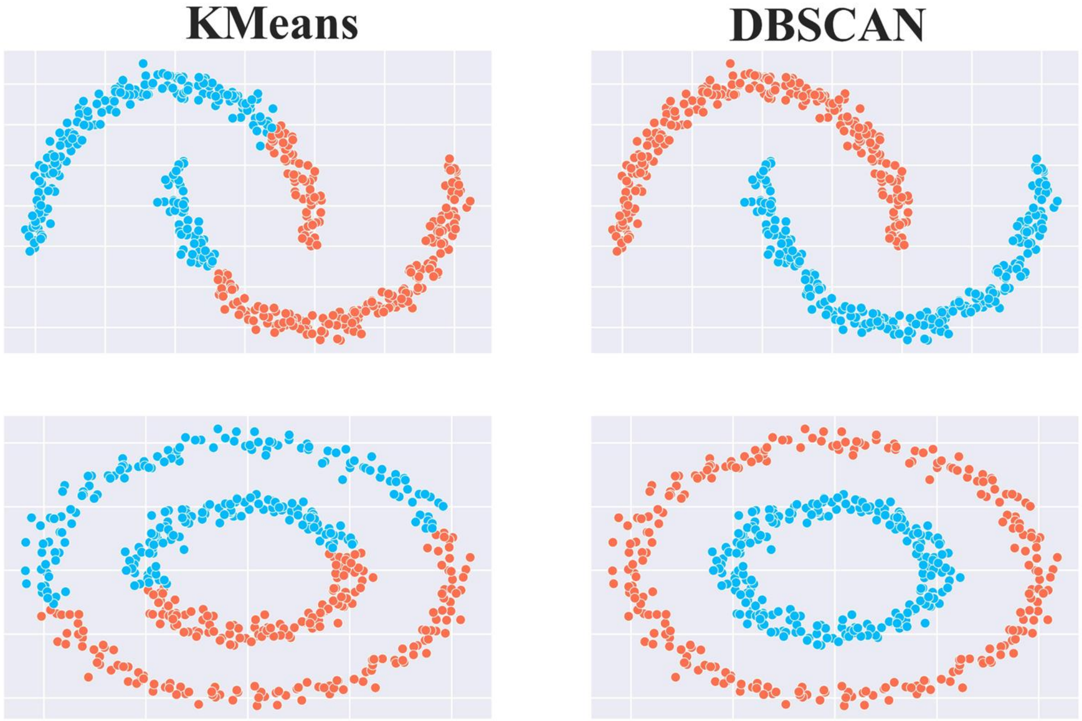
- **Handles Clusters of Arbitrary Shape:** DBSCAN can discover clusters of any shape, unlike K-Means which assumes spherical clusters.
- **Noise Tolerance:** It can effectively identify and handle noise points.
- **Does Not Require Specifying the Number of Clusters:** The number of clusters is determined automatically based on the data density.

Weaknesses of DBSCAN:

- **Sensitivity to Parameters:** The choice of *Eps* and *MinPts* can significantly impact the clustering results.
- **Clustering Density Variations:** DBSCAN may struggle with datasets that have varying densities. (Sol: Hierarchical DBSCAN)



Comparison



Choosing the Right Clustering Approach:

The choice of clustering algorithm depends on various factors:

- **Number of clusters:** Known or unknown.
- **Cluster shape:** Spherical, arbitrary, or overlapping.
- **Presence of noise and outliers:** Can the algorithm handle them?
- **Computational complexity:** How efficient is the algorithm?
- **Interpretability:** How easy is it to understand the results?

Additional Considerations:

- **Distance Metrics:** The choice of distance metric (e.g., Euclidean, Manhattan, cosine similarity) can significantly impact clustering results.
- **Feature Scaling:** Normalizing or standardizing features can improve clustering performance.

Relative Merits of Different Clustering Methods

Hierarchical Clustering

Strengths:

- Does not require specifying the number of clusters beforehand.
- Can handle non-spherical clusters.
- Provides a hierarchical structure that can be useful for visualization and interpretation.

Weaknesses:

- Sensitive to noise and outliers.
- Computational complexity can be high for large datasets.

Relative Merits of Different Clustering Methods

Partitional Clustering

K-Means

- **Strengths:**
  - Relatively simple and efficient.
  - Can handle large datasets.
- **Weaknesses:**
  - Requires specifying the number of clusters beforehand.
  - Sensitive to initial cluster centroids.
  - Assumes spherical clusters.

DBSCAN

- **Strengths:**
  - Can discover clusters of arbitrary shape.
  - Can handle noise and outliers effectively.
  - Does not require specifying the number of clusters.
- **Weaknesses:**
  - Sensitive to parameter settings (epsilon and minPts).
  - Can be computationally expensive for large datasets.

Relative Merits of Different Clustering Methods

Fuzzy Clustering

- **Strengths:**
  - Can handle overlapping clusters.
  - Provides a more realistic representation of data.
- **Weaknesses:**
  - Can be computationally expensive.
  - Interpretation of results can be challenging.

Model-Based Clustering (Gaussian Mixture Models)

- **Strengths:**
  - Can model complex data distributions.
  - Can handle overlapping clusters.
  - Can estimate the number of clusters automatically.
- **Weaknesses:**
  - Can be sensitive to initialization.
  - Computational complexity can be high.

Clustering Tendency

Clustering tendency refers to the inherent structure or pattern within a dataset that makes it suitable for clustering analysis. In simpler terms, it's the degree to which a dataset can be meaningfully divided into groups.

Why is it important?

- **Avoids Futile Clustering:** If a dataset lacks a clear underlying structure, applying clustering algorithms might lead to meaningless or arbitrary groupings.
- **Optimizes Algorithm Selection:** Different clustering algorithms are suited for different data distributions. Assessing clustering tendency helps select the most appropriate algorithm.

How to Assess Clustering Tendency: Several methods can be used to assess clustering tendency:

1. **Visual Inspection: (Scatter Plots, Density Plots)**
2. **Statistical Tests: (Hopkins Statistic)**
3. **Model-Based Approaches:**

Specific applications in Data Mining

Customer Segmentation:

- Grouping customers based on demographics, purchase history, and behavior to tailor marketing strategies.
- Identifying high-value customers for targeted promotions.

Anomaly Detection:

- Detecting unusual patterns in data, such as fraudulent transactions or network intrusions.
- Identifying outliers in sensor data to predict equipment failures.

Document Clustering:

- Grouping similar documents based on content, topic, or style.
- Organizing large document collections for efficient search and retrieval.

Specific applications in Network protocols

Network Protocols

Network Traffic Analysis:

- Identifying abnormal traffic patterns that may indicate attacks or security breaches.
- Classifying network traffic into different categories (e.g., web traffic, email, file transfer).
- Optimizing network performance by identifying bottlenecks and congestion points.

Protocol Engineering:

- Designing efficient and reliable network protocols.
- Analyzing protocol performance and identifying areas for improvement.

Specific applications in Web Traffic Analysis

Web Traffic Analysis

User Segmentation:

- Grouping users based on their browsing behavior, demographics, and interests.
- Personalizing web content and recommendations.

Web Log Analysis:

- Analyzing web server logs to identify popular pages, user trends, and potential security threats.
- Optimizing website performance by identifying slow-loading pages.

Clickstream Analysis:

- Analyzing user interactions with a website to understand user behavior and preferences.
- Improving website design and navigation.

Question: You are tasked with designing a data science solution to analyse the placement data of students from your college's placement cell. Assume some data and explain(7 Marks)

Ans: Designing a Data Science Solution for College Placement Data Analysis

Understanding the Data

Hypothetical Data Structure:

Let's assume we have a dataset with the following columns: - >

Column Name	Description
Student ID	Unique identifier for each student
Name	Student's name
Branch	Student's academic branch (e.g., CSE, ECE, ME)
Year of Passing	Year of graduation
CGPA	Cumulative Grade Point Average
10th Percentage	10th standard percentage
12th Percentage	12th standard percentage

Data Analysis Goals

1. **Identify Key Factors Influencing Placements:**
   - Determine the correlation between academic performance (CGPA, 10th, 12th percentages), coding proficiency, projects, internships, certifications, and placement outcomes.
   - Analyze the impact of different branches on placement rates and average salary packages.
2. **Predict Placement Outcomes:**
   - Build predictive models to forecast the likelihood of a student getting placed based on their profile.
   - Identify students who may need additional support or guidance to improve their placement chances.
3. **Optimize Placement Strategies:**
   - Analyze historical placement data to identify trends and patterns.
   - Identify areas where the college can improve its placement efforts, such as focusing on specific skills or industries.
   - Provide insights to the placement cell to refine their strategies and improve student outcomes.

Data Science Methodology

1. **Data Cleaning and Preprocessing:**
   - Handle missing values (e.g., imputation, deletion).
   - Convert categorical variables into numerical format (e.g., one-hot encoding).
   - Normalize numerical features to a common scale.
2. **Exploratory Data Analysis (EDA):**
   - Visualize the distribution of numerical variables (e.g., histograms, box plots).
   - Analyze the relationship between categorical variables and placement outcomes (e.g., bar charts, contingency tables).
   - Identify outliers and anomalies in the data.
3. **Feature Engineering:**
   - Create new features from existing ones (e.g., combine 10th and 12th percentages, calculate a combined academic score).
   - Consider feature selection techniques to reduce dimensionality and improve model performance.
4. **Model Building and Evaluation:**
   - **Classification Models:**
     - Logistic Regression
     - Decision Trees
     - Random Forest
     - Support Vector Machines (SVM)
     - XGBoost
   - **Regression Models:**
     - Linear Regression
     - Decision Tree Regression
     - Random Forest Regression
     - XGBoost Regression
   - Evaluate model performance using metrics like accuracy, precision, recall, F1-score, and mean squared error.
5. **Model Deployment and Monitoring:**
   - Deploy the best-performing model into a production environment (e.g., web application, API).
   - Continuously monitor the model's performance and retrain it as needed to maintain accuracy.

By following this data science approach, the college's placement cell can gain valuable insights into student profiles, identify areas for improvement, and make data-driven decisions to enhance placement outcomes.

Question : Assume an asteroid is expected to hit Gandhinagar. As a data scientist what would be your role.

Here's how a data scientist could contribute in the event of an asteroid expected to hit Gandhinagar.:

1. Impact Zone Modeling:

- **Data Analysis:** Analyze data on the asteroid's size, trajectory, and speed to predict the likely impact zone and its severity.
- **Simulation:** Create simulations to model the asteroid's path and potential damage, considering factors like terrain, population density, and infrastructure.
- **Visualization:** Develop maps and visualizations to communicate the predicted impact zone and its potential consequences to decision-makers and the public.

2. Evacuation Planning:

- **Population Data Analysis:** Analyze population data to identify areas at highest risk and optimize evacuation routes.
- **Real-time Tracking:** Develop tools to track the asteroid's progress and update evacuation plans in real-time.
- **Resource Allocation:** Use data to optimize the allocation of resources like emergency services, shelters, and supplies based on predicted impact zones.

3. Damage Assessment:

- **Satellite Imagery Analysis:** Analyze satellite images to assess pre-impact conditions and post-impact damage.
- **Damage Modeling:** Use machine learning models to estimate the extent of damage to infrastructure, buildings, and the environment.
- **Risk Assessment:** Identify areas with the highest risk of secondary hazards like fires, tsunamis, or landslides.

4. Post-Impact Recovery:

- **Damage Assessment:** Analyze data to assess the extent of damage and prioritize recovery efforts.
- **Resource Allocation:** Use data to optimize the allocation of resources for reconstruction and rehabilitation.
- **Long-term Planning:** Develop data-driven strategies for long-term recovery and resilience planning.

5. Public Communication:

- **Data Visualization:** Create clear and concise visualizations to communicate complex information about the asteroid's threat and the impact zone to the public.
- **Risk Communication:** Develop strategies to communicate risks and uncertainties effectively, avoiding panic while ensuring preparedness.

Tools and Techniques:

- **Machine Learning:** Use algorithms to predict impact zones, simulate damage, and analyze satellite imagery.
- **Geographic Information Systems (GIS):** Utilize GIS to create maps and visualizations of the impact zone, evacuation routes, and damage assessments.
- **Data Mining and Big Data:** Analyze large datasets from various sources to extract valuable insights.
- **Statistical Modeling:** Use statistical models to quantify uncertainties and risks associated with the asteroid impact.

By leveraging these skills and tools, data scientists can play a crucial role in mitigating the impact of the asteroid and supporting the recovery efforts in Gandhinagar.