Parul® University

NAAC A++

Unit 4:Unsupervised Learning

Dr. Vinod Patidar
Associate Professor
Computer Science and Engineering

INDEX

Content

- ❑ Introduction to Unsupervised Learning
- ❑ Clustering Basics
- ❑ Types of Clustering
- ❑ K-Means Clustering
- ❑ K-Mode Clustering
- ❑ Self-Organizing Maps (SOM)
- ❑ Expectation-Maximization (EM)
- ❑ Principal Component Analysis (PCA)

Introduction to Unsupervised Learning

- Unsupervised learning is a type of machine learning in which models are trained using unlabeled dataset and are allowed to act on that data without any supervision.
- Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data.
- In unsupervised learning, we don't have a predetermined result. The machine tries to find useful insights from the huge amount of data.
- The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format.

Introduction

- Example: Suppose the unsupervised learning algorithm is given an input dataset containing images of different types of cats and dogs.

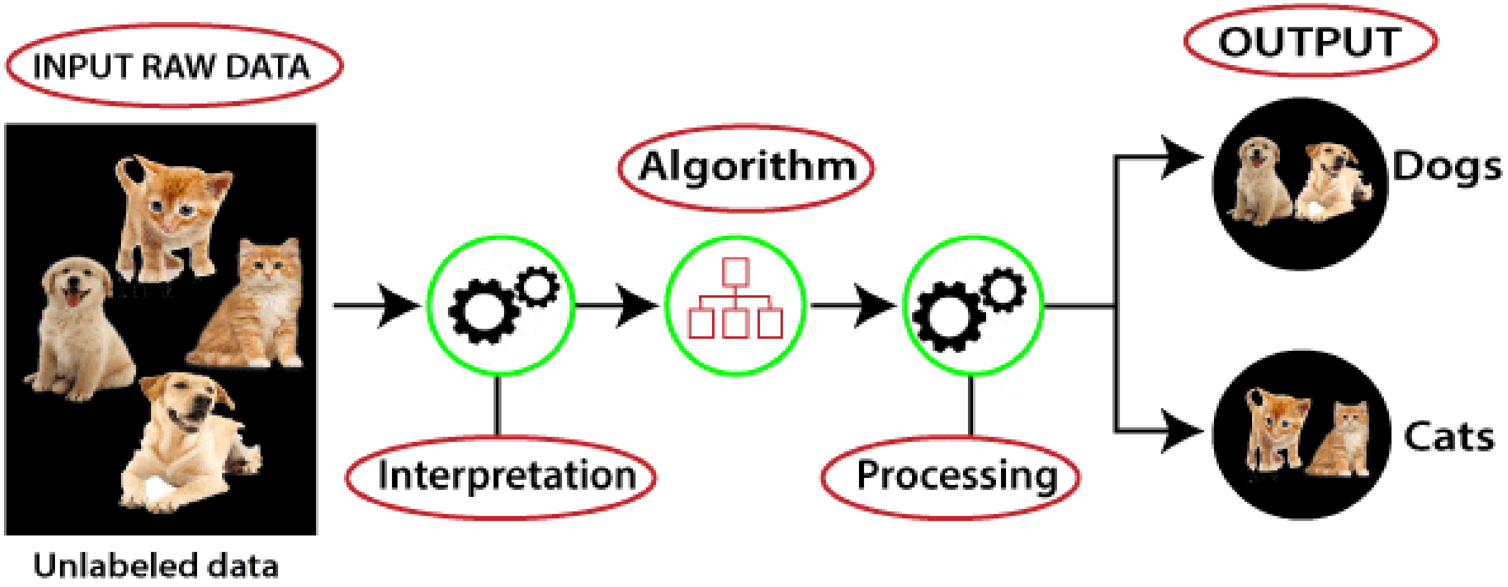
- The algorithm is never trained upon the given dataset, which means it does not have any idea about the features of the dataset.
- The task of the unsupervised learning algorithm is to identify the image features on their own. Unsupervised learning algorithm will perform this task by clustering the image dataset into the groups according to similarities between images.

Unsupervised Learning

Below are some main reasons which describe the importance of Unsupervised Learning:

- Unsupervised learning is helpful for finding useful insights from the data.
- Unsupervised learning is much similar as a human learns to think by their own experiences, which makes it closer to the real AI.
- Unsupervised learning works on unlabeled and uncategorized data which make unsupervised learning more important.
- In real-world, we do not always have input data with the corresponding output so to solve such cases, we need unsupervised learning.

Working of unsupervised learning can be understood by the below diagram:



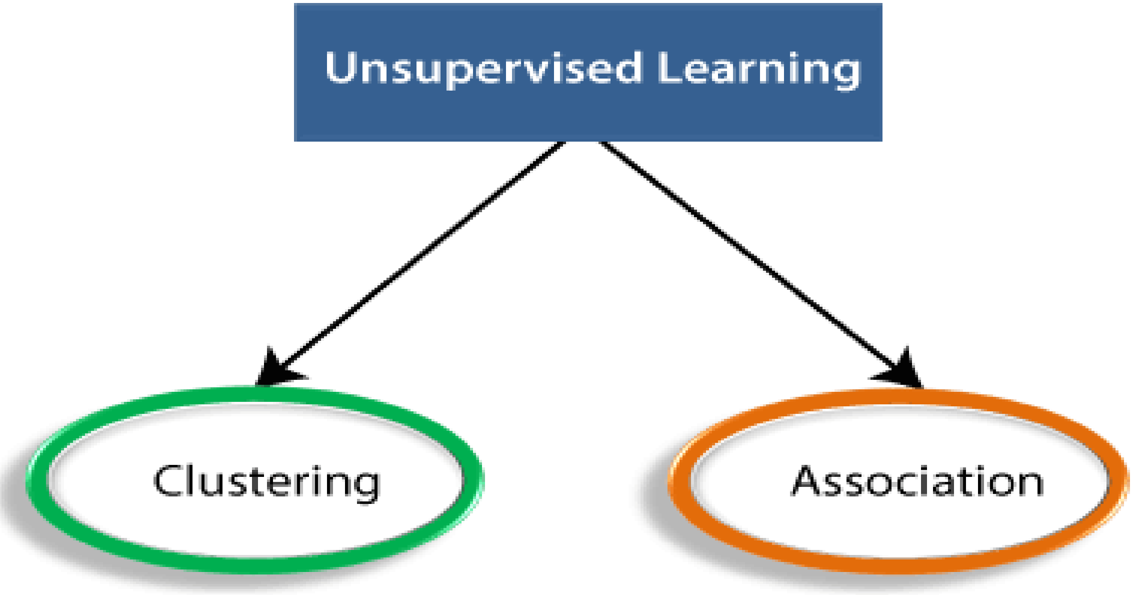
Working of Unsupervised Learning

- Here, we have taken an unlabeled input data, which means it is not categorized and corresponding outputs are also not given. Now, this unlabeled input data is fed to the machine learning model in order to train it. Firstly, it will interpret the raw data to find the hidden patterns from the data and then will apply suitable algorithms such as k-means clustering, Decision tree, etc.

- Once it applies the suitable algorithm, the algorithm divides the data objects into groups according to the similarities and difference between the objects.

Types of Unsupervised Learning Algorithm:

The unsupervised learning algorithm can be further categorized into two types of problems:



Types of Unsupervised Learning Algorithm:

- ❑ **Clustering:** Clustering is a method of grouping the objects into clusters such that objects with most similarities remains into a group and has less or no similarities with the objects of another group. Cluster analysis finds the commonalities between the data objects and categorizes them as per the presence and absence of those commonalities.

- ❑ **Association:** An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database. It determines the set of items that occurs together in the dataset. Association rule makes marketing strategy more effective. Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item. A typical example of Association rule is Market Basket Analysis.

Unsupervised Learning algorithms:

Below is the list of some popular unsupervised learning algorithms:

- K-means clustering
- KNN (k-nearest neighbors)
- Hierarchal clustering
- Anomaly detection
- Neural Networks
- Principle Component Analysis
- Independent Component Analysis
- Apriori algorithm

Unsupervised Learning algorithms:

Advantages of Unsupervised Learning

- Requires less manual data preparation (i.e., no hand labeling) than supervised machine learning.
- Capable of finding previously unknown patterns in data, which is impossible with supervised machine learning models.

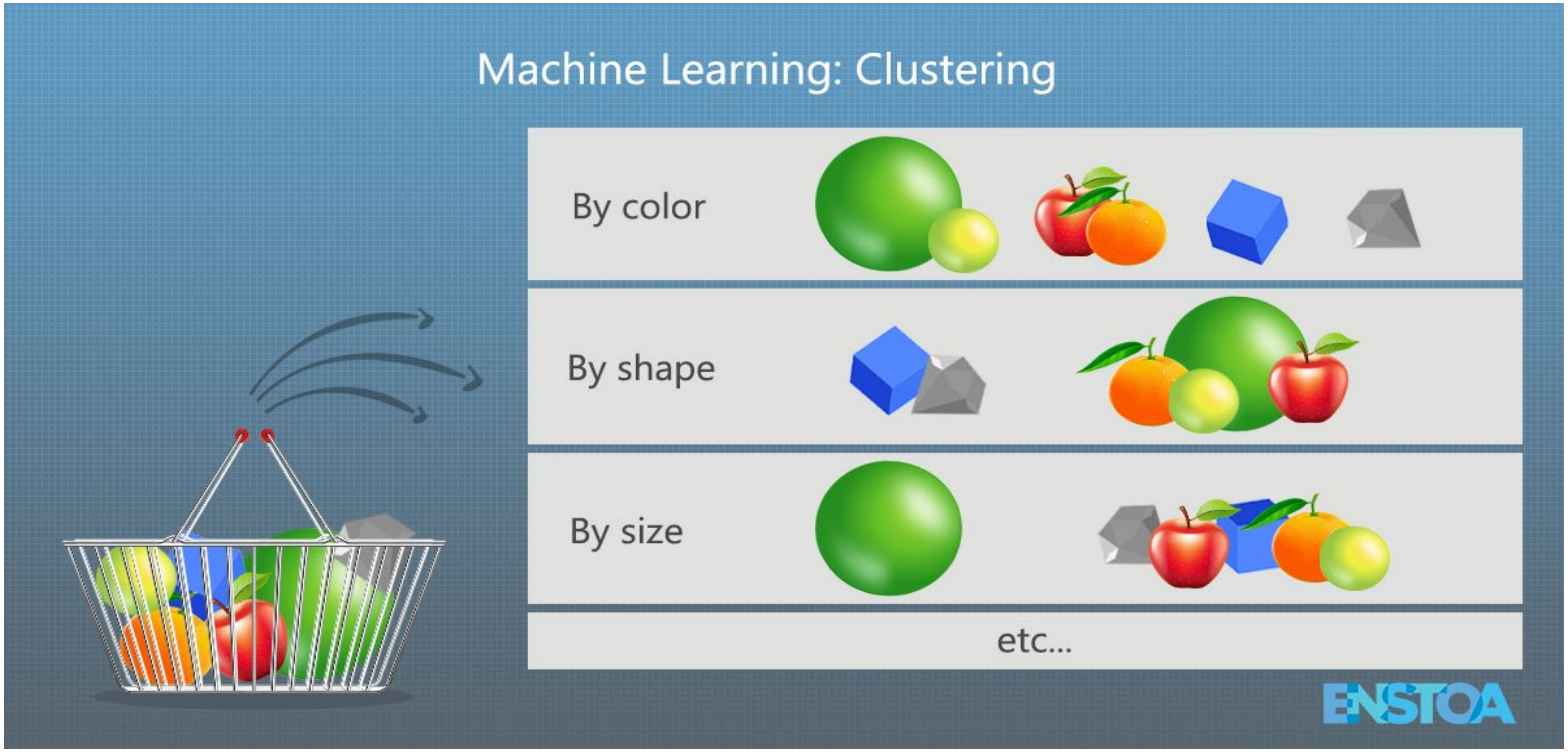
Disadvantages of Unsupervised Learning

- Unsupervised learning is intrinsically more difficult than supervised learning as it does not have corresponding output.
- The result of the unsupervised learning algorithm might be less accurate as input data is not labeled, and algorithms do not know the exact output in advance.

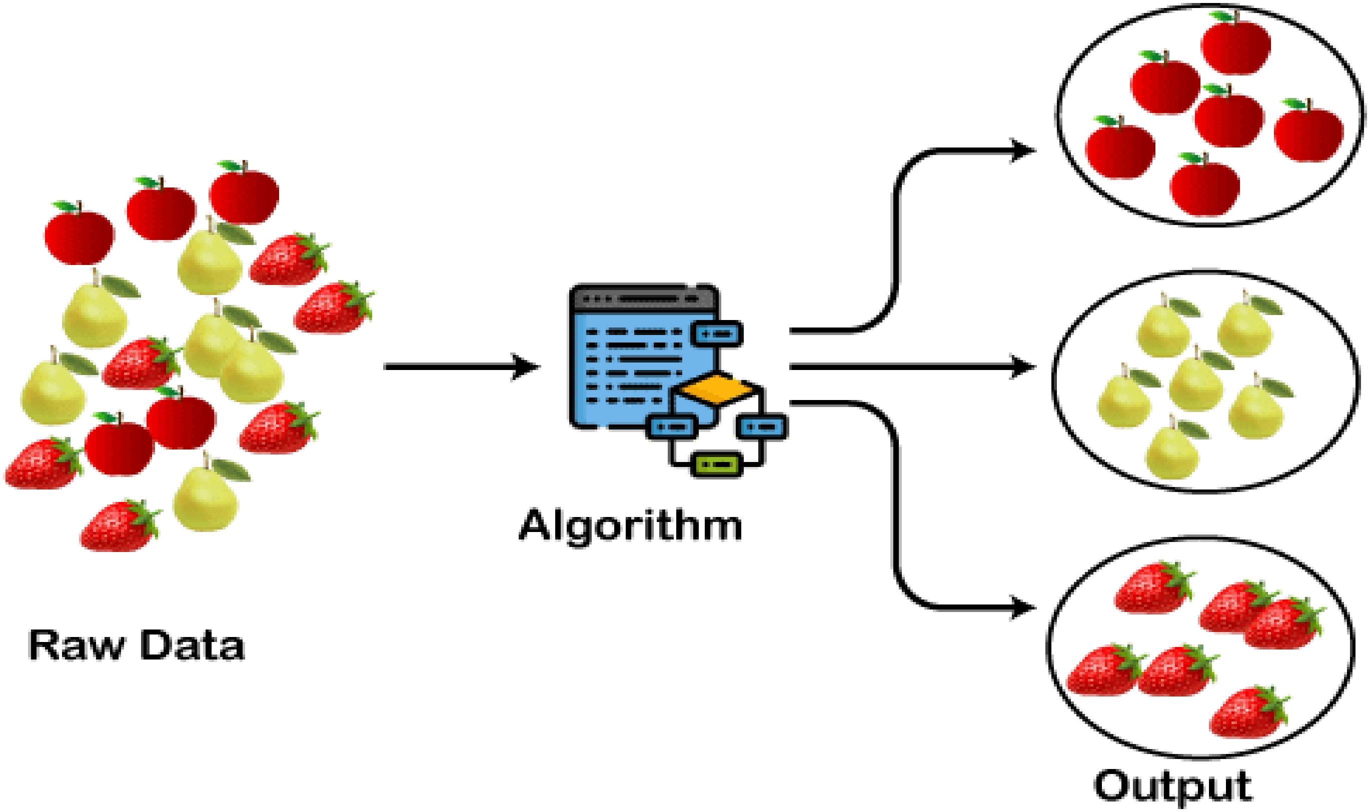
Clustering in Machine Learning :

- Clustering or cluster analysis is a machine learning technique, which groups the unlabeled dataset.
- It can be defined as "A way of grouping the data points into different clusters, consisting of similar data points. The objects with the possible similarities remain in a group that has less or no similarities with another group.
- It does it by finding some similar patterns in the unlabeled dataset such as shape, size, color, behavior, etc., and divides them as per the presence and absence of those similar patterns.
- It is an unsupervised learning method, hence no supervision is provided to the algorithm, and it deals with the unlabeled dataset.
- After applying this clustering technique, each cluster or group is provided with a cluster-ID. ML system can use this id to simplify the processing of large and complex datasets.

Clustering in Machine Learning :



Clustering in Machine Learning :



Clustering in Machine Learning :

Example:

Let's understand the clustering technique with the real-world example of Mall: When we visit any shopping mall, we can observe that the things with similar usage are grouped together. Such as the t-shirts are grouped in one section, and trousers are at other sections, similarly, at vegetable sections, apples, bananas, Mangoes, etc., are grouped in separate sections, so that we can easily find out the things. The clustering technique also works in the same way.

Clustering in Machine Learning :

The clustering technique can be widely used in various tasks. Some most common uses of this technique are:

- Market Segmentation
- Statistical data analysis
- Social network analysis
- Image segmentation
- Anomaly detection, etc.

- Apart from these general usages, it is used by the Amazon in its recommendation system to provide the recommendations as per the past search of products.
- Netflix also uses this technique to recommend the movies and web-series to its users as per the watch history.

Types of Clustering:

Broadly speaking, clustering can be divided into two subgroups:

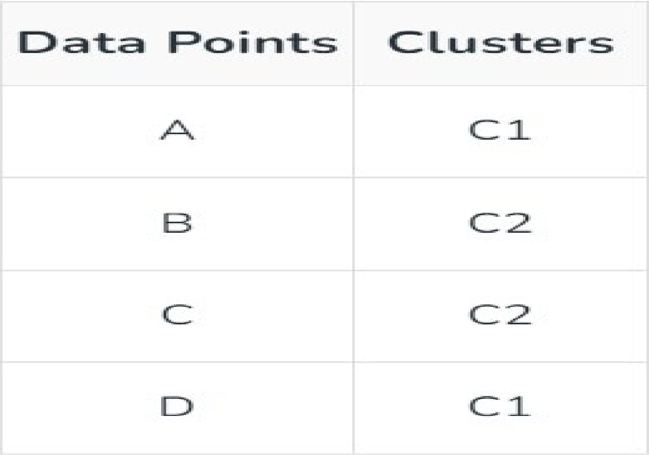
- ❑ **Hard Clustering:** In this, each input data point either belongs to a cluster completely or not.

- ❑ **Soft Clustering:** In this, instead of putting each input data point into a separate cluster, a probability or likelihood of that data point being in those clusters is assigned.

Types of Clustering:

Hard Clustering:

- In this type of clustering, each data point belongs to a cluster completely or not. For example, Let’s say there are 4 data point and we have to cluster them into 2 clusters. So each data point will either belong to cluster 1 or cluster 2.

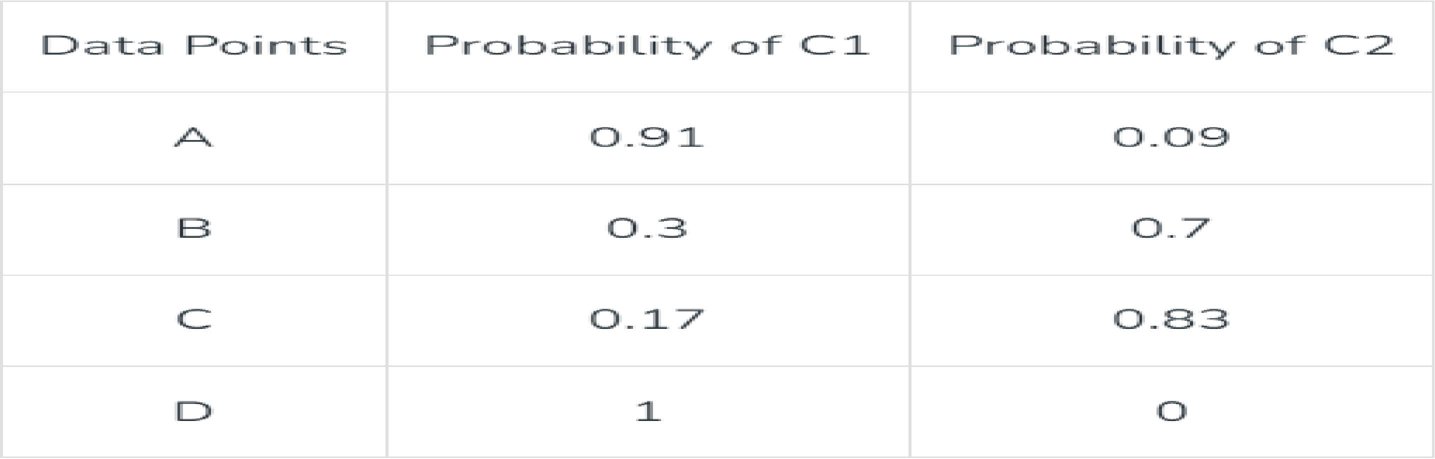


Data Points	Clusters
A	C1
B	C2
C	C2
D	C1

Types of Clustering:

Soft Clustering:

- In this, instead of putting each input data point into a separate cluster, a probability or likelihood of that data point being in those clusters is assigned.
- **For example,** Let’s say there are 4 data point and we have to cluster them into 2 clusters. So we will be evaluating a probability of a data point belonging to both clusters. This probability is calculated for all data points.



Data Points	Probability of C1	Probability of C2
A	0.91	0.09
B	0.3	0.7
C	0.17	0.83
D	1	0

Types of Clustering Methods:

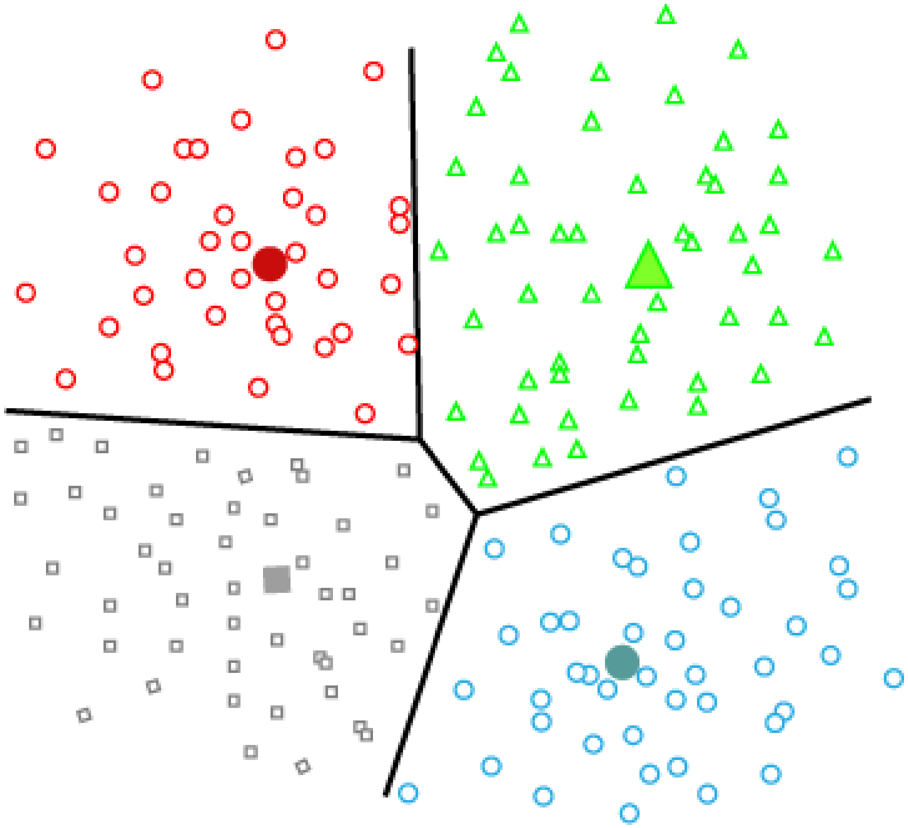
Below are the main clustering methods used in Machine learning:

□ Partitioning Clustering
□ Density-Based Clustering
□ Distribution Model-Based Clustering
□ Hierarchical Clustering
□ Fuzzy Clustering

Partitioning Clustering:

- Partitioned clustering is a type of clustering algorithm that aims to partition the dataset into a predefined number of clusters (K).
- It is also known as the centroid-based method.
- The algorithm iteratively assigns each data point to one of the K clusters, seeking to minimize the dissimilarity (distance) between the data points within each cluster.
- The cluster center is created in such a way that the distance between the data points of one cluster is minimum as compared to another cluster centroid.
- The most common example of partitioning clustering is the K-Means Clustering algorithm.

Partitioning Clustering:



Partitioning Clustering:

How it Works:

- Initialization: Randomly select K initial cluster centroids.
- Assignment: Assign each data point to the nearest centroid, forming K clusters.
- Update: Recalculate the centroids as the mean of all data points in each cluster.
- Repeat: Iterate the assignment and update steps until convergence or a predetermined number of iteration

Density-Based Clustering:

- Density-based clustering is a type of clustering algorithm that aims to discover clusters based on the density of data points in a given region.
- The key idea behind density-based clustering is to identify dense regions in the data space and consider them as clusters.
- A dense region is defined as an area with a sufficient number of data points.
- Data points that fall within a dense region are considered core points, while those in less dense regions are classified as boundary points.
- Noise points are data points that do not belong to any cluster and are located in low-density regions.
- One of the most popular density-based clustering algorithms is DBSCAN (Density-Based Spatial Clustering of Applications with Noise).

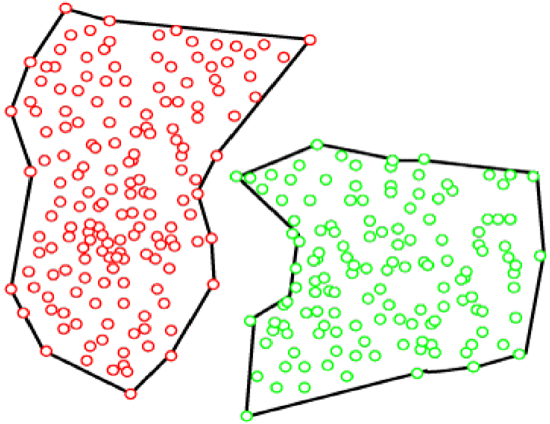
Density-Based Clustering:

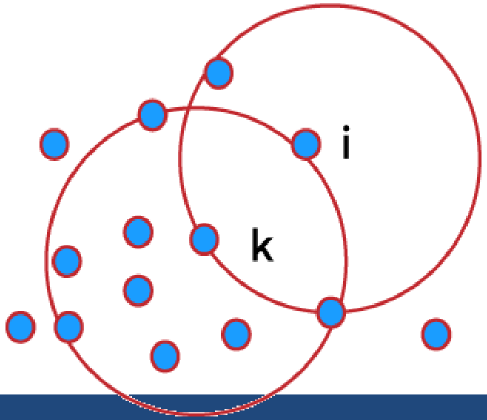
Two important parameters: eps (epsilon) and minPts.

- ❑ Eps: defines the maximum distance between two data points for them to be considered neighbors.

- ❑ minPts: specifies the minimum number of points that must be within the eps distance of a data point to be considered a core point.

Density-Based Clustering:





MinPts = 5

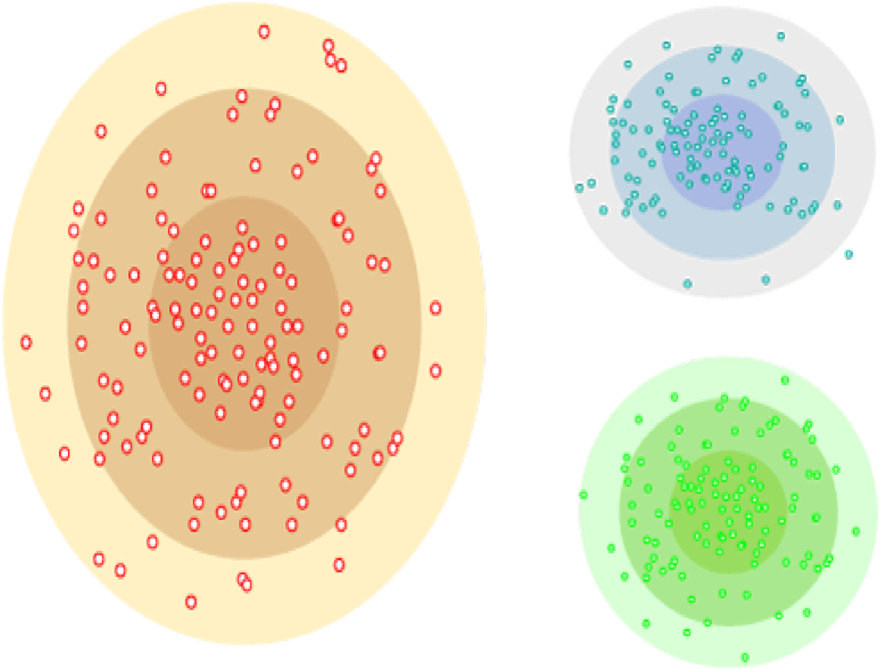
Eps = 1 cm

Distribution Model-Based Clustering

- In the distribution model-based clustering method, the data is divided based on the probability of how a dataset belongs to a particular distribution. The grouping is done by assuming some distributions commonly Gaussian Distribution.

- The example of this type is the Expectation-Maximization Clustering algorithm that uses Gaussian Mixture Models (GMM).

Distribution Model-Based Clustering



Hierarchical Clustering

- Hierarchical clustering can be used as an alternative for the partitioned clustering as there is no requirement of pre-specifying the number of clusters to be created.
- In this technique, the dataset is divided into clusters to create a tree-like structure, which is also called a dendrogram.
- The observations or any number of clusters can be selected by cutting the tree at the correct level.
- The most common example of this method is the Agglomerative Hierarchical algorithm.
- Sometimes the results of K-means clustering and hierarchical clustering may look similar, but they both differ depending on how they work. As there is no requirement to predetermine the number of clusters as we did in the K-Means algorithm.

Hierarchical Clustering

The hierarchical clustering technique has two approaches:

- **Agglomerative:** Agglomerative is a bottom-up approach, in which the algorithm starts with taking all data points as single clusters and merging them until one cluster is left.

- **Divisive:** Divisive algorithm is the reverse of the agglomerative algorithm as it is a top-down approach.

Hierarchical Clustering

Algorithm Steps:

- **Initialization:**

Each data point is initially treated as a separate cluster.

- **Pairwise Distance Calculation:**

The algorithm calculates the distance (similarity or dissimilarity) between all pairs of data points, typically using distance metrics like Euclidean distance.

- **Cluster Fusion (Agglomerative) or Division (Divisive):**

Agglomerative hierarchical clustering (bottom-up): Starts with individual data points as clusters and repeatedly merges the two closest clusters until all data points belong to a single cluster.

Divisive hierarchical clustering (top-down): Begins with all data points in a single cluster and recursively divides it into smaller clusters until each data point forms its own cluster.

- **Construction of Dendrogram:**

The results of the agglomerative or divisive process are represented as a dendrogram, a tree-like structure where the height of each branch represents the similarity level at which clusters are merged or divided.

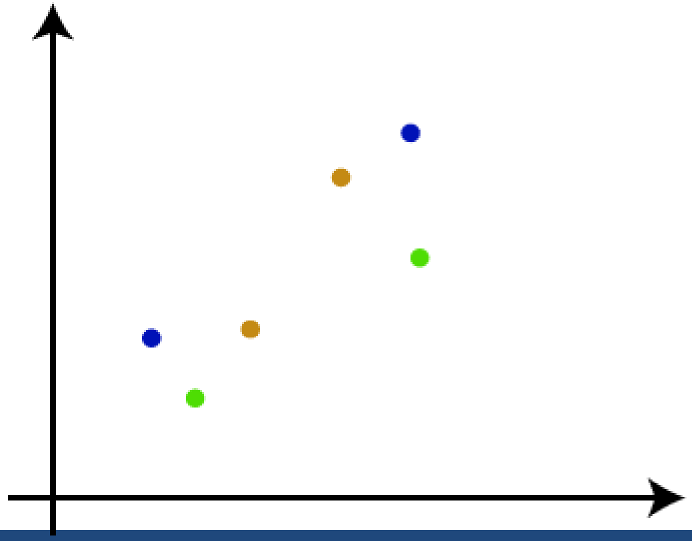
- **Dendrogram Cutting (Optional):**

To obtain a specific number of clusters, the dendrogram can be cut at a certain height, which corresponds to the desired number of clusters.

How the Agglomerative Hierarchical clustering Work?

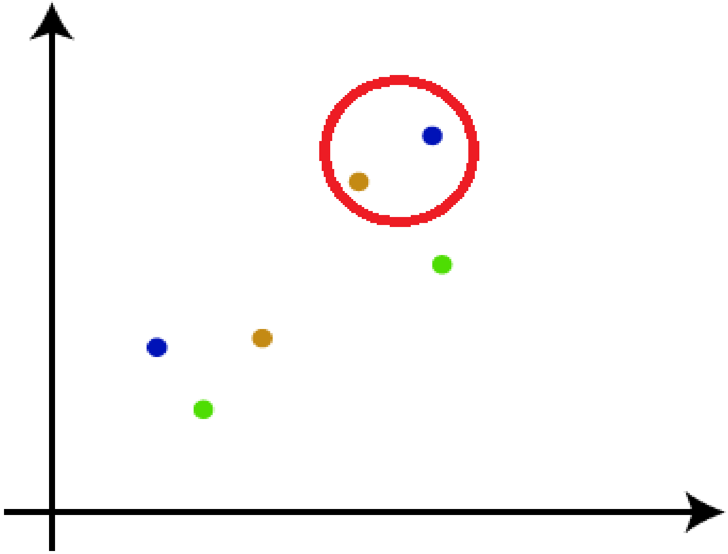
The working of the AHC algorithm can be explained using the below steps:

Step-1: Create each data point as a single cluster. Let's say there are N data points, so the number of clusters will also be N.



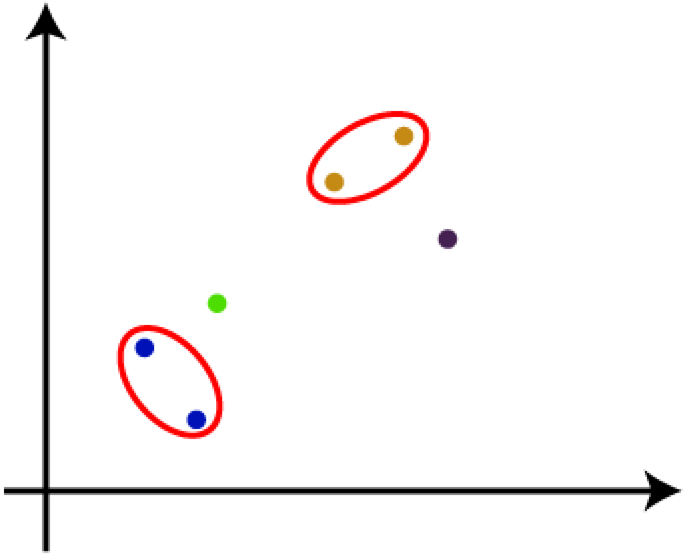
How the Agglomerative Hierarchical clustering Work?

Step-2: Take two closest data points or clusters and merge them to form one cluster. So, there will now be N-1 clusters.



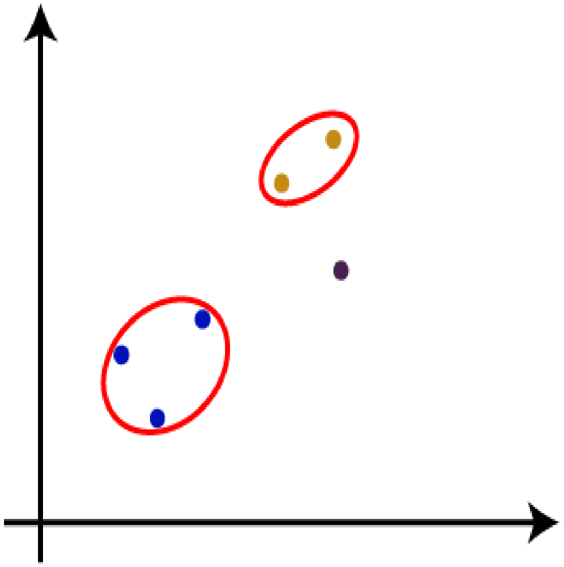
How the Agglomerative Hierarchical clustering Work?

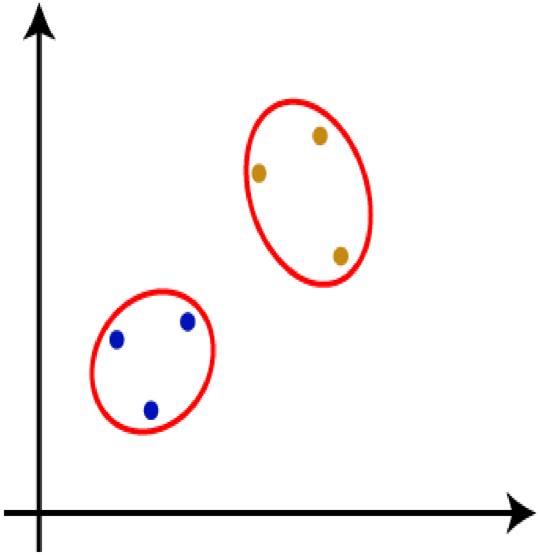
Step-3: Again, take the two closest clusters and merge them together to form one cluster. There will be N-2 clusters.



How the Agglomerative Hierarchical clustering Work?

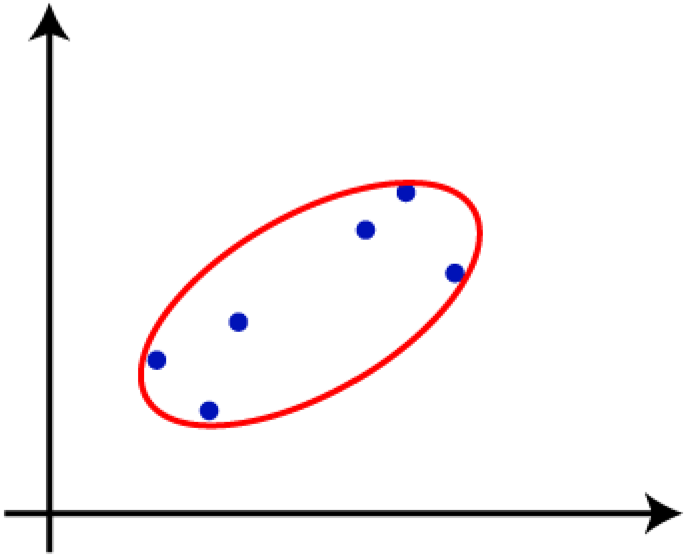
Step-4: Repeat Step 3 until only one cluster left. So, we will get the following clusters. Consider the below images:





How the Agglomerative Hierarchical clustering Work?

Step-4: Repeat Step 3 until only one cluster left. So, we will get the following clusters. Consider the below image:



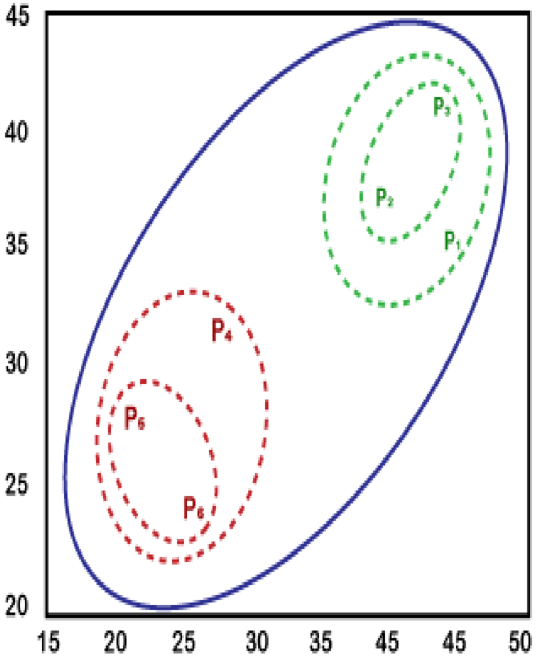
How the Agglomerative Hierarchical clustering Work?

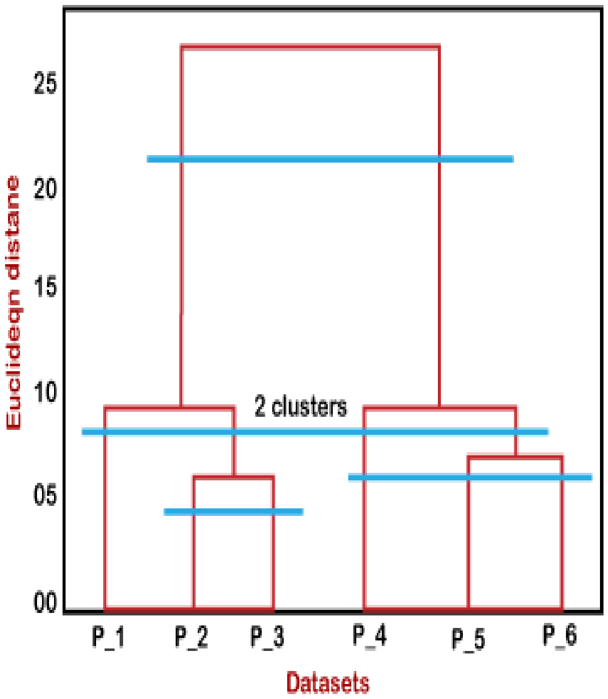
Step-5: Once all the clusters are combined into one big cluster, develop the dendrogram to divide the clusters as per the problem.

❑ **Dendrogram** is a tree-like structure that is mainly used to store each step as a memory that the HC algorithm performs. In the dendrogram plot, the Y-axis shows the Euclidean distances between the data points, and the x-axis shows all the data points of the given dataset.

How the Agglomerative Hierarchical clustering Work?

The working of the dendrogram can be explained using the below diagram:





Fuzzy clustering

Fuzzy clustering is a type of clustering algorithm that allows data points to belong to multiple clusters simultaneously, with varying degrees of membership. Unlike traditional clustering methods that assign each data point to a single cluster, fuzzy clustering provides a more flexible approach by assigning fuzzy membership values to each data point, indicating the degree of belongingness to each cluster

Fuzzy clustering

Concept: The concept of fuzzy clustering is inspired by fuzzy set theory, where membership degrees are expressed as values between 0 and 1, representing the degree of uncertainty or ambiguity in data point assignments. A data point may have partial membership in multiple clusters based on its similarity to the cluster centroids.

Fuzzy clustering

Algorithm Steps:

Initialization:

Start by specifying the number of clusters (K) and initializing the cluster centroids. Randomly assign initial membership values for each data point with respect to the centroids.

Membership Value Calculation:

Calculate the membership value for each data point for each cluster based on a specified membership function (often based on distance or similarity measures).

The membership function determines how close a data point is to a particular cluster centroid, influencing its membership degree.

Fuzzy clustering

Algorithm Steps:

Update Cluster Centroids:
Recalculate the cluster centroids based on the current fuzzy membership values of data points.
The updated centroids represent the weighted mean of data points, where the membership values serve as weights.

Iteration:
Repeat the membership value calculation and centroid update steps iteratively until convergence or a stopping criterion is met.

K-Means clustering

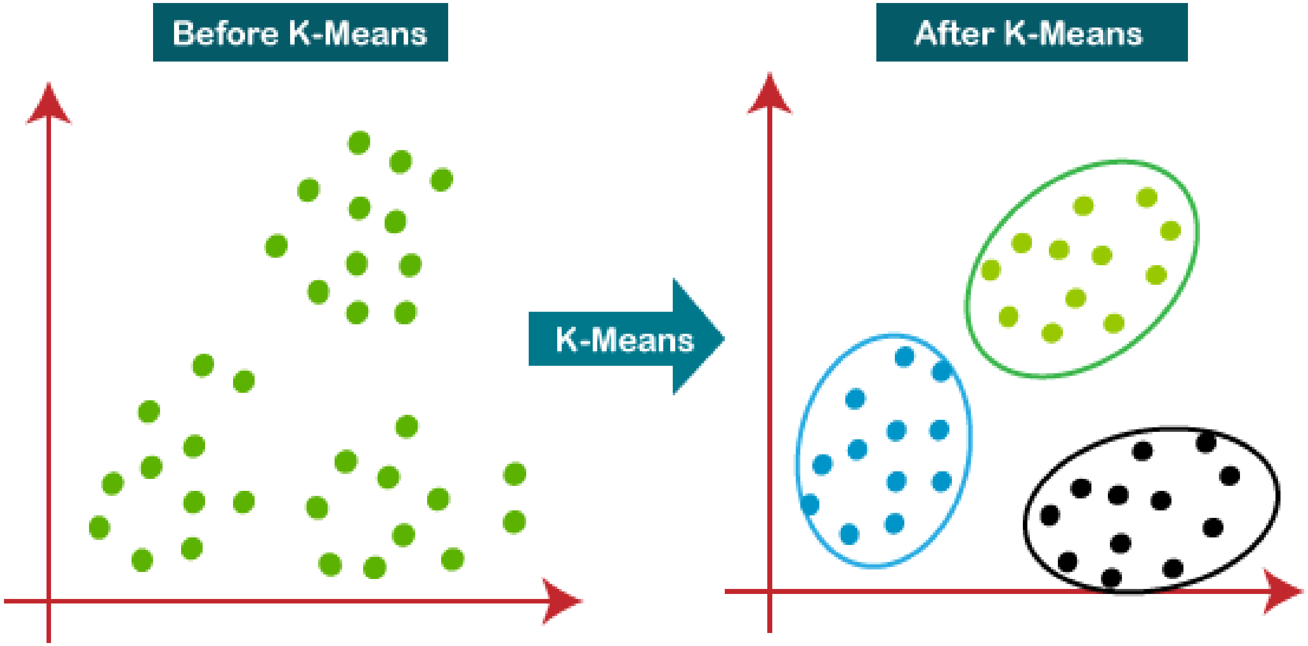
- K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science.
- It groups the unlabeled dataset into different clusters.
- Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.
- It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.
- It is a centroid-based algorithm, where each cluster is associated with a centroid.

K-Means clustering

- The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.
- The k-means clustering algorithm mainly performs two tasks:
- Determines the best value for K center points or centroids by an iterative process.
- Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

How does the K-Means Algorithm Work?

The below diagram explains the working of the K-means Clustering Algorithm:



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.

Advantages of k-means

- Simple and easy to implement: The k-means algorithm is easy to understand and implement, making it a popular choice for clustering tasks.
- Fast and efficient: K-means is computationally efficient and can handle large datasets with high dimensionality.
- Scalability: K-means can handle large datasets with a large number of data points and can be easily scaled to handle even larger datasets.
- Flexibility: K-means can be easily adapted to different applications and can be used with different distance metrics and initialization methods

Disadvantages of K-Means:

- Sensitivity to initial centroids: K-means is sensitive to the initial selection of centroids and can converge to a suboptimal solution.
- Requires specifying the number of clusters: The number of clusters k needs to be specified before running the algorithm, which can be challenging in some applications.
- Sensitive to outliers: K-means is sensitive to outliers, which can have a significant impact on the resulting clusters.

Applications of K-Means Clustering:

1. Customer Segmentation:
Segmenting customers based on their behavior, preferences, and purchase history.
Helps businesses tailor marketing strategies and personalize offerings to different customer groups.
2. Image Compression:
Reducing the size of images by clustering similar color pixels together.
Retains visual quality while reducing storage space and transmission time.
3. Anomaly Detection:
Identifying unusual patterns or outliers in data that deviate significantly from normal behavior.
Used in fraud detection, network intrusion detection, and identifying faulty equipment.
4. Document Clustering:
Organizing large text documents into clusters based on similarity.
Useful for topic modeling, information retrieval, and content organization.

Applications of K-Means Clustering:

5. Recommendation Systems:

Recommending products or content to users based on their preferences and behaviors.

Enables personalized and targeted recommendations in e-commerce and content platforms.

6. Market Segmentation:

Dividing a market into distinct segments based on characteristics like demographics, buying behavior, and geography.

Aids in identifying niche markets and crafting tailored marketing strategies.

7. Bioinformatics:

Analyzing biological data such as gene expression profiles to group genes with similar functions or properties.

Facilitates gene discovery and functional annotation.

Applications of K-Means Clustering:

8. Climate Pattern Identification:

Identifying weather or climate patterns based on historical data.

Useful in climate research and understanding climate change patterns.

9. Social Network Analysis:

Clustering users in social networks based on their connections, interests, or interactions.

Helps in identifying communities and influencers.

10. Retail Store Location Planning:

Identifying optimal locations for new retail stores based on customer demographics and preferences.

Maximizes potential customer reach and profitability.

KModes Clustering Algorithm:

- KModes is a clustering algorithm used in data science to group similar data points into clusters based on their categorical attributes.
- Unlike traditional clustering algorithms that use distance metrics, KModes works by identifying the modes or most frequent values within each cluster to determine its centroid.
- KModes is ideal for clustering categorical data such as customer demographics, market segments, or survey responses.
- It is a powerful tool for data analysts and scientists to gain insights into their data and make informed decisions.

KModes vs KMeans:

- KMeans uses mathematical measures (distance) to cluster continuous data.
- The lesser the distance, the more similar our data points are. Centroids are updated by Means.
- But for categorical data points, we cannot calculate the distance. So we go for KModes algorithm. It uses the dissimilarities(total mismatches) between the data points.
- The lesser the dissimilarities the more similar our data points are. It uses Modes instead of means.

How Does the KModes Algorithm Work?:

Unlike Hierarchical clustering methods, we need to upfront specify the K.

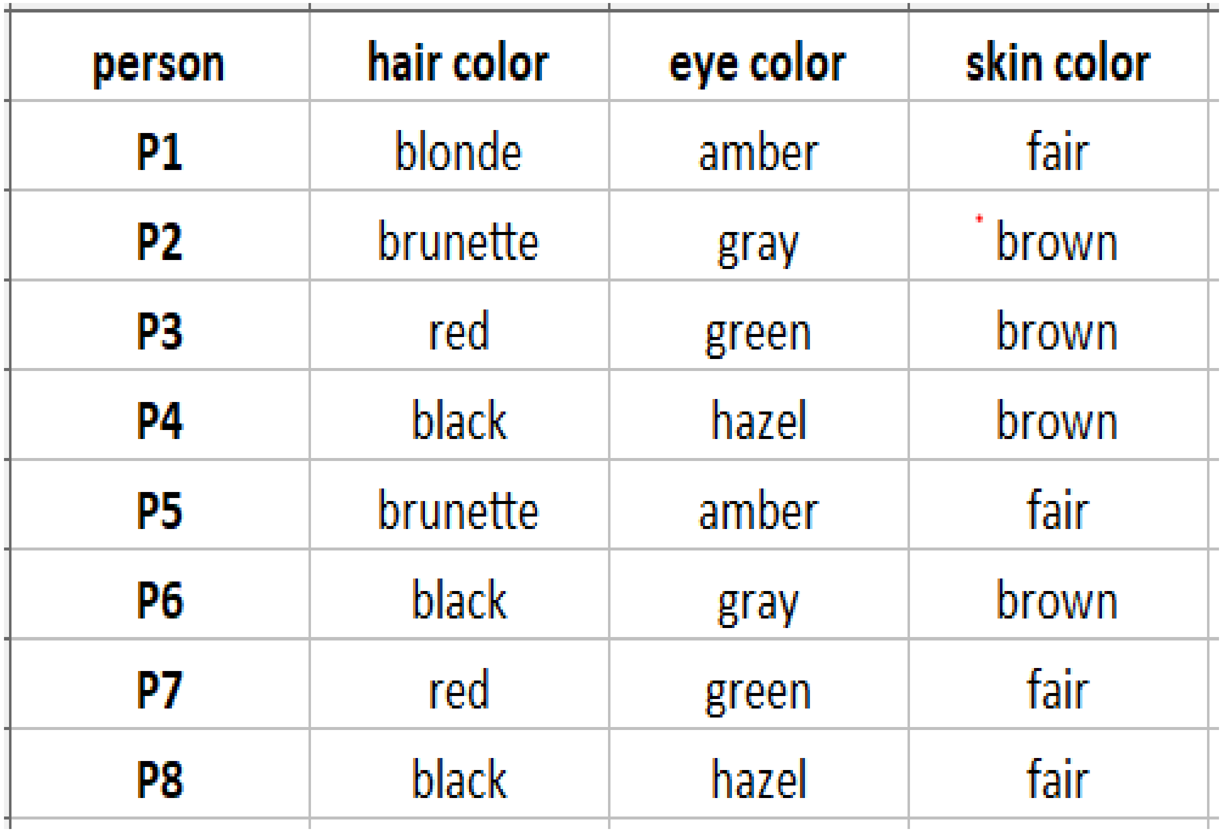
1. Pick K observations at random and use them as leaders/clusters
2. Calculate the dissimilarities and assign each observation to its closest cluster
3. Define new modes for the clusters
4. Repeat 2–3 steps until there are is no re-assignment required

How Does the KModes Algorithm Work?:

Example: Imagine we have a dataset that has the information about hair color, eye color, and skin color of persons. We aim to group them based on the available information(maybe we want to suggest some styling ideas)

Hair color, eye color, and skin color are all categorical variables.

How Does the KModes Algorithm Work?:

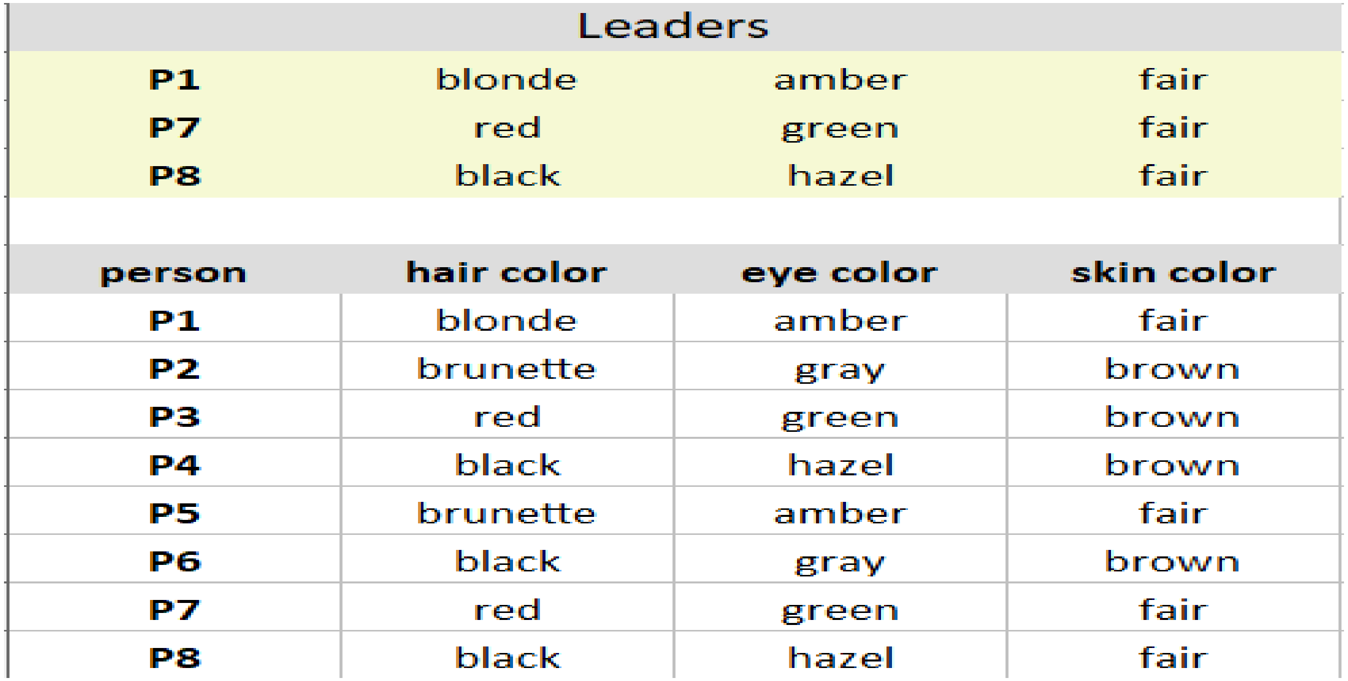


person	hair color	eye color	skin color
P1	blonde	amber	fair
P2	brunette	gray	brown
P3	red	green	brown
P4	black	hazel	brown
P5	brunette	amber	fair
P6	black	gray	brown
P7	red	green	fair
P8	black	hazel	fair

How Does the KModes Algorithm Work?:

Step 1: Pick K observations at random and use them as leaders/clusters

I am choosing P1, P7, P8 as leaders/clusters



Leaders

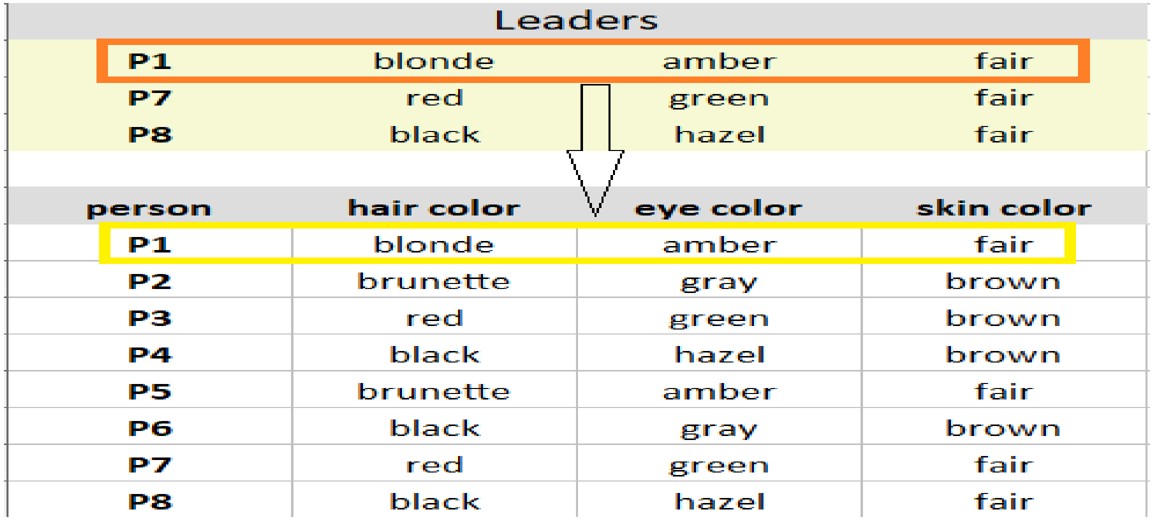
P1	blonde	amber	fair
P7	red	green	fair
P8	black	hazel	fair

person	hair color	eye color	skin color
P1	blonde	amber	fair
P2	brunette	gray	brown
P3	red	green	brown
P4	black	hazel	brown
P5	brunette	amber	fair
P6	black	gray	brown
P7	red	green	fair
P8	black	hazel	fair

How Does the KModes Algorithm Work?:

Step 2: Calculate the dissimilarities(no. of mismatches) and assign each observation to its closest cluster

Iteratively compare the cluster data points to each of the observations. Similar data points give 0, dissimilar data points give 1.

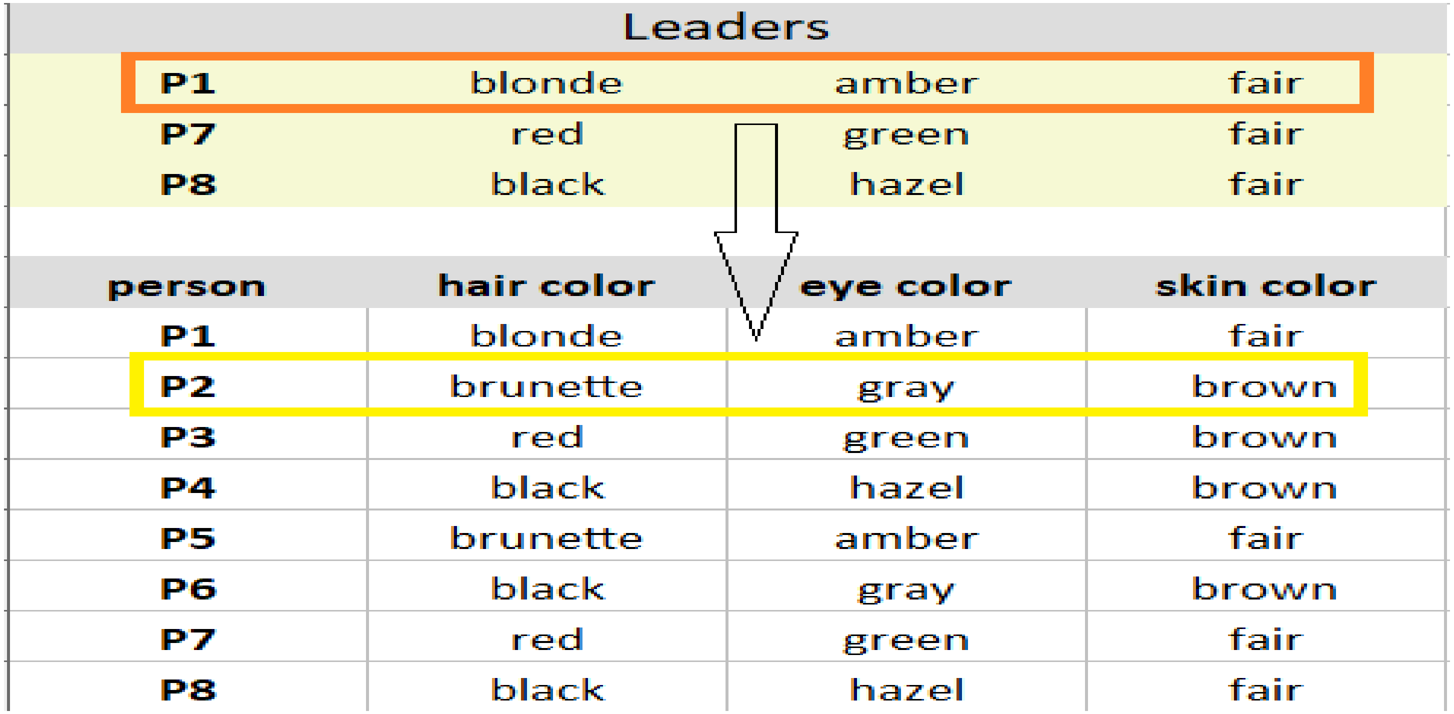


Leaders			
P1	blonde	amber	fair
P7	red	green	fair
P8	black	hazel	fair

person	hair color	eye color	skin color
P1	blonde	amber	fair
P2	brunette	gray	brown
P3	red	green	brown
P4	black	hazel	brown
P5	brunette	amber	fair
P6	black	gray	brown
P7	red	green	fair
P8	black	hazel	fair

How Does the KModes Algorithm Work?:

Comparing leader/Cluster P1 to the observation P1 gives 0 dissimilarities.

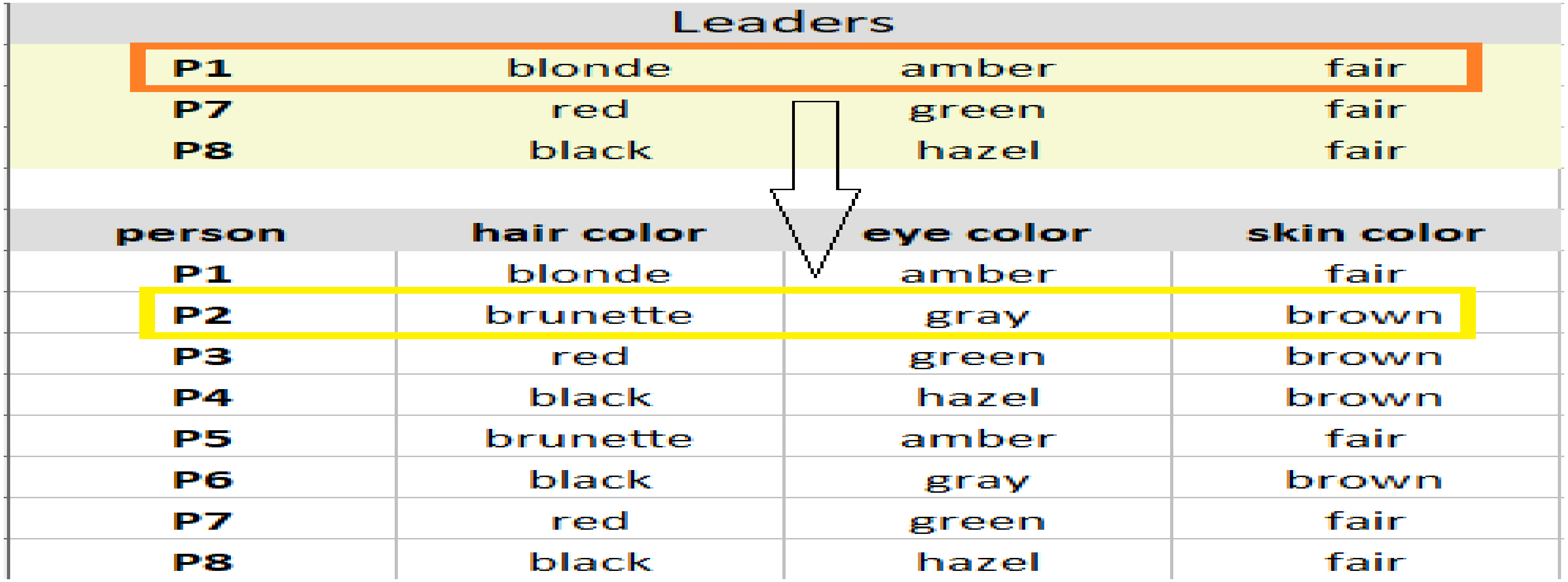


Leaders			
P1	blonde	amber	fair
P7	red	green	fair
P8	black	hazel	fair

person	hair color	eye color	skin color
P1	blonde	amber	fair
P2	brunette	gray	brown
P3	red	green	brown
P4	black	hazel	brown
P5	brunette	amber	fair
P6	black	gray	brown
P7	red	green	fair
P8	black	hazel	fair

How Does the KModes Algorithm Work?:

Comparing leader/Cluster P1 to the observation P1 gives 0 dissimilarities.



Leaders

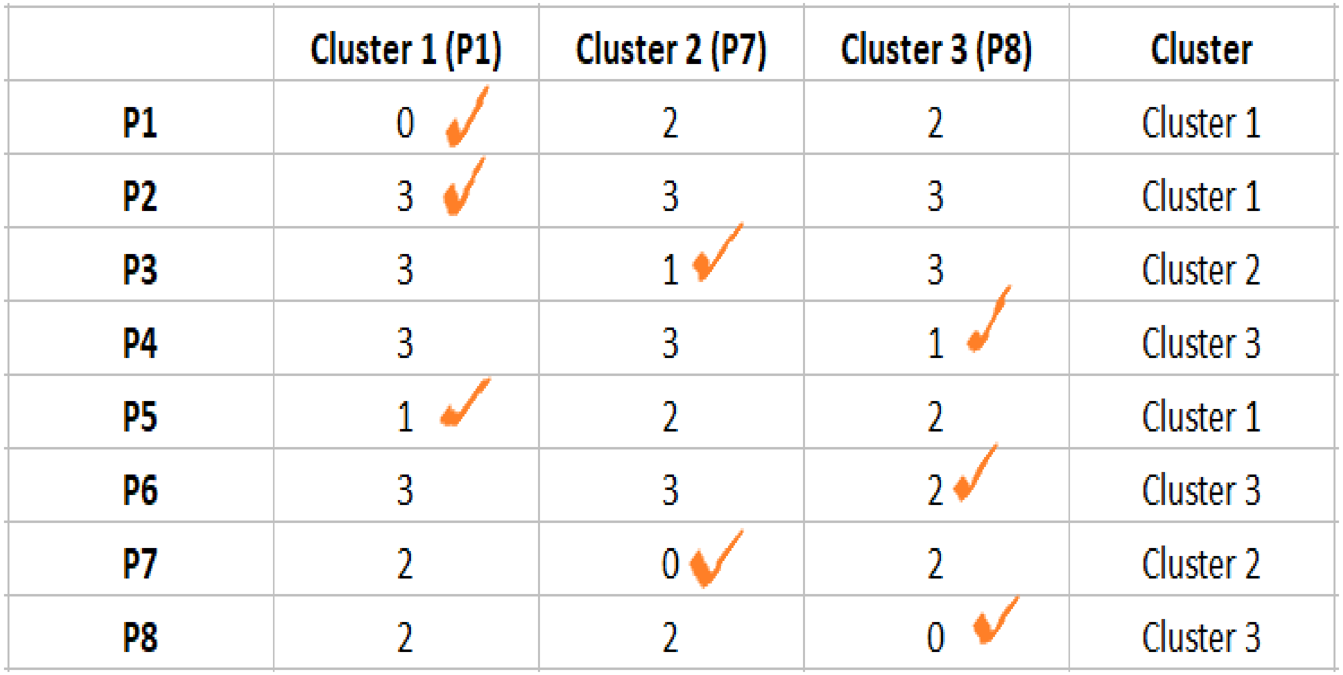
P1	blonde	amber	fair
P7	red	green	fair
P8	black	hazel	fair

person	hair color	eye color	skin color
P1	blonde	amber	fair
P2	brunette	gray	brown
P3	red	green	brown
P4	black	hazel	brown
P5	brunette	amber	fair
P6	black	gray	brown
P7	red	green	fair
P8	black	hazel	fair

Comparing leader/cluster P1 to the observation P2 gives 3(1+1+1) dissimilarities

How Does the KModes Algorithm Work?:

Likewise, calculate all the dissimilarities and put them in a matrix as shown below and assign the observations to their closest cluster(cluster that has the least dissimilarity).



	Cluster 1 (P1)	Cluster 2 (P7)	Cluster 3 (P8)	Cluster
P1	0 ✓	2	2	Cluster 1
P2	3 ✓	3	3	Cluster 1
P3	3	1 ✓	3	Cluster 2
P4	3	3	1 ✓	Cluster 3
P5	1 ✓	2	2	Cluster 1
P6	3	3	2 ✓	Cluster 3
P7	2	0 ✓	2	Cluster 2
P8	2	2	0 ✓	Cluster 3

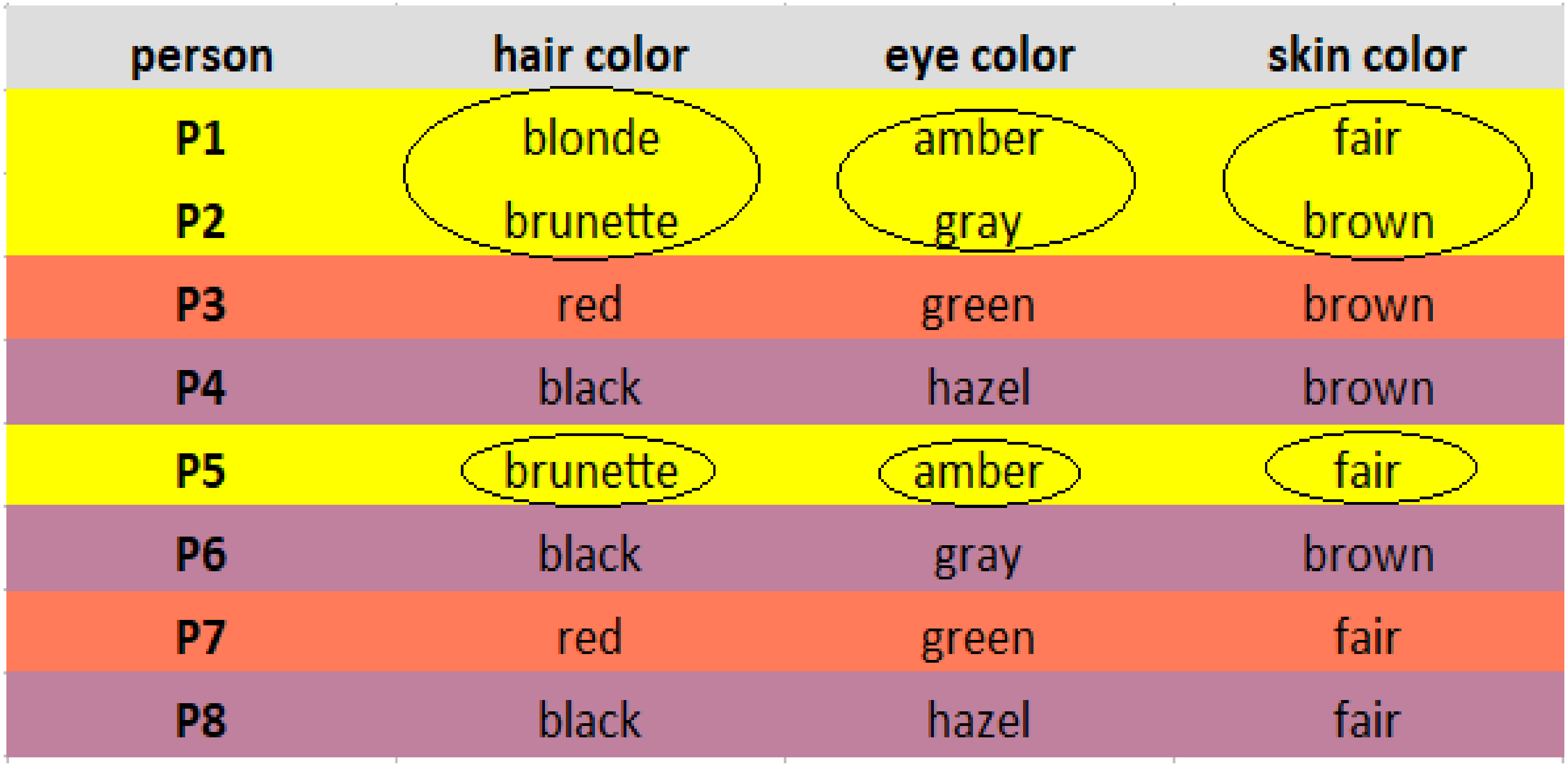
How Does the KModes Algorithm Work?:

- After step 2, the observations P1, P2, P5 are assigned to cluster 1; P3, P7 are assigned to Cluster 2; and P4, P6, P8 are assigned to cluster 3
- Note: If all the clusters have the same dissimilarity with an observation, assign to any cluster randomly. In our case, the observation P2 has 3 dissimilarities with all the leaders. I randomly assigned it to Cluster 1.

Step 3: Define new modes for the clusters

- Mode is simply the most observed value.
- Mark the observations according to the cluster they belong to. Observations of Cluster 1 are marked in Yellow, Cluster 2 are marked in Brick red, and Cluster 3 are marked in Purple.

How Does the KModes Algorithm Work?:



person	hair color	eye color	skin color
P1	blonde	amber	fair
P2	brunette	gray	brown
P3	red	green	brown
P4	black	hazel	brown
P5	brunette	amber	fair
P6	black	gray	brown
P7	red	green	fair
P8	black	hazel	fair

How Does the KModes Algorithm Work?:

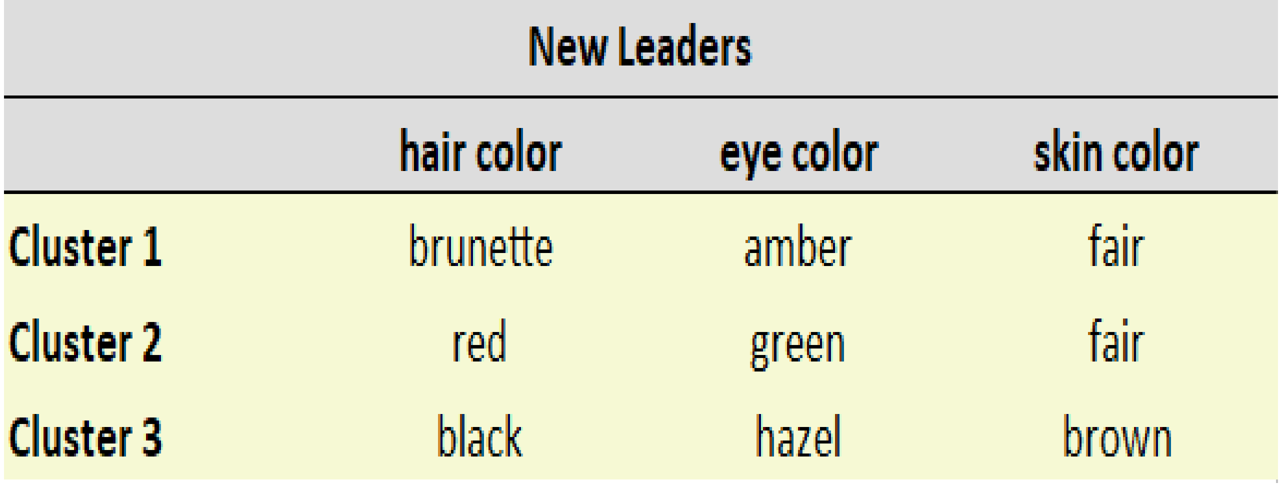
- Considering one cluster at a time, for each feature, look for the Mode and update the new leaders.

- Explanation: Cluster 1 observations(P1, P2, P5) has brunette as the most observed hair color, amber as the most observed eye color, and fair as the most observed skin color.

- Note: If you observe the same occurrence of values, take the mode randomly. In our case, the observations of Cluster 3(P3, P7) have one occurrence of brown, fair skin color. I randomly chose brown as the mode.

How Does the KModes Algorithm Work?:

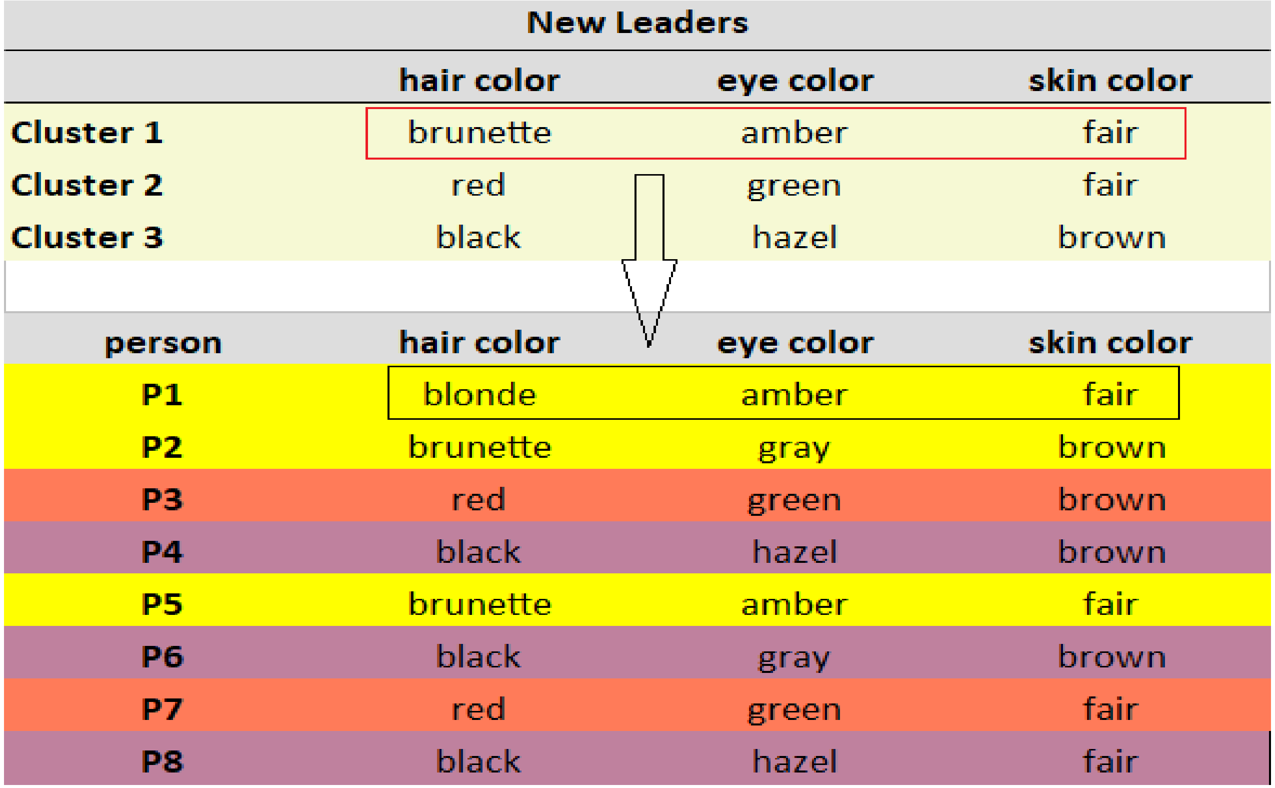
Below are our new leaders after the update.



New Leaders			
	hair color	eye color	skin color
Cluster 1	brunette	amber	fair
Cluster 2	red	green	fair
Cluster 3	black	hazel	brown

- Repeat steps 2–4
- After obtaining the new leaders, again calculate the dissimilarities between the observations and the newly obtained leaders.

How Does the KModes Algorithm Work?:

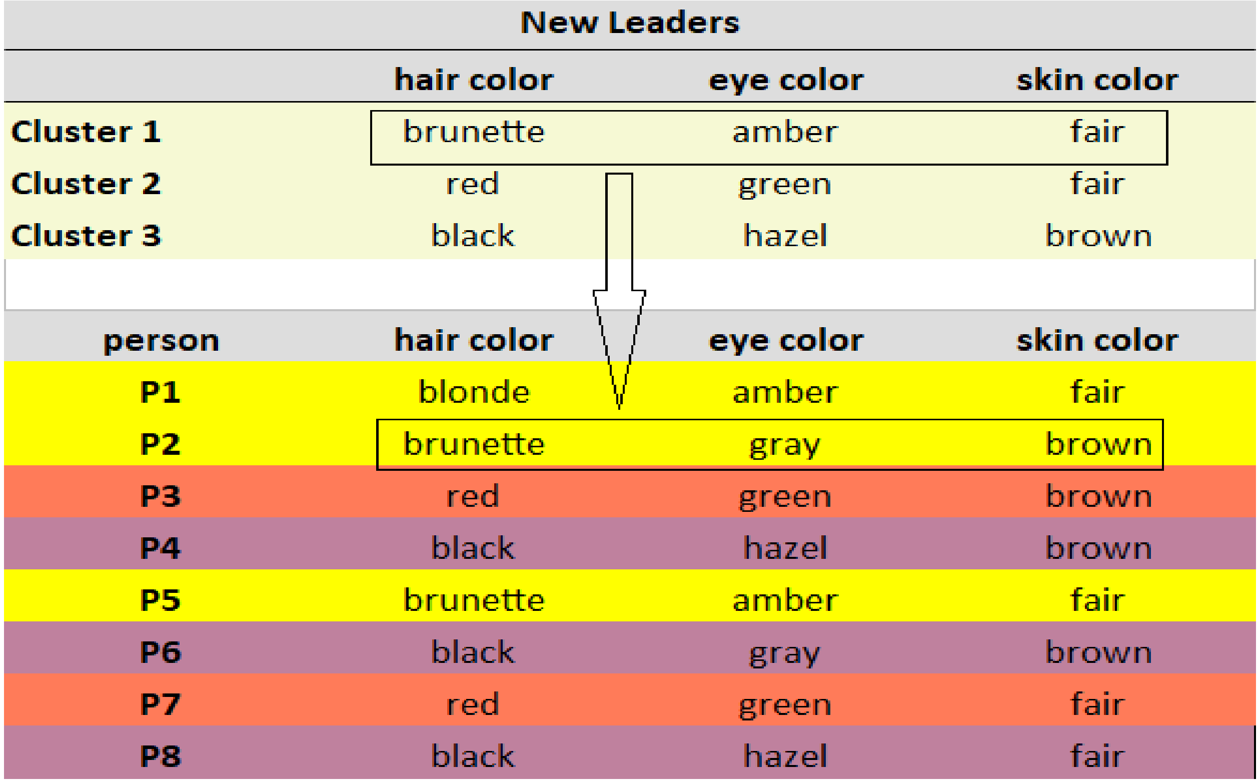


New Leaders			
	hair color	eye color	skin color
Cluster 1	brunette	amber	fair
Cluster 2	red	green	fair
Cluster 3	black	hazel	brown

person	hair color	eye color	skin color
P1	blonde	amber	fair
P2	brunette	gray	brown
P3	red	green	brown
P4	black	hazel	brown
P5	brunette	amber	fair
P6	black	gray	brown
P7	red	green	fair
P8	black	hazel	fair

Comparing Cluster 1 to the observation P1 gives 1 dissimilarity.

How Does the KModes Algorithm Work?:



New Leaders

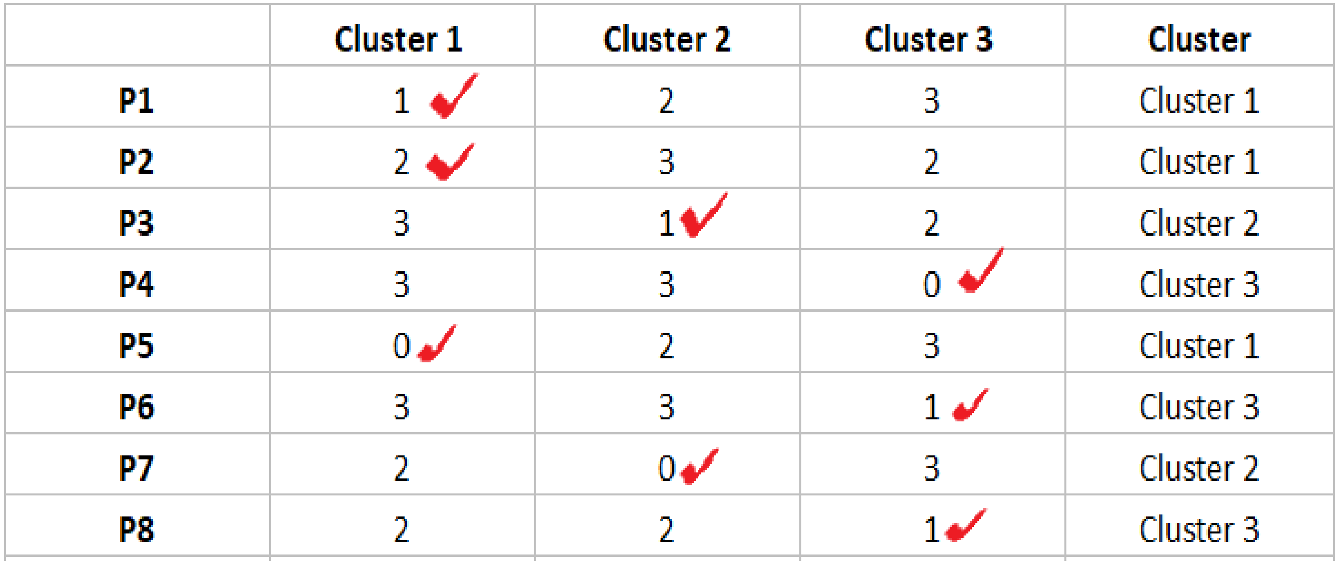
	hair color	eye color	skin color
Cluster 1	brunette	amber	fair
Cluster 2	red	green	fair
Cluster 3	black	hazel	brown

person	hair color	eye color	skin color
P1	blonde	amber	fair
P2	brunette	gray	brown
P3	red	green	brown
P4	black	hazel	brown
P5	brunette	amber	fair
P6	black	gray	brown
P7	red	green	fair
P8	black	hazel	fair

Comparing Custer 1 to the observation P2 gives 2 dissimilarities.

How Does the KModes Algorithm Work?:

Likewise, calculate all the dissimilarities and put them in a matrix. Assign each observation to its closest cluster.



	Cluster 1	Cluster 2	Cluster 3	Cluster
P1	1 ✓	2	3	Cluster 1
P2	2 ✓	3	2	Cluster 1
P3	3	1 ✓	2	Cluster 2
P4	3	3	0 ✓	Cluster 3
P5	0 ✓	2	3	Cluster 1
P6	3	3	1 ✓	Cluster 3
P7	2	0 ✓	3	Cluster 2
P8	2	2	1 ✓	Cluster 3

- The observations P1, P2, P5 are assigned to Cluster 1; P3, P7 are assigned to Cluster 2; and P4, P6, P8 are assigned to Cluster 3.
- We stop here as we see there is no change in the assignment of observations.

Self Organizing Maps:

- Self Organizing Map (or Kohonen Map or SOM) is a type of Artificial Neural Network which is also inspired by biological models of neural systems from the 1970s.

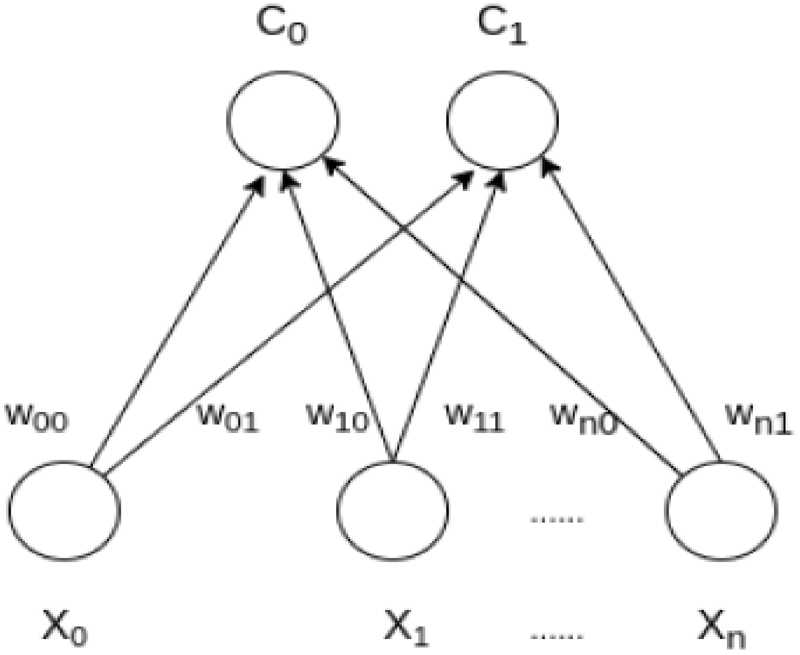
- It follows an unsupervised learning approach and trained its network through a competitive learning algorithm.

- SOM is used for clustering and mapping (or dimensionality reduction) techniques to map multidimensional data onto lower-dimensional which allows people to reduce complex problems for easy interpretation.

- SOM has two layers, one is the Input layer and the other one is the Output layer.

Self Organizing Maps:

The architecture of the Self Organizing Map with two clusters and n input features of any sample is given below:



How do SOM works?:

- Let’s say an input data of size (m, n) where m is the number of training examples and n is the number of features in each example.

- First, it initializes the weights of size (n, C) where C is the number of clusters.

- Then iterating over the input data, for each training example, it updates the winning vector (weight vector with the shortest distance (e.g Euclidean distance) from training example.

How do SOM works?:

Weight updation rule is given by :

wij = wij(old) + alpha(t) * (xik - wij(old))

Where,

- alpha is a learning rate at time t
- j denotes the winning vector
- i denotes the ith feature of training example
- k denotes the kth training example from the input data

After training the SOM network, trained weights are used for clustering new examples. A new example falls in the cluster of winning vectors.

SOM Algorithm:

Training:

Step 1: Initialize the weights wij random value may be assumed. Initialize the learning rate α.

Step 2: Calculate squared Euclidean distance.

D(j) = Σ (wij – xi)^2 where i=1 to n and j=1 to m

Step 3: Find index J, when D(j) is minimum that will be considered as winning index.

SOM Algorithm:

Training:

Step 4: For each j within a specific neighborhood of j and for all i, calculate the new weight.

$$wij(new)=wij(old) + \alpha[xi - wij(old)]$$

Step 5: Update the learning rule by using :

$$\alpha(t+1) = 0.5 * t$$

Step 6: Test the Stopping Condition.

Pros And Cons Of Self-Organizing Maps

Self-organizing maps have both advantages and disadvantages, some of which are shown below:

Pros:

1. Techniques like dimensionality reduction and grid clustering can make it simple to understand and comprehend data.

2. Self-organizing maps can handle a variety of categorization issues while simultaneously producing an insightful and practical summary of the data.

Pros And Cons Of Self-Organizing Maps

Cons :

1. The model cannot grasp how data is formed since it does not generate a generative data model.

2. When dealing with categorical data, Self-Organizing Maps perform poorly, and when dealing with mixed forms of data, they do much worse.

3. In comparison, the model preparation process is extremely slow, making it challenging to train against slowly evolving data.

Expectation-Maximization (EM) algorithm

- The Expectation-Maximization (EM) algorithm is defined as the combination of various unsupervised machine learning algorithms, which is used to determine the **local maximum likelihood estimates (MLE) or maximum a posteriori estimates (MAP)** for unobservable variables in statistical models.

- It is a technique to find maximum likelihood estimation when the latent variables are present.

- It is also referred to as the **latent variable model.**

Expectation-Maximization (EM) algorithm

first understand what is meant by the latent variable model?

- A latent variable model consists of observable variables along with **unobservable variables.** Observed variables are those variables in the dataset that can be measured whereas unobserved (latent/hidden) variables are inferred from the observed variables.

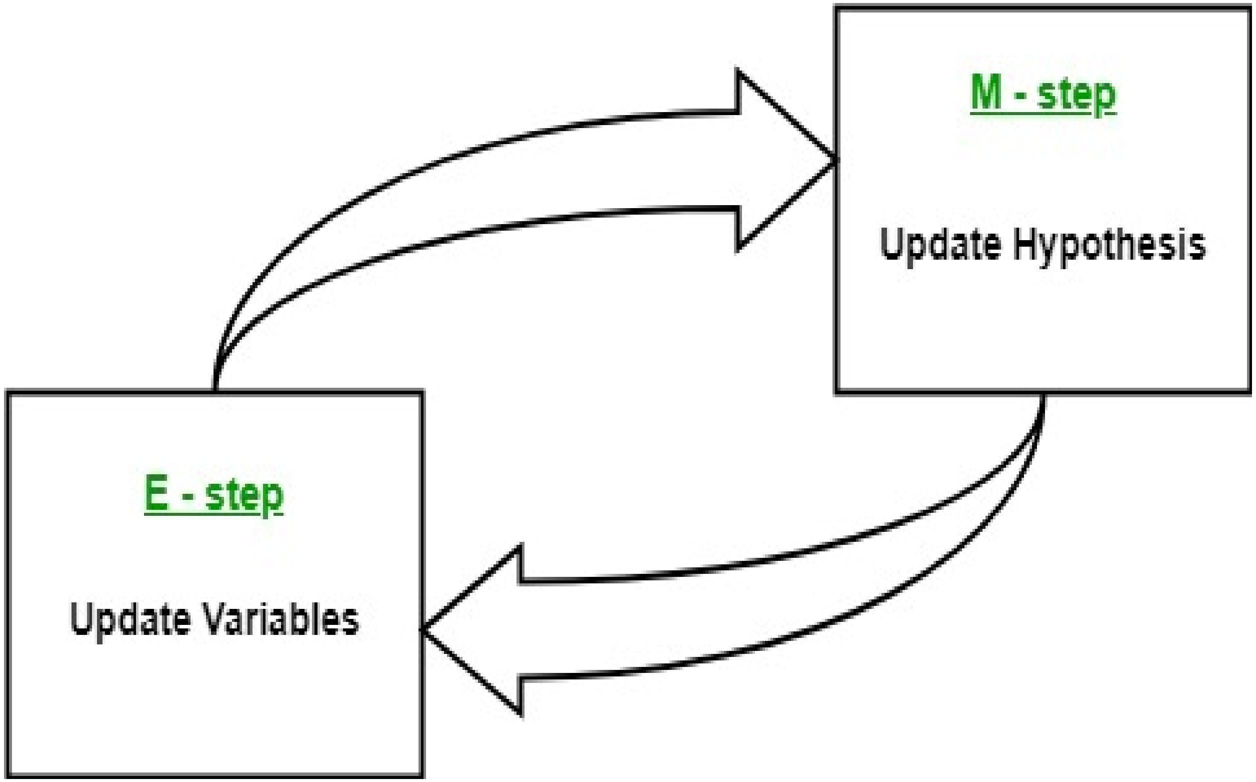
- It can be used to find the **local maximum likelihood (MLE) parameters or maximum a posteriori (MAP)** parameters for latent variables in a statistical or mathematical model.

Expectation-Maximization (EM) algorithm

- It is used to predict these missing values in the dataset, provided we know the general form of probability distribution associated with these latent variables.

- In simple words, the basic idea behind this algorithm is to use the observable samples of latent variables to predict the values of samples that are unobservable for learning. This process is repeated until the convergence of the values occurs.

How Expectation-Maximization (EM) Works:



How Expectation-Maximization (EM) Works

1. Given a set of incomplete data, start with a set of initialized parameters.

2. **Expectation step (E – step):** In this expectation step, by using the observed available data of the dataset, we can try to estimate or guess the values of the missing data. Finally, after this step, we get complete data having no missing values.

3. **Maximization step (M – step):** This step involves the use of estimated data in the E-step and updating the parameters.

4. Repeat step 2 and step 3 until we converge to our solution.

Expectation-Maximization (EM) algorithm

Let us understand the EM algorithm in a detailed manner:

- **Initialization Step:** In this step, we initialized the parameter values with a set of initial values, then give the set of incomplete observed data to the system with the assumption that the observed data comes from a specific model i.e., probability distribution.
- **Expectation Step:** In this step, by using the observed data to estimate or guess the values of the missing or incomplete data. It is used to update the variables.

Expectation-Maximization (EM) algorithm

- **Maximization Step:** In this step, we use the complete data generated in the “Expectation” step to update the values of the parameters i.e, update the hypothesis.

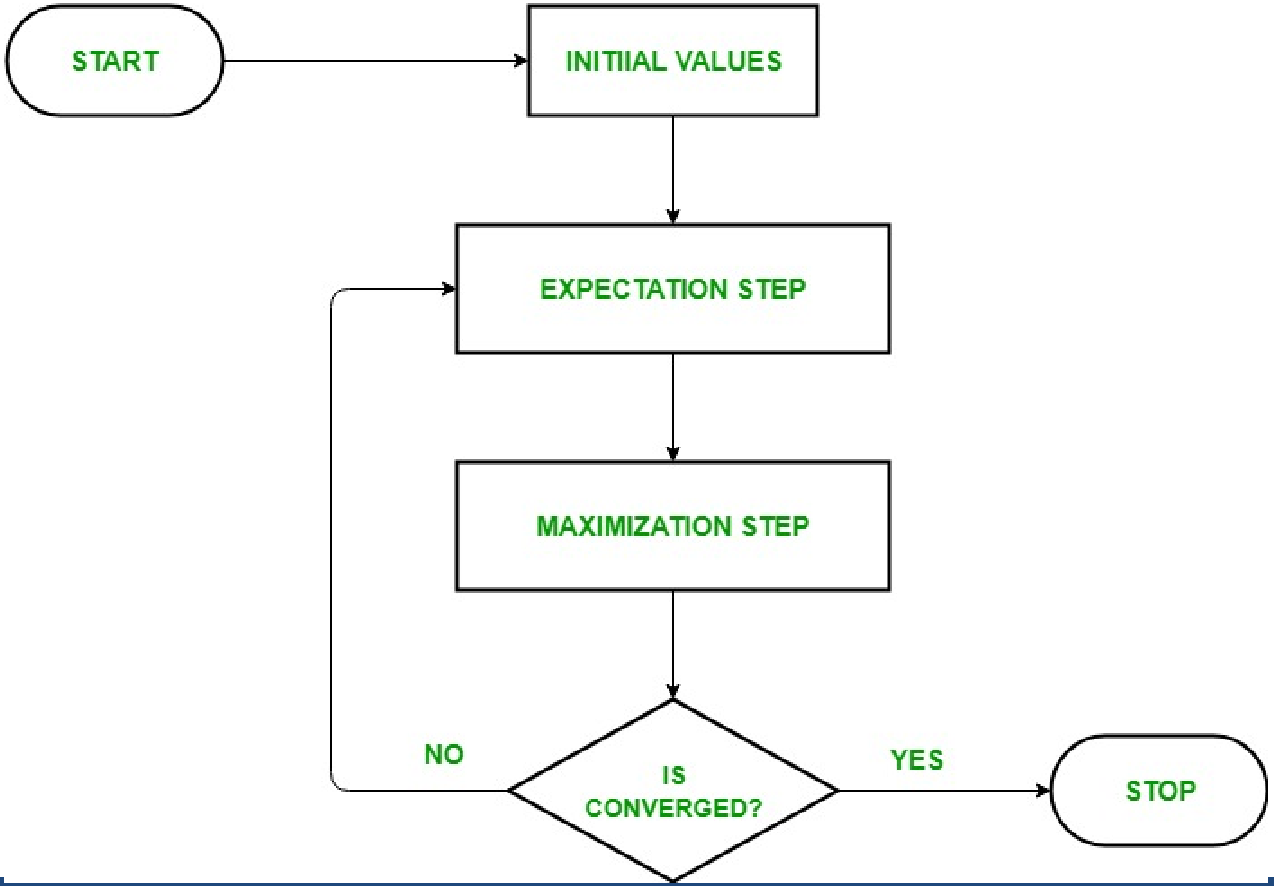
- **Checking of convergence Step:** Now, in this step, we checked whether the values are converging or not, if yes, then stop otherwise repeat these two steps i.e, the “Expectation” step and “Maximization” step until the convergence occurs.

Expectation-Maximization (EM) algorithm

What is Convergence in the EM algorithm?

- **Convergence is defined as the specific situation in probability based on intuition**, e.g., if there are two random variables that have very less difference in their probability, then they are known as converged. In other words, whenever the values of given variables are matched with each other, it is called convergence.

Flow chart for EM algorithm



Advantages and Disadvantages of EM algorithm

Advantages:

1. The basic two steps of the EM algorithm i.e, E-step and M-step are often pretty easy for many of the machine learning problems in terms of implementation.
2. The solution to the M-steps often exists in the closed-form.
3. It is always guaranteed that the value of likelihood will increase after each iteration.

Advantages and Disadvantages of EM algorithm

Disadvantages

1. It has **slow convergence.**
2. It converges to the **local optimum only**.
3. It takes both forward and backward probabilities into account. This thing is in contrast to that of numerical optimization which considers only **forward probabilities.**

Applications

The latent variable model has several real-life applications in Machine learning:

- Used to calculate the **Gaussian density of a function.**
- Helpful to fill in the **missing data** during a sample.
- It finds plenty of use in different domains such as **Natural Language Processing (NLP), Computer Vision, etc.**
- Used in image reconstruction in the field of **Medicine and Structural Engineering.**
- Used for estimating the parameters of the **Hidden Markov Model (HMM)** and also for some other mixed models like **Gaussian Mixture Models**, etc.
- Used for finding the values of latent variables.

Principal Component Analysis

- Principal Component Analysis is an unsupervised learning algorithm that is used for the dimensionality reduction in machine learning.
- It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation.
- These new transformed features are called the Principal Components.
- It is one of the popular tools that is used for exploratory data analysis and predictive modeling.
- It is a technique to draw strong patterns from the given dataset by reducing the variances.

Principal Component Analysis

- PCA generally tries to find the lower-dimensional surface to project the high-dimensional data.
- PCA works by considering the variance of each attribute because the high attribute shows the good split between the classes, and hence it reduces the dimensionality.
- Some real-world applications of PCA are image processing, movie recommendation system, optimizing the power allocation in various communication channels.
- It is a feature extraction technique, so it contains the important variables and drops the least important variable.

Principal Component Analysis

The PCA algorithm is based on some mathematical concepts such as:

- **Variance and Covariance**
- **Eigenvalues and Eigen factors**

Principal Component Analysis

Some common terms used in PCA algorithm:
Dimensionality: It is the number of features or variables present in the given dataset. More easily, it is the number of columns present in the dataset.

Correlation: It signifies that how strongly two variables are related to each other. Such as if one changes, the other variable also gets changed. The correlation value ranges from -1 to +1. Here, -1 occurs if variables are inversely proportional to each other, and +1 indicates that variables are directly proportional to each other.

Principal Component Analysis

Some common terms used in PCA algorithm:

Orthogonal: It defines that variables are not correlated to each other, and hence the correlation between the pair of variables is zero.

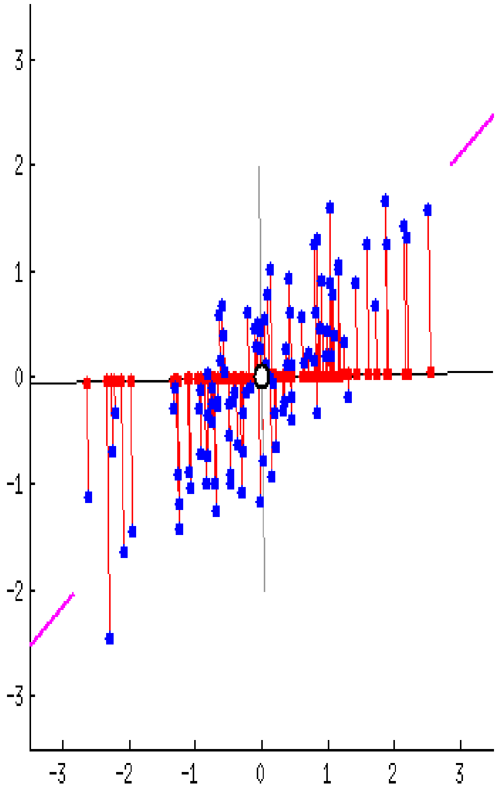
Eigenvectors: If there is a square matrix M, and a non-zero vector v is given. Then v will be eigenvector if Av is the scalar multiple of v.

Covariance Matrix: A matrix containing the covariance between the pair of variables is called the Covariance Matrix.

Principal Components in PCA:

- The transformed new features or the output of PCA are the Principal Components. The number of these PCs are either equal to or less than the original features present in the dataset. Some properties of these principal components are given below:
- The principal component must be the linear combination of the original features.
- These components are orthogonal, i.e., the correlation between a pair of variables is zero.
- The importance of each component decreases when going to 1 to n, it means the 1 PC has the most importance, and n PC will have the least importance.

Principal Components in PCA:



Steps for PCA algorithm:

1. Standardize the range of continuous initial variables
2. Compute the covariance matrix to identify correlations
3. Compute the eigenvectors and eigenvalues of the covariance matrix to identify the principal components
4. Create a feature vector to decide which principal components to keep
5. Recast the data along the principal components axes

Steps for PCA algorithm:

1. **Standardize the data:** PCA requires standardized data, so the first step is to standardize the data to ensure that all variables have a mean of 0 and a standard deviation of 1.

2. **Calculate the covariance matrix:** The next step is to calculate the covariance matrix of the standardized data. This matrix shows how each variable is related to every other variable in the dataset.

Steps for PCA algorithm:

3. Calculate the eigenvectors and eigenvalues: The eigenvectors and eigenvalues of the covariance matrix are then calculated. The eigenvectors represent the directions in which the data varies the most, while the eigenvalues represent the amount of variation along each eigenvector.

4. Choose the principal components: The principal components are the eigenvectors with the highest eigenvalues. These components represent the directions in which the data varies the most and are used to transform the original data into a lower-dimensional space.

5. transform the data: The final step is to transform the original data into the lower-dimensional space defined by the principal components.

Applications of Principal Component Analysis

- PCA is mainly used as the dimensionality reduction technique in various AI applications such as computer vision, image compression, etc.
- PCA is used to visualize multidimensional data.
- It is used to reduce the number of dimensions in healthcare data.
- PCA can help resize an image.
- It can be used in finance to analyze stock data and forecast returns.
- PCA helps to find patterns in the high-dimensional datasets.

Advantages of PCA

In terms of data analysis, PCA has a number of benefits, including:

❑ **Dimensionality reduction:** By determining the most crucial features or components, PCA reduces the dimensionality of the data, which is one of its primary benefits. This can be helpful when the initial data contains a lot of variables and is therefore challenging to visualize or analyze.

❑ **Feature Extraction:** PCA can also be used to derive new features or elements from the original data that might be more insightful or understandable than the original features. This is particularly helpful when the initial features are correlated or noisy

Advantages of PCA

- ❑ **Data visualization:** By projecting the data onto the first few principal components, PCA can be used to visualize high-dimensional data in two or three dimensions. This can aid in locating data patterns or clusters that may not have been visible in the initial high-dimensional space.

- ❑ **Noise Reduction:** By locating the underlying signal or pattern in the data, PCA can also be used to lessen the impacts of noise or measurement errors in the data.

- ❑ **Multicollinearity:** When two or more variables are strongly correlated, there is multicollinearity in the data, which PCA can handle. PCA can lessen the impacts of multicollinearity on the analysis by identifying the most crucial features or components.

Disadvantages of PCA

- **Interpretability:** Although principal component analysis (PCA) is effective at reducing the dimensionality of data and spotting patterns, the resulting principal components are not always simple to understand or describe in terms of the original features.

- **Information loss:** PCA involves choosing a subset of the most crucial features or components in order to reduce the dimensionality of the data. While this can be helpful for streamlining the data and lowering noise, if crucial features are not included in the components chosen, information loss may also result.

Disadvantages of PCA

- **Outliers:** Because PCA is susceptible to anomalies in the data, the resulting principal components may be significantly impacted. The covariance matrix can be distorted by outliers, which can make it harder to identify the most crucial characteristics.

- **Scaling:** PCA makes the assumption that the data is scaled and centralized, which can be a drawback in some circumstances. The resulting principal components might not correctly depict the underlying patterns in the data if the data is not scaled properly.

- **Computing complexity:** For big datasets, it may be costly to compute the eigenvectors and eigenvalues of the covariance matrix. This may restrict PCA's ability to scale and render it useless for some uses.

Parul® University | NAAC GRADE A++

https://paruluniversity.ac.in/