

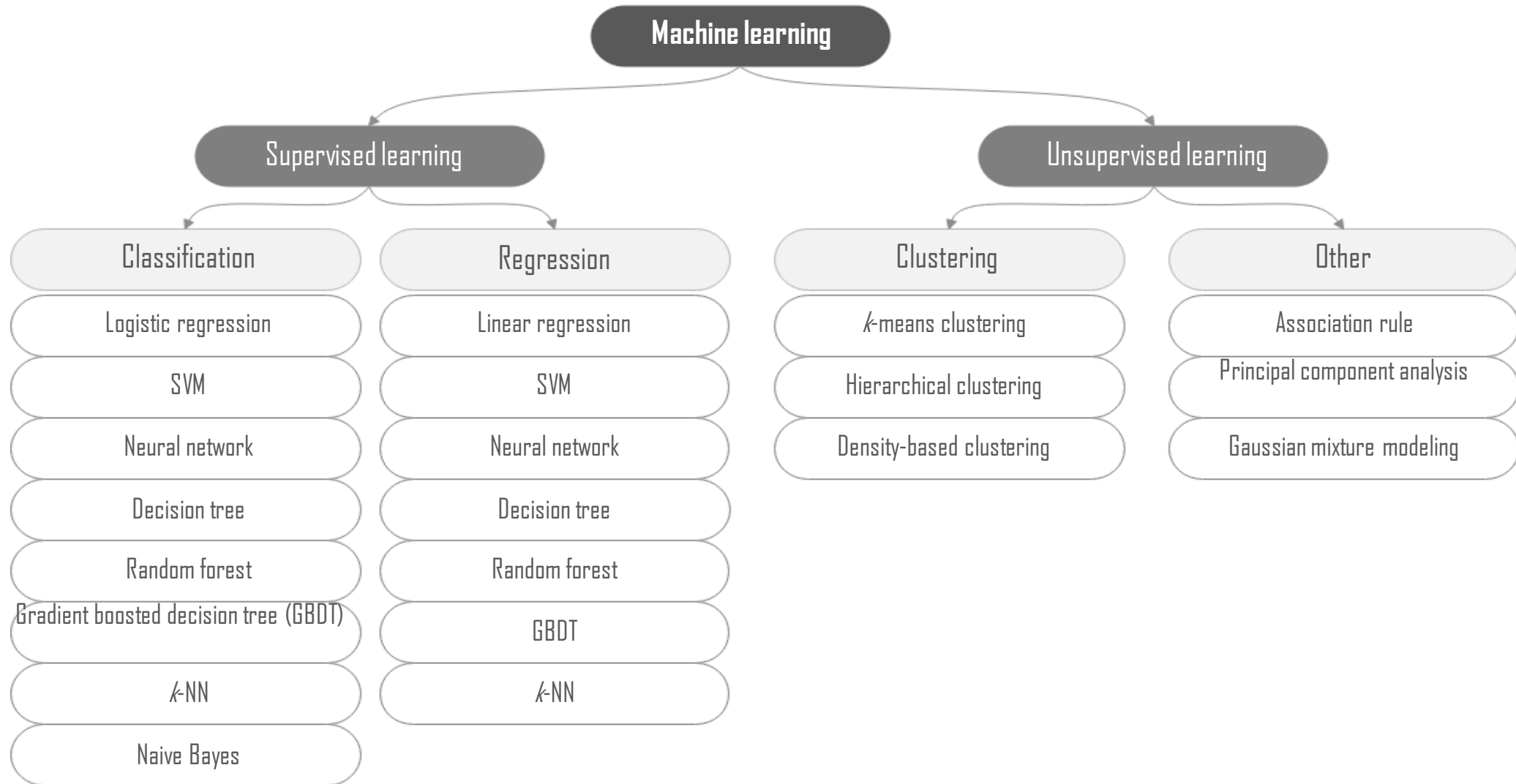
HCIA-AI Huawei

Eng.Eman Elgallad

Lecture 5- Unsupervised Learning

- Unsupervised Learning
 - Clustering
 - Association Rule
 - Dimensionality reduction.

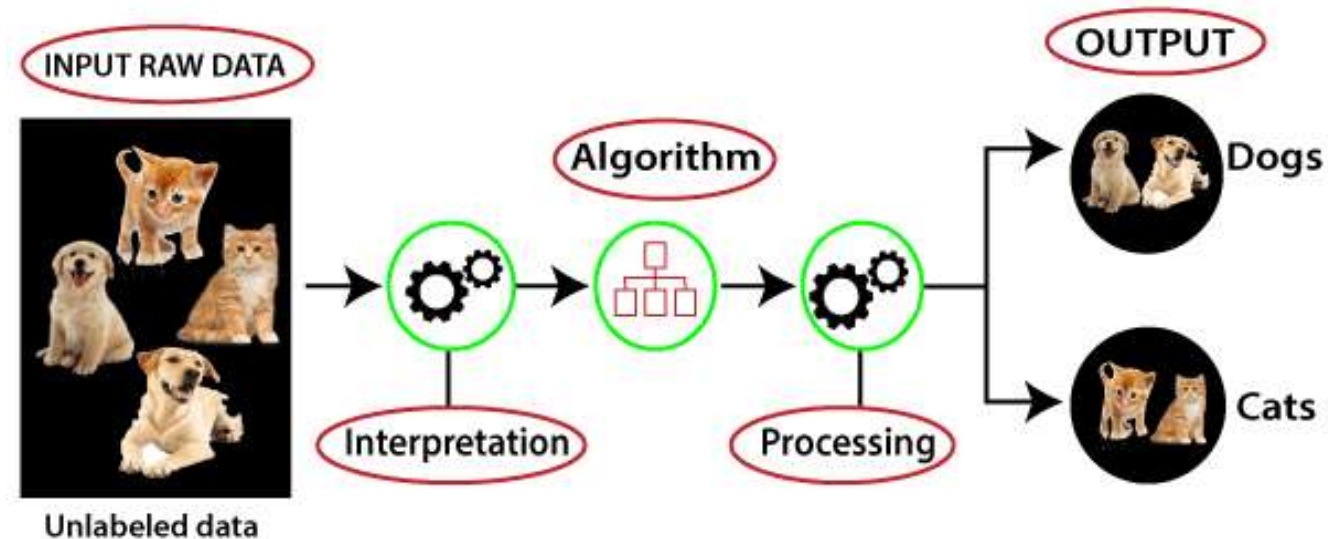
Machine Learning Algorithm Overview



Unsupervised Learning

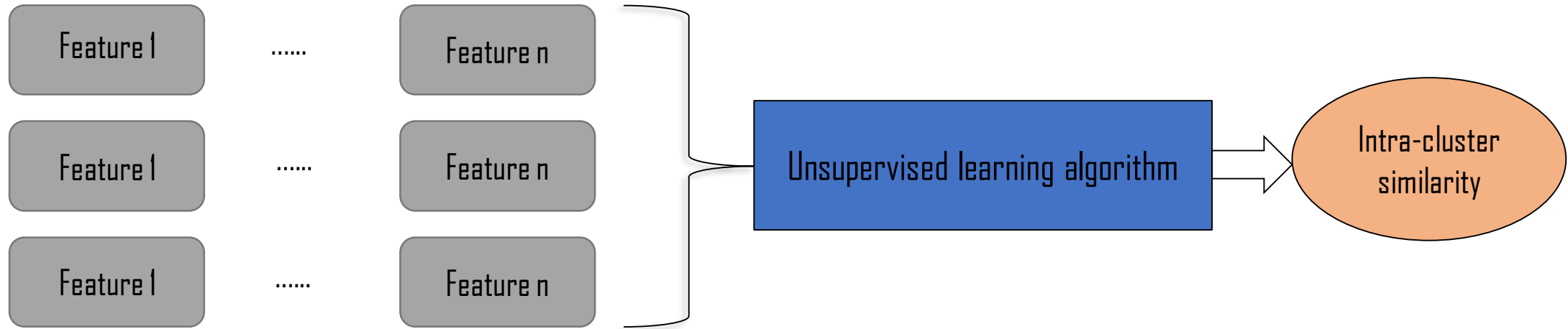
As the name suggests, unsupervised learning is a machine learning technique in which models are not supervised using training dataset. Instead, models itself find the hidden patterns and insights from the given data. It can be compared to learning which takes place in the human brain while learning new things. It can be defined as:

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

Data features



Monthly Sales Volume	Product	Sale Duration	Category
1000-2000	Badminton racket	6:00-12:00	Cluster 1
500-1000	Basketball	18:00-24:00	Cluster 2
1000-2000	Game console	00:00-6:00	

Why Unsupervised Learning

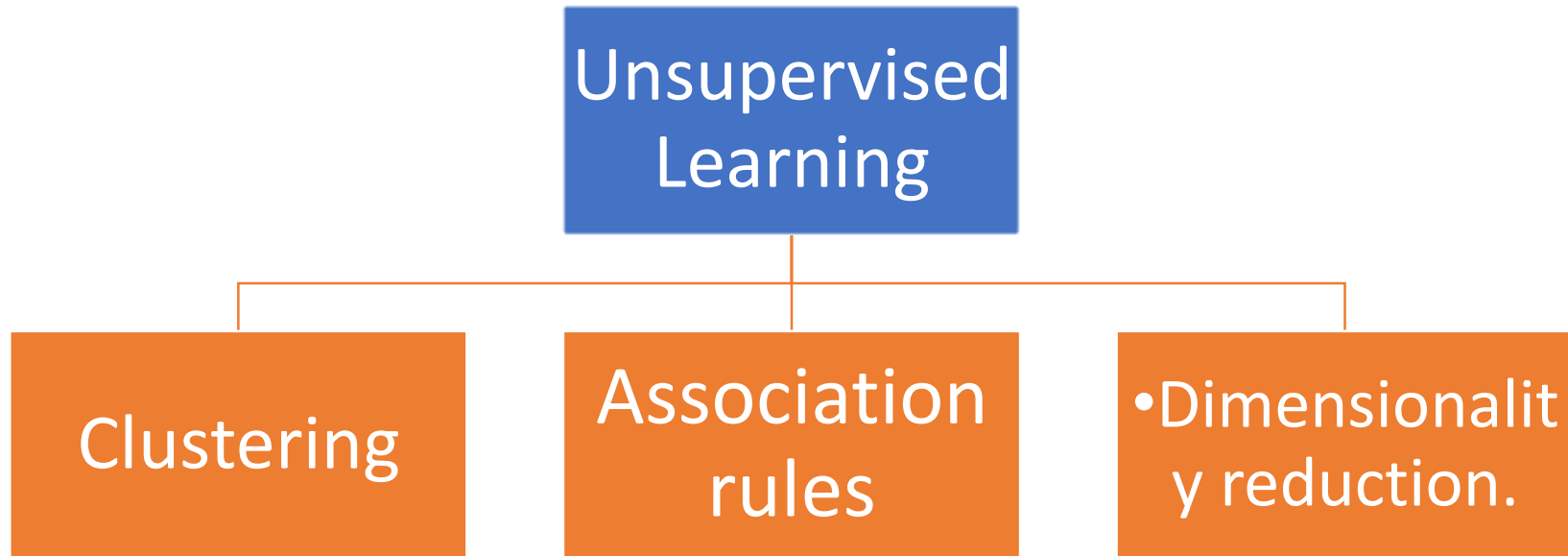
- Unsupervised Learning is helpful for finding useful insights from the data.
- It is much similar as a human learns to think by their own experiences, which makes it closer to the real AI.
- It works on unlabeled and uncategorized data which makes it more important.
- In real-world, we do not always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

Most common uses of Clustering:

- Natural language processing (NLP). Google News is known to leverage unsupervised learning to categorize articles based on the same story from various news outlets. For instance, the results of the football transfer window can all be categorized under football.
- Image and video analysis. Visual Perception tasks such as object recognition leverage unsupervised learning.
- Anomaly detection. Unsupervised learning is used to identify data points, events, and/or observations that deviate from a dataset's normal behavior.
- Customer segmentation. Interesting buyer persona profiles can be created using unsupervised learning. This helps businesses to understand their customers' common traits and purchasing habits, thus, enabling them to align their products more accordingly.
- Recommendation Engines. Past purchase behavior coupled with unsupervised learning can be used to help businesses discover data trends that they could use to develop effective cross-selling strategies.

Types of Unsupervised Learning Algorithms:

- Clustering
- Association rules
- Dimensionality reduction.



Unsupervised Learning - Clustering

- Clustering is the type of Unsupervised Learning where we find hidden patterns in the data based on their similarities or differences. These patterns can relate to the shape, size, or color and are used to group data items or create clusters.

Types of Clustering :

- **Exclusive clustering:** A hard cluster Data is grouped such that a single data point exclusively belongs to one cluster.
- **Overlapping clustering:** A soft cluster in which a single data point may belong to multiple clusters with varying degrees of membership.
- **Hierarchical clustering:** A type of clustering in which groups are created such that similar instances are within the same group and different objects are in other groups.
- **Probalistic clustering:** Clusters are created using probability distribution.

Unsupervised Learning - Clustering

- **Example:**

Clustering uses a clustering model to classify samples in a dataset into several categories based on similarity.

- Defining fish of the same species.
- Recommending movies for users.



Unsupervised Learning - Clustering

- **Another 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.



Types of clustering algorithms:

- **Centroid-based** is the one we mostly deal with where each data point is assigned to the cluster where it has the closest distance to its centroid. However, such algorithms need predefinition of the number of clusters which means that it needs prior information, before applying the algorithm, about the clusters that might exist in such data. This algorithm runs iteratively to find local optima and it is fast and efficient (K-means clustering)
- **Hierarchical-based** is a cluster ranking system where we start either from a single cluster and start getting subclusters or starting from separate clusters and start combining them where number of clusters can then be determined using statistical methods. No prior information is needed about number of clusters.
- **Density-based** is concerned with areas of high concentration of data points surrounded by areas of low concentration. The places with dense data points are clustered and while with low data points are considered outliers. (DBSCAN)

K-means Clustering

K-means clustering is a centroid-based clustering where it works in iterative manner to find the closest centroid to each data point to assign it to while keeping iterating until reaching the best centroids representing the data.

[Imagine you're opening a small book store.](#) You have a stack of different books, and 3 bookshelves.

Your goal is placing similar books in one shelf. What you would do, is pick up 3 books, one for each shelf in order to set a theme for every shelf. These books will now dictate which of the remaining books will go in which shelf. Every time you pick a new book up from the stack, you would compare it with those first 3 books, and place this new book on the shelf that has similar books. You would repeat this process until all the books have been placed. Once you're done, you might notice that changing the number of bookshelves, and picking up different initial books for those shelves (changing the theme for each shelf) would increase how well you've grouped the books. So, you repeat the process iteratively in hopes of a better outcome.

[K-Means algorithm works something just like this.](#)

K-means Clustering

K-means Clustering algorithm steps:

Choose the number of clusters K :

This number is based on the prior knowledge of the data and the definition of your problem.

Initialize Cluster Centroids :

Select K points at random from the data points to let them be the centroids (centers).

⇒ Iterate over the following two steps:

Assignment step:

Assign each data points to the closest center of the k centers which is done by computing distance (Euclidean, Manhattan, ..) to all centers and choose the least one.

Update step:

For all data points related to same cluster, compute the new cluster center which is done by calculating the mean of each cluster data points.

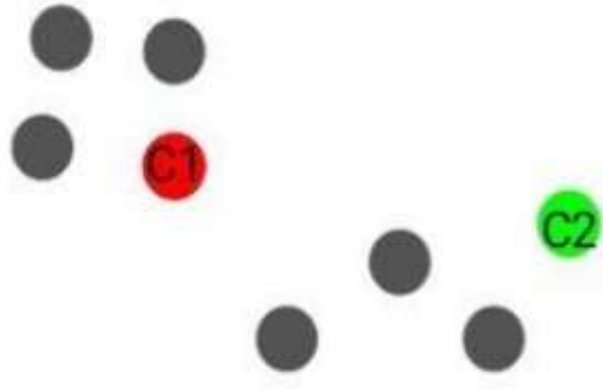
K-means Clustering

After each iteration, the centroids will move slowly so that the total distance from each point to the closest centroid gets lower, and this keeps performed until convergence where no changes takes place to the data points assignment hence the centroids won't change which guarantees reaching a local optimum cluster assignment.

Example:

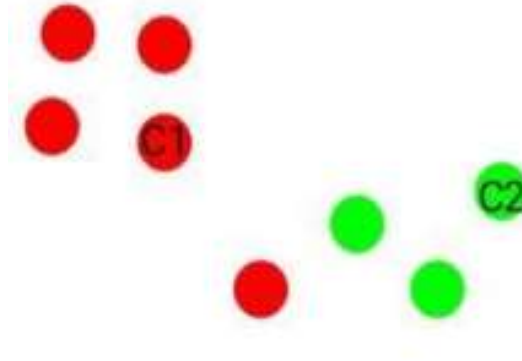
Step1: We choose $K=2$.

Step2: We take K random points from the data points to be the centroids as shown:

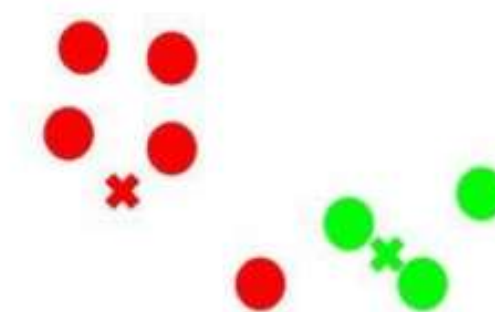


K-means Clustering

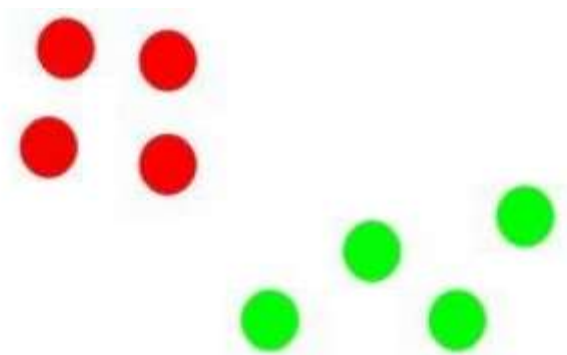
Step3: Assign all data points to the closest cluster centroid as shown below:



Step4: Recompute the cluster centroids once again:



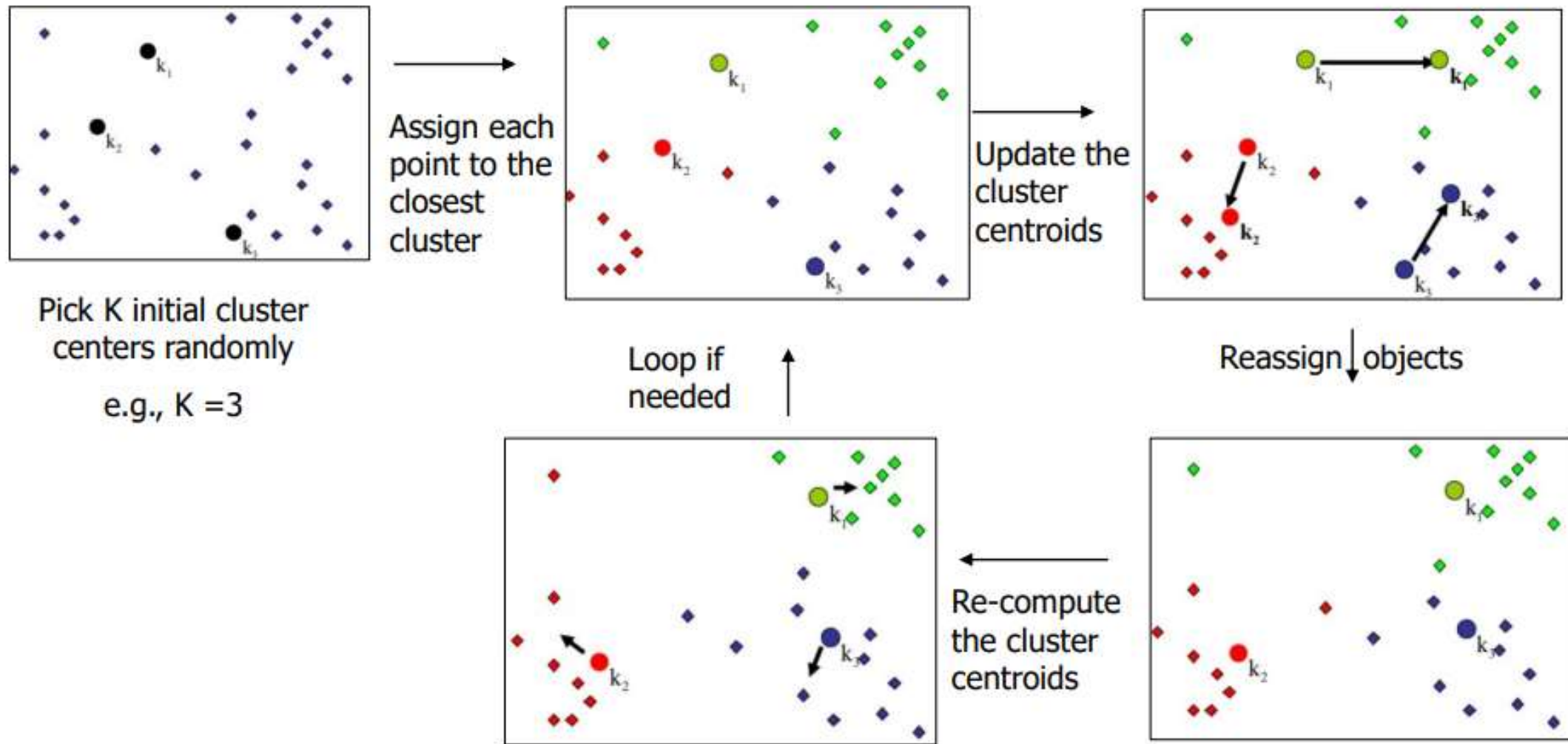
Repeat step 3 and step 4:



The step of computing the centroid and assigning all the points to the cluster based on their distance from the centroid is a single iteration.

But wait – when should we stop this process? It can't run till eternity, right?

K-means Steps- Summary:



- Step 0:
 - Select k objects as initial centroids.
- Step 1: (Assignment)
 - For each object, compute distances to k centroids.
 - Assign each object to the cluster to which it is closest based on mean value.
- Step 2: (New Centroids)
 - Update the centroid (i.e., mean value) for each cluster.
 - Assign each object to the cluster with the nearest centroid.
- Step 3: (Converge)
 - Stop when the assignment does not change
 - Otherwise, repeat Step 1.

K-means Clustering

What is the stopping criteria of the iterations done by K-means clustering?

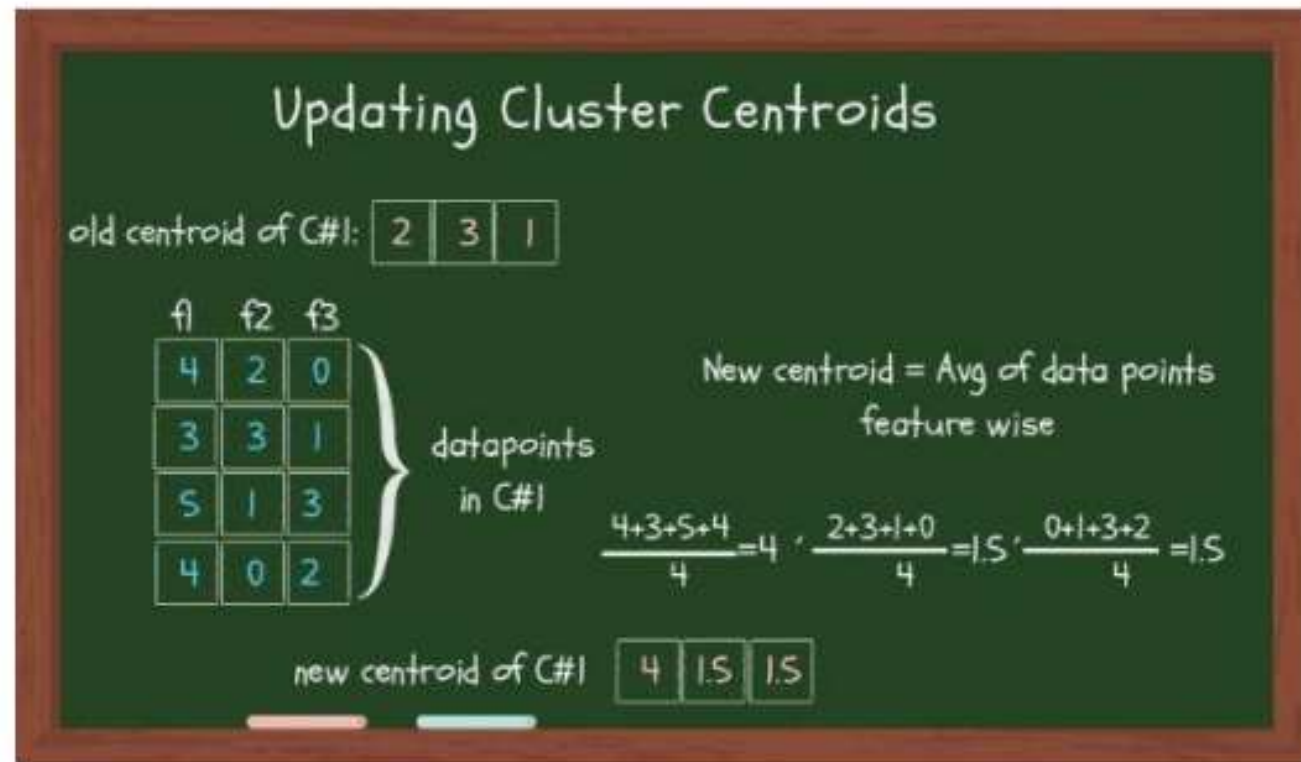
Since step 3 and 4 are performed iteratively, we should have stopping criteria for these iterations noting that setting this stopping criteria doesn't return necessarily the best clusters.

- Data points remains assigned to the same cluster. (time consuming)
- Centroids remains the same with no change. (More time consuming)
- The distance from each data point to its centroid is lower than a certain specified threshold that has been set.
- Maximum number of iterations has been set noting that low number of iterations can lead to poor results while high number of iterations can be time consuming.

You can choose a combination of stopping criteria as choosing whether data points remains assigned to same cluster or maximum number of iterations reached. Whoever comes first.

How is the new cluster centroid (update step) is computed?

As we mentioned before that we get the new cluster centroid by getting the mean of all data points belonging to the same cluster. This step is done feature wise, where the new centroid feature values is the mean of each feature set of these data points as shown below:



K-means Clustering

What if we don't have a prior knowledge of number of clusters? Are there any way of estimating the best number of clusters?

Before answering this question, let us define a terminology called inertia.

Inertia is the sum of distances of all points in a cluster to the centroid of this cluster which is summed up over all the clusters. From this definition, we can say that as inertia gets lower, we are having a better clustering algorithm.

The maximum possible number of clusters will be equivalent to the number of data points in the dataset where each data point is considered as cluster on its own.

This would give us the lowest inertia possible as all distances are zeros so the summation will be 0. Similarly we can say that making all the data as one cluster will result in the highest inertia possible. Hence we should find a way to choose the optimal reasonable value of K based on the inertia, This raises the Elbow method.

Note: The inertia ranges from 0 to infinity value.

K-Means

Advantages of K-Means Clustering

- Simple to understand and implement.
- Guarantees convergence.
- It is much faster and computationally efficient in a comparison with other clustering technique as hierarchical clustering.
- More suitable with large datasets than other clustering algorithms.

Disadvantages of K-Means Clustering

- Needs pre-definition of number of clusters which needs prior info about the data.
- The final result is dependable on the initial values of cluster centroids.
- Time consuming if initialization was bad.
- As number of features increases, the distance metric would become less accurate due to curse of dimensionality explained before.
- Handles numerical features only which can be impractical in many cases.
- Sensitive to scales so standardization or normalization would lead to different final result.

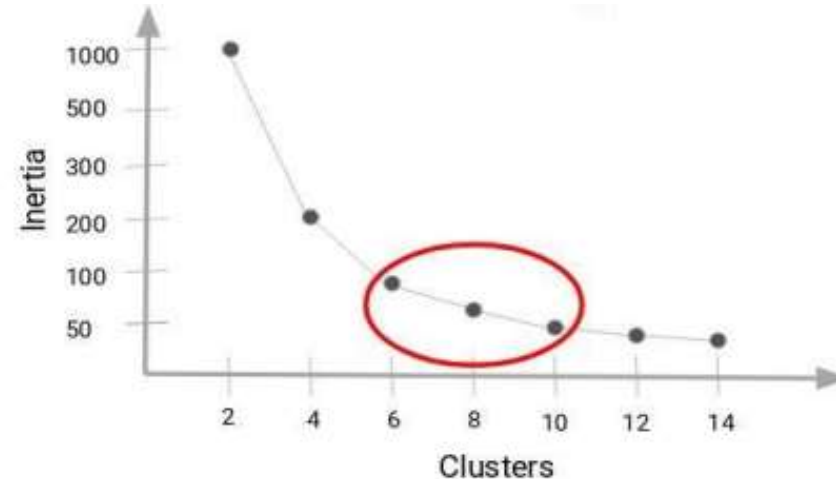
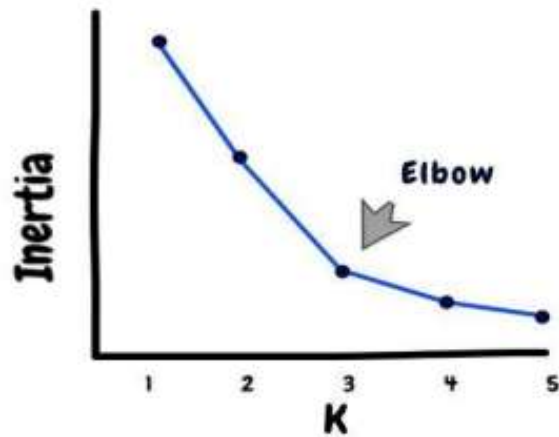
Choosing the Number of Clusters:

Choosing the optimal number of clusters is a big task. we will apply a methodology like hyper-parameter tuning by trying out several K values from 1 to n and for each value we calculate the inertia corresponding to this value.

The optimal value is one of the values at the elbow of the curve.

→ The performance of the K-Means clustering algorithm depends upon highly efficient clusters that it forms.

→ There are some different ways to find the optimal number of clusters, but here we are discussing the most appropriate method to find the number of clusters or value of K, Elbow Method.



Elbow Method:

The Elbow Method is one of the most popular ways to find the optimal number of clusters.

- This method uses the concept of WCSS value.
- WCSS stands for Within Cluster Sum of Squares, which defines the total variations within a cluster.
- The formula to calculate the value of WCSS (for 3 clusters):

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

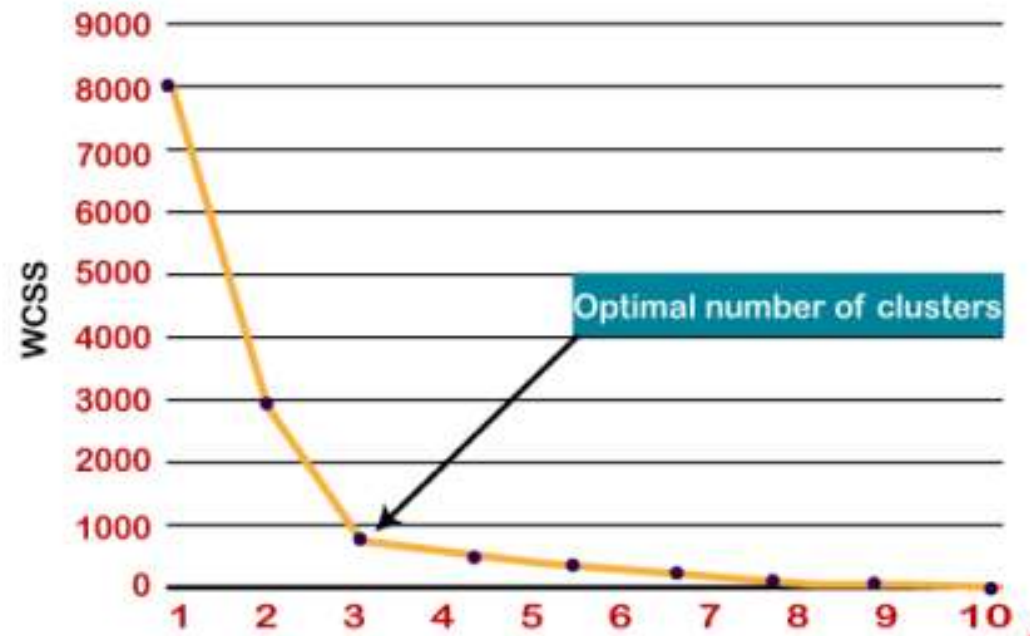
$$\sum_{P_i \text{ in Cluster } 1} \text{distance}(P_i, C_1)^2$$

It is the sum of the square of the distances between each data point and its centroid within a cluster 1. And the same for the other two terms.

Elbow Method:

The Elbow Method follows the below steps:

- It executes the K-means clustering on a given dataset for different K values (ranges from 1-10).
- For each value of K, calculates the WCSS value.
- Plots a curve between calculated WCSS values and the number of clusters K.
- The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value of K.



Unsupervised Learning – Association Rules

- This type of unsupervised machine learning takes a rule-based approach to discovering interesting relationships between features in a given dataset. It works by using a measure of interest to identify strong rules found within a dataset.
- We typically see association rule mining used for market basket analysis: this is a data mining technique retailers use to gain a better understanding of customer purchasing patterns based on the relationships between various products.
- The most widely used algorithm for association rule learning is the Apriori algorithm. However, other algorithms are used for this type of unsupervised learning, such as the Eclat and FP-growth algorithms.

Unsupervised Learning – Dimensionality Reduction

- Popular algorithms used for dimensionality reduction include principal component analysis (PCA) and Singular Value Decomposition (SVD). These algorithms seek to transform data from high-dimensional spaces to low-dimensional spaces without compromising meaningful properties in the original data. These techniques are typically deployed during exploratory data analysis (EDA) or data processing to prepare the data for modeling.
- It's helpful to reduce the dimensionality of a dataset during EDA to help visualize data: this is because visualizing data in more than three dimensions is difficult. From a data processing perspective, reducing the dimensionality of the data simplifies the modeling problem.
- When more input features are being fed into the model, the model must learn a more complex approximation function. This phenomenon can be summed up by a saying called the “curse of dimensionality.”

Unsupervised Learning:

Can Unsupervised Learning be used to improve supervised machine learning algorithms?

YES, we can use unsupervised techniques as clustering to enhance our supervised models by clustering data points of datasets to separate clusters and use the cluster labels as independent variables in the supervised machine learning algorithm.

When shouldn't you use Unsupervised Learning?

If you both don't know any prior information about the number of clusters you want to segment your data into and the dataset is very large as well that can lead to a very high computational power

Supervised VS Unsupervised Learning:

Unsupervised learning and supervised learning are frequently discussed together. Unlike unsupervised learning algorithms, supervised learning algorithms use labeled data. From that data, it either predicts future outcomes or assigns data to specific categories based on the regression or classification problem that it is trying to solve.

While supervised learning algorithms tend to be more accurate than unsupervised learning models, they require upfront human intervention to label the data appropriately. However, these labelled datasets allow supervised learning algorithms to avoid computational complexity as they don't need a large training set to produce intended outcomes. Common regression and classification techniques are linear and logistic regression, naïve bayes, KNN algorithm, and random forest.

Semi-supervised learning occurs when only part of the given input data has been labelled. Unsupervised and semi-supervised learning can be more appealing alternatives as it can be time-consuming and costly to rely on domain expertise to label data appropriately for supervised learning.

