جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

قسم الانظمة الطبية الذكية

المرحلة الثالثة

Subject: Artificial Intelligence AII

Class: Third

Lecturers: Dr. Muneera Abed Hadi, M. Sc. Rouaa Safi , M. Sc. Ansam Ali

University of Information Technology
and Communications
College of Medical Informatics
Intelligent Medical Systems Department

Machine Learning Course

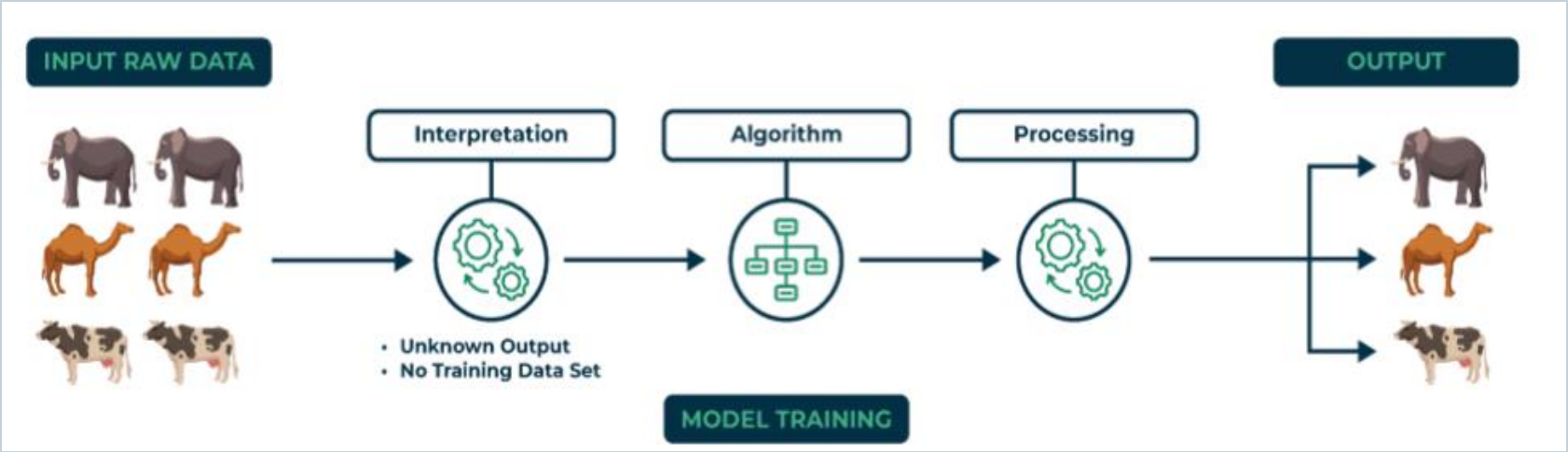
Lecture Five /P2

By: Dr. Muneera Abed Hmdi

What is Unsupervised Learning?

- ***Unsupervised learning*** is a branch of machine learning that deals with unlabeled data. Unlike supervised learning, where the data is labeled with a specific category or outcome, unsupervised learning algorithms are tasked with finding patterns and relationships within the data without any prior knowledge of the data's meaning.

- ***Unsupervised learning*** algorithms find hidden patterns and data without any human intervention, i.e., we don't give output to our model. The training model has only input parameter values and discovers the groups or patterns on its own.



How does unsupervised learning work?

- ***Unsupervised learning*** works by analyzing unlabeled data to identify patterns and relationships. The data is not labeled with any predefined categories or outcomes, so ***the algorithm must find these patterns and relationships on its own***.

- ***Dataset*** in Figure A is ***Mall data*** that contains information about its clients that subscribe to them. Once subscribed they are provided a ***membership card***, and the mall has complete information about the customer and his/her every ***purchase***.

- Now using this data and unsupervised learning techniques, ***the mall can easily group clients based on the parameters we are feeding in.***

CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
1	Male	19	15	39
2	Male	21	15	81
3	Female	20	16	6
4	Female	23	16	77
5	Female	31	17	40
6	Female	22	17	76
7	Female	35	18	6
8	Female	23	18	94
9	Male	64	19	3
10	Female	30	19	72
11	Male	67	19	14
12	Female	35	19	99
13	Female	58	20	15
14	Female	24	20	77
15	Male	37	20	13
16	Male	22	20	79
17	Female	35	21	35

Figure A

Getting Started with Classification

- ***The input to the unsupervised learning models is as follows:***

- ***Unstructured data:*** Is data that does not have a predefined model or organized format. It is typically qualitative and difficult to store in traditional relational databases. May contain noisy(meaningless) data, missing values, or unknown data. For an example:

1. Text-Based Data

- Emails (content, attachments, metadata)
- Chat messages (e.g., WhatsApp, Telegram)
- Social media posts (tweets, Facebook posts, LinkedIn updates)
- Reports, articles, and research papers Customer reviews and feedback

2. Multimedia Data

- Images (JPEG, PNG, TIFF)
- Videos (MP4, AVI, MOV)
- Audio recordings (MP3, WAV, podcasts)
- Handwritten notes and scanned documents

Getting Started with Classification

3. Healthcare Data

- MRI and X-ray images
- DNA sequencing data
- Doctor's handwritten prescriptions and case notes

4. Business and Legal Documents

- Contracts and agreements (PDFs, scanned documents)
- Financial reports (non-tabular sections)
- Insurance claims and policies

Getting Started with Classification

- ***The input to the unsupervised learning models is as follows:***
  - ***Unlabeled data:*** Data only contains a value for input parameters, there is no targeted value(output). It is easy to collect as compared to the labeled one in the supervised approach.

1. Text-Based Unlabeled Data

- Raw customer reviews (without sentiment labels like "positive" or "negative")
- Unclassified emails (without spam or non-spam labels)
- Social media posts (without topic or sentiment tags)
- News articles (without categorization like "sports" or "politics")

2. Image-Based Unlabeled Data

- Photos from a camera roll (without object or face labels)
- Medical X-rays (without disease classification)
- Satellite images (without land-use labels)
- Security camera footage (without identification of people or actions)

Understanding K-means Clustering Algorithms

- ***There are mainly 3 types of Algorithms which are used for Unsupervised learning:***

  1. Clustering
  2. Association Rule Learning
  3. Dimensionality Reduction

Some Common Clustering Algorithms

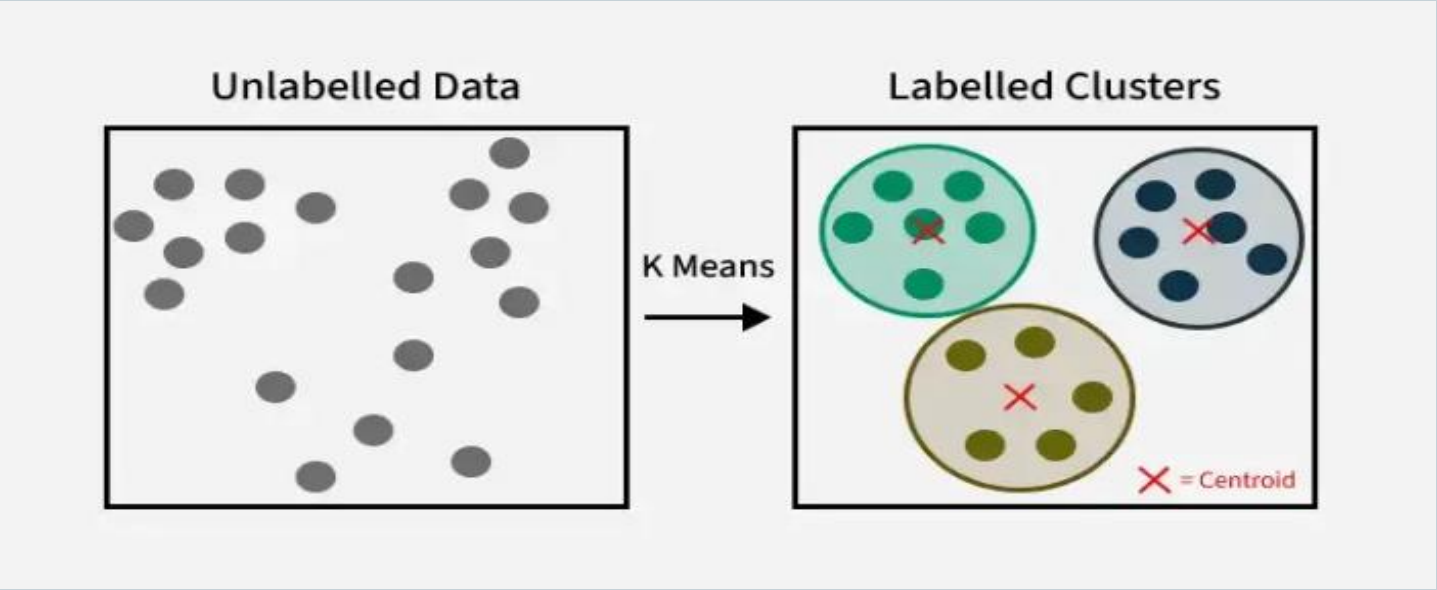
- ***Clustering Algorithms***
  - ***K-means Clustering:*** Groups data into K clusters based on how close the points are to each other.
  - ***Hierarchical Clustering:*** Creates clusters by building a tree step-by-step, either merging or splitting groups.
  - ***Density-Based Clustering (DBSCAN):*** Finds clusters in dense areas and treats scattered points as noise.
  - ***Mean-Shift Clustering:*** Discovers clusters by moving points toward the most crowded areas.
  - ***Spectral Clustering:*** Groups data by analyzing connections between points using graphs.

Understanding K-means Clustering Algorithms

- ***The most popular Clustering Algorithm***
  - ***K-means clustering*** is a technique used to organize data into groups based on their similarity. For example, online store uses ***K-Means*** to group customers based on ***purchase frequency and spending*** creating segments like Budget Shoppers, Frequent Buyers and Big Spenders for personalized marketing.
  - The algorithm works by first randomly picking some ***central points*** called ***centroids*** and each data point is then assigned to the closest centroid forming a cluster. After all the points are assigned to a cluster the centroids are updated by finding the average position of the points in each cluster. This process repeats until the centroids stop changing forming clusters. The goal of clustering is to divide the data points into clusters so that similar data points belong to same group.

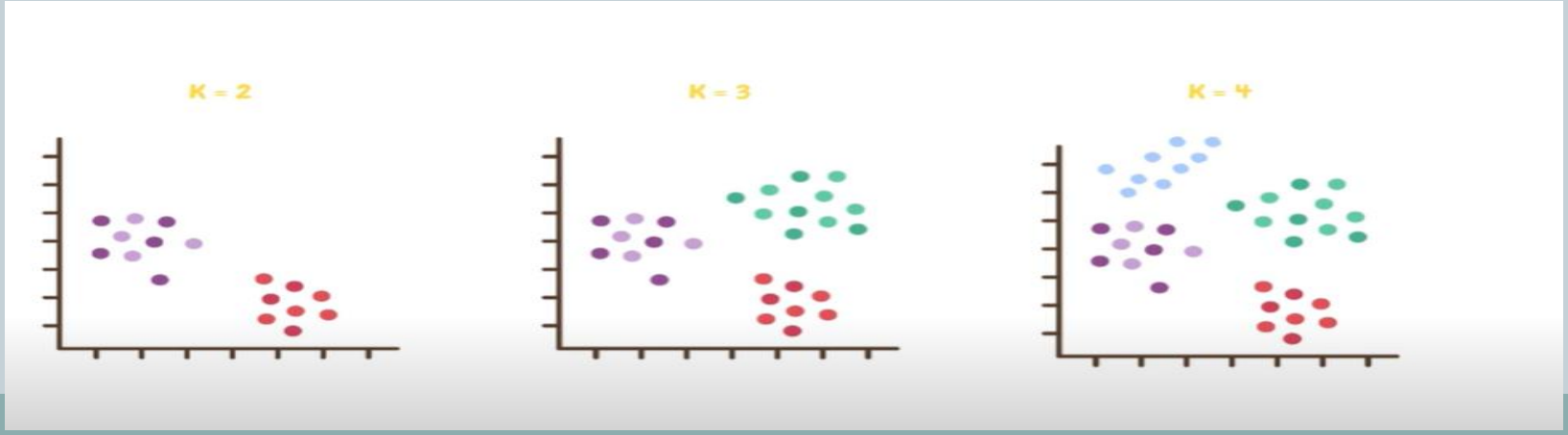
How k-means clustering works?

- We are given a data set of items with certain features and values for these features. The task is to categorize those items into groups. To achieve this, we will use the K-means algorithm. ‘K’ in the name of the algorithm represents the number of groups/clusters we want to classify our items into.



Understanding K-means Clustering Algorithms

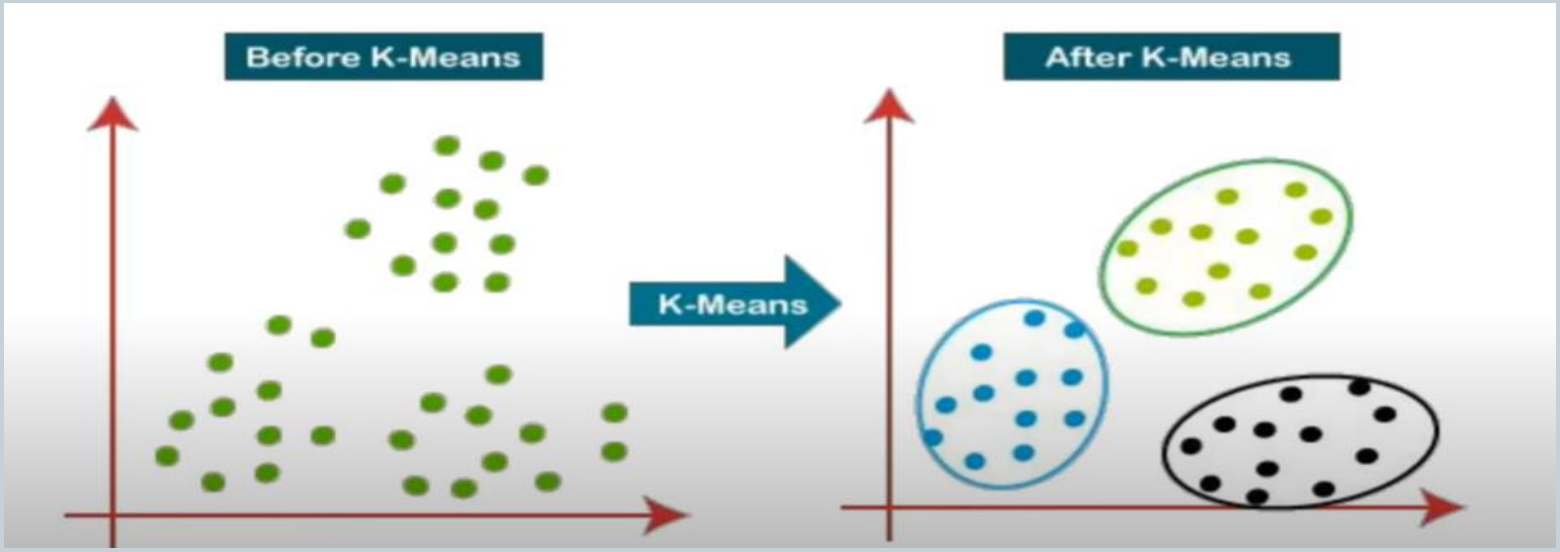
- The algorithm will categorize the items into k groups or clusters of similarity. To calculate that similarity, we will use the Euclidean distance as a measurement. The algorithm works as follows:
  - First, we randomly initialize k points, called means or cluster centroids.
  - We categorize each item to its closest mean, and we update the mean's coordinates, which are the averages of the items categorized in that cluster so far.
  - We repeat the process for a given number of iterations and at the end, we have our clusters.



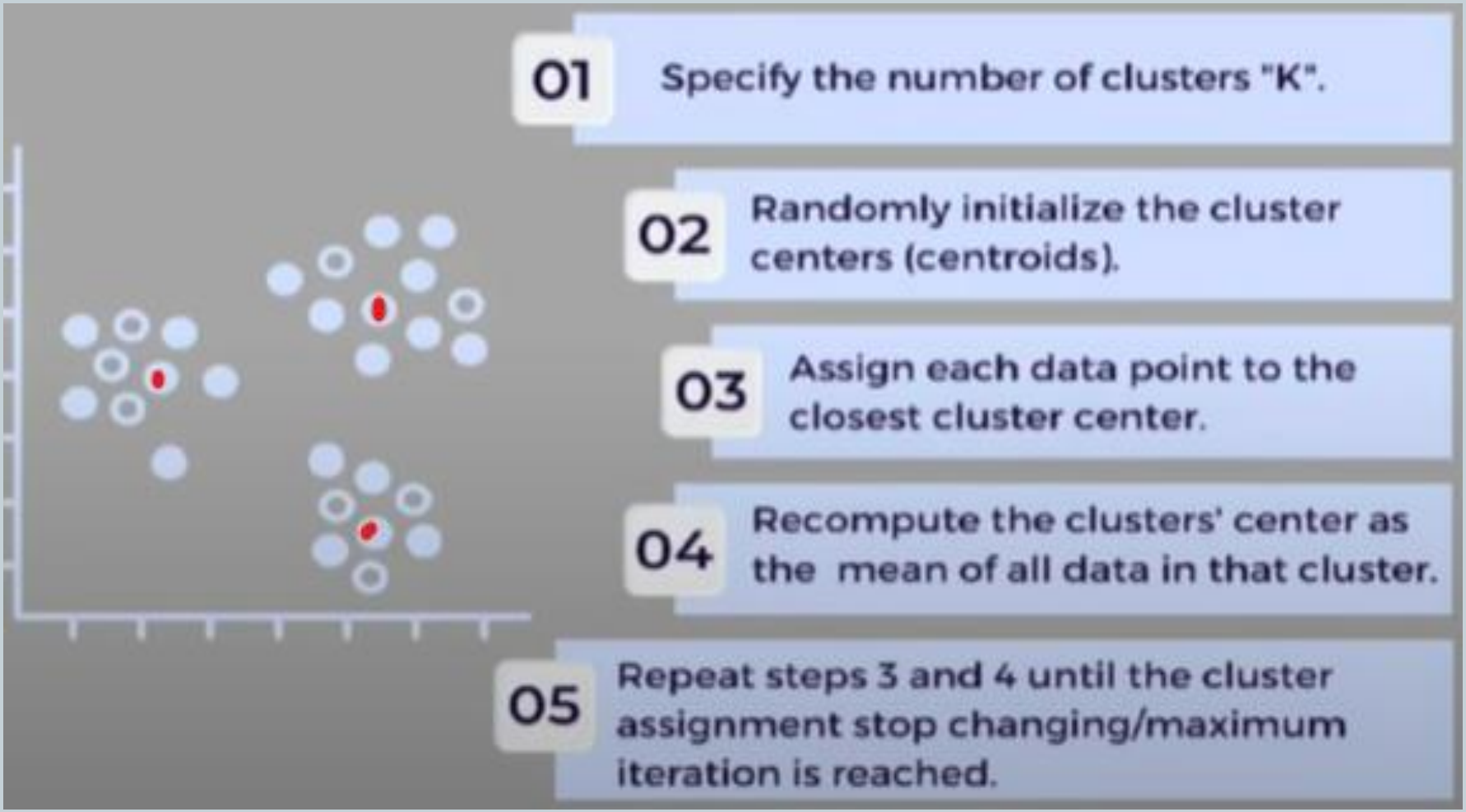
Understanding K-means Clustering Algorithms

- The "***points***" mentioned previously are called means because they are the mean values of the items categorized in them. To initialize these means, we have a lot of options. An intuitive method is to initialize the means at random items in the data set. Another method is to initialize the means at random values between the boundaries of the data set.

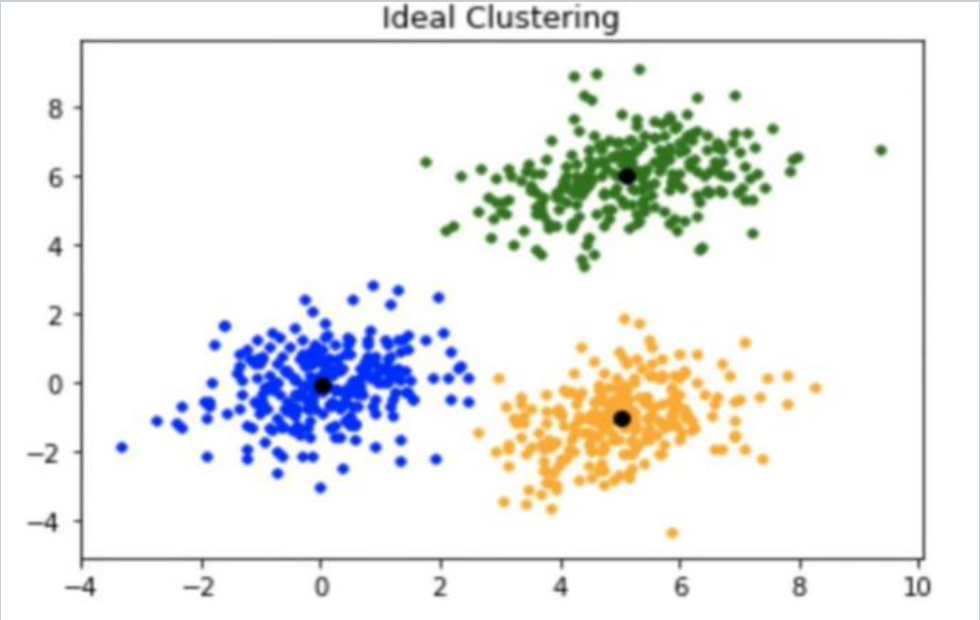
- ***New Clusters*** is the average points awarded for each cluster.



Main steps of K-means Clustering Algorithms



Understanding K-means Clustering Algorithms



- ***Euclidean Distance*** is defined as the distance between two points in Euclidean space. To find the distance between two points, the length of the line segment that connects the two points should be measured.

Euclidean Distance Formula

- The Euclidean distance formula helps to find the distance of a line segment. Let us assume two points, such as ***(x1, y1)*** and ***(x2, y2)*** in the two-dimensional coordinate plane.

- Thus, the ***Euclidean distance formula*** is given by:

$$d = \sqrt{[(x2 - x1)2 + (y2 - y1)2]}$$

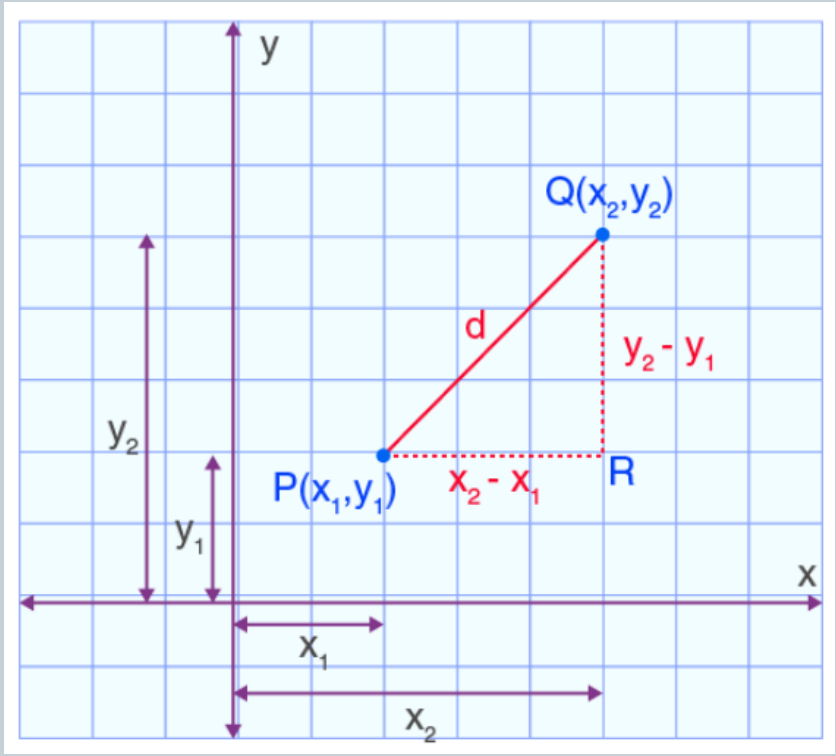
d: is the Euclidean distance

(x1, y1): is the coordinate of the first point

(x2, y2): is the coordinate of the second point.

Euclidean Distance Formula Derivation

- To derive the formula for ***Euclidean distance***, let us consider two points, say ***P(x1, y2) and Q(x2, y2)*** and ***d is the distance*** between the two points. Now, join the points ***P and Q using a line***. To derive the Euclidean distance formula, let us construct a ***right triangle***, whose ***hypotenuse is PQ***. Now, draw the ***horizontal line*** and ***vertical line*** from P and Q which meet at the point R as shown in the figure given below



Euclidean Distance Formula Derivation

- ***Now, we have to apply Pythagoras theorem to the triangle PQR to find the distance between two points.***

  - Using Pythagoras theorem,

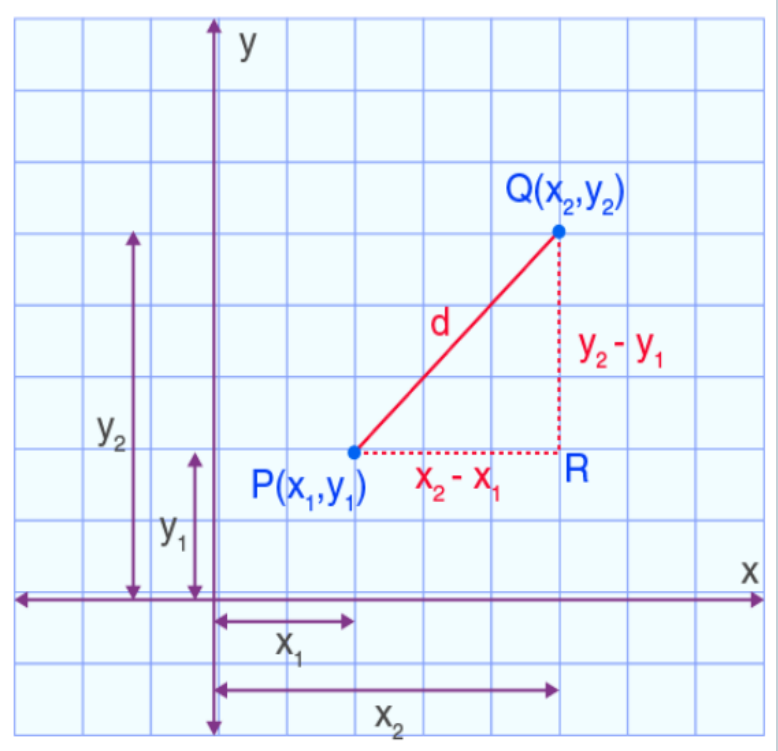
  - Hypotenuse2 = Base2 + Perpendicular2

  - PQ2 = PR2 + QR2

  - Therefore, d2 = (x2 – x1)2 + (y2 – y1)2

- ***Now, take square root on both sides of equation, we get***

  d =√[(x2 – x1)2 + (y2 – y1)2]



Example 1

- ***Determine the Euclidean distance between two points (a, b) and (-a, -b)*** ***- Solution:*** Let the point P be (a, b) and Q be (-a, -b)

  i.e. P(a, b) = (x1, y1) and Q(-a, -b) = (x2, y2)

  We know that the Euclidean distance formula is,

  Euclidean distance, $d = \sqrt{[(x2 - x1)2 + (y2 - y1)2]}$

- Now, substitute the values in the formula, we get

$d = \sqrt{[(-a - a)2 + (-b - b)2]}$

$d = \sqrt{[(-2a)2 + (-2b)2]}$

$d = \sqrt{[4a2 + 4b2]}$

$d = \sqrt{4(a2+b2)}$

$d = 2\sqrt{(a2+b2)}$.

- ***Hence, the distance between two points (a, b) and (-a, -b) is $2\sqrt{(a2+b2)}$.***

Example 2

- ***Find the distance between two points P(0, 4) and Q(6, 2)***

  - ***Solution:*** Given P(0, 4) = (x1, y1) and Q(6, 2) = (x2, y2)

  - ***The distance between the point PQ is:***

$$PQ = \sqrt{[(x2 - x1)2 + (y2 - y1)2]}$$

$$PQ = \sqrt{[(6 - 0)2 + (2 - 4)2]}$$

$$PQ = \sqrt{[(6)2 + (-2)2]}$$

$$PQ = \sqrt{(36+4)}$$

$$PQ = \sqrt{40}$$

$$PQ = 2\sqrt{10}.$$

- The prime factorization of ***40 is 2 × 2 × 2 × 5***. If it is written in the radical form (i.e.) ***$\sqrt{2}\times\sqrt{2}\times\sqrt{2}\times\sqrt{5}$***, we get the simplest radical form of the square root of 40. (i.e.) ***$2\sqrt{10}$***.

- ***Therefore, the distance between two points P(0,4) and Q(6, 2) is $2\sqrt{10}$.***

Implementation of K-Means Clustering in Python

- ***We will use blobs datasets and show how clusters are made:***
  - Step 1: Importing the necessary libraries
    - We are importing ***Numpy*** for statistical computations, ***Matplotlib*** to plot the graph, and ***make_blobs*** from ***sklearn.datasets***.

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
```

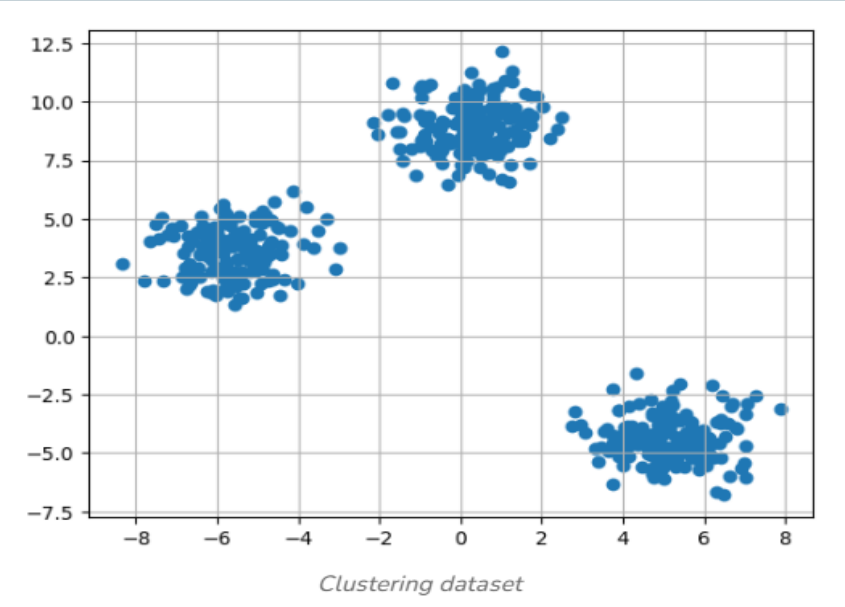
  - Step 2: Create the custom dataset with make_blobs and plot it

```
X,y = make_blobs(n_samples = 500,n_features = 2,centers = 3,random_state = 23)

fig = plt.figure(0)
plt.grid(True)
plt.scatter(X[:,0],X[:,1])
plt.show()
```

Output

- Step 2: Output of the creation of the custom dataset with make_blobs and plot it



- random_state is a parameter in train_test_split that controls the random number generator used to shuffle the data before splitting it. It ensures that the same randomization is used each time you run the code, resulting in the same splits of the data

Implementation of K-Means Clustering in Python

- Step 3: Initialize the random centroids
  - The code initializes three clusters for K-means clustering. It sets a random seed and generates random cluster centers within a specified range and creates an empty list of points for each cluster.

```
k = 3

clusters = {}
np.random.seed(23)

for idx in range(k):
    center = 2*(2*np.random.random((X.shape[1],))-1)
    points = []
    cluster = {
        'center' : center,
        'points' : []
    }

    clusters[idx] = cluster

clusters
```

Output

- Step 3: Output of the Initialize the random centroids

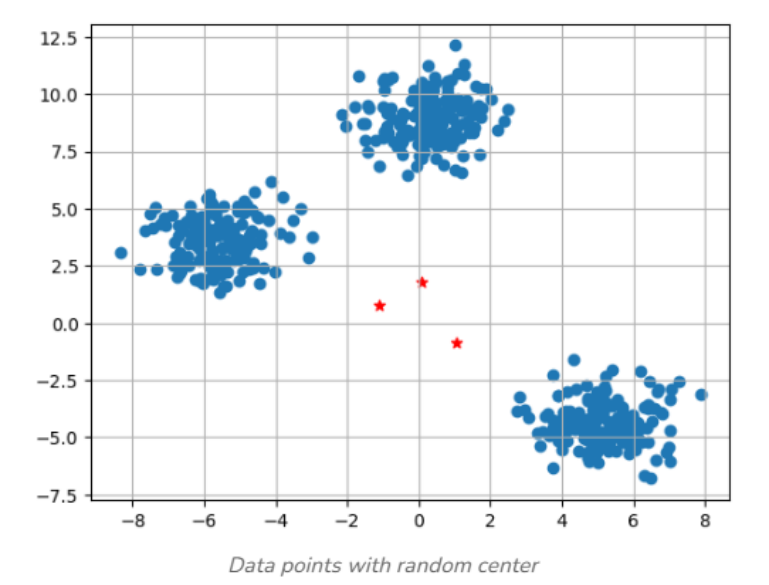
```
{0: {'center': array([0.06919154, 1.78785042]), 'points': []},
 1: {'center': array([ 1.06183904, -0.87041662]), 'points': []},
 2: {'center': array([-1.11581855,  0.74488834]), 'points': []}}
```

Implementation of K-Means Clustering in Python

- Step 4: Plot the random initialize center with data points

```
plt.scatter(X[:,0],X[:,1])
plt.grid(True)
for i in clusters:
    center = clusters[i]['center']
    plt.scatter(center[0],center[1],marker = '*',c = 'red')
plt.show()
```

Output



Data points with random center

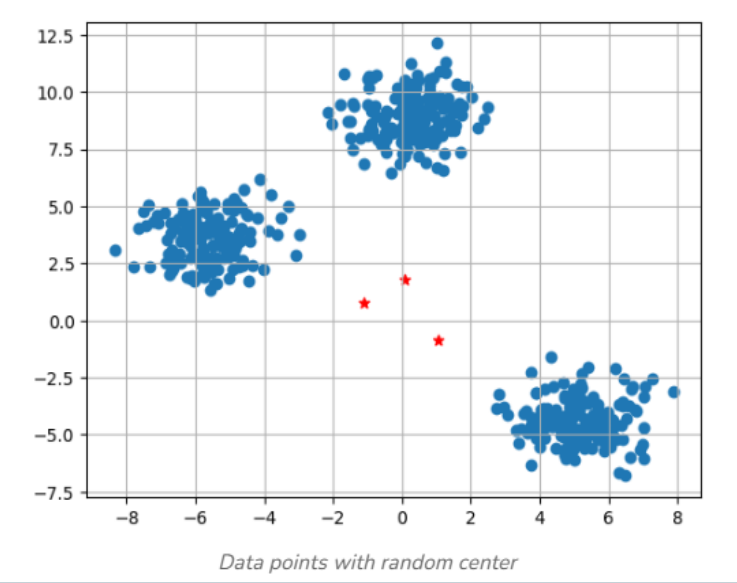
Implementation of K-Means Clustering in Python

- Step 4: Plot the random initialize center with data points

```
plt.scatter(X[:,0],X[:,1])
plt.grid(True)
for i in clusters:
    center = clusters[i]['center']
    plt.scatter(center[0],center[1],marker = '*',c = 'red')
plt.show()
```

Output

- ***The plot displays a scatter plot of data points (X[:,0], X[:,1]) with grid lines. It also marks the initial cluster centers (red stars) generated for K-means clustering.***



Implementation of K-Means Clustering in Python

- Step 5: Define Euclidean distance

```
def distance(p1,p2):
    return np.sqrt(np.sum((p1-p2)**2))
```

- Step 6: Create the function to Assign and Update the cluster center
  - This step assigns data points to the nearest cluster center, and the M-step updates cluster centers based on the mean of assigned points in K-means clustering.

Implementation of K-Means Clustering in Python

- Step 6: Create the function to Assign and Update the cluster center

```
def assign_clusters(X, clusters):
    for idx in range(X.shape[0]):
        dist = []

        curr_x = X[idx]

        for i in range(k):
            dis = distance(curr_x,clusters[i]['center'])
            dist.append(dis)
        curr_cluster = np.argmin(dist)
        clusters[curr_cluster]['points'].append(curr_x)
    return clusters

def update_clusters(X, clusters):
    for i in range(k):
        points = np.array(clusters[i]['points'])
        if points.shape[0] > 0:
            new_center = points.mean(axis =0)
            clusters[i]['center'] = new_center

            clusters[i]['points'] = []
    return clusters
```

Implementation of K-Means Clustering in Python

- Step 7: Create the function to Predict the cluster for the datapoints

```
def pred_cluster(X, clusters):
    pred = []
    for i in range(X.shape[0]):
        dist = []
        for j in range(k):
            dist.append(distance(X[i],clusters[j]['center']))
        pred.append(np.argmin(dist))
    return pred
```

Implementation of K-Means Clustering in Python

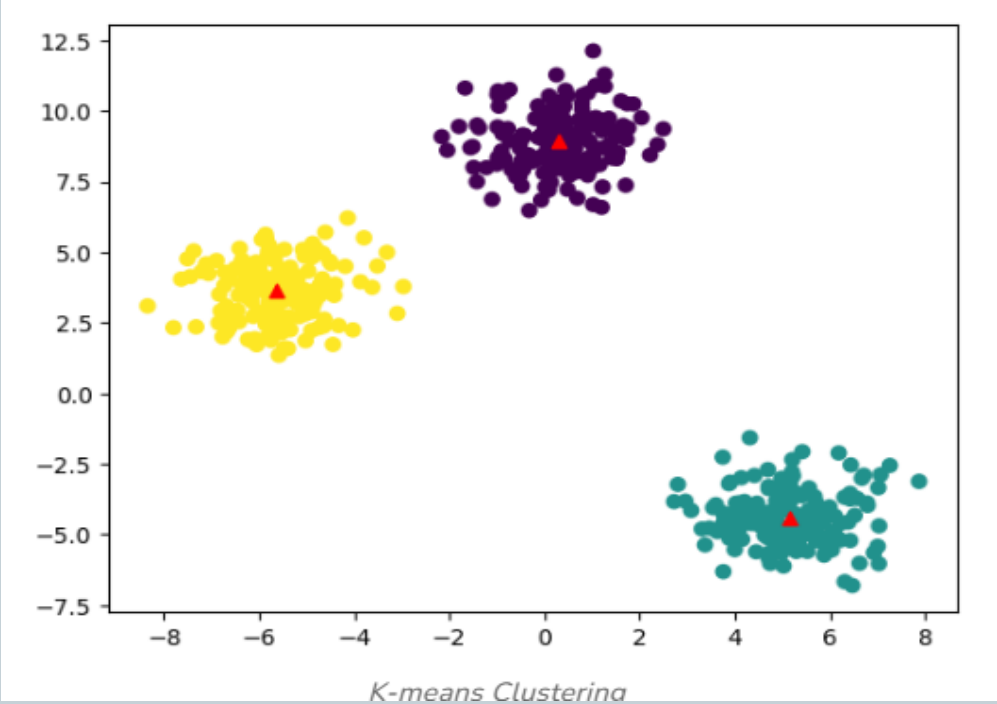
- Step 8: Assign, Update, and predict the cluster center

```
clusters = assign_clusters(X,clusters)
clusters = update_clusters(X,clusters)
pred = pred_cluster(X,clusters)
```

- Step 9: Plot the data points with their predicted cluster center

```
plt.scatter(X[:,0],X[:,1],c = pred)
for i in clusters:
    center = clusters[i]['center']
    plt.scatter(center[0],center[1],marker = '^',c = 'red')
plt.show()
```

Output



- The plot shows data points colored by their predicted clusters. The red markers represent the updated cluster centers after the E-M steps in the K-means clustering algorithm.

Unsupervised Learning Algorithms

- ***The second types of algorithms that used for unsupervised learning:***

2- Association Rule Learning

- Is also known as association rule mining is a common technique used to discover associations in unsupervised machine learning.
- This technique is a rule-based ML technique that finds out some very useful relations between parameters of a large data set. This technique is basically used for market basket analysis that helps to better understand the relationship between different products.
- E.g. ***shopping stores*** use algorithms based on this technique to find out the relationship between the sale of one product w.r.t to another's sales based on customer behavior. ***Like if a customer buys milk, then he may also buy bread, eggs, or butter.*** Once trained well, such models can be used to increase their sales by planning different offers.

Some Common Association Rule Learning Algorithms

- ***Apriori Algorithm:*** Finds patterns by exploring frequent item combinations step-by-step.
- ***FP-Growth Algorithm:*** An Efficient Alternative to Apriori. It quickly identifies frequent patterns without generating candidate sets.
- ***Eclat Algorithm:*** Uses intersections of itemsets to efficiently find frequent patterns.
- ***Efficient Tree-based Algorithms:*** Scales to handle large datasets by organizing data in tree structures.

Unsupervised Learning Algorithms

- ***The third types of algorithms that are used for Unsupervised learning:***

  3- Dimensionality Reduction

  - Dimensionality reduction is the process of reducing the number of features in a dataset while preserving as much information as possible. This technique is useful for improving the performance of machine learning algorithms and for data visualization.

  - Imagine a dataset of ***100 features*** about students ***(height, weight, grades, etc***.). To focus on key traits, you reduce it to just ***2 features***: height and grades, making it easier to visualize or analyze the data.

Some popular Dimensionality Reduction algorithms

- ***Principal Component Analysis (PCA):*** Reduces dimensions by transforming data into uncorrelated principal components.
- ***Linear Discriminant Analysis (LDA):*** Reduces dimensions while maximizing class separability for classification tasks.
- ***Non-negative Matrix Factorization (NMF):*** Breaks data into non-negative parts to simplify representation.
- ***Locally Linear Embedding (LLE):*** Reduces dimensions while preserving the relationships between nearby points.
- ***Isomap:*** Captures global data structure by preserving distances along a manifold.