Unsupervised Learning

Machine Learning Basics - Continue

Lara Wehbe - TheAIEngineers - August 2024

Outline

In this course, we will discuss

1. Introduction to Unsupervised Learning
2. Real Life Applications
3. Algorithms:
    1. Clustering:
        1. Kmeans Clustering
        2. Hierarchal Clustering
    2. Recommender Systems:
        1. Collaborative-based Recommendations
        2. Content-based Recommendations
4. Conclusion

Intro to Unsupervised Learning

Definition

Unsupervised machine learning models are given unlabeled data and allowed to discover patterns and insights without any explicit guidance or instruction.

How Unsupervised Learning works?

Imagine that you have a large dataset about weather. An unsupervised learning algorithm will go through the data and identify patterns in the data points. For instance, it might group data by temperature or similar weather patterns.

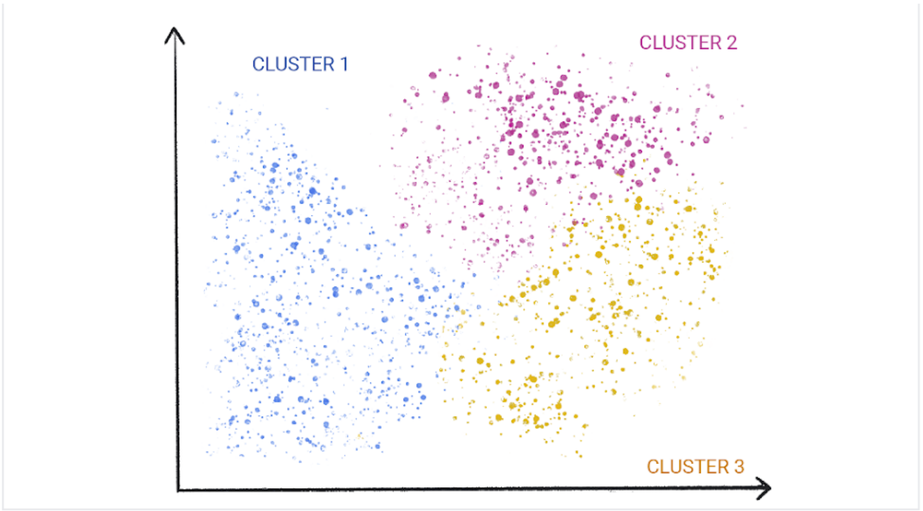


Figure 1. An ML model clustering similar data points.

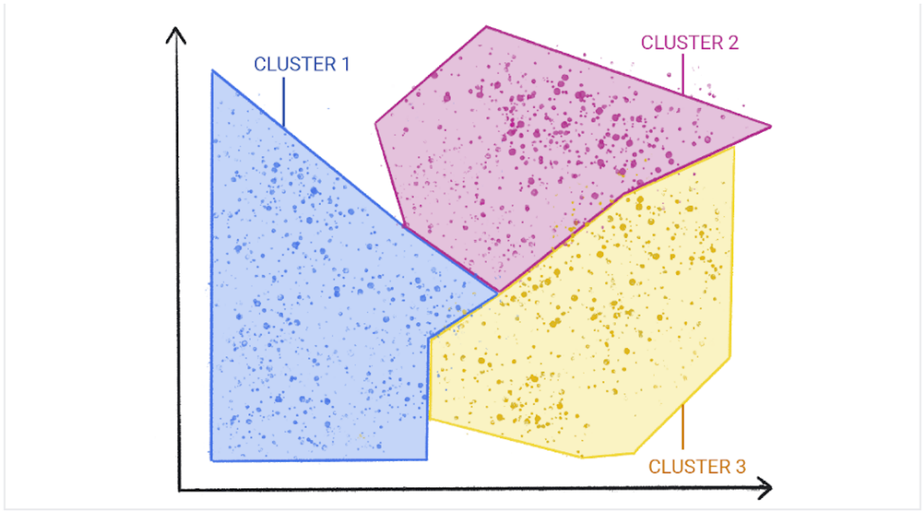


Figure 2. Groups of clusters with natural demarcations.

Applications

Movie Recommendations

Applications

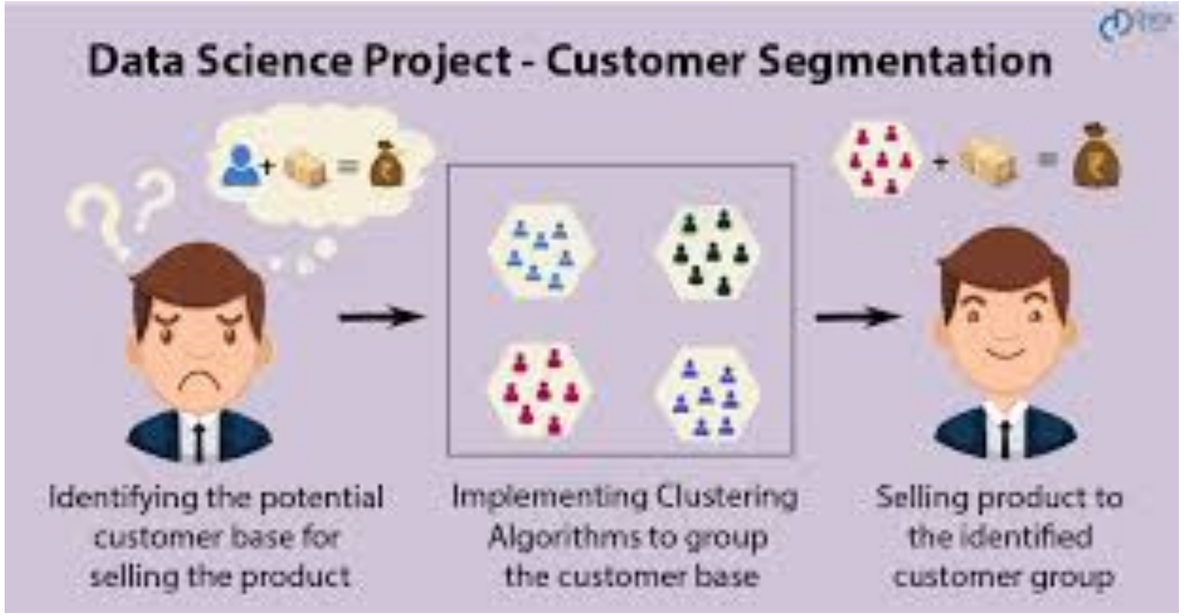
Google News

Google News
Unsupervised
Machine
Learning

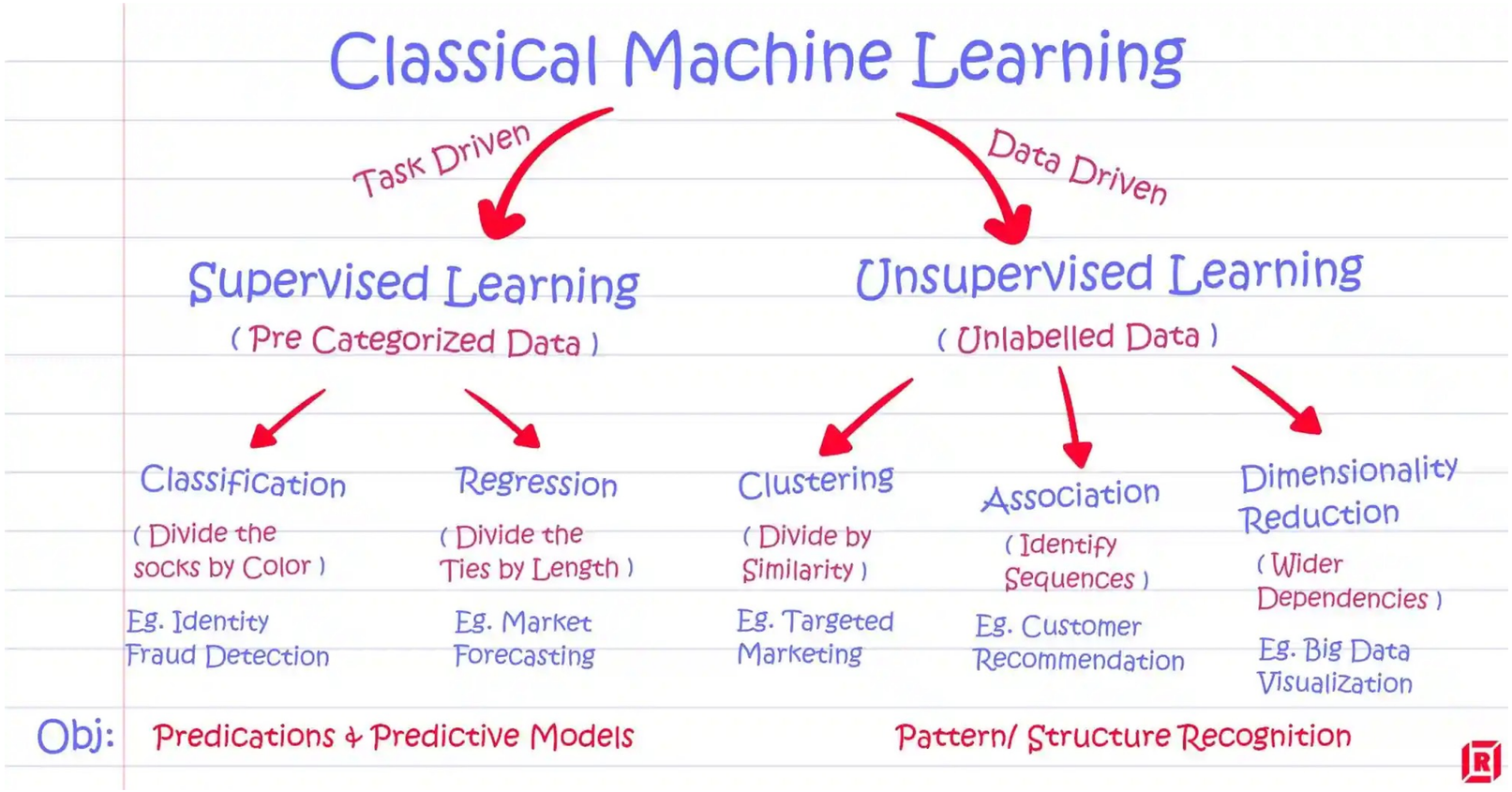
How Does Google Know
That These are Related?

Applications

Customer Segmentation



How it's different from Supervised Learning



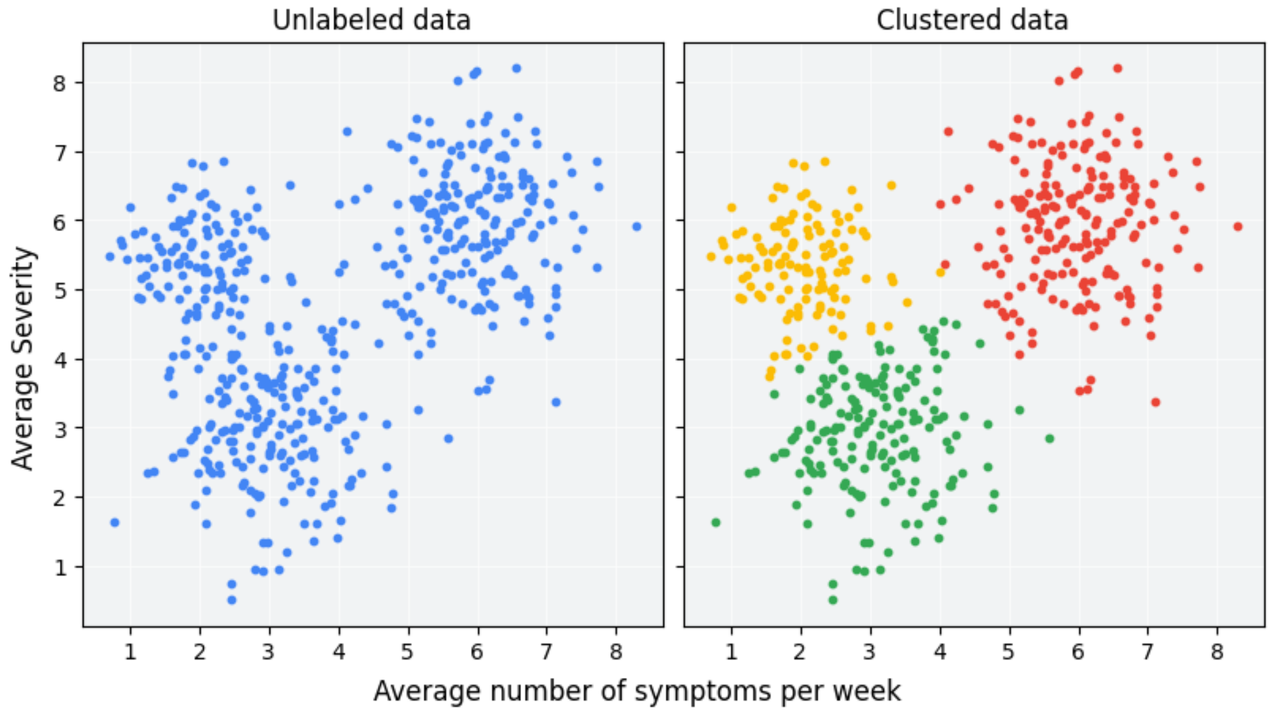
Unsupervised Learning - Algorithms

Clustering

Definition

What is Clustering?

Clustering is an unsupervised machine learning technique designed to group **unlabeled examples** based on their similarity to each other.



Applications

Social Media Analysis

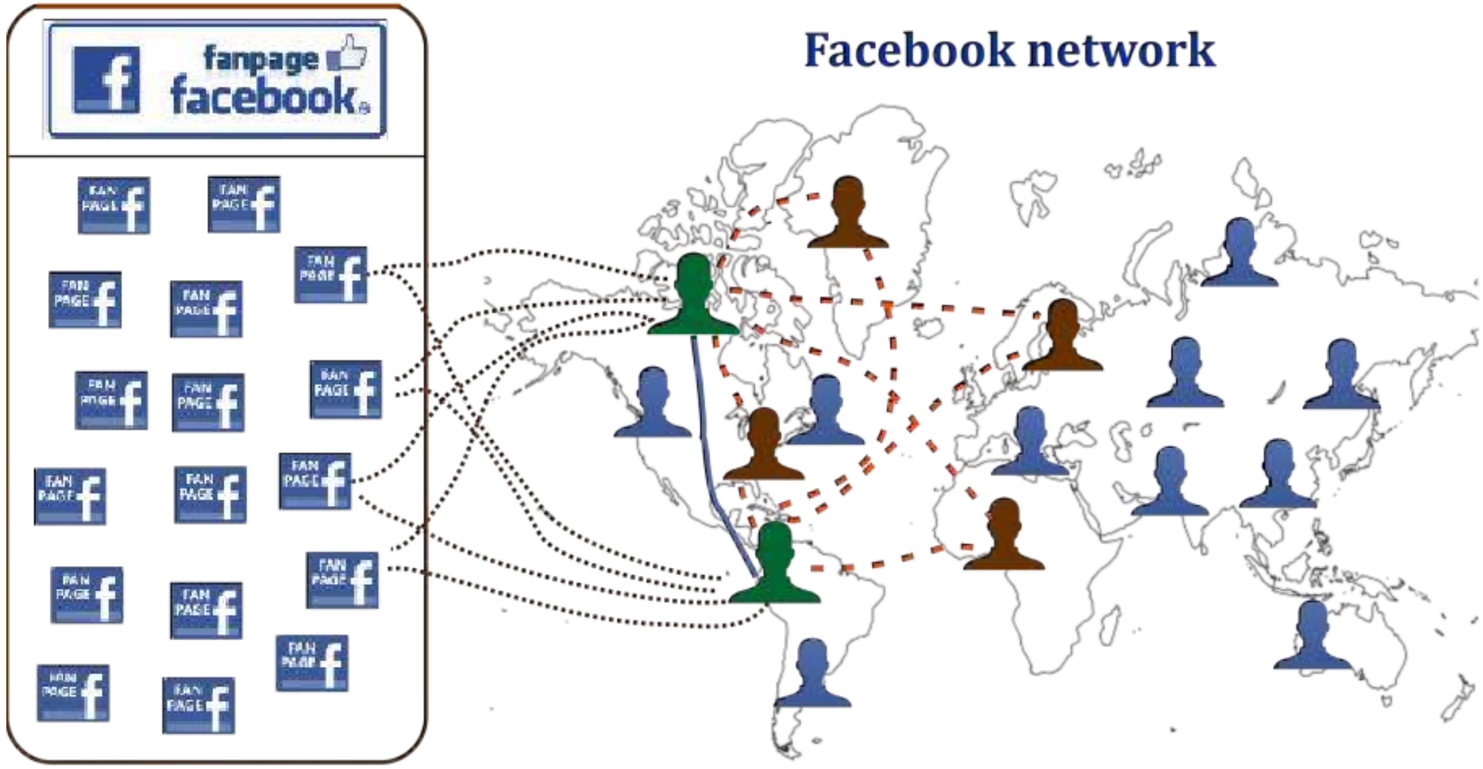
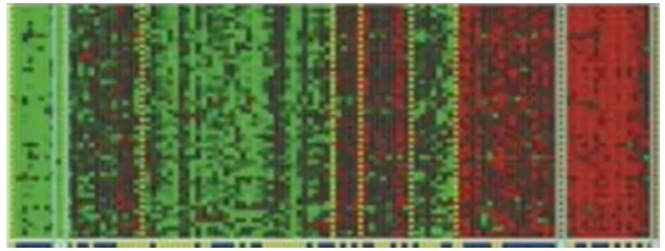
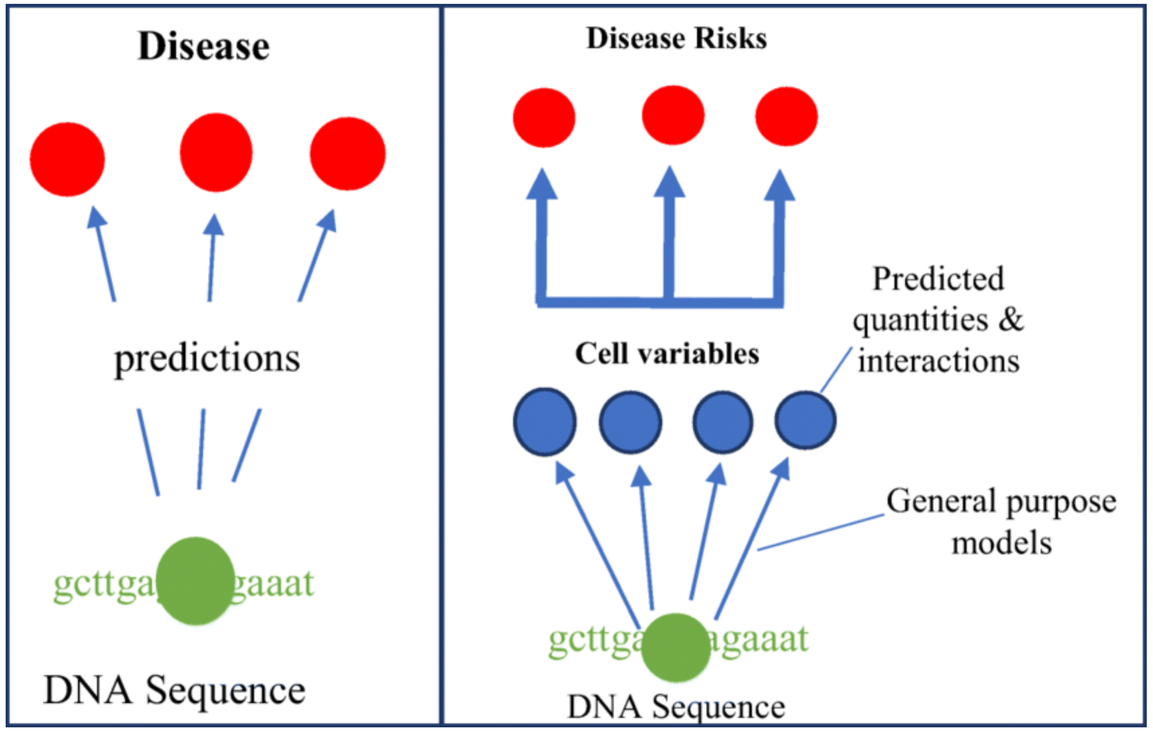


Figure 1 Friends (green) with four fanpages and four friends in common

Applications

DNA Sampling





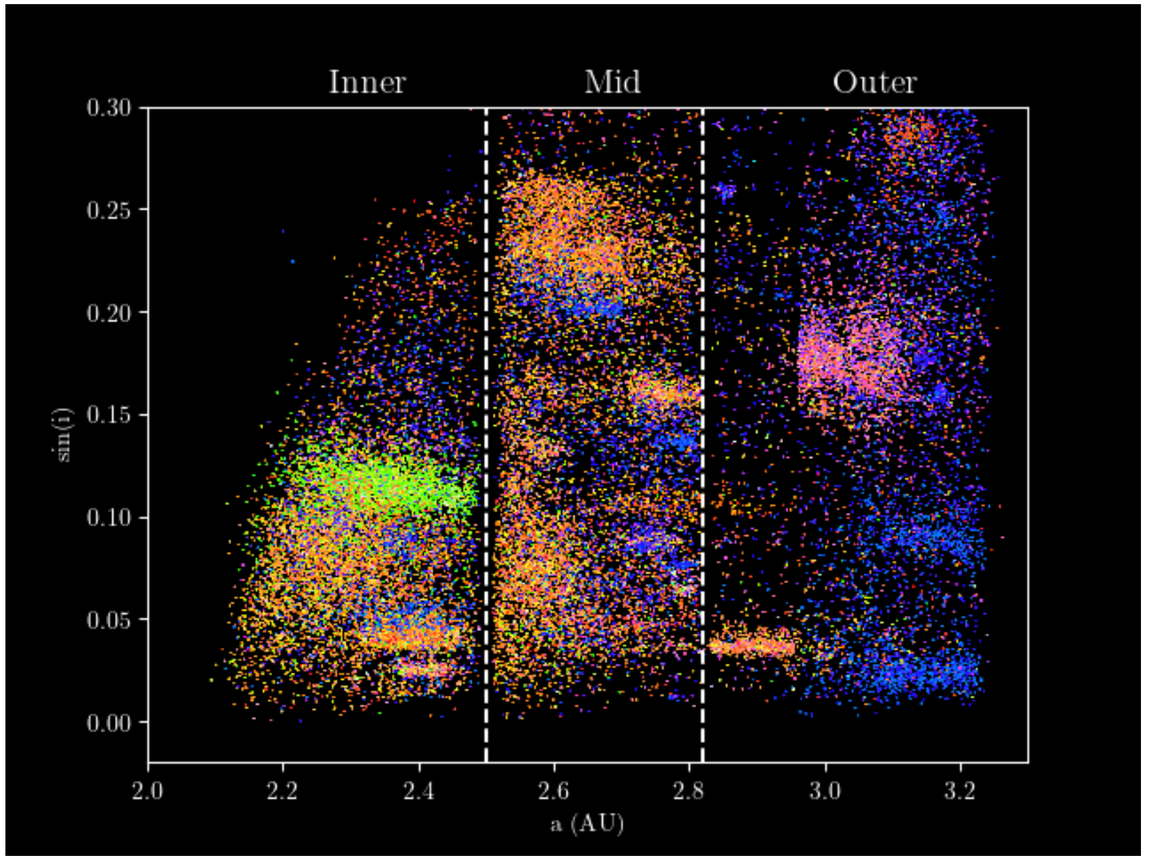
Applications

The Big 5 model of personality traits

Applications

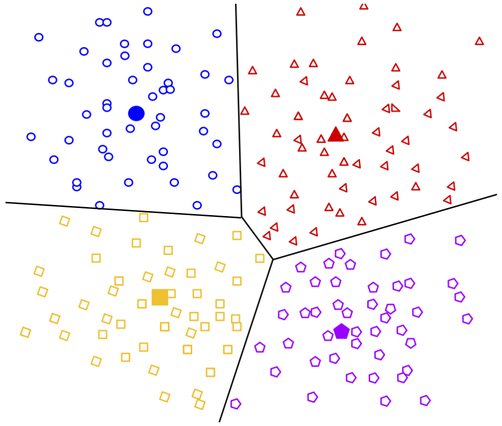
Astronomical Analysis

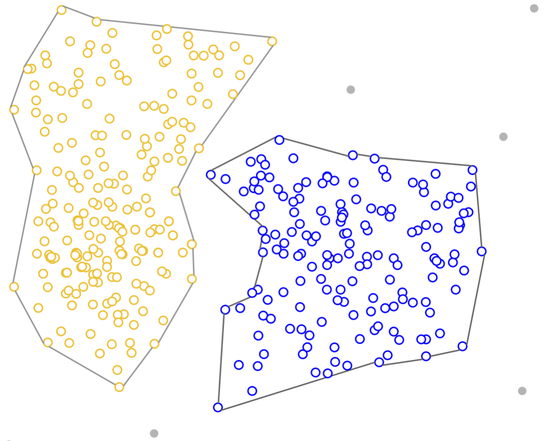
Classification of Celestial Objects

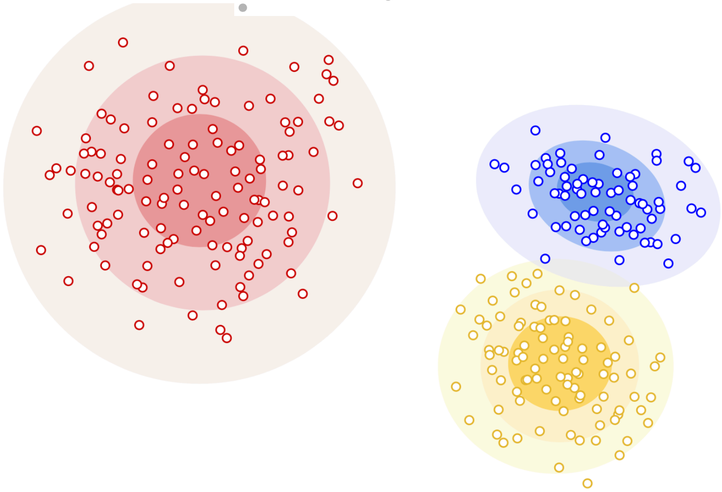


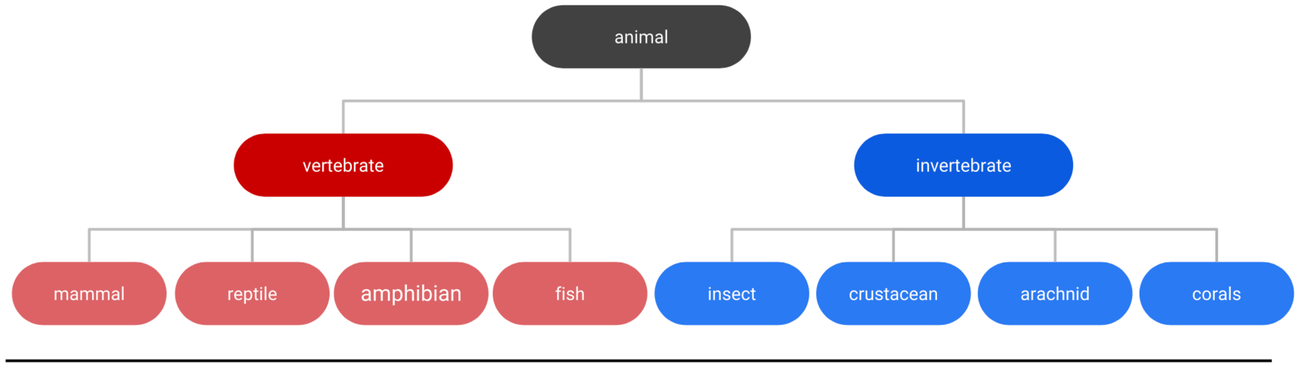
Types of Clustering

1. Centroid Clustering
2. Density-based Clustering
3. Distribution-based
4. Hierarchal Clustering









K-means Clustering

K-means Clustering

K-Means performs the division of objects into clusters that share similarities and are dissimilar to the objects belonging to another cluster.

The main objective of the K-Means algorithm is to minimize the sum of distances between the points and their respective cluster centroid.

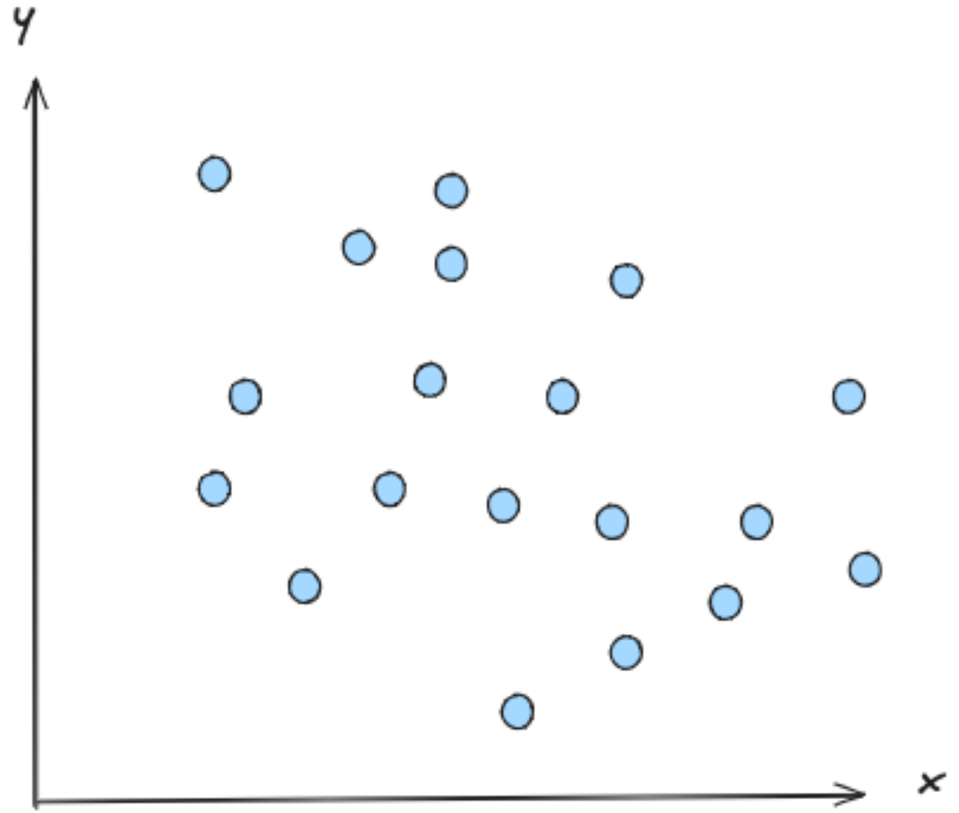
K-means Clustering

How does it work?

1. Initialization
2. Assignment
3. Update Centroids
4. Repeat
5. Final Result

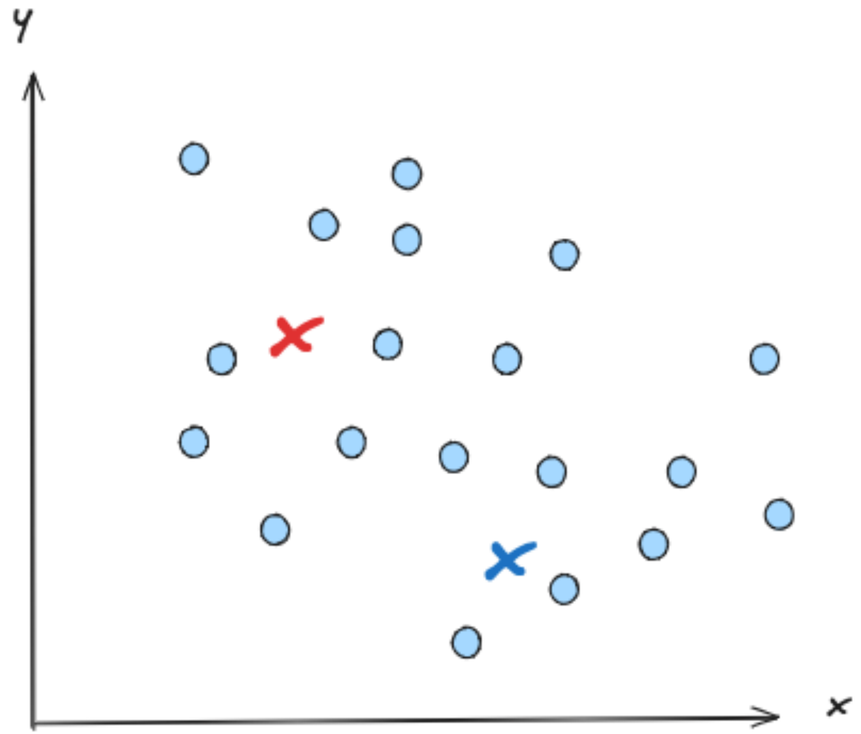
K-means Clustering

Let’s run K-means on this dataset



K-means Clustering

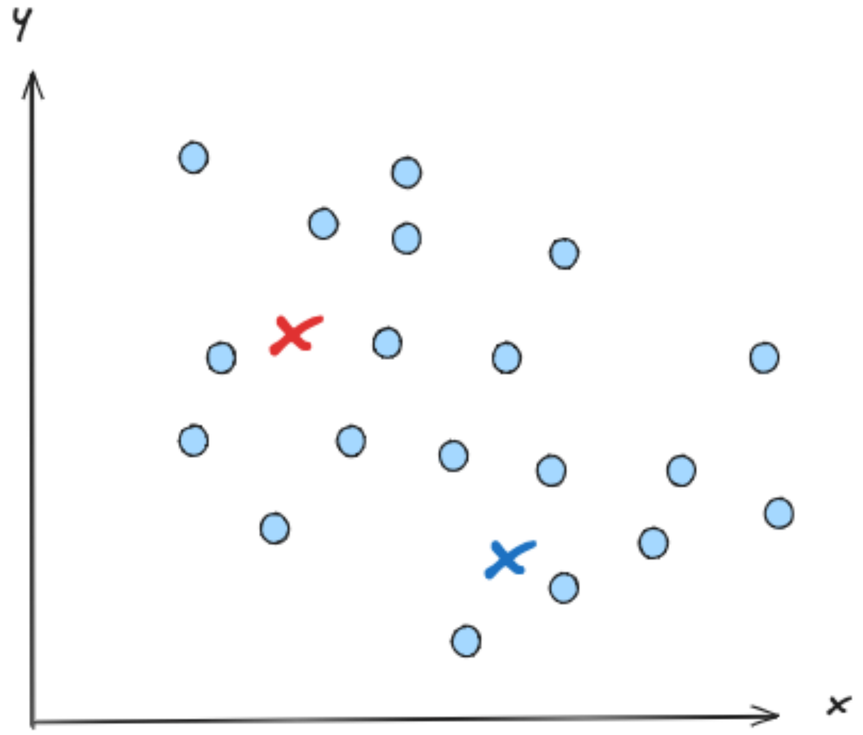
Randomly choose two centroids, k = 2



K-means Clustering

Assign each point to its closest centroid

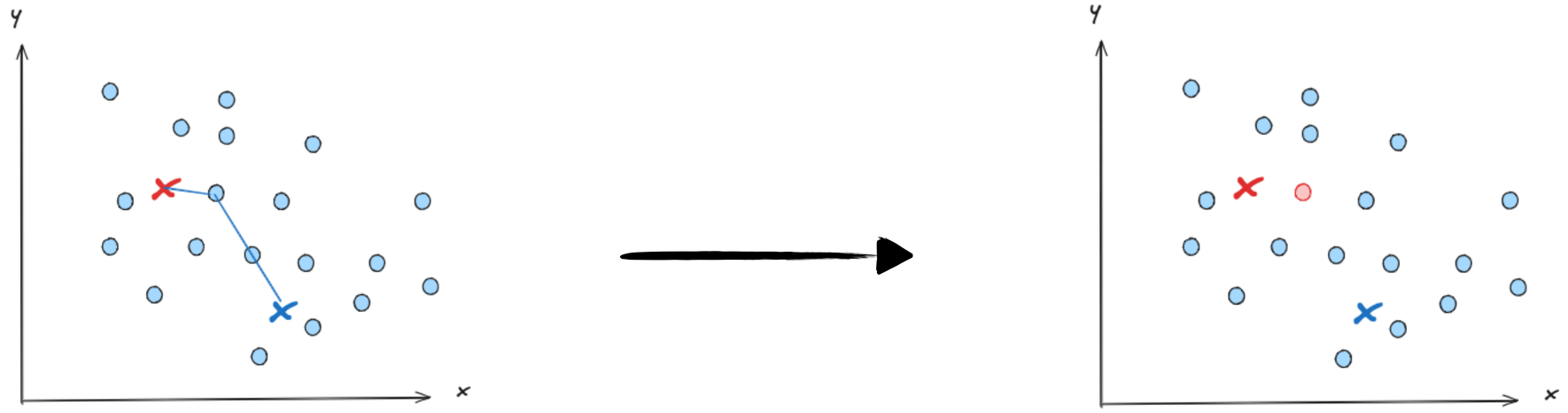
Let xi be the data point.
For each x in {x1,x2,…xn}:
Assign the point to its closest centroid



K-means Clustering

Assign each point to its closest centroid

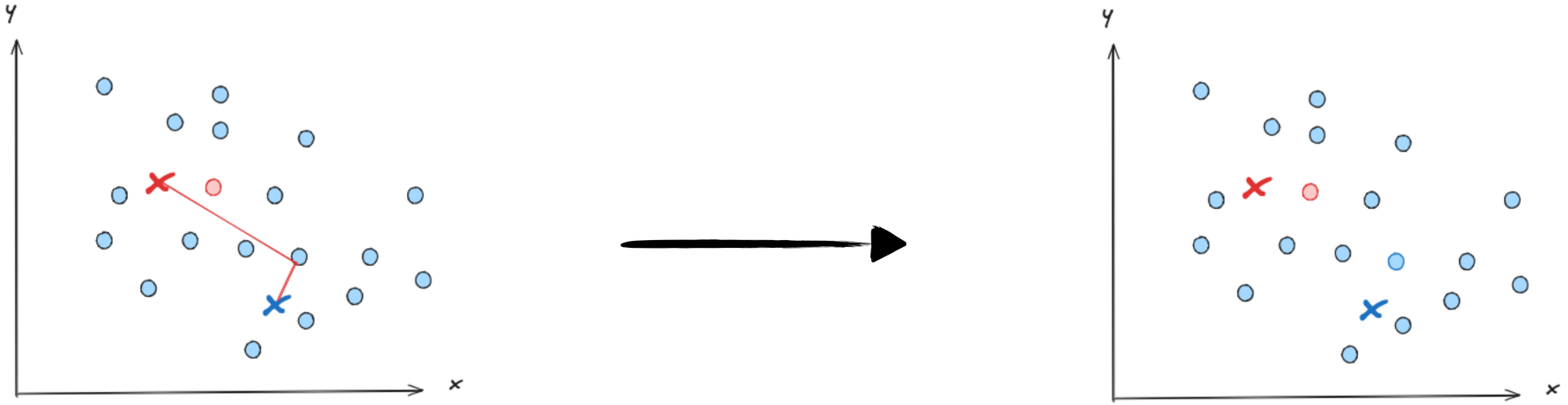
Let xi be the data point.
For each x in {x1,x2,…xn}:
Assign the point to its closest centroid



K-means Clustering

Assign each point to its closest centroid

Let xi be the data point.
For each x in {x1,x2,…xn}:
Assign the point to its closest centroid

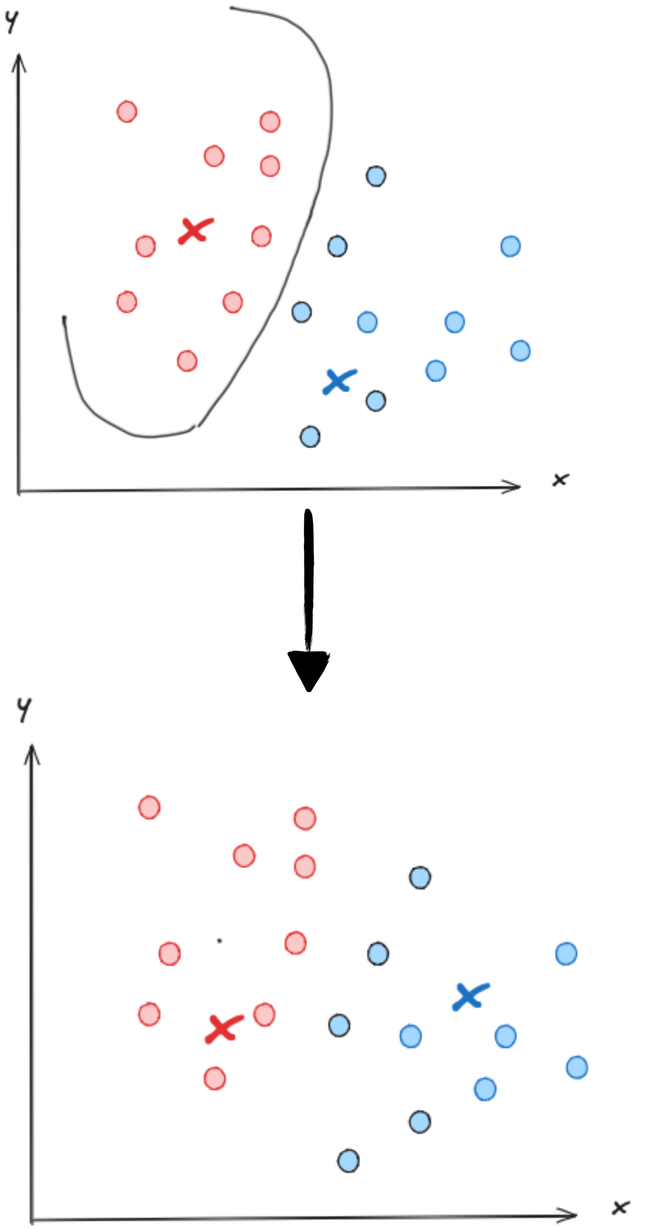


K-means Clustering

Recompute the Centroids

Let xi be the data point.
For each x in {x1,x2,…xn}:
Assign the point to its closest centroid

for μ_i in {μ1,μ2} : #μ here is the centroid.
Calculate the average of the points in the cluster i
And reassign the cluster centroid to the average value

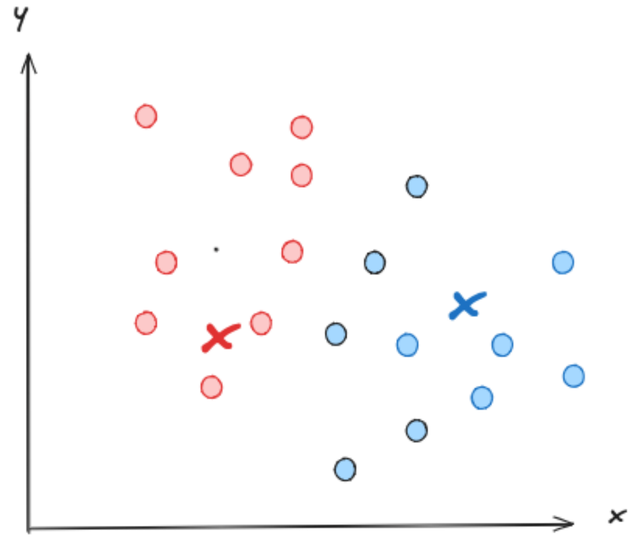


K-means Clustering

Repeat Step 1: Recalculate the distance between data points and new centroids

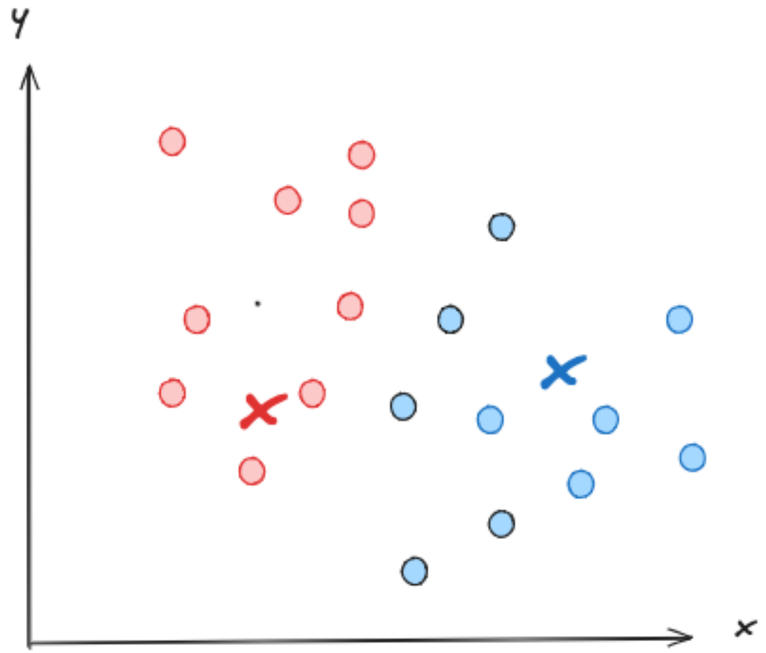
Let xi be the data point.
For each x in {x1,x2,…xn}:
Assign the point to its closest centroid

for μ_i in {μ1,μ2} : #μ here is the centroid.
Calculate the average of the points in the cluster i
And reassign the cluster centroid to the average value



K-means Clustering

Stop criteria: Repeat steps 1 and 2 until the centroid location doesn’t change anymore



K-means Clustering

Math behind K-means

Lets consider we have cluster points P1(1,3) , P2(2,2) , P3(5,8) , P4(8,5) , P5(3,9) , P6(10,7) , P7(3,3) , P8(9,4) , P9(3,7).

Step 1: Choose k randomly.

Let’s take k=3

assume that our Initial cluster centers are P7(3,3), P9(3,7), P8(9,4) as C1, C2, C3.

K-means Clustering

Math behind K-means

Step 2:

Calculate the distance between data points and centroids, where the data point having a minimum distance will be moved to the cluster c_i.

$$\text{Distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

K-means Clustering

Math behind K-means

P7(3,3), P9(3,7), P8(9,4) as C1, C2, C3.

P1(1,3)

Iteration 1:

Calculate the distance between data points and K (C1,C2,C3)

C1P1 =>(3,3)(1,3) => $\text{sqrt}[(1-3)^2+(3-3)^2]$ => sqrt[4] =>2

C2P1 =>(3,7)(1,3)=> $\text{sqrt}[(1-3)^2+(3-7)^2]$ => sqrt[20] =>4.5

C3P1 =>(9,4)(1,3) => $\text{sqrt}[(1-9)^2+(3-4)^2]$=> sqrt[65] =>8.1

K-means Clustering

Math behind K-means

P7(3,3), P9(3,7), P8(9,4) as C1, C2, C3.

P1(1,3)

Iteration 1:

Calculate the distance between data points and K (C1,C2,C3)

C1P1 =>(3,3)(1,3) => sqrt[$(1-3)^2+(3-3)^2$] => sqrt[4] =>2

C2P1 =>(3,7)(1,3)=> sqrt[$(1-3)^2+(3-7)^2$] => sqrt[20] =>4.5

C3P1 =>(9,4)(1,3) => sqrt[$(1-9)^2+(3-4)^2$]=> sqrt[65] =>8.1

Since C1P1 has the smallest value, hence P1 belongs to cluster C1

K-means Clustering

Math behind K-means

P7(3,3), P9(3,7), P8(9,4) as C1, C2, C3.

P1(1,3)

Doing the same for all points we get:

Data Points	Centroid (3,3)	Centroid (3,7)	Centroid (9,4)	Cluster
P1(1,3)	2	4.5	8.1	C1
P2(2,2)	1.4	5.1	7.3	C1
P3(5,8)	5.3	2.2	5.7	C2
P4(8,5)	5.4	5.4	5.1	C3
P5(3,9)	6	2	7.9	C2
P6(10,7)	8.1	7	3.2	C3
P7(3,3)	0	4	6.1	C1
P8(9,4)	6.1	6.7	0	C3
P9(3,7)	4	0	6.7	C2

K-means Clustering

P7(3,3), P9(3,7), P8(9,4) as C1, C2, C3.

P1(1,3)

Math behind K-means

Step 2: Recompute the new clusters and the new cluster centers by taking the mean of all points contained in a cluster.

Cluster 1 => P1(1,3) , P2(2,2) , P7(3,3)

New center of Cluster 1 => (1+2+3)/3 , (3+2+3)/3 => 2,2.7

Cost function for K-means

Inertia

Inertia actually calculates the sum of distances of all the points within a cluster from the centroid of that cluster.

$$J = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \|x_i^{(k)} - \mu_k\|^2$$

where:

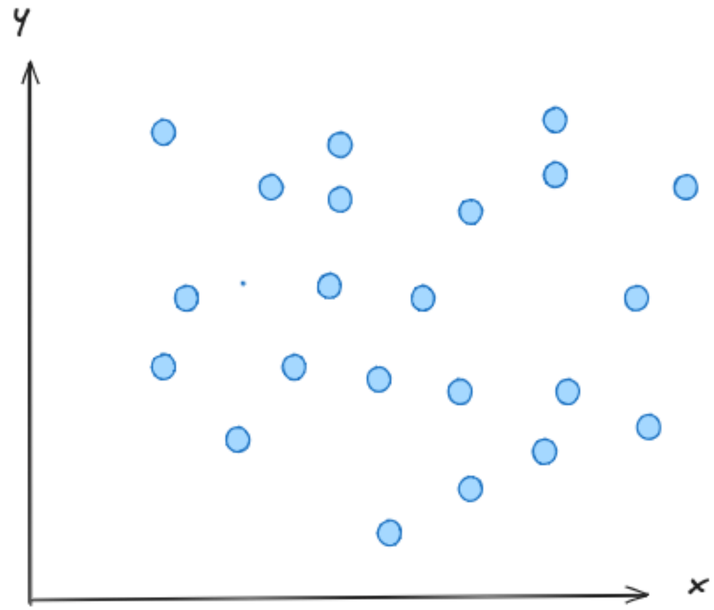
- K is the number of clusters.
- n_k is the number of points in cluster k.
- $x_i^{(k)}$ is the i-th data point in cluster k.
- μ_k is the centroid of cluster k.
- $\|x_i^{(k)} - \mu_k\|^2$ represents the squared Euclidean distance between a data point and the cluster centroid.

Choosing the number of clusters

K-means Clustering

We said that we randomly choose the value of K.

How many clusters do we have here?



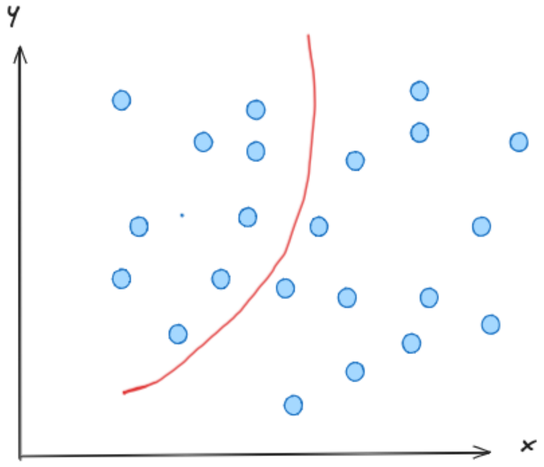
Choosing the number of clusters

K-means Clustering

We said that we randomly choose the value of K.

How many clusters do we have here?

Some people will say two clusters



Choosing the number of clusters

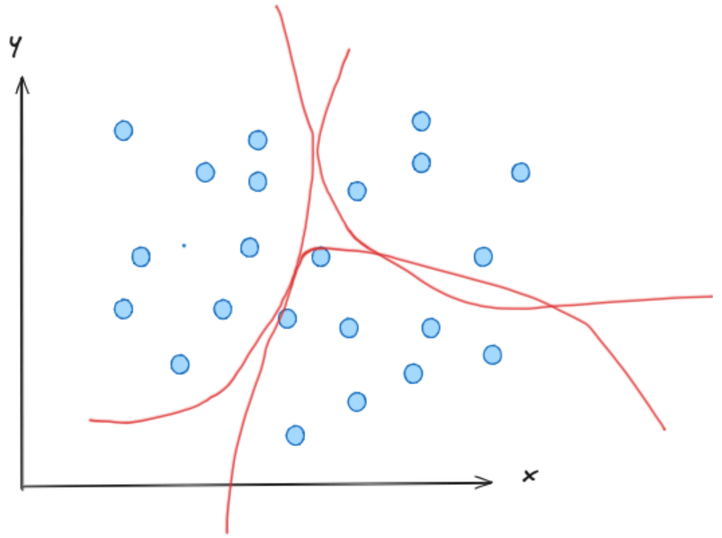
K-means Clustering

We said that we randomly choose the value of K.

How many clusters do we have here?

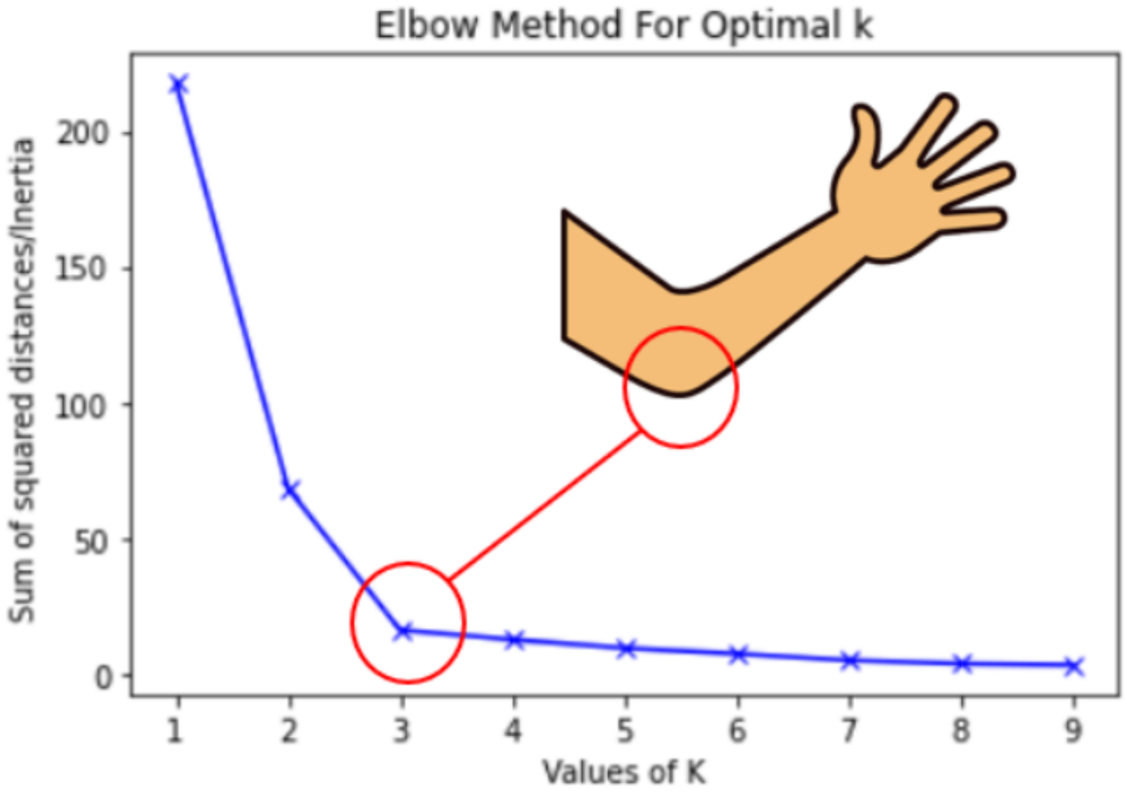
Other people will say Three clusters

And guess what? Both of them might be right



Choosing the number of clusters

K-means Clustering



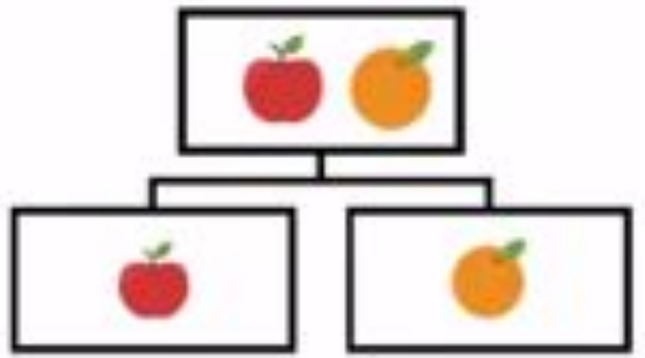
Line plot between K and inertia

Hierarchal Clustering

Hierarchal Clustering

Hierarchical clustering uses a tree-like structure, like so.

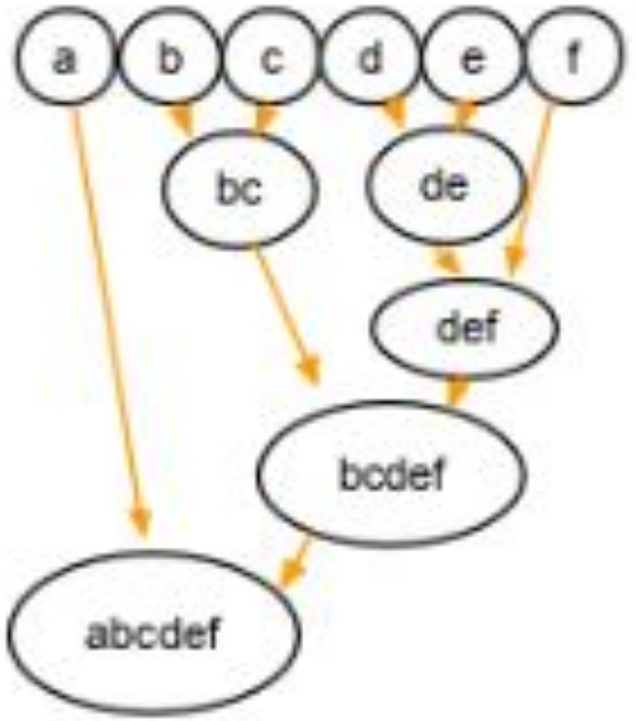
The tree represents the relationships between the objects.



Hierarchal Clustering

Agglomerative Clustering:

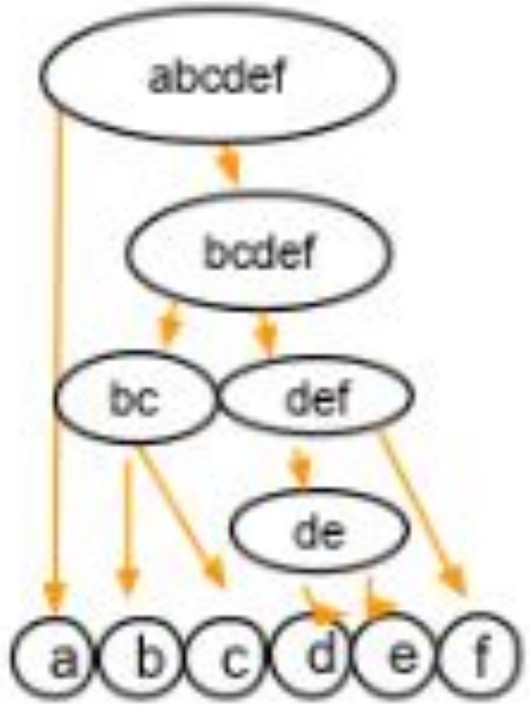
In agglomerative clustering, there is a bottom-up approach. We begin with each element as a separate cluster and merge them into successively more massive clusters, as shown below



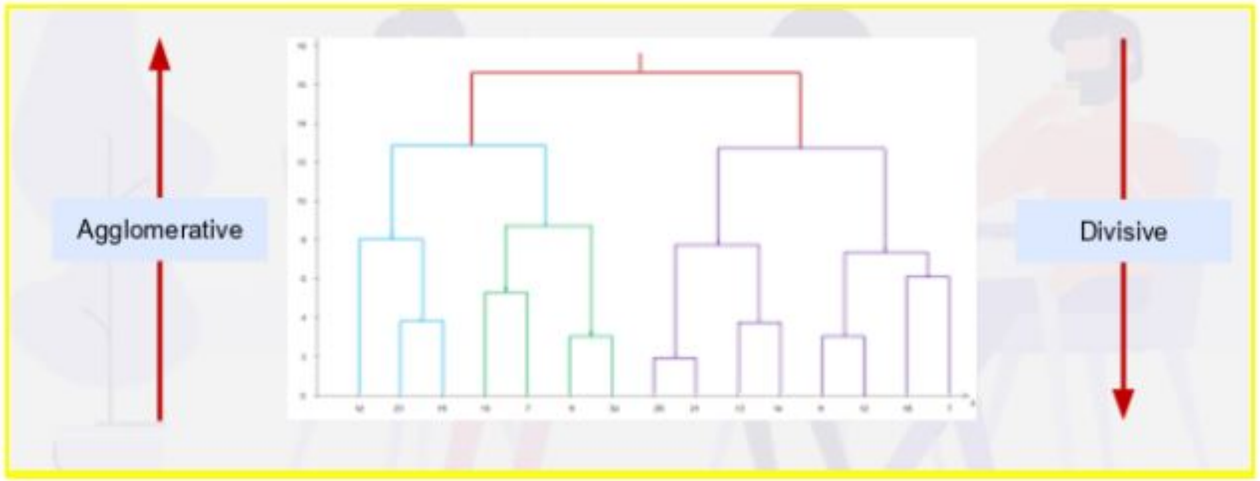
Hierarchal Clustering

Divisive Clustering:

Divisive clustering is a top-down approach. We begin with the whole set and proceed to divide it into successively smaller clusters, as you can see below



Hierarchal Clustering



Hierarchal Clustering

Let’s take an example

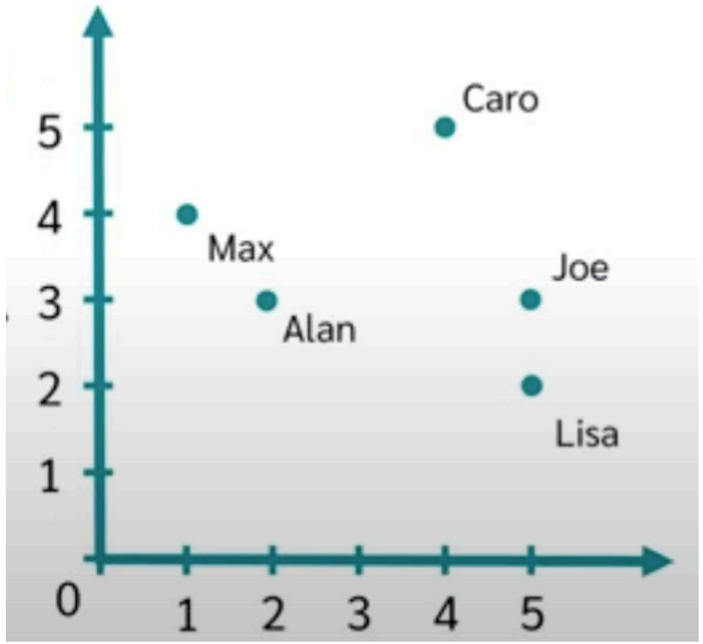
We asked people how many hours a week they spend on social media and gym

	Social Media	Gym
Alan	2	3
Lisa	5	2
Joe	5	3
Max	1	4
Caro	4	5

Hierarchal Clustering

Let’s take an example

First, plot the data points on a scatter plot.

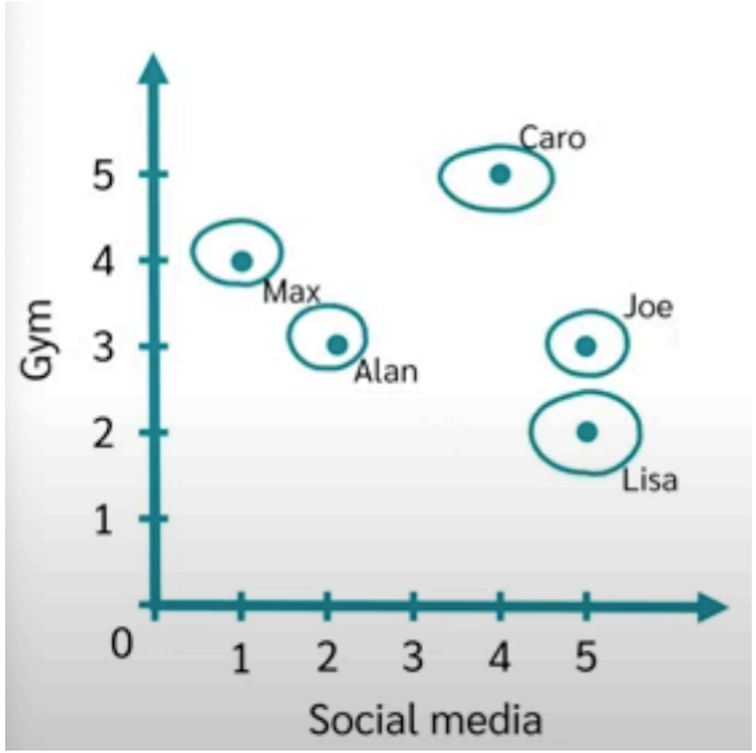


Hierarchal Clustering

Let’s take an example

First, plot the data points on a scatter plot.

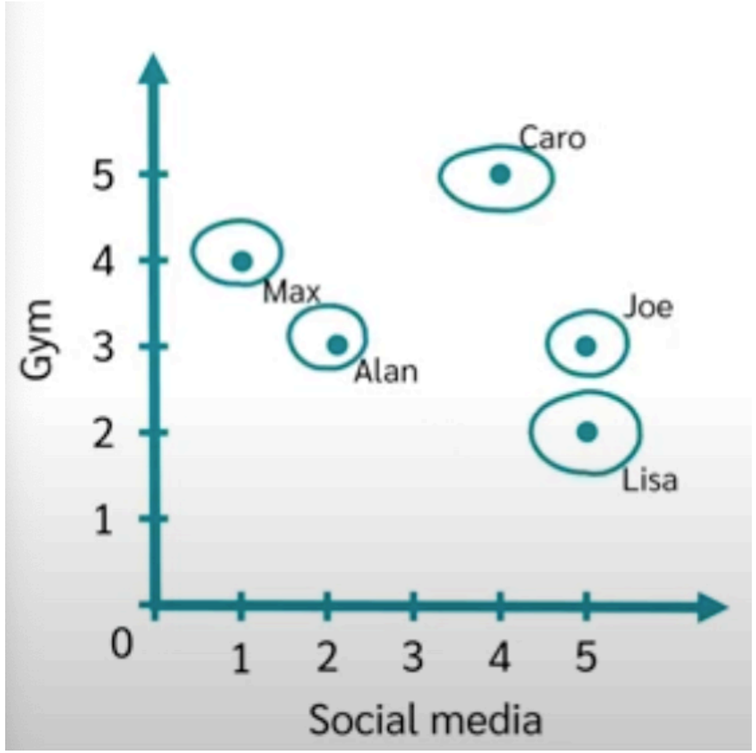
And, assign a closer to each individual point.



Hierarchal Clustering

Let’s take an example

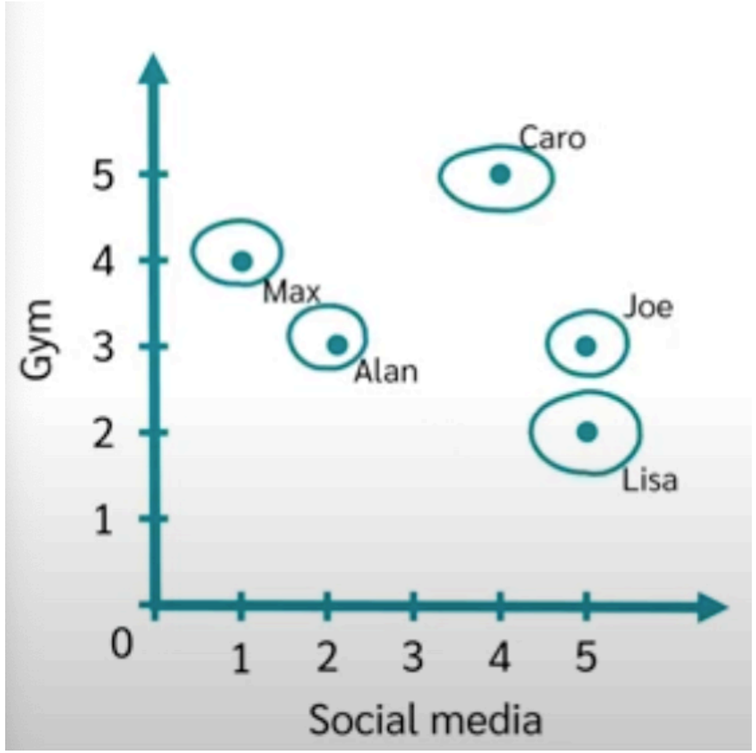
The goal is to **gradually merge more and more clusters.**



Hierarchal Clustering

Let’s take an example

In order to establish the hierarchal cluster, we need to merge the closest clusters together.

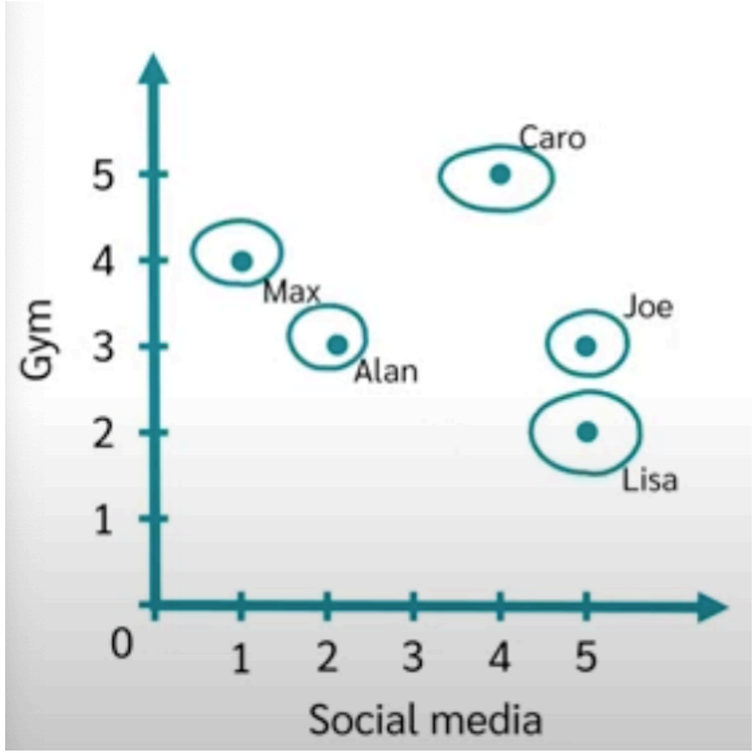


Hierarchal Clustering

Let’s take an example

How do we calculate the distance between two points?

1. Manhattan distance
2. Euclidean distance
3. Maximum distance

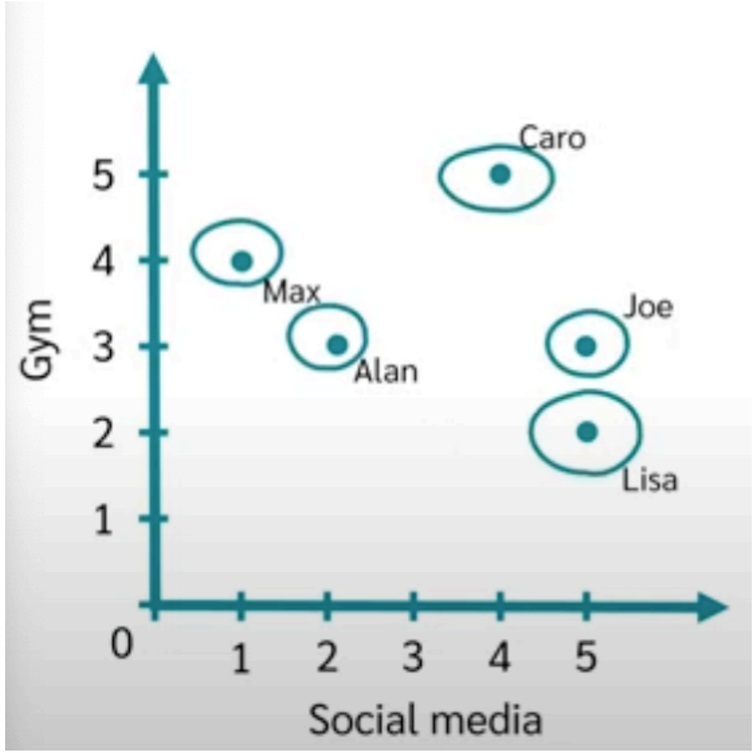


Hierarchal Clustering

Let’s take an example

How are the points linked together?

1. Single-linkage
2. Complete-linkage
3. Average-linkage

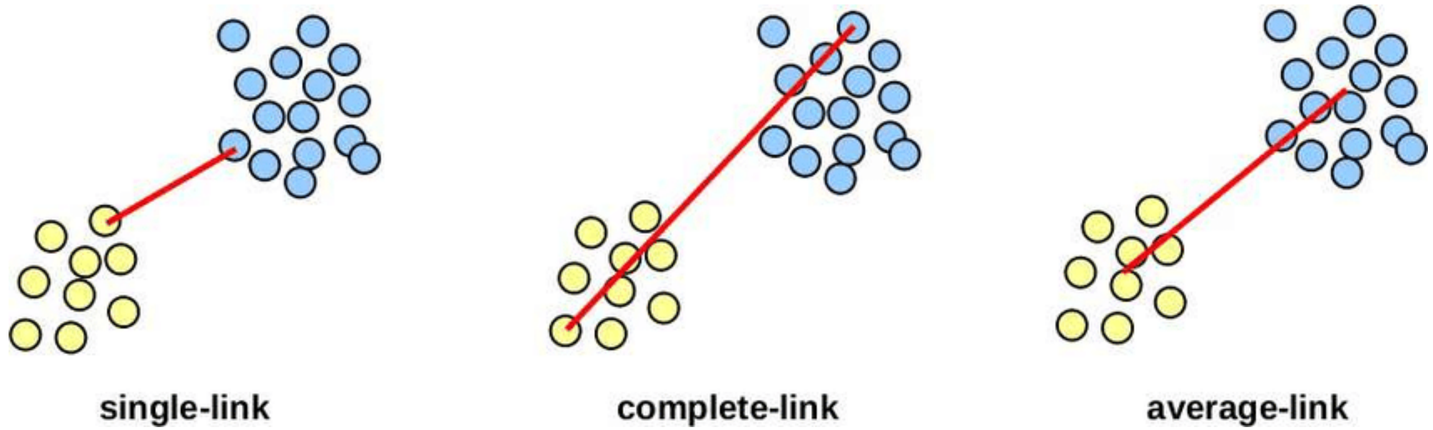


Hierarchal Clustering

Let’s take an example

How are the points linked together?

1. Single-linkage
2. Complete-linkage
3. Average-linkage



Hierarchal Clustering

Let’s take an example

For our example, we will use:

1. Euclidean distance
2. Single-linkage

Hierarchal Clustering

Let’s take an example

Calculate the distance Matrix

	Alan	Lisa	Joe	Max	Caro
Alan	0				
Lisa	3,16	0			
Joe	3,00	1,00	0		
Max	1,41	4,47	4,12	0	
Caro	2,83	3,16	2,24	3,16	0

The distance between **Alan and Lisa** is given by:

$$d = \sqrt{(5-2)^2+(2-3)^2} = 3{,}16$$

	Social media	Gym
Alan	2	3
Lisa	5	2
Joe	5	3
Max	1	4
Caro	4	5

Hierarchal Clustering

Let’s take an example

Next, we need to merge the first clusters.

Since we are using “Single-linkage”, we need to find the smallest distance between them.

Looking at the table, we get the smallest one is “Lisa and Joe”.

	Alan	Lisa	Joe	Max	Caro
Alan	0				
Lisa	3,16	0			
Joe	3,00	1,00	0		
Max	1,41	4,47	4,12	0	
Caro	2,83	3,16	2,24	3,16	0

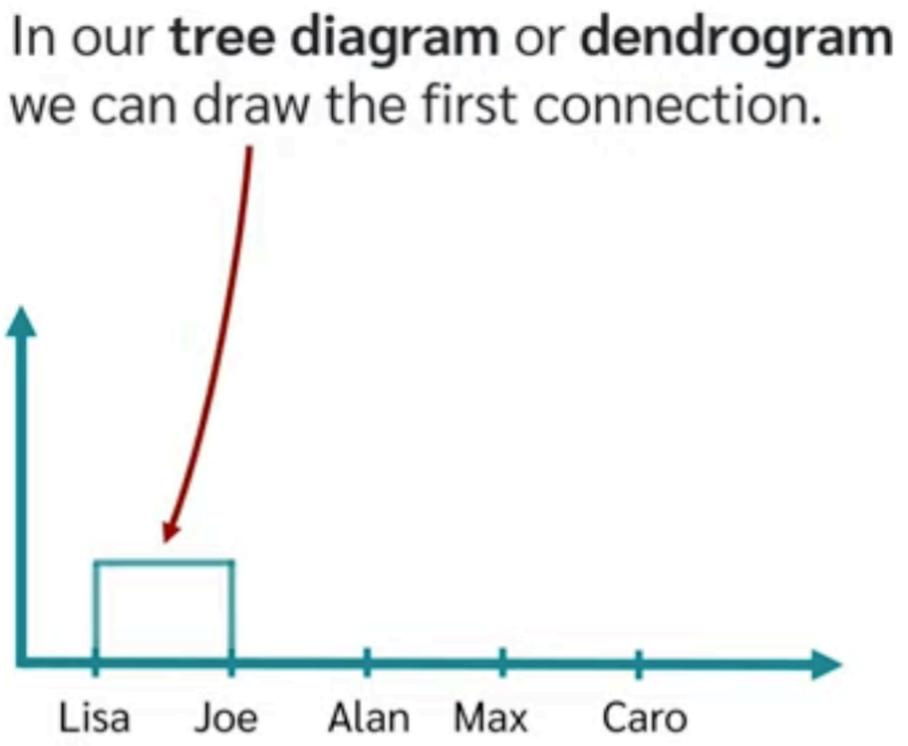
Hierarchal Clustering

Let’s take an example

Next, we need to merge the first clusters.

Since we are using “Single-linkage”, we need to find the smallest distance between them.

Looking at the table, we get the smallest one is “Lisa and Joe”.



Hierarchal Clustering

Let’s take an example

Now, repeat for other clusters, but calculating the distance between data points like Alan, and the new cluster that is (Lisa,Joe)

We get the following Matrix table:

	Alan	Lisa, Joe	Max	Caro
Alan	0			
Lisa, Joe	3,00	0		
Max	1,41	4,12	0	
Caro	2,83	2,24	3,16	0

Based on the table data, the next closest point is Alan.

Hierarchal Clustering

Let’s take an example

Now, repeat for other clusters, but calculating the distance between data points like Alan, and the new cluster that is (Lisa,Joe)

We get the following Matrix table:

	Alan	Lisa, Joe	Max	Caro
Alan	0			
Lisa, Joe	3,00	0		
Max	1,41	4,12	0	
Caro	2,83	2,24	3,16	0

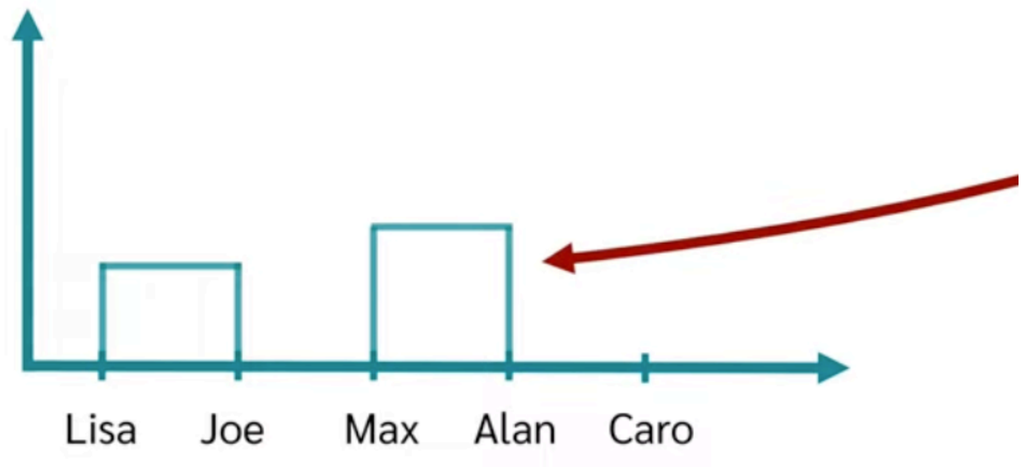
Based on the table data, the next closest points are Max and Alan

Hierarchal Clustering

Let’s take an example

Now, repeat for other clusters, but calculating the distance between data points like Alan, and the new cluster that is (Lisa,Joe)

We get the following Matrix table:



Based on the table data, the next closest points are Max and Alan

Hierarchal Clustering

Let’s take an example

This is the final Hierarchal Cluster!

Dimensionality Reduction

Curse of High Dimensionality

Can you train a GPT-like model locally?

Can you download and run llama3.1 model (Meta's gpt-like model) locally?

The answer to all this is **NO**.

Why?

Because they have **HUGE NUMBER OF DIMENSIONS.**

Curse of High Dimensionality

Curse of High Dimensionality

In many cases, dimensions of a model could be decreased.

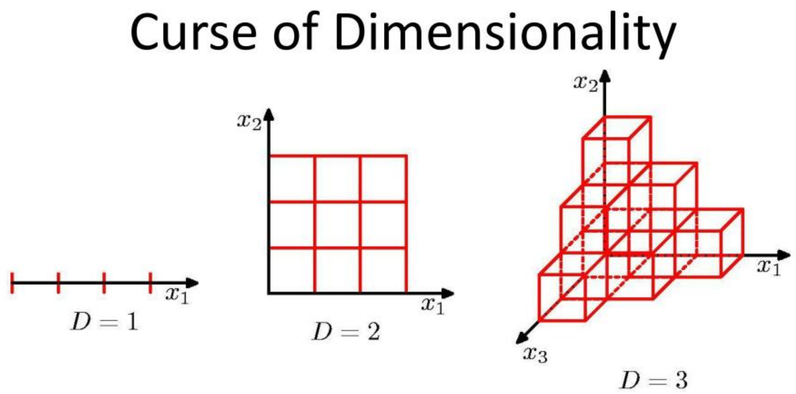
But first, what’s a dimension?

Dimension: Definition

In the context of data analysis and machine learning, dimensions refer to the features or attributes of data. For instance, if we consider a dataset of houses, the dimensions could include the house's price, size, number of bedrooms, location, and so on.

How does the curse of dimensionality occur?

As we add more dimensions to our dataset, the volume of the space increases exponentially. This means that the data becomes sparse



- No. of cells grow exponentially with D
- Need exponentially large no. of training data points
- Not a good approach for more than a few dimensions!

Reference: Christopher M Bishop: Pattern Recognition & Machine Leaning, 2006 Springer

Problems of high Dimensions:

1. Data Sparsity
2. Increased Computation
3. Overfitting
4. Performance Dégradation
5. Visualization challenges

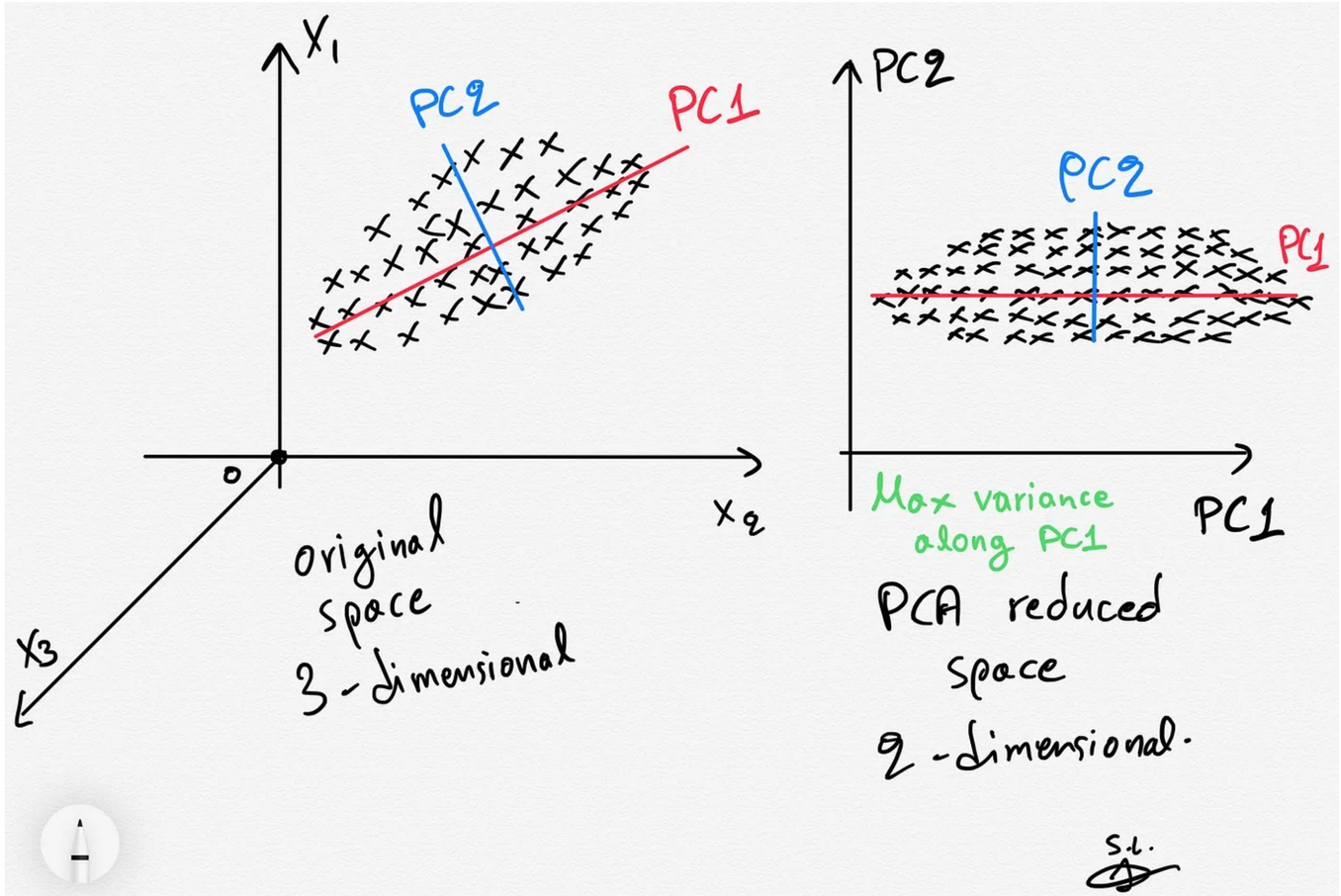
How to solve it?

Enter:
Dimensionality Reduction

Dimensionality Reduction

Principal Component Analysis (PCA)

PCA is a statistical method that transforms the original variables into a new set of variables, which are linear combinations of the original variables. These new variables are called principal components.



Recommender Systems

Recommender Systems

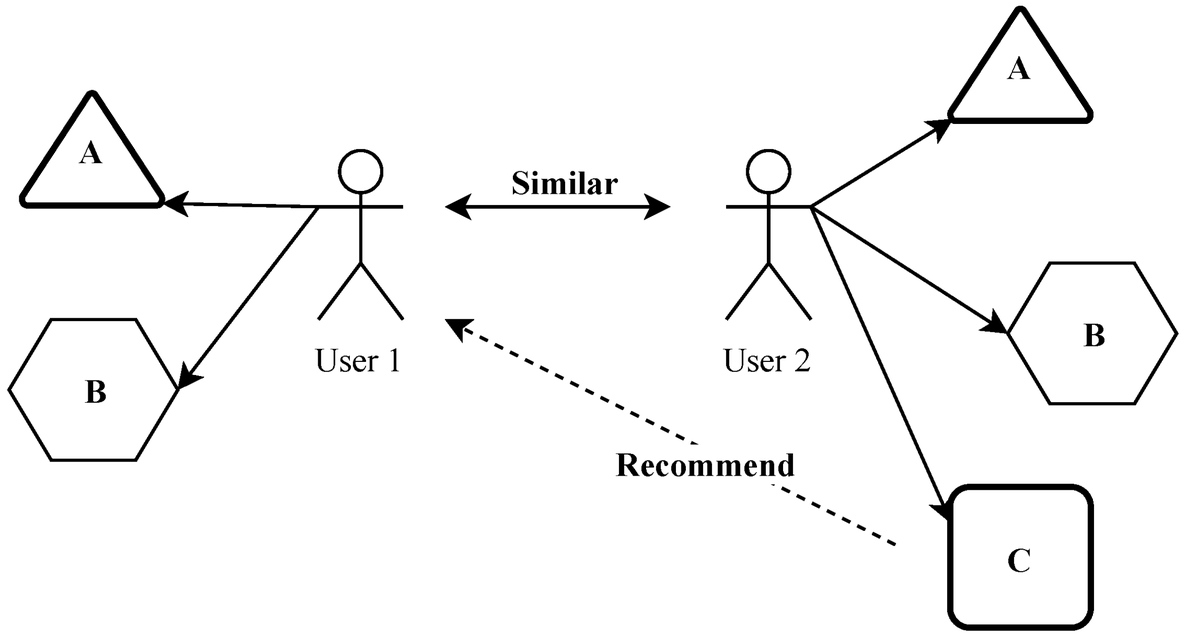
Definition

A **recommender** system is one of unsupervised learning that uses information filtering to suggest products, or content to users based on their preferences, interests, and behavior. These systems are widely used in e-commerce and online streaming settings, and other applications to help discover new products and content that may be of interest to users.

Recommender Systems

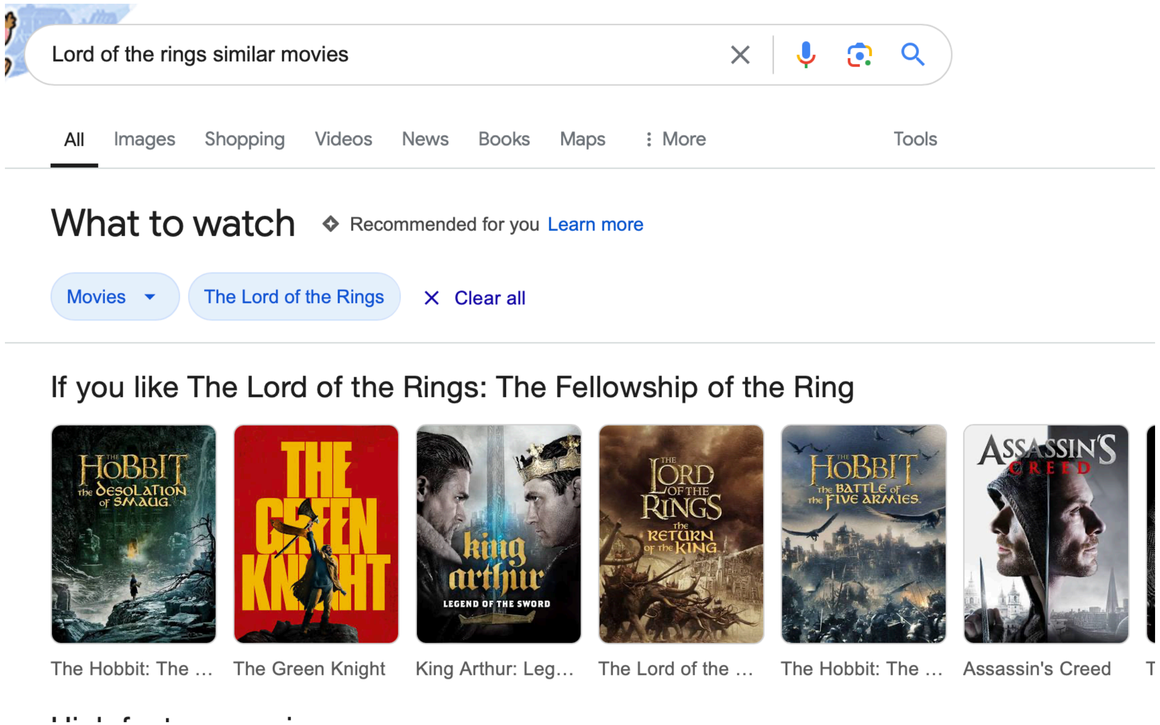
Definition

Recommender systems are trained to understand **user and product** preferences, **past decisions**, and characteristics using data **collected** about **user-product** interactions.



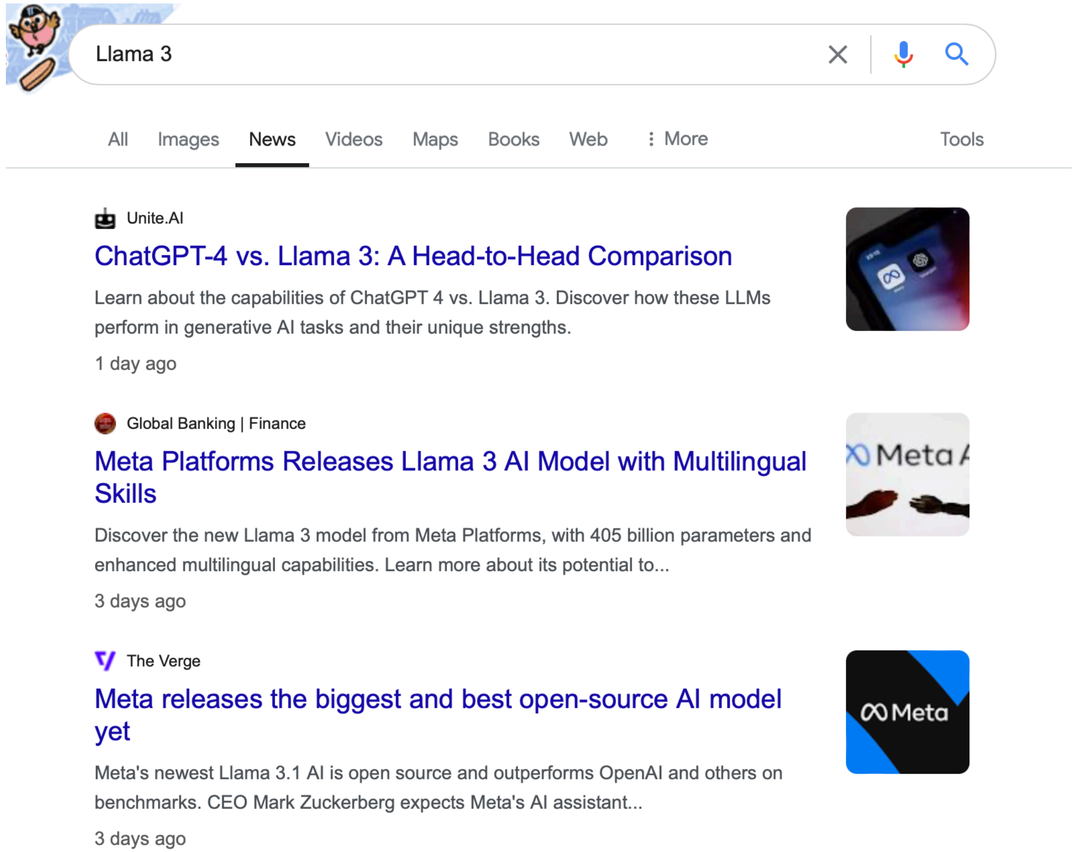
Recommender Systems

Real life example



Recommender Systems

Real life example



Recommender Systems

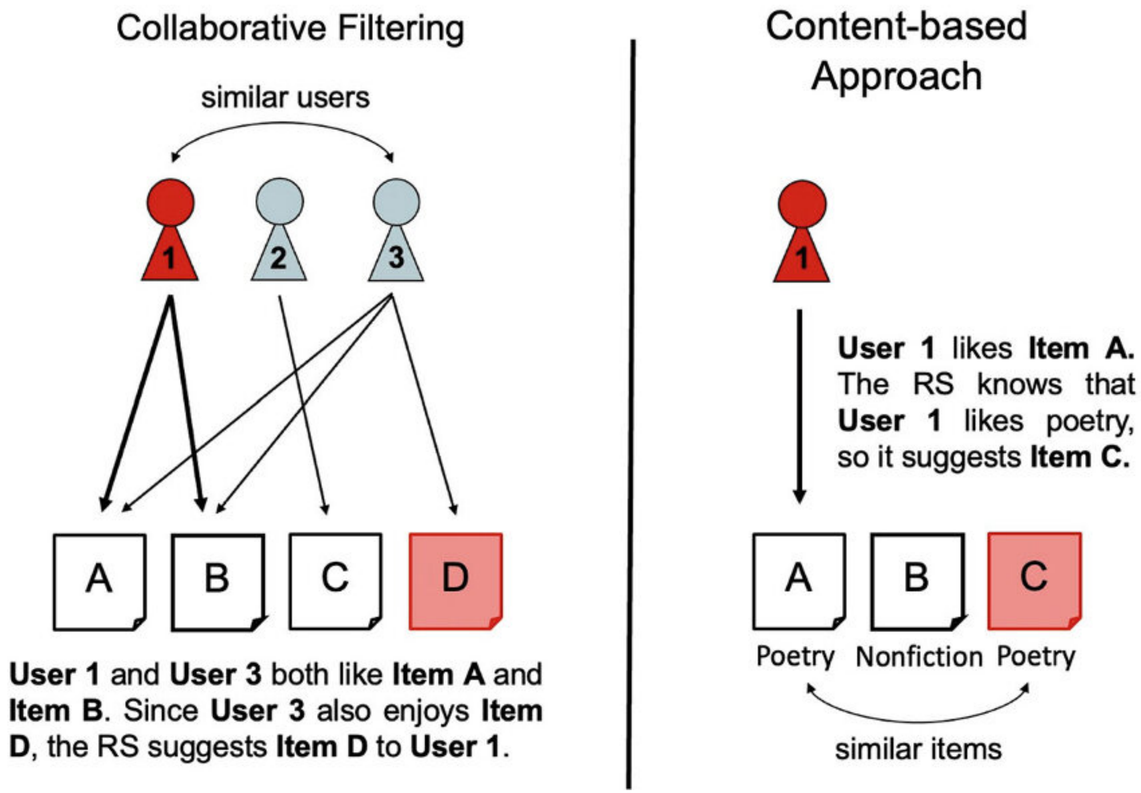
Types

We have two types:

1. Collaborative Filtering
2. Content-based Filtering

Recommender Systems

Types



Recommender Systems

Real life example

Movie/ Person	Salem	Loay	Mariam	Yara
Lord of the rings	4	2	2	1
Hunger Games	3	2	2	4
Love at last	0	4	2	1
No hard feelings	1	3	2	2
Anyone but you	?	?	?	?

How can we predict the ratings of these people for the new movie “Anyone but you”? Should we recommend it to anyone of them?

Let n_u = number of users

n_m = number of movies

r(I,j) = 1 if user j has rated movie j

y(I,j) = rating given by user j to movie I (defined only if r(I,j) = 1

Recommender Systems

Real life example

Movie/ Person	Salem	Loay	Mariam	Yara
Lord of the rings	4	2	2	1
Hunger Games	3	2	2	4
Love at last	0	4	2	1
No hard feelings	1	3	2	2
Anyone but you	?	?	?	?

Let n_u = number of users

n_m = number of movies

r(I,j) = 1 if user j has rated movie j

y(I,j) = rating given by user j to movie I (defined only if r(I,j) = 1

So, for example:
- n_u = 4
- n_m = 5
- r(1,1) = 1
- r(5,1) = 0 (Since Salem didn't rate movie 5)

Recommender Systems

Real life example - What if we added more features?

Movie/ Person	Salem	Loay	Mariam	Yara	X1 (Romance)	X2 (action)
Lord of the	4	2	2	1	0.9	0.1
Hunger	3	2	2	4	1.0	0
Love at last	0	4	2	1	0.99	0.01
No hard	1	3	2	2	0.1	1.0
Anyone but	?	?	?	?	0	0.9

Then, we add a new variable called n:
N = number of genres hence n = 2

Let x_1 = $\begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix}$

Let x_3= $\begin{bmatrix} 0.99 \\ 0.01 \end{bmatrix}$

For user 1: predict rating for movie I such as :
w.x(i) + b ***<- this is only linear regression***

For instance: $x^{(3)} = \begin{bmatrix} 0.99 \\ 0.01 \end{bmatrix}$ $w^1 = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$ $b^1 = 0$

$$w^1 . x^{(3)} + b^{(1)} = 4.95$$

Recommender Systems

Cost Function

Notation:

$r(i,j) = 1$ if user j has rated movie i (0 otherwise)
$y^{(i,j)}$ = rating given by user j on movie i (if defined)
$w^{(j)}$, $b^{(j)}$ = parameters for user j
$x^{(i)}$ = feature vector for movie i

For user j and movie i, predict rating: $w^{(j)} \cdot x^{(i)} + b^{(j)}$
$m^{(j)}$ = no. of movies rated by user j
To learn $w^{(j)}$, $b^{(j)}$

Recommender Systems

Cost Function

Notation:

r(*i*,*j*) = 1 if user *j* has rated movie *i* (0 otherwise)
$y^{(i,j)}$ = rating given by user *j* on movie *i* (if defined)
$w^{(j)}$, $b^{(j)}$ = parameters for user *j*
$x^{(i)}$ = feature vector for movie *i*

For user *j* and movie *i*, predict rating: $w^{(j)} \cdot x^{(i)} + b^{(j)}$
$m^{(j)}$ = no. of movies rated by user *j*
To learn $w^{(j)}$, $b^{(j)}$

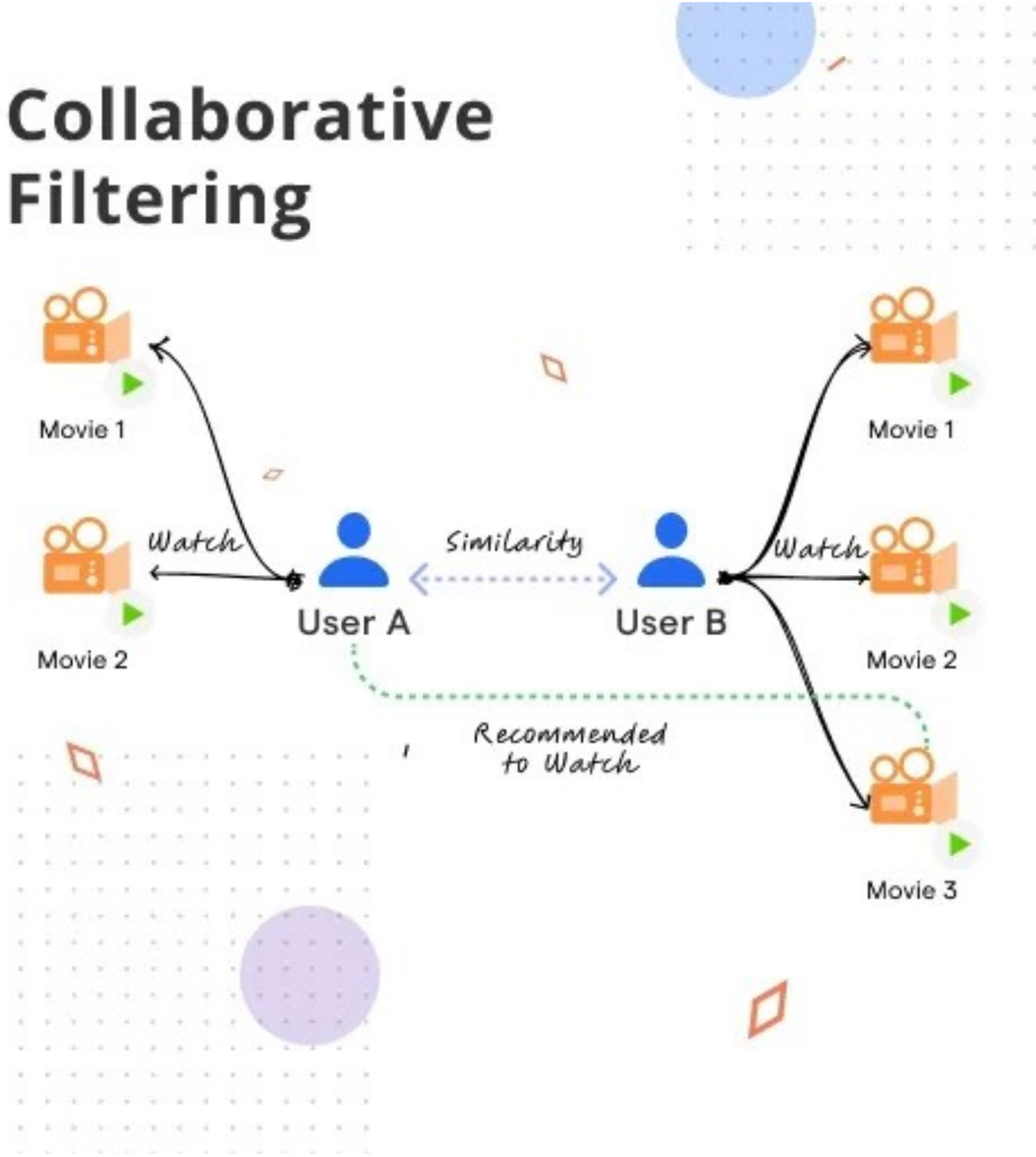
$$\min_{w^{(j)}b^{(j)}} J\left(w^{(j)}, b^{(j)}\right) = \frac{1}{2m^{(j)}} \sum_{i:r(i,j)=1} \left(w^{(j)} \cdot x^{(i)} + b^{(j)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2m^{(j)}} \sum_{k=1}^{n} \left(w_k^{(j)}\right)^2$$

Recommender Systems
Collaborative Filtering

Collaborative-based Filtering

Definition

Collaborative filtering is a technique that can filter out items that a user might like on the basis of reactions by similar users.



Recommender Systems
Content-based Filtering

Content-based Filtering

Definition

It recommends items to users according to individual item features.

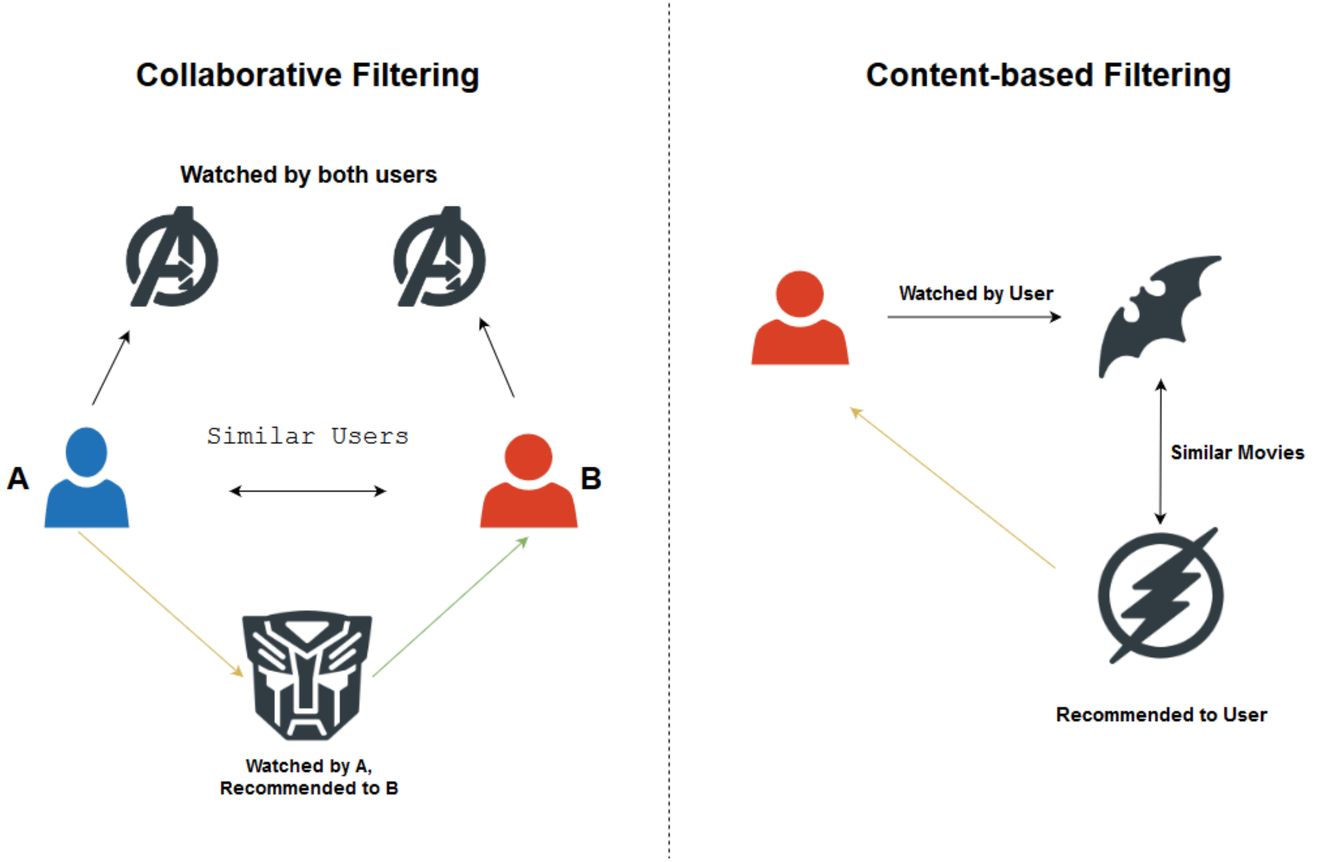
Example:

If you went to Amazon store online and chose 3 topics about machine learning, next time it will recommend for you other books related to machine learning.

Content-based Filtering

Definition

It recommends items to users according to individual item features.



Content-based Filtering

Definition

It recommends items to users according to individual item features.

Another Example:

If you enter on YouTube: How to install python.

Almost all next recommendations in YouTube home will about Python Programming Language.

Content-based Filtering

Definition

It recommends items to users according to individual item features.

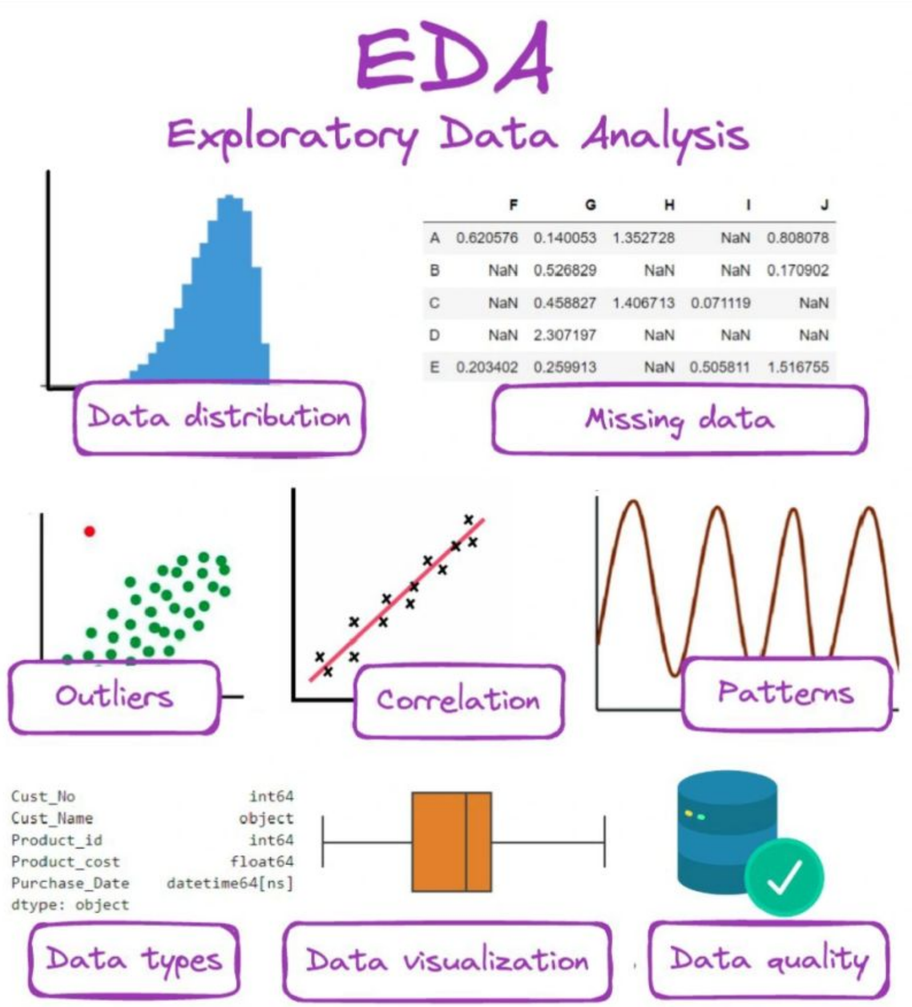
Hence, It’s recommending content **based on your content.**

EDA: Exploratory Data Analysis

EDA: Exploratory Data Analysis

Definition

Exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods.



EDA: Exploratory Data Analysis

Definition

It determines how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions.

EDA is primarily used to see what data can reveal beyond the formal modeling or hypothesis testing task and provides a provides a better understanding of data set variables and the relationships between them. It can also help determine if the statistical techniques you are considering for data analysis are appropriate. Originally developed by American mathematician John Tukey in the 1970s, EDA techniques continue to be a widely used method in the data discovery process today.

EDA: Exploratory Data Analysis

Definition

EDA is primarily used to see what data can reveal beyond the formal modeling or hypothesis testing task and provides a provides a better understanding of data set variables and the relationships between them. It can also help determine if the statistical techniques you are considering for data analysis are appropriate. Originally developed by American mathematician John Tukey in the 1970s, EDA techniques continue to be a widely used method in the data discovery process today.

EDA: Exploratory Data Analysis

Why do we need it?

In order to understand our data and reveal the its hidden gems.

Remember, Data speaks itself.

As a strong AI Engineer and/or Data Scientist, you must listen carefully to your data.

EDA: Exploratory Data Analysis

Is it part of AI Engineer's job or Data Scientist's Job?

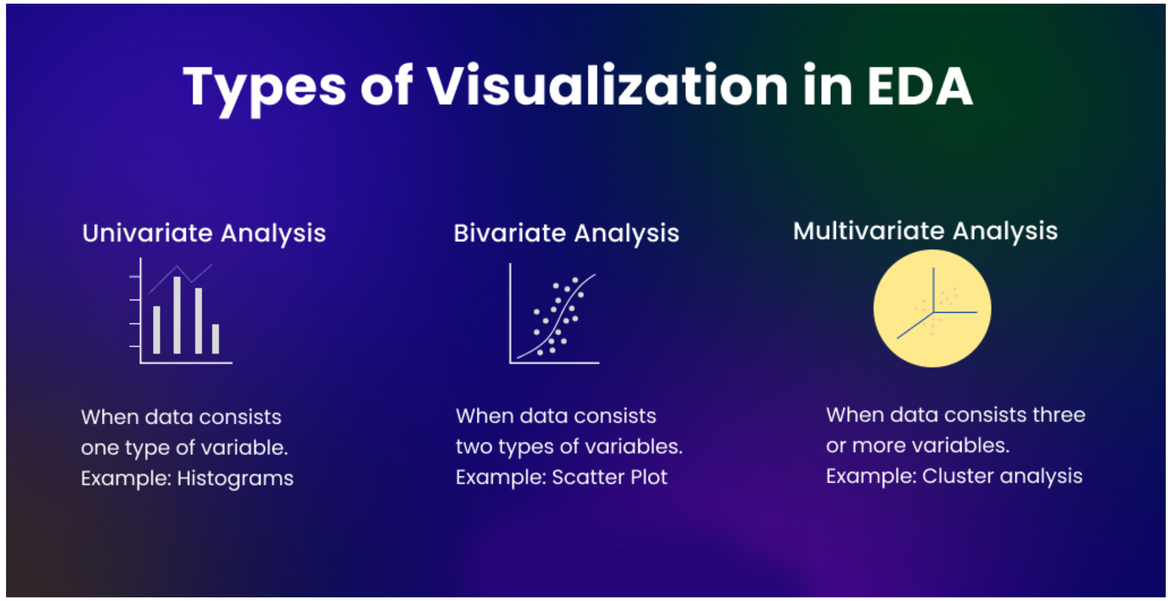
Both!
Can you be an AI Engineer without understanding the problem of your data and try to find/create the best model for it? No

Can you be a Data Scientist preparing your data with no clue of the next steps? No

EDA: Exploratory Data Analysis

Types

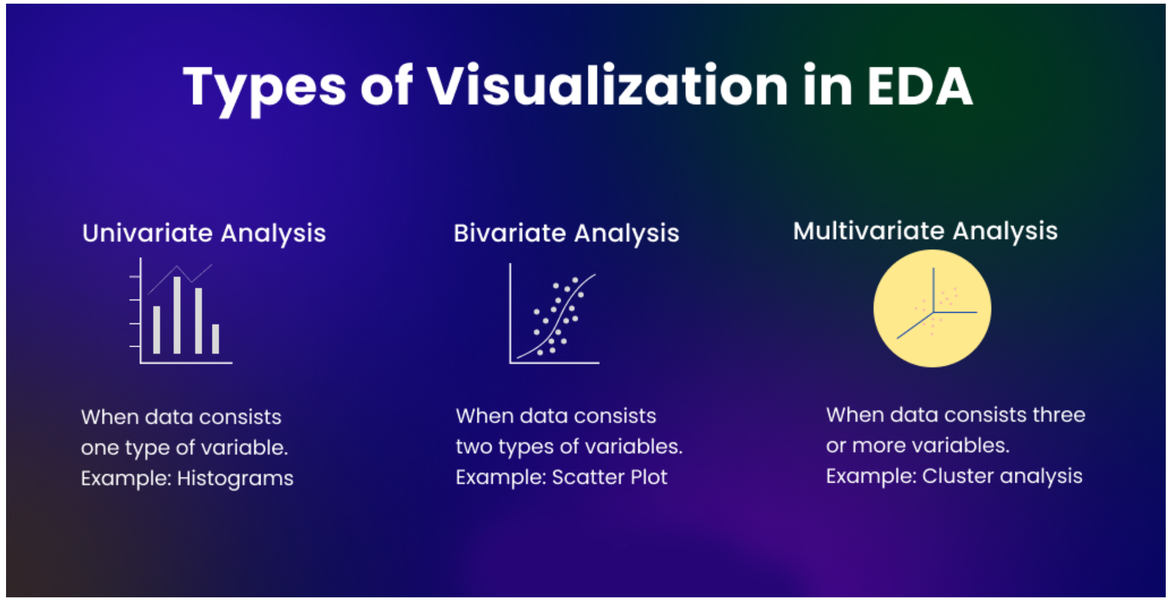
- Univariate Analysis: It has two sections:
    - Non-graphical methods: This is simplest form of data analysis, where the data being analyzed consists of just one variable. Since it’s a single variable, it doesn’t deal with causes or relationships.The main purpose of univariate analysis is to describe the data and find patterns that exist within it.
    - Graphical Methods: non-graphical methods don’t provide a full picture of the data. Graphical methods are therefore required.



EDA: Exploratory Data Analysis

Types

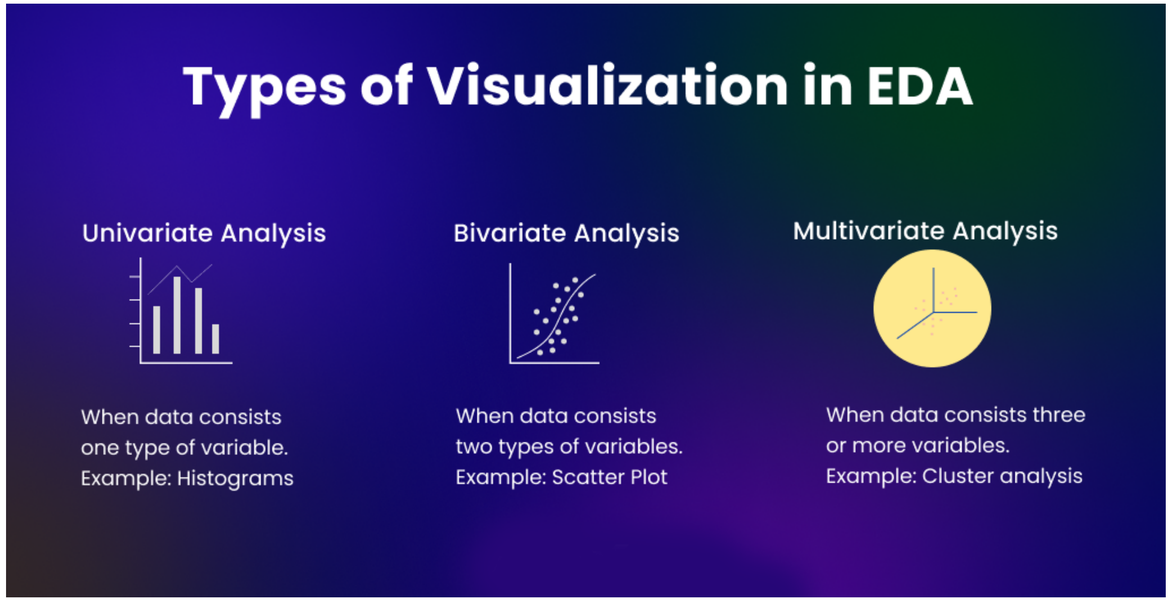
- Bivariate Analysis: Bivariate evaluation involves exploring the connection between variables. It enables find associations, correlations, and dependencies between pairs of variables. Some techniques used:
  - Scatter Plot: A scatter plot helps visualize the relationship between two continuous variables.
  - Correlation Coefficients: This statistical measure quantifies the degree to which two variables are related.



EDA: Exploratory Data Analysis

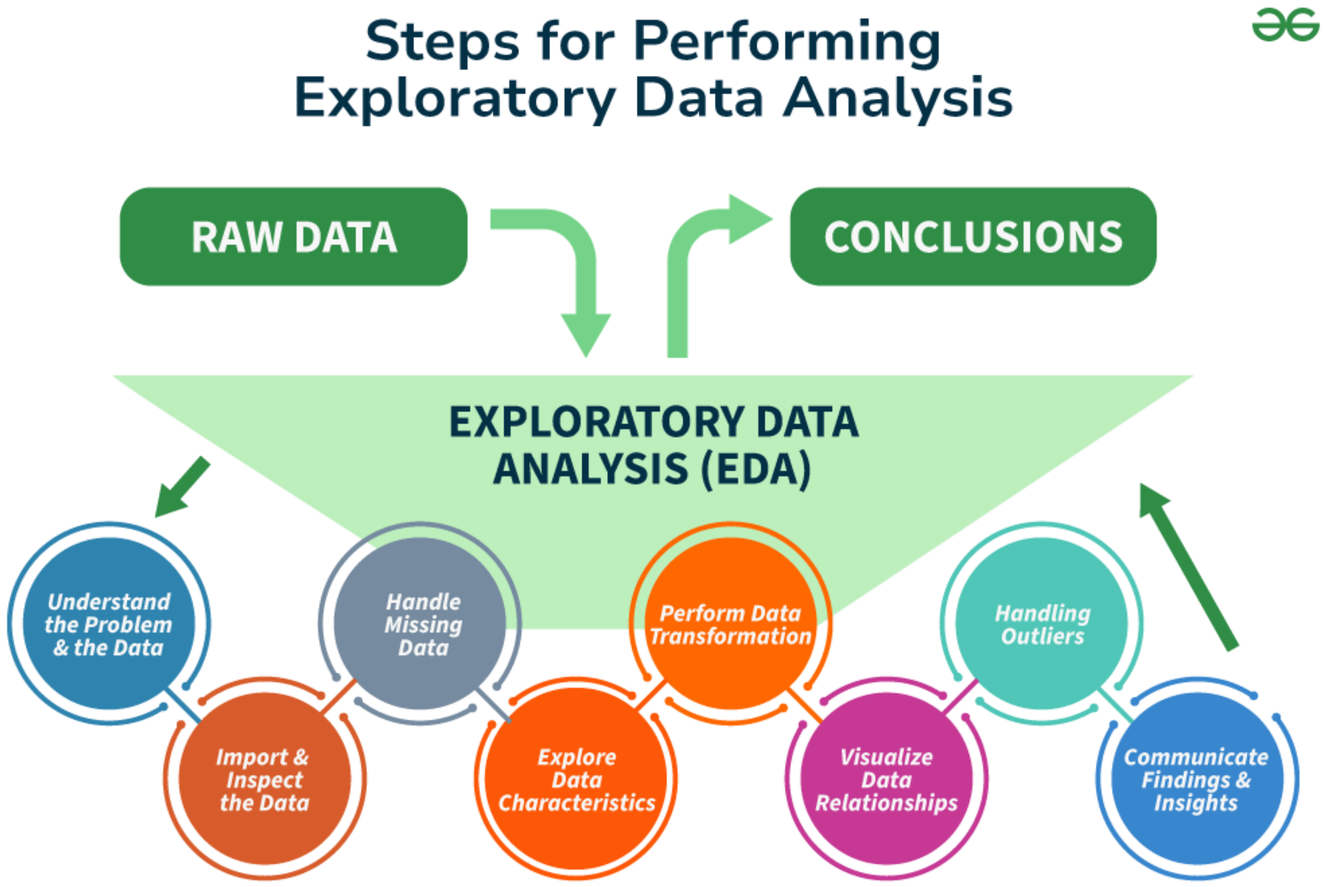
Types

- Multivariate Analysis: It examines the relationships between two or more variables in the dataset. It aims to understand how variables interact with one another, which is crucial for most statistical modeling techniques. Techniques include:
  - Pair Plots
  - PCA (Principle Component Analysis)



EDA: Exploratory Data Analysis

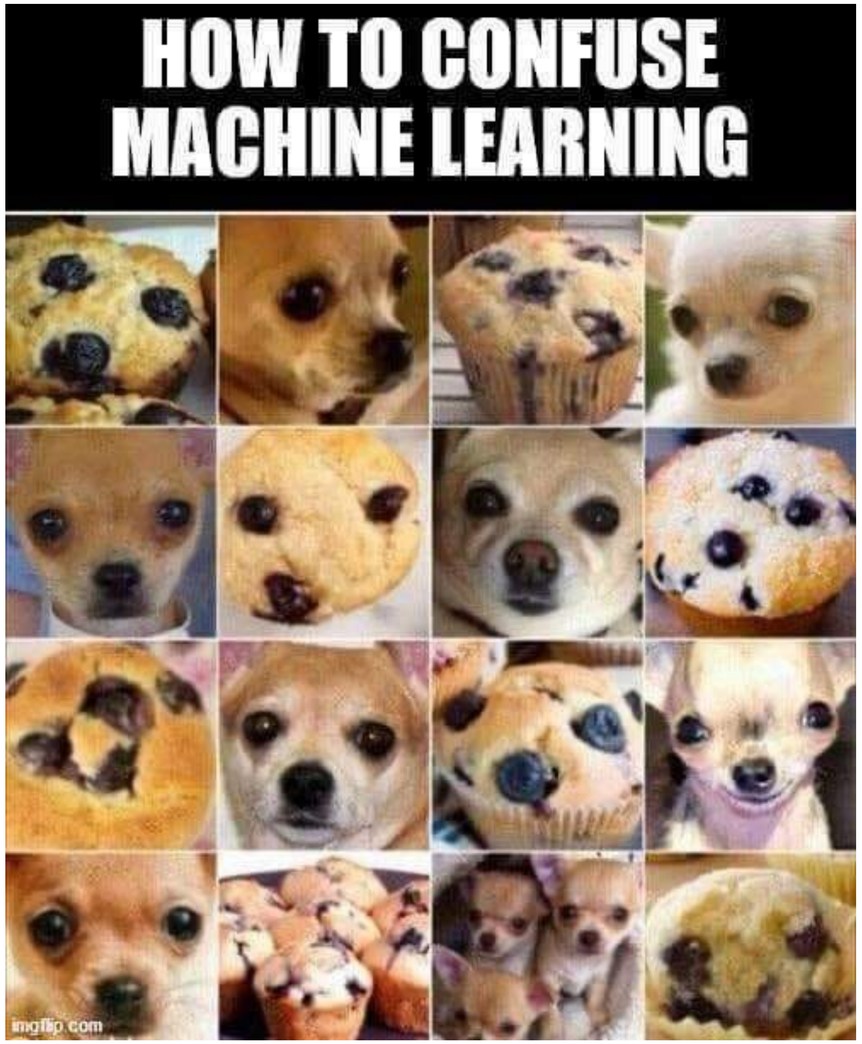
Steps:



EDA: Exploratory Data Analysis

Outliers

An outlier is a single data point that goes far outside the average value of a group of statistics.



EDA: Exploratory Data Analysis

Outliers: How to detect them?

Below are some of the techniques
of detecting outliers

- Z-score
- Boxplots
- Inter Quantile Range(IQR)

Let’s take this data as example: [15, 101, 18, 7, 13, 16, 11, 21, 5, 15, 10, 9].

Which one is the outlier?

EDA: Exploratory Data Analysis

Outliers: How to detect them?

Below are some of the techniques
of detecting outliers

- Z-score
- Boxplots
- Inter Quantile Range(IQR)

Let’s take this data as example: [15, 101, 18, 7, 13, 16, 11, 21, 5, 15, 10, 9].

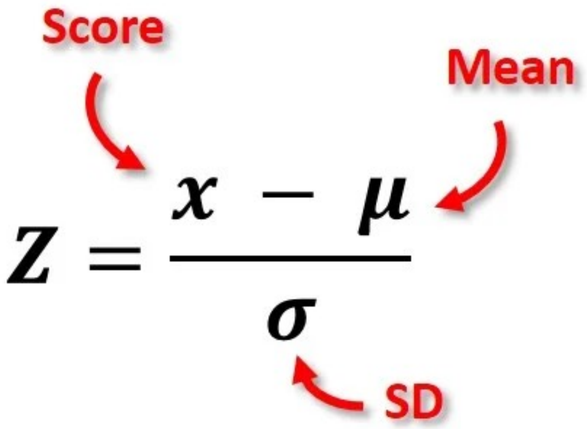
Which one is the outlier? **101**

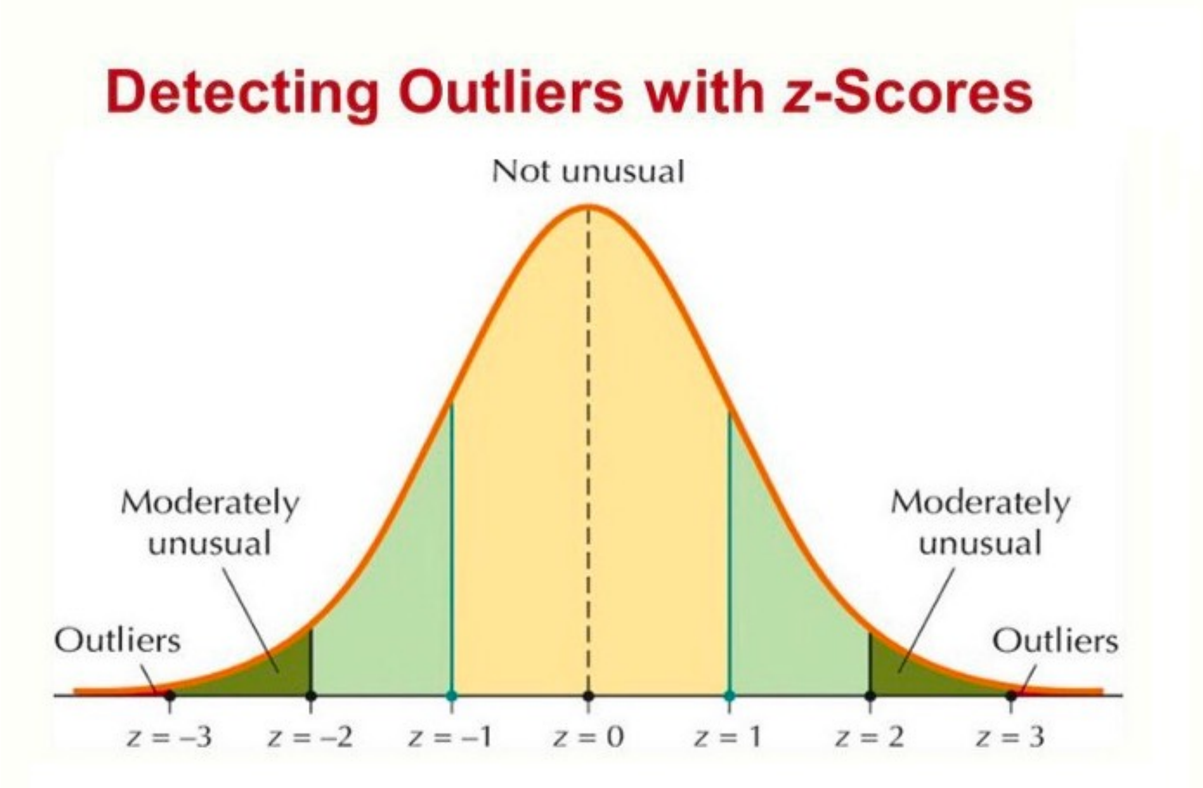
Let’s check if our answer is right with the methods above!

EDA: Exploratory Data Analysis

Outliers: Z-scores

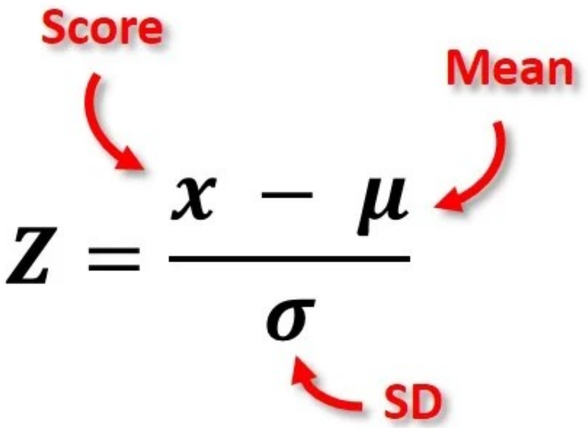
Criteria: any data point whose Z-score falls out of 3rd standard deviation is an outlier treatment.





EDA: Exploratory Data Analysis

Outliers: Z-scores



[15, 101, 18, 7, 13, 16, 11, 21, 5, 15, 10, 9]

Steps:

- loop through all the data points and compute the Z-score using the formula (Xi-mean)/std.
- define a threshold value of 3 and mark the datapoints whose absolute value of Z-score is greater than the threshold as outliers.

```
import numpy as np
outliers = []
def detect_outliers_zscore(data):
    thres = 3
    mean = np.mean(data)
    std = np.std(data)
    # print(mean, std)
    for i in data:
        z_score = (i-mean)/std
        if (np.abs(z_score) > thres):
            outliers.append(i)
    return outliers# Driver code
sample_outliers = detect_outliers_zscore(sample)
print("Outliers from Z-scores method: ", sample_outliers)
"""
Output:
Outliers from Z-scores method:  [101]
"""
```

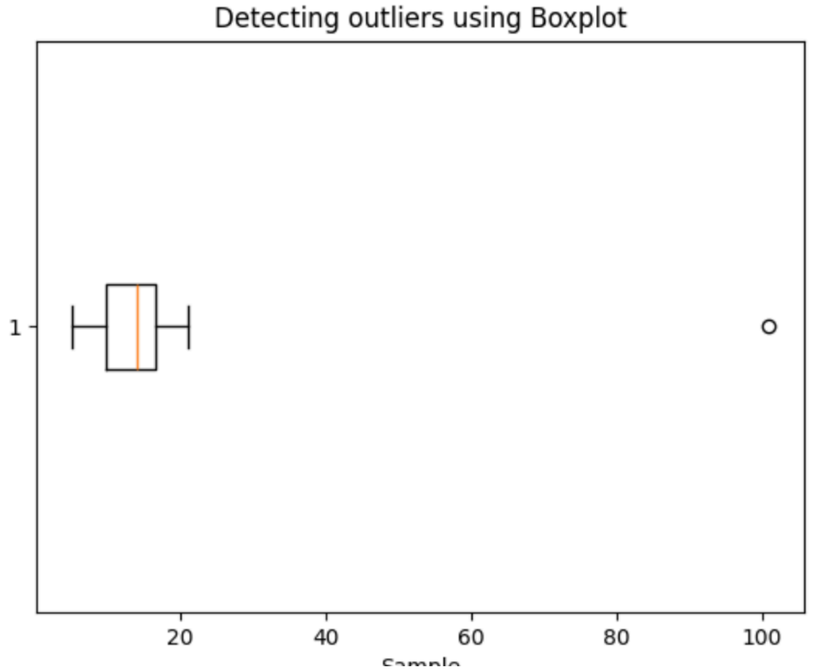
EDA: Exploratory Data Analysis

Outliers: Box Plots

Criteria: any point that is outside of the box plot is considered as an outlier

```python
import matplotlib.pyplot as plt

sample= [15, 101, 18, 7, 13, 16, 11, 21, 5, 15, 10, 9]
plt.boxplot(sample, vert=False)
plt.title("Detecting outliers using Boxplot")
plt.xlabel('Sample')
plt.show()
```



EDA: Exploratory Data Analysis

Outliers: Inter Quantile Range(IQR)

Criteria: data points that lie 1.5 times of IQR above Q3 and below Q1 are outliers. This shows in detail about outlier treatment in Python.

Percentile: Check's how many nth percentage of your data is below a specific value.

Example:

If your dataset has all data below number 4, hence the 100% *percentile* will be something close to 4.

If your dataset has half of its elements below 2, then the 50% *percentile* will be something close to 2.

EDA: Exploratory Data Analysis

Outliers: Inter Quantile Range(IQR)

Criteria:data points that lie 1.5 times of IQR above Q3 and below Q1 are outliers. This shows in detail about outlier treatment in Python.

```
outliers = []
def detect_outliers_iqr(data):
    data = sorted(data)
    q1 = np.percentile(data, 25)
    q3 = np.percentile(data, 75)
    # print(q1, q3)
    IQR = q3-q1
    lwr_bound = q1-(1.5*IQR)
    upr_bound = q3+(1.5*IQR)
    # print(lwr_bound, upr_bound)
    for i in data:
        if (i<lwr_bound or i>upr_bound):
            outliers.append(i)
    return outliers# Driver code
sample_outliers = detect_outliers_iqr(sample)
print("Outliers from IQR method: ", sample_outliers)
"""
Output:
Outliers from IQR method:  [101]
"""
```

EDA: Exploratory Data Analysis

How can I get rid of Outliers? Enter Feature Engineering

1. Data Trimming
2. Replace outliers with mean/ median values.
3. Remove the outliers based on the IQR method.

Feature Engineering is coming soon in the future courses!

EDA: Exploratory Data Analysis

Does it always work?

No! Sometimes we need to keep them, to avoid trimming important features from the dataset (More on this in the deep learning course)

Let’s Skip to the GOOD PART

The next course will provide hands-on notebooks on all the topics we discussed. See you there :)