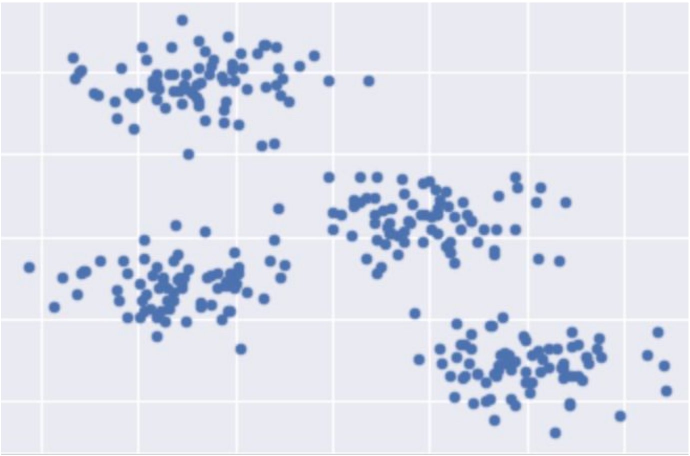
Clustering

Introduction to Clustering

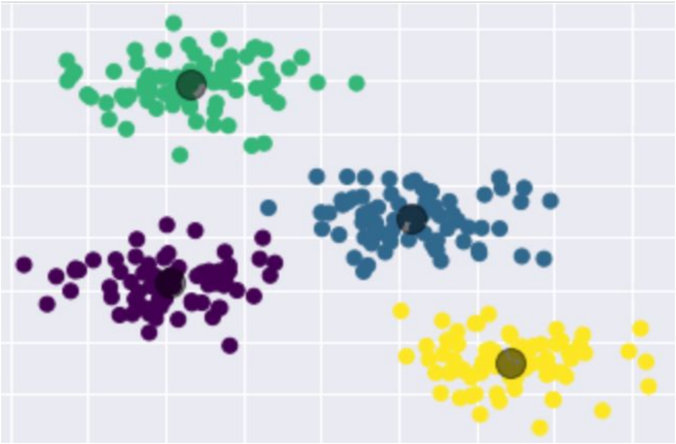
Prof. Dr. Jan Kirenz
HdM Stuttgart

Discover unknown subgroups in data.

Unlabeled examples (observations)



Grouping unlabeled examples is called clustering.



Clustering is **unsupervised learning**

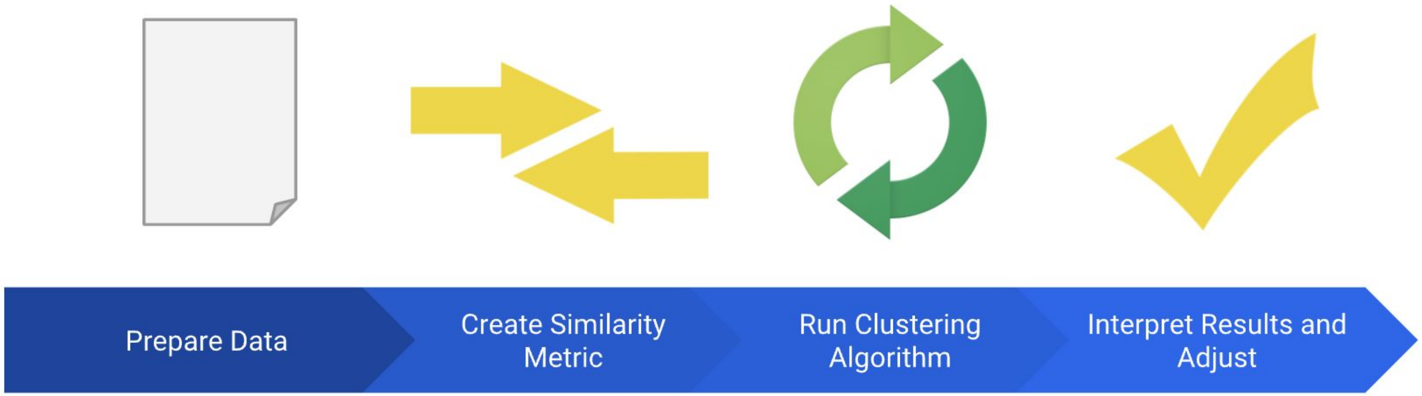
The goal is to discover interesting things about the observations:

- is there an informative way to **visualize** the data?
- Can we **discover subgroups** among the variables or among the observations?

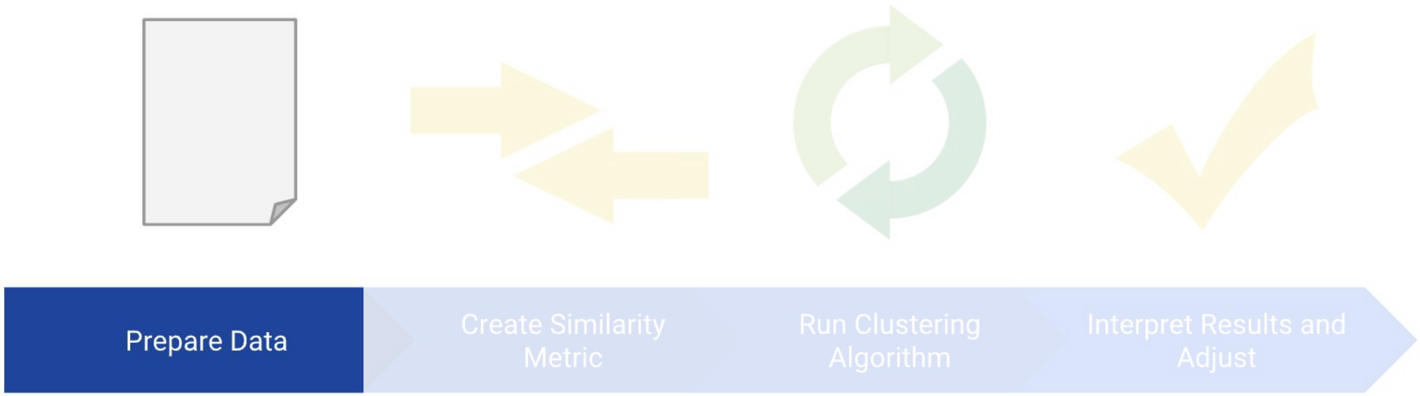
Use cases for cluster analysis

- **Customer segmentation**
  - understanding different customer segments to devise marketing strategies
- **Recommender systems**
  - grouping together users with similar viewing patterns on Netflix, in order to recommend similar content
- **Anomaly detection**
  - fraud detection, detecting defective mechanical parts

To cluster your data, you'll follow these steps:



To cluster your data, you'll follow these steps:



First of all,
you have to exclude all
missing values and
outliers

Then, we **normalize** data

If you only have a few values, you could use this simple rule:

$$x' = \frac{x}{\max(x)}$$

Normalizing data by **rescaling** (min-max normalization)

Typically, we use rescaling like this:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Rescalin results in a range [0, 1]

Data standardization with **z-score**

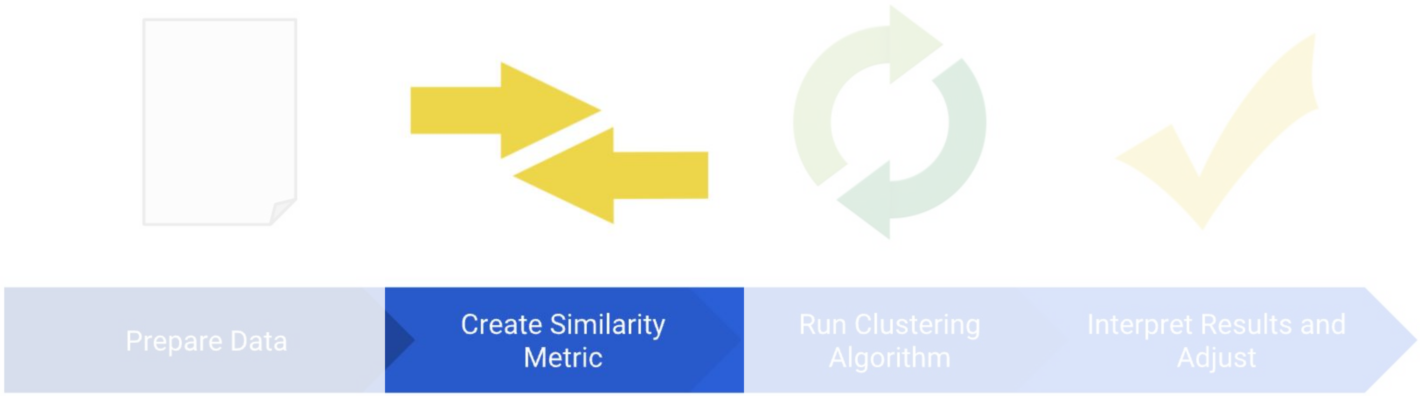
Another alternative (especially if data has a normal distribution):

$$x' = (x - \mu)/\sigma$$

$$\text{where:} \quad \mu = \text{mean}$$

$$\sigma = \text{standard deviation}$$

To cluster your data, you'll follow these steps:



How to create a similarity measure for a numeric feature?

- Feature X_1: shoe size

Shoe A

size: 8

Shoe B

size: 11

Simple similarity measure

- Feature X_1: shoe size

Picture source: Allbirds

How to create a manual **similarity measure** for two numeric features?

- Feature X_1: shoe size (numeric)
- Feature X_2: price (numeric)

size: 8
price: 120

size: 11
price: 150

Picture source: Allbirds

Similarity measure for two numeric features

- Feature X_1: shoe size (numeric)
- Feature X_2: price (numeric)

Picture source: Allbirds

Create a manual similarity measure for two numeric features

Since we don't have enough data, we'll simply scale the data without normalizing

Action	Method
Scale the size.	Assume a maximum possible shoe size of 20. Divide 8 and 11 by the maximum size 20 to get 0.4 and 0.55.
Scale the price.	Divide 120 and 150 by the maximum price 150 to get 0.8 and 1.
Find the difference in size.	$0.55 - 0.4 = 0.15$
Find the difference in price.	$1 - 0.8 = 0.2$
Find the RMSE.	$\sqrt{\frac{0.2^2+0.15^2}{2}} = 0.17$

$$x' = \frac{x}{\max(x)}$$

Create a manual similarity measure for two numeric features

Action	Method
Scale the size.	Assume a maximum possible shoe size of 20. Divide 8 and 11 by the maximum size 20 to get 0.4 and 0.55.
Scale the price.	Divide 120 and 150 by the maximum price 150 to get 0.8 and 1.
Find the difference in size.	$0.55 - 0.4 = 0.15$
Find the difference in price.	$1 - 0.8 = 0.2$
Find the RMSE.	$\sqrt{\frac{0.2^2+0.15^2}{2}} = 0.17$

$$x' = \frac{x}{\max(x)}$$

Create a manual similarity measure for two numeric features

Action	Method
Scale the size.	Assume a maximum possible shoe size of 20. Divide 8 and 11 by the maximum size 20 to get 0.4 and 0.55.
Scale the price.	Divide 120 and 150 by the maximum price 150 to get 0.8 and 1.
Find the difference in size.	$0.55 - 0.4 = 0.15$
Find the difference in price.	$1 - 0.8 = 0.2$
Find the RMSE.	$\sqrt{\frac{0.2^2+0.15^2}{2}} = 0.17$

Source: Google Developers: https://developers.google.com/machine-learning/clustering/similarity/manual-similarity

Create a manual similarity measure for two numeric features

Action	Method
Scale the size.	Assume a maximum possible shoe size of 20. Divide 8 and 11 by the maximum size 20 to get 0.4 and 0.55.
Scale the price.	Divide 120 and 150 by the maximum price 150 to get 0.8 and 1.
Find the difference in size.	$0.55 - 0.4 = 0.15$
Find the difference in price.	$1 - 0.8 = 0.2$
Find the RMSE.	$\sqrt{\frac{0.2^2+0.15^2}{2}} = 0.17$

Source: Google Developers: https://developers.google.com/machine-learning/clustering/similarity/manual-similarity

Create a manual similarity measure for two numeric features

Action	Method
Scale the size.	Assume a maximum possible shoe size of 20. Divide 8 and 11 by the maximum size 20 to get 0.4 and 0.55.
Scale the price.	Divide 120 and 150 by the maximum price 150 to get 0.8 and 1.
Find the difference in size.	$0.55 - 0.4 = 0.15$
Find the difference in price.	$1 - 0.8 = 0.2$
Find the RMSE.	$\sqrt{\frac{0.2^2+0.15^2}{2}} = 0.17$

Create a manual similarity measure for a **categorical feature**

- Feature X_3: color (categorical)

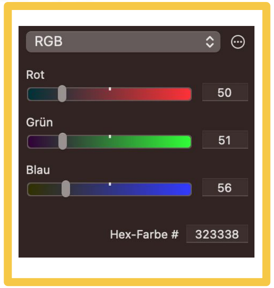
color: black

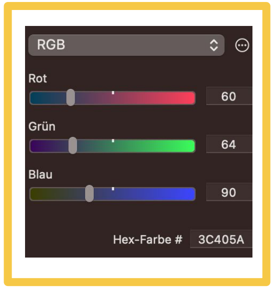
color: blue

Create a manual similarity measure for a **categorical feature**

- Feature X_3: color (categorical)

color: black





color: blue

What about **categorical** features with **multiple levels** (multivalent)

- Movie genres:
  - comedy,
  - action,
  - drama,
  - non-fiction,
  - biographical
- Can be "action" and "comedy" simultaneously, or just "action"

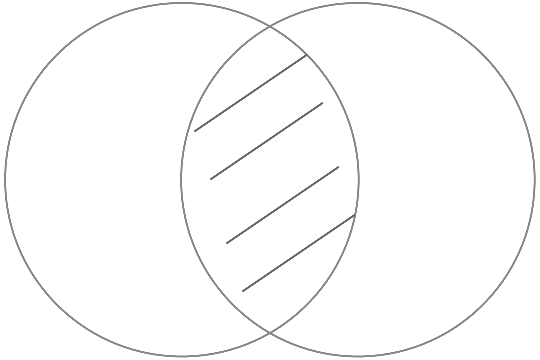
How to measure similarity?

1: [comedy, action]

2: [action, drama]

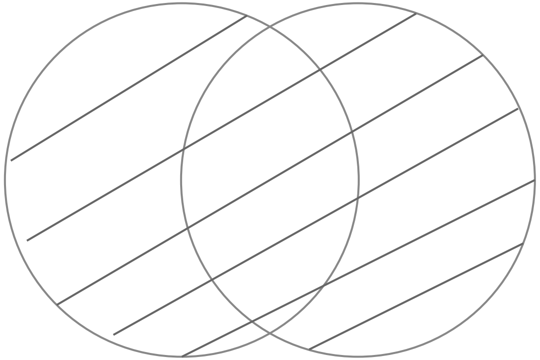
Intersection & union

Intersection



$A \cap B$

Union

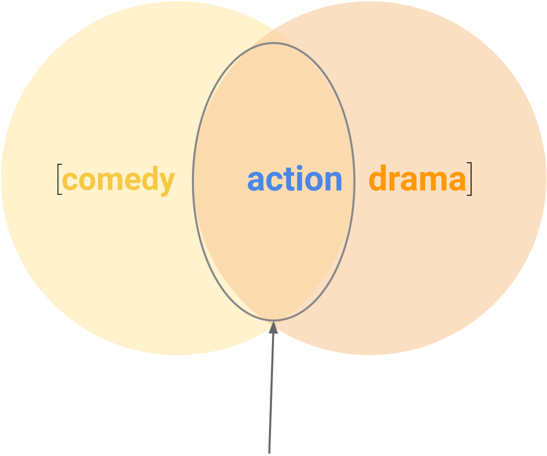


$A \cup B$

Jaccard distance

[comedy, action]

[action, drama]



Union:
[comedy, action, drama]

Intersection:
action

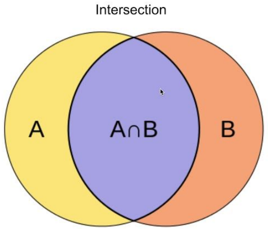
Jaccard distance

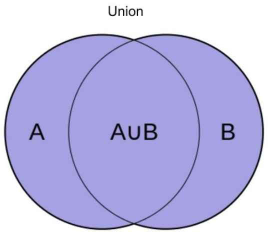
$$\frac{\textbf{Intersection} \\ \text{action}}{\textbf{Union} \\ \text{[comedy, action, drama]}}$$

$$\frac{A \cap B}{A \cup B}$$

Create a manual similarity measure for a categorical feature

- A: [“comedy”,”action”] and B: [“comedy”,”action”] = 1
- A: [“comedy”,”action”] and B: [“action”] = ½
- A: [“comedy”,”action”] and B: [“action”, "drama"] = ⅓
- A: [“comedy”,”action”] and B: [“non-fiction”,”biographical”] = 0



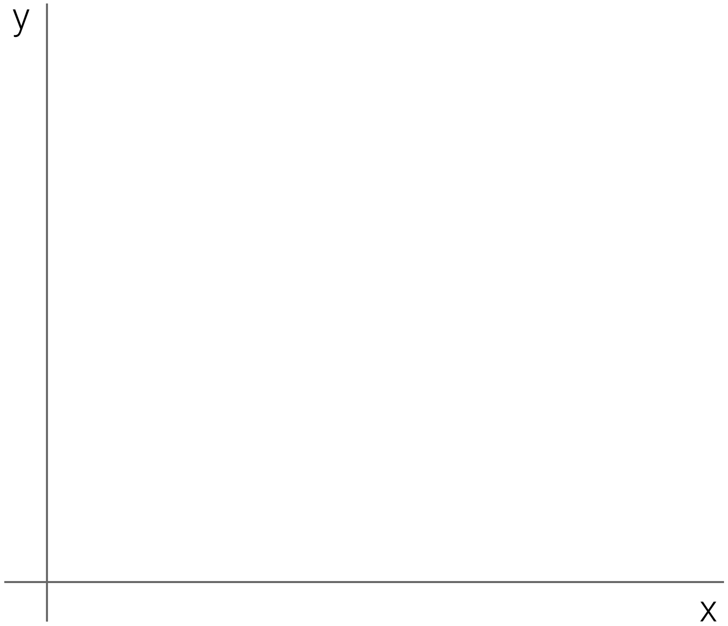


$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

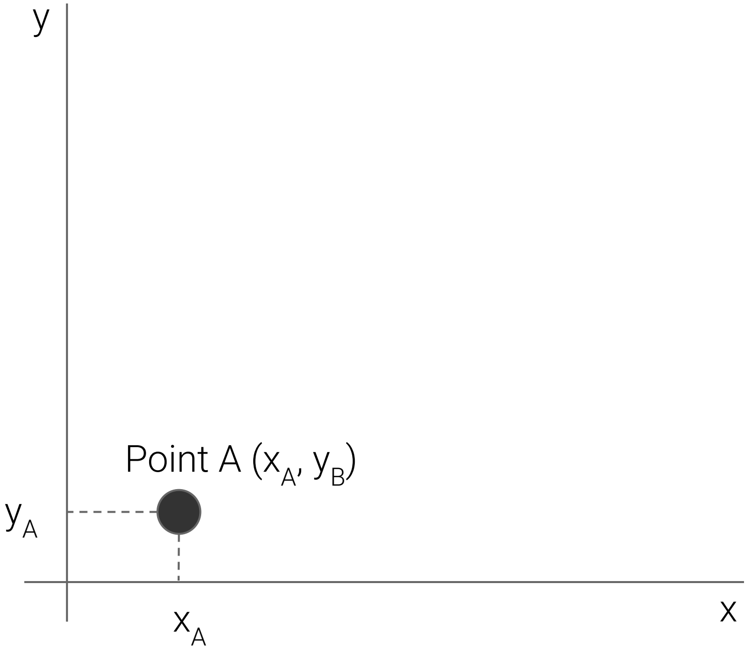
- **Jaccard similarity**. Calculate similarity using the ratio of common values
- **Jaccard distance**. Calculate distance using (1- Jaccard similarity)

Popular **distance metrics** for numerical features

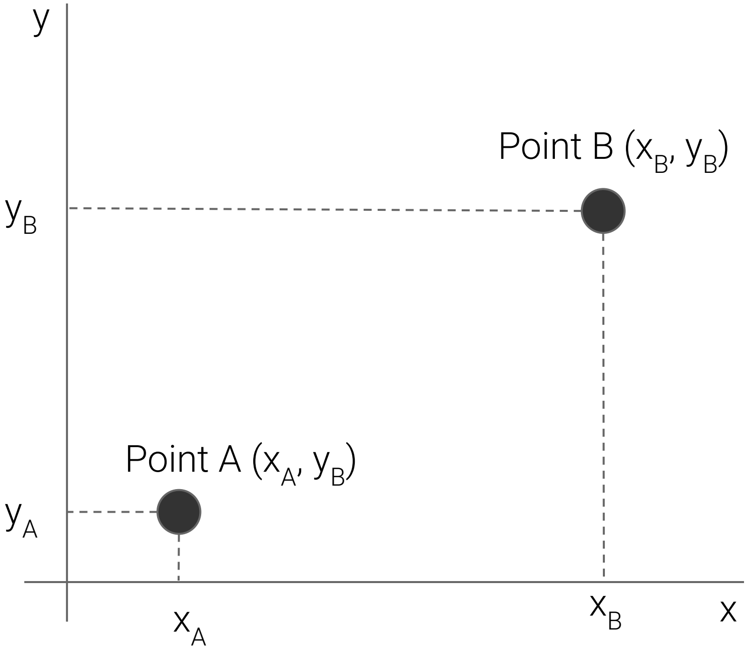
Let’s start with a simple coordinate system (CS)



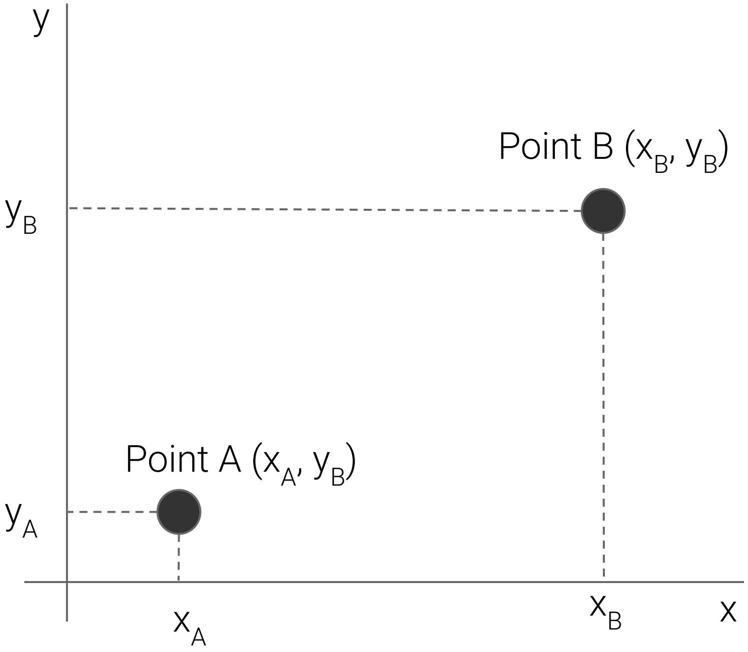
We include one observation “A”



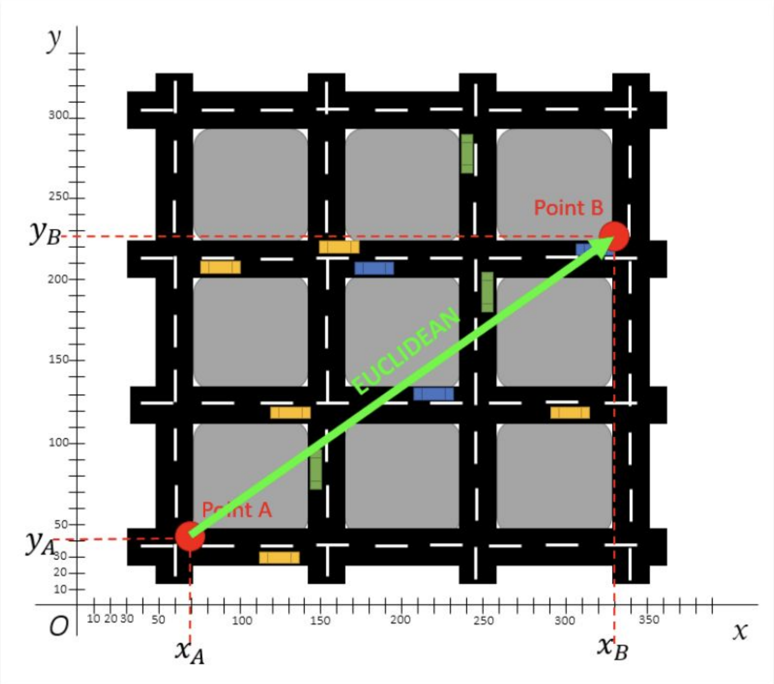
And another observation “B”



How can we measure the **distance** between A and B?



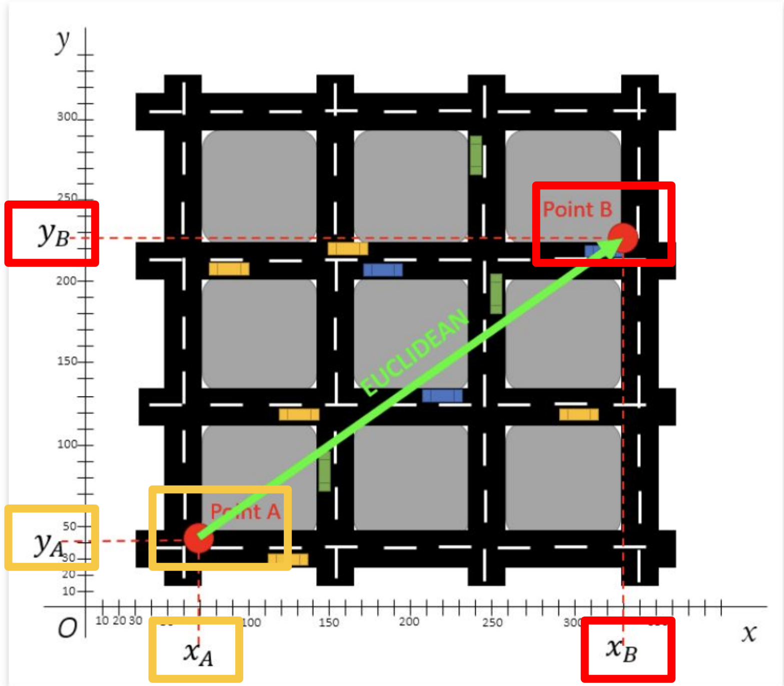
Imagine there are streets on the CS



This would be the shortest distance

$$d(A,B) = \sqrt{\sum_{i=1}^{n}(A_i - B_i)^2}$$

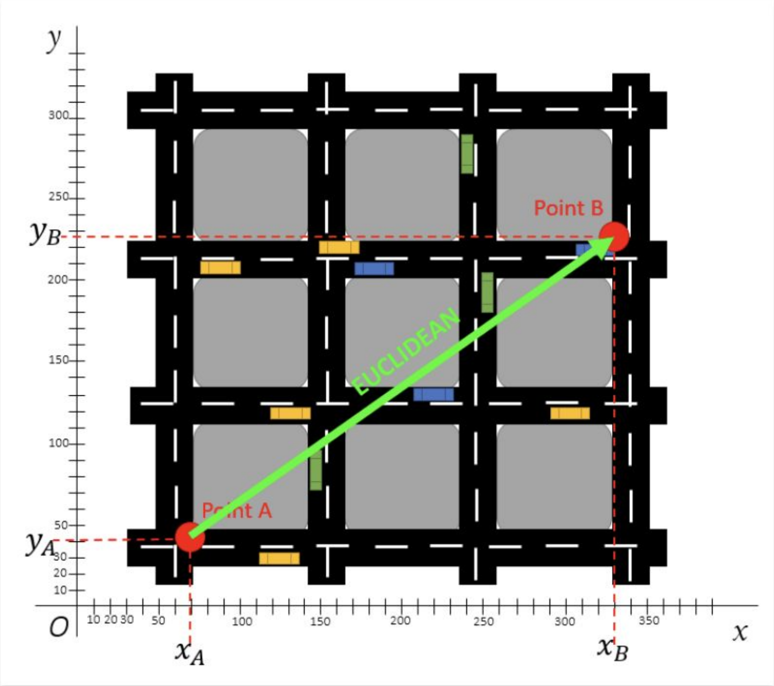
- x_A = 70
- x_B = 330
- y_A = 40
- y_B = 228



Euclidean distance (L_2 distance)

$$d(A,B) = \sqrt{\sum_{i=1}^{n}(A_i - B_i)^2}$$

- x_A = 70
- x_B = 330
- y_A = 40
- y_B = 228



$$d(A,B) = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$

$$d(A,B) = \sqrt{(70 - 330)^2 + (40 - 228)^2}$$

$$d(A,B) = \sqrt{(-260)^2 + (-188)^2}$$

$$d(A,B) = \sqrt{(76600 + 35344)}$$

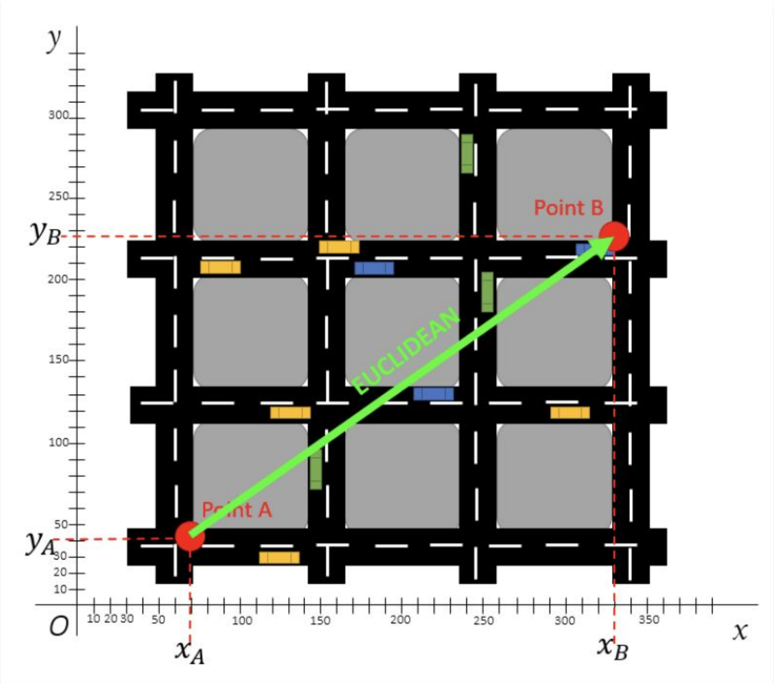
$$d(A,B) = \sqrt{(112225)}$$

$$d(A,B) = 335$$

Squared Euclidean distance (L_2)

$$d^2(A,B) = \sum_{i=1}^{n}(A_i - B_i)^2$$

- x_A = 70
- x_B = 330
- y_A = 40
- y_B = 228



$$d^2(A,B) = (x_A - x_B)^2 + (y_A - y_B)^2$$

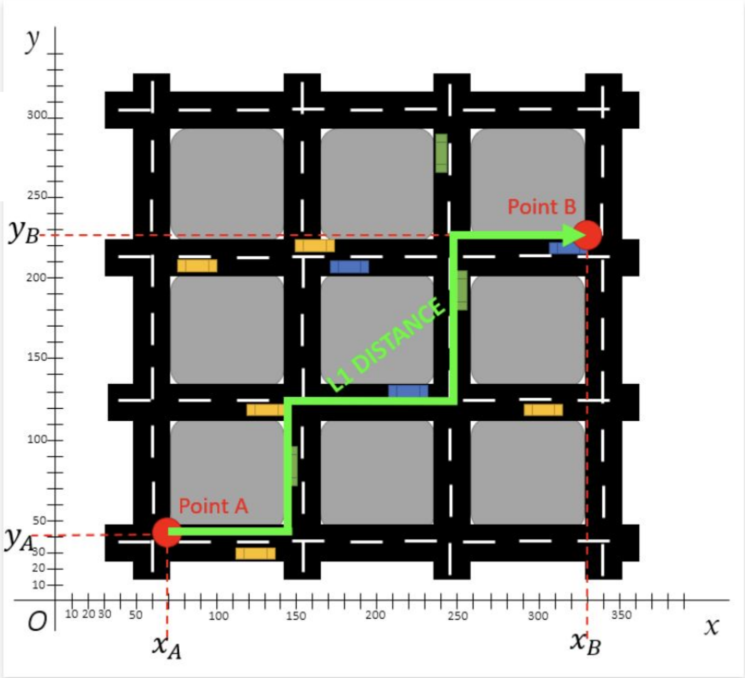
$$d^2(A,B) = (70 - 330)^2 + (40 - 228)^2$$

$$d^2(A,B) = 112225$$

L_1 distance (Manhattan distance)

$$d(A, B) = \sum_i |A_i - B_i|$$

- x_A = 70
- x_B = 330
- y_A = 40
- y_B = 228



$$d(A, B) = |x_A - x_B| + |y_A - y_B|$$

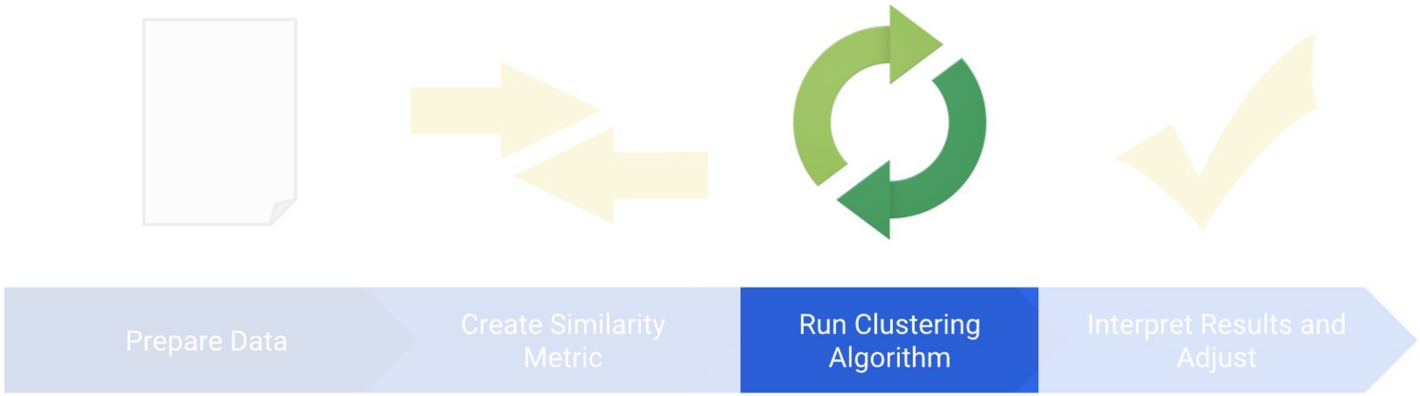
$$d(A, B) = |70 - 330| + |40 - 228|$$

$$d(A, B) = |-260| + |-188|$$

$$d(A, B) = 260 + 188$$

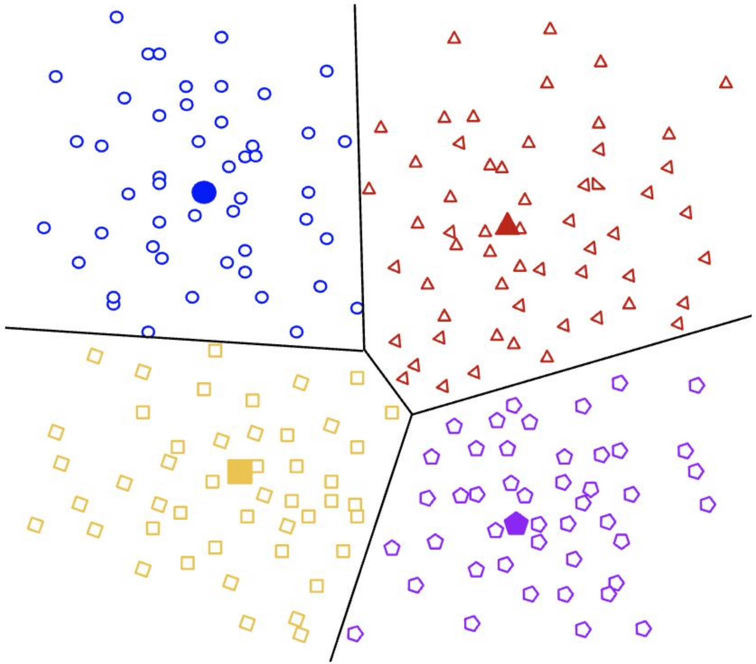
$$d(A, B) = 448$$

To cluster your data, you'll follow these steps:



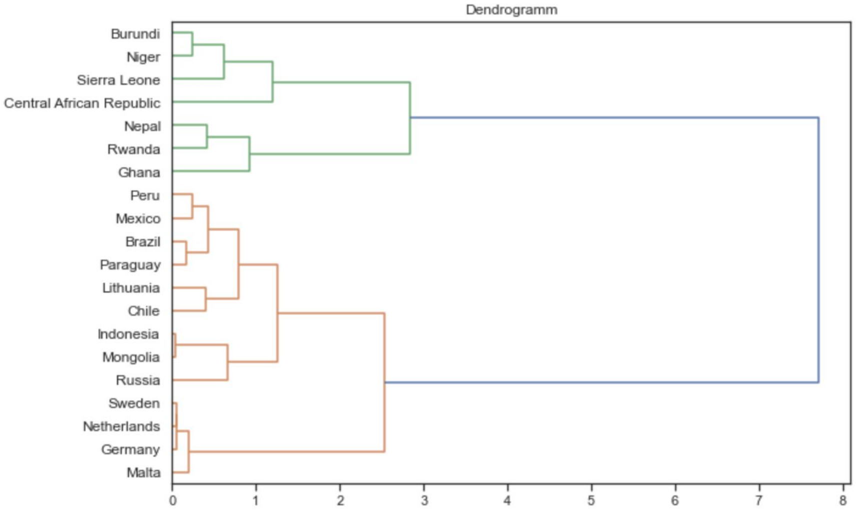
Centroid-based Clustering

- Centroid-based algorithms are **efficient**
- But sensitive to **initial conditions** and **outliers**.
- **k-means** is the most widely-used centroid-based clustering algorithm.



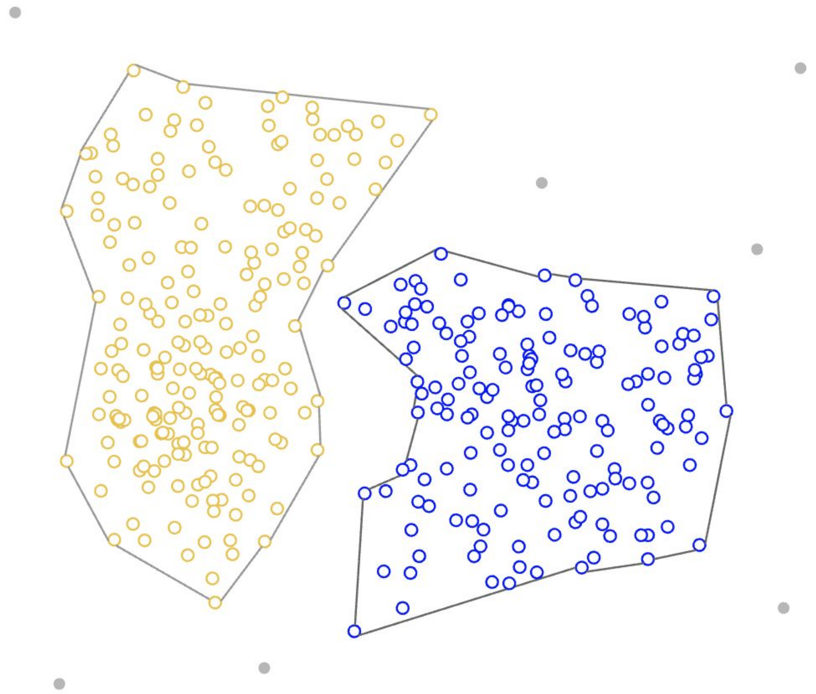
Hierarchical Clustering

- Hierarchical clustering creates a **tree** of clusters.
- One advantage is that any number of clusters can be chosen by **cutting the tree** at the right level.

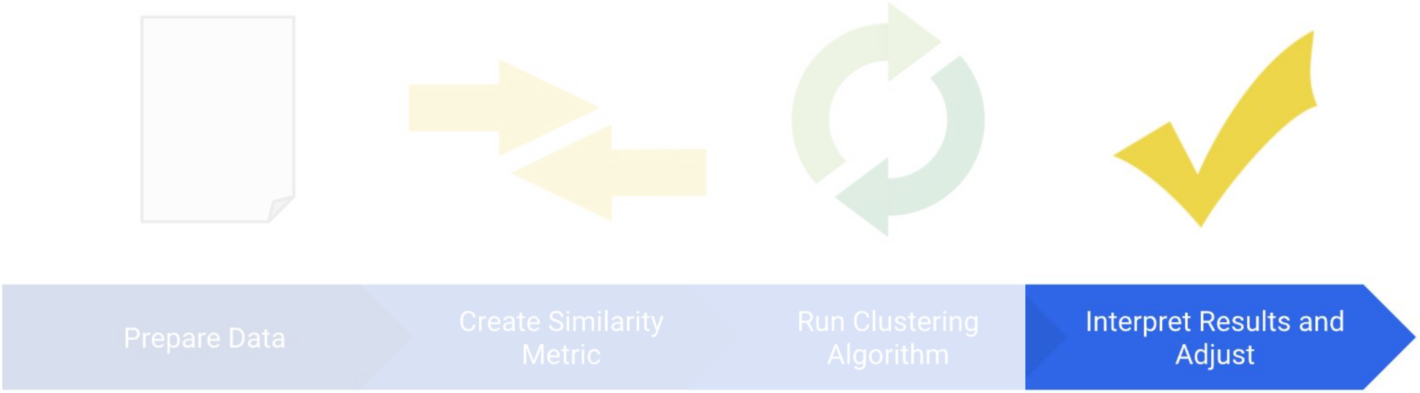


Density-based Clustering

- Density-based clustering connects areas of high example **density** into clusters
- Advantage:
    - they do not assign outliers to clusters.
- Disadvantage:
    - have difficulty with data of varying densities and high dimensions.



To cluster your data, you'll follow these steps:



Because clustering is unsupervised, **no “truth”** is available to verify results

- It mainly depends on the **subjective interpretability**
- We have some kind of **quality measures** for some algorithms (e.g. silhouette for k-Means)