Utrecht University

Lecture : Clustering

Utrecht Summer school on
Machine Learning with Python

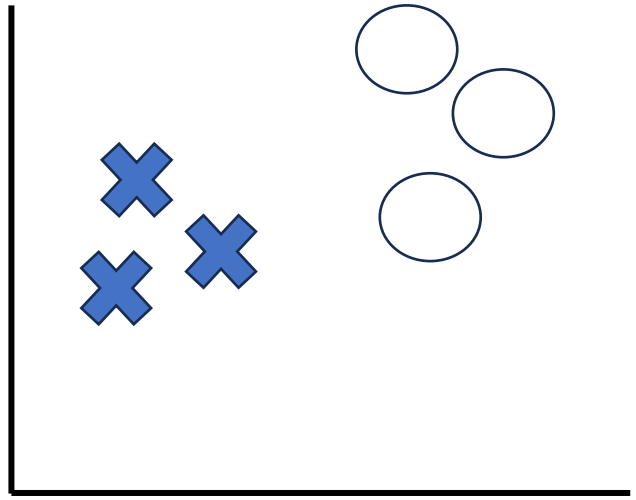
July 2024

Types of learning

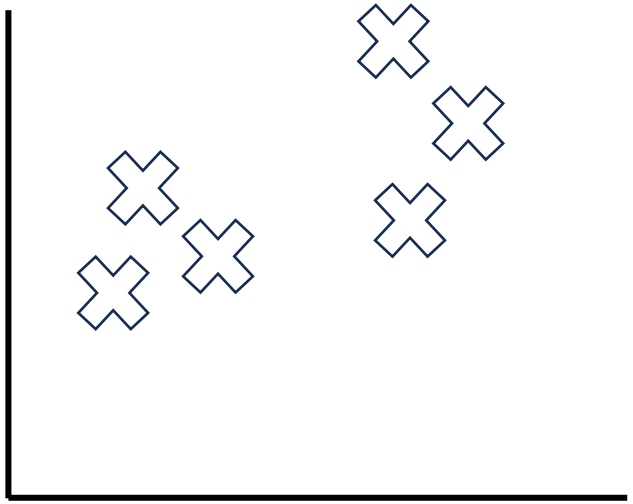
Why unsupervised learning?

- Clustering: expect groups in our data, but were not able to measure them
  - potential new subtypes of cancer tissue
  - groups of shoppers based on their purchase histories
- We want to summarize features into one feature to use in further decisions/analysis
  - subgrouping customers by their spending types
- Informative way to visualize high-dimensional data
  - Reduce dimensionaltiy of the data to visualise them in a 2-D plot

Supervised learning VS Unsupervised

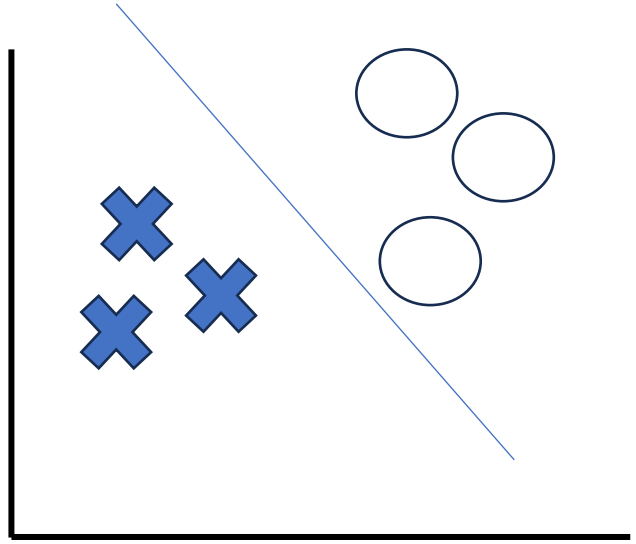


Training set: (x1, y1), (x2,y2)…, (xm,ym)

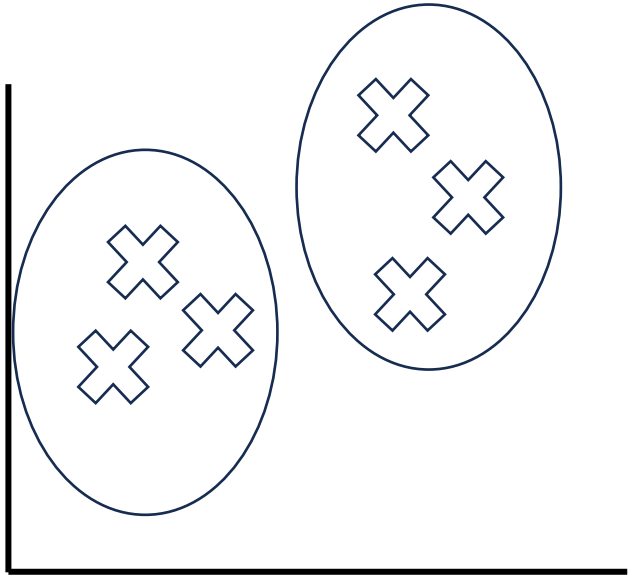


Training set: (x1, x2…, xm)

Supervised learning VS Clustering



Training set: (x1, y1), (x2,y2)…, (xm,ym)



Training set: (x1, x2…, xm)
No target/output

Clustering

Find subgroups (clusters) if there are similar examples in the dataset

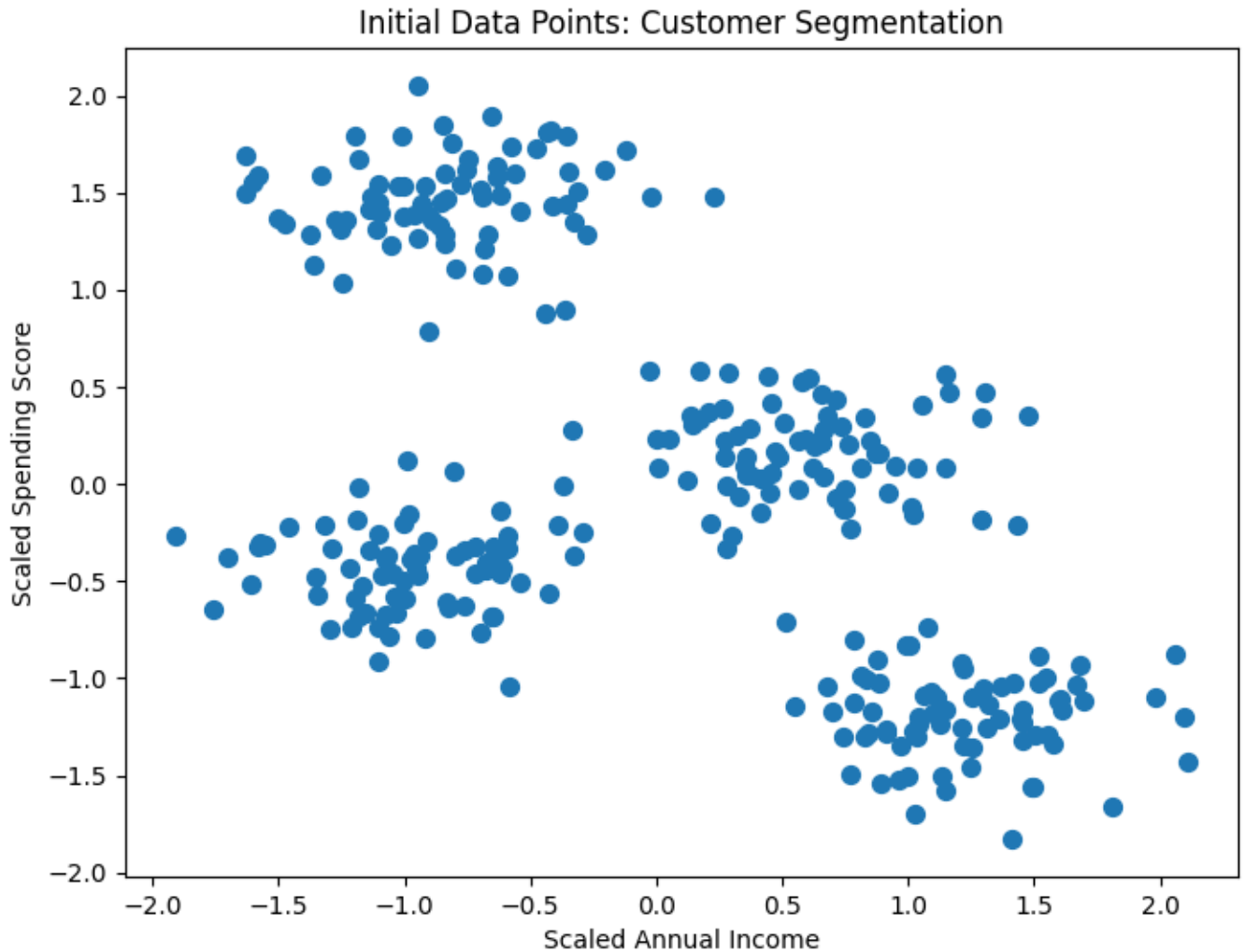
Applications of clustering

- Recommender systems
  - Cluster users with similar viewing habits
- Medical imaging
  - Cluster images to find patients with similar medical imaging (MRI, X-ray etc)
- Market segmentation
  - Divide consumers based on their spending hapits

k-Means algorithm

K-means

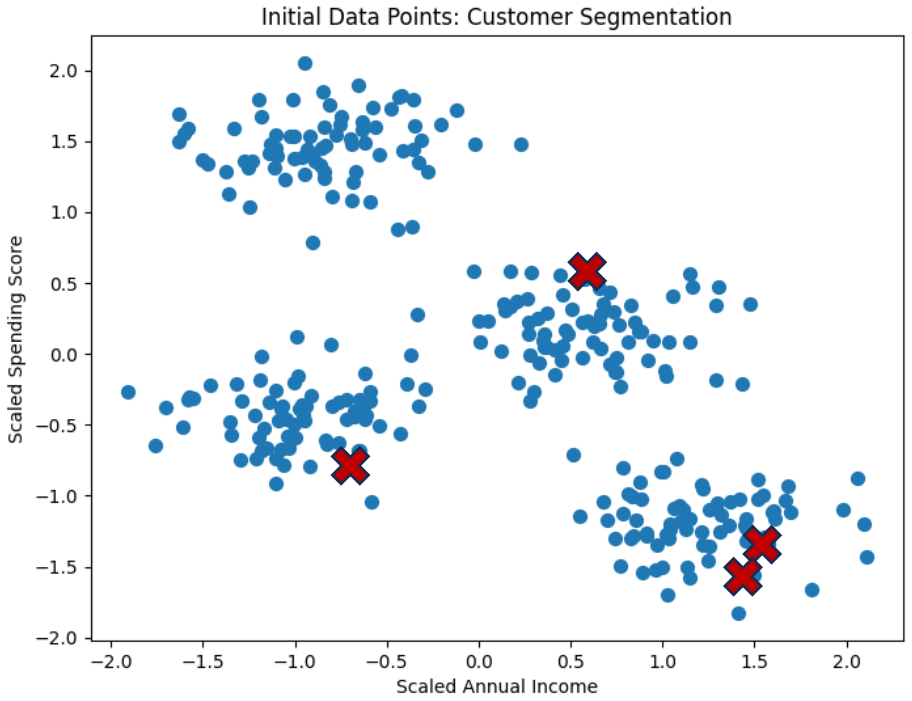
- One of the most popular clustering algorithms
- K-means is an iterative algorithm
- Each data point belongs to the cluster with the nearest mean


Initial Data Points: Customer Segmentation
Scaled Spending Score
Scaled Annual Income
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

K-means

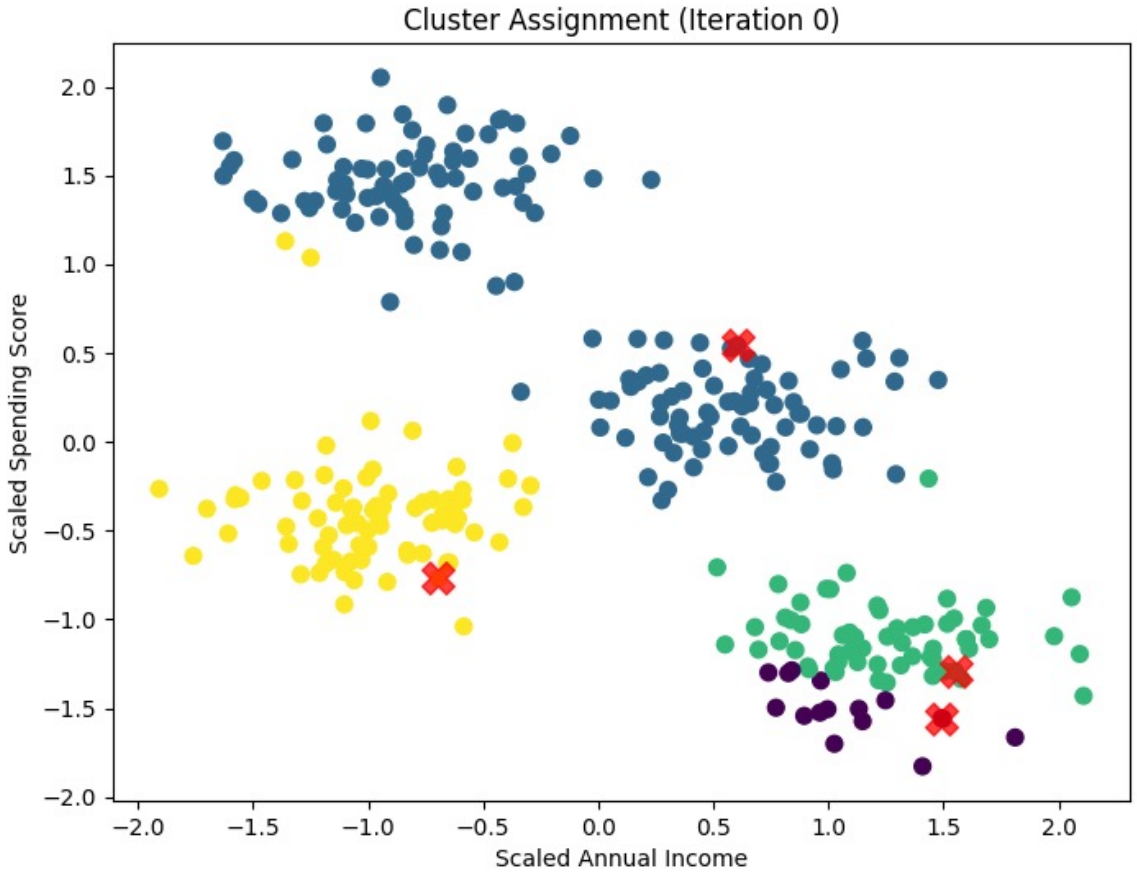
- **Step 1: Initialization:** Choose the number of clusters K and initialize the centroids randomly

- K = 4



K-means - Step 2

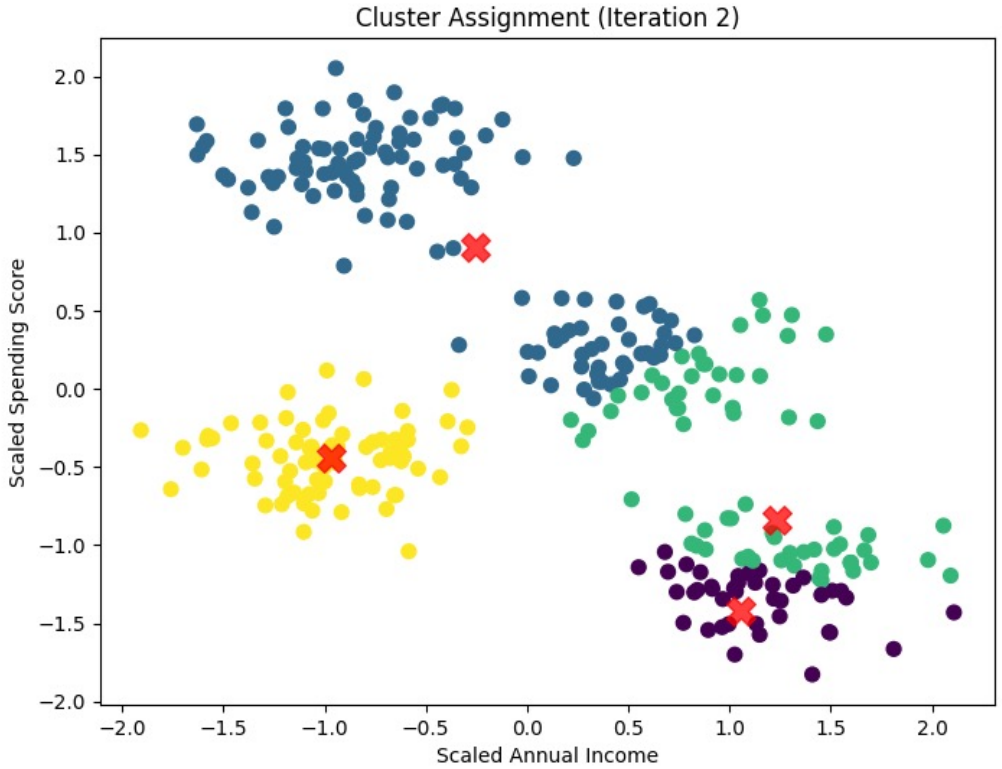
- **Step 2:** Assign each data point to the nearest centroid based on a distance metric (e.g., Euclidean distance).



K = 4

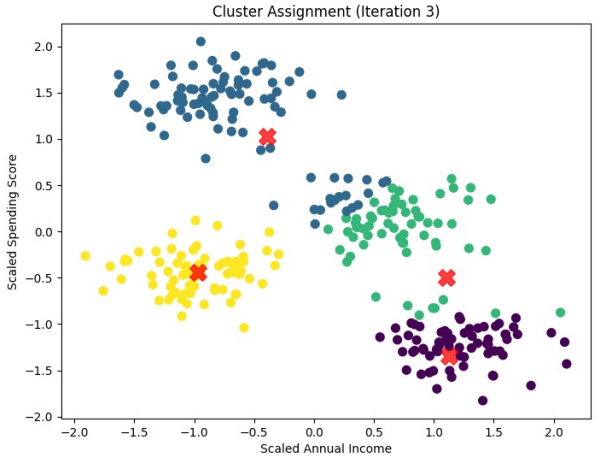
K-means - Step 3

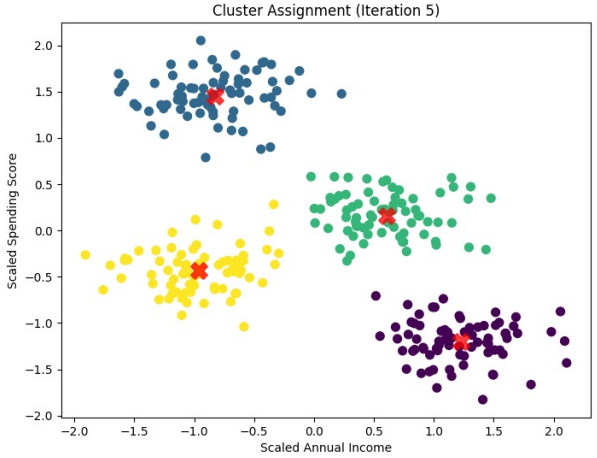
- **Step 3:** Calculate the new centroids as the mean of all points in the cluster.

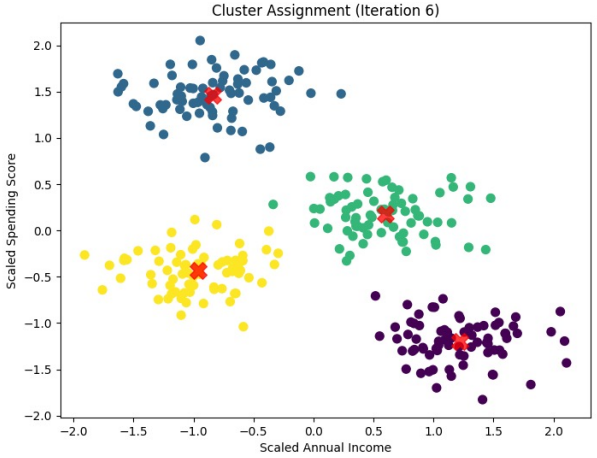


K = 4

K-means - Step 4







- **Step 4:** Repeat the assignment and update clusters until the centroids do not change significantly or a maximum number of iterations is reached

K - means

- Input

- K (number of clusters)
- Observations (x1, x2, x3, …, xm)

- Labels are not required

K-means algorithm

- Randomly initialize K cluster centroids $\mu_1, \mu_2, \cdots, \mu_K \in \mathbb{R}^n$

Repeat{

for i = 1 to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$

Cluster assignment step

for k = 1 to K

μ_k := average (mean) of points assigned to cluster k

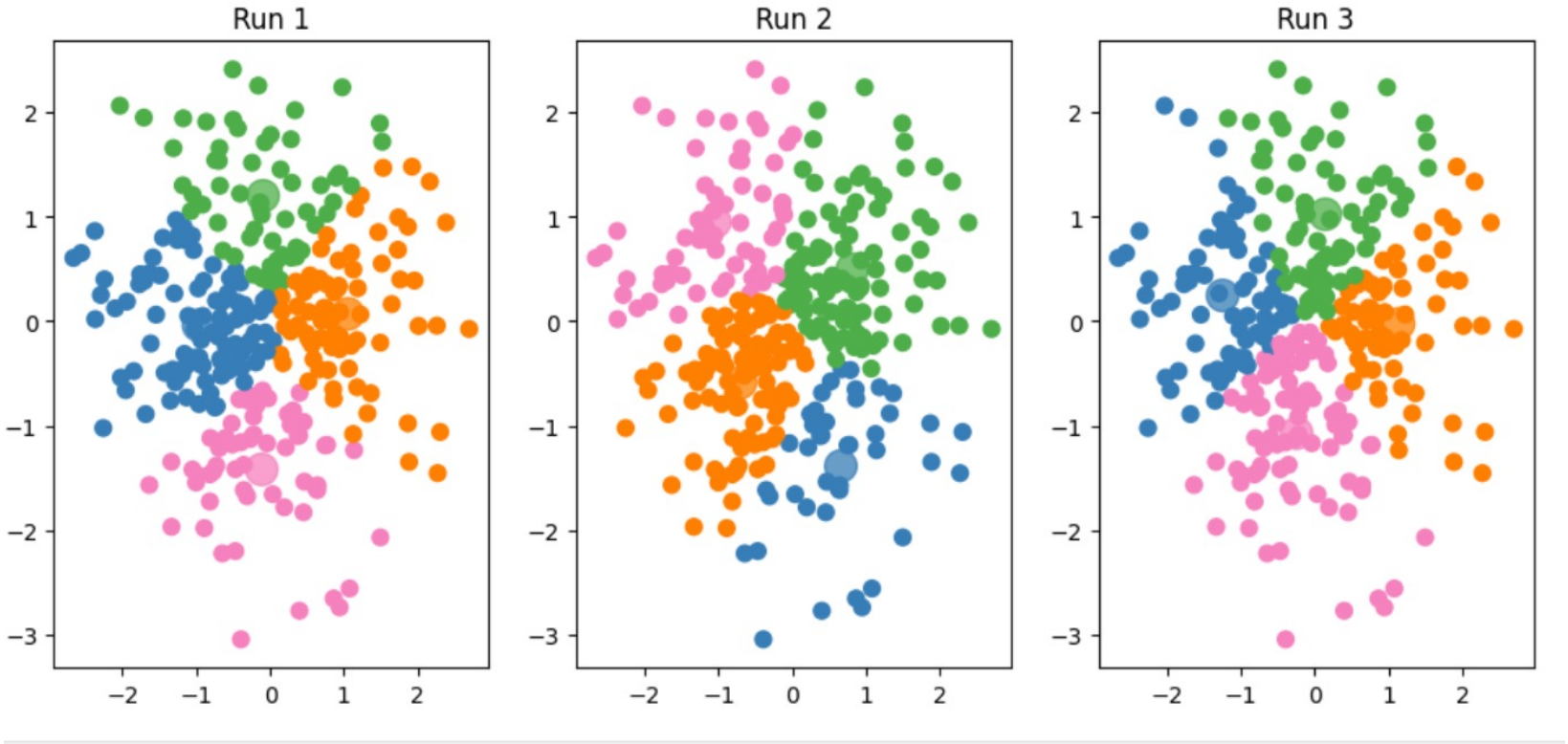
Centroid update step

}

Slide credit: Andrew Ng

Random initialization

The initialization of the centroids is random, and this can give different results on the same dataset.



Random initialization

- The random initialization can lead us to local optimal solution
- To avoid this inconsistency, we can run it 100 times and select the best performance among these 100

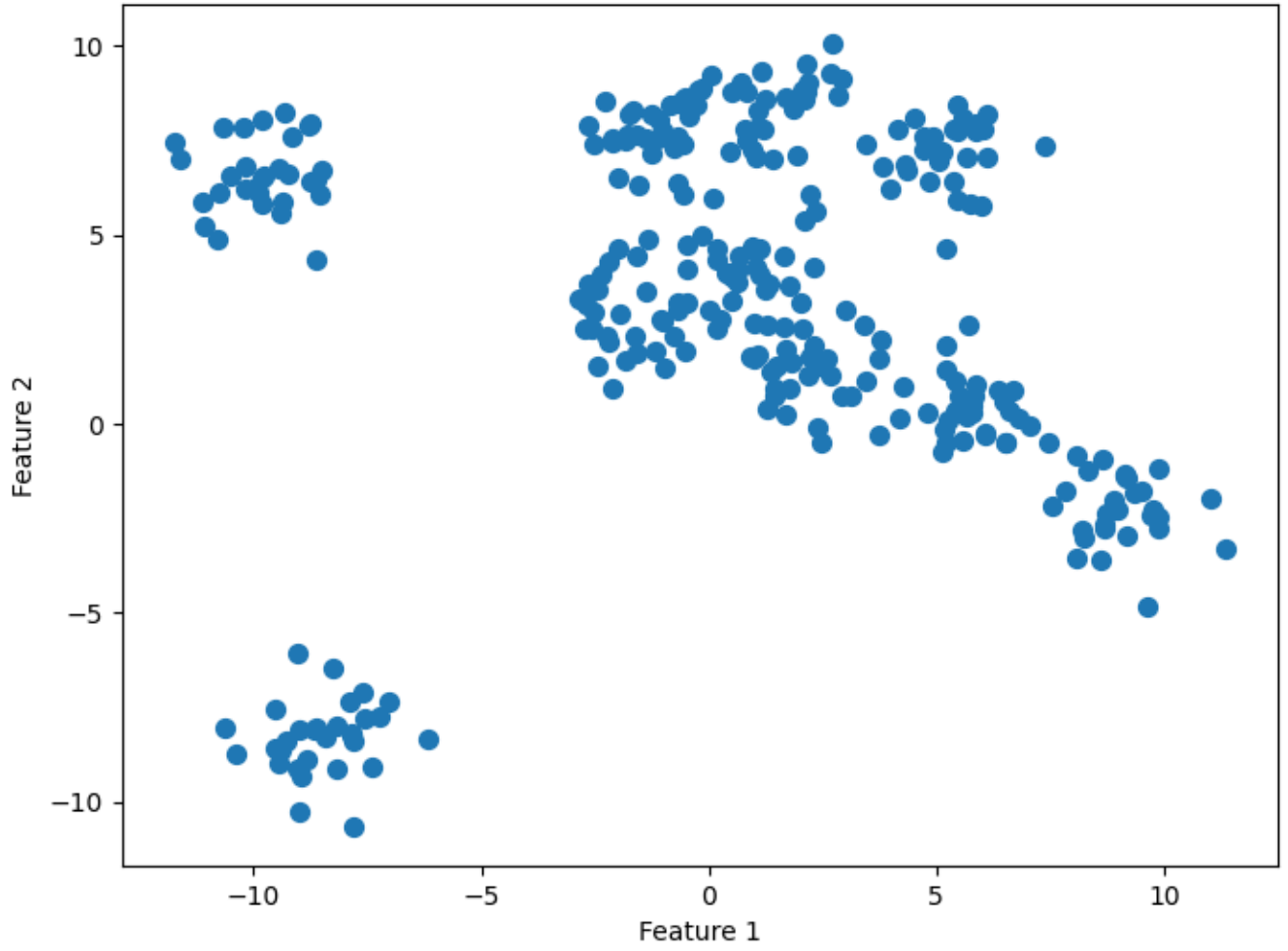
- Performance?

Random initialization

- Which of those is the best - > measure **inertia** (or also called distortion cost function)
- The mean square distance between each point and the centroid of its cluster
- The lower the value of distortion the better the solution

Choose the number of clusters

- Choose them manually with visualization
- But it is not always easy
- No clear answer

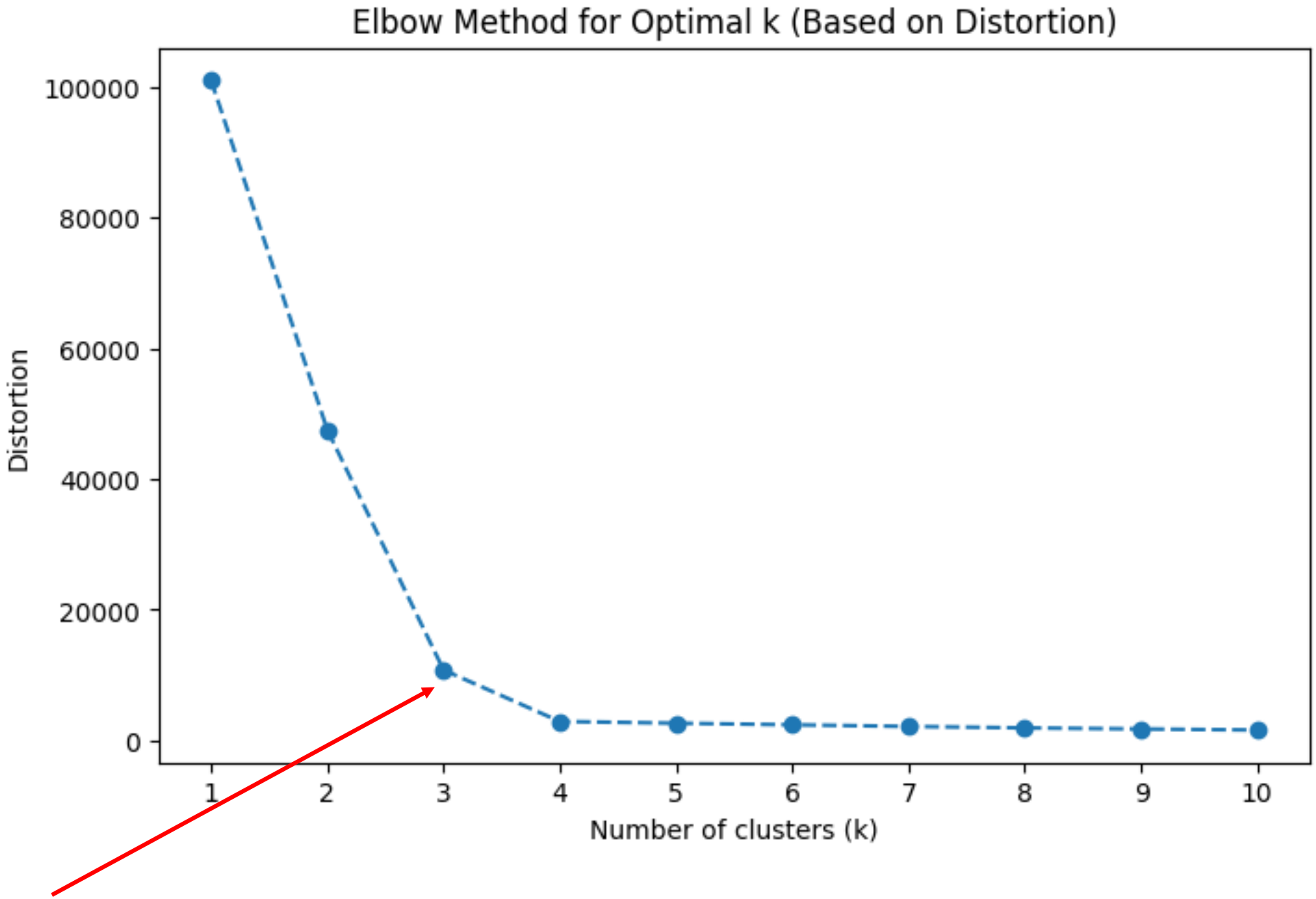


Choose k with elbow method

- Run K-means for different values of k
- For each k, calculate the distortion cost function
- Create a plot with the number of clusters k on the x-axis and the distortion on the y-axis.
- Distortion typically decreases as k increases
  - Adding more clusters reduces the distance from each point to its assigned centroid

Elbow method

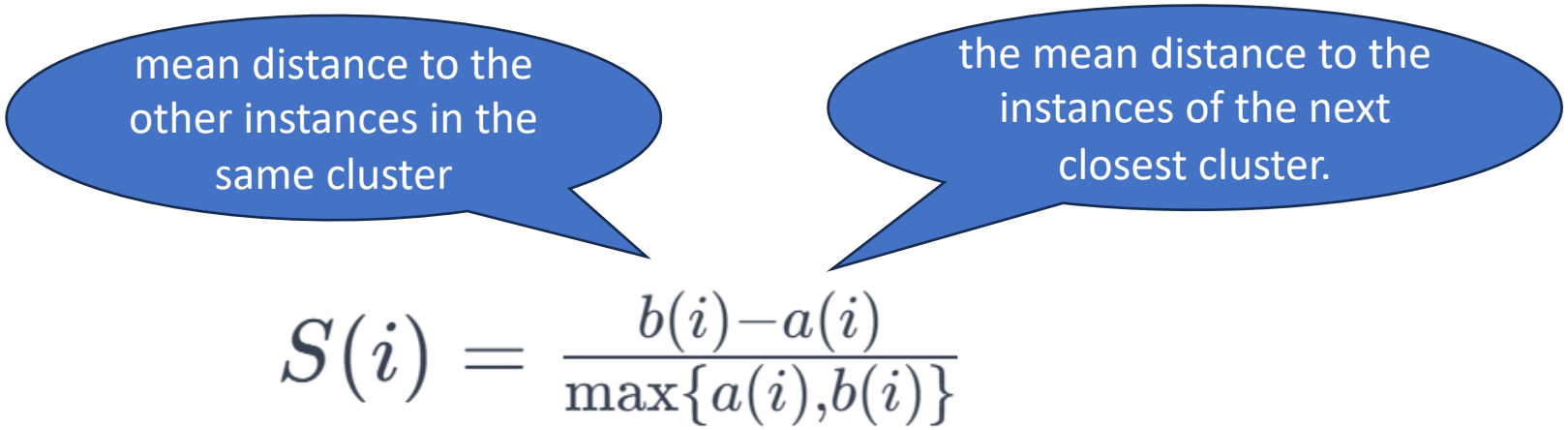
- Distortion drops quickly when we increase the number of clusters until a certain point, and then it slows down and decreases more and more slowly.
- Identify the "Elbow": where the rate of decrease sharply slows down.
- The optimal k is typically at the elbow; adding more clusters beyond this point provides does not reduce distortion significantly


Elbow Method for Optimal k (Based on Distortion)
Distortion
100000
80000
60000
40000
20000
0
1
2
3
4
5
6
7
8
9
10
Number of clusters (k)

Silhouette score

- Graphs are not always clear. Sometimes the curve decreases very smoothly.
- In this case, you should try the **Silhouette score**.
- The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation).
- The silhouette ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters.

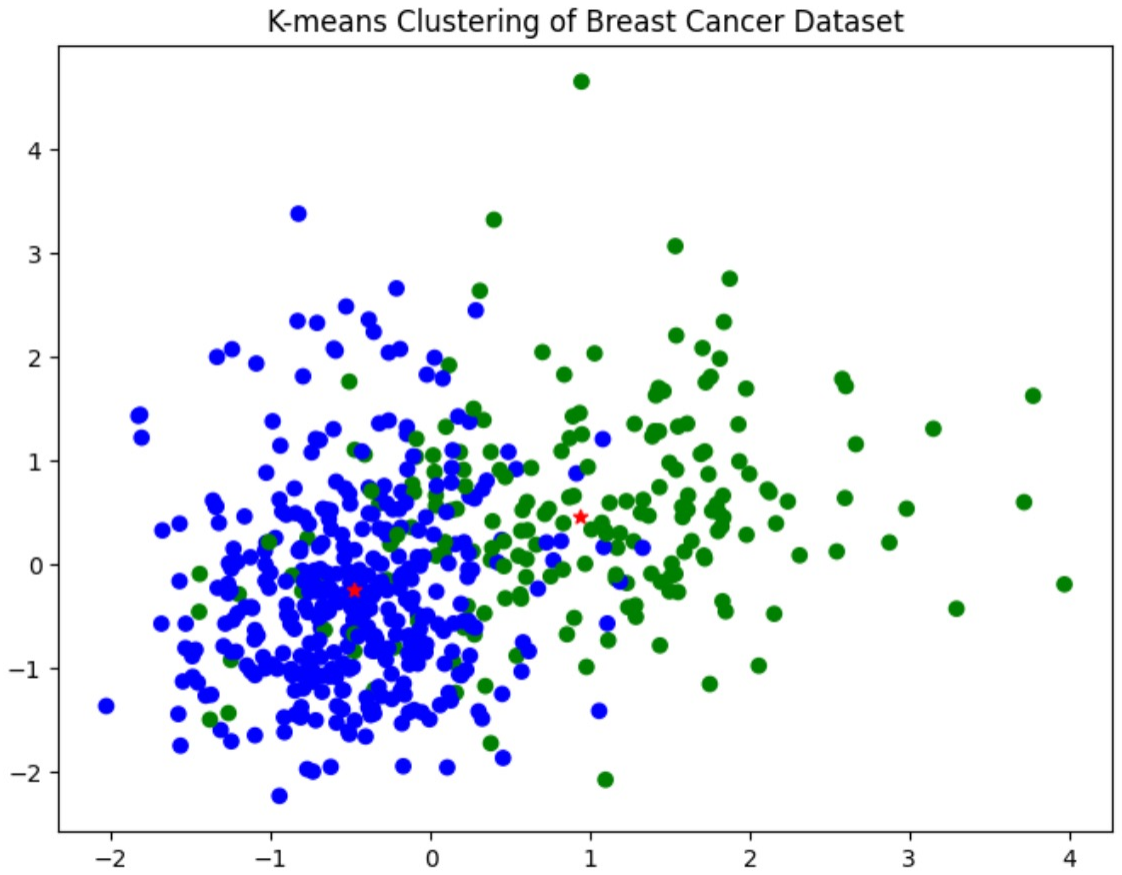
Silhouette score



- To calculate the silhouette score for the whole dataset, you take the mean of silhouette scores over all the instances.
- The higher the score, the better, and it does not constantly decrease as inertia.
- Number of clusters-> the one with the highest silhouette score.

K-means in Python

```
kmeans = KMeans(n_clusters=2, n_init= 10)
kmeans.fit(df_scaled)
print("Centroids of the clusters:", kmeans.cluster_centers_)
```



Exercise

- Number of observations = 6
- Number of clusters = 2
- Distance metric: Euclidean distance

Step-1: Choose random K points and set as cluster centers
C1 = (2,2)
C2 = (3, 3)

Question: Which are going to be the centroids after one iteration?
C1 = (?, ?)
C2 = (?, ?)

No	X	Y
1	1	1
2	2	3
3	1	2
4	3	3
5	2	2
6	3	1

Exercise

- Number of observations = 6
- Number of clusters = 2
- Distance metric: Euclidean distance

Step-1: Choose random K points and set as cluster centers
C1 = (2,2)
C2 = (3, 3)

Question: Which are going to be the centroids after one iteration?
C1 = (1.75, 1.5)
C2 = (2.5, 3)

No	X	Y
1	1	1
2	2	3
3	1	2
4	3	3
5	2	2
6	3	1

Evaluation of clusters

How to evaluate clustering results

1. Use of external information
2. Visual exploration
3. Stability assessment / sensitivity analysis

External validation

1. External validation

- Are the clusters associated with external feature Y?
- “Making unsupervised supervised”

- Examples:
- Are my customer segments based on spending associated with the demographics of the customers?

Visual exploration

- Problem: Kind of hard to see already…
- Wait till you get 1000 variables!
- New idea: Reduce variables into 2D “manifold” for visualization
- Popular techniques: PCA, t-SNE, Discriminant Coordinates

Cluster stability

- Three “stabilities”
- How much does clustering change when:

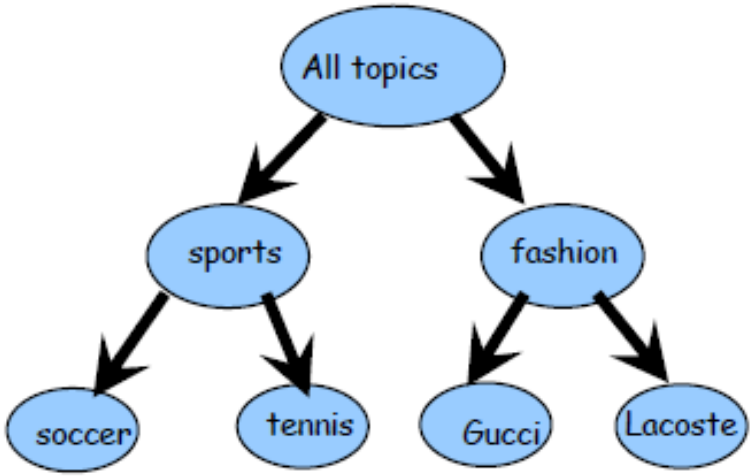
1. Changing some hyperparameters (distance metric, K, ...)
2. Changing some observations (bootstrapping, Hennig, 2007)
3. Changing some features

Check if observations are classified into same cluster across choices

10 minutes break

Hierarchical Clustering

Hierarchical clustering



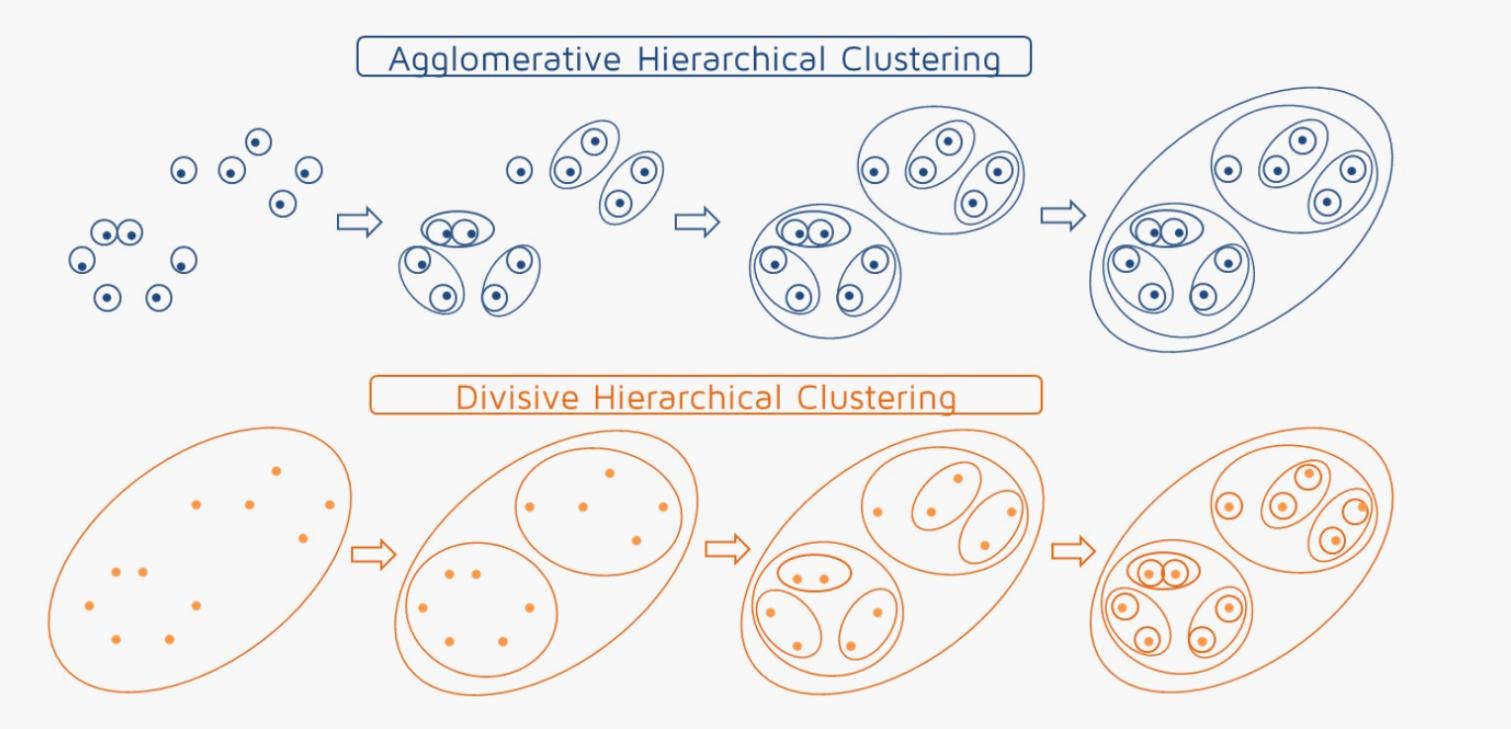
- Builds a hierarchy of clusters
- A hierarchy might be more natural to the type of data
- Different users might care about different levels of granularity

Hierarchical clustering

- **Top-down (divisive)**
  - Partition data into 2-groups (e.g., 2-means)
  - Recursively cluster each group

- **Bottom-up (agglomerative)**
  - Start with every point in its own cluster.
  - Repeatedly merge the “closest” two clusters
  - Different definitions of “closest” give different algorithms.

Hierarchical clustering



Bottom-up (agglomerative)

- Compute the distance matrix between the input data points
- Let each cluster be a point
- Repeat:
  - Merge the two closest clusters
  - Update the distance matrix
- Until only one cluster remains

Bottom-up (agglomerative)

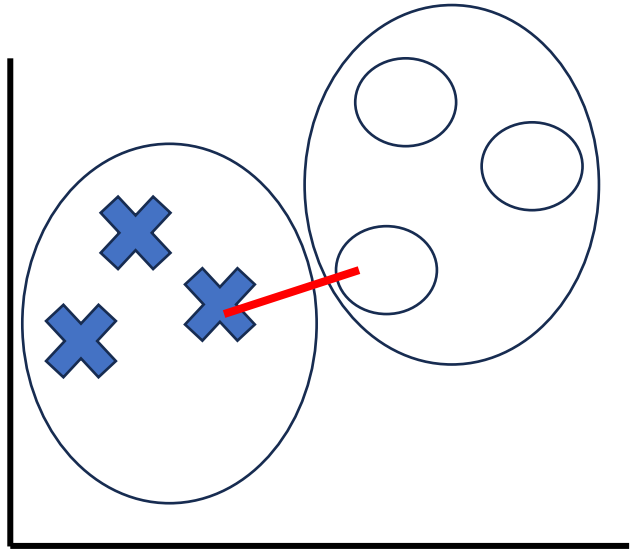
- Compute the distance matrix between the input data points
- Let each cluster be a point
- Repeat:
  - Merge the two closest clusters
  - Update the distance matrix
- Until only one cluster remains

- Distance matrix: Manhattan, Euclidean
- But we need a distance measure for the clusters (for the merging step).

Single linkage

- Have a distance measure on pairs of objects.
- $d(x, y)$: Distance between x and y

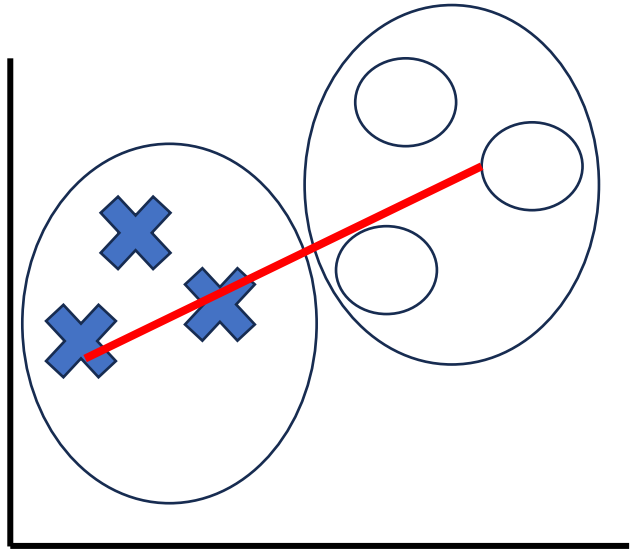
- Single linkage: $\mathrm{dist}(\mathrm{A}, \mathrm{B}) = \min_{\mathrm{x} \in A, x' \in B} \mathrm{d}(\mathrm{x}, \mathrm{x}')$



Bottom-up (agglomerative)

- Have a distance measure on pairs of objects.
- $d(x, y)$: Distance between x and y

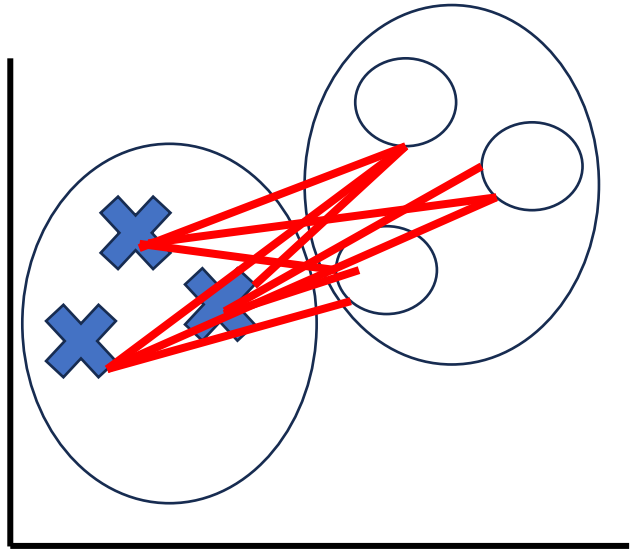
- Complete linkage: $\mathrm{dist}(\mathrm{A}, \mathrm{B}) = \max_{\mathrm{x} \in A, x' \in B} \mathrm{d}(\mathrm{x}, \mathrm{x}')$



Bottom-up (agglomerative)

- Have a distance measure on pairs of objects.
- $d(x, y)$: Distance between x and y

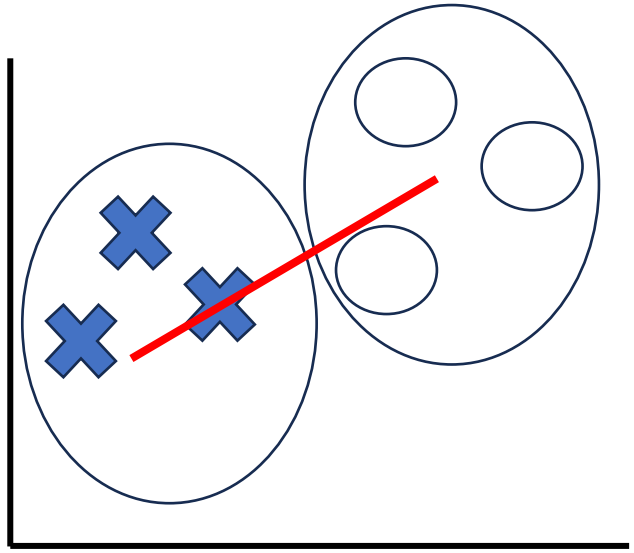
- Average linkage: $\mathrm{dist}(\mathrm{A}, \mathrm{B}) = \underset{\mathrm{x} \in A, x' \in B}{\text{average}}\, \mathrm{d}(\mathrm{x}, \mathrm{x}')$

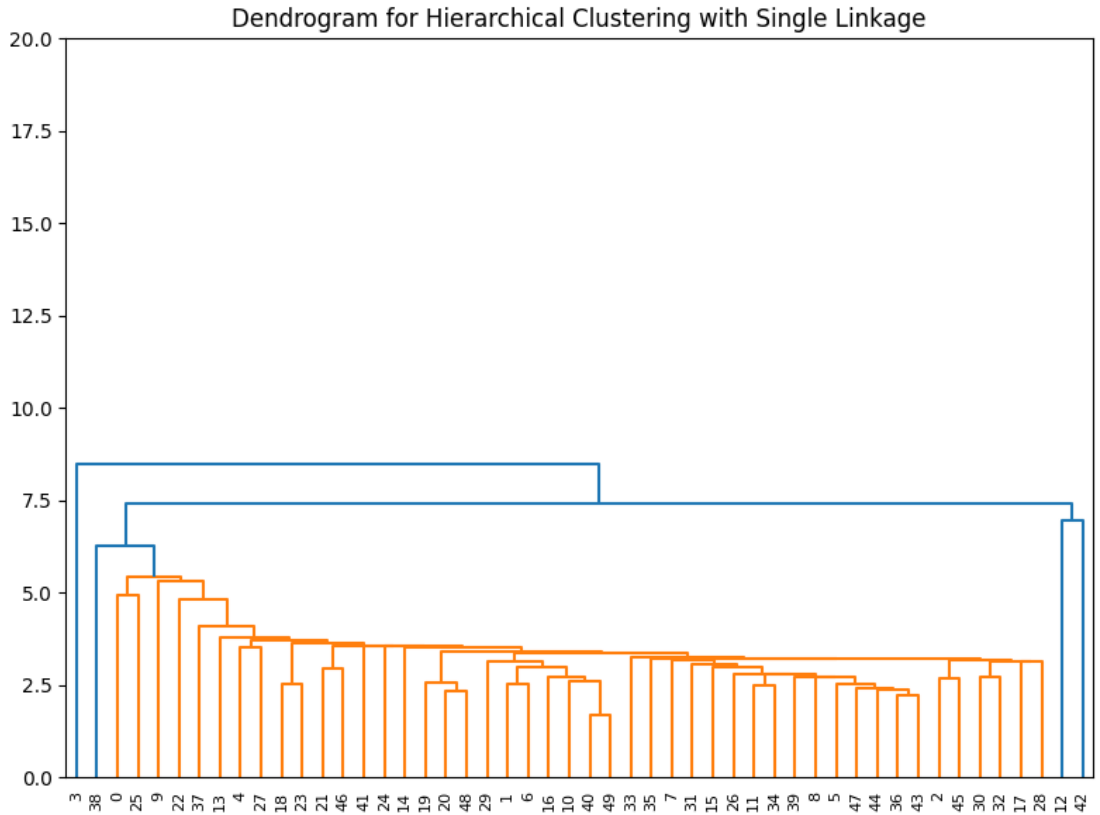


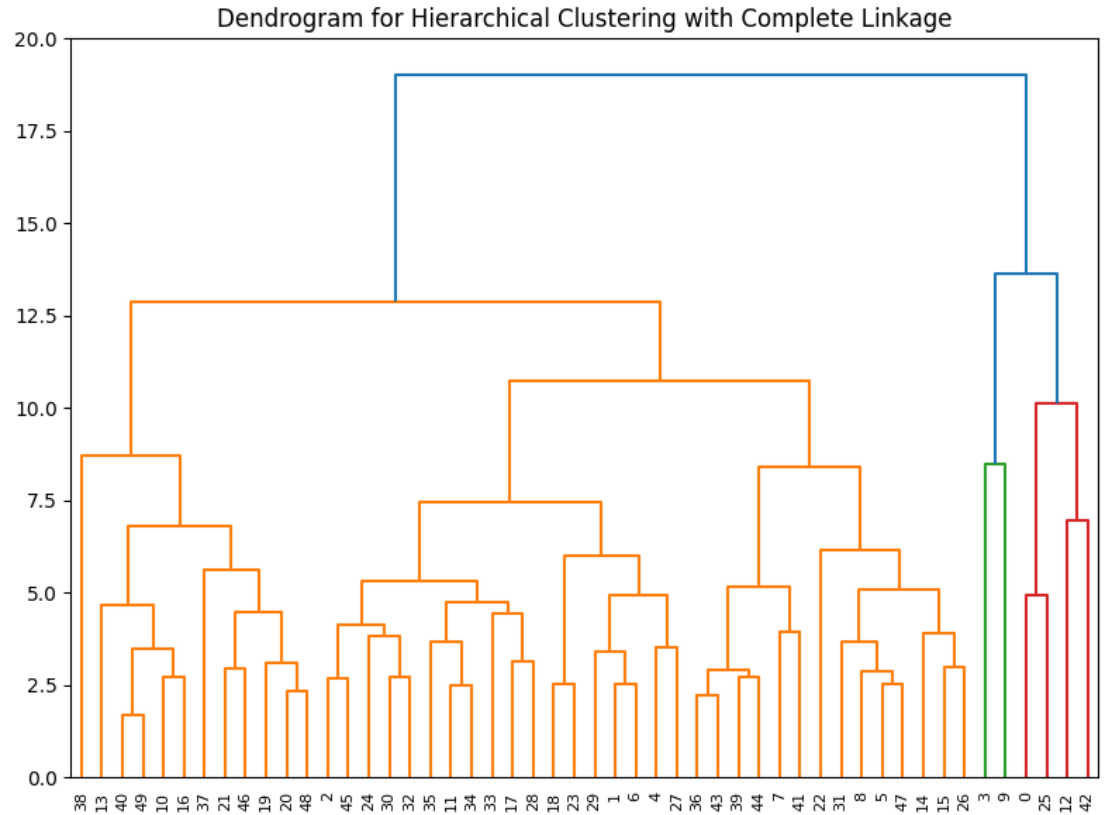
Bottom-up (agglomerative)

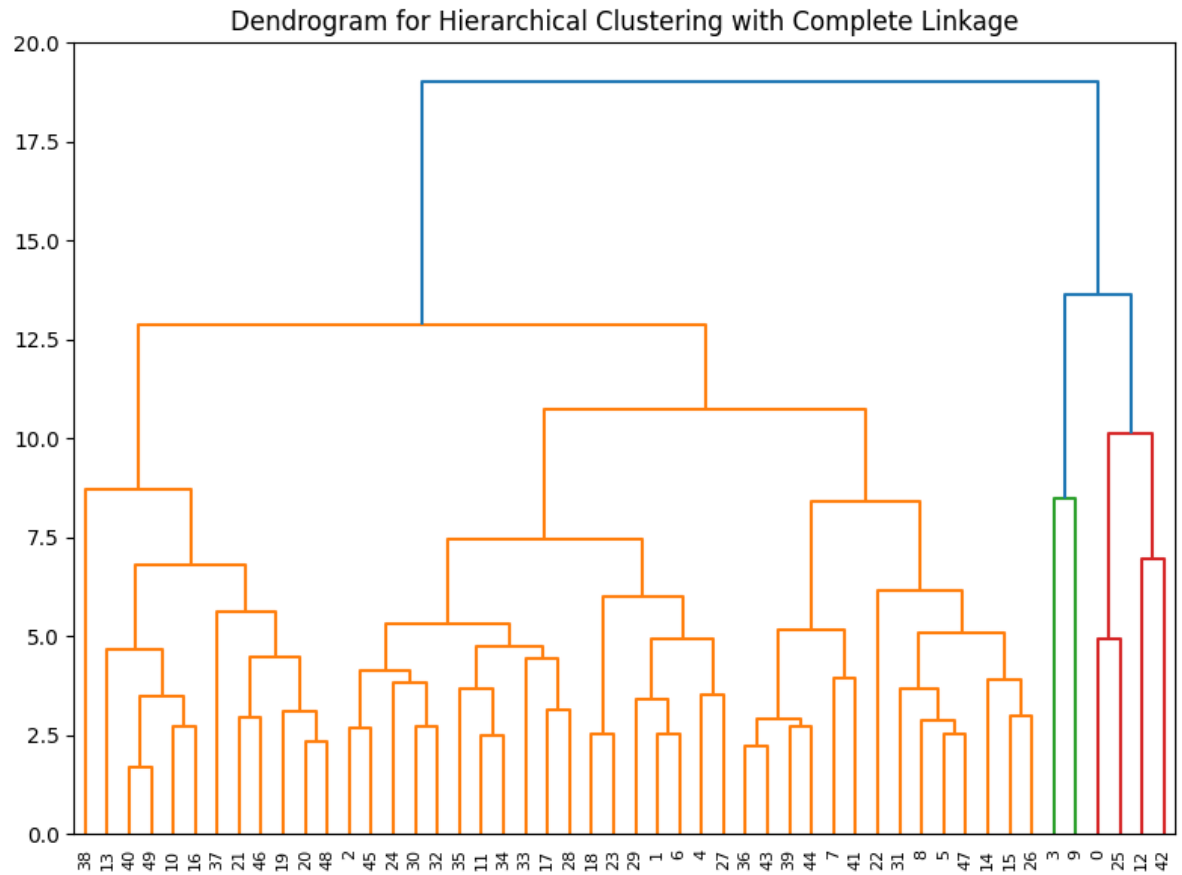
- Have a distance measure on pairs of objects.
- $d(x, y)$: Distance between x and y

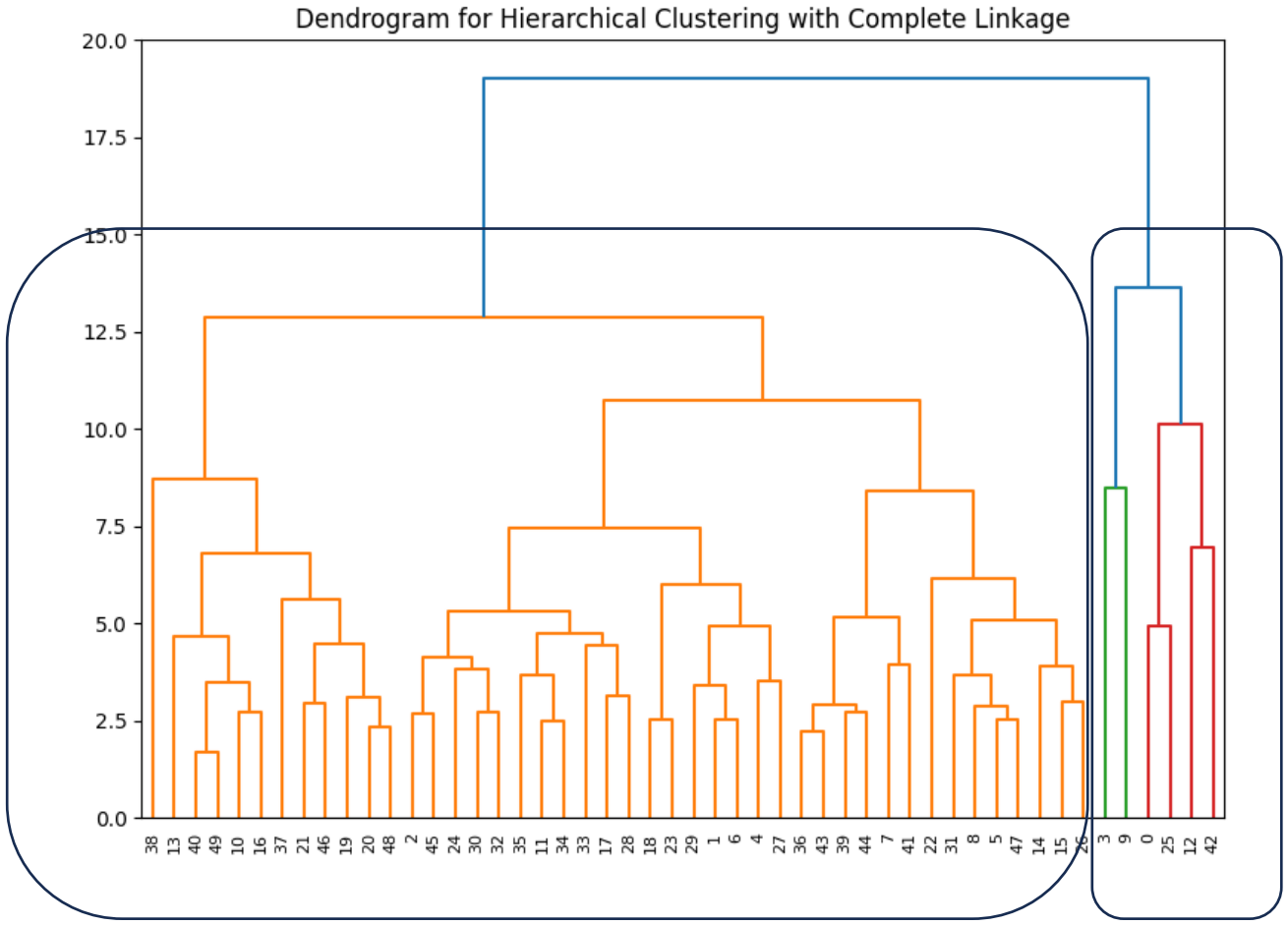
- Centroid linkage: $\mathrm{dist}(\mathrm{A}, \mathrm{B}) = \mathrm{d}(\mathrm{mean}(\mathrm{A}), \mathrm{mean}(\mathrm{B}))$




Dendrogram for Hierarchical Clustering with Single Linkage
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
3 38 0 25 9 22 37 13 4 27 18 23 21 46 41 24 14 19 20 48 29 1 6 16 10 40 49 33 35 7 31 15 26 11 34 39 8 5 47 44 36 43 2 45 30 32 17 28 12 42


Dendrogram for Hierarchical Clustering with Complete Linkage
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
38 13 40 49 10 16 37 21 46 19 20 48 2 45 24 30 32 35 11 34 33 17 28 18 23 29 1 6 4 27 36 43 39 44 7 41 22 31 8 5 47 14 15 26 3 9 0 25 12 42


Dendrogram for Hierarchical Clustering with Complete Linkage
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
38 13 40 49 10 16 37 21 46 19 20 48 2 45 24 30 32 35 11 34 33 17 28 18 23 29 1 6 4 27 36 43 39 44 7 41 22 31 8 5 47 14 15 26 3 9 0 25 12 42


Dendrogram for Hierarchical Clustering with Complete Linkage
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
38 13 40 49 10 16 37 21 46 19 20 48 2 45 24 30 32 35 11 34 33 17 28 18 23 29 1 6 4 27 36 43 39 44 7 41 22 31 8 5 47 14 15 26 3 9 0 25 12 42

k-means VS Hierarchical clustering

- k-means clustering is faster and simpler, but requires choosing the number of clusters beforehand and may not capture complex structures
- Hierarchical clustering is more flexible and intuitive, but can be computationally expensive and sensitive to outliers

Dimensionality Reduction

Plot high dimensional data

- Suppose you have data with many dimensions
- How are you going to plot those data?

```
Original Data:
   mean radius  mean texture  mean perimeter  mean area  mean smoothness  \
0        17.99         10.38           122.8     1001.0          0.11840
1        20.57         17.77           132.9     1326.0          0.08474
2        19.69         21.25           130.0     1203.0          0.10960

   mean compactness  mean concavity  mean concave points  mean symmetry  \
0           0.27760          0.3001              0.14710         0.2419
1           0.07864          0.0869              0.07017         0.1812
2           0.15990          0.1974              0.12790         0.2069

   mean fractal dimension  ...  worst radius  worst texture  worst perimeter  \
0                 0.07871  ...         25.38          17.33            184.6
1                 0.05667  ...         24.99          23.41            158.8
2                 0.05999  ...         23.57          25.53            152.5

   worst area  worst smoothness  worst compactness  worst concavity  \
0      2019.0            0.1622             0.6656           0.7119
1      1956.0            0.1238             0.1866           0.2416
```

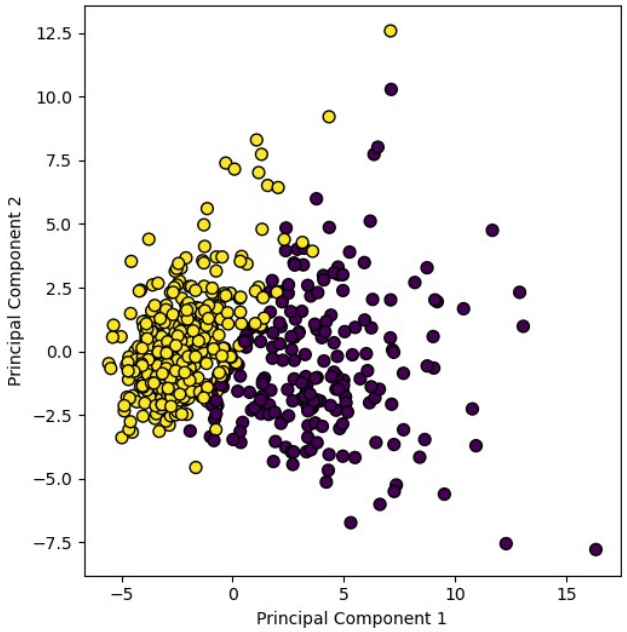
Dimensionality reduction

- Simplifies models
- Reduces computational cost
- Helps in visualizing high-dimensional data

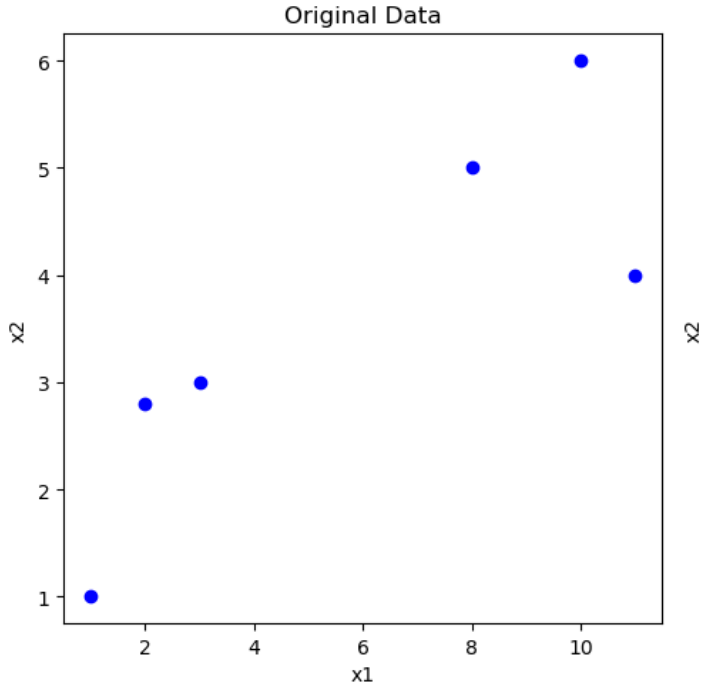
Principal Component Analysis

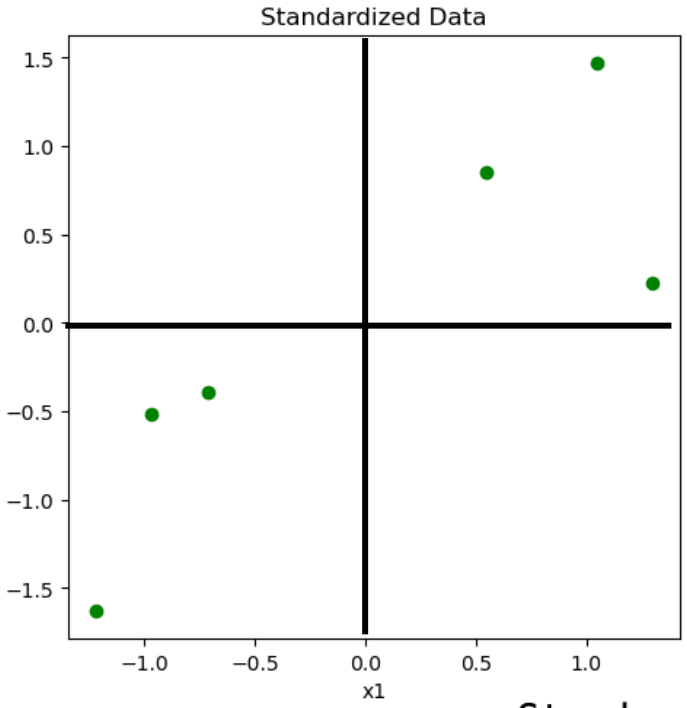
- Identifies directions (principal components) that maximize variance
- Projects data onto these new directions (PC components) to reduce dimensions while retaining most information of the data

```
Original Data:
   mean radius  mean texture  mean perimeter  mean area  mean smoothness  \
0        17.99         10.38           122.8     1001.0          0.11840
1        20.57         17.77           132.9     1326.0          0.08474
2        19.69         21.25           130.0     1203.0          0.10960

   mean compactness  mean concavity  mean concave points  mean symmetry  \
0           0.27760          0.3001              0.14710         0.2419
1           0.07864          0.0869              0.07017         0.1812
2           0.15990          0.1974              0.12790         0.2069

   mean fractal dimension  ...  worst radius  worst texture  worst perimeter  \
0                 0.07871  ...         25.38          17.33            184.6
1                 0.05667  ...         24.99          23.41            158.8
2                 0.05999  ...         23.57          25.53            152.5

   worst area  worst smoothness  worst compactness  worst concavity  \
0      2019.0            0.1622             0.6656           0.7119
1      1956.0            0.1238             0.1866           0.2416
```



Standarize each feature





Original Data:

	x1	x2
0	10	6.0
1	11	4.0
2	8	5.0
3	3	3.0
4	2	2.8
5	1	1.0

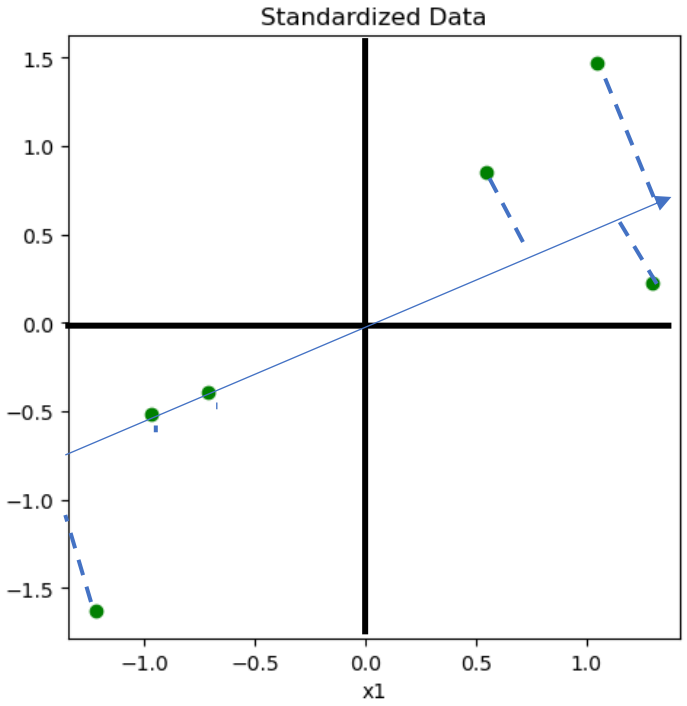
Standardized Data:

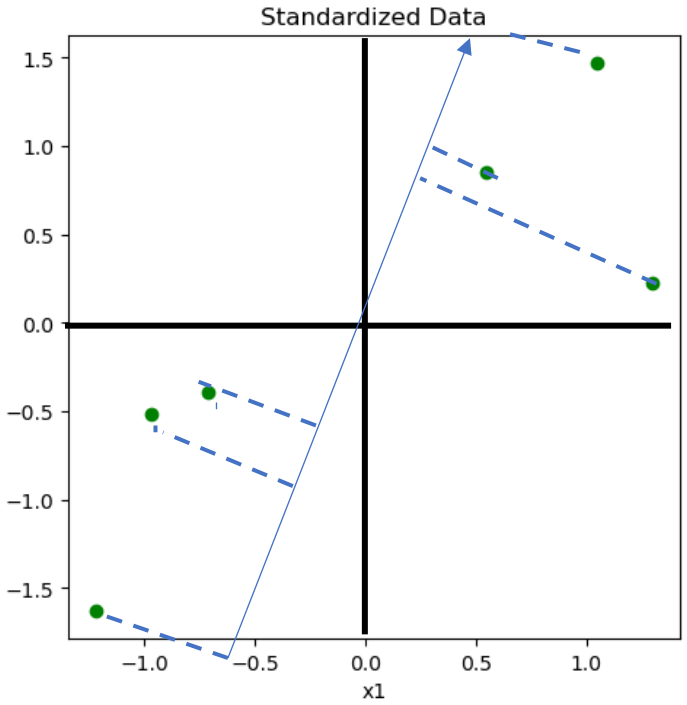
	x1	x2
0	1.048055	1.466179
1	1.299588	0.227154
2	0.544988	0.846667
3	-0.712677	-0.392358
4	-0.964210	-0.516260
5	-1.215743	-1.631382

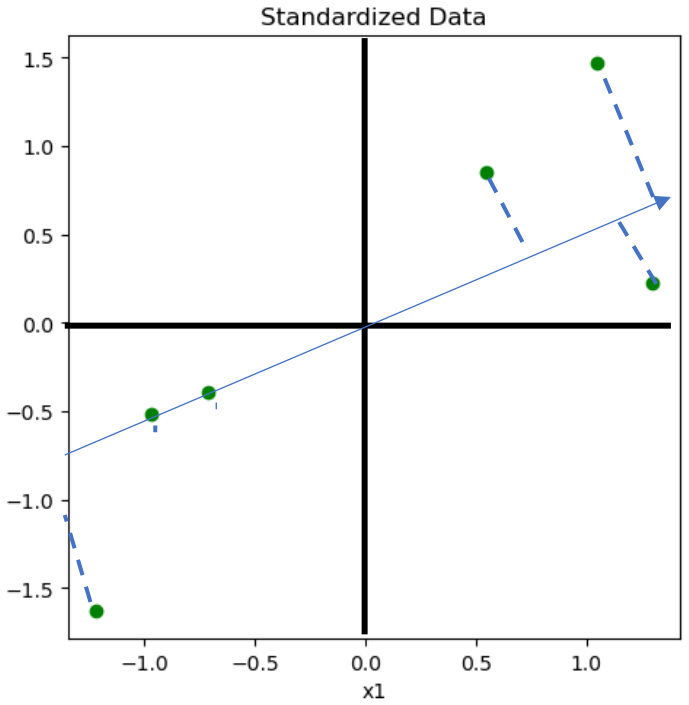
PCA

Find the line that goes through the origins and:

1. minimizes the sum of the distances from the points to the line
2. Or maximises the distances of the projected points on the line to the origin

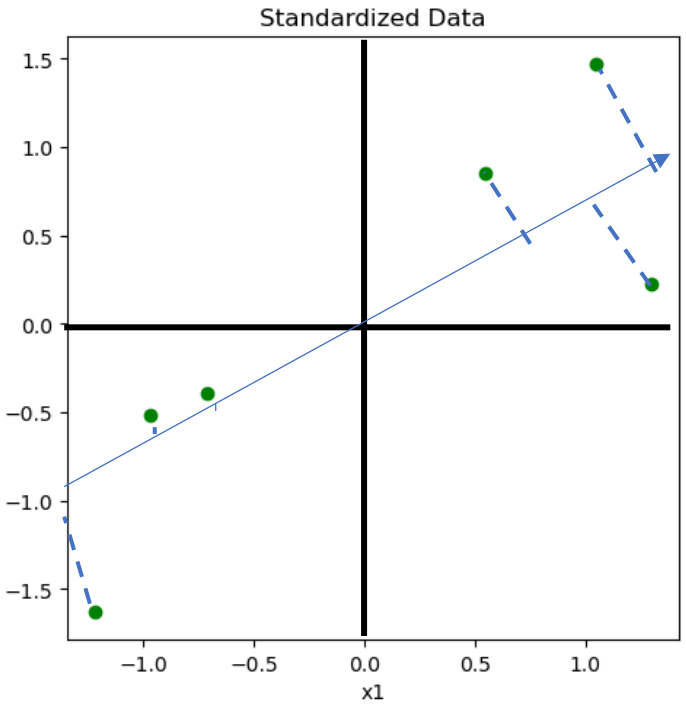



Standardized Data
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
x1

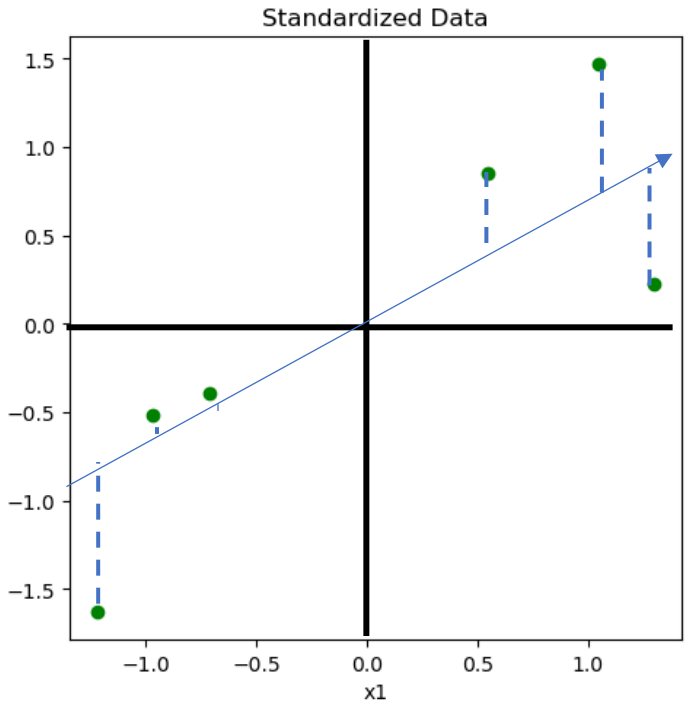

Standardized Data
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
x1

Not linear regression

PCA



Linear Regression

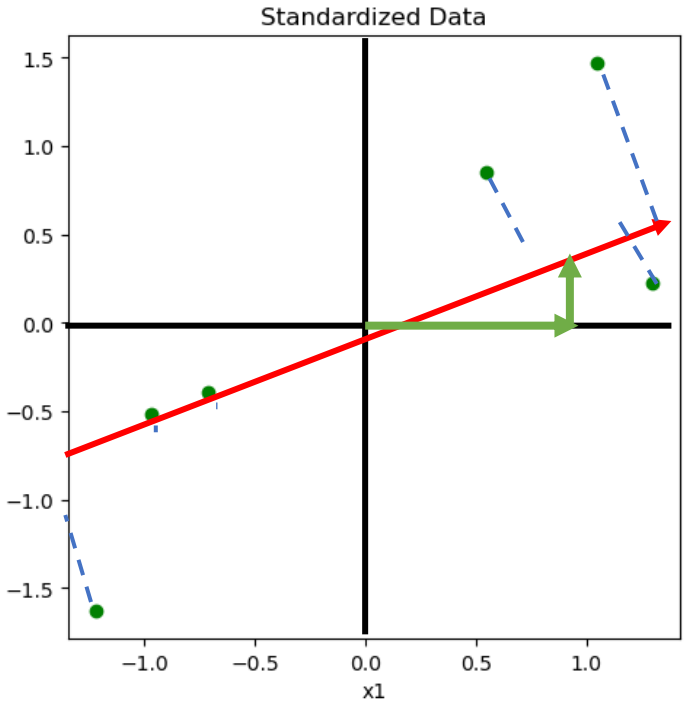


PCA

The line is called PC1

- PC1 has a slope of 0.25
- For every 1 unit of x1 axis, we go up 0,25 unit of x2 axis (green arrows)
- Data are mostly spread out on x1 axis
- PC1 is a linear combination of x1 and x2

-> to make PC1, mix 1 unit1 of x1 and 0,25 units of x2

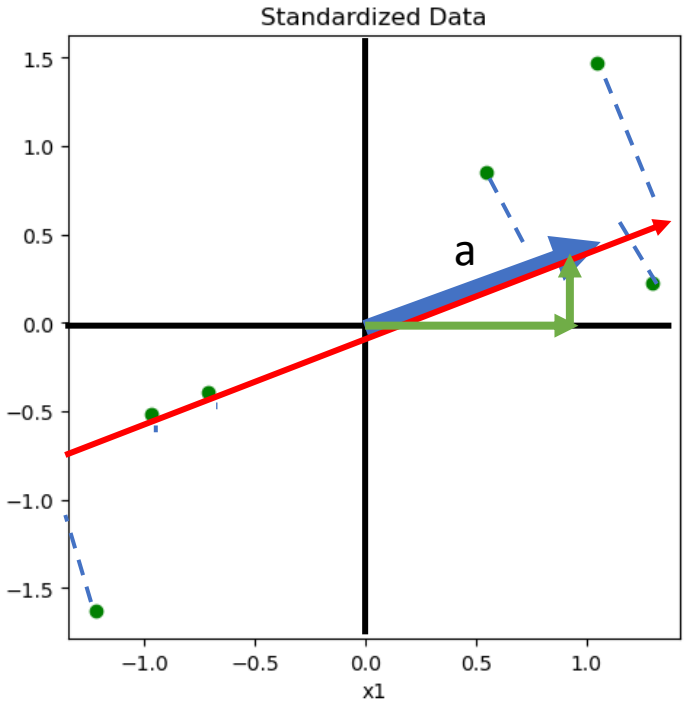


PCA

- $a^2 = b^2 + c^2$ -> a = 1,03
- PC1 is scaled so that its length is 1, so we can divide by 1,03
  - a = 1,
  - x1 = 1/ 1,03 = 0,97,
  - x2 = 0,25/ 1,03 = 0,3

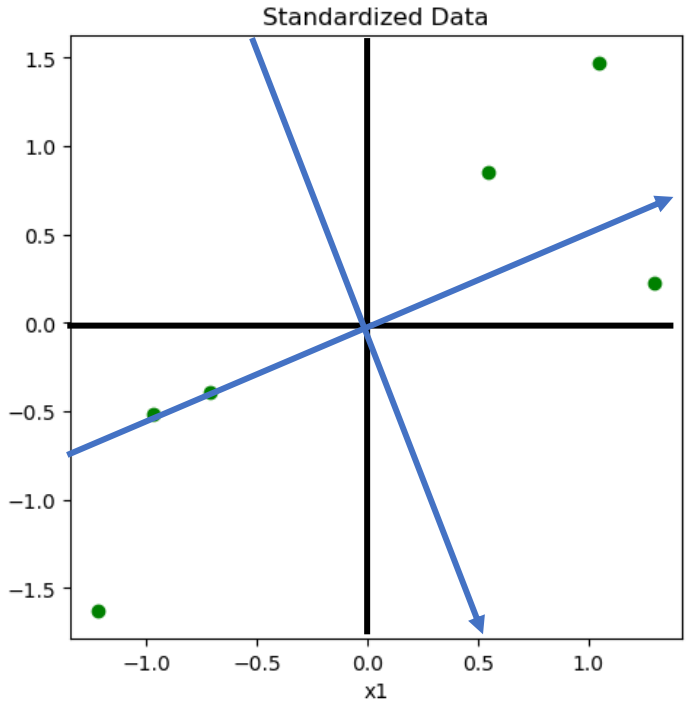
The one unit vector that consists of 0,97 of x1 and 0,3 of x2 is called **singular vector** or **eigenvector** for PC1

The proportions are called **loading scores**



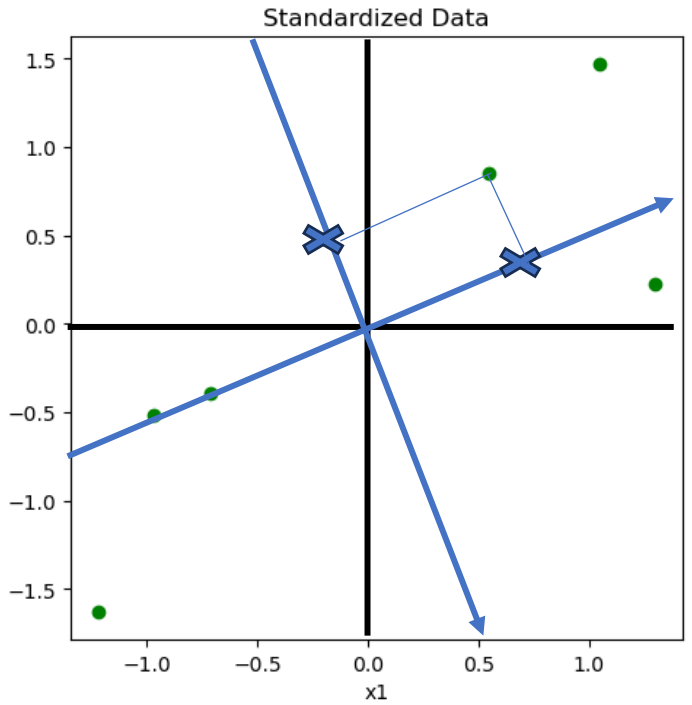
PC2

- PC2 is the line that is perpendicular to PC1 and goes through the origin
- Loading scores for PC2:
  - 0.97 for x2
  - - 0.3 for x1
- In terms of how the values are projected in PC2, x2 is 4 times as important as x1



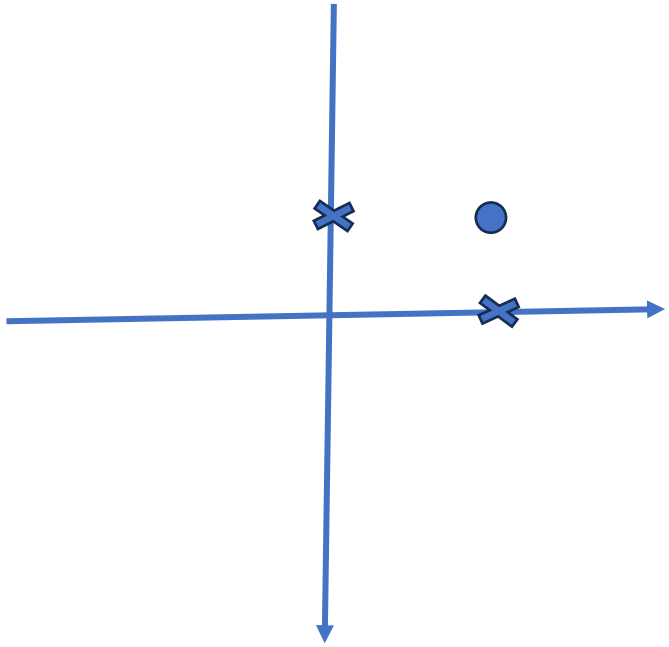
Final PCA plot

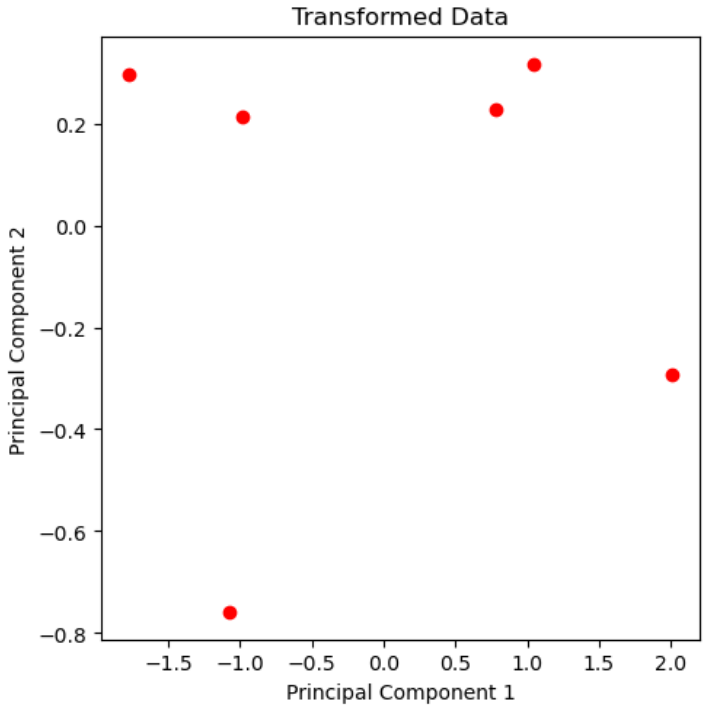
- Project the points on PC1 and PC2 and rotatate to PC1 is horizontal (x-axis) and PC2 is the y-axis



Final PCA plot

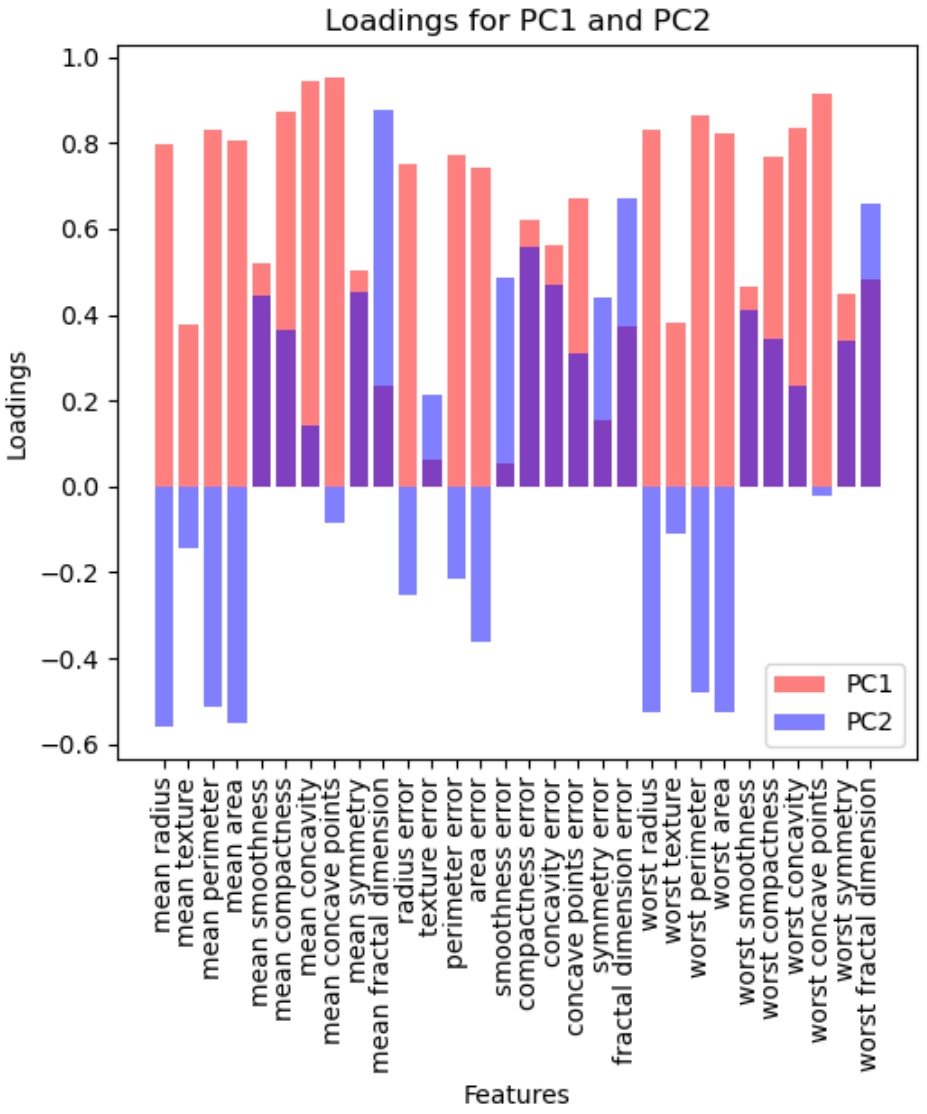
- Use the projected points to find the new coordinates





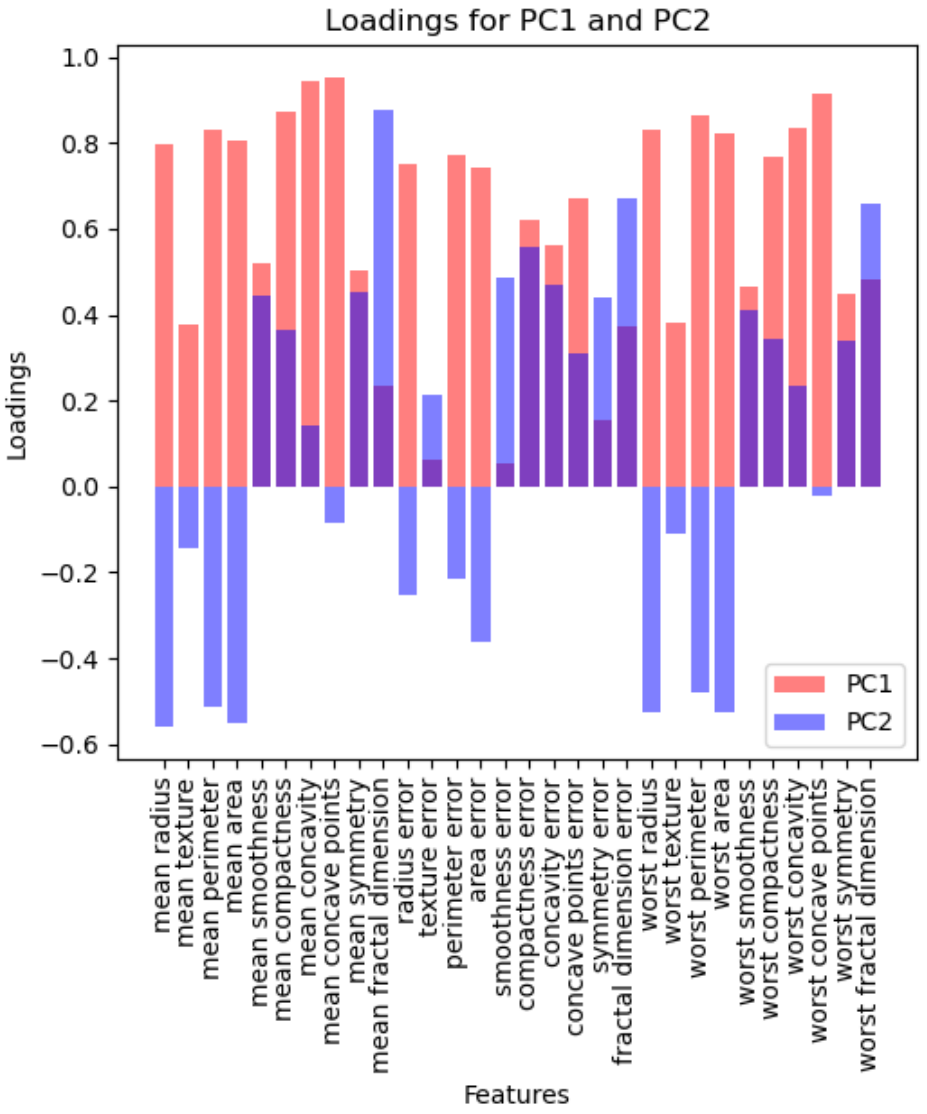
Loadings

- How much each variable contributes to each principal component (correlation between the original variables and the principal components)
- High absolute values: the original variable strongly influences the principal component.



Loadings

- Positive values: the variable and the PC are positively correlated
- Negative values: the variable and the PC are negatively correlated.
- A negative loading indicates that its absence contributes to some degree to the principal component



Summary

Unsupervised learning

- Advantages:
  - Requires less manual data preparation (i.e., no hand labeling) than supervised machine learning.
  - Capable of finding previously unknown patterns in data, which is impossible with supervised machine learning models.
- Disadvantages:
  - Results may be unpredictable or difficult to understand.
  - Difficult to measure accuracy or effectiveness due to lack of predefined answers during training.

Conclusion: clustering

- Clustering looks for “similar” groups of observations
- k-means is simple but needs to predefine k
- Hierarchical Clustering: no need to predefine the number of clusters. Doesn’t work well on vast amounts of data or huge datasets.
- PCA will reduce the dimensionality of your data into principal components. Very popular way to visualise high-dimensional data into a 2-d plot

Practical 7