Introduction to Machine Learning: Unsupervised vs Supervised K-Means Data reduction

CATÓLICA MEDICAL SCHOOL

LISBOA

25/06/2024

Agenda

- Learning: Supervised vs Unsupervised
- Unsupervised Learning: K-means
- Data Reduction – PCA

Agenda

- Learning: Supervised vs Unsupervised
- Unsupervised Learning: K-means
- Data Reduction – PCA

Machine Learning

Machine Learning (ML) is a subset of artificial intelligence (AI) that involves training algorithms to learn from and make predictions or decisions based on data.

There are various types of ML, but we will focus on **supervised** and **unsupervised** learning.

Supervised and Unsupervised Learning

Supervised Learning

Definition: Supervised learning involves training a model on labelled data, which means the input data is paired with the correct output.

Examples: Classification (e.g. spam detection), Regression (e.g. predicting treatment responses).

Unsupervised Learning

Definition: Unsupervised learning involves training a model on data without labelled responses, aiming to find hidden patterns or intrinsic structures in the data.

Examples: Clustering (e.g. patient segmentation), Association (e.g. identifying comorbidity patterns in patients)

Supervised and Unsupervised Learning – Key Differences

Supervised Learning

- Requires labelled data
- Focuses on prediction and classification
- More straightforward evaluation metrics
- *Most commonly used as it is more reliable*

Supervised Learning

- Does not require labelled data
- Focuses on finding hidden patterns
- Evaluation can be more subjective
- *Easier to use and can be applied to explore data*

Supervised and Unsupervised Learning – Pros & Cons

Supervised Learning

Pros

+ High accuracy with enough labelled data
+ Easy to understand and interpret results
+ Well-defined evaluation metrics

Cons

- Requires a large amount of labelled data.
- Can be time-consuming and expensive to label data
- Overfitting can occur if the model is too complex

Unsupervised Learning

Pros

+ Can work with unlabelled data, which is more readily available
+ Useful for discovering hidden patterns
+ Can adapt to new data without retraining

Cons

- Results can be less interpretable
- Evaluation of results can be challenging
- May require domain expertise to interpret clusters or patterns

Remember this from Linear Regression?...

$$\text{DATA} = \text{MODEL} + \text{ERROR}$$

…which can be written in terms of variances (sums of squares, SS)

SSTotal = SSRegression + SSError

$$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2 = (n-1)\, S'^2$$

$$SSR = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$

$$SSE = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Why is Regression a Supervised Learning algorithm?

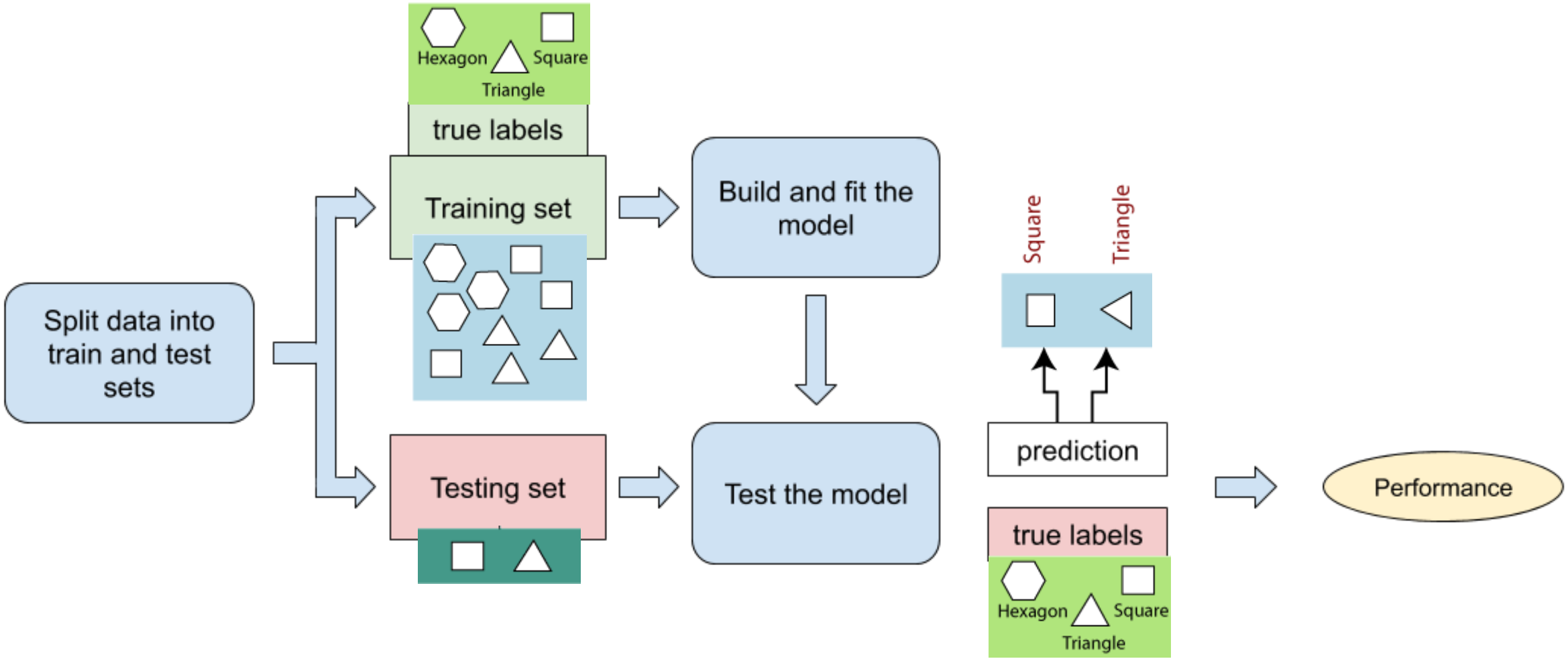
Observed data points, labelled

Difference between the actual data and the model's predictions

DATA = MODEL + ERROR

Function or algorithm used to make predictions for ***new*** (unlabelled) data points

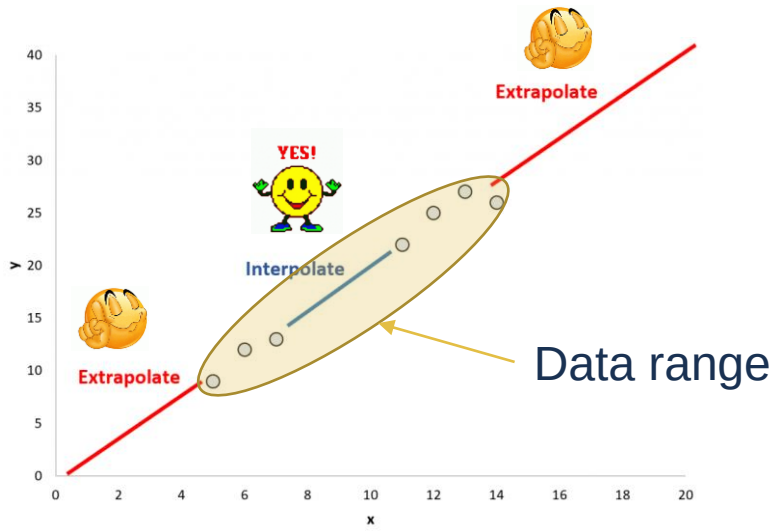
Supervised algorithms: Learning through labelled data

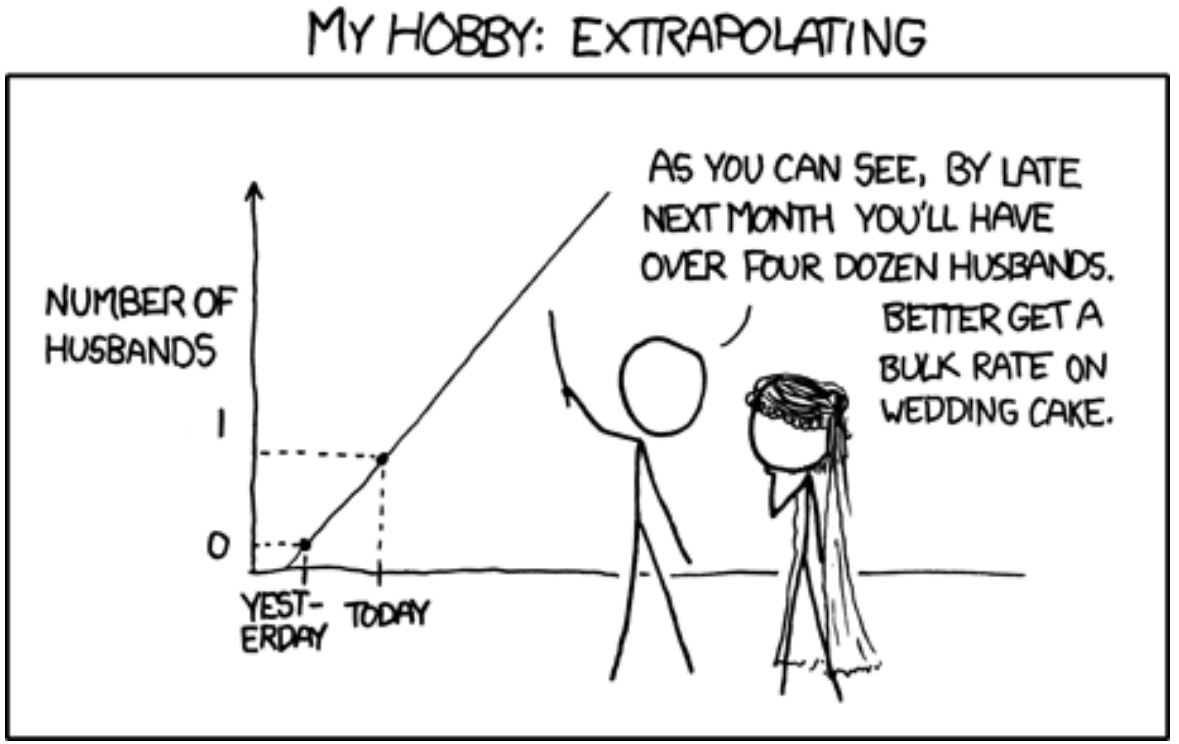


Source: https://www3.tuhh.de/sts/hoou/data-quality-explored/0-2-supervised-learning.html; https://www.javatpoint.com/supervised-machine-learning

Supervised algorithms – careful with extrapolations…

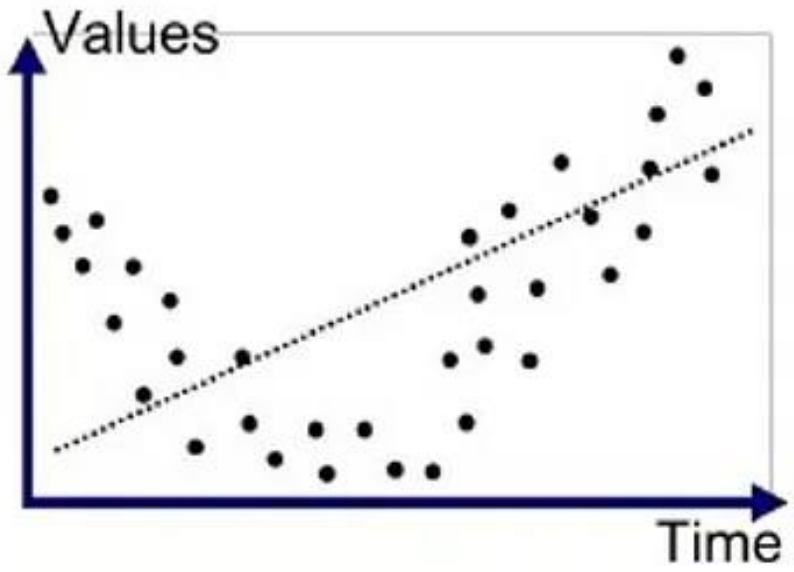
You can interpolate, but NOT extrapolate regression models!



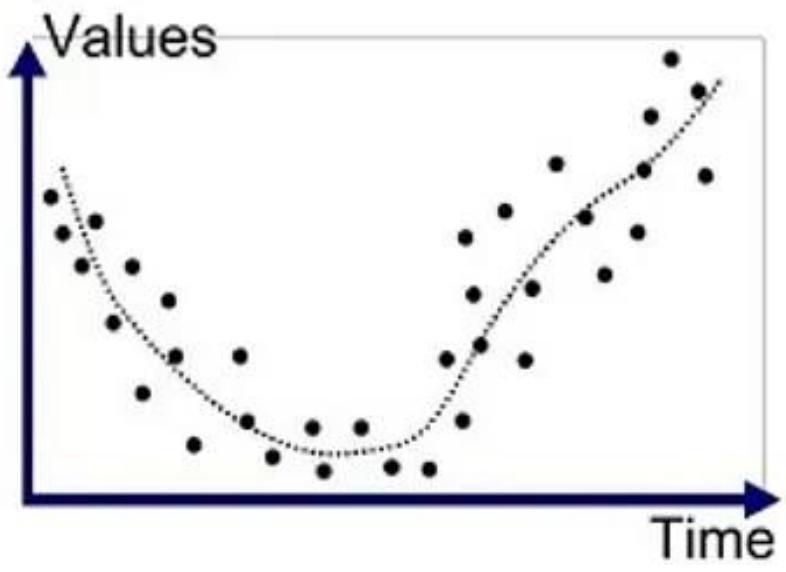


Source: https://www.statology.org/interpolation-vs-extrapolation/

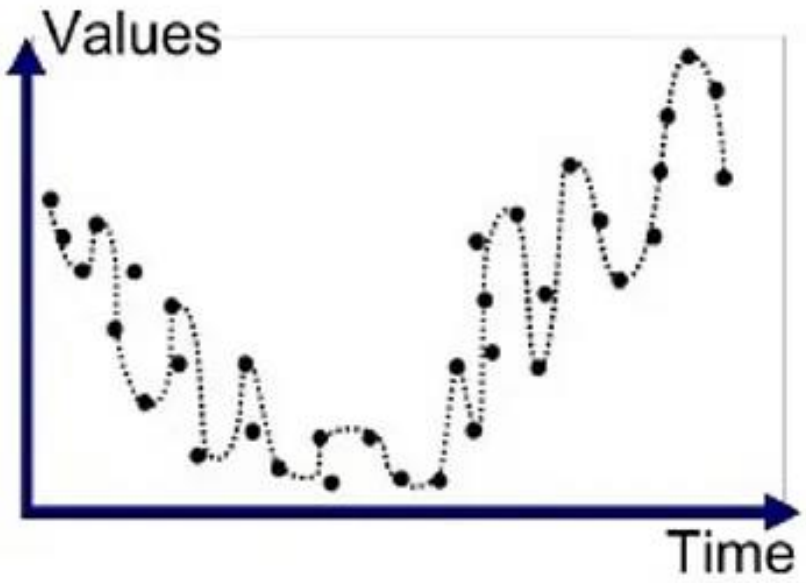
Supervised algorithms – …and with overfitting



Underfitted



Good Fit/Robust



Overfitted

Source: https://medium.com/@cs.sabaribalaji/overfitting-6c1cd9af589

Unsupervised Algorithms – find your own labels!

In **supervised** learning, the goal is to build a model that predicts the output (label) as accurately as possible given the input data, and the error represents the difference between the predicted and actual outputs.

However, in **unsupervised** learning, we do not have labelled data to directly calculate an error similarly. Instead, **the focus is on finding hidden patterns, structures, or relationships in the data**. While the concept of **modelling and minimizing error still exists in some forms**, it is applied differently.

In summary

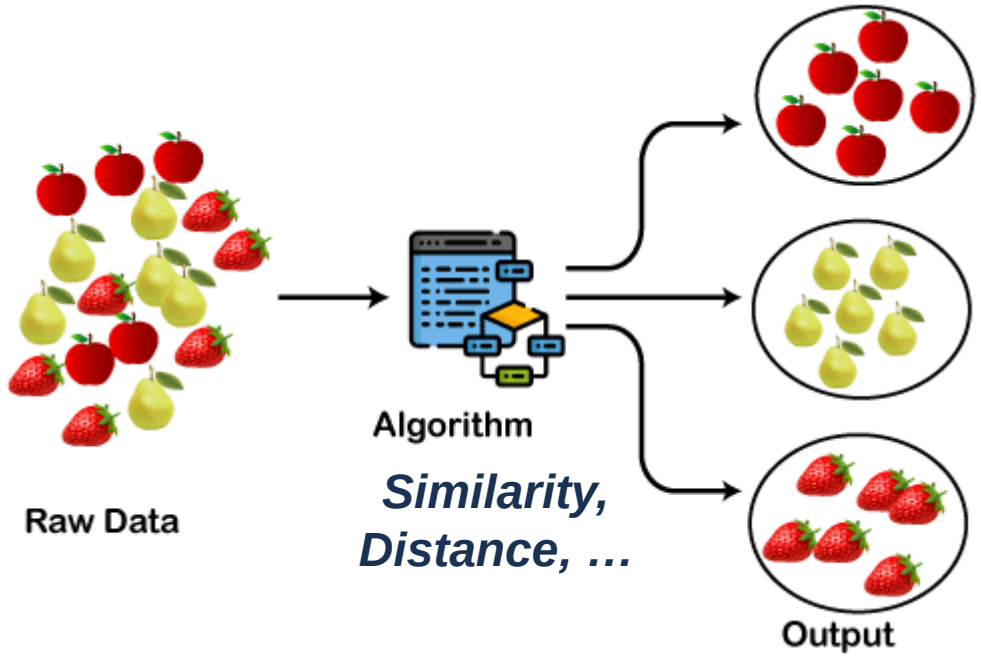
Source: https://www.youtube.com/watch?v=kE5QZ8G_78c

Agenda

- Learning: Supervised vs Unsupervised
- Unsupervised Learning: K-means
- Data Reduction – PCA

Clustering

- Clustering is used to group similar data points together in an **unsupervised** manner
- Clustering is the process of arranging a group of objects in such a manner that the objects in the same group (i.e. ***cluster***) are more similar to each other than to the objects in any other group
- Often used in the Exploratory Data Analysis (EDA) phase to discover new information and patterns



Source: https://www.datacamp.com/blog/clustering-in-machine-learning-5-essential-clustering-algorithms

Clustering – Disease Management

Hierarchical clustering analysis for predicting 1-year mortality after starting hemodialysis

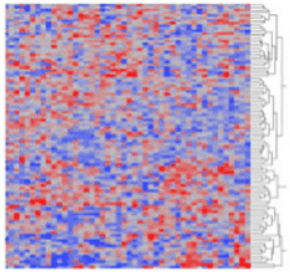
PATIENTS & DESIGN

101 patients
- started hemodialysis
- Used baseline demographics and laboratory data

Prospective observational study
- followed for 1 year
- Death and length of hospital stay

METHODS & CLUSTERING

Agglomerative hierarchical clustering
- Included 46 variables
- Classified into **3 clusters:**



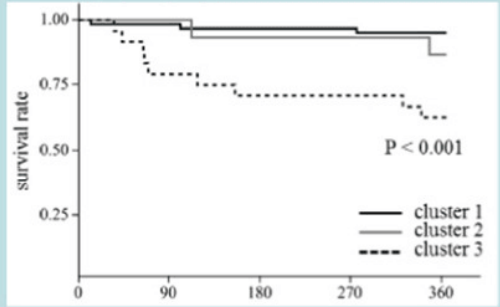
cluster 1:
- The largest cohort
- Low WBC & CRP

cluster 2:
- High BNP & serum K

cluster 3:
- Not hypertensive
- Low serum creatinine
- Low urinary L-FABP

RESULTS

1-year Mortality:
Cluster 1 (4.8%) < Cluster 3 (37.5%)
Hospital stay (days): Cluster 1 < Clusters 2, 3



Abbreviations: BNP, B-type natriuretic peptide; CRP, C-reactive protein; L-FABP, liver-type fatty acid-binding protein; WBC, white blood cell

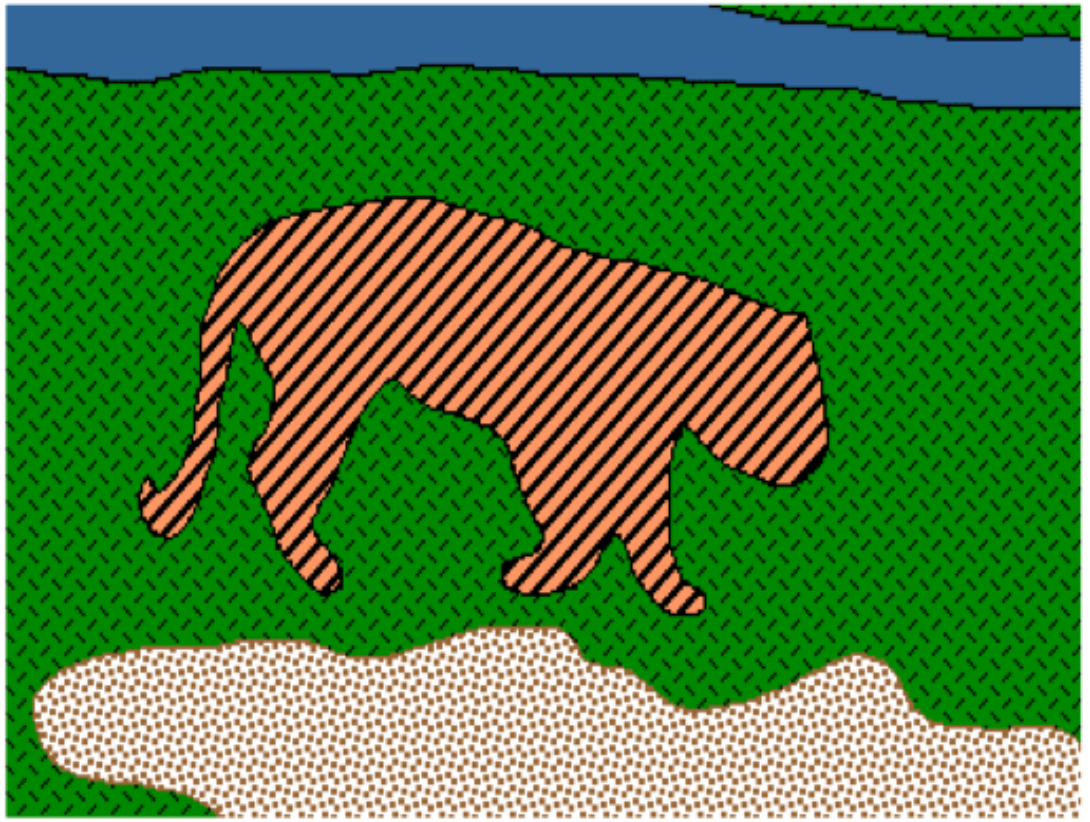
KIREPORTS KIReports.org

Komaru & Yoshida et al, 2020

CONCLUSION:
Agglomerative hierarchical clustering was applied to patients newly starting maintenance HD. The resulting clusters were associated with 1-year mortality and length of hospital stay.

Source: https://els-jbs-prod-cdn.jbs.elsevierhealth.com/cms/attachment/da4cb0c9-0a86-4702-8a78-80ffffcf1f9c/fx1_lrg.jpg

Clustering – Image Segmentation



Source: https://els-jbs-prod-cdn.jbs.elsevierhealth.com/cms/attachment/da4cb0c9-0a86-4702-8a78-80ffffcf1f9c/fx1_lrg.jpg

Types of Clustering

- **K-Means**
- MeanShift
- DBSCAN
- Hierarchical clustering
- BIRCH
- GMM
- …

Choice based on:

- ✓ Parameters required for the model
- ✓ Scalability
- ✓ Use-cases
- ✓ Geometry, i.e., metric used for calculation of distances

(You can find 10 methods in `scikit-learn`, for instance)

Source: https://www.datacamp.com/blog/clustering-in-machine-learning-5-essential-clustering-algorithms

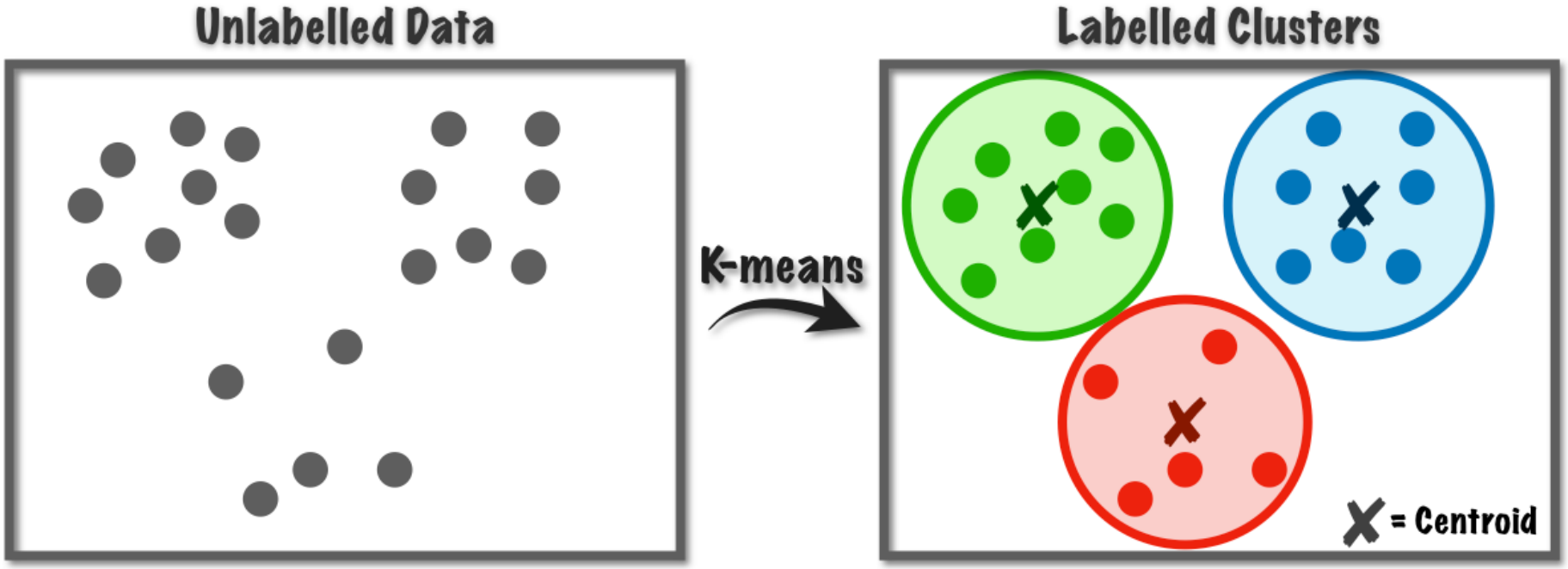
K-means

- **Definition**: k-Means is an unsupervised machine learning algorithm used for clustering. It partitions the data into “k” distinct, non-overlapping subsets (clusters)

- **Objective**: To minimize the error between data points and the centroid of their respective clusters.

- **Applications**: Group patients with similar characteristics for targeted treatment plans; Identify patterns and clusters in epidemiological data to detect and monitor disease outbreaks; Segment and analyze medical images for accurate diagnosis and treatment planning.

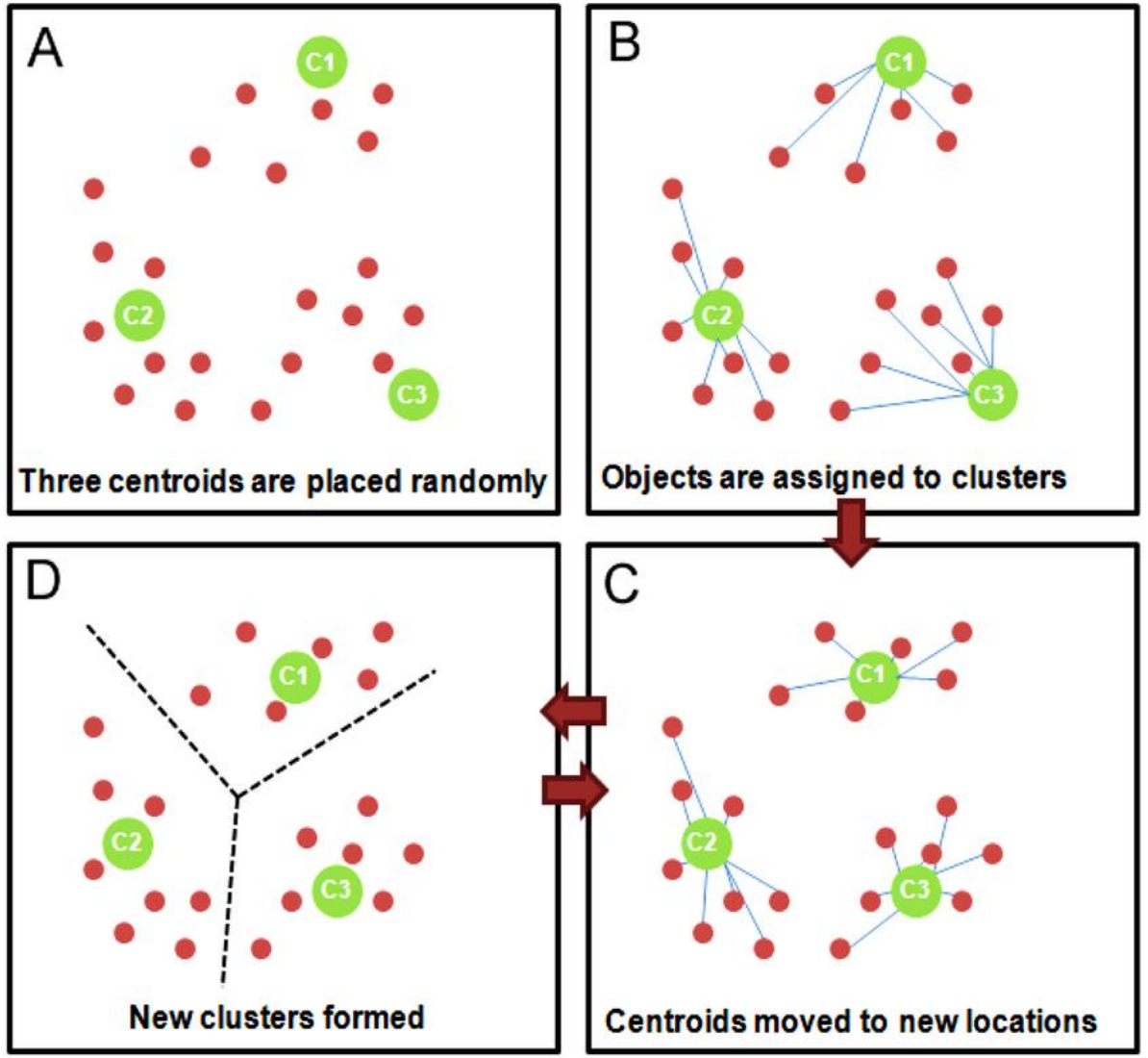
It clusters data, creating its own labels



- In K-means, each cluster is represented by its center (called a “centroid”), which **corresponds to the arithmetic mean of data points assigned to the cluster**.
- A centroid is a data point that represents the center of the cluster (the mean), and it might not necessarily be a member of the dataset.

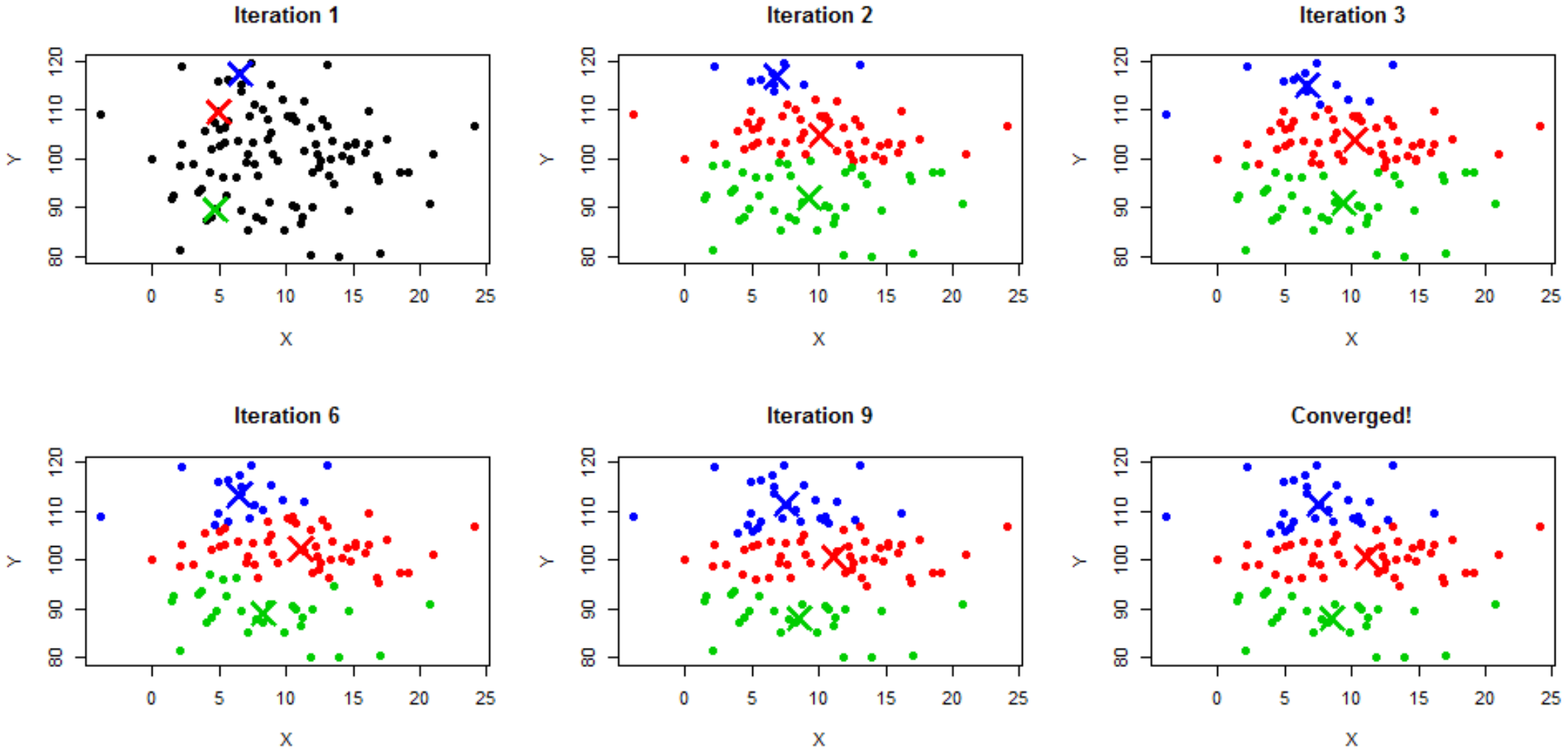
Source: https://towardsdatascience.com/k-means-a-complete-introduction-1702af9cd8c

In more detail



Source: https://www.researchgate.net/figure/K-means-clustering-A-Starting-with-three-randomly-placed-centroids-green-B-Next_fig2_276494113

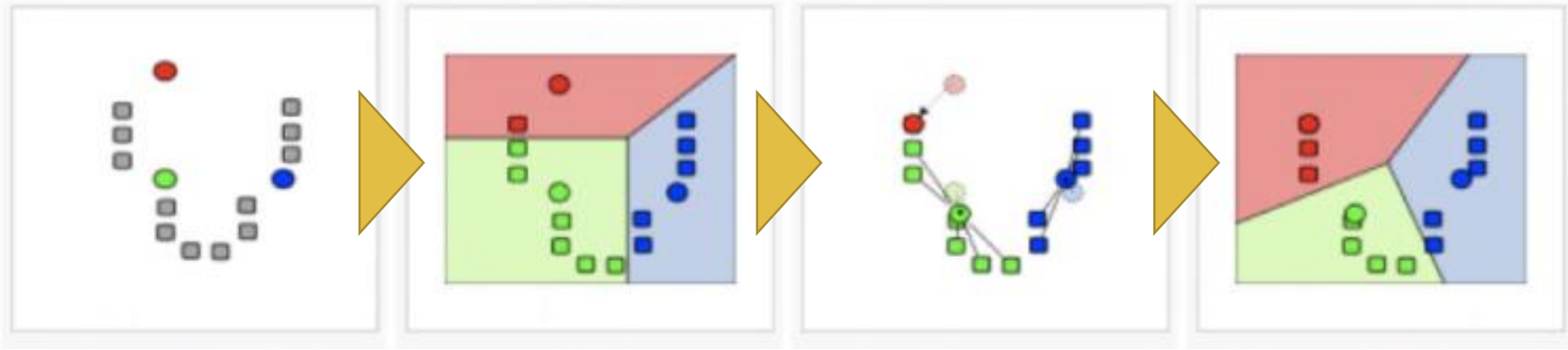
Another example



Source: https://www.learnbymarketing.com/wp-content/uploads/2015/01/method-k-means-steps-example.png

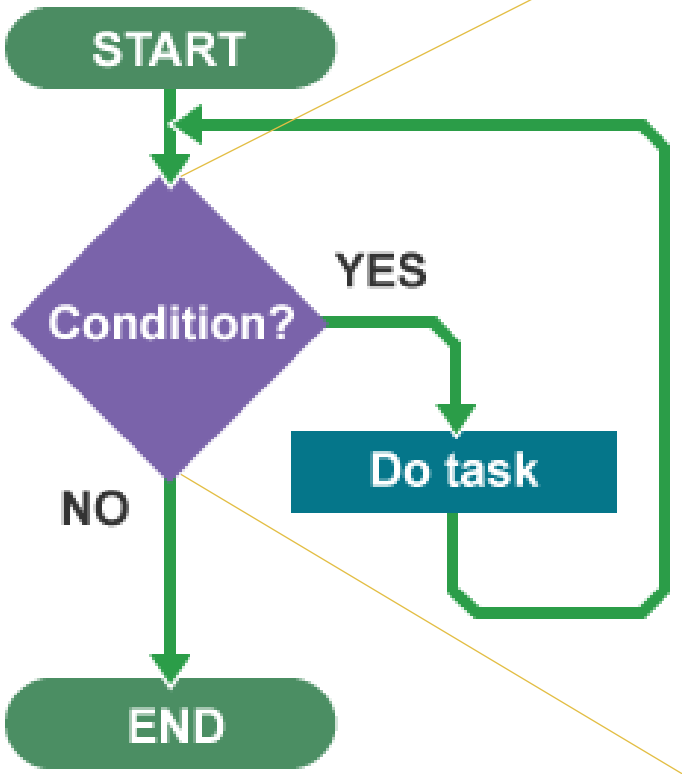
Steps

1. **Initialization:** Choose k initial centroids randomly from the dataset
2. **Assignment:** Assign each data point to the nearest centroid, forming k clusters
3. **Update:** Calculate the new centroids as the mean of all data points assigned to each cluster
4. **Iteration:** Repeat the assignment and update steps until the centroids no longer change or a maximum number of iterations is reached
5. **Convergence:** The algorithm converges when the centroids stabilize



http://www.astro.caltech.edu/~george/aybi199/Donalek_Classif.pdf

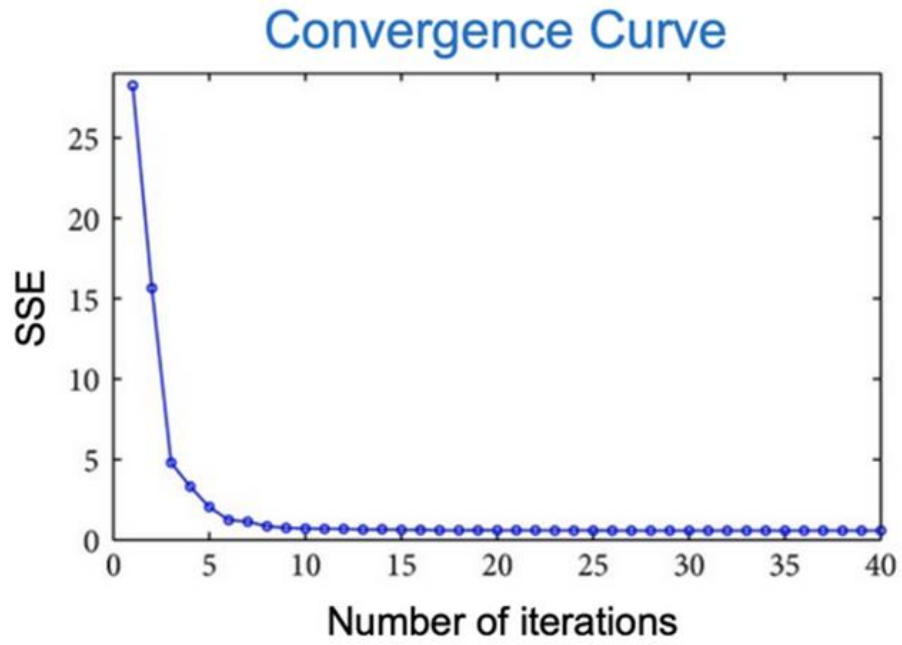
Iteration and Convergence



Sum of Squared Errors (SSE)

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

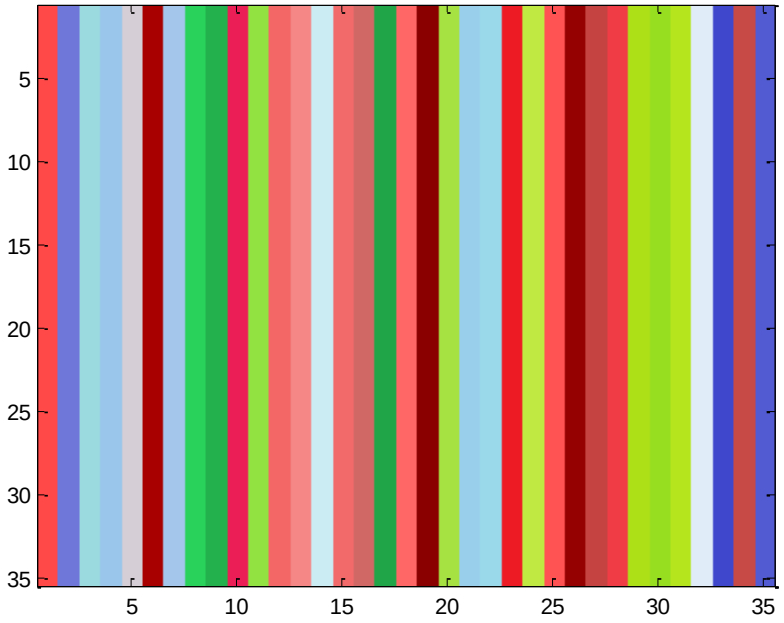
x is a data point in cluster C_i and m_i is the centroid of cluster C_i



SSE is monotonically decreasing

Source: https://www.chegg.com/homework-help/questions-and-answers/need-help-plotting-convergence-curve-k-means-python-scikit-learn-cannot-figure-get-output--q104780162; https://www.bbc.co.uk/bitesize/guides/zcg9kqt/revision/7

K-means clustering: 3D



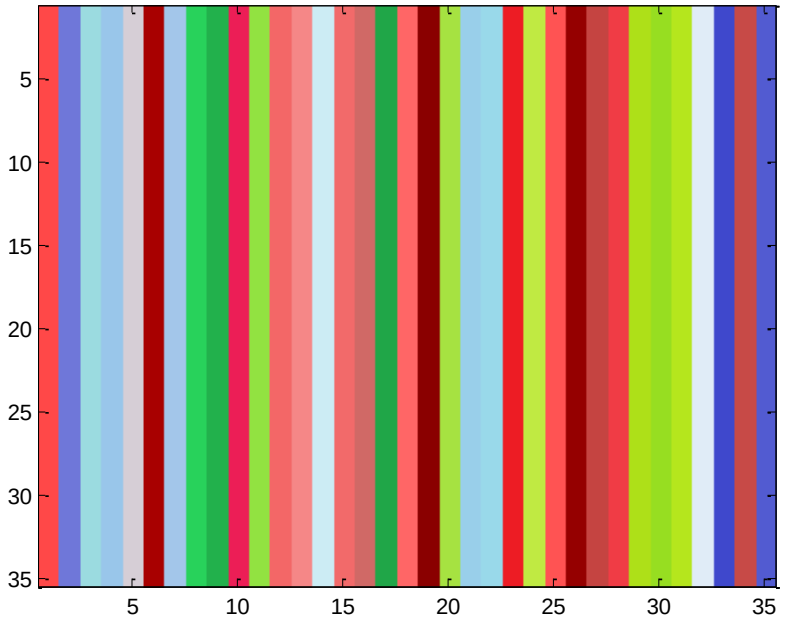
- ***Challenge***: Use 3 colors to sort each column of the figure

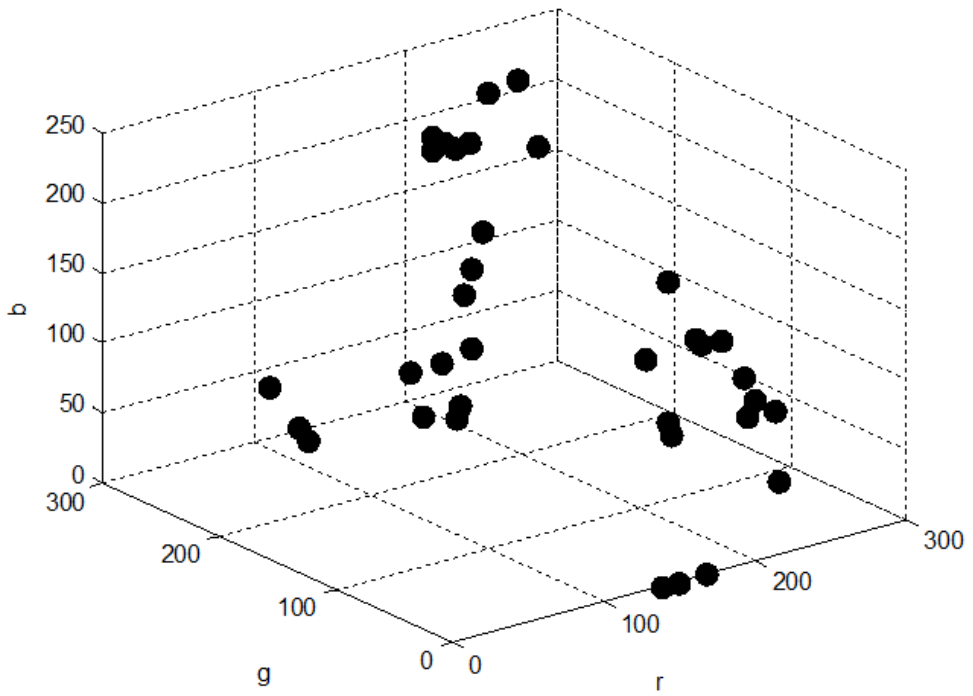
- Which algorithm can we use?

- Let's use the means algorithm, with k=3

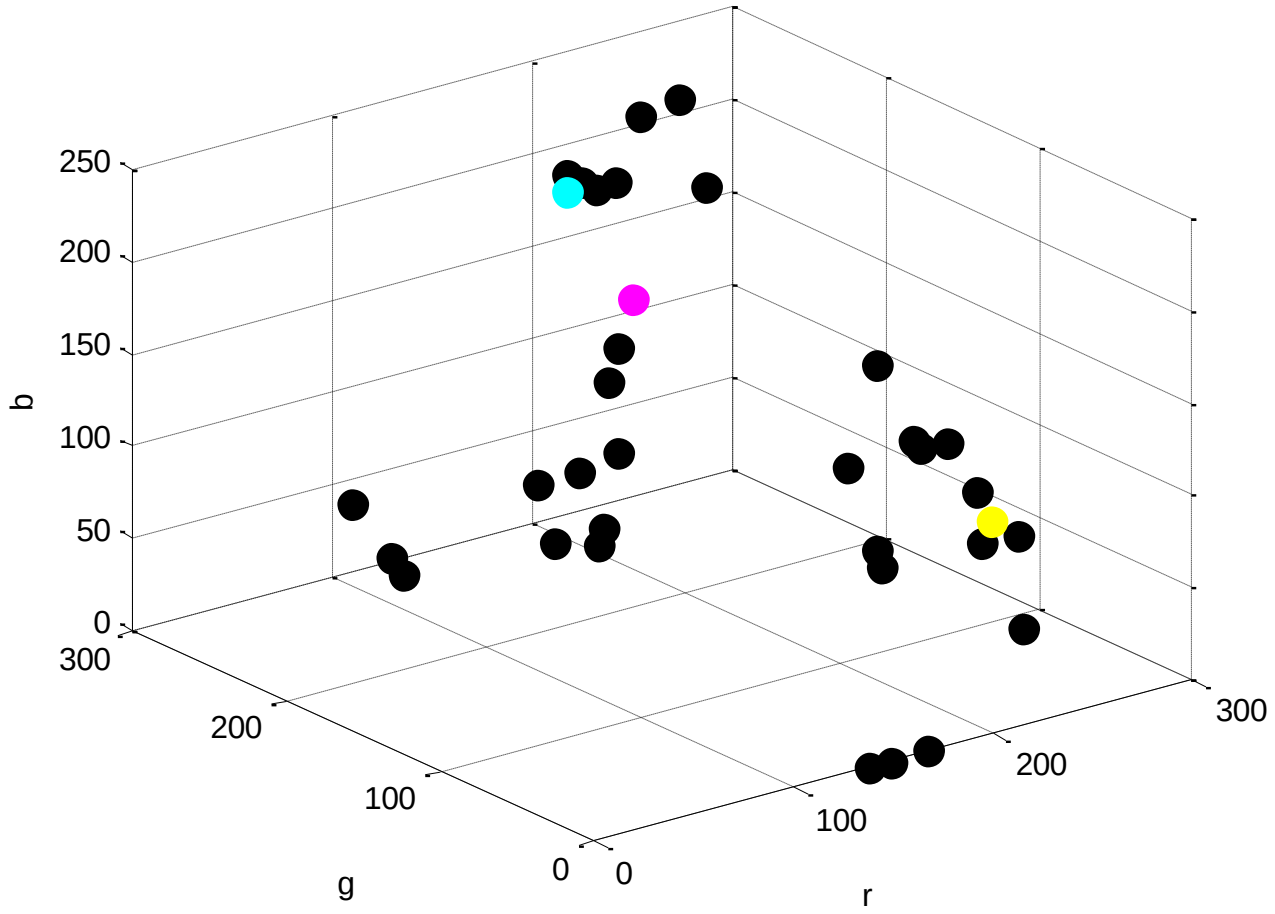
K-means clustering: 3D

The coordinates (R, G, B) of the colors in the figure can be represented in a 3D graph:

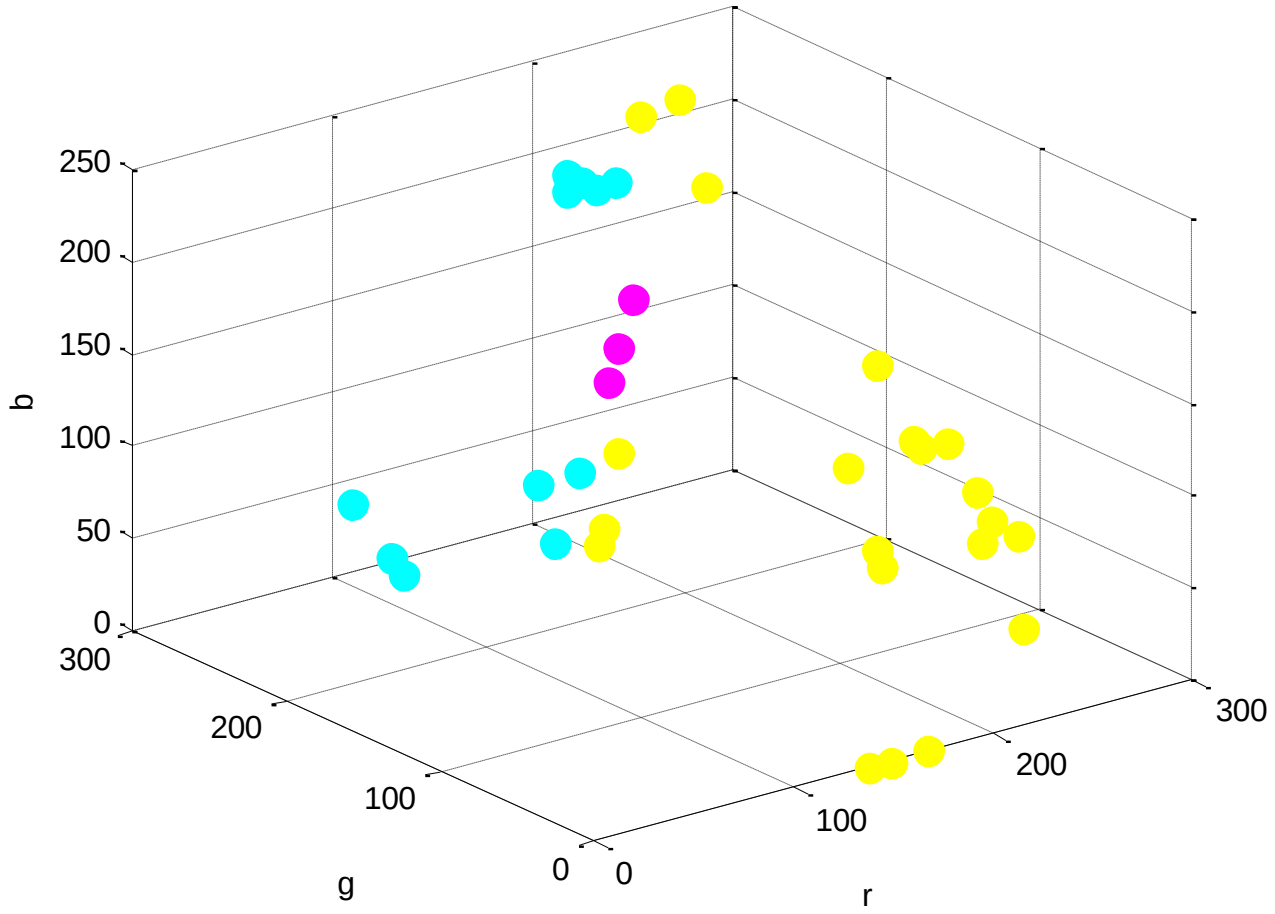




Start with 3 arbitrary centroids

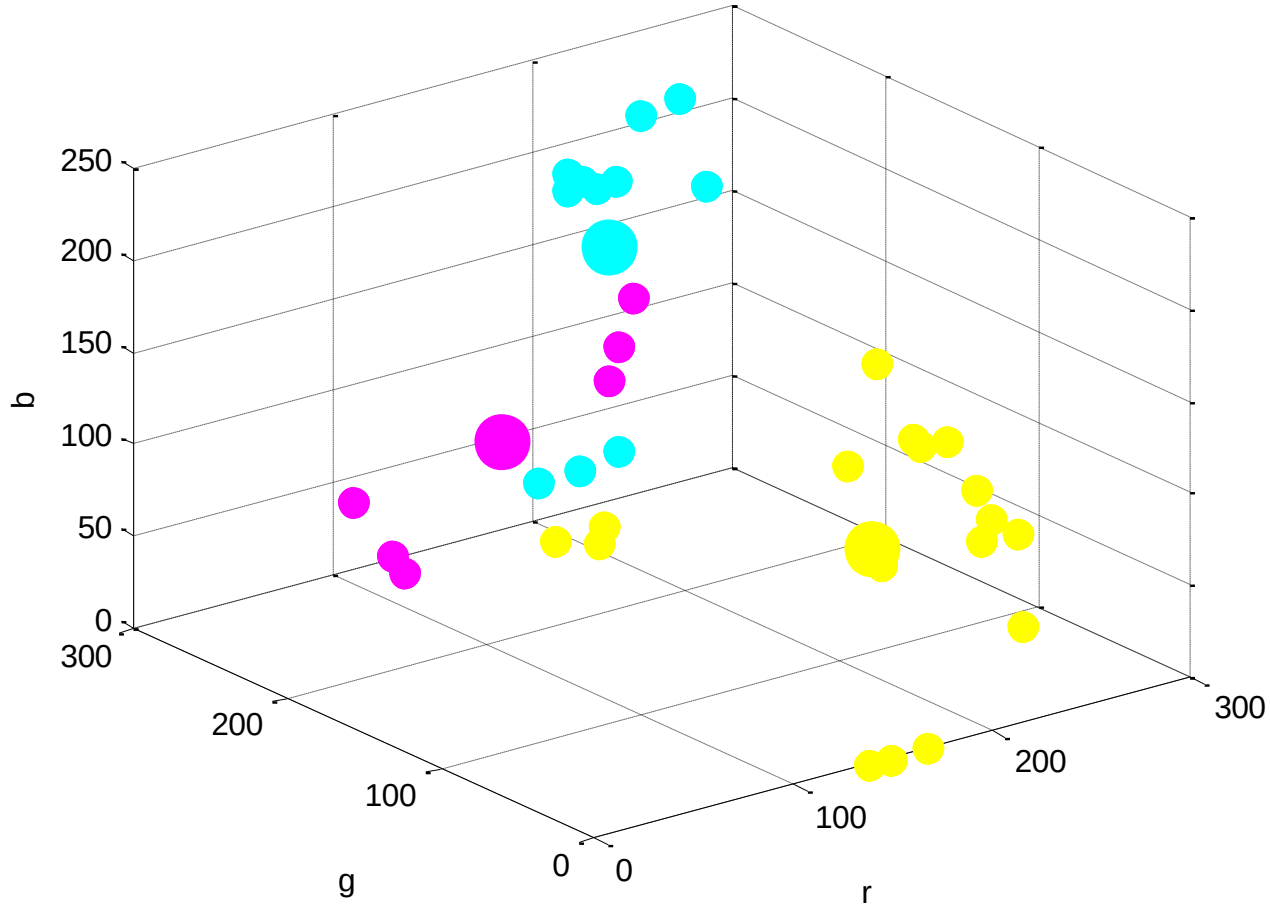


Each point is associated with the nearest centroid

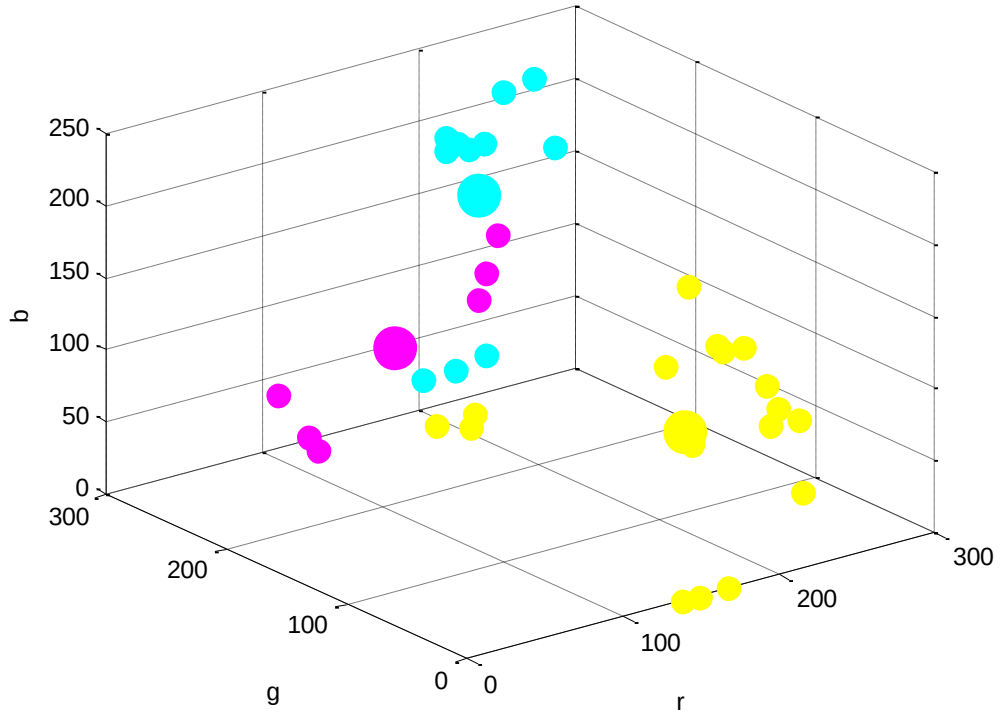


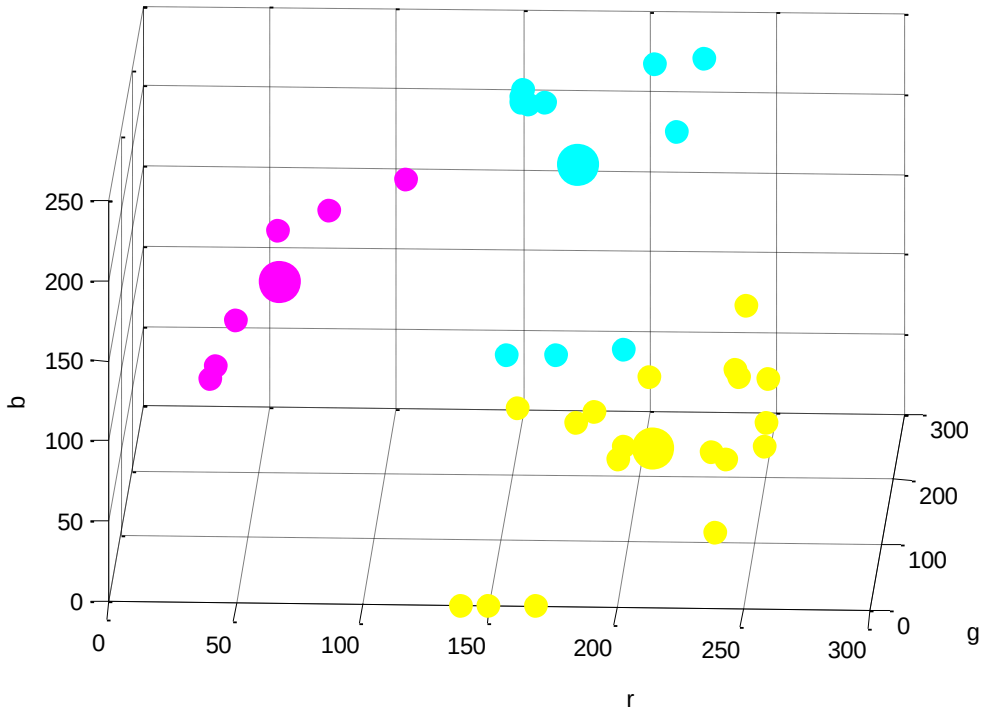
Clustering

Each point is associated with the nearest centroid (and new centroids are calculated)

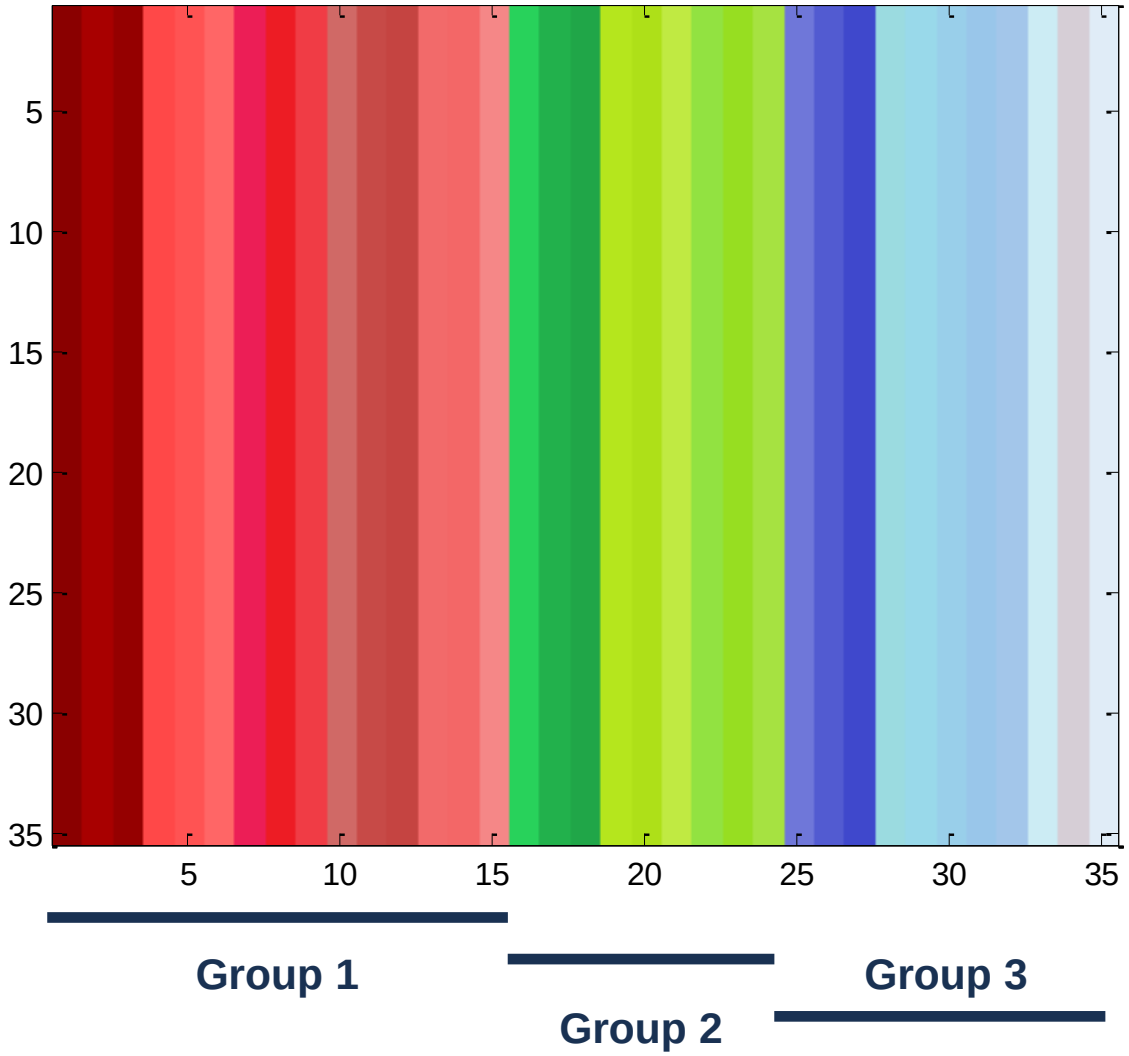


The process continues until convergence is achieved



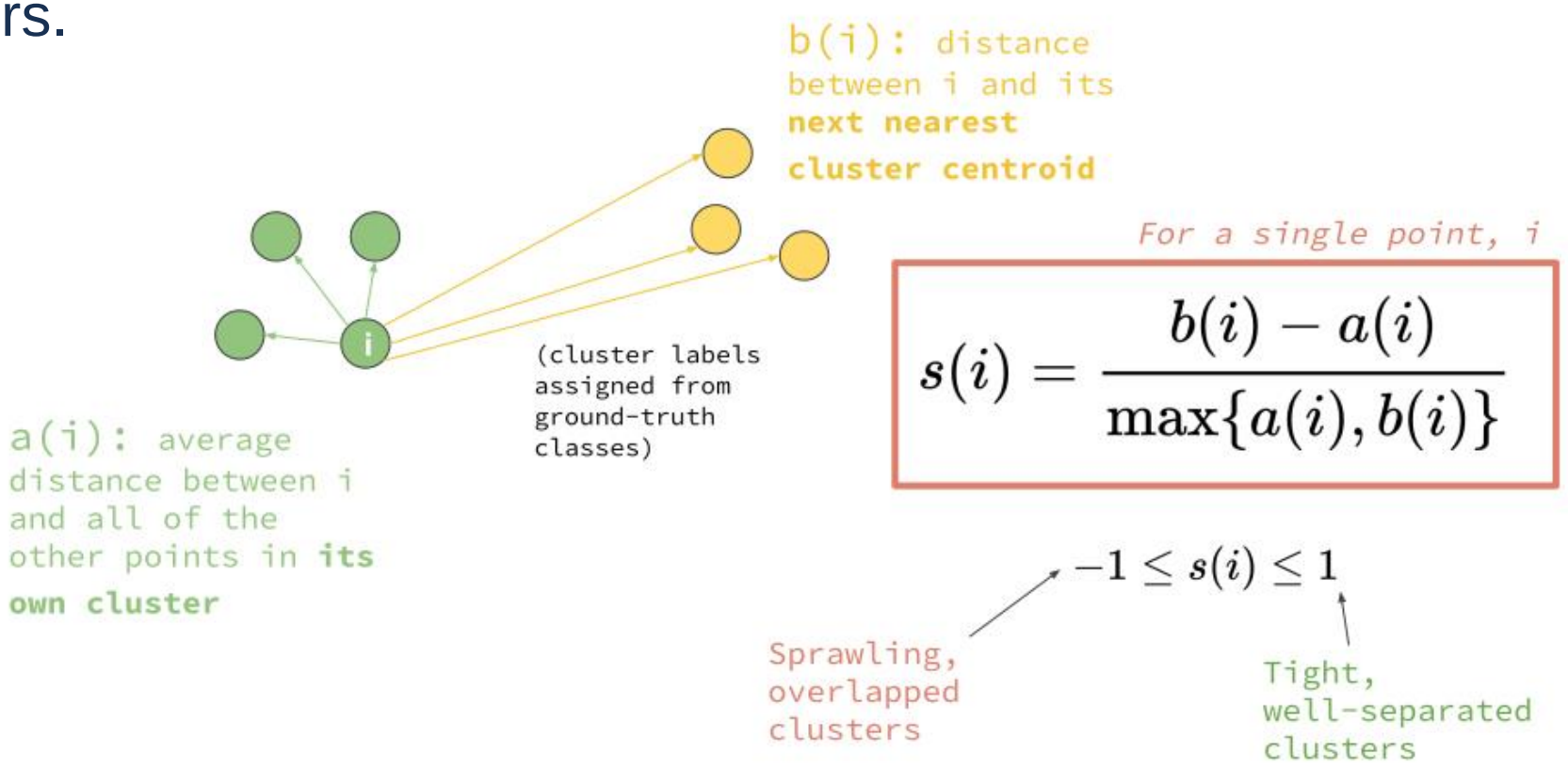


K-means clustering: 3D – Final Result



How good is my clustering? Silhouette score

The silhouette score measures how similar an object is to its own cluster compared to other clusters. It ranges from -1 to 1, where a higher value indicates better-defined clusters.



Source: https://www.platform.ai/post/the-silhouette-loss-function-metric-learning-with-a-cluster-validity-index

How good is my clustering? Squared Error

$$SSE = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

It gives an overall measure of the average distance between the data points of each cluster to its centroid.

It is better for assessing data fit in general, whereas the silhouette score is useful for understanding the cohesion or separation of clusters

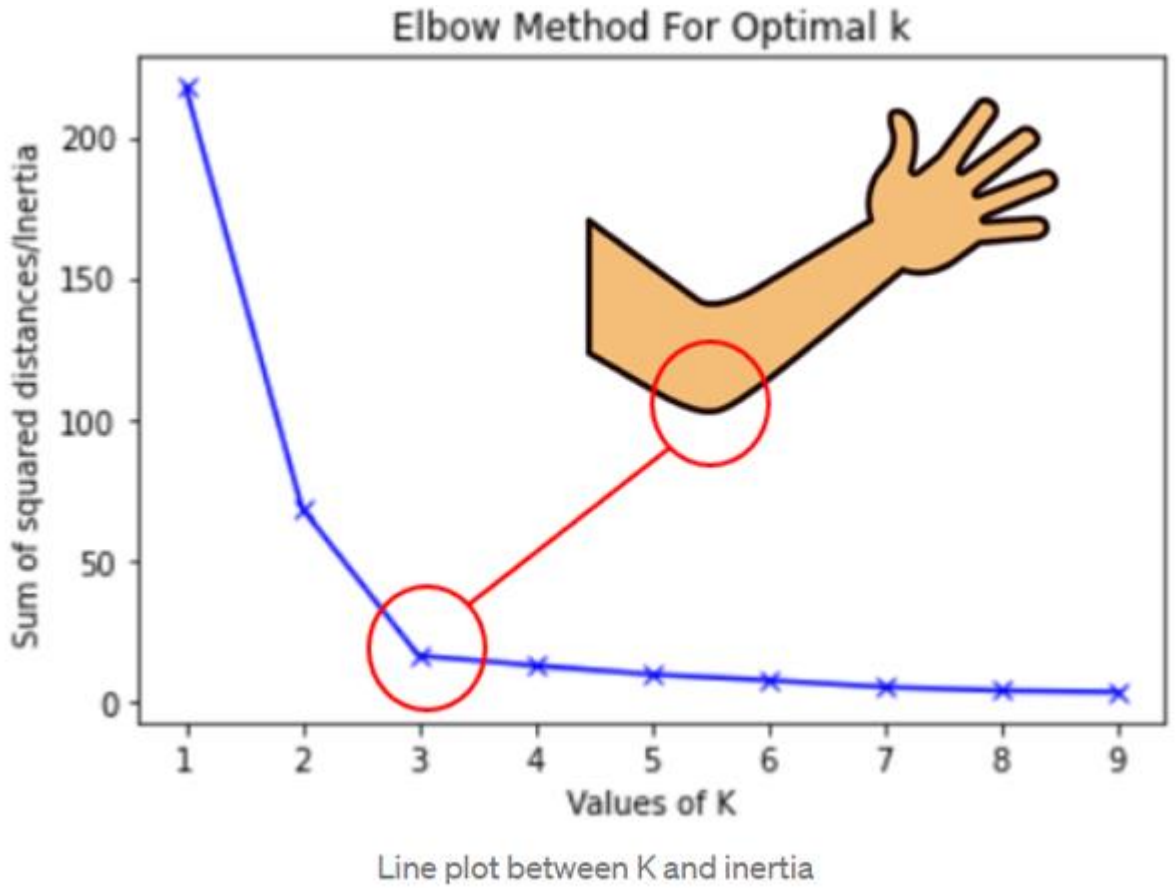
Choosing the right number of clusters, k
Elbow Method, with sum of squares

Description:

- Plot the sum of squared errors (SSE), or a similar metric, against the number of clusters k
- SSE decreases as k increases, but at some point, the rate of decrease sharply slows down, forming an "elbow" in the graph
- The "elbow" point suggests a balance between the number of clusters and the SSE

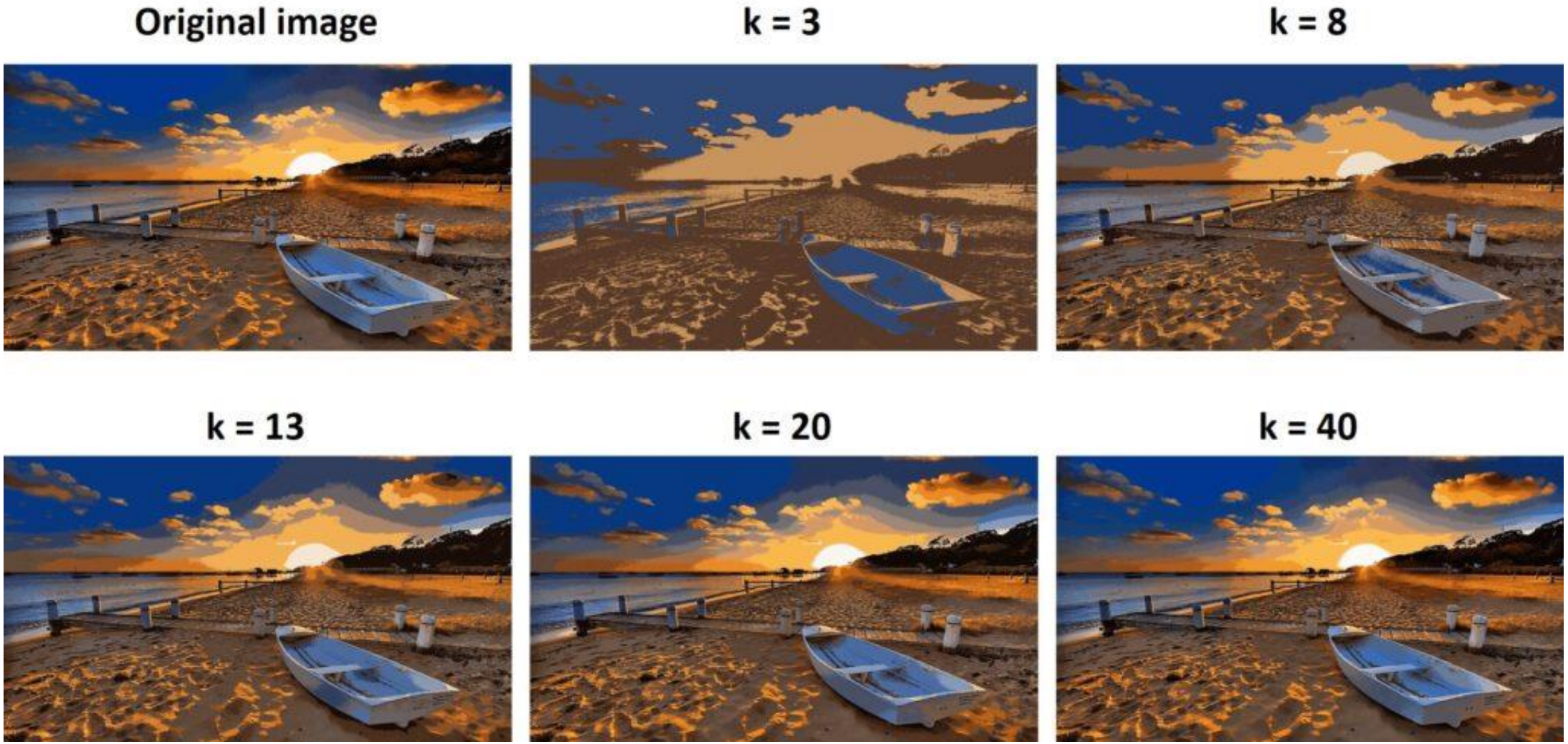
Steps:

1. Run k-means clustering for a range of k values (e.g., 1 to 9)
2. Calculate the SSE for each k
3. Plot SSE against k
4. Identify the elbow point where the rate of decrease sharply changes



Source: https://www.analyticsvidhya.com/blog/2021/05/k-mean-getting-the-optimal-number-of-clusters/

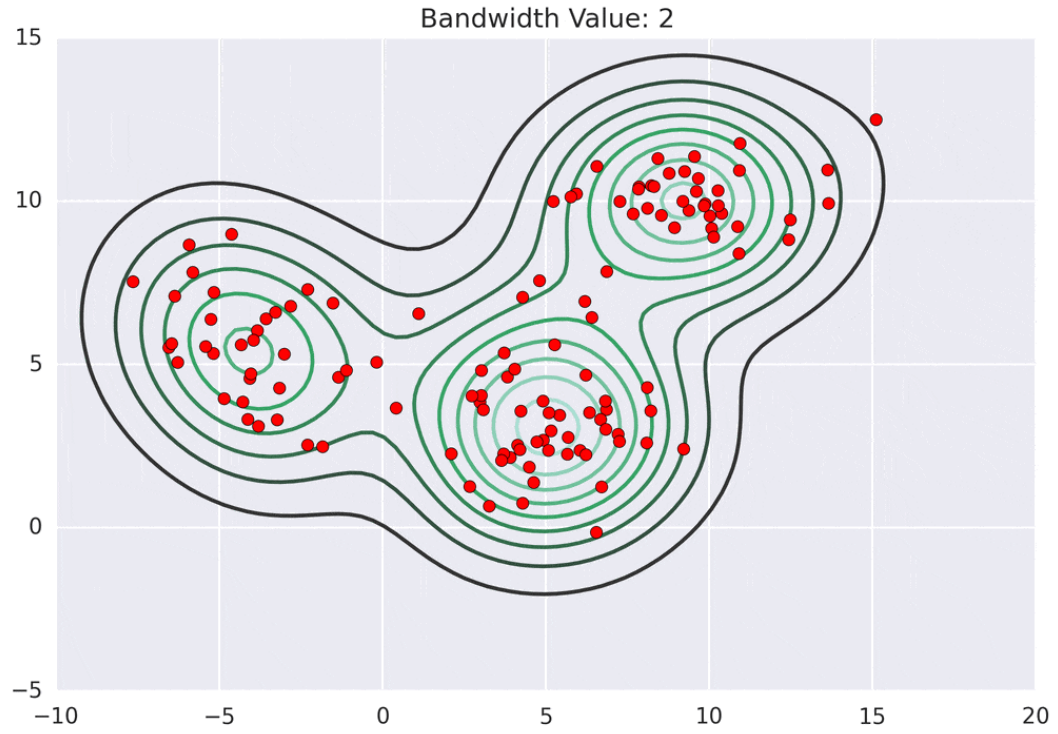
Impact of k



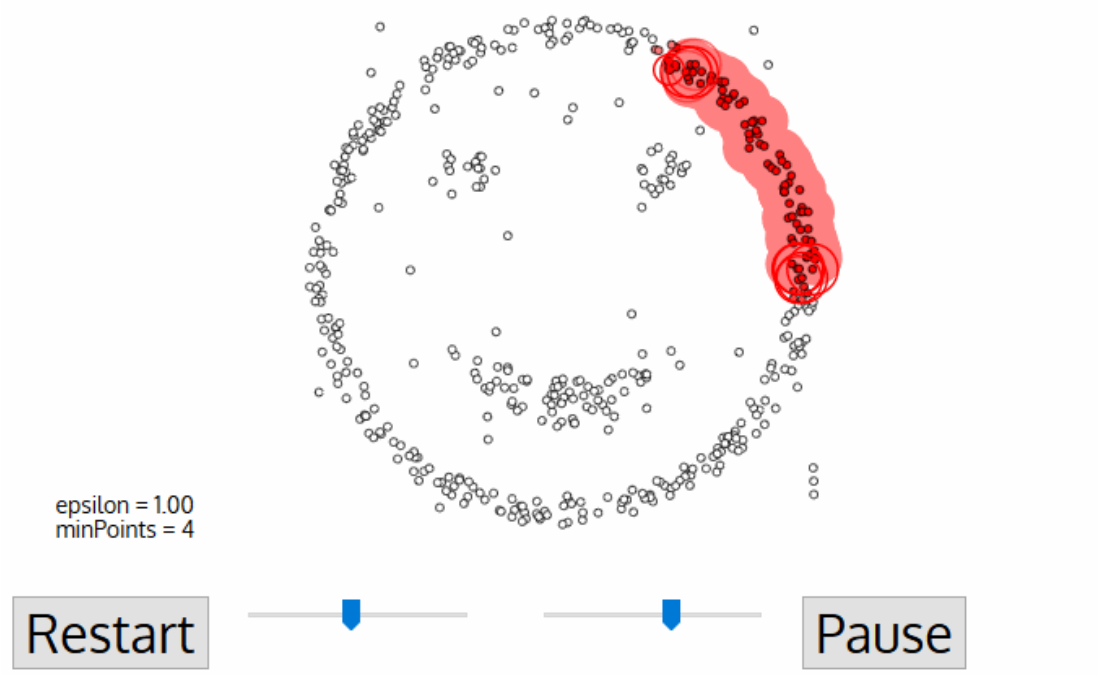
Source: https://datahacker.rs/007-color-quantization-using-k-means-clustering/

Some methods do not require “k”…

MeanShift



DBSCAN



Source: https://spin.atomicobject.com/mean-shift-clustering/; https://www.datacamp.com/blog/clustering-in-machine-learning-5-essential-clustering-algorithms

K-means: Pros & Cons

Pros

+Simple and easy to implement
+Computationally efficient for large datasets
+Works well with spherical clusters

Cons

- Requires specifying the number of clusters k in advance
- Sensitive to initial centroid placement
- Struggles with clusters of varying sizes and densities
- Not suitable for clusters with non-convex shapes

Agenda

- Learning: Supervised vs Unsupervised
- Unsupervised Learning: K-means
- Data Reduction – PCA

Data Reduction

- Data reduction is the process of reducing the amount of data while preserving the important information

- Helps in handling large datasets, improves computational efficiency, and enhances data visualization – i.e., it ***simplifies***

- It can be used for:
  - ✓ Model Simplification: reducing the number of parameters required to apply a model
  - ✓ Data visualization: makes for a simpler visual assessment
  - ✓ Prototype Selection: choice of adequate training set
  - ✓ Feature Selection: choice of the variables that best (and more fully) explain the observed variance

PCA – Principal Component Analysis

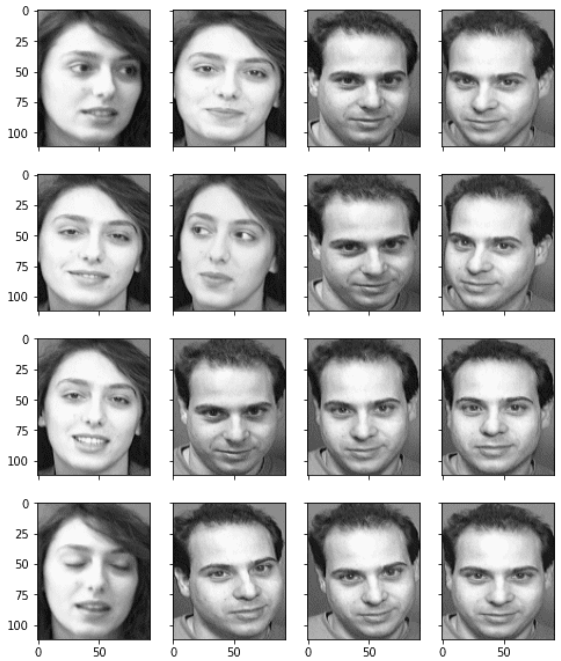
- PCA is a statistical procedure that converts observations of possibly correlated variables into a set of linearly uncorrelated variables called principal components

- The purpose is to ***reduce the dimensionality of the data while retaining most of the variation present in the dataset***

- It retains the traits that contribute most to the observed variance, and uses an orthogonal transformation to obtain a set of linearly independent values

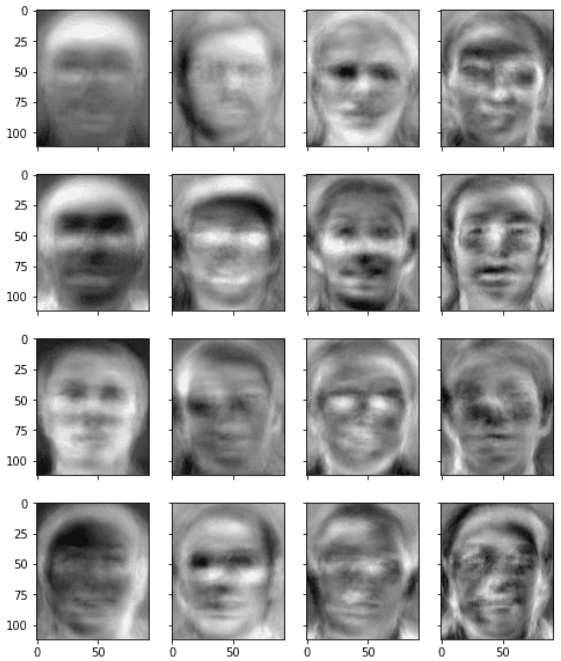
PCA – Facial Recognition Example

Training set

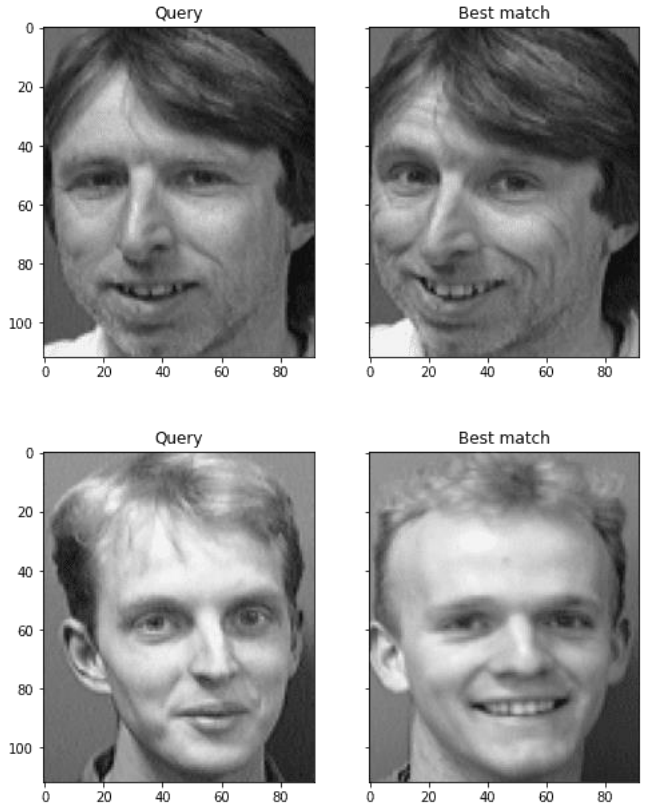


...

Eigenfaces



Results



Source: https://machinelearningmastery.com/face-recognition-using-principal-component-analysis/ - Scaling and centering is key…

PCA – What is happening?

Covariance Matrix: Measures the covariance (how much two random variables vary together) between different variables

Eigenvalues and Eigenvectors: Used to determine the principal components

Steps to Compute PCA:

1. Standardize the data.
2. Calculate the covariance matrix.
3. Compute eigenvalues and eigenvectors of the covariance matrix.
4. Select the top k eigenvectors to form a new subspace.
5. Transform the original data into the new subspace.

Covariance and Correlation

- A **correlation coefficient** *quantifies* a **linear association** between two **quantitative**
- It is a measure of how both variables **covary** between them
- It is a normalization (between –1 and 1) of the value of **covariance**

Pearson's correlation coefficient:

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y}$$

Population

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Sample

$$COV(X,Y) = \frac{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right)}{n-1}$$

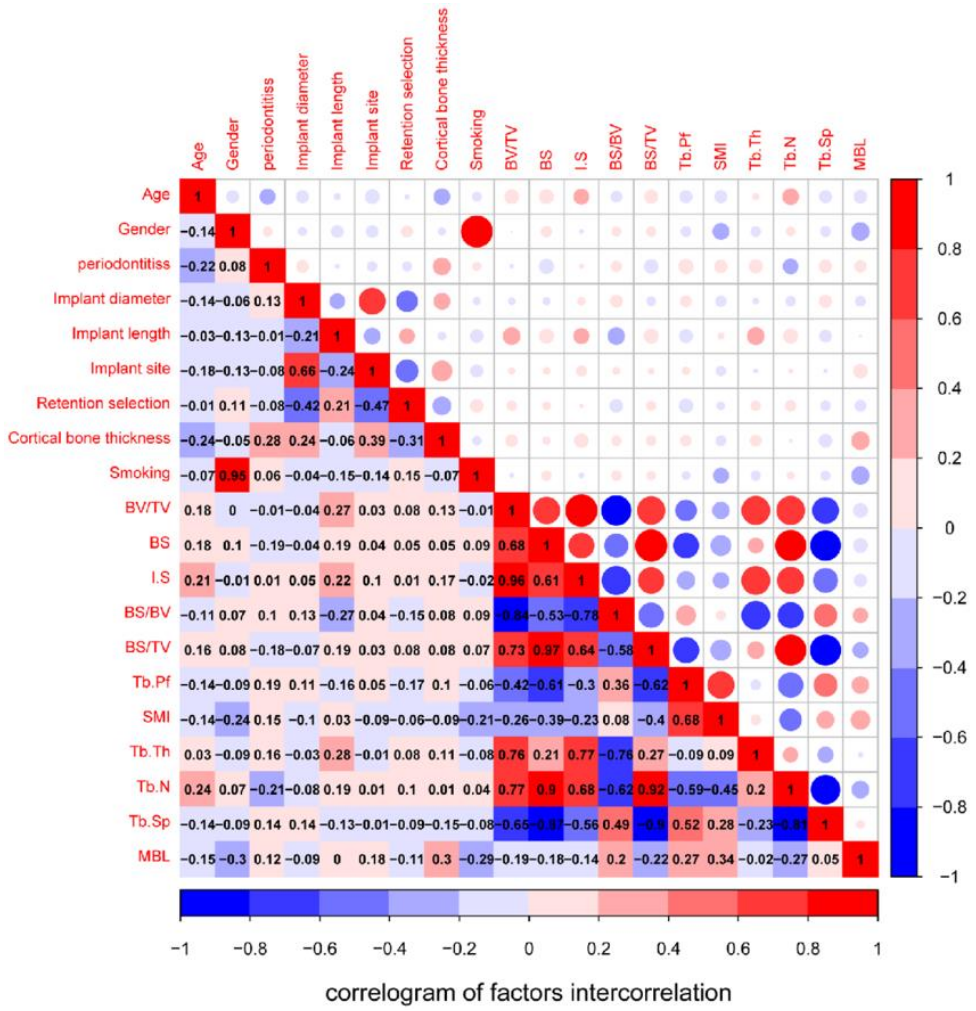
Covariance Matrix

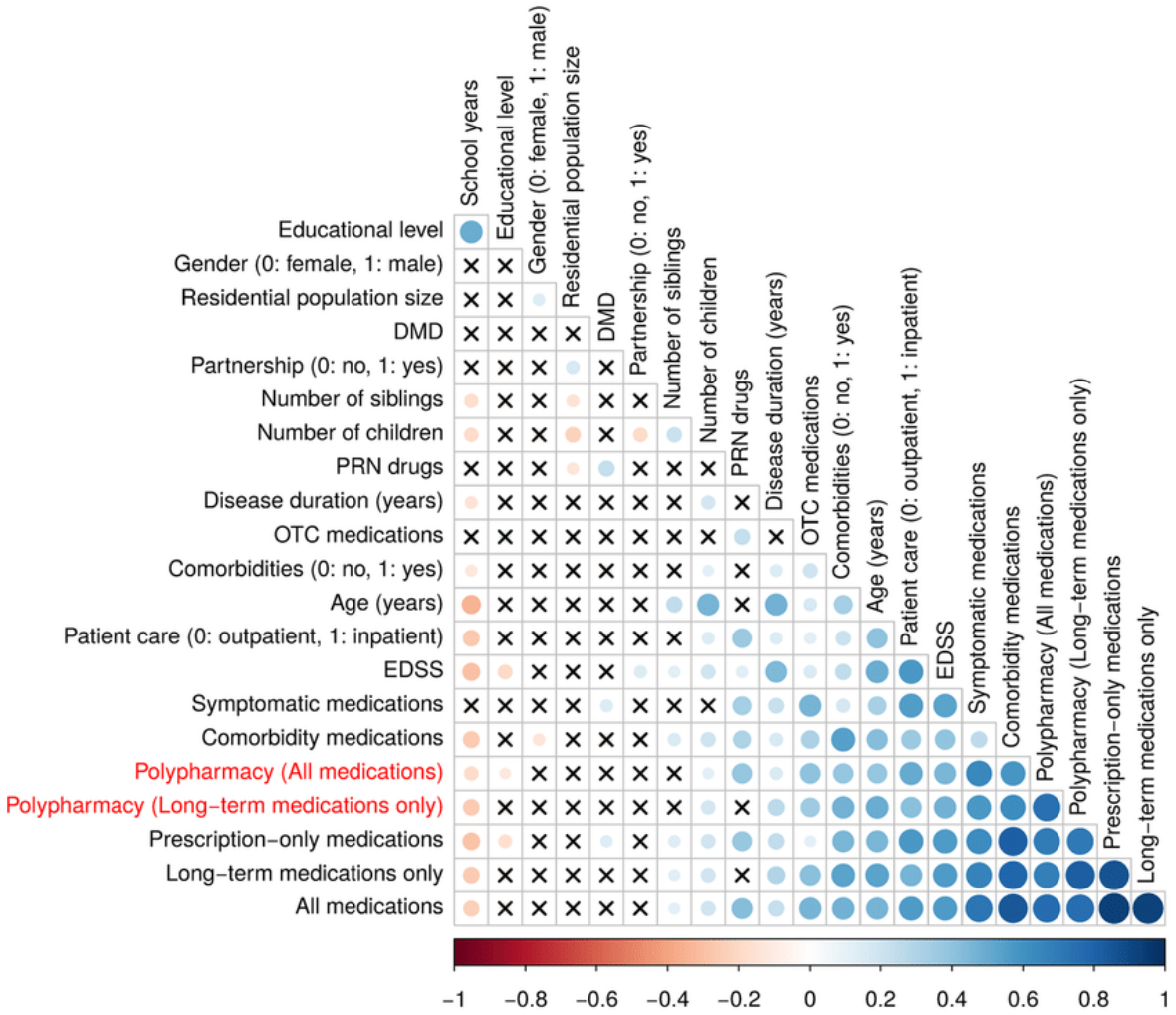
- The covariance matrix is a square matrix that **displays the variance exhibited by variables of datasets and the covariance between pairs of variables**

- **Example for 3 variables:**

$$\mathbf{\Sigma} = \begin{pmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1, X_2) & \mathrm{Cov}(X_1, X_3) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Var}(X_2) & \mathrm{Cov}(X_2, X_3) \\ \mathrm{Cov}(X_3, X_1) & \mathrm{Cov}(X_3, X_2) & \mathrm{Var}(X_3) \end{pmatrix}$$

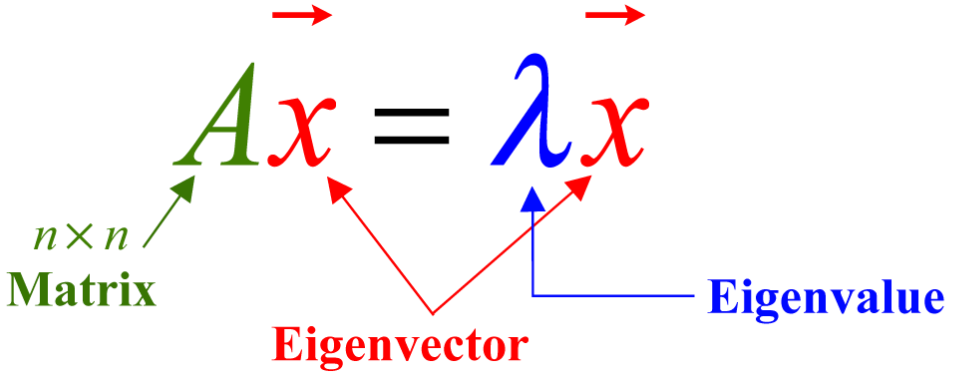
Similar to a Correlation Matrix





Source: https://www.researchgate.net/figure/Correlation-matrix-visualization-of-the-correlation-between-variables-and-polypharmacy_fig3_331551476; https://images.app.goo.gl/CDxct5DeRK9kTsRv7

Eigenvalues and Eigenvectors of a square matrix in PCA



Eigenvectors

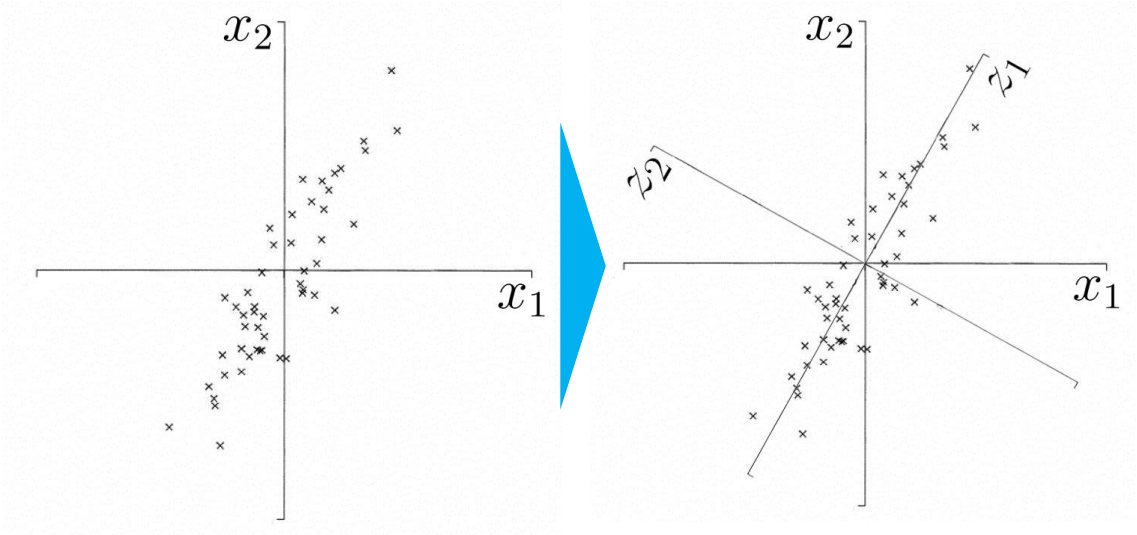
Think of eigenvectors as directions or axes in space. When you apply PCA, you are looking for new axes (directions) that better describe the spread of your data. These new axes (eigenvectors) are the most important directions where your data varies the most. I.e. eigenvectors help you find the main "themes" or "patterns" in your data.

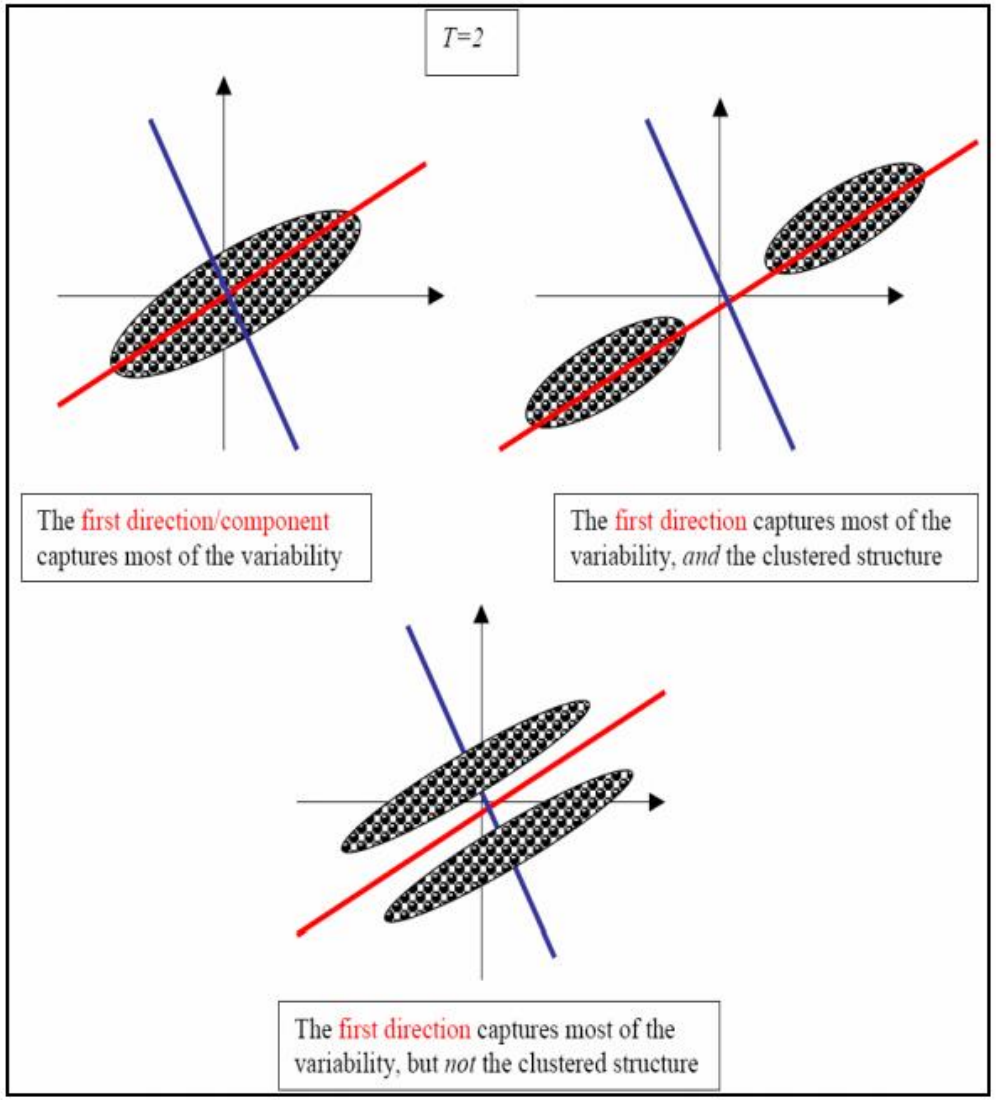
Eigenvalues

Now, for each of these new directions (eigenvectors), you have an eigenvalue. The eigenvalue tells you how important each direction is. A higher eigenvalue means that direction explains more of the variability in your data. Think of eigenvalues as the "weight" or "importance" of each eigenvector. The bigger the eigenvalue, the more significant the pattern or theme is in your data.

Source: calcworkshop.com

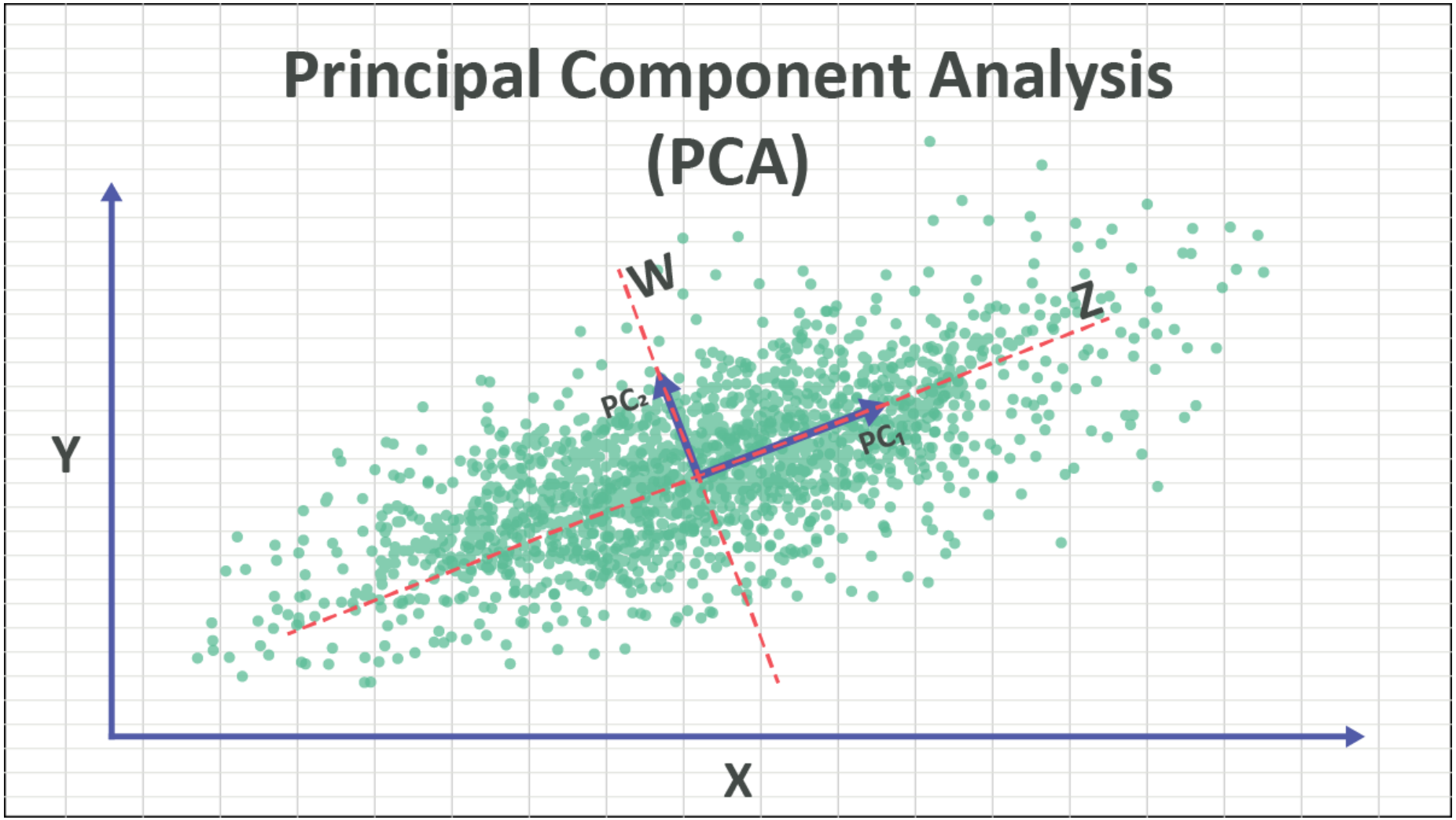
Main directions of variation





Sources: http://sites.stat.psu.edu/~chiaro/BioinfoII_08/; http://www.public.asu.edu/~jye02

For two linearly associated variables, 2 directions are enough



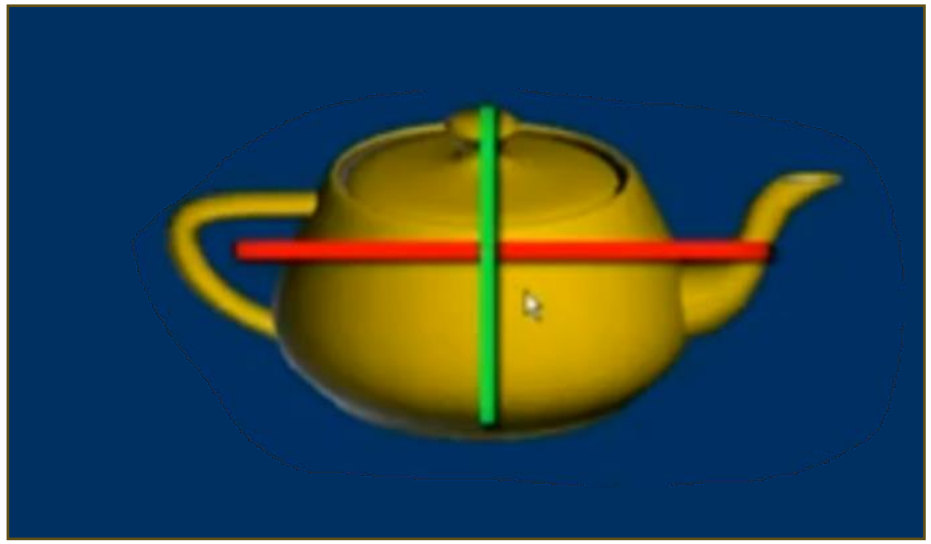
Source: https://numxl.com/blogs/principal-component-analysis-pca-101/

Visually

Rotate the object around its centre to find the best orientation:

- First find the axis so that the object has the largest extend on average along the axis

- Rotate the object around the first axis to find the axis that is perpendicular to the first axis, and the object has the largest extent on average along this axis



The two axes found are the **first** and the **second** principal components; The **extent in average** along the axes is called the **eigenvalues**.

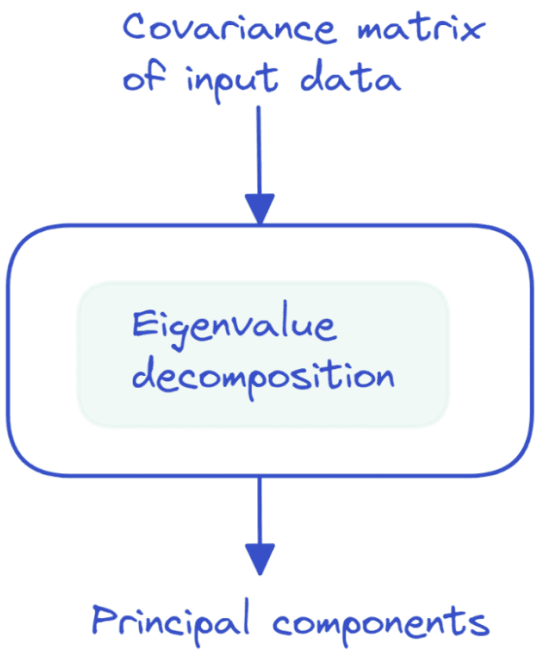
Reconstruction

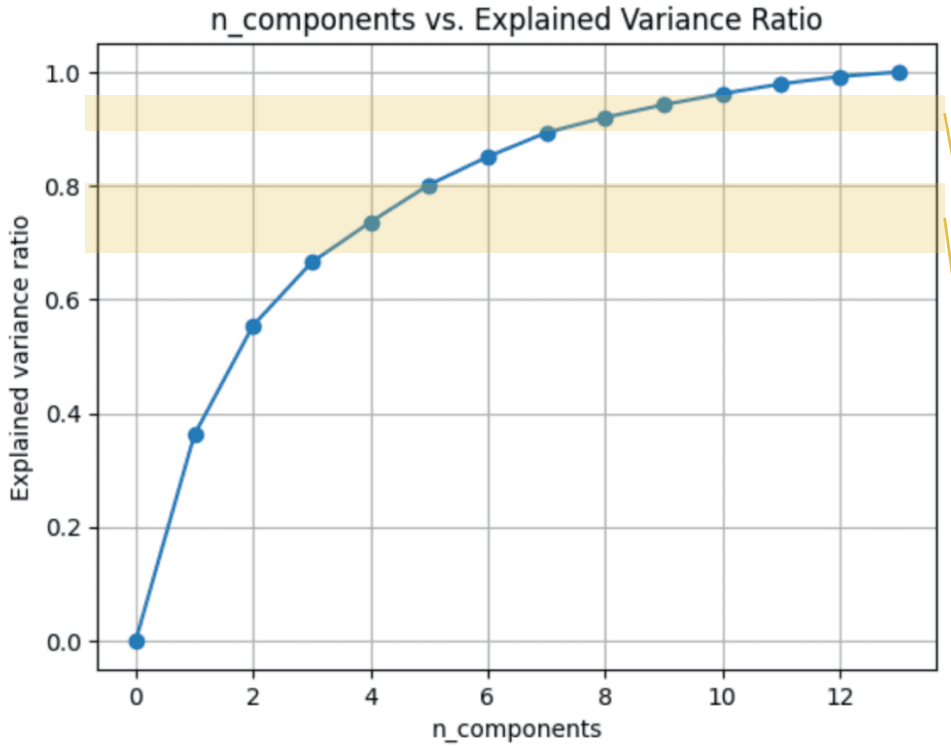
```
(525, 700)
1.0
<matplotlib.image.AxesImage at 0x7efccbccc610>
```

- When **adding back the components, multiplied by their eigenvalues, we have a reconstruction of the original data**. This reconstruction is an ***approximation*** that retains the most significant patterns captured by the principal components.
- The more components, the better – but not too much to avoid defeating the purpose!

Source: https://analyticsindiamag.com/guide-to-image-reconstruction-using-principal-component-analysis/

How many components? Similar to Elbow…





Principal Component Analysis in Scikit-Learn

Elbow Point: The "elbow point," where the slope of the plot sharply changes, is often chosen as the cutoff. This point typically indicates where additional components contribute less to explaining the variability.

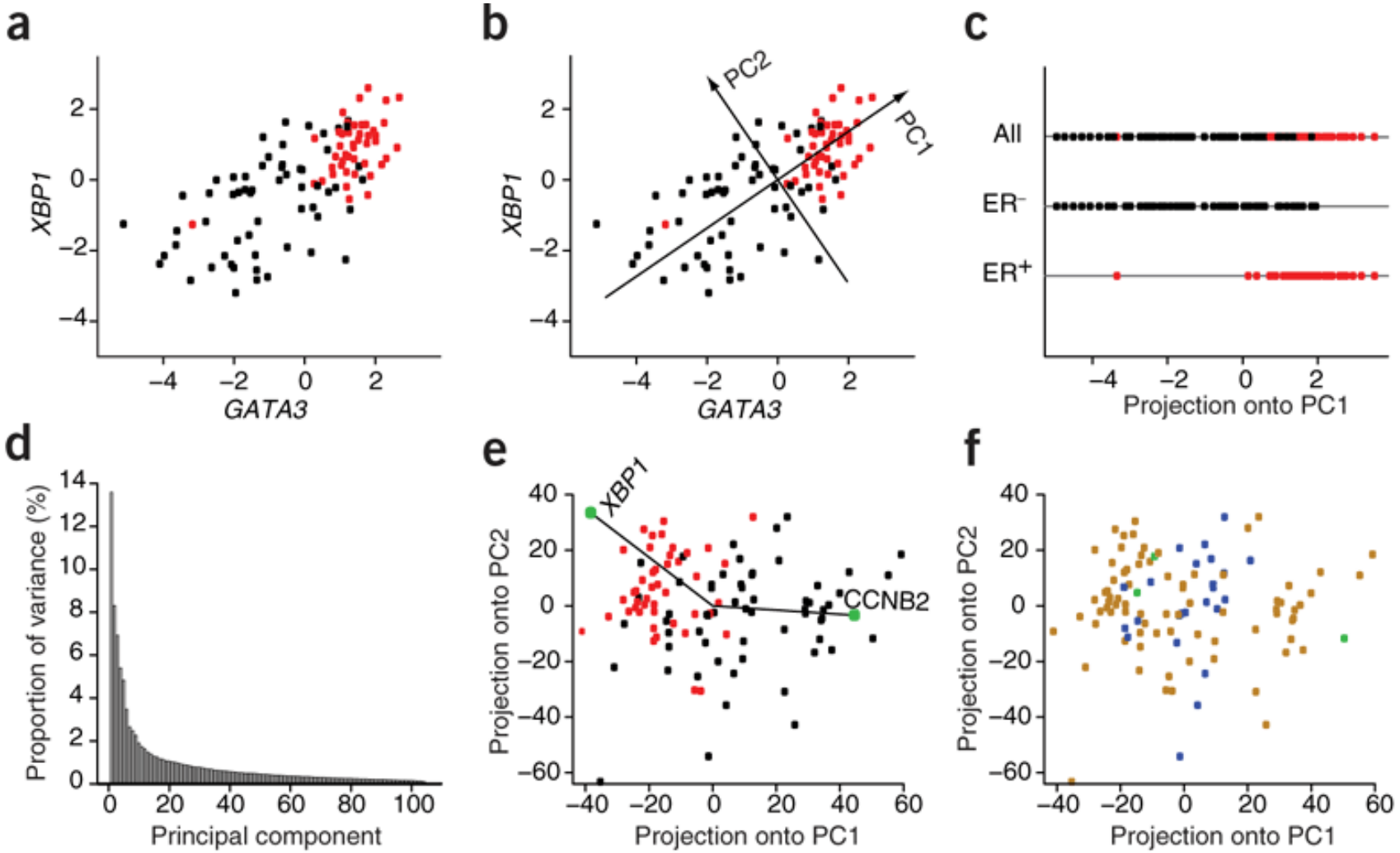
OR

Cumulative Explained Variance:

- **90-95%:** For applications requiring higher precision, such as certain scientific analyses, 90-95% of variability might be retained.
- **70-80%**: For exploratory data analysis where a more substantial dimensionality reduction is needed, 70-80% might be acceptable.

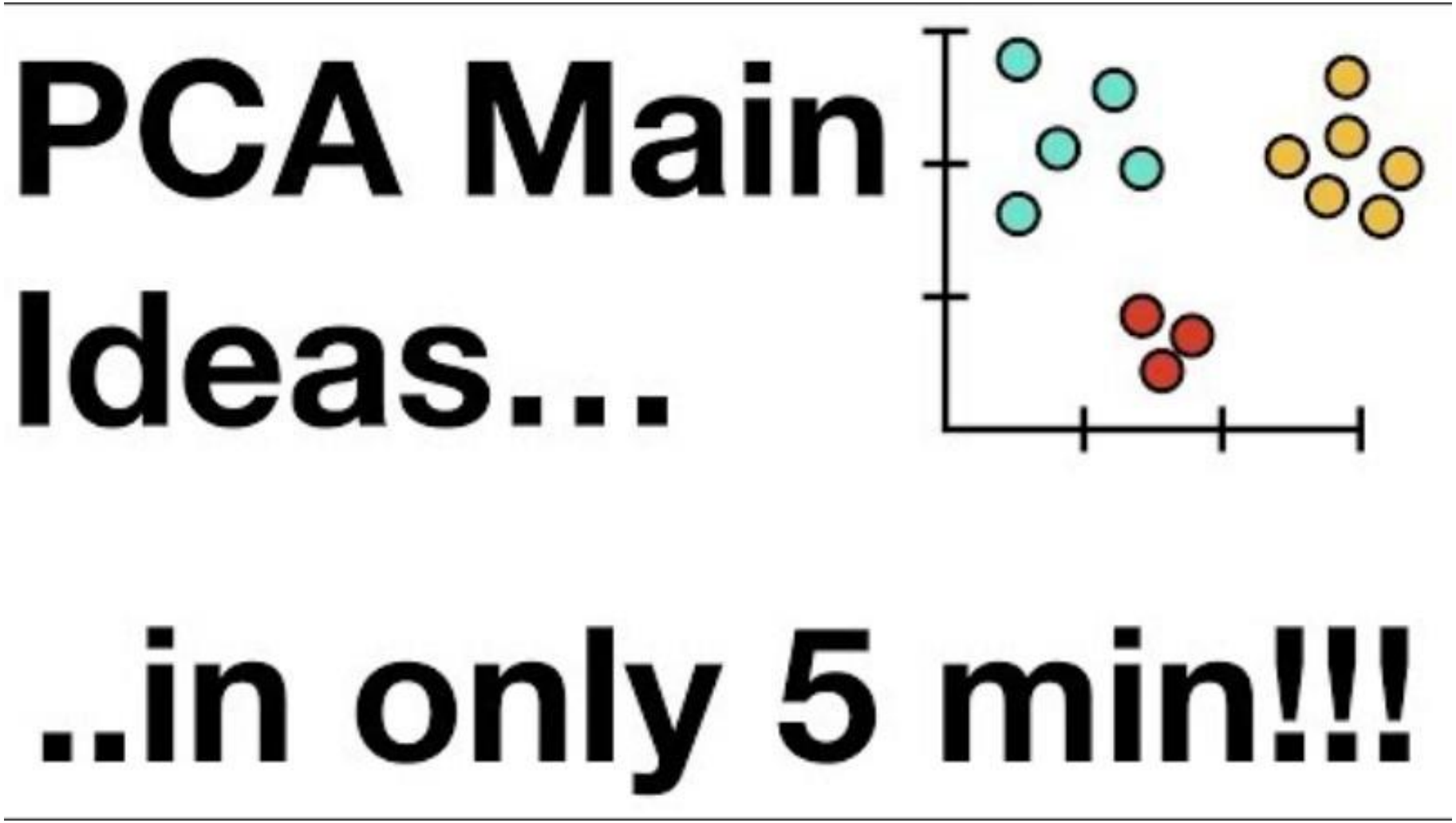
Source: https://www.kdnuggets.com/2023/05/principal-component-analysis-pca-scikitlearn.html

Example, as seen in a scientific paper



Source: https://www.nature.com/articles/nbt0308-303

In summary



Source: https://www.youtube.com/watch?v=HMOI_lkzW08

Summary – PCA

What It Does	Reduces dimensionality by transforming data to a new coordinate system
Main Goal	Maximize variance and minimize information loss
Mathematical Basis	Uses covariance matrix, eigenvalues, and eigenvectors
Output Components	Principal components (orthogonal vectors)
Applications	Image compression, feature reduction, exploratory data analysis

Pros	- Easy to understand and implement
	- Effective for reducing dimensionality
	- Preserves most of the variance
Cons	- Assumes linearity and orthogonality of components
	- Sensitive to scaling of data
	- Only captures second-order statistics (covariance)

Thank you!

João Pereira
jmspereira@ucp.pt

Jorge Cerejo
jorge.filipe.cerejo@luzsaude.pt

GPT4o
https://chat.openai.com/