Università degli Studi di Padova

Department of Industrial Engineering

Machine Learning
Lesson #13

Academic year 2025-2026

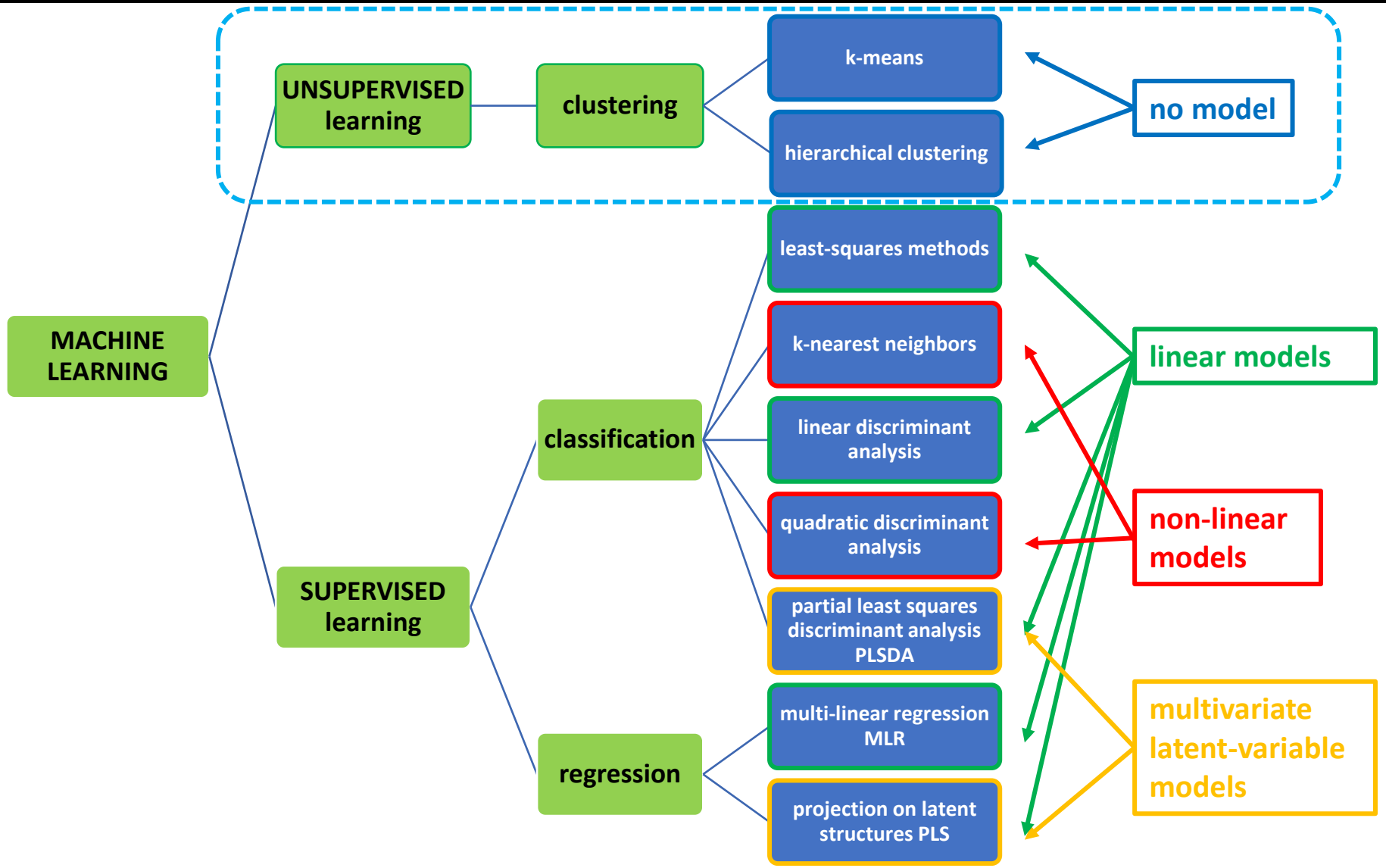
Prof. Pierantonio Facco

CAPE-Lab, Computer-Aided Process Engineering Laboratory

Email: pierantonio.facco@unipd.it

URL: https://research.dii.unipd.it/capelab/

Contextualization



Unsupervised learning

Unsupervised learning

- The goal of **unsupervised learning** (or “learning without a teacher”) is to directly infer, without the help of any supervisor, the properties of the probability density $P(\mathbf{X})$ for a set of N observations inputs $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$ where $\mathbf{x}_n$ is a random vector of V variables/features
  - in this case a “teacher” providing correct answers or degree-of-error for each observation is missing
  - the dimension of $\mathbf{X}$ is sometimes much higher than in supervised learning
  - the properties of interest are often more complicated than simple location estimates
  - these factors are somewhat mitigated by the fact that $\mathbf{X}$ represents all the variables under consideration

Pros and cons of unsupervised learning

- The good point in **unsupervised** learning is that *it is not required to infer how the properties of* $\mathrm{P}(\mathbf{X})$ *change, conditioned on the changing values of another set of variables* $\mathbf{Y}$ (i.e., the class labels)

- The bad point is that with unsupervised learning there is **not a direct measure of success** that can be used to judge model adequacy and to compare the effectiveness of different methods over various situations/models
  - this uncomfortable situation has led to heavy proliferation of methods, since effectiveness is a matter of opinion and cannot be verified

Cluster analysis

Cluster analysis

- **Cluster analysis** (a.k.a. **data segmentation**) goals relate to grouping or segmenting a collection of objects (i.e., observations) into subsets or “clusters”
  - objects within each cluster are more closely related to one another than objects assigned to different clusters
  - an object is a set of measurements that can be described by its relation to other objects
  - a clustering method groups the objects based on a definition of **similarity**
    - the notion of the **degree of similarity/dissimilarity** between the individual objects being clustered is central
    - the situation is somewhat similar to the specification of a loss or cost function in supervised learning

- An additional goal is sometimes to arrange the clusters into a natural **hierarchy**
  - this involves successively grouping the clusters themselves
  - at each level of the hierarchy, clusters within the same group are more similar to each other than those in different groups

- Cluster analysis is also used for **descriptive statistics** to ascertain whether the data consists of a set of distinct subgroups, each group representing objects with substantially different properties
  - this goal requires an assessment of the degree of difference between the objects assigned to the respective clusters

Discussion

- How would you define similarity or dissimilarity?

Similarity and dissimilarity measurements

- Sometimes the data are represented directly in terms of the **proximity** (*alikeness* or *affinity*) between pairs of objects
  - these can be **either similarities or dissimilarities** (difference or lack of affinity)
  - the most popular clustering algorithms take a dissimilarity matrix as their input

- **Pairwise dissimilarities** between the observations are usually calculated

- The most common measures of dissimilarity are:
  - **object dissimilarity**: for couple of observations n_1 and n_2 of V quantitative variables
    - **Euclidean distance**

$$d_{n_1,n_2} = \|\mathbf{x}_{n_1} - \mathbf{x}_{n_2}\| = \sqrt{\sum_{v=1}^{V} \left(x_{n_1,v} - x_{n_2,v}\right)^2}$$

    - **correlation**
  - **overall dissimilarity**: averaging the object dissimilarity

$$D = \frac{1}{N^2} \sum_{n=1}^{N} \sum_{n=1}^{N} d_{n_1,n_2}$$

Prototype methods and nearest neighbors

- Prototype and nearest-neighbor methods are simple and essentially **model-free methodologies** for pattern recognition and classification:
  - highly unstructured
  - *not useful for understanding the nature of the relationship between the features* and class outcome
  - **black-box** prediction engines
  - can be very effective, and are often among the best performers in real data problems

- The nearest-neighbor technique can also be used in regression:
  - works reasonably well for low-dimensional problems
  - with high-dimensional features, the bias–variance tradeoff does not work as favorably for nearest-neighbor regression as it does for classification

Prototype methods

- **Prototype methods** represent the training data by a set of points (prototypes) in the feature space
  - these prototypes are typically not samples from the training sample, except in the case of 1-nearest-neighbor classification
- Each prototype has an associated cluster/class label
- Classification of a query point $\mathbf{x}$ is made to the class of the closest prototype
  - the "closest" prototype is usually defined by **Euclidean distance** in the feature space
    - Euclidean distance is appropriate for quantitative features
  - each feature is typically **standardized** to have overall mean 0 and variance 1 in the training sample
- These methods can be very effective if the prototypes are well positioned to capture the distribution of each class
  - **irregular class boundaries** can be represented, with enough prototypes in the right places in feature space
  - methods differ according to the number and way in which prototypes are selected
    - the main challenge is to figure out **how many prototypes** to use and **where** to put them

k-means clustering

K-means clustering

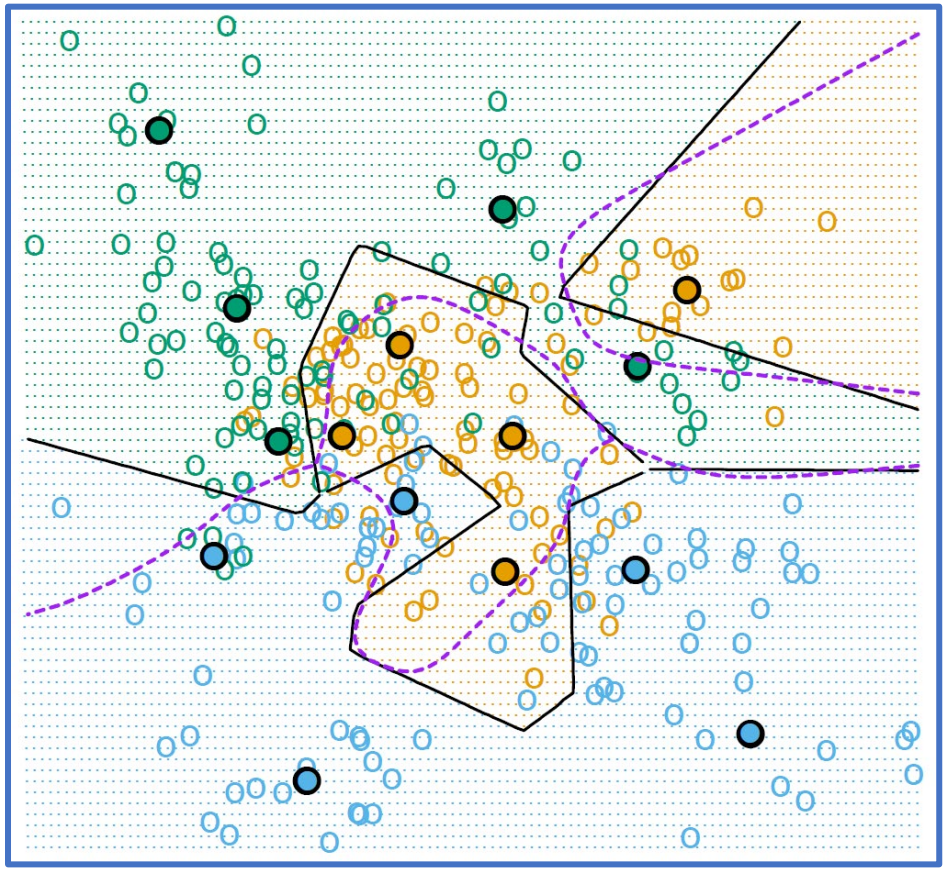
- ***K*-means clustering** is a **prototype clustering method** for finding clusters and cluster centers in a set of unlabeled data
  - the **desired number K of clusters** is selected a priori
  - the K-means procedure iteratively moves the centers of the clusters to minimize the total within cluster variance

- Given an initial set of centers, the **K-means algorithm** alternates the two steps (iterated until convergence):
  1. for each center we identify the subset of training points (its cluster) that is closer to it than any other center
  2. the means of each feature for the data points in each cluster are computed, and this mean vector becomes the new center for that cluster

K-means clustering

- Initialization:
  - the initial centers are K randomly chosen observations from the training data

- Notice that a number of the prototypes are near the class boundaries, leading to potential misclassification errors for points which are close to these boundaries
  - for each class, the other classes do not determine in any way the positioning of the prototypes for that class
  - a better approach may use all the data to position all prototypes



K-means clustering algorithm

- From the formal mathematical point of view the *K*-means algorithm is one of the most popular iterative descent clustering methods
  - intended for situations in which all **variables are quantitative**
  - **squared Euclidean distance** is chosen as the dissimilarity measure
    - weighted Euclidean distance (or other dissimilarity measurements) can be used, as well
- The criterion is minimized by assigning the N observations $\mathbf{x}_i$ to the K clusters in such a way that **within each cluster the average dissimilarity of the observations from the cluster mean, as defined by the points in that cluster, is minimized**
- This can be minimized by an alternating optimization procedure given in the following algorithm (iterated until convergence):
  1. for a given cluster assignment C, the total cluster variance:
     $$C = \min_{C} \sum_{k=1}^{K} N_K \sum_{C(i)=k} \|\mathbf{x}_i - \mathbf{m}_K\|^2$$
     is minimized with respect to $[\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_K]$ yielding the means of the currently assigned clusters
  2. given a current set of means $[\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_K]$, C is minimized by assigning each observation to the closest cluster mean:
     $$C(i) = \min_{1 \le k \le K} \|\mathbf{x}_i - \mathbf{m}_K\|^2$$

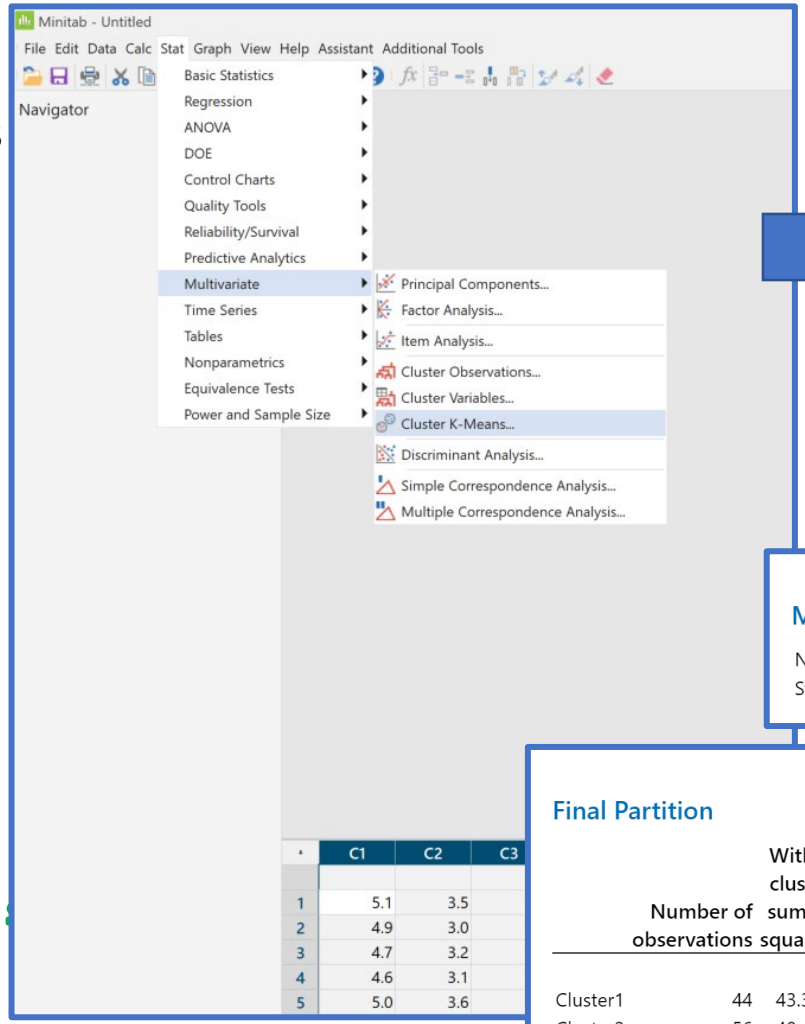
  where:
  - $C(i)$ is the encoder that assigns the observation i to the cluster k
  - N_K is the number of observations assigned to the k-th cluster

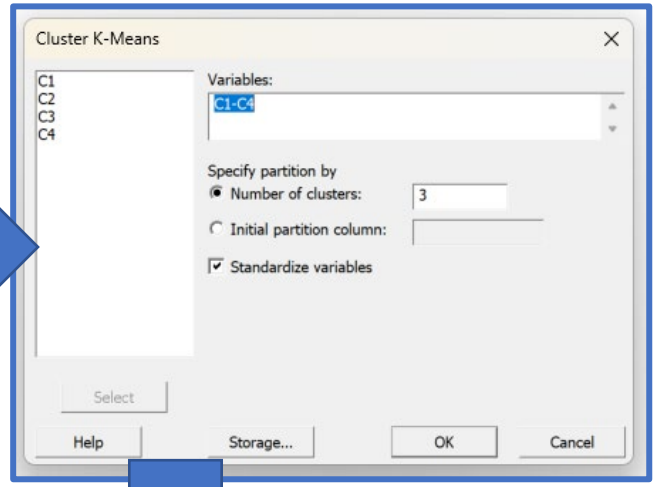
K-means clustering in practice

- Iterative procedure to determine clusters

Example of *k*-means: Fisher's iris

- Iris Flower dataset:
  - 4 variables:
    - length and width of petals and sepals
  - 3 varieties:
    - setosa: observations 1-50
    - versicolor: observations 51-100
    - virginica: observations 101-150
- In Matlab®:
  - `c=kmeans(fisheriris,3);`
- In Minitab®:
  - **Stat**
  - **Multivariate**
  - **Cluster K-Means**
  - select variables
  - **Number of clusters: 3**
  - select **Standardize variable**
  - click OK





Method

Number of clusters	3
Standardized variables	Yes

Final Partition

	Number of observations	Within cluster sum of squares	Average distance from centroid	Maximum distance from centroid
Cluster1	44	43.347	0.898	2.103
Cluster2	56	48.402	0.872	2.006
Cluster3	50	47.351	0.811	2.662

Cluster Centroids

Variable	Cluster1	Cluster2	Cluster3	Grand centroid
C1	1.1635	-0.0114	-1.0112	0.0000
C2	0.1448	-0.8731	0.8504	0.0000
C3	0.9997	0.3758	-1.3006	0.0000
C4	1.0266	0.3101	-1.2507	0.0000

Distances Between Cluster Centroids

	Cluster1	Cluster2	Cluster3
Cluster1	0.0000	1.8218	3.9629
Cluster2	1.8218	0.0000	3.0359
Cluster3	3.9629	3.0359	0.0000

Hierarchical clustering

Hierarchical clustering

- **Hierarchical clustering** is an unsupervised clustering methodology to identify samples with *similar features*:
  - data are organized in a **nested sequence of classes levels according to a linkage rule which measures pairwise observations similarity**

- Hierarchical clustering represents a particular grouping of the data into disjoint clusters of observations divided in different levels of a **hierarchy**
  - the hierarchy represents an ordered sequence of such groupings
  - the user decides which level (if any) actually represents a “natural” clustering of the observations
  - it provides the answer to the question: which observations within each of its groups are sufficiently more similar to each other than to observations assigned to different groups at that level?
  - it produces a hierarchical representations in which the clusters at each level of the hierarchy are created by merging clusters at the next lower level
    - at the lowest level, each cluster contains a single observation
    - at the highest level there is only one cluster containing all the data

Hierarchical clustering strategies and methods

- Strategies for hierarchical clustering are divided into two **basic paradigms** with $(N-1)$ levels in the hierarchy:
  - **agglomerative** (bottom-up)
    - start at the bottom
    - recursively merge a selected pair of clusters into a single cluster at each level
    - the pair chosen for merging consist of the two groups with the smallest intergroup dissimilarity
    - the grouping at the higher level has one less cluster
  - **divisive** (top-down)
    - start at the top
    - recursively split one of the existing clusters at that level into two new clusters at each level
    - the split is chosen to produce two new groups with the largest between-group dissimilarity

- Hierarchical clustering methods:
  - *do not require the a-priori selection of the **number of clusters**, nor a starting configuration assignment*
  - need:
    - a **measure of dissimilarity** between (disjoint) groups of observations
      - pairwise dissimilarities among the observations in two groups
    - a **linkage rule**
      - criterion to specify the distance between clusters as a function of the distance among observations

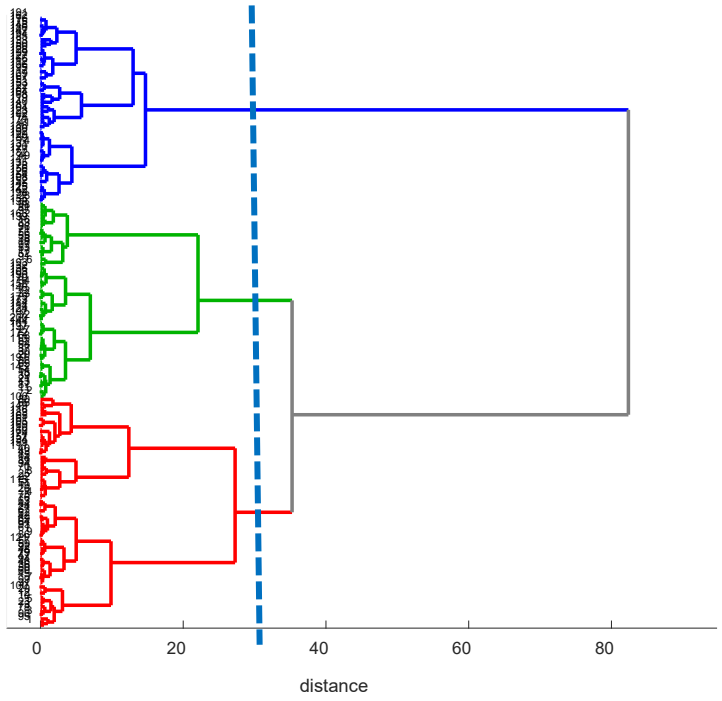
Hierarchical clustering representation

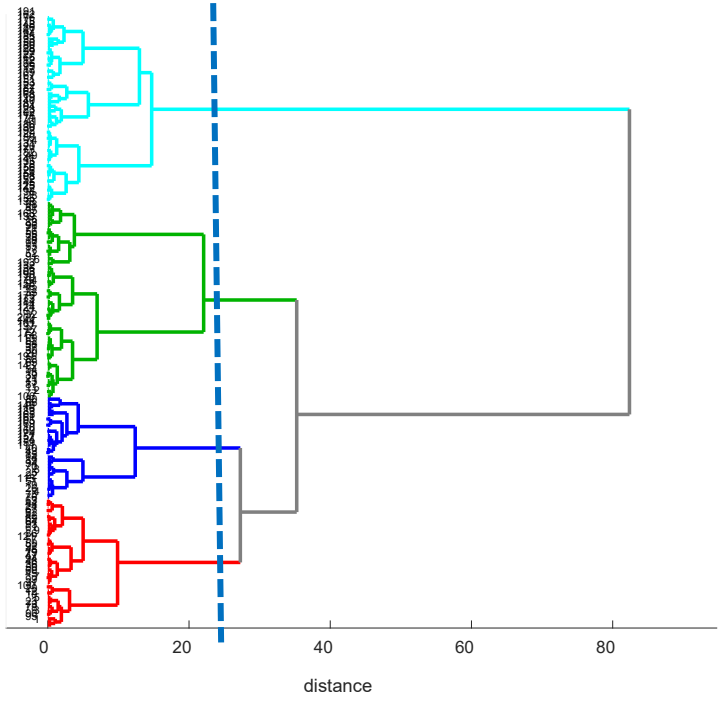
- Recursive binary splitting/agglomeration can be represented by a **rooted binary tree**:
  - the nodes of the trees represent groups
  - the root node represents the entire data set
  - the N terminal nodes each represent one individual observation (singleton clusters)
  - each nonterminal node (“parent”) has two daughter nodes
    - for divisive clustering the two daughters represent the two groups resulting from the split of the parent
    - for agglomerative clustering the daughters represent the two groups merged to form the parent

- All agglomerative and some divisive methods possess a **monotonicity** property
  - the dissimilarity between merged clusters is monotone increasing with the level of the merger
  - the binary tree can be plotted so that the height of each node is proportional to the value of the intergroup dissimilarity between its two daughters
  - the terminal nodes representing individual observations are all plotted at zero height

Dendrogram

- The graphical display is called a **dendrogram** (a graphical summary of the data)
  - refer to the Matlab® commands: `cluster; linkage; dendrogram; cophenet`
- Pros and cons:
  - cutting the dendrogram at a particular distance partitions the data into disjoint clusters
  - different hierarchical methods, as well as small changes in the data, can lead to quite different dendrograms
  - it is valid only to the extent that the pairwise observation dissimilarities possess the hierarchical structure





Cophenetic coefficient

- The extent to which the hierarchical structure produced by a dendrogram actually represents the data itself is judged by the **cophenetic correlation coefficient**:
  - a correlation between:
    - the $N(N-1)/2$ pairwise observation dissimilarities input to the algorithm
    - their corresponding cophenetic dissimilarities derived from the dendrogram
      - the cophenetic dissimilarity between two observations is the inter-group dissimilarity at which observations n_1 and n_2 are first joined together in the same cluster

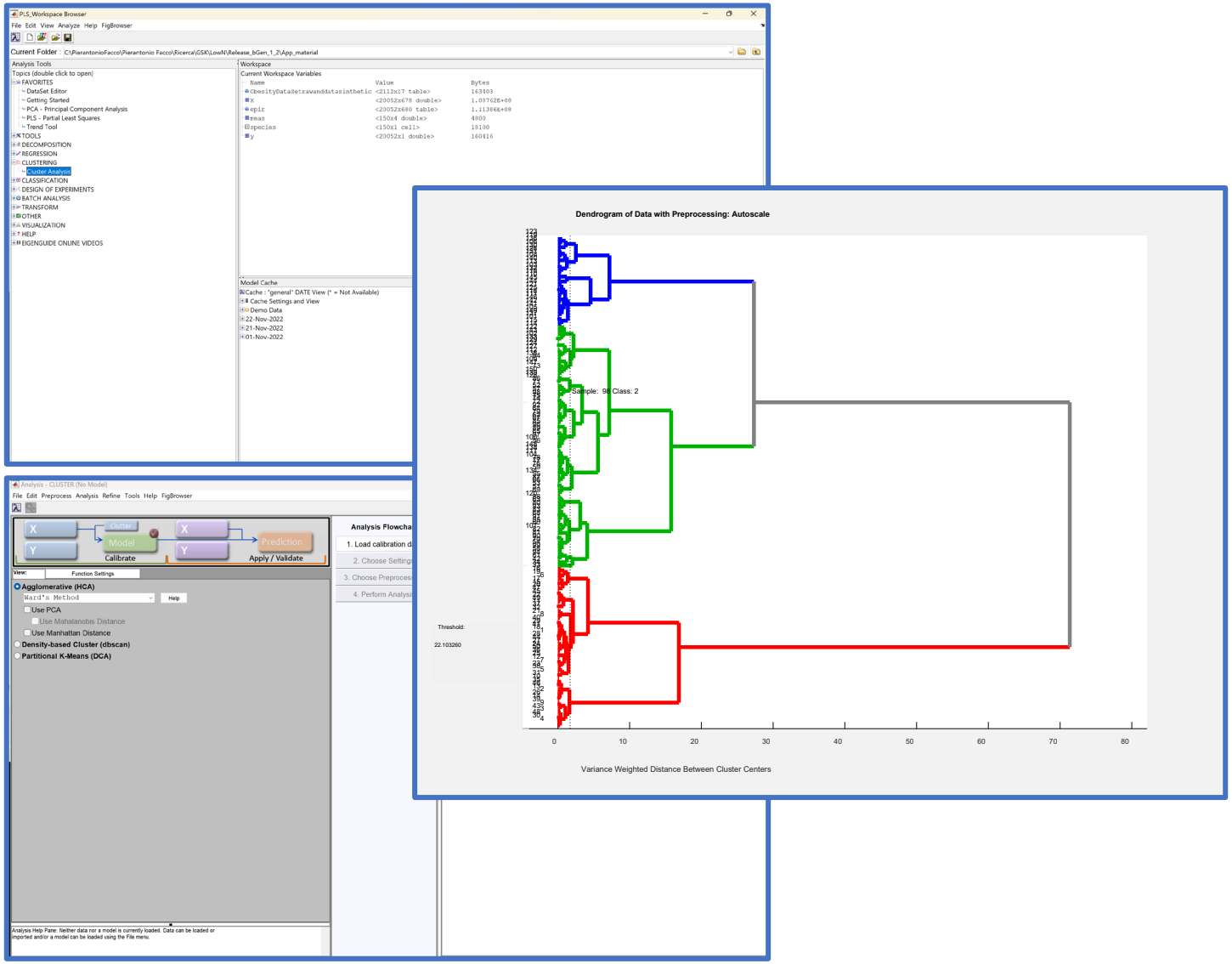
$$c_h = \frac{\sum_{n_2>n_1}\left(d_{n1,n2}-\bar{d}\right)\left(c_{n1,n2}-\bar{c}\right)}{\sqrt{\sum_{n_2>n_1}\left(d_{n1,n2}-\bar{d}\right)^2 \sum_{n_2>n_1}\left(c_{n1,n2}-\bar{c}\right)^2}}$$

  - where:
    - $d_{n1,n2}$ is the Euclidean distance among observations n_1 and n_2
    - $\bar{d}$ is the overall average of all the Euclidean distances
    - $c_{n1,n2}$ is the cophenetic distance among observations n_1 and n_2
      - the cophenetic distance is the distance among points in the dendrogram
    - $\bar{c}$ is the overall average of all the cophenetic distances

- If c_h is **close to 1**, the cluster algorithm is providing a faithful representation of the samples within-clusters similarities compared to the original distance matrix
  - a good algorithm should preserve the original information about the similarity between samples, and this means that the cophenetic coefficient should be high
  - however, this does not imply that the method with the largest coefficient is the most well-performing

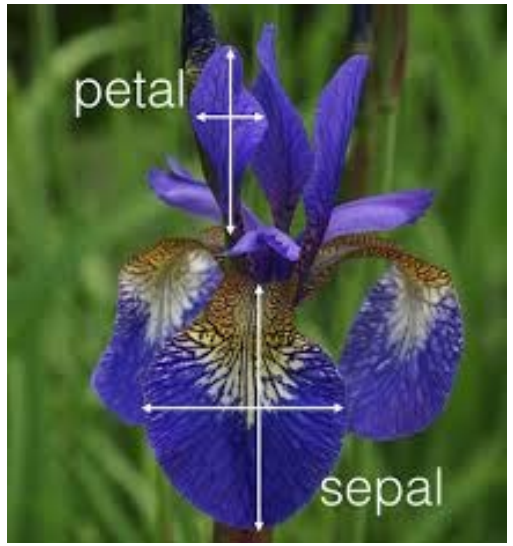
Example of hierarchical clustering: Fisher's iris

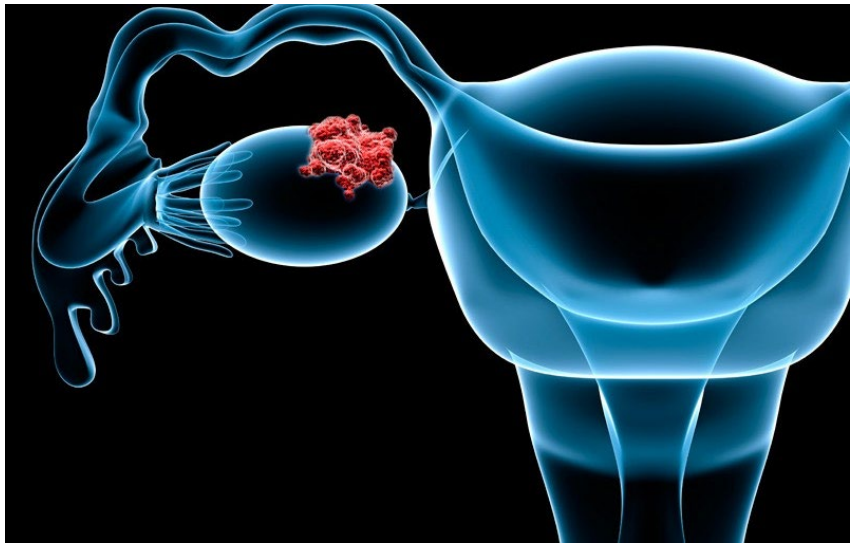
- In PLS_Toolbox®:
  - **Browse**
  - select **Clustering**
  - select **Cluster analysis**
  - **File**
  - **Load data**
  - **X-Block**
  - Select **meas**
  - Click OK
  - Click **Perform cluster analysis** button
  - Click to select the threshold in the dendrogram



Today homework

- Practice with Matlab® with the following 2 dataset:
  - **`fisheriris`**
    - 150 observations for 4 variables (petals and sepals measurements)
  - **`ovariancancer`**
    - 216 observations for 4000 variables from a determined analytical technology





… per sempre a fianco a me!