

SELECTED TOPICS IN ENGINEERING

INTR. TO PROG. FOR DATA SCIENCE
ENGR 350

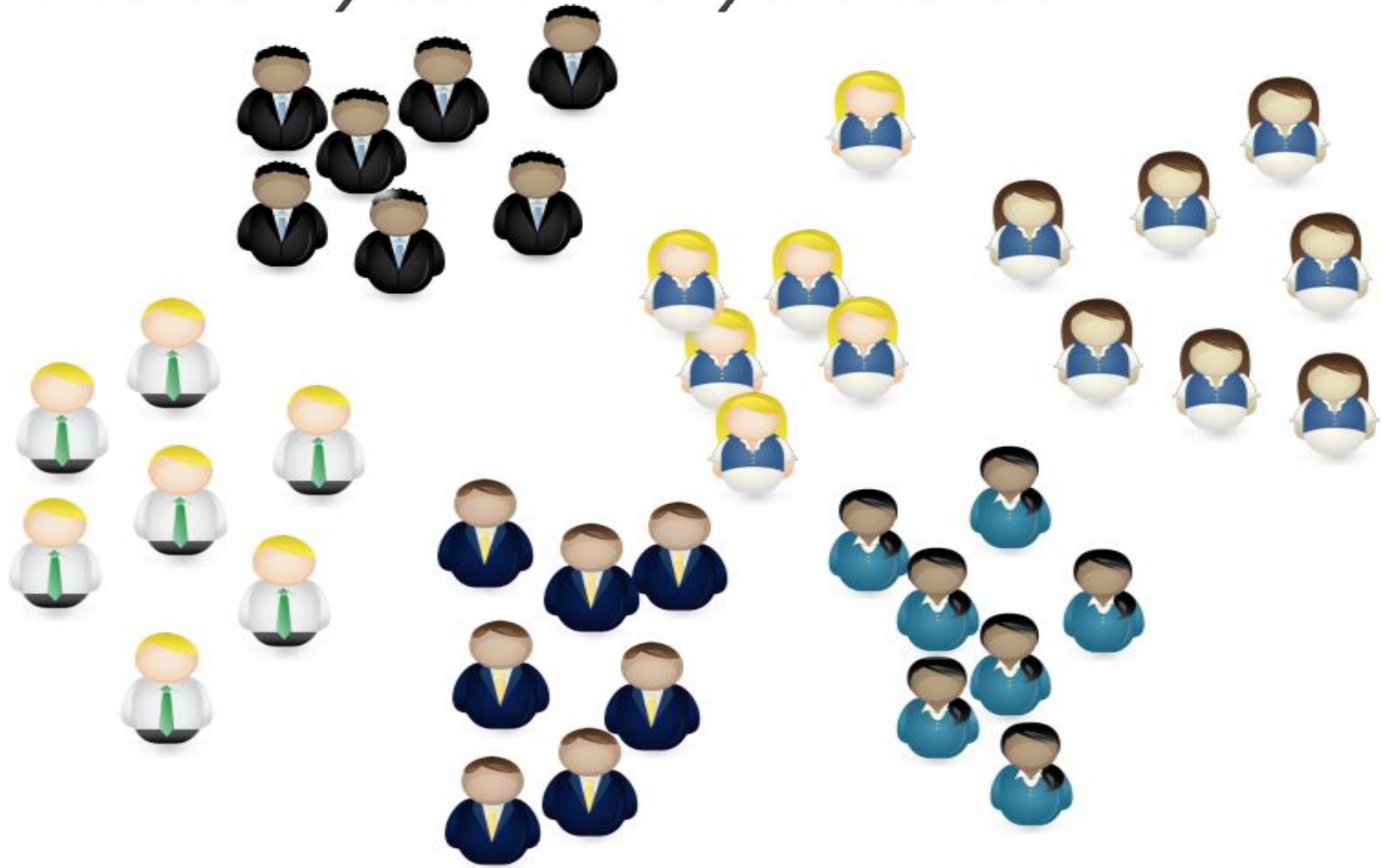
Tuesday–Thursday 10:00–12:45

ENG B05

2019 Summer

Dr. Banu Yobaş

*"Show me who your friends are
and I'll tell you who you are?"*



Clustering vs Classification

human beings are skilled at

- ▶ dividing objects into groups (clustering) and
- ▶ assigning particular objects to these groups (classification).

Used for:

- ▶ understanding or
- ▶ utility




Clustering

- ▶ Cluster analysis groups data objects based on information found only in the data that describes the objects and their relationships.
- ▶ The goal is that the objects within a group be similar (or related) to one another and different from (or unrelated to) the objects in other groups.
- ▶ The greater the similarity (or homogeneity) within a group and the greater the difference between groups, the better or more distinct the clustering.

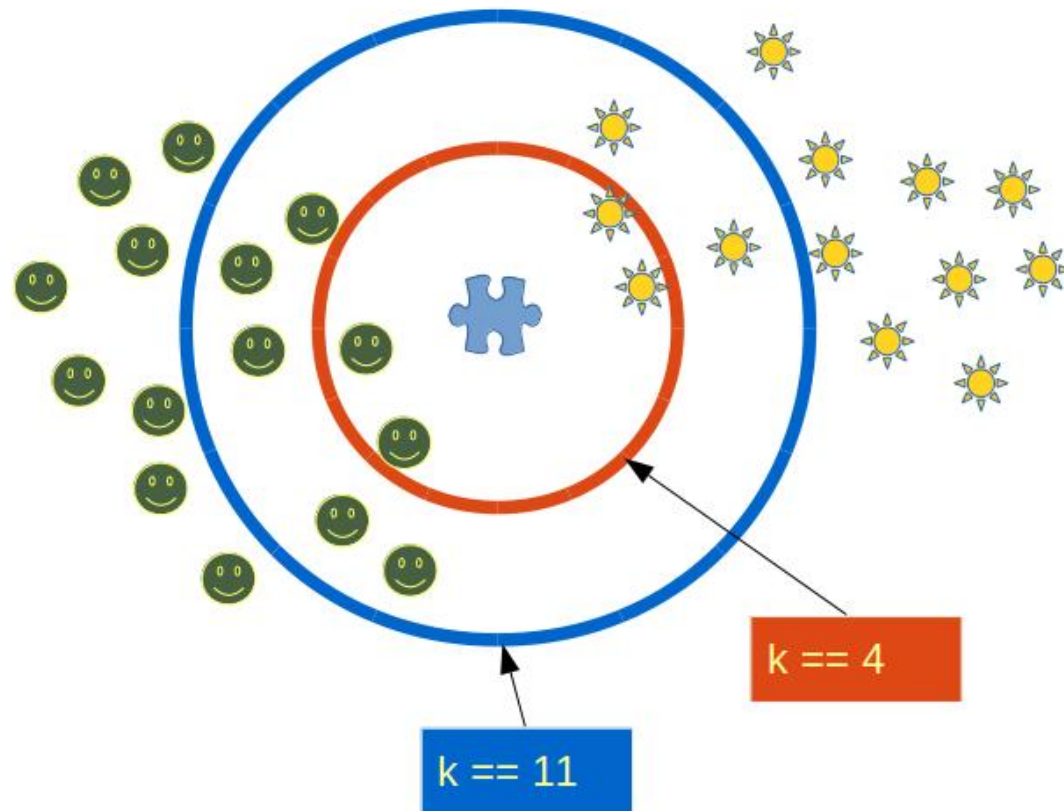


Cluster analysis

- ▶ provides an **abstraction** from individual data objects to the clusters in which those data objects reside.
 - ▶ some clustering techniques characterize each cluster in terms of a **cluster prototype**; i.e., a data object that is representative of the objects in the cluster.
 - ▶ Many data analysis techniques, such as regression or principal component analysis (PCA), have a time or space complexity of $O(m^2)$ or higher (where m is the number of objects), and thus, are not practical for large data sets.
 - ▶ Many times, cluster analysis is conducted as a part of an exploratory data analysis
- 

K Nearest

🧩 == 😊 or 🧩 == ☀️ ?



Clustering types

- ▶ Hierarchical versus Partitional:
- ▶ A **partitional clustering** is simply a division of the set of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
- ▶ If we permit clusters to have subclusters, then we obtain a **hierarchical clustering**, which is a set of nested clusters that are organized as a tree.



Clustering types

- ▶ Exclusive versus Overlapping versus Fuzzy:
- ▶ assign each object to a single cluster.
- ▶ There are many situations in which a point could reasonably be placed in more than one cluster
- ▶ an overlapping or non-exclusive clustering is used to reflect the fact that an object can *simultaneously* belong to more than one group (class)



Fuzzy clustering

- ▶ every object belongs to every cluster with a membership weight that is between 0 (absolutely doesn't belong) and 1 (absolutely belongs).
- ▶ clusters are treated as fuzzy sets.
- ▶ Mathematically, a fuzzy set is one in which an object belongs to every set with a weight that is between 0 and 1. In fuzzy clustering, we often impose the additional constraint that the sum of the weights for each object must equal 1.
- ▶ Because the membership weights or probabilities for any object sum to 1, a fuzzy or probabilistic clustering does not address true multiclass situations



Complete versus Partial

- ▶ A complete clustering assigns every object to a cluster, whereas
- ▶ a partial clustering does not.
- ▶ The motivation for a partial clustering is that some objects in a data set may not belong to welldefined groups. Many times objects in the data set represent noise, outliers, or “uninteresting background.”



K-means

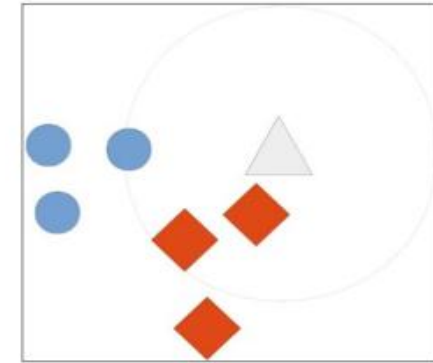
- ▶ This is a prototype-based, partitional clustering technique that attempts to find a user-specified number of clusters (K), which are represented by their centroids.

Table 7.2. K-means: Common choices for proximity, centroids, and objective functions.

Proximity Function	Centroid	Objective Function
Manhattan (L_1)	median	Minimize sum of the L_1 distance of an object to its cluster centroid
Squared Euclidean (L_2^2)	mean	Minimize sum of the squared L_2 distance of an object to its cluster centroid
cosine	mean	Maximize sum of the cosine similarity of an object to its cluster centroid
Bregman divergence	mean	Minimize sum of the Bregman divergence of an object to its cluster centroid



Nearest Neighbour



- ▶ Neighbors-based classification is a type of *instance-based learning* or *non-generalizing learning*: it does not attempt to construct a general internal model, but simply stores instances of the training data.
- ▶ To predict the class of a new sample, we look through the training dataset for the samples that are most similar to our new sample.
- ▶ We take the most similar sample and predict the class that the majority of those samples have.



Nearest neighbor methods

- ▶ The principle behind nearest neighbor methods is to find a predefined number of training samples closest in distance to the new point, and predict the label from these.

The number of samples can be

- ▶ a user-defined constant (k-nearest neighbor learning), or
- ▶ vary based on the local density of points (radius-based neighbor learning).

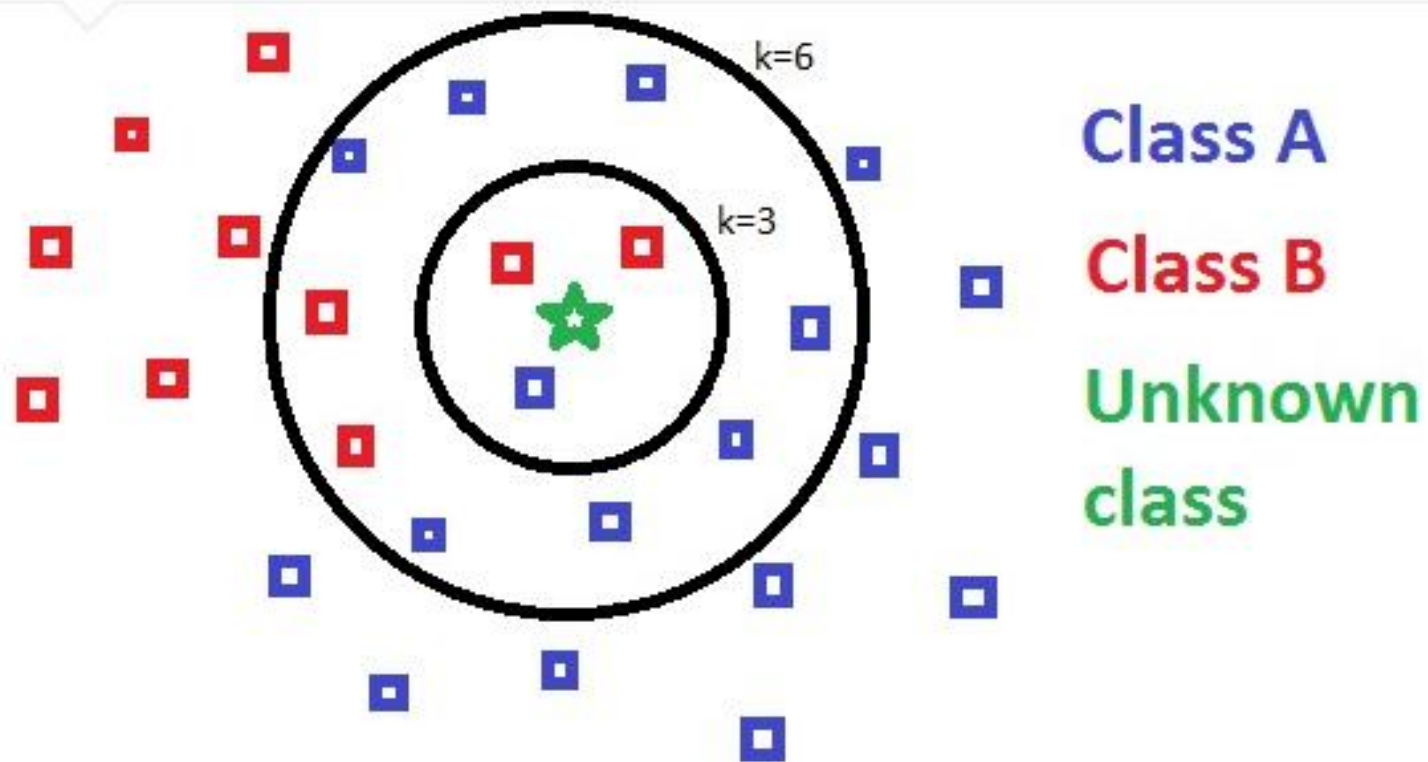


KNN can be summarized as

1. Computes the distance between the new data point with every training example.
2. For computing the distance measures such as Euclidean distance, Hamming distance or Manhattan distance will be used.
3. Model picks K entries in the database which are closest to the new data point.
4. Then it does the majority vote i.e the most common class/label among those K entries will be the class of the new data point.



K Nearest



Nearest neighbor methods

- ▶ Despite its simplicity, nearest neighbors has been successful in a large number of classification and regression problems, including handwritten digits and satellite image scenes.
- ▶ Being a non-parametric method, it is often successful in classification situations where the decision boundary is very irregular.



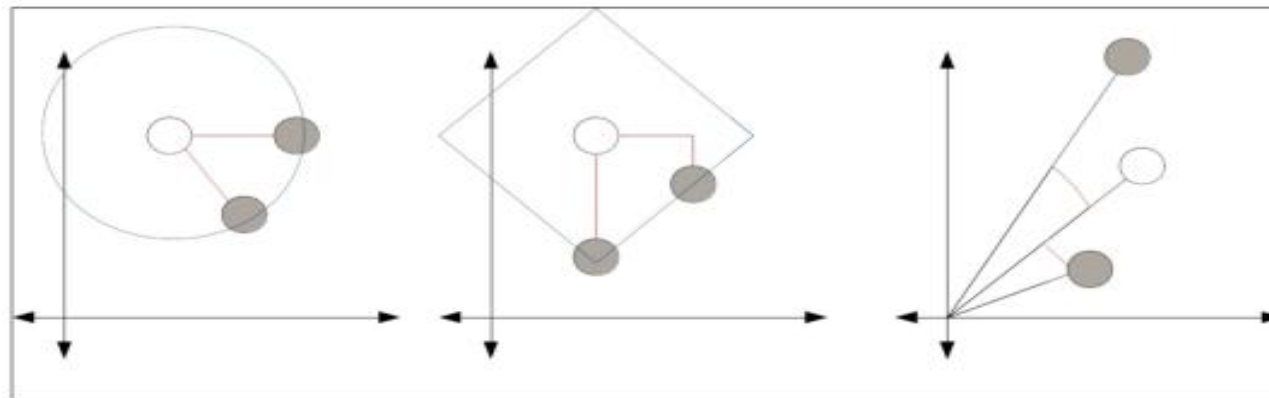
Nearest Neighbour

- ▶ Euclidean distance, which is the *real-world* distance.
- ▶ If you were to plot the points on a graph and measure the distance with a straight ruler, the result would be the Euclidean distance. A little more formally, it is the square root of the sum of the squared distances for each feature.
 - poor accuracy if some features have larger values than others
 - poor results when lots of features have a value of 0, known as a sparse matrix

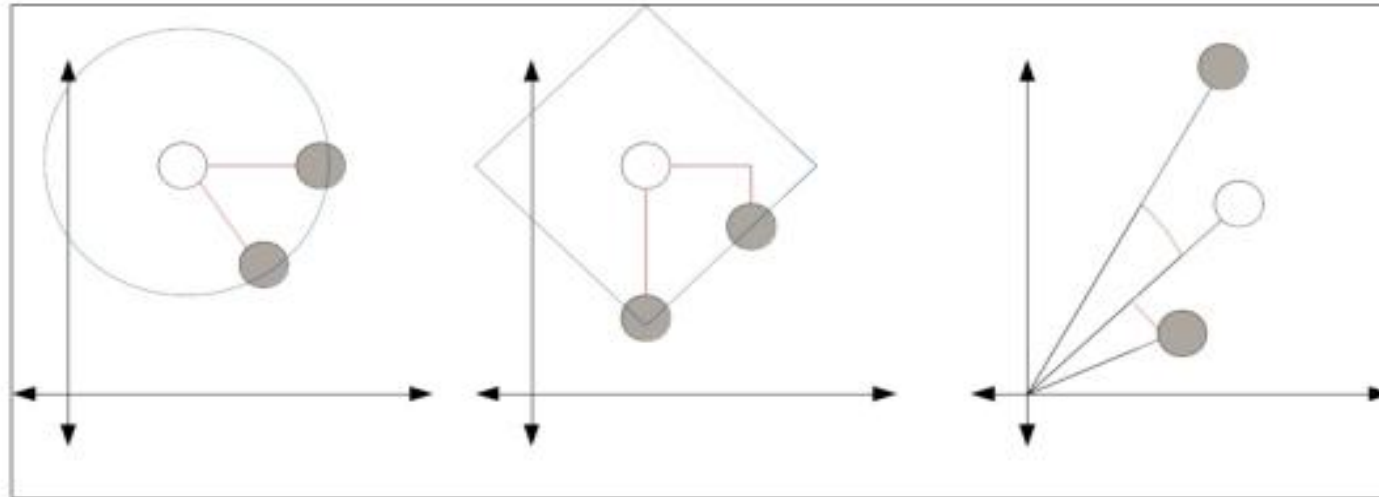


Nearest Neighbour

- ▶ The Manhattan distance is the sum of the absolute differences in each feature (with no use of square distances).
- ▶ While the Manhattan distance does suffer if some features have larger values than others, the effect is not as dramatic as in the case of Euclidean.



Nearest Neighbour



- ▶ The **Cosine** distance is better suited to cases where some features are larger than others and when there are lots of zeros in the dataset.
- ▶ we draw a line from the origin to each of the samples, and measure the angle between those lines.

sklearn.neighbors

- ▶ provides functionality for unsupervised and supervised neighbors-based learning methods.
- ▶ Supervised neighbors-based learning comes in two flavors:
 - classification for data with discrete labels, and
 - regression for data with continuous labels.
- ▶ The classes in sklearn.neighbors can handle either NumPy arrays or scipy.sparse matrices as input.

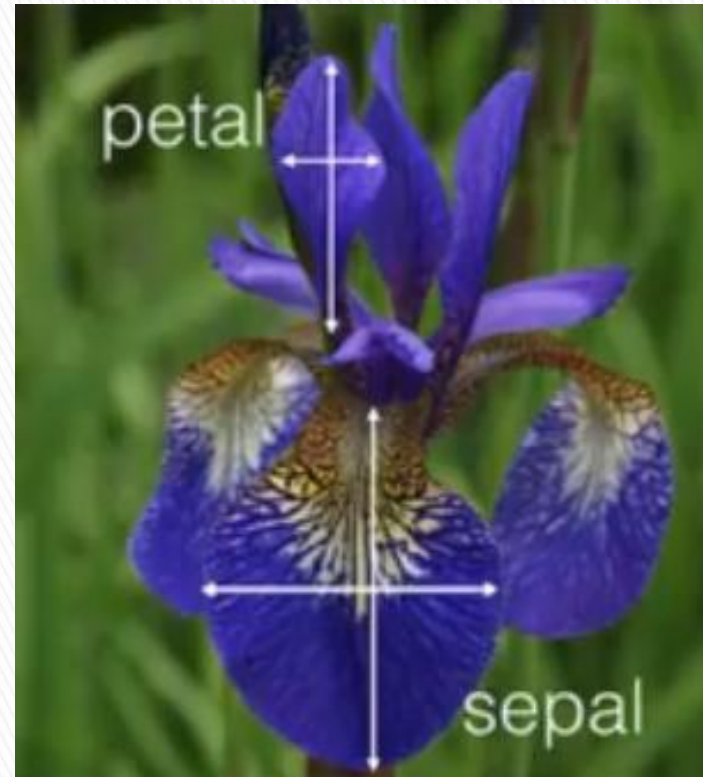
See documentation

<https://scikit-learn.org/stable/modules/neighbors.html>



Nearest Neighbour example data set

- ▶ First used in 1936!
This is a classic dataset
- ▶ Three classes:
Iris Setosa,
Iris Versicolour, and
Iris Virginica
- ▶ Response variable:
the iris species
- ▶ Classification problem
since response is
categorical.



iris

Nearest Neighbour –example

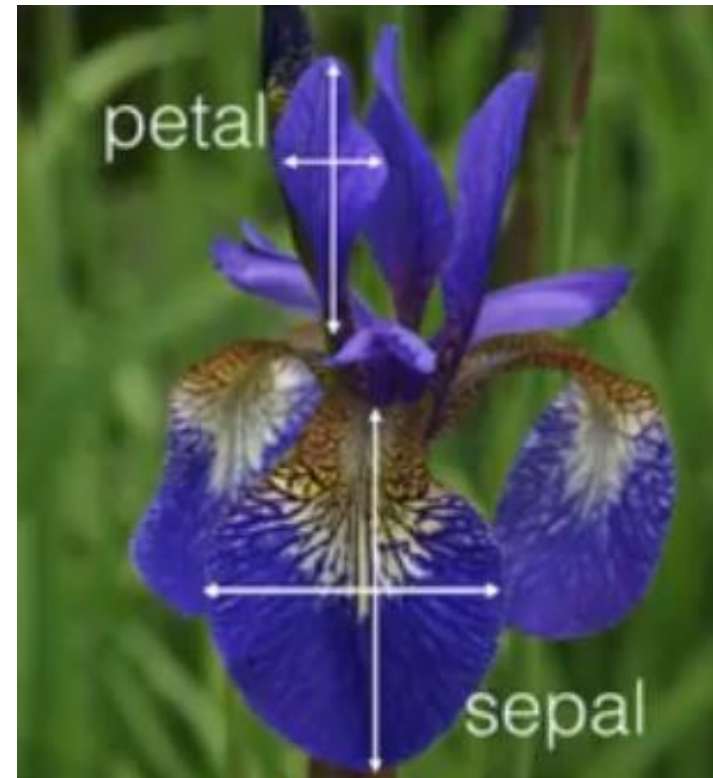
Iris data set

- ▶ 150 plant samples and four measurements of each:

sepal length,
sepal width,
petal length, and
petal width

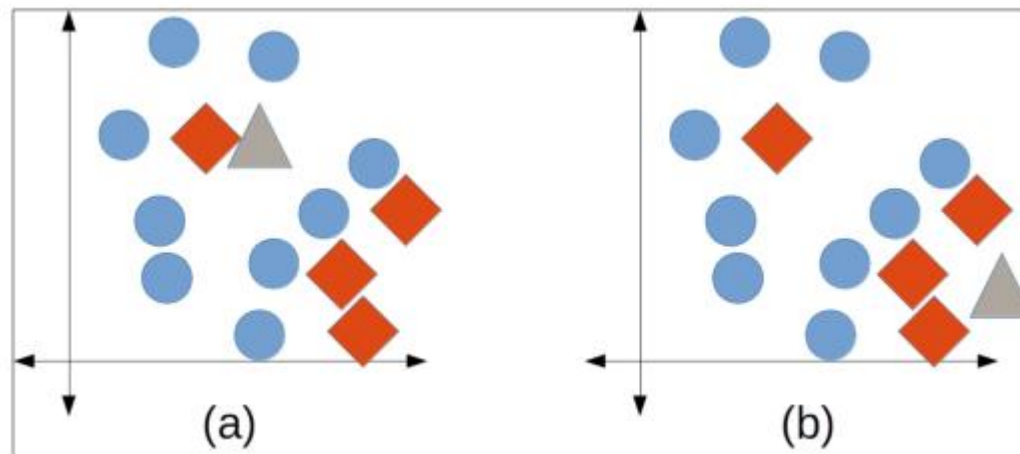
(all in centimeters).

- ▶ See `plot_cluster_iris.ipynb`



k-nearest neighbour

- ▶ The nearest neighbor algorithm has several parameters, but the most important one is that of the number of nearest neighbors to use when predicting the class of an unseen attribution.



Performance, clustering

- ▶ Finding nearest neighbors can require computing the pairwise distance between all points. Often clusters and their cluster prototypes can be found much more efficiently



Unsupervised

- ▶ Measures the goodness of a clustering structure without respect to external information. An example of this is the SSE.
- ▶ Unsupervised measures of cluster validity are often further divided into two classes:
- ▶ measures of **cluster cohesion** (compactness, tightness), which determine how closely related the objects in a cluster are, and
- ▶ measures of **cluster separation** (isolation), which determine how distinct or wellseparated a cluster is from other clusters.
- ▶ Unsupervised measures are often called **internal indices** because they use only information present in the data set.



Supervised

- ▶ Measures the extent to which the clustering structure discovered by a clustering algorithm matches some external structure. An example of a supervised index is entropy, which measures how well cluster labels match externally supplied class labels. Supervised measures are often called **external indices** because they use information not present in the data set



Classification–Oriented Measures of Cluster Validity

- ▶ **Entropy:** The degree to which each cluster consists of objects of a single class.
- ▶ **Purity:** Another measure of the extent to which a cluster contains objects of a single class.
- ▶ **Precision:** The fraction of a cluster that consists of objects of a specified class.
- ▶ **Recall:** The extent to which a cluster contains all objects of a specified class.
- ▶ **F–measure** A combination of both precision and recall that measures the extent to which a cluster contains *only* objects of a particular class and *all* objects of that class.

