Ingeniería de Sistemas

Data Analytics

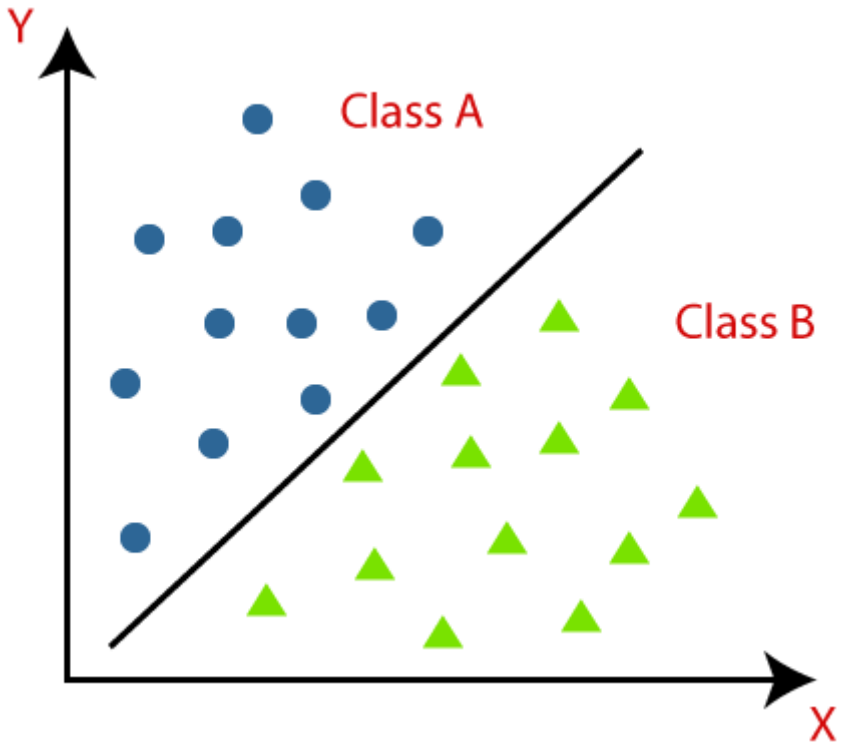
Prof: Hugo Franco

Session N° 13 | Clustering and Classification problems in Data Analysis

Bogotá D.C. Nov 08, 2022

Category identification as a data analysis problem

- Several real-life data analysis problems are related to qualitative descriptions of objects, processes, and/or phenomena.
- Such problems usually rely on decision-making tasks, such as assigning each new record or instance of the problem to a particular class, i.e., assigning a *categorical value.*

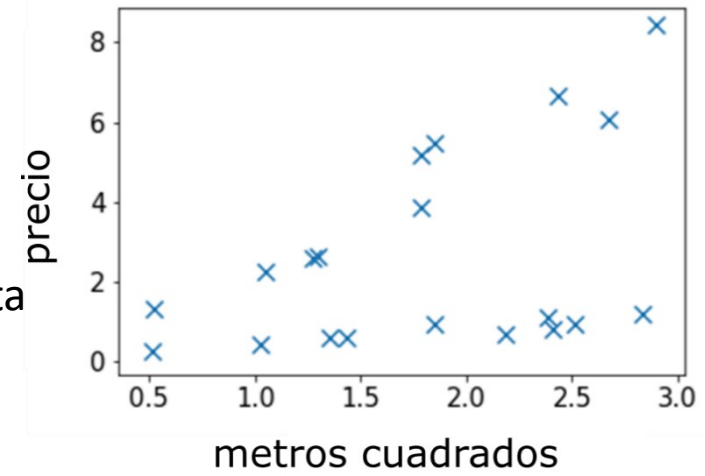


Clustering vs. Classification

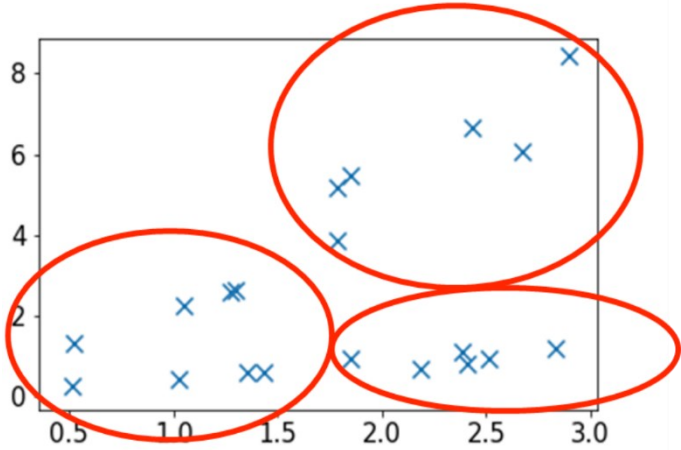
According to the knowledge provided by the supporting data, the labeling problem could be tackled by:

- **Unsupervised learning:** if the dataset does not have any (observed, expert-based) labeling information for the records (samples). Such problems are usually approached by **clustering** methods based on the internal structure of the data.
- **Supervised learning:** if the records in the dataset are previously labeled according to the data acquisition or by expert-based labeling. These problems can be approached by **classification** methods.

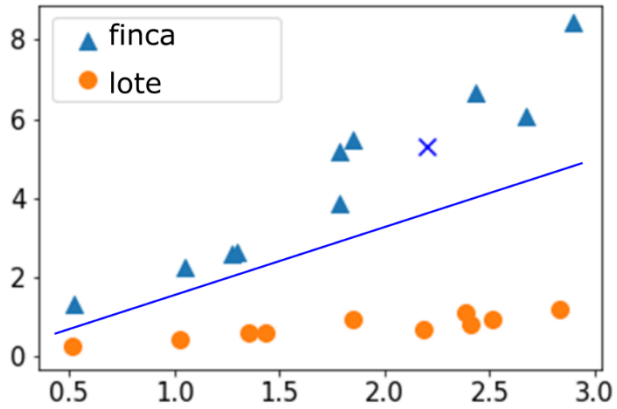
Problem data



Clustering
(no labels)



Classification
(labeled data)

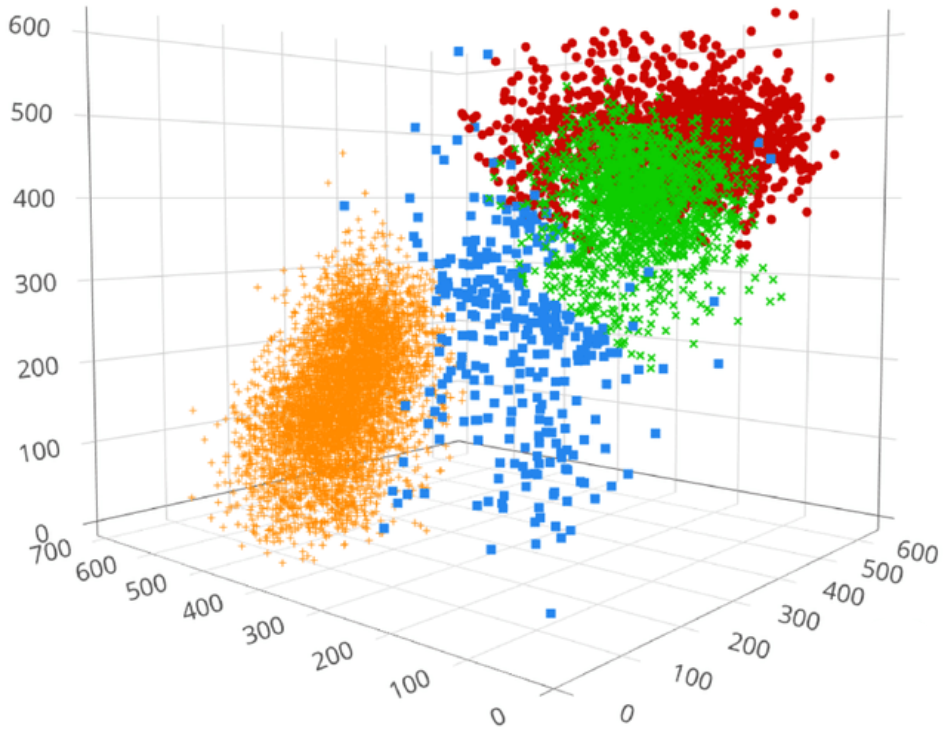


Clustering

Unsupervised learning

Clustering problems

- Given an unlabeled dataset (it has only values for the features describing each element/record) **associate a cluster (category) label to each record (*observation*) in the dataset,** according to the relative distances between different elements
- Such a distance is defined in terms of the respective positions of the samples within the *feature space*
  - *Each component (dimension) corresponds to a particular feature*

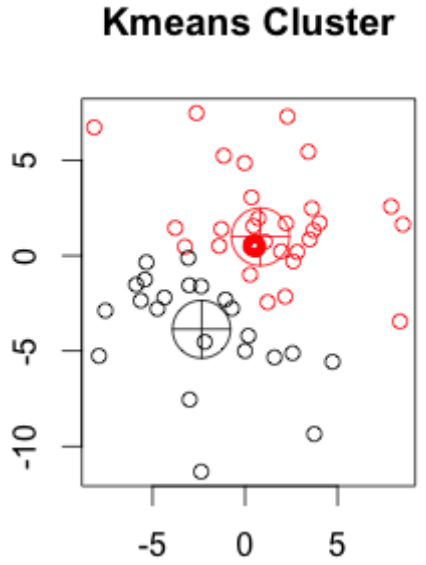


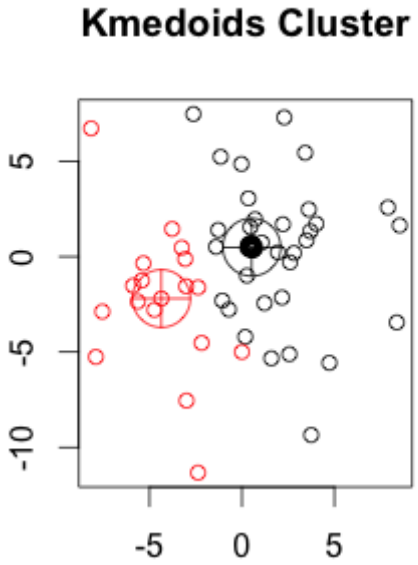
Clustering methods

- Clustering allows the grouping of similar data which helps in understanding the internal structure of the data
  - Most clustering approaches use the intrinsic distribution of the data in the “feature” space
- Clustering methods allow pattern identification within data without previous knowledge of the data source/distribution.
  - E.g.: **identifying** the target group (potential customers) for a specific product or service (segmentation)
- Clustering could be used to prepare the data for further Machine Learning methods/processes

Centroid-based methods I

- *K*-means: each cluster is defined according to its exact centroid (*not a real record in the dataset*) in terms of a pre-defined distance measure.
- Since there is no information on the underlying categories, *K* is an arbitrary parameter
  - The best value for *K* is usually selected according to a cost function, via the "Elbow plot": the critical value for K so that any further increase will not lead to a significant reduction in the cost function
- There are some *K*-means variations using different definitions for the cluster representative element
  - *K*-medians: each component of the representative element is the median value of the corresponding feature in the dataset (i.e., the representative element could not belong to the original dataset)
  - *K*-medoids: *the representant of each cluster is an actual record in the dataset*. The medoids are selected as those which minimize an overall cost function by successive exchanges (medoid vs. non-medoid elements) within the same cluster





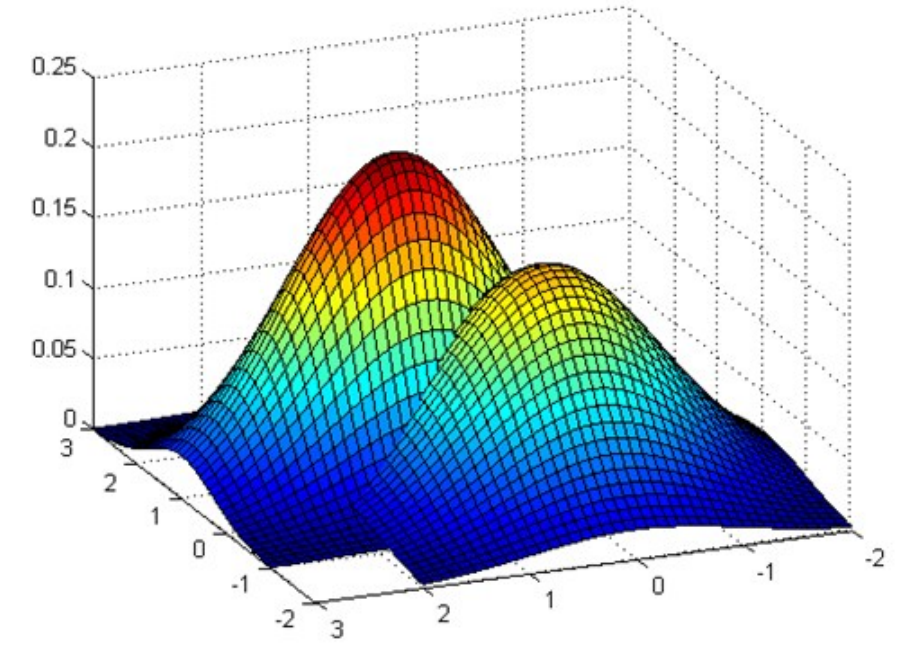
Centroid-based methods II

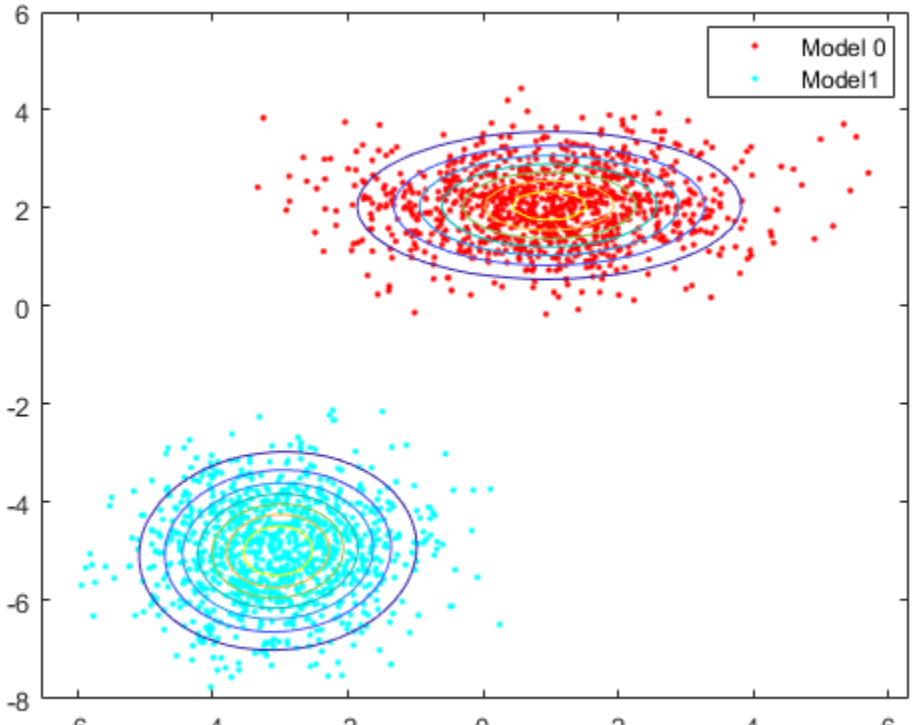
Gaussian Mixture methods:

- *K* Gaussian functions *are defined on the feature space*, assigning different class membership probabilities to each instance in the dataset:

$$G(X, \mu, \Sigma) := \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\top}\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]$$

  where Σ is a symmetric matrix, whose eigenvalues are the standard deviations for each feature, and $\boldsymbol{\mu}$ is the vector of feature mean values
- Each i-th Gaussian $i \in \{0, 1, \ldots, K-1\}$ provides the probability that each $x_j \in X$ belongs to the corresponding i-th cluster
- Σ and $\boldsymbol{\mu}$ calculated by using an Expectation Maximization approach within an iterative scheme (no changes on x_j labels between iterations





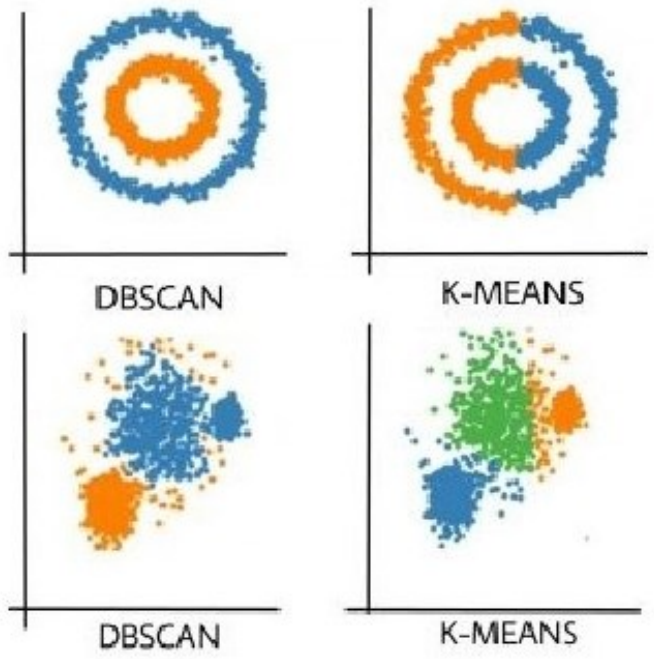
Local distance methods

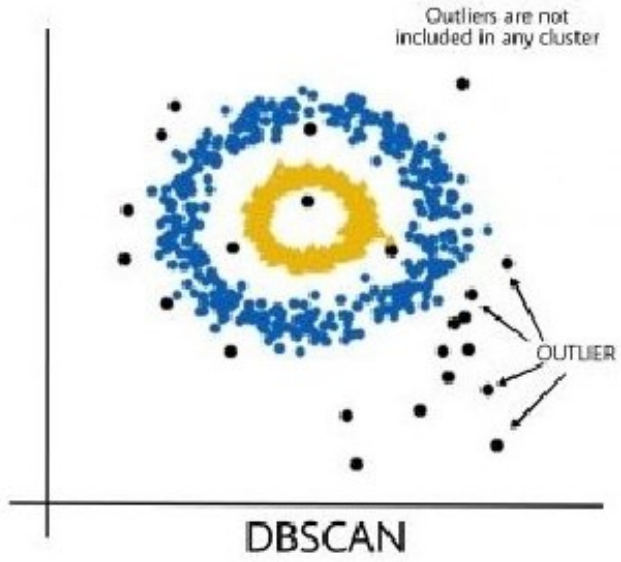
When the data topology exhibits a more complex behavior, the centroid-based methods could be insufficient to assign proper labels to the data

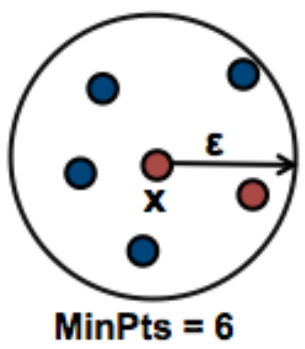
In such cases, it is desirable to use local relationships (distances) as the category assignment criterion.

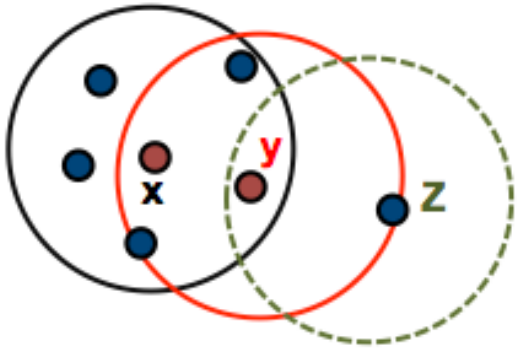
In the literature, there are several local distance methods, two of the most relevant of them are:

- **Mean-shift clustering**: based in sliding windows looking for local density maximization. Intersecting windows (clusters) are fused.
- **DBSCAN**: based on instance inclusion given point-wise densities (counts within a neighborhood)







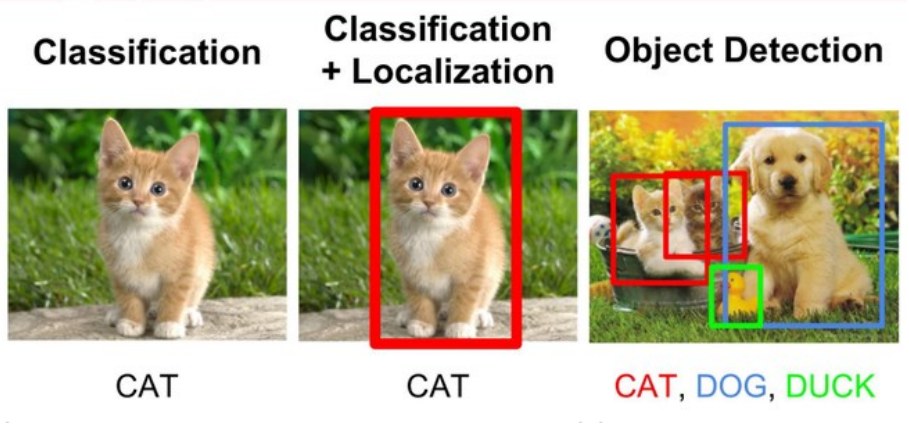


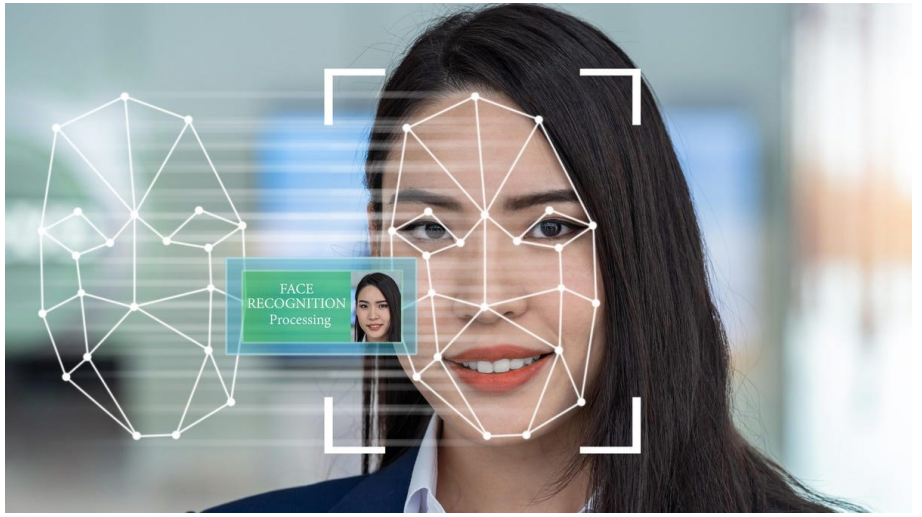
Classification

Supervised learning

Applications of classification methods

- Classification analysis can be used to :
  - Answer questions about a population (diagnostic data analysis)
  - Predict the behavior of a system (predictive data analysis)
  - Make decisions according to available data (prescriptive data analysis).
- *Example applications:*
  - **Detecting** the state of a mechanism according to descriptive data (i.e., sound recordings)
  - **Identifying** the type of each object contained in a digital image or the type of action in a digital video
  - **Recognizing** a particular face within a set of face captures of different individuals
  - Determining **(predicting)** the convenience of giving a credit to a particular bank client
  - **Detecting** the occurrence of a specific pattern in the values of the dataset records





Classification: Definitions

- **Classification model:** a statistical or computational model designed to assign a class/category label to an instance (input), given the values of its describing *features*.
  - The model *predicts* the corresponding *label* for each new instance.
- **Classifier:** a specific implementation of the classification model obtained by implementing the classification model using specific model parameter values.
- **Feature**: is an individual variable describing an attribute of the objects (instances) of a process or phenomenon under analysis.
- **Binary Classification:** problem based on the classification of the new instances presented to the model within two possible outcomes.
- **Multi-class classification:** problem using more than two classes for record/instance labeling; each instance is assigned to a unique class/category.
- **Multi-label classification:** problem of mapping each new instance to a set of non-excluding labels (each record can be assigned to more than one class).

Classifier training process

According to a problem specification (resulting from a formal problem analysis):

- **Select** an adequate classification model to approach the formal version of the problem.
- **Prepare** (transform) the dataset in order to support the corresponding training (learning) method. The dataset consists of features ($\mathbf{x}_i$) and labels (y_i), i=1, …, n
  - This usually includes the partition of the dataset in *training* and *testing* subsets
- **Train** the model, i.e., adjust the specific values of the model parameters to properly fit the training data using a specialized training (learning) algorithm and the training subset (only).
  - In scikit-learn, all models use a fit($\mathbf{x}_i$, y_i) method to train the model.
- **Evaluate** the classification model performance according to its behavior (classification abilities) for the testing dataset.

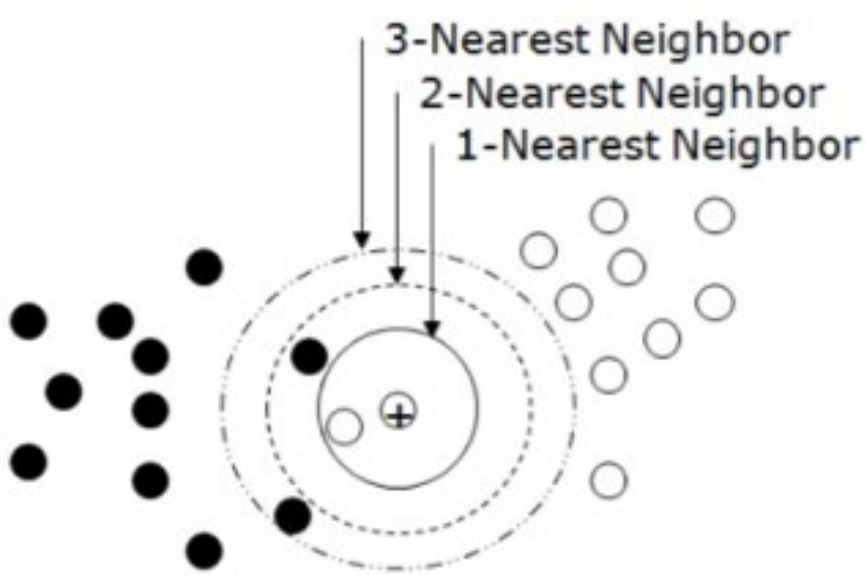
After the model has been properly validated, it is possible to predict the label target of new (unlabeled) observations $\mathbf{x}_j$: the prediction returns the corresponding label y_j.

Some relevant classification models

K nearest neighbors (KNN)

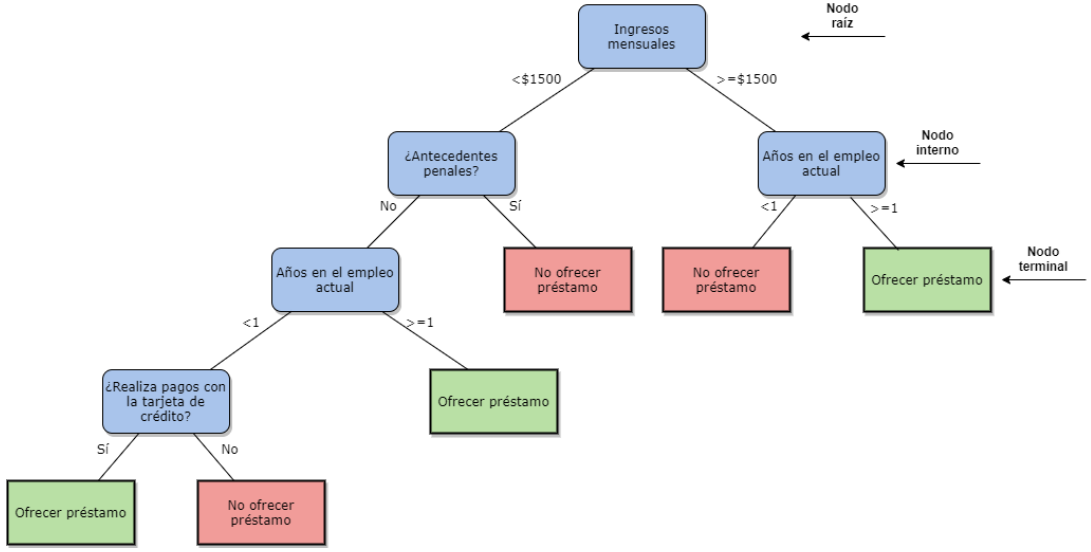
Classification approach based on the distance of each new instance to its neighbors in the training dataset

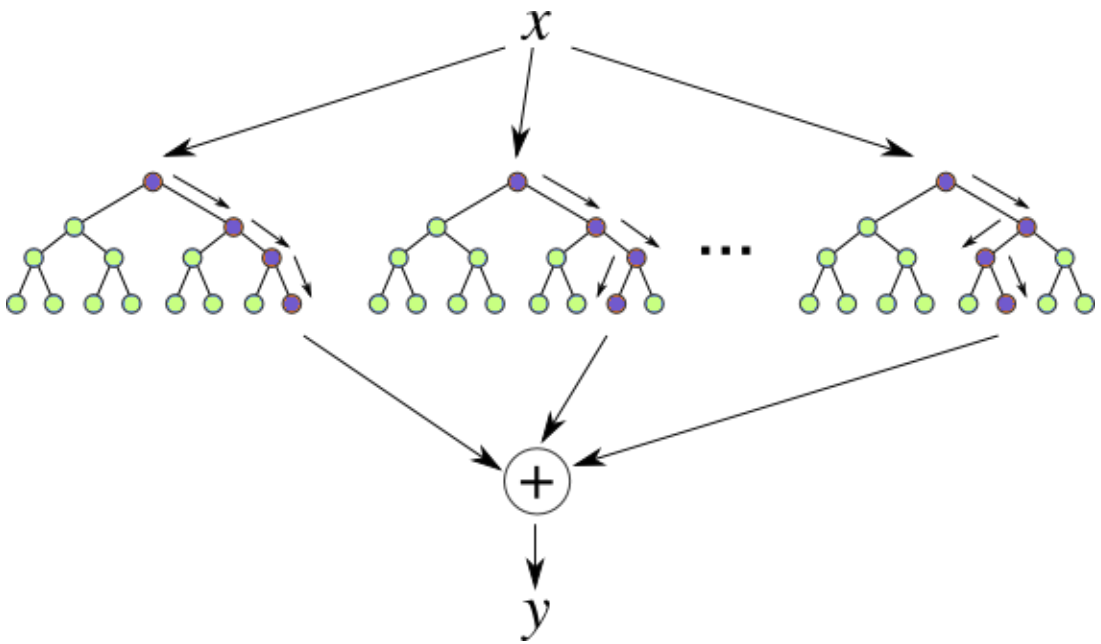
- The label for the new instance is assigned by the most numerous class within its K nearest neighbors (voting)
- **Advantages:** easy to implement
- **Disadvantages:** very inefficient for large datasets (high algorithmic complexity). Highly dependent on the parameter K and the distance definition



Decision trees and random forests

- Models based on decision rules obtained according to the ability of each individual feature to enhance the class separation of the instances in the dataset.
- Ensemble models (random forests/ gradient boosting) enhance the robustness of the classification by using multiple partial decision trees (some features are excluded).
  - Each tree in the random forest is trained using only a subset of the original dataset.
- **Advantages:** higher interpretability given its focus on decision rules for each feature (applies to random forest models for each subtree).
- **Disadvantages:** prone to overfitting (partially solved by ensemble models).





Logistic Regression

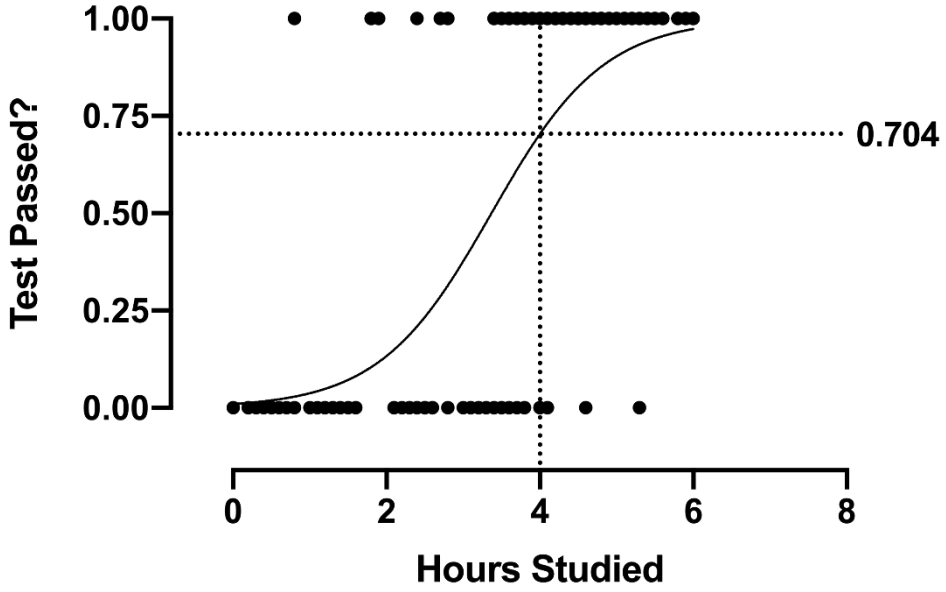
This method provide the probability of a particular outcome (label assignment) for a single instance using the logistic function.

$$P = \frac{1}{1 + e^{-(a+bX)}}$$

The training method corresponds to the estimation of parameters a and b

- **Advantages:** high interpretability according to statistical criteria
- **Disadvantages:** assumes all predictors are independent variables. Data must be free of missing values or extreme outliers.

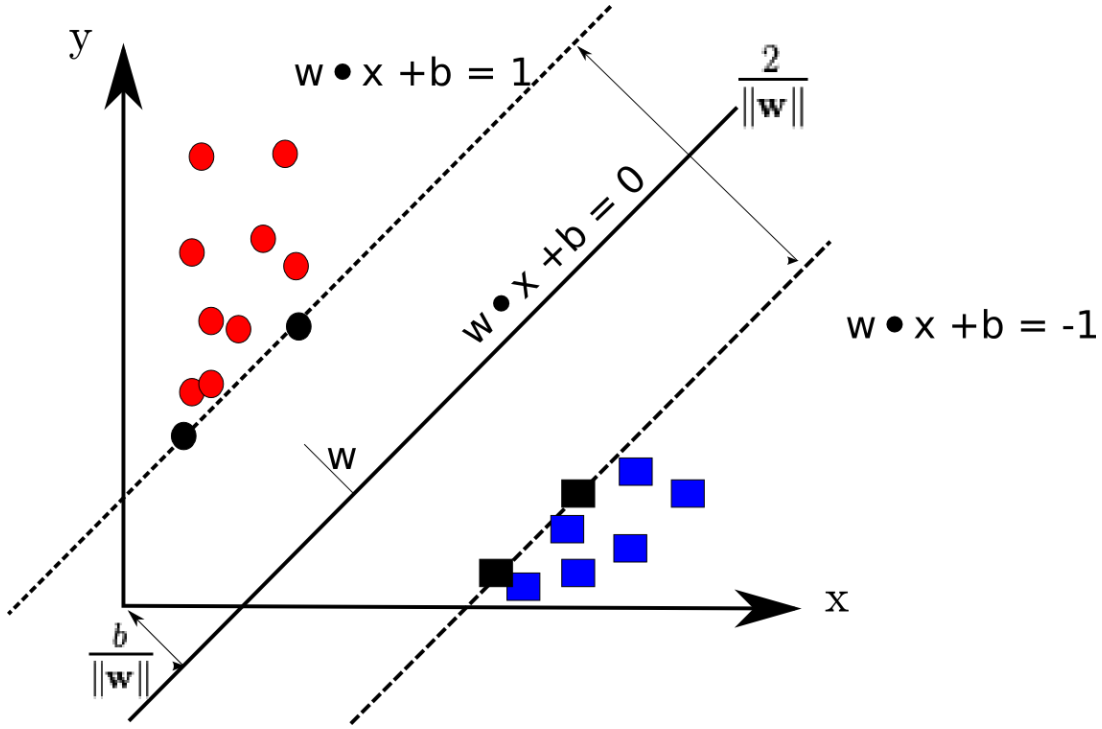
Example:



Support vector machines

This model approach uses a mathematical representation of the dataset as points in the feature space to obtain the optimal hypersurface in terms of class separation (distance between groups)

- The classification problem is then formulated as a separation between point clouds
- **Advantages:** robust to outliers, accurate according to data distribution, allows the use of different hypersurface models (kernels), both linear and non-linear
- **Disadvantages:** kernels must meet complex mathematical restrictions (kernel design is not immediate).



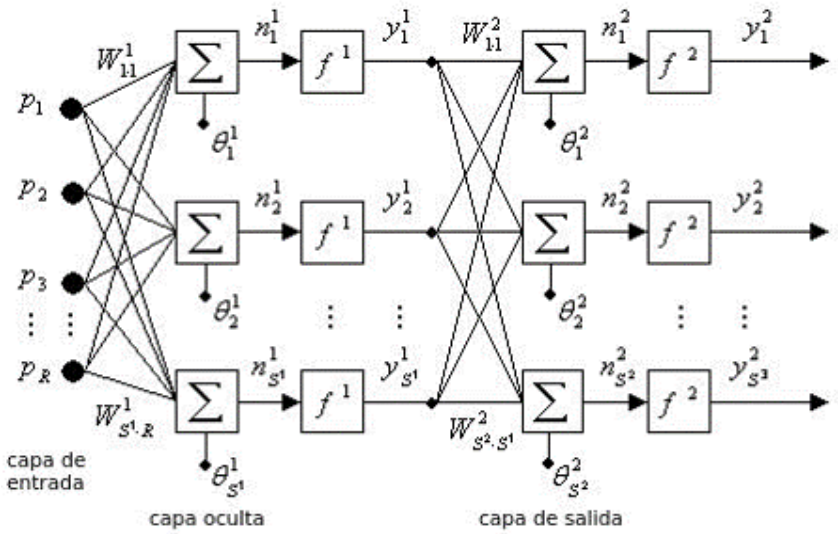
Neural network models

Bioinspired models based on the classification ability of several units working together in parallel

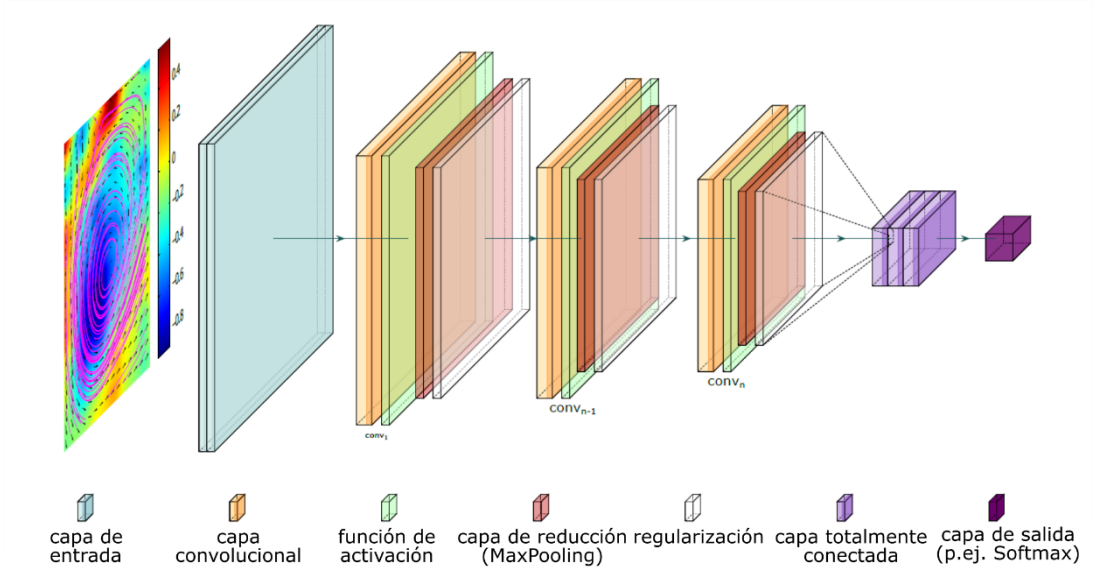
- Neural networks are trained (assignment of weight values) by error minimization algorithms (optimization by, e.g., gradient descent algorithms)

- **Advantages:** modern approaches provide high classification performances for complex problems
- **Disadvantages:** low interpretability since they are black boxes (newest models try to approach this problem). High computational cost.

Multilayer perceptron



Convolutional neural network (CNN)



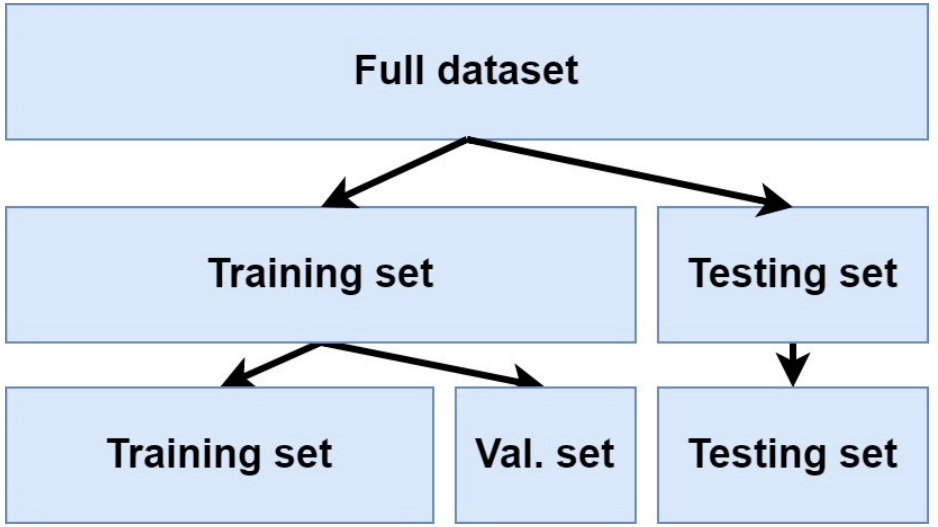
Classification model evaluation

Evaluation of the classification performance

- To evaluate the classification performance, the actual (real) label in the dataset (y_j) is compared with the predicted label ($\hat{y}_j$) yield by the classifier under evaluation:

$$y_j - \hat{y}_j$$

- The model must be *fairly evaluated*, i.e., all records used in the model training process must be **excluded** of every model evaluation calculation, so the values y_j and $\hat{y}_j$ must be taken from the testing set, thus preventing any evaluation bias



Confusion matrix

- In classification, the model output $\hat{y}_j$ assigns a label to every input to a specific class, i.e.

$$y_j, \hat{y}_j \in \{c_1, \dots, c_k\}$$

- From the point of view of a particular class c_i (label), a proper label assignment for the *j*-th record is obtained when $\hat{y}_j = y_j = c_i$ (a **true positive**). If the record is correctly excluded of the class $\hat{y}_j = y_j \neq c_i$, it is a true negative
- If the *j*-th record belongs to the class but the classifier assigns a different label, it is **false negative** $\hat{y}_j \neq y_j = c_i$
- If the *j*-th record does not belong to the class but the classifier assigns the label corresponding to c_i, it is **false positive** $\hat{y}_j = c_i \neq y_j$

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Classification performance metrics

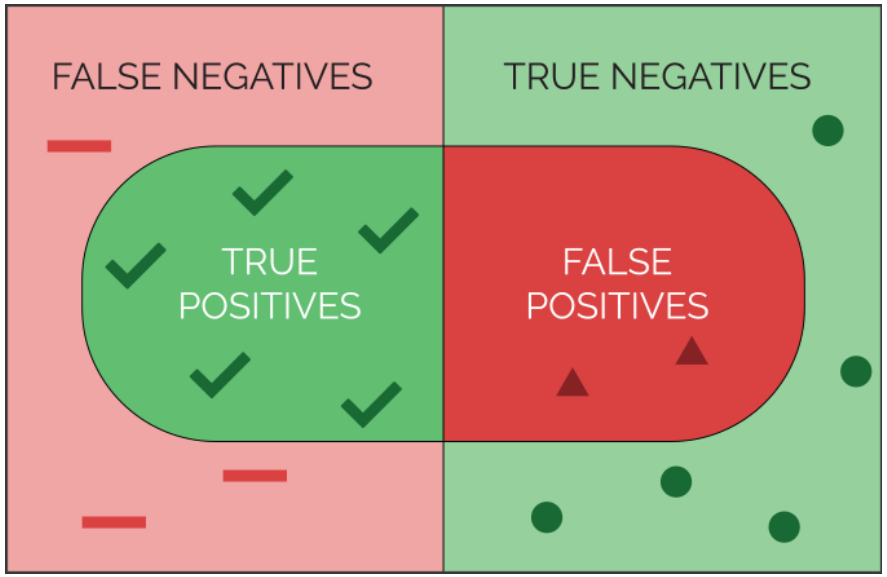
$$\text{Accuracy} = \frac{\text{properly classified records}}{\text{total of records}} = \frac{TP+TN}{TP+TN+TP+TN}$$

$$\text{Precision} = \frac{\text{records properly assigned to the class by the classifier}}{\text{total of records assigned by the classifier to the class}} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{\text{records properly assigned to the class by the classifier}}{\text{total of records identified as part of the class in the dataset}} = \frac{TP}{TP+FN}$$

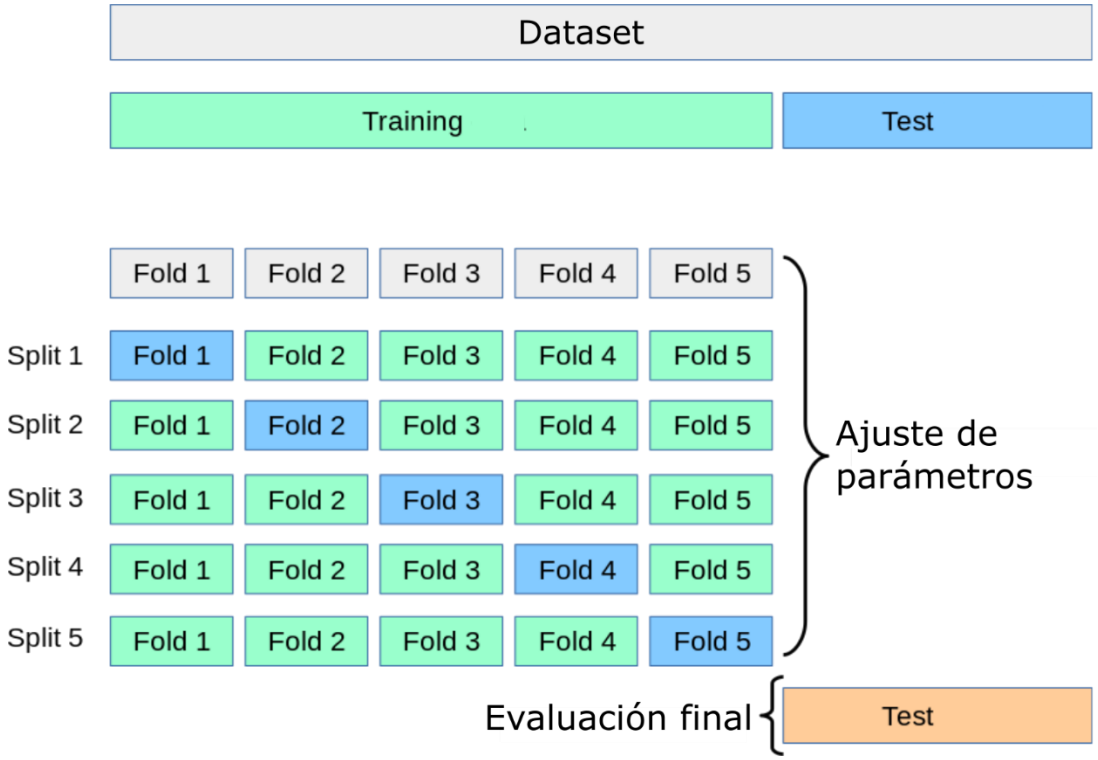
$$\text{Specificity} = \frac{\text{records properly excluded of the class by the classifier}}{\text{total of records identified as part of other classes in the dataset}} = \frac{TN}{TN+FN}$$

$$\text{F1-score} = 2\frac{\text{Precision} \times \text{Recall}}{\text{Precision}+\text{Recall}}$$



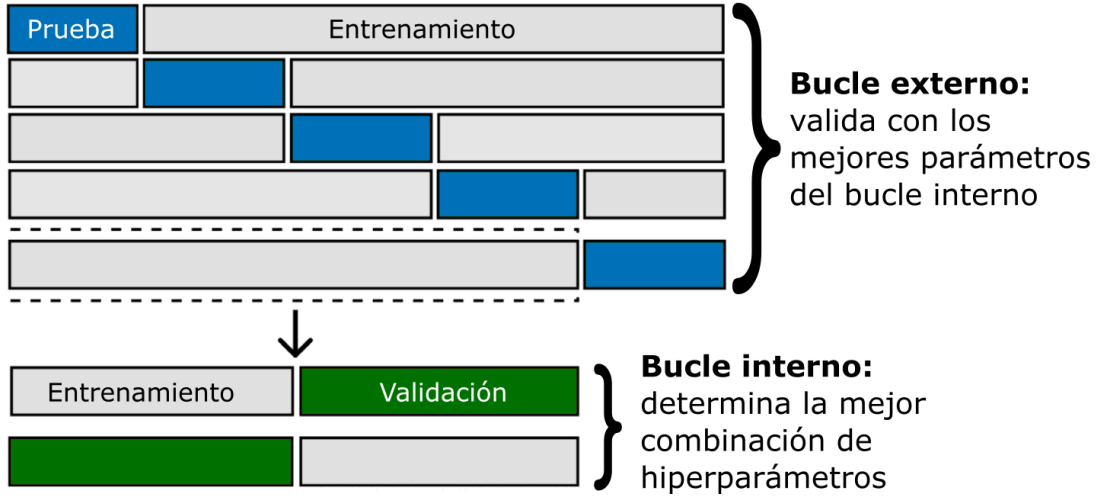
Cross validation

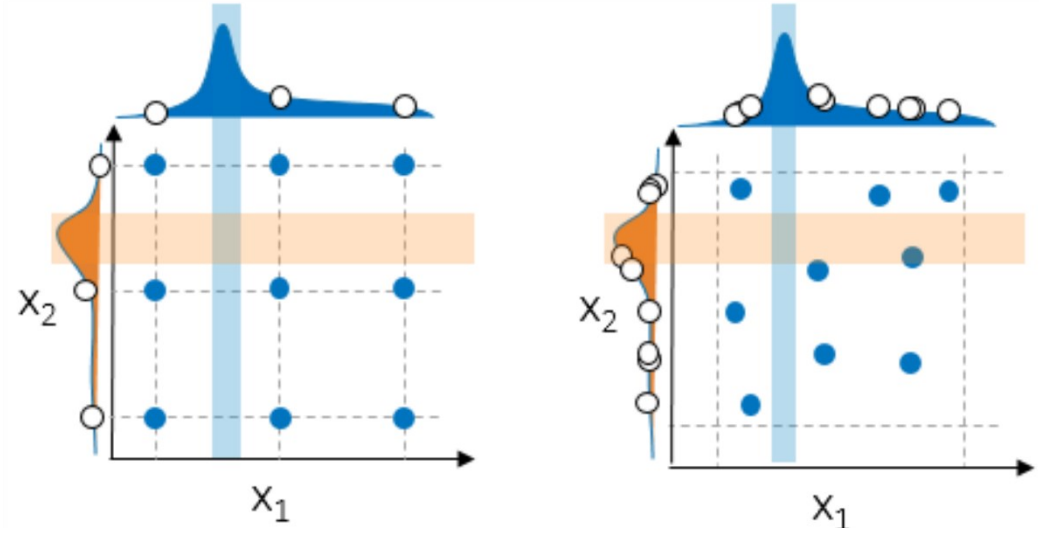
- To perform a reliable model training and obtain fair model evaluation metrics, cross validation methods are applied
- The K-fold cross validation consists in splitting the training set into K subsets, and iteratively train the model using only K-1 subsets until a metric (usually the average accuracy) reaches a threshold value for all the folds.
  - The excluded subset is used for *validation* (a partial performance measure)
- The resulting model is then evaluated against the testing set.



Hyperparameter tuning and model selection

- There are some model parameters not related to the training process (e.g., learning rates, kernel functions, loss functions, optimizers, etc.) that modify the classifier performance. They are called “hyperparameters”
- To obtain the best model (model selection), different versions of the general model are trained and evaluated using a cross validation strategy
- The hyperparameter space must be traversed using some parameter sweep strategy.





Grid search

Random search