THEORETICAL FOUNDATIONS OF ML: CLASSICAL PROBLEMS, ALGORITHMS AND VALIDATION

SOMACHINE 2021 | IAA-CSIC + UGR
April 19th 2021

IAA INSTITUTO DE ASTROFÍSICA DE ANDALUCÍA
EXCELENCIA SEVERO OCHOA
CSIC
DaSCI Andalusian Research Institute in Data Science and Computational Intelligence

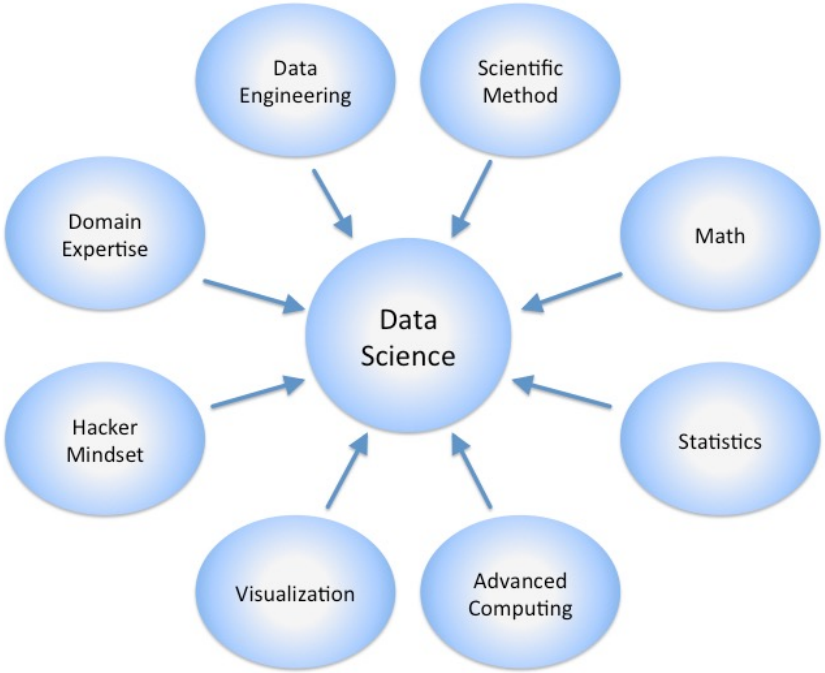
Table of Contents

BASIC CONCEPTS | 1

BASIC CONCEPTS

Data Science

Data Science is the field of knowledge that encompasses the skills associated with data knowledge extraction, including Big Data

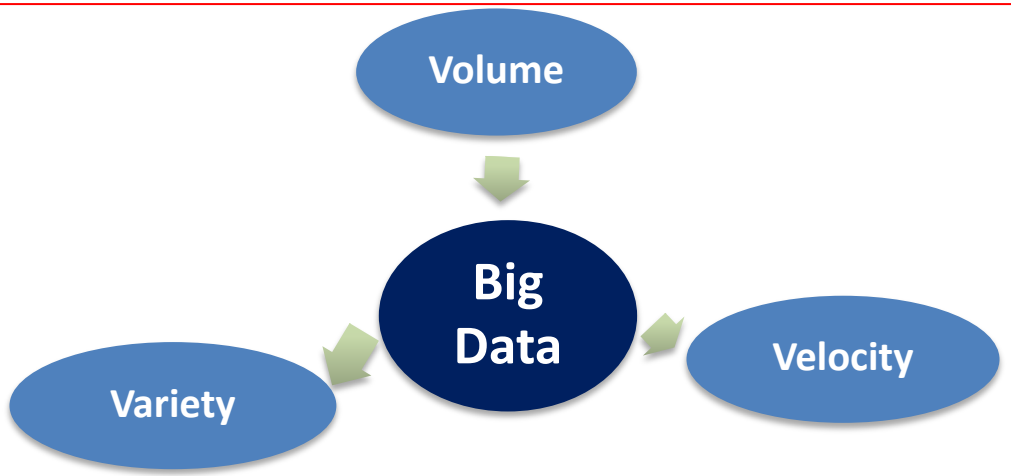


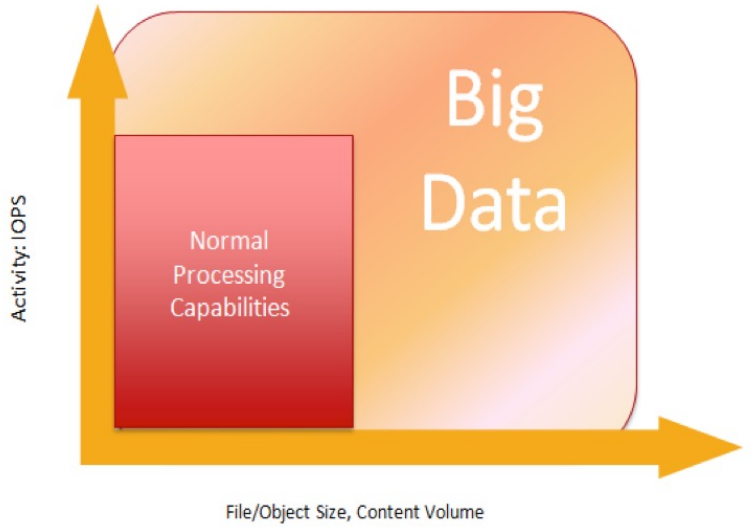
Machine Learning is a scientific discipline in the field of Artificial Intelligence that creates systems that learn automatically. Learning in this context means identifying complex patterns in data.

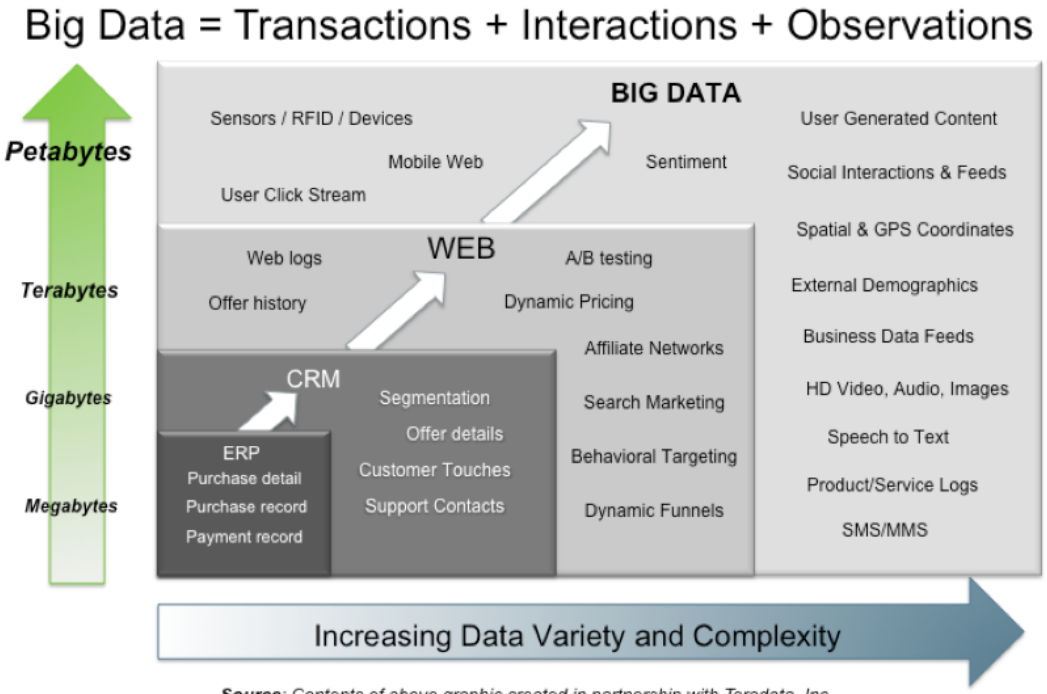
The machine that actually learns is an algorithm that examines the data and is able to predict future behavior.

BASIC CONCEPTS

"*Big Data*" are data whose volume, diversity and complexity require new architecture, techniques, algorithms and analysis to manage and extract value and knowledge hidden in them ...







***Source**: Contents of above graphic created in partnership with Teradata, Inc.*

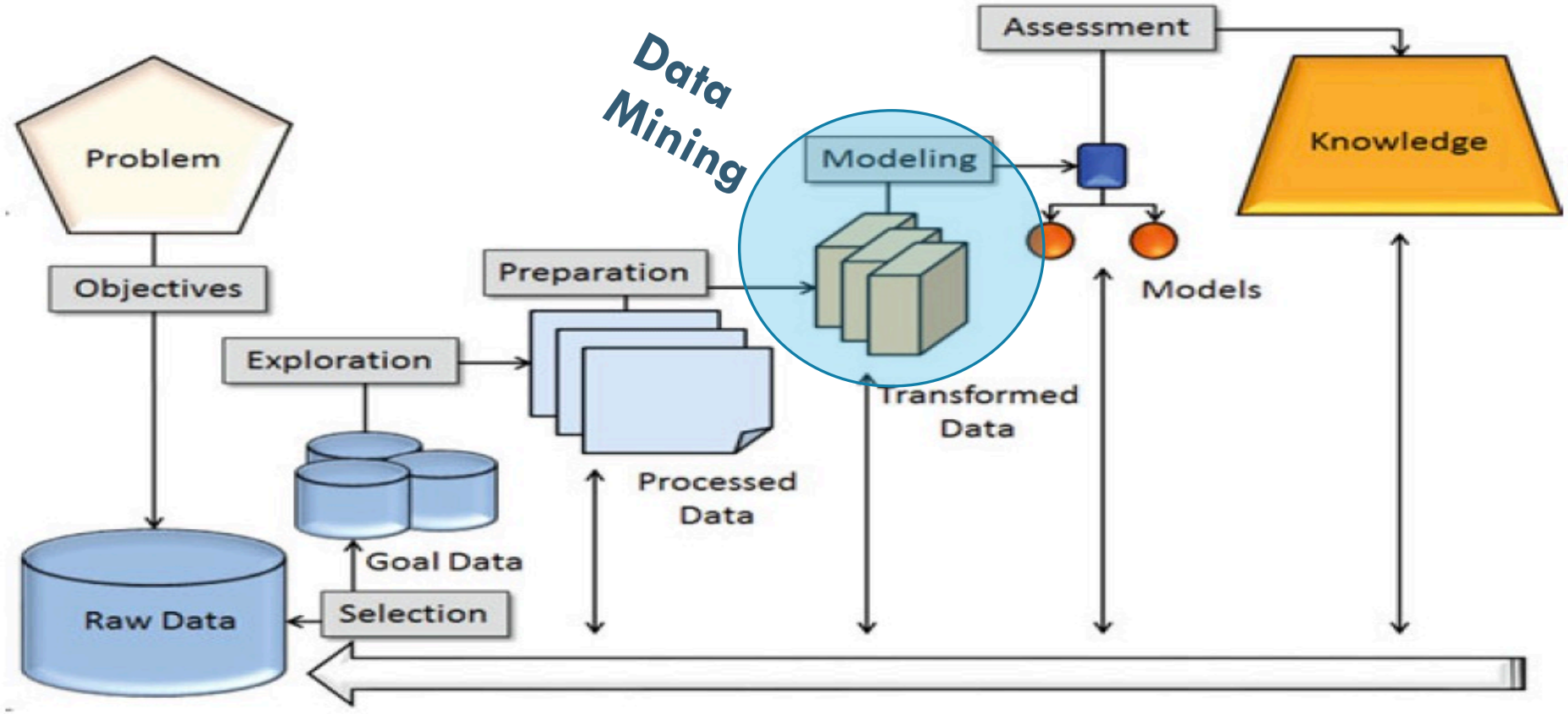
DATA MINING PROCESS | 2

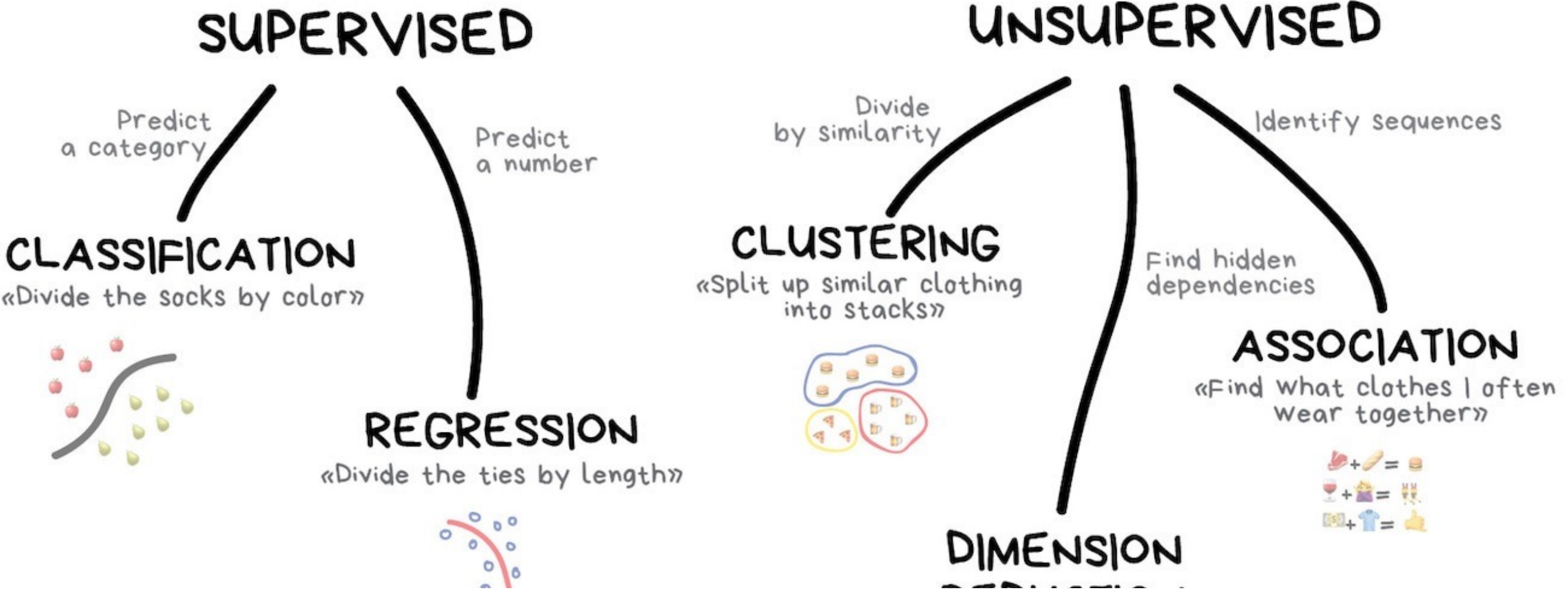
DATA MINING PROCESS

- KDD = *Knowledge Discovery from Databases*
- KDD is the complete process of knowledge extraction from databases
- The term was coined in 1989 to emphasize that knowledge is the end product of a data-driven discovery process
- Data Mining is only one step in the KDD process
- Informal association of Data Mining with KDD

DATA MINING PROCESS

STAGES IN A KDD PROCESS





ML CLASSICAL TASKS | 3

DATA REPRESENTATION

- The traditional data format is a tabular one:
  - Each real-world sample is encoded in a **row → instances**
  - Each sample is measured in several **properties → attributes**

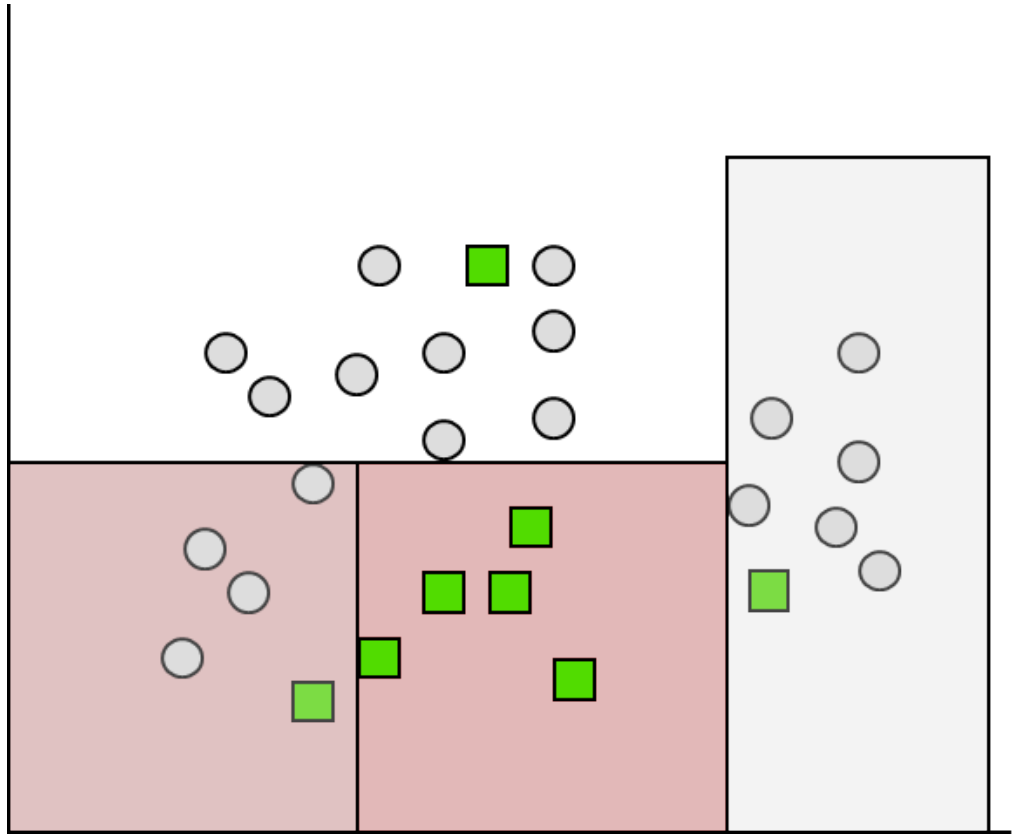
	Sepal length	Sepal width	Petal length	Petal width	Class
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
⋮	⋮	⋮	⋮	⋮	⋮
150	5.9	3.0	5.1	1.8	virginica

Data set with 150 samples

Tuple, instance, point or example

Feature, attribute or dimension

SUPERVISED VS. UNSUPERVISED LEARNING

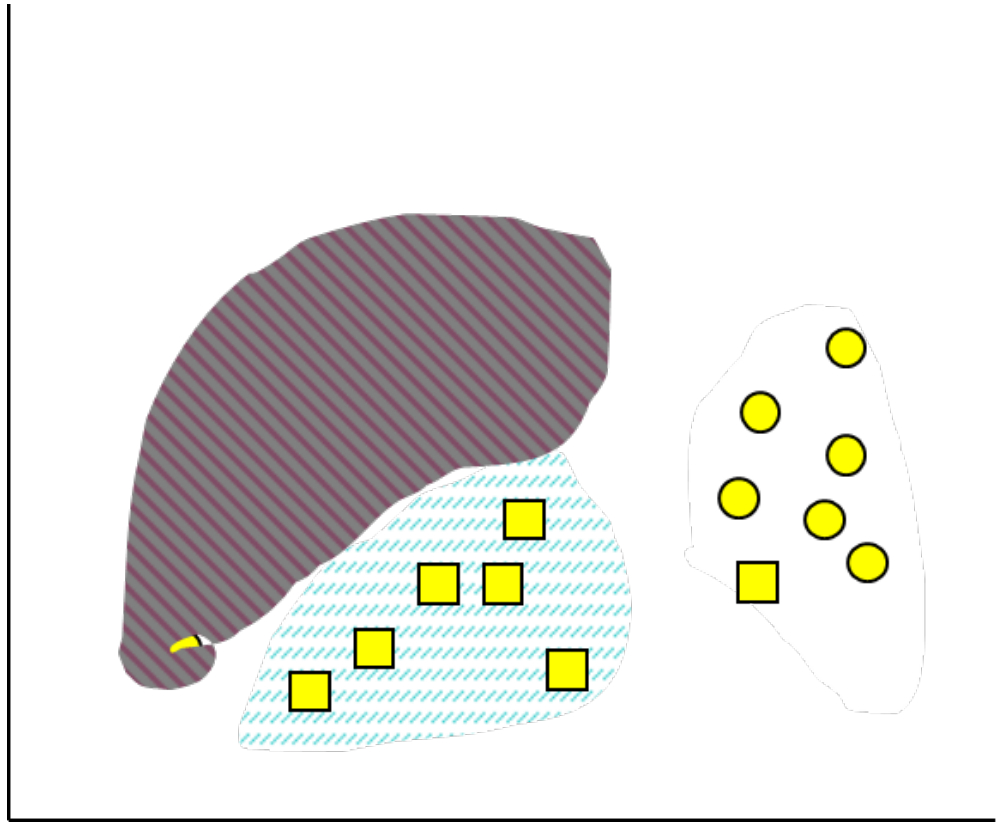


Supervised learning:

Learn, from a set of pre-labelled instances a model to future predictions

(Example, classification: the class to which a new instance belongs)

SUPERVISED VS. UNSUPERVISED LEARNING



Unsupervised learning:

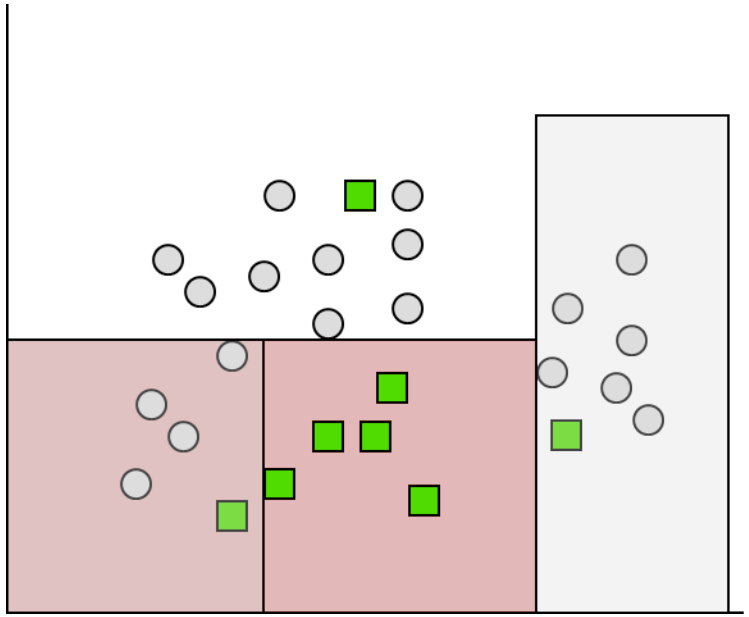
No a priori knowledge of the problem, no labeled instances, no supervision of the procedure.

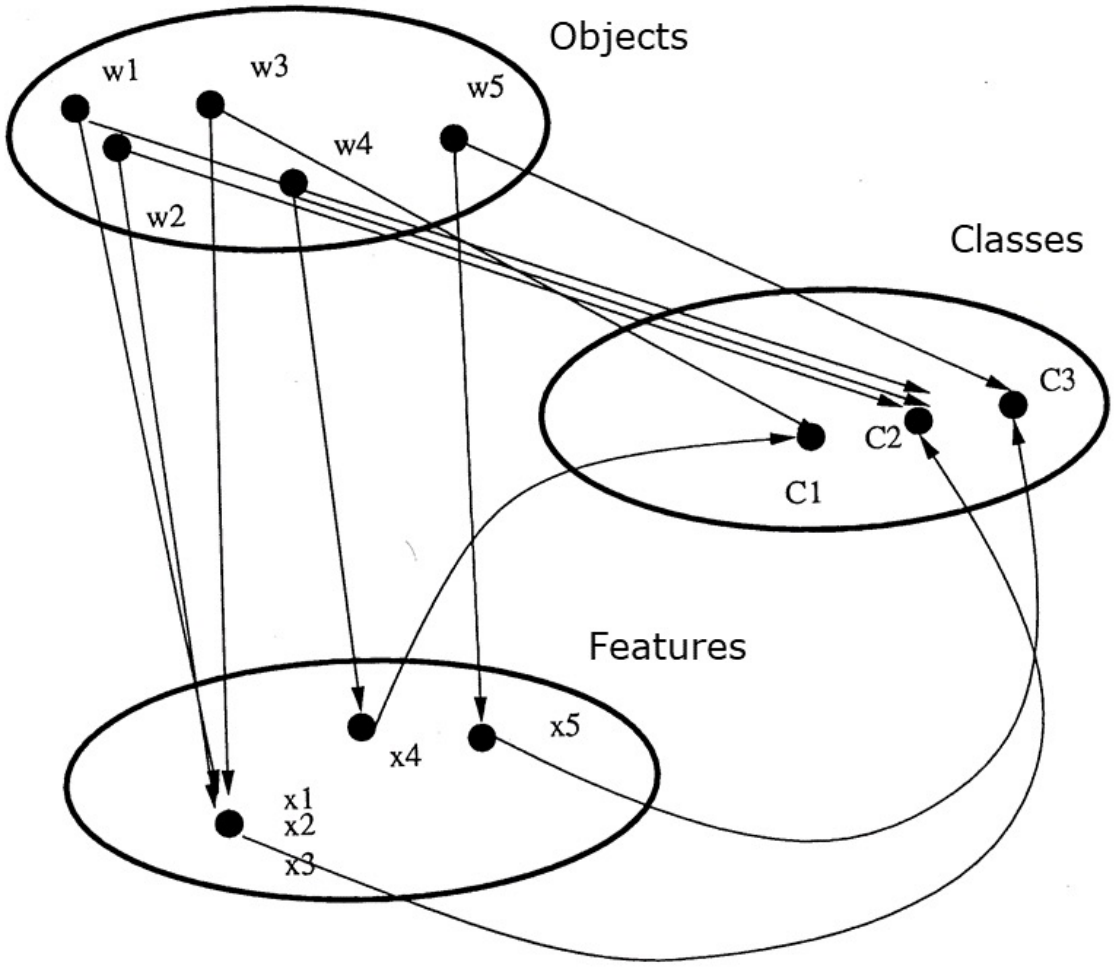
(Example, clustering: Find a "natural" grouping of instances given a set of unlabeled instances)

CLASSIFICATION

- **Clasification**

The fundamental **problem** of classification is directly related to the separability of classes.

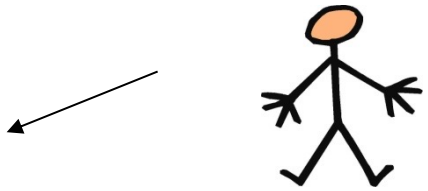




DEFINITION OF THE CLASSIFICATION PROBLEM

- An object is described by a set of characteristics (variables or attributes)

$$X \rightarrow \{X_1, X_2, \ldots, X_n\} \qquad e_i \rightarrow \{x_1, x_2, \ldots, x_n, c_k\}$$



- Age
- Astigmatism
- Tearing rate
- Myopia
- CLASIFICATION: Contact lens type

-Income
-Debts
-Assets…

-CLASIFICATION: Credit approval

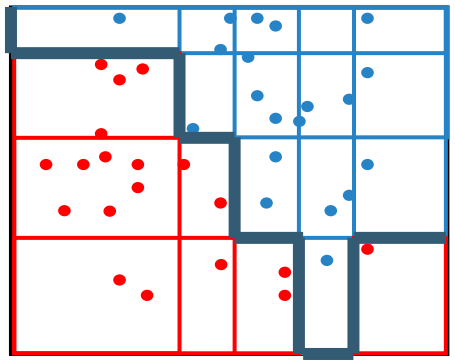
- Objective of the classification task : Classify the objective within one of the class categories $C=\{c_1,\ldots,c_k\}$

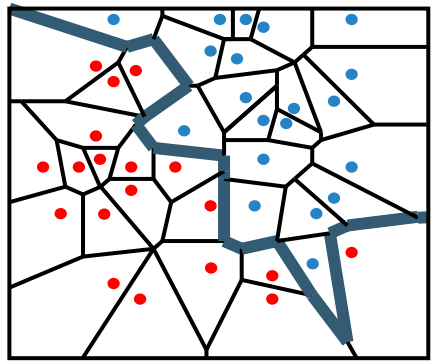
$$f: X_1 \times X_2 \times \ldots \times X_n \rightarrow C$$

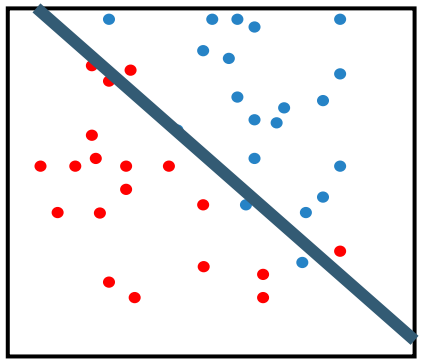
- The features or attributes chosen depend on the classification problem

DEFINITION OF THE CLASSIFICATION PROBLEM

- A **classifier** can be a set of rules, a decision tree, a neural network, etc.







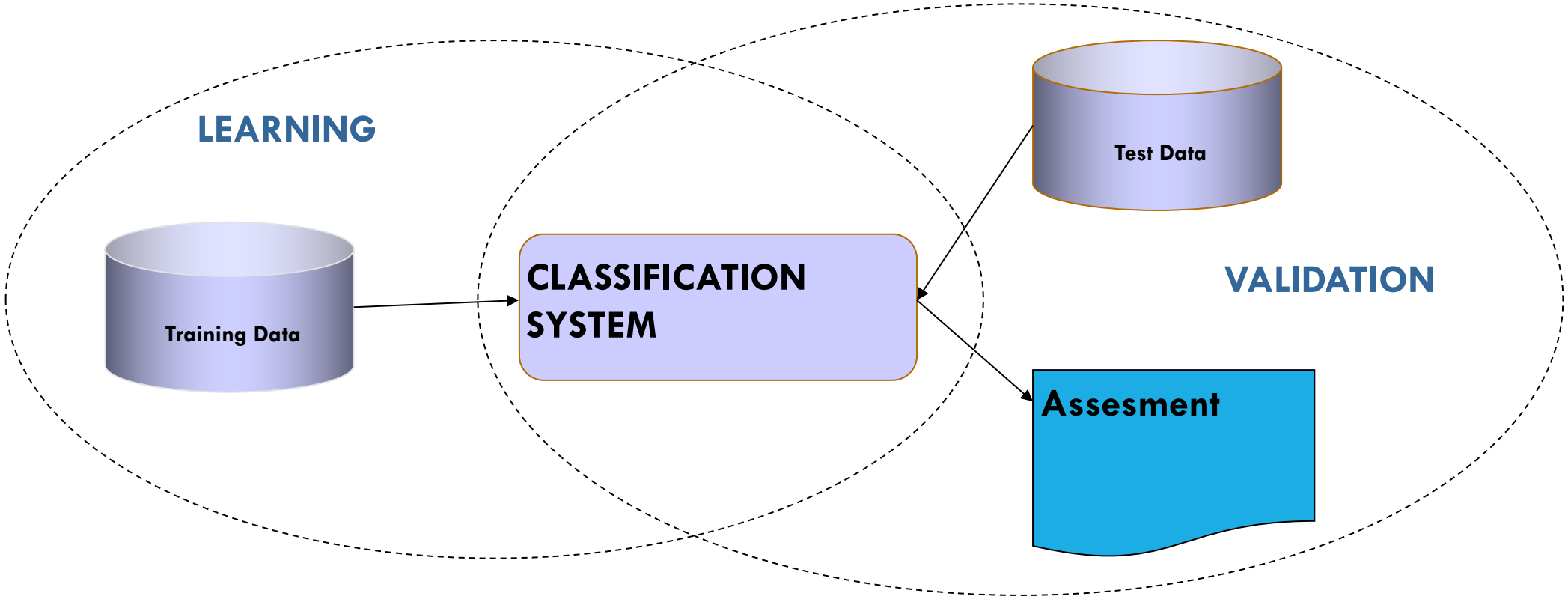
- Typical applications:
  - Credit approval,
    direct marketing, fraud detection,
    medical diagnosis...

STAGES IN THE CLASSIFICATION PROCESS

1. Model construction describing the set of (default) classes: Learning Phase
   - Each example (instance) is known to belong to a class (class attribute label)
   - A set of examples is used for the construction of the model: *training set*.
   - The model obtained is represented as a set of classification rules, decision trees, mathematical formula, …

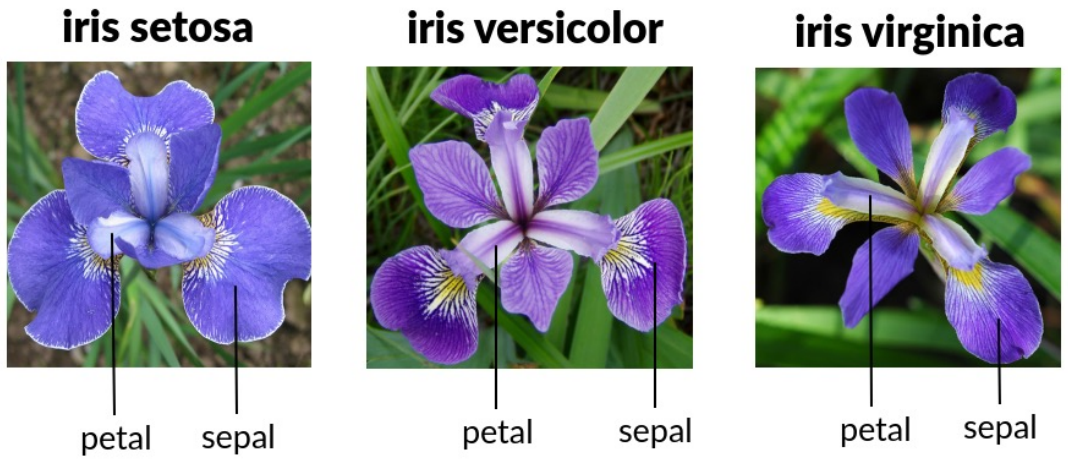
2. Use of the model: Validation
   - Model accuracy estimation:
   - A different set of examples is used from those used for the construction of the model → *test set*
     - If the test set was not independent from the training set, an over-adjustment process would occur (***overfitting***)
   - For each test example, the class determined by the model is compared with the real (known) class
   - The accuracy ratio is the percentage of test examples that the model correctly classifies

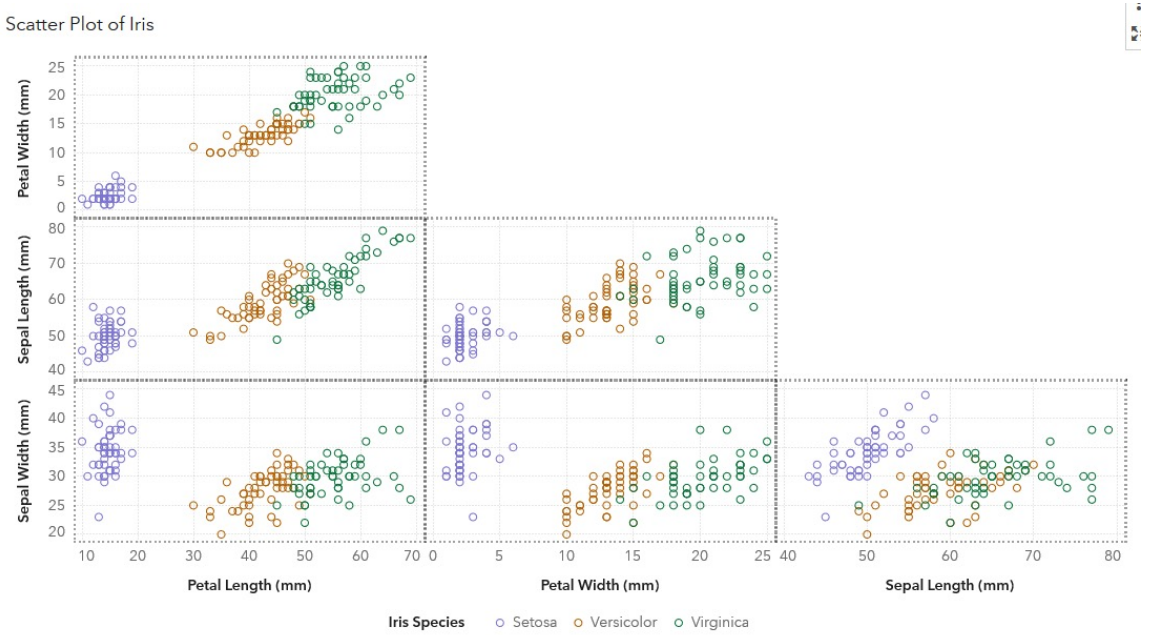
STAGES IN THE CLASSIFICATION PROCESS



CLASSIFICATION EXAMPLE WITH TUPLES

- Example: *Iris*
  - Three clases of iris flowers: *setosa, versicolor* y *virginica*.
  - Four attributes/features/variables: *length and width of petal and sepal* respectively.
  - 150 examples/samples/instances/patterns, 50 of each class/label.
  - Available in http://www.ics.uci.edu/~mlearn/MLRepository.html

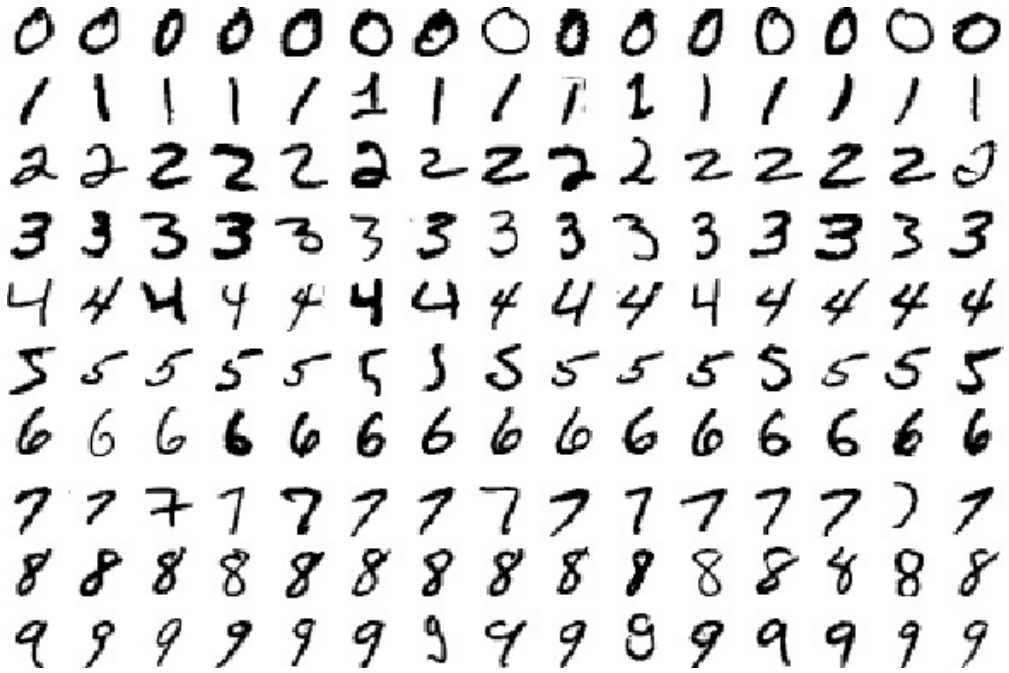




CLASSIFICATION EXAMPLE WITH IMAGES

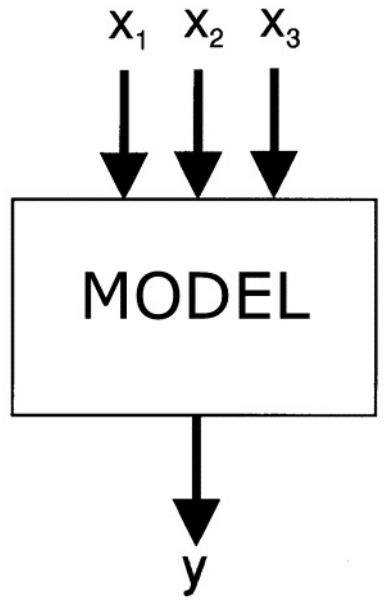
Handwritting recognition.

Assign a digit from 0 to 9.



REGRESSION

- Objective: to predict the numerical value for one variable from the values for others
- The definition of the problem is similar to that of the classification problem:
  - we have predictor variables and
  - a regression variable which in this case is **numerical**
- In regression most (and even all) predictor variables are numerical



REGRESSION ANALYSIS

- Regression analysis is the most commonly used method to perform the numerical prediction task
- **Goal:** to estimate the output variable *(y)* as an equation operating with the remaining inputs variables $(x_1,\ldots,x_n)$
- The simplest model is the **lineal regression** which reduced as a only predictor variable has the form:

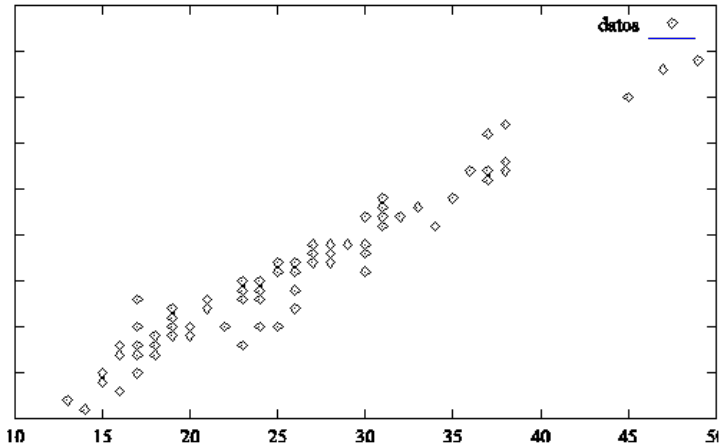
$$y = a + b \cdot x$$

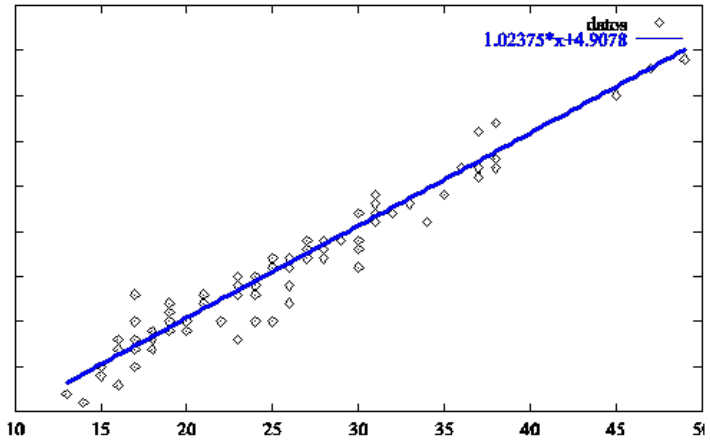
- These coefficients can easily be obtained using the method of least squares

$$b = \frac{\sum_{i=1}^{s}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{s}(x_i - \bar{x})^2}$$

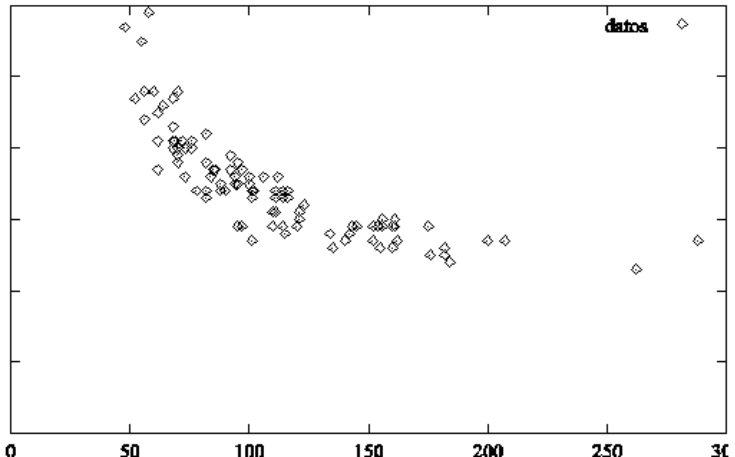
$$a = \bar{y} - b \cdot \bar{x}$$

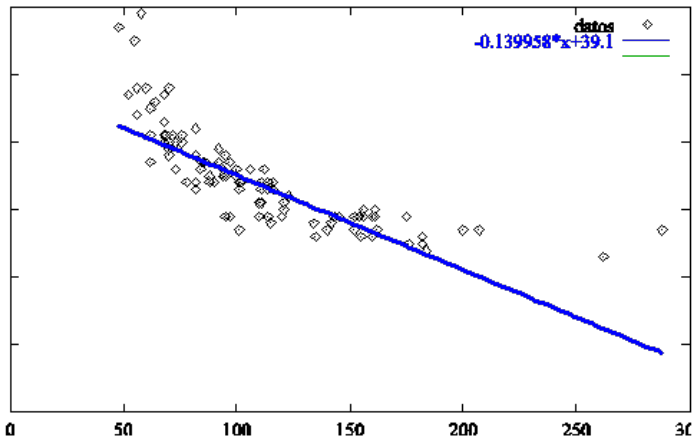
REGRESSION ANALYSIS





Quite appropriate



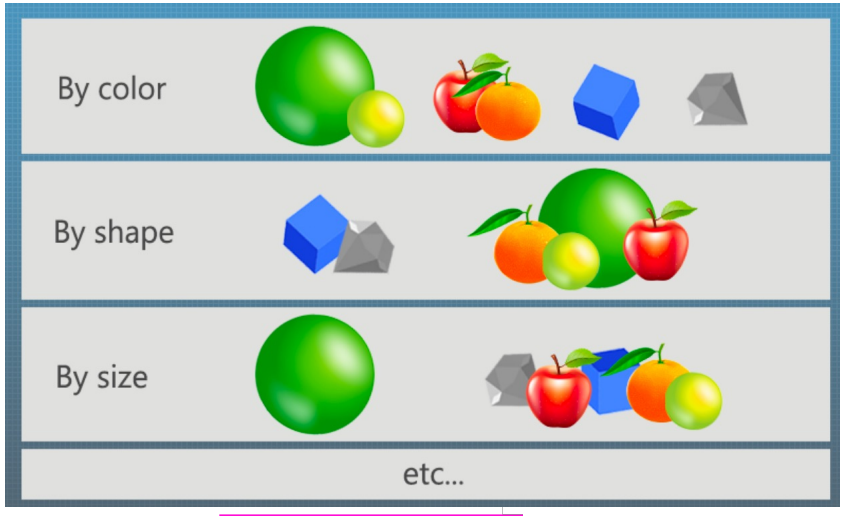


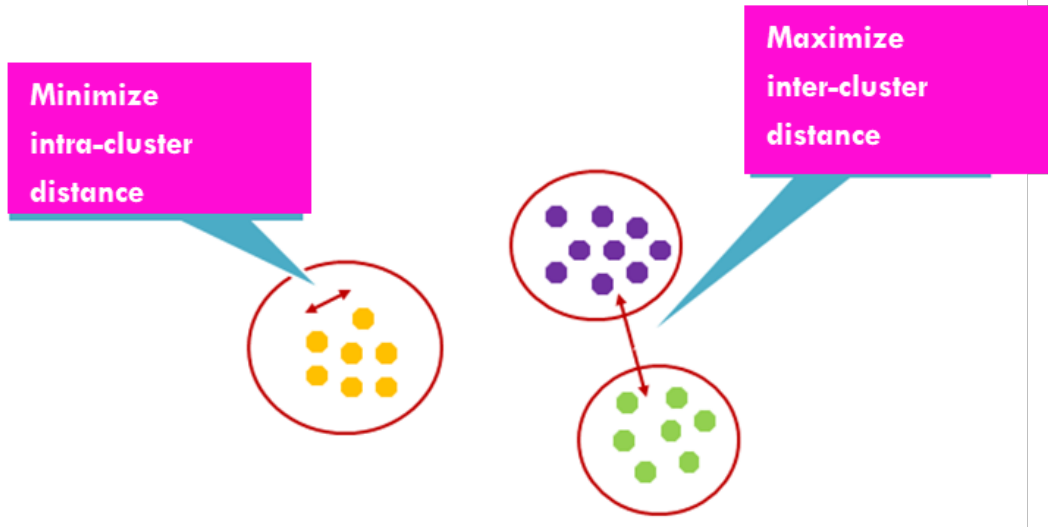
Not so appropriate

CLUSTERING

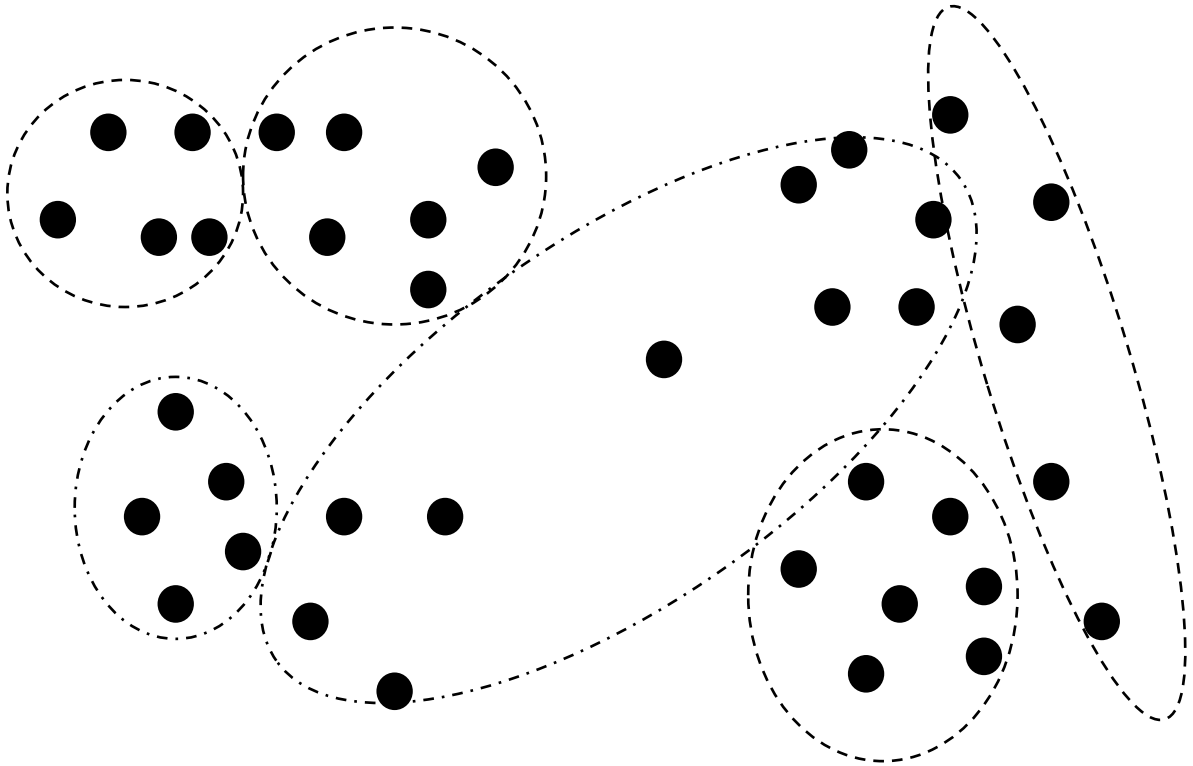
There are problems where we want to group the instances by creating clusters with similar characteristics

E.g. Objects in a backpack





CLUSTERING LEVELS

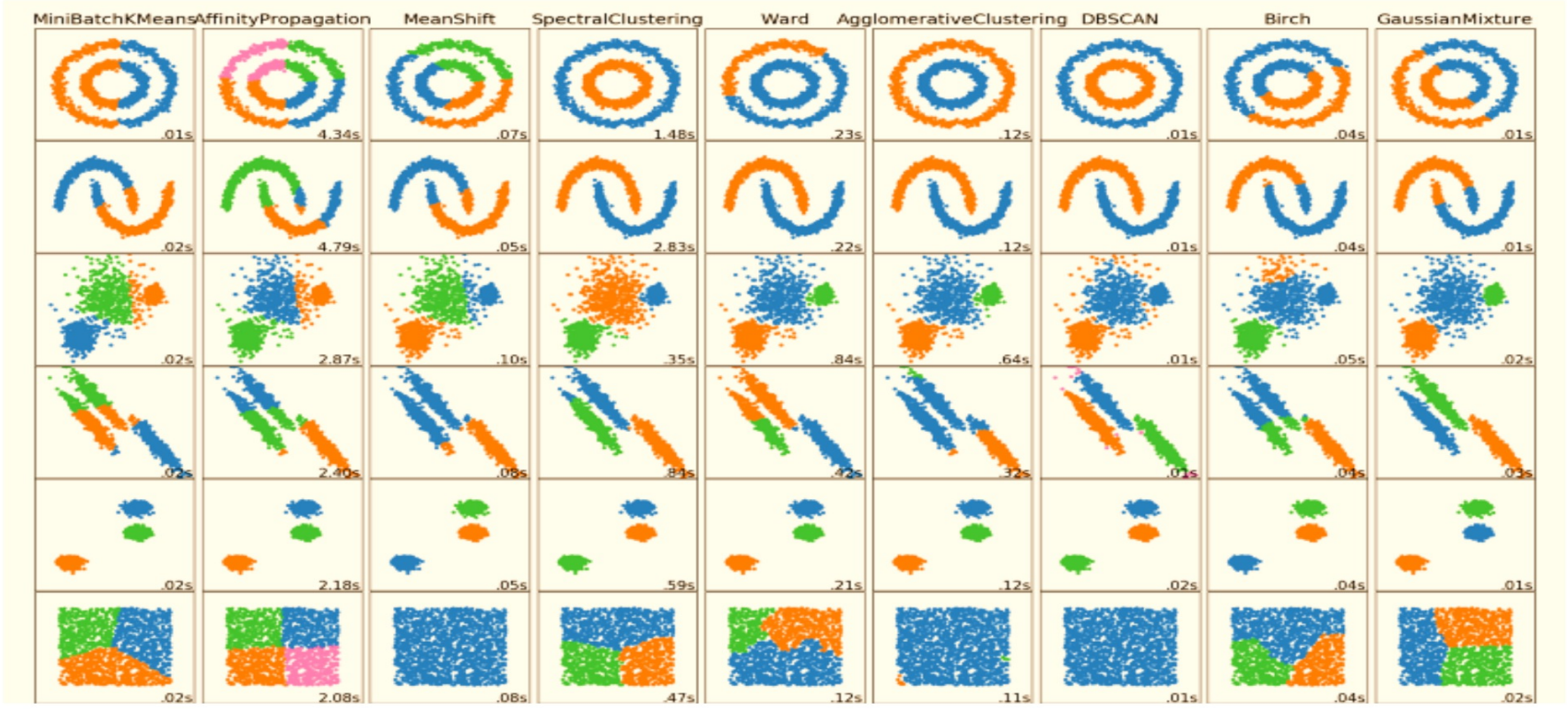


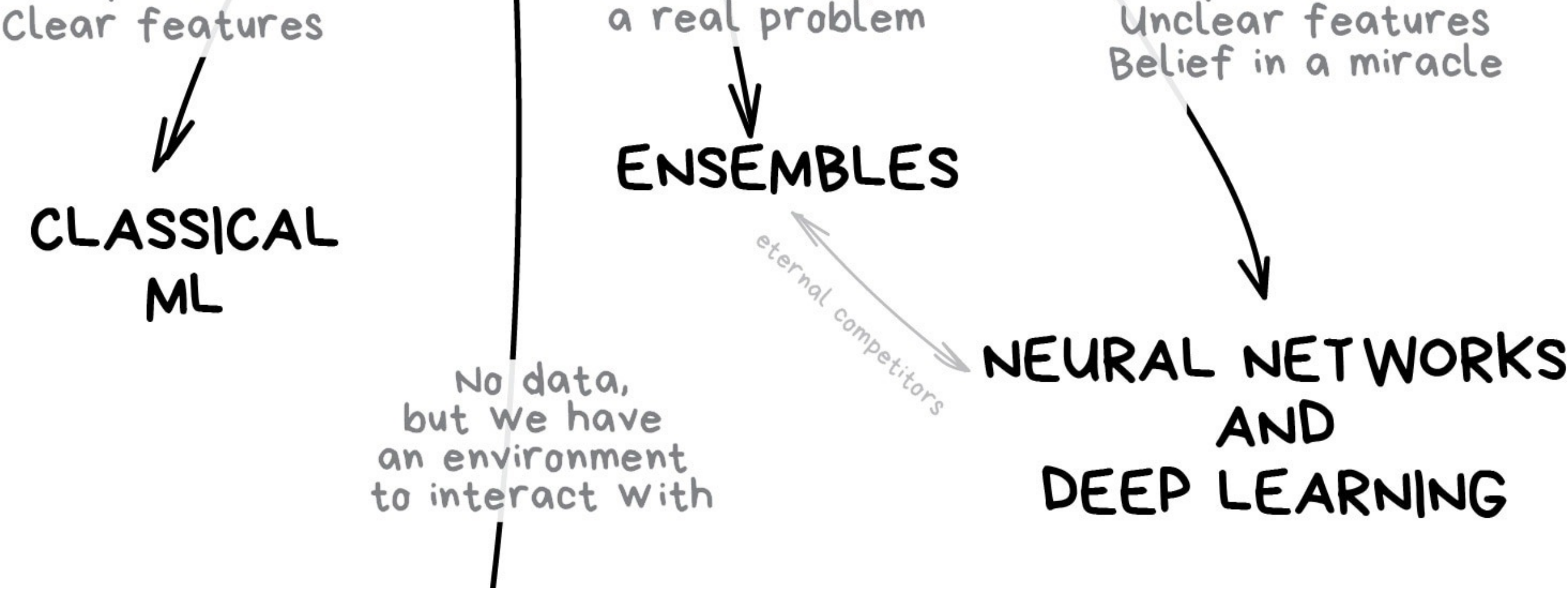
Deciding on the number of clusters
is one of the challenges in clustering

CLUSTERING ALGORITHMS

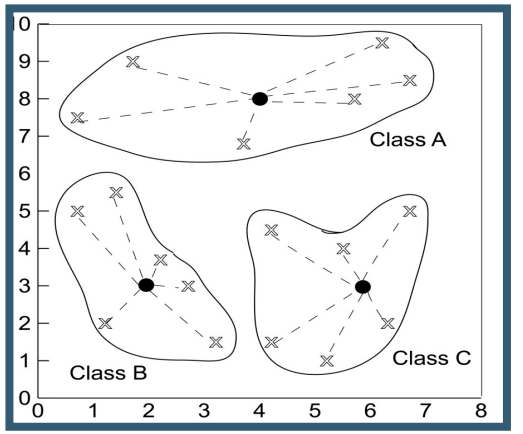
Goal

Finding groupings so that objects in one group are similar to each other and different from objects in other groups [*clusters*].




Clear features
CLASSICAL ML
No data, but we have an environment to interact with
a real problem
ENSEMBLES
eternal competitors
Unclear features
Belief in a miracle
NEURAL NETWORKS AND DEEP LEARNING

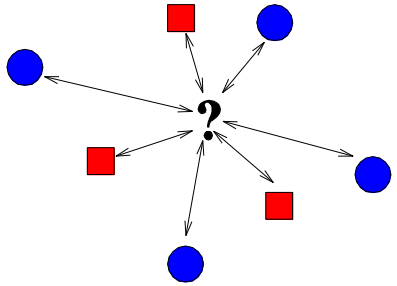
INSTANCE-BASED LEARNING



- They are based on learning by similarity
- They are part of the lazy learning paradigm, as opposed to the voracious one to which the previous paradigms belong
- **Lazy**: Computation is delayed as much as possible
  - No model is built, the model is the BD or training set itself
  - Work is done when a new case arrives for classification: the most similar cases are sought and the classification is constructed according to the class to which these cases belong
- The best known instance-based algorithms are based on the **nearest neighbor rule**

INSTANCE-BASED LEARNING

- The extension to the nearest neighbour rule is to consider the *k* nearest neighbours : k-NN (NN = 1-NN)
- Procedure: Give e as the example to clasify
  1. Select the k examples with $K = \{e_1, \ldots, e_k\}$ such there is no example e' out of *K* with $d(e,e') < d(e,e_i)$, $i=1,\ldots,k$
  2. Return the most repeated class in the set $\{class(e_1), \ldots, class(e_k)\}$ (majority class)
- For instance, if k=7 the next sample (?) will be clasified as ●



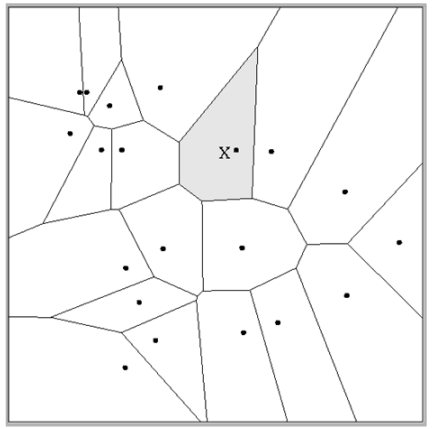
- It could be treated differently from k-neighbours, e.g., depending on the distance to the object to be sorted. In this way we would have:
  - Majority voting ●
  - Voting with distance-dependent weights □

INSTANCE-BASED LEARNING

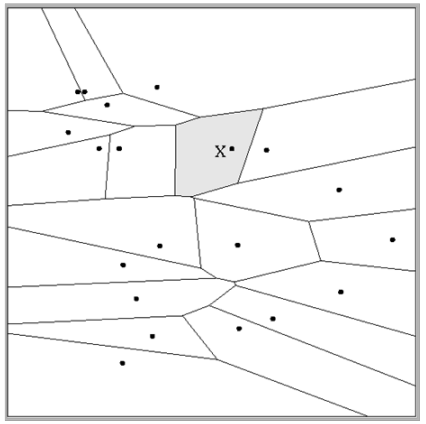
Distance functions for numerical variables

- Numerical variables are usually normalized to the interval [0,1]
- If e_i^j is the value of the j variable in e_i, namely $e_i=(e_i^1,\ldots,e_i^n)$ then some of the most used distances are
  - Euclidean $$d_e(e_1,e_2)=\sqrt{\sum_{i=1}^{n}(e_1^i-e_2^i)}$$
  - Manhattan: $$d_m(e_1,e_2)=\sum_{i=1}^{n}\left|e_1^i-e_2^i\right|$$
  - Minkowski $$d_m^k(e_1,e_2)=\left(\sum_{i=1}^{n}\left|e_1^i-e_2^i\right|^k\right)^{1/k}$$

As we can observe, $d_m^1=d_m$ y $d_m^2=d_e$



$Dist(\mathbf{x}_i,\mathbf{x}_j) = (x_{i1}-x_{j1})^2 + (x_{i2}-x_{j2})^2$



$Dist(\mathbf{x}_i,\mathbf{x}_j) = (x_{i1}-x_{j1})^2 + (3x_{i2}-3x_{j2})^2$

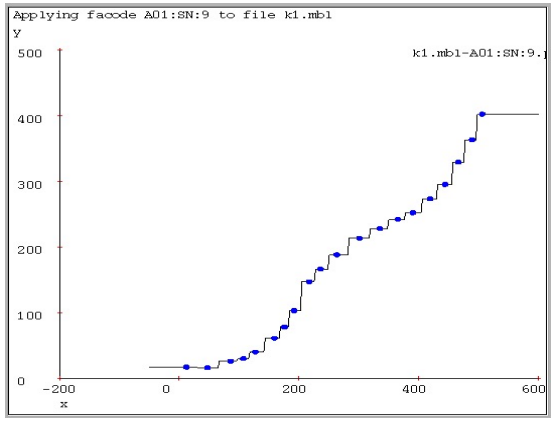
INSTANCE-BASED LEARNING

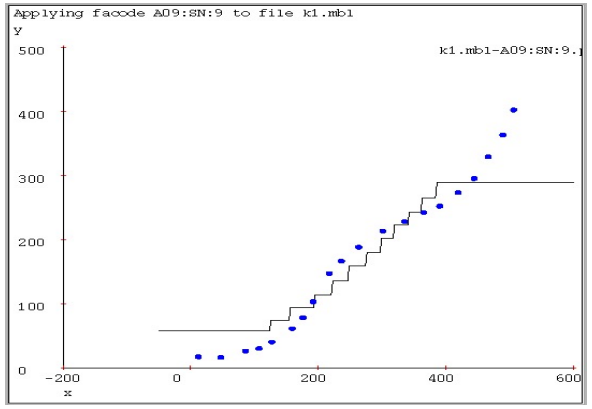
K-NN in Regression

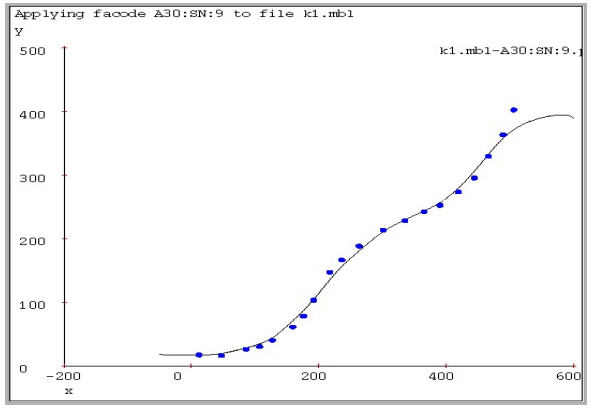
1-NN

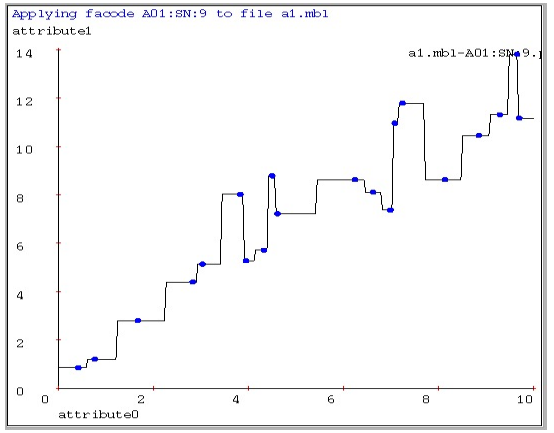
9-NN: Predicts according to the neighborhood average

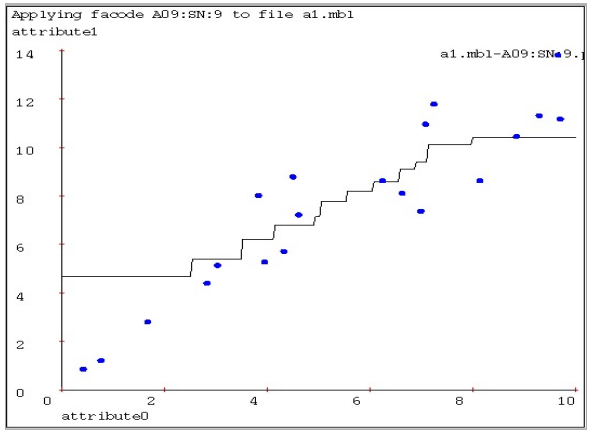
Kernel Regression

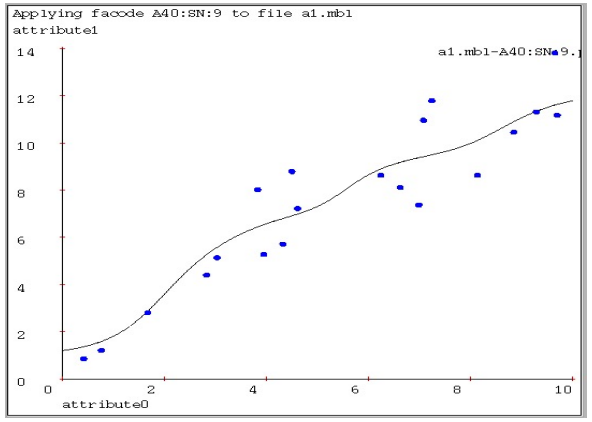












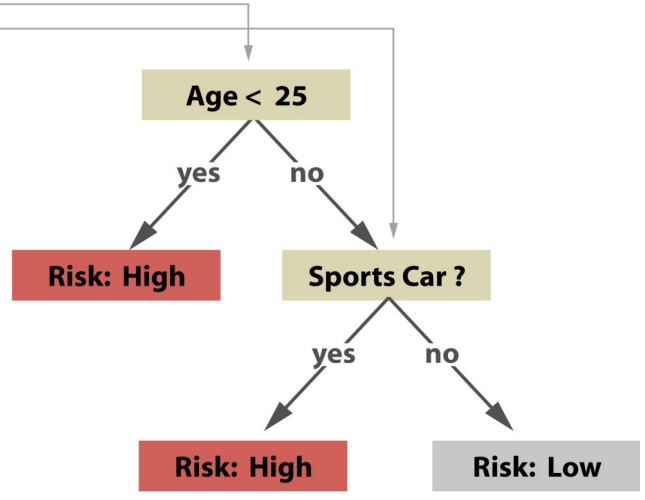
INSTANCE-BASED LEARNING

- The k-NN algorithm is robust to noise when moderate k values are used (k>1)
- It is quite effective, as it uses several local linear functions to approximate the target function
- It is valid for classification and for regression (in the simplest case, returning the average or the distance-weighted average)
- It is very **inefficient in memory** as all the DB has to be stored
- The distance between neighbours could be dominated by irrelevant variables
  - Pre-selection of characteristics: **Feature Selection**
- Its time complexity (to evaluate an example) is $O(dn^2)$ being O(d) the complexity of the distance used
  - One way to reduce this complexity is through the use of prototypes: **Prototype Selection**

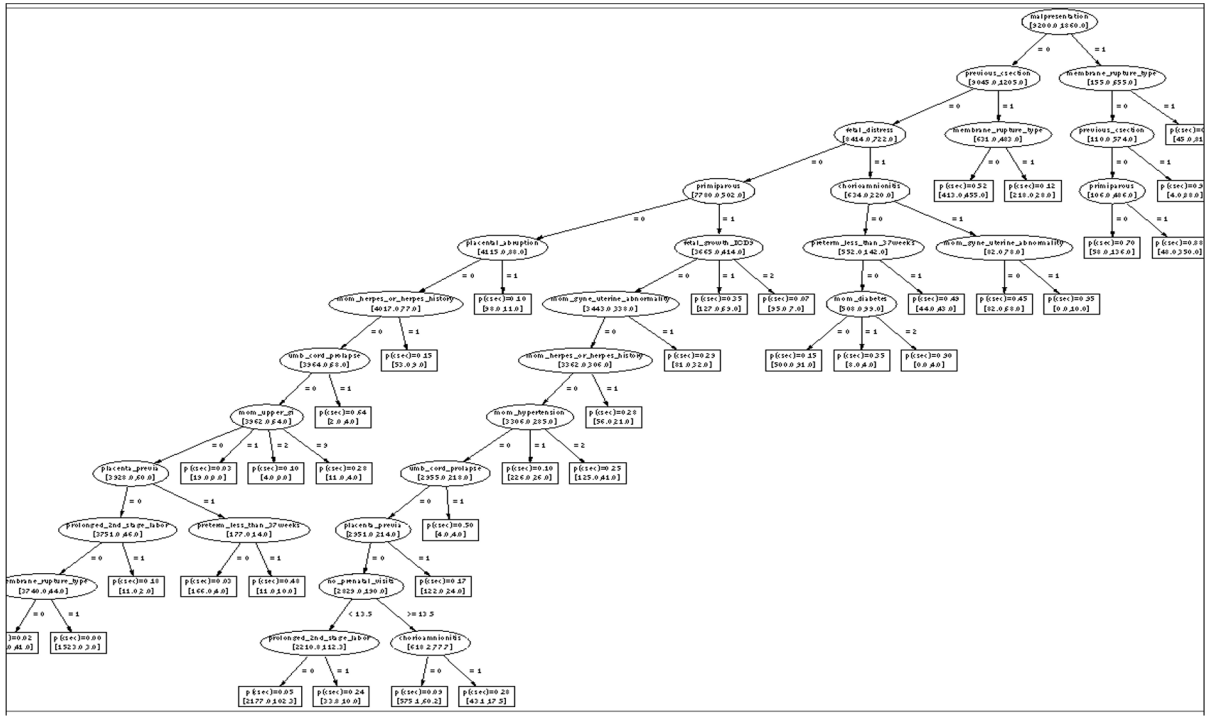
DEFINITION OF DECISION TREES

Insurance Risk Assessment

Age	Car Type	Risk
23	family	High
17	sports	High
43	sports	High
68	family	Low
32	truck	Low
20	family	High



Real Decision Tree



BUILDING DECISION TREES

- The process of generating a decision tree consists of two phases

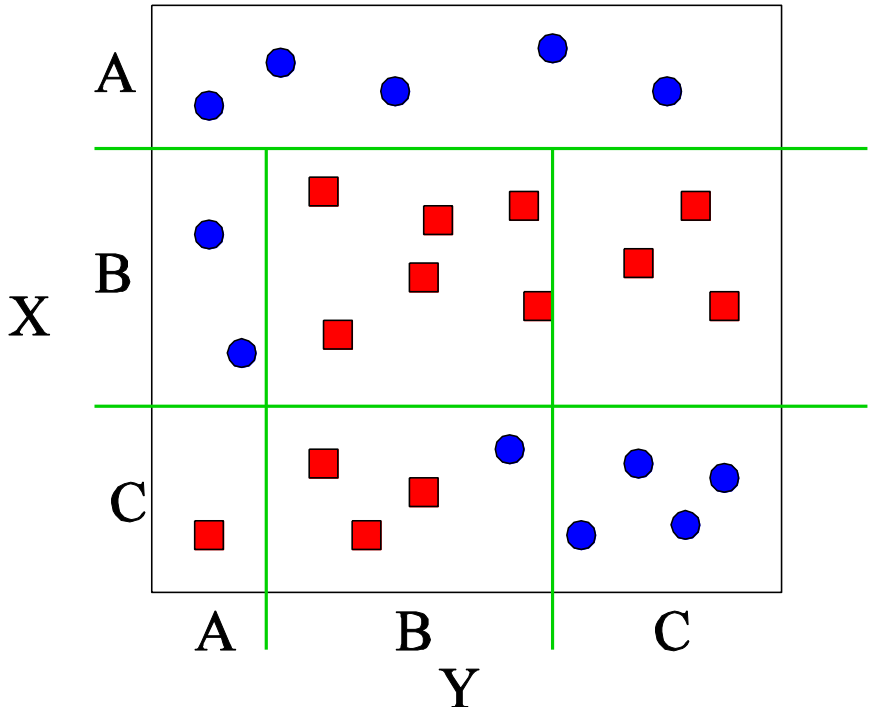
1. Tree building
   - In the beginning all training examples are in the root node
   - The examples are recursively divided according to the selected attributes

2. Tree pruning
   - Identify and remove branches that describe noise or abnormal data

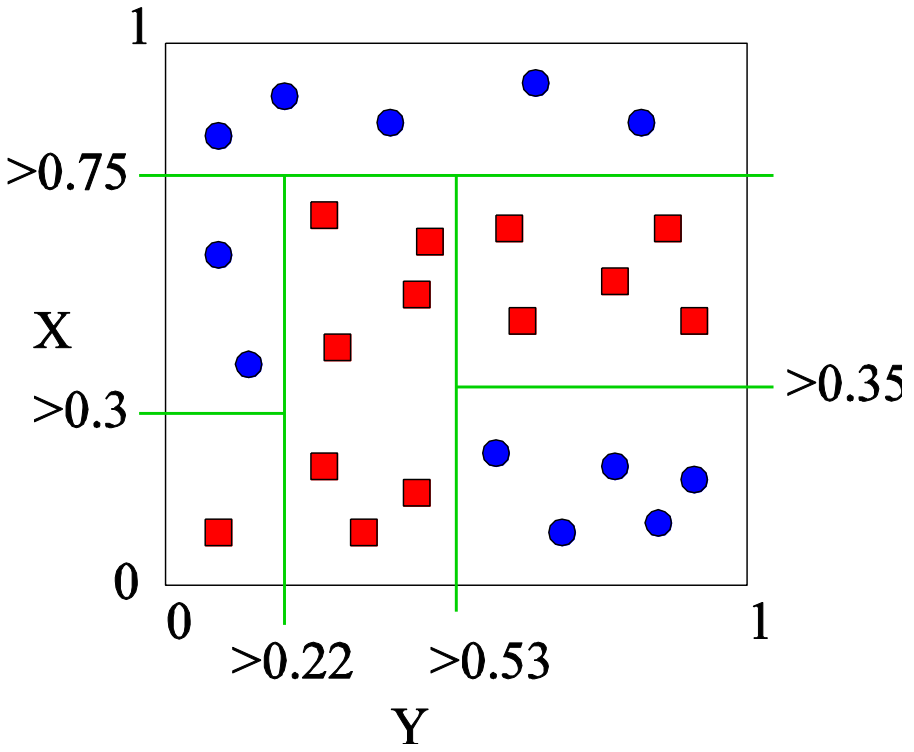
- Using the decision tree: Predicting an unknown example
  - Check the values of the attributes in the example through the decision tree

PARTITIONING THE SPACE IN A DECISION TREE

Trees partition the solution space comprehensively



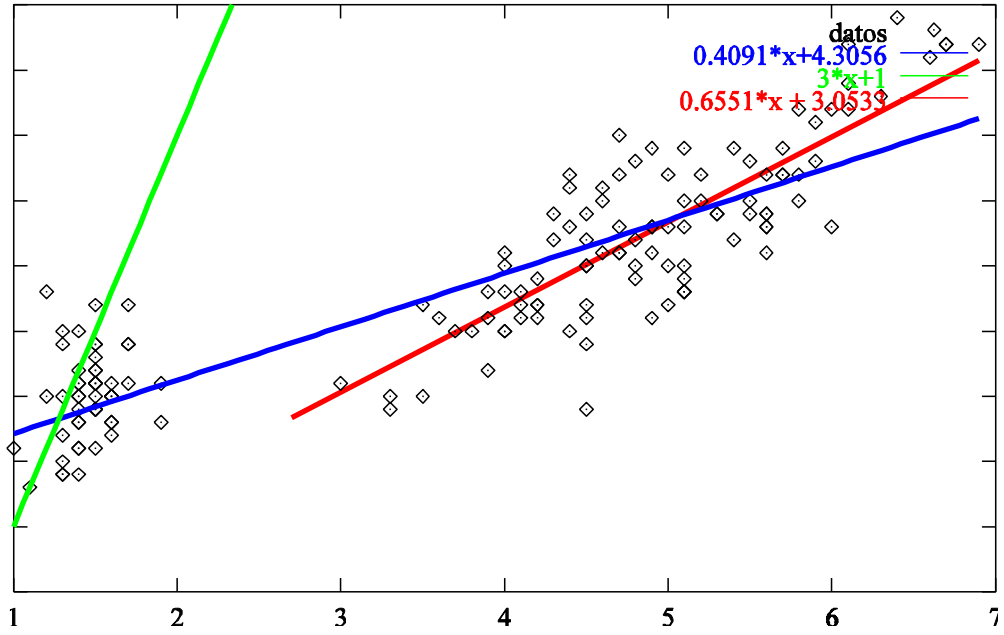
Variables X and Y discrete



Variables X and Y continuous

REGRESSION TREES

- They are trees in which the leaves are simpler regression models, fitted to a local subset of the space



MMAX <= 14000 :
| CACH <= 8.5 : LM1 (108/3.99%)
| CACH > 8.5 : LM2 (33/3.89%)
MMAX > 14000 :
| MMAX <= 22500 : LM3 (37/4.73%)
| MMAX > 22500 : LM4 (31/69.2%)

LM1: class = 15.9 - 0.00453MYCT + 0.00327MMAX
LM2: class = -0.609 + 0.004MMAX + 0.59CACH + 1.57CHMIN
LM3: class = 1.64 - 0.0266MYCT + 0.00485MMIN + 0.00346MMAX + 0.627CACH + 1.43CHMIN + 0.127CHMAX
LM4: class = -350 - 0.843MYCT + 0.0183MMAX + 1.62CACH

EMR 10.0402
EAR 11.4899 %

ADVANTAGES AND DISADVANTAGES OF USING DECISION TREES

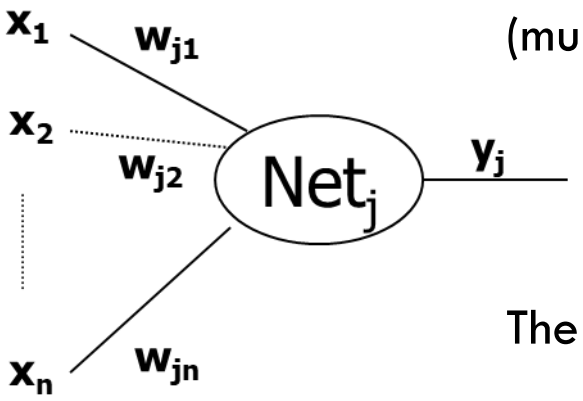
Advantages:

- Decision trees are easy to use and efficient
- The rules they generate are easy to interpret
- They scale better than other types of techniques
- They treat noise data well

Disadvantages:

- They do not easily handle continuous attributes
- They try to divide the domain of attributes into rectangular regions and not all problems are of that type
- Have difficulty working with missing values
- They may have problems of overfitting
- Thet don not detect correlations among attributes

NEURAL NETWORKS: ARTIFICIAL NEURONS



- The signals that reach the dendrites are represented as x_1, x_2, ..., x_n.

- The synaptic connections are represented by weights w_{j1}, w_{j2}, w_{jn} that weight (multiply) the entries. If the weight between neurons j and i is :

  a) positive, it represents an excitatory synapse

  b) negative, represents an inhibitory synapse

  c) zero, no connection

The integrative action of the cell body (or internal activity of each cell) is presented by

$$Net_j = w_{j1} \cdot x_1 + w_{j2} \cdot x_2 + \ldots + w_{jn} \cdot x_n = \sum_{i=1}^{n} w_{ji} \cdot x_i$$

- The output of the neuron is represented by y_j. It is obtained through a function that is generally called output, transfer or activation function. This function depends on Net_j and a parameter j which represents the activation threshold of the neuron

$$y_j = f(Net_j - \theta_j) = f\left(\sum w_{ji} \cdot x_i - \theta\right)$$

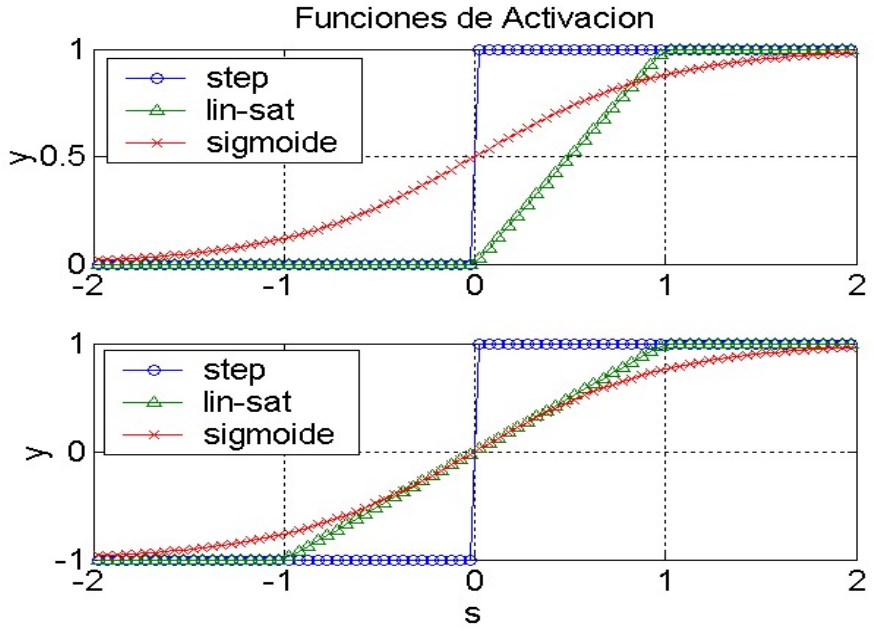
TYPES OF TRANSFER FUNCTIONS

- **Step function** or Haviside. It represents a neuron with only two states of activation: activated (1) and inhibited (0 ó -1)

$$y_j = H(Net_j - \theta_j) = \begin{cases} 1, & si\ Net_j >= \theta_j \\ -1, & si\ Net_j < \theta_j \end{cases}$$

- **Linear function**: $y_j = Net_j - \theta_j$

- **Linear function in sections**: $y_j = \begin{cases} 1, & si\ Net_j >= \theta_j + a \\ Net_j - \theta_j, & si\ |Net_j - \theta_j| < a \\ -1, & si\ Net_j < \theta_j - a \end{cases}$



- **Sigmoidal function**:

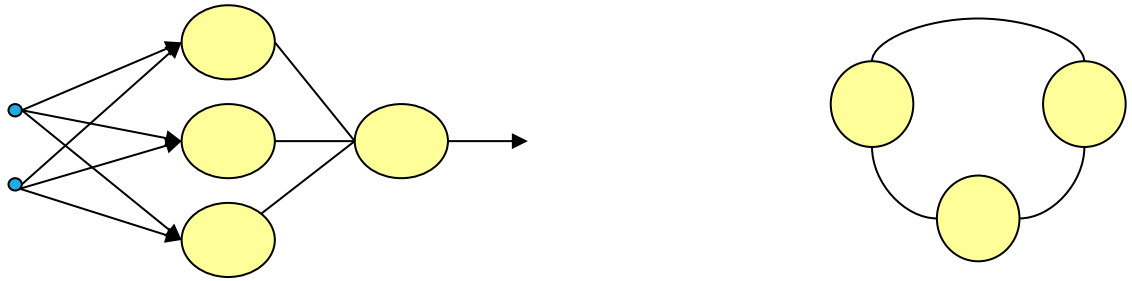
$$y_j = \frac{1}{1 + e^{-\lambda(Net_j - \theta_j)}} \qquad y_j = \frac{2}{1 + e^{-\lambda(Net_j - \theta_j)}} - 1$$

- **Radial-basis function**: $y_j = e^{-\left(\frac{Net_j - \theta_j}{\sigma}\right)^2}$

¿WHAT IS A NEURAL NETWORK?

An artificial neuron network is characterized by its :

- **Architecture:** Structure or pattern of connections between process units



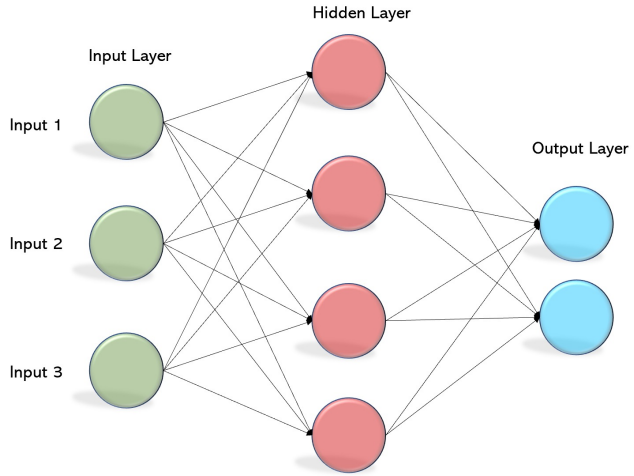
- **Computational dynamics** which expresses the value taken by the processing units and which is based on activation (or transfer) functions that specify how the input signals from the processing unit are transformed into the output signal
- **Training or Learning Algorithm:** Procedure for determining the weights of the connections. A very important feature of these networks is their adaptive nature, where 'learning by example' replaces 'programming' in problem solving.

MULTI-LAYER PERCEPTRON (MLP) IN CLASSIFICATION

Learning the structure of a neural network requires experience, although there are some guidelines.

1. **Inputs:** For each numerical or binary/boolean variable an input neuron is set. For each nominal variable (with more than two states) an input neuron is put for each possible state.
2. **Outputs:** If it is for clasification, a single output neuron is placed for each value of the class variable. For regression, only a output neuron is needed.
3. **Hidden layers**. You have to indicate how many layers and how many neurons have to be put in each layer.

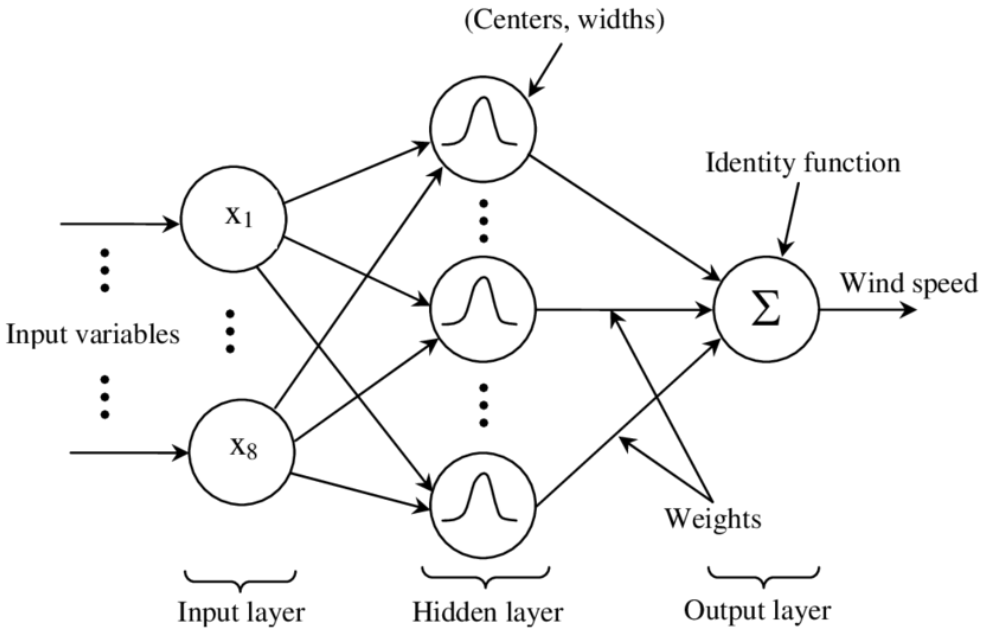
- When the neural network is trained, to predict an instance, the values of the same are introduced corresponding to the variables of the input nodes:
  - The output of each output node indicates the probability that the instance belongs to that class
  - The instance is assigned to the class with the highest probability.
- All numerical variables are standardised [-1,1].
- **Backprogation algorithm:**
  - Based on the descending gradient technique
  - Does not allow backward connections (feedback) in the network



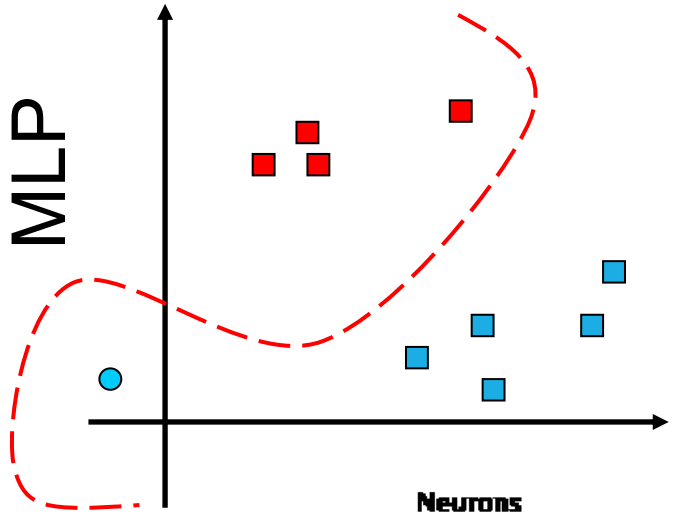
RADIAL-BASIS FUNCTIONS NETWORKS (RBFNS)

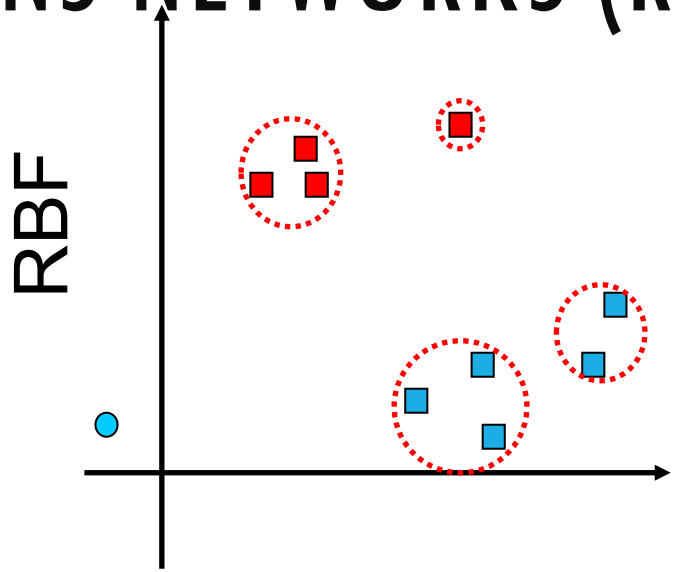
- Designing a neural network as a curve fitting problem → gaussian local mixture
- Learning: Finding the surface in multidimensional space that best fits the training data
- Generalization: Use this multidimensional surface to **interpolate test data**

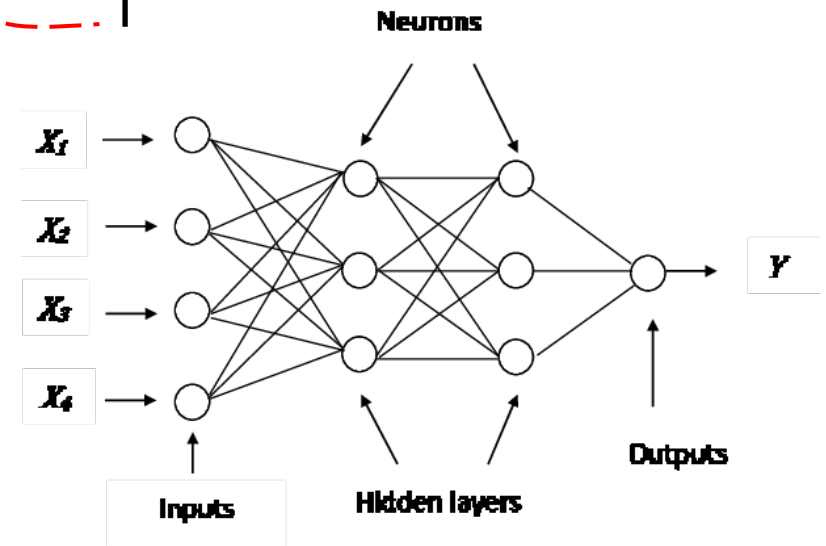
- **Three Layers:**
  1. Input layer
     - Source nodes that connect the network to your environment
  2. Hidden layer
     - Hidden units provide a set of basic functions
     - High dimensionality
  3. Output layer
     - Linear combination of hidden functions



RADIAL-BASIS FUNCTIONS NETWORKS (RBFNS)







Approximate function with linear combination of radially based functions

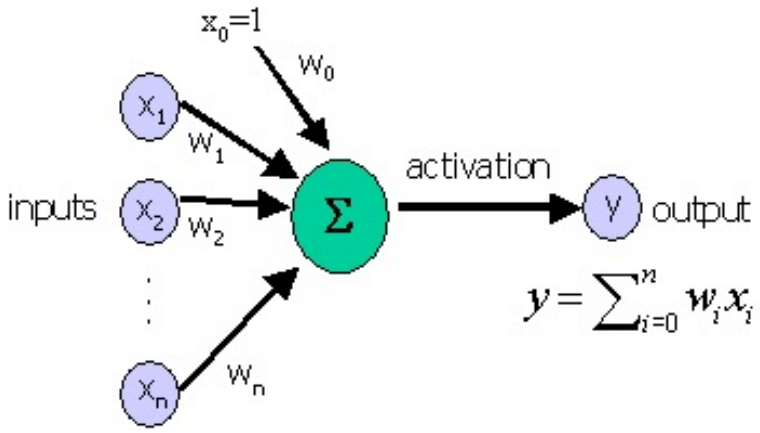
$$f(x) = \sum w_j h_j(x)$$

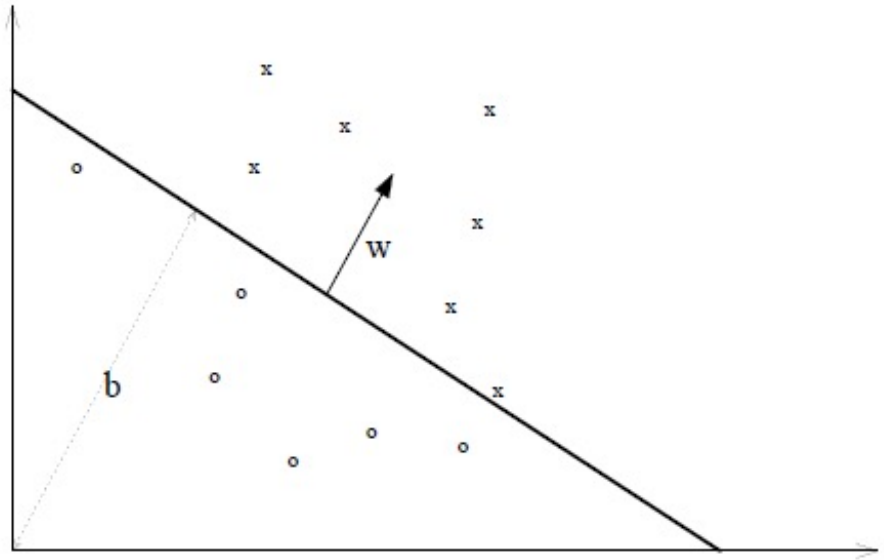
$$h_j(x) = \exp(-(x-c_j)^2 / r_j^2)$$

Where c_j is center of a region,
r_j is width of the receptive field

SUPPORT VECTOR MACHINES (SVMS)

- A perceptron represents a hyperplane in space: is a Linear Learning Machine





$$\langle w, x \rangle + b = 0$$

- For non linear boundaries, neural networks combine perceptrons

SVM DUAL REPRESENTATION

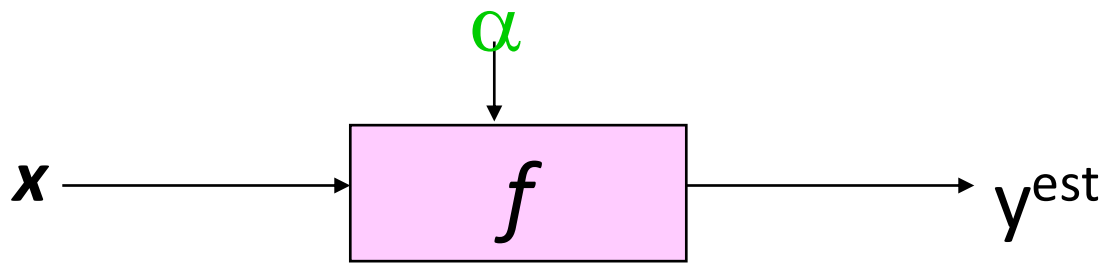
- The decision function can be rewritten as follows:

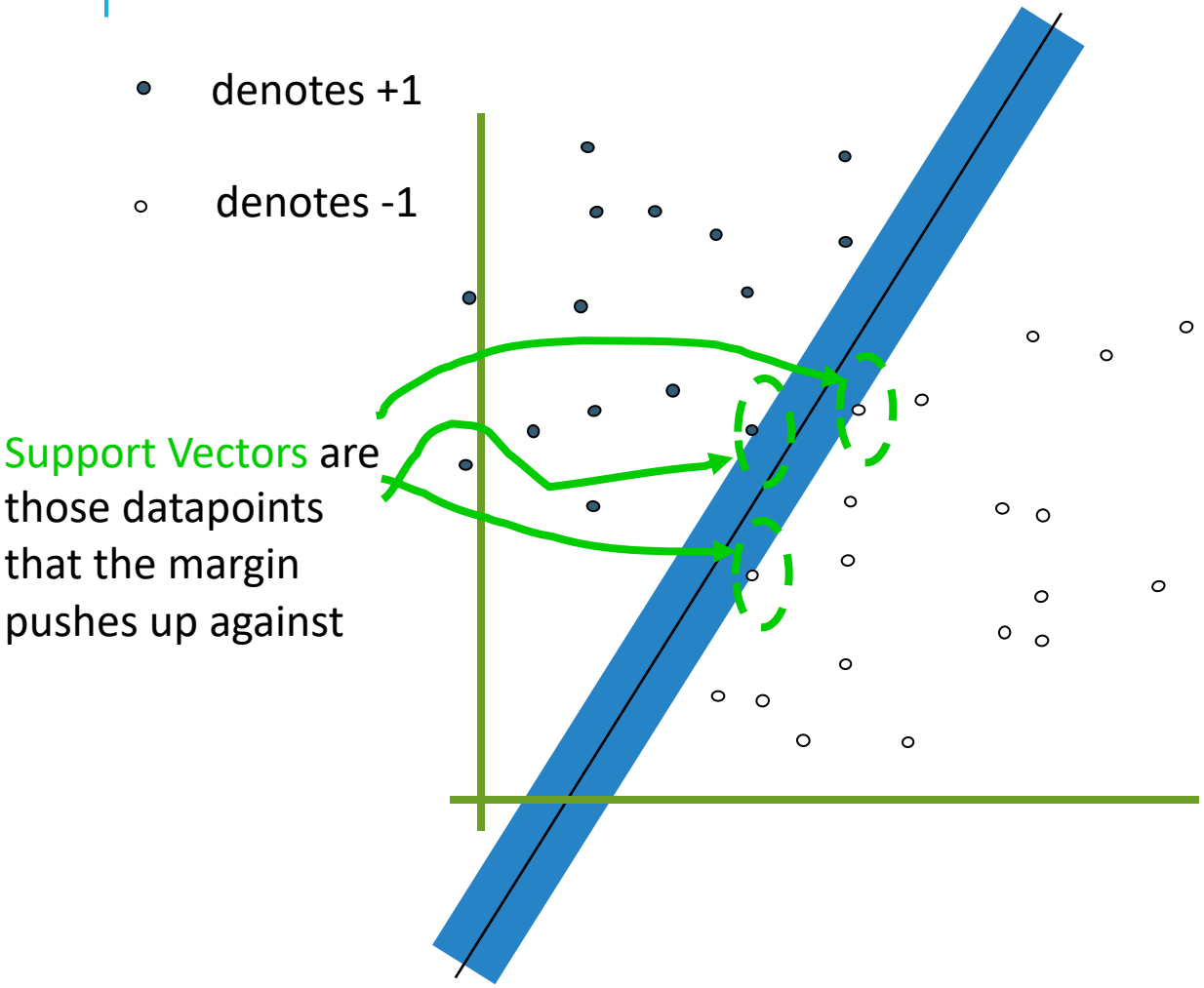
$$f(x) = \langle w, x \rangle + b = \sum \alpha_i y_i \langle x_i, x \rangle + b$$

$$w = \sum \alpha_i y_i x_i$$

- It is very important because it allows us to **define the decision function from the inputs**
- SVMs are Linear Learning Machines (LLMs) represented in a dual form
- The input data only acts on scalar products:
  - In the decision function
  - In the training algorithm

MAXIMUM MARGIN LINEAR CLASSIFIERS





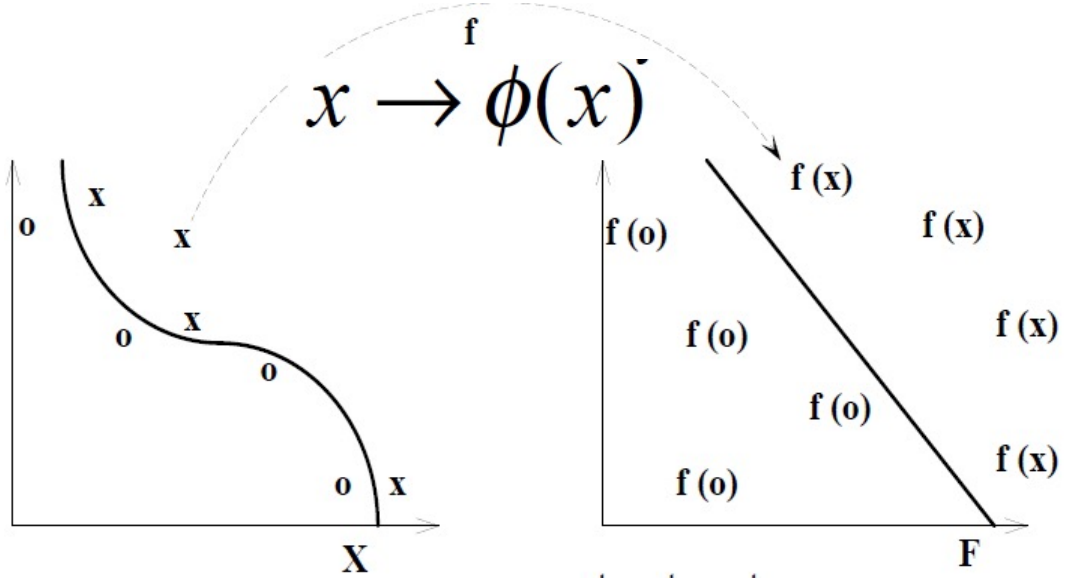
$f(x,w,b) = sign(w.x - b)$

The maximum margin linear classifier is the linear classifier with the, um, maximum margin. This is the simplest kind of SVM (Called an LSVM)

Linear SVM

KERNEL FUNCTIONS

- Map the data in a more complex space, with non-linear characteristics, and use a linear classifier in this space
- **Idea!** → we will move the data to another feature space where they are linearly separable



$$f(x) = \sum \alpha_i y_i \langle \phi(x_i), \phi(x) \rangle + b$$

Polinomial $K(x,z) = \langle x,z \rangle^d$

RBF $K(x,z) = e^{-\|x-z\|^2/2\sigma}$

EFFECTS OF PARAMETERS IN SVM

SVMs are **VERY SENSITIVE** to the parameters

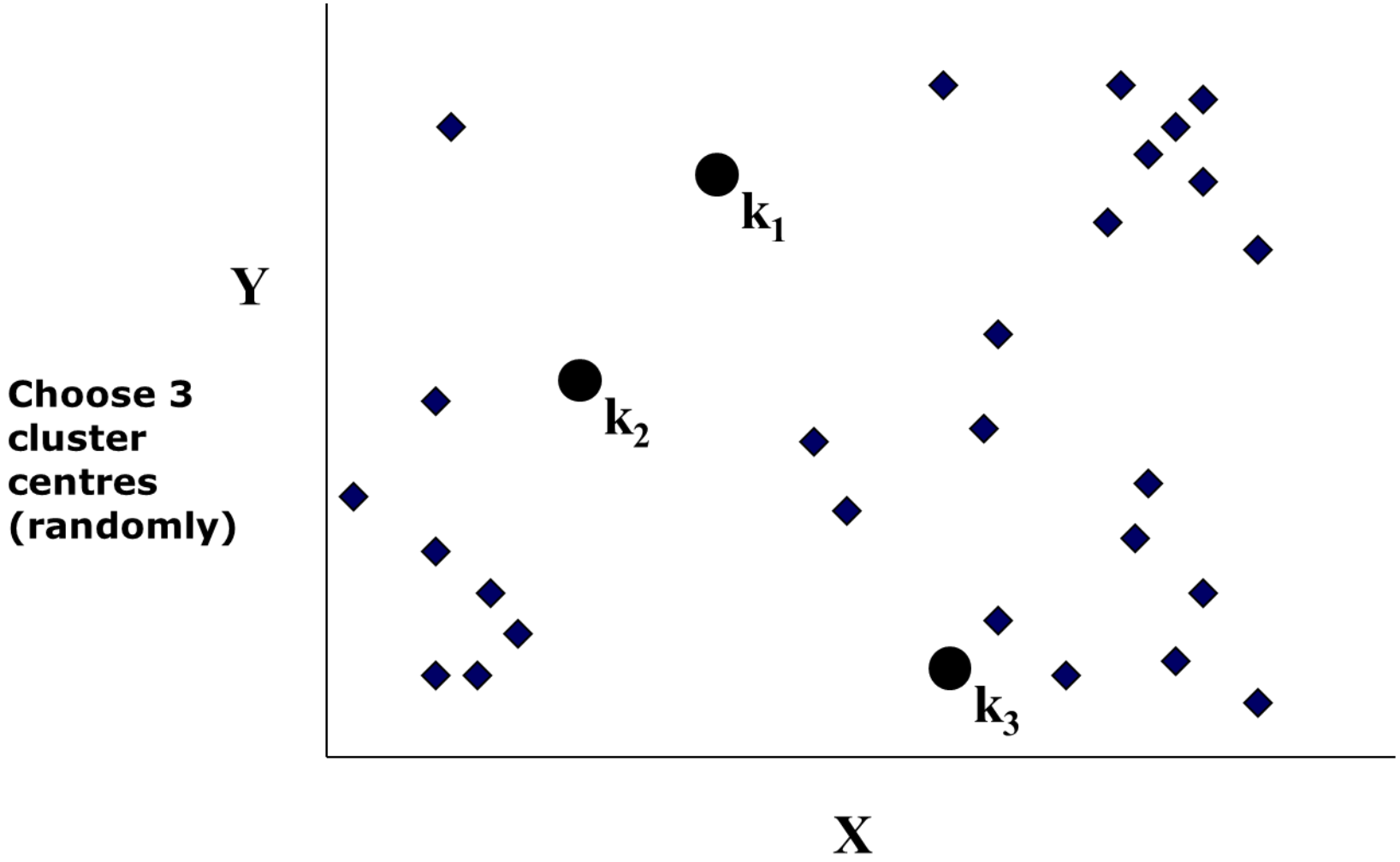
- The value of C changes the behaviour of the SVM:
  - A low value makes the SVM more permissive of outliers
  - A high value makes the SVM less permissive with strange examples
- High values of C → weights α other than zero for more examples (increases the number of support vectors) and lower average values (less significant support vectors)
- Why? Wrong classifications have a higher cost if C is higher ← **overfitting danger**

CLUSTERING: K-MEANS

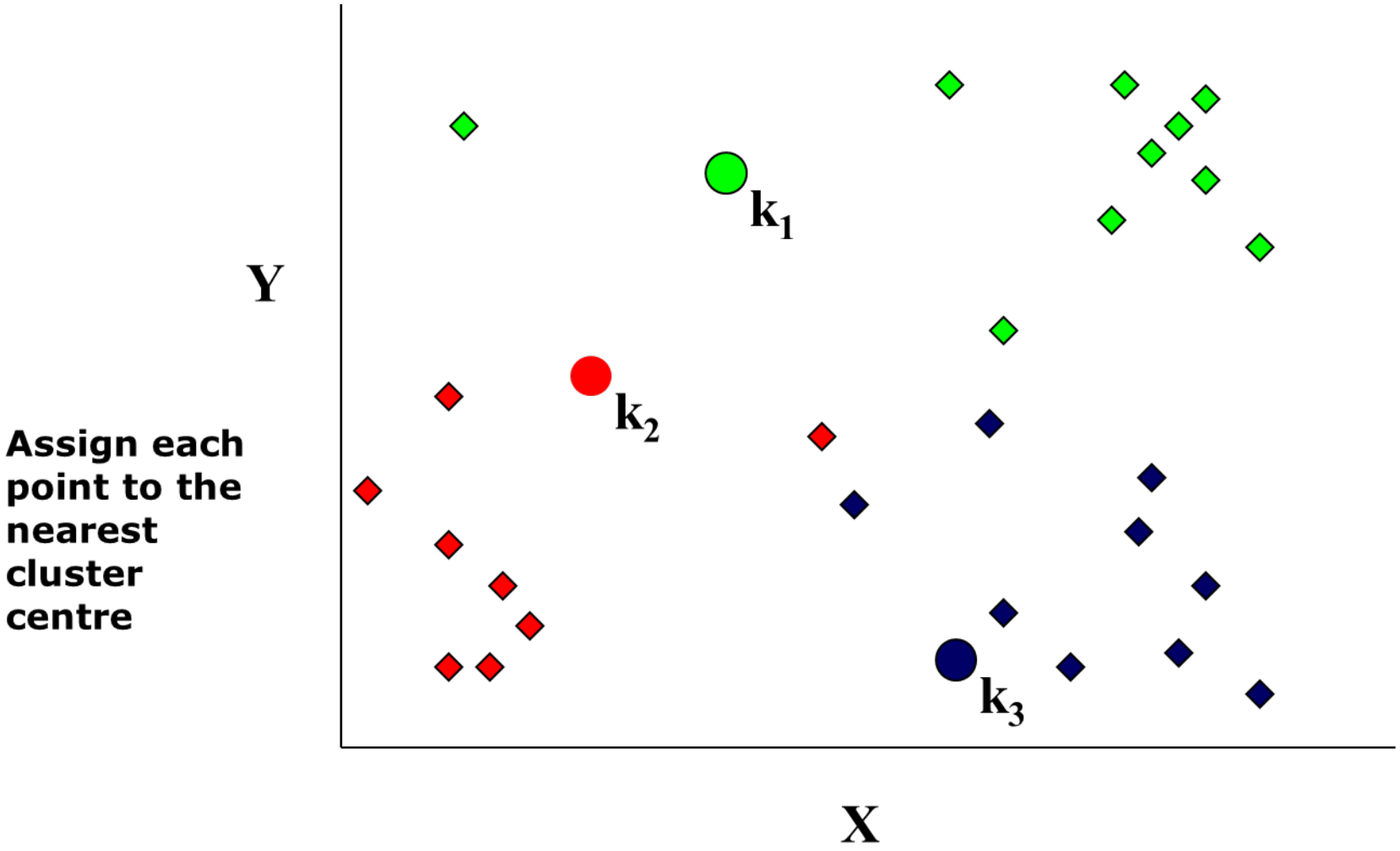
- It needs as an input parameter the desired number of clusters

- It is an iterative algorithm in which instances are moved between clusters until the desired set of clusters is reached

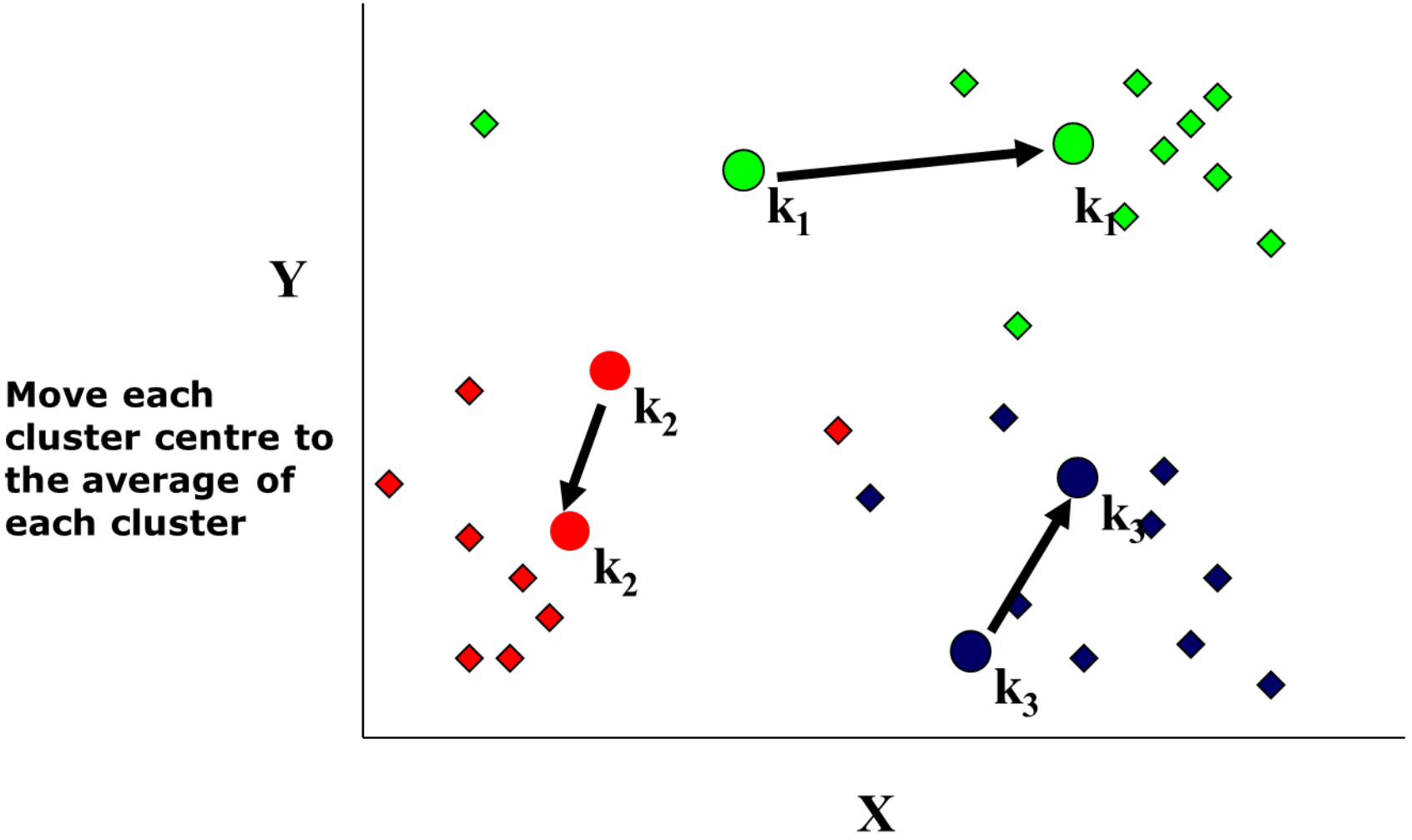
K-MEANS EXAMPLE



K-MEANS EXAMPLE



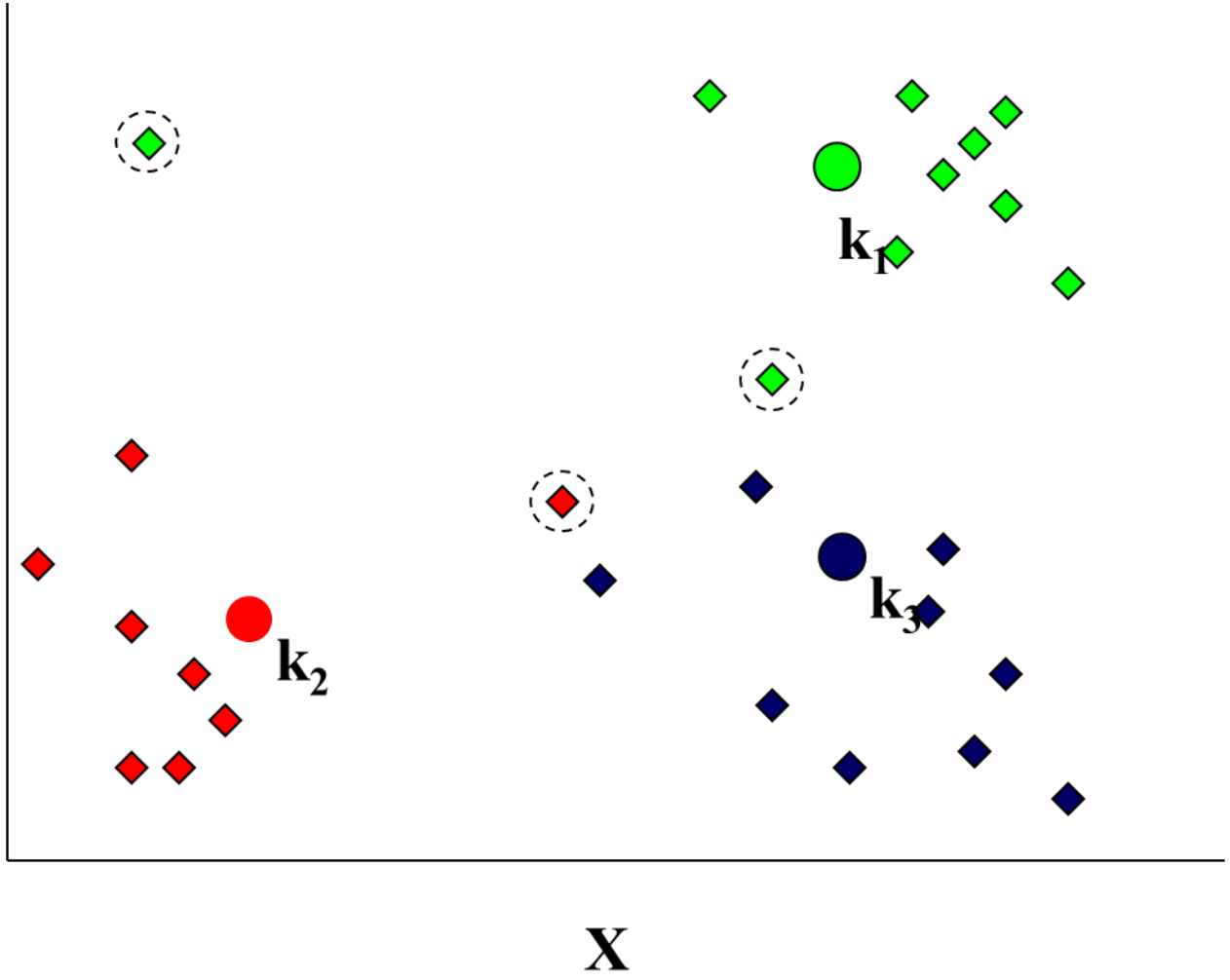
K-MEANS EXAMPLE



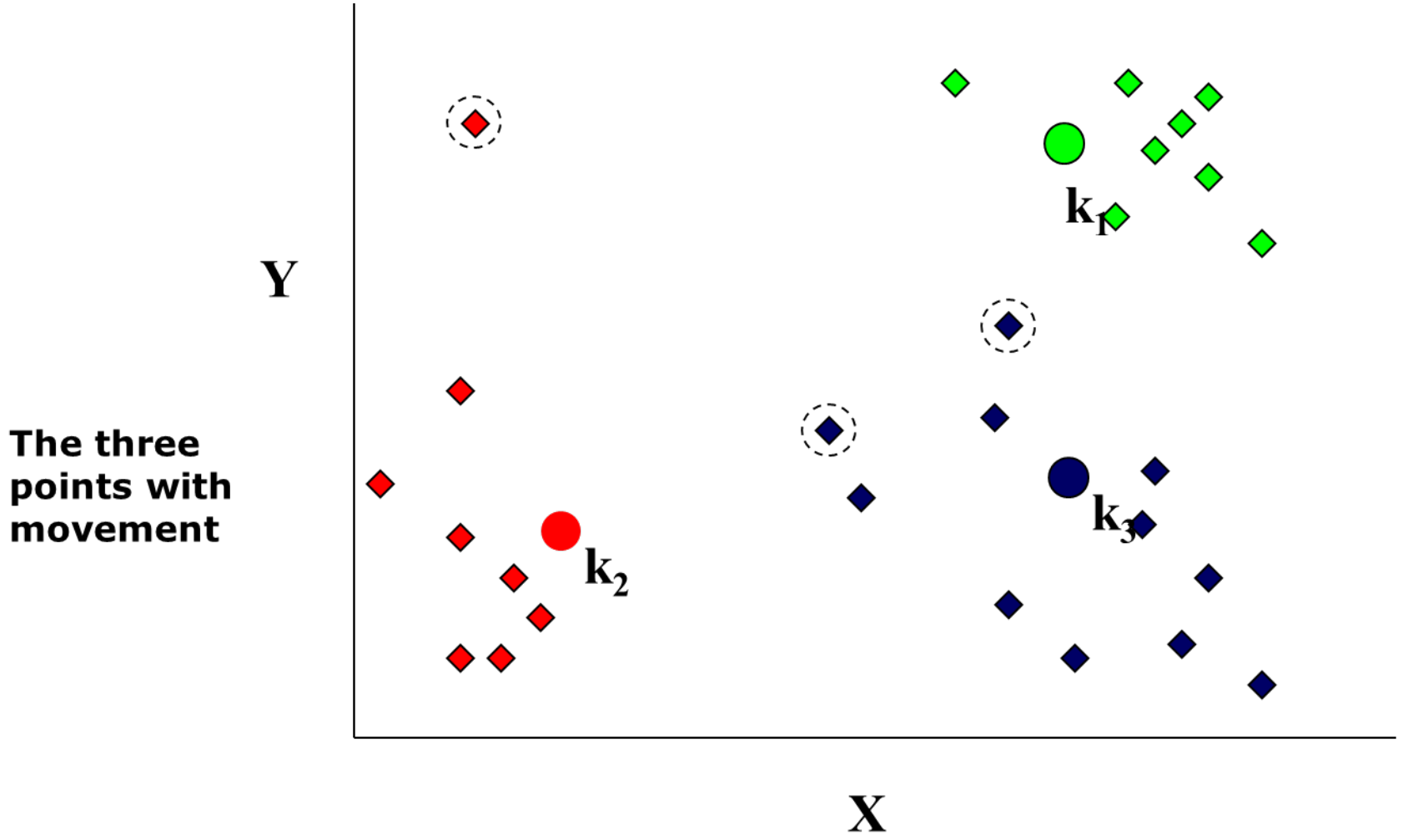
K-MEANS EXAMPLE

Reassign the points closest to different cluster centres

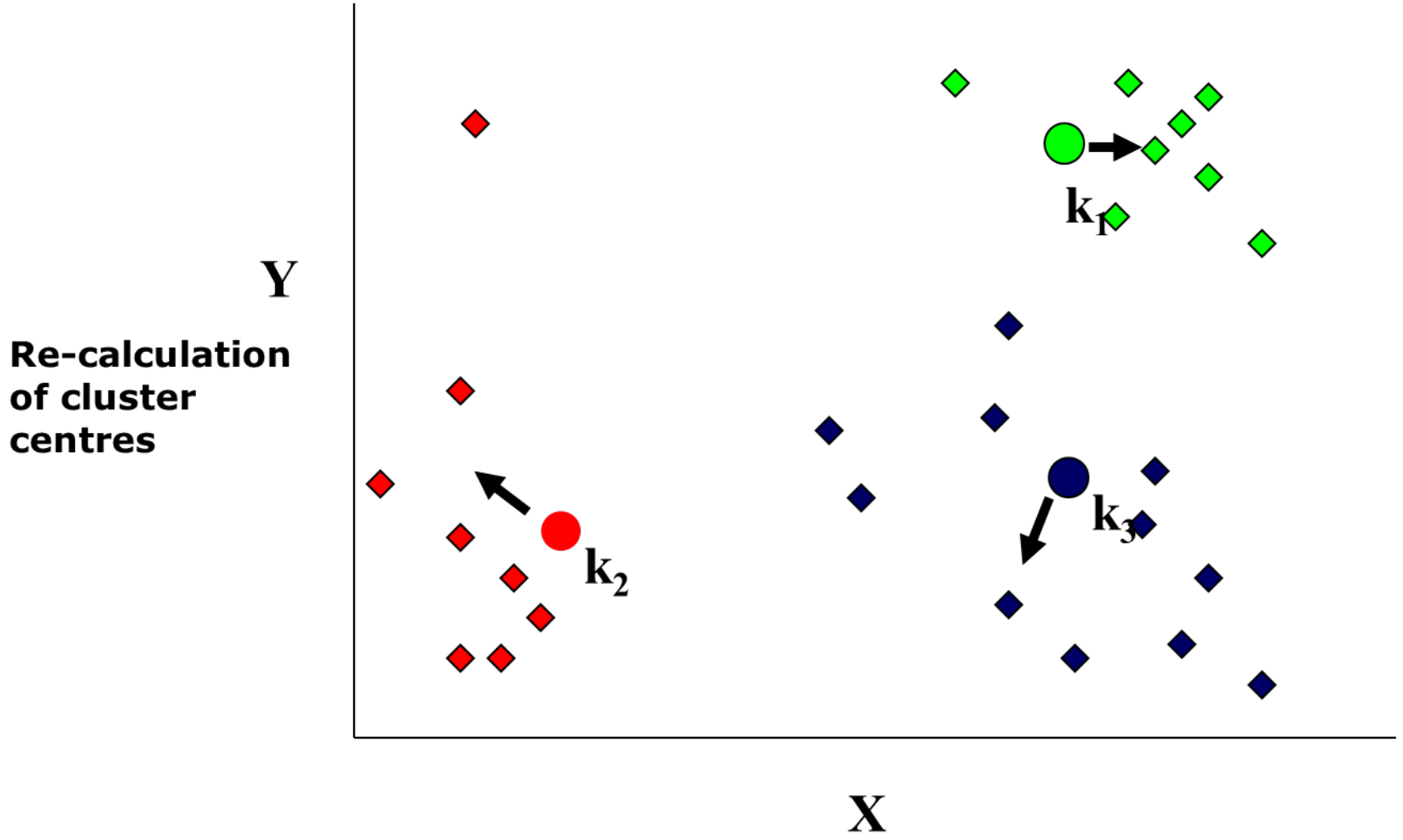
What points are reassigned?



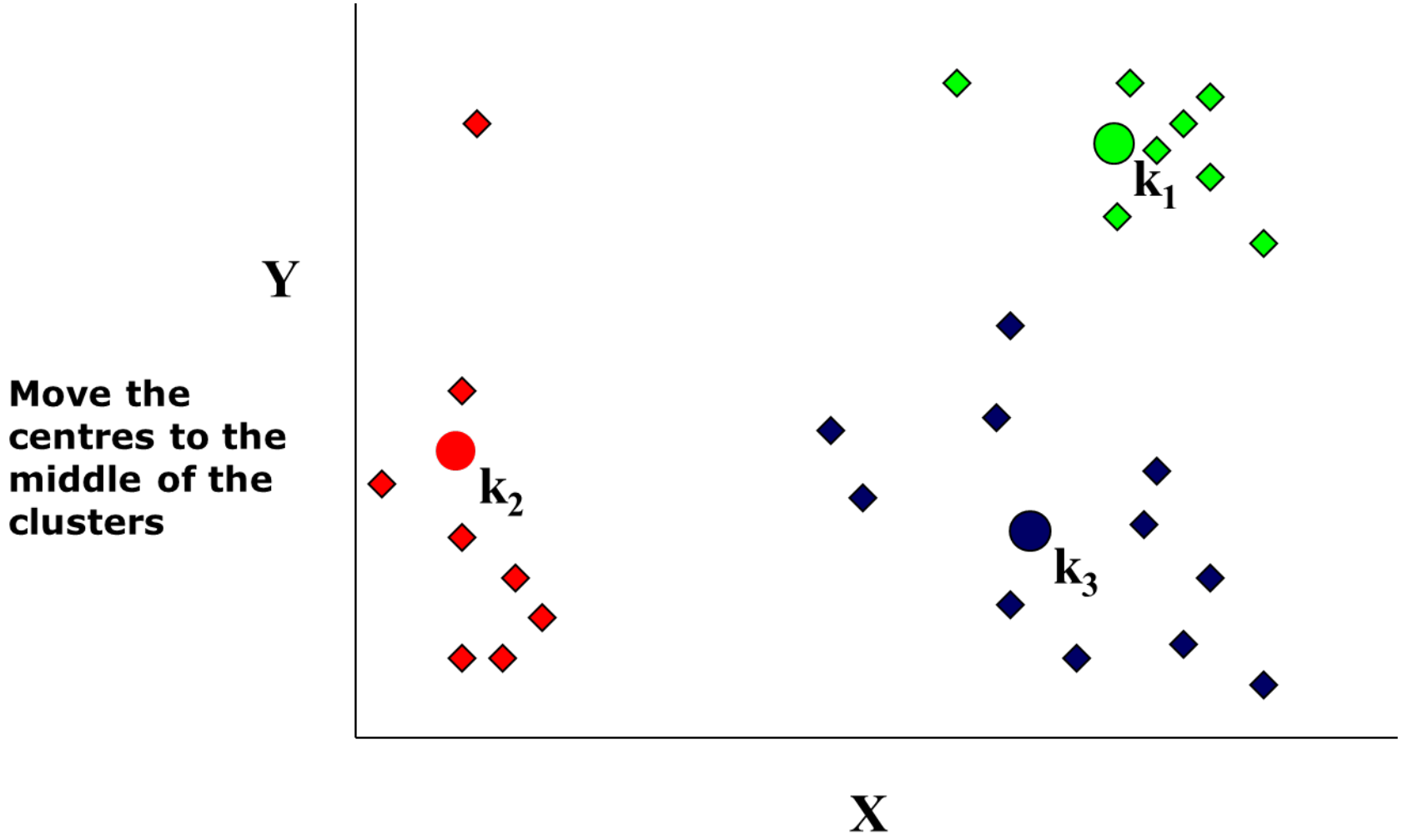
K-MEANS EXAMPLE



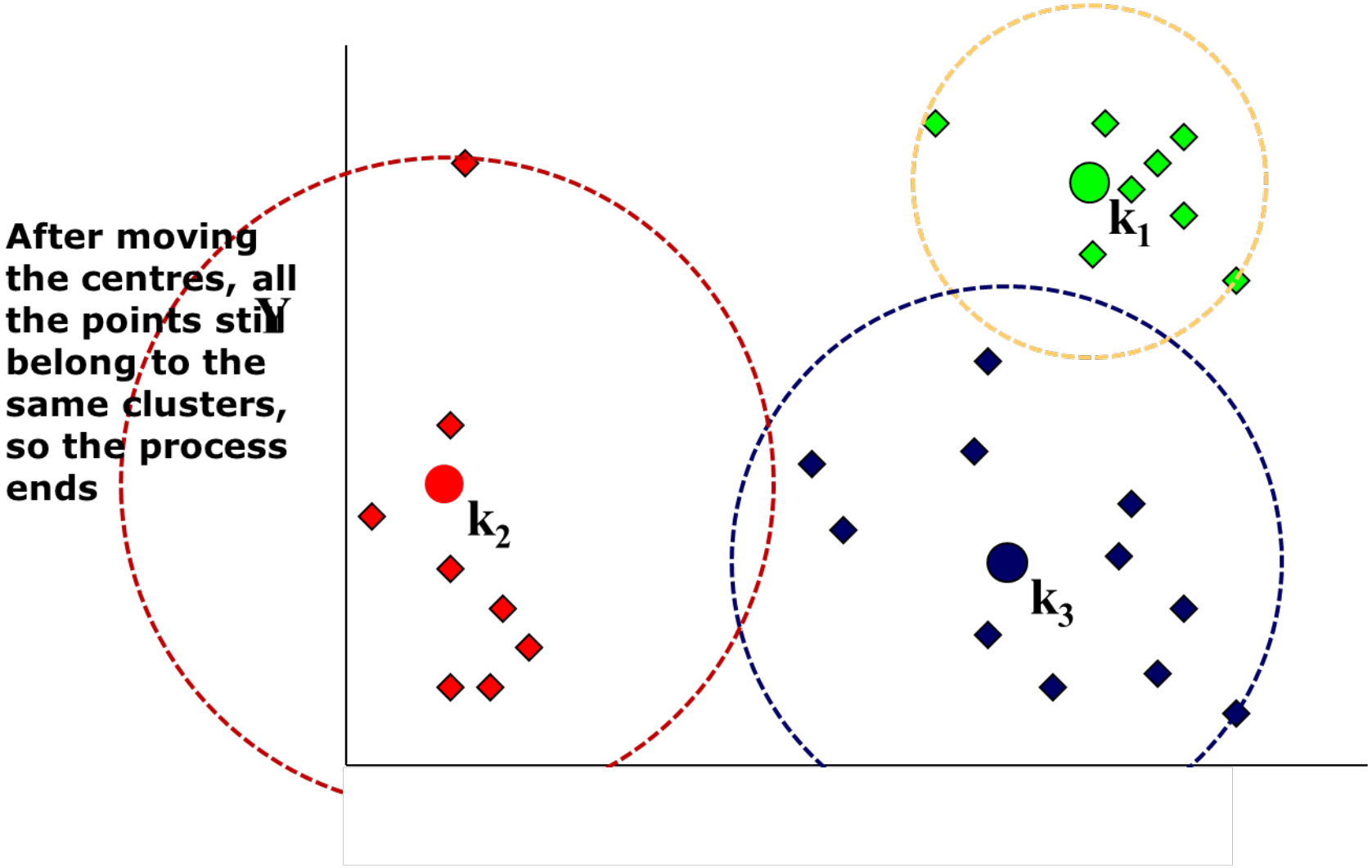
K-MEANS EXAMPLE



K-MEANS EXAMPLE



K-MEANS EXAMPLE



SOME COMMENTS ON K-MEANS

Advantages

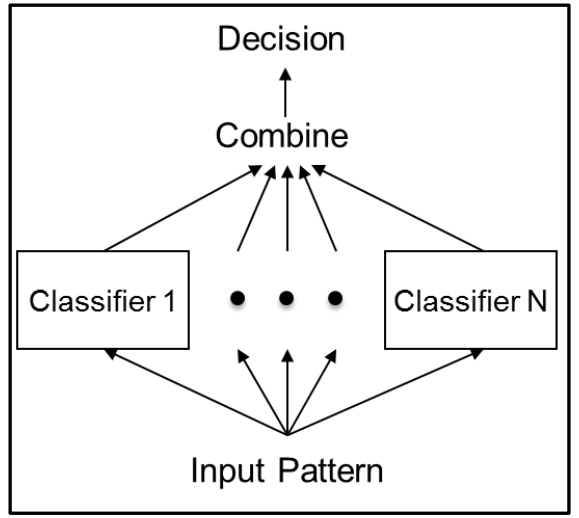
- *Relatively efficient*: O(*tkn*), where *n* is # object, *k* is # *clusters*, and *t* is # iterations. Usually, *k*, *t* << *n*.
- It often ends up in a local optimal, depending on the initial choice of cluster centres.
  - Reset the seeds
  - Use more powerful search techniques such as genetic algorithms or stochastic cooling

Disadvantages

- It is only applicable when the concept of average is definable. What to do with nominal data?
- Need to set the number of clusters in advance (k)
- Weak in the face of noisy data and/or outliers
- Only suitable for convex clusters (spherical...)

ENSEMBLE LEARNING

- No Free Lunch theorem: There is no algorithm that is always the most appropriate for ALL problems
- Generate a group of base-learners which when combined have higher accuracy
- **Diversity** and **Accuracy** among classifiers are the key points for the success of ensembles.
- Different learners use different
  - Algorithms / Parameters
  - Training sets / Subproblems

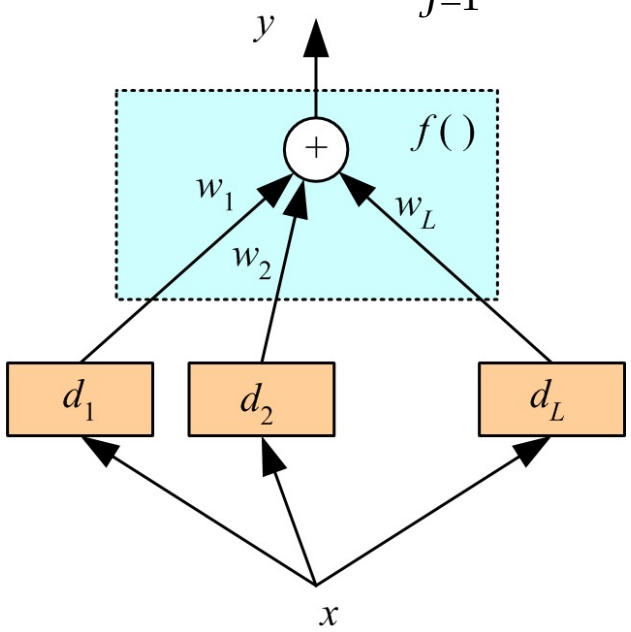


ENSEMBLES: VOTING & BOOTSTRAP

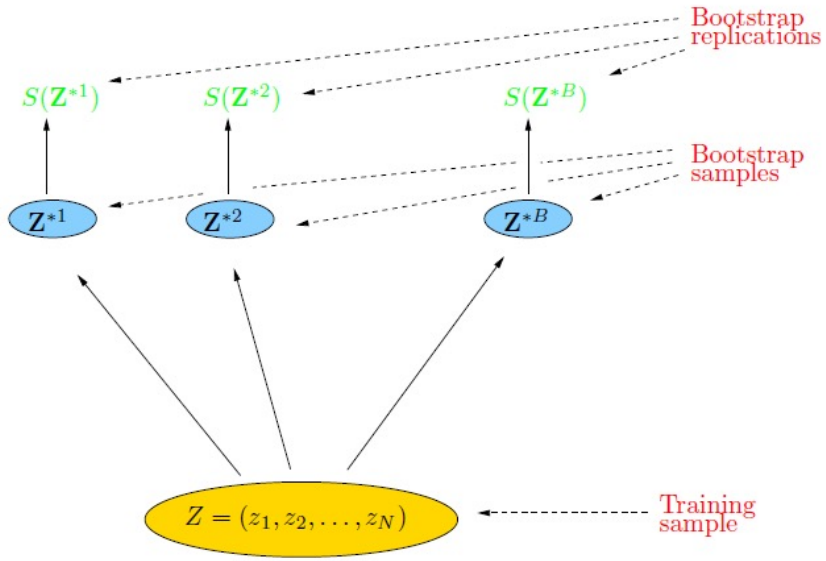
- Voting: Linear combination

$$y = \sum_{j=1}^{L} w_j d_j$$

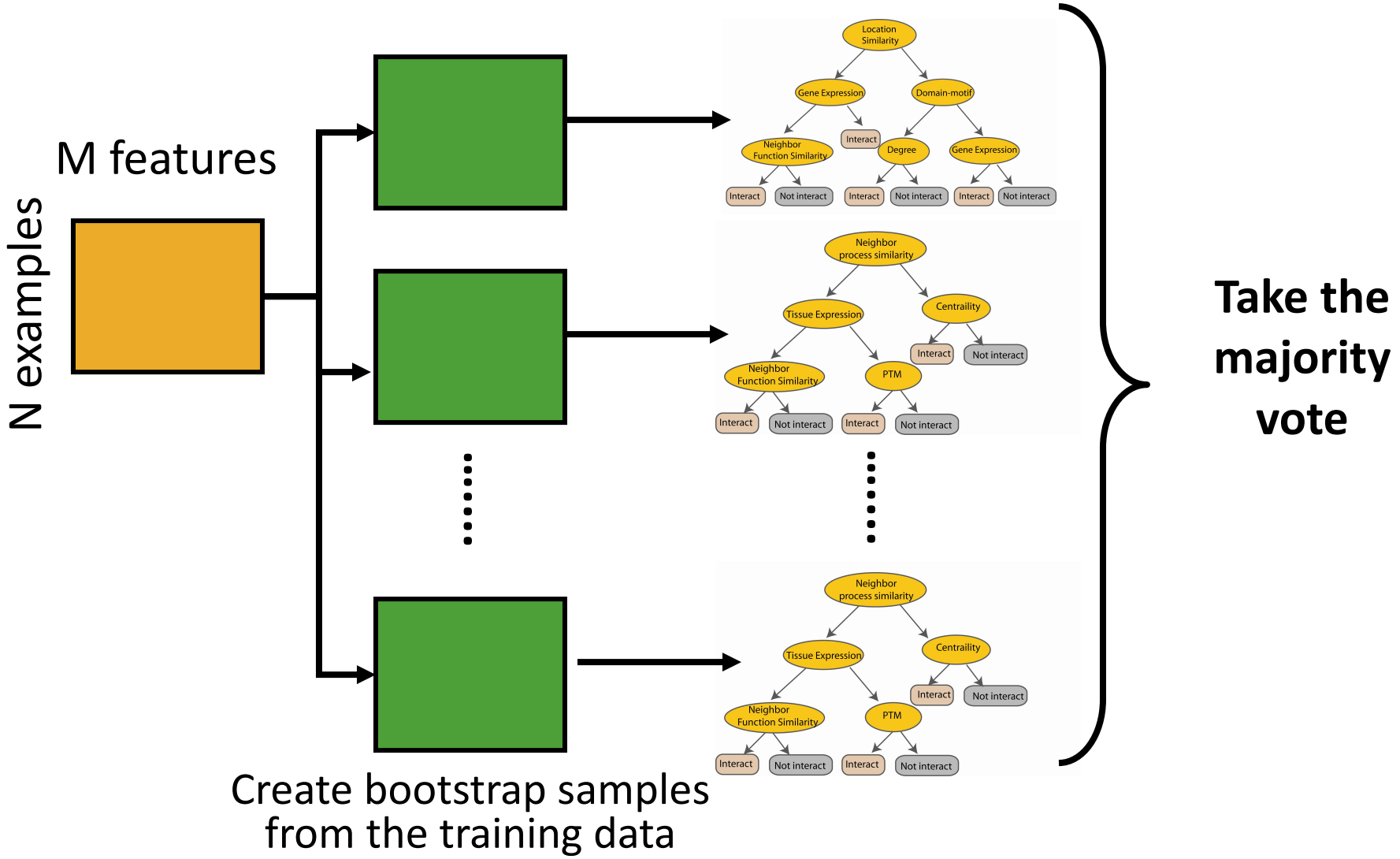
$$w_j \geq 0 \text{ and } \sum_{j=1}^{L} w_j = 1$$



- Bootstrap: randomly draw datasets *with replacement* from the training data, each sample *the same size as the original training set*



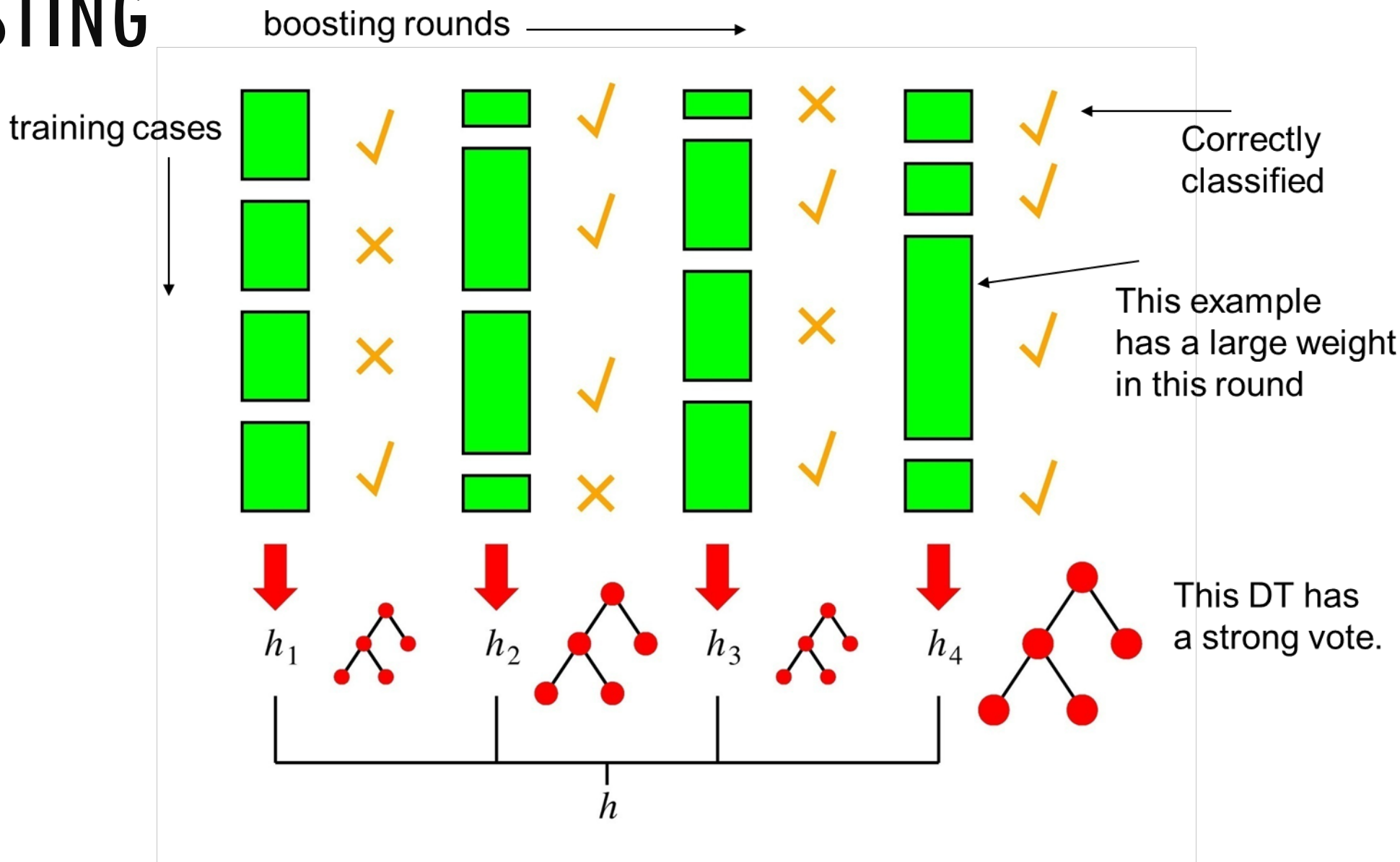
BAGGING: RANDOM FOREST



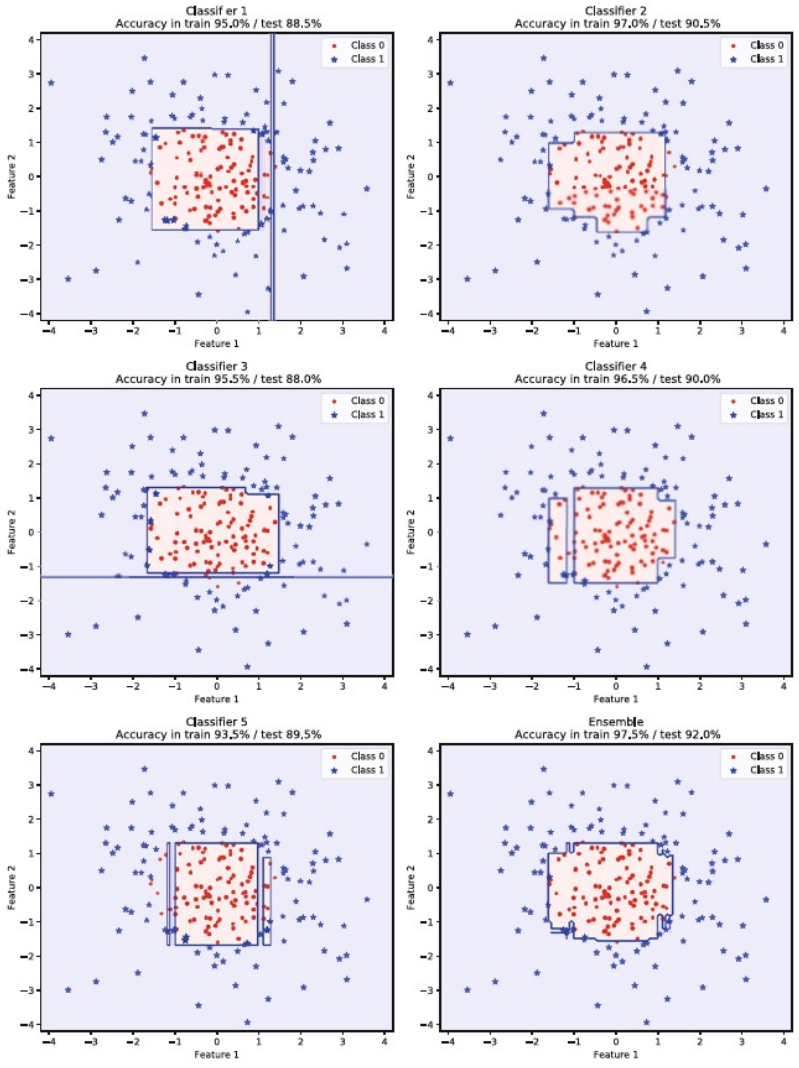
BOOSTING

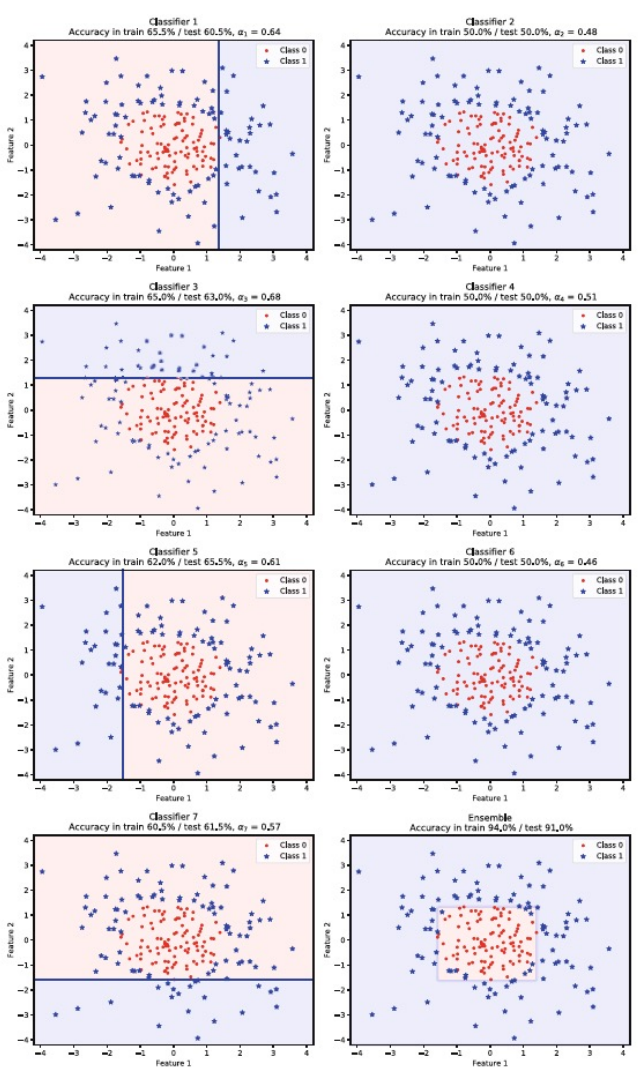
- Train classifiers (e.g. decision trees) in a sequence.
- A new classifier should focus on those cases which were incorrectly classified in the last round → **give more weight to misclassified examples**
- Combine the classifiers by letting them vote on the final prediction (like bagging).
- Each classifier is "weak" but the ensemble is "strong."
- AdaBoost or XGBoost are specific boosting methods.

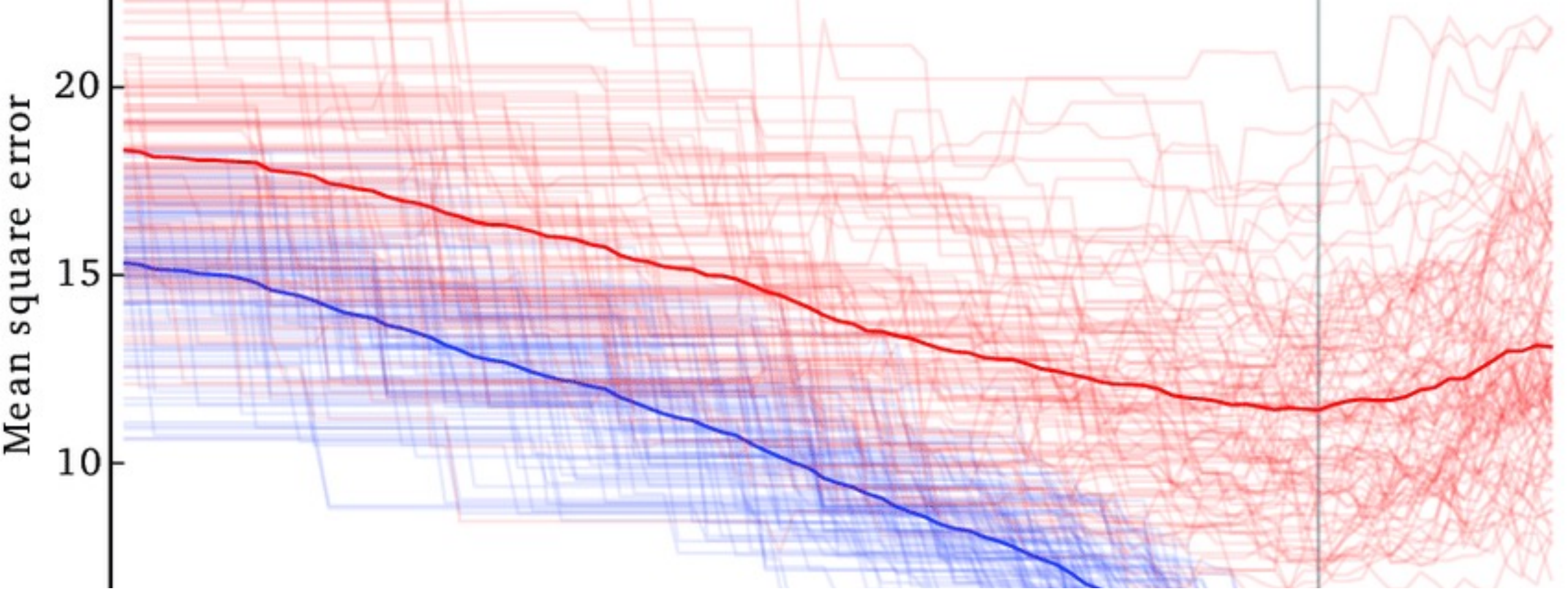
BOOSTING


boosting rounds
training cases
Correctly classified
This example has a large weight in this round
This DT has a strong vote.
h_1
h_2
h_3
h_4
h

RANDOM FOREST VS. BOOSTING







EVALUATION AND VALIDATION | 5

CRITERIA FOR EVALUATING A CLASSIFIER

- Confusion matrix

Given a classification problem with *m* classes, a confussion matrix is a matrix *m* x *m* in which an entry $c_{i,j}$ indicates the number of example assigned to the class c_j, being the correct class c_i

	Prediction		
	Short	Medium	Tall
Short	0	4	0
Medium	0	5	3
Tall	0	1	2

$N=TP+TN+FP+FN$

- Accuracy $s = \frac{TP + TN}{N}$
- Error Rate $\varepsilon = 1 - s$

METRICS

Confusion Matrix

		Prediction	
		C_P	C_N
Actual class	C_P	**TP**: True positive	**FN**: False negative
	C_N	**FP**: False positive	**TN**: True negative

accuracy = (TP+TN)/(TP+TN+FP+FN)

Limitations of accuracy:

Let's assume a problem with 2 classes:

- 9990 examples of class 1
- 10 examples of class 2

If the classification model always says that the examples are class 1, its accuracy is

9990/10000 = 99.9%

Totally misleading, as we will never detect any example of class 2

METRICS

Alternative: **Cost Matrix**

C(i|j)		Prediction	
		C_P	**C_N**
Actual Class	**C_P**	**C(P|P)**	**C(N|P)**
	C_P	**C(P|N)**	**C(N|N)**

The cost of classification will be proportional to the accuracy of the classifier only if

$$\forall i,j: i \neq j \qquad C(i|j) = C(j|i)$$
$$C(i|i) = C(j|j)$$

cost-sensitive metrics

		Prediction	
		C_P	**C_N**
Actual Class	**C_P**	**TP**: True positive	**FN**: False negative
	C_N	**FP**: False positive	**TN**: True negative

precision = TP/(TP+FP)

True positive recognition rate

recall = sensitivity = TP/P = TP/(TP+FN)

True negative recognition rate

specificity = TN/N = TN/(TN+FP)

F-measure

F = 2*precision*recall / (precision+recall)

METRICS

		Prediction	
		C_P	C_N
Real	C_P	TP	FN
	C_N	FP	TN

Accuracy

		Prediction	
		C_P	C_N
Real	C_P	TP	FN
	C_N	FP	TN

Recall

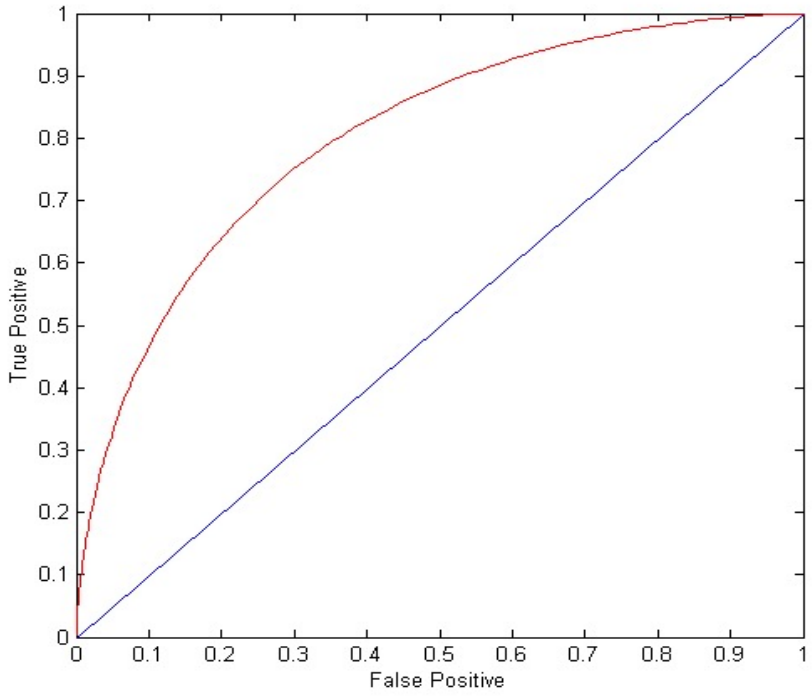
		Prediction	
		C_P	C_N
Real	C_P	TP	FN
	C_N	FP	TN

Precision

		Prediction	
		C_P	C_N
Real	C_P	TP	FN
	C_N	FP	TN

F-measure

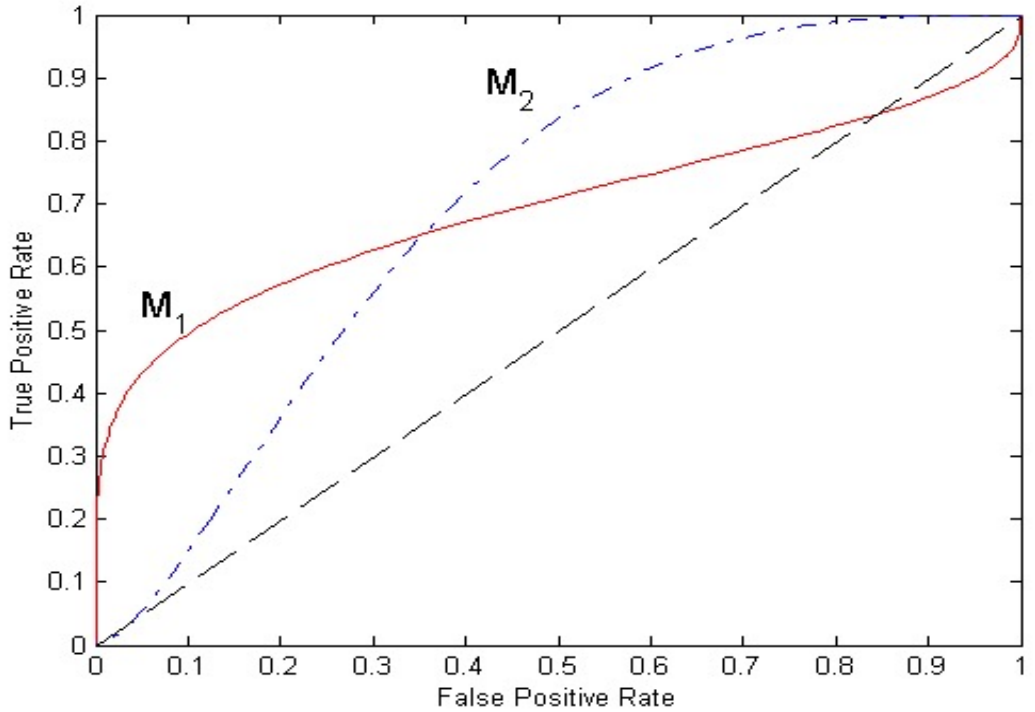
METRICS: ROC CURVES



Vertical axis:
"true positive rate" TPR = TP/(TP+FN)
Horizontal axis:
"false positive rate" FPR = FP/(FP+TN)

- Developed in the 1950s to analyze signals with noise: characterizing the trade-off between right and wrong alarms.
- They allow us to visually compare different classification models in **binary classification problems**.
- The area under the curve is a measure of the accuracy of the classifier:
  - The closer we are to the diagonal (area close to 0.5), the less accurate the model will be.
  - A "perfect" model will have area 1.

METRICS: ROC CURVES



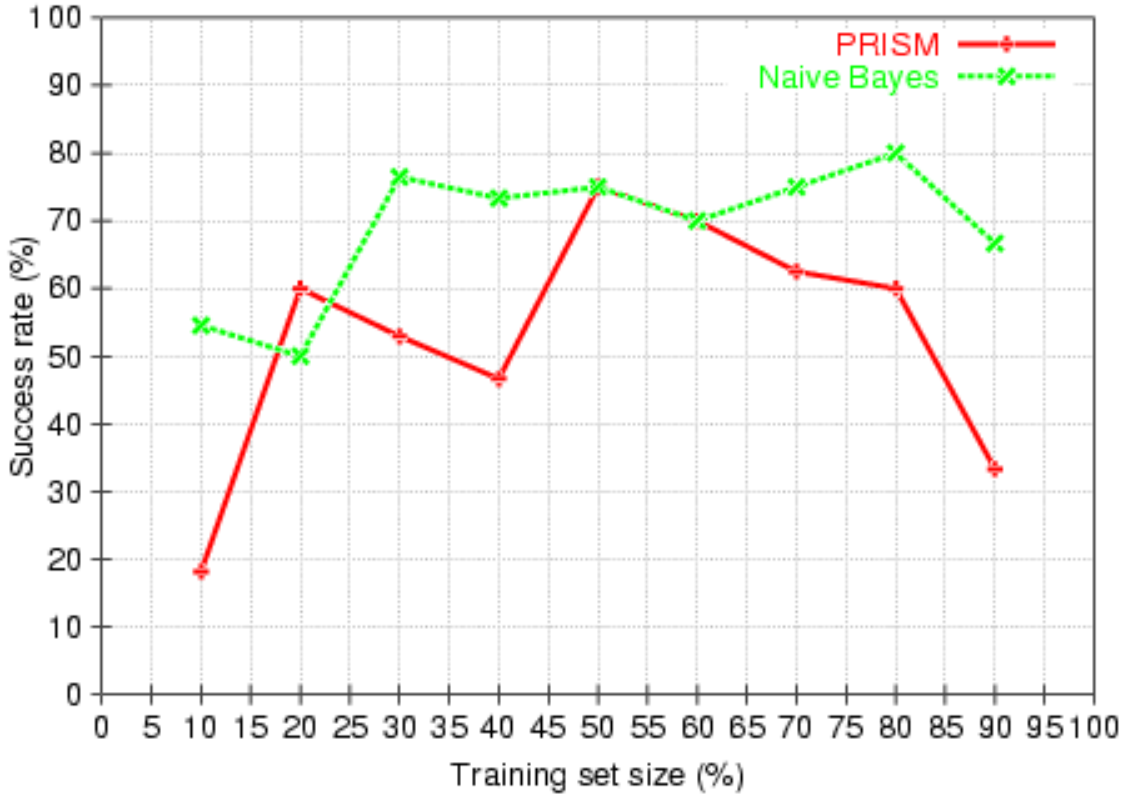
Neither model is consistently better than the other: M1 is better for low FPR, M2 for high FPR.

VALIDATION IN CLASSIFICATION AND REGRESSION

- The aim of the validation methods is to make an honest estimate of the goodness of the built learning algorithm
- Using the success rate on the training set is **unrealistic**
  - The error obtained is often too optimistic because the model will be over-adjusted to the data used during the learning process
- There are various techniques for validating learning algorithms, including
  - ***Hold-out***
  - ***Cross-validation***

HOLD-OUT VALIDATION

- It consists of dividing the DB into two independent sets: training (**TR**) and test (**TS**)
- The size of TR is usually greater than TS: (2/3, 1/3, 4/5, 1/5,...)
- TR elements are usually obtained by sampling without replacement of the initial DB. The TS is made up of the elements not included in the TR
- Usually used in **large** data sets



Test set (%) + Training set (%) = 100%

CROSS-VALIDATION

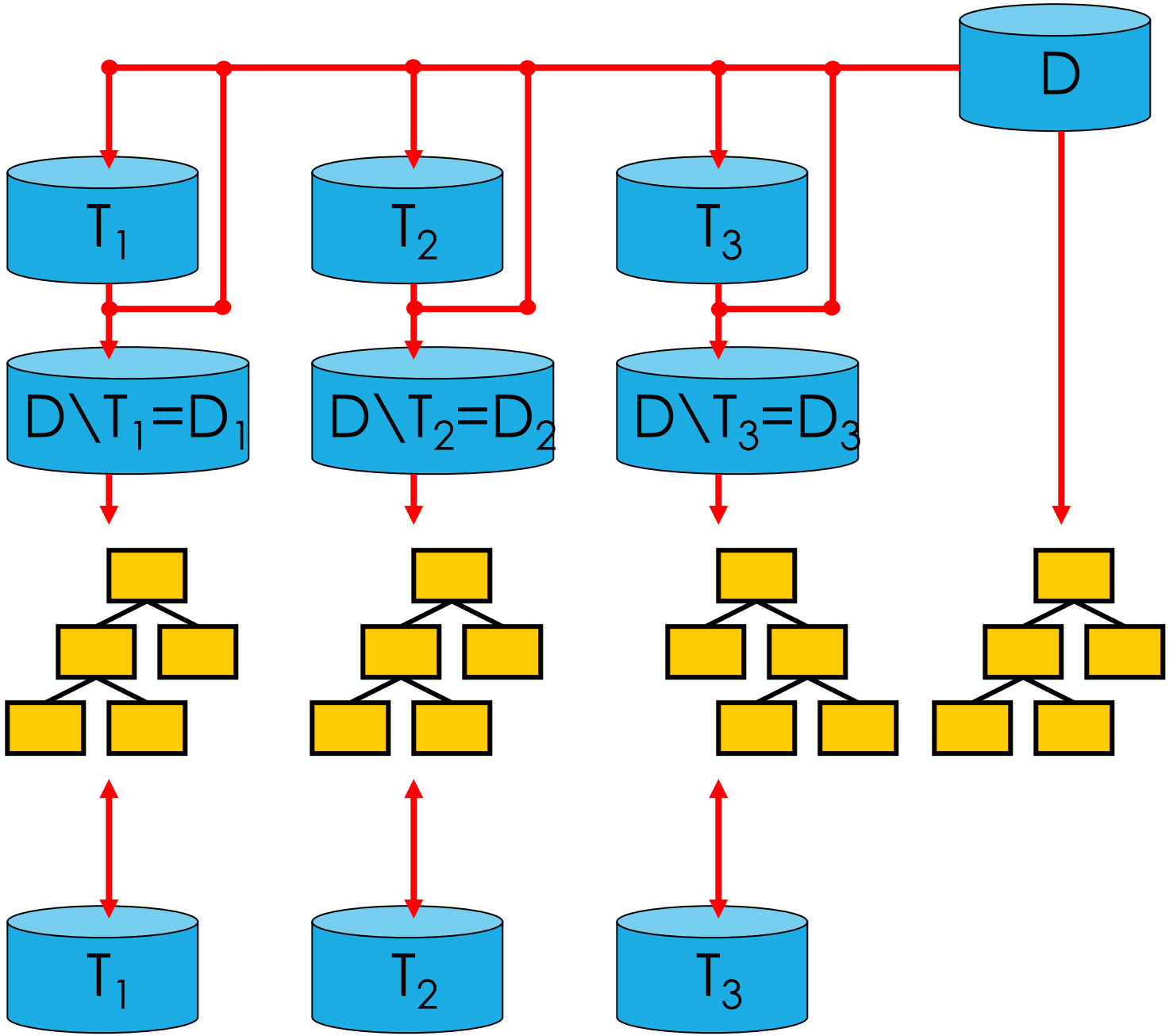
- It consists of:
  1. Divide the DB into *k* subsets (*folds*), $\{S_1,\ldots,S_k\}$ of equal size
  2. Learn ***k* models** by using a different TR in each of them. Check with the corresponding TS

$$TR = S_1 \cup \ldots \cup S_{i-1} \cup S_{i+1} \cup \ldots S_k$$
$$TS = S_i$$

  3. Return as the error rate the mean obtained in the *k* iterations

- Stratified cross validation (in classification): Subsets are stratified according to the class variable
- Typical values of k=5,10
- Usually used in **moderate sized** databases

- Partition
- Train
- Test



METRICS FOR EVALUATING A REGRESSOR

- All the validation techniques studied in classification are valid for numerical prediction
- The difference is that now we must measure the error in another way
- We must measure the error made in approximating a set of values $\{v_1,\ldots,v_n\}$ by its esimation $\{v'_1,\ldots,v'_n\}$

Mean Squared Error (MSE)

$$MSE = \frac{\sum_{i=1}^{n}(v_i - v_i')^2}{n}$$

Standarized MSE (SMSE)

$$SMSE = \sqrt{\frac{\sum_{i=1}^{n}(v_i - v_i')^2}{n}}$$

Mean Absolute Error (MAE)

$$MAE = \frac{\sum_{i=1}^{n}|v_i - v_i'|}{n}$$

Relative MAE (RMAE)

$$RMAE = \frac{\sum_{i=1}^{n}|v_i - v_i'|}{\sum_{i=1}^{n}|v_i - \bar{v}|}$$

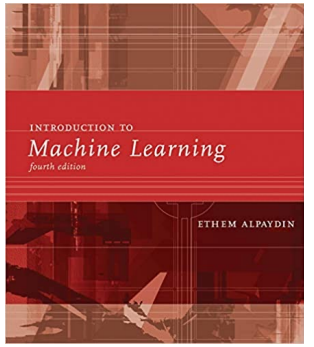
Correlation coefficient

$$r_{vv'} = \frac{\sum_{i=1}^{n}(v_i - \bar{v})(v_i' - \bar{v}')}{(n-1)\sigma_v\sigma_{v'}}$$

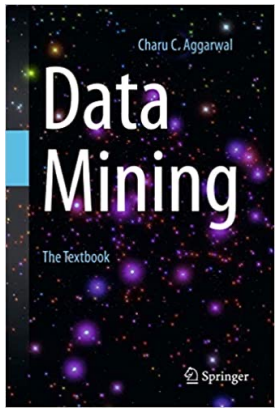
INTERESTING BIBLIO | 6

INTERESTING READINGS

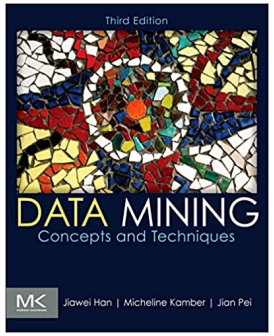
E. Alpaydin. Introduction to Machine Learning, fourth edition (Adaptive Computation and Machine Learning series). MIT Press, 2020.



C.C. Aggarwal. Data Mining: The Textbook. Springer, 2015.

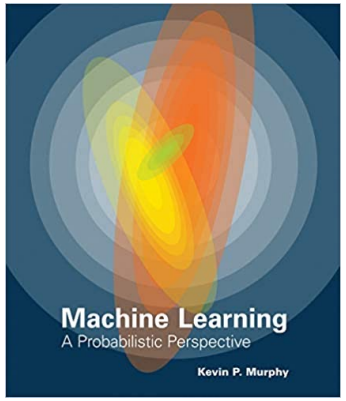


J. Han, M. Kamber, J. Pei. Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems). Morgan Kaufmann, 2011.

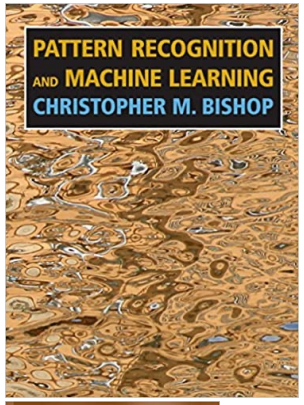


INTERESTING READINGS

K.P. Murphy. Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning series). MIT Press, 2012.



C.M. Bishop. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer, 2006.



T. Hastie, R. Tibshirani, J. Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics). Springer, 2016.