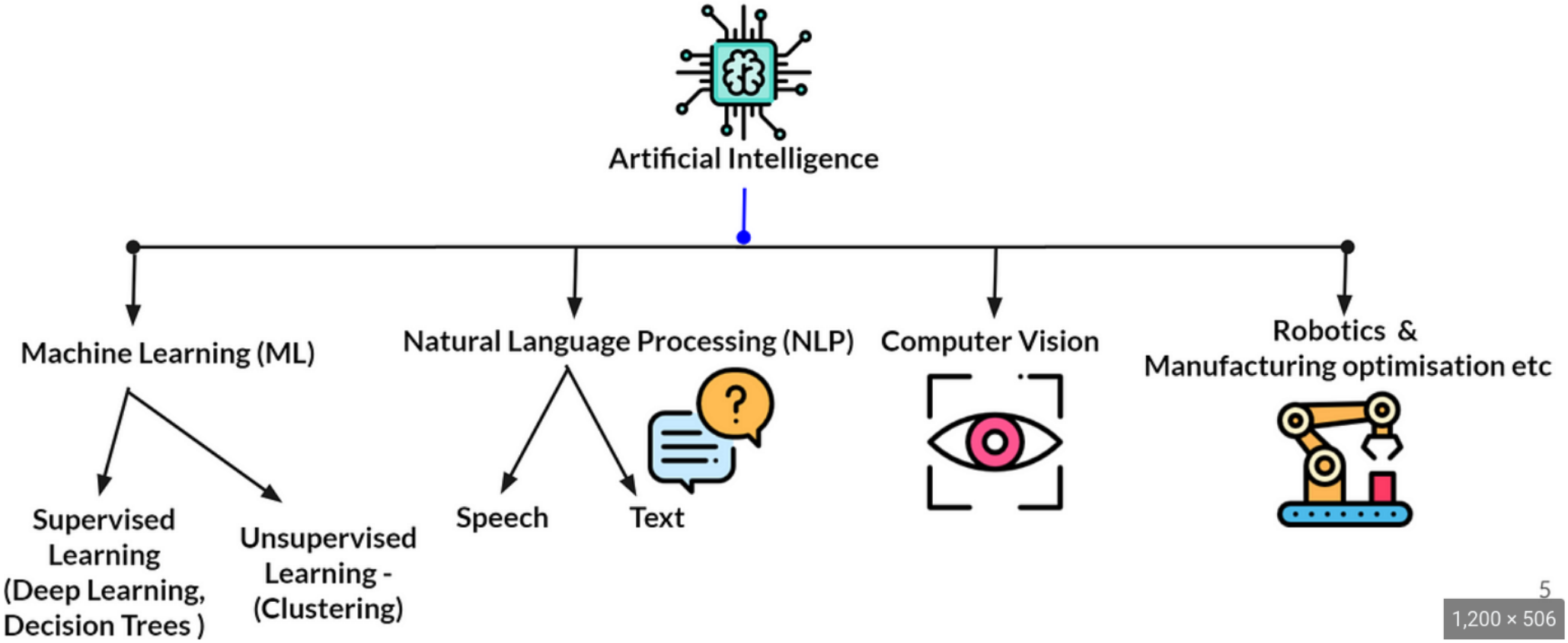
Fast introduction to ML models


Artificial Intelligence
Machine Learning (ML)
Supervised Learning (Deep Learning, Decision Trees)
Unsupervised Learning - (Clustering)
Natural Language Processing (NLP)
Speech
Text
Computer Vision
Robotics & Manufacturing optimisation etc

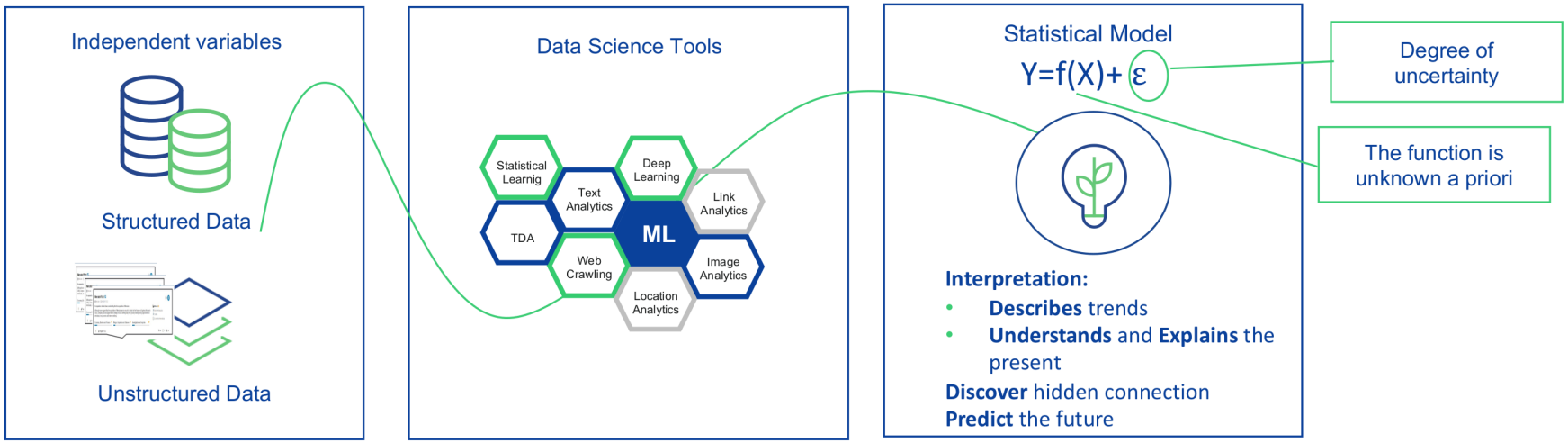
More scientific perspective of ML

What is a data model

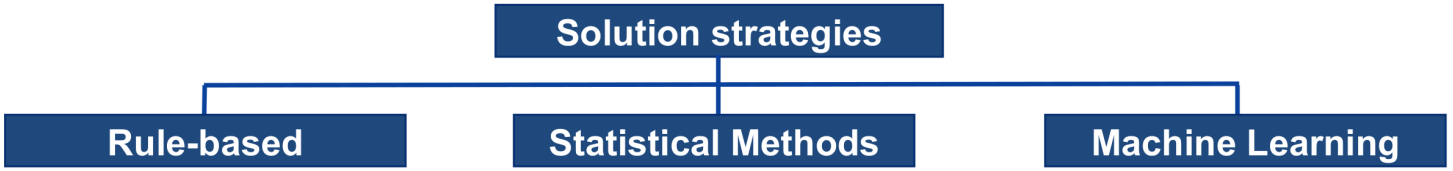
Statistical Model for a learning opportunity

Data model is a formal structure to **describe**, **understand, explain** and **simplify** relationship and complex phenomena, to discover hidden connection and predict future trends.

- ✓ The formalization is based on the use of **analytical methodologies** and tools typical of **data science**.
- ✓ It is accepted that some degree of **inaccuracy** and **uncertainty** can exist.
- ✓ The degree of uncertainty is estimated and described through appropriate mathematical structures: **random variables.**



History of data models

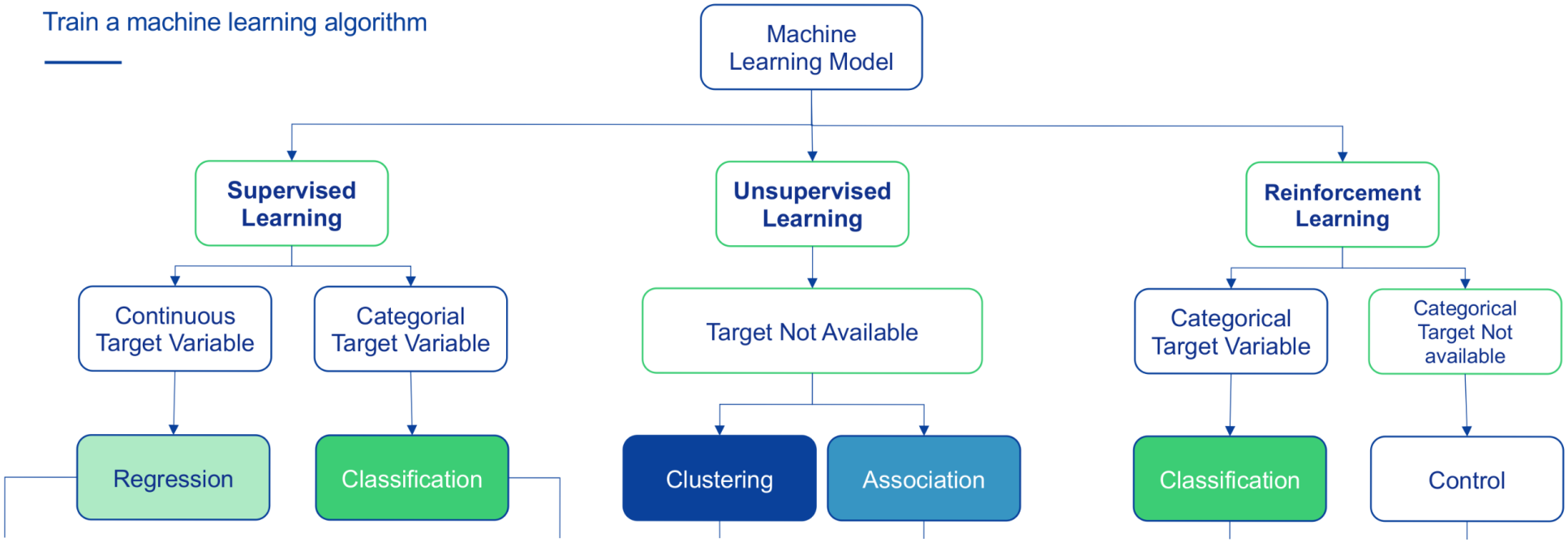


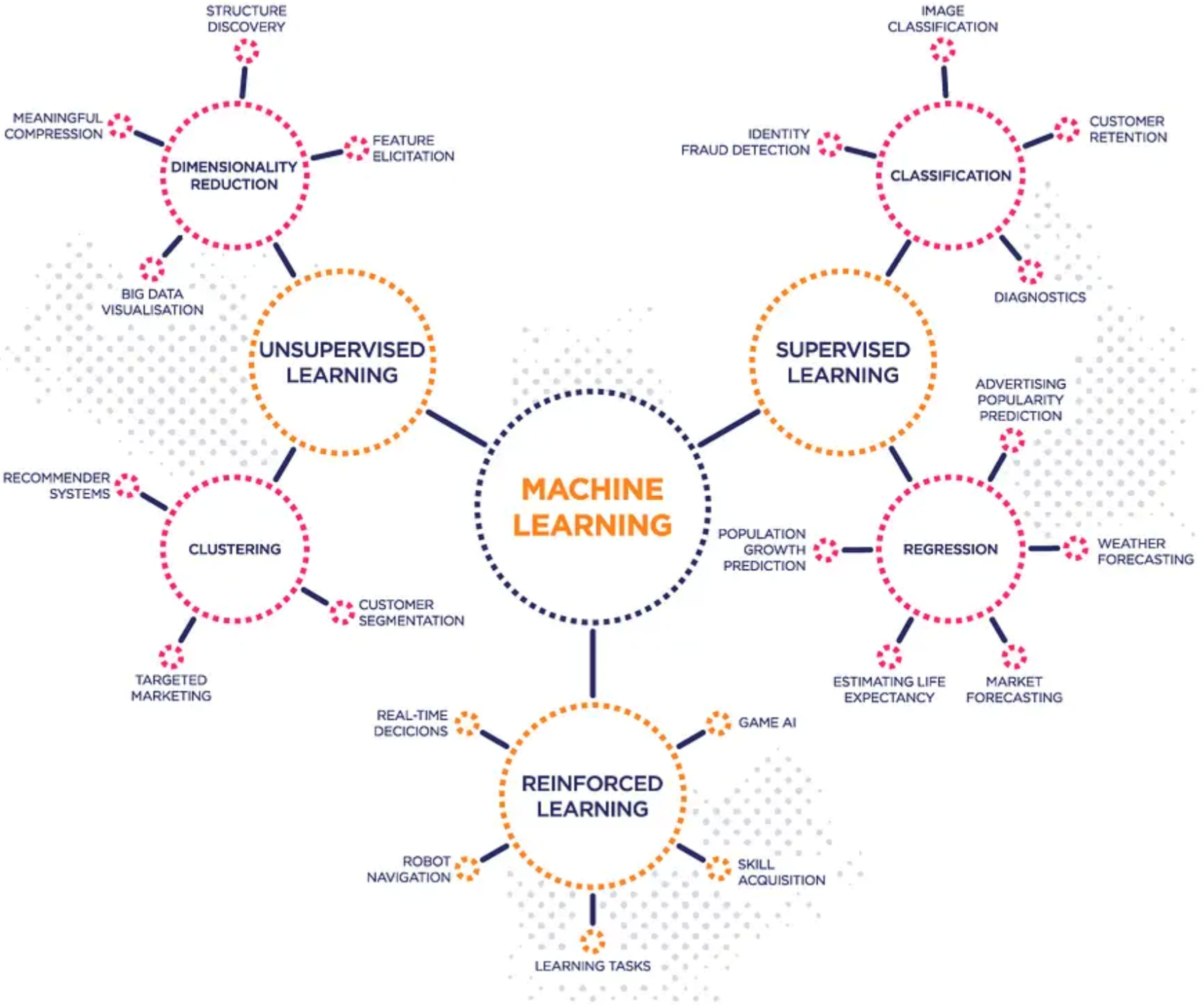
	Rule-based	Statistical Methods	Machine Learning
Data Source	Patterns are identified by business experts based on their previous experience	Models are developed through statistical analysis based on datasets of structured variables	The full volume and range of data available can be used (also textual data, images…) through advanced algorithms
Interpretability	Rules are easily understood by business users	Parameters used in the model derive from an initial understanding of the business issue and are interpretable	Depending on the technique used, the resulting model is not always interpretable
Maintenance	Rules are set up and implemented statically (do not automatically adapt to new emerging patterns)	The statistical model is static in nature but allows for more flexibility in its maintenance (through recalibration)	The model can be routinely updated through a retraining based on new available data
Implementation (time & effort)	Fast implementation, often on legacy data infrastructures	Its implementation requires time for data exploration and hypothesis testing	Its implementation requires more time in the initial set up and performance evaluation; it also requires the implementation of specific data and IT infrastructures

What ML model do I need to choose?

Data modelling

Train a machine learning algorithm




MACHINE LEARNING
UNSUPERVISED LEARNING
SUPERVISED LEARNING
REINFORCED LEARNING
DIMENSIONALITY REDUCTION
CLUSTERING
CLASSIFICATION
REGRESSION
STRUCTURE DISCOVERY
MEANINGFUL COMPRESSION
FEATURE ELICITATION
BIG DATA VISUALISATION
RECOMMENDER SYSTEMS
CUSTOMER SEGMENTATION
TARGETED MARKETING
IMAGE CLASSIFICATION
IDENTITY FRAUD DETECTION
CUSTOMER RETENTION
DIAGNOSTICS
ADVERTISING POPULARITY PREDICTION
POPULATION GROWTH PREDICTION
WEATHER FORECASTING
ESTIMATING LIFE EXPECTANCY
MARKET FORECASTING
REAL-TIME DECICIONS
GAME AI
ROBOT NAVIGATION
SKILL ACQUISITION
LEARNING TASKS

Classification models

Introduction to classification models

Problem setting

- They all share the same schema, by having:
  - a response variable (**target**) of categorical nature,
  - a number of explanatory variables (**features**) which are assumed to be helpful in classifying each observation,
  - and the following objectives

CLASSIFICATION

Predicting the **score or probability of an observation to belong to a given class** depending on a set of properties that describe that observation

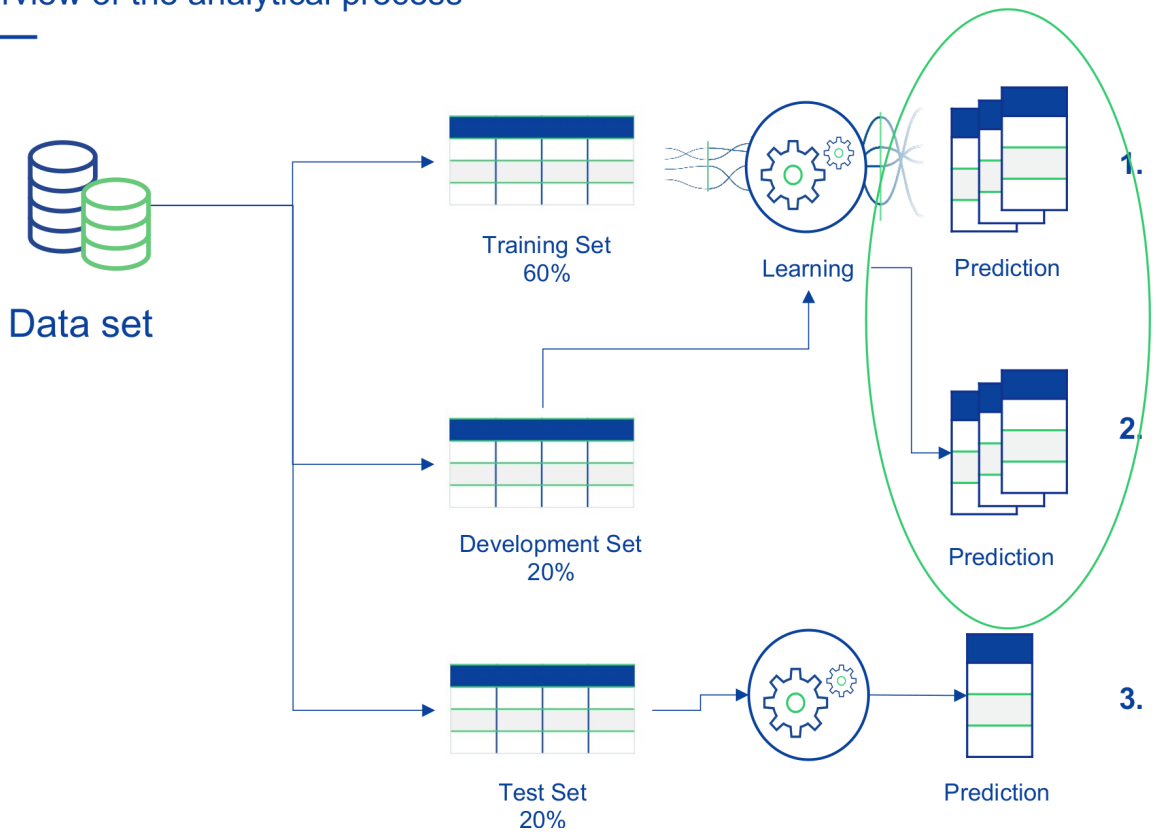
INTERPRETATION

Assessing how much each property is relevant in defining the categories and which set of properties better describe the population belonging to a certain class

Development of ML models

Training, development and test set

Overview of the analytical process

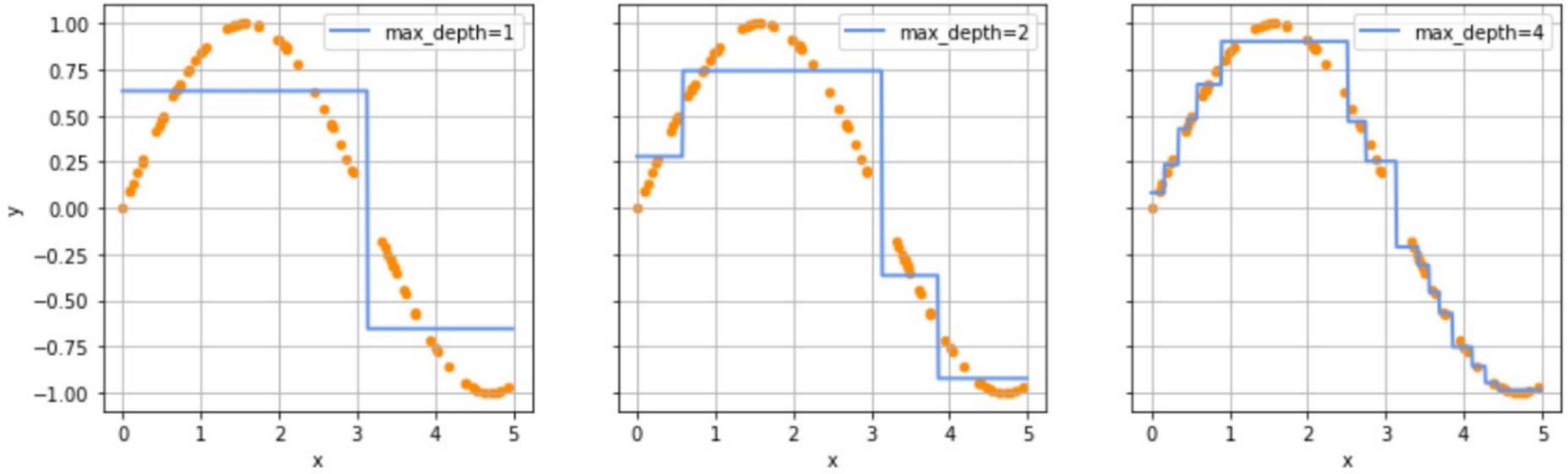


1. **Train** different models
2. **Apply** trained models on **development data set**, tuning and select the best model based on fit statistics
3. **Apply** choosen model to **new data** (or test set)

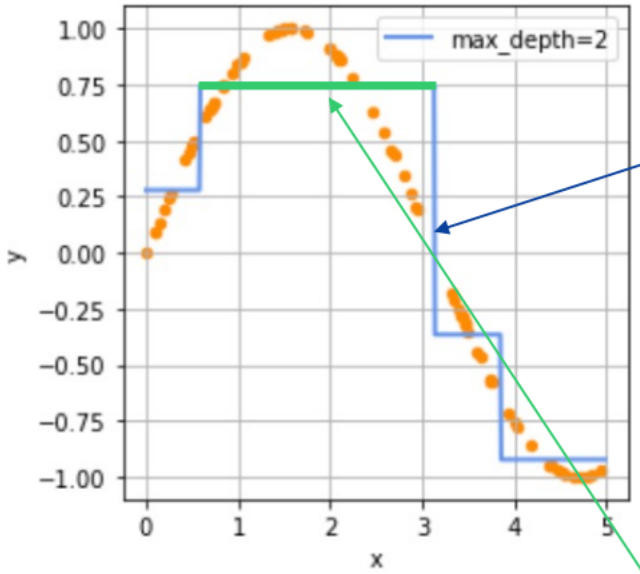
Simplest model: Decision Tree

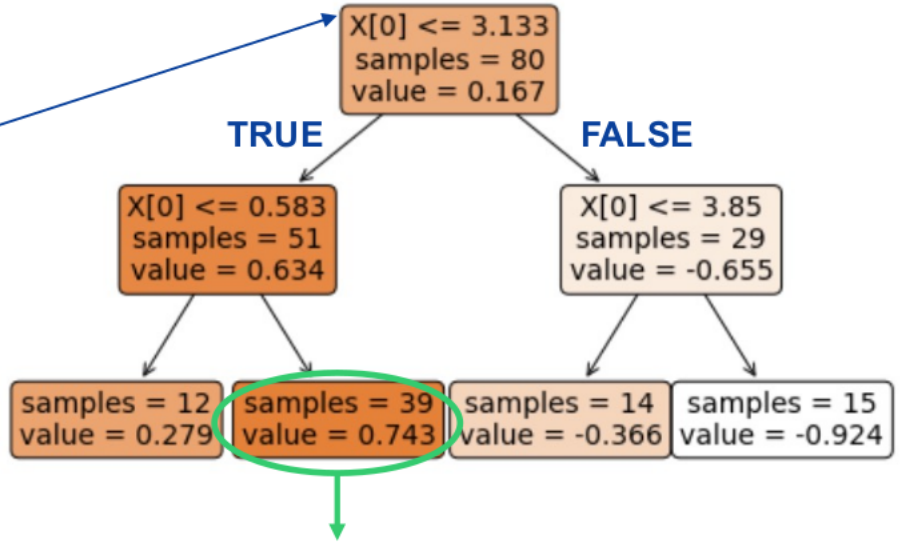
- Decision trees are simple yet **very powerful models** that can be used either for regression or classification tasks
- They provide an approximation of **any function** through a **multi-step function**

  → The more steps you take, the better the approximation



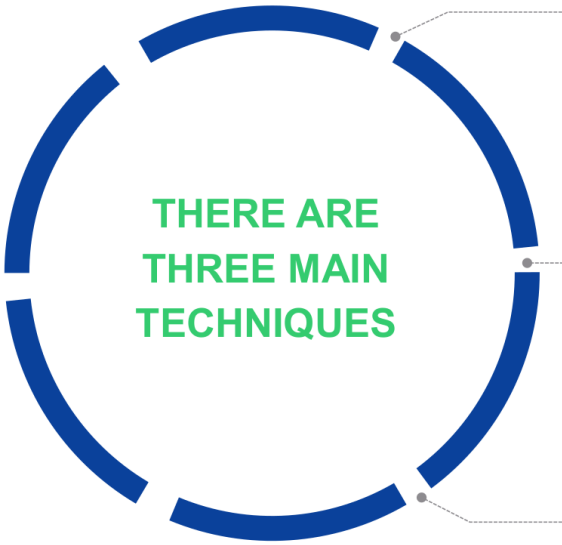
Decision Tree





The prediction 0.743 is computed as the average target $\bar{y}$ on the 39 samples for which $x > 0.583$ and $x \leq 3.133$

Ensemble models



BAGGING → parallel training with different training sets

BOOSTING → sequential training, iteratively re-weighting training examples so current classifier focuses on hard examples

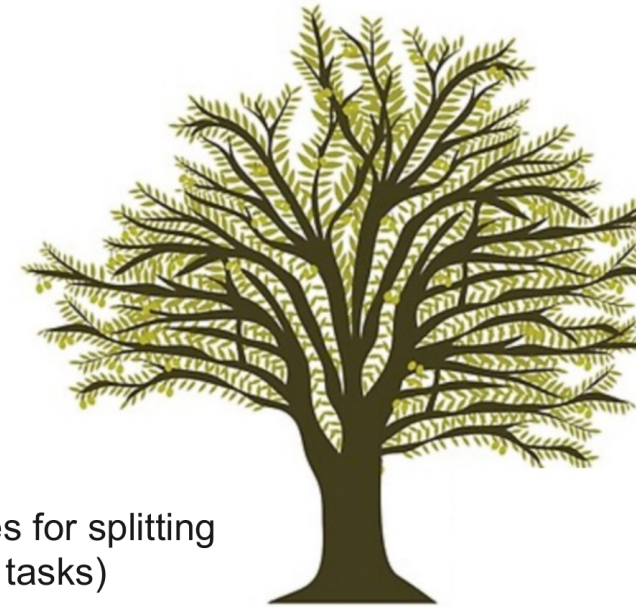
RANDOM FOREST → parallel training with different training sets and feature sets

Random Forest

- **Random forest** is a substantail modification of *bagging* that builds a large collection of ***de-correlated trees*** and then averages them (it is therefore specifically developed for trees as base learners)

The idea is to **improve the variance reduction** of *bagging*
by reducing the correlation of the trees

To do so, in the tree-growing process
features are randomly selected



- Before each split, $\boldsymbol{m \leq p}$ of the input features are selected at random as candidates for splitting (a common choice is $\boldsymbol{m = \sqrt{p}}$ for classification tasks and $\boldsymbol{m = \lfloor p/3 \rfloor}$ for regression tasks)

Random Forest

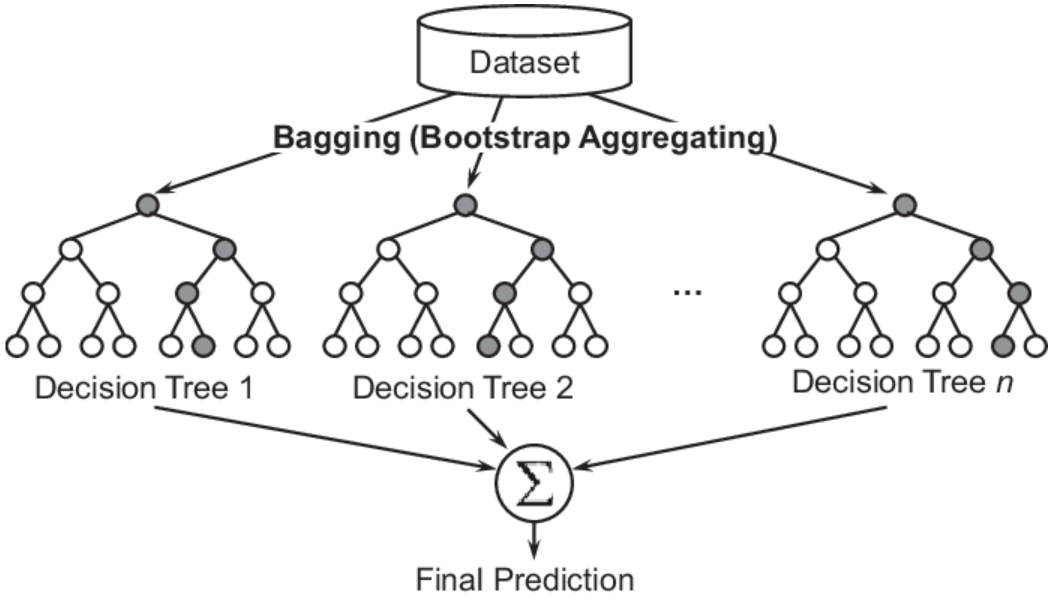


Fig. 1. An Illustration of Random Forest.

Support Vector Classifier

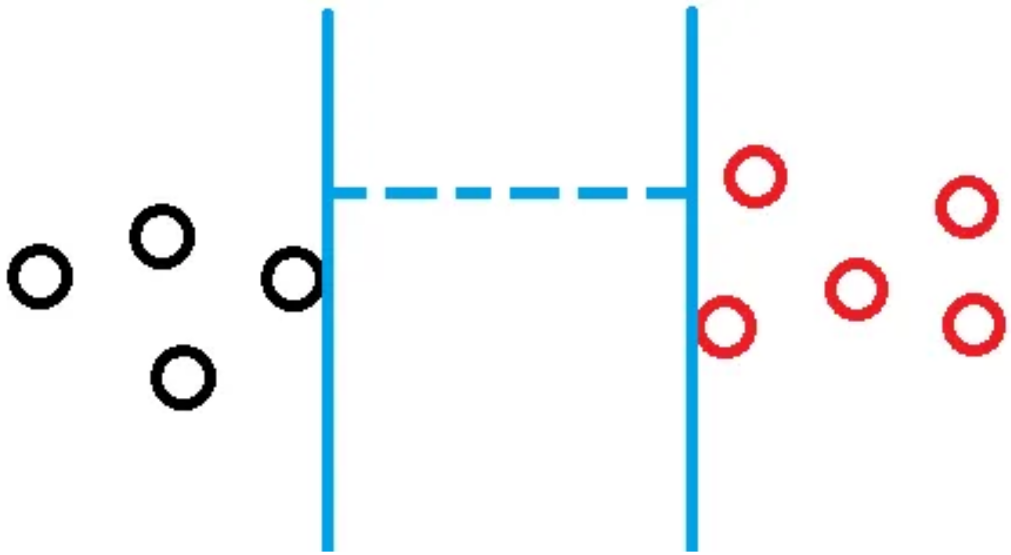
kernel: {'linear', 'poly', 'rbf', 'sigmoid', 'precomputed'} or callable, default='rbf'

This algorithm has two steps:

1) it classifies input data in 2 classes

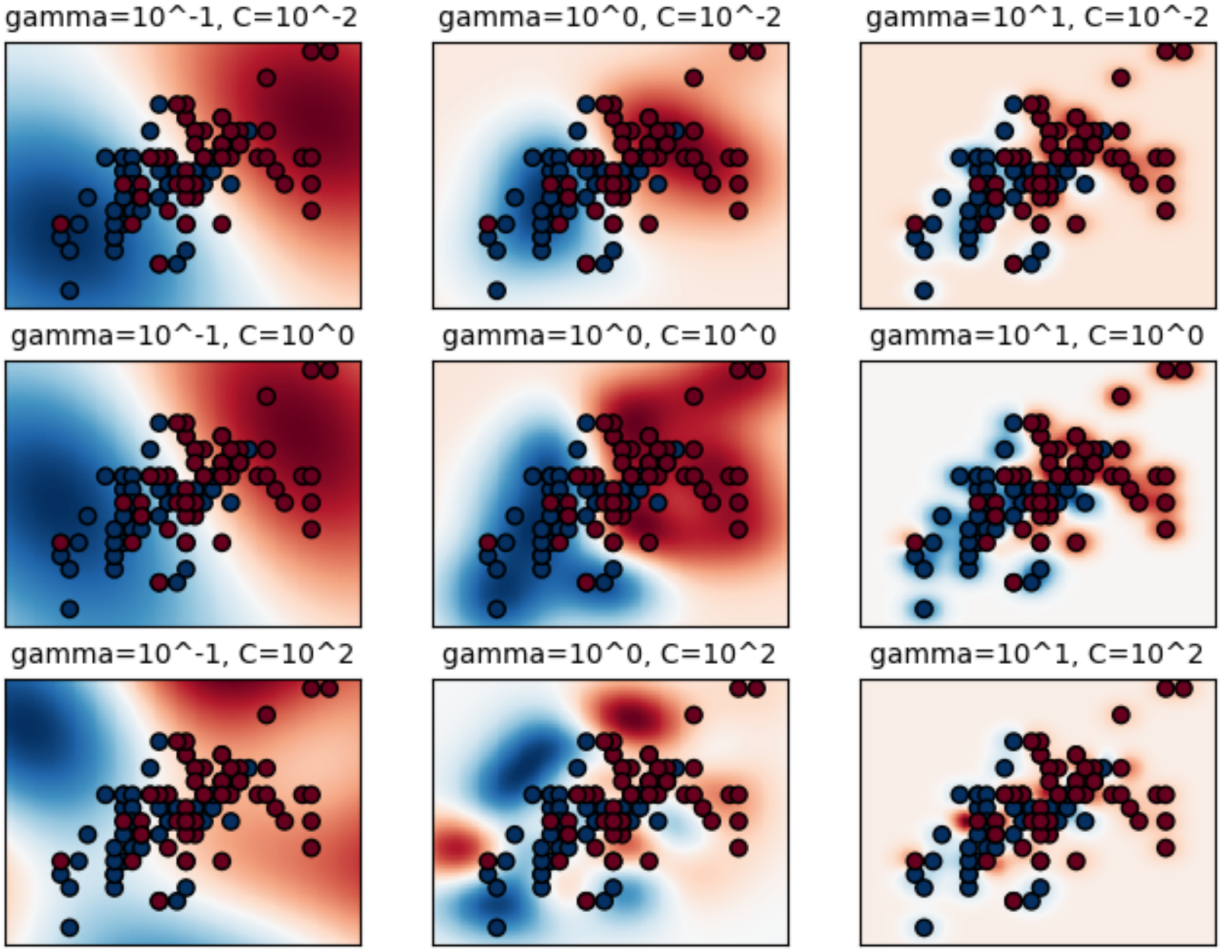
2) it starts to trace some lines
Separation between the two classes, with the objective of maximizing the distance between them.

In the Linear case :

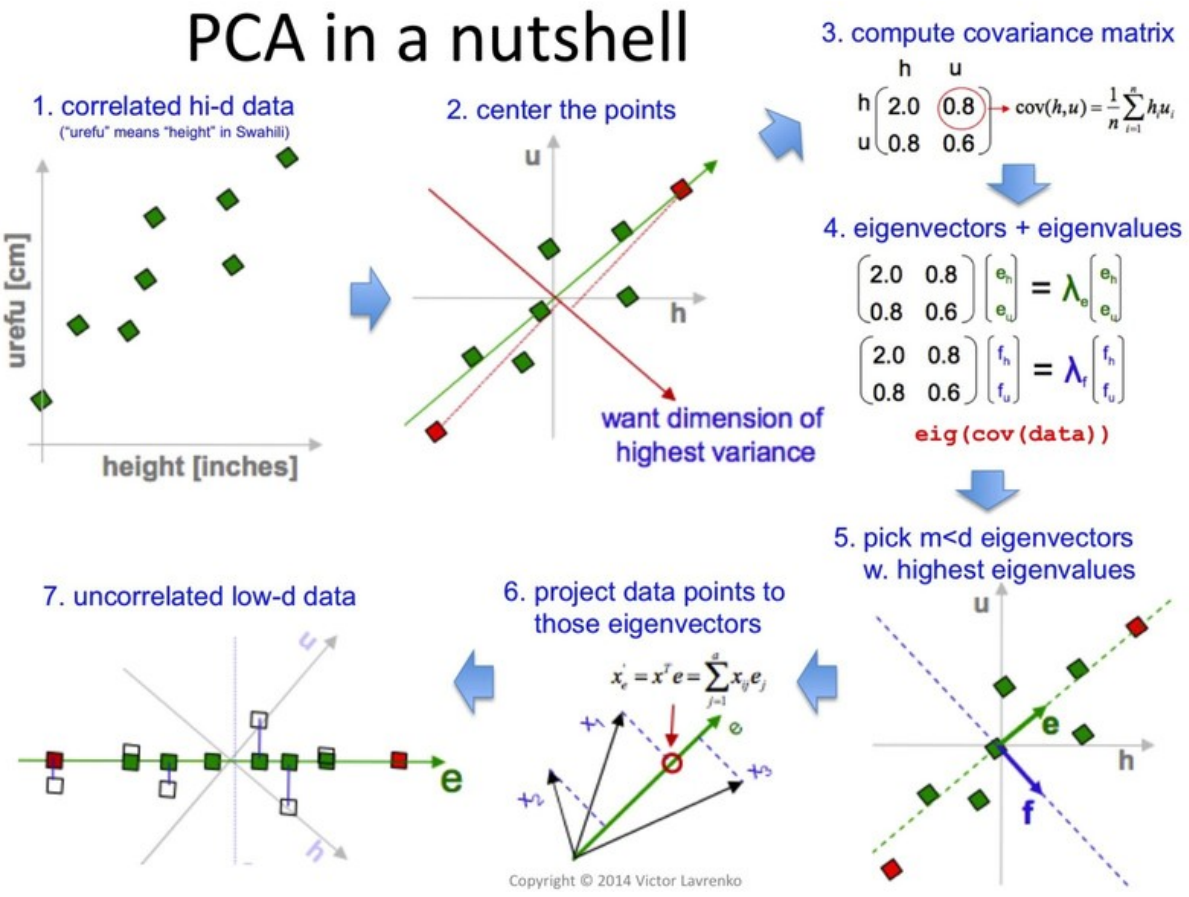


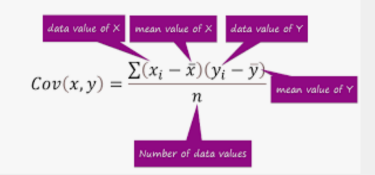
What is the effect of the hyperparameters on SVC accuracy?

Taking a closer look at the hyper parameters of RBF SVC:

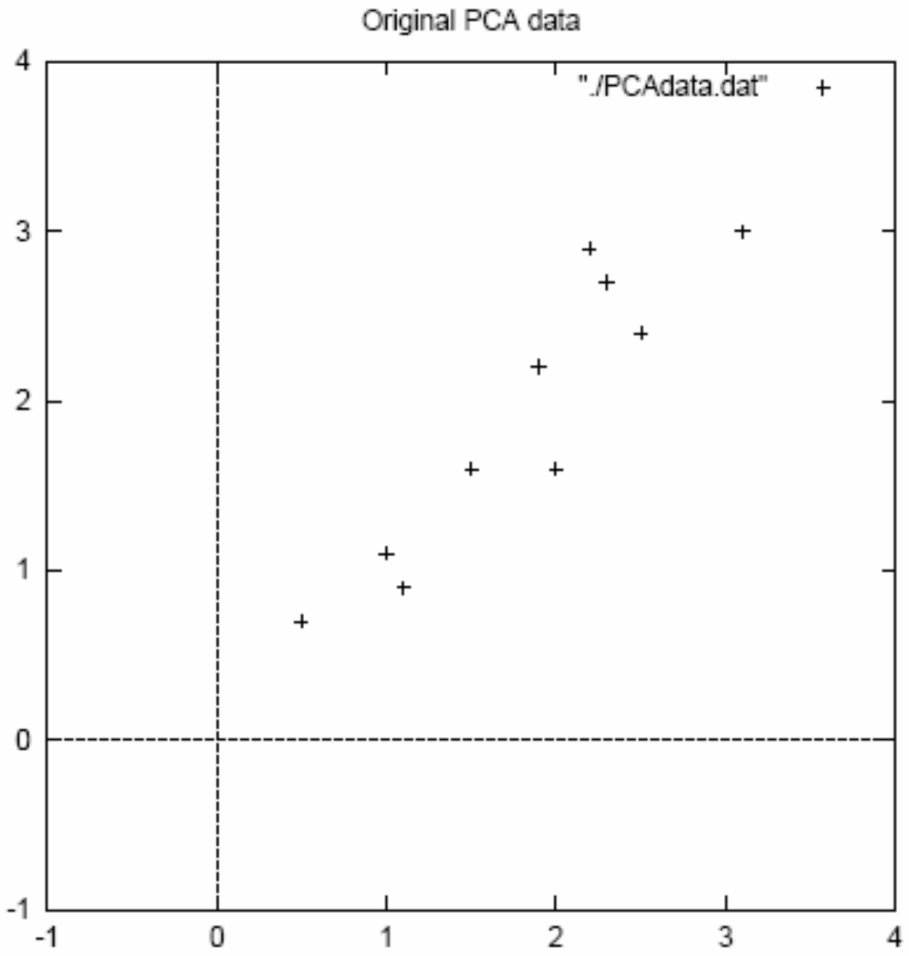


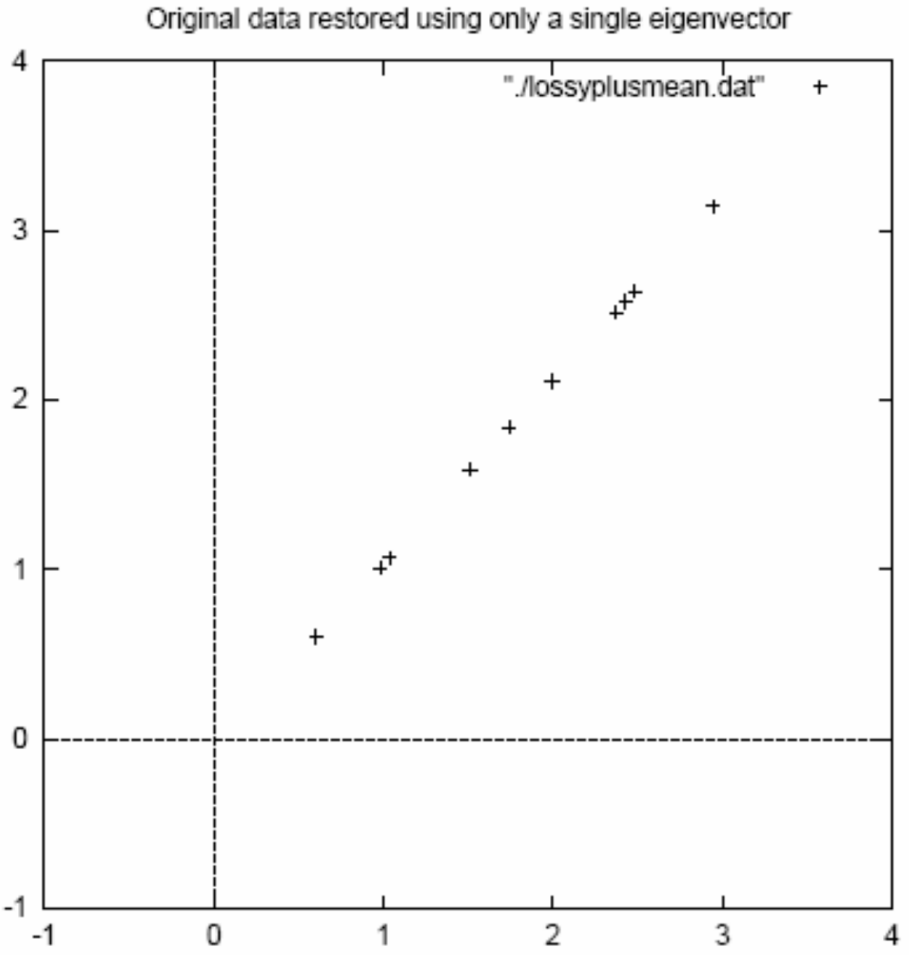
PCA : Principal Component Analysis





PCA : Principal Component Analysis





Model evaluation

ACCURACY

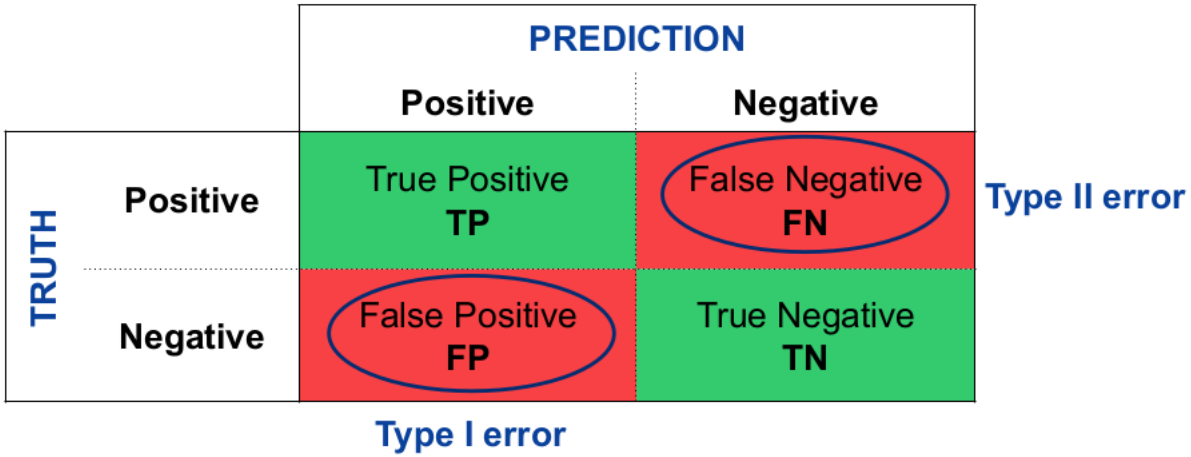
- *How many instances are we correctly predicting?*
- $\frac{TP+TN}{All}$ (should be as high as possible)
- To be used when the **class distribution is balanced** and classes are **equally important** from a problem-domain point of view

- **FPR (False Positive Rate)**
  - *What is the proportion of misclassified instances out of all the true negatives?*
  - $\frac{FP}{FP+TN}$ (should be as low as possible)

- **RECALL or TPR (True Positive Rate)**
  - *What proportion of actual positives is identified correctly?*
  - $\frac{TP}{TP+FN}$ (should be high as possible)

- **PRECISION**
  - *What proportion of positive predictions is actually correct?*
  - $\frac{TP}{TP+FP}$ (should be high as possible)

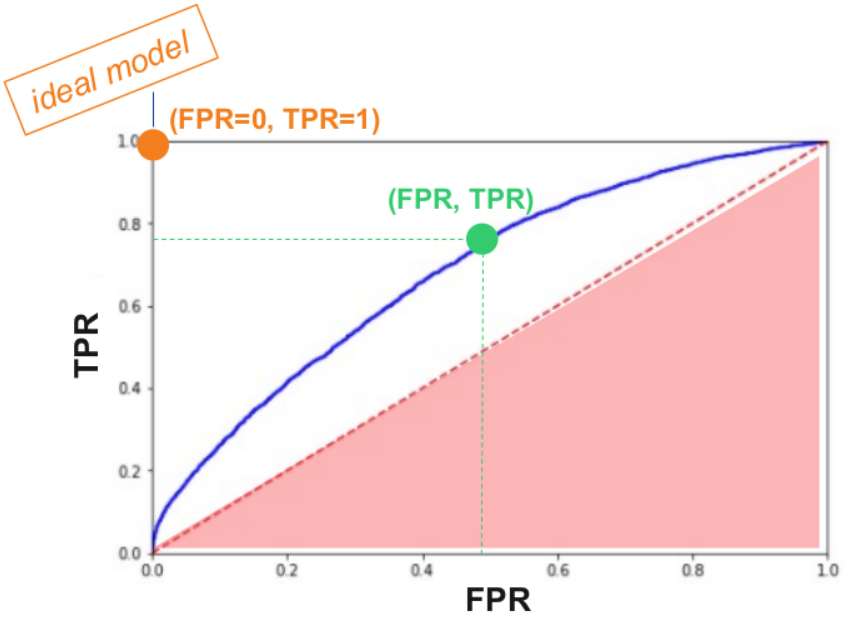
Model evaluation



- The **Type I error** produces a false positive
- The **Type II error** produces a false negative and it is also known as *error of omission*

Model evaluation

- The **ROC (Receiver Operating Characteristic) curve** is a graph showing the performance of a classification model as the classification threshold varies
- It is typically used either to select the best threshold for a given problem, or to compare different models, or to compute the **AUC (Area Under the Curve)**, which is an aggregate model evaluation metrics



The dotted red line is usally plotted to represent the **random model**: extract a random number i from a uniform distribution: $y = \begin{cases} 0, & i < p \\ 1, & i \geq p \end{cases}$

- Any ROC curve that dominates the random model curve (upper triangle) is then a good model