Data Mining

Predictive Modelling

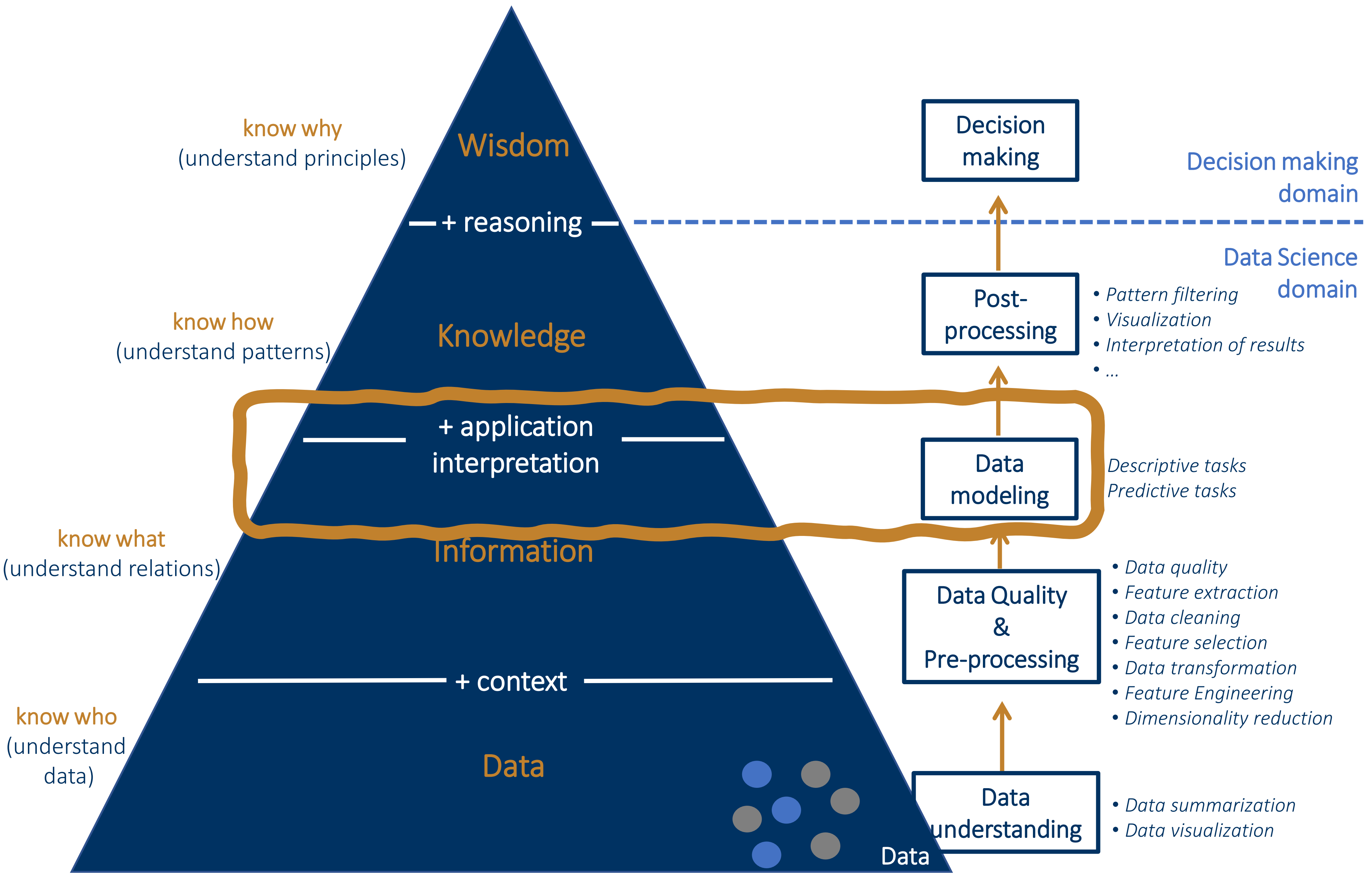
Raquel Sebastião

Departamento de Eletrónica, Telecomunicações e Informática

Universidade de Aveiro

raquel.sebastiao@ua.pt

2022/2023



know why
(understand principles)
Wisdom
+ reasoning
know how
(understand patterns)
Knowledge
+ application
interpretation
know what
(understand relations)
Information
+ context
know who
(understand
data)
Data
Data
Decision making
Decision making domain
Data Science domain
Post-processing
• Pattern filtering
• Visualization
• Interpretation of results
• …
Data modeling
Descriptive tasks
Predictive tasks
Data Quality & Pre-processing
• Data quality
• Feature extraction
• Data cleaning
• Feature selection
• Data transformation
• Feature Engineering
• Dimensionality reduction
Data understanding
• Data summarization
• Data visualization

Contents

Machine Learning

A computer program is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P** if its performance at tasks in **T**, as measured by **P**, improves with experience **E**.”
Mitchell, T. (1997)

“Machine Learning is the systematic study of algorithms and systems that improve their knowledge or performance with experience”
Flach, P. (2012)

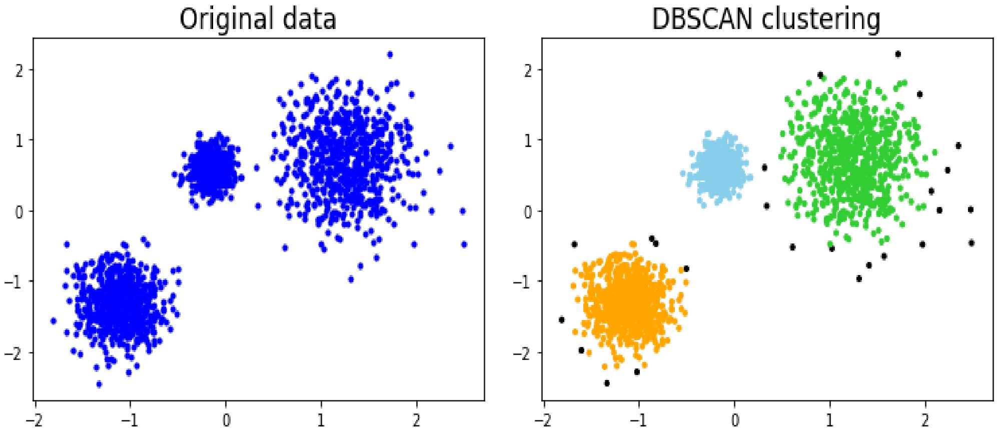
Goal:

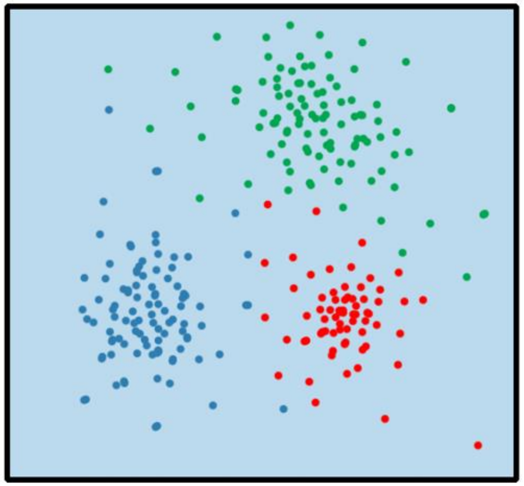
- Build models that capture the knowledge from observed cases to make inferences in unseen cases. In principle, more observations should lead to better models!

Machine Learning: Tasks

Unsupervised Learning: no target label/value is associated to each object (class labels of the training data are unknown)

- Given a set of observations/objects, the goal of learning is to obtain possible groups/clusters in the data (structure of the data)



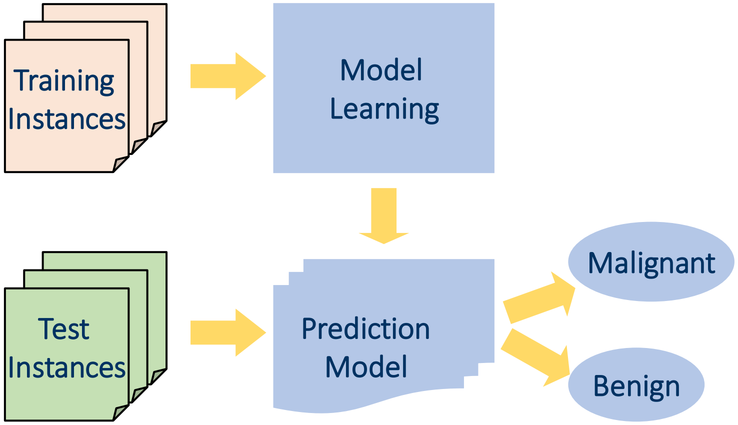


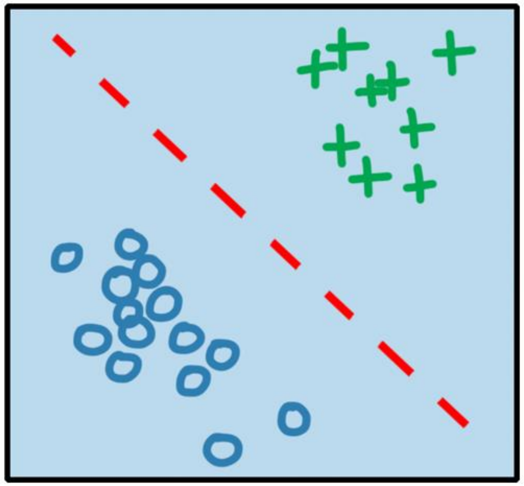
Machine Learning: Tasks

Supervised Learning: there is a target label/value that is associated to each object/example

- the goal of the learning task is to learn a function (model) that maps each example with its target variable

→ Predictive Modelling: New data is classified based on the models built from the training set

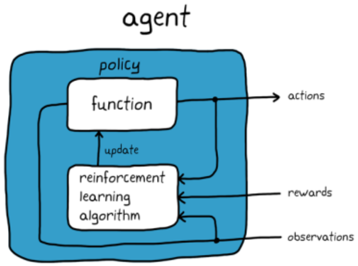




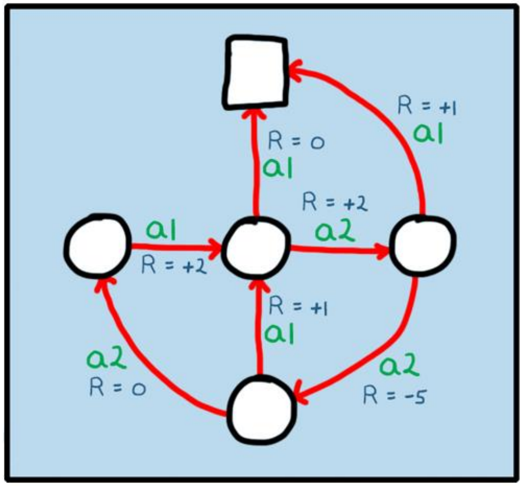
Machine Learning: Tasks

Reinforcement Learning: the learning algorithm builds examples from a set of rules; then an iterative process is used to improve (or "reinforce") the set of examples until some evaluation criterion is good enough.

- **example**: parking a vehicle using an automated driving system: teach the vehicle computer (agent) to park in the correct parking spot with reinforcement learning



https://www.mathworks.com/



Machine Learning

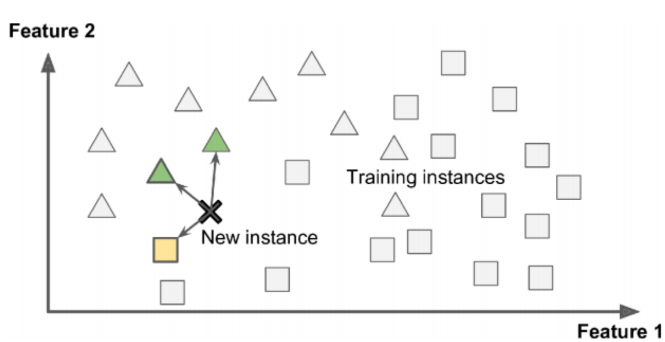
What are the main learning paradigms?

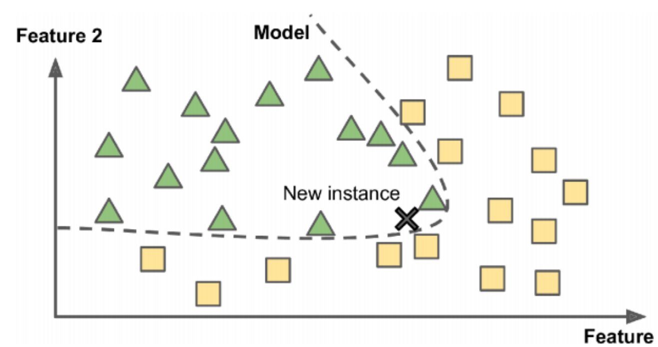
- Batch learning
- Online learning

Is there an assumption on data distribution?

- Parametric
- Non-parametric

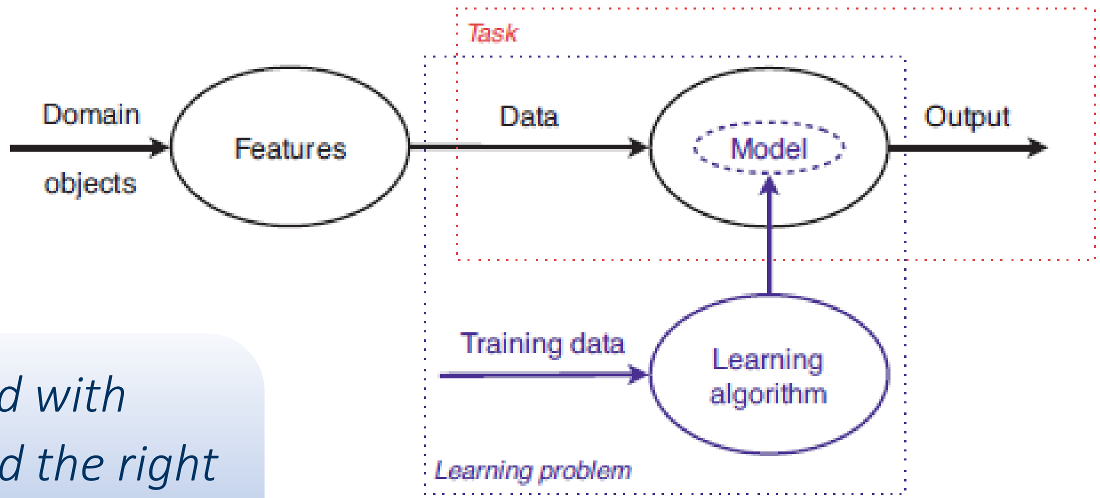
What to do when new data points arrive?





Machine Learning

- *Tasks are addressed by models*
- *Learning problems are solved by learning algorithms*
- *Learning algorithms produce models (when applied to training data)*



"machine learning is concerned with using the right features to build the right models that achieve the right tasks"

Flach, P. (2012)

Contents

- Machine Learning
- Predictive Modelling
  - Pipeline
  - Tasks
  - Prediction models – approaches
- Classification
- Summary

Predictive Modelling

Example: **Clinical diagnosis**

- Given a data set containing diverse features extracted from X-rays or MRI scans of several patients and the diagnosis

ID	r-mean	t-mean	per-mean	ar-mean	sm-mean	cm-mean	cn-mean	nc-mean	(...)	diagnosis
842302	17.99	10.38	122.8	1001	0.1184	0.2776	0.3001	0.1471	(...)	M
842517	20.57	17.77	132.9	1326	0.08474	0.07864	0.0869	0.07017	(...)	M
84300903	19.69	21.25	130	1203	0.1096	0.1599	0.1974	0.1279	(...)	M
84348301	11.42	20.38	77.58	386.1	0.1425	0.2839	0.2414	0.1052	(...)	M
84358402	20.29	14.34	135.1	1297	0.1003	0.1328	0.198	0.1043	(...)	M
843786	12.45	15.7	82.57	477.1	0.1278	0.17	0.1578	0.08089	(...)	M
844359	18.25	19.98	119.6	1040	0.09463	0.109	0.1127	0.074	(...)	M
84458202	13.71	20.83	90.2	577.9	0.1189	0.1645	0.09366	0.05985	(...)	M
844981	13	21.82	87.5	519.8	0.1273	0.1932	0.1859	0.09353	(...)	M
84501001	12.46	24.04	83.97	475.9	0.1186	0.2396	0.2273	0.08543	(...)	B
845636	16.02	23.24	102.7	797.8	0.08206	0.06669	0.03299	0.03323	(...)	B
84610002	15.78	17.89	103.6	781	0.0971	0.1292	0.09954	0.06606	(...)	B
846226	19.17	24.8	132.4	1123	0.0974	0.2458	0.2065	0.1118	(...)	M
846381	15.85	23.95	103.7	782.7	0.08401	0.1002	0.09938	0.05364	(...)	M
84667401	13.73	22.61	93.6	578.3	0.1131	0.2293	0.2128	0.08025	(...)	M
84799002	14.54	27.54	96.73	658.8	0.1139	0.1595	0.1639	0.07364	(...)	M
848406	14.68	20.13	94.74	684.5	0.09867	0.072	0.07395	0.05259	(...)	M
84862001	16.13	20.68	108.1	798.8	0.117	0.2022	0.1722	0.1028	(...)	M

- **Predict** the correct diagnosis for a new patient for which we know the features of X-ray and MRI scans

Predictive Modelling

Prediction Models are learnt on the basis of the assumption that there is an **unknown mechanism that** maps the characteristics/features of the observations into **conclusions**

- The goal of prediction models is to discover this mechanism

Clinical diagnosis

- how features/characteristics of the cells in the X-rays or MRI scans influence the diagnosis
- Use a data set with “examples” of this mapping, e.g., this patient had cells’ characteristics c1, c2, …, cD and the diagnosis was that tumor cells were benign

- Using the **available data**, obtain a **good approximation** of the unknown function that maps the **observation descriptors** into the **conclusions**

Predictive Modelling

- **Descriptors**: set of variables that describe the properties (features, attributes, predictors) of the objects in the data set (X_1, X_2, … ,X_D)
- **Target variable**: what we want to predict/conclude regarding the observations (Y)
- The **goal** is to obtain an approximation of the function

$$Y = f\,(X_1, X_2, \dots ,X_D)$$

- It is assumed that Y is a variable whose values depend on the values of the variables which describe the objects.

We just do not know how!

Predictive Modelling

- Given a set of predictor variables **X** and a target variable **Y**, there is a function $\boldsymbol{f}$, such that $\boldsymbol{f}$(X) = Y

Predictor variables → f → Target variable

- Since f is unknown, the goal is to learn the best approximation to $\boldsymbol{f}$, $\boldsymbol{h_\theta}$, so that the target labels/values can be obtained from the input data set

Data set → $\boldsymbol{h_\theta}$ (model)

- With the built model $\boldsymbol{h_\theta}$, it is possible to make predictions for new, unseen observations!

New (unseen) Data set → $\boldsymbol{h_\theta}$ (model) → Predicted target labels/values

Predictive Modelling: pipeline

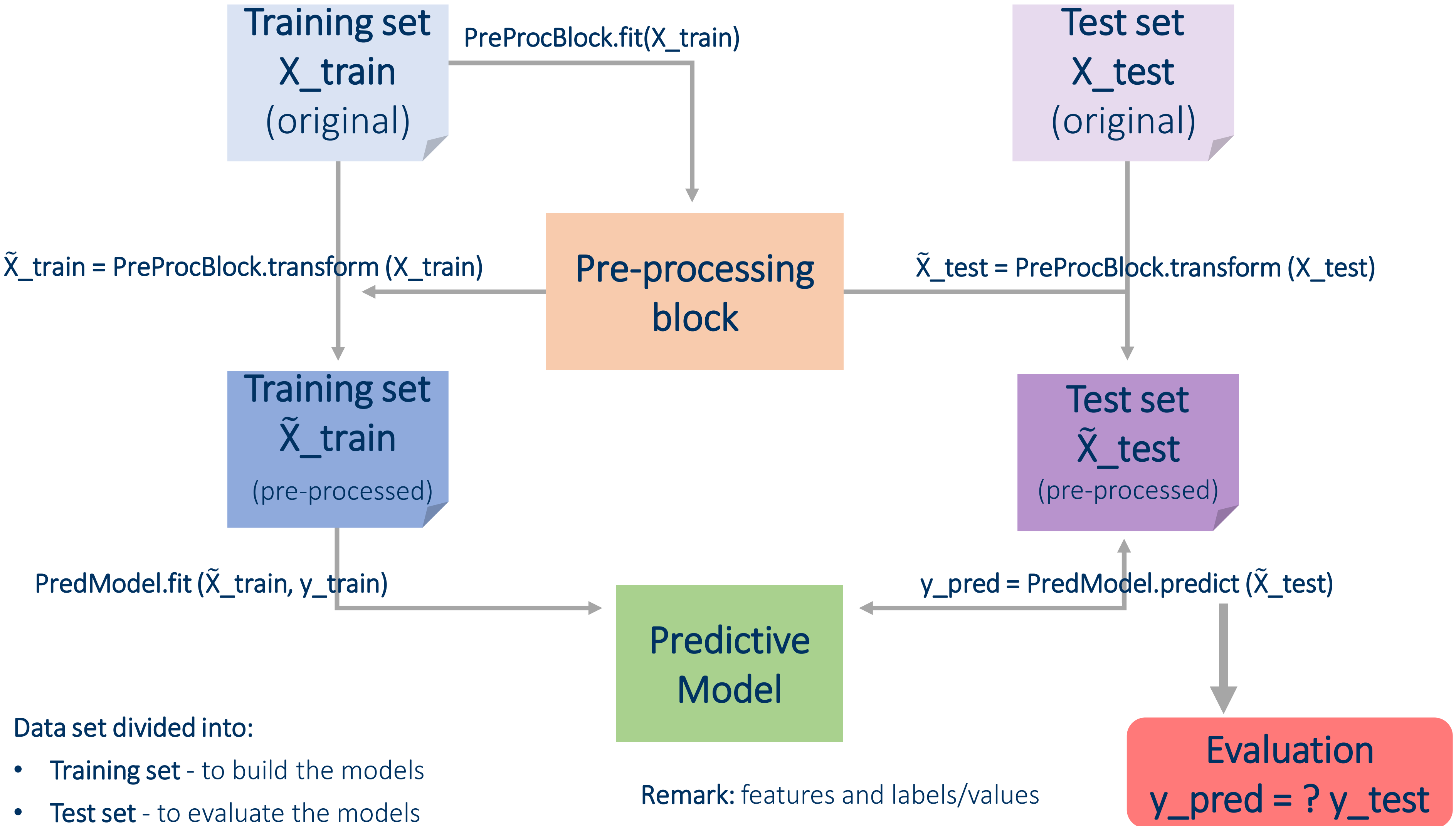
Pre-processing block

- To deal with heterogenous data
  - numerical and categorical
- To deal with different features’ ranges
  - Normalization/standardization
- Feature Engineering
- Data reduction
  - Feature selection
  - Projection of feature vector

Predictive model

- Classification model
  - Target labels
- Regression model
  - Target values

Predictive Modelling: pipeline



Data set divided into:

- **Training set** - to build the models
- **Test set** - to evaluate the models

Remark: features and labels/values

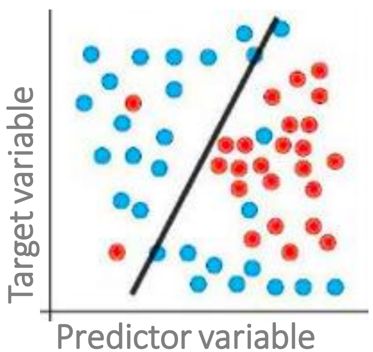
- **X** – matrix with feature vectors
- **y** – vector with labels/values

Predictive Modelling

Underfitting: model is too simple to capture patterns in data

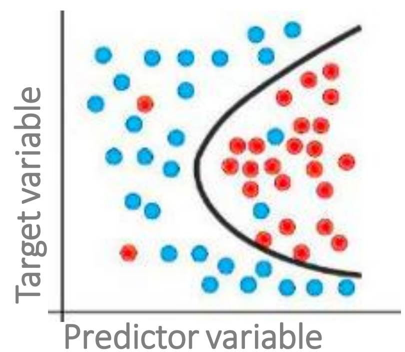
Overfitting: model performs well on training data but does not generalize well to unseen data

Underfitting
(high bias)



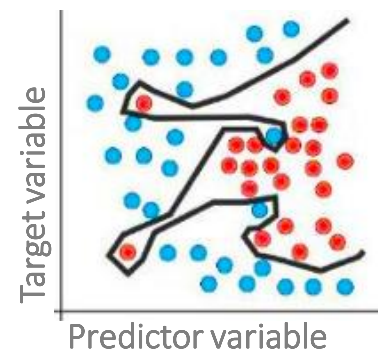
- High training error
- High test error

Optimal
(good compromise)



- Low training error
- Low test error

Overfitting
(high variance)



- Low training error
- High test error

Predictive Modelling

- Predictive models have two main uses:

1. Prediction:

    - use the obtained models to make predictions regards the target variable of new cases given their descriptors.

2. Comprehensibility:

    - use the models to better understand which are the factors that influence the conclusions.

Predictive Modelling - tasks

Types of Prediction tasks

Depending on the type of the target variable **Y** we may be facing two different types of prediction models:

- Classification tasks
  - the target variable **Y** is nominal, e.g., medical diagnosis - given the symptoms of a patient try to predict the diagnosis
- Regression tasks
  - the target variable **Y** is numeric e.g., forecast the market value of a certain asset given its characteristics

Prediction Models

There are many approaches that can be used to obtain **prediction models** based on a data set

Independently of the **pros** and **cons** of each alternative, all have some key characteristics:

- They assume a certain functional form for the unknown function $f()$
- Given this assumed form, the methods try to obtain the best possible model based on:
  - the given data set
  - a certain preference criterion that allows comparing the different alternative model variants

Prediction Models – approaches

Geometric approaches

- **Distance-based:** kNN
- **Linear models:** linear discriminants, logistic regression, perceptron,

SVM (w. linear kernel)

Probabilistic approaches

- naive Bayes, logistic regression

Logical approaches

- classification or regression trees, rules

Optimization approaches

- neural networks, SVM

Sets of models (ensembles)

- random forests, adaBoost

Prediction Models – approaches

Or…

Linear approaches

- linear discriminants, logistic regression, perceptron, SVM (w. linear kernel)

Non-linear approaches

- kNN, naive Bayes, classification trees, (w. non-linear kernel) SVM, neural networks

Sets of models (ensembles)

- random forests, adaBoost

Prediction Models

These different approaches entail different compromises in terms of:

- **assumptions** on the unknown form of dependency between the target and the predictors
- **computational complexity** of the search problem
- **flexibility** in terms of being able to approximate **different types of functions**
- **interpretability** of the resulting model
- . . .

Contents

- Machine Learning
- Predictive Modelling
- Classification
  - Problem definition
  - Binary and multiclass classification
  - Evaluation metrics
- Summary

Classification: problem definition

Setting

- Given a **data set** $\{(\mathrm{x}_i,\mathrm{y}_i)\}_{i=1}^{N}$, where each **object** is represented by a **D+1**–tuple: (D–dim) feature vector $\mathbf{x}_i = \left[x_i^1 \quad x_i^2 \quad \ldots x_i^D\right]^T \in \mathbb{R}^D$ and the corresponding **label** $\mathrm{y}_i \in \boldsymbol{Y}$
- There is an **unknown** function: $\mathrm{Y} = f\,(\mathrm{X})$

Goal

Learn the model that yields the best approximation of the unknown function ***f*()**

Approach

- Assume a functional form $\boldsymbol{h_\theta}(\mathbf{x})$ for the unknown function ***f*()**, where $\boldsymbol{\theta}$ are a set of parameters
- Assume a preference criterion over the space $\boldsymbol{\theta}$ of possible parameterizations of *h*()
- Search for the "best" $\boldsymbol{h_\theta}$ (according to the criterion and the data set)

Classification: learn/build a classifier

Given a training set $\{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^{N}$:

- $(\mathrm{x}_i, \omega_{c_i})$ - the label is **symbolic** or **categorical** (ex.: binary {malignant, benign})
- (x_i, t_i) - label is **numerical** (ex.: binary {−1, 1}, multiclass {0, 1, 2}))

Learning/Training phase

- find the "best" approximation to f, $\boldsymbol{h_\theta}$

Testing phase: Given a test set (data not included in the training set)

- Study the accuracy of the model (performance of the classifier).

Usually, the available data set is divided into training and test sets

Classification: binary classification problem

- when the target variable only assumes **two possible labels/values** (classes), usually referred as positive and negative class
  - e.g., flu: yes/no, credit: yes/no

Output of a classification model:

- class assigned to a case
- score / probability of case belonging to a certain class; a decision threshold is chosen to establish the predicted class
- e.g., if $\boldsymbol{h_\theta}(\mathbf{x}_i) \geq \mathbf{0.5}$ then is positive example, otherwise is negative

Classification: multiclass classification problem

- when the target variable assumes more than two possible classes
  - e.g., insurance risk: low, medium, high

Some algorithms **cannot handle multiclass**; the alternative is to combine several binary classifiers

- one-vs-all: train a model for each class; for k classes, we have k binary classifiers
- one-vs-one: train a model for each pair of classes; for k classes, we have $k(k-1)/2$ classifiers

Classification: performance of models

Metrics for evaluating a model's performance

- How can we measure the performance of a model?
  - **Accuracy, precision, recall, F-measure**

Use **test set (unseen data)** of class-labeled tuples instead of training set when assessing performance

Strategies for estimating a model's performance

- How can we evaluate the performance of a model?
  - **Holdout method, Cross-validation, Bootstrap method**

Comparing models:

- **Statistical Hypothesis Testing**

Classification: evaluation metrics

Goal: Obtain reliable estimates of performance and compare different classification models

- **Error Rate**: proportion of predictions that are incorrect

$$L_{0/1} = \frac{1}{N}\sum_{i=1}^{N} l\,(\hat{y}_i, y_i)$$

  - N is the number of cases
  - $\hat{y}_i = h_\theta(x_i)$ is the predicted class by the model for the object *i*
  - y_i is the respective true class
  - $l()$ is loss-function such that $l(\hat{y}_i, y_i) = 0$, *if* $\hat{y}_i = y_i$, and 1 otherwise

- **Accuracy** = 1 - Error Rate

Classification: evaluation metrics

- Confusion matrices
  - A square $c \times c$ matrix*, where c is the number of class values of the problem
  - A special kind of contingency table, with two dimensions ("true class" and "predicted class")
  - Each value reports the number of predictions made by the model of a class for a given true class

True class \ Predicted class	C_1	C_2	C_3
C_1	$n_{c1,\,c1}$	$n_{c1,\,c2}$	$n_{c1,\,c3}$
C_2	$n_{c2,\,c1}$	$n_{c2,\,c2}$	$n_{c2,\,c3}$
C_3	$n_{c3,\,c1}$	$n_{c3,\,c2}$	$n_{c3,\,c3}$

  - $n_{ci,\,cj}$ indicates # of objects in class i that were predicted by the model as class j

*(may have extra rows/columns to provide totals)

Classification: evaluation metrics

- Confusion matrix for a binary classification problem

		Predicted class	
		P	***N***
True class	***P***	**TP** True Positive	**FN** False Negative
	N	**FP** False Positive	**TN** True Negative

- TP: hit
- FN: miss (type II error)
- FP: false alarm (type I error)
- TN: correct rejection

Classification: evaluation metrics

- Confusion matrix for a binary classification problem

		Predicted class	
		P	***N***
True class	***P***	**TP** True Positive	**FN** False Negative
	N	**FP** False Positive	**TN** True Negative

- Accuracy $= \dfrac{TP+TN}{TP+FN+FP+TN}$ (proportion of correct predictions)
- Precision $= \dfrac{TP}{TP+FP}$ (proportion of correct positive predictions)
- Recall $= \dfrac{TP}{TP+FN}$ (proportion of positive objects correctly predicted)

Classification: evaluation metrics

- Confusion matrix for a <u>binary classification</u> problem

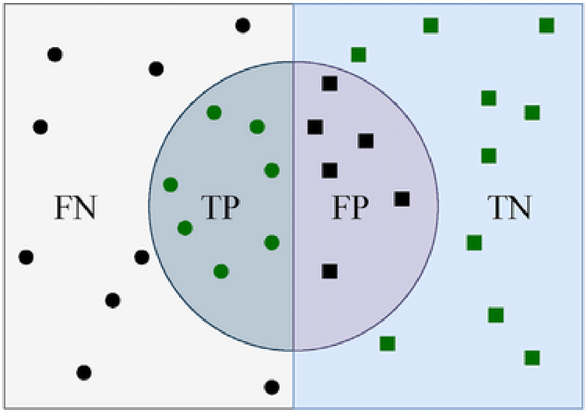
		Predicted class	
		P	***N***
True class	***P***	**TP** True Positive	**FN** False Negative
	N	**FP** False Positive	**TN** True Negative

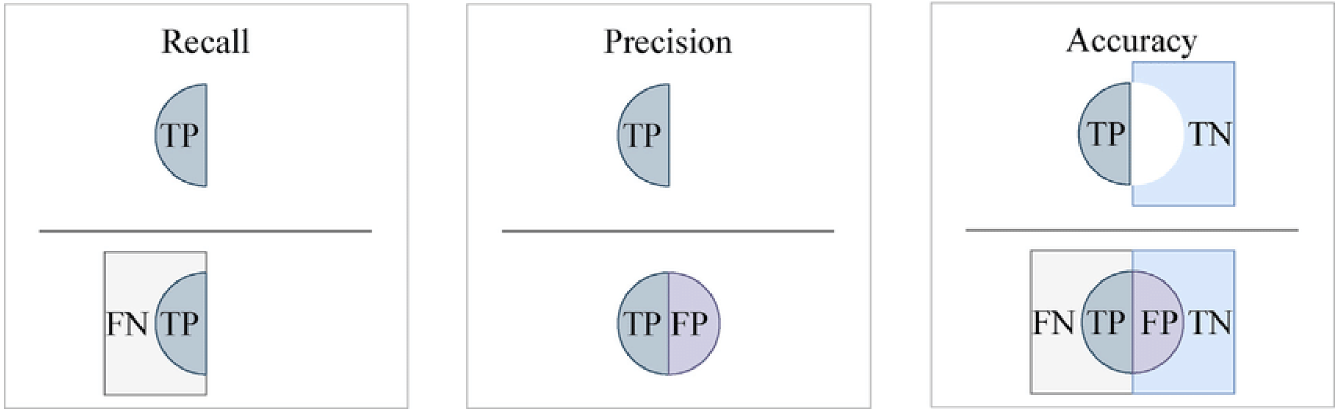
Recall

Precision

- Precision: proportion of correct positive predictions
- Recall: proportion of positive objects correctly predicted

Classification: evaluation metrics





Maleki, F., et al. (2020) Overview of Machine Learning Part 1. *Neuroimaging Clinics of North America*. 30. e17-e32. 10.1016/j.nic.2020.08.007

Classification: evaluation metrics

- Precision/Recall tradeoff
  - increasing precision may reduce recall and vice versa.
- It is easy to obtain **100% Recall**: always predict **Positive**

- F-measure: weighted combination of Precision and Recall

$$F_\beta = \frac{1}{\alpha \cdot \frac{1}{Precision} + (1-\alpha) \cdot \frac{1}{Recall}} = \frac{(\beta^2+1) \times Precision \times Recall}{\beta^2 Precision + Recall}$$

where β controls the weighted combination

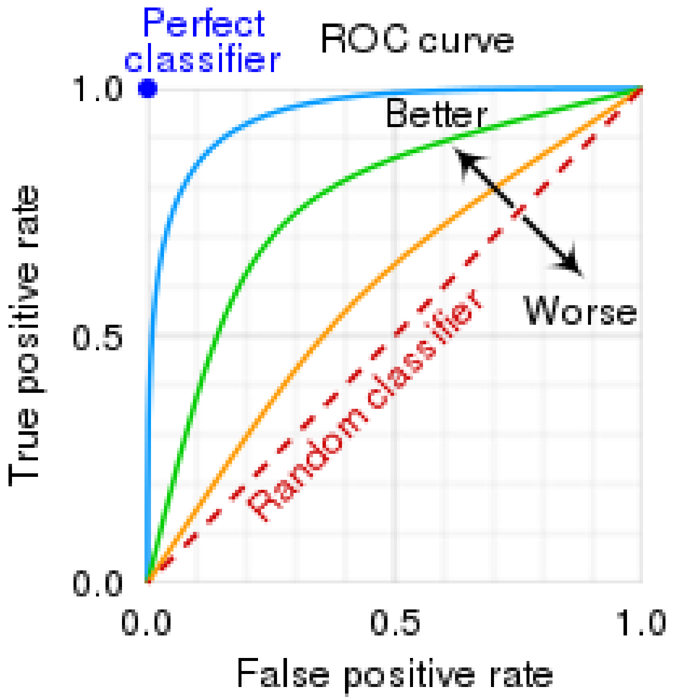
- if β = 1 then is the harmonic mean of ***Precision*** and ***Recall***
- when $\beta \rightarrow 0$ the weight of ***Recall*** decreases.
- when $\beta \rightarrow \infty$ the weight of ***Precision*** decreases

Classification: evaluation metrics

- Some classifiers may require higher precision:
  - e.g., classifier that detects videos that are safer for kids. Keep high precision with only safe videos and may reject other videos that are good (low recall)
- Some classifiers may require higher recall:
  - e.g., classifier that detects disease on image samples. High recall to get all disease samples. Can handle some false positives (lower precision) that later will be double checked by doctors.
- There are several tradeoff measures: e.g., **G-mean**, **IBA** (Index of Balanced Accuracy)

Classification: evaluation metrics

- Receiver Operator Characteristic (ROC) curve is a common tool for evaluation of binary classifiers.
- Plots TPR (Recall) vs FPR = FP/(TN + FP) for different decision thresholds.
- Area Under the Curve (AUC) allows to compare classifiers
  - A perfect classifier has AUC of 1
  - A random classifier has AUC of 0.5



en.wikipedia.org

Summary

- Machine Learning
- Predictive Modelling
  - Pipeline
  - Tasks
  - Prediction models - approaches
- Classification
  - Problem definition
  - Binary and multiclass classification
  - Evaluation metrics

Bibliography

Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar, *Pearson*, 2019 (chap 3)

Data Mining, the Textbook, Charu C. Aggarwal, *Springer*, 2015 (chap 6)

Machine Learning: The Art and Science of Algorithms That Make Sense of Data, P. Flach, *Cambridge University Press*, 2012 (ch 1, 2, 11)

Data Mining: Practical Machine Learning Tools and Techniques, I. H. Witten, E. Frank, M. A. Hall, C. J. Pal, *Morgan Kaufmann*, 2017 (ch 3, 5)