Supervised learning

LSSDS 2023

Mauricio Cerda, Eng. PhD

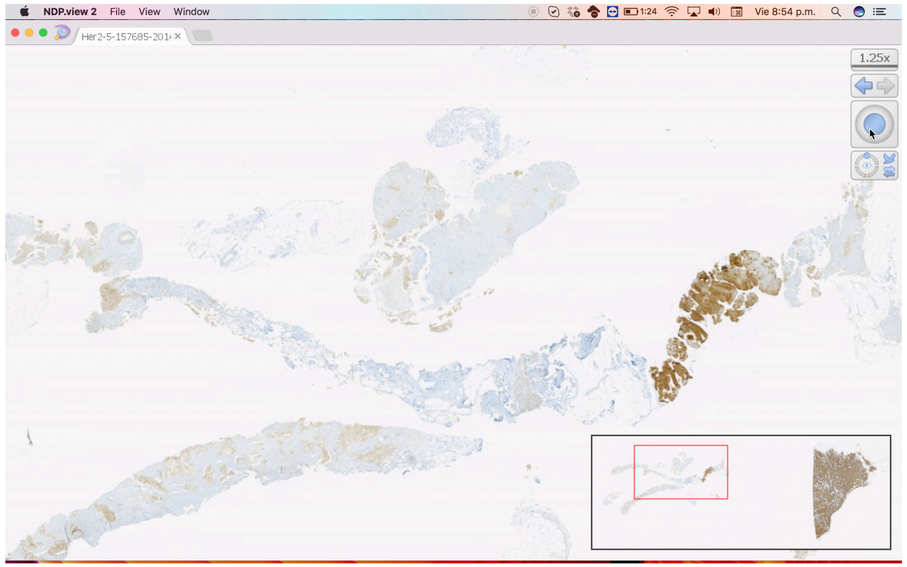
Programa de Biología Integrativa
I.C.B.M., Facultad de Medicina, Universidad de Chile

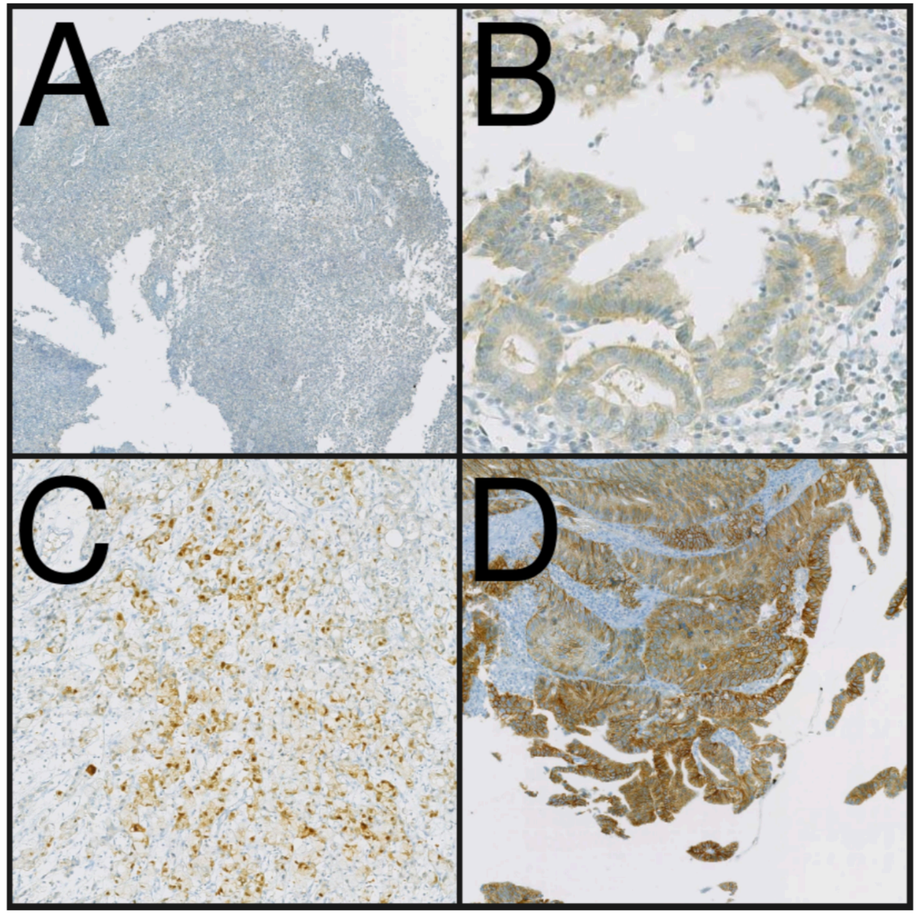
Outline

- Examples: gastric cancer, schizophrenia and language.
- Problem definitions, metrics, how to train.
- Algorithms: perceptron, SVM, Random Forest

Example: gastric cancer

- Classify groups defined by examples.
- Classify groups and get a set of rules that can be interpreted.



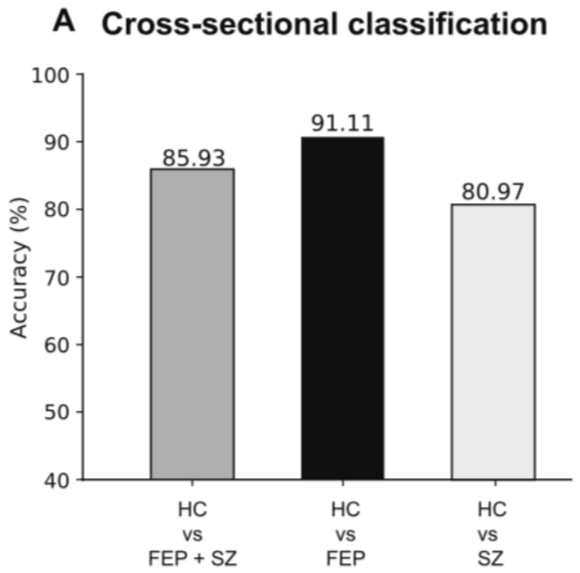


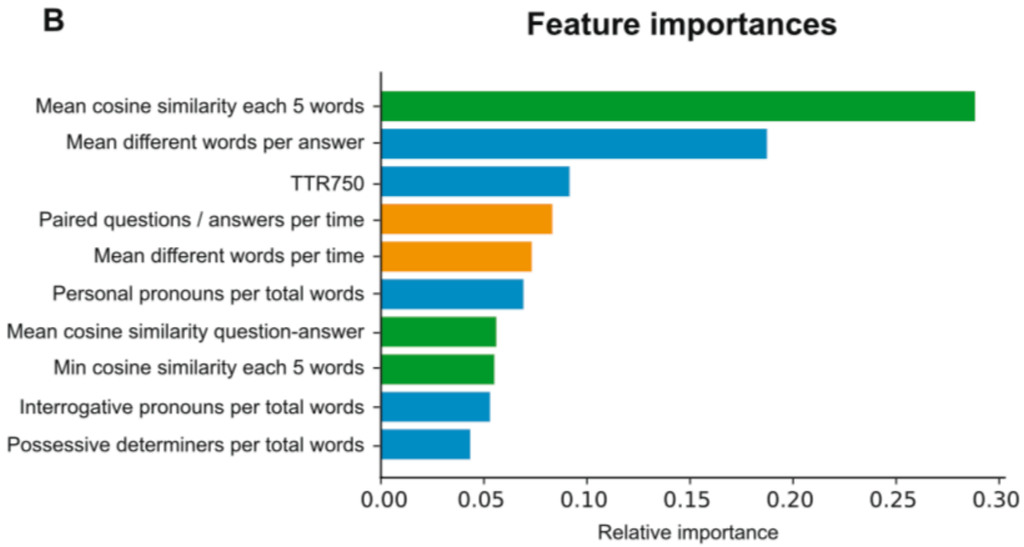
Her2 en cancer gástrico (estudio PRECISO, Inst. Nacional del Cancer, Chile)

Example: Schizophrenia and language

- Classify patients speech and assign a group (control, FEP, SZ).
- Classify if FEP will be really SZ or other pathologies.

"yo trabajé ocho años en en una barraca de fierros ehh de pioneta y eso como que ese era mi laburo anteh ehh lo que más me gusta trabajar eh eso cachai ese eh mi mi mi mi ambiente"

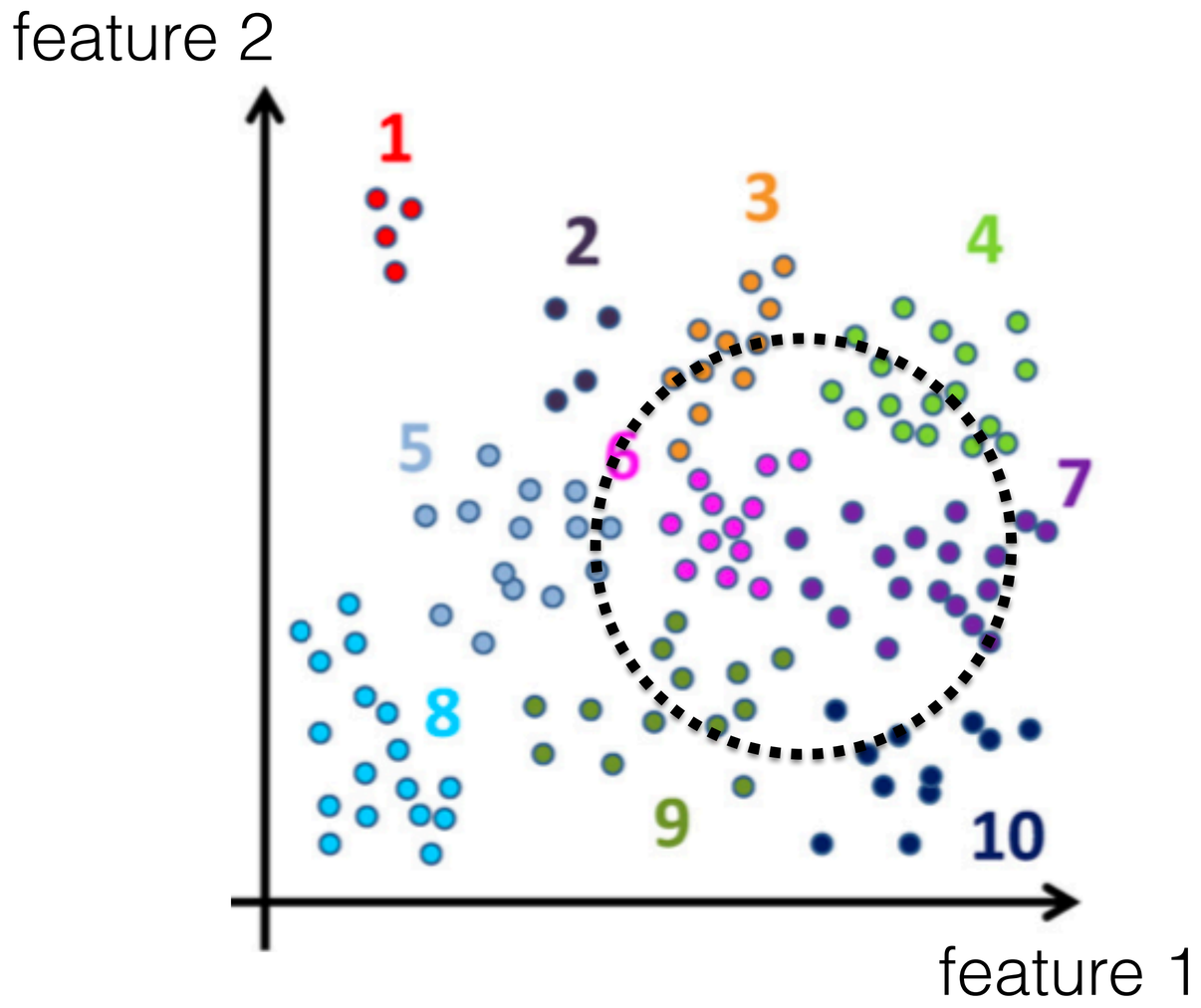




Her2 en cancer gástrico (estudio PRECISO, Inst. Nacional del Cancer, Chile)

Classification task

- Learn a function of input (features) and output (classes).

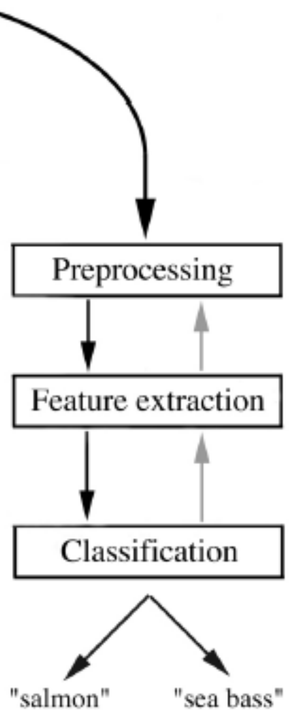


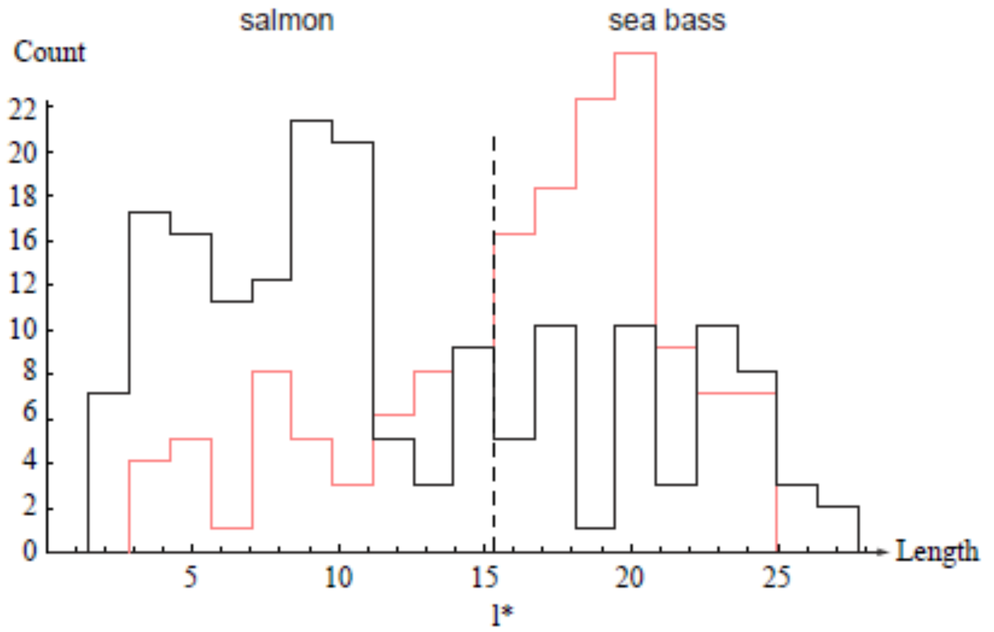
Classification: definitions

- Regression: 1 input, 1 numerical value.
- **Classification**: 1 input, 1 label.

- Training set -> model parameters.
- Validation set -> hyper-parameters stunning.
- Test set -> unbiased evaluation.

Classification: 1D

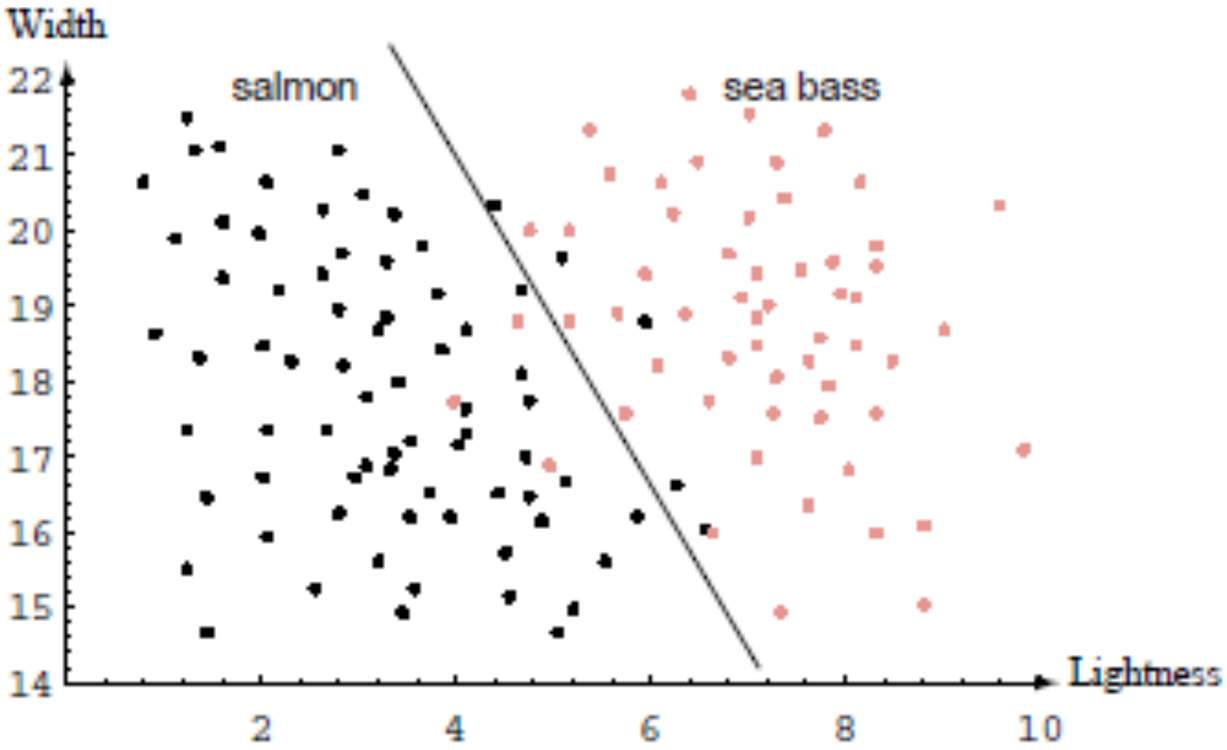




Sea bass=lubina

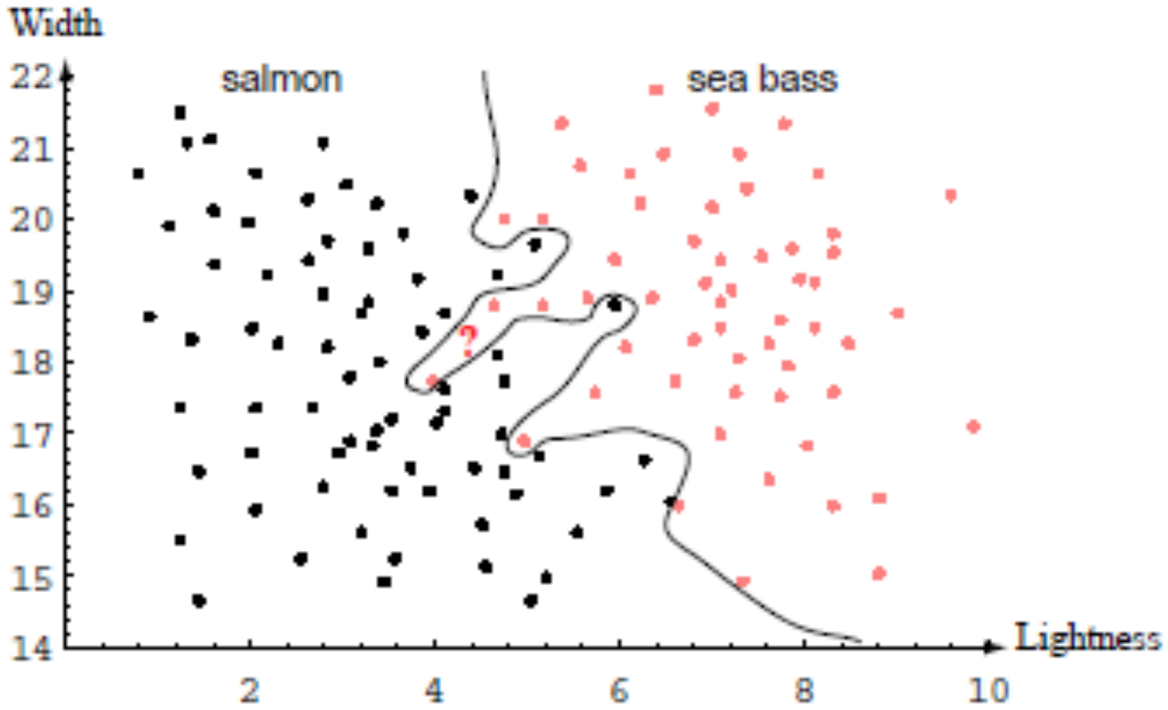
Classification: 2D

- In 2D a line divides both classes.
- In 3D an hyperplane.



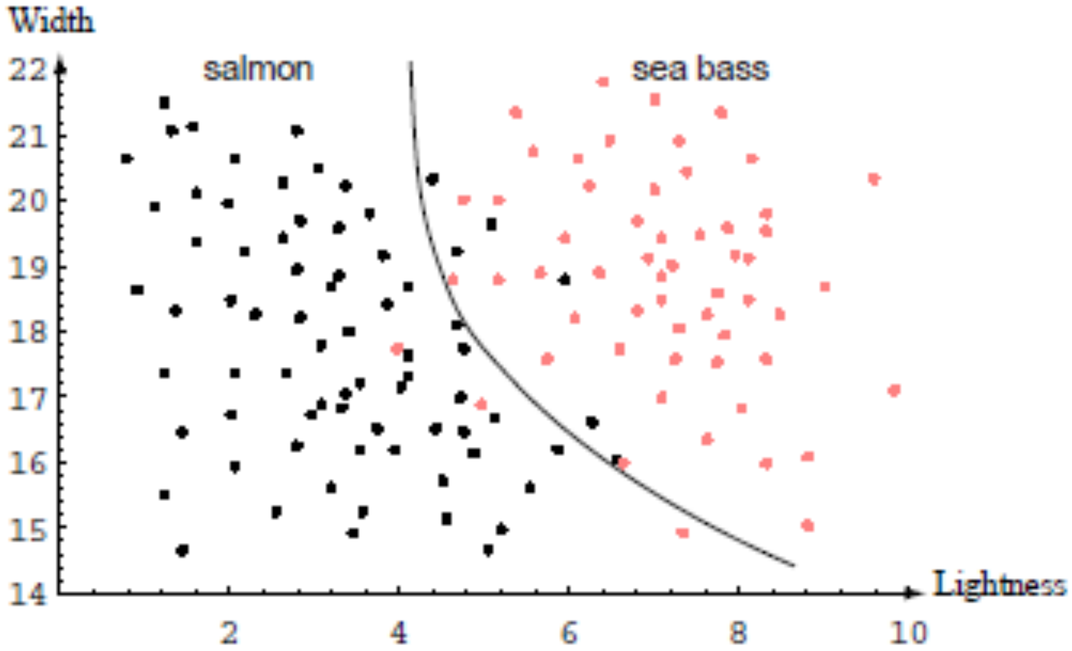
Classification: 2D

- If we do a perfect line fit, what problem do you see?
- How the **“?”** Is classified ?



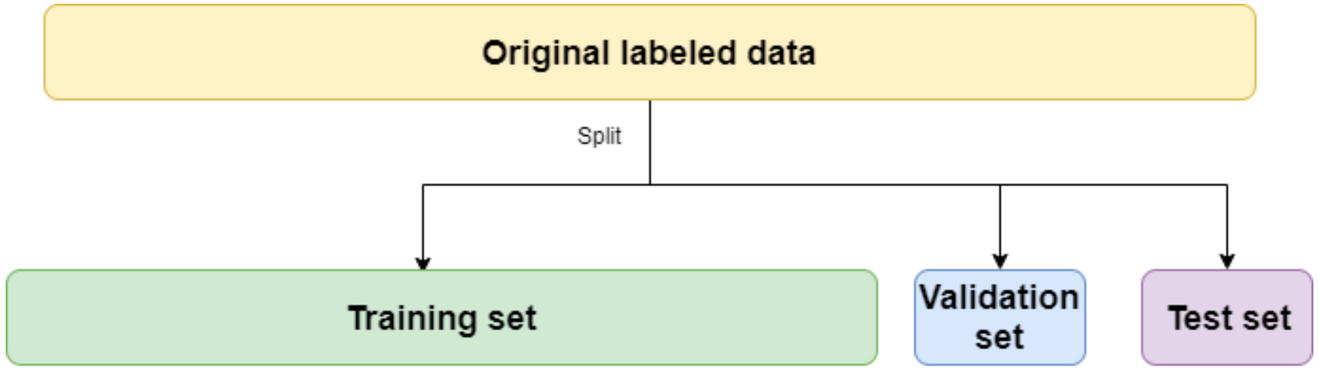
Classification: 2D

- We want a line/plane fit to data, without overfitting (better generalization).



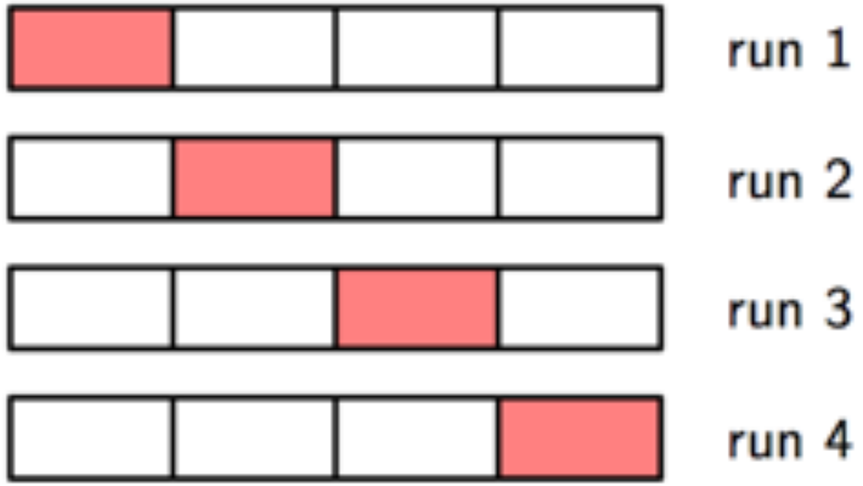
Classification: strategies

- As we saw, it is a bad idea to evaluate model in training data.
- Usual to split dataset into: (i) trainning, (ii) validate, (iii) test.



Classification: strategies

- When not so much data is available we split data multiple times: cross-validation.
- Divide dataset into S subsets. S-1 sets for trainning, the rest to validate/test.



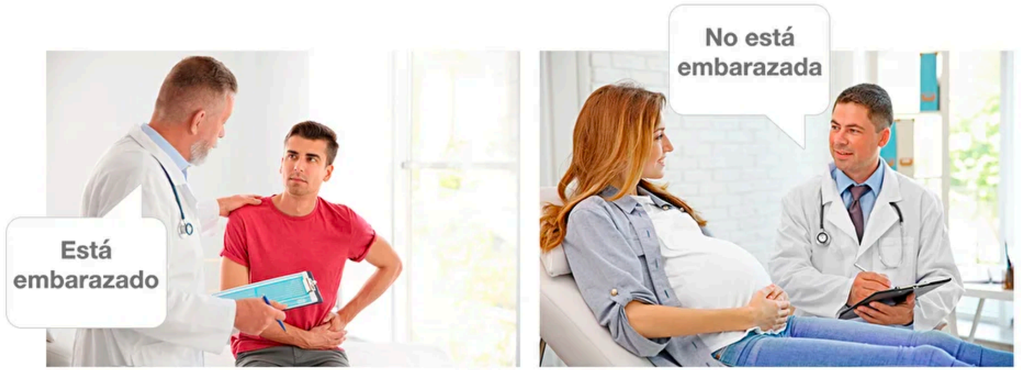
Classification: metrics

- How to measure classification quality?
- For a rate disease, if the classifier says “healthy” all the time, it will be right 99% of case (not very useful).
- In a binary classifier, we can identify error and matches.

		Actual	
		Positive	Negative
Predicted	Positive	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

Error tipo I
(falso positivo)

Error tipo II
(falso negativo)



Classification: results

ID	Clase	Predicccción	Resultado
1	Cáncer	No-Cáncer	FN
2	Cáncer	No-Cáncer	FN
3	No-Cáncer	No-Cáncer	TN
4	Cáncer	Cáncer	TP
5	No-Cáncer	No-Cáncer	TN
6	Cáncer	Cáncer	TP
7	No-Cáncer	No-Cáncer	TN
8	Cáncer	Cáncer	TP
9	Cáncer	Cáncer	TP
10	Cáncer	Cáncer	TP
11	No-Cáncer	No-Cáncer	TN
12	Cáncer	No-Cáncer	FN
13	No-Cáncer	No-Cáncer	TN
14	No-Cáncer	No-Cáncer	TN
15	No-Cáncer	No-Cáncer	TN
16	No-Cáncer	No-Cáncer	TN
17	No-Cáncer	Cáncer	FP
18	Cáncer	Cáncer	TP
19	No-Cáncer	No-Cáncer	TN
20	No-Cáncer	Cáncer	FP

TP=6, FP=2, TN=9, FN3

Classification: metrics

- Accuracy
- Proportion between true and total. To be used without imbalance.
- If 990 healthy (negativos) and 10 positives (Pathology). Accuracy will be 0.99 if classifier simply says “**healthy**” all the time!

$$\frac{TP+TN}{TP+TN+FP+FN}$$

$$\frac{6+9}{6+9+2+3} = 0{,}75$$

Classification: metrics

- Precision (P)
- True positives over all positives.
- High values imply **few false positives.**

$$P = \frac{TP}{TP+FP}$$

$$P = \frac{TP}{TP+FP} = \frac{6}{6+2} = 0{,}75$$

Classification: metrics

- Recall (R)
- True positives over all positives.
- High values imply **few false negatives** (important for screening).

$$R = \frac{TP}{TP+FN}$$

$$R = \frac{TP}{TP+FN} = \frac{6}{6+3} = \frac{2}{3}$$

Classification: metrics

- A classifier that always says positive will have R=1 but low P.
- A classifier that always says negative will have low R but P=1
- F1: Combines P and R
- From 0 to 1 (better).

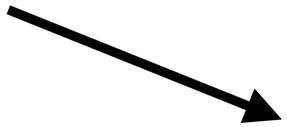
$$F_1 = 2 \times \frac{P \times R}{P + R} = 0{,}706$$

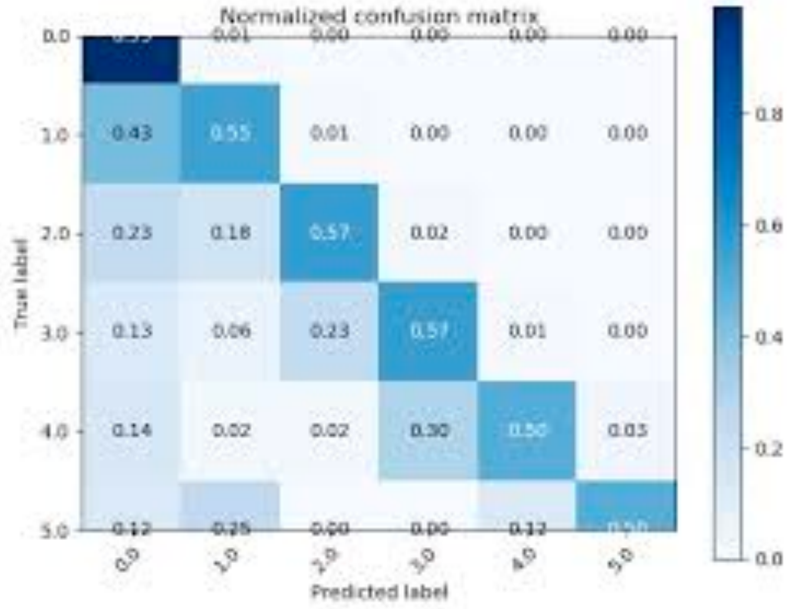
$$\mathrm{F1} = 2\,\frac{\mathrm{precision} * \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

Classification: metrics

- The 2x2 matrix can be extended to many clases: confusion matrix.

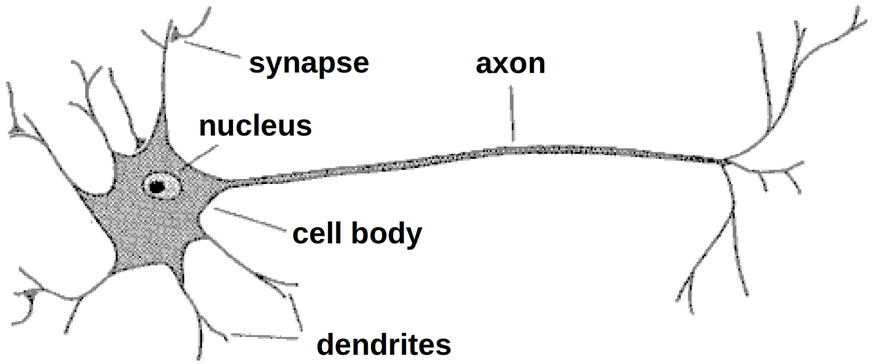
	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

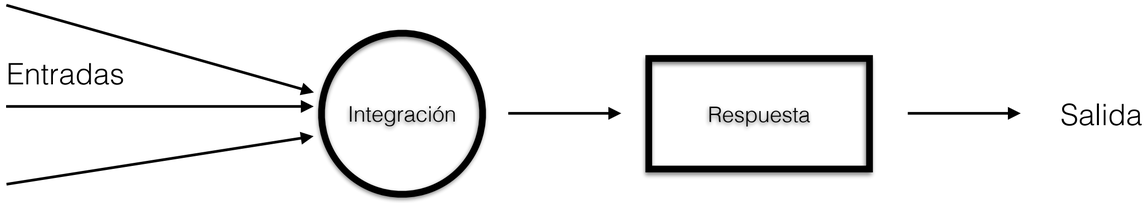




Classification: algorithms

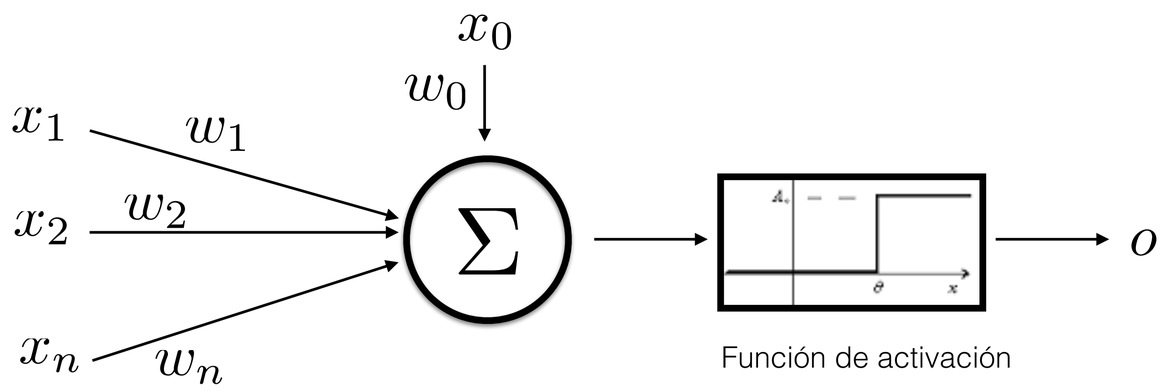
- IA beginnings a key idea was to mimic the brain with computers.
- Brain solves complex problems.
- But… brain it is complex, even a single neuron is very complex.





Classification: algorithms

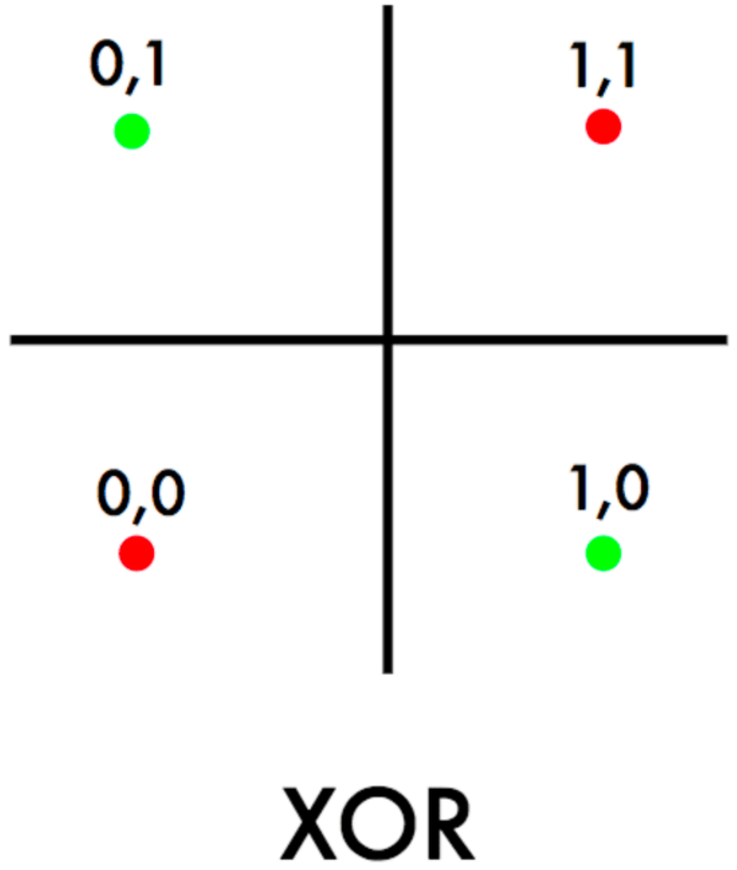
- The perceptron.
- A very simple neuron model.



$$w_i = w_i + \Delta w_i$$
$$\Delta w_i = \rho(t - o)x_i$$

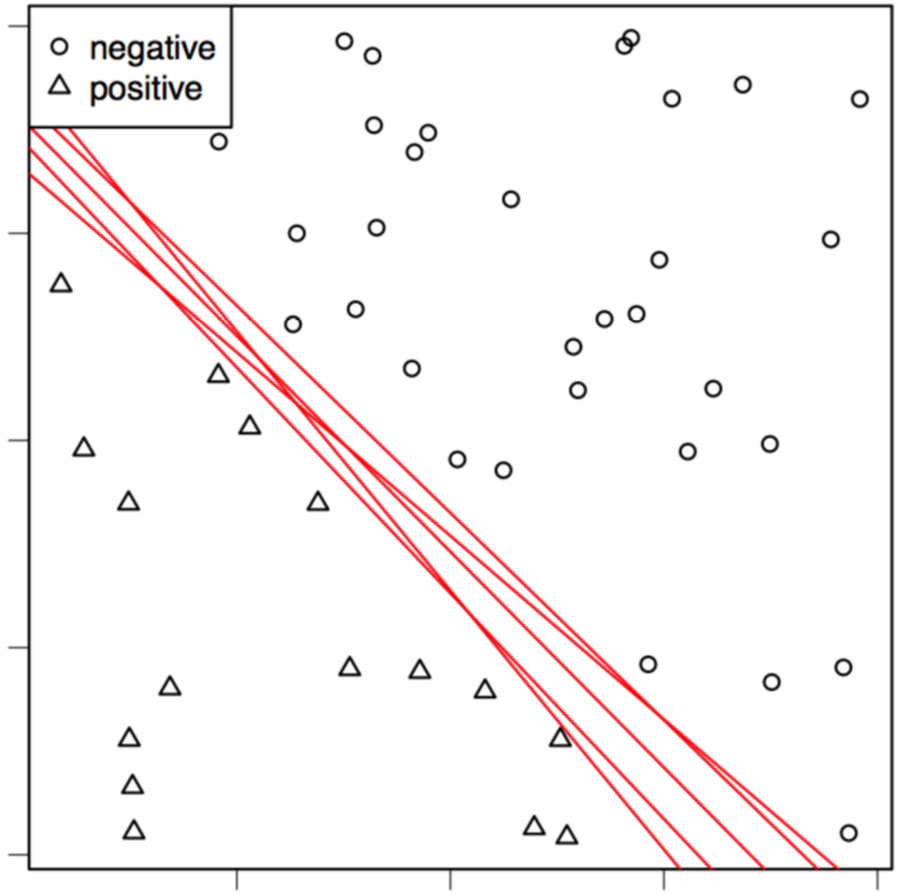
Classification: perceptron

- Perceptron can only do a linear separation.
- During 70’ this limitation made the community to lose interest in perceptrons.
- (Tomorrow you follow this story!)

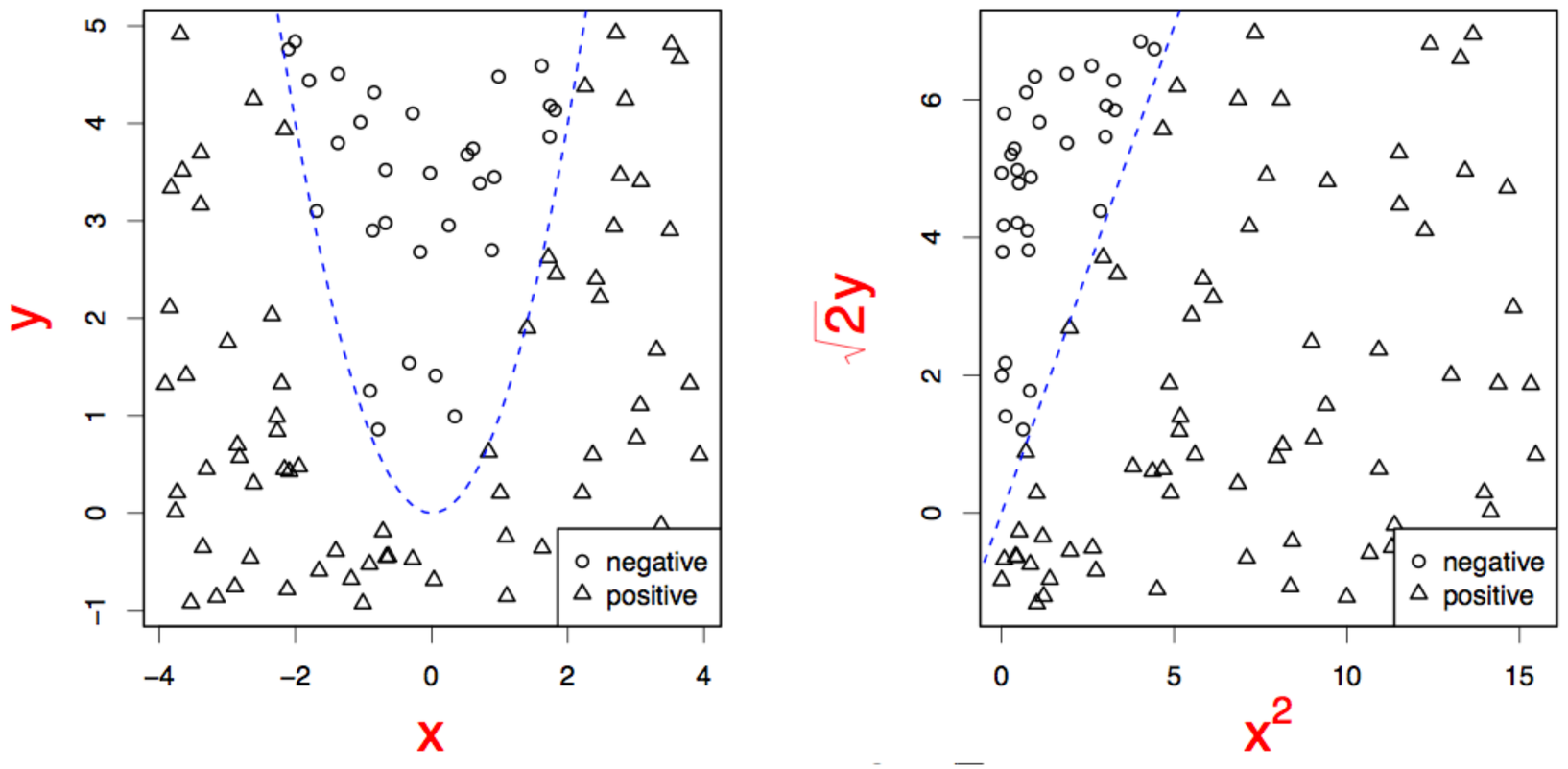


Classification: Support Vector Machine

- Perceptron will find a line/plane (any).
- We may look for the “best” line/plane, the one the maximize margin.



Classification: Support Vector Machine (SVM)



- SVM still solve only linear problems and it is designed for binary problems.
- But, we can solve other problems going to higher dimensions / transformations.

Classification: Support Vector Machine (SVM)

- Linear:

$$K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$$

- Polynomia:

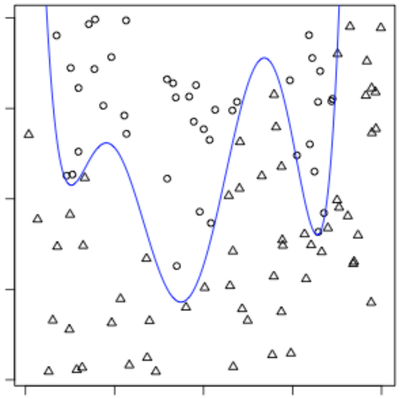
$$K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}' + c)^d$$

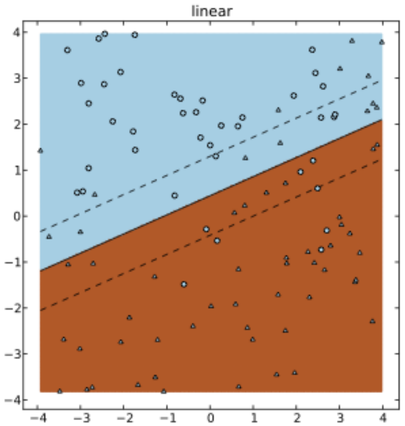
- Radial Basis Function (RBF):

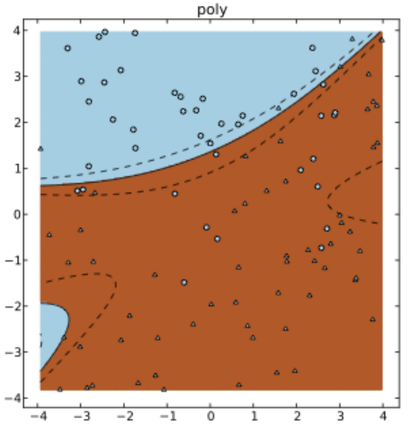
$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2\right)$$

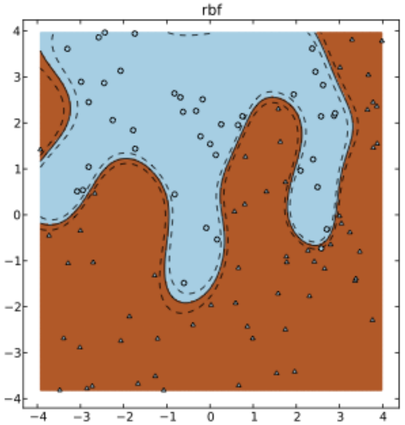
- Sigmoid:

$$K(\mathbf{x}, \mathbf{x}') = \tanh(\gamma \cdot \mathbf{x}^T \mathbf{x}' + c)$$





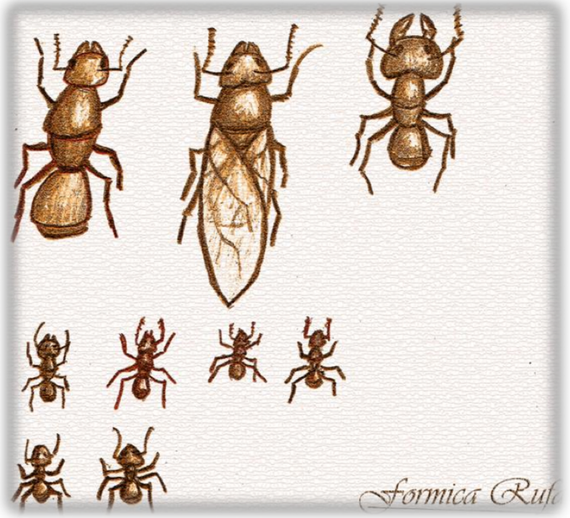


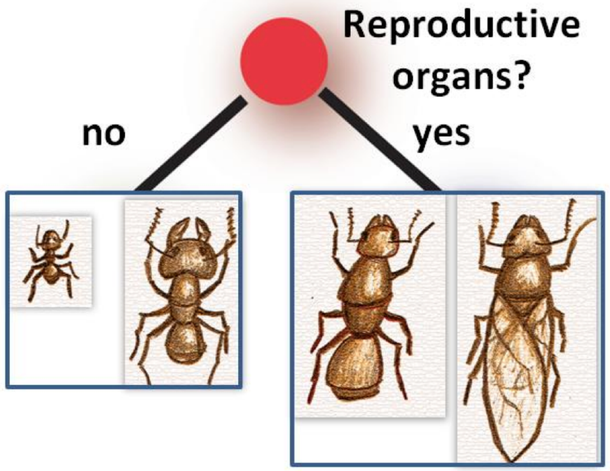


- Instead of manually proposing transformation usually explore parametrized transformations.

Classification: decision trees

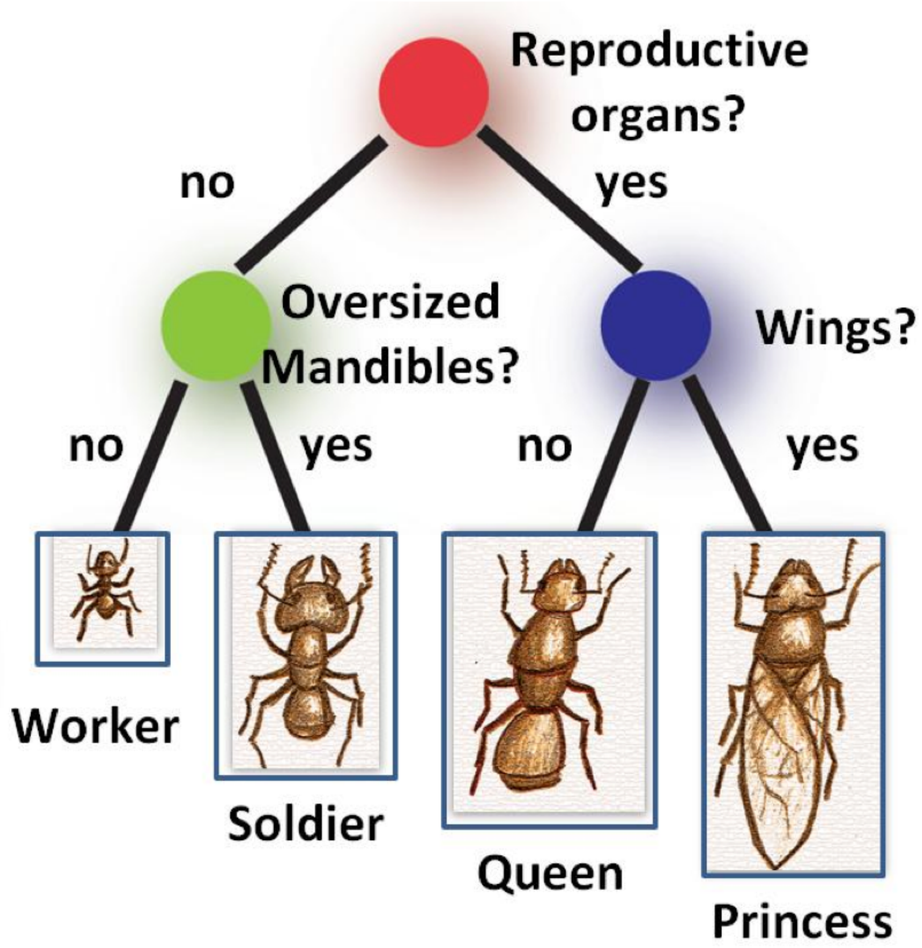
- How to make classifiers to generate “simple” rules?
- Idea: divide and conquer.





Classification: decision trees

- Idea: divide and conquer.
- (1) Divide: split observations using simple rules.
- (2) Conquer: model each subset of observations.

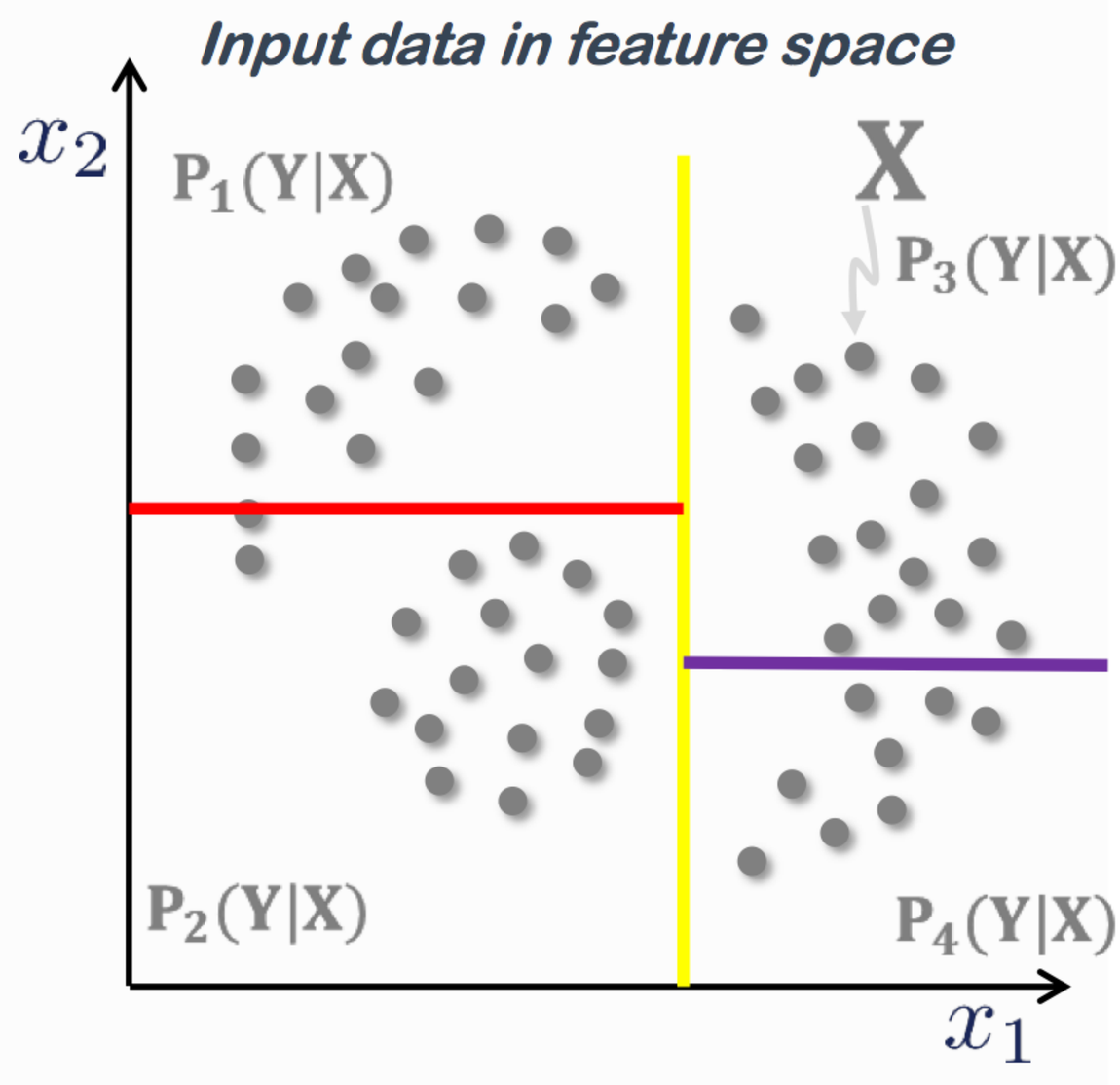


Classification: decision trees

- Objective: learn observations probability distribution P(Y|X).
- (1) Split features space.
- (2) Model probability distribution in each partition
- A tree deliver an estimation of P(Y|X) in parts.

Y=class

X=data



Classification: decision trees training

Node training

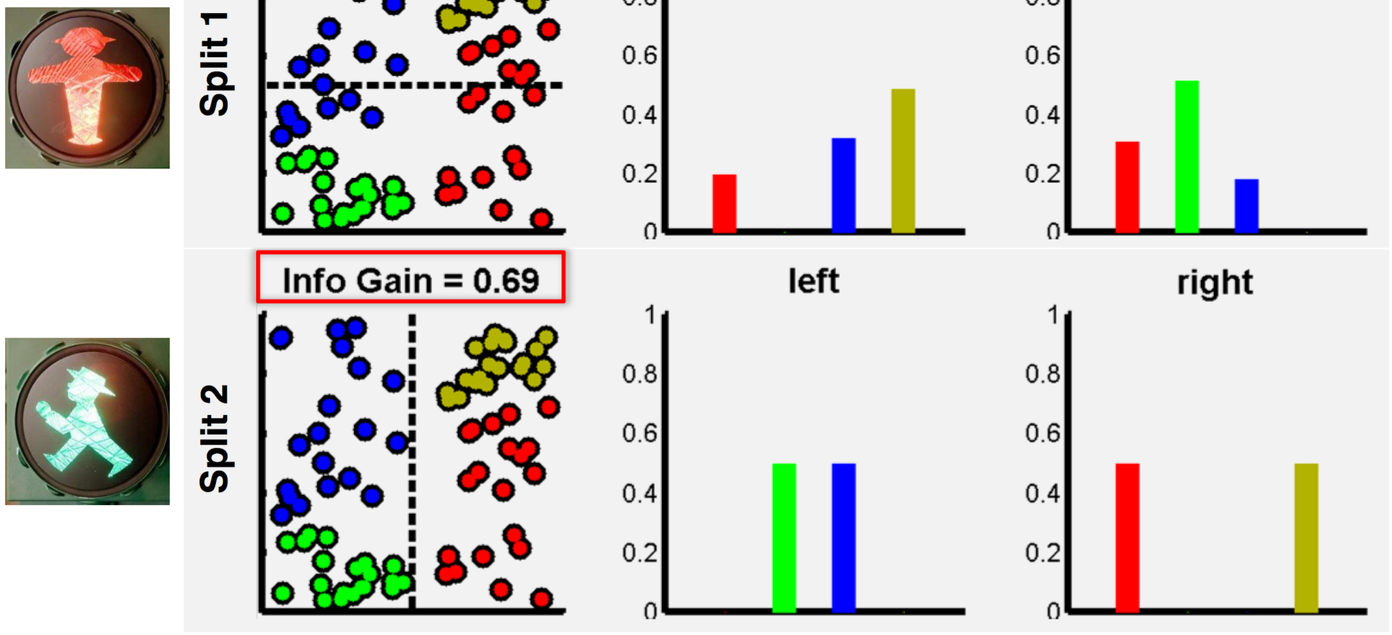
$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}\in\mathcal{T}} I$$

Information gain

$$I = H(\mathcal{S}) - \sum_{i\in\{1,2\}} \frac{|\mathcal{S}^i|}{|\mathcal{S}|} H(\mathcal{S}^i)$$

Shannon's entropy

$$H(\mathcal{S}) = -\sum_{c\in\mathcal{C}} p(c)\log(p(c))$$



Classification: trees to forest

- Idea: to replace a tree for a forest of trees.
- In a forest, trees are slightly different
- The set of uncorrelated trees improves generalization properties.
- Each tree is trained with a different dataset partition.