Language Technology

Chapter 6: Topics in Information Theory and Machine Learning

https://link.springer.com/chapter/10.1007/978-3-031-57549-5_6

Pierre Nugues

Pierre.Nugues@cs.lth.se

September 11, 2025

Why Machine Learning: Early Artificial Intelligence

Early artificial intelligence techniques used introspection to codify knowledge, often in the form of rules.
Expert systems, one of the most notable applications of traditional AI, were entirely based on the competence of experts.
This has two major drawbacks:

- Need of an expertise to understand and explain the rules
- Bias introduced by the expert

Why Machine Learning: What has Changed

Now terabytes of data available.
Makes it impossible to understand such volumes of data, organize them using manually-crafted rules.
Triggered a major move to empirical and statistical techniques.
In fact, most machine–learning techniques come from traditional statistics.
Applications in natural language processing, medicine, banking, online shopping, image recognition, etc.

> *The success of companies like Google, Facebook, Amazon, and Netflix, not to mention Wall Street firms and industries from manufacturing and retail to healthcare, is increasingly driven by better tools for extracting meaning from very large quantities of data. 'Data Scientist' is now the hottest job title in Silicon Valley.*
>
> – Tim O'Reilly

Entropy

Information theory models a text as a sequence of symbols.
Let $x_1, x_2, \ldots, x_N$ be a discrete set of N symbols representing the characters.
The information content of a symbol is defined as

$$I(x_i) = -\log_2 P(x_i) = \log_2 \frac{1}{P(x_i)},$$

Entropy, the average information content, is defined as:

$$H(X) = -\sum_{x \in X} P(x) \log_2 P(x),$$

By convention: $0 \log_2 0 = 0$.

Entropy of a Text

The entropy of the text is

$$\begin{aligned} H(X) &= -\sum_{x \in X} P(x) \log_2 P(x). \\ &= -P(A)\log_2 P(A) - P(B)\log_2 P(B) - \ldots \\ &\quad -P(Z)\log_2 P(Z) - P(\grave{A})\log_2 P(\grave{A}) - \ldots \\ &\quad -P(\ddot{Y})\log_2 P(\ddot{Y}) - P(\textit{blanks})\log_2 P(\textit{blanks}). \end{aligned}$$

Entropy of Gustave Flaubert's *Salammbô* in French is $H(X) = 4.37$.

Cross-Entropy

- The cross entropy of M on P is defined as:

$$H(P, M) = -\sum_{x \in X} P(x) \log_2 M(x).$$

- We have the inequality $H(P) \leq H(P, M)$.
- We can use cross-entropy to compare two statistical distributions: for instance the letters in a novel by Flaubert and one by Victor Hugo.
- We call first distribution the model M and we compare it with P.
- The difference:

$$D_{KL}(P||M) = H(P, M) - H(P)$$

is called the Kullback-Leibler divergence

Cross-Entropy on Texts

The model M is the letter distribution of the 14 first chapters of *Salammbô*.
We use other texts to derive P: the test distributions

	Entropy	Cross entropy	Divergence
Salammbô, chapters 1-14, training set	4.37168	4.37168	0.0
Salammbô, chapter 15, test set	4.31338	4.32544	0.01206
Notre Dame de Paris, test set	4.42285	4.44187	0.01902
Nineteen Eighty-Four, test set	4.34982	4.79617	0.44635

Code Example: Computing Entropy

Experiment: Jupyter Notebook:
https://github.com/pnugues/pnlp/tree/main/notebooks

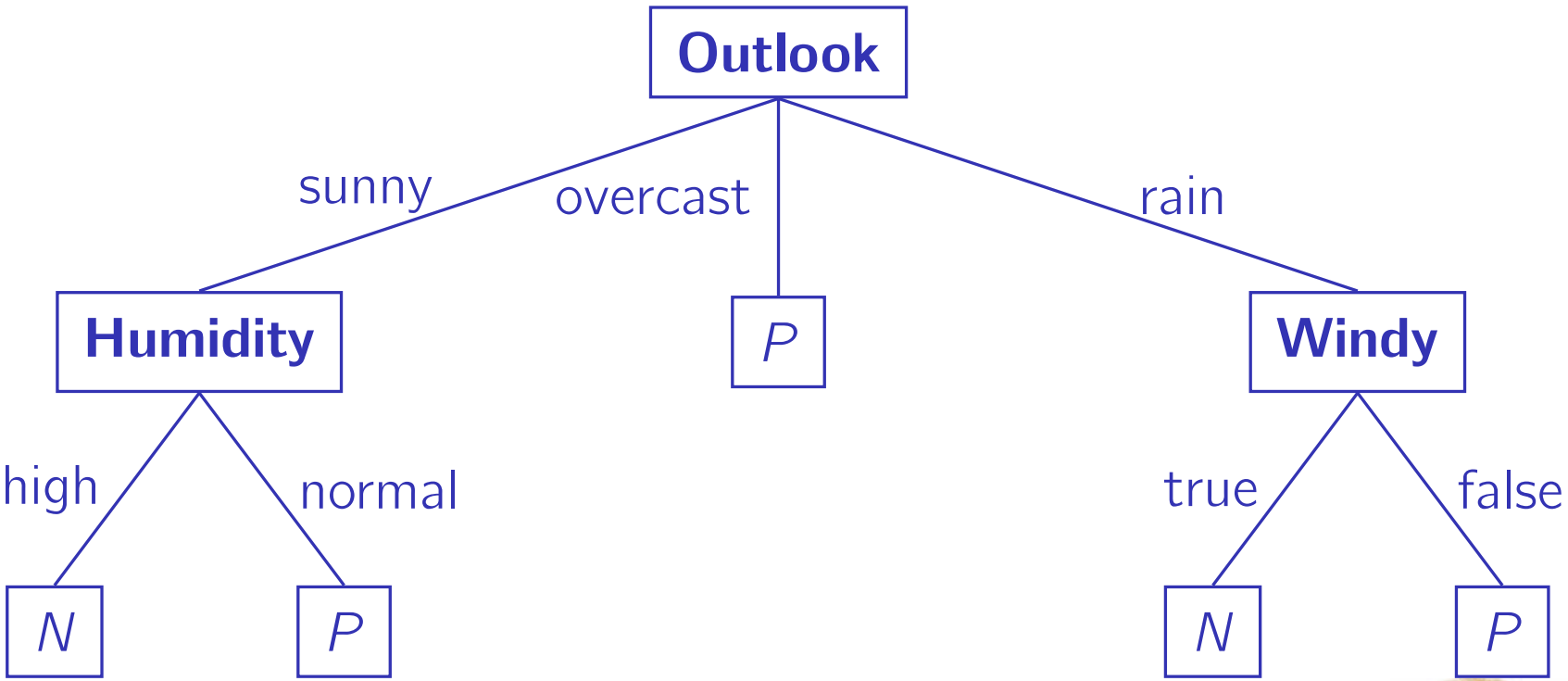
Entropy, Decision Trees, and Classification

Decision trees are useful devices to classify objects into a set of classes.
Entropy can help us learn automatically decision trees from a set of data.
The algorithm is one of the simplest machine-learning techniques to obtain a classifier.
There are many other machine-learning algorithms, which can be classified along two lines: supervised and unsupervised
Supervised algorithms need a training set.
Their performance is measured against a test set.
We can also use N-fold cross validation, where the test set is selected randomly from the training set N times, usually 10.

Objects, Classes, and Attributes. After Quinlan (1986)

Object	Attributes				Class
	Outlook	Temperature	Humidity	Windy	
1	Sunny	Hot	High	False	*N*
2	Sunny	Hot	High	True	*N*
3	Overcast	Hot	High	False	*P*
4	Rain	Mild	High	False	*P*
5	Rain	Cool	Normal	False	*P*
6	Rain	Cool	Normal	True	*N*
7	Overcast	Cool	Normal	True	*P*
8	Sunny	Mild	High	False	*N*
9	Sunny	Cool	Normal	False	*P*
10	Rain	Mild	Normal	False	*P*
11	Sunny	Mild	Normal	True	*P*
12	Overcast	Mild	High	True	*P*
13	Overcast	Hot	Normal	False	*P*
14	Rain	Mild	High	True	*N*

Classifying Objects with Decision Trees. After Quinlan (1986)



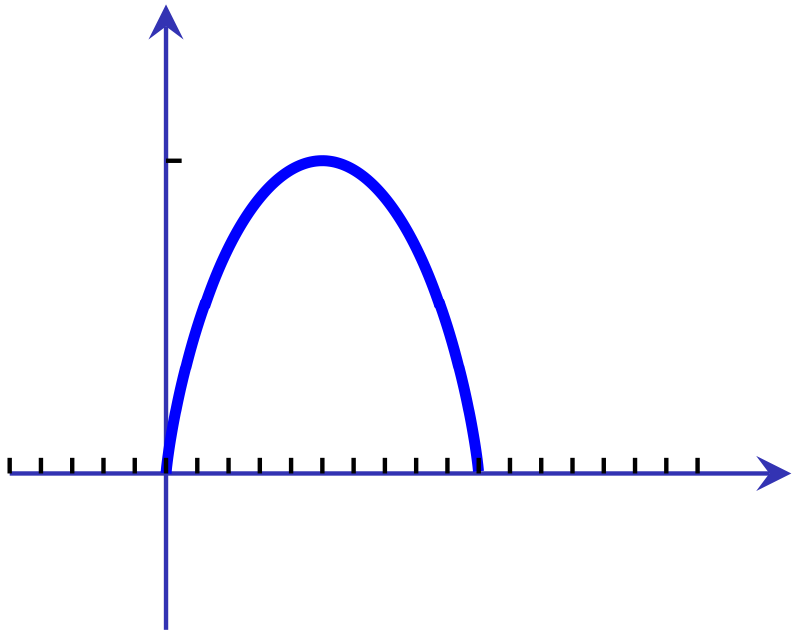
Decision Trees and Classification

Each object is defined by an attribute vector (or feature vector) $\{A_1, A_2, ..., A_v\}$
Each object belongs to a class $\{C_1, C_2, ..., C_n\}$
The attributes of the examples are:
$\{Outlook, Temperature, Humidity, Windy\}$ and the classes are: $\{N, P\}$.
The nodes of the tree are the attributes.
Each attribute has a set of possible values. The values of Outlook are $\{Sunny, Rain, Overcast\}$
The branches correspond to the values of each attribute
The optimal tree corresponds to a minimal number of tests.

The Entropy Function

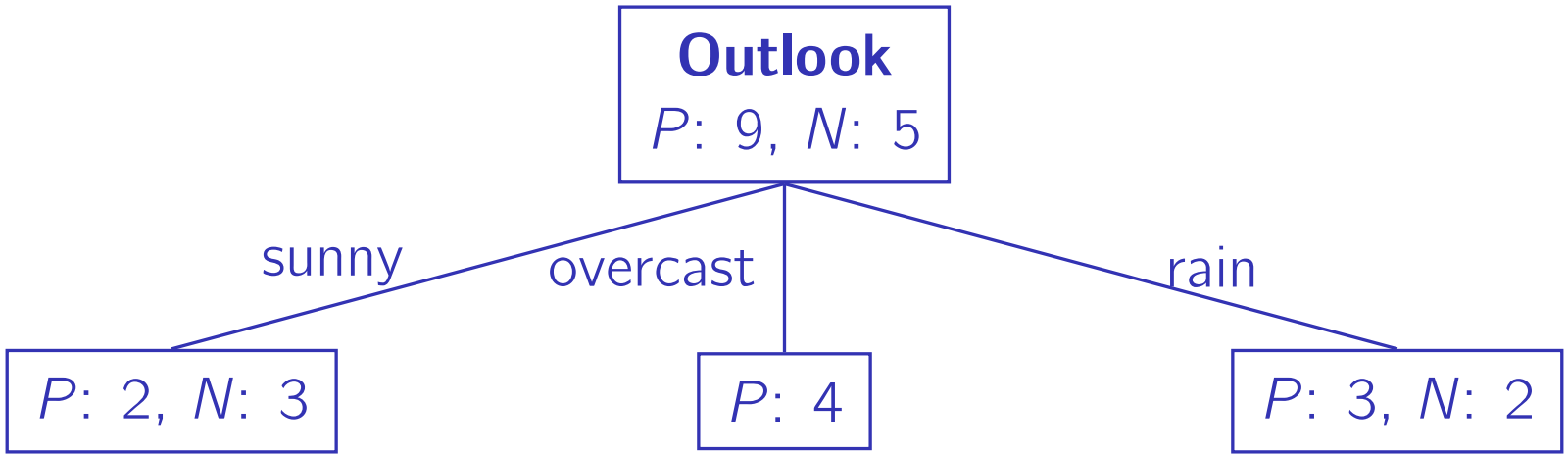
The entropy of a two-class set p and n is:

$$\begin{aligned} I(p,n) &= -P(p)\log_2 P(p) - P(n)\log_2 P(n) \\ &= -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n} \\ &= -x\log_2 x - (1-x)\log_2(1-x) \end{aligned}$$



ID3 (Quinlan, 1986)

Each attribute scatters the set into as many subsets as there are values for this attribute.



At each decision point, the "best" attribute has the maximal separation power, the maximal information gain

ID3 (Quinlan, 1986)

The weighted average of all the nodes below an attribute is:

$$\sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right).$$

The information gain is defined as $I_{before} - I_{after}$

Example

$$I_{before}(p,n) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.940.$$

Outlook has three values: sunny, overcast, and rain.

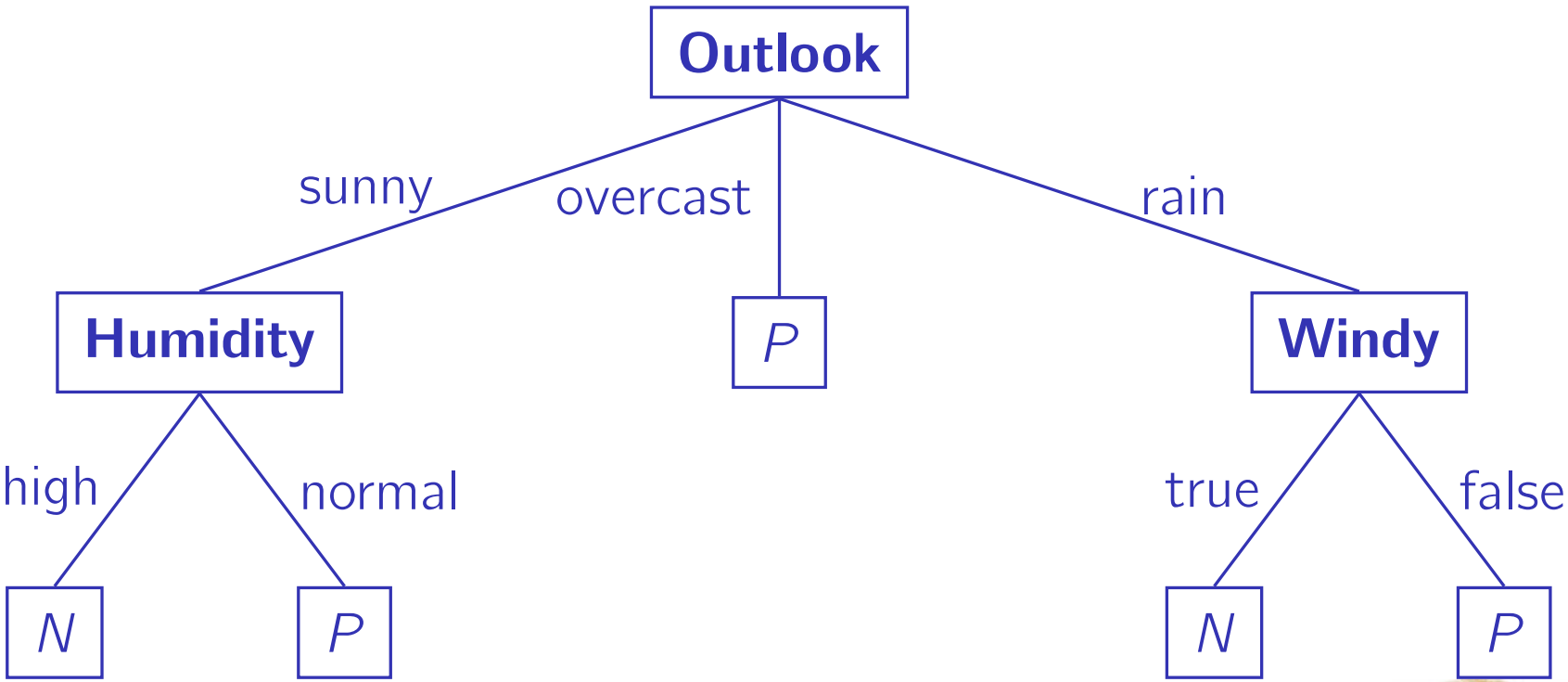
$$\begin{aligned} I(p_1,n_1) &= -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.971. \\ I(p_2,n_2) &= 0. \\ I(p_3,n_3) &= -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.971. \end{aligned}$$

The gain is $0.940 - 0.694 = 0.246$, the highest possible among the attributes

Objects, Attributes, and Classes. After Quinlan (1986)

Object	Attributes				Class
	Outlook	Temperature	Humidity	Windy	
1	Sunny	Hot	High	False	*N*
2	Sunny	Hot	High	True	*N*
3	Overcast	Hot	High	False	*P*
4	Rain	Mild	High	False	*P*
5	Rain	Cool	Normal	False	*P*
6	Rain	Cool	Normal	True	*N*
7	Overcast	Cool	Normal	True	*P*
8	Sunny	Mild	High	False	*N*
9	Sunny	Cool	Normal	False	*P*
10	Rain	Mild	Normal	False	*P*
11	Sunny	Mild	Normal	True	*P*
12	Overcast	Mild	High	True	*P*
13	Overcast	Hot	Normal	False	*P*
14	Rain	Mild	High	True	*N*

Classifying Objects with Decision Trees. After Quinlan (1986)



Matrix Notation

- A feature vector (predictors): $\mathbf{x}$, and feature matrix: X;
- The class: y and the class vector: $\mathbf{y}$;
- The predicted class (response): $\hat{y}$, and predicted class vector: $\hat{\mathbf{y}}$

$$X = \begin{bmatrix} \textit{Sunny} & \textit{Hot} & \textit{High} & \textit{False} \\ \textit{Sunny} & \textit{Hot} & \textit{High} & \textit{True} \\ \textit{Overcast} & \textit{Hot} & \textit{High} & \textit{False} \\ \textit{Rain} & \textit{Mild} & \textit{High} & \textit{False} \\ \textit{Rain} & \textit{Cool} & \textit{Normal} & \textit{False} \\ \textit{Rain} & \textit{Cool} & \textit{Normal} & \textit{True} \\ \textit{Overcast} & \textit{Cool} & \textit{Normal} & \textit{True} \\ \textit{Sunny} & \textit{Mild} & \textit{High} & \textit{False} \\ \textit{Sunny} & \textit{Cool} & \textit{Normal} & \textit{False} \\ \textit{Rain} & \textit{Mild} & \textit{Normal} & \textit{False} \\ \textit{Sunny} & \textit{Mild} & \textit{Normal} & \textit{True} \\ \textit{Overcast} & \textit{Mild} & \textit{High} & \textit{True} \\ \textit{Overcast} & \textit{Hot} & \textit{Normal} & \textit{False} \\ \textit{Rain} & \textit{Mild} & \textit{High} & \textit{True} \end{bmatrix}; \mathbf{y} = \begin{bmatrix} \text{N} \\ \text{N} \\ \text{P} \\ \text{P} \\ \text{P} \\ \text{N} \\ \text{P} \\ \text{N} \\ \text{P} \\ \text{P} \\ \text{P} \\ \text{P} \\ \text{P} \\ \text{N} \end{bmatrix}$$

Converting Symbolic Attributes into Numerical Vectors

Linear classifiers are numerical systems.

Symbolic – nominal – attributes are mapped onto vectors of binary values.

A conversion of the weather data set.

Object	Attributes										Class
	Outlook			Temperature			Humidity		Windy		
	Sunny	Overcast	Rain	Hot	Mild	Cool	High	Normal	True	False	
1	1	0	0	1	0	0	1	0	0	1	*N*
2	1	0	0	1	0	0	1	0	1	0	*N*
3	0	1	0	1	0	0	1	0	0	1	*P*
4	0	0	1	0	1	0	1	0	0	1	*P*
5	0	0	1	0	0	1	0	1	0	1	*P*
6	0	0	1	0	0	1	0	1	1	0	*N*
7	0	1	0	0	0	1	0	1	1	0	*P*
8	1	0	0	0	1	0	1	0	0	1	*N*
9	1	0	0	0	0	1	0	1	0	1	*P*
10	0	0	1	0	1	0	0	1	0	1	*P*
11	1	0	0	0	1	0	0	1	1	0	*P*
12	0	1	0	0	1	0	1	0	1	0	*P*
13	0	1	0	1	0	0	0	1	0	1	*P*
14	0	0	1	0	1	0	1	0	1	0	*N*

Code Example: Categorical Data with sklearn

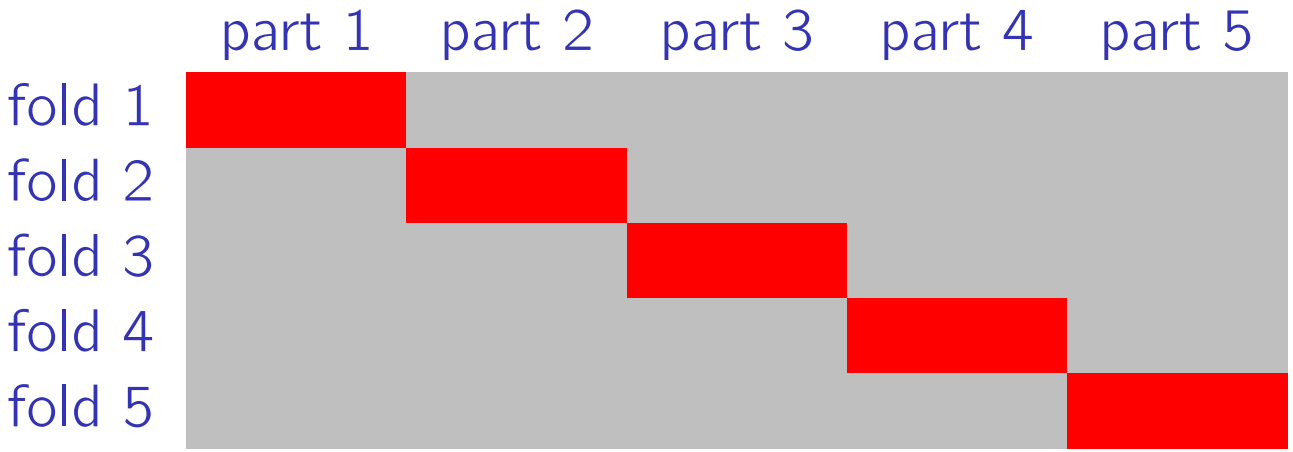
Experiment: Jupyter Notebook:
https://github.com/pnugues/pnlp/tree/main/notebooks

Evaluation

- The standard evaluation procedure is to train the classifier on a training set and evaluate the performance on a test set.
- When we have only one set, we divide it in two subsets: the training set and the test set (or holdout data).
- The split can be 90–10 or 80–20
- This often optimizes the classifier for a specific test set and creates an overfit

Cross Validation

- A N-fold cross validation mitigates the overfit
- The set is partitioned into N subsets, $N = 5$ for example, one of them being the test set (red) and the rest the training set (gray).
- The process is repeated N times with a different test set: N folds



At the extreme, leave-one-out cross-validation