

## DISCLAIMER

These notes are supposed to explain some major topics covered in the lessons.

While they may not provide an exhaustive overview, they strive to complement the information presented in the slides with some explanations taken from additional readings in various textbooks.

If someone finds some errors or have any suggestion, it's possible to contact me on [mmeschini001@gmail.com](mailto:mmeschini001@gmail.com)

# MACHINE LEARNING

Why machine learning? Many interesting problems are too complex to admit an algorithmic solution or even a complete description. For these problems only data are available. Machine learning is about using data to solve problems.

## Machine learning jargon

**MODEL:** a mathematical or computational representation of the real world

**TRAINING DATA:** the portion of the dataset used to train the machine learning model

**TEST DATA:** the portion of the dataset that is used to evaluate the performance of a trained machine learning model.

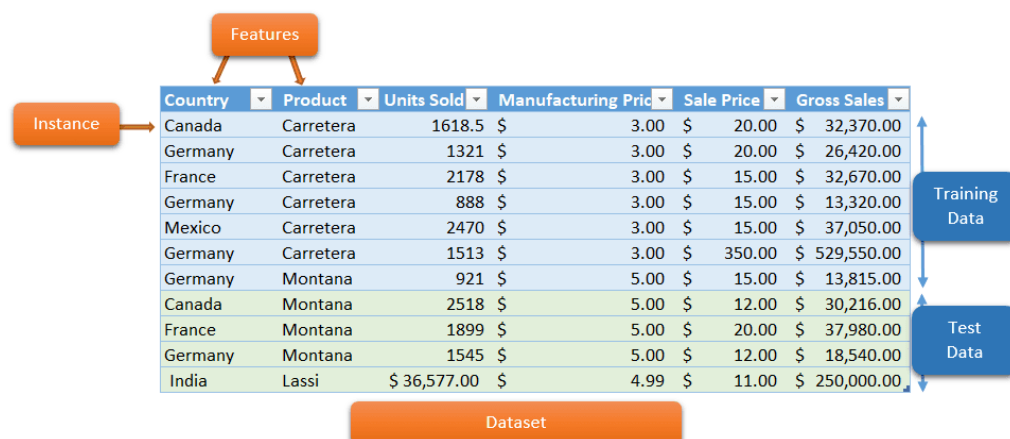
**FEATURE/ATTRIBUTE/VARIABLES:** an individual measurable property used to train the machine learning model

**OBSERVATIONS/SAMPLES/INSTANCES:** are individual data points that represent specific records. Typically they correspond to a single entry.

**CLASS/LABELS:** The output or target variable that a machine learning model is trying to predict.

**OVERFITTING:** When a machine learning model learns the training data too well, capturing noise and irrelevant details, which can lead to poor generalization to new data.

**UNDERFITTING:** The opposite of overfitting, where a model is too simple to capture the underlying patterns in the data, resulting in poor performance.



There are three different types of machines learning:

- **SUPERVISED LEARNING:** the algorithm “learns” from the training dataset by iteratively making predictions on the data and adjusting for the correct answer. While supervised learning models tend to be more accurate than unsupervised learning models, they require upfront human intervention to label the data appropriately.
- **UNSUPERVISED LEARNING/CLUSTERING:** Unsupervised learning models, in contrast, work on their own to discover the inherent structure of unlabelled data. Note that they still require some human intervention for validating output variables.

- REINFORCEMENT LEARNING algorithms work by training the agent through interaction with the environment. They learn to optimize their policy over time, improving their decision-making capabilities.

The types of quantity our model want to learn can be of two different types:

- Real values
- Categorical/nominal values: e.g. colour, name. (there isn't a natural ordering)

Depending on the model an the output variables we can distinguish:

		type of output	
		quantitative	nominal
supervised	YES	REGRESSION	CLASSIFICATION
	NO	LOW-DIMENSIONAL MAPPING	CLUSTERING

## Possible scenarios

....

## Probability

The probability quantifies the chance of an event or outcome occurring. It provides a way to express uncertainty to which something is likely to happen. Something that may or may not happen is called an **outcome** ( $\omega$ ). All possible outcomes constitute the **sample space** ( $\Omega$ ). An **event** refers to a specific outcome or a subsets of  $\Omega$ .

We defined as  $P(A)$  the probability of event  $A$ .

### Axioms of probability:

- ①  $P(A) \geq 0$
- ②  $\sum_{i=1}^N P(\omega_i) = 1$ , or  $P(\Omega) = 1$
- ③ If  $A_1$  and  $A_2$  are mutually exclusive events  
(viewed as sets: if they are **disjoint** = have zero intersection),  
then  $P(A_1 \text{ or } A_2) = P(A_1 \cup A_2) = P(A_1) + P(A_2)$

A **random variable** ( $X$ ) is a numerical variable that doesn't have a fixed value but changes according to a given probability law. The **Cumulative distribution function (CDF)** of a random variable  $X$  is the function given by:

$$F_X(x) = P(X \leq x) \quad (\text{Eq.1})$$

Where the right-hand side represents the probability that the random variable  $X$  takes on a value less than or equal to  $x$ .

The **probability mass function** is a function that gives the probability that a **discrete** random variable is exactly equal to some value.

$$p_X(x) = P(X = x)$$

The **probability density function** is a function whose value at any given sample in the sample space can be interpreted as providing a relative likelihood that the value of the random variable would be equal to that sample.

A function  $f_X$  such that

$$P_X(\hat{x}) = \int_{-\infty}^{\hat{x}} f_X(x) dx$$

is the **probability density function** of  $X$ .

We define the **conditional probability**  $P(E | F)$  as the probability of an event  $E$  given the knowledge that another event  $F$  has occurred.

Two events are **independent** if the outcome of one event does not influence the outcome of the second event.

## Bayesian decision theory

Bayesian decision theory is a fundamental statistical approach to the problem of pattern classification. This approach is based on quantifying the trade-offs between various classification decisions using probability and the costs that accompany such decisions. It makes the assumption that the decision problem is posed in probabilistic terms, and that **all the relevant probability values are known**.

We define as **state of nature ( $\omega$ )** the different possible states or scenarios that a system could be in. (e.g. salmon and sea bass are two different state of nature). We define a **prior probability** refers to the probability of an event occurring based on prior knowledge. It is represented by the likelihood in the occurrence of an event before any new evidence is taken into account. Then, is important to define the **class-conditional probability density function** expressed as  $P(x|\omega)$  that represents the distribution of the random variable  $x$  depending on the state of nature.

The probability density of finding a pattern that is in category  $\omega_j$  and has a feature value  $x$  can be written in two ways:  $P(\omega_j, x) = P(\omega_j|x) * P(x) = P(x|\omega_j) * P(\omega_j)$  rearranging these leads us to the **Bayes' formula**:

$$\boxed{P(\omega_j|x) = \frac{p(x|\omega_j)P(\omega_j)}{p(x)},} \quad \text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}.$$

Bayes' formula shows that by observing the value of  $x$  we can convert the prior probability  $P(\omega_j)$  to the **a posteriori probability**  $P(\omega_j|x)$ , that is the probability of the state of nature being  $\omega_j$  given that the feature value  $x$  has been measured.

*Oss: the evidence can be viewed as a scale factor that guarantees that the posterior probability sum to one.*

## RISK ...

## Naive Bayes classifier

A Naive Bayes classifier is a simple and probabilistic machine learning algorithm that is often used for classification tasks. It is based on Bayes' theorem and makes a "naïve" assumption that features used to describe data are conditionally independent.

Naïve Bayes is a conditional probability model: it assigns probabilities  $P(C_k | x_1, \dots, x_n)$  for each of the  $K$  possible classes  $C_k$  (= state of natures) given a problem instance to be classified represented by a vector  $\mathbf{x} = (x_1, \dots, x_n)$  encoding some  $n$  features. From the Bayes' theorem

$$p(C_k | \mathbf{x}) = \frac{p(C_k) p(\mathbf{x} | C_k)}{p(\mathbf{x})}$$

In practice, there is interest only in the numerator of that fraction, because the denominator does not depend on  $C$  and the values of the features  $x_i$  are given, so that the denominator is effectively constant. The numerator is equivalent to the [joint probability](#) model

$$p(C_k, x_1, \dots, x_n)$$

which can be rewritten as follows, using the [chain rule](#) for repeated applications of the definition of [conditional probability](#):

$$\begin{aligned} p(C_k, x_1, \dots, x_n) &= p(x_1, \dots, x_n, C_k) \\ &= p(x_1 | x_2, \dots, x_n, C_k) p(x_2, \dots, x_n, C_k) \\ &= p(x_1 | x_2, \dots, x_n, C_k) p(x_2 | x_3, \dots, x_n, C_k) p(x_3, \dots, x_n, C_k) \\ &= \dots \\ &= p(x_1 | x_2, \dots, x_n, C_k) p(x_2 | x_3, \dots, x_n, C_k) \dots p(x_{n-1} | x_n, C_k) p(x_n | C_k) p(C_k) \end{aligned}$$

Now the "naive" [conditional independence](#) assumptions come into play: assume that all features in  $\mathbf{x}$  are [mutually independent](#), conditional on the category  $C_k$ . Under this assumption,

$$p(x_i | x_{i+1}, \dots, x_n, C_k) = p(x_i | C_k).$$

Thus, the joint model can be expressed as

$$\begin{aligned} p(C_k | x_1, \dots, x_n) &\propto p(C_k, x_1, \dots, x_n) \\ &= p(C_k) p(x_1 | C_k) p(x_2 | C_k) p(x_3 | C_k) \dots \\ &= p(C_k) \prod_{i=1}^n p(x_i | C_k), \end{aligned}$$

## Linear Regression