# Introduction

*Albert Mata*

*12/11/2018*

# Contents

# Introducing Machine Learning

The goal of today's machine learning is to assist us in making sense of the world's massive data stores. The field of machine learning provides a set of algorithms that transform data into actionable knowledge.

**Machine Learning = Available Data + Statistical Methods + Computing Power**

Machine learning focuses on teaching computers how to use data to solve a problem. Data mining focuses on teaching computers to identify patterns that humans then use to solve a problem. Virtually all data mining involves the use of machine learning, but not all machine learning involves data mining. *The phrase "data mining" is also sometimes used as a pejorative to describe the deceptive practice of cherry-picking data to support a theory.*

Machines are not good at asking questions, or even knowing what questions to ask. They are much better at answering them, provided the question is stated in a way the computer can comprehend. Machine learning is most successful when it augments rather than replaces the specialized knowledge of a subject-matter expert. Regardless of the task, an algorithm takes data and identifies patterns that form the basis for further action. Machine learning has very little flexibility to extrapolate outside of the strict parameters it learned and knows no common sense. So, one should be extremely careful to recognize exactly what the algorithm has learned before setting it loose in the real-world settings. Also, let's keep in mind that machine learning is only as good as the data it learns from.

---

The basic learning process can be divided into four interrelated components:

- **Data storage**. All learning must begin with data. But the ability to store and retrieve data alone is not sufficient for learning. Without a higher level of understanding, knowledge is limited exclusively to recall, meaning exclusively what is seen before and nothing else.
- **Abstraction**. Stored data must be translated into broader representations and concepts. During a machine's process of knowledge representation, the computer summarizes stored raw data using a model, an explicit description of the patterns within the data. The choice of model is typically not left up to the machine. Instead, the learning task and data on hand inform model selection. The process of fitting a model to a dataset is known as training. When the model has been trained, the data is transformed into an abstract form that summarizes the original information. Imposing an assumed structure on the underlying data gives insight into the unseen by supposing a concept about how data elements are related (*gravity*).
- **Generalization**. This describes the process of turning abstracted knowledge into a form that can be utilized for future action, on tasks that are similar, but not identical, to those it has seen before. Generalization involves the reduction of an hypothetical set containing every possible theory that could be established from the data into a manageable number of important findings. The algorithm employs heuristics (educated guesses about where to find the most useful inferences,

*gut instinct*) to limit discovered patterns to those that will be most relevant to its future tasks. The heuristics employed by machine learning algorithms sometimes result in erroneous conclusions. The algorithm is said to have a bias if the conclusions are systematically erroneous, or wrong in a predictable manner. Paradoxically, bias is what blinds us from some information while also allowing us to utilize other information for action. It is how machine learning algorithms choose among the countless ways to understand a set of data.

- **Evaluation**. This measures the learner's success in spite of its biases and uses this information to inform additional training if needed. After a model has been trained on an initial training dataset, it is evaluated on a new test dataset in order to judge how well its characterization of the training data generalizes to new, unseen data. Models fail to perfectly generalize partly due to the problem of noise (measurement error, people boycotting surveys, missing data, etc). Trying to model noise is the basis of a problem called overfitting. Because most noisy data is unexplainable by definition, attempting to explain the noise will result in erroneous conclusions that do not generalize well to new cases. A model that seems to perform well during training, but does poorly during evaluation, is said to be overfitted to the training dataset, as it does not generalize well to the test dataset.

---

Any machine learning algorithm follows these 5 steps to apply the learning process to real-world tasks:

1. **Data collection**. Usually, the data will need to be combined into a single source like a text file, spreadsheet, or database.
2. **Data exploration and preparation**. Fixing or cleaning so-called "messy" data, eliminating unnecessary data, and recoding the data to conform to the learner's expected inputs.
3. **Model training**. The specific machine learning task chosen will inform the selection of an appropriate algorithm, and this algorithm will represent the data in the form of a model.
4. **Model evaluation**. Important as each machine learning model results in a biased solution to the learning problem.
5. **Model improvement**. If necessary, every previous step will need to be revised and improved.

---

Machine learning algorithms are divided into categories according to their purpose:

*Els que estudiem a l'assignatura són tots d'aprenentatge supervisat.*

- A **predictive model** is used for tasks that involve the prediction of one value (the target feature) using other values (features) in the dataset. It can be used to predict past, real time or future events. The process of training a predictive model is known as **supervised learning**, as the model is given clear instruction on what it needs to learn and how it is intended to learn it. Given a set of data, a supervised learning algorithm attempts to optimize a function (the model) to find the combination of feature values that result in the target output.
  - The often used supervised machine learning task of predicting which category an example belongs to is known as **classification**. In classification, the target feature to be predicted is a

categorical feature known as the *class*, and is divided into categories called *levels* (ordinal or not).
  - A common form of **numeric prediction** fits linear regression models to the input data in order to predict numeric data. Regression models are not the only type of numeric models, but they are, by far, the most widely used. They quantify in exact terms the association between inputs and the target, including both the magnitude and uncertainty of the relationship.
- A **descriptive model** is used for tasks that would benefit from the insight gained from summarizing data in new and interesting ways. Because there is no target to learn, the process of training a descriptive model is called **unsupervised learning**.
  - Common task called **pattern discovery** is used to identify useful associations within data (like items that are frequently purchased together).
  - Dividing a dataset into homogeneous groups is called **clustering**. This is sometimes used for segmentation analysis that identifies groups of individuals with similar behavior or demographic information, so that advertising campaigns could be tailored for particular audiences.

There are various reasons to choose one algorithm. For instance, within classification problems decision trees result in models that are readily understood, while the models of neural networks are notoriously difficult to interpret. If you were designing a credit-scoring model, this could be an important distinction because law often requires that the applicant must be notified about the reasons he or she was rejected for the loan. Even if the neural network is better at predicting loan defaults, if its predictions cannot be explained, then it is useless for this application.

## Managing and Understanding Data

Any learning algorithm is only as good as its input data, and in many cases, the input data is complex, messy, and spread across multiple sources and formats. Because of this complexity, often the largest portion of effort invested in machine learning projects is spent on data preparation and exploration.

### Basic data structures in R

- **Vectors**. Ordered set of *same type* elements.
- **Factors**. Special case of vector that is solely used to represent *categorical or ordinal* variables. Factors can be ordered or unordered. Machine learning algorithms capable of modeling ordinal data will expect *ordered* factors.
- **Lists**. Ordered set of *not necessarily same type* elements. Often used to store various types of input and output data and sets of configuration parameters for machine learning models. The result of using vector-style operators on a list object is another list object, which is a subset of the original list. To return a single list item in its native data type, we use double brackets (`[[` and `]]`) or `$` notation.

- **Data frames**. List of vectors or factors, each having exactly the same number of values, analogous to a spreadsheet or database as it has both rows and columns of data. In machine learning terms, data frame's columns are the features or attributes and rows are the examples.
- **Matrices**. A matrix is a data structure that represents a two-dimensional table with rows and columns of (typically numeric) data. R's default method for loading matrices is *column-major order* (can be overriden with `byrow = TRUE`).

## Useful functions to get data in and out of R

- `ls()`: Returns a vector with the names of all the data structures currently in memory.
- `rm()`: Removes one or more data structures from memory. `rm(list=ls())` clears the entire R session.
- `save()`: Writes one or more data structures to a file that can be reloaded later or transferred to another system. R data files have an `.RData` extension.
- `load()`: Recreates any data structures that have been saved to an `.RData` file (even overwriting existing data structures with same name).
- `read.csv()` and `write.csv()`: Imports/saves data from/to CSV files.

## Methods for understanding data

The better you understand your data, the better you will be able to match a machine learning model to your learning problem. Let's see some basic steps to achieve that.

**• Explore its structure**

How is the dataset organized? Is there a data dictionary maybe? Our data may or may not have meaningful variable names, so it may be necessary to do additional sleuthing to determine what a feature actually represents. Even when feature names are given, it is always prudent to be skeptical about those labels and investigate further.

**• Explore numeric variables**

- Measure the **central tendency** (mean and median). The median is useful as the mean is highly sensitive to outliers, or values that are atypically high or low in relation to the majority of data.
- Measure **spread** (quartiles and the five-number summary). How tightly or loosely are the values spaced? Knowing about the spread provides a sense of the data's highs and lows and whether most values are like or unlike the mean and median. Quartiles are a special case of a type of statistics called quantiles, which are numbers that divide data into equally sized quantities. The difference between Q1 and Q3 is known as the Interquartile Range (IQR) and in itself is a simple measure of spread.

- Visualize using **boxplots**. Also known as a box-and-whiskers plot. A widely used convention only allows the whiskers to extend to a minimum or maximum of 1.5 times the IQR below Q1 or above Q3. Any values that fall beyond this threshold are considered outliers and are denoted as circles or dots.
- Visualize using **histograms**. An histogram uses any number of bins of an identical width and allows them to contain different number of values (while a boxplot requires that each of the four portions of data must contain the same number of values, and widens or narrows the bins as needed). Quickly diagonosing skew patterns in our data is one of the strengths of histograms as a data exploration tool.
- Understand numeric data (uniform and normal **distributions**). Many real-world phenomena generate data that can be described by the normal distribution.
- Measure **spread** (variance and standard deviation). The variance is defined as the average of the squared differences between each value and the mean value, while the standard deviation is just the square root of the variance. Larger values for variance indicate that the data are spread more widely around the mean. The standard deviation indicates, on average, how much each value differs from the mean. *Next formulae use the population variance (which divides by n). R uses the sample variance (which divides by n - 1). Except for very small datasets, the distinction is minor.*

$$Var(X) = \sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

$$StdDev(X) = \sigma$$

- Consider **68-95-99.7 rule**: 68% of the values in a normal distribution fall within one standard deviation of the mean. 95% and 99.7% of the values fall within two and three standard deviations, respectively.

**• Explore categorical variables**

Categorical data is typically examined using tables rather than summary statistics. A table that presents a single categorical variable is known as a one-way table.

- Measure the **central tendency** (mode). The mode of a feature is the value occurring most often. Variables can be unimodal or multimodal (i.e. bimodal). The mode is used in a qualitative sense to gain an understanding of important values, keeping in mind that the most common value is not necessarily a majority. It is best to think about modes in relation to the other categories: is there one category that dominates all the others or are there several? From here, we may ask what the most common values tell us about the variable being measured. It can also be helpful to consider mode while exploring numeric data (thinking about modes as the highest bars on a histogram), particularly to examine whether or not the data is multimodal.

**• Explore relationships between variables**

Some type of questions can only be addressed by looking at bivariate relationships, which consider the relationship between two variables. Relationships of more than two variables are called multivariate relationships.

- Visualize relationships using **scatterplots**. A scatterplot is a diagram that visualizes a bivariate relationship. Patterns in the placement of dots reveal the underlying associations between the two features. Convention dictates that the *y* variable is the one that is presumed to depend on the other (and is therefore known as the dependent variable).
  - Dots in a line sloping downward → negative association.
  - Dots in a line sloping upward → positive association.
  - Dots in a flat line or randomly → no association at all.
  - The strength of a linear association between two variables is measured by a statistic known as **correlation**.
  - Keep in mind that not all associations form straight lines, as two variables may be related in a non linear way.
- Examine relationships using **two-way cross-tabulations**. Also known as a **crosstab** or **contingency table**, it's useful to examine a relationship between two nominal variables. It is a table in which the rows are the levels of one variable and the columns are the levels of another. Counts in each of the table's cells indicate the number of values falling into the particular row and column combination. Pearson's Chi-squared test for independence between two variables measures how likely it is that the difference in the cell counts in the table is due to chance alone. If the probability is very low, it provides strong evidence that the two variables are associated.

---

Useful functions for all previous steps: `str()`, `summary()`, `mean()`, `median()`, `min()`, `max()`, `range()`, `diff()`, `IQR()`, `quantile()`, `seq()`, `boxplot()`, `hist()`, `var()`, `sd()`, `table()`, `prop.table()`, `round()`, `plot()` and `CrossTable()` from `gmodels` package.

# References

Lantz, Brett. 2015. *Machine Learning with R*. Packt Publishing Ltd.