A Data Science Overview and approach for Noobs

Fabio Pontecchiani, May 2024

Machine Learning

“A computer program is said to learn from experience **E** with respect to some task **T** and some performance measure **P**, if its performance on **T**, as measured by **P**, improves with experience **E**.” (*Tom Mitchell 1998: Well-posed Learning Problem*)

Machine learning makes use of historic data in order to develop predictive models that could potentially interpret new sets of data to give important insights to new sets of data.

A data scientist has to define the problem, design the data set (transforming what is available for example in order to be used), thus prepare the data, decide on the type of data analysis to apply, and evaluate and interpret the results of the data analysis.
The computer’s role in this task is the ability to process the data and search for patterns in the data.
Machine learning is then the field of study that develops the algorithms that the computer follows in order to identify and extract patterns from the data.[1]

The scope of the ML algorithm is then to identify useful patterns in the data. Depending on the type of data and potential patterns, it’s possible to “fit” a different number of algorithms, decision trees, regression models and neural networks.

An example can be, applying a ML algorithm to a data set of stroke patients, to identify the factors that have a strong association with stroke. In another case, a model is used to label or classify new examples, for instance, a spam-filter model will have to label new emails as spam or not spam.

Supervised vs Unsupervised learning

Most of the machine learning algorithms can be classified as supervised or unsupervised learning.

Where does the difference reside?
In **supervised learning**, the data are already labelled with the value of the target. The challenge though is to have a data set already labelled, so a great part of time and effort is required to label the data with the target values, before a model can be trained with supervised learning. The spam/not spam filter is an example.
Another way to understand supervised learning is by saying that each training example has a ground truth label. The model learns a decision boundary and it replicates the labelling on the new data.

Other examples are: cryptocoins on a particular future date, predicting the gender of a person by their handwriting style, bank loan applications, predicting the number of copies a music album will sell the next month.

In **Unsupervised learning**, the training examples do not have ground truth labels. Therefore the model identifies structures such as clusters. New data can then be assigned to clusters. The purpose is to explore the data and to find some structures within. An unsupervised ML algorithm will use unlabelled data without having any predefined dataset for its training.
It is mainly used for clustering: detecting a certain logic in a group.
Examples are: identifying customers with similar attributes, segmenting text topics, recommending items on similar purchases.

Other types of ML algorithms are **semi-supervised learning** and **reinforcement learning**.

Semisupervised uses both labelled and unlabelled data, this last group representing the majority of the data, as it's less expensive and requires less time to acquire.
Semi supervised learning is useful when labelling costs are too high to allow a fully labelling training process.
An example is identifying a person's face on a webcam.

Reinforcement learning is instead based on a trial and error approach. This type of learning is often used in robotics, gaming and navigation, such as self-driving cars.

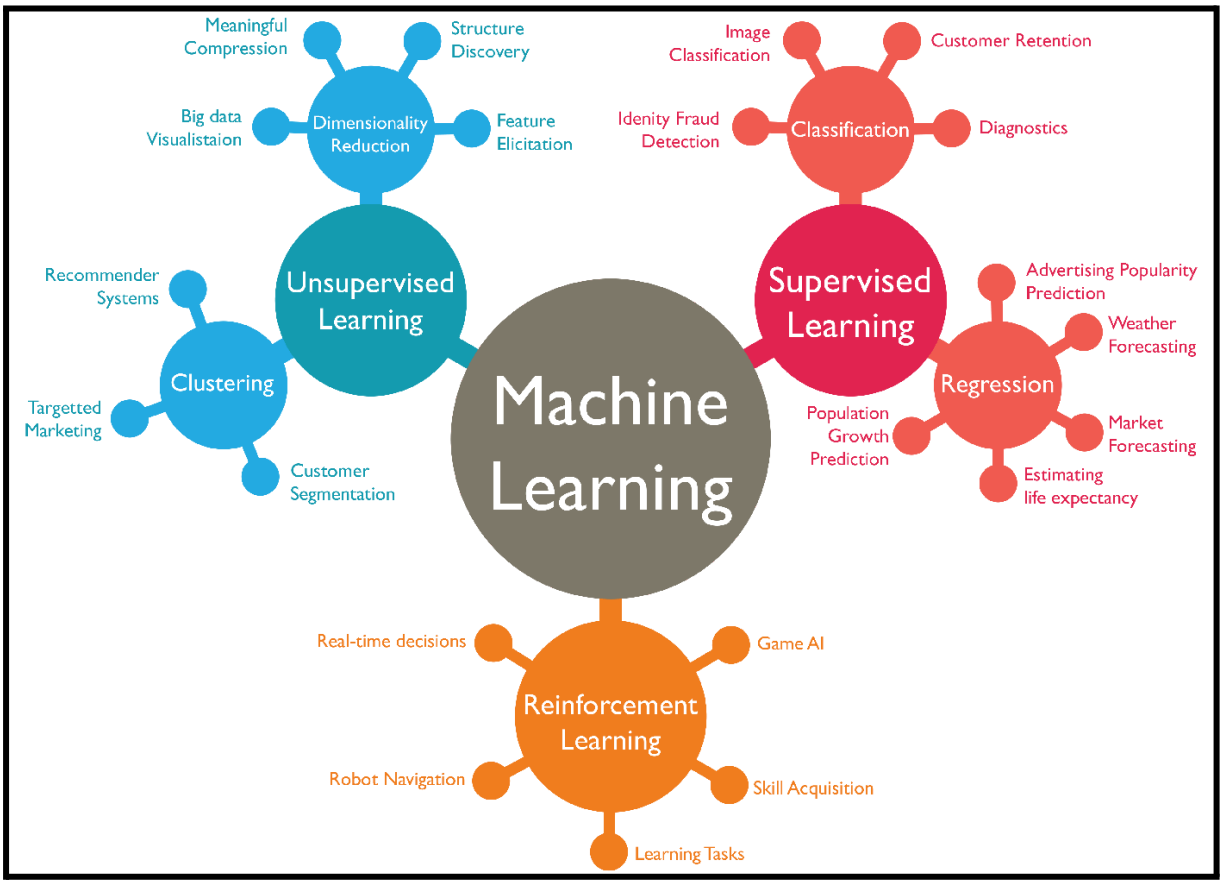

Fig.1 Classification of Machine Learning algorithms.[2]

Machine Learning Roadmap

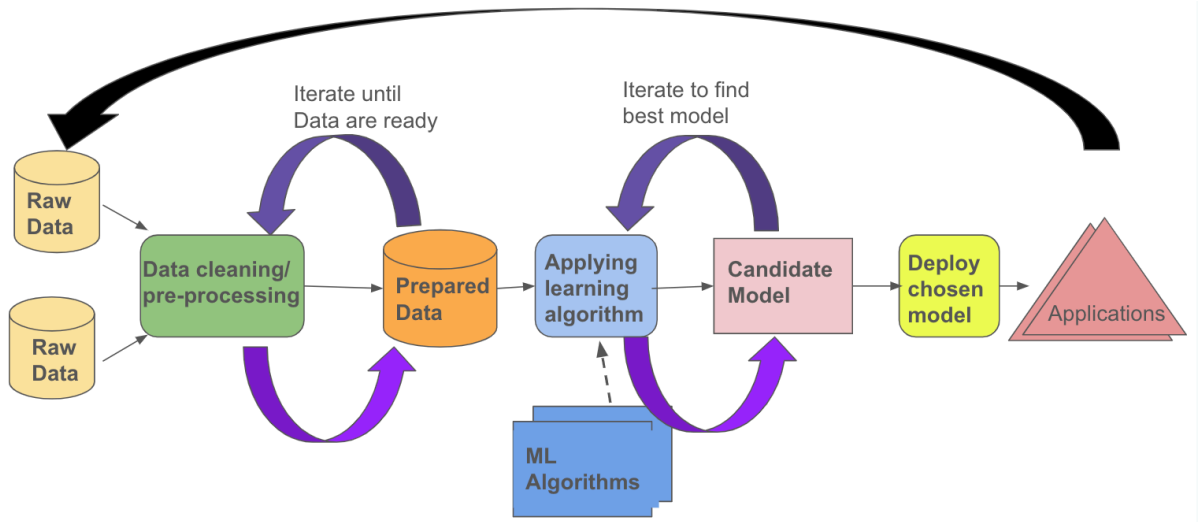

Fig.2 Machine learning roadmap scheme.

The data that have to be analysed/used for a machine learning model, are usually not ready for use, but need to be treated, what a data scientist would say is ‘data have to be cleaned’. This is because data are usually incomplete: missing values, invalid: negative values, inaccurate, or maybe non numerical: for example indicating genders F or M.

EDA (Exploratory Data Analysis)

EDA is the first step to have a global view and understanding of the data, in order to summarise the main characteristics, often with visual methods. What can the data tell us before a formal modelling step?

Major tasks to be performed during EDA:

- Data cleaning
- Relationships between features
- Relationships between each feature and target
- Descriptive statistics (mean, median, min, max, histograms, boxplots…)

The treatment is different depending if the data are categorical or numerical.

EDA techniques for categorical data are organising the features in a range of values. They can be a graphical (histograms or box plots) or non graphical method (tabulation). Numerical data are treated by calculating statistical features sum, median, mean, min and max, distribution through standard deviation, variance, quartiles and interquartiles ranges.

Skew and Kurtosis are two statistical methods used to calculate data distribution, while the first measures data symmetry, the second one measures the "peakedness" or the "tailedness" of a probability distribution relative to a normal distribution.

Missing Values treatment

Missing values can arise due to various reasons such as data collection errors, data corruption, or incomplete data recording. Handling missing values appropriately ensures that the analysis or model building process is not biased or inaccurate.

Identify missing values, missing values are often represented in Pandas as (none) or NaN (not a number).
Understanding why the values are missing. Data collection errors, data corruption, or some other reason? This understanding can help determine the appropriate treatment strategy.
If the missing values are relatively few and randomly distributed, you can simply remove the rows or columns containing missing values using the 'dropna( )' method.
Replacing missing values with estimated or calculated values (imputation). Common imputation techniques include:

- Mean, median, or mode imputation: Replace missing values with the mean, median, or mode of the column.
- Forward or backward fill: Replace missing values with the previous or next non-missing value in the column.
- Interpolation: Use interpolation techniques to estimate missing values based on existing data points.
- Regression imputation: Predict missing values using regression models trained on non-missing values.

Outliers treatment

a) Deletion, if they are due to error, *e.g.* observations outliers.
b) Treat separately: developing two different models, one for outliers and one for non-outliers observations.
c) Use log, square root, binning or any other transformation method to reduce the effects of outliers.
d) Using a PCA algorithm to reduce dimension, but this might bring to lose important information.

The treatment of outliers needs to be based on the ML algo that has to be used, for example:

- Regression models and KNN are very sensitive to outliers
- Decision trees are not too sensitive and the outliers treatment can be skipped if using a tree algorithm.

Multifeature analysis

Covariance and correlation describe how two variables are related. Two variables can be positively or inversely correlated. The correlation coefficient is a measure of the type of correlation between two features, being between -1 and 1.

Different correlation techniques can be used:

- Pearson, for normally distributed, continuous, linearly related variables,
- Spearman Rank, for ordinal variables (ranking).
- Chi-Square test to see if variables are related or not. For p-values < 0.05, the variables are correlated. For a p-value > 0.05, the variables are independent.

Encode textual data to numeric

Textual data has to be encoded, *i.e.* converted into numerical data to be able to be used by the ML algorithm. Data can be classified this way:

- Qualitative (categorical)
    - Nominal, mutually exclusive and unordered, e.g. gender
    - Mutually exclusive and orders, e.g. small, medium, large

- Quantitative
    - Discrete, integer values
    - Continuous, decimal values, height, weight…

Different encoding methods are used depending on the nature of the data.

a) Methods for Qualitative Nominal data: OneHot, LeaveOneOut encoding.
Ordinal encoding, *e.g*. Male = 0, Female = 1, can't be used as it will put an order in the data.

b) Methods for Qualitative Ordinal data: Ordinal, OneHot, LeaveOneOut encoding.

Feature engineering: Normalisation and scaling

Normalisation
Many Machine learning algorithms perform better when features are on a relatively similar scale and/or close to a normal distribution.
Techniques to test the normality or "Skenwness" of the data are by visual inspection, Q-Q plot and the Shapirto-Wilk test.

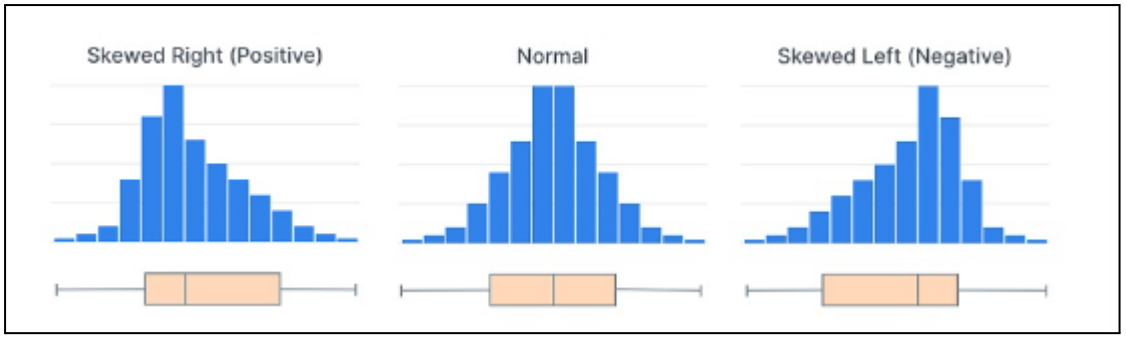

Fig.3 DIfferent type of data distribution (Skewness).[3]

Methods to normalised data are:

- Binning method
- If **right** skewed, log10, ln, square root, 1/values…
- If **left** skewed, 2^{value} , $value^2$, $value^3$, etc

Scaling

For example a feature is age, which ranges from 10-70, another feature is salary in the range 20,000 - 100,000. They need to be brought to the same scale.
Examples of algorithms that will perform better when data are normalised, are linear and logistic regressions, KNN, SVM, Neural Network.

A way to fix this issue is through scaling functions provided by SkitLearn:

- StandardScaler(), to achieve a normal distribution.
- MinMaxScaler(), default function
- RobustScaler(), which reduces the influence of outliers.

Splitting the Data into a Training and a Test Set

First step after data wrangling is to split the data.
Typically 80% of the data for training, 20 for the test. Or 70/30. The rows selected to form the two sets are selected randomly by the Python Sklearn train_test_split function.

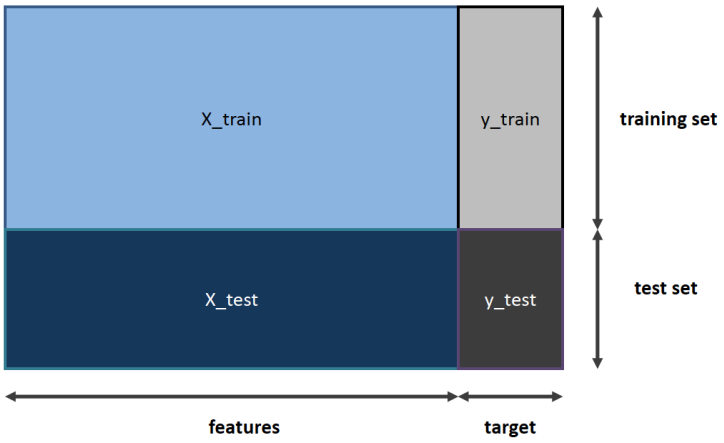

Fig. 4. Schematics of data splitting into a training set and a test set.

Summary of the development of a machine learning model

1. **Data Preprocessing**

This includes steps like cleaning the data, handling missing values, encoding categorical variables, and scaling features if necessary. Data preprocessing ensures that the data is in a suitable format for the machine learning algorithms.

2. **Splitting the Data**

The dataset is then typically split into training and testing sets. Sometimes, cross-validation techniques are used to ensure robustness in model evaluation.

3. **Model Selection.**

Based on the problem type (regression, clustering, classification etc.) and characteristics of the data. This could involve trying out multiple algorithms to see which one performs best.

4. **Model Training.**

The selected algorithms are trained on the training data. This involves learning the patterns in the data that will enable the model to make predictions.

5. **Model Evaluation.**

After training, the model's performance needs to be evaluated. This can be done using various metrics depending on the problem type. For classification problems, common evaluation metrics include accuracy, precision, recall, F1-score, ROC curve (Receiver Operating Characteristic), and AUC-ROC (area under the ROC curve).
The F1 score is a single metric that combines precision and recall into a single value. It is particularly useful when there are imbalanced classes in the dataset.
Precision (also called positive predictive value) measures the accuracy of positive predictions.
Recall (also called sensitivity or true positive rate) is the ratio of correctly predicted positive observations to all actual positives. It measures the ability of the model to capture all the positive instances.

6. **Confusion Matrix.**

This is particularly useful for classification problems. It gives a more detailed breakdown of the model's performance, showing true positives, false positives, true negatives, and false negatives.

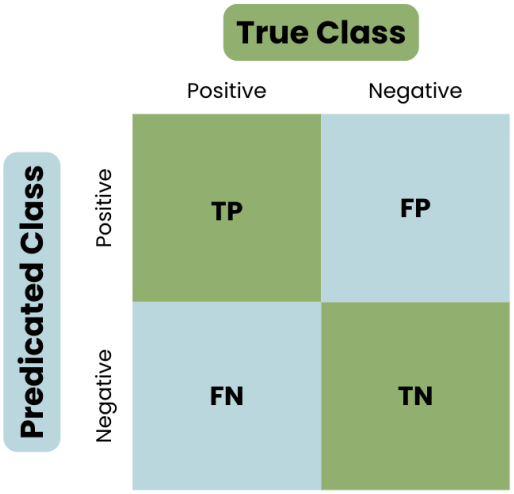

Fig. 5. Schematics of a Confusion Matrix.[4]

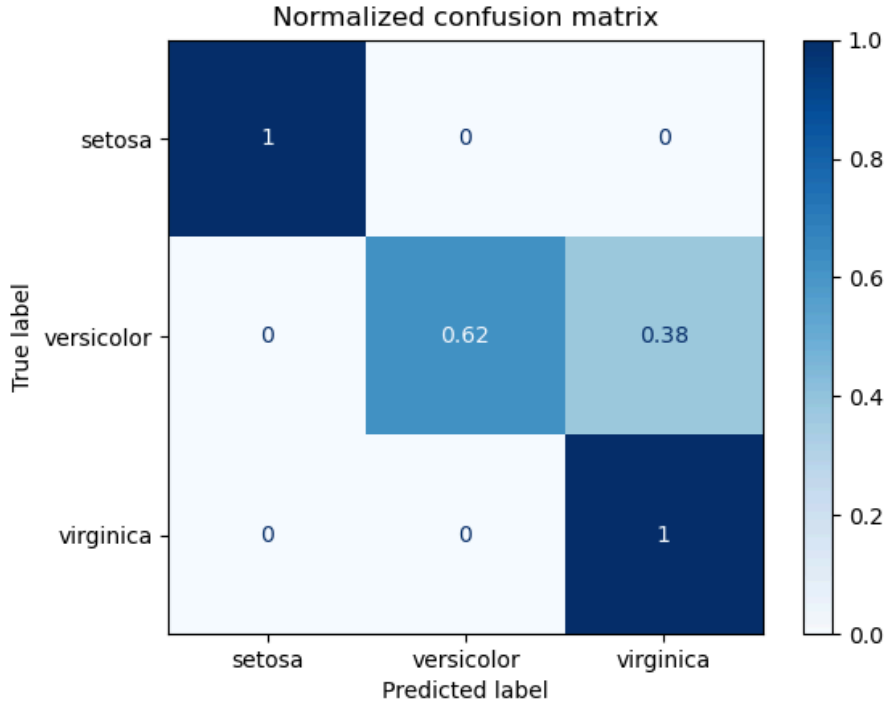

Fig. 6. Example of a Confusion Matrix.[5]

7. Further Analysis.

Depending on the specific requirements of the problem, additional analysis may be needed. This could involve analyzing feature importance, tuning hyperparameters, or examining the model's behavior on specific subsets of the data.

8. Iterative Process.

Model evaluation is often an iterative process. If the model's performance is not satisfactory, it may be needed to go back to earlier steps, such as trying different algorithms, adjusting preprocessing techniques, or collecting more data.

References

1. Kelleher, J. D., & Tierney, B. (2018). *Data Science*. MIT Press.
2. Howard, J., Goodfellow, I., Bengio, Y., & Courville, A. (n.d.). *shanmukh05/Machine-Learning-Roadmap: A roadmap for getting started with Machine Learning*. GitHub. Retrieved April 30, 2024, from https://github.com/shanmukh05/Machine-Learning-Roadmap.
3. Lee, I. (n.d.). *Data Skewness*. Wallarm | Integrated App and API Security Platform. Retrieved April 30, 2024, from https://www.wallarm.com/
4. guide, s., & Ahmed, N. A. (n.d.). *What is A Confusion Matrix in Machine Learning? The Model Evaluation Tool Explained*. DataCamp. Retrieved April 30, 2024, from https://www.datacamp.com/tutorial/what-is-a-confusion-matrix-in-machine-learning
5. *Confusion matrix — scikit-learn 1.4.2 documentation*. (n.d.). Scikit-learn. Retrieved April 30, 2024, from https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html