Successful Machine Learning project

Adil KORCHI, Fayçal MESSAOUDI, Lahcen Oughdir

Abstract— A Machine Learning (ML) project does not deal with a typical software project. Indeed, learning modeling differs completely from strict programming based on rules and exceptions and how to approach this type of project. A 100% agile method will not necessarily work, at least not at all stages. And to succeed his project of Machine Learning is therefore to respect some steps that we will address in the following.

Index Terms— Machine Learning - Algorithms - Project - Modeling - Clustering - Evaluation and scoring - Data Preparation

1 Introduction

Machine Learning is an artificial intelligence technology that allows computers to learn without having been explicitly programmed for that purpose. To learn and grow, computers need data to analyze and to train on.

If Machine Learning has been around for a while, its precise definition remains confusing for many people. Specifically, it is a modern science for discovering patterns and making predictions from data based on statistics, data mining, pattern recognition and predictive analysis. The first algorithms were created in the late 1950s. The best known of them is none other than the Perceptron which is a supervised learning algorithm of binary classifiers (separating two classes) (Collins, 2002). It is a type of linear classifier, and the simplest type of artificial neural network. A network of artificial neurons is a computer system inspired by the functioning of the human brain to learn (Maind, 2014).

And because a problem of Machine Learning is often complex to solve, breaking the problem into smaller steps will facilitate the management of his project and this is what we will demonstrate in this article in which we will go through the different stages and we will see above all the main elements for a Successful project using the machine learning technique.

2 Steps to Successful Machine Learning Project

A machine learning project is not a computer project like the others because it is not a classic development project and therefore has its own constraints but above all it will need great flexibility and regular readjustments.
The Machine Learning project does not deal with a typical software project. Indeed, learning modeling differs completely from strict programming based on rules and exceptions. So, of course, the same goes for how to approach this type of project. A 100% agile method will not necessarily work, at least not at all stages.
To succeed in such a project, it is necessary to respect the following steps:

1. Access & Data Analysis
2. Data Preparation
3. Modeling
4. Evaluation and scoring (Iteration)
5. Deployment (Regular Re-evaluation / Iteration)

It should be noted that if these phases are to be performed separately, in reality we will be more in an iterative approach. For example, a model adjustment is only possible if new features are added, so we will have to go back to phase 2 and 3 to add these new variables. Similarly it will be interesting in some cases to test new algorithms to test the relevance and level of error of our model (Snoek, 2012).
In what follows, we will go through all these different stages and we will mainly see the main elements to remember during the project.

2.1 Step N ° 1: Definition of objectives

While this step may seem obvious, it is nonetheless vital for the success of the project. Beyond the underlying business problematic, we need to determine what type of problem we need to solve. For this we need to know if we have experimental data with results or not (or even partial) to determine if we approach a problem of supervised or unsupervised type.
Then what is the typology of the problem to solve:

1. Regression
2. Classification
3. Clustering
4. Ranking
5. System of recommendations

For each typology, there are one or more algorithms that admit a particular mode to solve or at least approach the problem.

The following table is intended to summarize some of the most commonly used ML algorithms:

Learning	Algorithm	Typology	Comments
Linear Regression (univariate/multivariate)	Supervised	Regression	This is the most common and simplest mode of ML. The idea here is to make the model guess the equation that will allow future predictions. The univariate mode has only one variable (characteristics), it is therefore a simple straight line (y = ax + b) as for the multivariate mode it takes into account several other characteristics (attention to the normalization of the variables).
Polynomial Regression	Supervised	Regression	This is a particular extension of multivariate regression. To put it simply, the idea is to have a curve rather than a straight line (we are no longer in linearity)
Regulated Regression	Supervised	Regression	The idea here is to improve the regression models by adding shrink / penalty notions in order to reduce the space and thus remove the gross errors of the model. This is clearly a method of regularization. Penalty features: Regression Ridge, Lasso, Elastic-Net
Naives Bayes	Supervised	Classification	It's a classifier. Certainly the most used it is based on the probability law of Bayes. Its particularity: it is based on the fact that the characteristics are independent of each other.
Logistic Regression	Supervised	Regression	It is a classifier widely used because of its linear side. Its cost function is based on log loss, which strongly penalizes false positives & negatives.
K-NN (K nearest neighbors)	Supervised	Classification	Algorithm based on the proximity of the observations.
Random forest	Supervised Non Supervised	Classification	Fast, robust and parallelizable. The idea is to train several decision trees on random and different subsets of your dataset. In the end a democratic vote of your different groups gives you the prediction.
SVM (Support View Machine)	Supervised Non Supervised	Classification	Of type regression or classification. Very adapted to the classification of complex data but of reduced sizes.

Table 1. Most commonly used ML algorithms

IJSER

2.1. Step N ° 2: Access & Data Analysis

Here is a crucial step in which you will have to rework the data (features or variables). This is an essential operation because Machine Learning's algorithms do not accept any type of data. This is a necessary operation to refine the variables so that they are better managed by these same algorithms.

Data splitting

First you work with a dataset. You will have to cut it in two parts (minimum):

- Training Data: A subset for learning a model.
- Test data: A subset for the evaluation of the model. This dataset should not be used in the design of the model!

This division you will manage it from predefined functions for example via "sklearn.model_selection.train_test_split"; it is cross-validation and a resampling procedure used to evaluate machine learning models on a limited sample of data. But nothing is ever so simple because the way you're going to cut

up your data can be too big on your model. At this level already it will be necessary to be subtle and test several possibilities as "sklearn.model_selection.KFold" (Kahandagamage, 2017).

Data analysis

This is an equally important step in which you will have to:

1. Make an inventory of your data (data type) to define:
2. Typology: Digital, temporal, text, binary, etc.
3. Categorical variables, discrete or continuous variables?
4. Number of observations (number of lines)?
5. Number of features / variables (number of columns)?
6. Detect if you have forgotten and especially decide what you will do (delete or simply alter them)
7. Detect missing values
8. Detection of correlated variables / features

At this level, it is essential in my opinion to have a good dataviz tool!

2.2. Step 3: Preparing the data

Step N° 2 makes it possible to make a complete inventory of the data which you have, you will have now to prepare your features / variables so that they can be used by algorithms of Machine Learning.
To resume the previous points:

- ✓ You only need data (variables) in digital format. If you have type data:
  1. Date: Apply formulas to transform them into periods, etc. Why not add aggregations on slippery windows (on the week, the month, the year before)?
  2. Categorical: Use One-Hot encoding whenever possible. If you have too many variables, reduce the scope by grouping.
  3. Text: you will certainly have to cut, reformat your data to have categorical data
- ✓ If you have missing information (Null)
  1. Delete the entire line if you really have a lot of data (not recommended but sometimes you will have no choice)
  2. Replace them with a value, the median value, the average, etc.
- ✓ Scaling the numerical values (feature scaling)
- ✓ Going to logarithm when variables have extreme values. This reduces their importance.

This step is also called feature engineering! (Domingos, 2012)
Another important aspect is the management of its datasets: the creation of work datasets. If you have a unique dataset you will have to build a training dataset and a test game!

2.3. Step 4: Modeling

During this step you will choose the machine learning algorithms that seem best suited to you.

Depending on the problem you are going to deal with, you have the choice of algorithms, draw and test! This phase can be long (time) because training is a very heavy task especially when you have a lot of data (which is also recommended).
The difficulty is not in this choice but rather in the adjustment of hyper-parameters that you will have to do in order to obtain a powerful model.

2.4. Step 5: Evaluation & scoring

Your algorithm thus chooses and the hyper-parameters are adjusted, you will have to validate your model. It's impossible not to enter an iterative mode in which you will fumble with these hyper-parameters. Do not hesitate to use third-party tools or approaches such as search by gille (grid-search).
Be careful especially over-fitting (or over-training) that will give you the illusion of a good model!
Indeed if you exceed a certain score (around 95%) it is likely that your model is high-performance but for your training data. So try it with the test data, you will certainly be surprised!
Regarding how to measure performance it differs according to the type of problem but also according to what you really want to measure. Several measures are available (non-exhaustive list):

- ✓ **Classification**
  1. Confusion matrix
  2. ROC Curve
  3. Accuracy / Reminder
- ✓ **Regression**
  1. Prediction error
  2. XY graph value to predict / predicted value
- ✓ **Clustering**
  1. Intra-class variance, interclass
  2. Number of cut bow

You are therefore entering an optimization phase based on an inevitably iterative approach. Here are some ways to improve and / or optimize:

- ✓ Algorithm change
- ✓ The distribution of tests / entraintement is coherent, homogeneous?
- ✓ Add / delete variables
- ✓ Grouping of values: add averages, sum, number in groups.
- ✓ Add / delete rows (with new data sources)
- ✓ Adjustment of hyper-parameters
- ✓ Add combinations of variables difficult to learn for a model like a ratio
- ✓ Aggregate over longer periods (eg 1 month for a daily granularity) can be a good track
- ✓ Use the output of another machine learning model.

IJSER

- ✓ Look for information that could help a model correct errors

2.5. Step N° 6: Deployment

Your model is ready. It is efficient and can adapt to all situations. You can now deploy it through an API or integrate it directly into a computer program (Buitinck, 2013). Be careful because by nature a model can not live forever (it is indeed based on a learning on data and the data evolve constantly). It is therefore necessary to plan to check regularly.

3 Conclusion

This manuscript was intended to provide the steps to be followed in Machine Learning projects. We were able to know that such projects can not be carried out classically because learning modeling differs completely from strict programming based on rules and exceptions. However, there are several methods to succeed ML projects such as the CRISP method that seems to be the most suitable for conducting Big Data projects, provided that it is applied in its entirety. Indeed, no stage is superfluous, only the time spent on each of them can vary from one project to another, even from one iteration to another and the studies carried out in this field affirm that no methodology is perfect, and the key to success will always lie in the constant involvement of the businesses for a continuous improvement of the final product. After all, as the statistician George E. P.BOX said : "All models are wrong, but some are useful".

REFERENCES

[1] BEAM, Andrew L. & al. "Big data and machine learning in health care". Jama, 2018, vol. 319, no 13, p. 1317-1318.

[2] BIAMONTE, Jacob, & al. ".Quantum machine learning. Nature ", 2017, vol. 549, no 7671, p. 195.

[3] BOTTOU, Léon, & al. "Optimization methods for large- project",. arXiv preprint arXiv:1309.0238, 2013.

[4] BUITINCK, & al., "API design for machine learning software: experiences from the scikit-learn scale machine learning". Siam Review, 2018, vol. 60, no 2, p. 223-311.

[5] BRADLEY, Andrew P. "The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern recognition", 1997, vol. 30, no 7, p. 1145-1159.

[6] COLLINS, Michael. "Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In : Proceedings of the ACL-02 conference on Empirical methods in natural language processing", Volume 10. Association for Computational Linguistics, 2002. p. 1-8.

[7] DENG, Li, YU, Dong, & al. "Deep learning: methods and applications. Foundations and Trends® in Signal Processing", 2014, vol. 7, no 3–4, p. 197-387.

[8] DIETTERICH, Thomas G. "Ensemble methods in machine learning". In : International workshop on multiple classifier systems. Springer, Berlin, Heidelberg, 2000. p. 1-15.

[9] DOMINGOS, Pedro M. "A few useful things to know about machine learning". Commun. acm, 2012, vol. 55, no 10, p. 78-87.

[10] DONG, Chao, LOY, & al. "Learning a deep convolutional network for image super-resolution." In: European conference on computer vision. Springer, Cham, 2014. p. 184-199.

[11] EL KOURDI, Mohamed, & al. "Automatic Arabic document categorization based on the Naïve Bayes algorithm". In : proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages. Association for Computational Linguistics, 2004. p. 51-58.

[12] FERBER, Jacques & al. "Multi-agent systems: an introduction to distributed artificial intelligence". Reading : Addison-Wesley, 1999.

[13] GOLDBERG, David E. & al. "Genetic algorithms and machine learning. Machine learning", 1988, vol. 3, no 2, p. 95-99.

[14] GOODFELLOW, Ian & al. "Deep learning". MIT press, 2016.

[15] HAMMING, Richard W. & al. "A. Problem solving methods in artificial intelligence". 2017.

[16] HERTZ, John A. Introduction to the theory of neural computation. CRC Press, 2018.

[17] KAHANDAGAMAGE, & al. "K. S. Sinhala Intelligent Word Recognition with Content based Search Suggest". 2017. Thèse de doctorat.

[18] KRISHNAMOORTHY, C. S. & al. «Artificial intelligence and expert systems for engineers". CRC press, 2018.

[19] LU, Huimin, LI & al. "Brain intelligence: go beyond artificial intelligence. Mobile Networks and Applications", 2018, vol. 23, no 2, p. 368-375.

[20] MAIND, & al. "Priyanka, et al. Research paper on basic of artificial neural network. International Journal on Recent and Innovation Trends in Computing and Communication", 2014, vol. 2, no 1, p. 96-100.

[21] MÜLLER, Vincent C. & al. "Future progress in artificial intelligence: A survey of expert opinion". In Fundamental issues of artificial intelligence. Springer, Cham, 2016. p. 555-572.

[22] PAN, Yunhe. "Heading toward artificial intelligence 2.0. Engineering", 2016, vol. 2, no 4, p. 409-413.

[23] SEBASTIANI, Fabrizio. "Machine learning in automated text categorization". ACM computing surveys (CSUR), 2002, vol. 34, no 1, p. 1-47.

[24] SNOEK, Jasper, & al. "Practical bayesian optimization of machine learning algorithms. In: Advances in neural information processing systems". 2012. p. 2951-2959.

IJSER