Introductory Machine Learning for Non-Major Students

Anonymous Authors[1]

Abstract

Data Science in general, and Machine Learning in particular, is a powerful tool for decision-makers across non-STEM fields like Human Resources Management, Law or Marketing. Introductory Machine Learning, for non-majors that lack a strong background in Statistics and Computer Science, is a challenge for both teacher and students. The use of similes and games is a soft way to deal with definitions of concepts and procedures that are essential for further advanced courses on these subjects.

1. Introduction

Machine Learning (ML) is a field underpinned by Statistics, Calculus, Algebra and Computer Science, with a long history in STEM curricula. During the last decade, its practical applications made it quite popular outside academia. Fostered by the explosive growth of digital data and availability of cheap and ubiquitous computing power, Machine Learning is now a common subject in non-STEM graduate and undergraduate programs (Plasek, 2016; O'Neil, 2014).

Research about teaching ML to non-majors is a developing field (Way et al., 2017; Long & Magerko, 2020). A recent survey showed that maths and programming are the two main barriers for these students (Sulmont et al., 2019). There are three possible choices to deal with these obstacles. First one is ignoring the students' background and lecturing them as if they were following the STEM path. The second one is pure story-telling, just providing some application examples without a long-term strategy to teach effective Machine Learning skills. The last one is a middle way, where concepts and definitions are taught using analogies and games (Opel et al., 2019) as a preliminary step to introduce abstract concepts as information entropy or ROC. This approach may be misleading, as Edsger Dijkstra noted (Dijkstra, 1989), if they are not related to the mathematics and technology that are the basis of ML.

In this communication I describe the instructional design of an introductory Machine Learning course initially intended for undergraduate Law students.

Comillas is a Private Catholic University, with a century-long history and a main campus in the heart of Madrid, where it offers undergraduate and graduate courses in Engineering, Law and Business Managament. By mid 2010's the governing bodies made a strategic move to offer what are called Doubles Degrees in the context of the European Higher Education Area (Klebes-Pelissier, 2007), a ten semester program. The first one was *Law and Business Analytics* set to start by September 2017, followed by *Business Management and Business Analytics* and *Engineering and Business Analytics*. Students that get a Double Degree in Law and Business Analytics may follow a Master in Laws that is mandatory for the State Examination for Access to the Legal Profession, but they may also choose to build a career in the booming Legal-tech industry.

First contact of these students with Business Analytics is a 30 hours introductory course during the third semester. By the end of the five years they become very proficient in programming and building models, but at that moment they haven't written a line of code and have just followed the basic Calculus course. On the other hand, requests to enroll this Double Degree are high, so the selection process is quite competitive and average students' performance is remarkable.

The syllabus of the course starts with fundamentals of programming and statistics with the `R` language and Exploratory Data Analysis, 8 hours in total. They have to master both `R` and `Python` by the end of the ten semesters, but this is the first contact with a programming tool, so we decided to start with `R` because of the simplicity of the environment. Anyway, it is a challenge to install and configure `R` and `RStudio` with a variety of devices, and as a consequence installation of additional packages is hard.

2. Explaining ML Principles

Lessons 4 to 7 cover Machine Learning fundamentals. One of the problems when teaching Data Science is the misuse of vocabulary in mass media and the hype about its unlim-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

ited applications to solve business problems. Students have heard of read about Big Data, Machine Learning, Algorithms and Business Intelligence almost as synonyms. The goal of the first lecture (2 hours) is to make clear what learning and prediction mean in this context, what are the steps to train, evaluate and deploy a model and when you should and shouldn't use predictive models.

Non-major students, without a prior contact with Data Science, may have heard some kind of buzzword-filled discourses about the almost-magical powers of machines. To spoil this idea I start the lesson with a powerful citation: "Let's start by telling the truth: machines don't learn. What a typical learning machine does, is finding a mathematical formula" (Burkov, 2019).

This theatrical introduction sets the stage to explain how learning works (Fig. 1), following the steps described in (Abu-Mostafa et al., 2012). Although the book is excellent, its level of abstraction is hard for the group at that moment. To make it simpler, I have found that linear regression, is a quite convenient example to understand it step-by-step. Students know what linear regression is and how it works despite they do not identify it as a predictive model yet.

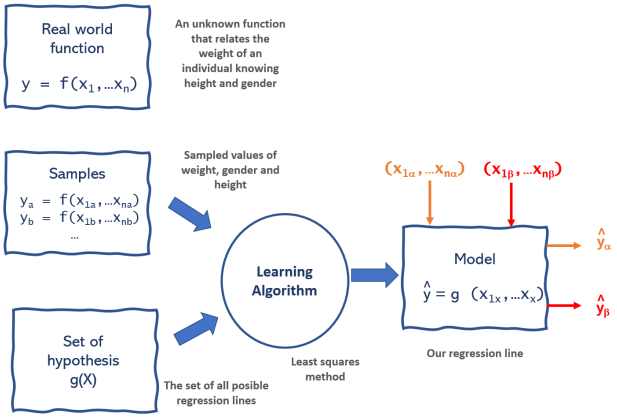

Figure 1. The machine learning process, a free adaptation of (Abu-Mostafa et al., 2012).

So, I propose an hypothetical project to predict the weight of an undergraduate student just knowing its height and gender. This example is simple enough to explain some fundamental ideas of Machine Learning:

- We have a set of vectors X that tell a story about some function $f(X)$ that we will never know in detail. This first statement is critical to understand that ML always work under uncertainty. If we had an analytical expression of $f(X)$ ML is pointless. Law and Business Management students understand it quickly as uncertainty is a natural element of their world view. Engineering undergrads have a much more difficult time to accept this fact.
- If the samples are a good representation of the population, we may make some educated guess about a function $g(X)$ that is close enough to f for predictive purposes.
- We choose a possible family of functions to search for that potential $g(X)$, that we call **model**, and an algorithm selects, among all possible $g_i(x)$, the best one according to a mathematical rule called **loss function**. Finally, students identify *least squares* as the algorithm and the regression line as the model.

We sample the height values of five students of the same gender and build a regression model with `R` and, in general, results of guesses are accurate because they all are healthy teenagers. Then, we check that it doesn't work as well if we try to predict values of individuals of the opposite gender. We start a short discussion on the possible origin of this mismatch and they identify that we haven't included the feature *gender*, so we rebuild it and things go more smoothly. When they are happy enough I tell we are going to play the *evil overfitting game*. I build a black box model that returns the exact weight of the individual if it was one of the building samples, and a random value otherwise, and that is a bit of a shock.

The example may sound a bit absurd from a business point of view. I translate it into an environment management agency that needs to know the biomass of salmons that return a given year through a ladder with a trap and a camera that measures the length of a percentage of individuals.

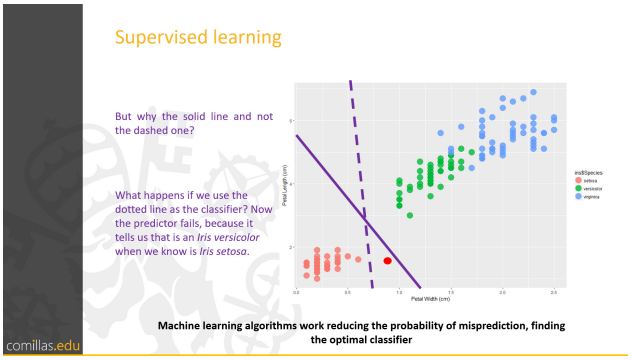

Figure 2. A simple classification problem.

After the introduction to regression, dangers of classification are illustrated with Fisher's Iris dataset. Despite criticism about its goodness to teach ML concepts, the dataset has the nice property that provides two classification problems that are visually quite evident. First, I draw a straight line to

divide *setosa* and *versicolor* samples. I make the pun that I used L'Oréal's Rule to choose the optimal slope (*Because I'm worth it*), and show how changing it misclassification happens (Fig. 2). Then we attack the non-linear separable classification of *virginica* versus *versicolor* to stress the idea that uncertainty is an inevitable fact for ML models and a visual example on how you can get a zero training error and a weak classifier.

The confusion matrix is well understood with the *boy who cried wolf example* of Type-I error. The ROC and AUC concepts are not easy to grasp, so I try to put it simple, AUC is just an ordinal index for a given prediction problem and a given dataset in order to compare results.

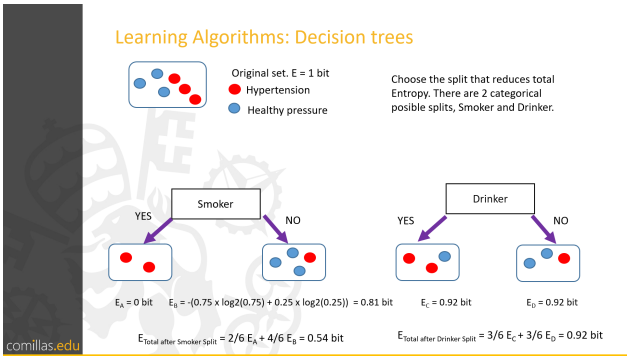

Figure 3. Building a decision tree by information entropy reduction.

Lesson 5 deals with mainstream ML models: decision trees and their families, logistic regression, neural networks and SVMs. This is tough stuff for beginners. We play another game to discover what *information entropy* is and how a decision tree works by reducing it, the classical *Guess Who*. A student thinks of another one and the group has to guess who is he/she with Yes/No questions. As these groups are quite well gender-balanced, the obvious first question is "Is a girl (or boy)?". Decision trees are very intuitive because our brain works the principle of entropy reduction but each round I give this lecture I find that entropy formula is the worst nightmare for students. The minus sign and $log_2(p_i)$ are the most feared beasts.

The basic idea of merging the results of several trees using a Random Forest or a Decision Jungle is easy to teach, as the use of a regression (logistic) to classify setting a simple boundary.

Neural networks are a hard mathematical topic, so I just show an animation of how the Perceptron training algorithm works (remembering the L'Oréal pun). As the backpropagation algorithm is out of reach at this level, I explain the concepts of **black box** and **interpretability of models**.

The second lesson ends with Support Vector Machines and an explanation on how a kernel transforms a linear non-separable problem into a separable one, using a visual example from one to two dimensions.

After these 4 hours of theory we have a 2 hours hands-on session. Despite students have got a basic knowledge of the `R` language I feared that trying to build a toy model would be a painful experience. Instead, we use a user-friendly graphical environment, `Azure Machine Learning Suite`. There are similar tools like `KNIME` or `BigML`, the choice was driven by reasons of convenience, as the agreement between the University and Microsoft includes a basic license.

With a visual environment like that, we build a classifier from scratch in 45 minutes, identifying each machine learning step. Students love the ability to test different models just adding components with a drag and drop order. Comparing AUCs awakes their competitive instinct, the basis of the final act. We also build a regression model although RMSEs are not as exciting as AUCs.

Figure 4. Example of competition, building a predictor to know if a defendant will be declared guilty on fraud charges and assign a senior or junior lawyer according to the result.

This part of the course ends with a Kaggle-like competition. I provide a synthetic dataset, a user case and the rules of competition. Students work in groups of six chosen at random, to teach them an important lesson, in real life you have to collaborate with people that are not part of your circle of friends. After three weeks they send a report, with the best AUC they got and publish their model. This exercise worths a 10% of the final grade, the winner team earns an extra 10%, so they put a lot of effort in this assignment.

I give some hints on features engineering and transformations and they apply all their knowledge with enthusiasm. Usually, some clever team discovers how to balance the samples with a SMOTE procedure, and even try ensemble methods that we did not see during the hands-on session.

Next lesson is an introduction to non-supervised models, in two hours and working with R. After the supervised experience, nothing is so exciting for them.

References

Abu-Mostafa, Y. S., Magdon-Ismail, M., and Lin, H.-T. *Learning from data*, volume 4. AMLBook New York, NY, USA:, 2012.

Burkov, A. *The hundred-page machine learning book*. Andriy Burkov Quebec City, Can., 2019.

Dijkstra, E. On the cruelty of really teaching computer science (the sigcse award lecture 1989). *SIGCSE Bulletin: A Quarterly Publication of the Special Interest Group on Computer Science Education*, 21(1):xxiv–xxxix, 1989.

Klebes-Pelissier, A. Double degrees in the context of the bologna process. *European Journal of Legal Education*, 4(2):173–180, 2007. doi: 10.1080/16841360802338738. URL https://doi.org/10.1080/16841360802338738.

Long, D. and Magerko, B. What is ai literacy? competencies and design considerations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–16, 2020.

Opel, S., Schlichtig, M., and Schulte, C. Developing teaching materials on artificial intelligence by using a simulation game (work in progress). In *Proceedings of the 14th Workshop in Primary and Secondary Computing Education*, pp. 1–2, 2019.

O'Neil, M. As data proliferate, so do data-related graduate programs. *The Chronicle of Higher Education*, 60, 2014.

Plasek, A. On the cruelty of really writing a history of machine learning. *IEEE Annals of the History of Computing*, 38(4):6–8, 2016.

Sulmont, E., Patitsas, E., and Cooperstock, J. R. Can you teach me to machine learn? In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, pp. 948–954, 2019.

Way, T., Papalaskari, M.-A., Cassel, L., Matuszek, P., Weiss, C., and Tella, Y. P. Machine learning modules for all disciplines. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education*, pp. 84–85, 2017.