Day 0 Intro to ML: Story of the Riding Bike

Benjamin Manning, PhD

Nov 15, 2018 · 8 min read

I believe we learn best from our experiences and we can share this same learning when we share stories about our experiences.

I enjoy teaching things to people; one topic I really enjoy talking about and teaching about is Machine Learning (ML). I have taught ML to all sorts of students at all skill levels over the years. Do not believe the hype — learners do not need any type of specific background or domain knowledge in any given area to understand and even apply ML concepts to problems. Sure — experience in any area helps, but many would have prospective learners that intense Stats or Math skills are needed in order for one to get started and that just is not the case. In fact, in the remainder of this article, I will show that only a general understanding of our own learning is needed for anyone to grasp and fully understand the entire ML process.

Before those more experienced folks scream at me, keep in mind this was written with the Day 0 learner in mind to provide an overall, easier-to-understand-without-scaring-the-hell-of-out-them methodology.

So let us get started — with a quick story!

One of my fondest childhood memories is of my father teaching me to ride my new bike. I think this was my first ‘real’ hard knocks experience since my father didn’t show me how to ride my bike, but rather allowed me to fail slightly each time I tried and eventually fell off. Seemingly, Id never get there, but slowly and surely I deduced what I was doing wrong based in the lessons I learned about what I was doing right each time I tried to ride. Throughout my experiences and experiments, I adjusted each on of these variables until the outcome resembled some semblance of what was needed to breeze down our dead-end street on my own and ride with the fastest of my friends.

Little did I know at the time, every iteration represented a learning experience regardless of the outcome — failure or success — I was learning how to manage things like balance, momentum, speed and even grit.

> The noblest pleasure is the joy of understanding.
>
> Leonardo da Vinci

How do we learn?

Without getting all Sciency, let’s boil learning down to two things: an experience of some type and the related skill or task we learned from the same experience.

Learning to ride (experience)

How did you learn to ride a bike? Chances are you either followed one of two paths:

You tried to learn on your without any help; you tried and failed numerous times, but after tons of scraped knees and bruises, you successfully rode a bike on your own for a distance… and then fell off again.

or

You had someone help you learn. Maybe a sibling (this often went very wrong for those of us with older siblings) or parent either showed you how to ride or you watched how they rode and mimicked what they did. You still tried and failed numerous times, but after tons of scraped knees and bruises, you successfully rode a bike on your own for a distance…and then fell off again.

The similarities between these two methods are easily reduced to this simple form: ***A repetitive experience, whether by failure or success, taught us: both what to do and what not to do.***

> Never let formal education get in the way of your learning.
>
> Mark Twain

Outcome (Learned Skill)

The end or outcome of each process was the same — we learned to do something — now what?

Application

This is the point in my classes where I often tell students — now use this new skill (riding a bike as a child) and use it to go ride a motorcycle! Hmm — puzzled looks abound. How can that work?

It is a simple and often too obvious (remember — people say learning this ML stuff is hard) answer — just add new experiences and outcomes to what you have already learned. Chances are — this time the learning will go much faster and there will be much less pain (scrapes and bruises) since riding a bike and a motorcycle share similar skills like balance and momentum.

So how does all of this relate to ML?

What makes ML sometimes difficult is not learning or even understanding the process (I will relate all of this in a sec), but rather the fact that we are studying experiences and outcomes that we have not already experienced or learned ourselves personally. More simply put — we cannot relate to the learning experience because we did not learn it ourselves at some point — literally.

Now — the whole ML process — related to learning to ride the bike.

A. Preprocessing

Any successful analytics or ML process starts with good data and that often has to be cleaned and even transformed prior to being used. Can you image learning to ride a bike with half of the steps involved in the process missing or even presented to you in another language? In order to fully learn from the experience, the data within it must be free of errors or we cannot fully learn from the experience or use it to learn any type of skill — this first step is called preprocessing and we correct errors and clean data during this process.

B. EDA

Remember what I mentioned sometimes makes learning ML difficult? If I asked you to teach me a skill that you had only watched someone do, but had never learned yourself could you teach me? Probably, because humans are excellent learners, but hands down you would learn new things during the process yourself and I would only learn some of what I needed to accomplish the new skill.

Bottom line is we have to first fully educate ourselves about a new task or skill when we do not have any prior experience learning it. In ML, this process of learning as much as possible about the data is called Exploratory Data Analysis or EDA for short. We use all sorts of visualizations and tools that can help us accomplish this, but, for now, just know why we do this — to learn what we do not know.

C. Feature Selection / Feature Engineering

Experiences, as we all know as adults, can be complicated. Parts of experiences can also be convoluted, overlap, intersect, subsect and even be irrelevant. In ML, the parts and pieces that make up experiences are called features and, at a high level, we often just relate features to outcomes or learned skills. As we have discussed, experiences are often iterated during the learning process, but the features remain the same type each time.

Back to the bike example: each time you tried to learn to ride your bike you likely varied the amount of balance each time until you learned just the right amount to keep you upright. During each repetition you also learned to vary your monumentum until you learned how much was needed to keep you moving forward until you started pedaling on your own. In this example balance and momentum are both features related to an outcome: either successfully riding or unsuccessfully riding.

So — why do we need to select features?

Because we are eventually going to use what we build in some type of process, our ML process has to be optimized — so we study the data internally in EDA to possibly decide if we can reduce the size of our problem by removing any unneeded features from the experiences. This can not only speed up the learning, but also make it better by removing features that the ML process might find detrimental to learning (think about learning to ride a bike if someone tells you the wrong way to ride).

Overall, this reduction helps to optimize computational resources as well. When the features are too numerous to manually reduce their number or when other more, specific algorithmic methods are needed, Feature Engineering methods are implemented to normally accomplish similar goals.

Now — my data and features are ready — what is next?

D. Modelling

Remember learning to ride the bike? Now we're going to package that entire process, including all of the repetitions, features and related outcomes of each repetition — all of it needed to adequately reproduce everything needed to learn a specific skill. This package is called a model and we have to bundle it up because we need to apply it to new experiences that occur later, but contain no outcomes — these will eventually be predictions.

But models cannot just be made on their own — this is where the ML comes into play — we must use the appropriate ML tool (called an learning algorithm) and train it on the experiences and outcomes we have collected (this is Supervised Machine Learning, but there are more types). As the model is constructed, the ML algorithm will try its best to learn how the experiences relate to each outcome — this happens for each repetition (we can now refer to repeated experiences as row(s) in the data).

Model building occurs in a separate training and testing phase until an optimal result is reached; there are numerous ways to measure this and it varies greatly on the problem, but this is a little outside of this introduction. For now — its adequate to just understand that models are assessed and this occurs through iteration, including retraining the model using a different ML tool, if needed. After an optimal model (remember this model now contains rules for how the experiences relate to each outcome) has been trained, tested, assessed and accepted, we can then use it on new data with the same features to make predictions.

At its core — this is Machine Learning — nothing more. ML tools are just that — tools — no magic hat.

Surely there is more to the process — yes there is and I will add a 2nd chapter to this introduction soon with a deeper dive, but beginners can stop here for a small and quick intro to what all of the fuss is all about.

Machine Learning Data Science

About Help Legal