



Photo by [Jakub Kriz](#) on [Unsplash](#)

A CAREER SWITCH AS ELSA SAID, IS A JOURNEY "INTO THE UNKNOWN"

# I had no idea how to write code two years ago. Now I'm an AI engineer.

My journey in its rawest form



David Chong

Follow

Dec 26, 2019 · 7 min read ★

**T**wo years ago, I graduated college where I studied Economics and Finance. I was all set for a career in finance. Investment Banking and Global Markets — those

were the dream jobs. 9 months before graduation, I snagged a role at an investment bank, feeling proud because it was typically hard to get a role if one hadn't interned at that bank before.

Months into the job, I picked up some *Excel VBA* and learnt how to use *Tableau*, *Power BI* and *UiPath* (a Robotics Process Automation software). I realized I was more interested in picking up these tools and learning to code rather than learning about banking products. Banking products that once intrigued me due to their complexity, is now seen merely as a way for banks to make profits off clients.

The banking environment is one that challenges my personal values greatly, but that's a topic for another day.

At the same time, a colleague I worked with opened my eyes to the world of 'Machine Learning'. I was intrigued by the fact that one could "predict" certain outcomes from certain inputs.

I was instantly hooked.

There was a problem though – my coding skills were elementary. In my dictionary, *Python* was a type of snake and *Pig* was well... a pig.

Fast forward two years later, here I am about to enter the AI industry to become an AI engineer. The journey wasn't easy, nor was it short. To me, a transition to the AI industry is merely the beginning — a new beginning for me to learn and grow. Here's what my journey was like.

## Disclaimer

Everyone's journey to data science is different. This article is not about "How to break into AI" and should not be viewed as a step by step guide. It is a personal anecdote that *I hope* will inspire people to take that leap of faith to do what they want, because life is too short to be living a life void of meaning.

## My Journey

### Jumping on the MOOC Hype Train

With an Economics and Finance background, I didn't know how to code. *Excel VBA* is pretty close to coding, but that's about it. Being an overachiever and wanting to jump onto the data science bandwagon, I signed up for a handful of Massive Open Online Courses (MOOCs). Not one, but a handful. Here's the list of courses that I signed up for:

- **Python BootCamp: Go from zero to hero in Python 3 [Udemy]**
- **Python for Data Science and Machine Learning Bootcamp [Udemy]**
- **Managing Big Data with MySQL [Coursera]**
- Java Tutorial for Beginners [Udemy]
- The Web Developer Bootcamp [Udemy]
- Machine Learning A-Z: Hands-On Python & R in Data Science [Udemy]
- Deploy Machine Learning & NLP Models with Docker [Udemy]

Needless to say, I didn't complete most of them (only the ones in bold). I was stuck in what I call a *MOOC self-perpetuating cycle*. The ease of access to knowledge made it natural to go from one course to the next, often not completing the prior course due to the brevity of the topics covered which made my interest fleeting.

To me, that was the biggest downside of MOOCs — the brevity of the content covered. Or perhaps my initial expectation that MOOCs could propel me to a career in data science was simply too naive.

To give perspective, a typical MOOC that teaches traditional Machine Learning (ML) methods often skims the surface of what the models actually do. **Random forests** are taught as an aggregation of decision trees, but how decision trees determine which features are selected at which branch (i.e. concept and mathematics of entropy) will not be covered. **Support Vector Machines** are simply taught as a classification method, but how the hyperplane is determined will not be covered.

This discrepancy between “*What I know*” versus “*What I need to know*” is exemplified when we look at more advanced areas of AI such as Deep Learning. MOOCs that teach Deep Learning often throw you a bunch of code in **Tensorflow** on a nice dataset like MNIST and conclude that you are now a Deep Learning expert (*I exaggerate a little but*

*you get the gist*). This is clearly far from reality as research papers often include complex architectures that involves understanding of feature extraction in deep neural net models, and other more complex features like transformers and bi-directional coding. Understanding why some state-of-the-art models work better than others is important as well, alongside concepts like transfer learning and meta learning.

In my opinion, MOOCs that taught ML *often* give people the false impression that anyone can be a ML practitioner. To the naive, ML is simply a few lines of code that involves `.fit()` and `.predict()`, and that is because MOOCs represent it in such a way that one can get started with ML with relative ease (*and perhaps monetizing these courses were so lucrative due to the massive hype around ML*).

Don't get me wrong — MOOCs are great. They provide a quick and easy way for people to gain knowledge and get started on a certain topic. Will they make you an expert? No. What you do *after* finishing the course, will determine whether or not you become an expert.

## Getting my hands dirty

After completing several MOOCs, I knew I was getting nowhere. Sure, I had some basic skills in *Python* and knew how to use *sci-kit learn* in a `.fit()` and `.predict()` sense, but that was about it.

To brush up my coding skills, I practiced on Hackerrank and completed questions related to *SQL* and *Python*. At the same time, I wanted to have a real-life project where I could utilize *Python*. This was when I started working on a bot that could book badminton courts for me. This mainly involves using Selenium to interact with the browser to navigate through the web pages and eventually cart out and pay for the badminton court (*this is similar to sneaker shoe bots*). The motivation was that badminton courts in Singapore are usually fully booked two weeks in advance and many people often camp on the booking sites everyday on the release time (*they usually get booked within a second or two*).

Even though I was somewhat more confident about writing code in *Python*, I knew nothing about code efficiency. Time and space complexity were completely foreign to me. Object-oriented programming was a concept in my head that never got its moment to shine (not to mention eventually forgotten).

On the ML side, I was a Jupyter notebook expert. I could change the theme of my Jupyter notebook to 'dark mode' and use all the keyboard shortcuts. *Clearly*, I was ready to take on the role of a Data Scientist.

Reality check — I failed interviews miserably. Technical tests involving code got both my foot out the door faster than I can say “data science”. A role which I applied to as a technology analyst got me referred to another department as they felt I would be better as a business analyst.

I was nowhere close to where I needed to be.

## Dirty but not dirty enough

To gain that depth in ML and hone my skills in Python, I decided to take up a Masters of IT in Business (with a specialization in AI) at Singapore Management University.

I learnt about the mathematics behind the traditional ML models and applied state-of-the-art deep learning architectures on a self-curated dataset. I learnt about important concepts about AI including common search algorithms, Q-learning and deep Q-learning. I learnt about algorithm design, including graph algorithms, time and space complexity, name-matching algorithms and many more algorithms that almost literally tore my non-computer-science brain apart. In essence, the course provided me the academic rigour that MOOCs lacked.

At this point in time, I had a few projects under my belt from my Masters. They weren't full fledged projects as the datasets were often provided or obtained from *Kaggle*, and they often culminated in a Jupyter notebook. Deep Learning models were ran on Docker for consistency but there was never an aspect of deployment in mind. They were after all, school projects.

In my opinion, a Masters provides the academic rigour that is necessary for AI professionals, but lacks the aspect of real-world application. A Masters course wouldn't tell you what is necessary to land a job in data science — you'd have to figure that out on your own. Software engineering and DevOps skills are often required as part of a Data Scientist's job scope (though not extensive). Collaboration of code is also important in large organizations. As such, knowing how to setup a Docker environment, launch an AWS EC2 instance, host a dataset on Azure blob storage, organize code efficiently and

using GitHub or GitLab for version control are some of the crucial skills needed but aren't taught in the classroom.

## Try, even when you think you're not good enough

I continued interviewing, raking up tons of experience in technical interviews and non-technical interviews (though flunking most of them). It also allowed me to know what I didn't know and took time to pick up those skills. More importantly, it allowed me to understand the different types of job descriptions that companies put up for the same role and how that corresponded to a company's maturity in AI adoption.

Two years later, I landed a role to train as an AI Engineer. To me, it is a great opportunity for me to learn and grow in a field that I am passionate about. More than that, it is a proof that anyone can accomplish what they set out to do, though some may take longer time than others.

At the end of the day, a career is a marathon, not a sprint. Do what you love, because you will be spending a huge proportion of your life working.

If you're feeling lost, remember what Elsa said and "do the next right thing".

. . .

Happy to receive any comments :)

Artificial Intelligence

Machine Learning

Careers

Work

Data Science

Medium

About Help Legal