This is the first installment of a blog thing I'll be trying to keep alive. The mission of the blog is essentially to act like an open-source guide (diary?) to the things that I learn about.

More fundamentally, I've wanted to keep a log or decision guide of some sort for a while now, just to allow me to look back in order to gauge growth/trajectory, and to look at the flow of my decisions and make improvements if need be. But I struggled with implementation, and social/peer pressure is always a great motivator (appealing to our base instincts as social animals) so I'm hoping this format will help me in this case.

Enough about me though, in order to provide a benefit to potential readers (hi) as well as to myself, I'll try to explain the things I've learned in my own voice and at various levels of difficulty/comprehension.

So, to kick things off:

**Machine Learning (basics)**

I've always found the human thought process (particularly in a stages/levels of thought type model and thinking about how suboptimal decisions are made) very interesting. So, it is quite rational that I pursue the intersection of my personal interest in human decision making with my interest in quantitative fields, and, albeit lesser, interest in coding.

1. What is Machine Learning?

Simply Machine Learning is a subset of Artificial Intelligence which deals with primarily two things: 1) classifying things (classification) and 2) making quantitative predictions (also called regression).

1. Why Machine Learning?

Similar to models in Economics, Machine Learning models help people approximate reality by exploring relationships (variables/features) and using these models to make predictions or give general insight into the underlying relationships.

That's as far as I'll go for general Machine Learning, however I'll cover more in-depth topics later, but if that seems at all interesting and you don't know where to start, there is a great book on the basics of Machine Learning by StatQuest which was recommended to me.

**Game Theory**

Following the general trend of exploring the intersection of thought processes and other fields, I've been learning more about game theory and general strategy (player vs. player) from a book titled "The Art of Strategy: A Game Theorists Guide..." by Dixit. In Microeconomics class I learned about Nash Equilibriums and business strategies, this book goes into more detail on those topics, and a lot more.

Just a quick intro for this blog however:

1. What is Strategic Behavior

Essentially, whenever you make a decision that has to account for the action of another person or for variables that change based on your decision, you're behaving (somewhat) strategically.

1. Why Strategic Behavior

One of the many examples throughout the book deals with the common scenario of someone trying to lose weight. Just about everyone knows the essentials of what they would need to do in order to lose weight; eat less sugar, drink only water, and move your legs around. Yet every year so many people fail to do just that, and the problem is that you are losing a strategic game... against yourself! The issue is that your current self wants to lose weight, but you in 30 minutes wants to do other things which would give you much more dopamine. Strategic behavior will help with this problem. A similar instance would also be wanting to keep a log or decision tracker, but failing because one's future self doesn't want to work 30 minutes of writing into their bed-time schedule... (see potential solution above :) )

**Python (for Data Science => Machine Learning)**

I've been programming for a while but, like most things that I do, I'm application focused to a fault. If I had to say why that is the case, it's because with my strangely wired brain I find the cost (including time) of learning something to be less than my cost of concretely remembering something. So, whenever I have some task or application of programming I want to do, I research the things that I'd need to know for the project, implement it, and move on. Now that I'm entering the stage of my life where I'm supposed to be specializing in certain topics this strategy is no good, and I need to start working towards more in depth knowledge on such things. In the realm of coding, this is pretty much going to be applied to allowing me to mess with data and build machine learning models. As such I'm currently reading a textbook on Python for Data Science, and when I'm done with that, I have another two lined up catered towards Machine Learning. As much as I dislike it, I'm already deriving utility from the textbook as I'm becoming quite familiar with python's newer syntax/structures. Coming from the Java's and C++, my python probably looked abhorrent, learning now about list/dictionary comprehensions, lambda functions, generators, and so on, my code is definitely a lot more concise. Not to mention I can now actually understand other people’s python code. Just writing a quick script to automate posting this blog on my website I utilized a list comprehension (which I'd just learned about earlier today), so the fact that it is already applicable is a great motivator. I won't go over why programming is awesome here, people can easily figure that out for themselves.

Recap:

Machine Learning is great, expect me to talk about further topics in the space

Game Theory is great, probably don't expect much from me on the topic however

Python is great, if I have something to share, I'll do so, and you can always look at my GitHub for projects (it's a barren wasteland at the moment however, but that should be changed once I get through these texts)