Hello girls and guys, welcome to an in-depth and practical machine learning course.
The objective of this course is to give you a wholistic understanding of machine learning, covering theory, application, and inner workings of supervised, unsupervised, and deep learning algorithms.

In this series, we'll be covering linear regression, K Nearest Neighbors, Support Vector Machines (SVM), flat clustering, hierarchical clustering, and neural networks.

For each major algorithm that we cover, we will discuss the high level intuitions of the algorithms and how they are logically meant to work. Next, we'll apply the algorithms in code using real world data sets along with a module, such as with Scikit-Learn. Finally, we'll be diving into the inner workings of each of the algorithms by recreating them in code, from scratch, ourselves, including all of the math involved. This should give you a complete understanding of exactly how the algorithms work, how they can be tweaked, what advantages are, and what their disadvantages are.

In order to follow along with the series, I suggest you have at the very least a basic understanding of Python. If you do not, I suggest you at least follow the Python 3 Basics tutorial until the module installation with pip tutorial. If you have a basic understanding of Python, and the willingness to learn/ask questions, you will be able to follow along here with no issues. Most of the machine learning algorithms are actually quite simple, since they need to be in order to scale to large datasets. Math involved is typically linear algebra, but I will do my best to still explain all of the math. If you are confused/lost/curious about anything, ask in the comments section on YouTube, the community here, or by emailing me. You will also need Scikit-Learn and Pandas installed, along with others that we'll grab along the way.

Machine learning was defined in 1959 by Arthur Samuel as the "field of study that gives computers the ability to learn without being explicitly programmed." This means imbuing knowledge to machines without hard-coding it. From what I have personally found, people outside the programming community mainly believe machine intelligence is hard-coded, completely unaware of the reality of the field. One of the largest challenges I had with machine learning was the abundance of material on the learning part. You can find formulas, charts, equations, and a bunch of theory on the topic of machine learning, but very little on the actual "machine" part, where you actually program the machine and run the algorithms on real data. This is mainly due to the history. In the 50s, machines were quite weak, and in very little supply, which remained very much the case for half a century. Machine Learning was relegated to being mainly theoretical and rarely actually employed. The Support Vector Machine (SVM), for example, was created by Vladimir Vapnik in the Soviet Union in 1963, but largely went unnoticed until the 90s when Vapnik was scooped out the Soviet Union to the United States by Bell Labs. The neural network was conceived in the 1940's, but computers at the time were nowhere near powerful enough to run them well, and have not been until the relatively recent times.

The "idea" of machine learning has come in and out of favor a few times through history, each time leaving people thinking it was merely a fad. It is really only very recently that we've been able to put much of machine learning to any decent test. Nowadays, you can spin up and rent a $100,000 GPU cluster for a few dollars an hour, the stuff of PhD student dreams just 10 years ago. Machine learning got another up tick in the mid 2000's and has been on the rise ever since, also benefitting in general from Moore's Law. Beyond this, there are ample resources out there to help you on your journey with machine learning, like this tutorial. You can just do a Google search on the topic and find more than enough information to keep you busy for a while.

This is so much so to the point where we now have modules and APIs at our disposal, and you can engage in machine learning very easily without almost any knowledge at all of how it works. With the defaults from Scikit-learn, you can get 90-95% accuracy on many tasks right out of the gate.

Machine learning is a lot like a car, you do not need to know much about how it works in order to get an incredible amount of utility from it. If you want to push the limits on performance and efficiency, however, you need to dig in under the hood, which is more how this course is geared. If you are just looking for a quick tutorial for employing machine learning on data, I already have a simple classification example tutorial and a simple clustering (unsupervised machine learning) example that you can check out.

Despite the apparent age and maturity of machine learning, I would say there's no better time than now to learn it, since you can actually use it. Machines are quite powerful, the one you are working on can probably do most of this series quickly. Data is also very plentiful lately.

The first topic we'll be covering is Regression, which is where we'll pick up in the next tutorial. Make sure you have Python 3 installed, along with Pandas and Scikit-Learn.