

# But What is Machine Learning? | An ML Guide for Dummies



Photo by [Kevin Ku](#) on [Unsplash](#)

I was sitting in an interview. It was related to machine learning in which I had no prior experience. I knew what ML is just like we say what a brain is. I was briefing on how neural networks have neurons (like the one in the brain) and how they are interconnected. Then we have layers, and we input data and get output. My speech was the literal definition of Abstraction in Programming.

Machine learning is a fascinating buzz word but what lies inside is nothing more than linear algebra and statistics (spoiler: a little calculus). But no need to worry as you will be working on a powerful calculator instead of a pen and paper.

(Note: References for machine learning resources are provided at the end of this post.)

For my luck, I was conditionally selected in the interview based on my skills in programming and app development. It gave me a chance to prove myself worthy to learn machine learning. But, more importantly, my time there made me realize how poor my understanding in this field was.

Going straight to the point, what is machine learning?

*Machine Learning is a method to find a relationship between different sets of values by utilizing statistical models and algorithms.*

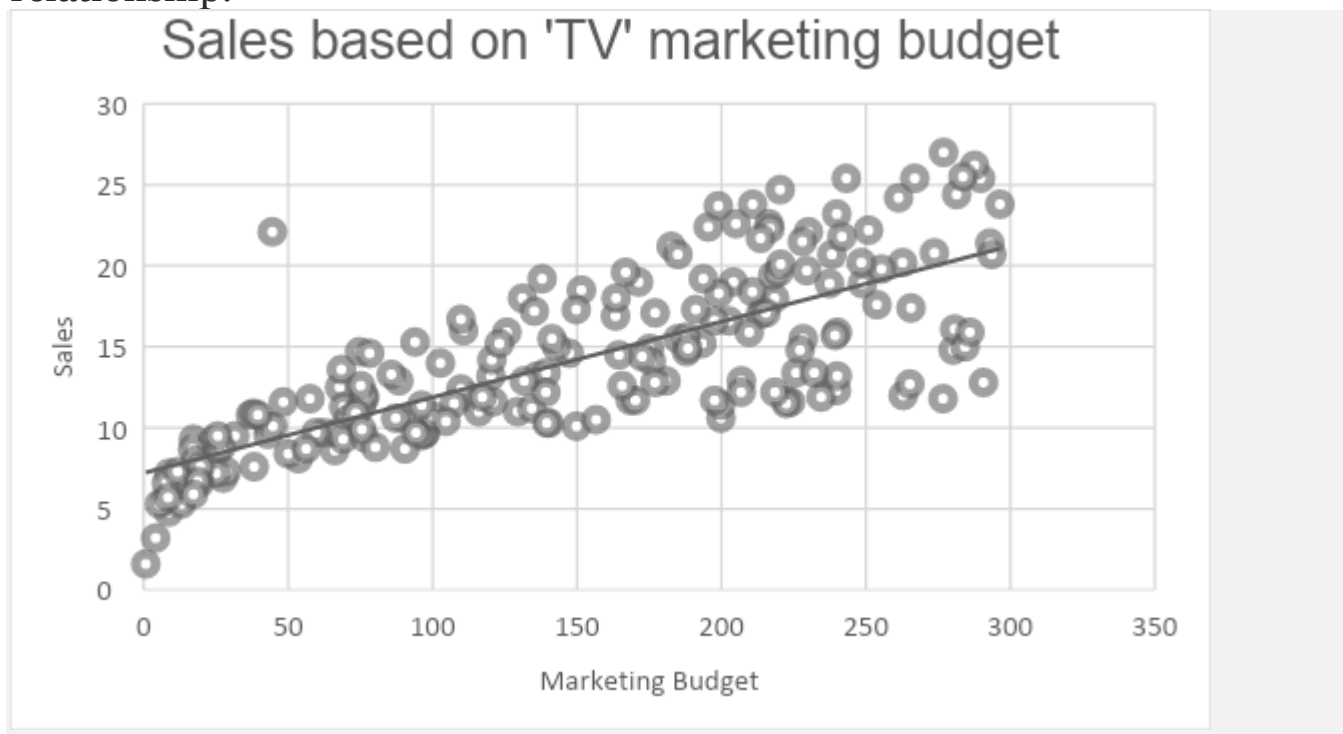
Let us look at an example:

Following is an example of a simple linear regression. We are trying to find a relationship between TV Marketing Budget and Sales of a company.

(Dataset is provided in the end.)

TV Marketing Budget	Sales
230.1	22.1
44.5	10.4
17.2	9.3
151.5	18.5
.....	.....

And now we will apply linear regression and obtain a straight-line relationship.



This straight-line can be represented as a linear equation  $\mathbf{y} = \mathbf{m}\mathbf{x} + \mathbf{c}$ .  $\mathbf{y}$  is the output sales, and  $\mathbf{x}$  is the input budget. By putting any value of  $\mathbf{x}$ , we can calculate the value of  $\mathbf{y}$ . But that is statistics, what is the role of machine learning? We need the value of  $\mathbf{m}$  (weight) and  $\mathbf{c}$  (bias) to calculate  $\mathbf{y}$  from the discrete values of input that can be plotted as follows.

Machine Learning is used to find the ' $m$ ' and the ' $c$ ' of this equation or any other equation that has unknown variables.

Now you might be thinking, "Hey, I didn't sign up for math, I watched tons of YouTube videos about crazy networks of neurons that predict the future." No worries, you can always follow the advice MJ gave to Peter Parker: "Expect disappointment and you won't be disappointed". Just kidding, you do not need to dig deep in math. You can start by using an online ML notebook like Google Colab and start from libraries like Keras. You can read (or watch videos) on:

1. Dataset
2. Layers
3. NumPy and Pandas libraries
4. Linear Regression
5. Weights and biases
6. Logistic Regression (Sigmoid and relu function)
7. Forward propagation
8. Backward propagation
9. Optimizers

Still, there are two ways to totally avoid math (but it limits your progress):

1. If you are a developer, you can use the already trained ML models in your applications. For example, “**MediaPipe**” can be used for hand detection, image segmentation, object detection and much more.
2. If you want to implement neural networks, you can exclusively learn the “Keras” library of Python from hundreds of simple neural network tutorials.

Along with linear regression, there are other equations used for machine learning. These equations are called activation functions. Different activation functions serve different purposes. Say you want to distinguish between a cat and a dog image, you need a “sigmoid” or “relu” equation. Unlike linear regression that can give a range of output, sigmoid is used for binary classification i.e., it outputs either 0 or 1 (dog or cat). The underlying goal is always the same: “Build a mathematical relation from an input data to classify, cluster or predict the output “. Note that I wrote “classify, cluster or predict” because machine learning has few types depending on the type of problem we are solving.

### **Following are the three types of Machine Learning:**

- **Supervised learning** — Here we have a large chunk of input data along with the expected. Algorithm takes the input data and applies activation functions with random weight and bias ( $m$  and  $c$ ) to predict an output. Then the predicted output is compared with the actual output and error value is calculated. Then we try to reduce that error by changing the values of weight and bias.

- **Unsupervised learning** — Here we do not know what the output should be. Huh, then what are we trying to achieve? We want to find a pattern in our input to group them together. The YouTube algorithm takes your browsing history and tries to find a pattern on what you watch, how long you watch, whether you like, dislike or comment to predict what videos should it recommend to you.

- **Reinforcement learning** — That's a straightforward carrot and stick model. It is used to solve problems that require decision making like playing chess or solving a maze. Imagine you have a rat, a maze, and a pack of cheese. You will place the rat in the beginning of the maze, let it wander and finally reward it when it reaches the cheese which is present at the end of the maze. If you repeat this frequently, the rat will learn the shortest path to cheese. In the same way, we can train an ML to model to solve the maze simulation. Here, neurons (activation functions with random weights and bias) act as mice. We deploy hundreds of neurons to wander around in the maze. A few of them might be able to find the path to the exit. Then we will stop the simulation and discard the neurons that underperformed. They will be replaced with neurons that reach the furthest of the maze. We can add random mutations to the batch to discover novel solutions. Then we repeat the process until all the neurons find the exit and we will choose the one that takes the shortest path.

Now, if you are still reading till this part of this post, then you are actually going to dive soon in machine learning. So, here is my advice for your journey:

- Do not start your ML journey from advanced concepts of language processing or image segmentation (or convolutional neural networks in my case).
- Follow a guide. Take the basic ML course like the one offered by Google or if you are non-traditional, you can ask an expert in ML to give you a road map.
- Python is the default language for ML beginners.
- Learn the tools used for ML. I use Google Colab, an online virtual environment. You can use offline applications like Jupyter Notebook but remember, ML is resource intensive, so you need a decent computer with an excellent GPU. In Colab, my code runs on Google server, so I do not have to worry about my potato pc getting fried.
- Enable hardware acceleration for GPU intensive tasks. It can be enabled both in Colab and Jupyter or any other ML environment.

Now that you have a rough estimate of what machine learning, you can start your journey from these courses, videos, and websites:

[Machine Learning | Google Developers](#)

[Free Courses in Data Science and Business Analytics — Analytics Vihdhya \(analyticsvidhya.com\)](#)

- Neural Networks from Scratch:

<https://youtube.com/playlist?list=PLQVvvaaoQuDcjD5BAw2DxE6OF2tius3V3>

[https://youtu.be/pauPCy\\_sOk](https://youtu.be/pauPCy_sOk)