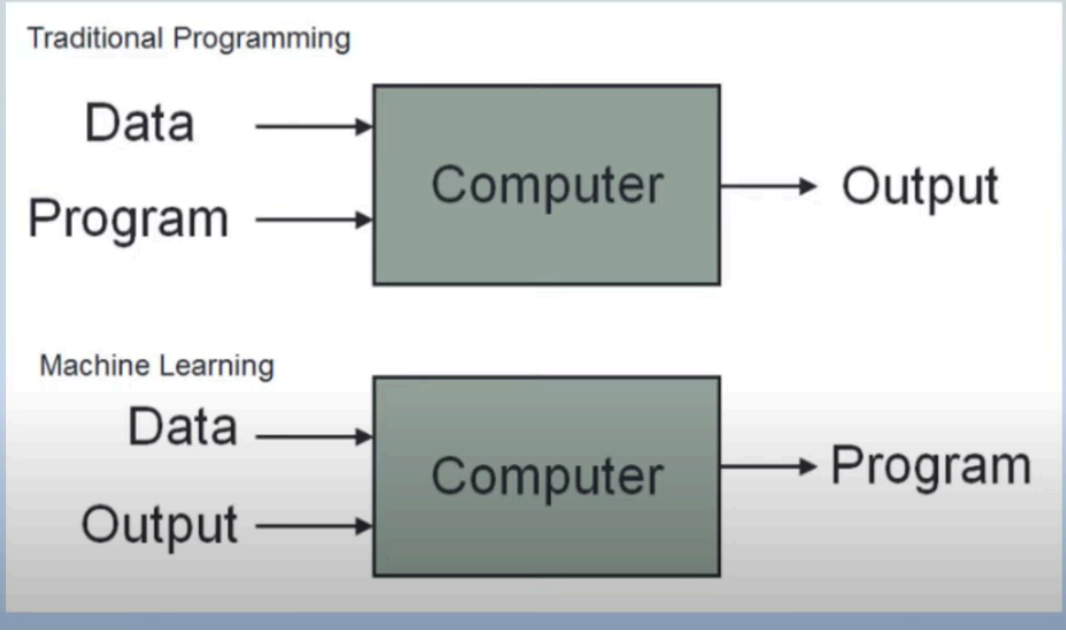
Machine Learning

Machine learning is a field of computer science that uses statistical techniques to give computer systems the ability to "learn" with data, without being explicitly programmed.

Explicit programming: Writing code for each specific scenario. To handle that scenario, you write a code. But in ML, you don't do that. What you do is that you got some data, and you have an algorithm. You instruct that algorithm to explore that data and identify patterns between inputs and outputs. Once you have identified the patterns, you provide new input to the algorithm to derive the output. If you check out this flow diagram, you can observe that, in conventional programming approaches, we write a program whose logic is written by us. If you give input to that program, you'll get your output.



But in ML, things are different, what you do is you provide some data. In that data, you give an input as well as an output. But you haven't written any program or logic. That logic is automatically generated by the ML algorithm. So the good part is you don't have to write code for each condition/case. It is handled automatically by the ML algorithm.
For example, you have written a code for adding two numbers. When you give two numbers to that program, the program returns you a sum.
But in ML, you will give an excel file in which the rows contain the numbers and their respective sum. Whenever ML models train on that data, the model realizes the pattern as addition. After training, irrespective of giving two or four or ten numbers as an input, your ML model knows that it has to perform addition. Whereas in the program, if you give more than two numbers, it doesn't work because it is explicitly coded for summing two numbers.

ML is more useful than traditional software development.

1. First scenario: You can't do specific things using programming. In this case, we use ML.
Consider that, you are trying to build an email-spam classifier to detect whether the given email is spam or not. A software engineer will pick a lot of emails, you'll have info whether the email is spam or not, then you will try to create patterns for them. Like any word such as 'discount' or 'sale' or 'awesome' is repeated more often or it is filled with a lot of pictures, then you create a long if-else ladder by using if-else for each and every condition. That will be a possible program for spam classifiers.

Somehow the advertisement company got to know that if code is written to classify the mail as spam if the word 'discount' is repeated more than three times then it will be a spam. Then what the advertisement company will do is instead of using 'discount' they can use 'price-cut' or something similar like that. Then your written program can't detect the mail as spam. Then you have to modify your code and logic again to handle this new scenario. If the advertisement company does the same things again and again, you have to change code frequently.

But in ML, it doesn't change. If the data changes, the logic will also be changed. If the data changes, then it'll automatically get reflected in logic. That's the beauty of ML. You have to write just one program, one algorithm and everything will be handled by that algorithm itself.

2. Second scenario: A scenario where you can't even imagine the number of cases, like 'image classification'. Let's say you want to classify dogs, if the dog is present in the picture or not. There will be hundreds of breeds which vary in looks, few large, few short, vary in colors and other characteristics. You have to create a program that detects the presence of a dog in a picture. Can you imagine the number of cases you've to write in order to cover characteristics of every breed? You cannot do so. In this case, you have to use that technique by which we, human beings, identify the dogs. We are taught that since childhood. Our brain mentally tags things in such a way. We learn things from the data. In this scenario, we can't use traditional software development approaches. You have to use the ML approach.

3. Third scenario(Data mining): Data analysis is a process where you extract patterns or search for hidden information by plotting graphs. But sometimes the information is more hidden which you won't be able to get through graphs.
Just by looking at an email content, maybe we are unable to detect key words for which the mail is detected as spam. Then you have to perform data mining. In data mining, you apply ML algorithms on the data and create a prediction model, such as email spam classifier. You can check the patterns, extracted by the ML model. If 'huge' comes 4 times, then it is labeled as spam. If not then it's not spam. After applying the ML algorithm, if you are able to extract important/hidden data from the information, then it is known as data mining.

History: All the theories and math have existed for a very long time. But ML was not that famous. Because ML requires a lot of significant data. Unfortunately in ancient ages, data gathering and labeling were quite hard tasks. Second problem was the hardware. The hardwares was inefficient to

run algorithms on such a large amount of data back then. After 2010, the evolution of the internet and smartphone, these problems were sorted out. We are generating data at a heavy pace nowadays. The amount of digital data created from the start of mankind till 2015, is generated in 2016 alone. That is the speed at which we are generating data. This data is aiding the growth of machine learning.

Second thing, hardware. In the modern day world, we ourselves are carrying up to 12GB of RAM in our pocket in the form of a mobile. We are carrying GPU's in our pockets, which were not available to ancient research scientists. Even 128 MB RAM was a big deal back then. But in the modern age, we are equipped with good hardware, algorithms and data for which ML has become successful. And to be honest, this is not going to stop anytime soon. The growth curve will keep on growing exponentially.

When JAVA entered the market, only a few people were familiar with the language. But the companies needed JAVA because their competitors were implementing it. But the JAVA developers were few in number. So the companies were fighting to hire those few available professionals. For this reason, the salary of JAVA developers was high.

Now a similar trend is going on with ML. Many of the engineers don't know ML. Over time, if the salary of ML engineers increases, then more people will become interested in learning ML to secure those jobs. So in the coming few years, the majority of people will know ML. Then the salaries will be normalized. That's why the companies won't pay high salaries.