TRANSCRIPT

Introduction

Machine learning is a phenomenon that has taken the world by storm. Touted as the must-have skill of the twenty first century, there are as many sources of knowledge as there are learners, if not more. Depending on your education and discipline, machine learning can mean a variety of things. For some, it is an oracle, which predicts the future whereas; to a few, it just implies coding away on a computer screen. A very popular opinion prevails that machine learning is a field with a steep learning curve; requiring high-level mathematical and coding expertise to understand and thrive in. The truth is somewhere in between all the hype and fluff. With this course, we intend to demystify machine learning and help you, the participant to get a hold of the basics by providing perspective and use-cases relevant to our business context. Completing this course will help you understand what machine learning is capable of doing and importantly, what it is NOT capable of doing. Machine learning or ML being an extremely dynamic field, where; today's state-of the art concept may be rendered obsolete tomorrow; this course cannot possibly be exhaustive and cover all the concepts under the sun. Thus, we have taken the core concepts and fundamentals, which always are the foundations of any new development in ML and help you understand and use ML in your field of work. Learning these fundamental concepts, those motivated to dwell on advanced topics will be able to do so with relative ease. Setting these expectations and the intent, let us formally begin our journey of understanding machine learning.

What is Machine Learning?

So, what exactly IS machine learning behind all the hype and frenzy? A field of science and a powerful technology that allows machines to learn from data and self-improve is the simplest way of describing machine learning. Here, by machines, we mean programs or code written and executed on computers and by learning, we imply that the machines are modelling the process based on input data without being explicitly programmed to do so. Essentially, Machine learning is a series of mathematical functions trying to model a process by understanding and transforming the existing data pertaining to the process. Though manually possible, computers are used in order to do number crunching on huge amount of data and do complex math, creating intricate models. By modelling, we mean that the computer program tries to replicate the behaviour of the process by mathematically linking the input of the process to its corresponding output, based on the existing data. Simply put, the model is 'learning' the math linking the available input to the output. Basically, the model is 'training' on the data to learn the math. Input data pertaining to a process are pairs of input conditions/parameters and the output they produce after interactions in the process.

In the case of modelling a part machining on CNC process, the input data can be pairs of the linear and rotational speed along the axes and the quality of the finished component for each job and the model is trained on these data pairs. Now, when a new input is provided to the ‘trained’ model, the output generated by the model is expected to be quite similar to the output of the actual process, when fed the same new input. Simply put, the model is now capable of ‘predicting’ the output of the process, given a new input. In our CNC model context, the model is now trained to classify a faulty component from a well-made one.

Why math for Machine Learning?

Knowing the logical steps in a concept is the best way to understand it and in case of machine learning, the logical steps are the mathematical functions of the model. Having an understanding of the underlying math helps create a model that mimics the process of interest in the best way possible. Learning happens from mistakes and the models are no different. A suitable model performance criterion such as accuracy or mean of errors helps the model learn from its mistakes the most. Mistake for a model is the difference between the prediction of the model & the actual process output. We would like to minimize the mistakes to greatest extent possible. Knowledge of underlying math helps us choose the most suited performance metric, making the model replicate the process to closest extent possible. Importantly, knowing the math behind a model’s working gives a context to the model’s prediction, greatly increasing its interpretability, confidence on the model and correction of wrong results. With the importance of familiarizing with the math in order to understand machine learning firmly established, let us list down the broad math faculties involved.

What math for Machine Learning?

The math required for understanding the machine learning models can be categorized into five main faculties. Each of these are listed on this slide along with the aspect of machine learning they are most relevant to. A ready reckoner of all the basic math concepts and formulae are attached as resources with this slide. We shall not dwell on the math right now and explain a concept as and when we require it to understand a particular machine learning model. With this, we now proceed with familiarizing ourselves with the basics and terminologies of machine learning.

Let’s get started with the basics

Let us start with an example many of us can relate to. Consider the process of manufacturing a job on a C N C machine and we need a machine learning model to predict quality of the finished part with least amount of additional instruments and human intervention. The input data to this model can be the tool station and chuck motion data logs, which record the linear and rotational speed and vibration data along with the records of the quality acceptance status of the corresponding part. Upon training over a sizeable amount of such job records, our model can learn the mathematical relation between the tool station and chuck motions with the quality of the final job. With this, the model is now capable of predicting the quality for new jobs based on the machine parameters. Now, many of you might think that this is too simple a job to deploy a machine learning model on or perhaps, the prediction can simply be done by defining allowable values for the machine parameters without all the modelling stuff. To address these doubts, we would like to highlight the potential of machine learning in this case. A trained m l model can not only predict the finished job quality but also estimate the time taken for each job. In case of expediting a job to account for exigencies, the model can provide optimized parameters for C N C machine tooling ensuring the desired quality level compliance. This level of control also can enable the model to estimate the remaining useful life for the tool, impact or affordability of expediting a job process, helping the shop floor team plan the job schedule and maintenance cycle; better and smarter. Some of these discrete features may be possible by human intuition or experience and some by simply using spreadsheets with formulae. Human experience is not uniform and spreadsheets need exact mathematical relationships to be known beforehand. M L models have no such prerequisites and that's its real utility. With this example, we hope you can get a context of the possibilities of m l models to our traditional processes as well as new potential applications. Hopefully, having kindled an interest in machine learning; we shall now introduce the basic m l jargon before learning about specific M L models.

Understanding the ML jargon

Let’s start with the simplest of machine learning jargon. We start with the features or parameters of the model. Features are the unique attributes of inputs fed to the model. In our C N C machining example, the linear and rotational speed along the axes are the features for the model. Each job on the machine for which the features are recorded is known as a data point. The model predicts an output for each data point it is fed. The features and output are also often referred to as variable and to distinguish between them, features are termed independent variables and the output is appropriately called dependent variable as it depends on the mathematical interactions among the independent variables. Focusing on data point J1, its four features can be visualized on the three axes giving a sense of spatial magnitudes.

Each feature is represented as a vector in the three dimensional space. We conventionally know vectors as a mathematical entity having magnitude and direction, and a scalar is an entity possessing only magnitude and no direction. In machine learning domain, vectors are used to represent the features, of a data point in a mathematical and easily analyzable way. This is an important concept which makes future mathematical modelling simpler. Vector operations such as dot product or cross product can be visualized geometrically. The vector operations may also be the math operators that the model in our CNC example arrived at, to predict the job quality. The vector representation for the data point J1 in three-dimensional space as well as the M L representation are presented. Three-dimensional representation gives us a visual sense and is easy to interpret. However, as we move towards modelling complex systems, the number of features increase substantially and visualizing them becomes incomprehensible as we are conditioned to operate in a three dimensional space. We need more axes to represent the features and three-dimensional space is no longer sufficient. Space with more than three dimensions is called hyperspace. Each basic element we are familiar with such as plane, cube, sphere; each have a corresponding entity in hyperspace such as hyperplane, hyper cube and hyperspheres. While the three dimensional sphere or cube can be visualized by us as well as mathematically expressed, the fourth, fifth or higher dimensional geometries cannot be intuitively visualized and we need to depend on math to represent or interact with it. Data points in M L often have high dimensionality and for such cases, vectorial representation is easier to understand as well as mathematically manipulate. Representing each data point as a column vector is the common convention. Thus, representing all the data points together gives us a rectangular list with the features as rows and the data points as the columns. This rectangular list is mathematically known as a matrix. Once, a dataset is represented as a matrix, we can use linear algebra to manipulate it and model the processes. A key takeaway of this slide is the matrix representation of data points which shall be used at later sessions.

Understanding the ML jargon

Now, let's simplify our C N C modelling example to consider only two features or independent variables for easy visualization. On the two dimensional graph of independent variables, the jobs are marked and their quality status is color-coded. Our model here is a line which is almost perfectly separating the defective jobs from the well-made ones. This line is the mathematical function. You can see the general equation of the line and its form specific to our case.
The terms a, b and c are coefficients for x one, x two and unity respectively. In machine learning terminology, these coefficients are often referred to as weights. During training, the model is basically finding the best possible combination of these weights to correctly classify maximum number of jobs. As this was a two feature model, we can easily visualize the line as well as the data points and also write the equation for the model trivially. We need to extend this terminology generalizing for more features and models where the math may not be as simple as a line.

Understanding the ML jargon

Here on the screen, we see the generalized form of the model equation. Though it does not resemble the simple equation of the line by a long shot, they are same when we see each component. Recalling the matrix notation for our data points, we have the feature or independent variable matrix. The features are often denoted as X'es. We have considered 'i' features and 'j' data points to express the generalization. To the right of the feature matrix is the matrix of weights or coefficients which is denoted as w for ease of understanding. It is these weight values that the model is actually computing when we say machine is learning. This weight matrix is the set of coefficients which we had seen in the equation of the line. Note that there is one weight term corresponding to each feature in the feature matrix. We have matrix multiplication between the features and weights matrix. In order to have dimensional compatibility, the Feature matrix is transposed as is denoted by the letter t. Thus, each data point which was initially a column vector of dimension j cross 1 now is transposed to a row vector of dimension 1 cross j. F is a mathematical operator which transforms the resultant product of matrix multiplication to the vector of output or dependent variables, which is commonly denoted as y. The vector Y is now a column vector of dimensions i cross 1, with each row corresponding to output for each of the data points. Upon expanding the matrix, we get the familiar equation of line, but with all the features instead of two seen earlier. Our simple model's line is written in this form at the bottom for two data points, for context and relatability. Note that the feature matrix is already written in transposed form for ease. In this case, the mathematical function F is the identity function which returns the input only as the output.