Module 2

Video Transcripts

Video 1: Introduction to module two

Welcome to module two, Introduction to machine learning. In this module, we will start with two very fundamental, yet important questions. We will discuss what is machine learning? We'll see that a lot of machine learning is about learning functions that map inputs to outputs from data. And we will discuss why we would want to use machine learning? We'll see that two major use cases are forecasting and inference. After that, I will provide an overview of the machine learning landscape, and I will discuss how machine learning differs from statistics?

And I will bring up some of the major dividing lines of machine learning, such as prediction problems versus classification problems, supervised versus unsupervised learning, as well as, parametric versus non-parametric methods. In the last part of this module, I will consider some of the practical aspects of machine learning. We will walk through the ten steps that a typical machine learning project consists of, and we'll discuss some of the real-life examples of machine learning.

Video 2: What is machine learning?

Let's start with a fundamental question, what is machine learning actually about? It turns out that a large part of machine learning is about a seemingly very simple problem. It is about understanding the relationship between a number of input variables, which are denoted here by 'X_1' up to 'X_P' and an output variable which is denoted by Y here. Now learning this relationship between these input variables and the output variable turns out to be rather tricky for two reasons. The first one is we only have limited data points that allow us to learn this relationship between the input variables and output variables from.

So, if we describe the relationship between the input variables and the output variables as a function 'f' that we would like to learn, there is limited data available to learn 'f' from. The second complication is the presence of noise. Even if we had as many data points as we wanted to, we will never be able to learn the function 'f' with absolute certainty because the connection between the input variables 'X_1' to 'X_P' and Y is the stochastic one. It is affected by noise, which could be other input variables that we have left out in our model, that we haven't measured for example.

Or it could be that the underlying true relationship between the input variables and the output variable is an inherently stochastic one. By the way, the input variables come in many different names in machine learning. We sometimes call them also independent variables or predictors, features or fields. Likewise, for the output variable, we sometimes use the term dependent variable or response variable, target variable, or outcome variable. Now let's explore why we may want to learn the relationship between input variables and the output variable in practice.

It turns out that there are two major reasons for this: one is to do forecasting; the other one is to do inference. Let's look at these two reasons in turn. In forecasting, we want to be able to predict the value of the outcome variable 'Y' from the input variables from the values of the input variables 'X_1' to 'X_P' on data that we have not yet seen.

On data where we have only seen the input variables, but not the output variable. Let's have a look at a couple of examples. Imagine for example you want to verify or determine whether a patient has cancer or not. In that case, you may get a blood sample.

The blood sample has many different characteristics which could be captured by input variables 'X_1' to 'X_P'. The output variable would be a binary variable which tells us whether the patient has a certain type of cancer or not.

Or it could be what we call later on a categorical variable which tells us which type of cancer the patient has, or hopefully none at all. A different example would be to detect fraudulent transactions from let's say tax claims or expense claims. In that case, the input variables would be certain properties of the let's say the expense claim, the output variable would be a binary variable which says this expense claim is fraudulent or not.

There are many other applications like that. The key to forecasting is that the sole the sole motivation for us is to predict with high accuracy what the value of the output variable is for given variables of the input variables. The function 'f' that we estimate could be arbitrarily complex here. In practice, for example, we use neural networks in many cases. These are predictors, we will see them later on. These are predictors that work often extremely well, but they lead to extremely complicated functional relationships 'f'. In forecasting we don't mind; all we care about is prediction accuracy.

The other class of problems that often arise in machine learning are so called inference problems. In inference problems, we actually do care about our explanation for the functional relationship 'f'. We don't want our estimation for this functional relationship 'f' which we often call f-hand. We don't want that explanation to be too complicated because we want to actually understand what the relationship is between input variables and output variables. Let's have a look at some examples. Imagine, for example, we want to predict the sales of different types of marketing campaigns depending on the medium that we use.

For example, it could be a TV campaign or newspaper campaign or radio campaign. In that case, we're not just interested in predicting what the increase in sales is. We want to understand how the increase in sales can be attributed to different campaign types because we want to then take a decision: what is our optimal marketing mix? Another example would be the prediction of house prices. In many cases, we do not just want to predict the price of a house based on properties of a house, such as is it a Victorian bill? Does it have a river view or things like that? But we also want to understand, what is the contribution of the river view to the house price? Those are influenced problems where we do not just care about prediction accuracy, but we also care about a simple model that can be interpreted by a human decision-maker.

Video 3: Prediction vs classification

Let's discuss now some of the major dividing lines of the machine learning landscape. We're going to discuss free pairs of terms that will help us to categorise machine learning algorithms that will help us later on to understand what an algorithm is good for and what the tradeoffs are in the design of the algorithm. The first dividing line is going to be prediction vs classification. Prediction problems have an output variable that is a number, that is continuous. Prediction problems are also called regression or estimation problems.

Let's have a look at some examples. You could for example, wonder what is the value of a house price? In that case, you would estimate the house price from various input variables. These could be categorical. There could be things like yes, no. For example, Is it a Victorian house or not? They could be numeric, the input variables. It could be the square footage of the house. But the important thing is if we have a prediction problem, the output variable that we want to estimate in this case the house price, is the number. In the classification problem on the other hand, the output variable is not a number. It is a category.

We also say the output variable is categorical. An example would be a spam filter. We could categorise emails into spam or non-spam emails which are also called ham emails depending on various properties of the emails. For example, words that appear in the email. A different example would be a patient that is tested for cancer. In that case, the outcome variable would perhaps not be yes, no. But it would be yes, the patient has a cancer of type A.

Yes, the patient has a cancer of type B. Or no, the patient does not have cancer. We often distinguish between ordinal and nominal categorical variables. A nominal categorical variable does not have any natural ordering. Think about cancer A vs cancer B vs no cancer. An ordinal categorical outcome variable on the other hand, has a natural ordering, such as high, medium, low.

Video 4: Parametric vs non-parametric

The next distinction is a slightly more technical one. You will probably struggle with that distinction in the beginning, but I hope as we go along in the course, and we see several different methods for machine learning, that distinction will become clearer. The distinction is between parametric and non-parametric approaches to machine learning. Parametric approaches make an assumption about the function 'f' that we want to estimate.

For example, they may make the assumption that the function 'f', which describes the relationship between the input variables and the output variable is a linear function. Such an assumption gives us a lot of power to build strong machine learning approaches because suddenly we only need to learn the slopes of the linear function. Once we have made the choice that the unknown true function that we would like to estimate is in fact linear. Note that of course, the true function may or may not be linear.

On the other hand, we have so called non-parametric approaches. In non-parametric approaches to machine learning, we do not make any assumption or any strong assumption I should say, about the shape of the function 'f'. Rather, these approaches are in principle able to learn any shape for the function 'f'. This gives us a lot of power. We don't need to make any assumptions about the true underlying relationship between the input variables and the output variables. But this power comes at a hefty price. And that is that non-parametric approaches typically need much more data to learn a reliable functional relationship between input variables and output variable.

Video 5: Supervised and unsupervised

The third and final distinction that I want to make is between supervised and unsupervised learning. In fact, everything we've discussed so far, relates to supervised learning. In supervised learning, we have a set of input variables 'X_1' to 'X_P', as well as an output variable Y. And we want to study the functional relationship between these input variables and the output variable, based on a training dataset.

In unsupervised learning, we do not have an output variable Y. Rather, our training data comes only as a set of records, where each record contains the values for all the input variables.

So, what we want to do in unsupervised learning then? Well, let's have a look at an example. Consider the customer database of a retailer. The retailer may have many different customers and each customer can be described through his or her previous purchases. Those could be the values of the input variables. A natural question then is, does your customer dataset naturally decompose into various groups of customers, into customer clusters, if you wish.

That is an example of unsupervised learning. Our customers do not come with labels. No customer comes to your company and says, "I'm a big customer" or "I'm a regular customer." Rather, based on the values of the input variables for each of the customers, we want to cluster the customers into groups. So, we have data that does not come with an output variable but still we want to make sense of, how this data decomposes structurally into different groups.

Video 6: The machine learning process (part 1)

Let's now look at the workflow of a typical machine learning project. It turns out that in an idealized form, a machine learning project can be decomposed into 10 different steps. In step one, we want to define what the purpose of the machine learning project is. In particular, you want to figure out with your client or with the sponsor within your company, whether the project is going to be a one-off effort or an ongoing procedure. This may have implications of the users that are actually applying your method or are going to interpret the results of your method.

If it is a one-off effort, it could well be that your method is going to be applied only by a handful of experts. Whereas if it is an ongoing procedure, it could be that there are many different end users with different requirements on your software. Closely related, you want to figure out what the results will be used for eventually. In the second step, you need to pull together the data for your analysis. The data can come from internal sources such as customer databases or purchasing databases, as well as from external sources such as for example, credit rating databases.

In the third step, you want to explore, clean, and preprocess the data. In particular and we will discuss this later on, you may want to scale some of the fields in your data in order to for it to be better handled by machine learning approaches. And you want to deal with the issue of missing data and outliers. Broadly speaking, there are three different ways in which we can deal with missing data and outliers. The first way is to simply remove the affected records. The second way is to manually fill in the data, which means that you have some domain experts that are looking at the data fields that are missing or data fields that are obvious outliers and try to replace these values with the correct values.

And the third approach is to do the same, but through an algorithm. For example, replace missing data values with average values across the other fields. Let's have a look at some of the intricacies that can arise when you deal with missing data and outliers.

Video 7: The machine learning process (part 2)

The fourth step of our machine learning workflow is closely related to the third one. We want to remove variables that are either not going to be available at a point in time at which we want to employ our machine learning technique to actually forecast or infer data, as well as we want to eliminate variables, input variables that are simply not relevant for our analysis, that are not correlated with the output variable. We may want to transform some of the input variables that are categorical into numeric values or the other way around.

This is closely related to a later step where we choose the appropriate machine learning methods and finally, we even may want to construct new features based on domain knowledge. We'll see that later on in this class. In the fifth step, we finally define the machine learning task at hand. In particular, we decide are we dealing with a classification task, a prediction task or perhaps even an unsupervised learning task such as a clustering task. In step six, we partition the data. If we are dealing with supervised machine learning problem into a training dataset, a validation dataset, and a test dataset. We will see later on what these different datasets mean. In step seven, we choose the appropriate machine learning technique for our task at hand.

This could be a single technique, or it could be multiple techniques. You will see later on in this class, we're going to explore many different techniques such as regression and classification trees, clustering techniques, K-nearest neighbor techniques and so on. In step eight, we then use the various machine learning techniques for the task at hand and we can then have a look at the results in step nine. We need to interpret the results. We need to compare our algorithms against each other as well as against simple benchmarking strategies.

And then finally in step 10, we will deploy the algorithm that we have chosen. Please keep in mind that in a real-life machine learning project, these steps do not appear sequentially like this. There will be many loops between the steps. Sometimes you may need to go back to a previous step because you realise that a later choice has an impact on an earlier one. But broadly speaking, these 10 steps are going to be present in most machine learning projects.

Video 8: Case study: predicting donor likelihood

Let's have a look at a real-life machine learning application that I've been involved in together with a group of students. This is the case of a major international charity that has asked us to look at one of their countries which they are piloting analytics schemes in, and in this pilot country there are about one million active donors, where a donor is called active if he or she has made a donation within the last three years.

Now this charity runs about one campaign a month and these campaigns repeat on an annual basis. For each campaign, the charity sends out a letter to all these active donors and it costs for each letter range between 75 pence, if it's just a letter and five pounds if there are other things included, like a calendar. The success rate, which is defined as the percentage of the donors contacted that actually afterwards donate to a particular cause depends on the campaign, but it typically ranges between five and ten percent. Now any successful contact leads to a donation and the donation ranges in the order of 40 pounds on average. Wait. Let's try to understand this a bit better.

Let's make this concrete. Let's assume each letter costs 1 pound and at the success rate is 7.5 per cent. In that case, the charity would spend 1 million pounds on sending out those letters. And they would raise 7.5 per cent times these one million letters, times 40 pounds in revenues, which would lead to 3 million pounds raised gross. In other words, 2 million pounds raised net for the campaign after these costs for sending out the letters. Now imagine we could avoid contacting 10 percent of the active donors that will definitely not donate to a particular campaign.

This would mean that we would decrease the cost by 100,000, which would mean a raise in the net money raised by five percent. So, this is substantial. Now the student group and I, what we have done is we have implemented decision trees to predict which of the active donors would donate for any particular campaign. Criteria that we used to contain amongst others, the demography of the donor, such as the age, the gender, whether the donor lives in a big city or in a rural area, as well as the donation history of the donor for the same campaign in previous years, as well as for other campaigns, both in previous years and in the current year.

This is important because donors typically tend to donate only for some of the campaigns and most donors have their favorite causes. And it turns out by using these decision trees, we can help the charity to save many tens of thousands of pounds each month. And the great thing about this is the decision trees are simple to implement. You will be able to do that easily yourself after completing this programme.

Video 9: Summarising module 2

This brings us to the end of module two, introduction to machine learning. In this module, we have discussed what machine learning is about, and what it is used for. We have discussed how machine learning differs from statistics. And, we have explored in first overview of the machine learning space. We have discussed the differences between prediction and classification problems, between supervised and unsupervised learning, between parametric and non-parametric methods, and we have seen an overview of the method landscape. We have also discussed the practical aspects of machine learning, such as the ten steps typically involved in completing a machine learning project, as well as some of the real-life examples of machine learning.