Learning Analytics Tools

Professor Ramkumar Rajendran

Department of Educational Technology

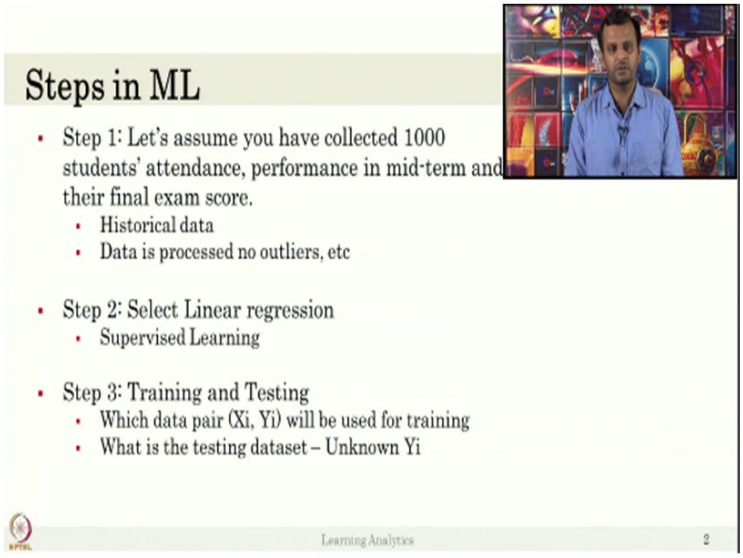
Indian Institute of Technology, Bombay

Lecture 3.3

Training and Testing Data

In this learning dialogue, we will talk about Training and Testing Data. In the last video, we saw there are 3 steps in machine learning.

(Refer Slide Time: 00:27)



The first step is data collection and selecting the algorithm is second step and training and testing is the third step. We talked about data collection last week and also we saw an introduction to what is machine learning algorithms in the previous video. We will talk about what is training and testing data in this video. Step 1, let us assume you have collected 1000 students attendance, performance in a midterm and also the final exam scores.

Since I mentioned 1000 students you may not have 1000 students in the current class so you have historical data say students across 10 departments or 5 departments or their scores across multiple courses something like that. So you have 1000 students attendance performance in midterm and also the final score. It can be that the same student's data from the different courses or it is also possible that there are 1000 students.

And you check the data and you pre-process the data to identify outliers, errors, and have ensured that everything is checked. So you have a data of 1000 students with all these two input variables i.e. attendance and performance in midterm and also the y is the final exam score. We want to create a model to predict the student's final exam score. Step 2 is selecting the algorithm let us say for this example you selected a or linear regression algorithm with two variables.

Step 3 is training is testing. The question is which data pair will be used for training? You want to create a linear regression model that linear regression model if you remember, is

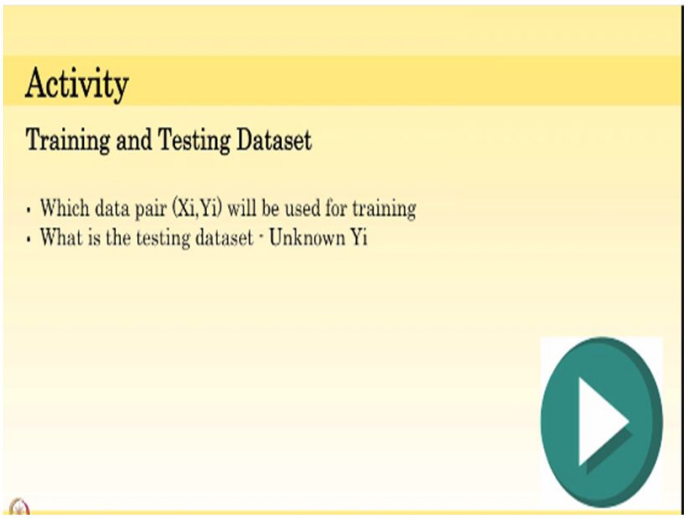
$$y = mx + c$$

Which data you will use to create that $y = mx + c$

to train the weights and m value and c.

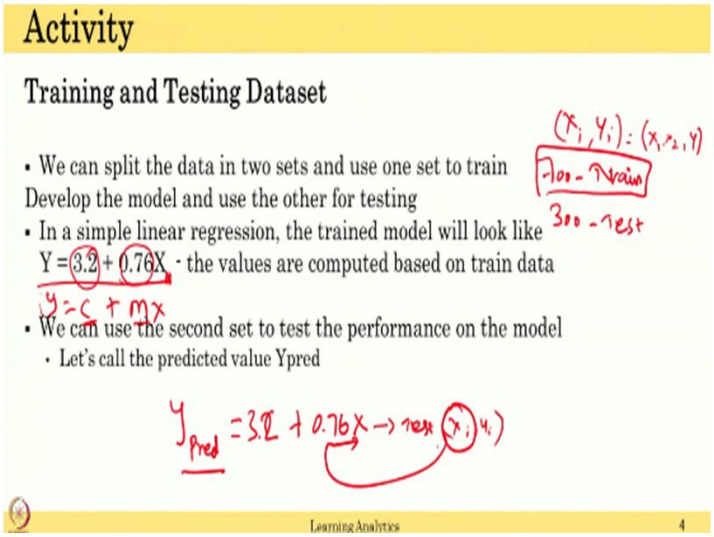
So, what the testing dataset means is students data without their final score. So what is the testing dataset you want to use it? You know all the y because I said that it is 1000 students data you already have the final score also, what is the testing dataset?

(Refer Slide Time: 02:46)



Let us think about that last two points as a question. Think what data pair will be used for training data and what will be used as testing dataset. List down your answers, after listing it down resume the video to continue.

(Refer Slide Time: 03:03)



So, in ML, we usually split the data into two sets and use one for training and one set for testing. For example, you have 1000 students data, so let us consider you have used 700 student's data for training and 300 for testing. So you are given the same data we will split the data into training and testing and use it.

It is very general you split the data into two, but recently not recently say 5 years before we started using a validation set also, train set, validation set and test set, but let us not go into that. Let us see keep it simple we have only two sets of data that is training and testing. So, that is a train data of 700 students historical data with x_i and y_i information.

So 700 pairs of x_i and y_i, x_i can be $x_{i,1}$ and $x_{i,2}$

So this pair is basically $(x_{i,1}, x_{i,2})$ and y_i, 700 samples like this. In simple linear regression, the trained model will look like this I consider only the simple linear regression which means I have only $x_{i,1}$ no $x_{i,2}, x_{i,3}$ so let us not keep $x_{i,2}$ only $x_{i,1} = x_i$.

So, this looks like $y = mx + c$

the slope equation we studied in class 8 or something. So, the values of this 0.76 or the 3.2 are computed based on training data. Given 700 training data, the algorithm computes the value of this $y = mx + c$.

The c and m can be computed from the training data or the 700 data. Now we used this model to predict the performance of the test data.

While testing data, we will use $c = 3.2$ and $m = 0.76$.

Although the test data set is having x_i and y_i but I will use only x_i and apply

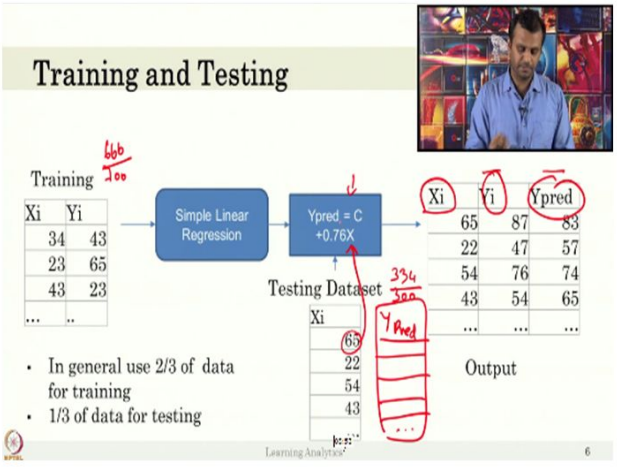
$y_{predict} = 0.76x + 3.2$ and I will get $y_{predict}$. So, what I am saying is just split the data into two sets i.e. training and testing. Use the training data set to create the model then you apply the test data set input variable to predict $y_{predict}$ let us discuss in detail.

(Refer Slide Time: 06:04)

Suppose, this is an equation of the model we talked in the last class, so

$y_{predict} = 0.56x + 31.96$, this is not the same example I have just given. Consider you have a training data of 700 points.

(Refer Slide Time: 06:20)



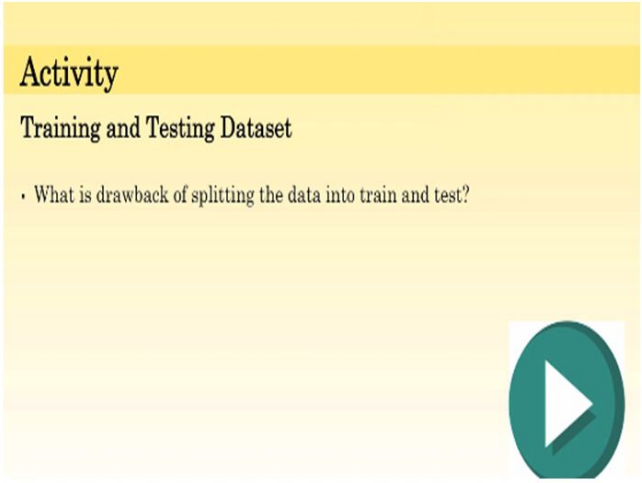
Consider you have a training data of 700 points here and you have a testing data that is a 300 data value here. So training data is x_i and y_i and you have used this data along with a linear regression model to come up with this equation. So how it works we will discuss that in detail later, but for now, consider it is a black box and you have given this input to linear regression model and we got this particular equation.

c can be some value 0.76 some value I have just given an example it is not the actual score to fit this value because I do not have 700 data I just have 3 data here to show. Now you want to test the performance of this particular linear regression model on a test data set. Now you have like a 300 test that is 300 values of x_i. Apply this x_i to this at each equation.

So you will have $y_{predict}$ for each of this score like for each x_i you will have a corresponding $y_{predict}$. So, in general, we use two-third of data for the training, one-third of data for the testing that is if there are 1000 data points 666 data for training and 334 data for testing. So, the output for testing data will be $y_{predict}$ after applying x_i the value here.

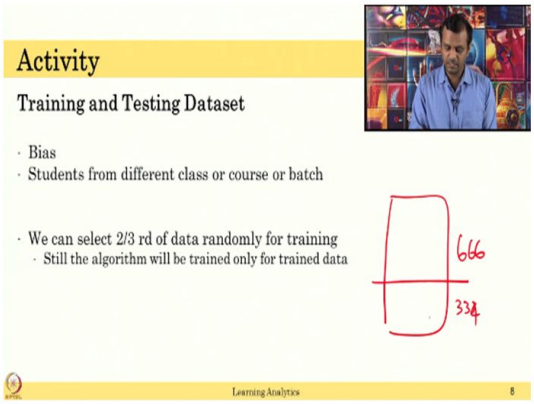
Also, you know what is y_i because you already collected a supervised learning algorithm so you have y_i. So, this is the predicted value, this is the actual value in a test dataset. Now to compare these two you might able to validate the performance of this system we will discuss that in detail in next lecture, but just want to tell you that you will get $y_{predict}$ here and comparing $y_{predict}$ and y_i, you get the validation of the performance of this particular algorithm.

(Refer Slide Time: 08:56)



So, you saw that we can split data into train and test. Do you think this is a drawback in splitting data into training and test just two-third and one-third? If you think what is a drawback? List down your answers and resume the video to continue.

(Refer Slide Time: 09:18)

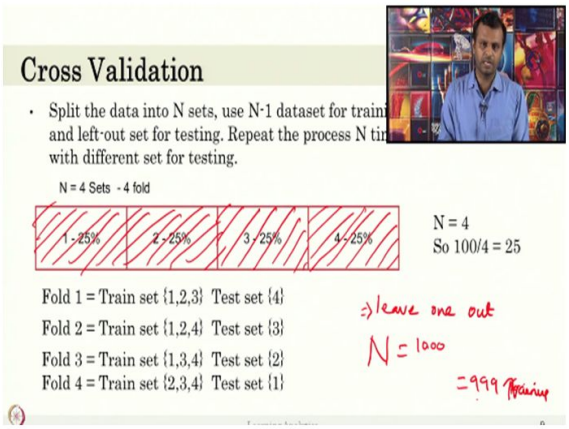


So that is a bias that is the main problem in the particular data I will still explain what is that. You classify the data into simple 666 and 333. How do you arrange the data? You might arrange the data based on the students in one class set for class 1, class 2, class 3 or based on the year they did the course in a subject or in the university or in the college.

So, when you split the data there is a chance the test data sets not belong or may not be even be related to the training data set. So splitting a data as simply 666 and 333 will lead to bias because the student's behaviour will be totally different in training and testing set. Also, the data you selected of the same students in different courses is valid, but the data is from different batches, from the different department will have different features. It is not just attendance only impacts the student's performance. Attendance plus some other external factors like a teacher, teaching material or the course, difficulty level, how many senior batches the students had all this information is needed. So, when we classify the data simply splitting into training and tested by random numbers say 666 or 333 it will not help, it will have a bias.

To avoid it, what we can do we can select the two-third of data randomly from training. Instead of selecting the first 666 data as the training and the last 334 data as the testing. You can randomly pick 666 data for training and the rest is for data testing that is the one way to do that. Still, the algorithm will be trained only for particular training data. The performance of the algorithm is tested only on a particular test data set you have and it has trained only for particular training data so how to avoid it?

(Refer Slide Time: 11:29)



So to avoid that error we will use cross-validation. In cross-validation, we split the data set into N sets. Use the N minus 1 data set for training and remaining set that is the set which you have not used for training is used for testing. Repeat the same process N time with a different test set for each time. So, to do that in detail let us see one example. Consider I want N equal to 4 which means 4-fold cross-validation because it is 4-fold cross-validation because you have to do the testing for 4 times for 4 folds.

So I have a data complete data I created into 4 different sets, set 1 as 25 per cent of data, set 2 25 per cent, set 3 is 25 per cent, set 4 is 25 per cent each. Why this 25 percentage because $100 \div 25$ will give 25 per cent in each set. I have an equal amount of data in the 4 groups. If you do not have an equal amount it is okay. So, what I will do in fold 1?

In fold 1, I will use set 1, set 2, set 3 for the training test set will be the 4. So, this test set will be tested now. Initially, I test the algorithm or the model which I created using the test set 4 all the rest set used as a training. In the second fold, I use 1, 2 and 4 as the training and 3 as a test set. So this data also will be tested now. Similarly in the fold 3 and 4 here we will use the other set all the data has been tested.

So this gives the performance of the algorithm on all the test data available that is complete data where N sample has been tested on the algorithm then this is more strong or more comparable as opposed to a simple split of 66 per cent with 33 per cent it is always advisable to use the cross-validation. So how many folds to choose is it 4 or 5 or 10?

It depends on the data set it is always good to choose N equal to 10 or 10 fold cross-validation is good. If you do not have that much data you can select a smaller number of folds.

One more type of cross-validation is called leave one out. So, in this particular "leave one out" approach suppose you have N number of dataset the N is the total number of data points here, in this case, we have a 1000 students data right.

If you have 1000 students data then "leave one out" will be this is the big N, you may consider the total number of students. The N fold will be 1000 fold cross-validation, but N minus 1 data will be used for training, we will repeat this for 1000 times.

That is more strongly suggested now, but it takes a lot of computational time and computation is costly also. So if you have huge data suggested go for 10 fold cross-validation or depends upon your algorithm you can choose the number of folds in cross-validation.

(Refer Slide Time: 15:21)

So, in this video we saw what is training data and what is test data split and also we discussed what is cross validation. In a next video, we will talk about some of the performance measures in a machine learning. Thank you.