Overfitting and the Bias-Variance Tradeoff

As we just discussed in the previous video, using various nonlinear functions of the axis to create new features and to have more variables in our regression. Let us recollect the results of our prediction exercise of sales based on the advertising budgets.

How about including a new variable, which is the product of how much you spend on television times and how much you spend on radio. We can say it as an interaction variable.

Why would you include such a variable?

Maybe TV advertisement by itself does nothing, maybe radio advertisement by itself does nothing, but if a person hears something both on TV and on radio, maybe that's what that person needs to be moved by it i.e. listening to an advertisement from two sources, maybe that has a bigger effect. So let's add this variable and run the regression. And the results of the regression are as follows.

- $\widehat{\text{Sales}} = 2.94 + 0.046 \cdot (\text{TV}) + 0.19 \cdot (\text{Radio}) - 0.001 \cdot (\text{NewsP})$

  $R^2 = 0.897$

$$\widehat{\text{Sales}} = 6.57 + 0.019 \cdot (\text{TV}) + 0.029 \cdot (\text{Radio}) + 0.001 \cdot \underset{\text{new feature}}{(\text{TV}) \cdot (\text{Radio})}$$

$$R^2 = 0.968$$

- Note: 0.001 looks small,

  but $(\text{TV}) \cdot (\text{Radio})$ is a relatively large number

By the way, in this regression, we removed the newspaper, we did not include it because as we discussed before, this coefficient is too small and we consider it insignificant, we have thrown newspaper out of our model, but we're including this new feature, which is multiplicative.

We noticed that the **coefficient** in front of **TV and radio** is quite small. On the other hand, prediction performance goes up quite significantly. So now it's up to 97% of the variation in sales is now explained if we use these three variables.

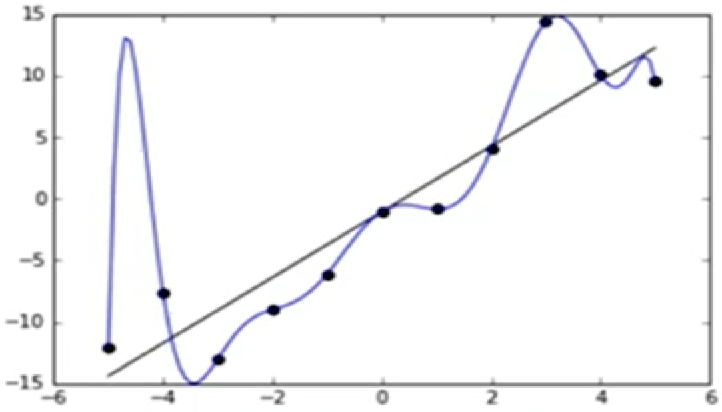
Then the question is, is this coefficient here really small or not? Well, that has to do with the units that are being used, if TV spending is in the 1000s, and the radio is in the 1000s. TV times radio would be in the millions. So we're multiplying a number that seems to be small but with big quantities, and so it could have a significant effect on the end results.

To avoid such situations and to be able to interpret the results a little better, it's a good practice to always rescale the variables so that they have comparable ranges. So take the TV times radio variable that would tend to be very big, but we normalize it, scale it somehow so that it's in the same scale in the same range as the other numbers, and by doing so, we will get more interpretable results i.e. more interpretable coefficients.

What this example suggests is that by adding new features, and new variables, you are going to do better.

So why not try to add more and more variables?

Well, if you add more and more variables, R squared is going to go up on the other hand, you might end up with curves like this.



https://en.wikipedia.org/wiki/Overfitting

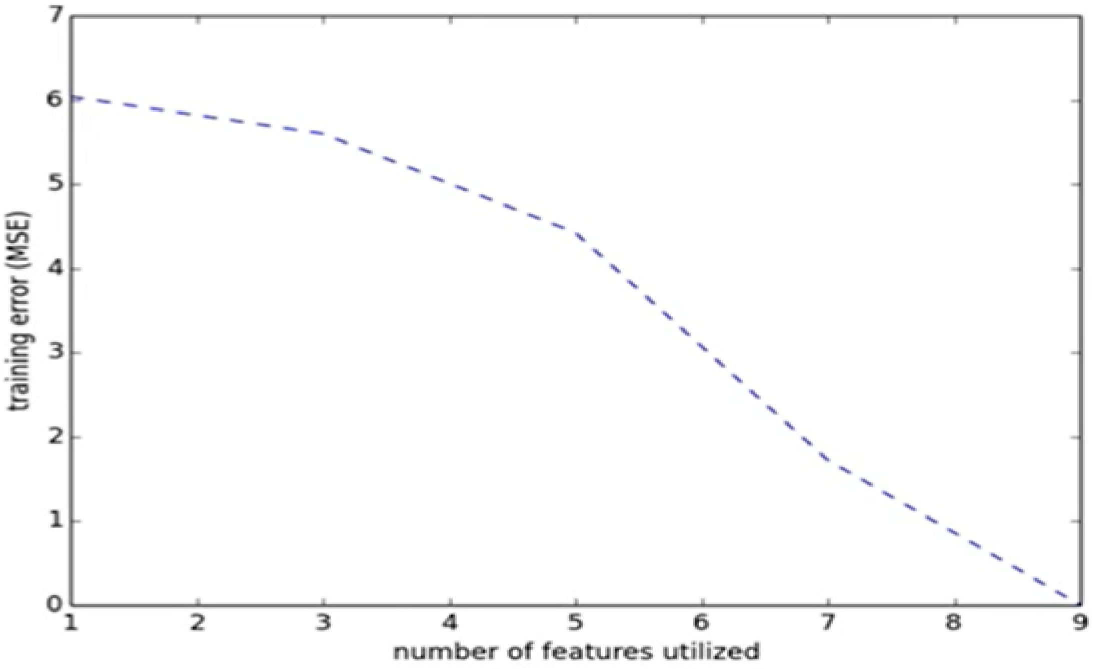
So here we have 10 data points and if I allow fitting the data using a polynomial of degree 20 that polynomial will fit the data perfectly. R squared will be equal to one, but the polynomial might be some nonsensical curve, like the one we have above.

We want to avoid that situation, on the one hand, we want to include useful variables, but we do not want to overdo it and we need some systematic ways of deciding how to do it.

How systematic can we be?

You can look at classical statistics texts and you will find various procedures and tests that kind of guide you on how to add and how to remove variables. On the other hand, the formulas that you will see there typically rely on certain mathematical assumptions that might be violated in the real world and you cannot really rely on those procedures.

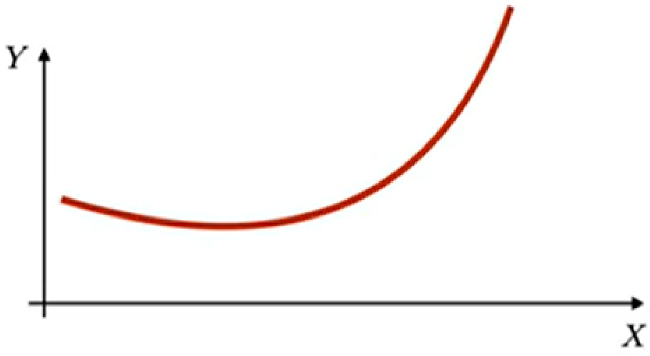
Instead, the more common approaches in machine learning and data science these days are, first to avoid overfitting by telling the mathematics somehow that you shouldn't overfit. The second approach is to use data-driven ways to decide when a variable should be included or not without having to rely on formulas that might have dubious assumptions.



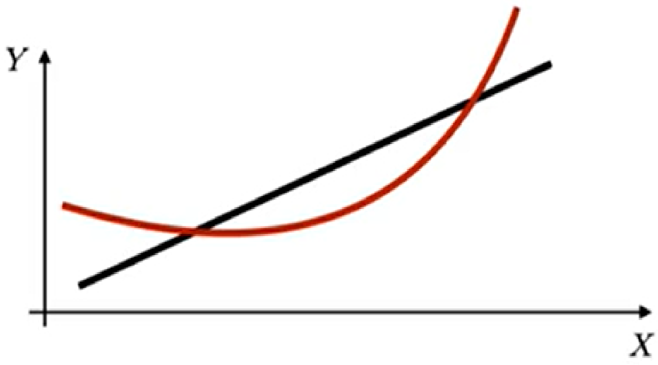
Here's a plot that illustrates our predicament. Suppose that we use more and more variables, the horizontal axis is the number of features utilized and the vertical axis is the mean squared error i.e. the prediction error. As we use more variables, the prediction error goes down, and with enough variables, we may get essentially down to zero. But this is the prediction performance on the datasets used to train and that might correspond to overfitting.

How can we tell what is the right number of features to use, and when does overfitting start?

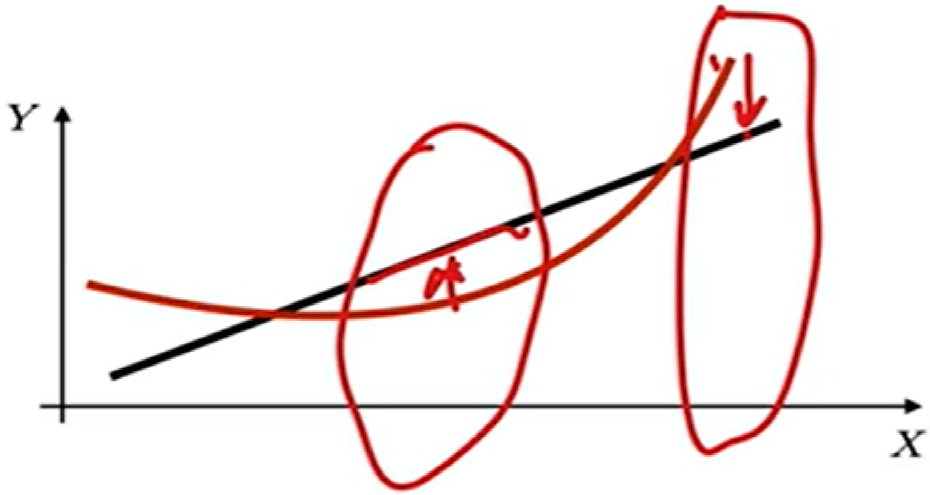
We will need a way of addressing the situation. Let's consider another point of view. Suppose that the true phenomenon is something like this, which is a quadratic relation between **X** and **Y.**



Even though the relationship is quadratic, suppose that we try to fit it with a linear function. A quadratic would have three coefficients, but we try to fit it, with just two coefficients using a linear function.



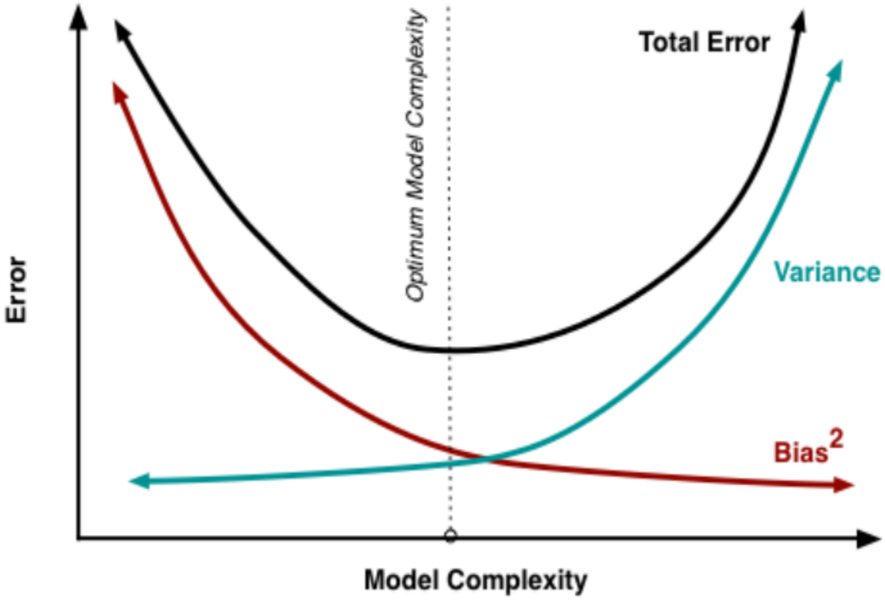
So in some sense, we're using too few variables, we're not going to do very well, in the sense that even if we had perfect data, even if somebody gave us the red curve, exactly without any noise, that's what you would get if we have infinitely many data. The black line would have some systematic errors which we call biases. For example, the black line is biased upwards systematically and in another region, it is systematically biased downwards.



These biases are impossible to remove if we have too few variables. At the other extreme, however, if we use lots of features and lots of explanatory variables, then what we have is we have too few data points per parameter.

There are some rules of thumb that in order to learn to identify a parameter, we kind of need 10 or 20 independent data points at least, so the more parameters you have, the more data points will be required. If your data set has only limited data, and if we keep using more and more variables eventually each coefficient is determined by very little data and so the estimates that we get, become very noisy. They're driven just by individual measurements and the situation is also prone to overfitting because, with lots of variables, we can try to fit exactly all the data that we have.

To avoid overfitting there's going to be a bias-variance tradeoff. If we have too few features, we suffer from bias and if we have too many features, we have too high variance in the sense that our estimates tend to be kind of completely random.



So somewhere in the middle is the right way of addressing this trade-off and finding these sweet spots for better model performance.