Module 6: Linear Regression: Part Two

Video Transcript

Video 6.1: Building and Interpreting Managerial Predictive Models (11:05)

So, last time we basically went from sort of the primordial ooze of modeling. We started with a dataset and visualizing the dataset and such things to actually building models. I kind of like want to really quickly recap where we were last time.

Last time, basically, we got to talking about multiple linear regression models. But basically, what we said we were doing was, we were modeling our dependent variable with as many independent variables as we liked, and hence, the x1, x2, ..., xk over here. And what building a model meant was really estimating that b0, b1, b2 all the way up to bk. Now, given a model, what we cared most deeply about was "Is this model any good?" And towards understanding that, we talked a bunch of different quantities. We said, "Well, one thing we could look at is what the sum of the squared errors is, the sum of the squared residuals of our model?"

Look at each data point, look at the actual value at that data point, what we're predicting at that data point, look at the sum of the squares. And so, we looked at that. Now, actually very early on last time, we said, "Look, we need some baseline. We need something to compare ourselves to."

And the thing we were comparing ourselves to was really what the sum of the squared errors for a baseline prediction, which in our case last time was simply the mean was going to be. So, in particular, there was a special name for that. We call that SST or the total variation, or some of you might simply recognize that as really kind of the variance, right?

And I was simply, the actual y sub i and the difference of that y sub i from the mean across all data points, And, that was our SST. And because we wanted sort of this relative measure of how good our model was, we introduced the R squared, the coefficient of determination, which simply really told us how much of that initial uncertainty that you had when you simply predicted the baseline, how much of that initial uncertainty you get to actually reduce, right?

So, that's where we were last time. And where we're going to pick up today is we're going to get along this path of building the best model we can build, that's really where we want to go. But before doing that, I wanted to take sort of a quick digression to say that in addition to just predicting a number,

predicting rentals during a specific hour or something like that, I could also say something more really about how certain I am about my prediction. So, given that, we're measuring all these quantities related to the fit of our model, we can actually go beyond just point predictions.

And I want to talk about that, and we can file that away. So, what do I mean by going beyond these point predictions? Really, this is how I visualize it, let's break this down. So, let's say, 'x' is our independent variable, 'y' is our dependent variable. The actual data, if we had to scatter plot the actual data, what you saw with the scatter plot was these dots, these little black dots that you see on the screen. And as you can see on your screen, the blue line are the linear fit, the straight line simple linear regression fit, is a really great fit. On the other hand, it's not a perfect fit. That is to say, on any given point, on any given independent variable 'x', we see a whole bunch of realizations of what the actual thing was, what actually happened, what the actual 'y' was, right? And as you look at that, if I draw your attention to let's say, the second 'x' from the left, what you're seeing is that at that x, the actual y runs the gamut, there's almost a bell curve to it where the mode that the top of that bell curve is exactly what you'd have predicted. The mode of the bell curve is exactly what your linear model says, but the actuals are going to be spread out across that as indicated by the bell curve that you're seeing on the screen.

Now, this bell curve should give us the tools in principle to talk about how certain we are about our prediction. And in order to do that, what do we need? Really, what we need is to understand how fat or skinny that bell curve actually is. In very simple terms, we want to understand at a given point if I thought of really the actuals as being a normal random variable, we know the mean of that normal random variable - the mean is simply what our prediction was. So, all we got to get is what is the standard deviation? And it turns out, it's pretty easy to get to what that standard deviation actually is. How do we find it? We find this really as an outcome of the linear regression that we ran last time. So, for instance, the best model that we had when we did our last session, when we first introduced multiple linear regression with blue bikes, was we compared rentals against temperature and relative humidity. That was our our best model.

If you fit that model, you could ask the model that was fit, "What the standard deviation of this bell curve actually is?" You'd get that by basically asking for a certain property of the estimated model called the MSE or the Mean Squared Error of the residuals. If you're following along in your Colab, you'll see this in your Colab. And by asking the model for the mean squared error of the residuals, we get this number 191.63, that's exactly the standard deviation that we're looking for. So, let me actually orient you to where we are. Again, the goal is to go beyond just a point prediction. The goal is to say, "Hey, you know what, if tomorrow the temperature is 75 F and it's a really humid day, 80%, then between 5:00 and 6:00 PM, you're expecting to rent, I'm making this up 500 bikes, but kind of a plus or minus on that is 400 or

it's 200 or it's something, right? Really, that plus or minus, that standard deviation, is what we just found. And in this particular case, that plus or minus, that standard deviation was 191.63. So, in essence, we have everything we need to go from just a point estimate, essentially predicting the mean to essentially talking about the entire distribution of outcomes that's actually possible.

So, as a concrete exercise to just get into that, let's say that we're told that tomorrow, just for fun, it's going to be 60F and 50% humidity. For me personally, I would love to bike in this kind of weather. The weather is just right, you're going to be feeling just great, but you probably don't care about that. I care about it, and maybe what I care about tomorrow, if I'm running blue bike is if it's 60 degrees F, 50% humidity, you've already told me what you're predicting rentals to be? We did this last time. You got this by simply plugging in 66 degrees for temperature, 50% humidity for humidity, and out of this model came a prediction of 296 bikes. But now, we're asking ourselves, "Listen, I'm not happy with just the mean. I want to understand the entire range of outcomes, what's the standard deviation?"

Well, we just found that the standard deviation was 191.6 for this particular model. If I were to look at it graphically, I'd draw your attention to the bell curve on the slide. What is this bell curve telling us? The bell curve is telling us that our expected prediction, our expected rentals, is 296 bikes. The plus or minus, the standard deviation is 191.6. And so, now if you ask me a question like, "What is the probability that you're going to rent more than 400 bikes?" So, the mean is 300 or 296. What's the probability that you go even above that? What's the property that you rent 400? That's the little shaded red area that you see under the bell curve. And to get that precise probability, to get that precise area, we're really looking at the area under that normal curve. This is something related to the normal CDF, the normal cumulative distribution function. Cumulative distribution functions are things you saw when you actually talked about random variables. In this particular case, we're looking at the cumulative distribution function for a normal random variable with mean 296 and standard deviation 191.6, and voila, the area under that curve is 0.29. Quick question: If somebody asked you, "What's the probability that we rent less than 400 bikes?"

What would that be? Hopefully, you quickly say, "Well, if the probability of renting more was 0.29, the probability of renting less is going to be one minus that, 1 - 0.29, that would be 0.71, 71%. Often, I find that in business applications, the point estimate takes you to a certain place, but then as you're making a decision, maybe you've got to make an investment, right?

As a result of what the model is actually saying, you need some characterization of risk, some sense of how confident you are in what the model is saying, and going beyond point predictions, like we just did, is a useful way of doing just that. It's a useful way of conveying to the folks that are consuming from this model that are making decisions based on this model, how sure we are of what the model is telling us

and what the range of potential outcomes can be. Having done this, what I want to move to next is, as I said, the task of the day, building the best possible model that we can.

Video 6.2: Building a Good Model (08:43)

So, getting to the work of the day, really what we want to do is build the best model we can build, right? And actually, deliberately I wanted to flash a slide that was sort of in some sense, the cliffhanger on which we ended linear regression one. What we did there, just in very succinctly, was we learned how to do multiple linear regression. If you remember there was a question at the end of the lecture saying, hey, look what happens when you add new independent variables? And what you concluded was R square can, you know, it either stays the same or goes up. And a natural outcome of that was well, if that happens, why don't you just keep adding variables? Like do that, keep adding variables and declare victory. And what I said last time was that's not as simple as that.

We still want to build the best possible model we can build but we can't just go around willy nilly, just adding everything we want to, that's available to us. And so, what I want to do today is break down what can go wrong into sort of three buckets, and we kind of deal with each of these buckets in turn. Number one, the presence of irrelevant independent variable. That is to say what I'm talking about over here is that you just throw in some junk that has nothing to do with the actual model at hand that could actually lead you astray. Here's a question I would ask. If we think of variable as irrelevant, let's say it's actually completely unrelated. Like I'm looking to predict blue bike rentals in Boston and the independent variable I throw in is pop tart consumption in San Francisco. I mean completely irrelevant potentially, right?

Or maybe I even just generate random numbers on my computer. That's clearly irrelevant. So, if I throw in a variable that's completely irrelevant to the task at hand, what do you expect the coefficient in front of pop tart or the coefficient in front of random to actually be? Hopefully, as you thought about this this, you're saying to yourself, zero its irrelevant, you are saying to yourself, zero its irrelevant so the coefficient should actually be zero. It turns out that it might not actually end up being zero. Turns out you may run this regression and the coefficient in front of this irrelevant variable is small but not zero. Why would something like that happen? Now, this is a bit of a tricky one, but it's this. We don't have the luxury of infinite data.

And this is something we've actually not thought about but it's going to be very relevant today. We had 9000 odd data records for a blue bike rental, maybe that's big, maybe that's small, we don't know. But it's finite, it's not infinity. So, because we don't have the luxury of infinite data, we have a finite dataset and there's noise, you may actually find that our model, because of the fact that we have finite data, starts picking up things that are not quite right.

We can be fooled by noise. So, if we can be fooled by noise, how do we you know kind of avoid that, right? How do we protect ourselves from getting fooled by noise? Well thankfully, when you actually run linear regression, there's a bunch of analysis going on that lets you understand whether the coefficients in front of the independent variables you're proposing for your model are 'significant or not significant'.

Now this is potentially the first time you're hearing this word significant or not significant. When you sort of hear the word significant, I want you to think that significant simply means I believe that coefficient. And not significant is hey, this is kind of like a pop tart type variable or a random one, like the coefficient actually is meaningless. Now, let me now kind of start with like getting down to brass tacks over here. Like in thinking about whether a coefficient is significant or not, the first thing you're actually going to look for is the p value corresponding to that particular variable. What are these p values? These p values essentially tell us whether a specific coefficient that you're looking at is significant, i.e. something you can trust or you can trust its sign, right?

or something that's actually not significant. And typically, if you actually look for p values, this isn't quite hard and fast, but this is typically what gets done by practitioners 99% of the time. A p value that's smaller than 0.05, we'll call that significant. If it's bigger than 0.05, we'll say it's not significant. So, in this particular case where we ran rentals versus temperature and humidity, you can actually look at that column which I've highlighted over here, looking at the probability that the column that says probability greater than T, that's actually the p value, and you'll see that the number is essentially up to three decimal places zero. For intercept temperature, relative humidity, all of those numbers are smaller than 0.05, and so, we're happy calling each of these variables significant. Now okay great right! What you should be taking away is okay, very good, there's this thing of insignificant and significant, this helps me avoid falling into the trap of putting pop tarts into my model or random stuff into my model.

The way I'm going to check for this is look at the p value and make sure that it's less than 0.05. But really, I find that it's useful to go kind of a little bit deeper and actually understand what is this p value thing. What's actually going on over here? That turns out the p value is actually a very simple concept. It's really, really simple. Really, the idea of the p value, it's basically, you're trying to ask yourself, what is the likelihood that you're seeing something other than zero by sure chance?

And sure, chance can actually come to bear because remember we only have finite data. So, one way of thinking about this, a different way of thinking about this is by looking at confidence intervals around our coefficients. So, what is this confidence interval? The idea with the confidence interval is it's some interval, 4 to 6, 4 to 9, 2 to 6, Like right like two numbers, right? It's an interval where you essentially say that this quantity that you're trying to estimate, that you want the estimate of is in that interval with some pre specified probability. So, for instance, if you ask for '95%' confidence intervals' and I come back and

I give you the 95% confidence interval for the temperature coefficient is 5,7, that's equivalent to me saying that I'm 95% confident that the true parameter in front of temperature lies between 5 and 7. So in particular, if I apply this to the regression model that I ran, turns out you get those confidence intervals for free. Coming to temperature, which I was just talking about, it tells me that the 95% confidence interval. So, 95% is basically, that's why you see 0.025 and 0.975. There's a 2.5% probability you're on the left, a 2.5% probability on the right that we won't be able to account for, but for the 95% in the middle, between 97.5% and 2.5%, the interval is 6.16 to 6.606. And coming back to that interpretation I was asking for, essentially all I'm saying is that I'm 95% certain that the true coefficient in front of temperature is between 6.16 and 6.61, or that the true coefficient in front of relative humidity is between -3.01 and -2.58. That's really sort of what's going on over here.

Video 6. 3: Challenge #1: Irrelevant Variables (09:22)

So, we're at this spot where we know what this sort of notion of a significant variable is, or a not significant variable. And in particular you know if I threw pop tarts or something random into the model, we want to know that that's kind of not significant. We want to know that it's irrelevant. We can't take this for granted because we have finite data. Because we have limited data, although something is irrelevant, you can throw it in there and the model may actually come away with a coefficient, that's not quite zero, right?

So, we've got the tools to do this, like we talked about p-values and so on and so forth, but I want us to sort of see this in action. I want us to see how this actually works. And I thought, well, what better way to do this than to actually generate something that's truly random. Throw it in there and see what happens. So, that's exactly what I'm going to do. And by the way, if you're following along in your collab, you'll see this in the collab too.

So, I'm going to generate as I said something entirely random and plunk it into that data frame. So, the random column that that we've added on is literally a random number. It has nothing to do with blue bikes, it was generated randomly right there in the collab. So, I've generated a bunch of these random numbers, they are random numbers between zero and one, and I've thrown it into my data. Right now, very suggestively labeled that column random. Maybe I should have called it pop tart, but I've thrown it in there. We know by design that it's obviously irrelevant.

First question. It's independent, so it's uncorrelated with rentals, right?

Well, let's actually go look at the correlation between random and rentals. Okay, here it is. This is the correlation between the rentals part of the data frame, what we're trying to predict, and these random numbers that we just generated. I've scattered it for you. You look at the scattered part, looks pretty random. Look at the co-relation. Look at the correlation. It's not zero. Well, here's the thing. If we had thousands upon thousands, upon thousands, upon thousands of rows of data, that number would get close to zero, it would get very, very, very close to zero, but we don't.

We have the 8,600 odd rows of data and because of that, just to assure noise, we can't know that the correlation is exactly zero. We know that it's close to zero, but the number we estimate isn't quite zero, it's 0.2. A correlation of 2% but it's not really 2%. You can be fooled by noise, given the size of this dataset and that's kind of the crux of what's going on where if we actually ran sort of linear regression, we might actually get fooled. The linear regression model might pick up like the ever so slight signs of a correlation, although there isn't one there. It's being fooled and we don't want to be fooled. So, why don't we just try doing this.

Let's look and see if the tools we have to detect whether we're being fooled or not, whether these tools actually work. And so, I suppose we're just doing this correlation, let's actually do linear regression. Now, this was the original linear regression model, right?

Our favorite sort of rentals versus temperature and relative humidity, we've seen this a bunch of times. What I want to do next, okay is I want to do a regression that throws in random. So, it's a regression of rentals versus temperature, relative humidity and random. So, let's do just that, I did just that. I've now run a regression of rentals versus temperature, relative humidity and random. Let's look at the coefficients. So, our good old intercept, that's like 56, 57.

The temp coefficient, sort of similar to what it was before, 6.3834; rel_humidity, also roughly close to what it was before, -2.8. And look at random! Random has a pretty big coefficient in front of it, -7.99. Well, what would happen if you deployed this in the real world?

That extra term?

That extra term is garbage. It's actually corrupting the actual model you've built. It's going to hurt you when you actually deploy this model on future data. And so, we obviously don't want to have that in the model. What's the tool we have?

The tool we had was p-values. What do we say if the p-value < .05, then significant, p-value > .05? Not significant, throw it away. What's the p-value in front of all of these variables, right?

Well, in front of Intercept, its zero, great. In front of the temp, zero, great. In front of rel_humidity, zero, great. But let's look at random, 00.264, way, way bigger than 0.5. We can't trust that coefficient random is not significant. I want to keep with it for a second. Look at the confidence intervals, in front of random. What are the confidence intervals?

The 95% confidence interval goes from -22.022 to 6.030. What's that telling us?

It's telling us that the coefficient in front of random could very well lie in that confidence interval between -22 and 6.030. But guess what number is between -22 and 6.030, 0, right?

So, we can't rule out zero. This is yet another way of saying that random is not significant, because you cannot rule out the fact that the coefficient of random is zero. And so indeed, we see that when we threw in something that was completely sort of unrelated, our tools, these tools that we now have in our back pocket, p-values and confidence intervals, led us we know a way, the sorts of things that can fool us when we deploy these models in the real world. We wouldn't be fooled by randomness over here.

Let's get back to the task at hand. The task at hand was, we wanted to build the best possible model. So, earlier we had temp and rel_humidity. I've added in now windspeed and precipitation. This is more or less everything we have. So, I'm looking at rentals versus temp, rel_humidity, windspeed, and precipitation. If I do this, by the way you'll notice that if you look at the R square, it's now snuck up a tiny bit. Not a huge amount, but it has snuck up. Okay, it's 0.290. Earlier, it was 0.289. And so, it's actually snuck up. But here's a question for you. This is the output, right?

You can run this on your collab as well. Are all the variables in our model significant, or have we let some insignificant variable sneak in. Take a second, look at what you got. Reflect on the output of the model. And let's think about whether we let insignificant variables actually sneak in. If you thought about this for a second, you've looked at the p-values and you've noted that there's significant everywhere. They're close to kind of zero everywhere, except in one place, windspeed. Windspeed has a p-value of 0.253. What's not telling us?

That's telling us that any coefficient in front of windspeed cannot be trusted. So, look at the confidence intervals, right?

The confidence interval is between -0.321 and 1.222, that includes zero. So, we can't rule out zero. And what does this tell us?

Windspeed is almost as good as kind of that random column that we had earlier. We want to actually throw it out. Except that in this particular case, this was not obvious because we didn't know Apriori that windspeed was irrelevant to the model.

Just think about that for a second. Just think about what we've done. Earlier, we went through this thought exercise where we constructed something irrelevant, and we were actually able to figure out indeed it is irrelevant.

Here we now have the power of looking at a data set that somebody else gives us. We don't know how it's been put together and we're able to tell whether a specific variable windspeed in this case, is irrelevant to our prediction task. And the model that we can now come up with is maybe we drop windspeed. We look at rentals versus temp, rel_humidity and precipitation. All of these are now significant. R square also higher than where we were earlier. Not a lot higher, but it is higher, and all these coefficients are significant. And so, this is potentially a better model.

We've gone from the model that we had at the start to one that's actually better than it.

Video 6.4: Challenge #2: The Impact of Highly Correlated Independent Variables (10:16)

So, we're on this journey where we're trying to build the best possible model we can, right?

And we start out by saying, "Look, it's not quite as simple as just throwing in all of the independent variables we can sort of get our hands around" because all these crazy things can happen. And we saw, for instance, one of the crazy things that could happen is we could fool ourselves into actually thinking that something that was truly irrelevant to the model was, in fact, relevant.

We saw this in a bunch of different ways, we created something random, and we found that the coefficient in front of that random thing could well be non zero. So, we didn't know that that random thing was not significant, we'd have been fooled by noise. And we've got a tool to deal with this: P values, right?We've got that squared away. The second problem is the presence of highly correlated independent variables. Now, if this is bringing about some deja vu, it should, because if you remember when we were actually talking about temperature in the last class, I mean, at the point that the dataset had two types of temperature in it, it had temperature and then it had this weird wet bulb temperature or whatever.

These two things presumably might have been related to each other. Well, let's say we hadn't paused to think about the name over there. Let's pause to imagine that the names weren't so suggestive, temperature and wet bulb temperature. And so, I'm running literally that regression, rentals versus temperature and wet bulb temperature, right?

And so, I'm expecting to see three coefficients come out of this. One, the coefficient for the intercept, which I'm showing you over here -119, and then two, the coefficient in front of temperature and wet bulb temperature. So, here's the thing. I'm not showing this to you because I want you to think about what you expect that coefficient to be. By the way, this is something I like doing, right?

It's very easy, by the way, to kind of rationalize things to yourself. But ex ante and you look at something like this, you should be able to think through this and say to yourself, "What do I expect the coefficient to be?" Its temperature, going back to just the modeling of blue bikes.

We expected temperature to have a big impact in the sense that if it got warmer, we expected more people to get out and ride bikes. So, if I look at temperature and wet bulb temperature, what do you think the coefficient is going to be? Now, if you're thinking to yourself, "Hey, you know what, 25F I expect not a lot of people to be riding bikes, 70F, I expect lots of people to be riding bikes. You know what? I expect this coefficient to be positive."

Let's look at what actually happened. Here's what actually happened. The coefficient in front of temperature, 19.3338. The coefficient in front of wet bulb temperature -14.72. Now, I told you what wet bulb temperature is. Again, not a meteorologist, but wet bulb temperature is just another way of looking at the temperature that adjust for humidity or some such thing, right? So, does it make sense that the coefficient in front of wet bulb temperature is negative? I mean, it doesn't to me.Both these things are telling us the same thing. If I just look at the wet bulb temperature, this is telling me the opposite. It's saying that on cold days, you're actually going to be riding more. That doesn't make sense to me. What's actually going on over here?

Here's what's going on over here. The issue is that temperature and wet bulb temperature are basically the same thing. Not quite exactly the same thing, but they're close to being exactly the same thing. Now, to build a sharp intuition for what might actually happen, let's imagine for a quick second that they were exactly the same thing. So, basically, somebody came along and, as a prank, put in a second column that was exactly the same as temperature. Let's call that column twin.

So, I have temperature, those are the original column and added in another column, which is twin, and twin is exactly temperature. It's no different. It's exactly the same number. That is, to say if temperature is 50F, twin will be 50F. If temperature is 42.7F, twin will be 42.7. Let's just think about this for a second. If twin is a clone of temperature, and you sort of built, let's say, the original model, the perfect model was that rentals had to be -118 + 4 * Temp.

Now, when you're throw in twin, you can't quite trust the coefficients in front of either of them because effectively, you're going to get the exact same prediction as long as the coefficients add up to four. Let's

think about this. So, if the exact model was 118 + 4 * Temp, one model you could bring back is 118 + 4 * Temp + 0 * Twin. No issues with that.

Here's another one though. I could put in -118 + 2 * Temp + 2 * Twin. That's going to give you the exact same prediction. Another thing I could do is I could say -118 + 19 * Temp - 15 * Twin or I could keep going, -118 + 100 * Temp - 96 * Twin. As long as the sum of the coefficients adds up to being roughly four, you're going to get exactly the same prediction. And so, this is the issue with what happens when some of these independent variables are super correlated with each other. As soon as these variables are super correlated with each other, you can't trust the sign in front of any of them.

This problem is called multicollinearity, and, in fact, as I said, this should bring about some deja vu. When we'd looked at correlations in the last session, we kind of noticed this, right?

When we look at the correlation between temp and wet bulb temperature, which I've circled over here in the cross tab of correlations, it's not quite one, but it's actually really close to one. It's 0.98. The temperature and the wet bulb temperature are very, very, very strongly correlated with each other.

And as a result, the situation at hand is not unlike the situation in our thought experiment where I can realize the same prediction with a whole bunch of different coefficients. I can no longer be confident in the predictions that I'm coming up with. So, what is the problem? So, this problem is called multicollinearity. Multicollinearity is essentially an issue where you can almost think about it as this thing of having temp and twin both in the model. Now again, because we're working with data, with finite data and there's noise in it, you're never really going to see an exact duplicate. But we are going to see things that are highly correlated either positively or negatively. There's no fixed rules for what constitutes high correlation but anything higher than 0.75, anything smaller than -0.75, maybe that's high. If you want to make it 0.9 and -0.9, that's fine too. But how do we detect it? We detect it simply by producing a correlation matrix. This is, by the way, as a good data scientist, something we did last time already.

As we were building up our model, we wanted to understand what were the features that were most related to each other, and we produced this. Well, that same correlation matrix can help us spot variables that are highly correlated with each other. And what's the problem with this? The problem with this is when these variables are highly correlated with each other, we get into the temp twin problem. It's like having temp and having a duplicate of that exact same column, that basically makes it so that I effectively can't really be confident in the signs of the coefficients. There's a whole bunch of things that happened over here that aren't so great. How do we correct for this? We correct for this in a really simple way. Simply remove things that are highly collinear. So, in particular in this model, we look at the correlations across these independent variables. Really the independent variables that are super

correlated with each other are really temp and temp_wb, like if you used your cut off of high correlation as 0.75, you'd basically say temp and temp_wb over here are very correlated with each other. So, given that temp and temp_wb are very correlated with each other to essentially avoid a multicollinearity problem, we're only going to keep one of them. So, we have two: temp and temp_wb. Which one should we keep? So actually, I want you to think about that.

Hopefully, you're convinced that we should only keep one because otherwise we get into this issue of temp and twin, won't be able to trust the coefficient in front of either. Given the choice between temp and temp_wb, which of these do you want to keep?

Now hopefully, you thought about this and you saw that effectively that when I look at the correlation between temp and rentals, that's 0.48. So, temp has a lot to do with rental, something we spotted a long time ago when we first looked at blue bikes, right?If you look at the correlation between temp_wb and rentals, it's also pretty heavily correlated as you might expect, but not quite as strongly correlated as temp. The correlation there is 0.43, and so given a choice between which of these I want, I'm going to pick temp because it's more correlated with what I care about predicting. That's quite simply how we correct for multicollinearity.

Video 6.5: Challenge #3: Too Many Independent Variables (08:45)

Okay, so continuing on this journey of building the best model we possibly can with linear regression, right, we've sort of looked at these potential stumbling box, we've kind of dealt with two of the three. One of them was irrelevant independent variables, p-values, we know that. Two, what about like things that are super correlated with each other, temp, wet-bulb temperature? Simply look at correlations and just remove things that are super correlated. Pick one. I want to get to the sort of the third one and the third one is to be honest with you, a little bit tricky. Third problem is: can there be such a thing as having too many variables relative to the size of the dataset?

Now, colloquially, you've potentially already heard of sort of this notion of overfitting. What is overfitting? Overfitting at some level in some sense is us sort of saying that like, we kind of have too many variables relative to the size of the data. So, I want to kind of break this down. I want to do one of these sorts of back-of-the-napkin thought experiments. Okay, so here's a thought experiment where let's say this is our data, y versus x and we got a bunch of these points. And your task is to fit a model to this data. I give you x, you want to predict y fit a model to the data. What sort of model should we fit? Now maybe just given that we've been talking about linear regression for this lecture, your mind left to kind of like doing the blue sort of dotted line to the right, the straight line. But you could equivalently have kind of drawn

this sort of squiggly curve that on the left, that went through all the points. And really, why not do the curve on the left? Because if you look at the sum of the squared errors for that curve, it would be zero. You fit each point perfectly. And as a consequence of that, the R square is 1. It doesn't get better than one. One is the best.

So, that feels weird, especially because if you look at the straight line, we don't need to do this precisely, but it's very clear that we don't predict any point perfectly. And since we don't predict any point perfectly, we're going to have errors and so the sum of the squared residuals, that's going to be bigger than zero, and as a result, R square is going to be less than one. So, why is it that we want to do the thing on the right versus the thing on the left?

Here's the thing: The thing on the left, the squiggly line that perfectly fits the data that we have; the thing on the right does not perfectly fit the data that we have. But here's the issue: We don't care about predicting the data we have because we already know it, right? I mean, right, that's not a prediction task, you have the data. Rather, the subtle thing that's going on over here is what we're really asking, what we really care about is how good is our prediction going to be on, "future data". It's sort of like L-pose, what does future data even mean? I mean is it part of the data like what is future data? So, we're going to try and make that more precise. But first, I want to kind of do a thought experiment related to future data.

So, here's my thought experiment. So, let's say like we focus on sort of the x value that I'm kind of circling over here. It's sort of like, the third dot from the right. So, I focus on that. And I plugged in that x value, and in my data, I got a certain y value. But let's say in the future, we sort of plugged in that same x value, we don't know this future world, but let's say in that future world, what we got were the green points. And really our task is to understand, well, what's our error on this future data? That's essentially what I'm asking. So, I've got a particular prediction. I've got these future data points and we can ask ourselves, what's our error on these future data points? Well, you can kind of work that out over here. I can look at the SSR on the left. I have three future data points, and on one of them let's say, I nail it. I get zero error and then on one of them I get an error of one, on the other of them, on the second, I get an error of two. So, what's the sum of the squared errors? 0square+1square+2square that's 1+4, which is five, right?

Let's look at the picture on the right. I've nailed one of the future points but look at the other two future points. One of them at some level is, its going to give us an error of one, the other is going to give us an error of one, the sum of the squared residuals, the sum of the squared errors, 0square+1square+1square=2. You're probably saying, "Well, wait a second, maybe you could have kind of drawn this differently or whatever and gotten a different answer." You'd be right. But in essence, what I'm kind of trying to convey over here, is that at some level if you believe that there's going to be

uncertainty in future data, your best guess is to kind of get at the mean value of what that future thing is going to be. If you're going to kind of the outcome is going to be something random, you don't know what that randomness is, your best bet, your best option is kind of predicting the mean and that in some sense is what the straight-line curve is actually trying to do. So, put a different way, the guy on the left, the squiggly curve, it almost fits the data too well and why is it, "fitting the data too well?" It's because it's sort of chasing the noise in the data. And that's kind of what we don't want to do guys. We don't want to chase the noise in the data. We want the model to be complex enough, so that we capture the signal in the data, but not so complex that it's kind of chasing the noise as well. That's kind of really what we're after. And this is really this problem of overfitting, and we sort of say, that the picture on the left is kind of overfitting the data, and the picture on the right hopefully is not. Now here's the thing, this problem of overfitting, serious problem. And in fact, with sort of the software tools that we have, the sort of big data tools that we actually have, it's almost trivial to kind of find yourself in this trap where you really overfit to the data. For instance, and you don't even need new independent variables. One of the things you could do, for instance, is you could transform variables you already have. You say, "Hey, I've got temperature. Why don't I create these sort of synthetic independent variables by looking at the squared temperature and the cube of the temperature and the fourth power of the temperature and so forth." And very soon you find yourself fitting rentals to really temperature, but this very complicated function of the temperature. And you might really kind of get yourself into a place where you're chasing noise.

At some level when we sort of think about the world of machine learning and you think of the world of what's called high-dimensional statistics today, fancy terms, it's all about how to kind of avoid this trap. How do we actually realize the fact that the data we have is limited and doesn't quite entertain the sort of dimensionality or complexity of model that we're trying to fit to it? That's kind of the task. And so, we want to kind of avoid that, and one way we have, one tool we have to doing this is sort of avoiding non-significant variables, like the sort of p-value thing that we've already done. What I want to sort of talk about over here is another tool which I think is super crucial to you, sort of having in your toolkit and that's out-of-sample testing. And so, really what is the idea of this sort of out-of-sample testing? It's a way of us kind of creating a sample of future data. So, if you come back to the thought exercise that we did over here, right, we were able to kind of judge that we prefer the blue curve over the red curve by having access to these three future data points, the three green dots that we made up. Now, wouldn't it be great if we had a mechanistic way of producing these future data points? That would be awesome. And out-of-sample testing is going to be just that.

Video 6.6 Out-of-Sample Testing (07:37)

So, we want to actually have some way of mechanistically generating future data for ourselves where we going to do it out of sample testing. The idea, super simple, and it's in your colab, by the way, if you want to follow along, right? How are we going to do this? We're going to take our dataset, and we're going to partition it randomly. So, we're going to take the dataset, we're going to create two datasets out of it, a training dataset and a test dataset. The idea is that let's say, we put half our data into the training dataset, half our data into the test dataset. Maybe we put 80% of our data in the training dataset, 20% in the test dataset. Doesn't quite matter, okay?

But the idea being that, if I look at a row of data, let's say, it's 50-50. The idea would be, if I look at a row of data, I flip a coin. If it comes up heads, I put it in the training dataset, comes up tails, I put it in the test dataset. All right? So, that's it. That's all I'm going to do. And you'll see in your colab, there's a slick one-line kind of thing of taking your dataset and splitting it in this sort of completely random fashion into a training dataset and a test dataset. The idea over here, really simple. What we're going to use the training dataset for? As the name suggests, it is to train our model, to fit our model. At the test dataset, it's kind of our future data, right? Now, the thinking over here is actually really simple. The idea is that when we fit our model, our model we fit did not get to look at the test data. It didn't get to peek at that. So, it stands to reason that this is a good stand-in for, "future data". And so, I might split training and test data. I might get these two buckets of training observations and test observations. What I'm trying to illustrate over here is what I said earlier, "We look at a row of data, we flip a coin, and either put it in the training dataset or we put it in the test dataset." So, we've generated this training dataset and test dataset. Why did we generate the test dataset?

We generated the test dataset to figure out how well our model would do on future data. Well, anybody remember what we used to measure how good our model does on data? From last lecture, it's the coefficient of determination, R squared, but the thing is that the R squared measures the fit to the training data. What we do is we look at the model that we built, and we ask ourselves, what is the SSR on the training dataset? And then, what's the SSR of the baseline model applied to that same training dataset? What we need to do, if we're going to understand this on the future data, is simply do that same calculation but on the test dataset. And I'm going to call that, out-of-sample R squared. What else would I call it? It's basically taking that R square concept, this coefficient of prediction, for a model that somebody else gave us and running it on this sort of dataset that the model has not had the benefit of seeing in the past. Now, just to give you a sense of really how that calculation might work, I've kind of worked out a simple calculation over here. I'm not necessarily going to talk through this in gory detail, but I'd love for you to pause the video and recreate this calculation for yourself. What I'm trying to do in this calculation is I'm saying, "Look, I have a simple linear regression model.

So, one variable model. And the model is 10 + 5x. And the baseline, I'm calling it 30, y-bar is 30. And then, somebody comes along and gives us this dataset of three data points, 1, 5; 2, 7; 3, 10, and our job is to figure out the out-of-sample R square, that's what I've worked out for you on this little example. I strongly encourage you to pause right here and work this calculation through for yourself. But in principle, what we're doing is actually really, really simple. We're basically generating this future data and we're asking ourselves, how good is our fit on the future data? That's it. Nothing more complex. Now, why are we doing this? This is the big thing. Here's a picture that we could potentially have drawn last time when we just did multiple linear regression. We said that with multiple linear regression, as we add additional independent variables, R square can't get worse. And so maybe, you see a curve that looks like this.

As you add number of independent variables on the x-axis over here, and you measure the R square we've been measuring all along in sample, that R square gets better and better. It starts flattening out, but it does never get worse. This is what we saw last time. Well now, what we have is out-of-sample R square. And so, if we believe there's bad things happening, that out-of-sample R square might start getting worse. Indeed. If we actually had a test set, we could similarly measure out-of-sample R square on that test set. And what we might see is a curve that looks something like the lower curve, that goes up and then goes down. When we look at that curve, what's actually going on in this curve?

Well, the right part of the curve, where you see the lower curve going down, that's an example of overfitting. Well, we've thrown really so many independent variables into our model, and we have so little data to learn the dependence on those independent variables, that what we learn is garbage. And when we throw the model to future data, we've kind of overfit. What's going on the right over there, in other words, is we're transitioning from doing straight lines to doing squiggly lines, in effect. That's what's going on, we're overfitting to the data. On the other hand, if you look to the left, the in-sample R square is getting better. But the out-of-sample R square is also getting better. I'm making up the term over here, but you could call this underfitting. That is to say we have other independent variables that we could throw in, that would genuinely make our model better in the real world, on future data. In this case, on the test set. And really, what we're looking for is that Goldilocks zone right in the middle. At some sense, we want to stop, where if we threw away independent variables, we'd get worse. If we added independent variables, we'd get worse. So, that's just right.

And so, what I want to do now at this point is we kind of looked at the three pitfalls that we might have run into. We looked at, as we just saw, building too complex of a model. We looked at multicollinearity. We looked at putting in insignificant variables. We know how to deal with these things. And so, what we want to do, what I want to do next is just summarize where we are. Put all of this together, so we have a

powerful recipe in our back pocket for linear regression and not just like any old linear regression, really building the best model possible with the dataset we actually have.

Video 6.7: Module Summary (06:45)

So, I want to put together everything we've done, right? And sort of end on this sort of very tacit recipe of how we're going to do linear regression in the real world, how we're going to do multiple linear regression in the real world, build models that kind of truly work. And recipe actually is not that complicated, now that we've kind of pieced together all of the things that actually matter. On the left, I'm starting with sort of a picture that we can hopefully all justify. We're looking at R square our proxy for kind of you know the predictive accuracy of our model.

That's a number between zero and one, as we all know. And on the X-axis, you're seeing that as you add independent variables, that can get worse. In fact, at this point, I should be able to shake you out of your slumber at three in the morning and you should be able to tell me this, right? So, we've got that, we're good with this. How are we going to start our model building process? Well, roughly speaking, here's how I want you to start. First off, let's start with all of the variables that might actually make sense. Your job as a data scientist is not to be an automaton. You've got to understand the context of what you're doing.

Like somebody said, "Hey, put in pop-tarts for sales predictions." Maybe that makes sense, but you better have a really darn good reason. So, start off with all of the variables that potentially make sense, that potentially make managerial sense in our model. We're going to build a model at that point, but the first thing we're going to do when we build this model is we might notice that a bunch of these variables are actually not significant. Their p-values are too high. What do you do if they're not significant? Throw them out. That's sort of like you know looking at wind speed, right? Not significant. p-value too high, throw it out. So, that's one thing we're going to do. What's the next thing we're actually going to do, right? We'll be left with significant variables, but amongst these significant variables, there maybe things that have very strong correlations. Because they have very strong correlations with each other, but that's not a good thing. And so, we want to take care of multicollinearity.

We already talked about this example of temperature, and wet bulb temperature, and how we deal with that. From things that are highly correlated with each other, pick one. And pick the one that has this highest absolute correlation with whatever your target is. And now you're kind of left over with a whole bunch of independent variables, all of which are potentially significant, none of which are kind of collinear with each other. What are you going to do next? You're going to generate this out-of-sample R

square curve. And you're going to sort of keep kind of chopping things off until this out-of-sample R square starts actually getting worse. The idea being we want to get to that Goldilocks Zone. So, that's the recipe. It's no more complex than this. And at the end of it, hopefully what you get back is a model that first off, you can trust the coefficients of the model. You know that those coefficients make sense. You know that none of them are like pop-tart or random. And two, the model is the best possible model you can build with an eye on using the model on future data, out-of-sample, just by construction.

So, to summarize what we've done in learning linear regression like we have. There's a whole bunch of takeaways over here. And as I think about it, maybe the most important thing is the quote that I started with when we talked about linear regression, which is that, even in the quote unquote age of AI, this should be the first model we try. It just has to be the first model we try. All right, that's number one. Long after you've finished thinking about this lecture, like all the little details, hopefully, you remember this, that this is the first model to look at. Number two, always have a baseline. As a data scientist, super tempting to just dive in and start building. But what are you going to compare that to?

So, always have a baseline. When we, in the course of doing linear regression, we've had an implicit baseline. That implicit baseline has simply been predicting the mean. That was this SST that we looked at all along. But the high-level thing is we always want a baseline. The R square metric has built into it a baseline where you're comparing against the model that just predicts the mean. Significance, right? We looked at p-values and confidence intervals and what was the point of that?

The point of that was that somebody threw in pop-tarts as an independent variable or more seriously, if somebody threw in wind speed, you could very well be fooled by limited data, and you don't want to be fooled by limited data. If you say that your model depends on something, you have a certain coefficient. Well, if you use the model to eventually look at predictions on new data, having that irrelevant thing in there is likely going to hurt you. And so, you want to throw away insignificant variables. Above all actually, these coefficients that you have should just make sense. And that led us to multiple linear regression, where we saw it was pretty subtle. Multiple linear regression had a whole bunch of subtleties. There was the issue with insignificant variables which I just mentioned, but there was also the issue with multicollinearity. Remember, temperature and its twin, you put both those things in there, you can't trust the coefficients on either. That's problematic.

And then finally, we said, look our eye is on actually having good predictions on future data. We don't really have future data, but how do we actually simulate that for ourselves? We simulate that for ourselves by, in essence, taking out a portion of the original dataset we have, treating that as a test dataset, treating that as our dataset of future data. And so, I genuinely believe that this is a powerful

Swiss Army knife to have at the back of our pocket if used correctly, which I think we're all in a position to do now. This is very powerful.