**Week 1 - Regression Example**

0:00  
If we first look at the documentation of linear regression on Sitekitlearn, we can see the constructor here, linear regression and the parameters.

0:10  
The option of parameters you can put in.

0:12  
You can also set it up with the blank constructor and just take the defaults of all of these here.

0:18  
Then there's attributes, some data that's related to it, number of features and so on.

0:27  
And then the example is kind of the best place I always find to start.

0:30  
So you import Numpy as usual, import your linear regression, set up your X and your Y.

0:35  
I'll do that in a notebook in a minute.

0:37  
And then you set up your linear regression.

0:39  
So normally you can kind of create your linear regression model.

0:43  
I would normally call that model as opposed to Reg.

0:46  
Then you can run your face on it X&Y and then you can calculate the score that's like your R-squared measure of the error, how good your model is at mapping at at your predictions versus your actual data.

1:00  
Then we can look at that coefficients of the module of the model, the slope and the kind of intercept and or the coefficient rather it is a slope and the intercept is the kind of MX plus C that's AC.

1:13  
And then we can do predictions based upon other data.

1:18  
So what I'll do is I'll just open up.

1:19  
I have a lot of this coded in a notebook already, and we'll just go and have a look at it.

1:28  
So I'll just rerun everything for a second.

1:35  
I'll make this available on the Moodle page.

1:37  
So first thing is step one is, as always, just important things.

1:41  
You'll always want Matlotlib, Pandas, Numpy, same as we've always done.

1:46  
And we're going to just set up.

1:47  
I like this style.

1:48  
I'm going to use it most.

1:50  
I use it most of the time for this workbook.

1:52  
I'm going to use it just to begin.

1:53  
I'm just going to take a very simple relationship.

1:56  
I'm going to just take some X values, 579 and so on, and some Y values, 1114 and so on.

2:02  
What we're going to look at is build a model where Y or our predicted values are going to be these coefficients apply to X.

2:11  
This is my FX, my function of X.

2:14  
It's going to be these two coefficients in a kind of a linear straight line model.

2:18  
Before we can see if a straight line model even applies, we have to have a look at it.

2:22  
So I set up my X here, which is this.

2:25  
I set up my Y which is this array of numbers.

2:30  
As I said, best thing to do is first plot.

2:33  
It might not even make sense.

2:34  
This could be a completely a curved model or anything, but I can see straight off.

2:37  
It's a nice linear model.

2:38  
It's not exactly maybe a straight line, the sum values off the straight line, but it looks pretty linear.

2:45  
Another way kind of to check the linearity of it or if the relationship between the two variables is to check the correlation coefficient.

2:52  
Don't worry if you don't know what that is, some of you will have seen it before.

2:56  
The correlation coefficient is coming out with the number 9932 nought .99332 where 1 is a perfect correlation, perfect positive relationship between the variables.

3:07  
Zero will be 0 correlation, so a .99 will be a very high correlation.

3:13  
Next thing we need to do is if we want to use linear regression where we have Y are outputs and X are inputs, we need to ensure that X is A2 dimensional vector.

3:25  
And in this case, we'll have a look.

3:26  
We'll see you in a moment.

3:27  
That X is A1 dimensional vector and we saw A1 dimensional input where we were looking at, say, cooking the Turkey where you have the weight predicts or is used to calculate the weight of the turkeys, used to calculate the time.

3:41  
So you've only one input, the weight 1 output the time.

3:44  
And it's the same here.

3:45  
I've got one input, which is whatever this is these numbers, and one output, which is whatever I'm trying to predict.

3:52  
So XP my input.

3:53  
But normally most real world, whether you're trying to calculate the probability of whatever, even just determine what's in an image, it's not just a single input that's passed in a single number to predict the output.

4:06  
It's normally an array of numbers, so you would have for each element in the data set, you would have an array of X values, input values, and a single Y output value.

4:19  
And then if you've got multiple observations, say like multiple images, you end up at a 2D array where one dimension is all the elements or all the observations in your data set and the other axis going across the way is all the features that you have.

4:37  
So let's just have a look down at this data structure.

4:39  
We can see that Y is A1 dimensional structure with six elements in it.

4:45  
1D is good.

4:46  
So 1D is as expected, a 1D array.

4:49  
That's good.

4:50  
If we look at X, it's going to come out also as a 1D array because that's how I defined it at the beginning.

4:54  
But that's going to throw an error.

4:55  
See it here.

4:56  
I defined it as A1 dimensional array.

4:58  
That's going to throw an error if I try try and throw that into the SK learns linear regression model.

5:06  
So what we need to do is remodify it so that it's a 2D array.

5:09  
Now on every column there's still going to be 1 value.

5:12  
But here's my X as it currently is.

5:14  
And as you can see it's written as one row across here.

5:17  
But what you really need is 6 rows and with each row there's only one column entry.

5:23  
But then at least it's a 2D array.

5:25  
So the way to reshape the array again, some of you have seen this before, is to use the reshape function.

5:30  
And now I can see after the reshape function I've got 2 dimensions, six on one axis and 1:00 on the other axis.

5:39  
And I can see here under the number of dimensions I've got 2 in the dimensions which is a 2D array.

5:43  
And if I look at it very similar to what I had before, but instead of one row, I now have 6 rows and in each row I only have one column.

5:54  
Now, if that was a 2D array with multiple observations or multiple features to predict the output, there will be multiple numbers in here for we'll see down in a minute the diabetes data set.

6:05  
I'm about to show you that there's multiple features like BMI and some blood work and so on that can produce the likelihood or that can provide information on the likelihood of someone developing further complications with diabetes.

6:18  
So building the model, same as we saw with the documentation, first we have to import linear regression from SK Learn.

6:24  
I can query the linear regression documentation by just putting in the question mark.

6:28  
I'm not sure if you've seen that before, but I've shown you can see how the linear regression, there's some parameters you can you can provide to set up the model, but I always just take the default.

6:39  
Most of the time I take these default values and you'll see here how I just create the model with an empty constructor, which is just the as I said the default values it provides for all of those parameters within the model.

6:52  
Next thing I have to do is fit my model providing X my input.

6:57  
I've been using capital X all the way through there.

6:59  
That should be a capital X because it's a 2D vector and Y the 1D vector.

7:07  
So the dot fish uses a gradient descent and we'll talk that about expects X to be a matrix.

7:12  
We have A1 dimensional array.

7:13  
So I've already, if you just put in the original, that's what I did initially is I put in the original X before I reshaped it and I got an error.

7:20  
So I have to go back up and do that section I just discussed.

7:23  
It returns me just this linear regression model, but nothing has been done with it yet.

7:27  
It's just a linear model.

7:29  
We can query it in the next section and we can also kind of query, you know, what are the coefficients of the model, the W zero and the West one.

7:37  
But we can also query what's the performance of the error on the model.

7:43  
So the model coefficient, which is the slope of the model is 2.12.

7:47  
The intercept, the Y intercept which is in Y is equal to MX plus C is where it intercepts the Y axis is 0.21.

7:54  
So these are the parameters of my equation of a line where predicted values is naughty .21 Y intercept +2 slope of 2 \* X That's my model.

8:07  
So what I can do is I can actually build a module model manually.

8:11  
Now, you normally wouldn't do this, but just to show you that the model, the linear regression is actually a very simple model with two coefficients.

8:17  
The predictions are W0, which is my constant plus MX.

8:21  
So what I'm doing here is multiplying W1, my weight not multiplied by X.

8:28  
So there's a multiplication.

8:29  
This is your Y is equal to MX.

8:31  
Here it's been MX plus C and I've you can list my predictions here.

8:37  
So this is one way of actually manually building the model.

8:39  
But what's nice about SK Learn and about Python in general is there's loads of built in functions that do all the heavy lifting for you, especially when it comes to multiple coefficients and multiple inputs or features.

8:52  
So this is a more standard way of doing it.

8:54  
You just put in model dot predict and pass in your X values and it'll give your Y hash or your predictions.

9:00  
So my predictions and as you can see here, the predictions by building the model myself versus the predictions by using SK learns built in predict model are exactly the same.

9:09  
If you look at these two, those two numbers, I can actually do it nicely here with Python where I can just query, are these predictions from the model equal to the model I built?

9:19  
Are they the same?

9:20  
And you can see they're all true.

9:22  
The next thing then is evaluating, as I said, the error and the quality of the model.

9:27  
So actually in my, one of the things I've been having issues with in my ATU laptop is like, I can't for some reason import root mean squared error.

9:35  
I have to import, I can import mean squared error and I can import the R-squared score or two scores.

9:40  
So people call it through the r ^2.

9:42  
So what I had to do is count that out.

9:44  
But it might work for you on your laptops.

9:45  
It probably will because yours isn't restricted.

9:48  
My laptop's heavily restricted for because of security breaches in third level sector in Ireland.

9:55  
Anyways, that's another story.

9:56  
So here I can look at my mean squared error.

9:59  
I can see there it's nought .72 and my R-squared, which is nought .98, which is a really good it shows up in model is is really, really good, very, very close to 1.

10:10  
So again, this shows that my model, my predictions are very close to the actual values of Y.

10:18  
Again, this is a manual way of calculating you know, you can use the OR two function, but SK learn again has built in.

10:26  
If I look at the root mean squared, I can see here it has a null .85.

10:32  
This is the, we'll be looking at these, all these error metrics and the various ways of measuring the quality of your model in upcoming lectures, in upcoming weeks.

10:41  
The only thing we can do, we'll just have a look, we'll come back to the R-squared again, Naughty .9864.

10:45  
But the other thing you can do is you can actually visualise the model and we can see here versus the data I provided and the model, they're a very close match and you can see where the sum of the errors are in the predictions.

10:56  
And that's accounting for it not being a perfect one.

10:59  
We're getting this snow .9864.

11:00  
Again, there's a built in function in SK Learn.

11:03  
You just do model dot score and you can see here in 9864, which just shows you that the under the hood scoring the model in SK Learn is using the R-squared metric.

11:13  
You can change that metric, I think in this parameter setting you can change it, but by default the R-squared metric.

11:18  
And again, we're going to look at these metrics in the coming weeks.

11:24  
What's the best way to and what do these, how are these metrics actually calculated?

11:30  
Next thing is I might just ask you to do is to run your own if you're interested, simple linear regression with some automobile data.

11:37  
So I've downloaded this car data set which has got, you can predict the miles per gallon of a car versus the features of the car.

11:49  
So you can read in the CSV, we can read in the different columns and we can look at the head of the car and it's got all the different models.

11:57  
It's got the models per gallon for each model.

11:59  
And then you can predict that maybe based upon some of the other things, I think I'm only guessing your displacement horsepower, I don't know, is this seconds to complete a quarter mile and various different things.

12:11  
So one of the things you could do is you could find out what which of these other features is the best predictor of miles per gallon for a particular car.

12:19  
Just could be an interesting task.

12:21  
So what you need to do is choose the variables you think would be good predictors.

12:26  
Look up what the variables are.

12:27  
You know, is it number of gears or whatever it is, create a mark and you can discuss.

12:33  
You can just have a look at it if you're interested.

12:35  
So you can put in some of the code, measure the error, create the model, make some predictions.

12:41  
I'll leave you to do that though, if you're interested, like I said.

12:44  
What I'm going to do though, just to show you a bit of multiple linear regression is that like I said previously, in the real world generally one feature cannot be used to predict the output in the same way as friends.

12:58  
Number of friends on a social media platform versus number of minutes spent online.

13:03  
That's not necessarily a very good fit between those because there's lots more features that will you might need to make a good prediction.

13:11  
So here what I'm going to do is just import the diabetes data set, which is already built into SKLR and which is very, very handy.

13:16  
And we can use and do a bit of multiple linear regression.

13:20  
So here I'm just importing the data set.

13:21  
I'm printing out a bit of a description of the data set tells me what's in it so that there's 442 data points or instances in the data set.

13:30  
There are 10 numeric predictive values and 11:00 then is a, which is our output.

13:37  
We're trying to measure the disease progression one year after the baseline, after we've taken this these measures.

13:45  
So there's age, sex, BMI, blood pressure and there's so on, S1S2 and so on are related.

13:51  
I would guess these are from blood work, total serum cholesterol, lipoproteins.

13:57  
They're probably some chemical tests that are done on the person's blood or on some biological samples.

14:03  
Other things are measured like blood pressure, BMI, and so on.

14:08  
More information on the data set right here.

14:11  
So what I'm going to do is you've probably seen this before is read straight into Pandas, which is just so useful for doing this kind of stuff and do a box plot.

14:18  
And I'm not sure how much statistics you've done, but I'll just really quickly go through a box plot.

14:22  
It's a very useful way of visualising information.

14:25  
I'm just looking at the head of the data set where I can see some of these things.

14:28  
But one of the things that I'll jump out and you can also see it from the box plot is that first thing is all the data is very, very similar.

14:35  
So all the data seems to be centred around 0, all these Bok plots.

14:38  
So each of these little blue lines here is the medium.

14:41  
It's not actually the mean.

14:42  
You can look up what the median is if you're interested.

14:45  
These red bars here are the inter quartile ranges.

14:48  
It's like the 25th percentile and the 75th percentile.

14:52  
These up here are known as the fences.

14:53  
And then you've got some outliers and you've got really kind of skewed data and data that's outside of these fences.

14:59  
I'm not going to talk too much about box plots or statistics at this point in time, but this is a box.

15:05  
But that shows the spread.

15:06  
Each of these things shows a measure of spread of each of the variables.

15:10  
So BMI is spread between this value here and this value here.

15:13  
Now the thing about BMI actually can't remember what scale BMI is measured over what kind of numbers there is, but I definitely know age doesn't go from minus .1 to plus .1.

15:26  
And I'm guess that as well, well sex as well doesn't go from this value to that value.

15:31  
So what is going on, and you can see it here, there's some, there seems to be discrete values, point O 5 or negative point O 4.

15:39  
So what happens is that in order to make all the data correspond to the same sort of ranges, often what people do is a very first step with a data set is they do something called normalising the data set.

15:51  
And normalising the data set is scaling all the values so that they're all within similar margins.

15:58  
I'll talk more about why they do this, but it's basically to ensure that one variable doesn't have a really large influence.

16:04  
That if all the values here were in the range of .2 to -.2, we had a value that was between 0 and 100.

16:10  
Like that's, you know, multiple orders of magnitude bigger than all the others.

16:15  
And it's going to have a really skewed effect.

16:16  
So it's best to keep all your samples, what we're talking about, again, within certain ranges in order that one of these features.

16:23  
Imagine you had a really, really tiny feature that was like milli or like micro 10 to the -6 here, even though it might contain really significant information, because it's so, so tiny, the numbers would be so, so tiny.

16:36  
The numbers would be lost in the algorithm.

16:38  
The algorithm wouldn't just detect those tiny numbers.

16:40  
So again, it's good to have everything normalised.

16:44  
Put everything in the data's frame, visualise it.

16:46  
It's all in a similar range.

16:49  
Or I just talked about the normalised.

16:50  
All of them mean a zero and a standard deviation of one, a very common step.

16:55  
We'll talk about again the E.

16:56  
The best example there is what age or sex, which has got two options when either you go with plus 1 -, 1 or 011 or whatever.

17:04  
So all of these numbers were changed.

17:06  
You know, these aren't the real numbers of S1 or S2.

17:09  
They're all changed because of the normalisation.

17:12  
X is the data that's going in all of this data which I've segmented out here.

17:18  
Data is the data frame diabetes dot data with the feature names.

17:22  
Why is diabetes dot target?

17:25  
I again build my model with an empty constructor with the defaults, run my fish pass in my inputs and my measured outputs, and then I can look at the scores.

17:37  
So I can look at, you know, what's the score of the model here?

17:41  
What's the R-squared measure in this case?

17:43  
Not as good as that perfect linear relationship we had before, means it's a bit more difficult to predict, but there still is correlations in there.

17:53  
And what I can do, which is interesting, is look at the actual coefficients.

17:56  
So if I see here and I'll see them in a graphical format a bit more, some of them are quite small and some of them are quite large.

18:03  
And what that means, the intercept isn't survivable.

18:06  
But what that means is that see this one here, which relates to age.

18:10  
That very first column is the age column.

18:13  
Because of such a tiny coefficient, it's scaled by only a tiny amount.

18:16  
It probably means that age isn't a very good predictor of the output of this measure of the progression of the disease, and some of these are better predictions.

18:26  
So sex would be a marginally better predictor.

18:29  
But this S1, whatever this blood work is, has got a very high influence on the output these coefficients show on.

18:36  
Similarly, the S5S3 doesn't seem to have so much of an influence.

18:41  
Some of the coefficients are very large.

18:43  
Looks like age doesn't contribute as much.

18:45  
Sometimes the large coefficients are problematic.

18:47  
We'll talk about that again.

18:49  
It's coming up in the next few weeks.

18:50  
Often removing these insignificant ones makes the model smaller and can actually improve the performance of the module.

18:58  
But we don't have, at this point in time, enough knowledge to figure it out.

19:01  
It would involve stripping some of these things out of the model and using various combinations.

19:05  
You know, maybe only these three features by itself might give you a better model.

19:09  
So this is the kind of work you do as a machine learning in the real world as you'd be looking at which of these features together in the model really give the best predictive performance of the outbush.

19:21  
You know, it's possible to get a better model without age.

19:24  
S3S4 and S6 strip all that out.

19:27  
You could actually get better performance and it's something you can have a look at yourselves.

19:30  
With pandas, you should have enough skill to strip out different columns, strip out these columns, refeed it all back into the model, and refish the model.

19:40  
And again, you can look at your outputs to see if you can beat this score, your R-squared score of .5 O at the moment.

19:47  
OK, so that's just a really quick overview of linear regression.

19:50  
Some of you might have seen it before in other modules, but this is just another another refresher on it.

19:58  
If if that's the case, OK, talk to you all soon.