Linear Regression

Udacity

What is a Linear Equation?

Equation of a line : $y = mx + b$, where m is the slope of the line and $(0, b)$ is the y-intercept. Notice that the degree of this equation is 1. In higher dimensions when we talk about linear equations we are talking about degree one equations. For example: $z = 5x - 3y$. Note that, this is a linear equation and it is equation of a plane and **not a line**.

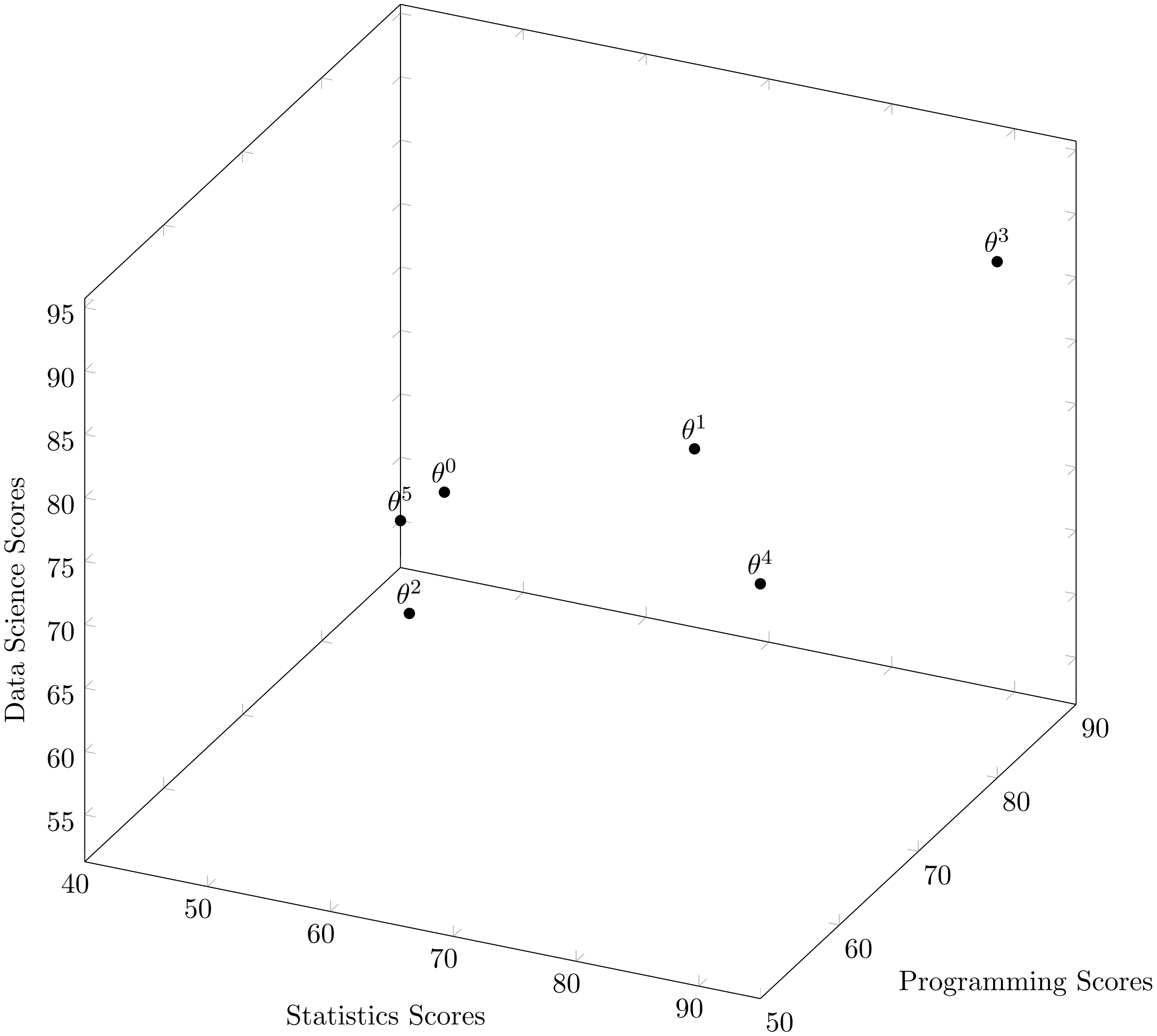
What is Regression Analysis?

Regression Analysis is a concept from Statistics. The idea here is to observe data and construct an equation so that we can make predictions for the missing data or future data.

Let's look at the following data showing the test scores (out of 100) from Statistics, Programming and Data Science courses.

Name	Statistics	Programming	Data Science
A	50	80	65
B	80	65	83
C	60	60	69
D	95	80	92
E	95	50	84
F	40	90	55

Let us try to model this data. That way, next time we meet a student with known Statistics and Programming scores, we can predict their Data Science score. This will also be useful for students who want to know how much they should focus on each of the topics in order to do well in Data Science.



What is Linear Regression?

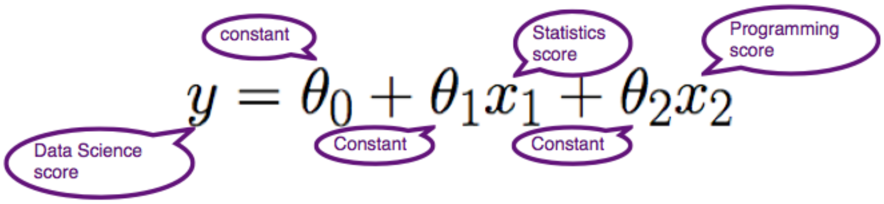
Linear regression is an attempt to model a linear relationship between a **dependent variable** (y) and **independent variables** ($x_1, x_2, \ldots, x_n$).

In other words, we want to write an equation of the type:

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

So here y is the **output variable**, $x_1, x_2, \ldots, x_n$ are **input variables** and $\theta_0, \theta_1, \ldots, \theta_n$ are called the **parameters** or the **weights** of our model.

So in the above scores dataset, y is the Data Science score (the value we want to predict). x_1 is the Statistics score and x_2 is the Programming score. Equation would look like,



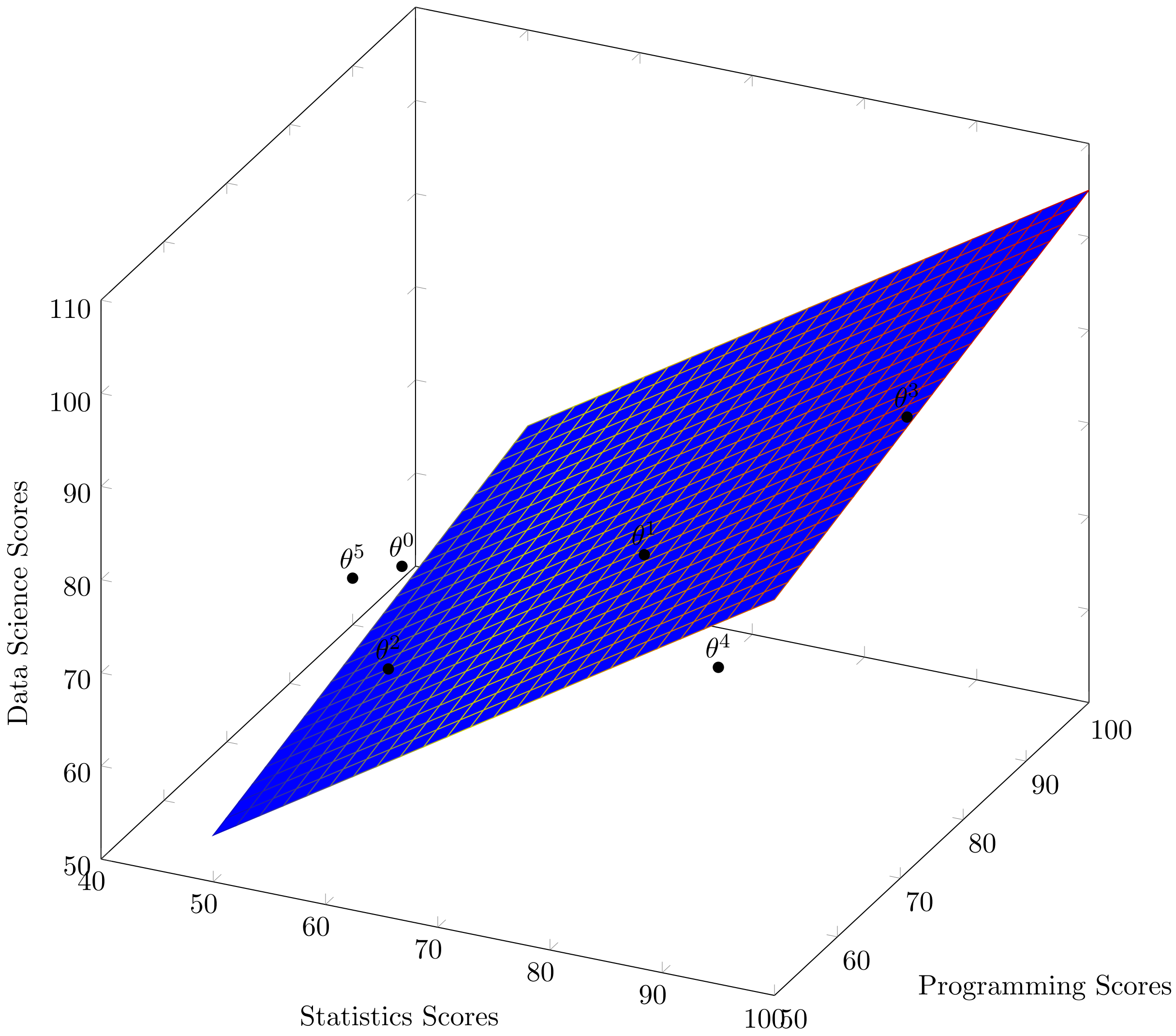
$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

Why are these θ's called weights?

Simply because each θ tells us how important the corresponding x is in predicting the output. This means, if a particular θ-value is very small compared to others, the corresponding x plays a little role in predicting the output.

For instance, Suppose the scores equation turns out to be $y = 5 + \frac{3}{4}x_1 + \frac{1}{4}x_2$. Notice the weight on Statistics is $\frac{3}{4}$ and that on Programming is $\frac{1}{4}$.This means that students should focus a little more on statistics if they want to do well in Data Science.

Note: We will learn how to find this equation later.



Whats Error?

It's great when we can accurately predict the output. However, in real world, that is not always the case. Our data may not perfectly fit-in with our linear model. Which leads to error in our model.

Let us say $\hat{y}$ is the predicted output, i.e. output we get using the above linear equation.

There are different ways of calculating error. Two of the most standard ones are:

Sum of Absolute Errors $\sum_{i=1}^{m} |\hat{y}_i - y_i|$

Sum of Squared Errors $\frac{1}{2} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2$ (we multiply by $\frac{1}{2}$ for computational convenience, we will be taking derivative of this expression later on in Gradient Descent Algorithm and the exponent 2 will cancel the 2 in the denominator)

Why does Gradient Descent work with Sum Of Squared Errors? Remember that, gradient descent algorithm uses the derivative of the function to be minimized. Squaring the differences makes this error function differentiable i.e. we can find the derivative of this function easily.

Let us consider our scores example again. Assume that $y = 5 + \frac{3}{4}x_1 + \frac{1}{4}x_2$ is the model equation that we choose to work with. Following is the table of predicted and observed values.

Name	Statistics	Programming	Observed Data Science Scored	Predicted Data Science Scores
A	50	80	65	62.5
B	80	65	83	81.25
C	60	60	69	65
D	95	80	92	96.25
E	95	50	84	88.75
F	40	90	55	57.5

Then, sum of absolute errors = 19.75 and sum of squared errors = 36.09375.

What do we do with the error?

The aim is to minimize the error. In other words, we want to find the values of θ that give us minimum possible error. This is what the Sum of Squared Errors looks like in linear models. Predicted value $\hat{y}_i = \theta_0 x_0^i + \theta_1 x_1^i + \cdots + \theta_n x_n^i = \sum_{j=0}^{n} \theta_j x_j^i$

$$\begin{aligned} \text{Sum of Squared Errors} &= \frac{1}{2} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2 \\ &= \frac{1}{2} \sum_{i=1}^{m} \left(\sum_{j=0}^{n} \theta_j x_j^i - y_i \right)^2 \end{aligned}$$

Once again, let's go back to our student scores example and see what this would look like when $y = \theta_1 x_1 + \theta_2 x_2$.

$$\begin{aligned}\text{Sum of Squared Errors} = \frac{1}{2}[&(65 - 50\theta_1 - 80\theta_2)^2 + (83 - 80\theta_1 - 65\theta_2)^2 + (69 - 60\theta_1 - 60\theta_2)^2 \\ &+ (92 - 95\theta_1 - 80\theta_2)^2 + (84 - 95\theta_1 - 50\theta_2)^2 + (55 - 40\theta_1 - 90\theta_2)^2]\end{aligned}$$

Our goal is to find the values of θ that minimize the above sum of squared errors. One of the common approach is to use calculus. This is where the gradient descent algorithm comes in handy. Also notice, how easy it is to take a derivative of this error function. So take a good look at the gradient descent algorithm document and come back here to find the linear equation that fits our data.