4.1.0 Measurement

Hi, today we're going to be talking about some of the challenges and principles when measuring big data.

So, to begin with, we're going to look at accuracy versus precision. Now, accuracy is defined as how close to the truth is the data and precision refers to whether the data can be repeated, given the same set of circumstances, or exactly the same testing. So, to put this in context, the easiest way to think about this is if you were to look at, say, a target for something like archery, let's take a look here.

If you were to shoot six arrows at said target, and all six of those arrows go right slap bang in the middle right in the bullseye, as we've got here, this would be referred to as high accuracy. Because it's right in the centre, it's so close to truth. And it's also high precision, because the grouping of the data or the grouping of those arrows, is so close together, which means that it's repeatable, because you did it six times in a row.

So, keeping with that theme, let's take a look at the exact opposite. If you were to fire those same six arrows, and they scattered all over the target, you would have what we would refer to as low accuracy and low precision. Now, these can also exist one at a time. So, it's not that you have one or the other completely.

So, if we were to look at something like this example, where we've got all of our arrows clustered together, but they’re nowhere near the centre of the target, this would mean that we have low accuracy, because we're way off target. But we do have high precision in that it can be repeated.

And similarly, if we can get all of our arrows to be near the target, but not very close together, then now we're at high accuracy and low precision.

So, what does this mean when we apply it to big data?

The real relevance to this is that people tend to confuse accuracy and precision. So, they'll sit there and say “Well, I can replicate my results, my results must be accurate, because I get the same result every time I do this test”. That's not necessarily true if the test is flawed.

So moving on. Now, when we take our data sets, and we now understand, we can now apply some theory around whether those data sets are accurate, precise, or both.

When we're looking at these, it's very, very important that we don't discount our highest and lowest values. The highest and lowest values in a data set can sometimes lead to unforeseen insights, they can lead to new questions, they can lead to new lines of thinking, and they can lead us to conclusions about our data that otherwise we may not have seen.

If we look at a quick example of this, if we took a data set of 1000 Records, and the highest value that we can see, is 500, so the highest value that appears in our thousand records is the number 500. But that number of 500 appears in 50% of all the records we have. If we then look at all of the other records, the other 500 records that exist, we can see that they're evenly spaced, so they're going 12345, etc, all the way up to 500. And then the following 500 are all the same figure.

This, knowing this maximum value in this case of 500 would lead us to ask some questions. Does it mean that the maximum value achievable was 500? So was this something like a test for argument's sake that somebody took and the maximum score that you could attain was 500? What does it mean that all the values that were over 500 were just recorded as 500? Does it mean that the scale that they were using only went up to 500, even if somebody was to attain or measure 501.

This gives us a bit of insight now into how the data may have been gathered, what kind of equipment might have been used or what kind of census might have been taken or you know whether it was a test that somebody took and it can possibly give us some insight into the accuracy of the data and leads To a bit of a conclusion now to say, we might want to go back and learn a little bit more about where this data came from, before we take a, you know, go looking for trends, because it might be flawed. So now that we know a bit about accuracy, we know a little bit about precision. And we know a little bit about understanding the range and the highs and the lows in our data values.

The next thing that we come on to, which is a real challenge when looking at Big Data is counting. Now I know it sounds strange, because counting is something that people do naturally, we all learned it in school, we learned at a young age, we know how to count, right 123. But actually, counting can get a lot more complex when you start trying to apply it to data sets and we actually need to define rules for what we mean when we say counting and these rules will be dictated by your need, so they'll be dictated by what you're trying to achieve with your analysis.

Let's just take another look at an example.

So very simply, we've got here a website URL. And the question being posed is how many words are contained within this URL. At first glance, that seems like a very easy task. It's one word, it's one thing. But actually, if we set the rules slightly differently, and we apply it to a few different scenarios, let's start with just removing the separators, so we're going to remove anything that is not an alphabet character, and everything else becomes a word.

So, we're going to remove the separators. We're removing the colons, the forward slashes, the full stops. And all of a sudden, we have nine words as we can see there in the example. So just removing the separators, we ended up with nine words. But what if we said we're only going to count words that appear in the dictionary, and we'll separate everything else out, and we'll remove it when we could take this to eight words and we do that by separating the top 10 tips for freshers into its component words. Now Actually, this could equally end up being four words.

If we didn't separate those top 10 tips for freshers is a single word doesn't appear in the dictionary. So, we could end up bringing this right the way down to three or four words. And if we set our rules as for it to be a word, it has to be separated by a space, then we're back to this idea that this whole URL counts as a single word.

So if you apply this or if you think about this in terms of a big data analysis, if you have a lot of qualitative data, a lot of somebody's giving you typed out paragraphs for argument's sake, or you're trying to ingest news articles, or research papers, how many words are in there, how many lines, how many paragraphs etc. This becomes quite complicated to count if you don't have a robust set of rules prior to doing your analysis. And this can get even more complicated if we start adding things like numbers, or different symbols. Have a think about this in terms of an email address.

So when we're bringing together data, and when we're putting all of our data sources into a single data set that we're then going to use for our analysis, there's some things that need to be considered.

Not all data can be compared directly. So to give you a good example of this, let's say we ask a question on a test. And some people decide to answer this question with an explanation. And other people decide to answer this question with a single word. Other people might decide to answer this question in the form of a number.

All of a sudden, we could have three different completely different types of data that we're now going to have to assess or we're not going to have to analyse all together in a single place. Now it can be done, the process for doing this is called normalising or normalisation. And what we have to do is break it all down into a single format that can then be compared against itself, so that we can compare these sets against each other.

To give you an example of this, we're going to have a little look at the kind of sources that would feed into a unified health record. Okay, so let's assume we have a health record, I'll use myself as an example, I go to my doctor, and I expect my doctor to have certain information about me. I expect them to be able to have some records of symptoms that I may have displayed in the past. And I'd also expect them to be able to take new symptoms from me when I visit the doctor. I'd expect them to know information about any diagnosis that they carried out. But these two things may not be in the same system. But they have to be brought together for a single record. I'd want them to know about treatments that they've given me in the past. I want full details of any testing and the results that came back as raw data. Separate from the conclusions that they drew to create the diagnosis; they might have stored some information about my family history. Aside from that, they may talk to my employer about work related injuries, I may have given the information about that. But actually, my work, my employer is going to keep details of any injuries I suffered at work separately to this, but I'd want my GP to be able to access those and to be able to use them when they're analysing my health. And again, live Vital Statistics. If I end up being admitted to a hospital, I'm going to want that hospital to be able to feed in what my Vital Statistics look like live so that they can monitor me, and so they can compare it to my previous records.

Now, this is just a selection of some of the sources of data that might go into a health record and actually, all of these statistics, all of this information should then be brought together for everybody's health record, so that we can then create a data set that will allow us to do things like creating trends of people's work environments versus their work injuries, people's lifestyles, versus their developed ailments, people's family histories, so that we can discover whether or not ailments are hereditary. And bringing all of those records together effectively means we're taking lots of different sets of data, compiling them into one record, and then taking lots of different records to compile them into one data set.

The way that my doctor stores data and processes information may not be the same as the way that your doctor processes and stores information, which means even bringing those records together, once they've been normalised for this process, they'll have to be another normalisation before we can actually go ahead and do anything with that information.

So, what are some of the techniques that that you need to be aware of and some of the challenges that that we face when we're trying to normalise data? One of the biggest ones, and one of the most obvious ones here is adjusting for changes through time. It is very, very rare. That analysis of data will happen at the same time as it is collected. We'll have a collection phase, which may be ongoing. And then once we have enough data, we can start to analyse it.

What if this data that we were collecting, though, happened and this collection happened? Quite a while quite a long time before the analysis happened. Let's take a look at something like population data. If we were to sit there and say, Okay, this town has is 50% male 50% female. But that 50/50 outcome was recorded in 1920 when there was only 3000 people living in the town. Now, we're 100 years later, that town has become a city and all of a sudden there's a few million people living there is our data still reflective? on our analysis, they're reflective of the current situation. We may need to adjust our techniques we need to adjust our datasets even and apply different methodologies to that data set as we normalise it to allow for those kinds of changes.

Another one is dimensionless rendering. So sometimes you just cannot compare two Things to each other. So, one good example of this is images. So, if we took two photographs, and we just wanted to compare those two photographs to each other, that's not very easy to do. And if we then said, I've got a million photographs, and I want to compare a million photographs with each other, and I just want to see what the similarities are. I'm not looking for anything specific. I'm just looking to see what the similarities are.

One way to deal with this is to change that image into something like a histogram, which simply outputs a representation of how many pixels there are in each shade or each colour. Now even then, you end up with what looks like a bar graph. And they can be hard to compare, especially if one of those images was taken on a Polaroid camera, and the other one was taken on and, you know, 4k high definition, new digital camera, you will end up with far more pixels on one than you did the other. So, what we can do is we can take the dimension out of that we can take all of the information, and all of the specifics away from those individual instances and convert everything down to a fraction. So, we can just say how many blues are there, there are x percentage of blue in this photo. And taking it down to that kind of fraction or percentage-based means that there's now it is purely a numeric value, which has taken all other dimension away from it. And now we can compare those two things to a degree having normalised values.

Another useful technique is data type conversion. So, there's a couple of very good examples here. One is human readable time to a Unix timestamp. Computer's, at their heart, data processing units at their heart are big calculators, they don't deal very well with saying it's the first of January 1970. But they do deal well, with Unix epoch with first of January 1970, being zero, and then counting every second thereafter, to give you a unique timestamp every second, I mean actually breaks down to every microsecond. But those kinds of timestamps mean that we can now take our human readable dates and times and compare them very quickly and at scale.

Another use for data type conversion, would be taking addresses and converting them to longitude and latitude. A written address in a human readable format across four or five lines is very, very hard to parse, it's very, very hard to compare. But if we convert that to a longitude and latitude reading, instead, we can now do distance and we can do time of travel, we can do exactly where it is spherically, we can start to compare it to the movements of the sun, you know, whether it was day, whether it's night, you know, those kind of things, just by changing that data type.

We can also look at adjustments of scale. So sometimes when you're dealing with very broad data, and you're dealing with things that where the range of data may be from zero, up to a million, there can be a whole lot of variants in there, and a lot of subtleties. But if we're just trying to look for something like a trendline, then we can bring that down to a zero to one interval where everything becomes a for a decimal point, and everything becomes a decimal representation between zero and one. And that allows us to handle that data a lot quicker, it means the computer is processing a lot less, it means that we have to deal with a lot less variation. And it means that our processing code can be a lot simpler.

Another technique that's used in normalisation is weighting. Now weighting allows you to give some values more or less importance than others. So, let's say we took a set a data set that contained information of people have a range of ages. So, a range of ages from zero to let's say 20. And we were trying to compare it to another data set, which has a zero of ages zero to 10. And another data set that had a range of ages from zero to 30. There's only one range within all three data sets that is common, that's the zero to 10. But does that mean that we discard everything that happened to everybody that was between 10 and 30? In the other two data sets? Or should we just apply a weighting to the results, meaning that we can give more importance to common results and thus allow for an equilibrium to be found or a norm or a normalisation of the importance of that data.

Finally, here we have data reduction. So big data bytes definition has a lot of data available. But actually, not all of that data is going to be useful. So, getting rid of any unrequired fields, any unrequired sources, breaking down some of the metadata. And another technique in here would be if data sets are the same. So if you are looking at, I don't know gravitational pulls across the universe, and you then break that down into galaxies. Certain galaxies that have the same type of sun, that have the same number of planets roughly the same distances, they will exhibit exactly the same behaviours. And if we can compare those datasets, and rule out the fact that they are different, then perhaps we can get rid of one of those because it's a redundant data set. We're duplicating our efforts, we're comparing everything against something that we already know. So, we can drop one of those data sets and just use the one that is the initial representation for something that we know has been duplicated.

So that concludes measurements, and how we will measure big data.

So next up, we'll be talking about speed, scalability and large collections and I will see you all then.