**Machine Learning**

**Essentially Machine learning is putting all factors in one view and finding the patterns through trial and error - > we couldn’t handle the complexity of this view, but a computer can.**

Machine learning is about predicting and categorising

Its maths but at higher levels of complexity than human brains can handle .

**With prediction:**

On the y axis you have this thing you are trying to predict

On the x axis you have multiple data points that the machine uses to assess the likelihood of the one outcome.

Prediction technique is still linear regression – its just over a bunch more data points.

**EG: Trying to predict customer saving behaviours:**

Y axis = How much is a customer likely to save each month

X axis = what is their job, savigns history, age, no of accounts and credit cards etc (ie lots of factors that the human eye could not take in in a glance.

It draws its linear regression line through masses of different data points

The way it does this is by training itself. – finding patterns that spot correlations – again that are beyond the capacity of the human mind.

**Preprocessing is very important for machine learning.**

Need to standardise the data before we feed it into the algorithm.

**A linear regression model using Health data (see Jupyter file)**

‘y’ – is one column in the data

‘X’ – is all other columns.

Apply model and check the results

Model has drawn a line of regression. Now we look at how accurate it is and try to ‘tune’ the code to make its more accurate.

What does accuracy mean? – it means how much data is close to the line.   
re\_score tells us this as a percentage.

Now. How do we pre-process the data to make it more readable/more regular for a patterned overlay?

**What is linear regression good for:** this is a good example. Working out an expected standard correlation that would not vary. – so ‘given these factors how likely is this outcome’   
Less suitable for a seasonality changing line. Or an exponentially growing line.

**Pre-processing / cleaning the data**

**Limit the number of correlations:** data sets you input that are too similar – otherwise you are biasing the results towards one very similar set of values. IC 1,2,3 & 4 are too similar – probably should just pick one.

Standardise patterns with re-scaling. – What is important in the data is the shape (histogram) not the actual values. – Machine is comparing percentages and amounts and other things so it looks at the position of the data point – not the data. (eg: doesn’t matter if value is 6 or 8 when machine is comparing to a list of percentages. – what matters is that it is somewhere around a percentage point of 7%.   
Hard to understand but headline is: **Need to make histograms look more like a normalised distribution by re-scaling)**

**Drop the outliers?** – one techniue

**The logarithmic scale**

A scale that compresses the gaps on the histogram. So the extreme values are closer to the bulk of values. –

*All the time the histogram is getting more ‘normalised’ – more regular looking.*

**Dealing with Categorical Data**

Split up your data into categorical and numerical data first

**Standardscaler Options (see python doc)**

– normalises the data. -gets rif of outliers. Auto scales   
ALWAXS – check the effectiveness of the sccalers by looking at the histogram.