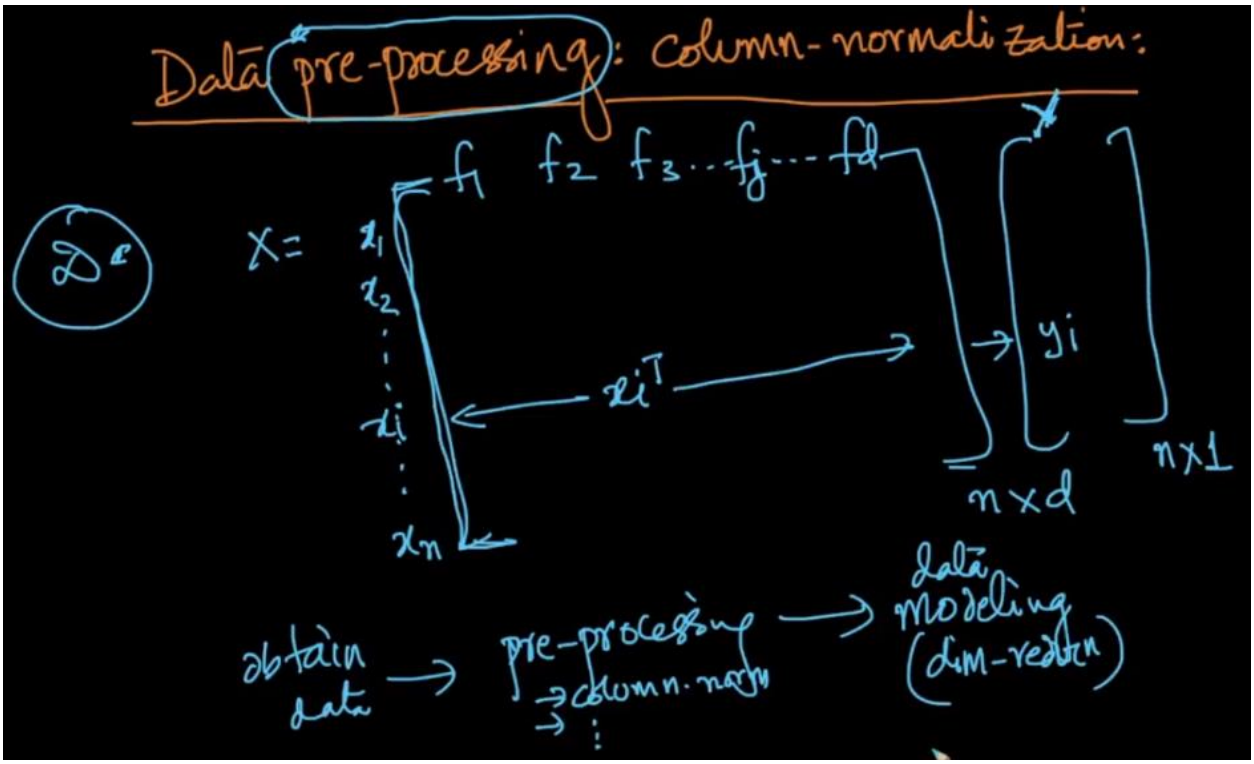


Data PreProcessing:

It's very important thing to do after obtaining all the required data. In this process we apply some mathematical operations and transformations to bring data in suitable state. The next step after data preprocessing is data modeling(dimensionality reduction)

One such process of Data preprocessing is **Feature/Column Normalization** which brings data between [0-1].

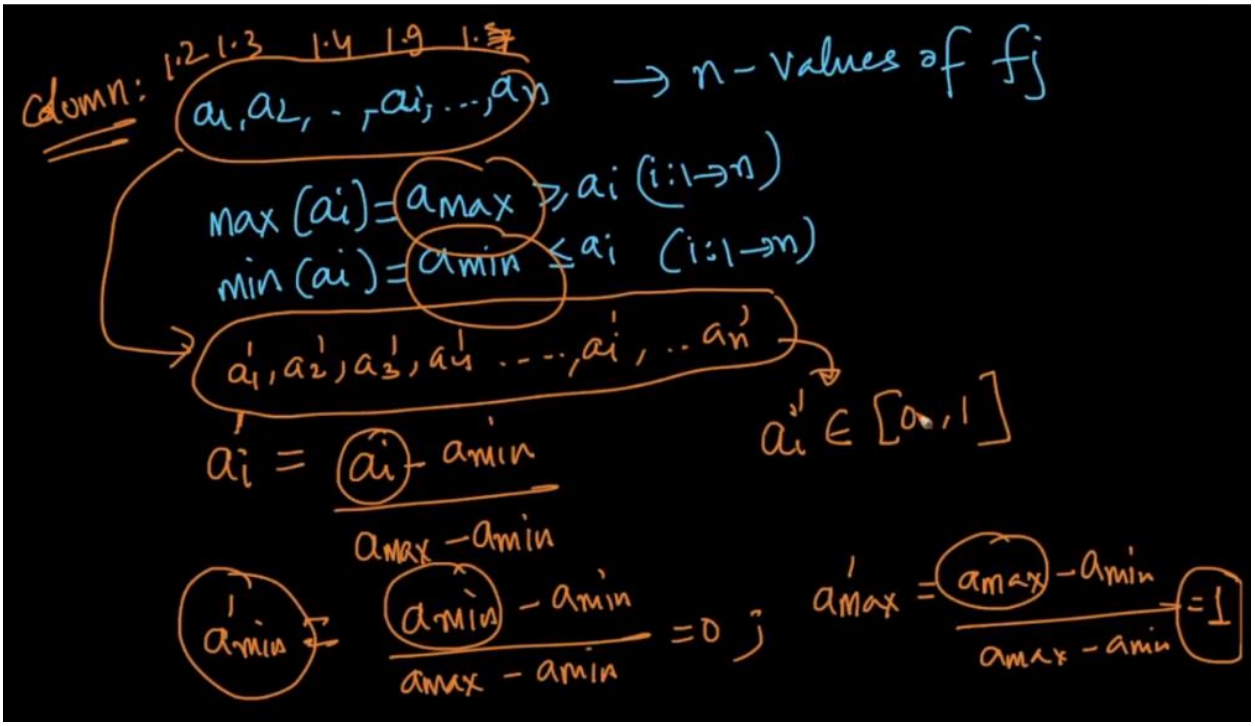
Let's take Example of iris, we have 4 features, and we do normalization of each feature.

How we do normalization?

- Pick each feature (like SL), then find max and min in that feature.
- Now apply formula (mentioned in below image) for each value/datapoint (a_i)

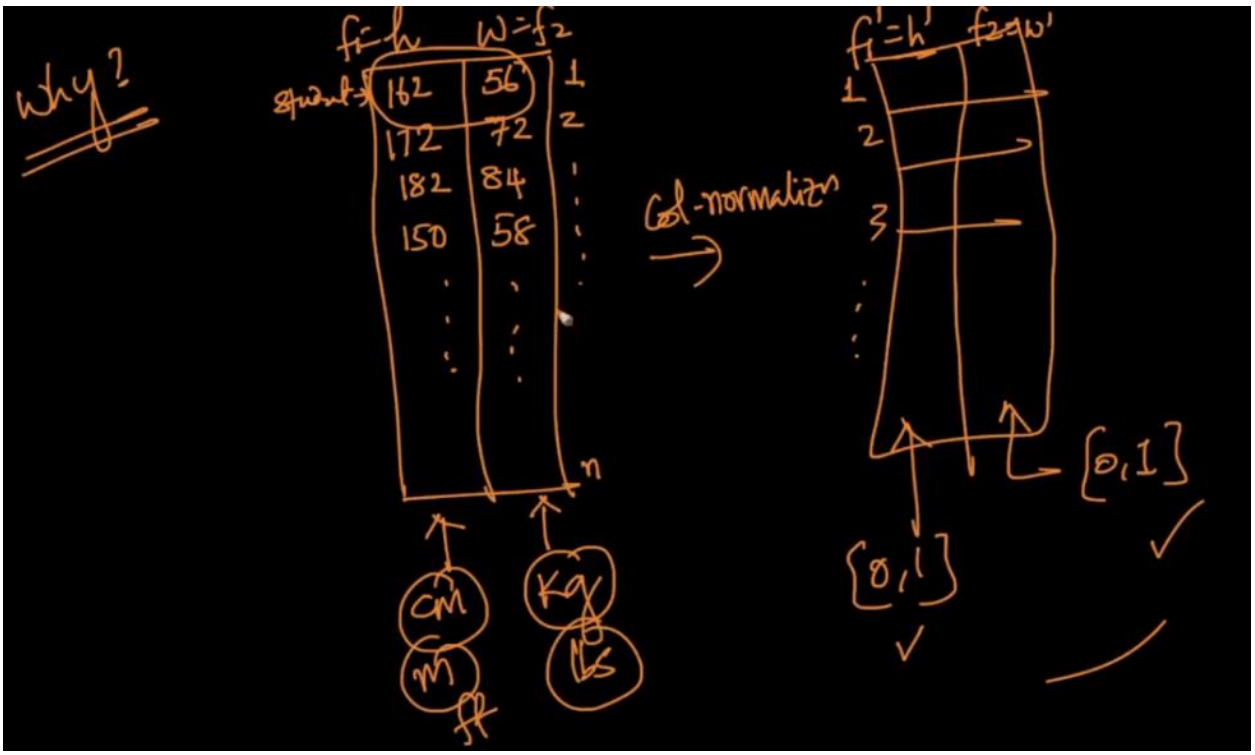
Why do we use this formula:

Because for min value it gives 0 and for max value it gives 1 as mentioned in below image.

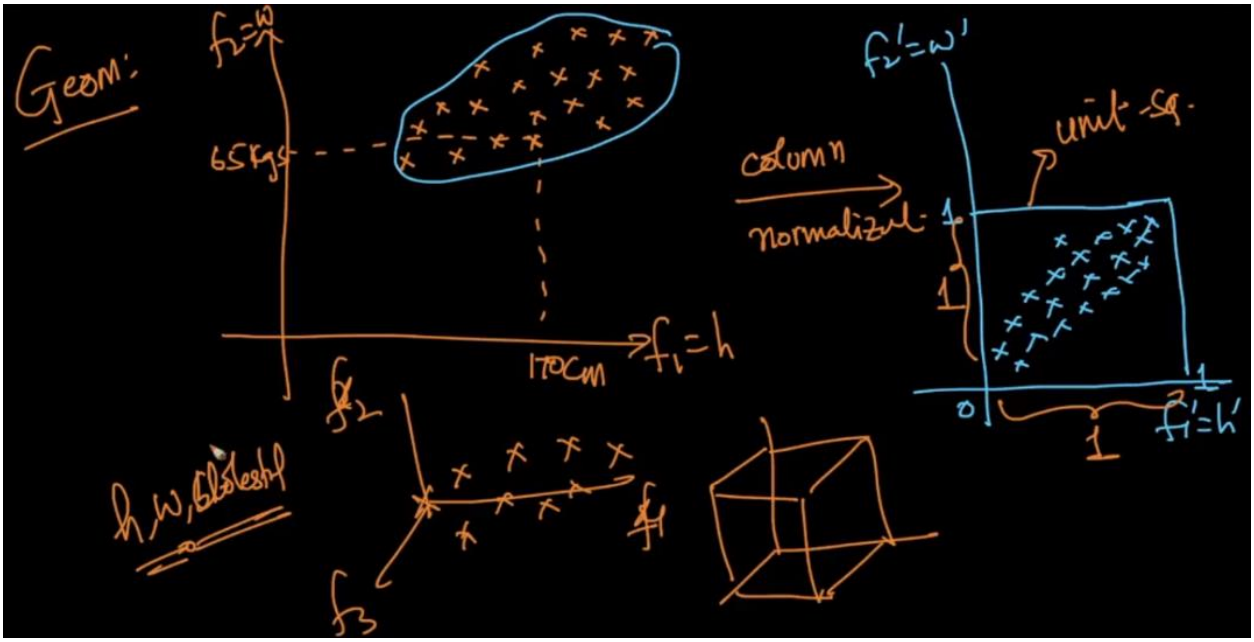


Why do we do Normalization?

It's Used to get rid of scalling, in real world data come from many scales example height and weight, where height can be measured in cm/m/f or weight can be measured in kg/lbs, but for now we have to take care of units and therefore we bring all the points within 0 to 1



While plotting also our graph will contains 1 unit in each dimensions, so we can say that normalization make to plot n-dimensional data into unit hypercube in same n-dimensions.



Note:

- Thus feature normalisation or features scaling results all value in [0, 1] with similar distribution for Data having metric values with different numerical values for mean , variances and standard deviation
- **Small Catch for Normalization:** The catch is that if you are normalizing a query point(new point), if one of the features values were greater/lesser than any training point feature then you'd end up with a value outside the range of 0,1. For example if in the train data you had the maximum value of one of the features to be 100 and minimum to be 2, and if the query point at that particular feature value had a value lesser than 2 or greater than 100 then the normalized value of the feature lies outside the range of [0,1].
- **Feature Scaling**. There are 2 types of Feature scaling techniques. They are**Normalization(which is also called 0-1 scaling)** and **Standardization(which is also called mean centred variance scaling)**

- Scaling of a feature is bringing all the features onto a scale with same mean and same variance. Here in K-NN we generally prefer Feature Standardization which transforms the values of a feature in such a way that the distribution of the feature has a mean of 0 and a variance of 1.

  Feature scaling is generally used when we are working on distance based algorithms like KNN, Logistic Regression, SVM, etc because having different scales for different features might lead to incorrect results and incorrect interpretation, Hence feature scaling is mandatory. In case if we are working on non-distance based algorithms like Decison Trees, Random Forests, it doesn't make much sense whether we apply feature scaling or not