SCALING

Most of the times, your dataset will contain features highly varying in magnitudes/units. But this is a problem since most of the machine learning algorithms use Euclidian distance between two data points in their computations. If left alone, the features with high magnitudes will weigh in a lot more in the distance calculations than features with low magnitudes and will start dominating when calculating distances. To suppress this effect, we need to bring all features to the same level of magnitudes/units

Scaling is the process of bringing the features to same scale. It helps to normalize the data within a particular range. Sometimes, it also helps in speeding up the calculations in an algorithm.

Standardization

Standardization is a transformation that centers the data by removing the mean value of each feature and then scale it by dividing (non-constant) features by their standard deviation.

Standard Scaler standardizes a feature by subtracting the mean and then scaling to unit variance. Unit variance means dividing all the values by the standard deviation.

x_scaled = (x — u) / s

where u is mean and s is standard deviation.

In Standard scaler we scale features to a scale between -3 and 3. This redistributes the features with their mean $\mu = 0$ and standard deviation $\sigma = 1$, variance = 1 (variance = standard deviation squared)

Min-Max Scaling

This scaler works better for cases where the distribution is not Gaussian or the standard deviation is very small. However, it is sensitive to outliers, so if there are outliers in the data, you might want to consider another scaler.

x_scaled = (x-min(x)) / (max(x)–min(x)

In Min max scaler we scale feature to a scale between -1 and 1

Mean Normalization

x_scaled = (x-mean(x)) / (max(x)–min(x)

In Mean Normalization, we scale feature to a scale between -1 and 1.

Which Method To Use

It is hard to know whether rescaling your data will improve the performance of your algorithms before you apply them. If often can, but not always.

The short answer is both and depending on the application. Each method has its practical use

A good tip is to create rescaled copies of your dataset and race them against each other using your test harness and a handful of algorithms you want to spot check.

Points to consider

- Algorithms that use distance calculations like K Nearest Neighbor, Regression, SVMs are the ones that requires feature scaling.

- Algorithms that does not use distance calculations like Naive Bayes, Tree based models, LDA do not require feature scaling.

- Feature technique to use varies from cases to case. For Example - For PCA, it is advisable to use standardization.