Feature Scaling

What is feature scaling?

Feature scaling is a data manipulation technique used in many machine-learning models and algorithms. The general idea is to transform raw values so that they are within the same scale. The general objective of feature scaling is to avoid having one feature dominate other features during training due to a having a differing range. For example, say we have two features: score and effort. Further, suppose the values for score exist on a scale from [0, 100], and those for effort exist on a scale from [0, 1000000]. During training, the relatively large values of effort will dominate those of score. To account for this, we scale each of the raw data sets so that the values for both score and effort exist within the same scale, say [0,1]. During training then the machine learning models that expect these ranges to be on the same scale will not inadvertently introduce any bias.

Why is feature scaling necessary?

In general, the ranges of raw data vary widely and machine learning algorithms see nothing but numbers. While humans understand that 10 lbs and $100 000 are vastly different things, machine learning algorithms simply see 10 and 100 000. So these numbers play an integral role while training machine models. One example of this can be seen in gradient descent algorithms that seek a minimum of a cost function. Their goal is to find parameters such that the cost function is minimized. The only thing the computer may do is randomly choose values for the parameters. The value chosen for the next parameter is a function of the current value, and some information about the gradient vector. But if the ranges are dissimilar, the gradient vector will likely fail to point directly towards the minima, thus presenting a jagged path, as opposed to a rather straight one. Thus, the time spent finding the minima will not be optimized. This explains just one reason for the scaling of features. More generally, without scaling features, we risk introducing bias to our model due to the algorithm being biased towards the features with relatively larger values. So we must scale the features of our model so that the machine learning process treats them all equally.

What does normalization and standardization do to the data and noise?

Standardization rescales features such that they have the characteristics of a standard normally distributed graph, with mean 0 and standard deviation of 1. It is widely to determine the distribution mean [2] (Eq 1) and standard deviation [3] (Eq 2) for each feature. The values of each of the features is divided by its standard deviation (Eq 3).

$$\text{Eq (1)}\ \ \mu = \frac{1}{N}\sum_{i=1}^{N} x \qquad \text{Eq (2)}\ \sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)} \qquad \text{Eq (3)}\ x' = \frac{x-\mu}{\sigma}$$

Normalization, on the other hand, transforms features to fit the scale of the range [0, 1] or [-1, 1] (Eq 4). The selection of target range depends on the aspects of the data.

$$\text{Eq (4)}\ x' = \frac{x-\min(x)}{\max(x)-\min(x)}$$

Standardization tends to be helpful when the data follows a Gaussian distribution; however, since standardization does not have a bounding range, outliers will continue to be outliers [7]. Note that standardization *can* become skewed or biased if the input contains heavy outliers, so one approach is to ignore the outliers from the calculation of the mean and standard deviation, then to scale the outliers themselves [8]. For standardization, generally, the mean and standard deviation calculations will be affected by a smaller margin. Standardization would be more intuitive to use when the data sample are in form of normal distribution and the data contains heavy outliers that would affect min/max calculations [6]. If the data does not follow a standard distribution, or if the algorithm being used does not assume any distribution at all, it may make more sense to use normalization [7]. In normalization the noise of the data is scaled into the appropriate range. Given min and max values, outliers will therefor weigh heavily in the calculations. We should use normalization when the data set is not normally distributed and is clean from particularly heavy outliers.

References

[1] Feature scaling : https://en.wikipedia.org/wiki/Feature_scaling

[2] Mean : https://en.wikipedia.org/wiki/Mean#Arithmetic_mean_(AM)

[3] Standard Deviation : https://en.wikipedia.org/wiki/Standard_deviation

[4] Gradient Descent : https://en.wikipedia.org/wiki/Gradient_descent

[5] Plot scaling importance https://scikit-learn.org/stable/auto_examples/preprocessing/plot_scaling_importance.html

[6] https://www.enjoyalgorithms.com/blog/need-of-feature-scaling-in-machine-learning

[7] https://www.atoti.io/articles/when-to-perform-a-feature-scaling/

[8] https://machinelearningmastery.com/robust-scaler-transforms-for-machine-learning/