Feaure Scaling Techniques

Tushar B. Kute,
http://tusharkute.com

Why to apply feature scaling ?

- Real world dataset contains features that highly vary in magnitudes, units, and range.
- Normalisation should be performed when the scale of a feature is irrelevant or misleading and not should Normalise when the the scale is meaningful.
- The algorithms which use Euclidean Distance measure are sensitive to Magnitudes. Here feature scaling helps to weigh all the features equally.
- Formally, If a feature in the dataset is big in scale compared to others then in algorithms where Euclidean distance is measured this big scaled feature becomes dominating and needs to be normalized.

Why to apply scaling ?

- For example, assume your input dataset contains one column with values ranging from 0 to 1, and another column with values ranging from 10,000 to 100,000.
- The great difference in the scale of the numbers could cause problems when you attempt to combine the values as features during modeling.
- Normalization avoids these problems by creating new values that maintain the general distribution and ratios in the source data, while keeping values within a scale applied across all numeric columns used in the model.

Gradient Descent Based Algorithms

- Machine learning algorithms like linear regression, logistic regression, neural network, etc. that use gradient descent as an optimization technique require data to be scaled.
- Take a look at the formula for gradient descent below:

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

Gradient Descent Based Algorithms

- The presence of feature value X in the formula will affect the step size of the gradient descent. The difference in ranges of features will cause different step sizes for each feature.
- To ensure that the gradient descent moves smoothly towards the minima and that the steps for gradient descent are updated at the same rate for all the features, we scale the data before feeding it to the model.
- Having features on a similar scale can help the gradient descent converge more quickly towards the minima.

Distance Based Algorithms

- Distance algorithms like KNN, K-means, and SVM are most affected by the range of features.
- This is because behind the scenes they are using distances between data points to determine their similarity.
- For example, let’s say we have data containing high school CGPA scores of students (ranging from 0 to 5) and their future incomes (in thousands Rupees):

Distance Based Algorithms

- Since both the features have different scales, there is a chance that higher weightage is given to features with higher magnitude.
- This will impact the performance of the machine learning algorithm and obviously, we do not want our algorithm to be biased towards one feature.

	Student	CGPA	Salary '000
0	1	3.0	60
1	2	3.0	40
2	3	4.0	40
3	4	4.5	50
4	5	4.2	52

Distance Based Algorithms

- Therefore, we scale our data before employing a distance based algorithm so that all the features contribute equally to the result.

	Student	CGPA	Salary '000
0	1	-1.184341	1.520013
1	2	-1.184341	-1.100699
2	3	0.416120	-1.100699
3	4	1.216350	0.209657
4	5	0.736212	0.471728

Distance Based Algorithms

- The effect of scaling is conspicuous when we compare the Euclidean distance between data points for students A and B, and between B and C, before and after scaling as shown below:

  - Distance AB before scaling =>

    $$\sqrt{(40-60)^2+(3-3)^2}=20$$

  - Distance BC before scaling =>

    $$\sqrt{(40-40)^2+(4-3)^2}=1$$

  - Distance AB after scaling =>

    $$\sqrt{(1.1+1.5)^2+(1.18-1.18)^2}=2.6$$

  - Distance BC after scaling =>

    $$\sqrt{(1.1-1.1)^2+(0.41+1.18)^2}=1.59$$

Tree Based Algorithms

- Tree-based algorithms, on the other hand, are fairly insensitive to the scale of the features.
- Think about it, a decision tree is only splitting a node based on a single feature.
- The decision tree splits a node on a feature that increases the homogeneity of the node. This split on a feature is not influenced by other features.
- So, there is virtually no effect of the remaining features on the split. This is what makes them invariant to the scale of the features!

Feature Scaling Techniques

- Min-Max Scaler
  - Normalization
- Standard Scaler
  - Standardization
- Robust Scaling
  - Robust Scaler

Normalization

- Normalization is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1. It is also known as Min-Max scaling.
- Here's the formula for normalization:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Normalization

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

- Here, Xmax and Xmin are the maximum and the minimum values of the feature respectively.
  - When the value of X is the minimum value in the column, the numerator will be 0, and hence X' is 0
  - On the other hand, when the value of X is the maximum value in the column, the numerator is equal to the denominator and thus the value of X' is 1
  - If the value of X is between the minimum and the maximum value, then the value of X' is between 0 and 1

Example:

- Data = 1000,2000,3000,9000

  Using min-max normalization by setting min:0 and max:1

- Solution:

  here,new_max(A)=1 , as given in question- max=1

  new_min(A)=0,as given in question- min=0

  max(A)=9000,as the maximum data among 1000,2000,3000,9000 is 9000

  min(A)=1000,as the minimum data among 1000,2000,3000,9000 is 1000

Example:

- Case-1: normalizing 1000 –

  v = 1000 , putting all values in the formula,we get

  v' = (1000-1000) X (1-0)

  ---------- + 0 = 0

  9000-1000

Example:

- Case-2: normalizing 2000 –

v = 2000, putting all values in the formula,we get

v '= (2000-1000) X (1-0)

---------- + 0 = 0 .125

9000-1000

Example:

- Case-3: normalizing 3000 –

  v=3000, putting all values in the formula,we get

  v'=(3000-1000) X (1-0)

  ---------- + 0 = 0 .25

  9000-1000

Example:

- Case-4: normalizing 9000 –

  v=9000, putting all values in the formula, we get

  v'=(9000-1000) X (1-0)

  ---------- + 0 = 1

  9000-1000

- Outcome :

  Hence, the normalized values of 1000,2000,3000,9000 are 0, 0.125, .25, 1.

When to apply?

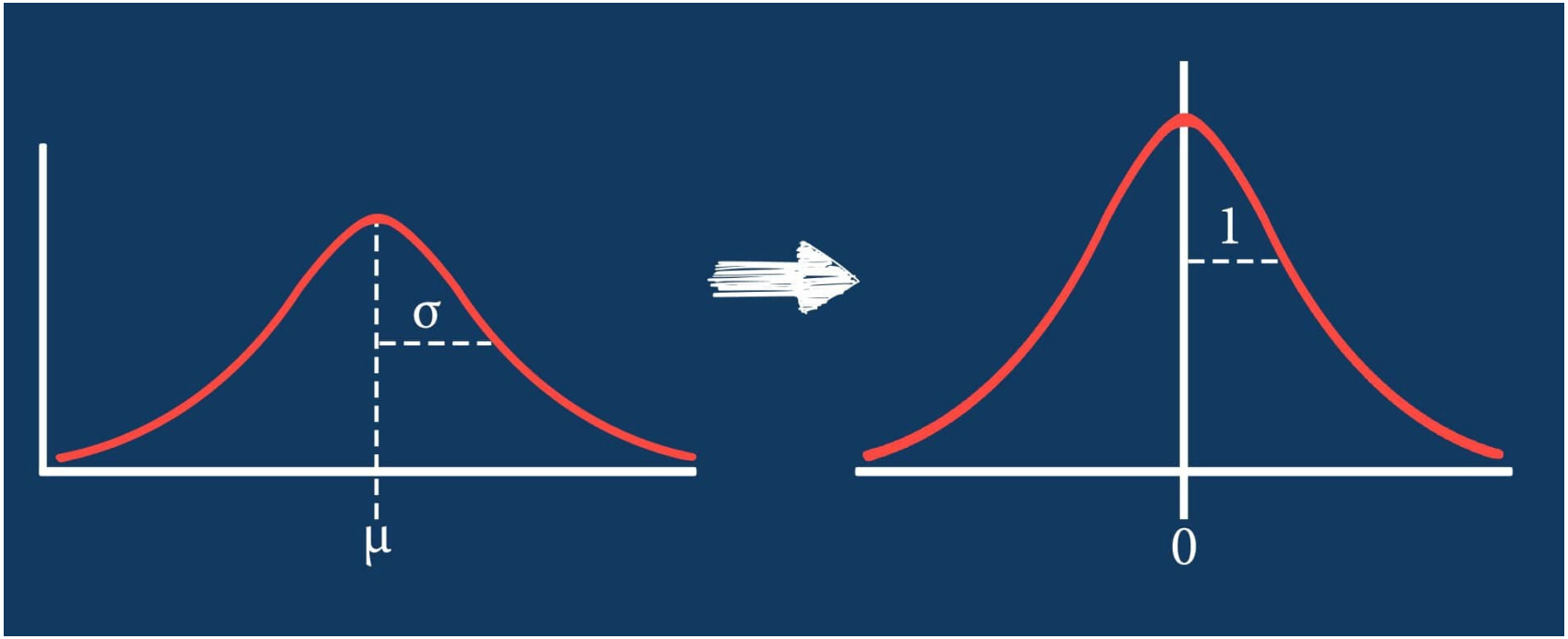
- Normalization is a good technique to use when you do not know the distribution of your data or when you know the distribution is not Gaussian (a bell curve).
- Normalization is useful when your data has varying scales and the algorithm you are using does not make assumptions about the distribution of your data, such as k-nearest neighbors and artificial neural networks.

Standardization

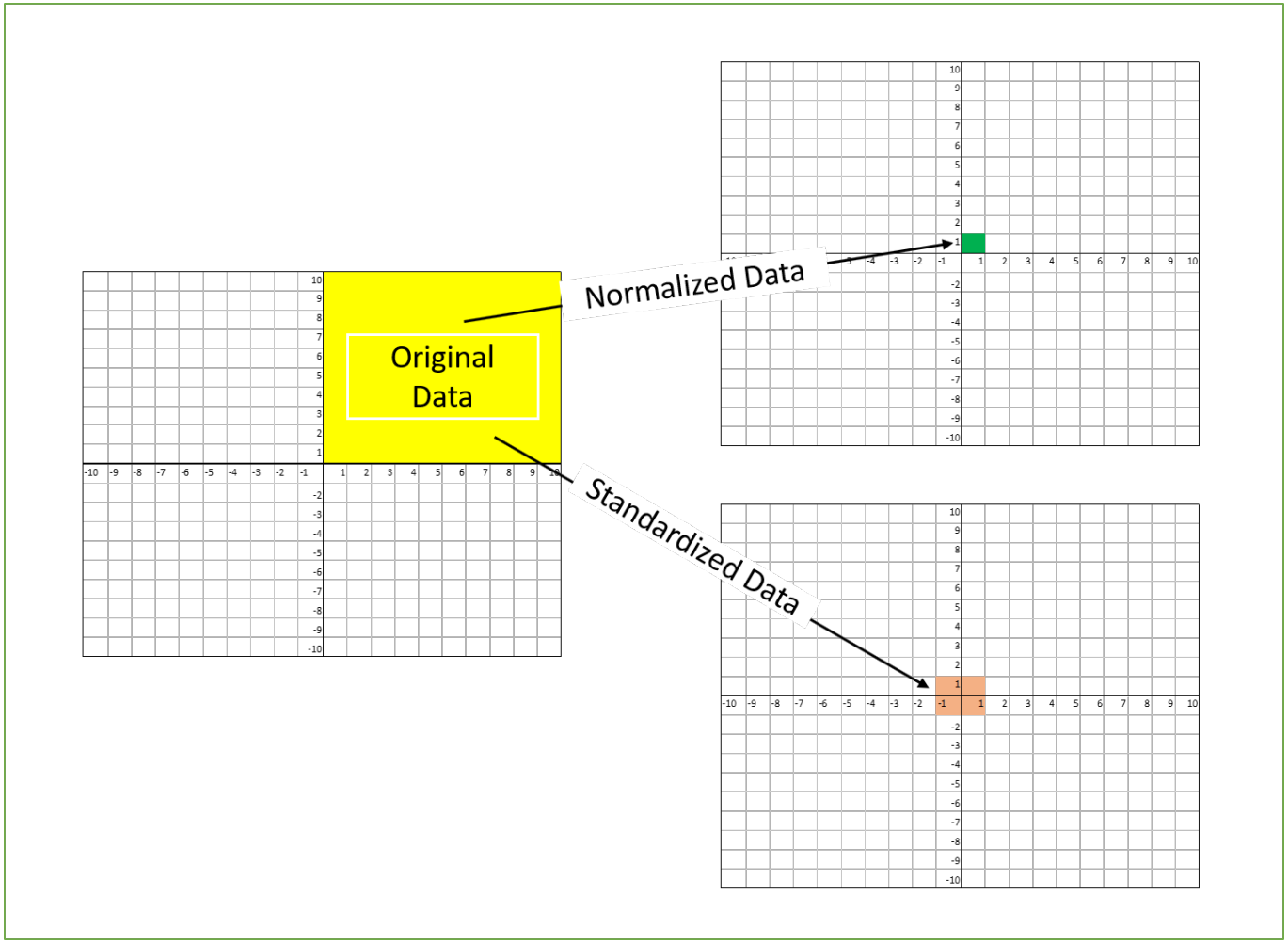
- Standardization is another scaling technique where the values are centered around the mean with a unit standard deviation.
- This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation.
- Here's the formula for standardization:

$$z = \frac{x_i - \mu}{\sigma}$$

Standardization



Comparison



Z-score

- Simply put, a z-score (also called a standard score) gives you an idea of how far from the mean a data point is.
- But more technically it's a measure of how many standard deviations below or above the population mean a raw score is.
- A z-score can be placed on a normal distribution curve. Z-scores range from -3 standard deviations (which would fall to the far left of the normal distribution curve) up to +3 standard deviations (which would fall to the far right of the normal distribution curve).
- In order to use a z-score, you need to know the mean μ and also the population standard deviation σ.

Z-score

- Z-scores are a way to compare results to a “normal” population. Results from tests or surveys have thousands of possible results and units; those results can often seem meaningless.
- For example, knowing that someone’s weight is 150 pounds might be good information, but if you want to compare it to the “average” person’s weight, looking at a vast table of data can be overwhelming (especially if some weights are recorded in kilograms).
- A z-score can tell you where that person’s weight is compared to the average population’s mean weight.

Z-score

- The basic z score formula for a sample is:

  $z = (x – \mu) / \sigma$

- For example, let's say you have a test score of 190. The test has a mean (μ) of 150 and a standard deviation (σ) of 25. Assuming a normal distribution, your z score would be:

  $z = (x – \mu) / \sigma$

  $= (190 – 150) / 25 = 1.6.$

- The z score tells you how many standard deviations from the mean your score is. In this example, your score is 1.6 standard deviations above the mean.

Z-score

- You may also see the z score formula shown to the left.
- This is exactly the same formula as z = x – μ / σ, except that $\bar{x}$ (the sample mean) is used instead of μ (the population mean) and s (the sample standard deviation) is used instead of σ (the population standard deviation). However, the steps for solving it are exactly the same.

$$z_i = \frac{x_i - \bar{x}}{s}$$

Standardization

- Standardization assumes that your data has a Gaussian (bell curve) distribution.
- This does not strictly have to be true, but the technique is more effective if your attribute distribution is Gaussian.
- Standardization is useful when your data has varying scales and the algorithm you are using does make assumptions about your data having a Gaussian distribution, such as linear regression, logistic regression, and linear discriminant analysis.

Maximum Absolute Scaling

- Maximum absolute scaling scales the data to its maximum value; that is, it divides every observation by the maximum value of the variable:

$$xscaled = \frac{x}{max(x)}$$

- The result of the preceding transformation is a distribution in which the values vary approximately within the range of -1 to 1.
- Scikit-learn recommends using this transformer on data that is centered at zero or on sparse data.
- This scaler is sensitive to outliers if all the values are positive.

Robust Scaler

- Robust Scaler algorithms scale features that are robust to outliers.
- The method it follows is almost similar to the MinMax Scaler but it uses the interquartile range (rather than the min-max used in MinMax Scaler).
- The median and scales of the data are removed by this scaling algorithm according to the quantile range.
- It, thus, follows the following formula:

$$\frac{x_i - Q_1(x)}{Q_3(x) - Q_1(x)}$$

- Where Q1 is the 1st quartile, and Q3 is the third quartile.

Thank you

@mitu_skillologies

/mITuSkillologies

@mitu_group

/company/mitu-skillologies

MITUSkillologies

Web Resources
https://mitu.co.in
http://tusharkute.com

contact@mitu.co.in

tushar@tusharkute.com