Feature Scaling: Standardization vs Normalization

In machine learning, feature scaling is crucial for improving model performance, especially when features have different units and ranges. Algorithms that compute the distance between the features are biased towards numerically larger values if the data is not scaled. Scaling ensures that all features contribute equally to the model.

Tree-based algorithms are fairly insensitive to the scale of the features. (e.g., decision trees, random forests, gradient boosting) are generally insensitive to feature scaling, meaning they don't require the data to be scaled for optimal performance.

Despite this, feature scaling can significantly improve the training efficiency of machine learning and deep learning models by helping them converge faster.

Two common techniques for feature scaling are **Standardization and Normalization.**

Why is Feature Scaling Important ?

When features have vastly different scales (e.g., one ranging from 1 to 10 and another from 1,000 to 10,000), machine learning models may prioritize the larger values. This can lead to bias in predictions and slower convergence during model training.

Feature scaling helps solve these issues by adjusting the range of the data without distorting the differences in feature values.
By doing so, it ensures that all features have a similar impact on the model, leading to better model performance and more efficient training.

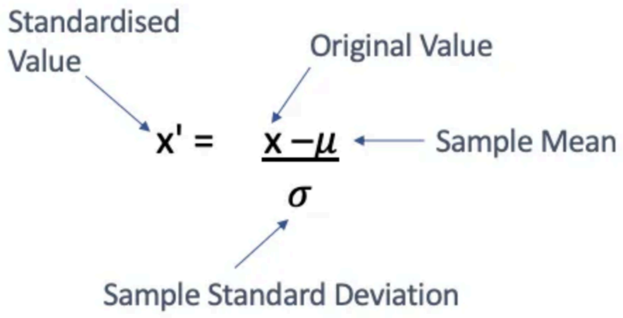
Benefits of Feature Scaling

- **Prevents bias** in predictions by ensuring features are on the same scale.
- **Reduces the impact of outliers**, which can skew model learning.
- **Improves convergence** during model training, especially for algorithms that rely on distance calculations (e.g., k-NN, SVM, gradient descent).

Standardization (Z-Score Normalization)

Standardization, also known as Z-Score Normalization, is the process of transforming features by subtracting the mean and dividing by the standard deviation. This technique is often referred to as Z-score normalization.

The formula is:

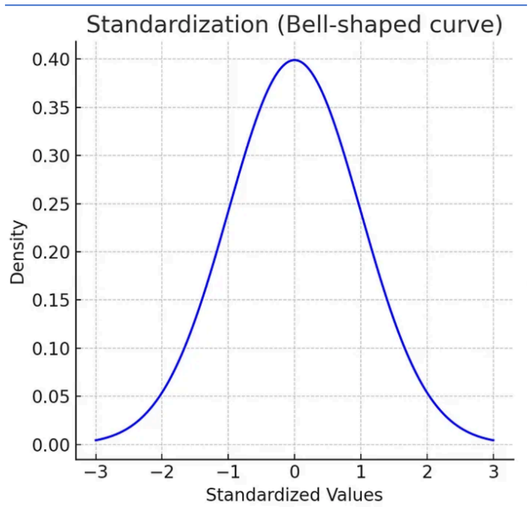


Formula for Standardisation

This transformation is particularly useful when the data is normally distributed (Gaussian distribution), but it's not a requirement.

Geometrically, standardization shifts the data by subtracting the mean, moving the mean to the origin (0), and scales the data based on its standard deviation.

The key point is that **standardization** doesn't affect the shape of the distribution — it just changes the mean to 0 and the standard deviation to 1, preserving the normal distribution.

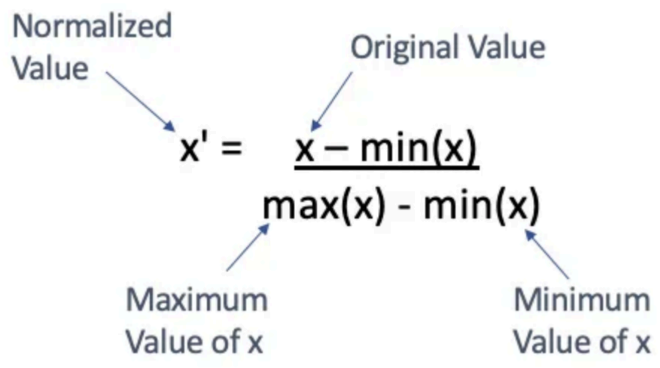


Why is Standardization Robust to Outliers?

Since standardization doesn't rely on a specific range of values, it is less sensitive to outliers compared to other scaling methods like normalization. The transformation doesn't compress the data to a fixed range, which can sometimes distort the impact of outliers.

Normalization (Min-Max Scaling)

Normalization, also known as Min-Max Scaling, is used to scale features so that they lie within a specific range, typically between 0 and 1 (or sometimes -1 and 1). This is achieved using the following formula :



Formula for Min-Max Normalization

Geometrically, normalization squishes the data into a unit **hypercube (in a multi-dimensional space)**, transforming the feature values into a fixed range. This makes it easier for machine learning models to process features with consistent scales.

Normalization is particularly useful when:

- Features do not have significant **outliers**. It can't handle extreme outliers well, as they will compress the range of all other data points.
- The data needs to be on a consistent scale for distance-based algorithms (e.g., k-NN, SVM), as these models are sensitive to the magnitude of values.
- You want all features to contribute equally to the model's learning process.

For example, age is generally a feature that is well-suited for normalization since it tends to be uniformly distributed. However, income often has a few very high values (outliers), which could distort the scaling. In this case, normalization may not be the best method for income, but could still be appropriate for age.

Normalization vs. Standardization: Key Differences

- **Rescaling Methods**
  - **Normalization (Min-Max Scaling):** Rescales the feature values to a specific range, usually between 0 and 1 or sometimes between -1 and 1. This is useful when the feature values have widely varying scales.

  - **Standardization (Z-score Normalization):** Centers the data around a **mean of 0 and scales it according to a standard deviation of 1**.

- **Sensitivity to Outliers**
  - **Normalization:** Outliers can distort the scaling process since they impact the minimum and maximum values. In some cases, normalization may not be effective in handling outliers because it is dependent on the range of the data.

  - **Standardization:** Standardization is more robust to outliers since it uses the mean and standard deviation, which are less sensitive to extreme values compared to the min and max values. However, extremely large outliers can still affect the scaling to some extent.

- **Use Cases**
  - **Normalization**
    - When you need the data within a specific range (e.g., neural networks, gradient-based models)
    - When you know that your features are bounded (e.g., pixel values, or ratings between 1-5).
  - **Standardization**
    - When the data is not bounded and might contain outliers (e.g., income, house prices).
    - When using algorithms like **SVM**, **Logistic Regression**, or **KNN**, where distance between data points matters.

Standardization is generally preferred when the model uses distances (e.g., KNN, SVM).

Normalization is useful for neural networks where the algorithm is sensitive to the scale of input data.