Why do we want to perform Feature Scaling?

Feature Scaling is a technique to standardize the independent features present in the data in a fixed range.

It is performed during data pre-processing to handle highly varying magnitudes, values, or units.

If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values.

In machine learning, feature scaling is employed for several purposes:

- Scaling guarantees that all features are on a comparable scale and have similar ranges. This process is known as feature normalization. This is significant because the magnitude of the features has an impact on many machine-learning techniques. Larger scale features may dominate the learning process and excessively impact the outcomes. We can avoid this problem and make sure that each feature contributes equally to the learning process by scaling the features.
- Algorithm performance improvement: When the features are scaled, several machine learning methods perform better or converge more quickly. The algorithm's performance can be enhanced by scaling the features, which can hasten the convergence of the algorithm to the ideal outcome.
- Preventing numerical instability: Numerical instability can be prevented by avoiding significant scale disparities between features. Examples include distance calculations or matrix operations, where having features with radically differing scales can result in numerical overflow or underflow problems. Stable computations are ensured and these issues are mitigated by scaling the features.
- Scaling features ensure that each characteristic is given the same consideration during the learning process. Without scaling, bigger scale features could dominate the learning, producing skewed outcomes. This bias is removed through scaling, which also guarantees that each feature contributes fairly to model predictions.

What is the difference between Standardization and Normalization?

Normalization or Min-Max Scaling is used to transform features to be on a similar scale.

The new point is calculated as:

X_new = (X - X_min)/(X_max - X_min)

This scales the range to [0, 1] or sometimes [-1, 1].

Normalization is useful when there are no outliers as it cannot cope up with them. Usually, we would scale age and not incomes because only a few people have high incomes but the age is close to uniform.

Standardization or Z-Score Normalization is the transformation of features by subtracting from mean and dividing by standard deviation. This is often called as Z-score.

X_new = (X - mean)/Std

Standardization can be helpful in cases where the data follows a Gaussian distribution. However, this does not have to be necessarily true. We can see that we are just changing mean and standard deviation to a standard normal distribution which is still normal thus the shape of the distribution is not affected.

Standardization does not get affected by outliers because there is no predefined range of transformed features.

Difference between Normalization and Standardization

S.NO.	Normalization	Standardization
1.	The minimum and maximum values of features are used for scaling	Mean and standard deviation is used for scaling.
2.	It is used when features are of different scales.	It is used when we want to ensure zero mean and unit standard deviation.
3.	Scales values between [0, 1] or [-1, 1].	It is not bound to a certain range.
4.	It is affected by outliers.	It is much less affected by outliers.
5.	Scikit-Learn provides a transformer called MinMaxScaler for Normalization.	Scikit-Learn provides a transformer called StandardScaler for Standardization.

S.NO.	Normalization	Standardization
6.	This transformation squishes the n-dimensional data into an n-dimensional unit hypercube.	It translates the data to the mean vector of the original data to the origin and squishes or expands.
7.	It is useful when we don't know about the distribution	It is useful when the feature distribution is Normal or Gaussian.
8.	It is often called as Scaling Normalization	It is often called as Z-Score Normalization.

Standardization on a given dataset

Standardization on the given Dataset

+ Code + Markdown

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)

print(x_train)
```

[68] ✓ 0.0s

```
[[0.         0.         1.         0.51304348 0.11428571]
 [1.         0.         0.         0.73913043 0.68571429]
 [0.         0.         1.         0.         0.        ]
 [0.         1.         0.         0.56521739 0.45079429]
 [1.         0.         0.         0.34782609 0.28571429]
 [0.         0.         1.         0.47826087 0.37142857]
 [0.         1.         0.         1.         1.        ]
 [1.         0.         0.         0.43478261 0.54285714]]
```

```
print(x_test)
```

[69] ✓ 0.0s

```
[[0.         1.         0.         0.13043478 0.17142857]
 [1.         0.         0.         0.91304348 0.88571429]]
```

Normalization on the given dataset