Feature Scaling

Feature Scaling is a very important step to **pre-process** the data set before apply the machine learning algorithm

Context

The motivation before **Feature Scaling** is when we have number multiple Features or Variables p

Feature A	Feature B
5	20000
10	30000
15	25000
20	40000
25	50000

- As you can notice both Features are a huge difference in the range of numbers
- In many machine learning algorithms they will interpret that the **Feature B** in this example have much more impact and will result in a very off results and predictions
- That's why we **Feature scale** our variable as a way to normalize the values which will results in a much better **approximations** and **convergence** in the case of Gradient Descent

Feature Scaling Methods

There is mainly two major methods that are used in feature scaling

1. **Normalization**
2. **Standardization**

Normalization

Also Known as *min-max scaling* Same as in **vector normalization** it shifts and scale the values within a range of $[0, 1]$ or $[-1, 1]$ Following this Formula :

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

- Both X_{max} and X_{min} are the upper and lower bound of the normalization interval , Calculated by taking the maximum and minimum of set **feature column**
- X' is the new calculated value of $X \to$ The normalized value of X

Standardization

Also Known as *z-score normalization* which follows the same formula from the **Standard normal distribution** it transforms linearly the data to have a mean μ of 0 and **Standard Deviation** of 1 Given by the *z-score* formula :

$$X' = \frac{X - \mu}{\sigma}$$

They Key Differences

Normalization	Standardization
Re-scale values to range between 0 and 1	Centers data around the **mean** and scales to a standard deviation of 1
Useful when the distribution of the data is **unknown** or **not Gaussian**	Useful when the distribution of the data is **Gaussian** (Normal) or **unknown**
Sensitive to **outliers**	Less sensitive to **outliers**
Retains the shape of the **original distribution**	Changes the shape of the original distribution
May not preserve the relationships between the data points	Preserves the relationships between the data points

Use cases

- **Standardization** and **Normalization** are generally preferred for any machine learning algorithm that require **Gradient** Calculation
- Scaling is not required for **distance-based** and **tree-based** algorithms
- **Standardization** is preferred when the data follows a **Normal Distribution** and when there is a lot of **outliers** in data set
- **Normalization** is preferred when there is no prior assumption on how the data is distributed