Normalization (Min-Max Scaling)

- **What it does**: Rescales data to a fixed range, usually **[0,1]** (or sometimes [-1,1]).

x'=x−xminxmax−xminx' = \frac{x - x_{min}}{x_{max} - x_{min}}x'=xmax−xminx−xmin

- **When to use**:
    1. **When features have different scales but you want them bounded.**
        - Example: Age (0–100) vs. Salary (0–100,000). Without normalization, salary dominates distance-based models.
    2. **For algorithms that rely on distance or magnitude**, like:
        - **KNN (K-Nearest Neighbors)**
        - **K-Means Clustering**
        - **Neural Networks (faster convergence with normalized inputs)**
    3. **When features are not normally distributed** but bounded scaling helps.

Standardization (Z-score Scaling)

- **What it does**: Centers data around **0 mean** and rescales to **unit variance**.

x'=x−μσx' = \frac{x - \mu}{\sigma}x'=σx−μ

- **When to use**:
    1. **When data follows (or approximately follows) a normal distribution.**
    2. **For algorithms that assume normality of data** or rely on variance:
        - **Linear Regression**
        - **Logistic Regression**
        - **SVM (Support Vector Machines)**
        - **PCA (Principal Component Analysis)**
    3. **When features can take on any range (no fixed boundary).**

Rule of Thumb

- Use **Normalization** when the distribution is *unknown* and you need everything in a fixed range (especially for distance-based ML).
- Use **Standardization** when the algorithm assumes Gaussian distribution or cares about variance.

👉 Think of it like this:

- **Normalization is like resizing different pictures to fit inside the same 1×1 box.**
- **Standardization is like shifting and stretching them so their “center” is the same, and their spread is comparable.**
