- Normalizes features based on their median and interquartile range (IQR), making it less sensitive to outliers.
- **Formula**:

$$x_{\text{norm}} = \frac{x - \text{median}}{\text{IQR}}$$

- **Use Case**: Ideal for datasets with many outliers or skewed distributions.

Example of Feature Normalization

Suppose we have a dataset with two features, **Age** and **Income**, where income has values in the thousands while age has values between 18 and 70. Applying Min-Max normalization (0-1 range) would make them comparable:

Original data:

- Age: [20, 50, 70]
- Income: [30000, 80000, 150000]

After Min-Max Normalization:

- Age: [0.05, 0.64, 1.0]
- Income: [0.0, 0.4, 1.0]

Mean of a Data Matrix

In a data matrix, the **mean** is calculated either for each **feature (column)** or each **observation (row)**, depending on the context. The mean provides a measure of the central tendency and is useful in many machine learning tasks, including normalization and standardization.

Let’s break down how to compute the mean in different contexts:

---

1. Column-wise Mean (Mean of Each Feature)

The **column-wise mean** calculates the mean of each feature (or attribute) across all observations in the dataset. This is commonly used to understand the average value of each feature.

For a data matrix X with m observations (rows) and n features (columns), the column-wise mean μj for each feature j is:

$$\mu_j = \frac{1}{m} \sum_{i=1}^{m} x_{ij}$$

Where:

- xij is the value of the j-th feature for the i-th observation.
- μj is the mean of feature j.

This results in a mean vector with n elements, representing the mean of each feature.

---

2. Row-wise Mean (Mean of Each Observation)

The **row-wise mean** calculates the mean for each observation across all features. This can be helpful when analyzing the average characteristic of each sample.

For the data matrix X, the row-wise mean μi for each observation i is:

$$\mu_i = \frac{1}{n} \sum_{j=1}^{n} x_{ij}$$

Where:

- xij is the value of the j-th feature for the i-th observation.
- μi is the mean of observation i.

This results in a mean vector with mmm elements, where each value represents the average across all features for a specific observation.

Example of Mean Calculation

Consider a small data matrix with 3 observations and 2 features:

$$\mathbf{X} = \begin{bmatrix} 4 & 10 \\ 8 & 15 \\ 6 & 12 \end{bmatrix}$$

- Column-wise Mean:
  - For the first column: $\mu_1 = \frac{4+8+6}{3} = 6$
  - For the second column: $\mu_2 = \frac{10+15+12}{3} = 12.33$

  So, the column-wise mean vector is $\mu = [6, 12.33]$.
- Row-wise Mean:
  - For the first row: $\mu_1 = \frac{4+10}{2} = 7$
  - For the second row: $\mu_2 = \frac{8+15}{2} = 11.5$
  - For the third row: $\mu_3 = \frac{6+12}{2} = 9$

  The row-wise mean vector is $\mu = [7, 11.$ [illegible] $]$.

Column Standardization

Column standardization is a preprocessing technique used in machine learning to scale each feature (column) so that it has a **mean of 0** and a **standard deviation of 1**. This transformation, also called **Z-score standardization** or **Z-score normalization**, makes data values comparable across different features. Standardization is especially helpful in algorithms sensitive to the scale of features, such as linear regression, logistic regression, and neural networks.

How Column Standardization Works

Given a data matrix Xwith m observations (rows) and n features (columns), we standardize each feature j as follows:

For each feature (column) j:

1. **Calculate the Mean (μj)**: Find the average value of all entries in column j.
2. **Calculate the Standard Deviation (σj)**: Measure the spread of values around the mean for column j.
3. **Apply Standardization Formula**: