Appropriate Scaling Techniques and Their Use Cases in Machine Learning

Applying **scaling techniques** is essential for improving the performance of machine learning models, especially for algorithms that are sensitive to the range and distribution of numerical features. Below are different scaling techniques and when to use them:

---

1. Min-Max Scaling (Normalization)

- **Formula**:

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

- **Range**: Scales values between **0 and 1** (or any defined range).
- **When to Use:**
    - When data has a fixed **minimum and maximum**.
    - Used in **Deep Learning (Neural Networks)** and **K-Nearest Neighbors (KNN)**.
    - Works well when feature values have different units (e.g., age in years, income in dollars).

---

2. Standardization (Z-score Scaling)

- **Formula**:

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

- **Range**: Scales values to have a **mean of 0** and a **standard deviation of 1**.
- **When to Use:**
    - When features have **different scales and distributions**.

- Used in algorithms like **Logistic Regression, SVM, and PCA**.
- Works well for **normally distributed** data.

---

3. Robust Scaling

- **Formula**:

$$X' = \frac{X - \text{Median}}{\text{IQR}}$$

- **Range**: Based on the **median and interquartile range (IQR)**.

- **When to Use:**
  - When data contains **outliers**.
  - Suitable for **decision trees, ensemble models, and SVM**.

---

4. Log Transformation

- **Formula**:

$$X' = \log(X + 1)$$

- **When to Use:**
  - When data is **highly skewed or has exponential distribution**.
  - Used in **linear regression, neural networks, and deep learning**.

---

5. Power Transformation (Box-Cox / Yeo-Johnson)

- **Purpose**: Reduces **skewness** and makes data more **normally distributed**.
- **When to Use:**
  - When data does **not follow a normal distribution**.
  - Used in **linear regression and PCA**.

Where Should You Apply Scaling?

- **Before training machine learning models** (e.g., Logistic Regression, SVM, Neural Networks, PCA).
- **Before distance-based algorithms** (e.g., KNN, K-Means, DBSCAN) because they are sensitive to feature scales.
- **Before applying Principal Component Analysis (PCA)** to ensure features contribute equally to variance.

What Data Should We Use for Scaling?

When applying scaling techniques, it's important to choose the correct data to scale to improve model performance. Here's what you need to consider:

---

1. Scale Only Numerical Features

- **Why?** Scaling applies only to continuous numerical data (e.g., age, salary, tenure).
- **What to Scale?**
  ✅ Age, income, monthly charges, tenure, height, weight, etc.
  ❌ Categorical variables (gender, contract type) should not be scaled.

Example from Your Dataset:

- ✅ **To Scale:** `tenure`, `MonthlyCharges`
- ❌ **Do Not Scale:** `gender_1`, `Contract_1`, `Contract_2` (One-Hot Encoded)

---

2. Do Not Scale Target Variables (Labels)

- **Why?** The target variable (e.g., `Churn_1`) is categorical (0 or 1), so scaling it **is unnecessary**.

---

3. Apply Scaling Only to the Training Set (Then Transform the Test Set)

- **Why?** To **avoid data leakage** (i.e., using information from the test set during training).
- **How?**

1. Fit the scaler **only on the training set**.
2. Transform both the training and test sets using the same fitted scaler.

---

4. When Not to Use Scaling

- **Tree-based models (Decision Trees, Random Forest, XGBoost) do not require scaling.**
- **Some models (like Naïve Bayes) work with probabilities and do not require scaling.**
