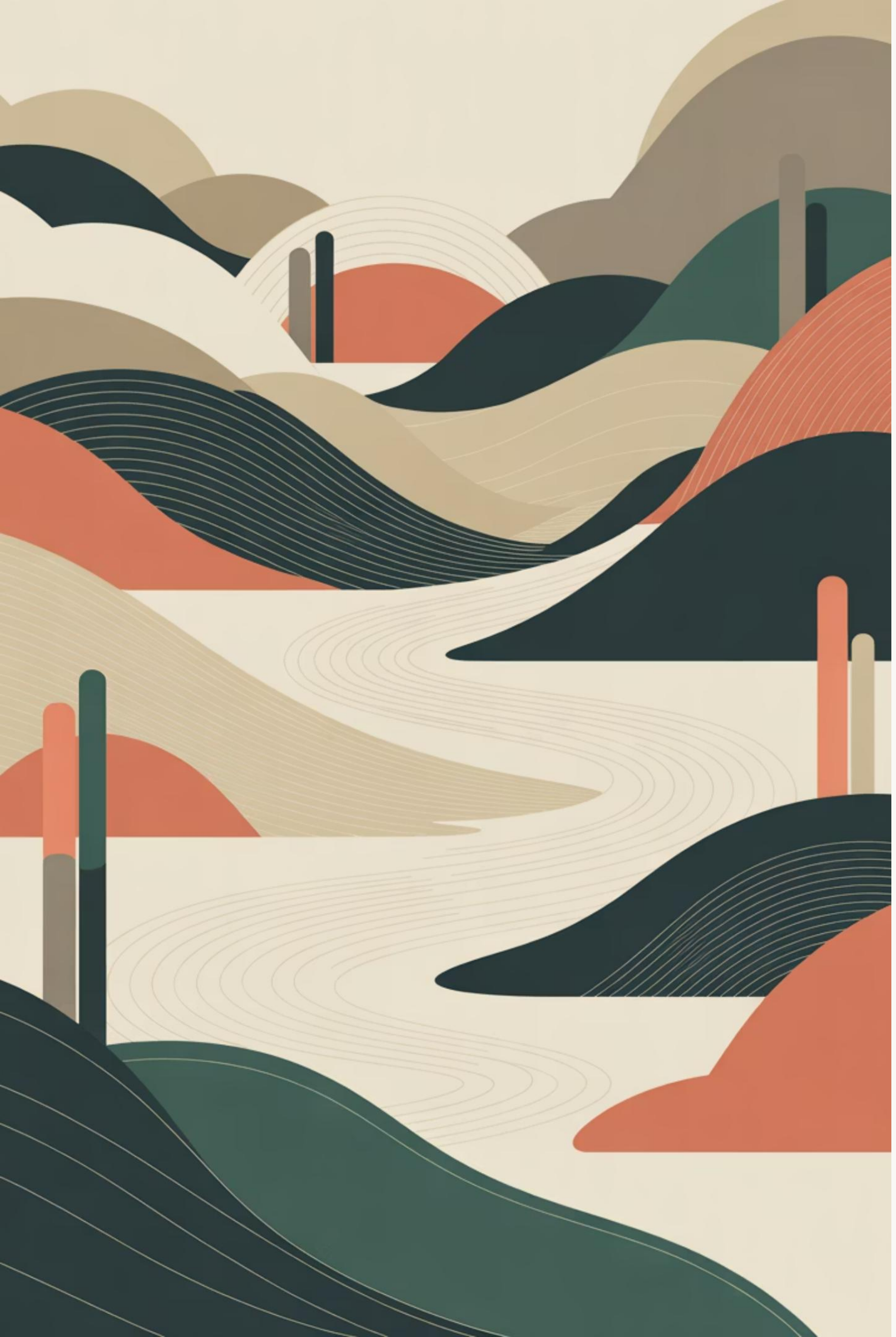


CEFET/RJ

Data Normalization in Machine Learning

Understanding how to prepare features for optimal model performance through normalization techniques

Eduardo Ogasawara
eduardo.ogasawara@cefet-rj.br
https://eic.cefet-rj.br/~eogasawara

Why Normalization Matters

The Challenge

Features in datasets often exist on vastly different scales. Age might range from 0-100, while income could span 0-1,000,000. Without normalization, scale-sensitive algorithms give disproportionate weight to larger-scale features.

This imbalance can severely impact model performance, especially in distance-based and gradient-descent algorithms.

The Solution

Normalization transforms features to comparable scales, ensuring each contributes appropriately to the model. Two fundamental approaches dominate:

- **Min-Max Normalization:** Rescales to a fixed range
- **Z-Score Standardization:** Centers around mean with unit variance

The DAL Toolbox provides both methods with simple, consistent syntax.

Min-Max Normalization: Theory

Core Concept

Rescales all values to fit within the range [0,1], preserving the original distribution's shape while standardizing the scale.

The Formula

$$x_{normalized} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Each value is adjusted relative to the feature's minimum and maximum.

Key Benefits

- Handles different units seamlessly
- Ideal for neural networks
- Bounded output range
- Preserves zero values

Important consideration: Min-Max normalization is sensitive to outliers, as extreme values define the range and can compress the majority of data points.

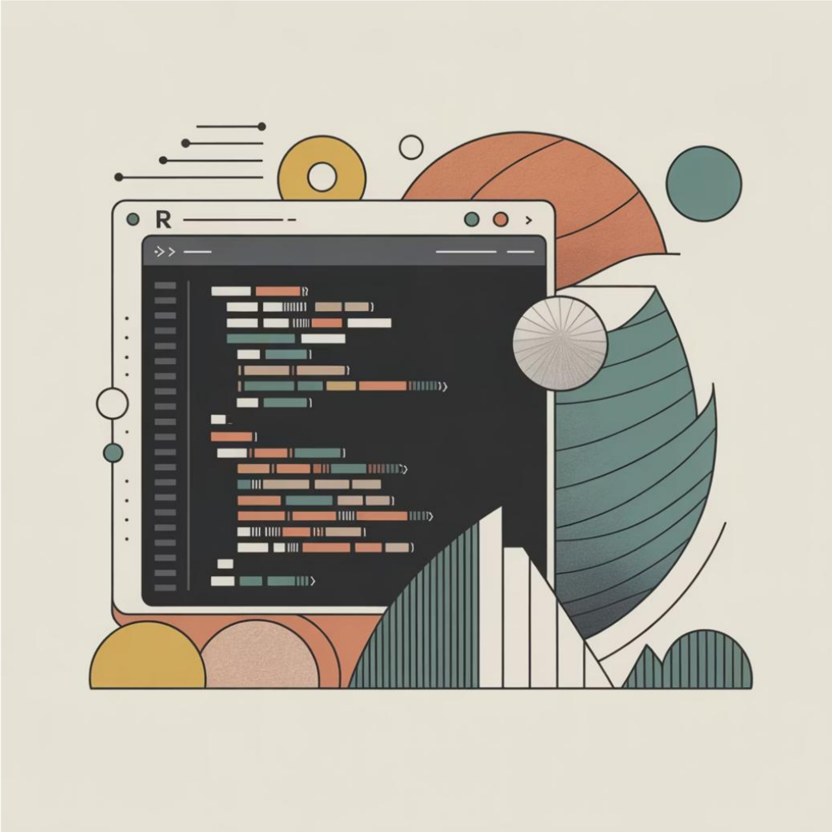
Reference: Han, J., Kamber, M., & Pei, J. – Data Mining: Concepts and Techniques (3rd Ed.); Witten, I. H., Frank, E., Hall, M. A., & Pal, C. – Data Mining: Practical Machine Learning Tools and Techniques (4th Ed.)

Min-Max in Practice: DAL Toolbox

Implementation Steps

The DAL Toolbox makes Min-Max normalization straightforward with a three-step workflow:

```r
# Step 1: Create normalizer object
norm <- minmax()
# Step 2: Fit to your data
norm <- fit(norm, datasets::iris)
# Step 3: Transform the dataset
idata <- transform(norm, datasets::iris)
# Verify results
summary(idata)
```



This pattern follows the familiar fit-transform paradigm, making it intuitive for practitioners experienced with scikit-learn or similar frameworks.

Full example available: https://github.com/cefet-rj-dal/daltoolbox/blob/main/transf/normalization_minmax.md

When to Use Min-Max Normalization

Neural Networks

Essential for deep learning models where bounded inputs [0,1] prevent saturation in activation functions and accelerate convergence during backpropagation.

Distance-Based Algorithms

Critical for k-Nearest Neighbors (kNN) and clustering algorithms where Euclidean distance calculations require features on comparable scales.

Gradient Descent Models

Improves optimization in linear regression, logistic regression, and support vector machines by creating more uniform gradient landscapes.

> **Caution:** Min-Max normalization is highly sensitive to outliers. A single extreme value can compress the entire distribution, reducing the method's effectiveness. Consider outlier detection and removal before applying Min-Max scaling.

Z-Score Standardization: Theory

1 Statistical Foundation

Z-Score standardization transforms data to have a mean of 0 and standard deviation of 1, creating a standard normal distribution.

$$z = \frac{x - \mu}{\sigma}$$

Where μ is the mean and σ is the standard deviation of the feature.

2 Advantages Over Min-Max

- Less sensitive to outliers
- No bounded range limitation
- Preserves information about outliers
- Handles different units effectively

3 Optimal Applications

Essential before Principal Component Analysis (PCA) and other techniques that assume normally distributed data. Particularly useful when features have varying scales but you want to maintain outlier information.

Reference: Han, J., Kamber, M., & Pei, J. – Data Mining: Concepts and Techniques (3rd Ed.); Witten, I. H., Frank, E., Hall, M. A., & Pal, C. – Data Mining: Practical Machine Learning Tools and Techniques (4th Ed.)

Z-Score in Practice & Custom Scaling

1

Standard Z-Score

```
# Create standardizer
norm <- zscore()
# Fit to data
norm <- fit(norm, datasets::iris)
# Transformzdata <- transform(norm, datasets::iris)
summary(zdata)
```

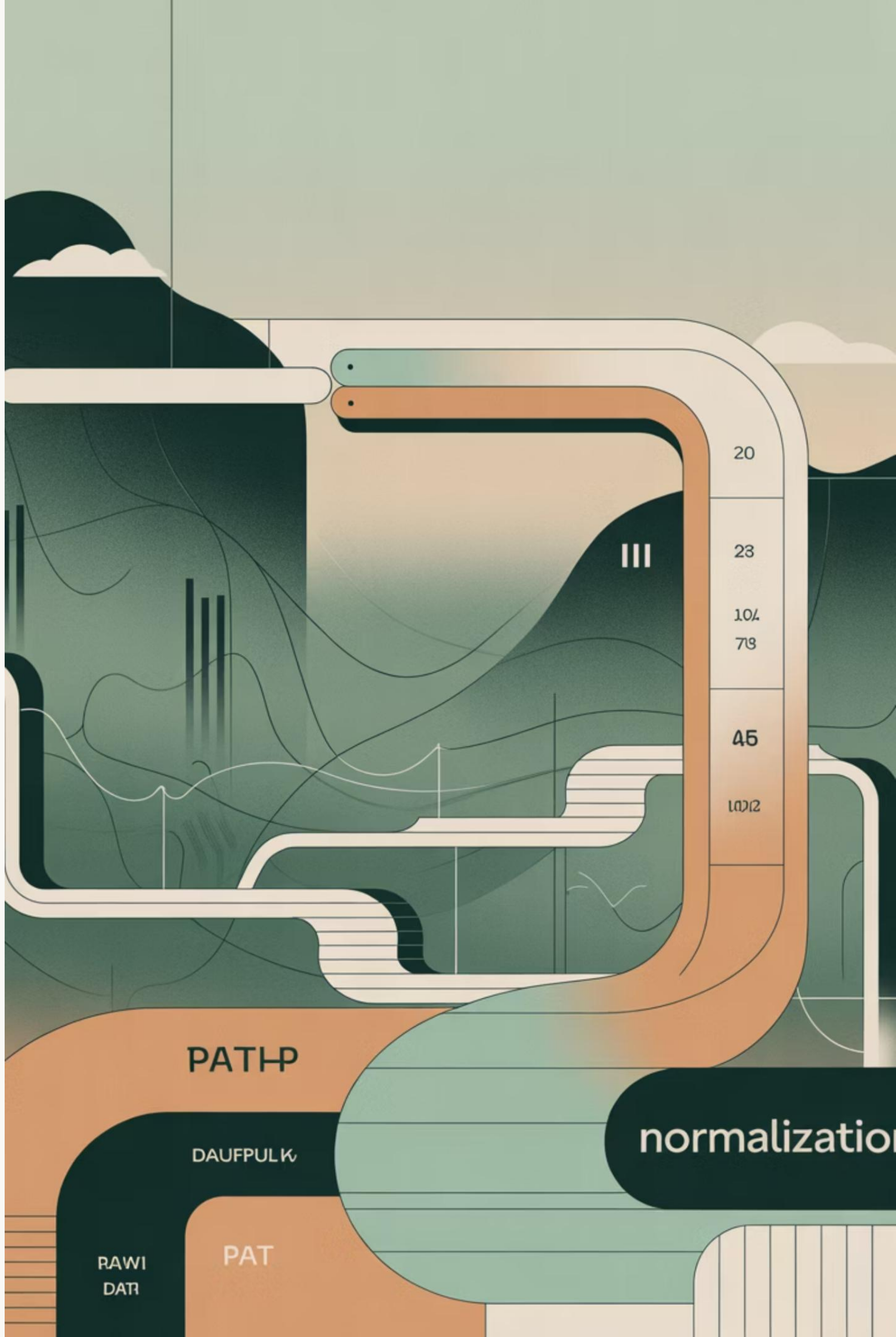
2

Custom Parameters

DAL Toolbox allows you to specify custom mean and standard deviation values, enabling flexible scaling for domain-specific requirements.

This maintains reversibility while mapping data to any target distribution you need.

Access complete examples:
https://github.com/cefet-rj-dal/daltoolbox/blob/main/transf/normalization_zscore.md



Key Takeaways

Balance is Essential

Normalization ensures all features contribute proportionally to model decisions, preventing scale-related bias.

Min-Max for Bounds

Use Min-Max normalization when you need values in [0,1] range—ideal for neural networks and bounded algorithms.

Z-Score for Centering

Choose Z-Score when outlier preservation matters and algorithms assume normally distributed data, like PCA.

Context-Driven Choice

Select your normalization method based on your algorithm's requirements, data distribution, and presence of outliers.