

DATA 3464: Fundamentals of Data Processing

Numeric Data Transformations

Charlotte Curtis

January 27, 2026

Topic overview

- Why transformations are necessary
- Common transformations
- Dimensionality reduction

Resources used:

- [Feature Engineering Chapter 6](#)
- Hands on Machine Learning with Scikit-Learn and Tensorflow/PyTorch, Chapter 4.
Available at [MRU Library](#)
- [Scikit-learn user guide: Chapter 7](#)
- Introduction to Machine Learning with Python. Available at [MRU Library](#)

Common 1:1 transformations

"Most models work best when each feature (and in regression also the target) is loosely Gaussian distributed" -- Introduction to Machine Learning with Python

- Scaling: normalization or standardization
- Nonlinear transforms: log, square root, polynomial
- Fancier methods: Box-Cox, Yeo-Johnson

A brief intro to gradient descent

- Many linear models minimize some cost function through **gradient descent**
- The **gradient** is a vector of partial derivatives

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}$$

for some scalar-valued $f(\mathbf{x})$

Descending the gradient

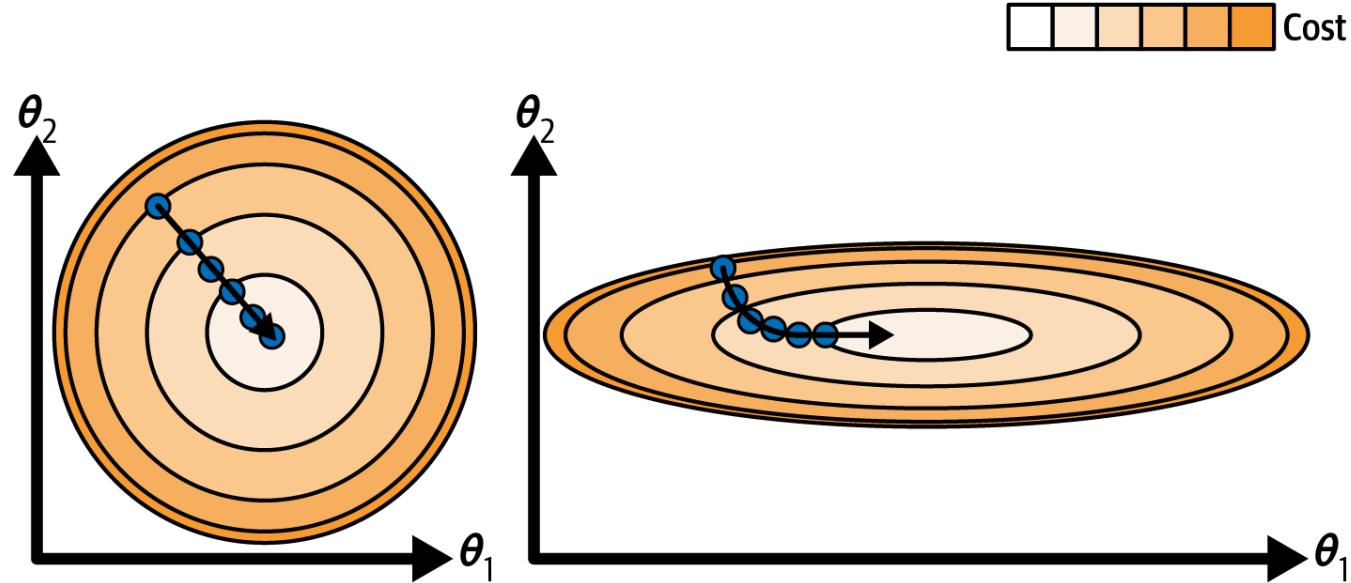
For a loss (or cost) function such as $MSE(\theta) = \frac{1}{m}(\mathbf{X}\theta - \mathbf{y})^T(\mathbf{X}\theta - \mathbf{y})$

1. Start with a random θ
2. Calculate the gradient ∇_{θ} for the current θ
3. Update θ as $\theta = \theta - \eta \nabla_{\theta}$
4. Repeat 2-3 until some stopping criterion is met

where η is the **learning rate**, or the size of step to take in the direction opposite the gradient.

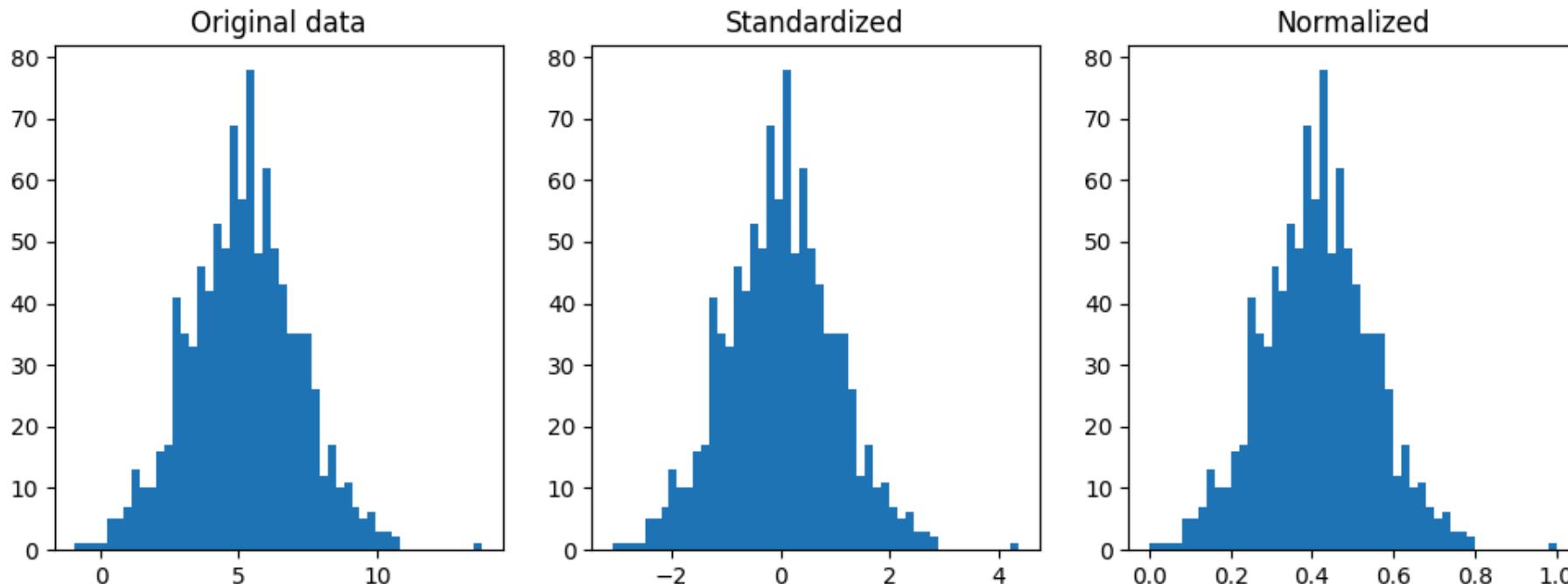
Visualizing in 2D

- The gradient has $m + 1$ dimensions, where m is the number of features
- step size η is a scalar parameter



Main takeaway: feature should be more or less on the same scale

Approaches to scaling



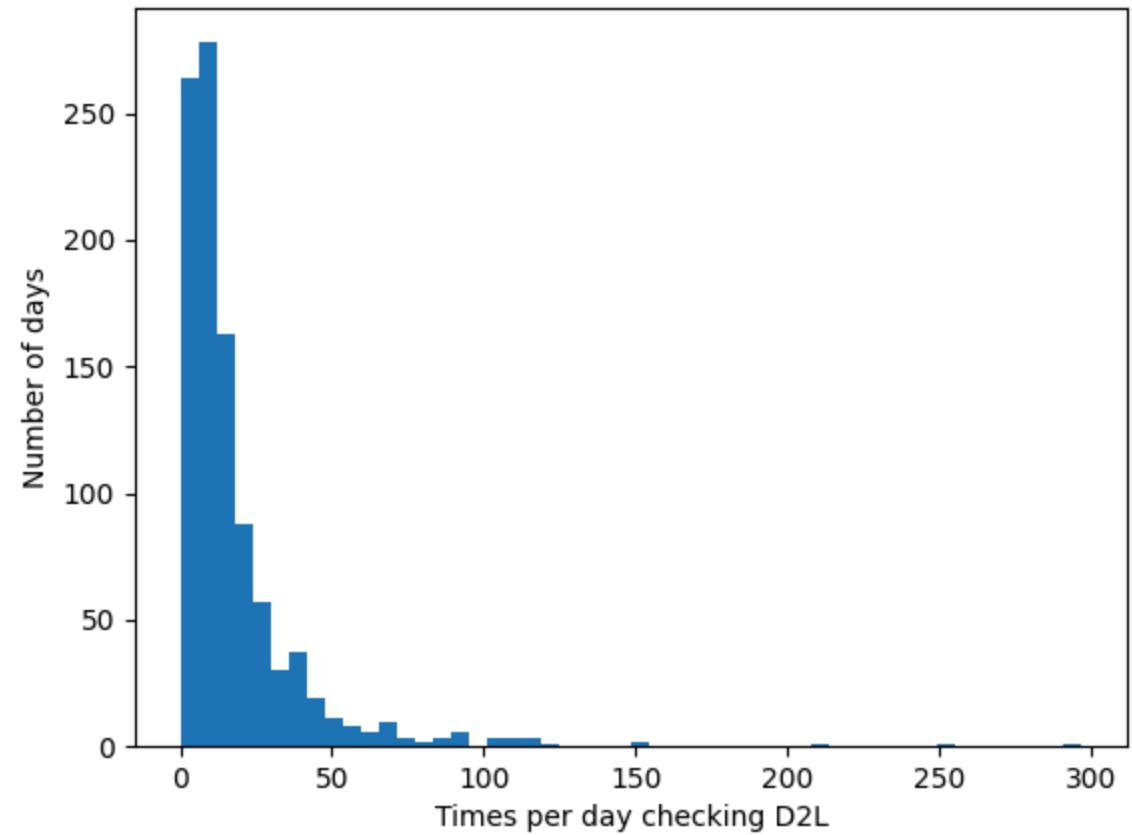
$$\text{Standardize: } x_{scaled} = \frac{x - \mu_x}{\sigma_x}$$

Normalize:

$$x_{scaled} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Nonlinear transforms

- Common case: count data
- Example: how does a student check D2L?
- Not a Gaussian distribution! (not that it strictly needs to be)



Pipelines in Scikit-learn
