Data Preprocessing

Transformation & Reduction

Techniques

Site Preparation

"Before building a house, you need to prepare the site."

Just as construction requires leveling ground and organizing tools, data mining requires transformation and reduction.

Today's tools are the bulldozers and cranes of data science.

Learning Objectives

- Apply **normalization** techniques to prepare data for analysis.
- Implement **discretization** methods for continuous data.
- Understand **sampling** strategies for large datasets.
- Apply dimensionality reduction techniques like **PCA**.
- Select appropriate transformation methods for different scenarios.

Data Transformation

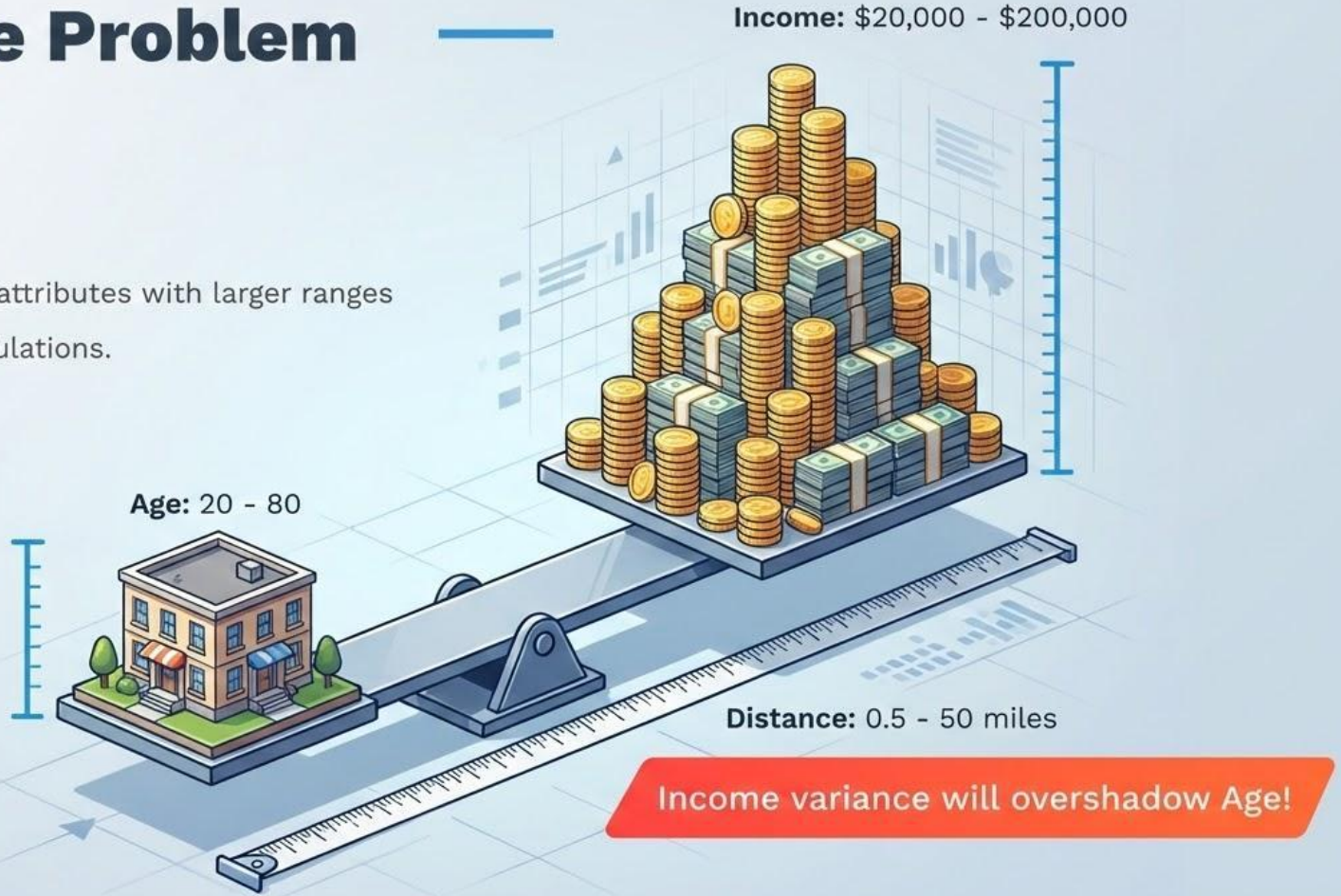
Normalization &

Standardization

The Scale Problem

Why Normalize?

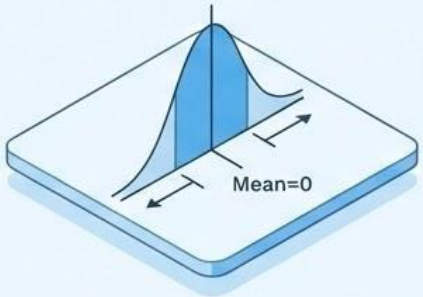
Without normalization, attributes with larger ranges dominate distance calculations.



Normalization Techniques

Min-Max Normalization

Scales data to [0, 1]. Preserves relationships but is sensitive to outliers.



Z-Score Standardization

Scales to Mean=0, Std=1. Handles outliers well but has no fixed range.

Decimal Scaling

Moves the decimal point based on the max absolute value.

$$v' = \frac{(v - min_a)}{(max_A - min_A)} \times (new_{max} - new_{min}) + new_{min}$$

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

$$v' = \frac{v}{10^j}$$ where j = smallest integer such that $max(|v'|) < 1$

Example: Age 35, range 20-80, scale to 0-1

```
text

v' = (35-20)/(80-20) = 15/60 = 0.25
```

Example: Test scores: μ=70, σ=10, score=85

```
text

z = (85-70)/10 = 1.5
```

Example: Max value = 985 → divide by 1000 (j=3)

Example 2.26. Min-max normalization. Suppose that the minimum and maximum values for the attribute *income* are \$12,000 and \$98,000, respectively. We would like to map *income* to the range [0.0, 1.0]. By min-max normalization, a value of \$73,600 for *income* is transformed to $\frac{73,600-12,000}{98,000-12,000}(1.0-0)+0=0.716$. □

Example 2.28. Decimal scaling. Suppose that the recorded values of A range from -986 to 917. The maximum absolute value of A is 986. To normalize by decimal scaling, we therefore divide each value by 1000 (i.e., $j=3$) so that -986 normalizes to -0.986 and 917 normalizes to 0.917. □

Example 2.27. z-score normalization. Suppose that the mean and standard deviation of the values for the attribute *income* are \$54,000 and \$16,000, respectively. With z-score normalization, a value of \$73,600 for *income* is transformed to $\frac{73,600-54,000}{16,000} = 1.225$. □

A variation of this z-score normalization replaces the standard deviation of Eq. (2.33) by the *mean absolute deviation* of A. The *mean absolute deviation* of A, denoted s_A, is

$$s_A = \frac{1}{n}(|v_1 - \bar{A}| + |v_2 - \bar{A}| + \cdots + |v_n - \bar{A}|). \tag{2.34}$$

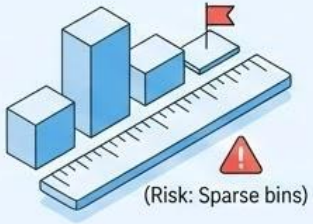
Thus z-score normalization using the mean absolute deviation is

$$v_i' = \frac{v_i - \bar{A}}{s_A}. \tag{2.35}$$

The mean absolute deviation, s_A, is more robust to outliers than the standard deviation, σ_A. When computing the mean absolute deviation, the deviations from the mean (i.e., $|x_i - \bar{x}|$) are not squared; hence, the effect of outliers is somewhat reduced.

Discretization

Converting continuous data into categorical buckets. Useful for algorithms like Decision Trees and Naive Bayes.

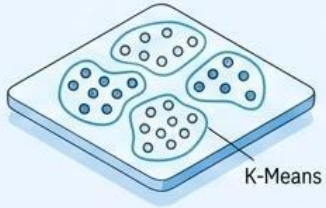


Equal-Width

Divide range into N equal intervals.

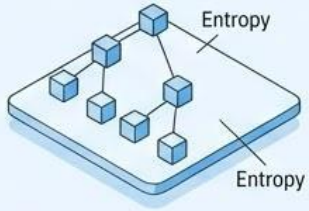
Equal-Frequency

Each bin has the same count.



Clustering-Based

Use K-Means to find natural groups.



Decision Tree

Use entropy for optimal splits.

Example: Credit Scores

Continuous: 300 - 850

Discrete Categories:

- Poor (300-579)
- Fair (580-669)
- Good (670-739)
- Very Good (740-799)
- Excellent (800-850)

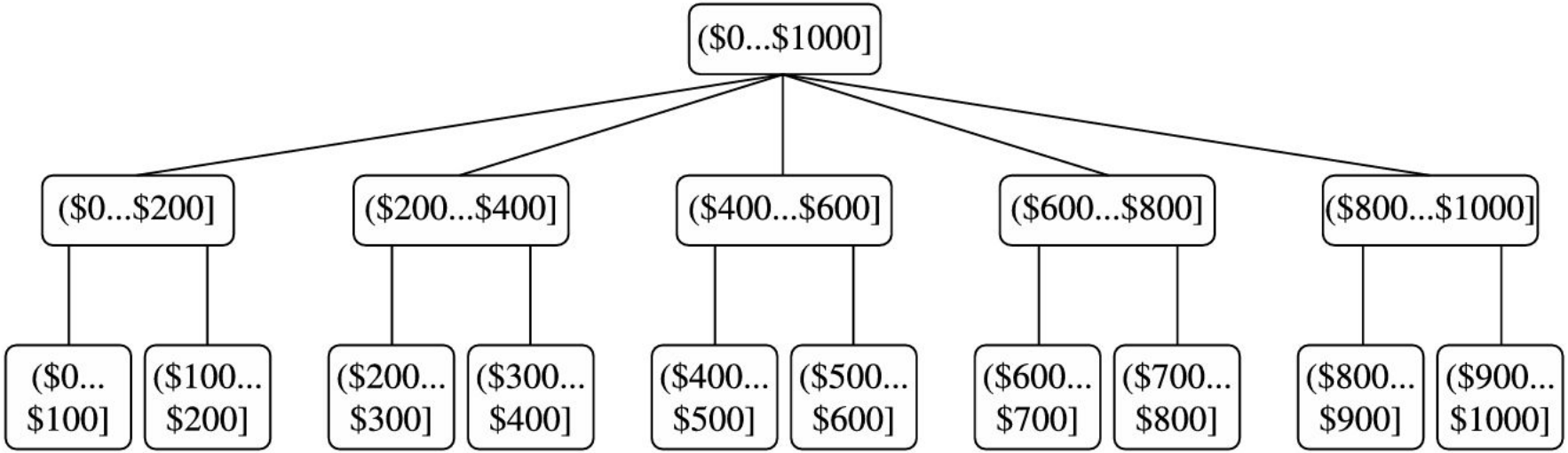


FIGURE 2.13

A concept hierarchy for the attribute *price*, where an interval ($\$X \ldots \$Y$] denotes the range from $\$X$ (exclusive) to $\$Y$ (inclusive).

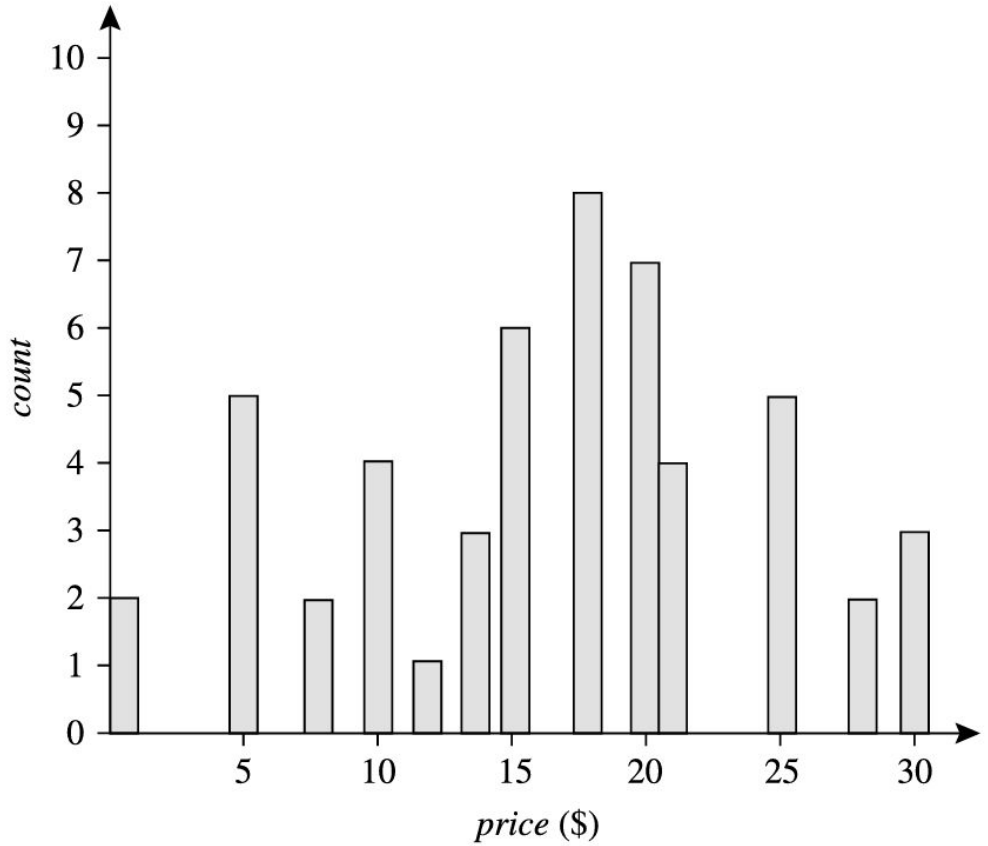


FIGURE 2.14

A histogram for *price* using singleton buckets—each bucket represents one price–value/frequency pair.

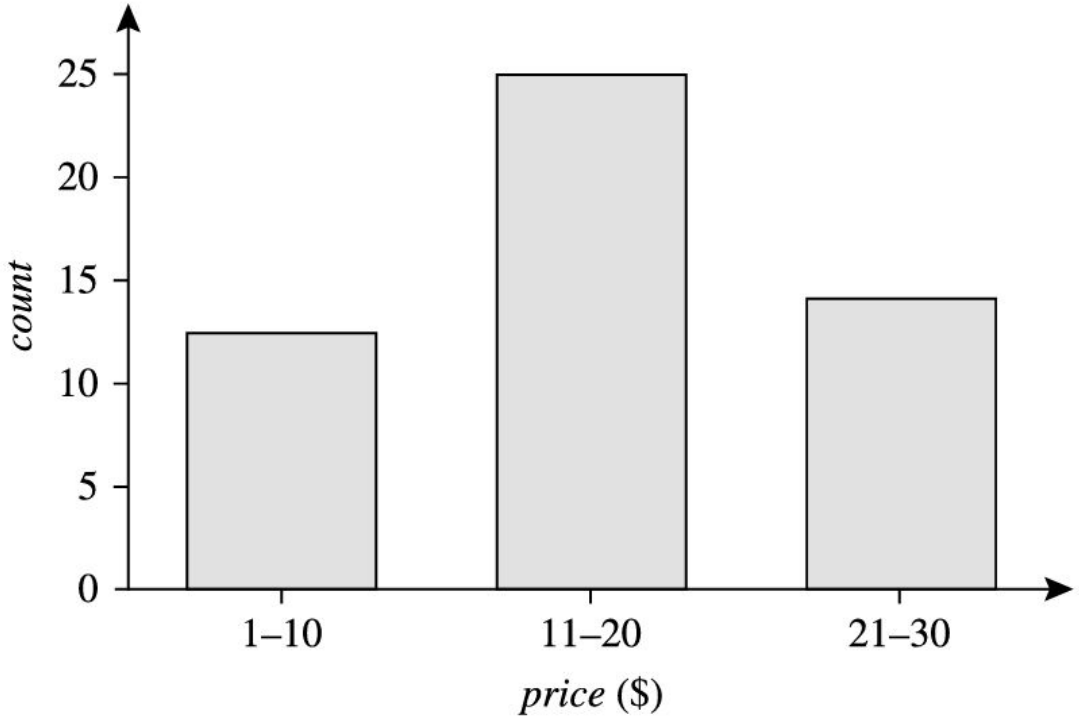


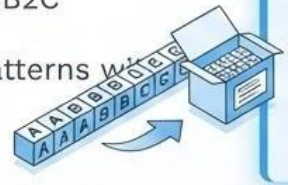
FIGURE 2.15

An equal-width histogram for *price*, where values are aggregated so that each bucket has a uniform width of $10.

Data Compression

Lossless Compression

- **Run-length encoding:** AAAABBBCC → 4A3B2C
- **Dictionary methods:** Replace frequent patterns with codes



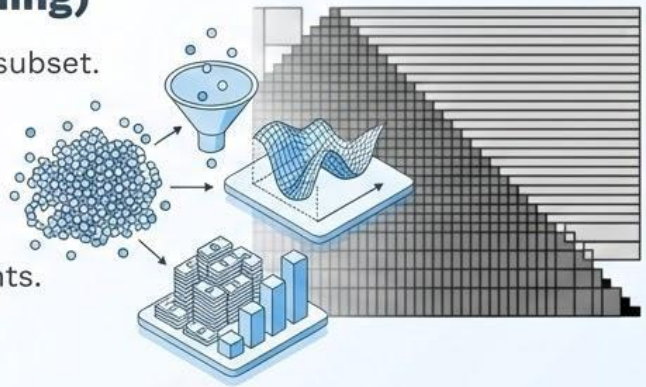
Example: Sensor Data

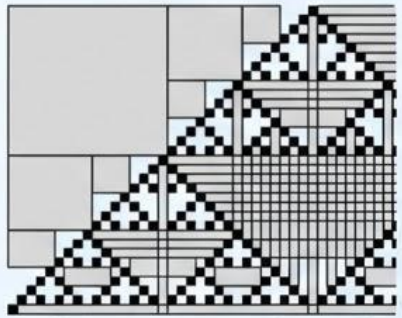
Raw: 23.4, 23.5, 23.7, 23.6, 23.5, 23.8, 23.9...

Compressed: Mean=23.6, Std=0.2, N=1000

Lossy Compression (Data Mining)

- **Sampling:** Working with a representative subset.
- **Dimensionality reduction:** PCA, etc.
- **Aggregation:** Daily sales → Monthly sales.
- **Histograms:** Replace values with bin counts.





The **discrete wavelet transform (DWT)** is a linear signal processing technique that, when applied to a data vector $\boldsymbol{x}$, transforms it to a numerically different vector, $\boldsymbol{x}'$, of **wavelet coefficients**. The two vectors are of the same length. When applying this technique to data reduction, we consider each tuple as an n-dimensional data vector, that is, $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, depicting n measurements made on the tuple from n database attributes.

"How can this technique be useful for data reduction if the wavelet transformed data are of the same length as the original data?" The usefulness lies in the fact that the wavelet transformed data can be truncated. A compressed approximation of the data can be retained by storing only a small fraction of the strongest wavelet coefficients. For example, all wavelet coefficients larger than some user-specified threshold can be retained. All other coefficients are set to 0. The resulting data representation is therefore very sparse, so that operations that can take advantage of data sparsity are computationally very fast if performed in wavelet space. The technique also works to remove noise without smoothing out the main features of the data, making it effective for data cleaning as well. Given a set of coefficients, an approximation of the original data can be constructed by applying the *inverse* of the DWT used.

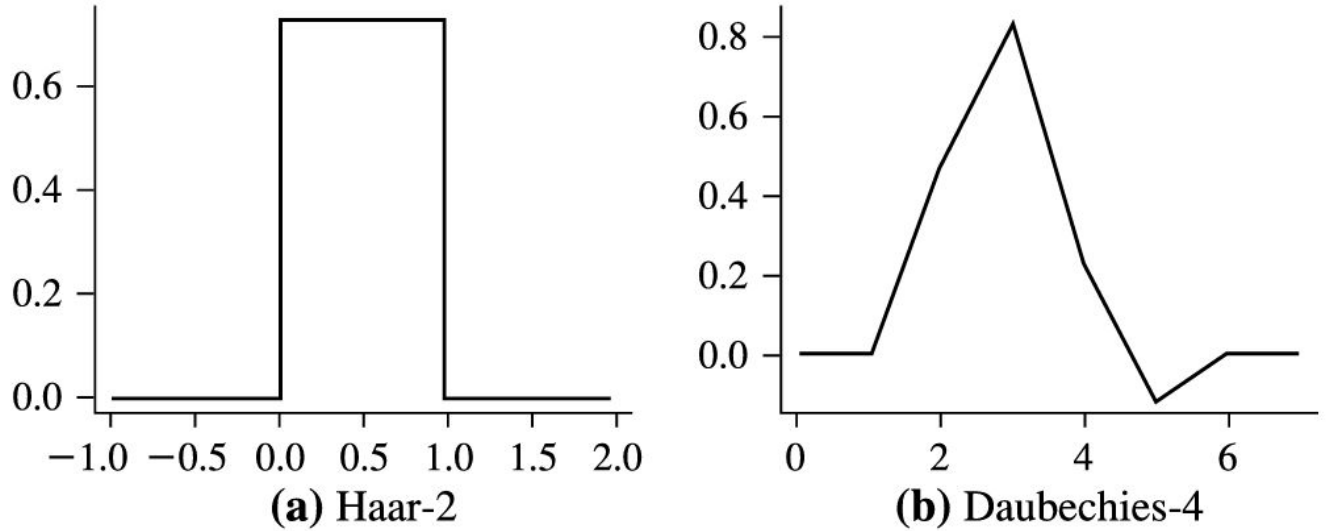
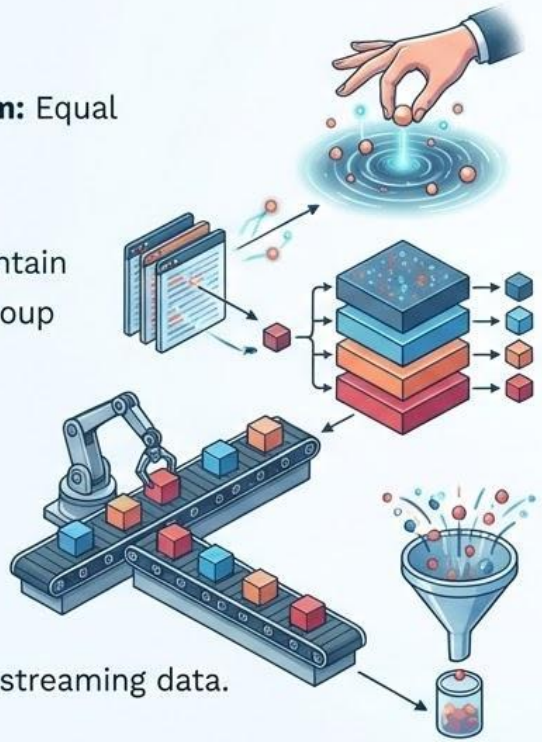


FIGURE 2.16

Examples of wavelet families. The number next to a wavelet name is the number of *vanishing moments* of the wavelet. This is a set of mathematical relationships that the coefficients must satisfy and is related to the number of coefficients.

Sampling Strategies

- **Simple Random:** Equal probability.
- **Stratified:** Maintain Maintain subgroup proportions.
- **Systematic:** Every k-th element.
- **Reservoir:** For streaming data.



⚠ Sampling Bias Warning

- **Survivorship bias:** Only sampling successful companies.
- **Time-period bias:** Only weekday data, missing weekends.
- **Selection bias:** Only voluntary survey respondents.

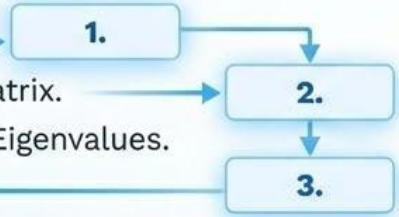
Dimensionality Reduction

Combatting the Curse of Dimensionality

PCA: Finding the Best View

Intuition: Find the angle that shows the mo
information (variance).

1. Standardize data.
2. Calculate Covariance Matrix.
3. Compute Eigenvectors/Eigenvalues.
4. Select top k vectors.



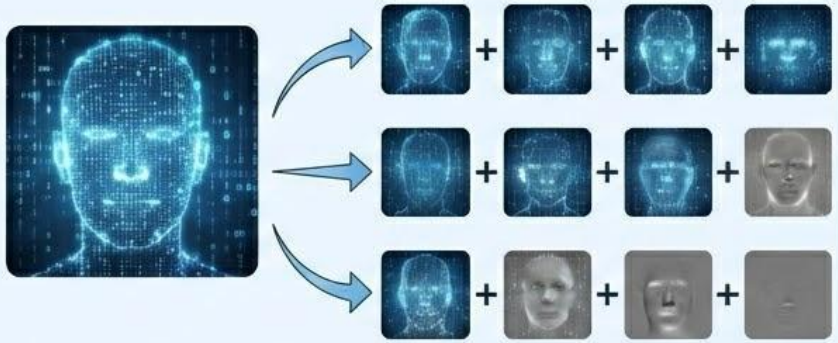
Kaiser Criterion

Keep PCs with eigenvalue > 1

Real Application: Face Recognition

Each face is a high-dimensional pixel vector. PCA finds "face components".

New face = combination of eigenfaces



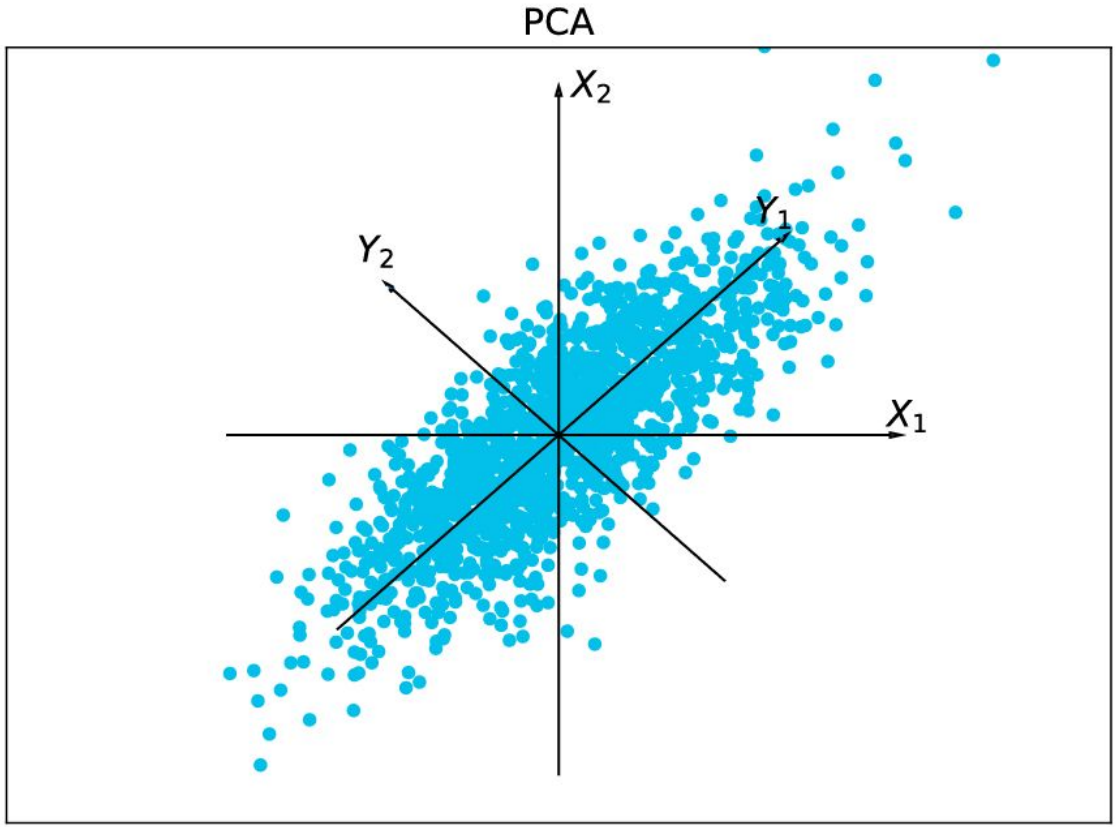


FIGURE 2.17

Principal components analysis. $\boldsymbol{Y}_1$ and $\boldsymbol{Y}_2$ are the first two principal components for the given data.

Attribute Subset Selection

Method Type	Techniques	Description
Filter	Variance Threshold, Correlation, Chi-square	Pre-processing steps to remove low-variance or highly correlated features.
Wrapper	Forward Selection, Backward Elimination	Uses a learning algorithm to evaluate feature sets iteratively.
Embedded	LASSO Regression, Random Forest	Feature selection is built directly into the algorithm training process.

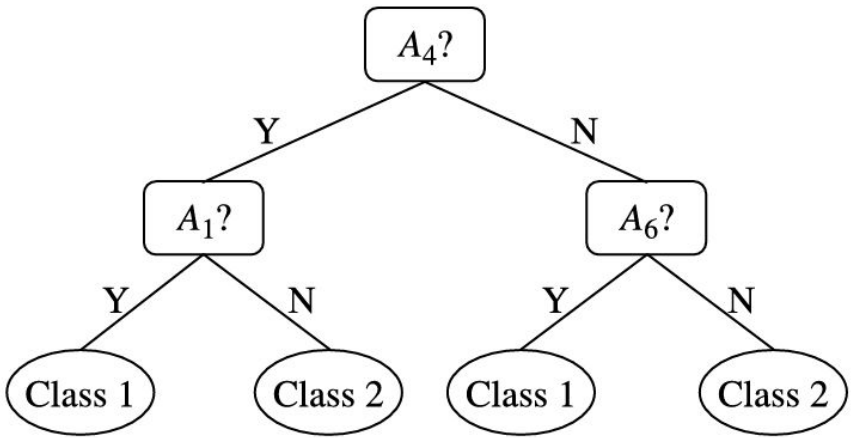
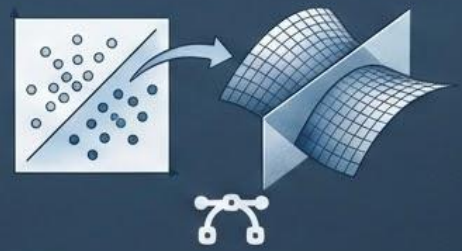
Forward selection	Backward elimination	Decision tree induction
Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ Initial reduced set: $\{\}$ => $\{A_1\}$ => $\{A_1, A_4\}$ => Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ => $\{A_1, A_3, A_4, A_5, A_6\}$ => $\{A_1, A_4, A_5, A_6\}$ => Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$  => Reduced attribute set: $\{A_1, A_4, A_6\}$

FIGURE 2.18

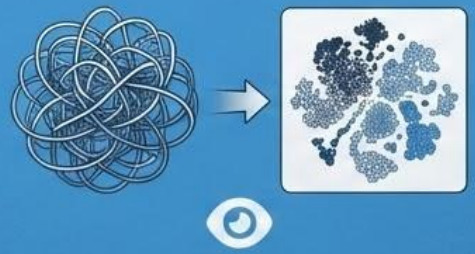
Greedy (heuristic) methods for attribute subset selection.

When Linear Isn't Enough



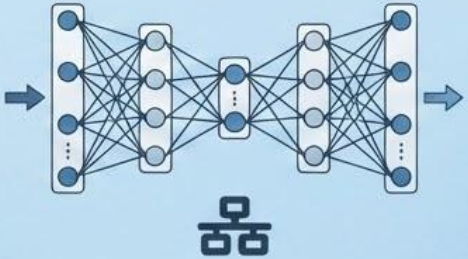
Kernel PCA

Maps data to higher dimensions to find linear separators. Good for complex patterns.



t-SNE

Excellent for 2D/3D visualization. Preserves local structure but not for feature extraction.



Autoencoders

Neural networks that compress (encode) and reconstruct (decode) data. Learns efficient representations.

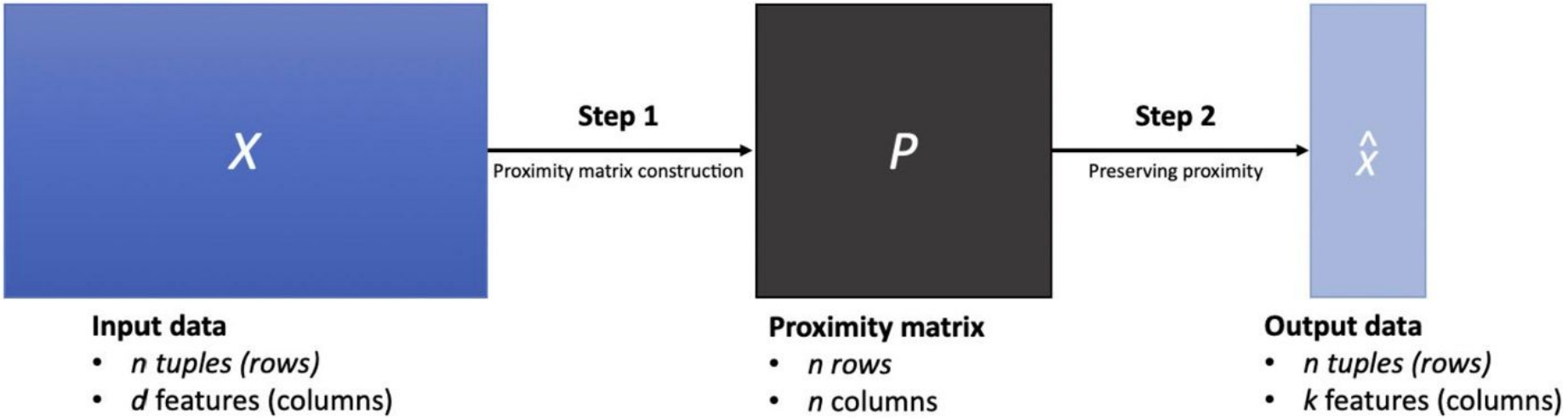


FIGURE 2.19

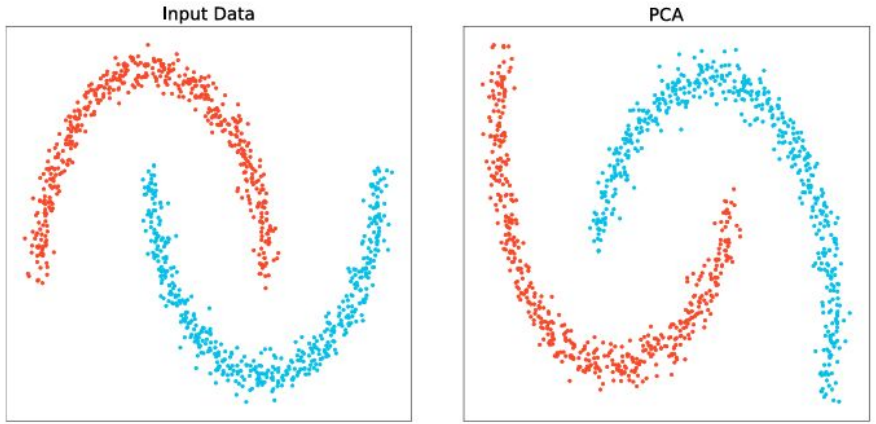
An illustration of nonlinear dimensionality reduction.

FIGURE 2.19

An illustration of nonlinear dimensionality reduction.

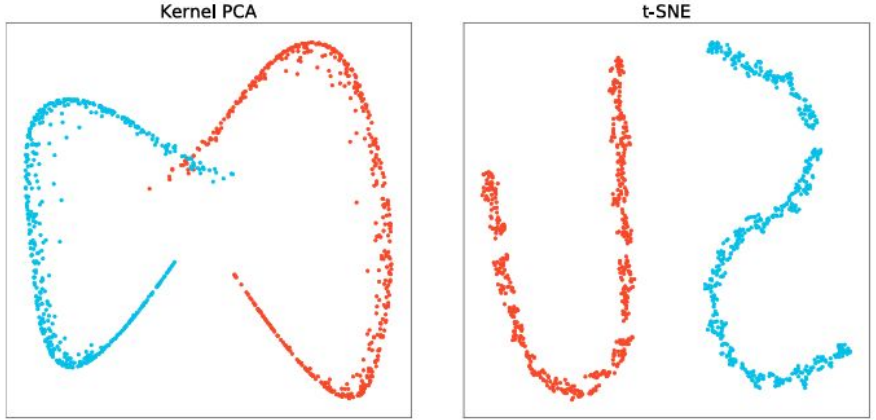
Table 2.8 Comparison of KPCA and SNE.

	Step 1: Proximity Construction	Step 2: Preserving Proximity
KPCA	$P(i,j) = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$	$\min \sum_{i,j=1}^{n} (P(i,j) - \hat{P}(i,j))^2 = \|P - \hat{P}\|_{fro}^2$
SNE	$P(i,j) = \frac{e^{-d_{ij}^2}}{\sum_{l=1, l\neq i}^{n} e^{-d_{il}^2}}$	$\min \sum_{i=1}^{n} \mathrm{KL}(P_i \| \hat{P}_i)$



(a) Input data

(b) PCA

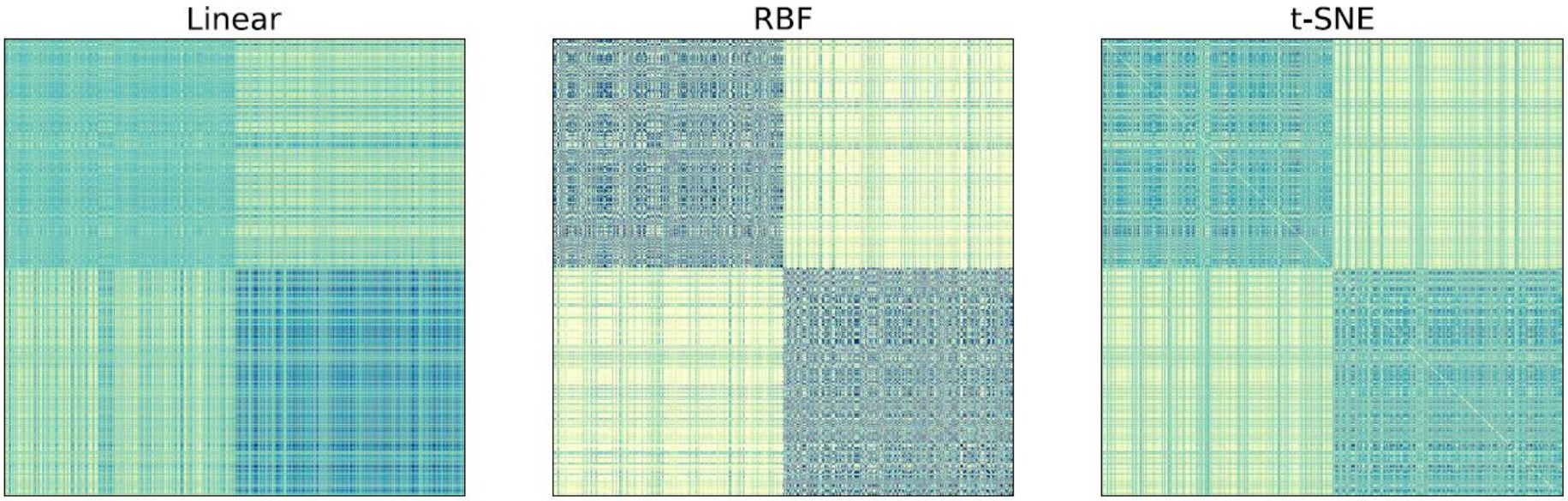


(c) KPCA

(d) t-SNE

FIGURE 2.20

An example of linear vs. nonlinear dimensionality reduction methods.



(a) PCA (b) KPCA t-SNE

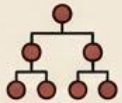
FIGURE 2.21

The heatmaps of the similarity or proximity matrices in PCA (a), KPCA (b), and t-SNE (c), respectively. The two diagonal blocks correspond to the two clusters in Fig. 2.20.

Decision Framework

K-Means Clustering

Use Min-Max normalization.



Decision Trees

Use Discretization.

Neural Networks

Use Z-Score standardization.

Visualization

Use t-SNE or PCA.

Common Pitfalls to Avoid

Bad Split

Normalization after train/test split. Always calculate min/max/mean from training set only!

Info Leakage

Don't use test data in any transformation parameter estimation.

Over-reduction

Don't lose important patterns in pursuit of simplicity.

Wrong Tech

Linear PCA won't capture circular patterns.

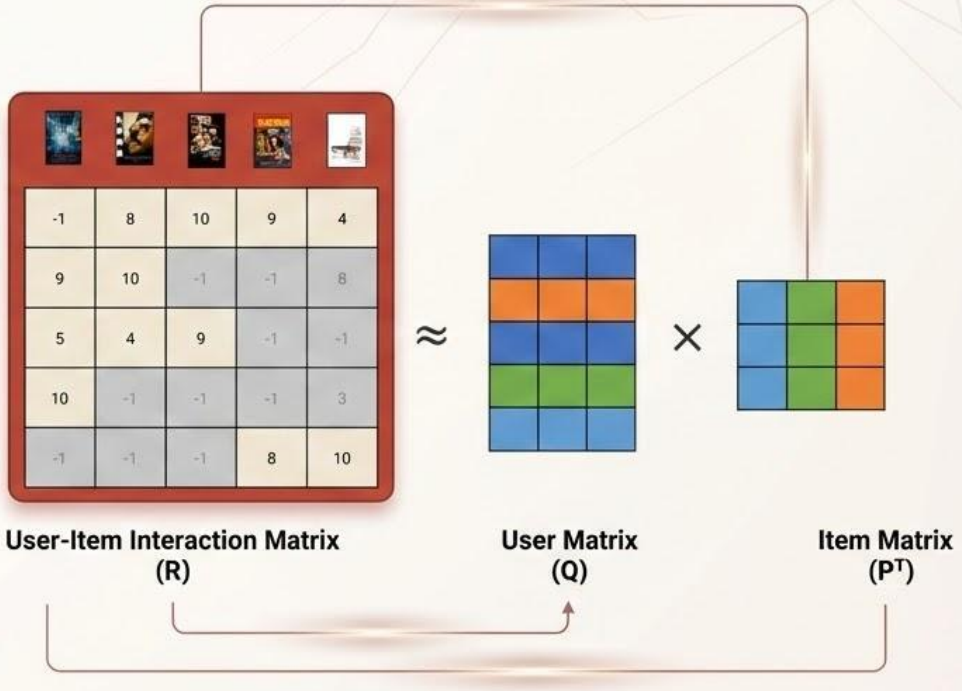
Case Study: Netflix Recommendation

Data Challenges

- Millions of users × thousands of movies = massive matrix.
- Mostly empty (sparse matrix).
- Different rating scales among users.

Solutions Applied

- **Normalization:** Adjust for user rating bias.
- **Dim. Reduction:** SVD (Matrix Factorization) for latent factors.
- **Sampling:** Subsets for testing.
- **Selection:** Genre, director, actors.



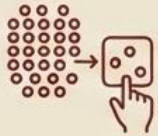
Key Takeaways

Normalization

Normalization ensures fair comparison across features.

Discretization

Discretization enables categorical algorithms on continuous data.



Sampling

Sampling makes large datasets manageable.

PCA

PCA finds the most informative linear projections.

Feature Selection

Feature Selection removes noise and redundancy.

Choose methods based on:

Data characteristics, algorithm requirements, and computational constraints.