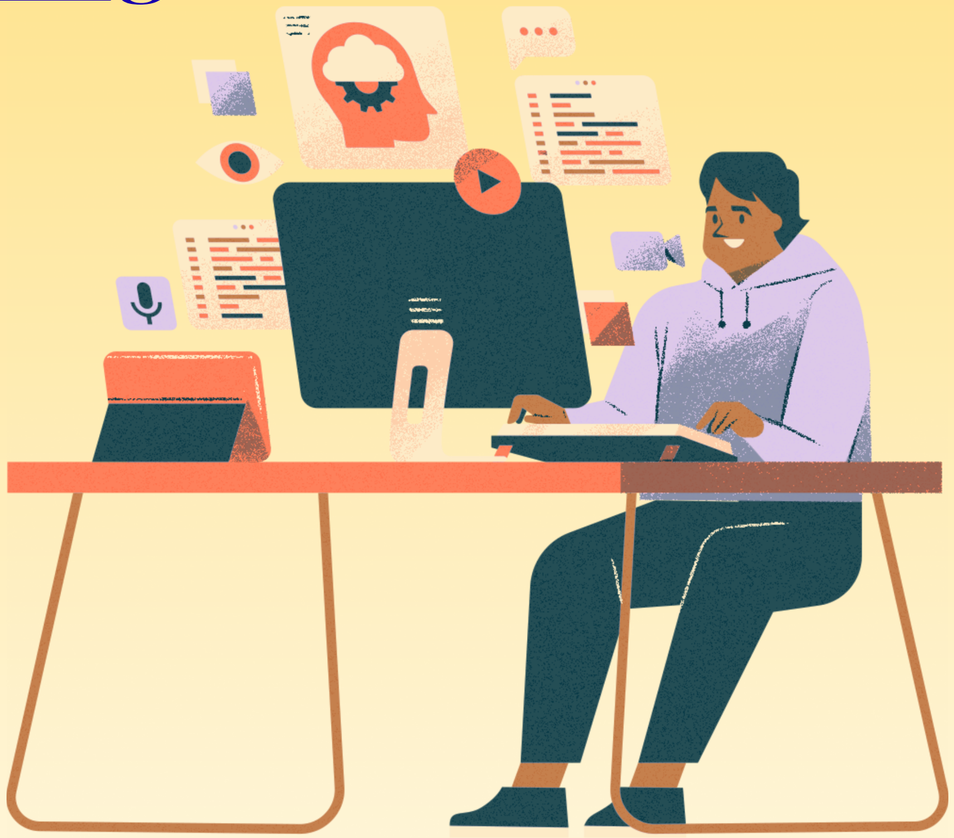
Machine Learning

Feature Engineering

unit 2



Contents

- Concept of Feature
- Preprocessing of data: Normalization and Scaling, Standardization, Managing missing values, Introduction to Dimensionality Reduction, Principal Component Analysis (PCA)
- Feature Extraction: Kernel PCA, Local Binary Pattern.
- Introduction to various Feature Selection Techniques :Sequential Forward Selection Sequential Backward Selection
- Statistical feature engineering: count-based, Length, Mean, Median, Mode etc. based feature vector creation.
- Multidimensional Scaling
- Matrix Factorization Techniques.

Concepts Of Features

A feature is an individual measurable property or characteristic of the data.

Features are the input variables used by machine learning models to make predictions.

Examples

- In house price prediction:
  - Size of house (sq. ft.), location, number of rooms, age of house ↠ features.
- In healthcare:
  - Age, blood pressure, cholesterol level ↠ features.

Types of Features

- Numerical – Continuous (height, weight) or Discrete (number of children).
- Categorical – Labels (gender, city).
- Ordinal – Ordered categories (small, medium, large).
- Derived – Created from existing ones (BMI = weight/height^2).

Preprocessing Of Data

- Datasets are often raw, messy, and inconsistent.

- Problems include:
  Different Scales ↠ (Age: 18–70, Salary: 20k–1L).
  Missing Values ↠ blanks or NaN.
  Noise/Errors ↠ typos, outliers, wrong entries.
  Categorical Data ↠ (e.g., Male/Female) not directly usable.
  Too Many Features ↠ increases complexity.

Why is it Needed?

- Ensures fair comparison among features.
- Makes training faster and more accurate.
- Prevents bias (large-scale features dominating).
- Improves model stability and generalization.

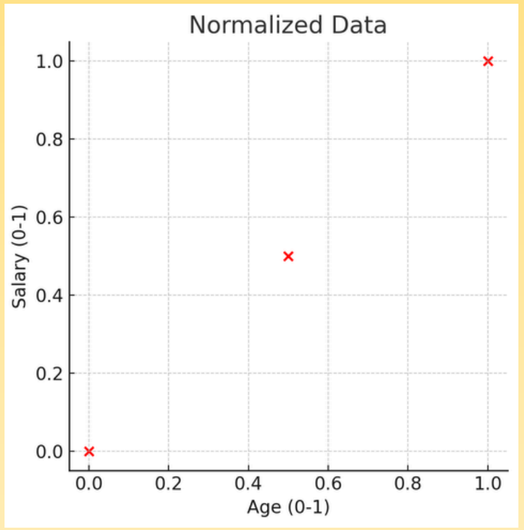
Normalization and Scaling

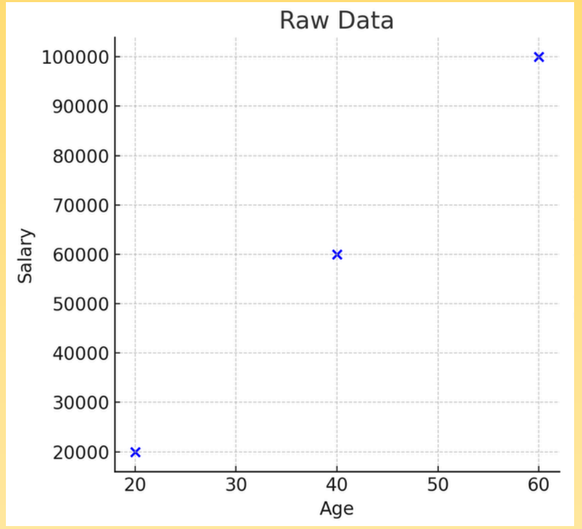
Normalization

- Rescales data into a fixed range (usually 0 to 1).
- Formula:

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

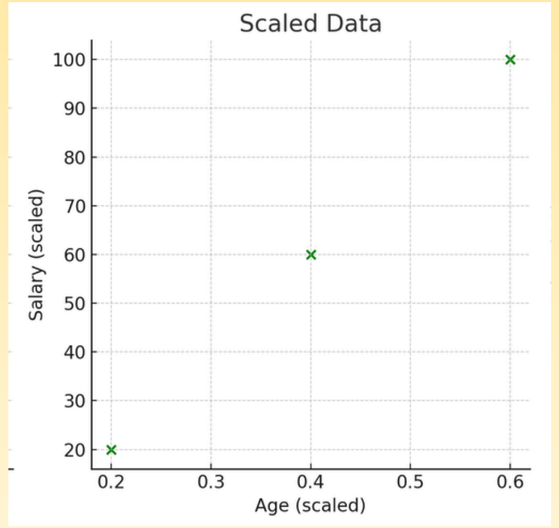
- Example: Marks = 70 (out of 100)
- Normalized = (70−0)/(100−0)=0.7
- Best for: Neural networks





Scaling (Rescaling)

- Adjusts values to a defined range but not always 0–1.
- Example: Converting height (cm ↠ meters).
- Another case: scaling data to [-1, 1] for algorithms like SVM.
- Best for: Algorithms sensitive to feature magnitude.

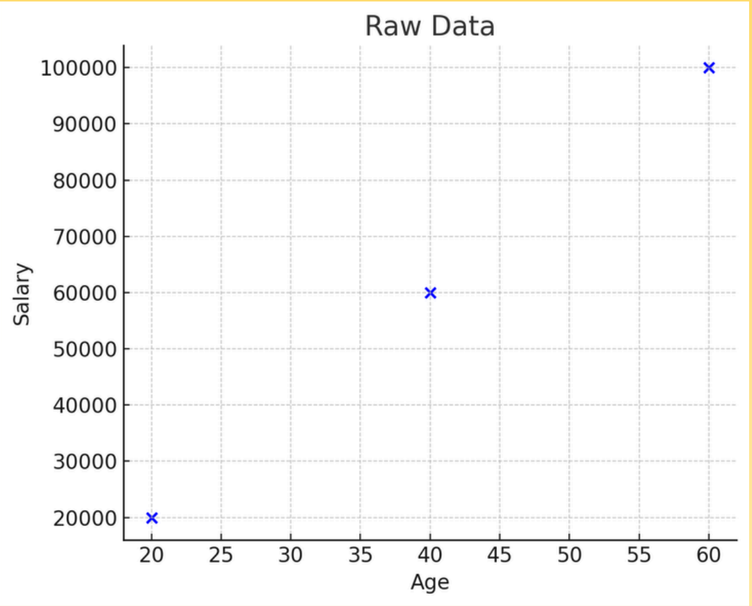


Standardization

- A preprocessing technique that transforms data so that:
- Mean = 0
- Standard Deviation = 1
- Formula:

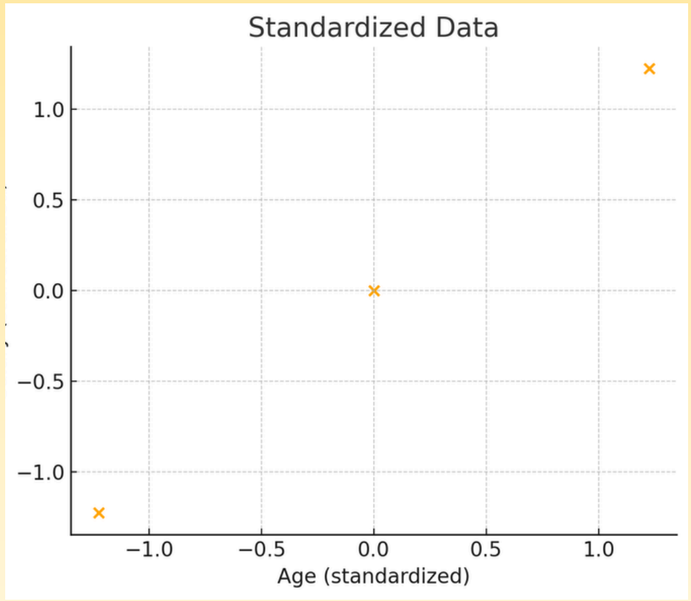
$$X' = \frac{X - \mu}{\sigma}$$

where μ = mean , σ = standard deviation.



Why Do We Use It?

- Features may have different scales and units (e.g., Age: 20–60 vs Income: 20k–1L).
- Models like Logistic Regression, SVM, KNN, PCA perform better when data is standardized.
- Makes optimization algorithms (like Gradient Descent) faster and more stable.



Managing missing values

Why Missing Values Occur?

Real-world data is rarely complete.

Causes:

- Human error (skipped survey answers, typing mistakes).
- Sensor/device failure.
- Data not recorded (e.g., optional fields).

Problems with Missing Values

Many ML algorithms cannot handle NaN (blank) values.

Leads to:

- Incorrect results.
- Biased models.
- Reduced accuracy.

Techniques to Handle Missing Data

Remove Missing Data

- Delete rows/columns with missing values.
- Simple, but ❌may lose useful data.

Imputation (Replacing Missing Values)

- Mean / Median / Mode Imputation
- Numerical data ↠ replace with mean/median.
- Categorical data ↠ replace with mode (most frequent value).
- Example: Age = [20, NaN, 40, 50] ↠ replace NaN with mean (36.6).

Advanced Methods

- KNN Imputation ↠ predict missing values using nearest neighbors.
- Regression Imputation ↠ predict using other features.
- Interpolation ↠ estimate from surrounding values (time-series).

Introduction to Dimensionality Reduction

- **Dimensionality** = number of features (columns/variables) in a dataset.
- **Example**:
- Student dataset: Name, Age, Marks, Attendance ↠ 4 features (4D).
- Image dataset: Each pixel is a feature ↠ can be thousands of dimensions!

- Process of reducing the number of features while keeping important information.
- Converts high-dimensional data ↠ lower-dimensional form.
- Goal: Keep data meaningful but simpler.

Imagine predicting exam marks using:

- Features: Hours studied, Attendance, Notes taken, Sleep hours, Mood, Food habits, etc.

- Not all features are useful.

- Dimensionality reduction finds and keeps only the most important ones (e.g., Hours studied, Attendance).

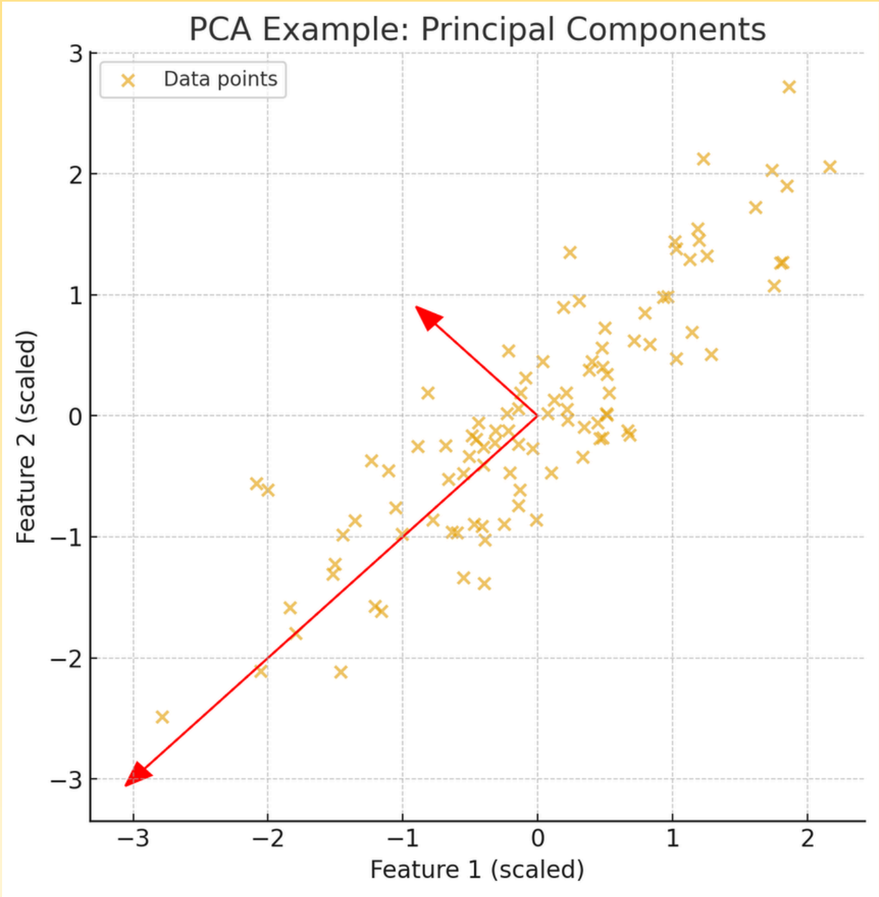
Principal Component Analysis (PCA),

- PCA is a dimensionality reduction technique.
- It finds new axes (directions) called Principal Components that capture maximum variance (information) in the data.
- It projects high-dimensional data into fewer dimensions.

- Instead of keeping all features, PCA creates new features (linear combinations of old ones).
- These new features = Principal Components.
- PC1 captures the most variance, PC2 captures the next most, etc.

Why PCA?

- Reduces dataset size while keeping most information.
- Removes redundancy (correlated features).
- Speeds up ML algorithms.
- Useful for visualizing data in 2D/3D.



Step in PCA

Standardize the Data

Scale features so each has mean =0 and variance=1

Compute Covariance Matrix

Captures how features vary with respect to each other.

Find Eigenvalues & Eigenvectors

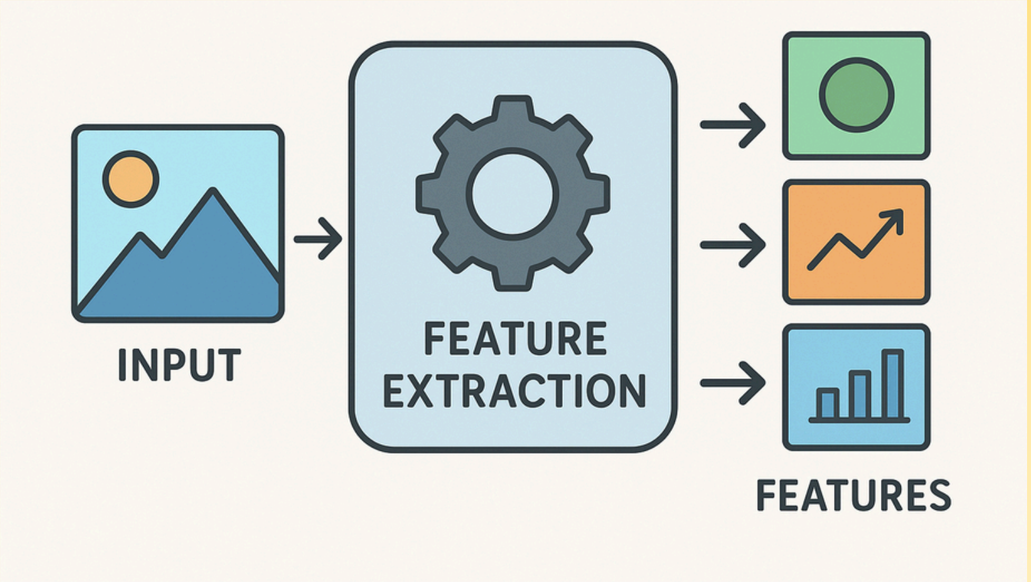
Eigenvectors = directions of maximum variance
Eigenvalues = importance/strength of each eigenvector.

Select Top k Eigenvectors → Principal Components

Project Data onto New Axes

Feature Extraction

Process of transforming raw data into a set of meaningful features that best represent the information for a machine learning model.



Why Feature Extraction?

- Reduces dimensionality of data.
- Improves accuracy of models.
- Reduces noise and redundancy.
- Makes training faster.

Examples

- Image Data ⇾ Extract edges, textures, colors.
- Text Data ⇾ Extract keywords, TF-IDF, embeddings.
- Audio Data ⇾ Extract frequency, pitch, MFCC features.

Kernel PCA (Principal Component Analysis with Kernels)

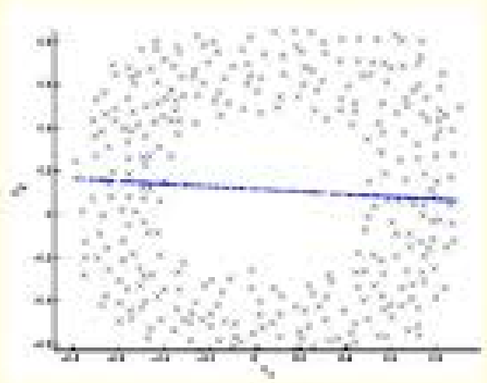
- A nonlinear version of PCA.
- Instead of just straight-line projections, it can capture curved patterns in data.

Why use it:

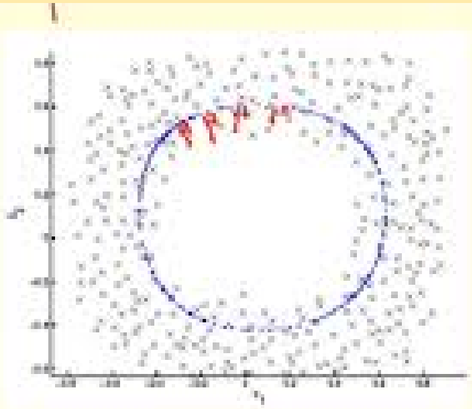
- Works better when data is not linearly separable.
- Useful in image recognition, text classification, and nonlinear datasets.

How it works (simple):

- Apply a kernel function (like polynomial, RBF).
- Map data into higher dimensions.
- Then perform PCA in that space.



PCA



KPCA

Local Binary Pattern (LBP)

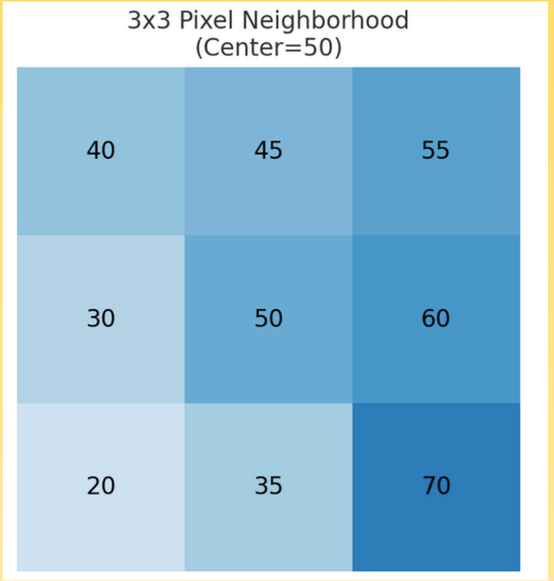
- A texture feature extraction method for images.
- Describes patterns like edges, corners, and spots.

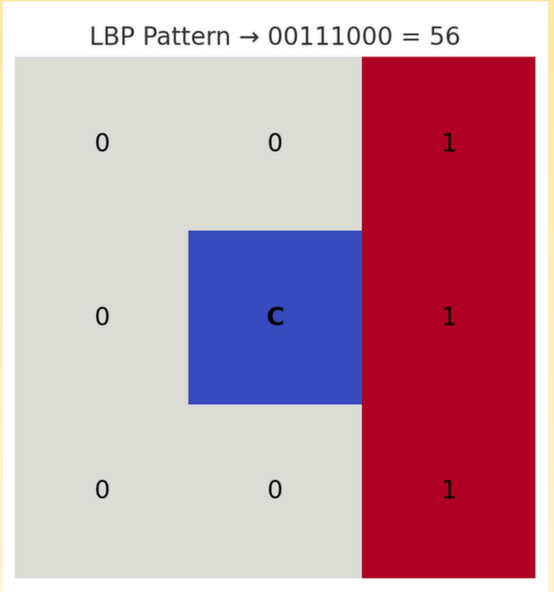
How it works (simple):

- Take a pixel.
- Compare it with surrounding pixels (neighbors).
- If neighbor ≥ center pixel ↠ assign 1, else 0.
- Form a binary number (LBP code).

Why use it:

- Very effective in face recognition, object detection, and texture classification.
- Fast and simple.





Feature Selection Techniques

Feature selection is the process of choosing a subset of the most relevant features (input variables) for model building. The goal is to:

- Reduce dimensionality
- Improve model performance
- Reduce overfitting
- Lower computational cost

Feature selection methods are generally categorized into:

Filter Methods

- Select features based on statistical tests or correlation (independent of ML algorithm).
- Examples: Chi-square test, Information Gain, Correlation Coefficient.

Wrapper Methods

- Use a predictive model to evaluate combinations of features.
- Computationally expensive but more accurate.
- Examples: Sequential Forward Selection, Sequential Backward Selection.

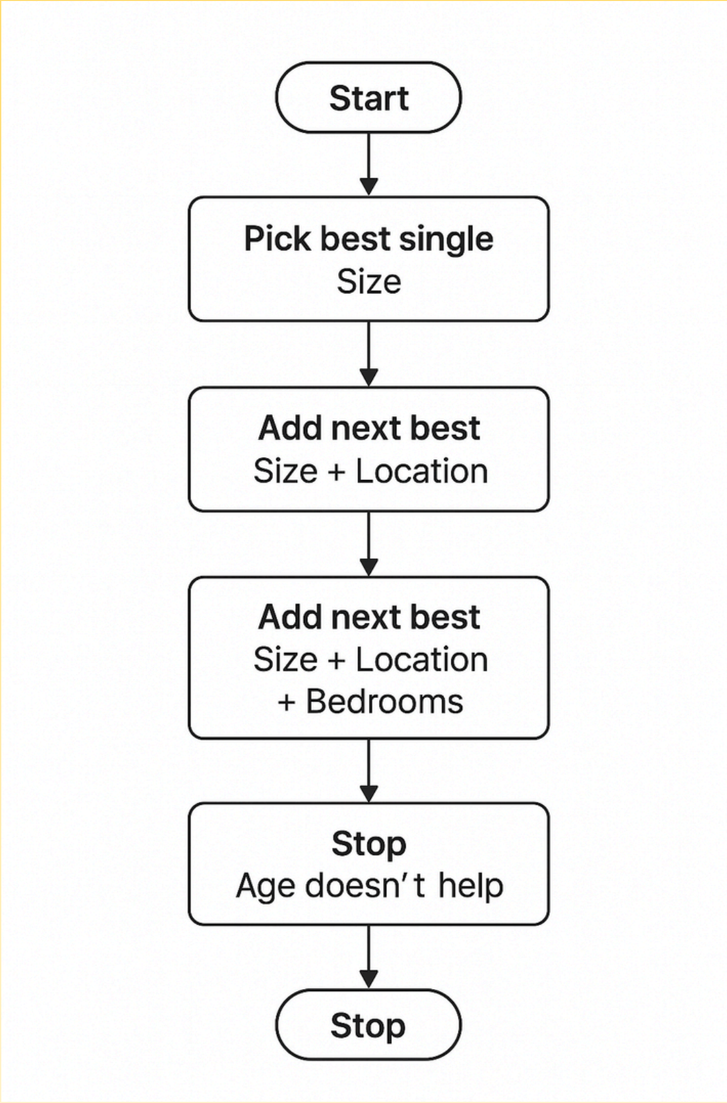
Embedded Methods

- Perform feature selection during model training.
- Examples: LASSO (L1 Regularization), Decision Trees, Random Forest Feature Importance.

Sequential Forward Selection (SFS)

- Starts with no features
- Adds features one by one
- Selects the feature that maximizes model performance at each step
- Stops when:
  - No significant improvement
  - Or desired number of features reached

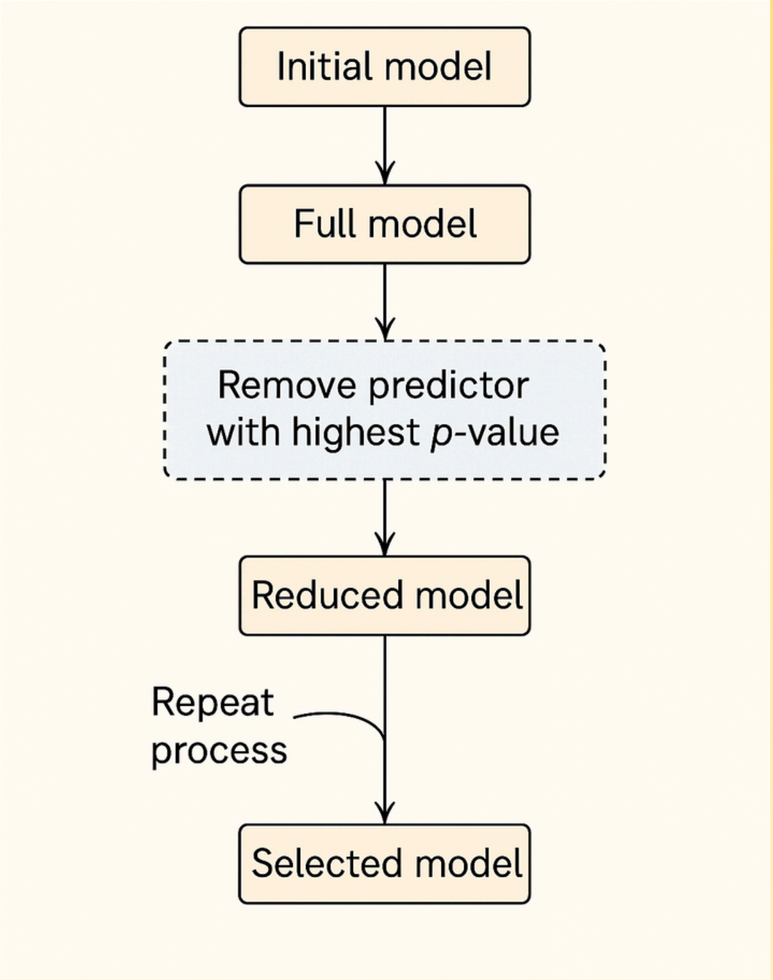
Pros: Simple, interpretable
Cons: Computationally expensive



Sequential Backward Selection (SBS)

- Starts with all features
- Removes features one by one
- Eliminates the feature that has least impact on performance
- Stops when:
  - Desired number of features reached

Pros: Handles irrelevant features well
Cons: Expensive for very large datasets



Statistical feature engineering

1. Count-based Features

- Measure frequency of occurrence
- Example: Number of words, characters, digits, or symbols in text
- Useful in text classification, sentiment analysis, spam detection
- Highlights repetition or emphasis patterns

2. Length-based Features

- Capture the size of data elements
- Examples: Length of a sentence, paragraph, document, or sequence
- Indicator of complexity or verbosity
- Helps in readability analysis and fraud detection

3. Mean-based Features

- Average value of a set of numbers/features
- Reduces data to central tendency
- Example: Mean word length in a sentence, mean transaction amount
- Smooths noise, gives generalized behavior

4. Median-based Features

- Middle value when data is ordered
- Robust against outliers (better than mean for skewed data)
- Example: Median response time, median salary in dataset
- Represents typical value in non-normal distributions

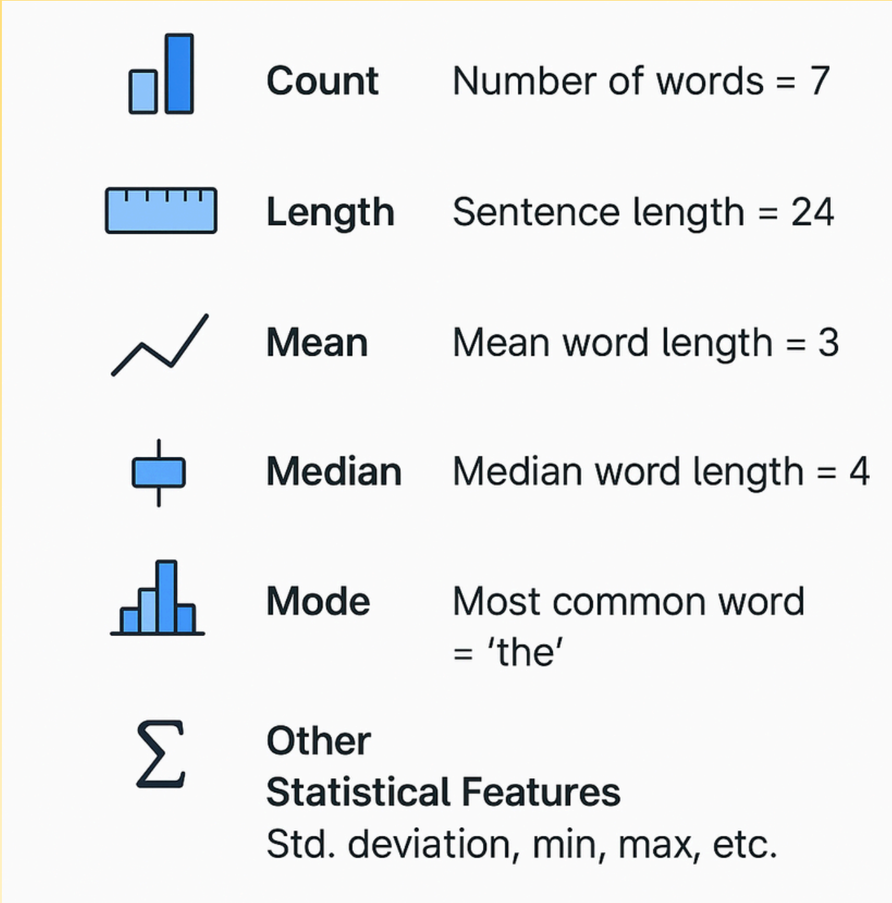
Statistical feature engineering

5. Mode-based Features

- Most frequently occurring value in a dataset
- Useful for categorical or discrete data
- Example: Most common word in a document, most frequent transaction type
- Captures dominant pattern/behavior

6. Other Statistical Features

- Standard Deviation / Variance ↠ Measures spread of data
- Min / Max values ↠ Captures boundaries
- Quantiles / Percentiles ↠ Summarize distribution
- Skewness & Kurtosis ↠ Shape of distribution



Multidimensional Scaling (MDS)

Definition: A dimensionality reduction technique that represents high-dimensional data in a low-dimensional space (usually 2D/3D).

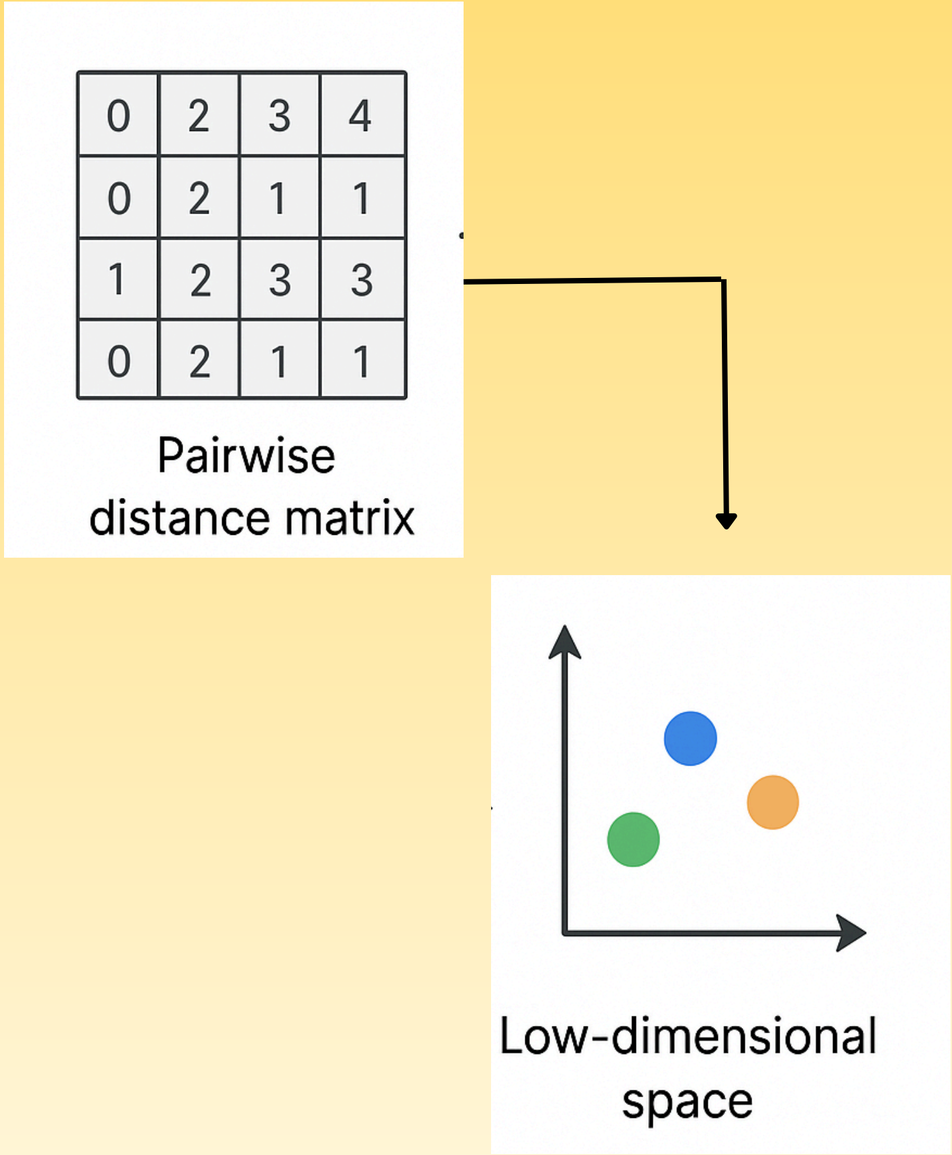
Goal: Preserve pairwise distances or similarities between data points.

Steps:

1. Compute dissimilarity (distance) matrix
2. Apply eigen decomposition
3. Project into low-dimensional space

Applications:

- Visualization of similarity/dissimilarity
- Psychometrics (perceptual mapping)
- Bioinformatics (gene expression data)



Matrix Factorization Techniques

Definition: Decompose a large matrix into smaller matrices to capture hidden structures.

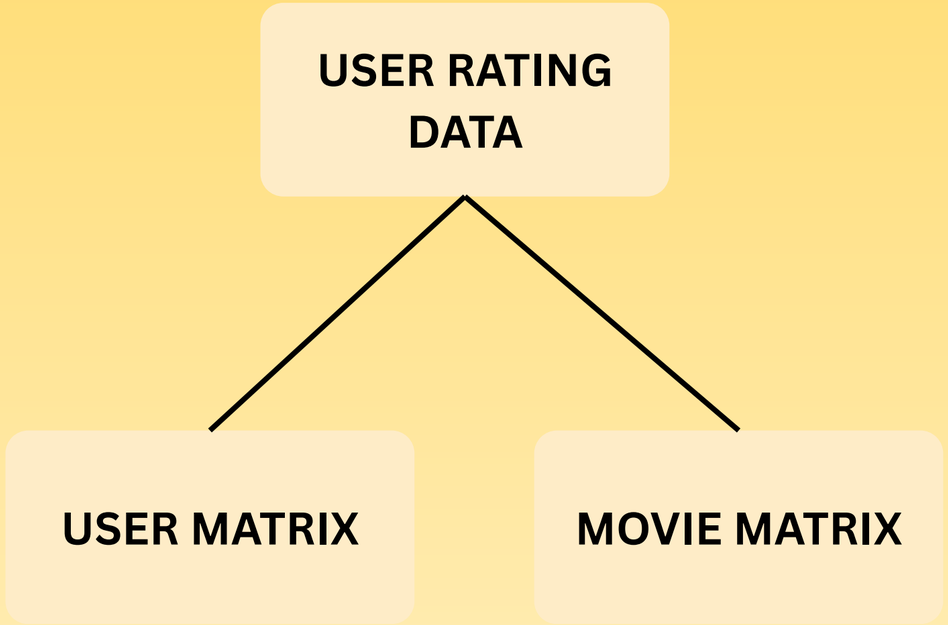
Goal: Reduce dimensionality, extract latent factors.

Popular Methods:

- SVD (Singular Value Decomposition) ⇝ General-purpose factorization
- PCA (Principal Component Analysis) ⇝ Variance maximization
- NMF (Non-negative Matrix Factorization) ⇝ Only additive parts (useful in images, text)

Applications:

- Recommender Systems (Netflix, Amazon, Spotify)
- Topic Modeling (text mining)
- Image Compression



Then, prediction is made by multiplying them:

$$R \approx U \times V^T$$

where R is the original user–movie rating matrix.

THANK YOU