Machine Learning: Data Preprocessing

Vikas Thammanna Gowda

01/23/2025

1 Feature Selection

Feature selection is the process of identifying and selecting the most relevant features (variables, attributes, or predictors) from a dataset for use in building a machine learning model. It aims to improve the model's performance, reduce overfitting, and make the model more interpretable.

1.1 Why Feature Selection is Important?

- **Improves Model Performance:**
  - Reduces noise and irrelevant data, leading to better predictions.
- **Reduces Overfitting:**
  - By using fewer features, the model focuses on the most important patterns, preventing it from fitting the noise.
- **Enhances Interpretability:**
  - Simplifies the model, making it easier to understand and explain.
- **Reduces Computational Cost:**
  - Fewer features mean reduced training time and resource requirements.

1.2 Types of Feature Selection Techniques

Feature selection can be broadly categorized into three main approaches:

1. Filter Methods

Features are selected based on their statistical relationship with the target variable, without involving a machine learning algorithm.
Techniques:

- Correlation: Measure the correlation between features and the target (e.g., Pearson correlation for numerical data).
- Chi-Square Test: Measures dependency between categorical features and the target.
- Mutual Information: Captures nonlinear dependencies between features and the target.

Advantages:

- Fast and easy to compute.
- Algorithm-independent.

Example:

- Removing features with low variance or those that have little correlation with the target variable.

2. Wrapper Methods

Use a machine learning algorithm to evaluate the importance of feature subsets by repeatedly training the model with different feature combinations.
Techniques:

- Forward Selection: Start with no features, add features one at a time, and evaluate performance.
- Backward Elimination: Start with all features, remove the least important one iteratively.
- Recursive Feature Elimination (RFE): Uses model coefficients or importance scores to remove features recursively.

Advantages:

- Takes into account interactions between features.

Example:

- Using RFE with a decision tree classifier to rank features by importance.

3. Embedded Methods

Feature selection is performed during the model training process as part of the algorithm.
Techniques:

- LASSO Regularization: Adds a penalty term to the model that forces some feature coefficients to become zero, effectively selecting important features.
- Tree-Based Feature Importance: Algorithms like Random Forest or XGBoost provide built-in feature importance measures.

Advantages:

- Efficient and less computationally expensive than wrapper methods.

Example:

- Using LASSO regression to shrink irrelevant feature coefficients to zero.

1.3 Examples of Feature Selection

- **Customer Segmentation:**
  - **Dataset:** Customer demographics and purchase behavior.
  - **Feature Selection:** Identify key features like income, age, and purchase frequency that affect customer segmentation, ignoring less impactful variables like phone number or address.
- **Medical Diagnosis:**
  - **Dataset:** Patient data including symptoms, test results, and medical history.
  - **Feature Selection:** Select the most relevant test results or symptoms to predict diseases, reducing noise from unrelated features.
- **Stock Market Prediction:**
  - **Dataset:** Financial indicators like stock prices, interest rates, and trading volume.
  - **Feature Selection:** Choose features with high predictive power, such as moving averages or volatility, while ignoring irrelevant features like company logos.

1.4 Challenges in Feature Selection

- **Curse of Dimensionality:** High-dimensional datasets can make it difficult to identify the most relevant features.
- **Feature Interactions:** Some features may not show importance individually but are significant when combined with others.
- **Overfitting:** Complex selection methods may overfit the training data, reducing generalization.

1.5 Best Practices for Feature Selection

- Understand the domain to identify potentially important features.
- Use correlation or statistical tests to filter irrelevant features.
- Apply wrapper or embedded methods for more refined selection.
- Validate feature selection using cross-validation to ensure generalization.
- Combine feature selection with dimensionality reduction (e.g., PCA) if necessary.

Feature selection is a crucial step in the machine learning pipeline, improving both the efficiency and effectiveness of models while enabling better insights into the problem at hand.

1.6 Exercise

1. What is the primary goal of feature selection in machine learning?
   (a) Reduce computational cost
   (b) Add more features to the dataset
   (c) Simplify feature scaling
   (d) Avoid data preprocessing

2. Which method is an example of a filter method?
   (a) Recursive Feature Elimination
   (b) LASSO Regularization
   (c) Pearson Correlation
   (d) Forward Selection

3. What is the primary advantage of wrapper methods?
   (a) Algorithm-independent
   (b) Fast computation
   (c) Accounts for feature interactions
   (d) Requires no labeled data

4. Which technique is an example of an embedded method?
   (a) Chi-Square Test
   (b) Random Forest Feature Importance
   (c) Backward Elimination
   (d) Forward Selection

5. What is a common challenge in feature selection?
   (a) Standardizing the dataset
   (b) Overfitting due to complex selection methods
   (c) Underfitting due to lack of data
   (d) Dealing with missing values

6. Which of the following is an example of feature selection in a medical dataset?
   (a) Choosing symptoms relevant to a disease
   (b) Applying PCA for dimensionality reduction
   (c) Training a deep learning model
   (d) Normalizing test results

2 Outlier Treatment

Outliers are data points that deviate significantly from the overall pattern of the dataset. Treating outliers appropriately is essential as they can distort statistical analyses and degrade the performance of machine learning models. Outlier treatment varies depending on the type of data distribution.

2.1 Outlier Treatment for Normal Distributions

In a normal distribution, the data follows a symmetric bell-shaped curve centered around the mean. Outliers are identified based on the distance from the mean using statistical thresholds.
Methods:

- **Z-Score Method:**
  - A Z-score measures how many standard deviations a data point is from the mean.
  - **Formula:** $Z = \frac{(X-\mu)}{\sigma}$
  - Where X is the data point, μ is the mean, and σ is the standard deviation.
  - **Threshold:** Commonly, Z-scores greater than 3 or less than -3 are considered outliers.
- **Interquartile Range (IQR) Method:**
  - Identify outliers using quartiles:
    - Q_1: 25th percentile.
    - Q_3: 75th percentile.
  - **Compute IQR:** $\text{IQR} = Q_3 - Q_1$
  - **Define outlier thresholds:**
    - Lower bound: $Q_1 - 1.5 \times \text{IQR}$
    - Upper bound: $Q_3 + 1.5 \times \text{IQR}$

Treatment Options:

- **Winsorization:** Replace outliers with the nearest threshold value.
- **Transformation:** Apply transformations like log or square root to reduce the impact of outliers.
- **Removal:** Remove data points identified as outliers if they are errors or irrelevant.

2.2 Outlier Treatment for Skewed Distributions

In skewed distributions, data is not symmetric, and outliers often lie in the longer tail of the distribution. Statistical methods for normal distributions may not be appropriate here.
Methods:

- **Modified Z-Score Method:**
  - Use the median and Median Absolute Deviation (MAD) instead of mean and standard deviation.
  - **Formula:** Modified Z-Score $= 0.6745 \times \frac{(X-\text{Median})}{\text{MAD}}$
  - $\text{MAD} = \text{Median}(|X - \text{Median}|)$
  - **Threshold:** Commonly, values with Modified Z-Scores greater than 3.5 are considered outliers.
- **Boxplot Method (IQR with Skewness Adjustment):**
  - Similar to the IQR method but with adjusted thresholds for skewed data.
  - The multiplier for IQR may be increased (e.g., $\pm 2 \times \text{IQR}$) to account for the skewness.

- **Transformation:**
  - Apply transformations like logarithmic or power transformations to normalize the distribution and reduce the effect of outliers.

Treatment Options:

- Cap or floor outliers at a specific percentile (e.g., top or bottom 1% of the data).
- Use robust methods like quantile regression or tree-based models that are less sensitive to outliers.

2.3 Outlier Treatment for Other Distributions

For distributions that are neither normal nor skewed, specialized methods may be needed based on the specific data characteristics.
Methods:

- **Density-Based Methods:**
  - Use density estimation techniques like DBSCAN (Density-Based Spatial Clustering of Applications with Noise) or Isolation Forest to identify data points that are isolated or in low-density regions.
- **Machine Learning-Based Detection:**
  - Use algorithms like Autoencoder Neural Networks or One-Class SVM to learn normal patterns in the data and flag deviations as outliers.
- **Custom Rules:**
  - For domain-specific datasets, create custom thresholds or business rules to define outliers.
  - **Example:** In stock prices, an outlier could be defined as a daily price change exceeding a certain percentage.

Treatment Options:

- Use robust statistical measures like median and MAD for outlier detection.
- Replace outliers using domain-specific knowledge or advanced imputation techniques (e.g., KNN imputation).

2.4 Comparison Table for Outlier Treatment

Distribution Type	Detection Method	Treatment Options
Normal Distribution	Z-Score, IQR	Winsorization, transformation, removal.
Skewed Distribution	Modified Z-Score, Adjusted IQR	Capping/flooring, robust transformations.
Other Distributions	Density-based methods, ML-based detection, custom rules	Domain-specific thresholds, robust measures, imputation.

2.5 Key Considerations

- **Context Matters:** Not all outliers are "bad." In some cases, they may contain important information (e.g., fraud detection).
- **Preserve Data Integrity:** Outlier treatment should not distort the dataset. Always validate the impact on the model's performance.
- **Automated vs Manual:** Automated methods (like Isolation Forest) are scalable, while manual methods (like IQR) are more interpretable.

2.6 Exercise

1. Which method is commonly used for outlier detection in a normal distribution?
   (a) Modified Z-Score
   (b) DBSCAN
   (c) Z-Score Method
   (d) One-Class SVM
2. What does the interquartile range (IQR) represent?
   (a) The mean of the dataset
   (b) The spread between Q1 and Q3
   (c) The standard deviation of the dataset
   (d) The sum of quartiles
3. Which outlier treatment is suitable for skewed distributions?
   (a) Winsorization
   (b) Modified Z-Score Method
   (c) Removal of outliers
   (d) Z-Score Method
4. What is a key consideration in treating outliers?
   (a) Outliers should always be removed
   (b) Outlier treatment should preserve data integrity
   (c) Outliers are always due to errors
   (d) Boxplots are ineffective for outlier detection
5. Which machine learning-based method can be used for outlier detection?
   (a) PCA
   (b) Isolation Forest
   (c) Logistic Regression
   (d) Gradient Boosting
6. What is an example of outliers providing meaningful information?
   (a) Fraud detection in financial data
   (b) Removing extreme values from weather data
   (c) Normalizing the dataset
   (d) Ignoring rare occurrences in training

3 Feature Scaling

Feature scaling is the process of transforming data to ensure that all features contribute equally to the model and are measured on the same scale. This is particularly important when features have different units or ranges, as machine learning algorithms often compute distances or rely on gradient calculations, which can be affected by unscaled features.

3.1 Why Use Feature Scaling?

- **Improves Model Performance:**
  - Algorithms like k-Nearest Neighbors (k-NN), Support Vector Machines (SVM), and Gradient Descent-based methods are sensitive to feature magnitudes.
  - Without scaling, features with larger ranges dominate distance or optimization calculations.
- **Speeds Up Convergence:**
  - For optimization algorithms (e.g., Gradient Descent), feature scaling ensures quicker convergence by avoiding steep gradients for certain features.
- **Ensures Equal Contribution:**
  - Scaling prevents bias toward features with larger values, ensuring all features contribute equally.
- **Maintains Consistency:**
  - Algorithms using weights (e.g., linear regression, logistic regression) benefit from scaled features for consistent parameter updates.

3.2 Techniques in Feature Scaling

1. Absolute Maximum Scaling

Definition: Scales each feature by dividing it by its absolute maximum value, ensuring all values fall within the range $[-1, 1]$.
Formula:

$$X' = \frac{X}{|X_{\max}|}$$

Where $X_{\max}$ is the maximum absolute value of feature X.
Use Case:

- Suitable when the dataset has both positive and negative values and absolute maximum scaling suffices for the application.

Example: For $X = [-10, -5, 0, 5, 10]$, $X' = [-1, -0.5, 0, 0.5, 1]$.
Advantages:

- Simple and intuitive.
- Handles Both Positive and Negative Values.

Disadvantage:

- Sensitive to outliers.

2. Min-Max Scaling

Definition: Scales features to a fixed range, typically $[0, 1]$, by shifting and rescaling the data.
Formula:

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

Where $X_{\min}$ and $X_{\max}$ are the minimum and maximum values of X.
Use Case:

- Widely used when the range of data values needs to be preserved but normalized.

Example: For $X = [10, 20, 30, 40, 50]$, $X' = [0, 0.25, 0.5, 0.75, 1]$.
Advantages:

- Simple and intuitive.
- Preserves relationships between values.

Disadvantages:

- Sensitive to outliers, as extreme values distort the scaling.

3. Normalization

Definition: Scales features to make the norm (magnitude) of each feature vector equal to 1.
Formula:

$$X' = \frac{X}{\|X\|}$$

Where $\|X\|$ is the norm (e.g., L_2-norm or L_1-norm).
Use Case:

- Common in text mining and image processing, where the magnitude of feature vectors varies significantly.

Example: For $X = [3, 4]$ (2D vector), $\|X\| = \sqrt{3^2 + 4^2} = 5$, $X' = [0.6, 0.8]$.
Advantages:

- Ensures all vectors have the same magnitude.
- Maintains relative distances between points.

Disadvantages:

- Not suitable for datasets with zero-centered features.

4. Standardization (Z-Score Scaling)

Definition: Scales features to have a mean of 0 and a standard deviation of 1, making them unit-less.
Formula:

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

Where μ is the mean and σ is the standard deviation.
Use Case:

- Preferred when features follow a normal distribution or algorithms assume Gaussian data.

Example: For $X = [10, 20, 30]$, $\mu = 20$, $\sigma = 10$, $X' = [-1, 0, 1]$.
Advantages:

- Handles outliers better than min-max scaling.
- Essential for PCA, k-means, and SVM.

Disadvantage:

- Less intuitive range compared to min-max scaling.

Key Considerations

- **Choose Technique Based on Algorithm:**
  - Algorithms like k-NN, SVM, and PCA benefit from standardization or normalization.
  - Neural networks often perform better with min-max scaling.
- **Outlier Impact:** Standardization is less sensitive to outliers compared to min-max scaling.
- **Domain Knowledge:** Choose scaling techniques considering the data's nature and the machine learning task.

Feature scaling is a crucial preprocessing step that ensures fair treatment of all features, improves convergence, and enhances model performance.

3.3 Exercise

1. What is the primary purpose of feature scaling?
   (a) Remove irrelevant features
   (b) Ensure features contribute equally
   (c) Simplify dimensionality reduction
   (d) Reduce dataset size

2. Which scaling technique ensures features have a mean of 0 and standard deviation of 1?
   (a) Min-Max Scaling
   (b) Standardization (Z-Score Scaling)
   (c) Normalization
   (d) Absolute Maximum Scaling

3. What is a limitation of min-max scaling?
   (a) It distorts relationships between features
   (b) It is computationally expensive
   (c) It is sensitive to outliers
   (d) It cannot handle negative values

4. Which scaling technique ensures all feature vectors have the same magnitude?
   (a) Standardization
   (b) Normalization
   (c) Min-Max Scaling
   (d) Logarithmic Transformation

5. Why is feature scaling important for algorithms like k-NN?
   (a) k-NN relies on distance calculations
   (b) k-NN is insensitive to feature ranges
   (c) k-NN does not require feature preprocessing
   (d) Feature scaling improves overfitting

6. What is the formula for Z-Score scaling?
   (a) $X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$
   (b) $X' = \frac{X}{|X_{\max}|}$
   (c) $X' = \frac{X - \mu}{\sigma}$
   (d) $X' = \frac{X}{||X||}$

4 Feature Extraction

Feature extraction is the process of transforming raw data into a set of meaningful, useful features that represent the data effectively. These extracted features are used as input to a machine learning model. Unlike feature selection, which involves choosing a subset of existing features, feature extraction involves creating new features based on existing data.

The goal is to reduce the dimensionality of the dataset while retaining its essential characteristics, thus improving the model's performance, interpretability, and computational efficiency.

4.1 Why Feature Extraction is important?

- **Simplifies Data:** Reduces the complexity of raw data, making it easier for models to process.
- **Improves Performance:** Helps machine learning algorithms focus on the most relevant information, leading to better predictions.
- **Handles Dimensionality:** Reduces the number of features while preserving important information, preventing the "curse of dimensionality."
- **Facilitates Interpretability:** Transforms raw data into human-interpretable formats, aiding in analysis.

4.2 Types of Feature Extraction Techniques

1. Numerical Data

For structured datasets, feature extraction focuses on aggregating, transforming, or engineering new features from existing ones.

Techniques:

- Statistical Features: Mean, median, standard deviation, skewness, kurtosis.
- Mathematical Transformations: Logarithms, squares, roots.
- Domain-Specific Features: E.g., total sales = unit price $\times$ quantity.

2. Text Data

Text data is inherently unstructured, so feature extraction focuses on converting it into numerical representations.

Techniques:

- **Bag of Words (BoW):** Represents text as a vector of word counts or frequencies.
- **TF-IDF (Term Frequency-Inverse Document Frequency):** Weighs terms based on their importance in the document relative to the entire dataset.
- **Word Embeddings:** Uses techniques like Word2Vec, GloVe, or BERT to represent words as dense vectors capturing semantic meaning.

3. Image Data

Images are represented as pixel values, and feature extraction involves deriving meaningful patterns or characteristics.

Techniques:

- **Edge Detection:** Sobel, Canny filters.
- **Color Histograms:** Represent distribution of colors in the image.
- **Convolutional Neural Networks (CNNs):** Automatically learn hierarchical features (edges, shapes, textures).

4. Audio Data

Audio data is often processed in the time or frequency domain for feature extraction.

Techniques:

- **Fourier Transform:** Converts time-domain signals to the frequency domain.
- **Mel-Frequency Cepstral Coefficients (MFCCs):** Widely used for speech and audio recognition.
- **Spectrograms:** Visual representation of frequency content over time.

4.3 Feature Extraction in Dimensionality Reduction

Feature extraction often involves dimensionality reduction techniques to summarize high-dimensional data effectively:

- **Principal Component Analysis (PCA):**
  - Projects data onto a lower-dimensional space by maximizing variance along new axes (principal components).
  - **Example:** Reducing hundreds of features in image data to a few key components.
- **t-Distributed Stochastic Neighbor Embedding (t-SNE):**
  - Maps high-dimensional data into a 2D or 3D space for visualization.
- **Autoencoders:**
  - Neural networks that compress data into a lower-dimensional representation and then reconstruct it.

4.4 Examples of Feature Extraction

- **E-Commerce Dataset:**
  - **Raw Data:** Purchase timestamps, prices, quantities.
  - **Extracted Features:** Total revenue, time between purchases, average purchase value.
- **Social Media Text:**
  - **Raw Data:** User posts.
  - **Extracted Features:** Word frequencies (BoW), sentiment scores, topic embeddings.
- **Medical Imaging:**
  - **Raw Data:** X-ray images.
  - **Extracted Features:** Tumor size, shape, texture, and location (using CNNs).
- **Speech Recognition:**
  - **Raw Data:** Audio signals.
  - **Extracted Features:** MFCCs, pitch, energy, or spectrograms.

4.5 Challenges in Feature Extraction

- **Domain Expertise:** Designing meaningful features often requires deep knowledge of the domain.
- **Overengineering:** Extracting too many features can lead to overfitting or computational inefficiency.
- **Automation:** Manually designing features is time-consuming; automated feature extraction using deep learning is gaining traction.

Feature extraction is a critical step in the machine learning pipeline that transforms raw data into meaningful, simplified, and actionable representations. By tailoring feature extraction techniques to the type of data (numerical, text, image, etc.), we can improve the performance, interpretability, and scalability of machine learning models.

4.6 Exercise

1. What is the main difference between feature selection and feature extraction?
   (a) Feature extraction reduces dimensionality by creating new features
   (b) Feature extraction eliminates redundant features
   (c) Feature selection is applied to image data only
   (d) Feature selection always involves deep learning
2. Which method is used to extract features from text data?
   (a) Fourier Transform
   (b) TF-IDF
   (c) PCA
   (d) LASSO Regularization
3. What is the purpose of PCA in feature extraction?
   (a) Normalize all features to a fixed range
   (b) Capture maximum variance in fewer dimensions
   (c) Remove outliers from the dataset
   (d) Scale features to have unit magnitude
4. What does "Bag of Words" (BoW) represent?
   (a) Relationships between sentences
   (b) Frequency of words in a document
   (c) Semantic meaning of words
   (d) Dimensionality reduction of text data
5. What is a key advantage of feature extraction?
   (a) Removes all irrelevant data points
   (b) Requires no domain knowledge
   (c) Simplifies raw data while preserving key patterns
   (d) Eliminates the need for preprocessing
6. Which feature extraction method is commonly used for audio data?
   (a) Edge Detection
   (b) Spectrograms
   (c) TF-IDF
   (d) Z-Score Scaling

5 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a dimensionality reduction technique used in machine learning and statistics to simplify datasets while retaining as much information as possible. PCA transforms the original data into a new set of uncorrelated variables called principal components that capture the maximum variance in the data.

5.1 Why Use PCA?

- **Reduce Dimensionality:** High-dimensional data can be computationally expensive and challenging to interpret. PCA reduces the number of features while preserving essential patterns.
- **Remove Multicollinearity:** PCA eliminates correlations between features by creating uncorrelated components.
- **Improve Visualization:** PCA projects high-dimensional data into 2D or 3D space for visualization.
- **Enhance Model Performance:** Reducing noise and redundant features can improve model accuracy and generalization.

5.2 How PCA Works

PCA follows these steps:

1. **Standardize the Data:**
   - Since PCA is affected by the scale of data, standardization (mean = 0, variance = 1) is performed to ensure all features contribute equally.
   - **Formula for standardization:** $Z = \frac{X - \mu}{\sigma}$
   - Where X is the feature, μ is the mean, and σ is the standard deviation.
2. **Compute the Covariance Matrix:**
   - The covariance matrix quantifies the relationships between all features. For n features, the covariance matrix is an $n \times n$ matrix.
3. **Calculate Eigenvalues and Eigenvectors:**
   - Eigenvalues represent the amount of variance captured by each principal component.
   - Eigenvectors define the directions (principal components) in which the data varies the most.
4. **Sort Eigenvalues in Descending Order:**
   - The principal components are ranked by the variance they explain (eigenvalues).
5. **Project Data onto Principal Components:**
   - Transform the original data onto the principal components with the highest eigenvalues to obtain the reduced dataset.

5.3 Mathematics Behind PCA

- **Covariance Matrix:** $\text{Cov}(X) = \frac{1}{n-1}(X^\top X)$
- **Eigenvalue Decomposition:** $\text{Cov}(X) \cdot v = \lambda \cdot v$
  - Where v is the eigenvector and λ is the eigenvalue.
- **Transformation:** Reduce the dimensionality by projecting data X onto the top k eigenvectors (principal components):

$$X_{\text{reduced}} = X \cdot W_k$$

- Where W_k contains the top k eigenvectors.

5.4 Explained Variance

The explained variance shows how much information (variance) is retained by each principal component.
Formula for explained variance ratio:

$$\text{Explained Variance Ratio} = \frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j}$$

Where λ_i is the eigenvalue of the ith component.

The cumulative explained variance helps decide the number of components to retain (e.g., keeping 95% of the variance).

5.5 Example of PCA

Dataset: Imagine a dataset with three features: Age, Salary, and Experience. These features are highly correlated (e.g., Salary increases with Age and Experience).
PCA Steps:

1. Standardize the data.
2. Compute the covariance matrix to assess relationships between features.
3. Perform eigenvalue decomposition to find principal components.
4. Retain the top 2 components that explain most of the variance.
5. Transform the data onto the new 2D space.

Visualization:

- **Before PCA:** The dataset is represented in a 3D space with Age, Salary, and Experience axes.
- **After PCA:** The dataset is projected onto a 2D space with two principal components (PC1 and PC2), simplifying analysis and visualization.

5.6 Applications

- **Image Compression:** Reduce the dimensionality of image data (pixel intensities) while retaining key features.
- **Genomics:** Simplify high-dimensional gene expression data for analysis.
- **Finance:** Identify latent factors driving asset prices in financial markets.
- **Text Analysis:** Reduce dimensionality of word embeddings for clustering or classification.

5.7 Advantages

- **Removes Redundancy:** Handles correlated features effectively.
- **Improves Efficiency:** Reduces computational costs by lowering dimensionality.
- **Enhances Interpretability:** Summarizes data with fewer features.

5.8 Disadvantages

- **Loss of Interpretability:** Principal components are linear combinations of original features, making them harder to interpret.
- **Assumes Linearity:** PCA captures linear relationships and may not perform well with non-linear patterns.
- **Sensitive to Scaling:** Requires standardized data for meaningful results.

PCA is a powerful dimensionality reduction tool that simplifies datasets, reduces noise, and highlights key patterns in data. It is widely used in machine learning and data analysis to improve performance and enable visualization in high-dimensional datasets.

5.9 Exercise

1. What is the primary goal of PCA?
   (a) To scale features to a uniform range
   (b) To reduce dimensionality while retaining key patterns
   (c) To eliminate missing data
   (d) To detect outliers

2. Which step is necessary before applying PCA?
   (a) Clustering data points
   (b) Scaling features to have mean 0 and standard deviation 1
   (c) Creating polynomial features
   (d) Removing redundant records

3. What is represented by the eigenvalues in PCA?
   (a) The direction of maximum variance
   (b) The amount of variance explained by each principal component
   (c) The mean of each feature
   (d) The total variance in the dataset

4. What does the explained variance ratio indicate?
   (a) The ratio of outliers in the data
   (b) The variance lost after scaling
   (c) The proportion of variance retained by a principal component
   (d) The number of principal components needed

5. Which of the following is a disadvantage of PCA?
   (a) Handles correlated features poorly
   (b) Assumes linear relationships in the data
   (c) Requires advanced machine learning models
   (d) Increases computational costs

6. What is a real-world application of PCA?
   (a) Normalizing salary data
   (b) Reducing dimensions of gene expression data
   (c) Removing outliers in financial datasets
   (d) Feature selection in medical diagnosis