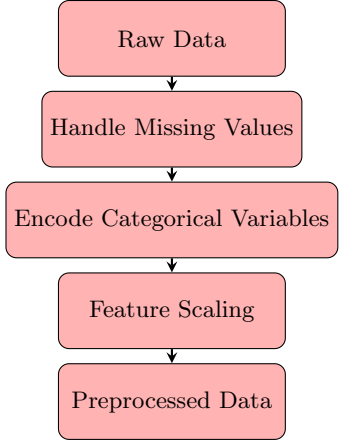

What is Data Preprocessing?

Why It Matters in Machine Learning!

Bindeshwar Singh Kushwaha

Real-World Data Challenges

- Missing or incomplete values

Real-World Data Challenges

- Missing or incomplete values
- Inconsistent formatting and typos

Real-World Data Challenges

- Missing or incomplete values
- Inconsistent formatting and typos
- Mixed data types (text, numeric, dates)

Real-World Data Challenges

- Missing or incomplete values
- Inconsistent formatting and typos
- Mixed data types (text, numeric, dates)
- Categorical variables needing encoding

Real-World Data Challenges

- Missing or incomplete values
- Inconsistent formatting and typos
- Mixed data types (text, numeric, dates)
- Categorical variables needing encoding
- Scale variations and outliers

What is Data Preprocessing?

- A set of techniques to clean and prepare raw data

What is Data Preprocessing?

- A set of techniques to clean and prepare raw data
- Essential for accurate and efficient machine learning

What is Data Preprocessing?

- A set of techniques to clean and prepare raw data
- Essential for accurate and efficient machine learning
- Includes cleaning, transformation, and structuring

Why is Preprocessing Important?

- Enhances model performance and accuracy

Why is Preprocessing Important?

- Enhances model performance and accuracy
- Prevents biases and errors during training

Why is Preprocessing Important?

- Enhances model performance and accuracy
- Prevents biases and errors during training
- Ensures model generalization

Why is Preprocessing Important?

- Enhances model performance and accuracy
- Prevents biases and errors during training
- Ensures model generalization
- Avoids data leakage and overfitting

Data Preprocessing Pipeline Overview

- Load and explore dataset

Data Preprocessing Pipeline Overview

- Load and explore dataset
- Handle missing values

Data Preprocessing Pipeline Overview

- Load and explore dataset
- Handle missing values
- Encode categorical features

Data Preprocessing Pipeline Overview

- Load and explore dataset
- Handle missing values
- Encode categorical features
- Scale and normalize numerical features

Data Preprocessing Pipeline Overview

- Load and explore dataset
- Handle missing values
- Encode categorical features
- Scale and normalize numerical features
- Detect and treat outliers

Data Preprocessing Pipeline Overview

- Load and explore dataset
- Handle missing values
- Encode categorical features
- Scale and normalize numerical features
- Detect and treat outliers
- Feature engineering and selection

Data Preprocessing Pipeline Overview

- Load and explore dataset
- Handle missing values
- Encode categorical features
- Scale and normalize numerical features
- Detect and treat outliers
- Feature engineering and selection
- Split data for training/testing

Dataset: Iris Dataset Overview

- 150 samples, 4 features, 1 target

Dataset: Iris Dataset Overview

- 150 samples, 4 features, 1 target
- Features: Sepal and petal measurements

Dataset: Iris Dataset Overview

- 150 samples, 4 features, 1 target
- Features: Sepal and petal measurements
- Target classes: Setosa, Versicolor, Virginica

Dataset: Iris Dataset Overview

- 150 samples, 4 features, 1 target
- Features: Sepal and petal measurements
- Target classes: Setosa, Versicolor, Virginica
- Used for classification and visualization

Dataset: Iris Dataset Overview

- 150 samples, 4 features, 1 target
- Features: Sepal and petal measurements
- Target classes: Setosa, Versicolor, Virginica
- Used for classification and visualization
- Load with: `from sklearn.datasets import load_iris`

Statistics Concepts Used in Preprocessing

- Mean, median, mode (for imputation)

Statistics Concepts Used in Preprocessing

- Mean, median, mode (for imputation)
- Standard deviation, variance (for scaling)

Statistics Concepts Used in Preprocessing

- Mean, median, mode (for imputation)
- Standard deviation, variance (for scaling)
- Z-score, IQR (for outlier detection)

Statistics Concepts Used in Preprocessing

- Mean, median, mode (for imputation)
- Standard deviation, variance (for scaling)
- Z-score, IQR (for outlier detection)
- Correlation (for feature selection)

Statistics Concepts Used in Preprocessing

- Mean, median, mode (for imputation)
- Standard deviation, variance (for scaling)
- Z-score, IQR (for outlier detection)
- Correlation (for feature selection)
- Chi-square test, ANOVA (for categorical relationships)

Python Libraries for Preprocessing

- `pandas` – Data handling and manipulation

Python Libraries for Preprocessing

- `pandas` – Data handling and manipulation
- `numpy` – Numerical computing

Python Libraries for Preprocessing

- `pandas` – Data handling and manipulation
- `numpy` – Numerical computing
- `scikit-learn` – Imputation, encoding, scaling, selection

Python Libraries for Preprocessing

- `pandas` – Data handling and manipulation
- `numpy` – Numerical computing
- `scikit-learn` – Imputation, encoding, scaling, selection
- `seaborn`, `matplotlib` – Data visualization

Python Libraries for Preprocessing

- `pandas` – Data handling and manipulation
- `numpy` – Numerical computing
- `scikit-learn` – Imputation, encoding, scaling, selection
- `seaborn`, `matplotlib` – Data visualization
- `scipy.stats` – Statistical analysis

Problems Without Preprocessing

- Models may crash with null values or strings

Problems Without Preprocessing

- Models may crash with null values or strings
- Features on different scales lead to bias

Problems Without Preprocessing

- Models may crash with null values or strings
- Features on different scales lead to bias
- Outliers distort parameter estimates

Problems Without Preprocessing

- Models may crash with null values or strings
- Features on different scales lead to bias
- Outliers distort parameter estimates
- Skewed distributions affect model assumptions

Preprocessing Flow: Iris Dataset

- Load with ‘load_iris()‘

Preprocessing Flow: Iris Dataset

- Load with ‘load_iris()‘
- Check data: ‘df.info()‘, ‘df.describe()‘

Preprocessing Flow: Iris Dataset

- Load with ‘load_iris()‘
- Check data: ‘df.info()‘, ‘df.describe()‘
- Handle missing values (if added for practice)

Preprocessing Flow: Iris Dataset

- Load with ‘load_iris()‘
- Check data: ‘df.info()‘, ‘df.describe()‘
- Handle missing values (if added for practice)
- Scale features using ‘StandardScaler‘

Preprocessing Flow: Iris Dataset

- Load with ‘load_iris()‘
- Check data: ‘df.info()‘, ‘df.describe()‘
- Handle missing values (if added for practice)
- Scale features using ‘StandardScaler‘
- Train-test split with ‘train_test_split‘

Key Takeaways

- Data preprocessing is foundational to ML success

Key Takeaways

- Data preprocessing is foundational to ML success
- Combines statistics and programming for data cleaning

Key Takeaways

- Data preprocessing is foundational to ML success
- Combines statistics and programming for data cleaning
- Tools: pandas, numpy, sklearn, matplotlib

Key Takeaways

- Data preprocessing is foundational to ML success
- Combines statistics and programming for data cleaning
- Tools: pandas, numpy, sklearn, matplotlib
- Concepts: Scaling, encoding, imputation, selection

Reach PostNetwork Academy

Website

www.postnetwork.co

Reach PostNetwork Academy

Website

www.postnetwork.co

YouTube Channel

www.youtube.com/@postnetworkacademy

Reach PostNetwork Academy

Website

www.postnetwork.co

YouTube Channel

www.youtube.com/@postnetworkacademy

Facebook Page

www.facebook.com/postnetworkacademy

Reach PostNetwork Academy

Website

www.postnetwork.co

YouTube Channel

www.youtube.com/@postnetworkacademy

Facebook Page

www.facebook.com/postnetworkacademy

LinkedIn Page

www.linkedin.com/company/postnetworkacademy

Reach PostNetwork Academy

Website

www.postnetwork.co

YouTube Channel

www.youtube.com/@postnetworkacademy

Facebook Page

www.facebook.com/postnetworkacademy

LinkedIn Page

www.linkedin.com/company/postnetworkacademy

GitHub Repositories

www.github.com/postnetworkacademy

Thank You!