Data Cleaning and Preprocessing for Scientific Computing

Dr Charalambos Chrysostomou
Associate Research Scientist
The Cyprus Institute

Brief overview of the importance of data quality

1. **Garbage In, Garbage Out**
2. **Bias and Fairness**
3. **Performance and Accuracy**
4. **Trust and Usability**
5. **Efficiency**

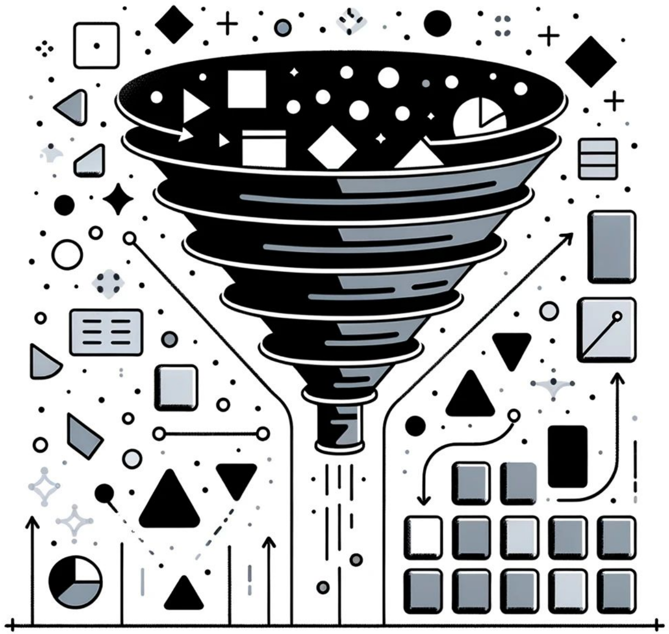
Introduction to data cleaning and preprocessing techniques

Data Cleaning

This process involves identifying and correcting errors in the dataset, such as dealing with missing or incorrect data. Data cleaning ensures that your dataset is accurate, consistent, and usable.

Data Preprocessing

Data preprocessing involves transforming raw data into an understandable format for algorithms. It prepares inputs for machine learning models.



Introduction to data cleaning and preprocessing techniques

Data Cleaning

1. **Missing Values**
2. **Outliers**
3. **Duplicates**
4. **Inconsistencies**

Handling Missing Values - Overview

Handling missing values is a critical step in data cleaning. Missing data can lead to biased or incorrect results, so it's important to handle them appropriately. The three primary strategies we'll discuss are:

- Dropping missing values
- Imputation techniques
- Interpolation

Dropping Missing Values

Dropping missing values is the simplest method, where we remove the rows or columns containing missing data.

Advantages:

- Easy to implement
- Doesn't introduce additional bias

Disadvantages:

- Can result in loss of valuable data
- Not ideal when missing data is not random

Use when:

- The dataset is large and a small proportion of data is missing
- Missing data is likely to be random

Imputation Techniques

Imputation involves filling missing values with estimated ones. The method can vary depending on the nature of the data and the type of variable.

- **Mean/Median/Mode Imputation:** Replace missing values with the mean (for continuous variables) or median/mode (for categorical variables).
- **Predictive Imputation:** Use statistical or machine learning algorithms to predict missing values based on other data.

Advantages:

- Prevents data loss
- Can handle non-random missingness if done correctly

Disadvantages:

- Can introduce bias if assumptions are incorrect
- More complex to implement

Use when:

- Missingness is not random
- Enough non-missing data is available to make reasonable estimates

Interpolation

Interpolation involves estimating missing values using other adjacent observed values. This method assumes a specific relationship between samples.

- **Linear Interpolation:** Assumes a straight-line relationship between points.
- **Polynomial/ Spline Interpolation:** Assumes a polynomial or flexible curve relationship between points.

Advantages:

- Can provide a good estimate for time-series data
- Can handle non-linear relationships

Disadvantages:

- Doesn't work well if assumption about relationship between samples is incorrect
- Not suitable for categorical data

Use when:

- Data is sequential (e.g., time-series)
- Missing values are likely to be a function of their nearest values

Handling Outliers - Overview

Outliers are data points that significantly differ from the rest of the dataset. They can affect the performance of machine learning models and lead to incorrect conclusions. In this section, we will discuss three methods for detecting and handling outliers:

- Z-score method
- IQR method
- Winsorization

Z-score Method

The Z-score method calculates the number of standard deviations a data point is from the mean. Data points with a Z-score above a specific threshold (e.g., 2 or 3) are considered outliers.

$$Z = (x - \mu) / \sigma$$

Where:

- **x** is the value you are interested in.
- **μ** (mu) is the mean of the dataset.
- **σ** (sigma) is the standard deviation of the dataset.

Z-score Method

Advantages:

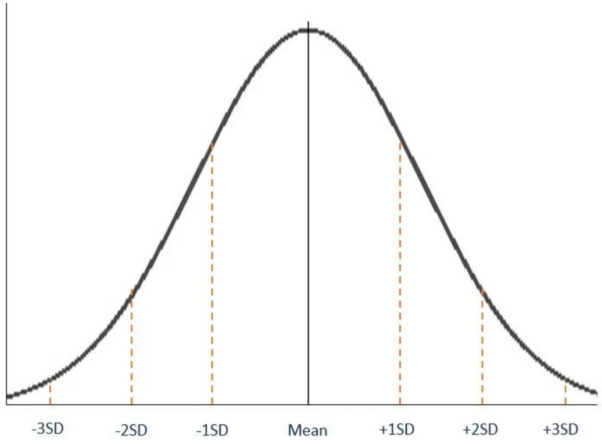
- Works well for normally distributed data
- Simple to calculate and implement

Disadvantages:

- Sensitive to extreme values
- Assumes data follows a normal distribution

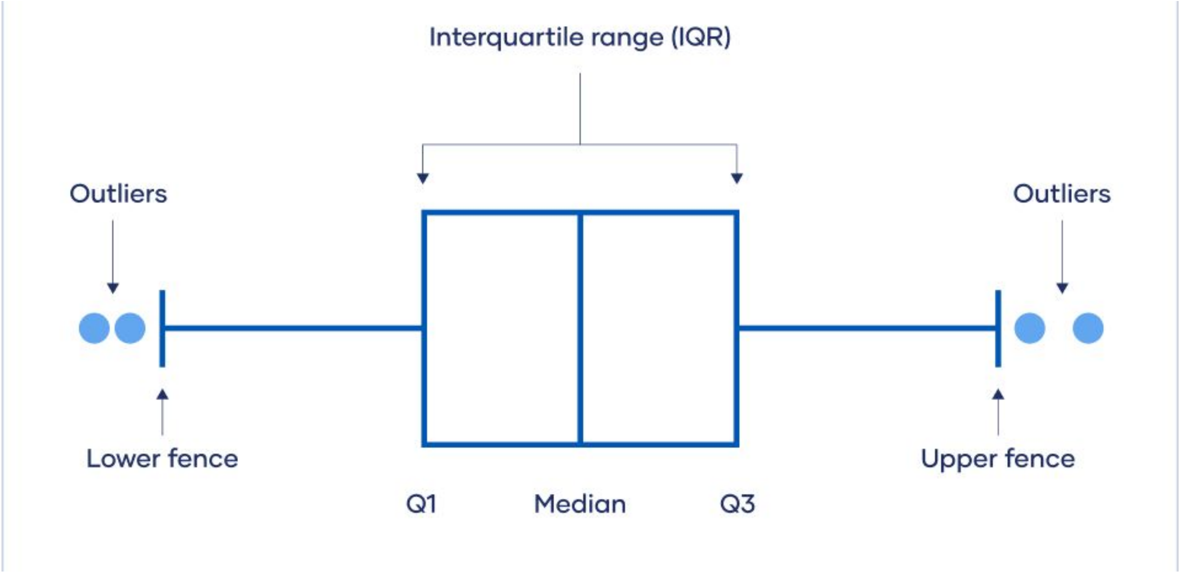
Use when:

- Data is approximately normally distributed
- The underlying distribution of data is symmetric



IQR Method

The IQR measures the statistical dispersion or spread of a dataset by indicating the range within which the central 50% of data lie.



IQR Method

Advantages:

- Robust to extreme values
- Doesn't assume normal distribution

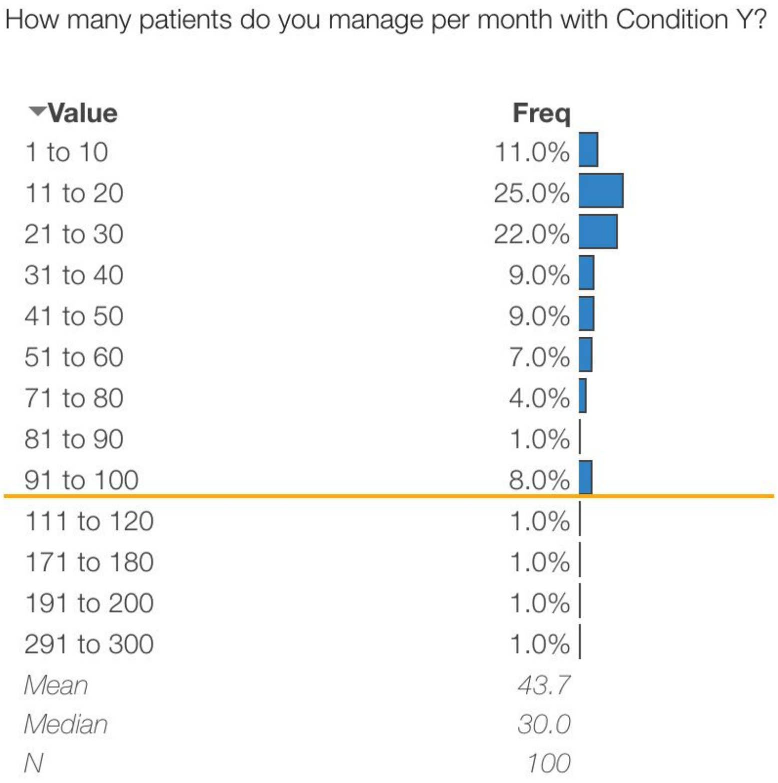
Disadvantages:

- Can be less effective for identifying outliers in small datasets
- Assumes data is unimodal

Use when:

- Data is not normally distributed
- Robustness to extreme values is important

Winsorization



How many patients do you manage per month with Condition Y?

Value	Freq
1 to 10	11.0%
11 to 20	25.0%
21 to 30	22.0%
31 to 40	9.0%
41 to 50	9.0%
51 to 60	7.0%
71 to 80	4.0%
81 to 90	1.0%
91 to 100	8.0%
111 to 120	1.0%
171 to 180	1.0%
191 to 200	1.0%
291 to 300	1.0%
Mean	*43.7*
Median	*30.0*
N	*100*

Winsorization involves replacing outliers with the nearest non-outlier value, typically using percentiles as a threshold (e.g., 1% and 99%).

Winsorization

Advantages:

- Reduces the impact of extreme values without completely removing them
- Doesn't require assumptions about the data distribution

Disadvantages:

- Can introduce bias if the threshold is set incorrectly
- Alters the original data, which might not be desirable

Use when:

- Retaining the overall structure of the data is important
- Reducing the impact of outliers without removing them is desired

Handling Duplicate Data in Datasets

- What Are Duplicated Data?
  - Identical entries in a dataset.
  - Can occur due to data entry errors, merging data from multiple sources, or incorrect data collection methods.
- Impact of Duplicated Data:
  - Leads to over-representation of certain information.
  - Biases the results of the analysis.
  - May affect model training and skew performance metrics.

Duplicate Data Removal

Advantages:

- Easy to implement
- Improves data accuracy

Disadvantages:

- Requires caution; duplicates could be valid data repetitions

Use when:

- There are certain data points repeated without any variations
- Duplication is not a natural aspect of your data collection

Inconsistent Data Entry Correction

Inconsistencies in data entries can occur due to various reasons like

- human error,
- different data entry conventions, or
- system glitches.

These inconsistencies can lead to errors in data analysis

Inconsistent Data Entry Correction

1. **Case Standardization**: Transform "apple", "BANANA", "Cherry" to "apple", "banana", "cherry".
2. **Formatting Dates**: Convert "12/31/2020", "2020-12-31", "31 Dec 2020" to "2020-12-31".
3. **Merging Categories:** Change all "NYC" entries to "New York City".
4. **Strip White Spaces:** Clean "John ", " Jane", " Mary " to "John", "Jane", "Mary".
5. **Master Dictionary:** Replace "grey" and "colour" with "gray" and "color" respectively.

Inconsistent Data Entry Correction

Advantages:

- Enhances data uniformity
- Improves accuracy of the analysis

Disadvantages:

- Can be time-consuming
- Requires thorough understanding of the data and domain

Use when:

- There are irregularities in data representation
- The dataset comes from different sources or conventions

Preprocessing techniques

Data Preprocessing

1. **Data Transformation:** Applying techniques to make data meet machine learning assumptions.
2. **Feature Scaling and Normalization:** Implementing methods to balance numerical input variables.
3. **Encoding Categorical Variables:** Using techniques to convert categorical variables into numerical.
4. **Feature Selection:** Selecting the most pertinent features using methods such as filter and wrapper methods.

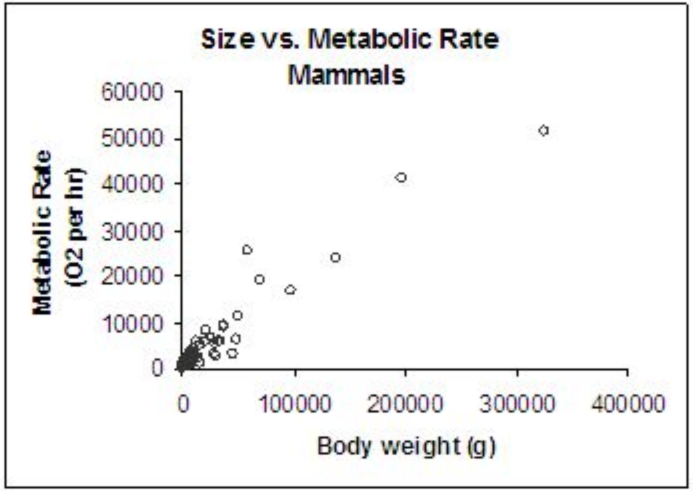
Data Transformation

Data transformation involves changing the scale or distribution of variables to better suit the assumptions of certain algorithms or improve model performance. We'll be discussing three popular methods:

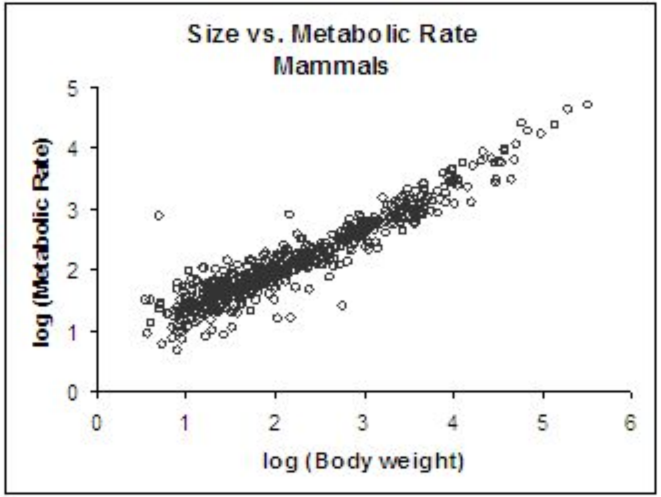
- Log Transformation,
- Box-Cox Transformation, and
- Yeo-Johnson Transformation.

Log Transformation

Log transformation is a commonly used technique to reduce skewness in a dataset, and it works by applying the logarithmic function to each data point.



Log → Trans-form



Log Transformation

Advantages:

- Helps manage skewed data
- Compresses the scale of the data, making it easier to handle

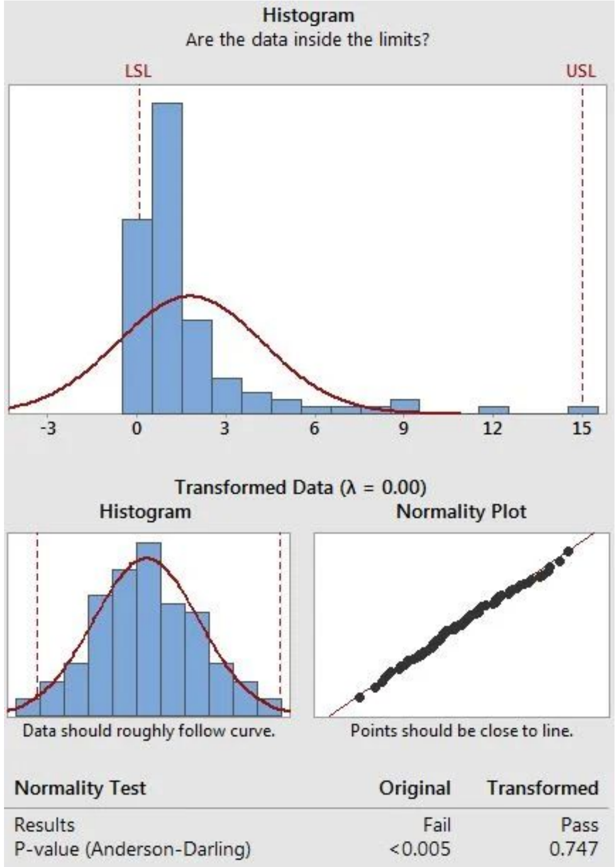
Disadvantages:

- Cannot be applied to zero or negative values
- May not normalize the distribution if data is heavily skewed

Use when:

- Data is right-skewed (tail to the right)
- All data points are positive

Box-Cox and Yeo-Johnson Transformations


Histogram
Are the data inside the limits?
LSL
USL
-3
0
3
6
9
12
15
Transformed Data (λ = 0.00)
Histogram
Normality Plot
Data should roughly follow curve.
Points should be close to line.

Normality Test	Original	Transformed
Results	Fail	Pass
P-value (Anderson-Darling)	<0.005	0.747

Box-Cox and Yeo-Johnson transformations are a flexible method for transforming data to approximate normality. It involves finding an optimal lambda parameter that best normalizes your data.

Box-Cox and Yeo-Johnson Transformations

Advantages:

- Dynamically determines the power transformation
- Works well for positive non-zero data

Disadvantages:

- Cannot handle zero or negative values
- Complexity of determining the right parameter

Use when:

- Data is skewed and all data points are positive
- You want a dynamic method to normalize data

Advantages:

- Can handle negative, zero, and positive values
- Works well for a variety of distributions

Disadvantages:

- Complexity of determining the right parameters
- May not be needed if simpler transformations are sufficient

Use when:

- Data contains negative values
- You need a more flexible method to normalize data

Data Preprocessing - Feature Scaling and Normalization

Feature scaling and normalization are techniques used to standardize the range of features in a dataset. This is crucial in machine learning, as models often perform better when numerical input variables are on a similar scale. We will discuss

- Min-Max Scaling,
- Standard Scaling (Z-score normalization), and
- Robust Scaling.

Feature Scaling and Normalization - Min-Max Scaling

Min-Max scaling, also known as normalization, rescales the features to a fixed range, usually 0 to 1.

X' = (X - X_min) / (X_max - X_min)

Where:

- **X** is the original value
- **X'** is the scaled value
- **X_min** is the smallest value in the feature column
- **X_max** is the largest value in the feature column

Pros:

- Simple and intuitive
- Preserves the original distribution

Cons:

- Sensitive to outliers
- Not suitable for data with large standard deviations

Use when:

- You want to preserve zero values in the dataset
- Your data doesn't contain extreme outliers

Feature Scaling and Normalization - Standard Scaling (Z-score normalization)

Standard scaling, or Z-score normalization, standardizes features by subtracting the mean and dividing by the standard deviation. The result has a mean of 0 and a standard deviation of 1.

$$Z = (X - \mu) / \sigma$$

Where:

- **X** is the original feature vector
- **μ** is the mean of the feature vector
- **σ** is the standard deviation of the feature vector
- **Z** is the standard score or Z-score.

Feature Scaling and Normalization - Standard Scaling (Z-score normalization)

Advantages:

- Centers the distribution around 0
- Handles outliers better than Min-Max Scaling

Disadvantages:

- Doesn't normalize the distribution to a specific range
- Can be influenced by outliers if extreme

Use when:

- Your data follows a Gaussian distribution
- Your algorithm assumes data is centered (e.g., Principal Component Analysis)

Feature Scaling and Normalization - Robust Scaling

Robust scaling uses the interquartile range, instead of the mean and standard deviation, making it robust to outliers.

$$X' = (X - Q1) / (Q3 - Q1)$$

Where:

- **X** is the original value
- **X'** is the scaled value
- **Q1** is the first quartile (25th percentile) of the feature column
- **Q3** is the third quartile (75th percentile) of the feature column

Feature Scaling and Normalization - Robust Scaling

Advantages:

- Reduces the effects of outliers
- Useful for data with heavy-tailed distributions

Disadvantages:

- Doesn't scale the data to a specific range
- May not perform well on non-Gaussian distributions

Use when:

- Your data contains many outliers
- You want to reduce the impact of outliers

Data Preprocessing - Encoding Categorical Variables

Categorical variables represent types of data which may be divided into groups. Since machine learning algorithms require numerical inputs, we need techniques to convert these categorical variables into a suitable numerical form. We'll cover three common methods:

- Label Encoding,
- One-Hot Encoding, and
- Target Encoding

Encoding Categorical Variables - Label Encoding

Label Encoding involves converting each value in a categorical column into a number. For example, if we have a feature called "Color" that has three categories: "Red", "Green", and "Blue". We could map these categories to numbers: "Red" = 1, "Green" = 2, "Blue" = 3.

Advantages:

- Simple and easy to implement
- Does not increase the dimensionality of the data

Disadvantages:

- Implies an ordered relationship between categories
- May lead to poor performance if the categorical variable is nominal

Use when:

- The categorical variable is ordinal (i.e., there's a logical order to the categories)

Encoding Categorical Variables - One-Hot Encoding

One-Hot Encoding involves creating new columns indicating the presence (or absence) of each unique value in the original data.

For example, if we have a feature called "Color" with three categories: "Red", "Green", and "Blue". One-Hot Encoding would create three new features: "Color_Red", "Color_Green", and "Color_Blue". If the original color was "Red", then "Color_Red" would be 1, and "Color_Green" and "Color_Blue" would both be 0.

Color	**Color_Red**	**Color_Green**	**Color_Blue**
Red	1	0	0
Green	0	1	0
Blue	0	0	1

Encoding Categorical Variables - One-Hot Encoding

Advantages:

- Creates binary vectors, eliminating any implied order
- Suitable for nominal categories (no inherent order)

Disadvantages:

- Can significantly increase data dimensionality
- Not suitable for categories with many unique values (high cardinality)

Use when:

- The categorical variable is nominal
- The number of unique categories is small

Encoding Categorical Variables - Target Encoding

Target Encoding, also known as mean encoding, is a method of encoding categorical variables based on the mean value of the target variable. This method can be particularly useful for high cardinality features where one-hot encoding might lead to high memory consumption.

Here's how Target Encoding works:

1. For each category in the feature, calculate the average value of the target variable.
2. Replace the category with the calculated average value.

For example, if we have a feature "City" and we want to predict the average house price (the target variable), we would replace each city name with the average house price in that city.

Encoding Categorical Variables - Target Encoding

Advantages:

- Can capture information within the category, improving model performance
- Prevents high dimensionality

Disadvantages:

- Risk of overfitting due to information leakage
- Requires careful validation strategies

Use when:

- Categorical feature is high cardinality
- There's a correlation between the category and the target variable"

Feature Selection

Feature selection is the process of reducing the number of input variables when developing a predictive model. It is crucial to remove irrelevant or partially relevant features that can negatively impact model performance. We'll cover three common methods:

- Filter Methods,
- Wrapper Methods

Filter Methods

Filter Methods for Feature Selection

- Statistical measures are applied to assign scores to each feature.
- Features are ranked based on their scores.
- High-scoring features are kept, while low-scoring features are removed.
- Methods are often univariate, considering features independently or in relation to the dependent variable.

Examples of Filter Methods

- **Pearson's Correlation:** Measures linear dependence between two continuous variables.
- **Chi-Squared Test:** Tests the relationship between categorical variables.
- **Mutual Information:** Measures dependency between features.
- **ANOVA F-Value:** Calculates linear dependency between a feature and the target variable.

Filter Methods

Advantages:

- Simple and easy to implement
- Fast and computationally efficient

Disadvantages:

- Does not consider interactions between features
- Based only on the intrinsic properties of the data

Use when:

- You have a large number of features
- You want to quickly filter out irrelevant features

Wrapper Methods

Wrapper Methods for Feature Selection

- Feature selection is treated as an optimization problem.
- Different combinations are evaluated using a predictive model.
- Model accuracy is used to assess the effectiveness of each combination.

Examples of Wrapper Methods

- Forward Selection: Adds features one by one, selecting the best-performing subset.
- Backward Elimination: Removes features one by one, retaining the best-performing subset.
- Recursive Feature Elimination: Removes variables iteratively based on model accuracy gains.

Wrapper Methods

Advantages:

- Searches for the best feature subset
- Considers interactions between features

Disadvantages:

- Computationally expensive
- Prone to overfitting

Use when:

- You have a moderate number of features
- Computational resources and time are not primary constraints

Case Study - Impact of Data Cleaning and Preprocessing Techniques on a Real-World Dataset

Titanic Dataset

https://t.ly/sLGB0

Exploring the Titanic: Data Cleaning & Preprocessing Impact

Objective: Understand the influence of data cleaning and preprocessing on a logistic regression model predicting Titanic passenger survival.

About the Dataset:

- Contains demographic and travel details of **1,309 Titanic passengers**.
- Mix of textual, Boolean, continuous, and categorical variables.
- Challenges include missing values, outliers, and text variables.

Key Attributes:

- **pclass**: Passenger class (1st, 2nd, 3rd).
- **survived**: Survival status (0 = No; 1 = Yes).
- **name**: Contains title and family names.
- **sex**: Gender (male/female).
- **age**: Age of the passenger.
- **sibsp**: Number of siblings/spouses aboard.
- **parch**: Number of parents/children aboard.
- **fare**: Passenger fare in British Pounds.
- **cabin**: Cabin location.
- **embarked**: Port of embarkation (C = Cherbourg; Q = Queenstown; S = Southampton).

Data Cleaning and Preprocessing

Initial Data Cleaning

Dropped Columns: 'Name', 'Age', 'Sex', 'Ticket', 'Cabin', 'Embarked'

Reason: Columns with missing or incomplete data sets were removed for the baseline model.

Initial Model Cross-Validation:

- Algorithm: Logistic Regression
- Evaluation Metrics:
  - Accuracy: 0.67 (+/- 0.08)
  - Precision: 0.64 (+/- 0.14)
  - Recall: 0.50 (+/- 0.09)
  - F1 Score: 0.56 (+/- 0.10)

Data Cleaning and Preprocessing

Advanced Data Cleaning & Preprocessing

- **Dropped Columns:** 'Name' and 'Ticket' (due to uniqueness and high variability).
- **Missing Values:** Filled 'Age' with median (to avoid outlier influence).
- **Categorical to Numerical:**
  - 'Cabin': Converted to binary (0 for absence, 1 for presence).
  - 'Sex': Label encoded ('male' as 0, 'female' as 1).
  - 'Embarked': One-hot encoded.
- **Standard Scaling:** Applied to 'Age' and 'Fare' for uniformity in feature scales.

Model Building and Evaluation

Improved Model Cross-Validation:

- Algorithm: Logistic Regression (with advanced cleaning & feature engineering)
- Evaluation Metrics:
  - Accuracy: 0.85 (+/- 0.06)
  - Precision: 0.84 (+/- 0.07)
  - Recall: 0.80 (+/- 0.11)
  - F1 Score: 0.82 (+/- 0.08)

Initial Model Cross-Validation:

- Algorithm: Logistic Regression (with minimal data cleaning)
- Evaluation Metrics:
  - Accuracy: 0.67 (+/- 0.08)
  - Precision: 0.64 (+/- 0.14)
  - Recall: 0.50 (+/- 0.09)
  - F1 Score: 0.56 (+/- 0.10)

Model Improvement Analysis & Key Takeaways

Improvement Percentages:

- Accuracy: +26.87%
- Precision: +31.25%
- Recall: +60%
- F1 Score: +46.43%

"Garbage in, garbage out." The importance of clean and well-preprocessed data cannot be overstated. It's the foundation upon which all data-driven decisions are made.

Q&A Session

Thank you for your attention!

Please feel free to ask your questions.