

Basis for Comparison

Definition

Qualitative Data

Qualitative data is information that can't be expressed as a number

Can data be counted?

NO

Data type

Words, objects, pictures, observations, and symbols

VS

Quantitative Data

Quantitative data is data that can be expressed as a number or can be quantified

YES

Number and statistics



A marketing analyst might use **quantitative data preprocessing** techniques to clean and transform data on customer demographics, purchase history, and product ratings.

This data could then be used to **create new features**, such as customer lifetime value or product affinity. The analyst could then use these features to build machine learning models to predict customer behavior or recommend products.



A social media analyst might use **qualitative data preprocessing** techniques to code and segment text data from social media posts.

This data could then be used to analyze customer sentiment, identify trends, or track the performance of marketing campaigns.



A medical researcher might use **quantitative and qualitative data preprocessing** techniques to clean and transform data from clinical trials.

This data could then be used to create new features, such as patient risk factors or treatment response. The researcher could then use these features to build machine learning models to predict patient outcomes or identify new treatment targets.

DATA PRE- PROCESSING TECHNIQUES



DATA PRE-PROCESSING TECHNIQUES

Data pre-processing techniques

Quantitative –
(Numerical)
data. **8 Types**

Qualitative –
(Categorical)
Data. **9 Types**

Data preprocessing is a crucial step in data analysis and machine learning, and it involves different techniques for quantitative (numerical) and qualitative (categorical) data.



DATA PRE-PROCESSING TECHNIQUES

▪ Qualitative (Categorical) Data

Label Encoding

- Convert categorical data into numerical format by assigning a unique integer to each category.

Label Encoding

Food Name	Categorical #	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50

One-Hot Encoding - Matrix

- Create binary (0/1) columns for each category to represent the presence or absence of a category.
- This is useful when there is no inherent order in the categories.

One Hot Encoding

Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

Dummy Variables

- Similar to one-hot encoding, but one category is dropped to avoid multicollinearity (the dummy variable trap).

$$x_{species} = [pine, oak, oak, pine, pine, pine, oak]$$

23-10-2023

$$x_{pine} = [1, 0, 0, 1, 1, 1, 0] \quad x_{oak} = [0, 1, 1, 0, 0, 0, 1]$$

DATA PRE-PROCESSING TECHNIQUES

▪ Qualitative (Categorical) Data

Frequency Encoding

- Encode categories with their frequency of occurrence.
- This can be useful when category frequency is related to the target variable.

Numerical value	Animal	Frequency encoding →	Numerical value	Animal_freq
1.5	cat		1.5	0.5
3.6	cat		3.6	0.5
42	dog		42	0.25
7.1	crocodile		7.1	0.25

Target Encoding (Mean Encoding)

- Replace categories with the mean of the target variable for each category.
- This can be powerful but must be handled carefully to avoid data leakage.

(EXAMPLE)

Ordinal Encoding

- Apply ordinal encoding when categories have an inherent order or rank.
Assign numerical values accordingly.

Original Encoding	Ordinal Encoding
Poor	1
Good	2
Very Good	3
Excellent	4

	Animal	Target	Encoded Animal
0	cat	1	0.40
1	hamster	0	0.50
2	cat	0	0.40
3	cat	1	0.40
4	dog	1	0.67
5	hamster	1	0.50
6	cat	0	0.40
7	dog	1	0.67
8	cat	0	0.40
9	dog	0	0.67

DATA PRE-PROCESSING TECHNIQUES

▪ Qualitative (Categorical) Data

- Data preprocessing methods for both quantitative and qualitative data should be chosen based on the specific characteristics of your data, the problem you are trying to solve, and the machine learning or statistical techniques you plan to use.
- Preprocessing is often an iterative process that requires experimentation and evaluation to achieve the best results.

Feature Hashing

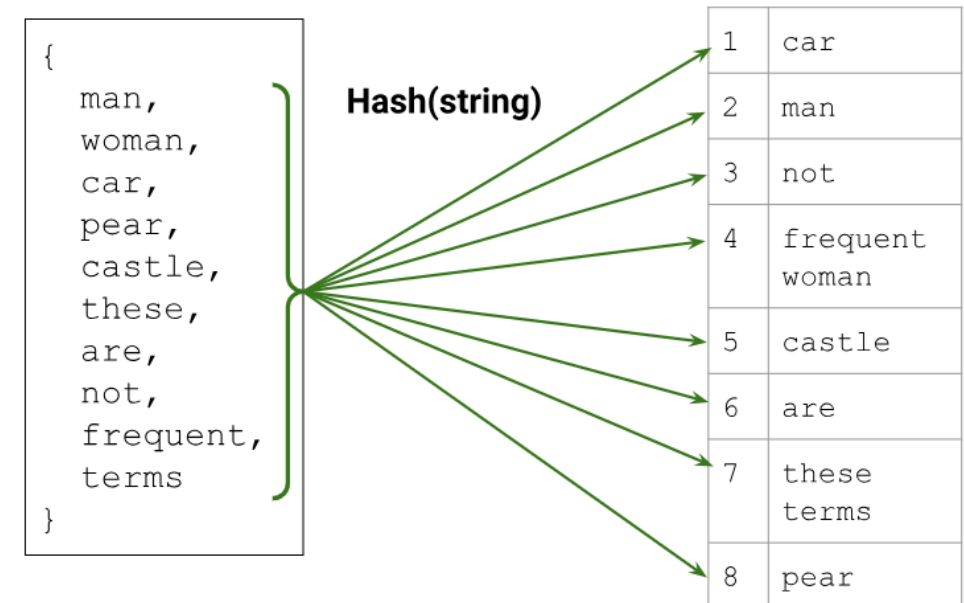
- Map categories to a fixed number of columns using a hash function to reduce dimensionality.

Binary Encoding

- Transform categories into binary code, which can be beneficial for machine learning algorithms.

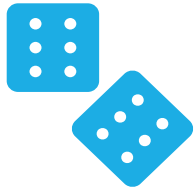
Combining Categories

- Sometimes, categories with low frequencies are merged into another category to reduce dimensionality.



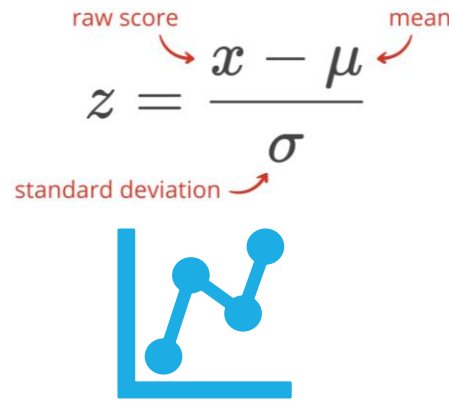
DATA PRE-PROCESSING TECHNIQUES

Quantitative (Numerical) Data



Handling Missing Values

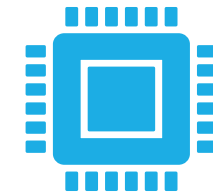
Identify and address missing values, which may involve imputation (replacing missing values with estimated values) using the mean, median, or a **statistical model**.



Outlier Detection and Handling

Detect and address outliers, which are extreme values that can adversely affect analyses or models.

Techniques include z-scores, the Interquartile Range (IQR), or domain-specific knowledge.

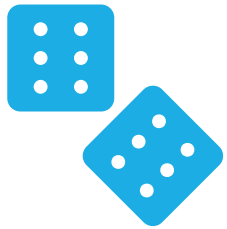


Feature Scaling/Normalization

Normalize numerical data to a common scale (e.g., $[0, 1]$ or $[-1, 1]$) to ensure that features have similar magnitudes, which is important for algorithms sensitive to feature scaling, such as K-means or Support Vector Machines (SVM).

DATA PRE-PROCESSING TECHNIQUES

Quantitative (Numerical) Data



Handling Missing Values

Identify and address missing values, which may involve imputation (replacing missing values with estimated values) using the mean, median, or a **statistical model**.

1. Polynomial Regression: If the relationship between the known values and the target variable is nonlinear, you can use polynomial regression to capture higher-order relationships.

2. Ridge Regression and Lasso Regression: These are variants of Linear Regression that include regularization to handle multicollinearity and prevent overfitting. Ridge regression adds L2 regularization, and Lasso regression adds L1 regularization.

3. Decision Trees and Random Forests: Decision trees can be used to predict missing values based on known values. Random Forest, an ensemble method of decision trees, can offer improved performance and robustness.

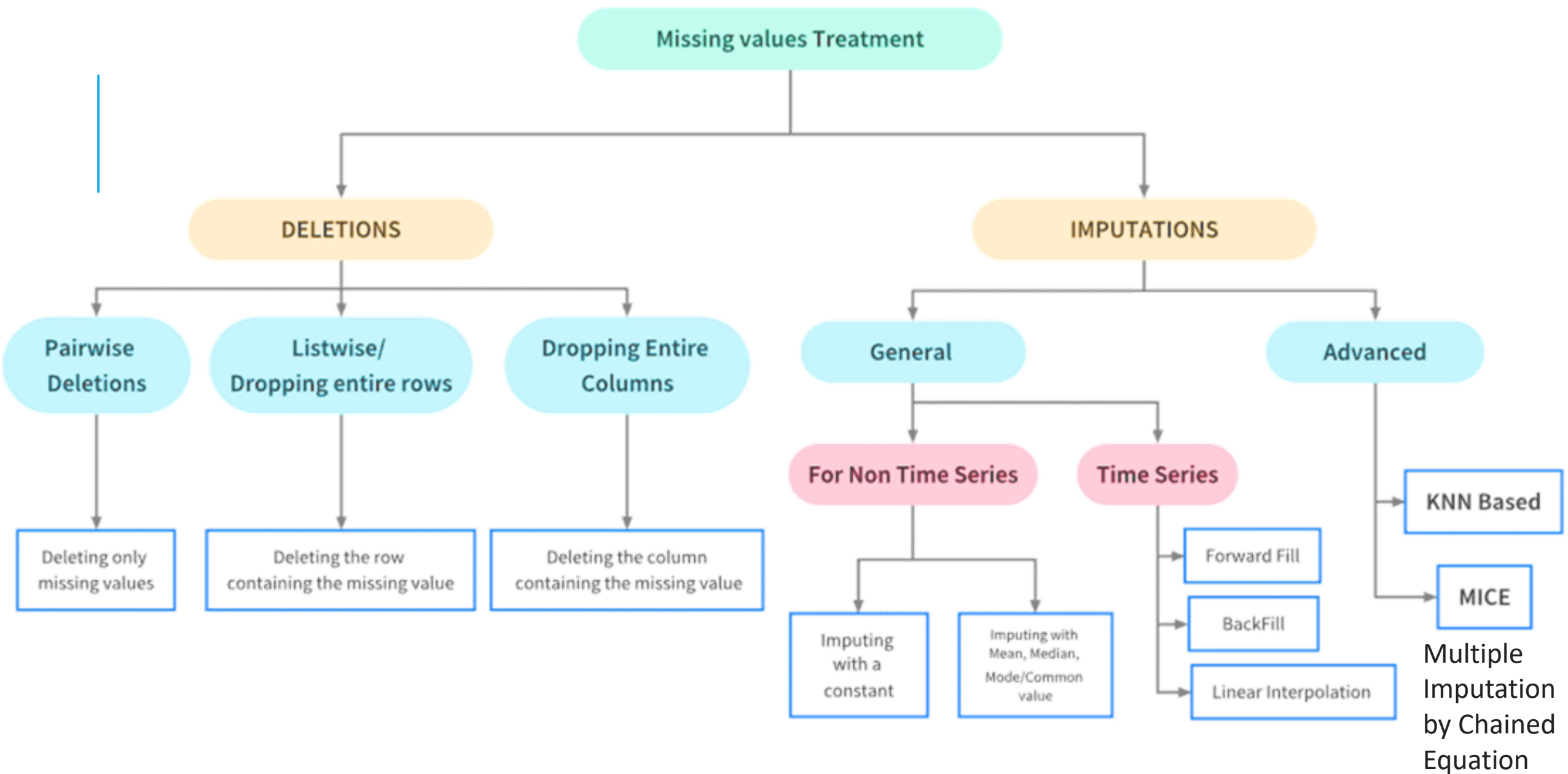
4. Support Vector Regression (SVR): SVR is a regression technique that uses support vector machines to find a hyperplane that best fits the data. It can handle both linear and non-linear relationships.

5. K-Nearest Neighbors (KNN): KNN is a non-parametric method that imputes missing values by averaging or interpolating values from the nearest neighbors.

6. Bayesian Regression: Bayesian regression models, such as Bayesian Linear Regression, incorporate prior beliefs about the data into the modeling process.

7. Gradient Boosting Regressors: Algorithms like Gradient Boosting, XGBoost, and LightGBM are powerful ensemble methods that can be used for imputation.

8. Linear regression




DATA PRE-PROCESSING TECHNIQUES

Quantitative (Numerical) Data

$$z = \frac{x - \mu}{\sigma}$$

raw score x mean μ standard deviation σ



Outlier Detection and Handling

Detect and address outliers, which are extreme values that can adversely affect analyses or models.

Techniques include z-scores, the Interquartile Range (IQR), or domain-specific knowledge.

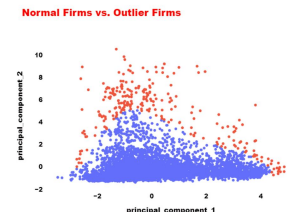
Outliers are data points that are significantly different from the majority of data points in a dataset.

There are different types of outliers, and they can be categorized into various groups based on their characteristics and causes. Here are some common types of outliers:

1. Univariate Outliers: These are outliers in a single variable or feature.

Univariate outliers can be classified into three categories:

- 1. Mild Outliers:** These are data points that are slightly different from the majority but not significantly so (Students age –varies from 18-22/32).
- 2. Extreme Outliers:** These are data points that are significantly different from the majority and are often the focus of outlier detection efforts (Only one student scored 100).
- 3. Tail Outliers:** These are data points that are located in the tails of a distribution and can be either mild or extreme outliers (last 10 rows of col).




DATA PRE-PROCESSING TECHNIQUES

Quantitative (Numerical) Data

Outliers are data points that are significantly different from the majority of data points in a dataset.

$$z = \frac{x - \mu}{\sigma}$$

raw score x mean μ standard deviation σ

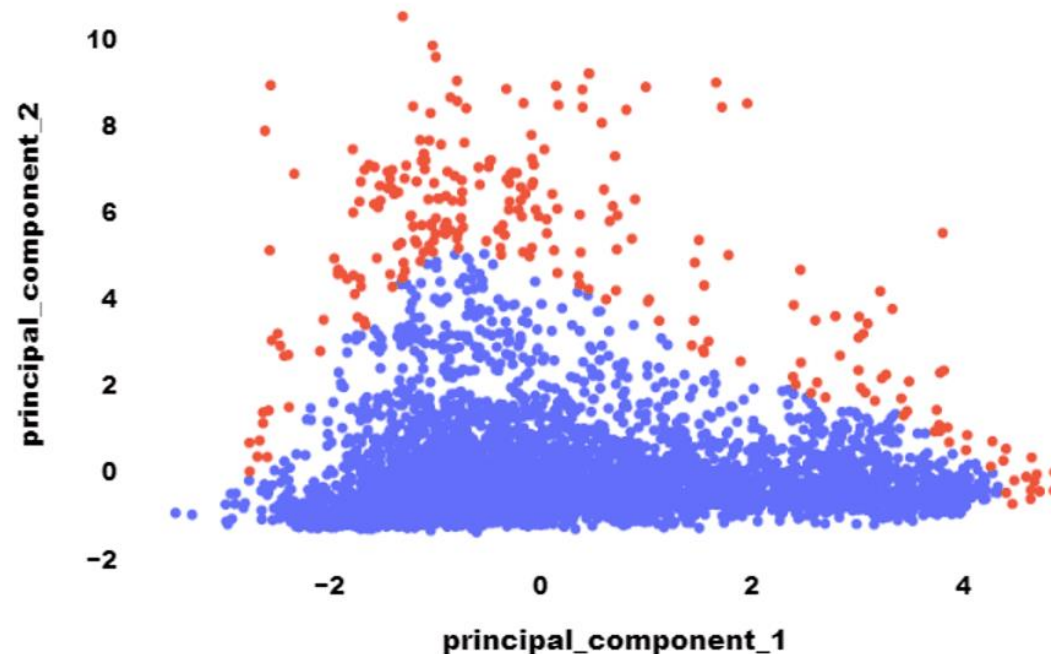


Outlier Detection and Handling

Detect and address outliers, which are extreme values that can adversely affect analyses or models.

Techniques include z-scores, the Interquartile Range (IQR), or domain-specific knowledge.

Normal Firms vs. Outlier Firms




DATA PRE-PROCESSING TECHNIQUES

Quantitative (Numerical) Data

$$z = \frac{x - \mu}{\sigma}$$

raw score x mean μ standard deviation σ



Outlier Detection and Handling

Detect and address outliers, which are extreme values that can adversely affect analyses or models.

Techniques include z-scores, the Interquartile Range (IQR), or domain-specific knowledge.

2. Multivariate Outliers: Multivariate outliers involve the consideration of multiple variables simultaneously. These outliers may not be detected when looking at each variable individually, but they are outliers when considering their joint distribution (**education level & salary – less education-high salary, high education-less salary**).

3. Global Outliers: Global outliers are data points that are outliers throughout the entire dataset. They are unusual in the context of the entire dataset

https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_by_population


4. Contextual Outliers: Contextual outliers are data points that are outliers within a specific context or subset of the data. They may not be outliers in the entire dataset but are considered as such within a particular subgroup (**daily sales and weekend sales**).

DATA PRE-PROCESSING TECHNIQUES

Quantitative (Numerical) Data

$$z = \frac{x - \mu}{\sigma}$$

raw score x mean μ standard deviation σ



Outlier Detection and Handling

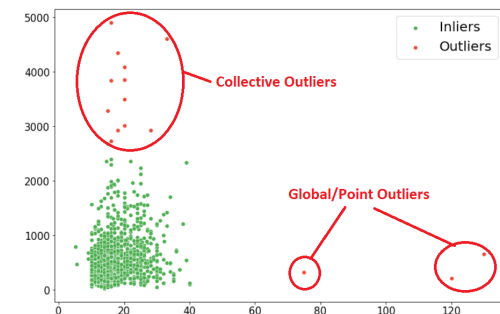
Detect and address outliers, which are extreme values that can adversely affect analyses or models.

Techniques include z-scores, the Interquartile Range (IQR), or domain-specific knowledge.

5. Point Outliers: These are individual data points that are far from the central tendency of the data distribution (daily sale vs offer sale).

6. Collective Outliers: Collective outliers refer to groups of data points that are collectively different from the rest of the data. They can be identified by patterns and relationships between data points.

7. Global Outliers vs. Local Outliers: Global outliers are unusual in the context of the entire dataset, while local outliers are unusual in a specific local region or subset of the data.




DATA PRE-PROCESSING TECHNIQUES

Quantitative (Numerical) Data

raw score x mean μ

$$z = \frac{x - \mu}{\sigma}$$

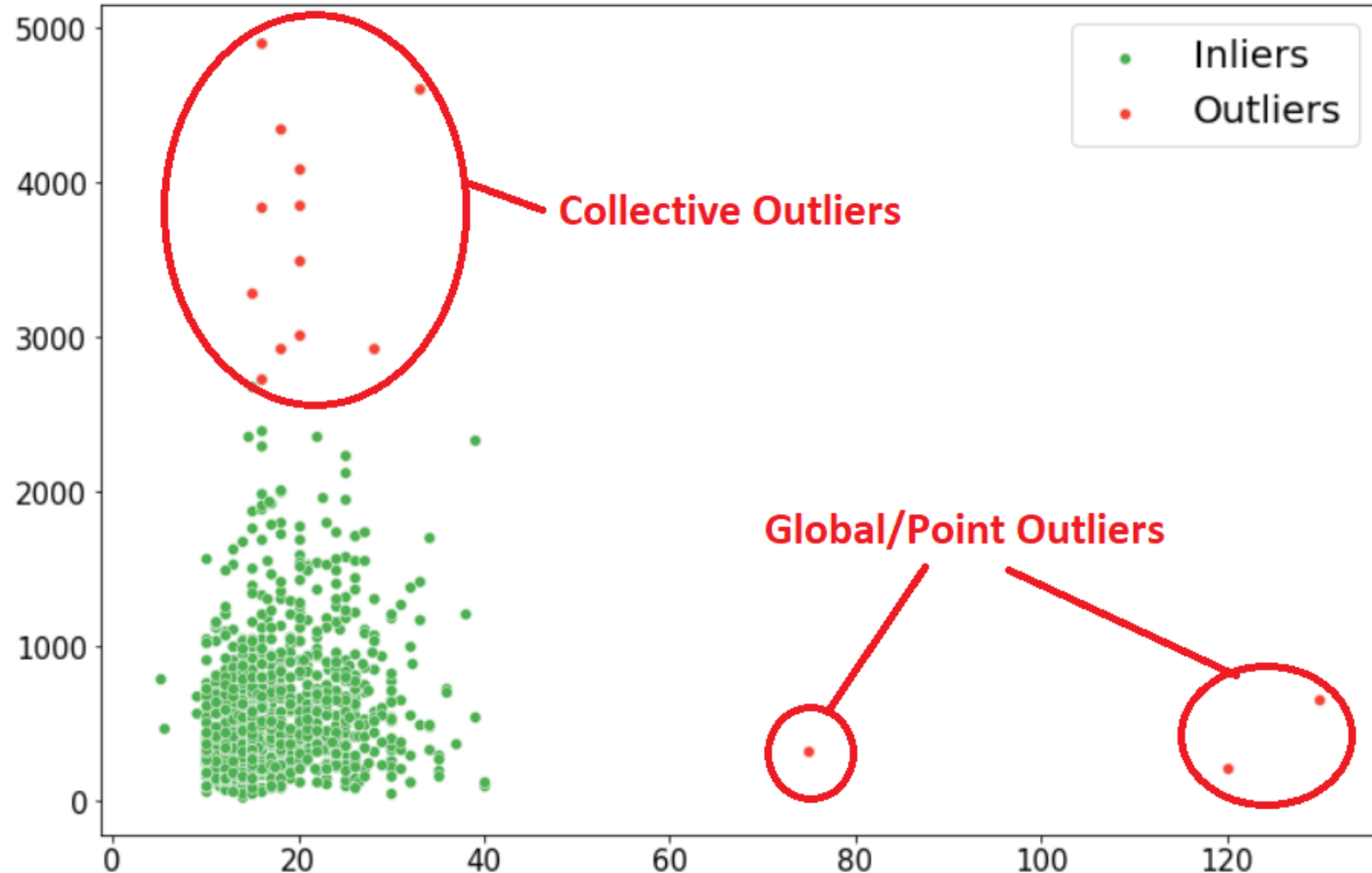
standard deviation σ



Outlier Detection and Handling

Detect and address outliers, which are extreme values that can adversely affect analyses or models.

Techniques include z-scores, the Interquartile Range (IQR), or domain-specific knowledge.




DATA PRE-PROCESSING TECHNIQUES

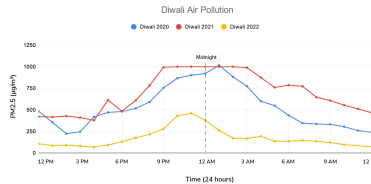
Quantitative (Numerical) Data

raw score x mean μ

$$z = \frac{x - \mu}{\sigma}$$

standard deviation σ





8. Additive Outliers vs. Multiplicative Outliers: Additive outliers are data points that deviate from the expected value by a fixed amount (Rs.120, Rs.130, Rs.140, **Rs.400**, Rs.125) while multiplicative outliers deviate by a fixed ratio (2,2,2,**8**,2).

9. Temporal Outliers: Temporal outliers are data points that are outliers in a time series data context. They may be caused by seasonality, trends, or external events <https://www.rupeerates.in/Gold>

10. Spatial Outliers: Spatial outliers occur in geospatial data and relate to data points that are outliers in a geographical context. They may be due to location-specific factors (pollution level-Diwali-Delhi).

Outlier Detection and Handling

Detect and address outliers, which are extreme values that can adversely affect analyses or models.

Techniques include z-scores, the Interquartile Range (IQR), or domain-specific knowledge.

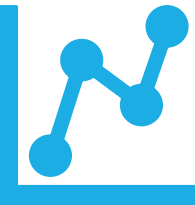
DATA PRE-PROCESSING TECHNIQUES

Quantitative (Numerical) Data

raw score x mean μ

$$z = \frac{x - \mu}{\sigma}$$

standard deviation σ

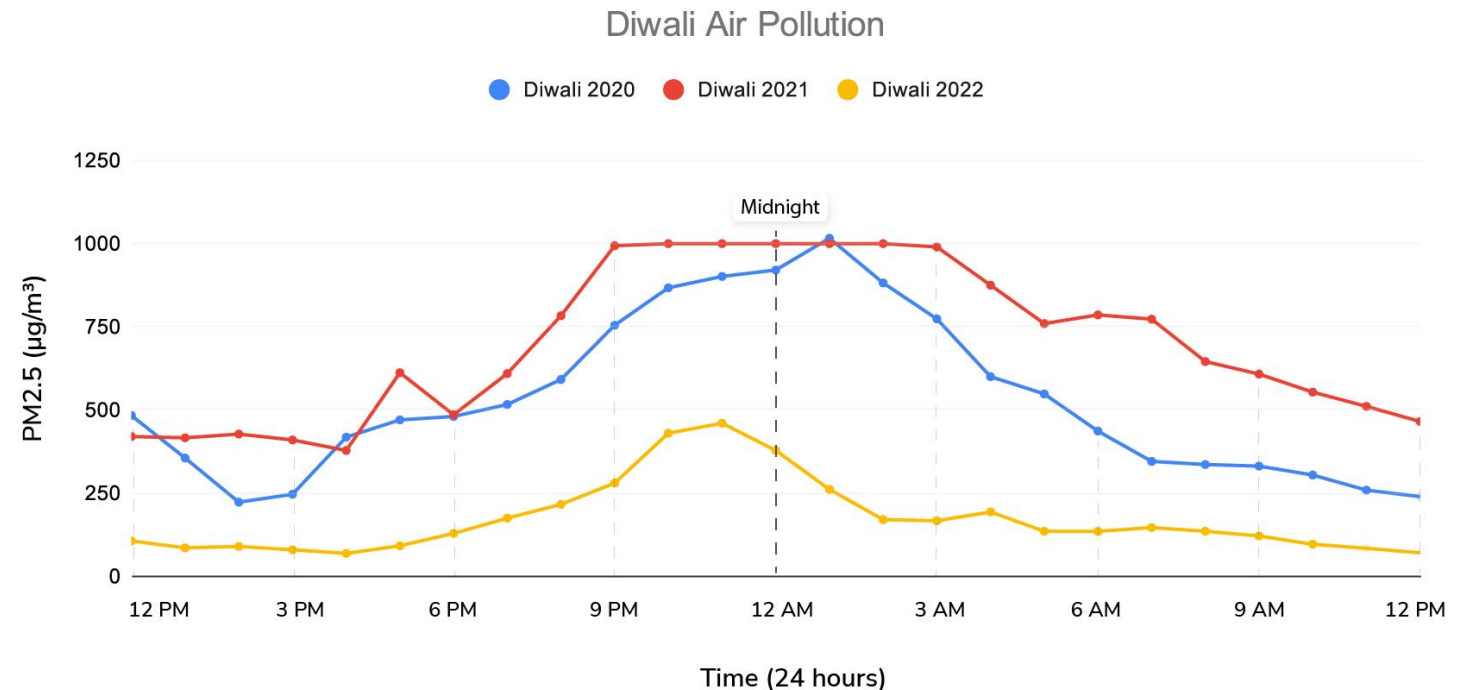


Outlier Detection and Handling

Detect and address outliers, which are extreme values that can adversely affect analyses or models.

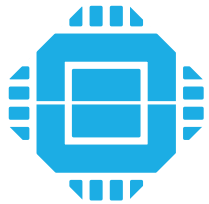
Techniques include z-scores, the Interquartile Range (IQR), or domain-specific knowledge.

11. Additive Outliers vs. Multiplicative Outliers: Additive outliers are data points that deviate from the expected value by a fixed amount (Rs.120, Rs.130, Rs.140, **Rs.400**, Rs.125) while multiplicative outliers deviate by a fixed ratio (2,2,2,**8**,2).



DATA PRE-PROCESSING TECHNIQUES

- Quantitative (Numerical) Data



Log Transformation or Power Transformation

Apply log or power transformations to make data more normally distributed, which can improve the performance of certain models and statistical tests.



Binning/Discretization

Group numerical data into bins or intervals to convert it into categorical data or make it more interpretable.

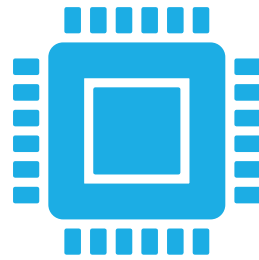


Feature Engineering

Create new features based on numerical data, such as interaction terms, polynomial features, or mathematical transformations to capture complex relationships.

DATA PRE-PROCESSING TECHNIQUES

- Quantitative (Numerical) Data



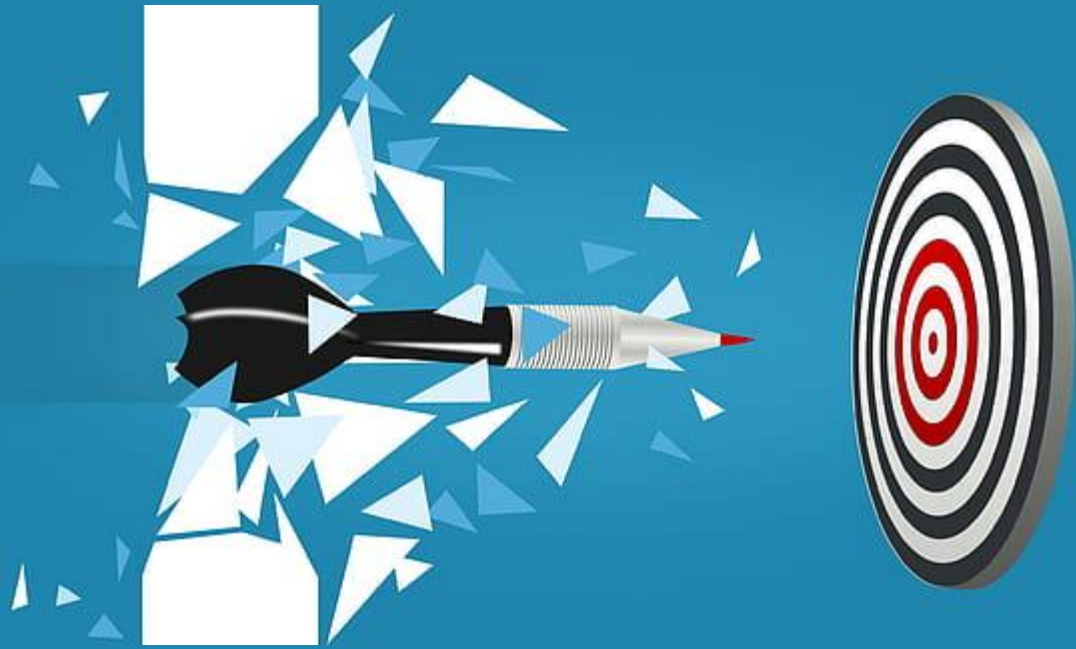
Principal Component Analysis (PCA)

Use dimensionality reduction techniques like PCA to reduce the number of numerical features while retaining important information.



Feature Selection

Choose the most relevant numerical features and discard irrelevant ones to improve model performance and reduce complexity.



Objectives of Preprocessing

Objectives of Preprocessing

Preprocessing is a critical step in machine learning that serves several important purposes to prepare data for analysis and modeling.

Data Quality Improvement

- Preprocessing helps clean and enhance the quality of the data.
- It involves identifying and addressing issues like missing values, outliers, and noise in the data, which can negatively impact the performance of machine learning models.

Data Transformation

- Preprocessing techniques transform data into a more suitable format for modeling.
- This includes converting data into numerical format, addressing scaling issues, and creating new features through feature engineering.

Feature Selection

- It helps identify and select the most relevant features or attributes to improve model performance.
- Feature selection reduces dimensionality and can lead to faster training times and less overfitting.

Objectives of Preprocessing

Normalization and Standardization

- Scaling and standardizing numerical features to a common scale can prevent certain machine learning algorithms from being biased towards features with larger magnitudes.
- This is crucial for distance-based algorithms like K-means clustering.

Handling Categorical Data

- Preprocessing techniques convert categorical data into a format that can be used by machine learning algorithms.
- This includes methods like one-hot encoding and label encoding.

Addressing Class Imbalance

- In classification problems, imbalanced datasets (where one class is much more prevalent than others) can lead to biased models.
- Preprocessing methods can help balance the dataset through techniques such as oversampling, undersampling, or synthetic data generation.

Objectives of Preprocessing

Text and Image Data Processing

- For unstructured data like text or images, preprocessing involves text tokenization, stop word removal, stemming or lemmatization, and image resizing or data augmentation to make it amenable for machine learning models.

Reduction of Overfitting

- Preprocessing can help reduce the likelihood of overfitting by reducing noise in the data, simplifying models through dimensionality reduction techniques (e.g., PCA), and applying feature selection.

Data Scaling for Optimization

- Machine learning algorithms often require that features are on a similar scale. Preprocessing ensures that this requirement is met, which can lead to faster convergence and better model performance.

Data Splitting

- Preprocessing is closely tied to data splitting.
- It involves partitioning the data into training and testing sets to assess model performance.
- Data splitting helps prevent overfitting and allows you to evaluate the model's generalization.

Objectives of Preprocessing

Feature Engineering

- Preprocessing may involve creating new features or modifying existing ones to capture important patterns or relationships in the data, which can lead to improved model performance.

Simplification and Interpretability

- Preprocessing can make the data more interpretable and understandable by simplifying it, which can be beneficial for certain types of analyses and for communicating results to non-technical stakeholders.

Overall, the purpose of data preprocessing in machine learning is to ensure that the data is in a suitable form for modeling, that it meets the assumptions and requirements of the chosen algorithm, and that it results in accurate and generalizable models.

Effective preprocessing can significantly impact the success of machine learning projects by reducing errors, improving model performance, and facilitating the extraction of meaningful insights from the data.

Missing Values Imputation



Missing value imputation

- Missing value imputation is a process in data preprocessing and analysis where missing or incomplete data points are estimated or filled in to make the dataset more complete.
- Missing data can arise for various reasons, including data collection errors, sensor malfunctions, survey non-responses, or incomplete records.
- It's important to handle missing values because they can adversely affect the quality of analyses and machine learning models.

	DATE	air_mv	air_mv_zero	air_mv_previous	air_mv_mean	air_expand
1	JAN49	112	112	112	112	112
2	FEB49	118	118	118	118	118
3	MAR49	132	132	132	132	132
4	APR49	129	129	129	129	129
5	MAY49		0	129	284.54385965	128.29783049
6	JUN49	135	135	135	135	135
7	JUL49		0	135	284.54385965	144.73734152
8	AUG49	148	148	148	148	148
9	SEP49	136	136	136	136	136
10	OCT49	119	119	119	119	119
11	NOV49		0	119	284.54385965	116.19900978
12	DEC49	118	118	118	118	118
13	JAN50	115	115	115	115	115
14	FEB50	126	126	126	126	126
15	MAR50	141	141	141	141	141

Missing value imputation

■ Mean, Median, or Mode Imputation

- This method involves replacing missing values with the mean (average), median (middle value), or mode (most frequent value) of the available data for that particular feature.
- This approach is simple but assumes that the missing data is missing at random and does not take into account potential relationships between features.

■ Regression Imputation

- Regression-based imputation uses other features as predictors to estimate the missing values.
- For numeric data, linear regression or other regression techniques can be used.
- For categorical data, logistic regression or similar models can be employed.

Missing value imputation

- **K-Nearest Neighbors (K-NN) Imputation**

- In K-NN imputation, missing values are estimated based on the values of the nearest neighbors in the dataset.
- The distance metric and the number of neighbors (K) are typically chosen by the user.

- **Multiple Imputation**

- Multiple imputation is a more advanced technique that generates multiple datasets, each with different imputed values, and then combines the results to create a single imputed dataset.
- This method accounts for the uncertainty associated with imputation.

- **Predictive Modeling (Machine Learning)**

- Machine learning models, such as decision trees, random forests, or deep learning, can be used to predict missing values based on the available data.
- This approach is powerful but may require more computational resources.

Missing value imputation

■ Hot-Deck Imputation

- Hot-deck imputation assigns the missing value the value of a similar observation within the dataset.
- The "similar" observation can be chosen based on specific criteria, such as similarity in other attributes or the nearest neighbor.

■ Statistical Imputation

- Statistical methods, like the Expectation-Maximization (EM) algorithm or Bayesian imputation, can be used to estimate missing values by modeling the underlying data distribution.

■ Domain-Specific Imputation

- In some cases, domain knowledge and business rules can guide the imputation process.
- For example, in a time series dataset, missing values could be imputed by carrying forward the last known value.

■ Zero or a Placeholder Value

- For some cases, it might be appropriate to replace missing values with zeros or another predefined placeholder value, especially when missing data represents a meaningful absence of information.

Missing value imputation

- The choice of imputation method should be guided by the nature of the data, the underlying assumptions, and the specific goals of the analysis or modeling task.
- It's important to carefully consider the implications of imputation on the results and to document the imputation process for transparency and reproducibility.
- Additionally, the imputation process should be validated to ensure it doesn't introduce bias or distort the original data distribution.



MISSING VALUE IMPUTATION – PURPOSE AND NEEDS

Missing value imputation – Purpose and Needs

■ **Preservation of Data Completeness**

- Imputing missing values helps maintain the completeness of the dataset, making it suitable for analysis or modeling.
- Missing data can result from various sources, including data collection errors or non-responses, and imputation allows for the utilization of available information.

■ **Improvement of Data Quality**

- By replacing missing values with estimated values, imputation can enhance the overall quality of the dataset.
- This helps prevent missing data from negatively impacting the quality of analyses and models.

■ **Enhanced Statistical Power**

- Missing data can lead to reduced statistical power, making it challenging to detect relationships or patterns in the data.
- Imputing missing values increases the effective sample size, which can improve the statistical power of analyses.

Missing value imputation – Purpose and Needs

■ Facilitation of Analysis

- Many statistical techniques and machine learning algorithms require complete datasets.
- Imputation enables the use of these methods and ensures that the analysis can be performed without errors or complications.

■ Improved Model Performance

- In machine learning, models may not perform well with missing data, and some algorithms may not accept datasets with missing values.
- Imputing missing data can lead to more accurate and robust model predictions.

■ Handling Time Series Data

- Time series data often contain missing values due to irregular observations or sensor failures.
- Imputation can enable the use of time series data for forecasting and analysis.

Missing value imputation – Purpose and Needs

- **Addressing Data Collection Bias**

- Missing data patterns can sometimes be related to specific attributes or the measurement process. Imputation can help mitigate potential biases introduced by these patterns.

- **Supporting Data Exploration**

- In data exploration and visualization, imputing missing values allows for a more comprehensive examination of the data, helping identify trends, patterns, and relationships.

- **Completing Datasets for Reporting**

- In data reporting or dashboard creation, missing value imputation ensures that the displayed data is complete and accurate, providing stakeholders with a more comprehensive view of the information.

- **Preservation of Record Integrity**

- In cases where individual records or samples with missing data need to be retained for downstream analyses or reporting, imputation allows these records to be included.

Missing value imputation – Purpose and Needs

- **Validation and Sensitivity Analysis**

- Imputation can be used to assess the sensitivity of analysis results to the imputation method and identify potential limitations in the conclusions drawn from the data.

- **Dealing with Real-World Data**

- Real-world datasets are often incomplete. Imputation allows for the use of practical, incomplete data without compromising the integrity of the analysis.

- It's important to choose appropriate imputation methods based on the specific context and characteristics of the data.

- Careful consideration should be given to the assumptions and potential impact of imputation on the results, and the process should be documented to ensure transparency and reproducibility in data analysis and modeling.

MISSING VALUE — REASONS



Missing value – Reasons

- **Non-Response**

- In surveys or questionnaires, some respondents may choose not to answer specific questions, resulting in missing data.
- This can be due to privacy concerns, sensitive topics, or the length and complexity of the survey.

- **Data Entry Errors**

- Mistakes made during data entry, such as typographical errors, can lead to missing or incorrect data.

- **Measurement or Sensor Errors**

- In scientific experiments or data collected from sensors, measurement errors, equipment malfunctions, or sensor failures can result in missing data points.

- **Data Not Collected**

- Sometimes, certain data may not be collected or recorded at all, either because it wasn't deemed necessary or due to resource constraints.

Missing value – Reasons

- **Data Not Available**

- In some cases, data may not be available for certain time periods or locations. For example, weather data may not be available for specific days or remote areas.

- **Data Deletion or Exclusion**

- Data might be intentionally deleted or excluded due to quality control procedures, privacy concerns, or because it was deemed irrelevant for the analysis.

- **Data Transformation**

- During data transformation or aggregation, some values may not be applicable or available, leading to missing data.

- **Incompatible Data Sources**

- When merging data from different sources, inconsistencies or discrepancies between the data can result in missing values.

Missing value – Reasons

- **Underreporting**

- In self-reporting systems, individuals may underreport specific information, such as their income or expenses, leading to missing or inaccurate data.

- **Data Skewness**

- Certain subgroups or categories within a dataset may have higher rates of missing data due to specific characteristics. This can introduce bias into analyses.

- **Non-Response Bias**

- Non-response may not be random and can lead to non-response bias, where the group of respondents who provide data differs systematically from the group that does not.

- **Lost or Damaged Records**

- Physical records or data files can be lost or damaged, leading to missing data for those records.

Missing value – Reasons

- **Survey or Question Design**

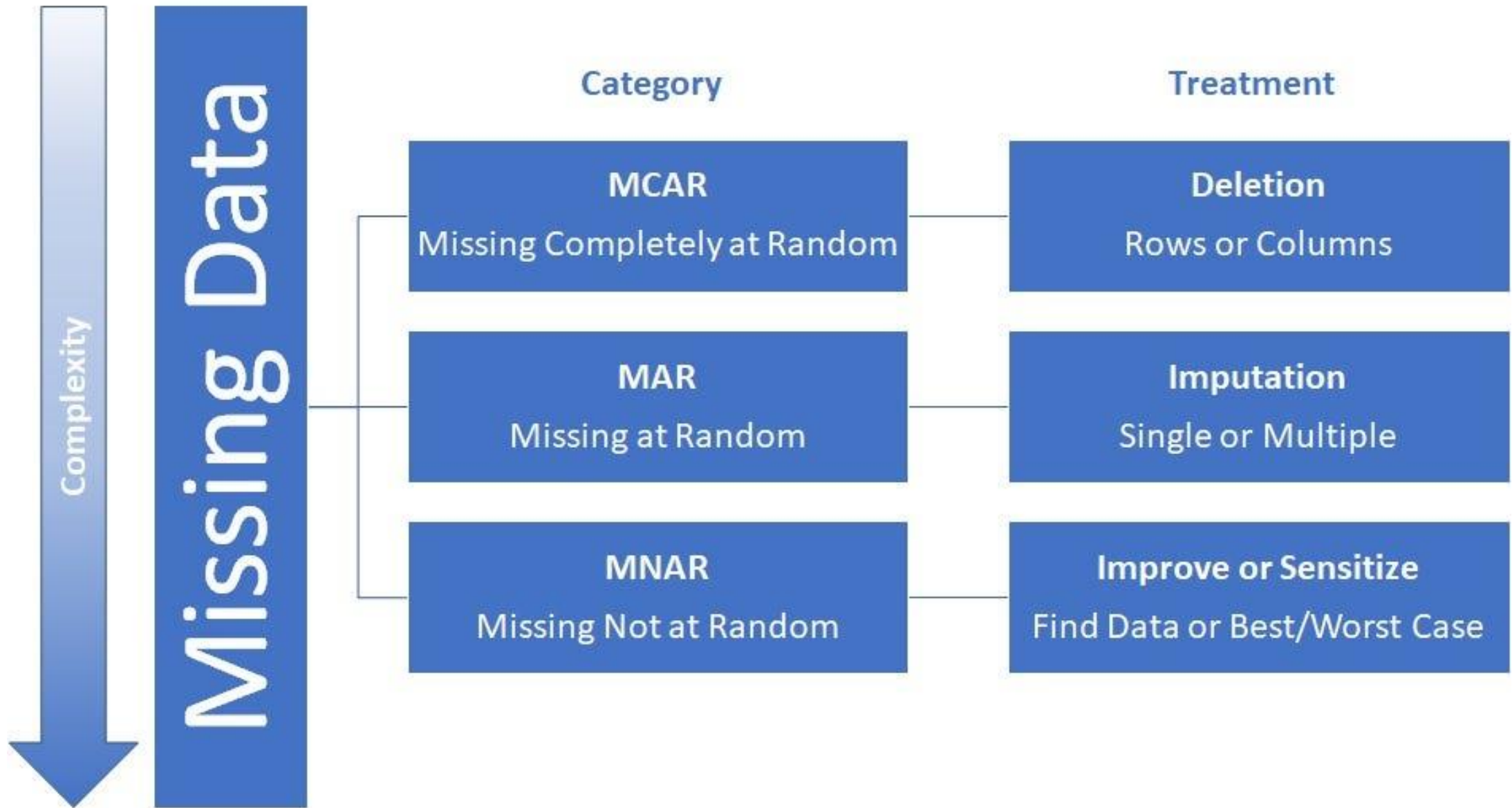
- The design of surveys or questionnaires can influence the likelihood of missing data. Poorly designed questions or response categories can result in non-response.

- **Privacy and Confidentiality Concerns**

- Individuals may choose not to provide certain data due to concerns about privacy, particularly in healthcare or sensitive research studies.

- Understanding the reasons for missing data is important because it can guide the selection of appropriate imputation methods and inform the handling of missing data in data analysis and modeling.

- Different imputation strategies may be more suitable depending on the nature of the missing data and its underlying causes.



Example: Getting FAIL in exam

Missing Value Imputation – MCAR

- Missing Completely at Random, is one of the patterns that missing data can exhibit.
- In an MCAR scenario, the missingness of data is unrelated to the observed or unobserved values in the dataset.
- In other words, for data to be MCAR, the probability of missingness is the same for all observations, regardless of the values of other variables or missing data.
- It occurs purely by chance, and there is no systematic reason for the data to be missing.
- MCAR is considered an ideal scenario for handling missing data because it implies that the missing data does not introduce bias into the analysis.
- In such cases, imputation methods that rely on the observed data alone (e.g., mean imputation, median imputation) are often appropriate and valid.
- Imputing missing values under MCAR assumes that there are no hidden patterns or reasons for data to be missing.

Missing Value Imputation – MCAR

- This means that any analysis or model built on imputed data should not produce biased results, as the missingness is unrelated to the values being imputed.
- It's crucial to acknowledge that MCAR is an idealized condition, and in many real-world situations, missing data is not completely random (MAR or MNAR), which may require more complex imputation methods and careful handling to avoid introducing bias into the analysis.

Missing Value Imputation – MAR

- MAR, or Missing at Random, is a pattern of missing data in which the probability of data being missing depends on the values of other observed variables but not on the missing data itself.
- In other words, for data to be considered MAR, the likelihood of missingness is related to the values of other variables in the dataset, even though the missing values themselves are not observed.
- The key distinction here is that the missingness is related to the observed data.
- When dealing with MAR, imputation methods often involve using observed data to predict or estimate the missing values.
- These methods typically assume that the observed data contains enough information to make reasonable predictions about the missing values.
- Some common imputation techniques for MAR data include regression imputation, k-nearest neighbors (K-NN) imputation, and multiple imputation.

Missing Value Imputation – MAR

- MAR is a less idealized scenario compared to Missing Completely at Random (MCAR), but it is still relatively straightforward to handle as long as the relationship between the missingness and the observed data is understood and accounted for.
- In contrast, the more challenging scenario is Missing Not at Random (MNAR), where the missingness is related to the missing data itself and not just the observed data.
- Handling MNAR data often requires more complex modeling and imputation techniques.

Missing Value Imputation – MNAR

- Missing Not at Random, is a pattern of missing data in which the probability of data being missing is related to the values of the missing data itself.
- In other words, for data to be considered MNAR, the missingness is related to the unobserved or missing values.
- This means that the mechanism causing data to be missing is not explained by the observed data but rather by the values that are not observed.
- Dealing with MNAR data is particularly challenging because imputation techniques that rely solely on the observed data may not be appropriate.
- Since the missing data itself is driving the pattern of missingness, imputing missing values based on observed data alone can introduce bias and distort the results of analyses or models.
- Addressing MNAR typically involves more complex modeling techniques that explicitly account for the mechanism causing the missingness.

Missing Value Imputation – MNAR

- This can include modeling the relationship between the missing data and the observed data, incorporating auxiliary information, and using specialized imputation methods designed for MNAR data.
- It's essential to understand the reasons behind MNAR data and choose appropriate methods to handle it, as it can significantly impact the validity of analyses and model results.



HANDLING MISSING VALUES — DROPPING ROWS AND COLUMNS

Handling missing values – Dropping rows and columns

- It is a straightforward approach, but it should be used judiciously as it can result in information loss.
- This strategy is suitable when the missing values are limited and do not significantly affect the overall dataset.
- **Dropping Rows (Instances)**
 - When you have a relatively small number of missing values, and the rows with missing values are limited, you can remove those specific rows from the dataset.
 - This approach is reasonable when the missing data does not represent a significant portion of the dataset and can be discarded without impacting the analysis.

`df.dropna(subset=['column_name'], inplace=True)`

- This method is simple but may not be suitable if the missing data is not Missing Completely at Random (MCAR).
- Dropping rows with missing data can lead to biased or incomplete results if the missingness is related to the variables you're analyzing.

Handling missing values – Dropping rows and columns

- **Dropping Columns (Features)**

- If a column has a high percentage of missing values or if the feature is not critical for your analysis, you may choose to drop the entire column.
- This approach reduces the dimensionality of your dataset and can be useful when the feature is not informative or when you have a more complete alternative.

`df.drop(['column_name'], axis=1, inplace=True)`

- Dropping columns may lead to a loss of potentially valuable information, so it should be considered carefully.
- It's advisable to perform feature importance analysis to assess the impact of dropping a column on your analysis.

Handling missing values – Dropping rows and columns

- **Dropping Rows with Missing Values Threshold**
- Another approach is to drop rows with missing values if the number of missing values in those rows exceeds a certain threshold.
- This approach allows you to retain more data while still reducing the impact of missing values.

threshold = 5 # Set the threshold for the number of missing values in a row

df.dropna(thresh=threshold, inplace=True)

- This approach balances data retention with data quality, and it is especially useful when you have a large dataset with sporadic missing values.
- **Combining Rows and Columns Dropping**
- In some cases, you may need to employ a combination of both row and column dropping to handle missing data. This depends on the specific nature and extent of the missing data.

Handling missing values – Dropping rows and columns

`df.dropna(subset=['column_name'], inplace=True)` # Drop rows with missing values

`df.drop(['column_to_drop'], axis=1, inplace=True)` # Drop a specific column

- Care should be taken to avoid excessive data loss and to assess the potential impacts on your analysis and modeling.
- It's important to note that while these strategies can be useful, they should be employed with caution.
- Consider the context of your data, the amount of missing data, and the goals of your analysis.
- Additionally, explore other imputation techniques, especially when the missing data may carry valuable information or when the missingness is related to the variables you are studying.



GROUP BASED IMPUTATION

Group Based Imputation

- Imputing missing values in numeric variables is a common task in data preprocessing and analysis.
- Group-based imputation strategies involve filling in missing values with statistics or values that are calculated within specific groups or categories of data.
- These strategies are particularly useful when dealing with data that has inherent grouping or categorization, such as data that can be grouped by a certain variable, like a class or category.
- **Mean/Median/Most Frequent Imputation by Group**
 - Calculate the mean, median, or most frequent value of the numeric variable within each group. You can use these group-specific statistics to impute missing values.
 - This approach helps preserve the central tendency of the data within each group.
- **Linear Regression Imputation**
 - Fit a linear regression model for the variable with missing values using other variables as predictors.
 - You can build separate regression models for each group or category.
 - Then, use the predicted values from the regression model to impute missing values within each group.

Group Based Imputation

- **K-Nearest Neighbors (KNN) Imputation by Group**
 - For each observation with a missing value, find the K-nearest neighbors within the same group using a distance metric like Euclidean distance.
 - Impute the missing value based on the average (or weighted average) of the values from the nearest neighbors within the same group.
- **Group-Specific Custom Functions**
 - Define custom imputation functions tailored to the specific characteristics of each group.
 - For example, you might impute missing values based on a function that considers the characteristics of a group, such as the group's historical performance.
- **Predictive Modeling by Group**
 - Train a predictive model (e.g., decision tree, random forest, or a more advanced machine learning model) for each group separately.
 - Use the trained models to predict missing values for each group.

Group Based Imputation

- **Interpolation/Extrapolation by Group**

- When dealing with time series data, you can use interpolation or extrapolation methods to fill in missing values within each time series group.
- These methods are based on the values observed before and after the missing data points.

- **Data-Driven Grouping**

- Use clustering or classification algorithms to group your data points based on similar characteristics.
- Then, impute missing values within each group separately using appropriate group-based strategies.

- **Hierarchical Imputation**

- In cases where you have hierarchical data (e.g., multiple levels of grouping), you can perform imputation at different levels of the hierarchy.
- Start by imputing at the highest level, then proceed to impute within subgroups.

Group Based Imputation

- When implementing these group-based imputation strategies, it's essential to consider the characteristics of your data and the specific context of your analysis.
- Group-based imputation can help you retain the structure and relationships present in your data while handling missing values effectively.
- Additionally, you should always evaluate the performance of your imputation strategy and its impact on your analysis to ensure it is appropriate for your dataset.

<https://cpcb.nic.in/AQI>

Central Pollution Control board

<https://t.ly/GAfH7>

AIR QUALITY INDEX of CITY DAY

<https://ai.invideo.io/>



<https://shorturl.at/ioHSU>

Download Bangaluru House Dataset