

DATA ANALYTICS

What is DATA?

- Data is a collection of facts or numbers that are organized to be analyzed, categorized, or used to help make decisions.
- It can be numerical or non-numerical, and can include discrete or continuous values.
- Quantitative Data: Numerical data that can be measured and quantified. It is further divided into:
 - ❖ Discrete Data: Countable data, like the number of students in a class.
 - ❖ Continuous Data: Data that can take any value within a range, like height or temperature.
- Qualitative Data: Non-numerical data that describes qualities or characteristics. It includes:
 - ❖ Nominal Data: Categorical data without a specific order, like colors or names.
 - ❖ Ordinal Data: Categorical data with a specific order, like rankings or grades.

- **Data Analytics:** The process of examining raw data to uncover patterns, draw conclusions, and make informed decisions. It involves techniques such as descriptive, diagnostic, predictive, and prescriptive analytics. The goal of data analytics is to transform data into actionable insights that can guide decision-making in various fields such as business, healthcare, and finance.
- **Data Mining:** In this it focused on discovering hidden patterns and relationships in large datasets using methods from statistics, machine learning, and database systems. It involves tasks such as clustering, classification, regression, and association rule learning. The primary aim of data mining is to extract previously unknown and potentially useful information from data.
- **Data Science:** It is an interdisciplinary field that combines computer science, statistics, and domain expertise to extract knowledge and insights from structured and unstructured data. It encompasses a broader scope than data analytics and data mining, including data collection, cleaning, analysis, and visualization. Data science also involves the development of algorithms and models to solve complex problems and predict future trends.

- **Datasets:** A dataset is a collection of data that is organized in a structured format, often presented in a tabular form consisting of rows and columns. Each row represents a record, and each column represents a variable or attribute of the data. Datasets are fundamental to data analysis, machine learning, and data science as they provide the raw information needed for analysis and model training. E.g. Iris dataset, MINST dataset.
- **Features:** They are individual measurable properties or characteristics of the phenomena being observed. Features are also referred to as attributes, variables, or predictors. They are the input variables used to build models that can predict outcomes or uncover patterns in the data. It can be in numerical and categorical form.
- **Data Scales:** Data scales refer to the levels of measurement that describe the nature of information within the values assigned to variables. They are essential in determining the appropriate statistical analysis and visualizations that can be applied to the data. There are four primary scales of measurement: nominal, ordinal, interval, and ratio.

Numerical and Categorical data:

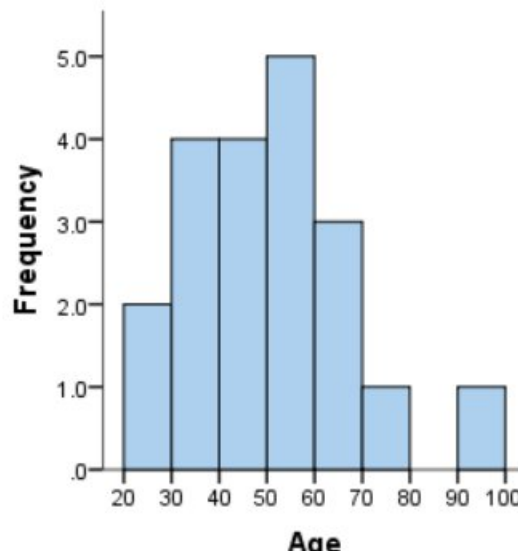
- **Numerical data** refers to data that represents measurable quantities and can be expressed as numbers. It is often used for quantitative analysis and can be further classified into two types: discrete and continuous. Discrete numerical data consists of countable values, such as the number of students in a class or the number of cars in a parking lot. Continuous numerical data, on the other hand, can take any value within a given range and includes measurements such as height, weight, and temperature.
- **Categorical data** represents characteristics or attributes that can be grouped into categories. Unlike numerical data, categorical data does not imply a quantity but rather a quality or classification. This type of data can be divided into nominal and ordinal categories. Nominal data consists of categories without any intrinsic order, such as gender, nationality, or hair color. Ordinal data, while still categorical, includes a meaningful order or ranking, such as levels of education (high school, bachelor's, master's) or customer satisfaction ratings (satisfied, neutral, dissatisfied).

Cross-sectional and Time series data:

- **Cross-sectional data** refers to data collected at a single point in time, capturing a snapshot of multiple subjects or entities. This type of data is used to analyze variations across different subjects, such as individuals, companies, countries, or any other units of analysis, at a particular moment. For example, a survey collecting data on the income levels, educational background, and employment status of different individuals at one point in time is cross-sectional.
- **Time series data**, in contrast, consists of data points collected or recorded at successive points in time, often at regular intervals. This type of data tracks the evolution of specific variables over time, enabling analysis of trends, cycles, and patterns. Examples include daily stock prices, monthly unemployment rates, or yearly GDP growth rates. Time series data is crucial for forecasting and understanding temporal dynamics.

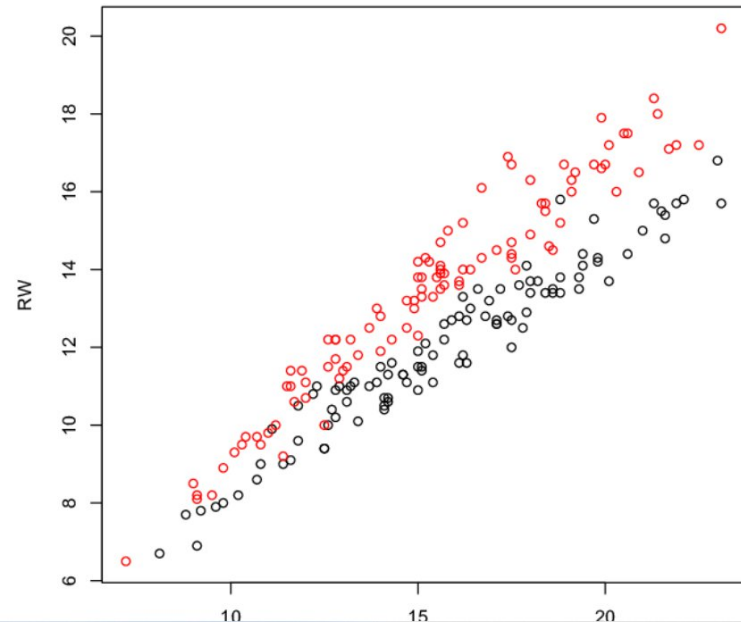
Univariate:

- **Univariate analysis** involves the examination and analysis of a single variable. The primary objective is to understand the distribution and characteristics of this variable within a dataset. Univariate analysis includes calculating measures of central tendency (such as mean, median, and mode) and measures of dispersion (such as range, variance, and standard deviation). It also involves creating visualizations like histograms, bar charts, and box plots. This type of analysis is fundamental in statistics as it provides a foundational understanding of each variable independently before considering their interactions with other variables.



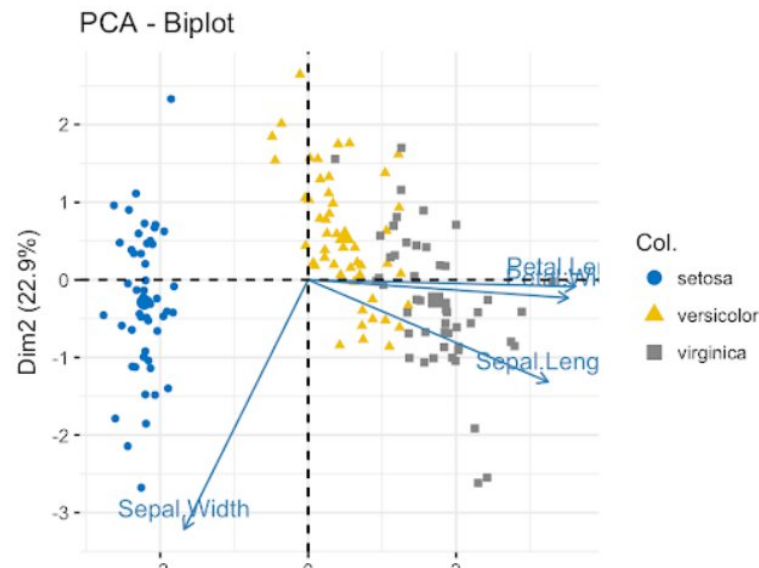
Bivariate:

- **Bivariate analysis** examines the relationship between two variables. This type of analysis aims to determine the association or correlation between the variables, exploring whether and how one variable influences or is related to another. Techniques used in bivariate analysis include scatter plots for visualizing relationships, correlation coefficients for measuring the strength and direction of linear relationships, and cross-tabulations for categorical data. Regression analysis, particularly simple linear regression, is also a common tool in bivariate analysis to model the relationship between an independent variable and a dependent variable. Bivariate analysis is crucial for identifying and understanding direct interactions and dependencies between two variables.



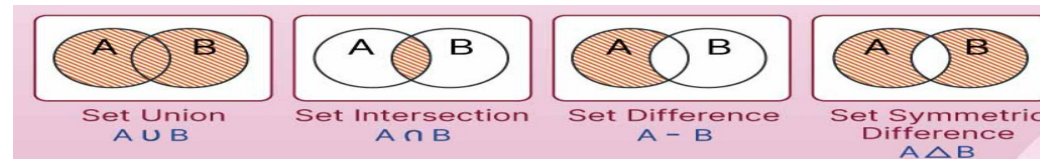
Multivariate:

- **Multivariate analysis** extends beyond two variables to simultaneously analyze three or more variables. This type of analysis aims to understand complex relationships and interactions among multiple variables, often in a more holistic manner. Techniques for multivariate analysis include multiple regression analysis, principal component analysis (PCA), factor analysis, and cluster analysis. These methods help in identifying patterns, reducing dimensionality, and uncovering underlying structures in the data. Multivariate analysis is particularly powerful in fields like data science, market research, and social sciences, where understanding the interplay among multiple factors is essential for comprehensive insights and decision-making.



Set and matrix representations ,relations:

- A set is a fundamental concept that refers to a collection of distinct objects, considered as an object in its own right. These objects, called elements or members, can be anything: numbers, people, letters, other sets, etc. Sets are typically denoted using curly braces, such as $\{a,b,c\}$ where a , b , and c are the elements of the set.



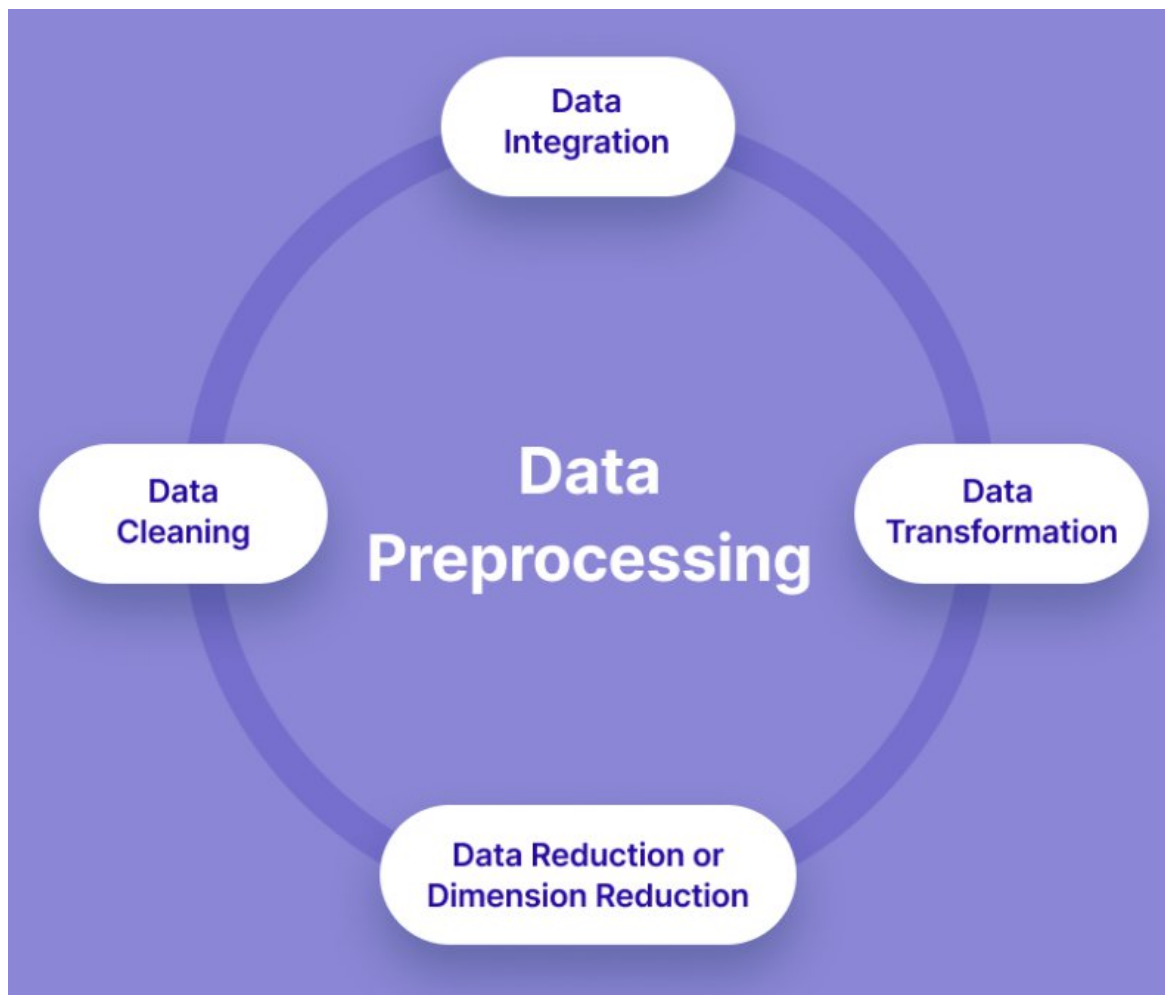
- A matrix representation is a way to encode algebraic structures, such as groups, rings, and vector spaces, into matrices, thereby facilitating their manipulation and analysis. By transforming abstract elements into concrete arrays of numbers, matrix representations allow for the application of linear algebra techniques, making complex problems more tractable.

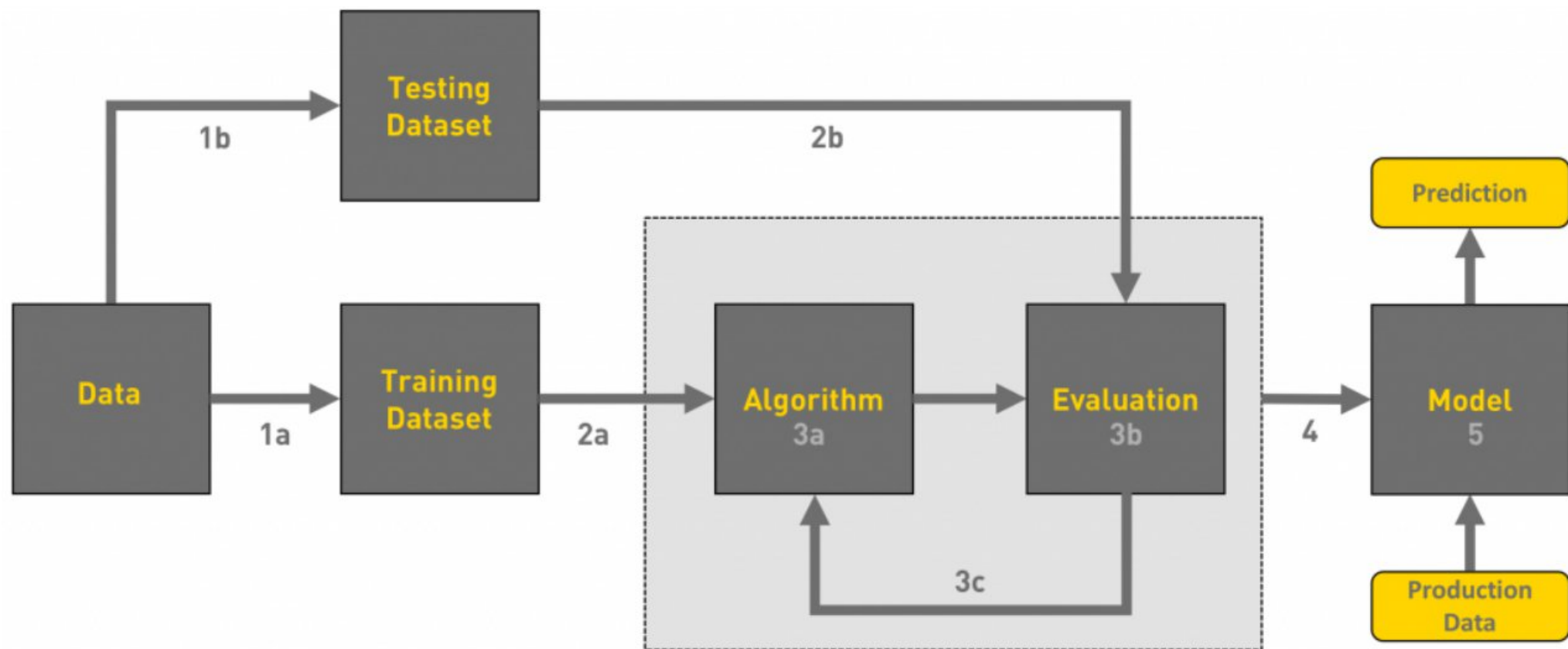
$$\mathbf{A} = \begin{matrix} & \begin{matrix} \xrightarrow{\text{Row (m)}} \end{matrix} \\ \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} & \begin{matrix} \downarrow \text{Columns (n)} \end{matrix} \end{matrix}$$

- A relation matrix is a two-dimensional array that represents the relationship between two sets using an algebraic method.

What is Data Preprocessing?

- Data preprocessing is a critical step in the data analysis and machine learning pipeline, involving a series of operations aimed at transforming raw data into a clean, structured format suitable for analysis.
- Key steps in data preprocessing include data cleaning, where missing values are handled, and inconsistencies or errors are corrected.
- Data integration follows, combining data from multiple sources to provide a unified view.
- Data transformation is another crucial step, involving normalization or scaling to ensure that the data fits within a specific range, making it suitable for algorithms that are sensitive to the scale of input data.
- Feature extraction and selection reduce the dimensionality of the data, focusing on the most relevant attributes, which helps in improving model efficiency and interpretability.





Handling Missing values:

- Handling missing values is a critical step in data preprocessing that can significantly impact the performance and accuracy of machine learning models.
- Missing data can arise from various sources, such as human error during data entry, equipment malfunctions, or incomplete data collection processes.
- Data can be missing for many reasons like technical issues, human errors, privacy concerns, data processing issues, or the nature of the variable itself.

`.isnull()`

Identifies missing values in a Series or
DataFrame

`.isna()`

similar to `notnull()` but returns True for
missing values and False for non-missing
values.

`.info()`

Displays information about the DataFrame,
including data types, memory usage, and
presence of missing values.

`dropna()`

Drops rows or columns containing missing
values based on custom criteria.

`drop_duplicates()`

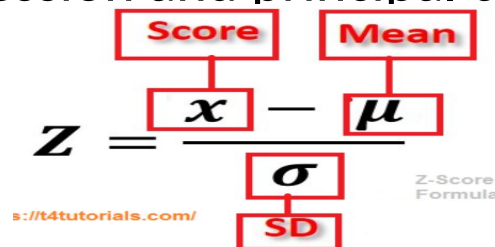
Removes duplicate rows based on
specified columns.

Data Normalization

- Normalization is a critical step in data preprocessing, particularly when working with machine learning algorithms.
- It involves adjusting the values of numeric data features to a common scale, without distorting differences in the ranges of values.
- Two common normalization techniques are min-max normalization and z-score normalization.
- Min-max normalization, also known as feature scaling, transforms the data to fit within a specific range, typically [0, 1]

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

- Z-score normalization, or standardization, adjusts the data to have a mean of zero and a standard deviation of one.
- Standardization is especially beneficial for algorithms that assume data is normally distributed, such as linear regression and principal component analysis.



The diagram illustrates the Z-Score Formula with variables highlighted in red boxes and labeled. The formula is $Z = \frac{x - \mu}{\sigma}$. Above the 'x' is a box labeled 'Score'. Above the ' μ ' is a box labeled 'Mean'. Below the ' σ ' is a box labeled 'SD' (Standard Deviation). The text 'Z-Score Formula' is written to the right of the denominator. A small URL 's://t4tutorials.com/' is visible at the bottom left.

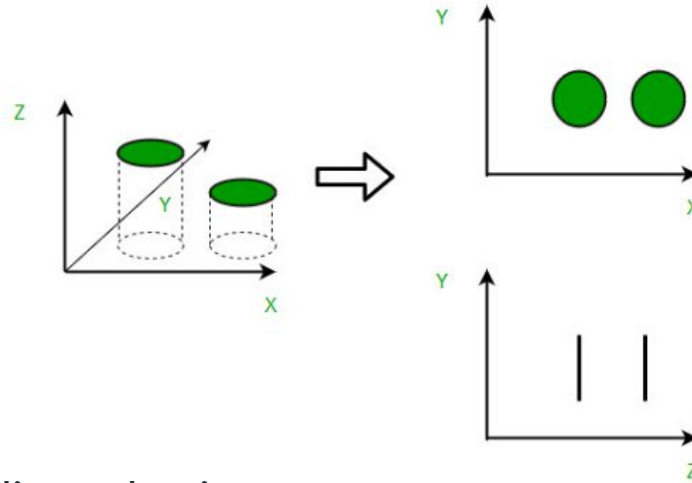
$$Z = \frac{x - \mu}{\sigma}$$

s://t4tutorials.com/ Z-Score Formula

Dimensionality Reduction

- Dimensionality reduction is a crucial step in the preprocessing of data, particularly in the realm of machine learning and data analysis.
- It involves reducing the number of random variables under consideration, simplifying the dataset while retaining its essential features.
- Firstly, it helps in mitigating the "curse of dimensionality," which refers to various phenomena that arise when analyzing and organizing data in high-dimensional spaces.
- As dimensions increase, the volume of the space increases exponentially, leading to sparse data that complicates analysis and model training.
- Dimensionality reduction techniques address this by transforming the data into a lower-dimensional space that captures most of the variability or discriminatory information.
- Dimensionality reduction enhances computational efficiency. High-dimensional data can be computationally expensive to process and can lead to increased training times for machine learning models.
- By reducing the number of features, the computational burden is significantly lowered, making the analysis more feasible and faster.

Dimensionality Reduction



- Two components of dimensionality reduction:

Feature selection: In this, we try to find a subset of the original set of variables, or features, to get a smaller subset which can be used to model the problem. It usually involves three ways:

- Filter
- Wrapper
- Embedded

Feature extraction: This reduces the data in a high dimensional space to a lower dimension space, i.e. a space with lesser no. of dimensions.

Feature Selection

- Feature selection is a technique in machine learning and data preprocessing that involves selecting a subset of relevant features (variables, predictors) from the original set of features in a dataset.
- The main objective of feature selection is to improve the performance of machine learning models by removing irrelevant, redundant, or noisy features.
- This process enhances the efficiency of model training, reduces overfitting, and improves the interpretability of models.

1. Filter method:

- The filter method for feature selection involves evaluating the relevance of each feature independently of any machine learning model.
- This method uses statistical techniques to assess the relationship between each feature and the target variable, selecting features based on their scores from these assessments.
- Common statistical measures used in filter methods include correlation coefficients for continuous data, chi-square tests for categorical data, and mutual information for non-linear relationships.
- By filtering out features that are less relevant or redundant before model training, filter methods improve the efficiency and performance of models without being computationally intensive.

2. Wrapper method:

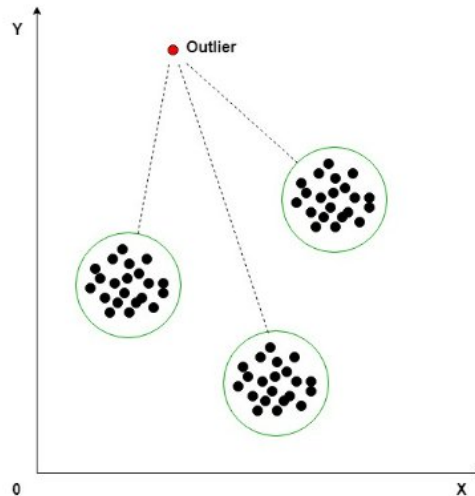
- The wrapper method for feature selection involves using a predictive model to evaluate the combination of features and iteratively adding or removing features based on their impact on model performance.
- Common techniques within wrapper methods include forward selection, where features are added one at a time, backward elimination, where features are removed one at a time, and recursive feature elimination (RFE), which repeatedly builds the model and eliminates the least important features.

3. Embedded method:

- Embedded methods are a type of feature selection technique that integrate the process of feature selection directly into the model training phase.
- This integration allows these methods to consider feature interactions and dependencies more effectively.
- Examples include regularization techniques like Lasso (L1 regularization), which penalizes the absolute size of coefficients, effectively shrinking some to zero and thus performing feature selection.

Outlier Reduction

- Outlier reduction, also known as outlier detection and removal, is a data preprocessing technique used to identify and mitigate the impact of outliers in a dataset.
- Outliers are data points that significantly deviate from the majority of the data and can arise due to measurement errors, data entry mistakes, or genuine variability in the data.
- Techniques for outlier reduction include statistical methods (e.g., Z-score, IQR), clustering methods (e.g., DBSCAN), and machine learning approaches (e.g., isolation forests).
- By detecting and handling outliers, we can ensure a more robust and reliable analysis, leading to more accurate and



Methods of Outlier Reduction

1. Statistical Methods:

- **Z-Score:** This method calculates the standard deviation of the data points and identifies outliers as those with Z-scores exceeding a certain threshold (typically 3 or -3).
- **Interquartile Range (IQR):** IQR identifies outliers as data points falling outside the range defined by $Q1 - k * (Q3 - Q1)$ and $Q3 + k * (Q3 - Q1)$, where $Q1$ and $Q3$ are the first and third quartiles, and k is a factor (typically 1.5).

2. Distance-Based Methods:

- **K-Nearest Neighbors (KNN):** KNN identifies outliers as data points whose K nearest neighbors are far away from them.
- **Local Outlier Factor (LOF):** This method calculates the local density of data points and identifies outliers as those with significantly lower density compared to their neighbors.

3. Clustering-Based Methods:

- **Density-Based Spatial Clustering of Applications with Noise (DBSCAN):** In DBSCAN, clusters data points based on their density and identifies outliers as points not belonging to any cluster.
- **Hierarchical clustering:** Hierarchical clustering involves building a hierarchy of clusters by iteratively merging or splitting clusters based on their similarity. Outliers can be identified as clusters containing only a single data point or clusters significantly smaller than others.

Importance of outlier detection

- 1. Biased models:** Outliers can bias a machine learning model towards the outlier values, leading to poor performance on the rest of the data. This can be particularly problematic for algorithms that are sensitive to outliers, such as linear regression.
- 2. Reduced accuracy:** Outliers can introduce noise into the data, making it difficult for a machine learning model to learn the true underlying patterns. This can lead to reduced accuracy and performance.
- 3. Reduced interpretability:** Outliers can make it difficult to understand what a machine learning model has learned from the data. This can make it difficult to trust the model's predictions and can hamper efforts to improve its performance.

Statistics Library (R):

- The R Language stands out as a powerful tool in the modern era of statistical computing and data analysis.
- Here are several reasons why professionals across various fields prefer R:

1. Comprehensive Statistical Analysis:

- R language is specifically designed for statistical analysis and provides a vast array of statistical techniques and tests, making it ideal for data-driven research.

2. Extensive Packages and Libraries:

- The R Language boasts a rich ecosystem of packages and libraries that extend its capabilities, allowing users to perform advanced data manipulation, visualization, and machine learning tasks with ease.

3. Strong Data Visualization Capabilities:

- R language excels in data visualization, offering powerful tools like ggplot2 and plotly, which enable the creation of detailed and aesthetically pleasing graphs and plots.

Numpy

- NumPy is a general-purpose array-processing package.
- It provides a high-performance multidimensional array object and tools for working with these arrays.
- It is the fundamental package for scientific computing with Python. It is open-source software.

Some of these important features include:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Arrays in NumPy

- NumPy's main object is the homogeneous multidimensional array.
- It is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers.
- In NumPy, dimensions are called axes. The number of axes is rank.
- NumPy's array class is called ndarray. It is also known by the alias array.

Pandas

- Pandas is a powerful and open-source Python library.
- The Pandas library is used for data manipulation and analysis.
- Pandas consist of data structures and functions to perform efficient operations on data.
- Pandas is well-suited for working with **tabular data**, such as **spreadsheets** or **SQL tables**.

Here is a list of things that we can do using Pandas.

- Data set cleaning, merging, and joining.
- Easy handling of missing data (represented as NaN) in floating point as well as non-floating point data.
- Columns can be inserted and deleted from DataFrame and higher-dimensional objects.
- Powerful group by functionality for performing split-apply-combine operations on data sets.
- Data Visualization.

Pandas generally provide two data structures for manipulating data. They are:

- Series
- DataFrame

Pandas Series

- A Pandas Series is a one-dimensional labeled array capable of holding data of any type (integer, string, float, Python objects, etc.).
- The axis labels are collectively called **indexes**.
- The Pandas Series is nothing but a column in an Excel sheet.
- Pandas Series can be created from lists, dictionaries, scalar values, etc.

```
import pandas as pd
import numpy as np

# Creating empty series
ser = pd.Series()
print("Pandas Series: ", ser)

# simple array
data = np.array(['g', 'e', 'e', 'k', 's'])

ser = pd.Series(data)
print("Pandas Series:\n", ser)
```

```
Pandas Series: Series([], dtype: float64)
Pandas Series:
0    g
1    e
2    e
3    k
4    s
dtype: object
```

Pandas DataFrame

- Pandas DataFrame is a two-dimensional data structure with labeled axes (rows and columns).
- Pandas DataFrame is created by loading the datasets from existing storage (which can be a SQL database, a CSV file, or an Excel file).
- Pandas DataFrame can be created from lists, dictionaries, a list of dictionaries, etc.

```
import pandas as pd

# Calling DataFrame constructor
df = pd.DataFrame()
print(df)

# list of strings
lst = ['Geeks', 'For', 'Geeks', 'is', 'portal', 'for', 'Geeks']

# Calling DataFrame constructor on list
df = pd.DataFrame(lst)
print(df)
```

```
Empty DataFrame
Columns: []
Index: []
0
0    Geeks
1     For
2    Geeks
3      is
4  portal
5     for
6    Geeks
```

Scipy

- SciPy is a scientific computation library that uses NumPy underneath.
- SciPy stands for Scientific Python.
- It provides more utility functions for optimization, stats and signal processing.
- SciPy has optimized and added functions that are frequently used in NumPy and Data Science.
- It is designed on the top of Numpy library that gives more extension of finding scientific mathematical formulae like Matrix Rank, Inverse, polynomial equations, LU Decomposition, etc. Using its high-level functions will significantly reduce the complexity of the code and helps better in analyzing the data.
- Use descriptive statistics from SciPy's stats module to gain insights into the dataset.
- Calculate measures such as mean, median, standard deviation, skewness, kurtosis, etc.