Chapter #3

“Data Understanding and Preprocessing”

10 Hours | 12 Marks

Compiled By
Er. Sushil Dyopala

Topics

1. Types of data: Structured, unstructured, semi-structured
2. Data preprocessing requirements
3. Data sources and collection methods
4. Data cleaning and preparation
5. Data wrangling and associated tools
6. Data enrichment, validation and publishing
7. Data transformation and normalization
8. Dimensionality reduction linear factor model, principal component analysis (PCA)

1.1 What is Data?

- Data is a collection of facts, observations, or measurements that can be processed and analyzed to extract meaningful insights.
- Data can be in the form of numbers, text, images, videos, or any digital representation of information.

1.1Types of Data

- Structured Data or Organized Data
  - Highly organized and stored in a predefined format like tables in a relational database.
  - Easy to access, query, and analyze using SQL.
  - Example: Customer information like Name, Age, Email, Phone Number stored in a tabular database
- Unstructured Data or Unorganized Data
  - Does not have a predefined format or organization.
  - Difficult to store in traditional relational databases.
  - Example: Emails, social media posts, images, videos, audio files, PDFs.
- Semi-Structured Data
  - Has some structure but does not fit into traditional databases.
  - Uses tags or metadata to organize information.
  - Example: JSON, XML, NoSQL databases.

1.2 Quantitative versus Qualitative Data

- **Quantitative Data**
  - Quantitative data represents numerical values that can be measured or counted.
  - It deals with quantities and can be used for mathematical calculations and statistical analysis.
  - It can be discrete or continuous data.
  - Discrete Data: This describes data that is counted. It can only take on certain values. e.g. number of students
  - Continuous Data: This describes data that is measured. It exists on an infinite range of values. e.g. height of students
- Characteristics of Quantitative Data
  - Measurable and numerical
  - Can be used in mathematical computations
  - Can be expressed in graphs and charts
  - Objective in nature
- Example: Age of a person (25 years, 30 years), Speed of a car (60 km/h, 80 km/h).

1.2 Quantitative versus Qualitative Data

- **Qualitative Data**
  - Qualitative data represents categorical information that describes characteristics, qualities, or labels.
  - It does not involve numerical values but rather descriptive attributes.
  - This data cannot be described using basic mathematics.
  - It can be nominal [e.g. gender: (Male, Female, Other)]or ordinal data [e.g. Economic Class: (Low, Middle, High)].
- Characteristics of Qualitative Data
  - Descriptive and categorical
  - Cannot be measured numerically
  - Often subjective in nature
  - Used for classification
- Example: Eye color (Brown, Blue, Green), Feedback rating (Excellent, Good, Average, Poor).

1.2 Quantitative versus Qualitative Data

- **Illustration: Coffee Shop Data**
- Say that we were processing observations of coffee shops in a major city using the following five descriptors (characteristics):
- Data: Coffee Shop
  - Name of coffee shop
  - Revenue (in thousands of dollars)
  - Zip code
  - Average monthly customers
  - Country of coffee origin
- Each of these characteristics can be classified as either quantitative or qualitative.

1.2 Quantitative versus Qualitative Data

- Name of coffee shop – Qualitative
    - The name of a coffee shop is not expressed as a number and we cannot perform math on the name of the shop.
- Revenue – Quantitative
    - How much money a cafe brings in can definitely be described using a number. Also, we can do basic operations such as adding up the revenue for 12 months to get a year's worth of revenue.
- Zip code – Qualitative
    - A zip code is always represented using numbers, but what makes it qualitative is that it does not fit the second part of the definition of quantitative—we cannot perform basic mathematical operations on a zip code. If we add together two zip codes, it is a nonsensical measurement. We don't necessarily get a new zip code and we definitely don't get "double the zip code".
- Average monthly customers – Quantitative
    - Describing this factor using numbers and addition makes sense. Add up all of your monthly customers and you get your yearly customers.
- Country of coffee origin – Qualitative
    - Country is described using a name and not numbers.

1.2 Quantitative versus Qualitative Data

Feature	Quantitative Data	Qualitative Data
Definition	Numerical data that represents measurable quantities	Descriptive data that represents characteristics or categories
Nature	Objective, measurable	Subjective, descriptive
Types	Discrete (whole numbers) & Continuous (fractions/decimals)	Nominal (no order) & Ordinal (ordered)
Examples	Age, Height, Income, Temperature	Color, Gender, Feedback, Nationality
Mathematical Operations	Can be added, subtracted, averaged, etc.	Cannot perform arithmetic operations
Analysis Methods	Statistical techniques (mean, median, standard deviation)	Classification, thematic analysis

1.3 The Four Levels of Data

- It is generally understood that a specific characteristic (feature/column) of structured data can be broken down into one of four levels of data.
- The levels are:
  - The nominal level
  - The ordinal level
  - The interval level
  - The ratio level

1.3 The Four Levels of Data: The Nominal Level

- The nominal level is the first and the most basic level of data measurement.
- It categorizes data into distinct groups, but these categories do not have any intrinsic order or ranking.
- Characteristics:
  - Data is qualitative (categorical)
  - No meaningful order or ranking between categories
  - Cannot perform mathematical operations (only counts or mode can be used)
- Allowed Operations:
  - Classification (Grouping items into categories)
  - Frequency Count (Counting occurrences of each category)
  - Mode (Finding the most common category)

Examples:
- Gender: Male, Female, Other
- Blood Type: A, B, AB, O
- Nationality: American, French, Indian, Nepali
- Eye Color: Brown, Blue, Green

1.3 The Four Levels of Data: The Ordinal Level

- The ordinal level categorizes data just like the nominal level, but with a meaningful order or ranking.
- However, the difference (interval) between the categories is not uniform or measurable.
- Characteristics:
  - Data is qualitative (categorical) but ordered
  - Categories have a meaningful ranking/order
  - Differences between categories are not measurable
  - Cannot perform arithmetic operations (e.g., subtraction or division)
- Allowed Operations:
  - Ranking/Sorting (Ordering categories) and Comparison
  - Median (Finding the middle value in ordered data)
  - Mode (Finding the most common category)

Examples:

- Education Level: High School < Bachelor's < Master's < PhD
- Customer Satisfaction: Poor < Average < Good < Excellent
- Movie Ratings: ⭐ < ⭐⭐ < ⭐⭐⭐ < ⭐⭐⭐⭐ < ⭐⭐⭐⭐⭐

1.3 The Four Levels of Data: The Interval Level

- The interval level consists of numerical data where differences between values are meaningful, but there is no natural zero (zero does not indicate the absence of value).
- Data at the interval level allows meaningful subtraction between data points.
- Characteristics:
  - Data is quantitative (numerical)
  - Ordered with equal intervals between values
  - No true zero point (zero does not mean "nothing")
  - Can perform addition and subtraction, but not division or ratios
- Allowed Operations:
  - Addition & Subtraction (Finding differences between values)
  - Mean, Median, Mode (Statistical summaries)

Examples:
- Temperature in: (0°C ≠ No temperature) and (20°C is not "twice as hot" as 10°C)
- Years on a Calendar

1.3 The Four Levels of Data: The Ratio Level

- The ratio level is the highest and most precise level of measurement.
- It has all the properties of the interval level, but with a true zero (zero means the absence of the quantity).
- Characteristics:
  - Data is quantitative (numerical)
  - Ordered with equal intervals between values
  - True zero exists (zero means "nothing")
  - Can perform all mathematical operations (addition, subtraction, multiplication, division)
- Allowed Operations:
  - All Arithmetic Operations (Addition, Subtraction, Multiplication, Division)
  - Statistical Analysis (Mean, Standard Deviation, Ratios, etc.)

Examples:
- Weight: (0 kg means no weight, 60 kg is twice as heavy as 30 kg)
- Speed of a Car: (0 km/h means no motion)

1.3 The Four Levels of Data

Level	Data Type	Ordered?	Equal Intervals?	True Zero?	Example
Nominal	Categorical	❌ No	❌ No	❌ No	Gender, Nationality, Eye Color
Ordinal	Categorical	✅ Yes	❌ No	❌ No	Education Level, Satisfaction Rating
Interval	Numerical	✅ Yes	✅ Yes	❌ No	Temperature (°C, °F), IQ Score
Ratio	Numerical	✅ Yes	✅ Yes	✅ Yes	Height, Weight, Income, Kelvin Temperature

2.1 Data Preprocessing Requirements

- Real-world data refers to raw data collected from various sources such as sensors, databases, social media, surveys, and transactions.
- This data is often incomplete, inconsistent, noisy, or redundant, making it unsuitable for direct analysis.
- Examples of Real-World Data:
  - Healthcare: Patient records, medical test results, wearable device data.
  - E-commerce: Customer purchase history, product reviews, website clicks.
  - Finance: Stock market trends, bank transactions, fraud detection.
  - Social Media: Tweets, Facebook posts, Instagram stories.
  - IoT & Sensors: Traffic data, temperature sensors, smart home devices.

2.2 Need of Data Preprocessing

- Since real-world data is often messy, data preprocessing is essential to ensure accurate and meaningful analysis.
- Without preprocessing, models may produce biased, misleading, or incorrect results.
- Key Reasons for Data Preprocessing:
  - Handles Missing Data: Many datasets have missing values (e.g., empty fields in surveys).
  - Removes Noise & Inconsistencies: Data may have errors, duplicates, or outliers that need correction.
  - Standardizes Data: Different sources may have inconsistent formats (e.g., date formats, units).
  - Improves Model Performance: Clean data ensures better accuracy in machine learning models.
  - Reduces Dimensionality: Helps in selecting the most important features for analysis.

2.3 General Steps of Data Preprocessing

1. **Data Collection**
   - Gather data from various sources (databases, APIs, web scraping, sensors).
2. **Data Cleaning**
   - Handling Missing Values:
     - Fill with mean/median/mode (imputation) or remove incomplete rows.
   - Removing Duplicates:
     - Detect and delete duplicate records to prevent bias.
   - Handling Outliers:
     - Detect anomalies using statistical methods (e.g., Z-score, IQR).
3. **Data Transformation**
   - Normalization:
     - Scale numeric values to a specific range (e.g., [0,1] or [-1,1]).
   - Encoding Categorical Data:
     - Convert text labels (e.g., "Male"/"Female") into numbers (e.g., 0/1).
   - Feature Engineering:
     - Create new meaningful features (e.g., extracting "year" from a "date" column).

2.3 General Steps of Data Preprocessing

4. Data Reduction

- Dimensionality Reduction:
  - Use PCA (Principal Component Analysis) or Feature Selection to reduce complexity.
- Sampling:
  - Use a subset of data when working with large datasets to reduce computation time.

5. Data Splitting

- Training, Validation, and Test Split:
  - Split data into training (for learning), validation (for tuning), and test (for evaluation).

2.4 Tools for Data Preprocessing

- Python Libraries: Numpy, Pandas, scikit-learn
- R Programming Language
- SQL: MySQL, PostgreSQL, SQLite
- ETL (Extract, Transform, Load) Tools: Power BI
- Excel
- Pytorch, Tensorflow for machine learning and deep learning

3.1 Data Collection

- Data collection is the process of gathering information from different sources to analyze and make informed decisions.
- Need for Data Collection
  - Informed Decision-Making: Organizations rely on accurate data to make strategic decisions.
  - Training Machine Learning Models: AI and ML models require high-quality data for training and predictions.
  - Understanding Market Trends: Businesses use collected data for customer insights, sales forecasting, and product improvement.
  - Scientific Research & Development: Researchers gather data to test hypotheses, analyze trends, and discover new knowledge.
  - Policy & Governance: Governments collect population, economic, and social data for policy making.

3.1 Data Sources: Primary Data Source

- Data collected directly from the source for a specific purpose is called primary data.
- Primary data is also referred as First Hand Data.
- Characteristics of Primary Data:
  - Collected first-hand from direct sources.
  - More accurate and relevant to specific research.
  - Expensive and time-consuming to collect.
- Examples of Primary Data Sources:
  - Surveys & Questionnaires: Online forms, Google Forms, polls.
  - Interviews: Face-to-face, phone, or online conversations.
  - Experiments: Scientific trials, laboratory research.
  - Observations: Watching and recording behaviors.
  - Sensor Data: IoT devices, temperature sensors, traffic monitoring.

3.1 Data Sources: Secondary Data Source

- Data collected by someone else and used for different purposes is called secondary data.
- Characteristics of Secondary Data:
  - Easily available and less costly compared to primary data.
  - May not be as accurate or tailored to specific needs.
  - Helps in historical analysis, benchmarking, and trend forecasting.
- Examples of Secondary Data Sources:
  - Government Reports: Census data, economic surveys.
  - Research Papers & Journals: Scientific studies, white papers.
  - Online Databases: Kaggle, Google Dataset Search.

3.2 Internal, External, and Open-Source Data

- **Internal Data** (Within the Organization)
  - Data generated and stored within a company or institution.
  - Examples: Sales records, employee data, customer feedback, CRM databases.
  - Usage: Business analytics, HR insights, customer personalization.
- **External Data** (Outside the Organization)
  - Data collected from sources outside the organization.
  - Examples: Market trends, competitor analysis, industry benchmarks.
  - Usage: Market research, competitive analysis, investment planning.
- **Open-Source Data** (Publicly Available Data)
  - Freely available data accessible to the public.
  - Examples: Government census, weather data, Wikipedia, OpenStreetMap, Kaggle datasets.
  - Usage: Research, AI model training, open-data projects.

3.3 Data Collection Methods

Method	Description	Example
Surveys & Questionnaires	Structured questions to gather opinions, feedback	Google Forms, customer satisfaction surveys
Interviews	One-on-one discussions to collect detailed information	Job interviews, expert opinions
Observations	Watching and recording behaviors/events in real-time	Store foot traffic analysis, CCTV monitoring
Web Scraping	Automated extraction of data from websites	Scraping e-commerce product prices
Sensor Data Collection	IoT devices and sensors capturing real-time data	Weather stations, smart meters
Experiments & Tests	Controlled trials and scientific experiments	A/B testing, medical trials
Database Extraction	Fetching data from structured databases	SQL queries from company CRM systems

3.4 Data Collection Steps

1. Define Objectives
    - What problem are you solving?
    - Example: A company wants to analyze customer purchase behavior.
2. Identify Data Sources
    - Choose between primary vs. secondary sources.
    - Example: Use sales records (internal data) and market reports (external data).
3. Select Collection Method
    - Choose surveys, interviews, sensors, or web scraping based on the need.
4. Gather Data
    - Implement the data collection strategy.
    - Ensure accuracy, consistency, and security.

4.1 Data Cleaning

- Data cleaning (or data cleansing) is the process of detecting and correcting errors, inconsistencies, and inaccuracies in a dataset to improve data quality.
- Common Issues in Raw Data:
  - Missing values (NaN, NULL, empty cells)
  - Duplicates
  - Inconsistent formats (e.g., date formats, currency symbols)
  - Outliers (extreme values affecting statistical analysis)
  - Typographical errors (spelling mistakes, incorrect labels)
- Why is Data Cleaning Important?
  - Improves Accuracy – Clean data leads to better model performance and decision-making.
  - Removes Redundancy – Eliminates duplicate or irrelevant data, reducing storage costs.
  - Enhances Reliability – Ensures consistency across different data sources.
  - Prepares Data for Analysis – Helps avoid misleading conclusions due to data errors.

4.2 Handling Missing Values

- Missing values occur when no data is recorded for certain observations in a dataset.
- They can negatively impact data analysis and machine learning model performance if not handled properly.
- Common Causes of Missing Values
    - Human error – Data entry mistakes, skipped survey questions.
    - Data corruption – Loss during transmission or storage failure.
    - Privacy concerns – Respondents avoid answering sensitive questions.
    - Sensor malfunctions – IoT devices fail to capture data.

4.2 Techniques for Handling Missing Values

1. Removing Missing Values

a. Deleting Rows with Missing Values
   - If only a few rows contain missing data, removing them might be a simple solution.
   - Best When: The dataset is large and missing values are rare (<5%).
   - Simple and effective for small amounts of missing data.
   - Leads to data loss if many rows are dropped.

b. Deleting Columns with Too Many Missing Values
   - If a column has more than 30-50% missing values, consider removing it.
   - Best When: The column is not critical for analysis.
   - Removes redundant or low-value features.
   - Risk of losing valuable information.

4.2 Techniques for Handling Missing Values

2. Imputing Missing Values

- When deleting data is not an option, we use imputation techniques.
- **Mean Imputation:** Replaces missing values with the average of the column.
- **Median Imputation:** Replaces missing values with the middle value (better for skewed data).
- **Mode Imputation:** Replaces missing values with the most frequent value (useful for categorical data).

4.2 Techniques for Handling Missing Values

3. Interpolation

- Estimates missing values based on trends in the data.
- Common Methods: Linear, polynomial, spline interpolation.
- Useful for numerical and time-series data.

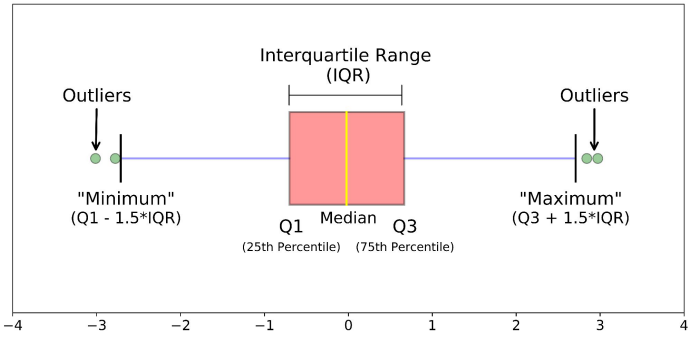
4. Regression Imputation

- Uses a regression model to predict missing values based on other features.
- More precise than simple statistical methods.
- Requires strong correlations between variables.

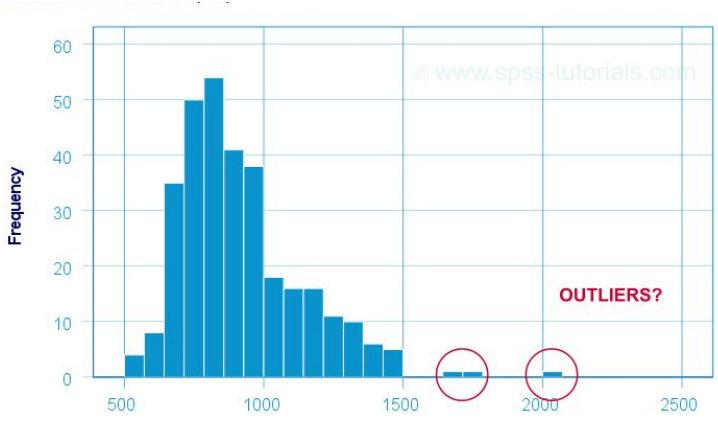
4.3 Handling Outliers

- Outliers are extreme values that significantly differ from the majority of data points in a dataset.
- **Causes of Outliers**
  - Measurement errors – Incorrect sensor readings, data entry mistakes.
  - Natural variation – Genuine extreme values in populations.
  - Data processing errors – Incorrect unit conversions, misinterpretations.
  - Rare events – Fraudulent transactions, anomalies in medical diagnoses.
- Why do we need to handle outliers?
  - They can skew statistical measures (mean, standard deviation).
  - They can mislead machine learning models, leading to poor predictions.
  - They may indicate errors, fraud, or rare events worth investigating.

4.3 Methods to Detect Outliers



1. **Box Plot**
   - Outliers appear as points beyond the whiskers.
2. **Histogram**
   - Outliers appear as isolated bars.
3. **Z-Score Method** (Standard Deviation Approach)
   - Measures how far a data point is from the mean in standard deviations (σ).
   - Any value with |Z-score| > 3 is often considered an outlier.
4. **Interquartile Range (IQR) Method**
   - Based on the spread of the middle 50% of data.
   - Outliers are values outside:
     i. Lower Bound=Q1−1.5×IQR
     ii. Upper Bound=Q3+1.5×IQR



4.3 Methods to Detect Outliers

1. **Removing Outliers**
   - This method involves removing the outliers from the dataset.
   - Pros:
     - Simple to implement.
     - Removes extreme values that could distort analysis.
   - Cons:
     - Can lead to loss of data, especially if the dataset is small.
     - Risk of removing valid data points that are naturally extreme.
2. **Capping Outliers**
   - Replace outliers with a specified percentile value.
   - Pros:
     - Keeps all data points.
     - Reduces the effect of outliers without losing data.
   - Cons:
     - The choice of percentile is arbitrary and may not always be optimal.

4.3 Methods to Detect Outliers

3. Imputation

- Replace outliers with mean, median, or mode.
- Pros:
    - Simple to implement.
    - Retains all data points.
- Cons:
    - Can introduce bias if outliers are numerous.
    - Not suitable if outliers contain valuable information.

4. Transformation

- Apply transformations like logarithmic, square root, or Box-Cox to reduce the impact of outliers.
- Pros:
    - Effective for right-skewed distributions.
    - Can normalize the data.
- Cons:
    - Not suitable for all types of data.
    - Log transformation is not defined for zero or negative values.

4.3 Methods to Detect Outliers

5. Z-Score Method

- Identify and remove outliers based on the standard deviation from the mean.
- Data points with a Z-score greater than a specified threshold (typically 3 or -3) are considered outliers.
- Pros:
  - Effective for normally distributed data.
  - Data-driven approach.
- Cons:
  - Not suitable for non-normal distributions.

4.4 Measures of Data Quality

- Accuracy – Data correctly represents real-world values.
- Completeness – No missing values.
- Consistency – Data across different sources is uniform.
- Validity – Data follows defined rules (e.g., date format, numeric range).
- Timeliness – Data is up to date.
- Uniqueness – No duplicate records exist.

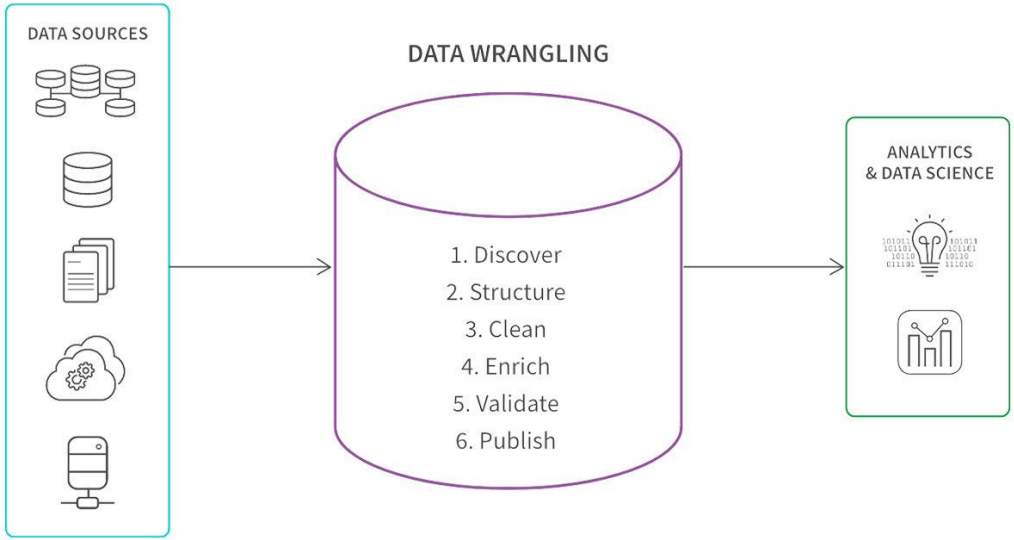
4.5 Benefits and Challenges of Data Cleaning

- Benefits
  - Better Decision-Making: Clean data leads to accurate insights.
  - Improved Model Performance: Reduces bias and errors in ML models.
  - Efficient Resource Utilization: Saves storage space and computational power.
- Challenges
  - Time-Consuming – Cleaning large datasets requires significant effort.
  - Handling Missing Values – Choosing the right imputation method is tricky.
  - Data Inconsistency – Different formats and sources lead to inconsistency.
  - Identifying Outliers – Difficult to differentiate real outliers from legitimate variations.
  - Data Security – Sensitive data must be cleaned while maintaining privacy.

5.1 Data Wrangling Process

- Data wrangling (also called data munging) is the process of cleaning, transforming, and structuring raw data into a usable format for analysis or machine learning.
- Data Wrangling is a broader process that involves transforming raw data into a usable format, including cleaning it, while Data Cleaning specifically focuses on identifying and correcting errors, inconsistencies, and inaccuracies within a dataset.
- We will typically go through the data wrangling process prior to conducting any data analysis in order to ensure your data is reliable and complete. This way, we can be confident that the insights we draw are accurate and valuable.

5.1 Data Wrangling Process [6 Steps]



5.1 Data Wrangling Process [6 Steps]

1. **Discover**
    - Initially, your focus is on understanding and exploring the data you've gathered.
    - This involves identifying data sources, assessing data quality, and gaining insights into the structure and format of the data.
    - Your goal is to establish a foundation for the subsequent data preparation steps by recognizing potential challenges and opportunities in the data.
2. **Structure**
    - In the data structuring step, you organize and format the raw data in a way that facilitates efficient analysis.
    - Structuring involves reshaping data, handling missing values, and converting data types. This ensures that the data is presented in a coherent and standardized manner, laying the groundwork for further manipulation and exploration.

5.1 Data Wrangling Process [6 Steps]

3. Cleaning

- Data cleansing is a crucial step to address inconsistencies, errors, and outliers within the dataset. This involves removing or correcting inaccurate data, handling duplicates, and addressing any anomalies that could impact the reliability of analyses.
- By cleaning the data, your focus is on enhancing data accuracy and to make sure the data is as error-free as possible.

4. Enriching

- Enriching data involves enhancing it with additional information to provide more context or depth.
- This can include merging datasets, extracting relevant features, or incorporating external data sources.
- The goal is to augment the original dataset, making it more comprehensive and valuable for analysis. If you do add data, be sure to structure and clean that new data.

5.1 Data Wrangling Process [6 Steps]

5. Validating

- Validation ensures the quality and reliability of processed data.
- You'll check for inconsistencies, verify data integrity, and confirm that the data adheres to predefined standards.
- Validation helps in building confidence in the accuracy of the dataset and ensures that it meets the requirements for meaningful analysis.

6. Publishing

- Curated and validated dataset is prepared for analysis to business users.
- This involves documenting data lineage and the steps taken during the entire wrangling process, sharing metadata, and preparing the data for storage or integration into data science and analytics tools.
- Publishing facilitates collaboration and allows others to use the data for their analyses or decision-making processes.

5.2 Tools for Data Wrangling

- Open Source Tools:
  - Python Libraries: Pandas, NumPy, SciPy, scikit-learn
  - Dask (Python) – Handles large datasets using parallel processing.
  - OpenRefine – GUI-based tool for cleaning messy data.
- Other Tools:
  - Microsoft Power Query (Excel & Power BI) – Easy-to-use interface for data wrangling.
  - Google DataPrep (Google Cloud) – Cloud-based data preparation tool.
  - Alteryx – Advanced tool for automated data wrangling.

6.1 Data Enrichment

- Data enrichment is the process of enhancing raw data by adding external or supplementary data to improve its accuracy, completeness, and value.
- It helps businesses make better decisions by providing a more comprehensive view of the data.
- Common ways to enrich data include:
    - Demographic Enrichment – Adding data like age, gender, income, and location to customer records.
    - Geographical Enrichment – Enhancing data with GPS coordinates, city, state, and country information.
    - Behavioral Enrichment – Adding customer behaviors, website visits, purchase history, and interactions.
    - Firmographic Enrichment – Adding company-related data (size, revenue, industry) to business records.
    - Social Media Enrichment – Enhancing customer profiles with social media activity, preferences, and engagement.
- Example (E-commerce Data Enrichment):
    - Raw Data: A customer's name and email.
    - Enriched Data: Adding their purchase history, location, social media preferences, and income range for better personalization.

6.1 Data Enrichment Methods

1. **Data Appending**
    - Data appending combines multiple data sources to create a more holistic data set.
    - It includes adding internal, external, and third-party data sources such as demographic, geographical, behavioral data and merging them into your dataset.
    - Using this, you can make data more centralized for optimized analytics and accessibility.
    - Example: Extracting customer data from financial systems, CRM, and marketing applications and bringing those together.
2. **Data Segmentation**
    - Data segmentation is the process of dividing a data object, like a customer or product, into groups based on a common set of attributes such as age or gender.
    - This segmentation is used to categorize and describe the data much better.
    - Examples: demographic, technographic, behavioral segmentation.

6.1 Data Enrichment Methods

3. Entity Extraction

- In entity extraction, you take unstructured or semi-structured data and extract meaningful structured data from that element.
- With this technique, you can identify people, organizations, and places, as well as temporal expressions such as dates, currency amounts, phone numbers, and times.

4. Derived Attributes

- Derived attributes are fields that are not stored in the original data set but can be derived from one or more fields.
- For example ‘Age’ is very rarely stored but you can derive it based on a ‘date of birth’ field.
- Derived attributes are very useful because often they contain logic that is repeatedly used for analysis.

6.1 Usages of Data Enrichment in Different Sectors

1. Financial Sector:
    - Identifying fraudulent transactions.
    - Enhancing customer profiles with credit scores, income levels, and spending habits.
2. Social Media Data:
    - Improving ad targeting by enriching user demographics and interests.
    - Analyzing sentiment and user interactions.
3. Customer Data (Customer Relationship Management):
    - Improving customer segmentation and personalization.
    - Predicting customer behavior and lifetime value.
4. E-commerce:
    - Enhancing product recommendations based on purchase history.
    - Analyzing browsing behavior to optimize marketing strategies.

6.2 Data Validation

- Data validation is the process of ensuring the accuracy, consistency, and quality of data before using it for analysis or storage.
- It prevents errors caused by incomplete, incorrect, or duplicate data.
- In automated systems, data is entered with minimal or no human supervision. Therefore, it is necessary to ensure that the data that enters the system is correct and meets the desired quality standards.
- The data will be of little use if it is not entered properly and can create bigger downstream reporting issues.

6.2 Types of Data Validation

- Data type validation
  - A data type is the kind of information contained in a data field.
  - Example: checking whether a numerical data field contains a number and rejecting other characters, such as uppercase letters or special symbols.
- Format validation
  - Checks if data follows a specific format.
  - Example: date format of YYYY-MM-DD compared to MM-DD-YYYY.
- Range validation
  - A range is the high and low end of a value and the values that fall between, for example, the range of 1-10.
  - In range validation, only valid values within the range are accepted.
  - Example: For data with a specified latitude range of -20 to 40 degrees, range validation will help reject a value of -25 degrees.

6.2 Types of Data Validation

- Uniqueness Check
  - Some data like IDs or e-mail addresses are unique by nature. A database should likely have unique entries on these fields.
  - A uniqueness check ensures that an item is not entered multiple times into a database.
- Presence check
  - A presence check simply checks that a field isn’t empty.
  - It ensures required fields are not empty (e.g., Name, Email).
- Cross-field validation
  - When two or more fields are checked and compared in relation to each other, this is cross-field validation.
  - An example would be taking the number of concertgoers with VIP tickets, premium seats, and general admission and adding them together to compare with the total number of tickets sold.

6.3 Data Publishing

- Data publishing refers to the process of making cleaned, validated, and structured data available for use by individuals, organizations, or the public.
- Depending on who can access the data and how it is shared, there are different methods of data publishing as:
  - Open Data Publishing
  - Internal Data Publishing
  - Restricted Access Data Publishing
  - Data Publishing via APIs
  - Data Publishing on Marketplaces
  - Data Publishing via Research Papers & Reports

6.3.1 Open Data Publishing

- Data is made freely available to the public.
- Users can download, use, and modify the data without restrictions.
- Often provided by governments, NGOs, and research institutions.
- Example: WHO's COVID-19 statistics
- How is it Published?
    - Data Portals: Websites like Kaggle, GitHub, or Google Dataset Search
    - APIs: Government APIs providing real-time data (e.g., U.S. Census Bureau API)
    - CSV, JSON, or XML Formats: Data files that anyone can download

6.3.2 Internal Data Publishing

- Data is published within an organization for internal use.
- Used for decision-making, analytics, and business intelligence.
- Not accessible to the public.
- Example: Company sales reports shared with executives
- How is it Published?
    - Company Dashboards (e.g., Power BI, Tableau)
    - Internal Databases (e.g., MySQL, Snowflake)
    - Shared Excel Sheets or Google Sheets

6.3.3 Restricted Access Data Publishing

- Data is shared only with specific individuals, teams, or organizations.
- Requires authentication or permission to access.
- Used in industries like healthcare, banking, and research.
- Example: Patient medical records shared only with doctors
- How is it Published?
    - Private APIs (e.g., Financial APIs requiring authentication)
    - Encrypted Cloud Storage (e.g., AWS S3 with restricted access)

6.3.4 Data Publishing via APIs

- Data is made accessible through an Application Programming Interface (API).
- Developers and businesses can query and retrieve data in real-time.
- Example: Google Maps API for location-based services
- How is it Published?
    - REST APIs: Most common method for data exchange over the web
    - GraphQL APIs: Allows specific data retrieval requests
    - Web Scraping APIs: Extracts data from web pages

6.3.5 Data Publishing on Marketplaces

- Organizations sell or distribute high-quality, structured datasets on data marketplaces.
- Businesses can purchase data for marketing, research, and AI model training.
- Example:
  - AWS Data Exchange – Buy and sell datasets
  - Google Cloud Public Datasets – Free & paid datasets
  - Kaggle Datasets – Free community datasets

6.3.6 Data Publishing via Research Papers & Reports

- Data is published through academic journals, white papers, and government reports.
- Used in scientific research, business analysis, and public policy.
- Example:
  - Scientific papers published on platforms like Google Scholar
  - Business reports by consulting firms
  - Government statistics published on portals like World Bank, UN Data

7.1 Data Transformation

- Data transformation is the process of converting raw data into a format suitable for machine learning models.
- It involves:
  - Converting categorical data into numerical format (Encoding)
  - Normalizing numerical data for uniformity
  - Standardizing data to improve model performance

7.2 Categorical Encoding

- Categorical values are non-numeric variables that represent categories and can take on one of a limited, fixed number of possible values.
- Categorical data represents information in labels or categories (e.g., "Red", "Green", "Blue") but ML models work with numbers, so we must convert categorical data into numerical values.
- Categorical Encoding is the process to convert non-numeric categorical values to the numeric values.
- Types of Categorical Encoding
    - One-Hot Encoding (OHE)
    - Ordinal Encoding
    - Label Encoding

7.2.1 One-Hot Encoding (OHE)

- One-hot encoding is the categorical encoding techniques that converts each unique category into a separate binary (0 or 1) column.
- Suitable for nominal (unordered) categories, such as colors, countries, or product types.
- One-hot encoding creates new columns indicating the presence (or absence) of each possible value in the original data.
- Limitation: Creates many columns if there are too many categories, leading to a "curse of dimensionality".

7.2.1 One-Hot Encoding (OHE): Illustration

Original Data

ID	Color
1	Red
2	Green
3	Blue
4	Red

One-Hot Encoded Data

ID	Red	Green	Blue
1	1	0	0
2	0	1	0
3	0	0	1
4	1	0	0

- Each category gets a separate column, and only one column is **activated (1)** for each row.

7.2.2 Ordinal Encoding

- Ordinal encoding is the categorical encoding techniques that assigns integer values to categories based on their order or ranking.
- Suitable for ordinal (ordered) categories, such as education levels, star ratings, or performance ratings.
- Limitation: Assumes that the difference between categories is equal, which may not always be true.

7.2.2 Ordinal Encoding: Illustration

Original Data (Education Level)

ID	Education Level
1	High School
2	Bachelor's
3	Master's
4	PhD

Ordinal Encoded Data

ID	Education Level (Encoded)
1	1
2	2
3	3
4	4

- Here, "High School" is assigned **1**, "Bachelor's" **2**, "Master's" **3**, and "PhD" **4**, preserving the **order**.

7.2.3 Label Encoding

- Ordinal encoding is the categorical encoding techniques that assigns unique integer values to each category without any order.
- Suitable when categories have no ranking or order, such as city names, brand names, or product categories.
- Limitation: The numerical values can be misinterpreted as ordered by ML models.

7.2.3 Label Encoding: Illustration

Original Data (City Names)

ID	City
1	Tokyo
2	London
3	New York
4	Tokyo

Label Encoded Data

ID	City (Encoded)
1	0
2	1
3	2
4	0

- "Tokyo" is assigned **0**, "London" **1**, and "New York" **2**, but there is **no inherent order** among them.

7.3 Data Normalization [Feature Normalization]

- Normalization is a technique used to scale numerical data so that all features have a similar range.
- It ensures that no feature dominates another due to differences in scale.
- Why do we need data normalization?
  - Prevents Bias: Many ML Algorithms rely on distance metrics, and normalization ensures that no feature unfairly dominates.
  - Improves Model Convergence: Helps models like gradient descent-based neural networks to converge faster.
  - Maintains Stability: Prevents numerical instability in computations.
  - Better Performance: Enhances accuracy and efficiency in training machine learning models.

7.3 Data Normalization Type: Min-Max Normalization

- Min-max normalization is usually called as **feature scaling**.
- Min-max normalization performs a linear transformation on the original data in such a way that all the data gets scaled to the range (0, 1).
- Best suited when data has a fixed minimum and maximum (e.g., percentages, image pixel values).
- Min-max Normalization Formula:

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

- Advantage: Keeps original distribution shape and works well for bounded data.
- Disadvantage: Sensitive to outliers—if an extreme value exists, it skews the entire scaling.

7.3 Data Normalization Type: Min-Max Normalization

Min-Max Normalized Data (0 to 1 Range)

Original	Normalized
10	0.0
20	0.25
30	0.5
50	1.0

7.3 Data Normalization Type: Z-Score Normalization

- Z-Score Normalization is usually called as **standardisation**.
- Z-Score normalization performs normalization a based on mean and standard deviation of the data in such a way that the mean and standard deviation becomes 0 and 1 respectively.
- Best for datasets with outliers or when data is normally distributed.
- Z-Score Normalization Formula:

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

Where:

- X = Original value
- μ = Mean of the dataset
- σ = Standard deviation

7.3 Data Normalization Type: Z-Score Normalization

Mean = 27.5, Standard Deviation = 15.81

Original	Z-Score Normalized
10	-1.11
20	-0.47
30	0.16
50	1.42

7.3 Data Normalization: Numerical

1. An e-commerce company tracks daily sales (in USD) for a week:

   Sales=[500,750,1200,1800,2000]

   Perform Normalization to scale the sales data between 0 and 1.

2. A company records salaries (in $1000) of five employees:

   Salaries=[30,35,40,50,55]

   Standardise these salaries.

3. For the given dataset, perform feature scaling.

Employee ID	Experience (Years)	Salary ($1000)	Performance Score (out of 100)
E101	2	30	60
E102	5	35	70
E103	7	40	75
E104	10	50	85
E105	12	55	90

7.3 Data Normalization: Numerical

4. Here is the customer complaint record of a service company for twelve consecutive days, and answer the following questions using this data.

Day	1	2	3	4	s	6	7	8	9	10	11	12
No. of Complaints	22	12	60	57	30	32	39	14	42	13	23	16

Normalize Day#9 complaints using both min-max and z-score normalization methods.

8.1 Data Dimensionality

- Data dimensionality refers to the number of features or attributes (columns) in a dataset.
- Low-dimensional data has only a few features (e.g., height & weight).
- High-dimensional data has many features (e.g., medical records with hundreds of tests).

8.1 Curse of Dimensionality

- Curse of Dimensionality refers to the phenomenon where the efficiency and effectiveness of algorithms deteriorate as the dimensionality of the data increases exponentially.
- As the number of dimensions increases, several problems arise:
  - Increased Computational Cost: More dimensions require more computations.
  - Sparse Data: High-dimensional data becomes sparse, making clustering and classification difficult.
  - Overfitting: More features increase the risk of overfitting in machine learning models.

8.2 Dimension Reduction Techniques

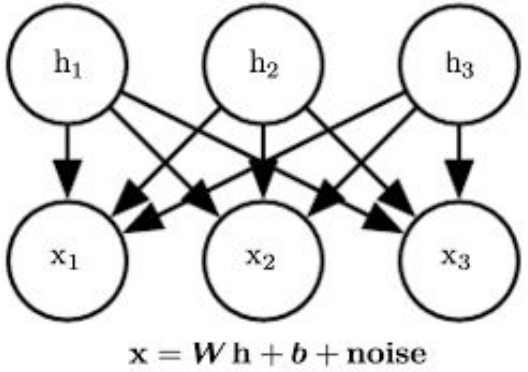
- To overcome the curse of dimensionality, we use Dimensionality Reduction techniques, which reduce the number of features while preserving essential information.
- Dimension Reduction Techniques

a. Feature Selection
   - Identify and select the most relevant features from the original dataset while discarding irrelevant or redundant ones.
   - This reduces the dimensionality of the data, simplifying the model and improving its efficiency.
   - Example: Filter Methods that uses statistical tests like correlation.

b. Feature Extraction
   - Transform the original high-dimensional data into a lower-dimensional space by creating new features that capture the essential information.
   - Techniques such as Principal Component Analysis (PCA) andLinear Discriminant Analysis (LDA) are commonly used for feature extraction.

8.3 Linear Factor Model (LFM)

- The Linear Factor Model (LFM) is a statistical model used to describe high-dimensional data using a smaller set of underlying factors.
- It assumes that observed variables are linear combinations of latent (hidden) factors plus some noise.
- The directed graphical model describes the linear factor model family, in which we assume that an observed data vector **x** is obtained by a linear combination of independent latent factors **h**, plus some noise.
- Different models, such as probabilistic PCA, factor analysis, make different choices about the form of the noise and of the prior p(h).



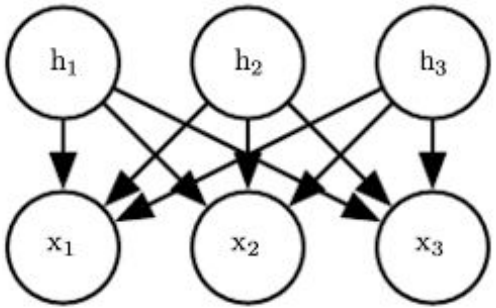
8.3 Mathematical Representation of LFM

- Let's assume we have p observed variables or features (x_1, x_2, ..., x_p) and k latent factors (h_1, h_2, ..., h_k), where k < p.
- The Linear Factor Model (LFM) assumes that observed data (X) can be represented as a linear combination of latent factors (h) plus some noise (ϵ).

$$X = Wh + b + \epsilon$$

Where:

- $X \rightarrow p \times 1$ vector of observed variables (**p-dimensional data**)
- $W \rightarrow p \times k$ factor loading matrix (**weights linking factors to observed variables**)
- $h \rightarrow k \times 1$ vector of latent factors (**hidden variables influencing the data**)
- $b \rightarrow p \times 1$ bias vector (**shifts the values of the observed variables**)
- $\epsilon \rightarrow p \times 1$ noise vector (**unexplained variance or residuals**)

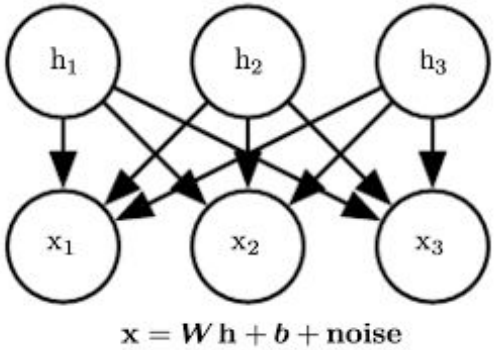


8.3 Mathematical Representation of LFM

$$X = Wh + b + \epsilon$$

Where:

- $X \rightarrow p \times 1$ vector of observed variables (**p-dimensional data**)
- $W \rightarrow p \times k$ factor loading matrix (**weights linking factors to observed variables**)
- $h \rightarrow k \times 1$ vector of latent factors (**hidden variables influencing the data**)
- $b \rightarrow p \times 1$ bias vector (**shifts the values of the observed variables**)
- $\epsilon \rightarrow p \times 1$ noise vector (**unexplained variance or residuals**)



If we have p observed variables and k latent factors, the equation expands as:

$$\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} & \dots & W_{1k} \\ W_{21} & W_{22} & \dots & W_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ W_{p1} & W_{p2} & \dots & W_{pk} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_k \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_p \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_p \end{bmatrix}$$

8.4 Principal Component Analysis (PCA)

- Principal Component Analysis (PCA) is a dimensionality reduction technique that transforms high-dimensional data into a lower-dimensional space while preserving as much variance as possible.
- It does this by transforming potentially correlated variables into a smaller set of variables, called principal components.
- Reduce the number of variables (features) while retaining the most important information.
- Applications of PCA:
  - Data compression
    - It helps create compact representations of feature, making them easier to store and transmit.
  - Feature extraction
    - It helps to extract significant features by retaining most important information.
  - Noise filtering
    - PCA can remove noise or redundant information from data by focusing on the principal components that capture the underlying patterns.
  - Visualization of high-dimensional data
    - PCA helps to visualize high-dimensional data by projecting it into a lower-dimensional space, such as a 2D or 3D plot.

8.4 Steps in Finding PCA

1. Standardize the Data
    - PCA works best when features have the same scale.
    - Features should reflect a normal distribution with a mean of zero and a standard deviation of one.
    - The standardization formula:

$$X_{\text{standardized}} = \frac{X - \mu}{\sigma}$$

Where:

- X is the original data
- μ is the mean of each feature
- σ is the standard deviation

8.4 Steps in Finding PCA

2. Compute the covariance matrix to identify correlations

- The covariance matrix captures relationships between features.
- Covariance (cov) measures how strongly correlated two or more variables are.
- The covariance matrix summarizes the covariances associated with all pair combinations of the initial variables in the dataset.
- Computing the covariance matrix helps identify the relationships between the variables–that is, how the variables vary from the mean with respect to each other.
- Covariance matrix is a symmetric matrix, meaning the variable combinations can be represented as d × d, where d is the number of dimensions. For example, for a 3-dimensional dataset, there would be 3 × 3 or 9 variable combinations in the covariance matrix.
- The sign of the variables in the matrix tells us whether combinations are correlated:
    - Positive (the variables are correlated and increase or decrease at the same time)
    - Negative (the variables are not correlated, meaning that one decreases while the other increases)
    - Zero (the variables are not related to each other)

8.4 Steps in Finding PCA

2. Compute the covariance matrix to identify correlations

- Covariance Formula:

For Population

$$Cov(x,y) = \frac{\Sigma (x_i - \bar{x}) * (y_i - \bar{y})}{N}$$

For Sample

$$Cov(x,y) = \frac{\Sigma (x_i - \bar{x}) * (y_i - \bar{y})}{(N-1)}$$

- Formula can be written in terms of matrix multiplication as:

$$C = \frac{1}{n-1} X^T X$$

Where:

- C is a $p \times p$ matrix
- X is the mean-centered data matrix
- n is the number of data points

	f1	f2	f3	f4
f1	var(f1)	cov(f1,f2)	cov(f1,f3)	cov(f1,f4)
f2	cov(f2,f1)	var(f2)	cov(f2,f3)	cov(f2,f4)
f3	cov(f3,f1)	cov(f3,f2)	var(f3)	cov(f3,f4)
f4	cov(f4,f1)	cov(f4,f2)	cov(f4,f3)	var(f4)

Covariance Matrix for 4 features

8.4 Steps in Finding PCA

3. Compute the eigenvectors and eigenvalues of the covariance matrix

- The eigenvectors are the principal components that represent the directions of maximum variance in the data.
- The eigenvalues represent the amount of variance in each component.
- Ranking the eigenvectors by eigenvalue identifies the order of principal components.

8.4 Steps in Finding PCA

3. Compute the eigenvectors and eigenvalues of the covariance matrix

- Eigenvector of a square matrix is defined as a non-vector in which when a given matrix is multiplied, it is equal to a scalar multiple of that vector.

$$CV = \lambda V$$

Where:

- C is the covariance matrix
- V are the **eigenvectors** (principal components)
- λ are the **eigenvalues** (variance explained by each component)

Solving,

$CV = \lambda V$

$CV - \lambda V = 0$

$(C - \lambda I)\, V = 0$ where I is the identity matrix of the same shape as matrix A.

And the above conditions will be true only if $(C–\lambda I)$ will be non-invertible (i.e. singular matrix). That means,

$$|C-\lambda I| = 0$$

This determinant equation is called the characteristic equation. Solving it gives eigenvalues and eigenvectors.

8.4 Steps in Finding PCA

4. Select the principal components

- Here, we decide which components to keep and those to discard.
- Components with low eigenvalues typically will not be as significant.
- Sort eigenvalues in descending order: The largest eigenvalue corresponds to the first principal component, which explains the most variance.
- Choose the top k components: The number of components k is selected based on the cumulative explained variance:

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{p} \lambda_i} \geq 95\%$$

This ensures at least **95% of the variance** is preserved.

8.4 Steps in Finding PCA

5. Transform Data to Lower Dimensions

- The new reduced dataset Z is obtained by multiplying the original data X by the selected principal components V_k:

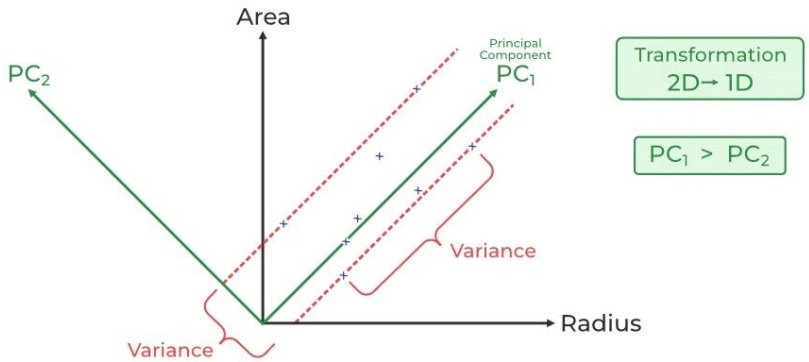
$$Z = XV_k$$

Where:

- Z is the transformed dataset with reduced dimensions
- V_k contains the top k eigenvectors

8.4 PCA Illustrated

- PC_1 (First Principal Component): The direction along which the data has the maximum variance. It captures the most important information.
- PC_2 (Second Principal Component): The direction orthogonal (perpendicular) to PC_1. It captures the remaining variance but is less significant.
- The variance along PC_1 is greater than PC_2, which means that PC_1 carries more useful information about the dataset.
- The data points (blue cross) are projected onto PC_1, effectively reducing the dataset from two dimensions (Radius, Area) to one dimension (PC_1).



8.4 Limitation of PCA

- Loss of Interpretability
  - PCA transforms original features into principal components, which are linear combinations of the original variables.
  - These new components lack physical meaning, making interpretation difficult in real-world applications.
- Sensitive to Scaling and Outliers
  - PCA is affected by differences in feature scales. If one feature has a larger range, it may dominate the principal components.
  - Outliers can distort the covariance matrix, leading to misleading results.
- Assumes Linearity Among Features
  - PCA assumes that the relationships between variables are linear, meaning it works best when data is normally distributed.
  - If data has nonlinear patterns, PCA may not capture important structures.
- Information Loss
  - PCA reduces dimensionality by removing low-variance components, but this may result in loss of important information.
  - If a low-variance feature is crucial for classification or prediction, PCA may degrade performance.
- Not Suitable for Categorical Data
  - PCA is designed for numerical data and does not handle categorical features well.

8.4 Numericals

1. Calculate covariance matrix for following sample data:

Student	Physics(X)	Biology(Y)
A	92	80
B	60	30
C	100	70

2. Compute Eigenvalues and Eigenvectors: $C = \begin{bmatrix} 0.62 & 0.59 \\ 0.59 & 0.63 \end{bmatrix}$

3. Reduce the following dataset to 1-D using PCA.

Features	E_1	E_2	E_3	E_4
X_1	2	3	4	5
X_2	4	6	8	10

PCA reference study material: link

8.4 Study Yourself

- Similarity and Differences between LFM and PCA.