Introduction to Data Science

Information is processed data e.g. employee database is just information but when we make a query we get some information.

Structured data has a well-defined format (like a table) e.g. relational databases, spreadsheets, csv files. Unstructured data lacks a specific format e.g. images, audio, video, text.

Data science is an interdisciplinary field that uses scientific methods, algorithms, processes, and systems to extract insights and knowledge from structured and unstructured data. It combines expertise from various domains such as statistics, computer science, domain knowledge, and data engineering to analyze, interpret, and leverage data for solving complex problems and making data-driven decisions.

The applications of data science are vast and continue to expand as the field evolves. Some common and emerging applications of data science include:

- Business Analytics: Data science is widely used in businesses to analyze customer behavior, optimize pricing, and improve marketing strategies. It helps in understanding market trends, customer preferences, and overall business performance.
- Recommendation Systems: Data science powers recommendation engines used in streaming services, e-commerce, and content platforms to suggest products, movies, music, and more to users.
- E-commerce: E-commerce companies use data science to recommend products to customers, optimize supply chain logistics, and enhance the shopping experience.
- Finance: Data science is applied in financial institutions for risk assessment, fraud detection, algorithmic trading, and credit scoring. It helps make investment decisions and manage portfolios.
- Manufacturing and Supply Chain: Data science is used to monitor and optimize production processes, predict equipment maintenance needs, and manage supply chains efficiently.
- Healthcare: Data science plays a crucial role in medical research, personalized medicine, disease prediction, and healthcare management. It is used to analyze patient records, predict disease outbreaks, and optimize healthcare delivery.
- Natural Language Processing (NLP): NLP techniques are employed in applications like sentiment analysis, chatbots, language translation, and speech recognition.
- Image and Video Analysis: Data science is used for image recognition, object detection, and video content analysis in various fields, including security, healthcare, and entertainment.
- Social Media Analysis: Social media platforms leverage data science to analyze user behavior, tailor content, and understand trends. This is valuable for advertising and public sentiment analysis.
- Environmental Science: Data science is applied to analyze climate data, predict natural disasters, and optimize resource management in agriculture and forestry.
- Energy and Utilities: Data science helps in optimizing energy consumption, monitoring the grid, and predicting equipment failures in the energy and utilities sector.
- Transportation and Logistics: In this field, data science is used for route optimization, demand forecasting, and real-time tracking of shipments.
- Government and Public Policy: Data science is used to analyze demographic data, crime statistics, and public health data for policy development and decision-making.
- Sports Analytics: Data science is employed in sports to analyze player performance, optimize training, and enhance fan engagement.

Data Science Life-cycle

1. **Understand the problem:** understanding what is to be done.
2. **Data collection:** acquiring and storing the data.
3. **Data preparation/pre-processing/wrangling/munging:** cleaning, transforming, etc.
4. **Data modelling:** application of ML algorithms.
5. **Model Evaluation:** evaluating and comparing results of ML algorithms.
6. **Model Deployment:** integrating the best model in the application.

The first step is to clearly define the problem or objective. What question are you trying to answer, or what goal are you trying to achieve through data analysis? For example, taking case study of sentiment analysis on some person who has been in news recently. So, we must understand what exactly is to be done.
Gather relevant data from various sources. This could include databases, APIs, web scraping, or other means. Ensuring data quality and reliability is crucial. In our case study, for sentiment analysis, data can be collected from Twitter.
Data preparation stage contains multiple steps like data cleaning, exploratory data analysis (EDA), feature engineering. Raw data often contains errors, missing values, and inconsistencies. Data cleaning involves handling these issues, such as imputing missing values, removing duplicates, and transforming data into a usable format. In EDA, we explore the data to gain insights and understand its characteristics. This step involves visualizations, statistical summaries, and identifying patterns or trends. In feature engineering, we create new features or transform existing ones to make the data more suitable for modeling. Feature engineering can improve the performance of machine learning algorithms.
In model building, we select and build machine learning or statistical models based on the problem and data. This may involve splitting the data into training and testing sets for model evaluation.
In model building, we assess the model's performance using relevant metrics (e.g., accuracy, precision, recall, F1-score for classification problems, or RMSE, MAE for regression problems). Fine-tune the model as needed.
Once a satisfactory model is built, it can be deployed in a real-world environment to make predictions or recommendations. Deployment can take various forms, from integrating it into a web application to automating regular reports.

Role of Domain Knowledge

Domain experts can collaborate with data scientists to work on data science projects. For example, a domain expert for credit card frauds can provide insights into the transactional data of credit cards and what features can be extracted to detect frauds.
Domain knowledge helps in:

- Problem formulation: domain knowledge helps data scientist go deep in understanding the problem and know exact specifications of what is to be done.
- Data Understanding & Preparation: domain knowledge also helps in understanding the dataset and how to pre-process data.
- Feature Engineering: the most important take away from domain knowledge is what features should be extracted from the given data which could help in developing a good model.
- Model Selection & Tuning: domain knowledge helps in identification of algorithms which can be used in modelling e.g. if project is related to anomaly detection then algorithms like isolation forest will give better results than random forest.
- Interpretation of Results: domain knowledge helps in identification of what criteria to use for evaluation of the model.

Data Wrangling/Munging/Pre-processing Life Cycle

- **Exploratory Data Analysis (EDA)** is an approach to analyzing data sets to summarize their main characteristics, often with the help of graphical representations. The primary goal of EDA is to understand the data, identify patterns, anomalies, and relationships within the data, and generate hypotheses that can guide further analysis or modelling.
    i. Data Summary: EDA begins with a summary of the key statistics and characteristics of the data. This includes measures like mean, median, variance, and standard deviation for numerical features, as well as frequency distributions and proportions for categorical features.
    ii. Data Visualization: Visualizing data is a fundamental aspect of EDA. Graphical representations, such as histograms, box plots, scatter plots, and bar charts, help to reveal patterns, outliers, and relationships in the data. Visualization allows data analysts to "see" the data and make initial inferences.
    iii. Multivariate Analysis: EDA explores relationships between pairs or groups of variables. This can include correlation analysis to understand the strength and direction of relationships between numerical variables, as well as cross-tabulations and contingency tables for categorical variables.
    iv. Anomaly Detection: Identifying unusual data points or outliers is an important aspect of EDA. Outliers can provide valuable insights or indicate data quality issues.

- **Data Cleaning**

i. Handling missing data: Decide whether to impute missing values or remove rows/columns with missing data. **Data Imputation** is filling missing values using techniques like mean, median, mode, or more advanced imputation methods, such as k-nearest neighbors (KNN) or regression-based imputation.
ii. Handling duplicate data: Identify and remove duplicate records if necessary.
iii. Outlier detection and treatment: Identify and handle outliers that might skew analysis.

- **Data Transformation**

i. Encoding categorical variables: Convert categorical variables into numerical representations, e.g., **one-hot encoding** or label encoding.
ii. Binning: Group continuous data into bins or categories.
iii. Feature engineering: Create new features by combining or transforming existing ones.
iv. Normalization/Scaling: Scale numerical features to a specific range, typically between 0 and 1 (min-max normalization).
v. Standardization: Transform numerical features to have a mean of 0 and a standard deviation of 1 (z-score standardization).

Normalization/Standardization is important as if features have significantly different scales, some may dominate the optimization process, resulting in non-optimal predictions.

- **Data Reduction**

i. Dimensionality reduction: Use techniques like Principal Component Analysis (PCA) or feature selection to reduce the number of features while preserving important information.
ii. Sampling: Down-sample or up-sample data to handle class imbalances in classification tasks.

- **Data Integration**

Combine data from multiple sources and merge datasets to create a unified dataset for analysis.
Resolve inconsistencies in data representations, such as units of measurement or data formats.

- **Data Splitting**

Split the dataset into training, validation, and test sets for model development and evaluation.

Dealing with Unstructured Data

Processing unstructured data comes with several challenges due to its lack of a predefined structure and the diversity of data types. Here are some common challenges in processing unstructured data:

- Lack of Structure: Unstructured data lacks a predefined structure, making it difficult to organize, query, and analyze compared to structured data.
- Data Heterogeneity: Unstructured data can take various forms, including text, images, audio, video, and sensor data. Each type of data requires different processing techniques and tools.
- Data Quality: Unstructured data may contain noise, errors, and inconsistencies, which can hinder analysis. Cleaning and preprocessing are often required to improve data quality.
- Context and Semantics: Understanding the context and semantics of unstructured data can be challenging. For example, interpreting the meaning of words in a text document or recognizing objects in an image.
- Data Integration: Combining unstructured data with structured data for comprehensive analysis can be complex, requiring data integration and transformation.
- Data Privacy and Security: Unstructured data often contains sensitive information, such as personal data in text or images. Ensuring data privacy and security is crucial.
- Data Labeling and Annotation: Preparing unstructured data for supervised machine learning often involves manual labeling and annotation, which can be time-consuming and subjective.
- Data Variability: Unstructured data often exhibits high variability, with different writing styles, languages, accents, or lighting conditions in the case of audio and visual data.

Here are some common techniques to process unstructured data:

Text Data Processing

- Text Tokenization: Divide text into smaller units such as words or sentences. Tokenization is the first step for many natural language processing (NLP) tasks.
- Text Cleaning: Remove noise, such as HTML tags, punctuation, and special characters. This is essential for improving data quality.
- Stop Word Removal: Eliminate common words (e.g., "and," "the," "is") that may not carry significant meaning for analysis.
- Stemming and Lemmatization: Reduce words to their base or root form to handle variations of words (e.g., "running" -> "run").
- Named Entity Recognition (NER): Identify and extract named entities like names, locations, organizations, and dates from text.
- Text Vectorization: Convert text data into numerical vectors using techniques like Bag of Words (BoW) or Term Frequency-Inverse Document Frequency (TF-IDF) for machine learning applications.

Image Data Processing

- Image Pre-processing: Techniques include resizing, cropping, and normalization to prepare images for analysis.
- Feature Extraction: Extract relevant features from images, such as edges, colors, textures, or object keypoints.

Audio Preprocessing: Extract features like spectrograms, Mel-frequency cepstral coefficients (MFCCs), and chroma features from audio signals.

Video Preprocessing: Extract frames or keyframes from videos to reduce dimensionality.

Skills needed to become a data scientist

Becoming a data scientist requires a diverse set of skills that encompass technical, analytical, and domain-specific knowledge. Here is a list of key skills needed to become a successful data scientist:

- Programming Skills

Proficiency in programming languages commonly used in data science, such as Python and R.

- Data Manipulation and Cleaning

Knowledge of data manipulation libraries like Pandas and NumPy.

- Statistical Analysis

Strong understanding of statistical concepts and techniques, including hypothesis testing and regression analysis.

- Machine Learning

Proficiency in machine learning algorithms for classification, regression, clustering, and recommendation. Experience with machine learning libraries such as scikit-learn and TensorFlow.

- Data Visualization

Skill in creating effective data visualizations using tools like Matplotlib, Seaborn, or Tableau.

- Big Data Technologies

Familiarity with big data tools and frameworks, including Hadoop, Spark, and SQL for managing and processing large datasets.

- Database Management

Knowledge of working with databases and querying data using SQL. Experience with both relational and NoSQL databases.

- Feature Engineering

Ability to engineer and select relevant features for machine learning models.

- Time Series Analysis

Skills in analyzing time-dependent data, including forecasting, trend analysis, and seasonality detection.

- Natural Language Processing (NLP)

Understanding of NLP techniques for text analysis, sentiment analysis, and text classification.

- A/B Testing

Experience in designing and analyzing A/B tests to measure the impact of changes in web applications or marketing strategies.

- Domain Knowledge

Familiarity with the domain or industry you work in to understand the data's context and business objectives.

- Data Ethics and Privacy

Awareness of ethical considerations in data science and compliance with data privacy regulations.

- Communication Skills

Effective communication of complex findings and insights to both technical and non-technical stakeholders.

- Project Management

Ability to manage data science projects from problem definition to deployment, including setting goals, timelines, and deliverables.

Traits of a data scientist

- Problem-Solving Abilities: Data scientists excel at framing and solving problems. They can break down complex issues into manageable components and use data-driven approaches to find solutions.
- Persistence: Data science projects can be challenging and time-consuming. A data scientist's determination and patience are valuable traits for tackling complex problems.
- Curiosity: Data scientists have a natural curiosity and an eagerness to explore data and discover hidden insights. They continually seek to improve their knowledge and skills.
- Adaptability: The field of data science is constantly evolving, and data scientists need to stay up to date with new tools, techniques, and technologies.
- Communication Skills: Effective communication is crucial in conveying findings and insights to a wider audience. Data scientists should be able to explain complex concepts in a clear and understandable manner.
- Collaboration: Data scientists often work in cross-functional teams with data engineers, analysts, and business stakeholders. They should be able to collaborate and align their work with others' goals and requirements.
- Ethical and Responsible Data Handling: Data scientists must be aware of ethical considerations when dealing with sensitive or private data and adhere to data privacy regulations and best practices.
- Business Acumen: Understanding the business context and objectives is essential for data scientists to provide insights that align with the organization's goals and strategies.

Roles and Responsibilities of a data scientist

- Data Collection and Cleaning

Collect, scrape, or access relevant data from various sources, including databases, APIs, and external datasets. Clean, pre-process, and wrangle data to remove inconsistencies, missing values, and outliers.

- Data Exploration and Visualization

Perform exploratory data analysis (EDA) to understand the structure and characteristics of the data. Create visualizations to convey insights and identify patterns in the data.

- Feature Engineering

Engineer and select relevant features for use in predictive models. Transform and pre-process features to improve model performance.

- Statistical Analysis

Apply statistical techniques to validate hypotheses and make data-driven decisions. Perform hypothesis testing, A/B testing, and regression analysis.

- Machine Learning Model Development

Build and train machine learning models for various tasks, such as classification, regression, clustering, and recommendation. Optimize model performance through hyperparameter tuning and model selection.

- Model Evaluation and Validation

Assess model performance using metrics like accuracy, precision, recall, F1-score, and cross-validation techniques. Ensure models are robust and reliable.

- Model Deployment

Deploy machine learning models into production systems for real-time predictions.

Data Scientist vs Data Engineer vs Data Analyst

Data scientists are primarily responsible for extracting insights and knowledge from data. They use a combination of statistical, programming, and domain expertise to analyze data and provide actionable insights. Their tasks often include data cleaning, exploratory data analysis, modeling, and developing predictive algorithms or machine learning models. Data scientists are expected to have strong programming skills, expertise in statistical analysis, and the ability to work with large datasets. They often use tools like Python, R, and machine learning libraries.

Data engineers focus on the design, construction, installation, and maintenance of data pipelines and architectures. They are responsible for ensuring that data is collected, stored, and made accessible for analysis. Their tasks include database design, data ETL (Extract, Transform, Load) processes, data warehousing, and building data infrastructure. Data engineers need strong knowledge of database systems, data integration tools, and programming languages like SQL, as well as data storage and data processing technologies like Hadoop and Spark.

Data analysts are responsible for interpreting and visualizing data to help organizations make informed decisions. They focus on generating reports, creating visualizations, and communicating findings. Their tasks involve querying databases, creating dashboards, and conducting exploratory data analysis to answer specific business questions. Data analysts need strong analytical and communication skills, and they often use tools like Excel, Tableau, Power BI, and SQL for their work.

Data Analysis Process

1. Identify business questions: we need to what is the aim of analysis process and what are the questions whose answers we are looking for?
2. Collect and Store data: Sources include case studies, surveys, interviews, questionnaires, direct observation, and focus groups.
3. Clean & Prepare data: remove white spaces, duplicate records, and basic errors.
4. Analyze data: use tools like PowerBI, Tableau, etc. to analyse data.
5. Visualization: Data visualization is a fancy way of saying, “graphically show your information in a way that people can read and understand it.” You can use charts, graphs, maps, bullet points, or a host of other methods. Visualization helps you derive valuable insights by helping you compare datasets and observe relationships.

Python

Python developed by Guido van Rossum and first released in 1991. It is a case-sensitive and indentation-sensitive language. Some of the important libraries for data science are numpy, pandas, matplotlib, seaborn, and sklearn.

Python holds significant importance in the field of data science due to its versatility, powerful libraries, and a vast ecosystem of tools and resources. Here are several reasons for the significance of Python in data science:

- Ease of Learning and Readability: Python is known for its clean and readable syntax, making it accessible for individuals new to programming. This reduces the barrier to entry for those seeking to enter the field of data science.
- Extensive Data Science Libraries: NumPy, Pandas, Matplotlib, Seaborn, Scikit-Learn, etc.
- Rich Ecosystem: Python's extensive ecosystem includes libraries and packages for various data-related tasks, from web scraping (Beautiful Soup, Scrapy) to natural language processing (NLTK, spaCy), and more. This abundance of resources allows data scientists to tackle diverse projects efficiently.
- Versatility: Python is not limited to data science but is a general-purpose programming language. This means data scientists can use Python to address a wide range of tasks beyond data analysis, such as web development, scripting, and automation.
- Scalability: Python is often the first choice for prototyping and developing data science solutions, and many data science projects can transition from prototype to production without major code changes.
- Open Source: Python is open source, which means it is free to use and continuously developed and improved by the community.
- Community Support: Python has a vast and active community of developers, data scientists, and enthusiasts. This community support ensures that there is a wealth of documentation, tutorials, and open-source projects to aid data scientists.
- Cross-Platform Compatibility: Python is compatible with multiple operating systems, allowing data scientists to work seamlessly across different platforms.
- Integration and Compatibility: Python can easily integrate with other programming languages (e.g., C, C++, Java) and software, making it an ideal choice for data scientists who need to work with diverse tools and systems.
- Big Data and Cloud Computing: Python libraries like PySpark and tools such as Dask enable data scientists to work with big data and leverage cloud platforms for scalable processing.

```
# Output
print(“hello world”)


# Declaring a Variable
a=5                                  # int
b=7.8                                # float
c= “hello”                           # string
d=True                               # bool
p = [“apple”, “mango”, “banana”]     # list (mutable)
q = (“apple”, “mango”, “banana”)     # tuple (immutable)
r = {“apple”, “mango”, “banana”}     # set (unique elements)
s = {“name”:John, “age”:16}          # dictionary (key:value pairs)
print(p)                             # display value of variable
print(type(p))                       #display type of variable


# Operators

x=5
y=2
print("x+y=",x+y)
print("x-y=",x-y)
print("x*y=",x*y)
print("x/y=",x/y)
print("Floor division: x//y=",x//y)
print("Exponential, x**y=",x**y)
print("Remainder, x%y=",x%y)


Comparison operators:  ==,  !=,  >,  <,  >=,  >=
Logical operators:  and,  or,  not
Membership operators: in, not in
Identity operators:  is,  is not
Bitwise operators: & (AND), | (OR), ~ (NOT), ^ (XOR), << (left shift), >> (right shift)
Assignment operators: =, +=, -+, *=, /=, //=, %=, **=, &=, |=, ^=, >>=, <<=

print(3 << 2) returns 12 as 0000000000000011 becomes 0000000000001100
NOTE: negative number are stored in 2s complement form.
```

```
# Looping
for x in range(5):
    print(x)
OUTPUT
0
1
2
3
4

for x in range(5):
    print(x, end=" ")
OUTPUT
0 1 2 3 4

for x in range(10,0,-2):
    print(x)
OUTPUT
10
8
6
4
2

arr = [2,3,5,7,9]
for x in arr:
    print(x)
OUTPUT
2
3
5
7
9



# Conditional Statements
marks = 85
if marks<40 and marks>=0:
    print("fail")
elif marks>40 and marks<=100:
    print("pass")
else:
    print("Incorrect Input")

# Function
def display():
    print(x)
display("hello")
```

```
# User Input
num = int(input(“Enter a number:  “))
print(num)

# Exception Handling
x = 5
y = "hello"
try:
    z = x + y
except:
    print("Error: cannot add an int and a str")
```

Importing dataset

Step-1: Install pandas: **pip install pandas**
Step-2: import pandas library: **import pandas as pd**
Step-3: Identify the location of file and type of file

```
df1 = pd.read_csv(“data.csv”)
df2 = pd.read_excel(“data.xlsx”, sheet_name=“Sheet1”)
df3 = pd.read_json(“data.json”)
```

Step-4: Display contents: `df1.head(5)` # this will display first 5 rows, by default 10

Exporting data

```
df.to_csv(“filename.csv”, encoding=“utf-8”)
```

Basic Insights from dataset

```
df.shape        # this will return dimensions of the dataset
df.info()       # this will display column names and their datatypes
df.summary()    #this will give statistics of each column like min, max, mean, std
```

Identify missing values

```
df.isnull().sum()
```

Handle missing value / Data Imputation

```
df.fillna(0)
mean_age = df['age'].mean()
df['age'].fillna(mean_age, inplace=True)

# Visualize categorical data
import matplotlib.pyplot as plt
plt.bar(data['day'], data['tip'])
plt.xlabel('Day')
plt.ylabel('Tip')
plt.show()

# Visualize distribution of a numeric feature in a dataset
import seabron as sns
sns.kdeplot(data=df, x=“total_bill”)
```

Synthetic datasets

A synthetic dataset is a dataset that is artificially generated rather than collected from real-world observations or measurements. Synthetic datasets are created to mimic certain characteristics or patterns of real data but are entirely fabricated. They are used in various fields, including machine learning, data analysis, and computer science, for various purposes such as testing algorithms, model validation, and experimentation.

Suppose you are developing a machine learning model to detect fraudulent credit card transactions. You need a dataset for training and testing your model. You can create a synthetic dataset by defining the characteristics of legitimate transactions and fraudulent transactions based on your knowledge of typical patterns and behaviors. This synthetic dataset can be used to validate and fine-tune your fraud detection model.

A **delimiter** is the symbol which separates columns in the file e.g. comma is the delimiter in csv file.

To create synthetic datatset in Excel, we can use the following functions to fill random values:
=RANDBETWEEN(1,1000)
=RANDBETWEEN(DATE(2020,1,1),DATE(2020,12,31)
=NORMINV(RAND(),0,1) // Random Normal distribution with mean=0, SD=1
To create a column in Excel with categorical data, we can use VLOOKUP. We first create a table with 2 columns, first having index, second with categories (say 10 categories), then we "Define Name" of this table e.g. products. Next we use this table to populate our dataset column with
=VLOOKUP(RANDBETWEEN(1,10),products,2) // use 2nd column of products table

To create synthetic datasets in Python, you can use libraries like faker, random, SDV (synthetic data vault), mimesis, etc.

```
from faker import Faker
fake = Faker()
print (fake.email())
print(fake.country())
print(fake.name())
print(fake.text())
print(fake.latitude(), fake.longitude())
print(fake.url())
```

```
import numpy as np
num_samples = 100       # Define the number of data points
# Create synthetic features
feature1 = np.random.randn(num_samples)          # Random data from a normal distribution
feature2 = np.random.uniform(0, 1, num_samples)  # Random data from a uniform distribution
# Combine features into a synthetic dataset
synthetic_data = np.column_stack((feature1, feature2))
print(synthetic_data)
```

Case Study : Consider a fictional e-commerce company looking to improve its customer retention.

Step 1: Problem Definition and Data Understanding
The e-commerce company wants to reduce customer churn (the rate at which customers stop buying from them). They have access to historical customer data, including purchase history, demographics, customer support interactions, and more.

Step 2: Data Collection
The company collects data from various sources, including their customer database, transaction records, website analytics, and customer surveys.

Step 3: Data Cleaning and Pre-processing
They clean the data, addressing issues such as missing values, duplicate records, and outliers. They also standardize and format the data.

Step 4: Exploratory Data Analysis (EDA)
They perform EDA to understand the data better. EDA helps identify customer characteristics or behaviors that may be related to churn.

Step 5: Feature Engineering
They engineer new features like customer lifetime value, purchase frequency, and customer segment based on EDA insights.

Step 6: Data Splitting
They split the data into training, validation, and test sets for model development and evaluation.

Step 7: Model Development
They build machine learning models to predict customer churn based on the features they've engineered. They try different models, such as logistic regression, random forest, and neural networks. They train the models on the training data and fine-tune hyperparameters.

Step 8: Model Evaluation and Validation
They evaluate model performance using metrics like accuracy, precision, recall, and F1 score.
They validate the models using the validation dataset to ensure they generalize well.

Step 9: Model Deployment
Once satisfied with a model's performance, they deploy it into their systems for real-time predictions on customer data.

Case Study: Data Cleaning for a Customer Database

Step 1: Problem Definition and Data Understanding
Problem: A retail company has a customer database with sales records, but the data is messy and contains errors.
Data: The database includes customer information, such as names, addresses, email addresses, and purchase history.

Step 2: Data Collection
The company collects data from various sources, including online and in-store transactions, loyalty program registrations, and email sign-ups.

Step 3: Data Cleaning and Pre-processing

- Identifying Missing Values

The data team begins by identifying missing values, such as incomplete email addresses, phone numbers, and addresses.

- Handling Missing Values:

For missing email addresses, they decide to use a default value like "unknown@example.com" to ensure consistency. For incomplete addresses, they implement address validation and correction algorithms to standardize addresses. Missing phone numbers are marked as "Not Provided."

- Dealing with Duplicate Records

The team identifies and removes duplicate customer records to avoid double-counting.

- Outlier Detection

They detect outliers in purchase history and validate them for accuracy. Suspicious or erroneous records are flagged for further review.

- Data Format Standardization

The team ensures consistent formatting of customer names, such as capitalizing the first letter of each name.

- Handling Data Entry Errors

Data entry errors like typos in names or addresses are corrected manually or using data validation rules.

- Encoding Categorical Data

Categorical variables like gender or customer type are encoded for analysis.

Step 4: Exploratory Data Analysis (EDA)
After data cleaning, the team performs EDA to understand the characteristics of their clean data. This includes summarizing key statistics, creating visualizations, and identifying patterns in customer behavior.

Step 5: Data Validation and Documentation
The team validates the cleaned dataset to ensure data quality. They document the data cleaning process, including the rules and decisions made, for transparency and future reference.

Step 6: Data Export
The cleaned and pre-processed data is exported for further analysis, such as customer segmentation, sales forecasting, and targeted marketing.