Unit-1
Fundamentals of AI and Machine Learning

Contents

1.1 Concept of Machine Learning

1.1.1 Understanding Machine Learning:

What is Machine Learning?

Machine Learning (ML) is a field in computer science where computers learn from data and make decisions or predictions — without being directly programmed for every task.

Instead of writing fixed instructions, we provide examples (called data) and let the machine find patterns on its own.

EXAMPLE:

Think of a child learning to identify animals. First, we show the child pictures of cats and dogs. Then we tell the child, “This is a cat” or “This is a dog". After seeing many examples, the child starts identifying them correctly, even if it's a new picture.

Machine learning does the same!

We feed a lot of data (examples) into the computer. The computer studies the patterns and then makes predictions on new data.

Real Example:

- Email spam filter: Learns which emails are spam based on previous examples.
- Netflix suggestions: Learns your taste and suggests shows.

1.1.2 Benefits of Machine Learning

Automation – Computers can do tasks without human help. Ex:

Self-driving cars learn how to drive.

Faster and Smarter Decisions – It can make quick decisions from data. Ex:

Credit card fraud detection.

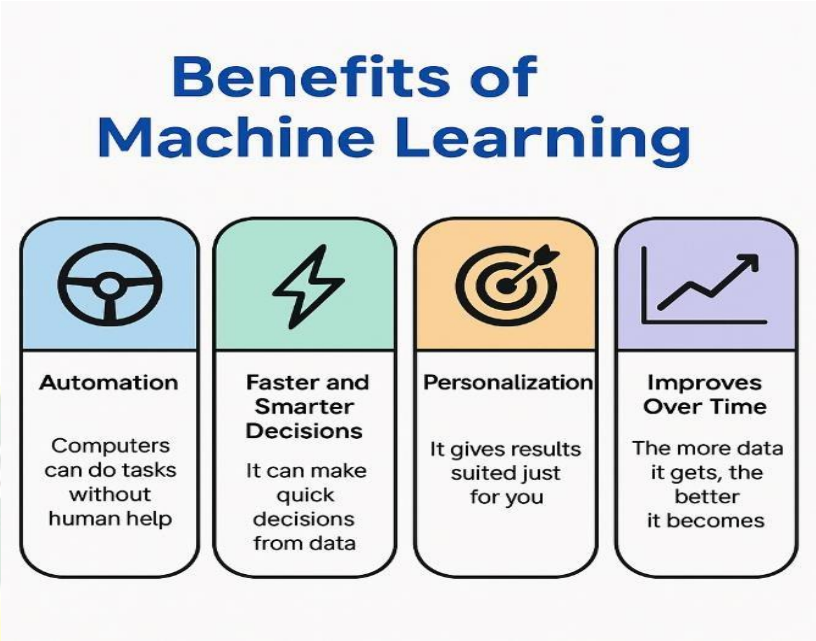
Personalization – It gives results suited just for you.

Ex: YouTube recommends videos based on your interests.

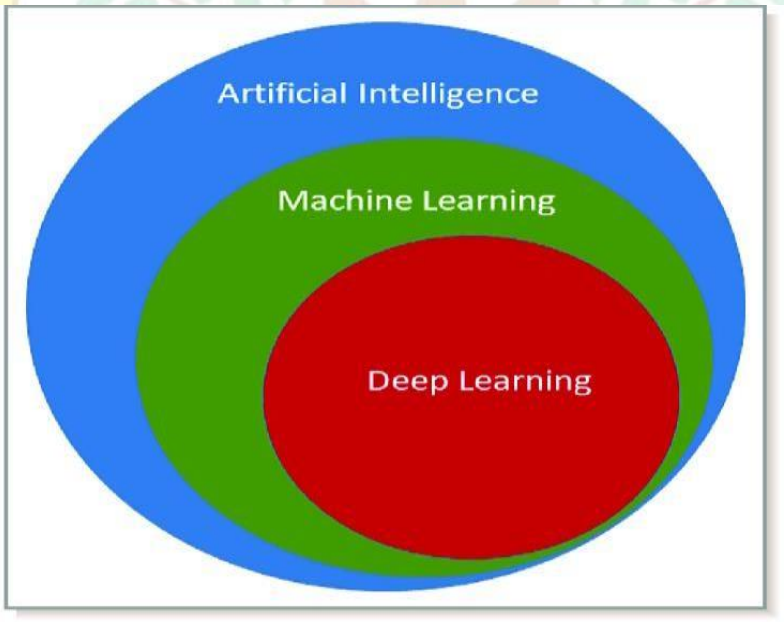
Improves Over Time – The more data it gets, the better it becomes. Ex:

Voice assistants like Alexa learn your voice over time.

Solves Complex Problems – Like medical diagnosis or language translation.



1.1.3 Difference between AI, ML, and Deep Learning



What is Artificial Intelligence (AI)?

- AI is the goal — It's the final product: making a machine or system think and act smart like a human.
- AI is about giving intelligence to computers so they can solve problems, understand, learn, and respond — just like humans.
- AI includes everything from if-else rules, machine learning, robotics, language

understanding, and more.

Analogy:
Think of AI as a self-driving car — that's the smart final product**.**

What is Machine Learning (ML)?

- ML is one of the main ways to reach AI.
- ML focuses on structured data (neatly arranged data in tables: rows & columns).
- In ML, we train the computer with data and let it find patterns so it can make decisions on its own.

Analogy: ML as a Student Learning from a Marks Sheet

Imagine a teacher giving a student a table (structured data) with the following columns:

Hours Studied	Attendance (%)	Assignment Score	Final Exam Score
5	90	8	70
10	95	9	85
2	70	5	50

The student looks at many such rows and starts finding a pattern:

"If someone studies more, attends class regularly, and scores well in assignments, then their final exam score is likely to be higher."

That student can now predict what a new student might score in the final exam based on similar values.

What This Means in ML:

- The **table** is structured data (rows and columns).
- The ML model **learns patterns** from this data.
- Then it can **predict or decide** based on new input.

What is Deep Learning (DL)?

DL is a **special part of ML** that uses structures called **neural networks** — inspired by the human brain.

- **DL is used when the data is unstructured**, like:
    - Images
    - Videos
    - Audio
    - Natural language (text, speech)
- DL models need **lots of data** and **powerful computers**, but they can handle very complex tasks.

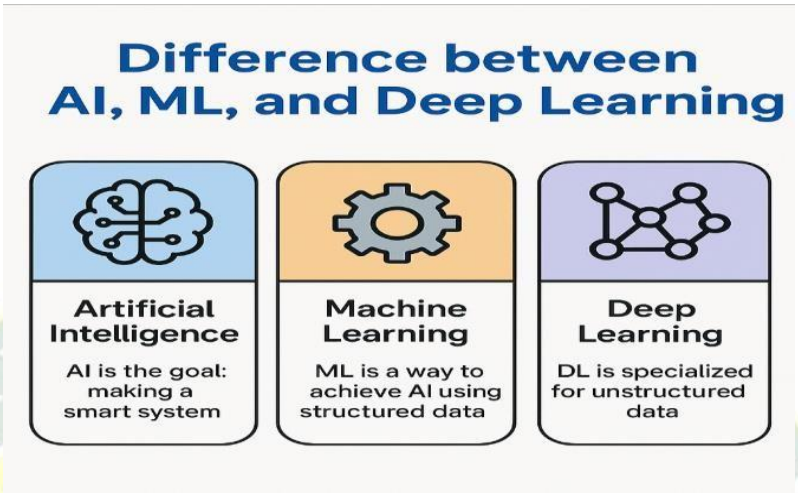
Analogy:
DL is like giving the self-driving car **eyes and ears** — so it can **see roads (images), listen to commands (voice), and respond smartly** without us giving it structured inputs.

Structured vs. Unstructured Data

Type of Learning	Works Best With	Examples
ML	Structured Data (organized like Excel sheets or databases)	Sales data, student scores
DL	Unstructured Data (messy, natural formats)	Photos, videos, emails, voice

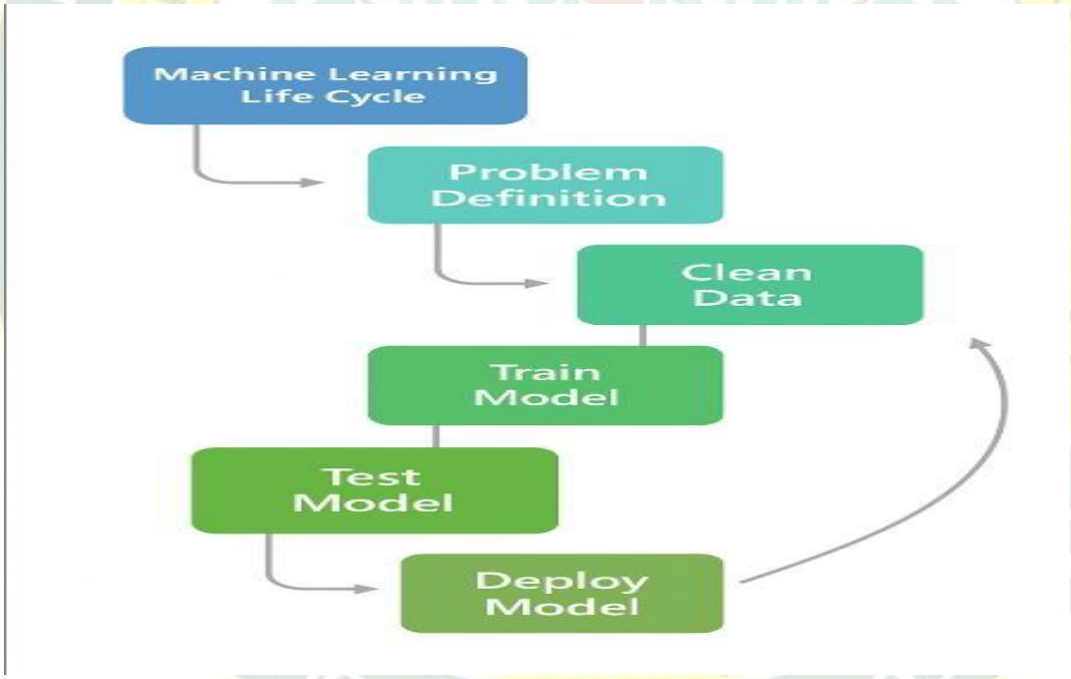
Main difference:

Concept	What it is	Analogy
AI	Final goal: Smart system	A full smart kitchen that cooks on its own
ML	A way to build AI using data	A cooking assistant that learns your favorite recipes from structured steps
DL	Advanced ML for unstructured data	A robot chef that watches you cook and learns everything, even how you move and talk



1.2 Machine learning Life Cycle

1.2.1 Stages of ML Life Cycle



The Machine Learning (ML) Life Cycle refers to the **step-by-step process** of building, training, testing, and deploying an ML model to solve a real-world problem using data.

1. Problem Definition

What it means:
Define clearly what problem you are trying to solve with ML. Without a clear goal, the project will fail.

Examples:

- Predict the price of a house
- Identify whether an email is spam
- Recommend products to users

Analogy:
It's like a student deciding which subject to study. The goal must be clearly understood before beginning the process.

2. Data Collection

What it means:
Gather the data that is relevant to the problem. This can be done using databases, APIs, surveys, or logs.

Examples:

- Customer demographics and purchase history
- Historical stock prices
- Patient health records

Analogy:
Just like a student collects books and notes to prepare for an exam, ML collects data as its learning material.

3. Data Cleaning / Preprocessing

What it means:
Raw data usually has missing values, errors, duplicates, or inconsistent formats. Cleaning it makes the data usable.

Tasks include:

- Removing null or duplicate entries
- Filling missing values with average or most frequent values
- Converting categorical data into numbers

Analogy:
Like organizing and rewriting messy notes before studying.

4. Feature Selection and Engineering

What it means:
Choose the most relevant variables (features) for the model and sometimes create new ones that help improve model performance.

Examples:

- Selecting “age” and “income” for a loan model
- Creating “total_marks” from multiple subject scores

Analogy:

Similar to summarizing your syllabus into key points so you can study better.5

5. Model Training

What it means:

Feed the cleaned and prepared data into a chosen ML algorithm so it can learn from the patterns in the data.

Examples of algorithms:

- Linear Regression
- Decision Trees
- Support Vector Machines

Analogy:

Just like a student study from solved papers and textbooks to prepare for the exam.

6. Model Testing / Evaluation

What it means:

Check how well the model performs using new data that it has never seen before.

Common evaluation metrics:

- Accuracy
- Precision and Recall
- Root Mean Square Error (RMSE)

Analogy:

Like taking a practice test to evaluate how well you’ve studied.

7. Model Deployment

What it means:

Once the model performs well, it is deployed into a real-world environment so that users or systems can use it.

Examples:

- A chatbot answering customer queries
- A recommendation engine on an e-commerce site

Analogy:

Like taking the final exam or presenting your project to a panel.

8. Monitoring and Maintenance

What it means:
Even after deployment, the model needs to be checked regularly. If it becomes less accurate due to changing data, it must be retrained or updated.

Examples:

- Updating the model with new customer data
- Fixing bugs or improving performance

Analogy:
Like continuing to revise or practice a subject after the exam is over to stay up to date.

1.2.2 Problem Definition to Model Monitoring

Stage	Description	Analogy
Problem Definition	Define what to solve	Choose your subject to study
Data Collection	Gather related data	Collect books and notes
Data Cleaning	Fix and format the data	Clean and organize your notes
Feature Engineering	Choose or create useful features	Summarize key study topics
Model Training	Teach the model using data	Study from books and examples
Model Testing	Evaluate with new data	Take a practice test
Model Deployment	Put the model into real use	Present the final answer or appear in exam
Monitoring	Keep checking and updating the model	Continue practicing and revising

1.3 Types of Exploratory Data Analysis (EDA)

What is EDA? (Exploratory Data Analysis)

Exploratory Data Analysis, EDA, is the **first step in understanding a dataset**.

Before we build any machine learning model, we must explore the data to:

- Understand what's in it
- Find patterns or trends
- Spot mistakes, missing values, or unusual data (outliers)

Think of it like checking ingredients before cooking a dish — you want to know

what you have, what's good, and what needs to be cleaned or removed.

1.3.1 Types of Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) can be divided into three main types based on the number of variables being analyzed:

- Univariate Analysis
- Bivariate Analysis
- Multivariate Analysis

Univariate analysis:

focuses on analyzing a single variable at a time. It helps us understand the basic properties of that variable such as its distribution, central values (like mean and median), and spread (like range or standard deviation). For example, if we want to explore the marks scored by students in a class, we will look at one column (marks) and use tools like histograms or box plots to understand how those marks are distributed.

Bivariate analysis:

involves the study of two variables to explore the relationship or correlation between them. It helps us find whether two variables move together (positive or negative correlation), or how one affects the other. For instance, we might analyze the relationship between "hours studied" and "marks scored" to see if studying more results in higher marks. Common tools for bivariate analysis include scatter plots, correlation coefficients, and line graphs.

Multivariate analysis:

deals with three or more variables at the same time. It is used when we want to explore more complex patterns and interactions within the data. For example, we might study how a student's marks are affected by multiple factors like study hours, sleep time, and attendance. This kind of analysis is especially useful in machine learning when many input features are used to predict an output. Visualization tools such as heatmaps, pair plots, and 3D scatter plots are often used in multivariate analysis.

In summary, univariate analysis gives us information about each individual variable, bivariate analysis reveals how two variables relate to each other, and multivariate analysis helps us understand relationships and patterns among several variables at once.

1.3.2 Univariate Analysis

What is Univariate Analysis?

Univariate analysis means analyzing **one variable at a time**.

The term "uni" means **one**, and "variate" refers to a **variable** (or a column in a dataset). So, univariate analysis is all about **understanding the properties and patterns of a single variable** without involving any other variable.

It helps us answer questions like:

- What kind of values does the variable take?
- What is the average value?
- Is the data spread out or tightly packed?
- Are there any very high or very low values (outliers)?

There are two types of variables in univariate analysis:

1. **Numerical (continuous or discrete)** – like marks, height, age, salary
2. **Categorical(non Numerical)** – like gender, city, product category

Depending on the type of variable, we use different techniques:

- **For numerical variables**, we calculate:
  - Mean (average)
  - Median (middle value)
  - Mode (most frequent value)
  - Range, Variance, Standard Deviation
  - We also plot **histograms** and **box plots**
- **For categorical variables**, we look at:
  - Frequency count (how many times each category appears)
  - Percentages or proportions
  - We use **bar charts** and **pie charts** to visualize

In short, univariate analysis gives us a **basic understanding of one column** of data. It is often the first step before analyzing relationships between variables (which is done in bivariate or multivariate analysis).

Univariate Analysis Example

Let's say we are working with a dataset named "Students Performance in Exams" that contains student information such as:

- Gender
- Math score
- Reading score
- Writing score

Now, let's perform univariate analysis on one variable:

Example 1: Numerical Variable – Math Score We

want to analyze the column **"math score"**. **We can**

ask:

What is the average math score?

What is the highest and lowest score?

Are most students scoring between 60 and 80?

Are there any students scoring very low or very high?

We can use:

- **Descriptive statistics** like mean, median, max, min
- **Histogram** to see the distribution
- **Box plot** to detect outliers

Code:

```
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv("StudentsPerformance.csv")
print(df['math score'].describe())
plt.hist(df['math score'],bins=10, edgecolor='black')
plt.show()
```

Example 2: Categorical Variable(Non Numerical) – Gender

Now we analyze the column **"gender"**.

We can ask:

- How many male and female students are there?
- Which gender is more in the class?

We can use:

- **Value counts** to get the frequency
- **Bar chart** to visualize

Code:

```
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv("StudentsPerformance.csv")
print(df['gender'].value_counts())
data=df['gender'].value_counts()
plt.bar(data,height=20)
plt.show()
```

Bivariate and Multivariate Analysis - Extended

In data analysis, understanding how variables relate to each other is crucial. Bivariate and multivariate analyses help us
explore relationships between two or more variables in a dataset. This document uses the Student Performance dataset, which

includes demographic and academic performance data for students, to demonstrate these techniques.

The dataset includes the following columns:
- gender: Student's gender (male/female)
- race/ethnicity: Student's racial or ethnic group
- parental level of education: Education level of a parent
- lunch: Type of lunch received (standard or free/reduced)
- test preparation course: Completion status of a test prep course
- math score: Student's score in mathematics
- reading score: Student's score in reading
- writing score: Student's score in writing

1.3.3 Bivariate Analysis

Bivariate analysis investigates the relationship between two variables. It is used to identify patterns, correlations,
or associations between these variables. Common techniques include scatter plots, correlation coefficients, and box plots.

Example 1: Math Score vs Reading Score (Scatter Plot)

This example examines whether there is a linear relationship between students' math and reading scores. A scatter plot is
used to visualize the distribution and trend.

Code:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv("StudentsPerformance.csv")
sns.scatterplot(data=df, x='math score', y='reading score')
plt.title('Math Score vs Reading Score')
plt.xlabel('Math Score')
plt.ylabel('Reading Score')
plt.grid(True)
plt.show()
```

Interpretation:

If the points form an upward trend, this indicates a positive correlation—students who perform well in math also tend to perform well in reading.

Example 2: Gender vs Writing Score (Box Plot)

This example evaluates if gender has an impact on writing scores using a box plot, which shows the spread and central tendency of the scores for each gender.

Code:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv("StudentsPerformance.csv")
sns.boxplot(data=df, x='gender', y='writing score')
plt.title('Gender vs Writing Score')
plt.xlabel('GENDER')
plt.ylabel('WRITING Score')
plt.show()
```

Interpretation:

The box plot can reveal differences in medians, spread, and outliers between genders.

1.3.4 Multivariate Analysis

Multivariate analysis involves examining relationships among three or more variables. It helps in understanding complex
interactions and patterns within the data. Common visual tools include pair plots and heatmaps.

Example 3: Math, Reading, and Writing Scores (Pair Plot)

A pair plot is used to simultaneously visualize the relationships among math, reading, and writing scores.

Code:

```
sns.pairplot(df[['math score', 'reading score', 'writing score']])
plt.suptitle("Multivariate Analysis: Math, Reading, and Writing Scores", y=1.02)
plt.show()
```

Interpretation:
This allows us to identify whether all three scores are positively related, and whether students who perform well in one
subject tend to do well in others.

Example 4: Effect of Test Preparation and Lunch on Math Score

This example explores how two categorical variables (test preparation course and lunch) together influence math scores.

Code:

```
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='test preparation course', y='math
score', hue='lunch')
plt.title('Effect of Test Preparation and Lunch Type on Math Score')
plt.show()
```

Interpretation:
By grouping the data, we can observe combined effects of two categorical variables on the target numeric variable.
Summary:

- Bivariate analysis focuses on relationships between two variables. It helps in

identifying correlations and trends.

- Multivariate analysis allows exploration of complex relationships involving three or more variables.
- Visual tools such as scatter plots, box plots, pair plots, and heatmaps are essential for understanding patterns in data.
- These techniques aid in decision making, feature selection, and predictive.

1.3.5 Handling Missing Data

What is Missing Data?

Missing data refers to empty or unavailable values in a dataset. It occurs when a value that should be there is not recorded. Missing data can appear in one or more cells of a dataset and may happen due to:

- Human error during data entry
- Faulty sensors in data collection
- Non-responses in surveys or forms
- Data loss during transfer or corruption

Example Table:

Student Name	Age	Gender	Math Score
Riya	16	Female	**85**
Aryan		Male	**78**
Simran	**17**		

- Aryan's **Age** is missing.
- Simran's **Gender** and **Math Score** are missing.

Why is Handling Missing Data Important?

1. **Missing values can lead to incorrect analysis**. For example, calculating an average score without handling missing values may give a wrong result.
2. **Some machine learning models do not accept missing values**. If they exist, the model will throw errors.
3. **Data quality is reduced**, which affects predictions and decisions made using that data.

Hence, before any analysis or modeling, we must **identify and handle missing values properly**.

Types of Missing Data

Understanding why data is missing helps us choose the best way to handle it. There are three types:

1. **MCAR (Missing Completely At Random)**
   The missingness is totally random.
   Example: A student accidentally skips a question in an online test.
2. **MAR (Missing At Random)**
   Missing values depend on some **other observed variable**.
   Example: People who did not complete a feedback form may all belong to a specific age group.
3. **MNAR (Missing Not At Random)**
   The missing value is related to the **value itself**.
   Example: People with very high income avoid disclosing it in a form.

Steps to Handle Missing Data

There are multiple strategies. The choice depends on:

- The **amount of missing data**
- Whether the column is **important**
- The **type of data** (numerical or categorical)

Step 1: Detect Missing Data

Use the following code to check missing values:

```
import pandas as pd
df = pd.read_csv('StudentsPerformance.csv')
print(df.isnull().sum())
```

This shows how many missing values are there in each column.

Step 2: Decide How to Handle Missing Data There

are three main ways:

A. Deletion (Dropping Rows or Columns)

1. **Drop Rows – Removes rows that contain any missing value.**

```
df.dropna(inplace=True)
```

Use this when:

- Only a **few rows** are missing.
- Dropping them won't affect the data much.

2. Drop Columns – Removes entire columns that are mostly empty.

```
df.drop(columns=['ColumnName'], inplace=True)
```

Use this when:

- A column has **more than 60-70% missing**.
- That column is **not critical** to your analysis.

Summary Table

Strategy	Type	Best For	Python Code Example
Drop rows	General	Very few missing values	df.dropna()
Drop columns	General	Columns with >70% missing	df.drop(columns=['col'])
Mean/Median/Mode	Numerical	Common and easy fix	df['col'].fillna(df['col'].mean())
Mode or "Unknown"	Categorical	Gender, city, class	df['col'].fillna('Unknown')
Forward/Backward Fill	Time-series	Data recorded in sequence	df.fillna(method='ffill')
Predictive Imputation	**Type**	Important missing info with patterns	Use ML model

1.3.6 Detecting and Handling Outliers

What is an Outlier?

An **outlier** is a data point that is **significantly different** from the rest of the data. It is **unusually high or low** and does not follow the pattern of the other values.

Example:

If most students scored between 60 and 90 in an exam, but one student scored **0** or **100**, that score may be an **outlier**.

Why Do Outliers Matter?

Outliers can:

- **Mislead statistical summaries** like mean and standard deviation
- **Affect visualizations** (e.g., they stretch the scale)
- **Negatively impact machine learning models**, especially those that use distance-based algorithms like linear regression or k-means clustering

So, it's important to **detect** and decide whether to **handle or keep** them.

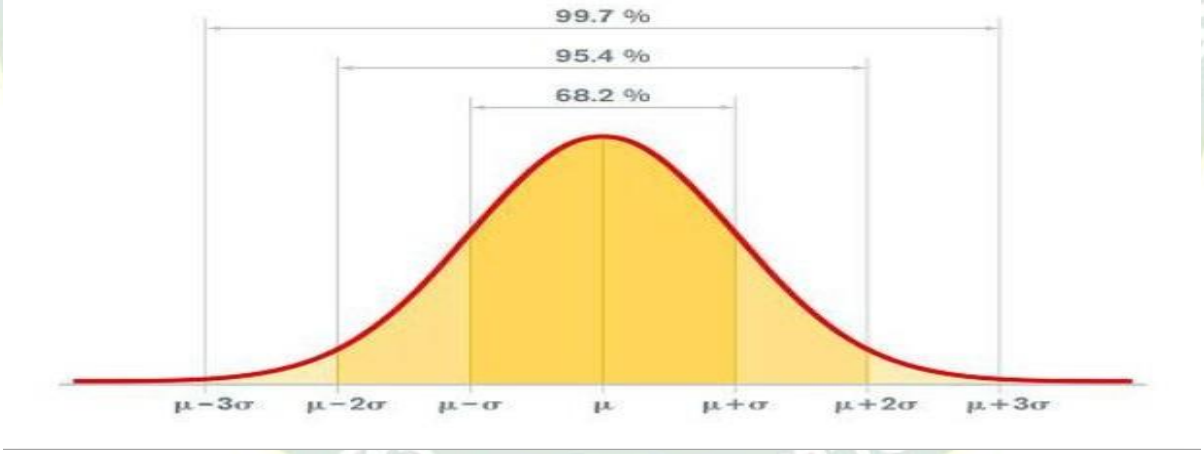
Common Causes of Outliers

- **Data entry errors** (e.g., typing 1000 instead of 100)
- **Measurement errors** (e.g., faulty sensor)
- **Genuine extreme values** (e.g., millionaire income in a population)

Methods to Detect Outliers

1. Using Descriptive Statistics

Calculate **mean** and **standard deviation**. A value far from the mean (typically > 3 standard deviations) is considered an outlier.




```
mean = df['Age'].mean()
std = df['Age'].std()
outliers = df[(df['Age'] > mean + 3*std) | (df['Age'] < mean - 3*std)]
```

2. Using Interquartile Range (IQR) Method (Most common)

- IQR = Q3 - Q1

The **Interquartile Range (IQR)** method is a **robust and commonly used** technique to detect

outliers, especially when the data is **not normally distributed**.

- **Q1 (1st Quartile):** 25% of the data lies **below** this value.
- **Q3 (3rd Quartile):** 75% of the data lies **below** this value.
- **IQR (Interquartile Range):** Measures the **spread of the middle 50%** of the data.

- Outliers are data points **below Q1 – 1.5 × IQR** or **above Q3 + 1.5 × IQR**

```
Q1 = df['Fare'].quantile(0.25)
Q3 = df['Fare'].quantile(0.75)
IQR = Q3 - Q1
lower_limit = Q1 - 1.5 * IQR
 upper_limit = Q3 + 1.5 * IQR
outliers = df[(df['Fare'] < lower_limit) | (df['Fare'] > upper_limit)]
```

3. Using Boxplots (Visual Detection)

Boxplots are a quick visual tool to **spot outliers**.

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.boxplot(x=df['Fare'])
plt.title('Boxplot of Fare')
plt.show()
```

In the plot:

- **Dots outside the box/whiskers** are outliers.
- Boxplots are easy to explain and ideal for student demos.

How to Handle Outliers

Once outliers are detected, we have a few options depending on the context:

A. Keep Them

- If the outliers are **genuine values**, they should be kept.
- Example: High fare paid by a VIP passenger on Titanic — it's valid data.

B. Remove Them

- If outliers are due to **errors or noise**, remove them to prevent skewing results.

Remove outliers using IQR

```
df = df[(df['Fare'] >= lower_limit) & (df['Fare'] <= upper_limit)]
```

C. Cap or Transform Them (Winsorization)

- Replace extreme values with the **upper/lower threshold**.

```
df['Fare'] = df['Fare'].apply(lambda x: upper_limit if x > upper_limit else x)
```

Detection Method	Boxplot
Standard Deviation	Simple numeric analysis
IQR Method	Best for most datasets
Boxplot	Easy visualization tool

Handling Method	Use When
Keep	Outlier is valid and important
Remove	Outlier is error/noise
Cap/Transform	You want to reduce its impact

Unit 1.4: Understanding the Data

In data science, statistics, and machine learning, understanding the type of data you are working with is one of the most important steps. It helps you choose the right analysis method, tools, and models. If you understand what kind of data you have, you can work with it more efficiently and accurately.

Data is mainly divided into two broad categories:

- **Quantitative Data (Numerical)**
- **Qualitative Data (Categorical)**

Let us now understand each type in detail in simple words with lots of examples to make things clearer.

1.4.1 Quantitative Data (Numerical Data)

Quantitative data is data that deals with **numbers**. It answers questions like:

- How much?
- How many?
- What is the value?

You can **count** or **measure** this type of data. Since it is in numeric form, you can do math on it. For example, you can add, subtract, find the average, etc.

Examples of Quantitative Data:

- The number of students in a class (e.g., 40)
- The height of a person (e.g., 172.5 cm)
- The weight of a parcel (e.g., 3.2 kg)
- The speed of a car (e.g., 60 km/h)
- The temperature in a city (e.g., 35.6°C)

Quantitative data is further divided into two types:

1. Discrete Data
2. Continuous Data

1.4.1.1 Discrete Data

Discrete data is data that can be **counted**. It usually consists of **whole numbers** and cannot take values in between. In other words, there are **gaps** between values.

In Simple Words:

You can count discrete data using your fingers. It does **not include decimals or fractions**.

Real-World Examples:

- Number of books on a shelf: 5, 10, 15 (you can't have 5.5 books)
- Number of children in a family: 0, 1, 2
- Number of goals scored in a football match: 1, 2, 3
- Number of cars parked: 12, 18, 20

Discrete data is easy to count and compare. It usually comes from processes where things are counted in units.

Key Characteristics:

- Always whole numbers
- Finite values or countable
- Cannot have decimals

Useful Tip:

If you are **counting something** and the result is a **whole number**, it's likely to be discrete data.

1.4.1.2 Continuous Data

Continuous data is data that can be **measured**, and it can take **any value** within a range. It can have **decimals and fractions**, and the values are not limited to whole numbers.

In Simple Words:

Continuous data is like measuring something very precisely — it can keep changing in small steps.

Real-World Examples:

- Height of a person: 170.2 cm, 165.8 cm
- Weight of a bag: 5.6 kg, 7.25 kg
- Temperature: 36.7°C, 22.3°C
- Distance traveled: 2.5 km, 4.2 km
- Time taken to complete a task: 10.5 minutes, 15.75 minutes

Key Characteristics:

- Infinite possible values within a range
- Can include decimals and fractions
- Measured using tools (ruler, scale, timer, etc.)

Useful Tip:

If you are **measuring** and the result could be **a decimal**, it's likely continuous data.

1.4.2 Qualitative Data (Categorical Data)

Qualitative data is not about numbers. It describes **qualities or categories**. This type of data tells us about **types**, **names**, or **labels**.

In Simple Words:

It is used to group things based on qualities or categories, not on how much or how many.

Examples of Qualitative Data:

- Gender: Male, Female, Other
- Eye color: Brown, Blue, Green
- City: Mumbai, Delhi, Kolkata
- Car brand: Toyota, Ford, Hyundai

You cannot do math on qualitative data. You cannot add two eye colors or average the names of cities.

Qualitative data is further divided into:

1. Nominal Data
2. Ordinal Data

1.4.2.1.1 Nominal Data

Nominal data is made up of categories that do **not have a specific order**. The labels or names are **just identifiers**.

In Simple Words:

These are categories that you can name, but you **cannot compare or rank** them.

Real-World Examples:

a. Blood group: A, B, AB, O
b. Marital Status: Single, Married, Divorced
c. Religion: Hindu, Muslim, Christian
d. Country of birth: India, USA, Canada

Key Characteristics:

e. No order or ranking between categories
f. Used for labelling or classification
g. Can be represented using words or numbers (e.g., 0 = Male, 1 = Female, but numbers have no mathematical meaning)

Useful Tip:

If changing the **order** of categories doesn't change the **meaning**, it's nominal.

1.4.2.2 Ordinal Data

Ordinal data also involves categories, but in this case, the **order or ranking matters**. However, the **difference between each level is not measurable**.

In Simple Words:

These are categories that you can **rank** or **order**, but you don't know how much better one is than the other.

Real-World Examples:

- Education level: High School < Bachelor's < Master's < PhD
- Customer satisfaction: Poor < Fair < Good < Excellent

- Class rank: First, Second, Third
- Spicy level: Mild < Medium < Hot

Key Characteristics:

- Categories have a meaningful order
- No measurable gap between levels
- Used when ranking is more important than the actual value

Useful Tip:

If changing the **order** changes the **meaning**, it's ordinal data.

Summary Table:

Category	Subtype	Nature	Can Do Math?	Examples
Quantitative	Discrete	Countable	Yes	Number of students, cars
	Continuous	Measurable	Yes	Weight, height, temperature
Qualitative	Nominal	Labels only	No	Gender, blood group, city names
	Ordinal	Ordered categories	No	Satisfaction level, education rank

Final Thoughts:

Understanding the type of data helps in:

h. Choosing the right graph (e.g., bar chart, pie chart, histogram)
i. Using the correct statistical tools
j. Making accurate data interpretations
k. Avoiding analysis errors

For example, you should not try to find the average of colors or names. Similarly, you should not ignore the order in ordinal data.

So always start with one question: **What type of data am I working with?**

Knowing the answer will save you time and give you better results in data analysis, machine learning, and research work.

1.5 Spread of Data

Includes:

- Normal Distribution

- Skewed Distribution
- Skewness
- Kurtosis

What is "Spread of Data"?

When we collect data, not all values are the same. Some are close together, others are far apart. The "spread" of data tells us how much the data values vary from each other and from the average.

Analogy:

Imagine 30 students running a 100-meter race.

- If most students finish around 15 seconds, the timing data is less spread out.
- But if some finish in 10 seconds, some in 30 seconds, then it's more spread out.

1.5.1 Normal Distribution

Simple Definition:

A Normal Distribution is a type of data distribution where most values cluster around the average (mean), and fewer values are found as we move away from the center.

Real-World Analogy:

Think of the height of adults in a school. Most adults are around 5.5 to 6 feet tall. Very few are extremely short or extremely tall.

If you plot everyone's height, the graph would look like a bell curve.

Shape:

- Looks like a symmetrical bell.
- Mean = Median = Mode → All are in the middle.

68-95-99.7 Rule:

If the average (mean) is 60 and standard deviation is 10:

- 68% of values lie between 50 and 70
- 95% between 40 and 80
- 99.7% between 30 and 90

This helps predict how "usual" or "unusual" a value is.

1.5.2 Skewed Distribution

Simple Definition:

A Skewed Distribution happens when data is not evenly spread around the average. One side has more values or longer tails.

There are two types of skewed distributions:

1. **Positively Skewed (Right-Skewed)**

   Analogy:

Think of monthly salaries in a company:

- Most employees earn ₹30,000–₹60,000
- But a few top executives earn ₹1 lakh–₹5 lakh

This creates a long tail towards the right (positive direction)

Features:

- Mean > Median > Mode
- Tail is longer on the right
- More small values, few very high values

2. **Negatively Skewed (Left-Skewed)**

   Analogy:

   Think of retirement age:
   Most people retire at 58–65
   Few retire early (at 30–40)
   This makes a long tail on the left **Features:**
   Mean < Median < Mode
   Tail is longer on the left
   More high values, few very low values

1.5.3 Skewness

Simple

Definition:

Skewness tells us how lopsided (or tilted) the data is.

Think of a seesaw:

- If both sides have equal weight → Balanced → Skewness = 0
- If one side is heavier → Tilted → Skewness ≠ 0

Types of Skewness:

Skewness Value	Shape	Meaning
0	Perfectly symmetrical	Normal distribution

> 0	Tail to the right	Positively skewed
< 0	Tail to the left	Negatively skewed

Real-Life Examples:

- Exam scores where most students scored low, but a few topped → Positive skew
- Age at which people die (most at old age, few young) → Negative skew

1.5.4 Kurtosis

Simple Definition:

Kurtosis tells us how "fat" or "thin" the tails of the distribution are. It describes the extremeness of outliers.

Analogy: Hills

Imagine hills with different shapes.

1. Mesokurtic (Normal)
   - Shape: Bell-like
   - Moderate height & tail
   - Like a normal hill
2. Leptokurtic (High kurtosis)
   - Shape: Tall & sharp
   - Heavy tails (many extreme values)
   - Like a sharp mountain
   - Example: Stock market returns
3. Platykurtic (Low kurtosis)
   - Shape: Flat & wide
   - Light tails (few outliers)
   - Like a flat hill
   - Example: Dice throws

Summary Table:

Type	Shape	Outliers	Example
Mesokurtic	Normal (bell)	Moderate	Heights of adults

Leptokurtic	**Tall & pointy**	**Many extreme**	**Crypto price changes**
Platykurtic	**Flat & wide**	**Few or no outliers**	**Rolling a fair dice**

Bonus Tip:

- Skewness = Tilt
- Kurtosis = Tails

Final Summary:

Term	Meaning (Simple)	Real-life Analogy
Spread of Data	**How far values are from each other and the average**	**Students finishing a race at different times**
Normal	**Symmetrical bell curve, average in the middle**	**Human height or test scores**
Skewed	**Lopsided graph with one tail longer**	**Salaries, retirement age**
Skewness	**Number that shows tilt**	**A seesaw – balance or tilt**
Kurtosis	**Tells how thick the tails are (outliers)**	**Shape of a hill (sharp or flat)**

`

Prepared By:

Dr. Pooja R. Negi