

Basics Of Data Science



Unit 1

Introduction to core concepts & technologies



Contents :-

- Introduction
- Terminology
- data science process
- data science toolkit
- Types of data
- Example
- Applications
- Mathematical Foundations for Data Science : linear algebra
- Analytical and numerical solutions of linear equations
- Mathematical structures
- concepts and notations used in discrete mathematics



Introduction :

Data Science is the area of study which involves extracting insights from vast amounts of data using various scientific methods, algorithms, and processes.

It helps you to discover hidden patterns from the raw data.

The term Data Science has emerged because of the evolution of mathematical statistics, data analysis, and big data.

Data Science is an interdisciplinary field that allows you to extract knowledge from structured or unstructured data.

Data science enables you to translate a business problem into a research project and then translate it back into a practical solution.





Terminology :-

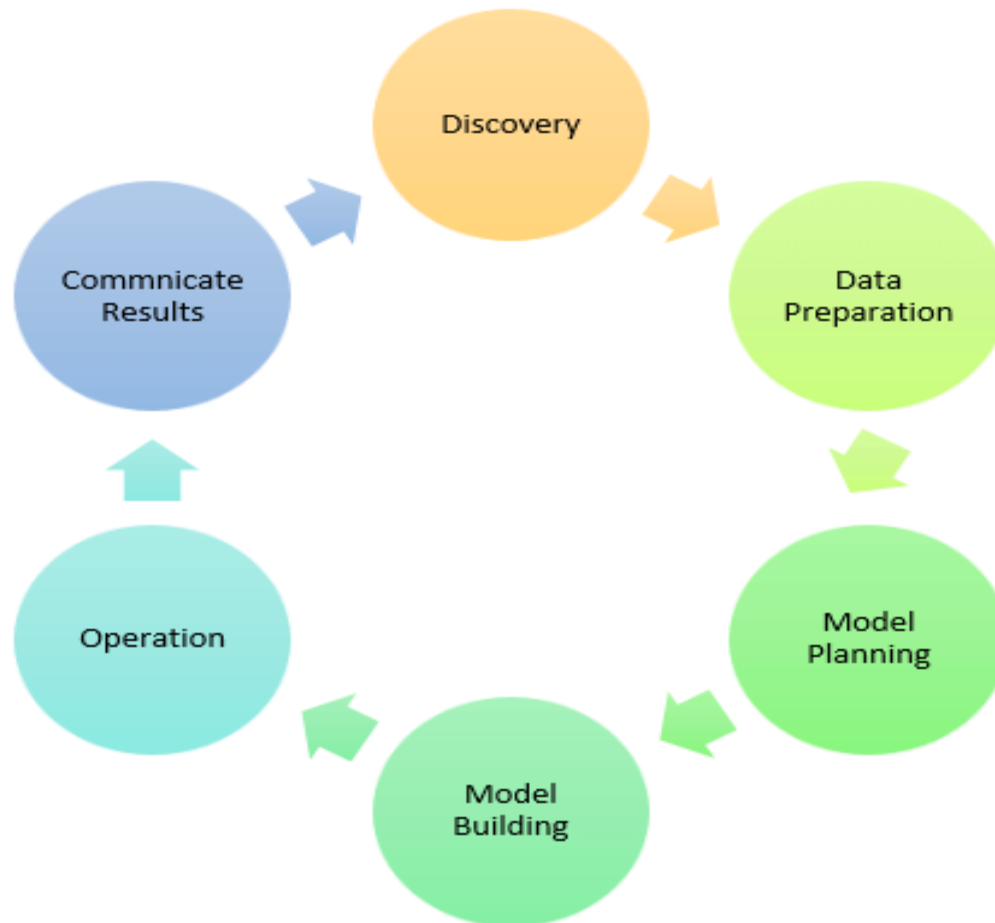
Data Science is a field that combines programming skills and knowledge of mathematics and statistics to derive insights from data.

In short: Data Scientists work with large amounts of data, which are systematically analyzed to provide meaningful information that can be used for decision making and problem solving.



Data Science Process :-

The data science process is **a systematic approach to solving a data problem**. It provides a structured framework for articulating your problem as a question, deciding how to solve it, and then presenting the solution to stakeholders.



- **Discovery**

To begin with, it is exceptionally imperative to get the different determinations, prerequisites, needs and required budget-related with the venture. You must have the capacity to inquire the correct questions like do you have got the desired assets. These assets can be in terms of individuals, innovation, time and information. In this stage, you too got to outline the trade issue and define starting hypotheses (IH) to test.

- **Information Preparation**

In this stage, you would like to investigate, preprocess and condition data for modeling. You'll be able to perform information cleaning, changing, and visualization. This will assist you to spot the exceptions and build up a relationship between the factors. Once you have got cleaned and arranged the information, it's time to do exploratory analytics on it.

- **Model Planning**

Here, you may decide the strategies and methods to draw the connections between factors. These connections will set the base for the calculations which you may execute within the following stage. You may apply Exploratory Data Analytics (EDA) utilizing different factual equations and visualization apparatuses.



- **Model Building**

In this stage, you'll create datasets for training and testing purposes. You may analyze different learning procedures like classification, association, and clustering and at last, actualize the most excellent fit technique to construct the show.

- **Operationalize**

In this stage, you convey the last briefings, code, and specialized reports. In expansion, now a pilot venture is additionally actualized in a real-time generation environment. This will give you a clear picture of the execution and other related limitations.

- **Communicate Results**

Presently, it is critical to assess the outcome of the objective. So, within the final stage, you recognize all the key discoveries, communicate to the partners and decide in the event that the outcomes about the venture are a victory or a disappointment based on the criteria created in Stage 1.



Data Science Toolkit :-

A Data Scientists primary role is to apply machine learning, statistical methods and exploratory analysis to data to extract insights and aid decision making. Programming and the use of computational tools are essential to this role.

The. Data Science Toolkit is **a collection of the best open data sets and open-source tools for data science**, wrapped in an easy-to-use REST/JSON API with command line, Python and Javascript interfaces.

1. Programming Languages:

- Python: Widely used for data analysis, machine learning, and data visualization.

Libraries like NumPy,

Pandas, Matplotlib, and Scikit-Learn are popular for data science tasks.

- R: Another popular language for data analysis and statistics, known for its extensive set of statistical packages.

2. Integrated Development Environments (IDEs):

- Jupyter Notebook: An interactive and web-based environment for writing and running code, making it

easy to document and share data analysis.

- RStudio: An IDE specifically designed for R programming.



3. Data Manipulation and Analysis Tools:

- Pandas: A Python library for data manipulation and analysis, providing data structures like DataFrames.
- SQL: For querying and working with relational databases.
- Excel: Often used for data exploration and basic analysis.

4. Data Visualization Tools:

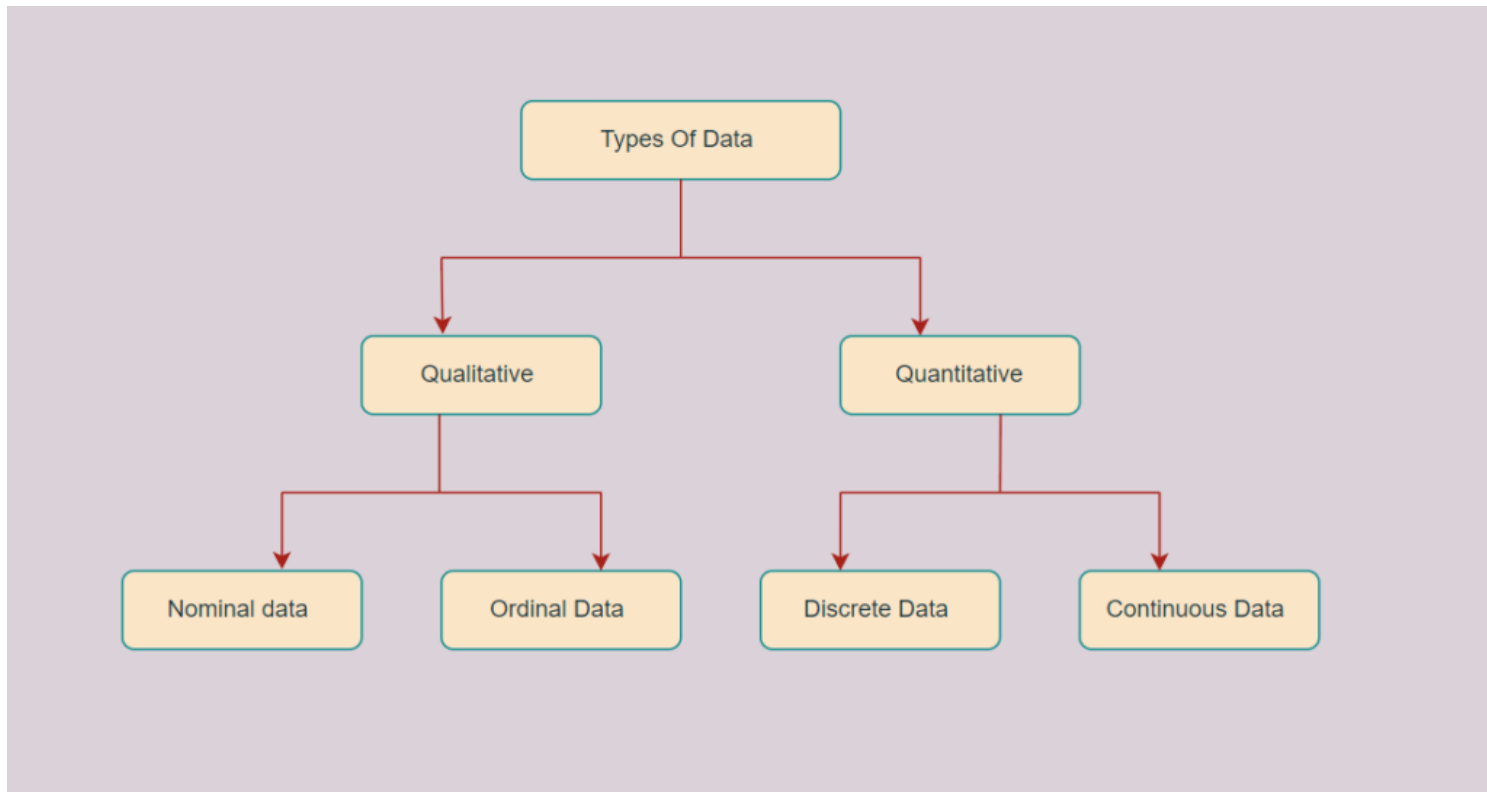
- Matplotlib: A Python library for creating static, animated, or interactive visualizations.
- Seaborn: A Python library built on top of Matplotlib for creating attractive statistical graphics.
- Tableau, Power BI, or Qlik: Tools for creating interactive and shareable data visualizations.

5. Machine Learning Librarie



Types of data :-

There are two types of data: **Qualitative and Quantitative data**, which are further classified into **four types data: nominal, ordinal, discrete, and Continuous**.



Numerical data: Numerical, or quantitative, data is a type of data that represents numbers rather than natural language descriptions, so it can only be collected in a numeric form.

Examples of quantitative data include arithmetic operations (addition, subtraction, division, and multiplication), and ways to measure a person's weight and height.

It is also divided into two subsets: discrete data and continuous data.

Discrete data:

The main feature of this data type is that it is countable, meaning that it can take certain values like numbers 1, 2, 3 and so on.

Examples of these types of data are age, the number of children you want to have (the number is a non-negative integer because you can't have 1.5 or -2 kids)

Continuous data:

Continuous data is a type of data with uncountable elements. It is represented as a set of intervals on a number line.



Examples of continuous data are the measure of weight, height, area, distance, time, etc.

This type of data can be further divided into interval data and ratio data.

Interval data:

Interval data is measured along a scale, in which each point is placed at an equal distance, or interval, from one another.

Ratio data:

Ratio data is almost the same as the previous type but the main difference is that it has a zero point. For instance, the zero point temperature can be measured in Kelvin. It is equal to -273.15 degrees Celsius

Categorical data

Categorical, or qualitative data, is information divided into groups or categories using labels or names. In such dataset, each item is placed in a single category depending on its qualities. All categories are mutually exclusive.

Numbers in this type of data do not have mathematical meaning, i.e. no arithmetical operations can be performed with numerical variables.

A good example of categorical data is when you are filling out forms for job applications. You may be asked to specify your level of education.



Categorical data is further divided into nominal data and ordinal data.

Nominal data:

Nominal data, also known as naming data, is descriptive and has a function of labeling or naming variables. Elements of this type of data do not have any order, or numerical value, and cannot be measured. Nominal data is usually collected via questionnaires or surveys. E.g.: Person's name, eye color, clothes brand.

Ordinal data:

This type of data represents elements that are ordered, ranked, or used on a rating scale. Generally speaking, these are categories with an implied order. Though ordinal data can be counted, it cannot be measured as well as nominal one. Examples of ordinal data include customer satisfaction rating.



Examples Applications :-

Data science has found its applications in almost every industry:

- Healthcare
- Gaming
- Image Recognition
- Recommendation Systems
- Logistics
- Fraud Detection
- Internet Search
- Speech Recognition



Mathematical Foundations for Data Science:

Linear Algebra :-

Mathematical topics fundamental to computing and statistics including trees and other graphs, counting in combinatorics, principles of elementary probability theory, linear algebra, and fundamental concepts of calculus in one and several variables.

Linear Algebra is used in machine learning to understand how algorithms work under the hood. It's all about vector/matrix/tensor operations;



Analytical & Numerical Solutions of Linear Equations :-

An **analytical solution** involves framing the problem in a well-understood form and calculating the exact solution.

a. Substitution and Elimination

Solve one equation for a variable and substitute it into others

Example:

$$\text{Solve: } x + y = 5$$

$$2x - y = 3.$$

Substitute $y = 5 - x$ into the second equation to find x , and then solve for y .

A **numerical solution** means making guesses at the solution and testing whether the problem is solved well enough to stop.



Comparison: Analytical vs. Numerical Solutions

Aspect	Analytical	Numerical
Exactness	Provides exact solutions.	Provides approximate solutions.
Scalability	Suitable for small systems.	Efficient for large-scale systems.
Stability	Less stable for ill-conditioned matrices.	More robust for ill-conditioned matrices.
Complexity	Computationally expensive for large systems.	Optimized for large and sparse systems.



Mathematical Structures :-

Algebraic Structure in Discrete Mathematics

The algebraic structure is a type of non-empty set G which is equipped with one or more than one binary operation. Let us assume that $*$ describes the binary operation on non-empty set G . In this case, $(G, *)$ will be known as the algebraic structure. $(1, -)$, $(1, +)$, $(\mathbb{N}, *)$ all are algebraic structures.

$(\mathbb{R}, +, \cdot)$ is a type of algebraic structure, which is equipped with two operations $(+)$ and (\cdot) .

Examples of Algebraic Structures $(1, -)$: Here, the set is $\{1\}$, and the operation is subtraction $(-)$.

Subtracting any element in the set results in another element in the set (though $\{1\}$ is trivial).

$(1, +)$: The set $\{1\}$ with addition $(+)$. Since adding $1+1=2$ leaves the set, this specific pair is valid only when constraints on the operation or set size are trivially satisfied.



Types of Algebraic Structures

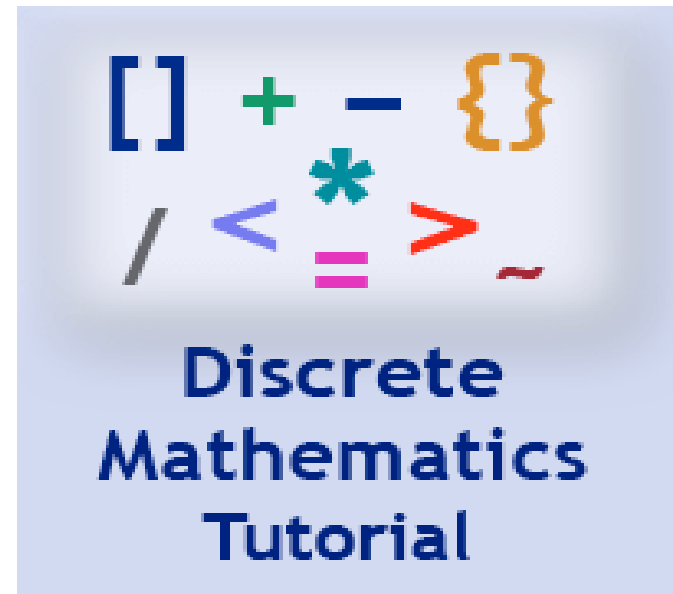
- Semigroup: A set $(G,*)$ where $*$ is associative.
- Monoid: A semigroup with an identity element.
- Group: A monoid where every element has an inverse.
- Abelian Group: A group where the operation is commutative.



Concepts & Notations used in Discrete mathematics :-

Discrete mathematics Tutorial provides basic and advanced concepts of Discrete mathematics. Our Discrete mathematics Structure Tutorial is designed for beginners and professionals both.

Discrete mathematics is the branch of mathematics dealing with objects that can consider only distinct, separated values. This tutorial includes the fundamental concepts of Sets, Relations and Functions, Mathematical Logic, Group theory, Counting Theory, Probability, Mathematical Induction, and Recurrence Relations, Graph Theory, Trees and Boolean Algebra.



Discrete Mathematics is the language of Computer Science. To learn or become master of many fields like data science, machine learning, and software engineering, it is necessary to have knowledge of discrete mathematics.

It is a branch of mathematics that deals with separable and distinct numbers. Combinations, graph theory, and logical statements are included, and numbers can be finite or infinite.

It's used in computer science to design the apps and programs we use every day. While there are no hard and fast definitions of discrete mathematics, it's well known for the things it excludes - continuously varying quantities and all things related to that.

Discrete mathematics is vital to digital devices. With tech continually on the rise, studying this overlooked area of mathematics could prove valuable for your career and your future.

The purpose of this course is to understand and use (abstract) discrete structures that are backbones of computer science. In particular, this class is meant to introduce logic, proofs, sets, relations, functions, counting, and probability, with an emphasis on applications in computer science.



Unit 2

Data collection and management



Contents :

- Introduction
- Sources of data
- Data collection and APIs
- Exploring and fixing data
- Data storage and management
- using multiple data sources



Introduction :-

- The process of gathering and analyzing accurate data from various sources to find answers to research problems, trends and probabilities, etc., to evaluate possible outcomes is Known as **Data Collection** .
- The main objective of data collection is to gather information-rich and reliable data, and analyze them to make critical business decisions.
- During data collection, the researchers must identify the data types, the sources of data, and what methods are being used

The data collection process has had to change and grow with the times, keeping pace with technology.

Here are several key points to consider regarding data collection in data science:

- Data collection is essential for both business and research.
- Data collecting helps in the gathering of information, the testing of hypothesis, and the production of relevant findings.
- It enables scientists to find correlations, trends, and patterns that lead to important discoveries.
- In business, data collection offers insights into consumer behavior, market trends, as well as operational effectiveness.
- It enables businesses to streamline operations and make data-driven decisions.
- A competitive advantage in the market is provided by data collection.



Data management refers to the process by which data is effectively acquired, stored, processed, and applied, aiming to bring the role of data into full play.

In terms of business, data management includes metadata management, data quality management, and data security management.

Key aspects of data management include:

1.Data Collection – Gathering data from various sources such as databases, APIs, IoT devices, and manual inputs.

2.Data Storage – Storing data in databases, data warehouses, or cloud storage solutions.

3.Data Organization – Structuring and categorizing data using metadata, indexing, and classification.



- **Data Security** – Protecting data from unauthorized access, cyber threats through encryption and access controls.
- **Data Governance** – Establishing policies and procedures to ensure data integrity, compliance, and ethical use.
- **Data Quality** – Ensuring data accuracy, consistency, and completeness through validation and cleansing processes.
- **Data Integration** – Combining data from different sources to create a unified view for analysis and decision-making.
- **Data Analytics** – Using tools like AI, machine learning, and business intelligence to extract insights from data.
- **Data Backup & Recovery** – Creating backups and disaster recovery plans to prevent data loss.
- **Data Lifecycle Management** – Managing data from creation to deletion while ensuring compliance with regulations.

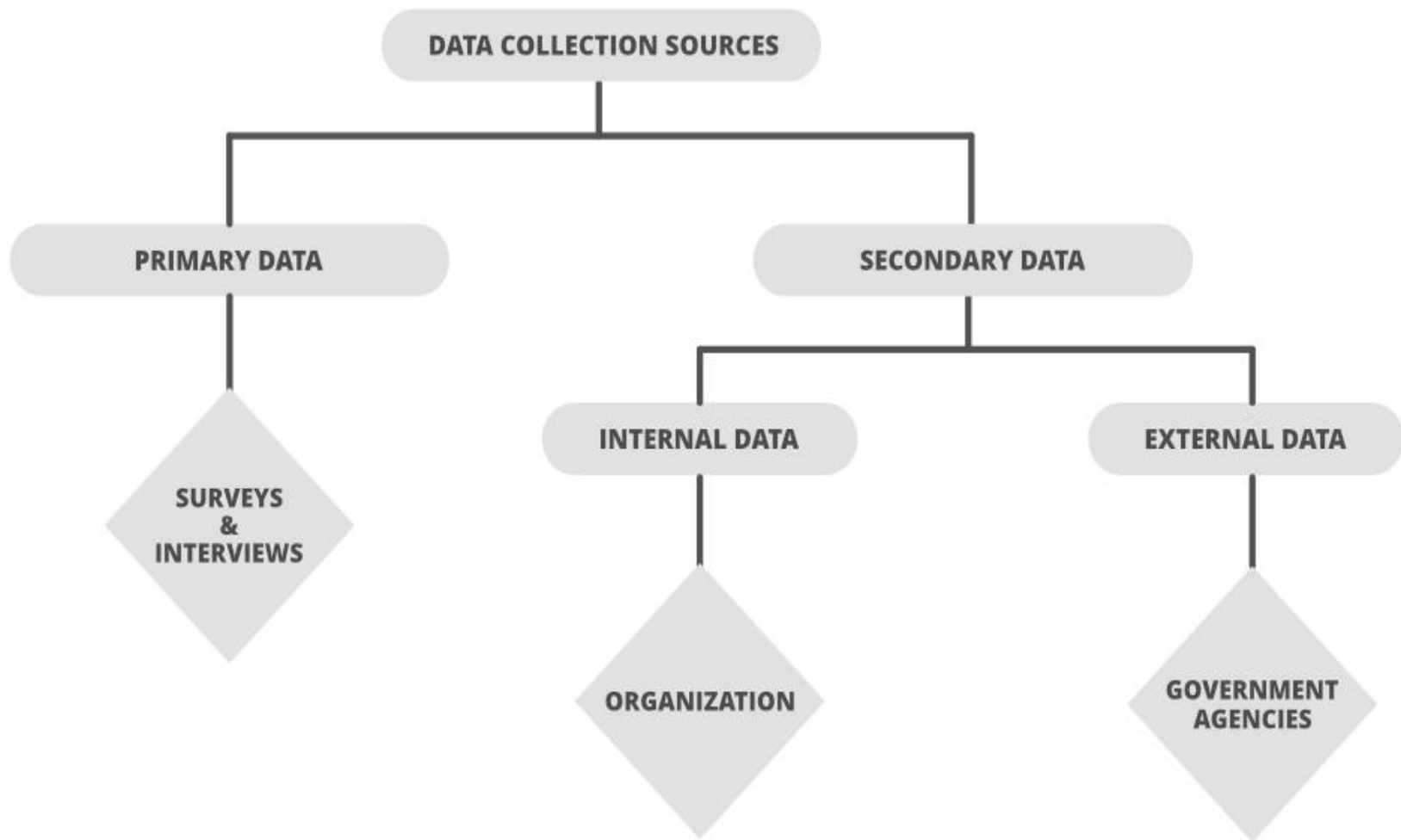


Sources of data :-

A **data source** can be the original site where data is created or where physical information is first digitized. Still, even the most polished data can be used as a source if it is accessed and used by another process.

A **data source** can be a database, a flat file, real-time measurements from physical equipment, scraped online data, or any of the numerous static and streaming data providers available on the internet.





Primary data:

- The data which is Raw, original, and extracted directly from the official sources is known as primary data.
- This type of data is collected directly by performing techniques such as questionnaires, interviews, and surveys.
- Few methods of collecting primary data:
- **Interview method**
- **Survey method**
- **Observation method:**
- **Experimental method**



Secondary data:

Sources of data can also be classified based on its collection methods, which are –

Internal Sources of Data

In several cases for a certain analysis, data is collected from records, archives, and various other sources within the organization itself. Such sources of data are termed internal sources of data.

Example: A school is performing an analysis to figure out the highest marks achieved in class 8 science subjects for the last 10 years.

External Sources of Data

Data may also be collected from various sources outside the organization for analytical purposes. Such sources of data collection are known as external sources of data.

Example: As a patient, you are analyzing the price charts of your nearby hospitals for the treatment of ulcers.



Examples of Secondary Data Sources:

- **Government & Public Datasets** – Census data, economic reports (e.g., Data.gov, WHO, IMF)
- **Company Databases & Reports** – Sales records, financial reports
- **Research Publications & Journals** – Scientific papers, industry reports (e.g., IEEE, PubMed)
- **Social Media & Web Data** – Twitter API, Google Trends, Wikipedia datasets
- **Third-party APIs & Market Data** – Google Analytics, Stock Market data (Yahoo Finance, Bloomberg)



Data collection and APIs :-

Data collection is the process of acquiring, collecting, extracting, and storing the voluminous amount of data which may be in the structured or unstructured form like text, video, audio, XML files, records, or other image files used in later stages of data analysis.

In the process of big data analysis, “Data collection” is the initial step before starting to analyze the patterns or useful information in data. The data which is to be analyzed must be collected from different valid sources.

Data collection starts with asking some questions such as what type of data is to be collected and what is the source of collection.

Most of the data collected are of two types known as “qualitative data“ which is a group of non-numerical data such as words, sentences mostly focus on behavior and actions of the group and another one is “quantitative data” which is in numerical forms and can be calculated using different scientific tools and sampling data.



APIs

APIs, or Application Program Interfaces, and Web Services are provided by many data providers and websites, allowing various users or programmes to communicate with and access data for processing or analysis.

APIs and Web Services often listen for incoming requests from users or applications, which might be in the form of web requests or network requests, and return data in plain text, XML, HTML, JSON, or media files.

APIs (Application Programming Interfaces) are widely used to retrieve data from a number of data sources. APIs are used by apps that demand data and access an end-point that contains the data. Databases, online services, and data markets are examples of end-points.

APIs are also used to validate data. An API might be used by a data analyst to validate postal addresses and zip codes,



Types of APIs for Data Science

1.REST APIs (Representational State Transfer)

1. Most common, uses HTTP requests (GET, POST, PUT, DELETE)
2. Example: Twitter API, Google Maps API

2.SOAP APIs (Simple Object Access Protocol)

1. More secure but heavier compared to REST
2. Example: Some banking and enterprise APIs

3.GraphQL APIs

1. More flexible, allows fetching only required data
2. Example: GitHub GraphQL API

4.Streaming APIs

1. Used for real-time data
2. Example: Twitter Streaming API for live tweets



Making API Requests in Python

In order to work with API some tools are required such as requests so we need to first install them in our system.

```
pip3 install requests
```

REST API (Representational state transfer) is an API that uses [HTTP requests](#) for communication with web services.

Types of Requests

Types of Requests or [HTTP Request Methods](#) characterize what action we are going to take by referring to the API.

In total, there are four main types of actions:

- **GET**: retrieve information (like search results). This is the most common type of request. Using it, we can get the data we are interested in from those that the API is ready to share.
- **POST**: adds new data to the server. Using this type of request, you can, for example, add a new item to your inventory.
- **PUT**: changes existing information. For example, using this type of request, it would be possible to change the color or value of an existing product.
- **DELETE**: deletes existing information

Example:

```
import requests
response = requests.get('https://google.com/')
print(response)
>> <Response [200]>
```



Exploring and fixing data :-

Data exploration refers to the initial step in data analysis. Data analysts use data visualization and statistical techniques to describe dataset characterizations, such as size, quantity, and accuracy, to understand the nature of the data better.

Data exploration techniques include both manual analysis and automated data exploration software solutions that visually explore and identify relationships between different data variables, the structure of the dataset, the presence of outliers, and the distribution of data values to reveal patterns and points of interest, enabling data analysts to gain greater insight into the raw data.

How Data Exploration Works?

Data Collection: Data exploration commences with collecting data from diverse sources such as databases, [APIs](#), or through web scraping techniques.

Data Cleaning: it involves handling missing values, fixing incorrect data types, removing duplicates, dealing with outliers, and ensuring data consistency.



3.Exploratory Data Analysis (EDA): This EDA phase involves the application of various statistical tools such as box plots, scatter plots, histograms, and distribution plots. Additionally, correlation matrices and descriptive statistics are utilized to uncover links, patterns, and trends within the data.

4.Feature Engineering: Feature engineering focuses on enhancing prediction models by introducing or modifying features. Techniques like data normalization, scaling, encoding, and creating new variables are applied..

5.Model Building and Validation: During this stage, preliminary models are developed to test hypotheses or predictions.



Steps involved in Data Exploration

Data exploration is an iterative process, but there are generally some key steps involved:

Data Understanding

- **Familiarization:** Get an overview of the data format, size, and source.
- **Variable Identification:** Understand the meaning and purpose of each variable in the dataset.

Data Cleaning

- **Identifying Missing Values:** Locate and address missing data points strategically (e.g., removal, imputation).
- **Error Correction:** Find and rectify any inconsistencies or errors within the data.
- **Outlier Treatment:** Identify and decide how to handle outliers that might skew the analysis.




Exploratory Data Analysis (EDA)

- **Univariate Analysis:** Analyze individual variables to understand their distribution (e.g., histograms, boxplots for numerical variables; frequency tables for categorical variables).
- **Bivariate Analysis:** Explore relationships between two variables using techniques like scatterplots to identify potential correlations.

Data Visualization

- **Creating Visualizations:** Use charts and graphs (bar charts, line charts, heatmaps) to effectively communicate patterns and trends within the data.
- **Choosing the Right Charts:** Select visualizations that best suit the type of data and the insights you're looking for.

Iteration and Refinement

- **Iterate:** As you explore, you may need to revisit previous steps.
 - **Refinement:** New discoveries might prompt you to clean further, analyze differently, or create new visualizations.
- 

Data cleaning or fixing is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled.

If data is incorrect, outcomes and algorithms are unreliable, even though they may look correct. There is no one absolute way to prescribe the exact steps in the data cleaning process because the processes will vary from dataset to dataset. But it is crucial to establish a template for your data cleaning process so you know you are doing it the right way every time.

Data cleaning is the process that removes data that does not belong in your dataset.



Data storage and management :-

Data storage management helps organizations understand where they have data, which is a major piece of compliance. Compliance best practices include documentation, automation, and use of governance tools. Immutable data storage also helps achieve compliance.

Data storage is the retention of information using technology specifically developed to keep that data and have it as accessible as necessary. Data storage refers to the use of recording media to retain data using computers or other devices.

Data management refers to the process by which data is effectively acquired, stored, processed, and applied, aiming to bring the role of data into full play. In terms of business, data management includes metadata management, data quality management, and data security management.



Data storage and management in data science include:

- **Informed Decision-Making Process**
- **Data Quality and Efficiency.**
- **Compliance and Customer Trust**
- **Strategy Development and Innovation**
- **Long-term Sustainability**

Data Management Tools and Technologies

Relational Database Management Systems (RDBMS):

Data Warehouse

- Amazon Redshift
- Google BigQuery
- Snowflake

ETL (Extract, Transform, Load) Tools:

- Apache NiFi
- Talend
- Apache Spark

Data Visualization and Business Intelligence:

- Tableau



Using multiple data sources :-

By using **multiple data sources** for your model, you can reduce the total volume of data processed . If used in combination with calculated columns, multiple data sources can minimize or eliminate the need to create database table joins in an external data access tool. Using multiple data sources also enables measure allocation.

For example, suppose your product, customer, and order data is stored in a set of tables. If you were to use this data from a single source, you would need separate tables for Product, Customer, CustomerSite, Order, and OrderDetail. This source would contain many duplicate values, and the joins between the tables would be relatively complex. Instead, you create three separate sources for Products, Customer/Site, and Order/Order Detail data. The volume of data contained in each is less than that in the single source, and there are only simple joins between Customer and CustomerSite tables, and Order and OrderDetail.



Here are some key considerations and benefits when using multiple data sources in data science

Data Enrichment: By integrating data from diverse sources, you can enrich your datasets with additional information.

Improved Accuracy: When you have data from multiple sources that cover different aspects of the same topic, you can cross-reference and validate the information. This can lead to more accurate and reliable results.

Uncovering Hidden Patterns: Different data sources may capture unique patterns or trends that are not apparent when analyzing each source independently. Combining them can reveal hidden insights and relationships.

Handling Missing Data: One data source may have missing data for certain records, while another source may have that missing information. Integrating these sources can help fill in the gaps and create a more complete dataset

Data Validation: Integrating data from multiple sources allows you to validate the consistency and accuracy of the information. Inconsistent data between sources may indicate potential errors or data quality issues.

Business Intelligence and Decision-Making: Data from various departments or external sources can be combined to provide a holistic view of an organization, enabling better business intelligence and strategic decision-making



Unit 3 : Data analysis

DR Mrs J. N. Jadhav Associate professor
Deptt of CSE, DYPCET



Contents :

- ❖ Introduction
- ❖ Terminology and concepts
- ❖ Introduction to statistics
- ❖ Central tendencies and Distributions
- ❖ Variance
- ❖ Distribution properties and arithmetic Samples/CLT
- ❖ Basic machine learning algorithms
- ❖ Linear regression
- ❖ SVM
- ❖ Naive Bayes



Introduction :-

Data analysis, is a process for obtaining raw data, and subsequently converting it into information useful for decision-making by users. Data, is collected and analyzed to answer questions, test hypotheses, or disprove theories.

Data analytics is a process of evaluating data using analytical and logical concepts to examine a complete insight of all the employees, customers and business. In this process, the user data is extracted from raw data using specialized computer systems.

A simple example of data analysis can be seen whenever we make a decision in our daily lives by evaluating what has happened in the past or what will happen if we make that decision.

Basically, this is **the process of analyzing the past or future and making a decision based on that analysis.**



Introduction to Data Analytics

- Start by learning the fundamentals of data analytics. Learn about its significance and uses also. Research the trends which are unnoticed, their correlations, and other perspectives that can help in overall decision-making.
- Learn how data analytics can be used in various different fields or domains such as healthcare systems, finance or e-commerce organizations, marketing, and many more. If we take an example learn about how finance organizations use data analytics methods in order to detect fraudulent transactions.
- Take your time to analyze and think about why you need to learn data analytics, whether your interest lies in this field or not, and why you wish to pursue a career in the field of data analytics.



Terminology and concepts :-

Terminology is a discipline that systematically studies the "labelling or designating of concepts" particular to one or more subject fields or domains of human activity. It does this through the research and analysis of terms in context for the purpose of documenting and promoting consistent usage.

Terminology term means different things in different contexts. **To a lay person, it might mean the automated searching of large databases.** To an analyst. it may refer to the collection of statistical and machine learning methods used with those databases (predictive modeling, clustering, recommendation systems, ...)

There are four key types of data analytics: **descriptive, diagnostic, predictive, and prescriptive.** Together, these four types of data analytics can help an organization make data-driven decisions.



Introduction to statistics :-

Statistics is a form of mathematical analysis that uses quantified models and representations for a given set of experimental data or real-life studies. The main advantage of statistics is that information is presented in an easy way.

Statistics is like the heart of Data Science that helps to analyze, transform and predict data.

Types of Statistics

There are commonly two types of statistics, which are discussed below:

1.Descriptive Statistics: Descriptive Statistics helps us simplify and organize big chunks of data. This makes large amounts of data easier to understand.

2.Inferential Statistics: Inferential Statistics is a little different. It uses smaller data to conclude a larger group. It helps us predict and draw conclusions about a population.



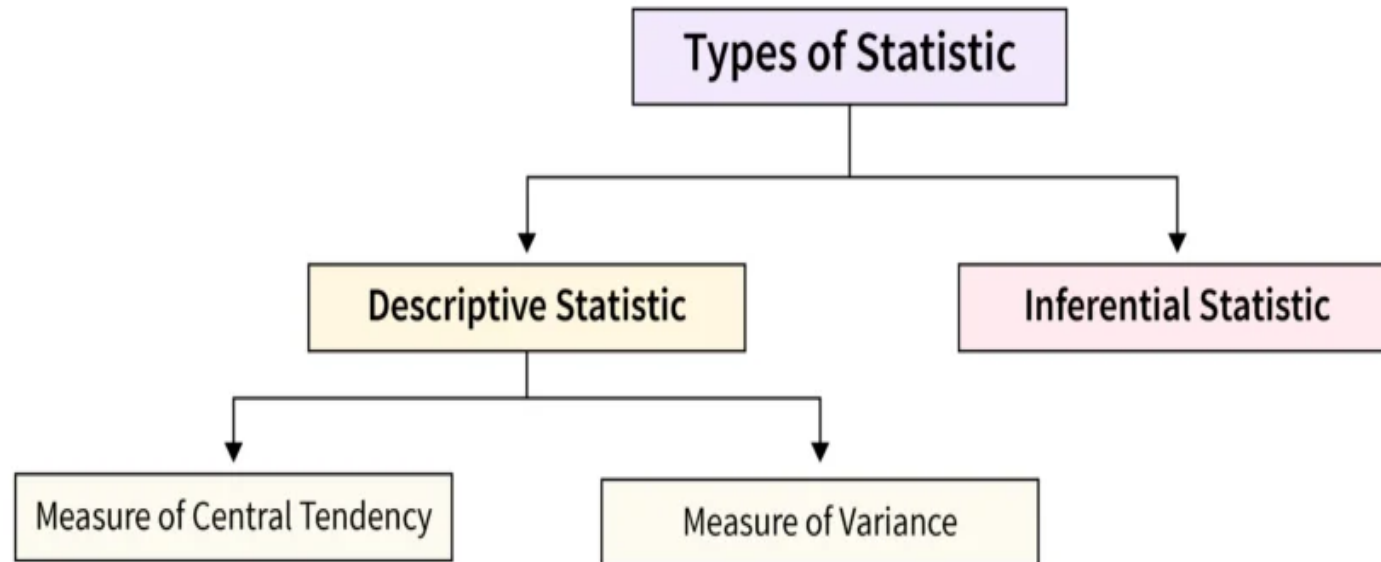
The seven basic Statistics Concepts for Data Science.

1. Descriptive Statistics
2. Variability
3. Correlation
4. Probability Distribution
5. Regression
6. Normal Distribution
7. Bias

These were some of the **statistics concepts for data science** that you need to work on. Apart from these, there are some other **statistics topics for data science** as well which includes:

- Central limit theorem
- Bias / variance tradeoff
- Hypothesis testing
- Relationship between variables
- Covariance





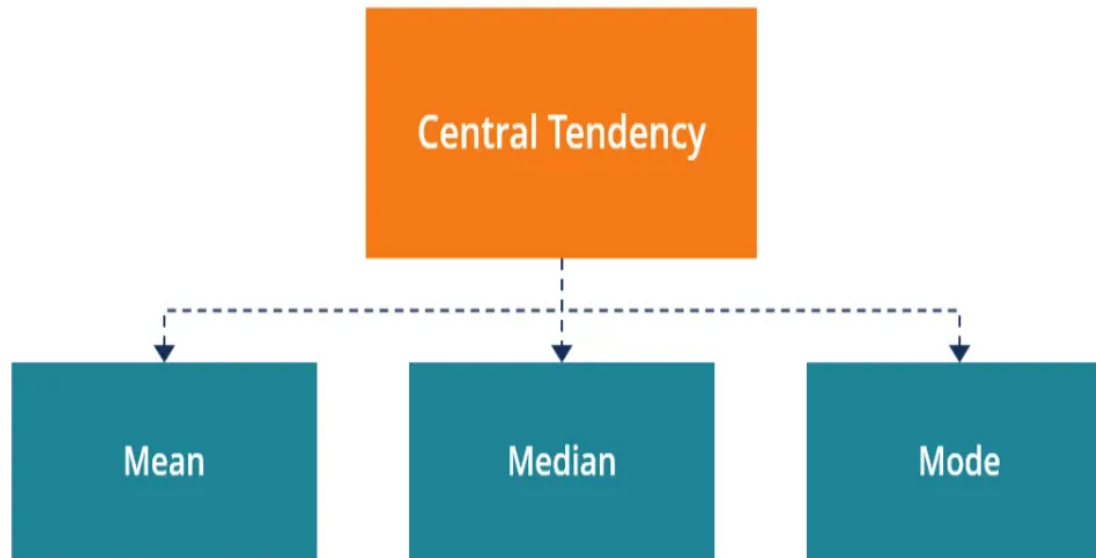
Basics of Statistics

Parameters	Definition	Formulas
Population Mean, (μ)	Entire group for which information is required.	$\sum x/N$
Sample Mean	Subset of population as entire population is too large to handle.	$\sum x/n$
Sample/Population Standard Deviation	Standard Deviation is a measure that shows how much variation from the mean exists.	$\sqrt{\frac{\sum (x - \bar{x})^2}{n-1}}$
Sample/Population Variance	Variance is the measure of spread of data along its central values.	Variance(Population) = $\frac{\sum (x - \bar{x})^2}{n}$ Variance(Sample) = $\frac{\sum (x - \bar{x})^2}{n-1}$
Class Interval(CI)	Class interval refers to the range of values assigned to a group of data points.	Class Interval = Upper Limit - Lower Limit
Frequency(f)	Number of time any particular value appears in a data set is called frequency of that value.	f is number of times any value comes in a article
Range, (R)	Range is the difference between the largest and smallest values of the data set	Range = (Largest Data Value - Smallest Data Value)

Central tendencies and Distributions :-

The central tendency measure is defined as the number used to represent the center or middle of a set of data values. The three commonly used measures of central tendency are the mean, median, and mode.

Central tendency is **a descriptive summary of a dataset through a single value that reflects the center of the data distribution**. Along with the variability (dispersion) of a dataset, central tendency is a branch of descriptive statistics.



•**Mean:** The mean can be calculated by summing all values present in the sample divided by total number of values present in the sample or population.

Formula:

$$\text{Mean}(\mu) = \frac{\text{Sum of Values}}{\text{Number of Values}} \quad \text{Mean}(\mu) = \frac{\text{Number of Values}}{\text{Sum of Values}} .$$

•**Median:** The median is the middle of a dataset when arranged from lowest to highest or highest to lowest in order to find the median, the data must be sorted. For an odd number of data points the median is the middle value and for an even number of data points median is the average of the two middle values.

- For odd number of data points:
Median = $(n+1)/2$
- For even number of data points:
Median = Average of $(n/2)$ th value and its next value

•**Mode:** The most frequently occurring value in the Sample or Population is called as Mode.



Given data set:

5, 2, 9, 2, 7, 3, 2

1. Mean (Average)

1. Formula: **Mean = (Sum of all values) / (Total number of values)**
2. Calculation: $(5+2+9+2+7+3+2)/7=30/7=4.29$
 $(5 + 2 + 9 + 2 + 7 + 3 + 2) / 7 = 30 / 7 = 4.29$
 $(5+2+9+2+7+3+2)/7=30/7=4.29$
3. **Mean = 4.29** (approximately)

2. Mode (Most Frequent Value)

1. The number that appears **most frequently** in the dataset.
2. Here, **2** appears **3 times** (more than any other number).
3. **Mode = 2**

3. Median (Middle Value)

1. Arrange the numbers in **ascending order**:
2, 2, 2, 3, 5, 7, 9
2. The middle value is **3** (since it is the 4th number in the ordered list).
3. **Median = 3**



```
import statistics
```

```
numbers = [5, 2, 9, 2, 7, 3, 2]
```

```
print("Mean:", statistics.mean(numbers))
```

```
print("Median:", statistics.median(numbers))
```

```
print("Mode:", statistics.mode(numbers))
```



A significant chunk of Data science is about understanding the behaviours and properties of variables, and this is not possible without knowing what distributions they belong to. Simply put, the probability distribution is a way to represent possible values a variable may take and their respective probability.

Binominal distribution -

- This distribution was discovered by a Swiss Mathematician James Bernoulli. It is used in such situation where an experiment results in two possibilities - success and failure.
- Binomial distribution is a discrete probability distribution which expresses the probability of one set of two alternatives-successes failure.
- A binomial distribution thus represents the probability for x successes in n trials, given a success probability p for each trial.
- A binomial distribution's expected value, or mean, is calculated by multiplying the number of trials (n) by the probability of successes (p), or $n \times p$.



The binomial distribution function is calculated as:

$$P_{(x:n,p)} = {}^nC_x p^x (1-p)^{n-x}$$

Where:

- n is the number of trials (occurrences)
- x is the number of successful trials
- p is the probability of success in a single trial
- nC_x is the combination of n and x..
- Note that ${}^nC_x = n! / x! (n - x)!$, where ! is factorial



```
import math

# Function to calculate binomial probability
def binomial_probability(n, x, p):
    # Binomial distribution formula:  $P(x : n, p) = C(n, x) * p^x * (1 - p)^{(n - x)}$ 

    return math.comb(n, x) * (p ** x) * ((1 - p) ** (n - x))

# Example usage
n = 10 # number of trials
p = 0.5 # probability of success (e.g., for a fair coin, p = 0.5)
x = 6 # number of successes

# Calculate binomial probability
prob = binomial_probability(n, x, p)

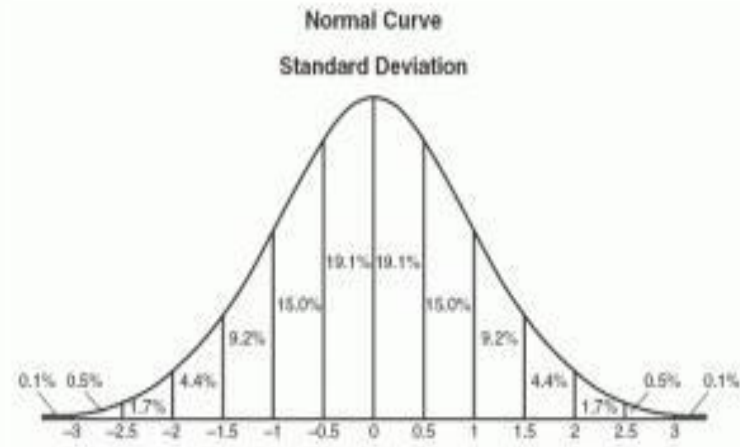
# Display the result
print(f'Probability of exactly {x} successes in {n} trials: {prob:.4f}')
```



Normal Distribution :-

A normal distribution is an arrangement of a data set in which most values cluster in the middle of the range and the rest taper off symmetrically toward either extreme. Height is one simple example of something that follows a normal distribution pattern: Most people are of average height the numbers of people that are taller and shorter than average are fairly equal and a very small (and still roughly equivalent) number of people are either extremely tall or extremely short. Here's an example of a normal distribution curve:

A graphical representation of a normal distribution is sometimes called a bell curve because of its flared shape. The precise shape can vary according to the distribution of the population but the peak is always in the middle and the curve is always symmetrical. In a normal distribution the mean mode and median are all the same.



$$y = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

μ is the mean of the distribution.

σ is the standard deviation.

$\pi \approx 3.14159$ is the mathematical constant pi.

$e \approx 2.71828$ is Euler's number, the base of the natural logarithm.

x is the variable of interest (often representing a data point).

Key Features of Normal Distribution

- **Symmetry:** The normal distribution is symmetric around its mean. This means the left side of the distribution mirrors the right side.
- **Mean, Median, and Mode:** In a normal distribution, the mean, median, and mode are all equal and located at the center of the distribution.
- **Bell-shaped Curve:** The curve is bell-shaped, indicating that most of the observations cluster around the central peak, and the probabilities for values further away from the mean taper off equally in both directions.



Measures of dispersion

Variance :-

Variance: Variance is a measure of how data points vary from the mean. Variance is the square deviation from the mean. It is denoted as „ σ^2 “.

In statistics, variance measures variability from the average or mean. It is calculated by taking the differences between each number in the data set and the mean, then squaring the differences to make them positive, and finally dividing the sum of the squares by the number of values in the data set.

Properties of Variance

- It is always non-negative since the variance sum is squared and therefore the result is either positive or zero.
- Variance always has squared units. For example, the variance of a set of weights estimated in kilograms will be given in kg squared



Example :

```
data = [2, 4, 6, 8, 10]
```

```
mean = sum(data) / len(data)
```

```
variance = sum((x - mean) ** 2 for x in data) / len(data) # Population  
variance
```

```
print("Variance:", variance)
```



Standard Deviation

Standard deviation is the measure of the distribution of statistical data. Standard Deviation is the square root of the variance. Standard deviation is denoted by the symbol, „ σ “.

Properties of Standard Deviation

☐ It describes the square root of the mean of the squares of all values in a data set and is also called the root-mean-square deviation.

☐ The smallest value of the standard deviation is 0 since it cannot be negative.

☐ When the data values of a group are similar, then the standard deviation will be very low or close to zero. But when the data values vary with each other, then the standard variation is high or far from zero.



Variance

Population

$$\sigma^2 = \frac{\sum_{i=1}^N (x_i - \mu)^2}{N}$$

Sample

$$s^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}$$

Standard deviation

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (x_i - \mu)^2}{N}}$$

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$$



Example: Let there be two cricket players: Pant and Kartik, and you have to select one for the cricket world cup. The score of both the players in the last five T-20 matches are as follows:

Kartik	Pant
23	34
28	85
45	02
59	15
63	77

Answer: Now, we will find the SD, and one who has the lesser value of SD will be more consistent.



Case -1: Kartik

Runs (x_i)	Squared Deviation $(x_i - \text{mean})^2$
23	$(23 - 43.6)^2$
28	$(28 - 43.6)^2$
45	$(45 - 43.6)^2$
59	$(59 - 43.6)^2$
63	$(63 - 43.6)^2$
Mean = $(23 + 38 + 45 + 59 + 63) / 5$ = 43.6	Sum of Squared Deviation = 1283.2



```
import math
```

```
data = [2, 4, 6, 8, 10]
```

```
mean = sum(data) / len(data)
```

```
variance = sum((x - mean) ** 2 for x in data) / len(data) # Population  
variance
```

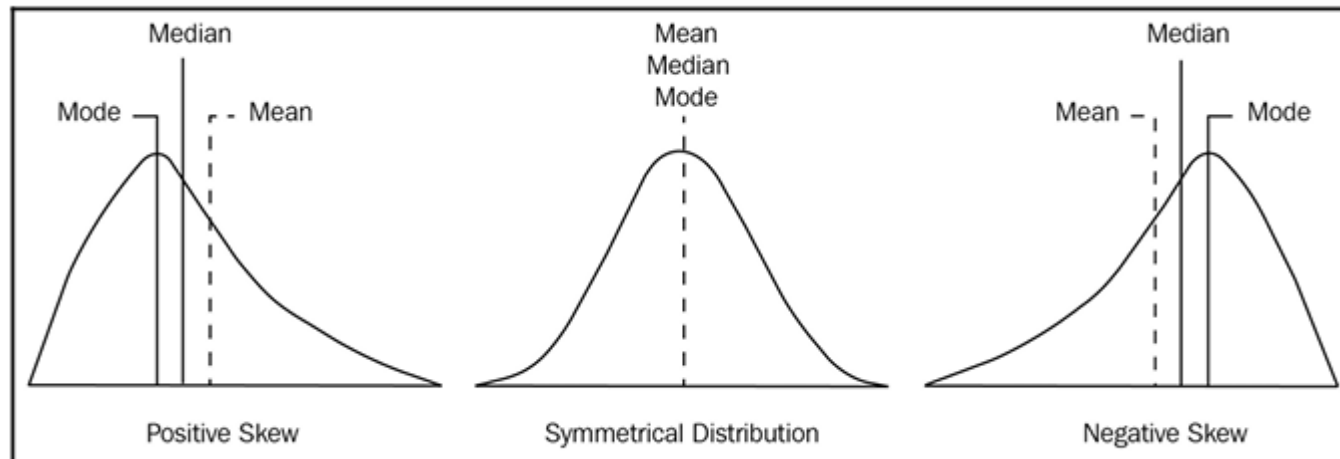
```
std_dev = math.sqrt(variance) # Standard deviation
```

```
print("Standard Deviation:", std_dev)
```



Skewness

Skewness measures the **asymmetry** of a data distribution. It tells us whether the data is **symmetrically distributed** or if it leans more to the left or right.



Symmetric (Zero Skewness)

The left and right sides of the distribution are mirror images.

$\text{Mean} \approx \text{Median} \approx \text{Mode}$

Example: Normal distribution

Right-Skewed (Positive Skewness) The right tail is longer.

$\text{Mean} > \text{Median} > \text{Mode}$

Example: Income distribution (a few people earn much more than the majority).

Left-Skewed (Negative Skewness) The left tail is longer.

$\text{Mean} < \text{Median} < \text{Mode}$

Example: Age at retirement (most people retire at a certain age, but a few retire much earlier).

Central Limit Theorem

Central Limit Theorem, also known as the **CLT**, is a crucial pillar of statistics and machine learning. It is at the heart of hypothesis testing. In this tutorial, you will understand the concept of the CLT and its applications.

The CLT is a statistical theory states that - if you take a sufficiently large sample size from a population with a finite level of variance, the mean of all samples from that population will be roughly equal to the population mean.

The CLT has several applications. Look at the places where you can use it. Political/election polling is a great example of how you can use CLT. These polls are used to estimate the number of people who support a specific candidate. You may have seen these results with confidence intervals on news channels. The CLT aids in this calculation.

The CLT use in various census fields to calculate various population details, such as family income, electricity consumption, individual salaries, and so on . The CLT is useful in a variety of fields.



Central Limit Theorem Formula

$$Z = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}}$$

Sample Mean = Population Mean = μ

Sample Standard Deviation = $\frac{\text{Standard Deviation}}{n}$

OR

Sample Standard Deviation = $\frac{\sigma}{\sqrt{n}}$

Given: $\mu = 70$ kg, $\sigma = 15$ kg, $n = 50$

As per the Central Limit Theorem, the sample mean is equal to the population mean.

Hence, $\mu_{\bar{x}} = \mu = 70$ kg

Now, $\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}} = 15/\sqrt{50}$

$\Rightarrow \sigma_{\bar{x}} \approx 2.1$ kg

Basic machine learning algorithms :-

Below is the list of Top 10 commonly used Machine Learning (ML) Algorithms:

- Linear regression.
- Logistic regression.
- Decision tree.
- SVM algorithm.
- Naive Bayes algorithm.
- KNN algorithm.
- K-means.
- Random forest algorithm.
- Dimensionality reduction algorithms
- Gradient boosting algorithm and AdaBoosting algorithm



Data Science Algorithms



01

Linear
Regression

02

Logistic
Regression

03

Decision Trees

04

Naive Bayes

05

KNN

06

Support Vector
Machine (SVM)

07

K-Means
Clustering

08

PCA

09

Neural Networks

10

Random Forests



Linear regression :-

The term regression is used when you try to find the relationship between variables.

In Machine Learning and in statistical modeling, that relationship is used to predict the outcome of events.

Linear regression uses the least square method.

The concept is to draw a line through all the plotted data points. The line is positioned in a way that it minimizes the distance to all of the data points.

The distance is called "residuals" or "errors".

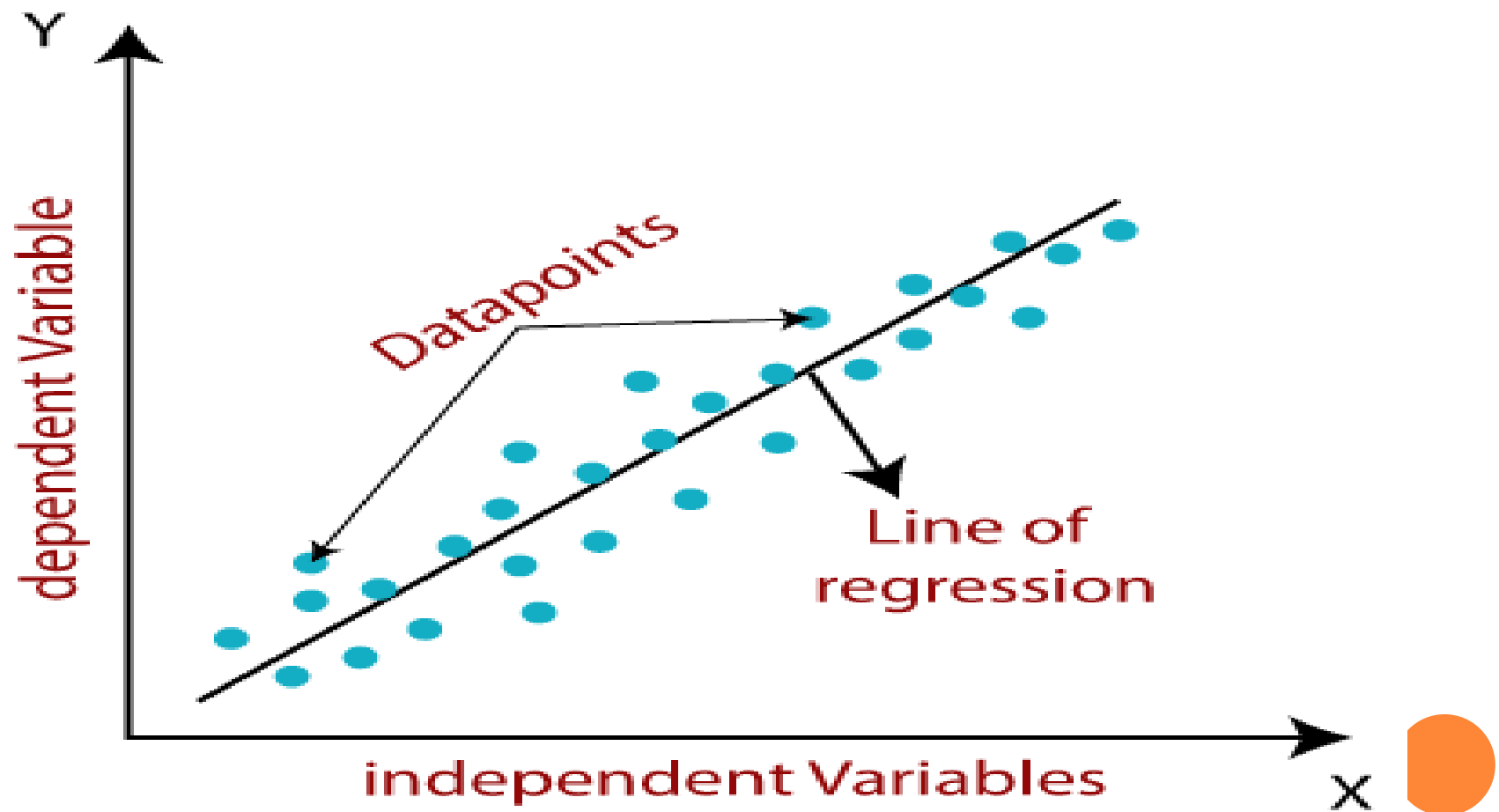
Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis.

Linear regression makes predictions for continuous/real or numeric variables such as **sales, salary, age, product price**, etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (x) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.



The linear regression model provides a sloped straight line representing the relationship between the variables. Consider the below image:



Mathematically, we can represent a linear regression as:

$$y = a_0 + a_1x + \varepsilon$$

Yes! That is the mathematical representation of a **simple linear regression** model, where:

- y = Dependent variable (the outcome or response)
- x = Independent variable (the predictor or feature)
- a_0 = Intercept (the value of y when $x = 0$)
- a_1 = Slope (how much y changes for a one-unit increase in x)
- ε = Error term (captures the variability in y that is not explained by x)

In multiple linear regression, where there are multiple independent variables x_1, x_2, \dots, x_n , the equation generalizes to:

$$y = a_0 + a_1x_1 + a \downarrow + \dots + a_nx_n + \varepsilon$$

Types of Linear Regression :

Linear regression can be further divided into **two types** of the algorithm:

1. **Simple Linear Regression:**

If a single independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression.

2. **Multiple Linear regression:**

If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.



SVM :-

Support Vector Machine(SVM) is a **supervised machine learning algorithm used for both classification and regression**. Though we say regression problems as well its best suited for classification. The objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points.

SVM or Support Vector Machine is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many practical problems. The idea of SVM is simple: The algorithm creates a line or a hyperplane which separates the data into classes.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

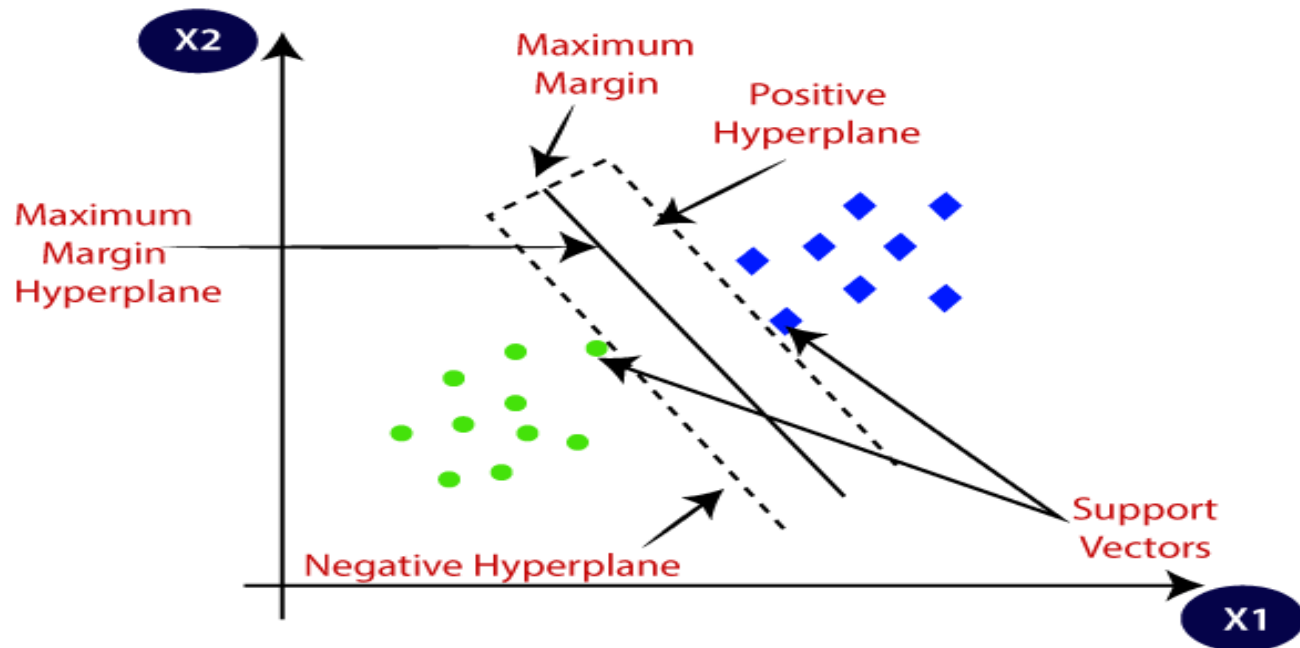
SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.



Types of SVM :

SVM can be of two types:

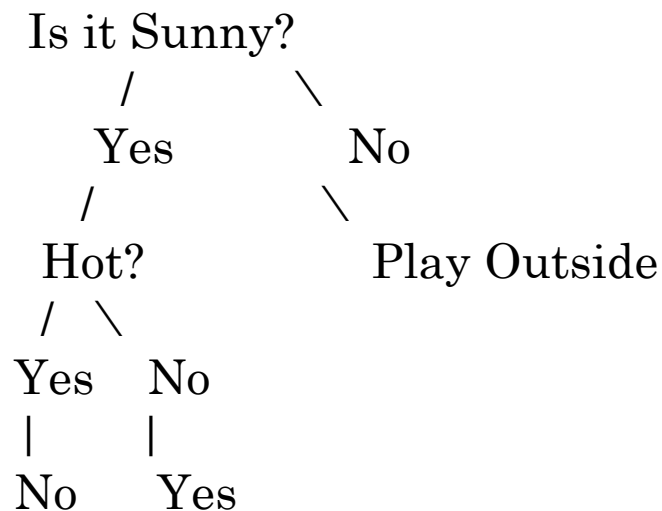
1. **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
2. **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.



DECISION TREE

○ Key Components of a Decision Tree

1. **Root Node:** The starting point that represents the entire dataset.
2. **Internal Nodes:** Represent feature-based decision points.
3. **Branches:** Show outcomes of decisions.
4. **Leaf Nodes:** Represent the final prediction/classification.



Naive Bayes :-

Naive Bayes is a probabilistic technique for constructing classifiers. The characteristic assumption of the naive Bayes classifier is to consider that the value of a particular feature is independent of the value of any other feature, given the class variable.

A Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter.



Bayes' Theorem

The Naïve Bayes classifier is based on **Bayes' Theorem**, which states:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Where:

- $P(A|B)$ = Probability of event A occurring given that B has occurred (posterior probability).
- $P(B|A)$ = Probability of event B occurring given that A is true (likelihood).
- $P(A)$ = Prior probability of event A.
- $P(B)$ = Prior probability of event B.

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

Naïve: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.

Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem.



Unit 4: Data visualization



Contents :-

- ☐ Introduction
- ☐ Types of data visualization
- ☐ Data for visualization: Data types
- ☐ Data encodings
- ☐ Retinal variables
- ☐ Mapping variables to encodings
- ☐ Visual encodings



Introduction :-

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

Data Visualization (DataViz) is the process of generating graphical representations of data for various purposes. These graphical representations are commonly known as plots or charts in data science terminology.

Understanding Data Visualization

Data visualization translates complex data sets into visual formats that are easier for the human brain to understand. This can include a variety of visual tools such as:


- **Charts:** Bar charts, line charts, pie charts, etc.
- **Graphs:** Scatter plots, histograms, etc.
- **Maps:** Geographic maps, heat maps, etc.
- **Dashboards:** Interactive platforms that combine multiple visualizations.




Why is Data Visualization Important?

Let's take an example. Suppose you compile data of the company's profits from 2013 to 2023 and create a line chart. It would be very easy to see the line going constantly up with a drop in just 2018. So you can observe in a second that the company has had continuous profits in all the years except a loss in 2018.


Importance of Data Visualization



Understanding
Complex Data



Improved
Decision
Making



Effective
Communication
of Insights



Identifying
Patterns
and Trends

1. Data Visualization Simplifies the Complex Data

Large and complex data sets can be challenging to understand. Data visualization helps break down complex information into simpler, visual formats making it easier

2. Enhances Data Interpretation

Visualization highlights patterns, trends, and correlations in data that might be missed in raw data form. This enhanced interpretation helps in making informed decisions.

3. Data Visualization Saves Time

It is definitely faster to gather some insights from the data using data visualization rather than just studying a chart.

4. Improves Communication

Visual representations of data make it easier to share findings with others especially those who may not have a technical background.



Advantages and Disadvantages of Data Visualization

Advantages

- There are many advantages of data visualization. Data visualization is used to:
- Communicate your results or findings with your audience
- Identify trends, patterns and correlations between variables
- Monitor the model's performance
- Clean data
- Validate the model's assumptions

Disadvantages

There are also some disadvantages of data visualization.

We need to download, install and configure software and open-source libraries. The process will be difficult and time-consuming for beginners.

Some data visualization tools are not available for free. We need to pay for those.

When we summarize the data, we'll lose the exact information.



Different Types of Analysis for Data Visualization

Some of the main data visualization techniques in data science are univariate analysis, bivariate analysis and multivariate analysis.

1. Univariate Analysis

In univariate analysis, as the name suggest, we analyze only one variable at a time. In other words, we analyze each variable separately. Bar charts, pie charts, box plots and histograms are common examples of univariate data visualization. Bar charts and pie charts are created for categorical variables, while box plots and histograms are created for numerical variables.

2. Bivariate Analysis

In bivariate analysis, we analyze two variables at a time. Often, we see whether there is a relationship between the two variables. The scatter plot is a classic example of bivariate data visualization.

3. Multivariate Analysis

In multivariate analysis, we analyze more than two variables simultaneously. The heatmap is a classic example of multivariate data visualization. Other examples are cluster analysis and principal component analysis (PCA).



Tools and Software for Data Visualization

There are multiple tools and software available for data visualization.

1. Python provides open-source libraries such as

Matplotlib

Seaborn

Plotty

Bokeh

Altair

2. R provides open-source libraries such as

Ggplot2

Lattice

3. Other data visualization libraries

IBM SPSS

Minitab

Matlab for data visualization

Tableau

Microsoft Power BI are popular among data scientists.



Data Visualization Process/Workflow

The data visualization process or workflow includes the following key steps.

1. Develop your research question
2. Get or create your data
3. Clean your data
4. Choose a chart type
5. Choose your tool
6. Prepare data
7. Create a chart



Examples of Data Visualization in Data Science

Here are some popular data visualization examples.

- **Weather reports:** Maps and other plot types are commonly used in weather reports.
- **Internet websites:** Social media analytics websites such as Social Blade and Google Analytics use data visualization techniques to analyze and compare the performance of websites.
- **Astronomy:** NASA uses advanced data visualization techniques in its reports and presentations.
- **Geography**
- **Gaming industry**



:-

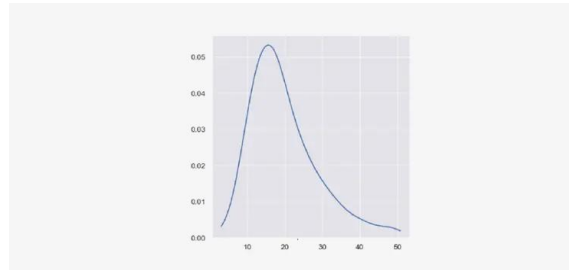
There are many data visualization types. The following are the commonly used data visualization charts.

1. Distribution plot
2. Box and whisker plot
3. Violin plot
4. Line plot
5. Bar plot
6. Scatter plot
7. Histogram
8. Pie chart
9. Area plot
10. Hexbin plot
11. Heatmap



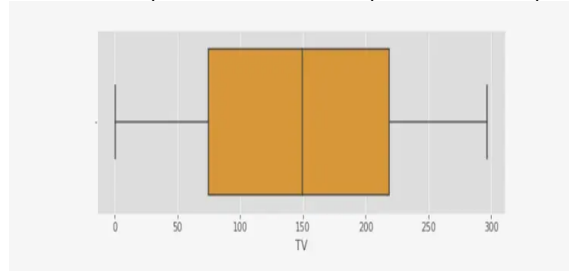
1. Distribution plot

A distribution plot, also known as a distplot. A distribution plot is used to visualize data distribution. Example: Probability distribution plot or density curve.



2. Box and whisker plot

This plot is used to plot the variation of the values of a numerical feature. You can get the values' minimum, maximum, median, lower and upper quartiles.

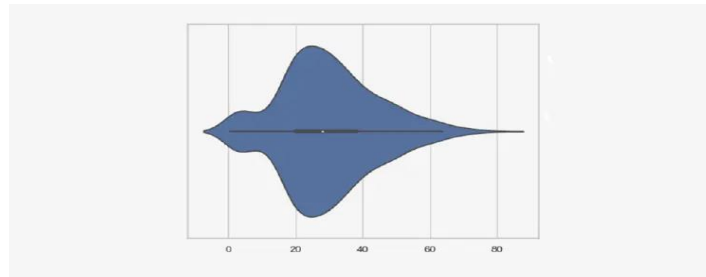


- Minimum: The lowest value in the dataset (excluding outliers).
- First Quartile (Q1): The 25th percentile of the data (the median of the lower half).
- Median (Q2): The 50th percentile of the data.
- Third Quartile (Q3): The 75th percentile of the data (the median of the upper half).
- Maximum: The highest value in the dataset (excluding outliers)



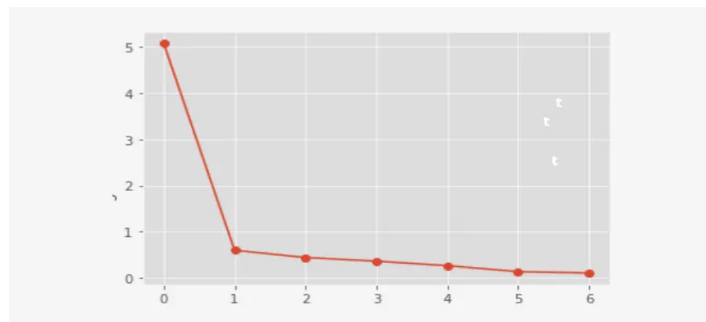
3. Violin plot

Similar to the box and whisker plot, the violin plot is used to plot the variation of a numerical feature. But it contains a kernel density curve in addition to the box plot. The kernel density curve estimates the underlying distribution of data.



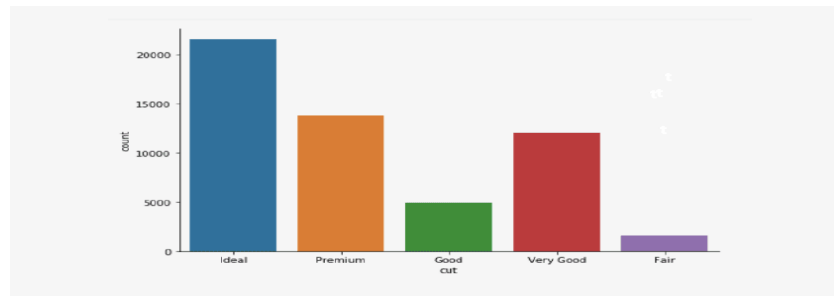
4. Line plot

A line plot is created by connecting a series of data points with straight lines. The number of periods is on the x-axis.



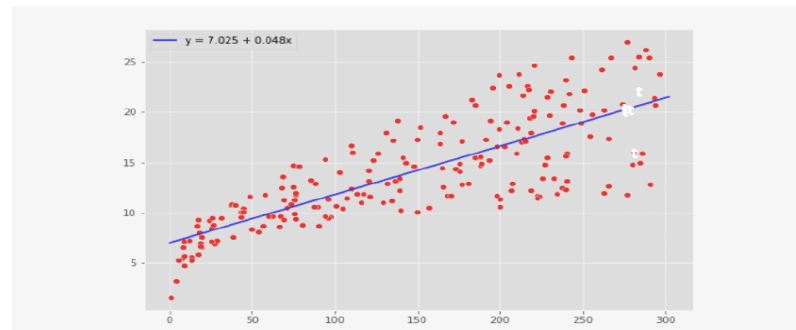
5. Bar plot

A bar plot is used to plot the frequency of occurring categorical data. Each category is represented by a bar. The bars can be created vertically or horizontally. Their heights or lengths are proportional to the values they represent.



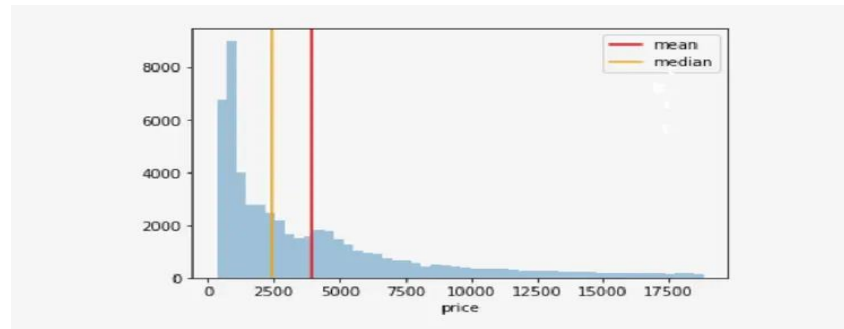
6. Scatter plot

Scatter plots are created to see whether there is a relationship (linear or non-linear and positive or negative) between two numerical variables. They are commonly used in regression analysis.



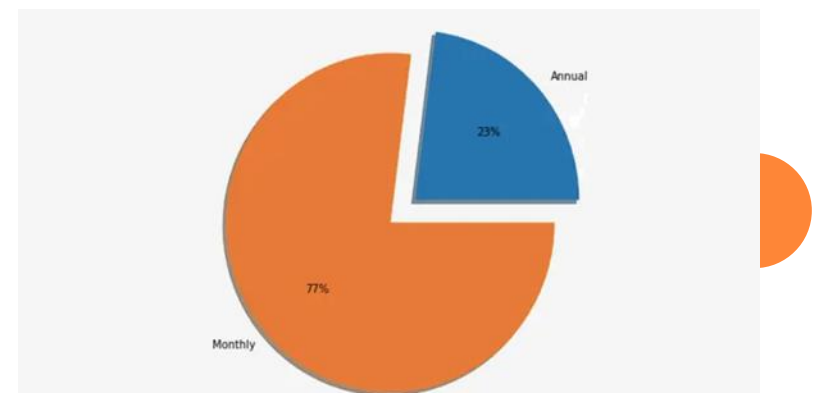
7. Histogram

A histogram represents the distribution of numerical data. Looking at a histogram, we can decide whether the values are normally distributed (a bell-shaped curve), skewed to the right or skewed left. A histogram of residuals is useful to validate important assumptions in regression analysis.



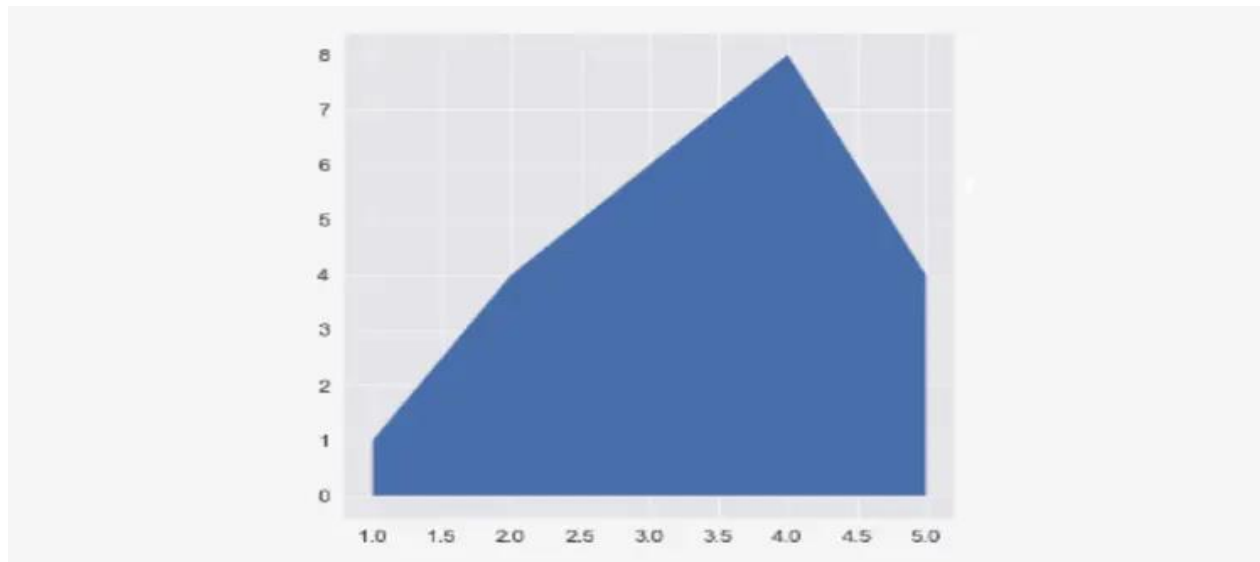
8. Pie chart

A categorical variable pie chart includes each category's values as slices whose sizes are proportional to the quantity they represent. It is a circular graph made with slices equal to the number of categories.



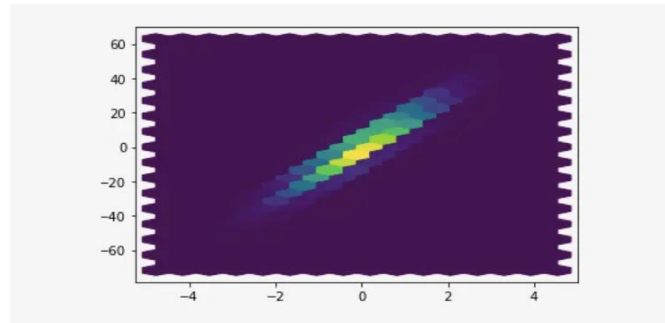
9. Area plot

The area plot is based on the line chart. We get the area plot when we cover the area between the line and the x-axis.



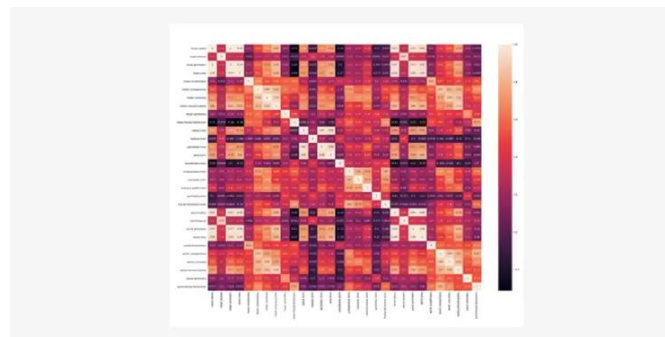
10. Hexbin plot

Similar to the scatter plot, a hexbin plot represents the relationship between two numerical variables. It is useful when there are a lot of data points in the two variables. When you have a lot of data points, they will overlap when represented in a scatter plot.



11. Heatmap

A heatmap visualizes the correlation coefficients of numerical features with a beautiful color map. Light colors show a high correlation, while dark colors show a low correlation. The heatmap is extremely useful for identifying multicollinearity that occurs when the input features are highly correlated with one or more of the other features in the dataset.



Data for visualization: Data types :-

Type of Data

consider the type of data you're working with: categorical, numerical, or time-based. Categorical data works well with bar or pie charts, while numerical data may be better visualized through histograms or scatter plots. Time-series data, like trends over months or years, is best represented by line graphs. Choosing the right chart based on data type is essential for clarity and insight.

Data visualization comes in two basic forms:
static visualization and interactive visualization.



Understanding Static Visualizations

static visualizations are images that offer a static view of the data in a non-usable format. They are ordinary and non complex and may commonly be found in reports, articles, presentations, and any printed or static electronic document.

Understanding Interactive Visualizations

Interactive visuals are different from other still images since users can interact with the information presented to them. They allow viewers to learn more and dig deeper into the attachments by hovering, clicking, filtering, and drilling down on the data set. Such as Salesforce, Exposure, tables, and charts, a dashboard is typically utilized in organizations or companies where people require up-to-date or individualized data regarding the organization



Aspect	Static Visualizations	Interactive Visualizations
Definition	Fixed visual representations of data, such as infographics or charts.	Dynamic visualizations that allow user interaction to explore data.
User Engagement	Limited engagement – users view the data as presented without interaction.	High engagement – users can manipulate data to uncover insights.
Data Exploration	Focused on a specific data story; users cannot explore beyond the presented view.	Enables exploration of multiple data stories through user actions.
Update Capability	Does not change unless edited by the creator; remains the same over time.	Can update in real-time based on user input or data changes.

Complexity	Generally simpler to create and understand; suitable for straightforward messages.	More complex to create; requires understanding of user interface design.
Use Cases	Ideal for reports, presentations, and print media where a single message is needed.	Best for dashboards, data analysis tools, and web applications where user interaction is beneficial.
Tools	Created using basic graphic design tools or software like Adobe Illustrator.	Developed using specialized tools like Tableau, D3.js, or Google Charts.
Accessibility	Easily accessible to all users without the need for technical skills.	May require some level of technical skill or familiarity with the interface.

Data encodings :-

Encoding in data viz basically means translating the data into a visual element on a chart/map/whatever you're making.

Encoding is the process of converting data into a format required for a number of information processing needs, including: Program compiling and execution.



Encoding Techniques

The data encoding technique is divided into the following types, depending upon the type of data conversion.

- **Analog data to Analog signals** – The modulation techniques such as Amplitude Modulation, Frequency Modulation and Phase Modulation of analog signals, fall under this category.
- **Analog data to Digital signals** – This process can be termed as digitization, which is done by Pulse Code Modulation PCM. Hence, it is nothing but digital modulation. As we have already discussed, sampling and quantization are the important factors in this. Delta Modulation gives a better output than PCM.
- **Digital data to Analog signals** – The modulation techniques such as Amplitude Shift Keying ASK, Frequency Shift Keying FSK, Phase Shift Keying PSK, etc., fall under this category. These will be discussed in subsequent chapters.
- **Digital data to Digital signals** – These are in this section. There are several ways to map digital data to digital signals.



Encoding Techniques Or Categorical encoding techniques in Data Science

There are many types of encoding techniques that can be used in data science depending on the nature and purpose of the data. Some of the common encoding techniques are detailed below.

One-hot Encoding

Label Encoding

Binary Encoding



One-hot encoding is a technique for handling categorical variables, which are variables that have a finite number of discrete values or categories. For example, gender, color, or country are categorical variables.

One-hot encoding converts each category into a binary vector of 0s and 1s, where only one element is 1 and the rest are 0. The length of the vector is equal to the number of categories. For example, if we have a variable color with three categories — red, green, and blue — we can encode it as follows:

Color	Red	Green	Blue
Red	1	0	0
Green	0	1	0
Blue	0	0	1



Label Encoding

Label encoding is another technique for encoding categorical variables, especially ordinal categorical variables, which are variables that have a natural order or ranking among their categories. For example, size, grade, or rating are ordinal categorical variables.

Label encoding assigns a numerical value to each category based on its order or rank. For example, if we have a variable size with four categories — small, medium, large, and extra large — we can encode it as follows:

Size	Label
Small	1
Medium	2
Large	3
Extra large	4



Binary Encoding

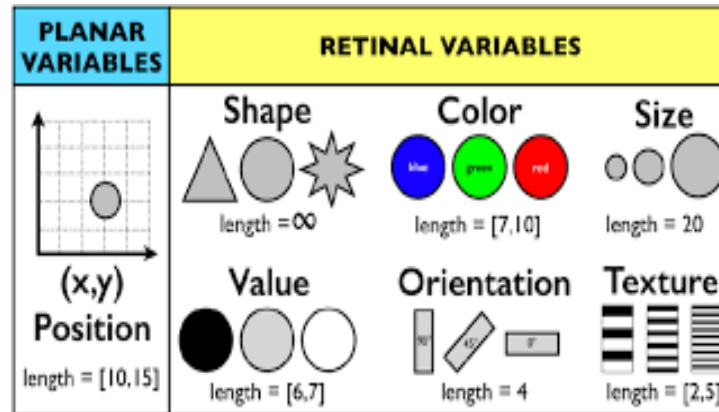
Binary encoding is a technique for encoding categorical variables with a large number of categories, which can pose a challenge for one-hot encoding or label encoding. Binary encoding converts each category into a binary code of 0s and 1s, where the length of the code is equal to the number of bits required to represent the number of categories. For example, if we have a variable country with 10 categories, we can encode it as follows:

Country	Binary Code
USA	0000
China	0001
India	0010
Brazil	0011
Russia	0100
Canada	0101
Germany	0110
France	0111
Japan	1000
Australia	1001



Retinal variables :-

The **retinal variables** are **size (length and area)**, **shape**, **texture**, **color**, **orientation (or slope)**, and **value**. Each variable can be classified using points, lines and areas.



Visual implantations need **retinal variables** to be encoded, and retinal variables take visual parameters.

For example, a point visual implantation can be encoded using the shape of a hollow circle and the colour blue. A line can be encoded using a solid pattern of thick size and green color. An area can be encoded using a 20% transparent red colour and thin line borders.











Key Retinal Variables

Originally defined by Bertin (1967), the main retinal variables include:

- 1.Size** – Variation in size can indicate quantity or magnitude (e.g., larger circles representing higher values in a bubble chart).
- 2.Shape** – Different shapes can be used to categorize distinct groups (e.g., triangles vs. squares in a scatter plot).
- 3.Color Hue** – Differences in color can represent categories or qualitative distinctions (e.g., red vs. blue for political affiliations).
- 4.Color Value (Brightness/Lightness)** – Lighter or darker shades can indicate intensity or magnitude (e.g., heat maps).
- 5.Texture** – Different patterns or densities can differentiate regions or categories (e.g., hatching in maps).
- 6.Orientation** – The direction of elements can be used to show contrast or patterns (e.g., slanted lines in bar charts).
- 7.Position** – The placement of elements in a visualization (e.g., x-y positioning in a scatter plot) is often the most effective variable for encoding numerical data.



Retinal Variable	Example Representation	Usage
Size	 (larger circles represent higher values)	Shows magnitude (e.g., bubble charts)
Shape	 (different symbols for categories)	Differentiates categories (e.g., scatter plots)
Color Hue	 (distinct colors for categories)	Categorical distinction (e.g., pie charts, maps)
Color Value (Brightness)	 (light to dark shades)	Shows intensity (e.g., heatmaps)
Texture	///// vs. 	Differentiates areas (e.g., maps, patterns)
Orientation	 vs. 	Shows direction or contrast (e.g., line charts)
Position	 (x-y plot positioning)	Used in scatter plots, maps, and line graphs

Mapping variables to encodings :-

Mapping of data types to encoding are:

Quantitative, ordinal and nominal are some of the types of data that we normally come across.

1 **Quantitative**: These are the data types that represent the quantity of certain data. Some attributes of this type includes position, length, volume, area etc.

2 **Ordinal**: These are the data types that holds data of some order. For example, days of the week, which holds the order in which they should be represented.

3 **Nominal**: In this kind of data types, the data is represented in the form of the names and categories.



Data Type	Best Retinal Variables (Encodings)	Example Visualizations
Categorical (Nominal) <i>(e.g., Countries, Species, Brands)</i>	<ul style="list-style-type: none"> - Shape (▲, ●, ■) - Color Hue (● ● ●) - Texture/Pattern (////, ▒▒▒▒) 	<ul style="list-style-type: none"> - Scatter Plots - Categorical Maps - Pie Charts
Ordinal <i>(e.g., Small/Medium/Large, Rankings)</i>	<ul style="list-style-type: none"> - Size (● ● ●) - Color Value (Light/Dark) (● ● ●) - Position (Y-axis) 	<ul style="list-style-type: none"> - Bar Charts - Heatmaps - Ordered Dot Plots
Quantitative (Numerical) <i>(e.g., Age, Sales, Temperature)</i>	<ul style="list-style-type: none"> - Size (Bigger = More) - Position (X, Y axes) - Color Value (Light/Dark for intensity) - Length (Bars/Lines) 	<ul style="list-style-type: none"> - Line Charts - Scatter Plots - Choropleth Maps



A map is **stored and analyzed in a data warehouse and in a single database**. The data is easily digested by business executives across the organization by using a BI tool to create data visualizations like charts, graphs, and dashboards.

Data mapping is a preliminary step to establish a cohesive data model for a data-driven organization. The data map provides instructions that align multiple data sets into a single configuration. A map is stored and analyzed in a data warehouse and in a single database.

The data is easily digested by business executives across the organization by using a BI tool to create data visualizations like charts, graphs, and dashboards.

Data mapping tools are commonly included in BI and analytics platforms. Be sure to choose a platform that includes a tool that adequately fulfills your organizational needs for customization and development. This will ensure that you get the most comprehensive, precise, and valuable results from your BI and analytics.



Target encoding is the method of converting a categorical value into the mean of the target variable. This type of encoding is a type of bayesian encoding method where bayesian encoders use target variables to encode the categorical value.

Encoding or continuization is **the transformation of categorical variables to binary or numerical counterparts**. An example is to treat male or female for gender as 1 or 0. Categorical variables must be encoded in many modeling methods (e.g., linear regression, SVM, neural networks).

The three popular techniques of converting Categorical values to Numeric values are done in two different methods. **Label Encoding. One Hot Encoding. Binary Encoding.**

In computers, encoding is the process of **putting a sequence of characters (letters, numbers, punctuation, and certain symbols) into a specialized format for efficient transmission or storage.**



Visual encodings :-

The visual encoding is **the way in which data is mapped into visual structures, upon which we build the images on a screen.** There are two types of visual encoding variables: planar and retinal. Humans are sensitive to the retinal variables.

“Sameness of a visual element implies sameness of what the visual element represents” is a saying about visual encoding. Visual encoding is the process of encoding images and visual sensory information. This means that people can convert the new information that they have stored into mental pictures. It is usually analyzed as combination of marks and channels showing abstract data dimensions.

The visual encoding is the way in which data is mapped into visual structures, upon which we build the images on a screen. The amygdala is a complex structure that has an important role in visual encoding. Amygdala is a part of the brain which understands the visual encoding patterns. It accepts visual input in addition to input from other systems and encodes the positive or negative values of conditioned stimuli.



Inputs for visual encoding:

The input for visual encoding can be of two models. 1. Mathematical model 2. Conceptual model

Mathematical model includes the raw data and the operations over the data,

Examples

- ✓ **Scatter plots, bar charts, and histograms** use numerical relationships.
- ✓ **Equations, statistics, and functions** define how data transforms.

whereas the **conceptual model** includes the semantics and their domain knowledge.

With the input from either of the models, certain relevant tasks are performed to deliver the output images using the visual encoding patterns that exists

Examples

- ✓ **Flowcharts, diagrams, network graphs**
- ✓ **Mind maps, organizational charts**



Types of visual encoding:

The visual encoding is broadly classified into 1. Retinal 2. Planar.

1 **Retinal:** Human beings are very sensitive to these kinds of retinal variables. Some of the retinal variables are colours, shapes, size and other kind of properties. Human beings can easily differentiate between these kinds of retinal variables.

2 **Planar:** Planar variables are another kind which can be applied to all types of data that are available



Visual encoding principles / variables:

- Position
- Size
- Orientation
- Color
- Shape
- Pattern
 - Texture
 - Focus
- Marks
- Channels
- Marks and channels
- Mark types
- Expressiveness
- Effectiveness



Elements of Visual Encoding Attribute :

Space

Space is the negative or positive area that an object or objects occupy in an area. Using simple principles we can control the relative position of every element. For example White space is used to control location of each and every element. Overlapping elements are used to control the position and values are used to control the relative positions.

Size

Size specifies how big or small objects are in relation to the space they occupy. The primary role size plays in design are given below:

- Function – For example, the age of the audience – older people would need type set larger to aid help in reading.
- Attractiveness is to add interest by cropping or scaling the elements.

Organization makes the important element the largest and the least important the smallest.

Texture

Texture is the look or feel of any object or surface. The appearance is either visual (illusionary) or tactile (physical to touch). Patterns are good examples of visual texture.



Unit 5

Technologies in Data Science



Contents :

- ☐ Computer science and engineering applications
- ☐ Data mining
- ☐ Network protocols
- ☐ Analysis of Web traffic
- ☐ Computer security
- ☐ Software engineering
- ☐ Computer architecture
- ☐ Operating systems
- ☐ Distributed systems
- ☐ Bioinformatics
- ☐ Machine learning



Computer science and engineering applications :-

- Artificial Intelligence
- Internet of Behaviours
- Robotic Process Automation
- Machine Learning
- 5G
- Virtual Reality -Augmented and Extended Reality
- Big Data Analytics



Artificial Intelligence

Artificial Intelligence Engineering is undoubtedly standing first in the list of top trending technology of Computer Science Engineering. AI refers to “man made thinking power” i.e when we try to incorporate the human intelligence into our machines thereby making them more smart and giving them an edge so that they can have human based skills like reasoning ,learning , problem solving and Decision making.AI is having a lot of practical applications like digital assistants like Siri, Alexa, Cortana in windows 10 , self driven cars ,expert systems designed for medical healthcare to name a few. This Technology has brought revolutionary changes in this century.

Internet of Behaviours

This is another trending technology when we talk of computer Science engineering, this aims at dealing with the basic entity DATA. IOB i.e. Internet of Behaviours refers to the use of the data to drive some specific behaviour patterns from the collection of the data. It aims at the gathering , combining and processing of the data from various sources like public domain, social media ,Government agencies etc.

For example, both Facebook and Google are using the behavioral data of their users to display advertisements to the people accordingly. This is helping businesses in getting connected with their potential audience .

Robotic Process Automation

RBA refers automation in the processes. The amount of human intervention is being reduced and the tasks are being replaced by bots. A lot of Coding needs to be done so as to enable the automation of computerised or non-computerised processes without human intervention e.g. automatic Email replies to automated data analysis and automatic processing and financial transactions approval. Robotic Automation of processes is very fast as it is programmed to do the automation work.

Machine Learning

This trending Technology cannot be given a miss in the list as it is empowering our machines to be smart thereby working on the learning models and improvising the process of decision making which is happening in the artificial devices whom we are making smart or intelligent .It is often called a subset of AI because it is aiming at making the device smart but through a learning model .

Learning can be of two types:- Supervised Learning and Unsupervised Learning.

Supervised Machine Learning is a type of machine learning technique that makes use of a supervisor.

Unsupervised Learning Method for machine learning focusses on sorted information that is grouped according to similarities and differences even though no categories are provided.

5G

Some of the Unique features of 5G that makes it worthwhile are High Speed, Massive Interconnections , Low latency and low power Consumption. Also when 5G is integrated with technologies like cloud Computing , IOT and Edge Computing , it can help businesses grow and can deliver significant economic advantages .

Virtual Reality -Augmented and Extended Reality

VR is technology that is generated with the help of the computers that give a real sense of surroundings , scenes and objects . Simulation is one technique that is used to achieve this task.

VR is used extensively in applications like gaming ,education , healthcare etc. This immersive technology has a lot of applications that are helping trainers as well students to embark on their learning journey.

Big Data Analytics

Big Data Analytics deals with the collection of data from different sources, merge it in a way that it becomes available to be consumed by personnels who will be analysing it later and finally deliver products useful to the businessess. The unstructured data that is raw data collected from various sources of the data is converted into a useful product for organisations. It is one of the trending and significant technology for businesses.

Data mining :-

Data mining is the process of sorting through large data sets to identify patterns and relationships that can help solve business problems through data analysis. Data mining techniques and tools enable enterprises to predict future trends and make more-informed business decisions.

Data mining is a key part of data analytics overall and one of the core disciplines in data science, which uses advanced analytics techniques to find useful information in data sets. At a more granular level, data mining is a step in the knowledge discovery in databases (KDD) process, a data science methodology for gathering, processing and analyzing data. Data mining and KDD are sometimes referred to interchangeably, but they're more commonly seen as distinct things.



Why is data mining important?

Data mining is a crucial component of successful analytics initiatives in organizations. The information it generates can be used in business intelligence (BI) and advanced analytics applications that involve analysis of historical data, as well as real-time analytics applications that examine streaming data as it's created or collected.

Effective data mining aids in various aspects of planning business strategies and managing operations. That includes customer-facing functions such as marketing, advertising, sales and customer support, plus manufacturing, supply chain management, finance and HR. Data mining supports fraud detection, risk management, cybersecurity planning and many other critical business use cases. It also plays an important role in healthcare, government, scientific research, mathematics, sports and more.



Data mining process:

The data mining process can be broken down into these four primary stages:

Data gathering.

Relevant data for an analytics application is identified and assembled. The data may be located in different source systems, a data warehouse or a data lake, an increasingly common repository in big data environments that contain a mix of structured and unstructured data. External data sources may also be used. Wherever the data comes from, a data scientist often moves it to a data lake for the remaining steps in the process.

Data preparation.

This stage includes a set of steps to get the data ready to be mined. It starts with data exploration, profiling and pre-processing, followed by data cleansing work to fix errors and other data quality issues. Data transformation is also done to make data sets consistent, unless a data scientist is looking to analyze unfiltered raw data for a particular application.

.



Mining the data.

Once the data is prepared, a data scientist chooses the appropriate data mining technique and then implements one or more algorithms to do the mining. In machine learning applications, the algorithms typically must be trained on sample data sets to look for the information being sought before they're run against the full set of data.

Data analysis and interpretation.

The data mining results are used to create analytical models that can help drive decision-making and other business actions. The data scientist or another member of a data science team also must communicate the findings to business executives and users, often through data visualization and the use of data storytelling techniques



Types of data mining techniques :-

- **Association rule mining.** In data mining, association rules are if-then statements that identify relationships between data elements.
- **Classification.** This approach assigns the elements in data sets to different categories defined as part of the data mining process. Decision trees, Naive Bayes classifiers, k-nearest neighbor and logistic regression are some examples of classification methods.
- **Clustering.** In this case, data elements that share particular characteristics are grouped together into clusters as part of data mining applications. Examples include k-means clustering, hierarchical clustering and Gaussian mixture models.
- **Regression.** This is another way to find relationships in data sets, by calculating predicted data values based on a set of variables. Linear regression and multivariate regression are examples. Decision trees and some other classification methods can be used to do regressions, too.
- **Sequence and path analysis.** Data can also be mined to look for patterns in which a particular set of events or values leads to later ones.
- **Neural networks.** A neural network is a set of algorithms that simulates the activity of the human brain. Neural networks are particularly useful in complex pattern recognition applications involving deep learning.

Benefits of data mining :-

- **More effective marketing and sales.** Data mining helps marketers better understand customer behavior and preferences, which enables them to create targeted marketing and advertising campaigns.
- **Better customer service.** Companies can identify potential customer service issues more promptly and give contact center agents up-to-date information to use in calls and online chats with customers.
- **Improved supply chain management.** Organizations can spot market trends and forecast product demand more accurately, enabling them to better manage inventories of goods and supplies.
- **Increased production uptime.** Mining operational data from sensors on manufacturing machines and other industrial equipment supports predictive maintenance applications .
- **Stronger risk management.** Risk managers and business executives can better assess financial, legal, cyber security and other risks to a company and develop plans for managing them.
- **Lower costs.** Data mining helps drive cost savings through operational efficiencies in business processes and reduced redundancy and waste in corporate spending.

Network protocols :-

A **network protocol** is an established set of rules that determine how data is transmitted between different devices in the same network.

Essentially, it allows connected devices to communicate with each other, regardless of any differences in their internal processes, structure or design.

Network protocols are the reason you can easily communicate with people all over the world, and thus play a critical role in modern digital communications.

Network protocols take large-scale processes and break them down into small, specific tasks or functions. This occurs at every level of the network, and each function must cooperate at each level to complete the larger task at hand. The term protocol suite refers to a set of smaller network protocols working in conjunction with each other.

Network protocols are typically created according to industry standard by various networking or information technology organizations.

There are three main types of network protocols. These include **network management protocols, network communication protocols and network security protocols:**



The following groups have defined and published different **network protocols**:

- The Institute of Electrical and Electronics Engineers (IEEE)
- The Internet Engineering Task Force (IETF)
- The International Organization for Standardization (ISO)
- The International Telecommunications Union (ITU)
- The World Wide Web Consortium (W3C)

While network protocol models generally work in similar ways, each protocol is unique and operates in the specific way detailed by the organization that created it.

There are thousands of different network protocols, but they all perform one of **three primary actions**:

- i. Communication
- ii. Network management
- iii. Security

Each type is necessary to use network devices swiftly and safely, and they work together to facilitate that usage.



Network Protocol Example :-

Here are a few examples of the most commonly used network protocols:

- **Hypertext Transfer Protocol (HTTP):** This Internet Protocol defines how data is transmitted over the internet and determines how web servers and browsers should respond to commands. This protocol (or its secure counterpart, HTTPS) appears at the beginning of various URLs or web addresses online.
- **Secure Socket Shell (SSH):** This protocol provides secure access to a computer, even if it's on an unsecured network. SSH is particularly useful for network administrators who need to manage different systems remotely.
- **Short Message Service (SMS):** This communications protocol was created to send and receive text messages over cellular networks. SMS refers exclusively to text-based messages. Pictures, videos or other media require Multimedia Messaging Service (MMS), an extension of the SMS protocol.

Network protocols do not simply define how devices and processes work; they define how devices and processes work together. Without these predetermined conventions and rules, the internet would lack the necessary infrastructure it needs to be functional and useable.

Network protocols are the foundation of modern communications, without which the digital world could not stand.

Types of network protocols :-

There are various types of protocols that support a major and compassionate role in communicating with different devices across the network. These are:

- Transmission Control Protocol (TCP)
- Internet Protocol (IP)
- User Datagram Protocol (UDP)
- Post office Protocol (POP)
- Simple mail transport Protocol (SMTP)
- File Transfer Protocol (FTP)
- Hyper Text Transfer Protocol (HTTP)
- Hyper Text Transfer Protocol Secure (HTTPS)
- Telnet
- Gopher



Some other popular protocols act as co-functioning protocols associated with these primary protocols for core functioning. These are:

- ARP (Address Resolution Protocol)
- DHCP (Dynamic Host Configuration Protocol)
- IMAP4 (Internet Message Access Protocol)
- SIP (Session Initiation Protocol)
- RTP (Real-Time Transport Protocol)
- RLP (Resource Location Protocol)
- RAP (Route Access Protocol)
- L2TP (Layer Two Tunnelling Protocol)
- PPTP (Point To Point Tunnelling Protocol)
- SNMP (Simple Network Management Protocol)
- TFTP (Trivial File Transfer Protocol)



Analysis of Web traffic :-

Web Analytics is a technique that you can employ to collect, measure, report, and analyze your website data. It is normally carried out to analyze the performance of a website and optimize its web usage.

Web Analytics is an indispensable technique for all those people who run their business online. This is a comprehensive tutorial that covers all the basics of web analytics.

Website traffic analysis is the process of collecting and interpreting key data points that describe web traffic to and from your site. (Web traffic is information about every user that visits your site.)

Web traffic analytics refers to collecting data about who comes to your website and what they do when they get there. That data is crucial to building effective sales and marketing strategies.

Web traffic analytics tells you who visits your website and what they do. Ideally, it'll tell you what content your users love and give you insights to help improve conversions.



Web traffic analysis breaks down data using specific metrics to organize that data and help you understand:

- Who's visiting your site
- How long they're on your site
- What they're doing while on your site
- Most-likely reasons they leave your site

When you're equipped with accurate and immediate website traffic data, it's possible to develop pattern models that identify potential weak points in your web design and inform ongoing development decisions.



Computer security :-

Computer security, also called **cyber security**, is the protection of computer systems and information from harm, theft, and unauthorized use.

Computer hardware is typically protected by the same means used to protect other valuable or sensitive equipment—namely, serial numbers, doors and locks, and alarms.

Computer Security (Cyber security) can be categorized into five distinct types:

- i. Critical infrastructure security.
- ii. Application security.
- iii. Network security.
- iv. Cloud security.
- v. Internet of Things (IoT) security.



Software engineering :-

software engineering focuses on the development of applications and features for users. A career in either data science or software engineering requires you to have programming skills.

Software engineering, on the other hand, is the process of developing software by systematically applying the principles of engineering. A software engineer analyzes user requirements, then designs, builds, and tests software applications if they fulfill the set requirements.

Software engineering serves as a foundation for understanding software in computer science and helps in the estimation of resources in economics. It employs management science for labor-intensive work. It's currently one of the most widely chosen careers worldwide.

Both Data Science and Software Engineering requires you to have programming skills. While Data Science includes statistics and Machine Learning, Software Engineering focuses more on coding languages. Both career choices are in demand and highly rewarding. Ultimately, it depends on your choice of interest.



Software Engineering is an engineering branch related to the evolution of software product using well-defined scientific principles, techniques, and procedures. The result of software engineering is an effective and reliable software product.



Software Engineering required :

Software Engineering is required due to the following reasons:

- To manage Large software
- For more Scalability
- Cost Management
- To manage the dynamic nature of software
- For better quality Management

Characteristics of a good software engineer :

The features that good software engineers should possess are as follows:

- Exposure to systematic methods, i.e., familiarity with software engineering principles.
- Good technical knowledge of the project range (Domain knowledge).
- Good programming abilities.
- Good communication skills. These skills comprise of oral, written, and interpersonal skills.
- High motivation.



Need of Software Engineering :

The necessity of software engineering appears because of a higher rate of progress in user requirements and the environment on which the program is working.

Huge Programming: It is simpler to manufacture a wall than to a house or building, similarly, as the measure of programming become extensive engineering has to step to give it a scientific process.

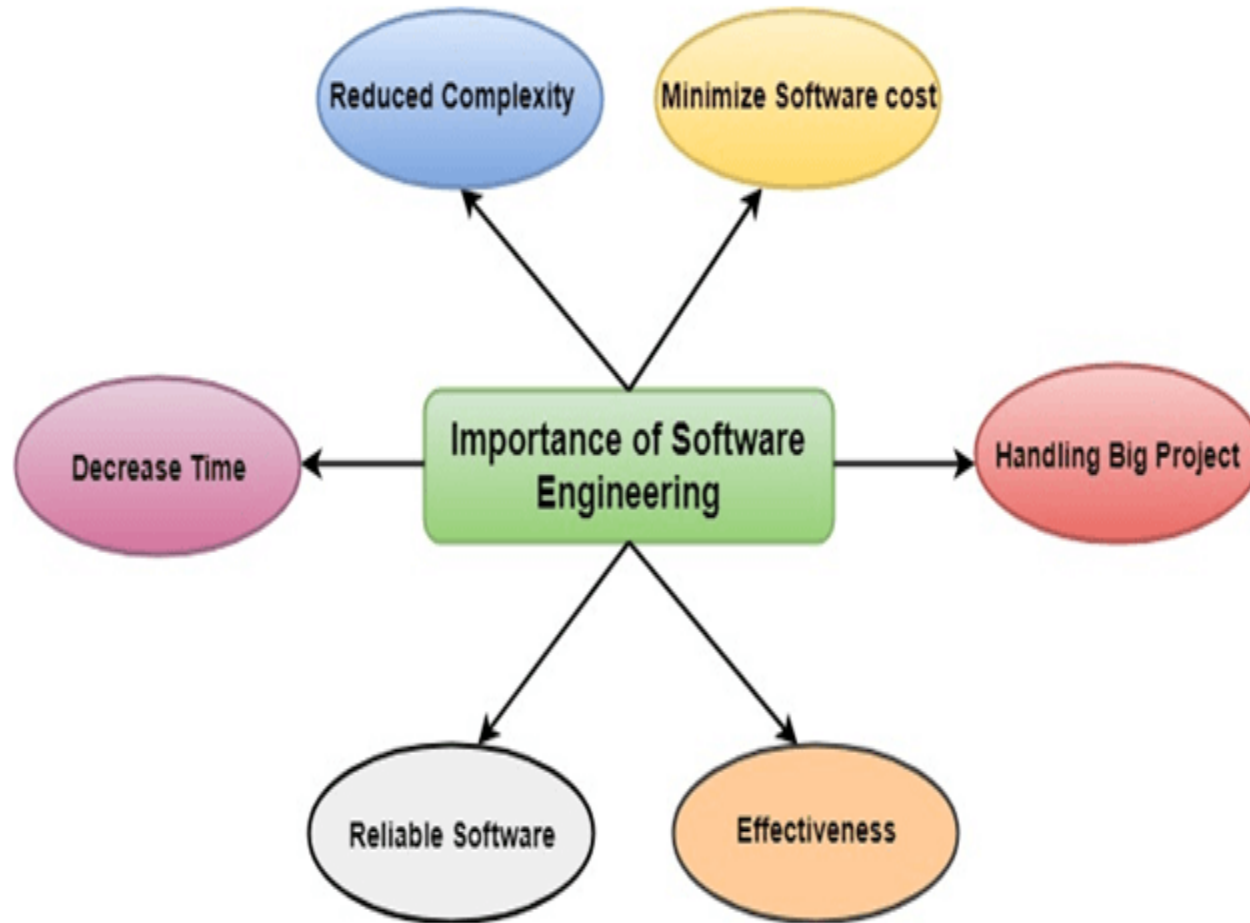
Adaptability: If the software procedure were not based on scientific and engineering ideas, it would be simpler to re-create new software than to scale an existing one.

Cost: As the hardware industry has demonstrated its skills and huge manufacturing has let down the cost of computer and electronic hardware. But the cost of programming remains high if the proper process is not adapted.

Dynamic Nature: The continually growing and adapting nature of programming hugely depends upon the environment in which the client works. If the quality of the software is continually changing, new upgrades need to be done in the existing one.

Quality Management: Better procedure of software development provides a better and quality software product.

Importance of Software Engineering :



1) Reduces complexity:

Big software is always complicated and challenging to progress. Software engineering has a great solution to reduce the complication of any project. Software engineering divides big problems into various small issues. And then start solving each small issue one by one. All these small problems are solved independently to each other.

2) To minimize software cost:

Software needs a lot of hardwork and software engineers are highly paid experts. A lot of manpower is required to develop software with a large number of codes. But in software engineering, programmers project everything and decrease all those things that are not needed. In turn, the cost for software productions becomes less as compared to any software that does not use software engineering method.

3) To decrease time:

Anything that is not made according to the project always wastes time. And if you are making great software, then you may need to run many codes to get the definitive running code. This is a very time-consuming procedure, and if it is not well handled, then this can take a lot of time. So if you are making your software according to the software engineering method, then it will decrease a lot of time.



4) Handling big projects:

Big projects are not done in a couple of days, and they need lots of patience, planning, and management. And to invest six and seven months of any company, it requires heaps of planning, direction, testing, and maintenance. No one can say that he has given four months of a company to the task, and the project is still in its first stage. Because the company has provided many resources to the plan and it should be completed. So to handle a big project without any problem, the company has to go for a software engineering method.

5) Reliable software:

Software should be secure, means if you have delivered the software, then it should work for at least its given time or subscription. And if any bugs come in the software, the company is responsible for solving all these bugs. Because in software engineering, testing and maintenance are given, so there is no worry of its reliability.

6) Effectiveness:

Effectiveness comes if anything has made according to the standards. Software standards are the big target of companies to make it more effective. So Software becomes more effective in the act with the help of software engineering.



Software Processes :

The term **software** specifies to the set of computer programs, procedures and associated documents (Flowcharts, manuals, etc.) that describe the program and how they are to be used.

A software process is the set of activities and associated outcome that produce a software product. Software engineers mostly carry out these activities. These are four key process activities, which are common to all software processes.

These activities are:

- **Software specifications:**
The functionality of the software and constraints on its operation must be defined.
- **Software development:**
The software to meet the requirement must be produced.
- **Software validation:**
The software must be validated to ensure that it does what the customer wants.
- **Software evolution:**
The software must evolve to meet changing client needs.



The Software Process Model :

A software process model is a specified definition of a software process, which is presented from a particular perspective. Models, by their nature, are a simplification, so a software process model is an abstraction of the actual process, which is being described. Process models may contain activities, which are part of the software process, software product, and the roles of people involved in software engineering.

Some examples of the types of software process models that may be produced are:

A workflow model: This shows the series of activities in the process along with their inputs, outputs and dependencies. The activities in this model perform human actions.

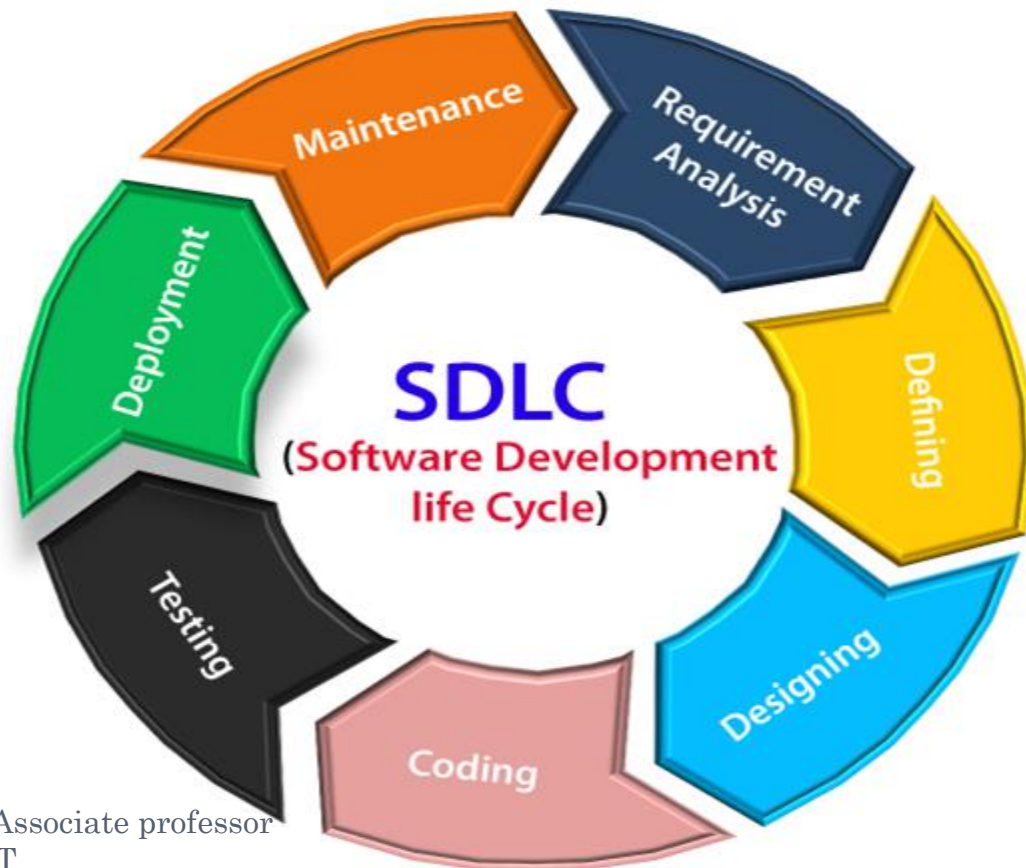
2. A dataflow or activity model: This represents the process as a set of activities, each of which carries out some data transformations. It shows how the input to the process, such as a specification is converted to an output such as a design. The activities here may be at a lower level than activities in a workflow model. They may perform transformations carried out by people or by computers.

3. A role/action model: This means the roles of the people involved in the software process and the activities for which they are responsible.



Software Development Life Cycle (SDLC) :

A software life cycle model (also termed process model) is a pictorial and diagrammatic representation of the software life cycle. A life cycle model represents all the methods required to make a software product transit through its life cycle stages. It also captures the structure in which these methods are to be undertaken



The stages of SDLC are as follows:

Stage1: Planning and requirement analysis

Stage2: Defining Requirements

Stage3: Designing the Software

Stage4: Developing the project

Stage5: Testing

Stage6: Deployment

Stage7: Maintenance



Stage1: Planning and requirement analysis

Requirement Analysis is the most important and necessary stage in SDLC.

The senior members of the team perform it with inputs from all the stakeholders and domain experts or SMEs in the industry.

Planning for the quality assurance requirements and identifications of the risks associated with the projects is also done at this stage.

Business analyst and Project organizer set up a meeting with the client to gather all the data like what the customer wants to build, who will be the end user, what is the objective of the product. Before creating a product, a core understanding or knowledge of the product is very necessary.

For Example,

A client wants to have an application which concerns money transactions. In this method, the requirement has to be precise like what kind of operations will be done, how it will be done, in which currency it will be done, etc.

Once the required function is done, an analysis is complete with auditing the feasibility of the growth of a product. In case of any ambiguity, a signal is set up for further discussion.

Once the requirement is understood, the SRS (Software Requirement Specification) document is created. The developers should thoroughly follow this document and also should be reviewed by the customer for future reference.

Stage2: Defining Requirements

Once the requirement analysis is done, the next stage is to certainly represent and document the software requirements and get them accepted from the project stakeholders.

This is accomplished through "SRS"- Software Requirement Specification document which contains all the product requirements to be constructed and developed during the project life cycle.

Stage3: Designing the Software

The next phase is about to bring down all the knowledge of requirements, analysis, and design of the software project. This phase is the product of the last two, like inputs from the customer and requirement gathering.

Stage4: Developing the project

In this phase of SDLC, the actual development begins, and the programming is built. The implementation of design begins concerning writing code. Developers have to follow the coding guidelines described by their management and programming tools like compilers, interpreters, debuggers, etc. are used to develop and implement the code.

Stage5: Testing

After the code is generated, it is tested against the requirements to make sure that the products are solving the needs addressed and gathered during the requirements stage. During this stage, unit testing, integration testing, system testing, acceptance testing are done.

Stage6: Deployment

Once the software is certified, and no bugs or errors are stated, then it is deployed. Then based on the assessment, the software may be released as it is or with suggested enhancement in the object segment.

After the software is deployed, then its maintenance begins.

Stage7: Maintenance

Once when the client starts using the developed systems, then the real issues come up and requirements to be solved from time to time.

This procedure where the care is taken for the developed product is known as maintenance.

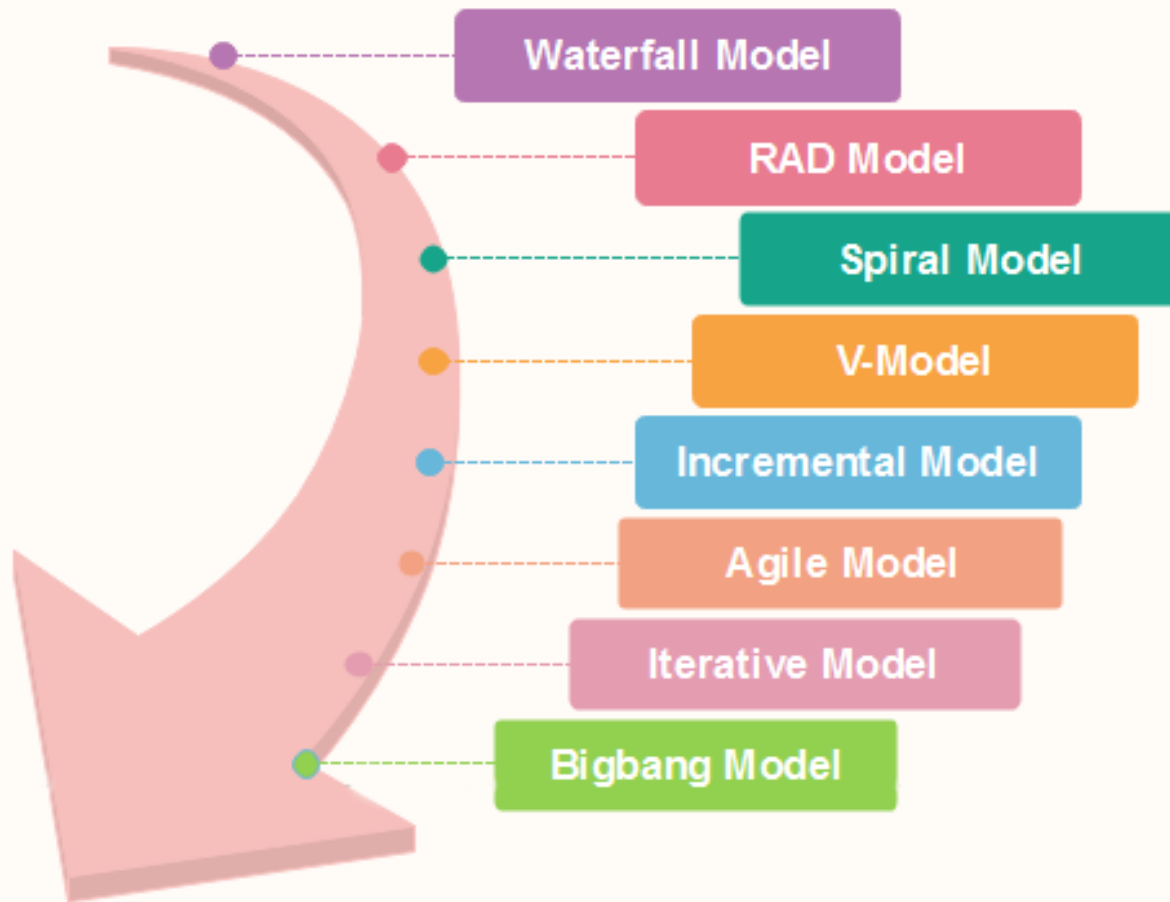
SDLC Models :

Software Development life cycle (SDLC) is a spiritual model used in project management that defines the stages include in an information system development project, from an initial feasibility study to the maintenance of the completed application.

There are different software development life cycle models specify and design, which are followed during the software development phase. These models are also called "**Software Development Process Models.**" Each process model follows a series of phase unique to its type to ensure success in the step of software development.



SDLC (Models)



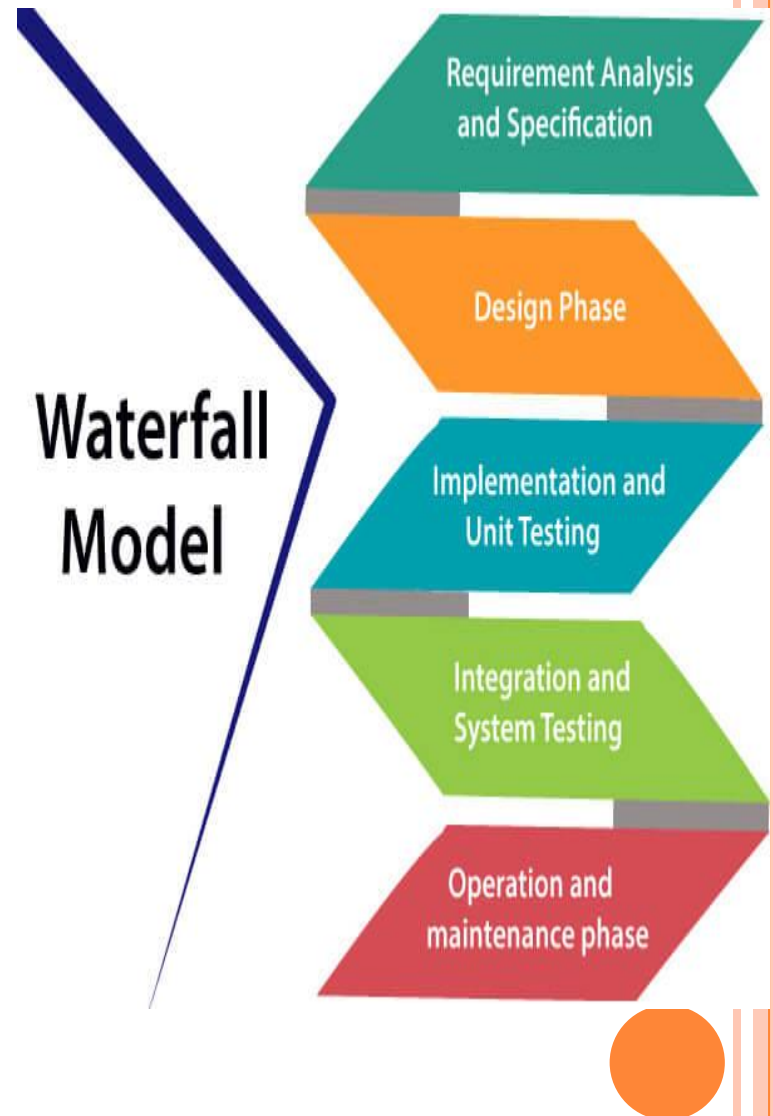
Waterfall Model

The waterfall is a universally accepted SDLC model. In this method, the whole process of software development is divided into various phases.

The waterfall model is a continuous software development model in which development is seen as flowing steadily downwards (like a waterfall) through the steps of requirements analysis, design, implementation, testing (validation), integration, and maintenance.

Linear ordering of activities has some significant consequences. First, to identify the end of a phase and the beginning of the next, some certification techniques have to be employed at the end of each step. Some verification and validation usually do this mean that will ensure that the output of the stage is consistent with its input (which is the output of the previous step), and that the output of the stage is consistent with the overall requirements of the system.

DR Mrs J. N. Jadhav Associate professor
Deptt of CSE, DYPCET



Advantages of Waterfall model -

This model is simple to implement also the number of resources that are required for it is minimal.

The requirements are simple and explicitly declared; they remain unchanged during the entire project development.

The start and end points for each phase is fixed, which makes it easy to cover progress.

The release date for the complete product, as well as its final cost, can be determined before development.

It gives easy to control and clarity for the customer due to a strict reporting system.

Disadvantages of Waterfall model -

In this model, the risk factor is higher, so this model is not suitable for more significant and complex projects.

This model cannot accept the changes in requirements during development.

It becomes tough to go back to the phase. For example, if the application has now shifted to the coding phase, and there is a change in requirement, It becomes tough to go back and change it.

Since the testing done at a later stage, it does not allow identifying the challenges and risks in the earlier phase, so the risk reduction strategy is difficult to prepare.

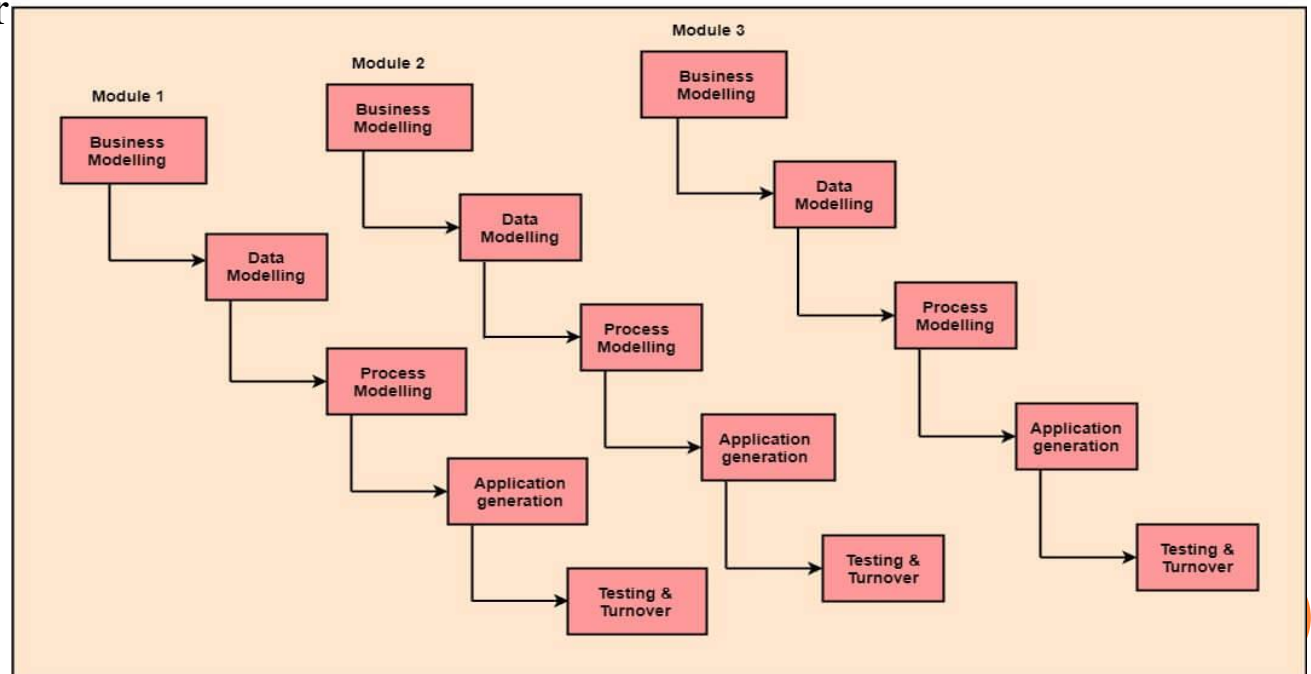


RAD Model

RAD or Rapid Application Development process is an adoption of the waterfall model; it targets developing software in a short period. The RAD model is based on the concept that a better system can be developed in lesser time by using focus groups to gather system requirements.

- Business Modeling
- Data Modeling
- Process Modeling
- Application Generation
- Testing and Turnover

Fig: RAD Model



Advantage of RAD Model -

This model is flexible for change.

In this model, changes are adoptable.

Each phase in RAD brings highest priority functionality to the customer.

It reduced development time.

It increases the reusability of features.

Disadvantage of RAD Model -

It required highly skilled designers.

All application is not compatible with RAD.

For smaller projects, we cannot use the RAD model.

On the high technical risk, it's not suitable.

Required user involvement.



Spiral Model

The spiral model is a **risk-driven process model**. This SDLC model helps the group to adopt elements of one or more process models like a waterfall, incremental, waterfall, etc. The spiral technique is a combination of rapid prototyping and concurrency in design and development activities.

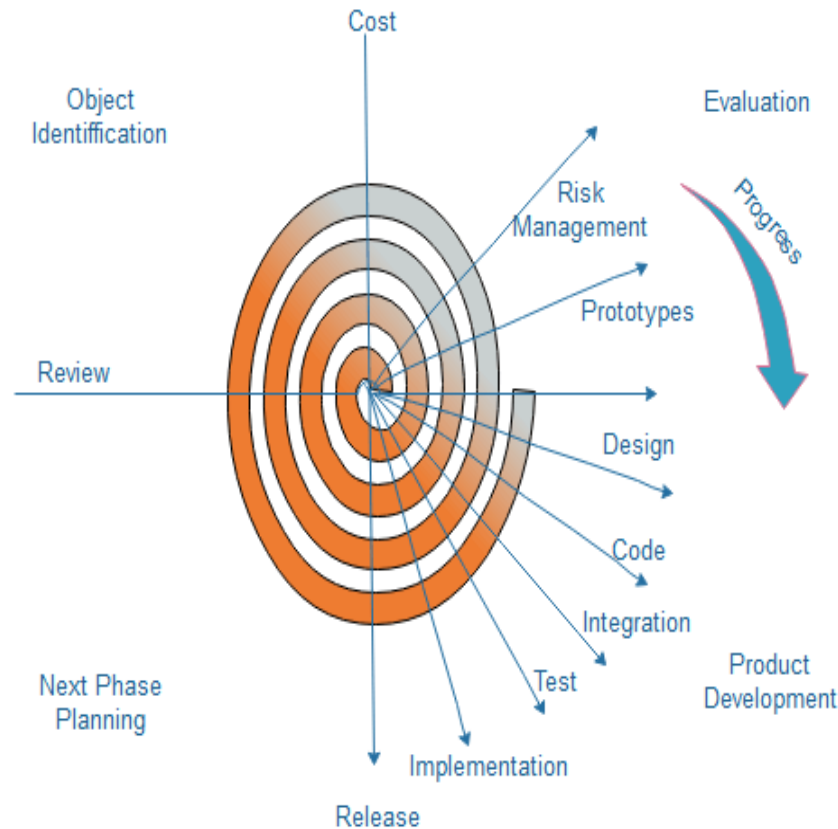


Fig. Spiral Model

Each cycle in the spiral begins with the identification of objectives for that cycle, the different alternatives that are possible for achieving the goals, and the constraints that exist. This is the first quadrant of the cycle (upper-left quadrant).

The next step in the cycle is to evaluate these different alternatives based on the objectives and constraints. The focus of evaluation in this step is based on the risk perception for the project.

The next step is to develop strategies that solve uncertainties and risks. This step may involve activities such as benchmarking, simulation, and prototyping.

Advantages -

High amount of risk analysis

Useful for large and mission-critical projects.

Disadvantages -

Can be a costly model to use.

Risk analysis needed highly particular expertise

Doesn't work well for smaller projects.



Computer architecture :-

Computer architectures represent the means of interconnectivity for a computer's hardware components as well as the mode of data transfer and processing exhibited.

Different computer architecture configurations have been developed to speed up the movement of data, allowing for increased data processing.

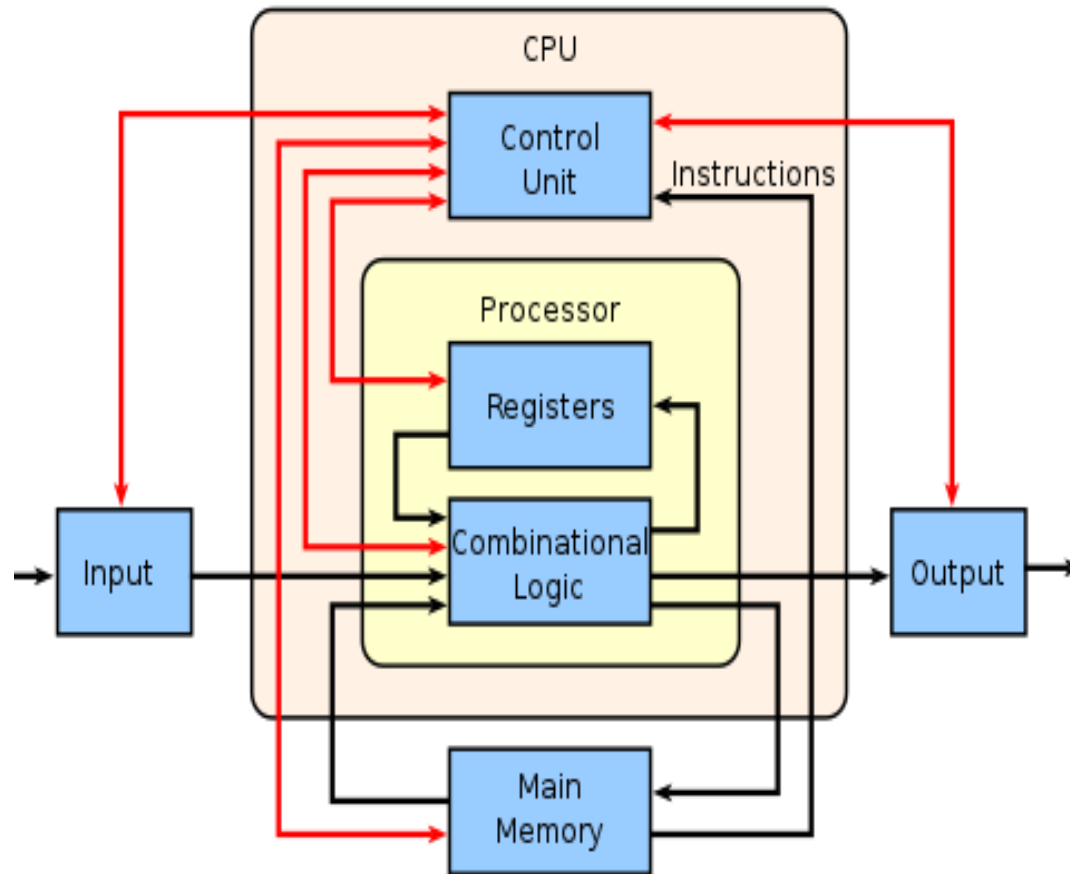
The developments in computer architecture, storage technology, networking, and software during the last several decades of the twentieth century coupled with the need to access and process information led to several large-scale distributed system developments.

Computer architecture is the organization of the components making up a computer system and the semantics or meaning of the operations that guide its function.

Early computer programs followed computer architecture, with data in one block of memory and program statements in another.

The first step in understanding any computer architecture is to learn its language. The words in a computer's language are called *instructions*.





Operating systems :-

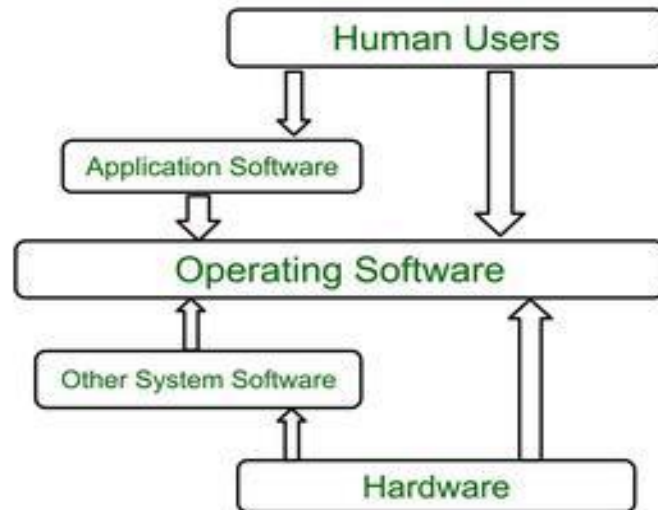
Operating System lies in the category of system software. It basically manages all the resources of the computer.

An operating system acts as an interface between the software and different parts of the computer or the computer hardware.

The operating system is designed in such a way that it can manage the overall resources and operations of the computer. It is a fully integrated set of specialized programs that handle all the operations of the computer.

It controls and monitors the execution of all other programs that reside in the computer, which also includes application programs and other system software of the computer.

Examples of operating system are Windows, Linux, Mac OS, etc.



Characteristics:

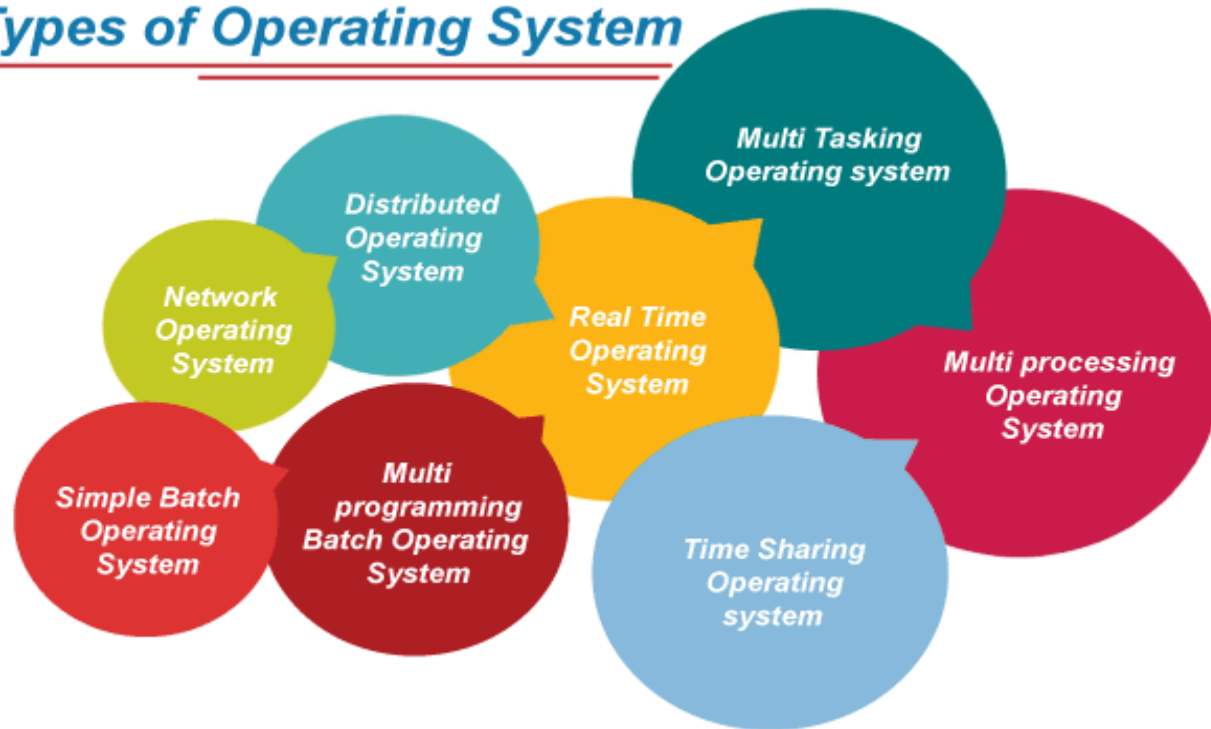
some of the important characteristic features of operating systems:

- **Device Management:** The operating system keeps track of all the devices. So, it is also called the Input / Output controller that decides which process gets the device, when, and for how much time.
- **File Management:** It allocates and de-allocates the resources and also decides who gets the resource.
- **Job Accounting:** It keeps the track of time and resources used by various jobs or users.
- **Error-detecting Aids:** It contains methods that include the production of dumps, traces, error messages, and other debugging and error-detecting methods.
- **Memory Management:** It keeps track of the primary memory, like what part of it is in use by whom, or what part is not in use, etc. and It also allocates the memory when a process or program requests it.
- **Processor Management:** It allocates the processor to a process and then de-allocates the processor when it is no longer required or the job is done.
- **Control on System Performance:** It records the delays between the request for a service and from the system.
- **Security:** It prevents unauthorized access to programs and data by means of passwords or some kind of protection technique.

Types of Operating Systems (OS) :

An operating system is a well-organized collection of programs that manages the computer hardware. It is a type of system software that is responsible for the smooth functioning of the computer system.

Types of Operating System



Batch Operating System :

In the 1970s, Batch processing was very popular. In this technique, similar types of jobs were batched together and executed in time. People were used to having a single computer which was called a mainframe.

In Batch operating system, access is given to more than one person; they submit their respective jobs to the system for the execution.

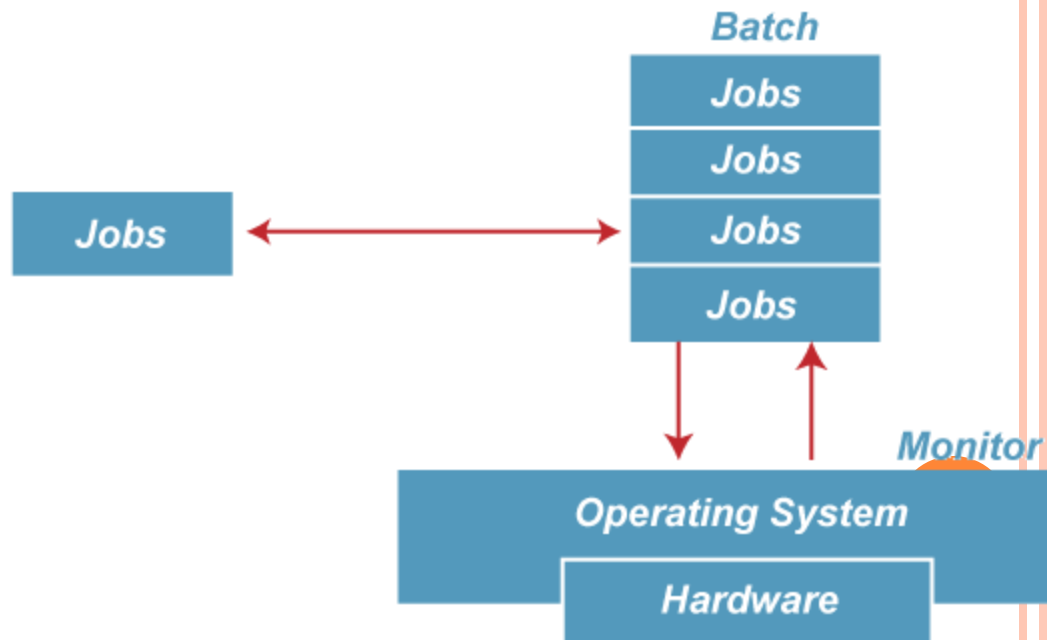
The system put all of the jobs in a queue on the basis of first come first serve and then executes the jobs one by one. The users collect their respective output when all the jobs get executed.

Advantages of Batch OS :-

- The use of a resident monitor improves computer efficiency as it eliminates CPU time between two jobs.

Disadvantages of Batch OS :-

1. Starvation
2. Not Interactive



Multiprogramming Operating System :

Multiprogramming is an extension to batch processing where the CPU is always kept busy. Each process needs two types of system time: CPU time and IO time.

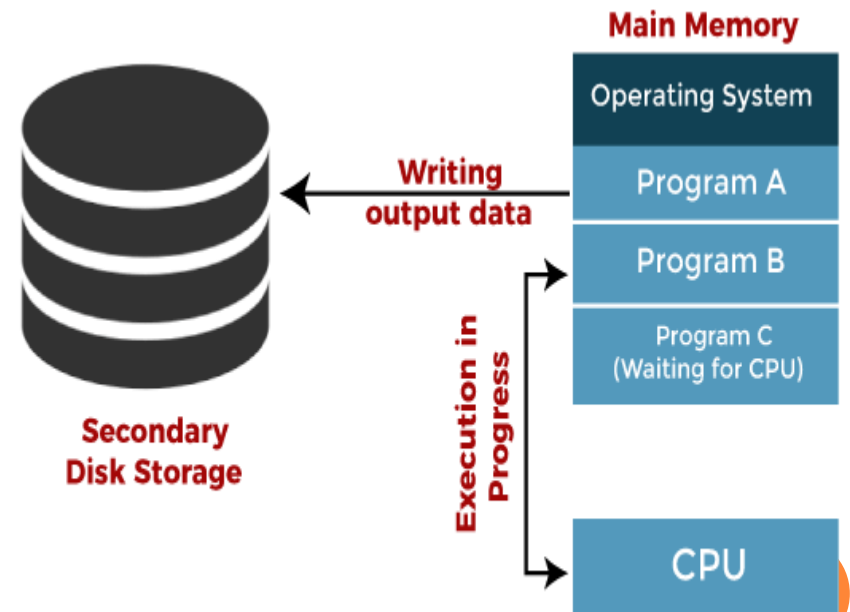
In a multiprogramming environment, when a process does its I/O, The CPU can start the execution of other processes. Therefore, multiprogramming improves the efficiency of the system.

Advantages of Multiprogramming OS :-

- Throughout the system, it increased as the CPU always had one program to execute.
- Response time can also be reduced.

Disadvantages of Multiprogramming OS :-

- Multiprogramming systems provide an environment in which various systems resources are used efficiently, but they do not provide any user interaction with the computer system.



Jobs in multiprogramming system

Multiprocessing Operating System :

In Multiprocessing, Parallel computing is achieved. There are more than one processors present in the system which can execute more than one process at the same time. This will increase the throughput of the system.

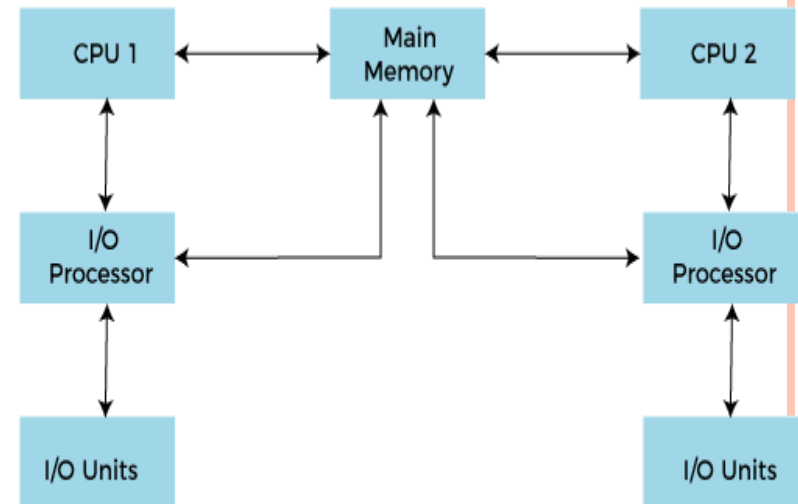
In Multiprocessing, Parallel computing is achieved. More than one processor present in the system can execute more than one process simultaneously, which will increase the throughput of the system.

Advantages of Multiprocessing OS:-

- Increased reliability
- Increased throughput

Disadvantages of Multiprocessing OS:-

- Multiprocessing operating system is more complex and sophisticated as it takes care of multiple CPUs simultaneously.



Working of Multiprocessor System



Multitasking Operating System :

The multitasking operating system is a logical extension of a multiprogramming system that enables **multiple** programs simultaneously. It allows a user to perform more than one computer task at the same time.

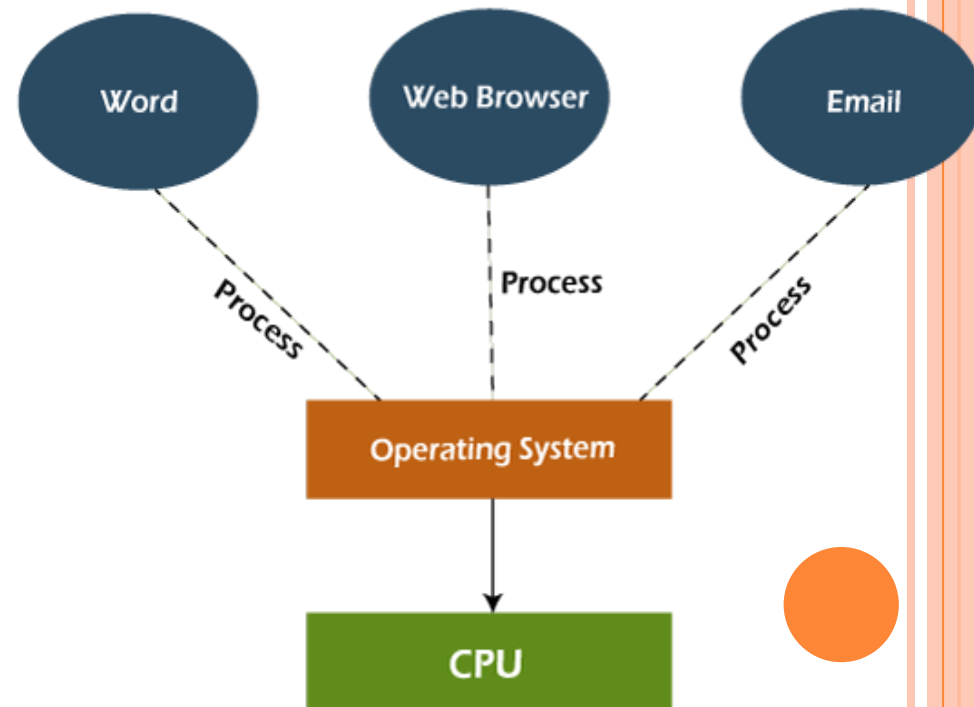
Types of Multitasking OS : Pre-emptive & Co-operative

Advantages of Multitasking OS :-

- This operating system is more suited to supporting multiple users simultaneously.
- The multitasking operating systems have well-defined memory management.

Disadvantages of Multitasking OS :-

- The multiple processors are busier at the same time to complete any task in a multitasking environment, so the CPU generates more heat.



Network Operating System :

An Operating system, which includes software and associated protocols to communicate with other computers via a network conveniently and cost-effectively, is called Network Operating System.

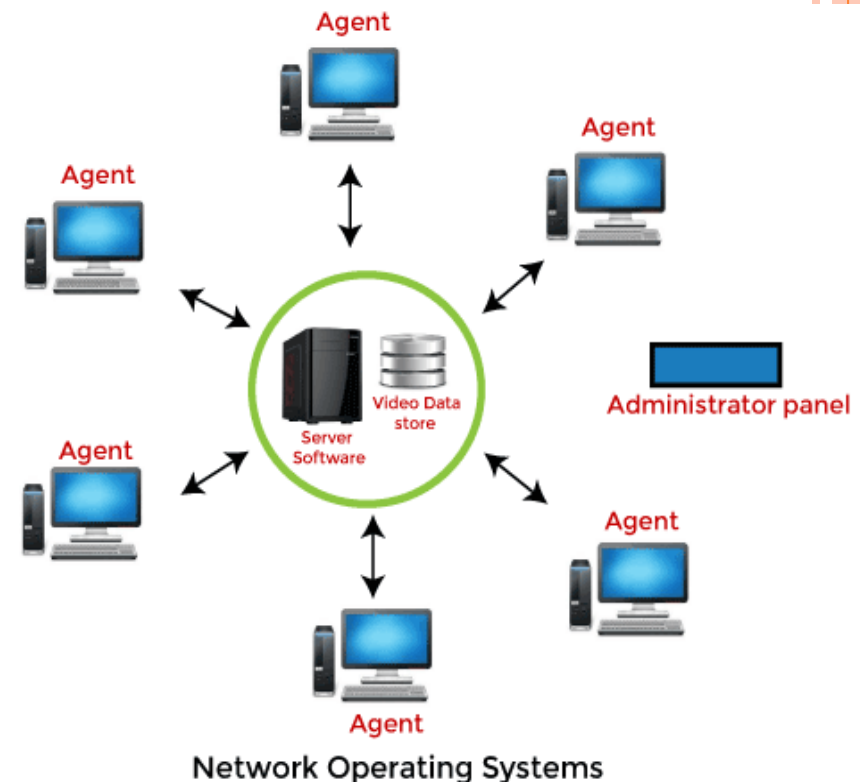
Types of Network OS : Peer-to-peer & Client-Server

Advantages of Network OS :

- In this type of operating system, network traffic reduces due to the division between clients and the server.
- This type of system is less expensive to set up and maintain.

Disadvantages of Network OS :

- In this type of operating system, the failure of any node in a system affects the whole system.
- Security and performance are important issues. So trained network administrators are required for network administration.



Real Time Operating System :

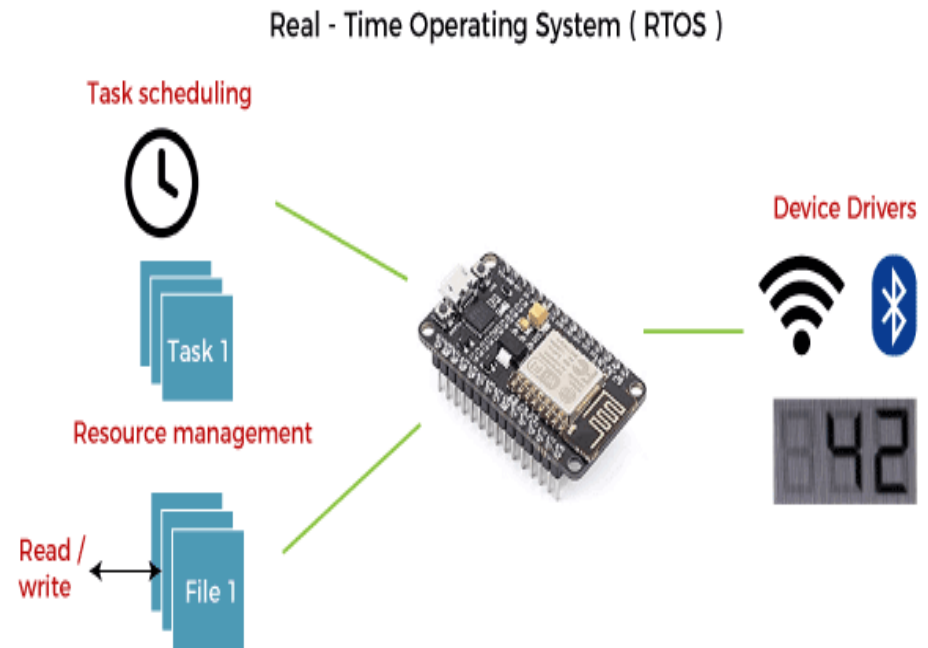
In Real-Time Systems, each job carries a certain deadline within which the job is supposed to be completed, otherwise, the huge loss will be there, or even if the result is produced, it will be completely useless.

Advantages of Real-time OS:

- Easy to layout, develop and execute real-time applications under the real-time operating system.
- In a Real-time operating system, the maximum utilization of devices and systems.

Disadvantages of Real-time OS:

- Real-time operating systems are very costly to develop.
- Real-time operating systems are very complex and can consume critical CPU cycles.



Time-Sharing Operating System :

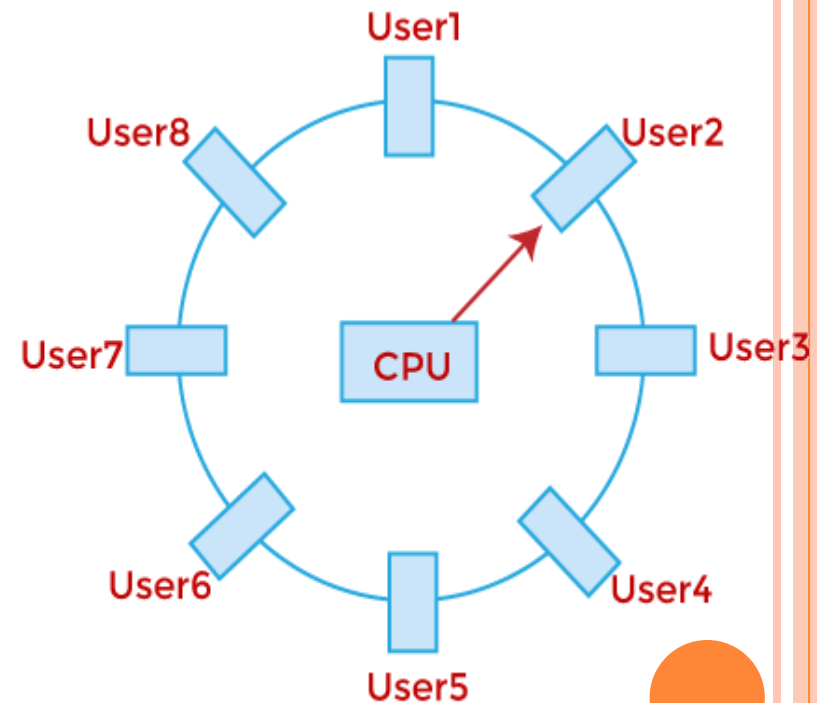
In the Time Sharing operating system, computer resources are allocated in a time-dependent fashion to several programs simultaneously. Thus it helps to provide a large number of user's direct access to the main computer. It is a logical extension of multiprogramming. In time-sharing, the CPU is switched among multiple programs given by different users on a scheduled basis.

Advantages of Time Sharing OS :

- The time-sharing operating system provides effective utilization and sharing of resources.
- This system reduces CPU idle and response time.

Disadvantages of Time Sharing OS :

- Data transmission rates are very high in comparison to other methods.
- Security and integrity of user programs loaded in memory and data need to be maintained as many users access the system at the same time.



Timesharing in case of 8 users

Distributed Operating System :

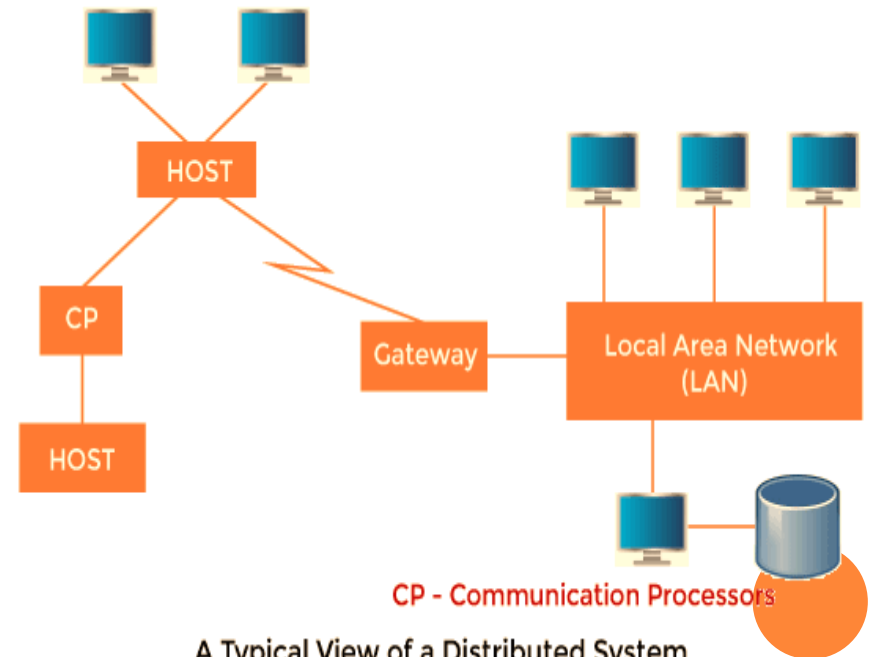
The Distributed Operating system is not installed on a single machine, it is divided into parts, and these parts are loaded on different machines. A part of the distributed Operating system is installed on each machine to make their communication possible. Distributed Operating systems are much more complex, large, and sophisticated than Network operating systems because they also have to take care of varying networking protocols.

Advantages of Distributed OS :

- The distributed operating system provides sharing of resources.
- This type of system is fault-tolerant.

Disadvantages of Distributed OS :

- Protocol overhead can dominate computation cost.

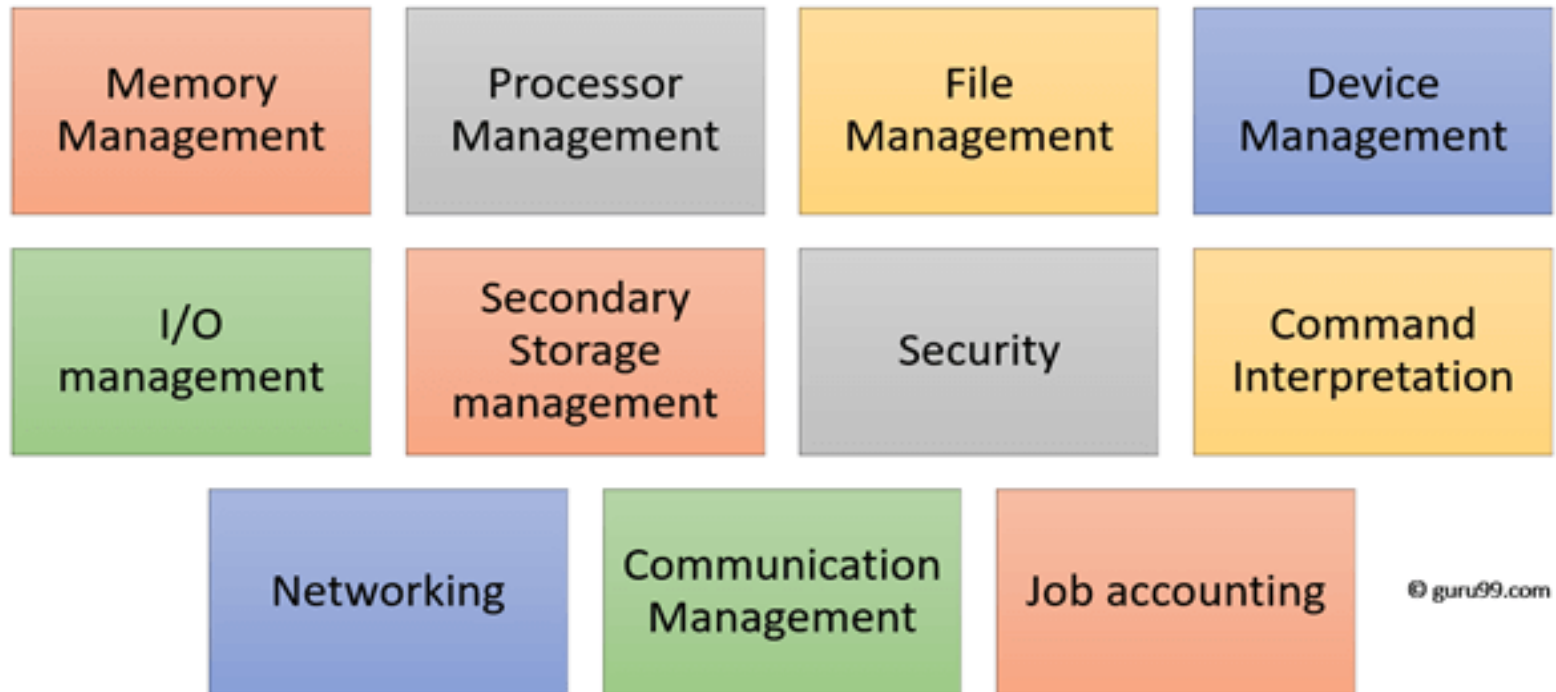


A Typical View of a Distributed System

Functions of Operating System :-

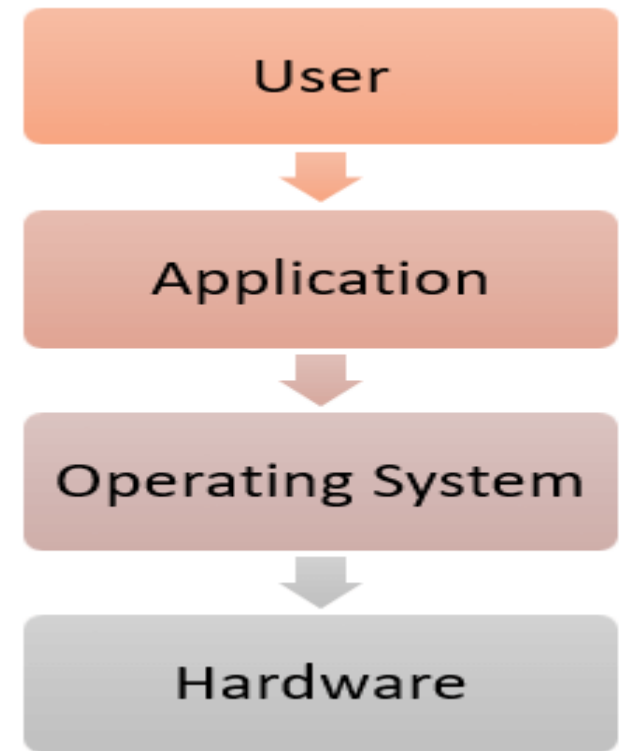
Some typical operating system functions may include managing memory, files, processes, I/O system & devices, security, etc.

Below are the **main functions** of Operating System:



Features of Operating System (OS) -

- Protected and supervisor mode
- Allows disk access and file systems Device drivers Networking Security
- Program Execution
- Memory management Virtual Memory Multitasking
- Handling I/O operations
- Manipulation of the file system
- Error Detection and handling
- Resource allocation
- Information and Resource Protection



Advantage of Operating System -

- Allows you to hide details of hardware by creating an abstraction.
- Easy to use with a GUI.
- Offers an environment in which a user may execute programs/applications.
- The operating system must make sure that the computer system convenient to use.
- Operating System acts as an intermediary among applications and the hardware components.
- It provides the computer system resources with easy to use format.
- Acts as an intermediary between all hardware's and software's of the system.

Disadvantages of Operating System –

- If any issue occurs in OS, you may lose all the contents which have been stored in your system.
- Operating system's software is quite expensive for small size organization which adds burden on them. Example Windows.
- It is never entirely secure as a threat can occur at any time.



Distributed systems :-

A **distributed system** is a computing environment in which various components are spread across multiple computers (or other computing devices) on a network.

Distributed computing is the method of making multiple computers work together to solve a common problem. It makes a computer network appear as a powerful single computer that provides large-scale resources to deal with complex challenges.

Distributed System is a collection of autonomous computer systems that are physically separated but are connected by a centralized computer network that is equipped with distributed system software. The autonomous computers will communicate among each system by sharing resources and files and performing the tasks assigned to them.

Example of Distributed System:

Any Social Media can have its Centralized Computer Network as its Headquarters and computer systems that can be accessed by any user and using their services will be the Autonomous Systems in the Distributed System Architecture.



Characteristics of Distributed System:

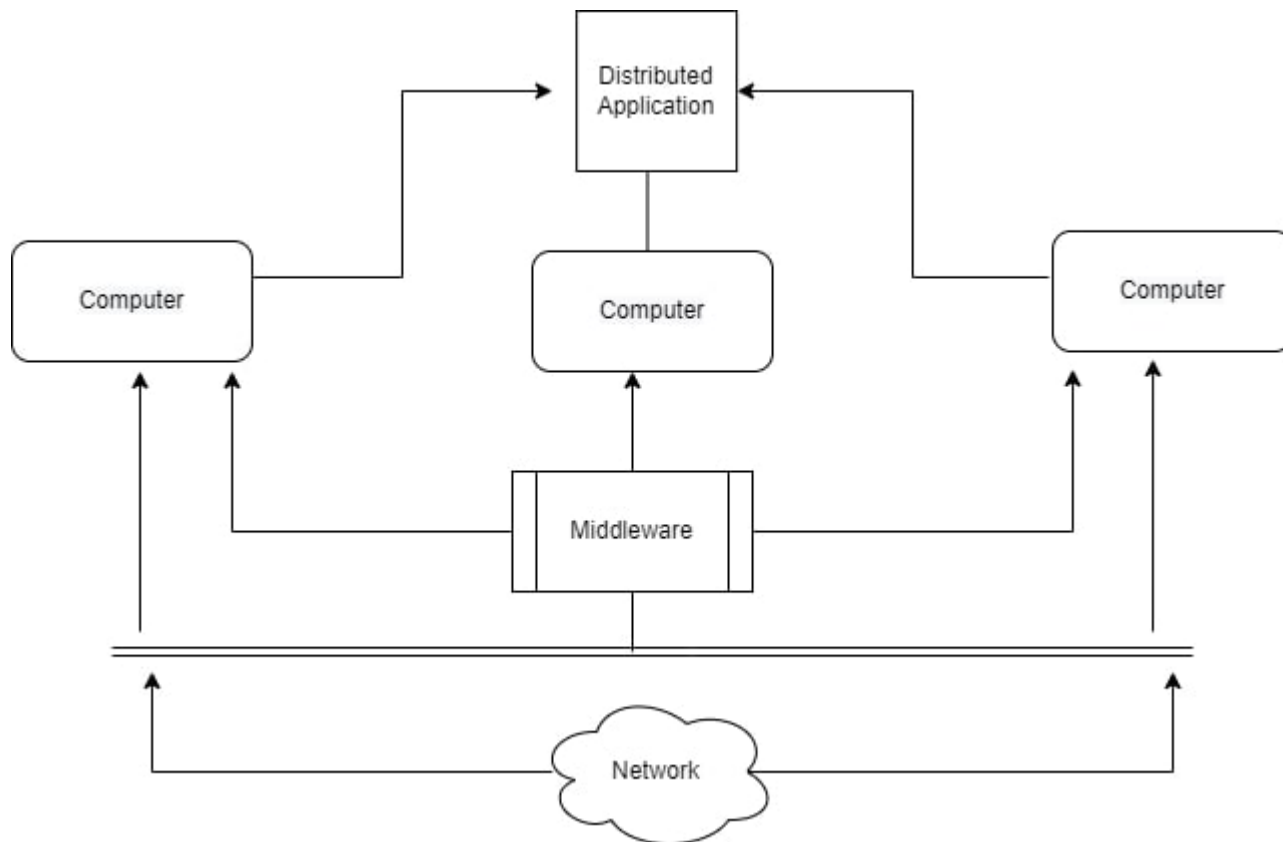
- **Resource Sharing:** It is the ability to use any Hardware, Software, or Data anywhere in the System.
- **Openness:** It is concerned with Extensions and improvements in the system (i.e., How openly the software is developed and shared with others)
- **Concurrency:** It is naturally present in the Distributed Systems, that deal with the same activity or functionality that can be performed by separate users who are in remote locations. Every local system has its independent Operating Systems and Resources.
- **Scalability:** It increases the scale of the system as a number of processors communicate with more users by accommodating to improve the responsiveness of the system.
- **Fault tolerance:** It cares about the reliability of the system if there is a failure in Hardware or Software, the system continues to operate properly without degrading the performance the system.
- **Transparency:** It hides the complexity of the Distributed Systems to the Users and Application programs as there should be privacy in every system.
- **Heterogeneity:** Networks, computer hardware, operating systems, programming languages, and developer implementations can all vary and differ among dispersed system components.

Advantages of Distributed System:

- Applications in Distributed Systems are Inherently Distributed Applications.
- Information in Distributed Systems is shared among geographically distributed users.
- Resource Sharing (Autonomous systems can share resources from remote locations).
- It has a better price performance ratio and flexibility.
- It has shorter response time and higher throughput.
- It has higher reliability and availability against component failure.
- It has extensibility so that systems can be extended in more remote locations and also incremental growth.

Disadvantages of Distributed System:

- Relevant Software for Distributed systems does not exist currently.
- Security possess a problem due to easy access to data as the resources are shared to multiple systems.
- Networking Saturation may cause a hurdle in data transfer i.e., if there is a lag in the network then the user will face a problem accessing data.
- In comparison to a single user system, the database associated with distributed systems is much more complex and challenging to manage.
- If every node in a distributed system tries to send data at once, the network may become overloaded.



Applications Area of Distributed System:

- **Finance and Commerce:** Amazon, eBay, Online Banking, E-Commerce websites.
- **Information Society:** Search Engines, Wikipedia, Social Networking, Cloud Computing.
- **Cloud Technologies:** AWS, Salesforce, Microsoft Azure, SAP.
- **Entertainment:** Online Gaming, Music, youtube.
- **Healthcare:** Online patient records, Health Informatics.
- **Education:** E-learning.
- **Transport and logistics:** GPS, Google Maps.
- **Environment Management:** Sensor technologies.



Bioinformatics :-

Bioinformatics is a field of data science that **focuses on analyzing biological data at the genomic and protein levels through software.**

The findings of bioinformatics can benefit health care, crops and biodiversity.

Bioinformatics focuses on parsing and analyzing biological data, while data science is a much broader field that can analyze data from any number of sources, like sales or financial markets.

Bioinformatics is a multidisciplinary field that utilizes computer programming, machine learning, algorithms, statistics, and other computational tools to organize and analyze large volumes of biological data.

Bioinformatics is a multidisciplinary field that utilizes computer programming, machine learning, algorithms, statistics, and other computational tools to organize and analyze large volumes of biological data. Fields of biology that generate massive amounts of data include genomics, transcriptomics, proteomics, and metabolomics.

Bioinformatics is to solve biological problems. This can range from genetics to biochemistry. The increasing necessity to process big data and develop algorithms in all fields of science mean that programming is becoming an essential skill for scientists, with **Python the language of choice for the majority of bioinformaticians.**

Bioinformatics Used For –

Bioinformatics entails the storage and management of biological data via the creation and maintenance of powerful databases, as well as the retrieval, analysis, and interpretation of data via algorithms and other computational tools. As such, it has applications for a wide range of fields.

Here are just a few examples of how bioinformatics helps tackle real-world problems:

- It can help cancer researchers identify which gene mutations cause cancer. Scientists can then develop targeted therapies exploiting that knowledge.
- It can help biologists map evolutionary connections and ancestry.
- It can help pharmaceutical companies develop new drugs customized to a person's individual genome.
- It can aid in the development of new vaccines.
- It can enable the development of crops that are more resistant to insects and disease.
- It can identify microbes that have the ability to clean-up environmental waste.
- It can improve the health of livestock.
- It can help forensic scientists identify incriminating DNA evidence.



Databases for bioinformatics –

- GenBank: Genetic sequence database from NCBI
- EMBL-EBI: Nucleotide Sequence Database
- UniProt: Protein sequence database
- GEO Database: Gene expression profiles from NCBI
- Expression Atlas: Gene expression across species and biological conditions

Technology is used in bioinformatics –

Key techniques include **database management, data modeling, pattern recognition, data mining, query processing, and visualization of biological data**. Until very recently, virtually all public databases were based on large flat files stored in simple formats.

The demand in data science and information technology is being pushed forward by the growing popularity of **cloud computing, Augmented Reality, Virtual Reality, Artificial Intelligence, Machine Learning, Decision Intelligence, quantum computing, big data analytics**, and other related technologies.



Machine learning :-

Machine learning **analyzes and examines large chunks of data automatically**. It automates the data analysis process and makes predictions in real-time without any human involvement. You can further build and train the data model to make real-time predictions.

Machine Learning is the core subarea of artificial intelligence. It makes computers get into a self-learning mode without explicit programming. When fed new data, these computers learn, grow, change, and develop by themselves

The concept of machine learning has been around for a while now. However, the ability to automatically and quickly apply mathematical calculations to big data is now gaining a bit of momentum.

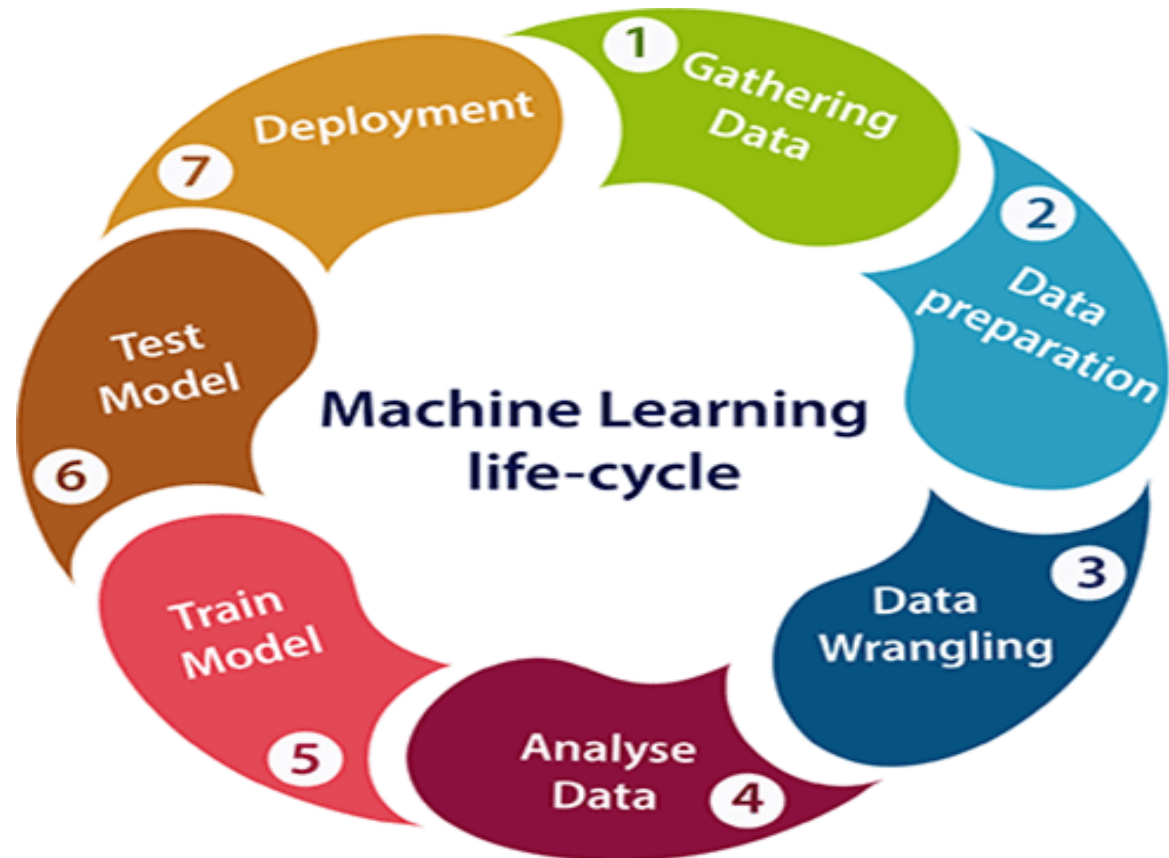
Machine learning has been used in several places like the self-driving Google car, the online recommendation engines – friend recommendations on Facebook, offer suggestions from Amazon, and in cyber fraud detection. In this article, we will learn about the importance of Machine Learning and why every Data Scientist must need it.



Machine learning Life cycle -

Machine learning **life cycle** involves seven major steps, which are given below:

- 1) Gathering Data
- 2) Data preparation
- 3) Data Wrangling
- 4) Analyse Data
- 5) Train the model
- 6) Test the model
- 7) Deployment



1. Gathering Data :

Data Gathering is the first step of the machine learning life cycle. The goal of this step is to identify and obtain all data-related problems.

In this step, we need to identify the different data sources, as data can be collected from various sources such as **files, database, internet, or mobile devices**. It is one of the most important steps of the life cycle. The quantity and quality of the collected data will determine the efficiency of the output. The more will be the data, the more accurate will be the prediction.

This step includes the below tasks:

- Identify various data sources
- Collect data
- Integrate the data obtained from different source

2. Data preparation :

After collecting the data, we need to prepare it for further steps. Data preparation is a step where we put our data into a suitable place and prepare it to use in our machine learning training.

In this step, first, we put all data together, and then randomize the ordering of data.

This step can be further divided into two processes:

- Data exploration
- Data pre-processing



3. Data Wrangling :

Data wrangling is the process of cleaning and converting raw data into a useable format. It is the process of cleaning the data, selecting the variable to use, and transforming the data in a proper format to make it more suitable for analysis in the next step. It is one of the most important steps of the complete process. Cleaning of data is required to address the quality issues.

In real-world applications, collected data may have various issues, including:

- Missing Values
- Duplicate data
- Invalid data
- Noise

4. Data Analysis :

The aim of this step is to build a machine learning model to analyze the data using various analytical techniques and review the outcome. It starts with the determination of the type of the problems, where we select the machine learning techniques such as **Classification, Regression, Cluster analysis, Association**, etc. then build the model using prepared data, and evaluate the model.

Now the cleaned and prepared data is passed on to the analysis step. This step involves:

- Selection of analytical techniques
- Building models
- Review the result



5. Train Model :

Now the next step is to train the model, in this step we train our model to improve its performance for better outcome of the problem.

We use datasets to train the model using various machine learning algorithms. Training a model is required so that it can understand the various patterns, rules, and, features.

6. Test Model :

Once our machine learning model has been trained on a given dataset, then we test the model. In this step, we check for the accuracy of our model by providing a test dataset to it.

Testing the model determines the percentage accuracy of the model as per the requirement of project or problem.

7. Deployment :

The last step of machine learning life cycle is deployment, where we deploy the model in the real-world system.

If the above-prepared model is producing an accurate result as per our requirement with acceptable speed, then we deploy the model in the real system. But before deploying the project, we will check whether it is improving its performance using available data or not. The deployment phase is similar to making the final report for a project.



Use –

Machine learning is used in internet search engines, email filters to sort out spam, websites to make personalised recommendations, banking software to detect unusual transactions, and lots of apps on our phones such as voice recognition.

Purpose –

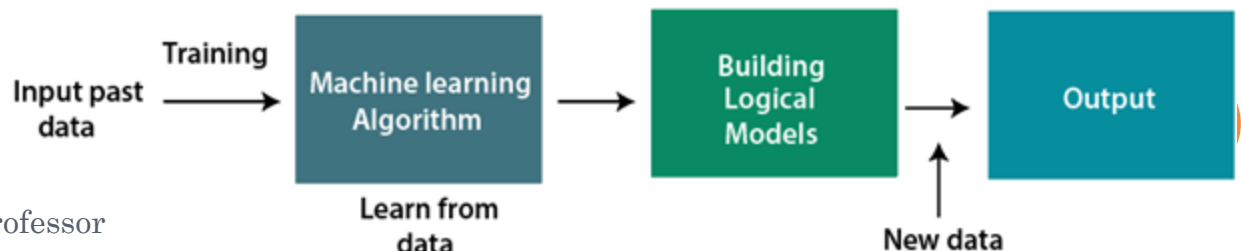
Machine learning **allows the user to feed a computer algorithm an immense amount of data and have the computer analyze and make data-driven recommendations and decisions based on only the input data.**

Types –

Machine learning models rely on four primary data types. These include **numerical data, categorical data, time series data, and text data.**

Working -

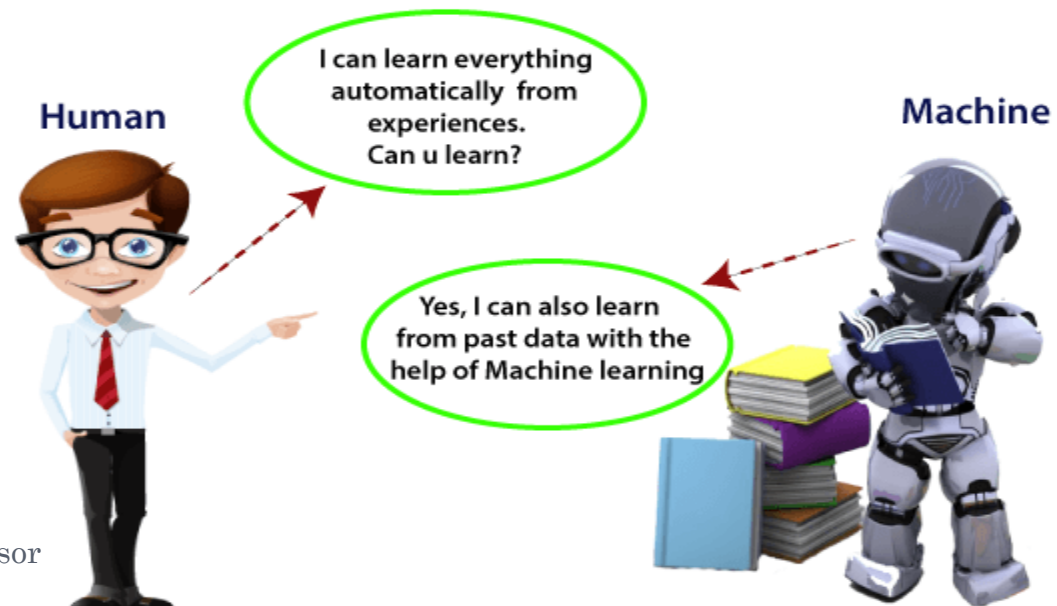
A Machine Learning system **learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it.**



Machine Learning Importance –

The machine learning field is continuously evolving. And along with evolution comes a rise in demand and importance. There is one crucial reason why data scientists need machine learning, and that is: ‘High-value predictions that can guide better decisions and smart actions in real-time without human intervention.’

Machine learning as technology helps analyze large chunks of data, easing the tasks of data scientists in an automated process and is gaining a lot of prominence and recognition. Machine learning has changed the way data extraction and interpretation works by involving automatic sets of generic methods that have replaced traditional statistical techniques.



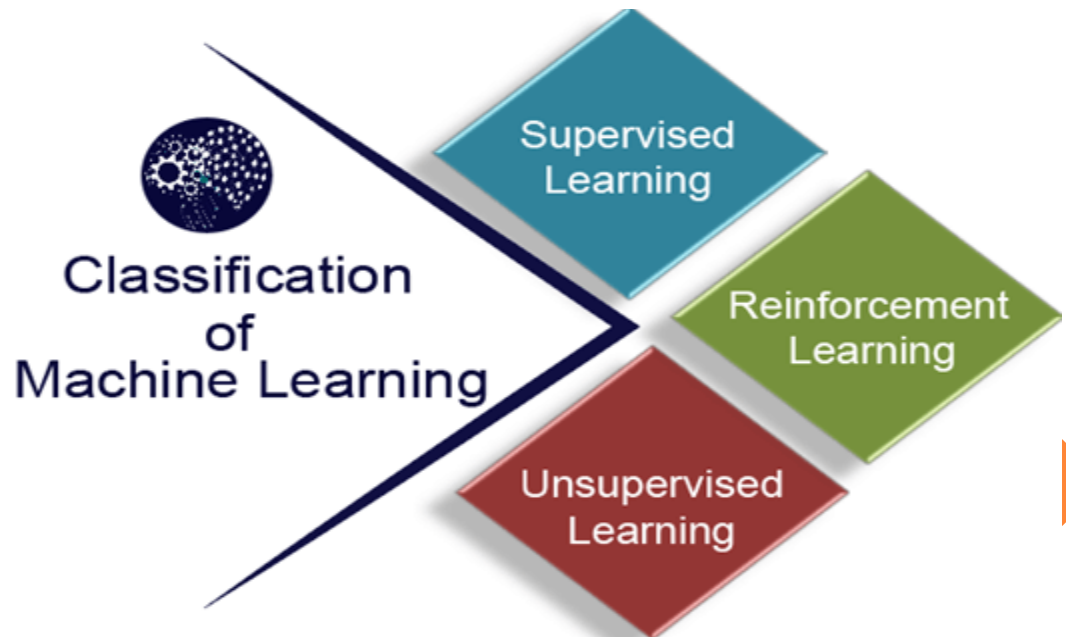
Features of Machine Learning :

- Machine learning uses data to detect various patterns in a given dataset.
- It can learn from past data and improve automatically.
- It is a data-driven technology.
- Machine learning is much similar to data mining as it also deals with the huge amount of the data.

Classification of Machine Learning :

At a broad level, machine learning can be classified into three types:

1. Supervised learning
2. Unsupervised learning
3. Reinforcement learning



1) Supervised Learning –

Supervised learning is a type of machine learning method in which we provide sample labeled data to the machine learning system in order to train it, and on that basis, it predicts the output.

The system creates a model using labeled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not.

The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher.

The example of supervised learning is **spam filtering**.

Supervised learning can be grouped further in two categories of algorithms:

- Classification
- Regression

2) Unsupervised Learning –

Unsupervised learning is a learning method in which a machine learns without any supervision.



The training is provided to the machine with the set of data that has not been labeled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns.

In unsupervised learning, we don't have a predetermined result. The machine tries to find useful insights from the huge amount of data.

It can be further classified into two categories of algorithms:

- Clustering
- Association

3) Reinforcement Learning –

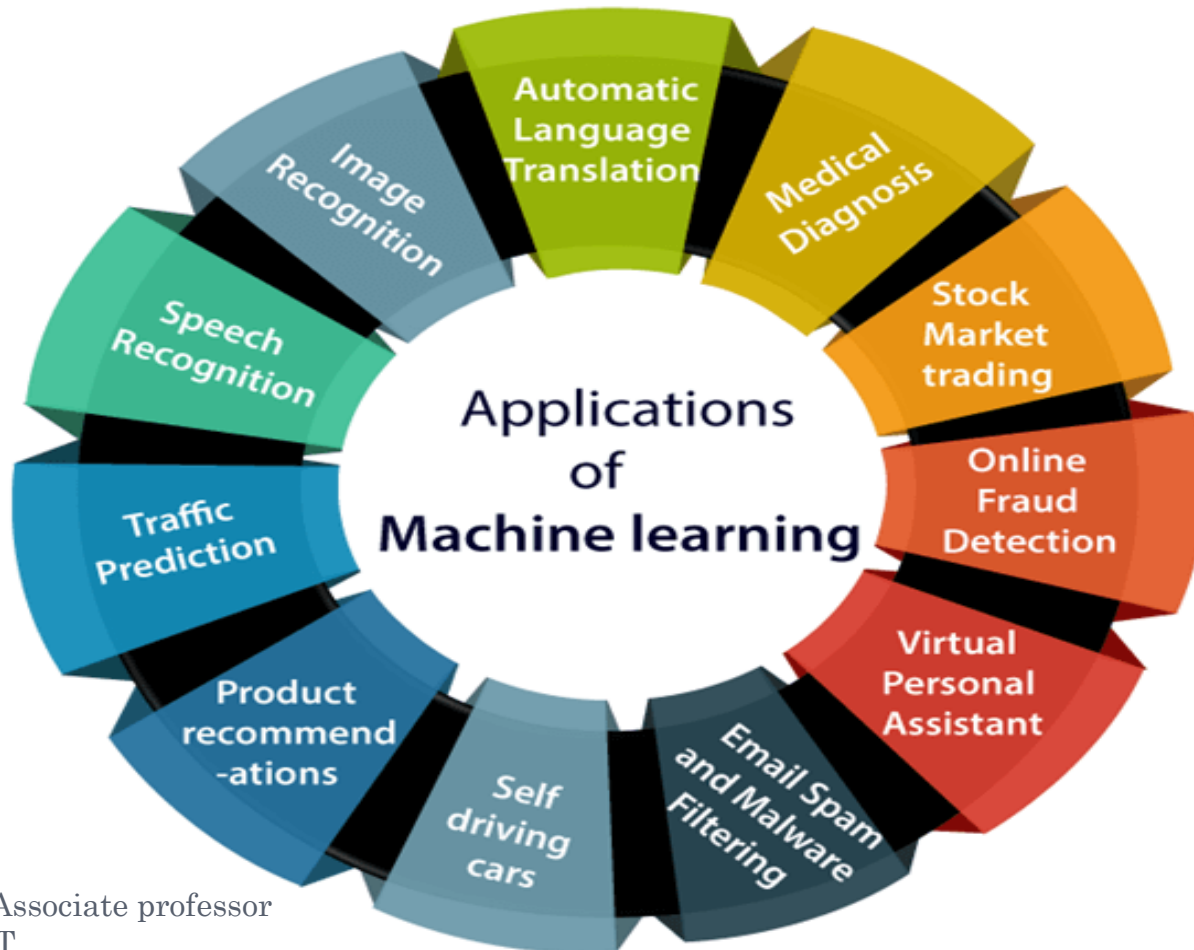
Reinforcement learning is a feedback-based learning method, in which a learning agent gets a reward for each right action and gets a penalty for each wrong action. The agent learns automatically with these feedbacks and improves its performance. In reinforcement learning, the agent interacts with the environment and explores it. The goal of an agent is to get the most reward points, and hence, it improves its performance.

The robotic dog, which automatically learns the movement of his arms, is an example of Reinforcement learning.



Applications of Machine learning –

Machine learning is a buzzword for today's technology, and it is growing very rapidly day by day. We are using machine learning in our daily life even without knowing it such as Google Maps, Google assistant, Alexa, etc. Below are some most trending real-world applications of Machine Learning:



1. Image Recognition :

Image recognition is one of the most common applications of machine learning. It is used to identify objects, persons, places, digital images, etc. The popular use case of image recognition and face detection is, **Automatic friend tagging suggestion**:

Facebook provides us a feature of auto friend tagging suggestion. Whenever we upload a photo with our Facebook friends, then we automatically get a tagging suggestion with name, and the technology behind this is machine learning's **face detection** and **recognition algorithm**.

It is based on the Facebook project named "**Deep Face**," which is responsible for face recognition and person identification in the picture.

2. Speech Recognition :

While using Google, we get an option of "**Search by voice**," it comes under speech recognition, and it's a popular application of machine learning.

Speech recognition is a process of converting voice instructions into text, and it is also known as "**Speech to text**", or "**Computer speech recognition**." At present, machine learning algorithms are widely used by various applications of speech recognition. **Google assistant, Siri, Cortana, and Alexa** are using speech recognition technology to follow the voice instructions.

3. Traffic prediction:

If we want to visit a new place, we take help of Google Maps, which shows us the correct path with the shortest route and predicts the traffic conditions.

It predicts the traffic conditions such as whether traffic is cleared, slow-moving, or heavily congested with the help of two ways:

Real Time location of the vehicle from Google Map app and sensors

Average time has taken on past days at the same time.

Everyone who is using Google Map is helping this app to make it better. It takes information from the user and sends back to its database to improve the performance.

4. Product recommendations:

Machine learning is widely used by various e-commerce and entertainment companies such as **Amazon**, **Netflix**, etc., for product recommendation to the user. Whenever we search for some product on Amazon, then we started getting an advertisement for the same product while internet surfing on the same browser and this is because of machine learning.

Google understands the user interest using various machine learning algorithms and suggests the product as per customer interest.

As similar, when we use Netflix, we find some recommendations for entertainment series, movies, etc., and this is also done with the help of machine learning.

5. Self-driving cars:

One of the most exciting applications of machine learning is self-driving cars. Machine learning plays a significant role in self-driving cars. Tesla, the most popular car manufacturing company is working on self-driving car. It is using unsupervised learning method to train the car models to detect people and objects while driving.

6. Email Spam and Malware Filtering:

Whenever we receive a new email, it is filtered automatically as important, normal, and spam. We always receive an important mail in our inbox with the important symbol and spam emails in our spam box, and the technology behind this is Machine learning. Below are some spam filters used by Gmail:

Content Filter

Header filter

General blacklists filter

Rules-based filters

Permission filters

Some machine learning algorithms such as **Multi-Layer Perceptron**, **Decision tree**, and **Naïve Bayes classifier** are used for email spam filtering and malware detection.

7. Virtual Personal Assistant:

We have various virtual personal assistants such as **Google assistant, Alexa, Cortana, Siri**. As the name suggests, they help us in finding the information using our voice instruction.

These assistants can help us in various ways just by our voice instructions such as Play music, call someone, Open an email, Scheduling an appointment, etc.

These virtual assistants use machine learning algorithms as an important part. These assistant record our voice instructions, send it over the server on a cloud, and decode it using ML algorithms and act accordingly.

8. Online Fraud Detection:

Machine learning is making our online transaction safe and secure by detecting fraud transaction. Whenever we perform some online transaction, there may be various ways that a fraudulent transaction can take place such as **fake accounts, fake ids, and steal money** in the middle of a transaction. So to detect this, **Feed Forward Neural network** helps us by checking whether it is a genuine transaction or a fraud transaction.

For each genuine transaction, the output is converted into some hash values, and these values become the input for the next round. For each genuine transaction, there is a specific pattern which gets change for the fraud transaction hence, it detects it and makes our online transactions more secure.

9. Stock Market trading:

Machine learning is widely used in stock market trading. In the stock market, there is always a risk of up and downs in shares, so for this machine learning's **long short term memory neural network** is used for the prediction of stock market trends.

10. Medical Diagnosis:

In medical science, machine learning is used for diseases diagnoses. With this, medical technology is growing very fast and able to build 3D models that can predict the exact position of lesions in the brain.

It helps in finding brain tumors and other brain-related diseases easily.

11. Automatic Language Translation:

Nowadays, if we visit a new place and we are not aware of the language then it is not a problem at all, as for this also machine learning helps us by converting the text into our known languages. Google's GNMT (Google Neural Machine Translation) provide this feature, which is a Neural Machine Learning that translates the text into our familiar language, and it called as automatic translation.

The technology behind the automatic translation is a sequence to sequence learning algorithm, which is used with image recognition and translates the text from one language to another language.

Common machine learning algorithms –

A number of machine learning algorithms are commonly used. These include:

- **Neural networks:** Neural networks simulate the way the human brain works, with a huge number of linked processing nodes. Neural networks are good at recognizing patterns and play an important role in applications including natural language translation, image recognition, speech recognition, and image creation.
- **Linear regression:** This algorithm is used to predict numerical values, based on a linear relationship between different values. For example, the technique could be used to predict house prices based on historical data for the area.
- **Logistic regression:** This supervised learning algorithm makes predictions for categorical response variables, such as “yes/no” answers to questions. It can be used for applications such as classifying spam and quality control on a production line.



- **Clustering:** Using unsupervised learning, clustering algorithms can identify patterns in data so that it can be grouped. Computers can help data scientists by identifying differences between data items that humans have overlooked.
- **Decision trees:** Decision trees can be used for both predicting numerical values (regression) and classifying data into categories. Decision trees use a branching sequence of linked decisions that can be represented with a tree diagram. One of the advantages of decision trees is that they are easy to validate and audit, unlike the black box of the neural network.
- **Random forests:** In a random forest, the machine learning algorithm predicts a value or category by combining the results from a number of decision trees.



Unit 6

Applications of Data Science



Contents :

- ❖ Technologies for visualization
- ❖ Bokeh (Python)
- ❖ Recent trends in various data collection and analysis techniques
- ❖ Various visualization techniques
- ❖ Application development methods of used in data science



Technologies for visualization :-

The increasing interest in data science and data analytics lead to a growing interest in data visualization and exploratory visual data analysis. However, there is still a clear gap between new developments in visualization research, and the visualization techniques currently applied in data analytics workflows.

Most of the commonly used tools provide basic charting options, but more advanced visualization techniques have hardly been integrated as features yet. This especially applies for interactive exploratory data analysis, which has already been addressed as the 'Interactive Visualization Gap' in the literature. In this paper we present a study on the usage of visualization techniques in common data science tools.



The results of the study confirm that the gap still exists. For example, we hardly found support for advanced techniques for temporal data visualization or radial visualizations in the evaluated tools and applications.

On the contrary, interviews with professional data analysts confirm strong interest in learning and applying new tools and techniques. Users are especially interested in techniques that can support their exploratory analysis workflow.

Based on these findings and our own experience with data science projects, we present suggestions and considerations towards a better integration of visualization techniques in current data science workflows.



By using visual elements like **charts, graphs, and maps**, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. This blog on data visualization techniques will help you understand detailed techniques and benefits.

Visualization technology is now an indispensable part of automation. What is called process visualization or image generation is whenever production processes and machine data are presented in the form of diagrams, curves and historic charts so that people can understand them better.





Bokeh is a data visualization library for Python. Unlike Matplotlib and Seaborn, they are also Python packages for data visualization, Bokeh renders its plots using HTML and JavaScript. Hence, it proves to be extremely useful for developing web based dashboards.

Bokeh primarily converts the data source into a JSON file which is used as input for BokehJS, a JavaScript library, which in turn is written in TypeScript and renders the visualizations in modern browsers.



Bokeh is a Python library for **creating interactive visualizations for modern web browsers**. It helps you build beautiful graphics, ranging from simple plots to complex dashboards with streaming datasets.

Bokeh is a data visualization library that allows a developer to code in Python and output JavaScript charts and visuals in web browsers.

It is a Python library that allows us to make interactive visualization of browsers. It enables us to create visually stunning graphics. We can use Bokeh to create JavaScript-powered visualizations without having to write any JavaScript.

Bokeh is a Python data visualization library that provides high performance. Bokeh output is available in a variety of formats, including notebook, html, and server. Bokeh plots can be included in Flask applications.

Users can choose between two different visualization interfaces offered by Bokeh. A low-level interface that allows application developers a great deal of freedom. A high-level interface for producing visual glyphs. As we know it is designed to be displayed in web browsers. This is where Bokeh differs from other visualization libraries.



Bokeh supports a variety of languages. These bindings generate a JSON file that serves and displays data to modern web browsers.

It enables us to quickly create complex statistical plots using simple commands. Bokeh provides output in a variety of formats, including HTML, notebook, and server.

We can also integrate the bokeh visualization into flask and Django apps. Python bokeh can transform visualizations created with other libraries such as matplotlib, seaborn, and ggplot.

It has the ability to apply interaction and various styling options to visualization.

Bokeh is primarily used to convert source data into JSON, which is then used as input for BokehJS. Some of the most appealing aspects of Bokeh is, that it provides charts as well as custom charts for complex use cases.

It has an easy-to-use interface and can be used with notebooks of jupyter. We have complete control over our chart and can easily modify it with custom Javascript.

It includes a plethora of examples and ideas to get us started, and it is distributed under the BSD license.

It is very useful and important in python to make interactive browser visualizations.



Recent trends in various data collection and analysis techniques :-

1. Big Data on the Cloud
2. Emphasis on Actionable Data
3. Data as a Service- Data Exchange in Marketplaces
4. Use of Augmented Analytics
5. Cloud Automation and Hybrid Cloud Services
6. Focus on Edge Intelligence
7. Hyperautomation
8. Use of Big Data in the Internet of Things (IoT)
9. Automation of Data Cleaning
10. Increase in Use of Natural Language Processing
11. Quantum Computing for Faster Analysis
12. Democratizing AI and Data Science
13. Automation of Machine Learning (AutoML)
14. Computer Vision for High Dimensional Data Analytics
15. Generative AI for Deepfake and Synthetic Data
16. Blockchain in Data Science
17. Python is Still the Top Programming Language



Various visualization techniques :-

Data visualization is a graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. This blog on data visualization techniques will help you understand detailed techniques and benefits. In the world of Big Data, data visualization in Python tools and technologies are essential to analyze massive amounts of information and make data-driven decisions.

Data Visualization Techniques are -

- Box plots
- Histograms
- Heat maps
- Charts
- Tree maps
- Word Cloud/Network diagram



List of Methods to Visualize Data -

Column Chart: It is also called a vertical bar chart where each category is represented by a rectangle. The height of the rectangle is proportional to the values that are plotted.

Bar Graph: It has rectangular bars in which the lengths are proportional to the values which are represented.

Stacked Bar Graph: It is a bar style graph that has various components stacked together so that apart from the bar, the components can also be compared to each other.

Stacked Column Chart: It is similar to a stacked bar; however, the data is stacked horizontally.

Area Chart: It combines the line chart and bar chart to show how the numeric values of one or more groups change over the progress of a viable area.

Dual Axis Chart: It combines a column chart and a line chart and then compares the two variables.

Line Graph: The data points are connected through a straight line; therefore, creating a representation of the changing trend.

Mekko Chart: It can be called a two-dimensional stacked chart with varying column widths.

Pie Chart: It is a chart where various components of a data set are presented in the form of a pie which represents their proportion in the entire data set.

Waterfall Chart: With the help of this chart, the increasing effect of sequentially introduced positive or negative values can be understood.

Bubble Chart: It is a multi-variable graph that is a hybrid of Scatter Plot and a Proportional Area Chart.

Scatter Plot Chart: It is also called a scatter chart or scatter graph. Dots are used to denote values for two different numeric variables.

Bullet Graph: It is a variation of a bar graph. A bullet graph is used to swap dashboard gauges and meters.

Funnel Chart: The chart determines the flow of users with the help of a business or sales process.

Heat Map: It is a technique of data visualization that shows the level of instances as color in two dimensions.

Box Plots –

A box plot is a graph that gives you a good indication of how the values in the data are spread out. Although box plots may seem primitive in comparison to a histogram or density plot, they have the advantage of taking up less space, which is useful when comparing distributions between many groups or datasets. For some distributions/datasets, you will find that you need more information than the measures of central tendency (median, mean, and mode). You need to have information on the variability or dispersion of the data.

Histograms

A histogram is a graphical display of data using bars of different heights. In a histogram, each bar groups numbers into ranges. Taller bars show that more data falls in that range. A histogram displays the shape and spread of continuous sample data.

Heat Maps

A heat map is data analysis software that uses colour the way a bar graph uses height and width: as a data visualization tool.



Charts

Line Chart

The simplest technique, a line plot is used to plot the relationship or dependence of one variable on another.

Bar Charts

Bar charts are used for comparing the quantities of different categories or groups.

Pie Chart

It is a circular statistical graph which divides slices to illustrate numerical proportion.

Scatter Charts

Another common visualization technique is a scatter plot that is a two-dimensional plot representing the joint variation of two data items.

Bubble Charts

It is a variation of scatter chart in which the data points are replaced with bubbles, and an additional dimension of data is represented in the size of the bubbles.

Timeline Charts

Timeline charts illustrate events, in chronological order . for example the progress of a project, advertising campaign, acquisition process in whatever unit of time the data was recorded . for example week, month, year, quarter.



Tree Maps

A treemap is a visualization that displays hierarchically organized data as a set of nested rectangles, parent elements being tiled with their child elements. The sizes and colours of rectangles are proportional to the values of the data points they represent. A leaf node rectangle has an area proportional to the specified dimension of the data. Depending on the choice, the leaf node is coloured, sized or both according to chosen attributes. They make efficient use of space, thus display thousands of items on the screen simultaneously.

Word Clouds and Network Diagrams for Unstructured Data

The variety of big data brings challenges because semi-structured, and unstructured data require new visualization techniques. A word cloud visual represents the frequency of a word within a body of text with its relative size in the cloud. This technique is used on unstructured data as a way to display high- or low-frequency words.

Another visualization technique that can be used for semi-structured or unstructured data is the network diagram. Network diagrams represent relationships as nodes (individual actors within the network) and ties (relationships between the individuals). They are used in many applications, for example for analysis of social networks or mapping product sales across geographic areas.



Application development methods of used in data science.:-

10 applications that build upon the concepts of Data Science, exploring various domains such as :

- Fraud and Risk Detection
- Healthcare
- Internet Search
- Targeted Advertising
- Website Recommendations
- Advanced Image Recognition
- Speech Recognition
- Airline Route Planning
- Gaming
- Augmented Reality

