DATA SCIENCE (SYMCA)

By,
Mr.Pramod Jadhao
Ms.Shubhangi Vitalkar

Unit I Introduction to Data Science

What Is Data Science?

Data science is the domain of study that deals with vast volumes of data using modern tools and techniques to find unseen patterns, derive meaningful information, and make business decisions. Data science uses complex machine learning algorithms to build predictive models.

The data used for analysis can come from many different sources and presented in various formats.

The Data Science Lifecycle / Data Science Process

Now that you know what is data science, next up let us focus on the data science lifecycle. Data science's lifecycle
consists of five distinct stages, each with its own tasks:

1. **Capture:** Data Acquisition, Data Entry, Signal Reception, Data Extraction. This stage involves gathering raw structured and unstructured data.
2. **Maintain:** Data Warehousing, Data Cleansing, Data Staging, Data Processing, Data Architecture. This stage covers taking the raw data and putting it in a form that can be used.
3. **Process:** Data Mining, Clustering/Classification, Data Modeling, Data Summarization. Data scientists takethe prepared data and examine its patterns, ranges, and biases to determine how useful it will be in predictive analysis.
4. **Analyze:** Exploratory/Confirmatory, Predictive Analysis, Regression, Text Mining, Qualitative Analysis. Here is the real meat of the lifecycle. This stage involves performing the various analyses on the data.
5. **Communicate:** Data Reporting, Data Visualization, Business Intelligence, Decision Making. In this final step, analysts prepare the analyses in easily readable forms such as charts, graphs, and reports.

Prerequisites for Data Science

Here are some of the technical concepts you should know about before starting to learn what is data science.

1. **Machine Learning:** Machine learning is the backbone of data science. Data Scientists need to have a solidgrasp of ML in addition to basic knowledge of statistics.
2. **Modeling:** Mathematical models enable you to make quick calculations and predictions based on what you already know about the data. Modeling is also a part of Machine Learning and involves identifying which algorithm is the most suitable to solve a given problem and how to train these models.
3. **Statistics:** Statistics are at the core of data science. A sturdy handle on statistics can help you extract more intelligence and obtain more meaningful results.
4. **Programming:** Some level of programming is required to execute a successful data science project. The most common programming languages are Python, and R. Python is especially popular because it's easy to learn, and it supports multiple libraries for data science and ML.
5. **Databases:** A capable data scientist needs to understand how databases work, how to manage them, and how to extract data from them.

Need of Data Science….

The principal purpose of Data Science is to find patterns within data. It uses various statistical techniques to analyze and draw insights from the data. From data extraction, wrangling and pre-processing, a Data Scientist must examine the data systematically.

Then, he has the responsibility of making predictions from the data. The goal of a Data Scientist is to derive conclusions from the data. Through these conclusions, he is able to assist companies in making smarter businessdecisions.

Big Data

What is Big Data?

Big Data literally means large amounts of data. Big data is the pillar behind the idea that one can make useful inferences with a large body of data that wasn't possible before with smaller datasets. So extremely large data sets may be analyzed computationally to reveal patterns, trends, and associations that are not transparent or easy to identify.

Why is everyone interested in Big Data?

Definition: Refers to large volumes of structured, semi-structured, and unstructured data that require advanced tools and techniques for processing and analysis

Big data is everywhere!

Every time you go to the web and do something that data is collected, every time you buy something from one of the e-commerce your data is collected. Whenever you go to store data is collected at the point of sale, when you do Bank transactions that data is there, when you go to Social networks like Facebook, Twitter that data is collected. Now, these are more social data, but the same thing is starting to happen with real engineering plants.Real-time data is collected from plants all over the world. Not only these if you are doing much more sophisticated simulation, molecular simulations, which generates tons of data that is also collected and stored.

Characteristics:

- **Volume**: Massive amounts of data generated at high velocity.
- **Velocity**: Data is generated and processed rapidly.
- **Variety**: Diverse types of data including text, images, videos, sensor data, etc.
- **Veracity**: Concerns the accuracy and reliability of data.

How much data is Big Data?

Google processes 20 Petabytes(PB) per day (2008)

Facebook has 2.5 PB of user data + 15 TB per day (2009)

eBay has 6.5 PB of user data + 50 TB per day (2009)

CERN's Large Hadron Collider(LHC) generates 15 PB a year

So one of the reasons for the acceleration of data science in recent years is the enormous volume of data (e.g Big Data) currently available and being generated. Not only are huge amounts of data being collected about many aspects of the world and our lives, but we concurrently have the rise of inexpensive computing. This has formed the perfect storm in which we have rich data and the tools to analyze it. Advancing computer memory capacities, more enhanced software, more competent processors, and now, more numerous data scientists with the skills to put this to use and solve questions using the data! And that's the big reason why do we need data science in the future.

Difference between big data and little data

Feature	Little Data	Big Data
Technology	Traditional	Modern
Collection	Generally, it is obtained in an organized manner than is inserted into the database	The Big Data collection is done by using pipelines having queues like AWS Kinesis or Google Pub / Sub to balance high-speed data
Volume	Data in the range of tens or hundreds of Gigabytes	Size of Data is more than Terabytes
Analysis Areas	Data marts(Analysts)	Clusters(Data Scientists), Data marts(Analysts)
Quality	Contains less noise as data is less collected in a controlled manner	Usually, the quality of data is not guaranteed
Processing	It requires batch-oriented processing pipelines	It has both batch and stream processing pipelines
Database	SQL	NoSQL
Velocity	A regulated and constant flow of data, data aggregation is slow	Data arrives at extremely high speeds, large volumes of data aggregation in a short time
Structure	Structured data in tabular format with fixed schema(Relational)	Numerous variety of data set including tabular data, text, audio, images, video, logs, JSON etc.(Non Relational)
Scalability	They are usually vertically scaled	They are mostly based on horizontally scaling architectures, which gives more versatility at a lower cost
Query Language	only Sequel	Python, R, Java, Sequel
Hardware	A single server is sufficient	Requires more than one server
Value	Business Intelligence, analysis and reporting	Complex data mining techniques for pattern finding, recommendation, prediction etc.
Optimization	Data can be optimized manually(human powered)	Requires machine learning techniques for data optimization
Storage	Storage within enterprises, local servers etc.	Usually requires distributed storage systems on cloud or in external file systems
People	Data Analysts, Database Administrators and Data Engineers	Data Scientists, Data Analysts, Database Administrators and Data Engineers
Security	Security practices for Small Data include user privileges, data encryption, hashing, etc.	Securing Big Data systems are much more complicated. Best security practices include data encryption, cluster network isolation, strong access control protocols etc.
Nomenclature	Database, Data Warehouse, Data Mart	Data Lake
Infrastructure	Predictable resource allocation, mostly vertically scalablehardware.	More agile infrastructure with horizontally scalable hardware

The current Scenario of data science

Expansion of Applications: Data science is increasingly being applied across diverse domains such as healthcare, finance, retail, manufacturing, and transportation. It plays a crucial role in optimizing processes, improving decision-making, and driving innovation.

Integration with AI and Machine Learning: Data science heavily intersects with artificial intelligence (AI) and machine learning (ML). AI and ML algorithms are utilized for predictive analytics, pattern recognition, natural language processing (NLP), and computer vision tasks, among others.

Big Data Infrastructure: With the proliferation of big data, data science projects often involve managing and analysing large datasets using distributed computing frameworks like Hadoop and Spark, as well as cloud-based solutions provided by Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure.

Focus on Ethical and Responsible AI: There is a growing emphasis on ethical considerations in data science and AI applications. Issues such as bias in algorithms, data privacy, transparency, and accountability are gaining attention, leading to frameworks and guidelines being developed to address these concerns.

Emerging Technologies: Data science is embracing emerging technologies such as edge computing, Internet of Things (IoT), and block chain, which generate new types of data and require innovative approaches for analysis and integration.

Interdisciplinary Collaboration: Data science teams often consist of professionals with diverse backgrounds in statistics, mathematics, computer science, domain expertise (e.g., healthcare, finance), and business acumen. Collaborative efforts are essential for successful implementation and deployment of data-driven solutions.

Demand for Data Professionals: There is a high demand for skilled data scientists, data engineers, and analysts across industries. Organizations are investing in building data science capabilities to gain competitive advantage and drive growth.

Education and Training: Educational institutions and online platforms offer a wide range of courses and programs in data science, catering to individuals seeking to enter or advance their careers in this field. Continuous learning and upskilling are essential due to the rapid pace of technological change.

Visualization and Communication: Effective data visualization and communication skills are crucial for data scientists to convey insights and recommendations to stakeholders, aiding in decision-making processes.

Regulatory Landscape: Data science practices are influenced by regulatory frameworks such as GDPR (General Data Protection Regulation) in Europe and similar data protection laws globally. Compliance with these regulations is essential for ethical data handling and user privacy.

Structured data?

Structured data — typically categorized as quantitative data — is highly organized and easily decipherable by machine learning algorithms. Developed by IBM in 1974, structured query language (SQL) is the programming language used to manage structured data. By using a relational (SQL) database, business users can quickly input,search and manipulate structured data.

Pros and cons of structured data

Examples of structured data include dates, names, addresses, credit card numbers, etc. Their benefits are

tiedto ease of use and access, while liabilities revolve around data inflexibility:

Pros

- Easily used by machine learning (ML) algorithms: The specific and organized architecture of structured data eases manipulation and querying of ML data.
- Easily used by business users: Structured data does not require an in-depth understanding of different types of data and how they function. With a basic understanding of the topic relative to the data, users can easily access and interpret the data.
- Accessible by more tools: Since structured data predates unstructured data, there are more tools available for using and analyzing structured data.

Cons

- Limited usage: Data with a predefined structure can only be used for its intended purpose, which limits its flexibility and usability.
- Limited storage options: Structured data is generally stored in data storage systems with rigid schemas (e.g., “data warehouses”). Therefore, changes in data requirements necessitate an update of all structured data, which leads to a massive expenditure of time and resources.

Example of structured data in a tabular format:

Consider a simple table representing sales data for a fictional company:

Order ID	Customer Name	Product Name	Quantity	Unit Price	Total Amount	Order Date
1001	John Doe	Laptop	2	$1200	$2400	2024-07-10
1002	Jane Smith	Smartphone	1	$800	$800	2024-07-11
1003	David Brown	Tablet	3	$500	$1500	2024-07-12

Structured data tools

- OLAP: Performs high-speed, multidimensional data analysis from unified, centralized data stores.
- SQLite: Implements a self-contained, server-less, zero-configuration, transactional relational databaseengine.
- MySQL: Embeds data into mass-deployed software, particularly mission-critical, heavy-load productionsystem.
- PostgreSQL: Supports SQL and JSON querying as well as high-tier programming languages (C/C+, Java,Python, etc.).

Use cases for structured data

- Customer relationship management (CRM): CRM software runs structured data through analytical toolsto create datasets that reveal customer behavior patterns and trends.
- Online booking: Hotel and ticket reservation data (e.g., dates, prices, destinations, etc.) fits the “rowsand columns” format indicative of the pre-defined data model.
- Accounting: Accounting firms or departments use structured data to process and record financialtransactions.

Unstructured data?

Unstructured data, typically categorized as qualitative data, cannot be processed and analyzed via conventional data tools and methods. Since unstructured data does not have a predefined data model, it is best managed in non-relational (NoSQL) databases. Another way to manage unstructured data is to use data lakes to preserve it in raw form.

The importance of unstructured data is rapidly increasing. Recent projections indicate that unstructured data is over 80% of all enterprise data, while 95% of businesses prioritize unstructured data management.

Pros and cons of unstructured data

Examples of unstructured data include text, mobile activity, social media posts, Internet of Things (IoT) sensor data, etc. Their benefits involve advantages in format, speed and storage, while liabilities revolve around expertise and available resources:

Pros

- Native format: Unstructured data, stored in its native format, remains undefined until needed. Its adaptability increases file formats in the database, which widens the data pool and enables data scientists to prepare and analyze only the data they need.
- Fast accumulation rates: Since there is no need to predefine the data, it can be collected quickly and easily.
- Data lake storage: Allows for massive storage and pay-as-you-use pricing, which cuts costs and eases scalability.

Cons

- Requires expertise: Due to its undefined/non-formatted nature, data science expertise is required to prepare and analyze unstructured data. This is beneficial to data analysts but alienates unspecialized business users who may not fully understand specialized data topics or how to utilize their data.
- Specialized tools: Specialized tools are required to manipulate unstructured data, which limits product choices for data managers.

Unstructured data tools

- MongoDB: Uses flexible documents to process data for cross-platform applications and services.
- DynamoDB: Delivers single-digit millisecond performance at any scale via built-in security, in-memorycaching and backup and restore.
- Hadoop: Provides distributed processing of large data sets using simple programming models and noformatting requirements.
- Azure: Enables agile cloud computing for creating and managing apps through Microsoft's data centers.

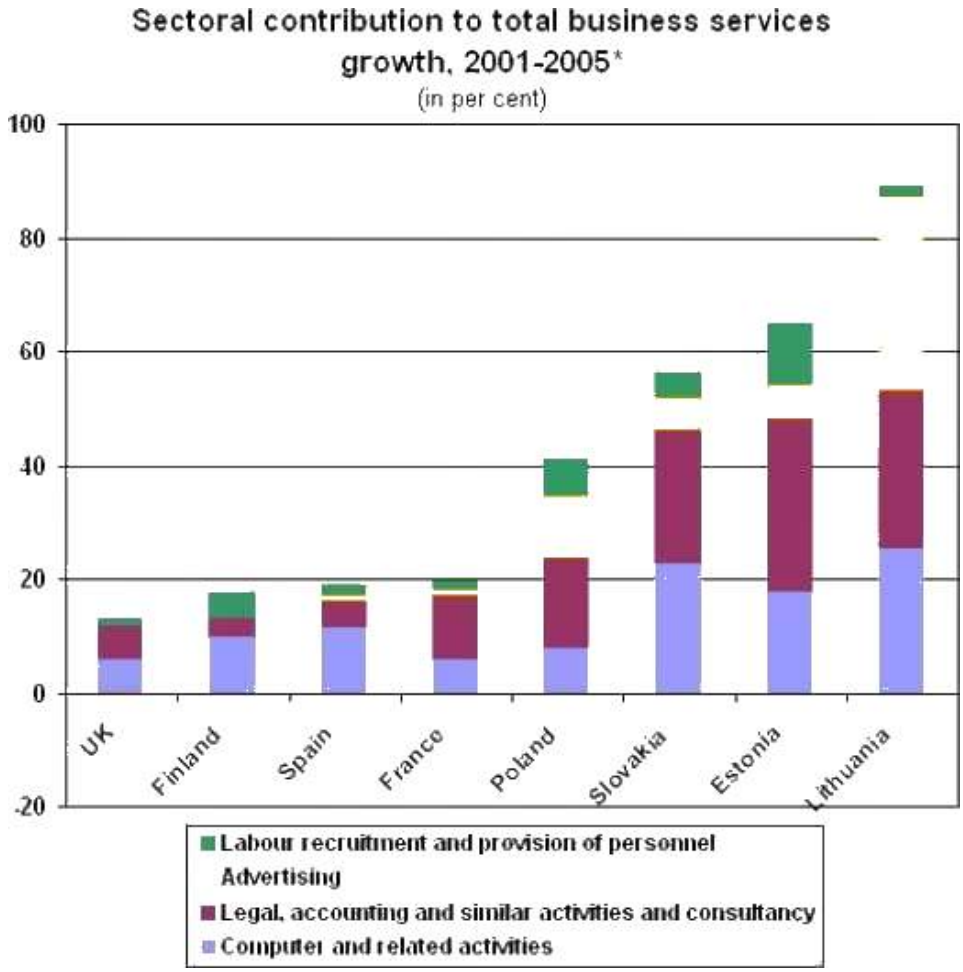
Use cases for unstructured data

- Data mining: Enables businesses to use unstructured data to identify consumer behavior, product sentiment, and purchasing patterns to better accommodate their customer base.
- Predictive data analytics: Alert businesses of important activity ahead of time so they can properly planand accordingly adjust to significant market shifts.
- Chatbots: Perform text analysis to route customer questions to the appropriate answer sources.

Categorical data?

Qualitative variables measure attributes that can be given only as a property of the variables. The political affiliation of a person, nationality of a person, the favorite color of a person, and the blood group of a patient can only be measured using qualitative attributes of each variable. Often these variables have limited number of possibilities and assume only one of the possible outcomes; i.e. the value is one of the given categories.

Therefore, these are commonly known as categorical variables. These possible values can be numbers, letters, names, or any symbol.

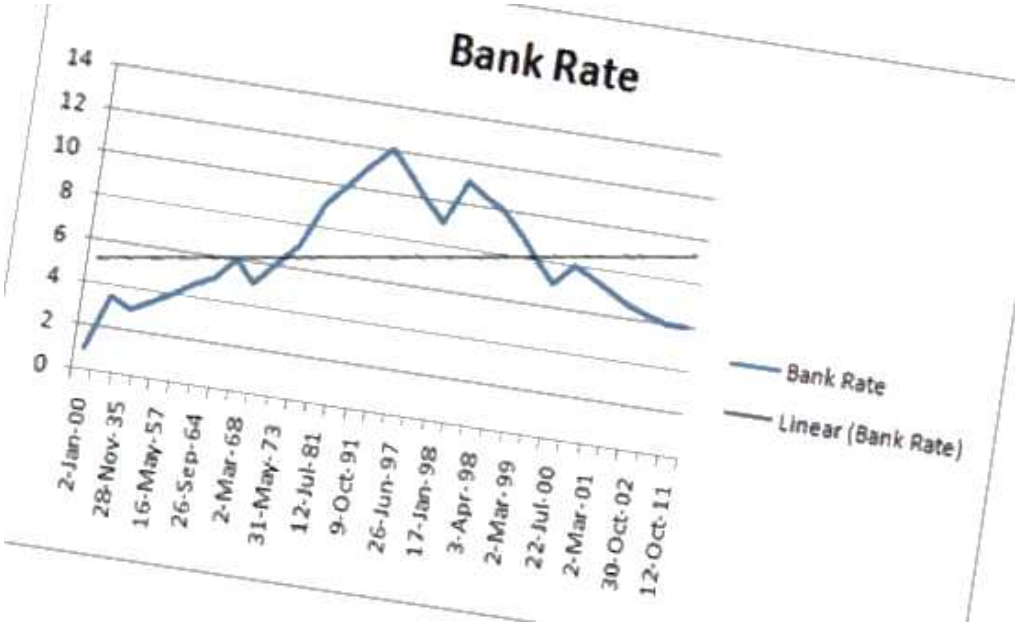


Quantitative data?

Quantitative variable records the attributes that can be measured by a magnitude or size; i.e., quantifiable. Variables measuring temperature, weight, mass or the height of a person or the annual income of a householdare quantitative variables. Not only all the values of these variables are numbers, but each number gives a sense of value too.

The data in quantitative type belong to either one of the three following types; Ordinal, Interval, and Ratio. Categorical data always belong to the nominal type. Above mentioned types are formally known as levels of measurement, and closely related to the way the measurements are made and the scale of each measurement.

Since the form of the data in the two categories is different, different techniques and methods are employed when gathering, analyzing, and describing.



What is the Difference Between Categorical and Quantitative data?

Definitions of Categorical and Quantitative data:

- Quantitative data are information that has a sensible meaning when referring to its magnitude.
- Categorical data are often information that takes values from a given set of categories or groups.

Characteristics of Categorical and Quantitative data:

Class of measurement:

- Quantitative data belong to ordinal, interval, or ratio classes of measurements.
- Categorical data belong to the nominal class of measurements.

Methods:

- Methods used to analyze quantitative data are different from the methods used for categorical data,even if the principles are the same, at least the application have significant differences.

Analysis:

- Quantitative data are analyzed using statistical methods in descriptive statistics, regression, timeseries, and many more.
- For categorical data, usually descriptive methods and graphical methods are employed. Some non-parametric tests are also used.

Roles & Responsibilities of a Data Scientist

- **Management:** The Data Scientist plays an insignificant managerial role where he supports the construction of the base of futuristic and technical abilities within the Data and Analytics field in orderto assist various planned and continuing data analytics projects.
- **Analytics:** The Data Scientist represents a scientific role where he plans, implements, and assesses high-level statistical models and strategies for application in the business's most complex issues. The Data Scientist develops econometric and statistical models for various problems including projections,classification, clustering, pattern analysis, sampling, simulations, and so forth.
- **Strategy/Design:** The Data Scientist performs a vital role in the advancement of innovative strategies to understand the business's consumer trends and management as well as ways to solve difficult business problems, for instance, the optimization of product fulfillment and entire profit.
- **Collaboration:** The role of the Data Scientist is not a solitary role and in this position, he collaborates with superior data scientists to communicate obstacles and findings to relevant stakeholders in an effort to enhance drive business performance and decision-making.
- **Knowledge:** The Data Scientist also takes leadership to explore different technologies and tools with the vision of creating innovative data-driven insights for the business at the most agile pace feasible. In this situation, the Data Scientist also uses initiative in assessing and utilizing new and enhanced data science methods for the business, which he delivers to senior management of approval.
- **Other Duties:** A Data Scientist also performs related tasks and tasks as assigned by the Senior DataScientist, Head of Data Science, Chief Data Officer, or the Employer.

Difference Between Data Scientist, Data Analyst, and Data Engineer

Data Scientist, Data Engineer, and Data Analyst are the three most common careers in data science. So let'sunderstand who's data science by comparing it with its similar jobs.

Data Scientist	Data Analyst	Data Engineer
The focus will be on the futuristic display of data.	The main focus of a data analyst is on optimization of scenarios, for example how an employee can enhance the company's product growth.	Data Engineers focus on optimization techniques and the construction of data in a conventional manner. The purpose of a data engineer is continuously advancing data consumption.
Data scientists present both supervised and unsupervised learning of data, say regression and classification of data, Neural networks, etc.	Data formation and cleaning of raw data, interpreting and visualization of data to perform the analysis and to perform the technical summary of data.	Frequently data engineers operate at the back end. Optimized machine learning algorithms were used for keeping data and making data to be prepared most accurately.
Skills required for Data Scientist are Python, R, SQL, Pig, SAS, Apache Hadoop, Java, Perl, Spark.	Skills required for Data Analyst are Python, R, SQL, SAS.	Skills required for Data Engineer are MapReduce, Hive, Pig Hadoop, techniques.

Some Inspiring Data Scientists

The variety of areas in which data science is used is embodied by looking at examples of data scientists.

- **Hilary Mason:** She is the co-founder of Fast Forward labs, a machine learning company recently owned by **Cloudera**, a data science company. She is a Data Scientist at Accel. Broadly, she works with data to solve questions about mining the web and also learning the method that how people communicate with each other through social media.
- **Nate Silver:** He is one of the most prominent data scientists or statisticians in the world today. He is the founder of FiveThirtyEight. FiveThirtyEight is a website that applies statistical analysis to tell compelling stories about elections, politics, sports, science, and lifestyle. He utilizes huge amounts of public data to predict a diversity of topics; most prominently he predicts who will win elections in the
  U.S. and has an extraordinary track record for accuracy in doing so.
- **Daryl Morey**: He is the general manager of a US basketball team, the Houston Rockets. He was awarded the job as GM based on his bachelor's degree in computer science and his M.B.A. from M.I.T.

UNIT II DATA PRE-PROCESSING AND WAREHOUSE

DATA PRE-PROCESSING AND WAREHOUSE

Data preprocessing is an important step in the data mining process. It refers to the cleaning, transforming, and integrating of data in order to make it ready for analysis. The goal of data preprocessing is to improve the quality of the data and to make it more suitable for the specific data mining task.

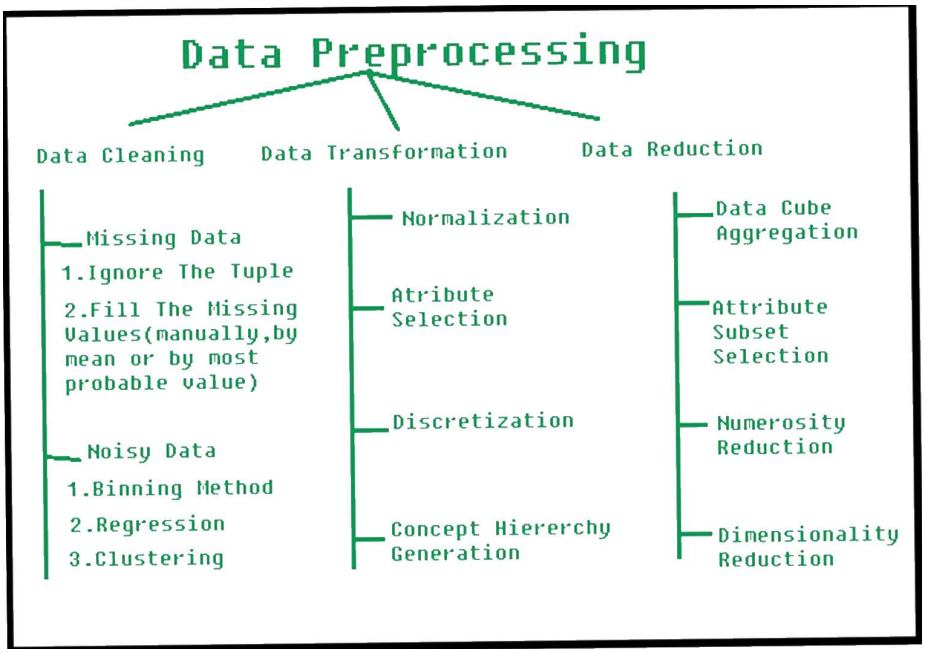
Some common steps in data preprocessing include:

Data preprocessing is an important step in the data mining process that involves cleaning and transforming raw data to make it suitable for analysis. Some common steps in data preprocessing include:

Data Cleaning: This involves identifying and correcting errors or inconsistencies in the data, such as missing values, outliers, and duplicates. Various techniques can be used for data cleaning, such as imputation, removal, and transformation.
Data Integration: This involves combining data from multiple sources to create a unified dataset. Data integration can be challenging as it requires handling data with different formats, structures, and semantics. Techniques such as record linkage and data fusion can be used for data integration.
Data Transformation: This involves converting the data into a suitable format for analysis. Common techniques used in data transformation include normalization, standardization, and discretization. Normalization is used to scale the data to a common range, while standardization is used to transform the data to have zero mean and unit variance. Discretization is used to convert continuous data into discrete categories.
Data Reduction: This involves reducing the size of the dataset while preserving the important information. Data reduction can be achieved through techniques such as feature selection and feature extraction. Feature selection involves selecting a subset of relevant features from the dataset, while feature extraction involves transforming the data into a lower-dimensional space while preserving the important information.
Data Discretization: This involves dividing continuous data into discrete categories or intervals. Discretization is often used in data mining and machine learning algorithms that require categorical data. Discretization can be achieved through techniques such as equal width binning, equal frequency binning, and clustering.
Data Normalization: This involves scaling the data to a common range, such as between 0 and 1 or -1 and 1. Normalization is often used to handle data with different units and scales. Common normalization techniques include min-max normalization, z-score normalization, and decimal scaling.
Data preprocessing plays a crucial role in ensuring the quality of data and the accuracy of the analysis results. The specific steps involved in data preprocessing may vary depending on the nature of the data and the analysis goals.

By performing these steps, the data mining process becomes more efficient and the results become more accurate.

Preprocessing in Data Mining:
Data preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format.



Steps Involved in Data Preprocessing:

1. Data Cleaning:

The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.

- **(a). Missing Data:**
  This situation arises when some data is missing in the data. It can be handled in various ways.
  Some of them are:
  1. **Ignore the tuples:**
     This approach is suitable only when the dataset we have is quite large and multiple values are missing within a tuple.

  2. **Fill the Missing values:**
     There are various ways to do this task. You can choose to fill the missing values manually, by attribute mean or the most probable value.

- **(b). Noisy Data:**
  Noisy data is a meaningless data that can't be interpreted by machines.It can be generated due to faulty data collection, data entry errors etc. It can be handled in following ways :
  1. **Binning Method:**
     This method works on sorted data in order to smooth it. The whole data is divided into segments of equal size and then various methods are performed to complete the task. Each segmented is handled separately. One can replace all data in a segment by its mean or boundary values can be used to complete the task.

  2. **Regression:**
     Here data can be made smooth by fitting it to a regression function.The regression used may be linear (having one independent variable) or multiple (having multiple

independent variables).

3. **Clustering:**
   This approach groups the similar data in a cluster. The outliers may be undetected or it will fall outside the clusters.

2. Data Transformation:

This step is taken in order to transform the data in appropriate forms suitable for mining process. This involves following ways:

1. **Normalization:**
   It is done in order to scale the data values in a specified range (-1.0 to 1.0 or 0.0 to 1.0)

2. **Attribute Selection:**
   In this strategy, new attributes are constructed from the given set of attributes to help the mining process.

3. **Discretization:**
   This is done to replace the raw values of numeric attribute by interval levels or conceptual levels.

4. **Concept Hierarchy Generation:**
   Here attributes are converted from lower level to higher level in hierarchy. For Example-The attribute "city" can be converted to "country".

3. Data Reduction:

Data reduction is a crucial step in the data mining process that involves reducing the size of the dataset while preserving the important information. This is done to improve the efficiency of data analysis and to avoid overfitting of the model. Some common steps involved in data reduction are:

Feature Selection: This involves selecting a subset of relevant features from the dataset. Feature selection is often performed to remove irrelevant or redundant features from the dataset. It can be done using various techniques such as correlation analysis, mutual information, and principal component analysis (PCA).

Feature Extraction: This involves transforming the data into a lower-dimensional space while preserving the important information. Feature extraction is often used when the original features are high-dimensional and complex. It can be done using techniques such as PCA, linear discriminant analysis (LDA), and non-negative matrix factorization (NMF).

Sampling: This involves selecting a subset of data points from the dataset. Sampling is often used to reduce the size of the dataset while preserving the important information. It can be done using techniques such as random sampling, stratified sampling, and systematic sampling.

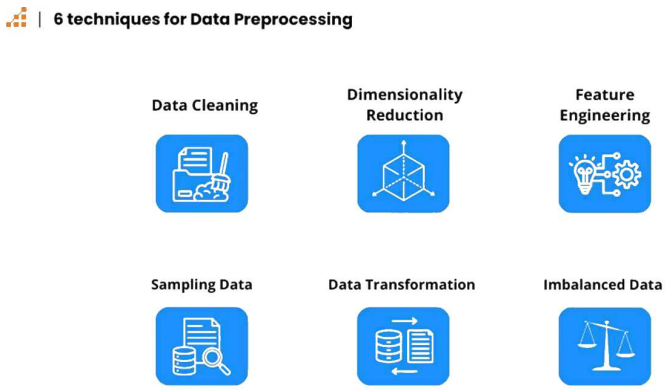
Clustering: This involves grouping similar data points together into clusters. Clustering is often used to reduce the size of the dataset by replacing similar data points with a representative centroid. It can be done using techniques such as k-means, hierarchical clustering, and density-based clustering.

Compression: This involves compressing the dataset while preserving the important information. Compression is often used to reduce the size of the dataset for storage and transmission purposes. It can be done using techniques such as wavelet compression, JPEG compression, and gzip compression.

NEED OF DATA PREPROCESSING

- **Handling Missing Data**: Real-world data often has missing values due to various reasons such as data corruption, human error, or system failures. Data preprocessing helps in identifying and handling missing data either by deleting the data points, imputing missing values using statistical methods, or predicting missing values using machine learning algorithms.
- **Dealing with Noisy Data**: Noisy data includes outliers or errors in the data that can skew the results and reduce the accuracy of machine learning models. Data preprocessing techniques such as smoothing, binning, or outlier detection and removal help in dealing with noisy data.
- **Normalization and Scaling**: Data preprocessing involves scaling numerical values to a standard range (like 0-1) or normalizing them to have a mean of 0 and a standard deviation of 1. This step ensures that all features contribute equally to the analysis and improves the performance of machine learning algorithms that are sensitive to the scale of input features (e.g., distance-based algorithms).
- **Handling Categorical Data**: Machine learning algorithms generally require input data to be in numerical form. Therefore, categorical data (variables that contain label values rather than numerical values) needs to be transformed into numerical values through techniques like one-hot encoding or label encoding.
- **Feature Selection and Engineering**: Data preprocessing involves selecting the most relevant features for the model training. Feature engineering techniques can also be applied to create new features that enhance the predictive power of machine learning algorithms.
- **Dimensionality Reduction**: In datasets with a large number of features, dimensionality reduction techniques like Principal Component Analysis (PCA) can be applied during preprocessing to reduce the number of variables while preserving important information.
- **Improving Model Performance**: By cleaning and preprocessing data properly, the overall quality of data is improved, leading to more accurate and reliable machine learning models. Proper preprocessing ensures that models are less likely to overfit or underfit the training data.

DATA PREPROCESSING TECHNIQUES



Now that you know more about the data preprocessing phase and why it's important, let's look at the main techniques to apply in the data, making it more usable for our future work. The techniques that we'll explore are:

- Data Cleaning
- Dimensionality Reduction
- Feature Engineering
- Sampling Data
- Data Transformation
- Imbalanced Data

Data Cleaning

One of the most important aspects of the data preprocessing phase is detecting and fixing bad and inaccurate observations from your dataset in order to improve its quality. This technique refers to identifying incomplete, inaccurate, duplicated, irrelevant or null values in the data. After identifying these issues, you will need to either modify or delete them. The strategy that you adopt depends on the problem domain and the goal of your project. Let's see some of the common issues we face when analyzing the data and how to handle them.

Example: **Data Cleaning and Preprocessing with Pandas:**

```
import pandas as pd
# Data cleaning example
df = pd.DataFrame({'A': [3, 2, None, 4], 'B': ['X', 'Y', 'Z', 'W']})
print(df)
df_cleaned = df.dropna()
print(df_cleaned)
```

Noisy Data

Usually, noisy data refers to meaningless data in your dataset, incorrect records, or duplicated observations. For example, imagine there is a column in your database for 'age' that has negative values. In this case, the observation doesn't make sense, so you could delete it or set the value as null (we'll cover how to treat this value in the "Missing Data" section).

Another case is when you need to remove unwanted or irrelevant data. For example, say you need to predict whether a woman is pregnant or not. You don't need the information about their hair color, marital status or height, as they are irrelevant for the model.

An outlier can be considered noise, even though it might be a valid record, depending on the outlier. You'll need to determine if the outlier can be considered noise data and if you can delete it from your dataset or not.

Missing Data

Another common issue that we face in real-world data is the absence of data points. Most machine learning models can't handle missing values in the data, so you need to intervene and

adjust the data to be properly used inside the model. There are different approaches you can take to handle it (usually called imputation):

Solution 1:

The simplest solution is to remove that observation. However, this is only recommended if:

1) You have a large dataset and a few missing records, so removing them won't impact the distribution of your dataset.

2) Most of the attributes of that observation are null, so the observation itself is meaningless.

Example: **Handling Missing Values with Pandas**

```
import pandas as pd
import numpy as np

# Create a DataFrame with missing values
data = {'A': [1, 2, np.nan, 4], 'B': ['X', 'Y', np.nan, 'Z']}
df = pd.DataFrame(data)
print(df)
# Handling missing values
df_filled = df.fillna(4)  # Fill missing values with 0
print(df_filled)
```

Structural Errors

Structural errors usually refer to some typos and inconsistencies in the values of the data.

For example, say that there is a marketplace and we sell shoes on our website. The data about the same product can be written in different ways by different sellers that sell the same shoes. Imagine that one of the attributes we have is the brand of the shoes, and aggregating the name of the brand for the same shoes we have: Nike, nike, NIKE. We need to fix this issue before giving this data to the model, otherwise, the model may treat them as different things. In this case, it's an easy fix: just transform all the words to lowercase. It may require more complex changes to fix inconsistencies and typos in other scenarios, though.

This issue generally requires manual intervention rather than applying some automated techniques.

Dimensionality Reduction

The dimensionality reduction is concerned with reducing the number of input features in training data.

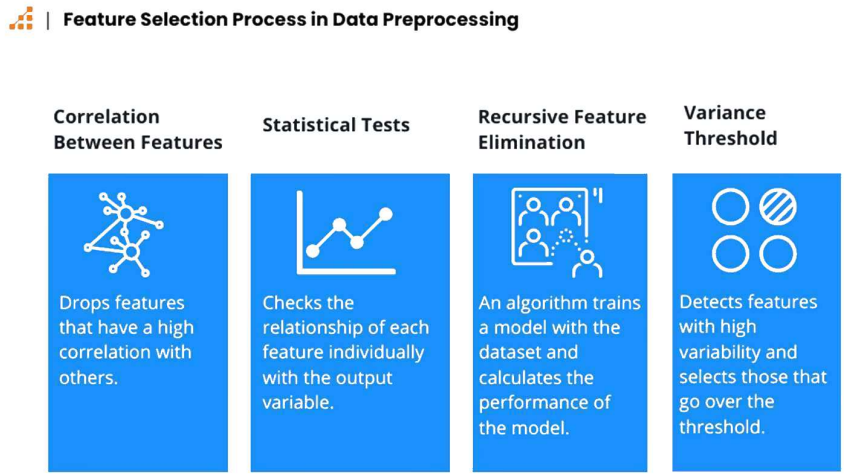
The Curse of Dimensionality in Your Dataset

With a real-world dataset, there are usually tons of attributes, and if we don't reduce this number, it may affect the model's performance later when we feed it this dataset. Reducing the

number of features while keeping as much variation in the dataset as possible will have a positive impact in many ways, such as:

- Requiring less computational resources
- Increasing the overall performance of the model
- Preventing overfitting (when the model becomes too complex and the model memorizes the training data, instead of learning, so in the test data the performance decreases a lot)
- Avoiding multicollinearity (high correlation of one or more independent variables). Also, applying this technique will reduce the noise data.

Feature Selection



Feature selection refers to the process of selecting the most important variables (features) related to your prediction variable, in other words, selecting the attributes that contribute most to your model. Here are some techniques for this approach that you can apply either automatically or manually:

IMPORTANCE OF DATA PREPROCESSING

The integrity of data analysis is highly dependent on the quality of data preprocessing. Data preprocessing determines the usability and interpretability of data, laying the groundwork for accurate machine learning and AI models.

Eliminating Errors

Cleaning is a pivotal data preprocessing technique. It allows you to eliminate errors, impute missing values, and rectify inconsistencies. For example, a customer dataset with redundant entries due to technical mistakes would undergo cleaning to ensure each customer record is unique and accurately represented.

Making Data Uniform

Normalization is comparable to establishing a level playing field, where disparate measures are adjusted to a uniform scale, enabling equitable comparisons. For instance, normalization can help you analyze the performance of stocks from different countries despite stock prices being available in various currencies and scales. With normalization techniques such as min-max, you can convert all stock prices into a common currency, for example, USD, and then apply a min-max scaling to compare the relative performance of stocks on a uniform scale.

Finding Hidden Patterns

Diligent preprocessing can reveal concealed patterns and insights. A marketing team analyzing social media data can identify peak engagement times aligned with spam activity. However, excluding anomalies through data cleaning will allow you to pinpoint genuine peak engagement periods and optimize strategy.

Big Data Preprocessing

As datasets grow in size and complexity, preprocessing becomes even more critical. Big data has a large volume, is heterogeneous, and needs to be processed rapidly. Preprocessing transforms raw big data into a cleaner, more structured format, removing noise and making it easier to process.

Similarly, advanced techniques such as parallel processing, distributed computing, and automated preprocessing pipelines are indispensable for processing big data effectively.

DATA PREPROCESSING TOOLS

Data preprocessing tools simplify how you interact with extensive data, making it easier to shape and polish complex data. Some data preprocessing tools that make this transformation possible are:

- **Pandas**: This Python library offers a wide array of functions for handling data, making it ideal for cleaning, filtering, and aggregating large datasets.
- **Scikit-learn**: Scikit-learn is equipped to handle everything from feature scaling to encoding categorical variables, ensuring your data is in the best shape for modeling.
- **OpenRefine**: Designed for the challenges of messy data, OpenRefine is a standalone tool that cleans and transforms data. It's beneficial for standardizing data formats and enriching datasets with information from external sources.

DATA WAREHOUSE

A data warehouse architecture is a method of defining the overall architecture of data communication processing and presentation that exist for end-clients computing within the

enterprise. Each data warehouse is different, but all are characterized by standard vital components.

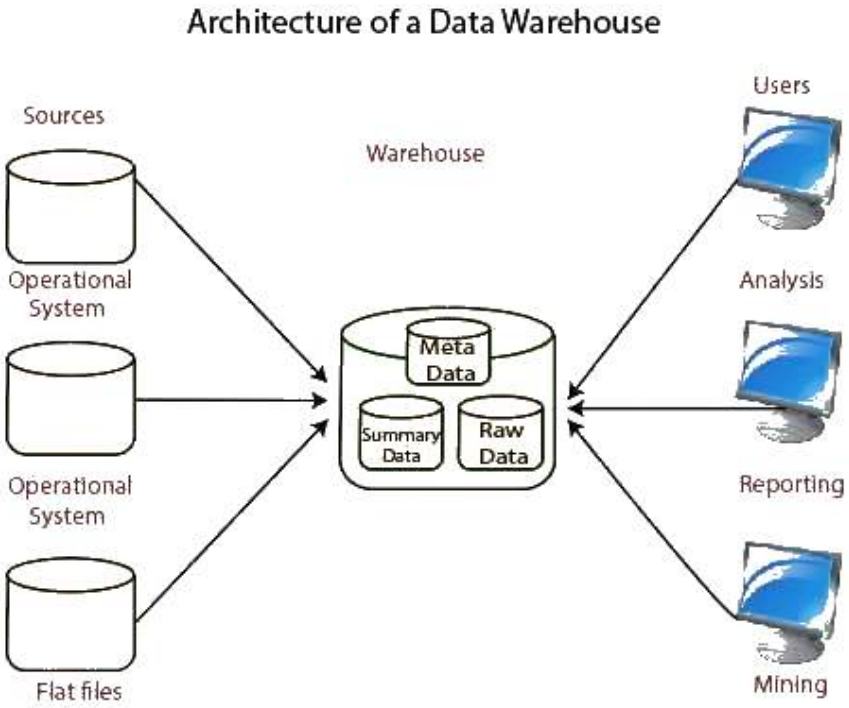
Production applications such as payroll accounts payable product purchasing and inventory control are designed for online transaction processing **(OLTP)**. Such applications gather detailed data from day to day operations.

Data Warehouse applications are designed to support the user ad-hoc data requirements, an activity recently dubbed online analytical processing (OLAP). These include applications such as forecasting, profiling, summary reporting, and trend analysis.

Production databases are updated continuously by either by hand or via OLTP applications. In contrast, a warehouse database is updated from operational systems periodically, usually during off-hours. As OLTP data accumulates in production databases, it is regularly extracted, filtered, and then loaded into a dedicated warehouse server that is accessible to users. As the warehouse is populated, it must be restructured tables de-normalized, data cleansed of errors and redundancies and new fields and keys added to reflect the needs to the user for sorting, combining, and summarizing data.

Data warehouses and their architectures very depending upon the elements of an organization's situation.

Data Warehouse Architecture



Operational System

An **operational system** is a method used in data warehousing to refer to a **system** that is used to process the day-to-day transactions of an organization.

Flat Files

A **Flat file** system is a system of files in which transactional data is stored, and every file in the system must have a different name.

Meta Data

- A set of data that defines and gives information about other data.
- Meta Data used in Data Warehouse for a variety of purpose, including:
- Meta Data summarizes necessary information about data, which can make finding and work with particular instances of data more accessible. For example, author, data build, and data changed, and file size are examples of very basic document metadata.
- Metadata is used to direct a query to the most appropriate data source.

Lightly and highly summarized data

The area of the data warehouse saves all the predefined lightly and highly summarized (aggregated) data generated by the warehouse manager.

The goals of the summarized information are to speed up query performance. The summarized record is updated continuously as new information is loaded into the warehouse.

End-User access Tools

The principal purpose of a data warehouse is to provide information to the business managers for strategic decision-making. These customers interact with the warehouse using end-client access tools.

The examples of some of the end-user access tools can be:

- Reporting and Query Tools
- Application Development Tools
- Executive Information Systems Tools
- Online Analytical Processing Tools
- Data Mining Tools

NEED OF DATA WAREHOUSE

- **Centralized Data Storage**: Data warehouses consolidate data from multiple sources into a single repository. This centralized storage simplifies data management and allows for easier access and analysis.

  Example: A retail company operates multiple stores across different regions. Each store collects sales data independently. By consolidating all sales data into a data

warehouse, the company can analyze overall sales trends, identify top-performing products across all stores, and make strategic decisions based on unified data.

- **Data Integration**: Organizations often have data spread across different systems and departments. A data warehouse integrates this disparate data into a unified format, making it easier to analyze relationships and trends across the organization.

  Example: A healthcare organization manages patient data across various departments such as admissions, diagnostics, and treatment. Integrating this data into a data warehouse allows healthcare administrators to analyze patient demographics, treatment outcomes, and resource utilization across the entire organization, leading to improved patient care and operational efficiency.

- **Historical Analysis**: Data warehouses store historical data over extended periods. This historical perspective enables organizations to perform trend analysis, track performance over time, and identify long-term patterns.

  Example: An e-commerce platform stores years of transactional data in a data warehouse. By analyzing historical sales trends, the platform can predict seasonal demand patterns, optimize inventory levels, and personalize marketing campaigns based on past customer behavior.

- **Decision Support**: Data warehouses provide a solid foundation for decision-making. By organizing and structuring data, they support complex queries and reporting, helping executives and managers make informed decisions based on reliable data.

  Example: A financial institution uses a data warehouse to analyze customer financial transactions, credit histories, and investment portfolios. This allows financial advisors to provide personalized investment recommendations, assess risk profiles, and improve client satisfaction through data-driven insights.

- **Data Quality**: Data warehouses often include processes for data cleaning and normalization. This improves data quality by resolving inconsistencies and errors, ensuring that analyses are based on accurate and reliable information.

  Example: A telecommunications company aggregates customer interaction data from various touchpoints (call centers, website, mobile apps) into a data warehouse. By cleansing and standardizing this data, the company ensures that customer analytics are accurate and reliable, leading to better customer service and retention strategies.

- **Scalability**: Data warehouses are designed to handle large volumes of data efficiently. They can scale as the organization grows, accommodating increasing data volumes and user demands without compromising performance.

  Example: A technology firm experiences rapid growth in user data from its mobile application. Scaling its data warehouse infrastructure allows the firm to handle increasing volumes of user-generated data, analyze usage patterns, and optimize app performance based on real-time insights.

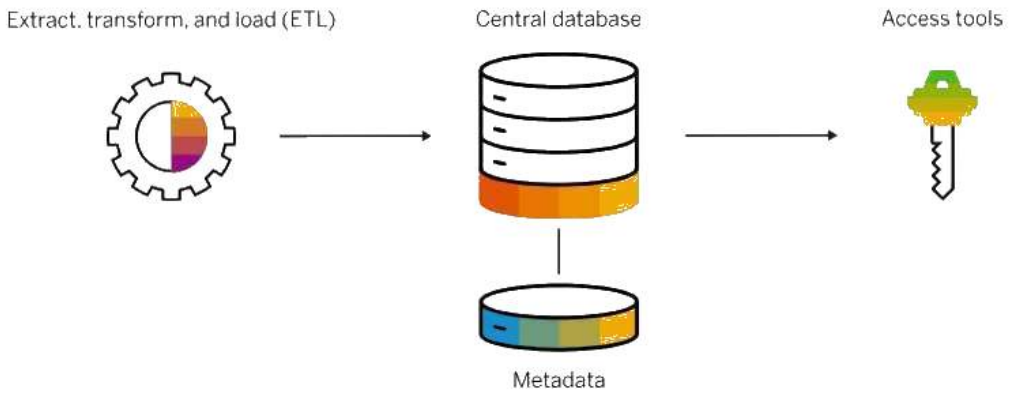
- **Business Intelligence**: Data warehouses are integral to business intelligence (BI) efforts. They serve as the backbone for BI tools and applications, providing the data necessary for generating reports, dashboards, and other analytical insights.

  Example: A manufacturing company uses a data warehouse to analyze production metrics, supply chain efficiency, and product quality across its global operations. Business intelligence tools leverage this data to generate real-time dashboards and reports, enabling executives to make timely decisions that enhance operational performance and profitability.

- **Regulatory Compliance**: Many industries have regulatory requirements for data storage and reporting. A data warehouse can help organizations comply with these regulations by providing a structured approach to data management and audit trails.

KEY COMPONENTS OF A DATA WAREHOUSE?

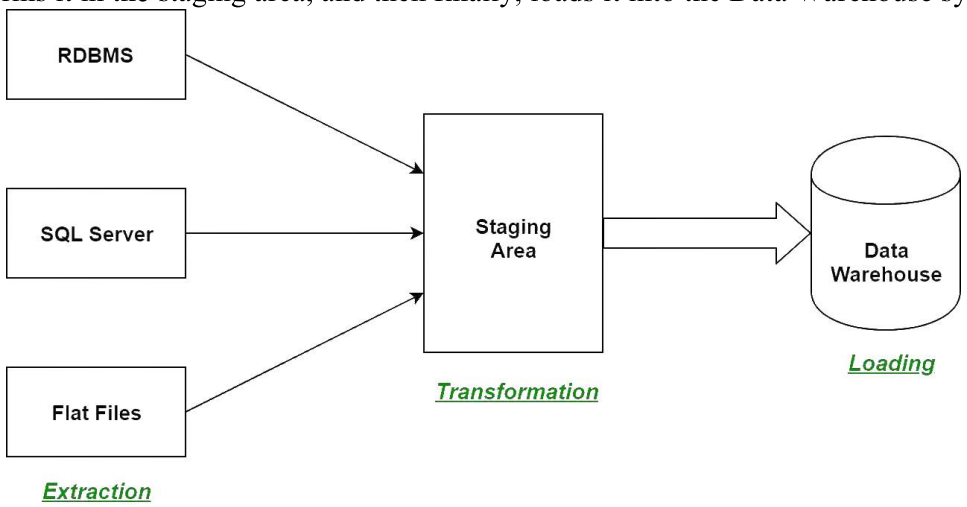
A typical data warehouse has four main components: a central database, ETL (extract, transform, load) tools, metadata, and access tools. All of these components are engineered for speed so that you can get results quickly and analyse data on the fly.



1. **Central database:** A database serves as the foundation of your data warehouse. Traditionally, these have been standard relational databases running on premise or in the cloud. But because of Big Data, the need for true, real-time performance, and a drastic reduction in the cost of RAM, in-memory databases are rapidly gaining in popularity.
2. **Data integration:** Data is pulled from source systems and modified to align the information for rapid analytical consumption using a variety of data integration approaches such as ETL (extract, transform, load) and ELT as well as real-time data replication, bulk-load processing, data transformation, and data quality and enrichment services.
3. **Metadata:** Metadata is data about your data. It specifies the source, usage, values, and other features of the data sets in your data warehouse. There is business metadata, which adds context to your data, and technical metadata, which describes how to access data – including where it resides and how it is structured.
4. **Data warehouse access tools:** Access tools allow users to interact with the data in your data warehouse. Examples of access tools include: query and reporting tools, application development tools, data mining tools, and OLAP tools.

5. ETL (extract, transform, load)

ETL is a process in Data Warehousing and it stands for **Extract**, **Transform** and **Load**. It is a process in which an ETL tool extracts the data from various data source systems, transforms it in the staging area, and then finally, loads it into the Data Warehouse system.



Let us understand each step of the ETL process in-depth:

1. **Extraction:**
   The first step of the ETL process is extraction. In this step, data from various source systems is extracted which can be in various formats like relational databases, No SQL, XML, and flat files into the staging area. It is important to extract the data from various source systems and store it into the staging area first and not directly into the data warehouse because the extracted data is in various formats and can be corrupted also. Hence loading it directly into the data warehouse may damage it and rollback will be much more difficult. Therefore, this is one of the most important steps of ETL process.
2. **Transformation:**
   The second step of the ETL process is transformation. In this step, a set of rules or functions are applied on the extracted data to convert it into a single standard format. It may involve following processes/tasks:
   - Filtering – loading only certain attributes into the data warehouse.
   - Cleaning – filling up the NULL values with some default values, mapping U.S.A, United States, and America into USA, etc.
   - Joining – joining multiple attributes into one.
   - Splitting – splitting a single attribute into multiple attributes.
   - Sorting – sorting tuples on the basis of some attribute (generally key-attribute).
3. **Loading:**
   The third and final step of the ETL process is loading. In this step, the transformed data is finally loaded into the data warehouse. Sometimes the data is updated by loading into the data warehouse very frequently and sometimes it is done after longer but regular intervals. The rate and period of loading solely depends on the requirements and varies from system to system.

TYPES OF DATA WAREHOUSE

Enterprise Data Warehouse (EDW)

- This type of enterprise data warehouse is optimized for fast querying and analytical purposes. It retrieves data from distinct resources such as transactional setups, operational systems, and other databases.
- An EDW is based on a multi-dimensional data model and is ideally customized for an organization's needs. Deploying the enterprise data warehousing benefits offers decision support and a unified approach to data visualization.

Operational Data Store (ODS)

- As the name states, this type of data warehousing model works for the operational needs of an organization. It typically collects and analyzes data in real-time.
- When the existing data warehouse systems lack reporting results, the role of ODS comes in. Operational data warehousing helps in the daily activities of a business, such as employee records, Customer Relationship Management, etc.

Data Mart

- A smaller version of a data warehouse is a data mart. It's developed for a specific unit, segment, and department operating within a business.
- Every section of the organization uses a data mart as a centralized database to store, access, & analyze data. The stored data in a data mart is further migrated to ODS, which later moves it to EDW, i.e., Enterprise Data Warehouse.

Cloud Data Warehouses

- When a data warehouse is hosted on a SaaS in cloud computing such as Amazon Web Services (AWS), it brings a barrage of advantages for an organization.
- Data warehousing in the cloud results in higher scalability and reduced IT expenses on internal resources. Companies can optimize their databases using cloud-based data warehouses & scale up or down their systems as required.

Big Data Warehouses

- This type of data warehouse model embraces the role of big data implementation. It processes a large volume of data, whether it's in unstructured, semi-structured, or structured form.
- Big data warehouses use non-relational database frameworks along with a schema-on-read approach to store data.

DATA WAREHOUSE TOOLS

1. Snowflake

Snowflake is an enterprise-grade cloud database that offers fast, secure, and reliable access to data via APIs and direct SQL queries. It's designed with simple factors and used by all levels of developers.

The tool comes with native support for NoSQL databases like MongoDB and Hadoop, as well as data processing services like Spark and Presto.

2. Google BigQuery

BigQuery is a fully managed, cloud-based data warehouse that allows you to analyze massive datasets using SQL. It can handle petabytes of data and billions of rows, and it's fast enough to enable real-time data exploration.
You can use BigQuery for ETL (extract, transform, load), data warehouse applications, and advanced analytics. It is great for handling large datasets because it automatically scales with your needs. When you run a query, the results are automatically loaded into your dataset so that they're ready to be analyzed.

3. Microsoft Azure

This tool offers its take on a cloud-based data warehousing solution called Azure Data Warehouse (ADW). It was created specifically for Microsoft users so that there are no compatibility issues when connecting your existing systems with this tool.

Like other similar offerings on this list, ADW supports both relational and non-relational databases as well as allows users to perform complex computations on their data sets in real-time through parallel processing capabilities.

4. Amazon Redshift

Amazon Redshift is another popular choice among enterprises looking for affordable cloud solutions. This tool offers high performance at scale for both batch and real-time workloads by using parallelism across nodes with a columnar storage format optimized specifically for analytics workloads.

It also comes with built-in security features such as encryption at rest, user authentication through Amazon Web Services Identity Provider (IAM), and audit logging.

5.IBM DB2 Warehouse

IBM DB2 Warehouse is a data warehousing tool that provides information management, data quality, and business intelligence. It uses the same technology as IBM DB2 Universal Database and InfoSphere Warehouse, which means it can support large-scale data warehousing environments with high availability and security.

IBM DB2 Warehouse also supports several industry-standard interfaces, including ODBC/JDBC, SQL/JDBC, XML/SQL, and more. The tool has been designed to work well with other IBM products, so you

6.Oracle Autonomous Warehouse

This tool is designed for managing large amounts of unstructured data and can help ensure that your business stays on top of all its important information.

Oracle Autonomous Warehouse is highly customizable and easy to use, making it a great option if you're looking for something easy enough for beginners but powerful enough for experts.

ADVANTAGES AND DISADVANTAGES OF DATA WAREHOUSE

Advantages of a Data Warehouse

Delivers Enhanced Business Intelligence

A significant benefit of data warehouses is their capacity to enhance business intelligence. They provide a unified view of data from various sources, making it easier for organizations to access and analyze the information they need for decision-making. By consolidating data in one place, data warehouses simplify the often complex process of data integration.

Ensures Data Quality and Consistency

Data quality and consistency are very important in data management. Data warehouses implement processes for data cleansing and transformation, which help in eliminating duplicate or inconsistent data. This ensures that the data stored in the warehouse is accurate and reliable. With high data quality standards in place, organizations can have confidence in the insights derived from their data warehouse.

Saves Time and Money

Data warehouse can is cost-effective in terms of time and money. They enable organizations to process and analyze large volumes of data efficiently. When data handling is streamlined and optimized, it reduces the need for manual data preparation and management, allowing employees to focus on more value-added tasks.

Tracks Historically Intelligent Data

Historical data is a goldmine for organizations. Data warehouses store historical data, facilitating trend analysis and a deeper understanding of business performance over time. By accessing and analyzing historical data, organizations can identify patterns, forecast future trends, and make informed strategic decisions. This historical perspective is invaluable for long-term planning and growth.

Disadvantages of a Data Warehouse

Additional Reporting

Additional work is required for using data warehouses because data stored in a warehouse is structured, meaning that to create reports, we must design and maintain predefined queries and data models. This is time-consuming and delays in generating essential reports.

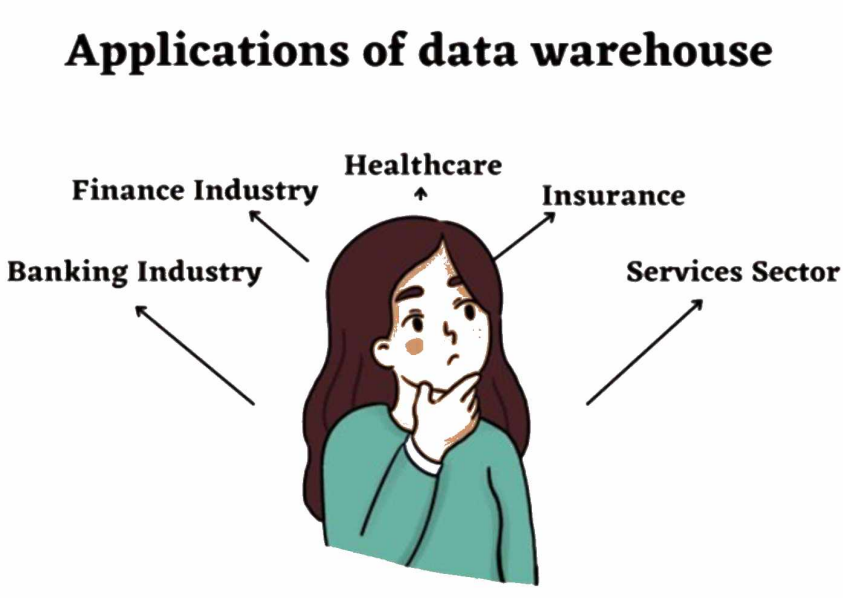
Inflexibility and Homogenization of Data
Data warehouses depend on structured data, which is organized into predefined formats. This can be a problem when handling unstructured data, such as social media posts or multimedia content. Unstructured data doesn't fit well within the structured framework of a data warehouse. If the organization deals with a lot of unstructured data, it will face problems in integrating the data effectively.

Ownership Concerns
Data warehousing systems often involve multiple departments and teams. This can lead to ownership issues. Determining who is responsible for various aspects of the data warehouse, including data governance and quality, can become a complex task. Lack of clear ownership can result in inefficiencies and data management problems.

Demands for Large Amounts of Resources
Implementing and maintaining a data warehouse can be resource-intensive. The organization has to invest in hardware, software, and skilled personnel to set up and manage the system. This upfront investment can be substantial and may not be feasible for small or resource-constrained organizations. The ongoing costs of maintaining a data warehouse can also add up over time.

Hidden Issues Consume Time
Data warehouses are not immune to hidden problems. Issues related to data quality, transformation errors, and data integration challenges can emerge, and uncovering these problems can consume valuable time and resources. Such hidden issues can also lead to inaccurate reports and analyses, affecting decision-making.

APPLICATIONS OF DATA WAREHOUSE



1. Banking Industry
Bankers can better manage all of their available resources with the right Data Warehousing solution. They can better analyze consumer data, government regulations, and market trends to facilitate better decision-making.
2. Finance Industry
Similar to the applications seen in banking, they mainly revolve around evaluation and trends of customer expenses which aid in maximizing the profits earned by their clients.
3. Consumer Goods Industry
They are used to predict consumer trends, inventory management, and market and advertising research. An in-depth analysis of sales and production is also carried out. Apart from these, information is exchanged between business partners and clientele.
4. Government and Education

- The federal government uses the warehouses for compliance research, while the state government uses them for human resources services such as recruitment and accounting services such as payroll management.
- The government uses data warehouses to store and analyze tax records, health policy records, and providers. Their entire criminal law database is also connected to the state's data warehouse. Illegal activity is predicted from the patterns, trends, and results of analyzing historical data associated with past criminals.
- Universities employ data warehouses to collect information for grant proposals, student demographic analysis, and human resource management. Most colleges' financial departments, including the Financial Aid department, rely on data warehouses.

5. Healthcare
The healthcare industry is another important application of data warehouses. All clinical, financial, and personnel data is saved in the warehouse, and analysis is performed to provide useful insights into allocating resources effectively.
6. Hospitality Industry
Hotel and restaurant services, automobile rental services, and vacation home services dominate this market. They employ warehousing services to create and assess their ad and promotion programs, which target customers based on their feedback and travel patterns.
7. Insurance
In the insurance industry, there's a saying that goes, "If it ain't broke, don't. "Insurance is not something that can be purchased. It can only be purchased. "Apart from keeping track of existing participants, the warehouses are largely used to evaluate data patterns and customer trends. Warehouses can also be used to create customized consumer offers and promotions.
8. Manufacturing and Distribution Industry
Manufacturing and distribution companies may gather all of their data under one roof with the help of a good data warehousing system, predict market changes, analyze the latest trends, view development areas, and finally make result-driven decisions.
9. The Retailers

- Retailers act as go-betweens for producers and customers. In order to ensure their continuous presence on the market, they must keep records of both parties.
- They employ warehouses to keep track of merchandise, advertising promotions, and consumer purchasing patterns. They also analyze sales to determine fast-selling and slow-selling product lines and determine their shelf space through elimination.

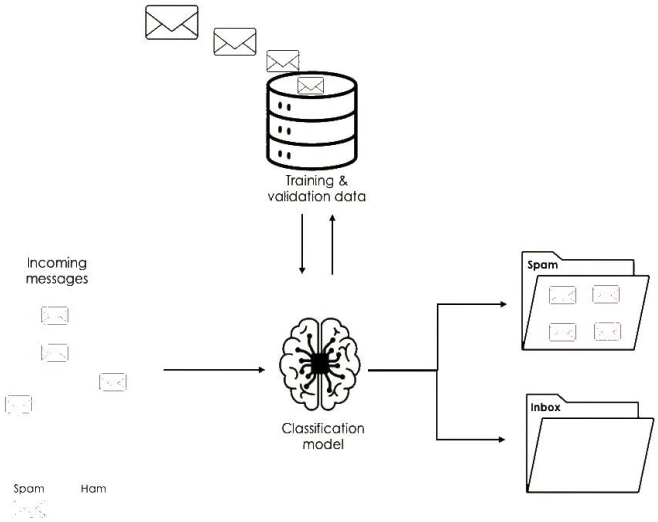
10. Services Sector
In the service industry, data warehouses are used to keep track of financial records, revenue trends, consumer profiling, resource management, and human resources.

11. Telephone Industry

- The telephone sector deals with both offline and online data, resulting in a large amount of historical data to be consolidated and integrated.
- A data warehouse is also required to study fixed assets, monitor customer calling patterns for salespeople to push advertising campaigns, and track consumer queries.

Unit-III Classification

Classification is a supervised machine learning method where the model tries to predict the correct label of a given input data. In classification, the model is fully trained using the training data, and then it is evaluated on test data before being used to perform prediction on new unseen data.



Types of Learners in Classification

We have two types of learners in respective to classification problems −

- **Lazy Learners**

  As the name suggests, such kind of learners waits for the testing data to be appeared after storing the training data. Classification is done only after getting the testing data. They spend less time on training but more time on predicting. Examples of lazy learners are K-nearest neighbor and case-based reasoning.

- **Eager Learners**

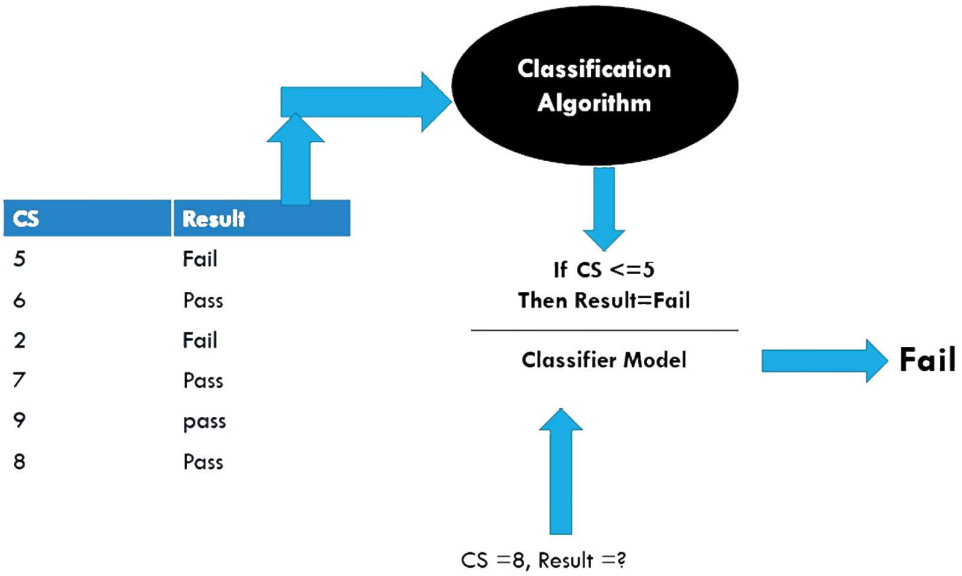
  As opposite to lazy learners, eager learners construct classification model without waiting for the testing data to be appeared after storing the training data. They spend more time on training but less time on predicting. Examples of eager learners are Decision Trees, Naïve Bayes and Artificial Neural Networks (ANN).

CLASSIFICATION REQUIREMENTS

Classification requirements

- Classification of unknown tuples or objects using the constructed model.
- Comparison of a class label of the test sample with the resultant class label

- Comparison of the accuracy of the model
- Model Construction
- Model Usage



Nearest Neighbour classifier

- K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
- It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

$$\text{Euclidean Distance between } A_1 \text{ and } B_2 = \sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$$

Example:

Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

How does K-NN work?

The K-NN working can be explained on the basis of the below algorithm:

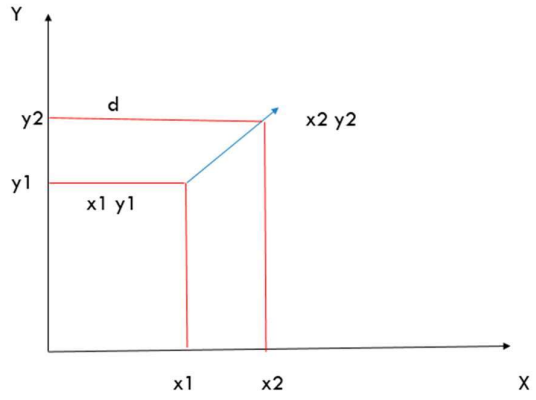
- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

Example K-NN Algorithm

Ex. Predicting Movie Genre

IMDB Rating	Duration	Genre
8.0 (Mission Impossible)	160	Action
6.2 (Gadar 2)	170	Action
7.2(Rocky and Rani	168	Comedy
8.2(OMG 2)	155	comedy

Now predict the genre of "Barbie" movie with IMDB rating 7.4 and duration 114 minutes



$$d(p,q) = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$$

Solution:

Step1 : Calculate Distance

Calculate The Euclidean distance between the new movie and each movie in the dataset

Distance to(8.0,160)=$\sqrt{((7.4 - 8.0)^2 + (114 - 160)^2}$ =$\sqrt{(0.36 + 2116} = 46.00$

Distance to(6.2,160)=$\sqrt{((7.4 - 6.2)^2 + (114 - 170)^2}$ =$\sqrt{(1.44 + 3136} = 56.01$

Distance to(7.2,168)=$\sqrt{((7.4 - 7.2)^2 + (114 - 168)^2}$ =$\sqrt{(0.04 + 2916} = 54.01$

Distance to(8.2,155)=$\sqrt{((7.4 - 8.2)^2 + (114 - 155)^2}$ =$\sqrt{(0.64 + 1681} = 41.00$

Step 2:Select K Nearest Neighbors

Step 3:Majority Voting (Classification)

How to select the value of K in the K-NN Algorithm.

- There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
- A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
- Large values for K are good, but it may find some difficulties.

Advantages of KNN Algorithm:

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

NAÏVE BAYES CLASSIFIER ALGORITHM

- Naïve bayes algorithm is supervised learning algorithm, which is based on bayes theorem and used for solving classification problems.
- It is mainly used in text classification that includes a high-dimensional training dataset.
- Naïve bayes classifier is one of the simple and most effective classification algorithm which helps in building the fast machine learning models that can make quick predictions.
- It is probabilistic classifier, which means it predicts on the basis of the probability of an object.
- Some popular examples of naïve bayes algorithm are Spam Filtrations, Sentimental Analysis, and classifying articles.

Bayes theorem

Bayes theorem is also known as Bayes Rule or Bayes Law which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability

The Formula for Bayes is as follows

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

Where,

- P(AIB) is Posterior Probability: Probability of hypothesis A on the observed event B.
- P(B|A) Is Likelihood Probability: Probability of the evidence given that the probability of hypothesis is true.
- P(A) is Prior probability: Probability of hypothesis before observing the evidence.
- P(B) is Marginal Probability: Probability of Evidence.

Types of naïve bayes model:

There are three types of Naive Bayes Model, which are given below:

Bernoulli: The Bernoulli classifier works similar to the Multinomial classifier, but the predictor variables are the independent Booleans variables. Such as if a particular word is present or not in a document. This model is also famous for document classification tasks.

P(Success)=P

P(Failure)-q-1-p

X=1[success]

X=0[failure]

X has Bernoulli distribution

P(X=x)=P^x(1-q)^1-x

Multinomial: The Multinomial Naïve Bayes classifier is used when the data is multinomial distributed. It is primarily used for document classification problems, it means a particular document belongs to which category such as Sports, Politics, education,etc. The classifier uses the frequency of words for the predictors.

Descrete Distribution.

P(x1=x1,…,xk=xk)

$\frac{n!}{x1!-xk!}p1^{x1}\ldots\ldots p^{xk}$

BG	O	A	B	AB
P	0.44	0.42	0.10	0.04

6 INDIAN

1:0, 2:A, 2:B, 1:AB

P(x1=1,x2=2 ,x3=2, x4=1

1 2 2 1

$\frac{6!}{1!2!2!1!}$ 0.44 0.42 0.10 0.04

Gaussian: The Gaussian model assumes that features follow a normal distribution. This means if predictors take continuous values instead of discrete, then the model assumes that these values are sampled from the Gaussian distribution.

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

$-\infty < x < \infty$

Example:

Fruit ={yellow, Sweet, long}

Fruit	Yellow	Sweet	Long	Total
Orange	350	450	0	**650**
Banana	400	300	350	**400**
Others	50	100	50	**150**
Total	**800**	**850**	**400**	**1200**

$$\mathbf{P(A \mid B)=\frac{P(B \mid A).P(A)}{P(B)}}$$

For Orange

$$\mathbf{P(Yellow \mid Orange)=\frac{P(Orange \mid Yellow).P(Yellow)}{P(Orange)} =\frac{\left(\frac{350}{800}\right)*(800/1200)}{(650/1200)} =0.5}$$

$$\mathbf{P(Sweet \mid Orange)=\frac{P(Orange|Sweet).P(Sweet)}{P(Orange)} =\frac{\left(\frac{450}{850}\right)*(850/1200)}{(650/1200)} =0.69}$$

$$\mathbf{P(Long \mid Orange)=\frac{P(Orange|Long).P(Long)}{P(Orange)} =\frac{\left(\frac{0}{400}\right)*(400/1200)}{(650/1200)} =0.0}$$

$$P(A \mid B)=\frac{P(B \mid A).P(A)}{P(B)}$$

For Banana

$$P(\text{Yellow} \mid \text{Banana})=\frac{P(banana|Yellow).P(Yellow)}{P(banana)}=\frac{\left(\frac{400}{800}\right)*(800/1200)}{(400/1200)}=1$$

$$P(\text{Sweet} \mid \text{Banana})=\frac{P(Banana|Sweet).P(Sweet)}{P(Banana)}=\frac{\left(\frac{300}{850}\right)*(850/1200)}{(400/1200)}=0.75$$

$$P(\text{Long} \mid \text{Banana})=\frac{P(\text{Banana}|Long).P(Long)}{P(\text{Banana})}=\frac{\left(\frac{350}{400}\right)*(400/1200)}{(400/1200)}=0.87$$

For Banana

$$P(\text{Yellow} \mid \text{Other})=\frac{P(\text{Other}|Yellow).P(Yellow)}{P(\text{Other})}=\frac{\left(\frac{50}{800}\right)*(800/1200)}{(150/1200)}=0.33$$

$$P(\text{Sweet} \mid \text{Other})=\frac{P(\text{Other}|Sweet).P(Sweet)}{P(\text{Other})}=\frac{\left(\frac{100}{850}\right)*(850/1200)}{(150/1200)}=0.66$$

$$P(\text{Long} \mid \text{Banana})=\frac{P(\text{Other}/Long).P(Long)}{P(\text{Other})}=\frac{\left(\frac{80}{400}\right)*(400/1200)}{(150/1200)}=0.33$$

P(fruit | Orange) = 0.53*0.69*0 = 0

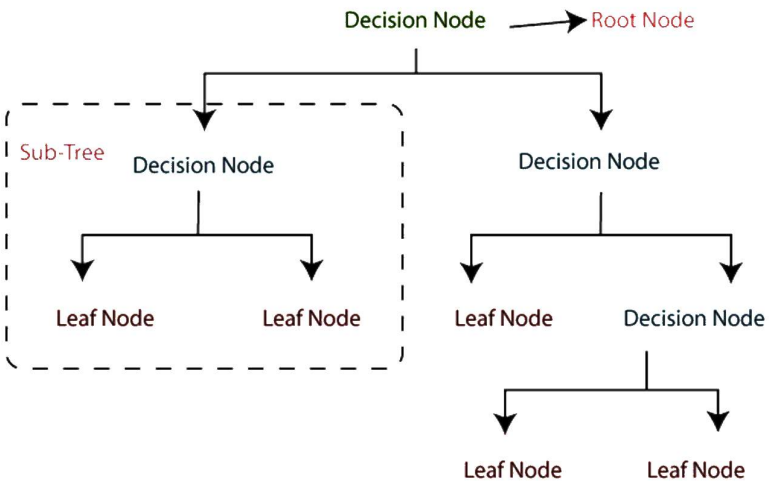
P(Fruit | Banana)= 1 * 0.75 *0.87 =0.65

P(Fruit | Others)= 0.33*0.66 *0.33 0.072

Final Classification: P(Fruit | Banana)= 1 * 0.75 *0.87 =0.65

Decision Tree:

* Decision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where **internal nodes represent the features of a dataset, branches represent the decision rules** and **each leaf node represents the outcome.**
* In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
* The decisions or the test are performed on the basis of features of the given dataset.
* ***It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.***
* It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
* In order to build a tree, we use the **CART algorithm,** which stands for **Classification and Regression Tree algorithm.**
* A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.



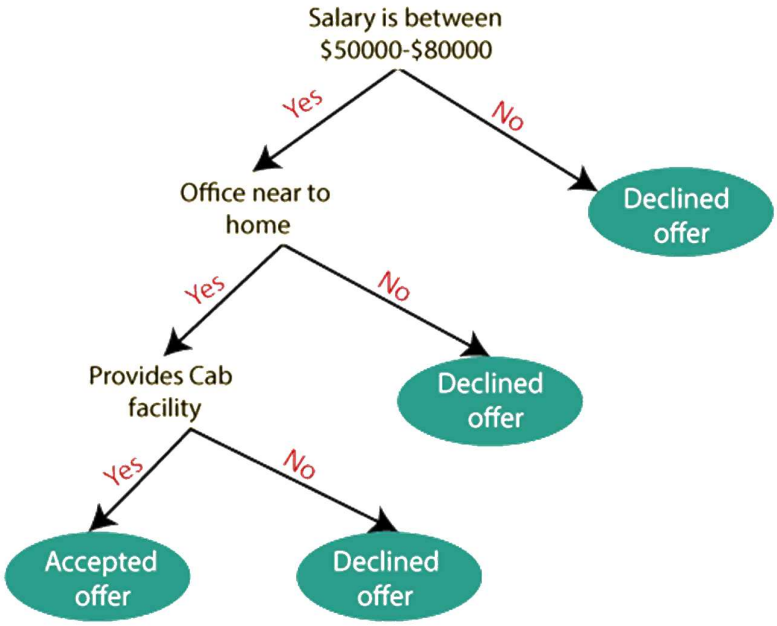
Decision Tree Terminology

* **Root Node:** Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
* **Leaf Node:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
* **Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
* **Branch/Sub Tree:** A tree formed by splitting the tree.
* **Pruning:** Pruning is the process of removing the unwanted branches from the tree.
* **Parent/Child node:** The root node of the tree is called the parent node, and other nodes are called the child nodes.

Algorithm

* **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
* **Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**
* **Step-3:** Divide the S into subsets that contains possible values for the best attributes.
* **Step-4:** Generate the decision tree node, which contains the best attribute.
* **Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.
* **Example:** Suppose there is a candidate who has a job offer and wants to decide whether he should accept the offer or Not. So, to solve this problem, the decision tree starts with the root node (Salary attribute by ASM). The root node splits further into the next decision node (distance from the office) and one leaf node based on the corresponding

labels. The next decision node further gets split into one decision node (Cab facility) and one leaf node. Finally, the decision node splits into two leaf nodes (Accepted offers and Declined offer). Consider the below diagram:



Attribute Selection Measures:

* While implementing a Decision tree, the main issue arises that how to select the best attribute for the root node and for sub-nodes. So, to solve such problems there is a technique which is called as **Attribute selection measure or ASM.** By this measurement, we can easily select the best attribute for the nodes of the tree. There are two popular techniques for ASM, which are:
* **Information Gain**
* **Gini Index**

Information Gain:

* Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute.
* It calculates how much information a feature provides us about a class.
* According to the value of information gain, we split the node and build the decision tree.
* A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first. It can be calculated using the below formula:
* Information Gain= Entropy(S)- [(Weighted Avg) *Entropy(each feature)
* **Entropy:** Entropy is a metric to measure the impurity in a given attribute. It specifies randomness in data. Entropy can be calculated as:

* Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no)**Where,**
* **S= Total number of samples**
* **P(yes)= probability of yes**
* **P(no)= probability of no**

Gini Index:

* Gini index is a measure of impurity or purity used while creating a decision tree in the CART(Classification and Regression Tree) algorithm.
* An attribute with the low Gini index should be preferred as compared to the high Gini index.
* It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.
* Gini index can be calculated using the below formula:
* Gini Index= $1- \sum_j P_j^2$

Example:

Day	**Weather**	**Temp**	**Humidity**	**Wind**	**Play Football**
Day 1	sunny	Hot	High	weak	no
Day 2	sunny	Hot	High	strong	no
Day 3	cloudy	Hot	High	weak	yes
Day 4	rain	Mild	High	weak	yes
Day 5	rain	Cool	Normal	weak	yes
Day 6	rain	Cool	Normal	strong	no
Day 7	cloudy	Cool	Normal	strong	yes
Day 8	sunny	Mild	High	weak	no
Day 9	sunny	Cool	Normal	weak	yes
Day 10	rain	Mild	Normal	weak	yes
Day 11	sunny	Mild	Normal	strong	yes

Day 12	cloudy	Mild	High	strong	yes
Day 13	cloudy	Hot	Normal	weak	yes
Day 14	rain	Mild	High	strong	no

Calculate IG of Weather(ID3 Algorithm iterative dicomiser)

Step1 : Entropy of entire dataset

S{+9,-5}=$-\frac{9}{14}log_2\frac{9}{14}$-$\frac{5}{14}log_2\frac{5}{14}$=0.94

Step 2 Entropy of all attributes:

Entropy of Sunny{+2,-3}=-$\frac{2}{5}log_2\frac{2}{5}$-$\frac{3}{5}log_2\frac{3}{5}$=0.97

Entropy of Cloudy{+4,-0}=-$\frac{4}{4}log_2\frac{4}{4}$-$\frac{0}{4}log_2\frac{0}{4}$=0

Entropy of Rain{+3,-2}=-$\frac{3}{5}log_2\frac{3}{5}$-$\frac{2}{5}log_2\frac{2}{5}$=0.97

Information Gain = Entropy(Whole Data)

$-\frac{5}{14}Ent(S)-\frac{4}{14}$Ent(C)-$\frac{5}{14}$Ent(R)

=0.246

Calculate IG of Temp(ID3 Algorithm iterative dicomiser)

Step1 : Entropy of entire dataset

S{+9,-5}=-$\frac{9}{14}log_2\frac{9}{14}$-$\frac{5}{14}log_2\frac{5}{14}$=0.94

Step 2 Entropy of all attributes:

Entropy of Hot{+2,-2}=-$\frac{2}{4}log_2\frac{2}{4}$-$\frac{2}{4}log_2\frac{2}{4}$=1.0

Entropy of Mild{+4,-2}=-$\frac{4}{6}log_2\frac{4}{6}$-$\frac{2}{6}log_2\frac{2}{6}$=0.91

Entropy of Cold{+3,-1}=-$\frac{3}{4}log_2\frac{3}{4}$-$\frac{1}{3}log_2\frac{1}{3}$=0.81

Information Gain = Entropy(Whole Data)

$-\frac{1}{14}Ent(H)-\frac{6}{14}$Ent(M)-$\frac{4}{14}$Ent(C)

=0.029

Calculate IG of Humidity(ID3 Algorithm iterative dicomiser)

Step1 : Entropy of entire dataset

S{+9,-5}=-$\frac{9}{14} log_2 \frac{9}{14}$-$\frac{5}{14} log_2 \frac{5}{14}$=0.94

Step 2 Entropy of all attributes:

Entropy of High{+3,-4}=-$\frac{3}{7} log_2 \frac{3}{7}$-$\frac{4}{7} log_2 \frac{4}{7}$=0.98

Entropy of Normal{+6,-1}=-$\frac{6}{7} log_2 \frac{6}{7}$-$\frac{1}{7} log_2 \frac{1}{7}$=0.59

Information Gain = Entropy(Whole Data)

$-\frac{7}{14} Ent(H) - \frac{7}{14}$Ent(N)

=0.15

Calculate IG of Wind(ID3 Algorithm iterative dicomiser)

Step1 : Entropy of entire dataset

S{+9,-5}=-$\frac{9}{14} log_2 \frac{9}{14}$-$\frac{5}{14} log_2 \frac{5}{14}$=0.94

Step 2 Entropy of all attributes:

Entropy of Strong{+3,-3}=-$\frac{3}{6} log_2 \frac{3}{6}$-$\frac{3}{6} log_2 \frac{3}{6}$ = 1.0

Entropy of Normal{+6,-2}=-$\frac{6}{8} log_2 \frac{6}{8}$-$\frac{2}{8} log_2 \frac{2}{8}$=0.81

Information Gain = Entropy(Whole Data)

$-\frac{6}{14} Ent(S) - \frac{8}{14}$Ent(N)

=0.0478

Gain (S, Weather)=0.246

Gain(S, Temp)= 0.029

Gain(S, Humidity) = 0.15

Gain(S, Wind) = 0.0478

Gain(Ssunny, Temp)=0.57

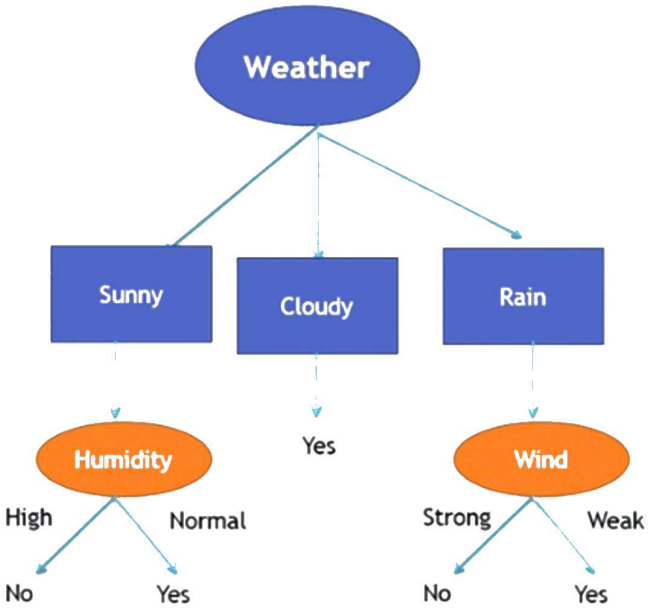
Gain(Ssunny, Humidity)=0.97

Gain(Ssunny, Wind)=0.0.19

Gain(Srain, Temp)=0.019

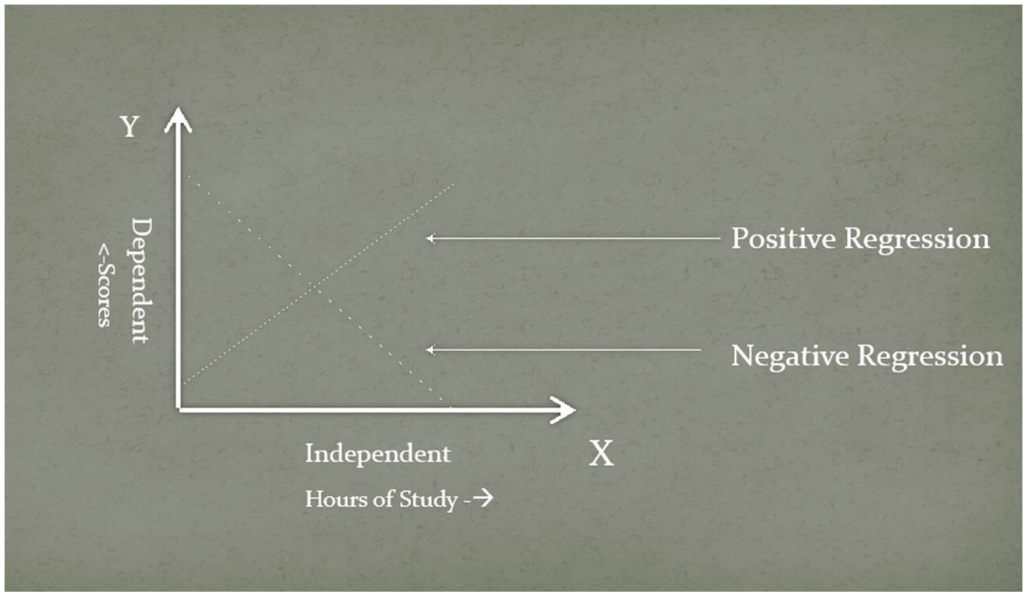
Gain(Ssunny, Humidity)=0.019

Gain(Ssunny, Wind)=0.97



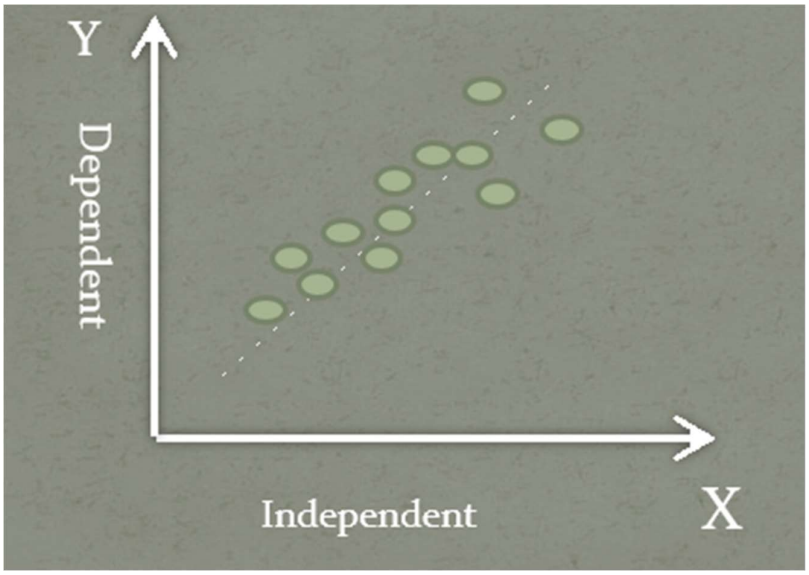
REGRESSION

- Regression is a statistical method that method that helps us understand and predict the relation between variable.
- Describe how one variable (Dependent Variable) changes as another variable (independent variable) changes.
- **Dependent Variable** :We are trying to predict or explain(Y).
- **Independent Variable**: That are use to predict or explain the changes in the dependent variable(X).
- For Example : predicting salary based on years of experience ,predicting exam score based study hours, predicting resale price based on vehicle age.



linear regression

linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable.



Equation:

$Y = \alpha 0 + \alpha 1 X1$

$Y = c + mX$

Multiple Linear equation

$Y = \alpha 0 + \alpha 1 X1 + \alpha 2 X2 \ldots\ldots \alpha m Xm$

$\alpha\ 1$=Reg coeff

Xi=Independent Var

Y=Dependendent var

Example:

	x	y	xy	x^2
	1	**3**	**3**	**1**
	2	**4**	**8**	**4**
	3	**5**	**15**	**9**
	4	**7**	**28**	**16**
Total	**10**	**19**	**54**	**30**

Y = bx+a

a=$((\sum y)(\sum x\text{^}2) - (\sum x)(\sum xy))/(n(\sum x\text{^}2) - (\sum x)\text{^}2)$

B=$\frac{n(xy)-(\sum x)(\sum y)}{n(\sum x^2)-(\sum x)^2}$

a=$\frac{(19)(30)-(10)(54)}{(4)(30)-(100)}$

=$\frac{570-540}{120-100}$

=$\frac{30}{20} = \frac{3}{2}$ = **1.5**

b=$\frac{(4)(54)-(10)(19)}{(4)(30)-(100)}$

=$\frac{216-190}{120-100}$

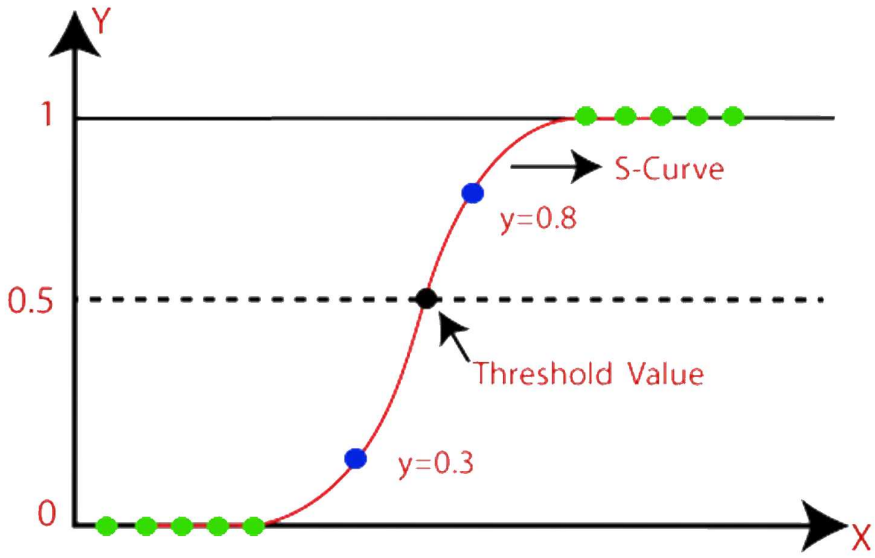
=$\frac{26}{20}$ =$\frac{13}{10}$ =**1.3**

Y=1.3X+1.5

LOGISTIC REGRESSION

- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
- Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.

- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
- The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
- Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.
- Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:



Logistic Function (Sigmoid Function):

- The sigmoid function is a mathematical function used to map the predicted values to probabilities.
- It maps any real value into another value within a range of 0 and 1.
- The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.
- In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

Logistic Regression Equation:

The Logistic regression equation can be obtained from the Linear Regression equation. The mathematical steps to get Logistic Regression equations are given below:

- We know the equation of the straight line can be written as:

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

- In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y):

$$\frac{y}{1-y}$$; 0 for y= 0, and infinity for y=1

- But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will become:

$$log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

The above equation is the final equation for Logistic Regression.

Polynomial Regression

- Polynomial Regression is a regression algorithm that models the relationship between a dependent(y) and independent variable(x) as nth degree polynomial. The Polynomial Regression equation is given below:

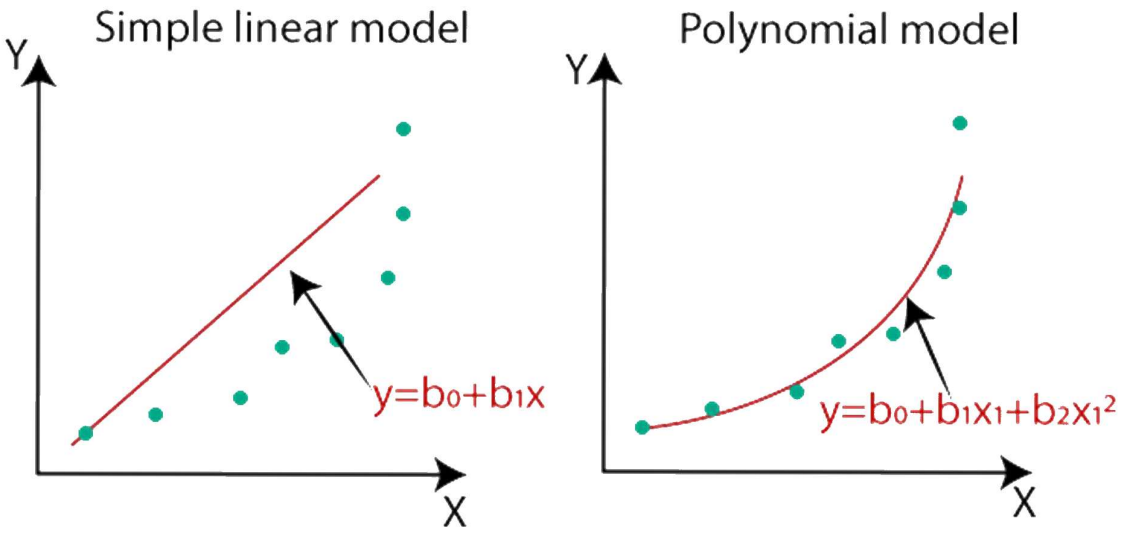
y= b0+b1x1+ b2x12+ b2x13+..... bnx1n

- It is also called the special case of Multiple Linear Regression in ML. Because we add some polynomial terms to the Multiple Linear regression equation to convert it into Polynomial Regression.
- It is a linear model with some modification in order to increase the accuracy.
- The dataset used in Polynomial regression for training is of non-linear nature.
- It makes use of a linear regression model to fit the complicated and non-linear functions and datasets.
- **Hence,** *"In Polynomial regression, the original features are converted into Polynomial features of required degree (2,3,..,n) and then modeled using a linear model."*

Need for Polynomial Regression:

The need of Polynomial Regression in ML can be understood in the below points:

- If we apply a linear model on a **linear dataset**, then it provides us a good result as we have seen in Simple Linear Regression, but if we apply the same model without any modification on a **non-linear dataset**, then it will produce a drastic output. Due to which loss function will increase, the error rate will be high, and accuracy will be decreased.
- So for such cases, **where data points are arranged in a non-linear fashion, we need the Polynomial Regression model**. We can understand it in a better way using the below comparison diagram of the linear dataset and non-linear dataset.



- In the above image, we have taken a dataset which is arranged non-linearly. So if we try to cover it with a linear model, then we can clearly see that it hardly covers any data point. On the other hand, a curve is suitable to cover most of the data points, which is of the Polynomial model.
- Hence, *if the datasets are arranged in a non-linear fashion, then we should use the Polynomial Regression model instead of Simple Linear Regression.*

Equation of the Polynomial Regression Model:

Simple Linear Regression equation: $y = b0+b1x$ .....(a)

Multiple Linear Regression equation: $y= b0+b1x+ b2x2+ b3x3+....+ bnxn$ .....(b)

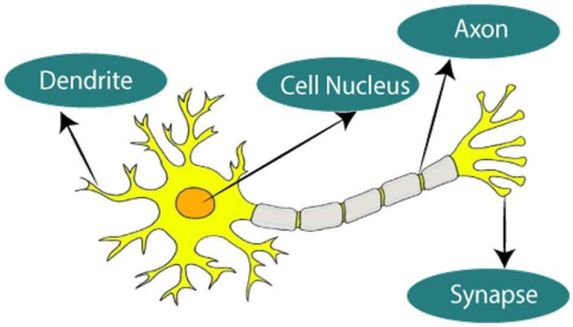
Polynomial Regression equation: $y= b0+b1x + b2x2+ b3x3+....+ bnxn$ .....(c)

Neural networks classifiers

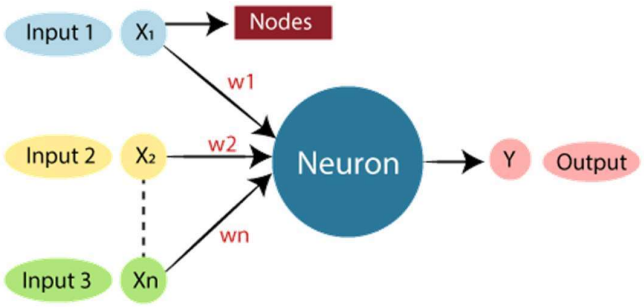
"Artificial Neural Network" is derived from Biological neural networks that develop the structure of a human brain. Similar to the human brain that has neurons interconnected to one another, artificial neural networks also have neurons that are interconnected to one another in various layers of the networks. These neurons are known as nodes.

Biological Neural Network.

Biological Neural Network.



Artificial Neural Network



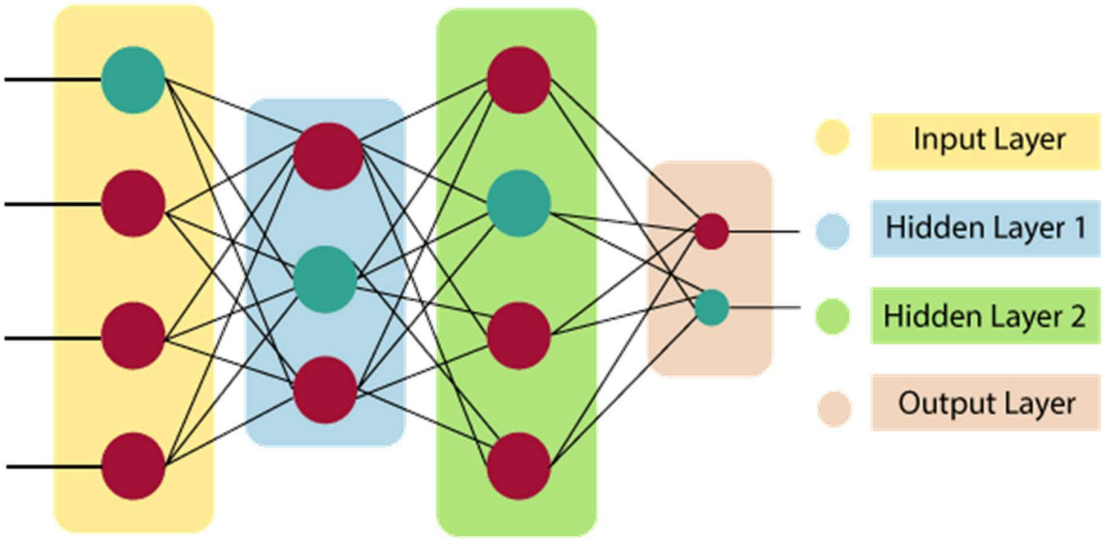
Relationship between Biological neural network and artificial neural network:

Biological Neural Network	Artificial Neural Network
Dendrites	Inputs
Cell nucleus	Nodes
Synapse	Weights
Axon	Output

An Artificial Neural Network in the field of Artificial intelligence where it attempts to mimic the network of neurons makes up a human brain so that computers will have an option to

understand things and make decisions in a human-like manner. The artificial neural network is designed by programming computers to behave simply like interconnected brain cells.

Architecture of an artificial neural network



Hidden Layer:

The hidden layer presents in-between input and output layers. It performs all the calculations to find hidden features and patterns.

Output Layer:

The input goes through a series of transformations using the hidden layer, which finally results in output that is conveyed using this layer.

The artificial neural network takes input and computes the weighted sum of the inputs and includes a bias. This computation is represented in the form of a transfer function.

$$\sum_{i=1}^{n} Wi * Xi + b$$

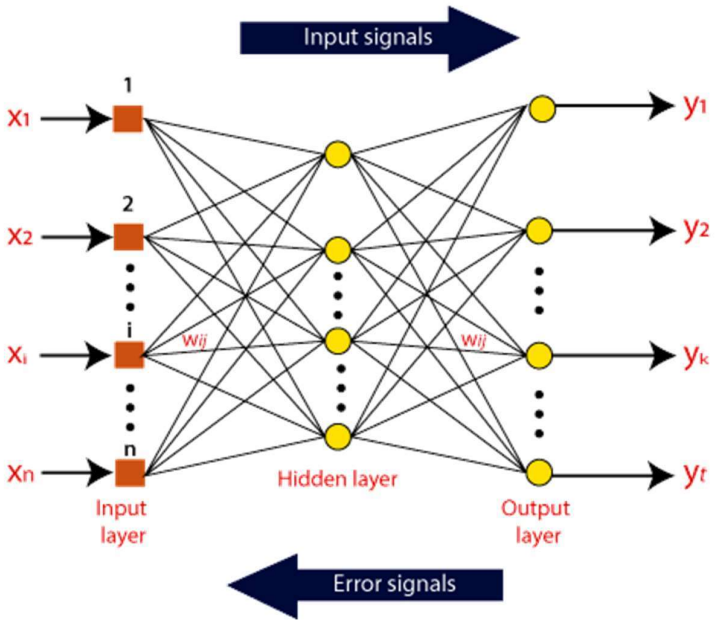
Input Layer:

As the name suggests, it accepts inputs in several different formats provided by the programmer.

- Advantages of Artificial Neural Network (ANN)
- Parallel processing capability
- Storing data on the entire network
- Capability to work with incomplete knowledge
- Having a memory distribution
- Having fault tolerance

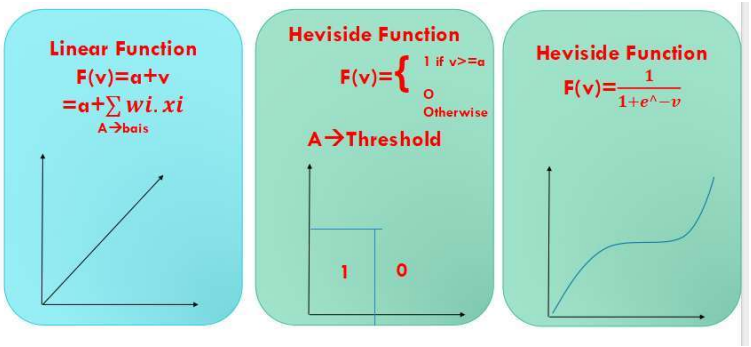
How do artificial neural networks work?

Artificial Neural Network can be best represented as a weighted directed graph, where the artificial neurons form the nodes. The association between the neurons outputs and neuron inputs can be viewed as the directed edges with weights. The Artificial Neural Network receives the input signal from the external source in the form of a pattern and image in the form of a vector. These inputs are then mathematically assigned by the notations x(n) for every n number of inputs.

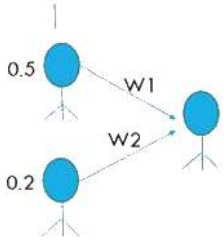


Afterward, each of the input is multiplied by its corresponding weights (these weights are the details utilized by the artificial neural networks to solve a specific problem). In general terms, these weights normally represent the strength of the interconnection between neurons inside the artificial neural network. All the weighted inputs are summarized inside the computing unit.

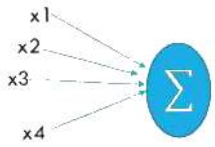
Activation Function

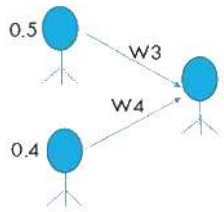


Example



0.5xw1+02xw2+b
=0.5+0.3+0.7+0.1
=0.15+0.35+0.1
=0.6





Evaluating Model performance:

Model performance in general refers to how well a model accomplishes its intended task, but it is important to define exactly what element of a model is being considered, and what "doing well" means for that element.

For instance, in a model designed to look for credit card fraud, identifying as many fraudulent transactions as possible will likely be the goal. The number of false positives (where non-fraudulent activity was misidentified as fraud) will be less important than the number of false negatives (where fraudulent activity is not identified). In this case, the recall of the model is likely to be the most important performance indicator.

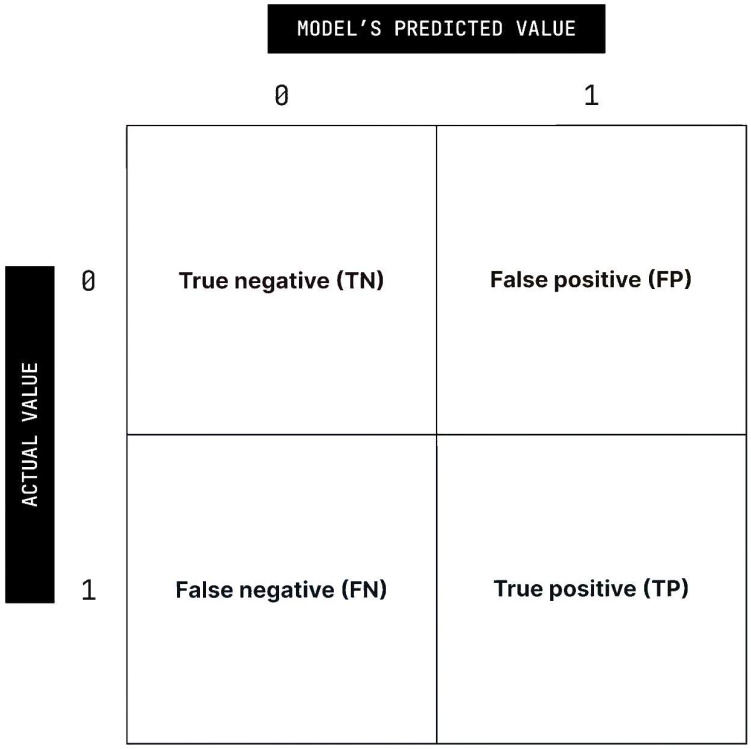
Performance evaluation is the quantitative measure of how well a trained model performs on specific model evaluation metrics in machine learning. This information can then be used to determine if a model is ready to move onto the next stage of testing, be deployed more broadly, or is in need of more training or retraining.

What are the model evaluation methods?

Two of the most important categories of evaluation methods are classification and regression model performance metrics. Understanding how these metrics are calculated will enable you to choose what is most important within a given model, and provide quantitative measures of performance within that metric.

Classification metrics

Classification metrics are generally used for discrete values a model might produce when it has finished classifying all the given data. In order to clearly display the raw data needed to calculate desired classification metrics, a confusion matrix for a model can be created.



This matrix makes clear not only how often the model predictions were correct, but also in which ways it was correct or incorrect. These variables are listed in formulas as TN (true negative), FP (false positive), etc.

These are some of the most commonly useful classification metrics that can be calculated from the data contained in a confusion matrix.

- **Accuracy -** percentage of the total variables that were correctly classified. Use the formula Accuracy = (TP+TN) / (TP+TN+FP+FN)
- **False positive rate -** how often the model predicts a positive for a value that is actually negative. Use the formula False Positive Rate = FP / (FP+TN)
- **Precision -** percentage of positive cases that were true positives as opposed to false positives. Use the formula Precision = TP / (TP+FP)
- **Recall -** percentage of actual positive cases that were predicted as positives, as opposed to those classified as false negatives. Use the formula Recall = TP/(TP+FN)

- **Logarithmic loss -** measure of how many total errors a model has. The closer to zero, the more correct predictions a model makes in classifications.
- **Area under curve -** method of visualizing true and false positive rates against each other.

Regression Metrics

Regression metrics are techniques generally better suited to be applied to continuous output of a machine learning model, as opposed to classification metrics which tend to work better to analyze discrete final results.

Some of the most useful regression metrics include:

- **Coefficient of determination (or R-squared) -** measures variance of a model compared to the actual data.
- **Mean squared error -** measures the amount of average divergence of the model from the observed data.
- **Mean absolute error -** measures the vertical and horizontal distance between data points and a linear regression line to illustrate how much a model deviates from observed data.
- **Mean absolute percentage error -** shows mean absolute error as a percentage.
- **Weighted mean absolute percentage error -** uses actual values (rather than absolute values) to measure percentage errors.

Unit IV Association Rule mining

Introduction to frequent pattern mining:

Frequent Pattern Mining (AKA Association Rule Mining) is an analytical process that finds frequent patterns, associations, or causal structures from data sets found in various kinds of databases such as relational databases, transactional databases, and other data repositories.

Given a set of transactions, this process aims to find the rules that enable us to predict the occurrence of a specific item based on the occurrence of other items in the transaction.

Let's look at an example of Frequent Pattern Mining. First, we will want to understand the terminology used in this type of analysis. While there are numerous metrics and factors used in this technique, for this example, we will only consider two factors namely, Support and Confidence.

Support: The support of a rule x -> y (where x and y are each items/events etc.) is defined as the proportion of transactions in the data set which contain the item set x as well as y. So, Support (x -> y)= no. of transactions which contain the item set x & y / total no. of transactions.

Confidence: The confidence of a rule x -> y is defined as: Support (x -> y) / support (x). So, it is the ratio of the number of transactions that include all items in the consequent (y in this case), as well as the antecedent (x in this case) to the number of transactions that include all items in the antecedent (x in this case).

In the table below, Support (milk->bread) = 0.4 means milk and bread are purchased together occur in 40% of all transactions. Confidence (milk->bread) = 0.5 means that if there are 100 transactions containing milk then there will be 50 that will also contain bread.

TID	Milk	Bread	Butter	Beer
1	1	0	1	1
2	1	1	1	0
3	0	1	1	0
4	1	0	0	1
5	1	1	1	1

How Does Frequent Pattern Mining Support Business Analysis?

This method of analysis can be useful in evaluating data for various business functions and industries.

- **Basket Data Analysis**: To analyze the association of purchased items in a single basket or single purchase.
- **Cross Marketing and Selling**: To work with other businesses that complement your own, not competitors. For example, vehicle dealerships and manufacturers have cross marketing campaigns with oil and gas companies for obvious reasons.
- **Catalog Design**: The selection of items in a business' catalog are often designed to complement each other, so that buying one item will lead to buying another, so these items are often complements or closely related.
- **Medical Treatments**: Each patient is represented as a transaction containing the ordered set of diseases, and which diseases are likely to occur simultaneously/sequentially can be predicted.

To understand the value of this applied technique, let's consider two business use cases.

Use Case One

- **Business Problem:** A retail store manager wants to conduct Market Basket analysis to come up with a better strategy of product placement and product bundling.
- **Business Benefit:** Based on the rules generated, the store manager can strategically place the products together or in sequence leading to growth in sales and, in turn, revenue of the store. Offers such as "Buy this and get this free" or "Buy this and get % off on this" can be designed based on the rules generated.

Use Case Two

- **Business Problem:** A bank-marketing manager wishes to analyze which products are frequently and sequentially bought together. Each customer is represented as a transaction, containing the ordered set of products, and which products are likely to be purchased simultaneously/sequentially can then be predicted.
- **Business Benefit:** Based on the rules generated, banking products can be cross-sold to each existing or prospective customer to drive sales and bank revenue. For example, if savings, personal loan and credit cards are frequently/sequentially bought, then a new saving account customer can be cross-sold with a personal loan and credit card.
- Frequent Pattern Mining (AKA Association Rule Mining) is an analytical process that finds frequent patterns, associations, or causal structures from data sets found in various kinds of data repositories. This method of analysis can be useful in evaluating data for various business functions and industries and is useful in determining the frequent patterns in buying behavior for various products and services, and in analyzing the relationships among various data points to cross-sell and bundle products, and service offerings, and to understand target audiences.

Understanding association rule:

Association rules are "if-then" statements, that help to show the probability of relationships between data items, within large data sets in various types of databases. Association rule mining has a number of applications and is widely used to help discover sales correlations in transactional data.

Use cases for association rules

In data science, association rules are used to find correlations and co-occurrences between data sets. They are ideally used to explain patterns in data from seemingly independent information repositories, such as relational databases and transactional databases. The act of using association rules is sometimes referred to as "association rule mining" or "mining associations."

Below are a few real-world use cases for association rules:

- **Medicine.** Doctors can use association rules to help diagnose patients. There are many variables to consider when making a diagnosis, as many diseases share symptoms. By using association rules and machine learning-fueled data analysis, doctors can determine the conditional probability of a given illness by comparing symptom relationships in the data from past cases. As new diagnoses get made, the machine learning model can adapt the rules to reflect the updated data.
- **Retail.** Retailers can collect data about purchasing patterns, recording purchase data as item barcodes are scanned by point-of-sale systems. Machine learning models can look for co-occurrence in this data to determine which products are most likely to be purchased together. The retailer can then adjust marketing and sales strategy to take advantage of this information.
- **User experience (UX) design.** Developers can collect data on how consumers use a website they create. They can then use associations in the data to optimize the website user interface -- by analyzing where users tend to click and what maximizes the chance that they engage with a call to action, for example.
- **Entertainment.** Services like Netflix and Spotify can use association rules to fuel their content recommendation engines. Machine learning models analyze past user behavior data

for frequent patterns, develop association rules and use those rules to recommend content that a user is likely to engage with, or organize content in a way that is likely to put the most interesting content for a given user first.

How association rules work

Association rule mining, at a basic level, involves the use of machine learning models to analyze data for patterns, or co-occurrences, in a database. It identifies frequent if-then associations, which themselves are the *association rules*.

Association rules are created by searching data for frequent if-then patterns and using the criteria *support* and *confidence* to identify the most important relationships. Support is an indication of how frequently the items appear in the data. Confidence indicates the number of times the if-then statements are found true. A third metric, called *lift*, can be used to compare confidence with expected confidence, or how many times an if-then statement is expected to be found true. Association rules are calculated from *itemsets*, which are made up of two or more items. If rules are built from analyzing all the possible itemsets, there could be so many rules that the rules hold little meaning. With that, association rules are typically created from rules well represented in data.

Characteristics of association rules and item sets

An association rule consists of a set of items, the *rule body*, leading to another item, the *rule head*. The association rule relates the rule body with the rule head.

An association rule can contain the following characteristics:

- Statistical information about the frequency of occurrence
- Reliability
- Importance of this relation

In the following example, swimsuits and beach towels represent the rule body. Sun glasses represent the rule head. Support and Confidence are measures of the reliability and the frequency of occurrence.

Example

```
[Swimsuits] + [Beach towels] ==> [Sun glasses] Support=24% Confidence=60%
Lift=2.0
```

Apriori Algorithm :

Apriori algorithm is given by R. Agrawal and R. Srikant in 1994 for finding frequent itemsets in a dataset for boolean association rule. Name of the algorithm is Apriori because it uses prior knowledge of frequent itemset properties.

Apriori Property

All non-empty subset of frequent itemset must be frequent. The key concept of Apriori algorithm is its anti-monotonicity of support measure. Apriori assumes thatBefore we start understanding the algorithm, go through some definitions.

Consider the following dataset and we will find frequent itemsets and generate association rules for them.

TID	items
T1	I1, I2 , I5
T2	I2,I4
T3	I2,I3
T4	I1,I2,I4
T5	I1,I3
T6	I2,I3
T7	I1,I3
T8	I1,I2,I3,I5
T9	I1,I2,I3

minimum support count is 2
minimum confidence is 60%

Step-1: K=1
(I) Create a table containing support count of each item present in dataset – Called **C1(candidate set)**

Itemset	sup_count
I1	6
I2	7
I3	6
I4	2
I5	2

(II) compare candidate set item's support count with minimum support count(here min_support=2 if support_count of candidate set items is less than min_support then remove those items). This gives us itemset L1.

Itemset	sup_count
I1	6
I2	7
I3	6
I4	2
I5	2

Step-2: K=2

- Generate candidate set C2 using L1 (this is called join step). Condition of joining L_{k-1} and L_{k-1} is that it should have (K-2) elements in common.
- Check all subsets of an itemset are frequent or not and if not frequent remove that itemset.(Example subset of{I1, I2} are {I1}, {I2} they are frequent.Check for each itemset)
- Now find support count of these itemsets by searching in dataset.

Itemset	sup_count
I1,I2	4
I1,I3	4
I1,I4	1
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2
I3,I4	0
I3,I5	1
I4,I5	0

(II) compare candidate (C2) support count with minimum support count(here min_support=2 if support_count of candidate set item is less than min_support then remove those items) this gives us itemset L2.

Itemset	sup_count
I1,I2	4
I1,I3	4
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2
I2,I5	2

Step-3:

- Generate candidate set C3 using L2 (join step). Condition of joining L_{k-1} and L_{k-1} is that it should have (K-2) elements in common. So here, for L2, first element should match.

So itemset generated by joining L2 is {I1, I2, I3}{I1, I2, I5}{I1, I3, i5}{I2, I3, I4}{I2, I4, I5}{I2, I3, I5}

- Check if all subsets of these itemsets are frequent or not and if not, then remove that itemset.(Here subset of {I1, I2, I3} are {I1, I2},{I2, I3},{I1, I3} which are frequent. For {I2, I3, I4}, subset {I3, I4} is not frequent so remove it. Similarly check for every itemset)
- find support count of these remaining itemset by searching in dataset.

Itemset	sup_count
I1,I2,I3	2
I1,I2,I5	2

(II) Compare candidate (C3) support count with minimum support count(here min_support=2 if support_count of candidate set item is less than min_support then remove those items) this gives us itemset L3.

Itemset	sup_count
I1,I2,I3	2
I1,I2,I5	2

Step-4:

- Generate candidate set C4 using L3 (join step). Condition of joining L_{k-1} and L_{k-1} (K=4) is that, they should have (K-2) elements in common. So here, for L3, first 2 elements (items) should match.
- Check all subsets of these itemsets are frequent or not (Here itemset formed by joining L3 is {I1, I2, I3, I5} so its subset contains {I1, I3, I5}, which is not frequent). So no itemset in C4
- We stop here because no frequent itemsets are found further

Thus, we have discovered all the frequent item-sets. Now generation of strong association rule comes into picture. For that we need to calculate confidence of each rule.

Confidence –

A confidence of 60% means that 60% of the customers, who purchased milk and bread also bought butter.

Confidence(A->B)=Support_count(A∪B)/Support_count(A)

So here, by taking an example of any frequent itemset, we will show the rule generation.

Itemset {I1, I2, I3} //from L3

SO rules can be

[I1^I2]=>[I3] //confidence = sup(I1^I2^I3)/sup(I1^I2) = 2/4*100=50%

[I1^I3]=>[I2] //confidence = sup(I1^I2^I3)/sup(I1^I3) = 2/4*100=50%

[I2^I3]=>[I1] //confidence = sup(I1^I2^I3)/sup(I2^I3) = 2/4*100=50%

[I1]=>[I2^I3] //confidence = sup(I1^I2^I3)/sup(I1) = 2/6*100=33%

[I2]=>[I1^I3] //confidence = sup(I1^I2^I3)/sup(I2) = 2/7*100=28%

[I3]=>[I1^I2] //confidence = sup(I1^I2^I3)/sup(I3) = 2/6*100=33%

So if minimum confidence is 50%, then first 3 rules can be considered as strong association rules.

Limitations of Apriori Algorithm

Apriori Algorithm can be slow. The main limitation is time required to hold a vast number of candidate sets with much frequent itemsets, low minimum support or large itemsets i.e. it is not an efficient approach for large number of datasets. For example, if there are 10^4 from frequent 1- itemsets, it need to generate more than 10^7 candidates into 2-length which in turn they will be tested and accumulate. Furthermore, to detect frequent pattern in size 100 i.e. v1, v2… v100, it have to generate 2^100 candidate itemsets that yield on costly and wasting of time of candidate generation. So, it will check for many sets from candidate itemsets, also it will scan database many times repeatedly for finding candidate itemsets. Apriori will be very low and inefficiency when memory capacity is limited with large number of transactions

Frequent Pattern Growth Algorithm:

The two primary drawbacks of the Apriori Algorithm are:

1. At each step, candidate sets have to be built.
2. To build the candidate sets, the algorithm has to repeatedly scan the database.

These two properties inevitably make the algorithm slower. To overcome these redundant steps, a new association-rule mining algorithm was developed named Frequent Pattern Growth Algorithm. It overcomes the disadvantages of the Apriori algorithm by storing all the transactions in a Trie Data Structure.

Consider the following data:-

Transaction ID	Items
T1	{E, K, M, N, O, Y}
T2	{D, E, K, N, **O**, Y}
T3	{A, E, K, M}
T4	{C, K, M, U, Y}
T5	{C, E, I, K, O, O}

The above-given data is a hypothetical dataset of transactions with each letter representing an item. The frequency of each individual item is computed:-

Item	Frequency
A	1
C	2
D	1
E	4
I	1
K	5
M	3
N	2
0	3
U	1
Y	3

Let the minimum support be 3. A **Frequent Pattern set** is built which will contain all the elements whose frequency is greater than or equal to the minimum support. These elements are stored in descending order of their respective frequencies. After insertion of the relevant items, the set L looks like this:-

L = {K : 5, E : 4, M : 3, O : 3, Y : 3}

Now, for each transaction, the respective **Ordered-Item set** is built. It is done by iterating the Frequent Pattern set and checking if the current item is contained in the transaction in question. If the current item is contained, the item is inserted in the Ordered-Item set for the current transaction. The following table is built for all the transactions:

Transaction ID	Items	Ordered-Item Set
T1	{E, K, M, N, O, Y}	{K, E, M, O, Y}
T2	{D, E, K, N, O, Y}	{K, E, O, Y}
T3	{ A, E, K, M}	{K, E, M}
T4	{C, K, M, U, Y}	{K, M, Y}
T5	{C, E, I, K, O, O}	{K, E, O}

Now, all the Ordered-Item sets are inserted into a Trie Data Structure.

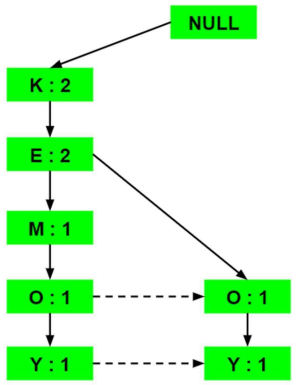
a) **Inserting the set {K, E, M, O, Y}:**

b) Here, all the items are simply linked one after the other in the order of occurrence in the set and initialize the support count for each item as 1.

b) **Inserting the set {K, E, O, Y}:**

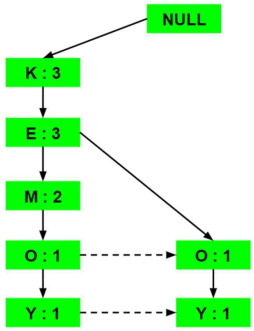
c) Till the insertion of the elements K and E, simply the support count is increased by 1. On inserting O we can see that there is no direct link between E and O, therefore a new node for the item O is initialized with the support count as 1 and item E is linked to this new node. On inserting Y, we first initialize a new node for the item Y with support count as 1 and link the new node of O with the new node of Y.

d)



c) **Inserting the set {K, E, M}:**

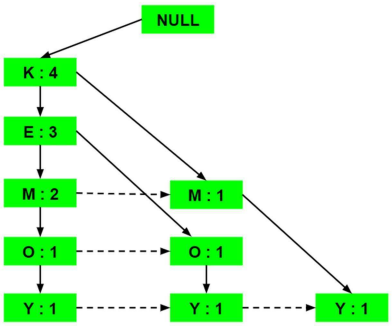
e) Here simply the support count of each element is increased by 1.



d) **Inserting the set {K, M, Y}:**

f) Similar to step b), first the support count of K is increased, then new nodes for M and Y are initialized and linked accordingly.

g)

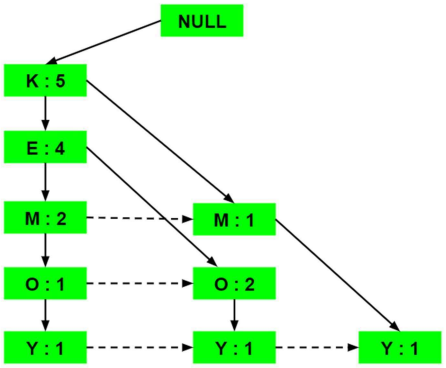


h)

i) e) **Inserting the set {K, E, O}:**

j) Here simply the support counts of the respective elements are increased. Note that the support count of the new node of item O is increased.

k)



l) Now, for each item, the **Conditional Pattern Base** is computed which is path labels of all the paths which lead to any node of the given item in the frequent-pattern tree. Note that the items in the below table are arranged in the ascending order of their frequencies.

Items	Conditional Pattern Base
Y	{{K,E,M,O : 1}, {K,E,O : 1}, {K,M : 1}}
O	{{K,E,M : 1}, {K,E : 2}}
M	{{K,E : 2}, {K : 1}}
E	{K : 4}
K	

Now for each item, the **Conditional Frequent Pattern Tree is built.** It is done by taking the set of elements that is common in all the paths in the Conditional Pattern Base of that item and calculating its support count by summing the support counts of all the paths in the Conditional Pattern Base.

Items	Conditional Pattern Base	Conditional Frequent Pattern Tree
Y	{{K,E,M,O : 1}, {K,E,O : 1}, {K,M : 1}}	{K : 3}
O	{{K,E,M : 1}, {K,E : 2}}	{K,E : 3}
M	{{K,E : 2}, {K : 1}}	{K : 3}
E	{K : 4}	{K : 4}
K		

From the Conditional Frequent Pattern tree, the **Frequent Pattern rules** are generated by pairing the items of the Conditional Frequent Pattern Tree set to the corresponding to the item as given in the below table.

Items	Frequent Pattern Generated
Y	{<K,Y : 3>}
O	{<K,O : 3>, <E,O : 3>, <E,K,O : 3>}
M	{<K,M : 3>}
E	{<E,K : 4>}
K	

For each row, two types of association rules can be inferred for example for the first row which contains the element, the rules K -> Y and Y -> K can be inferred. To determine the valid rule, the confidence of both the rules is calculated and the one with confidence greater than or equal to the minimum confidence value is retained.

Eclat algorithm:

The ECLAT algorithm stands for **Equivalence Class Clustering and bottom-up Lattice Traversal**. It is one of the popular methods of Association Rule mining. It is a more efficient and scalable version of the Apriori algorithm.

How the algorithm work? :

The basic idea is to use Transaction Id Sets(tidsets) intersections to compute the support value of a candidate and avoiding the generation of subsets which do not exist in the prefix tree. In the first call of the function, all single items are used along with their tidsets. Then the function is called recursively and in each recursive call, each item-tidset pair is verified and combined with other item-tidset pairs. This process is continued until no candidate item-tidset pairs can be combined.
Let us now understand the above stated working with an example:-

Consider the following transactions record:-

Transaction Id	Bread	Butter	Milk	Coke	Jam
T1	1	1	0	0	1
T2	0	1	0	1	0
T3	0	1	1	0	0
T4	1	1	0	1	0
T5	1	0	1	0	0
T6	0	1	1	0	0
T7	1	0	1	0	0
T8	1	1	1	0	1
T9	1	1	1	0	0

The above-given data is a boolean matrix where for each cell (i, j), the value denotes whether the j'th item is included in the i'th transaction or not. 1 means true while 0 means false.

We now call the function for the first time and arrange each item with it's tidset in a tabular fashion:-

k = 1, minimum support = 2

Item	**Tidset**
Bread	{T1, T4, T5, T7, T8, T9}
Butter	{T1, T2, T3, T4, T6, T8, T9}
Milk	{T3, T5, T6, T7, T8, T9}
Coke	{T2, T4}
Jam	{T1, T8}

We now recursively call the function till no more item-tidset pairs can be combined:-

k = 2

Item	Tidset
{Bread, Butter}	{T1, T4, T8, T9}
{Bread, Milk}	{T5, T7, T8, T9}
{Bread, Coke}	{T4}
{Bread, Jam}	{T1, T8}
{Butter, Milk}	{T3, T6, T8, T9}
{Butter, Coke}	{T2, T4}
{Butter, Jam}	{T1, T8}
{Milk, Jam}	{T8}

k = 3

Item	Tidset
{Bread, Butter, Milk}	{T8, T9}
{Bread, Butter, Jam}	{T1, T8}

k = 4

Item	Tidset
{Bread, Butter, Milk, Jam}	{T8}

We stop at k = 4 because there are no more item-tidset pairs to combine.

Since minimum support = 2, we conclude the following rules from the given dataset:-

Items Bought	Recommended Products
Bread	Butter
Bread	Milk
Bread	Jam
Butter	Milk
Butter	Coke
Butter	Jam
Bread and Butter	Milk
Bread and Butter	Jam

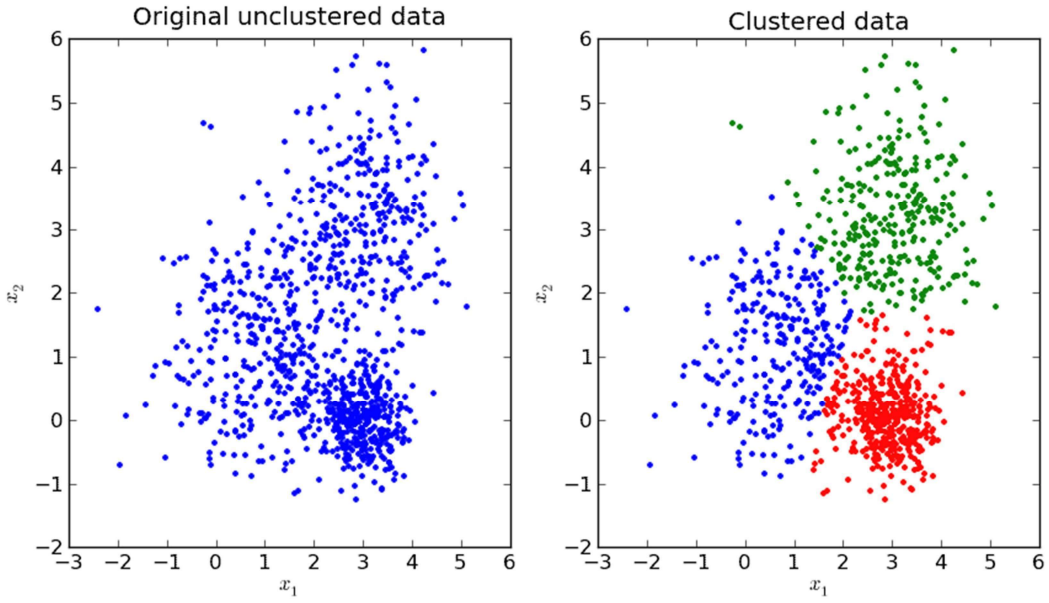
Advantages over Apriori algorithm:-

1. **Memory Requirements:** Since the ECLAT algorithm uses a Depth-First Search approach, it uses less memory than Apriori algorithm.
2. **Speed:** The ECLAT algorithm is typically faster than the Apriori algorithm.
3. **Number of Computations:** The ECLAT algorithm does not involve the repeated scanning of the data to compute the individual support values.

Unit V- Clustering

WHAT IS CLUSTERING AND HOW IT WORKS?

Clustering is the task of dividing the population or data points into several groups such that data points in the same groups are similar to other data points in that group and dissimilar to the data points in other groups. It is basically an assembly of objects based on similarity and dissimilarity between them.



THE USAGE OF CLUSTERING ALGORITHMS IN REAL WORLD

1. It is widely used in many applications such as image processing, data analysis, and pattern recognition.
2. It can be used in the field of biology, by deriving animal and plant taxonomies, identifying genes with the same capabilities.
3. It also helps in information discovery by classifying documents on the web.
4. It helps marketers to find the distinct groups in their customer base and they can characterize their customer groups by using purchasing patterns.

DIFFERENT TYPES OF CLUSTERING METHODS

Partitional Clustering :

Partitional clustering decomposes a data set into a set of disjoint clusters. Given a data set of N points, a partitioning method constructs K ($N \geq K$) partitions of the data, with each partition representing a cluster.

That is, it classifies the data into K groups by satisfying the following requirements: (1) each group contains at least one point, and (2) each point belongs to exactly one group. Notice that for fuzzy partitioning, a point can belong to more than one group.

Many partitional clustering algorithms try to minimize an objective function. For example, in K-means and K-medoids the function (also referred to as the distortion function) is

$$\sum i=1K\sum j=1|Ci|Dist(xj,center(i)),\sum i=1K\sum j=1|Ci|Dist(xj,center(i)), \quad (1)$$

where $|C_i|$ is the number of points in cluster i, Dist(x_j, center(i)) is the distance between point x_j and center i. Many distance functions can be used, such as Euclidean distance

Partitioning Method (K-Mean):

Partitioning Method: This clustering method classifies the information into multiple groups based on the characteristics and similarity of the data. Its the data analysts to specify the number of clusters that has to be generated for the clustering methods. In the partitioning method when database(D) that contains multiple(N) objects then the partitioning method constructs user-specified(K) partitions of the data in which each partition represents a cluster and a particular region. There are many algorithms that come under partitioning method some of the popular ones are K-Mean, PAM(K-Medoids), CLARA algorithm (Clustering Large Applications) etc. In this article, we will be seeing the working of K Mean algorithm in detail. **K-Mean (A centroid based Technique):**

The K means algorithm takes the input parameter K from the user and partitions the dataset containing N objects into K clusters so that resulting similarity among the data objects inside the group (intracluster) is high but the similarity of data objects with the data objects from outside the cluster is low (intercluster). The similarity of the cluster is determined with respect to the mean value of the cluster. It is a type of square error algorithm. At the start randomly k objects from the dataset are chosen in which each of the objects represents a cluster mean(centre). For the rest of the data objects, they are assigned to the nearest cluster based on their distance from the cluster mean. The new mean of each of the cluster is then calculated with the added data objects.

Algorithm: K mean:

Input:

K: The number of clusters in which the dataset has to be divided

D: A dataset containing N number of objects

Output:
A dataset of K clusters
Method:
1. Randomly assign K objects from the dataset(D) as cluster centres(C)
2. (Re) Assign each object to which object is most similar based upon mean values.
3. Update Cluster means, i.e., Recalculate the mean of each cluster with the updated values.
4. Repeat Step 2 until no change occurs.

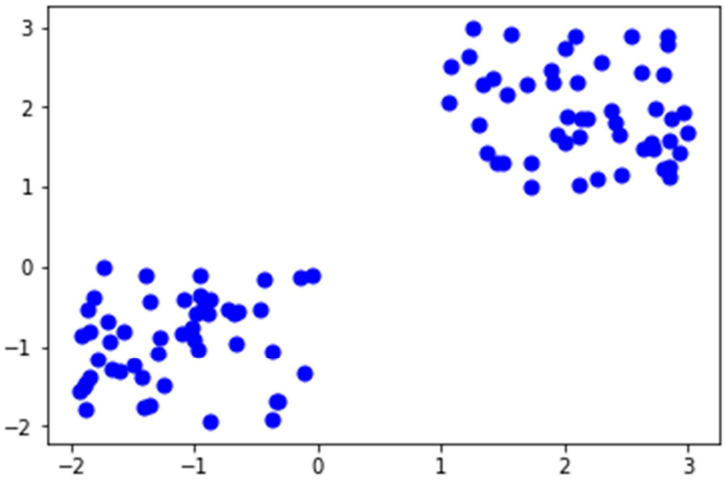


Figure – K-mean

Clustering**Flowchart:**

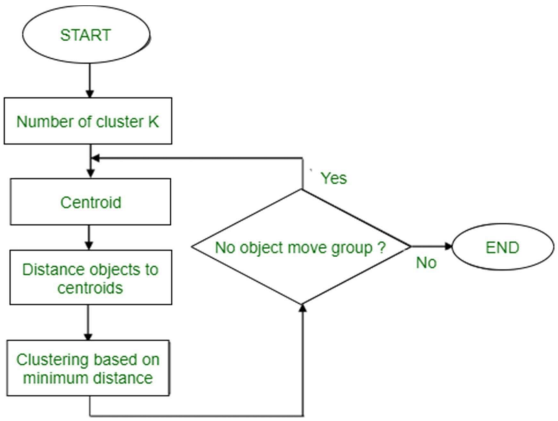


Figure – K-mean Clustering

Example:
Suppose we want to group the visitors to a website using just their age as follows:
16, 16, 17, 20, 20, 21, 21, 22, 23, 29, 36, 41, 42, 43, 44, 45, 61, 62, 66

Initial Cluster:
K=2

Centroid(C1) = 16 [16]

Centroid(C2) = 22 [22]

Note: These two points are chosen randomly from the dataset. **Iteration-1:**
C1 = 16.33 [16, 16, 17]

C2 = 37.25 [20, 20, 21, 21, 22, 23, 29, 36, 41, 42, 43, 44, 45, 61, 62, 66]

Iteration-2:
C1 = 19.55 [16, 16, 17, 20, 20, 21, 21, 22, 23]

C2 = 46.90 [29, 36, 41, 42, 43, 44, 45, 61, 62, 66]

Iteration-3:
C1 = 20.50 [16, 16, 17, 20, 20, 21, 21, 22, 23, 29]

C2 = 48.89 [36, 41, 42, 43, 44, 45, 61, 62, 66]

Iteration-4:
C1 = 20.50 [16, 16, 17, 20, 20, 21, 21, 22, 23, 29]

C2 = 48.89 [36, 41, 42, 43, 44, 45, 61, 62, 66]

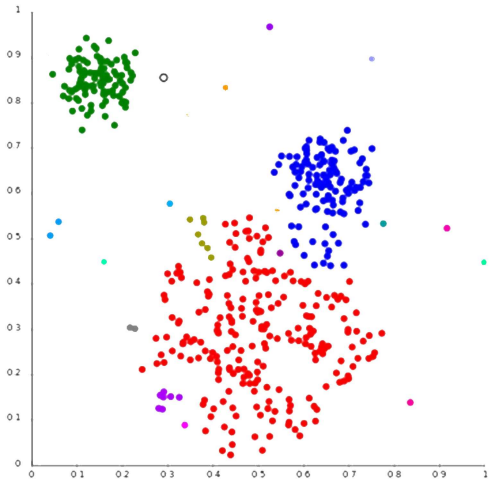
No change Between Iteration 3 and 4, so we stop. Therefore we get the clusters **(16-29)** and **(36-66)** as 2 clusters we get using K Mean Algorithm.

Connectivity-based Clustering (Hierarchical clustering):

Hierarchical Clustering is a method of unsupervised machine learning clustering where it begins with a pre-defined top to bottom hierarchy of clusters. It then proceeds to perform a decomposition of the data objects based on this hierarchy, hence obtaining the clusters

The core idea of the connectivity-based model is similar to Centroid based model which is basically defining clusters on the basis of the closeness of data points. Here we work on a notion that the data points which are closer have similar behavior as compared to data points that are farther. It is not a single partitioning of the data set, instead, it provides an extensive hierarchy of clusters that merge with each other at certain distances. Here the choice of distance function is subjective. These models are very easy to interpret but it lacks scalability.

For Ex- *hierarchical algorithm* and its variants



hierarchical algorithm:

A **Hierarchical clustering** method works via grouping data into a tree of clusters. Hierarchical clustering begins by treating every data point as a separate cluster. Then, it repeatedly executes the subsequent steps:

1. Identify the 2 clusters which can be closest together, and
2. Merge the 2 maximum comparable clusters. We need to continue these steps until all the clusters are merged together.

The basic method to generate hierarchical clustering is

1. **Agglomerative:** Initially consider every data point as an **individual** Cluster and at every step, **merge** the nearest pairs of the cluster. (It is a bottom-up method). At first, every dataset is considered as an individual entity or cluster. At every iteration, the clusters merge with different clusters until one cluster is formed.

The algorithm for Agglomerative Hierarchical Clustering is:

- Calculate the similarity of one cluster with all the other clusters (calculate proximity matrix)
- Consider every data point as an individual cluster
- Merge the clusters which are highly similar or close to each other.
- Recalculate the proximity matrix for each cluster
- Repeat Steps 3 and 4 until only a single cluster remains.

Let's say we have six data points **A, B, C, D, E, and F**.

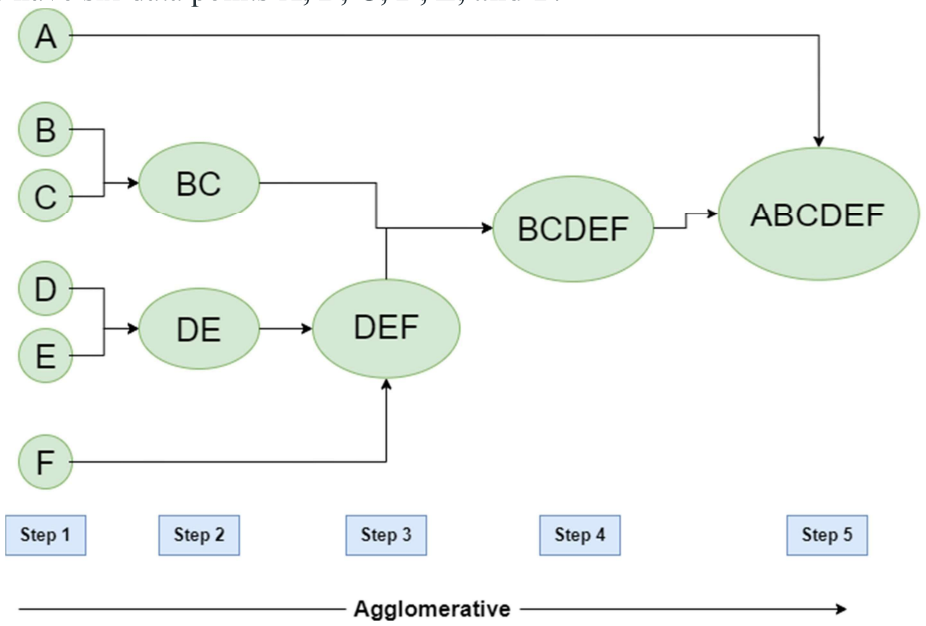


Figure – Agglomerative Hierarchical clustering

- **Step-1:** Consider each alphabet as a single cluster and calculate the distance of one cluster from all the other clusters.
- **Step-2:** In the second step comparable clusters are merged together to form a single cluster. Let's say cluster (B) and cluster (C) are very similar to each other therefore we merge them in the second step similarly to cluster (D) and (E) and at last, we get the clusters [(A), (BC), (DE), (F)]
- **Step-3:** We recalculate the proximity according to the algorithm and merge the two nearest clusters([(DE), (F)]) together to form new clusters as [(A), (BC), (DEF)]
- **Step-4:** Repeating the same process; The clusters DEF and BC are comparable and merged together to form a new cluster. We're now left with clusters [(A), (BCDEF)].
- **Step-5:** At last the two remaining clusters are merged together to form a single cluster [(ABCDEF)].

2. Divisive:

We can say that the Divisive Hierarchical clustering is precisely the **opposite** of the Agglomerative Hierarchical clustering. In Divisive Hierarchical clustering, we take into account all of the data points as a single cluster and in every iteration, we separate the data points from the clusters which aren't comparable. In the end, we are left with N clusters.

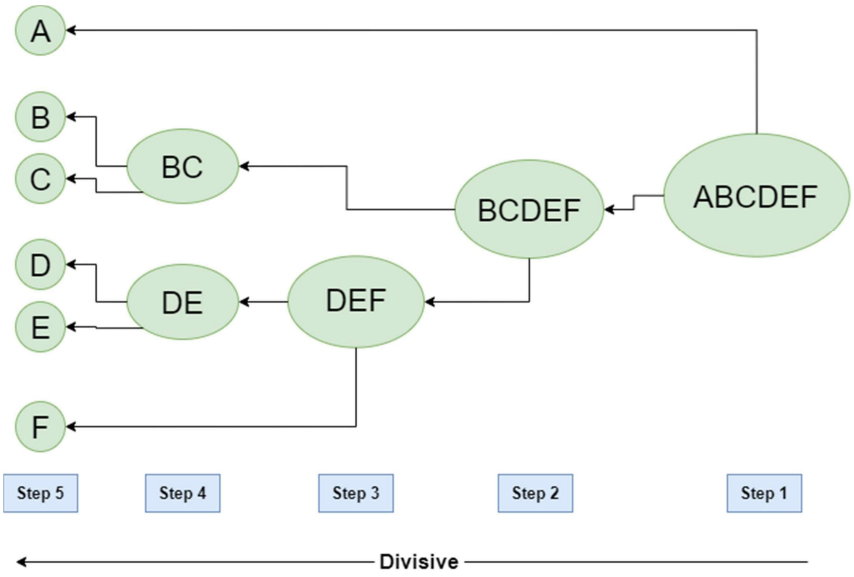
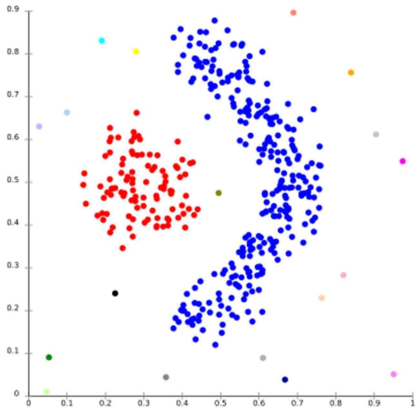


Figure – Divisive Hierarchical clustering

Density Models :

In this clustering model, there will be searching of data space for areas of the varied density of data points in the data space. It isolates various density regions based on different densities present in the data space.
For Ex- *DBSCAN and OPTICS*.

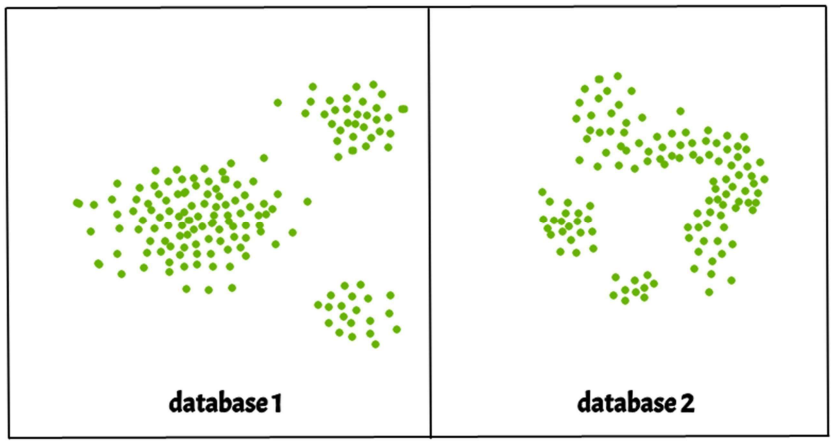


DBSCAN Clustering Density based clustering:

Clustering analysis or simply Clustering is basically an Unsupervised learning method that divides the data points into a number of specific batches or groups, such that the data points in the same groups have similar properties and data points in different groups have different properties in some sense. It comprises many different methods based on differential evolution.
E.g. K-Means (distance between points), Affinity propagation (graph distance), Mean-shift

(distance between points), DBSCAN (distance between nearest points), Gaussian mixtures (Mahalanobis distance to centers), Spectral clustering (graph distance) etc.

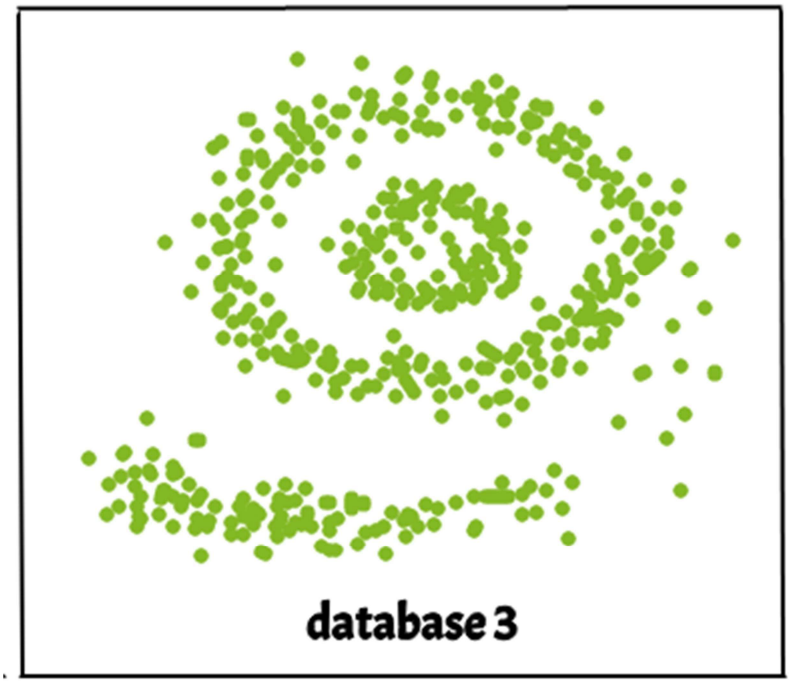
Fundamentally, all clustering methods use the same approach i.e. first we calculate similarities and then we use it to cluster the data points into groups or batches. Here we will focus on **Density-based spatial clustering of applications with noise** (DBSCAN) clustering method.
Clusters are dense regions in the data space, separated by regions of the lower density of points. The ***DBSCAN algorithm*** is based on this intuitive notion of "clusters" and "noise". The key idea is that for each point of a cluster, the neighborhood of a given radius has to contain at least a minimum number of points.



Why DBSCAN?

Partitioning methods (K-means, PAM clustering) and hierarchical clustering work for finding spherical-shaped clusters or convex clusters. In other words, they are suitable only for compact and well-separated clusters. Moreover, they are also severely affected by the presence of noise and outliers in the data.
Real life data may contain irregularities, like:

1. Clusters can be of arbitrary shape such as those shown in the figure below.
2. Data may contain noise.

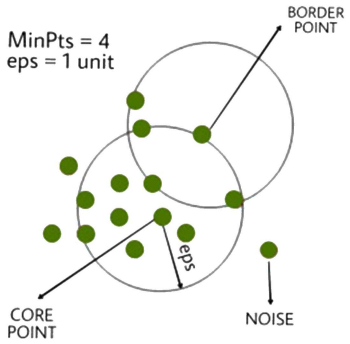


The figure below shows a data set containing nonconvex clusters and outliers/noises. Given such data, k-means algorithm has difficulties in identifying these clusters with arbitrary shapes.
DBSCAN algorithm requires two parameters:

1. **eps** : It defines the neighborhood around a data point i.e. if the distance between two points is lower or equal to 'eps' then they are considered neighbors. If the eps value is chosen too small then large part of the data will be considered as outliers. If it is chosen very large then the clusters will merge and the majority of the data points will be in the same clusters. One way to find the eps value is based on the ***k-distance graph***.
2. **MinPts**: Minimum number of neighbors (data points) within eps radius. Larger the dataset, the larger value of MinPts must be chosen. As a general rule, the minimum MinPts can be derived from the number of dimensions D in the dataset as, MinPts >= D+1. The minimum value of MinPts must be chosen at least 3.

In this algorithm, we have 3 types of data points.

Core Point*: A point is a core point if it has more than MinPts points within eps.*
Border Point*: A point which has fewer than MinPts within eps but it is in the neighborhood of a core point.*
Noise or outlier*: A point which is not a core point or border point.*



DBSCAN algorithm can be abstracted in the following steps:

1. Find all the neighbor points within eps and identify the core points or visited with more than MinPts neighbors.
2. For each core point if it is not already assigned to a cluster, create a new cluster.
3. Find recursively all its density connected points and assign them to the same cluster as the core point.
   A point *a* and *b* are said to be density connected if there exist a point *c* which has a sufficient number of points in its neighbors and both the points *a* and *b* are within the *eps distance*. This is a chaining process. So, if *b* is neighbor of *c*, *c* is neighbor of *d*, *d* is neighbor of *e*, which in turn is neighbor of *a* implies that *b* is neighbor of *a*.
4. Iterate through the remaining unvisited points in the dataset. Those points that do not belong to any cluster are noise.

Below is the DBSCAN clustering algorithm in pseudocode:

```
DBSCAN(dataset, eps, MinPts){

# cluster index

C = 1

for each unvisited point p in dataset {

      mark p as visited

      # find neighbors

      Neighbors N = find the neighboring points of p


      if |N|>=MinPts:

         N = N U N'

         if p' is not a member of any cluster:

            add p' to cluster C

}
```

Applications of Clustering in different fields:

1. **Marketing:** It can be used to characterize & discover customer segments for marketing purposes.
2. **Biology:** It can be used for classification among different species of plants and animals.
3. **Libraries:** It is used in clustering different books on the basis of topics and information.
4. **Insurance:** It is used to acknowledge the customers, their policies and identifying the frauds.
5. **City Planning:** It is used to make groups of houses and to study their values based on their geographical locations and other factors present.
6. **Earthquake studies:** By learning the earthquake-affected areas we can determine the dangerous zones.
7. **Image Processing**: Clustering can be used to group similar images together, classify images based on content, and identify patterns in image data.
8. **Genetics:** Clustering is used to group genes that have similar expression patterns and identify gene networks that work together in biological processes.
9. **Finance:** Clustering is used to identify market segments based on customer behavior, identify patterns in stock market data, and analyze risk in investment portfolios.
10. **Customer Service:** Clustering is used to group customer inquiries and complaints into categories, identify common issues, and develop targeted solutions.
11. **Manufacturing**: Clustering is used to group similar products together, optimize production processes, and identify defects in manufacturing processes.
12. **Medical diagnosis:** Clustering is used to group patients with similar symptoms or diseases, which helps in making accurate diagnoses and identifying effective treatments.
13. **Fraud detection:** Clustering is used to identify suspicious patterns or anomalies in financial transactions, which can help in detecting fraud or other financial crimes.

Unit VI- Data visualization

Data visualization:

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. Additionally, it provides an excellent way for employees or business owners to present data to non-technical audiences without confusion.

In the world of Big Data, data visualization tools and technologies are essential to analyze massive amounts of information and make data-driven decisions.

Characteristics of Effective Graphical Visual :

- It shows or visualizes data very clearly in an understandable manner.
- It encourages viewers to compare different pieces of data.
- It closely integrates statistical and verbal descriptions of data set.
- It grabs our interest, focuses our mind, and keeps our eyes on message as human brain tends to focus on visual data more than written data.
- It also helps in identifying area that needs more attention and improvement.
- Using graphical representation, a story can be told more efficiently. Also, it requires less time to understand picture than it takes to understand textual data.

What are the advantages and disadvantages of data visualization?

Something as simple as presenting data in graphic format may seem to have no downsides. But sometimes data can be misrepresented or misinterpreted when placed in the wrong style of data visualization. When choosing to create a data visualization, it's best to keep both the advantages and disadvantages in mind.

Advantages

Our eyes are drawn to colors and patterns. We can quickly identify red from blue, and squares from circles. Our culture is visual, including everything from art and advertisements to TV and movies. Data visualization is another form of visual art that grabs our interest and keeps our eyes on the message. When we see a chart, we quickly see trends and outliers. If we can see something, we internalize it quickly. It's storytelling with a purpose. If you've ever stared at a massive spreadsheet of data and couldn't see a trend, you know how much more effective a visualization can be.

Some other advantages of data visualization include:

- Easily sharing information.
- Interactively explore opportunities.
- Visualize patterns and relationships.

Disadvantages

While there are many advantages, some of the disadvantages may seem less obvious. For example, when viewing a visualization with many different datapoints, it's easy to make an inaccurate assumption. Or sometimes the visualization is just designed wrong so that it's biased or confusing.

Some other disadvantages include:

- Biased or inaccurate information.
- Correlation doesn't always mean causation.
- Core messages can get lost in translation.

Types of Data Visualization ;

Data visualization is very critical to market research where both numerical and categorical data can be visualized that helps in an increase in impacts of insights and also helps in reducing risk of analysis paralysis. So, data visualization is categorized into following categories :

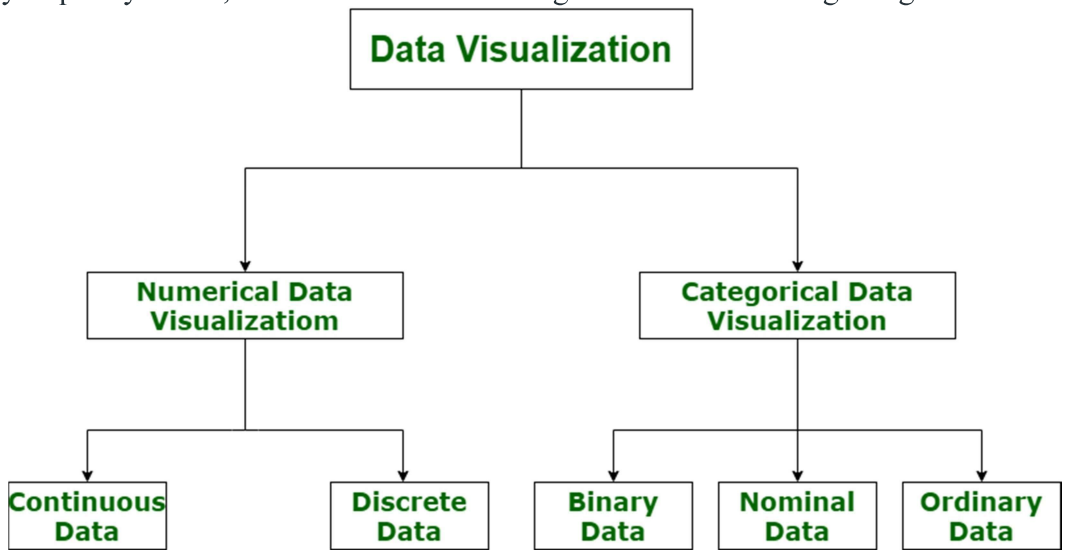


Figure – Types of Data Visualization

1. **Numerical Data :**
   Numerical data is also known as Quantitative data. Numerical data is any data where data generally represents amount such as height, weight, age of a person, etc. Numerical data visualization is easiest way to visualize data. It is generally used for helping others to digest large data sets and raw numbers in a way that makes it easier to interpret into action. Numerical data is categorized into two categories :
   - **Continuous Data –**
     It can be narrowed or categorized (Example: Height measurements).

- **Discrete Data –**
  This type of data is not "continuous" (Example: Number of cars or children's a household has).

The type of visualization techniques that are used to represent numerical data visualization is Charts and Numerical Values. Examples are Pie Charts, Bar Charts, Averages, Scorecards, etc.

2. **Categorical Data :**
   Categorical data is also known as Qualitative data. Categorical data is any data where data generally represents groups. It simply consists of categorical variables that are used to represent characteristics such as a person's ranking, a person's gender, etc. Categorical data visualization is all about depicting key themes, establishing connections, and lending context. Categorical data is classified into three categories :
   - **Binary Data –**
     In this, classification is based on positioning (Example: Agrees or Disagrees).
   - **Nominal Data –**
     In this, classification is based on attributes (Example: Male or Female).
   - **Ordinal Data –**
     In this, classification is based on ordering of information (Example: Timeline or processes).

   The type of visualization techniques that are used to represent categorical data is Graphics, Diagrams, and Flowcharts. Examples are Word clouds, Sentiment Mapping, Venn Diagram, etc.

Data Visualization Techniques

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

Box Plots

The image above is a box plot**.** A boxplot is a standardized way of displaying the distribution of data based on a five-number summary ("minimum", first quartile (Q1), median, third quartile (Q3), and "maximum"). It can tell you about your outliers and what their values are. It can also tell you if your data is symmetrical, how tightly your data is grouped, and if and how your data is skewed.

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.

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 Plot:**

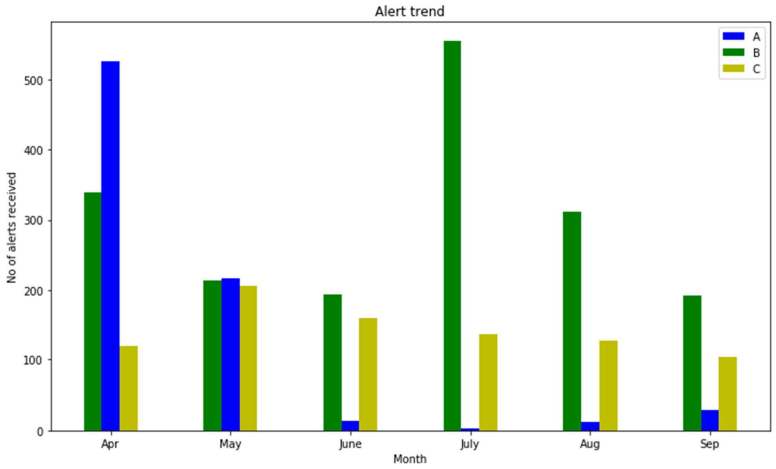
1. This is one of the widely used plots, that we would have seen multiple times not just in data analysis, but we use this plot also wherever there is a trend analysis in many fields.
2. Though it may seem simple it is powerful in analyzing data like **sales figures every week, revenue from a product**, **Number of visitors to a site on each day of a week**, etc.

Implementation:

- The bar plot is present in the **Matplotlib** package.

The code snippet is as follows:

plt.bar(x,y)



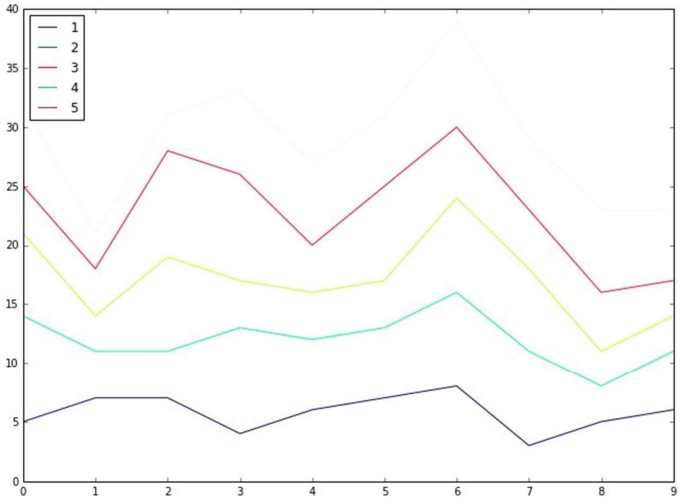
- **Stacked Bar Graph:** It is a bar style graph that has various components stacked together so that apart from the bar, the components can also be compared to each other.
- **Stacked Column Chart:** It is similar to a stacked bar; however, the data is stacked horizontally.
- **Area Chart:** It combines 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 plot:**

1. This is the plot that you can see in the nook and corners of any sort of analysis between 2 variables.

2. The line plots are nothing but the values on a series of data points will be connected with straight lines.
3. The plot may seem very simple but it has more applications not only in machine learning but in many other areas.

4. **Implementation:**
5. The line plot is present in the **Matplotlib** package.

6. The code snippet is as follows:
7. plt.plot(x,y)



- **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:**

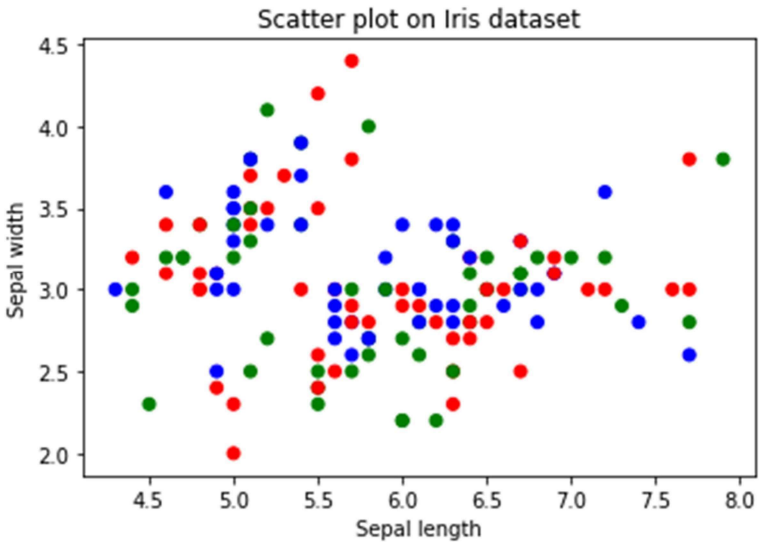
1. It is one of the most commonly used plots used for visualizing simple data in Machine learning and Data Science.
2. This plot describes us as a representation, where each point in the entire dataset is present with respect to any 2 to 3 features(Columns).
3. Scatter plots are available in both 2-D as well as in 3-D. The 2-D scatter plot is the common one, where we will primarily try to find the patterns, clusters, and separability of the data.

Implementation:

- The scatter plot is present in the **Matplotlib** package.

The code snippet is as follows:

plt.scatter(x,y)



- **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.

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.
If you're looking at a web page and you want to know which areas get the most attention, a heat map shows you in a visual way that's easy to assimilate and make decisions from. It is a graphical representation of data where the individual values contained in a matrix are represented as colours. Useful for two purposes: for visualizing correlation tables and for visualizing missing values in the data. In both cases, the information is conveyed in a two-dimensional table.

Note that heat maps are useful when examining a large number of values, but they are not a replacement for more precise graphical displays, such as bar charts, because colour differences cannot be perceived accurately.

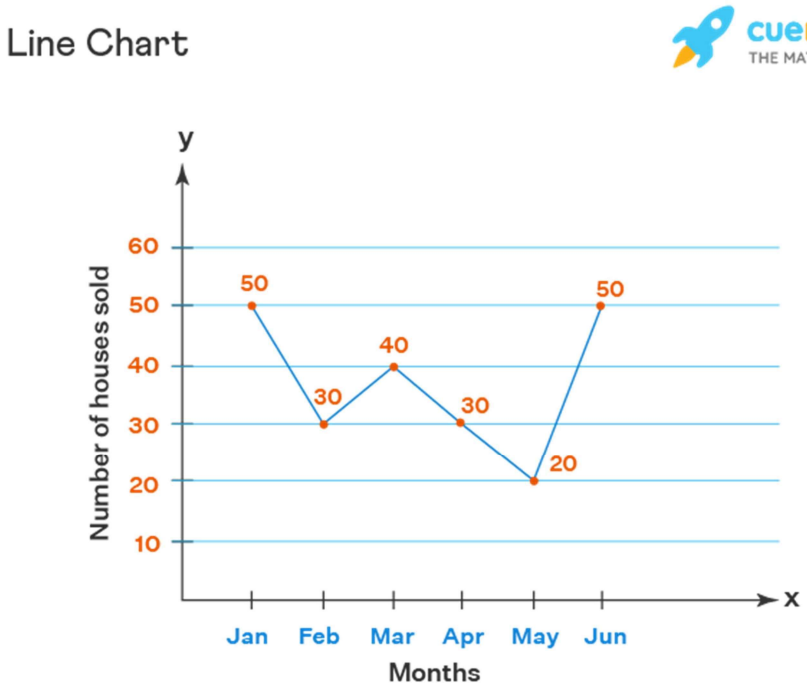
Charts

Line Chart

The simplest technique, a line plot is used to plot the relationship or dependence of one variable on another. To plot the relationship between the two variables, we can simply call the plot function.

Example:

Traders, investors, and financial officers use the line chart to depict the high and low in the market for a particular value since it provides a clear visualization of the data.

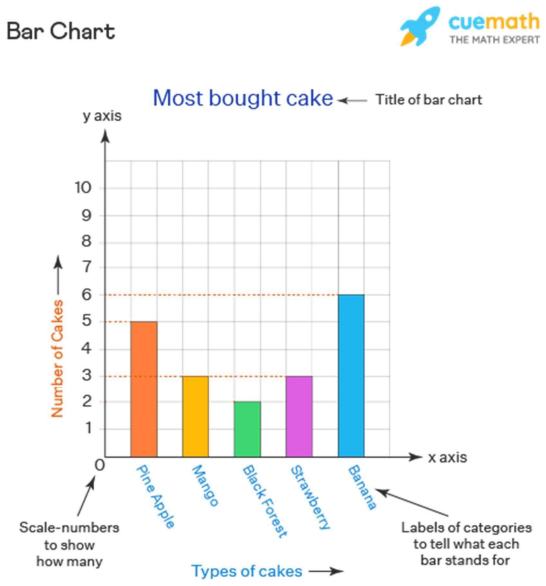


A line chart consists of a horizontal line i.e. x-axis and a vertical line i.e. y-axis to represent the data. The x-axis shows the time period whereas the y-axis shows the item that is being measured. A line chart clearly shows the increasing or decreasing trend of a particular item. In the line chart above, we can see the dark points, those are the data points that show the quantity or a number that matches a particular time in the x-axis. For example, in the month of March, around 40 houses were sold. The line connecting these data points depicts a line chart.

Bar Charts

Bar charts are used for comparing the quantities of different categories or groups. Values of a category are represented with the help of bars and they can be configured with vertical or horizontal bars, with the length or height of each bar representing the value.

Example:



From the above bar chart, we can easily say that banana flavored cake is the most bought fruit cake that Rose purchased.

Types of Bar Chart

Bar Charts are mainly classified into two types:

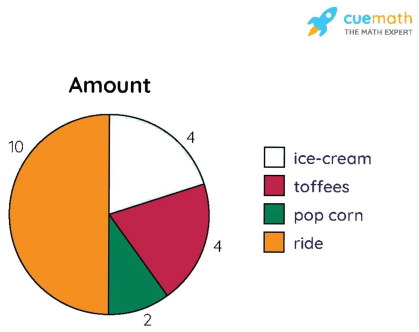
- **Horizontal Bar Charts:** When the given data is represented via horizontal bars on a graph (chart) paper such graphs are known as horizontal bar charts. These horizontal rectangular bars show the measures of the given data. In this type, the categories of the data are marked on the x-axis and y-axis. The y-axis category shows the horizontal representation of the bar chart.
- **Vertical Bar Charts:** When the given data is represented via vertical bars on a graph (chart) paper it is known as a vertical bar chart. These vertical rectangular bars represent the measure of data. The rectangular bars are vertically drawn on the x-axis and the y-axis. These rectangular bars represent the quantity of the variables written on the x-axis.

Pie Chart

It is a circular statistical graph which decides slices to illustrate numerical proportion. Here the arc length of each slide is proportional to the quantity it represents. As a rule, they are used to compare the parts of a whole and are most effective when there are limited components and when text and percentages are included to describe the content. However, they can be difficult to interpret because the human eye has a hard time estimating areas and comparing visual angles.

Example: Observe the following pie chart that represents the money spent by Ana at the funfair. The indicated color shows the amount spent on each category. The total value of the data is 20 and the amount spent on each category is interpreted as follows:

- Ice Cream - 4
- Toffees - 4
- Popcorn - 2
- Rides - 10



Scatter Charts

Another common visualization technique is a scatter plot that is a two-dimensional plot representing the joint variation of two data items. Each marker (symbols such as dots, squares and plus signs) represents an observation. The marker position indicates the value for each observation. When you assign more than two measures, a scatter plot matrix is produced that is a series scatter plot displaying every possible pairing of the measures that are assigned to the visualization. Scatter plots are used for examining the relationship, or correlations, between X and Y variables.

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. It shows the chronological sequence of past or future events on a timescale.

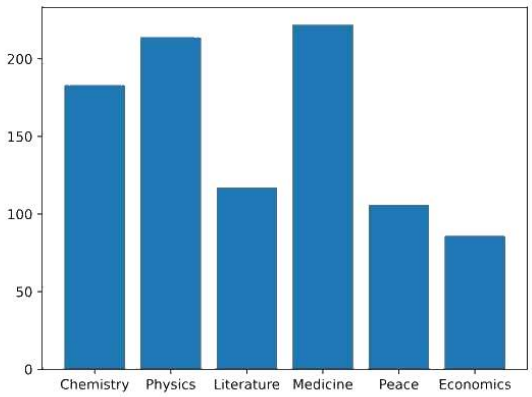
Tree Maps

A tree map 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 colors 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 colored, sized or both according to chosen attributes. They make efficient use of space, thus display thousands of items on the screen simultaneously.

Types Of Graphs:

Bar Graphs

1. Bar graphs are used show the distribution of qualitative (categorical) data.
2. It shows the frequency of values in the data. Frequency is the amount of times that value appeared in the data.
3. Each category is represented with a bar. The height of the bar represents the frequency of values from that category in the data.
4. Here is a bar graph of the number of people who have won a Nobel Prize in each category up to the year 2020:



Some of the categories have existed longer than others. Multiple winners are also more common in some categories. So there is a different number of winners in each category.
Note: Bar graphs are similar to histograms, which are used for quantitative data.

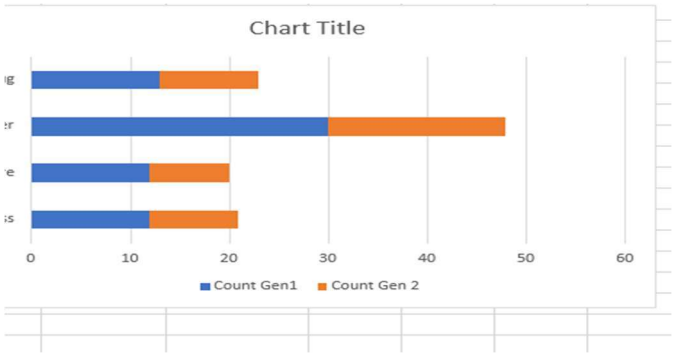
Stacked Bar Chart

1. Stacked bar charts are used to highlights the total amount of contribution for each category.
2. This is done by stacking the bars at the end of each other.
3. The charts are used when you have more than one data column.

Example
We want to find out the total number of generation 1 and 2 Pokemons in each of these type 1 categories: "Grass", "Fire", "Water" and "Bug".

Type	Count Gen1	Count Gen 2
Grass	12	9
Fire	12	8
Water	30	18
Bug	13	10

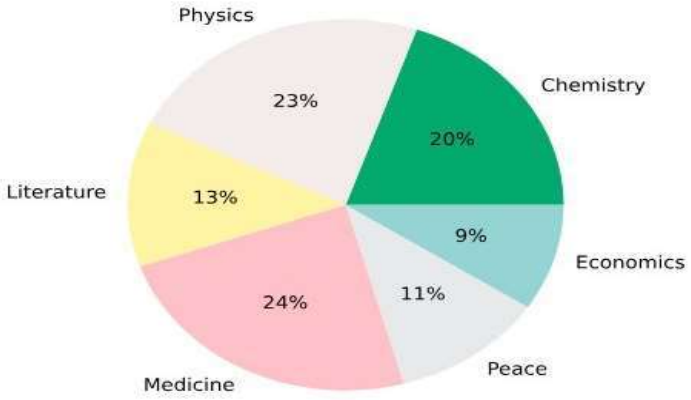
Stacked Bar chart based on above values



1. The chart gives a visual overview for the total number of "Grass", "Fire", "Water" and "Bug" type Pokemons in both generation 1 and 2.
2. Generation 1 Pokemons are shown in blue and generation 2 Pokemons are shown in orange.
3. This chart shows that "Water" type Pokemons are the most common and "Fire" type Pokemons are the least common.

Pie Charts

1. Pie graphs are used to show the distribution of qualitative (categorical) data.
2. It shows the frequency or relative frequency of values in the data.
3. Frequency is the amount of times that value appeared in the data. Relative frequency is the percentage of the total.
4. Each category is represented with a slice in the 'pie' (circle). The size of each slice represents the frequency of values from that category in the data.
5. Here is a pie chart of the number of people who have won a Nobel Prize in each category up to the year 2020:



1. This pie chart shows relative frequency. So each slice is sized by the percentage for each category.

2. Some of the categories have existed longer than others. Multiple winners are also more common in some categories. So there is a different number of winners in each category.

Doughnut Chart

1. Doughnut charts arrange the data as slices in a circle with hollow center.
2. Doughnut charts are often used when you have more than one data column.

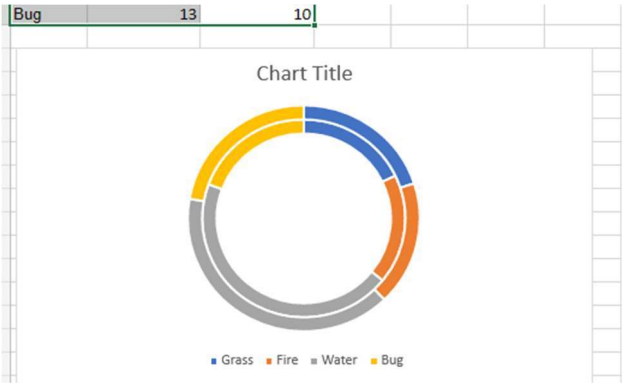
Note: A doughnut chart with one data column shows the same information as a 2-D pie chart.

Example

We want to find the proportion of types "Grass", "Fire", "Water" and "Bug". in generation 1 Pokemon and compare it to the proportions in generation 2.

Type	Count Gen1	Count Gen 2
Grass	12	9
Fire	12	8
Water	30	18
Bug	13	10

Doughnut chart based on above data



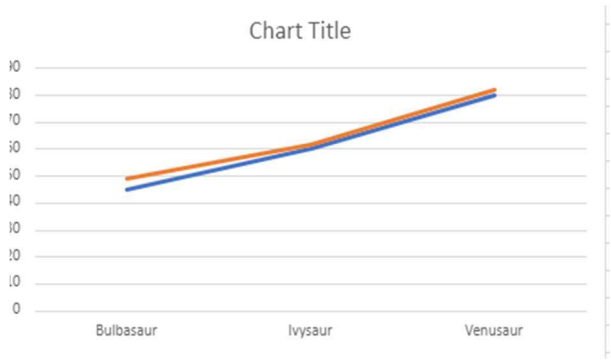
1. The chart gives a visual overview of the number of 1st and 2nd generation Grass, Fire, Water and Bug type Pokemon.
2. Generation 1 Pokemon are shown in the inner circle and generation 2 Pokemon are shown in the outer circle.
3. Type "Grass" is shown in blue, "Fire" in orange, "Water" in gray and "Bug" in yellow.
4. This chart shows that "Water" type Pokemon are the most common in both generations.

Line Charts

1. Line charts show the data as a continuous line.
2. Line charts are typically used for showing trends over time.
3. In Line charts, the horizontal axis typically represents time.
4. Line charts are used with data which can be placed in an order, from low to high.

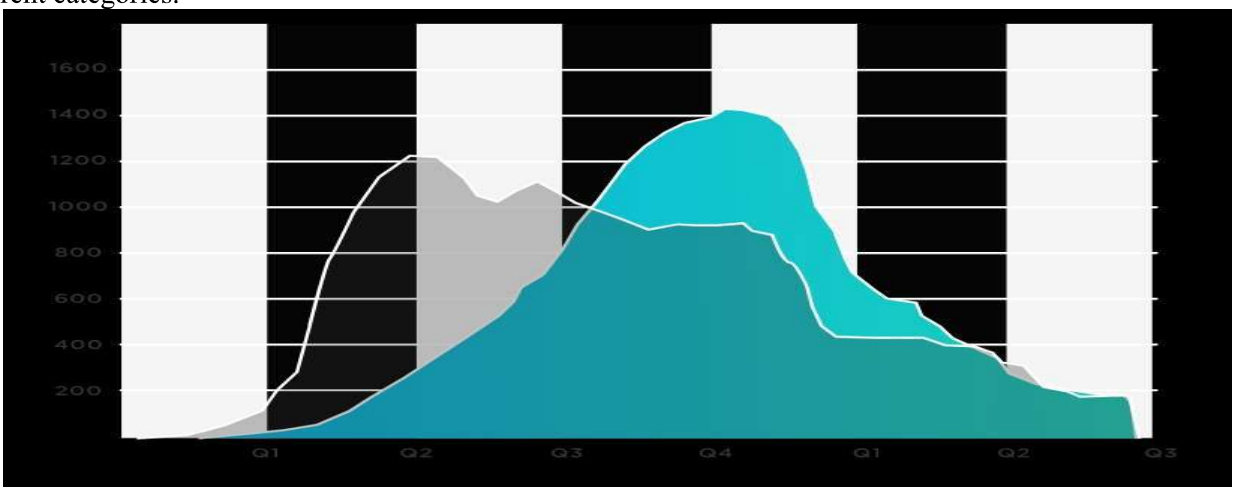
	Name	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed
2	Bulbasaur	45	49	49	65	65	45
3	Ivysaur	60	62	63	80	80	60
4	Venusaur	80	82	83	100	100	80
5							

Line chart



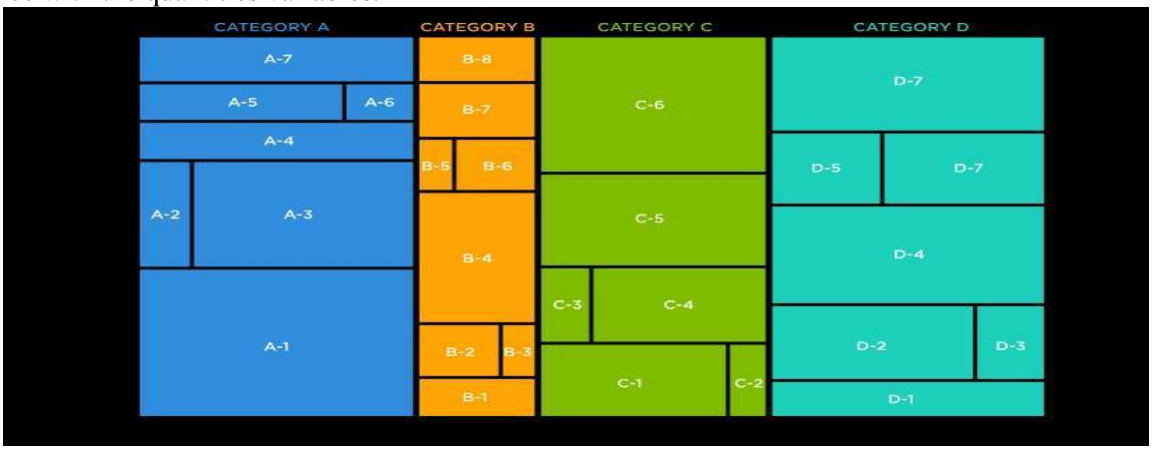
Area Chart

An area chart is a visual representation of data that utilizes both lines and filled areas to convey information. This type of chart is particularly effective in showcasing data trends and variations over a specified period or across different categories.



Treemapping Chart

A treemap chart is created using a data visualization technique that visualizes hierarchical data in the form of nested rectangles. The tree-like structure uses rectangles of decreasing sizes, hence called “nesting.” The data in a treemap chart is organized using rectangles. The plot colors and dimensions of the rectangles are calculated in accordance with the quantities variables.



Waterfall Chart?

A waterfall chart is a graphical tool primarily used to show the collective influence of successive positive and negative variables on an initial starting point. In essence, it furnishes a well-structured and lucid manner of depicting the gradual transitions, emphasizing the intricate process through which diverse factors contribute to a final consequence, along with their net impact.


125
100
75
50
JAN
MAR
JUNE
AUG
OCT
DEC
2022
INCREASE
DECREASE
TOTAL

THANK YOU