1

INTRODUCTION TO DATA SCIENCE

Unit Structure

1.1 What is Data?

1.2 Different kinds of data

1.3 Introduction to high level programming language

1.4 Integrated Development Environment (IDE)

1.5 Exploratory Data Analysis (EDA)

1.6 Data Visualization

1.7 Different types of data sources

1.8 Summary

1.9 Questions

1.1 WHAT IS DATA?

In general, data is a distinct piece of information that is gathered and translated for some purpose. If data is not formatted in a specific way, it does not valuable to computers or humans. Data can be available in terms of different forms, such as bits and bytes stored in electronic memory, numbers or text on pieces of paper, or facts stored in a person's mind. Since the invention of computers, people have used the word data to mean computer information, and this information is transmitted or stored. There are different kinds of data; such are as follows:

Sound

Video

Single character

Number (integer or floating-point)

Picture

Boolean (true or false)

Text (string)

In a computer's storage, data is stored in the form of a series of binary digits (bits) that contain the value 1 or 0. The information can be in terms of pictures, text documents, software programs, audio or video clips, or other kinds of data. The computer data may be stored in files and folders on the computer's storage, and processed by the computer's CPU, which utilizes logical operations to generate output (new data) form input data.

As the data is stored on the computer in binary form (zero or one), which can be processed, created, saved, and stored digitally. This allows data to be sent from one computer to another with the help of various media devices or a network connection. Furthermore, if you use data multiple times, it does not deteriorate over time or lose quality.

Examples of computer data

0143 0157 0155 0160 0165 0164 0145 0162 0040 0150 0157 0160

Joe,Smith, 1234 Circle,SLC,UT,8404,8015553211

01100011011011110110110101110000011101010111010001

1.2 DIFFERENT KINDS OF DATA

There are two types of data that are given below:

1. Qualitative Data: Qualitative data is information that represents some characteristics or attributes. It depicts descriptions that cannot be counted, measured, or easily expressed with the help of numbers. It can be collected from audio, text, and pictures. It is shared via data visualization tools, such as concept maps, clouds, infographics, timelines, and databases. For instance, collecting data on attributes such as honesty, intelligence, creativity, wisdom, and cleanliness about students of any class would be considered as a sample of qualitative data.

Typically, it has two types: ethnographic data and interpretive data. The collection of data for understanding how a group assigns context for an event, it is known as ethnographic data. The data, which is collected to understand the experience and feelings of an individual's personal about the event, it is known as interpretive data.

Methods of collecting qualitative data

Qualitative data is widely collected by asking open-ended questions, or through direct or indirect observation. Below is given common methodologies of collecting the Qualitative data:

- Interviews
- Focus groups
- Case studies
- Cultural, or ethnographic, research
- Pulling from existing records
- Participant observation
- Open-ended survey questions
- Fieldwork

Qualitative data analysis

Qualitative data can be analyzed through being either deductive or inductive approach. In the deductive technique, the analyst starts with a question and evaluate data subjectively in terms of the question. In the inductive technique, he or she simply evaluates the data to look for patterns as in this approach; the analyst has no agenda. Frequently, the inductive process is also known as grounded theory. Generally, an inductive technique takes more time as compared to the deductive technique.

Qualitative analysis tools

Qualitative data analysis depends on the analog and digital tools to organize, systematize, and analyze non-numeric data.

- **SWOT analysis:** It is a framework that means strengths, weaknesses, opportunities, and threats analysis. It is used to identify and analyze the internal and external factors that can have an effect on the activity of a place, person, project, or product. The tool is beneficial to offer a snapshot to understand the qualitative dynamics that able to affect success.
- **Porter's five forces:** It is a framework that is used to improve the SWOT analysis. It is developed by Harvard professor Michael E. Porter, which improves SWOT analysis with the help of identifying and analyzing the internal and external factors that able to effect success.

Furthermore, QDAS (qualitative data analysis software) helps to collect and analyze qualitative data in a short time. It includes features such as coding for sentiment analysis and text interpretation, transcription analysis, and recursive abstraction.

Pros and cons of qualitative data

The methods content and observational help qualitative data researchers to collect the correct data to the actual experience and also help to avoid the Hawthorne effect. Including the qualitative data in reporting helps to add color to the story with the help of carrying a generalized solution into a less abstract view through real examples from actual people.

To collect and analyze qualitative data can be very time-consuming. Therefore, the researchers of qualitative data use sampling in their analysis. It can be difficult to scale the result out to discover when small samples of data are used.

The analysts can use numerical calculations and mathematical formulas to analyze the quantifiable data, and it can be put directly into a database. Before the qualitative data statistically examined for patterns or meaning, it must be classified by descriptive parameters, such as physical or traits characteristics.

Although the analysts can easily analyze the quantitative data through any software tool like spreadsheets, the analysis of qualitative data depends on the researcher's how they have skills and experience, which helps to create parameters from a small sampling, and larger data set can be examined.

2. Quantitative Data: These types of data can be measured but not simply observed. The data can be numerically represented and used for statistical analysis and mathematical calculations. For example, these mathematical derivations can be used in real-life decisions. Also, the number of students participate in different games from a class; the mathematical calculation gives an estimate of how many students are playing in which sport.

This data is any quantifiable information that is used to answer questions such as How much?" "How often?" "How many?". These data can be conveniently evaluated by using mathematical techniques and also can be verified. Usually, quantitative data is collected for statistical analysis sent across to a particular section of a population with the help of surveys, questionnaires, or polls. Furthermore, quantitative data helps to measure several parameters controllable as it includes mathematical derivations.

Types of Quantitative Data

There are various types of quantitative data; such are as follows:

- **Measurement of physical objects:** It is commonly used to calculate the measurement of any physical thing, for instance, assigned each cubicle to the newly joined employees in any organization is carefully measured by the HR executive.
- **Counter:** It is used to count equally with entities. For example, the calculation of a particular application of how many people have downloaded it from the App Store.
- **Sensory calculation:** It is a mechanism to sense naturally the measured parameters that help to create a constant source of information. For example, electromagnetic information is converted to a string of numerical data through a digital camera.
- **Quantification of qualitative entities:** It helps to identify numbers to qualitative information. For example, you are asking to share the likelihood of recommendation on a scale of 0-10 with respondents of an online survey.
- **Projection of data:** It can be used for future projection of data with the help of mathematical analysis tools and algorithms. For instance, a marketer, after launching a new product with a thorough analysis, predicts growth in production.

The methods of collection the quantitative data

The main two types of quantitative data collection methods are given below:

I. **Surveys**

Surveys were traditionally conducted with the help of paper-based methods and have gradually evolved into online mediums. Collecting the closed-ended questions form a major part of these surveys is more appropriate in the collection of quantitative data. The survey contains answer options for a particular question. Also, surveys are unified to collect feedback from an audience. Surveys are classified into different category on the basis of the time involved in completing surveys:

- **Longitudinal Studies:** In this, a market researcher conducts surveys from a specific time period to another as it is a type of observational research. When the primary objective is to collect and analyze a pattern in data, this survey is often implemented.
- **Cross-sectional studies:** In this, a market researcher conducts surveys at a particular time period. It helps to understand a particular subject from the sample at a certain time period by implementing a questionnaire.

There are some principles given below to administer a survey to collect quantitative data:

- **Use of Different Question Types:** Closed-ended questions have to be used in a survey to collect quantitative data. These questions can be a combination of several types of questions as well as multiple-choice questions like rating scale questions, semantic differential scale questions, and more. It helps to collect data, which can be understood and analyzed.
- **Fundamental Levels of Measurement:** Collection of the quantitative data, four measurement scales, ordinal, nominal, interval, and ratio scales, are fundamental for creating a multiple-choice question in a survey. These four fundamentals are most important as no multiple-choice questions can be created without the fundamentals.
- **Survey Distribution and Survey Data Collection:** To collect quantitative data, it is also the other important principle of the survey process. There are various ways of survey distribution for collecting data, some common methods are Email, SMS survey, QR code, Embed survey in a website, QuestionPro app, etc.

II. **One-on-one Interviews :**

It was also a traditional method to collect quantitative data. Although it was conducted face-to-face, it has been moved to telephonic and online platforms. A marketer can collect extensive data from the participants with the help of interviews. Quantitative interviews are extremely and play an important role in collecting information. There are three important sections that help to gather quantitative data through interviews. These major sections are given below:

- **Face-to-Face Interviews:** In addition to the already asked survey questions, an interviewer can prepare a list of important interview questions. Thus, interviewers will be capable of providing complete details about the topic under discussion. Also, an interviewer will get help to collect more details about the topic by managing to bond with the interviewee on a personal level, through which the responses also improve.

- **Computer-Assisted Personal Interview:** In this method, the interviewers are able to enter the collected data directly into the computer or any other similar device. It is also called a one-on-one interview technique, which technique helps to reduce the processing time and provides benefits to interviewers as they do not require to carry a hardcopy of questionnaires and only enter the answers on the laptop.

- **Online/Telephonic Interviews:** Although, telephone-based interviews are not a modern technique. These types of interviews have also moved to online mediums like Zoom or Skype, which provides the option to online interviews over the network. Online interview is beneficial that helps to overcome the issue of distance between interviewer and interviewee and save their time.

- However, the interview is only a phone call in case of telephonic interviews.

Analysis methods of quantitative data

Although the collection of data is a crucial part of the research process, it also needs to analyze for making it understandable. So, there are several methods to analyze quantitative data that have collected in surveys. These methods are given below:

- **Cross-tabulation:** It is the most preferred and widely used method for quantitative data analysis. To evaluate an effective result between different data-sets in the research study, it uses a basic tabular form. It contains data that have some connection with each other.

- **Trend analysis:** It provides the option to check out the quantitative data if it has been gathered over a long period of time. It also helps to gather feedback about data changes over time.

- **MaxDiff analysis**: It helps to gauss the customer preferences for purchase and determine what rank of parameters is higher as compared to others in this process. This method is also known as the "best-worst" method as it is time-consuming. Furthermore, this method can be used interchangeably, and it is much easier to implement.

- **Conjoint analysis:** It is most similar to the MaxDiff analysis method that helps to analyze parameters to make a better decision. This method has the ability to gather and analyze advanced metrics that offer the parameters that rank the most important, including in-depth insight into purchasing decisions.

- **Gap analysis:** It is another type of quantitative data analysis method that uses a side-by-side matrix to describe data, which provides a way to measure the difference between actual performance and expected performance. The data analysis by this method helps to describe the things that need to be complete this gap and also helps to measure gaps in performance.

- **SWOT analysis:** It is a framework that means strengths, weaknesses, opportunities, and threats analysis. It has the ability to identify SWOT of an organization or product or service. Also, it helps to create effective business strategies and offers a complete picture of the competition.

- **TURF analysis:** It evaluates the total market reach of a product or service or a mix of both, which stands for total unduplicated reach and frequency analysis. This method is helpful in understanding the avenues and the frequency in any organization.

- **Text analysis:** In this method, intelligent tools work on easily understandable data. They make more quantify or fashion qualitative and open-ended data of this data. This method is helpful in the case when the collected data is unstructured and needs to convert into a structural way that makes it understandable.

Examples of Quantitative Data

Some examples of quantitative data are given below that can help to easily understand which types of data are known as quantitative data.

- I updated my cellphone six times in a quarter.
- My uncle lost 20 kg last year.
- The latest mobile application is downloaded by 83 people.
- My son grew up by 2 inches last year.
- 600 employees attended the meeting.
- 44% of people like online shopping rather than going to the mall.

Advantages of Quantitative Data

- **Conduct in-depth research:** It is highly possible that the research will be detailed, as quantitative data can be statistically analyzed.

- **Minimum bias:** There are many examples in research if personal bias is involved, it generates incorrect results. The numerical nature of quantitative data reduces the personal bias that helps to lead correct data.

Disadvantages of Quantitative Data

Some of the disadvantages of quantitative data are as follows:

- **Depends on question types:** Collection of the quantitative data, the result is dependent on the types of questions. While collecting quantitative data, the researcher's objective of research and knowledge of questions are most important.
- **Restricted information:** On the basis of the collected data, it can be more difficult for researchers to make decisions as quantitative data is not descriptive.

How a computer process data into information?

A computer uses following four functions to process data into information by using software and hardware.

1. Input

First, the data must receive input before a computer starts to process anything. For instance, to enter input into the computer have to type on the keyboard.

2. Process

A computer uses a program to process the data into information, which data has received through input. The program may organize, calculate, or manipulate the data to create understandable information.

3. Output

It is displayed as output to the user after the data is processed into information. For example, when you use the Windows Calculator, the program displays the information on your monitor screen.

4. Storage

Finally, the created information is stored on the computer for future retrieval. It uses storage media like hard disk, floppy disk, etc.

5. What is the difference between data and information?

Parameters	**Data**	**Information**
Description	There are two variables qualitative and quantitative that helps to develop ideas or conclusions.	It is a collection of data that contains valuable meaning and news.
Format	It is in the form of letters, numbers, or a collection of characters.	It is in the form of Ideas and inferences.
Represented in	The data can be represented in a graph, tabular data, structured data tree, and more.	It can be represented based on the given data, language, ideas, etc.

Feature	The data alone doesn't have any valuable meaning. It is raw and a single unit.	It is a collection of data and a product that has a logical meaning.
Interrelation	It is related to the collected information.	It is related to the information that is processed.
Meaning	It includes row data that does not have any specific purpose.	It does have logical meaning, which has assigned by interpreting data.
Contains	It is unprocessed raw factors.	It is processed in a meaningful way.
Dependence	The data does not depend on information.	A piece of information depends on Data.
Support for Decision making	As it does not have any specific purpose, hence cannot be used for decision making.	It provides useful information, hence widely used for decision making.
Measuring unit	The data is measured in bytes and bits.	The information is measured in meaningful units such as quantity, time, and more.
Knowledge level	Data is low-level knowledge.	Information is the second level of knowledge.
Characteristic	It cannot be sold to the public as it is the property of an organization.	Information can be sold to the public.
Usefulness	The data may not be useful as it is collected by the researcher.	Information is easily available to the researcher for use; hence it is valuable and useful.

1.3 INTRODUCTION TO HIGH LEVEL PROGRAMMING LANGUAGE

The high-level language is a programming language that allows a programmer to write the programs which are independent of a particular type of computer. The high-level languages are considered as high-level because they are closer to human languages than machine-level languages.

When writing a program in a high-level language, then the whole attention needs to be paid to the logic of the problem.

A compiler is required to translate a high-level language into a low-level language.

Advantages of a high-level language

o The high-level language is easy to read, write, and maintain as it is written in English like words.

o The high-level languages are designed to overcome the limitation of low-level language, i.e., portability. The high-level language is portable; i.e., these languages are machine-independent.

1.4 INTEGRATED DEVELOPMENT ENVIRONMENT (IDE)

An integrated development environment (IDE) is a software application that helps programmers develop software code efficiently. It increases developer productivity by combining capabilities such as software editing, building, testing, and packaging in an easy-to-use application. Just as writers use text editors and accountants use spreadsheets, software developers use IDEs to make their job easier.

Why are IDEs important?

You can use any text editor to write code. However, most integrated development environments (IDEs) include functionality that goes beyond text editing. They provide a central interface for common developer tools, making the software development process much more efficient. Developers can start programming new applications quickly instead of manually integrating and configuring different software. They also don't have to learn about all the tools and can instead focus on just one application. The following are some reasons why developers use IDEs:

- **Code editing automation**

Programming languages have rules for how statements must be structured. Because an IDE knows these rules, it contains many intelligent features for automatically writing or editing the source code.

- **Syntax highlighting**

An IDE can format the written text by automatically making some words bold or italic, or by using different font colors. These visual cues make the source code more readable and give instant feedback about accidental syntax errors.

- **Intelligent code completion**

Various search terms show up when you start typing words in a search engine. Similarly, an IDE can make suggestions to complete a code statement when the developer begins typing.

- **Refactoring support**

Code refactoring is the process of restructuring the source code to make it more efficient and readable without changing its core functionality. IDEs can auto-refactor to some extent, allowing developers to improve their code quickly and easily. Other team members understand readable code faster, which supports collaboration within the team.

- **Local build automation**

IDEs increase programmer productivity by performing repeatable development tasks that are typically part of every code change. The following are some examples of regular coding tasks that an IDE carries out.

- **Compilation**

An IDE compiles or converts the code into a simplified language that the operating system can understand. Some programming languages implement just-in-time compiling, in which the IDE converts human-readable code into machine code from within the application.

- **Testing**

The IDE allows developers to automate unit tests locally before the software is integrated with other developers' code and more complex integration tests are run.

- **Debugging**

Debugging is the process of fixing any errors or bugs that testing reveals. One of the biggest values of an IDE for debugging purposes is that you can step through the code, line by line, as it runs and inspect code behavior. IDEs also integrate several debugging tools that highlight bugs caused by human error in real time, even as the developer is typing.

What are the types of IDEs?

Integrated development environments (IDEs) can be broadly classified into several different categories, depending on the application development they support and how they work. However, many IDE software applications can fit into multiple categories. The following are some types of IDEs:

Local IDEs

Developers install and run local IDEs directly on their local machines. They also have to download and install various additional libraries depending on their coding preferences, project requirements, and development language. While local IDEs are customizable and do not require an internet connection once installed, they present several challenges:

- They can be time consuming and difficult to set up.
- They consume local machine resources and can slow down machine performance significantly.
- Configuration differences between the local machine and the production environment can give rise to software errors.

Cloud IDEs

Developers use cloud IDEs to write, edit, and compile code directly in the browser so that they don't need to download software on their local machines. Cloud-based IDEs have several advantages over traditional IDEs. The following are some of these advantages:

Standardized development environment

Software development teams can centrally configure a cloud-based IDE to create a standard development environment. This method helps them avoid errors that might occur due to local machine configuration differences.

Platform independence

Cloud IDEs work on the browser and are independent of local development environments. This means they connect directly to the cloud vendor's platform, and developers can use them from any machine.

Better performance

Building and compiling functions in an IDE requires a lot of memory and can slow down the developer's computer. The cloud IDE uses compute resources from the cloud and frees up the local machine’s resources.

1.5 EXPLORATORY DATA ANALYSIS (EDA)

Learn everything you need to know about exploratory data analysis, a method used to analyze and summarize data sets.

Exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods. It helps determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions.

EDA is primarily used to see what data can reveal beyond the formal modeling or hypothesis testing task and provides a provides a better understanding of data set variables and the relationships between them. It can also help determine if the statistical techniques you are considering for data analysis are appropriate. Originally developed by American mathematician John Tukey in the 1970s, EDA techniques continue to be a widely used method in the data discovery process today.

Why is exploratory data analysis important in data science?

The main purpose of EDA is to help look at data before making any assumptions. It can help identify obvious errors, as well as better understand patterns within the data, detect outliers or anomalous events, find interesting relations among the variables.

Data scientists can use exploratory analysis to ensure the results they produce are valid and applicable to any desired business outcomes and goals. EDA also helps stakeholders by confirming they are asking the right questions. EDA can help answer questions about standard deviations, categorical variables, and confidence intervals. Once EDA is complete and insights are drawn, its features can then be used for more sophisticated data analysis or modeling, including machine learning.

Exploratory data analysis tools

Specific statistical functions and techniques you can perform with EDA tools include:

- Clustering and dimension reduction techniques, which help create graphical displays of high-dimensional data containing many variables.
- Univariate visualization of each field in the raw dataset, with summary statistics.
- Bivariate visualizations and summary statistics that allow you to assess the relationship between each variable in the dataset and the target variable you're looking at.
- Multivariate visualizations, for mapping and understanding interactions between different fields in the data.
- K-means Clustering is a clustering method in unsupervised learning where data points are assigned into K groups, i.e. the number of clusters, based on the distance from each group's centroid. The data points closest to a particular centroid will be clustered under the same category. K-means Clustering is commonly used in market segmentation, pattern recognition, and image compression.
- Predictive models, such as linear regression, use statistics and data to predict outcomes.

Types of exploratory data analysis

There are four primary types of EDA:

- **Univariate non-graphical.** This is simplest form of data analysis, where the data being analyzed consists of just one variable. Since it's a single variable, it doesn't deal with causes or relationships. The main purpose of univariate analysis is to describe the data and find patterns that exist within it.

- **Univariate graphical.** Non-graphical methods don’t provide a full picture of the data. Graphical methods are therefore required. Common types of univariate graphics include:

  - Stem-and-leaf plots, which show all data values and the shape of the distribution.

  - Histograms, a bar plot in which each bar represents the frequency (count) or proportion (count/total count) of cases for a range of values.

  - Box plots, which graphically depict the five-number summary of minimum, first quartile, median, third quartile, and maximum.

- **Multivariate nongraphical:** Multivariate data arises from more than one variable. Multivariate non-graphical EDA techniques generally show the relationship between two or more variables of the data through cross-tabulation or statistics.

- **Multivariate graphical:** Multivariate data uses graphics to display relationships between two or more sets of data. The most used graphic is a grouped bar plot or bar chart with each group representing one level of one of the variables and each bar within a group representing the levels of the other variable.

Other common types of multivariate graphics include:

- Scatter plot, which is used to plot data points on a horizontal and a vertical axis to show how much one variable is affected by another.

- Multivariate chart, which is a graphical representation of the relationships between factors and a response.

- Run chart, which is a line graph of data plotted over time.

- Bubble chart, which is a data visualization that displays multiple circles (bubbles) in a two-dimensional plot.

- Heat map, which is a graphical representation of data where values are depicted by color.

Exploratory Data Analysis Tools

Some of the most common data science tools used to create an EDA include:

- **Python:** An interpreted, object-oriented programming language with dynamic semantics. Its high-level, built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for rapid application development, as well as for use as a scripting or glue language to connect existing components together. Python and EDA can be used together to identify missing values in a data set, which is important so you can decide how to handle missing values for machine learning.

- **R:** An open-source programming language and free software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing. The R language is widely used among statisticians in data science in developing statistical observations and data analysis.

1.6 DATA VISUALIZATION?

- Data visualization is a graphical representation of quantitative information and data by using visual elements like graphs, charts, and maps.

- Data visualization convert large and small data sets into visuals, which is easy to understand and process for humans.

- Data visualization tools provide accessible ways to understand outliers, patterns, and trends in the data.

- In the world of Big Data, the data visualization tools and technologies are required to analyze vast amounts of information.

- Data visualizations are common in your everyday life, but they always appear in the form of graphs and charts. The combination of multiple visualizations and bits of information are still referred to as Infographics.

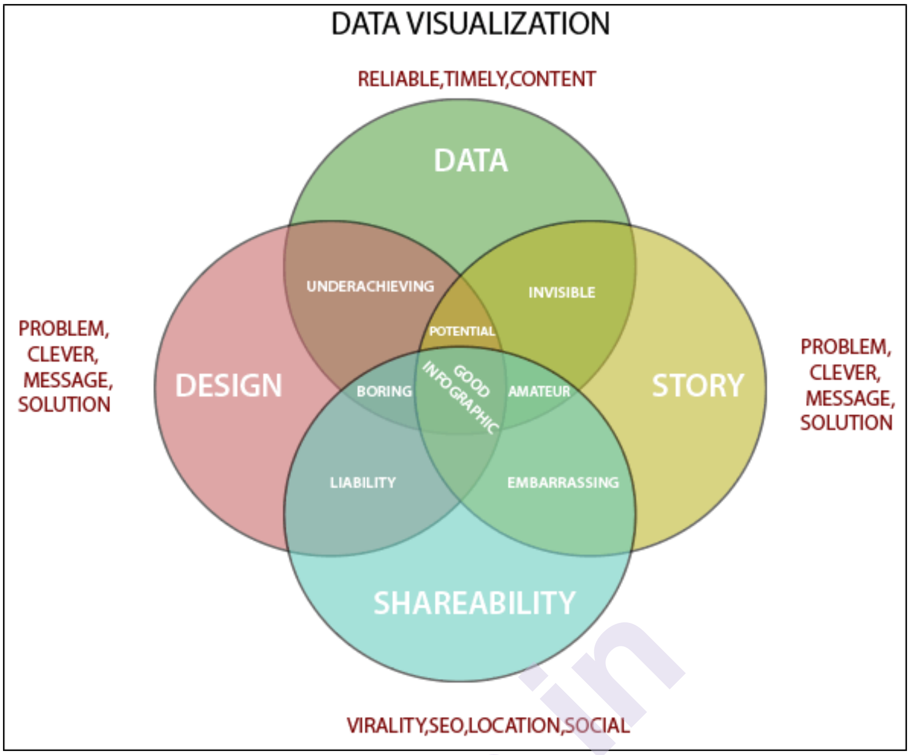
- Data visualizations are used to discover unknown facts and trends. You can see visualizations in the form of line charts to display change over time. Bar and column charts are useful for observing relationships and making comparisons. A pie chart is a great way to show parts-of-a-whole. And maps are the best way to share geographical data visually.

- Today's data visualization tools go beyond the charts and graphs used in the Microsoft Excel spreadsheet, which displays the data in more sophisticated ways such as dials and gauges, geographic maps, heat maps, pie chart, and fever chart.

What makes Data Visualization Effective?

- Effective data visualization are created by communication, data science, and design collide. Data visualizations did right key insights into complicated data sets into meaningful and natural.

- American statistician and Yale professor **Edward Tufte** believe useful data visualizations consist of ?complex ideas communicated with clarity, precision, and efficiency.



To craft an effective data visualization, you need to start with clean data that is well-sourced and complete. After the data is ready to visualize, you need to pick the right chart.

After you have decided the chart type, you need to design and customize your visualization to your liking. Simplicity is essential - you don't want to add any elements that distract from the data.

Importance of Data Visualization

Data visualization is important because of the processing of information in human brains. Using graphs and charts to visualize a large amount of the complex data sets is more comfortable in comparison to studying the spreadsheet and reports.

Data visualization is an easy and quick way to convey concepts universally. You can experiment with a different outline by making a slight adjustment.

Data visualization have some more specialties such as:

- Data visualization can identify areas that need improvement or modifications.
- Data visualization can clarify which factor influence customer behavior.
- Data visualization helps you to understand which products to place where.
- Data visualization can predict sales volumes.

Data visualization tools have been necessary for democratizing data, analytics, and making data-driven perception available to workers throughout an organization. They are easy to operate in comparison to earlier versions of BI software or traditional statistical analysis software. This guide to a rise in lines of business implementing data visualization tools on their own, without support from IT.

Why Use Data Visualization?

1. To make easier in understand and remember.
2. To discover unknown facts, outliers, and trends.
3. To visualize relationships and patterns quickly.
4. To ask a better question and make better decisions.
5. To competitive analyze.
6. To improve insights.

1.7 DIFFERENT TYPES OF DATA SOURCES

The sources of data can be classified into two types: statistical and non-statistical. Statistical sources refer to data that is gathered for some official purposes, incorporate censuses, and officially administered surveys. Non-statistical sources refer to the collection of data for other administrative purposes or for the private sector.

The following are the two sources of data:

1. **Internal sources**

- When data is collected from reports and records of the organisation itself, they are known as the internal sources.
- For example, a company publishes its annual report' on profit and loss, total sales, loans, wages, etc.

2. **External sources**

- When data is collected from sources outside the organisation, they are known as the external sources. For example, if a tour and travel company obtains information on Karnataka tourism from Karnataka Transport Corporation, it would be known as an external source of data.

Types of Data

A) Primary data

- Primary data means first-hand information collected by an investigator.
- It is collected for the first time.

- It is original and more reliable.
- For example, the population census conducted by the government of India after every ten years is primary data.

B) Secondary data

- Secondary data refers to second-hand information.
- It is not originally collected and rather obtained from already published or unpublished sources.
- For example, the address of a person taken from the telephone directory or the phone number of a company taken from Just Dial are secondary data.

1.8 SUMMARY

In this chapter we learn about Basic introduction of Data Science that is what is data and difference between data and information. Types of data and how we can collect the data, Introduction to high level programming language, what is use of Integrated Development Environment (IDE) types of IDEs,use of IDE. How to explore the data and how to analysis the data to fetch the proper data form complied data. Also how to display fetched data in proper way and user understandable format. Finally we understand the different ways and types of data sources to collect the data.

1.9 QUESTIONS

1. What is Data?
2. What are the Different kinds of data?
3. How a computer process data into information?
4. Why is exploratory data analysis important in data science?
5. Why are IDEs important?
6. What makes Data Visualization Effective?

❖❖❖❖

2

DATA MANAGEMENT

Unit Structure

2.1 Objetive

2.2 Introduction

2.3 Data Collection

2.4 Data cleaning/extraction

2.5 Data analysis

2.6 Modeling

2.7 Summary

2.8 Questions

2.1 OBJECTIVE

1) To study the data management technics.

2) To Understand the what is data collection, Need Data Collection, Different Methods of Data Collection **.**

3) To understand how to do Data cleaning/extraction.

4) To understand the Data Cleaning Process.

5) To understand Data modeling and Data modeling techniques

2.2 INTRODUCTION

Data management is the process of ingesting, storing, organizing and maintaining the data created and collected by an organization. Effective data management is a crucial piece of deploying the IT systems that run business applications and provide analytical information to help drive operational decision-making and strategic planning by corporate executives, business managers and other end users

2.3 DATA COLLECTION

Before we define what is data collection, it's essential to ask the question, "What is data?" The abridged answer is, data is various kinds of information formatted in a particular way. Therefore, data collection is the process of gathering, measuring, and analyzing accurate data from a variety of relevant sources to find answers to research problems, answer questions, evaluate outcomes, and forecast trends and probabilities.

Our society is highly dependent on data, which underscores the importance of collecting it. Accurate data collection is necessary to make informed business decisions, ensure quality assurance, and keep research integrity.

During data collection, the researchers must identify the data types, the sources of data, and what methods are being used. We will soon see that there are many different data collection methods. There is heavy reliance on data collection in research, commercial, and government fields.

Before an analyst begins collecting data, they must answer three questions first:

- What's the goal or purpose of this research?
- What kinds of data are they planning on gathering?
- What methods and procedures will be used to collect, store, and process the information?

Additionally, we can break up data into qualitative and quantitative types. Qualitative data covers descriptions such as color, size, quality, and appearance. Quantitative data, unsurprisingly, deals with numbers, such as statistics, poll numbers, percentages, etc.

Why Do We Need Data Collection?

Before a judge makes a ruling in a court case or a general creates a plan of attack, they must have as many relevant facts as possible. The best courses of action come from informed decisions, and information and data are synonymous.

The concept of data collection isn't a new one, as we'll see later, but the world has changed. There is far more data available today, and it exists in forms that were unheard of a century ago. The data collection process has had to change and grow with the times, keeping pace with technology.

Whether you're in the world of academia, trying to conduct research, or part of the commercial sector, thinking of how to promote a new product, you need data collection to help you make better choices.

Now that you know what is data collection and why we need it, let's take a look at the different methods of data collection. While the phrase "data collection" may sound all high-tech and digital, it doesn't necessarily entail things like computers, big data, and the internet. Data collection could mean a telephone survey, a mail-in comment card, or even some guy with a clipboard asking passersby some questions. But let's see if we can sort the different data collection methods into a semblance of organized categories.

What Are the Different Methods of Data Collection?

The following are seven primary methods of collecting data in business analytics.

- Surveys
- Transnational Tracking
- Interviews and Focus Groups
- Observation
- Online Tracking
- Forms
- Social Media Monitoring

Data collection breaks down into two methods. As a side note, many terms, such as techniques, methods, and types, are interchangeable and depending on who uses them. One source may call data collection techniques "methods," for instance. But whatever labels we use, the general concepts and breakdowns apply across the board whether we're talking about marketing analysis or a scientific research project.

The two methods are:

- **Primary**

As the name implies, this is original, first-hand data collected by the data researchers. This process is the initial information gathering step, performed before anyone carries out any further or related research. Primary data results are highly accurate provided the researcher collects the information. However, there's a downside, as first-hand research is potentially time-consuming and expensive.

- **Secondary**

Secondary data is second-hand data collected by other parties and already having undergone statistical analysis. This data is either information that the researcher has tasked other people to collect or information the researcher has looked up. Simply put, it's second-hand information. Although it's easier and cheaper to obtain than primary information, secondary information raises concerns regarding accuracy and authenticity. Quantitative data makes up a majority of secondary data.

2.4 DATA CLEANING/EXTRACTION

Data cleaning is the process of identifying and fixing incorrect data. It can be in incorrect format, duplicates, corrupt, inaccurate, incomplete, or irrelevant. Various fixes can be made to the data values representing incorrectness in the data. The data cleaning and validation steps undertaken for any data science project are implemented using a data pipeline. Each stage in a data pipeline consumes input and produces output. The main advantage of the data pipeline is that each step is small, self-contained, and easier to check. Some data pipeline systems also allow you to resume the pipeline from the middle, thus, saving time. In this article, we will look at eight common steps in the data cleaning process, as mentioned below.

1. Removing duplicates
2. Remove irrelevant data
3. Standardize capitalization
4. Convert data type
5. Handling outliers
6. Fix errors
7. Language Translation
8. Handle missing values

Why is Data Cleaning So Important?

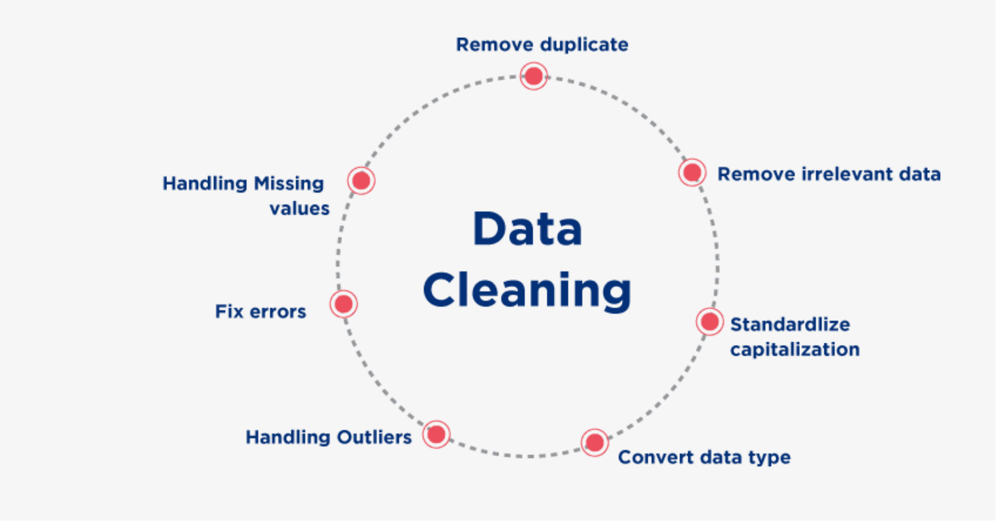
As an experienced Data Scientist, I have hardly seen any perfect data. Real-world data is noisy and contains a lot of errors. They are not in their best format. So, it becomes important that these data points need to be fixed.

It is estimated that data scientists spend between 80 to 90 percent of their time in data cleaning. Your workflow should start with data cleaning. You may likely duplicate or incorrectly classify data while working with large datasets and merging several data sources. Your algorithms and results will lose their accuracy if you have wrong or incomplete data.

For example: consider data where we have the gender column. If the data is being filled manually, then there is a chance that the data column can contain records of 'male' 'female', 'M', 'F', 'Male', 'Female', 'MALE', 'FEMALE', etc. In such cases, while we perform analysis on the columns, all these values will be considered distinct. But in reality, 'Male', 'M', 'male', and 'MALE' refer to the same information. The data cleaning step will identify such incorrect formats and fix them.

Consider another example where you are running a promotional campaign and you have collected data from different individuals. The data you collected contains the name of the individual, along with contact number, email, age, gender, etc. If you were to contact these individuals by mobile number or email, you must make sure that these are valid entries. For contact number, it should be a 10-digit numeric field and the email should follow a defined pattern. There can also be entries where you might not have either or both contact information. You would like to drop these entries since they are irrelevant and do not serve any purpose. This is again identified and fixed during data cleansing in data science before using it for our analysis or other purposes.

Data Cleaning Process [In 8 Steps]



Step 1: Remove Duplicates

- When you are working with large datasets, working across multiple data sources, or have not implemented any quality checks before adding an entry, your data will likely show duplicated values.
- These duplicated values add redundancy to your data and can make your calculations go wrong. Duplicate serial numbers of products in a dataset will give you a higher count of products than the actual numbers.
- Duplicate email IDs or mobile numbers might cause your communication to look more like spam. We take care of these duplicate records by keeping just one occurrence of any unique observation in our data.

Step 2: Remove Irrelevant Data

- Consider you are analyzing the after-sales service of a product. You get data that contains various fields like service request date, unique service request number, product serial number, product type, product purchase date, etc.
- While these fields seem to be relevant, the data may also contain other fields like attended by (name of the person who initiated the service request), location of the service center, customer contact details, etc., which might not serve our purpose if we were to analyze the expected period for a product to undergo servicing. In such cases, we remove those fields irrelevant to our scope of work. This is the column-level check we perform initially.
- Next comes the row-level checks. Assume the customer visited the service center and was asked to visit again after 3 days to collect the serviced product. In this case, let us also assume that there are two different records in the data representing the same service number.

- For the first record, the service type is ‘first visit’ and the service type is ‘pickup’ for the second record. Since both records represent the same service request number so we will likely drop one of them. For our problem statement, we need the first occurrence of the record or the ones which correspond to the service type as ‘first visit’.

- To remove irrelevant data while cleaning data for effective data science, we must understand the data and the problem statement.

Step 3: Standardize capitalization

You must ensure that the text in your data is consistent. If your capitalization is inconsistent, it could result in the creation of many false categories.

- **For example:** having column name as “Total_Sales” and “total_sales” is different (most programming languages are case-sensitive).

- To avoid confusion and maintain uniformity among the column names, we should follow a standardized way of providing the column names. The most preferred code case is the snake case or cobra case.

- Cobra case is a writing style in which the first letter of each word is written in uppercase, and each space is substituted by the underscore (_) character. While, in the snake case, the first letter of each word is written in lowercase and each space is substituted by the underscore. Therefore, the column name “Total Sales” can be written as “Total_Sales” in the cobra case and “Total Sales” in the snake case. Along with the column names, the capitalization of the data points should also be fixed.

- **For example:** while collecting names and email IDs through online forms, surveys, or other means, we can get inputs in assorted styles. We can fix them to avoid duplicate entries getting ignored. The email Id's‘myemail@hostname.com’and ‘MYEMAIL@HOSTNAME.COM’ can be interpreted as different email IDs, so it is better to make all the email ID values in the field lowercase. Similarly, for the email, we can follow the title case where all words are capitalized.

Step 4: Convert data type

When working with CSV data in python, pandas will attempt to guess the types for us; for the most part, it succeeds, but occasionally we'll need to provide a little assistance.

The most common data types that we find in the data are text, numeric, and date data types. The text data types can accept any kind of mixed values including alphabets, digits, or even special characters. A person’s name, type of product, store location, email ID, password, etc., are some examples of text data types.

Numeric data types contain integer values or decimal point numbers, also called float. Having a numeric data type column means you can perform mathematical computations like finding the minimum, maximum, average, and median, or analyzing the distribution using histogram, box plot, q-q plot, etc.

Having a numeric column as an integer column will not allow you to perform this numerical analysis. Therefore, it becomes important to convert the data types in the required formats if they are not already.

The monthly sales figures of a store, the price of a product, units of electricity consumed, etc., are examples of a numeric column. However, it is worth noting that columns like a numeric ID or phone number should not be represented as numeric columns but instead as text columns. Though they represent numeric values, operations like minimum or average values on these columns do not provide any significant information. Therefore, these columns should be represented as text columns.

The data type if not identified correctly will end up being identified as a string or text column. In such cases, we need to explicitly define the data type of the column and the date format which is mentioned in the data. The date column can be represented in different formats:

- October 02, 2023
- 02-10-2023
- 2023/10/02
- 2-Oct-2023

Step 5: Handling Outliers

An outlier is a data point in statistics that dramatically deviates from other observations. An outlier may reflect measurement variability, or it may point to an experimental error; the latter is occasionally removed from the data set.

For example: let us consider pizza prices in a region. The pizza prizes vary between INR 100 to INR 7500 in the region after surveying around 500 restaurants. But after analysis, we found that there is just one record in the dataset with the pizza price as INR 7500, while the rest of the other pizza prices are between INR 100 to INR 1500. Therefore, the observation with pizza price as INR 7500 is an outlier since it significantly deviates from the population. These outliers are usually identified using a box plot or scatter plot. These outliers result in skewed data. There are models which assume the data to follow a normal distribution, and the outliers can affect the model performance if the data is skewed thus, these outliers must be handled before the data is fed for model training. There are two common ways to deal with these outliers.

- Remove the observations that consist of outlier values.
- Apply transformations like a log, square root, box-cox, etc., to make the data values follow the normal or near-normal distribution.

You can learn about these methods and other data cleaning or wrangling skills with Bootcamp for Data Science. Develop your programming and analytical abilities as you gain confidence as a data scientist under the direction of expert professionals with six capstone projects and over 280 hours of on-demand self-paced learning.

Step 6: Fix errors

Errors in your data can lead you to miss out on the key findings. This needs to be avoided by fixing the errors that your data might have. Systems that manually input data without any provision for data checks are almost always going to contain errors. To fix them, we need to first get the data understanding. Post that, we can define logic or check the data and accordingly get the data errors fixed. Consider the following example cases.

- Removing the country code from the mobile field so that all the values are exactly 10 digits.
- Remove any unit mentioned in columns like weight, height, etc. to make it a numeric field.
- Identifying any incorrect data format like email address and then either fixing it or removing it.
- Making some validation checks like customer purchase date should be greater than the manufacturing date, the total amount should be equal to the sum of the other amounts, any punctuation or special characters found in a field that does not allow it, etc.

Step 7: Language Translation

Datasets for machine translation are frequently combined from several sources, which can result in linguistic discrepancies. Software used to evaluate data typically uses monolingual Natural Language Processing (NLP) models, which are unable to process more than one language. Therefore, you must translate everything into a single language. There are a few language transnational AI models that we can use for the task.

Step 8: Handle missing values

During **cleaning and munging in data science**, handling missing values is one of the most common tasks. The real-life data might contain missing values which need a fix before the data can be used for analysis. We can handle missing values by:

- Either removing the records that have missing values or
- Filling the missing values using some statistical technique or by gathering data understanding.

A rule of thumb is that you can drop the missing values if they make up for less than five percent of the total number of records but however it depends on the analysis, the importance of the missing values, the size of the data, and the use case we are working on.

Consider a dataset that contains certain health parameters like glucose, blood pressure, insulin, BMI, age, diabetes, etc. The goal is to create a supervised classification model that predicts if the person is likely to have diabetes or not based on the health parameters. If the data has missing values for glucose and blood pressure columns for a few individuals, there is no way we can fill these values through any technique. And assuming these two columns are of high importance in predicting the presence of diabetes in an individual then we must look to drop these observations from our records.

Consider another dataset where we have information about the laborers working on a construction site. If the gender column in this dataset has around 30 percent missing values. We cannot drop 30 percent of data observations but on further digging, we found that among the rest 70 percent of observations and 90 percent of records are male. Therefore, we can choose to fill these missing values as the male gender. By doing this, we have made an assumption, but it can be a safe assumption because the laborers working on the construction site are male dominant and even the data suggests the same. We have used a measure of central tendency called Mode, in this case. There are also other ways of filling missing values in a numerical field by using Mean or Median values based on whether the field values follow a gaussian distribution or not.

Data Cleaning Tools

1. Microsoft Excel (Popular data cleaning tool)
2. Programming languages (Python, Ruby, SQL)
3. Data Visualizations (To spot errors in your dataset)
4. Proprietary software (OpenRefine, Trifacta, etc.,)

Benefits of Data Cleaning in Data Science

Your analysis will be reliable and free of bias if you have a clean and correct data collection. We have looked at eight steps for data cleansing in data science. Let us discuss some of the benefits of cleaning data science.

- **Avoiding mistakes:** Your analysis results will be accurate and consistent if data cleansing techniques are effective.

- **Improving productivity:** Maintaining data quality and enabling more precise analytics that support the overall decision-making process are made possible by cleaning the data.
- **Avoiding unnecessary costs and errors:** Correcting faulty or mistaken data in the future is made easier by keeping track of errors and improving reporting to determine where errors originate.
- Staying organized
- Improved mapping

2.5 DATA ANALYSIS

Although many groups, organizations, and experts have different ways of approaching data analysis, most of them can be distilled into a one-size-fits-all definition. Data analysis is the process of cleaning, changing, and processing raw data and extracting actionable, relevant information that helps businesses make informed decisions. The procedure helps reduce the risks inherent in decision-making by providing useful insights and statistics, often presented in charts, images, tables, and graphs.

A simple example of data analysis can be seen whenever we make a decision in our daily lives by evaluating what has happened in the past or what will happen if we make that decision. Basically, this is the process of analyzing the past or future and making a decision based on that analysis.

It’s not uncommon to hear the term “big data” brought up in discussions about data analysis. Data analysis plays a crucial role in processing big data into useful information. Neophyte data analysts who want to dig deeper by revisiting big data fundamentals should go back to the basic question, “What is data?”

Why is Data Analysis Important?

Here is a list of reasons why data analysis is crucial to doing business today.

- Better Customer Targeting: You don’t want to waste your business’s precious time, resources, and money putting together advertising campaigns targeted at demographic groups that have little to no interest in the goods and services you offer. Data analysis helps you see where you should be focusing your advertising and marketing efforts.
- You Will Know Your Target Customers Better: Data analysis tracks how well your products and campaigns are performing within your target demographic. Through data analysis, your business can get a better idea of your target audience’s spending habits, disposable income, and most likely areas of interest. This data helps businesses set prices, determine the length of ad campaigns, and even help project the number of goods needed.

- Reduce Operational Costs: Data analysis shows you which areas in your business need more resources and money, and which areas are not producing and thus should be scaled back or eliminated outright.

- Better Problem-Solving Methods: Informed decisions are more likely to be successful decisions. Data provides businesses with information. You can see where this progression is leading. Data analysis helps businesses make the right choices and avoid costly pitfalls.

- You Get More Accurate Data: If you want to make informed decisions, you need data, but there's more to it. The data in question must be accurate. Data analysis helps businesses acquire relevant, accurate information, suitable for developing future marketing strategies, business plans, and realigning the company's vision or mission.

What Is the Data Analysis Process?

Answering the question "what is data analysis" is only the first step. Now we will look at how it's performed. The process of data analysis, or alternately, data analysis steps, involves gathering all the information, processing it, exploring the data, and using it to find patterns and other insights. The process of data analysis consists of:

- **Data Requirement Gathering**: Ask yourself why you're doing this analysis, what type of data you want to use, and what data you plan to analyze.

- **Data Collection:** Guided by your identified requirements, it's time to collect the data from your sources. Sources include case studies, surveys, interviews, questionnaires, direct observation, and focus groups. Make sure to organize the collected data for analysis.

- **Data Cleaning:** Not all of the data you collect will be useful, so it's time to clean it up. This process is where you remove white spaces, duplicate records, and basic errors. Data cleaning is mandatory before sending the information on for analysis.

- **Data Analysis:** Here is where you use data analysis software and other tools to help you interpret and understand the data and arrive at conclusions. Data analysis tools include Excel, Python, R, Looker, Rapid Miner, Chartio, Metabase, Redash, and Microsoft Power BI.

- **Data Interpretation:** Now that you have your results, you need to interpret them and come up with the best courses of action based on your findings.

- **Data Visualization:** Data visualization is a fancy way of saying, "graphically show your information in a way that people can read and understand it." You can use charts, graphs, maps, bullet points, or a host of other methods. Visualization helps you derive valuable insights by helping you compare datasets and observe relationships.

Types of Data Analysis

A half-dozen popular types of data analysis are available today, commonly employed in the worlds of technology and business. They are:

- **Diagnostic Analysis:** Diagnostic analysis answers the question, “Why did this happen?” Using insights gained from statistical analysis (more on that later!), analysts use diagnostic analysis to identify patterns in data. Ideally, the analysts find similar patterns that existed in the past, and consequently, use those solutions to resolve the present challenges hopefully.
- **Predictive Analysis:** Predictive analysis answers the question, “What is most likely to happen?” By using patterns found in older data as well as current events, analysts predict future events. While there’s no such thing as 100 percent accurate forecasting, the odds improve if the analysts have plenty of detailed information and the discipline to research it thoroughly.
- **Prescriptive Analysis**: Mix all the insights gained from the other data analysis types, and you have prescriptive analysis. Sometimes, an issue can’t be solved solely with one analysis type, and instead requires multiple insights.
- **Statistical Analysis:** Statistical analysis answers the question, “What happened?” This analysis covers data collection, analysis, modeling, interpretation, and presentation using dashboards. The statistical analysis breaks down into two sub-categories:
- **Descriptive:** Descriptive analysis works with either complete or selections of summarized numerical data. It illustrates means and deviations in continuous data and percentages and frequencies in categorical data.
- **Inferential:** Inferential analysis works with samples derived from complete data. An analyst can arrive at different conclusions from the same comprehensive data set just by choosing different samplings.
- **Text Analysis:** Also called “data mining,” text analysis uses databases and data mining tools to discover patterns residing in large datasets. It transforms raw data into useful business information. Text analysis is arguably the most straightforward and the most direct method of data analysis.

2.6 DATA MODELING

Data modeling is the process of creating a simplified diagram of a software system and the data elements it contains, using text and symbols to represent the data and how it flows. Data models provide a blueprint for designing a new database or reengineering a legacy application. Overall, data modeling helps an organization use its data effectively to meet business needs for information.

A data model can be thought of as a flowchart that illustrates data entities, their attributes and the relationships between entities. It enables data management and analytics teams to document data requirements for applications and identify errors in development plans before any code is written.

Alternatively, data models can be created through reverse-engineering efforts that extract them from existing systems. That's done to document the structure of relational databases that were built on an ad hoc basis without upfront data modeling and to define schemas for sets of raw data stored in data lakes or NoSQL databases to support specific analytics applications.

Why is data modeling done?

Data modeling is a core data management discipline. By providing a visual representation of data sets and their business context, it helps pinpoint information needs for different business processes. It then specifies the characteristics of the data elements that will be included in applications and in the database or file system structures used to process, store and manage the data.

Data modeling can also help establish common data definitions and internal data standards, often in connection with data governance programs. In addition, it plays a big role in data architecture processes that document data assets, map how data moves through IT systems and create a conceptual data management framework. Data models are a key data architecture component, along with data flow diagrams, architectural blueprints, a unified data vocabulary and other artifacts.

Traditionally, data models have been built by data modelers, data architects and other data management professionals with input from business analysts, executives and users. But data modeling is also now an important skill for data scientists and analysts involved in developing business intelligence applications and more complex data science and advanced analytics ones.

What are the different types of data models?

Data modelers use three types of models to separately represent business concepts and workflows, relevant data entities and their attributes and relationships, and technical structures for managing the data. The models typically are created in a progression as organizations plan new applications and databases. These are the different types of data models and what they include:

1. **Conceptual data model.** This is a high-level visualization of the business or analytics processes that a system will support. It maps out the kinds of data that are needed, how different business entities interrelate and associated business rules. Business executives are the main audience for conceptual data models, to help them see how a

system will work and ensure that it meets business needs. Conceptual models aren't tied to specific database or application technologies.

2. **Logical data model.** Once a conceptual data model is finished, it can be used to create a less-abstract logical one. Logical data models show how data entities are related and describe the data from a technical perspective. For example, they define data structures and provide details on attributes, keys, data types and other characteristics. The technical side of an organization uses logical models to help understand required application and database designs. But like conceptual models, they aren't connected to a particular technology platform.

3. **Physical data model.** A logical model serves as the basis for the creation of a physical data model. Physical models are specific to the database management system (DBMS) or application software that will be implemented. They define the structures that the database or a file system will use to store and manage the data. That includes tables, columns, fields, indexes, constraints, triggers and other DBMS elements. Database designers use physical data models to create designs and generate schema for databases.

Data modeling techniques

Data modeling emerged in the 1960s as databases became more widely used on mainframes and then minicomputers. It enabled organizations to bring consistency, repeatability and disciplined development to data processing and management. That's still the case, but the techniques used to create data models have evolved along with the development of new types of databases and computer systems.

These are the data modeling approaches used most widely over the years, including several that have largely been supplanted by newer techniques.

1. Hierarchical data modeling

Hierarchical data models organize data in a treelike arrangement of parent and child records. A child record can have only one parent, making this a one-to-many modeling method. The hierarchical approach originated in mainframe databases -- IBM's Information Management System (IMS) is the best-known example. Although hierarchical data models were mostly superseded by relational ones beginning in the 1980s, IMS is still available and used by many organizations. A similar hierarchical method is also used today in XML, formally known as Extensible Markup Language.

2. Network data modeling

This was also a popular data modeling option in mainframe databases that isn't used as much now. Network data models expanded on hierarchical ones by allowing child records to be connected to multiple parent records. The Conference on Data Systems Languages, a now-defunct technical standards group commonly called CODASYL, adopted a network data model specification in 1969. Because of that, the network technique is often referred to as the CODASYL model.

3. Relational data modeling

The relational data model was created as a more flexible alternative to hierarchical and network ones. First described in a 1970 technical paper by IBM researcher Edgar F. Codd, the relational model maps the relationships between data elements stored in different tables that contain sets of rows and columns. Relational modeling set the stage for the development of relational databases, and their widespread use made it the dominant data modeling technique by the mid-1990s.

4. Entity-relationship data modeling

A variation of the relational model that can also be used with other types of databases, entity-relationship (ER) models visually map entities, their attributes and the relationships between different entities. For example, the attributes of an employee data entity could include last name, first name, years employed and other relevant data. ER models provide an efficient approach for data capture and update processes, making them particularly suitable for transaction processing applications.

5. Dimensional data modeling

Dimensional data models are primarily used in data warehouses and data marts that support business intelligence applications. They consist of fact tables that contain data about transactions or other events and dimension tables that list attributes of the entities in the fact tables. For example, a fact table could detail product purchases by customers, while connected dimension tables hold data about the products and customers. Notable types of dimensional models are star schemas, which connect a fact table to different dimension tables, and snowflake schemas, which include multiple levels of dimension tables.

6. Object-oriented data modeling

As object-oriented programming advanced in the 1990s and software vendors developed object databases, object-oriented data modeling also emerged. The object-oriented approach is similar to the ER method in how it represents data, attributes and relationships, but it abstracts entities into objects. Different objects that have the same attributes and behaviors can be grouped into classes, and new classes can inherit the attributes and behaviors of existing ones. But object databases remain a niche technology for particular applications, which has limited the use of object-oriented modeling.

7. Graph data modeling

The graph data model is a more modern offshoot of network and hierarchical models. Typically paired with graph databases, it's often used to describe data sets that contain complex relationships. For example, graph data modeling is a popular approach in social networks, recommendation engines and fraud detection applications. Property graph data models are a common type -- in them, nodes that represent data entities and document their properties are connected by relationships, also known as edges or links, that define how different nodes are related to one another.

2.7 SUMMARY

In this chapter we learn about the Data management for that we want to collect data so first know about the data collection then we know about what Are the Different Methods of Data Collection. After that we learn about Data Cleaning in Data Science, need of data cleaning and various methods to clean the data. After cleaning the data we analyses the data, then we understand the process of data analyses. To do that process we understand the what is data modeling then Data modeling techniques.

2.8 QUESTIONS

1. Why Do We Need Data Collection?
2. What Are the Different Methods of Data Collection?
3. What is Data Cleaning in Data Science?
4. Why is Data Cleaning So Important?
5. What are the different types of data models?

2.9 REFERENCE

www.google.com

www.javatpoint.com

❖❖❖❖

3

DATA CURATION

Unit Structure

3.0 Objective

3.1 Data Curation

3.1.1 Introduction

3.1.2 Data Curation Life Cycle

3.2 Query Languages and Operations to Speicify and Transform Data

3.2.1 Query Languages and Operations

3.2.2 Relational Algebra

3.2.3 Joins

3.2.4 Aggregate/Group Functions

3.2,5 Structured Query Languages (Sql, Non- Procedural Query Languages)

3.3 Structured Data

3.4 Semi-Structural Data

3.4.1 XML

3.4.2 X Query

3.4.3 X Path

3.4.4 Json

3.5 Unstructured Data

3.0 OBJECTIVES

In this chapter the students will learn about:

- Data Curation
- Data Curation Life Cycle
- Query Languages
- Structured and Unstructured Data

3.1 DATA CURATION

Curation is the round-the-clock maintenance of data. Data curation refers to the data management.

It is the process of creating, organizing, and maintaining of data sets. With the help of this process, we can access and used the information or data as per the requirements.

It involves the various methods like collecting, structuring, indexing, and cataloguing data for users in an organization, any other as well.

3.1.1 Introduction to Data Curation

In this modern technology era of the big data, the data curation has become more prominent, particularly for processing of high volume of data with complex data systems.

It is an art of maintaining the valuable of data, in this process curation means a range of activities and processes done to create, manage, maintain, and validate data.

Data curation is used to determine what information is worth saving and for how much duration.

The main aim of data curation is to ensure that the data is reliably retrievable for future use or reuse.

It provides a graphical representation of high level of data with required for successful curation and preservation methods of data from initial conceptualization or receipt through the iterative curation cycle.

Data curation is an active and ongoing management of the data through its life cycle of interest and usefulness. It main role of data curation is managing and maintaining the metadata. It is an iterative process which include following three main stages

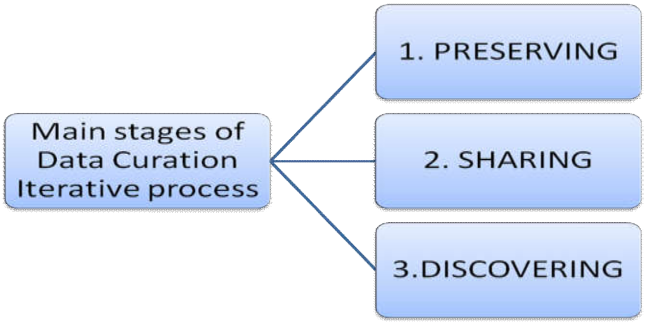


Figure 3.1Main stages of Data Curation Iterative process

1. PRESERVING

It is first step in Data Curation, in this method we are gathering data from many sources, after gathering maintaining gathered data is known as Preserving.

2. SHARING

It is second step in Data Curation, in this method we are making sure that data is available and retrievable for future purpose with authenticated user, it is known as Sharing.

3. DISCOVERING

It is third step in Data Curation, in this method we are reusing the data which we have collected with different combinations, with the help of these various combinations of data we can discover with new patterns and trends of data, it is known as Discovering.

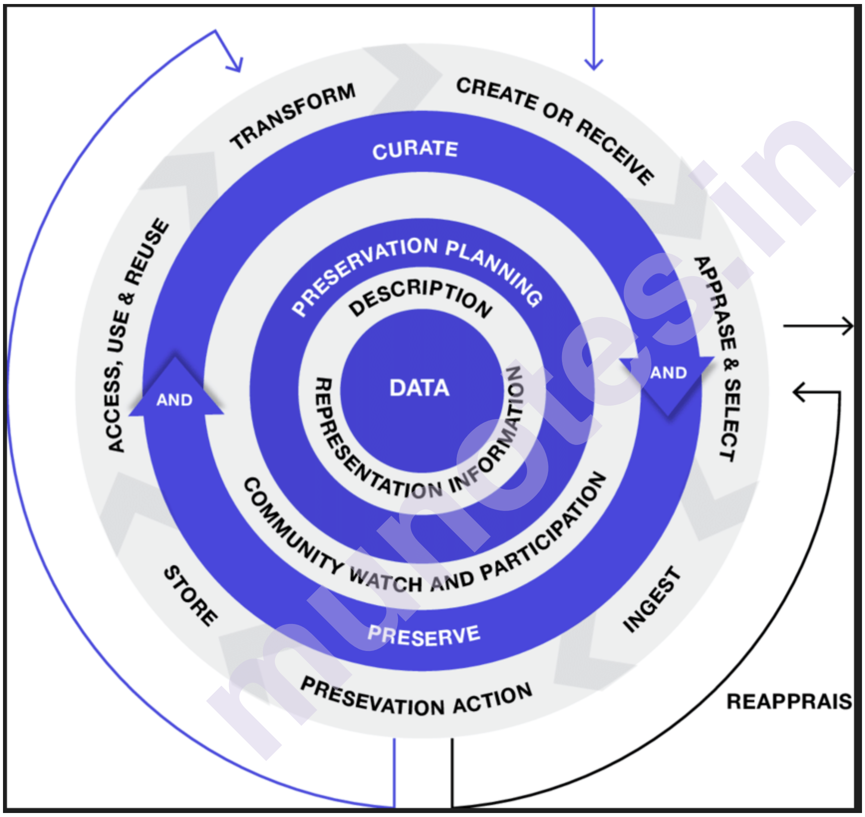


Figure 3.2 Data Curation

[Source: https://www.researchgate.net]

Life cycle of data curation represents all of stages of data from creation, distribution, and reuse. There are different components in data curation life cycle.

Components of Data Curation

1. **Data or Digital Objects or Databases:**
   It is layer 1 of the data curation life cycle model. Core components of the model in layer 1 is Data or database or digital objects. Digital objects are the type of file or the complex digital objects.

2. **Description and Representation of Information:**

   It is layer 2 of data curation life cycle model, in this layer assignment of administrative, descriptive, technical, structural and preservation of the data or database or digital objects is done depending on the standards defined.

3. **Preservation and Planning:**
   In this layer, the planning for the preservations of the digital objects, data, and databases are carried out for throughout the life cycle.

4. **Community watch and Participation:**
   In this layer, to track the activities which are in process with the help of various tools and standards.

5. **Curate and Preserve:**
   In this layer, action plans are prepared for curation and for preservation of databases or data, or digital objects throughout the lifecycle.

6. **Create and Receive:**
   In this layer, creation of data using descriptive and technical metadata, also it includes receiving of data from the various formats.

7. **Appraise and Select:**
   In this layer, evaluation and the selection of data is carried out which will help in preserving the data for long time.

8. **Ingest:**
   In this layer, data is transferred to an archive, various data centre or repository.

9. **Preservation Action:**
   In this layer, various actions are carried out with the aim long term preservations and retention of the data. Preservation actions includes for data to remain reliable, authentic, and usable.

10. **Store:**
    In this layer, data is get stored in the secured manner.

11. **Access, use and reuse:**
    In this layer, it is ensured that we can access easily to all the users, proper authentication and controlling of access are being given to the user depending on their types.

12. **Transform:**
    In this layer, it consists of very important component, where we can create new data from the original material and then transform that data into the meaningful form i.e. different format for generation of final results.

13. Conceptualize:
In this layer, the data to form with an idea or any principle which the user wants for final result generation.

14. Dispose:
In this layer, the data which is not useful for longer time or in future from the database it is also known as unwanted data. The unwanted data can be dispose to create a new space for the upcoming data.

15. Reappraise:
In this layer, the data which are not able to processed or fails the validation process of data is return back.

16. Migrate:
In this layer, data is migrated at various places depends on the need and then converted according to the new environment.

In this method, data is migrated at different places as well as it also converted according to new environment depending on the need.

3.2 QUERY LANGUAGES AND OPERATIONS TO SPEICIFY AND TRANSFORM DATA

Query languages is also known as database query languages (DQL). They are computer languages, that are used to make the queries in databases. Example of DQL is Structured Query Language (SQL).

3.2.1 Query Languages and Operations

Query language is any computer language that sends queries to the databases and it will retrieve the data from that specified databases or information system.

Query languages are special type of languages used for retrieving required information or data from the given data or database or repository. Examples XQuery, GraphQl, Xpath etc.

There are two main types of Query Languages:

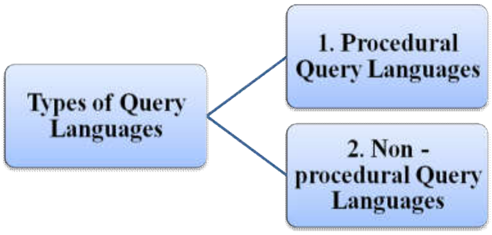

Types of Query Languages
1. Procedural Query Languages
2. Non - procedural Query Languages

Figure 3.3 Types of Query Languages

1. **Procedural Query Languages:**
   It is the formal way to access data from the database.
   When the user knows what is to be retrieved and how it is to be retrieved from the given database with the help of Query.
   Examples of Procedural Query Language: Relational Algebra.

2. **Non – procedural Query Languages:**
   It is the informal way to access data from the database.
   When the user knows what is to be retrieved but there is no idea about how to retrieved the data or information from the database.
   Examples of Non – procedural Query Languages: Structured Query Languages (SQL), Query By Example (QBE) etc.

3.2.2 RELATIONAL ALGEBRA

Relational Algebra is a procedural query language, It uses different operators to retrieve the data from the given database.
Relational Algebra uses two types of operators namely, unary and binary.
Unary is also known as single value, while binary means two values.
List of operations performed by Relational Algebra are as follows:

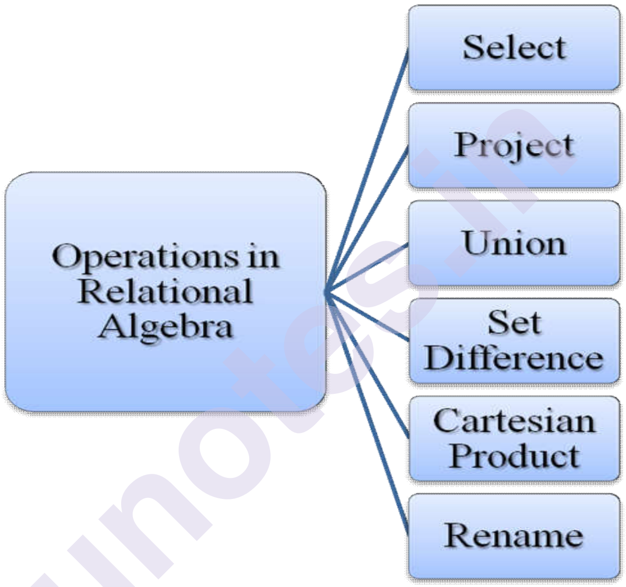


Figure 3.4 Operations in Relational Algebra

1. Select Operation (σ)

It is used to select particular rows (tuples) from the given database which satisfies the given condition. Sigma (σ) symbol is used to denote the select operation.
The operations performed by select operation is represented as $\sigma_P(r)$
Where σstands for selection predicate and r stands for relation. P is prepositional logic formula which may use connectors like and, or, and not.
It can also use the relational operators like =, <, >, >=, <=, ≠, etc.
For example:

$\sigma_{\text{subject}} = \text{“Datascience”(Books)}$

The output will select the rows from Books where name of the subject is “Datascience”.

2. Project (π)

It is used to project columns which satisfy the given condition. Pi (π) Symbol is used to denote the project operation. Where$\pi_{A_1,A_2 \dots A_n}(r)$

Where A_1, $A_{2,}$ A_n are attribute names of relation r. Automatically removal of duplicate records because as relation is a set.
For example:

$\pi_{\text{subject, author}}$ **(Books)**

The output will select the columns which projects the subject & author from Books table or relation.

3. Union (∪)

It is used to performs binary union between two tables or relations.
∪ Symbol is used to denote the union operation.

$r \cup s = \{t \mid t \in r \text{ or } t \in s\}$

Where r and s are either database relation or relations.
For union operation r and s must have number of attributes equal and same.
Attribute domains must be compatible.
Automatically elimination of duplicate records.
For example:

π_{author} **(Books)** $\cup$ π_{author} **(Articles)**

The output will projects the names of the authors who have either written a book or an article or both.

4. Set Difference (-)

It is used to remove the tuples or rows which are present in one relation but not present in another relation.
- Symbol is used to denote the union operation.

r – s

Where it finds all the rows that are present in r but not in s.

For example

π_{author} **(Books)** $-\pi_{\text{author}}$ **(Articles)**

The output will give the name of author who have written books but not articles.

5. Cartesian Product (X)

It is used to combine the information from two different relations or tables into one single relation.
X symbol is used to denote the cartesian product.

$r \, X \, s = \{ q\,t \mid q \in r \text{ and } t \in s \}$

where r and s denote the relations.
For example

$\sigma_{\text{author}} =' Balguruswamy'(BooksXArticles)$

The output will show all the books and articles written by author ‘Balguruswamy’.

6. Rename (p)

It is used when output of relational algebra queries produce relation without name, so this operation, is used to rename produced output relation.

p symbol is used to denote rename operation.

For example

$p\ (X, E)$

Where E and X are the name of the table or relation and here E is renamed by X.

3.2.3 JOINS

Join is used to combines the two different relations or tables and makes it as a single relation.

There are three types of joins as follows:

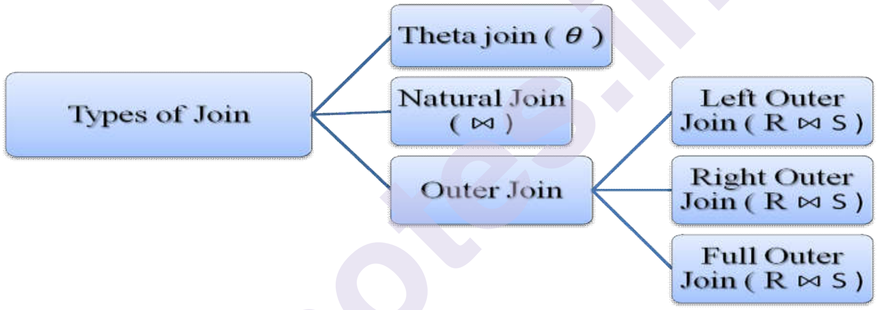

Types of Join
Theta join (θ)
Natural Join (⋈)
Outer Join
Left Outer Join (R ⋈ S)
Right Outer Join (R ⋈ S)
Full Outer Join (R ⋈ S)

Figure 3.5 Types of Join

1. Theta Join (θ)

It combines rows or tuples from two different relations if the given theta condition is satisfied. The join condition is denoted by the symbol θ .

$R_1 \bowtie \theta\, R_2$

Where R_1 and R_2 are relations having attributes (A_1, $A_2, \ldots A_n$) and (B_1, $B_2, \ldots B_n$) such that the attributes don’t have anything in common, i.e. $R_1 \cap R_2 = \phi$.

We can use all symbols of comparison operators.

2. Natural Join ($\bowtie$)

Natural join does not use any comparison operator.

It does not concatenate the way a Cartesian product does. Natural join will work only if there is at least one common attribute that exists between any two relations.

Theta and Natural join are also known as inner joins.

In case of inner join it will includes only the rows or tuples which are matching attributes and the remaining are get discarded in the resulting relation or table.

3. Outer Join

It is used to deal with the unmatched attributes of the table.

There are 3 different types of outer join

1. Left outer join
2. Right outer join
3. Full outer join

1. Left outer join

All the rows or tuples from the Left relation, R, are included in the output relation.

R ⋈ **S**

Here the left outer join will take all the rows or tuples from the table or relation left that is R are considered in the output. The rows or tuples which do not have any matching rows or tuples in S from R then that tuples of S are made NULL.

2. Right outer join

All the rows or tuples from the right relation S are included in the output relation.

R ⋈ **S**

Here the right outer join will take all the rows or tuples from the table or relation i.e S considered in the output. The rows or tuples which do not have any matching rows or tuples in R form S then that rows or tuples of R are made NULL.

3. Full outer join

All the rows or tuples from left and right relations are included in the output relation. If there are no matching found in the tuples then both relations are made NULL for non-matching attributes.

3.2.4 AGGREGATE/GROUP FUNCTIONS

In database management an aggregate function is a function in which the values of the values of multiple rows are grouped together as input on certain to form a single value of more significant meanings.

There are various aggregate functions which are as follows:

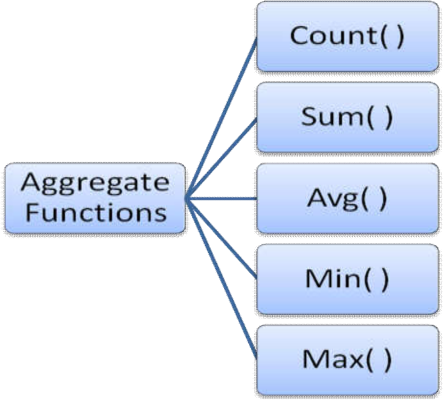


Figure 3.6: Aggregate Functions

1. Count( )

It returns the total number of records present in a given relation.

For example

Count (*)

It shows the given number of rows or tuples of in that relation.

Count (columnname)

It shows the number of non-null values over the given columnname.

Count (salary)

It shows the number of non-null values over the column salary.

2. Sum( )

It returns the sum of values for a particular attribute.

For example

Sum(salary)

It shows the sum of salary for all the employee in that relation or table.

It will display sum of salary for non - null values.

3. Avg( )

It returns the average of the values over the given attribute.

For example

Avg(salary)

It will give the average value of salary that is total or sum of all salary divided by total count and returns its value.

4. Min( )

It returns minimum value of a particular attribute.

For example

Min(salary)

It will return the minimum salary from the salary attribute.

5. Max( )

It returns maximum value of a particular attribute.

For example

Max(salary)

It will return the maximum salary from the salary attribute.

3.2,5 STRUCTURED QUERY LANGUAGES (SQL, NON-PROCEDURAL QUERY LANGUAGES)

Structure Query Language is a non-procedural Query Language, and used for retrieval, store and manipulation of databases.

There are four different types of SQL statements

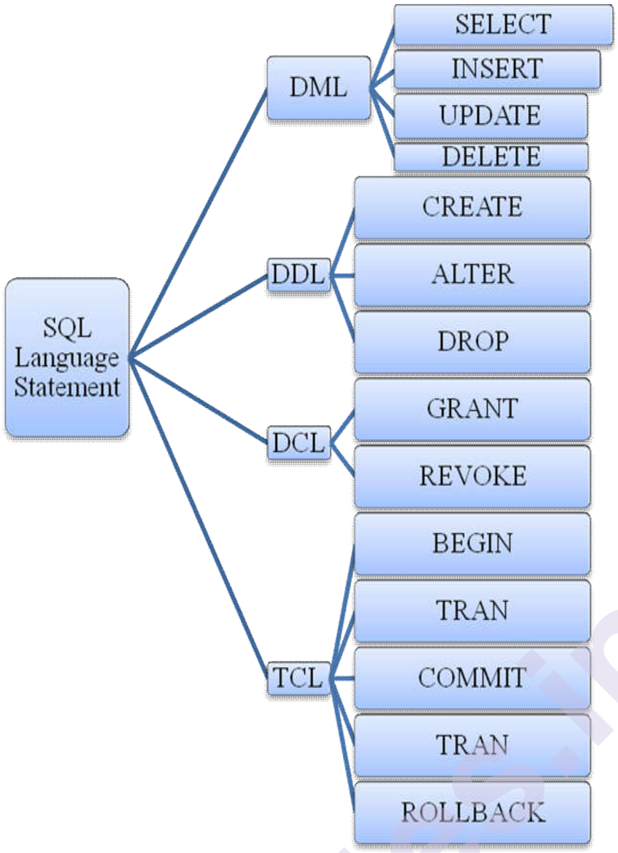


Figure 3.7 SQL Language Statement

1. DML (Data Manipulation Language)

It is used to select the records from a table, inserting, deleting and updating or modifying existing records.

Following SQL commands are used for DML

- Select: It is used to select the records from a table or schema.
  Select <column name/s> from <table name>
- Insert: It is used to insert new records in a table.
  Insert into <table name> values <col values>
- Update: It is used to update or modify existing table or records from the given database.
  Update <table name> set <col name = value>
- Delete: It is used to delete the existing records from the database.
  Delete from <table name><col name = value>

2. DDL (Data Definition Language)

It is used to modify or alter a database or table. Mainly it is used for database design and storage.

Following SQL commands are used for DDL

- Create: It is used to create new database, table or schema.
  Create Database, Create table

- Alter: It is used to alter the existing table or column description.
  Alter Database, Alter table
- Drop: It is used to delete existing table.
  Drop table

3. DCL (Data Control Language)

It is used to control the level of accessing the database.
Following SQL commands are used for DCL

- Grant: It is used to allows user to read or write on specific database.
  Grant privileges ON object TO user;
  Privileges may be Select, Insert, Update, Delete, Alter etc.
- Revoke: It is used to keeps user from read and write permission on database objects.
  Revoke privileges ON object FROM user;
  Privileges to revoke may be Select, Insert, Update, Delete, Alter etc.

4. TCL (Transaction Control Language)

Transaction Control Language is used to control and manage the transactions to maintain the integrity of the database with the help of SQL statements.

Following SQL commands are used for TCL

- Begin Transaction: It is used to opens a transaction.
- Commit Transaction: It is used to commits a transaction.
- Rollback Transaction: It is used to Rollback a transaction.

3.3 STRUCTURED DATA

Data: Data is a collection of facts and figures that can be processed to produce information.

Database: Database is a collection of data, Database is one important components for many different applications. It is used for storing the variety of data, and stored information can access very easily so that we can able to do data management as well as updation of data.

Structured Data: It is defined by a data model,which gives the data confirms to a pre-defined schema or structure. It is easily used and accessed by the users. Generally it stores in tabular form (rows and columns) with its attributes.

SQL (Structure Query Language) is used to store, manage and access data stored in databases in structured form.

Sources of Structured Data:

If the available data is highly structured like RDBMS (Oracle, Microsoft SQL Server, PostgreSQL (advanced open source) etc. It is used to hold operational data or transactional data generation and collection of day-to-day business activity. The data which comes with on line transaction processing (OLTP) system are structured data.

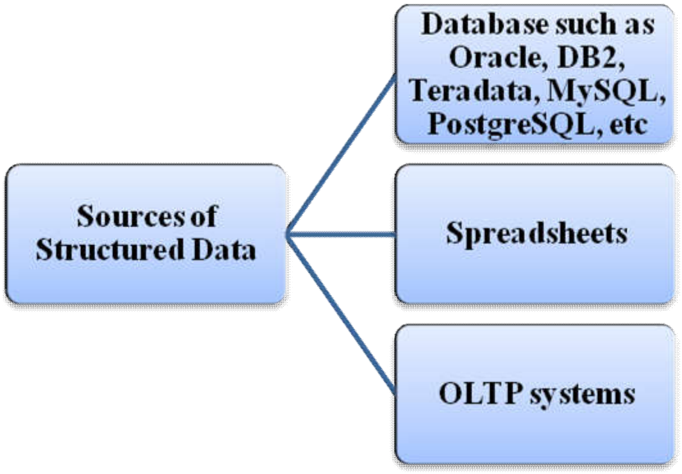

Sources of Structured Data
Database such as Oracle, DB2, Teradata, MySQL, PostgreSQL, etc
Spreadsheets
OLTP systems

Figure 3.8: Sources of Structured Data

[source: https://nitsri.ac.in/Department/Computer%20Science%20&%20Engineering/BDL2.pdf]

Uses of Structured Data

1. For Insert, update or delete these commands which are generally used in DML (Data Manipulation Language) operations for input, store, access and analysisof data.
2. It is stored in well-defined format of the database like tabular form with rows and columns.
3. It is very easy to access and manage the data from tabular form.
4. Data mining is easy as we can extract knowledge from data easily.

3.4 SEMI-STRUCTURAL DATA

Semi structured data is partially structured and partially unstructured data. The data which has not been organized into a specific database or repository, but that nevertheless has associated information, such as metadata.

Semi structured data does not have a specific format but it contains semantic tages.

Examples of semi structured data type: Email, JSON, NOSQL, XML etc.

3.4.1 XML

- As we all know that XML stands for Extensible Markup Language.
- It is a markup language and file format that helps in storing and transporting of data.
- It is designed to carry data and not just to display data as it is self-descriptive.

- It was formed from extracting the properties of SGML (Standard Generalized Markup Language).
- It supports exchanging of information between computer systems. They can be websites, databases, and any third-party applications.
- It consists of predefined rules which makes it easy to transmit data as XML files over any network.

The components of an XML file are

XML Document:

The content which are mentioned between <xml></xml> tags are called as XML Document. It is mainly at the beginning and the end of an XML code.

XML Declaration:

The content begins with come information about XML itself. It also mentions the XML version.

For example: <?xml version="1.0 encoding="UTF-8"?>

XML Elements:

The other tags you create within an XML document are called as XML Elements. It consists of the following features:

1. Text
2. Attributes
3. Other elements

For example:

```
<Fruits>

<Berries>

<type> Strawberry </type>

<type> Blueberry </type>

<type> Raspberry </type>

</Berries>

<Citrus>

<type> Oranges </type>

<type> Lemons </type>

<type> Limes </type>
```

</Citrus>

</Fruits>

Here, <Fruits></Fruits> are root elements and <Berries></Berries>&<Citrus></Citrus> are other element names.

4. XML Attributes:

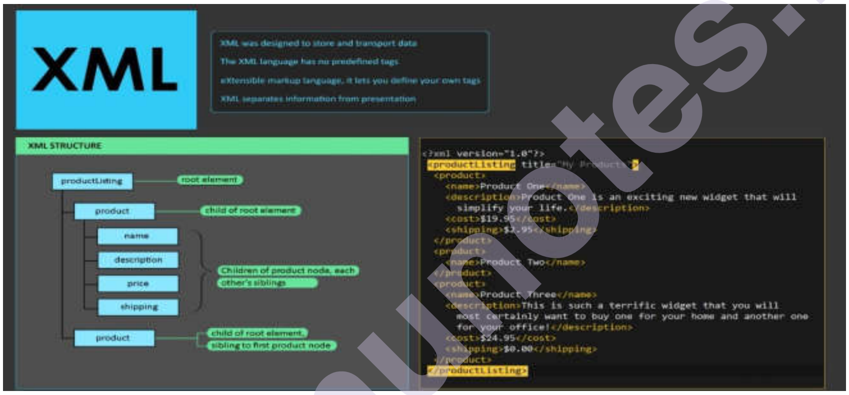
The XML elements which can have other descriptors are called as XML Attributes. One can define his/her own attribute name and attribute values within the quotation marks.

For example: <Score=”80”>

5. XML Content:

The data that is present in the XML file is called as XML content. In the given example in XML Elements, Strawberry, Blueberry, Raspberry, Oranges, Lemons and Limes are the content.

EXAMPLE:



XML was designed to store and transport data

The XML language has no predefined tags

eXtensible markup language, it lets you define your own tags

XML separates information from presentation

```
<?xml version="1.0"?>
<productListing title="My Products">
  <product>
    <name>Product One</name>
    <description>Product One is an exciting new widget that will
      simplify your life.</description>
    <cost>$19.95</cost>
    <shipping>$2.95</shipping>
  </product>
  <product>
    <name>Product Two</name>
  </product>
  <product>
    <name>Product Three</name>
    <description>This is such a terrific widget that you will
      most certainly want to buy one for your home and another one
      for your office!</description>
    <cost>$24.95</cost>
    <shipping>$0.00</shipping>
  </product>
</productListing>
```

Figure XML Document

Source: [tutorials.com]

3.4.2 XQuery

- XQuery is an abbreviation for XML Query.
- XQuery is basically considered as the language for querying XML data.
- It is built on XPath expressions.
- XQuery for XML = SQL for Databases.
- All major databases support XQuery.
- It is used for finding and extracting elements and attributes from XML documents.

- One can search web documents for relevant information and generate summary reports.
- It replaces complex Java or C++ programs with a few lines of code.

3.4.3 XPath

XPath (XML Path Language) is a query language used to navigate through an XML document and select specific elements or attributes. It is widely used in web scraping and data extraction, as well as in data science for parsing and analyzing XML data.

In data science, XPath can be used to extract information from XML files or APIs. For example, you might use XPath to extract specific data fields from an XML response returned by a web API, such as stock prices or weather data.

XPath can also be used in combination with other tools and languages commonly used in data science, such as Python and Beautiful Soup, to scrape data from websites and extract structured data for analysis. By using XPath to select specific elements and attributes, you can quickly and easily extract the data you need for analysis.

The operator in Xpath

Different types of Operators are:

- Addition (+): It does Addition Manipulation in the given field.
- Subtraction (-): It does Subtraction Manipulation in the given field.
- Multiplication (*): It does Multiplication Manipulation in the given field.
- Div (-): It does Addition Division in the given field.
- Mod: It does Modulation Manipulation in the given field.
- [/] : This Stepping Operator helps in selecting a specific node (specific path) from a root node.
- [//] : This operator Being Descendant is used to select a node directing from a root node.
- […] : This operator helps in checking a node value from the node-set.
- [|] : It is used to compute union between two node – sets by this the duplicate values are filtered out and arranged in a sorted manner.

3.4.4 JSON

- JSON (JavaScript Object Notation) is a lightweight data interchange format that is easy for humans to read and write, and easy for machines to parse and generate.

- It is used for exchanging data between web applications and servers, and can be used with many programming languages.
- JSON data is represented in key-value pairs, similar to a dictionary or a hash table.
- The key represents a string that identifies the value, and the value can be a string, number, Boolean, array, or another JSON object.
- JSON objects are enclosed in curly braces {}, and arrays are enclosed in square brackets [].
- JSON is often used in web development because it can easily be parsed by JavaScript, which is a commonly used programming language for front-end web development. JSON data can be easily converted to JavaScript objects, and vice versa, Additionally,JSON is supported by many modern web APIs, making it a popular choice for exchanging data between web applications and servers.

3.5 UNSTRUCTURED DATA

- Unstructured data is a data that is which is not organized in a pre-defined manner or does not have a pre-defined data model, thus it is not a good fit for a mainstream relational database.
- For Unstructured data, there are alternative platforms for storing and managing, it is increasingly prevalent in IT systems and is used by organizations in a variety of business intelligence and analytics applications.
- Example: Word, PDF, Text, Media logs.
- Characteristics of Unstructured Data

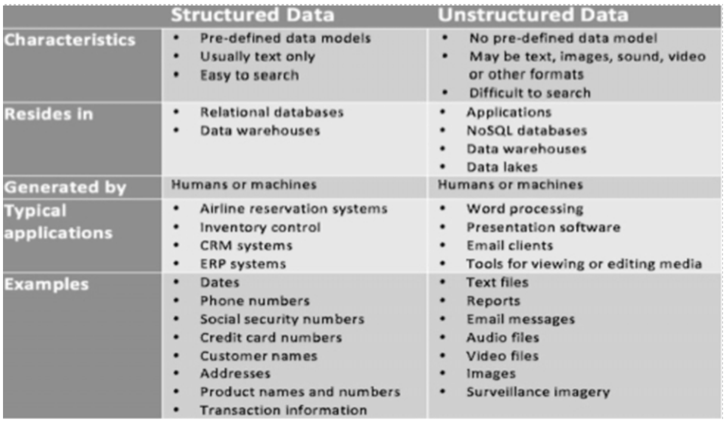
o It is not based on Schema

o It is not suitable for relational database

o 90% of unstructured data is growing today

o It includes digital media files, Word doc, pdf files,

o It is stored in NoSQL database



	Structured Data	Unstructured Data
Characteristics	• Pre-defined data models • Usually text only • Easy to search	• No pre-defined data model • May be text, images, sound, video or other formats • Difficult to search
Resides in	• Relational databases • Data warehouses	• Applications • NoSQL databases • Data warehouses • Data lakes
Generated by	Humans or machines	Humans or machines
Typical applications	• Airline reservation systems • Inventory control • CRM systems • ERP systems	• Word processing • Presentation software • Email clients • Tools for viewing or editing media
Examples	• Dates • Phone numbers • Social security numbers • Credit card numbers • Customer names • Addresses • Product names and numbers • Transaction information	• Text files • Reports • Email messages • Audio files • Video files • Images • Surveillance imagery

3.6 SUMMARY

This chapter contains the detail study of what is data, different types of data, data curation and its various steps. Query languages and the various operators of query languages, Structured, unstructured and semi structured data with example, what is aggregate and group functions, detail study of structured query languages like SQL, non-procedural query languages, structured, semi-structured and unstructured data, XML, XQuery, XPath, JSON.

3.7 UNIT END QUESTIONS

1. What is Data? Discuss different types of data.
2. What is Data Curation?
3. Explain Query languages and their operations.
4. Explain in detail structured data.
5. Explain in detail unstructured data.
6. Write a note on semi structured data with example.
7. What is XML? Explain its advantages and disadvantages.
8. Write a note on

a. XQueryb. XPath c. JSON

9. Give the difference between structured and unstructured data.

3.8 REFERENCES

1. Doing Data Science, Rachel Schutt and Cathy O'Neil, O'Reilly,2013
2. MasteringMachine Learning with R, Cory Lesmeister, PACKT Publication,2015
3. Hands-On Programming with R, Garrett Grolemund,1st Edition, 2014
4. An Introduction to Statistical Learning, James, G., Witten, D., Hastie, T., Tibshirani, R.,Springer,2015
5. http://www.icet.ac.in/Uploads/Downloads/1._MOdule_1_PDD_KQB__(1)%20(1).PDF
6. https://www.researchgate.net/figure/Diagram-of-the-digital-curation-lifecycle_fig3_340183022

❖❖❖❖

4

DATA BASE SYSTEMS

Unit Structure

4.0 Objective

4.1 Web Crawler & Web Scraping

4.1.1 Difference between Web Crawler and Web Scraping

4.2 Security and Ethical Considerations in Relation to Authenticating And Authorizing

4.2.1 ACCESS TO DATA ON REMOTE SYSTEMS

4.3 Software Development Tools

4.3.1 Version Control/Source Control

4.3.2 Github

4.4 Large Scale Data Systems

4.4.1 Paradigms of Distributed Database Storage

4.4.1.1 Homogeneous Distributed Databases

4.4.1.2 Heterogeneous Distributed Databases

4.4.2 Nosql

4.4.3 Mongodb

4.4.4 Hbase

4.5 AWS (Amazon Web Servies)

4.5.1 AWS Bsic Architecture

4.5.2 Cloud Services

4.5.3 Map Reduce

4.0 OBJECTIVES

In this chapter the students will learn about:

- Web Crawler
- Web Scraping
- Security and Ethical considerations in relation to authenticating and authorizing

To data on remote systems

- Software Development Tools
- Version control terminology and functionalities
- Github
- Large Scale Data Systems
- Distributed Database Storage
- NOSQL
- MongoDB
- HBase
- AWS
- Cloud Services
- Map Reduce

4.1 WEB CRAWLER&WEB SCRAPER

Web Crawler

Web crawler is also known as web spider, search engine bot. It takes the content from internet then downloads and indexes it.

The main aim of web crawler or bot is to learn from every webpage on the web, the content or the information can be retrieved when the user needs.

It is known as "web crawlers" as crawling is the technical term for automatically it access a website and we can obtain the information or data with the help of software programs. These bots are operated with the help of search engines. In search engine it stores the various search algorithms for collection of data by web crawlers, Search engine also provides the relevant link for the same information or content. Search engine will generate the list of webpages that contains a user types a search into Google or Bing or any other search engine like Yahoo.

Web crawler is a bot which will like in a book it will go through the various books in a disorganized library and combines a card catalog, where anyone who want to visit the library can easily and quickly fine the content or information they need.

With the help of sorted and categorized data in library's book topic-wise, then the organizer will read first the title, summary of each text book to find out what is content of particular book, if the reader needs then user can download it use it as per need.

In short book is nothing but the information on the web library which organized in a systematic manner (sort and index). User can download which one is relevant and use as per the need.

The sequence of searching in a book or web library, it will start with a certain set of known webpages and then follow the links i.e. hyperlinks from those pages to the other pages, after following the hyperlinks from those other pages to additional pages will open and user can get the information or data on it. Internet is crawled by search engine bots.

Example of web crawlers: Amazonbot (Amazon), Bingbot (Bing), Yahoo, Baiduspider (Baidu), Googlebot (Google), DuckDuckbot (DuckDuckGo) etc.

Search Indexing

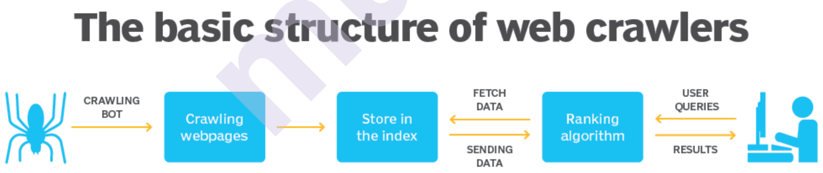
With the help of search engine on internet, it is like creating a library card catalog, where on internet it will retrieve the information or data when the user searches for it. It can also be compared to the index in the back of the book, where it will lists all the places in the book where a particular topic or phrase is typed by the user on any search engine.

The main aim of search indexing is on the web library the text that appears will search with the help of internet.

Metadata is the data that gives the details about search engines what a webpage is about. Meta means what the description will appear on search engine result pages.

Web crawlers working

It is a programming script developed by the vendors like Google. The main of these crawlers is to collect the data and send it to the Google or respective search engine. The name of the crawler comes from the commands from the programming script.



[source: techtarget.com/whatis/definition/crawler]

Crawling process: It collects data from various websites that allow crawling and indexing. Once collected data then it sends to the respective search engine like Google or user defines any other search engine.

Indexing process: After crawling process the Google or respective search engine shelves the data base on its relevance and the importance to users. With the help of hyperlink or URLs the data which are present on various sites get processed and stored in a Google or respective search engine database.

Ranking Process: After completing indexing process, user enters a query on search engine (Google), then the search engine shows the results from the stored database to the user. Sharing results with relevant keyword is also cumulate with result. Ranking of a website on a particular search engine is the key factor of relevance on search engine.

Web Scraper

Online scraping is a computerised technique for gathering copious volumes of data from websites. The majority of this data is unstructured in HTML format and is transformed into structured data in a database or spreadsheet so that it can be used in multiple applications. To collect data from websites, web scraping can be done in a variety of methods. Options include leveraging specific APIs, online services, or even writing your own code from scratch for web scraping. You may access the structured data on many huge websites, including Google, Twitter, Facebook, StackOverflow, and others, using their APIs.

Difference between Web Scraping and Web Crawling

1.	It is used for downloading information	It is used for indexing of Web pages
2.	It need not visit all the pages of website for information.	It visits each and every page, until the last line for information.
3.	A Web Scraper doesn't obey robots.txt in most of cases.	Not all web crawlers obey robots.txt.
4.	It is done on both small and large scale.	It is mostly employed in large scale.
5.	Application areas include Retail Marketing, Equity search, and Machine learning.	Used in search engines to give search results to the user.
6.	Data de-duplication is not necessarily a part of Web Scraping.	Data de-duplication is an integral part of Web Scraping.
7.	This needs crawl agent and a parser for parsing the response.	This only needs only crawl agent.
8.	ProWebScraper, Web Scraper.io are the examples	Google, Yahoo or Bing do Web Crawling

4.2 SECURITY AND ETHICAL CONSIDERATIONS IN RELATION TO AUTHENTICATING AND AUTHORIZING

Authentication And Authorization for Storage System

Security is an important parameter for any data storage system. Various security attacks that can be faced in any system can be:

1. Password guessing attack

2. Replay attack.

3. Man-in-the-middle attack

6. Phishing attack

4. Masquerade attack

5. Shoulder surfing attack.

6. Insider attack

Authentication and Authorization are two major processes used for security of data on the emote system.

1. Denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks overwhelms a system's resources so that it cannot respond to service requests.

2. Man-in-the-middle (MitM) attack : A MitM attack occurs when a hacker inserts itself between communications of a client and a server. It causes misuse of data.

3. Phishing and spear phishing attacks : Phishing attack is the act of sending emails that appear to be from trusted sources in-order to get personal information or influencing users to do something.

4. Drive-by attack: Drive-by download attacks is used to spread malware, Hackers look for insecure web malware an email message or directly onto the computer of someone who visits the site, or it might re-direct the victim to a site controlled and plant a malicious script into HTTP or PHP code on one of the pages. This script might install by the hackers, Drive-by downloads can happen when visiting a website or viewing a pop-up window.

5. Password attack: Because passwords are used to authenticate users to an information system, obtaining passwords is a common and effective attack approach. Access to a person's password can be obtained looking around the person's desk, "sniffing" the connection to the network to acquire unencrypt passwords, using social engineering, gaining access to a password database or outright guessing.

6. SQL injection attack: SQL injection has become a common issue with database-driven websites. It occurs when a malefactor executes a SQL query to the database via the input data from the client to server.

7. Cross-site scripting (XSS) attack: XSS attacks use third-party web resources to run scripts in the victim web browser or scriptable application.

8. Malware attack: Malicious software can be described as unwanted software that is installed in your system without your consent.

- Examples of data security technologies include data backups, data masking and data erasure.

- A key data security technology measure is encryption, where digital data, software/hardware, and h drives are encrypted so that it is made unreadable to unauthorized users and hackers.

- One of the most commonly used methods for data security is the use of authentication and authorization.

- With authentication, users must provide a password, code, biometric data, or some other form of data verify identity of user before we grant access to a system or data.

4.2.1 ACCESS TO DATA ON REMOTE SYSTEMS

There are various major process used for security of data on remote system.

Authentication

It is a process for confirming the identity of the user. The basic way of providing authentication is through username and password, but many a time this approach fails due to hackers or attackers if some hacker will be able to crack the password and username than even the hacker will able to use the system.

Authentication is part of a three-step process for gaining access to digital resources:

1. Identification—Who are you?
2. Authentication—Prove it.
3. Authorization—Do you have permission?

Identification requires a user ID like a username. But without identity authentication, there’s no way to know if that username actually belongs to them. That’s where authentication comes in—pairing the username with a password or other verifying credentials.

The most common method of authentication is a unique login and password, but as cybersecurity threats have increased in recent years, most organizations use and recommend additional authentication factors for layered security.

Authorization

It follows the authentication step which means that once the authentication of a particular user is done the next step is authorization which is to check what rights are given to that user.

-During the process of authentication policies are made which define the authorities of that user.

Various algorithms used for authentication and authorization are:

1. RSA algorithm.

2. AES algorithm and MD5 hashing algorithm.

3. OTP password algorithm.

4. Data encryption standard algorithm.

5. Rijndael encryption algorithm.

4.3 SOFTWARE DEVELOPMENT TOOLS

Software development tools plays a crucial role in data science workflows, especially as projects become more complex and involve larger amounts of data.

Here are some of the most commonly used software development tools in data science:

4.3.1 VERSION CONTROL/SOURCE CONTROL

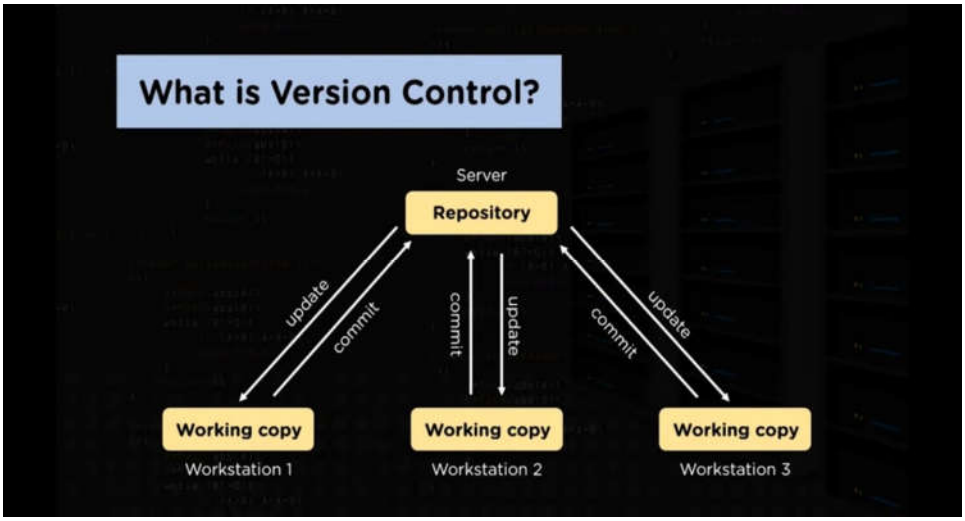
Version control systems (VCS)

Basically, Version control also known as source control is the practice of tracking and managing changes to software code.

Any multinational companies may face several problems like collaboration among employees, storing several versions of files being made, and data backing up. All these challenges are essential to be resolved for a company to be successful. This is when a Version Control System comes into the picture.

In other words, it allows developers to track changes to code and collaborate on projects with other team members.

Git is the most commonly used VCS in data science, and platforms like **GitHub and GitLab** provide hosting services for Git repositories.



Let's try to understand the process with the help of this diagram

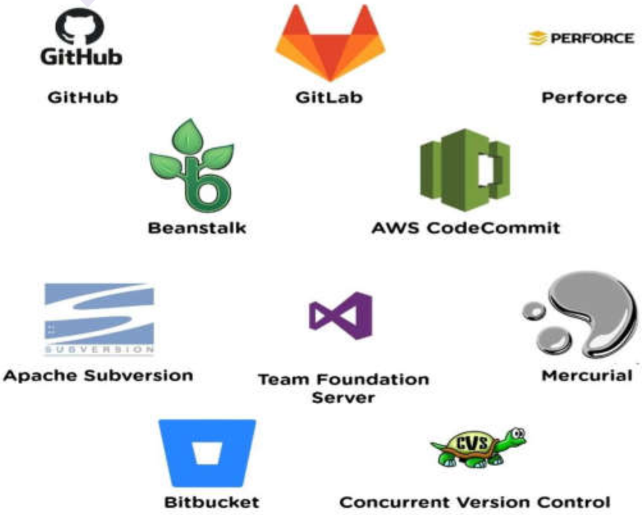
Source: (https://youtu.be/Yc8sCSeMhi4)

There are 3 workstations or three different developers at three other locations, and there's one repository acting as a server. The work stations are using that repository either for the process of committing or updating the tasks.

There may be a large number of workstations using a single server repository. Each workstation will have its working copy, and all these workstations will be saving their source codes into a particular server repository.

This makes it easy for any developer to access the task being done using the repository. If any specific developer's system breaks down, then the work won't stop, as there will be a copy of the source code in the central repository.

Finally, let's have a look at some of the best Version Control Systems in the market.



- **Integrated Development Environments (IDEs)**

IDEs are software applications that provide a comprehensive environment for coding, debugging, and testing code.

Popular IDEs for data science include

- PyCharm
- Spyder
- Jupyter Notebook
- **Package managers**

Package managers make it easy to install, update, and manage software libraries and dependencies.

Popular package managers for Python include

- pip
- conda
- **Data analysis and visualization tools**

Data analysis and visualization tools help data scientists to explore, clean, and visualize data.

Popular tools include

- Pandas
- NumPy
- Matplotlib
- **Automated testing tools**

Automated testing tools help to ensure the quality and correctness of code.

Popular tools include

- pytest
- unittest
- **Deployment tools**

Deployment tools are used to deploy models and applications to production environments.

Popular deployment tools include

- Docker
- Kubernetes

In addition to these tools, data scientists may also use cloud platforms such as AWS, Google Cloud, and Microsoft Azure for data storage, computing resources, and machine learning services.

4.3.2 GITHUB

Github is an Internet hosting service for software development and version control using Git. It provides the distributed version control of Git plus access control, bug tracking, software feature requests, task management, continuous integration, and wikis for every project.

Projects on GitHub.com can be accessed and managed using the standard Git command-line interface; all standard Git commands work with it. GitHub.com also allows users to browse public repositories on the site. Multiple desktop clients and Git plugins are also available. The site provides social networking-like functions such as feeds, followers, wikis is newest. Anyone can browse and download public repositories but only registered users can contribute content to repositories.

Git

GIT full form is “Global Information Tracker,” Git is a DevOps tool used for source code management. It is a free and open-source version control system used to handle small to very large projects efficiently. Git is used to tracking changes in the source code, enabling multiple developers to work together on non-linear development. While Git is a tool that's used to manage multiple versions of source code edits that are then transferred to files in a Git repository, GitHub serves as a location for uploading copies of a Git repository.

Need of Github

It's used for storing, tracking, and collaborating on software projects. It makes it easy for developers to share code files and collaborate with fellow developers on open-source projects. GitHub also serves as a social networking site where developers can openly network, collaborate, and pitch their work Languages used in Git : Core languages for GitHub features include C, C++, C#, Go, Java, JavaScript, PHP, Python, Ruby, Scala, and TypeScript

4.4 LARGE SCALE DATA SYSTEMS

To store the large data normal databases cannot be used and hence databases like NoSQL, MongoDB and HBase etc are good option for large scale data systems. Large scale systems do not always have centralized data storage. Distributed database approach is widely used in many applications.

4.4.1 PARADIGMS OF DISTRIBUTED DATABASE STORAGE

A distributed database is basically a database that is not limited to one system, it is spread over different sites, i.e, on multiple computers or over a network of computers. A distributed database system is located on various sites that don’t share physical components. This may be required when a particular database needs to be accessed by various users globally. It needs to be managed such that for the users it looks like one single database.

Distributed databases are capable of modular development, meaning that systems can be expanded by adding new computers and local data to the new site and connecting them to the distributed system without interruption. When failures occur in centralized databases, the system comes to a complete stop. When a component fails in distributed database systems, however, the system will continue to function at reduced performance until the error is fixed. Data is physically stored across multiple sites. Data in each site can be managed by a DBMS independent of the other sites. The processors in the sites are connected via a network. They do not have any multiprocessor configuration. A distributed database is not a loosely connected file system.

A distributed database incorporates transaction processing, but it is not synonymous with a transaction processing system.

Distributed database systems are mainly classified as homogenous and heterogeneous database.

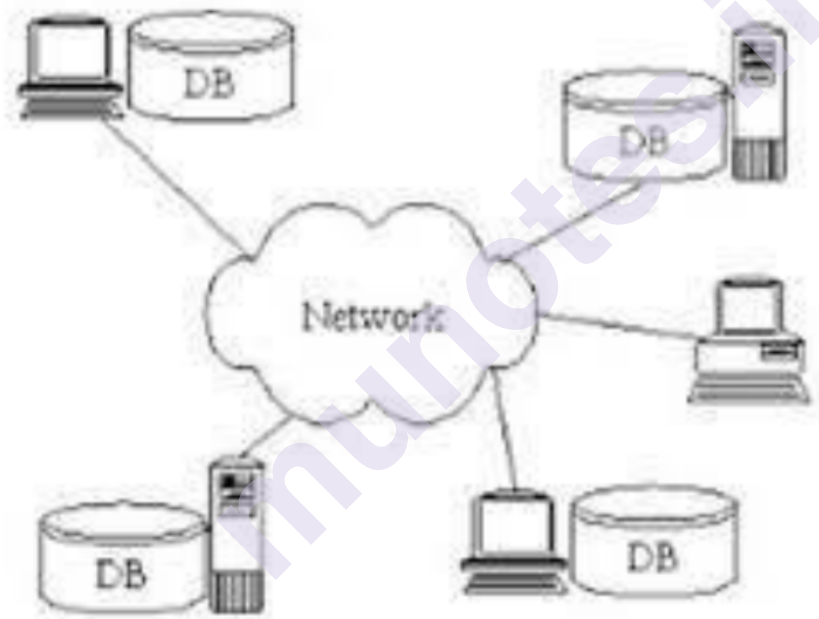


Figure : Distributed Database

4.4.1.1 HOMOGENEOUS DISTRIBUTED DATABASES

Homogeneous Distributed Databases:

- Homogeneous Distributed Databases are the systems in which on all the sites identical DBMS and OS are used.
- Homogeneous Distributed Databases have identical software's and here every site know what is happening at other site and where it is located.
- Homogeneous Distributed Databases are further classified as autonomous and non-autonomous.
- In autonomous database each site is independent for processing only the integration is done using some controlling application.

- In non-autonomous databases, data is distributed across the various nodes or sites and one node manages all the other nodes as if like client server model.

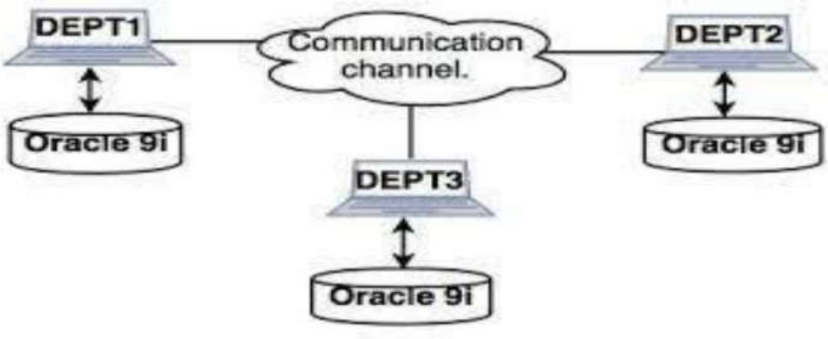


Figure : Homogeneous distributed system

4.4.1.2 HETEROGENEOUS DISTRIBUTED DATABASES

Heterogeneous Distributed Databases:

- In Heterogeneous Distributed Databases every site has different database, OS and different software's.
- In such system querying is complex as the environment and all other tools are different.
- Heterogeneous Distributed Database is further classified as Federated and Un-federated databases.
- In Federated database system every site is independent of each other and hence it acts as a single database system individually.
- In Un-federated database system there is a single central coordinator module through which all the sites communicates.

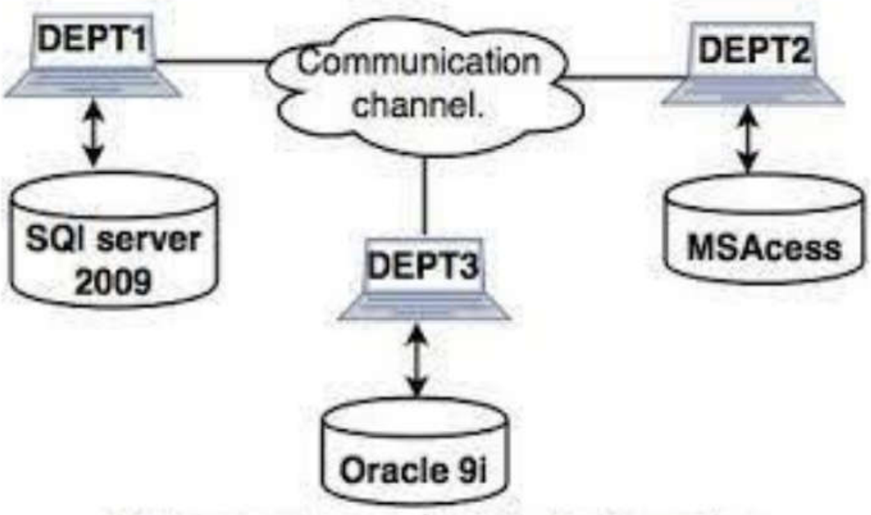


Figure : Heterogeneous database system

4.4.2 NOSQL

NoSQL is a broad term that refers to non-relational databases that don't use the traditional SQL querying language. NoSQL databases come in different types, such as key-value stores, document-oriented databases, graph databases, and column-family stores.

1. **Schema flexibility:** NoSQL databases allow for flexible schema designs that can be easily adapted to changing data requirements. This allows for more agile development and easier scaling of the database.
2. **Horizontal scalability:** NoSQL databases are designed to scale horizontally, meaning that new nodes can be added to the cluster to increase storage and processing capacity. This allows for virtually unlimited scalability and high availability.
3. **High performance:** NoSQL databases are designed for high performance and low latency, which makes them well-suited for handling real-time data processing and analytics workloads.
4. **Replication and availability:** Most NoSQL databases provide built-in replication and fault-tolerance features that ensure high availability and data durability.
5. **Distributed architecture:** NoSQL databases are typically designed as distributed systems, which allows them to distribute data across multiple nodes in the cluster. This enables efficient handling of large volumes of data and high performance at scale.
6. **No fixed schema:** Unlike traditional relational databases, NoSQL databases do not require a predefined schema. This means that you can add new fields or attributes to the data on the fly, without having to modify the entire database schema.

4.4.3 MONGODB

MongoDB is a document-oriented NoSQL database that stores data in the form of JSON-like documents.

- **Automatic sharing:** MongoDB can automatically split data across multiple servers, allowing it to handle large volumes of data and scale horizontally.
- **Indexing:** MongoDB supports indexes on any field, including fields within nested documents and arrays.
- **Rich query language:** MongoDB supports a rich query language that includes filtering, sorting, and aggregation.
- **Dynamic schema:** MongoDB's flexible schema allows you to add new fields or change existing ones without affecting the existing data.
- **Replication:** MongoDB supports replica sets, which provide automatic failover and data redundancy.

4.4.4 HBASE

HBase is also a NoSQL database, but it is a column-oriented database built on top of Hadoop. HBase is an excellent choice for applications that require random read/write access to large amounts of data.

- **Built on Hadoop:** HBase is built on top of Hadoop, allowing it to leverage Hadoop's distributed file system (HDFS) for storage and MapReduce for processing.
- **Strong consistency:** HBase provides strong consistency guarantees, ensuring that all reads and writes are seen by all nodes in the cluster.
- **Scalability:** HBase can scale to handle petabytes of data and billions of rows.
- **Data compression:** HBase provides data compression options, reducing the amount of storage required for large datasets.
- **Transactions:** HBase supports multi-row transactions, allowing for complex operations to be executed atomically.

4.5 AWS (AMAZON WEB SERVICES)

Amazon Web Service

Amazon Web Service is a platform that offers scalable, easy to use, flexible and cost-effective cloud computing platforms, API's and solutions to individuals, businesses and companies. AWS provides different IT resources available on demand. It also provides different services such as infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS). Amazon's first cloud computing service was S3(Simple Storage Service) released in 2006, March. Using AWS, instead of building large-scale infrastructures and storage; companies can opt for Amazon Cloud Services where they can get all the infrastructure they could ever need.

4.5.1 AWS BASIC ARCHITECTURE

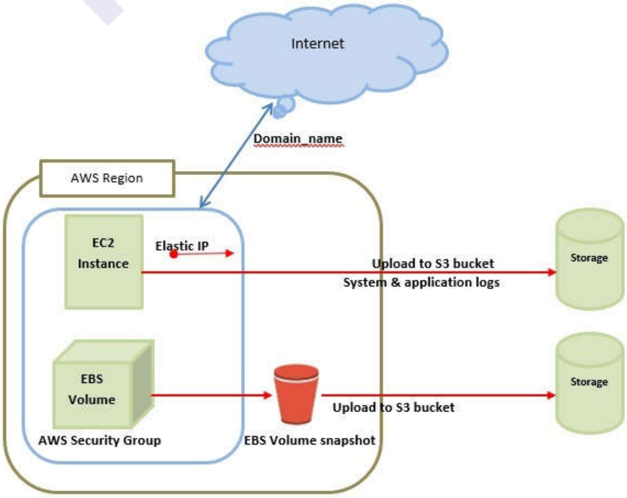


Figure: Amazon web services architecture

Source [Tutorialspoint.com]

This includes EC2, S3, EBS Volume.

EC2 which stands for Elastic Compute Cloud. EC2 provides the opportunity to the users to choose a virtual machine as per their requirement. It gives freedom to the user to choose between a variety of storage, configurations, services, etc.

S3 stands for Simple Storage Service, using which online backup and archiving of data becomes easier. It allows the users to store and retrieve various types of data using API calls. It doesn't contain any computing element.

EBS also known as Elastic Block Store, provides persistent block storage volumes which are to be used in instances created by EC2. It has the ability to replicate itself for maintaining its availability throughout.

The Important Cloud Services according to various categories that are provided by AWS are given below :

1. Compute

Amazon EC2: Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It allows organisations to obtain and configure virtual compute capacity in the cloud. Amazon EC2 is an example of Infrastructure as a Service(IaaS).

AWS Elastic Beanstalk: AWS Elastic Beanstalk is a Platform as a Service that facilitates quick deployment of your applications by providing all the application services that you need for your application. Elastic Beanstalk supports a large range of platforms like Node js, Java, PHP, Python, and Ruby.

2. Networking

Amazon Route 53: Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. It is designed to give developers and businesses an extremely reliable and cost-effective way to route end users to Internet applications by translating human-readable names, such as www.geeksforgeeks.com, into the numeric IP addresses that computers use to connect to each other. Amazon Route 53 is fully compliant with IPv6 as well.

3. Storage

Amazon S3 (Simple Storage Service): Amazon Simple Storage Service (Amazon S3) is object storage with a simple web service interface to store and retrieve any amount of data from anywhere on the web. You can use Amazon S3 as primary storage for cloud-native applications as a target for backup and recovery and disaster recovery.

Amazon Glacier: Amazon Glacier is a secure, durable, and extremely low-cost storage service for data archiving and long-term backup. Data stored in Amazon Glacier takes several hours to retrieve, which is why it's ideal for archiving.

4. Databases

Amazon RDS (Relational Database Service): Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. You can find Amazon RDS is also available on several database instance types – optimised for memory, performance, or I/O.

Benefits of AWS

High Availability.

Parallel Processing.

Security.

Low Latency.

Fault Tolerance and disaster recovery.

Cost effective.

4.5.2 CLOUD SERVICES

What is Cloud Computing?

Cloud computing is a technology that allows use storage, applications, and services over the internet, without having to own or manage their own infrastructure. Instead of having to purchase, configure, and maintain hardware and software, users can simply rent resources from cloud service providers. There are several types of cloud computing models, including:

1. **Infrastructure as a Service (IaaS):** Provides users with access to computing resources such as servers, storage, and networking.
2. **Platform as a Service (PaaS):** Provides users with access to a platform for developing, testing, and deploying applications.
3. **Software as a Service (SaaS):** Provides users with access to software applications over the internet. Cloud computing has revolutionized the way computing resources without having to invest in expensive infrastructure. As a result, cloud computing has become an essential technology for businesses of all sizes

Advantages of Cloud Computing There are numerous advantages of cloud computing, some of which include:

1. **Scalability:** Cloud computing offers the ability to quickly scale up or down computing resources based on demand. This can be done automatically or manually, making it easier for businesses to manage spikes in usage or traffic.

2. **Cost-effectiveness:** Cloud computing reduces the need for businesses to invest in expensive hardware and infrastructure. Instead, they can rent computing resources from cloud service providers on a pay-as-you-go basis. This allows businesses to only pay for what they use, reducing overall costs.

3. **Accessibility:** With cloud computing, users can access computing resources from anywhere with an internet connection. This means that employees can work remotely and collaborate on projects from different locations.

4. **Security:** Cloud service providers offer robust security measures, including encryption, firewalls, and access controls to protect data and applications. Additionally, cloud providers often employ dedicated security teams to monitor and respond to potential security threats.

5. **Reliability:** Cloud service providers offer high levels of uptime and availability, ensuring that resources are always accessible when needed. Additionally, cloud providers typically have redundant infrastructure in place to ensure that services remain available even if there is an outage in one location.

6. **Flexibility:** Cloud computing allows businesses to experiment with new applications and services without having to commit to long-term investments. This means that businesses can test new ideas quickly and easily, without worrying about the cost of hardware or infrastructure. Overall, cloud computing offers numerous advantages for businesses of all sizes, making it a popular choice for many organizations.

Disadvantages of Cloud Computing While cloud computing offers many benefits, there are also some potential disadvantages to consider. Some of these include:

1. **Dependence on the Internet:** Cloud computing requires a reliable internet connection in order to access computing resources. If the internet is slow or unavailable, this can impact the ability to access critical resources.

2. **Security concerns:** While cloud providers often offer robust security measures, there is still the potential for security breaches and data theft. Additionally, if a cloud provider experiences a security breach, this can impact multiple customers at once.

3. **Limited control:** When using cloud computing, businesses may have limited control over their computing resources. This can make it more difficult to customize applications or infrastructure to meet specific needs.

4. **Downtime:** While cloud providers offer high levels of uptime, there is still the potential for downtime due to outages, maintenance, or other issues. This can impact productivity and cause disruption to business operations.

5. **Cost:** While cloud computing can be cost-effective in some cases, it can also be expensive if usage levels are high or if resources are not managed effectively. Additionally, cloud providers may raise prices or change their pricing models over time, which can impact the cost of using cloud computing.

6. **Data privacy and compliance:** Businesses may face challenges in ensuring that data stored in the cloud is compliant with regulatory requirements. Additionally, some organizations may have concerns about data privacy and how data is used by cloud providers.

Overall, while cloud computing offers many benefits, it is important for businesses to carefully consider the potential drawbacks and risks before deciding to adopt cloud computing.

Need for Cloud Computing

The need for cloud computing arises from the fact that businesses require access to powerful computing resources to support their operations, but investing in and maintaining their own infrastructure can be costly and time-consuming. Cloud computing allows businesses to access computing resources over the internet, rather than having to build and maintain their own infrastructure. Overall, cloud computing addresses many of the key needs that businesses face, including scalability, flexibility, cost-efficiency, reliability, security, and innovation. As a result, cloud computing has become an essential technology for many businesses.

4.5.3 MAP REDUCE

MapReduce

MapReduce is a programming model and data processing framework used for parallel computing of large datasets on clusters of commodity hardware. It was originally developed by Google to process large amounts of data in a distributed environment. The MapReduce programming model allows developers to write simple and scalable code for processing large datasets. It also provides fault tolerance and automatic parallelization, making it well-suited for big data applications

The basic idea of MapReduce is to split a large data processing task into smaller sub-tasks and execute them in parallel across a cluster of computers. The sub-tasks are divided into two phases:

1. **Map phase:** In this phase, the input data is divided into smaller chunks and processed by individual nodes in the cluster. Each node processes its assigned data and produces key-value pairs as output.

2. **Reduce phase:** In this phase, the output of the map phase is collected and processed to produce the final result. The reduce phase takes in the key-value pairs produced by the map phase and applies a reduce function to aggregate the values with the same key.

MapReduce is widely used in big data processing because it allows developers to write code that can be easily parallelized and distributed across a large number of machines. This enables the processing of very large datasets that would otherwise be difficult or impossible to handle with traditional data processing techniques.

Uses of MapReduce

- **Scalability:** MapReduce is highly scalable as it allows parallel processing of large datasets across a large number of machines. This makes it ideal for handling big data workloads.
- **Fault tolerance:** MapReduce is designed to handle failures in the cluster. If a machine fails, the MapReduce framework automatically reassigns the tasks to other machines, ensuring the job is completed without any data loss or errors.
- **Flexibility:** MapReduce is flexible as it can be used with a variety of data storage systems, including Hadoop Distributed File System (HDFS), Amazon S3, and Google Cloud Storage.
- **Cost-effective:** MapReduce is cost-effective as it uses commodity hardware to process data. This makes it an affordable solution for handling big data workloads.
- **Efficient:** MapReduce is efficient because it performs data processing operations in parallel, which reduces the overall processing time. This makes it possible to process large datasets in a reasonable amount of time.

Overall, MapReduce is used to process and analyze large volumes of data in a distributed computing environment, making it an essential tool for handling big data workloads

4.6 SUMMARY

This chapter gives brief introduction of Database System. After studying this chapter, you will learn about the concept of web crawling and web scraping, what are the various security and ethical considerations in relation to authentication and authorization, what are the software development tools, what is version control, GitHub, detail study of large-scale systems with the different types namely homogeneous distributed system and heterogeneous distributed system, NoSQL, HBase, Mongo DB, what is AWS, cloud services and MapReduce.

4.7 UNIT END QUESTION

Q.1) What is Web Crawling and Web Scraping?

Q.2) Give the difference between Web Crawling and Web Scraping.

Q.3) Explain in briefly about Authentication and Authorization for Storage System.

Q.4) Elaborate the concept of version control.

Q.5) Write a note on GitHub.

Q.6) What is Distributed Database Storage? Explain with its types.

Q.7) Write briefly about large scale data systems.

Q.8) Give the difference between Homogeneous and Hetrogeneous Data storage.

Q.9) Write a note on

a) NoSQL b)HBase c) Mong DB

Q.10) Explain AWS in detail.

Q.11) What is MapReduce? Explain with its architecture.

Q.12) What is cloud computing? Explain with its types.

4.8 REFERENCES

1. Doing Data Science, Rachel Schutt and Cathy O'Neil, O'Reilly,2013
2. Mastering Machine Learning with R, Cory Lesmeister, PACKT Publication,2015
3. Hands-On Programming with R, Garrett Grolemund,1st Edition, 2014
4. An Introduction to Statistical Learning, James, G., Witten, D., Hastie, T., Tibshirani, R.,Springer,2015
5. https://www.cloudflare.com/learning/bots/what-is-a-web-crawler/#:~:text=A%20web%20crawler%2C%20or%20spider,appear%20in%20search%20engine%20results
6. (https://capsicummediaworks.com/web-crawler-guide/)

❖❖❖❖

5

INTRODUCTION TO MODEL SELECTION

Unit Structure

5.0 Objectives

5.1 Introduction

5.2 Regularization

5.2.1 Regularization techniques

5.3 Bias/variance tradeoff

5.3.1 What is Bias?

5.3.2 What is Variance?

5.3.3 Bias-Variance Tradeoff

5.4 Parsimony Model

5.4.1 How to choose a Parsimonious Model

5.4.1.1 AIC

5.4.1.2 BIC

5.4.1.3 MDL

5.5 Cross validation

5.5.1 Methods used for Cross-Validation

5.5.2 Limitations of Cross-Validation

5.5.3Applications of Cross-Validation

5.6 Summary

5.7 List of References

5.8 Unit End Exercises

5.0 OBJECTIVES

- To understand the factors that needs to be considered while selecting a model
- To get familiar with the regularization techniques and bias-variance tradeoffs
- To understand the parsimony and cross-validation techniques

5.1 INTRODUCTION

The process of choosing a single machine learning model out of a group of potential candidates for a training dataset is known as model selection.

Model selection is a procedure that can be used to compare models of the same type that have been set up with various model hyperparameters (e.g., different kernels in an SVM)and models of other types (such as logistic regression, SVM, KNN, etc).

A "good enough" model is particular to your project and might mean many different things, including:

- A design that satisfies the demands and limitations of project stakeholders
- A model that, given the time and resources at hand, is suitably skilled
- A skilled model as opposed to unsophisticated models
- A model that performs well compared to other models that have been examined
- A model that is proficient in terms of current technology

5.2 REGULARIZATION

The term "regularization" describes methods for calibrating machine learning models to reduce the adjusted loss function and avoid overfitting or underfitting.

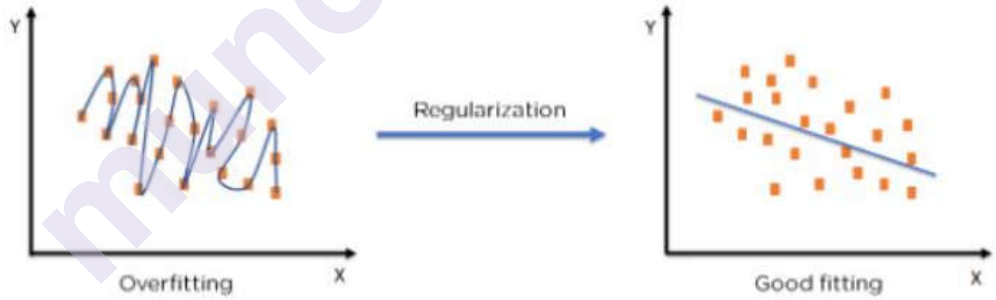


Figure 1: Regularization on an over-fitted model

We can properly fit our machine learning model on a particular test set using regularization, which lowers the mistakes in the test set.

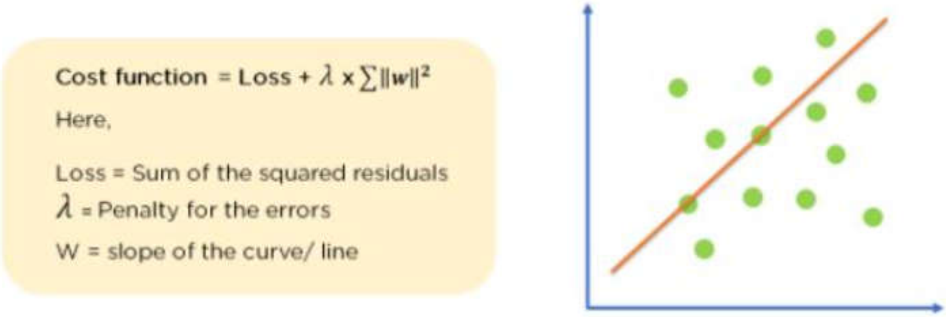
5.2.1 Regularization techniques

There are two main types of regularization techniques: Ridge Regularization and Lasso Regularization.

1] Ridge Regularization

It is also referred to as Ridge Regression and modifies over- or under-fitted models by applying a penalty equal to the sum of the squares of the coefficient magnitude.

As a result, coefficients are produced and the mathematical function that represents our machine learning model is minimized. The coefficients' magnitudes are squared and summed. Ridge Regression applies regularization by reducing the number of coefficients. The cost function of ridge regression is shown in the function below:



Cost function = Loss + λ x $\sum ||w||^2$

Here,

Loss = Sum of the squared residuals

λ = Penalty for the errors

W = slope of the curve/ line

Figure 2:Cost Function of Ridge Regression

The penalty term is represented by Lambda in the cost function. We can control the punishment term by varying the values of the penalty function. The magnitude of the coefficients decreases as the penalty increases. The settings are trimmed. As a result, it serves to prevent multicollinearity and, through coefficient shrinkage, lower the model's complexity.

Have a look at the graph below, which shows linear regression:

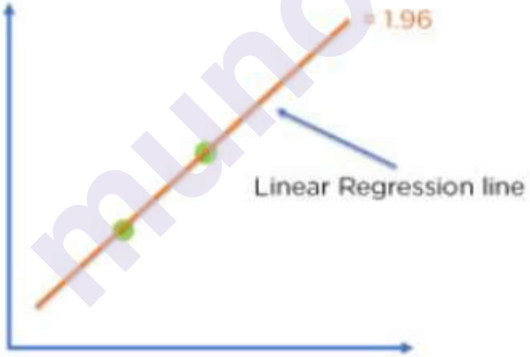


Figure 3: Linear regression model

Cost function = Loss + λ x∑ ||w ||^2

For Linear Regression line, let’s consider two points that are on the line,

Loss = 0 (considering the two points on the line)

λ= 1

w = 1.4

Then, Cost function = 0 + 1 x 1.42

= 1.96

For Ridge Regression, let's assume,

Loss = 0.32 + 0.22 = 0.13

λ = 1

w = 0.7

Then, Cost function = 0.13 + 1 x 0.72

= 0.62

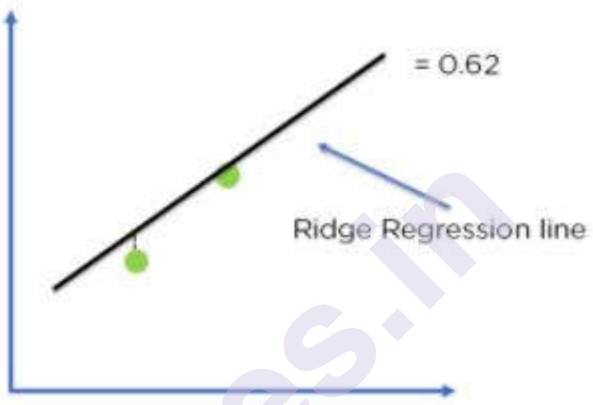


Figure 4: Ridge regression model

Comparing the two models, with all data points, we can see that the Ridge regression line fits the model more accurately than the linear regression line.

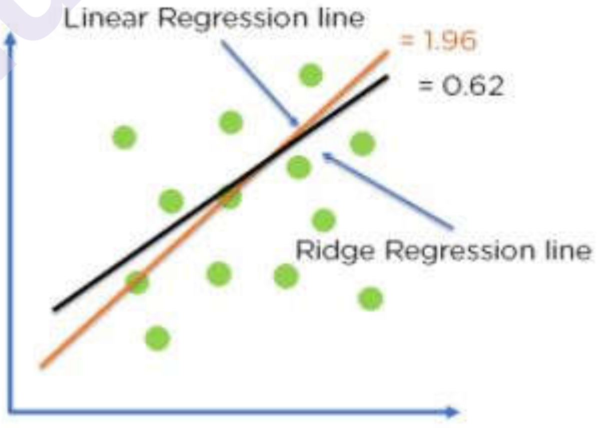
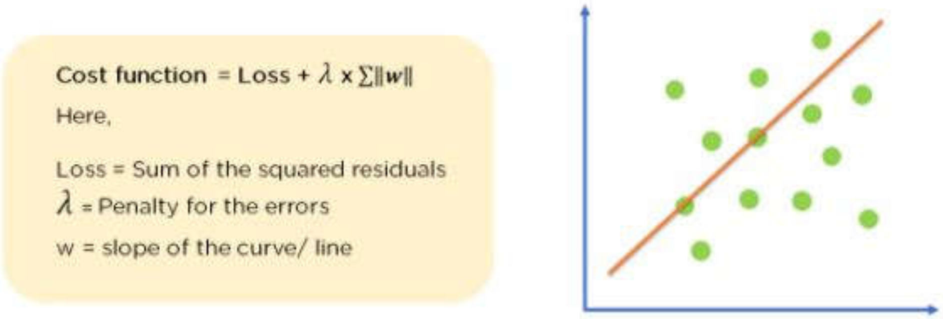


Figure 5: Optimization of model fit using Ridge Regression

2]Lasso Regularization

By imposing a penalty equal to the total of the absolute values of the coefficients, it alters the models that are either overfitted or underfitted.

Lasso regression likewise attempts coefficient minimization, but it uses the actual coefficient values rather than squaring the magnitudes of the coefficients. As a result of the occurrence of negative coefficients, the coefficient sum can also be 0. Think about the Lasso regression cost function:



Cost function = Loss + λ x $\sum\|w\|$

Here,

Loss = Sum of the squared residuals

λ = Penalty for the errors

w = slope of the curve/ line

Figure 6:Cost function for Lasso Regression

We can control the coefficient values by controlling the penalty terms, just like we did in Ridge Regression. Again, consider a Linear Regression model:

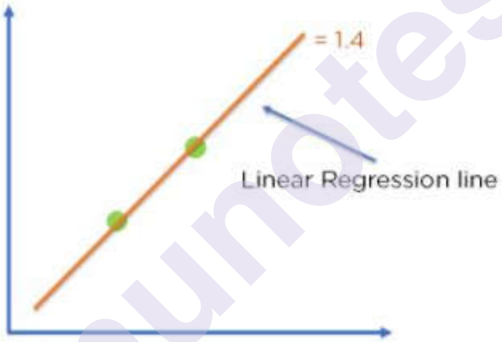


Figure 7: Linear Regression Model

Cost function = Loss + λ x ∑ ||w ||

For Linear Regression line, let’s assume,

Loss = 0 (considering the two points on the line)

λ = 1

w = 1.4

Then, Cost function = 0 + 1 x 1.4

= 1.4

For Ridge Regression, let’s assume,

Loss = 0.32 + 0.12 = 0.1

λ = 1

w = 0.7

Then, Cost function = 0.1 + 1 x 0.7

= 0.8

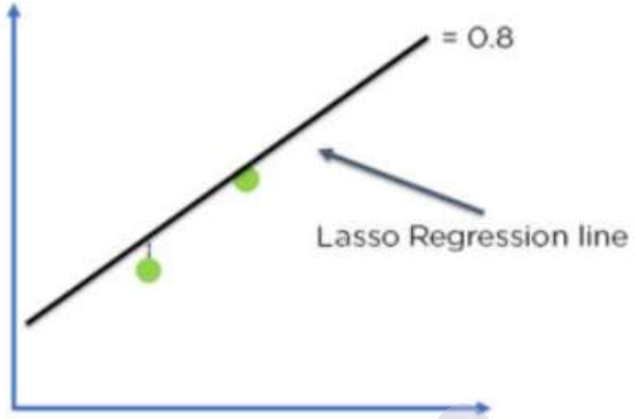


Figure 8: Lasso regression

Comparing the two models, with all data points, we can see that the Lasso regression line fits the model more accurately than the linear regression line.

5.3 BIAS/VARIANCE TRADEOFF

5.3.1 What is Bias?

Our model will examine our data and look for patterns before making predictions. We can draw conclusions about specific cases in our data using these patterns. Following training, our model picks up on these trends and uses them to predict the test set.

The bias is the discrepancy between our actual values and the predictions. In order for our model to be able to forecast new data, it must make some basic assumptions about our data.

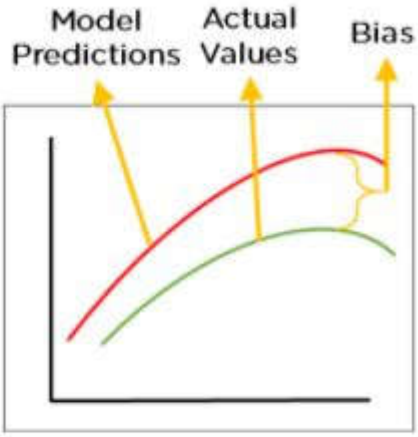


Figure 9: Bias

When the bias is significant, our model's assumptions are too simplistic, and the model is unable to capture the crucial aspects of our data. As a result, our model cannot successfully analyze the testing data because it has not been able to recognize patterns in the training data. If so, our model is unable to operate on fresh data and cannot be put into use.

Underfitting refers to the situation where the model is unable to recognize patterns in our training set and hence fails for both seen and unseen data.

Figure following provides an illustration of underfitting. The line of best fit is a straight line that doesn't go through any of the data points, as can be seen by the model's lack of pattern detection in our data. The model was unable to effectively train on the provided data and is also unable to predict fresh data.

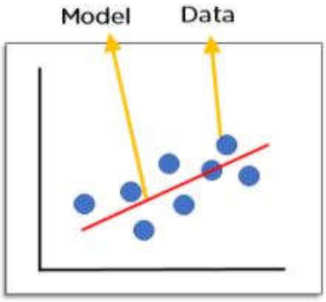


Figure 10: Underfitting

5.3.2 What is Variance?

Bias' complete opposite is variation. Our model is given a limited number of opportunities to "view" the data during training in order to look for patterns. Insufficient time spent working with the data will result in bias because patterns won't be discovered. On the other hand, if our model is given access to the data too frequently, it will only be able to train very well for that data. The majority of patterns in the data will be captured, but it will also learn from extraneous data or noise that is there.

Variance can be thought of as the model's sensitivity to changes in the data. From noise, our model might learn. This will lead our model to value unimportant features highly.

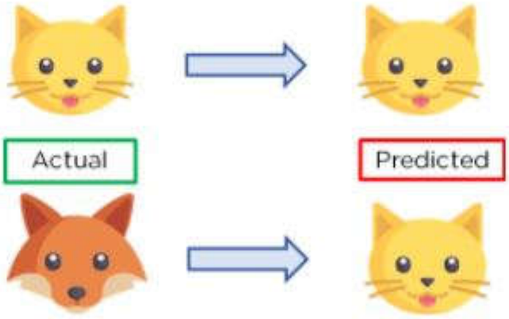


Figure 11: Example of Variance

We can see from the above picture how effectively our model has learned from the training data, which has trained it to recognize cats. Nevertheless, given fresh information, like the image of a fox, our model predicts it to be a cat because that is what it has learnt to do. When variance is high, our model will catch all the properties of the data provided, including the noise, will adjust to the data, and predict it extremely well. However, when given new data, it is unable to forecast since it is too specific to training data.

As a result, while our model will perform admirably on test data and achieve high accuracy, it will underperform on brand-new, unforeseen data. The model won't be able to forecast new data very effectively because it could not have the exact same characteristics. Overfitting is the term for this.

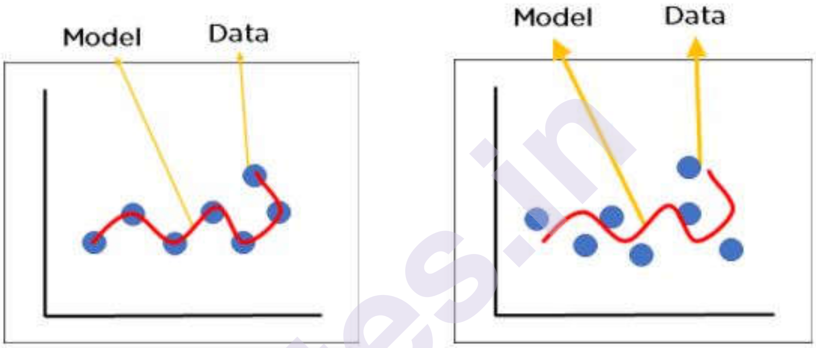


Figure 12:Over-fitted model where we see model performance on, a) training data b) new data

5.3.3 Bias-Variance Tradeoff

We need to strike the ideal balance between bias and variance for every model. This only makes sure that we record the key patterns in our model and ignore the noise it generates. The term for this is bias-variance tradeoff. It aids in optimizing and maintaining the lowest feasible level of inaccuracy in our model.

A model that has been optimized will be sensitive to the patterns in our data while also being able to generalize to new data. This should have a modest bias and variance to avoid overfitting and underfitting.

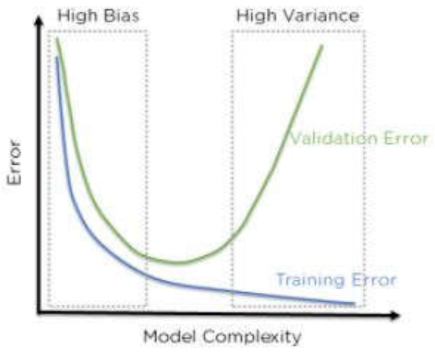


Figure 13:Error in Training and Testing with high Bias and Variance

We can observe from the above figure that when bias is large, the error in both the training set and the test set is also high. When the variance is high, the model performs well on the testing set and the error is low, but the error on the training set is significant. We can see that there is a zone in the middle where the bias and variance are perfectly balanced and the error in both the training and testing set is minimal.

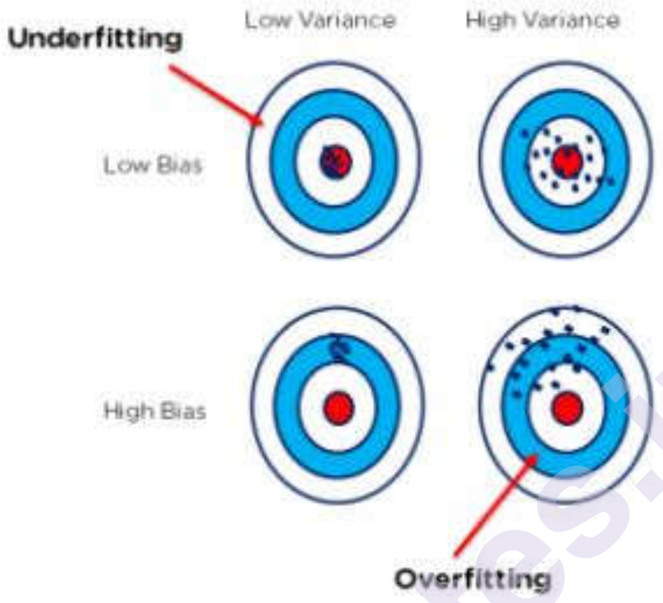


Figure 14:Bull's Eye Graph for Bias and Variance

The bull's eye graphs up top clarifies the bias and variance tradeoff. When the data is concentrated in the center, or at the target, the fit is optimal. We can see that the error in our model grows as we move farther and farther from the center. The ideal model has little bias and little volatility.

5.4 PARSIMONY MODEL

A parsimonious model is one that employs the fewest number of explanatory variables necessary to reach the desired level of goodness of fit.

The theory behind this kind of model is Occam's Razor, often known as the "Principle of Parsimony," which holds that the best explanation is usually the simplest one.

In terms of statistics, a model with fewer parameters but a reasonable degree of goodness of fit ought to be chosen over one with many parameters but a marginally higher level of goodness of fit.

This is due to two factors:

1. It is simpler to interpret and comprehend parsimonious models. Less complicated models are simpler to comprehend and justify.

2. Parsimonious models typically exhibit higher forecasting accuracy. When used on fresh data, models with fewer parameters typically perform better.

To demonstrate these concepts, think about the following two situations.

Example 1: Parsimonious Models=Simple Interpretation

Assume that we wish to create a model to forecast house prices using a set of real estate-related explanatory factors. Take into account the two models below, together with their modified R-squared:

Model 1:

- **Equation:** House price = 8,830 + 81*(sq. ft.)
- **Adjusted R^2:** 0.7734

Model 2:

- **Equation:** House price = 8,921 + 77*(sq. ft.) + 7*(sq. ft.)^2 – 9*(age) + 600*(rooms) + 38*(baths)
- **Adjusted R^2:** 0.7823

While the second model includes five explanatory factors and an only marginally higher adjusted R^2, the first model only has one explanatory variable with an adjusted R^2 of.7734.

According to the parsimony principle, we would like to choose the first model since it is simpler to comprehend and explain and has nearly the same ability to explain the fluctuation in home prices as the other models.

For instance, according to the first model, an increase of one unit in a home's square footage corresponds to a $81 rise in the average price of a home. That is easy to comprehend and explain.

The coefficient estimates in the second example, however, are significantly more challenging to understand. For instance, if the house's square footage, age, and number of bathrooms are all kept same, adding one room to the home will boost the price by an average of $600. That is considerably more difficult to comprehend and justify.

Example 2: Parsimonious Models = Better Predictions

Because they are less prone to overfit the initial dataset, parsimonious models also have a tendency to make predictions on fresh datasets that are more correct.

In comparison to models with fewer parameters, models with more parameters typically result in tighter fits and higher R2 values. Sadly, if a model has too many parameters, the model may end up fitting the data's noise or "randomness" rather than the actual underlying link between the explanatory and response variables.

This indicates that compared to a simpler model with fewer parameters, a very complicated model with many parameters is likely to perform poorly on a fresh dataset that it hasn't seen before.

5.4.1 How to choose a Parsimonious Model

Model selection could be the subject of an entire course, but ultimately, picking a parsimonious model comes down to picking one that performs well based on some criteria.

Typical metrics that assess a model's effectiveness on a training dataset and the quantity of its parameters include:

5.4.1.1Akaike Information Criterion (AIC)

The AIC of a model can be calculated as:

$AIC = -2/n * LL + 2 * k/n$

where:

- n: Number of observations in the training dataset.
- LL: Log-likelihood of the model on the training dataset.
- k: Number of parameters in the model.

The AIC of each model may be determined using this procedure, and the model with the lowest AIC value will be chosen as the best model.

When compared to the next method, BIC, this strategy tends to prefer more intricate models.

5.4.1.2Bayesian Information Criterion (BIC)

The BIC of a model can be calculated as:

$BIC = -2 * LL + \log(n) * k$

where:

- n: Number of observations in the training dataset.
- log: The natural logarithm (with base e)
- LL: Log-likelihood of the model on the training dataset.
- k: Number of parameters in the model.

Using this method, you can calculate the BIC of each model and then select the model with the lowest BIC value as the best model.

5.4.1.3. Minimum Description Length (MDL)

The MDL is a way of evaluating models that comes from the field of information theory. It can be calculated as:

MDL = L(h) + L(D | h)

where:

- h: The model.
- D: Predictions made by the model.
- L(h): Number of bits required to represent the model.
- L(D | h): Number of bits required to represent the predictions from the model on the training data.

Using this method, you can calculate the MDL of each model and then select the model with the lowest MDL value as the best model.

Depending on the type of problem you're working on, one of these methods – AIC, BIC, or MDL – may be preferred over the others as a way of selecting a parsimonious model.

5.5 CROSS VALIDATION

By training the model on a subset of the input data and testing it on a subset of the input data that hasn't been used before, you may validate the model's effectiveness. It is also a method for determining how well a statistical model generalizes to a different dataset.

Testing the model's stability is a necessary step in machine learning (ML). This indicates that we cannot fit our model to the training dataset alone. We set aside a specific sample of the datasetone that wasn't included in the training datasetfor this use. After that, before deployment, we test our model on that sample, and the entire procedure is referred to as cross-validation. It differs from the typical train-test split in this way.

Hence, the fundamental cross-validation stages are:

- As a validation set, set aside a portion of the dataset.
- Use the training dataset to provide the model with training.
- Use the validation set to assess the model's performance right now. Do the next step if the model works well on the validation set; otherwise, look for problems.

5.5.1 Methods used for Cross-Validation

There are some common methods that are used for cross-validation. These methods are given below:

1] Validation Set Approach

With the validation set approach, we separate our input dataset into a training set and a test or validation set. 50% of the dataset is divided between the two subsets.

Nevertheless, it has a significant drawback in that we are only using 50% of the dataset to train our model, which means that the model can fail to capture crucial dataset information. It frequently produces the underfitted model as well.

2] Cross-validation using Leave-P-out

The training data in this method excludes the p datasets. This means that if the original input dataset has a total of n datapoints, n-p datapoints will be utilised as the training dataset, and p datapoints will be used as the validation set. For each sample, the entire procedure is carried out once, and the average error is determined to determine the model's efficacy.

This method has a drawback in that it can be computationally challenging for large p.

3] Leave one out cross-validation

This technique is similar to leave-p-out cross-validation, but we need to exclude one dataset from training instead of p. It indicates that in this method, only one data point is set aside for each learning set, while the remaining dataset is used to train the model. Each datapoint in this process is repeated again. Hence, for n samples, n distinct training sets and n test sets are obtained. It has these characteristics:

- As all the data points are used, the bias is minimal in this method.
- Because the process is run n times, the execution time is long.
- Due to the iterative nature of this method, measuring the model's efficacy against a single data point is highly variable.

4] K-Fold Cross-Validation

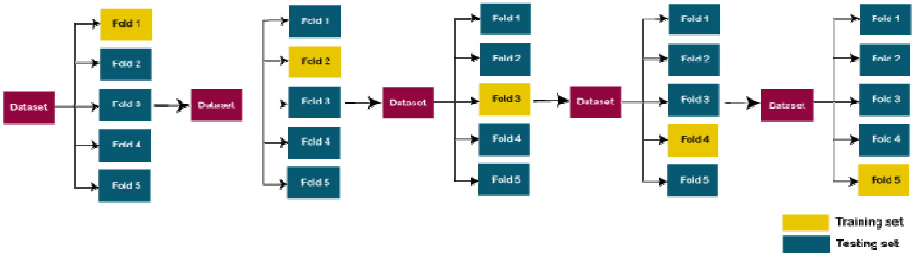
K-fold cross-validation approach divides the input dataset into K groups of samples of equal sizes. These samples are called folds. For each learning set, the prediction function uses k-1 folds, and the rest of the folds are used for the test set. This approach is a very popular CV approach because it is easy to understand, and the output is less biased than other methods.

The steps for k-fold cross-validation are:

- Split the input dataset into K groups
- For each group:
- Take one group as the reserve or test data set.
- Use remaining groups as the training dataset
- Fit the model on the training set and evaluate the performance of the model using the test set.

Let's take an example of 5-folds cross-validation. So, the dataset is grouped into 5 folds. On 1st iteration, the first fold is reserved for test the model, and rest are used to train the model. On 2nd iteration, the second fold is used to test the model, and rest are used to train the model. This process will continue until each fold is not used for the test fold.

Consider the below diagram:



5] Cross-validation with a stratified k-fold

With a few minor adjustments, this method is identical to k-fold cross-validation. The stratification principle underlies this method, which involves rearranging the data to make sure that each fold or group is a good representation of the entire dataset. It is one of the finest strategies for addressing bias and variation.

It can be understood by utilizing the example of housing costs, where some homes may have substantially higher prices than others. A stratified k-fold cross-validation technique is helpful to handle such circumstances.

6] Holdout Method

This methodology for cross-validation is the simplest one available. With this technique, we must take out a portion of the training data and train the remaining dataset on it to obtain the prediction results.

The inaccuracy that results from this procedure provides insight into how effectively our model will work with the unidentified dataset. Although this method is straightforward to use, it still struggles with large volatility and occasionally yields inaccurate findings.

5.5.2 Limitations of Cross-Validation

There are some limitations of the cross-validation technique, which are given below:

- It delivers the best results under the best circumstances. Nonetheless, the contradictory data could lead to a dramatic outcome. When there is uncertainty over the type of data used in machine learning, this is one of the major drawbacks of cross-validation.

- Because data in predictive modelling changes over time, there may be variations between the training set and validation sets. For instance, if we develop a stock market value prediction model and the data is trained on the stock prices from the previous five years, but the realistic future stock prices for the following five years could be very different, it is challenging to predict the correct output in such circumstances.

5.5.3Applications of Cross-Validation

- This method can be used to evaluate how well various predictive modelling approaches work.
- It has a lot of potential for medical study.
- Although data scientists are already using it in the field of medical statistics, it can also be utilised for meta-analysis.

5.6 SUMMARY

We have studied the following points from this chapter:

- Model selection is a procedure used by statisticians to examine the relative merits of various statistical models and ascertain which one best fits the observed data.
- The process of selecting a model from a large pool of potential models for a predictive modelling issue is known as model selection.
- Beyond model performance, there may be several competing considerations to consider throughout the model selection process, including complexity, maintainability, and resource availability.
- Probabilistic measurements and resampling procedures are the two primary groups of model selection strategies.

5.7 LIST OF REFERENCES

1. Doing Data Science, Rachel Schutt and Cathy O'Neil, O'Reilly,2013.
2. Mastering Machine Learning with R, Cory Lesmeister, PACKT Publication,2015.
3. Hands-On Programming with R, Garrett Grolemund,1st Edition, 2014.
4. An Introduction to Statistical Learning, James, G., Witten, D., Hastie, T., Tibshirani, R.,Springer,2015.

5.8 UNIT END EXERCISES

1) What is Regularization?

2) What are the different Regularization techniques?

3) Explain the Bias/variance tradeoff.

4) What is Bias?

5) What is Variance?

6) Describe the Bias-Variance Tradeoff.

7) Explain the Parsimony Model.

8) How you willchoose a Parsimonious Model?

9) Explain: AIC, BIC and MDL.

10) Explain the Cross validation.

11) Describe the methods used for Cross-Validation.

12) Write a note on limitations and applications of Cross-Validation techniques.

❖❖❖❖

6

DATA TRANSFORMATIONS

Unit Structure

6.0 Objectives

6.1 Introduction

6.2 Dimension reduction

6.2.1 The curse of dimensionality

6.2.2 Benefits of applying dimensionality reduction

6.2.3 Disadvantages of dimensionality reduction

6.2.4 Approaches of dimension reduction

6.2.5 Common techniques of dimensionality reduction

6.3 Feature extraction

6.3.1 Why feature extraction is useful?

6.3.2Applications of Feature Extraction

6.3.3 Benefits

6.3.4 Feature extraction techniques

6.4 Smoothing

6.5 Aggregating

6.5.1 Working of data aggregation

6.5.2 Examples of aggregate data

6.5.3 Data aggregators

6.6 Summary

6.7 List of References

6.8 Unit End Exercises

6.0 OBJECTIVES

- To understand the various data transformations involved in machine learning
- To get familiar with the concept of dimensionality reduction and its effect on performance
- To acquaint with the concepts of data aggregation and smoothing

6.1 INTRODUCTION

It's challenging to track or comprehend raw data. Because of this, it needs to be preprocessed before any information can be extracted from it. The process of transforming raw data into a format that makes it easier to conduct data mining and recover strategic information is known as data transformation. In order to change the data into the right form, data transformation techniques also include data cleansing and data reduction.

To produce patterns that are simpler to grasp, data transformation is a crucial data preprocessing technique that must be applied to the data before data mining.

Data transformation transforms the data into clean, useable data by altering its format, structure, or values. In two steps of the data pipeline for data analytics projects, data can be modified. Data transformation is the middle phase of an ETL (extract, transform, and load) process, which is commonly used by businesses with on-premises data warehouses. The majority of businesses now increase their compute and storage resources with latency measured in seconds or minutes by using cloud-based data warehouses. Organizations can load raw data directly into the data warehouse and perform preload transformations at query time thanks to the scalability of the cloud platform.

Data transformation may be used in data warehousing, data wrangling, data integration, and migration. Data transformation makes business and analytical processes more effective and improves the quality of data-driven decisions made by organizations. The structure of the data will be determined by an analyst throughout the data transformation process. Hence, data transformation might be:

- **Constructive:** The data transformation process adds, copies, or replicates data.
- **Destructive:** The system deletes fields or records.
- **Aesthetic:** The transformation standardizes the data to meet requirements or parameters.
- **Structural:** The database is reorganized by renaming, moving, or combining columns

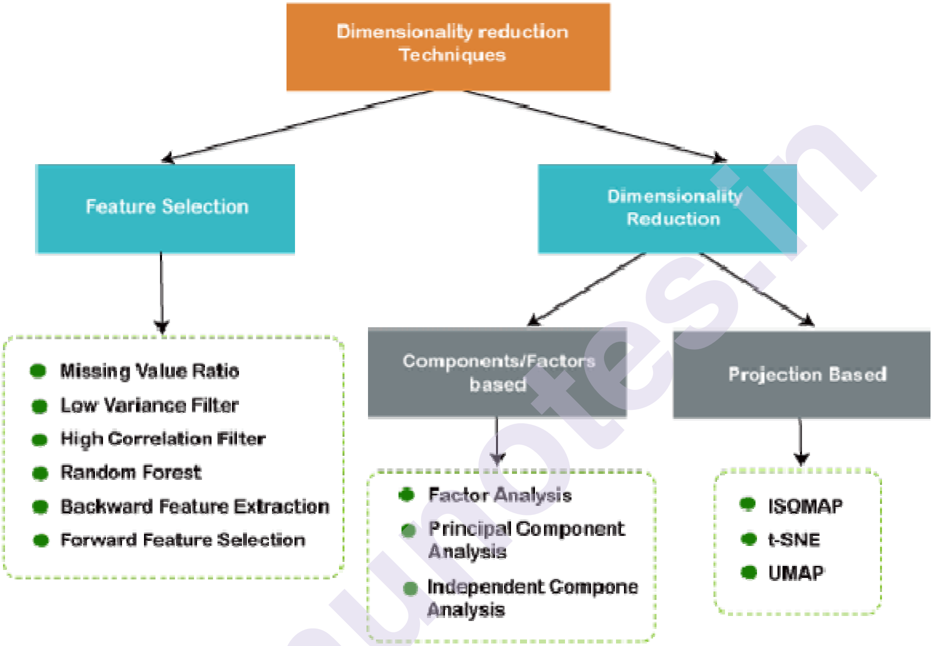
6.2 DIMENSION REDUCTION

Dimensionality refers to how many input features, variables, or columns are present in a given dataset, while dimensionality reduction refers to the process of reducing these features.

In many circumstances, a dataset has a significant number of input features, which complicates the process of predictive modelling. For training datasets with a large number of features, it is extremely challenging to visualize or anticipate the results; hence, dimensionality reduction techniques must be used.

The phrase "it is a manner of turning the higher dimensions dataset into lower dimensions dataset, guaranteeing that it gives identical information" can be used to describe the technique of "dimensionality reduction." These methods are frequently used in machine learning to solve classification and regression issues while producing a more accurate predictive model.

It is frequently utilized in disciplines like speech recognition, signal processing, bioinformatics, etc. that deal with high-dimensional data. Moreover, it can be applied to cluster analysis, noise reduction, and data visualization.



6.2.1 The curse of dimensionality

The "curse of dimensionality"the difficulty in handling high-dimensional datais a well-known phenomenon. Any machine learning algorithm and model become increasingly sophisticated as the dimensionality of the input dataset rises. As the number of characteristics rises, the number of samples rises correspondingly as well, raising the possibility of overfitting. The machine learning model performs poorly if it is overfitted after being trained on high-dimensional data.

As a result, it is frequently necessary to decrease the number of features, which can be accomplished by dimensionality reduction.

6.2.2 Benefits of applying dimensionality reduction

Following are some advantages of using the dimensionality reduction technique on the provided dataset:

- The space needed to store the dataset is decreased by lowering the dimensionality of the features.

- Reduced feature dimensions call for shorter computation training times.
- The dataset's features with reduced dimensions make the data easier to visualize rapidly.
- By taking care of the multicollinearity, it removes the redundant features (if any are present).

6.2.3 Disadvantages of dimensionality reduction

The following list of drawbacks of using the dimensionality reduction also includes:

- The reduction in dimensionality may result in some data loss.
- Sometimes the primary components needed to consider in the PCA dimensionality reduction technique are unknown.

6.2.4 Approaches of dimension reduction

There are two ways to apply the dimension reduction technique, which are given below:

A] Feature Selection

In order to create a high accuracy model, a subset of the important features from a dataset must be chosen, and the irrelevant characteristics must be excluded. This process is known as feature selection. To put it another way, it is a method of choosing the best characteristics from the input dataset.

The feature selection process employs three techniques:

1] Filter methods

In this method, the dataset is filtered, and a subset that contains only the relevant features is taken. Some common techniques of filters method are:

- Correlation
- Chi-Square Test
- ANOVA
- Information Gain, etc.

2] Wrapper methods

The wrapper technique uses a machine learning model to evaluate itself, but it has the same objective as the filter method. With this approach, some features are provided to the ML model, and performance is assessed. To improve the model's accuracy, the performance determines whether to include or exclude certain features. Although it is more difficult to use, this method is more accurate than the filtering method. The following are some typical wrapper method techniques:

- Forward Selection
- Backward Selection
- Bi-directional Elimination

3] Embedded Methods: Embedded methods check the different training iterations of the machine learning model and evaluate the importance of each feature. Some common techniques of Embedded methods are:

- LASSO
- Elastic Net
- Ridge Regression, etc.

B] Feature extraction

The process of converting a space with many dimensions into one with fewer dimensions is known as feature extraction. This strategy is helpful when we want to retain all of the information while processing it with fewer resources.

Some common feature extraction techniques are:

- Principal Component Analysis
- Linear Discriminant Analysis
- Kernel PCA
- Quadratic Discriminant Analysis

6.2.5 Common techniques of dimensionality reduction

- Principal Component Analysis
- Backward Elimination
- Forward Selection
- Score comparison
- Missing Value Ratio
- Low Variance Filter
- High Correlation Filter
- Random Forest
- Factor Analysis
- Auto-Encoder

Principal Component Analysis (PCA)

Principal Component Analysis is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the **Principal Components**. It is one of the popular tools that is used for exploratory data analysis and predictive modelling.

PCA works by considering the variance of each attribute because the high attribute shows the good split between the classes, and hence it reduces the dimensionality. Some real-world applications of PCA are *image processing, movie recommendation system, optimizing the power allocation in various communication channels.*

Backward Feature Elimination

The backward feature elimination technique is mainly used while developing Linear Regression or Logistic Regression model. Below steps are performed in this technique to reduce the dimensionality or in feature selection:

- In this technique, firstly, all the n variables of the given dataset are taken to train the model.
- The performance of the model is checked.
- Now we will remove one feature each time and train the model on n-1 features for n times, and will compute the performance of the model.
- We will check the variable that has made the smallest or no change in the performance of the model, and then we will drop that variable or features; after that, we will be left with n-1 features.
- Repeat the complete process until no feature can be dropped.

In this technique, by selecting the optimum performance of the model and maximum tolerable error rate, we can define the optimal number of features require for the machine learning algorithms.

Forward Feature Selection

Forward feature selection follows the inverse process of the backward elimination process. It means, in this technique, we don't eliminate the feature; instead, we will find the best features that can produce the highest increase in the performance of the model. Below steps are performed in this technique:

- We start with a single feature only, and progressively we will add each feature at a time.
- Here we will train the model on each feature separately.
- The feature with the best performance is selected.
- The process will be repeated until we get a significant increase in the performance of the model.

Missing Value Ratio

If a dataset has too many missing values, then we drop those variables as they do not carry much useful information. To perform this, we can set a threshold level, and if a variable has missing values more than that threshold, we will drop that variable. The higher the threshold value, the more efficient the reduction.

Low Variance Filter

As same as missing value ratio technique, data columns with some changes in the data have less information. Therefore, we need to calculate the variance of each variable, and all data columns with variance lower than a given threshold are dropped because low variance features will not affect the target variable.

High Correlation Filter

High Correlation refers to the case when two variables carry approximately similar information. Due to this factor, the performance of the model can be degraded. This correlation between the independent numerical variable gives the calculated value of the correlation coefficient. If this value is higher than the threshold value, we can remove one of the variables from the dataset. We can consider those variables or features that show a high correlation with the target variable.

Random Forest

Random Forest is a popular and very useful feature selection algorithm in machine learning. This algorithm contains an in-built feature importance package, so we do not need to program it separately. In this technique, we need to generate a large set of trees against the target variable, and with the help of usage statistics of each attribute, we need to find the subset of features.

Random forest algorithm takes only numerical variables, so we need to convert the input data into numeric data using hot encoding.

Factor Analysis

Factor analysis is a technique in which each variable is kept within a group according to the correlation with other variables, it means variables within a group can have a high correlation between themselves, but they have a low correlation with variables of other groups.

We can understand it by an example, such as if we have two variables Income and spend. These two variables have a high correlation, which means people with high income spends more, and vice versa. So, such variables are put into a group, and that group is known as the factor. The number of these factors will be reduced as compared to the original dimension of the dataset.

Auto-encoders

One of the popular methods of dimensionality reduction is auto-encoder, which is a type of ANN or artificial neural network, and its main aim is to copy the inputs to their outputs. In this, the input is compressed into latent-space representation, and output is occurred using this representation. It has mainly two parts:

- Encoder: The function of the encoder is to compress the input to form the latent-space representation.
- Decoder: The function of the decoder is to recreate the output from the latent-space representation.

6.3 FEATURE EXTRACTION

Feature extraction is a method for extracting important features from a huge input data collection. Dimensionality reduction is used in this procedure to break up enormous input data sets into more manageable processing units.

The dimensionality reduction method, which divides and condenses a starting set of raw data into smaller, easier-to-manage groupings, includes feature extraction. As a result, processing will be simpler. The fact that these enormous data sets contain a lot of different variables is their most crucial feature. Processing these variables takes a lot of computing power. In order to efficiently reduce the amount of data, feature extraction helps to extract the best feature from those large data sets by choosing and combining variables into features. These features are simple to use while still accurately and uniquely describing the real data set.

6.3.1 Why feature extraction is useful?

When you have a large data set and need to conserve resources without losing any crucial or pertinent information, the feature extraction technique can be helpful. The amount of redundant data in the data collection is decreased with the aid of feature extraction.

In the end, the data reduction speeds up the learning and generalization phases of the machine learning process while also enabling the model to be built with less machine effort.

6.3.2 Applications of Feature Extraction

- **Bag of Words:** Bag-of-Words is the most used technique for natural language processing. In this process they extract the words or the features from a sentence, document, website, etc. and then they classify them into the frequency of use. So, in this whole process feature extraction is one of the most important parts.
- **Image Processing**: Image processing is one of the best and most interesting domains. In this domain basically you will start playing with your images in order to understand them. So here we use many

techniques which includes feature extraction as well and algorithms to detect features such as shaped, edges, or motion in a digital image or video to process them.

- **Auto-encoders:** The main purpose of the auto-encoders is efficient data coding which is unsupervised in nature. this process comes under unsupervised learning. So, Feature extraction procedure is applicable here to identify the key features from the data to code by learning from the coding of the original data set to derive new ones.

6.3.3 Benefits

Feature extraction can prove helpful when training a machine learning model. It leads to:

- A Boost in training speed
- An improvement in model accuracy
- A reduction in risk of overfitting
- A rise in model explainability
- Better data visualization

6.3.4 Feature extraction techniques

The following is a list of some common feature extraction techniques:

- Principle Components Analysis (PCA)
- Independent Component Analysis (ICA)
- Linear Discriminant Analysis (LDA)
- Locally Linear Embedding (LLE)
- t-distributed Stochastic Neighbor Embedding (t-SNE)

6.4 DATA SMOOTHING

Data smoothing is the process of taking out noise from a data set using an algorithm. Important patterns can then be more easily distinguished as a result.

Data smoothing can be used in economic analysis as well as to assist predict trends, such as those seen in securities prices. The purpose of data smoothing is to eliminate singular outliers and account for seasonality.

- **Advantages and disadvantages**

The identification of patterns in the economy, in financial instruments like stocks, and in consumer mood can be aided by data smoothing. Further commercial uses for data smoothing are possible.

By minimizing the changes that may occur each month, such as vacations or petrol prices, an economist can smooth out data to make seasonal adjustments for particular indicators, such retail sales.

Yet, there are drawbacks to using this technology. When identifying trends or patterns, data smoothing doesn't necessarily explain them. It might also cause certain data points to be overlooked in favor of others.

Pros

- Helps identify real trends by eliminating noise from the data
- Allows for seasonal adjustments of economic data
- Easily achieved through several techniques including moving averages

Cons

- Removing data always comes with less information to analyze, increasing the risk of errors in analysis
- Smoothing may emphasize analysts' biases and ignore outliers that may be meaningful

6.5 AGGREGATING

Finding, gathering, and presenting data in a condensed style is the process of aggregation, which is used to do statistical analysis of business plans or analysis of behavioral patterns in people. It's essential to acquire reliable data when a lot of data is collected from several sources in order to produce meaningful results. Aggregating data can assist in making wise selections in marketing, finances, product pricing, etc. The statistical summaries replace aggregated data groups. As aggregated data is present in the data warehouse, using it to address rational issues might speed up the process of answering queries from data sets.

6.5.1 Working of data aggregation

When a dataset's total amount of information is useless and unable to be used for analysis, data aggregation is required. To achieve desired results and improve the user experience or the application itself, the datasets are compiled into useable aggregates. They offer aggregation metrics including sum, count, and average. Summarized data is useful for researching client demographics and patterns of activity. After being written as reports, aggregated data assist in uncovering insightful facts about a group. Understanding, capturing, and visualizing data aids in data lineage, which aids in identifying the primary causes of errors in data analytics. An aggregated element does not necessarily have to be a number.We can also find the count of non-numeric data. Aggregation must be done for a group of data and not based on individual data.

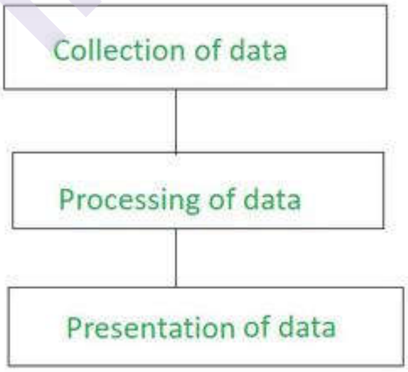
6.5.2 Examples of aggregate data

- Finding the average age of customer buying a particular product which can help in finding out the targeted age group for that particular product. Instead of dealing with an individual customer, the average age of the customer is calculated.
- Finding the number of consumers by country. This can increase sales in the country with more buyers and help the company to enhance its marketing in a country with low buyers. Here also, instead of an individual buyer, a group of buyers in a country are considered.
- By collecting the data from online buyers, the company can analyze the consumer behaviour pattern, the success of the product which helps the marketing and finance department to find new marketing strategies and planning the budget.
- Finding the value of voter turnout in a state or country. It is done by counting the total votes of a candidate in a particular region instead of counting the individual voter records.

6.5.3 Data aggregators

A system used in data mining called a "data aggregator" gathers information from many sources, analyses it, and then repackages it in usable packages. They significantly contribute to the improvement of client data by serving as an agent. When a consumer asks data examples concerning a specific product, it aids in the query and delivery process. The customers receive matching records for the goods from the aggregators. The consumer can thereby purchase any matching record instances.

Working



The working of data aggregators takes place in three steps:

- **Collection of data:** Collecting data from different datasets from the enormous database. The data can be extracted using IoT(Internet of things) such as

- Communications in social media
- Speech recognition like call centers
- Headlines of a news
- Browsing history and other personal data of devices.
- **Processing of data:** After collecting data, the data aggregator finds the atomic data and aggregates it. In the processing technique, aggregators use various algorithms from the field of Artificial Intelligence or Machine learning techniques. It also incorporates statistical methods to process it, like the predictive analysis. By this, various useful insights can be extracted from raw data.
- **Presentation of data:** After the processing step, the data will be in a summarized format which can provide a desirable statistical result with detailed and accurate data.

6.6 SUMMARY

The modification of data characteristics for better access or storage is known as data transformation. Data's format, structure, or values may all undergo transformation. Data analytics transformation typically takes place after data has been extracted or loaded (ETL/ELT).

Data transformation improves the effectiveness of analytical procedures and makes it possible to make judgements using data. There is a need for clean, usable data since raw data is frequently challenging to examine and has a size that is too great to yield useful insight.

An analyst or engineer will choose the data structure before starting the transformation procedure. The following are the most typical types of data transformation:

- **Constructive:** The process of data transformation adds, duplicates, or copies data.
- **Destructive:** System deletes fields or records, which is destructive.
- **Aesthetic:** The data are standardized through the transformation to adhere to specifications or guidelines.
- **Structural:** a re-structured database consists of combining, re-naming, or shifting a number of columns.

A practitioner may additionally map data and save data using the right database technology.

6.7 LIST OF REFERENCES

1. Doing Data Science, Rachel Schutt and Cathy O'Neil, O'Reilly,2013.
2. Mastering Machine Learning with R, Cory Lesmeister, PACKT Publication,2015.
3. Hands-On Programming with R, Garrett Grolemund,1st Edition, 2014.
4. An Introduction to Statistical Learning, James, G., Witten, D., Hastie, T., Tibshirani, R.,Springer,2015.

6.8 UNIT END EXERCISES

1] What do you mean by Dimension reduction?

2] What is the curse of dimensionality?

3] What are the benefits of applying dimensionality reduction?

4] State the disadvantages of dimensionality reduction.

5] Explain the different approaches of dimension reduction.

6] What are the common techniques of dimensionality reduction?

7] What is Feature extraction?

8] Why feature extraction is useful?

9] State the benefits and applications of Feature Extraction.

10] Describe various feature extraction techniques.

11] Define Smoothing and Aggregating.

12] Explain the working of data aggregation.

12] What are the different examples of aggregate data.

13] What is Data aggregators?

❖❖❖❖

7

SUPERVISED LEARNING

Unit Structure

7.0 Objectives

7.1 Introduction

7.2 Linear models

7.2.1 What is linear model?

7.2.2 Types of linear model

7.2.3 Applications of linear model

7.3 Regression trees

7.3.1 What are regression trees?

7.3.2 Mean square error

7.3.3 Building a regression tree

7.4 Time-series Analysis

7.4.1 What is time series analysis?

7.4.2 Types of time series analysis

7.4.3 Value of time series analysis

7.4.4 Time series models and techniques

7.5 Forecasting

7.5.1 Time series forecasting in machine learning

7.5.2 Machine learning models for time series forecasting

7.5.3 Machine learning time series forecasting applications

7.6 Classification trees

7.7 Logistic regression

7.8 Classification using separating hyperplanes

7.9 k-NN

7.9.1 Need of KNN Algorithm

7.9.2 Working of KNN Algorithm

7.9.3 Selecting value of k in KNN Algorithm

7.9.4 Advantages of KNN Algorithm

7.9.5 Disadvantages of KNN Algorithm

7.10 Summary

7.11 List of References

7.12 Unit End Exercises

7.0 OBJECTIVES

- To understand the supervised learning mechanisms
- To learn about different regression and classification models

7.1 INTRODUCTION

A class of techniques and algorithms known as "supervised learning" in machine learning and artificial intelligence develop predictive models utilizing data points with predetermined outcomes. The model is trained using an appropriate learning technique (such as neural networks, linear regression, or random forests), which often employs some sort of optimization procedure to reduce a loss or error function.

In other words, supervised learning is the process of training a model by providing both the right input data and output data. The term "labelled data" is typically used to describe this input/output pair. Consider a teacher who, armed with the right answers, will award or deduct points from a student depending on how accurately she answered a question. For two different sorts of issues, supervised learning is frequently utilized to develop machine learning models.

- Regression: The model identifies outputs that correspond to actual variables (number which can have decimals.)
- Classification: The model creates categories for its inputs.

7.2 LINEAR MODELS

One of the simplest models in machine learning is the linear model. It serves as the foundation for many sophisticated machine learning techniques, such as deep neural networks. Using a linear function of the input data, linear models forecast the target variable. Here, we've covered linear regression and logistic regression, two essential linear models in machine learning. While logistic regression is a classification algorithm, linear regression is utilized for jobs involving regression.

7.2.1 What is linear model?

One of the simplest models in machine learning is the linear model. It attempts to determine the importance of each feature while assuming that the data can be linearly separated. In mathematics, it may be expressed as

$Y = W^T X$

To turn the continuous-valued variable Y into a discrete category for the classification issue, we apply a transformation function or threshold. Here, we'll quickly go over the models for the classification and regression tasks, respectively: logistic and linear regression.

7.2.2 Types of linear model

1] Linear regression

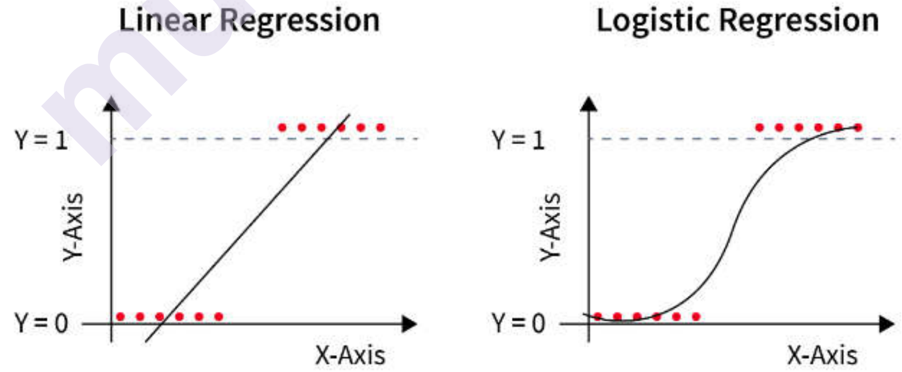
A statistical method known as "linear regression" makes predictions about the outcome of a response variable by fusing a variety of affecting variables. It makes an effort to depict the target's (dependent variables)linear relationship with features (independent variables). We can determine the ideal model parameter values using the cost function.

Example: An analyst would be interested in seeing how market movement influences the price of ExxonMobil (XOM). The value of the S&P 500 index will be the independent variable, or predictor, in this example, while the price of XOM will be the dependent variable. In reality, various elements influence an event's result. Hence, we usually have many independent features.

2] Logistic regression

A progression from linear regression is logistic regression. The result of the linear regression is first transformed between 0 and 1 by the sigmoid function. Following that, a predetermined threshold aids in calculating the likelihood of the output values. Values over the threshold value have a tendency to have a probability of 1, whereas values below the threshold value have a tendency to have a probability of 0.

Example: A bank wants to predict if a customer will default on their loan based on their credit score and income. The independent variables would be credit score and income, while the dependent variable would be whether the customer defaults (1) or not (0).



7.2.3 Applications of linear model

There are many situations in real life when dependent and independent variables follow linear relationships. Such instances include:

- The connection between elevation variation and the boiling point of water.
- The connection between an organization's revenue and its advertising expenditures.

- The connection between fertilizer application rates and agricultural yields.
- Athletes' performances and training schedule.

7.3 REGRESSION TREES

7.3.1 What are regression trees?

A regression tree, which is used to predict continuous valued outputs rather than discrete outputs, is essentially a decision tree that is employed for the regression job.

7.3.2 Mean square error

To provide accurate and effective classifications, decision trees for classification pose the proper questions at the appropriate nodes. Entropy and Information Gain are the two metrics used in Classifier Trees to accomplish this. But, since we are making predictions about continuous variables, we are unable to compute the entropy and follow the same procedure. Now, we require a different approach. The mean square error is a measurement that indicates how much our projections stray from the initial goal.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

We only care about how far the prediction deviates from the target; Y is the actual value, and Y hat is the prediction. not which way around. In order to divide the total amount by the total number of records, we square the difference.

We follow the same procedure as with classification trees in the regression tree approach. But rather than focusing on entropy, we strive to lower the Mean Square Error for each child.

7.3.3 Building a regression tree

Consider the dataset below, which has 2 variables.

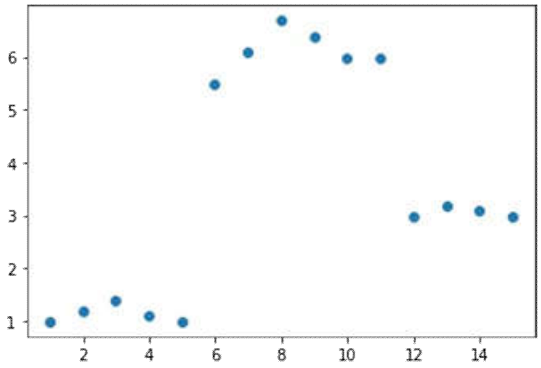
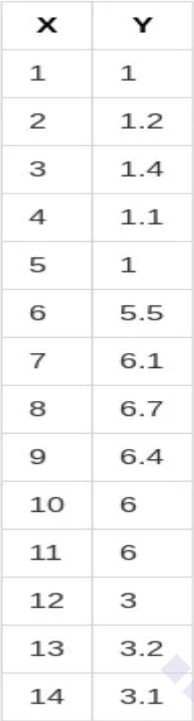


Figure 1: Dataset where X and Y are a continuous variable



X	Y
1	1
2	1.2
3	1.4
4	1.1
5	1
6	5.5
7	6.1
8	6.7
9	6.4
10	6
11	6
12	3
13	3.2
14	3.1

Figure 2: Actual dataset

We need to build a Regression tree that best predicts the Y given the X.

Step 1

The first step is to sort the data based on X (*In this case, it is already sorted)*. Then, take the average of the first 2 rows in variable X (which is (1+2)/2 = 1.5 according to the given dataset). Divide the dataset into 2 parts *(Part A and Part B)*, separated by x < 1.5 and X ≥ 1.5.

Now, Part A consist only of one point, which is the first row (1,1) and all the other points are in Part — B. Now, take the average of all the Y values in Part A and average of all Y values in Part B separately. These 2 values are the predicted output of the decision tree for x < 1.5 and x ≥ 1.5 respectively. Using the predicted and original values, calculate the mean square error and note it down.

Step 2

In step 1, we calculated the average for the first 2 numbers of sorted X and split the dataset based on that and calculated the predictions. Then, we do the same process again but this time, we calculate the average for the second 2 numbers of sorted X ((2+3)/2 = 2.5). Then, we split the dataset again based on X < 2.5 and X ≥ 2.5 into Part A and Part B again and predict outputs, find mean square error as shown in step 1. This process is repeated for the third 2 numbers, the fourth 2 numbers, the 5th, 6th, 7th till n-1th 2 numbers (*where n is the number of records or rows in the dataset*).

Step 3

Now that we have n-1 mean squared errors calculated, we need to choose the point at which we are going to split the dataset. and that point is the point, which resulted in the lowest mean squared error on splitting at it. In this case, the point is x=5.5. Hence the tree will be split into 2 parts. x<5.5 and x≥ 5.5. The Root node is selected this way and the data points that go towards the left child and right child of the root node are further recursively exposed to the same algorithm for further splitting.

Brief Explanation of working of the algorithm:

The basic idea behind the algorithm is to find the point in the independent variable to split the data-set into 2 parts, so that the mean squared error is the minimised at that point. The algorithm does this in a repetitive fashion and forms a tree-like structure.

A regression tree for the above shown dataset would look like this

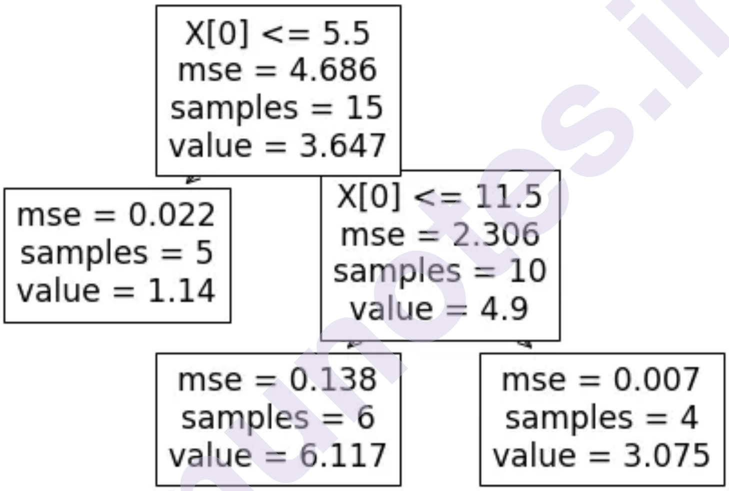


Figure 3: Resultant decision tree and the resultant prediction visualisation would be this

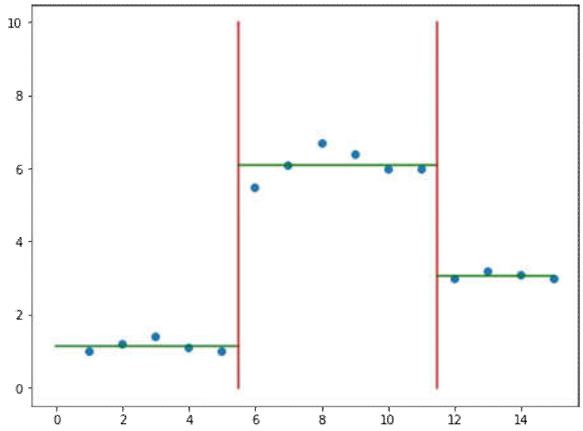


Figure 4: The decision boundary

7.4 TIME-SERIES ANALYSIS

7.4.1 What is time series analysis?

A method of examining a collection of data points gathered over time is a time series analysis. Additionally, it is specifically utilized for non-stationary data, or data that is constantly changing over time. The time series data varies from all other data due to this component as well. Time series analysis is also used to predict future data based on the past. As a result, we can conclude that it goes beyond simply gathering data.

Predictive analytics includes the subfield of time series analysis. It supports in forecasting by projecting anticipated variations in data, such as seasonality or cyclical activity, which provides a greater understanding of the variables.

7.4.2 Types of time series analysis

Time series are used to collect a variety of data kinds;thus, analysts have created some intricate models to help with understanding. Analysts, on the other hand, are unable to take into account all variations or generalize a specific model to all samples. These are the typical time series analysis methods:

- **Classification:** This model is used for the identification of data. It also allocates categories to the data.

- **Descriptive Analysis:** As time series data has various components, this descriptive analysis helps to identify the varied patterns of time series including trend, seasonal, or cyclic fluctuations.

- **Curve Fitting:** Under this type of time series analysis, we generally plot data along some curve in order to investigate the correlations between variables in the data.

- **Explanative Analysis:** This model basically explains the correlations between the data and the variables within it, and also explains the causes and effects of the data on the time series.

- **Exploratory Analysis:** The main function of this model is to highlight the key features of time series data, generally in a graphic style.

- **Forecasting:** As the name implies, this form of analysis is used to forecast future data. Interestingly, this model uses the past data (trend) to forecast the forthcoming data, thus, projecting what could happen at future plot points.

- **Intervention Analysis:** This analysis model of time series denotes or investigates how a single incident may alter data.

- **Segmentation:** This type typically divides the data into several segments in order to display the underlying attributes of the original data.

- **Data Variation:** It includes

- **Functional Analysis**: It helps in the picking of patterns within data and also correlates a notable relationship.

- **Trend Analysis:** It refers to the constant movement in a specific direction. Trends are classified into two types: deterministic (determining core causes) and stochastic (inexplicable).

- **Seasonal Variation**: It defines event occurrences that specifically happen at certain and regular periods throughout the year.

7.4.3 Value of time series analysis

Our lives are significantly impacted by time series analysis. It aids businesses and organizations in examining the root causes of trends or other systematic patterns across time. Moreover, with all these facts, you can put them down in a chart visually and that assists in a deeper knowledge of the industry. In turn, this will help firms delve more into the fundamental causes of seasonal patterns or trends.

Also, it aids organizations in projecting how specific occurrences will turn out in the future. This may be achieved by performing ongoing analyses of past data. Predictive analytics therefore includes time series forecasting as a subset. It enables more precise data analysis and forecasting by anticipating projected variations in data, such as seasonality, trend, or cyclic behavior.

Time series forecasting also contains additional crucial components.

- **Dependable:** Time series forecasting is one of the most dependable methods available today. It is trustworthy when the data represents a long-time span. At regular periods, various significant information can be gleaned from the data fluctuations.

- **Seasonal Patterns:** Changes in data points indicate a seasonal fluctuation that forms the basis for projections of the future. This information is essential to the market since it allows for a basic strategy for production and other costs in a market where the product swings seasonally.

- **Estimated trend:** Time series analysis, together with seasonal patterns, is helpful in identifying trends. This will eventually assist the management in keeping track of data trends that show an uptick or decline in sales of a particular product.

- **Growth:** Another significant feature of time series analysis is that it also adds to the financial as well as endogenous growth of an organization. Endogenous growth is the internal expansion of a business that resulted in increased financial capital. Time series analysis can be used to detect changes in policy factors, which is a great illustration of the value of this series in many domains.

Several sectors have noted the prevalence of time series analysis. Statistics professionals frequently utilize it to determine probability and other fundamentals. Also, it is crucial in the medical sectors.

Mathematicians also prefer time series because econometrics uses them as well. It is crucial for predicting earthquakes and other natural disasters, estimating their impact zones, and identifying weather patterns for forecasting.

7.4.4 Time series models and techniques

Data in a time series can be examined in a variety of ways. These popular time series models can be applied to data analysis:

1. Decompositional Models

The time series data shows certain patterns. Consequently, it is quite beneficial to divide the time series into different parts for simple comprehension. Each element represents a particular pattern. The term "decompositional models" refers to this procedure. The time series is primarily broken down into three main components: trend, seasonality, and noise. Predictability and change-rate decomposition are the two types of decomposition.

2. Smoothing-based Model

This technique is one of the most statistical ones for time series because it concentrates on removing outliers from the data and enhancing the pattern's visibility. The process of gathering data over time involves some random fluctuation. In order to show underlying patterns and cyclic components, data smoothing removes or reduces random fluctuation.

3. Moving Average Model

Moving Average, or MA model, is a well-liked technique for modelling univariate time series in time series analysis. The anticipated output is linearly correlated with the present and other prior values of a probabilistic term, according to the moving-average model.

4. Exponential Smoothing Model

A quick method for blending time series data that use the "exponential window function" is exponential smoothing. This process is simple to learn and can be used to base decisions on historical user expectations, such as seasonality. This model comes in essentially three varieties: single, double, and triple exponential smoothing.

Moreover, it is a crucial component of the ARMA and ARIMA models. Moreover, this model is employed because of the TBATS forecasting model.

5. ARIMA

AutoRegressive Integrated Moving Average is abbreviated as ARMA. It is the forecasting technique in time series analysis that is most frequently utilised. The Moving Average Model and the Autoregressive Model are combined to create it.

Hence, rather than focusing on individual values in the series, the model instead seeks to estimate future time series movement. When there is evidence of non-stationarity in the data, ARIMA models are applied.

A linear mixture of seasonal historical values and/or prediction errors is added to the SARIMA model, also known as the Seasonal ARIMA model, in addition to these.

7.5 FORECASTING

Forecasting is a method of foretelling the future using the outcomes of the past data. In order to anticipate future events, a thorough analysis of past and present trends or events is required. It makes use of statistical methods and tools.

Time series forecasting is employed in various sectors, including finance, supply chain management, production, and inventory planning, making it one of the most widely used data science approaches. Time series forecasting has many applications, including resource allocation, business planning, weather forecasts, and stock price prediction.

7.5.1 Time series forecasting in machine learning

A set of observations made through time, whether daily, weekly, monthly, or yearly, make up a time series forecasting method. Time series analysis involves building models in order to describe the observed time series and understand the "why" underlying its dataset. This includes speculating and interpreting scenarios based on the information at hand. In time series forecasting, the best-fitting model is employed to predict future observations based on meticulously processed recent and previous data.

Machine learning-based time series analysis forecasting has been demonstrated to be the most effective at finding trends in both structured and unstructured data.

- Understanding the components of the time series data is essential to using an appropriate deep learning model for time series forecasting.

- Finding repeating changes in a time series and determining whether they are cyclical.

- The term "trends" is used to characterize the upward or downward motion of time series, which is often displayed in linear modes.

- Seasonality: To highlight the repeating patterns of behavior across time.

- To take into account the random component of time series that deviates from the conventional model values.

7.5.2 Machine learning models for time series forecasting

There are numerous models that can be used for time series forecasting. One particular kind of neural network that uses historical data to forecast outcomes is the LSTM Network. It is frequently employed for a variety of tasks, including language recognition and time series analysis. Models like the random forest, gradient boosting regressor, and time delay neural networks can contain temporal information and represent the data at different points in time by adding a series of delays to the input.

1] Naive model

Naive models are often implemented as a random walk and a seasonal random walk, with the most recent value observed serving as the unit for the forecast for the following period (a forecast is made using a value from the same time period as the most recent observation).

2] Exponential smoothing model

An exponential smoothing time series forecasting technique can be expanded to support data with a systematic trend or seasonal component. It is a potent forecasting technique that can be employed in place of the well-known Box-Jenkins ARIMA family of techniques.

3] ARIMA/ SARIMA

For building a composite time series model, the approaches of Autoregressive (AR) and Moving Average (MA) are combined under the term ARIMA. ARIMA models incorporate seasonal and trend factors (for example, dummy variables for weekdays and their ability to differentiate). Additionally, they allow the handling of the underlying autocorrelation in the data by using moving averages and autoregressive terms.

By incorporating a linear mixture of previous seasonal values and/or forecast mistakes, the seasonal autoregressive integrated moving average, or SARIMA, expands the use of the ARIMA.

4] Linear regression method

The simple statistical technique known as linear regression is commonly used in predictive modelling. In its most basic form, providing an equation of independent variables upon which our goal variable is based is all that is required.

5] Multi-layer perceptron (MLP)

The term “MLP” is used ambiguously; sometimes, it is used broadly to refer to any feedforward ANN and other times, it is used specifically to describe networks made up of several layers of perceptron’s.

6] Recurrent neural network (RNN)

RNNs may predict time-dependent objectives because they are essentially memory-enhanced neural networks. Recurrent neural networks are capable of remembering the status of previously acquired input while determining the appropriate time step. Recurrent networks have lately undergone several improvements that can be used in a variety of sectors.

7] Long short-term memory (LSTM)

By giving the model multiple gate options, LSTM cells (special RNN cells) were developed to solve the gradient problem. These gates let the model to choose which data to recognize as meaningful and which data to disregard. The GRU is yet another variety of gated recurrent network.

CNNs, also referred to as convolutional neural network models, decision tree-based models like Random Forest, and variations of gradient boosting (LightGBM, CatBoost, etc.) can be used for time series forecasting in addition to the methods mentioned above.

7.5.3 Machine learning time series forecasting applications

Time series forecasting can be used by any business or organization dealing with continuously generated data and the requirement to adjust to operational shifts and changes. Here, machine learning acts as the greatest enabler, improving our ability to:

- **Web traffic forecasting:** Common data on normal traffic rates among competing websites is paired with input data on traffic-related trends to anticipate online traffic rates for certain times.

- **Sales and demand forecasting:** Machine learning models can identify the most in-demand products and arrange them precisely in the dynamic market using data on customer behavior patterns along with inputs from purchase history, demand history, seasonal influence, etc.

- **Weather forecasting:** Time-based data are regularly gathered from numerous worldwide networked weather stations, and machine learning techniques enable in-depth analysis and interpretation of the data for upcoming forecasts based on statistical dynamics.

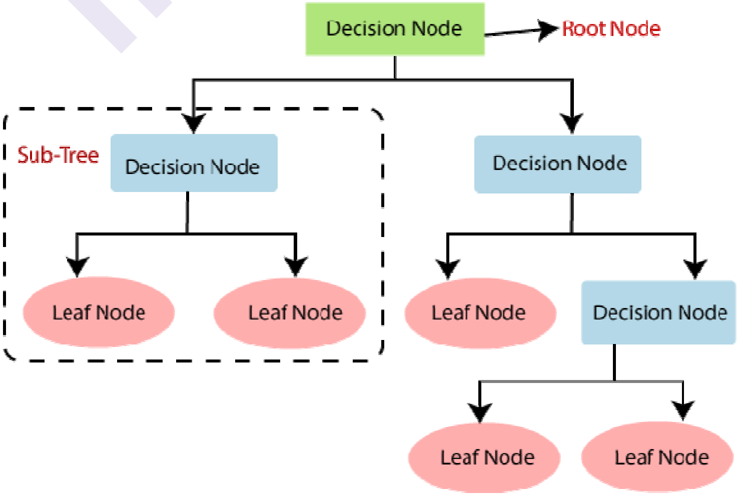
- **Stock price forecasting:** To produce precise predictions of the most likely upcoming stock price movements, one can combine historical stock price data with knowledge of both common and atypical stock market surges and drops.

- **Forecasting based on economic and demographic factors:** Economic and demographic factors contain a wealth of statistical information that can be used to accurately forecast time series data. Hence, the optimum target market may be defined, and the most effective tactics to interact with that specific TA may be developed.

- **Academics:** Deep learning and machine learning theories significantly speed up the procedures of developing and presenting scientific theories. For instance, machine learning patterns may enable the analysis of scientific data that must undergo countless iterations of analysis much more swiftly.

7.6 CLASSIFICATION TREES

A supervised learning method called a decision tree can be used to solve classification and regression problems, but it is typically favored for doing so. It is a tree-structured classifier, where internal nodes stand in for a dataset's features, branches for the decision-making process, and each leaf node for the classification result. The Decision Node and Leaf Node are the two nodes of a decision tree. Whereas Leaf nodes are the results of decisions and do not have any more branches, Decision nodes are used to create decisions and have numerous branches.The given dataset's features are used to execute the test or make the decisions.The given dataset's features are used to execute the test or make the decisions.It is a graphical depiction for obtaining all feasible answers to a choice or problem based on predetermined conditions.It is known as a decision tree because, like a tree, it begins with the root node and grows on subsequent branches to form a structure resembling a tree.The CART algorithm, which stands for Classification and Regression Tree algorithm, is used to construct a tree.A decision tree simply asks a question, then based on the answer (Yes/No), it further split the tree into subtrees.The decision tree's general structure is shown in the diagram below:



Decision Tree Terminologies:

- **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.

Working of an algorithm:

In a decision tree, the algorithm begins at the root node and works its way up to forecast the class of the given dataset. This algorithm follows the branch and jumps to the following node by comparing the values of the root attribute with those of the record (real dataset) attribute.

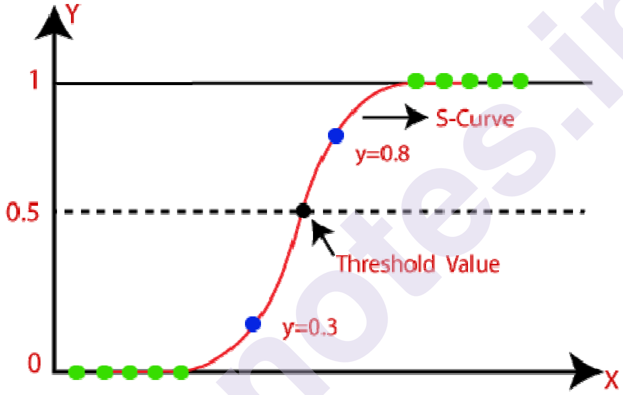
The algorithm verifies the attribute value with the other sub-nodes once again for the following node before continuing. It keeps doing this until it reaches the tree's leaf node. The following algorithm can help you comprehend the entire procedure:

- **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.

7.7 LOGISTIC REGRESSION

One of the most often used Machine Learning algorithms, within the category of Supervised Learning, is logistic regression. With a predetermined set of independent factors, it is used to predict the categorical dependent variable. In a categorical dependent variable, the output is predicted via logistic regression. As a result, the result must be a discrete or categorical value. Rather of providing the exact values of 0 and 1, it provides the probabilistic values that fall between 0 and 1. It can be either Yes or No, 0 or 1, true or false, etc. With the exception of how they are applied, logistic regression and linear regression are very similar. Whereas logistic regression is used to solve classification difficulties, linear regression is used to solve regression problems.

In logistic regression, we fit a "S" shaped logistic function, which predicts two maximum values, rather than a regression line (0 or 1). The logistic function's curve shows the possibility of several things, including whether or not the cells are malignant, whether or not a mouse is obese depending on its weight, etc.Logistic Regression is a major machine learning technique since it has the capacity to offer probabilities and categorize new data using continuous and discrete datasets. When classifying observations using various sources of data, logistic regression can be used to quickly identify the factors that will work well. The logistic function is displayed in the graphic below:



Logistic function (Sigmoid function):

The projected values are converted to probabilities using a mathematical tool called the sigmoid function.It transforms any real value between 0 and 1 into another value.The logistic regression's result must fall within the range of 0 and 1, and because it cannot go beyond this value, it has the shape of a "S" curve. The S-form curve is called the Sigmoid function or the logistic function.We apply the threshold value idea in logistic regression, which establishes the likelihood of either 0 or 1. Examples include values that incline to 1 over the threshold value and to 0 below it.

Assumptions for logistic regression:

- The dependent variable must be categorical in nature.
- The independent variable should not have multi-collinearity.

Type of logistic regression:

On the basis of the categories, Logistic Regression can be classified into three types:

- **Binomial:** In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.

- **Multinomial:** In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"
- **Ordinal:** In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

7.8 CLASSIFICATION USING SEPARATING HYPERPLANES

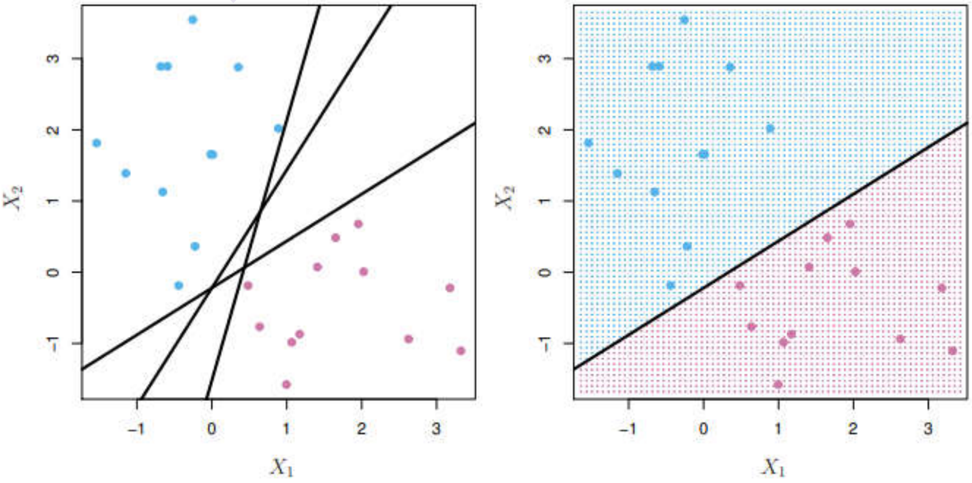
Suppose that we have a n×p data matrix X that consists of n training observations in p-dimensional space

$$x_1 = \begin{pmatrix} x_{11} \\ \vdots \\ x_{1p} \end{pmatrix}, \ldots, x_n = \begin{pmatrix} x_{n1} \\ \vdots \\ x_{np} \end{pmatrix},$$

and that these observations fall into two classesthat is, $y_1,\ldots,y_n \in \{-1, 1\}$ where −1 represents one class and 1 the other class. We also have a test observation, a p-vector of observed features $x^* = (x_1^* \ldots. X_p^*)^T$. Our goal is to develop a classifier based on the training data that will correctly classify the test observation using its feature measurements

Suppose that it is possible to construct a hyperplane that separates the training observations perfectly according to their class labels. Examples of three such separating hyperplanes are shown in the left-hand panel of Figure. We can label the observations from the blue class as $y_i = 1$ and those from the purple class as $y_i = -1$. Then a separating hyperplane has the property that

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} > 0 \text{ if } y_i = 1,$$



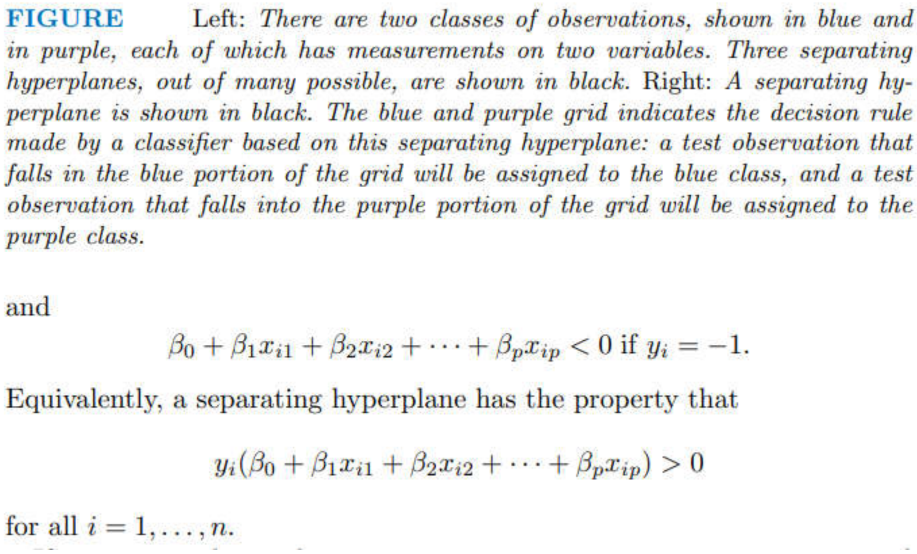


FIGURE Left: *There are two classes of observations, shown in blue and in purple, each of which has measurements on two variables. Three separating hyperplanes, out of many possible, are shown in black.* Right: *A separating hyperplane is shown in black. The blue and purple grid indicates the decision rule made by a classifier based on this separating hyperplane: a test observation that falls in the blue portion of the grid will be assigned to the blue class, and a test observation that falls into the purple portion of the grid will be assigned to the purple class.*

and

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} < 0 \text{ if } y_i = -1.$$

Equivalently, a separating hyperplane has the property that

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) > 0$$

for all $i = 1, \ldots, n$.

If a separating hyperplane exists, we can use it to construct a very natural classifier: a test observation is assigned a class depending on which side of the hyperplane it is located. The right-hand panel of Figure shows an example of such a classifier. That is, we classify the test observation x∗ based on the sign of

$$f(x^*) = \beta_0+\beta_1 x_1^*+\beta_2 x_2^*+\cdots+\beta_p x_p^*$$

If f(x*) is positive, then we assign the test observation to class 1, and if f(x*) is negative, then we assign it to class −1. We can also make use of the magnitude of f(x*). If f(x*) is far from zero, then this means that x* lies far from the hyperplane, and so we can be confident about our class assignment for x*. On the other hand, if f(x*) is close to zero, then x∗ is located near the hyperplane, and so we are less certain about the class assignment for x*. As we see in Figure, a classifier that is based on a separating hyperplane leads to a linear decision boundary.

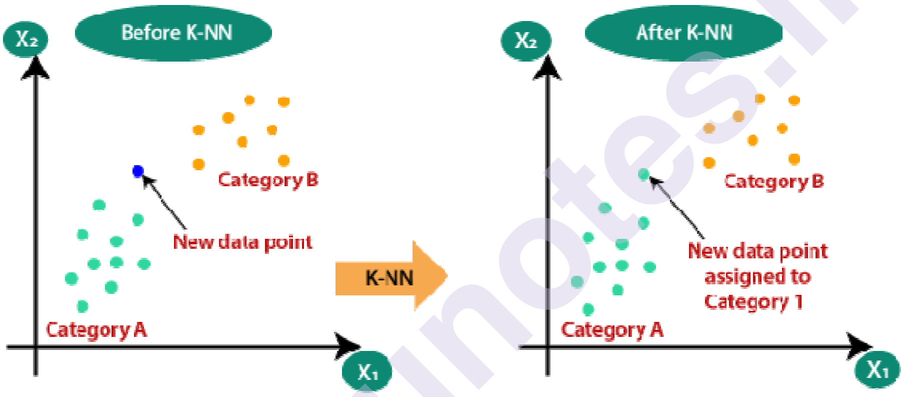
7.9 K-NN

One of the simplest machine learning algorithms, based on the supervised learning method, is K-Nearest Neighbor.The K-NN algorithm makes the assumption that the new case and the existing cases are comparable, and it places the new instance in the category that is most like the existing categories.A new data point is classified using the K-NN algorithm based on similarity after all the existing data has been stored. This means that utilizing the K-NN method, fresh data can be quickly and accurately sorted into a suitable category.Although the K-NN approach is most frequently employed for classification problems, it can also be utilized for regression.Since K-NN is a non-parametric technique, it makes no assumptions about the underlying data.It is also known as a lazy learner algorithm since it saves the training dataset rather than learning from it immediately. Instead, it uses the dataset to perform an action when classifying data.KNN method maintains the dataset during the training phase and subsequently classifies new data into a category that is quite similar to the new data.

Consider the following scenario: We have a photograph of a creature that resembles both cats and dogs, but we are unsure of its identity. However, since the KNN algorithm is based on a similarity metric, we can utilize it for this identification. Our KNN model will examine the new data set for features that are comparable to those found in the photographs of cats and dogs, and based on those features, it will classify the data as belonging to either the cat or dog group.

7.9.1 Need of KNN Algorithm

If there are two categories, Category A and Category B, and we have a new data point, x1, which category does this data point belong in? We require a K-NN algorithm to address this kind of issue. K-NN makes it simple to determine the category or class of a given dataset. Take a look at the diagram below:

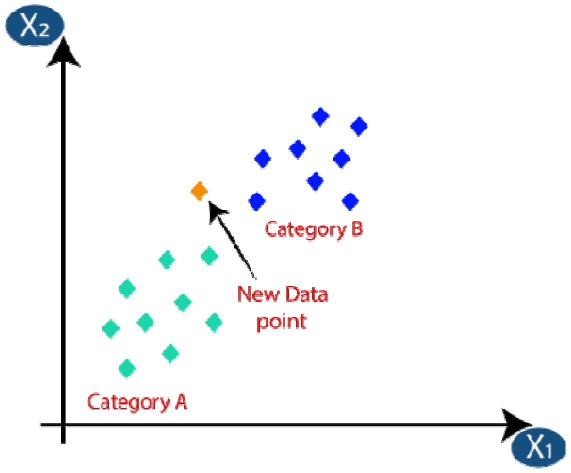


7.9.2 Working of KNN Algorithm

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

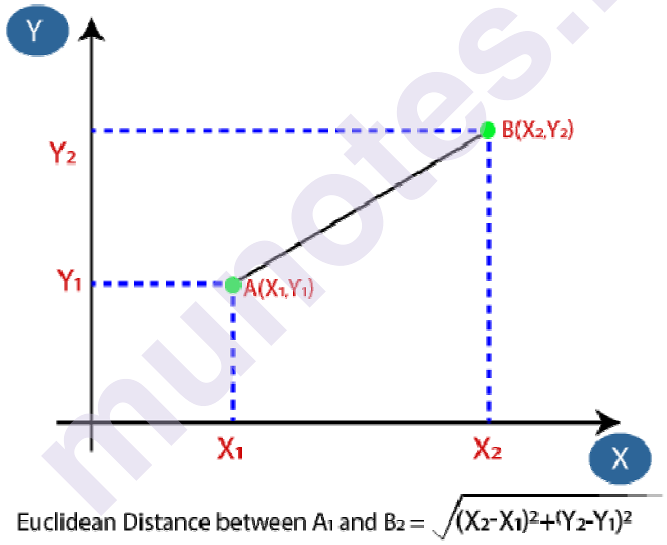
- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of K number of neighbors
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **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.

Suppose we have a new data point and we need to put it in the required category. Consider the below image:

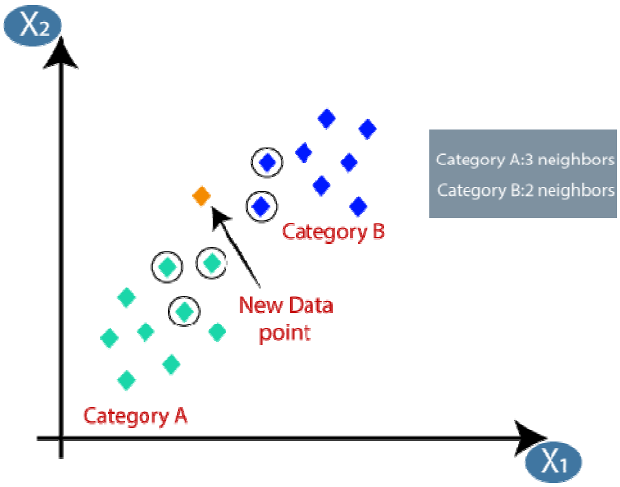


- Firstly, we will choose the number of neighbors, so we will choose the k=5.

- Next, we will calculate the Euclidean distance between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



- By calculating the Euclidean distance, we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:



- As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

7.9.3 Selecting value of k in KNN 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.

7.9.4 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

7.9.5 Disadvantages of KNN Algorithm

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

7.10 SUMMARY

A subset of machine learning and artificial intelligence is supervised learning, commonly referred to as supervised machine learning. It is defined by its use of labelled datasets to train algorithms that to classify data or predict outcomes effectively.The most widely used machine learning algorithm is supervised learning since it is simple to comprehend and apply. The model uses labelled data and variables as inputs to get reliable results.Building an artificial system that can learn the relationship between the input and the output and anticipate the system's output given new inputs is the aim of supervised learning. We have also covered several supervised learning algorithms along with its working and delved into its various fundamental's aspects affecting the performance.

7.11 LIST OF REFERENCES

1] Doing Data Science, Rachel Schutt and Cathy O'Neil, O'Reilly,2013.

2] Mastering Machine Learning with R, Cory Lesmeister, PACKT Publication,2015.

3] Hands-On Programming with R, Garrett Grolemund,1st Edition, 2014.

4] An Introduction to Statistical Learning, James, G., Witten, D., Hastie, T., Tibshirani, R.,Springer,2015.

7.12 UNIT END EXERCISES

1] Explain the concept oflinear models.

2] State the types of linear model.

3] Illustrate the applications of linear model.

4] What are regression trees?

5] Explain the steps involved in building a regression tree.

6] What is time-series Analysis?

7] Explain the term Forecasting.

8] What is classification trees?

9] What do you mean by logistic regression?

10] Describe the classification process using separating hyperplanes.

11] Explain the k-NN algorithm in detail.

❖❖❖❖

8

UNSUPERVISED LEARNING

Unit Structure

8.0 Objectives

8.1 Introduction

8.2 Principal Components Analysis (PCA)

8.2.1 Principal components in PCA

8.2.2 Steps for PCA algorithm

8.2.3 Applications of PCA

8.3 k-means clustering

8.3.1 k-means algorithm

8.3.2 Working of k-means algorithm

8.4 Hierarchical clustering

8.5 Ensemble methods

8.5.1 Categories of ensemble methods

8.5.2 Main types of ensemble methods

8.6 Summary

8.7 List of References

8.8 Unit End Exercises

8.0 OBJECTIVES

- To get familiar with the fundamentals and principles involved in unsupervised learning
- To get acquaint with the different algorithms associated with the unsupervised learning

8.1 INTRODUCTION

As the name suggests, unsupervised learning is a machine learning technique in which models are not supervised using training dataset. Instead, models itself find the hidden patterns and insights from the given data. It can be compared to learning which takes place in the human brain while learning new things. It can be defined as:

"Unsupervised learning is a type of machine learning in which models are trained using unlabeled dataset and are allowed to act on that data without any supervision"

As unlike supervised learning, we have the input data but no corresponding output data, unsupervised learning cannot be used to solve a regression or classification problem directly. Finding the underlying structure of a dataset, classifying the data into groups based on similarities, and representing the dataset in a compressed format are the objectives of unsupervised learning.

Consider the following scenario: An input dataset including photos of various breeds of cats and dogs is provided to the unsupervised learning algorithm. The algorithm is never trained on the provided dataset;thus, it has no knowledge of its characteristics. The unsupervised learning algorithm's job is to let the image features speak for themselves. This work will be carried out by an unsupervised learning algorithm, which will cluster the image collection into groups based on visual similarities.

The following are a few key arguments for the significance of unsupervised learning:

- Finding valuable insights from the data is made easier with the aid of unsupervised learning.

- Unsupervised learning is considerably more like how humans learn to think via their own experiences, which brings it closer to actual artificial intelligence.

- Unsupervised learning is more significant because it operates on unlabeled and uncategorized data.

- Unsupervised learning is necessary to handle situations when the input and output are not always the same in the real world.

8.2 PRINCIPAL COMPONENTS ANALYSIS (PCA)

An unsupervised learning approach called principal component analysis is used in machine learning to reduce dimensionality. With the use of orthogonal transformation, it is a statistical process that transforms the observations of correlated features into a set of linearly uncorrelated data. The Main Components are these newly altered features. One of the widely used tools for exploratory data analysis and predictive modelling is this one. It is a method for identifying significant patterns in the provided dataset by lowering the variances.

Typically, PCA looks for the surface with the lowest dimensionality onto which to project the high-dimensional data.

PCA functions by taking into account each attribute's variance since a high attribute demonstrates a solid split between classes, which lowers the dimensionality. Image processing, movie recommendation systems, and power allocation optimization in multiple communication channels are some examples of PCA's practical uses. Since it uses a feature extraction technique, it keeps the crucial variables and discards the unimportant ones.

The PCA algorithm is founded on mathematical ideas like:

- Variance and covariance
- Eigen values and eigen factors

Some common terms used in PCA algorithm:

- **Dimensionality:** It is the number of features or variables present in the given dataset. More easily, it is the number of columns present in the dataset.
- **Correlation:** It signifies that how strongly two variables are related to each other. Such as if one changes, the other variable also gets changed. The correlation value ranges from -1 to +1. Here, -1 occurs if variables are inversely proportional to each other, and +1 indicates that variables are directly proportional to each other.
- **Orthogonal:** It defines that variables are not correlated to each other, and hence the correlation between the pair of variables is zero.
- **Eigenvectors:** If there is a square matrix M, and a non-zero vector v is given. Then v will be eigenvector if Av is the scalar multiple of v.
- **Covariance Matrix:** A matrix containing the covariance between the pair of variables is called the Covariance Matrix.

8.2.1 Principal components in PCA

The Principal Components are the newly altered characteristics or the result of PCA, as previously said. These PCs are either the same number or fewer than the initial characteristics that were included in the dataset. Following are a few characteristics of these primary components:

- The linear combination of the unique traits must be the major component.
- Because these components are orthogonal, there is no association between any two variables.
- Going from 1 to n, the importance of each component declines, making PC-1 the most important and PC-n the least important.

8.2.2 Steps for PCA algorithm

1] Obtaining the dataset

Firstly, we need to take the input dataset and divide it into two halves, X and Y, where X represents the training set and Y represents the validation set.

2] Data representation in a structure

We will now create a structure to represent our dataset. We'll use the two-dimensional matrix of independent variable X as an example. Thus, each row represents a data item and each column represents a feature. The dataset's dimensions are determined by the number of columns.

3.] Data standardization

We will normalize our dataset in this stage. For instance, in a given column, features with higher variation are more significant than features with smaller variance.We shall split each piece of data in a column by the column's standard deviation if the importance of features is independent of the variance of the feature. The matrix in this case will be called Z.

4] Determining Z's Covariance

We shall transpose the Z matrix in order to determine Z's covariance. Transposing it first, we'll multiply it by Z. The Covariance matrix of Z will be the output matrix.

5] Calculating the Eigen Values and Eigen Vectors

The resulting covariance matrix Z's eigenvalues and eigenvectors must now be determined. The high information axis' directions are represented by eigenvectors or the covariance matrix. Moreover, the eigenvalues are defined as the coefficients of these eigenvectors.

6] Sorting the Eigen Vectors.

This phase involves taking all of the eigenvalues and sorting them from largest to lowest in a decreasing order. Also, in the eigenvalues matrix P, simultaneously sort the eigenvectors in accordance. The matrix that results will be known as P*.

7] Figuring out the new features or Primary Constituents

We will compute the new features here. We'll multiply the P* matrix by Z to achieve this. Each observation in the resulting matrix Z* is the linear combination of the original features. The Z* matrix's columns are independent of one another.

8] Remove less significant or irrelevant features from the new dataset.

We will determine here what to keep and what to eliminate now that the new feature set has been implemented. It indicates that we will only retain relevant or significant features in the new dataset and will exclude irrelevant information.

8.2.3 Applications of PCA

- PCA is primarily utilized as a dimensionality reduction technique in a variety of AI applications, including image compression and computer vision.

- If the data has a high dimension, it can also be used to uncover hidden patterns. Banking, data mining, psychology, and other industries are just a few ways PCA is applied.

8.3 K-MEANS CLUSTERING

The clustering issues in machine learning or data science are resolved using the unsupervised learning algorithm K-Means Clustering.

8.3.1 k-means algorithm

Unsupervised learning algorithm K-Means Clustering divides the unlabeled dataset into various clusters. Here, K specifies how many pre-defined clusters must be produced as part of the process; for example, if K=2, there will be two clusters, if K=3, there will be three clusters, and so on.

The unlabeled dataset is divided into k separate clusters using an iterative process, and each dataset is only a part of one group that shares characteristics with the others.

It gives us the ability to divide the data into various groups and provides a practical method for automatically identifying the groups in the unlabeled dataset without the need for any training.

Each cluster has a centroid assigned to it because the algorithm is centroid-based. This algorithm's primary goal is to reduce the total distances between each data point and its corresponding clusters.

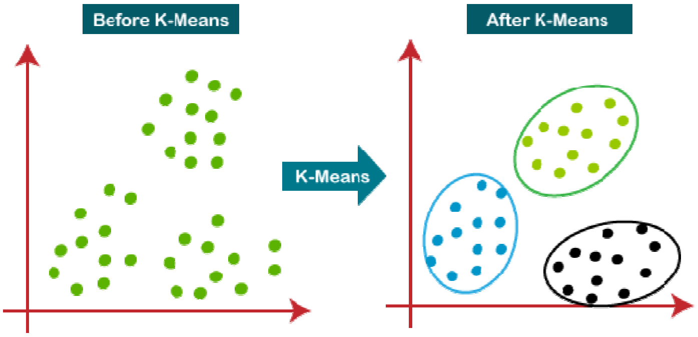
The algorithm starts with an unlabeled dataset as its input, separates it into k clusters, and then continues the procedure until it runs out of clusters to use. In this algorithm, the value of k should be predetermined.

The two major functions of the k-means clustering algorithm are:

- Uses an iterative technique to choose the best value for K centre points or centroids.

- Each data point is matched with the nearest k-center. A cluster is formed by the data points that are close to a specific k-center.As a result, each cluster is distinct from the others and contains datapoints with some commonality.

The K-means Clustering Algorithm is explained in the diagram below:



8.3.2 Working of k-means algorithm

The following stages illustrate how the K-Means algorithm functions:

Step 1: To determine the number of clusters, choose K.

Step-2: Pick random K locations or centroids. That might not be the input dataset.

Step 3: Assign each data point to its nearest centroid, which will create the K clusters that have been predetermined.

Step 4: Determine the variance and relocate each cluster's centroid.

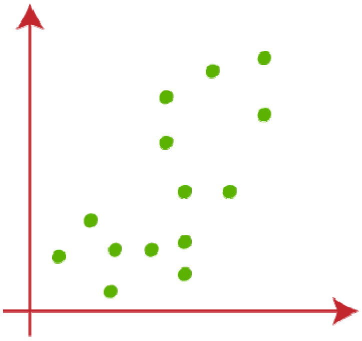
Step 5: Re-assign each data point to the new centroid of each cluster by repeating the third step.

Step 6: Go to step 4 if there is a reassignment; otherwise, move to FINISH.

Step 7: The model is finished.

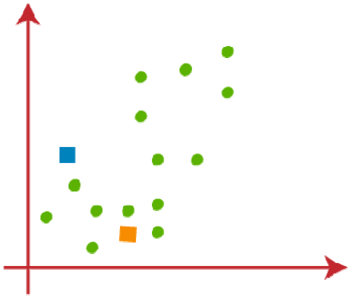
Let's analyze the visual plots in order to comprehend the aforementioned steps:

Consider that there are two variables, M1 and M2. The following shows the x-y axis scatter plot of these two variables:

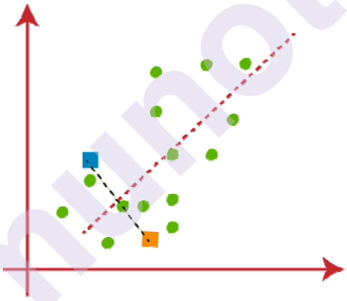


- Let's take number k of clusters, i.e., K=2, to identify the dataset and to put them into different clusters. It means here we will try to group these datasets into two different clusters.

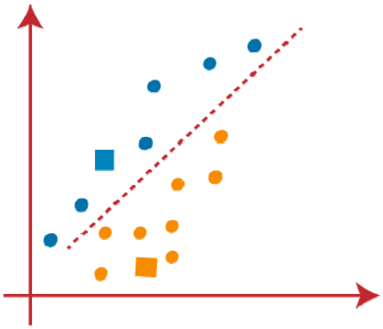
- We need to choose some random k points or centroid to form the cluster. These points can be either the points from the dataset or any other point. So, here we are selecting the below two points as k points, which are not the part of our dataset. Consider the belowimage:



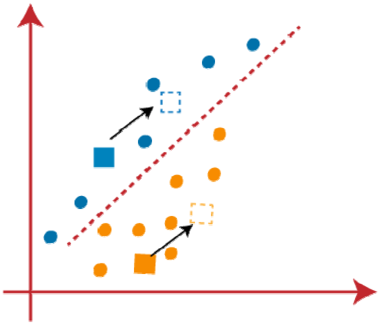
- Now we will assign each data point of the scatter plot to its closest K-point or centroid. We will compute it by applying some mathematics that we have studied to calculate the distance between two points. So, we will draw a median between boththe centroids. Consider the below image:



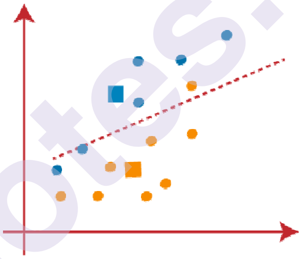
From the above image, it is clear that points left side of the line is near to the K1 or blue centroid, and points to the right of the line are close to the yellow centroid. Let's color them as blue and yellow for clear visualization.



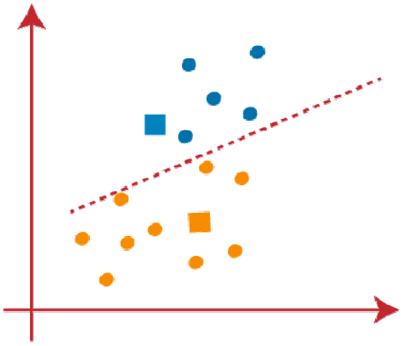
- As we need to find the closest cluster, so we will repeat the process by choosing **a new centroid**. To choose the new centroids, we will compute the center of gravity ofthese centroids, and will find new centroids as below:



- Next, we will reassign each datapoint to the new centroid. For this, we will repeat the same process of finding a median line. The median will be like below image:

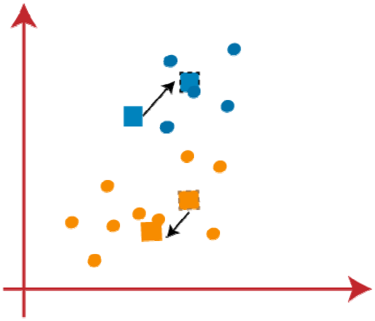


- From the above image, we can see, one yellow point is on the left side of the line, and two blue points are right to the line. So, these three points will be assigned to new centroids.

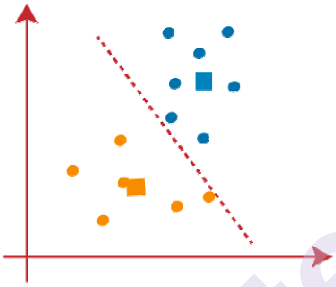


As reassignment has taken place, so we will again go to the step-4, which is finding new centroids or K-points.

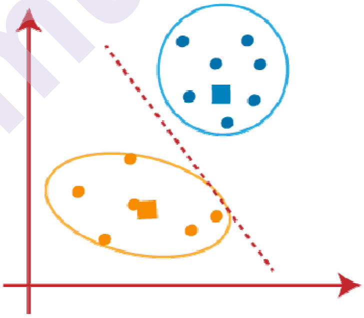
- We will repeat the process by finding the center of gravity of centroids, so the new centroids will be as shown in the below image:



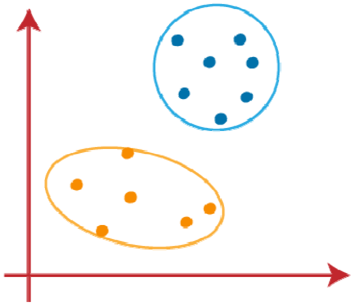
- As we got the new centroids so again will draw the median line and reassign the data points. So, the image will be:



- We can see in the above image; there are no dissimilar data points on either side ofthe line, which means our model is formed. Consider the below image:



As our model is ready, so we can now remove the assumed centroids, and the two final clusters will be as shown in the below image:



8.4 HIERARCHICAL CLUSTERING

Data are grouped into groups in a tree structure in a hierarchical clustering method. Every data point is first treated as a separate cluster in a hierarchical clustering process. The following steps are then repeatedly carried out by it:

Choose the two clusters that are the closest to one another, and then combine the two clusters that are the most similar. These procedures must be repeated until all of the clusters are combined.

The goal of hierarchical clustering is to create a hierarchy of nested clusters. a Dendrogram, a type of graph (A Dendrogram is a tree-like diagram that statistics the sequences of merges or splits) depicts this hierarchy graphically and is an inverted tree that explains the sequence in which elements are combined (bottom-up view) or clusters are dispersed (top-down view).

A data mining technique called hierarchical clustering builds a hierarchical representation of the clusters in a dataset. Each data point is initially treated as an independent cluster, and from there, the algorithm iteratively aggregates the nearest clusters until a stopping requirement is met. A dendrogram - a tree-like structure that shows the hierarchical links between the clustersis the outcome of hierarchical clustering.

Compared to other clustering techniques, hierarchical clustering has a variety of benefits, such as:

1. The capacity for non-convex clusters as well as clusters of various densities and sizes.
2. The capacity to deal with noisy and missing data.
3. The capacity to display the data's hierarchical structure, which is useful for comprehending the connections between the clusters.

It does, however, have several shortcomings, such as:

1. The requirement for a threshold to halt clustering and establish the total number of clusters.

2. The approach can have high processing costs and memory needs, particularly for huge datasets.

3. The initial conditions, linkage criterion, and distance metric can have an impact on the outcomes.

   In conclusion, hierarchical clustering is a data mining technique that groups related data points into clusters by giving the clusters a hierarchical structure.

4. This technique can handle various data formats and show the connections between the clusters. Unfortunately, the results could be sensitive to certain circumstances and have a large computational cost.

1. **Agglomerative:** At first, treat each data point as a separate cluster. Next, at each step, combine the cluster's closest pairs. It uses a bottom-up approach. Every dataset is first viewed as a distinct entity or cluster. The clusters combine with other clusters at each iteration until only one cluster remains.

Agglomerative Hierarchical Clustering uses the following algorithm:

- Determine how similar each cluster is to each of the other clusters (calculate proximity matrix)
- Think of each data point as a separate cluster.
- Combine the groups that are quite similar to one another or those are nearby.
- For each cluster, recalculate the proximity matrix.
- Once there is just one cluster left, repeat steps 3 and 4 as necessary.

Let's look at this algorithm's visual representation using a dendrogram.

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

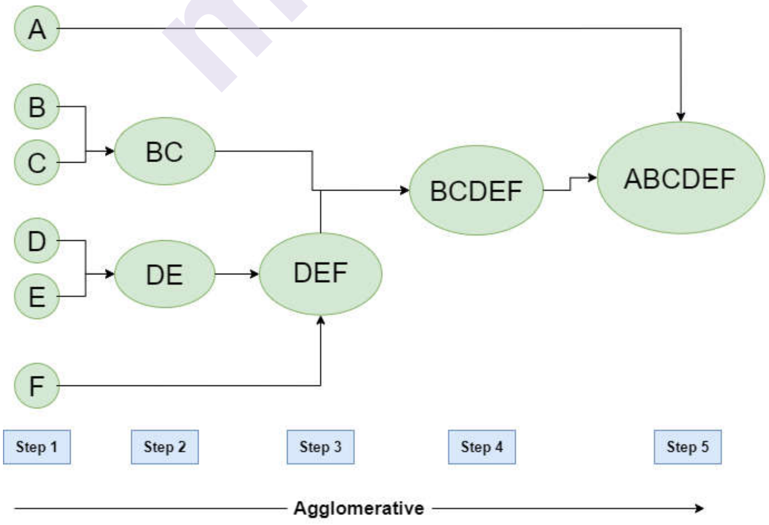


Figure: Agglomerative Hierarchical clustering

- **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 Divisive Hierarchical clustering is precisely the **opposite** of 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.

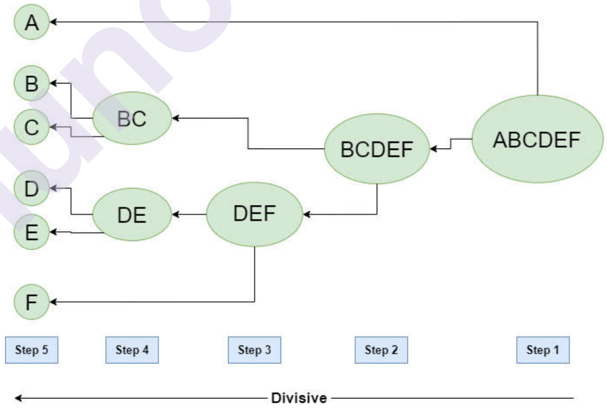
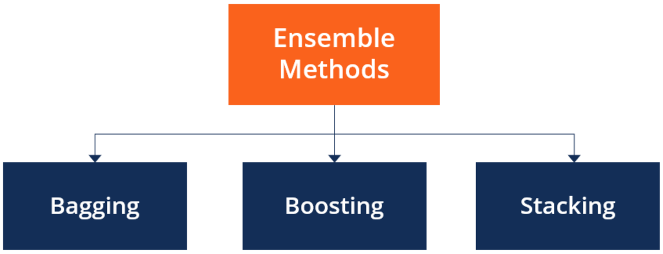


Figure: Divisive Hierarchical clustering

8.5 ENSEMBLE METHODS

A machine learning technique called ensemble techniques combines multiple base models to create a single, ideal predictive model.By mixing numerous models rather than relying just on one, ensemble approaches seek to increase the accuracy of findings in models. The integrated models considerably improve the results' accuracy. Due of this, ensemble approaches in machine learning have gained prominence.



8.5.1 Categories of ensemble methods

Sequential ensemble techniques and parallel ensemble techniques are the two main categories into which ensemble methods belong. Base learners are produced via sequential ensemble approaches, such as adaptive boosting (AdaBoost). The dependency between the base learners is encouraged by their consecutive generation. The model's performance is then enhanced by giving previously misrepresented learners more weight.

Base learners are created in a parallel fashion, such as random forest, in parallel ensemble approaches. To promote independence among the basis learners, parallel techniques make use of parallel generations of base learners. The mistake resulting from the use of averages is greatly decreased by the independence of base learners.

The majority of ensemble techniques only use one algorithm for base learning, which makes all base learners homogeneous. Base learners who have comparable traits and are of the same type are referred to as homogenous base learners. Some approaches create heterogeneous ensembles by using heterogeneous base learners. Many sorts of learners make up heterogeneous base learners.

8.5.2 Main types of ensemble methods

1] Bagging

Bootstrap aggregating is commonly used in classification and regression, and also known as bagging. Using decision trees, it improves the models' accuracy, greatly reducing variation. Many prediction models struggle with overfitting, which is eliminated by reducing variation and improving accuracy.

Bootstrapping and aggregation are the two categories under which bagging is categorized. Bootstrapping is a sampling strategy where samples are taken utilizing the replacement procedure from the entire population (set). The sampling with replacement method aids in the randomization of the selection process. The process is finished by applying the base learning algorithm to the samples.

In bagging, aggregation is used to include all potential outcomes of the prediction and randomize the result. Predictions made without aggregation won't be accurate because all possible outcomes won't be taken into account. As a result, the aggregate is based either on all of the results from the predictive models or on the probability bootstrapping techniques.

Bagging is useful because it creates a single strong learner that is more stable than individual weak base learners. Moreover, it gets rid of any variance, which lessens overfitting in models. The computational cost of bagging is one of its drawbacks. Hence, ignoring the correct bagging technique can result in higher bias in models.

2] Boosting

Boosting is an ensemble strategy that improves future predictions by learning from previous predictor errors. The method greatly increases model predictability by combining numerous weak base learners into one strong learner. Boosting works by placing weak learners in a sequential order so that they can learn from the subsequent learner to improve their predictive models.

There are many different types of boosting, such as gradient boosting, Adaptive Boosting (AdaBoost), and XGBoost (Extreme Gradient Boosting). AdaBoost employs weak learners in the form of decision trees, the majority of which include a single split known as a decision stump. The primary decision stump in AdaBoost consists of observations with equal weights.

Gradient boosting increases the ensemble's predictors in a progressive manner, allowing earlier forecasters to correct later ones, improving the model's accuracy. To offset the consequences of errors in the earlier models, new predictors are fitted. The gradient booster can identify and address issues with learners' predictions thanks to the gradient of descent.

Decision trees with boosted gradients are used in XGBoost, which offers faster performance. It largely depends on the goal model's efficiency and effectiveness in terms of computing. Gradient boosted machines must be implemented slowly since model training must proceed sequentially.

3] Stacking

Another ensemble method called stacking is sometimes known as layered generalization. This method works by allowing a training algorithm to combine the predictions of numerous different learning algorithms that are similar. Regression, density estimations, distance learning, and classifications have all effectively used stacking. It can also be used to gauge the amount of inaccuracy that occurs when bagging.

8.6 SUMMARY

Any machine learning challenge aims to choose a single model that can most accurately forecast the desired result. Ensemble approaches consider a wide range of models and average those models to build one final model, as opposed to creating one model and hoping that this model is the best/most accurate predictor we can make.

Unsupervised learning, commonly referred to as unsupervised machine learning, analyses and groups unlabeled datasets using machine learning algorithms. These algorithms identify hidden patterns or data clusters without the assistance of a human.

Unsupervised learning's main objective is to find hidden and intriguing patterns in unlabeled data. Unsupervised learning techniques, in contrast to supervised learning, cannot be used to solve a regression or classification problem directly because it is unknown what the output values will be. We have also studied different techniques and algorithms for classification and to boost the performance.

In conclusion, the unsupervised learning algorithms allow you to accomplish more complex processing jobs. There are various benefits of unsupervised learnings such as it is taken in place issue solving time, hence all of the input data which is to be examined and categorized in the appearance of learners.

8.7 LIST OF REFERENCES

1] Doing Data Science, Rachel Schutt and Cathy O'Neil, O'Reilly,2013.

2] Mastering Machine Learning with R, Cory Lesmeister, PACKT Publication,2015.

3] Hands-On Programming with R, Garrett Grolemund,1st Edition, 2014.

4] An Introduction to Statistical Learning, James, G., Witten, D., Hastie, T., Tibshirani, R.,Springer,2015.

8.8 UNIT END EXERCISES

1] Explain the Principal Components Analysis (PCA).

2] What are the principal components in PCA?

3] Explain the steps involved and applications for PCA algorithm.

4] Describe the k-means clustering.

5] Explain the working of k-means algorithm.

6] Write a note on Hierarchical clustering.

7] Explain Ensemble methods.

8] What are the categories of ensemble methods?

9] Describe the main types of ensemble methods.

❖❖❖❖