SPPU New Syllabus

A Book Of

BIG DATA

For BBA (Computer Applications): Semester - III

[Course Code CA-305]

CBCS Pattern

As Per New Syllabus, Effective from June 2020

Mr. Kamil Ajmal Khan

Assistant Professor

B.C.A./M.C.A. Department

Abeda Inamadar Senior College,

Camp, Pune-01

Price ₹ 160.00

NIRALI PRAKASHAN

ADVANCEMENT OF KNOWLEDGE

N4

BIG DATA

First Edition : July 2020 ISBN 978-93-90225-51-4

Published By :
NIRALI PRAKASHAN
Abhyudaya Pragati, 1312, Shivaji Nagar
Off J.M. Road, PUNE – 411005
Tel - (020) 25512336/37/39, Fax - (020) 25511379
Email : niralipune@pragationline.com

Printed By :
YOGIRAJ PRINTERS AND BINDERS
Survey No. 10/1A, Ghule Industrial Estate
Nanded Gaon Road
Nanded, Pune - 411041
Mobile No. 9404233041/9850046517

DISTRIBUTION CENTRES

PUNE

Nirali Prakashan : 119, Budhwar Peth, Jogeshwari Mandir Lane, Pune 411002, Maharashtra
(For orders within Pune) Tel : (020) 2445 2044, Mobile : 9657703145
Email : niralilocal@pragationline.com

Nirali Prakashan : S. No. 28/27, Dhayari, Near Asian College Pune 411041
(For orders outside Pune) Tel : (020) 24690204 Fax : (020) 24690316; Mobile : 9657703143
Email : bookorder@pragationline.com

MUMBAI

Nirali Prakashan : 385, S.V.P. Road, Rasdhara Co-op. Hsg. Society Ltd.,
Girgaum, Mumbai 400004, Maharashtra; Mobile : 9320129587
Tel : (022) 2385 6339 / 2386 9976, Fax : (022) 2386 9976
Email : niralimumbai@pragationline.com

DISTRIBUTION BRANCHES

JALGAON

Nirali Prakashan : 34, V. V. Golani Market, Navi Peth, Jalgaon 425001, Maharashtra,
Tel : (0257) 222 0395, Mob : 94234 91860; Email : niralijalgaon@pragationline.com

KOLHAPUR

Nirali Prakashan : New Mahadvar Road, Kedar Plaza, 1st Floor Opp. IDBI Bank, Kolhapur 416 012
Maharashtra. Mob : 9850046155; Email : niralikolhapur@pragationline.com

NAGPUR

Nirali Prakashan : Above Maratha Mandir, Shop No. 3, First Floor,
Rani Jhanshi Square, Sitabuldi, Nagpur 440012, Maharashtra
Tel : (0712) 254 7129; Email : niralinagpur@pragationline.com

DELHI

Nirali Prakashan : 4593/15, Basement, Agarwal Lane, Ansari Road, Daryaganj
Near Times of India Building, New Delhi 110002 Mob : 08505972553
Email : niralidelhi@pragationline.com

BENGALURU

Nirali Prakashan : Maitri Ground Floor, Jaya Apartments, No. 99, 6th Cross, 6th Main,
Malleswaram, Bengaluru 560003, Karnataka; Mob : 9449043034
Email: niralibangalore@pragationline.com

Other Branches : Hyderabad, Chennai

niralipune@pragationline.com | www.pragationline.com

Also find us on www.facebook.com/niralibooks

Preface ...

I take an opportunity to present this book entitled as **"Big Data"** to the students of **B.B.A.(Computer Applications), Semester-III** as per the New Syllabus (CBCS Pattern), June 2020.

The book covers theory of Introduction to Big Data, Introduction to Data Science, Introduction to Machine Learning, Data Analytics with R/ Weka Machine Learning.

A special words of thank to Shri. Dineshbhai Furia, Mr. Jignesh Furia for showing full faith in me to write this text book. We also thank to Mrs. Anita Panajkar and Mrs. Prachi Sawant of M/s Nirali Prakashan for their excellent co-operation.

I also thank Mr. Ravindra Walodare, Mr. Sachin Shinde, Mr. Nilesh Deshmukh, Mr. Ashok Bodke, Mr. Moshin Sayyed and Mr. Nitin Thorat.

Although every care has been taken to check mistakes and misprints, any errors, omission and suggestions from teachers and students for the improvement of this text book shall be most welcome.

Author

Syllabus ...

1. Introduction to Big Data (5 Hrs.)

1.1 Introduction to Big Data

1.2 Types of Digital Data

1.3 Big Data Analytics

1.4 Application of Big data

2. Introduction to Data Science (10 Hrs.)

2.1 Basics of Data Analytics

2.2 Types of Analytics –

- 2.2.1 Descriptive,
- 2.2.2 Predictive,
- 2.2.3 Prescriptive
- 2.2.4 Statistical Inference

2.3 Populations and samples

- 2.3.1 Statistical modelling,
- 2.3.2 Probability
- 2.3.3 Distribution
- 2.3.4 Correlation
- 2.3.5 Regression

3. Introduction to Machine Learning (20 Hrs.)

3.1 Basics of Machine Leaning

3.2 Supervised Machine Learning

- 3.2.1 K- Nearest-Neighbours,
- 3.2.2 Naïve Bayes
- 3.2.3 Decision tree
- 3.2.4 Support Vector Machines

3.3 Unsupervised Machine Learning

- 3.3.1 Cluster analysis
- 3.3.2 K-means

❖❖❖

Contents ...

❖❖❖

1... Introduction to Big Data

Objectives...

- To get familiar with the fundamental concept of Big Data.
- To understand the Big Data challenges.
- To know about Digital Data.
- To get information about Big Data applications.

1.1 INTRODUCTION TO BIG DATA

- Big Data is creating new contingency for organizations to derive new value and create competitive advantage from their most valuable asset: information. In many cases, Big Data analytics merging structured and unstructured data with real time feeds and queries, opening new paths to innovation and insight.
- For businesses, Big Data helps drive efficiency, quality, and personalized products and services, producing improved levels of customer satisfaction and profit. For scientific efforts, Big Data analytics enables new paths of investigation with potentially richer results and deeper insights than previously available.
  - Since 1990s the term 'Big Data' is used. John Mashey popularize this term. Big data usually added data sets with sizes beyond the ability of commonly used software tools to capture, organize, manage, and process data within a tolerable elapsed time. Big data philosophy includes unstructured, semi-structured and structured data. However it focuses on unstructured data. Big data "size" is a constantly moving target, as of 2012 ranging from a few dozen terabytes to many zettabytes of data.
  - In Big data, many concepts are related. Basically there were 3 concepts: Volume, Variety, and Velocity. Data storage, search, transfer, visualization, sharing, querying, capturing data, information privacy, updating, and data analysis and data source all these things included in Big Data challenges.

- Large and varied set of data can be processed by big data analytics. Big data to show hidden patterns, unknown correlations, customer preferences, market trends and other useful information that can help organizations make more-informed business decisions. Driven by specialized analytics systems and software, big data analytics can point the way to various business benefits, including new revenue opportunities, more effective marketing, better customer service, improved operational efficiency and competitive advantages over rivals.

1.1.1 Big Data Concepts

What is Big Data?

- The data lying in the servers of your company was just data until yesterday – sorted and filed. Suddenly, the slang Big Data got popular, and now the data in your company is Big Data. The term covers each and every piece of data your organization has stored till now. It includes data stored in clouds and even the URLs that you bookmarked. Your company might not have digitized all the data. You may not have structured all the data already. But then, all the digital, papers, structured and non-structured data with your company is now Big Data.
- In short, all the data whether or not categorized – present in your servers is collectively called BIG DATA. All this data can be used to get different results using different types of analysis. It is not necessary that all analysis use all the data. The different analysis uses different parts of the BIG DATA to produce the results and predictions necessary.
- An analysis of these definitions shows that all of them highlight the hugeness of the data which the term implies, its variety and rate of change and that the data need to be analyzed to gain insight into what it implies.
- In addition, the analysis needs new tools and huge computing resources. In Data Science, big data analytics is an important component that has helped in the fourth paradigm in science.
- Two earlier paradigms were Theoretical Science and Experimental Science.
- This was followed in the last two decades by a third paradigm, Simulation. The advent of high performance computer simulations reduced the number of experiments conducted to authenticate a hypothesis and led to the field of computational science.
- The latest, namely, the fourth paradigm has been enabled by Data Science. The fourth paradigm is data-driven discovery of new hypotheses which may lead to the formulation of novel theories.

What's Unique about Big Data?

- Companies have searched for decades to make the best use of information to improve their business capabilities. However, it's the structure (or lack thereof) and size of Big Data that makes it so unique.
- Big Data is also special because it represents both significant information - which can open new doors - and the way this information is analyzed to help open those doors. The analysis goes hand-in-hand with the information, so in this sense "Big Data" represents a noun - "the data" - and a verb - "combing the data to find value."

How can we make sense of Big Data?

- Interpretation of Big Data can bring about insights which might not be immediately visible or which would be impossible to find using traditional methods. This process focuses on finding hidden threads, trends, or patterns which may be invisible to the naked eye. It requires new technologies and skills to analyze the flow of material and draw conclusions.
- Just as Big Data can be both a noun and a verb, Hadoop involves specifically data storage and data processing. Both of these occur in a distributed manner to improve efficiency and results. Apache Hadoop is one such technology, and it is generally the software most commonly associated with Big Data. Apache calls it "a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models."
- *"Big data is a term that is used to describe data that is high volume, high velocity, and/or high variety; requires new technologies and techniques to capture, store, and analyze it; and is used to enhance decision making, provide insight and discovery, and support and optimize processes."*

1.1.2 Advantages and Disadvantages of Big Data

Advantages of Big Data:

1. Big data analysis derives innovative solutions. Big data analysis helps in understanding and targeting customers. It helps in optimizing business processes.
2. It helps in improving science and research.
3. It improves healthcare and public health with availability of record of patients.
4. It helps in financial trading's, sports, polling, security/law enforcement etc.
5. Anyone can access vast information via surveys and deliver answers of any query.
6. Every second additions are made.
7. One platform carries unlimited information.

Disadvantages of Big Data:

1. Traditional storage can cost lot of money to store big data.
2. Lots of big data is unstructured.
3. Big data analysis violates principles of privacy.
4. It can be used for manipulation of customer records.
5. It may increase social stratification.
6. Big data analysis is not useful in short run. It needs to be analyzed for longer duration to leverage its benefits.
7. Big data analysis results are misleading sometimes.
8. Speedy updates in big data can mismatch real figures.

1.1.3 5 'V's of Big Data

- The term big data emphasizes volume or size. Size is a relative term. In the 1960s, 20 Megabytes was considered large. Now data is not considered big unless it is several hundred Petabytes (PB) (Petabyte = 10^{15} bytes). Size is not the only property used to describe big data.
- In addition to volume, there are other important properties that we will discuss in what follows:

Volume:

- Amount of global digital data created, replicated, and consumed in 2013 was estimated by the International Data Corporation (a company which publishes research reports) as 4.4 Zettabytes (ZB) (Zettabyte = 10^{21}bytes). It is doubling every 2 years.
- By 2015, digital data grew to 8 ZB and is expected to grow to 12 ZB in 2016. To give an idea of a ZB, it is the storage required to store 200 billion high definition movies which will take a person 24 million years to watch!

Variety:

- In the 1960s, the predominant data types were numbers and text. Today, in addition to numbers and text, there are image, audio, and video data. Large Hadron Collider (LHC), earth and polar observations generate mainly numeric data. Word processors, emails, tweets, blogs, and other social media generate primarily unstructured textual data.
- Medical images and billions of photographs which people take using their mobile phones are image data. Surveillance cameras and movies produce video data. Music sites store audio data. Most data in the 80s were structured and organized as tables with keys. Today there are unstructured and multimedia data often used together.

Velocity:

- Data in conventional databases used to change slowly. Now most data are real time. For example, phone conversations, data acquired from experiments, data sent by sensors, data exchanged using the Internet, and stock price data are all real time. Large amounts of data are transient and need to be analyzed as and when they are generated. They become irrelevant fast.

Veracity:

- A lot of data generated are noisy, e.g., data from sensors. Data are often incorrect. For example, many websites you access may not have the correct information. It is difficult to be absolutely certain about the veracity of big data.

Value:

- Data by itself is of no value unless it is processed to obtain information using which one may initiate actions. The large volume of data makes processing difficult. Fortunately, computing power and storage capacity have also increased enormously.
- A huge number of inexpensive processors working in parallel have made it feasible to extract useful information to detect patterns from big data. Distributed file systems such as Hadoop Distributed File System (HDFS) coupled with parallel processing programs such as Map Reduce are associated with big data as software tools to derive value from big data.

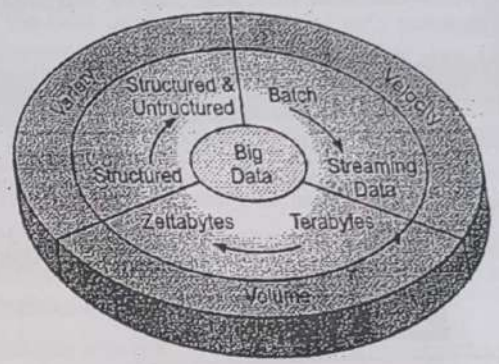


Fig. 1.1: Characteristics of Big Data

1.1.4 Big Data Sources

- Big data has many sources. For example, every mouse click on a web site can be captured in Web log files and analyzed in order to better understand shoppers' buying behaviors and to influence their shopping by dynamically recommending products.

- Social media sources such as Facebook and Twitter generate tremendous amounts of comments and tweets. This data can be captured and analyzed to understand, for example, what people think about new product introductions.
- Machines, such as smart meters, generate data. These meters continuously stream data about electricity, water, or gas consumption that can be shared with customers and combined with pricing plans to motivate customers to move some of their energy consumption, such as for washing clothes, to non-peak hours.
- There is a tremendous amount of geospatial (e.g., GPS) data, such as that created by cell phones, that can be used by applications like Four Square to help you know the locations of friends and to receive offers from nearby stores and restaurants.
- Image, voice, and audio data can be analyzed for applications such as facial recognition systems in security systems.

1.1.5 Tools used for Big Data

- **NoSQL:** Databases MongoDB, CouchDB, Cassandra, Redis, BigTable, Hbase, Hypertable, Voldemort, Riak, ZooKeeper.
- **MapReduce:** Hadoop, Hive, Pig, Cascading, Cascalog, mrjob, Caffeine, S4, MapR, Acunu, Flume, Kafka, Azkaban, Oozie, Greenplum.
- **Storage:** S3, Hadoop Distributed File System.
- **Server:** EC2, Google App Engine, Elastic, Beanstalk, Heroku.
- **Processing:** R, Yahoo! Pipes, Mechanical Turk, Solr/Lucene, ElasticSearch, Datameer, BigSheets, Tinkerpop.

1.2 TYPES OF DIGITAL DATA

- Digital data, in information theory and information systems, is the discrete, discontinuous representation of information or works. Numbers and letters are commonly used representations.
- Digital data can be classified into three forms as shown in following figure.

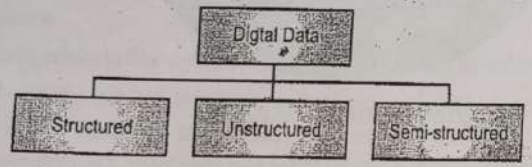


Fig. 1.2: Classification of Digital Data

(1) **Structured data** is data whose elements are addressable for effective analysis. It has been organized into a formatted repository that is typically a database. It concerns all data which can be stored in database SQL in a table with rows and

columns. They have relational keys and can easily be mapped into pre-designed fields. Today, those data are most processed in the development and simplest way to manage information. Example: Relational data.

Structured data depends on the existence of a data model – a model of how data can be stored, processed and accessed. Because of a data model, each field is discrete and can be accesses separately or jointly along with data from other fields. This makes structured data extremely powerful: it is possible to quickly aggregate data from various locations in the database.

(2) **Unstructured data** is a data that 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. So 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.

The ability to analyse unstructured data is especially relevant in the context of Big Data, since a large part of data in organisations is unstructured. Think about pictures, videos or PDF documents. The ability to extract value from unstructured data is one of main drivers behind the quick growth of Big Data.

(3) **Semi-structured data** is information that does not reside in a relational database but that have some organizational properties that make it easier to analyze. With some process, you can store them in the relation database (it could be very hard for some kind of semi-structured data), but Semi-structured exist to ease space. Example: XML data.

The reason that this third category exists (between structured and unstructured data) is because semi-structured data is considerably easier to analyse than unstructured data. Many Big Data solutions and tools have the ability to 'read' and process either JSON or XML. This reduces the complexity to analyse structured data, compared to unstructured data.

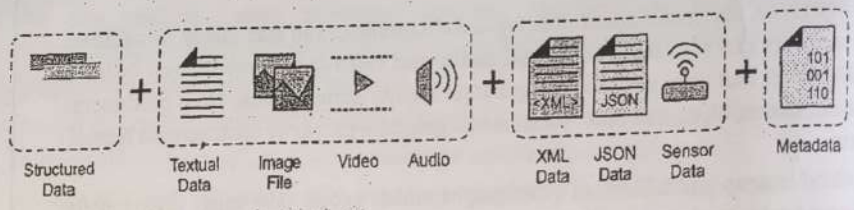


Fig. 1.3: Semi-structured Data

1.2.1 Differences between Structured, Semi-structured and Unstructured Data

Table 1.1: Comparison of types of digital data

Properties	Structured Data	Semi-Structured Data	Unstructured Data
Technology	It is based on Relational database table	It is based on XML/RDF	It is based on character and binary data
Transaction management	Matured transaction and various concurrency technique	Transaction is adapted from DBMS not matured	No transaction management and no concurrency
Version management	Versioning over tuples, row, tables	Versioning over tuples or graph is possible	Versioned as whole
Flexibility	It is schema dependent and less flexible	It is more flexible than structured data but less than flexible than unstructured data	it very flexible and there is absence of schema
Scalability	It is very difficult to scale DB schema	It's scaling is simpler than structured data	It is very scalable
Robustness	Very robust	New technology, not very spread	—
Query performance	Structured query allow complex joining	Queries over anonymous nodes are possible	Only textual query are possible

1.3 BIG DATA ANALYTICS

- By itself, stored data does not generate business value, and this is true of traditional databases, data warehouses, and the new technologies such as Hadoop for storing big data. Once the data is appropriately stored, however, it can be analyzed, which can create tremendous value. A variety of analysis technologies, approaches, and products have emerged that are especially applicable to big data, such as in-memory analytics, in-database analytics, and appliances.
- Big data analytics is the process of extracting useful information by analysing different types of big data sets. Big data analytics is used to discover hidden patterns, market trends and consumer preferences, for the benefit of organizational decision making.

1.3.1 Need of Big Data Analytics

- As one of the most "hyped" terms in the market today, there is no consensus as to how to define big data. The term is often used synonymously with related concept such as Business Intelligence (BI) and data mining.
- It is true that all three terms is about analyzing data and in many cases advanced analytics. But big data concept is different from the two others when data volumes, number of transactions and the number of data sources are so big and complex that they require special methods and technologies in order to draw insight out of data (for instance, traditional data warehouse solutions may fall short when dealing with big data).

1.3.2 Analyzing Big Data

- Data analytics is concerned with extraction of actionable knowledge and insights from big data. This is done by hypothesis formulation that is often based on assumptions gathered from experience and discovering (sometimes serendipitously) correlations among variables.
- There are four types of data analytics. They are:

(1) **Descriptive Analytics:** This analytics tells what happened in the past and presents it in an easily understandable form. Data gathered is organized as bar charts, line graphs, pie charts, maps, scatter diagrams, etc., for easy visualization which gives insight into what the data implies. This form of data presentation is often called a dash board, representing the dashboard of a car which gives information on speed, engine status, petrol left in the tank, distance travelled etc.

(2) **Predictive Analytics:** It extrapolates from available data and tells what is expected to happen in the near future. The tools used for extrapolation are time series analysis using statistical methods, neural networks, and machine learning algorithms.

One major use of predictive analytics is in marketing by comprehending customers' needs and preferences. An example is the advertisement on socks that appears when you buy shoes from an e-shop.

Another common use is in managing election campaigns by collecting variety of data such as composition of electorates in various locations, perception of their requirements such as infrastructure and local issues.

(3) **Exploratory or Discovery Analytics:** This finds unexpected relationships among parameters in collections of big data. Collection of data from a variety of sources and analyzing them provides additional opportunities for insights and unexpected discovery.

One of the major applications is discovering patterns in customers' behavior by companies using their feedback, tweets, blogs, Facebook data, emails, sales trends etc. Based on the customers' behavior it may be possible for companies to predict their actions such as renewing magazine subscription, changing mobile phone service provider, cancelling a hotel reservation. A company may then come up with an attractive offer to try and change the customer's anticipated action.

(4) **Prescriptive analytics:** This identifies, based on data gathered, opportunities to optimize solutions to existing problems. In other words, the analysis tells us what to do to achieve a goal. One of the common uses is in airlines' pricing of seats based on historical data of travel patterns, popular origins and destinations, major events, holidays, etc., to maximize profit.

1.4 BIG DATA APPLICATIONS

- The application of big data analytics in various sectors is discussed as follows:

(1) Healthcare:

- Data analysts obtain and analyze information from multiple sources to gain insights. The multiple sources are electronic patient record; clinical decision support system including medical imaging, physician's written notes and prescription, pharmacy and laboratories; clinical data; and machine generated sensor data.
- The integration of clinical, public health and behavioral data helps to develop a robust treatment system, which can reduce the cost and at the same time, improve the quality of treatment.

(2) Telecommunication:

- Low adoption of mobile services and Churn Management (a term that describes an operator's process to retain profitable customers) are few of the most common problems faced by the Mobile Service Providers (MSPs). The cost of acquiring new customer is higher than retaining the existing ones. Customer experience is correlated with customer loyalty and revenue.
- In order to improve the customer experience, MSPs analyze a number of factors such as demographic data (gender, age, marital status, and language preferences), customer preferences, and household structure and usage details to model the customer preferences and offer a relevant personalized service to them. This is known as Targeted Marketing, which improves the adoption of mobile services, reduces Churn rate (this is the percentage of subscribers to a service that discontinue their subscription to that service in a given time period), thus, increasing the revenue of MSPs.

- Network Analytics is the next big thing in Telecom, where MSPs can monitor the network speed and manage the entire network. This helps to resolve the network problems within few minutes and helps to improve the quality of service and the customer experience.
- With the diffusion of Smartphones, based on analysis of real-time location and bevioural data, location-based services/context-based services can be offered to the customers when requested. This would increase the adoption of mobile services.

(3) Financial Firms:

- Currently, capital firms are using advanced technology to store huge volumes of data. But Increasing data sources like Internet and Social media require them to adopt big data storage systems.
- Capital markets are using big data in preparation for regulations like EMIR, Solvency II, Basel II etc, anti-money laundering, fraud mitigation, pre-trade decision-support analytics including sentiment analysis, predictive analytics and data tagging to identify trades.
- The timeliness of finding value plays an important role in both investment banking and capital markets. Hence, there is a need for real-time processing of data.

(4) Retail:

- Evolution of e-commerce, online purchasing, social-network conversations and recently location specific smartphone interactions contribute to the volume and the quality of data for data-driven customization in retailing.
- Major retail stores might place CCTV not only to observe the instances of theft but also to track the flow of customers. It helps to observe the age group, gender and purchasing patterns of the customers during weekdays and weekends.
- Based on the purchasing patterns of the customers, retailers group their items using a well-known data mining technique called Market Basket
- Analytics help the retail companies to manage their inventory.

(5) Law Enforcement:

- Law enforcement officials try to predict the next crime location using past data i.e., type of crime, place and time; social media data; drone and smartphone tracking.
- Researchers at Rutgers University developed an app called RTM Dx to prevent crime and is being used by police department at Illinois, Texas, Arizona, New Jersey, Missouri and Colorado. With the help the app, the police department could measure the spatial correlation between the location of crime and features of the environment.

(6) Marketing:

- Marketing analytics helps the organizations to evaluate their marketing performance, to analyze the consumer behavior and their purchasing patterns, to analyze the marketing trends which would aid in modifying the marketing strategies like the positioning of advertisements in a webpage, implementation of dynamic pricing and offering personalized products

(7) New Product Development:

- There is a huge risk associated with new product development. Enterprises can integrate both external sources, i.e., twitter and Facebook page and internal data sources, i.e., Customer Relationship Management (CRM) systems to understand the customers' requirement for a new product, to gather ideas for new product and to understand the added feature included in a competitor's product.
- Proper analysis and planning during the development stage can minimize the risk associated with the product, increase the customer lifetime value and promote brand engagement.

(8) Banking:

- The investment worthiness of the customers can be analyzed using demographic details, behavioral data, and financial employment. The concept of cross-selling can be used here to target specific customer segments based on past buying behavior, demographic details, sentiment analysis along with CRM data.

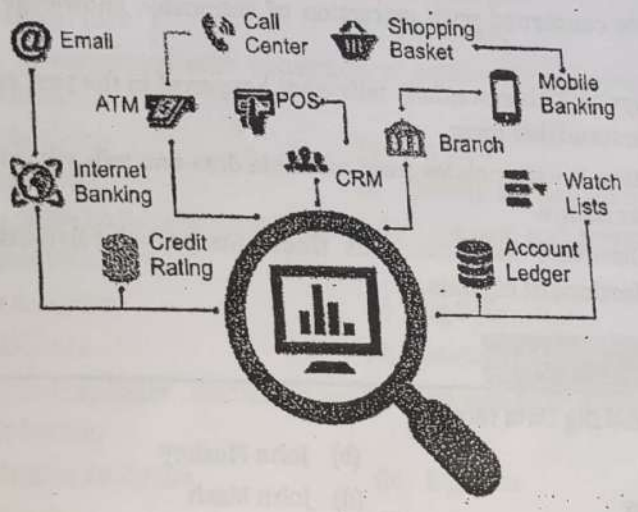


Fig 1.4: Big data application in Banking

(9) Energy and Utilities:

- Consumption of water, gas and electricity can be measured using smart meters at regular intervals of one hour. During this interval, a huge amount of data is generated and analyzed to change the patterns of power usage.
- The real-time analysis reveals energy consumption pattern, instances of electricity thefts and price fluctuations.

(10) Insurance:

- Personalized insurance plan is tailored for each customer using updated profiles of changes in wealth, customer risk, home asset value, and other data inputs). Recently, driving data of customers such as miles driven, routes driven, time of day, and braking abruptness are collected by the insurance companies by using sensors in their cars.
- Comparing individual driving pattern and driver risk with the statistical information available such as peak hours of drivers on the road develops a personalized insurance plan. This analysis of driver risk and policy gives a competitive advantage to the insurance companies.

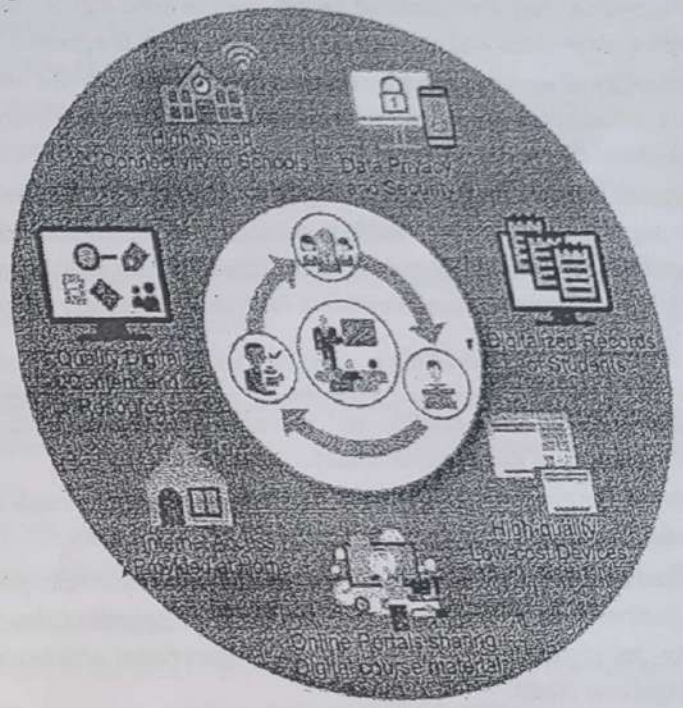


Fig 1.5: Big Data applications in Education

(11) Education:

- With the advent of computerized course modules, it is possible to assess the academic performance real time. This helps to monitor the performance of the students after each module and give immediate feedback on their learning pattern.
- It also helps the teachers to assess their teaching pedagogy and modify based on the students' performance and needs. Dropout patterns, students requiring special attention and students who can handle challenging assignments can be predicted.

(12) Agriculture:

- In agriculture, Big data is playing an influential role to enhance the performance of the firms. Big data provides farmers granular data on rainfall patterns, water cycles, fertilizer requirements, and more.
- This enables them to make smart decisions, such as what crops to plant for better profitability and when to harvest. The right decisions ultimately improve farm yields.

(13) Media and Entertainment:

- Media companies and entertainment sectors need to drive digital transformation to distribute their products and contents as fast as possible at the present market.
- The availability of searching and accessing any content anywhere with any device becomes a widespread practice. It can even help to figure out the views or likes of an artist to measure the popularity in the digital media sector.
- The importance of big data lies in how an organization is using the collected data and not in how much data they have been able to collect. There are Big Data solutions that make the analysis of big data easy and efficient. These Big Data solutions are used to gain benefits from the heaping amounts of data in almost all industry verticals.

Summary

- Big Data helps drive efficiency, quality, and personalized products and services, producing improved levels of customer satisfaction and profit.
- In Big data, so many concepts are associated: basically there were 3 concepts volume, variety, and velocity.
- Driven by specialized analytics systems and software, big data analytics can point the way to various business benefits, including new revenue opportunities, more effective marketing, better customer service, improved operational efficiency and competitive advantages over rivals.
- Companies have searched for decades to make the best use of information to improve their business capabilities.

- Interpretation of Big Data can bring about insights which might not be immediately visible or which would be impossible to find using traditional methods.
- Big data is a term that is used to describe data that is high volume, high velocity, and/or high variety; requires new technologies and techniques to capture, store, and analyze it; and is used to enhance decision making, provide insight and discovery, and support and optimize processes
- The term big data emphasizes volume or size. Size is a relative term. In the 1960s, 20 Megabytes was considered large. Now-data is not considered big unless it is several hundred Petabytes (Petabyte = 1015). Size is not the only property used to describe big data.
- Image, voice, and audio data can be analyzed for applications such as facial recognition systems in security
- Structured data is data whose elements are addressable for effective analysis. It has been organized into a formatted repository that is typically a database.
- 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.
- Semi-structured data is information that does not reside in a relational database but that have some organizational properties that make it easier to analyze.
- It is helpful to recognize that the term analytics is not used consistently; it is used in at least three different yet related ways.
- Data analytics is concerned with extraction of actionable knowledge and insights from big data.
- Descriptive Analytics: This essentially tells what happened in the past and presents it in an easily understandable form.
- Predictive Analytics: It extrapolates from available data and tells what is expected to happen in the near future.
- Exploratory or Discovery Analytics: This finds unexpected relationships among parameters in collections of big data.

Check Your Understanding

1. Who popularized Big Data term ?
   (a) John Deere (b) John Mashey
   (c) Johny Mashe (d) John Mash

2. Numbers, text, image, audio, and video data is ..... .
   (a) Volume (b) Value
   (c) Varity (d) Variety
3. Real time data is ..... .
   (a) fields (b) primary key
   (c) unique (d) record
4. .....is a term that is used to describe data that is high volume, high velocity, and/or high variety.
   (a) Analytics (b) Big Data
   (c) Hadoop data (d) Big Data analytics
5. ..... digital data is based on Relational database table.
   (a) Structured (b) Unstructured
   (c) Semi-structured (d) Semi-Unstructured
6. ..... digital data is based on XML/RDF.
   (a) Structured (b) Unstructured
   (c) Semi-structured (d) Semi-Unstructured
7. ..... digital data is based on character and binary data.
   (a) Structured (b) Unstructured
   (c) Semi-structured (d) Semi-Unstructured
8. ..... is not processing source of Big Data.
   (a) R (b) Yahoo! Pipes,
   (c) Mechanical Turk (d) Datameter
9. ..... is concerned with extraction of actionable knowledge and insights from big data.
   (a) Data analytics (b) Big Data
   (c) Digital data (d) Descriptive Analytics
10. ..... essentially tells what happened in the past and presents it in an easily understandable form.
    (a) Data analytics (b) Big Data
    (c) Digital data (d) Descriptive Analytics
11. ..... extrapolates from available data and tells what is expected to happen in the near future.
    (a) Predictive Analytics (b) Big Data
    (c) Digital data (d) Descriptive Analytics

12. ..... finds unexpected relationships among parameters in collections of big data.
    (a) Exploratory or Discovery Analytics (b) Predictive Analytics
    (c) Digital data (d) Descriptive Analytics

ANSWER KEY

1. (b)	2. (d)	3. (c)	4. (b)	5. (a)
6. (b)	7. (c)	8. (d)	9. (a)	10. (d)
11. (a)	12. (a)			

Practice Questions

Q.I: Answer the following Questions in short.

1. What is the term 'Big Data'?
2. Enlist Tools used for Big Data.
3. What are the five V's of Big Data?
4. What is meant by Petabyte and Zettabyte?
5. Enlist Big Data Applications.
6. What is Big Data Analytics?
7. Enlist Types of Analytics.
8. What is Digital Data?

Q.II: Answer the following Questions.

1. What is Big Data? What is unique about Big Data?
2. Explain Tools used in Big Data.
3. Write Advantages and disadvantages of Big Data.
4. Write difference between Structured Digital Data and Unstructured Digital Data.
5. Differentiate between Unstructured Digital Data and Semi-structured Digital Data.
6. Explain and Draw Diagram for 5 V's.
7. Explain Applications of Big Data.
8. Explain with example types of Digital Data.
9. What is Big Data Analytics? Enlist Types of Analytics with example.
10. Which are the Big Data Sources with example?

Q.III: Define the following terms.

1. Structured data
2. Unstructured Data
3. Semi-Structured Data
4. Volume
5. Value
6. Velocity
7. Variety
8. Descriptive Analytics
9. Digital Data
10. Predictive Analytics

❖❖❖

2... Introduction to Data Science

Objectives...

- To Learn basic concepts of Data Science
- To study Data Analytics with it types
- To study Data Analytics with Statistical Analysis

2.1 INTRODUCTION TO DATA SCIENCE

- The fundamental concepts of data science are drawn from many fields that study data analytics.
- Fundamental concept: Extracting useful knowledge from data to solve business problems can be treated systematically by following a process with reasonably well-defined stages.
- Data science includes data analysis as an important component of the skill set required for many jobs in this area, but is not the only necessary skill.
- A brief case study of a supermarket point of sale system illustrates the many challenges involved in data science work.
- Data scientists play active roles in the design and implementation work of four related areas: data architecture, data acquisition, data analysis, and data archiving.
- Key skills highlighted by the brief case study include communication skills, data analysis skills, and ethical reasoning skills.
- R language is most important for data science.

Components of Data Science:

- The main components of Data Science are given below:

1. **Statistics:** Statistics is one of the most important components of data science. Statistics is a way to collect and analyze the numerical data in a large amount and finding meaningful insights from it.

2. **Visualization:** Data visualization is meant by representing data in a visual context so that people can easily understand the significance of data. Data visualization makes it easy to access the huge amount of data in visuals.
3. **Data engineering:** Data engineering is a part of data science, which involves acquiring, storing, retrieving, and transforming the data. Data engineering also includes metadata (data about data) to the data.
4. **Advanced computing:** Heavy lifting of data science is advanced computing. Advanced computing involves designing, writing, debugging, and maintaining the source code of computer programs.
5. **Machine learning:** Machine learning is all about to provide training to a machine so that it can act as a human brain. In data science, we use various machine learning algorithms to solve the problems.

Advantages of Data Science:

1. Data Science helps organizations knowing how and when their products sell best and that's why the products are delivered always to the right place and right time.
2. Faster and better decisions are taken by the organization to improve efficiency and earn higher profits.
3. It helps the marketing and sales team of organizations in understanding by refining and identifying the target audience.
4. It has made it comparatively easier to sort data and look for best of candidates for an organization. Big Data and data mining have made processing and selection of CVs, aptitude tests and games easier for the recruitment teams.

Disadvantages of Data Science:

1. Extracted information from the structured as well as unstructured data for further use can also misused against a group of people of a country or some committee.
2. Tools used for the data science and analytics are more expensive to use to obtain information. The tools are also more complex, so people have to learn how to use them.

Applications of Data Science:

1. Fraud and Risk Detection.
2. Healthcare.
   (i) Medical Image Analysis
   (ii) Genetics & Genomics
   (iii) Drug Development

3. Virtual assistance for patients and customer support.
4. Internet Search.
5. Targeted Advertising.
6. Website Recommendations.
7. Advanced Image Recognition.
8. Speech Recognition.
9. Airline Route Planning.
10. Gaming.
11. Augmented Reality.

2.1.1 Data Science Process

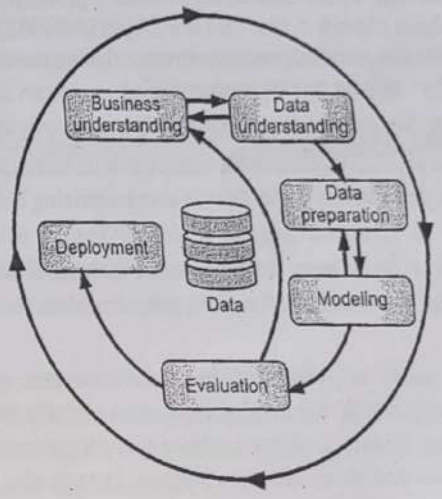


Fig. 2.1: Data Science Process Stages

- **Business Understanding:** The first step of this process is setting a research goal. The main purpose here is making sure all the users understand the what, how, and why of the project. In every serious project this will result in a project charter.
- **Data Understanding:** The second phase is data retrieval. You want to have data available for analysis, so this step includes finding suitable data and getting access to the data from the data owner. The result is data in its raw form, which probably needs polishing and transformation before it becomes usable.
- **Data Preparation:** The data understanding phase starts with an initial data collection and proceeds with activities in order to get familiar with the data, to identify data quality problems, to discover first insights into the data, or to detect interesting subsets to form hypotheses for hidden information.

- **Modeling:** In this phase, various modeling techniques are selected and applied and their parameters are standardized to optimal values. Typically, there are several techniques for the same data mining problem type. Some techniques have specific requirements on the form of data. Therefore, it is often required to step back to the data preparation phase.
- **Evaluation:** At this stage in the project, you have built a model (or models) that appears to have high quality, from a data analysis perspective. Before proceeding to final deployment of the model, it is important to evaluate the model thoroughly and review the steps executed to construct the model, to be certain it properly achieves the business objectives.
- **Deployment:** The last step of the data science model is presenting your results and automating the analysis, if needed. One goal of a project is to change a process and/or make better decisions. You may still need to convince the business that your findings will indeed change the business process as expected.

2.2 BASIC OF DATA ANALYSIS

- We live in a data rich, data driven world. Data is revolutionizing business in ways we never conceived.
- So much of what we do is being recorded and stored somewhere. Companies big and small, in traditional and non-traditional sectors, are using data to understand their customers better.
- Data is helping with better targeting and improved customer experiences. The insights gained from analyzing data is helping companies identify new growth areas and product opportunities, streamline costs, increase operating margins, make better human resource decisions and more effective budgets. Data is also impacting our world, our lives. Health care, the environment, travel....the list is endless.
- Analytics is the systematic computational analysis of data. Analytics is the discovery, interpretation, and communication of meaningful patterns in data. Especially valuable in areas rich with recorded information, analytics relies on the simultaneous application of statistics, computer programming and operations research to quantify performance.
- Data Analytics is the science of examining raw data with the purpose of drawing conclusions about that information. Data Analytics is a lifeline for the IT industry right now.
- Data analysis is a process of inspecting, cleansing, transforming, and modeling data with the goal of discovering useful information, informing conclusions, and supporting decision-making.

What is Analytics?

- Before defining data science, let us learn three terms – Data, Information and Insight.
  (1) **Data** is raw unorganized set of information. We call it as raw data. It contains information; however, the information is not readily available. It is not processed.
  (2) **Information** is when you analyze raw data so that it provides some sort of understanding of the data. The term we use is as information extraction (from data).
  (3) **Insight** is gained by analyzing data and information to understand what is going on with a particular situation. This can be used to make better decisions.
- **Definition:** *"Data analytics is the science of drawing insights from raw information sources. Many of the techniques and processes of data analytics have been automated into mechanical processes and algorithms that work over raw data for human consumption".*
- Data analytics techniques can reveal trends and metrics that would otherwise be lost in the mass of information. This information can then be used to optimize processes to increase the overall efficiency of a business or system.

2.2.1 Phases of Data Analytics Lifecycle

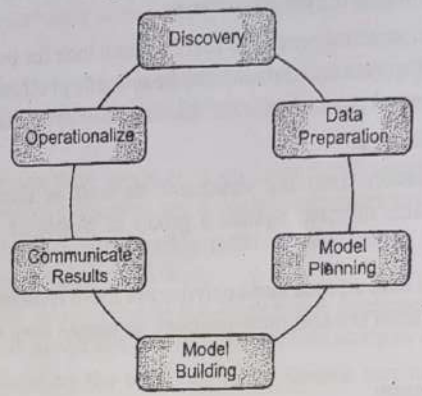


Fig. 2.2: Data Analytics Lifecycle

- **Phase 1: Discovery:** In Phase 1, the team learns the business domain, including relevant history such as whether the organization or business unit has attempted similar projects in the past from which they can learn. The team assesses the resources available to support the project in terms of people, technology, time, and data. Important activities in this phase include framing the business problem as an

analytics challenge that can be addressed in subsequent phases and formulating initial hypotheses (IHs) to test and begin learning the data.

- **Phase 2: Data preparation:** Phase 2 requires the presence of an analytic sandbox, in which the team can work with data and perform analytics for the duration of the project. The team needs to execute Extract, Load, and Transform (ELT) or Extract, Transform and Load (ETL) to get data into the sandbox. The ELT and ETL are sometimes abbreviated as ETLT. Data should be transformed in the ETLT process so the team can work with it and analyze it. In this phase, the team also needs to familiarize itself with the data thoroughly and take steps to condition the data
- **Phase 3: Model planning:** Phase 3 is model planning, where the team determines the methods, techniques, and workflow it intends to follow for the subsequent model building phase. The team explores the data to learn about the relationships between variables and subsequently selects key variables and the most suitable models.
- **Phase 4: Model building:** In Phase 4, the team develops datasets for testing, training, and production purposes. In addition, in this phase the team builds and executes models based on the work done in the model planning phase. The team also considers whether its existing tools will suffice for running the models, or if it will need a more robust environment for executing models and workflows (for example, fast hardware and parallel processing, if applicable).
- **Phase 5: Communicate results:** In Phase 5, the team, in collaboration with major stakeholders, determines if the results of the project are a success or a failure based on the criteria developed in Phase 1. The team should identify key findings, quantify the business value, and develop a narrative to summarize and convey findings to stakeholders.
- **Phase 6: Operationalize:** In Phase 6, the team delivers final reports, briefings, code, and technical documents. In addition, the team may run a pilot project to implement the models in a production environment.

2.3 TYPES OF ANALYTICS

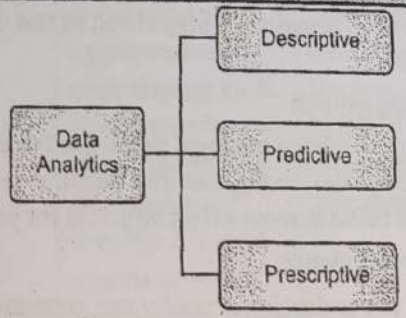


Fig. 2.3: Types of Analytics

2.3.1 Descriptive Analytics

- Descriptive analysis does exactly what the name implies they "Describe", or summarize raw data and make it something that is interpretable by humans. They are analytics that describe the past. The simplest way to define descriptive analytics is that, it answers the question "What has happened?"
- **Definition:** Descriptive Analytics is a statistical method that is used to search and summarize historical data in order to identify patterns or meaning.

How does Descriptive Analytics work?

(1) Data aggregation and data mining are two techniques used in descriptive analytics to discover historical data. Data is first gathered and sorted by data aggregation in order to make the datasets more manageable by analysts.

(2) Data mining describes the next step of the analysis and involves a search of the data to identify patterns and meaning. Identified patterns are analyzed to discover the specific ways that learners interacted with the learning content and within the learning environment.

Examples:

(1) A company reports that simply provide a historic review of an organization's operations, sales, financials, customers, and stakeholders.

(2) Here are some examples of how descriptive analytics is being used in the field of learning analytics:

- Tracking course enrollments, course compliance rates.
- Recording which learning resources are accessed and how often.
- Summarizing the number of times a learner posts in a discussion board.
- Tracking assignment and assessment grades.
- Comparing pre-test and post-test assessments.
- Analyzing course completion rates by learner or by course.
- Collating course survey results.
- Identifying length of time that learners took to complete a course.

Advantages of Descriptive Analytics:

1. Quickly and easily report on the Return On Investment (ROI) by showing how performance achieved business or target goals.
2. Identify gaps and performance issues early - before they become problems.
3. Identify specific learners who require additional support, regardless of how many students or employees there are.

4. Identify successful learners in order to offer positive feedback or additional resources.

- Analyze the value and impact of course design and learning resources.

2.3.2 Predictive Analytics

- Predictive Analytics analyze past data patterns and trends can accurately inform a business about what could happen in the future. It has in the ability to "Predict" what might happen. These analytics are about understanding the future.
- **Definition:** Predictive Analytics is a statistical method that utilizes algorithms and machine learning to identify trends in data and predict future behaviors.

How does Predictive Analytics work?

(1) The software for predictive analytics has moved beyond the realm of statisticians and is becoming more affordable and accessible for different markets and industries, including the field of learning & development.

(2) For the learner, predictive forecasting could be as simple as a dashboard located on the main screen after logging- in to access a course. Analyzing data from past and current progress, visual indicators in the dashboard could be provided to signal whether the employee was on track with training requirements.

(3) At the business level, an LMS system with predictive analytic capability can help improve decision-making by offering in-depth insight to strategic questions and concerns. This could range from anything to course enrolment, to course completion rates, to employee performance.

(4) Some common basic models that are utilized at a broad level include:

- **Decision trees** use branching to show possibilities stemming from each outcome or choice.
- **Regression** techniques assist with understanding relationships between variables.
- **Neural networks** utilize algorithms to figure out possible relationships within data sets.

Examples:

- Training targets
- Talent management

Advantages of Predictive analytics:

1. Personalize the training needs of employees by identifying their gaps, strengths, and weaknesses; specific learning resources and training can be offered to support individual needs.

2. Retain Talent by tracking and understanding employee career progression and forecasting what skills and learning resources would best benefit their career paths. Knowing what skills employees need also benefits the design of future training.
3. Support employees who may be falling behind or not reaching their potential by offering intervention support before their performance puts them at risk.
4. Simplified reporting and visuals that keep everyone updated when predictive forecasting is required.

2.3.3 Prescriptive Analytics

- Prescriptive analytics not only anticipates what will happen and when it will happen, but also why it will happen and provides recommendations regarding actions that will take advantage of the predictions.
- **Definition:** Prescriptive analytics is a statistical method used to generate recommendations and make decisions based on the computational findings of algorithmic models.

How does Prescriptive analytics work?

- Generating automated decisions or recommendations requires specific and unique algorithmic models and clear direction from those utilizing the analytical technique. A recommendation cannot be generated without knowing what to look for or what problem is desired to be solved. In this way, prescriptive analytics begins with a problem.

Example:

- A Training Manager uses predictive analysis.

Advantages of Prescriptive analytics:

- **Make real-time data-driven decisions**

  Prescriptive analytics streamlines the decision-making process by using a data-driven approach. In this way, you can remain one step ahead of the competition and take advantage of new opportunities.

- **Solve issues that may be impeding growth**

  Prescriptive analytics is also a way of solving business problems. For example, if your current pricing strategy is reducing sales, using machine learning to measure many different market conditions could result in more effective prices for your products.

- **Create a highly customized user experience**

  Prescriptive analytics also allows you to develop a quality and customized experience for all customers. Businesses can use this approach to recommend products, deliver timely discounts, and provide website recommendations in real time.

2.3.4 Difference between Descriptive, Predictive and Perspective Analysis

Table 2.1: Difference between various types of Analytics

Sr. No.	Points for differentiation	Descriptive Analysis	Predictive Analysis	Perspective Analysis
1.	**Purpose**	It summarizes raw data and makes it interpretable by humans. They are analytics that describe the past.	Predictive analytics provide estimates about the likelihood of a future outcome.	It attempts to quantify the effect of future decisions in order to advise on possible outcomes before the decisions are actually made.
2.	**Focus on**	Insight into the past.	Understanding the future.	Advise on possible outcomes.
3.	**Answer the Questions**	What has happened?"	What might happen?	What should we do?
4.	**Tools used**	Data aggregation, Data Mining.	Statistics Modeling, Simulation.	Business rules, Heuristic algorithms, Machine Learning, Computational Modeling, Optimization.
5.	**When to Use**	When you need to understand at an aggregate level what is going on in your company, and when you want to summarize and describe different aspects of your business.	Any time you need to know something about the future, or fill in the information that you do not have.	Anytime you need to provide users with advice on what action to take.

Contd...

6.	Example	Company Reports that simply provide a historic review of an organization's operations, sales, financials, customers, and stakeholders.	Training targets, Talent management.	A Training Manager uses predictive analysis.

2.4 STATISTICAL INFERENCE

- **Definition:** Statistical Inference is the process of using data analysis to deduce properties of an underlying distribution of probability. Inferential statistical analysis infers properties of a population, for example by testing hypotheses and deriving estimates. It is assumed that the observed data set is sampled from a larger population.

OR

- Statistical inference is method for drawing and measuring the reliability of conclusions about population based on information obtained from a sample of the population.

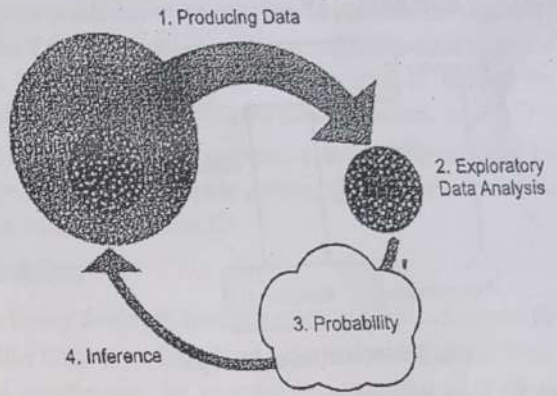


Fig. 2.4: Statistical Inference

- The four-step process that encompasses statistics: Data Production, Exploratory Data Analysis, Probability, and Inference.
- The specific form of inference called for depends on the type of variables involved — either a single categorical or quantitative variable, or a combination of two variables whose relationship is of interest.

- There are two broad areas of Statistical Inference: Statistical Estimation and Statistical Hypothesis Testing.

(1) Statistical Estimation:

- Statistical estimation is concerned with best estimating a value or range of values for a particular population parameter. There are two types of statistical estimation:
  - (i) **Point Estimation:** In point estimation, we estimate an unknown parameter using a single number that is calculated from the sample data.
  - (ii) **Interval Estimation:** In interval estimation, we estimate an unknown parameter using an interval of values that is likely to contain the true value of that parameter (and state how confident we are that this interval indeed captures the true value of the parameter).

(2) Statistical Hypothesis Testing:

- Hypothesis testing is concerned with deciding whether the study data are consistent at some level of agreement with a particular population parameter. In hypothesis testing, we begin with a claim about the population (we will call the null hypothesis), and we check whether or not the data obtained from the sample provide evidence against this claim.

2.4.1 Populations and Samples

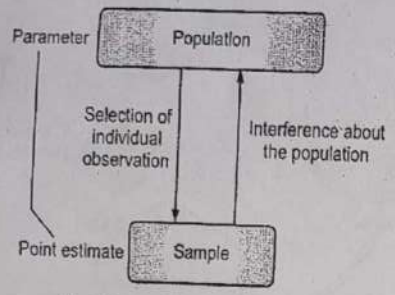


Fig. 2.5: Population and Sample

- Whenever we hear the term 'population', the first thing that strikes our mind is a large group of people. In the same way, in statistics population denotes a large group consisting of elements having at least one common feature.
- The term is often contrasted with the sample, which is nothing but a part of the population that is so selected to represent the entire group.
- A Population consists of the totality of the observations with which we are concerned. In other words, it is the total collection of measurements or observations being studied by a decision maker.

- A part of population selected according to some rule or plan for drawing conclusions regarding the population is called a Sample.
- **Definition of Population:** The aggregate of all elements under study having one or more common characteristic or Population is the collection of all individuals or items under consideration in a statistical study.
  - For example, all people living in India constitutes the population. The population is not confined to people only, but it may also include animals, events, objects, buildings, etc.
  - The different types of population are discussed as under:
    - (1) **Finite Population:** When the number of elements of the population is fixed and thus making it possible to enumerate it in totality, the population is said to be finite.
    - (2) **Infinite Population:** When the numbers of units in a population are uncountable, and so it is impossible to observe all the items of the universe, then the population is considered as infinite.
    - (3) **Existent Population:** The population which comprises of objects that exist in reality is called existent population.
    - (4) **Hypothetical Population:** Hypothetical or imaginary population is the population which exists hypothetically.
- **Definition of Sample:** Sample is a part of population chosen at random for participation in the study or Sample is that part of the population from which information is collected.
  - The sample selected should be such that it represents the population in all its characteristics, and it should be free from bias, so as to produce miniature cross-section, as the sample observations are used to make generalizations about the population.
  - In other words, the respondents are selected out of population constitutes a 'sample', and the process of selecting respondents is known as 'sampling.' The units under study are called sampling units, and the number of units in a sample is called sample size.
  - While conducting statistical testing, samples are mainly used when the sample size is too large to include all the members of the population under study.
- The following table shows difference between Population and Sample:

Table 2.2: Difference between Population and Sample

Sr. No.	Basis for comparison	Population	Sample
1.	Meaning	Population refers to the collection of all elements possessing common characteristics that comprises universe.	Sample means a subgroup of the Members of population chosen for participation in the study.
2.	Includes	Each and every unit of the group.	Only a handful of units of population.
3.	Characteristic	Parameter	Statistic
4.	Data collection	Complete enumeration or census	Sample survey or sampling
5.	Focus on	Identifying the characteristics.	Making inferences about population.
6.	Definition	It is defined as a total of the items under consideration.	It is defined as a proportion of the population selected.
7.	Example	Automobiles with four wheels, People who consume olive oil.	20 cars from each Make.

2.4.2 Statistical Modeling

- Before learning Statistical Modeling we see what is Statistical Model.
- **Definition:** A Statistical Model is a combination of inferences based on collected data and population understanding used to predict information in an idealized form. This means that a statistical model can be an equation or a visual representation of information based on research that's already been collected over time.
- **Definition Statistical Modeling:** Statistical modeling is an approach to statistical data analysis that helps researchers discovers something about a phenomenon that is assumed to exist. This approach helps explain the variability found in the dataset.
- It is a unifying strategy which brings together estimation and hypothesis tests under the same umbrella. This modeling approach constructs summary model that displays current knowledge. The models are then 'fitted' to the data.
- All commonly used statistical procedures can be put into a general modeling framework. This is of the form: Data = Pattern + Residual

Where variation in the observed data can be split into two components:

1. **Pattern:** Systematic or 'explained' variation
2. **Residual:** Leftover or 'unexplained' variation

- In simple terms, statistical modeling is a simplified, mathematically-formalized way to approximate reality (i.e. what generates your data) and optionally to make predictions from this approximation. The statistical model represents the mathematical equation that is used.
- The basic steps of the statistical model-building process are:
  (1) Model selection.
  (2) Model fitting.
  (3) Model validation.
- These three basic steps are used iteratively until an appropriate model for the data has been developed.
- In the model selection step, plots of the data, process knowledge and assumptions about the process are used to determine the form of the model to be fit to the data.
- Then, using the selected model and possibly information about the data, an appropriate model-fitting method is used to estimate the unknown parameters in the model. When the parameter estimates have been made, the model is then carefully assessed to see if the underlying assumptions of the analysis appear plausible. If the assumptions seem valid, the model can be used to answer the scientific or engineering questions that prompted the modeling effort.
- If the model validation identifies problems with the current model, however, then the modeling process is repeated using information from the model validation step to select and/or fit an improved model.

2.4.3 Probability

- Probability theory developed from the study of games of chance like dice and cards. Processes like flipping a coin, rolling a die or drawing a card from a deck are called probability experiments. An outcome is a specific result of a single trial of a probability experiment.
- Probability theory is the foundation for statistical inference. A probability distribution is a device for indicating the values that a random variable may have.
- Probability is the likelihood or chance of an event occurring.

  Probability = The number of ways of achieving success / The total number of possible outcomes.

- To understand probability distributions, it is important to understand some basic terms and some notations.

(1) Probability:

- It is the measure of the likelihood that an event will occur in a Random Experiment. Probability is quantified as a number between 0 and 1, where, loosely speaking, 0 indicates impossibility and 1 indicates certainty. The higher the probability of an event, the more likely it is that the event will occur.
- Probability is important in selecting individuals from a population into a sample, and again in statistical inference when we make generalizations about the population based on that sample. When we select a sample from a population, we want that sample to be representative of the population.

(2) Random Experiment:

- A Random Experiment is a physical situation whose outcome cannot be predicted until it is observed.
  - (i) Trial is the performance or exercise of that experiment.
  - (ii) An outcome is the result of a given trial.
  - (iii) An event is an outcome or any combination of outcomes.

(3) Sample Space:

- A Sample Space is a set of all possible outcomes of a random experiment.

(4) Random variable:

- It is a variable whose possible values are numerical outcomes of a random experiment. A random variable is a symbol (A, B, x, y, etc.) that can take on any of a specified set of values. Generally, statisticians use a capital letter to represent a random variable and a lowercase letter, to represent one of its values. For example, X represents the random variable X.
- P(X) represents the probability of X.
- P(X = x) refers to the probability that the random variable X is equal to a particular value, denoted by x. As an example, P(X = 1) refers to the probability that the random variable X is equal to 1.
- Consider an example will make clear the relationship between random variables and probability distributions. Suppose you flip a coin two times. This simple statistical experiment can have four possible outcomes: HH, HT, TH, and TT. Now, let the variable X represent the number of Heads that result from this experiment. The variable X can take on the values 0, 1, or 2. In this example, X is a random variable; because its value is determined by the outcome of a statistical experiment.

Table 2.3 : Probability Distribution of the Random Variable X

Numbers of heads	Probability
0	0.25
1	0.50
2	0.25

- The above table represents the probability distribution of the random variable X.
- A probability distribution is a function that describes the likelihood of obtaining the possible values that a random variable can assume. In other words, the values of the variable vary based on the underlying probability distribution. It is a table or an equation that links each outcome of a statistical experiment with its probability of occurrence.
- Suppose you draw a random sample and measure the heights of the subjects. As you measure heights, you can create a distribution of heights. This type of distribution is useful when you need to know which outcomes are most likely, the spread of potential values, and the likelihood of different results.

2.4.4 Distribution

- In general, the probability distribution of a random variable X is defined as follows:
- **Definition:** *The probability distribution of a discrete random variable X is a list of each possible value of X together with the probability that X takes that value in one trial of the experiment.*

$$f(x_i) \geq 0 \text{ for all } i, \sum_i f(x_i) = 1,$$

and

$$P\{a < X \leq b\} = \sum_{a < x_i \leq b} f(x_i),$$

- $P(x)$ = the likelihood that random variable takes a specific value of x.
- The probabilities in the probability distribution of a random variable X must satisfy the two conditions: The sum of all probabilities for all possible values must equal 1. Furthermore, the probability for a particular value or range of values must be between 0 and 1.
- A probability distribution may be either discrete or continuous. A discrete distribution means that X can assume one of a countable (usually finite) number of

values, while a continuous distribution means that X can assume one of an infinite (uncountable) number of different values.

- A probability distribution is the function that describes the mapping from any realized value of the random variable to probability.
- The probability distribution of a statistic is called a Sampling Distribution.
- A probability distribution of all the possible means of the samples is a distribution of the sample means called as sampling distribution of the mean; similarly we have the sampling distribution of the proportion. We have a sampling distribution of any statistic that we use.

(1) Discrete Probability Distributions:

- Several specialized discrete probability distributions are useful for specific applications. For business applications, three frequently used discrete distributions are:

  (i) The **Binomial Distribution** is used to compute probabilities for a process where only one of two possible outcomes may occur on each trial.

  (ii) The **Geometric Distribution** is related to the binomial distribution; you use the geometric distribution to determine the probability that a specified number of trials will take place before the first success occurs.

  (iii) The **Poisson Distribution** is used to measure the probability that a given number of events will occur during a given time frame.

(2) Continuous Probability Distributions:

- Many continuous distributions may be used for business applications two of the most widely used are Uniform and Normal.

  (i) The Uniform Distribution is useful because it represents variables that are evenly distributed over a given interval. For example, if the length of time until the next defective part arrives on an assembly line is equally likely to be any value between one and ten minutes, then you may use the uniform distribution to compute probabilities for the time until the next defective part arrives.

  (ii) The Normal Distribution is useful for a wide array of applications in many disciplines. In business applications, variables such as stock returns are often assumed to follow the normal distribution. The normal distribution is characterized by a bell-shaped curve, and areas under this curve represent probabilities.

2.4.5 Correlation

- Correlation analysis is a method of statistical evaluation used to study the strength of a relationship between two, numerically measured, continuous variables (e.g. height and weight). This particular type of analysis is useful when we want to establish if there are possible connections between variables.
- In short, the tendency of simultaneous variation between two variables is called correlation or co-variation. For example, there may be exist a relationship between heights and weights of a group of students, the scores of students in two different subjects are expected to have an interdependence or relationship between them.
- If correlation is found between two variables it means that when there is a systematic change in one variable, there is also a systematic change in the other; the variables, alter together over a certain period of time. If there is correlation found, depending upon the numerical values measured, this can be either positive or negative.
- The knowledge of correlation gives us an idea of the direction and intensity of change in a variable when the correlated variable changes.
- To measure the degree of association or relationship between two variables quantitatively, an index of relationship is used and is termed as co-efficient of correlation.
- Co-efficient of correlation is a numerical index that tells us to what extent the two variables are related and to what extent the variations in one variable changes with the variations in the other. The co-efficient of correlation is always symbolized either by r or ρ(Rho) range from $(-1 \leq r \geq +1)$

2.4.5.1 Techniques for Measuring Correlation

- Three important statistical tools used to measure correlation are: Scatter diagrams, Karl Pearson's coefficient of correlation, and Spearman's rank correlation. We discuss here only scatter diagrams.

Scatter Diagram:

- A scatter diagram visually presents the nature of association without giving any specific numerical value. In this technique, the values of the two variables are plotted as points on a graph paper.
- From a scatter diagram, one can get a fairly good idea of the nature of relationship. In a scatter diagram the degree of closeness of the scatter points and their overall direction enable us to examine the relationship.
- If all the points lie on a line, the correlation is perfect and is said to be unity. If the scatter points are widely dispersed around the line, the correlation is low.

2.4.5.2 Types of Correlation

- Correlation is of following types:

(1) Positive Correlation:

- A positive correlation indicates a positive association between the variables (increasing values in one variable correspond to increasing values in the other variable).
- The values of two variables are changing with same direction. The high numerical values of one variable relate to the high numerical values of the other. i.e. $0 < r < 1$.

For example, Height and weight, study time and grades.

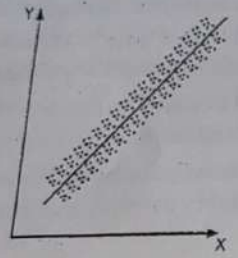


Fig. 2.6 (a): Positive Correlation

(2) Negative Correlation:

- Negative correlation indicates a negative association between the variables (increasing values is one variable correspond to decreasing values in the other variable)
- The values of variables change with opposite direction i.e. the high numerical values of one variable relate to the low numerical values of the other. i.e. $-1 < r < 0$.

For example, Price and quantity demanded, alcohol consumption and driving ability.

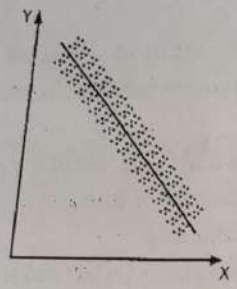


Fig. 2.6 (b) : Negative Correlation

(3) No Correlation:

- There is no impact on one variable with an increase or decrease of values of another variable. If $r = 0$ the two variables are uncorrelated. There is no linear relation between them.

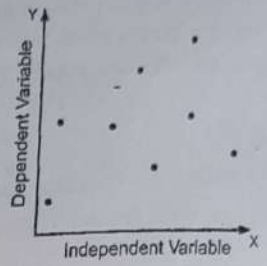


Fig. 2.6 (c): No Correlation

(4) Perfect Positive Correlation:

- When there is a change in one variable, and if there is equal proportion of change in the other variable say Y in the same direction, then these two variables are said to have a Perfect Positive Correlation. i.e. $r = 1$.

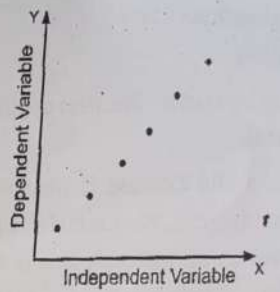


Fig. 2.6 (d): Perfect Positive Correlation

(5) Perfectly Negative Correlation:

- Between two variables X and Y, if the change in X causes the same amount of change in Y in equal proportion but in opposite direction, then this correlation is called as Perfectly Negative Correlation. i.e. $r = -1$.
- If there is correlation between two numerical sets of data, positive or negative, the coefficient worked out can allow you to predict future trends between the two

variables. However, you must remember that you cannot be 100% sure that your prediction will be correct because correlation does not determine cause or effect.

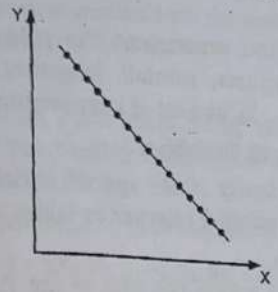


Fig. 2.6 (e): Perfectly Negative Correlation

2.4.6 Regression

- "The statistical technique that expresses a functional relationship between two or more variables in the form of an equation to estimate the value of a variable, based on the given value of another variable is called regression analysis".
- The variable whose value is to be estimated is called dependent variable and the variable whose value is used to estimate this value is called independent variable.
- The linear algebraic equations that express a dependent variable in terms of an independent variable are called Linear Regression Equation.
- In terms of statistical inference, regression analysis is concerned with the parameters of the regression equation that obtains between two or more variables in the population.
- There are a variety of regression methodologies that you choose based on the type of response variable, the type of model that is required to provide an adequate fit to the data, and the estimation method.
- The overall objectives of regression analysis can be summarized as follows:
  (1) To determine whether or not a relationship exists between two variables.
  (2) To describe the nature of the relationship, should one exist, in the form of a mathematical equation?
  (3) To assess the degree of accuracy of description or prediction achieved by the regression equation, and
  (4) In the case of multiple regression, to assess the relative importance of the various predictor variables in their contribution to variation in the criterion variable.

2.4.6.1 Types of Regression Models

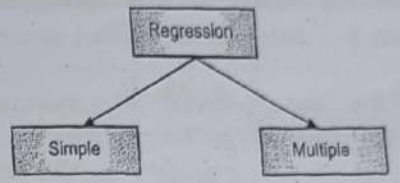


Fig. 2.7: Types of Regression Models

- The two basic types of regression analysis are:

(1) Simple Regression Analysis:

- It is used to estimate the relationship between a dependent variable and a single independent variable.
- Regression models that involve one explanatory variable are called Simple Regression.
- For example, the relationship between crop yields and rainfall.

(2) Multiple Regression Analysis:

- It is used to estimate the relationship between a dependent variable and two or more independent variables.
- When two or more explanatory variables are involved, the relationships are called Multiple Regressions.
- For example, the relationship between the salaries of employees and their experience and education.
- Multiple regression analysis introduces several additional complexities but may produce more realistic results than simple regression analysis.
- Regression models are also divided into linear and nonlinear models, depending on whether the relationship between the response and explanatory variables is linear or nonlinear.
- In a simple linear regression, there are two variables x and y, wherein y depends on x or say influenced by x. Here y is called as dependent, or criterion variable and x is independent or predictor variable.
- The relationship between the independent and dependent variables is linear is the major assumption in a linear regression model.
- The regression line of y on x is expressed as under:

$$y = a + bx$$

Where, a = constant, b = regression coefficient. In this equation, a and b are the two regression parameter. While there are a number of possible criteria for choosing a best-fitting line, one of the most useful is the least squares criterion.

- The slope b of the best-fitting line, based on the least squares criterion, can be shown be

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

Where the summation is overall n pairs of (x_1, y_1) values.

- The value of a, the y-intercept, can be turn be shown to be a function of b, x and y i.e.

$$a = \bar{y} - b\bar{x}$$

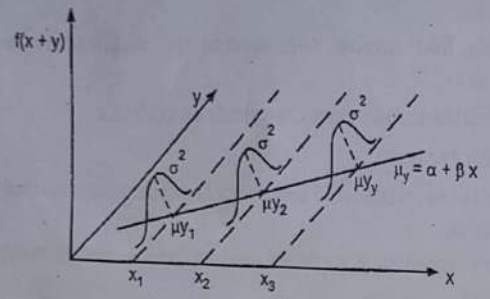


Fig. 2.8: Regression Model y populations with equal σ^2 and μ

- Based on above, the individual observations of y_i are,

$$y_i = \alpha + \beta x + e_i$$

2.4.6.2 Regression Analysis

- Regression analysis includes the following steps:

Step 1: Statement of the Problem under Consideration:

- The first important step in conducting any regression analysis is to specify the problem and the objectives to be addressed by the regression analysis.
- The wrong formulation or the wrong understanding of the problem will give the wrong statistical inferences. The choice of variables depends upon the objectives of study and understanding of the problem.

Step 2: Choice of Relevant Variables:

- Once the problem is carefully formulated and objectives have been decided, the next question is to choose the relevant variables.

- It has to keep in mind that the correct choice of variables will determine the statistical inferences correctly.
- For example, in any agricultural experiment, the yield depends on explanatory variables like quantity of fertilizer, rainfall, irrigation, temperature etc. These variables are denoted by $X_1, X_2, ..., X_k$, as a set of k explanatory variables.

Step 3: Collection of Data on Relevant Variables:

- Once the objective of study is clearly stated and the variables are chosen, the next question arises is to collect data on such relevant variables. The data is essentially the measurement on these variables.
- For example, suppose we want to collect the data on age. For this, it is important to know how to record it. Then either the date of birth can be recorded which will provide the exact age on any specific date or the age in terms of completed years as on a specific date.
- Moreover, it is also important to decide that whether the data has to be collected on variables as quantitative variables or qualitative variables

Step 4: Specification of Model:

- The experimenter or the person working in the subject usually helps in determining the form of the model. Only the form of the tentative model can be ascertained and it will depend on some unknown parameters.

For example, a general form will be like

$$y = f(X_1, X_2, ..., X_k; \beta_1, \beta_2, ... \beta_k) + \varepsilon$$

where, ε is the random error reflecting mainly the difference in the observed value of y and the value of y obtained through the model. The form of $f(X_1, X_2, ..., X_k, \beta_1, \beta_2, ..., \beta_k)$ can be linear as well as nonlinear depending on the form of parameters $\beta_1, \beta_2, ..., \beta_k$. A model is said to be linear if it is linear in parameters.

For example,

$$y = \beta_1 X_1 + \beta_2 X^2_1 + \beta_3 X_2 + \varepsilon$$

$$y = \beta_1 + \beta_2 \ln X_2 + \varepsilon$$

are linear models whereas

$$y = \beta_1 X_1 + \beta^2_2 X_2 + \beta_3 X_2 + \varepsilon$$

$$y = (\ln \beta_1) X_1 + \beta_2 X_2 + \varepsilon$$

are non-linear models.

Step 5: Choice of Method for Fitting the Data:

- After the model has been defined and the data have been collected, the next task is to estimate the parameters of the model based on the collected data. This is also referred to as parameter estimation or model fitting.

- The most commonly used method of estimation is the least squares method. Under certain assumptions, the least squares method produces estimators with desirable properties. The other estimation methods are the maximum likelihood method, ridge method, principal components method etc.

Step 6: Fitting of Model:

- The estimation of unknown parameters using appropriate method provides the values of the parameter. Substituting these values in the equation gives us a usable model. This is termed as model fitting.
- The estimates of parameters $\beta_1, \ldots, \ldots, \beta_k$ in the model

$$y = f(X_1, X_2, \ldots, X_k, \beta_1, \beta_2, \ldots, \beta_k) + \varepsilon$$

are denoted as $\hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_k$ which gives the fitted model as

$$y = f(X_1, X_2, \ldots, X_k, \hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k)$$

- When the value of y is obtained for the given values of $X_1, X_2, \ldots, X_k$, it is denoted as $\hat{y}$ and called as **fitted value**.
- The fitted equation is used for prediction. In this case, $\hat{y}$ is termed as **predicted value**. Note that the fitted value is where the values used for explanatory variables correspond to one of the n observations in the data **whereas predicted value is the** one obtained for any set of values of explanatory variables. It is not generally recommended to predict the y - values for the set of those values of explanatory variables which lie outside the range of data. When the values of explanatory variables are the future values of explanatory variables, the predicted values are called **forecasted values.**

Step 7: Model Validation and Criticism:

- The validity of statistical method to be used for regression analysis depends on various assumptions. These assumptions are essentially the assumptions for the model and the data.
- The quality of statistical inferences heavily depends on whether these assumptions are satisfied or not. For making these assumptions to be valid and to be satisfied, care is needed from beginning of the experiment.
- One has to be careful in choosing the required assumptions and to examine whether the assumptions are valid for the given experimental conditions or not. It is also important to decide the situations in which the assumptions may not meet.
- The validation of the assumptions must be made before drawing any statistical conclusion. Any departure from validity of assumptions will be reflected in the

statistical inferences. In fact, the regression analysis is an iterative process where the outputs are used to diagnose, validate, criticize and modify the inputs.

Step 8: Using the Chosen Model(s) for the Solution of the posed problem and forecasting:

- The determination of explicit form of regression equation is the ultimate objective of regression analysis. It is finally a good and valid relationship between study variable and explanatory variables.
- The regression equation helps in understanding the interrelationships among the variables. Such regression equation can be used for several purposes.
- For example, to determine the role of any explanatory variable in the joint relationship in any policy formulation, to forecast the values of response variable for given set of values of explanatory variables.

2.4.6.3 Applications or Uses of Regression Analysis

(1) Predictive Analytics:

- Predictive analytics i.e. forecasting future opportunities and risks is the most prominent application of regression analysis in business. Demand analysis, for instance, predicts the number of items which a consumer will probably purchase.
- However, demand is not the only dependent variable when it comes to business. Regression analysis can go far beyond forecasting impact on direct revenue.
- For example, Insurance companies heavily rely on regression analysis to estimate the credit standing of policyholders and a possible number of claims in a given time period.

(2) Operation Efficiency:

- Regression models can also be used to optimize business processes. A factory manager, for example, can create a statistical model to understand the impact of oven temperature on the shelf life of the cookies baked in those ovens.
- In a call center, we can analyze the relationship between wait times of callers and number of complaints. Data-driven decision making eliminates guesswork, hypothesis and corporate politics from decision making.
- This improves the business performance by highlighting the areas that have the maximum impact on the operational efficiency and revenues.

(3) Supporting Decisions:

- Today businesses are overloaded with data on finances, operations and customer purchases. Increasingly, executives are now leaning on data analytics to make informed business decisions.

- Regression analysis can bring a scientific angle to the management of any businesses. By reducing the tremendous amount of raw data into actionable information, regression analysis leads the way to smarter and more accurate decisions. This technique acts as a perfect tool to test a hypothesis before diving into execution.

(4) Correcting Errors:

- Regression is not only great for lending empirical support to management decisions but also for identifying errors in judgment. For example, a retail store manager may believe that extending shopping hours will greatly increase sales.
- Regression analysis, however, may indicate that the increase in revenue might not be sufficient to support the rise in operating expenses due to longer working hours (such as additional employee labor charges).
- Hence, regression analysis can provide quantitative support for decisions and prevent mistakes due to manager's intuitions.

(5) New Insights:

- Over time businesses have gathered a large volume of unorganized data that has the potential to yield valuable insights. However, this data is useless without proper analysis.
- Regression analysis techniques can find a relationship between different variables by uncovering patterns that were previously unnoticed.
- For example, analysis of data from point of sales systems and purchase accounts may highlight market patterns like increase in demand on certain days of the week or at certain times of the year. You can maintain optimal stock and personnel before a spike in demand arises by acknowledging these insights.

2.4.6.4 Difference between Correlation and Regression

- Let us see difference between Correlation and Regression.

Table 2.4: Difference between Correlation and Regression

Sr. No.	Basic for comparison	Correlation	Regression
1.	Meaning	Correlation is a statistical measure which determines co-relationship or association of two variables.	Regression describes how an independent variable is numerically related to the dependent variable.

Contd..

2.	Usage	To represent linear relationship between two variables.	To fit a best line and estimate one variable on the basis of another variable.
3.	Dependent and Independent variables	No difference	Both variables are different.
4.	Indicates	Correlation coefficient indicates the extent to which two variables move together.	Regression indicates the impact of a unit change in the known variable (x) on the estimated variable (y).
5.	Objective	To find a numerical value expressing the relationship between variables.	To estimate values of random variable on the basis of the values of fixed variable.

Summary

- The fundamental concepts of data science are drawn from many fields that study data analytics.
- Fundamental concept: Extracting useful knowledge from data to solve business problems can be treated systematically by following a process with reasonably well-defined stages.
- Data science includes data analysis as an important component of the skill set required for many jobs in this area, but is not the only necessary skill.
- Data Science helps organizations knowing how and when their products sell best and that's why the products are delivered always to the right place and right time.
- Data Science Process stages: 1. Business Understanding, 2. Data Understanding, 3. Data Preparation, 4. Modeling, 5. Evaluation, 6. Deployment.
- Descriptive analysis does exactly what the name implies they "Describe", or summarize raw data and make it something that is interpretable by humans. They are analytics that describe the past. The simplest way to define descriptive analytics is that, it answers the question "What has happened?"
- Predictive analytics analyze past data patterns and trends can accurately inform a business about what could happen in the future. It has in the ability to "Predict" what might happen. These analytics are about understanding the future.

- Prescriptive analytics not only anticipates what will happen and when it will happen, but also why it will happen and provides recommendations regarding actions that will take advantage of the predictions.
- Statistical inference is the process of using data analysis to deduce properties of an underlying distribution of probability. Inferential statistical analysis infers properties of a population, for example by testing hypotheses and deriving estimates. It is assumed that the observed data set is sampled from a larger population.
- Probability theory developed from the study of games of chance like dice and cards. Processes like flipping a coin, rolling a die or drawing a card from a deck are called probability experiments. An outcome is a specific result of a single trial of a probability experiment.
- In general, the probability distribution of a random variable X is defined as follows: Definition: The probability distribution of a random variable X is the system of numbers
- The real numbers x1, x2, ... xn, are the possible values of the random variable X and (i = 1, 2, ..., n) is the probability of the random variable X taking the value x, i.e., P (X = xi) = pi.
- Correlation analysis is a method of statistical evaluation used to study the strength of a relationship between two, numerically measured, continuous variables (e.g. height and weight). This particular type of analysis is useful when we want to establish if there are possible connections between variables.
- "The statistical technique that expresses a functional relationship between two or more variables in the form of an equation to estimate the value of a variable, based on the given value of another variable is called regression analysis".

Check Your Understanding

1. ..... useful knowledge from data to solve business problems can be treated systematically by following a process with reasonably well-defined stages.
   (a) Extracting (b) Preparing
   (c) Prescribing (d) None
2. ..... answers the question "What has happened?"
   (a) Descriptive analytics (b) Predictive analytics
   (c) Prescriptive analytics (d) None

3. ..... answers the question "What will happen?"
   (a) Descriptive analytics (b) Predictive analytics
   (c) Prescriptive analytics (d) None
4. ..... answers the question "How can we make it happen?"
   (a) Descriptive analytics (b) Predictive analytics
   (c) Prescriptive analytics (d) None
5. ..... is most important language for Data Science.
   (a) Java (b) Ruby
   (c) R (d) None of the mentioned
6. ..... phase of the data analytics lifecycle usually takes the longest time.
   (a) Phase 2: Data Preparation (b) Phase 3: Model Planning
   (c) Phase 4: Model Building (d) Phase 5: Communicate Results
7. When data are collected in a statistical study for only a portion or subset of all elements of interest we are using ..... .
   (a) Sample (b) Parameter
   (c) Population (d) None
8. In Statistics, a population consists of ..... .
   (a) All People living in a country.
   (b) All People living in the city are under study.
   (c) All subjects or objects whose characteristics are being studied.
   (d) None of the above
9. The strength (degree) of the correlation between a set of independent variables X and a dependent variable Y is measured by .....
   (a) Coefficient of Correlation (b) Coefficient of Determination
   (c) Standard error of estimate (d) All of the above
10. Correlation Coefficient values lies between .....
    (a) −1 and +1 (b) 0 and 1
    (c) −1 and 0 (d) None of these
11. In correlation, both variables are always .....
    (a) Random (b) Non Random
    (c) Same (d) None

12. If two variables oppose each other then the correlation will be.....
    (a) Positive Correlation (b) Zero Correlation
    (c) Perfect Correlation (d) Negative Correlation
13. A perfect negative correlation is signified by .....
    (a) 0 (b) 1
    (c) 0.5 (d) −1
14. If X and Y are independent to each other, the coefficient of correlation is .....
    (a) −1 (b) 0
    (c) +1 (d) None
15. If the scatter diagram is drawn the scatter points lie on a straight line then it indicate .....
    (a) Regression (b) Skewness
    (c) No correlation (d) Perfect correction
16. ..... is the major assumption in a linear regression model.
    (a) The independent variables are numeric variables.
    (b) There is only one dependent variable.
    (c) The relationship between the independent and dependent variables is linear.
    (d) The error term is a normally distributed random variable with mean zero and constant variance.
17. ..... input (independent) variables are in used a simple linear regression model
    (a) 1
    (b) 2
    (c) 3
    (d) Depends on the number of features/attributes involved Bottom of Form.
18. ..... of the following is not a step in data analysis.
    (a) Obtain the data
    (b) Clean the data
    (c) EDA
    (d) None of the mentioned
19. Regression analysis .....
    (a) Establishes a relationship between two variables.
    (b) Establishes cause and effect.

    (c) Measures growth.
    (d) Measures the demand for a good.
20. The dependent variable is also called .....
    (a) Regression (b) Regressand
    (c) Continuous variable (d) Independent
21. The independent variable is also called .....
    (a) Regressor (b) Predictand variable
    (c) Explained variable (d) All of these

ANSWER KEY

1. (a)	2. (a)	3. (b)	4. (c)	5. (c)
6. (a)	7. (a)	8. (c)	9. (a)	10. (a)
11. (a)	12. (d)	13. (d)	14. (a)	15. (c)
16. (c)	17. (a)	18. (d)	19. (a)	20. (b)
21. (a)				

Practice Questions

Q.I: Answer the following Questions in short.

1. Enlist Phases of data analytics life cycle.
2. What is Data Science?
3. Enlist Stages of Data Science.
4. Which are types of data analytics?
5. What is the Statistical Inference?
6. What is Regression and Correlation?

Q.II: Answer the following Questions.

1. What is data science? Advantages and disadvantages of data science.
2. Explain need of data analytics.
3. Explain life cycle of data analytics.
4. Explain different types of data analytics.
5. Differentiate between descriptive, perspective and predictive data analytics.

6. What is statistical Inference? With example.
7. Differentiate between population and sample.
8. What is statistical modeling? With example.
9. Explain Probability Distribution modeling.
10. Explain Phases of Data science Projects.

Q.III: Define the following terms.

1. Data Science
2. Data Analytics
3. Probability
4. Population
5. Sampling
6. Descriptive Analytics
7. Predictive Analytics
8. Prescriptive Analytics

❖❖❖

3... Introduction to Machine Learning

Objectives...

- To learn basic concepts of Machine Learning.
- To study Supervised and Unsupervised Machine Learnings.
- To understand Regression Analysis with its types.
- To understand different algorithms such as EM, Apriori.

3.1 BASICS OF MACHINE LEARNING

- Machine Learning is an idea to learn from examples and experience, without being explicitly programmed. Instead of writing code, you feed data to the generic algorithm, and it builds logic based on the data given.
- Machine Learning theory is a field that meets statistical, probabilistic, computer science and algorithmic aspects arising from learning iteratively from data which can be used to build intelligent applications.
- Machine Learning allows us to predict things accurately using simple statistical methods, algorithms, and modern computing power.
- Machine Learning can best be understood through four terms:
  (1) **The Broad:** Machine Learning is the process of predicting things, usually based on what they have done in the past.
  (2) **The Practical:** Machine Learning tries to find relationships in your data that can helps you to predict what will happen next.
  (3) **The Technical:** Machine Learning uses statistical methods to predict the value of a target variable using a set of input data.
  (4) **The Mathematical:** Machine Learning attempts to predict the value of a variable Y given an input of feature set X.

- The basic principle of Machine Learning is the automatic modeling of underlying processes that have generated the collected data. Learning from data results in rules, functions, relations, equation systems, probability distributions and other knowledge representations such as decision rules, decision and regression trees, Bayesian nets, neural nets etc.

Definition:

- "Machine learning is a scientific discipline that is concerned with the design and development of algorithms that allow computers to evolve behaviors based on empirical data, such as from sensor data or databases".

Need of Machine Learning:

- Recent progress in machine learning has been driven by the development of new learning algorithms and innovative researches, backed by the ongoing explosion in online and offline data. Also, the availability of low cost computation plays an important role. Here are the few driving forces that justify the need of machine learning and look at the business drivers of it.
  - **Diversity of data:** Data is being generated from different channels and its nature and format are different.
  - **Capacity and dimension:** The increase in the number of data sources and the globalization of diversification of businesses has led to the exponential growth of the data.
  - **Speed:** As data volume increases, so must the speed at which data is captured and transformed.
  - **Complexity:** With the increasing complexity of data, high data quality and security is required to enable data collection, transformation, and analysis to achieve expedient decision making.
  - **Applicability:** These aforementioned factors can compromise the applicability of the data to business process and performance improvement.

Types of Machine Learning:

- Machine learning covers two main types of Data Analysis:

(1) Exploratory Analysis (Unsupervised Learning):

- Discover the structure within the data. e.g. Experience (in years in a company) and salary are correlated.
- Unsupervised learning which provides the algorithm with no labelled data in order to allow it to find structure within its input data.

(2) Predictive Analysis (Supervised Learning):

- This is sometimes described as "learn from the past to predict the future".

- Supervised learning which trains algorithms based on example input and output data that is labelled by humans Scenario: a company wants to detect potential future clients among a base of prospects.

Retrospective Data Analysis: Reinforcement learning describes a class of problems where an agent operates in an environment and must learn to operate using feedback.

For Example, we go through the data constituted of previous prospected companies, with their characteristics (size, domain, localization, etc.).

- Some of these companies became clients, others did not. The question is, can we possibly predict which of the new companies are more likely to become clients, based on their characteristics, based on previous observations?
- In this example, the training data consists of a set of n training samples. Each sample, x, is a vector of p input features (company characteristics) and a target feature (yi ∈ {Yes, No}) (whether they became a client or not).

Advantages of Machine Learning:

1. It is used in variety of applications such as banking and financial sector, healthcare, retail, publishing and social media, robot locomotion, game playing etc.
2. It has capabilities to handle multi-dimensional and multi-variety data in dynamic or uncertain environments.
3. It allows time cycle reduction and efficient utilization of resources.
4. Source programs such as 'Rapidminer' helps in increased usability of algorithms for various applications.
5. Due to Machine Learning, there are tools available to provide continuous quality improvement in large and complex process environments.
6. The process of automation of tasks is easily possible.

Disadvantages of Machine Learning:

1. Acquisition of relevant data is the major challenge. Based on different algorithms data need to be processed before providing as input to respective algorithms. This has significant impact on results to be achieved or obtained.
2. It is impossible to make immediate accurate predictions with a machine learning system.
3. Machine learning needs a lot of training data for future prediction.
4. Interpretation of results is also a major challenge to determine effectiveness of machine learning algorithms.

- Machine learning is used for data analysis and knowledge discovery in databases, data mining, automatic generation of knowledge bases for expert systems, game playing, text classification and text mining, automatic recognition of speech, handwriting, images, etc.

3.1.1 Basic Interaction with R

- Familiarity with software such as R allows users to visualize data, run statistical tests, and apply machine learning algorithms.
- R is a programming language and software environment for statistical analysis, graphics representation and reporting.
- R was created by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand, and is currently developed by the R Development Core Team. R is freely available under the GNU General Public License, and pre-compiled binary versions are provided for various operating systems like Linux, Windows and Mac.
- This programming language was named R, based on the first letter of first name of the two R authors (Robert Gentleman and Ross Ihaka), and partly a play on the name of the Bell Labs Language S.

Features of R:

The following are the important features of R :

- R is a well-developed, simple and effective programming language which includes conditionals, loops, user defined recursive functions and input and output facilities.
- R has an effective data handling and storage facility,
- R provides a suite of operators for calculations on arrays, lists, vectors and matrices.
- R provides a large, coherent and integrated collection of tools for data analysis.
- R provides graphical facilities for data analysis and display either directly at the computer or printing at the papers.

3.1.1.1 Installation of R

(a) On Windows OS:

(1) Open an internet browser and go to www.r-project.org.

(2) Click the "download R" link in the middle of the page under "Getting Started."

(3) Select a CRAN location (a mirror site) and click the corresponding link.

(4) Click on the "Download R for Windows" link at the top of the page.

(5) Click on the "install R for the first time" link at the top of the page.

(6) Click "Download R for Windows" and save the executable file somewhere on your computer. Run the .exe file and follow the installation instructions.

(b) On Linux OS:

(1) R is available as a binary for many versions of Linux at the location R Binaries.

(2) The instruction to install Linux varies from flavor to flavor. These steps are mentioned under each type of Linux version in the mentioned link. However, if you are in a hurry, then you can use yum command to install R as follows –

```
$ yum install R
```


R Command Prompt

- Once you have R environment setup, then it's easy to start your R command prompt by just typing the following command at your command prompt :

```
$ R
```

- This will launch R interpreter and you will get a prompt > where you can start typing your program as follows :

```
> myString ← "HI, KAMIL KHAN!"
> print ( myString )
#output → [1] " HI, KAMIL KHAN!"
```

- Here first statement defines a string variable myString, where we assign a string " HI, KAMIL KHAN!" and then next statement print() is being used to print the value stored in variable myString.

R Script File

- Usually, you will do your programming by writing your programs in script files and then you execute those scripts at your command prompt with the help of R interpreter called Rscript. So let's start with writing following code in a text file called "test.R" as under :

```
# My first program in R Programming
myString ← "HI, KAMIL KHAN!"
print ( myString )
```

- Save the above code in a file test.R and execute it at Linux command prompt as given below. Even if you are using Windows or other system, syntax will remain same.

```
$ Rscript test.R
```

When we run the above program, it produces the following result.

```
"HI, KAMIL KHAN!"
```

Comments

- Comments are like helping text in your R program and they are ignored by the interpreter while executing actual program. Single comment is written using # in the beginning of the statement as follows :

```
# My first program in R Programming
```

- R does not support multi-line comments but you can perform a trick which is something as follows :

```
If (FALSE) {
    "This is a demo for multi-line comments and it should be put inside
     either a single OR double quote"
```

```
}
myString ← " HI, KAMIL KHAN!"
print ( myString)
[1] " HI, KAMIL KHAN!"
```

3.1.1.2 Basic Computations in R

- Let's begin with basics. To get familiar with R coding environment, start with some basic calculations. R console can be used as an interactive calculator too. Type the following in your console:

```
> 2 + 3
> 5
> 6 / 3
> 2
> (3*8)/(2*3)
> 4
> log(12)
> 1.07
> sqrt (121)
> 11
```

- Similarly, you can experiment various combinations of calculations and get the results. In case, you want to obtain the previous calculation, this can be done in two ways.
- First, click in R console, and press 'Up / Down Arrow' keys on your keyboard. This will activate the previously executed commands. Press Enter.
- But, if you have done too many calculations, creating variable is a helpful way.
- In R, you can create a variable using ← or = sign.
- Let's say I want to create a variable x to compute the sum of 7 and 8. It will write as:

```
> x ← 8 + 7
> x
> 15
```

- Once we create a variable, you no longer get the output directly (like calculator), unless you call the variable in the next line.
- Remember, variables can be alphabets, alphanumeric but not numeric. You can't create numeric variables.

3.1.1.3 Essentials of R Programming

Objects of R:

- R has five basic or 'atomic' classes of objects. Everything you see or create in R is an object. A vector, matrix, data frame, even a variable is an object. R treats it that way. So, R has 5 basic classes of objects which are:
  (1) Character
  (2) Numeric (Real Numbers)
  (3) Integer (Whole Numbers)
  (4) Complex
  (5) Logical (True / False)

Attributes:

- These classes have attributes. Think of attributes as their 'identifier', a name or number which appropriately identifies them.
- An object can have following attributes:
  (1) names, dimension names
  (2) dimensions
  (3) class
  (4) length
- Attributes of an object can be accessed using attributes() function. The attributes() function returns or sets all attributes of a data object. We will see more on this topic in following section.
- Let's understand the concept of object and attributes practically.

Data Types in R

- R has various 'data types' which includes vector (numeric, integer etc.), matrices, data frames and list. Let's understand them one by one.

(1) Vector:

- The most basic data object in R is known as vector. You can create an empty vector using vector( ).
- A vector contains object of same class. A vector's type can be logical, integer, double, character, complex or raw.
- The length of vector is the number of elements in the vector and can be checked with the function length( ).
- For example: Let's create vectors of different classes. We can create vector using c() or concatenate command also.

```
> a ← c(1.8, 4.5)   #numeric
> b ← c(1 + 2i, 3 - 6i) #complex
```

```
> d ← c(23, 44)   #integer
> e ← vector("logical", length = 5)
```

- Similarly, you can create vector of various classes. When objects of different classes are mixed in a list, coercion occurs. This effect causes the objects of different types to 'convert' into one class. That means coercion is from lower to higher types from logical to integer to double to character.

  For example:

```
> qt ← c("Time", 24, "October", TRUE, 3.33)  #character
> ab ← c(TRUE, 24) #numeric
> cd ← c(2.5, "May") #character
```

- To check the class of any object, use class("vector name") function.

```
> class(qt)
"character"
```

- If we want to create a vector of consecutive numbers, the : operator is useful.
- To convert the class of a vector, you can use as. command.

```
> bar ← 0:5
> class(bar)
> "integer"
> as.numeric(bar)
> class(bar)
> "numeric"
> as.character(bar)
> class(bar)
> "character"
```

- Similarly, you can change the class of any vector. But, you should pay attention here. If you try to convert a "character" vector to "numeric", NAs will be introduced. Hence, you should be careful to use this command.

(2) List:

- A list is a special type of vector which contains elements of different data types.
- List can be created using the list() function. Its structure can be examined with the str() function.

  For example:

```
> my_list ← list(22, "ab", TRUE, 1 + 2i)
> my_list
[[1]]
[1] 22
```



```
[[2]]
[1] "ab"
[[3]]
[1] TRUE
[[4]]
[1] 1+2i
```

Access components of a list:

- As you can see, the output of a list is different from a vector. This is because all the objects are of different types. The double bracket [[1]] shows the index of first element and so on. Hence, you can easily extract the element of lists depending on their index. Like this:

```
> my_list[[3]]
> [1] TRUE
```

- You can use [] single bracket too. But, that would return the list element with its index number, instead of the result above. Like this:

```
> my_list[3]
> [[1]]
 [1] TRUE
```

(3) Matrices:

- When a vector is introduced with row and column i.e. a dimension attribute, it becomes a matrix. A matrix is represented by set of rows and columns. It is a 2 dimensional data structure. It consists of elements of same class.
- Matrix can be created using the matrix() function. Dimension of the matrix can be defined by passing appropriate value for arguments *nrow* and *ncol*.
- Let's create a matrix of 3 rows and 2 columns:

```
> my_matrix ← matrix(1:6, nrow=3, ncol=2)
> my_matrix
 [,1] [,2]
[1,] 1 4
[2,] 2 5
[3,] 3 6
> dim(my_matrix)
[1] 3 2>
attributes(my_matrix)
$dim
[1] 3 2
```

- As you can see, the dimensions of a matrix can be obtained using either dim( ) or attributes( ) command. To extract a particular element from a matrix, simply use the index shown above. For example (try this at your end):

```
> my_matrix[,2]   #extracts second column
> my_matrix[,1]   #extracts first column
> my_matrix[2,]   #extracts second row
> my_matrix[1,]   #extracts first row
```

- As an interesting fact, you can also create a matrix from a vector. All you need to do is, assign dimension dim( ) later. Like this:

```
> age ← c(23, 44, 15, 12, 31, 16)
> age
[1] 23 44 15 12 31 16
> dim(age) <- c(2,3)
> age
     [,1] [,2] [,3]
[1,]   23   15   31
[2,]   44   12   16
> class(age)
[1] "matrix"
```

- You can also join two vectors using cbind() and rbind() functions. But, make sure that both vectors have same number of elements. If not, it will return NA values.

```
> x ← c(1, 2, 3, 4, 5, 6)
> y ← c(20, 30, 40, 50, 60)
> cbind(x, y)
> cbind(x, y)
     x  y
[1,] 1  20
[2,] 2  30
[3,] 3  40
[4,] 4  50
[5,] 5  60
[6,] 6  70
> class(cbind(x, y))
[1] "matrix"
```


(4) Data Frame:

- This is the most commonly used member of data type's family. It is used to sto tabular data. It is different from matrix. In a matrix, every element must have sam class. But, in a data frame, you can put list of vectors containing different classe This means, every column of a data frame acts like a list.
- Every time you will read data in R, it will be stored in the form of a data frame. Henc it is important to understand the majorly used commands on data frame.
- We can create a data frame using the data.frame( ) function.
- We can use either [,[[or $ operator to access columns of data frame.
- A data frame can be examined using functions like str( ) and head( ).

```
> df ← data.frame(name = c("Kamil","Ajmal","Aadil","Mub"), score
c(67,56,87,91))
> df
  name score
1 Kamil 67
2 Ajmal 56
3 Adil 87
4 Mub 91
```

Functions of Data Frame:

```
> dim(df)
[1] 4 2
> str(df)
'data.frame': 4 obs. of 2 variables:
$ name : Factor w/ 4 levels "Kamil","Ajmal","Mub",..1 2 4 3
$ score: num 67 56 87 91
> nrow(df)
[1] 4
> ncol(df)
[1] 2
```

3.1.1.4 Control Structures in R

- A control structure 'controls' the flow of code / commands written inside a function A function is a set of multiple commands written to automate a repetitive codin task.
- For example: You have 10 data sets. You want to find the mean of 'Age' colum present in every data set. This can be done in 2 ways: Either you write the code t compute mean 10 times or you simply create a function and pass the data set to it.

- Let's understand the control structures in R with simple examples:

(1) **if...else :** This structure is used to test a condition.

Syntax:

```
if (<condition>)
{
##do something
}
else
 {
##do something
}
```

Example:

```
#initialize a variable
N <- 10
#check if this variable * 5 is > 40
if (N * 5 > 40){
print("This is easy!")
} else {
print ("It's not easy!")
}
```

Output:

```
[1] "This is easy!"
```

(2) **for :** This structure is used when a loop is to be executed fixed number of times. It is commonly used for iterating over the elements of an object (list, vector).

Syntax:

```
for (<search condition>)
{
#do something
}
```

Example:

```
#initialize a vector
y ← c (99,45,34,65,76,23)
#print the first 4 numbers of this vector
For (i in 1:4)
{
```



```
print (y[i])
}
```

Output:

```
[1] 99
[1] 45
[1] 34
[1] 65
```

(3) **while:** It begins by testing a condition, and executes only if the condition is found to be true. Once the loop is executed, the condition is tested again. Hence, it's necessary to alter the condition such that the loop doesn't go infinity.

Syntax:

```
While(test_expression)
{
Statement
}
```

Example:

```
#initialize a condition
Age <- 12
#check if age is less than 17
while(Age < 17)
{
print(Age)
Age <- Age + 1 #Once the loop is executed, this code breaks the loop
}
```

Output:

```
[1] 12
[1] 13
[1] 14
[1] 15
[1] 16
```

- There are other control structures as well but are less frequently used than explained above. Those structures are:
  (1) **repeat** – It executes an infinite loop.
  (2) **break** – It breaks the execution of a loop.
  (3) **next** – It allows to skip an iteration in a loop.
  (4) **return** – It help to exit a function.

3.2 SUPERVISED MACHINE LEARNING

- In the Supervised Learning, the training is controlled by an external supervisor or teacher which watches the answer that the network is supposed to generate from a determined input.
- For example, with supervised learning, an algorithm may be fed data with images of sharks labeled as fish and images of oceans labeled as water. By being trained on this data, the supervised learning algorithm should be able to later identify unlabeled shark images as fish and unlabeled ocean images as water.
- A common use case of supervised learning is to use historical data to predict statistically likely future events.
- In supervised learning, the computer is provided with example inputs that are labeled with their desired outputs.
- The purpose of this method is for the algorithm to be able to "learn" by comparing its actual output with the "target" outputs to find errors, and modify the model accordingly.
- So Supervised learning or training is the process of providing the network with a series of sample inputs and comparing the output with the expected responses.
- In fact, the supervised learning is a typical case of pure inductive inference, where the free variables of the network are adjusted by knowing a priori the desired outputs for the investigated system.
- It is mostly used in pattern classification and regression.

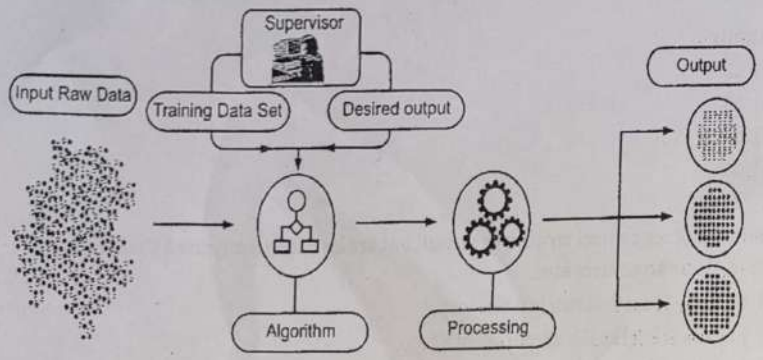


Fig. 3.1 : Supervised Learning

3.2.1 K-Nearest Neighbors (KNN)

- K-Nearest Neighbors is one of the simplest Machine Learning algorithms based Supervised Learning technique.
- KNN algorithm assumes the similarity between the new case/data and available ca and put the new case into the category that is most similar to the available categori
- KNN is called Lazy Learner (Instance based learning). It does not learn anything in training period. It does not derive any discriminative function from the training da
- The K-Nearest Neighbor algorithm (KNN) is a pattern recognition model that can used for classification as well as regression.
- The K in K-nearest neighbor is a positive integer, which is typically small. In eit classification or regression, the input will consist of the K closest training examp within a space. It is commonly used for its easy of interpretation and low calculati time.

KNN for Classification:

- When KNN is used for classification, the output can be calculated as the class with t highest frequency from the K-most similar instances. Each instance in essence vot for their class and the class with the most votes is taken as the prediction.
- Class probabilities can be calculated as the normalized frequency of samples th belong to each class in the set of K most similar instances for a new data instance.
- For example, in a binary classification problem (class is 0 or 1):
  p(class=0) = count(class=0) / (count(class=0)+count(class=1))
- If you are using K and you have an even number of classes (For Example,. 2) it is good idea to choose a K value with an odd number to avoid a tie. And the inverse, u an even number for K when you have an odd number of classes.
- Ties can be broken consistently by expanding K by 1 and looking at the class of t next most similar instance in the training dataset.
- KNN stores the entire training dataset which it uses as its representation.
- KNN does not built any model using the training set until a query of the dataset performed.
- KNN makes predictions just-in-time by calculating the similarity between an inpu sample and each training instance.
- There are many distance measures to choose from to match the structure of you input data. That it is a good idea to rescale your data, such as using normalization when using KNN.

Example:

- The Fig. 3.2 (a) shows the spread of circles (C) and squares (S).

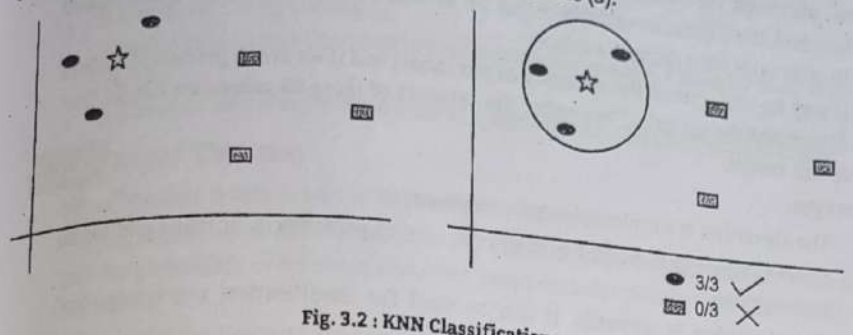


Fig. 3.2 : KNN Classification

- To find out the class of the star (S) can either be C or S and nothing else. The "K" in KNN algorithm is the nearest neighbors we wish to take vote from. Let's say K = 3. Hence, we will now make a circle with S as center just as big as to enclose only three data points on the plane. (Refer to Fig. 3.2 (b))
- The three closest points to S is all C. Hence, with good confidence level we can say that the S should belong to the class C. Here, the choice became very obvious as all three votes from the closest neighbor went to C.
- The choice of the parameter K is very crucial in this algorithm.

K-Nearest Neighbors – Algorithm

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

Step 1 : Load the training and test data.

Step 2 : Choose the value of K.

Step 3 : For each point in test data:

Find the Euclidean distance to all training data points.

Store the Euclidean distances in a list and sort it.

Choose the first k points.

Assign a class to the test point based on the majority of classes present in the chosen points.

Step 4 : End

Implementation in R:

- We will be using the popular Iris dataset for building our KNN model

```
df <- data(iris)      ##load data
head(iris)   ## see the studcture
```

##	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
## 1	5.1	3.5	1.4	0.2	setosa
## 2	4.9	3.0	1.4	0.2	setosa
## 3	4.7	3.2	1.3	0.2	setosa
## 4	4.6	3.1	1.5	0.2	setosa
## 5	5.0	3.6	1.4	0.2	setosa
## 6	5.4	3.9	1.7	0.4	setosa

```
## Generate a random number that is 90% of the total number of rows in
dataset.
ran <- sample(1:nrow(iris), 0.9 * nrow(iris))

## the normalization function is created
nor <-function(x) { (x -min(x))/(max(x)-min(x)) }

##Run normalization on first 4 columns of dataset because they are the
predictors
iris_norm <- as.data.frame(lapply(iris[,c(1,2,3,4)], nor))
summary(iris_norm)
```

##	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
##	Min. :0.0000	Min. :0.0000	Min. :0.0000	Min. :0.00
##	1st Qu. : 0.2222	1st Qu. : 0.3333	1st Qu. : 0.1017	1st Qu. : 0.08
##	Median :0.4167	Median :0.4167	Median :0.5678	Median :0.50
##	Mean :0.4287	Mean :0.4167	Mean :0.4675	Mean :0.45
##	3rd Qu. :0.5833	3rd Qu. :0.5417	3rd Qu. :0.6949	3rd Qu. :0.70
##	Max. :1.0000	Max. :1.0000	Max. :1.0000	Max. :1.00

```
##extract training set
iris_train ← iris_norm[ran,]

##extract testing set
```

```
iris_test ← iris_norm[-ran,]

##extract 5th column of train dataset because it will be used as 'cl'
argument in knn function.

iris_target_category ← iris[ran,5]

##extract 5th column if test dataset to measure the accuracy
iris_test_category ← iris[-ran,5]

##load the package class
library(class)

##run knn function
pr ← knn(iris_train,iris_test,cl=iris_target_category,k=13)

##create confusion matrix
tab ← table(pr,iris_test_category)

##this function divides the correct predictions by total number of
predictions that tell us how accurate the model is
accuracy ← function(x){sum(diag(x)/(sum(rowSums(x)))) * 100}
accuracy(tab)

## [1] 80
```

- In the iris dataset that is already available in R, I have run the k-nearest neighbor algorithm that gave me 80% accurate result.
- First, normalized the data to convert petal.length, sepal.length, petal.width and sepal.length into a standardized 0-to-1 form so that we can fit them into one box (one graph) and also because our main objective is to predict whether a flower is virginica, Versicolor, or setosa and that is why excluded the column 5 and stored it into another variable called *iris_target_category*.
- Then, separated the normalized values into training and testing dataset. Imagine it this way that the values from training dataset are firstly drawn on a graph and after we run KNN function with all the necessary arguments, we introduce testing dataset's values into the graph and calculate Euclidean distance with each and every already stored point in graph.

- Now, although we know which flower it is in testing dataset, we still predict the values and store them in variable called 'pr' so that we can compare predicted values with original testing dataset's values.
- This way we understand the accuracy of our model and if we are to get new 50 values in future and we are asked to predict the category of those 50 values, we can do that with this model.

Advantages:

1. The algorithm is simple and easy to implement.
2. There is no need to build a model, tune several parameters, or make additional assumptions.
3. The algorithm is versatile. It can be used for classification, regression, and search.
4. For nonlinear data no assumptions about data are useful.

Disadvantages:

1. The algorithm gets significantly slower as the number of examples and/or predictors/independent variables increase.
2. Computationally expensive because the algorithm stores all of the training data.
3. High memory requirement for storing the results.
4. It has poor interpretability.

3.2.2 Naïve Bayes Algorithm

- Naïve Bayes algorithm is a Supervised Learning Algorithm.
- It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors.
- In simple terms, a Naïve Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.
- For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.
- Naïve Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.
- The principle behind Naïve Bayes is the Bayes theorem also known as the Bayes Rule.
- It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

Applications:

- It is used in medical data classification.
- It is used for Credit Scoring.
- It can be used in real-time predictions because Naïve Bayes Classifier is an eager learner.
- It is used in Text classification such as Spam filtering and Sentiment analysis.

3.2.2.1 Bayes' Theorem

- Bayes' theorem is also known as **Bayes' Rule** or **Bayes' law.**
- The Bayes theorem is used to calculate the conditional probability, which is nothing but the probability of an event occurring based on information about the events in the past.
- Mathematically, the Bayes theorem is represented as:

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

- In the above equation:
  - P(A|B): Conditional probability of event A occurring, given the event B
  - P(A): Probability of event A occurring
  - P(B): Probability of event B occurring
  - P(B|A): Conditional probability of event B occurring, given the event A
- Formally, the terminologies of the Bayesian Theorem are as follows:
  - A is known as the proposition and B is the evidence
  - P(A) represents the prior probability of the proposition
  - P(B) represents the prior probability of evidence
  - P(A|B) is called the Posterior Probability
  - P(B|A) is the Likelihood Probability
- Therefore, the Bayes theorem can be summed up as:
- Posterior probability= (Likelihood Probability).(Proposition Prior Probability) / Evidence Prior Probability

How Naïve Baye's Algorithm Works?

- A Naïve Bayes is a supervised machine-learning technique/algorithm that uses the Bayes' Theorem, which assumes that features are statistically independent.
- The theorem relies on the naive assumption that input variables are independent of each other, i.e. there is no way to know anything about other variables when given an additional variable.

- **Example:** Consider a training data set of weather and corresponding target variable 'Play' (suggesting possibilities of playing). Now, we need to classify whether players will play or not based on weather condition.
- Let's see the following steps to perform it.

Step 1 : Convert the data set into a frequency table.

Step 2 : Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.

Step 3 : Now, use Naive Bayesian equation to calculate the posterior probability for each class.

The class with the highest posterior probability is the outcome of prediction.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

DataSet

Frequency Table

Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand total	5	9

Likelihood table

Weather	No	Yes		
Overcast		4	= 4/14	0.29
Rainy	3	2	= 5/14	0.35
Sunny	2	3	= 5/14	0.35
All	5	9		
	= 5/14	= 9/14		
	0.36	0.64		

Fig. 3.3: Working of Naïve Bayes' Algorithm

Implementation in R:

Library use: naiveBayes {e1071}

```
## Categorical data only:
data(HouseVotes84)
model <- naiveBayes(Class ~ ., data = HouseVotes84)
predict(model, HouseVotes84[1:10,-1])
predict(model, HouseVotes84[1:10,-1], type = "raw")
pred <- predict(model, HouseVotes84[,-1])
table(pred, HouseVotes84$Class)

## Example of using a contingency table:
data(Titanic)
<- naiveBayes(Survived ~ ., data = Titanic)
predict(m, as.data.frame(Titanic)[,1:3])

## Example with metric predictors:
data(iris)
<- naiveBayes(Species ~ ., data = iris)

## alternatively:
<- naiveBayes(iris[,-5], iris[,5])
table(predict(m, iris[,-5]), iris[,5])
```

3.2.2.2 Advantages and Disadvantages of Naive Baye's Algorithm

Advantages:

1. It is easy and fast to predict class of test data set. It also performs well in multi-class prediction.
2. When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like Logistic Regression and you need less training data.
3. It performs well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).
4. It is effortlessly trainable, even with a small available data set.

Disadvantages:

1. If categorical variable has a category (in test data set), which was not observed training data set, then model will assign a 0 (zero) probability and will be una to make a prediction. This is often known as "Zero Frequency". To solve this, can use the smoothing technique. One of the simplest smoothing technique called Laplace estimation.
2. On the other side, Naive Bayes is also known as a bad estimator, so the probabi outputs from *predict_proba* method are not to be taken too seriously.
3. Another limitation of Naive Bayes is the assumption of independent predict In real life, it is almost impossible that we get a set of predictors which completely independent.

3.2.2.3 Types of Naïve Bayes Model

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

(1) **Gaussian:** The Gaussian model assumes that features follow a norm distribution. This means if predictors take continuous values instead of discre then the model assumes that these values are sampled from the Gaussi distribution.

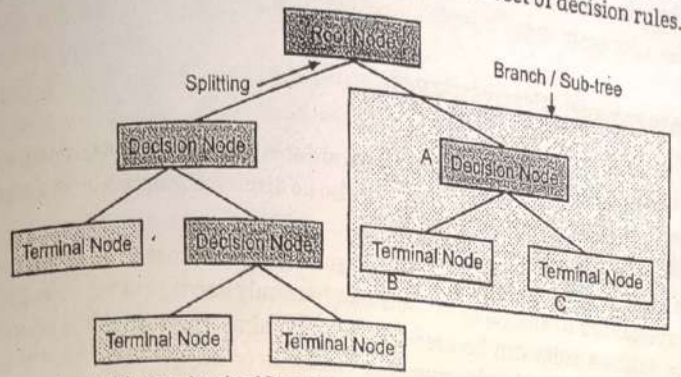
(2) **Multinomial:** The Multinomial Naïve Bayes classifier is used when the data multinomial distributed. It is primarily used for document classificati problems, it means a particular document belongs to which category such Sports, Politics, education, etc. The classifier uses the frequency of words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

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

3.2.3 Decision Tree

- A decision tree (also called prediction tree) uses a tree structure to specify sequence of decisions and consequences (conditions).
- It is called a decision tree because, similar to a tree, it starts with the root node, whic expands on further branches and constructs a tree-like structure.
- In order to build a tree, we use the CART algorithm, which stands for Classificatio and Regression Tree algorithm.
- A decision tree simply asks a question, and based on the answer (Yes/No), it furthe split the tree into subtrees.

- Given input $X = \{x_1, x_i, ..., x_n\}$, the goal is to predict a response or output variable Y. Each member of the set $\{x_1, x_i, ..., x_n\}$ is called an input variable. The prediction can be achieved by constructing a decision tree with test points and branches. At each test point, a decision is made to pick a specific branch and traverse down the tree.
- Eventually, a final point is reached, and a prediction can be made. Each test point in a decision tree involves testing a particular input variable (or attribute), and each branch represents the decision being made. Due to its flexibility and easy visualization, decision trees are commonly deployed in data mining applications for classification purposes.
- The input values of a decision tree can be categorical or continuous. A decision tree employs a structure of test points (called nodes) and branches, which represent the decision being made. A node without further branches is called a leaf node.
- The leaf nodes return class labels and, in some implementations, they return the probability scores. A decision tree can be converted into a set of decision rules.



Note: A is parent node of B and C

Fig. 3.4: Decision Tree

Decision Tree Terminologies:

- **Root Node:** Root Node represents the entire population or sample. It further gets divided into two or more homogeneous sets.
- **Splitting:** Splitting is a process of dividing a node into two or more sub-nodes.
- **Decision Node:** When a sub-node splits into further sub-nodes, it is called a Decision Node.
- **Terminal Node :** Nodes that do not split is called a Terminal Node or a Leaf.
- **Pruning:** When you remove sub-nodes of a decision node, this process is called Pruning. The opposite of pruning is Splitting.

- **Branch/subtree:** A sub-section of an entire tree is called Branch.
- **Parent Node/ Child node:** A node, which is divided into sub-nodes is called a parent node of the sub-nodes; whereas the sub-nodes are called the child of the parent node.

3.2.3.1 How does the Decision Tree algorithm work?

- The basic idea behind any decision tree algorithm is as follows:

Step 1 : Select the best attribute using **Attribute Selection Measure(ASM)** to split the records.

Step 2 : Make that attribute a decision node and breaks the dataset into smaller subsets.

Step 3 : Starts tree building by repeating this process recursively for each child until one of the condition will match:

- All the tuples belong to the same attribute value.
- There are no more remaining attributes.
- There are no more instances.

Attribute Selection Measures:

- Attribute selection measure is a heuristic for selecting the splitting criterion that partition data into the best possible manner. It is also known as splitting rules because it helps us to determine breakpoints for tuples on a given node. ASM provides a rank to each feature (or attribute) by explaining the given dataset.
- Best score attribute will be selected as a splitting attribute. In the case of a continuous-valued attribute, split points for branches also need to define. Most popular selection measures are Information Gain and Gini Index.

1. **Information Gain:**

- When we use a node in a decision tree to partition the training instances into smaller subsets the entropy changes. Information gain is a measure of this change in entropy.
- Information gain (IG) measures how much "information" a feature gives us about the class:
  - Features that perfectly partition should give maximal information.
  - Unrelated features should give no information.
  - It measures the reduction in entropy.
- Definition: Suppose S is a set of instances, A is an attribute, S_v is the subset of S with A = v, and Values (A) is the set of all possible values of A, then

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \cdot \text{Entropy}(S_v)$$

Entropy:

- Entropy is the measure of uncertainty of a random variable, it characterizes the impurity of an arbitrary collection of examples. The higher the entropy, more the information content.
- **Definition:** Suppose S is a set of instances, A is an attribute, Sv is the subset of S with A = v, and Values
  (A) is the set of all possible values of A, then

$$\text{Gain (S, A = Entropy (S)} - \sum_{v \in \text{Values(A)}} \frac{|S_v|}{|S|} \cdot \text{Entropy } (S_v)$$

- **Classification Error Rate:** Rather than seeing how far a numerical response is away from the mean value, as in the regression setting, you can instead define the "hit rate" as the fraction of training observations in a particular region that don't belong to the most widely occurring class. The error is given by this equation:

$$E = 1 - \text{argmax}_c \ (\hat{\pi}_{mc})$$

  in which π^mc represents the fraction of training data in region Rm that belong to class c.
- **Gini Index:** The Gini Index is an alternative error metric that is designed to show how "pure" a region is. "Purity" in this case means how much of the training data in a particular region belongs to a single class. If a region Rm contains data that is mostly from a single class c then the Gini Index value will be small:

$$G = \sum_{c=1}^{C} \hat{\pi}_{mc} (1 - \hat{\pi}_{mc})$$

- **Cross-Entropy:** A third alternative, which is similar to the Gini Index, is known as the Cross-Entropy or Deviance:

$$G = \sum_{c=1}^{C} \hat{\pi}_{mc} (1 - \hat{\pi}_{mc})$$

Implementation in R:

- The R package "party" is used to create decision trees.

```
install.packages("party")

# Load the party package. It will automatically load other
# dependent packages.
library(party)

# Create the input data frame.
```



```
input.dat <- readingSkills[c(1:105),]

# Give the chart file a name.
png(file = "decision_tree.png")

# Create the tree.
output.tree <- ctree(  nativeSpeaker ~ age + shoeSize + score,  data =
input.dat)

# Plot the tree.
plot(output.tree)

# Save the file.
dev.off()
```

3.2.3.2 Advantages and Disadvantages

Advantages:

1. **Simple to understand and interpret:** People are able to understand decision tree models after a brief explanation. Trees can also be displayed graphically in a way that is easy for non-experts to interpret.
2. **Able to handle both numerical and categorical data:** Other techniques are usually specialized in analyzing datasets that have only one type of variable. (For example, relation rules can be used only with nominal variables while neural networks can be used only with numerical variables or categorical converted to 0-1 values.)
3. **Requires little data preparation:** Other techniques often require data normalization. Since trees can handle qualitative predictors, there is no need to create dummy variables.
4. **Performs well with large datasets:** Large amounts of data can be analyzed using standard computing resources in reasonable time.
5. **Fast and Accurate:** It is fast, accurate (compared to other classification techniques), simple and inexpensive to construct and classify unknown records.

Disadvantages:

1. **Trees can be very non-robust:** A small change in the training data can result in a larger change in the tree and consequently the final predictions.
2. **Poor performance:** Decision trees performance is not good if there are lots of uncorrelated variables in the data set.

3. **Expensive:** Computationally expensive to train as it forms many subtrees, which are compared.

3.2.4 Support Vector Machines

- A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimensional space, this hyperplane is a line dividing a plane in two parts where in each class lay in either side.
- The Support Vector Machine (SVM) is a classification technique based on statistical learning theory, and can be applied in many challenging non-linear classification problems with large data sets.
- The SVM is originally binary classification method developed by Vapnik and his co-workers at Bell laboratories. The SVM algorithm finds a hyper plane that optimally splits the training set.
- The optimal hyper plane can be distinguished by the maximum margin of separation between all training points and the hyper plane. Support vector machines (SVM) are a group of supervised learning methods that can be applied to classification or regression.
- The standard SVM classifier takes the set of input data and predicts to classify them in one of the only two distinct classes. The discrimination between the two classes is achieved by defining a separating hyper plane.
- The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N – the number of features) that distinctly classifies the data points.

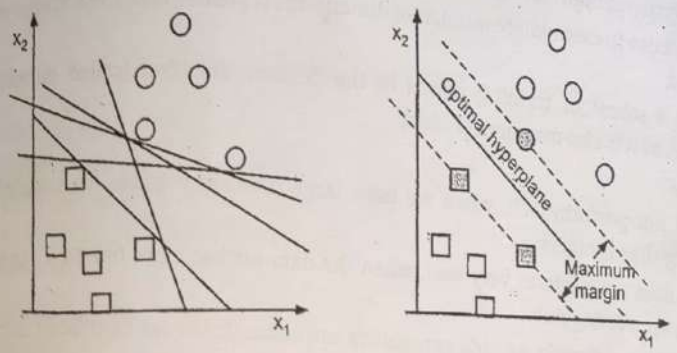


Fig. 3.5: Possible Hyperplanes

- To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e. the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.
- Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features.
- If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds 3.

$$x + b = 0, \text{ where } w = \text{weight vector and } b = \text{bias}$$

- The hypothesis space is the set of functions,

$$f(x,w,b) = \text{sign}(w \cdot x + b),$$

which classify data points based on whether the quantity $w \cdot x + b$ is positive or negative. The distance d+ and d- are the distance of positive data points from the hyper plane and the distance of negative data points from the hyper plane respectively.

Objective is to maximize:

$$\text{Margin} = \frac{2}{\|\vec{w}\|}$$

Which is equivalent to minimizing:

$$L\vec{w} = \frac{\|\vec{w}\|^2}{2}$$

Subject to the following constraints:

$$y_i = \begin{cases} 1 & \text{if } \vec{w} \cdot \vec{x_i} + b \geq 1 \\ -1 & \text{if } \vec{w} \cdot \vec{x_i} + b \leq -1 \end{cases}$$

- Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.

SVM Implementation in R:

```
x=c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20)
y=c(3,4,5,4,8,10,10,11,14,20,23,24,32,34,35,37,42,48,53,60)
```

```
#Create a data frame of the data
train=data.frame(x,y)

#Plot the dataset
plot(train,pch=16)

#Linear regression
model <- lm(y ~ x, train)

#Plot the model using abline
abline(model)

#SVM
library(e1071)

#Fit a model. The function syntax is very similar to lm function
model_svm <- svm(y ~ x, train)

#Use the predictions on the data
pred <- predict(model_svm, train)

#Plot the predictions and the plot to see our model fit
points(train$x, pred, col = "blue", pch=4)

#Linear model has a residuals part which we can extract and directly
calculate rmse
error <- model$residuals
lm_error <- sqrt(mean(error^2)) # 3.832974

#For svm, we have to manually calculate the difference between actual
values (train$y) with our predictions (pred)
error_2 <- train$y - pred
svm_error <- sqrt(mean(error_2^2)) # 2.696281

# perform a grid search
svm_tune <- tune(svm, y ~ x, data = train,
```



```
ranges = list(epsilon = seq(0,1,0.01), cost = 2^(2:9))
print(svm_tune)
#Parameter tuning of 'svm':
# - sampling method: 10-fold cross validation
#- best parameters:
# epsilon cost
# 0 8

#- best performance: 2.872047
#The best model
best_mod <- svm_tune$best.model
best_mod_pred <- predict(best_mod, train)
error_best_mod <- train$y - best_mod_pred

# this value can be different on your computer
# because the tune method randomly shuffles the data
best_mod_RMSE <- sqrt(mean(error_best_mod^2)) # 1.290738
plot(svm_tune)
plot(train,pch=16)
points(train$x, best_mod_pred, col = "blue", pch=4)
```

3.2.4.1 Advantages and Disadvantages of SVM

Advantages:

1. It works really well with clear margin of separation.
2. It is effective in high dimensional spaces.
3. It is effective in cases where number of dimensions is greater than the number of samples.
4. It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

Disadvantages:

1. It does not perform well, when we have large data set because the required training time is higher.
2. It also does not perform very well, when the data set has more noise i.e. target classes are overlapping.
3. SVM does not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

3.3 UNSUPERVISED MACHINE LEARNING

- In the unsupervised learning, there is no teacher or supervisor to oversee the process.
- In this type, the network is provided with inputs but not with desired outputs.
- In unsupervised learning, data is unlabeled, so the learning algorithm is left to find commonalities among its input data. As unlabeled data are more abundant than labeled data, machine learning methods that facilitate unsupervised learning are particularly valuable.
- The system itself must then decide which features it will use to group the input data.
- The training process extracts the statistical properties of the training set and groups of similar vectors into classes or clusters.
- Unsupervised learning is often used for anomaly detection including for fraudulent credit card purchases, and recommender systems that recommend what products to buy next. In unsupervised learning, untagged photos of dogs can be used as input data for the algorithm to find likenesses and classify dog photos together.
- Algorithm might emphasize cooperation among clusters of processing elements.
- Unsupervised learning mostly used in application like clustering, anomaly detection etc.

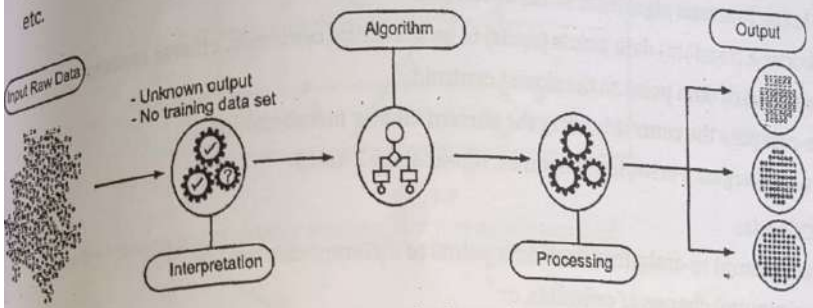


Fig. 3.6: Unsupervised Learning

- Unsupervised machine learning is the task of drawing inference of a function from data set containing input without labeled data or target value to describe hidden patterns from unlabeled data. The most common unsupervised machine learning method is hierarchical cluster analysis. It is used for exploratory data analysis in order to find the hidden patterns.
- **Examples:** Some examples of unsupervised learning applications include:
  (a) In marketing segmentation, when a company wants to segment its customers to better adjust products and offerings.

(b) Social network analysis.
(c) Image Segmentation.
(d) Google news.

3.3.1 Cluster Analysis

- Cluster analysis is a staple of unsupervised machine learning and data science.
- It is very useful for data mining and big data because it automatically finds patterns in the data, without the need for labels, unlike supervised machine learning.
- Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups.
- In simple words, the aim is to segregate groups with similar traits and assign them into clusters.
- Let's understand this with an example. Suppose, you are the head of a rental store and wish to understand preferences of your costumers to scale up your business. Is it possible for you to look at details of each costumer and devise a unique business strategy for each one of them? Definitely not. But, what you can do is to cluster all of your costumers into say 10 groups based on their purchasing habits and use a separate strategy for costumers in each of these 10 groups. This is called clustering.
- Cluster analysis is a data explorative technique aiming to divide a multivariate dataset into clusters (groups) based on a set of measured variables in same group, which have similar subjects. Cluster analysis played an important tool in data analysis in a wide variety of fields including medicine, pharmaceutical science, biological science, statistics, and computer and information science.
- Now, that we understand what is clustering. Let's take a look at the types of clustering.

Types of Clustering:

- Broadly speaking, clustering can be divided into two subgroups:
  (a) **Hard Clustering:** In Hard Clustering, each data point either belongs to a cluster completely or not. For example, in the above example each customer is put into one group out of the 10 groups.
  (b) **Soft Clustering:** In Soft Clustering, instead of putting each data point into a separate cluster, a probability or likelihood of that data point to be in those clusters is assigned. For example, from the above scenario each costumer is assigned a probability to be in either of 10 clusters of the retail store.

3.3.1.1 Types of Clustering Algorithms

- Since the task of clustering is subjective, the means that can be used for achieving this goal are plenty. Every methodology follows a different set of rules for defining the 'similarity' among data points. In fact, there are more than 100 clustering algorithms known.
- But few of the algorithms are used popularly, let's look at them in detail:

(1) Connectivity Models:

- As the name suggests, these models are based on the notion that the data points closer in data space exhibit more similarity to each other than the data points lying farther away.
- These models can follow two approaches. In the first approach, they start with classifying all data points into separate clusters and then aggregating them as the distance decreases.
- In the second approach, all data points are classified as a single cluster and then partitioned as the distance increases. Also, the choice of distance function is subjective.
- These models are very easy to interpret but lacks scalability for handling big datasets. Examples of these models are hierarchical clustering algorithm and its variants.

(2) Centroid Models:

- These are iterative clustering algorithms in which the notion of similarity is derived by the closeness of a data point to the centroid of the clusters.
- K-Means clustering algorithm is a popular algorithm that falls into this category. In these models, the no. of clusters required at the end has to be mentioned beforehand, which makes it important to have prior knowledge of the dataset. These models run iteratively to find the local optima.

(3) Distribution Models:

- These clustering models are based on the notion of how probable is it that all data points in the cluster belong to the same distribution (For example: Normal, Gaussian). These models often suffer from overfitting.
- A popular example of these models is Expectation-maximization algorithm which uses multivariate normal distributions.

(4) Density Models: These models search the data space for areas of varied density of data points in the data space.

- It isolates various different density regions and assigns the data points within these regions in the same cluster.

3.3.2 K-Means

- In terms of telephone example, hanging up the phone is the most important job.
- K-means clustering is one of the simplest and popular unsupervised ma learning algorithms.
- This algorithm involves you telling the algorithms how many possible cluster there are in the dataset.
- The algorithm then iteratively moves the k-centers and selects the data points are closest to that centroid in the cluster.
- K-means is a partitional clustering algorithm.
- Let the set of data points D be $\{X_1, X_2, \ldots, X_n\}$, where $x_i = (x_{i1}, x_{i2}, \ldots, x_{ir})$ is a vector real-valued space $X \subseteq R^r$ and r is the number of dimensions of the data space.
- The k-means algorithm partitions the given data into k clusters:
  - Each cluster has a cluster center, called centroid.
  - k is specified by the user.

3.3.2.1 K-Means Algorithm

- Given k, the K-means algorithm works as follows:
  (1) Choose k (random) data points (seeds) to be the initial centroids, cluster center
  (2) Assign each data point to the closest centroid.
  (3) Re-compute the centroids using the current cluster memberships.
  (4) If a convergence criterion is not met, repeat steps 2 and 3.

Stopping Criteria:

- No (or minimum) re-assignments of data points to different clusters, or
- No (or minimum) change of centroids, or
- Minimum decrease in the Sum of Squared Error (SSE)

$$\sum_{j=1}^{k} \sum_{x \in C_j} d(x, m_j)^2$$

Where, C_j is the j^{th} cluster.

m_j is the centroid of cluster C_j (the mean vector of all the data points in C_j).

$d(x, m_j)$ is the (Eucledian) distance between data point x.

K-means clustering example:

Step 1 : Randomly initialize the cluster centers (synaptic weights)

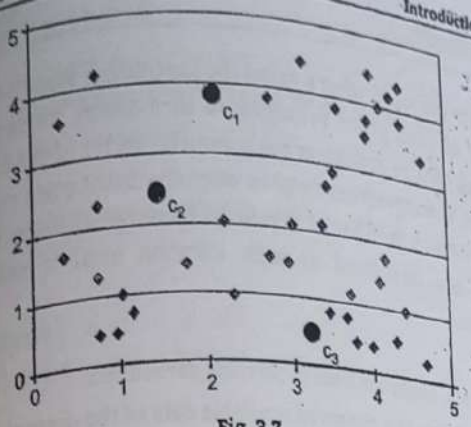


Fig. 3.7

Step 2 : Determine cluster membership for each input ("winner-takes-all" inhibitory circuit)

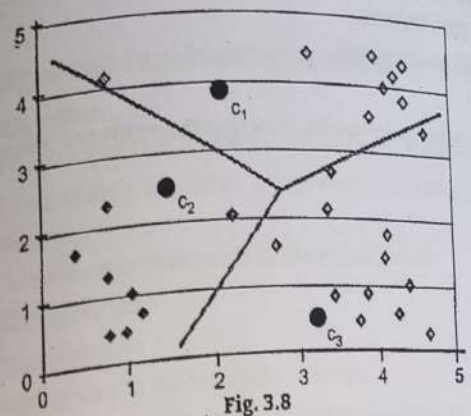


Fig. 3.8

Step 3 : Re-estimate cluster centers (adapt synaptic weights)

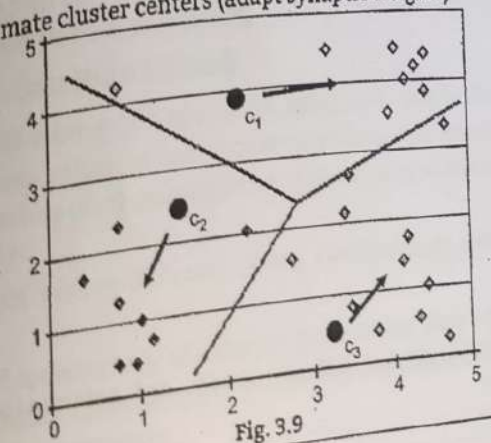


Fig. 3.9

Result of First Iteration:

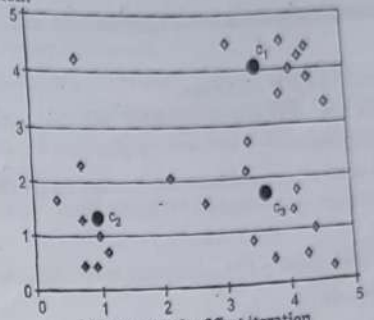


Fig. 3.10 : Result of first iteration

Second Iteration:

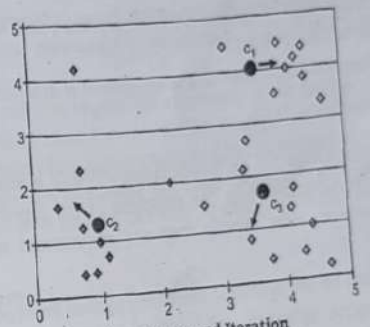


Fig. 3.11 : Second Iteration

Results of second Iteration:

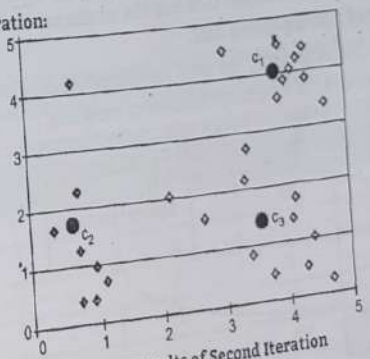


Fig. 3.12 : Results of Second Iteration

3.3.2.2 Advantages and Disadvantages of K-means algorithm

Advantages:

1. Simple easy to understand and to implement.
2. K-means is the most popular clustering algorithm, because it provides easily interpretable clustering results.
3. Fast and efficient in terms of computational cost. Excellent for pre-clustering in comparison to other clustering algorithms.

Disadvantages:

1. The algorithm is only applicable if the mean is defined.
2. For categorical data, k-mode.
   (i) The centroid is represented by most frequent values.
   (ii) The algorithm is sensitive to outliers (Outliers are data points that are very far away from other data points).
3. The algorithms are slow and do not scale to a large number of data points.

3.3.3 EM Algorithm

- The EM algorithm is a methodology for algorithm construction, it is not a specific algorithm. Each problem is different, only the structure of the Expectation and Maximization steps is common. How exactly they are programmed is problem dependent.
- The EM algorithm can be seen an unsupervised clustering method based on mixture models. It follows an iterative approach, sub-optimal, which tries to find the parameters of the probability distribution that has the maximum likelihood of its attributes in the presence of missing/latent data.

The algorithm's input are the data set X, the total number of clusters/models K, the accepted error to converge ϵand the maximum number of iterations.

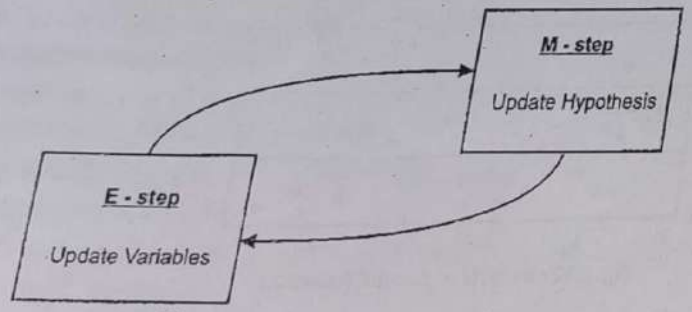


Fig. 3.13: Expectation-Maximization Algorithm

- For each iteration, first it is executed what's called the Expectation Step (E-step), that estimates the probability of each point belonging to each model, followed by the Maximization step (M-step), that re-estimates the parameter vector of the probability distribution of each model. The algorithm finishes when the distribution parameters converge or reach the maximum number of iterations. Convergence is assured since the algorithm increases the likelihood at each iteration until it reaches the (eventually local) maximum.

Algorithm:

- Given a set of incomplete data, consider a set of starting parameters.
- Expectation step (E – step): Using the observed available data of the dataset, estimate (guess) the values of the missing data.
- Maximization step (M – step): Complete data generated after the expectation (E) step is used in order to update the parameters.
- Repeat step 2 and step 3 until convergence.

Implementation in R

- EM clustering with diabetes data using Mclust. Note that the summary command with an Mclust object generates:

```
data(diabetes)
head(diabetes)

class.d = diabetes$class
table(class.d)

X = diabetes[,-1]
head(X)

clPairs(X, class.d)

fit <- Mclust(X)
fit

## 'Mclust' model object:
##  best model: ellipsoidal, varying volume, shape, and orientation (VVV)
with 3 components
summary(fit)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVV (ellipsoidal, varying volume, shape, and orientation) model
with 3 components:
##
##  log.likelihood   n df       BIC       ICL
##       -2307.883 145 29 -4760.091 -4776.086
##
##
## Clustering table:
##  1  2  3
## 82 33 30
```

3.3.3.1 Advantages and Disadvantages of EM Algorithm

Advantages of EM algorithm :

(1) It is always guaranteed that likelihood will increase with each iteration.

(2) The E-step and M-step are often pretty easy for many problems in terms of implementation.

(3) Solutions to the M-steps often exist in the closed form.

Disadvantages of EM algorithm :

(1) It has slow convergence.

(2) It makes convergence to the local optima only.

(3) It requires both the probabilities, forward and backward (numerical optimization requires only forward probability).

3.3.4 Association Rule Mining

- Association rule mining is a procedure which is meant to find frequent patterns, correlations, associations, or causal structures from data sets found in various kinds of databases such as relational databases, transactional databases, and other forms of data repositories.
- Association rule mining discovers strong association or correlation relationships among data.
- Given a set of transactions, association rule mining aims to find the rules which enable us to predict the occurrence of a specific item based on the occurrences of the other items in the transaction.

Definition:

- Following the original definition by Agrawal, Imielinski and Swami the problem of Association rule mining is defined as:

  Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of n binary attributes called items.

  Let $D = \{t_1, t_2, \ldots, t_m\}$ be a set of transactions called the database.

  Each transaction in D has a unique transaction ID and contains a subset of the items in I.

  A rule is defined as an implication of the form:

  $$X \rightarrow Y, \text{ where } X, Y, \subseteq I$$

- In Agrawal, Imielinski, Swami definition, a rule is defined only between a set and a single item,

  $$X \rightarrow i_j \text{ or } i_j \in I$$

- Every rule is composed by two different sets of items, also known as **itemsets**, X and Y, where X is called **antecedent** or left-hand-side (LHS) and Y **consequent** or right-hand-side (RHS).
- An association rule is an implication expression of the form $X \longrightarrow Y$, where X and Y are disjoint itemsets, i.e., $X \cap Y = \phi$. The strength of an association rule can be measured in terms of its support and confidence. Support determines how often a rule is applicable to a given data set, while confidence determines how frequently items in Y appear in transactions that contain X. The formal definitions of these metrics are:

$$\text{Support, } s(X \longrightarrow Y) = \frac{\sigma(X \cup Y)}{N}$$

$$\text{Confidence, } c(X \longrightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

Example :

- An example rule for the supermarket could be **{butter, bread} → {milk}** meaning that if butter and bread are bought, customers also buy milk.

Table 3.1: Supermarket Database with 5 transactions and 5 items

Transaction ID	Milk	Bread	Butter	Juice	Sauce
1.	1	1	0	0	0
2.	0	0	1	0	0
3.	0	0	0	1	1
4.	1	1	1	0	0
5.	0	1	0	0	0

3.3.4.1 Applications of Association Rule Mining

- The main applications of Association Rule Mining:
  (1) **Basket Data Analysis** is to analyze the association of purchased items in a single basket or single purchase, for example, peanut butter and jelly are often bought together because a lot of people like to make PB and J sandwiches.
  (2) **Cross Marketing** is to work with other businesses that complement your own, not competitors. For example, vehicle dealerships and manufacturers have cross marketing campaigns with oil and gas companies for obvious reasons.
  (3) **Catalog Design** the selection of items in a business' catalog are often designed to complement each other so that buying one item will lead to buying of another. So these items are often complements or very related.
- Association rule mining discovers strong association or correlation relationships among data.
- Association rule learning is a rule-based machine learning method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using some measures of interestingness.

3.3.4.2 Useful Concepts of Association Rule Mining

- In order to select interesting rules among all existing ones, many constraints are applied. There are certain important concepts related to Associative Rule Learning such as Support, Confidence and Lift.
- Let X be an itemset, $X \rightarrow Y$ an association rule and T a set of transactions of a given database.

1. **Support**

- Support is an indication of how frequently the itemset appears in the dataset.
- The support of X with respect to T is defined as the proportion of transactions t in the dataset which contains the itemset X.

$$\text{Supp}(X) = \frac{|\{t \in T; X \subseteq t\}|}{|T|}$$

- In the example dataset, the itemset X = {juice, sauce} has a support of 1/5 = 0.2 since it occurs in 20% → of all transactions (1 out of 5 transactions). The argument of supp( ) is a set of preconditions, and thus becomes more restrictive as it grows (instead of more inclusive).

2. **Confidence:**

- Confidence is an indication of how often the rule has been found to be true.

- The confidence value of a rule, $X \rightarrow Y$, with respect to a set of transactions T, is the proportion of the transactions that contains X which also contains Y.
- Confidence is defined as:

$$\text{conf}(X \rightarrow Y) = \text{supp}(X \cup Y)/\text{supp}(X)$$

- For example, the rule {butter, bread}→{milk} has a confidence of 0.2 /0.2= 1.0 in the database, which means that for 100% of the transactions containing butter and bread the rule is correct (100% of the times a customer buys butter and bread, milk is bought as well).
- Note that supp $(X \cup Y)$ means the support of the union of the items in X and Y. This is somewhat confusing since we normally think in terms of probabilities of events and not sets of items. We can rewrite supp $(X \cup Y)$ as the probability $P(E_X \cap E_Y)$, where E_X and E_Y are the events that a transaction contains itemset X and Y, respectively.
- Thus confidence can be interpreted as an estimate of the conditional probability $P(E_Y \cap E_X)$, the probability of finding the RHS of the rule in transactions under the condition that these transactions also contain the LHS.

3. **Lift:**

- The lift of a rule is defined as:

$$\text{lift}(X \Rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X) \times \text{supp}(Y)}$$

or the ratio of the observed support to that expected if X and Y were independent.

- For example, the rule {milk, bread} → {butter} has a lift of $\frac{0.2}{0.4 \times 0.4} = 1.25$
- If the rule had a lift of 1, it would imply that the probability of occurrence of the antecedent and that of the consequent are independent of each other. When two events are independent of each other, no rule can be drawn involving those two events.
- If the lift is > 1, that lets us know the degree to which those two occurrences are dependent on one another, and makes those rules potentially useful for predicting the consequent in future data sets.
- If the lift is < 1, that lets us know the items are substitute to each other. This means that presence of one item has negative effect on presence of other item and vice versa.
- The value of lift is that it considers both the support of the rule and the overall data set.

3.5 Apriori Algorithms

The Apriori algorithm was proposed by Agrawal and Srikant in 1994. Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation.

Other algorithms are designed for finding association rules in data having no transactions (Winepi and Minepi), or having no timestamps (DNA sequencing). Each transaction is seen as a set of items (an itemset). Given a threshold C, the Apriori algorithm identifies the item sets which are subsets of at least C transactions in the database.

Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found.

Apriori uses breadth-first search and a Hash tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length k-1. Then it prunes the candidates which have an infrequent sub pattern.

According to the downward closure lemma, the candidate set contains all frequent k-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.

Table 3.2 : Notations of Apriori

k-itemset	An itemset having k items
L_k	Set of large k-item sets (those with minimum support) Each member of this set has two fields: (i) itemset (ii) support count
C_k	Set of candidate k-itemsets (potentially large itemsets). Each member of this set has two fields: (i) itemset (ii) support count

3.5.1 The Apriori Algorithm

The algorithm is shown below:

Step 1: L_1 = {large 1-itemsets};

Step 2: For ($k = 2; L_k - 1 \neq \varphi k$++) do begin

Step 3: C_k = apriori-gen (L_{k-1}); //New candidates

Step 4: For all transactions $t \in D$ do begin.

Step 5: C_t = subset(C_k, t); //Candidates contained in t

Step 6: For all candidates $c \in C_t$ do

Step 7: c.count++;

Step 8: end

Step 9: $L_k = \{c \in C_k \mid c.count \geq minsup\}$

Step 10: End

Answer = $U_k L_k$;

- The first pass of the algorithm simply counts the number of occurrences of each item to determine the large 1-itemsets. A subsequent pass, say pass k, consists of two phases. First, the large itemsets L_{k-1}, found in the $(k-1)^{th}$ pass are used to generate the candidate itemsets C_k, using the apriori-gen function described in the following section. Next, the database is scanned and the support of candidates in C_k is counted. The last step of the algorithm is to prune those candidates in C_k whose support is less than minimum support, hence generating L_k.

Example 1: Consider the following database, where each row is a transaction and each cell is an individual item of the transaction:

alpha	beta	epsilon
alpha	beta	theta
alpha	beta	epsilon
alpha	beta	theta

- The association rules that can be determined from this database are the following:
  (1) 100% of sets with alpha also contain beta.
  (2) 50% of sets with alpha, beta also have epsilon.
  (3) 50% of sets with alpha, beta also have theta.

Implementation Using R:

```
install.packages ("arules")
install.packages ("arulesViz")
```

We use dataset that comes bundled with the arules package.

```
>data("Groceries")

>class(Groceries)
        [1] "transactions"
        attr(,"package")
         [1] "arules"
>inspect(head(Groceries, 2))
items
[1] {citrus fruit, semi-finished bread, margarine, ready soups}
[2] {tropical fruit, yogurt, coffee}

//Lets find out the rules using the apriori algorithm.
> grocery_rules <- apriori(Groceries, parameter = list(support = 0.01,
confidence = 0.5))

Apriori Parameter specification:
confidence minval smax arem  aval originalSupport maxtime support minlen
maxlen target   ext
0.5      0.1    1    none   FALSE          TRUE       5     0.01         1
10      rules FALSE
Algorithmic control:
Filter tree  heap memopt load sort verbose
  0.1  TRUE  TRUE  FALSE  TRUE  2  TRUE

Absolute minimum support count:            98
    set item appearances ...[0 item(s)] done [0.00s].
    set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
    sorting and recoding items ... [88 item(s)] done [0.00s].
    creating transaction tree ... done [0.00s].
    checking subsets of size 1 2 3 4 done [0.00s].
    writing ... [15 rule(s)] done [0.00s].
    creating S4 object  ... done [0.00s].
```

Parameter Specification of the output:

- `minval`: the minimum value of the support an itemset should satisfy to be a part of a rule.
- `smax`: the maximum support value for an itemset.

- `arem`: an Additional Rule Evaluation Parameter. In the above code, we have constrained the number of rules using Support and Confidence.
- `aval`: a logical indicating whether to return the additional rule evaluation measure selected with arem.
- `originalSupport`: The traditional support value only considers both LHS and RHS items for calculating support. If you want to use only the LHS items for the calculation then you need to set this to FALSE.
- `maxtime`: is the maximum amount of time allowed to check for subsets.
- `minlen`: is the minimum number of items required in the rule.
- `maxlen`: is the maximum number of items that can be present in the rule.

Three rules sorted by confidence are shown below :

```
> inspect(head(sort(rules, by = "confidence"), 3))
    Lhs rhs support    confidence lift     count
[1] {citrus fruit,root vegetables} => {other vegetables} 0.0103711
0.5862069  3.029608 102
[2] {tropical fruit,root vegetables} => {other vegetables} 0.0123983
0.5845411  3.020999 121
[3] {curd,yogurt} => {whole milk} 0.01006609 0.5823529  2.279125  9
```

3.3.5.2 Theory of Apriori Algorithm

- There are three major components of Apriori algorithm namely Support, Confidence and Lift.

We will explain these three concepts with the help of an example.

- Suppose we have a record of 1 thousand customer transactions, and we want to find the Support, Confidence, and Lift for two items e.g. burgers and ketchup. Out of one thousand transactions, 100 contain ketchup while 150 contain a burger. Out of 150 transactions where a burger is purchased, 50 transactions contain ketchup as well. Using this data, we want to find the support, confidence, and lift.

(1) Support:

- Support refers to the default popularity of an item and can be calculated by finding number of transactions containing a particular item divided by total number of transactions. Suppose we want to find support for item B. This can be calculated as:

$$\text{Support}(B) = \frac{(\text{Transactions containing}(B))}{(\text{Total Transactions})}$$

- For instance if out of 1000 transactions, 100 transactions contain Ketchup then the support for item Ketchup can be calculated as:

$$\text{Support(Ketchup)} = \frac{\text{(Transactions containing Ketchup)}}{\text{(Total Transactions)}}$$

$$\text{Support(Ketchup)} = \frac{100}{1000} = 10\%$$

(2) Confidence:

- Confidence refers to the likelihood that an item B is also bought if item A is bought. It can be calculated by finding the number of transactions where A and B are bought together, divided by total number of transactions where A is bought. Mathematically, it can be represented as:

$$\text{Confidence } (A \rightarrow B) = \frac{\text{(Transactions containing both(A and B))}}{\text{(Transactions containing A)}}$$

- Coming back to our problem, we had 50 transactions where Burger and Ketchup were bought together. While in 150 transactions, burgers are bought. Then we can find likelihood of buying ketchup when a burger is bought can be represented as confidence of Burger -> Ketchup and can be mathematically written as:

$$\text{Confidence (Burger} \rightarrow \text{Ketchup)} = \frac{\text{(Transactions containing both(Burger and Ketchup))}}{\text{(Transactions containing A)}}$$

$$\text{Confidence (Burger} \rightarrow \text{Ketchup)} = \frac{50}{150} = 33.3\%$$

(3) Lift:

- Lift(A → B) refers to the increase in the ratio of sale of B when A is sold. Lift(A –B) can be calculated by dividing Confidence(A → B) divided by Support(B). Mathematically it can be represented as:

$$\text{Lift } (A \rightarrow B) = \frac{\text{(Confidence}(A \rightarrow B))}{\text{(Support (B))}}$$

- Coming back to our Burger and Ketchup problem, the Lift(Burger →Ketchup) can be calculated as:

$$= \frac{\text{(Confidence(Burger} \rightarrow \text{Ketchup))}}{\text{(Support (Ketchup))}}$$

$$= \frac{33.3}{10} = 3.33$$

- Lift basically tells us that the likelihood of buying a Burger and Ketchup together is 3.33 times more than the likelihood of just buying the ketchup. A Lift of 1 means there

is no association between products A and B. Lift of greater than 1 means products A and B are more likely to be bought together. Finally, Lift of less than 1 refers to the case where two products are unlikely to be bought together.

Steps Involved in Apriori Algorithm:

- For large sets of data, there can be hundreds of items in hundreds of thousands transactions. The Apriori algorithm tries to extract rules for each possible combination of items.
- For instance, Lift can be calculated for item 1 and item 2, item 1 and item 3, item 1 and item 4 and then item 2 and item 3, item 2 and item 4 and then combinations of items e.g. item 1, item 2 and item 3; similarly item 1, item2, and item 4, and so on.
- As you can see from the above example, this process can be extremely slow due to the, number of /combinations. To speed up the process, we need to perform the following steps:

Step 1: Set a minimum value for support and confidence. This means that we are only interested in finding rules for the items that have certain default existence (e.g. support) and have a minimum value for co-occurrence with other items (e.g. confidence).

Step 2: Extract all the subsets having higher value of support than minimum threshold.

Step 3: Select all the rules from the subsets with confidence value higher than minimum threshold.

Step 4: Order the rules by descending order of Lift.

3.3.5.3 Advantages and Disadvantages of Apriori

Advantages of Apriori:

(1) Efficient and effective while dealing with large item sets.
(2) Easily implementable.
(3) Provides real insight on affinity promotion (which product customers may purchase based on analysis of his previous purchase behavior).

Disadvantages of Apriori:

(1) Assumes transitional database is memory resident.
(2) Requires many database scans.
(3) Slow.
(4) Higher runtime for execution.

3.4 REGRESSION ANALYSIS

- **Regression analysis** is a set of statistical processes for **estimating** the relationships among variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a **dependent variable** and one or more **independent variables** (or 'predictors').
- More specifically, regression analysis helps one understand how the typical value of the dependent variable (or 'criterion variable') changes when any one of the independent variables is varied, while the other independent variables are held fixed.
- Regression analysis is widely used for prediction and forecasting, where its use has substantial overlap with the field of machine learning.
- Regression analysis is also used to understand which among the independent variables are related to the dependent variable, and to explore the forms of these relationships.
- It is a very powerful technique and can be used to understand the factors that influence profitability. It can be used to forecast sales in the coming months by analyzing the sales data for previous months. It can also be used to gain various insights about customer behaviour.

3.4.1 Linear Regression

- The objective of a linear regression model is to find a relationship between one or more features (independent variables) and a continuous target variable(dependent variable).
- When there is only feature it is called Uni-variate Linear Regression and if there are multiple features, it is called Multiple Linear Regression.
- Simple linear regression is an approach for predicting a response using a single feature.
- It is assumed that the two variables are linearly related. Hence, we try to find a linear function that predicts the response value(y) as accurately as possible as a function of the feature or independent variable(x).
- In Linear Regression these two variables are related through an equation, where exponent (power) of both these variables is 1. Mathematically a linear relationship represents a straight line when plotted as a graph. A non-linear relationship where the exponent of any variable is not equal to 1 creates a curve.
- The general mathematical equation for a linear regression is –

$$y = ax + b$$

- The steps to create the relationship is:
  - Carry out the experiment of gathering a sample of observed values of height a corresponding weight.
  - Create a relationship model using the lm() functions in R.
  - Find the coefficients from the model created and create the mathemati equation using these
  - Get a summary of the relationship model to know the average error in predicti Also called residuals.
  - To predict the weight of new persons, use the predict() function in R.

Implementation Using R:

```
# The predictor vector.
x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)

# The resposne vector.
y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)

# Apply the lm() function.
relation <- lm(y~x)

# Find weight of a person with height 170.
a <- data.frame(x = 170)
result <-  predict(relation,a)
print(result)
```

Visualize the Regression Graphically

```
# Create the predictor and response variable.
x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
relation <- lm(y~x)

# Give the chart file a name.
png(file = "linearregression.png")

# Plot the chart.
plot(y,x,col = "blue",main = "Height & Weight Regression",
```

```
abline(lm(x~y)),cex = 1.3,pch = 16,xlab = "Weight in Kg",ylab = "Height in
cm")

# Save the file.
dev.off()
```

Applications:

(1) **Trend lines:** A trend line represents the variation in some quantitative data with passage of time (like GDP, oil prices, etc.). These trends usually follows a linear relationship. Hence, linear regression can be applied to predict future values. However, this method suffers from a lack of scientific validity in cases where other potential changes can affect the data.

(2) **Economics:** Linear regression is the predominant empirical tool in economics. For example, it is used to predict consumption spending, fixed investment spending, inventory investment, purchases of a country's exports, spending on imports, the demand to hold liquid assets, labor demand, and labor supply.

(3) **Finance:** Capital price asset model uses linear regression to analyze and quantify the systematic risks of an investment.

(4) **Biology:** Linear regression is used to model causal relationships between parameters in biological systems.

3.4.2 Nonlinear Regression

- In nonlinear regression, a statistical model of the form,

$$y \sim f(x, \beta)$$

relates a vector of independent variables, x, and their associated observed dependent variables, y.

- The function f is nonlinear in the components of the vector of parameters, but otherwise arbitrary. For example, the Michaelis–Menten model for enzyme kinetics has two parameters and one independent variable, related by f by:

$$f(x, \beta) = \frac{\beta_1 x}{\beta_2 + x}$$

- This function is nonlinear because it cannot be expressed as a linear combination of the two β.
- Systematic error may be present in the independent variables but its treatment is outside the scope of regression analysis. If the independent variables are not error-free, this is an errors-in-variables model, also outside this scope.

- Other examples of nonlinear functions include exponential functions, logarithmic functions, trigonometric functions, power functions, Gaussian function, and Lorenz curves. Some functions, such as the exponential or logarithmic functions, can be transformed so that they are linear. When so transformed, standard linear regression can be performed but must be applied with caution.
- In general, there is no closed-form expression for the best-fitting parameters, as there is in linear regression. Usually numerical optimization algorithms are applied to determine the best-fitting parameters. Again in contrast to linear regression, there may be many local minima of the function to be optimized and even the global minimum may produce a biased estimate.
- In practice, estimated values of the parameters are used, in conjunction with the optimization algorithm, to attempt to find the global minimum of a sum of squares.,

Implementation Using R:

```
#simulate some data
set.seed(20160227)
x<-seq(0,50,1)
y<-((runif(1,10,20)*x)/(runif(1,0,10)+x))+rnorm(51,0,1)

#for simple models nls find good starting values for the parameters even if
it throw a warning
m<-nls(y~a*x/(b+x))

#get some estimation of goodness of fit
cor(y,predict(m))

[1] 0.9496598

#plot
plot(x,y)
lines(x,predict(m),lty=2,col="red",lwd=3)
```

Assumptions:

- The data level in must be quantitative, the categorical variables must be coded as binary variables.
- The value of the coefficients can be correctly interpreted, only if the correct model has been fitted, therefore it is important to identify useful models.
- A good choice of starting points can lead to a desirable output, a poor choice will make the output misleading.

Summary

- R is a programming language and software environment for statistical analysis, graphics representation and reporting.
- R is freely available under the GNU General Public License, and pre-compiled binary versions are provided for various operating systems like Linux, Windows and Mac.
- R was created by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand, and is currently developed by the R Development Core Team.
- Machine learning must be one of the fastest growing fields in computer science. It is not only that the data is continuously getting "bigger" but also the theory to process it and turn it into knowledge.
- There are several ways to implement machine learning techniques, however the most commonly used ones are supervised and unsupervised learning.
- Supervised learning is all about operating to a known expectation and in this case, what needs to be analyzed from the data being defined. The input datasets in this context are also referred to as "labeled" datasets. It includes Support Vector Machines (SVMs), Naive Bayes classifiers etc.
- Unsupervised learning accept unlabeled data and attempt to group observations into categories based on underlying similarities in input features. Cluster analysis, k-means clustering etc. are all examples of unsupervised machine learning.
- Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters).
- K-Nearest Neighbors (KNN) is a supervised learning algorithm. The output depends on whether k-NN is used for classification (the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor) or regression (the output is the property value for the object. This value is the average of the values of its k nearest neighbors)
- Classification, also called categorization, is a machine learning technique that uses known data to determine how the new data should be classified into a set of existing labels/classes/categories.
- Naive Bayes Classifier is a simple technique for constructing classifiers. A Bayes classifier constructs models to classify problem instances. These classifications are made using the available data.
- Decision Tree is a type of supervised learning algorithm that is mostly used for classification problems. In Decision trees the data is continuously split according to a certain parameter. The structure of a decision tree composes of a root node which

describes the dataset, decision nodes, which perform the computation and leaf nodes which perform the classification. The leaves are the decisions or the final outcomes. And the decision nodes are where the data is split.

- A Support Vector Machine (SVM) is a supervised machine learning algorithm that can be employed for both classification and regression purposes. SVM vectors are classified by optimizing the line so that the closest point in each of the groups will be the farthest away from each other. Support vector machine algorithms are used to analyze data and find patterns.
- k-means clustering is an important clustering algorithm. The k in k-means clustering algorithm represents the number of clusters the data is to be divided into. For example, if the k value specified in the algorithm is 3, then algorithm will divide the data into 3 clusters.
- Association rule learning is one of the most well-renowned methods to uncover the association between variables in large databases. The rule focuses on looking for frequent co-occurring associations among a collection of random items. There are certain important concepts related to Associative Rule Learning such as Support, Confidence and Lift.
- Apriori is an unsupervised algorithm used for frequent item set mining. It generates associated rules from given data set and uses 'bottom-up' approach where frequently used subsets are extended one at a time and algorithm terminates when no further extension could be carried forward.
- Regression analysis is a conceptually simple method for investigating functional relationships among variables.
- In regression, the program predicts the value of a continuous output or response variable. In regression tasks, the program predicts the value of a continuous output or response variable from the input or explanatory variables.
- Linear regression is a linear approach to modelling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables).
- Nonlinear regression is a form of regression analysis in which observational data are modeled by a function which is a nonlinear combination of the model parameters and depends on one or more independent variables. The data are fitted by a method of successive approximations.

Check Your Understanding

1. ..... is about developing code to enable the machine to learn to perform tasks and its basic principle is the automatic modeling of underlying processes that have generated the collected data.

   (a) Data Science (b) Machine Learning (ML)

(d) All of the Mentioned

(c) Data Analytics

2. Groups of related observations are called ..... and the procedure to organize items of a given collection into groups based on some similar features called as..... .
   (a) clusters, analysis (b) regression, clustering
   (c) clusters, clustering (d) None of the Mentioned
3. ..... rule mining is a technique to identify underlying relations between different items.
   (a) Association (b) Analytics
   (c) Learning (d) All of the Mentioned
4. ..... is a form of supervised learning. Mail service providers like Gmail, Yahoo and others use this technique to classify a new mail as spam or not spam.
   (a) Machine Learning (b) Classification
   (c) Regression (d) All of the Mentioned
5. Examples of supervised learning includes ..... .
   (a) Regression (b) Decision Tree
   (c) KNN (d) All of the Mentioned
6. A Naive Bayes Classifier is a ..... machine learning algorithm which relies on the assumption of feature independent to classify input data.
   (a) supervised (b) unsupervised
   (c) semi-supervised (d) All of the Mentioned
7. ..... analysis has become one of the most widely used statistical tools for analyzing multifactor data and it is appealing because it provides a conceptually simple method for investigating functional relationships among variables.
   (a) Clustering (b) Regression
   (c) Data (d) All of the Mentioned
8. A ..... is an example of the most widely used machine learning algorithms much of its popularity is because it can be adapted to almost any type of data.
   (a) Clustering (b) Regression
   (c) Decision Tree (d) All of the Mentioned
9. Unsupervised learning makes sense of ..... data without having any predefined dataset for its training.
   (a) unlabeled (b) labeled
   (c) semi-abeled (d) All of the Mentioned
10. Support Vector Machines (SVMs) are well-known ..... classification algorithms that separate different categories of data.
    (a) semi-supervised (b) unsupervised

   (c) supervised (d) All of the Mentioned
11. ..... are some of the examples of unsupervised learning.
    (a) Apriori algorithm (b) k-means
    (c) Cluster analysis (d) All of the Mentioned
12. ..... is an algorithm for frequent item set mining and association rule learning over transactional databases and it proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database.
    (a) Apriori (b) k-means
    (c) KNN (d) All of the Mentioned
13. The ..... algorithm is the simplest machine learning algorithm, which building the model consists only of storing the training dataset. To, make a prediction for a new data point, the algorithm finds the closest data points in the training dataset-its nearest neighbors.
    (a) Apriori (b) k-means
    (c) KNN (d) All of the Mentioned
14. Association rule mining discovers strong ..... relationships among data.
    (a) association (b) correlation
    (c) Both (a) and (b) (d) None of the Mentioned

ANSWER KEY

1. (b)	2. (c)	3. (a)	4. (b)	5. (d)
6. (a)	7. (b)	8. (c)	9. (a)	10. (c)
11. (d)	12. (a)	13. (c)	14. (c)	

Practice Questions

Q.I: Answer the following Questions in short.

1. What is Machine Learning?
2. What supervised machine leaning?
3. What is unsupervised machine leaning? Enlist its examples.
4. Write short note on: Naïve Baye's.
5. What is Regression analysis? Define it

Q.II: Answer the following Questions.

1. With the help of example describe K-means algorithm.
2. Describe KNN in detail.

3. What is non-linear regression? Explain with example.
4. Write short note on: Linear regression.
5. Compare supervised and unsupervised machine leaning.
6. Describe association rules for mining.
7. Write short note on: Apriori algorithm.
8. What is decision tree? Explain with example.
9. State advantages and disadvantages of Naïve Bayes.
10. State advantages and disadvantages of SVM.
11. Explain Naïve Bayes with the help of example.
12. Give advantages and disadvantages of machine learning.
13. State advantages and disadvantages EM Algorithm.

Q.III: Define the following terms.

1. Objects in R
2. Root Node
3. Pruning
4. Vector
5. Support
6. Confidence
7. lift
8. Decision Node
9. cluster analysis
10. SVM

❖❖❖

4... Data Analytics with R/WEKA Machine Learning

Objectives...

- To understand the Data Manipulation
- To understand the Data Visualization
- To know the concept of Data Analysis

4.1 INTRODUCTION

- Data Analytics is the science of examining raw data to conclude that information.
- Data Analytics involves applying an algorithmic or mechanical process to derive insights. For example, running through several data sets to look for meaningful correlations between each other.

4.1.1 R Language

- We will use R for our data analysis so we need to know the basics of programming in the R language. R is a full programming language with both functional programming and object-oriented programming features.
- The typical data analysis workflow looks like this: collect your data and put it in a file or spreadsheet or database. Then you run some analyses, written in various scripts, perhaps saving some intermediate results along the way or maybe always working on the raw data.
- R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes:
  - An effective data handling and storage facility.
  - A suite of operators for calculations on arrays, in particular matrices.

- A large, coherent, integrated collection of intermediate tools for data analysis.
- Graphical facilities for data analysis and display either on-screen or on hardcopy.
- A well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

4.1.2 Basic Interaction with RStudio

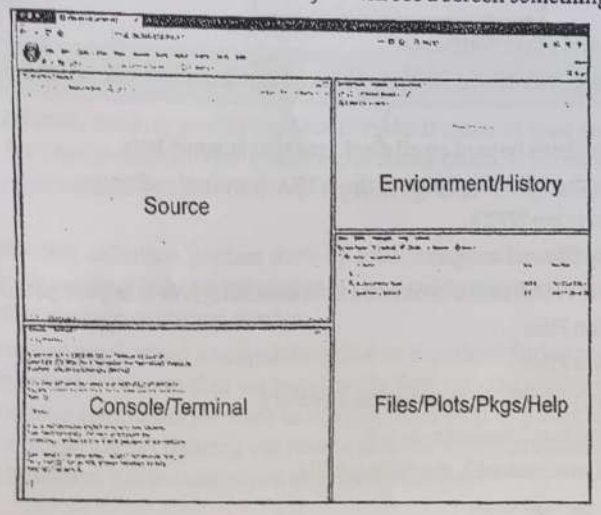
- RStudio is an **Integrated Development Environment or IDE** for the R programming language. It is an **open source** software developed by RStudio Inc. It is written in Java, C++, and JavaScript.
- RStudio includes a console, syntax-highlighting editor that supports direct code execution, and a variety of robust tools for plotting, viewing history, debugging and managing your workspace.
- After installation of R , you need to download and install RStudio

To Install RStudio

- Go to www.rstudio.com and click on the "Download RStudio" button.
- Click on "Download RStudio Desktop."
- Click on the version recommended for your system, or the latest Windows version, and save the executable file. Run the .exe file and follow the installation instructions.

4.1.3 Screen Layout of RStudio

- When you open RStudio for the first time you will see a screen something like this:



4.1: RStudio Screen Layout

RStudio Interface:

- The RStudio interface has four main panels:

(1) **Source:** This panel is where you will write/view R scripts. This is also called as Script Editor. R commands are typed into this panel and submitted to the R Console panel. Some outputs (such as if you view a dataset using View()) will appear as a tab here.

- The RStudio script editor allows you to 'send' the current line or the currently highlighted text to the R console by clicking on the Run button in the upper-right hand corner of the script editor. Alternatively, you can run by simply pressing the Ctrl and Enter keys at the same time as a shortcut.

(2) **Console/Terminal:** Once you have opened RStudio, you can type R expressions into the console, which is the frame on the left of the RStudio window. This is actually where you see the execution of commands. You can work interactively (i.e. enter R' commands here), but for the most part we will run a script (or lines in a script) in the source panel and watch their execution and output here.

- The "Terminal" tab give you access to the BASH terminal (the Linux operating system, unrelated to R).

Console Command Prompt:

- R is a command line driven program. The user enters commands at the prompt (> by default) and each command is executed one at a time.
- When the console receives a command (by directly typing into the console or running from the script editor (Ctrl-Enter), R will try to execute it.
- After running, the console will show the results and come back with a new → prompt to wait for new commands.

(3) **Environment/History:** Here, RStudio will show you what datasets and objects (variables) you have created and which are defined in memory. You can also see some properties of objects/datasets such as their type and dimensions.

- The "History" tab contains a history of the R commands you've executed R.

(4) **Files/Plots/Packages/Help:** This multipurpose panel will show you the contents of directories on your computer. You can also use the "Files" tab to navigate and set the working directory.

- The "Plots" tab will show the output of any plots generated.
- In "Packages" you will see what packages are actively loaded, or you can attach installed packages.
- "Help" will display help files for R functions and packages.

Create New project:

- You always want to be working on a project. Projects keep track of the state of your analysis by remembering variables and functions you have written and keep track of which files you have opened and such.

- Choose File → New Project to create a project. You can create a project from an existing directory, but if this is the first time you are working with R you probably just want to create an empty project in a new directory, so do that.

4.1.4 Creating an R script

- You can open a new empty script by clicking the New File icon in the upper left of the main RStudio toolbar (File → New file). You can select several different file types. We are interested in the *R Script* and *R Markdown* types. The former type is the file type for pure R code, while the latter type is used for creating reports where documentation text is mixed with R code.

4.1.5 Interaction with WEKA

- WEKA is an open source software provides tools for data preprocessing, implementation of several Machine Learning algorithms, and visualization tools so that you can develop machine learning techniques and apply them to real-world data mining problems.

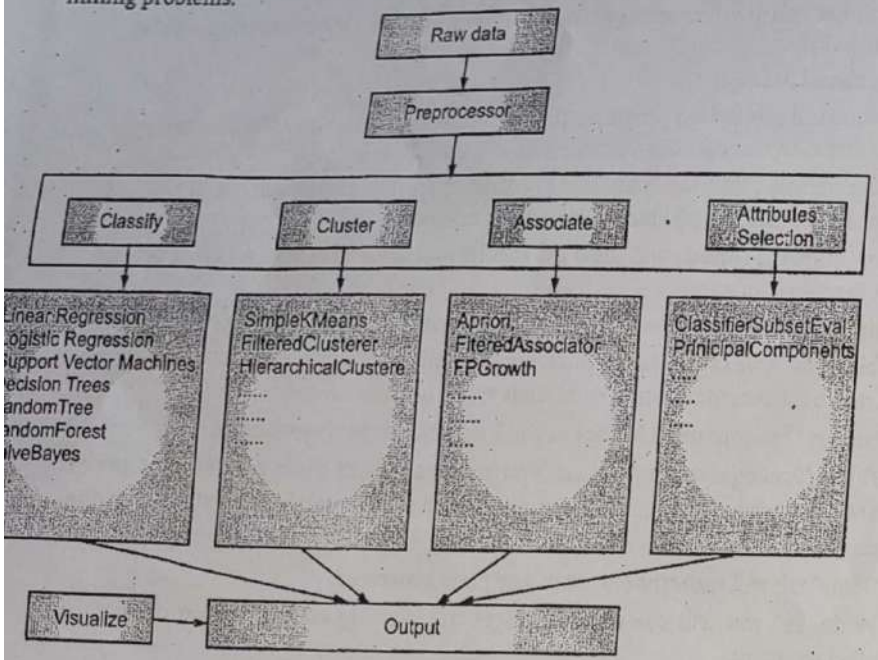


Fig. 4.2: Elements of WEKA

is WEKA?

EKA is a collection of machine learning algorithms for data mining tasks. The orithms can either be applied directly to a dataset or called from your own Java

code. WEKA contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.

Installation on Windows:

- On windows the all-in-one version of WEKA is provided as a self-extracting executable.
- You must choose whether you would like the 32-bit version of the package or the 64-bit version of the package. If you have a modern version of Windows, you should select the 64-bit version.
- On the Weka download webpage, these packages are called:
  - Self-extracting executable for 64-bit Windows that includes Oracle's 64-bit Java.
  - Self-extracting executable for 32-bit Windows that includes Oracle's 32-bit Java.
- The download is about 100 megabytes. After you have downloaded the package, double click on the icon to start the installation process.
- Follow the prompts for the installation and WEKA will be added to your Program Menu.
- Start WEKA by clicking on the bird icon.

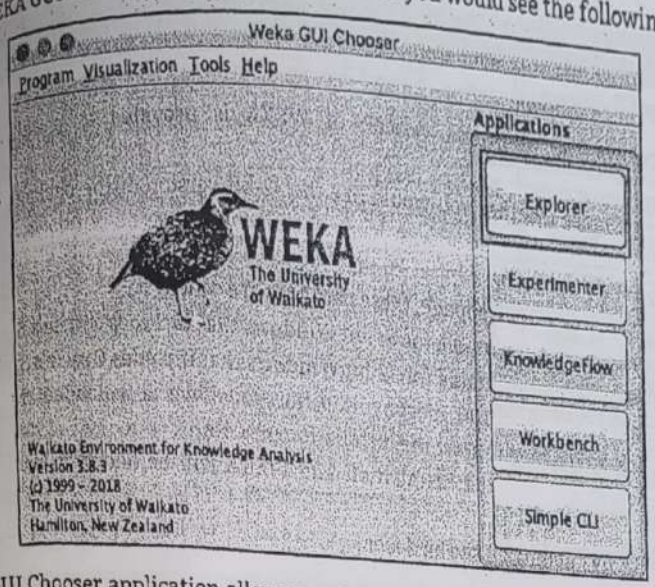
Install WEKA on Linux and other Platforms:

- WEKA also provides a standalone version that you can install on Linux and other platforms.
- WEKA runs on Java and can be used on all platforms that support Java.
- It is a zip file and has the following name of the WEKA download webpage:
  - Zip archive containing WEKA.
  - Download the zip file and unzip it.
  - You can also start WEKA on the command line, assuming Java is in your path.
  - WEKA Installation Files
  - WEKA Installation Files

(1) Change directory into your WEKA installation directory.
For example: `cd /Applications/weka-3-8-0`

(2) Start the Java virtual machine with the weka.jar file,
For example: `java -jar weka.jar`

- The WEKA GUI Chooser application will start and you would see the following screen:



- The GUI Chooser application allows you to run five different types of applications as listed here –
  - Explorer
  - Experimenter
  - KnowledgeFlow
  - Workbench
  - Simple CLI

4.2 DATA MANIPULATION

- Data manipulation involves modifying data to make it easier to read and to be more organized. We manipulate data for analysis and visualization. It is also used with the term 'data exploration' which involves organizing data using available sets of variables.
- At times, the data collection process done by machines involves a lot of errors and inaccuracies in reading. Data manipulation is also used to remove these inaccuracies and make data more accurate and precise.
- Data science is as much about manipulating data as it is about fitting models to data. Data rarely arrives in a form that we can directly feed into the statistical models or machine learning algorithms we want to analyze them with. The first stages of data analysis are almost always figuring out how to load the data into R and then figuring out how to transform it into a shape you can readily analyze.

Built-in datasets in R:

- In R, some datasets already built or available in R packages. Those are useful for learning how to use new methods. If you already know a dataset and what it can tell you, it is easier to evaluate how a new method performs. It is also useful for benchmarking methods you implement. They are of course less helpful when it comes to analyzing new data.
- Distributed together with R is the package dataset. You can load the package into R using the `library()` function and get a list of the datasets in it, together with a short description of each, like this:
  - `library(datasets)`
  - `library(help = "datasets")`

  For example:
- We will use the default iris table in R, as follows:

```
#To load datasets package
    library("datasets")
#To load iris dataset
    data(iris)
    summary(iris)
```

Output:

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
Min. :4.300	Min. :2.000	Min. :1.000	Min. :0.100	setosa: 50
1st Qu.:5.100	1st Qu.:2.800	1st Qu.:1.600	versicolor:0.300	versicolor:50
Median: 5.800	Median: 3.000	Median: 4.350	Median: 1.300	Virginica: 50
Mean: 5.843	Mean: 3.057	Mean: 3.758	Mean: 1.199	
3rd Qu.:6.400	3rd Qu.:3.300	3rd Qu.:5.100	3rd Qu.:1.800	
Max. :7.900	Max. :4.400	Max. :6.900	Max. :2.500	
Species setosa :50 versicolor:50 virginica :50				

4.2.1 Data Manipulation in R With 'dplyr' Package

- To manipulate data different ways to perform in R, such as using Base R functions like subset(), with(), within(), etc.,
- Packages like data.table, ggplot2, reshape2, readr etc., and different Machine Learning algorithms.

dplyr package:

- In this chapter, we are going to use the **dplyr** package to perform data manipulation in R.

  When working with data you must:

  - Figure out what you want to do.
  - Describe those tasks in the form of a computer program.
  - Execute the program.
- The **dplyr** package makes these steps fast and easy:
  - By constraining your options, it helps you think about your data manipulation challenges.
  - It provides simple "verbs", functions that correspond to the most common data manipulation tasks, to help you translate your thoughts into code.
  - It uses efficient backends, so you spend less time waiting for the computer.
- The **dplyr** package consists of many functions specifically used for data manipulation. These functions process data faster than Base R functions and are known the best for data exploration and transformation, as well.

Install and load dplyr package:

- To install the dplyr package, type the following command.

```
install.packages("dplyr")
```

- To load dplyr package, type the command below:

```
library(dplyr)
```

- In this below example, we are going to use the *iris* dataset from the datasets package in R programming that can be loaded as follows:

```
#To load dplyr package
    library("dplyr")
#To load datasets package
    library("datasets")
#To load iris dataset
    data(iris)
    summary(iris)
```


Output:

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
Min. :4.300	Min. :2.000	Min. :1.000	Min. :0.100	setosa: 50
1st Qu.:5.100	1st Qu.:2.800	1st Qu.:1.600	versicolor:0.300	versicolor:50
Median: 5.800	Median: 3.000	Median: 4.350	Median: 1.300	virginica: 50
Mean: 5.843	Mean: 3.057	Mean: 3.758	Mean: 1.199	
3rd Qu.:6.400	3rd Qu.:3.300	3rd Qu.:5.100	3rd Qu.:1.800	
Max. :7.900	Max. :4.400	Max. :6.900	Max. :2.500	

- It contains 150 samples of three plant species (setosa, virginica, and versicolor) and four features measured for each sample.

4.2.1.1 Functions included in "dplyr" package

- Following are some of the important functions included in the *dplyr* package:

1. **Head ( ):** This function returns the first n rows of a matrix or data frame

   Syntax:

```
head(df)
head(df,n=number)
```

   where, `df` – Data frame and `n` – number of rows

2. **Tail ( ):** This function returns the last n rows of a matrix or data frame

   Syntax:

```
tail(df)
tail(df,n=number)
```

   where, `df` – Data frame and `n` – number of rows

3. **Select( ):** It is used to select data by its column name. We can select any number of columns in a number of ways.

   For example: Do spacing as shown in box

```
#To select the following columns
    selected <- select(iris, Sepal.Length, Sepal.Width, Petal.Length)
    head(selected)
#To select all columns from Sepal.Length to Petal.Length
    selected1 <- select(iris, Sepal.Length:Petal.Length)
#To print first four rows
    head(selected1, 4)
#To select columns with numeric indexes
    selected1 <- select(iris,c(3:5))
    head(selected1)
```

Output:

Sr. No.	Sepal.Length	Sepal.Width	Petal.Length
1.	5.1	3.5	1.4
2.	4.9	3.0	1.4
3.	4.7	3.2	1.3
4.	4.6	3.1	1.5
5.	5.0	3.6	1.4
6.	5.4	3.9	1.7

Output:

Sr. No.	Sepal.Length	Sepal.Width	Petal.Length
1.	5.1	3.5	1.4
2.	4.9	3.0	1.4
3.	4.7	3.2	1.3
4.	4.6	3.1	1.5

Output:

Sr. No.	Petal.Length	Petal.Width	Species
1.	1.4	0.2	Setosa
2.	1.4	0.2	Setosa
3.	1.3	0.2	Setosa
4.	1.5	0.2	Setosa
5.	1.4	0.2	Setosa
6.	1.7	0.4	Setosa

```
#We use(-)to hide a particular column
selected <- select(iris, -Sepal.Length, -Sepal.Width)
head(selected)
```

Output:

Sr. No.	Petal.Length	Petal.Width	Species
1.	1.4	0.2	Setosa
2.	1.4	0.2	Setosa
3.	1.3	0.2	Setosa
4.	1.5	0.2	Setosa
5.	1.4	0.2	Setosa
6.	1.7	0.4	Setosa

4. Filter()

- It is used to find rows with matching criteria. It also works like the select( ) function, i.e., we pass a data frame along with a condition separated by a comma. For example:

```
#To select the first 3 rows with Species as setosa
filtered <- filter(iris, Species = = "setosa" )
head(filtered,3)
```

Output:

Sr. No.	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1.	5.1	3.5	1.4	0.2	Setosa
2.	4.9	3.0	1.4	0.2	Setosa
3.	4.7	3.2	1.3	0.2	Setosa

```
#To select the last 5 rows with Species as versicolor and Sepal width more
than 3
filtered1 <- filter(iris, Species == "versicolor", Sepal.Width > 3)
tail(filtered1)
```

Output:

Sr. No.	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
4.	6.3	3.3	4.7	1.6	Versicolor
5.	6.7	3.1	4.4	1.4	Versicolor
6.	5.9	3.2	4.8	1.8	Versicolor
7.	6.0	3.4	4.5	1.6	Versicolor
8.	6.7	3.1	4.7	1.5	Versicolor

5. Mutate()

- mutate( ) is used to create new columns and preserve the existing columns in a dataset.
- It is useful to create attributes that are functions of other attributes in the dataset.

For example:

```
#To create a column "Greater.Half" which stores TRUE if given condition
    is TRUE
col1 <- mutate(iris, Greater.Half = Sepal.Width > 0.5 * Sepal.Length)
    tail(col1)
```

Output:

S.N.	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species	Greater.Half
145.	6.7	3.3	5.7	2.5	Virginica	FALSE
146.	6.7	3.0	5.2	2.3	Virginica	FALSE
147.	6.3	2.5	5.0	1.9	Virginica	FALSE
148.	6.5	3.0	5.2	2.0	Virginica	FALSE
149.	6.2	3.4	5.4	2.3	Virginica	TRUE
150.	5.9	3.0	5.1	1.8	Virginica	TRUE

```
#To check how many flowers satisfy this condition
table(col1$Greater.Half)
```

- Here, Table function performs categorical tabulation of data with the variable and its frequency. Table( ) function is also helpful in creating Frequency tables with condition and cross tabulations.

Output:

FALSE=84 TRUE=66

6. Arrange()

- It is used to sort rows by variables in both an ascending and descending order.

For example:

```
#To arrange Sepal Width in ascending order
    arranged <- arrange(col1, Sepal.Width)
    head(arranged)
#To arrange Sepal Width in descending order
    arranged <- arrange(col1, desc(Sepal.Width))
    head(arranged)
```

Output:

S.N.	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species	Greater.Half
1.	5.0	2.0	3.5	1.0	Versicolor	FALSE
2.	6.0	2.2	4.0	1.0	Versicolor	FALSE
3.	6.2	2.2	4.5	1.5	Versicolor	FALSE
4.	6.0	2.2	5.0	1.5	Virginica	FALSE
5.	4.5	2.3	1.3	0.3	Setosa	TRUE
6.	5.5	2.3	4.0	1.3	Versicolor	FALSE

S.N.	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species	Greater.Half
1.	5.7	4.4	1.5	0.4	Setosa	TRUE
2.	5.5	4.2	1.4	0.2	Setosa	TRUE
3.	5.2	4.1	1.5	0.1	Setosa	TRUE
4.	5.8	4.0	1.2	0.2	Setosa	TRUE
5.	5.4	3.9	1.7	0.4	Setosa	TRUE
6.	5.4	3.9	1.3	0.4	Setosa	TRUE

7. Summarise()

- It is used to find insights (mean, median, mode, etc.) from a dataset.
- It is used to aggregate multiple values to a single value.
- It is most often used with the group_by function, and the output has one row per group.

For example:

```
summarised <- summarise(arranged, Mean.Width = mean(Sepal.Width))
head(summarised)
```

Output:

Mean.Width

1 3.057333

8. Group_by()

- group_by function is used to group observations within a dataset by one or more variables. Most data operations are performed on groups defined by variables.

For example:

```
#To find mean sepal width by Species, we use grouping as follows
gp <- group_by(iris,Species)
mn <- summarise(gp,Mean.Sepal = mean(Sepal.Width))
head(mn)
```

Output:

Sr. No.	Species <fct>	Mean.Length <dbl>
1.	Setosa	5.01
2.	Versicolor	5.94
3.	Virginica	6.59

9. Join()

- It is used two join two data frames.
- Currently **dplyr** supports four types of mutating joins, two types of filtering joins, and a nesting join.
- **Mutating joins** combine variables from the two data frames x and y:
  - **inner_join( ):** return all rows from x where there are matching values in y, and all columns from x and y. If there are multiple matches between x and y, all combination of the matches are returned.
  - **left_join( ):** return all rows from x, and all columns from x and y. Rows in x with no match in y will have NA values in the new columns. If there are multiple matches between x and y, all combinations of the matches are returned.
  - **right_join( ):** return all rows from y, and all columns from x and y. Rows in y with no match in x will have NA values in the new columns. If there are multiple matches between x and y, all combinations of the matches are returned.
  - **full_join( ):** return all rows and all columns from both x and y. Where there are not matching values, returns NA for the one missing.
- **Filtering joins** keep cases from the left-hand data.frame:
  - **semi_join( ):** return all rows from x where there are matching values in y, keeping just columns from x. A semi join differs from an inner join because an inner join will return one row of x for each matching row of y, where a semi join will never duplicate rows of x.
  - **anti_join( ):** return all rows from x where there are not matching values in y, keeping just columns from x.
- **Nesting joins** create a list column of dataframes:
- **nest_join( ):** return all rows and all columns from x. Adds a list column of tibbles. Each tibble contains all the rows from y that match that row of x. When there is no match, the list column is a 0-row tibble with the same column names and types as y.

Syntax:

```
inner_join(x, y, by = )
left_join(x, y, by = )
right_join(x, y, by = )
full_join(x, y, by = )
semi_join(x, y, by = )
anti_join(x, y, by = )
```

where,

x, y - datasets (or tables) to merge / join

by - common variable (primary key) to join by.

4.2.1.2 Pipe Operator

- Pipe operator lets us wrap multiple functions together. It is denoted as %>% .
- This makes it easy when we need to perform various operations on a dataset to derive the results.
- It can be used with functions like filter( ), select( ), arrange( ), summarise( ), group_by ( ), etc.

For example:

```
#To get rows with the following conditions
iris %>% filter(Species == "setosa",Sepal.Width > 3.8)
```

Output:

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.4	3.9	1.7	0.4	Setosa
5.8	4.0	1.2	0.2	Setosa
5.7	4.4	1.5	0.4	Setosa
5.4	3.9	1.3	0.4	Setosa
5.2	4.1	1.5	0.1	Setosa
5.5	4.2	1.4	0.2	Setosa

```
#To find mean Sepal Length by Species, we use pipe operator as follows
iris %>% group_by(Species) %>% summarise(Mean.Length = mean(Sepal.Length))
```

Output:

Species <fct>	Mean.Length <dbl>
Setosa	5.01
Versicolor	5.94
Virginica	6.59

4.3 DATA VISUALIZATION

- Data visualization is a technique used for the graphical representation of data. By using elements like scatter plots, charts, graphs, histograms, maps, etc., we make our data more understandable. Data visualization makes it easy to recognize patterns, trends, and exceptions in our data. It enables us to convey information and results in a quick and visual way.
- It is easier for a human brain to understand and retain information when it is represented in a pictorial form. Therefore, Data Visualization helps us interpret data quickly, examine different variables to see their effects on the patterns, and derive insights from our data.

- R programming provides comprehensive sets of tools such as in-built functions and a wide range of packages to perform data analysis, represent data and build visualizations.

4.3.1 Data Visualization in R

- Data visualization in R can be performed in the following ways:
  - **Base Graphics:** It is the graphics system that was originally developed for R.
  - **Grid Graphics:** It is an alternative graphics system that was later added to R. The big difference between grid and the original base graphics system is that grid allows for the creation of multiple regions, called viewports, on a single graphics page.
  - **Lattice Graphics:** The lattice add-on package is an implementation of Trellis graphics for R. It was originally developed for the languages S and S-Plus at Bell Labs. Lattice graphics in R make use of grid graphics.
  - **ggplot2:** ggplot 2 is an enhanced data visualization package for *R*.

4.3.2 Base R Graphics

- R provides some built-in functions which are included in the **graphics** package for data visualization in R.
- In this section, we are going to use the default **mtcars** dataset for data visualization in R.

```
#To load graphics package
    library("graphics")
#To load datasets package
    library("datasets")
#To load mtcars dataset
    data(mtcars)
#To analyze the structure of the dataset
    str(mtcars)
```

Output:

```
'data.frame': 32 obs. of 11 variables:
$ mpg : num 21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
$ cyl : num 6 6 4 6 8 6 8 4 4 6 ...
$ disp: num 160 160 108 258 360 ...
$ hp  : num 110 110 93 110 175 105 245 62 95 123 ...
$ drat: num 3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
```



```
$ wt  : num 2.62 2.88 2.32 3.21 3.44 ...
$ qsec: num 16.5 17 18.6 19.4 17 ...
$ vs  : num 0 0 1 1 0 1 0 1 1 1 ...
$ am  : num 1 1 1 0 0 0 0 0 0 0 ...
$ gear: num 4 4 4 3 3 3 3 4 4 4 ...
$ carb: num 4 4 1 1 2 1 4 2 2 4 ...
```

- It contains data about the design, performance and fuel economy of 32 automobiles from 1973 to 1974, extracted from the 1974 Motor Trend US magazine.

The plot( ) Function:

- The plot( ) function is used to plot R objects.
- The basic syntax for the **plot( )** function is given below:

```
plot(x,y,type,main,sub,xlab,ylab,asp,col,..)
```

where,

x : The x coordinate of the plot, a single plotting structure, a function, or an R object

y : The Y coordinate points in the plot (optional if x coordinate is a single structure)

type : 'p' for points, 'l' for lines, 'b' for both, 'h' for high-density vertical lines, etc.

main : Title of the plot

sub : Subtitle of the plot

xlab: Title for the x-axis

ylab: Title for the y-axis

asp : Aspect ratio(y/x)

col : Color of the plot(points, lines, etc.)

For example:

```
#To plot mpg(Miles per Gallon) vs Number of cars
plot(mtcars$mpg, xlab = "Number of cars", ylab = "Miles per Gallon",
col = "red")
```

Output:

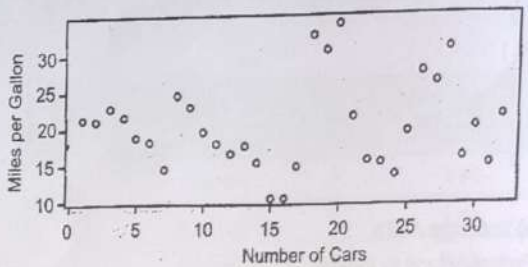


Fig. 4.3: Use of plot( ) for Scatter Plot

- Here, we get a scatter/dot plot wherein we can observe that there are only six cars with miles per gallon (mpg) more than 25.

```
#To find relation between hp (Horse Power) and mpg (Miles per Gallon)
plot(mtcars$hp, mtcars$mpg, xlab = "HorsePower", ylab = "Miles per
Gallon", type = "h", col = "blue")
```

Output:

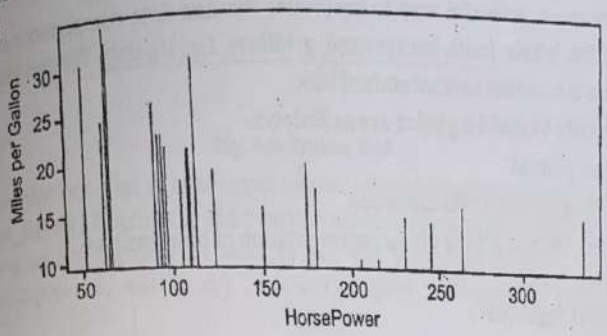


Fig. 4.4: Use of plot( ) for Line Chart

(a) Barplot

- It is used to represent data in the form of rectangular bars, both in vertical and horizontal ways, and the length of the bar is proportional to the value of the variable.
- Bar plots can be created in R using the barplot() function. We can supply a vector or matrix to this function. If we supply a vector, the plot will have bars with their heights equal to the elements in the vector.

For example:

```
#To draw a barplot of hp
#Horizontal
barplot(mtcars$hp,xlab = "HorsePower", col = "cyan", horiz = TRUE)
#Vertical
barplot(mtcars$hp, ylab = "HorsePower", col = "cyan", horiz = FALSE)
```

Output:

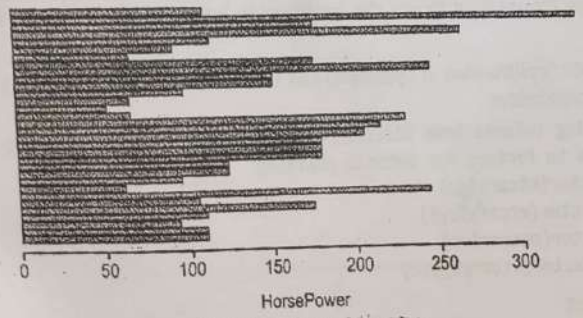


Fig. 4.5: Horizontal Bar Plot

Output:

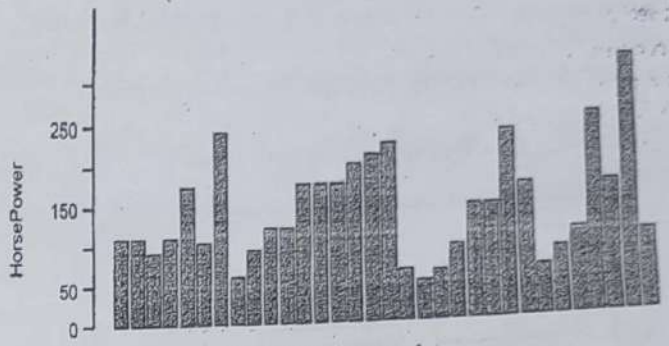


Fig. 4.6: Vertical Bar Plot

(b) Histogram

- It is used to divide values into groups of continuous ranges measured against the frequency range of the variable.
- Histogram can be created using the hist( ) function in R programming language. This function takes in a vector of values for which the histogram is plotted.

For example:

```
#To find histogram for mpg (Miles per Gallon)
hist(mtcars$mpg, xlab = "Miles Per Gallon", main = "Histogram for MPG", col
= "yellow")
```

Output:

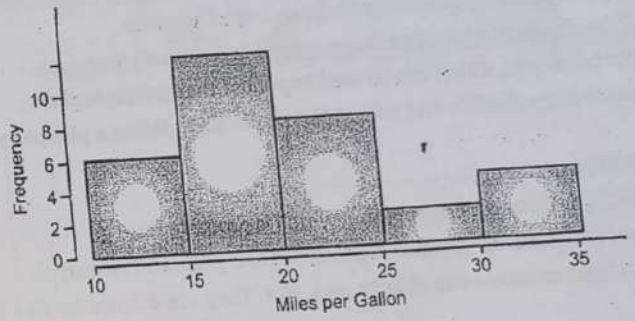


Fig. 4.7: Histogram

(c) Boxplot

- It is used to represent descriptive statistics of each variable in a dataset. It shows five statistically significant numbers - the minimum, first quartile(the 25th percentile), median, third quartile(the 75th percentile), and the maximum values of a variable.
- boxplot (and whisker plot) is created using the **boxplot( )** function.

- The boxplot() function takes in any number of numeric vectors, drawing a boxplot for each vector.
- You can also pass in a list (or data frame) with numeric vectors as its components.

For Example,

```
#To draw boxplots for disp (Displacement) and hp (Horse Power)
boxplot(mtcars[,3:4])
```

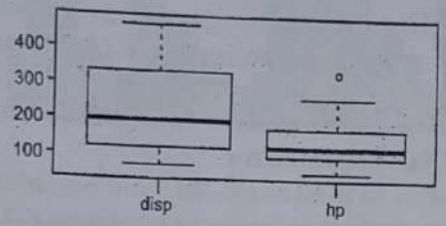


Fig. 4.8: Box Plots

4.3.3 Data Visualization in R with ggplot2 package

- ggplot2 is one of the most sophisticated packages in R for data visualization, and it helps create the most elegant and versatile print-quality plots with minimal adjustments. It is very simple to create single and multivariable graphs with the help of the ggplot2 package.
- The ggplot2 package in R is based on the **Grammar of Graphics**, which is a set of rules for describing and building graphs. By breaking up graphs into semantic components such as scales and layers, ggplot2 implements the grammar of graphics.
- There are two major functions in ggplot2 package: qplot() and ggplot() functions.
  - (i) **qplot()** stands for quick plot, which can be used to produce easily simple plots.
  - (ii) **ggplot()** function is more flexible and robust than qplot for building a plot piece by piece.

Elements of ggplot2 package:

- The ggplot2 grammar of graphics is composed of the following elements:
  - **Data:** Data is the most crucial thing which is processed and generates an output.
  - **Layers:** Layers are used to create the objects on a plot. They are defined by five basic parts:

    Data, Mapping, Statistical transformation(stat),Geometric object(geom), Position adjustment (position)
  - **Scales:** It is used to map the data values into values present in the coordinate system of the graphics device.
  - **Coordinates:** A coordinate system (coord) maps the position of objects onto the plane of the plot, and controls how the axes and grid lines are drawn. Plots

typically use two coordinates ( ), but could use any number of coordinates. Most plots are drawn using the Cartesian coordinate system.

  - **Faceting:** Faceting is used to split the data into subgroups and draw sub-graphs for each group.
  - **Themes:** Themes are a powerful way to customize the non-data components of your plots: i.e. titles, labels, fonts, background, gridlines, and legends. Themes can be used to give plots a consistent customized look.

The three basic components to build a ggplot are as follows:

1. **Data:** Dataset to be plotted
2. **Aesthetics:** Mapping of data to visualization.
3. **Geometry/Layers:** We use it for a visual representation of observations.

To load ggplot2 package:

```
install.packages("ggplot2")
library(ggplot2)
```

- The basic syntax for **ggplot** is given below:

```
ggplot(data = NULL, mapping = aes()) + geom_function()
#To Install and load the ggplot2 package
```

Plotting with ggplot2:

- In this section, we are going to use the **mtcars** dataset from the datasets package in R that can be loaded as follows:

```
#To load datasets package
library("datasets")
#To load iris dataset
data(mtcars)
#To analyze the structure of the dataset
str(mtcars)
```

(1) Scatter Plots

- **When to use:** Scatter Plot is used to see the relationship between two continuous variables.
- To draw a scatter plot of cyl(Number of Cylinders) and vs(Engine Type(0 = V-shaped, 1 = straight)), run the code below:

```
# Since the following columns have discrete (categorical) set of values, So
we can convert them to factors for optimal plotting
mtcars$am <- as.factor(mtcars$am)
mtcars$cyl <- as.factor(mtcars$cyl)
mtcars$vs <- as.factor(mtcars$vs)
mtcars$gear <- as.factor(mtcars$gear)

#To draw scatter plot
ggplot(mtcars, aes(x= cyl , y= vs)) + geom_point()
```

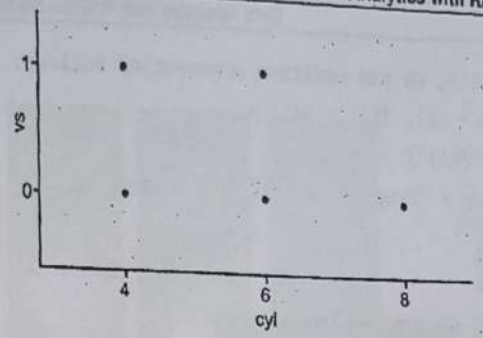


Fig. 4.9: Scatter Plot

- Since this plot has a lot of overlapped values, which is known as *overplotting*, we will use **geom_jitter( )** function to add a certain amount of noise to avoid it.

```
#Here width argument is used to set the amount of jitter
ggplot(mtcars, aes(x= cyl , y= vs)) + geom_jitter(width = 0.1)
```

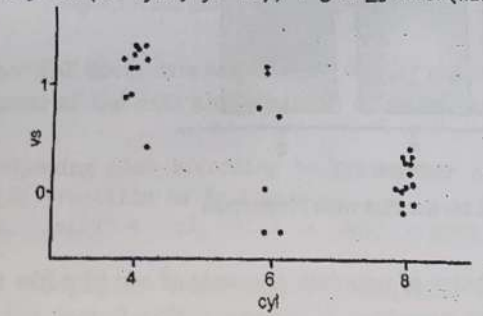


Fig. 4.10: Overplotting in Scatter plot

- Here, we can also use the argument alpha to set the transparency of the points to further reduce overplotting for data visualization in R.

```
#Transparency set to 50%
ggplot(mtcars, aes(x= cyl, y= vs)) + geom_jitter(width = 0.1, alpha= 0.5)
```

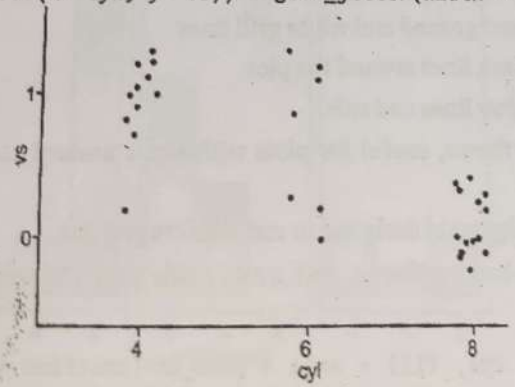


Fig. 4.11: Set transparency for Scatter plot

- With ggplot2, we can plot multivariate plots effectively.
- For example:

To draw a scatter plot of cyl(Number of Cylinders) and vs(Engine Type(0 = V-shaped, 1 = straight)) according to am Transmission (0 = automatic, 1 = manual), run the following code:

```
#We use the color aesthetic to introduce third variable with a legend on
the right side
ggplot(mtcars, aes(x= cyl,y= vs,color = am)) + geom_jitter(width = 0.1,
alpha = 0.5)
```

Output:

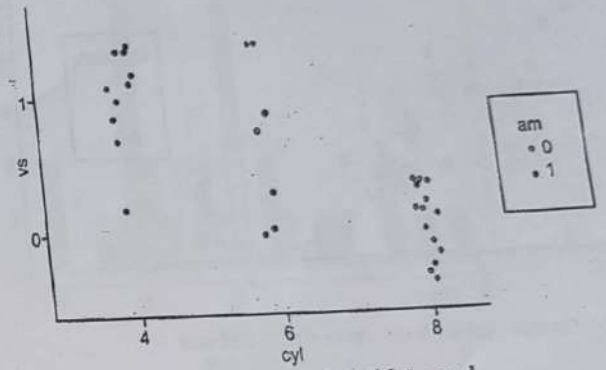


Fig. 4.12: Scatter plot with Legend

```
#To add the labels
ggplot(mtcars, aes(x= cyl , y= vs , color = am)) +
geom_jitter(width = 0.1, alpha = 0.5) +
labs(x = "Cylinders",y = "Engine Type", color = "Transmission(0
automatic, 1 = manual)")
```

Output:

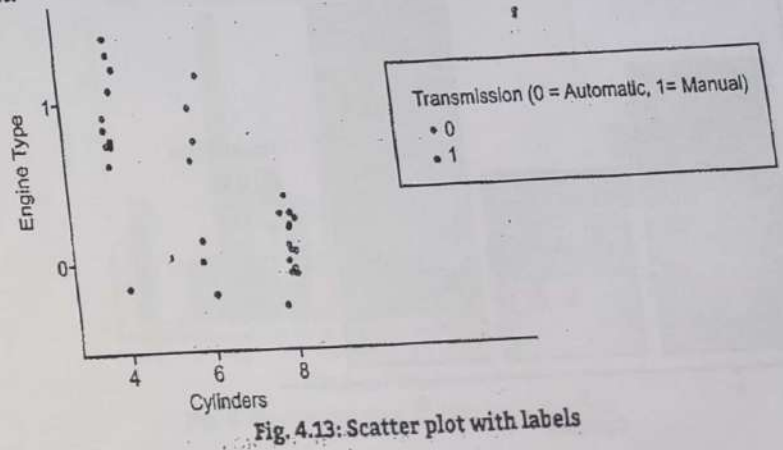


Fig. 4.13: Scatter plot with labels

```
#To plot with shape =1 and size = 4
ggplot(mtcars, aes(x = wt, y = mpg, col = cyl)) + geom_point(size = 4,
shape = 1, alpha = 0.6) +
labs(x = "Weight",y = "Miles per Gallon", color = "Cylinders")
```

Output:

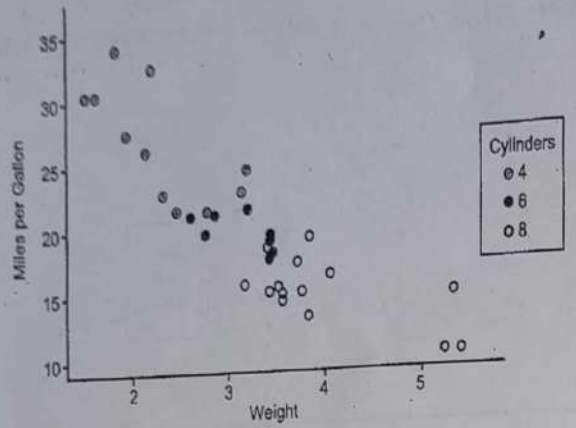


Fig. 4.14: Scatter plot with different shape and size

(2) Bar Plots

```
#To draw a bar plot of cyl(Number of Cylinders) according to the
Transmission type using geom_bar() and fill()
ggplot(mtcars, aes(x = cyl, fill = am)) + geom_bar() +
labs(x = "Cylinders", y = "Car count", fill = "Transmission")
```

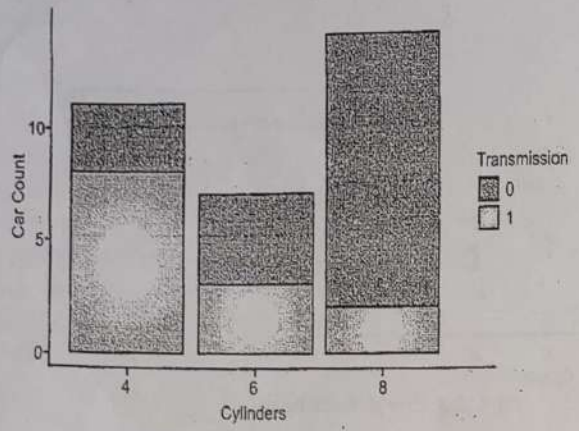


Fig. 4.15: Bar plot with different transmission type


```
#To find the proportion, we use position argument,as follows:
ggplot(mtcars, aes(x = cyl, fill = am)) +
geom_bar(position = "fill") +
labs(x = "Cylinders",y = "Proportion",fill = "Transmission")
```

Output:

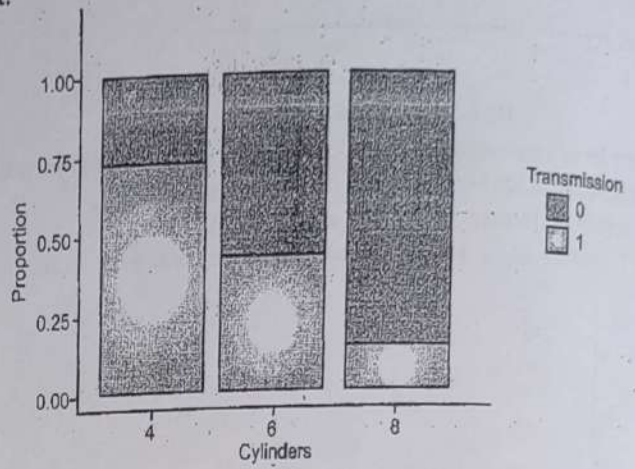


Fig. 4.16: Bar Plot with Proportion

(3) Themes

- It is used to change the attributes of non-data elements of our plot like text, lines, background, etc.
- We use the **theme_function( )** to make changes to these elements for data visualization in R.
- Some of the commonly used theme functions are as follows:
  - **theme_bw( ):** For white background and gray grid lines
  - **theme_gray( ):** For gray background and white grid lines
  - **theme_linedraw( ):** For black lines around the plot
  - **theme_light( ):** For light gray lines and axis
  - **theme_void( ):** An empty theme, useful for plots with non-standard coordinates or for drawings
  - **theme_dark( ):** A dark background designed to make colors pop out
  - **theme_classic( ):** A classic-looking theme, with x and y axis lines and no gridlines.

For example:

```
ggplot(mtcars, aes(x = cyl, fill = am)) + geom_bar(position = "fill")
+ theme_classic()+
labs(x = "Cylinders",y = "Proportion",fill = "Transmission")
```

Output:

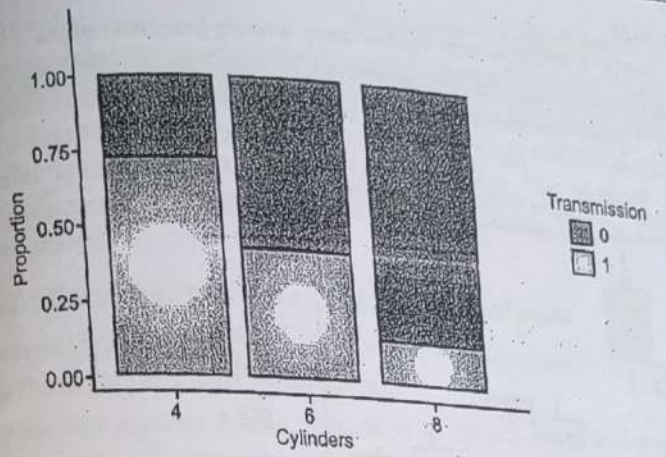


Fig. 4.17: Bar plot with Changed Theme

(4) Faceting

- It is used to further drill down data and split the data by one or more variables, and then plot the subsets of the data altogether for optimum data visualization in R.

For example:

```
#To facet the following plot according to gear(Number of Gears(3,4,5)), we
   use facet_grid() function as follows:
   ggplot(mtcars, aes(x = cyl, fill = am)) + geom_bar() + facet_grid
   (.~gear)+
#facet_grid(rows ~ columns)  theme_bw() +  labs(title = "Cylinder count by
   transmission and Gears",x="Cylinders",   y="Count",fill="Transmission")
```

Output:

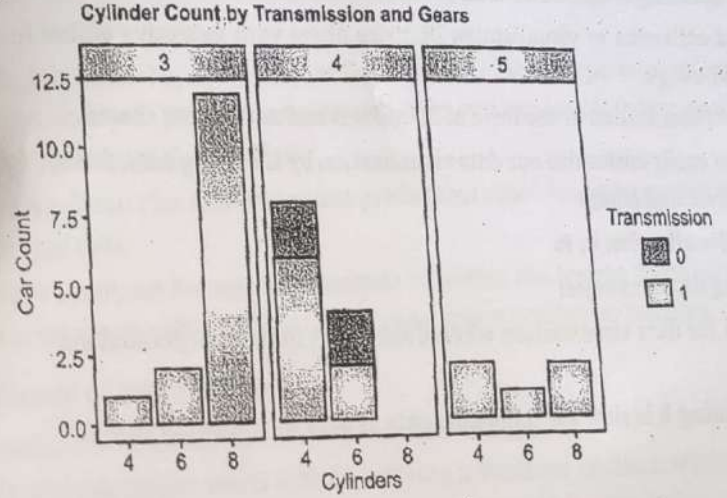


Fig. 4.18: Bar plot by data subsets

(5) Histograms

- **When to use:** Histogram is used to plot continuous variable. It breaks the data into bins and shows frequency distribution of these bins. We can always change the bin size and see the effect it has on visualization.

```
#To plot a histogram for mpg (Miles per Gallon), according to cyl(Number of
   Cylinders), we use the geom_histogram() function
   gplot(mtcars, aes(mpg,fill = cyl)) +geom_histogram(binwidth = 1) +
   theme_bw()+ labs(title = "Miles per Gallon by Cylinders",x = "Miles per
   Gallon",y= "Count",fill = "Cylinders")
```

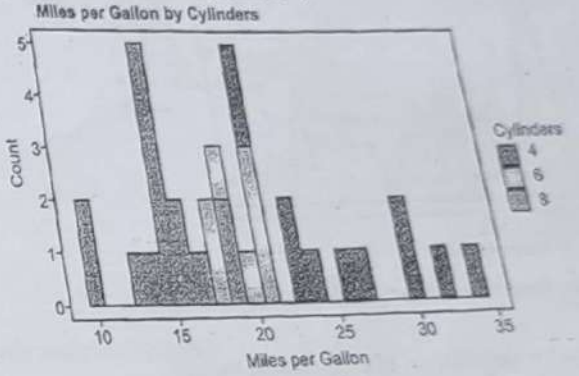


Fig. 4.19: Histogram

```
#To show overlapping, we set position to identity and alpha to 0.5
 ggplot(mtcars, aes(mpg,fill = cyl)) + geom_histogram(binwidth = 1,
 position = "identity", alpha = 0.5) + theme_bw() +
 labs(title = "Miles per Gallon by Cylinders",x = "Miles per Gallon",y
 "Count",fill = "Cylinders")
```

Output:

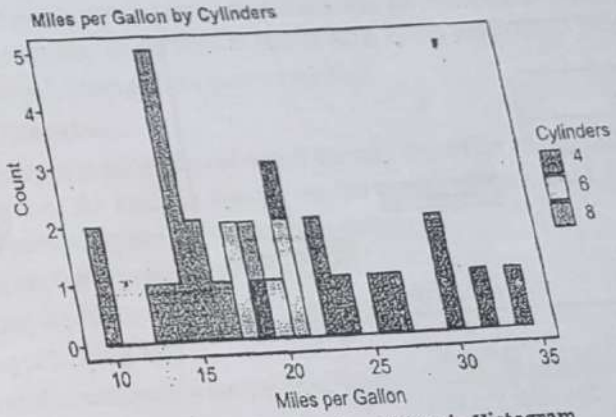


Fig. 4.20: Overlapping of data in Histogram

```
#To overcome overlapping, we can use the frequency polygon,
ggplot(mtcars, aes(mpg, color = cyl)) + geom_freqpoly(binwidth = 1)+
theme_bw()+
labs(title = "Miles per Gallon by Cylinders",x = "Miles per Gallon",y =
"Count",fill = "Cylinders")
```

Output:

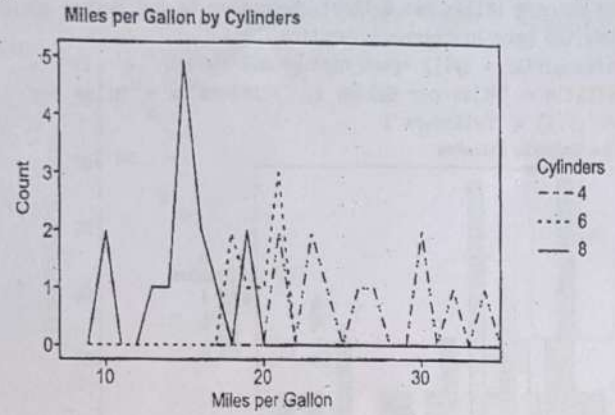


Fig. 4.21: Frequency Polygon in Histogram

(6) Boxplot

- The boxplot compactly displays the distribution of a continuous variable.

```
#To draw a Box plot
ggplot(mtcars, aes(x = cyl,y = mpg)) + geom_boxplot(fill = "cyan", alpha
= 0.5) + theme_bw() +
labs(title = "Cylinder count vs Miles per Gallon",x = "Cylinders",  y =
"Miles per Gallon")
```

Output:

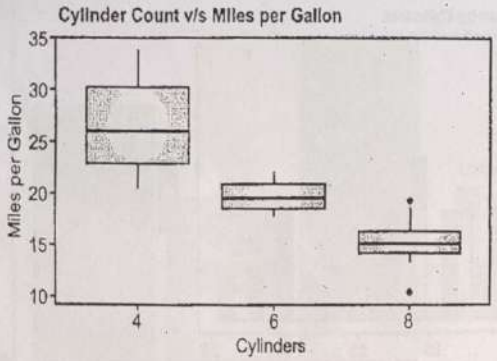


Fig. 4.22: Box Plot

For example:

```
#To draw a Box plot
ggplot(mtcars, aes(x = cyl,y = mpg,fill = am)) + geom_boxplot( alpha =
0.5) + theme_bw() +  labs(title = "Cylinder vs MPG by Transmission",x =
"Cylinders", y = "Miles per Gallon",fill = "Transmission")
```

Output:

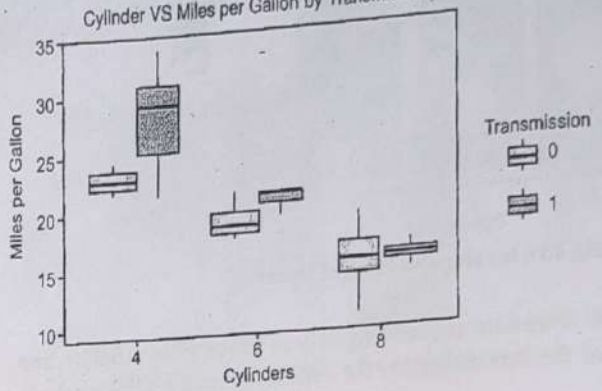


Fig. 4.23: Box Plot by groups

4.3.4 Advantages and Disadvantages of Data Visualization in R

- Let us see which are the advantages and disadvantages of data visualization in R programming:

Advantages of Data Visualization in R:

- R has the following advantages over other tools for data visualization:
  1. R offers a broad collection of visualization libraries along with extensive online guidance on their usage.
  2. R also offers data visualization in the form of 3D models and multipanel charts.
  3. Through R, we can easily customize our data visualization by changing axes, fonts, legends, annotations, and labels.

Disadvantages of Data Visualization in R:

- R also has the following disadvantages:
  1. R is only preferred for data visualization when done on an individual standalone server.
  2. Data visualization using R is slow for large amounts of data as compared to other counterparts.

4.4 DATA ANALYSIS

"Data analysis is defined as a process of cleaning, transforming, and modeling data to discover useful information for business decision-making. The purpose of Data Analysis is to extract useful information from data and taking the decision based upon the data analysis".

4.4.1 Types of Data Analysis : Techniques and Methods

There are several types of data analysis techniques that exist based on business and technology. The major types of data analysis are:

Text Analysis: Text Analysis is also referred to as Data Mining. It is a method to discover a pattern in large data sets using databases or data mining tools. It used to transform raw data into business information.

Statistical Analysis: Statistical Analysis includes collection, Analysis, interpretation, presentation, and modeling of data. It analyses a set of data or a sample of data. There are two categories of this type of Analysis - Descriptive Analysis and Inferential Analysis.

(a) **Descriptive Analysis:** This analyses complete data or a sample of summarized numerical data. It shows mean and deviation for continuous data whereas percentage and frequency for categorical data.

(b) **Inferential Analysis:** This analyses sample from complete data. In this type of Analysis, you can find different conclusions from the same data by selecting different samples.

Diagnostic Analysis: This Analysis is useful to identify behavior patterns of data. If a new problem arrives in your business process, then you can look into this Analysis to find similar patterns of that problem.

Predictive Analysis: This Analysis makes predictions about future outcomes based on current or past data.

Prescriptive Analysis: Prescriptive Analysis combines the insight from all previous Analysis to determine which action to take in a current problem or decision.

4.4.2 Process of Data Analysis

In this section, we discuss how to do data analysis using R.

Any data analysis project starts with identifying a business problem where historical data exists. A business problem can be anything which can include prediction problems, analyzing customer behavior, identifying new patterns from past events, building recommendation engines etc.

- The steps for solving a data analysis problem can be shown as below:

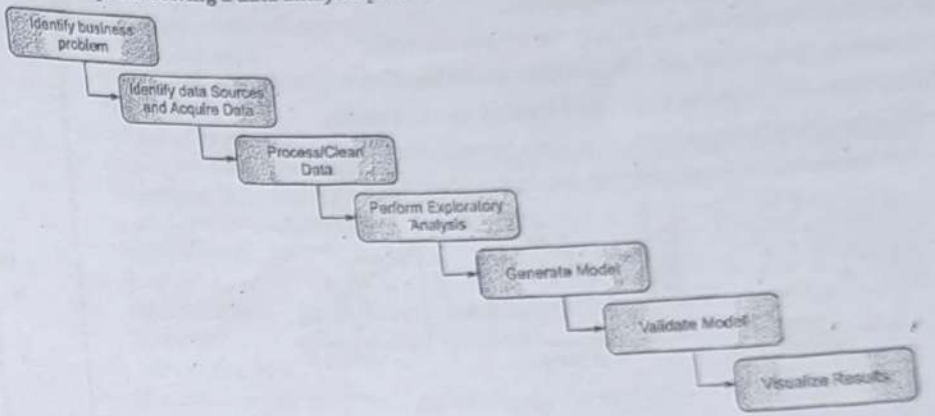


Fig. 4.24: Process of Data Analysis

Step 1: Identify Business Problem:

- This is the first step of analysis. Business identifies a problem and a problem statement with desired outcome is defined. In this stage, a Data Scientist should understand the problem statement, the domain knowledge of the problem. After thorough understanding of the problem statement, a Hypothesis will be proposed.

Step 2: Data Acquisition:

- In a second step, all the data sources related to the problem statement will be identified and pulled into a central repository. The data sources can vary from SQL databases to text files to csv files to online data. If the data size is large we may use Hadoop to pull, store and pre-process the data.

Step 3: Process/Clean Data:

- Data Clean step is considered to be one of the very important phases in Data analysis. The accuracy of the analysis depends on the quality of data. Data cleaning is the process of preventing and correcting these errors.
- Following are few approaches of Data Clean:
- Formatting the data as per the data analytical tools we use.
- Handling of Missing data.
- Data Transformations like normalizing the data Identifying outliers & handling etc.

Step 4: Exploratory Analysis:

- The objective of this step is to understand the main characteristics of the data. This analysis is generally done using visualizing tools.

- Performing an Exploratory analysis helps us:
  - To understand causes of an observed event.
  - To understand the nature of the data we are dealing with.
  - Assess assumptions on which our analysis will be based.
  - To identify the key features in the data needed for the analysis.
- **Graphical Techniques:** Scatter plots, box plots, histograms
- **Quantitative techniques:** Mean, median, Mode, Standard deviation

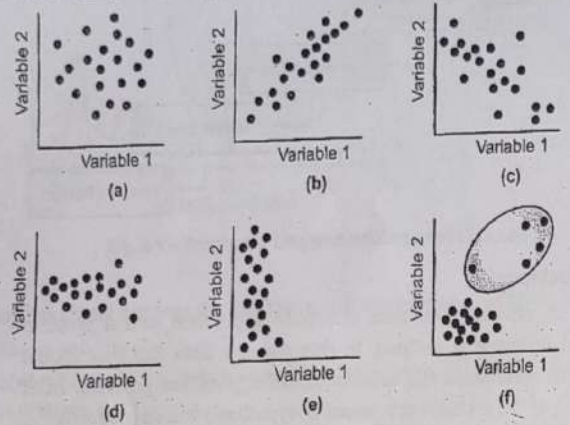


Fig. 4.25: Examples of Scatter Plots

Step 5: Model Generation & Validation:

- This step involves extracting features from the data and feeding them into the machine learning algorithms to build a model. Model is the solution proposed for the problem statement. This step involves: Model Selection, Model Training and Model Evaluation.
  - **Model Selection:** Based on the type of business problem we are dealing, a model will be built. For example, if the objective of the analysis is to predict a future event, we need to build a Regression model for prediction.
  - **Model Training:** After selecting the Model for the analysis, the entire dataset is divided into two parts – Training data & Test Data. $3/4^{th}$ of the entire data will be fed as input to the Model Algorithms.
  - **Model Evaluation:** Once the model is built. The next step is to test the model and validate it. The data used for testing the model is the remaining $1/3^{rd}$ of the dataset in the previous step.

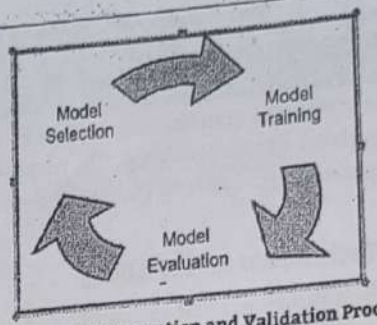


Fig. 4.26: Model Generation and Validation Process

Step 6: Visualize Results:

- This is the final step of Data analysis where the results of the model and problem solved will be presented generally in visual plots/graphs.
- Few visualizing tools: d3.js, ggplot2, tableau.

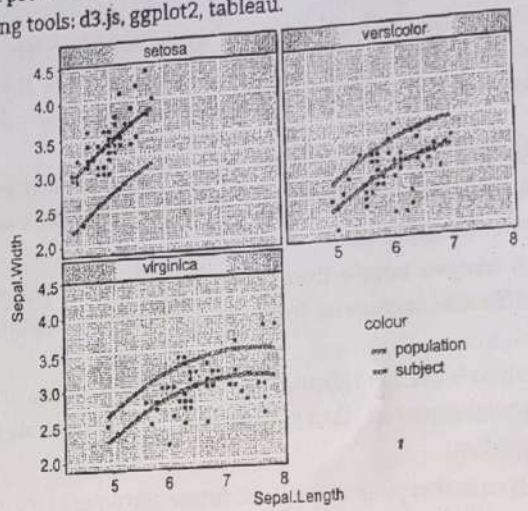


Fig. 4.27: Visual Plots / Graphs

Project

- The process of a machine learning project may not be linear, but there are a number of well-known steps:
  1. Define Problem.
  2. Prepare Data.
  3. Evaluate Algorithms.
  4. Improve Results.
  5. Present Results.

Here we use data which is provided by R platform for us **iris** dataset.

Here is what we are going to do in this step:

1. Load the iris data the easy way.
2. Load the iris data from CSV (optional, for purists)

(I) Load Data: The Easy Way

```
# attach the iris dataset to the environment
data(iris)
# rename the dataset
dataset <- iris
```

OR

Load the dataset from the CSV file as follows:

```
# define the filename
filename <- "iris.csv"

# load the CSV file from the local directory
dataset <- read.csv(filename, header=FALSE)

# set the column names in the dataset
colnames(dataset) <-
c("Sepal.Length","Sepal.Width","Petal.Length","Petal.Width","Species")
```

(II) Create a Validation Dataset

```
# create a list of 80% of the rows in the original dataset we can use for
training
validation_index <- createDataPartition( dataset$Species, p=0.80,
list=FALSE)
# select 20% of the data for validation
validation <- dataset[-validation_index,]

# use the remaining 80% of data to training and testing the models
dataset <- dataset[validation_index,]
```

(III) Summarize Dataset

- In this step, we are going to take a look at the data a few different ways:
  1. Dimensions of the dataset.
  2. Types of the attributes.
  3. Peek at the data itself.
  4. Levels of the class attribute.
  5. Breakdown of the instances in each class.
  6. Statistical summary of all attributes.

(1) Dimensions of dataset

- We can get a quick idea of how many instances (rows) and how many attributes (columns) the data contains with the dim function.

```
# dimensions of dataset
dim(dataset)
```

You should see 120 instances and 5 attributes:

```
[1] 120 5
```

(2) Types of attributes

- It is a good idea to get an idea of the types of the attributes. They could be doubles, integers, strings, factors and other types.

```
# list types for each attribute
sapply(dataset, class)
```

Output:

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
"numeric"	"numeric"	"numeric"	"numeric"	"factor"

(3) Peek at the Data

- It is also always a good idea to actually eyeball your data.

```
# take a peek at the first 5 rows of the data
head(dataset)
```

Output:

1. **Sepal.Length Sepal.Width Petal.Length Petal.Width Species**
2. 1 5.1 3.5 1.4 0.2 setosa
3. 2 4.9 3.0 1.4 0.2 setosa
4. 3 4.7 3.2 1.3 0.2 setosa
5. 5 5.0 3.6 1.4 0.2 setosa
6. 6 5.4 3.9 1.7 0.4 setosa
7. 7 4.6 3.4 1.4 0.3 setosa

(4) Levels of the Class attribute

- The class variable is a factor. A factor is a class that has multiple class labels or levels. Let's look at the levels:

```
# list the levels for the class
levels(dataset$Species)
```

(5) Class Distribution

```
# summarize the class distribution
percentage <- prop.table(table(dataset$Species)) * 100
cbind(freq=table(dataset$Species), percentage=percentage)
```

Output:

1. **freq percentage**
2. setosa 40 33.33333
3. versicolor 40 33.33333
4. virginica 40 33.33333

(6) **Statistical Summary**

Summary of each attribute.

```
# summarize attribute distributions
summary(dataset)
```

Output:

1. **Sepal.Length Sepal.Width Petal.Length Petal.Width Species**
2. Min. :4.300 Min. :2.00 Min. :1.000 Min. :0.100 setosa :40
3. 1st Qu.:5.100 1st Qu.:2.80 1st Qu.:1.575 1st Qu.:0.300 versicolor:40
4. Median :5.800 Median :3.00 Median :4.300 Median :1.350 virginica :40
5. Mean :5.834 Mean :3.07 Mean :3.748 Mean :1.213
6. 3rd Qu.:6.400 3rd Qu.:3.40 3rd Qu.:5.100 3rd Qu.:1.800
7. Max. :7.900 Max. :4.40 Max. :6.900 Max. :2.500

(IV) Visualize Dataset

- We are going to look at two types of plots:
  (1) Univariate plots to better understand each attribute.
  (2) Multivariate plots to better understand the relationships between attributes
- **Univariate Plot**

```
# split input and output
x <- dataset[,1:4]
y <- dataset[,5]
# boxplot for each attribute on one image
par(mfrow=c(1,4))
  for(i in 1:4) {
  boxplot(x[,i], main=names(iris)[i])
}
```

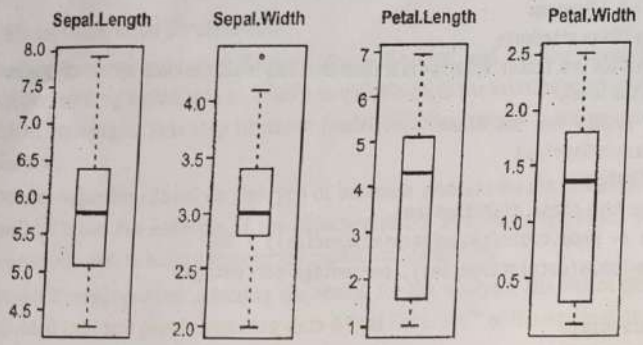


Fig. P1: Box and Whisker Plots in R

- **Multivariate Plots**

```
#scatterplot matrix
featurePlot(x=x, y=y, plot="ellipse")
```

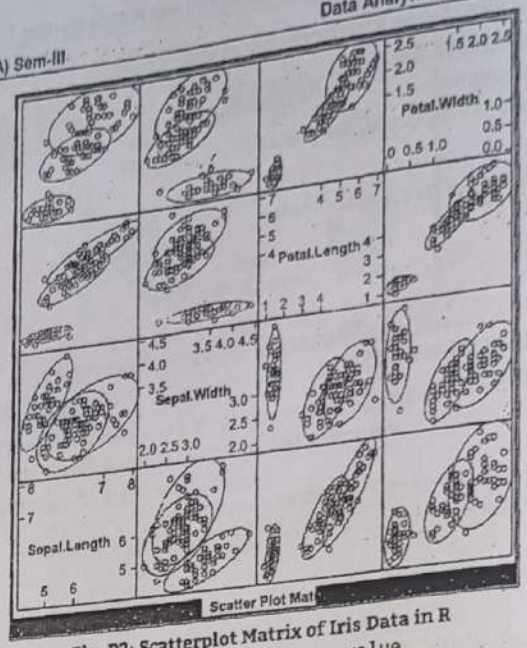



Fig. P2: Scatterplot Matrix of Iris Data in R

```
# density plots for each attribute by class value
scales <- list(x=list(relation="free"), y=list(relation="free"))
featurePlot(x=x, y=y, plot="density", scales=scales)
```

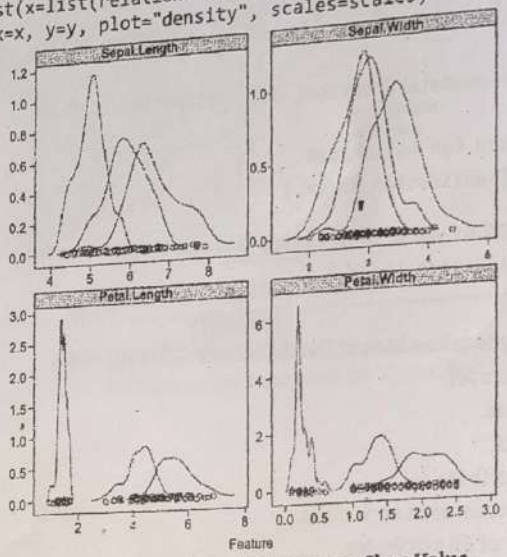


Fig. P3: Density Plots of Iris Data By Class Value

(V) Evaluate Some Algorithms

```
# Run algorithms using 10-fold cross validation
control <- trainControl(method="cv", number=10)
metric <- "Accuracy"
```

Build Models

- Let's evaluate 2 different algorithms:
  - k-Nearest Neighbors (kNN).
  - Support Vector Machines (SVM) with a linear kernel.

```
# kNN
set.seed(7)
fit.knn <- train(Species~., data=dataset, method="knn", metric=metric,
trControl=control)
# advanced algorithm
# SVM
set.seed(7)
fit.svm <- train(Species~., data=dataset, method="svmRadial",
metric=metric, trControl=control)
```

Select Best Model

```
# summarize accuracy of models
results <- resamples(list(lda=fit.lda, cart=fit.cart, knn=fit.knn,
svm=fit.svm, rf=fit.rf))
summary(results)
```

Output:

Models: lda, cart, knn, svm, rf
Number of resamples: 10
Accuracy

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
lda	0.9167	0.9375	1.0000	0.9750	1	1	0
cart	0.8333	0.9167	0.9167	0.9417	1	1	0
knn	0.8333	0.9167	1.0000	0.9583	1	1	0
svm	0.8333	0.9167	0.9167	0.9417	1	1	0
rf	0.8333	0.9167	0.9583	0.9500	1	1	0

Kappa

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
lda	0.875	0.9062	1.0000	0.9625	1	1	0
cart	0.750	0.8750	0.8750	0.9125	1	1	0
knn	0.750	0.8750	1.0000	0.9375	1	1	0
svm	0.750	0.8750	0.8750	0.9125	1	1	0
rf	0.750	0.8750	0.9375	0.9250	1	1	0

```
# compare accuracy of models
dotplot(results)
```

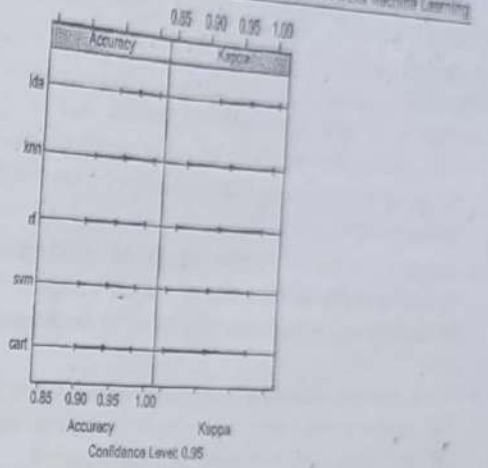



Fig. P 4: Comparison of Machine Learning on Iris Dataset in R

(VI) Make Predictions

```
# estimate skill of LDA on the validation dataset
predictions <- predict(fit.lda, validation)
confusionMatrix(predictions, validation$Species)
```

Output:

Confusion Matrix and Statistics
Reference

Prediction	setosa	versicolor	virginica
setosa	10	0	0
versicolor	0	10	0
virginica	0	0	10

Overall Statistics
Accuracy : 1
95% CI : (0.8843, 1)
No Information Rate : 0.3333
P-Value [Acc > NIR] : 4.857e-15
Kappa : 1
Mcnemar's Test P-Value : NA
Statistics by Class:

Class:	setosa	Class: versicolor	Class: virginica
Sensitivity	1.0000	1.0000	1.0000
Specificity	1.0000	1.0000	1.0000
Pos Pred Value	1.0000	1.0000	1.0000
Neg Pred Value	1.0000	1.0000	1.0000
Prevalence	0.3333	0.3333	0.3333
Detection Rate	0.3333	0.3333	0.3333
Detection Prevalence	0.3333	0.3333	0.3333
Balanced Accuracy	1.0000	1.0000	1.0000

Summary

- WEKA is an open source software provides tools for data preprocessing, implementation of several Machine Learning algorithms, and visualization tools so that you can develop machine learning techniques and apply them to real-world data mining problems.
- The typical data analysis workflow looks like this: collect data and put it in a file or spreadsheet or database. Then run some analyses, written in various scripts, perhaps saving some intermediate results along the way or maybe always working on the raw data.
- Data science is as much about manipulating data as it is about fitting models to data. Data rarely arrives in a form that we can directly feed into the statistical models or machine learning algorithms we want to analyze them with.
- The first stages of data analysis are almost always figuring out how to load the data into R and then figuring out how to transform it into a shape you can readily analyze.
- Data visualization is a technique used for the graphical representation of data. By using elements like scatter plots, charts, graphs, histograms, maps, etc., we make our data more understandable. Data visualization makes it easy to recognize patterns, trends, and exceptions in our data. It enables us to convey information and results in a quick and visual way.
- It is easier for a human brain to understand and retain information when it is represented in a pictorial form. Therefore, Data Visualization helps us interpret data quickly, examine different variables to see their effects on the patterns, and derive insights from our data.
- R programming provides comprehensive sets of tools such as in-built functions and a wide range of packages to perform data analysis, represent data and build visualizations.
- "Data analysis is defined as a process of cleaning, transforming, and modeling data to discover useful information for business decision-making. The purpose of Data Analysis is to extract useful information from data and taking the decision based upon the data analysis".
- There are several types of data analysis techniques that exist based on business and technology. The major types of data analysis are:
  - Text Analysis
  - Statistical Analysis
  - Diagnostic Analysis
  - Predictive Analysis
  - Prescriptive Analysis

Check Your Understanding

1. ..... programming language is a dialect of S.
   (a) B (b) C
   (c) R (d) K
2. Point out the wrong statement?
   (a) R is a language for data analysis and graphics
   (b) K is language for statistical modelling and graphics
   (c) One key limitation of the S language was that it was only available in a commercial package, S-PLUS
   (d) C is a language for data and graphics
3. R functionality is divided into a number of .....
   (a) Packages (b) Functions
   (c) Domains (d) Classes
4. Which Package contains most fundamental functions to run R?
   (a) root (b) child
   (c) base (d) parent
5. Which of the following is used for Statistical analysis in R language?
   (a) RStudio (b) Studio
   (c) Heck (d) KStudio
6. Which language is best for the statistical environment?
   (a) C (b) R
   (c) Java (d) Python
7. R has many functions regarding ..... .
   (a) Statistics, Biotechnology (b) Probability, Microbiology
   (c) Distributions, Physics (d) Statistics, Probability, Distributions
8. ..... hosts many add-on packages that can be used to extend the functionality of R.
   (a) CRAN (b) GNU
   (c) R studio (d) 450
9. Files containing R scripts ends with extension ..... .
   (a) .S (b) .R
   (c) .Rp (d) .SP
10. Collection of objects currently stored in R is called as ..... .
    (a) package (b) workspace
    (c) list (d) task
11. To select columns (variables) ..... function is used.
    (a) Select( ) (b) filter( )
    (c) mutate( ) (d) summarise( )
12. To filter (subset) rows ..... function is used.
    (a) Select( ) (b) filter( )

(c) mutate()
(d) summarise()

13. To create new variables ..... function is used.
(a) Select()
(b) filter()
(c) mutate()
(d) summarise()

ANSWER KEY

1. (c)	2. (b)	3. (a)	4. (c)	5. (a)
6. (b)	7. (d)	8. (a)	9. (b)	10. (b)
11. (a)	12. (b)	13. (c)		

Practice Questions

Q.I: Answer the following Questions in short.

1. What is Data Manipulation?
2. What is Data Visualization?
3. What is Data Analytics?
4. Enlist Steps of Data Analytics.
5. What is WEKA?

Q.II: Answer the following Questions.

1. What is Data Manipulation? Explain with example in R.
2. What is Data Visualization? Explain with example in R.
3. What is Data Analytics? Explain steps of Data Analytics.
4. What is Pipe Operator? Explain with example in R.
5. What is Summarise? Explain with example in R.
6. What is Base R Graphics? Explain with example in R.
7. What is Histograms? Explain with example in R.
8. What is Box Plot? Explain with example in R.
9. What is WEKA ? How to install it in Windows.

Q.III: Define the following terms.

1. Structured data
2. Select()
3. Filter()
4. Mutate()
5. Arrange()
6. Base R Graphics
7. Histogram
8. Bar Plot
9. ggplot2
10. group_by()