Savitribai Phule Pune University
Semester VI (Information Technology)

Data Science and Big Data Analytics

Unit III : Big Data Processing

Q. 1 Draw and explain Cloud File Systems GFS. **SPPU - April 19, 3 Marks**

Ans. :

- **Definition :** Google File System (GFS or GoogleFS) is a scalable distributed file system for large distributed data-intensive applications.
- It was developed around 2003. A new version of GFS, code named Colossus, was released in 2010. Colossus is also called as **GFS2.**
- Applications that are used by the world, such as Google Search, require to quickly read the data from disks, process it, and provide information to users. The information is stored on the disks and the disks are formatted using a file system. GFS2 provides a distributed file system where the information is stored on several disks.
- The following are the key characteristics of GFS2.

Characteristics and Features of GFS

1. **Fault tolerant :** If a few disks are corrupted, the data stored on them can still be restored and used.
2. **Big data size :** The file system can manage several peta bytes of data without crashing.
3. **High Availability :** The data is highly available (copied to several disks) and is present across various clusters of disks.
4. **Performance :** The file system provides very high performance for read and write from the disks.
5. **Resource Sharing :** The file system allows sharing disk resources across users.
6. **Google Cloud Services :** There are quite a few Google Cloud Services, such as Big Table, that are built on GFS2. Also, other Google apps, such as Gmail and Maps, use GFS2 as well.

Architecture of GFS : Fig. 3.1 shows block diagram that depicts the architecture of GFS at a high-level.

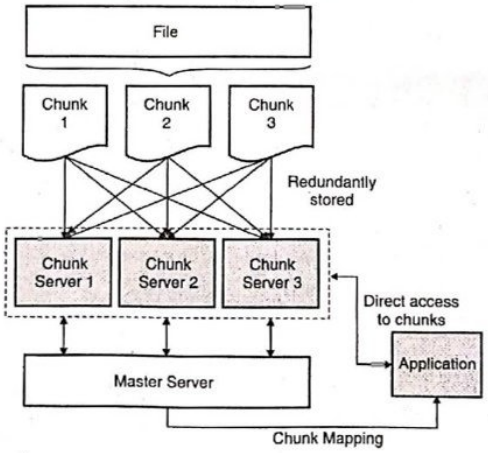


Fig. 3.1 : Architecture of Google File System

1

- The file is broken into multiple chunks. These chunks are redundantly stored across various chunk servers. There is a master server that manage the chunk mappings for files required for an application. The application then uses these chunk mappings to access the files it requires directly from the chunk servers.

Q. 2 Explain MapReduce paradigm with example. **SPPU - Dec. 18, 6 Marks**

OR What is MapReduce? Explain working of MapReduce with example. **SPPU - May 19, 9 Marks**

OR Explain working of MapReduce. **SPPU - Dec. 19, 5 Marks**

Ans. :

- **Definition :** MapReduce is a programming model for processing large datasets using parallel, distributed, and clustered compute nodes.
- Suppose that you have a large dataset of animal data in various forests around the world (sample data not real).

Animal Name	US	India	Australia	UK
Lion	200	300	400	500
Tiger	80	60	40	20
Rabbit	2000	1000	3000	5000
Elephant	40	20	10	5

- You have to find the total number of animals in a particular category.
- The MapReduce algorithm contains two important tasks - Map and Reduce.
  1. Map task converts the set of data into key-value pairs.
  2. Reduce task shuffles the key-value pairs and combines the similar data to produce the desired results.
- The MapReduce framework operates exclusively on <key, value> pairs, that is, the framework views the input to the job as a set of <key, value> pairs and produces a set of <key, value> pairs as the output of the job. Input and Output types of a MapReduce job are of the form:

Input < k1 , v1 > → **Map** → < k2, v2 > → **Combine** → < k2, v2 > → **Reduce** → <k3, v3 > **(Output)**

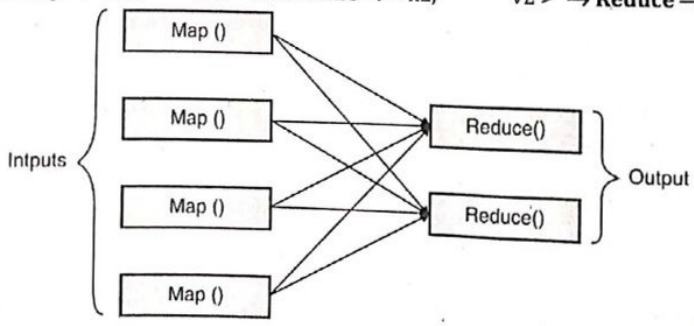


Fig. 3.2(a) : Map Reduce Tasks

Step 1 : Let's assume that there are four mapper tasks one for each country.

1. Mapper_US : (Lion, 200), (Tiger, 80), (Rabbit, 2000), (Elephant, 40)
2. Mapper_India : (Lion, 300), (Tiger, 60), (Rabbit, 1000), (Elephant, 20)
3. Mapper_Australia : (Lion, 400), (Tiger, 40), (Rabbit, 3000), (Elephant, 10)
4. Mapper_UK : (Lion, 500), (Tiger, 20), (Rabbit, 5000), (Elephant, 5)

Step 2 : Shuffling by animal category

1. Lion : (US, 200), (India, 300), (Australia, 400), (UK, 500)
2. Tiger : (US, 80), (India, 60), (Australia, 40), (UK, 20)
3. Rabbit : (US, 2000), (India, 1000), (Australia, 3000), (UK, 5000)
4. Elephant : (US, 40), (India, 20), (Australia, 10), (UK, 5)

Step 3 : Reduce by animal category

1. Lion : 1,400 2. Tiger : 200 3. Rabbit : 11,000 4. Elephant : 75

This MapReduce problem can be diagrammatically represented as shown in Fig. 3.2(b).

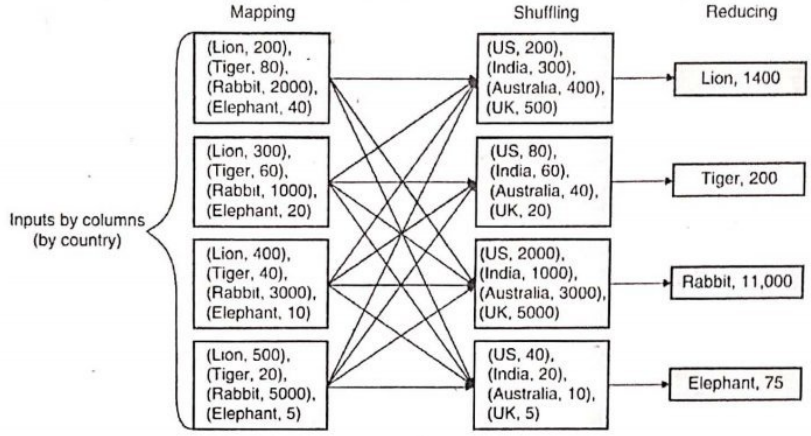


Fig. 3.2(b) : MapReduce problem

Q. 3 Explain the architecture of MapReduce in Hadoop. **(6 Marks)**

Ans. :

- **Definition :** Hadoop MapReduce is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.
- Fig. 3.3 shows the high-level architecture of MapReduce in Hadoop.

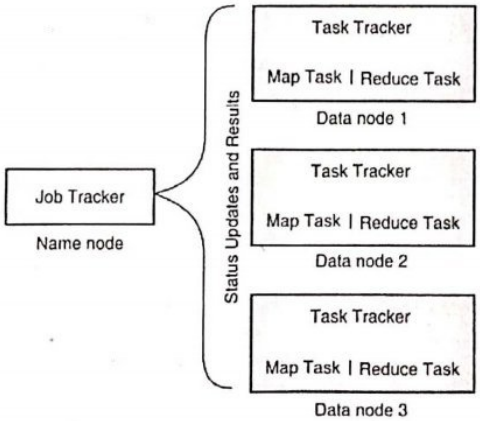


Fig. 3.3 : High-level architecture of MapReduce in Hadoop

There are two types of nodes.

1. **Name Node** : The MapReduce job is submitted to the name node that works as a master node in the cluster. It schedules the map and reduce tasks on other nodes in the cluster. It tracks the status reported by other nodes in the cluster and the overall MapReduce job submitted to it.
2. **Data Node** : Data notes are the processing units of the MapReduce job. Each data node is capable of running the map as well as the reduce task assigned to it by the name node. The data nodes keep track of the tasks assigned to them and update the name node accordingly.

   The reduce task takes the output from the map task as input and combines the data to produce results. MapReduce in Hadoop provides several benefits such as

   1. **Scalability :** You can process huge datasets stored in the Hadoop Distributed File System (HDFS).
   2. **Flexibility :** Hadoop can consume data from various sources for analysis.
   3. **Speed :** It provides parallel processing and requires minimal data movement between the nodes.
   4. **Simple :** You can write map and reduce functions in your choice of programming languages such as Java, C++, and Python.

Q. 4 Explain Hadoop Distributed File System. **SPPU - Dec. 18, 5 Marks**

OR Explain HDFS with respect to NameNode, DataNodes, Secondary NameNode with example. **SPPU - May 19, 8 Marks**

OR Explain working of Apache Hadoop with HDFS. **SPPU - Dec. 19, 4 Marks**

OR Describe the characteristics and features of HDFS. **(4 Mark)**

OR Explain the architecture of HDFS. **(6 Marks)**

Ans. :

- **Definition :** The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on general and low-cost hardware.

  It is mainly designed for processing large datasets.

Characteristics and Features of HDFS

1. **Fault tolerant :** HDFS is designed to be distributed across several servers. It is designed in such a way that if a few servers were to fail, it would not impact the overall data and its processing.
2. **Streaming data access :** HDFS is designed more for batch processing rather than interactive use by users. The emphasis is on high throughput of data access rather than low latency of data access.
3. **Large Data Sets:** HDFS is designed for processing large datasets in batches. It is typically used for data mining, analytics, and other big data projects.
4. **Write-once read-many model :** As you understand that HDFS is mainly used for processing large datasets, it is assumed that the underlying and historical data does not change. New data can be added but the older data remains as-is. A file once created, written, and closed need not be changed. This assumption simplifies data coherency issues and enables high throughput data access.
5. **Highly portable :** HDFS has been designed to be easily portable from one platform to another. This facilitates widespread adoption of HDFS as a platform of choice for a large set of applications.

Architecture of HDFS :

The Fig. 3.4 shows block diagram depicts the architecture of HDFS at a high-level.

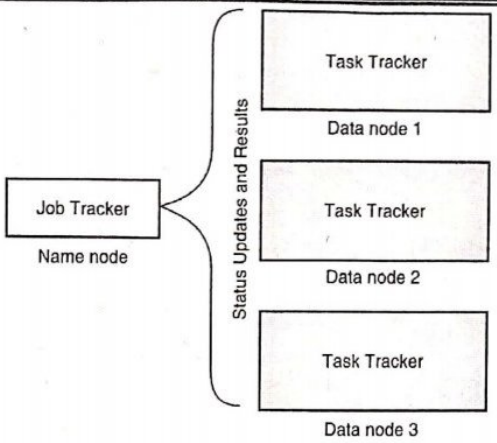


Fig. 3.4 : Architecture of HDFS

- HDFS has a master-slave architecture. There are two types of nodes.
  1. **Name Nodes** :An HDFS cluster consists of a single NameNode. Name node runs a master server that manages the file system and controls the access to files by clients.
  2. **Data Nodes** : There can be multiple Data Nodes. Data notes are the processing units in an HDFS cluster. Each data node manages the storage attached to it.
- HDFS exposes a file system namespace (grouping) and allows user data to be stored in files. Internally, a file is split into one or more blocks and these blocks are stored in a set of DataNodes. The NameNode executes file system namespace operations like opening, closing, and renaming files and directories. It also determines the mapping of blocks to DataNodes. The DataNodes are responsible for serving read and write requests from the file system's clients.
- The DataNodes also perform block creation, deletion, and replication upon instruction from the NameNode.
- Name Node and Data Node are designed to run on commodity machines.
- These machines typically run a Linux OS. HDFS is built using the Java language and hence any machine that supports Java can run the Name Node or the Data Node software. Java is highly portable and hence HDFS can be deployed on a wide range of machines.

Q. 5 What are the reasons for choosing NoSQL databases. **(6 Marks)**

Ans. : The major reasons for choosing NoSQL databases over the traditional relational databases are as shown in Fig. 3.5.

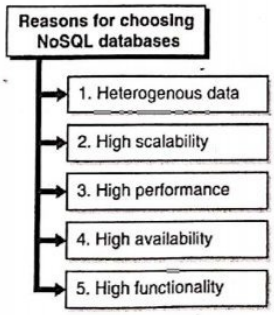


Fig. 3.5 : Reasons for Choosing NoSQL Databases

1. **Heterogenous Data :** A relational database requires fixed and declared schema before you can store data in it. It assumes that the data is homogenous meaning that all data would have same attributes or fields. Also, if you need to change the schema, you need to change all the rows in the table affecting all the previously stored records.

   Example : Assume that you own a simple online bookstore. Your books table looks like the following currently.

ISBN	Book Name	Author Name	Price
1234	ABCD	X1	100
2345	EFGH	X2	200

Slowly your business grows, and your customers now want more products such as laptops, mobile phones, and electronic items. How do you store information regarding these products in your books table?

You would either need to change the schema to something as following (just showing change required to accommodate one type of product).

ISBN	Book Name	Author Name	Price	Model	Company	Screen Size
1234	ABCD	X1	100	Null	Null	Null
2345	EFGH	X2	200	Null	Null	Null
Null	Null	Null	10,000	M1	S1	5 inch
Null	Null	Null	11,000	M2	E1	6 inch

Or you need to have separate tables – one for each type of item. So, if you sell 500 types of products, you end up managing 500 tables. Do you see the problems with these approaches?

That is precisely where NoSQL databases help where there is no need to store data under the constraints of a fixed and declared schema. You could store data with the relationships and fields as required for that particular data item without depending on the previous items or future items.

```
{
 "Books": {
  "Book1": {
   "ISBN": "1234",
   "Book_Name": "ABCD",
   "Author_Name": "X1",
   "Price": "100"
  },
  "Book2": {
   "ISBN": "2345",
   "Book_Name": "EFGH",
   "Author_Name": "X2",
   "Price": "200"
  }
```

```
},
"Mobiles": {
 "Mobile1": {
  "Model": "M1",
  "Company": "S1",
  "Screen_Size": "5 inch",
  "Price": "10,000"
 },
 "Mobile2": {
  "Model": "M2",
  "Company": "E1",
  "Screen_Size": "6 inch",
  "Price": "11,000"
 }
}
}
```


Each record could have its own set of attributes without depending upon the previous or future records that you can store in it.

Also, when you are carrying out data analytics on third-party collected data or data from various sources, not all fields may have values. It is thus useful to use NoSQL in this scenario where you do not have enough control over the schema that the collected data would surely follow.

2. **High Scalability :** Imagine companies like Google and Facebook that deal with millions of data reads and writes per second for such a wide user base globally. You require highly scalable database systems that could quickly respond to such high workload demands. Relational databases are hard to partition and run across distributed clusters. The only way to scale relational databases is to add more hardware resources (scaling up) instead of distribution them (scaling out). However, NoSQL databases are generally designed to scale out by using distributed clusters of hardware instead of scaling up by adding expensive and robust servers.

   Also, the transactional and consistency guarantees provided by the relational databases make it almost impossible to scale a relation database across more than a few machines, especially in the environments that require significantly high number of writes as in the case of modern web applications.

   NoSQL databases do not provide tight and immediate consistency (all tables are consistently updated at the same time). Instead, they provide eventual consistency which means that there could be inconsistent data across various tables for some time and only after the elapse of certain time period the data is made consistent. For example, if you just accepted someone's friend request on Facebook, it is possible that someone else visiting your profile (accessing Facebook records from a different database server) may not see your new friend immediately. Your new friend could be visible only after sometime (say 2 seconds). This is called eventual consistency.

3. **High Performance :** NoSQL database are optimised for specific data models and access patterns that enable higher performance than trying to accomplish similar functionality with relational databases. Because of highly distributed clusters, you could perform multiple reads and writes at a time in parallel.

4. **High Availability :** Because of the distributed nature of NoSQL databases, they provide high availability. High availability provides a guarantee that every application request receives a response from the database server. Even if a few nodes (machines) were to fail, the overall system could still be responsive as the workload would be distributed against other available database instances. With traditional relational databases, if the database server was to go down then it could mean unavailability of the application.
5. **High Functionality :** NoSQL databases provide highly functional APIs and data types that are purposely built for each of their respective data models. APIs have mostly become the default standard for doing things with respect to modern and cloud-based applications.

Q.6 What are four major categories of NoSQL Tools (stores) ? **SPPU - Dec. 18, 4 Marks**

OR Explain term : Key-value store **SPPU - May 19, 2 Marks**

OR Explain NoSQL databases : Key value store **SPPU - Dec. 19, 2 Marks**

OR Explain term : Document store **SPPU - May 19, 2 Marks**

OR Explain NoSQL databases : Document store **SPPU - Dec. 19, 2 Marks**

OR Explain term : Column family store **SPPU - May 19, 2 Marks**

OR Explain NoSQL databases : Column family store **SPPU - Dec. 19, 2 Marks**

OR Explain term : Graph Databases **SPPU - May 19, 2 Marks**

OR Explain NoSQL databases : Graph Databases **SPPU - Dec. 19, 2 Marks**

Ans. :

- At a high-level, there are four types of NoSQL databases.

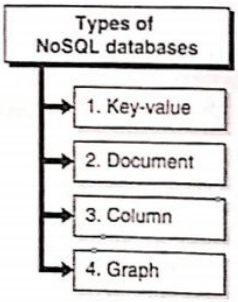


Fig. 3.6(a) : Types of NoSQL databases

1. Key-Value

- **Definition :** A key-value database is a type of NoSQL database that uses a simple key-value method to store data.
- A key-value database stores data as a collection of key-value pairs in which a key serves as a unique identifier. Both keys and values can be anything, ranging from simple objects to complex compound objects such as binary files containing audio, images, and videos. Understand key-value store as a dictionary where each word is a key and its meaning (which could be anything and any long based on the word) as the value.
- How long is the value or what precisely is the value does not matter. It is up to the reader to make the right sense of the value retrieved from the key.

- Following is an example of how data is stored as a key-value pair.

Key	Value
Apple	Red, 100, Sweet, Healthy
Chips	Yellow, 50, Spicy, Unhealthy, High Sodium
Rice	White, 80, Plain, Healthy, Carbohydrate, 3 years, Kerala

- Key-value stores are also called associative arrays, hash tables or maps. Key-value databases are highly partitionable and allow horizontal scaling at scales that other types of databases cannot achieve. Queries are performed on the basis of keys. In general, key-value stores have no query language. They provide a way to store, retrieve and update data using simple get, put, and delete commands. There are usually no fields to update and instead the entire value other than the key must be updated if changes are required. Some of the common usage of the key-value store are for storing session IDs, shopping cart information and array of information such as customer or product details. Some of the common key-value databases are Apache Cassandra, Amazon DynamoDB, Redis, MemcacheDB and Couchbase.

2. Document

- **Definition :** A Document database is a type of NoSQL database that stores data as a document in a well-formatted structure.
- Document data store is very much like key-value store with an exception that the value is more structured and is stored in a document format such as XML or JSON. Using document store, you can store data as well as some field like metadata to give hints on how the data should be interpreted. Here are a few simple examples of how data could be stored as a document using JSON.
- Table Name = Shopping_Cart

```
{
 "ID": "1",
 "Product": "Book",
 "Price": "100"
}
```

- Table Name = My Amazon Storage buckets

```
{
 "Filename": "My_Assignment.rar",
 "URL": "https://my.assignment.com/My_Assignment.rar",
 "Size": "100 MB",
 "Last_Updated": "12-Apr-2020"
}
```

- The underlying structure of the documents can be used to query and customise the interpretation of the document's content. The documents could be indexed for fast searches and also query based on specific values. Document databases enable flexible indexing, powerful queries, and analytics over collections of documents.

- The flexible, semi-structured, and hierarchical nature of documents and document databases allow developers to evolve the data store as their applications require without restricting themselves to the predefined and fixed relational schema. The document model works well with use cases such as catalogues, user profiles, and content management systems where each document is unique and could evolve over time. Some of the common document databases are Apache CouchDB, Amazon DocumentDB, Couchbase, and MongoDB.

3. Column

- **Definition :** Column database is a` type of NoSQL database that stores data as a collection of columns optimised to retrieve the entire column values at a time.
- Relational databases store all the data in a particular table's rows together on the disk, making retrieval of a particular row (entire record) fast whereas column databases serialise all the values of a particular column together on the disk, which makes retrieval of a large amount of a specific attribute fast.
- This approach works better in scenarios where aggregate queries and analytics are required over specific fields and entire record is not of so much importance. All data within each column has the same type which makes it ideal for compression and aggregation (count, sum, avg, min, max).Columns are logically grouped into column families. Each row can have any number of columns and is not dependent on any particular schema.
- Following is an example of how data could be stored column wise.

Row Key	Name		Address				Order		
1	First Name	Last Name	Street	Area	Pincode	State	Items	Quantity	Value
	Adam	Ness	S1	A1	101010	MH	Item 1	2	100

Row Key	Name			Address				Order			Delivery Instructions	
	First Name	Middle Name	Last Name	Street	Area	Pincode	State	Items	Quantity	Value	Where?	When?
2	R	K	Singh	A3	West	201020	KA	Item 1	2	100	Home	17:00 - 20:00
								Item 2	3	200		
								Item 3	1	50		

- There are column families such as Name, Address, order with fields such as first name, last name, quantity, etc. Each row can have its own set of columns without depending on a pre-defined schema.
- Some of the common usages of the column store are for content management systems, blogs, reviews and ratings management systems, and storing metrics and counters. Some of the common column databases are Cassandra, HBase and Bigtable.

4. Graph

- **Definition :** Graph database is a type of NoSQL database that stores relationships between entities and makes it easy to navigate through them.

- Graph databases are purposely built to store and navigate relationships. Graph databases use nodes to store data entities, and edges to store relationships between entities. An edge always has a start node, end node, type, and direction, and an edge can describe parent-child relationships, actions, ownership, and other attributes. There is no limit to the number and kind of relationships a node can have.
- A graph in a graph database can be traversed along specific edge types or across the entire graph. In graph databases, traversing the joins or relationships is very fast because the relationships between nodes are not calculated at query time but are persisted in the database.
- Fig. 3.6(b) shows a simple family graph.

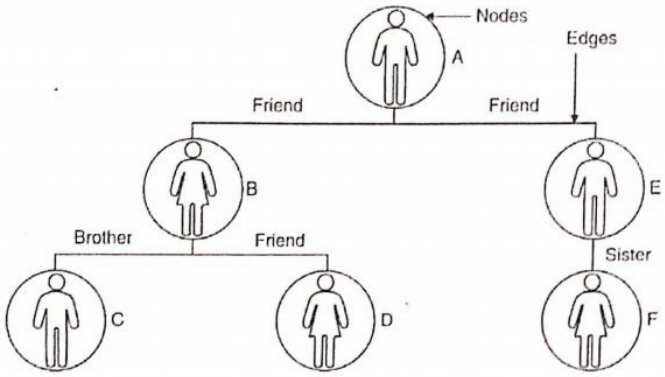


Fig. 3.6(b) : Simple family graph

- A graph could really be dense and complex as the number of edges and nodes increase. For example, here is slightly denser graph than the previous one.

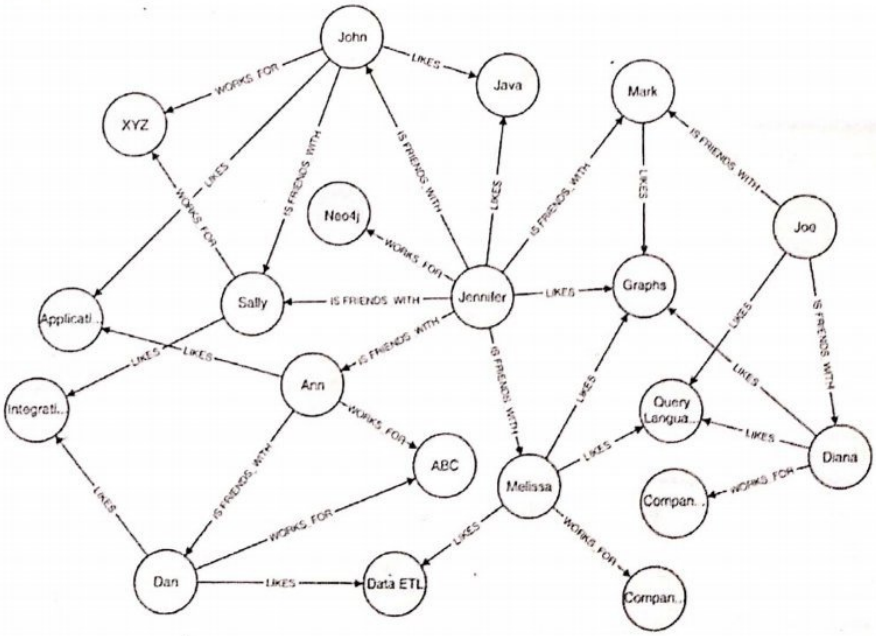


Fig. 3.6(c) : Sample graph

11

- Some of the common usage of the graph database are social network management, fraud detection, recommendation engines, and path optimisation systems. Graph databases are useful when you need to create relationships between data and quickly query the relationships between them. Some of the common graph databases are Neo4j, Amazon Neptune, FlockDB and OrientDB.

Q. 7 Compare between relational database and No SQL database. **(6 Marks)**

Ans. :

Table 3.1 summarises the high-level comparison between relational databases and NoSQL databases.

Table 3.1 : Comparison between Relational Database and NoSQL Database

Comparison Attribute	Relational DB	NoSQL DB
Schema	Required	Not required
Can store	Structured data only	Both structured and unstructured data
Scalability	Low	High
Complex queries	Easy to write	Difficult to write
Consistency	Immediate	Eventual
Performance	Low	High
Flexibility	Low	High
Data Model	Tables (Rows and Columns)	Various – such as Key-Value, Graph, Document
Good for	Transaction processing	Low latency data access and analytics
Cost of Operation	High	Low

Q. 8 What is Brewer's Theorem? Explain all CAP parameters. **(6 Marks)**

Ans. :

NoSQL databases provide great scalability and performance as they are deployed over distributed systems. However, distributed systems need to work consistently and be in sync with each other. The three key parameters that define the behaviour of any distributed system are

1. **Consistency :** When data is stored on multiple nodes then all the nodes should have the same data. When the data is updated at one node then the same update should be made at the other nodes storing the same data as well. A distributed system is said to be in a consistent state if the transaction starts on any node with the system being in a reliable state and ends with the system being in a reliable state as well. Remember the difference between tight consistency and eventual consistency?
2. **Availability :** Availability means that the system is responsive. To achieve a higher order of availability it is required that the system remains operational 100% all the time. So, if you make a request you should always get a non-error response.
3. **Partition Tolerance:** Distributed systems have machines (nodes) that are connected to each other over a network. Partition tolerance means that the system continues to function even when the network communication among the nodes is unreliable. The nodes may not be able to communicate with each other all the time and there could be network delays or any other network related errors such as packet drops.

They are abbreviated as CAP and are guided by CAP theorem.

- **Definition :** The CAP theorem states that no distributed system can guarantee Consistency, Availability and Partition Tolerance at the same time. You can only pick any two out of the three parameters.

- This is also called Brewer's Theorem. So, it depends upon your functional and performance requirements on what two parameters you choose.

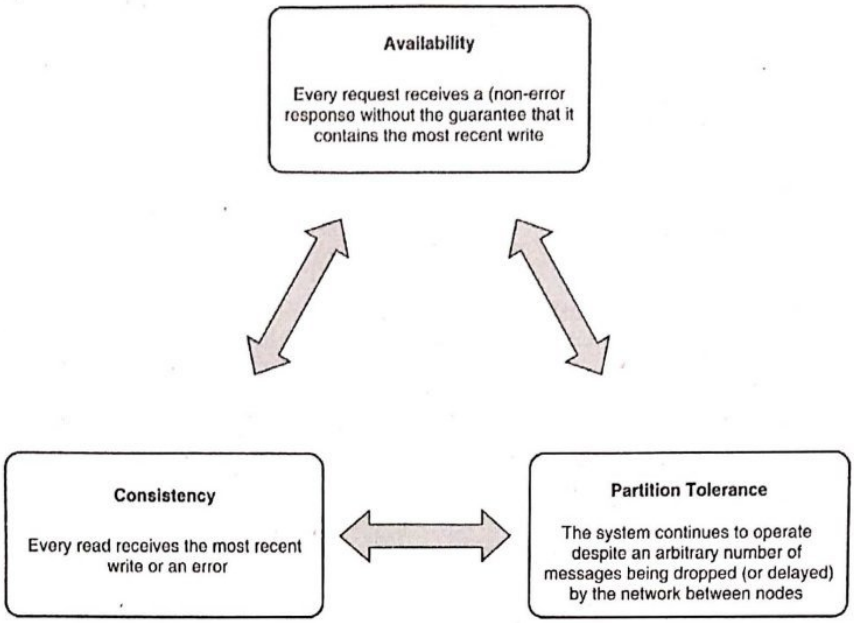


Fig. 3.7 : CAP parameters

You may choose

1. **Consistency and Availability (CA) :** These systems may not work well over unreliable networks. In the absence of network failure both availability and consistency can be satisfied. But, network failures cannot be totally eliminated. So, a system could have consistency and availability but may not have partition tolerance at the same time. These systems would have problems when there are network issues. Traditional RDBMS systems work better in such scenarios as they are not significantly distributed.
2. **Consistency and Partition Tolerance (CP) :** In this combination, you compromise with availability. A system may be consistent and may tolerate network failures but cannot then provide availability guarantee. A few examples of such systems are MongoDB, HBase and Redis.
3. **Availability and Partition Tolerance (AP) :** In this combination, you compromise with consistency. A system may be highly available and may tolerate network failures but cannot then provide consistency guarantee. A system can however provide eventual consistency that is consistency after some time has elapsed. A few examples of such systems are Cassandra, CouchDB and Amazon DynamoDB.

- Like traditional relational databases provide ACID (Atomic, Consistency, Isolation, Durable) properties, similarly NoSQL databases usually provide BASE properties derived out of CAP theorem.
- A BASE system gives up on consistency
  - Basically Available indicates that the system guarantees availability.
  - Soft state indicates that the state of the system may change over time, even without input. This is because of the eventual consistency model.
  - Eventual consistency indicates that the system will become consistent over time, given that the system doesn't receive input during that time.

Q. 9 Explain the high-level steps involved in text analysis. **(4 Marks)**

Ans. : Steps in Text Analysis

The high-level steps involved in text analysis are as following.

1. **Parsing :** It is the process that takes unstructured text and imposes a structure for further analysis. The unstructured text could be a plain text file, a weblog, an Extensible Markup Language (XML) file, a Hypertext Markup Language (HTML) file, or a Word document. Parsing deconstructs the provided text and renders it in a more structured way for the subsequent steps.
2. **Search and retrieval :** It is the identification of the documents in a corpus that contain search items such as specific words, phrases, topics, or entities like people or organizations. These search items are generally called key terms. Search and retrieval originated from the field of library science and is now used extensively by web search engines.
3. **Text mining :** It uses the terms and indexes produced by the prior two steps to discover meaningful insights pertaining to domains or problems of interest. With the proper representation of the text, many of the techniques you learnt earlier, such as clustering and classification, can be adapted to text mining. For example, the k-means clustering can be modified to cluster text documents into groups, where each group represents a collection of documents with a similar topic. The distance of a document to a centroid represents how closely the document talks about that topic. Classification tasks, such as sentiment analysis and spam filtering, are prominent use cases for the naïve Bayes classifier. Text mining may utilise methods and techniques from various fields of study, such as statistical analysis, information retrieval, data mining, and natural language processing.

Q. 10 Describe a few text pre-processing techniques. **(6 Marks)**

Ans. : Text Pre-Processing Techniques

Some of the common text pre-processing techniques are as shown in Fig. 3.8.

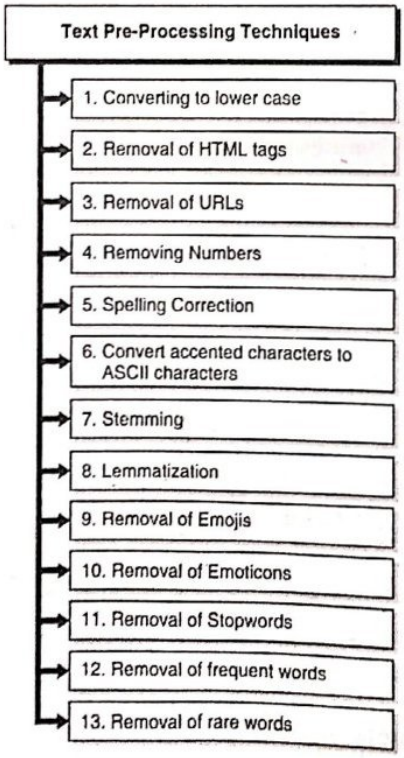


Fig. 3.8

1. **Converting to Lower case (Lowercasing) :** Converting all the text into the lower case is a simple and effective approach for text analysis. If you are not applying lower case conversion on words like DOG, doG, dog, dOG, then these all words would be treated as different.

Before lowercasing	After lowercasing
DOG	dog
dOG	dog
dOg	dog

2. **Removal of HTML tags :** The chances to get HTML tags in your text data is quite common specially when you are extracting or scraping data from different websites. You don't get any valuable information from these HTML tags. So, it is better to remove them from textual data.

Before HTML tags removal	After HTML tags removal
Data science project	Data science project

3. **Removal of URLs :** URL is the short-form of Uniform Resource Locator. The URLs within the text refer to the location of another website or anything else. These URLs are of no use to you in textual analysis. You can remove them.

Before URL removal	After URL removal
For more information go to https://www.google.com	For more information go to

4. **Removing Numbers :** If your analysis does not require numbers, you can remove them.

Before number removal	After number removal
The weight of panda is 650 Kgs	The weight of panda is Kgs

5. **Spelling Correction :** Similar to lowercasing, spelling correction is another important pre-processing technique that avoids treating wrongly spelled words different from correctly spelled words.

Before spelling correction	After spelling correction
Team India wno today	Team India won today

6. **Convert accented characters to ASCII characters :** You might have seen special characters at the top of the common letter or characters. These are accented characters. For example, e in the word résumé has accents. If you don't remove these, then the text analysis model will consider resume and résumé as different words, even if both are the same.

With accented characters	Without accented characters
I submitted my résumé on the portal	I submitted my resume on the portal

7. **Stemming :** Stemming is reducing words to their base or root form by removing a few suffix characters from words. Stemming is a text normalisation technique. There are various stemming algorithms but the most widely used one is porter stemming.

Before stemming	After stemming
Learning	Learn
Books	Book
Caring	Car
Obesity	Obes
Causes	Caus

But stemming doesn't always provide the correct form of words because it blindly follows the rules like removing suffix characters to get base words irrespective of ensuring correctness. Sometimes, stemmed words don't relate to original ones and sometimes they may also give non-dictionary or improper words.

8. **Lemmatization :** Lemmatization is similar to the stemming technique that aims to get the base words. But, unlike stemming that might produce improper words, the lemmatization process does not only trim the suffix characters but also uses lexical knowledge bases to get original words in their right forms.

   Hence, the result of lemmatization is better than stemming.

Before lemmatization	After lemmatization
Learning	Learn
Books	Book
Caring	Care
Obesity	Obesity
Causes	Cause

9. **Removal of Emojis :** In today's online communication, emojis play a very crucial role. Emojis are small images using which users may express their feelings. Until and unless you are making sense out of these emojis, you can remove them for text analysis.

With emojis	Without emojis
I am super happy ☺	I am super happy

10. **Removal of Emoticons :** Unlike emojis which are tiny images, an emoticon portrays a human facial expression using just keyboard characters, such as letters, numbers, and punctuation marks without using any images. Until and unless you are making sense out of these emoticons, you can remove them for text analysis.

With emoticons	Without emoticons
I am super happy ;)	I am super happy

11. **Removal of Stopwords :** Stopwords are common words that are mostly irrelevant for text analysis. For example, "a", "an", "the", "is", "for", etc.

With stopwords	Without stopwords
I ate an apple	I ate apple

12. **Removal of frequent words :** Stopwords are language specific. If you are working on text analysis for a particular domain, it may involve frequent words that may not give a lot of useful information. For example, the word "experiment" may appear several times if you are performing text analysis on a scientific research or thesis.

    You can remove such frequent words from text analysis.

With frequent word	Without frequent word
In the experiment it was found that	In the it was found that

13. **Removal of rare words :** You can also remove rare words from text analysis as it is unlikely to be found multiple times or provide any useful information.

With rare word	Without rare word
Petrichor was all around	Was all around

Q. 11 Write a short note on Bag-of-Words. **(4 Marks)**

Ans. :

Bag-of-Words

- In bag-of-words (BoW), a text document is converted into a vector of counts. The vector contains an entry for every possible word in the vocabulary.
- If the word say, "aardvark" appears three times in the document, then the feature vector has a count of 3 in the position corresponding to that word. If a word in the vocabulary doesn't appear in the document, then it gets a count of 0.

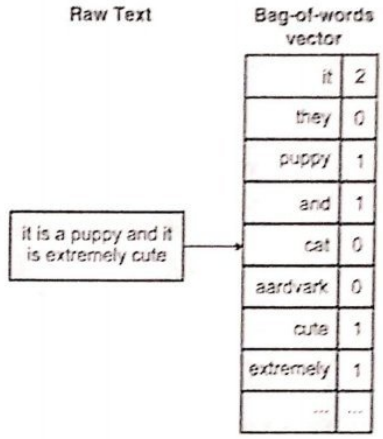


Fig. 3.9

- Bag-of-words converts a text document into a flat vector. It is "flat" because it does not contain any of the original textual structures. The original text is a sequence of words. But a bag-of-words has no sequence. It just remembers how many times each word appears in the text.
- The ordering of words in the vector is not important, as long as it is consistent for all documents in the dataset. Neither does bag-of-words represent any concept of word hierarchy. For example, the concept of "animal" includes "dog," "cat," "raven," etc. But in a bag-of-words representation, these words are all equal elements of the vector.

Q. 12 Write a short note on Bag-of-n-Grams. **(4 Marks)**

Ans. :

Bag-of-n-Grams

- Bag-of-n-Grams is a natural extension of bag-of-words. An n-gram is a sequence of n tokens. A word is essentially a 1-gram, also known as a unigram. After tokenisation, the counting mechanism can group individual tokens into word counts or count overlapping sequences as n-grams. For example, the sentence "Rahul knocked on the door" generates the n-grams "Rahul knocked," "knocked on," "on the," and "the door" for n = 2.
- n-grams retain more of the original sequence structure of the text, and therefore the bag-of-n-grams representation can be more informative. However, this comes at a cost. Theoretically, with k unique words, there could be k^2 unique 2-grams (also called bigrams). In practice, there are not nearly so many, because not every word can follow every other word. Nevertheless, there are usually a lot more distinct n-grams (n > 1) than words.

- This means that bag-of-n-grams is a much bigger and sparser feature space. It also means that n-grams are more expensive to compute, store, and model. The larger n is, the richer the information, and the greater the cost.

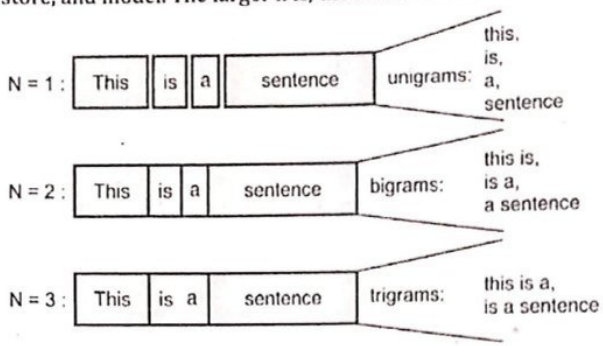


Fig. 3.10

Q. 13 Explain Term Frequency and Inverse Document Frequency. **(4 Marks)**

Ans. :

1. Term Frequency (TF)

Term Frequency (TF) measures the frequency of a word in a given document. It highly depends on the length of the document and the generality of word. For example, a very common word such as "the" can appear multiple times in a document. But if you take two documents, the one which has 100 words and the other which has 10,000 words, then there is a high probability that the count of "the" would be much higher in the 10,000 worded document. But it would be incorrect to assume that the longer document is more important than the shorter document just on the basis of term frequency. Hence, you normalise the term frequency value. You divide the frequency with the total number of words in the document to get the normalised term frequency value.

You can calculate term frequency as following.

$$\text{Term Frequency (TF)} = \frac{\text{count of the term in the document (t)}}{\text{total number of terms in the document (d)}}$$

One thing to keep in mind is that you need to finally vectorise the document. When you are planning to vectorise the documents, you cannot just consider the words that are present in that particular document. If you do that, then the vector length will be different between the documents, and it will not be feasible to compute the similarity. So, you vectorise the documents on the vocab. Vocab is the entire list of all possible words in the corpus.

When you are vectorising the documents, you check for each words count. In worst case if the term doesn't exist in the document, then that particular TF value will be 0 and in other extreme case, if all the words in the document are same, then it will be 1. The final value of the normalised TF value will be in the range of [0 to 1].

2. Inverse Document Frequency (IDF)

Inverse Document Frequency (IDF) is the measure of the importance of a word. Term frequency (TF) does not consider the importance of words. Some words such as "of", "and", etc. can be most frequently present but are of little significance. IDF provides weightage to each word based on its frequency in the corpus D.

IDF can be calculated as following.

$$\text{IDF} = \log\left(\frac{\text{Total number of documents N in corpus D}}{\text{number of documents containing the term t}}\right)$$

When you calculate IDF, it will be very low for the most occurring words such as stopwords ("the", "is", "was", etc.). IDF is constant per corpus whereas TF is document specific.

18

Q. 14 How is the word fox relevant to the following documents?

Document 1 = A quick brown fox jumps over the lazy dog. What a fox!

Document 2 = A quick brown fox jumps over the lazy fox. What a fox! **(6 Marks)**

Ans. :

Document 1 and 2 both have 12 words each.

Let's calculate the term frequency for the word "fox" in the respective documents.

$$\text{Term Frequency (TF)} = \frac{\text{count of the term in the document (t)}}{\text{total number of terms in the document (d)}}$$

You are only interested in the word "fox" here. Other words, hence, do not matter.

$$\text{Term frequency } TF_1 \text{ (fox, document 1)} = \frac{2}{12} = 0.17$$

$$\text{Term frequency } TF_2 \text{ (fox, document 2)} = \frac{3}{12} = 0.25$$

IDF can be calculated as following.

$$\text{IDF} = \log\left(\frac{\text{Total number of documents N in corpus D}}{\text{number of documents containing the term t}}\right)$$

The entire corpus D has 2 documents. Both the documents contain the word "fox".

$$\text{Hence, IDF} = \log\left(\frac{2}{2}\right) = 0$$

Let's calculate TFIDF score for both the documents.

TFIDF = Term Frequency (TF) × Inverse Document Frequency (IDF)

$TFIDF_1 = TF_1 \times IDF = 0.17 \times 0 = 0$

$TFIDF_2 = TF_2 \times IDF = 0.25 \times 0 = 0$

Hence, based on TFIDF scores, you can conclude that the word "fox" is equally relevant for both the documents, as TFIDF scores for both the documents is the same.

□□□

Unit IV : Big Data Analytics

Q. 1 Write a short note on Pandas. **(4 Marks)**

Ans. :

- Pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with "relational" or "labelled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python.
- Pandas is well suited for many different kinds of data such as the following.
  - Tabular data with heterogeneously-typed columns, as in an SQL table or Excel spreadsheet
  - Ordered and unordered (not necessarily fixed-frequency) time series data
  - Arbitrary matrix data (homogeneously typed or heterogeneous) with row and column labels
  - Any other form of observational / statistical data sets. The data need not be labelled at all to be placed into a pandas data structure
- The two primary data structures of pandas are as following.
  1. Series (1-dimensional) 2. DataFrame (2-dimensional)
- In pandas, a DataFrame is 2-dimensional.
- Each column in a DataFrame is a Series.

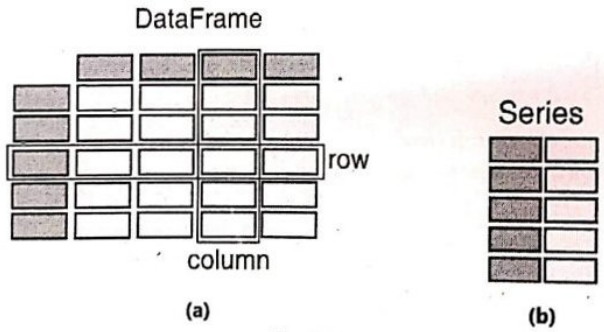


Fig. 4.1

- Both are suitable for handling the vast majority of typical use cases in finance, statistics, social science, and many areas of engineering.
- Pandas also carries out additional functions such as the following.
- Easy handling of missing data (represented as NaN) in floating point as well as non-floating point data
- Size mutability: columns can be inserted and deleted from DataFrame and higher dimensional objects
- Automatic and explicit data alignment: objects can be explicitly aligned to a set of labels, or the user can simply ignore the labels and let Series, DataFrame, etc. automatically align the data for you in computations
- Powerful, flexible group by functionality to perform split-apply-combine operations on data sets, for both aggregating and transforming data
- Make it easy to convert ragged, differently-indexed data in other Python and NumPy data structures into DataFrame objects
- Intelligent label-based slicing, fancy indexing, and subsetting of large data sets

- Intuitive merging and joining data sets
- Flexible reshaping and pivoting of data sets
- Hierarchical labelling of axes (possible to have multiple labels per tick)
- Robust IO tools for loading data from flat files (CSV and delimited), Excel files, databases, and saving / loading data from the ultrafast HDF5 format
- Time series-specific functionality: date range generation and frequency conversion, moving window statistics, date shifting, and lagging.

Q. 2 Write a program for storing data in Pandas DataFrame. Assume suitable data. **(4 Marks)**

Ans. :

```
import pandas as pd
df = pd.DataFrame(
  {
    "Name": ["Ajay","Sunil","Ravi","Pinky","Raju","Shiva","Raj","Rinku","Ramesh","Adam"],
    "Maths": [50,60,30,80,45,43,23,56,78,98],
    "English": [45,55,67,88,65,59,80,92,41,64],
    "Science": [65,54,96,94,93,85,84,39,44,40],
  }
)
print(df)
```

This would print the following.

	Name	Maths	English	Science
0	Ajay	50	45	65
1	Sunil	60	55	54
2	Ravi	30	67	96
3	Pinky	80	88	94
4	Raju	45	65	93
5	Shiva	43	59	85
6	Raj	23	80	84
7	Rinku	56	92	39
8	Ramesh	78	41	44
9	Adam	98	64	40

Q. 3 Write a program to fill missing values in a Pandas DataFrame with zeroes. Assume suitable data. **(4 Marks)**

Ans. : The data set that you would use would likely have missing values. Missing values are a ubiquitous problem in data wrangling, yet many underestimate the difficulty of working with missing data. pandas uses NumPy's NaN ("Not A Number") value to denote missing values. There are several techniques for handling missing values such as the following.

- Deleting the row
- Deleting the column
- Fill with the median value (Median imputation)
- Fill with the mean value (Mean imputation)

- Fill with the majority value
- Fill with 0
- Fill with a particular value

Assume that you have a DataFrame as following.

```
import pandas as pd
df = pd.DataFrame()
df["Name"] = ["Ajay","Sunil","Ravi","Pinky","Raju","Shiva","Raj","Rinku","Ramesh","Adam"]
df["Maths"] = [50,60,30,80,45,43,23,56,None,None]
print(df)
```

If you print this DataFrame, you will get the following.

```
    Name    Maths
0   Ajay    50.0
1   Sunil   60.0
2   Ravi    30.0
3   Pinky   80.0
4   Raju    45.0
5   Shiva   43.0
6   Raj     23.0
7   Rinku   56.0
8   Ramesh  NaN
9   Adam    NaN
```

The last two values are missing. You could use **`fillna()`** function to fill the missing values with your choice such as 0, mean value, min value, max value, linear interpolation, etc., as following.

```
import pandas as pd
df = pd.DataFrame()
df["Name"] = ["Ajay","Sunil","Ravi","Pinky","Raju","Shiva","Raj","Rinku","Ramesh","Adam"]
#Fill Missing Values with 0
df["Maths"] = [50,60,30,80,45,43,23,56,None,None]
df["Maths"] = df["Maths"].fillna(0)
print("\nMissing values filled with 0\n",df)
#Fill Missing vales with Mean
df["Maths"] = [50,60,30,80,45,43,23,56,None,None]
df["Maths"] = df["Maths"].fillna(df["Maths"].mean())
print("\nMissing values filled with Mean\n",df)
#Fill Missing vales with Min
df["Maths"] = [50,60,30,80,45,43,23,56,None,None]
df["Maths"] = df["Maths"].fillna(df["Maths"].min())
print("\nMissing values filled with Min\n",df)
#Fill Missing vales with Max
df["Maths"] = [50,60,30,80,45,43,23,56,None,None]
df["Maths"] = df["Maths"].fillna(df["Maths"].max())
```

```
print("\nMissing values filled with Max\n",df)
#Fill Missing vales with Linear Interpolation
df["Maths"] = [50,60,30,80,45,43,23,56,None,None]
df["Maths"] = df["Maths"].fillna(df["Maths"].interpolate())
print("\nMissing values filled with Linear Interpolation\n",df)
```

You will get the following.

Missing values filled with 0

	Name	Maths
0	Ajay	50.0
1	Sunil	60.0
2	Ravi	30.0
3	Pinky	80.0
4	Raju	45.0
5	Shiva	43.0
6	Raj	23.0
7	Rinku	56.0
8	Ramesh	0.0
9	Adam	0.0

Missing values filled with Mean

	Name	Maths
0	Ajay	50.000
1	Sunil	60.000
2	Ravi	30.000
3	Pinky	80.000
4	Raju	45.000
5	Shiva	43.000
6	Raj	23.000
7	Rinku	56.000
8	Ramesh	48.375
9	Adam	48.375

Missing values filled with Min

	Name	Maths
0	Ajay	50.0
1	Sunil	60.0
2	Ravi	30.0
3	Pinky	80.0
4	Raju	45.0
5	Shiva	43.0
6	Raj	23.0
7	Rinku	56.0
8	Ramesh	23.0
9	Adam	23.0

Missing values filled with Max

	Name	Maths
0	Ajay	50.0
1	Sunil	60.0
2	Ravi	30.0
3	Pinky	80.0
4	Raju	45.0
5	Shiva	43.0
6	Raj	23.0
7	Rinku	56.0
8	Ramesh	80.0
9	Adam	80.0

Missing values filled with Linear Interpolation

	Name	Maths
0	Ajay	50.0
1	Sunil	60.0
2	Ravi	30.0
3	Pinky	80.0
4	Raju	45.0
5	Shiva	43.0
6	Raj	23.0
7	Rinku	56.0
8	Ramesh	56.0
9	Adam	56.0

Q. 4 Explain a few data quality issues. **(6 Marks)**

Ans. : Common Data Quality Issues

- Data that is fit for use : For example, if you are working on a cancer project, you would need a dataset that has cancer patients and their health details.
- Data that meets your analytics requirements : For example, if you are trying to relate consumption of meat with probability of having cancer, you would need eating habits of the patients in the dataset.
- Relevance and timeliness : Take an example where you are building an analytics model on modern lifestyle. Could you use a dataset from 18th century? Perhaps not, right? Things and surroundings at that time were quite different than what they are in 21st century. So, the dataset that you pick must be relevant from timeliness or freshness perspective.
- Completeness, correctness, and formatting of data: For example, you would require that the important fields in the data set are fully populated and there are as few rows as possible that have missing values for particular fields. After eliminating such incomplete rows of data, are you left with enough data that you can use for both developing and testing your model?
- There could be other types of errors (or mix ups) as well in the data, such as the following, that require handling or data cleaning before you can use the data for training your machine learning model.

- **Spelling mistakes :** It is common to find spelling mistakes in names of countries, people, things, etc. There could also be abbreviations such as US, USA, United States of America, America - they all refer to the same country!
- **Date formatting :** Asian users typically use dd-mm-yyyy date format whereas American users could use mm-dd-yyyy format.
- **Incorrect labels :** For example, age could be labelled as year born. So, if age column is incorrectly labelled as year born, then age of 56 could be mistakenly assumed to be born in the year 1956.
- **Scaling and units :** Sometimes the units could be wrongly entered. For example, weight of the person could be entered in Kgs or Pounds. Rows having incorrect units could skew the analysis.
- **Skewed data (data anomalies) :** For a given field in the dataset, some of the values could be quite high and some of the values could be quite low. Such skewed data could wrongly train the model.

Q. 5 What could you do to fix the poor quality data? **(4 Marks)**

Ans. :

Remediating (Fixing) Data Quality Issues

Data cleaning is one of the most time consuming exercise in building a machine learning model. Some of the common measures to clean the data are as following.

1. Delete rows with missing values.
2. Fix any formatting issues.
3. Fix labelling issues.
4. Fix spelling mistakes and abbreviations.
5. Insert new columns based on other columns.
6. Delete rows with skewed values.

Q. 6 List a few data quality metrics used in defining data quality constraints. **(4 Marks)**

Ans. : Some of the common data quality metrics used in defining data quality constraints are as shown in Table 4.1

Table 4.1

Metric	Description	Usage Example
ApproxCountDistinct	Approximate number of distinct value, computed with HyperLogLogPlusPlus sketches.	ApproxCountDistinct("review_id")
ApproxQuantile	Approximate quantile of a distribution.	ApproxQuantile("star_rating", quantile = 0.5)
ApproxQuantiles	Approximate quantiles of a distribution.	ApproxQuantiles("star_rating", quantiles = Seq(0.1, 0.5, 0.9))
Completeness	Fraction of non-null values in a column.	Completeness("review_id")
Compliance	Fraction of rows that comply with the given column constraint.	Compliance("top star_rating", "star_rating >= 4.0")
Correlation	Pearson correlation coefficient measures the linear correlation between two columns. The result is in the range [-1, 1], where 1 means positive linear correlation, -1 means negative linear correlation, and 0 means no correlation.	Correlation("total_votes", "star_rating")
CountDistinct	Number of distinct values.	CountDistinct("review_id")
DataType	Distribution of data types such as Boolean, Fractional, Integral, and String. The resulting histogram allows filtering by relative or absolute fractions.	DataType("year")

Metric	Description	Usage Example
Distinctness	Fraction of distinct values of a column over the number of all values of a column. Distinct values occur at least once. Example: [a, a, b] contains two distinct values a and b, so distinctness is 2/3.	Distinctness("review_id")
Entropy	Entropy is a measure of the level of information contained in an event (value in a column) when considering all possible events (values in a column). It is measured in nats (natural units of information). Entropy is estimated using observed value counts as the negative sum of (value_count/total_count) * log(value_count/total_count). Example: [a, b, b, c, c] has three distinct values with counts [1, 2, 2]. Entropy is then (-1/5*log(1/5)-2/5*log(2/5)-2/5*log(2/5)) = 1.055.	Entropy("star_rating")
Maximum	Maximum value.	Maximum("star_rating")
Mean	Mean value; null values are excluded.	Mean("star_rating")
Minimum	Minimum value.	Minimum("star_rating")
MutualInformation	Mutual information describes how much information about one column (one random variable) can be inferred from another column (another random variable). If the two columns are independent, mutual information is zero. If one column is a function of the other column, mutual information is the entropy of the column. Mutual information is symmetric and nonnegative.	MutualInformation(Seq("total_votes", "star_rating"))
PatternMatch	Fraction of rows that comply with a given regular expression.	PatternMatch("marketplace", pattern = raw"\w{2}".r)
Size	Number of rows in a DataFrame.	Size()
Sum	Sum of all values of a column.	Sum("total_votes")
UniqueValueRatio	Fraction of unique values over the number of all distinct values of a column. Unique values occur exactly once; distinct values occur at least once. Example: [a, a, b] contains one unique value b, and two distinct values a and b, so the unique value ratio is 1/2.	UniqueValueRatio("star_rating")
Uniqueness	Fraction of unique values over the number of all values of a column. Unique values occur exactly once. Example: [a, a, b] contains one unique value b, so uniqueness is 1/3.	Uniqueness("star_rating")

Q. 7 What is data wrangling? Why do you need it? **(4 Marks)**

Ans. :

- **Definition :** Data wrangling is the process of cleaning and unifying messy and complex data sets to make them more appropriate and valuable for a variety of downstream purposes such as analytics
- Data wrangling is also called as data munging or data pre-processing. Before you proceed, let's take a step back and look at a few concepts.

Need for Data Wrangling

1. **Data**

- The raw data, or just data is a collection of observations of real-world phenomena. For instance, stock market data might involve observations of daily stock prices, announcements of earnings by individual companies, and even opinion articles from pundits.
- Sports data could have information on matches, environment in which those matches were played, player performances, and several other observations.
- Similarly, personal biometric data can include measurements of your minute-by-minute heart rate, blood sugar level, blood pressure, oxygen level, etc. You can come up with endless examples of data across different domains.
- Each piece of data provides a small window into a limited aspect of reality. The collection of all of these observations gives you a picture of the whole. But the picture is messy because it is composed of a thousand little pieces, and there's always measurement noise and missing pieces.

2. **Tasks**

Why do you collect data? Some of the popular questions are :

- How likely is that a customer buying product A will also buy product B?
- Which team is likely to win?
- How will be the weather next month?
- What food you should eat to get healthier?
- What is the risk of getting diabetes based on your biometric data?
- The path from data to answers is full of false starts and dead ends.

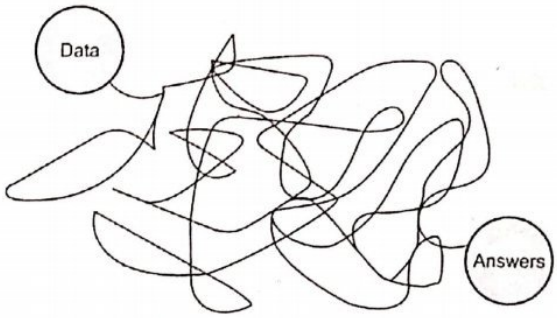


Fig. 4.2

- What starts out as a promising approach may not work in reality. What was originally just a hunch may end up leading to the best solution. Workflows with data are frequently multistage and iterative processes. For instance, stock prices are observed at the exchange, aggregated by an intermediary like Thomson Reuters, stored in a database, bought by a company, converted into a Hive store on a Hadoop cluster, pulled out of the store by a script, subsampled, massaged, and cleaned by another script, dumped to a file, and converted to a format that you can try out in your favourite modelling library in R, Python, or Scala.
- The predictions are then dumped back out to a CSV file and parsed by an evaluator, and the model is iterated multiple times, rewritten in C++ or Java by your production team, and run on all of the data before the final predictions are pumped out to another database.

- However, if you disregard the mess of tools and systems for a moment, you might see that the process involves two mathematical entities that are at the centre of machine learning models and features.

3. Models

- Trying to understand the world through data is like trying to piece together reality using a noisy, incomplete jigsaw puzzle with a bunch of extra pieces. This is where mathematical modelling in particular statistical modelling comes in. The language of statistics contains concepts for many frequent characteristics of data, such as wrong, redundant, or missing. Wrong data is the result of a mistake in measurement.
- Redundant data contains multiple aspects that convey exactly the same information. For instance, the day of week may be present as a categorical variable with values of "Monday," "Tuesday," ... "Sunday," and again included as an integer value between 0 and 6. If this day-of-week information is not present for some data points, then you have got missing data on your hands.
- A mathematical model of data describes the relationships between different aspects of the data. For instance, a model that predicts stock prices might be a formula that maps a company's earning history, past stock prices, and industry to the predicted stock price. A model that recommends music might measure the similarity between users (based on their listening habits) and recommend the same artists to users who have listened to a lot of the same songs.
- Mathematical formulas relate numeric quantities to each other. But raw data is often not numeric. For example, the action "Rohit bought Motorola G9 on Friday", is not numeric. Similarly, product reviews may not be numeric. This is where features come in.

4. Features

- Feature is anything that you can measure and build data for. For example, the typical length of various animals. Feature could be numeric, set of characters, Boolean values, or anything else that describes the data.
- But, for most machine learning mathematic models, features are required to be numeric so that they can be used in various computation.
- Redefine features as,
- **Definition :** A feature is a numeric representation of raw data.

Q. 8 What is feature engineering? **(4 Marks)**

OR With a flow-chart, explain the high-level process of feature engineering. **(6 Marks)**

OR What does feature engineering typically include? **(4 Marks)**

OR Write a short note on feature engineering. **(4 Marks)**

Ans. :

Feature Engineering

- There are many ways to turn raw data into numeric measurements (I just showed you one earlier), which is why features can end up looking like a lot of things.
- Naturally, features must derive from the type of data that is available. Features are also tied to the model. Some models are more appropriate for some types of features, and vice versa.
- The right features are relevant to the task at hand and should be easy for the model to ingest.
- **Definition :** Feature engineering is the process of formulating the most appropriate features given the data, the model, and the task.
- The Fig. 4.3(a) depicts where feature engineering sits in the machine learning pipeline.

28

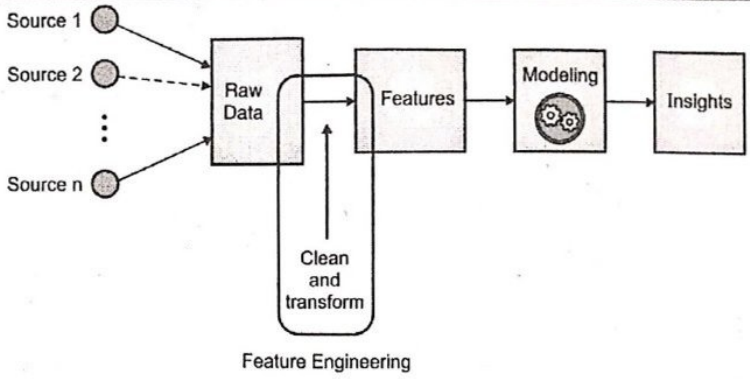


Fig. 4.3(a)

- Features and models sit between raw data and the desired insights. In a machine learning workflow, you pick not only the model, but also the features.
- This is a double-jointed lever, and the choice of one affects the other. Good features make the subsequent modelling step easy and the resulting model more capable of completing the desired task.
- Bad features may require a much more complicated model to achieve the same level of performance.
- The number of features is also important. If there are not enough informative features, then the model will be unable to perform the ultimate task.
- If there are too many features, or if most of them are irrelevant, then the model will be more expensive and trickier to train. Something might go wrong in the training process that impacts the model's performance.
  - Feature engineering typically includes feature creation, feature transformation, feature extraction, and feature selection.

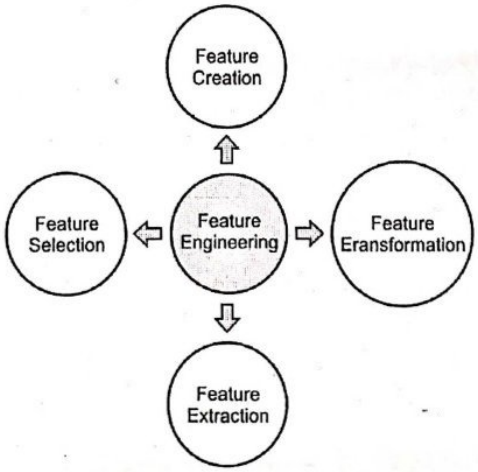


Fig. 4.3(b)

  - Feature creation identifies the features in the dataset that are relevant to the problem at hand.
  - Feature transformation manages replacing missing features or features that are not valid.
  - Feature extraction is the process of creating new features from existing features, typically with the goal of reducing the dimensionality of the features.

- Feature selection is the filtering of irrelevant or redundant features from your dataset. This is usually done by observing variance or correlation thresholds to determine which features to remove.

- At a high-level, the feature engineering process looks like shown in Fig. 4.3(c).

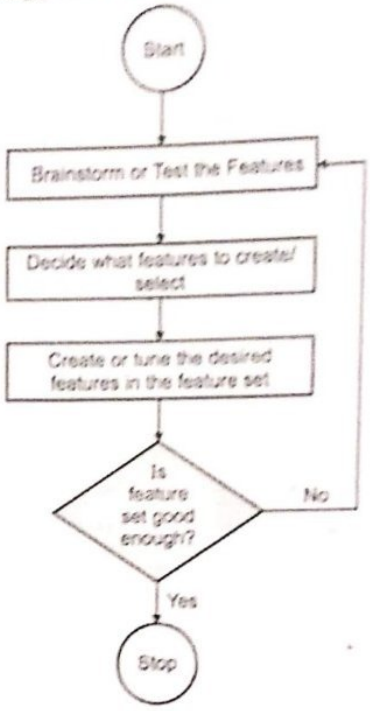


Fig. 4.3(c)

Q. 9 Explain data wrangling methods. (6 Marks)

OR One method is good enough for data wrangling. Comment. (4 Marks)

OR With an example, explain dimensionality reduction. (6 Marks)

OR With a suitable example , explain binning. (6 Marks)

OR With a suitable example, explain quantization. (6 Marks)

OR Explain log transform. (6 Marks)

Ans. : There are several data pre-processing (wrangling) techniques as following.

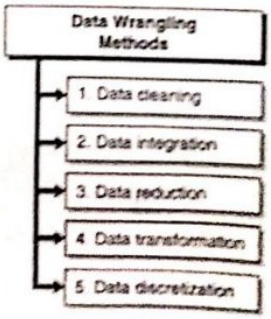


Fig. 4.4(a)

1. **Data Cleaning**
   - Data cleaning can be applied to remove noise and correct inconsistencies in data. Data cleaning routines work to "clean" the data by filling in missing values, smoothing noisy data, identifying or removing outliers, and resolving inconsistencies. If the data is dirty, then it could lead to inaccurate results. For example, the day Monday could be represented as "Mon", "M", "1", "Monday".
   - You need to fix such inconsistencies before you proceed with using your data. Another vary common example of inconsistent data could be country names. You could find USA represented as "America", "The US", "US", or "United States of America". Such inconsistencies need to be fixed.
2. **Data Integration**
   - Data integration merges data from multiple sources into a coherent data store such as a data warehouse. However, data integration could also lead to inconsistencies. For example, the attribute for customer identification may be referred to as customer_id in one data store and cust_id in another. Naming inconsistencies may also occur for attribute values. For example, the same first name could be registered as "Bill" in one database, "William" in another, and "B." in a third.
   - Furthermore, you suspect that some attributes may be inferred from others (e.g., annual revenue). Having a large amount of redundant data may slow down or confuse the knowledge discovery process. Clearly, in addition to data cleaning, steps must be taken to help avoid redundancies during data integration. Typically, data cleaning and data integration are performed as a pre-processing step when preparing data for a data warehouse. Additional data cleaning can be performed to detect and remove redundancies that may have resulted from data integration.
3. **Data Reduction**
   - Data reduction can reduce data size by aggregating, eliminating redundant features, or clustering. Data reduction obtains a reduced representation of the data set that is much smaller in volume yet produces the same (or almost the same) analytical results. Data reduction strategies include dimensionality reduction and numerosity reduction.
     - In dimensionality reduction, data encoding schemes are applied so as to obtain a reduced or "compressed" representation of the original data.
     - In numerosity reduction, the data are replaced by alternative, smaller representations using parametric models (e.g., regression or log-linear models) or nonparametric models (e.g., histograms, clusters, sampling, or data aggregation).
   - **Example :** Assume that you are watching cricket on your 4K TV. A 4K TV has 3,840 horizontal pixels and 2,160 vertical pixels, for a total of about 8.3 million pixels! Most of the times, you are focusing on the ball. Say that the ball occupies 10,000 pixels to form a 3D image. Your brain is filtering out rest of the pixels and helping you to focus on 10,000 pixels out of the total 8.3 million pixels presented to it. That is close to just 0.12% of the entire set of pixels that is in front of you. This is precisely what happens in dimensional reduction. You reduce the number of dimensions in your dataset to just what matters the most.
   - Often times, you find that your dataset could have 100s of features (or dimensions). Practically, you know that not all dimensions are equally important for analysis or classification of data. Also, it becomes computationally intensive and visually difficult to understand which dimensions have the most influence on the dataset if you have 100s of dimensions.
   - **Definition :** Dimensionality reduction techniques help you to reduce the number of dimensions to only keep important dimensions of data and discard all other dimensions.

- In most learning algorithms, the complexity depends on the number of input dimensions, d, as well as on the size of the data sample, N. As you increase the number of dimensions, you would also require collecting increasing number of samples to support those many dimensions (in order to ensure that every combination of features is well represented in the dataset). As the number of dimensions increase, working with it becomes increasingly harder. This problem is often cited as "the curse of dimensionality".
- To make it less computationally intensive, you reduce the dimensionality of the problem. Decreasing d also decreases the overall complexity of the problem and make it more plausible to understand most important dimensions of data.

Fig. 4.4(b)

- **Example :**

Income	Credit Score	Age	Location	Give Loan
50,000	High	34	Mumbai	Yes
75,000	Low	33	Bangalore	No
80,000	High	37	Mumbai	Yes
90,000	High	29	Kolkata	Yes

- This dataset has 4 dimensions (Income, Credit Score, Age, and Location) based on which loan approval seems to be granted. But, if you look closely, then you would find that the other dimensions do not influence the decision as much as Credit Score dimension. So, the same dataset with reduced dimensions could be as following.

Credit Score	Give Loan
High	Yes
Low	No
High	Yes
High	Yes

- This dimensional reduction not only makes the algorithms computationally less intensive but also makes it simple to understand and visualise the dataset as well as the results.

4. Data Transformation

- Data transformations (e.g., normalisation) may be applied, where data are scaled to fall within a smaller range like 0.0 to 1.0. This can improve the accuracy and efficiency of modelling algorithms involving distance measurements. For example, your customer data may contain the attributes "age" and "annual salary". The "annual salary" attribute usually takes much larger values than age.
- Therefore, if the attributes are left unnormalized, the distance measurements taken on "annual salary" will generally outweigh distance measurements taken on age. One such common transformation technique is log transform.

Log Transform

- The log transform is a powerful tool for dealing with large positive numbers with a heavy-tailed distribution. A heavy-tailed distribution places more entries towards the tail end of the plot rather than centre. It compresses the long tail in the high end of the distribution into a shorter tail and expands the low end into a longer head. Let's understand how.
- The log function is the inverse of the exponential function. It is defined such that $\log_a(a^x) = x$, where a is a positive constant, and x can be any positive number. Since $a^0 = 1$, you get $\log_a(1) = 0$. This means that the log function maps the small range of numbers between (0, 1) to the entire range of negative numbers $(-\infty, 0)$. The function $\log_{10}(x)$ maps the range of [1, 10] to [0, 1], [10, 100] to [1, 2], and so on. In other words, the log function compresses the range of large numbers and expands the range of small numbers. The larger x is, the slower log (x) increments.
- For example, in the Fig. 4.4(c), note how the horizontal x values from 100 to 1,000 get compressed into just 2.0 to 3.0 in the vertical y range, while the tiny horizontal portion of x values less than 100 are mapped to the rest of the vertical range.

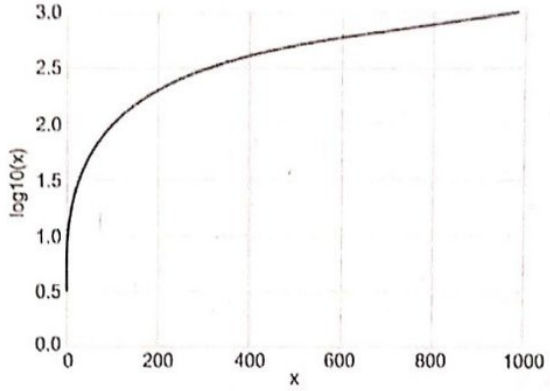


Fig. 4.4(c)

- Log transform is commonly used when you are dealing with large numbers. For example, some businesses have a lot of reviews (say over 2000) and some have only few (say in 20s).
- In such a scenario, it becomes difficult to compare and correlate one business with the other because a large count in one element of the data set would outweigh the similarity in all other elements, which could throw off the entire similarity measurement for various machine learning algorithms. Log transformation helps to normalise skewed data.

5. Data Discretisation

- Data discretisation and concept hierarchy generation can also be useful, where raw data values for attributes are replaced by ranges or higher conceptual levels. For example, raw values for age may be replaced by higher-level concepts, such as youth, adult, or senior.
- One common data discretisation technique is quantization or binning.

6. Quantization or Binning

- Quantization or binning is a feature construction technique where you could combine features to create segments or bins of information. In other words, you group the counts into bins, and get rid of the actual count values.

- For example, consider the following data set for real-estate sites.

Site Length	Site Breadth	Site Price
30	40	40 Lakhs
40	32	40 Lakhs
30	30	30 Lakhs
35	45	45 Lakhs
40	60	60 Lakhs
60	80	90 Lakhs

- Instead of having site length and site breadth that are not very useful for establishing a modelling pattern, you could create a new feature such as "site area". The "site area" feature could then provide a good estimate of site price.

Site Length	Site Breadth	Site Area	Site Price
30	40	1200	40 Lakhs
40	32	1280	40 Lakhs
30	30	900	30 Lakhs
35	45	1575	45 Lakhs
40	60	2400	60 Lakhs
60	80	4800	90 Lakhs

- There could be several other examples of binning. For example, it is common to see custom-designed age ranges that better correspond to stages of life, such as:
  - 0-12 years old
  - 12-17 years old
  - 18-24 years old
  - 25-34 years old
  - 35-44 years old
  - 45-54 years old
  - 55-64 years old
  - 65-74 years old
  - 75 years or older
- You could get rid of the actual age in a large data set and create bins based on age groups. You could then model things such as product or service preferences, reviews, demographics, eating habits, etc. more elegantly.
- It is up to your requirements to create bins as you need. For example, you may need to create bins for income. You could then have something like the following (remember how income tax or electricity bills have various slabs?).
  - Below 5 Lakhs
  - 5-10 lakhs
  - 10-20 Lakhs

34

- 20-50 Lakhs
- Over 50 Lakhs

- You can create bins on fixed value range or take value range based on other mathematical derivations such as quantile, percentile, median, etc.

BMI	Nutritional status
Below 18.5	Underweight
18.5 – 24.9	Normal weight
25.0 – 29.9	Pre-obesity
30.0 – 34.9	Obesity class I
35.0 – 39.9	Obesity class II
Above 40	Obesity class III

Q. 10 Explain the Hadoop Ecosystem in detail with Hive. **SPPU - Dec. 18, 2 Marks**

OR Explain term : Hive **SPPU - Dec. 19, 2 Marks**

OR Describe the characteristics and features of Hive. **(4 Mark)**

OR Explain the architecture of Hive. **(8 Marks)**

Ans. :

- **Definition :** The Apache Hive data warehouse software facilitates reading, writing, and managing large datasets residing in distributed storage using SQL.
- Hive provides a SQL like query language called HiveQL (Hive Query Language) to carry out data processing.
- Hive is designed to enable easy data summarisation, ad-hoc querying, and analysis of large volumes of data. Hive's SQL also gives users multiple places to integrate their own functionality to do custom analysis with User Defined Functions (UDFs).

(A) Characteristics and Features of Hive

1. **SQL like data model :** Hive data is organised into SQL like data model. A Hive table structure consists of rows and columns. The rows typically correspond to some record, transaction, or particular entity. The values of the corresponding columns represent the various attributes or characteristics for each row. Hive data model has the following data units.
   - **(a) Databases :** Databases provide namespaces function to avoid naming conflicts for tables, views, partitions, columns, and so on. Databases can also be used to enforce security for a user or group of users.
   - **(b) Tables :** Tables are homogeneous units of data and follow a relational kind of schema and structure.
   - **(c) Partitions :** Each Table can have one or more partition Keys which determines how the data is stored. Partitions allow the user to efficiently identify the rows that satisfy specified criteria.
   - **(d) Buckets (or Clusters) :** Data in each partition may in turn be divided into Buckets based on the values.
2. **SQL like operations :** Hive QL is very similar to the traditional SQL. It is a good choice if a user has experience with SQL and the data is already in HDFS. Hive does not have a steep learning curve for users who already know how to work with SQL queries.
3. **Not suitable for real-time querying :** Hive is not designed for transaction processing and real time querying. A Hive query is first translated into a MapReduce job, which is then submitted to the Hadoop

cluster for execution. Thus, the execution of the query has to compete for resources with any other submitted job.

4. **Support for many client applications :** Hive supports a variety of client application written in Java, PHP, Python, C++, or Ruby. You can also use these clients to access the data like a traditional database.

(B) Architecture of Hive

Fig. 4.5 shows a high-level architecture of Hive.

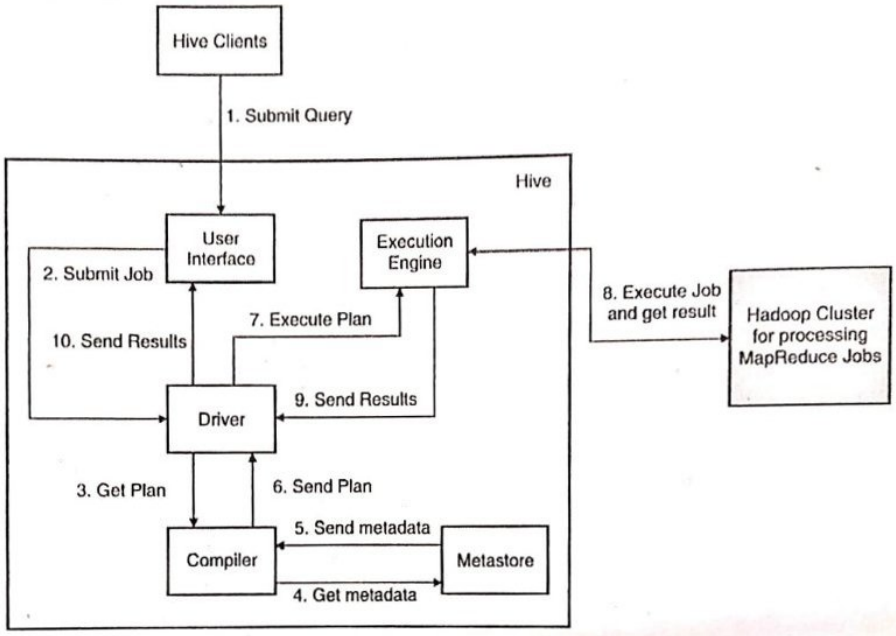


Fig. 4.5 : High-level architecture of Hive

The main components of Hive are as following.

1. **Hive Clients :** Apache Hive supports different types of client applications for performing queries on the Hive. There are three types of Hive clients supported – Thrift clients, JDBC clients and ODBC clients. You can choose amongst them as appropriate for your application.
2. **User Interface :** The user interface is for users to submit Hive queries and other operations to the system. As of now, Hive provides a command line interface (CLI) and a web based GUI.
3. **Driver :** It receives the queries from Hive clients via the User Interface. It passes on the query to the compiler to get the execution details (plan).
4. **Compiler :** It parses the query, does semantic analysis on the different query blocks and query expressions, and eventually generates an execution plan with the help of the table and partition metadata looked up from the Metastore.
5. **Metastore :** It stores all the structure information of the various tables and partitions in the warehouse including column and column type information, the serialisers and deserialisers necessary to read and write data and the corresponding HDFS files where the data is stored.
6. **Execution Engine :** It executes the execution plan created by the compiler on the Hadoop cluster. The plan is a DAG of stages. The execution engine manages the dependencies between these different stages of the plan and executes these stages on the appropriate system components.

36

Unit V : Big Data Visualization

Q. 1 What is data visualisation ? **SPPU - Dec. 18, 9 Marks, May 19, 8 Marks**

OR What is mean by Data conditioning and data visualisation ? **SPPU - Dec. 19, 5 Marks**

Ans. :

- It is one thing to process the massive amount of data and another to make it human friendly to be able to comprehend it, even from a distance, without getting into nitty-gritty of the complex calculations.

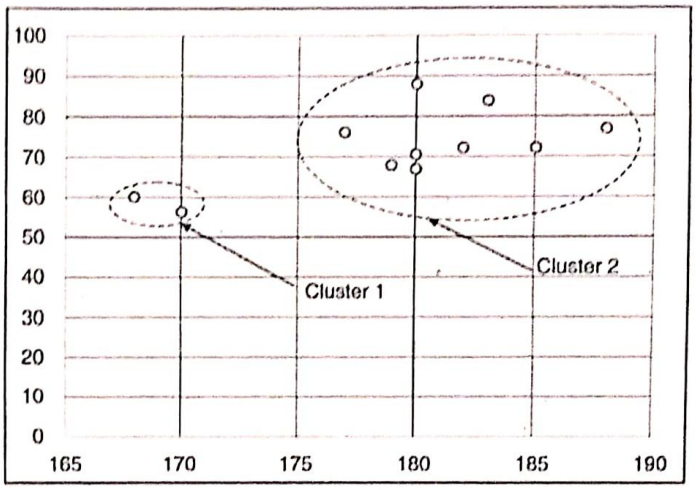


Fig. 5.1

Table 5.1

Sr. No.	Height	Weight
1.	185	72
2.	170	56
3.	168	60
4.	179	68
5.	182	72
6.	188	77
7.	180	71
8.	180	70
9.	183	84
10.	180	88
11.	180	67
12.	177	76

efinition : Data visualisation is a graphical or pictorial representation of data that makes it easy to mmunicate the information to humans.

ta visualisation uses various forms of representations to match the data and the relationship amongst its data ributes so as to communicate the desired information effectively.

Describe the goals of data visualisation. **(6 Marks)**

oals (Objectives) of Data Visualisation

5.2(a) shows some of the major goals or objectives of data visualisation.

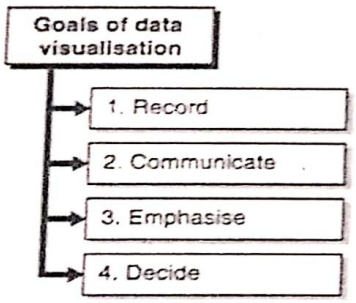


Fig. 5.2(a) : Goals (Objectives) of Data Visualisation

d : Data visualisation helps you to record the information that might be lying in various forms such as logs, emails, conversations, audio, video, or any other form of information sharing. A user may not have to ough several of these information sources to get to the data she desires. For example, a music player could e a visual representation of the musical notes being played on your phone and you could according tune s parameters such as bass and treble as you like it.

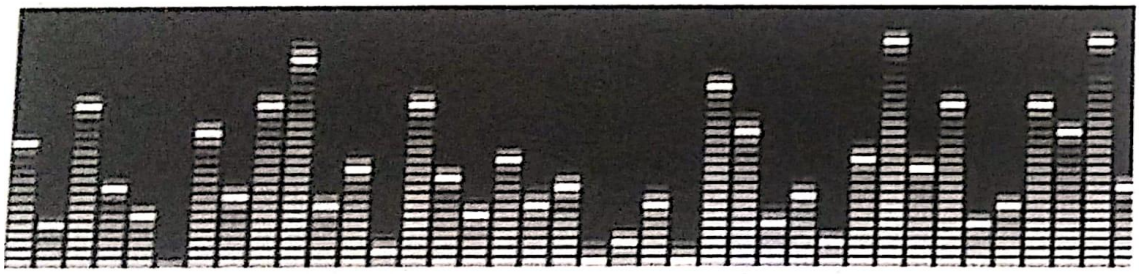


Fig. 5.2(b) : Examples of Record

nicate : A primary goal of data visualisation is to communicate the information in the most effective way iven data. There are several types of charts, maps and graphs that could be effectively used to visualise forms of data as well as relationship amongst its data attributes as suitable. For example, Microsoft® vides various types of charts for plotting your data.

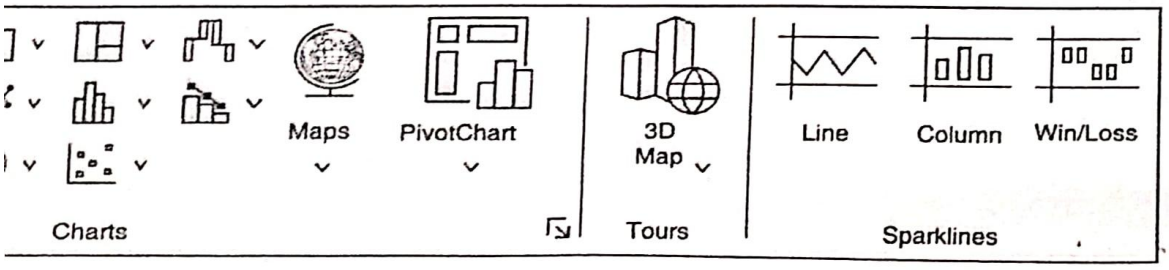


Fig. 5.2(c) : Various types of charts provided by Microsoft Excel

3. **Emphasise :** Using data visualisation, you can emphasise or highlight a portion of data, find patterns, show trends, or depict relationships between various data attributes. For example, the Fig. 6.2(d) gives a quick view of rainfall in various states of India in a particular year.

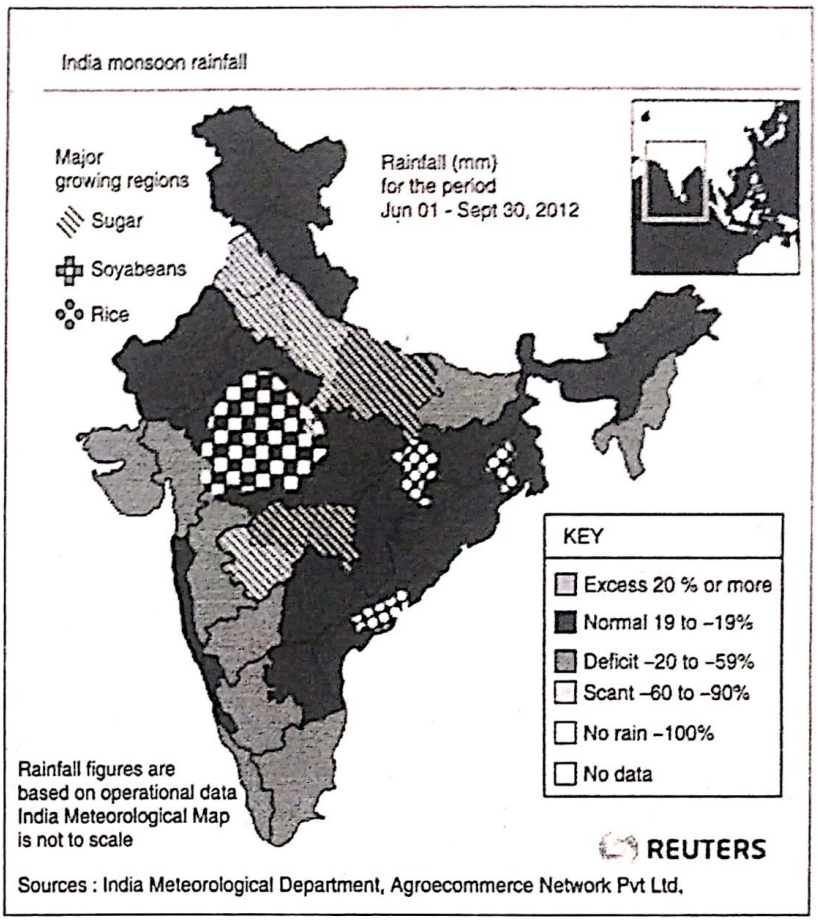


Fig. 5.2(d) : Example of a chart emphasising the data points

As shown in Fig. 5.2(d) if the country has received adequate rainfall or shortage of rainfall. Which states were most affected, and which were least affected. This representation is much easier to understand and explain than to go through 25+ entries for rainfall measurement for each state collected for each day of the rainy season.

4. **Decide :** Data visualisation makes it easier to quickly make decisions and take actions. You do not have to go through length reports and complex data to understand what needs to be done next. For example, from surveying 1,000 customers about the customer service, you might have detailed responses.

   But, representing the cause visually makes you to quickly decide the action plan that training is required for the customer service representative to improve customer satisfaction.

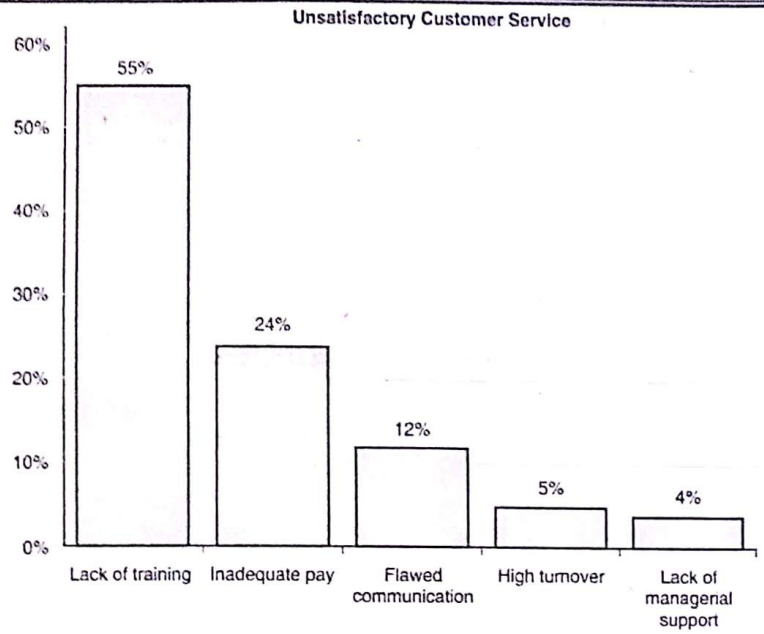


Fig. 5.2(e) : Example of a chart helping in decision making

Q. 3 What are the challenges in Big data visualization ? **SPPU - Dec. 18, 8 Marks**

OR Why it is difficult to visualize Big Data ? **SPPU - May 19, 9 Marks , Dec. 19, 8 Marks**

Ans. : Some of the major challenges or difficulties with visualising Big Data are as shown in Fig. 5.3.

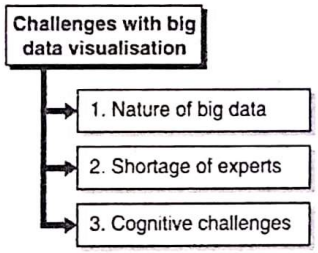


Fig. 5.3 : Challenges with Big Data Visualisation

1. **Nature of Big Data :** The Five Vs of Big Data – Volume, Variety, Velocity, Veracity and Value. The nature of Big Data itself is one of the biggest challenges of visualising it. The massive volume of data requires special software and hardware for handing the visualisation. The heterogeneity (variety) of data attributes makes it further hard to conceptualise the right forms of visualisation to show relationship between the data attributes. The velocity of data is so fast that you require to update your visualisation very frequently to keep it accurate. You may want to interact with it in the real time as well which makes it further complex to visualise it. The veracity and value characteristics require that your visualisation meets the data quality and usefulness as well. If the visualisation is not useful or was formed with poor quality data, it may not fulfill the desired objectives of the visualisation.
2. **Shortage of Experts :** Data analytics is an emerging field and there are not many experts around the world. You require experts who can
   (a) Understand the wide variety of data
   (b) Model the data correctly so as to meet the desired objectives

(c) Build and manage software and hardware tools and techniques required for Big Data processing
(d) Design appropriate visual interfaces and
(e) Also communicate the findings effectively

Building a team of experts that have all the required capabilities is challenging.

3. **Cognitive Challenges :** Finally, irrespective of what you have got, visualisation is something that a human needs to understand and make sense about. Overloaded charts with full of various colours and gauges make it difficult to get the real sense of data. Also, plots like regression line, ROC curve etc. are difficult to understand and make sense about if you do not know how to read and interpret them.

Q. 4 What are the challenges in Big data visualization ? **SPPU - Dec. 18, 8 Marks**

OR Why it is difficult to visualize Big Data ? **SPPU - May 19, 9 Marks , Dec. 19, 8 Marks**

Ans. : Some of the major challenges or difficulties with visualising Big Data are as shown in Fig. 5.4.

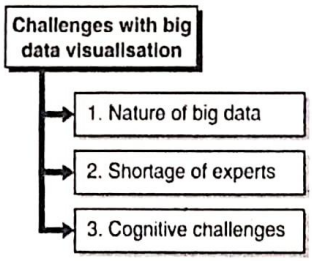


Fig. 5.4 : Challenges with Big Data Visualisation

1. **Nature of Big Data :** The Five Vs of Big Data – Volume, Variety, Velocity, Veracity and Value. The nature of Big Data itself is one of the biggest challenges of visualising it. The massive volume of data requires special software and hardware for handing the visualisation. The heterogeneity (variety) of data attributes makes it further hard to conceptualise the right forms of visualisation to show relationship between the data attributes. The velocity of data is so fast that you require to update your visualisation very frequently to keep it accurate. You may want to interact with it in the real time as well which makes it further complex to visualise it. The veracity and value characteristics require that your visualisation meets the data quality and usefulness as well. If the visualisation is not useful or was formed with poor quality data, it may not fulfill the desired objectives of the visualisation.
2. **Shortage of Experts :** Data analytics is an emerging field and there are not many experts around the world. You require experts who can
   (a) Understand the wide variety of data
   (b) Model the data correctly so as to meet the desired objectives
   (c) Build and manage software and hardware tools and techniques required for Big Data processing
   (d) Design appropriate visual interfaces and
   (e) Also communicate the findings effectively

   Building a team of experts that have all the required capabilities is challenging.
3. **Cognitive Challenges :** Finally, irrespective of what you have got, visualisation is something that a human needs to understand and make sense about. Overloaded charts with full of various colours and gauges make it difficult to get the real sense of data. Also, plots like regression line, ROC curve etc. are difficult to understand and make sense about if you do not know how to read and interpret them.

niques. SPPU - Dec 18, 9 Mark, May 19, 8 Marks

(4 Marks)

gram and explain its usage. (8 Marks)

Cloud and explain its usage. (8 Marks)

ations could be grouped as shown in Fig. 5.5(a).

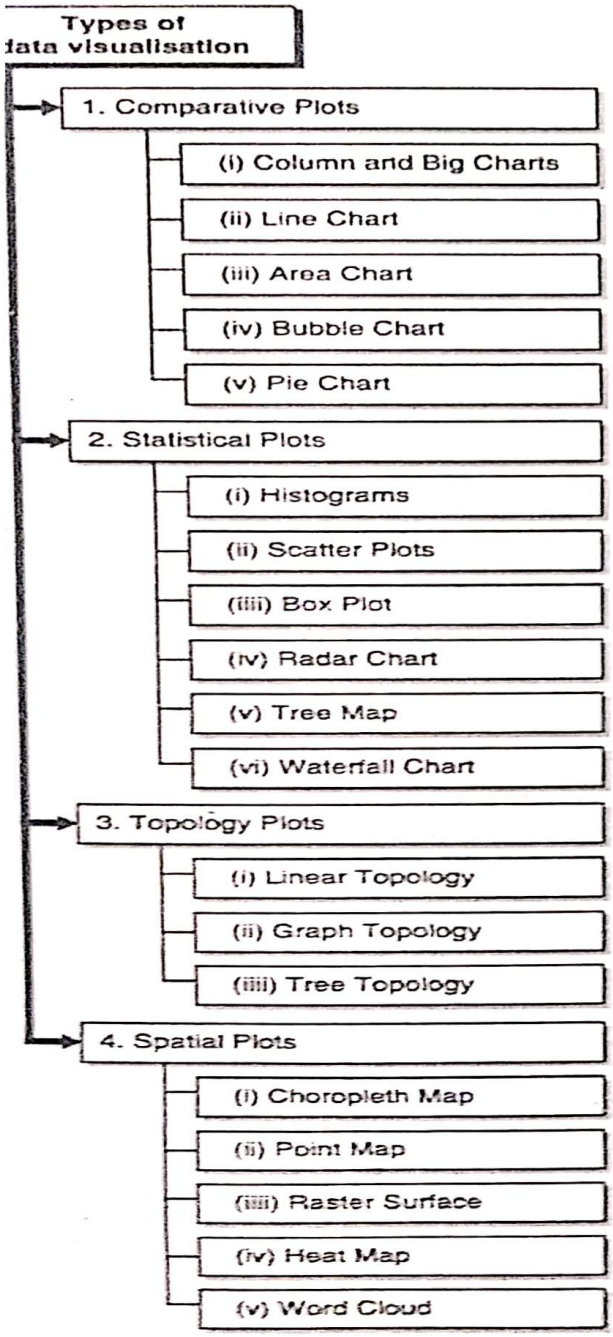


Fig. 5.5(a) : Types of Data Visualisation

(A) Comparative Plots

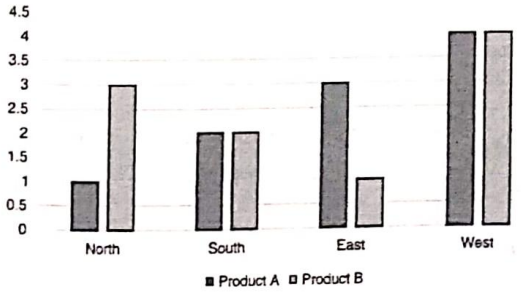
Comparative plots are used for comparing the datapoints. Some of the commonly used comparative plots.

1. Column and Bar Charts

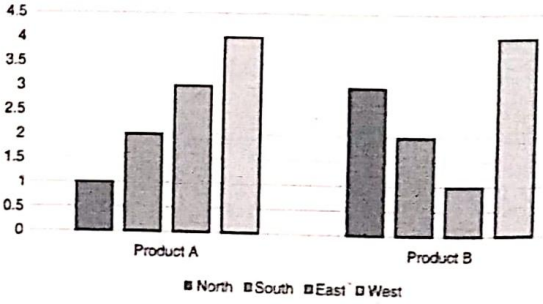
- Column and Bar charts are perhaps the most common, most simple, and most popular chart that you might have ever seen. It is used for comparing two or more values in the same category.
- There could be several variations of the chart such as column chart, bar chart, stack chart and their 2D and 3D plots. Some of the column and bar charts are as following for the following data shown in Table 5.2(a).

Table 5.2(a)

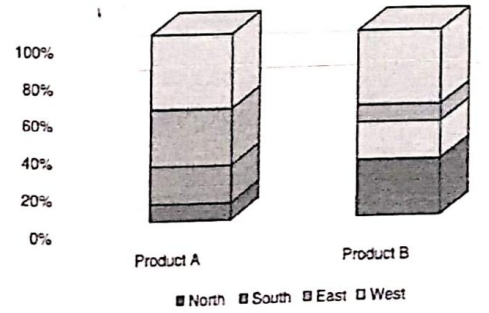
Sales Region	Product A	Product B
North	1	3
South	2	2
East	3	1
West	4	4



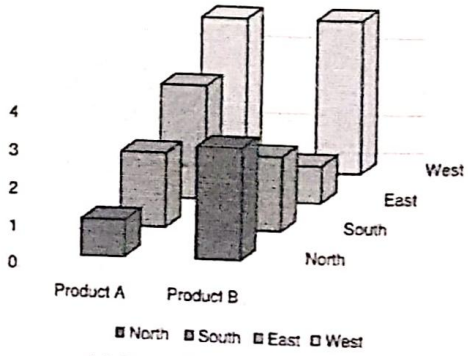
(b) Example of a column chart



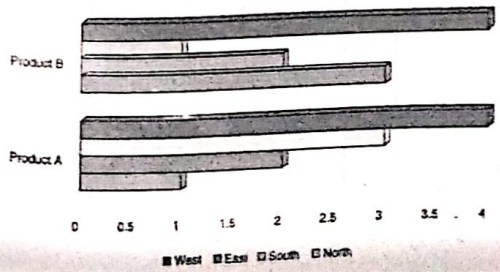
(c) Example of a column chart



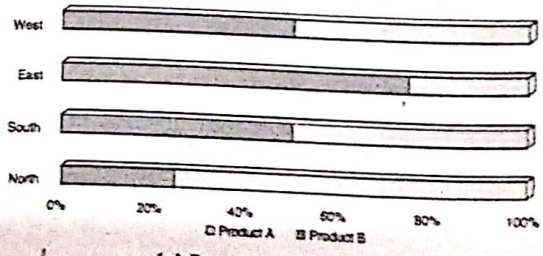
(d) Example of a stacked column chart



(e) Example of a 3D column chart



(f) Example of a 3D bar chart



(g) Example of a 3D bar chart

Fig. 5.5

- Column and Bar charts are easy to read and understand and individual datapoints can be changed without affecting others. Column and Bar charts do not work well if there are several categories.

2. Line Chart

- Line chart is normally used to show time-series data and could also be used to compare categories. Line chart is used to understand trend and patterns in your data and also make future projections.

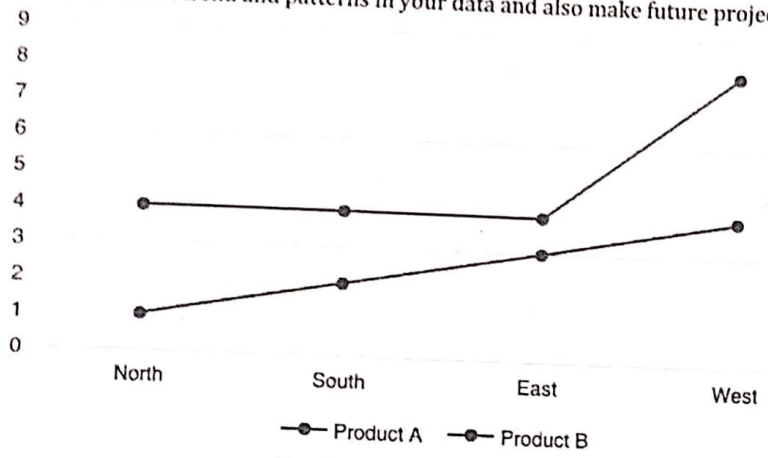


Fig. 5.5(h) : Line charts

- There could also be minor variations of the line chart. For example, the Fig. 5.5(i) is for SENSEX. It adds a spark line to highlight the market trends in a day.

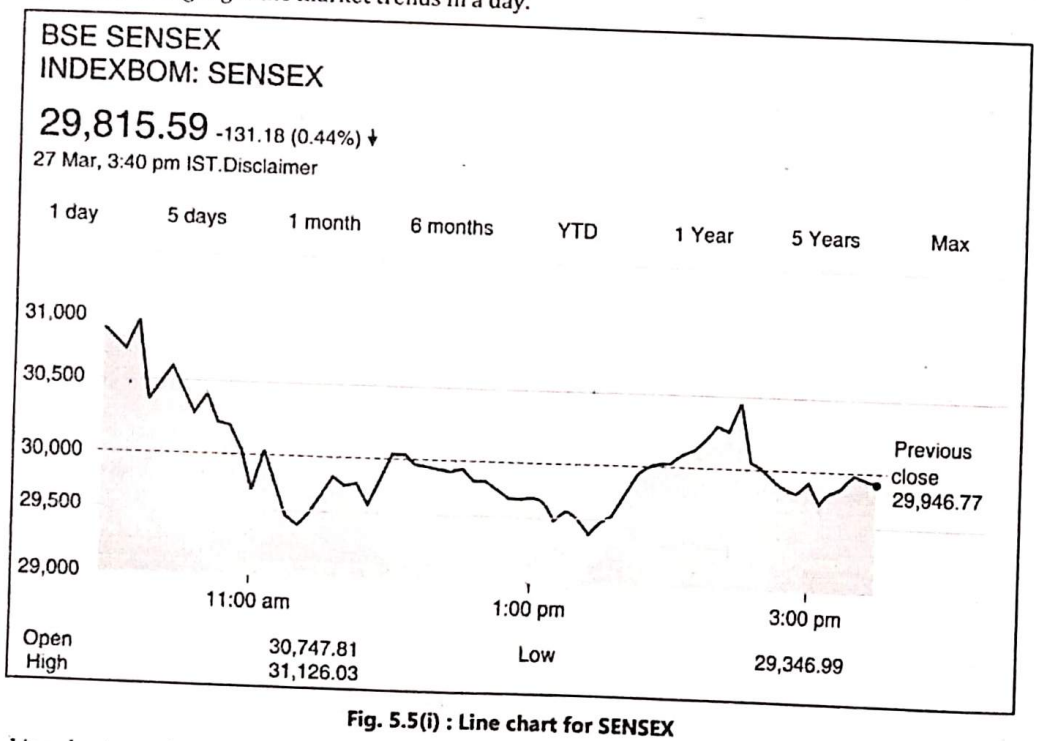


Fig. 5.5(i) : Line chart for SENSEX

- Line charts work great if you want to show variation in data for few categories. Line charts become cluttered if there are several categories.

3. Area Chart

- An area chart is very similar to a line graph, but it highlights the relative differences between items. You can use an area chart when you want to show how different items stack up or contribute to the whole.

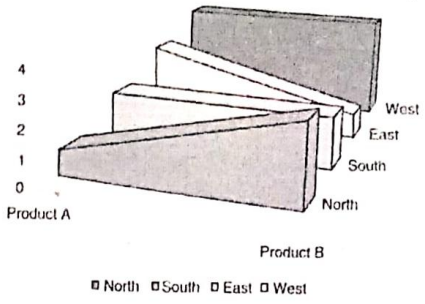


Fig. 5.5(j) : Area chart

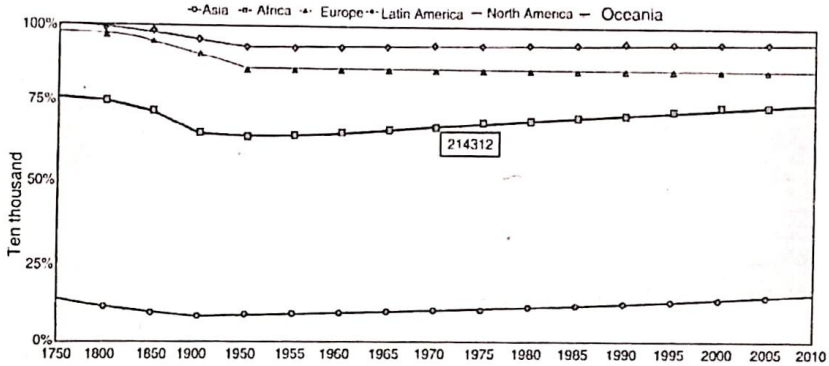


Fig. 5.5(k) : Example of an Area chart

4. Bubble Chart

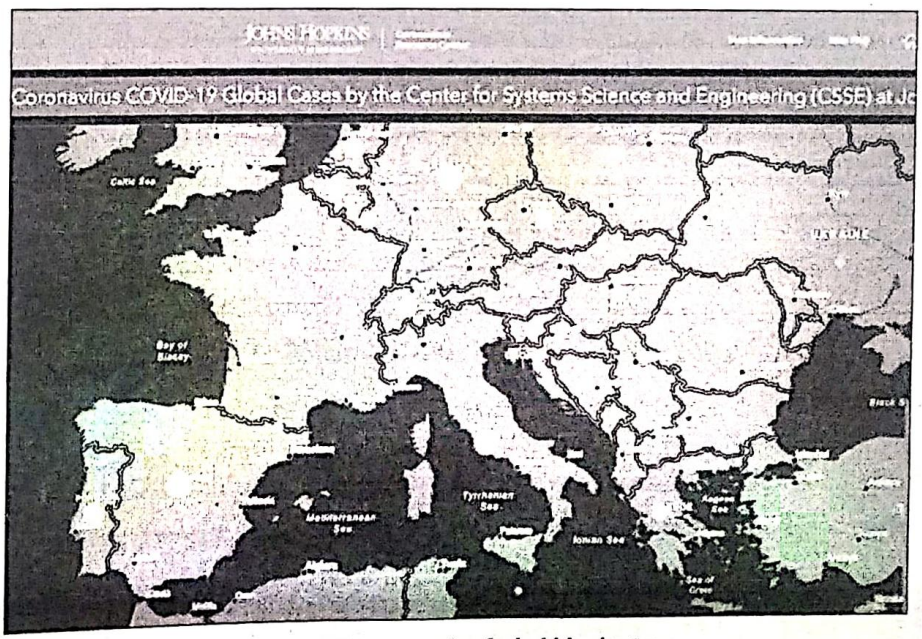


Fig. 5.5(l) : Example of a bubble chart

- A bubble chart can show comparison as well as distribution. The size of the bubble represents the value of the datapoint. The bigger the bubble, the higher is the value it represents. Bubbles can be overlaid upon a map or a geography or could stand by itself on a normal XY plot.
- For example, the Fig. 5.5(l) snapshot from John Hopkins hospital shows the distribution of Corona Virus on 29-Mar-2020 in Europe and surrounding continents. The larger the bubble, the more were the cases.

5. Pie Chart

- Like Column and Bar charts, Pie charts are also very common and simple to use and understand. A pie chart represents one static number which is divided into several categories that constitute its individual portions. Each portion is normally represented as a percentage that it contributes to the whole. When you sum up all of the separate portions, then they should add up to 100%.

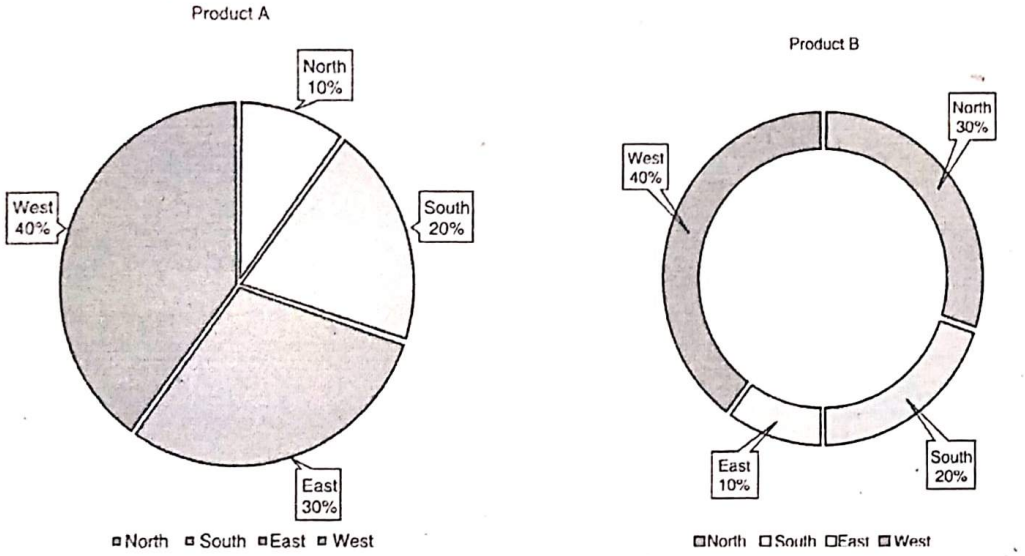


Fig. 5.5(m) : Pie chart

- Pie charts are helpful if you want to show the breakdown of data into several categories and their relationship to whole. Pie charts show individual portion sizes as well as their respective contribution to the whole.
- Pie charts are not suitable if you have several categories of data because as the number of data categories increase, the contribution of each slice in the pie tends to be become smaller and indistinguishable.

(B) Statistical Plots :

- Statistical plots are useful to show results of statistical analyses. It may not make a lot of sense to non-statistical audience. Some of the commonly used statistical plots are as follows.

1. Histograms

- A histogram is used to represent the frequency of the data in a dataset as well as its distribution. It looks similar to Column chart but is different as it plots the frequency for each distribution rather than the actual value of any datapoint itself. For example, following is a sample histogram for age group of a population.

Table 5.2(b)

Age Group	Number of People
10-18	5
19-27	7
28-36	3
37-45	12
46-54	8
55-63	9
64-72	11
73-81	2

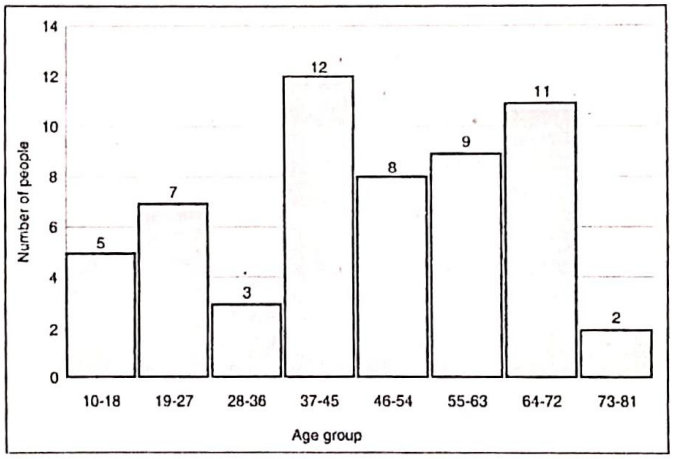


Fig. 5.5(n) : Sample histogram for age group of a population

2. Scatter Plot

- Scatter plot is used to find trends, clustering and outliers from a given dataset. In a scatter plot, the datapoints are plotted according to their coordinate values to reveal patterns.
- This is useful when looking for outliers or for understanding the distribution of your data. Scatter plots require quite a few datapoints for plotting before they start to make sense and reveal patterns.

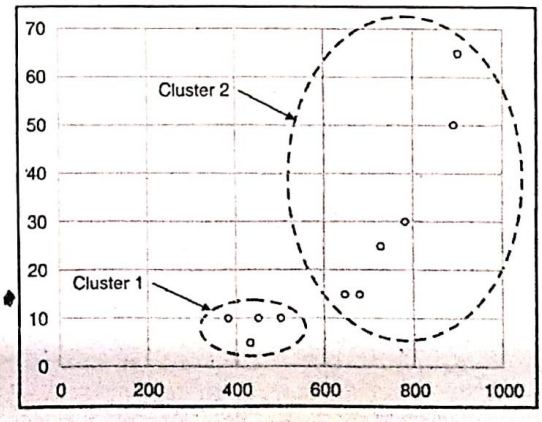


Fig. 5.5(o) : Scatter Plot

3. Box Plot

- A Box plot, or Box and Whisker diagram, shows how datapoints in a dataset are distributed. This plot is quite interesting and useful as it shows several statistical values at once. Some of the statistical values that are shown are
  - Minimum
  - 1st Quartile
  - Mean
  - Median
  - 2nd Quartile
  - 3rd Quartile
  - Maximum
- Following is a simple example of a Box Plot for marks distribution.

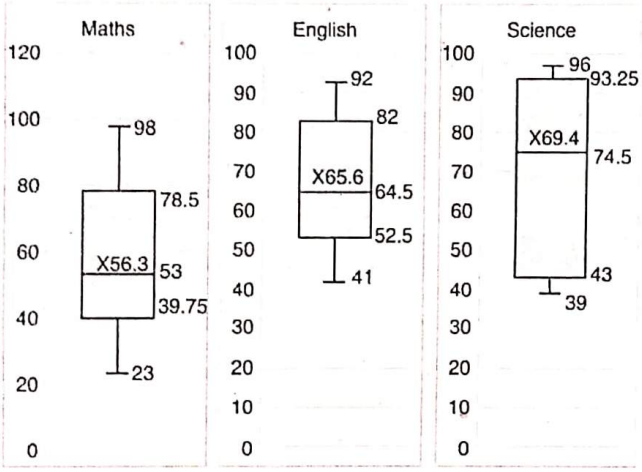


Table 5.2(c)

Student	Maths	English	Science
Ajay	50	45	65
Sunil	60	55	54
Ravi	30	67	96
Pinky	80	88	94
Raju	45	65	93
Shiva	43	59	85
Raj	23	80	84
Rinku	56	92	39
Ramesh	78	41	44
Adam	98	64	40

For example, for *Science*, the Box plot shows the following values.

Table 5.2(d)

Science Plot	
Minimum	39
1st Quartile	43
Mean	69.4
Median	74.5
2nd Quartile	74.5
3rd Quartile	93.25
Maximum	96

- Box plot is very useful if you want to statistically compare two or more distributions on the basis of mean, median, minimum, maximum and quartiles. For example, in the plot of subjects, you can make out how many students are below or above mean and what is quartile distribution amongst the subjects.

4. Radar Chart

- Radar Chart or Spider Chart is useful for understanding the relative difference between datapoints in your dataset. Radar charts help you to highlight the relative differences between the datapoints and help you focus where there are major differences that you care about.
- For example, for the following dataset, the Radar Chart visualises that sales had no difference between South and West region for Products A and B but there were differences in North and East region. How much was the difference is precisely what the Radar Chart shows. If the difference is too high, you might want to find out more about why that is.

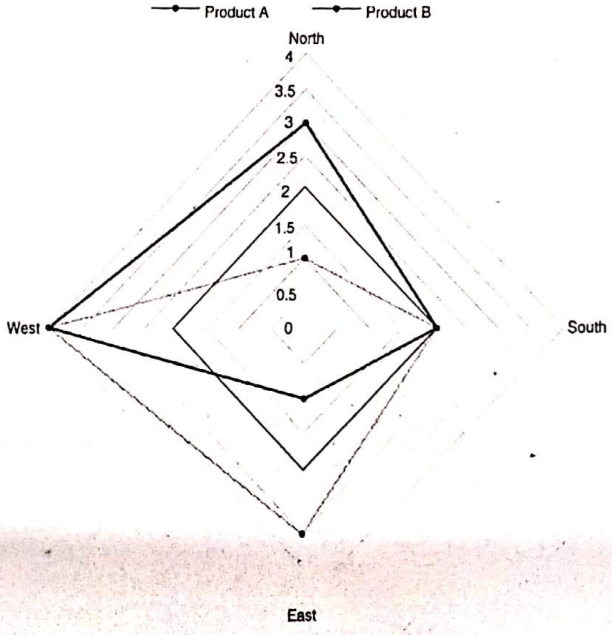


Fig. 5.5(q) : Radar chart

Table 5.2(e)

Sales Region	Product A	Product B
North	1	3
South	2	2
East	3	1
West	4	4

5. Tree Map

- Tree maps are useful to aggregate parameters of similar categories and then use area to show the relative size of each category compared to the whole. It can be used to breakdown the relationships between multiple categories in your dataset and provide a comparison as well as relationship to whole. It provides a hierarchical view of your data and makes it easy to spot patterns. The tree branches are represented by rectangles and each sub-branch is shown as a smaller rectangle.
- For example, the tree map for the following dataset shows that lunch sales are higher than the breakfast sales (since the lunch rectangle is bigger than the breakfast rectangle) and noodles is the top most sales item for lunch (as noodles rectangle's area is the highest within the lunch rectangle) and Idly and Sandwich orders are mostly equal (as the Idly and Sandwich rectangles have almost the same area).

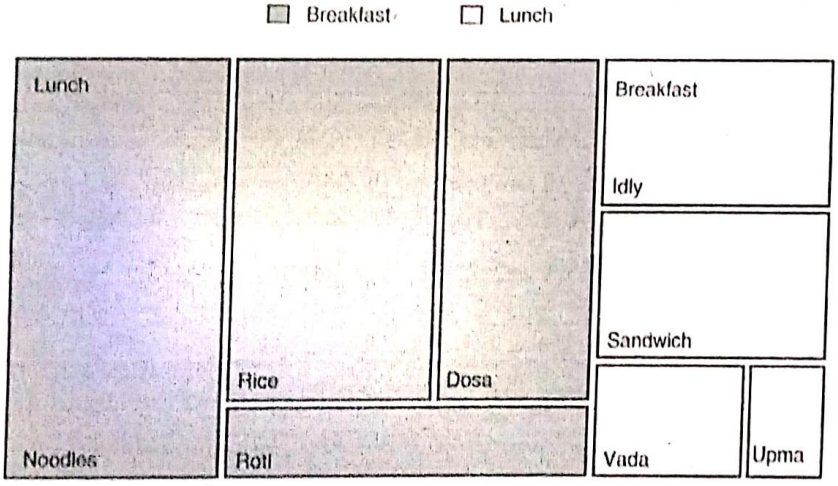


Fig. 5.5(r) : Tree map

Table 5.2(f)

Menu Type	Menu Item	Sales
Breakfast	Idly	200
Breakfast	Vada	100
Breakfast	Upma	50
Breakfast	Sandwich	200
Lunch	Dosa	300
Lunch	Rice	400
Lunch	Roti	150
Lunch	Noodles	500

6. Waterfall Chart

- A waterfall chart shows how a particular value is affected by intermediate values. The intermediate values could be negative or positive. The best thing about the waterfall chart is that each subsequent datapoint plot automatically considers all the previous datapoint plots and it is easy to understand how the particular value is changing with each datapoint on the chart.
- For example, consider that you want to plot your income sources and expenses and ultimately want to know what you are saving (how much money you are left with).
- The waterfall chart for the same would look like the Fig. 5.5(s).

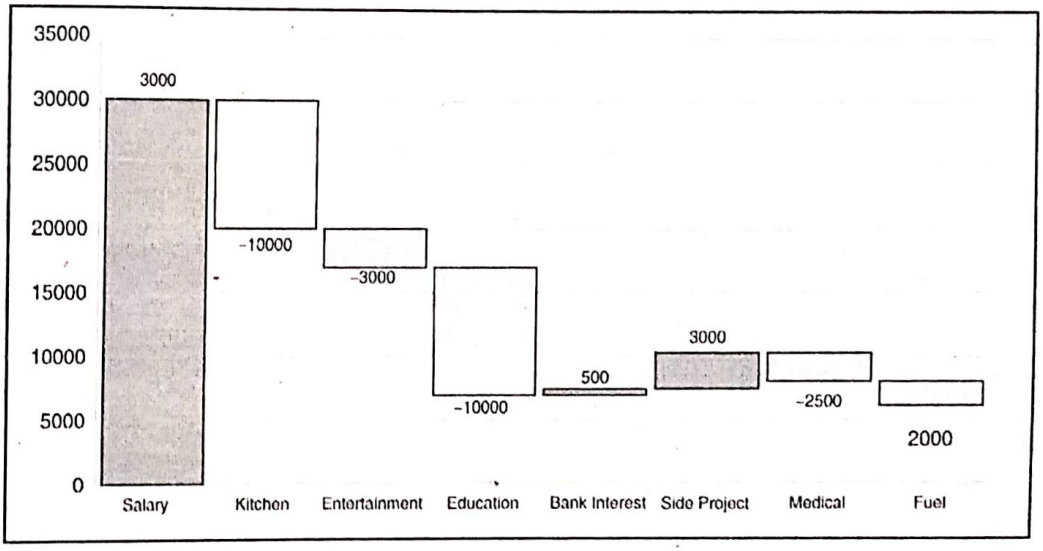


Fig. 5.5(s) : Waterfall chart

Table 5.2(g)

Particulars	Transactions
Salary	30000
Kitchen	- 10000
Entertainment	- 3000
Education	- 10000
Bank Interest	500
Side Project	3000
Medical	- 2500
Fuel	- 2000

At each datapoint plot, you know your how much money you are left with.

So,

- You got 30,000 from salary. So, you have 30,000
- After expensing your kitchen essentials you are left with 20,000

- After expensing your entertainment you are left with 17,000
- After expensing your education you are left with 7,000
- You got bank interest of 500. Hence, you now have 7,500
- You also got 3,000 from your side project. You now have 10,500.
- After expensing your medical requirements you are left with 8,000
- After expensing your fuel requirements you are left with 6,000

(C) Topology Plots

- Topology plots use geometric structures to show relationship and connectedness between datapoints in your dataset. Unlike other plots that work on numerical values, topology plots are great to show relationships, hierarchies, connectedness, etc. for datapoints that do not have numerical values associated with them.

1. Linear Topology

- Linear topology shows one-to-one relationships between entities. It depicts a process over time that must be followed to achieve a particular outcome.
- For example, after analysing the share trading data over time, you might conclude that the best time to invest in market is when the market is down. To represent this, you might create a visual depicting the linear relationship between activities based on your analysis.

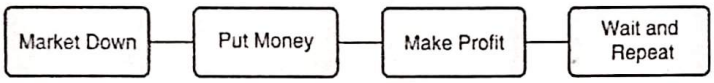


Fig. 5.5(t) : Examples of Linear Topology

- Simple, but yet very powerful to convey the message.

2. Graph Topology

- Graph topology is a great way to show multiple relationships amongst datapoints. It is highly used in social media connection analysis, finding patterns in communication and information flow and show how entities are related.
- For example, Fig. 5.5(u) a simple family graph.

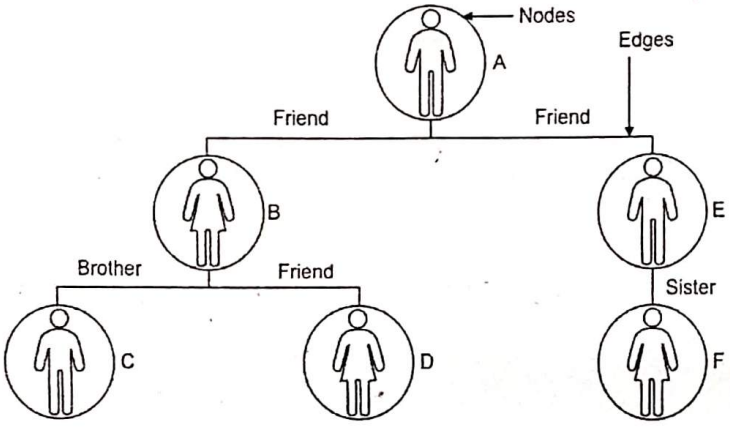


Fig. 5.5(u) : Example of Graph Topology

- A graph could really be dense and complex as the number of edges and nodes increase. For example, here is slightly denser graph than the previous one.

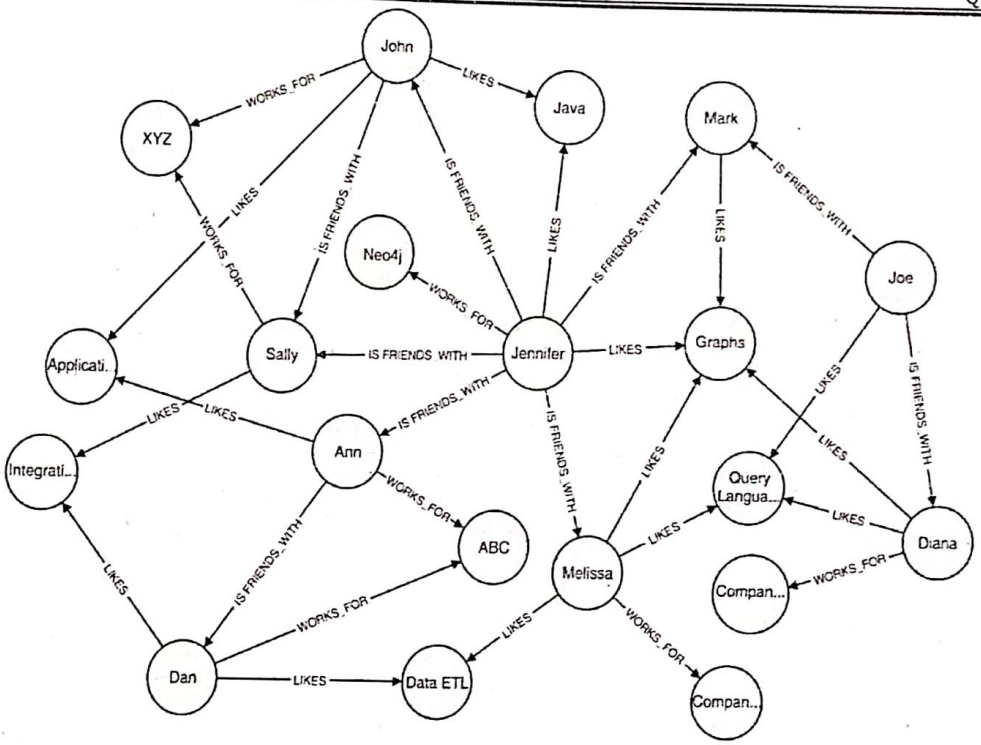


Fig. 5.5(v) : Example of Graph Topology

3. Tree Topology

A Tree Topology represents a hierarchical classification. There is a root node that begins the tree followed by several branches and leaves. The nodes act as receivers and distributors of connections, and lines represent the connections between nodes.

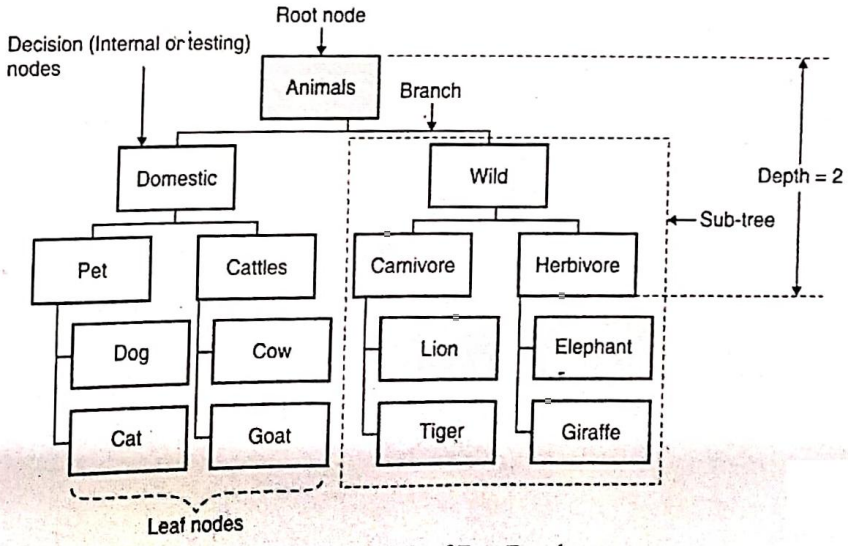


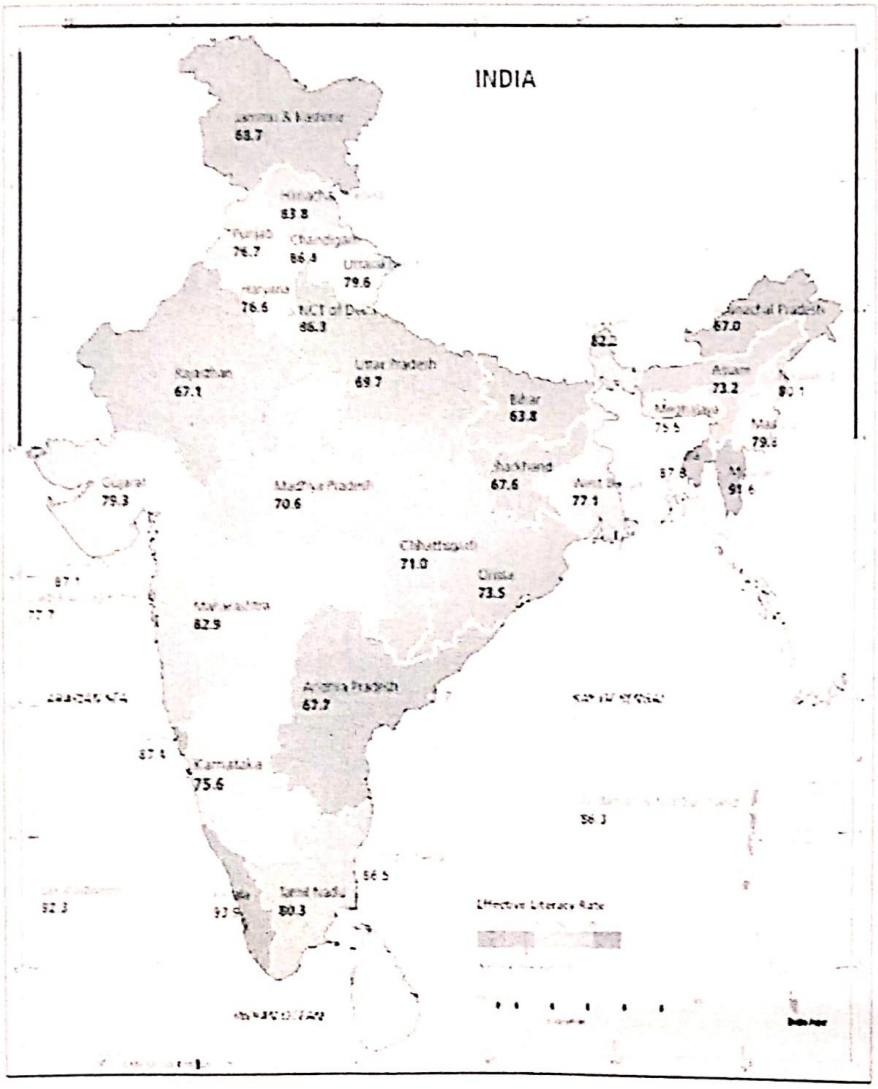
Fig. 5.5(w) : Example of Tree Topology

(D) Spatial Plots

Spatial Plots use logical space view for visualising the data. The logical space could represent a map, location, shape, or size of a data attribute.

1. Choropleth Map

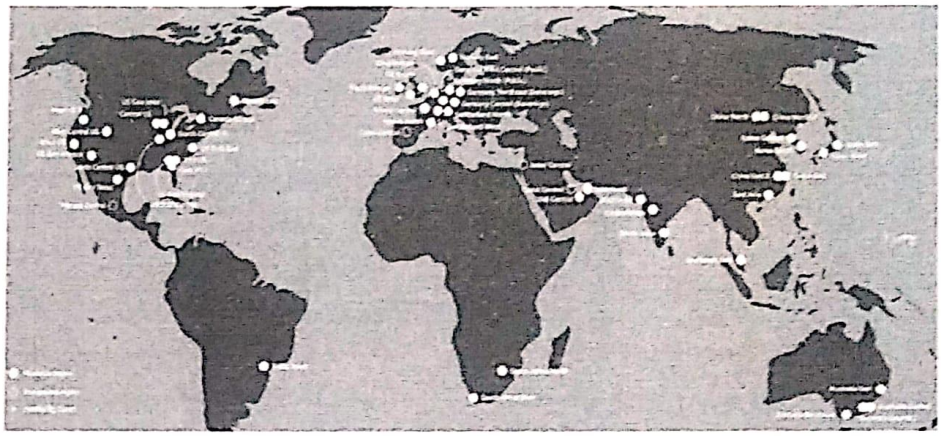
- Choropleth map uses differences in shading, colouring, or the placing of symbols within predefined areas to indicate the average values of a particular quantity in those areas. The datapoints are plotted (shaded) according to area boundary. The colour and shade of the area within each boundary represent the relative value of the attribute for that region. Usually, darker shading represents higher value of the attribute whereas lighter shades represent lower values of the attribute. Sometimes, different colours are used to highlight the differences.
- For example, the followimg Choropleth map is for literacy rate in India as per Census 2011. The darker is the shade, the higher is the literacy rate in the given state.



Choropleth Map for literacy rate in India as per census 2011

2. **Point Map**

- Instead of using colours and shading, a point map uses a symbol (such as a bubble) to highlight the datapoints and coverage. You have seen an example of Point Map in Bubble chart. The points on the map are used to highlight the distribution of data and as well as the respective size of the distribution.
- For example, following map shows are the regions available in Microsoft® Azure Cloud Platform as of 31-Mar-2020.
- The point map gives a quick view of the availability of the cloud platform in various regions very effectively.



Example of a point map

3. **Raster Surface**

- A Raster Surface is an evenly spaced grid containing a value in each cell. It could be used for anything from a satellite image map to a surface coverage map with values that have been interpolated from underlying spatial data points.
- A surface is a set of values that may vary over an infinite number of points. For example, points in an area may vary in elevation, proximity to a feature, or concentration of a particular chemical. Any of these values may be represented on the Z-axis in a three-dimensional X, Y, Z coordinate system to produce a continuous 3D surface. Raster surface data represents a surface as a grid of equally sized cells that contain the attribute values for representing the Z-value and at the X, Y location coordinates.
- A Raster Surface usually looks like the Fig. 5.5(x).

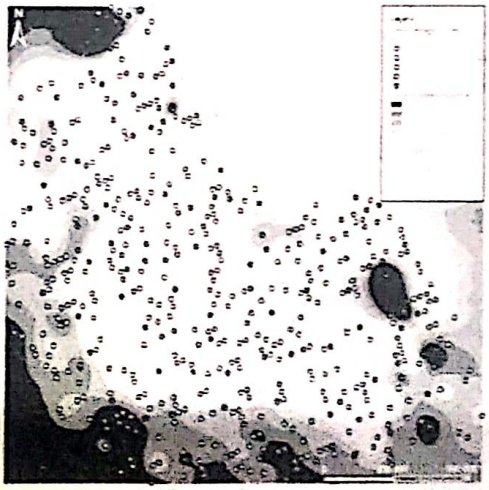


Fig. 5.5(x) : Raster Surface

4. **Heat Map**

- A Heat Map is very similar to a Choropleth Map with one major difference that the plotted area need not be a geographical boundary or region. You still use colour and shading to highlight the intensity of a data attribute on the heat map.
- For example, the Fig. 5.5(y) shows heat map highlights that researchers have found, based on eye tracking technology, that on a screen people tend to focus on face and where the face is looking. So, while creating ads, you should use the face direction appropriately to draw viewer's attention towards key points of your ad. As usual, darker, and larger areas mean higher intensity of the attribute than the lighter and smaller areas.

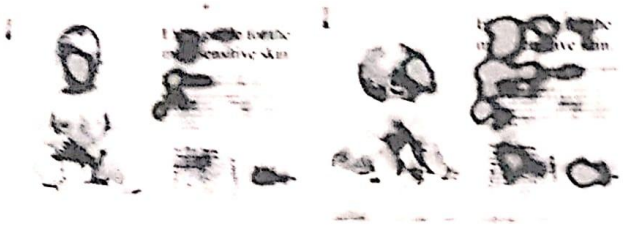


Fig. 5.5(y) : Heat Map

5. **Word Cloud**

- Word Cloud chart is very similar to a heat map chart, but it uses font size and colours to highlight the most prominent textual data. You can think of it as a heat map chart for the textual data. it is usually used to highlight keywords being used by the users or most frequently appearing text in a dataset. It is common to find these on blog sites, search pages and discussion forums to help the user to quickly navigate to a particular type of a discussion or search.
- Figs. 5.5(z) shows some of the examples of word cloud chart.

(1) (2)

Fig. 5.5(z) : Example of Word Cloud

Q. 5 Explain how data visualization is done or visually represented, if data is 1-D, if data 2-D and data is 3-Diamentional ?

SPPU - Dec. 18, 6 Marks

OR Explain data visualization with respect to 1-D, 2-D and 3-D data. **SPPU - Dec. 19, 9 Marks**

OR Describe Data Visualisation Taxonomy. **(8 Marks)**

Ans. :

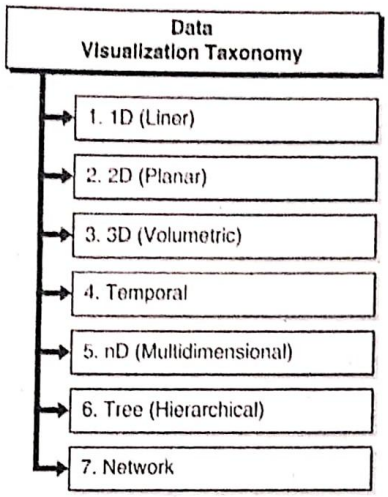


Fig. 5.6(a) : Data Visualisation Taxonomy

1. 1D (Linear) Visualisation

- 1D or 1-Dimensional visualisations are linearly or sequentially visualised.
- These could be textual documents, program source code or just alphabetical lists of names which are all organised in a sequential manner. Each item could be a line of text and may also contain additional attributes such as date, time, colour, size, etc.

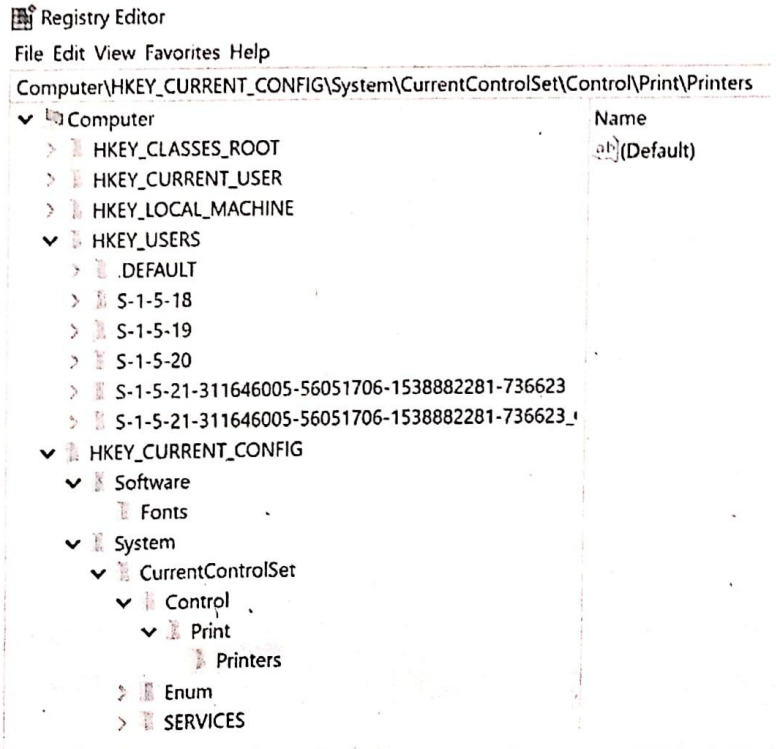


Fig. 5.6(b) : Example of 1D Visualisation

- For example, you could arrange books on a shelf in the alphabetical order of their names, or by author or by genre or some other way so as to make it easy for you to find them. Ever seen how a medicine shop owner arranges thousands of medicines and you go with a prescription and you are given medicines from big shelves within minutes!
- 1D is generally not graphically visualised. It could just be a simple listing organised in a particular way. Linear topology might be used in such a scenario. A common example of 1D visualisation could be how files or registry keys are shown on a computer.

2. **2D (Planar)**
   - 2D or 2-dimensional visualisations use planar approach for visualisation. These generally include geographic maps, floor plans, or newspaper type of layouts. Each datapoint in the dataset covers some part of the total area. Each datapoint could have various attributes such name, colour, size, etc. and may represent the distribution of data using these attributes and their respective intensities.
   - Spatial plots such as choropleth map, point map, raster surface and heat map are the most suitable forms of visualising 2D data.
3. **3D (Volumetric)**
   - 3D or 3-dimensional visualisations are used to depict real-world objects such as molecules, the human body, and buildings that have volume and some potentially complex relationship with other items (or datapoints). Computer-assisted design (CAD) systems for architects, solid modelers, and mechanical engineers are built to handle complex 3-dimensional relationships. While looking at a 3D visualisation, the user must be able to understand the position and orientation of the visualisation to get the right sense of what is seen.
   - Some of the common examples of 3D visualisations are Raster Surface, Surface Rendering, Volume Rendering, Computer Simulation, 3D Scatter Plots and 3D Bubble Charts.
4. **Temporal**
   - Temporal plots use timeline for plotting the data. The objective is to find out how a datapoint of interest changes over time. The temporal data has a start and finish time and it can overlap with other datapoints plotted on the same timeline.
   - A user might be interested in finding all events before, after, or during a time period or moment. For example, in hospitals, it is common to see a pulse meter that constantly fluctuates based on various conditions of the patient. Similarly, the Sensex graph fluctuates based on market transactions. Line charts, Area chart, Gantt chart are some of the common examples of temporal visualisations.
5. **nD (Multidimensional)**
   - Most of the data around are multidimensional involving multiple attributes and you plot the relationships between various data attributes based on the problem at hand. Most relational and statistical datasets have datapoints with n-attributes in a n-dimensional space. The visual representation can be 2-dimensional and additional dimensions could be added as desired.
   - Common visualisations such as Bar charts, Pie charts, Histograms, etc. can be used, as desired, to visualise the chosen n-dimensions of the datapoints.
6. **Tree (Hierarchical)**
   - Tree or hierarchies are collections of datapoints with each datapoint having a link to one parent datapoint (except the root). Datapoints have links between parent and child nodes and can have multiple attributes that link them with each other. You can use a tree structure to find levels in the tree, children of a parent node, or parent of children.

Tree map and Tree topology are some of the great visualisations to show hierarchies and relationships.

7. **Network**

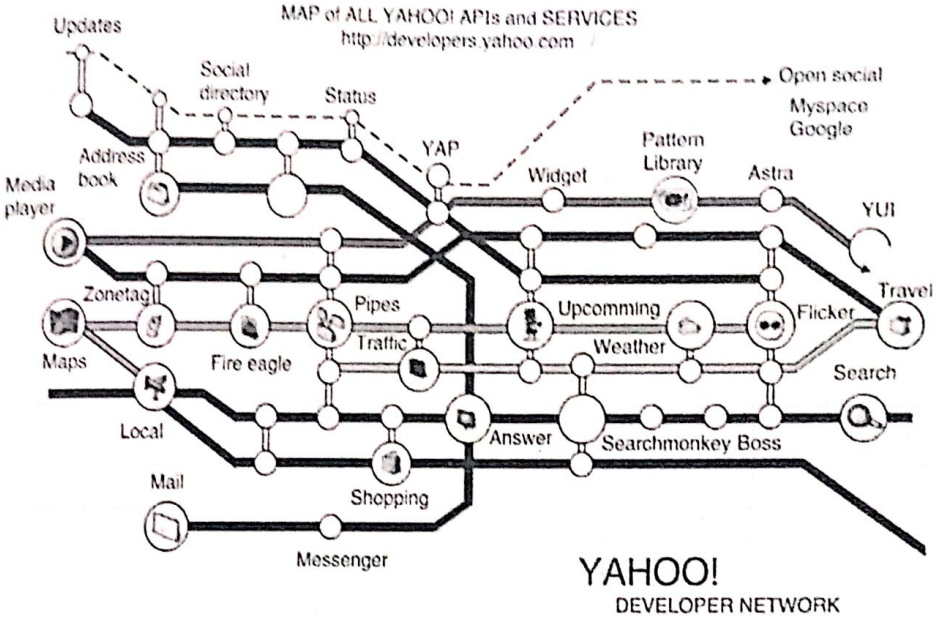


Fig. 5.6(c) : Example of a network relationship diagram

- Sometimes relationships among datapoints cannot be conveniently captured with a tree structure and it is useful to have the datapoints linked to an arbitrary number of other datapoints. Network plots are used to create links between datapoints. You can then find datapoints with certain links or attributes or could do more complex analysis such as finding the shortest or least costly paths connecting two datapoints.
- Node and link diagrams and Graph topology are some of the great visualisations to show network kind of relationship. Check out the Fig. 5.6(c) historical but interesting diagram from Yahoo! for its APIs and Services drawn as a network relationship.

Q. 6 Describe the General Workflow of Analytics and Visualisation irrespective of what tool is chosen. **(8 Marks)**

Ans. : General Workflow of Analytics and Visualisation

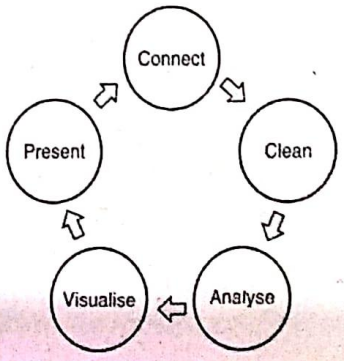


Fig. 5.7 : General workflow of analytics and visualisation

1. **Connect :** You start with connecting the analytics and visualisation tool to a data source. You can connect to various data sources as supported by the tool of your choice. Usually, the sources are databases, data warehouses, live streaming data, log files or anything else.
2. **Clean :** The collected data may not be fit for analytics and visualisation as-is. You might have to clean it up. Some of the cleaning exercises involve de-duplicating the data, filling for missing values or pruning unwanted entries.
3. **Analyse :** Once your data is clean, you decide how you need to analyse it. What is the problem at hand or the questions that you want your data to answer for you. Based on the problem at hand and the desired analysis, you carry out the analytics on the data.
4. **Visualise :** After successfully carrying out analytics on your data, you choose an appropriate visualisation based on what you want to communicate and to whom.
5. **Present :** Finally, you present your visualisation to the desired audience, share it with others or collaborate with other stakeholders to make the best use of it.

Each tool provides varied capabilities around each of these steps. But, in general, these are the steps that any tool would require to analyse and visualise the data.

Q. 7 Explain Big data visualization tools in short (any four tools). **SPPU - Dec. 18, 8 Marks**

OR Explain various tools to visualize Big Data. (Any four) **SPPU - May 19, Dec. 19, 8 Marks**

OR Enlist tools used during data preparation phase. **SPPU - Dec. 19, 5 Marks**

OR Explain data visualization Tool - Tableau. **SPPU - Dec. 19, 8 Marks**

OR Write a short note on Microsoft Power BI. **(4 Marks)**

OR Describe the data visualisation tool Qlik. **(8 Marks)**

OR Describe the data visualisation tool ThoughtSpot. **(8 Marks)**

Ans. : Some of the commonly used data analytics and visualisation tools are as following.

I. Tableau

- Tableau is a business analytics company with data visualization at its core. The company was founded in 2003. Tableau helps extract meaning from information by providing various analytics and visualising capabilities. It's an analytics platform that supports the cycle of analytics, offers visual feedback, and helps you answer questions, regardless of their evolving complexity.
- Tableau has various products around analytics and data visualisation. Some of the major ones are as following.
  1. **Tableau Desktop :** You can install it locally on your computer and perform analytics and visualisation. You connect the data sources (from where to get data for analysis and visualisation) and carry out the required steps for analysing and rightly visualising the data based on your requirements.
  2. **Tableau Prep Builder :** Your data may not be clean and thus fit for analysis and visualising. Tableau Prep Builder helps you to combine, shape, and clean your data for analysis.
  3. **Tableau Server :** You can use Tableau Server to share the data, analytics and visualisations that are created in the Tableau Desktop across the organisation. You must first publish your work in the Tableau Desktop to the Tableau Server after which it is accessible by other users. Also, various users can collaborate using Tableau Server to create new visualisations and analytics. Tableau Server needs to be managed by the organisation itself.

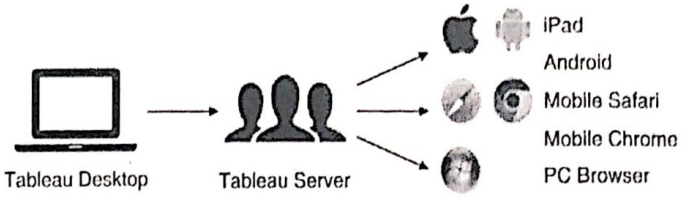


Fig. 5.8(a) : Tableau

4. **Tableau Online :** If an organisation does not want to manage its own instance of Tableau Server, then it can use Tableau Online which is a cloud service provided by Tableau itself. Tableau manages and operates the Tableau Server and any organisation can just subscribe to it. The functionalities provided are almost similar to Tableau Server. It is just that the organisation need not have to manage the Tableau Server instance when using Tableau Online.

- The high-level architecture of Tableau Server is as shown in Fig. 5.8(b).

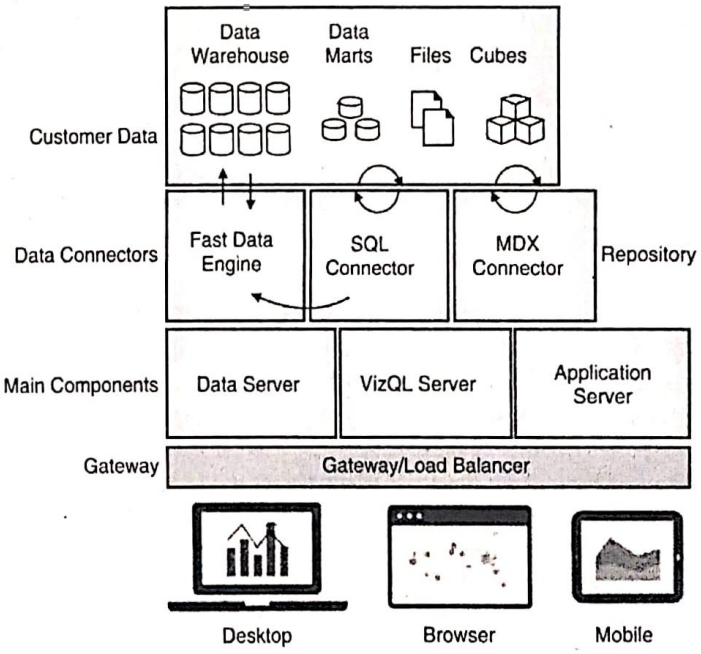


Fig. 5.8(b) : High-level architecture of Tableau

1. **Data Layer :** Tableau supports several data sources from where you can bring your data into Tableau. Most organisations have a heterogeneous data environment that might range from traditional data warehouses, live databases, or flat files such as Excel.
2. **Data Connectors :** Tableau includes over 40 optimised data connectors for data sources such as Microsoft Excel, SQL Server, Google BigQuery, Amazon Redshift, Oracle, SAP HANA, Salesforce.com, Teradata, Vertica, Cloudera, and Hadoop. Also, new data connectors are added on a regular basis. There is also a generic ODBC connector for any systems without a native connector. Tableau provides two modes for interacting with data - live connection or in-memory. Users can switch between a live and in-memory connection as they choose.

3. **Tableau Server Components :** At a high-level, Tableau has the following main components.
   (a) **Data Server :** It centrally manages, and stores Tableau data sources and provides end users with secure access to trusted data.
   (b) **VizQL Server :** VizQL is a patented query language that translates visual, drag-and-drop actions into a database query and then expresses the response graphically. This is the core of Tableau visualisation capability.
   (c) **Application Server :** It manages content browsing, server administration and permissions for the Tableau Server web and mobile interfaces.
4. **Gateway/Load Balancer :** It is used to balance the workload on the Tableau Server. The tasks are distributed across various machines. When running in a distributed environment, one physical machine is designated the primary server and the others are designated as worker servers which can run any number of other processes.
5. **Clients :** Tableau Server supports a wide variety of web browsers and mobile apps. It provides interactive dashboards to users via HTML5 in a web or mobile browser, or natively via a mobile app.

II. Microsoft Power BI

- Microsoft Power BI is another analytics and visualisation product that is a leader in the Gartner's magic quadrant. It offers data preparation, visual-based data discovery, interactive dashboards, and augmented analytics.
- Power BI is a collection of software services, apps, and connectors that work together to help you create, share, and consume business insights in the way that serves you and your business most effectively.
- Power BI has the following components.
  1. **Power BI Desktop :** It is a Windows desktop application using which you can create reports and visualisations. It can be used as a stand-alone, free, and personal analysis tool.
     It supports hundreds of connectors to fetch the data from various data sources.
  2. **Power BI service :** It is an online cloud service where you can publish your reports and collaborate with other stakeholders. The organisation need not manage the Power BI server itself and can just subscribe to the Power BI cloud service.
  3. **Power BI Report Server :** It allows you to publish Power BI reports to an on-premises report server, after creating them in Power BI Desktop. Unlike the cloud service, the organisation needs to manage the Power BI Report Server of its own.
  4. **Power BI Mobile Apps :** You can build apps to interact with the Power BI reports for Windows, iOS, and Android devices.

III. Qlik

- Qlik is another company that provides analytics and visualisation products and is a leader in the Gartner's magic quadrant. Qlik Sense is a platform for modern and self-service oriented analytics. Qlik Sense runs on the patented Qlik Associative Engine that allows you to explore information freely without the limitations of query-based tools.
- Fig. 5.8(c) shows a high-level architecture of Qlik.

1. **Hub :** The Hub is the analytics portal that lets users access the applications they are entitled to access.
2. **Management Console :** It is used to manage all aspects of the Qlik Sense platform such as data source connectivity to application, task management, security administration, monitoring, and auditing.
3. **Qlik Sense Proxy (QPS) :** It is the entry point via the Hub and Management Console. QPS integrates with various identity providers and also manages your sessions, provisions your licenses, and handles load balancing to the other components.

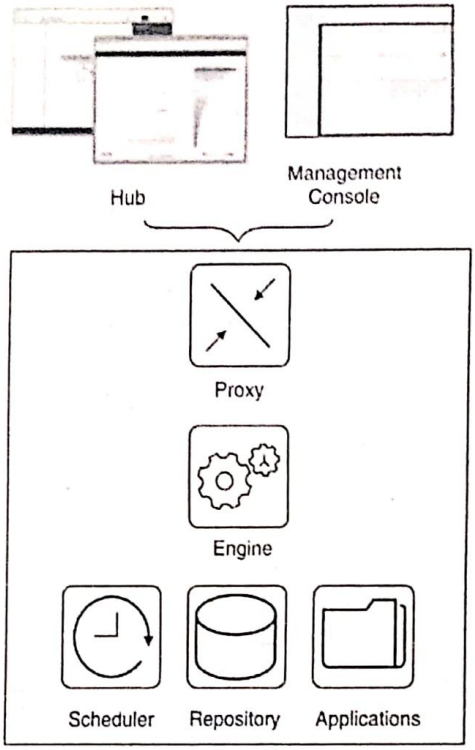


Fig. 5.8(c) : High-level architecture of Qlik

4. **Qlik Sense Engine (QES) :** It is the associative and in-memory data indexing engine. It provides highly interactive and self-service visualisations along with search and calculations at runtime.
5. **Qlik Sense Scheduler (QSS) :** It coordinates data loads from data sources. QSS supports automated data reloads.
6. **Qlik Sense Repository (QRS) :** It is a centralised storage for configuration and management information. QRS manages user definitions, security, and many other elements of the platform. It uses PostgreSQL database.
7. **Qlik Sense Applications :** It includes highly compressed data, a data model, and the presentation layer. Applications are persistently stored on a file system and are loaded into memory by the QES as users request them.

IV. ThoughtSpot

- ThoughtSpot is another analytics and visualisation product that is a leader in the Gartner's magic quadrant. ThoughtSpot differentiates itself with its search-based interface, which supports analytically complex questions with augmented analytics at scale.
- Fig. 5.8(d) shows a high-level architecture of ThoughtSpot.

1. **Data sources :** ThoughtSpot can handle a wide variety of different data sources. ThoughtSpot does all analysis against data in memory to help achieve fast results across millions and billions of records of data. ThoughtSpot caches the data in order to process it.
2. **ThoughtSpot Appliance :** The ThoughtSpot appliance can be a physical appliance or a set of virtual machines clustered together in the cloud or on-premises. From an external interface, regardless of the appliance type, the appliance appears to be a single instance.

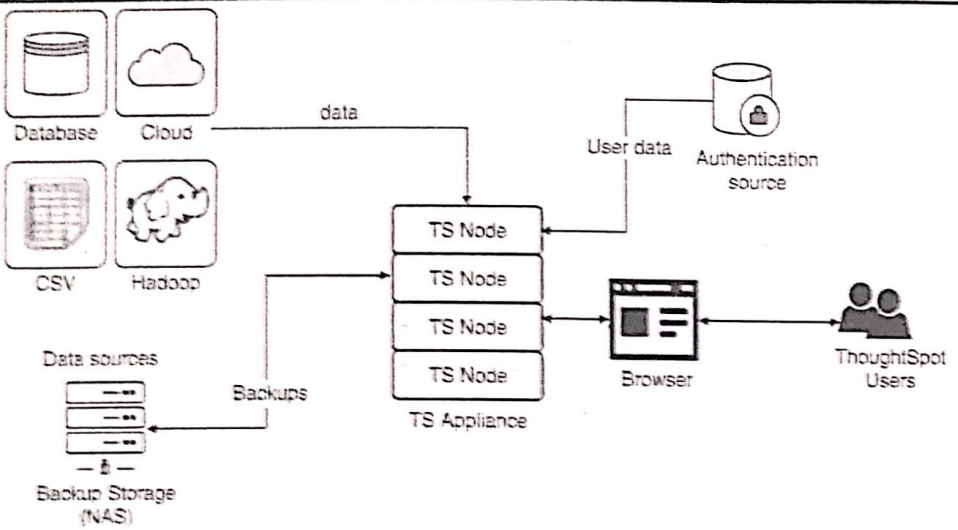


Fig. 5.8(d) : High-level architecture of ThoughtSpot

3. **Client (Browsers, apps) :** Users can access the data from a supported browser to view saved content or perform searched-based analytics.

Q. 8 Explain analytical techniques used in Big data visualization. **SPPU - Dec. 18, 3 Marks, May 19, 9 Marks**

Ans. :

- The analytical techniques that you have learnt so far are used to analyse data before creating visualisations out of them. Some of the techniques themselves form a visualisation of data points as they are analysed while others can provide with the analysis and then you can decide an appropriate visualisation for them separately.
- Some of the common analytical techniques used in big data visualisation are shown in Fig. 5.9.

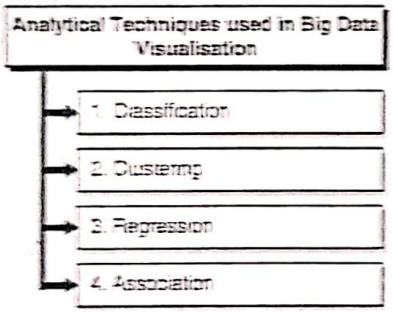


Fig. 5.9 : Analytical techniques used in big data visualisation

1. **Classification :** You can use logistic regression, decision tree or Naive Bayes classifier to classify your data points and then appropriately visualise them. Topology plots such as linear topology, graph topology or tree topology are some of the most suitable formats of such creating visualisation based on classification.
2. **Clustering :** Clustering is a technique in which the data points are arranged in similar groups dynamically without any pre-assignment of groups. Clustering provides a good visualisation of similar data points and helps to differentiate or find patterns amongst them.

   Spatial plots such as choropleth map, point map, heat map, word cloud and statistical plots such as scatter plot and histograms are some of the most suitable visualisations for clustering.

3. **Regression :** Regression analysis provides a functional relationship between two or more correlated variables that is often empirically determined from data and is used specially to predict values of one variable when given values of the others. Comparative plots such as column charts, line charts, area chart, bubble chart and statistical plots such as radar chart, tree map, box plot and waterfall chart are some of the most suitable visualisations for regression.

4. **Association :** Association rule is a method to learn about relationships amongst objects (or items) in a large dataset. Topology plots such as linear topology, graph topology or tree topology are some of the most suitable formats of such creating visualisation based on association rules. You may also use line chart and area chart for visualising association rules.

□□□

Unit VI : Big Data Technologies Application and Impact

Q. 1 Write a short note on Social Network Analysis (SNA). **(4 Marks)**

Ans. : Social Network Analysis (SNA)

The dictionary meaning of graph is "the collection of all points whose coordinates satisfy a given relation". Social networks use graph theory to model pairwise relations between objects.

- **Definition:** Social network analysis (SNA) is the process of investigating social structures through the use of networks and graph theory.

  Each graph has nodes (or vertex) and edges.
- **Definition:** Vertex or node represents an object in the graph.
- **Definition:** The connection between the nodes is called an edge or a link.

The nodes in the graph represent the users or objects and the edges represent the relationship between the nodes.

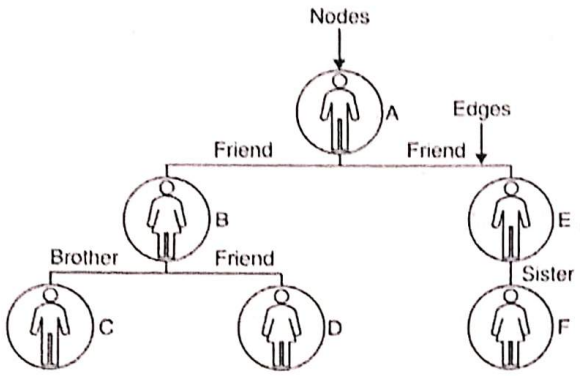


Fig. 6.1

Some of the basic properties of a graph that are used for Social Network Analysis.

1. **Degree of a Node :** Degree of a node is the number of edges it has. For example, degree of node A is 2 whereas degree of node F is 1.
2. **Path Length :** Path length is the distance between two nodes. For example, the path length between node A and node D is 2 (node A→ node B → node D) whereas the path length between node D and node F is 4 (node D → node B → node A → node E → node F).
3. **Centrality :** Centrality refers to the "importance" or "influence" of a particular node within a network. It is often a key node that joins several networks. For example, node A can be treated to have high centrality because it joins the two groups formed by node B and node E.
4. **Density :** Density refers to the proportion of actual direct connections versus total number of direct connections possible in the network. For example, node A is only directly connected to node B and node E. Had it been also directly connected to other nodes (node C, node D and node F), then the network would be considered denser.
5. **Closeness :** Closeness of a node is the average length of the shortest path between the node and all other nodes in the graph.
6. **Betweenness :** Betweenness refers to the number of times a node acts as a bridge along the shortest path between two other nodes. For example, node A is a bridge between node B and node E and also node C and node F.

Q. 2 What are some of the major applications of social network analysis? **(4 Marks)**

Ans. :

Need (Applications) of Social Network Analysis

Some of the major applications of SNA are as following.

1. **Identifying the Influencers :** In social networks, influencers are people who have the ability to influence potential buyers of a product or service by promoting or recommending the items on social media. Influencer marketing (also known as influence marketing) is a form of social media marketing involving endorsements and product placement from influencers, people and organisations who have a purported expert level of knowledge or social influence in their field.

   Influencers are someone with the power to affect the buying habits or quantifiable actions of others by uploading some form of original-often sponsored-content to social media platforms like Instagram, YouTube, Snapchat or other online channels. Influencer marketing is when a brand enrols influencers who have an established credibility and audience on social media platforms to discuss or mention the brand in a social media post. Influencer content may be framed as testimonial advertising.

   Social network analysis helps in identifying such influencers based on several criteria such as geography, demographics, topics, etc.

2. **Human Resource Management (HRM) :** HRM often strives to identify critical resources and understand their contribution to the organisation flow, collaboration, participation, and information flow. Using SNA, an organisation can optimise the talent connections, productivity, and utilisation. It also helps to identify the reach of an individual, identify accelerators of growth and poorly connected resources, and decide whom to give more opportunity.

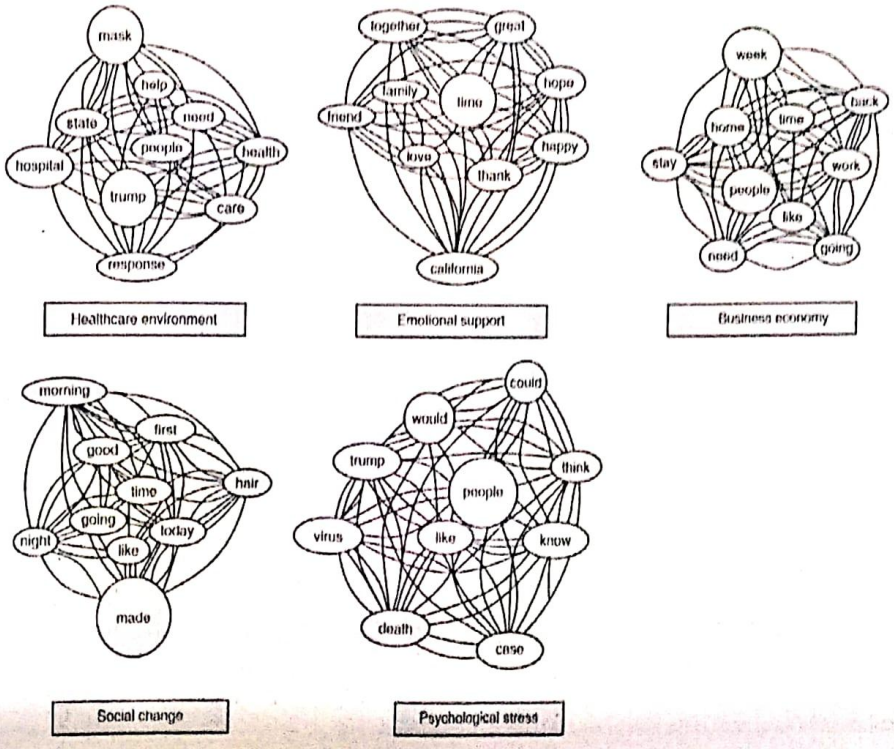


Fig. 6.2

3. **Contact tracing :** SNA could also be used for contract tracing for infectious diseases (such as Covid-19). SNA could help to identify and isolate individuals and groups with high betweenness and out-degree centrality (transmitters of disease) and implement sound contact tracing activities to reduce the impact and spread.
4. **Identify themes and connections :** SNA can also identify dominant themes and relations between keywords and identify the sentiments. For example, the Pgs. 6.2, from Journal of Medical Internet Search, shows the connection between the top 10 words for COVID-19 themes.
5. **Fraud detection :** Financial organisations can use SNA for fraud detection. Fraud is often organised by groups of people loosely connected to each other. Such a network mapping enables financial institutions to identify customers who may have relations to individuals or organisations on their criminal watchlist (network) and take precautionary measures.

Q. 3 Write a short note on mobile analytics. **(4 Marks)**

Ans. :

- **Definition :** Mobile analytics allows companies to track and analyse the behaviour of customers on mobile applications and devices.
- Mobile analytics captures data from mobile app, website, and web app visitors to identify unique users, track their journeys, record their behaviour, and report on the app's performance. Similar to traditional web analytics, mobile analytics are used to improve conversions (browse to purchase), and are the key to crafting world-class mobile experiences.
- Mobile analytics gives companies unparalleled insights into the otherwise hidden lives of app users. Analytics usually comes in the form of a software that integrates into companies' existing websites and apps to capture, store, and analyse the data. This data is vitally important to marketing, sales, and product management teams who use it to make more informed decisions. Without a mobile analytics solution, companies are left flying blind. They are unable to tell what users engage with, who those users are, what brings them to the site or app, and why they leave. Companies in this situation must rely on intuition or domain expertise and often underperform compared to their peers.
- Mobile analytics typically track the following.
  - Page views
  - Visits
  - Visitors
  - Source data
  - Strings of actions
  - Location
  - Device information
  - Downloads / Shares
  - Login / logout
  - Custom event data
- A typical mobile analytics dashboard could look like the following.

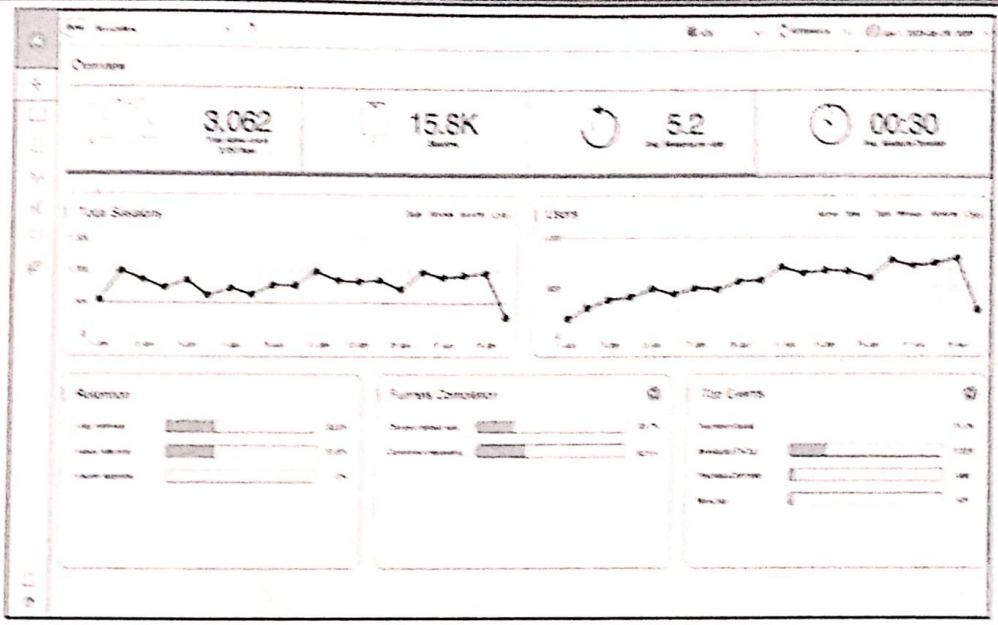


Fig. 6.3

- Companies use this data to figure out what users want in order to deliver a more satisfying user experience. For example, they are able to get insights into the following.
  - What draws visitors to the mobile site or app
  - How long visitors typically stay
  - What features visitors interact with
  - Where visitors encounter problems
  - What factors are correlated with outcomes like purchases
  - What factors lead to higher usage and long-term retention
- Different teams use mobile analytics for different purposes. For example,
  - Marketing team tracks campaign ROI, segments users, automates marketing
  - User experience team tracks behaviours, tests features, measures user experience
  - Product management team tracks usage, A/B test features, debugs, sets alerts
  - Technical and development teams track performance metrics such as app crashes

---

Q. 4 Explain the primary activities of Michael Porter's value chain. **(4 Marks)**

OR Explain the secondary activities of Michael Porter's value chain. **(4 Marks)**

OR Explain Michael Porter's value chain. **(6 Marks)**

Ans. :

- Michael Porter is an American academic known for his theories on economics, business strategy, and social causes. He is credited for creating Porter's five forces analysis, which is instrumental in business strategy development today.

ıbination of the systems a company or organisation uses to make money. That is, a value rious subsystems that are used to create products or services. This includes the process ven the importance of the value chain, Michael Porter developed a strategic management mpany's value chain. Porter sought to define a company's competitive advantage noting pany's processes, such as marketing and supporting activities.

hael Porter's value chain activities.

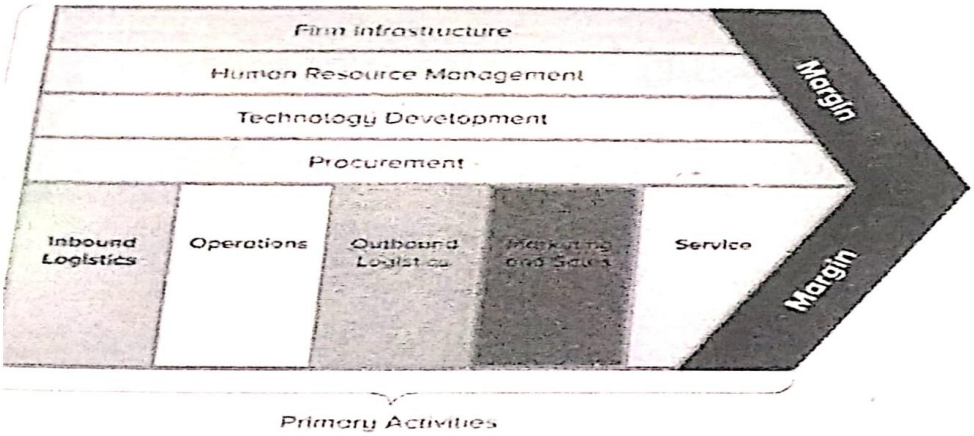


Fig. 6.4

ılysis into five primary activities. Then, he further breaks those down into four nary activities.

el Porter's value chain are as following.

und logistics include the receiving, warehousing, and inventory control of a his also covers all relationships with suppliers. For example, for an e-commerce would be the receiving and storing of products from a manufacturer that it plans

lude procedures for converting raw materials into a finished product or service. puts to ready them as outputs. In the above e-commerce example, this would ding or packaging several products as a bundle to add value to the product.

ivities to distribute a final product to a consumer are considered outbound y of the product but also includes storage and distribution systems and can be commerce company above, this includes storing products for shipping and the s.

tegies to enhance visibility and target appropriate customers-such as cing-are included in marketing and sales. Basically, these is all activities that rchase a company's product or service. Continuing with the above example, n ads on Instagram or build an email list for email marketing.

s to maintain products and enhance consumer experience. For example, repair, refund, and exchange. For an e-commerce company, this could r a warranty.

- The goal of the five sets of activities is to create value that exceeds the cost of conducting that activity, therefore generating a higher profit.
- Companies can further improve the primary activities of their value chain with secondary activities. Value chain support activities do just that, they support the primary activities. The support, or secondary, activity generally plays a role in each primary activity.
- The four supporting activities in the value chain are as following.
  1. **Procurement :** Procurement is the acquisition of inputs, or resources, for the firm. This is how a company obtains raw materials. Thus, it includes finding and negotiating prices with suppliers and vendors. This relates heavily to the inbound logistics primary activity, where an e-commerce company would look to procure materials or goods for resale.
  2. **Human Resource Management :** Human resource management is about hiring and retaining employees who will fulfil business strategy, as well as help design, market, and sell the product. Overall, managing employees is useful for all primary activities, where employees and effective hiring are needed for marketing, logistics, and operations, among others.
  3. **Infrastructure :** Infrastructure covers a company's support systems and the functions that allow it to maintain operations. This includes all accounting, legal, and administrative functions. A solid infrastructure is necessary for all primary functions.
  4. **Technological Development :** Technological development is used during research and development and can include designing and developing manufacturing techniques and automating processes. This includes equipment, hardware, software, procedures, and technical knowledge. Overall, a business working to reduce technology costs, such as shifting from a hardware storage system to the cloud, is technological development.

Q. 5 Describe the Big Data strategy document. **[6 Marks]**

Ans. :

- One of the key challenges IT organisations face in building support for a Big Data initiative is to ensure that the Big Data initiative is valued by, or of value to, the business stakeholders. Unfortunately, business stakeholders have become numb to the IT promises of the next great technology "silver bullet." They are hesitant to believe that another new technology is going to solve all of their data and analytic problems. Time and time again, the business stakeholders have been misled about the ease-of-use and the capabilities of these new technologies. This has led to walls being built between the IT and business teams.
- The Big Data strategy document ensures the business relevance of your Big Data initiative. While this exercise is not trivial, it does provide a repeatable process and framework to ensure that your Big Data efforts support the business's key initiatives. The document enforces a discipline that any organisation can follow, as long as you truly understand and are focused on your organisation's key business initiatives.
- The Big Data strategy document is
  1. Concise in that it fits onto a single page so that anyone can review it quickly to ensure they are working on the top priority items.
  2. Clear in defining what the organisation and individuals need to do and accomplish in order to achieve the targeted strategic initiatives.
  3. Relevant to the business stakeholders by starting and focusing the process on supporting the organisation's overall business strategy, and identifying the supporting business initiatives before diving into the technology, architecture, data, and analytic requirements.

- The Big Data strategy document has the following sections.
  1. **Business Strategy :** The targeted business strategy is captured as the title of the document and clearly defines the scope upon which the Big Data initiative will be focused. The title should not be more than one sentence but should still provide enough detail to clearly identify the overall business objective. For example, "Improve customer intimacy" or "Reduce operational maintenance costs" or "Improve new product launch effectiveness."
  2. **Business Initiatives :** This section breaks down the business strategy into its supporting business initiatives. A business initiative is defined as a cross-functional project lasting 9 to 12 months in duration, with clearly stated financial or business goals against which success of the business initiative will be measured. Note that there should not be more than three to five business initiatives per business strategy. More than that and you have a wish list.
  3. **Outcomes and Critical Success Factors (CSF) :** This section captures the outcomes and critical success factors necessary to support the successful execution of the organisation's key business initiatives. Outcomes define the desired or ideal end state. Critical success factors define "what needs to be done" for the business initiative to be successful.
  4. **Tasks :** This section provides the next level of detail by documenting the specific tasks that need to be executed to perfection to be successful in support of the targeted business initiatives. These are the key tasks around which the different parts of the organisation will need to collaborate to achieve the business initiatives. This is the "how to do it" section of the document, and it is at this level of detail where personal assignments and management objectives can be defined, assigned, and measured. One would normally expect 8 to 12 key tasks being identified and linked to the targeted business initiatives as part of the Big Data Strategy Document.
  5. **Data Sources :** Finally, the document highlights the key data sources required to support the business strategy and the supporting key business initiatives. From the definition of the tasks, you should have a strong understanding of the key metrics and measures, important business dimensions, level of granularity, and frequency of data access.
- The Fig. 6.5 gives an example of a Big Data strategy document.

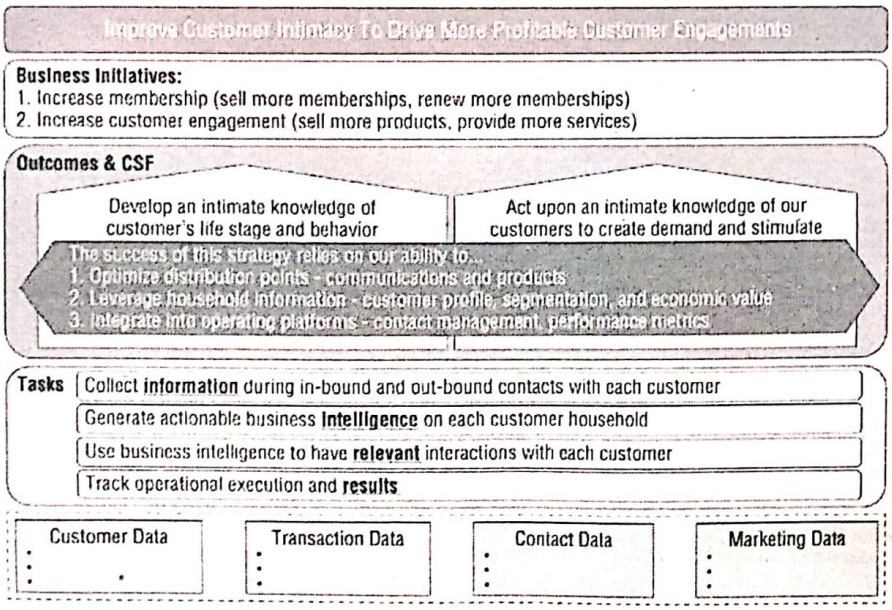


Fig. 6.5

Q. 6 Describe the steps that you would take for identifying Big Data use cases. **(6 Marks)**

Ans. :

- The biggest challenge with most Big Data projects is identifying where and how to start the Big Data journey. These selections are complicated by the fact that most business users (as well as most IT leaders) have a hard time envisioning the realm of what is possible with respect to leveraging new sources of big data (social, mobile, logs, telemetry, sensor, and others) and new Big Data technology innovations (Hadoop, MapReduce, NoSQL, and others). This is precisely where the Big Data envisioning process, which is called the vision workshop, comes into play.
- The envisioning process, which is called the vision workshop, defines where and how Big Data and advanced analytics can be deployed to transform your business. The vision workshop process is typically carried out in a facilitated, half-day ideation workshop that leverages group dynamics and the envisioning techniques (such as brainstorming and Delphi methods) to tease out and prioritise the Big Data business opportunities. The workshop process does this by helping the business and IT stakeholders envision the "realm of the possible" with respect to new Big Data sources and new Big Data technologies. The vision workshop process is comprised of the following steps.

Fig. 6.6

1. **Research and interviews to understand the targeted business initiative or business process :** Prior to the facilitated ideation workshop, the business and IT teams need to identify the targeted business opportunity, challenge, or initiative upon which to focus the vision workshop. Some of the common business initiative examples are as following.
   (a) Leverage subscriber behavioural insights to reduce churn rate and optimise customer engagement points.
   (b) Leverage predictive analytics to improve turbine maintenance predictability and reduce unplanned maintenance.
   (c) Leverage in-store behavioural patterns combined with customer historical purchases to power location-based offers.
   (d) Leverage internal and external customer communications data to flag service and product problem areas and improve customer satisfaction levels.
   (e) Leverage real-time student test results, combined with historical student test performance data, to dynamically measure student learning effectiveness.
   (f) Leverage patient health and lifestyle data to improve hospital and care readmissions predictability.

With the targeted business initiative established, the facilitation team now gets engaged. The facilitation team knows what business stakeholders to engage and what data will need to be acquired, given the targeted business initiative, to fuel the ideation exercises. After the facilitation team has gained consensus on the vision workshop scope (targeted business initiative) and business and IT participants have been finalised, the facilitation team should research the targeted business initiative and collect relevant background information.

2. **Data preparation and client-specific analytics development :** Next, the facilitation team collaborates with the IT team to identify and acquire a small sample set of your data that is relevant to the targeted business initiative. This data is used to develop a business initiative-specific "art of the possible" envisioning exercise during the ideation workshop. The data scientist member of the facilitation team is chartered to explore, enrich, and analyse the data using advanced analytics and data visualisation techniques.

3. **Envisioning exercises to convey the "realm of the possible" :** At this stage, you are ready for the one-day ideation workshop. The goal of the ideation workshop is to employ the various business valuation techniques coupled with the client-specific envisioning exercise that you just developed using the client's data, to help the business stakeholders to brainstorm how these new sources of big data (both internal and external data sources) coupled with advanced analytics can provide unique insights for use with their targeted business initiative. You should inspire the business stakeholders to envision how they might leverage internal and external data sources to help them:

   (a) Answer the business questions they need to answer in support of the targeted business initiative. You should challenge them to rethink the questions they ask of the business, and to contemplate the potential business impact of answering those questions at a lower level of granularity, with new metrics (gleaned from structured and unstructured data sources, both internal and external to the organization), and across more dimensions of the business.

   (b) Make the decisions that are necessary to support the targeted business initiative. You should challenge the business users to explore more detailed, timelier, and more robust decisions enabled by access to new sources of data, coupled with advanced analytics to uncover the drivers for each of the key decisions.

   (c) The ideation workshop should cover three key envisioning steps: brainstorming, prioritization, and documentation.

4. **Brainstorm and prioritise big data use cases :** Finally, you should guide the workshop participants through a prioritisation process where each use case is judged based on its relative business value with respect to its implementation feasibility. During this process, you should capture details regarding the business value drivers (for instance, why one business opportunity was valued more highly than another) and the reasons behind the feasibility determination (such as why one business opportunity is more difficult to implement than another).

5. **Capture implementation risks and business value drivers :** As the last step, you should summarise the identified and prioritised business opportunities, and recommend steps for deploying advanced analytics in support of the targeted business initiatives. You should document the results of the envisioning process which include:

   (a) Key interview findings as related to the targeted business initiative including key business questions, business decisions, and required data sources

   (b) Analytic use cases that came out of the brainstorming step

   (c) The Prioritisation Matrix results including details on the placement of each use case, business value drivers, and implementation risk items

   (d) Recommended next steps

The final stage of the vision process workshop is a presentation of the findings and recommendations, as well, as the detailed insights from the envisioning exercise to executive management. The findings and recommendations should confirm the relevance of big data to help drive the targeted business initiative and determine next steps for implementation.

□□□