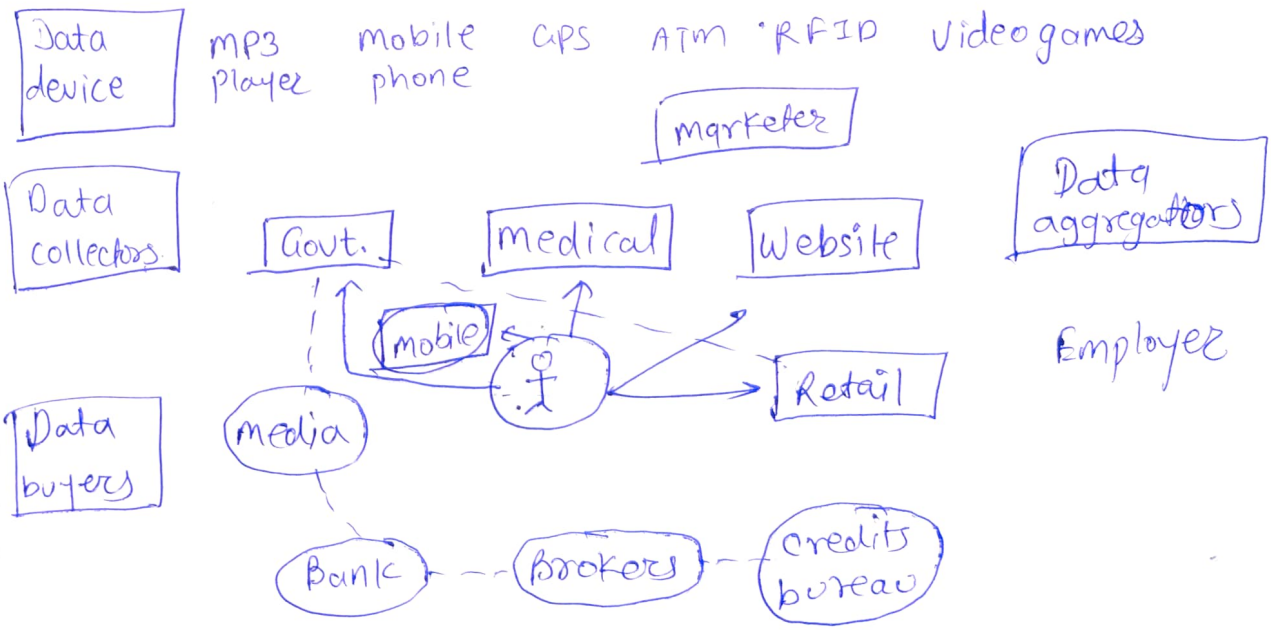
UNIT-3 Big Data Processing

Q1] Explain Big data Ecosystem with Suitable diagram.

→ • Big data ecosystem is comprehensive of massive functional components with various enabling tools. Capabilities of Big data ecosystem is not only about storing & computing big data, but also about providing systematic platform & big data Analysis.

- Organizations & data collectors that can gather data from individuals started realizing that new economy is emerging as data business.
- As ecosystem is growing, groups of inbest have been formed. Currently there are four main group associated with each other. Data devices, data collectors, data aggregators & data users & buyers.



1] Data device: Data device & sensor network gather data from various locations & continuosly generate new data.

example of data devices: playing games, smart phones

Sensor data: Growing network of sensor devices generate data based on monitoring, such as temperature, sound.

Mobile networks: Mobile network generates large number of data to share picture, video, audio file & text.

2] Data Collectors :→

- It includes sample entities that collect data from the device & user.
  - Retail stores tracks customers that what they buy most what they may buy.
  - Data collector example: Government, Retail Stores,

3] Data aggregators :→

- The entities which process collected data from the first layer make them understandable.
- They give them additional value to prepare them for the handling over process.

4] Data users & buyers.

- These entities represent a group of the final layer from the Big Data ecosystem.

- Data users may want to track or prepare for nature disasters by identifying which areas a hurricane will affect first hand. It can be observed by tracking tweets about it or discussing it in social media.

8] Explain Google File System? Explain its architecture.

→ - The Google file system is a distributed file system designed to handle large-scale data processing & storage
- GFS was build as the fundamental storage service for Google search engine.
- It's build for fault-tolerance, scalability & high Performance.
- There is no data cache in GFS as large streaming reads & writes represent neither time nor space locality.

★ GFS architecture ★

- A GFS cluster consists of single master and multiple chunk servers and is accessed by multiple clients.

Basic terms :

1) Master :→ Stores metadata, manages namespace, chunk creation & replical placement.
   - Handles communication with client for ~~storing~~ locating chunks.

2] Chunk → • fixed sized unit of data ; It can be of 64mB block.
- Each file is divided into multiple chunks
- It is identified by unique 64 bit chunk_id.

3] chunk server → • Stores actual chunk data.
- Manage read/write operations requested by client.
- Stores multiple replica for fault tolerance

4] Replica :→ • A duplication of a chunk for redundancy
- Help ensure data availability & reliability

5] Client :→ Request file operations (Read/write) on GFS.
- Communicates with master to get chunk operation locations.

- It is easy to run clientserver & client on same machine, as long as machine resources permit.
- Files are divided into fixed size chunk where they are assigned with fix 64 bit size unique key at the time of creation
- Chunkserver store chunk on local disk as Linux files & read or write chunk specified by chunk handle & byte range.
- for reliability, each chunk is replicated on multiple chunkservers.
- The master is responsible for maintaining metadata. That includes namespace, chunk creation etc..
- Client communicate with master for metadata operation but all communication goes directly to the chunkserver
- Neither client nor clientserver caches file data
- Client never read & write file data through the master. Instead client asks the master which chunkserver it should contact. It caches the information for limited time & interacts with chunkserver directly
- further read of the same chunk requires no more client-master interaction until the cached information expires..

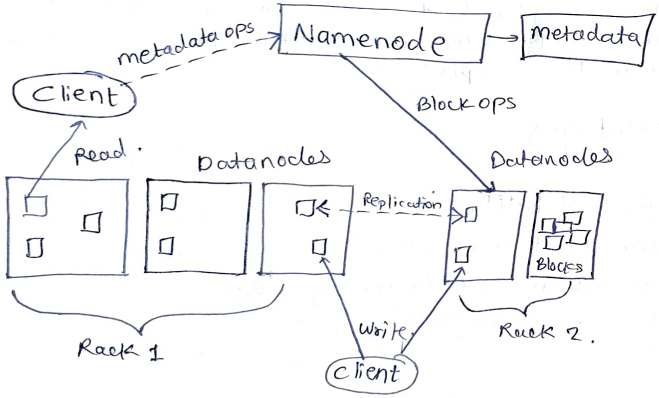
Advantages

- High scalability
- High fault tollerance.
- Efficient metadata management
- Better performance.

Q5] Explain Hadoop Distributed File System.

→ • Hadoop Distributed file system (HDFS) is a distributed file system inspired by afs that organizes files & store their data on a distributed computing system.

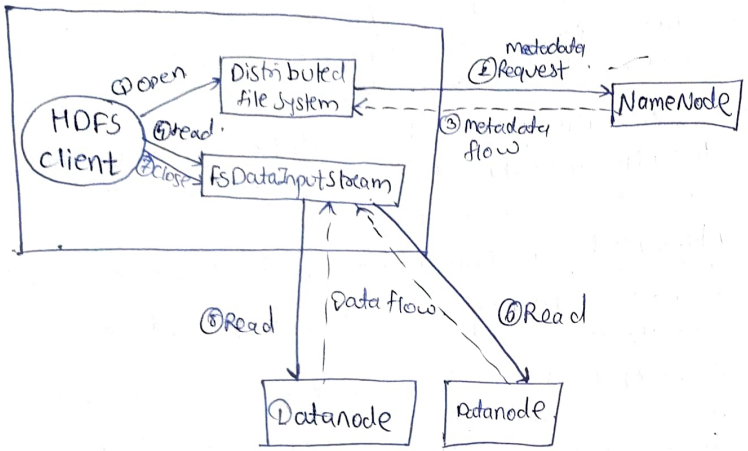
- The Hadoop core is divided into two fundamental layers : mapreduce engine & HDFS.
- The mapreduce engine runs on the top of HDFS as its data storage manager.
- HDFS is designed to run on commodity hardware, which is high fault-tolerant, scalable & efficient.



- HDFS is a block-structured file system where each file is divided into block of pre-determined size. These blocks are stored across a cluster of one or several machines
- HDFS follows master/slave architecture, where cluster comprises of a single Namenode (master node) & multiple dataNode (slave Node)
- To store the file in this architecture HDFS splits the file into fixed-sized blocks (eg, 64 mb) & stores them on workers (DataNodes)
- NameNode is the master node in hadoop that maintains & manages the blocks present on the DataNodes
- NameNode is very highly available server that manages the file system Namespace & control access to file by client.
- DataNodes are the nodes that act as slave in HDFS architecture which are cheap & of low quality.
- Checkpoint is record of the image, which is stored on local hosts that allows recovery.

Q] Explain the anatomy of file read & write in HDFS.

→ Anatomy of file read operation



Steps

① Open
- The client initates a requests to opena file stored in HDFS.

② Request metadata.
- The client sends a request to the NameNode to get metadata of the file (block location, ID)

③ metadata flow.
- The NameNode Responses with.
  - A list of block IDS
  - A DataNodes that holds replicas of each block.

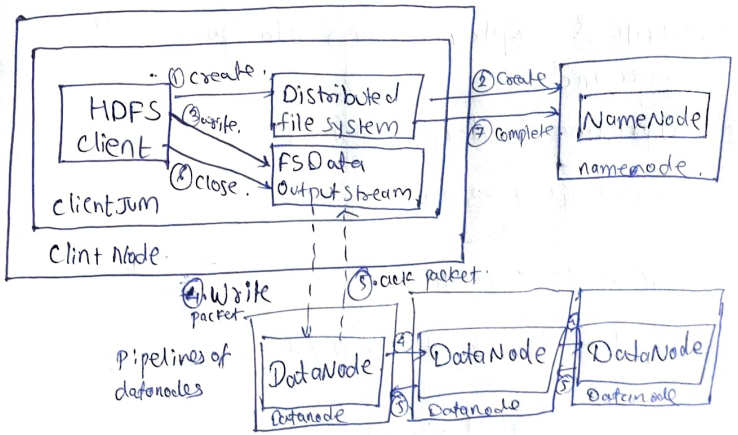
④ Read.
- The Client prepares to read the file using the block & DataNode information

⑤ Read DataNodes.
- for each block, the client selects the nearest/ optimal DataNodes to begin reading.

⑥ Reads from DataNodes.
- The Client opens a connections to the chosen Datanodes & stard reading block by block.

⑦ Close.
- Once all the blocks are read, the client closes the files.

★ Anatomy for file write operation ★



Steps

1. Client calls create() on Distributed File System to create a file.
2. An RPC call to namenode happens through the DFS to create new file
3. As client writes the data, the data is splited into packets by DFS output stream, which is then writes to an internal queue, called data queue.
4. Datastreamer sends the packet to first Datanode in pipeline. It stores packet & forward it to second DataNode in pipeline
5. In additional to the internal queue, DFS outputstream also manages "Ackqueue" of the packets that are waiting for acknowledged by DataNodes
6. When the client finishes writing the file, it calls() close() on the stream.

Q7) Write & explain ~~of~~ 5 Hadoop shell commands.

→ 1) printing hadoop version

→ hadoop version.

2) Create Directory.

→ hadoop fs — mkdir /path/dir_name.

3) See content of a file.

→ hadoop fs-cat <path (Filename)>

4) Copy file from source of destination

→ hadoop fs -cp <source> <dest>

5) move file from source to destination.

→ hadoop fs-mv <src> <destination>

6) Remove a file or directory in HDFS

→ hadoop fs-rm <arg>.

Q) Explain mapReduce with proper diagram & example.

→ mapReduce is the process of breaking the large datasets into individual tasks that can be executed in parallel across a cluster. The results of tasks can be joined together to compute final results.

- The mapReduce algorithm contains two important tasks — Map & Reduce.

1) Map task converts the set of data into key-value pair.

2) Reduce tasks shuffles the ~~value~~ key-value pairs & combines the similar data to produce desire result.

eg → Lets take eg of animal dataset.

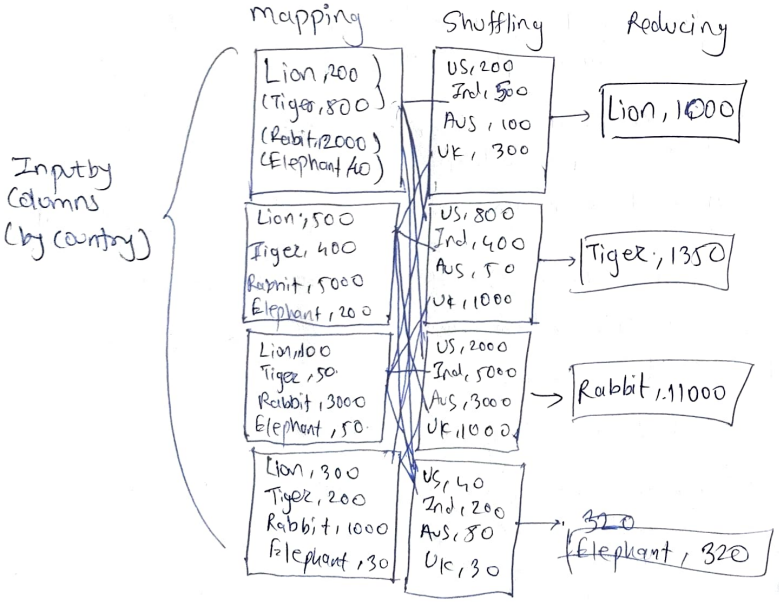
Stage1] US : (Lion, 200), (Tiger, 800), (Rabbit 2000), (Elephant, 40)
Ind : (Lion, 500), (Tiger, 400), (Rabbit, 5000), (Elephant, 200)
Aus: (Lion, 100), (Tiger, 50), (Rabbit, 3000), (Elephant, 10)
UK: (Lion, 300), (Tiger, 200), (Rabbit, 1000), (Elephant, 50)

Step 2 Shuffling by Animal category.

1) Lion = (US, 200), (Ind, 500), (AUS, 100), (UK, 300)
2) Tiger = (US, 800), (Ind, 400), (AUS, 50), (UK, 200)
3) Rabbit = (US, 2000), (Ind, 5000), (AUS, 3000), (UK, 1000)
4) Elephant = (US, 40), (Ind, 200), (AUS, 50), (UK, 30)

Step 3 : Reduce by animal category.

1) Lion : 1400
2) Tiger : 200
3) Rabbit : 11000
4) Elephant, 75.



Q10] Explain Role Job tracker and Task tracker in Hadoop Architecture.

→ * Function of Job Tracker:

- There is single job tracker that runs on master node. It is driver for map-reduce jobs.

1) Accepts job from client & divides into tasks
2) Schedule tasks on worker nodes called task trackers
3) Keeps heartbeat info from task trackers on worker nodes
4) Reschedule tasks on alternate worker if a worker fails

* Function of task tracker:

- Task tracker runs on each worker node & there are as many task trackers as worker node.

f^n of task tracker are -

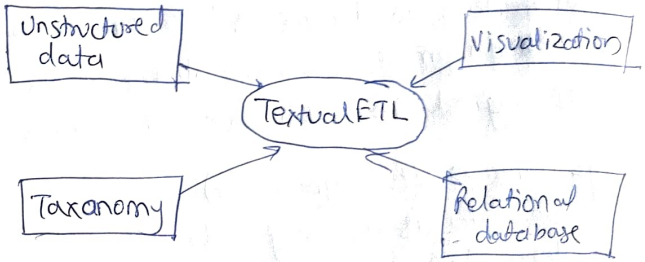
a) Take assignment from job tracker.
b) Executes the tasks locally
c) Each worker has specific number of mapper and reducers task it can take at one time.
d) Task assigned are run in parallel
e) Normally they can take more map jobs than reduce tasks.
f) Task tracker sends a regular heartbeat signal to job tracker indicating its status.
g) Task tracker does ~~tract~~ task-attempt before executing task.

Q] Explain difference between pig & Hive

Sr No	Pig	Hive
1)	Pig used for data transformations & processing	Hive used for ware housing & querying data.
2)	Pig works on structured, semi structured & unstructured data	2) Hive works only on structured data.
3)	Pig does not support web interface	3) Hive ~~works~~ support web interface.
4)	Pig uses language called Pig Latin which is similar to SQL	4) Hive uses language called HiveQL which is similar to SQL
5)	Pig supports Avro file format	5) Hive does not supports Avro file format
6)	Creating schema is not required to store data in Pig	6) Hive supports schema.
7)	Pig Loads data quickly	7) Hive takes time to load but execute quickly
8)	Pig works on client-side of server	8) Hive works on server-side of the cluster
9)	Used for programming	9) Used for Reporting.

Q] Explain ETL processing in Big Data.

- The components of textual ETL (Extract, Transform, Load) processing are Textual ETL rule engine.
- The textual ETL rule Engine takes large unstructured data & parse it to extract value for integration. Rules engines contains series of data processing steps & algorithms such as classification, clustering.
- Taxanomy integration, master data integration, metadata integration are some ~~steps~~ processing techniques available in rules engine.



- UI helps in creating & supply processing rules through drag & drop & free from text interface. in different language.
- Taxanomy are required that of several categories of multi-structures and multi-hierarchical data.
- Output database in textual ETL is any RDBMS or NOSQL db. To integrate structured & unstructured database uses key value pair.

Q] Define NoSQL, Describe the various types of NoSQL databases with example and also compare them.

→ NoSQL stands for Notonly SQL. it's a type of database ~~that~~ that store data that may not fit properly into table. They are faster and more flexible.

Types of NoSQL databases.

1) Document-based (e.g., MongoDB, CouchDB)
   - stores data in documents (like JSON or BSON)
   - Each document can have different structure
   - It is good for semi-structured data & flexibility

2) Key-Value stores (DynamoDB)
   - Store data as its key & its value (like a dictionary)
   - Very fast & simple.
   - Best for caching

3) Column-family stores
   - Stores data in columns instead of rows
   - Optimized for read write operation on large datasets
   - Used in analytics & ~~querying~~ time-series data
   - eg → Cassandra, HBase.

4] Graph database
   - → Stores data as nodes and relationships like a network
   - Good at managing and querying relationships.

Feature	Document DB	Key-Value pair	Column DB	Graph DB
Data Model	JSON-like docs	Key-value pairs	column families	Nodes & Edges
Flexibility	High	Very high	Medium	High
Speed	High	Very high	~~Medium~~ High	Medium-High
Schema	Schema-less	Schema-less	Flexible	Schema-less
Relationships	Limited	None	Limited	Strong Support
Example	Mongo DB	DynamoDB	Cassandra, HBase	ArangoDB

Q] Write short note on YARN & Map-Reduce technique. Is there any difference between both.

→ • YARN stands for Yet another resource negotiator. It is responsible for managing resources like CPU memory and running different jobs on group of computers called Cluster.

What does YARN do?

- YARN assigns tasks to different computers
- It checks how many resources are available
- It allows many types of application to run not just map reduce
- It makes hadoop more flexible and powerful

Components of YARN?

1) Resource Manager – Decides who gets how much CPU & memory
2) Node Manager → Monitors & controls each computer
3) Application Master → Handles one specific job

Map Reduce

- MapReduce is the process or method of dividing big data into smaller chunks and process them parallelly.

How it works?

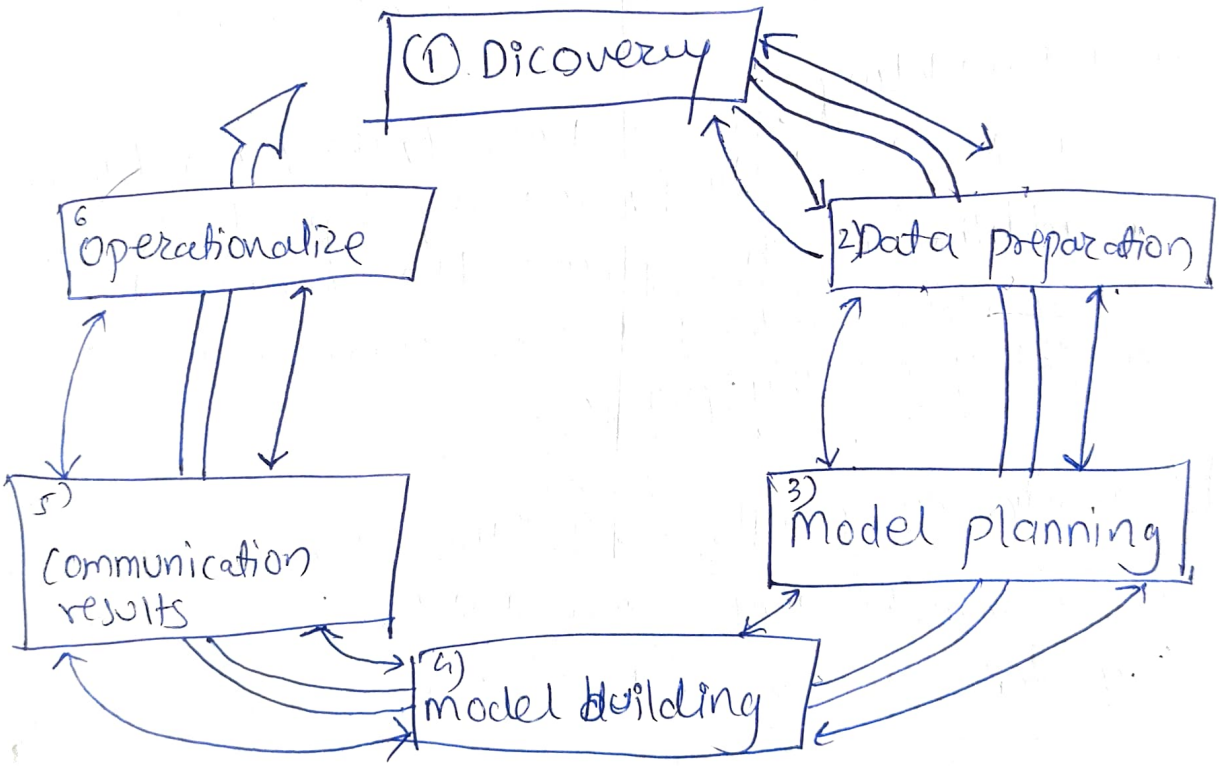
1) Map → Breaks Big data into smaller chunks & process parallelly
2) Reduce → Combine steps of map to give final o/p

YARN	Map Reduce
It's like a manager	It's like a worker
manages jobs & resources	Processes the actual data
Can run MapReduce, Spark, Tez etc.	Can only run MapReduce
Not dependent on MapReduce	Can run on top of YARN

UNIT-4 Big Data Analytics

Q1] Explain different steps in data analytics project life cycle.

→ • The data analytics lifecycle is designed for Big Data problems and data science projects. It have majorly six phases.



1) Discovery → In phase 1 the team learns the buisness domain, including history such as whether the organization has attempted similar projects in the past from which they can learn. The team states the resources available for support in terms of, people, Tech stack, time & data.

2] Data preparation → phase 2 requires the presence of analytics sandbox, in which the team can work with data & perform analytics for the duration of the project. The team needs to extract, Transform & Load to get data into the sandbox.

3] Model planning → The team decides method, technology & workflow for model building phase.

4] Model building → In phase 4, the team develops datasets for testing, training & production purpose. In addition in this phase the team builds & executes model based on work done in the model planning phase.

5] Communicate results → In this phase 5, the team communicates with stakeholder whether the requirements are satisfied or not.

6) Operationalize → In phase 6, the team delivers final reports, ~~briefings~~ code, tech stack including the deployment to their environment, accordingly.

8] Explain different kinds of big data analysis

→ There are many types of data analysis. Some of them are more basic in nature such as descriptive, predictive, prescriptive & some are more specific like qualitative analysis & quantitative analysis.

1] Descriptive Analysis

- Descriptive Analysis answers questions about event that have already occured.
- The simplest form of analytics & typically answer questions such as.
  a) How many units of items are sold?
  b) How many patient died with cancer?
  c) How many calls did you receive?
- This type of analysis is done using database queries or simplest spreadsheet ~~filters~~ filters.

2) Diagnostic Analysis

- Diagnostic Analysis is done to find out cause of a phenomenon behind event
- This analytics goes deeper to provide information that can be used to fix event.
- eg →
  a) Why sales a is lower than Sales b?
  b) why people falling ill in XYZ area?

3) Predictive Analysis

- Predictive analytics is carried out to forecast and predict future events
- Predictive data models are carefully created that can based on future predictions based on the past events.

eg → a) what would be improved life expendancy if choosing medicine A over medicine B
b) what would be sales for next year?

4) Prescriptive Analysis

- Prescriptive Analysis takes the analysis from predective analysis and add the note or additional in human judgement to prescribe or advice.

eg → a) what should you do to delay Cancer?
b) what is best time to ~~leave~~ ~~home~~ leave home to reach airport?

5] Exploratory Analysis

- Helps you find Patterns or unknown relationship in data.
- These Kind of analysis is also called hypothesis-generating because rather than testing hypothesis you are looking for pattern that would support hypothesis.

Q] What is data ingestion? How data can be ingested in python? write syntax in python for same.

→ • Data ingestion is means bringing data into your program or system from different resources so you can analyse or use it.

- Data ingestion in pandas refers to shifting the data from a variety of sources into the Pandas Dataframe structures.
- Types of data are → 1) structured data
  2) Semi-structured data
  3) Unstructured data.

1) Structured data ingestion from csv

```
import pandas as pd
df = pd.read_csv("data.csv")
print(df.head())
```

2) Semi-structured data ingestion from JSON file

```
import pandas as pd
df = pd.read_json("data.json")
print(df.head())
```

3) Unstructure data ingestion from text file

```
with open("file.txt", 'r') as file:
    text = file.read()
print(text[:100])
```

Q] Explain data standardization.

→ • Data standardization is process of converting different datasets structures into one common format of data

- Data standardization converts data into standard format that computer can read & understand. It is important because different sources use different format.
- Machine Learning perform better when data in consistent.
- Data standardization is also essential for preserving data quality. When data is standardized, it is much easier to detect error and ensure that it is accurate.
- Without data standardization it would be very hard to exchange the data and communicate information.

Q] Explain different data transformation techniques

→ 1] Smoothing: Reduce noise or fluctuations in the data by applying a smoothing technique.

- Smoothing technique applies moving average to smooth out irregularities
- This technique includes regression & clustering.

2] Aggregation: Combines multiple data points into a summary metric eg- mean, sum, count.

- Group data by a certain feature & calculate summary statistics.
- It can be used to analyse the tends over time.

3] Generalization: Reduces the level of detail in the data to make it more abstract & less specific

- In Generalization of data Low Level data is ~~co~~ replaced to higher-level concepts through use of concept hierarchies.

4) Normalization:

- Rescales features to a fixed range, usually between [0,1]

$$X_{norm} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

- ~~Can~~ Useful in Neural network or K-mean clustering

~~5] Standardization (Z-score Scaling)~~

5] One-Hot Encoding:

- Converts categorical variables into a series of binary variable.

6) Missing Value Imputation:

- Replace missing or null values with estimates.
- There are certain methods and those are -
  - mean / median / mode
  - K-nearest Neighbors.
- Helps in filling NAN values as many Machine Learning algorithms cannot work with NAN values.

Q] Explain mean, mode & variance & standard deviation with suitable example.

→ 1) mean → Sum of all divided by no of values

$$\bar{X} = \frac{\sum x_i}{N}$$

eg → [10, 20, 30, 20, 40]

$$= \frac{10+20+30+20+40}{6}$$

$$= 20$$

2) mode → Value that appears most frequently.

Here → 20

3] Variance

→ measure of how far value is from mean

→

$$\sigma^2 = \frac{\sum (x_i - \bar{x})^2}{N} \qquad \sigma^2 = \frac{\sum (x_i - \mu)^2}{N}$$

4] Standard Deviation:

Shows how spread the data is.

$$\sigma = \sqrt{\sigma^2}$$

Q] Draw and explain architecture of HIVE.

→ Hive is a datawarehouse tool build on top of hadoop. It helps you to analyse big data using a language similar to SQL called HiveQL

JDBC/ODBC

Major components of Hive Architecture.

1] Hive Clients → There are three main type of Hive clients JDBC, ODBC & Thrift Client. One can choose their own according to need of application.

2] User Interface → The user interface is for users to submit Hive queries & other operations to system as of now command Line interface & web GUI are there.

3] Driver → It gets queries from the Hive client via user Interface. It passes the query to compiler to get execution details/ Plan. It monitors the progress of various life cycle and it stores the metadata that is generated while executing the HiveQL statement.

4] Compiler → The compiler is assigned with the task of converting a HiveQL query into a MapReduce input. It parse the query and does the analysis on the blocks of query & generates the execution plan.

5] Metastore → It is repository of metadata. This metadata consists of data for each table like its location & schema. The metadata keeps track of the data, replicates it & provides a backup in case of data loss.

6] Optimizer →

- It improves the query execution plan so that it run faster & user fewer resources.
- Improves efficiency & scalability.

Q] How missing values are filled in pandas Data frame with the zero? Assume suitable data.

→ In pandas missing values are filled in 0 with the method of fillna() method. To fill all missing values with the zero.

```
df.fillna(0)

import pandas as pd
import numpy as np

data = {
        'Name' : ['Alice', 'NIKKI', 'Pratik', 'Roshan', 'Ram']
        'Marks': [ 'NaN', '29', '28', 'NAN', '26']
        'Age' : [20, 21, np.NaN, 17, np.NAN]

df = pd.Dataframe (data)
df.

o/p.

df_filled = df.fillna(0)
print (df_filled)
```

Q] Explain three ways of Data Normalization used in data analytics

→ Data Normalization is process of scaling numeric data into common range, making it easier for machine learning algorithm to learn efficiently & perform accurately

1] Min max Normalization.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Scales data between. 0 & 1

eg $X = 70$, min = 50, max 100

$$= \frac{70 - 50}{100 - 50} = \frac{20}{50} = 0.4$$

2] Z-score Normalization

$$X_{std} = \frac{X - \mu}{\sigma}$$

μ = means of the feature.
σ = Standard deviation

- Convert data into a standard normal distribution (mean = 0, std = 1)

eg → $X = 70$, mean = 60, std = 5.

$$X_{std} = \frac{70 - 60}{5} = 2.0$$

3. Decimal Scaling

$$X_{norm} = \frac{X}{10^j}$$

move decimal point to scale the data.

eg if $X = 500$, $j = 3$

$$X_{norm} = \frac{500}{10^3} = 0.5$$

Q] what is categorical variable? Why do you need categorical variable encoding? With an example, explain one-hot encoding

→ A categorical variable is a variable that contains labels or categories instead of numeric value. These labels represent distinct group or classes.

Example

Gender : Male, Female
Color : Red, Blue, Green
City : Pune, Mumbai, Delhi

These values cannot be directly used in most machine learning because they are non-numeric & have no inherit order.

Types of Categorical variables

1) Nominal Categorical Variables

- No meaningful order or ranking
- Example → Color → Red, Blue, Green.
- No value is greater or less.

2) Ordinal categorical variable :

- Have an inherit order or ranking

eg → Size → Small < medium < large.

Why Need Encoding ?

Most machine learning models (like linear regression, SVM, neural networks) require numeric inputs. Hence, we must convert categorical variable into numerical format so that algorithm can use them.

One-Hot Encoding

One-hot encoding converts each category into a separate column with binary value (0 or 1)

example :-

Age	Eye color
15	Blue
19	Black
22	Brown.

Age	Blue	Black	Brown
15	1	0	0
19	0	1	0
22	0	0	1

1 → Matching Category
0 → No Match

Q] What is data wrangling? Why do you need it? explain data wrangling methods?

→ Data wrangling also known as data munging, it is the process of cleaning, organizing, & transforming raw data into a structured and usable format for analysis & modeling.

- It involves handling missing value, removing duplicates, correcting errors, changing formats & more.

Why do we need data wrangling ?

Data in real world is often

1) Incorrect (typos, wrong entities)
2) Incomplete (missing values).
3) Inconsistent (mixed format)
4) Unstructured (text, JSON)

Results → Inconsistency, Fail to predict.

Therefore Data Wrangling ensures.

- Data quality & Consistency

There are 6 main steps →

1) Discovery → Before going deeply, we must better understand what is in data, which will inform how to analyze it. How to wrangle customer data. for ex → what they bought, location what they received.
   - To understand raw data, its structure, quality & issues.

2) Structuring → This means organization of data. which is ~~ness~~ necessary because raw data comes in many different shapes & sizes.
   - To convert unstructured or semi-structured data into usable tabular format.
   - It is done for easier computation & analysis

3) Cleaning → Removing duplicates, handles missing data, fix types.
   eg → Replacing all NAN with mean in table

4) Enriching → ~~Add the addit~~ Enhance the dataset by adding useful information from internal or external sources

5] Validating → It checks data quality, consistency or security. Ensures data follows the expected format/type

6] Publishing → store or export the clean, structured data for analysis or machine learning data.
   - Data is Ready to use.

Q] Comparision HBase vs Hive.

Parameter	HBase	Hive
Type	1) NOSQL	1) SQL-like. data warehouse
Data Storage	2) Store data in HDFS in columnar format	2) Store data in HDFS in tabular format.
Data Model	Schema-less	fixed schema
Query Language	HBase Shell	HiveQL
Latency	Low	High
Data updates	Supports updates	Not ideal for frequent update
Integration	Apache phoenix, Spark	Tez, spark, MapReduce
Schema	Does not required predefined schema.	defined Required schema.
Structure	Complex	Easy.
eg →	Real time ex → Last entry	Big data analysis month entries.