Sub: Big Data Analytics

Subcode: 18CS72

Staff Name: Prof. Yasmeen. Shaikh

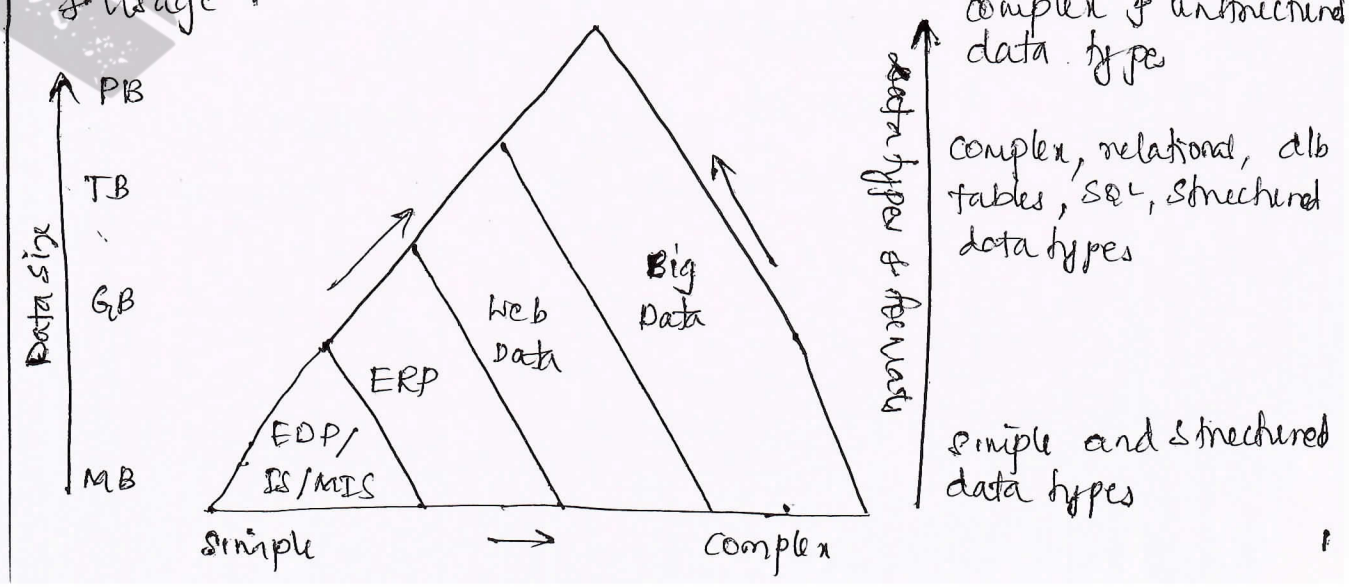
Question Paper: 7th Sem BE Exam, Feb/Mar 2022

MODULE - 1

1a) Discuss the evolution of Big Data.

→ The rise in technology has led to the production & storage of voluminous amount of data.

- Earlier megabytes were used but nowadays petabytes are used for processing, analysis, discovering new facts & generating new knowledge.
- Conventional systems for storage, processing & analysis pose challenges in large growth in volume of data; variety of data, various forms & formats; increasing complexity; faster generation of data & need of quickly processing, analyzing & usage.



1b Explain the characteristics of Big Data.

→ Characteristics of Big data are

i) Volume - Big data contains term big, which is related to size of data & hence the characteristic
- size defines the amount or quantity of data, which is generated from application.
- The size determines the processing considerations needed for handling that data.

ii) Velocity - The term velocity refers to the speed of generation of data.
velocity is a measure of how fast the data generates & processes.

iii) Variety - Big data comprises of a variety of data.
- data is generated from multiple sources in a system.
- data consists of various forms & formats.
- The variety is due to the availability of large number of heterogeneous platforms in the industry.

iv) Veracity - is also considered as an important characteristic to take into account the quality of data captured, which can vary greatly, affecting it accurate analysis.

1C. With a neat block diagram, explain data Architecture design.

→

Layer 5 Data Consumption	Export of datasets to cloud web, etc	Data sets usage: BPs, BIs, knowledge discovery	Analytic (real-time, near real-time, scheduled batches) reporting, visualization
Layer 4 Data Processing	processing technology: Map Reduce, Hive, Pig, Spark	Processing is real-time, scheduled batches or hybrid	Synchronous or asynchronous processing
Layer 3 Data Storage	Consideration of types, formats, compression frequency of incoming data, patterns of querying & data consumption	HDFS (Scalable, self managing & self healing), Spark, Mesos or S3	NoSQL data stores - Hbase, MongoDB, Cassandra, Graph database
Layer 2 Data Ingestion & acquisition	Ingestion using Extract Load & Transform (ETL)	Data semantic (replace, append, fuse, aggregate, compact)	Pre-processing (validation, transformation or transcoding) requirement	Ingestion of data, from sources in batches to real time
Layer 1 Identification of internal & external sources of data	Sources for ingestion of data	Push or pull of data from the sources of ingestion	Data types for databases, files, Web or service	Data formats: structured, semi or unstructured for ingestion

- Data analytics need the number of sequential steps.
- Big data architecture design task simplifies when using the logical layers approach.
- figure above shows the logical layer & the functions which are considered in Big data Architecture.

(Layer 1) considers the following aspects in a design

- Amount of data needed at ingestion layer 2 (L2)
- Push from L1 or pull by L2 as per the mechanism for the usage
- Source data - types: database, files, web or service
- Source formats; i.e. semi-structured, unstructured or structured.

L_2 considers the following aspects:
- Ingestion & ETL processing either in real time, which means store & use the data as generated, or in batches,

L_3 - considers the following aspects:
- Data storage type, format compression, incoming data frequency, querying patterns, & consumption requirements for L_4 & L_5.
- Data storage using HDFS or NoSQL data stores - HBase, Cassandra, MongoDB.

L_4 - considers the following aspects:
- Data processing software such as MapReduce, Hive, Pig, Spark, Spark Shark, Spark Streaming
- Processing is scheduled batches or real time or hybrid.
- processing as per synchronous or asynchronous requirements at L_5.

L_5 - considers the consumption of data for the following
- Data integration
- Dataset usages for reporting & visualization
- Analytics (real time, near real time, scheduled batches), BPs, BIs, knowledge discovery.
- Export of datasets to cloud, web or other systems.

Q9) Write notes on Analytics Scalability to Big Data & massive parallel processing platforms.

→ i) Analytics Scalability to Big Data.

- Scalability enables increase or decrease in the capacity of data storage, processing and analytics
- vertical scalability means scaling up the given systems resources & increasing the systems analytics, reporting & visualization capabilities.
  - This is an additional way to solve problems of greater complexities
  - scaling up means designing the algorithm according to the architecture that uses resources efficiently.
- Horizontal scalability means increasing the number of systems working in coherence & scaling out the workload.
- Processing different datasets of a large dataset deploys horizontal scalability.
- scaling out means using more resources & distributing the processing & storage tasks in parallel.
- The easiest way to scale up & scale out execution of analytics s/w is to implement it on a bigger machine with more CPUs for greater volume, velocity, variety & complexity of data.

The system will perform better on a bigger machine.

However, buying faster CPUs, bigger & faster RAM modules & harddisks, faster & bigger mother board will be expensive compared to extra performance achieved by efficient design of algorithms.

ii) Massively Parallel processing platforms

- Scaling uses parallel processing systems.
- Many programs are so large and/or complex that, especially in limited computer memory.
- Here, it is required to enhance (scale) up the computer system or use massive parallel processing (MPP) platforms.
- parallelization of tasks can be done at several levels
  i) distributing separate tasks onto separate threads on the same CPU
  ii) distributing separate tasks onto separate CPUs on the same computer.
  iii) distributing separate tasks onto separate computers.
- multiple compute resources are used in parallel processing systems.
- The computational problem is broken into discrete pieces of sub-tasks that can be processed simultaneously
- The system executes multiple program instructions or sub-tasks at any moment in time.

Total time taken will be much less than with a single compute resource.

i) Distributed computing model.

- A distributed computing model uses cloud, grid or clusters, which process & analyze big & large datasets on distributed computing nodes connected by high-speed n/w's.

- Big data processing uses a parallel, scalable & no-sharing program model, such as mapReduce for computations on it.

2b) Highlight Big Data Analytics applications with one case study.

→ Big data analytics applications are

1) marketing & sales
2) detection of marketing frauds
3) Big data Risks
4) Credit Risk management
5) Algorithmic Trading
6) Healthcare
7) Medicine
8) Advertising

1) Big Data in marketing and sales

- Data are important for most aspect of marketing, sales & advertising.
- Big data analytics deploy large volume of data to identify & derive intelligence using predictive models about the individuals.

2) Big data Analytics in detection of marketing frauds

- Fraud detection is vital to prevent financial loses to users.
- Fraud means someone deceiving deliberately.
- Big data analytics enable fraud detecting.

3) Big Data Risks

- Large volume and velocity of Big data provide greater insights but also associate risks with the data used.
- Data included may be erroneous, un accurate or far from reality.
- Analytics introduces new errors due to such data.
- Big data can harm potential harm to individuals. A company may suffer financial losses.

4) Big data Credit Risk Management

- Financial institutions, such as banks, extend loans to industrial & household sectors.
- These institutions in many countries face credit risks mainly risks of i) loan defaults ii) timely return of interest & principal amount
- The insights using Big data decreases the default rate in screening of loan, greater accuracy in issuing credit & faster identification of the non-payment or fraud issues of the loan receiving entities.

5) Big data & Algorithmic Trading

- Algorithmic trading is a method of executing a large order using automated pre-programmed

trading instructions according for variables such as time, price & volume.

- Complex mathematical computations enable algorithmic trading & business investment decisions to buy & sell.
- The input data are insights gathered from the risk analytics of market data.
- Big data bigger volume, velocity & variety in the trading provide an edge over other trading entities.

6) Big Data & Healthcare

Big data analytics in healthcare use the following data source i) clinical records ii) pharmacy records iii) electronic medical records iv) diagnosis logs & notes v) Additional data such as deviations from person usual activities, medical leaves from job, social interactions.

Healthcare analytics using Big data can facilitate customer-centric healthcare, prevent fraud, waste, abuse in healthcare industry & reduce health cost, monitoring patients in real-time etc.

7) Big data in medicine

Big data analytics deploys large volume of data to identify & derive intelligence using predictive models about individuals.

- Big data driven approaches help in research in medicine which can help patients.
- Big data offers potential to transform medicine & healthcare system.

8) Big data in Advertising

- Big data technology & analytics provide insights patterns & models, which relate the media exposure of all consumers to the purchase activity of all consumers using multiple digital channels.
- Big data help in identity management & can provide an advertising mix for better branding exercises

Case study

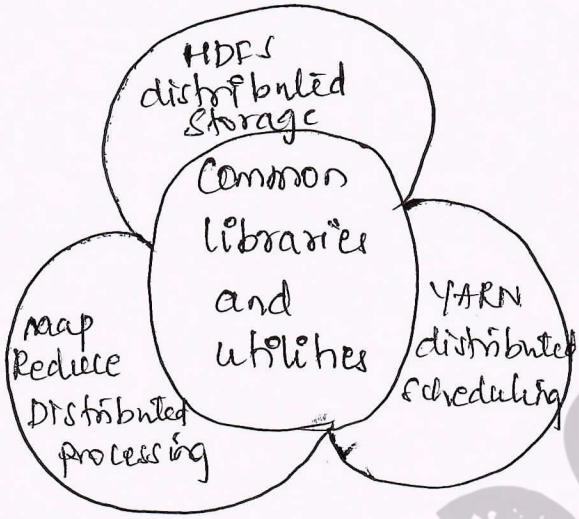
Consider a travel agency website offers search results for flights between two destination A & C which do not connect directly.

- The search shows the results in order of increasing travel cost through stopover at an intermediate airport B.
- The customers find uncomfortable solutions - i.e results generated are mechanical without intelligence & optimization.
- The searches show the cheaper options but sometimes show results such as customer would reach C through stopover at B after 8 hours.
- The searches therefore need optimization for parameters of travel cost, multiple intermediate stopovers & airlines that will provide maximum customer convenience as well as cost.
- Big data algorithms & advance analytics techniques enable price optimization for a given product or service & pricing decisions.

MODULE - 2

3a. What are the core components of Hadoop? Explain in brief the each of its components.

→ Figure below shows the core components of Hadoop



The Hadoop core components of the framework are:

1) Hadoop Common – The common module contains the libraries & utilities that are required by the other modules of Hadoop.
   Ex Hadoop common provide various components & interfaces for distributed file system & general I/O. This includes serialization, Java RPC & file based data structures.

2) Hadoop distributed file system (HDFS) – A Java-based distributed file system which can store all kinds of data on the disks at the cluster.

3) MapReduce V1 – software programming model in Hadoop1 using Mappers & Reducers. The V1 processes large datasets of data in parallel & in batches.

4) YARN – software for managing resources for computing. The user application tasks or sub-tasks run in parallel

at the Hadoop, uses scheduling & handles the requests for the resources in distributed running of the tasks

- MapReduce v2 - Hadoop 2 YARN based system for parallel processing of large datasets & distributed processing of the application tasks.

3b) Explain Hadoop distributed file system

→ Hadoop distributed file system (HDFS) stores the data in a distributed manner in order to compute fast.

- The distributed data store in HDFS stores data in any format regardless of schema.
- HDFS provides high throughput access to data-centric applications that require large-scale data processing workloads.

i) HDFS data storage

- Hadoop data store, stores data at a number of clusters.

Each cluster has a number of data stores, called racks.

- Each rack stores a number of data nodes.
- Each DataNode has a large number of data blocks.
- The racks distribute across a cluster.
- The nodes have processing & storage capabilities.
- The nodes have the data in data blocks to run the application tasks.

The data blocks replicate by default at least on three DataNodes in same or remote nodes.

- Data at the stores enable running the distributed applications including analytics, data mining, OLAP using the clusters.

- A file containing the data divides into data blocks.

- A data block default size is 64MBs

Hadoop HDFS features are as follows

i) Create, append, delete, rename and attribute modifications functions

ii) Content of individual file cannot be modified or replaced but appended with new data at the end of the file

iii) Write once but read many times during usages and processing

iv) Average file size can be more than 500MB.

Figure below shows the client, master NameNode, Primary & secondary master nodes and slave nodes in the Hadoop physical architecture.

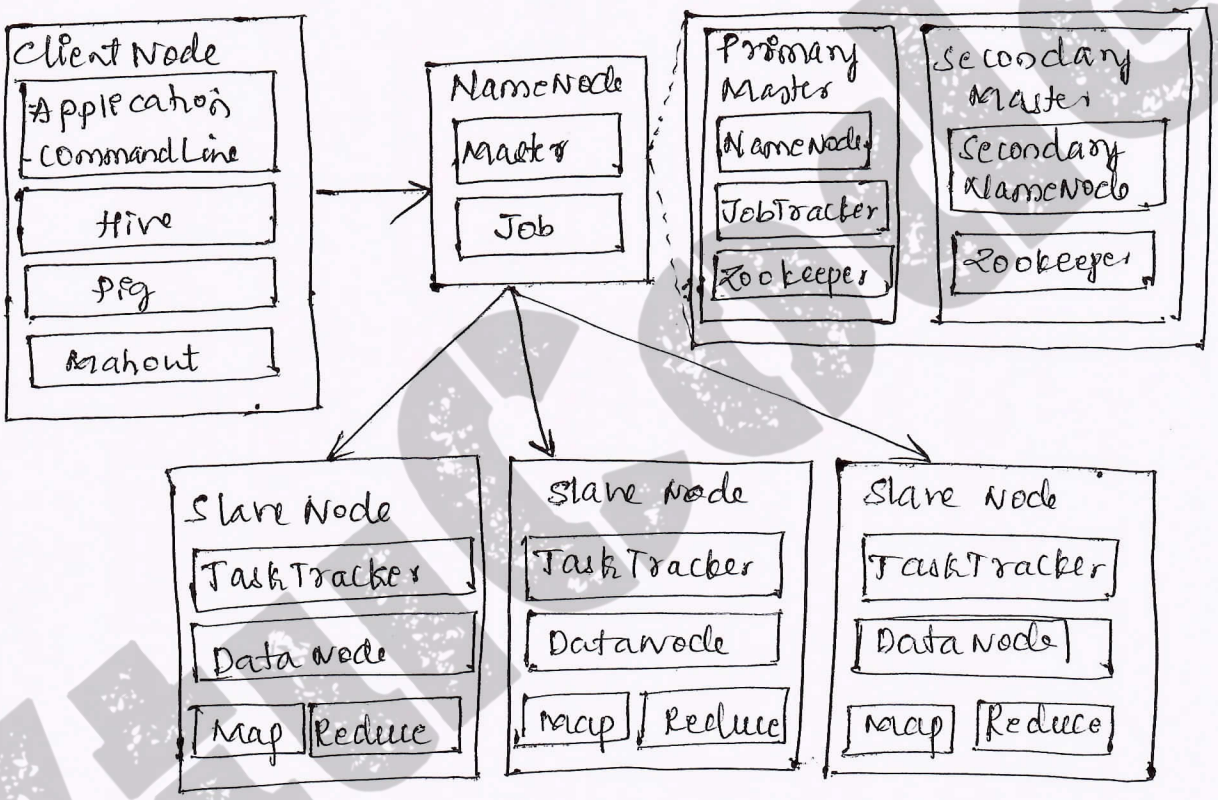
- Clients as the users run the applications with the help of Hadoop ecosystem projects.

- A single Master node provides HDFS, MapReduce, & HBase using threads in small to medium sized clusters.

- When the cluster size is large, multiple servers are used, to balance the load.

The secondary NameNode provides NameNode Management services & Zookeeper is used by HBase for metadata storage.

- The Master Node fundamentally plays the role of a coordinator.
- The MasterNode receives client connections, maintains the description of the global file system namespace & the allocation of file blocks



The Master consists of three components NameNode, Secondary NameNode & JobTracker.

The NameNode stores all the file system related information. Secondary NameNode is alternate for NameNode. It keeps a copy of NameNode metadata. Thus stored metadata can be rebuilt easily, in case of NameNode failure.

The JobTracker coordinates the parallel processing of data.

4a. Define MapReduce framework and its functions.

→ MapReduce is a programming model for distributed computing

- Mapper: means software for doing the assigned task after organizing the data blocks imported using the keys.
- key specifies in the command line of a mapper. The command maps the key to the data, which an application uses.

Reducer: means software for reducing the mapped data by using the aggregation, query or user-specified function.
The reducer provides a concise cohesive response for the application.

Aggregation function means the function that groups the values of multiple rows together to result a single value of more significant meaning or measurement.
ex: functions such as count, sum, maximum, minimum, deviation & std deviation.

Querying function means a function that finds the desired value,
ex: finding for a best student of a class who has shown the best performance in examination

MapReduce allows writing applications to process reliably the huge amounts of data, in parallel

on large clusters of servers.
- The cluster size does not limit large scale data analysis using multiple machines of the cluster.

4b Write down the steps on the request to MapReduce & the types of process in MapReduce.

→ - MapReduce provides two important functions.
- The distribution of job based on client application task or user query to various nodes within a cluster is one function.
- The second function is organizing and reducing the results from each node into a cohesive response to the application or answer to the query.
- The processing tasks are submitted to the Hadoop.
- The Hadoop framework in turn manages the task of issuing jobs, job completion & copying data around the cluster between the datanodes with the help of JobTracker.
- Daemon refers to a highly dedicated program that runs in the background in a system.
- The user does not control or interact with that; an example is MapReduce in Hadoop system.
- MapReduce runs as per assigned Job by JobTracker, which keeps track of the job submitted for execution & runs TaskTracker for tracking the tasks.
- MapReduce programming enables job scheduling & task execution as follows:

- A client node submits a request of an application to the JobTracker.
- A JobTracker is a Hadoop daemon.

The following are the steps on the request to MapReduce

i) estimate the need of resources for processing that request
ii) analyze the states of the slave nodes
iii) place the mapping tasks in queue.
iv) Monitor the progress of task, & on the failure, restart the task on slots of time available.

The job execution is controlled by two types of processes in MapReduce:

1. The Mapper deploys map tasks on the slots
   - Map tasks assign to those nodes where the data for the application is stored.
   - The Reducer output transfers to the client node after the data serialization using AVRO.
2. The Hadoop system sends the Map & Reduce jobs to the appropriate servers in the cluster.
   - The Hadoop framework in turn manages the task of issuing jobs, job completion & copying data around the cluster between the slave nodes.
   - Finally, the cluster collects & reduces the data to obtain the result & send it back to the Hadoop server after completion of the given tasks.

The Job execution is controlled by two types of processes in MapReduce.

- A single master process called JobTracker is one.
- This process coordinates all jobs running on the cluster & assigns map & reduce tasks to run on the TaskTrackers.
- The second is a number of subordinate processes called TaskTrackers.
- These processes run assigned tasks & periodically report the progress to the JobTracker.

4c. Write short notes on Flume Hadoop Tool.

→ Apache Flume provides a distributed, reliable & available service.

- Flume efficiently collects, aggregates and transfers a large amount of streaming data into HDFS.
- Flume enables upload of large files into Hadoop cluster.
- Feature of Flume include robustness & fault tolerance.
- Flume provides data transfer which is reliable & provides recovery in case of failure.
- Flume is useful for transferring a large amount of data in applications related to logs of network traffic, sensor data, geo-location data, emails & social media messages.

Apache Flume has four components

i) Source - which accept data from server or an application

ii) Sink - which receive data & store it in HDFS repository or transmit the data to another source.

iii) Channels - connect between sources & sink by queuing event data for transactions.

iv) Agents - run the sinks & sources in Flume.

MODULE - 3

5a. Discuss the characteristics of NOSQL data store along with the features in NOSQL transactions.

→ Characteristics of NOSQL data store

1) High & easy scalability : NOSQL data stores are designed to expand horizontally.

2) Support to replication : Multiple copies of data store across multiple nodes of a cluster. This ensures high availability, partition, reliability & fault tolerance

3) Distributable : Big-data solutions permit sharding & distribution of shards on multiple clusters which enhances performance & throughput.

4) Usages of NOSQL servers which are less expensive :
- NOSQL data stores require less management efforts.
- It supports features like repair, easier data distribution & simple data models that makes database administrator & tuning requirement less stringent.

5) Usages of open-source tools : NOSQL data stores are cheap & open source.
- Database - implementation is easy & typically use cheap servers to manage the exploding data &

transactions while RDBMS databases are expensive & use big servers & storage systems.

6) Support to schemaless data model :- data can be inserted in NoSQL data store without any predefined schema.

7) Support to integrated caching : This increases system performance. SQL database needs a separate infrastructure for that

8) No flexibility unlike the SQL/RDBMS ; NoSQL DBs are flexible & have no structured way of storing & manipulating data.

Features in NoSQL transactions

i) Relax one or more of the ACID properties

ii) Characterize by 2 out of 3 properties of CAP theorem

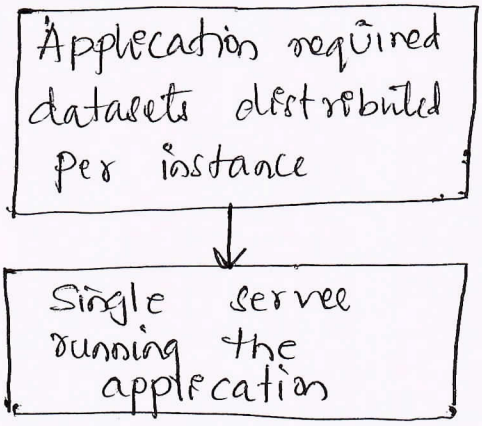
iii) Can be characterized by BASE properties.

5b. With neat diagram, explain the following for Shared-Nothing Architecture for Big Data Tasks.

i) Single-Server Model ii) Sharding very large databases

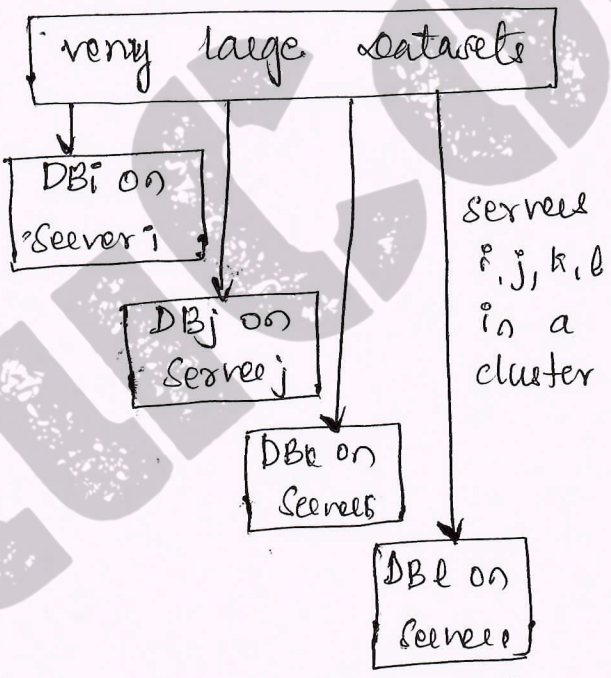
iii) Master slave model iv) Peer to peer model

→ i) Single Server Model



- It is the simplest distribution option for NOSQL data store.
- A graph database processes the relationships between nodes at a server.
- The SSD model suits well for graph DBs.
- Aggregates of datasets be key-value, column-family or Big Table data stores which require sequential processing.
- These data stores also use the SSP model.
- An application executes the data sequentially on a single server.

ii) Sharding very Large Databases

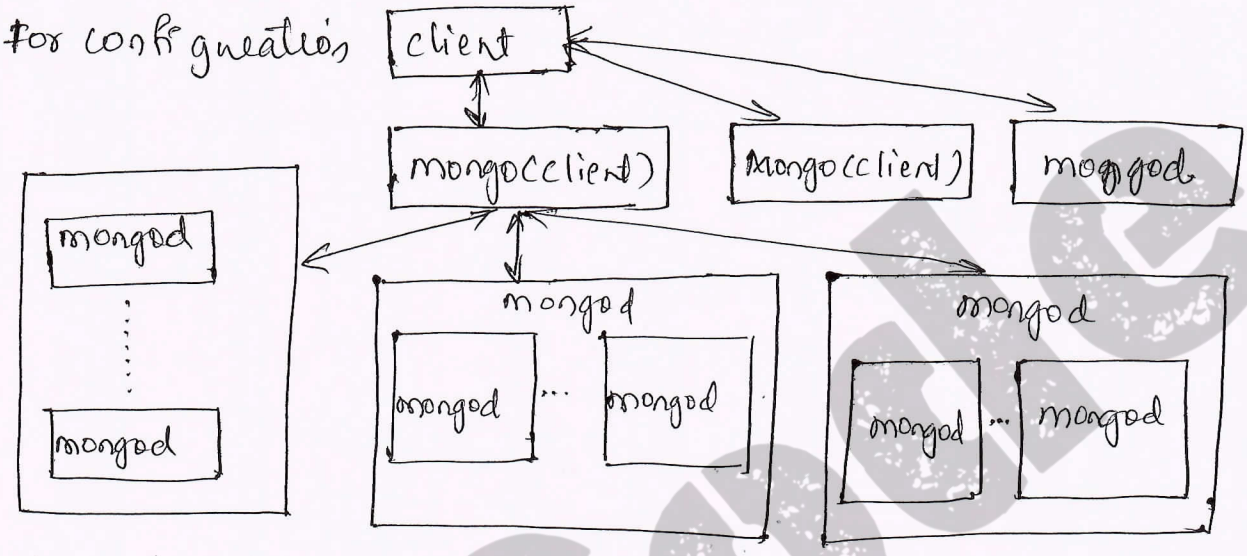


- Figure above shows sharding of very large datasets into four divisions, each running the application on four different servers i, j, k & l at the cluster.
- DB_i, DB_j, DB_k & DB_l are four shards.
- The application programming model is SN architecture is such that an application process runs on multiple shards in parallel.

- Sharding provides horizontal scalability.
- A data store may add an auto-sharding feature.
- The performance improves in CN.

iii) Master-slave Distribution model

For configuration



- A node serves as a master or primary node & the other nodes are slave nodes.
- Master directs the slaves.
- Slave nodes data replicate on multiple slave servers in Master Slave distribution (MSD) model.
- When a process updates the master, it updates the slaves also.
- A process uses the slaves for read operations
- processing performance improves when process runs large datasets distributed onto the slave nodes.

iv) Peer to peer distribution model

- Peer to peer distribution (PPD) model and replication show the following characteristics:

1) All replication nodes accept read request & send the responses.

2) All replicas function equally

3) Node failures do not cause loss of write capability, as other replicated node responds.

- Cassandra adopts the PPD model.
- The data distributes among all the nodes in a cluster.
- Performance can be further enhanced by adding the nodes.
- Since nodes read & write both, a replicated node also has updated data.

∴ the biggest advantage in the model is consistency.

6a) Define key-value store with example. What are the advantages of key-value store?

→ The simplest way to implement a schema-less data store is to use key-value pairs

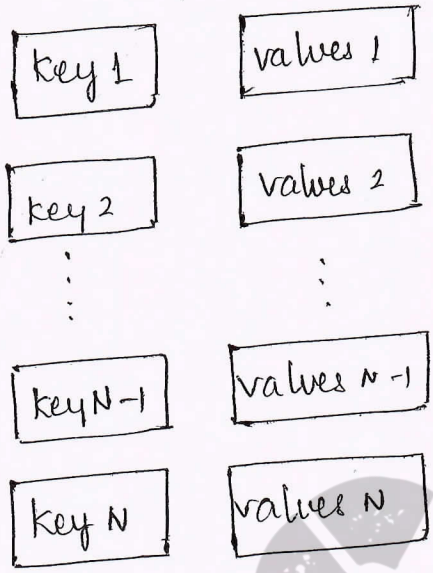
- The data store characteristics are
  i) high performance
  ii) scalability
  iii) flexibility.

Data retrieval is fast in key-value pair data store.

- A simple string called, key maps to a large data string or BLOB.
- key value store accesses use a primary key

for accessing the values.
∴ the store can be easily scaled up for very large data.

Figure below shows key-value pair architectural pattern & example of students data base as key-value pair



key	value
"Ashish"	"category": student; class: BTech; semester: VII; Branch: Engineering; Mobile: 9999988888"
"Mayuri"	"category: student; class: MTech; Mobile: 8888844444"

The advantages of key-value store are as follows

1. Data store can store any data type in a value field.
   The key-value system stores the information as a BLOB of data & return the same BLOB when the data is retrieved.

2. A query just requests the value and returns the values as a single item, values can be of any type
3. key-value store is eventually consistent
4. key-value data store may be hierarchical or may be ordered key-value store.
5. Returned values on queries can be used to connect it to lists, table-columns, data-frame fields & columns.
6. Have i) Scalability ii) Reliability iii) portability iv) low operational cost
7. The key can be synthetic or auto-generated. The key is flexible & can be represented in many formats.

6b Write down the steps to provide client to read and write values using key-value store. What are the typical uses of key value store?

→ The key-value store provides client to read & write values using a key as follows

i) Get(key): returns the value associated with the key

ii) Put(key, value): associates the value with the key & updates a value if this key is already present

iii) Multi-get(key1, key2, ... keyN): returns the list of values associated

with the list of keys.

iv) Delete (key): removes a key & its value from the data store.

The typical uses of key-value store are

i) Image store

ii) Document or file store

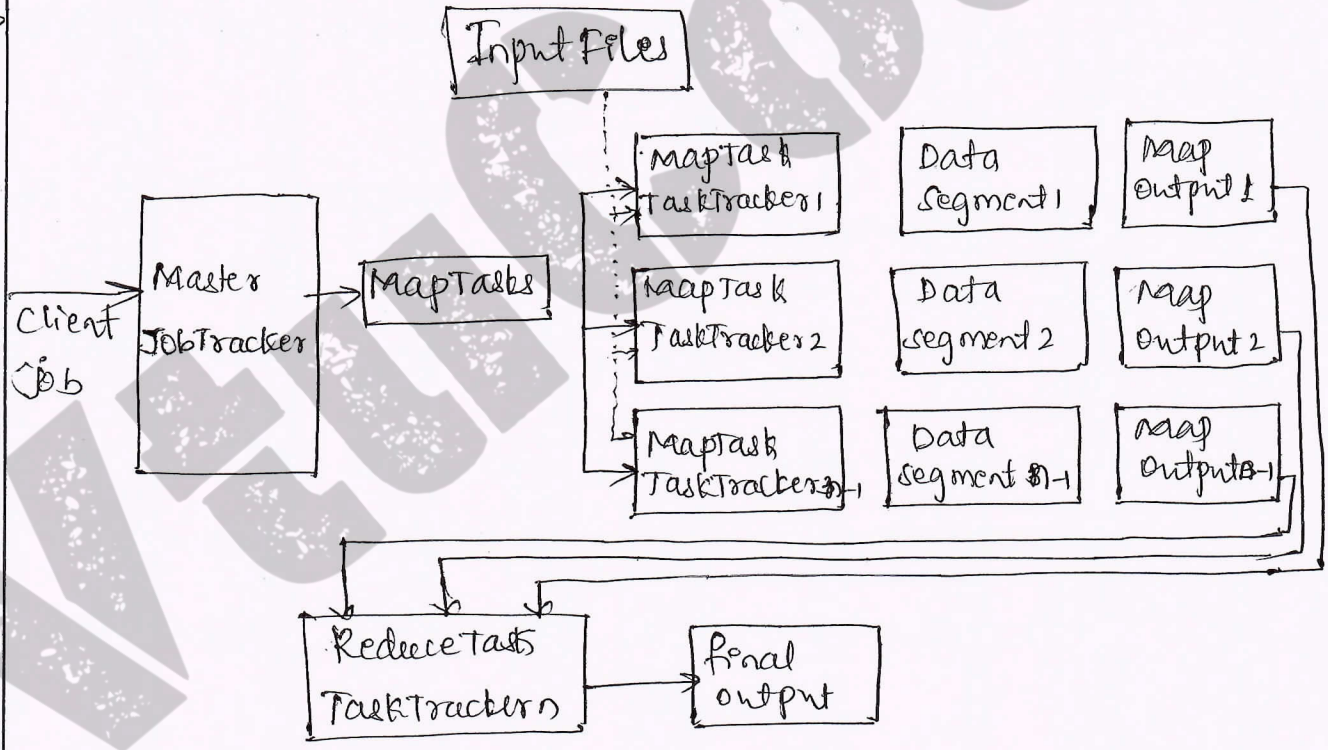
iii) Look up table

iv) Query-cache.

MODULE - 4

7a. With a neat diagram, explain the process in MapReduce when client submitting a job.

→



- A job means a MapReduce program.
- Each job consists of several smaller units, called MapReduce tasks.
- The input data is in the form of an HDFS file.
- The output of the tasks also gets stored in the HDFS

- A user application specifies locations of the input/output data and translates into map & reduce functions.
- The job does implementation of appropriate interfaces and/or abstract classes.
- These & other job parameters, together comprise the job configuration
- The Hadoop job client then submits the job and configuration to the Job Tracker, which then assumes the responsibility of distributing the software/configuration to the slaves by scheduling tasks, monitoring them & provides status & diagnostic information to the jobclient.

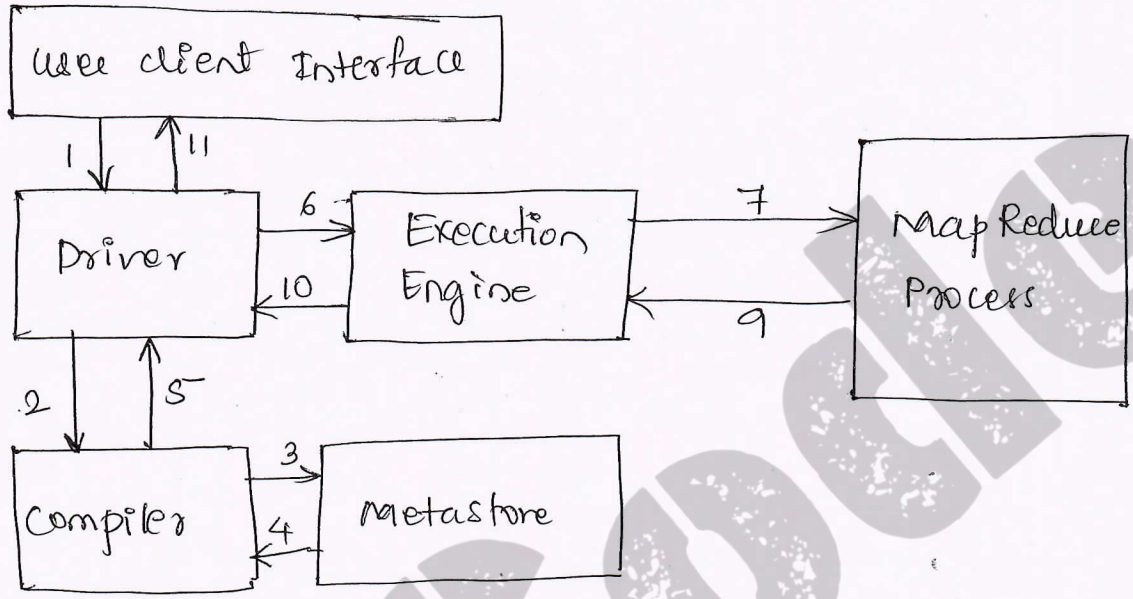
MapReduce consists of a single master JobTracker & one Slave TaskTracker per cluster-node.

The master is responsible for scheduling the component tasks in a job onto the slaves, monitoring them & re-executing the failed tasks.

- The slaves execute the tasks as directed by the master.
- The MapReduce framework operates entirely on key-value pairs.
- The framework views the input to the task as a set of (key, value) pairs & produces a set of (key, value) pairs as the output of the task

16 Explain Hive Integeation's and workflow steps involved with a diageam.

→ - Hive integeates with the MapReduce & HDFS.
- Figure below shows the data flow sequences & workflow steps between Hive & Hadoop



Step 1 to 11 are as follows

1. Execute Query : Hive Interface (CLI or web interface) sends a query to database deiver to execute the query.

2. Get Plan : Deiver sends the query to query compiler that parses the query to check the syntax & query plan or the requirement of the query.

3. Get Metadata : compiler sends metadata request to Metastore

4. Send metadata : Metastore sends metadata as a response to compiler.

5. Send Plan : compiler checks the requirement and resends the plan to deiver. The parsing & compiling of query is complete at this place.

6. Execute Plan: Driver sends the execute plan to execution engine.

7. Execute Job: Internally, the process of execution job is a MapReduce job. The execution engine sends the job to Job Tracker, which is in NameNode & it assigns this job to TaskTracker, which is in DataNode. Then the query executes the job

8. Metadata operations: Meanwhile the execution engine can execute the metadata operations with metastore.

9. Fetch Result: Execution engine receives the results from DataNodes

10. Send Results: Execution engine sends the result to Driver

11. Send Results: Driver sends the results to Hive interface.

8a) Use Hive QL for the following
i) Create a table with partition
ii) Add, rename & drop a partition to a table.

→ i) Create a table with partition.

following command is used to create a table with partition

CREATE [EXTERNAL] TABLE <table name> (<column name 1> <data type 1>, ...) PARTITIONED BY (<column name n> <data type n> [COMMENT <column comment>], ...);

example:

```
CREATE TABLE IF NOT EXISTS toy-airplane
(ProductCategory STRING, ProductId INT,
ProductName STRING, ProdMfgDate YYYY-MM-DD),
PARTITIONED BY (PProductId INT);
```

ii) Add, rename & drop a partition to a table

a) Add a partition to the existing table using the following command:

```
ALTER TABLE <tablename> ADD [IF NOT EXISTS]
PARTITION partition-spec [LOCATION 'location']
partition-spec [LOCATION 'location2'] ....;
```

ex: ALTER TABLE toy-tbl ADD PARTITION
(category = 'Toy-Airplane') location
'/Toy-Airplane/partAirplane';

b) rename a partition in the existing table using the following command

```
ALTER TABLE <tablename> PARTITION
partition-spec RENAME TO PARTITION
partition-spec;
```

ex:

```
ALTER TABLE toy-tbl PARTITION
(category = 'Toy-Airplane') RENAMES TO
PARTITION (name = 'Fighter');
```

c) drop a partition to the existing table using the following command:

ALTER TABLE <table name> DROP [IF EXISTS]
PARTITION partition-spec, PARTITION partition-spec,

ex: ALTER TABLE toy-tbl DROP [IF EXISTS]
PARTITION (category = 'Toy - Airplane');

8b) What is PIG in Big Data ? Explain the features of PIG.

→ Apache developed PIG, which:

- is an abstraction over mapReduce
- is an execution framework for parallel processing
- reduces the complexities of writing a mapReduce program
- is a high-level data flow language.
- is mostly used in HDFS environment.
- perform data manipulation operations at files at data nodes in Hadoop.

Features of PIG:

1) Apache PIG helps programmers write complex data transformations using scripts.
Pig Latin language is very similar to SQL & possess a rich set of built-in operators, such as group, join, filter, limit, order by, parallel, sort & split
- Programmers write scripts using Pig Latin to analyze data.

- The scripts are internally converted to Map & Reduce tasks with the help of the component known as Execution Engine, that accepts the Pig Latin scripts as input & converts these scripts into MapReduce jobs.

ii) Creates user defined functions (UDFs) to write customer functions which are not available in Pig.

- A UDF can be in other programming languages, such as Java, Python, Ruby, Jython, JRuby.
- They easily embed into pig scripts written in Pig Latin.

iii) Process any kind of data, structured, semi-structured or unstructured data, coming from various sources.

iv) Reduces the length of codes using multi-query approach.
Pig code of 10 lines is equal to MapReduce code of 200 lines., Thus processing is very fast

v) Handles inconsistent schema in case of unstructured data as well.

vi) Extracts the data, performs operations on that data & dumps the data in the required format in HDFS.
The operation is called ETL (Extract Transform Load)

vii) Performs automatic optimization of tasks before execution.

viii) Programmers & developers can concentrate on the whole operation without a need to create mapper & reducer tasks separately.

ix) Reads the input data files from HDFS or the data files from other sources such as local file system, stores the intermediate data & writes back the output in HDFS.

x) Pig characteristics are data reading, processing, programming the UDFs in multiple languages & programming multiple queries by fewer codes, This causes fast processing.

xi) Pig derives guidance from four philosophies, live anywhere, take anything, domestic & non as it flying.

This justifies the name pig, as the animal pig also has three characteristics.

MODULE - 5

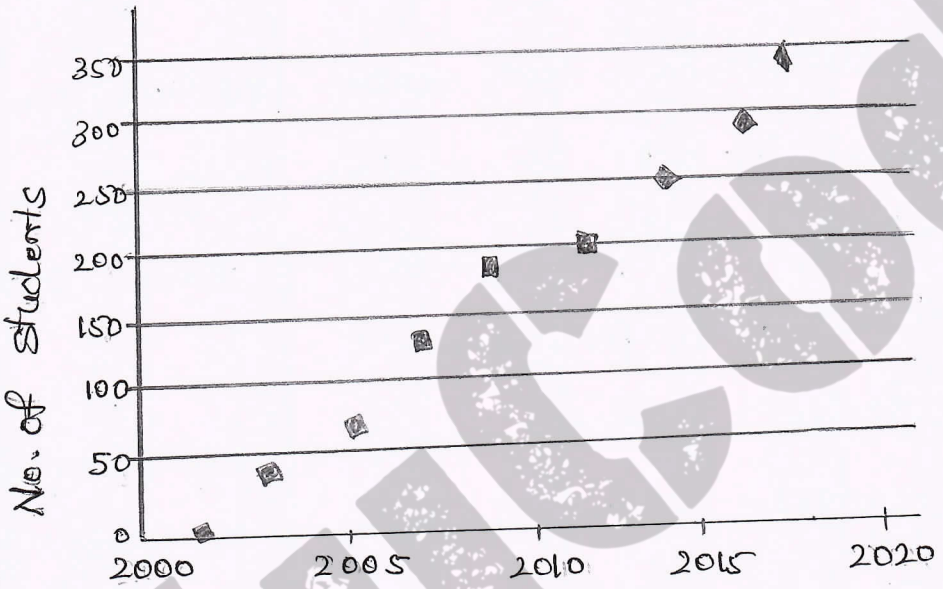
9a. In machine Learning explain linear & non-linear relationship with essential graphs.

→ **Linear relationship**

- A linear relationship exists between two variables, say x & y, when a straight line ($y = a_0 + a_1 \cdot x$) can fit on/ a graph, with atleast some reasonable

degree of accuracy.

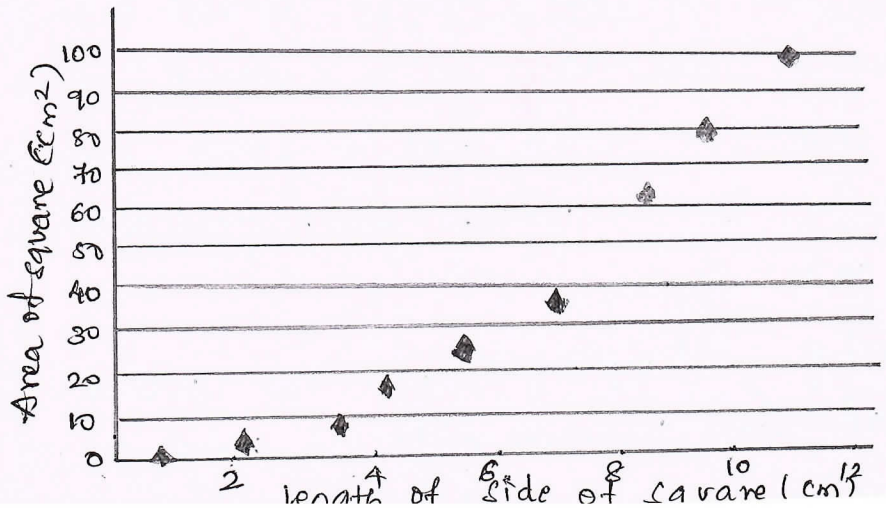
- The a_1 is the linearity coefficient.
- Example: a scatter chart can suggest a linear relationship, which means a straight line.
- Figure below shows a scatter plot, which fit a linear relationship between the number of students opting for computer courses in years between 2000 & 2017.



- A linear relationship can be negative or positive.
- A positive relationship implies if one variable increases in value, the other also increases in value.
- A negative relationship, implies, when one ~~value~~ increases in value, the other decreases in value.
- Perfect, strong, or weak linearship categories depend upon the bonding between the two variables.

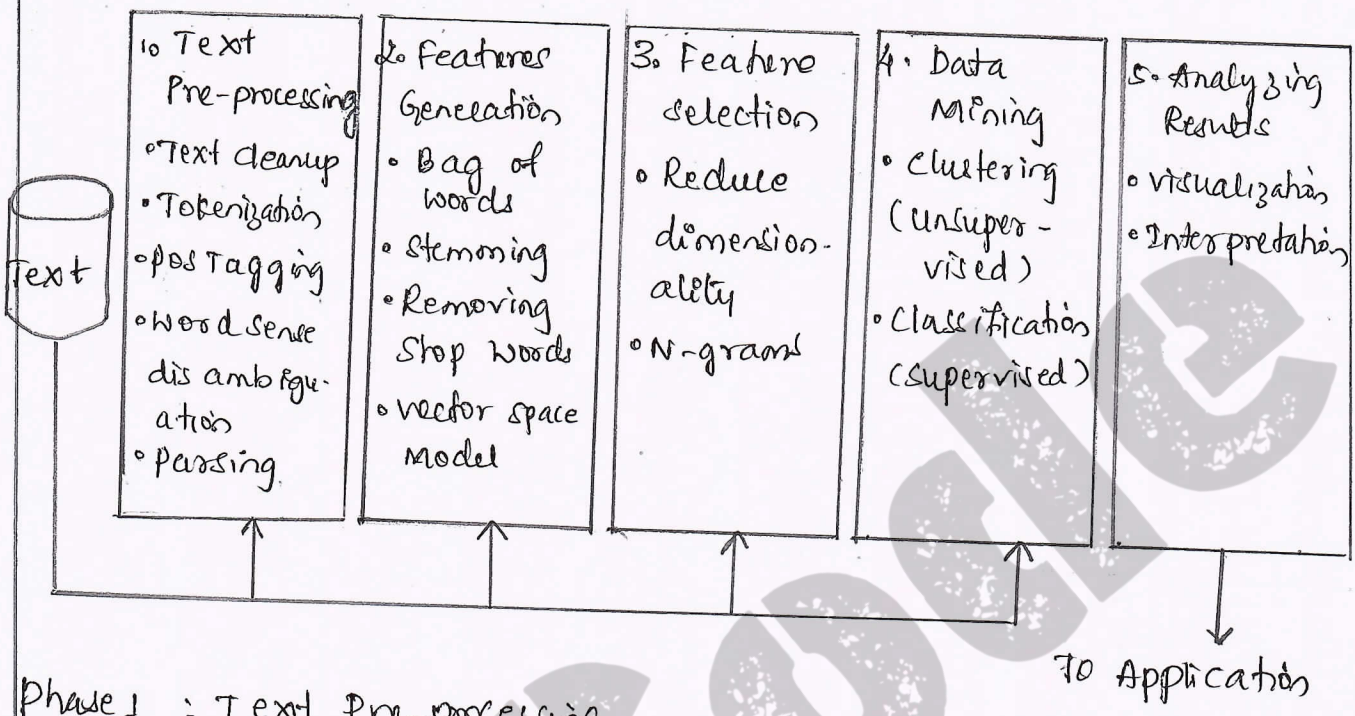
Non-linear relationship

- A non-linear relationship is said to exist between two quantitative variables when a curve $y = a_0 + a_1 x + a_2 x^2 + \ldots$) can be used to fit the data points.
- The fit should be with at least some reasonable degree of accuracy for the fitted parameters, $a_0, a_1, a_2, \ldots$
- Expression of y then generally predicts the values of one quantitative variable from the values of the other quantitative variable with considerably more accuracy than a straight line.
- consider an example of non-linear relationship!
- The side of a square & its area are not linear.
- In fact, they have quadratic relationship.
- If the side of a square doubles, then its area increases four times
- The relationship predicts the area from the side. Figure below shows this non-linear relationship.



9b. Write the block diagram of text mining process and explain its phases.

→ Figure below shows the text mining process pipeline.



Phase 1 : Text Pre-processing

It enables syntatic / semantic text-analysis & does the following.

1. Text cleanup - process of removing unnecessary or unwanted information
2. Tokenization - process of splitting the cleanup text into token. using white spaces & punctuation marks.
3. Part of speech Tagging - pos tagging is a method that attempts labelling of each token with an appropriate pos.
4. Word sense disambiguation - method which identifies the sense of word used in a sentence, that gives meaning in case the word has multiple meanings.
5. Parsing is a method which generates a parse-tree for each sentence.

Parsing attempts & infers the precise grammatical relationship between different words in a sentence.

Phase 2 : Feature Generation :

is a process which first defines features. Some of the ways of feature generation are

1. Bag of words - Text document is represented by words it contains. Document classification methods commonly use bag-of-words model.
2. Stemming - identifies word by its root. It reduces word to its most basic element.
   ex impurification → pure
3. Removing stop words from the feature space - they are the common words, unlikely to help text mining.
4. Vector space model (VSM) - an algebraic model for representing ~~text~~ documents as vector of identifiers, word frequencies or terms in the document index.

Phase 3 : Features selection

- is the process that selects a subset of features by rejecting irrelevant and/or redundant features.
- It performs dimensionality reduction, N-gram evaluation, noise detection & evaluation of outliers
- Feature selected, algorithm reduces dimensionality that not only improves the performance of learning algorithm but also reduces the storage requirement for a dataset.
- The process enhances data understanding & visualization

Phase 4 : Data mining techniques

- This enable insights about the structured database that resulted from the previous phases.

- It involves unsupervised learning (ex clustering), supervised learning (ex classification), Identifying evolutionary patterns in temporal text streams

Phase 5: Analysing results

i) Evaluate the outcome of the complete process

ii) Interpretation of Result - If acceptable then results obtained can be used as an input for next set of sequences.
Else, the result can be discarded & try to understand what & why the process failed.

iii) Visualization - prepare visual from data & build a prototype

iv) use the results for further improvement in activities at the enterprise, industry or institution.

10a) Define multiple regression. Write down the examples involved in forecasting and optimization in regression.

→ Multiple regressions are used when two or more independent factors are involved.

- These regressions are also widely used to make short-to mid-term predictions to assess which factors to include & which to exclude.
- Multiple regressions can be used to develop alternate models with different factors.
- More than one variable can be used as predictor with multiple regressions.

The prediction takes the form

$$y = a + c_1 x_1 + c_2 x_2 + \cdots + c_n x_n$$

where a is the intercept of the line on the y axis

$c_1, c_2, \cdots c_n$ are coefficients, representing the contribution of the independent variables $x_1, x_2, \cdots x_n$ in the calculation of y.

The examples involved in forecasting and optimization in regression are

i) Using linear analysis on sales data with monthly sales, a company could forecast sales for future months.

ii) For the funds that a company has invested in marketing a particular brand, an analysis of whether the investment has given substantial returns or not can be made.

iii) Suppose two promotion campaigns are running on TV, Radio in parallel. A linear regression can confine the individual as well as the combined impact of running these advertisements together.

iv) An insurance company exploits a linear regression model to obtain a tentative premium table using predicted claims to insured declared value ratio.

v) A financial company may be interested in minimizing its risk portfolio & hence want to understand the top five factors or reasons for default by a customer.

vi) To predict the characteristics of child based on the characteristic of their parents.

vii) A company faces an employment descrimination matter in which a claim that women are being descriminated against in terms of salary is raised.

viii) Predicting the prices of houses, considering the locality & builder characteristics in a locality of a techniques.

ix) Finding relationships between the structure & the biological activity of compounds through their physical, chemical & physiochemical traits is most commonly performed with regression techniques

x) To predict compound with higher bioactivity within groups.

10b) Explain parameters in social graph network topological analysis using centralities & PageRank.

→ Parameters in social graph network topological analysis using centralities & PageRank are

1) degree

- degree of a graph vertex means the total number of edges linked to that.
- In-degree of a vertex means the number of in-edges from the other vertices.
- out degree means the number of out edges from that vertex.

2) Closeness

Graph vertex Closeness $C_c(v)$ is a way of defining the centrality of a vertex in reference to other vertices.

- The centrality, c is function of distances of vertices

$$C_c(v) = \left[\sum_{u \in V} d(u,v)\right]^{-1}$$

where $d(u,v)$ is distance between u & v

3) Effec for path traversal.

3) Effective closeness

effective closeness $C_{ec}(v)$ can also be analysed. We approximate average distance from v to all other vertices in place of the shortest paths.

C_{ec} reduces run time for cases with a large number of edges & near linear scalability in computations.

4) Betweenness

Graph vertices betweenness means the number of times a vertex exists between the shortest path & the extent to which a vertex is located between other pair of vertices.

5) Page Rank

Page Rank is a metric for the importance of each vertex in a graph, assuming an edge from v_1 to v_2 represents endorsement of importance of v_2 by v_1 by connecting, following, interacting, opting for relationship, sharing belief or some ~~other~~ means.

6) Contacts Size

Contacts Size means a meter connection to many neurons.

7) Indirect contacts

Indirect Contacts metric means betweenness, which is the sum of the shortest paths within geodesic distances from all other pairing neurons.

- Three-step contact metric means a number of edges to other neurons plus the number of edges from other neurons within geodesic distances ≤ 3.

8) Structure diversity

Structure diversity means that social graph has access to diverse sub-graphs.