6.

Data Visualization
- graphical representation of information & data. using visual elements like charts, graphs & maps.
- helps users identify complex dataset quickly by identifying patterns, trends & outliers.

importance
- makes large & complex data easier to understand.
- common tools could be used.
- used in BI, data analytics & decision making
- makes insights of the data clear & easy

Dis
- may cause misinterpretation
- complex visuals → overwhelm users.
- limitations of visualization tools.

challanges in Big Data Visualization.
1. Volume - visualizing immense datasets is difficult due to sheer size & complexity
2. Variety:- combining variety of data source into a visual is challenging.
3. Velocity:- Real time data streaming requires fast processing & updating visuals.
4. Visual Noise:- high density of data points makes hard to distinguish individual obj's
5. Information loss:- data reduction has risk of info loss.
6. large image perception:- Devise constraints limits. how data can be effectively displayed.
7. High rate of change:- Rapid data update can overwhelm the user
8. High performance requirement.

⇒ Solutions → ① Meeting the Need for speed ② understanding the data ③ Addressing the data Quality ④ only meaningful result ⑤ Dealing with outliers.

Applications of Data Visualization

(1). Business sailes Analysis. (barchaut)
- Track sails peufoumance by uegion or product.

(2). Website tuafic monitoring. (line chaut)

(3) Customeu feed back & Rating. (pichaut, Histoguam)

(4). outlieu Detection in finance & helthcase. (Boxplot)

Types of data Visualization

(1). line chaut → shows tuends oveutime

(2). Bauchaut → compare quantity aviross catogany

(3). Pie chaut → %. or psapoutional data.

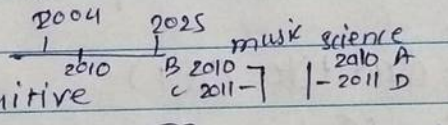
(4). Scatteuplot → show uelation b/w 2 vauiables.

(5). Histoguam → distuibution of numeuical data.

(6). aued chaut

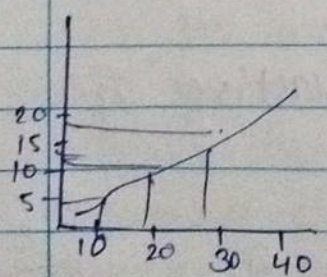
(7). Flowchaut

(8). timeline. → (1) lineas (2) compauitive



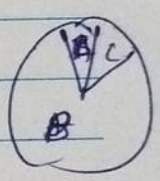
(1). line chaut.
- cueated using plot() fuanc.
- show uel of x & Y

```
x = [10,20,30,40]
y = [20,25,30,35]
plt.plot(x,y)
plt.title("ABC")
plt.xlable("X")
plt.ylable("Y")
plt.show()
```



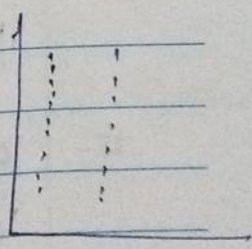
(2). pie chaut.

```
cars = ['A','B','C']
data = [10,80,10]
plt.pie(data, labels=cars)
plt.show()
```



(3).

```
x = data['day']
y = data['total']
plt.scatter(x,y)
plt.show()
```



(4). (2) plt.hist(data, bins=10, color='skyblue
(1) data = np.random.randn(1000)

Tools used in data visualization.

(5). Density plot. - descreat
- smoothed version of histogram
- use to visualize distribution of data & identify peaks, spread & skewness.
- uses ~~dimenti~~ kernel density estimation
- good for comparing multiple distributions.
- curve based.

```
eg. data = np.random.normal(loc=0, scale=1, size=1000) //random data
sns.kdeplot(data, shade=True)
plt.show()
```
- not fixed interval.

(6). Box plot.
- shows distribution of dataset. through 5 summary stats.
1. min.
2. first Quartile (Q1)
3. median (Q_2)
4. Third Quartile (Q_3)
5. max.

- shows outliers if present.

```
eg. data = np.random.normal(0,1,1000)
sns.boxplot(data=data, color="skyblue")
plt.show()
```

Tools used in Data Visualization / Analytical tools in Big data visualization.
- analyzing massive datasets
- transform raw data into interactive & understandable visuals.

⇒ Tools are.

(1) Tableau — D3.js — Pentaho
(2) Power BI — plotly — Datameer.
(3) Qlik — kibana.
(4) Google datastudio.

(1). Tableau
- powerful tool for interactive data visualization
- connects to big data platforms like hadoop, spark, Azure Aws & SQL
- has drag & drop interface
- also used for BI

(2). Po- allows user to analize, visualise & share data insights through interactive & shareable dashboard
- converts raw data to understandable format.

key features
(1). user friendly UI - No. coding required
- drag & drop functionality
(2). connects to multiple data source - SQL, Excel, google sheets.
(3). Real time data Analysis. - supports live data connections & real time dashboard updates.
(4). Interactive Dashboard. - visualization can be filtered, drilled down, & updated dynamically
(5). Advanced chart types - scatterplot, histogram, geo plots.
(6). Data blending & joining. - data from multiple sources.
(7). Storytelling - creates a sequence of visualization to tell a data story
(8). Tableau (Public / Desktop / server)
Public - free. paid (full featured) → to share dashboard

adv.
- easy to learn, highly interactive & dynamic.
- Supports large datasets.

limitations
- high cost, limited custom visualization.
- Requires clean data.

Use case :- Analyze monthly sails, product wise performance.

(2) Power BI
- Business analytics tools developed by microsoft
- enables users to visualize & share insights across organizations
- helps create interactive dashboard, reports & data models.
- widely used in business & academic env. for data-driven decision making

Key Features.
(1). Multiple data source connectivity.
(2). Interactive Visualization
(3). Power Query editor - to clean, Transform & shape the data
(4). DAX (Data analysis expressions)
- a formula lang. used to create custom calc.
(5). Dashboards & reports,
- dashboard shows real time KPI's
- reports, multi paged visuals / charts
(6). Data modeling:-
- Relationship b/w tables can be created & analyse
- Supports star & snowflake schema.
(7). cloud & on premise Deployment.
(8). AI ~~capabilities~~ compatibility

adv.
- user friendly
- Integrated with MS. tools.
- cloud based sharing & collaboration
- Real time data analysis

Limitation - slow with large dataset
DAX complex for beginners.
limited customization for advanced visualizations.

(3) Qlick
- data analytics & BI platfoum
- allows users to make interactive visualizations dashboards & data apps.
- they have 2 main types / products.
  1. Qlik View - script based dashboard dev. tool
  2. Qlik Sense - Self survise & cloud friendly BI platfor.
- Qlik is known for its in-Memory data processing & associative data model, allows users to explor & analyze data without being restricted by predefined queries

key features.
(1). Associative Data model:-
unlike SQL based tools, Qlik links data from multiple sources. in non-lineau associative way.
(2). In memory processing. Data is loaded in memory → fast processing & calculations.
(3). Data loading & ETL:- powerful scripting engine. allows data transform, cleaning & loading from diverse sources
(4). Rich visualization capabilities.
(5). Self service BI → no heavy IT involvement
(6). AI-powered insights.
(7). collaboration & sharing. via Qlik cloud & servers.

adv.
- fast & intuitive data discovery
- easy explanation.
- strong data compression & performance
- scalable (small, large) business.
- can integrate with R, python, Rest APIs

Limitations.
- less intuitive
- hard to learn.
- paid licence

-suppouts data intensive distributed applications.

Hadoop Ecosystem.

- platfoum ou framework to solve Big data problm
- Ecosystem refers to the various components in Software library of hadoop (accessaries & Tools)
- Hadoop is Java based framework for handling & analysing large set of data
- use of different parts of core hadoop set such as MapReduce, to handle vast amount of data. HDFS, a file handling system, YARN - hadoop resource manager.

Tools of hadoop ecosystem

(1). HDFS, Hive, Pig, YARN, MapReduce, Spark, HBase etc.

Ecosystem.

coordination	workflow & scheduling.	scripting (Pig)	ML (Mahout)	Query (Hive)	NOSQL HBase	Data integration
			(MapReduce)			
		Distributed processing				
		Distributed storage (HDFS)				

(1) HDFS: Hadoop distributed file system
- primary storage sys. of hadoop,
- uses NameNode & DataNode architecture.
- it is a distributed file system.
- NameNode (Master) & DataNode (acts as Slave)

(2). Hive - ETL & Data warehouse tool
- used to query ou analyze datasets.
- 3 main func.
  (1). Data summarization.
  (2) Query
  (3) analysis of unstructured & semi-structured data.

(3). MapReduce
- core component of processing (provides logic of processing)
- software framework helps in writing applications.

that process large datasets.
- distributed & parallel flow of tasks.

④ Apache Pig
- high level scripting lang.
- to execute queries for large dataset.

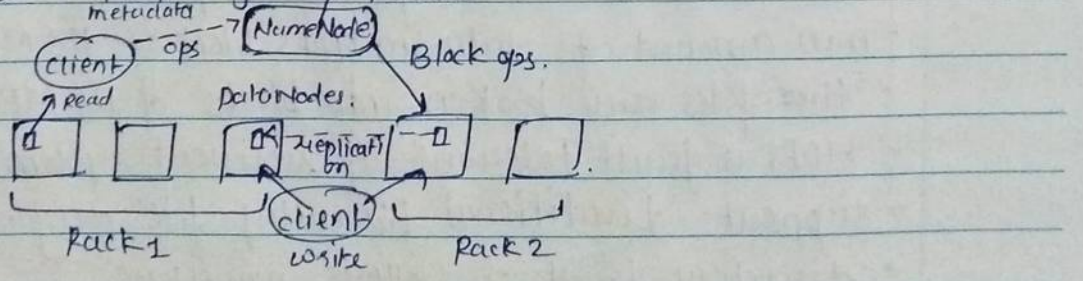
⑤ Apache Spark
- fast in memory data processing engine
- Supports, Java, py, scala, R & SQL

⑥ Apachase HBase
- hadoop eco-sys. component. for distributed database
- designed to store structured data
- scalable, distributed, NoSQL database.
- build on top of HDFS.

Hadoop architecture.

consist of ① Named Node
② Data Node
③ Secondary NameNode
④ HDFS client
⑤ Block structure

- to process & store vast amount of data hadoop ~~use~~ mostly uses. HDFS & Map Reduce.



HDFS is a block structured file system where each file is divided into blocks of pre-determined size
- blocks are stored across clusters
- HDFS follows Master slave architecture ie NameNode (Master) &

data Node (slave) → they process & store blocks.
- NameNodes - maintain meta data of dataNodes. & controles client access.
  - info like Namespace info, block info
  - this info only brought to memory when some activity is requested else its stored in disk/ persistant storage.
- fsimage - logs. of Name node, (snapshot)
  editlog - changes to read & write logs. (when namenode starts)
- To restart a NameNode we need fs image (to initialize) & edittog. (to update the logs. of NameNode - Time consuming)
- if the NameNode shuts down the fsimage is used to start it at the same position as it was using the image it stores.
- & the edit log updates the NameNode by updating its logs. it requires a lot of time.
- to reduce that time secondary NameNode updates the editlog to nameNode in a fixed interval.
- Reducing the time for restart.
- HDFS & MapReduce is must without that nothing can happen.

HDFS.
- Hadoop distributed file system
- used to scale a single hadoop cluster to 100's of nodes.
- min amount of data in HDFS block is 128MB
- files are broken into blocks of 128MB size
- HDFS is fault tolerant & resilient system.
- support traditional hierarchy file organization
- directory based & follows namespace.
- follows Master Slave architecture using NameNodes & dataNodes.
- can be deployed on machines that operate on Java

MapReduce

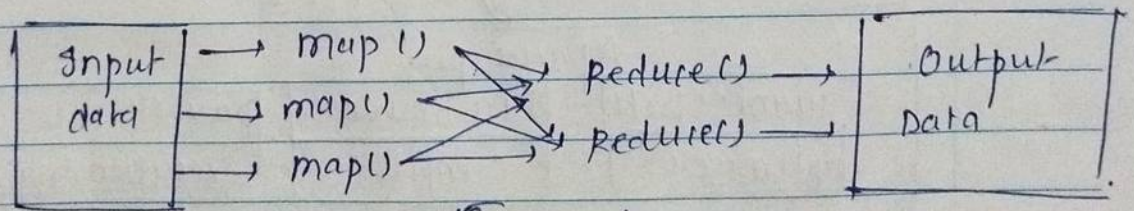
- is a programming model & software framework
- Intends to simplify the process of vast amount of data in parallel on large clusters of commodity hardware in a reliable fault-tolerant manner.

characteristics.

- works on large scale data
- write once read many times(data).
- allows parallelism without mutexes.
- map & reduce. main operations.

Map() & Reduce()

- performs actions like filtering, grouping & sorting
- reduce() func. aggregates the summarize the result
- result generated by map is (key, value) pair which acts as input to Reduce func.
- Every Map/Reduce program must specify a Mapper. & a Reducer., mapper has map method that transforms input into(key, val), Reducer has reduce method. the transform (key, val) arguments int any no. of output. (key, val) pairs



flow. (optional)

Input → split → map → combine → shuffel → Reduce → o/p

Split data to small pieces

process each split according to map() func.

sort

How Mapfedure fits in Hadoop ecosys

1. Data Processing Engine.
2. working with HDFS
3. distributed processing
4. Data locality optimization.
5. fault tolerance
6. Job scheduling & resource management
7. Scalability (horizontally) (among hardware nodes)

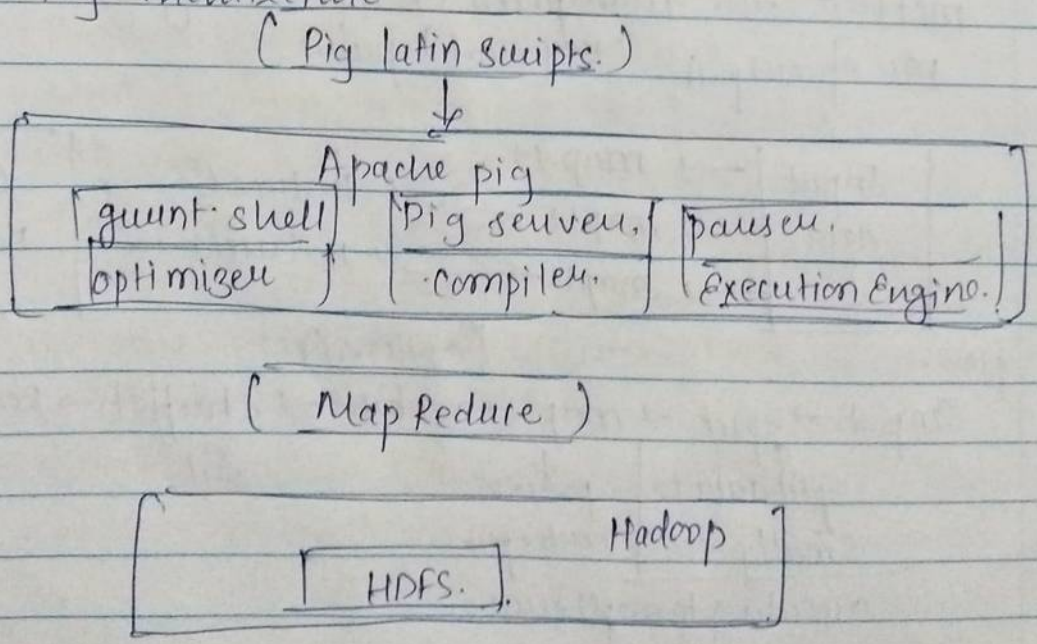
Apache Pig.

- high level platform for creating mapReduce programs used with hadoop.
- uses scripting language called pig latin.
- simplifies coding required for processing large dataset

Features

- handles structured & semi unstructured data.
- provides data flow language
- supports user defined func
- used for ETL (Extract, Transform, load)

Pig Architecture



① Pig latin script → user writes script to define data operations
② Grunt shell → interactive command line interface (CLI) where user write pig latin script.
③ Pig server → acts as interface b/w user & Pig runtime env. managing the execution of pig scripts.
④ Parser → parses the script & checks syntax producing a logical plan
⑤ optimizer → optimizes the logical plan. by applying optimization rules to make data processing more efficient
⑥ compiler :- converts the optimized plan to a series of MapReduce jobs.
⑦ Execution Engine :- Runs compiled MapReduce jobs on hadoop cluster, processing the data stored in HDFS.
⑧ MapReduce :- Executes the tasks generated by pig
⑨ HDFS. - stores input & output.

- Pig acts as interface b/w user & hadoop
- Reduces complexity of writing raw MapReduce code.
- Supports batch & interactive mode.

HIVE (Apache)
- data warehouse infrastructure build on top of hadoop.
- provides an SQL like interface (HiveQL) to query & manage large dataset stored in HDFS.
- for users who are comfortable with SQL bt not with Java

#Key features.
① HiveQL (Query lang) :- similar to SQL, easy data analysis
② Schema on read. - allows flexiblity
③ Batch processing ④ Build on top of hadoop
⑤ Supports user defined functions.

Hive architecture

1. UI (CLI/web UI) - interface for queries
2. Driver - manage lifecycle of HiveQL statment
   - includes parser, planner, optimizer, executor.
3. Compiler - converts query into logical execution plan
4. Metastore - stores metadata like table schema.
   - usually stored in RDBMS
5. Execution Engine - interact with hadoop to run job & return result
6. HDFS :-
   stores data.

adv.

- SQL like lang
- easy to use.
- work on masive dataset
- scalable with HDFS & MapReduce
- supports partitioning & Bucketing.

limitations.

- No real time processing
- high latency.
- limited transactional support.
- No row level operations.

use cases

1. data summarization & reporting
2. log analysis in batch mode
3. ETL