Data Engineering:

- Data engineering is the practice of designing and building systems for collecting, storing, and analyzing data at scale.
- Data engineers work in a variety of settings to build systems that collect, manage, and convert raw data into usable information for data scientists and business analysts to interpret

What are the 5 V’s in Big Data?

1)Volume: A considerable amount of data stored in data warehouses reflects the volume. The data may reach random heights; these large volumes of data need to be examined and processed. Which may exist up to or more than terabytes and petabytes.

2)Velocity: Velocity basically introduces the pace at which data is being produced in real-time. To give a simple example for recognition, imagine the rate at which Facebook, Instagram, or Twitter posts are generated per second, an hour or more.

3)Variety: Big Data comprises structured, unstructured, and semi-structured data collected from varied sources. This different variety of data requires very different and specific analyzing and processing techniques with unique and appropriate algorithms.

4)Veracity: Data veracity basically relates to how reliable the data is, or in a fundamental way, we can define it as the quality of the data analyzed.

5)Value: Raw data is of no use or meaning but once converted into something valuable. We can extract helpful information.

Why businesses are using Big Data for competitive advantage.

Irrespective of the division and scope of the firm, data is now an essential tool for businesses to utilise. Companies are frequently using big data to gain a competing edge over business rivals.

Checking the datasets a company collects is just one part of the big data process. Big data professionals also need to know what the company requires from the application and how they plan to use the data to their advantage

How to deploy a Big Data Model? Mention the key steps involved.

Deploying a model into a Big Data Platform involves mainly three key steps they are,

- Data ingestion
- Data Storage
- Data Processing

Let's have a look at what these are,

Data Ingestion: This process involves collecting data from different sources like social media platforms, business applications, log files, etc.

Data Storage: When data extraction is completed, the challenge is to store this large volume of data in the database in which the Hadoop Distributed File system (HDFS) plays a vital role.

Data Processing: After storing the data in HDFS or HBase, the next task is to analyze and visualize these large amounts of data using specific algorithms for better data processing. And yet again, this task is more straightforward if we use Hadoop, Apache Spark, Pig, etc. After performing these essential steps, one can deploy a big data model successfully

What are the different big data processing techniques?

Big Data processing methods analyze big data sets at a massive scale. Offline batch data processing is typically full power and full scale, tackling arbitrary BI scenarios. In contrast, real-time stream processing is conducted on the most recent slice of data for data profiling to pick outliers, impostor transaction exposures, safety monitoring, etc. However, the most challenging task is to do fast or real-time ad-hoc analytics on a big comprehensive data set. It substantially means you need to scan tons of data within seconds. This is only probable when data is processed with high parallelism.

Different techniques of Big Data Processing are:

- Batch Processing of Big Data
- Big Data Stream Processing
- Real-Time Big Data Processing
- Map Reduce

Explain Features Selection.

During processing, Big data may contain a large amount of data that is not required at a particular time, So we may be required to select only some specific features that we are interested in. The process of extracting only the needed features from the Big data is called Feature selection.

Feature selection Methods are -

Filters Method: In this method of variable ranking, we only consider the importance and usefulness of a feature.

Wrappers Method: In this method, 'induction algorithm' is used, Which can be used to produce a classifier. **Embedded Method:** This method is a combination of efficiencies of both Filters and wrappers methods

How do you convert unstructured data to structured data?

An open-ended question and there are many ways to achieve this.

Programming: Coding/ Programming is the most tried out method to transform unstructured data into a structured form. Programming is advantageous to accomplish because we get independence with it, which you can use to change the structure of the data in any form possible. Several programming languages, such as Python, Java, etc., can be used. **Data/Business Tools:** Many BI (Business Intelligence) tools support the drag and drop functionality for converting unstructured data into structured data. One cautious thing before using BI tools is that most of these tools are paid, and we have to be financially capable to support these tools. For people who lack both experience and skills needed for option 1, this is the way to go.

What is data preparation?

Data preparation is the method of cleansing and modifying raw data before processing and analyzing it. It is a crucial step before processing and usually requires reformatting data, making improvements to data, and consolidating data sets to enrich data.

Data preparation is an unending task for data specialists or business users. But, it is essential to convert data into context to get insights and then, can eliminate the biased results found due to poor data quality.

For instance, the data construction process typically includes standardizing data formats, enhancing source data, and/or eliminating outliers

- Gather data
- Discover and assess data
- Clean and verify data

- Transform and enrich data
- Store data

Double-click (or enter) to edit

Cloud Data Engineering:

A cloud data engineer, also known as a cloud engineer or cloud developer, is someone responsible for the management of corporate apps and data in the cloud and all the technical tasks involved in planning, architecting, migrating, monitoring, and managing a company's cloud systems

Cloud tools for data engineering:

-Adf

-Azure databicks

-Adls

-Storage account

-Azure Synapse

-Azure logic app

-Power Bi

-Model building

Structured, Semi structure, Unstructure data:

Structured: csv ,excel , database, Parquet/orc/avro can we structured and semi-structured

Unstructured: text, video ,audio, images, number, messages, social media post, survey forms,

Semi structured data: email, www,xml,pdf

ETL:

ETL is an abbreviation of Extract, Transform and Load. In this process, an ETL tool extracts the data from different RDBMS source systems then transforms the data like applying calculations, concatenations, etc. and then load the data into the Data Warehouse system

ELT:

- ELT is a different method of looking at the tool approach to data movement. Instead of transforming the data before it’s written, ELT lets the target system to do the transformation. The data first copied to the target and then transformed in place.
- ELT usually used with no-Sql databases like Hadoop cluster, data appliance or cloud installation.

OLTP:

- OLTP or Online Transaction Processing is a type of data processing that consists of executing a number of transactions occurring concurrently—online banking, shopping, order entry, or sending text messages, for example. These transactions traditionally are referred to as economic or financialtransactions,.recorded and secured so that an enterprise can access the information anytime for accounting or reporting purposes

OLAP:

- OLAP on big data is a powerful concept that involves the pre-aggregation of massive amounts of data and builds multidimensional cubes to get super-fast query results

---

Azure Cosmosdb:

Azure Cosmos DB is Microsoft’s globally distributed, multi-model database. Azure Cosmos DB enables you to elastically and independently scale throughput and storage across any number of Azure’s geographic regions. It offers throughput, latency, availability, and consistency guarantees with comprehensive service level agreements (SLAs).

Azure Cosmos DB provides APIs for the following data models, with SDKs available in multiple languages

- SQL API
- MongoDB API
- Cassandra API
- Graph (Gremlin) API

- Table API

Apache Cassandra:

Apache Cassandra is an open-source no SQL database that is used for handling big data. Apache Cassandra has the capability to handle structure, semi-structured, and unstructured data. Apache Cassandra was originally developed at Facebook after that it was open-sourced in 2008 and after that, it become one of the top-level Apache projects in 2010.

- It is scalable, fault-tolerant, and consistent.
- It is column-oriented database.
- Its distributed design is based on Amazon's Dynamo and its data model on Google's Big table.
- It is Created at Facebook and it differs sharply from relational database management systems

Features of Cassandra:

- Easy data distribution
- Flexible data storage
- Elastic scalability
- Fast writes
- Always on Architecture
- Fast linear-scale performance
- Transaction support

Apache hive:

- Apache Hive is open-source data warehouse software designed to read, write, and manage large datasets extracted from the Apache Hadoop Distributed File System (HDFS) , one aspect of a larger Hadoop Ecosystem.
- Apache Hive is an open source project that was conceived of by co-creators Joydeep Sen Sarma and Ashish Thusoo during their time at Facebook

Apache Hive architecture and key Apache Hive components:

1)Hive Server 2: The Hive Server 2 accepts incoming requests from users and applications and creates an execution plan and auto generates a YARN job to process SQL queries. The server also supports the Hive optimizer and Hive compiler to streamline data extraction and processing.

2)Hive Query Language: By enabling the implementation of SQL-reminiscent code, the Apache Hive negates the need for long-winded JavaScript codes to sort through unstructured data and allows users to make queries using built-in HQL statements (HQL). These statements can be used to navigate large datasets, refine results, and share data in a cost-effective and time-efficient manner.

3)The Hive Metastore: The central repository of the Apache Hive infrastructure, the metastore is where all of the Hive's metadata is stored. In the metastore, metadata can also be formatted into Hive tables and partitions to compare data across relational databases. This includes table names, column names, data types, partition information, and data location on HDFS.

4)Hive Beeline Shell: In line with other database management systems (DBMS), Hive has its own built-in command-line interface where users can run HQL statements. Also, the Hive shell also runs Hive JDBC and ODBC drivers and so can conduct queries from an Open Database Connectivity or Java Database Connectivity application.

What are the five different data types used by Apache Hive?: 1)Numeric Data Types: As the name suggests, these data types are integer-based data types. Examples of these data types are 'TINYINT,' 'SMALLINT,' 'INT,' and 'BIGINT'.

2)Date/Time Data Types: These data types allow users to input a time and a date, with 'TIMESTAMP,' 'DATE,' and 'INTERVAL,' all being accepted inputs.

3)String Data Types: Again this type of data is very straightforward and allows for written text, or 'strings,' to be implemented as data for processing. String data types include 'STRING,' 'VARCHAR,' and 'CHAR.'

4)Complex Data Types: One of the more advanced data types, complex types record more elaborate data and consist of types like 'STRUCT', 'MAP,' 'ARRAY,' and 'UNION.'

5)Misc. Types: Data types that don't fit into any of the other four categories are known as miscellaneous data types and can take inputs such as 'BOOLEAN' or 'BINARY.'

Hbase:

- HBase is a column-oriented non-relational database management system that runs on top of Hadoop Distributed File System (HDFS). HBase provides a fault-tolerant way of storing sparse data sets, which are common in many big data use cases. It is well suited for real-time data processing or random read/write access to large volumes of data.
- HBase is a data model that is similar to Google's big table designed to provide quick random access to huge amounts of structured data. It leverages the fault tolerance provided by the Hadoop File System (HDFS).
- It is a part of the Hadoop ecosystem that provides random real-time read/write access to data in the Hadoop File System

Apache kafka:

Apache Kafka is an open-source distributed event streaming platform used by thousands of companies for high-performance data pipelines, streaming analytics, data integration, and mission-critical applications

Apache Snowflake:

- Snowflake is a single platform comprised of storage, compute, and services layers that are logically integrated but scale infinitely and independent from one another.
- Snowflake is a cloud data warehouse built on top of the public cloud (AWS/Azure / GCP) infrastructure and is a true SaaS offering. There is no hardware (virtual or physical) for you to select, install, configure, or manage. There is no software for you to install, configure, or manage. All ongoing maintenance, management, and tuning is handled by Snowflake.

Difference among data warehouse, data lake, Delta lake

1] DATA WAREHOUSE:

- Only Structured Data
- Schema-on-Write
- Supports ACID Transaction
- Does not corrupt the system

2]Data Lake:

- Structured/Semi structured/Unstructured
- Schema-on-Read
- Minimal support to ACID Transactions
- Leaves system in corrupted state

3]Delta lake:

- Structured/Semi structured/Unstructured/ Streaming
- bSchema-on-Read
- Supports ACID Transaction
- Does not corrupt the system

File Formats:

1]CSV: Good option for compatibility, spreadsheet processing and human readable data. The data must be flat. It is not efficient and cannot handle nested data. There may be issues with the separator which can lead to data quality issues. Use this format for exploratory analysis, POCs or small data sets.

2]JSON: Heavily used in APIs. Nested format. It is widely adopted and human readable but it can be difficult to read if there are lots of nested fields. Great for small data sets, landing data or API integration. If possible convert to more efficient format before processing large amounts of data.

3]Avro: Great for storing row data, very efficient. It has a schema and supports evolution. Great integration with Kafka. Supports file splitting. Use it for row level operations or in Kafka. Great to write data, slower to read.

4]Protocol Buffers: Great for APIs, especially for gRPC. Supports Schema and it is very fast. Use for APIs or machine learning.

5]Parquet: Columnar storage. It has schema support. It works very well with Hive and Spark as a way to store columnar data in deep storage that is queried using SQL. Because it stores data in columns, query engines will only read files that have the selected columns and not the entire data set as opposed to Avro. Use it as a reporting layer.

6]ORC: Similar to Parquet, it offers better compression. It also provides better schema evolution support as well, but it is less popular.

1] Avro: Schema Evolution -Best Compression - Good Splitability - Good Row or Column - Row Read or Write- Write

2] Parquet: Schema Evolution -Good Compression - Better Splitability - Good Row or Column - Column Read or Write- Read

3] ORC: Schema Evolution -Better Compression - best Splitability - best Row or Column - column Read or Write- read

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit