DSBDA END

Sure! Here's a detailed answer for your question:

Q1) a) Explain all the steps for writing a file in HDFS with neat diagram. [8]

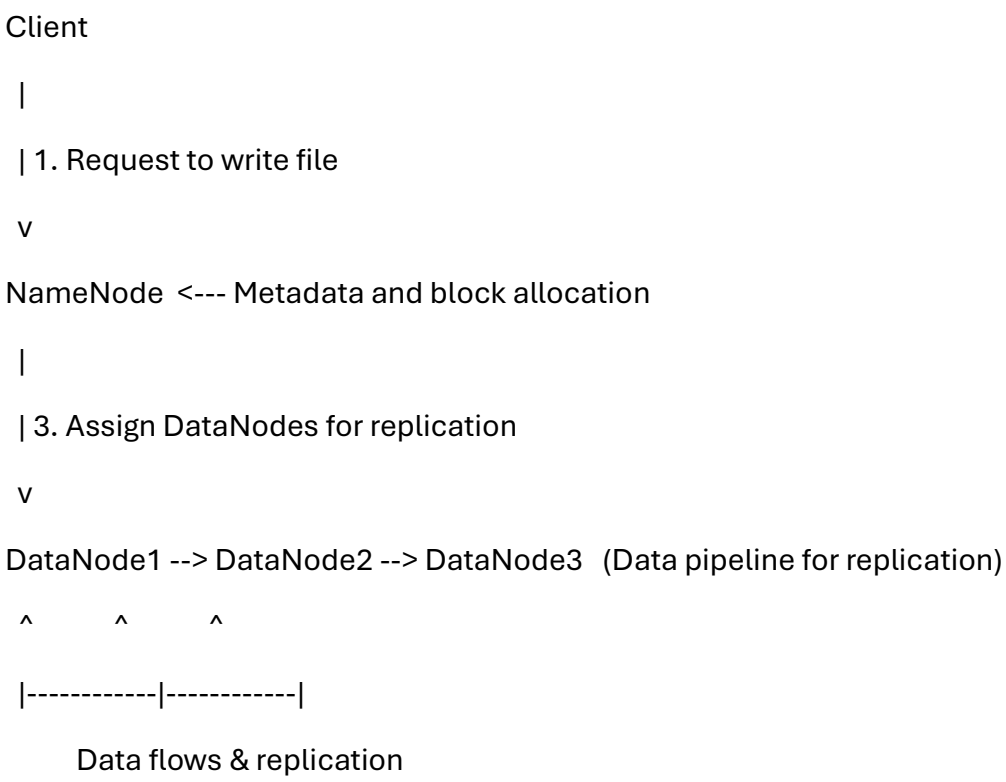
HDFS (Hadoop Distributed File System) is designed to store very large files across machines in a reliable and fault-tolerant manner. Writing a file to HDFS involves several key steps to ensure data is stored efficiently and replicated.

Steps to Write a File in HDFS:

1. **Client Request**
   The client wants to write a file to HDFS and contacts the **NameNode** to create a file.
2. **NameNode Validation**
   The NameNode checks:
   - If the file already exists.
   - If the client has the necessary permissions.
     If all is okay, NameNode records the file creation metadata and responds to the client.
3. **Data Split and Block Allocation**
   The file is split into blocks (default 128 MB or 64 MB). For each block, the NameNode selects a set of **DataNodes** to store the replicas of the block (default replication factor is 3).
4. **Data Pipeline Setup**
   The client gets the list of DataNodes for each block and starts sending the data to the first DataNode.
5. **Data Streaming and Replication**
   - The client sends data to the first DataNode.
   - The first DataNode writes data to its local disk and forwards the data to the second DataNode.
   - The second DataNode writes data and forwards to the third DataNode.
   - The third DataNode writes the data.
     This forms a pipeline ensuring replication.

6. **Acknowledgments**
   Once the third DataNode writes the data, an acknowledgment is sent back through the pipeline to the client confirming successful write.
7. **File Close**
   After all blocks are written and acknowledged, the client sends a close request. The NameNode marks the file as closed and available for read.

Diagram:



Q1) b) Describe the various types of NoSQL Databases with example and also compare them.

NoSQL databases are designed for handling large volumes of data with flexible schemas, high scalability, and high availability. There are mainly **four types of NoSQL databases**:

Type	Description	Example	Use Case
1. Key-Value Stores	Store data as a collection of key-value pairs. Fast for simple lookups.	Redis, Amazon DynamoDB, Riak	Caching, session management
2. Document Stores	Store data in documents (usually JSON, BSON) which can have nested structures and fields.	MongoDB, CouchDB	Content management, real-time analytics
3. Column-Family Stores	Store data in columns rather than rows, optimized for large-scale distributed data.	Apache Cassandra, HBase	Time-series data, recommendation engines
4. Graph Databases	Store data as nodes and edges (relationships), optimized for complex relationship queries.	Neo4j, Amazon Neptune	Social networks, fraud detection, recommendation

---

Comparison:

Feature	Key-Value Stores	Document Stores	Column-Family Stores	Graph Databases
Data Model	Key-Value pairs	Semi-structured documents	Columns grouped into families	Nodes and edges
Schema Flexibility	Highly flexible	Flexible	Semi-structured	Flexible
Query Complexity	Simple key-based queries	Query on document fields	Complex queries on columns	Complex traversals & relationships
Scalability	Horizontally scalable	Horizontally scalable	Highly scalable	Moderately scalable
Examples	Redis, DynamoDB	MongoDB, CouchDB	Cassandra, HBase	Neo4j, Amazon Neptune

Feature	Key-Value Stores	Document Stores	Column-Family Stores	Graph Databases
Best for	Simple lookups, caching	Rich data representation	Large datasets with complex queries	Connected data & relationships

---

If you want, I can also help prepare a neat diagram or a formatted table for your notes. Would you like that?

Certainly! Here's a detailed answer to your question:

Q2) a) Why MapReduce is required in Hadoop? Explain the stages involved in MapReduce task with a suitable example. [9]

Why MapReduce is required in Hadoop?

- **Big Data Processing:** Hadoop stores huge volumes of data in HDFS, but processing this massive data efficiently is challenging.
- **Distributed Processing:** MapReduce allows parallel processing of large data sets by dividing the tasks across many nodes.
- **Fault Tolerance:** It handles node failures automatically by re-executing failed tasks on other nodes.
- **Simplicity:** Developers can write simple map and reduce functions without worrying about the complexity of distributed computing, data distribution, or fault tolerance.
- **Scalability:** It can scale to thousands of nodes easily.

In short, MapReduce is a programming model that processes large datasets in a distributed and fault-tolerant manner on the Hadoop ecosystem.

Stages involved in MapReduce Task

The MapReduce processing consists mainly of **three phases**: **Map**, **Shuffle & Sort**, and **Reduce**.

---

1. **Map Phase:**

- The input data is split into independent chunks (usually blocks from HDFS).
- Each chunk is processed by a **Mapper** function.
- The Mapper takes input key-value pairs and outputs intermediate key-value pairs.

2. **Shuffle & Sort Phase:**
   - The intermediate key-value pairs from all mappers are shuffled and grouped by keys.
   - This phase redistributes data such that all values corresponding to the same key go to the same Reducer.
   - Data is also sorted by keys before being sent to reducers.
3. **Reduce Phase:**
   - The Reducer receives the sorted key and list of values.
   - It processes the values to generate the final output key-value pairs.
   - The output is written back to HDFS or storage.

Example: Word Count

- **Input:** A large text file.
- **Map function:** Reads the text line by line, splits it into words, and emits (word, 1) for each word.
- **Shuffle & Sort:** Groups all (word, 1) pairs by the word.
- **Reduce function:** Sums up all the counts for each word, emitting (word, total_count).

Diagram of MapReduce Workflow:

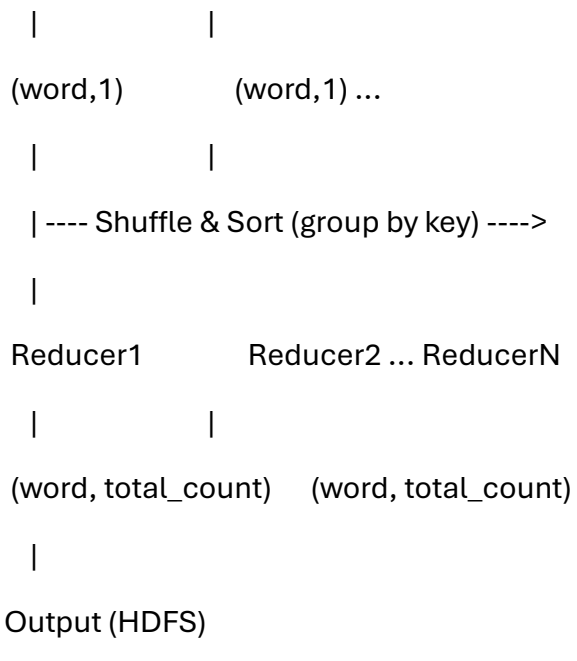
Input Data (HDFS)

|

| --> Split into chunks -->

| |

Mapper1 Mapper2 ... MapperN



Q2) b) What is Hadoop Distributed System? What is the advantage of heartbeat message in Hadoop?

What is Hadoop Distributed System?

- Hadoop Distributed System (or Hadoop Distributed File System - HDFS) is a scalable, fault-tolerant, distributed storage system designed to store very large datasets across multiple machines (nodes).
- It breaks files into blocks and distributes them across a cluster of commodity servers.
- It ensures high availability by replicating blocks across different nodes.
- It is designed to work closely with MapReduce for distributed data processing.

Advantage of Heartbeat Message in Hadoop

- **Heartbeat** is a periodic signal sent from each DataNode to the NameNode.
- It serves as a "health check" to inform the NameNode that the DataNode is alive and functioning.
- **Advantages:**
    - Helps the NameNode monitor the status of DataNodes.
    - Enables early detection of node failures.

- Allows the NameNode to take corrective action, such as re-replicating data blocks from failed nodes to healthy ones.
- Helps maintain cluster health and data reliability.

If you want, I can prepare diagrams or a concise summary as well! Let me know.

Sure! Here's a clear, detailed answer for your Q3 parts:

Q3) a) Compare HBASE and HIVE with suitable parameters. [8]

Parameter	HBase	Hive
Type	NoSQL column-family database (real-time read/write)	Data warehouse system built on top of Hadoop (SQL-like)
Data Model	Column-oriented, schema-less	Table-based with schema, supports structured data
Query Language	API-based (Java API), no SQL support	HiveQL (SQL-like query language)
Use Case	Real-time, random read/write access	Batch processing, ad-hoc queries on large datasets
Storage	Stores data in HDFS but optimized for fast random access	Stores data in HDFS in files (text, ORC, Parquet, etc.)
Latency	Low latency for reads and writes	High latency due to MapReduce jobs
Schema Evolution	Flexible, schema can evolve dynamically	Schema defined upfront, changing schema is more involved
Integration	Integrated with HBase API, supports MapReduce, Spark	Integrates with MapReduce, Tez, Spark

---

Q3) b) How missing values are filled in Pandas DataFrame with zeros? Assume suitable data. [3]

In Pandas, you can fill missing values (NaN) using the fillna() method. To fill missing values with zeros:

```
import pandas as pd

# Example DataFrame with missing values
data = {'A': [1, 2, None, 4],
        'B': [None, 2, 3, 4]}

df = pd.DataFrame(data)

# Fill missing values with 0
df_filled = df.fillna(0)

print(df_filled)
```

Output:

```
    A    B
0  1.0  0.0
1  2.0  2.0
2  0.0  3.0
3  4.0  4.0
```

Q3) c) Explain Min-Max Scaling. For the following dataset carry out min-max scaling:

Dataset: X = [24, 28, 53, 30, 40, 18, 15, 21]

Min-Max Scaling Explanation:

- Min-Max scaling rescales the data to a fixed range, typically [0,1].
- Formula for each value xx:

$$x_{\text{scaled}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

where xmin x_{\min} and xmax x_{\max} are the minimum and maximum values in the dataset.

Calculation:

- xmin =15x_{\min} = 15
- xmax =53x_{\max} = 53
- Range = 53−15=3853 - 15 = 38

Calculate scaled values:

Original xx	Calculation	Scaled xx
24	(24 - 15)/38 = 9/38	0.2368
28	(28 - 15)/38 = 13/38	0.3421
53	(53 - 15)/38 = 38/38	1.0000
30	(30 - 15)/38 = 15/38	0.3947
40	(40 - 15)/38 = 25/38	0.6579
18	(18 - 15)/38 = 3/38	0.0789
15	(15 - 15)/38 = 0/38	0.0000
21	(21 - 15)/38 = 6/38	0.1579

Final scaled dataset:

[0.2368,0.3421,1.0000,0.3947,0.6579,0.0789,0.0000,0.1579][0.2368, 0.3421, 1.0000, 0.3947, 0.6579, 0.0789, 0.0000, 0.1579]

If you'd like, I can also help with code snippets or graphs for these!

Here’s a structured answer for the given question (Q4 a & b) for 8 + 8 = **[16 marks]**, suitable for university exams:

Q4 a) Categorical Variable, Need for Encoding & One-Hot Encoding [8 Marks]

1. What is a Categorical Variable?

- A **categorical variable** is a variable that represents **discrete categories** or labels rather than numeric values.
- Examples: Gender (Male, Female), Color (Red, Green, Blue), City (Pune, Mumbai, Delhi).

2. Why Do You Need Categorical Variable Encoding?

- **Machine learning algorithms** work with **numerical data**, not text.
- Categorical data must be **converted into numerical format** so that models can process and learn from it.
- Encoding improves **model performance**, **accuracy**, and **interpretability**.

3. One-Hot Encoding (with Example):

- One-Hot Encoding converts each category into a **binary vector**.
- It creates **new columns** for each unique category and assigns 1 or 0.

Example:

City
Pune
Mumbai
Delhi

→ After One-Hot Encoding:

City_Pune	**City_Mumbai**	**City_Delhi**
1	0	0
0	1	0
0	0	1

4. Advantages:

- No ordinal relationship assumed between categories.
- Prevents models from interpreting category ranking incorrectly.

Q4 b) What is Data Wrangling? Need & Methods [8 Marks]

1. What is Data Wrangling?

- Data Wrangling is the **process of cleaning, structuring, and enriching raw data** into a desired format for better decision-making in less time.
- Also known as **Data Munging**.

2. Why is Data Wrangling Needed?

- Real-world data is often **incomplete, inconsistent, or messy**.
- Clean and organized data is crucial for:
    - Better analysis and visualization.
    - Improved machine learning performance.
    - Accurate business insights.

3. Data Wrangling Methods:

Method	Description
Data Cleaning	Handling missing, duplicate, or incorrect values.
Data Transformation	Converting data into suitable format (e.g., normalization, scaling).
Data Integration	Combining data from multiple sources.
Data Reduction	Reducing data volume while preserving key information.
Data Filtering	Removing unwanted data or outliers.
Data Validation	Ensuring data quality and consistency through checks.

4. Example:

- Raw data: ['10', 'Twenty', 30, '', '40']
- After wrangling: [10, 20, 30, 0, 40] (converted strings, filled missing, fixed types)

Let me know if you'd like a diagram or tabular form included in LaTeX format for your report or notes.

Certainly! Here's a detailed answer to your Q5:

Q5) a) How Data Visualization is important in Big Data? Explain challenges to big data visualization? [6]

Importance of Data Visualization in Big Data:

- **Insight Extraction:** Visualization helps to understand complex patterns, trends, and outliers in massive datasets.
- **Decision Making:** Provides an intuitive way for stakeholders to grasp key information quickly, aiding faster and better decisions.
- **Communication:** Simplifies communication of results and insights to non-technical users.
- **Exploration:** Enables interactive exploration of big data, helping users discover hidden relationships.
- **Error Detection:** Helps detect anomalies, errors, or inconsistencies in data.
- **Trend Identification:** Identifies trends over time or between variables effectively.

Challenges in Big Data Visualization:

- **Volume:** The huge size of data makes it difficult to load, process, and render visualizations in real-time.
- **Variety:** Different data types (structured, unstructured) need flexible visualization techniques.
- **Velocity:** High-speed data generation requires fast, often real-time visualization.
- **Scalability:** Visualization tools must scale efficiently to handle big data without performance loss.
- **Complexity:** Representing multidimensional, high-dimensional data meaningfully is challenging.
- **Data Quality:** Incomplete or noisy data can lead to misleading visualizations.
- **Resource Intensive:** Big data visualization demands significant computational and memory resources.

Q5) b) Explain various techniques for visual data representation. [6]

Some popular techniques for data visualization include:

1. **Bar Charts:** Compare quantities across categories using rectangular bars.
2. **Line Charts:** Show trends over time or continuous data.
3. **Pie Charts:** Represent proportions of a whole using slices.
4. **Scatter Plots:** Show relationships or correlations between two variables.

5. **Heatmaps:** Represent data intensity or density using color gradients.
6. **Histograms:** Show frequency distributions of data.
7. **Box Plots:** Display distribution summaries with quartiles, medians, and outliers.
8. **Tree Maps:** Represent hierarchical data as nested rectangles.
9. **Network Graphs:** Visualize relationships and connections between entities (nodes and edges).
10. **Geospatial Maps:** Plot data points on geographic maps.
11. **Dashboards:** Combine multiple visualizations for holistic data monitoring.

Q5) c) Explain the following data visualization techniques. [6]

i) Google Chart API

- A free, powerful, and easy-to-use JavaScript-based API by Google.
- Enables creation of interactive and customizable charts like line charts, bar charts, pie charts, scatter plots, geo charts, and more.
- Works seamlessly with web applications.
- Supports dynamic data loading and rich customization options.
- Advantages:
    - Cross-browser compatibility
    - Interactive features like zoom, tooltips, animations
    - Can be embedded easily with minimal coding

ii) D3.js (Data-Driven Documents)

- A powerful JavaScript library for creating complex and dynamic visualizations based on web standards (SVG, HTML, CSS).
- Allows binding arbitrary data to the Document Object Model (DOM) and applying data-driven transformations.
- Provides fine control over visual elements and interactions.
- Supports a wide variety of visualizations including custom charts, graphs, maps, and animations.
- Advantages:

    - Highly customizable and flexible
    - Supports animation and transitions
    - Can integrate with other web frameworks
- Used for advanced, interactive, and high-performance data visualizations.

If you want, I can provide examples or code snippets for Google Charts or D3.js too!

Here's a clear and structured answer for **Q6 a, b, c** with points and examples for easy understanding and exam use:

Q6 a) Data Visualization with respect to 1-D, 2-D, 3-D Data [6 Marks]

- **Data Visualization** is the graphical representation of data to help understand patterns, trends, and insights.

Dimension	**Explanation**	**Common Visualization Types**
1-D **Example:** Sales numbers per product category (only one variable).	Visualization of data with **one variable** (univariate data).	- **Histogram**- **Bar Chart**- **Pie Chart**
2-D **Example:** Plotting height vs weight of individuals.	Visualization involving **two variables** (bivariate data). Shows relationship between variables.	- **Scatter Plot**- **Line Chart**- **Heat Map**
3-D **Example:** Visualizing temperature, humidity, and time together.	Visualization involving **three variables** (multivariate). Displays complex relationships.	- **3D Scatter Plot**- **3D Surface Plot**- **Bubble Chart**

Q6 b) Analytical Techniques and Tools used in Data Visualization [6 Marks]

Analytical Techniques:

1. **Descriptive Analysis:**
    - Summarizes basic features of data (mean, median, mode).
    - Visualized using histograms, boxplots.
2. **Exploratory Data Analysis (EDA):**
    - Identifies patterns, anomalies.
    - Techniques: scatter plots, correlation matrices.
3. **Comparative Analysis:**
    - Compares groups or categories.
    - Tools: bar charts, grouped boxplots.
4. **Trend Analysis:**
    - Examines data over time.
    - Techniques: line charts, time series plots.
5. **Multivariate Analysis:**
    - Analyzes relationships among multiple variables.
    - Tools: heatmaps, 3D plots, pair plots.

Popular Tools for Data Visualization:

Tool	Description
Matplotlib	Python library for 2D plotting.
Seaborn	Built on Matplotlib, for statistical plots.
Tableau	User-friendly BI tool for interactive dashboards.
Power BI	Microsoft tool for data visualization and business analytics.
Excel	Basic charts and graphs, widely used.
Plotly	Interactive and 3D plotting in Python/JavaScript.

Q6 c) Boxplot: Example, Diagram & Usage [6 Marks]

What is a Boxplot?

- A **boxplot** (or box-and-whisker plot) summarizes a dataset using **five-number summary**:
    - Minimum
    - First quartile (Q1)
    - Median (Q2)
    - Third quartile (Q3)
    - Maximum

Boxplot Diagram:

```
  |-----|---------|------|---------|-----|
 Min   Q1     Median  Q3      Max
```

Example Data: [7, 8, 10, 12, 15, 18, 22, 25, 27, 30]

- Minimum = 7
- Q1 = 10 (25th percentile)
- Median = 16.5
- Q3 = 25 (75th percentile)
- Maximum = 30

Usage:

- Detects **outliers** (points outside whiskers).
- Shows **data spread** and **skewness**.
- Useful for comparing distributions across groups.
- Helps summarize data concisely.

If you want, I can provide a neat hand-drawn style boxplot image or LaTeX code for your notes. Would you like that?

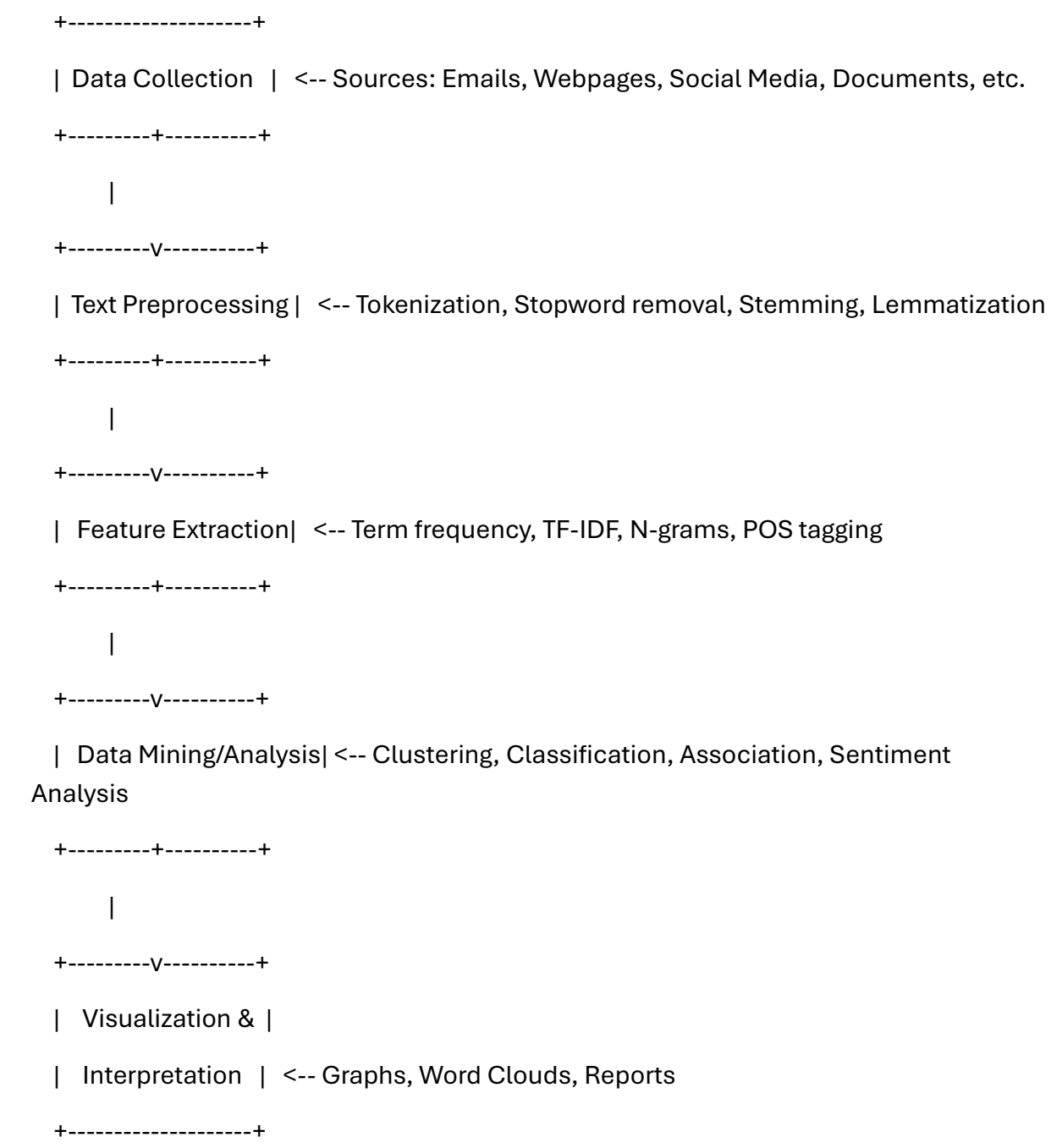
Sure! Here's a detailed answer for Q7 parts:

Q7) a) What is Text Mining? Draw and explain Text Mining Architecture and its use. [8]

What is Text Mining?

Text Mining (also called Text Data Mining or Text Analytics) is the process of extracting useful information, patterns, or knowledge from unstructured text data. It applies techniques from natural language processing (NLP), machine learning, and statistics to analyze textual data and convert it into structured, meaningful insights.

Text Mining Architecture:



Explanation of Components:

1. **Data Collection:** Gather raw text data from various sources such as social media, emails, articles, and databases.
2. **Text Preprocessing:** Clean and prepare text by removing noise, tokenizing into words or sentences, removing stopwords (common words like "the", "is"), stemming (reducing words to root form), and lemmatization.
3. **Feature Extraction:** Convert text into numerical features that algorithms can process. Common methods include TF-IDF (term frequency-inverse document frequency), bag of words, or word embeddings.
4. **Data Mining/Analysis:** Apply machine learning or statistical algorithms to discover patterns such as topic modeling, sentiment analysis, clustering, and classification.
5. **Visualization & Interpretation:** Present results in a human-understandable way using charts, word clouds, or summaries for decision-making.

Uses of Text Mining:

- Sentiment analysis of customer feedback or social media.
- Spam detection in emails.
- Topic detection in news articles.
- Fraud detection in documents.
- Market research and trend analysis.
- Enhancing search engines.

Q7) b) Explain Primary Activities of Michael Porter's Value Chain. [5]

Michael Porter's Value Chain describes a series of activities that an organization performs to deliver a valuable product or service. The **primary activities** are directly involved in creating and delivering the product:

1. **Inbound Logistics:** Receiving, storing, and distributing raw materials and components used in production.
2. **Operations:** Transforming inputs into final products through manufacturing or assembly.

3. **Outbound Logistics:** Distribution of finished products to customers, including warehousing and delivery.
4. **Marketing and Sales:** Activities that inform customers about products, generate demand, and facilitate purchases.
5. **Service:** Post-sale services like installation, repair, training, and customer support that maintain or enhance product value.

Q7) c) How Mobile Analytics is different from Social Media Analytics? [4]

Aspect	Mobile Analytics	Social Media Analytics
Focus	Tracks user behavior on mobile apps/devices	Analyzes data from social media platforms
Data Sources	App usage logs, mobile device data, location data	Social media posts, comments, likes, shares
Purpose	Improve app performance, user engagement, retention	Understand social trends, brand sentiment, customer opinions
Metrics	Session duration, app installs, crash rates	Sentiment analysis, influencer impact, engagement rates

---

If you want me to prepare diagrams or examples for these topics, just let me know!

Sure! Here's a detailed answer to your Q8:

Q8) a) What is Porter's Value Creation Model? Explain Porter's Value Chain Analysis. [9]

Porter's Value Creation Model:

- Developed by Michael Porter, the **Value Creation Model** explains how businesses create value for their customers and gain competitive advantage by performing activities efficiently and effectively.
- It emphasizes analyzing company activities to identify where value is added and how to improve these activities to increase profitability.

- The core idea is that value is created through a series of activities (the value chain), and managing these activities strategically leads to superior performance.

Porter's Value Chain Analysis:

- The **Value Chain** breaks down a company's activities into **primary** and **support** activities to analyze how value is created at each stage.

Primary Activities (directly add value):

1. **Inbound Logistics:** Receiving, storing, and distributing raw materials.
2. **Operations:** Converting inputs into finished products/services.
3. **Outbound Logistics:** Delivering products to customers.
4. **Marketing & Sales:** Promoting and selling products.
5. **Service:** Providing after-sales support and services.

Support Activities (indirectly add value by supporting primary activities):

1. **Firm Infrastructure:** Organizational structure, management, finance, legal support.
2. **Human Resource Management:** Recruiting, training, employee development.
3. **Technology Development:** R&D, process automation, product design.
4. **Procurement:** Sourcing and purchasing raw materials, equipment.

Purpose of Value Chain Analysis:

- Identify competitive advantages by optimizing value-adding activities.
- Reduce costs or increase differentiation.
- Improve customer satisfaction.
- Highlight areas for investment or improvement.

Q8) b) What is Social Media Analytics? Explain the process of Social Media Data Analytics.

What is Social Media Analytics?

- Social Media Analytics is the process of collecting, measuring, analyzing, and interpreting data generated from social media platforms like Facebook, Twitter, Instagram, LinkedIn, etc.
- It helps organizations understand customer sentiments, market trends, brand reputation, and user engagement to make informed decisions.

Process of Social Media Data Analytics:

1. **Data Collection:**
    - Gather data from social media sources through APIs, web scraping, or third-party tools.
    - Data includes posts, comments, likes, shares, hashtags, user profiles, etc.
2. **Data Cleaning & Preprocessing:**
    - Remove noise like spam, irrelevant posts.
    - Normalize text (lowercase, remove punctuation, stopwords).
    - Handle emojis, slang, and language variations.
3. **Data Storage:**
    - Store cleaned data in databases or data lakes for processing.
    - Often uses Big Data platforms due to large volumes.
4. **Data Analysis:**
    - Perform various analyses like sentiment analysis, trend detection, influencer identification, network analysis.
    - Use NLP, machine learning, and statistical techniques.
5. **Visualization & Reporting:**
    - Present insights through dashboards, charts, and reports.
    - Helps stakeholders understand social media performance and customer opinions.

6. **Decision Making:**
    - Use insights to adjust marketing strategies, improve products, manage reputation, or launch campaigns.

If you want, I can help prepare diagrams or give examples for these concepts!

PAPAR2

Certainly! Here's a detailed answer for Q1 parts:

Q1) a) Explain Google File System (GFS) and Its Advantages. [10]

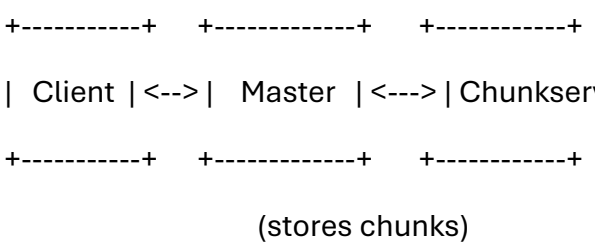
Google File System (GFS):

- **Google File System (GFS)** is a scalable distributed file system designed by Google to handle large data-intensive applications.
- It is optimized for **large files**, **high throughput**, and **fault tolerance** on commodity hardware.
- GFS is built to support **large-scale data processing**, especially for Google's own services like search indexing and analytics.

Key Features of GFS:

- **Master-Slave Architecture:** One Master server manages metadata and multiple Chunkservers store the actual data.
- **File Chunking:** Files are divided into fixed-size chunks (usually 64 MB).
- **Replication:** Each chunk is replicated (typically 3 copies) across different Chunkservers for fault tolerance.
- **Append-Only Writes:** Supports high throughput appending to files rather than random writes.
- **Fault Tolerance:** Master monitors Chunkservers with heartbeats, detects failures, and replicates lost chunks.
- **High Throughput:** Designed to optimize data throughput rather than low latency.
- **Consistency Model:** Provides relaxed consistency; supports atomic record appends.

GFS Architecture Diagram (Simplified):



Advantages of GFS:

1. **Scalability:** Can scale to thousands of servers and petabytes of data.
2. **Fault Tolerance:** Automatic replication and recovery from chunkserver failures.
3. **High Throughput:** Designed for large streaming reads and writes.
4. **Cost-Effective:** Runs on commodity hardware, reducing infrastructure costs.
5. **Simplified Data Management:** Master stores only metadata; chunkservers handle actual data.
6. **Support for Large Files:** Efficiently manages very large files by splitting into chunks.
7. **Optimized for Batch Processing:** Works well with data processing frameworks like MapReduce.

Q1) b) Explain Hadoop Distributed File System (HDFS). [8]

Hadoop Distributed File System (HDFS):

- HDFS is the distributed file system used by Hadoop to store big data across multiple commodity servers.
- It is designed for **high throughput**, **fault tolerance**, and **scalability**.
- HDFS stores large files by splitting them into blocks (default 128 MB), replicating blocks across multiple nodes.

Key Components:

- **NameNode:** The master server that manages the filesystem metadata, directory structure, and block locations.

- **DataNodes:** Slave servers that store the actual data blocks and serve read/write requests from clients.
- **Secondary NameNode:** Helps in checkpointing NameNode metadata (not a failover).

Working:

- When a client writes a file, HDFS splits it into blocks.
- Each block is replicated (default 3 copies) on different DataNodes.
- Client reads file blocks directly from DataNodes based on NameNode metadata.

Features of HDFS:

- **Fault Tolerance:** Through data replication.
- **High Throughput:** Suitable for batch processing, streaming reads/writes.
- **Scalability:** Can scale to thousands of nodes.
- **Data Locality:** Processing tasks (MapReduce) are scheduled close to data location.
- **Write Once, Read Many:** HDFS files are typically written once and read multiple times.
- **Reliability:** Heartbeat mechanism monitors DataNodes.

HDFS Architecture Diagram (Simplified):

```
+-----------+       +-----------+        +-----------+
|  Client   | <-----> |  NameNode | <--------> | DataNodes |
+-----------+       +-----------+        +-----------+
                        (store replicated blocks)
```

If you want, I can provide more details or diagrams for better clarity!

Here's a clear and detailed answer for **Q2 a & b** with explanations and examples for your exam preparation:

Q2 a) Why MapReduce is Required in Hadoop? Explain Stages of MapReduce Task with Example [9 Marks]

Why MapReduce is Required in Hadoop?

- Hadoop deals with **large-scale distributed data** stored in HDFS.
- Processing huge data in parallel on many nodes needs a **scalable, fault-tolerant** system.
- **MapReduce** is a programming model designed for:
    - **Distributed processing** of big data.
    - Automatically **splitting tasks** into small chunks (map tasks).
    - **Aggregating results** (reduce tasks).
- It simplifies complex data processing on large clusters.

Stages of MapReduce Task:

Stage	Description
1. Input Splitting	Input data is split into fixed-size chunks (blocks).
2. Map Phase	Mapper processes each input split and generates (key, value) pairs. Example: Counting words in a document. Each word is a key, count is value 1.
3. Shuffle and Sort	System automatically groups and sorts mapper outputs by keys, preparing for reduction.
4. Reduce Phase	Reducer aggregates values of each key. Example: Summing counts for each word.
5. Output	Final results are written back to HDFS.

Example: Word Count

- Input: Text documents.
- Map: (word, 1) for every word.
- Shuffle & Sort: Groups same words.
- Reduce: Sum counts → (word, total_count).

Q2 b) Types of NoSQL Databases with Examples and Comparison [9 Marks]

Type	Description	Example	Use Case
1. Key-Value Stores	Store data as key-value pairs, simple and fast.	Redis, DynamoDB	Session management, caching
2. Document Stores	Store data as JSON-like documents with flexible schema.	MongoDB, CouchDB	Content management, user profiles
3. Column-Family Stores	Store data in columns instead of rows, good for big data.	Cassandra, HBase	Analytics, time-series data
4. Graph Databases	Store data as nodes and edges representing relationships.	Neo4j, JanusGraph	Social networks, recommendation engines

Comparison:

Feature	Key-Value	Document	Column-Family	Graph
Data Model	Key-value pairs	JSON-like docs	Columns & column families	Nodes and relationships
Schema	Schema-less	Flexible schema	Schema optional	Flexible
Query Complexity	Simple key lookup	Query on documents	Complex queries possible	Complex graph traversal
Scalability	Highly scalable	Scalable	Highly scalable	Moderate scalability
Use Case	Simple, fast access	Semi-structured data	Big data, analytics	Relationship-heavy data

---

Let me know if you want a diagram of MapReduce or a comparison table in LaTeX for your notes!

Sure! Here's a detailed answer for Q3:

Q3) a) Explain Mean, Mode, Variance, and Standard Deviation with suitable example. [9]

1. Mean (Average):

- The mean is the sum of all values divided by the number of values.
- Formula:

Mean=∑i=1nxin\text{Mean} = \frac{\sum_{i=1}^n x_i}{n}

- **Example:** For data set [2,4,6,8,10][2, 4, 6, 8, 10],

Mean=2+4+6+8+105=305=6\text{Mean} = \frac{2 + 4 + 6 + 8 + 10}{5} = \frac{30}{5} = 6

---

2. Mode:

- The mode is the value that appears most frequently in the data set.
- **Example:** For data set [3,7,7,2,5][3, 7, 7, 2, 5], the mode is **7** because it appears twice.

---

3. Variance:

- Variance measures the spread of the data points from the mean.
- Formula:

Variance=∑i=1n(xi−Mean)2n\text{Variance} = \frac{\sum_{i=1}^n (x_i - \text{Mean})^2}{n}

- **Example:** For data set [2,4,6,8,10][2, 4, 6, 8, 10], mean = 6,

Variance=(2−6)2+(4−6)2+(6−6)2+(8−6)2+(10−6)25=16+4+0+4+165=405=8\begin{align*} \text{Variance} &= \frac{(2-6)^2 + (4-6)^2 + (6-6)^2 + (8-6)^2 + (10-6)^2}{5} \\ &= \frac{16 + 4 + 0 + 4 + 16}{5} = \frac{40}{5} = 8 \end{align*}

---

4. Standard Deviation:

- Standard deviation is the square root of variance and gives the spread in the original units.
- Formula:

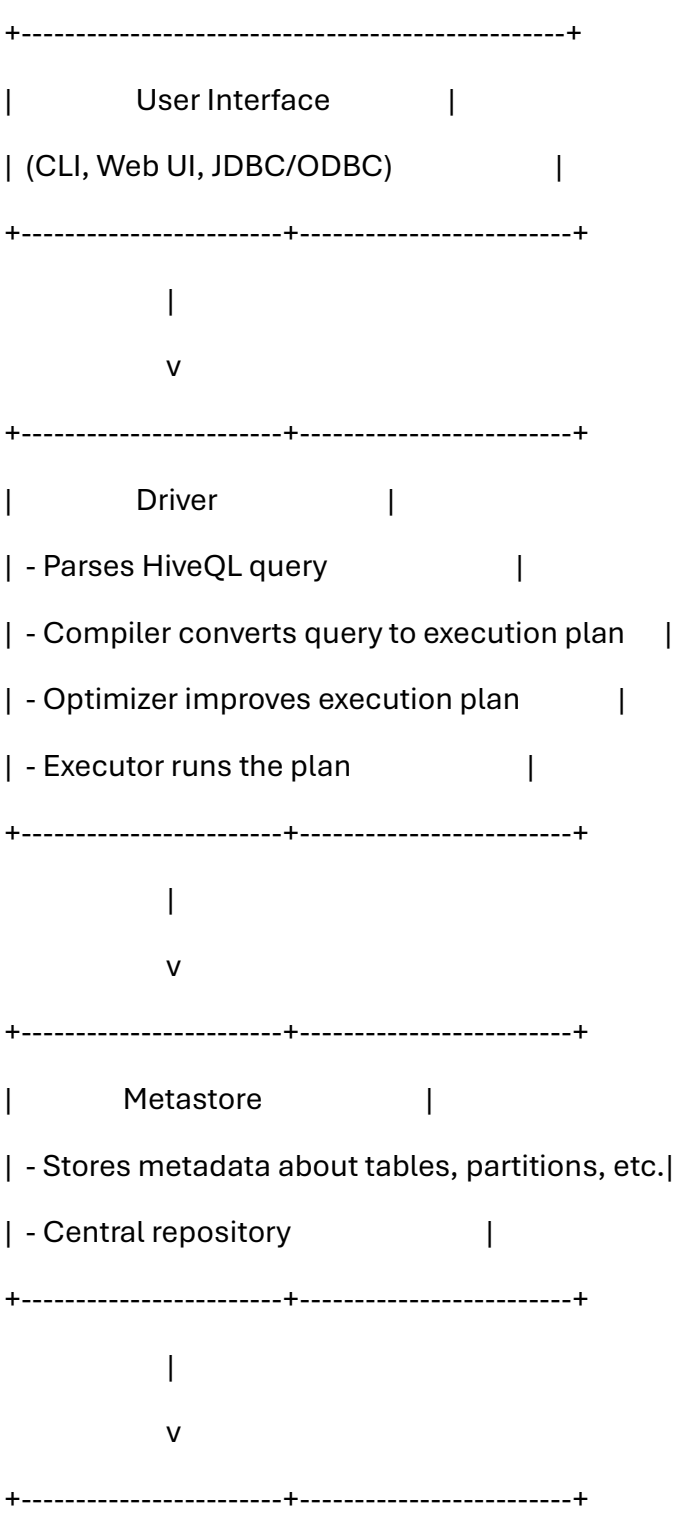
Standard Deviation=Variance\text{Standard Deviation} = \sqrt{\text{Variance}}

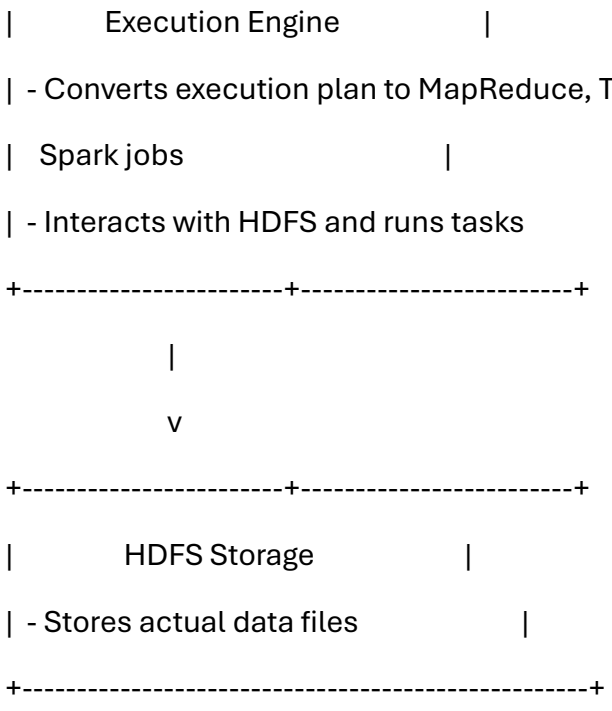
- **Example:** Using the variance 8 from above,

Standard Deviation=8≈2.83\text{Standard Deviation} = \sqrt{8} \approx 2.83

Q3) b) Draw and Explain Architecture of HIVE

Hive Architecture:





Explanation:

- **User Interface:** Allows users to submit Hive queries through CLI, web UI, or via JDBC/ODBC connectors.
- **Driver:** Manages the lifecycle of the query including parsing, compiling, optimizing, and executing the HiveQL query.
- **Metastore:** Stores metadata like schema, table definitions, partitions, and data location in a relational database. This is essential for query planning.
- **Execution Engine:** Converts the compiled query plan into a series of MapReduce (or Spark/Tez) jobs, which are then executed on Hadoop.
- **HDFS Storage:** The distributed file system where the actual data is stored in files. Hive reads and writes data here.

If you want, I can prepare a diagram image for the Hive architecture!

Here's a clear and structured answer for **Q4 a & b** suitable for exams:

Q4 a) Min-Max Scaling & Example Calculation [9 Marks]

What is Min-Max Scaling?

- Min-Max Scaling is a **normalization technique** to scale data into a fixed range, usually **0 to 1**.
- Formula:

$$X_{\text{scaled}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

- Ensures features contribute equally, improves convergence for ML algorithms.

Given Dataset:

$X = [24, 28, 53, 30, 40, 18, 15, 21]$

- $X_{\min} = 15$
- $X_{\max} = 53$

Min-Max Scaling Calculation:

Original X	Calculation	Scaled X
24	$(24-15)/(53-15) = 9/38$	0.237
28	$(28-15)/38 = 13/38$	0.342
53	$(53-15)/38 = 38/38$	1.000
30	$(30-15)/38 = 15/38$	0.395
40	$(40-15)/38 = 25/38$	0.658
18	$(18-15)/38 = 3/38$	0.079
15	$(15-15)/38 = 0/38$	0.000
21	$(21-15)/38 = 6/38$	0.158

Q4 b) Data Wrangling: Definition, Need & Methods [9 Marks]

What is Data Wrangling?

- The process of **cleaning, transforming, and preparing raw data** for analysis.
- Also called **data munging**.

Why Do You Need Data Wrangling?

- Raw data is often **incomplete, inconsistent, noisy, or messy**.

- Wrangling ensures **data quality, consistency, and accuracy**.
- Essential for **better analytics and machine learning results**.

Data Wrangling Methods:

Method	Description
Data Cleaning	Handle missing values, remove duplicates, correct errors.
Data Transformation	Normalize, scale, encode categorical variables.
Data Integration	Combine data from multiple sources.
Data Reduction	Reduce volume by aggregation or sampling.
Data Filtering	Remove outliers or irrelevant data points.
Data Validation	Verify data quality and consistency checks.

If you want, I can provide a Python code snippet to demonstrate min-max scaling or a diagram of data wrangling steps!

Sure! Here's a detailed answer for Q5:

Q5) a) Explain any 4 Types of Data Visualization with Example. [9]

1. **Bar Chart:**
    - Used to compare quantities of different categories.
    - Displays data as rectangular bars with lengths proportional to values.
    - **Example:** Comparing sales figures of different products.
2. **Line Chart:**
    - Shows trends over time by connecting data points with lines.
    - Ideal for time-series data analysis.
    - **Example:** Tracking stock prices over a year.
3. **Pie Chart:**
    - Displays data as slices of a circle representing parts of a whole.
    - Best for showing percentage or proportional data.

- **Example:** Market share of different companies in a sector.

4. **Scatter Plot:**
    - Shows relationship/correlation between two variables by plotting points.
    - Helps identify patterns or clusters.
    - **Example:** Analyzing the relationship between advertising spend and sales revenue.

Q5) b) Explain Different Data Visualization Tools. [9]

1. **Tableau:**
    - Popular commercial tool known for drag-and-drop interface.
    - Supports real-time data analytics and rich visualizations.
    - Can connect to multiple data sources.
    - Widely used in industry for dashboards and reports.
2. **Power BI:**
    - Microsoft's business analytics tool.
    - Integrates well with Microsoft products and cloud services.
    - Offers interactive visualizations and business intelligence capabilities.
    - Suitable for enterprise-scale reporting.
3. **D3.js (Data-Driven Documents):**
    - A JavaScript library for creating dynamic, interactive web-based visualizations.
    - Provides fine control over visuals with HTML, SVG, and CSS.
    - Requires programming knowledge but very flexible and powerful.
4. **Google Charts:**
    - Free tool to create simple, interactive charts on websites.
    - Supports various chart types like bar, pie, line, and maps.
    - Easy integration with Google products and simple API usage.
5. **QlikView/Qlik Sense:**
    - Self-service data visualization and discovery tools.

- Uses associative data indexing for fast analytics.
- Supports interactive dashboards and data storytelling.

6. **Matplotlib / Seaborn (Python libraries):**
    - Open-source libraries for statistical and scientific plotting.
    - Used widely in data science for static, animated, and interactive graphs.
    - Highly customizable and integrates with Python's data ecosystem.

If you want, I can help create example visualizations or recommend which tool suits your project best!

Here's a detailed and structured answer for **Q6 a & b** suitable for exams:

Q6 a) Data Visualization with Example & Advantages [9 Marks]

What is Data Visualization?

- Data visualization is the process of **representing data graphically** to help understand patterns, trends, and insights.
- It transforms raw data into **visual formats** like charts, graphs, and maps, making complex data easier to interpret.

Example of Data Visualization:

- Suppose we have monthly sales data for a store:

Month	Sales (in $)
Jan	2000
Feb	2500
Mar	2200
Apr	2700

- A **line chart** plotting months on the X-axis and sales on the Y-axis can clearly show trends over time, like rising or falling sales.

Advantages of Data Visualization:

1. **Simplifies Complex Data:** Makes large datasets understandable at a glance.

2. **Identifies Patterns & Trends:** Helps detect correlations and outliers.
3. **Facilitates Better Decision Making:** Visual insights guide strategic choices.
4. **Improves Communication:** Easy to share and explain data findings.
5. **Saves Time:** Faster interpretation compared to raw numbers.
6. **Supports Data Exploration:** Interactive visualizations enable drilling down into details.

Q6 b) Data Visualization with Tableau [9 Marks]

What is Tableau?

- Tableau is a **powerful and user-friendly data visualization tool** used for creating interactive and shareable dashboards.
- It connects easily to various data sources like Excel, SQL databases, and cloud services.

Features of Tableau:

- **Drag-and-drop Interface:** No programming needed to create charts.
- **Wide Range of Visualizations:** Bar charts, scatter plots, maps, heat maps, dashboards.
- **Real-Time Data Analysis:** Updates visualizations automatically with data changes.
- **Data Blending:** Combines data from multiple sources easily.
- **Interactive Dashboards:** Users can filter and drill down data interactively.
- **Sharing & Collaboration:** Export visualizations or publish online.

How Tableau Works (Basic Workflow):

1. **Connect to Data Source:** Import data from files or databases.
2. **Data Preparation:** Clean and organize data if needed.
3. **Create Visualizations:** Drag fields to rows and columns to create charts.
4. **Build Dashboards:** Combine multiple charts into an interactive dashboard.
5. **Share Results:** Publish dashboards or export reports.

If you want, I can prepare a sample Tableau dashboard screenshot or step-by-step workflow diagram for your notes. Would you like that?

Sure! Here's a detailed answer for Q7:

Q7) a) Explain Big Data Analytics Challenges in Brief. [9]

1. **Volume:**
    - The massive amount of data generated every second requires scalable storage and processing capabilities.
    - Managing and analyzing terabytes or petabytes of data efficiently is challenging.
2. **Velocity:**
    - Data is generated at high speed, especially from real-time sources like social media, sensors, and IoT devices.
    - Processing and analyzing streaming data with low latency is difficult.
3. **Variety:**
    - Data comes in various formats: structured (databases), semi-structured (JSON, XML), and unstructured (text, images, videos).
    - Integrating and analyzing heterogeneous data types require sophisticated tools.
4. **Veracity:**
    - Data quality and accuracy issues arise due to noise, inconsistencies, and incompleteness.
    - Ensuring reliability and trustworthiness of big data is a major challenge.
5. **Scalability:**
    - Systems must scale efficiently with growing data without performance degradation.
6. **Data Security and Privacy:**
    - Protecting sensitive data from breaches and misuse is complex, especially with distributed data storage.
7. **Data Integration:**

- Combining data from multiple sources and ensuring consistency is difficult.

8. **Skill Shortage:**
    - There is a lack of skilled professionals proficient in big data technologies and analytics.
9. **Cost:**
    - Infrastructure, tools, and skilled personnel for big data analytics can be expensive.

Q7) b) Explain Types of Mobile Analytics

Mobile analytics involves collecting and analyzing data from mobile apps and devices to understand user behavior, app performance, and marketing effectiveness. Types include:

1. **App Usage Analytics:**
    - Tracks how users interact with the app, session duration, frequency, user retention, and screen flow.
    - Helps improve user experience and app design.
2. **User Behavior Analytics:**
    - Analyzes user actions like clicks, taps, purchases, and navigation patterns.
    - Identifies popular features and drop-off points.
3. **Performance Analytics:**
    - Monitors app performance metrics like load time, crashes, errors, and network issues.
    - Essential for improving app stability and responsiveness.
4. **Marketing Analytics:**
    - Measures effectiveness of marketing campaigns by tracking user acquisition, conversion rates, and ROI.
    - Includes attribution modeling and user segmentation.
5. **Location Analytics:**

- Analyzes user location data to provide location-based services and personalized content.
- Used for targeted advertising and user engagement.

6. **Monetization Analytics:**
   - Tracks in-app purchases, subscriptions, ad revenue, and overall revenue metrics.
   - Helps optimize revenue streams.

If you want, I can provide examples or tools commonly used for mobile analytics!

Here's a detailed and structured answer for **Q8 a & b** with clear explanations:

Q8 a) Porter's Value Creation Model & Value Chain Analysis [9 Marks]

What is Porter's Value Creation Model?

- Porter's Value Creation Model explains how a company creates **competitive advantage** by performing activities that add value to its products or services.
- The focus is on **value addition at each step**, which increases customer satisfaction and profitability.

Porter's Value Chain Analysis:

- A tool to analyze internal activities of a firm to understand how value is created and where improvements can be made.
- Divides business activities into **Primary Activities** and **Support Activities**.

Primary Activities	Description
Inbound Logistics	Receiving, storing, and distributing inputs (raw materials).
Operations	Transforming inputs into finished products.
Outbound Logistics	Distributing finished goods to customers.
Marketing & Sales	Activities to promote and sell products.
Service	After-sale services like support and maintenance.

Support Activities	Description
Procurement	Acquiring raw materials and resources.
Technology Development	Innovation, R&D to improve products or processes.
Human Resource Management	Recruiting and training employees.
Firm Infrastructure	Organizational structure, finance, planning, and control.

- **Goal:** Optimize activities to create maximum value at minimum cost, leading to competitive advantage.

Q8 b) Social Media Analytics & Process of Social Media Data Analytics [9 Marks]

What is Social Media Analytics?

- Social Media Analytics is the process of **collecting, analyzing, and interpreting data** from social media platforms (like Facebook, Twitter, Instagram) to gain insights about trends, customer behavior, brand reputation, and marketing effectiveness.

Process of Social Media Data Analytics:

Step	Description
1. Data Collection	Gather data from social platforms using APIs, scraping, or third-party tools.
2. Data Cleaning & Preparation	Remove noise, duplicates, and irrelevant data to ensure quality.
3. Data Storage	Store cleaned data in databases or cloud storage.
4. Data Analysis	Use techniques like sentiment analysis, trend analysis, network analysis.
5. Visualization & Reporting	Create dashboards, charts to summarize findings and support decision-making.
6. Actionable Insights	Use insights to improve marketing strategies, product development, or customer engagement.

Let me know if you want diagrams for Porter's Value Chain or a flowchart for Social Media Analytics!

PAPAR 3

Here's a detailed and structured answer for **Q1 a, b, c** with explanations and examples:

Q1 a) Big Data Ecosystem with Diagram [7 Marks]

What is Big Data Ecosystem?

- A Big Data ecosystem is a collection of tools, technologies, and frameworks that work together to **store, process, analyze, and visualize** large-scale data.

Components of Big Data Ecosystem:

Component	Description
Data Sources	Social media, sensors, logs, transactions, etc.
Data Ingestion	Tools to import data (e.g., Kafka, Flume)
Storage	Distributed storage systems (e.g., HDFS, NoSQL DB)
Processing	Batch & real-time processing (MapReduce, Spark)
Data Management	Metadata, data catalog (e.g., Hive, HBase)
Analytics & BI	Data analysis and visualization tools (e.g., Tableau, Pig)
Security & Governance	Authentication, authorization, auditing

Diagram:

Data Sources → Data Ingestion → Storage → Processing → Analytics & Visualization

↑ ↓

Security & Governance <----------

Q1 b) Anatomy of File Read and Write in HDFS [7 Marks]

File Write in HDFS:

1. Client contacts **NameNode** to create a file.
2. NameNode checks permissions and returns a list of DataNodes for block storage.

3. Client writes data in **blocks** to DataNodes, replicated as per replication factor.
4. DataNodes acknowledge to Client and NameNode updates metadata.

File Read in HDFS:

1. Client contacts **NameNode** to get block locations of the file.
2. NameNode responds with DataNodes holding replicas.
3. Client reads blocks from nearest DataNode.
4. Data is reassembled and provided to the client application.

Q1 c) Two Hadoop Shell Commands with Explanation [4 Marks]

Command	Explanation
hdfs dfs -ls /path	Lists files and directories at the specified HDFS path.
hdfs dfs -put localfile /hdfs/path	Uploads a file from local filesystem to HDFS directory.

Let me know if you want a graphical diagram or code examples for Hadoop commands!

Certainly! Here's a detailed answer for Q2 parts:

Q2) a) Explain MapReduce with proper diagram for Word Count example. [7]

MapReduce Overview:

- MapReduce is a programming model used for processing large data sets in a distributed environment.
- It consists of two main phases: **Map** and **Reduce**.
- The **Map** function processes input data and produces intermediate key-value pairs.
- The **Reduce** function aggregates intermediate data to produce final results.

Word Count Example:

- **Objective:** Count the number of occurrences of each word in a text.

Steps:

1. **Map Phase:**
    - Input: Lines of text.
    - Operation: Split each line into words.
    - Output: Emit each word as a key with value 1.
      Example: ("word", 1)
2. **Shuffle and Sort Phase:**
    - Groups all intermediate values by keys (words).
    - Prepares data for Reduce phase.
3. **Reduce Phase:**
    - Input: Key (word) and list of values (counts).
    - Operation: Sum the counts for each word.
    - Output: (word, total count)

Diagram:

Input Text:

"The cat and the dog"

Map Phase:

"The" -> ("The", 1)

"cat" -> ("cat", 1)

"and" -> ("and", 1)

"the" -> ("the", 1)

"dog" -> ("dog", 1)

Shuffle and Sort:

("and", [1])

("cat", [1])

("dog", [1])

("the", [1, 1])

Reduce Phase:

("and", 1) -> ("and", 1)

("cat", 1) -> ("cat", 1)

("dog", 1) -> ("dog", 1)

("the", 2) -> ("the", 2)

Output:

and 1

cat 1

dog 1

the 2

Q2) b) Explain Google File System. [7]

- **Google File System (GFS)** is a scalable, distributed file system developed by Google for large data-intensive applications.
- It is designed to run on commodity hardware and handle large files with high fault tolerance and reliability.
- **Architecture:**
    - **Master Server:** Manages metadata (file namespace, access control, chunk locations).
    - **Chunkservers:** Store fixed-size chunks of data (usually 64 MB).
    - Files are split into chunks and replicated across chunkservers.
- **Key Features:**
    - Automatic chunk replication (default 3 copies) for fault tolerance.
    - Optimized for large streaming reads and appends.
    - Heartbeat and monitoring for detecting chunkserver failures.

- Relaxed consistency with atomic record append.

Q2) c) Explain ETL Processing. [4]

- **ETL** stands for **Extract, Transform, Load**.
- It is a data integration process used to collect data from multiple sources, transform it into a suitable format, and load it into a target database or data warehouse.

Steps:

1. **Extract:**
   - Data is collected from different sources such as databases, files, APIs.
2. **Transform:**
   - Data is cleaned, filtered, formatted, and transformed (e.g., data type conversions, aggregation).
3. **Load:**
   - Transformed data is loaded into the target system (data warehouse, database).

ETL is crucial for data warehousing and analytics to ensure high-quality, consistent data.

If you'd like, I can also provide a diagram or example code snippets for MapReduce!

Sure! Here's a detailed answer for Q3 parts:

Q3) a) Explain Different Steps in Data Analytics Project Life Cycle [7]

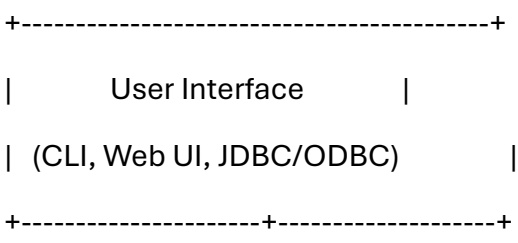
The Data Analytics Project Life Cycle consists of the following key steps:

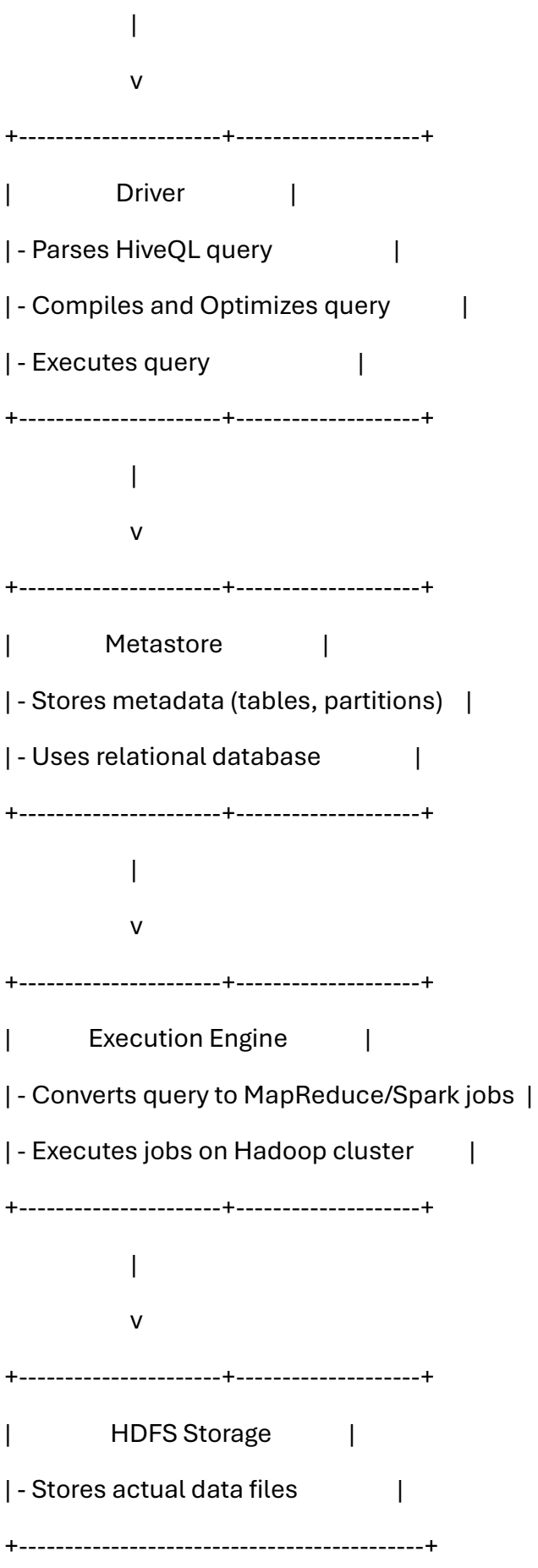
1. **Problem Definition:**
   - Clearly define the business problem or question that needs to be solved.
   - Understand objectives and success criteria.

2. **Data Collection:**
    - Gather data from various sources (databases, files, APIs, sensors).
    - Ensure data is relevant and sufficient.
3. **Data Cleaning and Preparation:**
    - Handle missing values, outliers, and errors.
    - Convert data into usable formats.
    - Feature engineering to create meaningful variables.
4. **Exploratory Data Analysis (EDA):**
    - Analyze data patterns, distributions, correlations.
    - Visualize data to gain insights.
5. **Model Building:**
    - Choose appropriate algorithms and techniques (e.g., regression, classification).
    - Train models on training datasets.
6. **Model Evaluation:**
    - Test model accuracy and performance using validation/test data.
    - Use metrics like accuracy, precision, recall, RMSE.
7. **Deployment and Monitoring:**
    - Deploy the model in production.
    - Monitor performance and update as needed.

Q3) b) Draw and Explain Architecture of HIVE [7]

Hive Architecture Diagram (Simplified):




Driver
- Parses HiveQL query
- Compiles and Optimizes query
- Executes query
Metastore
- Stores metadata (tables, partitions)
- Uses relational database
Execution Engine
- Converts query to MapReduce/Spark jobs
- Executes jobs on Hadoop cluster
HDFS Storage
- Stores actual data files

Explanation:

- **User Interface:** Users submit Hive queries through command line, web UI, or JDBC/ODBC connectors.
- **Driver:** Manages the query lifecycle — parsing, compiling, optimization, and execution.
- **Metastore:** Central repository for metadata like schema and data location. It uses a traditional RDBMS.
- **Execution Engine:** Translates the HiveQL query plan into MapReduce, Tez, or Spark jobs and executes them.
- **HDFS:** The distributed storage system where actual data resides.

Q3) c) Explain Different Data Transformation Techniques

Data transformation is the process of converting data from one format or structure into another. Techniques include:

1. **Data Cleaning:**
    - Removing or correcting inaccurate, corrupted, or irrelevant data.
2. **Normalization:**
    - Scaling data to a standard range, e.g., 0 to 1, to remove units and enable comparison.
3. **Aggregation:**
    - Summarizing data by grouping and computing aggregates like sum, average, count.
4. **Discretization:**
    - Converting continuous data into categorical or discrete bins.
5. **Encoding:**
    - Transforming categorical variables into numeric form (e.g., One-Hot Encoding, Label Encoding).
6. **Data Integration:**
    - Combining data from different sources into a unified dataset.

7. **Data Reduction:**
    - Reducing data volume via dimensionality reduction (PCA), sampling, or feature selection.
8. **Feature Engineering:**
    - Creating new features based on existing data to improve model performance.

If you want, I can prepare diagrams or provide examples for any of these points!

Here's a concise and clear answer for **Q4 a, b, c**:

Q4 a) Different Kinds of Big Data Analysis [7 Marks]

1. **Descriptive Analysis:**
    - Summarizes past data to understand what happened.
    - Tools: Reports, dashboards, histograms.
2. **Diagnostic Analysis:**
    - Explains why something happened by finding correlations and patterns.
    - Techniques: Drill-down, data discovery.
3. **Predictive Analysis:**
    - Uses historical data and statistical models to predict future outcomes.
    - Methods: Machine learning, regression.
4. **Prescriptive Analysis:**
    - Suggests actions based on predictions to optimize outcomes.
    - Techniques: Optimization algorithms, simulations.
5. **Exploratory Analysis:**
    - Investigates data to find new patterns or hypotheses.
    - Tools: Visualization, clustering.

Q4 b) Data Ingestion in Python & Syntax [7 Marks]

- Data ingestion means **importing data from various sources** into Python for analysis.

Common methods for data ingestion in Python:

Source	Syntax Example
CSV file	python import pandas as pd data = pd.read_csv('file.csv')
Excel file	python data = pd.read_excel('file.xlsx')
JSON file	python data = pd.read_json('file.json')
Database (SQL)	python import sqlite3 conn = sqlite3.connect('db.db') data = pd.read_sql_query("SELECT * FROM table", conn)

Q4 c) Role of Visualization in Big Data Analytics [3 Marks]

- Helps **interpret complex data** by representing it graphically.
- Enables **quick identification** of trends, outliers, and patterns.
- Facilitates **better communication** of insights to stakeholders.

Let me know if you want example Python code for data ingestion or some visualization libraries!

Sure! Here's a detailed answer for Q5:

Q5) a) Explain Different Techniques of Big Data Visualization [7]

1. **Charts and Graphs:**
    - Bar charts, line charts, pie charts, and scatter plots are common ways to summarize data visually, making patterns and trends easier to spot.
2. **Heatmaps:**
    - Represent data intensity or frequency through color gradients. Useful for showing correlation matrices or density in data.
3. **Geospatial Visualization:**
    - Maps and spatial plots help visualize location-based data like customer distribution or sensor readings.

4. **Dashboards:**
    - Combine multiple visualizations and KPIs in one interface to provide a comprehensive real-time overview.
5. **Network Graphs:**
    - Show relationships between entities using nodes and edges, used in social networks or fraud detection.
6. **Tree Maps:**
    - Display hierarchical data as nested rectangles; the size and color represent data attributes.
7. **Parallel Coordinates:**
    - Visualize multi-dimensional data by plotting each variable on parallel axes and connecting data points across axes.

Q5) b) Explain Challenges in Big Data Visualization [7]

1. **Volume and Scalability:**
    - Huge datasets make it difficult to process and render visualizations quickly without lag.
2. **Variety:**
    - Data comes in different formats (text, images, videos), complicating visualization.
3. **Velocity:**
    - Real-time streaming data requires fast updating visualizations, which is computationally intensive.
4. **Complexity:**
    - High-dimensional data is hard to visualize in two or three dimensions effectively.
5. **Data Quality:**
    - Incomplete or noisy data can mislead visual analysis.
6. **Interpretability:**
    - Visualizations need to be intuitive and easy to understand for non-experts.

7. **Resource Constraints:**
    - Visualization tools may require significant computing power and storage.

Q5) c) Write Two Data Visualization Functions from Matplotlib [3]

1. **plt.plot()**
    - Used to create line plots.
2. import matplotlib.pyplot as plt
3. plt.plot([1, 2, 3, 4], [10, 20, 25, 30])
4. plt.show()
5. **plt.bar()**
    - Used to create bar charts.
6. plt.bar(['A', 'B', 'C'], [5, 7, 3])
7. plt.show()

If you want, I can provide code examples or visual samples!

Here's a structured answer for **Q6 a, b, c** with clear explanations:

Q6 a) Different Tools for Data Visualization [7 Marks]

Tool	Description	Use Case
Tableau	Interactive dashboard creation, drag-and-drop interface.	Business intelligence and reporting.
Power BI	Microsoft's tool for data visualization and sharing.	Enterprise analytics and dashboards.
Matplotlib	Python library for 2D plotting and graphs.	Custom plots in Python scripts.
Seaborn	Python library built on Matplotlib, specialized for statistical graphics.	Statistical visualizations in Python.

Tool	Description	Use Case
D3.js	JavaScript library for creating dynamic, interactive web graphics.	Web-based custom visualizations.
QlikView	Self-service data visualization and guided analytics.	Business dashboards and reporting.
Google Data Studio	Cloud-based data visualization and reporting.	Google ecosystem analytics sharing.

Q6 b) Scatter Plot, Histogram and Heat Map with Examples [7 Marks]

Visualization	Explanation	Example
Scatter Plot	Plots points on X and Y axes to show relationship between two variables.	Plot height vs weight of individuals to see correlation.
Histogram	Shows frequency distribution of a single variable by grouping data into bins.	Plotting age distribution of a population.
Heat Map	Uses color gradients to represent values in a matrix format, showing intensity or correlation.	Correlation matrix heatmap showing relationships between variables.

Q6 c) Two Data Visualization Functions from Seaborn [3 Marks]

Function	Description
sns.scatterplot()	Creates scatter plots to visualize relationships between two variables.
sns.heatmap()	Creates heatmaps to visualize data intensity or correlation matrices.

If you want, I can also provide Python code examples for these visualizations!

Sure! Here's a detailed answer for Q7:

Q7) a) How Social Media Analytics Helps in Value Creation? Explain with Suitable Examples. [7]

Social Media Analytics (SMA) involves collecting, measuring, and analyzing social media data to gain insights that create business value. Here's how it helps in value creation:

1. **Customer Insights:**
    - Understand customer preferences, opinions, and behavior through sentiment analysis and trend detection.
    - *Example:* A company analyzing Twitter mentions to gauge customer sentiment about a product launch.
2. **Brand Management:**
    - Monitor brand reputation in real-time and manage public relations proactively.
    - *Example:* Detecting and responding quickly to negative comments or viral issues.
3. **Marketing Effectiveness:**
    - Measure the performance of marketing campaigns by tracking engagement, reach, and conversion.
    - *Example:* Using Facebook analytics to optimize ad targeting and increase ROI.
4. **Product Development:**
    - Gain feedback and feature requests from social media to guide product improvements.
    - *Example:* A software company identifying popular feature requests via online forums and social platforms.
5. **Competitive Analysis:**
    - Track competitor activities and customer reactions to benchmark and strategize.
    - *Example:* Monitoring competitor hashtags and campaigns on Instagram to spot opportunities.
6. **Crisis Management:**
    - Quickly detect and mitigate PR crises by analyzing spikes in negative sentiment or unusual activity.

- *Example:* A brand responding to a product recall by tracking social chatter and addressing concerns.

Q7) b) Explain in Brief Data Analytics Life Cycle. [7]

The Data Analytics Life Cycle is a series of steps followed to extract actionable insights from data:

1. **Discovery:**
    - Define the business problem and identify analytics objectives.
2. **Data Preparation:**
    - Collect, clean, transform, and organize data for analysis.
3. **Model Planning:**
    - Select suitable analytical techniques and develop hypotheses.
4. **Model Building:**
    - Train and validate models using statistical or machine learning methods.
5. **Communicate Results:**
    - Visualize findings and present insights to stakeholders.
6. **Operationalize:**
    - Deploy the model into production and monitor its performance.
7. **Feedback & Iteration:**
    - Use feedback to refine models and improve processes continuously.

Q7) c) Explain Big Data Value Terminology. [4]

1. **Volume:**
    - The amount of data generated and stored, ranging from terabytes to petabytes.
2. **Velocity:**
    - The speed at which data is generated, processed, and analyzed in real-time or near real-time.
3. **Variety:**

- Different types and sources of data including structured, semi-structured, and unstructured formats.

4. **Veracity:**
    - The trustworthiness and quality of data, ensuring accuracy and reliability.

These "4 Vs" define the core challenges and potential value in big data analytics.

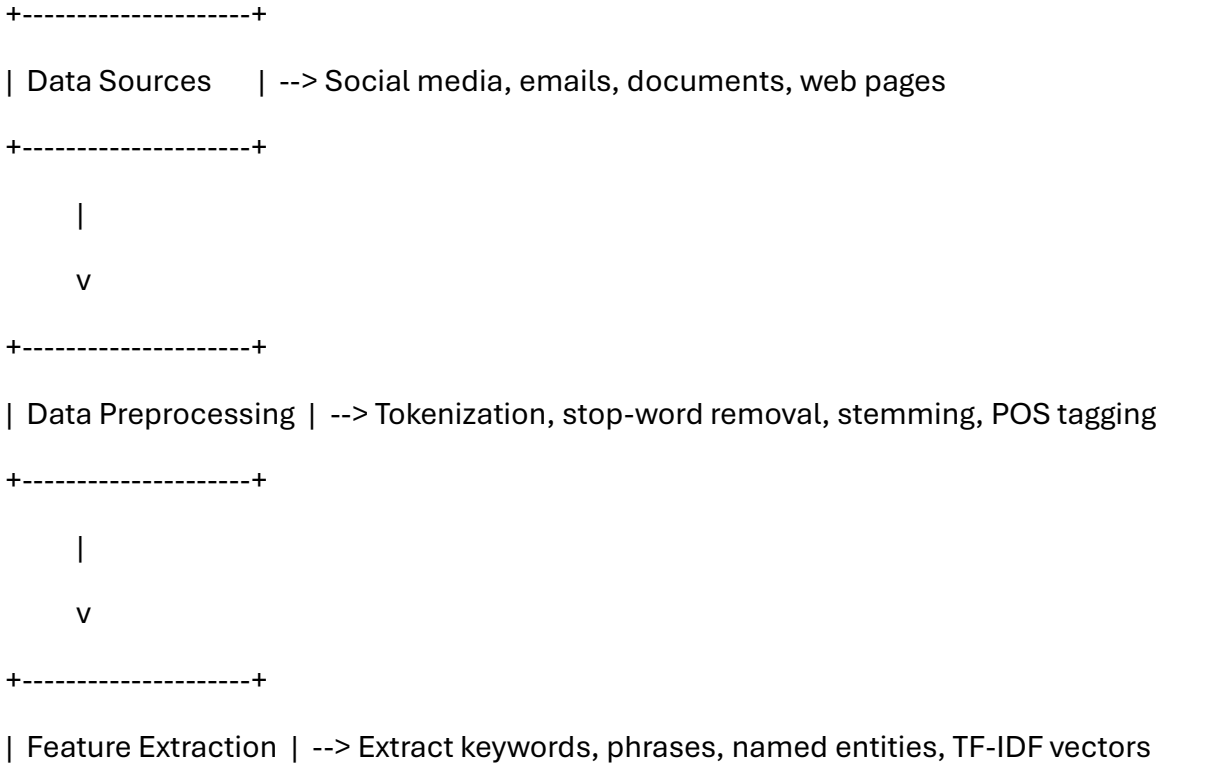
If you want, I can provide examples of tools used in social media analytics or details on each life cycle phase!

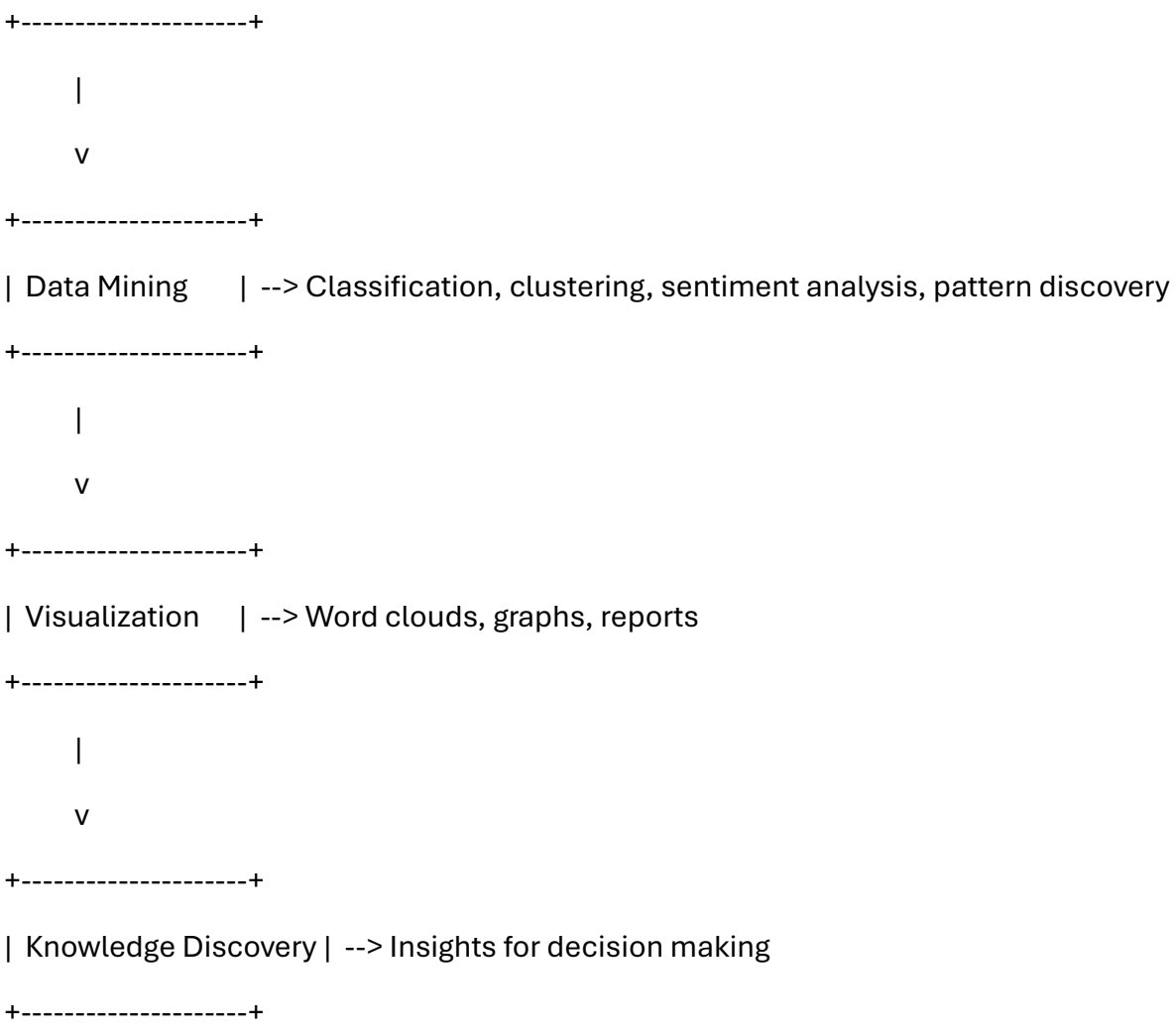
Sure! Here's a detailed answer for Q8:

Q8) a) What is Text Mining? Draw and Explain Text Mining Architecture and Its Use. [7]

Text Mining is the process of extracting useful information and knowledge from unstructured text data using techniques from natural language processing (NLP), machine learning, and statistics.

Text Mining Architecture:





Explanation:

1. **Data Sources:** Collect text from multiple sources like emails, social media, articles, etc.
2. **Data Preprocessing:** Clean and prepare text by removing noise.
3. **Feature Extraction:** Convert text into numerical features that algorithms can process.
4. **Data Mining:** Apply algorithms to extract patterns or classify text.
5. **Visualization:** Present results in an understandable format.
6. **Knowledge Discovery:** Use insights for business or research decisions.

Uses of Text Mining:

- Sentiment analysis of customer reviews.

- Spam detection in emails.
- Topic modeling for research papers.
- Social media monitoring.
- Fraud detection from textual data.

Q8) b) Explain Big Data Analytics in Research. [7]

- **Big Data Analytics in Research** refers to the use of large-scale data processing and advanced analytical techniques to discover patterns, correlations, and insights in research data.

Key points:

- **Handling large datasets:** Researchers can analyze massive data collected from experiments, sensors, simulations, or social media.
- **Cross-disciplinary insights:** Combines data from diverse fields for holistic understanding.
- **Accelerates discovery:** Automates pattern detection and hypothesis testing, speeding up research.
- **Examples:**
    - Genomics: Analyzing genetic data for disease research.
    - Climate Science: Processing sensor and satellite data to study environmental changes.
    - Social Sciences: Studying social behavior by analyzing social media and communication data.

Q8) c) Explain Big Data Impact on Organizations

1. **Improved Decision Making:**
    - Organizations use data-driven insights to make informed strategic and operational decisions.
2. **Enhanced Customer Experience:**
    - Personalized services and targeted marketing based on customer data.
3. **Operational Efficiency:**

- Streamlining processes by analyzing workflow data to reduce costs and improve productivity.

4. **Innovation and New Business Models:**
    - Big data enables the development of new products, services, and business strategies.
5. **Competitive Advantage:**
    - Organizations leveraging big data can outperform competitors by faster and smarter responses.
6. **Risk Management:**
    - Detecting fraud, predicting failures, and managing compliance more effectively.

If you want, I can help with diagrams or examples for specific parts!