

# **Big Data and AI-ML Top 20 Interview Questions**

## **Big Data / AI-ML Interview Questions (20)**

### **1. Explain the difference between Big Data and traditional data processing systems.**

Big Data systems handle high volume, velocity, and variety of data using distributed architectures, whereas traditional systems rely on centralized databases and scale vertically.

### **2. What problem does Hadoop solve, and what are its core components?**

Hadoop enables distributed storage and processing of large datasets. Core components are HDFS (storage), YARN (resource management), and MapReduce (processing).

### **3. How does Apache Spark differ from Hadoop MapReduce?**

Spark processes data in-memory, making it significantly faster than disk-based MapReduce. It also supports batch, streaming, ML, and graph workloads in one engine.

### **4. What is HDFS, and why is data replication important?**

HDFS is a distributed file system designed for fault tolerance. Replication ensures data availability even if nodes fail.

### **5. Explain RDD, DataFrame, and Dataset in Spark.**

RDD is low-level and immutable, DataFrame is optimized with schema and Catalyst optimizer, and Dataset provides type safety with performance benefits.

### **6. What is data partitioning, and why is it important in Big Data?**

Partitioning splits data across nodes to enable parallel processing, reduce data shuffling, and improve performance.

### **7. What is Kafka, and where is it used?**

Kafka is a distributed event streaming platform used for real-time data ingestion, messaging, and streaming pipelines.

### **8. Explain the role of Apache Hive in a Big Data ecosystem.**

Hive provides SQL-like querying (HiveQL) on large datasets stored in HDFS, enabling easier data analysis without writing complex MapReduce code.

## **9. What is the difference between batch processing and stream processing?**

Batch processing handles large volumes of historical data, while stream processing handles real-time, continuously arriving data.

## **10. What are features and labels in Machine Learning?**

Features are input variables used for prediction, while labels are the target outputs the model learns to predict.

## **11. Explain overfitting and underfitting in ML models.**

Overfitting occurs when a model learns noise and performs poorly on new data, while underfitting occurs when the model is too simple to capture patterns.

## **12. What is the difference between supervised and unsupervised learning?**

Supervised learning uses labeled data, while unsupervised learning finds patterns in unlabeled data.

## **13. How does gradient descent work?**

Gradient descent iteratively updates model parameters by minimizing the loss function using the direction of steepest descent.

## **14. What evaluation metrics would you use for a classification problem?**

Common metrics include accuracy, precision, recall, F1-score, and ROC-AUC, depending on class imbalance and business needs.

## **15. Why is feature scaling important?**

Feature scaling ensures all features contribute equally to the model, improving convergence speed and model performance.

## **16. What role do NumPy and Pandas play in ML workflows?**

NumPy provides numerical computing support, while Pandas is used for data manipulation, cleaning, and analysis.

## **17. What is a data pipeline?**

A data pipeline automates data ingestion, transformation, processing, and storage from source systems to analytics or ML models.

## **18. How does Docker help in ML or Big Data projects?**

Docker ensures consistent environments, simplifies deployment, and improves reproducibility across development and production.

## **19. What is CI/CD, and why is it useful in data engineering or ML?**

CI/CD automates testing, integration, and deployment, reducing errors and speeding up model and pipeline releases.

## **20. How would you approach learning a new Big Data or ML tool quickly?**

Start with core concepts, follow official docs, build small hands-on projects, analyze real datasets, and gradually scale complexity.