Certainly! Here's a breakdown of each topic, written from the perspective of an expert in AI, Machine Learning, and Data Science. Each section gives context on where these tools/technologies fit within the data science lifecycle and their practical applications:

**1. Use Hadoop & Spark to store and process data at scale (using MapReduce and Spark)**

* **Hadoop** and **Spark** are **big data frameworks** used to process massive datasets across distributed systems.
* **Hadoop** uses **MapReduce**, a programming model for batch processing, while **Spark** uses **in-memory computation**, offering faster performance for iterative ML workloads.
* **Used in**: Data Engineering / Data Processing  
  **Scope**: Storage, ETL, and distributed computation  
  **Phase**: Data Collection, Cleaning, and Preprocessing

**2. Use Hive, Impala and Spark SQL to analyze data at scale (using Hive Query Language, Scala, Python)**

* **Hive** and **Impala** provide SQL-like querying over big datasets stored in Hadoop; Spark SQL allows querying structured data using SQL in Spark.
* These tools bridge the gap between data warehouses and data lakes, enabling structured queries on semi-structured and structured data.
* **Used in**: Data Analysis / Query Optimization  
  **Scope**: Analytical queries on large-scale datasets  
  **Phase**: Exploratory Data Analysis (EDA), Feature Engineering

**3. Use Pig to analyze unstructured data at scale (using Pig Latin language)**

* **Apache Pig** is a high-level platform that uses **Pig Latin**, a scripting language designed for processing and analyzing large unstructured data sets.
* It simplifies the programming effort required with MapReduce and is suited for ETL jobs involving logs or text data.
* **Used in**: Data Processing of raw data  
  **Scope**: Unstructured data transformation  
  **Phase**: Data Wrangling / Preprocessing

**4. Improve querying data time (using Avro and Parquet file formats)**

* **Avro** and **Parquet** are efficient data serialization formats: Avro is row-based and ideal for write-heavy operations, while Parquet is columnar, making it great for read-heavy analytic workloads.
* They optimize storage and retrieval time, enabling faster data access and reduced I/O operations.
* **Used in**: Data Storage / I/O Optimization  
  **Scope**: Data serialization in big data systems  
  **Phase**: Data Storage, Preprocessing, Feature Extraction

**5. Import and Export data at scale (using Sqoop)**

* **Apache Sqoop** is used to transfer bulk data between Hadoop and relational databases (like MySQL, Oracle).
* It automates import/export of structured data, enabling seamless movement for analysis in big data platforms.
* **Used in**: Data Integration / ETL  
  **Scope**: Data pipeline for structured databases  
  **Phase**: Data Collection / Ingestion

**6. Install and configure ODBC/JDBC connectors to connect third-party tools (MS Excel, Tableau Software, MicroStrategy, etc.)**

* **ODBC/JDBC connectors** allow data scientists to link databases and big data engines with visualization or BI tools.
* These connectors enable analysts to query and visualize data stored in platforms like Hive, Impala, or Spark.
* **Used in**: Business Intelligence / Visualization  
  **Scope**: Database connectivity for dashboards  
  **Phase**: Post-analysis / Reporting

**7. Build Nifi Data Flows to ingest, transform and route data at scale**

* **Apache NiFi** is a visual tool for building scalable data pipelines through drag-and-drop UI, handling data routing, transformation, and system mediation.
* It supports real-time and batch processing, with flow-based programming and robust error handling.
* **Used in**: Data Engineering / Real-time ingestion  
  **Scope**: Flow-based data integration and routing  
  **Phase**: Data Collection, Preprocessing, Streaming

**8. Implement Real-Time dashboard (using Nifi, HBase, Kafka, Banana Solr Dashboard)**

* This stack enables real-time streaming (Kafka), ingestion (NiFi), storage (HBase), and visualization (Banana over Solr).
* Data scientists and engineers can monitor live data flows, detect anomalies, and power decision-making dashboards.
* **Used in**: Real-time Analytics / Monitoring  
  **Scope**: Streaming data pipelines and dashboards  
  **Phase**: Real-time Processing, Monitoring, Deployment

Would you like a visual summary table or diagram to map these tools to the Data Science Lifecycle (e.g., CRISP-DM or OSEMN framework)?

A screenshot of a data tool

AI-generated content may be incorrect.