Progress Report on App Development (Week 02)

-by Digvijay Jadhav

An Intern in UpSkill Campus, 16/01/2026

Digvijay Jadhav

I am pleased to present you with a comprehensive report on data science and machine learning (Week 02), which provides an overview of the process, challenges, and best practices for successful data science and machine learning. This report aims to make understanding of the key aspects of data science and machine learning and making informed decisions in this domain.

Data Science in the Big Data world refers to the process of extracting meaningful insights, patterns, and knowledge from **very large, fast, and complex datasets** that traditional data processing tools cannot handle efficiently.

What is Big Data?

Big Data is characterized by the **5 V's**:

1. **Volume** – Massive amounts of data (TBs, PBs)
2. **Velocity** – Data generated at high speed (real-time streams)
3. **Variety** – Structured, semi-structured, and unstructured data (text, images, videos)
4. **Veracity** – Data uncertainty and quality issues
5. **Value** – Useful insights derived from data

Role of Data Science in Big Data

Data Science applies **statistics, machine learning, and analytics** to Big Data to:

- Discover hidden patterns
- Predict future trends
- Support data-driven decision making
- Automate intelligent systems

Key Components

- **Data Collection**: From sensors, social media, logs, transactions
- **Data Storage**: Hadoop HDFS, NoSQL databases
- **Data Processing**: Apache Spark, MapReduce
- **Data Analysis**: Machine Learning, statistical models
- **Visualization**: Dashboards, graphs, reports

Tools & Technologies

- **Big Data Frameworks**: Hadoop, Spark
- **Programming Languages**: Python, R, Scala
- **Databases**: MongoDB, Cassandra, HBase
- **ML Libraries**: Scikit-learn, TensorFlow, PyTorch

Applications

- **Healthcare**: Disease prediction, medical imaging
- **Finance**: Fraud detection, risk analysis
- **E-commerce**: Recommendation systems
- **Agriculture**: Crop yield prediction
- **Smart Cities**: Traffic and energy optimization

Data Science Process ():

1. **Data Collection** – Gather data from various sources
2. **Data Cleaning** – Remove errors, missing values, and noise
3. **Data Exploration** – Understand data using statistics & visualization
4. **Feature Engineering** – Select and transform important features
5. **Model Building** – Apply machine learning algorithms
6. **Model Evaluation** – Check accuracy and performance
7. **Deployment & Monitoring** – Use the model and monitor results

Handling Large Data on a Single Computer ():

1. **Efficient Storage** – Use compressed formats (CSV → Parquet, HDF5)
2. **Chunk Processing** – Load data in parts instead of whole (batch-wise)
3. **Indexing** – Use indexes to speed up data access
4. **Sampling** – Analyze a representative subset of data
5. **Optimized Algorithms** – Use time- and memory-efficient methods
6. **Memory Management** – Use generators, avoid duplicates in memory
7. **Out-of-Core Processing** – Process data directly from disk (e.g., Pandas chunks)
8. **Hardware Utilization** – Use SSDs, multi-core CPUs, and RAM efficiently

Data Collection (Data Acquisition)

Explanation (short):
It involves gathering large volumes of data from different sources such as sensors, social media, logs, databases, transactions, and web applications.

Why it is important:

- Without data, no analysis is possible
- Quality and type of data affect all later steps
- It defines the scope of Big Data processing

Example sources:

- IoT devices
- Social media platforms
- Enterprise databases
- Web clicks and logs

n **NoSQL databases**, traditional SQL-style joins are **not common**. Instead, joins are handled using:

- **Embedding**
- **Application-side joins**
- **Aggregation frameworks** (most common)

Below is the **most standard NoSQL join example using MongoDB**.

1️⃣ Join in MongoDB using `$lookup`

MongoDB performs joins using the **aggregation pipeline**.

Example Scenario

- **users** collection
- **orders** collection

users collection

```
{ "_id": 1, "name": "Amit" }
{ "_id": 2, "name": "Ravi" }
```

orders collection

```
{ "_id": 101, "user_id": 1, "product": "Laptop" }
{ "_id": 102, "user_id": 1, "product": "Mobile" }
{ "_id": 103, "user_id": 2, "product": "Tablet" }
```

MongoDB Join Query (`$lookup`)

```
db.users.aggregate([
  {
    $lookup: {
      from: "orders",
      localField: "_id",
      foreignField: "user_id",
      as: "user_orders"
    }
  }
])
```

Output

```
{
  "_id": 1,
  "name": "Amit",
  "user_orders": [
    { "product": "Laptop" },
    { "product": "Mobile" }
  ]
}
```

2️⃣ Join using Embedding (Preferred in NoSQL)

```
{
  "_id": 1,
  "name": "Amit",
  "orders": [
    { "product": "Laptop" },
    { "product": "Mobile" }
  ]
}
```

✔ Faster reads
□ Data duplication

3️⃣ Application-Level Join (Pseudo Code)

```
user = db.users.find_one({"_id": 1})
orders = db.orders.find({"user_id": 1})
```

Exam-Oriented Summary

- NoSQL avoids joins for **performance & scalability**
- MongoDB supports joins using **$lookup**
- Embedding is preferred over joining
- Used in **Big Data & distributed systems**

Rise of Graph Databases

The **rise of graph databases** is driven by the need to efficiently manage and analyze **highly connected data** that traditional relational and NoSQL databases struggle to handle.

What is a Graph Database?

A graph database stores data as:

- **Nodes** → entities (people, products, places)
- **Edges** → relationships between entities
- **Properties** → attributes of nodes and edges

Reasons for the Rise of Graph Databases

1. **Explosion of Connected Data**
   - Social networks, recommendation systems, fraud networks
   - Relationships are as important as data itself
2. **Performance Limitations of RDBMS**
   - SQL joins become slow with deep relationships
   - Graph databases use **index-free adjacency**, making relationship traversal fast
3. **Growth of Big Data & NoSQL**
   - Semi-structured and unstructured data
   - Need flexible schema and fast relationship queries
4. **Real-Time Relationship Queries**
   - Shortest path, pattern matching, influence analysis
   - Graph DBs perform these operations efficiently
5. **Rise of AI & Machine Learning**
   - Knowledge graphs
   - Graph-based ML and recommendations

Key Advantages

- High performance for complex relationships
- Flexible schema

- Easy visualization of relationships
- Scales well for connected datasets

Popular Graph Databases

- **Neo4j**
- **Amazon Neptune**
- **ArangoDB**
- **JanusGraph**

Applications

- Social networking (friends, followers)
- Recommendation engines
- Fraud detection
- Network & IT operations
- Knowledge graphs (Google, AI systems)

Text Mining

Text Mining focuses on **discovering hidden patterns and knowledge** from large volumes of text.

Key Tasks:

- Text collection
- Text preprocessing (tokenization, stop-word removal)
- Pattern discovery
- Information extraction

Text Analytics

Text Analytics focuses on **analyzing and interpreting text data** to support decision-making.

Key Tasks:

- Sentiment analysis
- Text classification
- Topic modeling
- Trend analysis

Data Visualization to End User

Data Visualization is the process of presenting data in **graphical or visual formats** so that **end users can easily understand insights, patterns, and trends** without needing technical knowledge.

Purpose of Data Visualization for End Users

- Simplifies complex data
- Enables quick decision-making
- Improves understanding and engagement
- Highlights trends, comparisons, and outliers

Common Visualization Types

- **Bar Charts** – Comparison
- **Line Charts** – Trends over time
- **Pie Charts** – Proportions
- **Heat Maps** – Intensity and patterns
- **Dashboards** – Multiple insights in one view

Key Characteristics for End Users

- Simple and clear design
- Minimal technical jargon
- Interactive (filters, drill-downs)
- Real-time or near real-time updates
- Mobile and web-friendly

Tools Used

- Tableau
- Power BI
- Google Data Studio
- Excel Dashboards
- Web-based charts (D3.js, Chart.js)