# PROJECT REPORT
# ON
# From Traditional Business Intelligence to Big Data: Concepts, Architecture Design and Business Justification

Subject Name: Big Data Analytics
Subject Code: BAD601

Submitted By:
Prateek Kumar Bal
1AY23CD047

Submitted To:
Ms. Surbhi

Table of Contents: -

## TASK 1: Big Data in Daily Life – Visual Storytelling

### Amazon Case Study

Amazon is one of the world's largest e-commerce platforms, serving millions of customers daily. Every click, search, purchase, and review generates valuable data. Managing and analyzing such massive and diverse data cannot be handled by traditional Business Intelligence (BI) systems alone, making Big Data technologies essential.

### Classification of Data

In Amazon's ecosystem, data can be classified into:

**Structured Data**

Product IDs, prices, ratings, transaction records, and inventory data stored in relational databases.

**Semi-Structured Data**

JSON order logs, clickstream logs, and user activity records.

**Unstructured Data**

- Customer reviews, product images, search queries, and multimedia content.
- This combination of different formats increases the complexity of data management.
- Characteristics of Big Data (5V's)

**Volume**

Amazon handles millions of products and billions of transactions globally.

**Velocity**

Data is generated in real-time through browsing, purchases, and dynamic pricing updates.

**Variety**

Data exists in multiple formats: structured tables, text reviews, images, and logs.

**Veracity**

Issues such as fake reviews, incorrect ratings, and incomplete information affect data reliability.

**Value**

When analyzed effectively, data enables personalized recommendations, fraud detection, demand forecasting, and revenue optimization.

**Why Traditional BI Fails**

Traditional BI systems rely on centralized data warehouses and structured data formats. They face several limitations:

- Limited scalability when data grows exponentially
- Inability to process unstructured text efficiently
- Difficulty handling real-time streaming data
- High infrastructure costs
- Slow query performance with very large datasets

These constraints make traditional BI insufficient for modern e-commerce operations.

**Why Big Data is Required**

Big Data technologies such as Hadoop, NoSQL databases, and Spark provide distributed storage and parallel processing capabilities. They allow:

- Scalable storage across clusters
- Real-time analytics
- Text mining and sentiment analysis
- Machine learning-based recommendations
- Efficient fraud detection

Thus, Big Data enables Amazon to transform raw data into actionable insights and competitive advantage.

**Dataset Used –**

Amazon Product Reviews Dataset

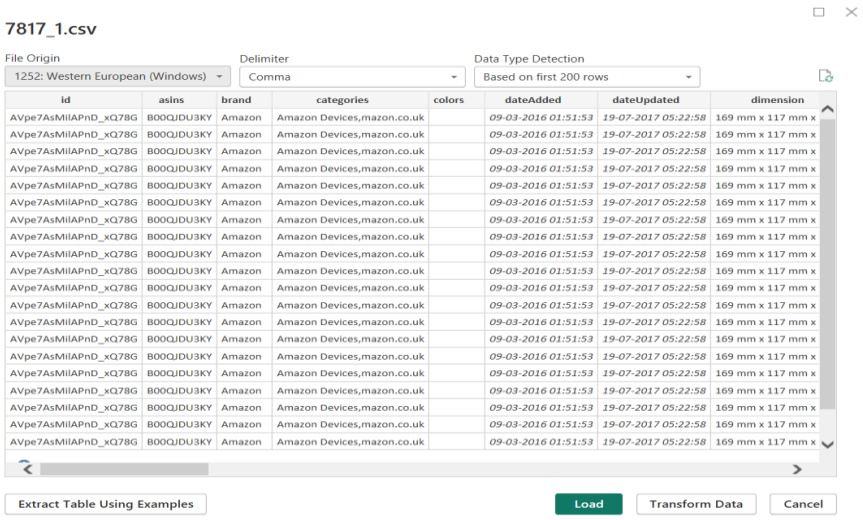


Fig. Dataset - Amazon Product Reviews Dataset

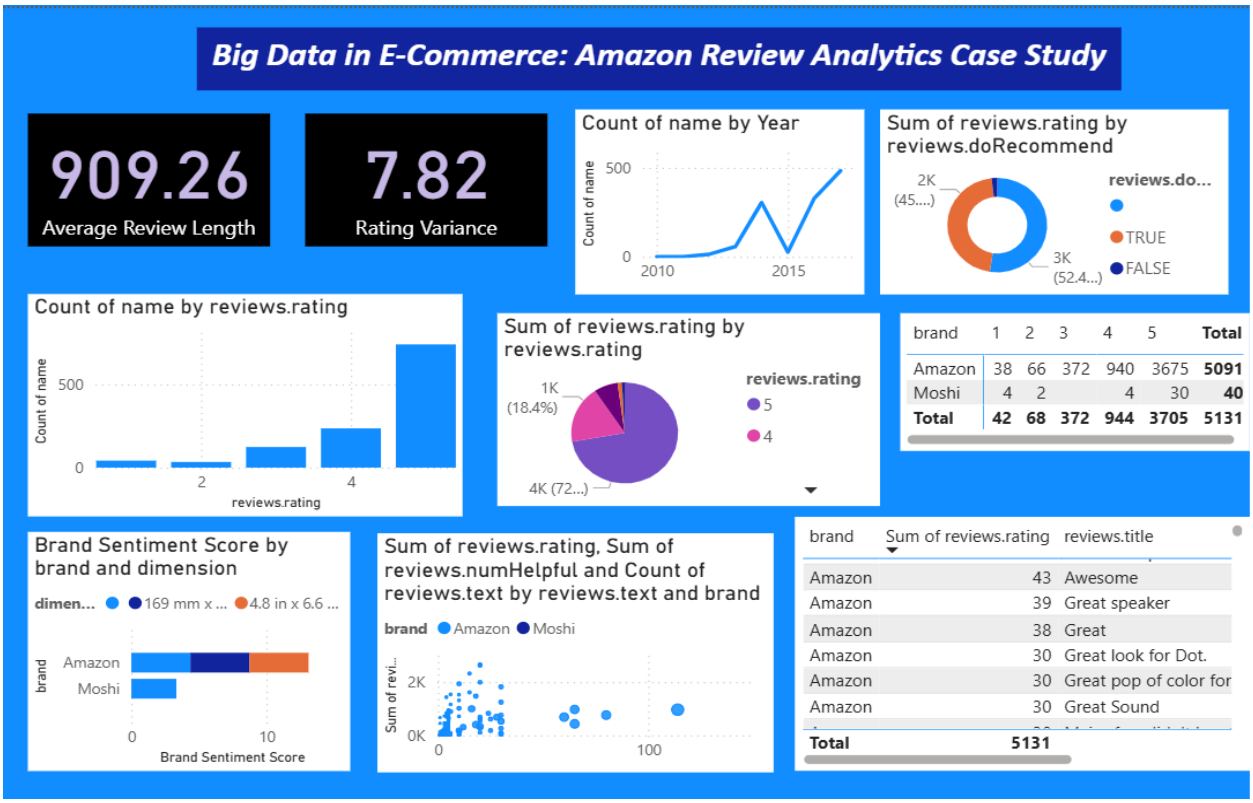


Fig. Power BI report




Acharya Dr. Sarvepalli Radhakrishnan Road, Soladevanahalli, ACHIT Nagar P. O., Bangalore-560 107
https://ait.ac.in Ph.: 080 22555555

## TASK 2: BI vs Big Data – Role Play

**Roleplay Scenario**

Manager:

I don't understand this Big Data hype. We already use Excel and SQL. That's enough for our e-commerce reports.

Consultant:

Excel and SQL are good for structured data and historical reports. But our Amazon-scale data includes millions of reviews, real-time clicks, and unstructured text. Traditional BI tools struggle with that scale.

**********

Manager:

But we already have a data warehouse. Why complicate things?

Consultant:

Data warehouses are optimized for structured tables. But customer reviews, browsing logs, and recommendation data are semi-structured or unstructured. They don't fit neatly into relational tables.

**********

Manager:

We can just increase server capacity if data grows.

Consultant:

That's vertical scaling — expensive and limited. Big Data uses horizontal scaling, meaning distributed clusters like Hadoop that can scale across multiple machines cost-effectively.

**********

Manager:

But do we really need that much computing power?

Consultant:

Consider this: Every second, customers browse products, leave reviews, and compare prices. That's high-velocity data. Traditional BI works in batch mode, not real-time processing.

**********

Manager:

We don't need real-time insights for everything.

Consultant:

For recommendations, we do. When a customer clicks a product, the system instantly suggests similar items. That requires fast distributed processing using Spark or similar frameworks.

**********

Manager:

Can't SQL handle large joins and aggregations?

Consultant:

It can — up to a limit. But when dealing with terabytes or petabytes of data, query performance degrades significantly. Hadoop distributes processing across nodes.

**********

Manager:

What about analyzing reviews? We can store them as text fields.

Consultant:

Storing is not the issue. Analyzing thousands of text reviews for sentiment requires natural language processing, which traditional BI systems aren't built for.

**********

Manager:

So you're saying Big Data is mainly about size?

Consultant:

Not just size. It's about the 5V's — Volume, Velocity, Variety, Veracity, and Value. Traditional BI handles volume poorly and struggles with variety and velocity.

**********

Manager:

How does Hadoop help exactly?

Consultant:

Hadoop provides distributed storage (HDFS) and parallel processing using MapReduce. It allows massive datasets to be processed across clusters efficiently.

**********

Manager:

And NoSQL? Why not just relational databases?

Consultant:

NoSQL databases allow flexible schemas. E-commerce data changes frequently — new attributes, new product fields. Relational databases require schema redesign; NoSQL adapts easily.

**********

Manager:

What's the actual business benefit?

Consultant:

Personalized recommendations, fraud detection, demand forecasting, and dynamic pricing. These directly increase revenue and customer retention.

**********

Manager:

But implementing Big Data sounds costly.

Consultant:

Initially, yes. But distributed systems use commodity hardware, reducing long-term infrastructure costs compared to high-end centralized servers.

**********



Acharya Dr. Sarvepalli Radhakrishnan Road, Soladevanahalli, ACHIT Nagar P. O., Bangalore-560 107
https://ait.ac.in Ph.: 080 22555555

## TASK 3: Architecture Design Challenge

### ARCHITECTURE 1: Traditional Data Warehouse Architecture

1. Data Sources

In a traditional BI setup, data comes from structured internal systems:

- Transaction database (Orders, Payments)
- Product catalog database
- Customer database
- Inventory system
- CRM system

These systems typically use relational databases (RDBMS) such as MySQL or Oracle.

2. Data Integration Layer (ETL)

ETL Process:

- Extract data from operational databases
- Transform data (cleaning, aggregation, schema mapping)
- Load into central data warehouse

Tools commonly used:

- Informatica
- Talend
- SQL Server Integration Services (SSIS)

Limitations:

- Batch-based (daily/weekly)
- Slow for large-scale data
- Not suitable for streaming data

3. Storage Layer – Data Warehouse

Centralized storage:

- Structured tables
- Star or Snowflake schema
- Fact tables (Sales, Orders)
- Dimension tables (Customer, Product, Time)

Characteristics:

- Optimized for structured queries
- Schema-on-write
- Limited scalability (vertical scaling)

4. Processing Layer

- SQL-based querying
- OLAP cubes
- Aggregations and joins
- Historical reporting

Problems:

- Slow with very large datasets
- Cannot efficiently process text reviews
- No real-time capability

5. Analytics & Reporting Layer

- Power BI
- Tableau
- Excel dashboards

Used for:

- Sales reports
- Revenue analysis
- Monthly performance tracking

Why This Architecture Fails for Amazon Scale

- Cannot handle unstructured review text efficiently
- No real-time recommendation capability
- Expensive scaling
- Performance degradation with growing data
- Limited support for machine learning

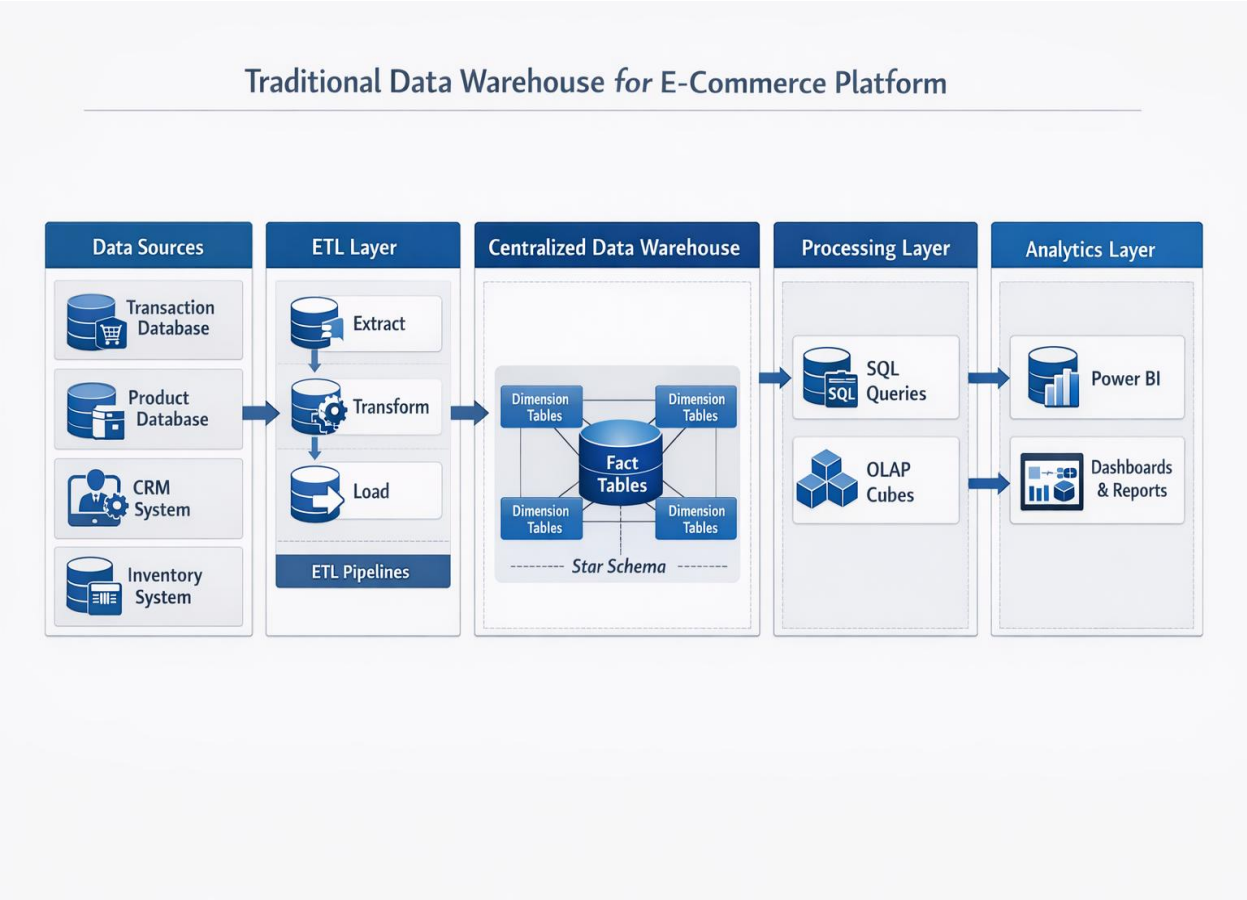


Fig. Traditional Data Warehouse Architecture

**ARCHITECTURE 2: Hadoop-Based Big Data Architecture**

1. Data Sources

Structured:

- Transaction records
- Payment data
- Inventory data

Semi-Structured:

- JSON clickstream logs
- User session logs

Unstructured:

- Customer reviews
- Images
- Search queries

Streaming Sources:

- Real-time browsing activity
- Live purchase events

2. Data Ingestion Layer

Tools:

- Apache Kafka (real-time streaming)
- Apache Flume (log ingestion)
- Sqoop (RDBMS import)

Purpose:

- Collect batch + streaming data
- Push data into distributed storage

3. Storage Layer – Distributed Storage

Primary Storage:

- Hadoop Distributed File System (HDFS)

Database Layer:

- NoSQL (MongoDB / Cassandra)
- HBase

Characteristics:

- Horizontal scaling
- Fault tolerance
- Schema-on-read
- Handles structured + unstructured data

4. Processing Layer

Batch Processing:

- MapReduce

Fast Distributed Processing:

- Apache Spark

Real-Time Processing:

- Spark Streaming

Capabilities:

- Sentiment analysis on reviews
- Recommendation systems
- Fraud detection
- Demand forecasting

5. Analytics & Serving Layer

- Machine Learning models
- Real-time recommendation engine

- API layer
- Power BI dashboards
- Business reporting

6. Security & Governance Layer

- Data encryption
- Role-based access control
- Data quality checks
- Metadata management


Fig. HADOOP-BASED BIG DATA ARCHITECTURE

## TASK 4: Analytics & Tool Match

| Business Question | Analytics Type | Tool | Explanation |
|---|---|---|---|
| What happened? | Descriptive Analytics | Power BI, SQL | Used to summarize past data such as total reviews, average ratings, sales trends, and brand performance. |
| Why did it happen? | Diagnostic Analytics | Spark, Hadoop, SQL | Used to analyze rating variance, sentiment patterns in reviews, and identify causes behind low ratings or product returns. |
| What will happen next? | Predictive Analytics | Apache Spark MLlib, Python (Machine Learning), Hadoop | Used to forecast demand, predict customer churn, and anticipate product rating trends. |
| What action should be taken? | Prescriptive Analytics | Spark + Recommendation Engine + NoSQL | Suggests personalized product recommendations, dynamic pricing strategies, and inventory optimization. |

**What Happened? – Descriptive Analytics**

Descriptive analytics focuses on summarizing historical data. In Amazon's case, tools like Power BI and SQL help visualize total reviews, average ratings, recommendation percentages, and brand performance. This stage answers basic questions about past performance using dashboards and reports.

Tools Used:

- Power BI
- SQL

**Why Did It Happen? – Diagnostic Analytics**

Diagnostic analytics investigates patterns and root causes. Using distributed processing tools such as Apache Spark and Hadoop, Amazon can analyze large-scale review data, detect sentiment patterns, identify low-performing brands, and examine rating variance. This requires processing unstructured text data, which traditional BI struggles with.

Tools Used:

- Apache Spark
- Hadoop (HDFS storage)

**What Will Happen Next? – Predictive Analytics**

Predictive analytics uses machine learning models to forecast future outcomes. Tools like Spark MLlib enable demand forecasting, customer behavior prediction, and product rating trend analysis. This helps Amazon anticipate sales surges and customer preferences.

Tools Used:

- Spark MLlib
- Machine Learning models

**What Action Should Be Taken? – Prescriptive Analytics**

Prescriptive analytics recommends optimal actions based on predictions. Using recommendation engines, NoSQL databases, and distributed processing systems, Amazon suggests personalized products, adjusts pricing dynamically, and optimizes inventory levels. This stage directly impacts revenue and customer satisfaction.

Tools Used:

- Spark
- Real-time processing engines

## Bonus Challenge – Explaining Big Data to a Kid

Imagine you have a huge online toy shop.

Every day:

- Millions of kids visit
- They search for toys
- They buy things
- They write reviews

That creates a massive amount of information.

A normal computer is like a small notebook — it can handle only limited data.

But Big Data is like a team of super computers working together to:

- Store huge amounts of information
- Understand reviews
- Predict which toy will be popular
- Suggest toys you might like

So, in simple words:

Big Data means using many powerful computers together to understand huge amounts of information and make smart decisions.

## 1. About the Project

This project analyzes Amazon product review data to demonstrate the transition from Traditional Business Intelligence (BI) systems to Big Data architectures.

The objective of the project is to:

- Understand how e-commerce platforms generate large-scale data
- Classify structured and unstructured data
- Demonstrate the limitations of traditional BI systems
- Design both Traditional Data Warehouse and Hadoop-based architectures
- Perform multi-level analytics (Descriptive, Diagnostic, Predictive, Prescriptive)
- Implement data analysis using Power BI and advanced DAX measures

The project highlights how Amazon-scale operations require distributed storage, parallel processing, and scalable analytics frameworks.

## 2. About the Tools and Technologies Used

**Power BI**

Used for:

- Data visualization
- Dashboard creation
- KPI generation
- Analytical modeling using DAX

Why used:
Power BI enables structured reporting and business insight visualization.

**DAX (Data Analysis Expressions)**

Used to:

- Create calculated measures
- Perform advanced aggregations
- Compute engagement ratios
- Perform time-based analysis

**Big Data Technologies**

**1. Hadoop (HDFS)**

- Distributed storage
- Fault tolerance
- Horizontal scalability

**2. Apache Spark**

- Fast in-memory processing
- Machine learning support
- Real-time analytics capability

**3. NoSQL Databases**

- Flexible schema
- Handles semi-structured data
- High scalability

## 3. Detailed Description of My Contribution

### 3.1 What is Done and How it is Done

**Step 1: Data Collection**

Amazon review dataset was imported into Power BI.

Dataset includes:

- reviews.rating
- reviews.text
- reviews.date
- reviews.doRecommend
- reviews.numHelpful
- brand
- categories
- prices

**Step 2: Data Cleaning & Preparation**

- Removed null values
- Converted review dates to proper date format
- Ensured rating column is numeric
- Standardized brand names

**Step 3: Data Modeling**

- Created calculated columns
- Created advanced DAX measures
- Built KPI cards and analytical visuals

**Step 4: Analytics Performed**

✓ Descriptive Analytics
✓ Diagnostic Analytics
✓ Engagement Analysis
✓ Sentiment approximation
✓ Trend analysis

### 3.2 Code Explanation (DAX Measures)

Below are the main DAX formulas implemented in Power BI.

**Recommendation Rate %**

```
Recommendation Rate % =
VAR RecommendCount =
   CALCULATE(
      COUNTROWS('Amazon'),
      'Amazon'[reviews.doRecommend] = TRUE()
   )
VAR TotalReviews =
   COUNTROWS('Amazon')

RETURN
DIVIDE(RecommendCount, TotalReviews, 0)
```

Explanation:
Calculates percentage of users who recommend the product.

**Helpful Engagement Ratio**

```
Helpful Engagement Ratio =
VAR TotalHelpful =
   SUM('Amazon'[reviews.numHelpful])
VAR TotalReviews =
   COUNTROWS('Amazon')

RETURN
DIVIDE(TotalHelpful, TotalReviews, 0)
```

Explanation:
Measures average helpful votes per review.

**Review Length (Calculated Column)**

```
Review Length = LEN('Amazon'[reviews.text])
```

Explanation:
Calculates character length of each review to represent unstructured data volume.

**Average Review Length**

Average Review Length =
AVERAGE('Amazon'[Review Length])

Explanation:
Measures overall text data size intensity.

**Brand Sentiment Score**

```
Brand Sentiment Score =
CALCULATE(
    AVERAGE('Amazon'[reviews.rating]),
    ALLEXCEPT('Amazon', 'Amazon'[brand])
)
```

Explanation:
Computes average rating per brand, ignoring other filters.

**Rating Variance**

```
Rating Variance =
VAR AvgRating =
    AVERAGE('Amazon'[reviews.rating])
RETURN
    DIVIDE(
        SUMX(
            'Amazon',
            POWER('Amazon'[reviews.rating] - AvgRating, 2)
        ),
        COUNT('Amazon'[reviews.rating])
    )
```

Explanation:
Measures consistency of product ratings.

**Monthly Review Growth %**

```
Monthly Review Growth % =
VAR CurrentMonthReviews =
   COUNT('Amazon'[reviews.text])

VAR PreviousMonthReviews =
   CALCULATE(
      COUNT('Amazon'[reviews.text]),
      DATEADD('Amazon'[reviews.date], -1, MONTH)
   )

RETURN
DIVIDE(CurrentMonthReviews - PreviousMonthReviews, PreviousMonthReviews, 0)
```

Explanation:
Measures review growth rate to demonstrate data velocity.

## 4. Implementation Code

Implementation involved:

1. Importing CSV into Power BI
2. Data transformation using Power Query
3. Creating calculated columns
4. Writing advanced DAX measures
5. Designing visualizations:
    - Donut chart (Recommendation %)
    - Pie chart (Rating distribution)
    - Scatter plot (Helpful votes vs Rating)
    - Matrix (Brand vs Average Rating)

- Table (Detailed review insights)
- Line chart (Review growth over time)

**5. Results**

The analysis revealed:

- Majority of ratings are 4–5 stars
- Certain brands show high rating variance
- Recommendation percentage is strongly correlated with rating
- Review growth demonstrates continuous data generation (Velocity)
- Long review texts indicate presence of unstructured data (Variety)

These results validate the presence of Big Data characteristics.

## 6. Conclusion

This project demonstrates the evolution from Traditional BI systems to Big Data architectures in the context of Amazon's e-commerce platform.

While traditional BI tools efficiently handle structured historical reporting, they are insufficient for processing:

- Large-scale unstructured review text
- Real-time customer interaction data
- Machine learning-driven predictions

Big Data technologies such as Hadoop, Spark, and NoSQL enable distributed storage, parallel processing, and scalable analytics.

Thus, Big Data is essential for modern e-commerce systems to achieve personalization, scalability, and data-driven decision-making.