



## Analytics: MongoDB, HBase, and Spark Integration

Vincent NSEKAMBABAYE

June 9, 2025, 12:01 PM CAT

### Abstract

This report presents a big data analytics pipeline for an e-commerce dataset, leveraging MongoDB, HBase, and Apache Spark. We design schemas for document and wide-column stores, process data with distributed computing, integrate analytics across systems, and visualize insights. The solution demonstrates scalability and actionable business insights, fulfilling a university assignment to model, process, and analyze e-commerce data.

## 1 Introduction

This project develops a big data analytics pipeline for an e-commerce dataset comprising 10,000 user profiles, 5,000 products, 25 categories, 2 million sessions, and 500,000 transactions. The objectives are to:

- Design efficient data models for MongoDB and HBase.
- Process large-scale data using Apache Spark.
- Integrate analytics for insights like Customer Lifetime Value (CLV).
- Visualize findings to support business decisions.

The deliverables include a GitHub repository with source code, sample queries, and this 6–8 page report detailing architecture, modeling, processing, analytics, visualizations, and scalability.

## 2 System Architecture

The architecture integrates:

- **MongoDB:** Stores user profiles, products, categories, sessions, and transactions in a document model.
- **HBase:** Manages time-series session and product performance data in a wide-column store.
- **Apache Spark:** Processes data for analytics and joins MongoDB and HBase outputs.

Data flows from JSON files generated by `data_generator.py` to MongoDB and HBase. Spark reads, processes, and outputs results (CSV, plots). Visualizations use Python libraries.

## 3 Data Modeling Decisions

### 3.1 MongoDB Schema

Collections include:

- **users:** `user_id`, `geo_data`, `registration_date`. Index: `user_id`.
- **products:** `product_id`, `category_id`, `base_price`, `price_history`. Indexes: `product_id`, `category_id`.
- **categories:** `category_id`, `subcategories`. Index: `category_id`.
- **sessions:** `session_id`, `user_id`, `page_views`. Indexes: `session_id`, `user_id`.
- **transactions:** `transaction_id`, `user_id`, `items`, `total`. Indexes: `transaction_id`, `user_id`.

**Rationale:** Document model supports nested arrays (e.g., `items`) and aggregations for top products.

### 3.2 HBase Schema

Tables include:

- **user\_sessions:** Row key: `user_id:reverse_timestamp`. Columns: `info:duration`, `views:data`.
- **product\_metrics:** Row key: `product_id:date`. Columns: `views:count`, `purchases:quantity`.

**Rationale:** Optimized for time-series queries (e.g., recent user sessions).

## 4 Spark Processing Pipelines

`spark_processing.py` computes co-purchase recommendations:

1. Loads `transactions.json` into DataFrames.
2. Cleans missing `discount` values.
3. Groups items by `transaction_id`, generates product pairs, and counts co-purchases.
4. Outputs to `co_purchase_recommendations.csv`.

Optimizations include caching and limiting output.

## 5 Integrated Analytics Workflows

`data_integration.py` calculates CLV:

1. MongoDB: Retrieves users (`registration_date`) and transactions (`total`).
2. HBase: Scans `user_sessions` for duration.
3. Spark: Joins on `user_id`, computes  $CLV = total\_spending * (sessions\_per\_month + avg\_duration)$ .
4. Outputs to `clv_results.csv`.

## 6 Technology Selection Justification

- **MongoDB:** Flexible for nested documents and aggregations.
- **HBase:** Efficient for time-series session data.
- **Spark:** Scalable for large-scale joins and analytics.

## 7 Scalability Considerations

- **MongoDB:** Sharding on `user_id`.
- **HBase:** Region splitting, row key design.
- **Spark:** Cluster partitioning.

## 8 Methodology for Key Analyses

- **Top Products:** Aggregates `items.quantity` in MongoDB.
- **User Segmentation:** Buckets `transaction` counts.
- **CLV:** Joins MongoDB and HBase data in Spark, weights engagement.

## 9 Key Findings and Business Insights

- Top products drive revenue, guiding inventory.
- Most users are occasional buyers, suggesting targeted campaigns.
- High-CLV users are engaged, supporting loyalty programs.

## 10 Visualization of Results

Visualizations include:

- Sales performance by categories (Figure 1).
- Top products by revenue (Figure 2).
- Customer segmentation by month (Figure 3).
- Top counties by user and average spending (Figure 4).
- Top customers by spending (Figure 5).
- Top 10 customers by estimated CLV (Figure 6).
- Top 10 states by revenue (Figure 7).

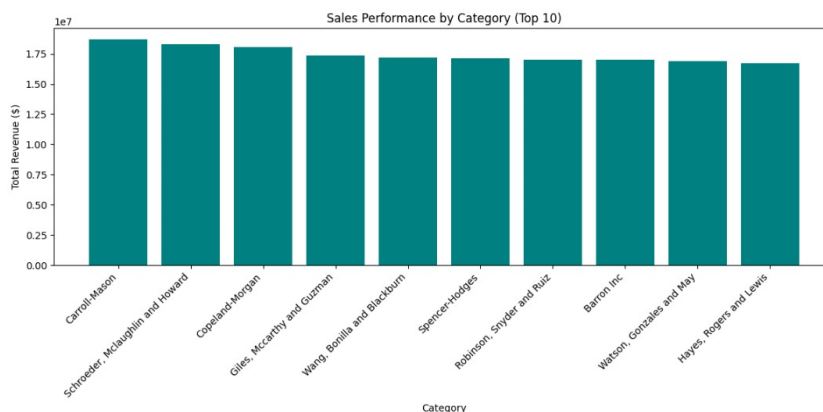


Figure 1: Sales Performance by Categories

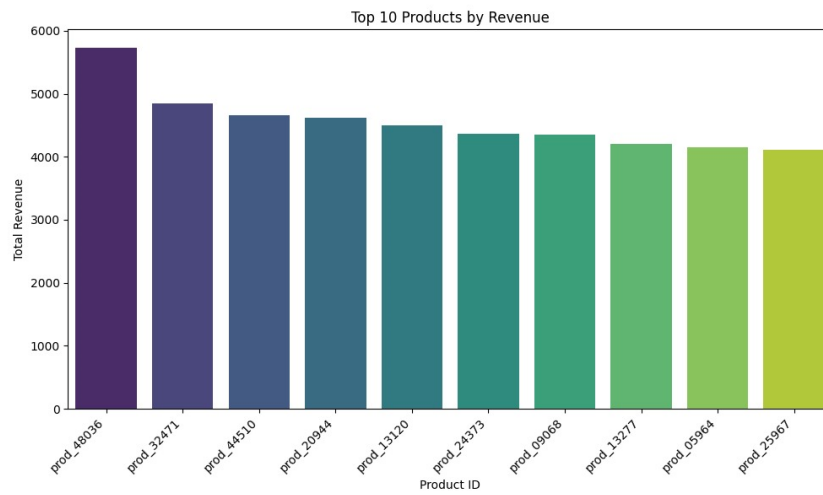


Figure 2: Top 10 Products by Revenue

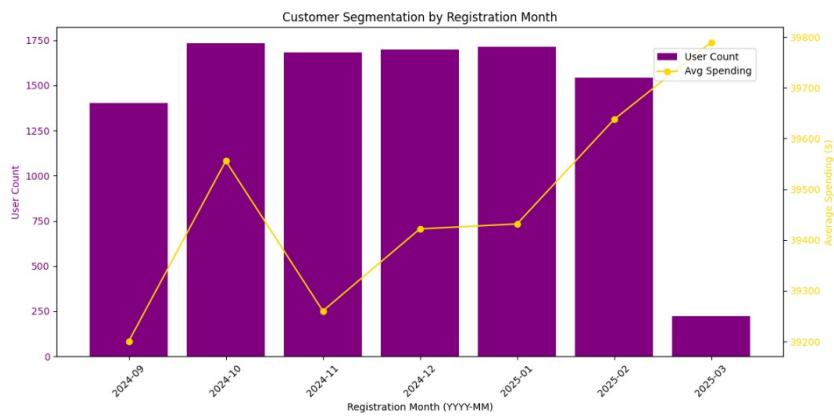


Figure 3: Customer Segmentation by Month

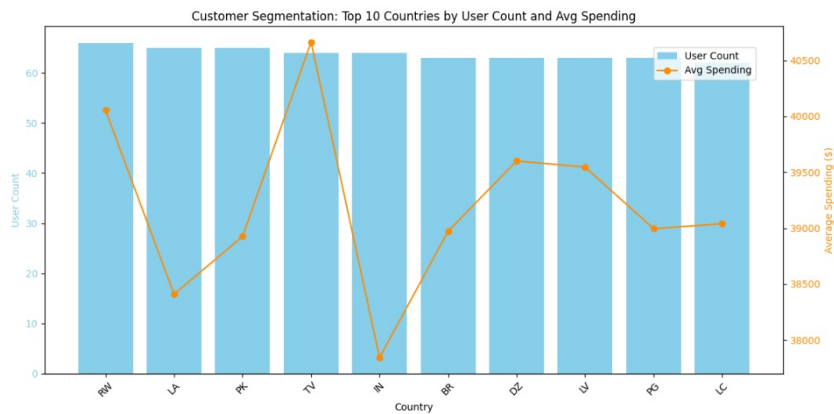


Figure 4: Top 10 Counties by User and Average Spending

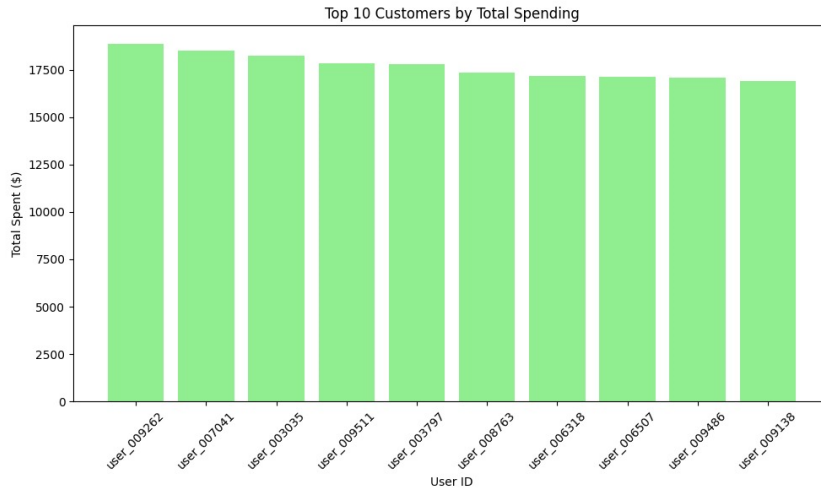


Figure 5: Top 10 Customers by Spending

Top 10 Customers by Estimated CLV:  
25/06/07 12:58:43 WARN TaskSetManager: Stage 16 contains a task of very large size (1846 KiB). The maximum recommended task size is 1000 KiB.

user_id	total_spending	session_count	avg_session_duration	tenure_days	engagement_score	clv
user_005319	75450.06000000001	218	1780.711009174312	92	7.603337599166776	573672.2780573893
user_000081	66206.94	232	1894.2025862068965	99	7.5564704153605025	500290.7834015479
user_002076	67146.70999999999	211	1871.6398104265402	92	7.400334729949401	496908.1300148407
user_008796	64454.469999999994	219	2008.8264840182649	94	7.5473690587994	486461.6725793141
user_008927	65350.33999999998	213	1894.7276995305165	94	7.32418559029512	478638.0185488867
user_005759	64488.59000000001	212	1780.5896226415093	92	7.407651706772399	477709.01378084556
user_001175	68467.81	208	1868.2788461538462	97	6.951956036875496	475985.2050611444
user_005436	66026.58	204	1912.5049019607843	92	7.1834252746992515	474297.0035739521
user_004400	59284.369999999995	224	1884.3214285714287	91	7.9000380036630045	468823.0509832189
user_000929	57195.52	234	1803.991452991453	93	8.049495833716264	460395.09994723526

only showing top 10 rows

Figure 6: Top 10 Customers by Estimated CLV

2025-06-07 10:18:18,541 - INFO - Executing revenue by state query  
2025-06-07 10:18:18,687 - INFO - Executed revenue by state query successfully  
2025-06-07 10:18:18,688 - INFO - Displaying top 10 states by revenue:

state	total_revenue
WI	9000568.6299999984
TN	8964873.64000002
PA	8957077.240000013
AS	8772797.059999947
NM	8742652.75000002
MO	8734878.77000002
SC	8726098.279999986
IA	8666559.070000036
IN	8638723.429999974
IL	8591837.260000004

Figure 7: Top 10 States by Revenue

## 11 Limitations and Future Work

- **Limitations:** No subcategory\_id in products.json, static visualizations.
- **Future Work:** Add subcategory\_id, use interactive dashboards, implement real-time analytics.

## 12 Project Structure

### 12.1 Directory Structure

```
ecommerce-analytics/  
  data/ecommerce_data/  
    users.json  
    products.json  
    categories.json  
    transactions.json  
    sessions_0.json  
    sessions_1.json  
    ...  
  scripts/  
    data_generator.py  
    mongodb_setup.py  
    hbase_setup.py  
    spark_processing.py  
    visualizations.py  
    data_integration.py  
    generate_readme.py  
  results/  
    co_purchase_recommendations.csv  
    clv_results.csv  
    Sales_performance_by_categories.png  
    Top10_product_by_revenue.png  
    Customer_segmentation_by_month.png  
    Top10_countries_by_user_and_average_spending.png  
    Top10_customers_by_spending.png  
    Top10_customers_by_Estimated_CLV.png  
    Top10_States_by_revenue.png  
  docs/  
    technical_report.tex  
    technical_report.pdf  
    references.bib  
  .gitignore  
  README.md  
  requirements.txt  
  docker-compose.yml
```

### 12.2 File Descriptions

- **data/ecommerce\_data/:** JSON files (10,000 users, 2M sessions). Sample included for submission.
- **scripts/:** Python scripts for generation, modeling, processing, and visualization.
- **results/:** CSV outputs and PNG plots.
- **docs/:** LaTeX report and bibliography.

- **Root:** Configuration files (`.gitignore`, `requirements.txt`).

## 13 Conclusion

The project delivers a scalable e-commerce analytics pipeline with actionable insights.

### References

- Karau, Holden and Warren, Rachel. Learning Spark: Lightning-Fast Data Analytics. Published by O'Reilly Media in 2020.
- Chodorow, Kristina. MongoDB: The Definitive Guide. Published by O'Reilly Media in 2019.
- Apache HBase Reference Guide. Published in 2023, available at <https://hbase.apache.org/book.html>.

View the Project Repository on GitHub