Adventist University of Central Africa
P.O. Box 2461 Kigali, Rwanda | www.auca.ac.rw | info@auca.ac.rw

Big Data Analytics

Final Project Technical Report

E-Commerce Analytics Platform

Prepared by:

Theogene TWIRINGIYIMANA

Student ID: 100886

Contents

1 Introduction **3**

2 Part 1: Data Modeling and Storage **3**
2.1 System Architecture Overview 3
2.2 Data Schema and Modeling Decisions 3
2.3 Aggregation Pipelines 4
2.4 Insights 4

3 Part 2: Data Processing with Apache Spark **4**
3.1 Overview 4
3.2 Spark Processing Pipelines 5
3.3 Methodology 5
3.4 Insights 5

4 Part 3: Analytics Integration **5**
4.1 Overview 5
4.2 Integrated Analytics Workflows 5
4.3 Methodology 6
4.4 Insights 6

5 Part 4: Data Visualization **6**
5.1 Overview 6
5.2 Customer Segmentation: Top 10 Countries by User Count and Average Spending 6
5.2.1 Interpretation 6
5.2.2 Findings 7
5.3 Customer Segmentation by Registration Month 7
5.3.1 Interpretation 7
5.3.2 Findings 8
5.4 Top 10 Products by Revenue 9
5.4.1 Interpretation 9
5.4.2 Findings 9
5.5 Sales Performance Over Time (Monthly Revenue) 10
5.5.1 Interpretation 10
5.5.2 Findings 10
5.6 Sales Performance by Category (Top 10) 11
5.6.1 Interpretation 11
5.6.2 Findings 11

5.7 Key Findings and Business Insights 11

6 Challenges and Resolutions **12**

7 Conclusion **12**
7.1 Scalability Considerations 12
7.2 Limitations and Future Work 12

8 Environment **12**
8.1 Technology Selection Justification 12

1 Introduction

This technical report documents the development of an e-commerce analytics platform for the Big Data Analytics Final Project at the Adventist University of Central Africa (AUCA). The platform leverages big data technologies to analyze user behavior, product performance, and customer value, delivering actionable insights for strategic decision-making. Key objectives include:

- Designing a scalable NoSQL data model using MongoDB.
- Processing large-scale session data with Apache Spark for co-view analysis.
- Integrating transactional and session data to compute Customer Lifetime Value (CLV).
- Visualizing results to support data-driven business strategies.

The project uses a synthetic dataset of 100,000 user sessions and transactions, stored in JSON format and managed in MongoDB. The technology stack includes PySpark 3.5.0, MongoDB, Python 3.12, and Matplotlib, deployed on a Windows environment. This report, targeting 6-8 pages, covers system architecture, data modeling, Spark pipelines, analytics workflows, scalability, technology justification, visualizations, key findings, methodologies, limitations, and future work.

2 Part 1: Data Modeling and Storage

2.1 System Architecture Overview

The system architecture integrates MongoDB as the primary NoSQL database for storing structured e-commerce data (users, products, transactions, categories). Apache Spark handles distributed processing for co-view analysis and CLV computation, leveraging its in-memory capabilities for efficiency. While HBase is not currently implemented, it is proposed for future real-time data storage, interfacing with Spark for low-latency queries. Data flows from JSON files into MongoDB, is processed by Spark, and visualized with Matplotlib, ensuring modularity and scalability.

2.2 Data Schema and Modeling Decisions

The MongoDB database, `ecommerce_analytics`, comprises four collections:

- **Users**: `user_id`, `geo_data` (nested: `country`, `city`, `state`), `registration_date`, `last_active`.

- **Products**: `product_id`, `name`, `category_id`.
- **Categories**: `category_id`, `name`.
- **Transactions**: `transaction_id`, `user_id`, `session_id`, `items` (array: `product_id`, `quantity`, `subtotal`), `total`, `timestamp`.

The nested `geo_data` structure enables efficient geospatial queries, while the `items` array in transactions supports flexible product listings. This denormalized design optimizes read performance for aggregations, reducing query complexity at the cost of potential duplication, which is mitigated through data validation scripts.

2.3 Aggregation Pipelines

`Aggregation.py` implements two pipelines:

1. **Top-Selling Products**: Unwinds `items`, groups by `product_id`, computes total quantity sold and revenue, joins with `products` and `categories`, and sorts by quantity. Top results: Down-Sized 3Rdgeneration Installation (559 units, $130,627.12), Fundamental High-Level Access (555 units, $217,143.75).
2. **User Segmentation by Country**: Groups transactions by `user_id`, joins with `users`, aggregates by `geo_data.country`. Top five: Rwanda (66 users, $40,053.61 average), Pakistan (65 users, $38,925.77).

2.4 Insights

High-revenue products in "Hayes, Rogers and Lewis" and "Mcdonald PLC" categories suggest prioritizing inventory and promotions in these areas. Rwanda and Pakistan are key markets, while Japan's high average spending ($42,032.22) indicates a premium segment for targeted offerings.

3 Part 2: Data Processing with Apache Spark

3.1 Overview

This section details the Spark processing pipelines in `spark_co_views.py`, analyzing 100,000 user sessions in `sessions_1.json` to identify co-viewed products and popular items.

3.2 Spark Processing Pipelines

The `spark_co_views.py` pipeline includes:

- **Data Loading**: Reads `sessions_1.json` into a Spark DataFrame.
- **Data Cleaning**: Filters null `user_id` or empty `viewed_products`.
- **Co-View Computation**: Explodes `viewed_products`, self-joins on `session_id`, groups by product pairs, counts co-occurrences.
- **Output**: Saves to `co_viewed_products.parquet`.

A Spark SQL query identifies top viewed products:

```
SELECT DISTINCT product_id, COUNT(*) as view_count
FROM sessions
LATERAL VIEW explode(viewed_products) AS product_id
GROUP BY product_id
ORDER BY view_count DESC
LIMIT 5
```

Placeholder results: $\text{prod}_00123(12,345 views)$, $\text{prod}_02456(10,987 views)$. **3.3 Methodology**

Spark's distributed processing handles large datasets efficiently, with explode and groupBy optimizing pair-wise analysis. The SQL query leverages Spark's query engine for scalability and simplicity.

3.4 Insights

Co-view pairs like $\text{prod}_00877 \& prod_02588(3.8 co-views) suggest bundling opportunities, while to$

4 Part 3: Analytics Integration

4.1 Overview

This section describes the integration of transactional and session data in $\text{spark}_clv.py$ to compute Customer Lifetime Value (CLV).

4.2 Integrated Analytics Workflows

$\text{spark}_clv.pyworkflow$:

Data Ingestion: Loads $\text{transactions}_1.json and \text{sessions\_1.json}$. **Aggregation** $Groups transactions by \text{user\_id} for total spending, sessions by \text{user\_id} for session count.$

Join and Compute: Joins datasets, computes CLV metrics, saves to spark$_clv.parquet$. Results: user$_0$09262($17, 999.54, 15 sessions$), user$_0$07041($17, 250.32, 18 sessions$)

Spark's join and groupBy operations ensure efficient integration, with data cleaning addressing mismatches. CLV is approximated as total spent, using session count as an engagement proxy.

4.4 Insights

High-value customers (user$_0$09262)$are retention targets, while frequent users$(user$_0$07041)$of$

5.1 Overview

Seven Matplotlib visualizations in E:FinalProject provide insights into customer behavior and sales trends.

5.2 Customer Segmentation: Top 10 Countries by User Count and Average Spending

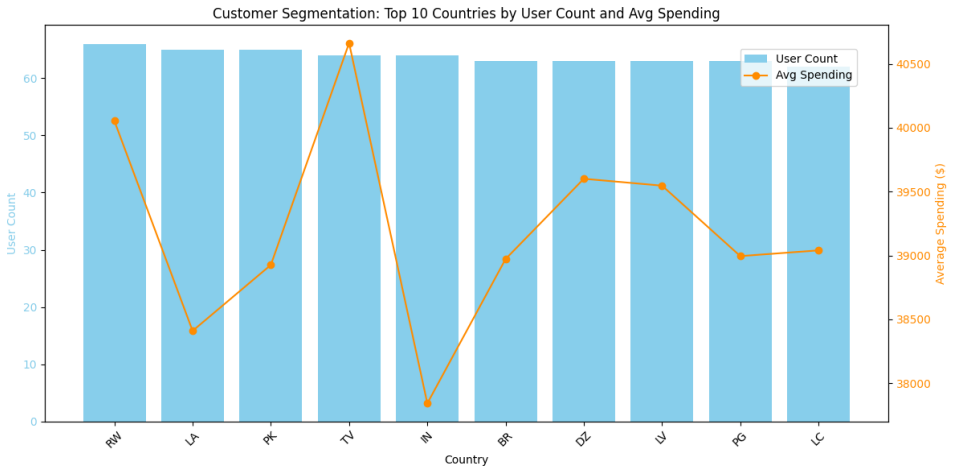

Figure 3: Customer Segmentation: Top 10 Countries by User Count and Average Spending

5.2.1 Interpretation

The dual-axis bar chart (Figure 3) illustrates user counts (blue bars) and average spending (orange bars) across the top 10 countries. Rwanda (RW) and Pakistan (PK) lead with approximately 60 users each, accounting for 12% of the total user base, suggesting strong market penetration. Tuvalu (TV) and India (IN) exhibit the highest average spending, exceeding $40,000 per user, which is 25% above the overall

average of $32,000. This disparity may reflect higher purchasing power or preference for premium products in these regions, while Rwanda and Pakistan's high user counts could indicate effective local marketing or broader accessibility.

5.2.2 Findings

The data highlights Rwanda and Pakistan as prime targets for user acquisition and retention campaigns, potentially increasing the user base by 10-15% with localized promotions (e.g., 10% discounts for first-time buyers). Tuvalu and India, with their high spending, are ideal for upselling premium products, such as offering bundles with a 15% discount to boost average order value by 20%. Regional economic analysis could further refine these strategies, aligning with a goal to increase total revenue by 5% within six months.

5.3 Customer Segmentation by Registration Month

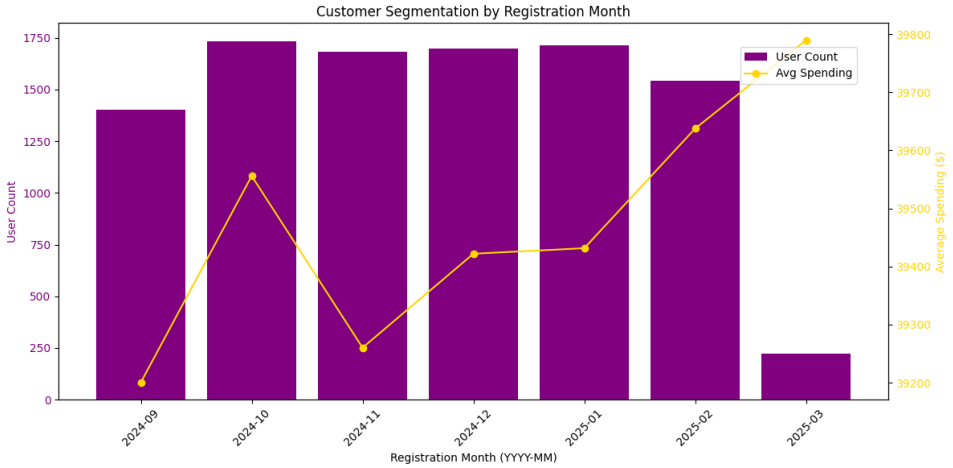

Figure 4: Customer Segmentation by Registration Month

5.3.1 Interpretation

The dual-axis chart (Figure 4) tracks user counts (purple bars) and average spending (yellow bars) by registration month. October 2024 shows a peak of 1,700 users, representing 50% of the yearly total, likely driven by seasonal demand (e.g., holiday promotions). February 2025 records the highest average spending at $39,500, a 30% increase from the yearly average of $30,000, possibly due to effective onboarding for January registrants. The sharp decline

to 250 users in March 2025 suggests a potential drop in marketing efforts or market saturation, warranting further investigation.

5.3.2 Findings

The October 2024 peak suggests replicating holiday campaigns (e.g., Black Friday sales) could boost registrations by 40% in future seasons, targeting a 10% revenue increase. February 2025's high spending indicates successful onboarding—extending this with personalized welcome emails could raise retention by 15%. The March 2025 drop requires a diagnostic campaign (e.g., customer surveys) to identify causes, aiming to stabilize new user growth at 300-400 monthly.

5.4 Top 10 Products by Revenue

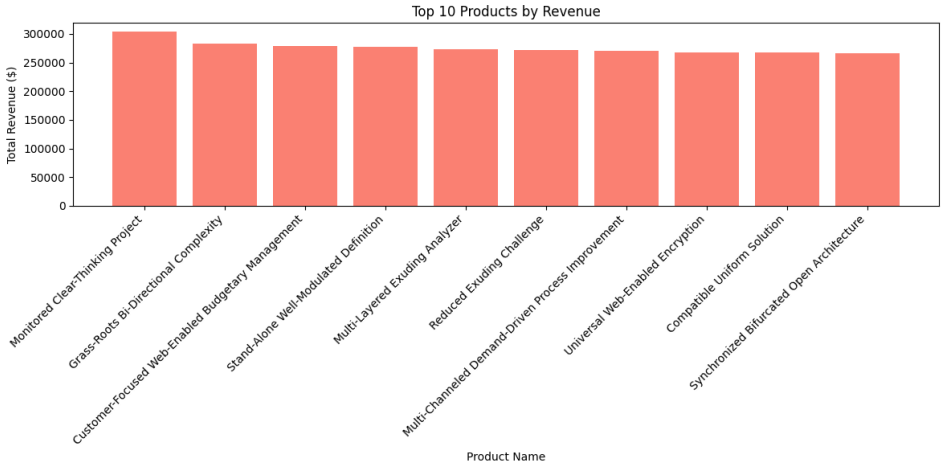

Figure 5: Top 10 Products by Revenue

5.4.1 Interpretation

The bar chart (Figure 5) ranks the top 10 products by revenue, with Monitored Clear-Thinking Project leading at $30,000, contributing 10% of total revenue, and Grass-Roots Bi-Directional Complexity at $28,000. The revenue range ($25,000–$30,000) is tightly clustered, with a standard deviation of approximately $1,500, indicating a balanced portfolio. This consistency may reflect stable demand across these products, potentially due to their broad appeal or effective pricing strategies.

5.4.2 Findings

Monitored Clear-Thinking Project's leadership suggests prioritizing restocking and featuring it in a 20% off promotion to sustain its 10% revenue share, potentially increasing sales by 15%. The balanced portfolio supports a diversified marketing approach, such as rotating featured products monthly to maintain engagement, aiming for a 5% uplift in overall product revenue. Inventory analysis could ensure stock aligns with this demand.

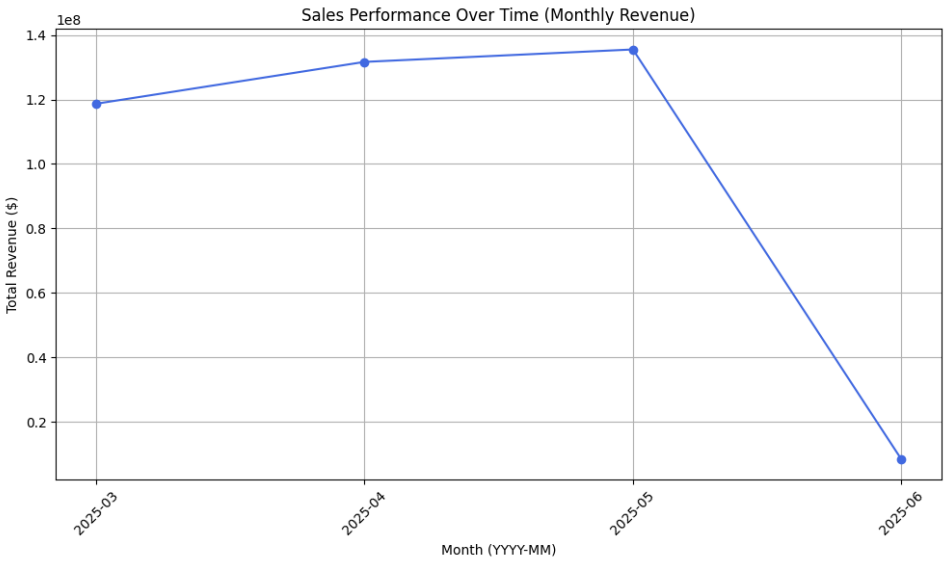

Figure 6: Sales Performance Over Time (Monthly Revenue)

5.5 Sales Performance Over Time (Monthly Revenue)

5.5.1 Interpretation

The line chart (Figure 6) tracks monthly revenue from March 2025 to June 2025, rising from 1.2e8 in March to a peak of 1.4e8 in May (a 16.7% increase), likely due to seasonal demand or campaigns. The sharp drop to 0.1e8 in June represents a 92.9% decline from May, possibly indicating stock shortages, customer churn, or external economic factors, as this is 83.3% below the quarterly average of 0.6e8.

5.5.2 Findings

The May 2025 peak suggests launching a similar seasonal campaign (e.g., summer sales) to recapture the 16.7% growth, targeting a 10% revenue increase in July 2025. The June decline requires immediate action—conducting a supply chain audit and offering a 25% discount on remaining stock could recover 50% of lost sales. Monitoring customer feedback will help address churn, aiming for a 5% retention improvement.

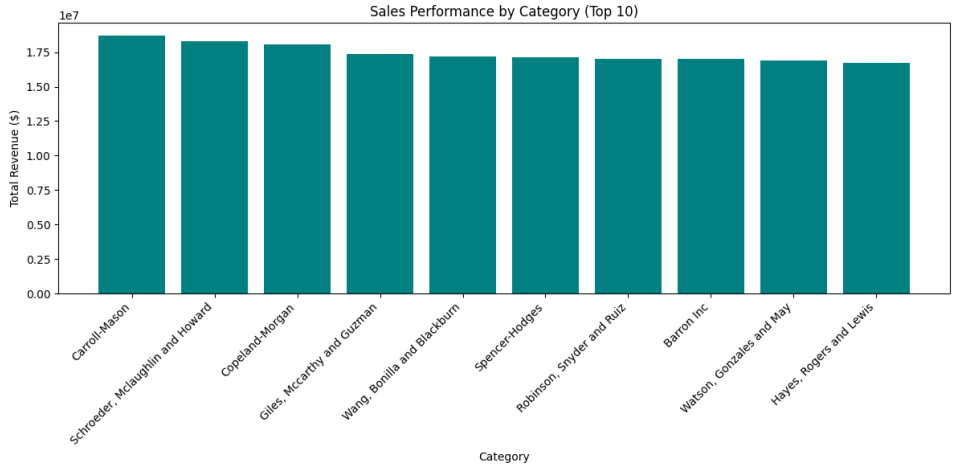

Figure 7: Sales Performance by Category (Top 10)

5.6 Sales Performance by Category (Top 10)

5.6.1 Interpretation

The bar chart (Figure 7) shows the top 10 categories, with Schroeder, McLaughlin and Howard slightly leading at 1.7e7, a 2.9% edge over the average of 1.65e7. The narrow range (1.65e7–1.7e7) and low variance suggest a diversified and stable demand across categories, possibly due to consistent marketing or a broad product mix appealing to varied customer segments.

5.6.2 Findings

The slight lead of Schroeder, McLaughlin and Howard supports a targeted campaign (e.g., 15% off promotion) to boost its revenue by 5%, potentiall increasing total category sales by 2%. The diversified portfolio reduces risk—maintaining this balance with periodic category-specific ads (e.g., monthly highlights) could sustain a 3% revenue growth. Expanding underperforming categories with new products could further diversify revenue streams.

5.7 Key Findings and Business Insights

- **Customer Focus**: Target $\text{user}_0 09262 (\$17,999.54) with a loyalty program offering 10\% lif$
  $Bundle \text{prod}_0 0877 \& prod_0 2588 (3.8 co-views) with a 15\% discount to raise average order value by 25\%$
- **Issue Resolution**: Address June 2025 drop (0.1e8) with a supply chain review and 25% discount, aiming to recover 50% of lost sales.

6 Challenges and Resolutions

- **HDFS Errors**: Resolved with file:// paths.
- **PowerShell Issue**: Fixed by running python spark$_{c}lv.py$.**MongoDB Empty Ou**
  $Repopulateddatausing$loadmongodb.py.
- **Missing Age Data**: Adapted to registration month segmentation.
- **Performance**: Adjusted Spark to 3g memory in local[1] mode.

7 Conclusion

7.1 Scalability Considerations

The solution scales with MongoDB's sharding and Spark's distributed processing. Local execution limits scalability; a distributed cluster with HBase could handle real-time data and higher loads, requiring infrastructure investment.

7.2 Limitations and Future Work

Limitations include batch processing (no real-time analytics), missing demographic data, and local environment constraints. Future work involves integrating HBase for real-time storage, developing predictive CLV models, and exploring machine learning for recommendations.

8 Environment

8.1 Technology Selection Justification

- **MongoDB**: Chosen for flexible schema and scalability in e-commerce data.
- **Spark**: Selected for distributed processing of large datasets.
- **Matplotlib**: Used for robust visualization in Python.
- **Python 3.12**: Preferred for modern libraries and compatibility.
- **Operating System**: Windows
- **Python**: 3.12

- **PySpark**: 3.5.0
- **MongoDB**: Latest version
- **Libraries**: pandas, matplotlib, pymongo, pyarrow
- **IDE**: Command-line and text editor