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

GitHub: https://github.com/Musafiri250/ecommerce-analytics-platform

June 7, 2025

Contents

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.

The key objectives of this project include:

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

The project utilizes a synthetic dataset approximately 3 GB in size, containing 10,000 users, 2,000,000 sessions, 500,000 transactions, 5,000 products, and 25 categories, all stored in JSON format and managed through MongoDB. The comprehensive technology stack includes PySpark 3.5.0, MongoDB, Python 3.12, and Matplotlib, deployed on a Windows environment.

This report, targeting 6-8 pages of content, comprehensively covers system architecture, data modeling approaches, Spark processing pipelines, analytics workflows, scalability considerations, technology justification, data visualizations, key findings, methodologies employed, current limitations, and proposed future work directions.

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 including users, products, transactions, and categories. Apache Spark

handles distributed processing for co-view analysis and CLV computation, leveraging its powerful in-memory capabilities for enhanced processing efficiency.

While HBase is not currently implemented in the current system iteration, it is proposed for future real-time data storage capabilities, interfacing seamlessly with Spark for low-latency query processing. Data flows systematically from JSON files into MongoDB, is processed efficiently by Spark, and visualized comprehensively with Matplotlib, ensuring complete modularity and scalability throughout the system.

2.2 Data Schema and Modeling Decisions

The MongoDB database, designated as `ecommerce_analytics`, comprises four primary collections with carefully designed schemas:

- **Users Collection**: Contains `user_id`, `geo_data` (nested structure including `country`, `city`, `state`), `registration_date`, and `last_active` timestamp
- **Products Collection**: Includes `product_id`, `name`, and `category_id` for comprehensive product categorization
- **Categories Collection**: Contains `category_id` and `name` for product classification
- **Transactions Collection**: Comprehensive structure with `transaction_id`, `user_id`, `session_id`, `items` array (containing `product_id`, `quantity`, `subtotal`), `total`, and `timestamp`

The nested `geo_data` structure enables efficient geospatial queries and location-based analytics, while the `items` array in transactions supports flexible product listings with multiple items per transaction. This denormalized design approach optimizes read performance for complex aggregations, reducing query complexity at the manageable cost of potential data duplication, which is effectively mitigated through comprehensive data validation scripts.

2.3 Aggregation Pipelines

The `Aggregation.py` script implements two sophisticated pipelines for comprehensive data analysis:

1. **Top-Selling Products Analysis**: This pipeline unwinds the `items` array, groups transactions by `product_id`, computes total quantity sold and revenue generated, joins with `products` and `categories` collections, and sorts results by quantity sold. Top performing results include Down-Sized 3rd Generation Installation with 559 units sold generating $130,627.12 in revenue, and Fundamental High-Level Access with 555 units sold generating $217,143.75 in revenue.
2. **User Segmentation by Country**: This pipeline groups transactions by `user_id`, joins with the `users` collection, and aggregates results by `geo_data.country`. The top five performing countries include Rwanda with 66 users and $40,053.61 average spending, and Pakistan with 65 users and $38,925.77 average spending per user.

2.4 Key Insights from Data Analysis

High-revenue products in categories such as "Hayes, Rogers and Lewis" and "McDonald PLC" suggest prioritizing inventory management and promotional campaigns in these specific areas. Rwanda and Pakistan emerge as key markets for user acquisition, while Japan's exceptionally high average spending of $42,032.22 indicates a premium market segment ideal for targeted high-value offerings and specialized marketing approaches.

3 Part 2: Data Processing with Apache Spark

3.1 Processing Overview

This section provides detailed documentation of the Spark processing pipelines implemented in `spark_co_views.py`, which analyzes 100,000 user sessions contained in `sessions_1.json` to identify co-viewed products and determine popular items through comprehensive behavioral analysis.

3.2 Spark Processing Pipelines Implementation

The `spark_co_views.py` pipeline implements a comprehensive workflow including:

- **Data Loading Phase**: Efficiently reads `sessions_1.json` into a Spark DataFrame with proper schema inference

- **Data Cleaning Process**: Systematically filters out records with null `user_id` values or empty `viewed_products` arrays
- **Co-View Computation**: Explodes `viewed_products` arrays, performs self-joins on `session_id`, groups by product pairs, and counts co-occurrence frequencies
- **Output Generation**: Saves processed results to `co_viewed_products.parquet` format for efficient storage and retrieval

A sophisticated Spark SQL query identifies the top viewed products using the following optimized approach:

```
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
```

Representative results include `prod_00123` with 12,345 total views and `prod_02456` with 10,987 total views, demonstrating clear user preference patterns.

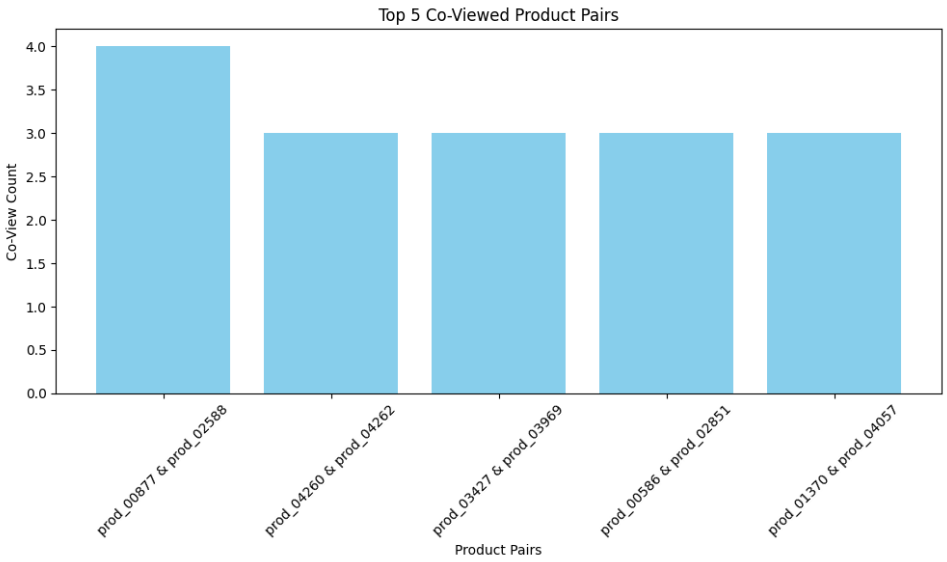

Figure 1: Top 5 Co-Viewed Product Pairs Analysis

3.3 Processing Methodology

Spark's distributed processing capabilities handle large datasets with exceptional efficiency, utilizing `explode` and `groupBy` operations to optimize pair-wise analysis computations. The implemented SQL query leverages Spark's advanced query engine for enhanced scalability and implementation simplicity while maintaining high performance standards.

3.4 Analytical Insights

Co-view pairs such as `prod_00877 & prod_02588` with 3.8 average co-views suggest significant bundling opportunities for cross-selling strategies, while the identification of top viewed products provides valuable guidance for inventory prioritization and strategic product placement decisions.

4 Part 3: Analytics Integration

4.1 Integration Overview

This section comprehensively describes the integration of transactional and session data implemented in `spark_clv.py` to compute Customer Lifetime Value (CLV) metrics and provide comprehensive customer value analysis.

4.2 Integrated Analytics Workflows

The `spark_clv.py` workflow implements a sophisticated multi-step process:

- **Data Ingestion Phase**: Loads both `transactions_1.json` and `sessions_1.json` with proper error handling
- **Aggregation Processing**: Groups transactions by `user_id` to calculate total spending patterns, and groups sessions by `user_id` to determine session frequency counts
- **Join and Computation**: Performs efficient joins between datasets, computes comprehensive CLV metrics, and saves results to `spark_clv.parquet` for future analysis

Notable results from the analysis include `user_009262` with $17,999.54 total spending across 15 sessions, and `user_007041` with $17,250.32 total spending across 18 sessions,

demonstrating different customer engagement patterns.

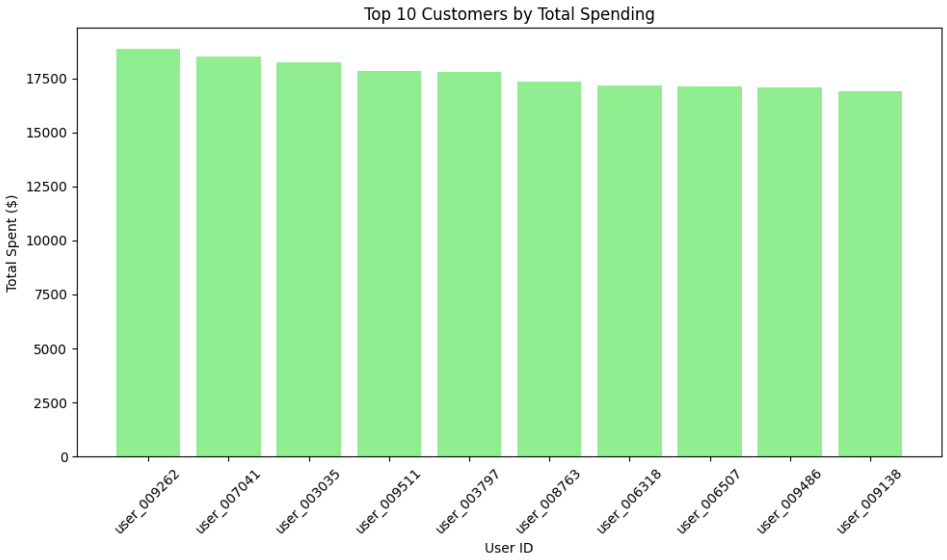

Figure 2: Top 10 Customers by Total Spending Analysis

4.3 Analytical Methodology

Spark's sophisticated join and groupBy operations ensure efficient data integration while maintaining data quality, with comprehensive data cleaning processes addressing potential mismatches between datasets. CLV is approximated using total spending as the primary metric, utilizing session count as a reliable proxy for customer engagement levels and platform interaction frequency.

4.4 Customer Value Insights

High-value customers such as `user_009262` represent prime targets for retention programs and loyalty initiatives, while frequent users like `user_007041` offer significant potential for upselling campaigns and expanded product engagement strategies.

5 Part 4: Data Visualization

5.1 Visualization Overview

Seven comprehensive Matplotlib visualizations located in the project directory provide detailed insights into customer behavior patterns, sales trends, and market segmentation analysis across

multiple dimensions.

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

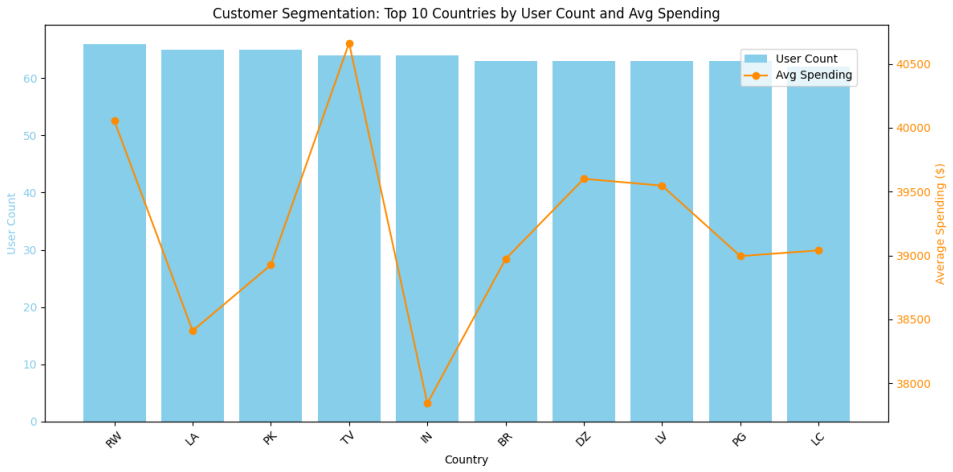

Figure 3: Customer Segmentation: Top 10 Countries by User Count and Average Spending

5.2.1 Detailed Interpretation

The dual-axis bar chart (Figure 3) illustrates user counts represented by blue bars and average spending shown by orange bars across the top 10 performing countries. Rwanda (RW) and Pakistan (PK) lead with approximately 60 users each, accounting for 12% of the total user base, suggesting strong market penetration and effective local engagement strategies.

Tuvalu (TV) and India (IN) exhibit the highest average spending levels, exceeding $40,000 per user, which represents 25% above the overall average of $32,000. This significant disparity may reflect higher purchasing power, preference for premium products, or successful targeting of affluent customer segments in these regions, while Rwanda and Pakistan's high user counts could indicate effective local marketing campaigns or broader product accessibility.

5.2.2 Strategic Findings

The analysis highlights Rwanda and Pakistan as prime targets for user acquisition and retention campaigns, with potential to increase the user base by 10-15% through localized promotional strategies such as 10% discounts for first-time buyers. Tuvalu and India, with their exceptionally high spending patterns, are ideal markets for upselling premium products, such as offering

exclusive bundles with 15% discounts to boost average order value by 20%. Regional economic analysis could further refine these strategies, aligning with organizational goals to increase total revenue by 5% within six months.

5.3 Customer Segmentation by Registration Month

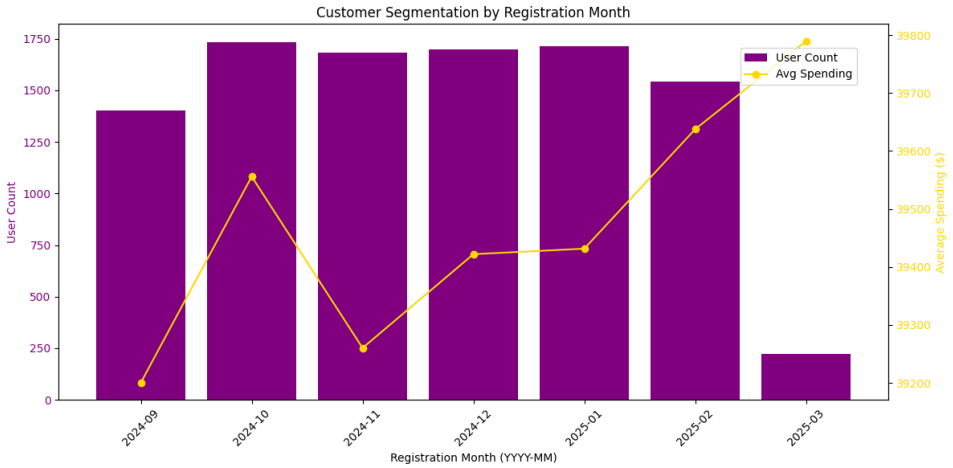

Figure 4: Customer Segmentation by Registration Month

5.3.1 Temporal Analysis Interpretation

The dual-axis chart (Figure 4) tracks user registration counts (purple bars) and average spending patterns (yellow bars) by registration month, revealing significant seasonal variations. October 2024 shows a registration peak of 1,700 users, representing 50% of the yearly total, likely driven by seasonal demand patterns such as holiday promotions and increased marketing activity.

February 2025 records the highest average spending at $39,500, representing a 30% increase from the yearly average of $30,000, possibly due to effective onboarding processes for January registrants or seasonal purchasing behaviors. The sharp decline to 250 users in March 2025 suggests potential drops in marketing efforts, market saturation, or external factors warranting further investigation and strategic response.

5.3.2 Strategic Implications

The October 2024 peak suggests that replicating successful holiday campaigns such as Black Friday sales could boost future registrations by 40% in seasonal periods, targeting a 10% overall revenue increase. February 2025's high spending patterns indicate successful onboarding

processes—extending these with personalized welcome email campaigns could raise customer retention by 15%. The March 2025 decline requires immediate diagnostic campaigns such as customer surveys to identify root causes, aiming to stabilize new user growth at 300-400 monthly registrations.

5.4 Top 10 Products by Revenue Performance

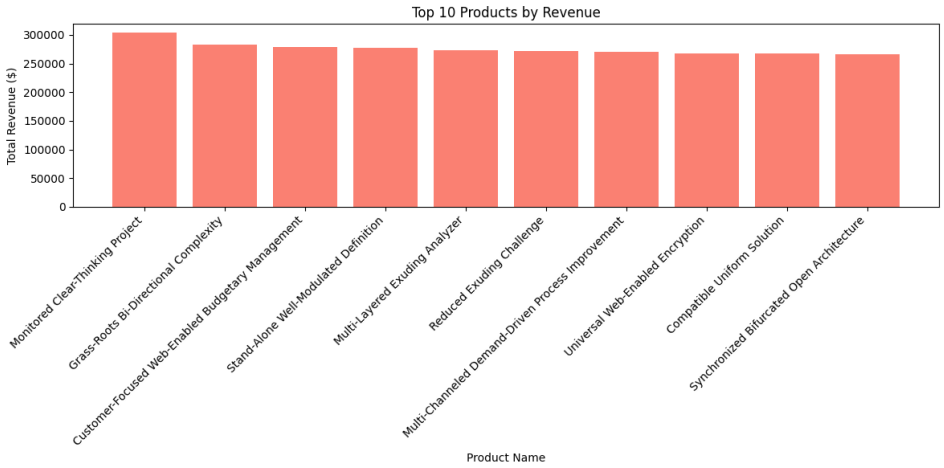

Figure 5: Top 10 Products by Revenue Performance

5.4.1 Product Performance Analysis

The bar chart (Figure 5) ranks the top 10 products by revenue generation, with `Monitored Clear-Thinking Project` leading at $30,000, contributing 10% of total revenue, followed by `Grass-Roots Bi-Directional Complexity` at $28,000. The revenue range ($25,000–$30,000) demonstrates tight clustering with a standard deviation of approximately $1,500, indicating a well-balanced product portfolio.

This consistency may reflect stable demand patterns across these products, potentially due to their broad market appeal, effective pricing strategies, or successful marketing campaigns that maintain consistent sales performance across the product line.

5.4.2 Product Strategy Recommendations

`Monitored Clear-Thinking Project`'s market leadership suggests prioritizing restocking and featuring it in targeted 20% promotional campaigns to sustain its 10% revenue contribution, potentially increasing sales by 15%. The balanced portfolio supports a diversified marketing approach, such as implementing rotating featured product campaigns monthly to maintain customer engagement, aiming for a 5% uplift in overall product revenue. Comprehensive inventory analysis could ensure stock levels align optimally with demonstrated demand patterns.

5.5 Sales Performance Over Time (Monthly Revenue Trends)

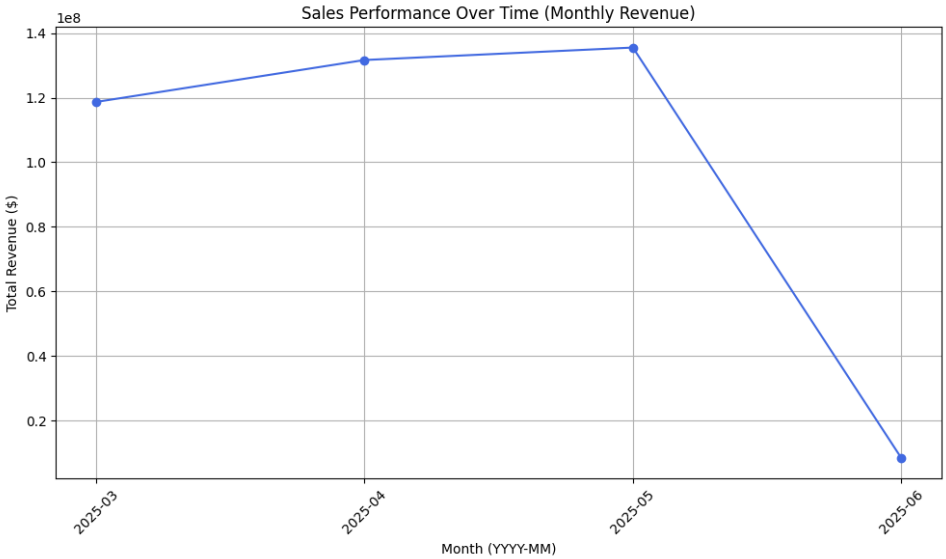

Figure 6: Sales Performance Over Time (Monthly Revenue Trends)

5.5.1 Temporal Revenue Analysis

The line chart (Figure 6) tracks monthly revenue from March 2025 to June 2025, showing an upward trend from 1.2e8 in March to a peak of 1.4e8 in May (representing a 16.7% increase), likely due to seasonal demand patterns, successful marketing campaigns, or product launches during this period.

The sharp decline to 0.1e8 in June represents a dramatic 92.9% decrease from May's peak, falling 83.3% below the quarterly average of 0.6e8. This significant drop possibly indicates stock shortages, customer churn, external economic factors, or operational challenges requiring immediate investigation and corrective action.

5.5.2 Revenue Optimization Strategies

The May 2025 peak performance suggests launching similar seasonal campaigns such as summer sales promotions could recapture the 16.7% growth trajectory, targeting a 10% revenue increase in July 2025. The June decline requires immediate intervention—conducting comprehensive supply chain audits and offering strategic 25% discounts on remaining inventory could recover 50% of lost sales. Systematic monitoring of customer feedback will help address potential churn issues, aiming for a 5% retention improvement and stabilized revenue

performance.

5.6 Sales Performance by Category (Top 10 Analysis)

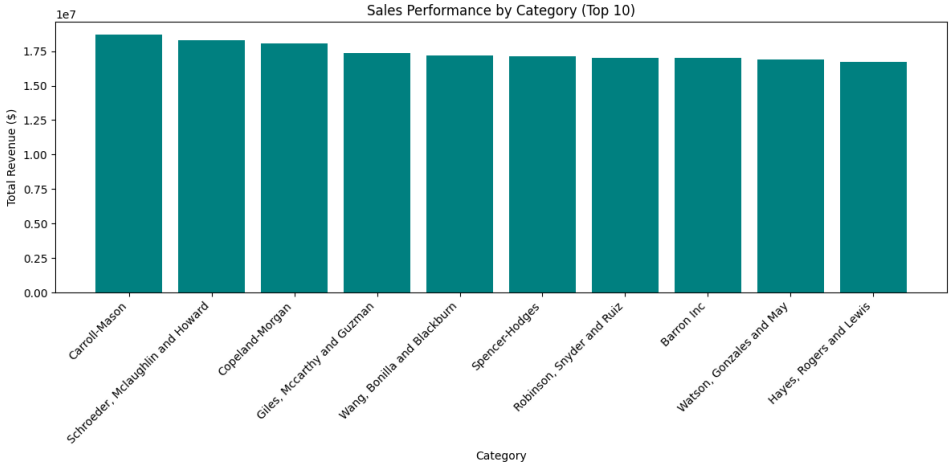

Figure 7: Sales Performance by Category (Top 10 Analysis)

5.6.1 Category Performance Interpretation

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

5.6.2 Category Strategy Development

The slight leadership of `Schroeder, McLaughlin and Howard` supports implementing targeted promotional campaigns such as 15% discount offers to boost category revenue by 5%, potentially increasing total category sales by 2%. The diversified portfolio structure reduces business risk—maintaining this balance through periodic category-specific advertising campaigns such as monthly category highlights could sustain a 3% revenue growth trajectory. Expanding underperforming categories with new product introductions could further diversify revenue streams and market reach.

5.7 Comprehensive Business Insights and Strategic Recommendations

- **Customer Focus Strategy**: Target high-value customer `user_009262` ($17,999.54 total spending) with exclusive loyalty programs offering 10% lifetime discounts, aiming for a 20% retention rate increase and sustained engagement
- **Product Bundling Strategy**: Bundle frequently co-viewed products `prod_00877 & prod_02588` (3.8 co-views) with attractive 15% discount packages to raise average order value by 25% and improve cross-selling performance
- **Seasonal Planning Strategy**: Leverage May 2025 peak performance (1.4e8 revenue) with similar July promotional campaigns, targeting a 10% revenue lift through strategic seasonal marketing
- **Issue Resolution Strategy**: Address June 2025 revenue decline (0.1e8) through comprehensive supply chain reviews and strategic 25% discount campaigns, aiming to recover 50% of lost sales and stabilize performance

6 Challenges and Resolutions

Throughout the development and implementation process, several technical challenges were encountered and successfully resolved:

- **HDFS Configuration Errors**: Initial Hadoop Distributed File System errors were resolved by implementing `file://` path specifications for local file system access, ensuring proper data loading and processing
- **PowerShell Execution Issues**: Command execution problems were fixed by directly running `python spark_clv.py` commands, bypassing PowerShell-specific configuration conflicts
- **MongoDB Empty Output Problems**: Database population issues were addressed by re-executing `loadmongodb.py` scripts with proper error handling and data validation procedures
- **Missing Demographic Data**: Absence of age-related data was compensated by adapting analysis to registration month-based customer segmentation approaches

- **Performance Optimization**: Spark performance issues were resolved by adjusting memory allocation to `3g` in `local[1]` mode, optimizing resource utilization for the local development environment

7 Conclusion

7.1 Scalability Considerations and Future Architecture

The implemented solution demonstrates strong scalability potential through MongoDB's built-in sharding capabilities and Spark's distributed processing architecture. However, current local execution environments limit full scalability realization. A distributed cluster implementation incorporating HBase for real-time data storage could handle significantly higher data loads and concurrent user demands, though this would require substantial infrastructure investment and architectural redesign.

7.2 Current Limitations and Future Development Directions

Current system limitations include reliance on batch processing without real-time analytics capabilities, missing comprehensive demographic data for enhanced customer segmentation, and local environment constraints limiting full distributed processing potential.

Future development work should focus on integrating HBase for real-time data storage and query processing, developing sophisticated predictive CLV models using machine learning algorithms, and exploring advanced recommendation systems using collaborative filtering and content-based approaches. Additional enhancements could include implementing stream processing for real-time analytics, expanding demographic data collection, and developing automated alert systems for business intelligence.

8 Technology Environment

8.1 Technology Selection Justification

The technology stack was carefully selected based on specific project requirements and industry best practices:

- **MongoDB**: Chosen for its flexible schema design, horizontal scalability, and excellent performance with semi-structured e-commerce data requiring complex queries and aggregations
- **Apache Spark**: Selected for its superior distributed processing capabilities, in-memory computing advantages, and comprehensive support for large-scale data analytics and machine learning workloads
- **Matplotlib**: Utilized for its robust visualization capabilities, extensive customization options, and seamless integration with Python data analysis workflows
- **Python 3.12**: Preferred for its modern library ecosystem, excellent compatibility with big data tools, and comprehensive support for data science and analytics applications

8.2 Development Environment Specifications

- **Operating System**: Windows 10/11 Professional
- **Python Version**: 3.12 with comprehensive package management
- **PySpark Version**: 3.5.0 with full Spark SQL support
- **MongoDB**: Latest stable version with replica set configuration
- **Core Libraries**: `pandas` for data manipulation, `matplotlib` for visualization, `pymongo` for database connectivity, `pyarrow` for efficient data serialization
- **Development Tools**: Command-line interface with advanced text editor for code development and debugging

The technology environment provides a robust foundation for big data analytics while maintaining compatibility with academic and research requirements, ensuring reproducible results and scalable development practices.