**Technical Project Report**

# Business Intelligence Dashboard for Malawian SMEs

**Project Title:** SME BI Dashboard (Malawi)
**Role:** Business Analyst & Developer
**Tools Used:** Python, Streamlit, Plotly, Facebook Prophet, Scikit-learn
**Duration:** 3 Weeks
**Date:** June 29, 2025

---

## 1. Project Overview

The SME BI Dashboard is a data-driven business intelligence web application built to serve the needs of Small and Medium Enterprises (SMEs) in Malawi. It enables SME owners and analysts to convert raw sales and customer data into strategic insights without requiring deep technical expertise.

The dashboard is a multi-page, interactive tool that includes modules for data visualization, anomaly detection, forecasting, customer segmentation, and downloadable reports. It is designed for accessibility, simplicity, and practical usability.

---

## 2. Objectives

- Equip Malawian SMEs with a low-cost, open-source alternative to expensive BI tools.
- Enable real-time performance monitoring, trend analysis, and customer insights.
- Simplify complex ML tasks (e.g., forecasting, clustering) into intuitive visual components.
- Ensure the system is scalable and adaptable for various SME industries (retail, agriculture, services).

---

## 3. Core Features

### Dashboard Modules

- **Sales KPIs**: Total sales, profit, and profit margin in real-time.

- **Regional & Demographic Breakdown**: Interactive visualizations by region, age group, and gender.
- **Top/Bottom Product Analysis**: Ranked insights into best and worst performing products.

**Analytics & Modeling**

- **Anomaly Detection**: Isolation Forest model highlights unusual sales days.
- **Forecasting**: Facebook Prophet projects future sales trends with confidence intervals.
- **Customer Segmentation**: K-means clustering identifies key customer groups based on purchase behavior.

**Data Handling**

- **Upload**: Users can upload custom sales data in CSV format.
- **Export**: Filtered datasets and reports can be downloaded in Excel or CSV.

**User Experience**

- Sidebar navigation for different modules.
- Clean, minimalist layout with a white background and bold text for clarity.
- Fully responsive across devices.

---

### 4. Technical Stack

- **Frontend**: Streamlit (custom CSS for styling)
- **Backend**: Python
- **Data Visualization**: Plotly
- **Machine Learning**: Scikit-learn (for Isolation Forest and K-means), Prophet (for forecasting)
- **Data Handling**: Pandas, NumPy

---

### 5. Challenges & Resolutions

### 1. Streamlit Learning Curve

Streamlit was a new framework for me at the start of this project. While its rapid prototyping capabilities are great, its structure and reactivity model initially felt limiting — especially when designing multi-page navigation and customized UI components.

**Resolution:**

- Studied the Streamlit documentation and community examples.
- Broke down each module (forecasting, anomaly detection, etc.) into isolated scripts to simplify logic.
- Implemented a sidebar-based navigation system to mimic a multi-page layout.

### 2. UI/UX Design Complexity

Creating a professional, user-friendly UI was the most challenging part. Streamlit has limited built-in customization options, and I had to rely heavily on CSS injection to achieve a consistent visual design.

**Resolution:**

- Used custom CSS in an external stylesheet to override default styles.
- Focused on readability: high contrast, minimal color distractions, and mobile-friendly layout.
- Maintained visual consistency across pages and components using reusable UI functions.

### 3. Code Consistency & Modularity

As the project grew in size and complexity, maintaining consistent logic and structure became difficult — especially when reusing functions across modules.

**Resolution:**

- Restructured the codebase using a modular approach: separated logic into files like kpis.py, forecasting.py, insights.py, etc.
- Followed a naming convention and standard formatting to keep the code readable.
- Implemented helper functions and centralized styling/configuration.

---

## 6. Outcomes

- Developed a full-stack, production-ready analytics dashboard tailored to SMEs in Malawi.
- Gained hands-on experience with machine learning in business applications.
- Enhanced my skills in data visualization, UI/UX design, and Streamlit-based development.
- Created a reusable framework that can be adapted for future dashboards or different business contexts.

---

### 7. Lessons Learned

- Streamlit is powerful but best suited for dashboards with well-defined logic and limited UI complexity.
- Good UI/UX is not just about how it looks — it's about guiding the user to insights quickly and effortlessly.
- Investing early in folder structure, code modularity, and naming consistency pays off in scalability and debugging.

---

### 8. Next Steps / Improvements

- Add authentication and user session handling.
- Enable dynamic database integration (e.g., PostgreSQL) for live data feeds.
- Add visual theming options for light/dark mode.
- Expand churn prediction logic and build a recommendation engine.

---

### 9. Conclusion

This project was a practical exercise in bridging business analysis and data science. It taught me how to turn messy, real-world data into clean visual insights that decision-makers can act on. Despite the learning curve with Streamlit and UI styling, the end result is a versatile, high-impact tool that demonstrates how open-source solutions can empower SMEs in developing economies.

---

**Prepared by:**
Mr. Kaonga
Business Analyst
mathiusmelo@gmail.com