E-Commerce with watsonx.data and Open Table Formats

Netra C N

Table of Contents

[1. Introduction 2](#_Toc183470283)

[1.1 Open Table Formats as Catalogs 2](#_Toc183470284)

[1.2 Why Integrate in a Unified Platform? 2](#_Toc183470285)

[2. E-Commerce Use Case Implementation 3](#_Toc183470286)

[2.1 Overview 3](#_Toc183470287)

[2.2 ERD Diagram 3](#_Toc183470288)

[2.3 Data Flow Overview 4](#_Toc183470289)

[2.4 Prerequisites 5](#_Toc183470290)

[*Other Execution Methods* 5](#_Toc183470291)

[3. E-commerce Technical Details 6](#_Toc183470292)

[3.1 Data Tables and Structure 6](#_Toc183470293)

[3.2 Code Breakdown: Iceberg and Delta Lake Integration 7](#_Toc183470294)

[**3.2.1 Iceberg Data Load - Code Overview** 7](#_Toc183470295)

[**3.2.2 Realtime Data Load generator - Code Overview** 8](#_Toc183470296)

[**3.2.3 Deltalake Data Load - Code Overview** 8](#_Toc183470297)

[4 Analytics with Cognos and Watsonx.data Integration 9](#_Toc183470298)

[4.1 Integrating Cognos with Watsonx.data 9](#_Toc183470299)

[4.2 Product Profile Dashboard 10](#_Toc183470300)

[5 Conclusion 11](#_Toc183470301)

# **1. Introduction**

The architecture of watsonx.data is purpose-built to enable scalable, efficient, and cost-effective data management for modern workloads. It comprises three key layers.

**1. Storage Layer:** The storage layer leverages low-cost object storage solutions, offering flexibility and scalability for storing large volumes of data. watsonx.data supports any S3-compatible storage, ensuring compatibility with popular cloud and on-premise solutions.

**2. Engines Layer:** watsonx.data integrates powerful compute engines, such as Apache Spark and Presto, which come native to the platform. Additionally, it supports Spark IAE (Integrated Analytics Engine) for enhanced processing. Beyond native engines, the architecture also supports other engines that can access the open storage buckets, providing flexibility for running diverse analytics workloads.

**3. Metadata Layer:** Moving further up the stack, the metadata store plays a critical role in the watsonx.data lakehouse architecture. It ensures that all query engines have a consistent view of the data being managed in the lakehouse storage. The metadata layer enables the implementation of open table formats that drive transactional consistency, schema evolution, and data versioning.

## Open Table Formats as Catalogs

watsonx.data supports three widely adopted open table formats to enhance flexibility and interoperability across analytics engines:

**Delta Lake**: Offers ACID transactions, schema enforcement, and the ability to handle both batch and streaming data seamlessly.

**Apache Hudi**: Focused on managing large-scale data, Hudi enables incremental updates and supports data mutation capabilities, making it ideal for e-commerce scenarios.

**Apache Iceberg**: Designed for handling petabyte-scale datasets, Iceberg provides advanced schema evolution, time travel, and partitioning flexibility.

These open table formats act as catalogs within the metadata layer, enabling robust management and efficient query execution across diverse engines and workloads.

## Why Integrate in a Unified Platform?

The watsonx.data lakehouse leverages the combined strengths of Delta Lake, Apache Hudi, and Apache Iceberg to address diverse business needs efficiently for example.

* In E-Commerce, Delta Lake enables real-time inventory updates, Hudi supports efficient customer behaviour tracking, and Iceberg powers sales trend analysis and forecasting for deeper insights.
* In Financial Services, Delta Lake ensures real-time fraud detection, Hudi manages incremental portfolio updates, and Iceberg secures compliance reporting with scalable governance.
* In Healthcare, Delta Lake facilitates timely predictive analytics, Hudi synchronizes patient records, and Iceberg handles long-term storage for audits.

By combining these open table formats, businesses can seamlessly manage real-time operations, process incremental updates, and ensure scalable, compliant analytics—all within a single, unified lakehouse platform.

# **2. E-Commerce Use Case Implementation**

## 2.1 Overview

In today’s fast-paced digital economy, e-commerce platforms must continuously evolve to meet the growing demands of both customers and businesses. To stay competitive, companies need to optimize their platforms for efficiency, scalability, and performance. Advanced data solutions are key to achieving these objectives.

This document explores how **watsonx.data**, combined with **Spark** and Open Table Formats like **Delta Lake** and **Apache Iceberg**, can be leveraged to optimize e-commerce operations. By integrating these powerful technologies, businesses can streamline data processing, enhance real-time analytics, and enable seamless scalability, all of which lead to better decision-making and improved customer experiences.

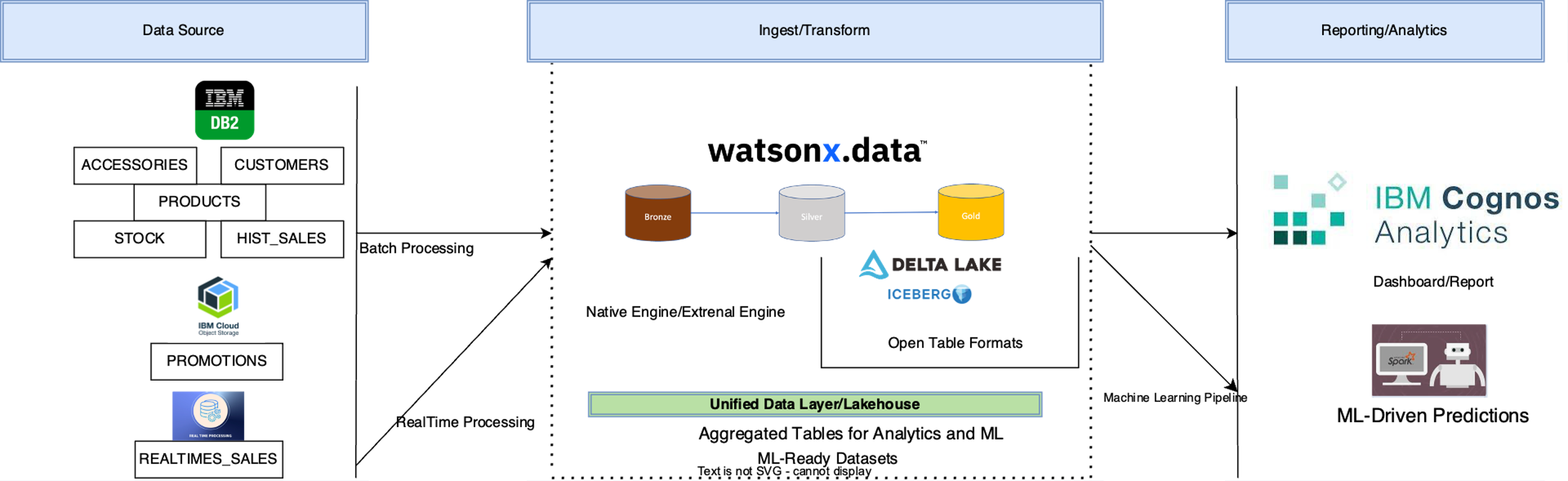
The following sections will outline the key challenges faced by e-commerce platforms, the technical solutions provided by these technologies, and the benefits of implementing this optimization strategy.

## 2.2 ERD Diagram

**A diagram of a retail system

Description automatically generated**

## 2.3 Data Flow Overview



1. Data Ingestion:  
• Real-time: Ingest web and app interactions into Delta Lake via watsonx.data.  
• Batch: Import historical data periodically into Iceberg.  
2. Real-Time Processing:  
• Spark on watsonx.data joins Product, Sales, and Customer tables for real-time queries like product recommendations and marketing.  
3. Federated Queries:  
• Combines real-time sales data from Delta Lake with historical data from Iceberg to provide insights on trends and customer behavior.

## 2.4 Prerequisites

IBM Watsonx.data provides multiple ways to run Spark workloads, catering to diverse analytical and operational needs. Below are the key methods:

#### 1. Native Spark Engine

* Watsonx.data allows provisioning a *Native Spark Engine* directly within the infrastructure.
* This option is ideal for running high-performance Spark applications and managing complex analytical workloads.
* Key Features:
  + Supports configurable Spark runtime versions.
  + Integration with associated catalogs for metadata management.
  + Utilizes storage volumes for Spark logs and events.
* *Setup*:
  + Provision through the Watsonx.data console under Infrastructure Manager.
  + Select or create storage volumes and specify the Spark runtime.

#### 2. IBM Analytics Engine (IAE)

* Provides a *serverless Spark environment* integrated with Watsonx.data.
* Best suited for lightweight and ad hoc Spark jobs without requiring infrastructure management.
* Key Features:
  + Automatic scaling and resource optimization.
  + Integrated with Hive Metastore (HMS) for catalog management.

### *Other Execution Methods*

#### 3. Watson Studio Notebooks

* Use interactive Python or Scala notebooks in Watson Studio to develop and run Spark applications.
* Ideal for real-time data exploration, visualization, and iterative development.

#### 4. Command-Line Execution (Python Scripts)

* Spark applications can be packaged as .py files and executed outside Watson Studio.
* Use curl commands or APIs to submit these jobs to Watsonx.data for execution.

To understand more about spark set up go through this[**link**](https://www.ibm.com/docs/en/watsonx/watsonxdata/2.0.x?topic=working-spark)**.**

For an easier setup, it is recommended to first try the Watson Studio method. This approach simplifies the integration of Spark, watsonx.data, and Open Table Formats like Delta Lake and Apache Iceberg. Watson Studio provides an intuitive environment with built-in tools for managing and analyzing data, which makes the process more straightforward. You can quickly deploy and manage Spark jobs and integrate with your e-commerce data for real-time analytics and scalable processing.

To get started with Watson Studio, you can follow this [**link**](https://www.ibm.com/docs/en/watsonx/watsonxdata/2.0.x?topic=wscpd-running-spark-notebook-from-watson-studio-cloud-pak-data)for detailed instructions on how to set up and use the platform.

# **3. E-commerce Technical Details**

## 3.1 Data Tables and Structure

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name** | **Data Source** | **Purpose** | **Storage Type** |
| Products | DB2 | Stores product details and metadata for current products | DB2 |
| Customers | DB2 | Tracks customer details and behaviour (real-time and historical) | DB2 |
| Accessories | DB2 | Stores accessories details and metadata for current accessories | DB2 |
| Hist\_sales | DB2🡪 (wxd) | Historical sales data | Iceberg |
| Realtime\_Sales | Realtime code🡪wxd | Current real time sale generated through a python code | Delta lake |
| Stock | DB2 | Stock data for products and accessories | Db2 |
| Promotions | DB2 | Promotions related to products | IBM COS |

## 3.2 Code Breakdown: Iceberg and Delta Lake Integration

As part of our implementation, we leverage Medallion Architecture to ensure a robust and scalable data pipeline. This architecture is structured around three layers:

* Bronze Layer: Raw, unprocessed data captured in its original format.
* Silver Layer: Cleaned and enriched data ready for business-specific processing.
* Gold Layer: Aggregated, transformed data optimized for analytics and reporting.

To support this architecture, we utilize three key files:

### **3.2.1 Iceberg Data Load - Code Overview**

This file is part of our Medallion architecture, and its primary focus is on loading data into the bronze, silver, and gold layers using Iceberg tables. Below are the key functions:

**initialize\_spark():** Sets up the Spark session with necessary configurations, including the connection to Watsonx.data, DB2, and Iceberg setup.

**list\_databases(spark):** Lists the available databases from the lakehouse to ensure the connection is correctly established.

**create\_databases(spark):** Creates the Bronze, Silver, and Gold databases in the lakehouse, ensuring proper organization of the data layers.

**migrate\_db2\_to\_bronze(spark):** Loads data from DB2 tables (e.g., HIST\_SALES) into the Bronze layer as Iceberg tables, making it available for further processing in the Silver layer.

**bronze\_ingest\_from\_csv\_temp\_table(spark):** Ingests CSV data (e.g., promotions) into the Bronze layer and creates a temporary table, ensuring the data is ready for transformation.

**create\_enhanced\_aggregated\_sales\_table(spark):** Aggregates sales, stock, and promotions data to generate an enhanced sales table in the silver layer for further analysis.

**create\_customer\_segmentation\_table(spark):** Segments customers based on their purchase patterns and demographic information, loading the results into the silver layer for actionable insights.

The entire process follows the Medallion architecture to ensure seamless data processing from raw (Bronze) to transformed (Silver) and finally aggregated (Gold) layers.

For the complete code, please check the [GitHub link](https://github.ibm.com/ecosystem-engineering-si-lab/E-Commerce-with-watsonx.data/blob/main/src/Iceberg_code_ecommerce.ipynb).

### **3.2.2 Realtime Data Load generator - Code Overview**

**setup\_and\_ingest\_realtime\_sales(spark):**

This function is designed to set up the necessary configurations for the Spark session and handle the real-time ingestion of sales data into the bronze layer of the Delta Lake architecture. It creates a Delta table for real-time sales data and continuously ingests synthetic sales data, simulating a live sales environment. The data is stored in the realtime\_sales\_new table in the bronze layer for further processing and analysis.

For the complete code, please check the [GitHub link](https://github.ibm.com/ecosystem-engineering-si-lab/E-Commerce-with-watsonx.data/blob/main/src/code_realtime.ipynb).

### **3.2.3 Delta Lake Data Load - Code Overview**

This file is part of our Medallion architecture, and its primary focus is on processing customer segmentation, sales data, promotional messages, product recommendations, and creating databases for seamless data flow using Delta Lake. The data is processed and stored in the bronze, silver, and gold layers, following a structured approach. Below are the key functions:

**create\_databases(spark):**

Sets up the Bronze, Silver, and Gold databases in the lakehouse to ensure the proper organization of data across different layers. This function initializes the database structure required for seamless data processing.

**process\_promotions(spark):**

Generates promotional messages for different customer segments based on their purchase patterns and segments. It processes customer data, creates targeted promotional messages, and saves the results into the Gold layer.

**product\_Sales\_Data(spark):**

Loads and processes product sales data from various sources, aggregates it for analysis, and saves it in the Silver layer for deeper insights into sales performance.

**process\_sales\_data(spark):**

Handles the transformation of raw sales data into usable insights by aggregating and processing it. This data is then stored in the Silver layer for further analysis and reporting.

**recommend\_products\_and\_accessories\_per\_customer(spark):**

Uses machine learning models or rule-based logic to recommend products and accessories to customers based on their purchase history, increasing customer engagement and sales.

These functions ensure seamless data processing from raw data ingestion to transformation and final output generation.

For the complete code, please check the [GitHub link](https://github.ibm.com/ecosystem-engineering-si-lab/E-Commerce-with-watsonx.data/blob/main/src/spark%20and%20delta.ipynb).

# **4 Analytics with Cognos and Watsonx.data Integration**

In today's data-driven business environment, leveraging advanced analytics tools is crucial for gaining insights and making informed decisions. Cognos Analytics provides an intuitive platform to create powerful dashboards and reports. By integrating Cognos with Watsonx.data, you can easily access and analyze large datasets stored in your Gold layer to uncover actionable insights.

This integration enables you to connect seamlessly to your Watsonx.data environment, allowing you to use the data stored in Gold layer tables for various analytics purposes. With this setup, you can efficiently explore data, build reports, and design dashboards that meet your business needs.

## 4.1 Integrating Cognos with Watsonx.data

To integrate Cognos Analytics with Watsonx.data, follow these key steps:

**Connection Setup:** Configure Cognos to connect to your Watsonx.data environment. This will allow Cognos to access the data stored in your Gold layer tables, making it available for further analysis.

**Accessing Data:** Once the integration is complete, Cognos can pull data directly from the Gold layer tables stored in Watsonx.data. This data can include various business metrics like sales, inventory, customer behaviour, and more.

**Creating Dashboards/Reports:** With the data now available in Cognos, you can create custom dashboards that provide actionable insights. The Gold layer tables serve as the foundation for these dashboards, offering real-time and historical data that can be analysed to track business performance.

For a detailed guide on how to connect Cognos with Watsonx.data, please refer to the screenshots and link provided. These resources will walk you through the steps to integrate and begin using your Gold/Silver/Bronze layer data in Cognos Analytics.

Refer [this document](https://www.ibm.com/docs/en/cognos-analytics/12.0.0?topic=administration-new-watsonxdata-connection-editor).

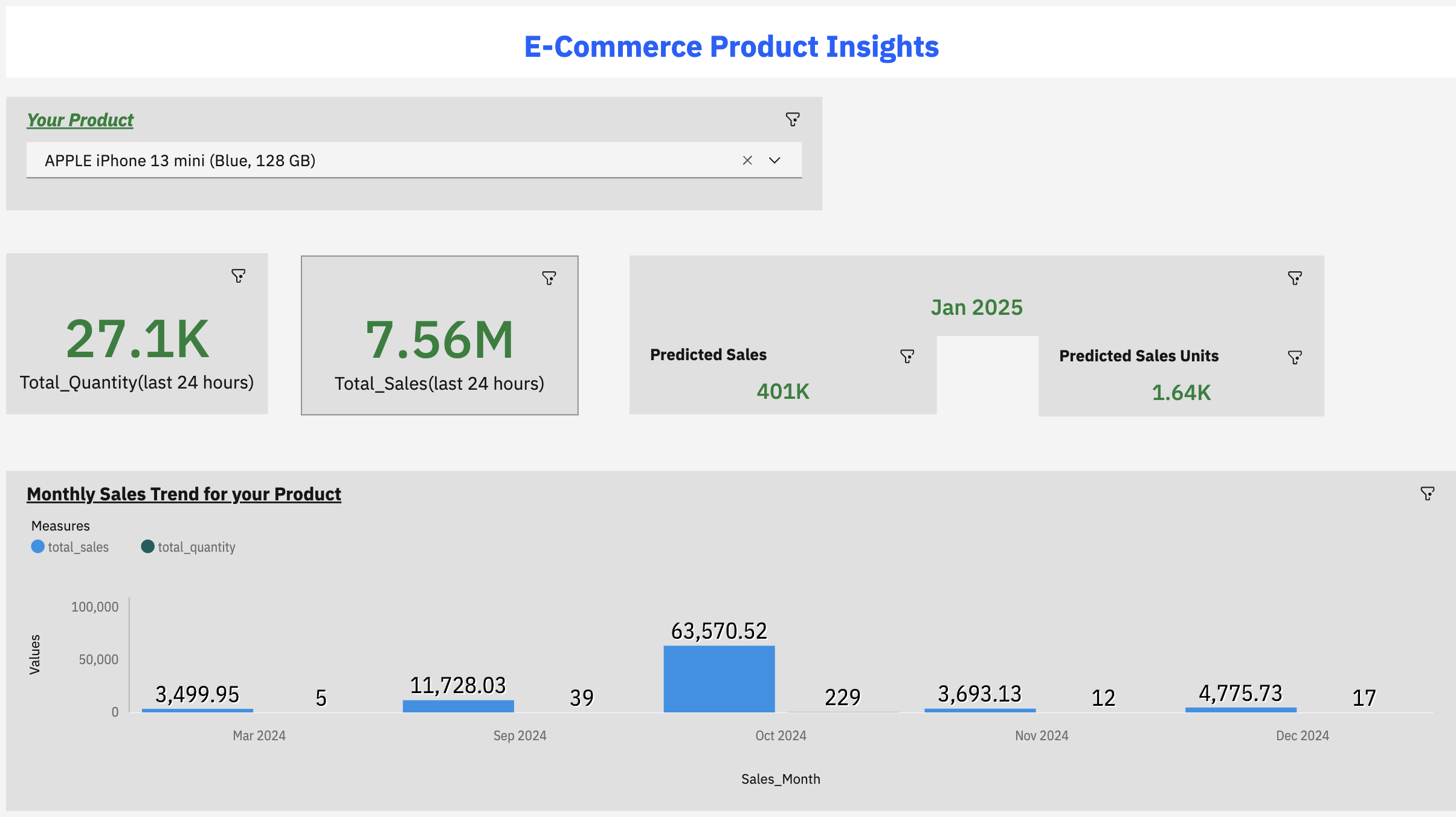
## 4.2 Product Profile Dashboard

As part of the initial analytics implementation, the Product Profile dashboard has been created. This dashboard leverages sales trend and prediction data from your tables to provide an in-depth view of product performance. Key features of this dashboard include:

Sales Trend Analysis: Visualizing product sales over time to track performance.

Sales Predictions: Forecasting future sales based on historical data and trends.

While the Product Profile dashboard serves as a starting point, it is only one example of the many insights you can gain by leveraging other tables.



**Expanding with Additional Dashboards**

Beyond the Product Profile dashboard, there are several other dashboards you can create using the data stored in Gold layer tables. Some potential dashboards include:

**Customer Segmentation:** Segment customers based on their purchasing behaviour and demographics, enabling personalized marketing strategies.

**Promotional Effectiveness:** Measure the impact of different promotional campaigns on sales and customer engagement.

**Inventory Management:** Visualize stock levels and forecast demand to optimize inventory management.

These dashboards, built on data from the tables, offer deeper insights into various aspects of the business, helping you to make data-driven decisions across multiple functions.

# **5 Conclusion**

This document showcases the integration of **Watsonx.data**, **Spark**, and **Cognos Analytics** to enable advanced data analytics and reporting. Using the **Medallion Architecture** with open table formats like **Delta Lake** and **Iceberg**, data is processed through the **Bronze**, **Silver**, and **Gold** layers for clean, enriched insights.

Connecting **Watsonx.data** to **Cognos Analytics** allows users to build dynamic dashboards, such as product sales trends, using data from the **Gold layer**. Spark ensures scalable data processing, and open table formats provide flexibility for future growth.

Overall, this setup empowers businesses to leverage data efficiently, driving better decision-making and personalized customer experiences.