Se former autrement

HONORIS UNITED UNIVERSITIES

2025 / 2026

Specialization: ERP-BI

Development of a Business Intelligence Solution: Price Comparison Platform for Retail and Tourism

Prepared by:

Nour Ben Abid, Nour Boukhris

Assma Hajbi, Erij Triaa

Roua Mtar, Mohamed Aziz Ben Ajmia

Supervised by:

Mrs. Sirine Zaabouti, Mrs. Hela Mejri, Mrs. Ines Mhaya

Acknowledgements

This project represents the culmination of months of dedication, collaboration, and learning—and it would not have been possible without the unwavering support of those who guided us along the way.

We are proud to be students of **ESPRIT**, a university that fosters innovation, critical thinking, and excellence. Throughout this journey, we have not only built a technical solution but also grown as aspiring professionals, thanks to the expertise and mentorship we received.

Our sincere gratitude goes to:

- **Mrs. Sirine Zaabouti** and **Mrs. Hela Mejri**, for their sharp expertise in Business Intelligence and their constant guidance, which shaped the foundation of this project with clarity and structure.

- **Mrs. Ines Mhaya**, whose strategic approach to Data Marketing challenged us to think beyond data and understand the business value behind every insight.
- **Mrs. Safa Toumi**, for her unwavering support in English communication, helping us express our work in a way that reaches both technical and non-technical audiences.

- **Mrs. Ines Chouchane**, for igniting our creativity through her teachings in innovation, encouraging us to build a solution that goes beyond expectations.

To all our professors, thank you for your dedication, your trust, and the time you invested in our growth. Your passion for teaching and excellence has left a lasting impact on us.

With deep appreciation and sincere thanks,
The DealDynamo Team,

Table of Figures:

General Introduction

In today's digital world, consumers are exposed to a wide range of products and services offered across numerous online platforms. While this variety brings greater choice, it also creates complexity—especially when it comes to comparing prices and selecting the best deal quickly and efficiently. As e-commerce continues to expand, both consumers and businesses are in growing need of smart systems that can track, analyse, and visualize price variations across the market. In this context, Business Intelligence (BI) plays a crucial role in transforming raw data into valuable insights for decision-making.

The main objective of this project is to design and develop a **price comparison and decision-support system** using Business Intelligence and Machine Learning techniques. This solution targets two types of decision-makers: **clients** (who seek the best prices available) and **sales managers** (from companies such as **Carrefour, Géant, MG** for products, and **Travel To Do, Tunisie Booking, and Touring** for services like hotel reservations and travel packages). The system allows both user groups to access reliable information on current prices, trends, and competitor offerings, helping them make data-driven decisions.

The project consists of collecting price data from multiple internal and external sources, transforming and cleaning the data, storing it in a structured format, and presenting it through **interactive dashboards** built with Power BI. These dashboards enable users to compare offers between vendors, monitor pricing dynamics, and gain strategic insights. In addition, we have integrated **Machine Learning models** for tasks such as **price prediction** and **recommendation** , offering even deeper analytical capabilities.

This is not an e-commerce website but a **Business Intelligence application** designed to support pricing strategies and consumer choices. The system follows a clear pipeline: data extraction, transformation, modelling, and visualization. It showcases how powerful tools such as **Power BI**, **Python**, and **SQL** can work together to transform raw market data into actionable insights.

Throughout this report, we describe the different stages of our work: the context of the project, the technical and functional requirements, the development process, and the presentation of the final dashboards and results. We also highlight the challenges we encountered and how they were resolved.

By working on this project, we not only demonstrated the practical application of BI and ML tools but also gained valuable experience in building real-world data-driven systems. This project prepares us to handle future professional scenarios where decision-making relies heavily on clear, visual, and accurate data analysis.

Chapter 1: General Project Context

1.1 Introduction

This chapter provides an overview of the project context. It outlines the business problem addressed, the goals we aim to achieve, and the technological choices made to design and implement the solution.

1.2 Existing Solutions

1.2.1 Study of Existing Solutions

Several platforms and tools already exist in the market for price comparison across different sectors such as retail, travel, and e-commerce. Well-known international platforms like **Google Shopping**, **Pricena**, and **Skyscanner** allow users to compare prices for products, hotel rooms, or flights in real time. In Tunisia, however, such solutions remain limited in functionality, coverage, or localization.

Some solutions are focused solely on consumer needs, offering basic comparisons without advanced analytics. Others are integrated into large e-commerce sites but lack transparency or multi-vendor comparisons. Furthermore, many existing platforms do not provide predictive insights or personalized recommendations based on user behavior and market trends.

1.2.2 Critique of Existing Solutions

Despite the presence of several price comparison tools in the market, most existing solutions reveal important shortcomings—especially when analyzed through the lens of Business Intelligence and local market relevance.

Fragmented Experience: Many platforms specialize in a single domain—either retail, travel, or electronics—without offering a unified experience for comparing different product categories (e.g., flights, hotels, and grocery items) in one place. This fragmentation forces users to switch between multiple services.

No Strategic Value for Vendors: While consumers are offered basic comparisons, vendors have no access to meaningful insights like pricing trends, competitive positioning, or customer interest over time. This one-sided value proposition limits the usefulness for businesses.

Minimal Intelligence, Maximum Manual Effort: Many tools rely on static data or simple scraping techniques, with no data modeling, historical analytics, or machine learning components. This results in platforms that show what prices are, but not **why** they fluctuate or **when** to buy.

Poor Integration Capabilities: These platforms rarely offer APIs, dashboards, or exportable reports that would enable sales managers or analysts to leverage the data for broader decision-making processes.

1.2.3 Actors and Market Trends

The price comparison and data-driven marketing ecosystem is evolving rapidly, influenced by trends such as:

- **Increased Consumer Demand for Transparency**: Users want to make informed decisions based on accurate, real-time data.

- **Growing Role of AI and Machine Learning**: Predictive analytics is becoming a differentiator in e-commerce platforms.

- **Data-Driven Decision Making**: Businesses increasingly rely on BI tools to optimize pricing strategies and gain competitive advantages.

In this dynamic landscape, **DealDynamo** positions itself as an innovative solution tailored to the needs of the Tunisian market, offering interactive dashboards, advanced price analytics, and actionable insights for both clients and sales managers.

1.3 Project Presentation

1.3.1 Project Background

In the digital age, online shopping has become a daily habit. However, the abundance of platforms and offers makes it difficult for users to identify the best price for a given product. Prices may differ significantly from one vendor to another. This creates a real need for a tool that gathers and compares prices efficiently.

Our project responds to this challenge by delivering a smart Business Intelligence solution that empowers users to effortlessly compare prices across multiple platforms, while enabling sales managers to stay ahead of the competition with real-time market insights.

1.3.2 Project Objectives

The main objective of this project is to build a comprehensive Business Intelligence solution that contributes to revenue growth by serving **two key decision-makers: sales managers and customers**.

For sales managers, the system helps in one hand to tracktheir CA and compare it to their competitors, in the other hand to improve sales by recruiting new customers, retaining loyal ones, and analyzing product purchase behavior across regions and customer types.

It allows managers to compare product prices across stores, identify major clients, and segment them geographically or by loyalty level.

On the customer side, the system enhances product selection by centralizing price comparisons, tracking price trends among competitors, and predicting the ideal products based on personal preferences.

Additionally, it provides intelligent recommendations for nearby stores and generates predictive alerts when prices are expected to drop. By integrating internal and external price data, organizing it in a structured data warehouse, and visualizing it through Power BI dashboards, this project empowers both business users and consumers to make better, data-driven decisions.

1.3.2.1 Business objectives

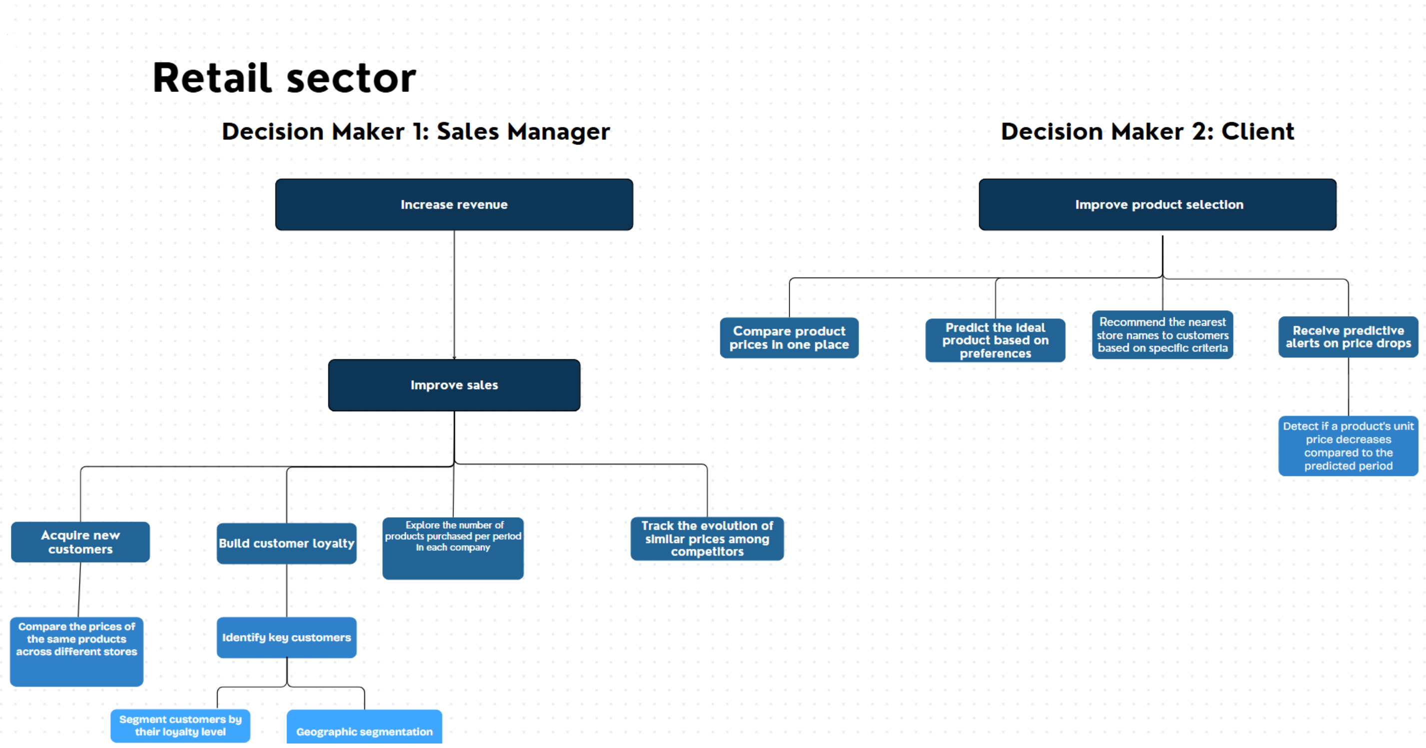


Figure 1 – equivalence tree-for the Retail Sector

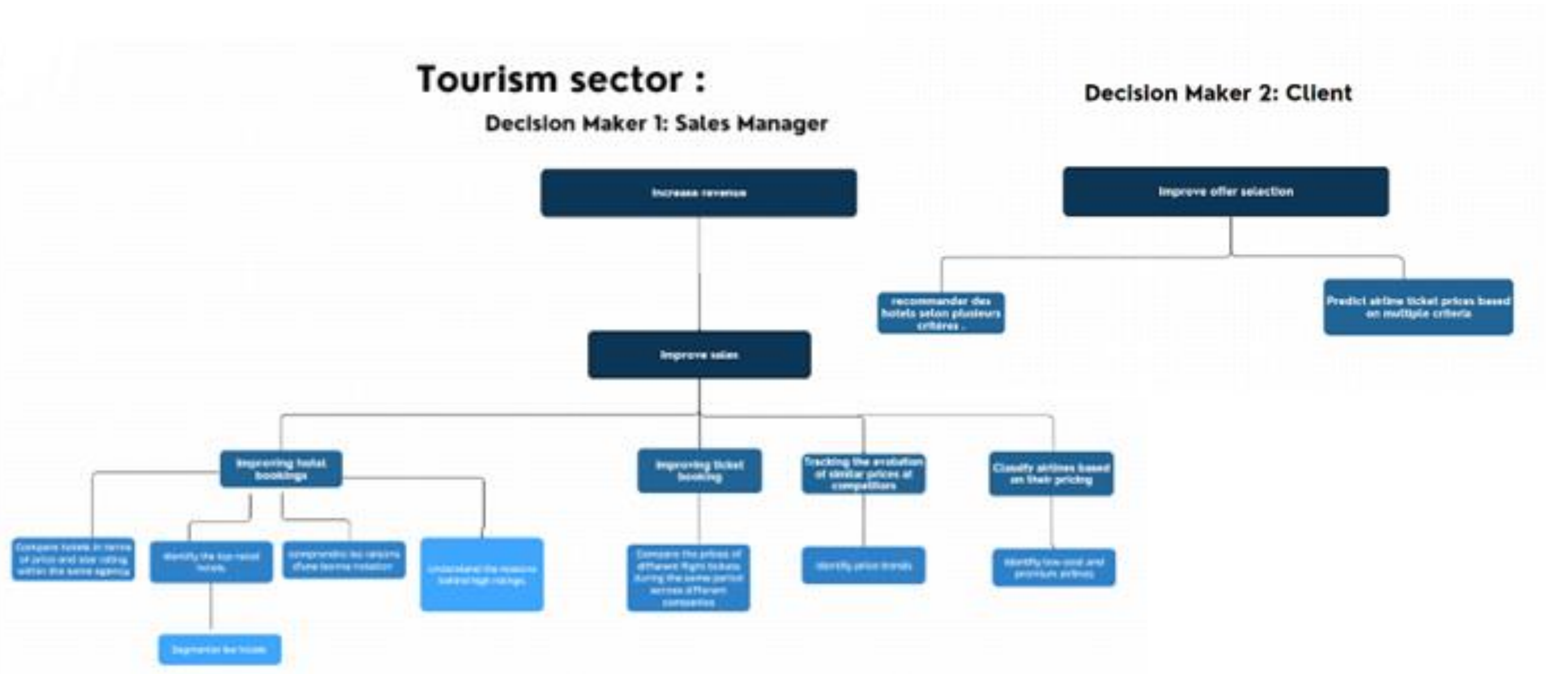


Figure 2 – equivalence tree-for the Tourism Sector

1.3.3 Proposed Approach:

To address the identified gaps in existing solutions, our approach focuses on developing a comprehensive Business Intelligence system that combines data integration, visualization, and predictive analytics. The goal is to empower both consumers and sales managers with actionable insights derived from real-time and historical data.

1. Data Collection

We began by gathering data from both internal sources (such as product catalogs and historical pricing) and external sources (notably through web scraping). This step was essential to obtain a comprehensive and diverse dataset for analysis.

2. Staging Area

The staging area acted as a temporary storage zone for raw data before integration. It allowed us to consolidate information from various sources—internal and external—while managing inconsistencies. This preparation phase ensured the data was clean and structured for further ETL operations using Talend.

3. Data Warehouse

We designed a data warehouse based on a constellation schema (also known as a galaxy schema) to support multiple business domains: products, hotels, and flights.

This approach enabled the reuse of shared dimension tables across several fact tables, offering greater flexibility and efficiency in handling complex queries.

4. Data Integration (ETL)

We utilized **SSIS (SQL Server Integration Services)** and **SSMS (SQL Server Management Studio)** for the extraction, transformation, and loading of raw data. These tools helped automate the data pipeline and ensure consistent integration into the warehouse.

5. Machine Learning with Python

Predictive models were developed using Python to support recommendations and price predictions

6. Data Visualization with Power BI

Interactive dashboards were built with Power BI to display key insights for both clients and decision-makers.

7. Web Application Development

To deliver an intuitive and seamless experience, we built a complete web application using a modern full-stack architecture:

a. **Backend – Flask**

We leveraged Flask, a lightweight Python web framework, to integrate machine learning models and manage API communication with the data warehouse.

b. **Frontend – Angular**

The frontend was built with Angular, providing:

- A responsive and intuitive user interface
- Real-time visualization of dashboards and predictive results

This integrated platform successfully brought together business intelligence, machine learning, and user interface design, empowering both clients and decision-makers with timely and relevant insights.

1.4 System Architecture

The system architecture of our project showcases the complete data pipeline—from gathering internal and external pricing data to processing it through ETL and storing it in a data warehouse.

This structured data is then used to build insightful dashboards and train machine learning models, all seamlessly integrated into a web application for both clients and sales managers to make informed decisions.

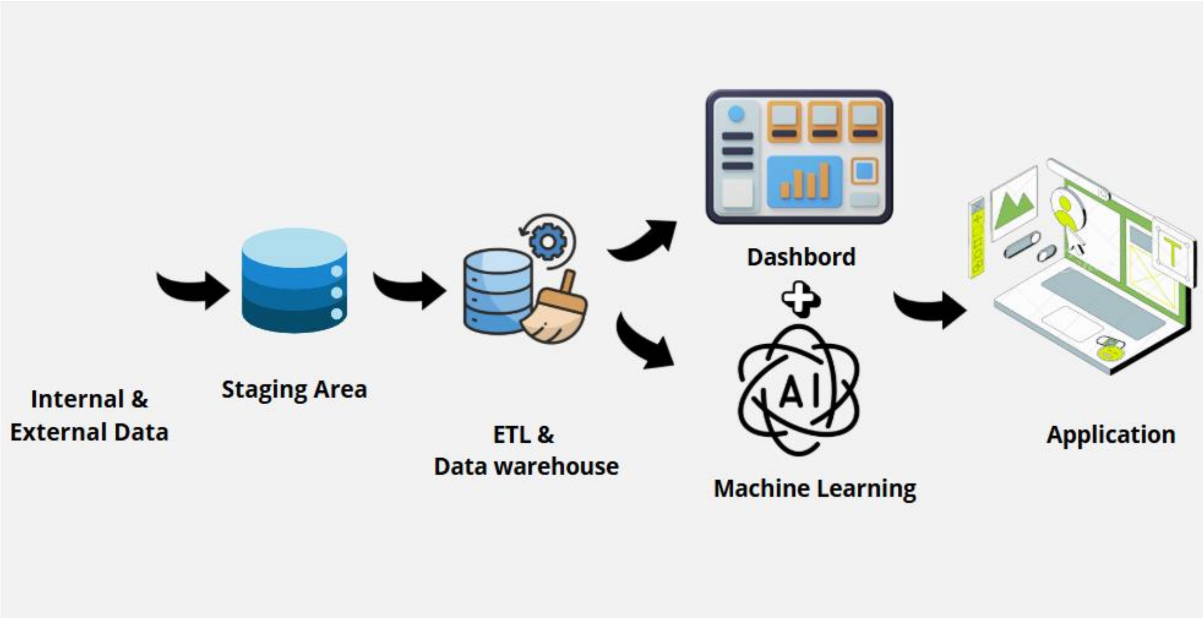


Figure 3 – System Architecture

Conclusion

This chapter provided a comprehensive overview of the current landscape surrounding price comparison tools, highlighting both existing market solutions and their limitations. We examined major competitors, evaluated their strengths and weaknesses, and identified market trends influencing consumer behaviour and business decisions.

Through this analysis, we positioned our solution—**DealDynamo**—as a value-added platform combining Business Intelligence and Machine Learning to empower both customers and sales managers.

Chapter 2: Requirements Analysis

1 Introduction

This chapter describes the functional and non-functional requirements of the project, presents the technologies used, and outlines the implementation procedures. These procedures include the data importation, integration, cleaning processes, and the development of interactive dashboards and application components.

2.1 Working Environment

2.1.1 Software Stack

Several tools and technologies were used to develop this price comparison platform:

- **SQL Server:**
  The data warehouse was implemented using Microsoft SQL Server. It provides a reliable and efficient environment for storing structured and cleaned data and supports complex queries.

Figure4-SQL Server Management studio (MSSQL)

- **Talend**

  An open-source ETL (Extract, Transform, Load) platform used to automate the data integration processes. Talend helped in retrieving data from various sources, transforming it, and loading it into the data warehouse.

Figure5 -Talend Logo

- **Power BI:**

  A powerful data analytics tool developed by Microsoft. It was used to design interactive dashboards and generate insightful reports based on the structured data.

Figure6-PowerBI Logo

- **Flask:**

  A lightweight Python web framework used for developing backend APIs that can serve real-time data or predictions to the frontend application

Figure7-Flask Logo

- **Angular:**

  A front-end web framework used to build the user interface of the application. It allows users to interact with price data, search for products, and view detailed comparisons.

Figure8-Angular Logo

- **Python :**

  Python was also used for developing machine learning models. These models help in predicting price trends and generating alerts based on historical data

Figure9-Python Logo

2.1.2 Development tools

To ensure efficient development and collaboration, we relied on a suite of professional-grade tools:

- **Visual Studio Code (VS Code)**:
  A lightweight, powerful code editor used for writing and managing Angular (frontend) and Python (backend) code

Figure 10- VS Code Logo

- **Git/GitHub**:
  A platform for version control and team collaboration, used to manage code repositories and track changes throughout the project.

Figure11-Git/Gihub Logo

2.2 Functional Requirements:

This section outlines the key functions the system must provide for both **clients** and **sales managers**. These features aim to support informed decision-making through real-time comparisons, predictive analytics, and intelligent recommendations.

For Clients:

- **Real-Time Price Comparison**
  - Users can compare the prices of products, flights, and hotels across multiple vendors in one interface.

- **Historical Price Tracking**
  - The system displays the evolution of prices over time to help users understand market trends.
- **Price Alerts and Notifications**
  - Users receive alerts when prices drop or match a predefined threshold.
- **Smart Recommendations**
  - The system suggests alternative or similar products based on user preferences, history, and pricing patterns.
- **Search and Filter Options**
  - Users can filter offers by category, vendor, price range, location, or rating.
- **Interactive Dashboards**
  - Users can visualize pricing data in an easy-to-understand format using charts and filter

For Companies (Sales Managers):

- **Competitor Price Monitoring**
  - Sales managers can track competitor prices to remain competitive in the market.
- **Sales and Pricing KPIs**
  - The system provides metrics such as average price, price fluctuation, and vendor ranking.
- **Predictive Price Analysis**
  - Machine learning models forecast future price trends to support pricing strategies.
- **Vendor Performance Insights**
  - Dashboards show which vendors are most competitive and which categories have the highest price variation.

2.4. Non-Functional Requirements:

This section describes the non-functional requirements that ensure the system's usability, reliability, performance, and future scalability. These requirements are essential for delivering a robust and professional solution to both clients and business stakeholders.

- **Performance**
  - **Fast Dashboard Loading**: Visualizations and reports must load quickly, even with large datasets.
  - **Low Latency API Responses**: The Flask backend must handle user requests with minimal delay.

- **Accuracy and Reliability**
  - **Accurate Predictions**: Machine learning algorithms should produce reliable price forecasts based on historical data.

- **Scalability**
  - **Modular Design**: The architecture must support easy addition of new data sources (vendors, product types).
  - **Flexible Schema**: The data warehouse (constellation model) must allow future expansion without major redesign.

- **Usability**
  - **User-Friendly Interface**: The Angular-based frontend must be intuitive and responsive, suitable for both technical and non-technical users.
  - **Customizable Dashboards**: Users should be able to filter and interact with visuals easily in Power BI.

- **Security**
  - **Secure Data Handling**: Sensitive data must be stored and transferred securely, following best practices in backend development.
  - **Role-Based Access Control**: including user roles to limit access to certain features or data.

- ***Availability and Maintainability***

  - **Reliable System Uptime**: The system should be stable and available during expected hours of operation.

Conclusion

This chapter defined the functional and non-functional requirements of our project, reflecting the needs of both end users (clients) and business users (sales managers). We detailed how the platform should behave, what features it must include, and the performance and usability standards it must meet.

We also presented the working environment, including the software architecture and development tools used throughout the project. These requirements and tools collectively ensure that our system is not only technically sound but also aligned with user expectations and scalable for future improvements.

Chapter 3: Project Development Process

This chapter outlines the complete development of our integrated project, from the initial design phases to the final deployment and visualization of results. The process was structured and iterative, combinin**3.1. Introduction**

g best practices in data engineering, backend and frontend development, machine learning integration, and user interface design.

Each step played a crucial role in shaping a reliable and user-friendly solution. The chapter begins with the database design and modeling approach, followed by the development of backend and frontend components. It then explores the integration and testing phase and concludes with the creation of the dashboard interface that allows users to explore insights and make informed decisions.

Our goal throughout this process was to deliver a scalable, responsive, and intelligent platform capable of processing and visualizing travel and retail data effectively.

3.2 Staging Area

3.2.1 Data Sources

After collecting both **internal data** (provided by our institution) and **external data** (scrapped from travel agency websites), we entered the **staging phase**, which represents the **Extract and Load** stages of the ETL process.

3.2.2 EL with Talend

To automate and manage this process, we used **Talend Open Studio**, a visual ETL tool that allows the creation of Jobs to extract, transform, and load data.

In Talend, a **Job** is a graphical workflow composed of components, each performing a specific function (e.g., reading, filtering, transforming, or writing data). Jobs are the backbone of Talend's data orchestration.

3.2.3 Loading External Travel Data

The figure below shows an example of a Talend Job we developed to load external data collected from the **Traveltodo** website.

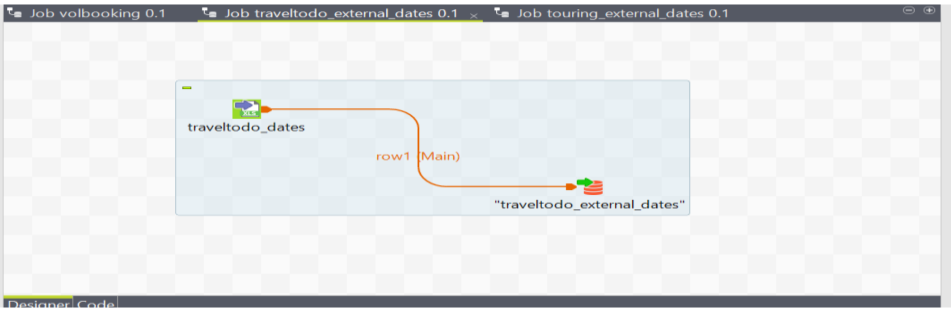


Figure12: Extract & Load of External Traveltodo Data (Dates)

<u>**Description of the Job**</u>:
This Job consists of the following components:

- `tFileInputExcel` (named `traveltodo_dates`) – This component reads raw data from an Excel file containing the scraped travel data.
- `tMSSqlOutput` – This component is used to load the data into the **MSSQL** table named `"traveltodo_external_dates"`.

These components are connected via a **row (Main)** link, which defines the data flow between the input and the output.

Thanks to this setup, the external data is extracted from the file system and loaded directly into our database, forming the **first layer of raw data storage**

3.2.4 Loading Carrefour Files

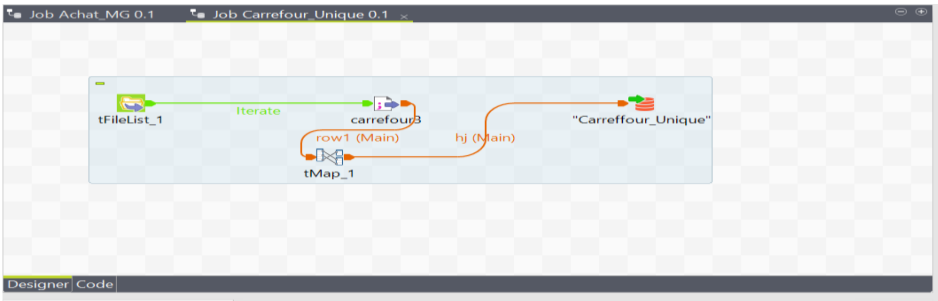


Figure13: Extract & Load of External Carrefour data

<u>**Description of the Job**</u>:

The following figure illustrates a **Talend Job** designed to load Excel files containing **product data from the Carrefour retailer**. These files come from **different product categories** (e.g., food, electronics, clothing) and are stored in the same directory.

Objective:
Automate the reading of **multiple Carrefour product files** and consolidate them into a **single unified space** within our database.

Job Components Description:

- `tFileList_1`:
  This component scans and **iterates over all Excel files** present in a specified folder. Since each file represents a **different product category**, we implemented a **loop (Iterate)** to automatically process all files without manual intervention.
- `carrefour5` **(**`tFileInputExcel`**)**:
  During each iteration, this component reads one Excel file, extracts its content, and sends the data to the next component.
- `tMap_1`:
  This component performs the **mapping of the input columns** to the target schema. It can also include **data cleaning and transformation rules** if needed.
- `tMSSqlOutput`:
  The transformed data is **loaded into the** `Carrefour_Unique` **table** in our **MSSQL database**. This table consolidates all Carrefour product information, regardless of the original category.

3.2.5 Loading Traveltodo Files:

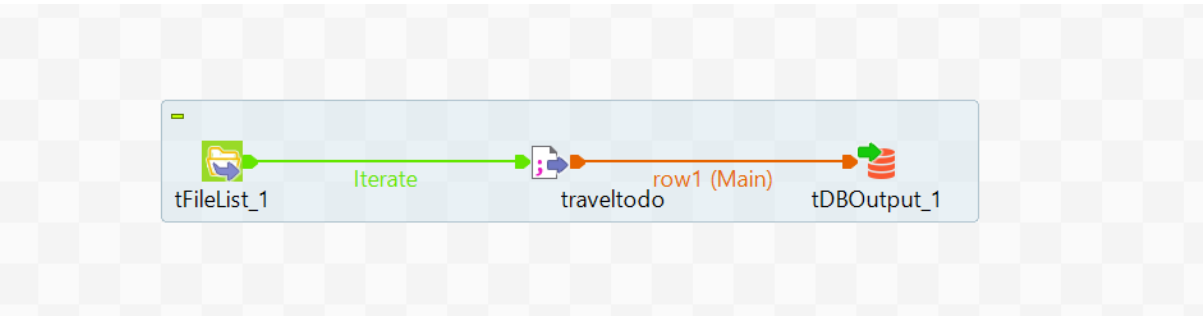


Figure14: Extract & Load of External Traveltodo data(files)

This job automates the **loading of multiple data files** from the travel agency **Traveltodo**, which were obtained via web scraping or internal collection.

Objective:
To read and load **multiple Excel files** containing travel offers (e.g., hotel bookings, flight deals) from Traveltodo into a **centralized database table**.

Job Components Description:

- **tFileList_1**:
  This component is responsible for **listing all the files** in a specified directory. Since we have **several files**, each containing different offers or datasets, we use the **Iterate** feature to process them **one by one**.
- **traveltodo (tFileInputExcel)**:
  For each file iterated by tFileList_1, this component reads the **Excel content** and sends the data as rows to the next component in the flow.

- **tDBOutput_1**:
  This component loads the extracted data into a **Microsoft SQL Server (MSSQL)** table. This table serves as a **staging table** for all Traveltodo-related data.

3.2.6 Summary of EL Jobs in Talend:

The screenshot below provides an overview of the different **EL jobs** created using **Talend Open Studio**. Each job corresponds to a specific task within the staging or preprocessing phases of our data pipeline.

Purpose of this summary:
To showcase how we **modularized** our TL process by building **separate jobs** for each partner, data source, or transformation requirement.

Examples of the jobs listed:

- Achat_MG, MG, Geant_Unique, Carrefour_Unique:
  Jobs dedicated to loading product data from the MG, Géant, and Carrefour retailers.
- touring_external_dates, traveltodo_external_dates:
  Jobs that handle loading external travel data scraped from websites like **Touring** and **Traveltodo**.
- children_products, hotel_rating:
  Specific jobs for categorizing or enhancing certain datasets, like children's products or hotel reviews.
- alimenterproduit:
  A job responsible for feeding the global product catalog with cleaned and unified data.

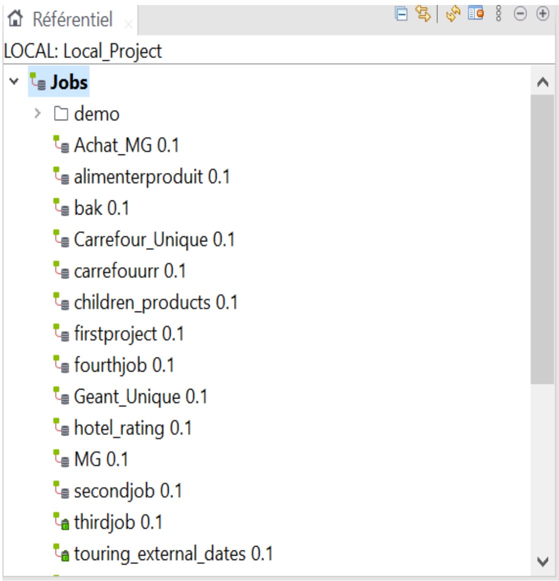


Figure15: Summary of all jobs in talend

This modular approach allowed us to maintain a **clear and scalable architecture**, making it easy to update, test, or rerun only the needed part of the process when data changes.

3.2.7 Job Automation and Scalability

Each data source (internal or external) has its own Talend Job, which can be scheduled and reused. This modular approach makes the data integration process **scalable**, **maintainable**, and **easy to troubleshoot**. It also ensures that every dataset is correctly formatted and safely stored in our staging environment before entering the transformation phase.

3.3. Data Warehouse

To support the analytical needs of our **multi-sector price comparison platform**, we designed a **Data Warehouse** using a **Constellation Schema** (also known as a *Galaxy Schema*). A **Data Warehouse (DW)** is specifically built to support business decision-making by enabling **data consolidation, analysis, and reporting** at various levels of aggregation. Through **Extraction, Transformation, and Loading (ETL)** processes, data from multiple sources is integrated into the DW to ensure consistency and reliability.

This architecture is particularly well suited for our project, which involves comparing data across **retail**, **air travel**, and **tourism** sectors. The constellation schema enhances this flexibility by organizing multiple fact tables that share common dimensions. Thanks to this structure and the underlying technologies, decision-makers can quickly and effectively analyze price trends, consumer behavior, and performance across all domains

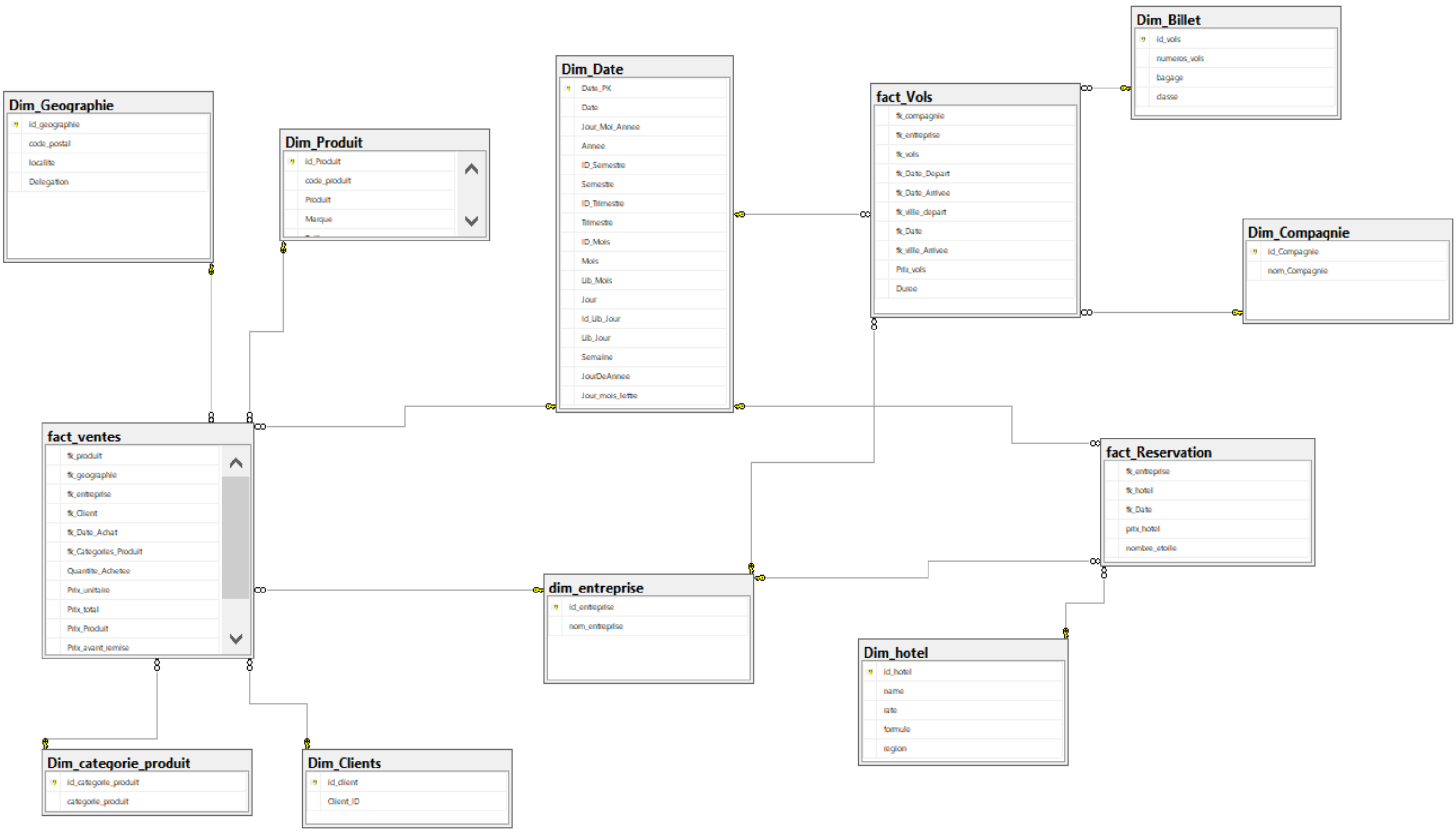


Figure 16:Data warehouse Diagram

3.4 Alimentation of dimensions and facts:

A fact table works with dimension tables. A fact table holds the data to be analyzed, and a dimension table stores data about the ways in which the data in the fact table can be analyzed. Thus, the fact table consists of two types of columns. The foreign keys column allows joins with dimension tables, and the measures columns contain the data that is being analyzed.

3.4.1.Product Dimension (Dimension Produits)

Description:
Stores product-related data such as name, brand, category, and unit price, used for comparing items across retailers.

ETL Process:
Product data from MG, Géant, and Carrefour is merged using tUnite, transformed via tMap, and loaded into Dim_Produit. This ensures consistent product info across sources.

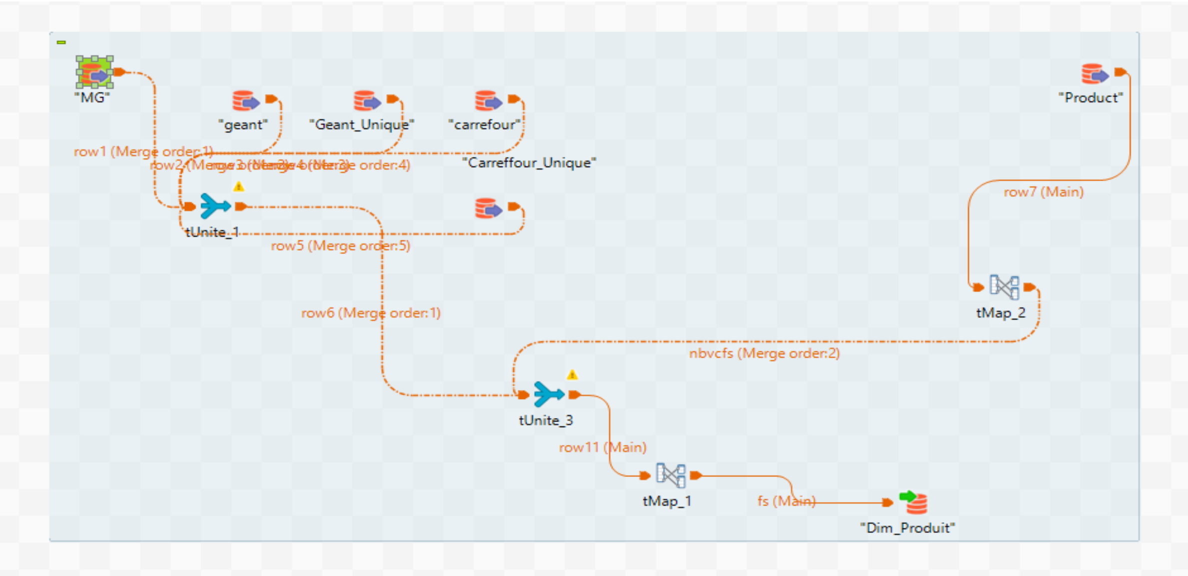


Figure 17:Alimentaion Product Dimension

3.4.2.Store Dimension(Dimension entreprise)

Description: Contains metadata about the stores (name, id).

Purpose: Used to filter and compare prices or sales by store location.

Data Source: Gathered from retail and tourism partners

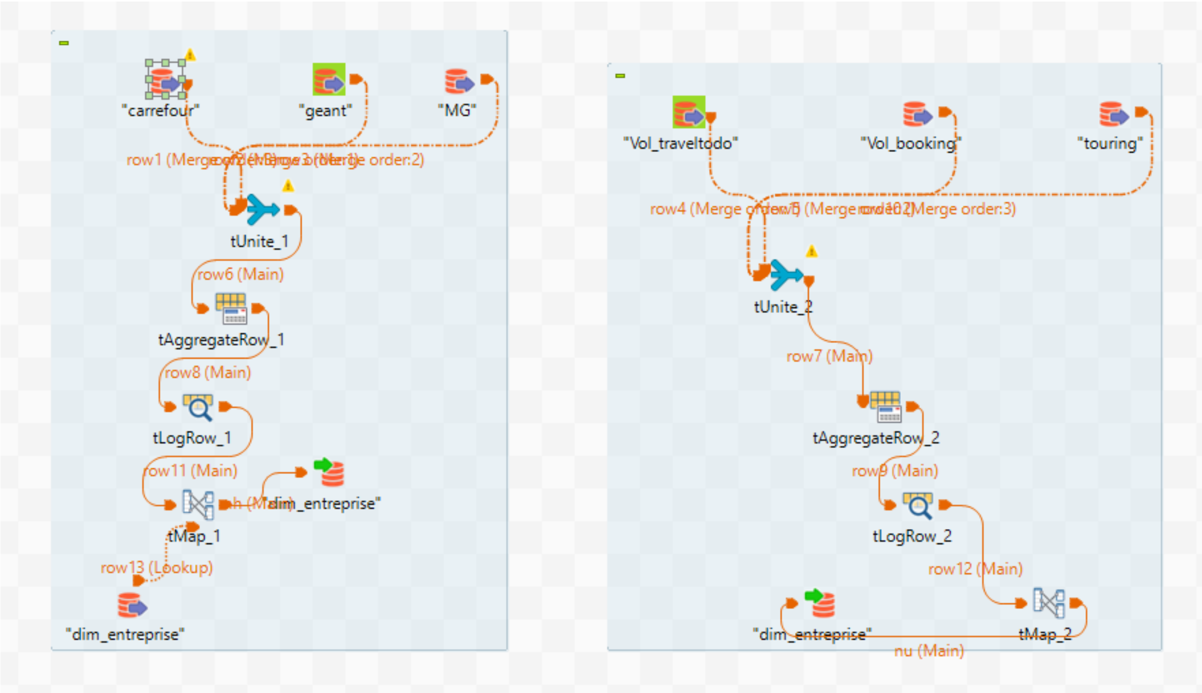


Figure 18: Alimentaion of store Dimension

3.4.3Dimension Date : (Date Dimension)

- Description: Covers all calendar-related attributes (day, month, year, quarter, etc.).
- Purpose: Allows time-based analysis (e.g., monthly sales trends, seasonal travel peaks).
- Data Source: Auto-generated using calendar scripts during ETL.

3.4.4 Customer Dimension : (Dimension Client)

- **Description**: Includes customer information like name, age, gender, and membership type.
- **Purpose**: Enables analysis of purchasing or booking behavior by demographic segments.

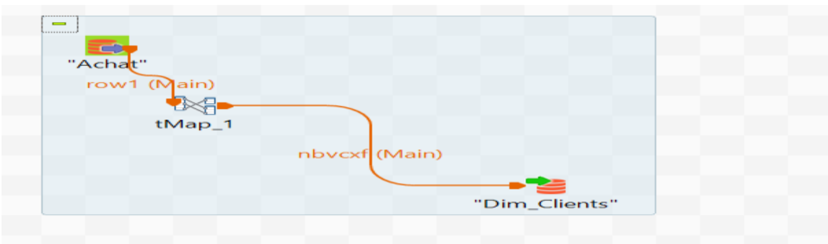


Figure 19: Alimentaion of Clients Dimension

3.4.5Geography Dimension (Dimension géographique):

Description:
Contains geographical information such as postal code, locality, and delegation. It helps analyze customer behavior and sales trends by region.

ETL Process:
Geographic data is extracted from source files, transformed to standardize location names and codes, then loaded into the `Dim_Geographie` table

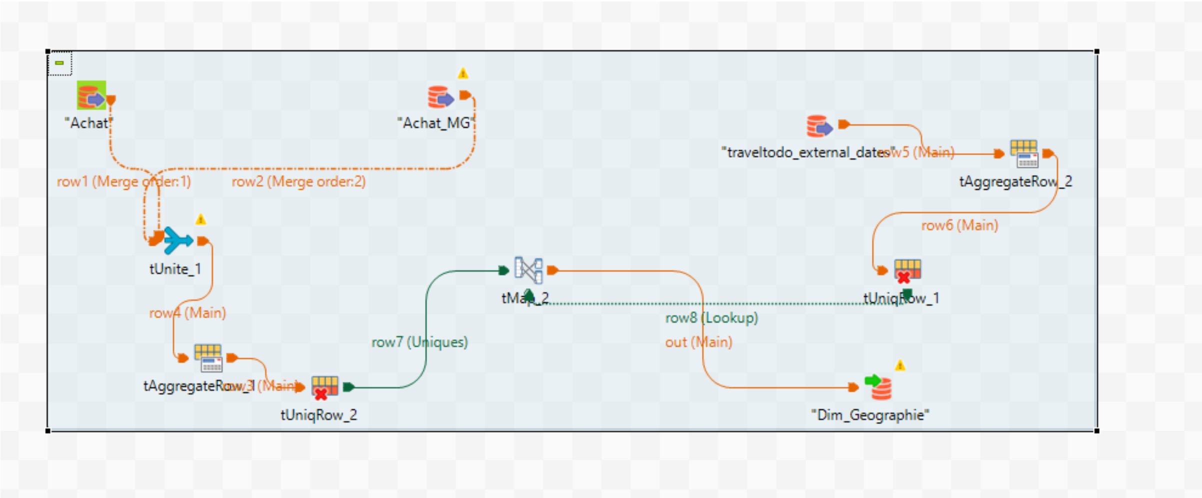


Figure 20: Alimentaion of Geography Dimension

3.4.6 Airline Dimension : (Dimension Compagnie_Aerienne)

- **Description**: Lists airline details including name, code, and type of service.
- **Purpose**: Helps analyze flight bookings by airline.

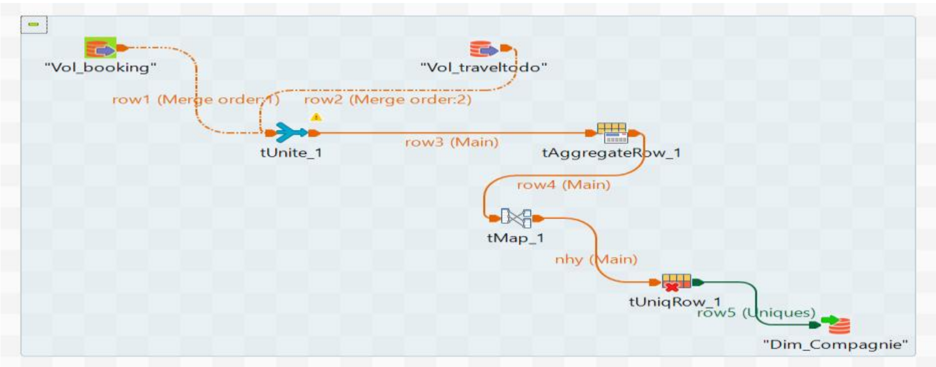


Figure 21: Alimentaion of Geography Dimension

3.4.7 Hotel Dimension

Description:
Stores hotel-related data such as name, location, and star rating. It supports analysis of tourism offers and accommodation comparisons.

ETL Process:
Hotel data is collected from both **TravelTodo** and **Touring**, merged, enriched with star ratings, and loaded into the Dim_Hotel table.

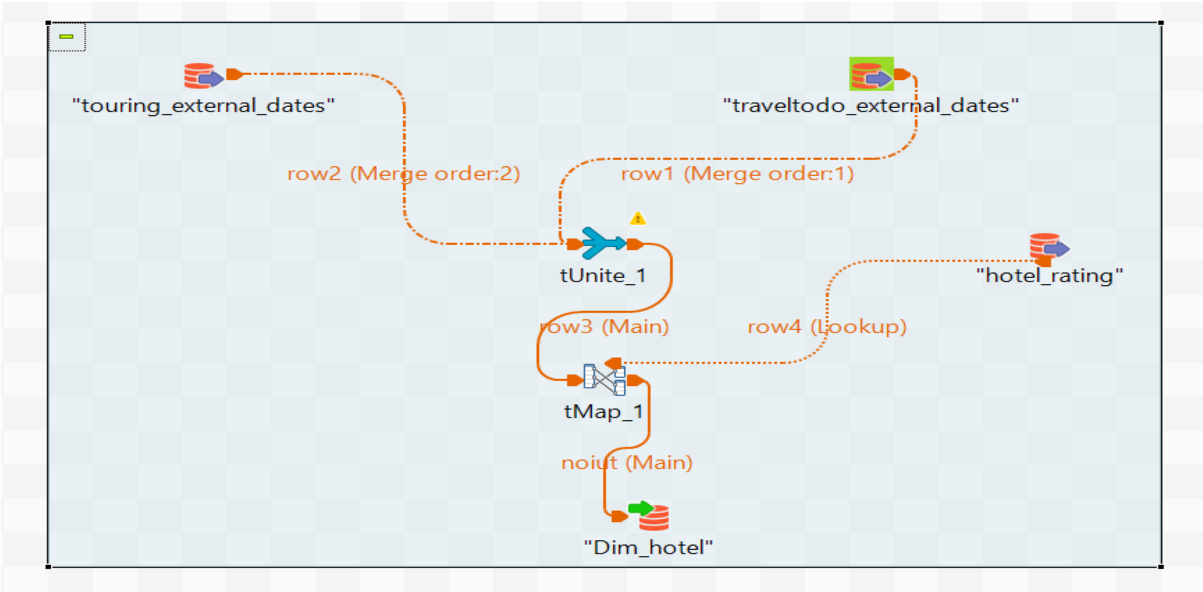


Figure 22: Alimentaion of Hotel Dimension

3.4.8 Product Category Dimension

Description:
Contains product category details like category name and type. It enables grouping and filtering of products for retail price comparison.

ETL Process: Category data is extracted from product sources, standardized, and loaded into the Dim_Categorie_Produit table.

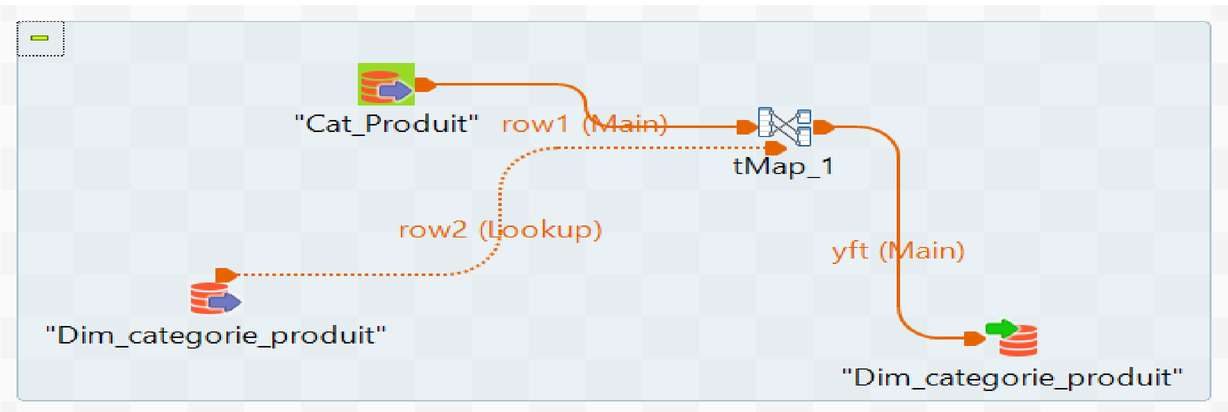


Figure 23: Alimentaion of Category Dimension

3.4.9 fact_Reservation:

This ETL flow loads the fact_Reservation fact table by merging reservation data from **TravelTodo** and **Touring**. Using Talend components (tUnite, tMap), the data is cleaned, unified, and enriched with lookup dimensions such as **hotels** and **agencies** (dim_hotel, dim_entreprise). The result is a structured fact table ready for price comparison and business analysis.

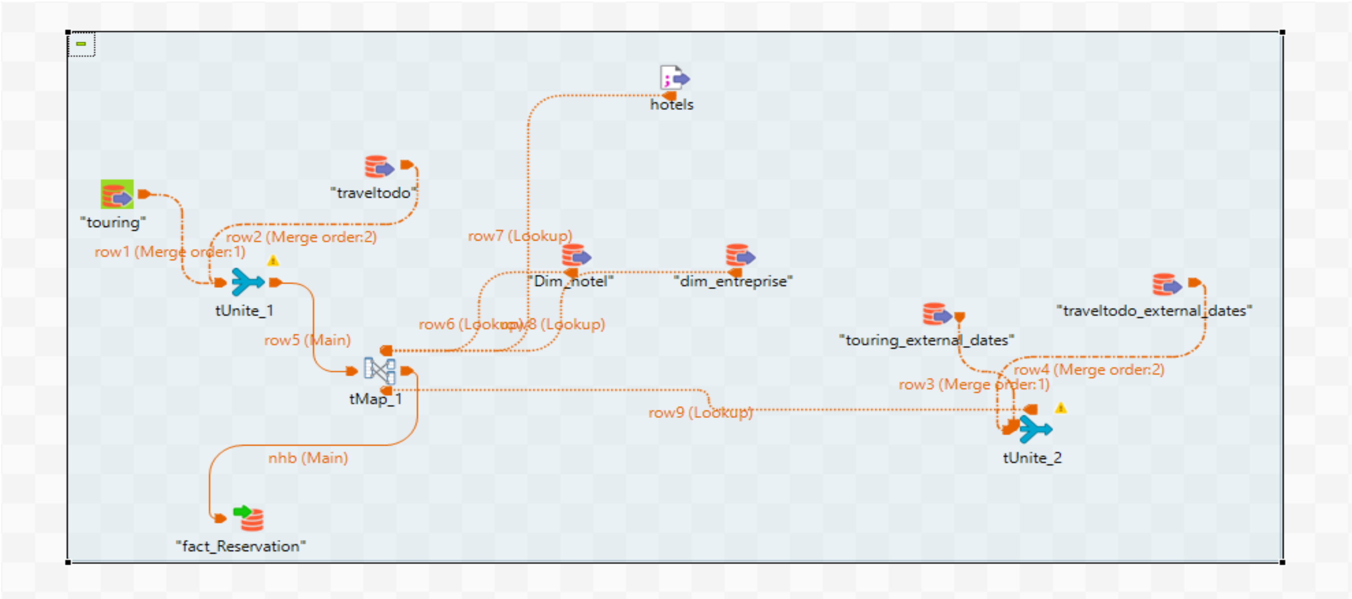


Figure 24: Alimentaion of Fact_Reservation

3.4.10 Fact_Ventes :

This ETL process populates the fact_ventes fact table by merging purchase data from Carrefour, Géant, and MG. The unified data is enriched through lookups on dimensions such as clients, products, geography, categories, and companies. The fact table stores key metrics including unit prices, quantities purchased, and total sales, enabling detailed price analysis and comparison.

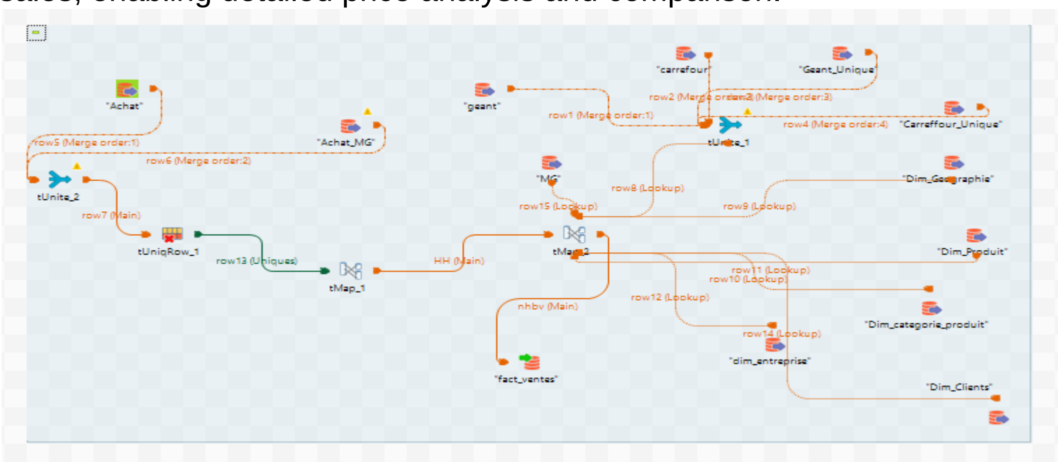


Figure 25: Alimentaion of Fact_Ventes

3.4.11 Fact_Vols:

This ETL flow loads the fact_Vols fact table by merging flight booking data from Vol_booking and Traveltodo sources. Lookup operations enrich the dataset using dimensions such as companies, enterprises, and ticket types.
The fact table captures key metrics such as ticket prices, booking dates, and associated enterprises, enabling detailed analysis of flight trends and offers.

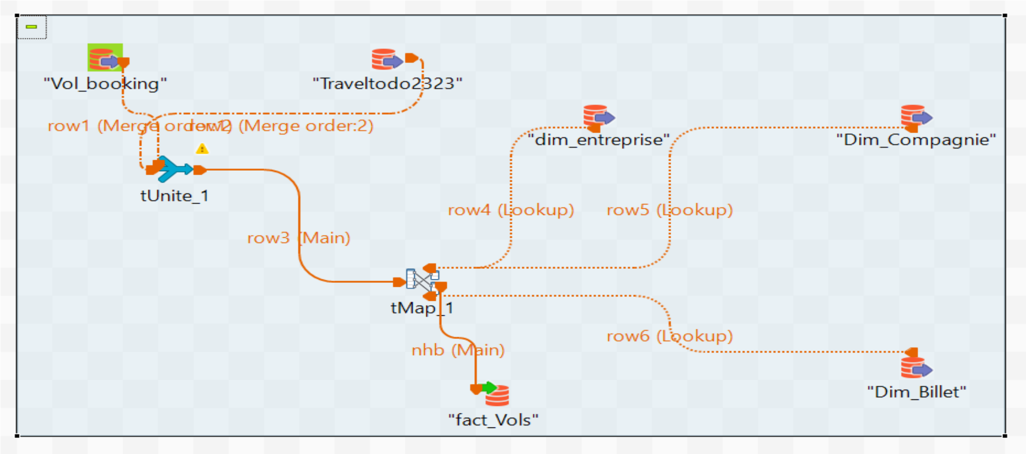


Figure 26: Alimentaion of Fact_Vols

3.5 Storage in MSSQL

After completing the ETL process and designing our data warehouse model, all cleaned and integrated data was finally loaded into **Microsoft SQL Server (MSSQL)**, which served as our centralized and structured storage system.

Thanks to the powerful integration between **Talend** and MSSQL, we ensured that:

- All **fact tables** (e.g., `Fact_Sales`, `Fact_Reservations`, `Fact_Flights`) were correctly populated with accurate, up-to-date data.
- All **dimension tables** (e.g., `Dim_Product`, `Dim_Date`, `Dim_Customer`, `Dim_Airline`, `Dim_Location`) were well normalized and fully filled, ensuring referential integrity and consistency across the data warehouse.

This final storage step allowed us to:

- Securely manage large volumes of historical and real-time data.
- Enable fast querying, aggregation, and reporting for analytical and decision-making purposes.
- Seamlessly connect the database to **BI tools** like Power BI for data visualization and insights.

In conclusion, MSSQL provided a solid foundation for our entire architecture, ensuring reliable storage and access to all components of our **constellation schema**.

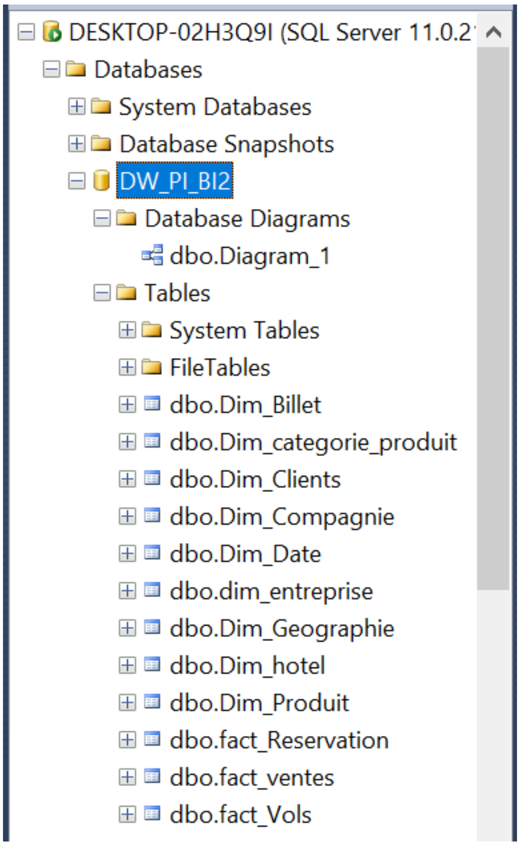


Figure 27: Storage of the data in MSSQL

Chapter 4: Platform Development and Integration

4.1 Machine Learning Models

4.1.1 Predictive Objectives

In our project, we developed multiple predictive models to serve two main user groups: **sales managers** and **consumers**. Each group has distinct needs, and our models were tailored accordingly.

- **For sales managers**, the key predictive objective was to **forecast sales revenue (chiffre d'affaires)** across major retail companies, including **Carrefour** and **MGG**. We aimed to provide accurate and timely forecasts using historical sales data, enabling data-driven business decisions such as stock management, pricing strategies, and demand anticipation.
- **For consumers**, several predictive features were implemented to enhance the shopping and travel experience:
    - **Price Drop Prediction**: The model estimates the probability and percentage of future price drops for products, allowing customers to buy at the optimal time.
    - **Flight Price Prediction**: Users can predict the expected cost of a flight based on origin, destination, date, class, and baggage details.
    - **Nearest Store Prediction**: By using geolocation inputs, the system recommends the **closest store** (e.g., Carrefour in Marseille vs. Carrefour in Ariana).
    - **Product Recommendation**: The system suggests the **best products** from multiple stores based on past trends and consumer preferences.
    - **Hotel Segmentation**: Hotels are segmented based on **review scores** and **geographic regions**, helping users choose top-rated accommodations.

Each model was carefully selected, trained, and evaluated based on the nature of the prediction task, using algorithms such as KNN, Random Forest, and Gradient Boosting. The deployment of these models provides real-time, intelligent support for both decision-makers and end users.

4.1.2 Data Used and Preprocessing

The source of data used to train all our machine learning models is the **Data Warehouse** we previously built and populated during the ETL process. It contains cleaned, structured, and integrated data from both internal sources and scraped external sources. This centralized and well-organized storage allowed us to extract the relevant features needed for our predictive tasks.

We performed our modeling and experimentation using **Python on Google Colab**, which provided a flexible and collaborative environment for data analysis, model training, and evaluation.

4.2 Model Evaluation and Results

4.2.1 Sales Prediction

Our first predictive objective was to forecast the **sales prediction** for three key retail companies: **MG**, **Carrefour**, and **Géant**.

To achieve this, we experimented with several machine learning algorithms, including:

- **K-Nearest Neighbors (KNN)**
- **Support Vector Machine (SVM)**
- **XGBoost**
- **Gradient Boosting**

For each company, we trained and evaluated the models separately using historical sales data extracted from the Data Warehouse. After thorough evaluation using standard metrics such as **R^2 score**, **RMSE**, and **MAE**, we found that **Gradient Boosting** consistently outperformed the other models across all three companies, with an R^2 score reaching **0.9**.

Given its superior performance and robustness, **Gradient Boosting was selected for deployment** as the predictive engine for sales forecasting in our system.

- **Models Evaluation for Sales Prediction**

```
KNN - R² Score: 0.2006
Random Forest - R² Score: 0.9905
Gradient Boosting - R² Score: 0.9909

✅ Best Model: GradientBoostingRegressor with R² Score: 0.9909
```

Figure28 –Evaluation of the model

- **Classification report**

```
🔷 MAE: 7039.92
🔷 MSE: 75484699.39
🔷 RMSE: 8688.19
🔷 R² Score: 0.9909
```

Figure29-classification

4.2.2 Forecasting of the future demand for product categories

We also used **Sarima**, a dedicated forecasting module, to **predict future demand for product categories**. This helped us anticipate which categories would generate the highest sales and allowed for better stock management and marketing decisions.

We applied time series forecasting on each major product category. For instance:

- **Category 1 Forecasting**
  - **Mean Absolute Error (MAE):** 85.33
  - **Root Mean Squared Error (RMSE):** 113.05
- **Category 2 Forecasting**
  - **Mean Absolute Error (MAE):** 84.67
  - **Root Mean Squared Error (RMSE):** 112.08

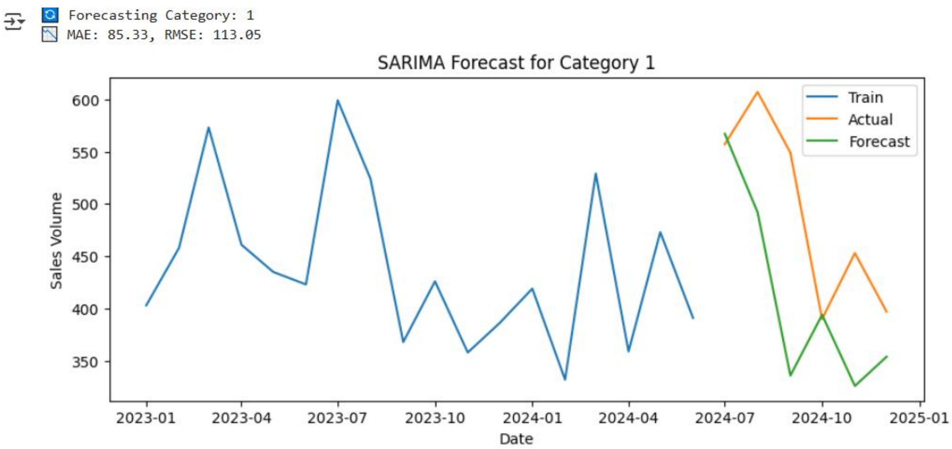


Figure 30: SARIMA Model- Most Purchased Categories by Sales Volume

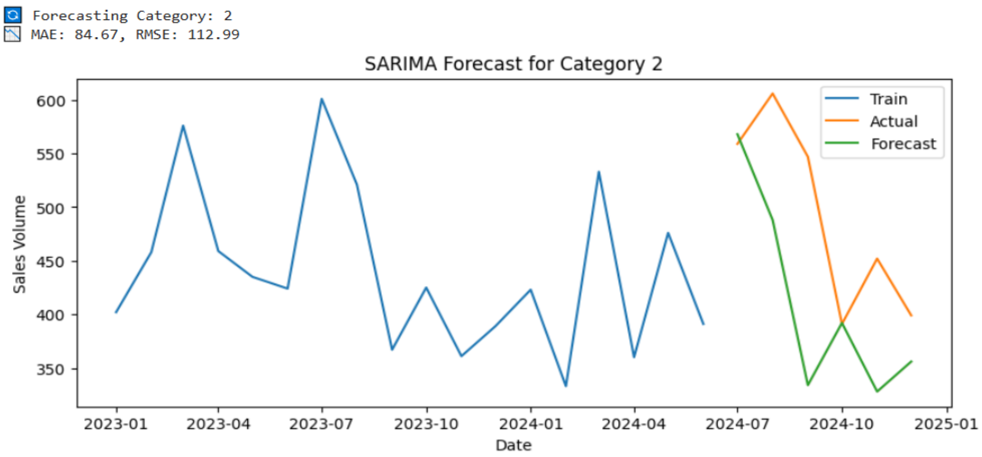


Figure 31: SARIMA Model- Most Purchased Categories by Sales Volume

The results demonstrate that our model could reasonably estimate future demand trends per category. These forecasts were visualized and validated through graphical dashboards, ensuring interpretability for decision-makers.

4.2.3 Price Drop Prediction

The second prediction objective was designed for **end-users or consumers**, aiming to estimate the **price drop percentage** of a given product over time. This feature allows users to input several parameters—such as the **product name**, **brand**, **size**, **quantity**, and **desired purchase date** (year, month, day)—and receive an estimated percentage of price reduction, helping them decide the best time to buy.

To accomplish this, we framed the problem as a **regression task** and tested multiple algorithms, including:

- **K-Nearest Neighbors (KNN)**
- **XGBoost Regressor**

The evaluation metrics showed the following results:

- **KNN**: $R^2 = 0.34$, RMSE = 5.31
- **XGBoost**: $R^2 = 0.86$, RMSE = 7.16

Although the RMSE for XGBoost was slightly higher, the **R^2 score was significantly better** than that of KNN, indicating a stronger ability to capture the variance in the data.

Based on this performance, we selected **XGBoost** as the final model for deployment in our platform, as it provides **reliable and accurate predictions**, especially given the complexity of pricing behavior in the retail market.

- **Models Evaluation**

```
KNN (avec meilleurs paramètres) - RMSE: 5.31,
KNN - RMSE: 5.31, R² : 0.34
```

```
35 7
XGBoost - RMSE: 7.16, R² : 0.86
```

Figure 32-Model Evaluation

4.2.4 Flight Ticket Price Prediction

The third predictive objective focused on forecasting **flight ticket prices** based on user-specified travel details. This feature is tailored for consumers who want to estimate the cost of a flight before booking, enabling better planning and comparison across airlines.

The input features for this model included:

- **Departure Date**
- **Arrival Date**
- **Departure City**
- **Arrival City**
- **Travel Class**
- **Baggage Option (included in ticket type)**

The model was expected to output:

- The **estimated ticket price**
- The **most likely airline** for the given travel scenario

We experimented with the following regression algorithms:

- **XGBoost Regressor**
- **Random Forest Regressor**

Among the two, **Random Forest Regressor** provided the best performance with:

- **R^2 score**: 0.80
- **Root Mean Square Error (RMSE)**: 375

Given its high accuracy and generalization capability, **Random Forest** was selected for deployment to handle flight ticket price prediction on our platform.

- **Models Evaluation**

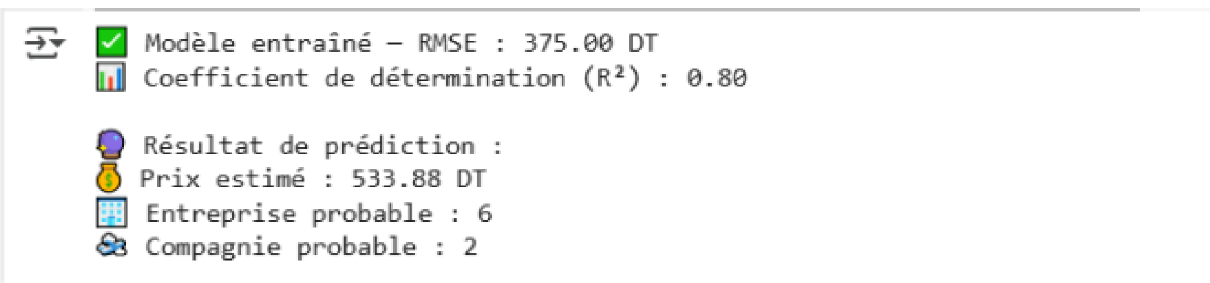


Figure 33-model evaluation

4.2.5 Predicting Distance from the Nearest Store

Another consumer-focused predictive objective was to estimate the **distance between a user's location and the nearest store** from a chosen retailer (MG, Carrefour, or Géant). This helps consumers determine which store location is most accessible to them.

The input features included:

- **Retailer name** (MG, Carrefour, Géant)
- **Delegation**
- **Locality**
- **Postal code**

The output of the model was:

- **Predicted distance (in kilometers)** to the nearest physical store of the selected brand.

We tested several regression models:

- **K-Nearest Neighbors (KNN)**: $R^2 = 1.00$
- **Random Forest Regressor**: $R^2 = 0.90$
- **Gradient Boosting Regressor**: $R^2 = 0.90$

Although KNN achieved a perfect R^2 score, it tends to overfit and lacks generalization for unseen data. Therefore, **Gradient Boosting Regressor** was chosen for deployment due to its strong predictive performance and robustness.

- **Models Evaluation and Classification Report**

```
📊 Résultats des modèles :

🔷 KNN :
R² Score: 1.0000
MAE: 0.00
RMSE: 0.00

🔷 Random Forest :
R² Score: 0.9415
MAE: 20.37
RMSE: 26.03

🔷 Gradient Boosting :
R² Score: 0.9966
MAE: 5.79
RMSE: 6.29

✅ Modèle sauvegardé (hors KNN) : Gradient Boosting avec R² = 0.9966
```

Figure34-model evaluation

4.3 Backend Development

For the backend development, we used **Flask**, a lightweight Python web framework, to deploy all the machine learning models as RESTful APIs. Each predictive model selected for deployment was first saved using **Joblib**, allowing us to serialize the trained models and load them easily for inference.

We created a dedicated Flask route for each prediction use case (e.g., sales forecasting, price drop prediction, flight ticket estimation, nearest store distance, etc.). Each route is associated with a specific **HTML form** where users can input the required parameters. Once the form is submitted, the backend processes the inputs, loads the corresponding model, performs the prediction, and returns the output (e.g., predicted revenue, price, percentage drop, or distance).

To maintain modularity and avoid conflicts, **each predictive service was deployed on a separate port**, ensuring that the APIs remained isolated and easy to test individually. This architecture made the system flexible and scalable for future model integration.

Figure 35: Prediction Input Form in Flask App

4.4 Frontend Development (Angular Integration)

For the frontend, we used **Angular**, a robust and scalable framework for building dynamic web applications. Angular allowed us to design a responsive and user-friendly interface for each machine learning use case. Each prediction module was linked to a dedicated form that collected the necessary inputs from the user (e.g., product details, flight information, location, etc.).

We implemented **HTTP client services** in Angular to send user input from the frontend to the corresponding Flask API endpoints. Once the prediction was processed by the backend, the result was returned in real-time and displayed to the user in a clean and intuitive format.

We also integrated feedback messages, loading indicators, and basic error handling to improve user experience.

Each predictive service was integrated into a separate **component**, allowing for modular development and easy navigation. The design was kept simple and consistent with clear labels, dropdowns, and date pickers to guide the user through the prediction process.

4.5 Integration into the Web Platform

In this final phase, all the components developed during the project were integrated into a unified and functional web platform. The objective was to deliver an intelligent, user-friendly interface that connects data, machine learning models, and dashboards into a seamless experience.

The integration process began by connecting the **Flask APIs**, which expose our trained machine learning models (saved in `.joblib` format), with the **Angular frontend**. Each predictive service—such as revenue forecasting, price drop prediction, flight ticket estimation, store distance calculation, and product recommendation—was deployed on a dedicated Flask route. These APIs were consumed by Angular components using HTTP requests.

The platform was designed to serve two main types of users:

- **Business users** (e.g., sales managers), who can access revenue predictions, dashboards, and analytical insights.
- **Consumers**, who can use features like flight price prediction, store distance finder, and best product recommendations.

All predictive models fetch their input data from the **MSSQL Data Warehouse**, ensuring consistency and reliability across the system. Angular handles the user interface and form submission, while Flask handles the prediction logic and returns results in real time.

Special attention was paid to **API communication, security (CORS policies)**, and performance optimization to ensure smooth interaction between the backend and frontend. The platform is responsive, scalable, and ready for deployment in a real-world environment.

This integration step successfully brought together all parts of the project into a robust, centralized solution accessible to end users.

Conclusion

Throughout this development phase, we successfully implemented and deployed a wide range of machine learning models tailored to different business and consumer needs. From predicting sales revenues to estimating flight prices and recommending best products, each model was trained, evaluated, and deployed using modern data science tools. The backend infrastructure was built using Flask APIs, and the frontend interface was developed with Angular to ensure usability and responsiveness. All components were then integrated into a coherent and scalable web platform. This stage marked the transition from model development to full-scale application, setting the foundation for insightful and interactive data visualization.

Chapter 5: Data Visualization and Business Intelligence

Introduction

In this chapter, we focus on the presentation and interpretation of the data and predictions through interactive dashboards and visualization tools. Using **Power BI**, we developed multiple dashboards that provide strategic insights for both internal stakeholders and end users. These dashboards not only reflect historical and real-time data but also incorporate the results of our predictive models.

We will first explore how sales and flight data were visualized, then highlight the use of key performance indicators (KPIs) and filters. Finally, we discuss how these dashboards were embedded into the web application to allow seamless access and decision-making support.

5.1. Power BI Dashboards

5.1.1.Retail Sector

5.1.1.1 Sales Manager's Dashboard

The sales manager dashboards are divided into three main sections to support strategic and operational decision-making:

1. **Sales Analysis** – Provides insights into overall sales performance, including revenues by category, top-selling products, and monthly trends.
2. **Transaction Details** – Displays detailed records of individual transactions, allowing for tracking of sales volumes, dates, and customer purchases.
3. **Customer Behavior** – Analyzes customer preferences and buying patterns to better understand demand, loyalty, and potential upselling opportunities.

These dashboards are designed to help sales managers monitor key performance metrics, identify trends, and adapt their strategies accordingly.

a) Sales manager Carrefour :

Figure 36:Sales Analysis dashboard

Figure 37: Transactions Details Dashboard

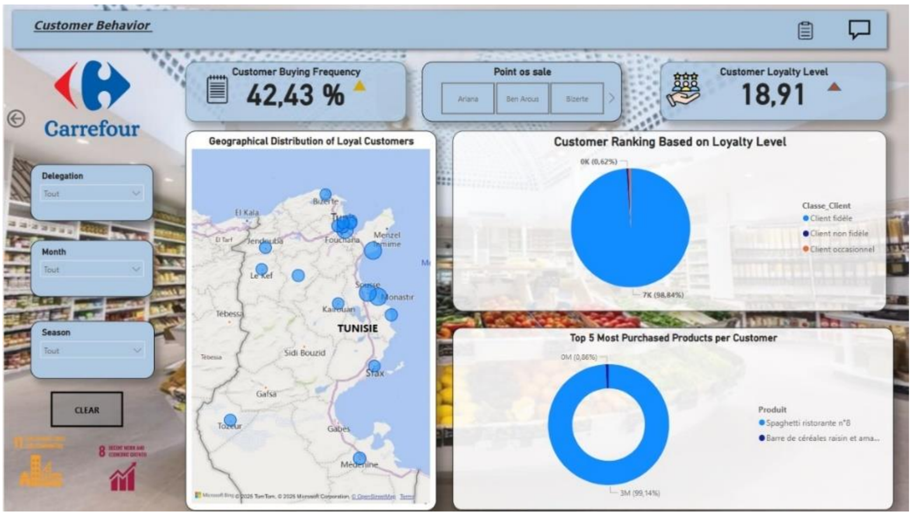


Figure 38: Customer behavior Dashboard

b)Sales manager Géant:

Figure 39: Overview of the Transactions Details

c)Sales manager MG:

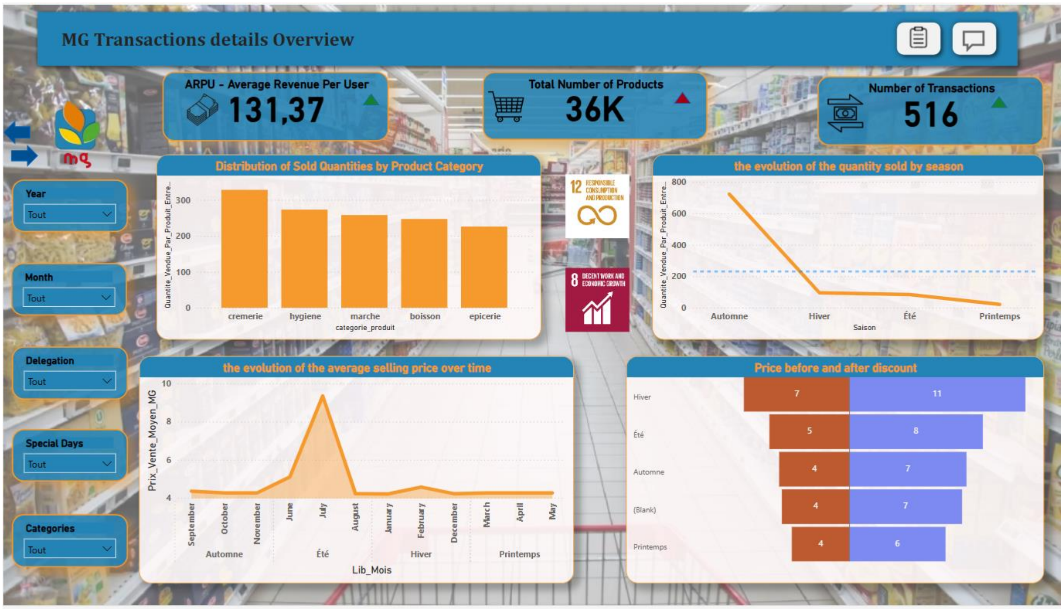


Figure 40: Overview of sales Analysis

5.1.2 Tourism Sector

5.1.2.1. Flights Dashboard

a)Sales manager Booking :

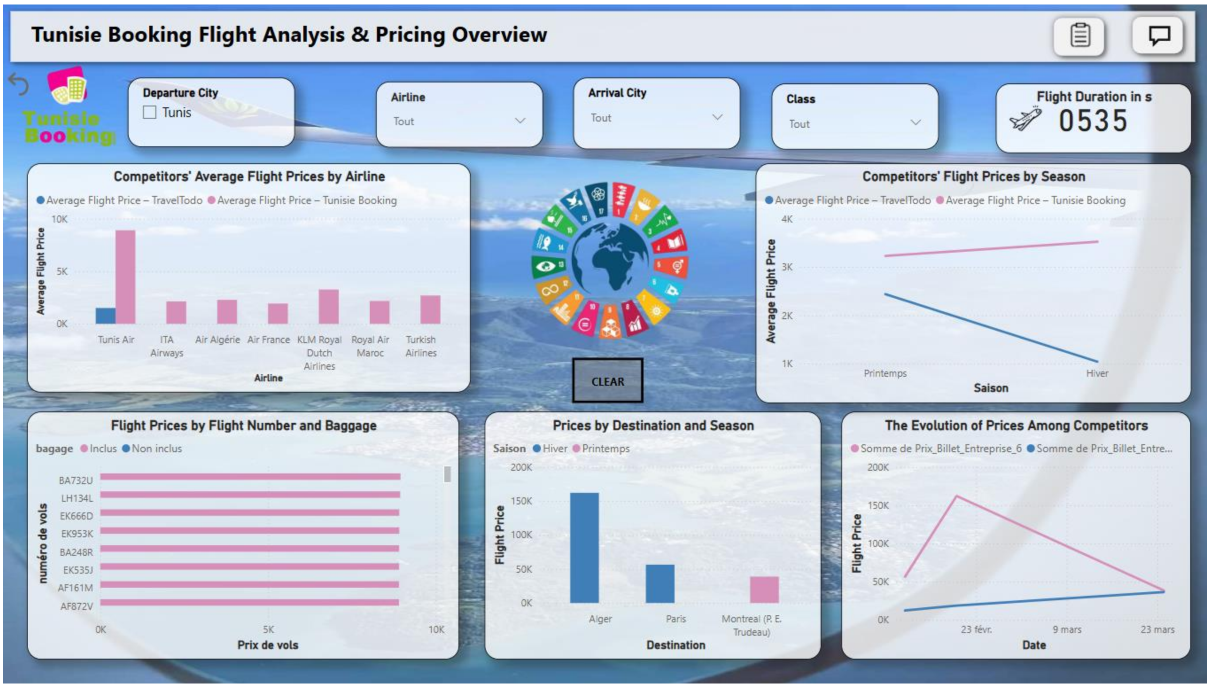


Figure 42: Overview Of Flights Analysis and Pricing

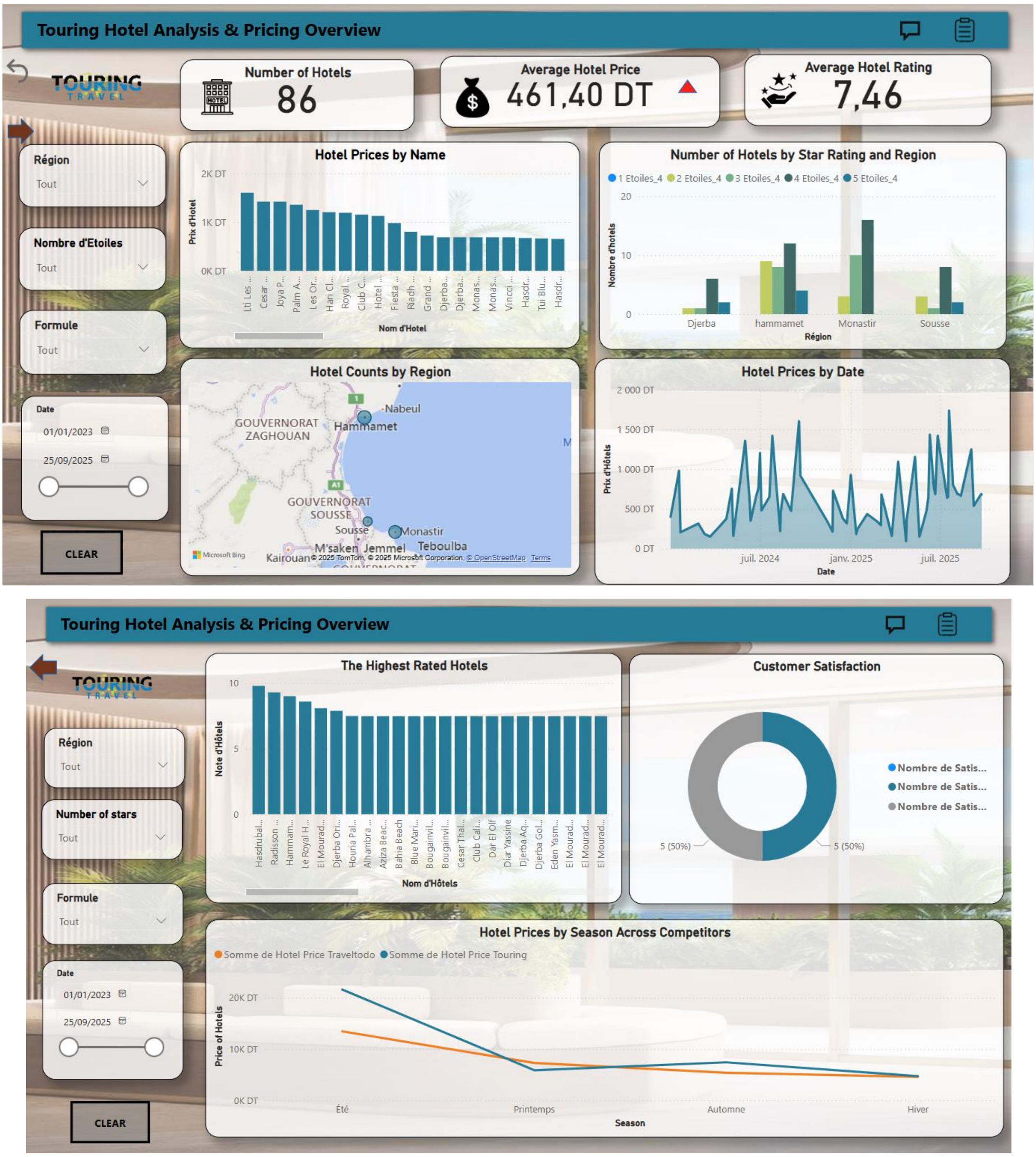


Figure 43- Overview Of Hotels Analysis and Pricing

b)Sales manager Traveltodo :

Figure 44- Overview Of Flights Analysis and Pricing

5.1.1.2 Customer's Dashboard :

Figure45- Overview Customer Dashboards

5.2. KPIs and Filters:

To enhance the usability and clarity of our Power BI dashboards, we integrated several **Key Performance Indicators (KPIs)** and **interactive filters** to help users quickly access relevant insights and make data-driven decisions.

Key Performance Indicators (KPIs)

We selected KPIs based on the most critical business metrics for both sales and flights. These KPIs are displayed clearly at the top of each dashboard and automatically update based on the applied filters. Some of the main KPIs include:

- **Total Sales Revenue**: Displays the total turnover for a selected period and company.
- **Average Price Drop Percentage**: Shows the average decrease in product prices over time.
- **Most Purchased Product Category**: Highlights the best-performing product type.
- **Total Number of Reservations**: Tracks the number of flight bookings for each company.
- **Flight Occupancy Rate**: Indicates how full flights are across selected dates and destinations.

These indicators allow both managers and customers to instantly understand trends and performance without exploring raw data.

Filters and Slicers

To allow dynamic and user-friendly data exploration, we included **interactive filters** and **slicers** throughout the dashboards. These include:

- **Date Range Filters** (month/year selectors)
- **Company Selector** (e.g., MG, Carrefour, Géant, or airlines)
- **Product Category Filter**
- **Destination or City Selector** (for flight dashboards)
- **Store Location Filter** (delegation/locality for commerce dashboards)

With these filters, users can customize the view according to their interests and perform comparative analyses (e.g., comparing sales performance between MG and Carrefour, or between different regions).

Nombre_Achats_Entreprise1	CALCULATE(COUNT(fact_ventes[fk_Client]), fact_ventes[fk_entreprise] = 1)	Compte le nombre total d'achats effectués chez l'entreprise Géant
Nombre_Client geant	CALCULATE(DISTINCTCOUNT(fact_ventes[fk_client]), fact_ventes[fk_entreprise] = 1)	Compte le nombre de clients uniques ayant acheté chez l'entreprise Géant.
Classe_Client	IF(fact_ventes[Nb_Achats_Par_Client] >= 7, "Client fidèle", IF(fact_ventes[Nb_Achats_Par_Client] >= 3, "Client occasionnel", "Client non fidèle"))	Classe les clients selon le nombre d'achats en fidèles, occasionnels ou non fidèles.
Prix_AuDessous_Moyenne_E3	VAR Moyenne = CALCULATE(AVERAGE(fact_ventes[Prix_Produit]) , fact_ventes[fk_entreprise] = 3) RETURN Moyenne * 0.75	Calcule 75 % du prix moyen des produits pour l'entreprise 3.
Prix_AuDessus_Moyenne_E3	VAR Moyenne = CALCULATE(AVERAGE(fact_ventes[Prix_Produit]) , fact_ventes[fk_entreprise] = 3) RETURN Moyenne * 1.25	Calcule 125 % du prix moyen des produits pour l'entreprise 3.
Panier_Moyen	CALCULATE(DIVIDE(SUMX(fact_ventes, fact_ventes[Prix_unitaire] * fact_ventes[Quantite_Achetee]), DISTINCTCOUNT(fact_ventes[fk_Client])), ALLEXCEPT('Dim_Date', 'Dim_Date'[Annee], 'Dim_Date'[Mois])) / 10	Mesure la dépense moyenne par client chaque mois
Prix_Unitaire_produit_Carrefour	CALCULATE(AVERAGE(fact_ventes[Prix_Unitaire]), fact_ventes[fk_entreprise] = 2)	Calcule le prix unitaire moyen des produits vendus par l'entreprise Carrefour (id = 2).

Figure46-KPI and DAX Measures for Commercial Sector

Nom	Mesure Dax	Utilité
Nombre d'hotel	CALCULATE(COUNTROWS('fact_Reservation'), 'fact_Reservation'[fk_entreprise] = 5)	Compte le nombre d'hôtels liés à l'entreprise ayant l'ID 5.
Nombre d'hotel Totale	COUNTROWS('fact_Reservation')	Compte le nombre total de réservations d'hôtels.
Average Hotel Price – TravelTodo	CALCULATE(AVERAGE(fact_Reservation[prix_hotel]), fact_Reservation[fk_entreprise] = 5)	Calcule le prix moyen des hôtels réservés via l'entreprise TravelTodo
Rate_Hotel_Individuel	VAR HotelRate = LOOKUPVALUE(Dim_hotel[rate], Dim_hotel[id_hotel], fact_Reservation[fk_hotel]) RETURN IF(fact_Reservation[fk_entreprise] = 5, HotelRate)	Cette mesure retourne le rate de l'hôtel pour chaque réservation associée à l'entreprise 5.
Nom_Hotel_Individuel	VAR HotelNom = LOOKUPVALUE(Dim_hotel[name], Dim_hotel[id_hotel], fact_Reservation[fk_hotel]) RETURN IF(fact_Reservation[fk_entreprise] = 5, HotelNom)	Cette mesure retourne le nom de l'hôtel pour chaque réservation liée à l'entreprise 5.
Hotel Price Traveltodo	IF(fact_Reservation[fk_entreprise] = 5, fact_Reservation[prix_hotel], BLANK())	retourne le prix de l'hôtel uniquement pour les réservations de l'entreprise 5 (Traveltodo)
1 Etoiles	CALCULATE(COUNTROWS('fact_Reservation'), 'fact_Reservation'[fk_entreprise] = 5, 'fact_Reservation'[nombre_etoile] = 1)	Compte les réservations d'hôtels 1 étoiles pour l'entreprise 5.
2 Etoiles	CALCULATE(COUNTROWS('fact_Reservation'), 'fact_Reservation'[fk_entreprise] = 5, 'fact_Reservation'[nombre_etoile] = 2)	Compte les réservations d'hôtels 2 étoiles pour l'entreprise 5.

Figure47- KPI and DAX Measures for Tourism Sector

3 Etoiles	CALCULATE(COUNTROWS('fact_Reservation'), 'fact_Reservation'[fk_entreprise] = 5, 'fact_Reservation'[nombre_etoile] = 3)	Compte les réservations d'hôtels 3 étoiles pour l'entreprise 5.
4 Etoiles	CALCULATE(COUNTROWS('fact_Reservation'), 'fact_Reservation'[fk_entreprise] = 5, 'fact_Reservation'[nombre_etoile] = 4)	Compte les réservations d'hôtels 4 étoiles pour l'entreprise 5.
5 Etoiles	CALCULATE(COUNTROWS('fact_Reservation'), 'fact_Reservation'[fk_entreprise] = 5, 'fact_Reservation'[nombre_etoile] = 5)	Compte les réservations d'hôtels 5 étoiles pour l'entreprise 5.
Satisfaction élevée	VAR HotelRateRaw = LOOKUPVALUE(Dim_hotel[rate], Dim_hotel[id_hotel], fact_Reservation[fk_hotel]) VAR HotelRate = VALUE(HotelRateRaw) RETURN IF(fact_Reservation[fk_entreprise] = 5 && HotelRate >= 8, "Satisfaction élevée", BLANK())	Retourne "Satisfaction élevée" si l'hôtel a un score ≥ 8 pour l'entreprise 5.
Satisfaction Moyenne	VAR HotelRateRaw = LOOKUPVALUE(Dim_hotel[rate], Dim_hotel[id_hotel], fact_Reservation[fk_hotel]) VAR HotelRate = VALUE(HotelRateRaw) RETURN IF(fact_Reservation[fk_entreprise] = 5 && HotelRate >= 4 && HotelRate <= 7, "Satisfaction Moyenne", BLANK())	Retourne "Satisfaction Moyenne" si l'hôtel a un score entre 4 et 7 pour l'entreprise 5.
Satisfaction faible	VAR HotelRateRaw = LOOKUPVALUE(Dim_hotel[rate], Dim_hotel[id_hotel], fact_Reservation[fk_hotel]) VAR HotelRate = VALUE(HotelRateRaw) RETURN IF(fact_Reservation[fk_entreprise] = 5 && HotelRate <= 3, "Satisfaction faible", BLANK())	Retourne "Satisfaction faible" si l'hôtel a un score ≤ 3 pour l'entreprise 5.

Figure 48- KPI and DAX Measures for Tourism Sector

5.3 Machine Learning Results Visualization

All ML predictions are accessible through our Angular-based web platform, allowing users to test models in real time. The results are displayed dynamically after input submission via Flask APIs, making the system responsive and user-friendly.

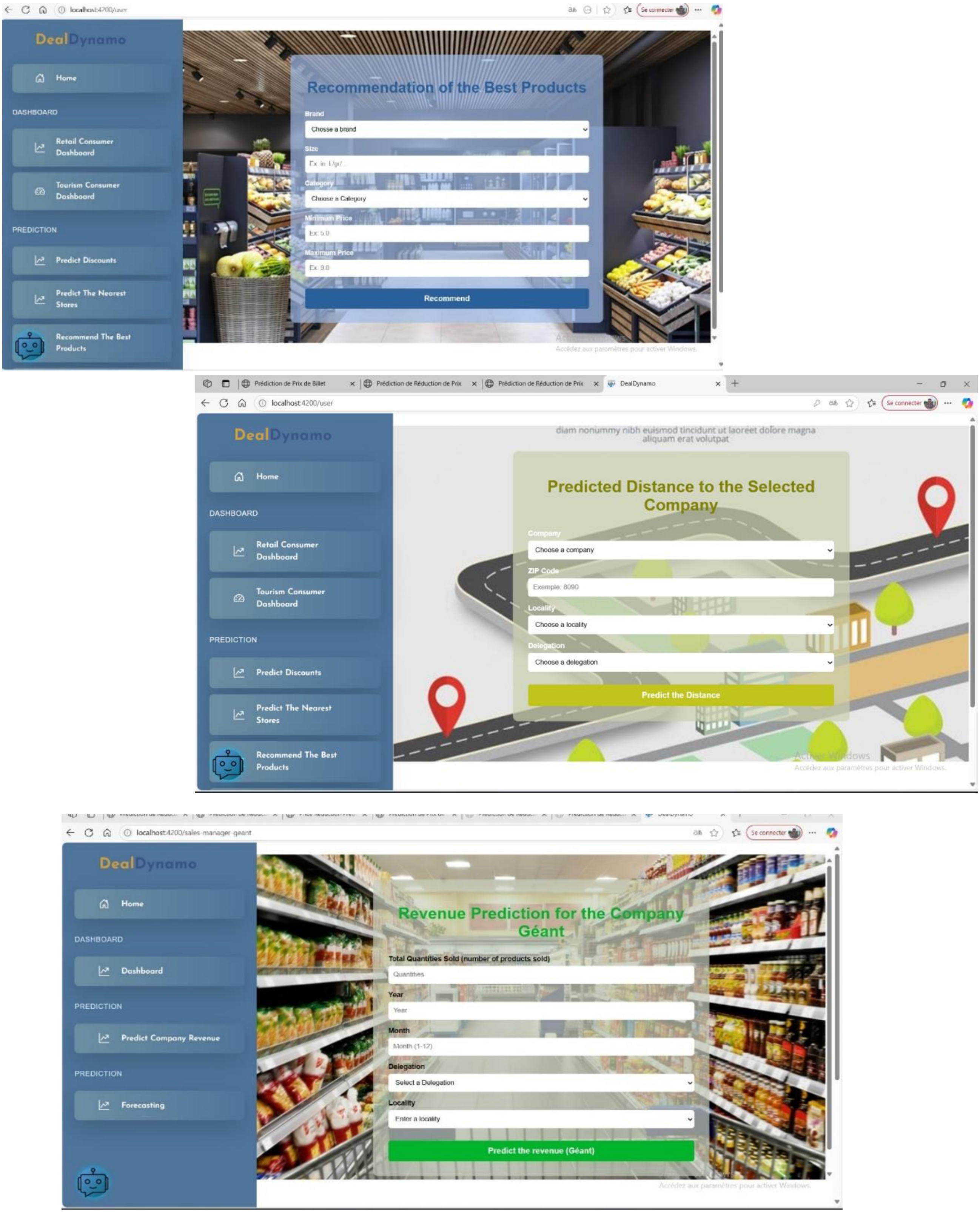
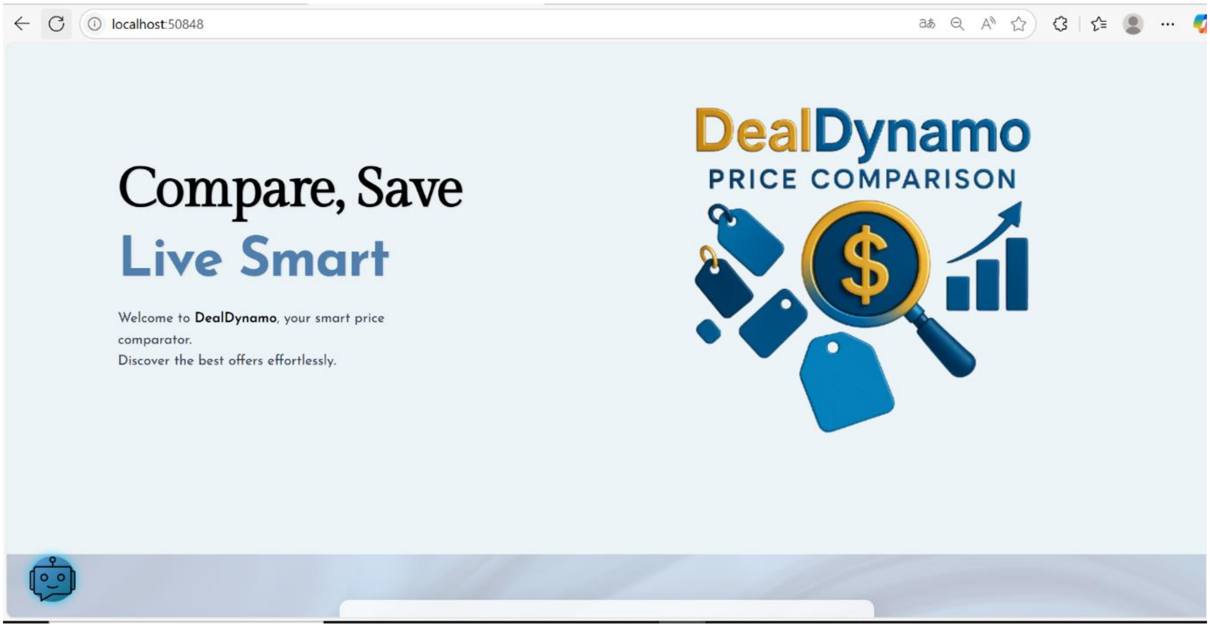


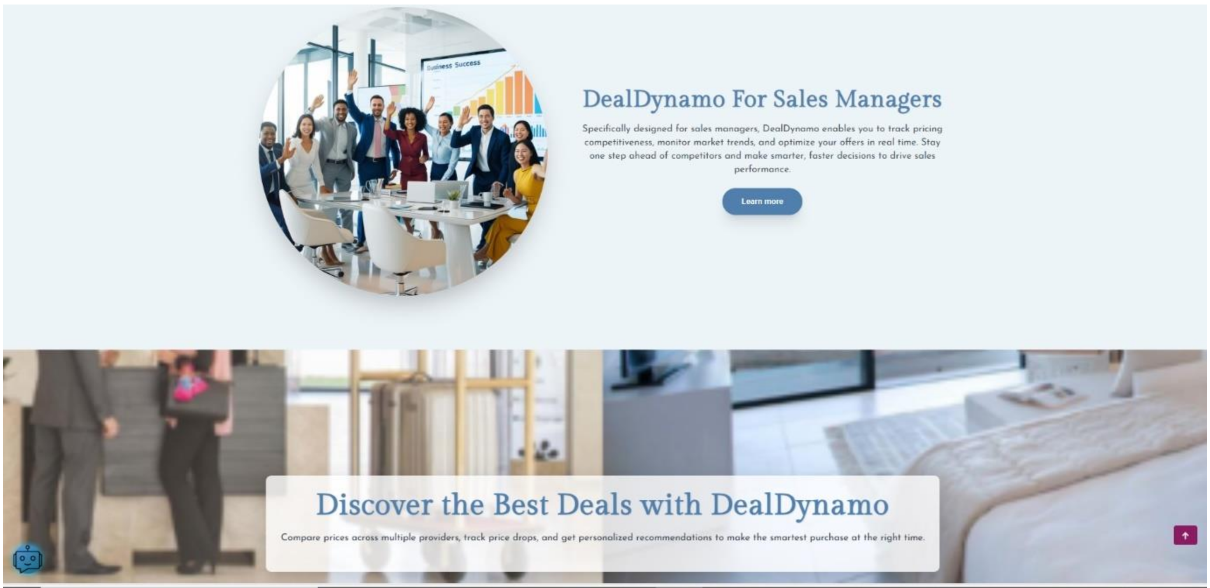
Figure49- Machine Learning Results Visualization

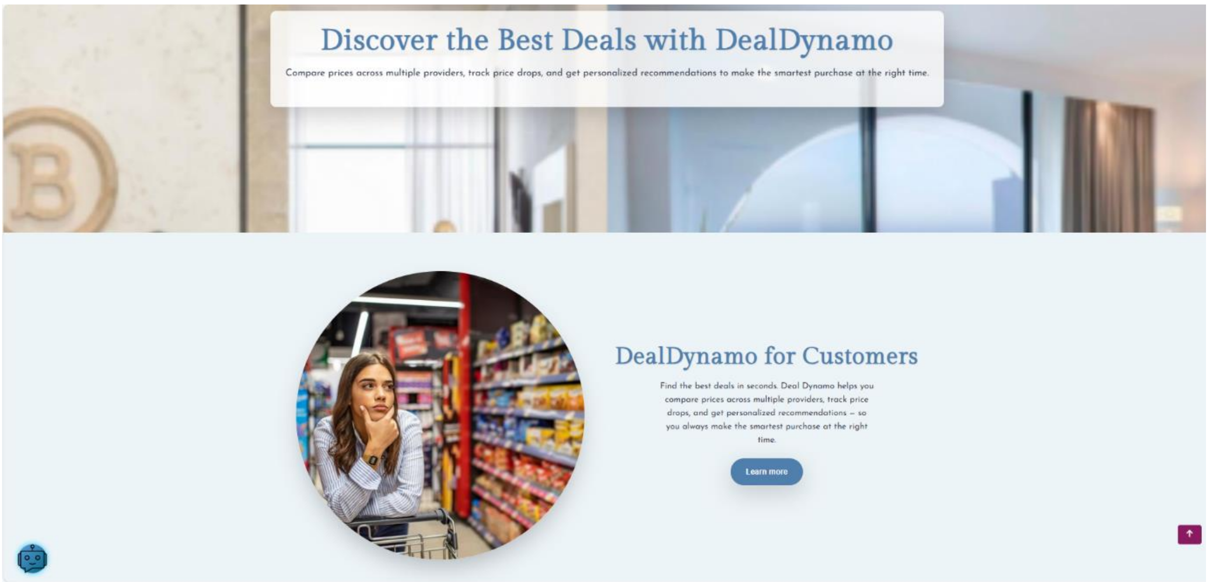
5.4 Full Integration and Final Interface Presentation

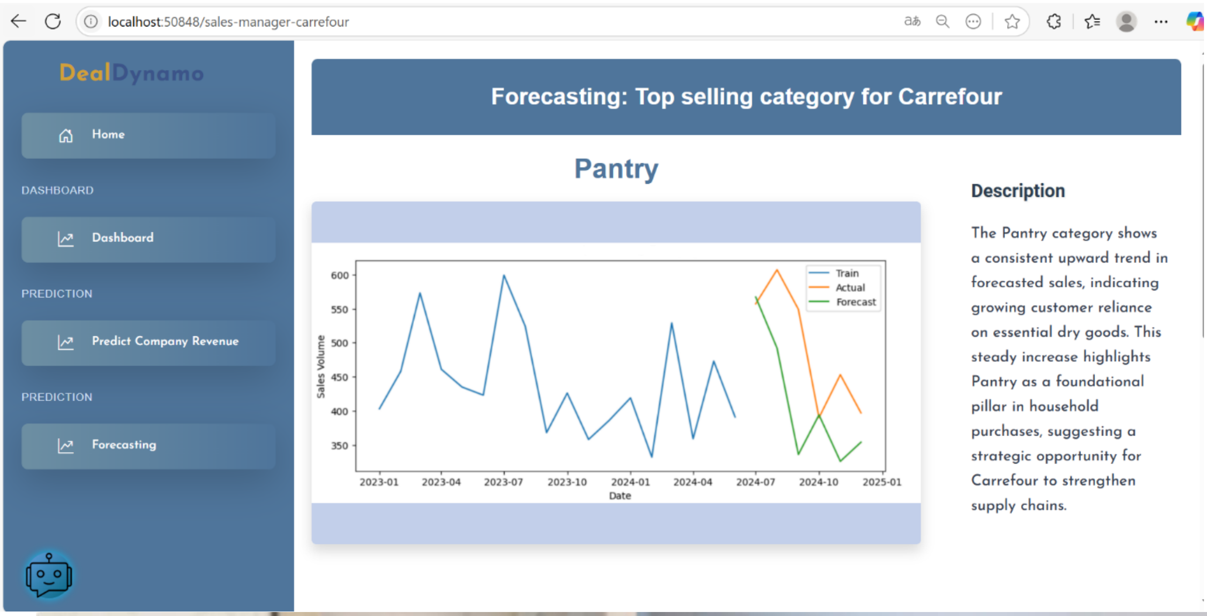
his section presents the final version of our integrated web platform, which brings together all the components developed throughout the project. It combines machine learning models, interactive dashboards, and user-friendly interfaces into a single accessible and efficient system. Whether it's for clients seeking insights or sales managers needing decision-support tools, the platform delivers a seamless experience, showcasing both the predictive results and key business indicators in real-time.

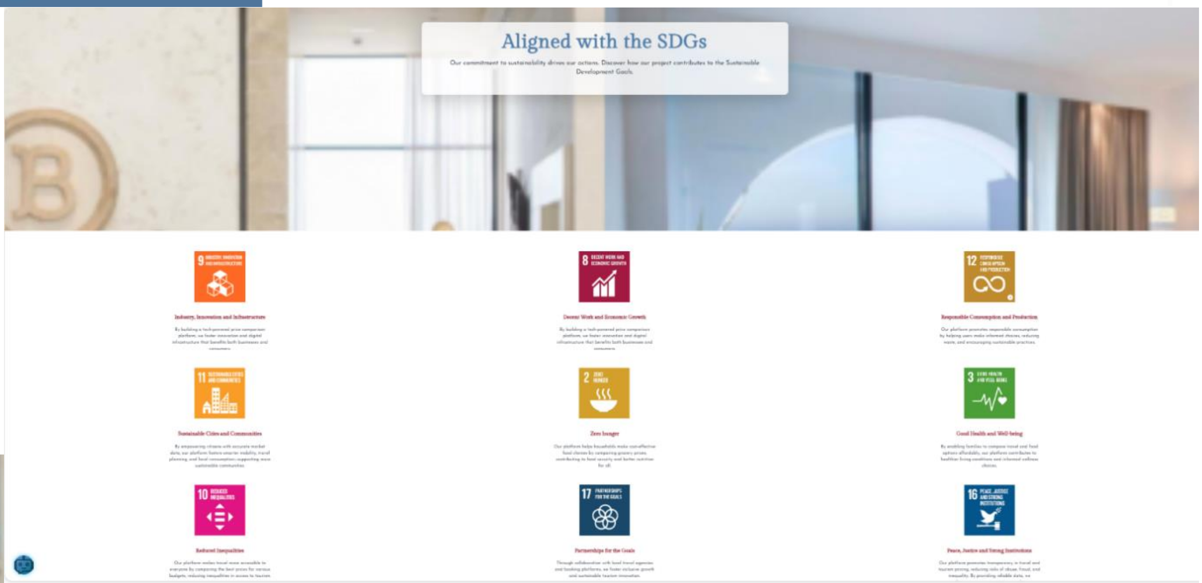
5.4.1 Some pictures of our plateform

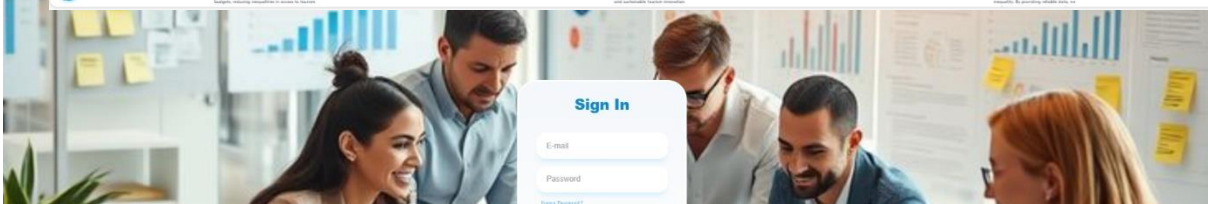





Discover the Best Deals with DealDynamo
Compare prices across multiple providers, track price drops, and get personalized recommendations to make the smartest purchase at the right time.
DealDynamo for Customers
Find the best deals in seconds. Deal Dynamo helps you compare prices across multiple providers, track price drops, and get personalized recommendations – so you always make the smartest purchase at the right time.
Learn more


localhost:50848/sales-manager-carrefour
DealDynamo
Home
DASHBOARD
Dashboard
PREDICTION
Predict Company Revenue
PREDICTION
Forecasting
Forecasting: Top selling category for Carrefour
Pantry
Train
Actual
Forecast
Sales Volume
Date
600
550
500
450
400
350
2023-01
2023-04
2023-07
2023-10
2024-01
2024-04
2024-07
2024-10
2025-01
Description
The Pantry category shows a consistent upward trend in forecasted sales, indicating growing customer reliance on essential dry goods. This steady increase highlights Pantry as a foundational pillar in household purchases, suggesting a strategic opportunity for Carrefour to strengthen supply chains.


Aligned with the SDGs


Sign In
E-mail
Password

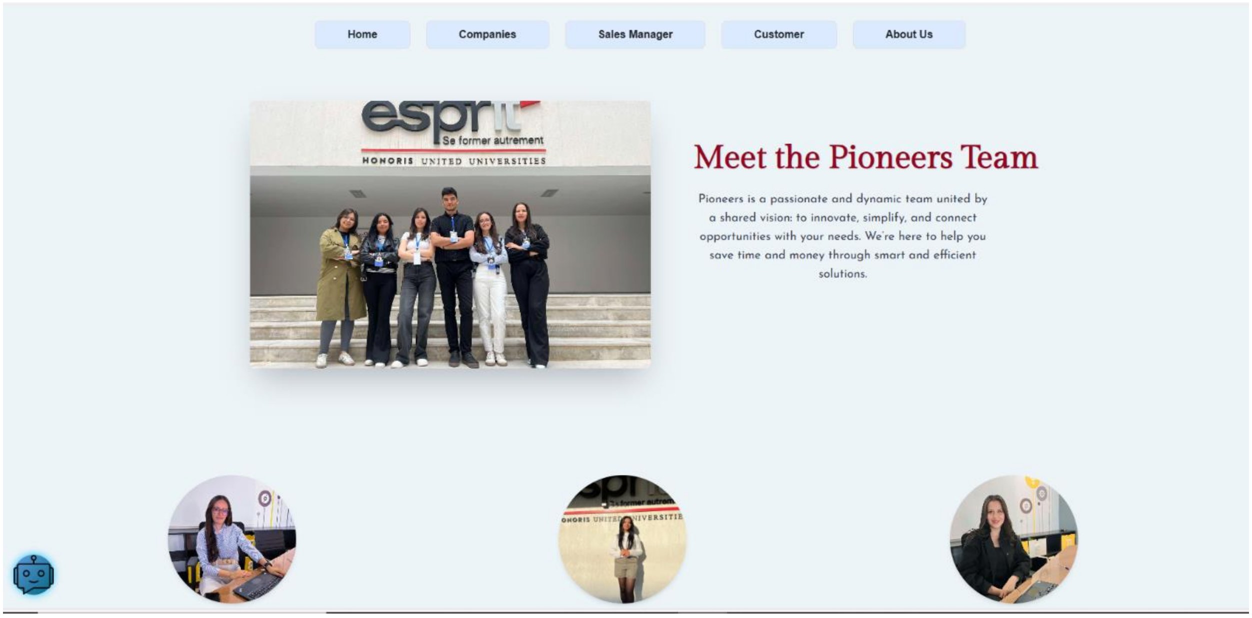


Figure50-Web Site Overview

Conclusion

In this chapter, we showcased the full power of data visualization and business intelligence within our project. By leveraging Power BI dashboards, we provided decision-makers with clear insights into sales trends, flight analytics, and customer behaviors. Key performance indicators and filters allowed users to explore data dynamically and make informed decisions.

We also highlighted how machine learning results were visualized in a meaningful way, ensuring transparency and interpretability of our predictive models. Finally, we presented the complete integration of all components—dashboards, ML models, and interactive interfaces—into a unified web platform, offering both usability and scalability. This chapter marks the transition from technical development to real-world application, where data-driven solutions become actionable tools for users and business stakeholders alike.

Chapter 6: Project Dissemination and Collaboration

This final chapter highlights the dissemination and collaborative aspects of the project, particularly through LinkedIn communications and GitHub activity. These demonstrate real-world engagement and team synergy throughout the development process.

6.1 Promotion on LinkedIn

To increase the visibility and impact of our project *DealDynamo*, we shared a detailed post on LinkedIn highlighting its purpose, technologies used, team collaboration, and alignment with Sustainable Development Goals (SDGs).

This professional communication helped us promote our work, showcase our skills, and engage with a broader audience, including peers and industry professionals.

Nour Ben Abid • 1er
Étudiante en ERP/BI à ESPRIT (Ecole Supérieure Privée d'Ingénierie et de Tech...
16 h • Modifié •

🌍 What if one platform could empower both consumers and companies to make smarter, data-driven decisions while supporting sustainable development?

🚀 After months of hard work and collaboration, I'm thrilled to introduce DealDynamo — a smart, user-friendly platform developed as part of our final-year integrated project at **ESPRIT (Ecole Supérieure Privée d'Ingénierie et de Technologies)**

📦 DealDynamo simplifies price comparison in two major sectors:

🛍️ Retail and 🌍 Tourism

🎯 Our mission is to empower both consumers and businesses with real-time, reliable insights — helping them make smarter, data-driven decisions through personalized recommendations and predictive analytics.

🧠 Technologies & Methods Used:
- Uses web scraping, OCR, and AI for product matching
- Built dashboards with Power BI
- Integrated ETL with Talend
- Developed Python Flask APIs for models and backend
- Chatbot assistant integrated into the Angular front-end

🔍 For consumers, Deal Dynamo offers:
- 📉 Price drop predictions on retail and travel products
- ✈️ Flight fare estimation based on custom inputs
- 📍 Store distance prediction by location
- 🛒 Product recommendations & hotel clustering

📊 For businesses, we provide dynamic dashboards to track:
• Sales performance
• Customer behavior
• Transaction insights
• Revenue forecasting/prediction

✅ SDG alignment:
• ODD 2 – Zero Hunger
• ODD 3 – Good Health & Well-being
• ODD 9 – Industry, Innovation & Infrastructure
• ODD 10 – Reduced Inequalities
• ODD 12 – Responsible Consumption
• ODD 17 – Partnerships for the Goals

🧑‍💻 I had the pleasure of working with an amazing team:Nour Boukhris, Assma Hajbi, Med Aziz Ben Ajmia, Roua Mtar , Erij triaa

🙏 Huge thanks to our mentors for their guidance and support throughout the project : Inès Mhaya, PhD , Sirine ZAABOUTI , hela mejri , malek naghmouchi

👉 Would you use DealDynamo to make smarter purchase or travel decisions? Share your thoughts!
#IntegratedProject #Innovation #ESPRIT #BI #ODDs #Teamwork #Power_Bi #AI

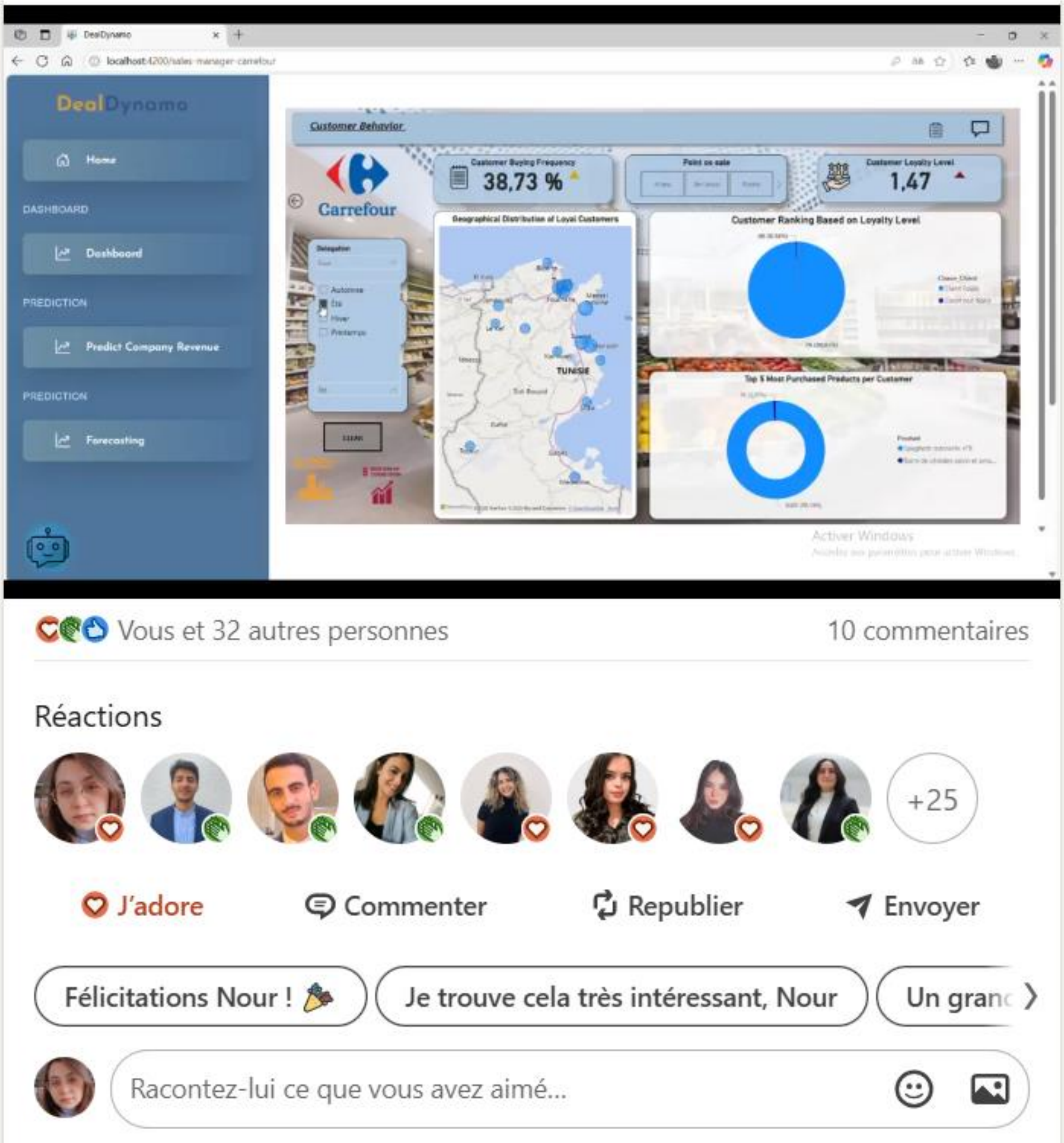

DealDynamo
Home
DASHBOARD
Dashboard
PREDICTION
Predict Company Revenue
PREDICTION
Forecasting
Customer Behavior
Carrefour
Customer Buying Frequency
38,73 %
Point on sale
Customer Loyalty Level
1,47
Geographical Distribution of Loyal Customers
Customer Ranking Based on Loyalty Level
Top 5 Most Purchased Products per Customer
TUNISIE
Vous et 32 autres personnes
10 commentaires
Réactions
+25
J'adore
Commenter
Republier
Envoyer
Félicitations Nour !
Je trouve cela très intéressant, Nour
Racontez-lui ce que vous avez aimé…

Figure51-Nour Ben Abid's Linkedin Post

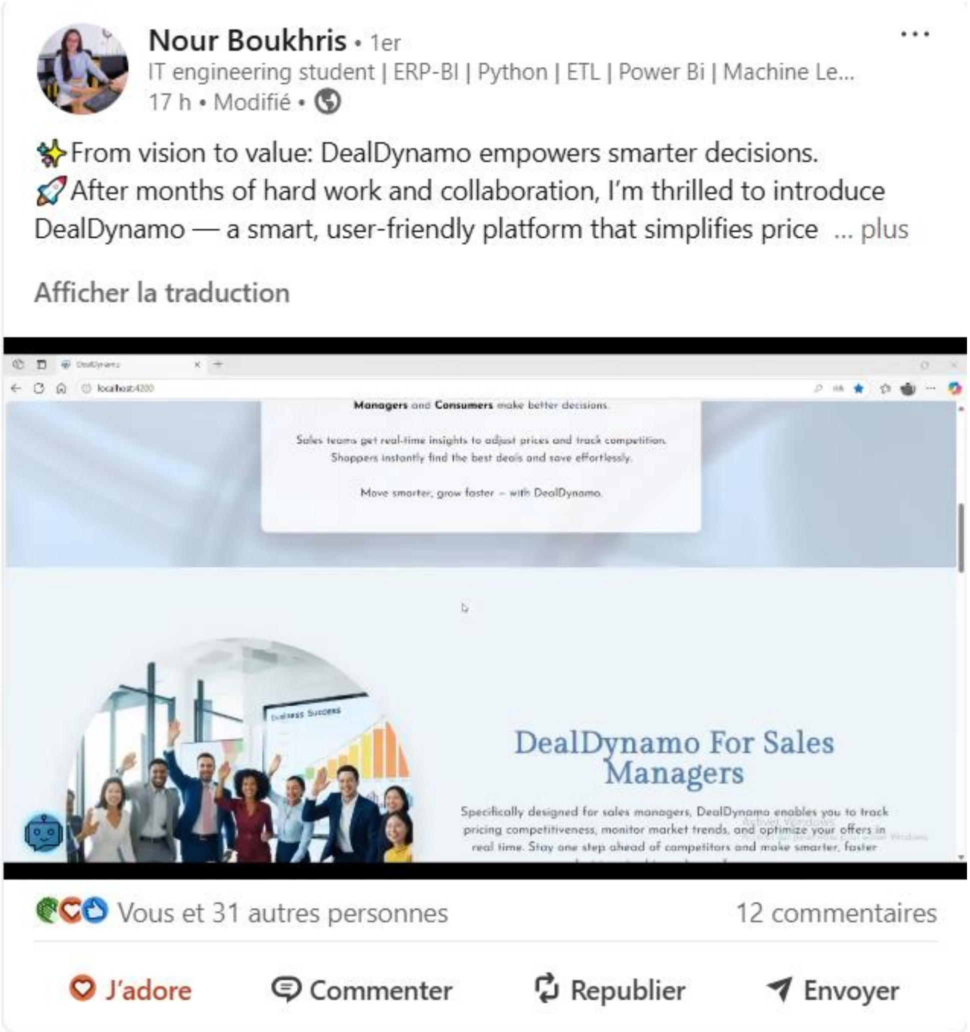


Figure52-Nour Boukhris' Linkedin Post

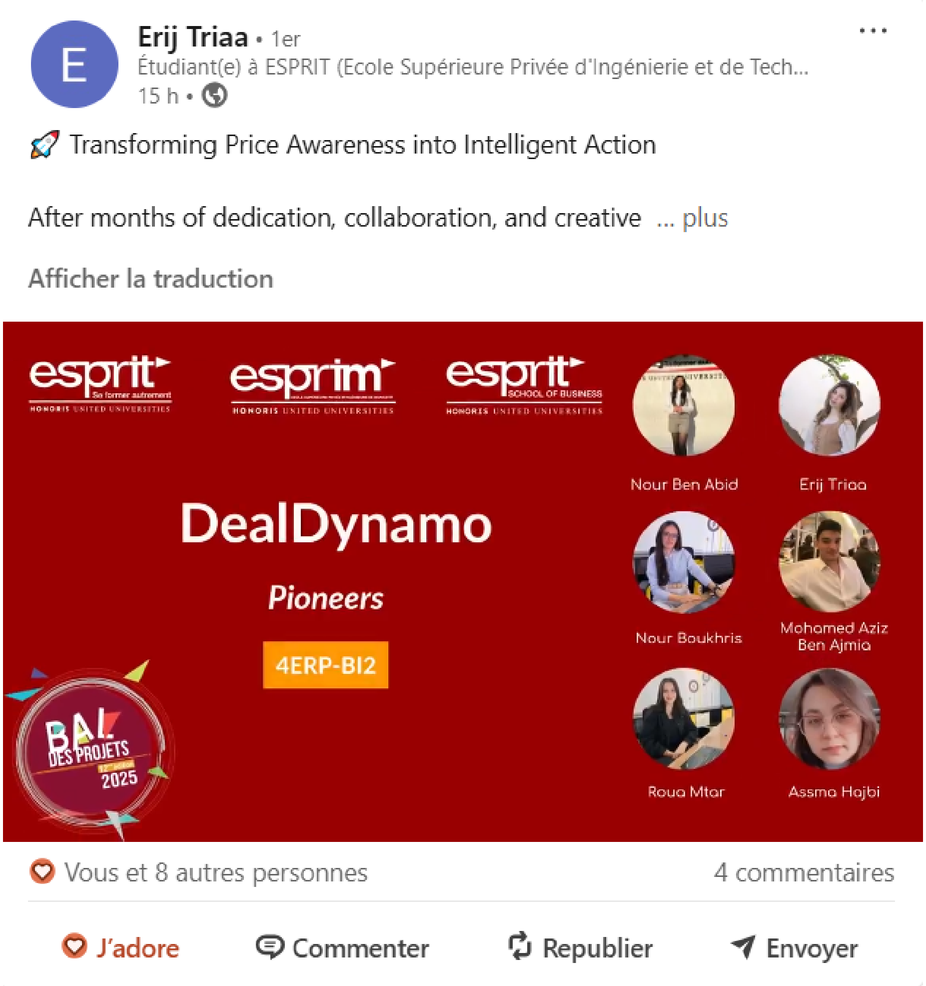


Figure53-Erij Triaa's Linkedin Post

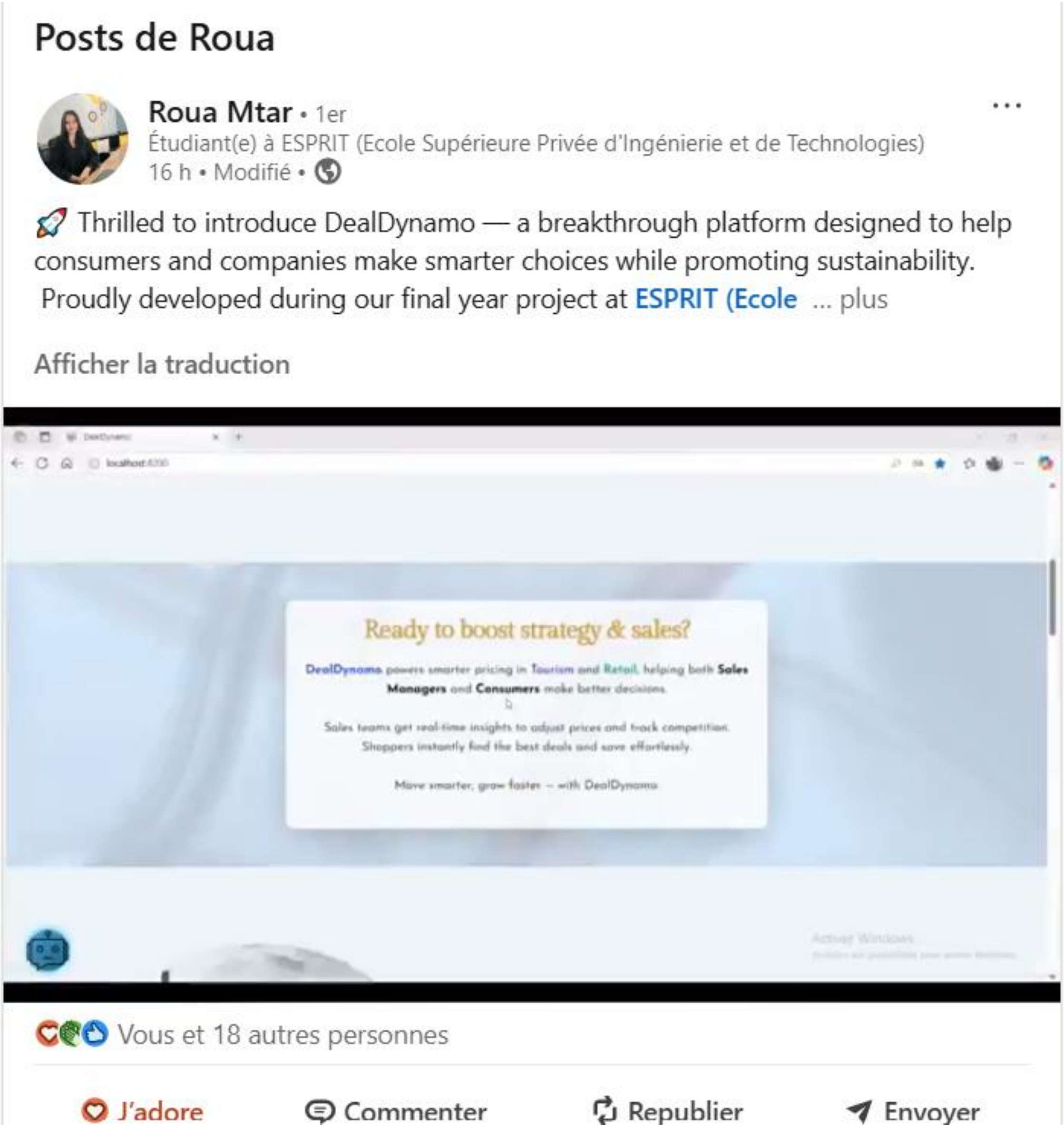


Figure54- Roua Mtar's Linkedin Post

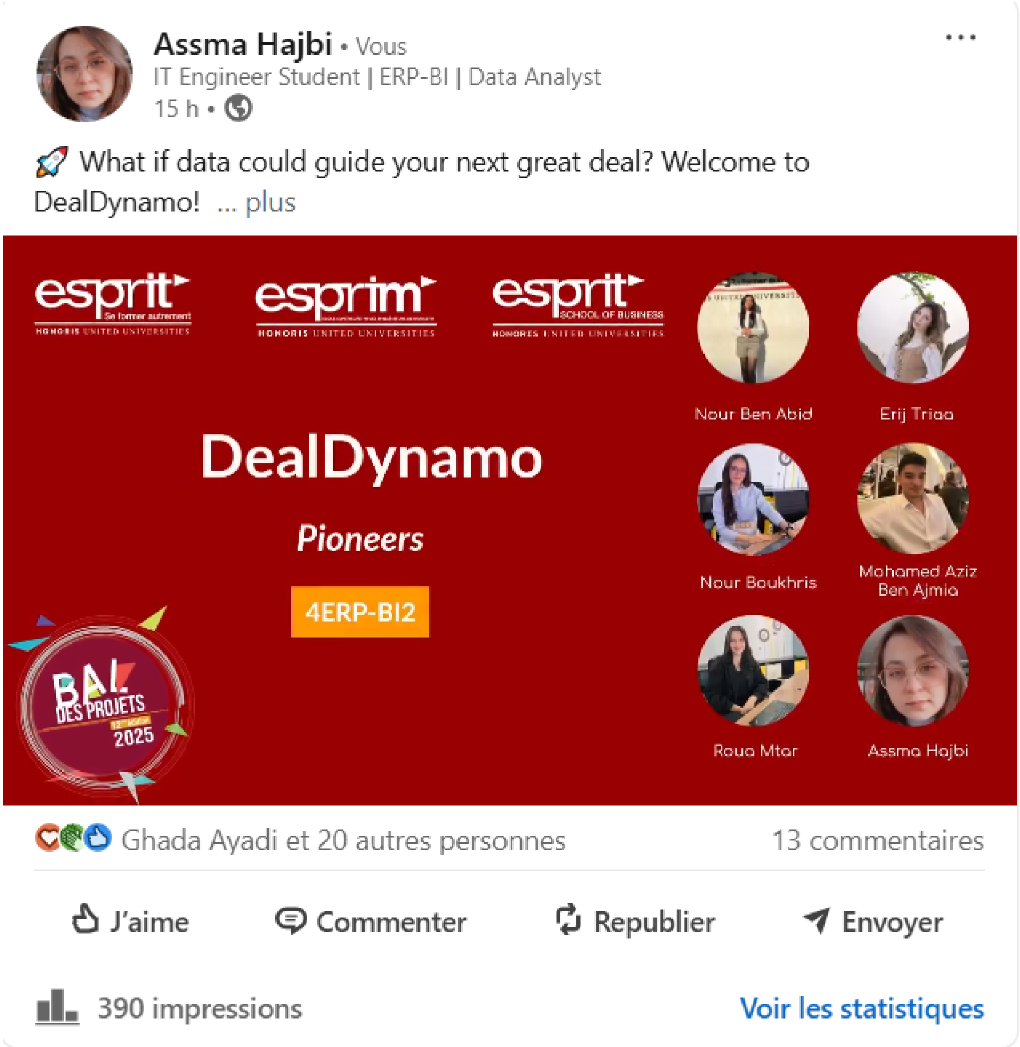


Figure55- Assma Hajbi's Linkedin Post

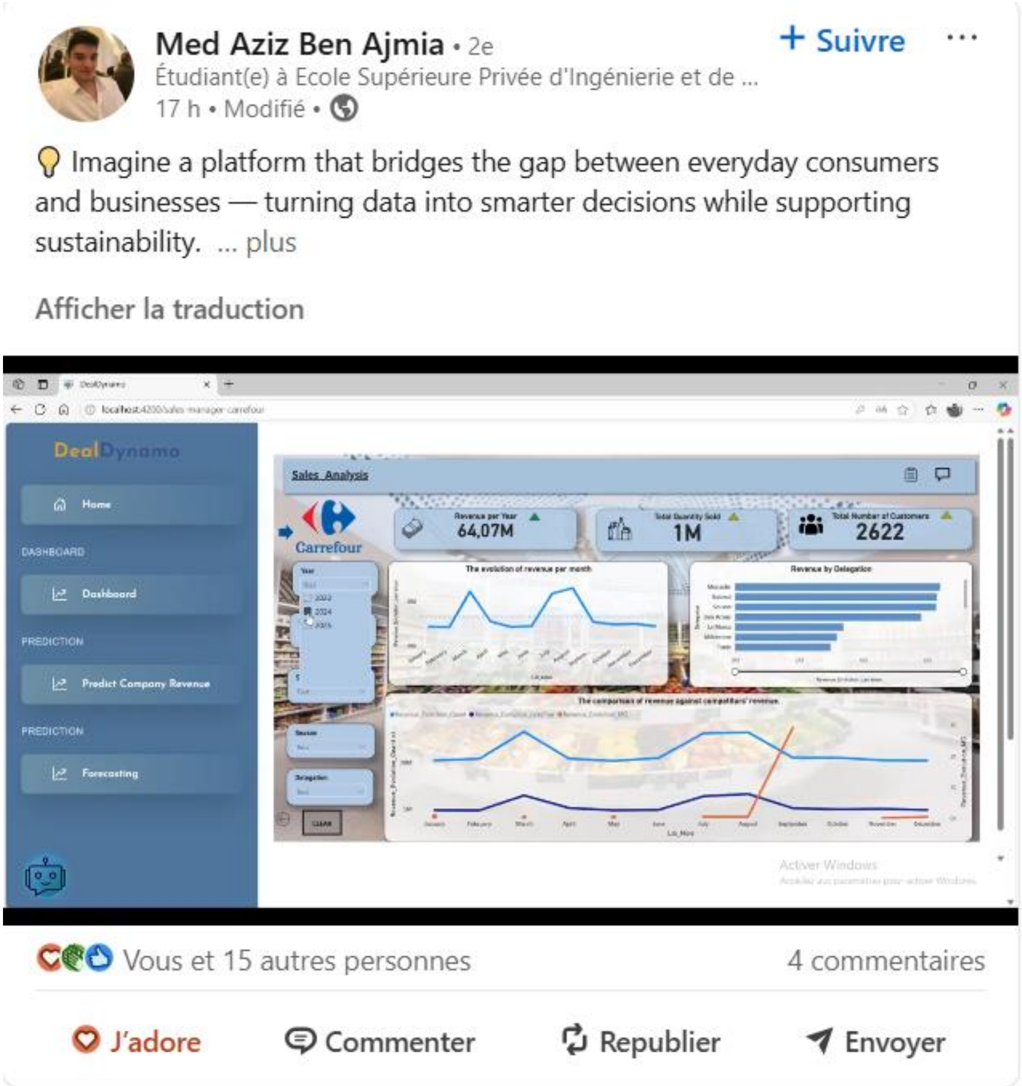


Figure56-Mohamed Aziz Ben Ajmia's Linkedin Post

6.2 GitHub Collaboration

To ensure effective collaboration and agile project management, we used GitHub throughout the development process.

The repository showcases our commits, issue tracking, and teamwork, reflecting continuous integration and transparency in code contributions

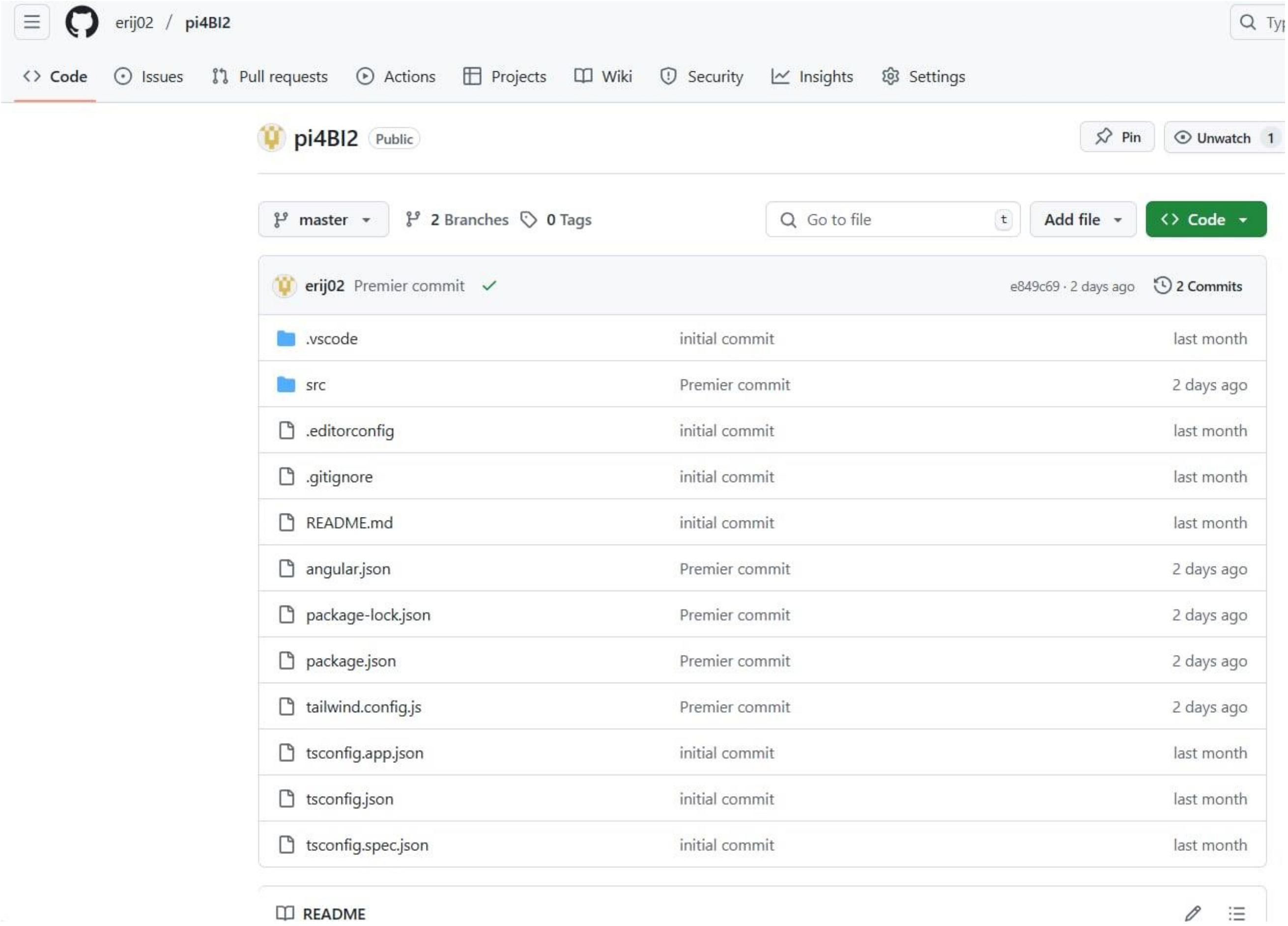


Figure57-Git Implementation

The full source code and project details are available on our GitHub repository:

https://erij02.github.io/pi4BI2/

Conclusion

This chapter highlighted the efforts made to promote and showcase our project beyond technical implementation. By sharing our work on professional platforms like LinkedIn and GitHub, we aimed to increase its visibility, demonstrate collaboration, and reflect our commitment to innovation and sustainable impact.

Conclusion and Future Enhancements

Summary of Achievements

This project culminated in the successful development of an intelligent price comparison system, offering both end-users and company decision-makers a powerful tool for market insight and strategic planning. By leveraging Business Intelligence techniques and Machine Learning models, we created an end-to-end data-driven solution integrating data extraction, warehousing, visualization, and predictive analytics within a web-based interface.

Challenges Faced

During the project, we encountered several technical and operational challenges, such as heterogeneous data formats, integration complexities, and optimization of performance for large datasets. These hurdles allowed us to deepen our expertise and refine our methodology across the entire BI pipeline.

Future Directions

Having already delivered intelligent recommendations, predictive alerts, and real-time price tracking, DealDynamo has proven its value as a smart decision-support system. But our ambition doesn’t stop here.

We envision a mobile-friendly platform and tighter integration with business CRMs, enabling companies to act on insights instantly. These directions are not just technical upgrades—they are steps toward turning DealDynamo into a truly adaptive, enterprise-grade solution.

The foundation is strong. The future is scalable. The vision is clear: transform data into strategy, and insights into action.