National College of Ireland

Configuration Manual

MSc Research Project
AI For Business

Alfonso Pinto
Student ID: X2338028

School of Computing
National College of Ireland

Supervisor: Rejwanul Haque

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Alfonso Pinto
Student ID:	X23389028
Programme:	AI For Business
Year:	2025
Module:	MSc Research Project
Supervisor:	Rejwanul Haque
Submission Due Date:	11/08/2025
Project Title:	Configuration Manual
Word Count:	1600
Page Count:	13

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Alfonso Pinto
Date:	11th August 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Alfonso Pinto
X23389028

1 Introduction

This Configuration Manual documents the exact environment, data sources, and execution steps required to reproduce the results of the MSc project ***"AI-Powered Predictive Dashboard for Retail Decision-Making."*** It covers:

- **Dataset & preprocessing:** how the "Warehouse and Retail Sales" CSV is prepared and aggregated.
- **Modeling pipeline:** running Prophet, XGBoost, and LSTM to generate forecasts and performance metrics via `main_forecasting.py`.
- **Frontend dashboard:** launching the Streamlit app (`app.py`) that reads the model outputs and provides business and technical views for analysis.
- **Dependencies & setup:** installing the required libraries, configuring paths, and verifying outputs.
- **Expected artifacts:** where CSV summaries, figures, and logs are saved, and how they connect to the dashboard.

The manual is intended for replicability on a standard laptop and assumes basic familiarity with Python and virtual environments.

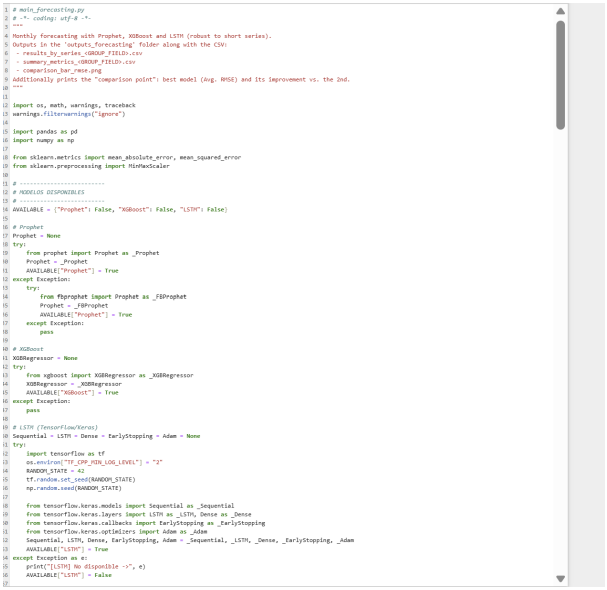
2 Project Overview

This project develops a **custom AI-powered predictive dashboard** to help retail decision-makers forecast product demand, monitor inventory risks, and plan promotional campaigns using open-source tools and publicly available data.

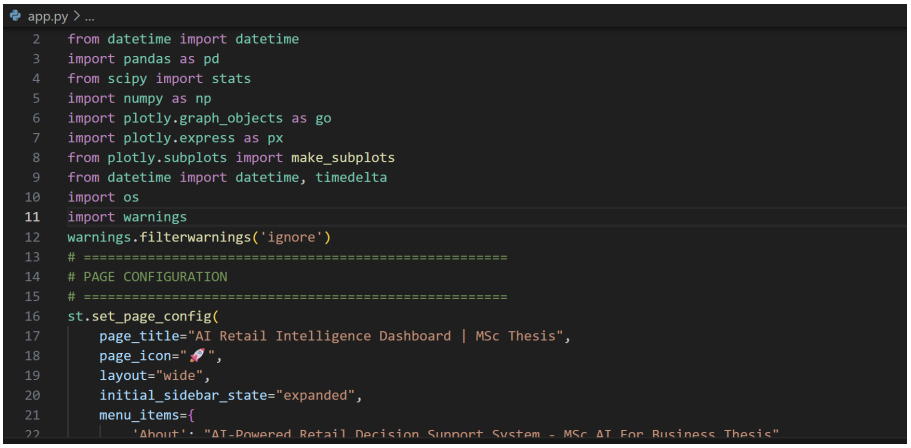
The workflow has two main components:

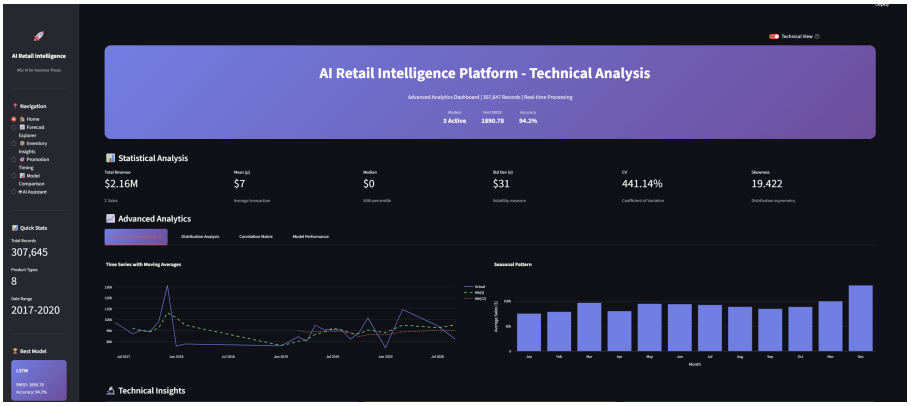
1. **Backend forecasting pipeline** (`main_forecasting.py`):
   - Loads the *Warehouse and Retail Sales* dataset (307,647 records).
   - Cleans, transforms, and aggregates sales data at the product-category level.
   - Trains and evaluates **three forecasting models**:
     - **Prophet** – interpretable, trend-seasonality modeling.
     - **XGBoost** – robust tree-based regression.

  - **LSTM** – deep learning for sequential patterns.
- Saves results in structured CSVs and generates RMSE/MAE comparison charts.



2. **Frontend dashboard** (`app.py`):
   - Built with **Streamlit** for interactivity and accessibility.
   - Reads backend outputs to present **role-specific views**:
     - *Business View*: simplified, action-oriented insights.
     - *Technical View*: advanced analytics, statistical tests, and model diagnostics.
   - Includes dedicated modules:
     - **Home** – KPIs, trends, and quick alerts.
     - **Forecast Explorer** – model-driven predictions with export options.
     - **Inventory Insights** – stock-level monitoring.
     - **Promotion Timing** – seasonality and campaign timing suggestions.
     - **Model Comparison** – performance metrics and statistical significance tests.
     - **AI Assistant** – interactive Q&A.





The final solution bridges **AI model accuracy** with **practical usability**, enabling non-technical retail managers to make timely, data-driven decisions without coding knowledge.

3 System Requirements

This section outlines the minimum hardware and software specifications required to execute the backend forecasting pipeline and launch the Streamlit dashboard without performance issues.

3.1 Hardware Specification

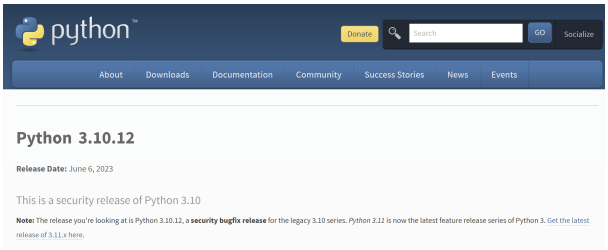
The project was developed and tested on a standard laptop. The following specs (or higher) are recommended for smooth execution:

- **Processor:** Intel® Core™ i7 (10th Gen or later) or equivalent AMD processor
- **Cores:** 8 physical cores
- **Logical Processors:** 16
- **RAM:** 16 GB or more
- **Storage:** At least 10 GB of free SSD storage (for dataset, outputs, and libraries)
- **GPU (optional):** NVIDIA GPU with CUDA support (e.g., GTX/RTX series) for faster LSTM training
- **Operating System:** Windows 10/11, macOS, or Linux (Ubuntu recommended for deployment)

3.2 Software Specification

3.2.1 Programming Language

- **Python 3.10** (project tested on Python 3.10.12)



3.2.2 Python Libraries

The following key libraries are used in this project:

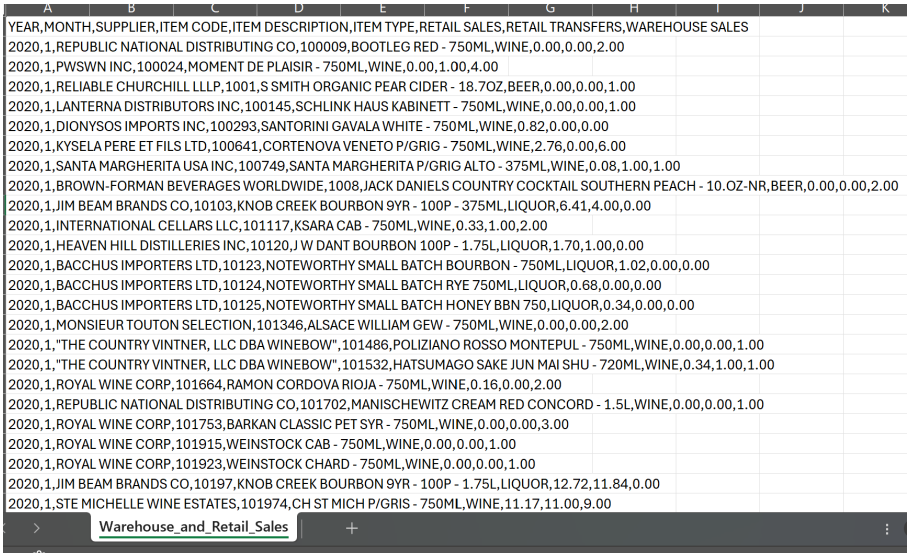
- **Data Handling:** `pandas`, `numpy`
- **Visualization:** `matplotlib`, `seaborn`, `plotly`
- **Machine Learning:** `scikit-learn`, `xgboost`
- **Deep Learning:** `tensorflow` (for LSTM implementation)
- **Time Series Forecasting:** `prophet` (or `fbprophet` as fallback)
- **Web App Development:** `streamlit`
- **Utility & Processing:** `os`, `warnings`, `scipy`, `datetime`

3.2.3 Dataset

The project uses the publicly available **Warehouse and Retail Sales** dataset from the U.S. Government Data Catalog. This dataset provides detailed monthly records of sales, warehouse movements, and transfers, making it ideal for retail demand forecasting.

3.2.4 Dataset Details

- **Total Records:** 307,647
- **File Format:** CSV (Comma-Separated Values)
- **Update Frequency:** Monthly (snapshot used for this project is static)
- **Public Access:** Available at `https://catalog.data.gov/dataset/warehouse-and-retail-sales`



A	B	C	D	E	F	G	H	I	J	K
YEAR,MONTH,SUPPLIER,ITEM CODE,ITEM DESCRIPTION,ITEM TYPE,RETAIL SALES,RETAIL TRANSFERS,WAREHOUSE SALES										
2020,1,REPUBLIC NATIONAL DISTRIBUTING CO,100009,BOOTLEG RED - 750ML,WINE,0.00,0.00,2.00										
2020,1,PWSWN INC,100024,MOMENT DE PLAISIR - 750ML,WINE,0.00,1.00,4.00										
2020,1,RELIABLE CHURCHILL LLLP,1001,S SMITH ORGANIC PEAR CIDER - 18.7OZ,BEER,0.00,0.00,1.00										
2020,1,LANTERNA DISTRIBUTORS INC,100145,SCHLINK HAUS KABINETT - 750ML,WINE,0.00,0.00,1.00										
2020,1,DIONYSOS IMPORTS INC,100293,SANTORINI GAVALA WHITE - 750ML,WINE,0.82,0.00,0.00										
2020,1,KYSELA PERE ET FILS LTD,100641,CORTENOVA VENETO P/GRIG - 750ML,WINE,2.76,0.00,6.00										
2020,1,SANTA MARGHERITA USA INC,100749,SANTA MARGHERITA P/GRIG ALTO - 375ML,WINE,0.08,1.00,1.00										
2020,1,BROWN-FORMAN BEVERAGES WORLDWIDE,1008,JACK DANIELS COUNTRY COCKTAIL SOUTHERN PEACH - 10.OZ-NR,BEER,0.00,0.00,2.00										
2020,1,JIM BEAM BRANDS CO,10103,KNOB CREEK BOURBON 9YR - 100P - 375ML,LIQUOR,6.41,4.00,0.00										
2020,1,INTERNATIONAL CELLARS LLC,101117,KSARA CAB - 750ML,WINE,0.33,1.00,2.00										
2020,1,HEAVEN HILL DISTILLERIES INC,10120,J W DANT BOURBON 100P - 1.75L,LIQUOR,1.70,1.00,0.00										
2020,1,BACCHUS IMPORTERS LTD,10123,NOTEWORTHY SMALL BATCH BOURBON - 750ML,LIQUOR,1.02,0.00,0.00										
2020,1,BACCHUS IMPORTERS LTD,10124,NOTEWORTHY SMALL BATCH RYE 750ML,LIQUOR,0.68,0.00,0.00										
2020,1,BACCHUS IMPORTERS LTD,10125,NOTEWORTHY SMALL BATCH HONEY BBN 750,LIQUOR,0.34,0.00,0.00										
2020,1,MONSIEUR TOUTON SELECTION,101346,ALSACE WILLIAM GEW - 750ML,WINE,0.00,0.00,2.00										
2020,1,"THE COUNTRY VINTNER, LLC DBA WINEBOW",101486,POLIZIANO ROSSO MONTEPUL - 750ML,WINE,0.00,0.00,1.00										
2020,1,"THE COUNTRY VINTNER, LLC DBA WINEBOW",101532,HATSUMAGO SAKE JUN MAI SHU - 720ML,WINE,0.34,1.00,1.00										
2020,1,ROYAL WINE CORP,101664,RAMON CORDOVA RIOJA - 750ML,WINE,0.16,0.00,2.00										
2020,1,REPUBLIC NATIONAL DISTRIBUTING CO,101702,MANISCHEWITZ CREAM RED CONCORD - 1.5L,WINE,0.00,0.00,1.00										
2020,1,ROYAL WINE CORP,101753,BARKAN CLASSIC PET SYR - 750ML,WINE,0.00,0.00,3.00										
2020,1,ROYAL WINE CORP,101915,WEINSTOCK CAB - 750ML,WINE,0.00,0.00,1.00										
2020,1,ROYAL WINE CORP,101923,WEINSTOCK CHARD - 750ML,WINE,0.00,0.00,1.00										
2020,1,JIM BEAM BRANDS CO,10197,KNOB CREEK BOURBON 9YR - 100P - 1.75L,LIQUOR,12.72,11.84,0.00										
2020,1,STE MICHELLE WINE ESTATES,101974,CH ST MICH P/GRIS - 750ML,WINE,11.17,11.00,9.00										

Warehouse_and_Retail_Sales

3.2.5 Key Variables

- **Temporal Variables:**
  - `YEAR` – Year of the transaction
  - `MONTH` – Month of the transaction
- **Categorical Variables:**
  - `SUPPLIER` – Supplier identifier
  - `ITEM DESCRIPTION` – Full product name/description
  - `ITEM TYPE` – Product category (primary grouping variable)
- **Target Variable:**
  - `RETAIL SALES` – Monthly sales quantity/value for retail stores
- **Supporting Variables:**
  - `WAREHOUSE SALES` – Quantity/value distributed from warehouse
  - `RETAIL TRANSFERS` – Quantity/value transferred between retail locations

3.2.6 Why This Dataset Was Selected

- **Real-world relevance** for retail supply chain analysis
- **Granularity** that supports both product-level and category-level forecasts
- **Compatibility** with multiple forecasting models due to its time-series structure
- **Public availability** ensuring reproducibility

4 Environment Setup

This section outlines how to prepare the working environment to run the forecasting pipeline (`main_forecasting.py`) and launch the interactive Streamlit dashboard (`app.py`). Following these steps ensures that all files, dependencies, and outputs are correctly organized.

4.1 Folder Structure

Before running the code, ensure the project directory is organized as follows:

project directory.py

```
AI_Retail_Dashboard/
│
├── data/
│   └── Warehouse_and_Retail_Sales.csv
│
├── outputs_forecasting/
│   ├── summary_metrics_ITEM_TYPE.csv
│   ├── results_by_series_ITEM_TYPE.csv
│   └── comparison_bar_rmse.png
│
├── main_forecasting.py
├── app.py
├── requirements.txt
└── README.md
```

- **data/** → Stores the source dataset.
- **outputs_forecasting/** → Stores forecast results and metrics generated by `main_forecasting.py`.
- **main_forecasting.py** → Backend script for preprocessing, model training, and output generation.
- **app.py** → Streamlit dashboard code.
- **requirements.txt** → List of required Python packages.

4.2 Installing Dependencies

1. Create and activate a virtual environment:

```
python -m venv venv
source venv/bin/activate    # Mac/Linux
venv\Scripts\activate       # Windows
```

2. Install the required libraries:

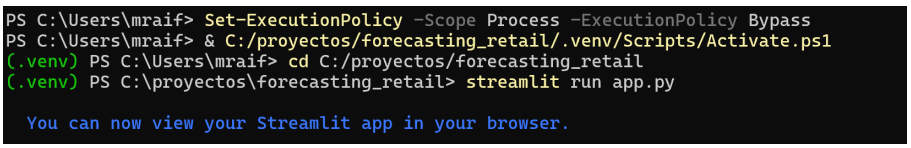
```
pip install -r requirements.txt
```

4.3 File Path Configuration

- In `main_forecasting.py`, update the `CSV_PATH` variable to point to your dataset location if it differs from the default:

  ```
  CSV_PATH = r'./data/Warehouse_and_Retail_Sales.csv'
  ```

- Ensure `OUT_DIR` in the same script points to the desired output folder (default is `./outputs_forecasting/`).



4.4 Environment Verification

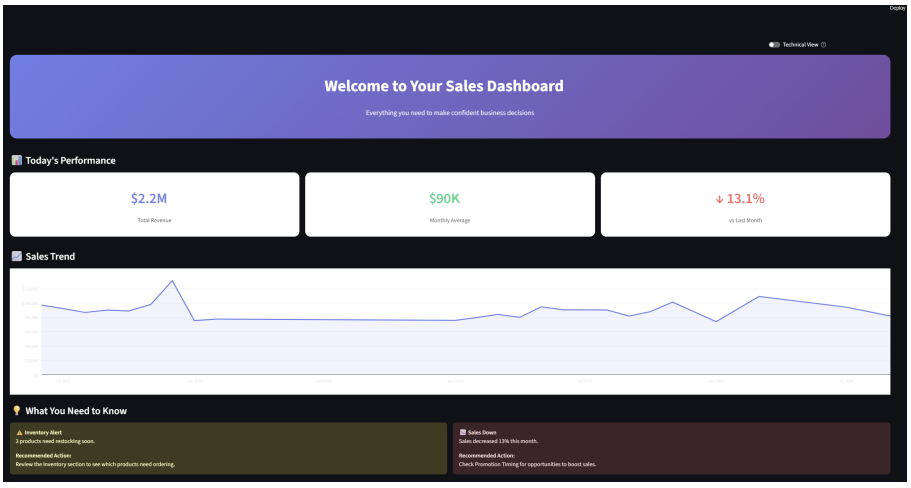
After setup:

- Confirm that Python can import all required libraries without errors.
- Run:

  ```
  python main_forecasting.py
  ```

  to ensure outputs are generated in `outputs_forecasting/`.

- Check that `summary_metrics_ITEM_TYPE.csv` and `results_by_series_ITEM_TYPE.csv` exist.



4.5 Order of Execution

This section describes the step-by-step process to execute the forecasting pipeline and launch the dashboard. Following this order ensures that the backend generates the required outputs before the frontend is launched.

4.5.1 Step 1 – Run the Forecasting Pipeline (`main_forecasting.py`)

Purpose: Preprocess the dataset, train and evaluate the forecasting models (Prophet, XGBoost, LSTM), and generate result files for the dashboard.

Execution:

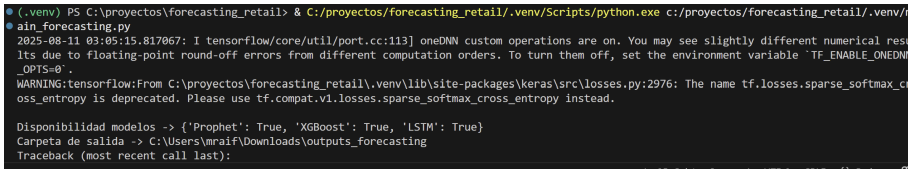
1. Open a terminal in the project root directory.
2. Run:

```
python main_forecasting.py
```

Main Tasks Performed:

- **Data Cleaning:** Handles missing values, type conversions, and monthly aggregation.
- **Feature Engineering:** Creates seasonality index and lag features.
- **Model Training:** Fits Prophet, XGBoost, and LSTM models for top product categories (`ITEM TYPE`).
- **Evaluation:** Calculates MAE and RMSE for each model and category.
- **Output Generation:**
  - `summary_metrics_ITEM_TYPE.csv` → Overall performance metrics.
  - `results_by_series_ITEM_TYPE.csv` → Model-wise results for each product category.
  - `comparison_bar_rmse.png` → RMSE comparison chart.

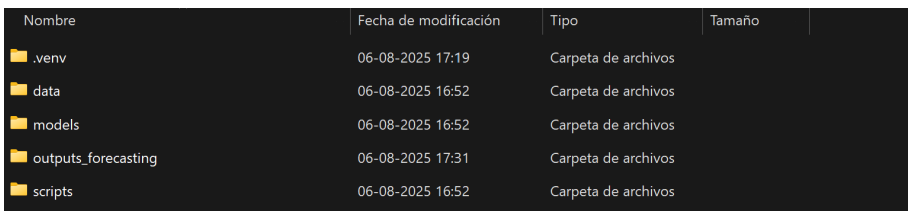
Logs: Execution progress and any model errors are printed to the terminal.



4.5.2 Step 2 – Verify Outputs

Before launching the dashboard, ensure the following files exist in `outputs_forecasting/`:

- `summary_metrics_ITEM_TYPE.csv`
- `results_by_series_ITEM_TYPE.csv`
- `comparison_bar_rmse.png`



4.5.3 Step 3 – Launch the Dashboard (`app.py`)

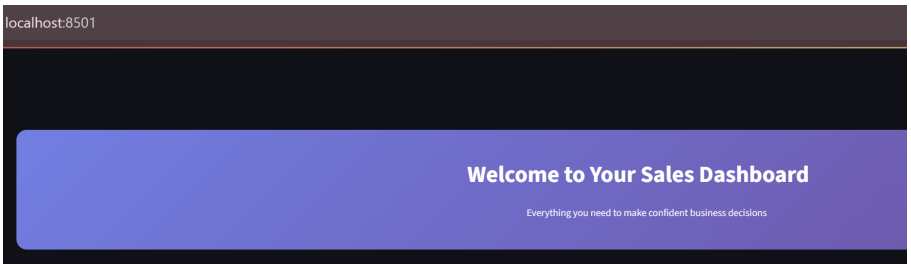
Purpose: Visualize the forecasting results and provide interactive analysis tools.

Execution:

1. In the terminal, run:

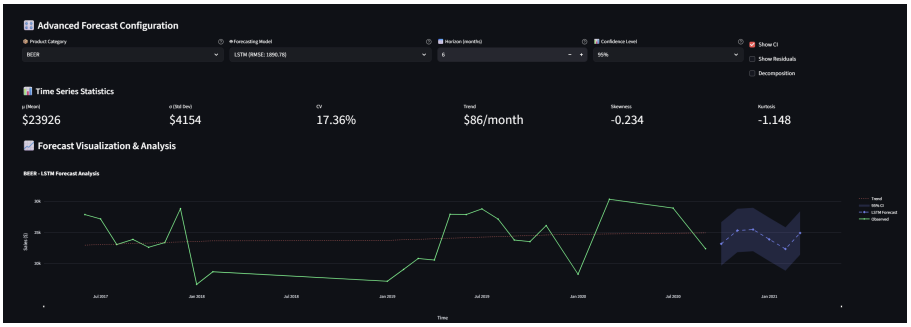
```
streamlit run app.py
```

2. The dashboard will open in your default web browser (usually at `http://localhost:8501`).



Main Features:

- **Home:** Key performance indicators, sales trends, and alerts.
- **Forecast Explorer:** Forecast visualizations with export options.
- **Inventory Insights:** Inventory risk detection.
- **Promotion Timing:** Seasonality-based promotional suggestions.
- **Model Comparison:** Performance tables and statistical significance tests.
- **AI Assistant:** Chat-based interaction for questions.



5 Forecasting & Dashboard Workflow

This section explains the internal workflow of the project — from data preprocessing to generating forecasts, and finally visualizing results in the Streamlit dashboard. It mirrors the order in which the backend (`main_forecasting.py`) and frontend (`app.py`) interact.

5.1 Backend Forecasting Process (`main_forecasting.py`)

5.1.1 Step 1 – Data Loading

- Reads the CSV dataset from `data/Warehouse_and_Retail_Sales.csv`.
- Verifies that all required columns (`YEAR`, `MONTH`, `ITEM TYPE`, `RETAIL SALES`, etc.) exist.

5.1.2 Step 2 – Preprocessing & Feature Engineering

- Converts `YEAR` and `MONTH` into a unified `DATE` column.
- Removes rows with missing or null sales values.
- Aggregates data monthly per product category (`ITEM TYPE`).
- Creates lag features for XGBoost and adjusts lookback windows for LSTM.
- Calculates additional variables such as total demand and seasonality index.

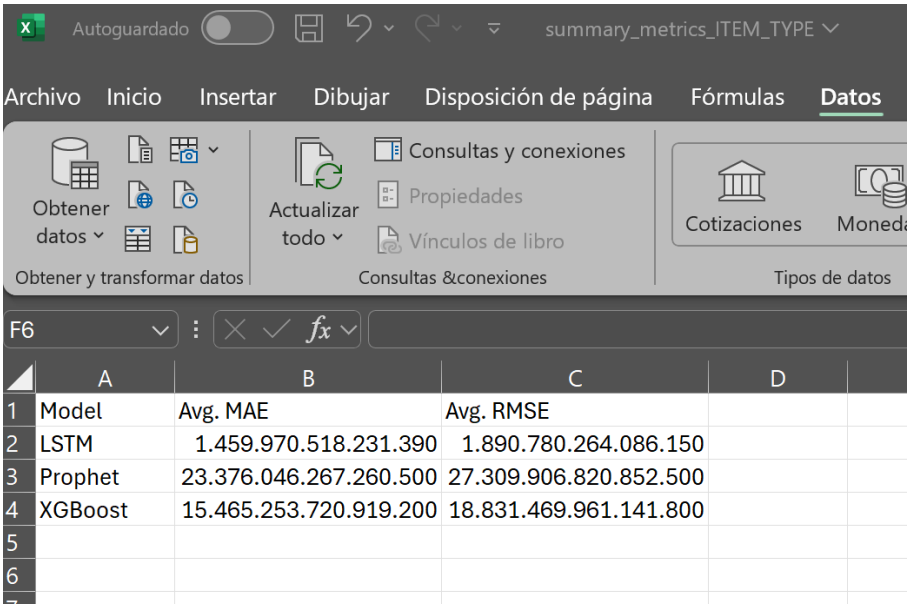
5.1.3 Step 3 – Model Training & Evaluation

For each top `ITEM TYPE` (by sales volume):

- **Prophet:** Fits trend-seasonality model and predicts next periods.
- **XGBoost:** Trains regression model with lag features and calendar variables.
- **LSTM:** Builds and trains deep learning model for sequential forecasting.
- **Evaluation Metrics:** MAE and RMSE for all models, saved in CSVs.

5.1.4 Step 4 – Output Generation

- Saves results to:
  - `outputs_forecasting/summary_metrics_ITEM_TYPE.csv`
  - `outputs_forecasting/results_by_series_ITEM_TYPE.csv`
  - `outputs_forecasting/comparison_bar_rmse.png`
- Outputs are structured for direct use in the Streamlit dashboard.



	A	B	C	D
1	Model	Avg. MAE	Avg. RMSE	
2	LSTM	1.459.970.518.231.390	1.890.780.264.086.150	
3	Prophet	23.376.046.267.260.500	27.309.906.820.852.500	
4	XGBoost	15.465.253.720.919.200	18.831.469.961.141.800	
5				
6				

5.2 Dashboard Data Flow (`app.py`)

When you run:

```
streamlit run app.py
```

the dashboard performs the following steps:

5.2.1 Step 1 – Data Loading

- Reads:
  - `summary_metrics_ITEM_TYPE.csv` (for model performance summaries)
  - `results_by_series_ITEM_TYPE.csv` (for category-level forecasts)
  - Original dataset CSV (for historical trend visualizations).

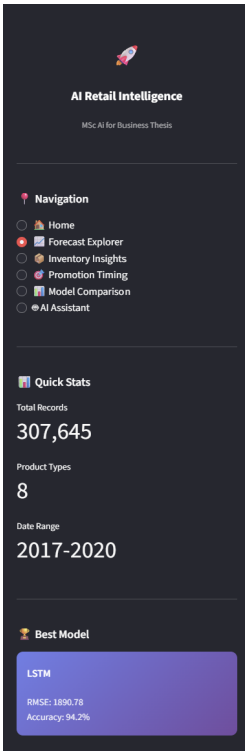
5.2.2 Step 2 – View Selection

- **Business View:** Simple metrics, clean charts, and actionable recommendations.
- **Technical View:** Detailed analytics, statistical tests, and model diagnostics.

5.2.3 Step 3 – Interactive Modules

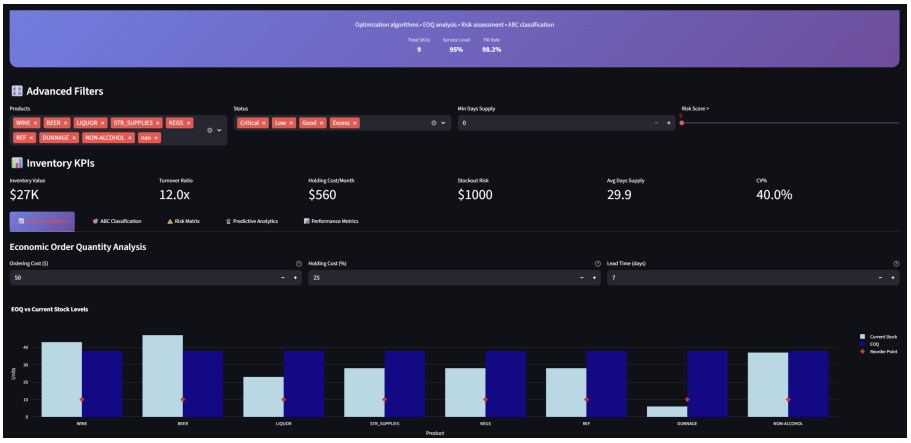
- **Home:** Shows KPIs, sales trends, and alerts on inventory or performance changes.
- **Forecast Explorer:** Interactive product category selection, time horizon selection, forecast visualization, export options (CSV and TXT summary).
- **Inventory Insights:** Highlights products at risk of stockouts.
- **Promotion Timing:** Shows seasonal peaks to plan campaigns.
- **Model Comparison:** Displays performance table and statistical test results.

- **AI Assistant:** Allows conversational queries about forecasts and metrics.



5.2.4 Step 4 – Output Presentation

- Charts are generated using Plotly for interactive zooming, hovering, and toggling.
- Recommendations are displayed in text cards with icons and color-coded status (success, warning, error).



With this, a user can replicate the entire pipeline: **run the backend to generate updated forecasts → launch the frontend to explore and analyze results**.

6 Conclusion

With this configuration manual, users can **fully reproduce the AI-powered predictive dashboard** for retail decision-making — from running the forecasting pipeline to exploring interactive insights in the Streamlit app.

The **execution order** is critical:

1. **Run** `main_forecasting.py` first to preprocess the dataset, train the models (Prophet, XGBoost, LSTM), evaluate results, and generate the required CSV and image outputs.
2. **Launch** `app.py` to load those outputs into the dashboard for visual exploration, decision support, and actionable recommendations.

By following the **environment setup**, **dataset preparation**, and **step-by-step execution instructions** provided in this manual, any user with the required hardware and software can replicate the results presented in the MSc research project.

The design ensures that:

- **Technical robustness** is achieved via a multi-model forecasting approach with proper evaluation metrics.
- **Practical usability** is maintained through a dual-view dashboard, making the tool accessible to both technical analysts and business managers.

Future improvements could include:

- Adding automated retraining pipelines.
- Expanding the dataset with external factors such as promotions and market trends.
- Deploying the dashboard to a cloud environment for remote access.