End-to-End Sales Forecasting and Data Warehousing Project

Complete Technical Implementation Guide

1 Project Overview

This project aims to design an end-to-end data pipeline for sales analysis, forecasting, and decision support. The workflow covers data cleaning, data warehousing, exploratory analysis, predictive modeling, and business visualization using industry-standard tools.

2 Task 1 – Data Collection and Preparation

2.1 Objective

Ensure data quality, consistency, and readiness for analysis.

2.2 Tools Used

- Python (Google Colab)
- Pandas, NumPy, Matplotlib

2.3 Steps Performed

2.3.1 Data Loading

The raw Excel dataset was loaded into Python using Pandas, and date fields were converted into proper datetime format.

```
df = pd.read_excel("BAWAHA_DATABASE.xlsx")
df["Date_Vente"] = pd.to_datetime(df["Date_Vente"])
```

2.3.2 Data Cleaning

- Removed negative or zero sales quantities
- Filled missing size values with "Unknown"

- Standardized text fields
- Removed duplicate and redundant columns

2.3.3 Output

A cleaned dataset was saved as a CSV file to be used in downstream tasks.

```
df.to_csv("BAWAHA_CLEANED_SALES.csv", index=False)
```

3 Task 2 – Data Warehouse Design

3.1 Objective

Create a structured data warehouse optimized for analytical queries.

3.2 Tools Used

- SQL Server
- SQL Server Management Studio (SSMS)
- SQL Server Integration Services (SSIS)

3.3 Star Schema Design

The warehouse follows a star schema with:

- One fact table: `Fact_Sales`
- Multiple dimension tables: Product, Date, Size, Color

3.4 SSIS Implementation

3.4.1 Project Creation

1. Open Visual Studio
2. Create a new **Integration Services Project**
3. Name the project and choose a save location

3.4.2 Flat File Configuration

1. Add a Data Flow Task
2. Configure Flat File Source
3. Select `BAWAHA_CLEANED_SALES.csv`
4. Set delimiter to comma and encoding to UTF-8

3.4.3 OLE DB Destination

1. Configure SQL Server connection
2. Map CSV columns to staging table
3. Resolve encoding issues using Data Conversion where needed

4 Task 3 – Exploratory Data Analysis (EDA)

4.1 Objective

Understand historical trends and customer behavior.

4.2 Tool Used

- Power BI

4.3 Steps Performed

- Connected Power BI to SQL Server warehouse
- Created measures for total sales and quantities
- Built visuals for:
  - Fast- and slow-moving products
  - Popular sizes and colors
  - Seasonal sales patterns
  - Stock-out frequency

5 Task 4 – Predictive Modeling

5.1 Objective

Forecast future sales and stock requirements.

5.2 Tools Used

- Python (Google Colab)
- Scikit-learn

5.3 Model Selection

A Random Forest Regressor was chosen due to:

- Ability to model non-linear relationships
- Robustness to noise and outliers
- No assumption of data distribution

5.4 Model Training

A 70/30 train-test split was applied.

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42)
```

5.5 Model Evaluation

Performance was measured using RMSE and R-squared.

```
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
r2 = r2_score(y_test, y_pred)
```

5.6 Forecast Storage

Predictions were stored in SQL Server in a dedicated table `Sales_Forecasts`.

6 Task 5 – Visualization and Reporting

6.1 Objective

Convert forecasts into actionable business insights.

6.2 Steps Performed

- Connected Power BI to forecast tables
- Created dashboards comparing:
  - Actual vs forecasted sales
  - Stock-out risks
  - Recommended reorder quantities
  - Forecast errors
- Configured scheduled data refresh

7 Conclusion

This project demonstrates a complete analytics pipeline from raw data ingestion to predictive insights and decision support. The integration of data engineering, machine learning, and visualization ensures scalability and business value.