

ANNEXE – SOURCE CODES

Sales Forecasting & Data Warehouse Project

A. Task 1 – Data Collection & Preparation (Python – Google Colab)

A.1 Data Loading

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt


# Load raw dataset

df = pd.read_excel("BAWAHA_DATABASE.xlsx")


# Convert date column

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

A.2 Data Cleaning

```
# Remove negative or zero sales

df = df[df["Quantité_Vendue"] > 0]


# Handle missing sizes
```

```
df["Taille"] = df["Taille"].fillna("Unknown")

df["Taille"] = df["Taille"].str.upper()


# Standardize text columns

text_cols = ["SKU", "Collection", "Nom_Produit", "Couleur", "Canal", "Paiement", "Pays"]

for col in text_cols:

    df[col] = df[col].astype(str).str.strip()


# Remove duplicate columns

df = df.drop(columns=["date_vente", "pays_commande"], errors="ignore")


# Save cleaned dataset

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

A.3 Exploratory Visualization

```
sales_over_time = df.groupby(df["Date_Vente"].dt.date)["Quantité_Vendue"].sum()


plt.figure(figsize=(14, 6))

sales_over_time.plot()

plt.xlabel("Date")

plt.ylabel("Units Sold")

plt.title("Historical Sales Trend Over Time")

plt.show()
```

B. Task 2 – Data Warehouse Design (SQL Server)

B.1 Staging Table

```
CREATE TABLE Stg_Sales_Cleaned (  
    Date_Vente DATE,  
    SKU VARCHAR(50),  
    Collection VARCHAR(100),  
    Nom_Produit VARCHAR(150),  
    Couleur VARCHAR(50),  
    Taille VARCHAR(20),  
    Quantite_Vendue INT,  
    Prix_Unitaire FLOAT,  
    Remise FLOAT,  
    Prix_Final FLOAT,  
    Canal VARCHAR(50),  
    Paiement VARCHAR(50),  
    Pays VARCHAR(50)  
);
```

B.2 Dimension Tables

```
CREATE TABLE Dim_Product (  
    Product_ID INT IDENTITY PRIMARY KEY,
```

```
SKU VARCHAR(50),  
Nom_Produit VARCHAR(150),  
Collection VARCHAR(100)  
);
```

```
CREATE TABLE Dim_Size (  
    Size_ID INT IDENTITY PRIMARY KEY,  
    Taille VARCHAR(20)  
);
```

```
CREATE TABLE Dim_Color (  
    Color_ID INT IDENTITY PRIMARY KEY,  
    Couleur VARCHAR(50)  
);
```

```
CREATE TABLE Dim_Date (  
    Date_ID INT IDENTITY PRIMARY KEY,  
    Date_Vente DATE,  
    Year INT,  
    Month INT,  
    Day INT  
);
```

B.3 Fact Table

```
CREATE TABLE Fact_Sales (  
    Sales_ID INT IDENTITY PRIMARY KEY,  
    Date_ID INT,  
    Product_ID INT,  
    Size_ID INT,  
    Color_ID INT,  
    Quantite_Vendue INT,  
    Prix_Unitaire FLOAT,  
    Remise FLOAT,  
    Prix_Final FLOAT,  
    Canal VARCHAR(50),  
    Paiement VARCHAR(50)  
);
```

C. Task 3 – Exploratory Data Analysis (Power BI)

Power BI visuals were created using the Fact_Sales table and dimensions.

- Bar charts: Sales by product, size, color
 - Line charts: Seasonal sales trends
 - Filters: Date, country, product
-

D. Task 4 – Predictive Modeling (Python – Google Colab)

D.1 Feature Engineering

```
from sklearn.model_selection import train_test_split
```

```
X = df[["Prix_Unitaire", "Remise"]]
```

```
y = df["Quantité_Vendue"]
```

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

D.2 Model Training

```
from sklearn.ensemble import RandomForestRegressor
```

```
model = RandomForestRegressor(  
    n_estimators=200,  
    random_state=42  
)
```

```
model.fit(X_train, y_train)
```

D.3 Model Evaluation

```
from sklearn.metrics import mean_squared_error, r2_score
```

```
import numpy as np
```

```
y_pred = model.predict(X_test)

rmse = np.sqrt(mean_squared_error(y_test, y_pred))

r2 = r2_score(y_test, y_pred)

print("RMSE:", rmse)

print("R-squared:", r2)
```

D.4 Forecast Generation

```
df["Predicted_Sales"] = model.predict(X)

forecast = df[[

    "Date_Vente", "SKU", "Nom_Produit",

    "Quantité_Vendue", "Predicted_Sales"

]]
```

E. Task 4 – Store Predictions in SQL Server

```
CREATE TABLE Sales_Forecasts (

    Forecast_ID INT IDENTITY PRIMARY KEY,

    Date_Vente DATE,

    SKU VARCHAR(50),
```

Actual_Sales INT,
Forecasted_Sales FLOAT
);

F. Task 5 – Visualization & Reporting (Power BI)

Power BI dashboards include:

- Actual vs Forecasted Sales
- Stock-out risk indicators
- Reorder quantity recommendations
- Forecast error analysis

Scheduled refresh configured using SQL Server connection.