

Task: Multi-Source Data Integration, Predictive Analytics & Optimization

Scenario:

You are tasked with analyzing and forecasting key performance metrics for a multinational company with complex interdependencies between regions, products, inventory, and customer behavior. Additionally, you need to identify potential operational bottlenecks and recommend optimization strategies.

Data Sources:

1. Excel File (sales_data.xlsx):

Contains **daily sales data** for each region.

Columns:

- OrderID (unique identifier)
- Region (e.g., North America, Europe, APAC, Middle East)
- Date
- Revenue
- ProductCategory
- UnitsSold
- DiscountApplied (Percentage discount on each sale)

2. SQL Database (inventory_db):

Contains **inventory and product data** across all regions.

- **Table 1: products**
 - ProductID
 - ProductCategory
 - Price
 - Supplier (e.g., Local, Imported)
 - ProductionCost

- **Table 2: inventory**

- ProductID
- Region
- StockLevel
- ReorderPoint
- LeadTime (in days)

3. Python-generated CSV File (customer_behavior.csv):

Contains **customer behavior metrics** and feedback scores.

Columns:

- CustomerID
 - Region
 - SatisfactionScore (1–10)
 - PurchaseFrequency
 - AverageSpend
 - ChurnRisk (High/Medium/Low)
 - ReferralCount (Number of new customers referred)
-

Additional Constraints:

1. Time Dependency:

- Sales and inventory levels are updated daily. Use time-series techniques for modeling trends and seasonality.
- Inventory has a **lead time** before replenishment. You must predict out-of-stock scenarios.

2. Revenue Leakage:

- Incorporate the effect of discounts (DiscountApplied) on revenue. Calculate **potential revenue loss** due to over-discounting.

3. Optimization:

- Develop an algorithm to optimize the ReorderPoint for each product to minimize out-of-stock incidents while reducing excess inventory costs.

4. **Supplier Analysis:**

- Compare performance between **Local** and **Imported** suppliers. Identify regions where imported suppliers are causing delays or higher costs.

5. **Customer Retention:**

- Use SatisfactionScore and ReferralCount to predict churn. Propose strategies to reduce churn risk in high-risk regions.
-

Requirements:

Step 1: Data Integration

1. Import the datasets into Python.
 2. Create relationships between the data sources (e.g., ProductID as a foreign key).
 3. Handle missing values using advanced imputation techniques (e.g., KNN or regression-based imputation).
-

Step 2: Advanced Analytics

1. Revenue Analysis:

- Calculate the **impact of discounts** on total revenue and profitability by region.
- Identify regions and product categories with **highest revenue leakage**.

2. Inventory Analysis:

- Predict inventory shortfalls using time-series forecasting (e.g., ARIMA, SARIMA).
- Suggest optimized ReorderPoint values based on predicted demand and lead times.

3. Supplier Analysis:

- Analyze cost and delays introduced by imported suppliers compared to local ones.
 - Provide recommendations for better supplier selection in underperforming regions.
-

Step 3: Predictive Modeling

1. Revenue Prediction:

- Build a predictive model using scikit-learn or XGBoost to forecast daily revenue. Include the following as predictors:
 - Region
 - ProductCategory
 - UnitsSold
 - DiscountApplied
 - AverageSpend

2. Churn Prediction:

- Use a classification model (e.g., Logistic Regression, Random Forest) to predict the churn risk for customers based on their SatisfactionScore, PurchaseFrequency, and ReferralCount.
-

Step 4: Reporting & Visualization

1. Create an **interactive dashboard** using Python libraries like **Dash** or **Plotly**.
 - Include filters for region and product category.
 - Provide visualizations for revenue trends, inventory shortages, and customer churn risk.
 2. Generate an Excel report summarizing:
 - Revenue impact of discounts.
 - Inventory bottlenecks.
 - Predicted churn rates and customer retention strategies.
-

Step 5: Automation

1. Write a script that:
 - Automatically fetches new data from the SQL database.
 - Re-trains predictive models when significant data changes are detected.
 - Updates the dashboard and report automatically.