Project Title: Supply Chain Optimization
Technologies: Data Cleansing, EDA, Visualization, Tableau/Streamlit
Domain: Supply Chain

Objective:

To develop an Tableau-based analysis model that optimizes the routing of orders from warehouses to customers, ensuring the fulfillment of constraints such as warehouse capacities, product-specific storage limitations, and transportation costs. The goal is to minimize the overall cost while satisfying customer demand and adhering to operational constraints.

Dataset:

Data Tables Description:

1. **OrderList:**
    - **Columns:** Order ID, Product ID, Customer ID, Destination Port, Unit Quantity, Unit Weight, Historical Route, Historical Cost
    - **Description:** Contains historical records of orders, their respective routing, and costs.
2. **FreightRates:**
    - **Columns:** Courier ID, Source, Destination, Weight Gap, Rate, tpt_day_cnt
    - **Description:** Lists available couriers, their rates based on weight gaps for specific lanes, and estimated shipping times.
3. **PlantPorts:**
    - **Columns:** Warehouse ID, Port ID
    - **Description:** Specifies allowed links between warehouses and shipping ports.

4. **ProductsPerPlant:**
    - **Columns:** Warehouse ID, Product ID
    - **Description:** Lists supported warehouse-product combinations.
5. **VmiCustomers:**
    - **Columns:** Customer ID, Warehouse ID
    - **Description:** Lists special cases where a warehouse is only allowed to support specific customers.
6. **WhCapacities:**
    - **Columns:** Warehouse ID, Capacity (Orders per day)
    - **Description:** Specifies the maximum number of orders that can be processed by each warehouse per day.
7. **WhCosts:**
    - **Columns:** Warehouse ID, Product ID, Cost per Unit
    - **Description:** Lists the cost associated with storing products in each warehouse.

Problem Constraints:

- **Warehouse Capacities:** Each warehouse has a maximum capacity of orders it can handle per day (WhCapacities).
- **Product-Warehouse Compatibility:** Only specific products can be stored in specific warehouses (ProductsPerPlant).
- **Customer-Warehouse Restrictions:** Some customers can only be served by specific warehouses (VmiCustomers).
- **Freight Costs:** Transportation costs depend on the weight of the shipment and the distance between source and destination (FreightRates).
- **Transportation Time:** Estimated shipping time is considered for optimal route planning (tpt_day_cnt in FreightRates).
- **Storage Costs:** Costs associated with storing products in warehouses (WhCosts).

Analysis Tasks:

1. **Historical Cost Calculation:**
    - Calculate the total historical cost of order fulfillment using data from OrderList and WhCosts.
2. **Capacity Utilization:**
    - Analyze the historical capacity utilization of each warehouse using WhCapacities and OrderList.
3. **Freight Cost Analysis:**
    - Calculate and analyze the freight costs based on historical routes using FreightRates.
4. **Optimization Model:**
    - Develop a linear programming (LP) model to optimize order routing under current constraints (WhCapacities, ProductsPerPlant, VmiCustomers, FreightRates, WhCosts).
    - Objective: Minimize total cost (storage + freight) while fulfilling all orders and adhering to warehouse capacities.
5. **Scenario Analysis:**
    - Perform scenario analysis to assess the impact of changes in warehouse capacities, freight rates, and product compatibility on overall costs and order fulfillment efficiency.
6. **Visualization:**
    - Create visualizations to present capacity utilization, cost distribution (storage vs. freight), and optimal routing solutions.

Deliverables:

1. **Visualization Dashboard:**
    - Charts and graphs displaying key metrics such as total cost, capacity utilization, and optimal routing solutions.

2. **Summary Report:**
    - Executive summary of findings and recommendations.
    - Detailed analysis of historical costs, capacity utilization, and optimization results.
    - Insights from scenario analysis and potential cost-saving strategies.
    -

Project Evaluation metrics:

- Project evaluation will be done in the live session and have to showcase the approaches done to complete the project
- You are supposed to write code in a modular fashion (in functional blocks) Maintainable: It can be maintained, even as your codebase grows.
- Portable: It works the same in every environment (operating system) You have to maintain your code on GitHub.(Mandatory)
- You have to keep your GitHub repo public so that anyone can check yourcode.(Mandatory)
- Proper readme file you have to maintain for any project development(Mandatory)
- Follow the coding standards: https://www.python.org/dev/peps/pep-0008/You should
- include basic workflow and execution of the entire project in the readme file on GitHub