Transportation Route Optimization for Supply Chain Management

---

Objective:

The primary goal of this project was to optimize the transportation routes for delivery vehicles, thereby reducing overall transportation costs, improving fuel efficiency, and enhancing delivery times. The focus was on solving the **Traveling Salesman Problem (TSP)** with added complexity such as simulating traffic congestion, which directly impacts travel time and cost. The project aimed to provide an efficient solution for businesses to manage and optimize their delivery logistics through data analysis and optimization algorithms.

Abstract:

In today's fast-paced world, logistics and supply chain management are integral parts of any business. Efficient route planning can significantly reduce costs and improve customer satisfaction. This project aims to optimize the transportation routes for a set of delivery points by considering various factors like distance, traffic congestion, and delivery time. Using the **Traveling Salesman Problem (TSP)** approach and incorporating simulations for traffic congestion, the project computes the shortest and most efficient delivery routes. The results are visualized on a map and analyzed for potential cost savings and operational improvements.

Methodology:

1. **Data Collection and Preprocessing:**
   - The first step involved collecting geospatial data (latitude, longitude) for delivery points. This data was used to plot locations on a map.
   - Traffic congestion data was simulated to reflect different travel conditions (peak vs. off-peak hours), which influences the total travel time.
   - The data was cleaned and preprocessed to remove outliers and handle missing values.
2. **Traffic Congestion Simulation:**
   - Traffic congestion was simulated using predefined congestion levels that were factored into the travel time between points. The congestion factor represented the delay during peak traffic hours and was used to adjust the weight (cost) of each edge in the graph.
3. **Route Optimization Using TSP:**

- The **Traveling Salesman Problem (TSP)** was applied to find the optimal route for visiting all delivery points. The optimization considers the total distance and travel time, taking into account the simulated traffic congestion.
- The **networkx** library in Python was used to construct a graph, where nodes represent delivery points, and edges represent travel times (adjusted for congestion).
- The TSP was solved using the **approximation algorithm** provided by networkx, which provides a near-optimal solution to the problem.

4. **Optimization Algorithm:**
   - The traveling_salesman_problem function from the **networkx** library was used to compute the optimal route, considering the edge weights as the travel times between delivery points.
   - The TSP algorithm used a heuristic to find a near-optimal solution efficiently, as the exact solution for large datasets can be computationally expensive.

Code for TSP Optimaization:

```
import networkx as nx


# Create a graph

G = nx.Graph()


# Add nodes (delivery points)

G.add_nodes_from(delivery_points)


# Add edges with weights (travel time)

for i in range(len(delivery_points)):

    for j in range(i + 1, len(delivery_points)):

        weight = calculate_travel_time(delivery_points[i], delivery_points[j], traffic_conditions)

        G.add_edge(delivery_points[i], delivery_points[j], weight=weight)
```

Apply TSP approximation

optimized_route = nx.approximation.traveling_salesman_problem(G, cycle=False, weight='weight')

1. **Visualization and Analysis:**
   - The optimized route was plotted on a map using **folium** to provide a visual representation of the delivery route.
   - The route was compared with the original, non-optimized route to highlight the efficiency improvements.
   - Key performance metrics, such as total travel time and cost savings, were computed to assess the impact of optimization.
2. **Cost and Time Analysis:**
   - The total cost was calculated based on the optimized travel time and the number of miles traveled. The project also compared these values with the non-optimized route to quantify the savings.

Code Summary:

1. **Importing Libraries and Data:** The following libraries were used to handle data manipulation, optimization, and visualization:
   - pandas for data handling and manipulation.
   - networkx for constructing graphs and solving the TSP.
   - folium for map visualization.
   - matplotlib for visualizing the results.

import pandas as pd

import networkx as nx

import folium

import matplotlib.pyplot as plt

Creating a Graph and Adding Nodes/Edges: Nodes represent delivery points, and edges represent travel time (which is adjusted for traffic congestion). The graph was built using networkx.

```
G = nx.Graph()
G.add_nodes_from(delivery_points)

# Loop to add edges with travel time (accounting for traffic congestion)
for i in range(len(delivery_points)):
    for j in range(i + 1, len(delivery_points)):
        weight = calculate_travel_time(delivery_points[i], delivery_points[j], traffic_conditions)
        G.add_edge(delivery_points[i], delivery_points[j], weight=weight)
```

Applying the TSP Algorithm: The algorithm solves for the most efficient route, minimizing total distance and travel time.

```
optimized_route = nx.approximation.traveling_salesman_problem(G, cycle=False, weight='weight')
```

Visualization: Using **folium**, the optimized route was visualized on a map with delivery points marked.

```
m = folium.Map(location=[latitude, longitude], zoom_start=12)
folium.PolyLine(locations=optimized_route, color='blue').add_to(m)
```

Cost and Time Calculation: Total cost was calculated by considering the fuel cost per mile and the total distance traveled.

```
total_cost = calculate_total_cost(optimized_route, fuel_cost_per_mile)
```

Results:

- **Optimized Route:** The TSP optimization successfully provided the shortest route for all delivery points, minimizing the travel distance while accounting for traffic congestion.
- **Cost Savings:** The optimized route resulted in significant cost savings by reducing fuel consumption and travel time compared to the original, non-optimized route.
- **Time Savings:** The travel time was significantly reduced by eliminating unnecessary detours, especially during peak traffic hours.
- **Visualization:** The optimized route was visualized on a map using folium, clearly showing the improved delivery path.

Challenges and Limitations:

- **Data Quality:** The accuracy of the optimization is directly dependent on the quality of the input data, particularly the traffic congestion data. Inaccurate or outdated data could lead to suboptimal routes.
- **Scalability:** The TSP becomes computationally expensive as the number of delivery points increases. For very large datasets, more advanced algorithms or parallel computing could be considered.
- **Real-Time Data:** Incorporating real-time traffic data could further improve the accuracy of the optimized route by adjusting for current traffic conditions.

Conclusion:

This project demonstrated the application of **Traveling Salesman Problem (TSP)** optimization to improve delivery route efficiency for supply chain management. By simulating traffic congestion and utilizing advanced algorithms, the project successfully minimized transportation costs and reduced delivery times.

- **Optimized Routes:** Using the TSP algorithm, the project provided a near-optimal solution for the delivery route, considering both distance and time.
- **Cost and Time Efficiency:** The optimization led to significant cost and time savings by reducing fuel consumption and improving travel efficiency.
- **Operational Insights:** Businesses can use the methods developed in this project to enhance their logistics operations, providing a scalable solution to route optimization.

The project provides a solid foundation for businesses to adopt optimization techniques that improve both logistics efficiency and customer satisfaction.

Future Work:

- **Integration of Real-Time Data:** Future improvements could include incorporating real-time traffic data to further optimize routes dynamically.
- **Handling Larger Datasets:** For larger datasets, more efficient algorithms or parallel computation methods could be applied.
- **Machine Learning Integration:** Machine learning algorithms could be used to predict traffic patterns and optimize routes further.