
T · · Systems ·
INNOVATIVE SOLUTION FOR THE DIGITAL AGE OF AUTONOMOUS CARS
TRUST SYSTEMS BRO

QUESTIONS:

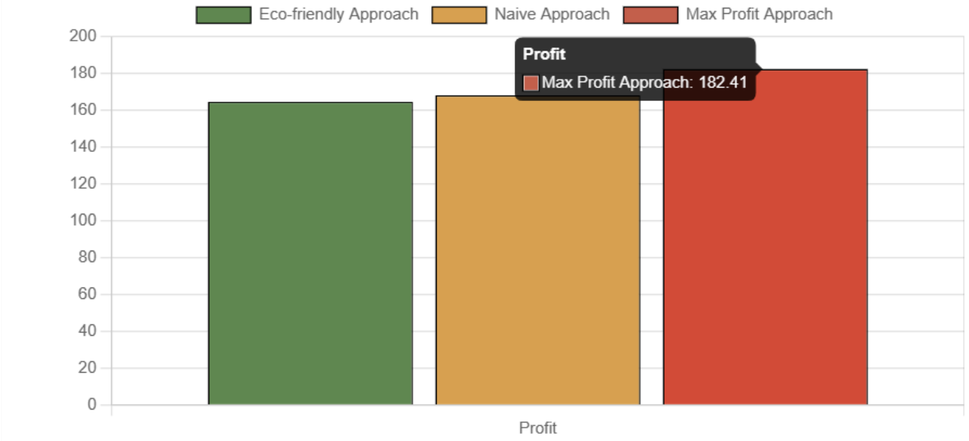
What will the future of data analytics platforms look like for managers?

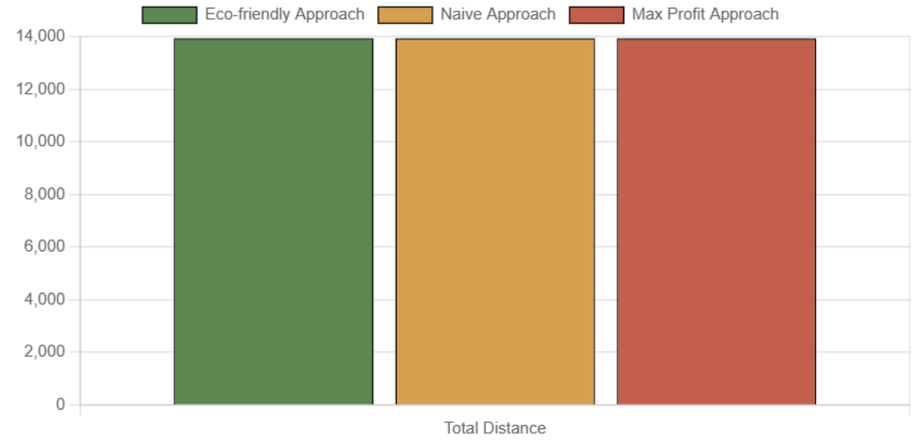
How can we optimize fleet operations for taxis?

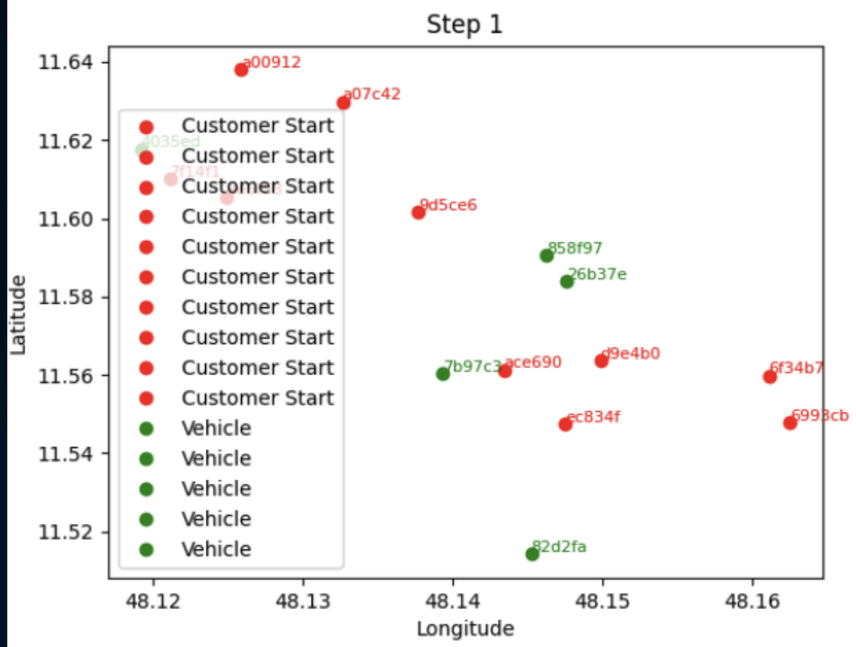
Fleet Management System

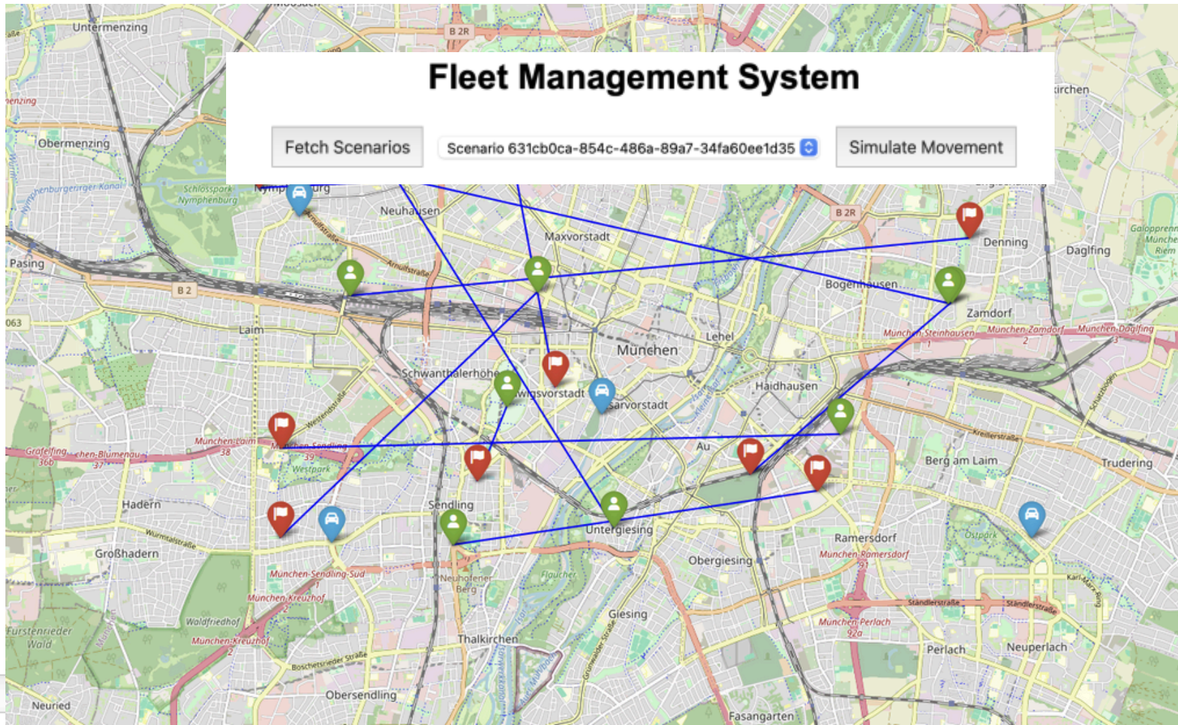
Fetch Scenarios | Scenario 7103b39d-7115-4f07-9be9-b4977b4826ae | Simulate Movement

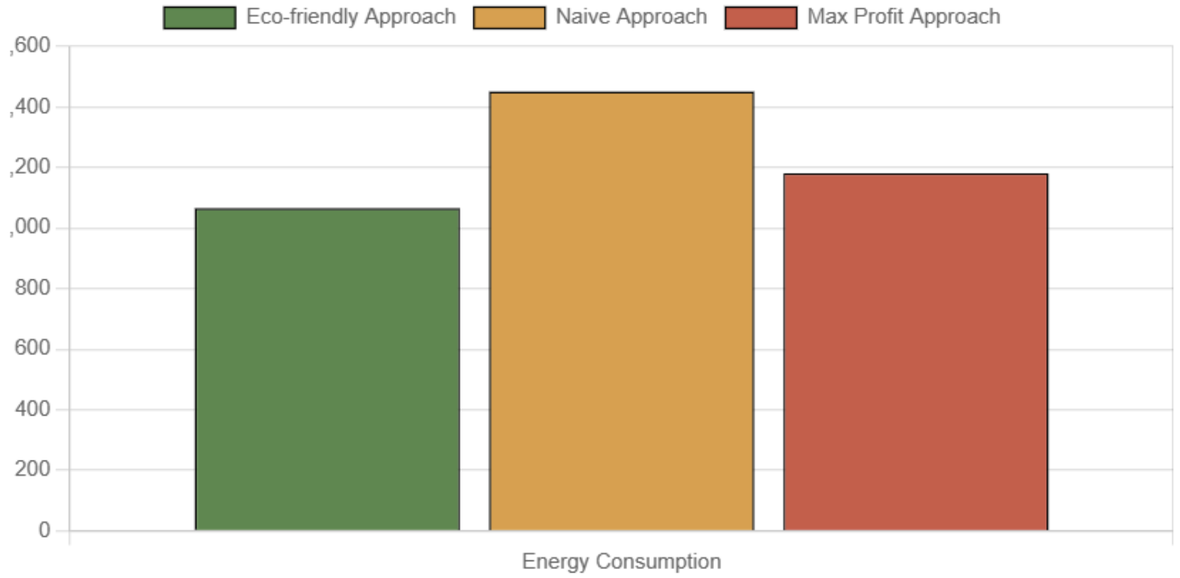
Statistics Dashboard

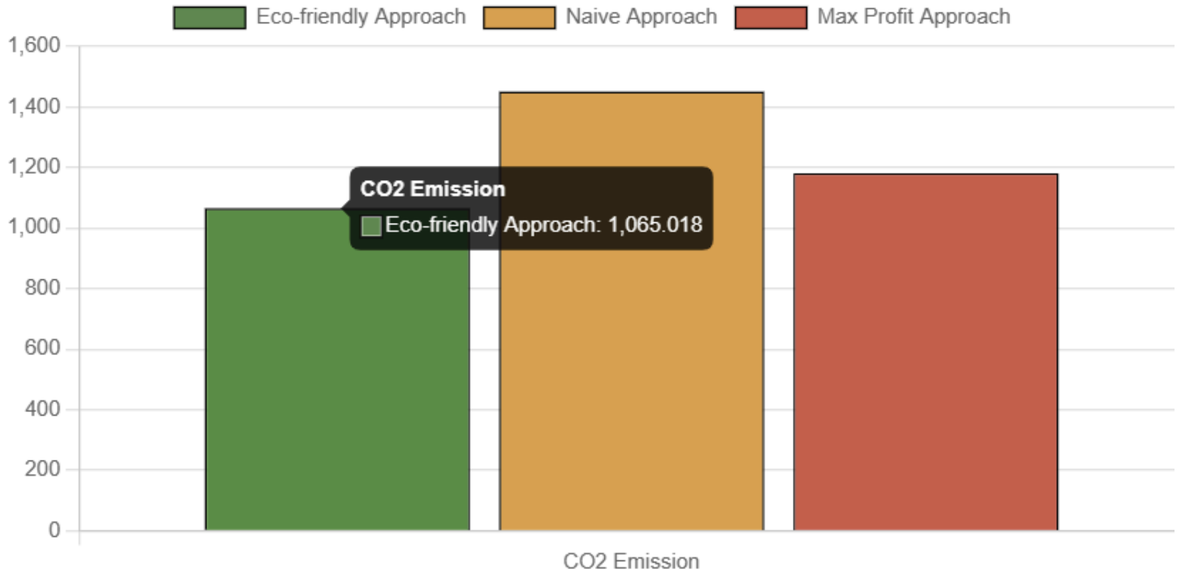








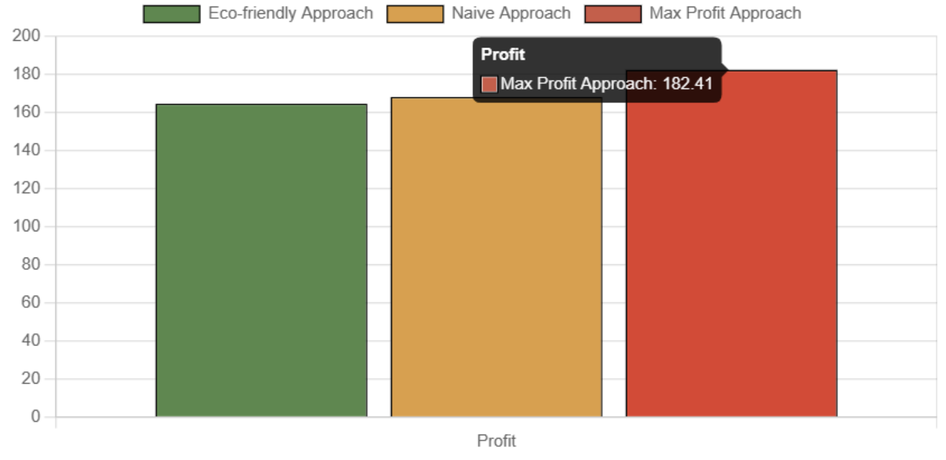


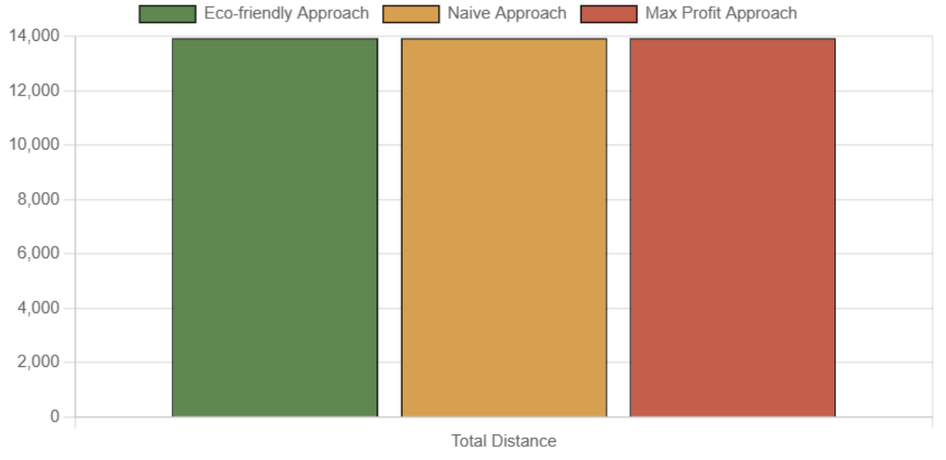


Fleet Management System

Fetch Scenarios | Scenario 7103b39d-7115-4f07-9be9-b4977b4826ae | Simulate Movement

Statistics Dashboard





Decision-Making and Planning Algorithms

01

Shortest Path

- Naive soultion to benchmark against.
- One passenger at a Time.
- Calculating the distance between each and using the shiotest path

02

Algorithmic Optimisation Car Sharing

- Euclidean distance.
- K-means or any other clustering technique to group nearby customers.
- Assigning customers to the nearest vehicle.
- Optimizing vehicle routes using VRP techniques.

03

Multi-Objective Optimization with Weighted Paths

- Adjust the route based on factors such as traffic conditions, profit and CO2 emissions. Path have weights that can be adjusted with AI

Shortest Distance
Time Travel (Customer Happines) (Traffic)
C02 Emission
Most Econonomical

When Car Sharing is Better:

- Large Number of Customers: If you have a large number of customers, car sharing can reduce the total number of vehicles required. By grouping customers with similar destinations, you can maximize vehicle usage and reduce the number of vehicles needed.
- Geographically Close Customers: If the customers are geographically close to each other, grouping them into shared rides will likely reduce total travel distance and emissions.
- Environmental Considerations: If minimizing CO_2 emissions is a priority, car sharing will likely provide better results, as it minimizes the total distance driven and ensures that vehicles are carrying the maximum number of passengers.

When Dijkstra's Algorithm Might Be Better:

- Few Customers: If the number of customers is small, the complexity of car sharing may not provide significant benefits over Dijkstra's, which is simpler and easier to implement.
- Customers Are Far Apart: If customers are spread out over large distances, grouping them for shared rides may not be practical, and the vehicles may end up driving inefficient routes that would negate the benefits of shared rides.
- Simpler Problem: If your goal is just to get the shortest path for each vehicle to pick up and drop off customers, Dijkstra's algorithm provides a simple and fast solution. It doesn't require the complexities of car sharing or VRP algorithms

Multi-Objective Optimization with Weighted Paths

Why Is This Approach Better Than Just Dijkstra?

Dijkstra's Limitation:

- Dijkstra's algorithm will give you the shortest path between two points based on distance (or time, if we tweak it).
- But it ignores other factors like CO2 emissions or the global optimization of vehicle assignment and customer groupings. It cannot consider trade-offs between multiple objectives, and it doesn't have the flexibility to minimize CO2 emissions while also reducing travel time.

Real-World Applicability:

- Real-world applications like ride-sharing services, logistics, and transportation require a balance of time, cost, and emissions. The weighted path approach gives the flexibility to address multiple goals simultaneously.
- You can also adjust weights dynamically depending on the business objective at the time (e.g., if reducing emissions is more critical at certain times, you can increase the weight for CO2).

. Multi-Objective Routing (Weighted Score):

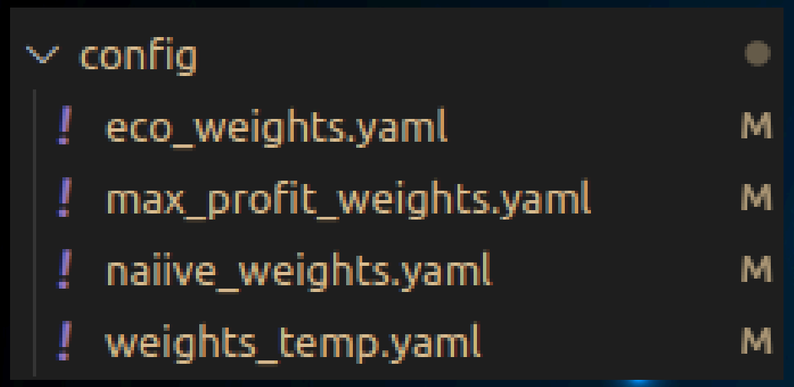
- Multi-objective optimization (like the weighted path approach) allows us to consider multiple factors simultaneously (time, distance, CO2).
- By introducing weights based on the importance of each factor, you can fine-tune the system to prioritize certain objectives over others, such as prioritizing reduced CO2 emissions while balancing travel time.

- **Min-Max Normalization**:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

Scales all metrics to a range of [0, 1].

Can it get even better.....



2. **Score Calculation**:

- Combine normalized metrics into a weighted score:

$$\text{score} = \sum_i w_i \cdot \text{metric}'_i$$

Multiply each normalized metric by its corresponding weight.

Use ML to Optimize Weights

Data-Driven Optimization:

- In traditional optimization, you have to manually tune the weights, which is often trial and error. ML, on the other hand, can automatically adjust weights based on historical data, making it possible to find the optimal trade-off between time, distance, and CO2 emissions.
- For example, the system can learn from past data about how much impact distance has on CO2 emissions, and adjust the weight dynamically to reduce emissions during certain times of the day when CO2 is a more significant concern.

Dynamic Adaptation:

- Traffic conditions, weather, and customer demand fluctuate over time. ML models can continuously adapt the weights to optimize for the most relevant objective. For example, if traffic congestion increases, time may become a higher priority, while in times of low traffic, CO2 emissions may be more critical.
- By learning from historical patterns, the model can predict when and where certain objectives should be prioritized more heavily.

Better Decision-Making:

- The system can learn which objectives lead to the best outcomes under real-world conditions. For example, it may find that minimizing time for specific routes results in fewer overall vehicles or less CO2 output in the long run, and can adjust the weight accordingly.
- Reinforcement learning (RL) can be applied where an agent learns by interacting with its environment (i.e., the vehicle routing problem) and gets feedback on its actions, helping it find the best strategy over time.