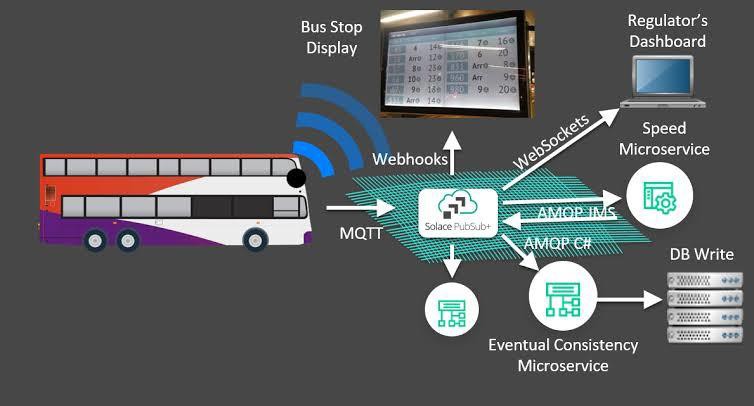
# Phase 3 submission Document

### Project Title : public Transport And Optimization

**Phase 3:** *Development Part 1*

**Topic :** In this section begin building your project by loading and preprocessing the dataset.



# Introduction:

* Public transportation systems are the lifeblood of urban centers, connecting people to their destinations and reducing traffic congestion and pollution. However, the ever-increasing demands for efficiency, reliability, and sustainability have brought about new challenges that demand innovative solutions. In this era of digital transformation, the integration of Internet of

Things (IoT) technology into public transport systems is revolutionizing the way we conceive, operate, and experience public transportation.

* The convergence of IoT and public transport has opened up a world of possibilities, ushering in an era of smarter and more user-centric mobility solutions. It empowers transportation authorities, service providers, and passengers alike with real-time data and intelligent algorithms to make informed decisions, improve operations, and enhance the passenger experience.
* This document delves into the realm of Public Transport Optimization in IoT, exploring the multifaceted applications, benefits, and challenges of this transformative fusion. We will examine how IoT sensors and data-driven insights are reshaping public transport by enabling dynamic route optimization, predictive maintenance, real-time passenger information systems, and much more. Furthermore, it will elucidate how this evolution is contributing to a more sustainable and environmentally conscious future, while making public transport a more attractive and convenient choice for commuters.
* The chapters that follow will take you on a journey through the various facets of this technological marvel, providing a comprehensive understanding of the role IoT plays in redefining public transportation. Whether you are a transportation authority seeking cost-effective solutions, a technology enthusiast intrigued by the power of IoT, or a passenger looking for a more seamless and enjoyable journey, this exploration of Public Transport Optimization in IoT promises insights and inspiration for all.
* Join us as we embark on this enlightening journey into the world of Public Transport Optimization in the Internet of Things, where the future of urban mobility begins to unfold.

Road transportation is a critical component of supply chain operations as it represents a significant cost for companies.

With the increase in **diesel prices** and the ongoing **pressure to reduce CO2** emissions, there is a growing need for **transportation optimization**.

Fortunately, data analytics technologies are enabling businesses to improve transportation networks, reduce their environmental footprint, and enhance their bottom line.

In this article, we will explore how to build **visualizations of road transportation network performance** using Python.

In the next sections, you can find insights on how to

* + process and analyze transportation records
  + improve visibility into current routing and truck loading rates
  + simulate multiple routing scenarios to estimate the impact on the average cost per ton

Introduction

Following the series of [Introduction](https://towardsdatascience.com/optimizing-warehouse-operations-with-python-part-1-83d02d001845)

[Following the series of Warehousing Operations Optimization, we will use the same](https://towardsdatascience.com/optimizing-warehouse-operations-with-python-part-1-83d02d001845) [methodology for improving Road Transportation efficiency by](https://towardsdatascience.com/optimizing-warehouse-operations-with-python-part-1-83d02d001845)

[Processing Data: extract unstructured transportation records and process them to build](https://towardsdatascience.com/optimizing-warehouse-operations-with-python-part-1-83d02d001845) [your optimization model](https://towardsdatascience.com/optimizing-warehouse-operations-with-python-part-1-83d02d001845)

[Improving Visibility: using Python visualization libraries to get clarity on current](https://towardsdatascience.com/optimizing-warehouse-operations-with-python-part-1-83d02d001845) [routing and truck loading rate](https://towardsdatascience.com/optimizing-warehouse-operations-with-python-part-1-83d02d001845)

[Simulating Scenarios: build a model to simulate multiple routing scenarios and](https://towardsdatascience.com/optimizing-warehouse-operations-with-python-part-1-83d02d001845) [estimate the impact on average cost per ton](https://towardsdatascience.com/optimizing-warehouse-operations-with-python-part-1-83d02d001845), we will use the same methodology for improving **Road Transportation** efficiency by

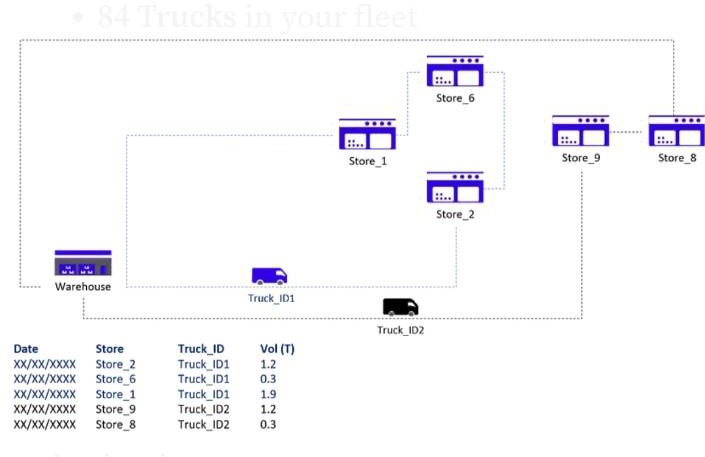
1. **Processing Data:** extract unstructured transportation records and process them to build your optimization model
2. **Improving Visibility:** using Python visualization libraries to get clarity on current routing and **truck loading rate**
3. **Simulating Scenarios:** build a model to simulate multiple routing scenarios and estimate the impact on **average cost per ton**

## How do you make a transport plan with Python?

* 1. Problem Statement

Retail Stores Distribution with Full Truck Load

* + - **1 Warehouse** delivering stores by using **three** types of Trucks (3.5T, 5T, 8T)
    - **49 Stores** delivered
    - **12 Months** of Historical Data with **10,000 Deliveries**
    - **7 days** a week of Operations
    - **23 Cities**
    - **84 Trucks** in your fleet



* 1. Objective: Reduce the Cost per Ton

Method: Shipment Consolidation

In this scenario, you are using 3rd party carriers that charge full trucks per destination:

Transportation Costs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **City\_En** | **3.5T (Rmb)** | **5T (Rmb)** | **8T (Rmb)** | **3.5T (Rmb/Ton)** | **5T (Rmb/Ton)** | **8T (Rmb/Ton)** |
| City\_1 | 485 | 650 | 800 | 139 | 130 | 100 |
| City\_2 | 640 | 700 | 820 | 183 | 140 | 103 |
| City\_3 | 690 | 780 | 890 | 197 | 156 | 111 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **City\_En** | **3.5T (Rmb)** | **5T (Rmb)** | **8T (Rmb)** | **3.5T (Rmb/Ton)** | **5T (Rmb/Ton)** | **8T (Rmb/Ton)** |
| City\_4 | 810 | 1,000 | 1,150 | 231 | 200 | 144 |
| City\_5 | 1,300 | 1,568 | 1,723 | 371 | 314 | 215 |
| City\_6 | 1,498 | 1,900 | 2,100 | 428 | 380 | 263 |
| City\_7 | 980 | 1,250 | 1,450 | 280 | 250 | 181 |
| City\_8 | 1,350 | 1,450 | 1,500 | 386 | 290 | 188 |
| City\_9 | 1,350 | 1,450 | 1,500 | 386 | 290 | 188 |
| City\_10 | 850 | 1,000 | 1,200 | 243 | 200 | 150 |

The table above shows rates applied by carriers for each city delivered for each type of truck. Observing **costs per ton are lower for larger trucks**, one lever of improvement is **maximizing shipments consolidation when building routes**.

Thus, the Route **Transportation Planning Optimization** main target will be to cover a maximum number of stores per route.

## Data Processing: Understand the Current Situation

* 1. Import Datasets

Before starting to think about the Optimization Model , your priority is to understand the current situation.

Starting with unstructured data coming from several sources, we’ll need to build a set of data frames to model our network and provide visibility on the loading rate and list of stores delivered for each

**Records of Deliveries per Store**

Deliveries Records

Delveries\_record\_ccv

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Date** | **Truck\_ID** | **Store\_ID** | **FTL** | **Order** | **BOX** | **SKU** | **Loading (Tons)** |
| 9/1/2016 | Truck\_ID1 | Store\_ID1 | 3.5 | 16 | 311 | 83 | 2.404 |
| 9/1/2016 | Truck\_ID1 | Store\_ID2 | 3.5 | 18 | 178 | 83 | 1.668 |
| 9/1/2016 | Truck\_ID2 | Store\_ID3 | 3.5 | 10 | 74 | 54 | 0.81 |
| 9/1/2016 | Truck\_ID2 | Store\_ID4 | 3.5 | 19 | 216 | 88 | 2.413 |
| 9/1/2016 | Truck\_ID3 | Store\_ID5 | 3.5 | 10 | 117 | 54 | 1.119 |
| 9/1/2016 | Truck\_ID3 | Store\_ID6 | 3.5 | 15 | 294 | 92 | 2.962 |
| 9/1/2016 | Truck\_ID4 | Store\_ID7 | 3.5 | 5 | 42 | 19 | 0.421 |
| 9/1/2016 | Truck\_ID4 | Store\_ID8 | 3.5 | 12 | 125 | 88 | 1.138 |
| 9/1/2016 | Truck\_ID5 | Store\_ID9 | 5 | 18 | 201 | 95 | 2.19 |

**Store Address** Store\_address.csv Search this file…

Code city Long Lat address

|  |  |  |  |
| --- | --- | --- | --- |
| Store\_ID1 | City\_Store1 31.952792 | 118.8192708 | Address\_1 |
| Store\_ID2 | City\_Store2 31.952792 | 118.8192718 | Address\_2 |
| Store\_ID3 | City\_Store3 31.675948 | 120.7468221 | Address\_3 |
| Store\_ID4 | City\_Store4 31.664448 | 120.7700006 | Address\_4 |
| Store\_ID5 | City\_Store5 31.750971 | 119.9478857 | Address\_5 |
| Store\_ID6 | City\_Store6 31.791351 | 119.9232302 | Address\_6 |
| Store\_ID7 | City\_Store7 31.79233 | 119.9768294 | Address\_7 |
| Store\_ID8 | City\_Store8 31.982972 | 119.5832084 | Address\_8 |
| Store\_ID9 | City\_Store9 31.996161 | 119.6341775 | Address\_9 |
| Store\_ID10 | City\_Store10 31.885547 | 121.1886473 | Address\_10 |
| Store\_ID11 | City\_Store11 30.310079 | 120.1515734 | Address\_11 |
| Store\_ID12 | City\_Store12 31.383616 | 121.2569408 | Address\_12 |
| Store\_ID13 | City\_Store13 31.387863 | 121.2797154 | Address\_13 |

**Transportation Costs** Transportation\_cost.csv Search this file…

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| City\_En | 3.5T (Rmb) 5T (Rmb) | | | | 8T (Rmb) | | 3.5T (Rmb/Ton) | 5T |
| (Rmb/Ton) | 8T (Rmb/Ton) | | | |  | |  |  |
| City\_1 | 485 | 650 | 800 | 139 | 130 | 100 | | |
| City\_2 | 640 | 700 | 820 | 183 | 140 | 103 | | |
| City\_3 | 690 | 780 | 890 | 197 | 156 | 111 | | |
| City\_4 | 810 | 1,000 1,150 231 | | | 200 | 144 | | |
| City\_5 | 1,300 | 1,568 1,723 371 | | | 314 | 215 | | |
| City\_6 | 1,498 | 1,900 2,100 428 | | | 380 | 263 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| City\_7 | 980 | 1,250 1,450 280 | 250 | 181 |
| City\_8 | 1,350 | 1,450 1,500 386 | 290 | 188 |
| City\_9 | 1,350 | 1,450 1,500 386 | 290 | 188 |
| City\_10 | 850 | 1,000 1,200 243 | 200 | 150 |

* 1. Listing of stores delivered by each route

Let us process the initial data frame to list all stores delivered for each route.

### 1 Route = 1 Truck ID + 1 Date

# Create Transport Plan

Def transport\_plan(data, dict\_trucks, capacity\_dict):

# List of Stores per Truck for each DAY Df\_plan = pd.DataFrame(data.groupby([‘Date’,

‘TruckID’])[‘Code’].apply(list)) Df\_plan.columns = [‘List\_Code’] # List of Box Quantity

Df\_plan[‘List\_BOX’] = data.groupby([‘Date’, ‘TruckID’])[‘BOX’].apply(list)

# Mean of FTL

Df\_plan[‘FTL’] = data.groupby([‘Date’, ‘TruckID’])[‘FTL’].mean() Df\_plan[‘Capacity(T)’] = df\_plan[‘FTL’].map(capacity\_dict)

Df\_plan[‘List\_Loading’] = data.groupby([‘Date’, ‘TruckID’])[‘Loading(T)’].apply(list)

Df\_plan[‘Count’] = df\_plan[‘List\_Loading’].apply(lambda t: len(t)) Df\_plan[‘Total\_tons(T)’] = data.groupby([‘Date’,

‘TruckID’])[‘Loading(T)’].sum()

# Distribute: one shipment per col

# Stores

D = df\_plan[‘List\_Code’].apply(pd.Series) For col in d:

Df\_plan[“Store%d” % (col+1)] = d[col] # Boxes number

D = df\_plan[‘List\_BOX’].apply(pd.Series) For col in d:

Df\_plan[“Box%d” % (col+1)] = d[col] # Shipments Tonnage

D = df\_plan[‘List\_Loading’].apply(pd.Series) For col in d:

Df\_plan[“Tons%d” % (col+1)] = d[col]

# Fill NaN + Drop useless columns Df\_plan.fillna(0, inplace = True)

If 1 == 0:

Df\_plan.drop([‘List\_Code’], axis = 1, inplace = True) Df\_plan.drop([‘List\_BOX’], axis = 1, inplace = True) Df\_plan.drop([‘List\_Loading’], axis = 1, inplace = True)

Return df\_plan **Example Transport Plan** Transport\_plan.csv Search this file…

Date TruckID List\_Code Capacity(T) List\_Loading Count Total\_tons(T) Store1 Store2 Store3 Store4 Box1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Box2 | Box3 Box4 | | Tons1 Tons2 Tons3 Tons4 Occupation(%) | | | | | Available(T) | |
| 9/1/2016 | Truck\_ID1 | | [‘Store\_ID6’] 3.5 [2.91] 1 2.91 | | | | | ID6 0 | |
| 0 | 0 243 | | 0 0 0 2.91 0 0 0 | | | | | 83.14 0.59 | |
| 9/1/2016 | Truck\_ID2 | | [‘Store\_ID34’, ‘Store\_ID22’, ‘Store\_ID9’] | | | | | 3.5 [0.3, | |
| 1.37, 0.47] | 3 | 2.14 | ID34 ID22 ID9 | 0 | 31 | 116 | 44 | 0 | 0.3 |
| 1.37 | 0.47 | 0 | 61.14 1.36 |  |  |  |  |  |  |
| 9/1/2016 | Truck\_ID3 | | [‘Store\_ID18’] | 3.5 | [1.5] | 1 | 1.5 | ID18 | 0 |
| 0 | 0 174 | | 0 0 0 | 1.5 | 0 | 0 | 0 | 42.86 | 2 |
| 9/1/2016 | Truck\_ID4 | | [‘Store\_ID37’] | 3.5 | [2.3] | 1 | 2.3 | ID37 0 | |
| 0 | 0 179 | | 0 0 0 | 2.3 | 0 | 0 | 0 | 65.71 1.2 | |
| 9/1/2016 | Truck\_ID5 | | [‘Store\_ID34’, ‘Store\_ID48’] | | | 3.5 | [2.14, 0.51] 2 | | |
| 2.65 | ID34 ID48 | | 0 0 168 46 0 | | | 0 | 2.14 0.51 0 | | |
| 0 | 75.71 0.85 | |  | | |  |  | | |

# Add cities covered by each route

Let us now calculate Transportation Costs invoiced by carriers for each route: ## Pricing Functions

Def f\_maxcity(list\_cities, list\_price):

Return list\_cities[list\_price.index(max(list\_price))] # Index of Maximum

Price

Def inner\_stops(list\_cities, max\_city): Return list\_cities.count(max\_city) – 1

Def outer\_stops(list\_cities, max\_city):

Return len(list\_cities) – (list\_cities.count(max\_city))

Def total\_price(max\_price, inner\_stops, outer\_stops, inner\_price, outer\_price): Return max\_price + inner\_stops \* inner\_price + outer\_stops \* outer\_price

# Calculate Price

Def plan\_price(df\_strinfo, df\_plan, inner\_price, outer\_price):

# Dictionnary Ville

Dict\_ville = dict(zip(df\_strinfo.Code.values, df\_strinfo.City.values))

# Price per Truck Size： 3.5T, 5T, 8T

Dict\_35, dict\_5, dict\_8 = [dict(zip(df\_strinfo.City.values, df\_strinfo[col].values)) for col in [‘3.5T’, ‘5T’, ‘8T’]]

# Mapping Cities

F\_ville = lambda t: [dict\_ville[i] for I in t] # literal\_eval(t)

# Mapping Price

F\_35 = lambda t: [dict\_35[i] for I in t] F\_5 = lambda t: [dict\_5[i] for I in t] F\_8 = lambda t: [dict\_8[i] for I in t]

# Mapping Price

Df\_plan[‘List\_City’] = df\_plan[‘List\_Code’].map(f\_ville) Df\_plan[‘List\_Price35’] = df\_plan[‘List\_City’].map(f\_35) Df\_plan[‘List\_Price5’] = df\_plan[‘List\_City’].map(f\_5) Df\_plan[‘List\_Price8’] = df\_plan[‘List\_City’].map(f\_8)

# Maximum Price City

F\_maxprice = lambda t: max(t) # Maximum Price

# Mapping First City

Df\_plan[‘Max\_Price35’] = df\_plan[‘List\_Price35’].map(f\_maxprice) Df\_plan[‘Max\_Price5’] = df\_plan[‘List\_Price5’].map(f\_maxprice) Df\_plan[‘Max\_Price8’] = df\_plan[‘List\_Price8’].map(f\_maxprice)

Df\_plan[‘Max\_City’] = df\_plan.apply(lambda x: f\_maxcity(x.List\_City, x.List\_Price35), axis = 1)

# Inner City Stop

Df\_plan[‘Inner\_Stops’] = df\_plan.apply(lambda x: inner\_stops(x.List\_City, x.Max\_City), axis = 1)

Df\_plan[‘Outer\_Stops’] = df\_plan.apply(lambda x: outer\_stops(x.List\_City, x.Max\_City), axis = 1)

# Total Price

Df\_plan[‘Price35’] = df\_plan.apply(lambda x: total\_price(x.Max\_Price35, x.Inner\_Stops, x.Outer\_Stops,

Inner\_price, outer\_price), axis = 1)

Df\_plan[‘Price5’] = df\_plan.apply(lambda x: total\_price(x.Max\_Price5, x.Inner\_Stops, x.Outer\_Stops,

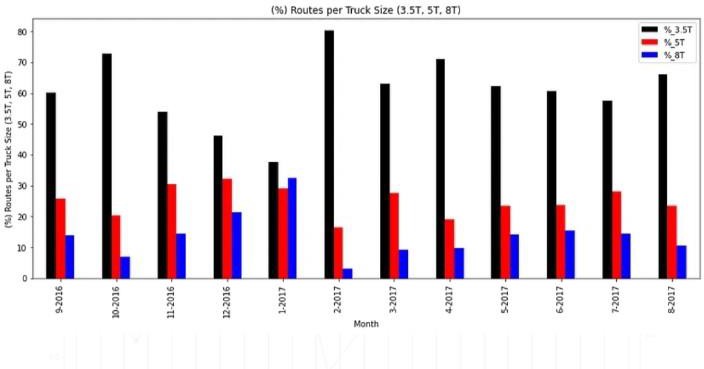
Inner\_price, outer\_price), axis = 1)

Df\_plan[‘Price8’] = df\_plan.apply(lambda x: total\_price(x.Max\_Price8, x.Inner\_Stops, x.Outer\_Stops,

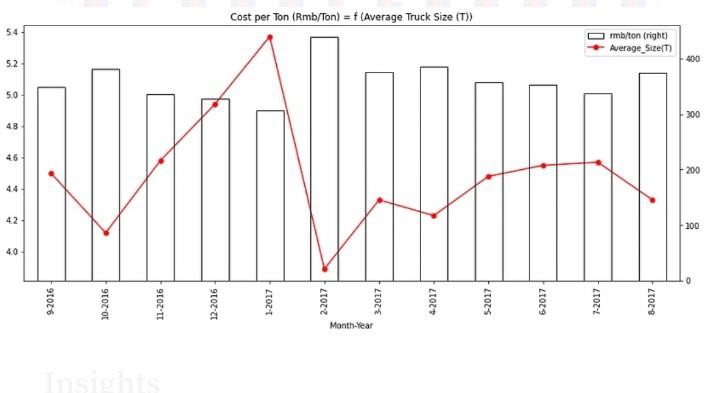
Inner\_price, outer\_price), axis = 1)

Return df\_plan

### Visualization: % Deliveries per Truck Size



(%) of Route per Truck Size (3.5T, 5T, 8T) — (Image by Author)



Impact of Average Truck Size (Ton) on Overall Cost per Ton (Rmb/Ton) — (Image by Author)

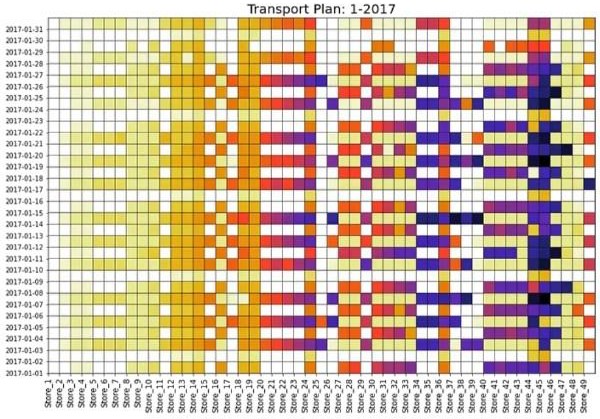
**Insights**

* **Average Truck Size:** a large majority of small trucks
* **Cost per ton:** the inverse proportion of cost per ton and average truck size

### Understand Current Situation: Visualisation

1. Transportation Plan Visualisation

Objective: Get a simple visualisation of all deliveries per day with a focus on the number of different routes.



Transportation Plan: January 2017 — (Image by Author)

**Solution**: Python’s Matplotlib grid function

* **Columns:** 1 Column = 1 Store
* **Rows:** 1 Row = 1 Day
* **Colour = White:** 0 delivery
* **Colours:** 1 Color = 1 Route (1 Truck) **Geographical Visualization of Store Deliveries** Objective

Visualisation of geographical locations delivered in the same route



### Solution

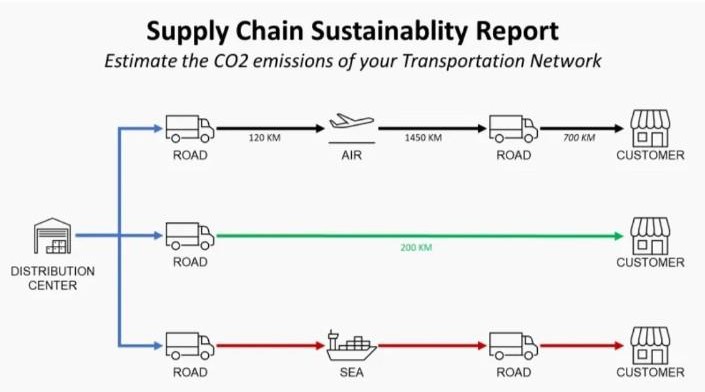
OpenStreet Map + Matplotlib Scatter Plot Visualization of the different routes covered per day



### Next Steps

**1. Measure the Environmental Impact**

In addition to cost reduction, you can also target CO2 Emissions reductions by



Optimizing your Transportation Network.

**Routing Optimization:** Number of Deliveries per Route Dataframe with historical records processed

Current transportation plan

A model to calculate transportation cost per route based on cities delivered Visualisation of the number of different routes per day

Visualisation of geographical locations delivered per Route Next steps are

Routing: increase the number of stores delivered for each route Fleet Allocation: ensure uniform workload distribution

Delivery Frequency: reduce the number of deliveries per week to increase the quantity per shipment

Simulate Impact: savings we can get from optimization listed above

# Conclusion:

In the realm of public transport optimization in the Internet of Things (IoT), the significance of data cannot be overstated. This journey through the amalgamation of smart technology and public transportation has demonstrated the transformative power of data-driven insights. The utilization of IoT datasets has not only redefined public transportation but has also set the stage for a more efficient, reliable, and passenger-centric future.

As we draw this exploration to a close, it’s evident that IoT datasets are the cornerstone upon which all optimization efforts are built. They provide the critical

real-time information needed to make informed decisions and adjustments in a dynamic urban environment. Here, we reflect on the key takeaways:

1. Real-Time Intelligence: IoT datasets enable public transport operators to gain real-time intelligence on vehicle locations, passenger counts, traffic conditions, and more. This information empowers them to adapt swiftly to changing circumstances, ultimately resulting in improved service reliability and passenger satisfaction.
2. Dynamic Route Optimization:Public transport systems can now optimize their routes based on up-to-the-minute data. This leads to reduced travel times, minimized delays, and a more efficient use of resources, which is essential for meeting the growing demands of urban transportation.
3. Predictive Maintenance: IoT datasets support predictive maintenance, ensuring that vehicles are kept in optimal working condition. This proactive approach minimizes downtime, extends the lifespan of vehicles, and enhances safety, which is a win-win for both operators and passengers.
4. . Passenger-Centric Information:Real-time passenger information systems, made possible by IoT datasets, provide travelers with the latest updates on routes, schedules, and vehicle locations. This not only enhances the passenger experience but also encourages more people to choose public transport.
5. Sustainability and Efficiency: With the help of IoT data, public transportation systems can reduce their environmental footprint. They can optimize routes, reduce fuel consumption, and minimize congestion, all of which contribute to a more sustainable and efficient urban transportation ecosystem.

In closing, the fusion of IoT technology and public transport optimization through datasets is a pivotal turning point in urban mobility. The journey doesn’t end here;

it continues to evolve as new technologies and innovations emerge. As we move forward, it is crucial for stakeholders – from transportation authorities to technology providers and passengers – to recognize the immense potential of IoT datasets and work collaboratively to shape a future where public transportation is not just a mode of travel but a seamless, sustainable, and smart way of urban life. The road to public transport optimization in IoT is illuminated by the data it relies on, and the destination is a more connected, efficient, and user-friendly urban world.

