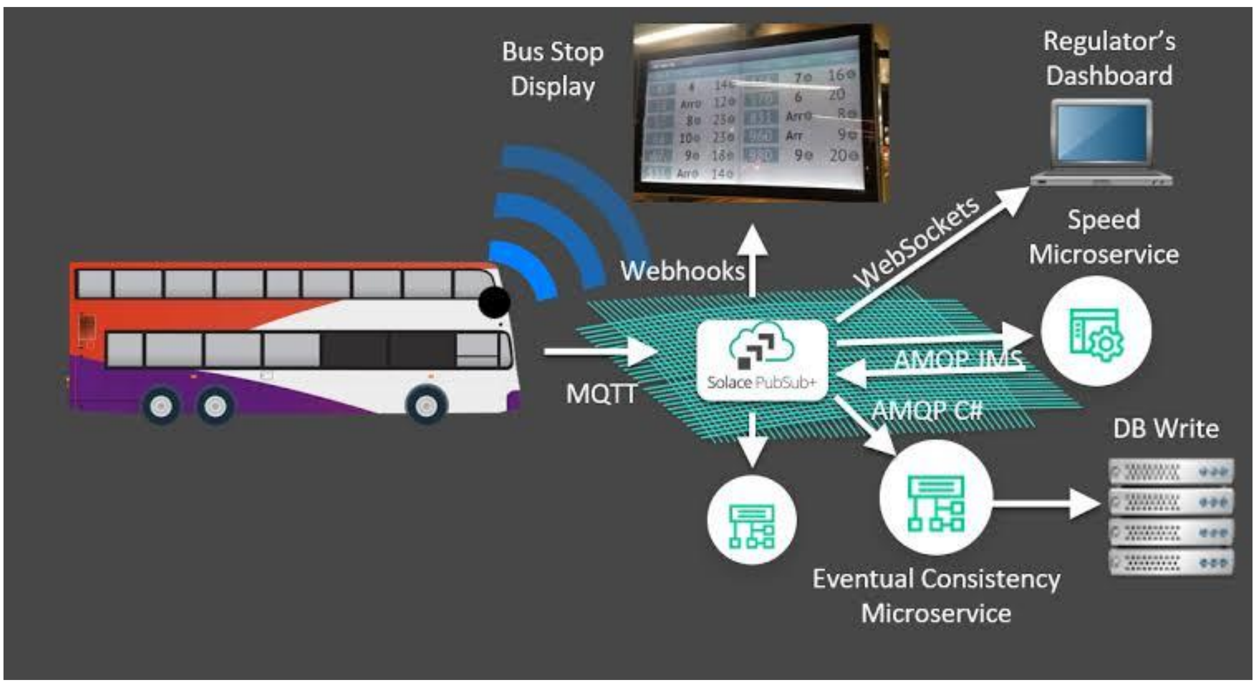
Phase 3 submission Document

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

Following the series of Warehousing Operations Optimization, we will use the same methodology for improving Road Transportation efficiency by

Processing Data: extract unstructured transportation records and process them to build your optimization model

Improving Visibility: using Python visualization libraries to get clarity on current routing and truck loading rate

Simulating Scenarios: build a model to simulate multiple routing scenarios and estimate the impact on average cost per ton, we will use the same methodology for improving **Road Transportation** efficiency by

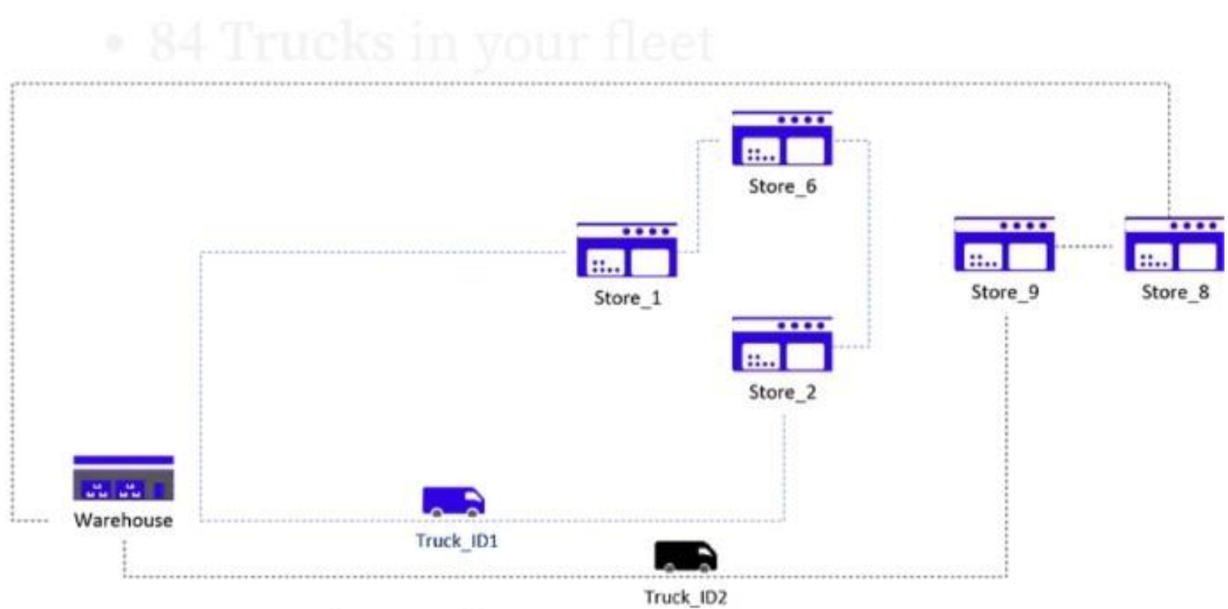
1. **Processing Data:** extract unstructured transportation records and process them to build your optimization model
2. **Improving Visibility:** using Python visualization libraries to get clarity on current routing and **truck loading rate**
3. **Simulating Scenarios:** build a model to simulate multiple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

I. 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



Date	Store	Truck_ID	Vol (T)
XX/XX/XXXX	Store_2	Truck_ID1	1.2
XX/XX/XXXX	Store_6	Truck_ID1	0.3
XX/XX/XXXX	Store_1	Truck_ID1	1.9
XX/XX/XXXX	Store_9	Truck_ID2	1.2
XX/XX/XXXX	Store_8	Truck_ID2	0.3

2. 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.

II. 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

2. 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

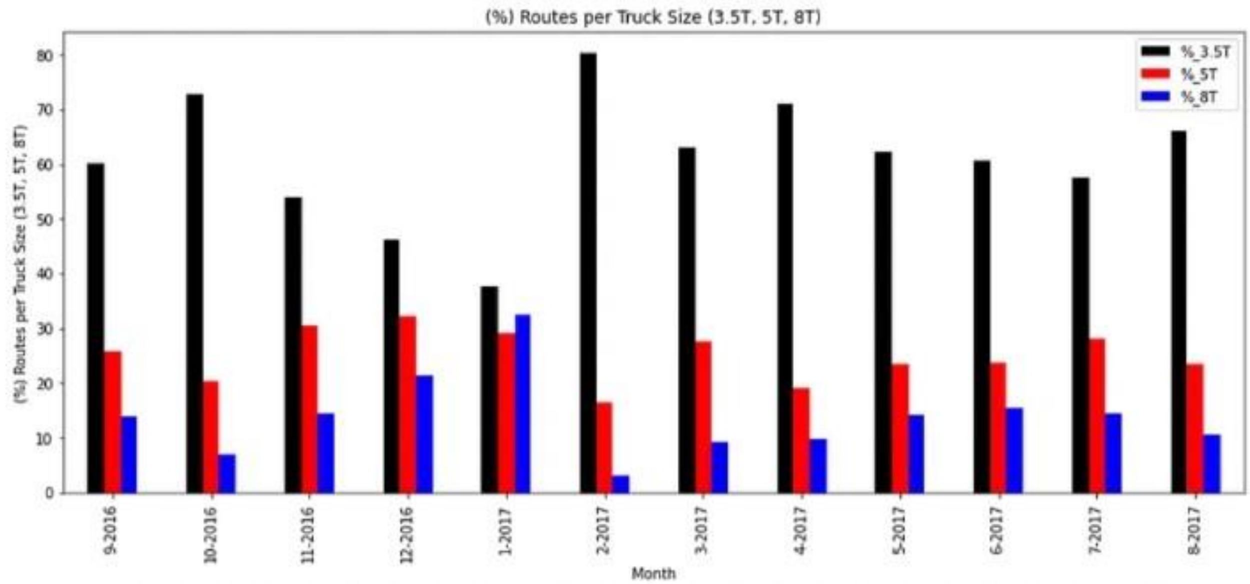
Let us now calculate Transportation Costs invoiced by carriers for each route:

```
## 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
```

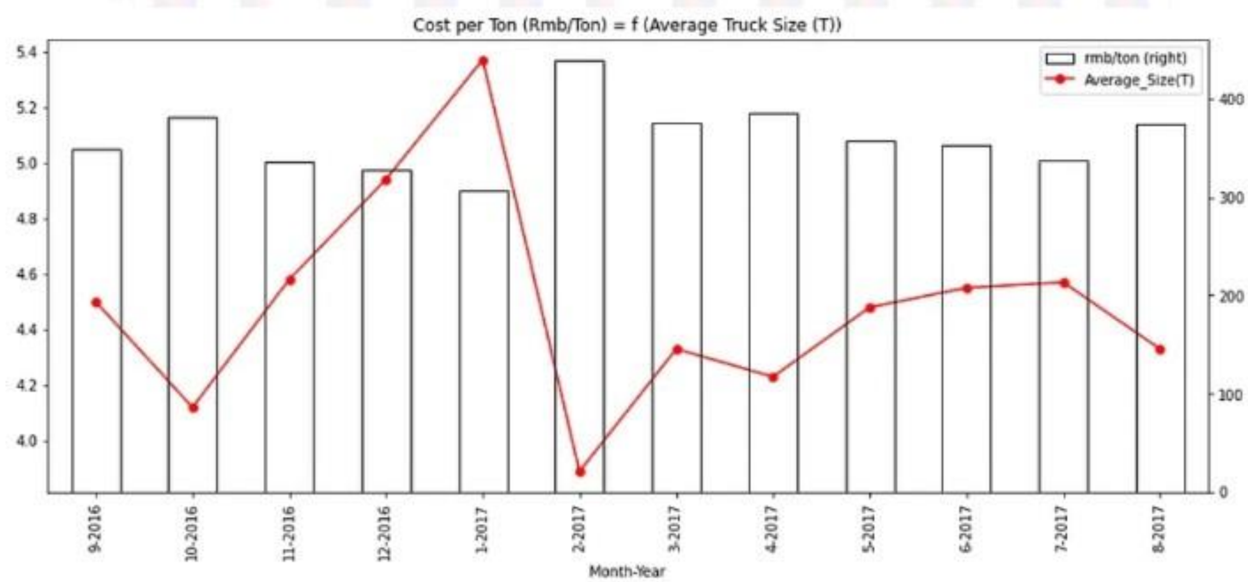
```
# Calculate Price
Def plan_price(df_strinfo, df_plan, inner_price, outer_price):

    # 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)

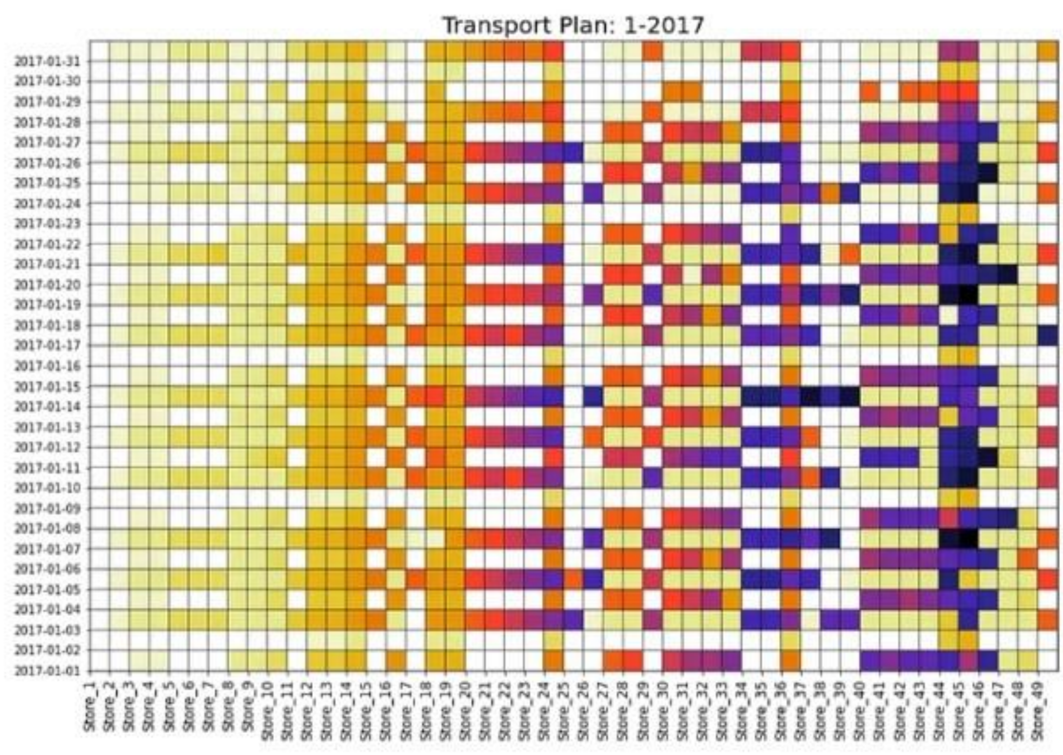
Insights

- **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)

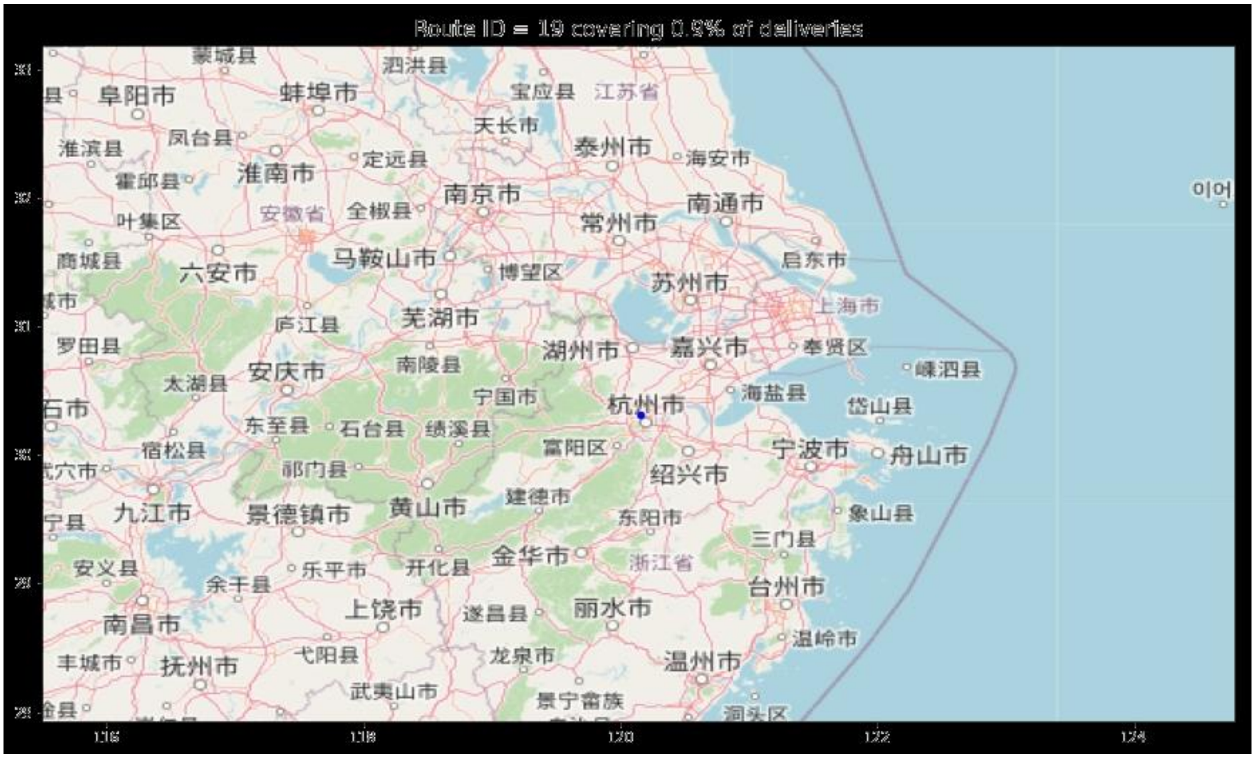
Solution: Python's Matplotlib grid function

- **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

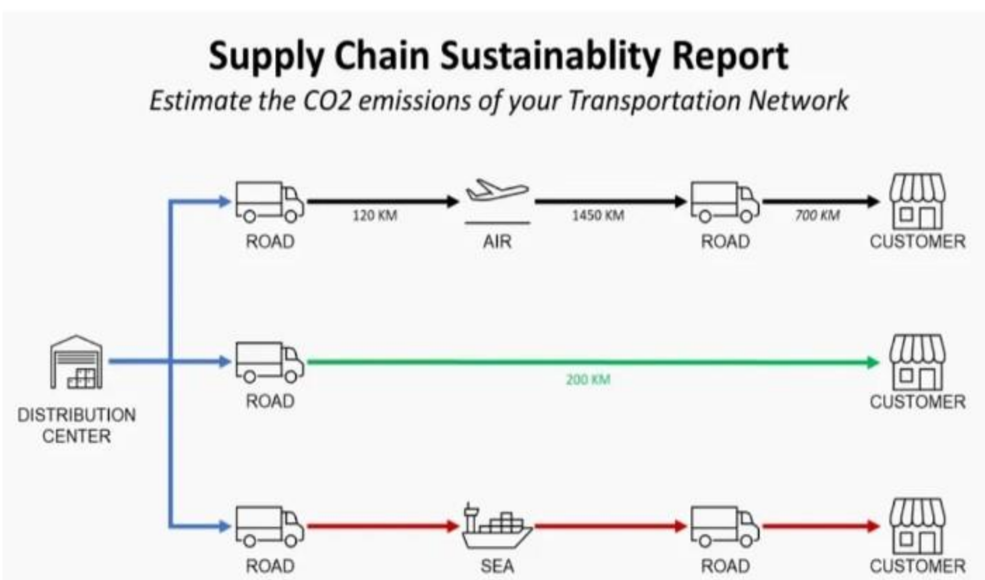
OpenStreet Map + Matplotlib Scatter Plot

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.