

**... I have always found that plans are
useless, but planning is indispensable.**

Dwight David Eisenhower

THE ESSENCE OF THE PROBLEM

The vehicle routing problem (VRP) is a problem in combinatorial optimization, computer science, and integer programming that asks:

“What is the optimal set of routes for a fleet of vehicles to deliver to a given group of customers?”

1830 ● Was mentioned in the German book “**Advice from an Old Traveling Salesman.**”

1930 ● In the 1930s, the problem was first formulated by **Carl Menger**

1954 ● In an article by **George Dantzig** and **John Ramser** in 1959, it was applied to gasoline delivery.

THE PROBLEM: Why is this so important?

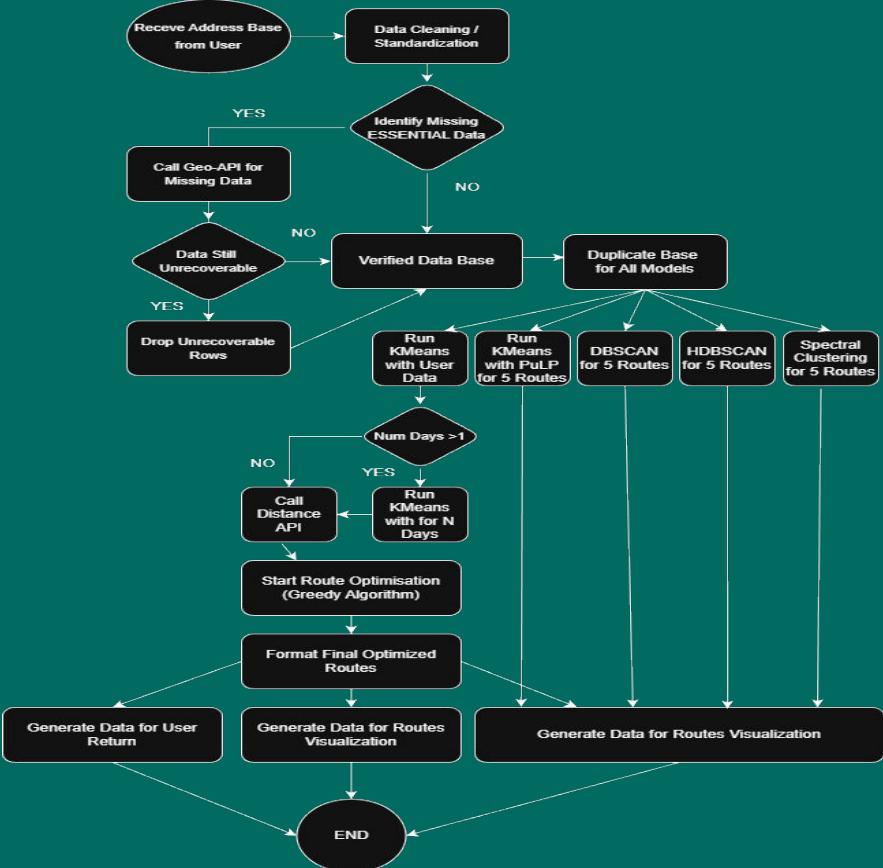


Everything consumers and businesses need is being shipped and transported from one place to another, often this includes different countries



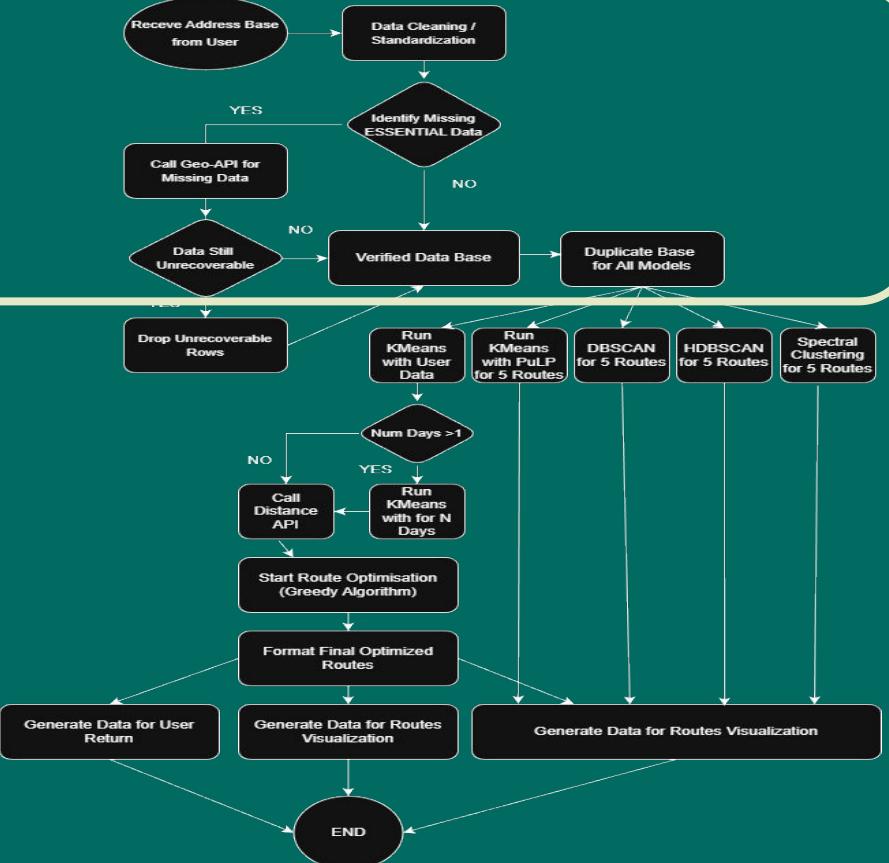
I have built routes for my sales teams dozens of times, and each time this procedure has been a huge headache for me and my colleagues.

THE ALGORITHM & SOLUTIONS



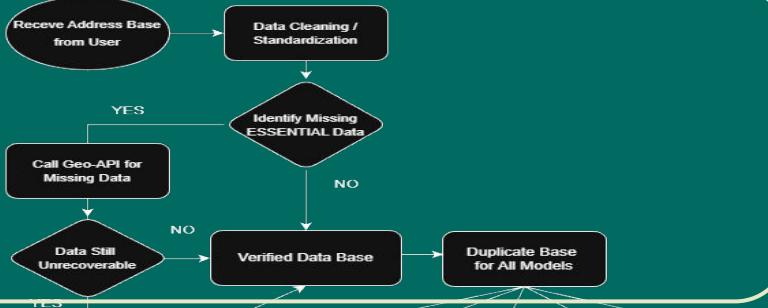
THE ALGORITHM & SOLUTIONS

- Get data from user, clean and check.
- Get missing data using API service
- Clean again

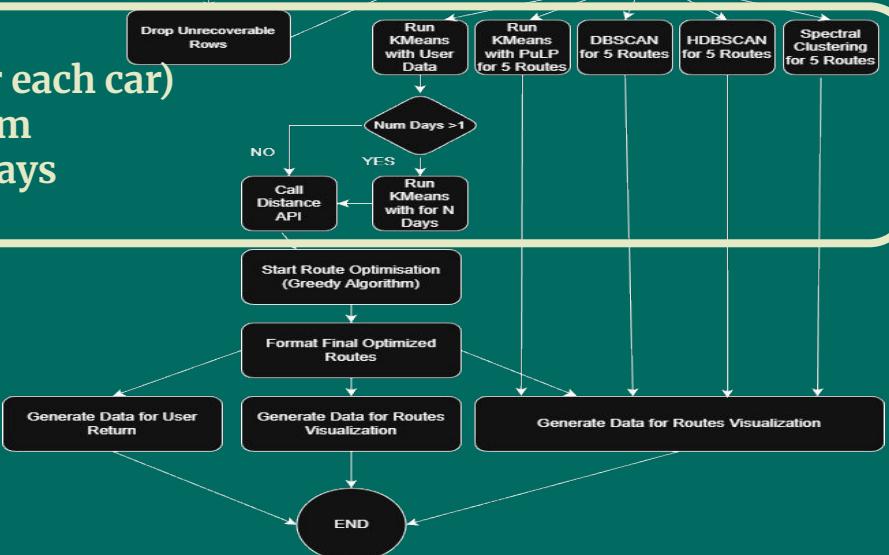


THE ALGORITHM & SOLUTIONS

- Get data from user, clean and check.
- Get missing data using API service
- Clean again



- Using the KMeans, obtain clusters (for each car)
- Based on the obtained clusters, perform another clustering by the number of days

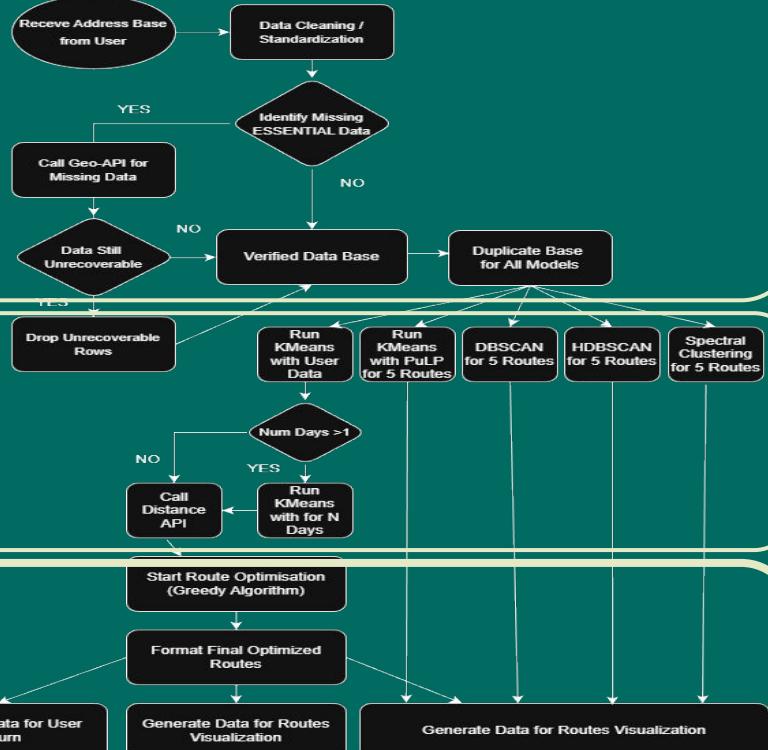


THE ALGORITHM & SOLUTIONS

- Get data from user, clean and check.
- Get missing data using API service
- Clean again

- Using the **KMeans**, obtain clusters (for each car)
- Based on the obtained clusters, perform another clustering by the number of days

- Create a distance matrix
- Calculate the nearest neighbor (distance of each point to each point)
- Format the results for return to the user and plotting on the map



THE PROJECT LIMITATION

geocode.maps.co:

API limitation:

Rate limit: No more than 1 request per second.

Note: If the limits are violated, the service will block the user.

OpenRoutingService:

API limitation:

Rate limit: No more than 2000 requests per second.

Local Server Limitations:

Route Limit: Maximum route distance (100 km).

Stop radius: near a point (this parameter can be changed).

Note: If these limits are violated for:

Local Server – the route will not be displayed on the map.

API endpoint – will be blocked

THE PROJECT TASK STATEMENT



You are a restaurant owner, you need to deliver dishes to your customers?



You are the owner of a small service company that repairs large household appliances?



You are the Head of sales of a distribution company who has to plan and calculate sales representatives' visits to customers on a weekly basis?

THE PROJECT SETTINGS

Deploy ⋮

Navigation

- Go To:
- Main
 - Settings
 - Tables
 - Routes Maps
 - Statistic
 - Other UML and models
 - About



Routing optimisation and VRP ↗

Input data for calculation

Upload a file with the customer address database and enter the data necessary for the calculation. Below, you can find a template for the database and a list of necessary fields and data types for each.

[Download template CSV](#)

Customer DataBase

Upload Customer Base File (.csv)



Drag and drop file here

Limit 200MB per file + CSV

Browse files

Please Upload the Customer Base file

Calculate

Other Parameters

Number of vehicles

1

Geo coordinates

Latitude

50,163908

- +

Longitude

8,684189

- +

THE PROJECT TABLES

Navigation

- Go To:
- Main
 - Settings
 - Tables
 - Routes Maps
 - Statistic
 - Other UML and models
 - About



Routing optimisation and VRP

Here you will find Tables

Choose necessary parameters:

Choose the route, please:

1

Choose the day, please:

1

The Route for vehicle 0 day 0

#	Next point of visit (ID)	Distance (m)	Duration (sec)	Customer Address	Latitude	Longitude
1		297	5696.1	Frankfurt am Main, Hartmannweg, 3	50.1444	8.7076
2		376	2601.6	Frankfurt am Main, Weidenbornstraße, 44	50.1318	8.7045
3		39	467.9	Frankfurt am Main, Rendeler Straße, 17	50.1298	8.7095
4		114	307	Frankfurt am Main, Alt-Bornheim, 27c	50.1297	8.7102
5		398	248.8	Frankfurt am Main, Berger Straße, 276	50.1288	8.7125
6		401	170.3	Frankfurt am Main, Freihofstraße, 7	50.13	8.713
7		457	1003.1	Frankfurt am Main, Ortenberger Straße, 55	50.1297	8.7177
8		418	904	Frankfurt am Main, Saalburgallee, 42	50.1234	8.7165
9		487	343.8	Frankfurt am Main, Simsonstraße, 28	50.1219	8.7151
10		324	540.4	Frankfurt am Main, Bornheimer Landwehr, 70	50.1222	8.7133

Download Route (.csv)

THE PROJECT MAPS

Navigation

- Go To:
 - Main
 - Settings
 - Tables
 - Routes Maps**
 - Statistic
 - Other UML and models
 - About

🚚 Routing optimisation and VRP ➔

Here you will find Routes Maps

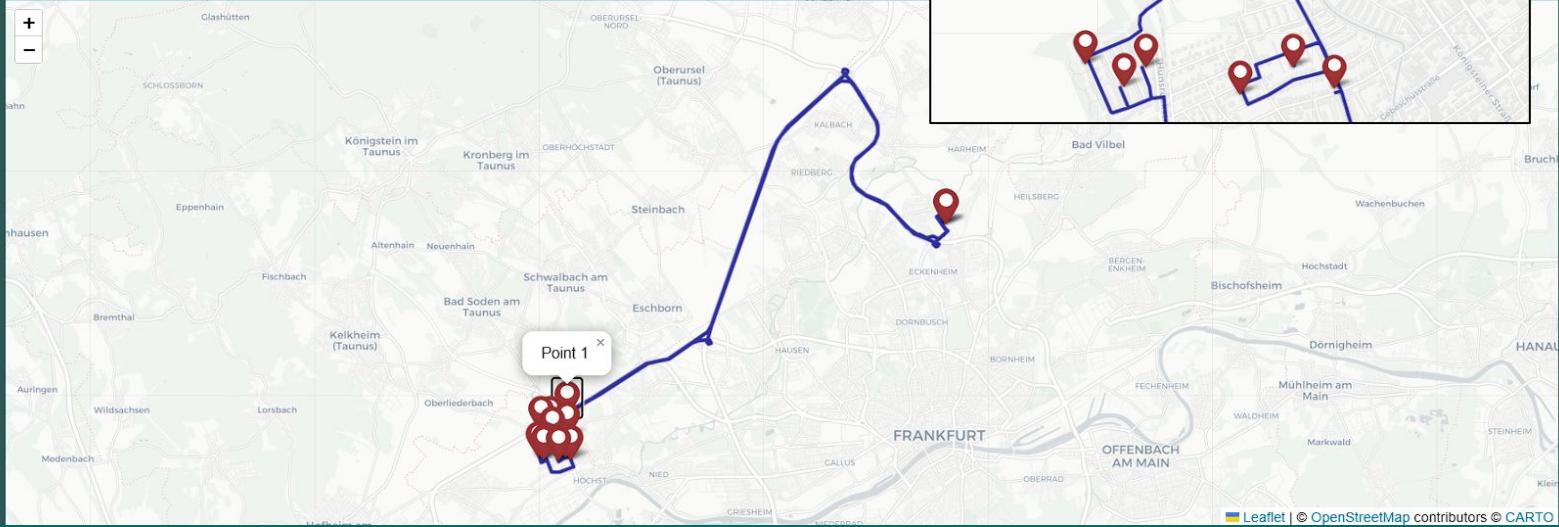
Choose necessary parameters:

Choose the route, please:

3

Choose the day, please:

4



THE PROJECT STATISTICS

« Deploy : :

Navigation

Go To:

- Main
- Settings
- Tables
- Routes Maps
- Statistic**
- Other UML and models
- About

 **Routing optimisation and VRP**

Here you will find statistics on calculated routes.

General overview KPI

General overview

Route #	Number of Customers	Distance (km)	Duration
1	101	174.08	6:39:40
2	57	219.39	5:37:12
3	55	287.55	5:04:12
4	68	99.77	4:17:49
5	45	130.69	4:03:55
6	79	104.55	4:53:55
7	58	292.05	5:56:03
8	65	200.31	6:02:06

THE PROJECT STATISTICS

« Deploy ⋮

Navigation

Go To:

- Main
- Settings
- Tables
- Routes Maps
- Statistic**
- Other UML and models
- About

 **Routing optimisation and VRP**

Here you will find statistics on calculated routes.

General overview KPI

KPI

Setting the Service Time Duration

Service duration per stop (min)

- + 3

Choose the route, please:

KPI Detailing by Route

The Route for vehicle 2

Day	Total_Customers_Num	Total Distance (km)	Total Travel Duration	Total Service Duration	Total Duration
0	11	61.28	1:08:14	2:03:14	3:11:29
1	16	57.95	1:14:23	2:34:23	3:48:46
2	5	61.01	0:55:34	1:20:34	2:16:09
3	11	52.66	1:00:10	1:55:10	2:55:21
4	7	54.65	0:45:49	1:20:49	2:06:38

THE PROJECT STATISTICS

Deploy 

Navigation

Go To:

- Main
- Settings
- Tables
- Routes Maps
- Statistic
- Other UML and models
- About

Additional information:

- About Project
- Workflow

Project Summary: Vehicle Routing Problem (VRP)

Project Summary and Goal

Credentials Used: Openrouteservice (based on OpenStreetMap data: © OpenStreetMap contributors) and geocode.maps.co

This project was developed as part of the [WBS Web Coding School](#) training program.

Project Goal: To solve the **Traveling Salesman Problem (TSP)** and the **Vehicle Routing Problem (VRP)** using a combination of clustering and a greedy algorithm (Nearest Neighbor search). Additionally, the goal was to create a user interface for data upload, retrieval, and visualization.



WBS Coding School

LinkedIn profile

Github profile

Project Workflow

- **Receive Data:** Obtain data and instructions from the user (Web interface).
- **Data Prep:** Data Cleaning (Pandas) and Data Verification (Pandas).
- **Geolocation:** If data is incomplete, retrieve necessary data (WEB API).
- **Clustering:** Data Clustering using Clustering Algorithms (K-Means, K-Means constrained, K-Means with PuLP, DBSCAN, HDBSCAN, Spectral Clustering).
- **Route Calculation:** Obtain distance data between points (WEB API).
- **Optimization:** Route generation (Greedy Algorithm, PyVRP).
- **Output:** Formatting the resulting information, preparation of visualizations, and analytical data for the user (Pandas, Folium).
- **Comparison:** Visualization of results from other algorithms for comparison.

THE PROJECT STATISTICS

Deploy :

- **Comparison:** Visualization of results from other algorithms for comparison.

Conclusions

Clustering Algorithms can be effectively utilized to solve the Traveling Salesman Problem, but with certain limitations:

1. **Closely Located Points (VRP):** For building multiple routes with a large number of closely located points, K-Means, K-Means constrained, and K-Means with PuLP are recommended.
2. **Distant Clusters:** For building multiple routes where points are grouped into distant clusters, DBSCAN, HDBSCAN, and Spectral Clustering can be used.

Constraint: Clustering (excluding the K-Means constrained algorithm) demonstrates a significant **disproportion** in cluster sizes.

Navigation

Go To:

- Main
- Settings
- Tables
- Routes Maps
- Statistic
- Other UML and models

- About

Additional information:

- About Project
- Workflow

Notes on API Service Limitations

Openrouteservice Server Limitations

- Maximum route distance (100 km).
- Stop radius near a point (this parameter can be changed).
- *Note:* If these limits are violated, the route will not be displayed on the map.

geocode.maps.co API Service Limitations

- Rate limit: No more than 1 request per second.
- *Note:* If the limits are violated, the service will block the user. For significant volumes of undefined geospatial coordinates, the retrieval process will take: `(number of undefined coordinates) * 1 second`.