**Feynn Labs**

Data science Internship

**Project Report Submission**

**EV Charging Station**

**Team Nikhil**

**Krishna Mahajan**

Problem statement

As electric vehicles (EVs) become more popular in India, one of the biggest challenges is ensuring that there are enough charging stations for drivers. Without easy access to charging, people may hesitate to switch to electric cars. This makes it crucial for companies to choose the best locations for launching their new EVs, where drivers can easily find charging stations.The problem is that charging stations are not evenly spread across the country. Some areas, especially big cities, have many stations, while others have very few or none. To address this, we need to figure out which parts of India have the best infrastructure to support a new electric vehicle launch. This project aims to use detailed data on charging stations—such as their locations, how many chargers they have, and when they're open—to identify the best places in India for launching an EV.

Dataset

<https://www.kaggle.com/datasets/nezukokamaado/e-v-charging-stations>

Source : official website of government of India .

<https://thecleverprogrammer.com/2024/04/18/data-science-projects-to-boost-your-resume/?fbclid=PAZXh0bgNhZW0CMTEAAaasT0srvLWFduDQGE-tb3eJJMMlLZ60-I4tGdrU3_LSpAGxRMfLwjorljg_aem_OPyPWhY2gOmMFelRITP1sg>

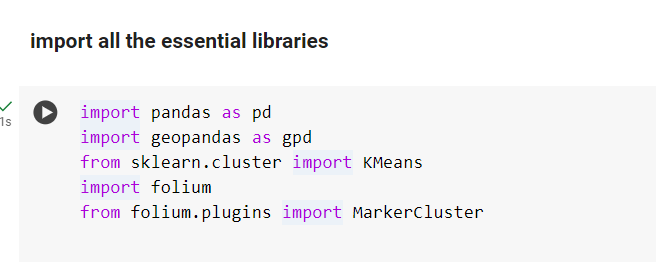
Github : <https://github.com/krishnx27/Evchargingstation>

Keyword Analysis

1. Charging stations.
2. Geospatial data.
3. Charging Networks.

Step 1.

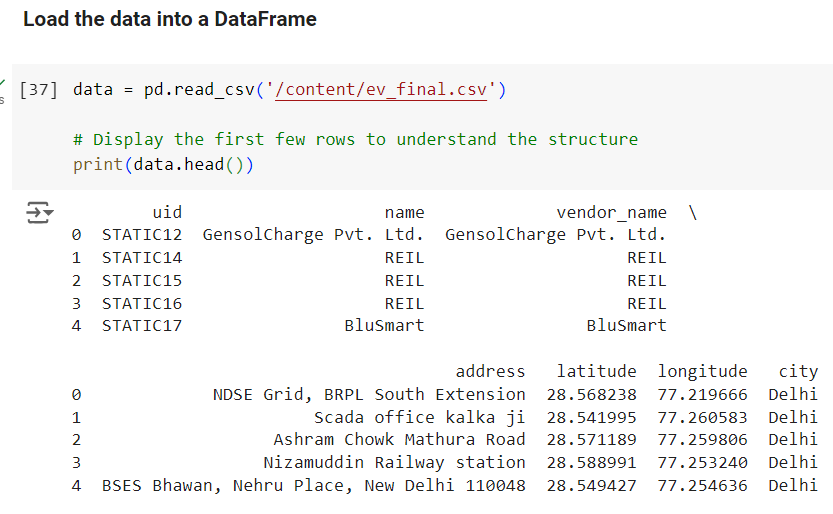
Import Libraries



The code you provided imports several libraries that will help you work with data: Pandas is used for handling and analyzing regular tabular data (like spreadsheets), while Geopandas adds the ability to work with geographic data, such as maps. KMeans, a part of the Scikit-learn library, is used to group similar data points together, which helps in identifying patterns in the data. Folium allows you to create interactive maps to visualize your data, and with the Marker Cluster feature, you can group many map markers into clusters to avoid clutter when displaying a lot of information. Together, these tools will enable you to analyze geographic information and show it effectively on maps.

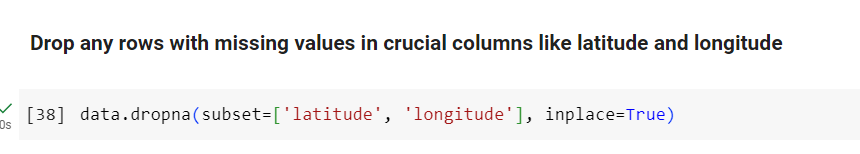
Step 2.

Load the data



Step 3.

Clean the data



Step 4.

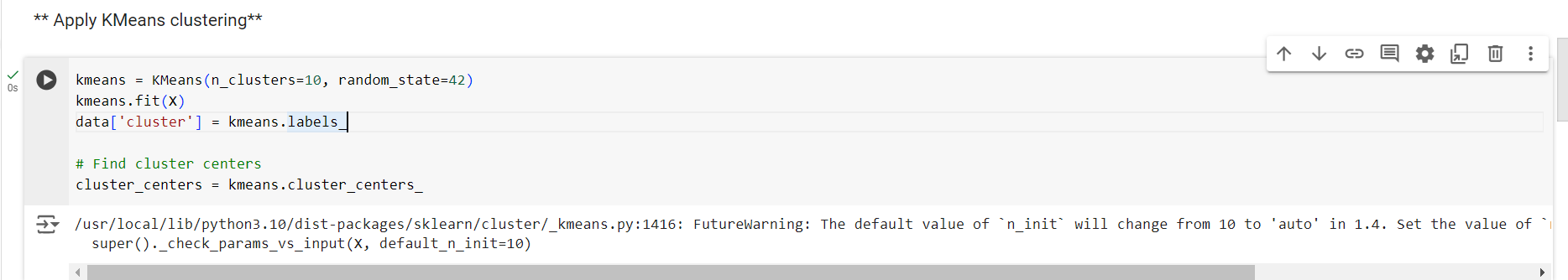
Convert to Geodata frame and Extract the columns



Step 5.

Apply KMeans cluster

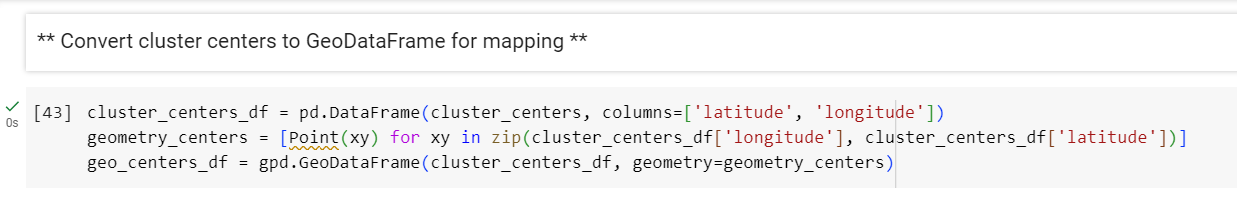
we start by creating a KMeans clustering model that groups data into 10 clusters, using a random state for consistency in results. The `kmeans.fit(X)` line fits this model to a dataset represented by `X`, which means it learns how to categorize the data based on its features. After the model is trained, we add a new column called 'cluster' to the original data, which contains the labels indicating which cluster each data point belongs to. Finally, the model calculates the center points of these clusters, which are stored in the variable `cluster\_centers`. These centers represent the average position of the data points in each cluster, helping us understand the characteristics of each group.



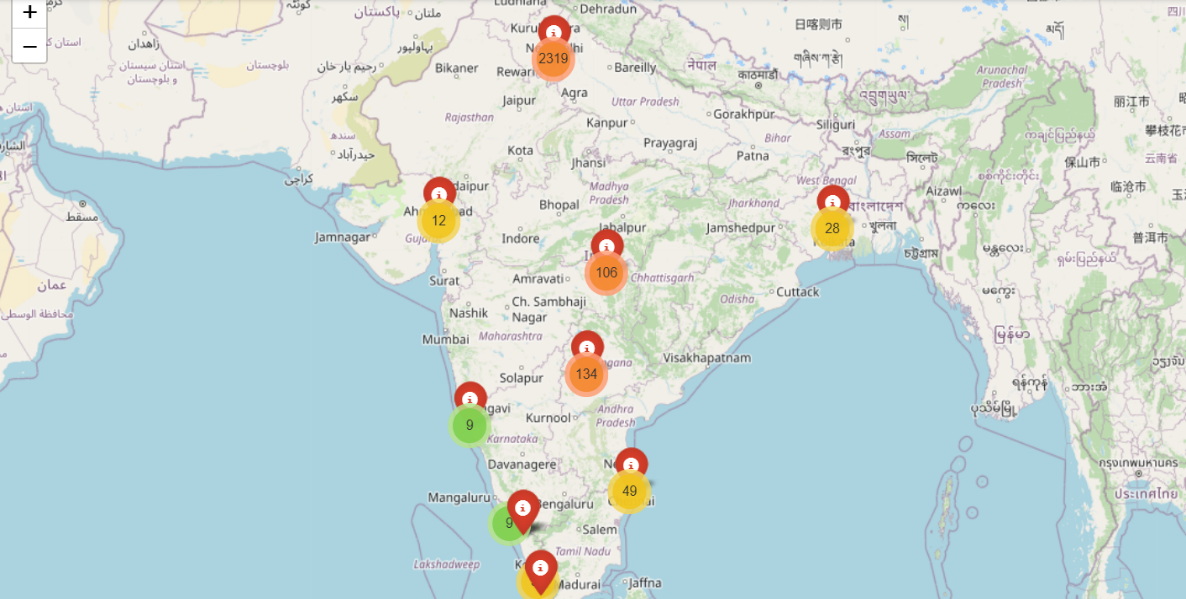
Step 6.

Converting GeoData framing for mapping and initialize the map centered on india.

we start by creating a Data Frame called `cluster\_centers\_df` that organizes the cluster center information, specifically the latitude and longitude of each center. We then create a list of geometric points, named `geometry\_centers`, by combining the longitude and latitude of each center to form coordinates. This is done using a simple loop that pairs each longitude with its corresponding latitude. Finally, we create a new GeoDataFrame called `geo\_centers\_df` using the original cluster centers DataFrame and attach the geometric points to it. This GeoDataFrame allows us to handle geographic data easily, making it ready for mapping and spatial analysis.





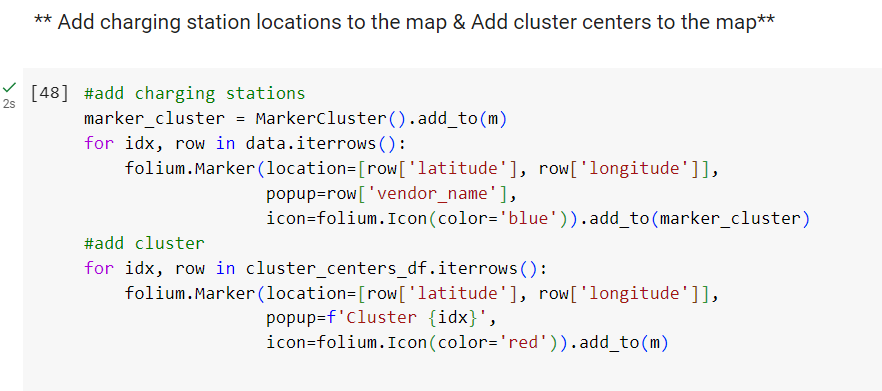


The location parameter specifies the starting point of the map using latitude and longitude coordinates, in this case, it centers the map on India (latitude 20.5937 and longitude 78.9629). The zoom\_start parameter sets the initial zoom level of the map to 5.

Step 7.

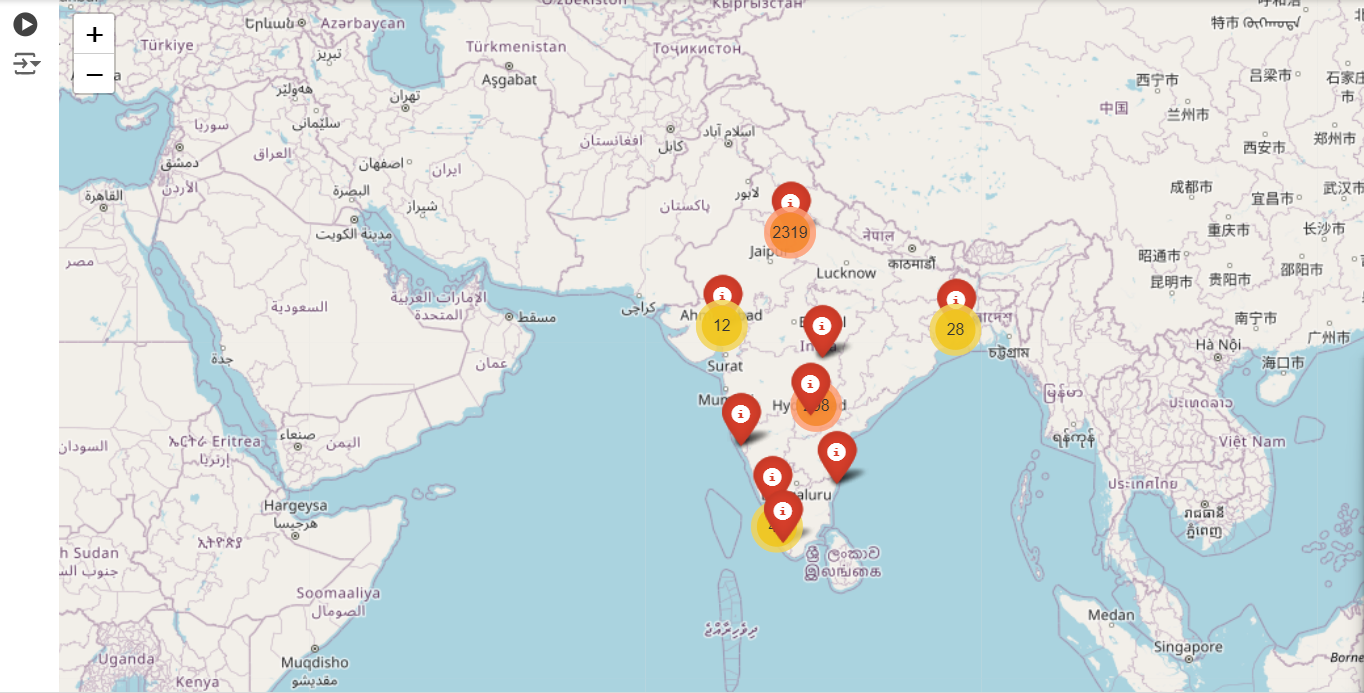
we first create a cluster for markers on the map using MarkerCluster(), which helps group nearby markers together for a cleaner view. Then, we loop through each row in the data DataFrame, which contains information about charging stations. For each charging station, we add a blue marker on the map at its respective latitude and longitude. When you click on a marker, it will show the name of the vendor associated with that station.

Next, we loop through the cluster\_centers\_df DataFrame, which contains the locations of the cluster centers we identified earlier. For each cluster center, we add a red marker on the map that represents that specific cluster. When you click on these red markers, they display which cluster they represent. Overall, this code adds visual markers for both individual charging stations and their respective clusters on the map, making it easy to visualize and identify them.



Step 8.





**Conclusion of the EV Charging Station Dataset and Visualization**

This project successfully transformed a dataset of EV charging stations into an engaging and practical interactive map. It serves as a vital resource for EV users, promoting convenience and encouraging the adoption of electric vehicles in India. The intuitive design and functionality demonstrate the potential for data visualization in enhancing user experience and supporting sustainable practices in transportation.

In conclusion, this project transforms raw data into a powerful visual tool that aids EV users in locating charging stations effectively. By following a systematic approach—starting from data collection, through analysis, and finally to visualization—we have created an accessible and informative map. This solution not only enhances user experience but also contributes to the growth of the electric vehicle infrastructure by increasing awareness and facilitating the use of electric vehicles in India.

**The cities with the highest number of charging stations in the Given dataset are :**

1.New Delhi – with 1734 stations

2.Delhi – with 704 stations

3.Nagpur- with 98 stations

4.Chennai-with 48 stations

5.Kolkata-with 25 stations

These cities have the most significant number of charging stations according to the data.

**The charging station name with the highest number of stations in the dataset**

The charging station name with the highest number of stations in the dataset is “REVOS” , which has 1,868 stations. Other names with notable counts include “EESL” with 117 stations and “E-Fill Electric” with 18 stations.