Traveler Perspectives: Analysis of Hotel Reviews and Ratings

Lokesh Pallikonda, Siva Ram Sai Paruchuri, Prashanth Reddy Dandyala, Aravind Kumar Koyyala.

Northwest Missouri State University, Maryville MO 64468, USA

1 Project Idea

Our goal in this big data project is to extract valuable insights into the hospitality industry by analyzing a dataset of European hotel reviews. We will be using PySpark queries to track the average review scores over time, identify the top-performing hotels, explore the distribution of reviewer scores, uncover common tags given by reviewers, assess nationality-wise reviewer scores, generate word clouds for positive and negative reviews, investigate the correlation between scores and word counts, and visualize the geospatial distribution of hotels. We will use Tableau for visualization to present the findings clearly and concisely. The visualizations will include dynamic charts and maps that depict trends, rankings, and correlations within the dataset, enabling stakeholders to make data-driven decisions for strategic improvements in hotel services and guest experiences

2 Technology Summary

We will use Pandas for data cleaning i.e., which includes handling data that are repeated, missing etc and PySpark queries to track desired goals. Furthermore, Tableau is used for visualizing the Result data from PySpark queries .

3 Architecture Diagram

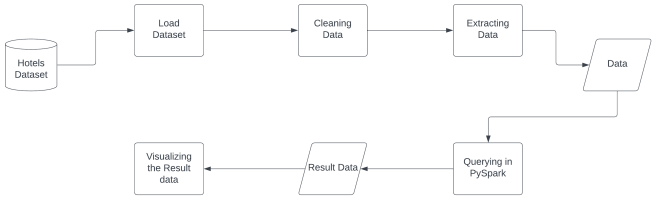

Fig. 1. Data Flow Diagram

4 Architecture Summary

1. The first step is to load the raw data from the Hotels Dataset
2. Next, we perform data cleaning to handle any inconsistencies, errors, or missing data in the dataset.
3. After that, we extract specific information from the dataset using data extraction techniques.
4. Once the data is cleaned and extracted, we process it further by querying in PySpark to execute commands and operations on the dataset using the PySpark framework.
5. Finally, we visualize the data to gain an understanding of the overall trend and pattern.

5 Project Goals

1. Calculate the average reviewer score for hotels over different review dates.
2. Identify the top N hotels with the highest average scores.
3. Analyze the distribution of reviewer scores across all hotels.
4. Identify the most common tags given by reviewers.
5. Calculate average reviewer scores grouped by the nationality of the reviewers.
6. Temporal Analysis of Review Counts.
7. The correlation between reviewer scores and the word counts in reviews.
8. Geospatial distribution of hotels based on latitude and longitude.

6 Project Description

This dataset contains 515,000 customer reviews and scoring of 1493 luxury hotels across Europe. Meanwhile, the geographical location of hotels are also provided for further analysis.
The csv file contains 17 fields. The description of each field is as below:

⇒ Hotel_Address: Address of hotel.
⇒ Review_Date: Date when reviewer posted the corresponding review.
⇒ Average_Score: Average Score of the hotel, calculated based on the latest comment in the last year.
⇒ Hotel_Name: Name of Hotel
⇒ Reviewer_Nationality: Nationality of Reviewer
⇒ Negative_Review: Negative Review the reviewer gave to the hotel. If the reviewer does not give the negative review, then it should be: 'No Negative'
⇒ Review_Total_Negative_Word_Counts: Total number of words in the negative review.
⇒ Positive_Review: Positive Review the reviewer gave to the hotel. If the reviewer does not give the negative review, then it should be: 'No Positive'
⇒ Review__Positive_Word_Counts: Total number of words in the positive review.
⇒ Reviewer_Score: Score the reviewer has given to the hotel, based on his/her experience
⇒ Total_Number_of_Reviews_Reviewer_Has_Given: Number of Reviews the reviewers has given in the past.
⇒ Total_Number_of_Reviews: Total number of valid reviews the hotel has.
⇒ Tags: Tags reviewer gave the hotel.
⇒ days_since_review: Duration between the review date and scrape date.
⇒ Additional_Number_of_Scoring: There are also some guests who just made a scoring on the service rather than a review. This number indicates how many valid scores without review in there.
⇒ lat: Latitude of the hotel
⇒ lng: longtitude of the hotel

6.1 Implementation steps

1. Load the datset into the JupyterLab's notebook using "df = pd.read_csv('Hotel_Reviews.csv')"
2. Now, drop the null values using "df.dropna()"
3. After the removing the null values, we have to check every cell where user may have entered just spaces. This process can done column wise for example, for x in df.index:
   if df.loc[x,"Negative_Review"]==" ":
   df.drop(x, inplace=True)
   the above code drops all the records where Negative_Review is " ". We have to repeat this process for all the columns i.e., 17 times.
4. Finally, save the pandas data frame's data into file called Result.csv by code "df.to_csv("Result.csv",index=False)
5. Now perform the necessary steps to visualize data for our desired goals

7 Results Summary

A discussion of the results achieved for each goal, which may include providing clear screenshots or snippets of our source code.
First, load the result dataset which was cleaned using pandas in the PySpark DataFrame

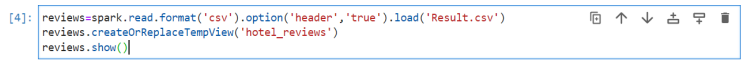

Fig. 2. Creation of Data Frame in PySpark

7.1 Goal 1 :Average Review Scores Over Time

Fig. 3.

The above code snippet is to Calculate the average reviewer score for hotels over different review date and the Line chart showing the trend of average scores over time in tableau. Before Visualizing the data the data present average_scores_over_time dataframe should be stored into a csv file. Ive stored it into a folder called goal_01. Here the Year of the Review date is taken as X-axis and

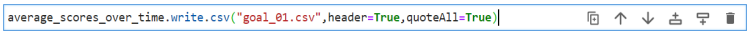

Fig. 4. Saving the data in data frame

on Y-axis average review score is taken. Metrics:

Data quality: accuracy of the Reviewer_Score data was good because we have cleaned using pandas. Latency: time taken to compute the average scores over time was around few milliseconds. Resource utilization: CPU and memory usage during the computation was around 450 mb.

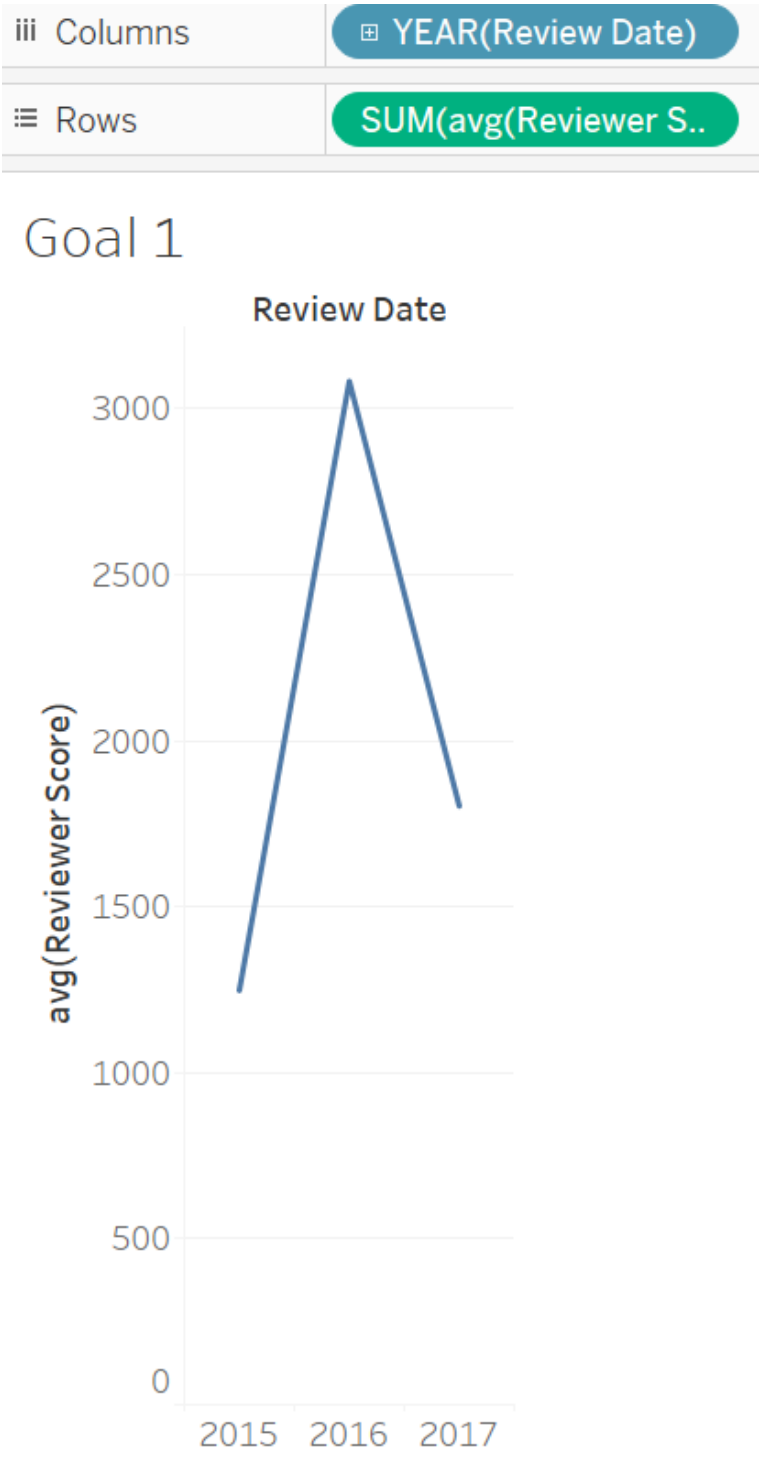

Fig. 5. Visualization of Goal 1

7.2 Goal 2: Top N Hotels with the Highest Average Scores

```
[55]: top_hotels = reviews.groupBy("Hotel_Name").avg("Reviewer_Score").orderBy("avg(Reviewer_Score)", ascending=False).
top_hotels.show(25)
```

Fig. 6. Goal 2 code snippet

The above code snippet is to identify the top N hotels with the highest average scores. The data present in top_hotels dataframe is stored into a folder goal_02.

```
[57]: top_hotels.write.csv("goal_02.csv", header=True)
```

Fig. 7. Saving the data in data frame

The csv file in the goal_02 is loaded into the tableau and it is visualized as bar graph where X-axis consists of different hotels name and Y- axis is average of reviewer score.(The bar chart shows all the hotels) Metrics Data quality:

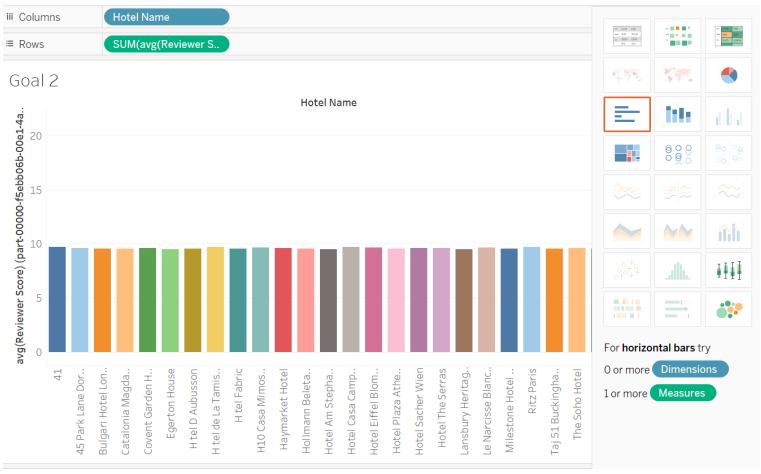

Fig. 8. Visualization of Goal 2

accuracy of the average scores and hotel rankings is good Processing time: Few milliseconds. Memory Utilization: 450 MB

7.3 Goal 3: Distribution of Reviewer Scores

```
[58]: reviewer_scores_distribution = reviews.select("Reviewer_Score")
      reviewer_scores_distribution.show()
```

Fig. 9. Goal 3 code snippet

The above code snippet is to analyze the distribution of reviewer scores across all hotels. The The data present in top_hotels dataframe is stored into a folder goal_03. Later, the data present in that folder is visualized as histogram.

```
[59]: reviewer_scores_distribution.write.csv("goal_03.csv",header=True)
```

Fig. 10. Saving the code present in data frame

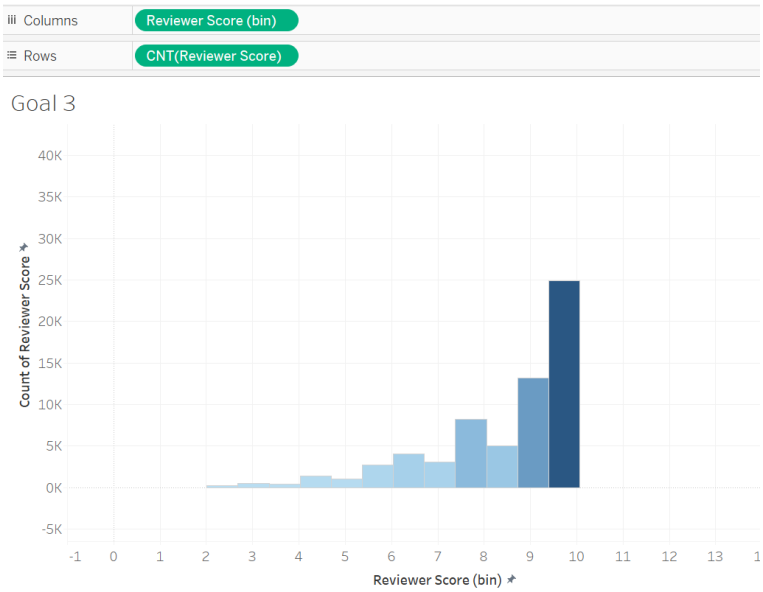

Fig. 11. Visualization of Goal 3

7.4 Goal 4: Most Common Tags Given by Reviewers

```
[62]: from pyspark.sql.functions import explode, split, col

tags_df = reviews.select("Tags")
tags_df = tags_df.withColumn("Tag", explode(split(col("Tags"), ",")))
most_common_tags = tags_df.groupBy("Tag").count().orderBy("count", ascending=False)
most_common_tags.show()
```

Fig. 12. Goal 4 code snippet

The code Snippet is to Identify the most common tags given by reviewers. The data is stored a folder called goal_04. The data present in the above folder

```
most_common_tags.write.csv("goal_04",header=True)
```

Fig. 13. Saving the code present in data frame

is visualized as bar graph where x-axis represents tags and y-axis shows the count of the tags. The data which was used above to visualize the bar graph was accuarte and memory utilization was around 500 MB.

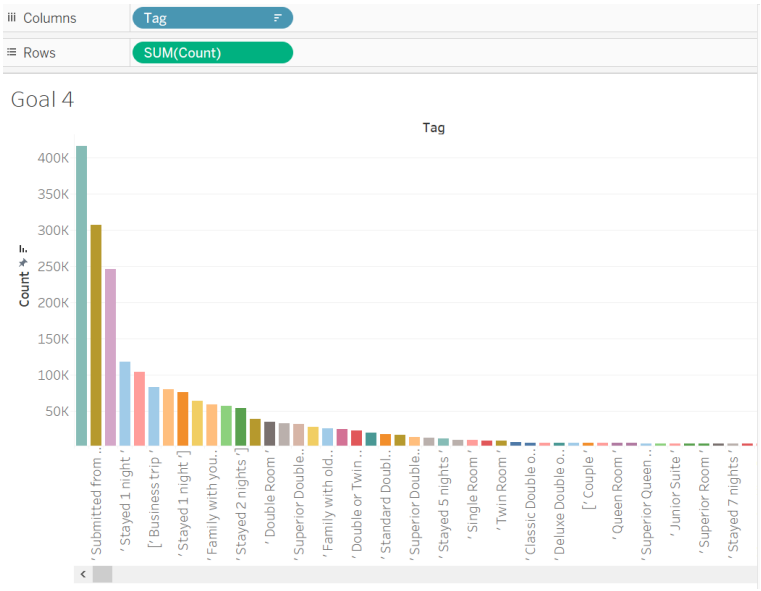

Fig. 14. Visualization of Goal 4

7.5 Goal 5: Nationality-wise Reviewer Scores

```
[64]: nationality_wise_scores = reviews.groupBy("Reviewer_Nationality").avg("Reviewer_Score")
      nationality_wise_scores.show()
```

Fig. 15. code snippet for goal 5

The goal is to Calculate average reviewer scores grouped by the nationality of the reviewers The data is stored in the folder named goal_05. The above

```
[67]: nationality_wise_scores.write.csv("goal_05",header=True)
```

Fig. 16. Saving the code present in data frame

visualized as tree map where the dark color indicates the average is greater than the other along with the reviwer's nationality.

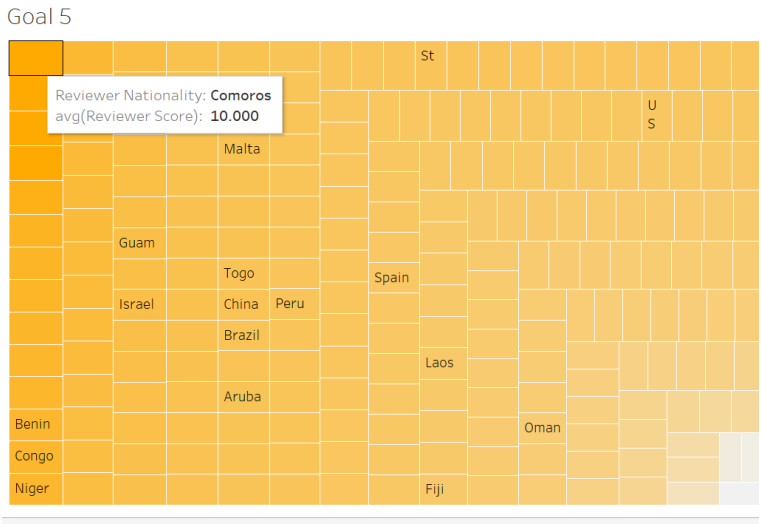

Fig. 17. Saving the code present in data frame

7.6 Goal 6: Circle of Positive and Negative Reviews

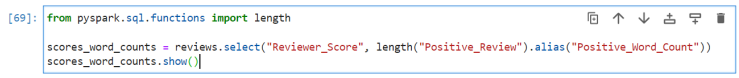

Fig. 18. code snippet

The above code snippet calculates total length of the positive review and stores it in the dataframe called scores_word_counts along with reviewer_score. The data is stored in the folder called goal_06 and here coalesce(1) represents

Fig. 19. saving the code

that the entire output data in dataframe is stored in one file. Now we have to repeat the same process for the negative review and calculate its length. And store it in the folder named goal_6.1 Later, we have to visualize both data present in tha folders goal 6 and 6.1 with the help of tableau. The above circle chart represents a relationship between positive review word, Negative review word

```
[8]: from pyspark.sql.functions import length

scores_word_counts_n = reviews.select("Reviewer_Score", length("Negative_Review").alias("Negative_Word_Count"))
scores_word_counts_n.show()
```

Fig. 20. code snippet

```
[9]: scores_word_counts_n.coalesce(1).write.csv("goal_6.1",header=True)
```

Fig. 21. code snippet

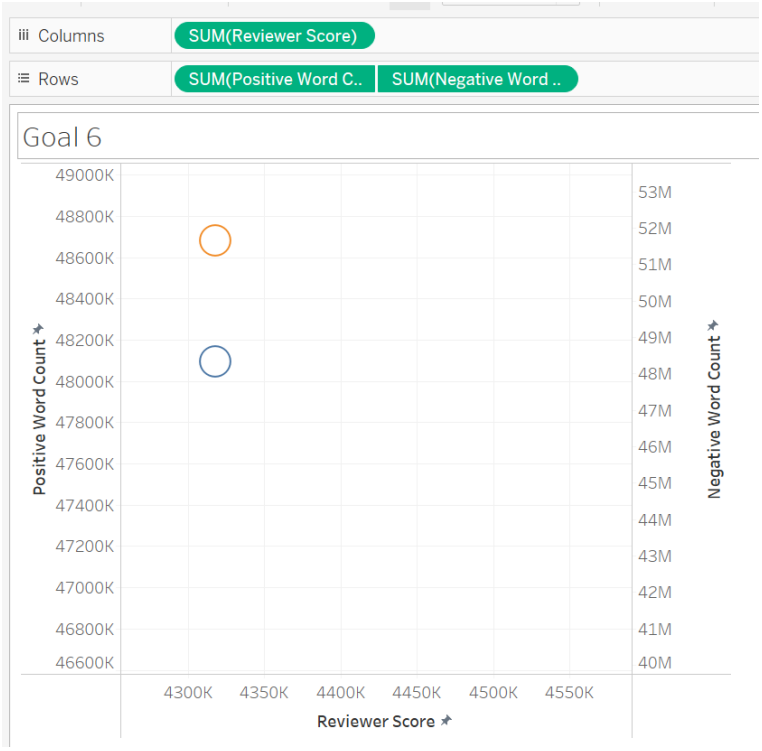

Fig. 22. Visualization of goal 6

and Reviewer' score. The blue circle represents the sum of the positive words and orange circle represents sum of Negative words.

7.7 Goal 7: Analyzing the various Reviewer nationalities.

```
[81]: reviewer_engagement = reviews.groupBy("Reviewer_Nationality").count()
      reviewer_engagement.show()
```

Fig. 23. code snippet

The above code snippet calculates the count of various nationalities of reviewers and stores it in the reviewer_engagement along with the country. Store the data into folder named goal_07. The above map shows the number of review-

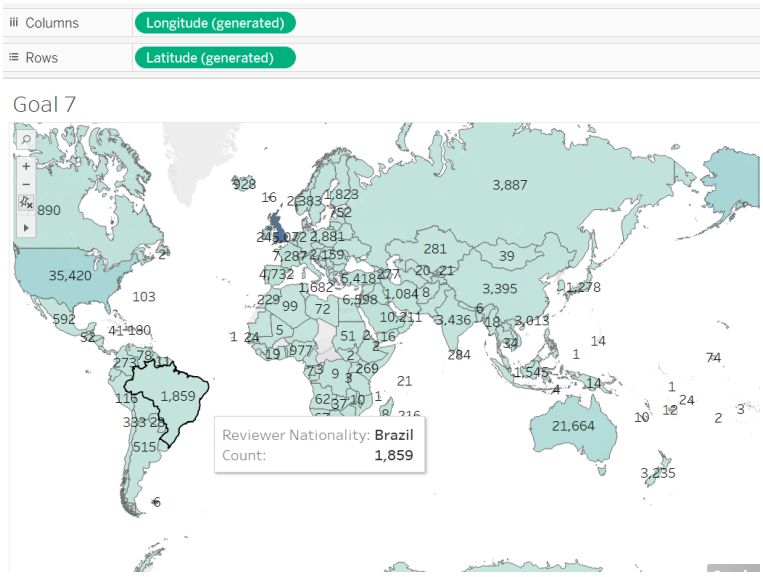

Fig. 24. Visualization of Goal 7

ers from each country, more intense the color means more number of reviewers from each country.

7.8 Goal 8: Geospatial Distribution of Hotels

```
[75]: selected_columns = ["Hotel_Name", "lat", "lng"]
      unique_hotels_geo_df = reviews.select(selected_columns).dropDuplicates(["lat", "lng"])
      unique_hotels_geo_df.show()
```

Fig. 25. Code Snippet

The above code snippet stores Hotel names, unique latitude, unique longitude and store it in a folder goal_08. Then map the data in the tableau.

```
[76]: unique_hotels_geo_df.coalesce(1).write.csv("goal_08", header=True)
```

Fig. 26. Saving the code

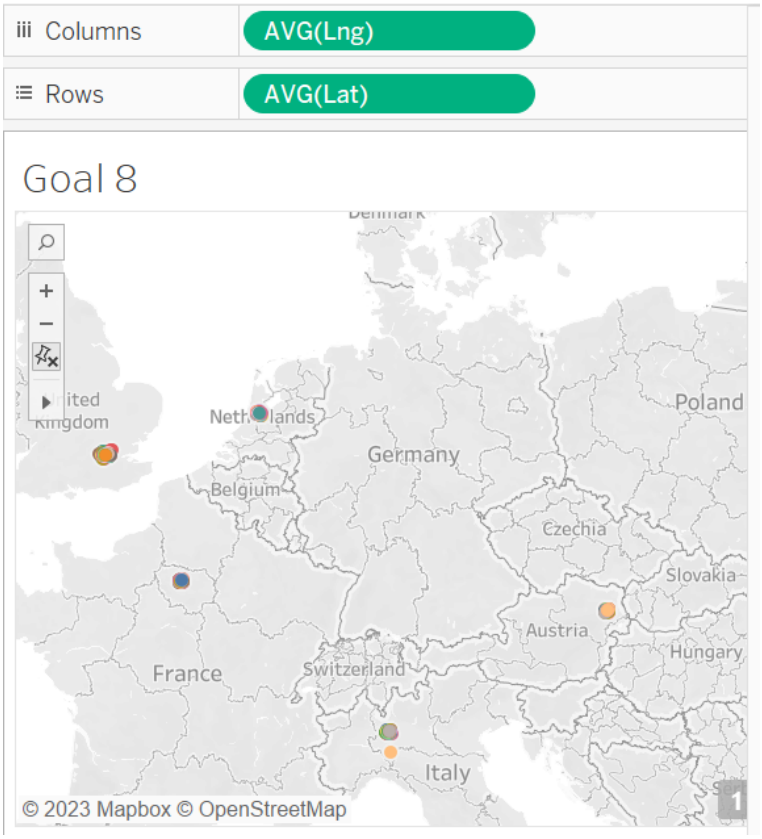

Fig. 27. Visualization of Goal 8

8 Conclusion

The completion of the Big Data project, focusing on the analysis of hotel reviews using PySpark for data processing and Tableau for visualization, has yielded valuable insights into customer sentiments, reviewer behaviors, and the performance of luxury hotels across Europe. The project successfully addressed eight distinct goals, each contributing to a comprehensive understanding of the dataset.issues.

References

https://www.kaggle.com/datasets/jiashenliu/515k-hotel-reviews-data-in-europe

Git Hub link: https://github.com/Lokesh156/Big_Data_Project