Project Report
Tripadvisor Data Analysis

Kunal Jayesh Modi (kjm597)
Romil Hemant Chauhan(rhc294)
Keval Jayesh Lakhani(kjl438)
Chinmay Ramesh Patil(crp382)

Instructor: Professor Rodriguez

Abstract

Given the rampant growth of travel-related user generated content on the Internet, enormous amount of data is generated every minute. By analyzing this large amount of information – both structured and unstructured – quickly, users can valuable insight immediately. On a broad scale, data analytics technologies and techniques provide a means of analyzing data sets and drawing conclusions about them to help organizations make decisions.

Keeping customers happy is key to the travel and hotel industry, but customer satisfaction can be hard to gauge – especially in a timely manner. TripAdvisor, a popular travel review site. Since its launch in 2000, it has garnered more than some 75 million reviews. With such dime a dozen opinions, it is no wonder that TripAdvisor is recognized as an important information source among users for travel planning.

Contents

1. Introduction

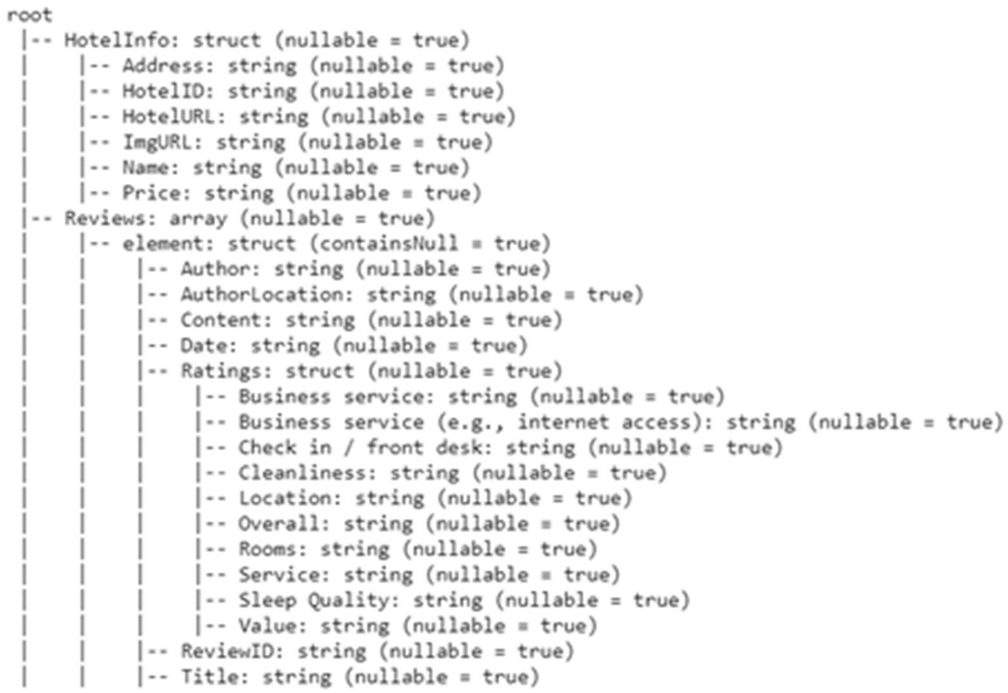
It is the season of holidays and it is important that the hotels should be able to provide better service to the travellers. In order for hotels to improve and grow, it is necessary to know where they need to improve. More importantly, they should be able to improve based on the traveller’s feedback.

Inspired from this we have developed a application by performing analysis on hotels all over the USA and traveller’s reviews given for the hotels. The different kind of analysis will help the hotels to improve based on specific parameters like cleanliness, food, service, rooms and many more. Hotels could also compare their growth over a period of time compared to previous years and also, compare with other hotels based on different parameters distributed geographically.

2. Methodology

2.1 Data Capture

TripAdvisor data was obtained from this particular website http://times.cs.uiuc.edu/~wang296/Data/. The data is in the JSON format and it gives the overview of different hotels in USA and the reviews given by user for those hotels over a period of few years. The data initially obtained was not cleaned and missed few of the required attributes. The data cleaning is further explained in its respective section. The overall structure of the data looked as follows:



```
root
 |-- HotelInfo: struct (nullable = true)
 |    |-- Address: string (nullable = true)
 |    |-- HotelID: string (nullable = true)
 |    |-- HotelURL: string (nullable = true)
 |    |-- ImgURL: string (nullable = true)
 |    |-- Name: string (nullable = true)
 |    |-- Price: string (nullable = true)
 |-- Reviews: array (nullable = true)
 |    |-- element: struct (containsNull = true)
 |    |    |-- Author: string (nullable = true)
 |    |    |-- AuthorLocation: string (nullable = true)
 |    |    |-- Content: string (nullable = true)
 |    |    |-- Date: string (nullable = true)
 |    |    |-- Ratings: struct (nullable = true)
 |    |    |    |-- Business service: string (nullable = true)
 |    |    |    |-- Business service (e.g., internet access): string (nullable = true)
 |    |    |    |-- Check in / front desk: string (nullable = true)
 |    |    |    |-- Cleanliness: string (nullable = true)
 |    |    |    |-- Location: string (nullable = true)
 |    |    |    |-- Overall: string (nullable = true)
 |    |    |    |-- Rooms: string (nullable = true)
 |    |    |    |-- Service: string (nullable = true)
 |    |    |    |-- Sleep Quality: string (nullable = true)
 |    |    |    |-- Value: string (nullable = true)
 |    |    |-- ReviewID: string (nullable = true)
 |    |    |-- Title: string (nullable = true)
```

2.2 Tech Stack

- PySpark
- Python MatplotLib
- Plotly/Seaborn
- Python Pandas
- R/RShiny
- TextBlob (Sentiment Analysis)

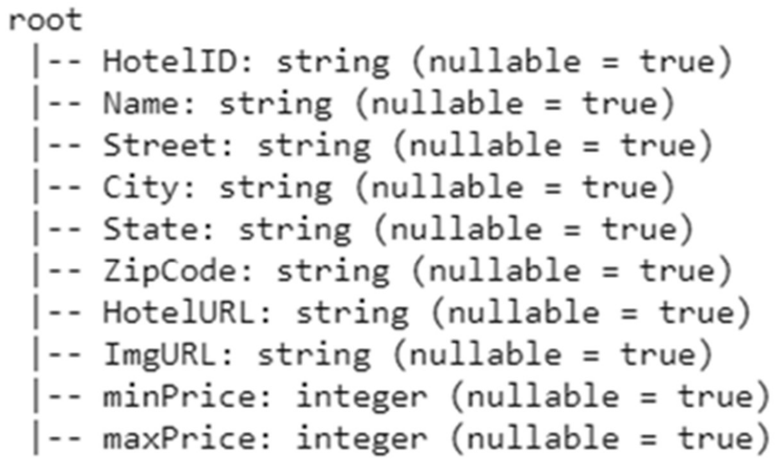
2.3 Data Cleaning

The data initially obtained was not cleaned and missed few of the required attributes. The cleaning of the data was done using PySpark as follows:

- The hotels and reviews data were distinguished.
- The prices for each hotels were given a definite range and represented using the min and max price.
- The address for each hotels were in the html format. They were properly parsed and segregated into streets, city, state and zipcode.
- Latitude and longitude was retrieved using the zipcode and added to the data in order to plot on the map.
- In the reviews data, the sentiment analysis using the textblob library was performed to get the sentiment scores for each reviews. These sentiment scores are further used for data analysis.
- Removing the NA values from the data
- Reformatting the date in the reviews.

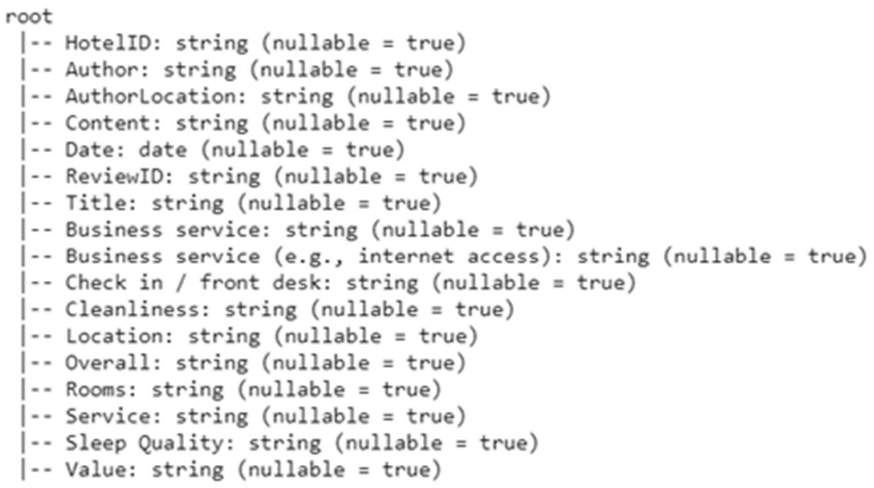
The cleaned data was quite helpful in doing the further data analysis and get useful information.

Hotel Data:



```
root
 |-- HotelID: string (nullable = true)
 |-- Name: string (nullable = true)
 |-- Street: string (nullable = true)
 |-- City: string (nullable = true)
 |-- State: string (nullable = true)
 |-- ZipCode: string (nullable = true)
 |-- HotelURL: string (nullable = true)
 |-- ImgURL: string (nullable = true)
 |-- minPrice: integer (nullable = true)
 |-- maxPrice: integer (nullable = true)
```

Reviews Data:

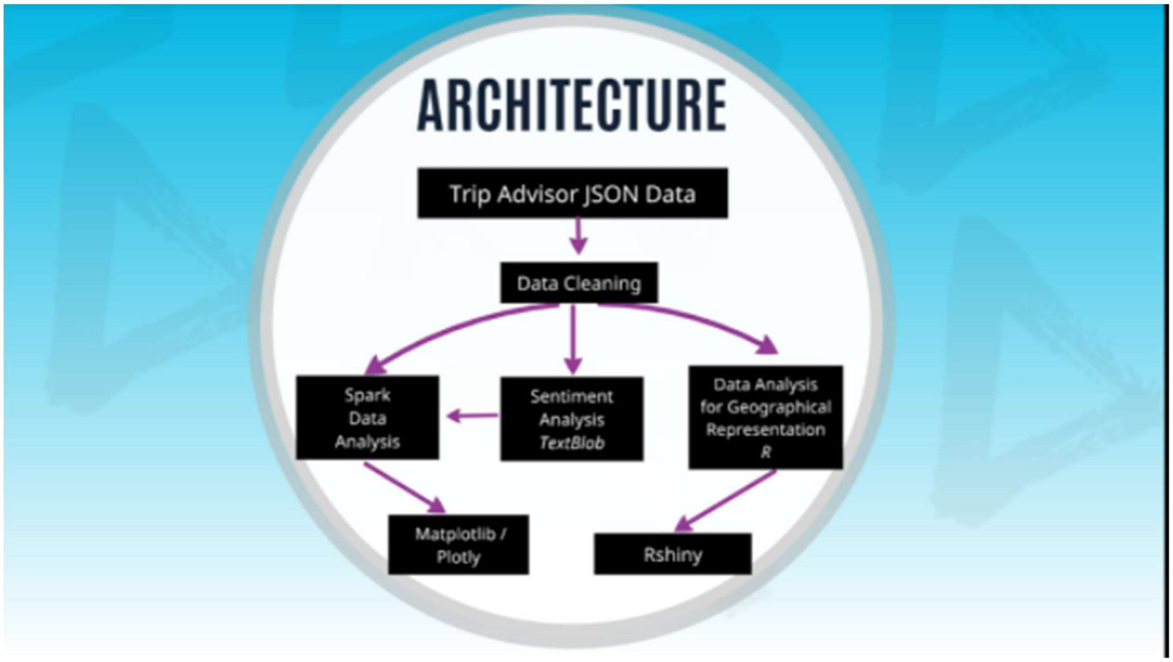


```
root
 |-- HotelID: string (nullable = true)
 |-- Author: string (nullable = true)
 |-- AuthorLocation: string (nullable = true)
 |-- Content: string (nullable = true)
 |-- Date: date (nullable = true)
 |-- ReviewID: string (nullable = true)
 |-- Title: string (nullable = true)
 |-- Business service: string (nullable = true)
 |-- Business service (e.g., internet access): string (nullable = true)
 |-- Check in / front desk: string (nullable = true)
 |-- Cleanliness: string (nullable = true)
 |-- Location: string (nullable = true)
 |-- Overall: string (nullable = true)
 |-- Rooms: string (nullable = true)
 |-- Service: string (nullable = true)
 |-- Sleep Quality: string (nullable = true)
 |-- Value: string (nullable = true)
```

2.4 Architecture

The architecture of the application mainly includes of four layers:

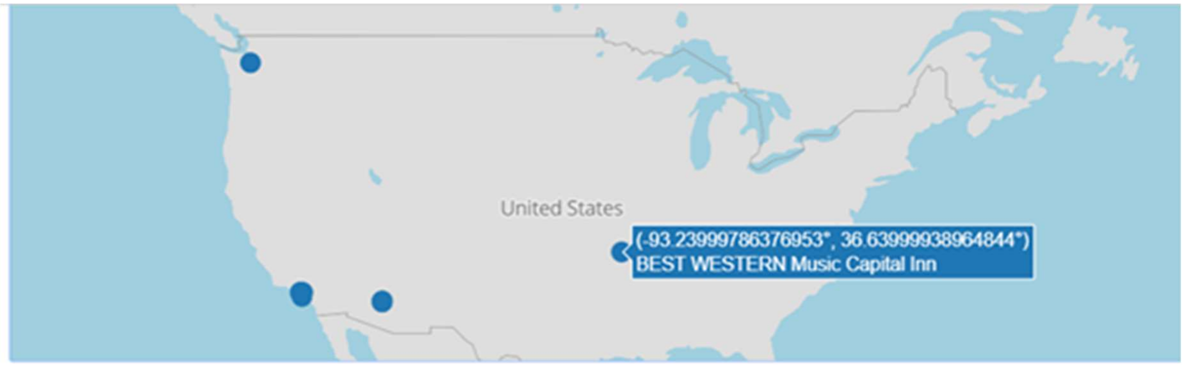
- Capturing the JSON Trip Advisor Data
- Cleaning the data to remove useful and add required data.
- Performing sentiment analysis and doing data analysis using PySpark and R/RShiny.
- Display the analyzed data in form of UI using MatplotLib, Plotly, Seaborn and RShiny.



3. Results and Discussion

3.1 Top 10 Hotels

We first started off by plotting the top 10 Hotels based on the Average Overall Rating of the Hotel. We noticed that most of the top hotels are cluttered in some regions and thus we could see 3 to 4 regions containing the top 10 Hotels.

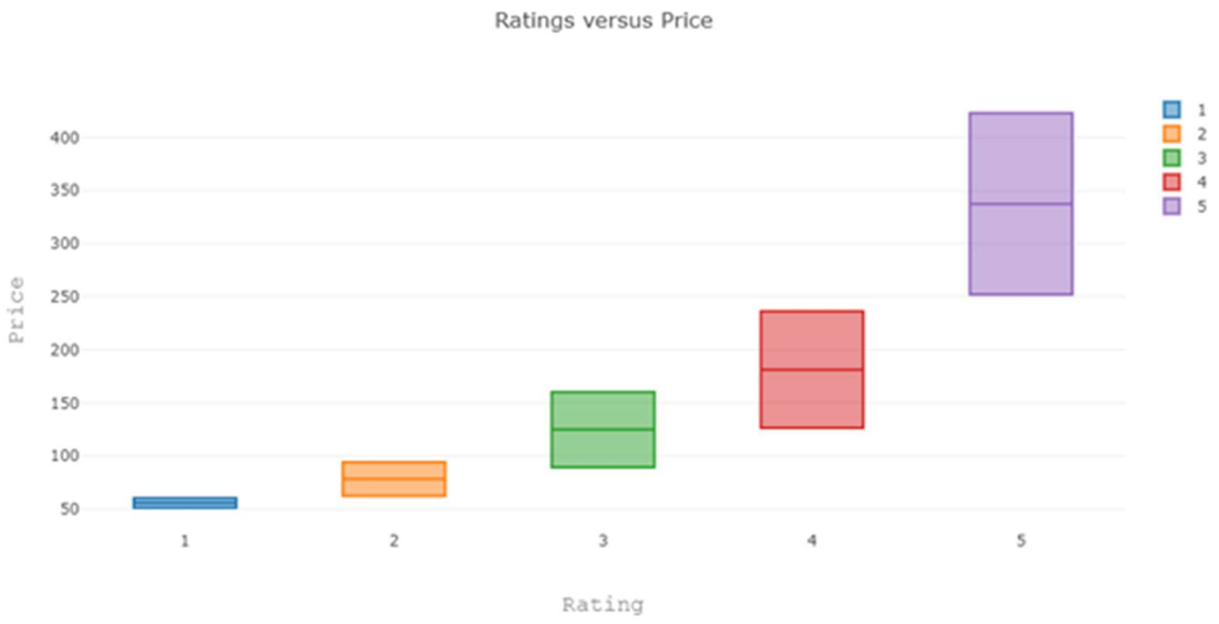


Here we can see that the top 10 hotels are cluttered in these 4 regions showing the blue markers. We used plottly to show this map.

3.2 Correlation (Hotel Rating vs Price)

Now, the next thing we did was correlate the hotel ratings with a price range. This would be particularly useful for the customers to figure out if they are paying the right amount for the Hotel rooms based on the hotel ratings.

Here is the Box plot for the same:

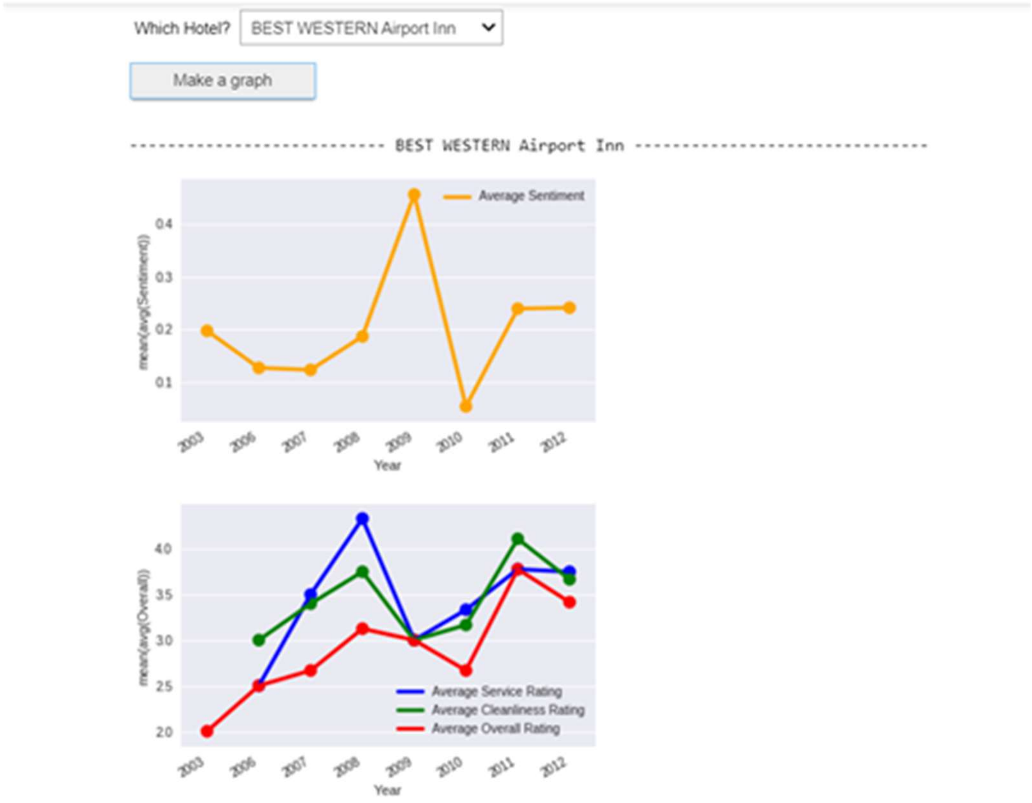


Thus, Looking at the above figure, customers can clearly determine the price range for a particular Hotel.

3.3 Average Sentiment vs Average Ratings(Cleanliness, Rooms, Service)

This feature was implemented with a goal to help the Hotels determine what feature of hotel has improved or deteriorated over the years which has caused the rise and fall of the sentiment of the ratings by users.

We provided a dropdown which contained all the hotels in the data. 2 graphs, one containing the average sentiment and other containing the average ratings for the years would be plotted. The graph would give a clear picture to hotel owner what feature of the hotel needs to be improved.



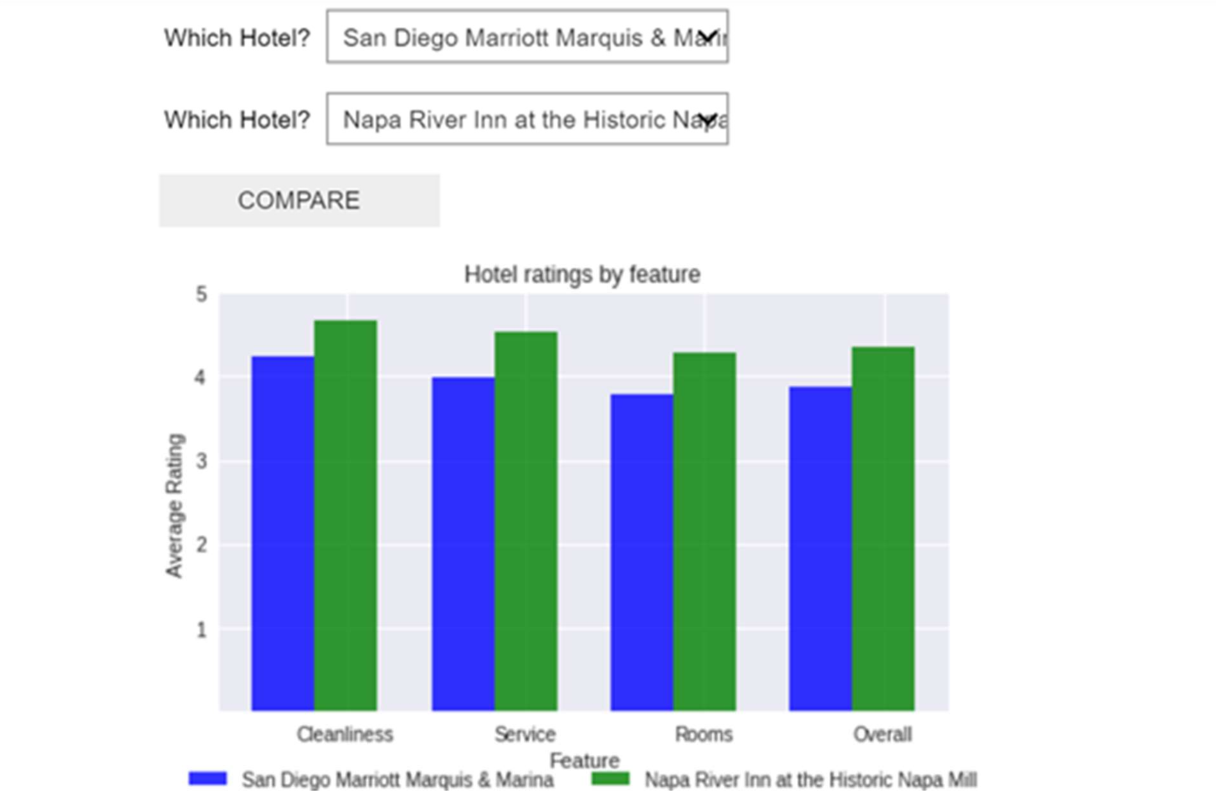
Which Hotel? BEST WESTERN Airport Inn

```
-------------------------- BEST WESTERN Airport Inn ------------------------------
```

3.4. Compare Ratings for 2 Hotels

The next feature implemented is one of the most useful for the customers. It compared the ratings of two hotels based on every feature.

We provided two dropdowns each containing all the hotels. Users can select the hotels they want to compare and a bar graph comparing every feature for the hotels would be plotted.



Which Hotel? San Diego Marriott Marquis & Mari

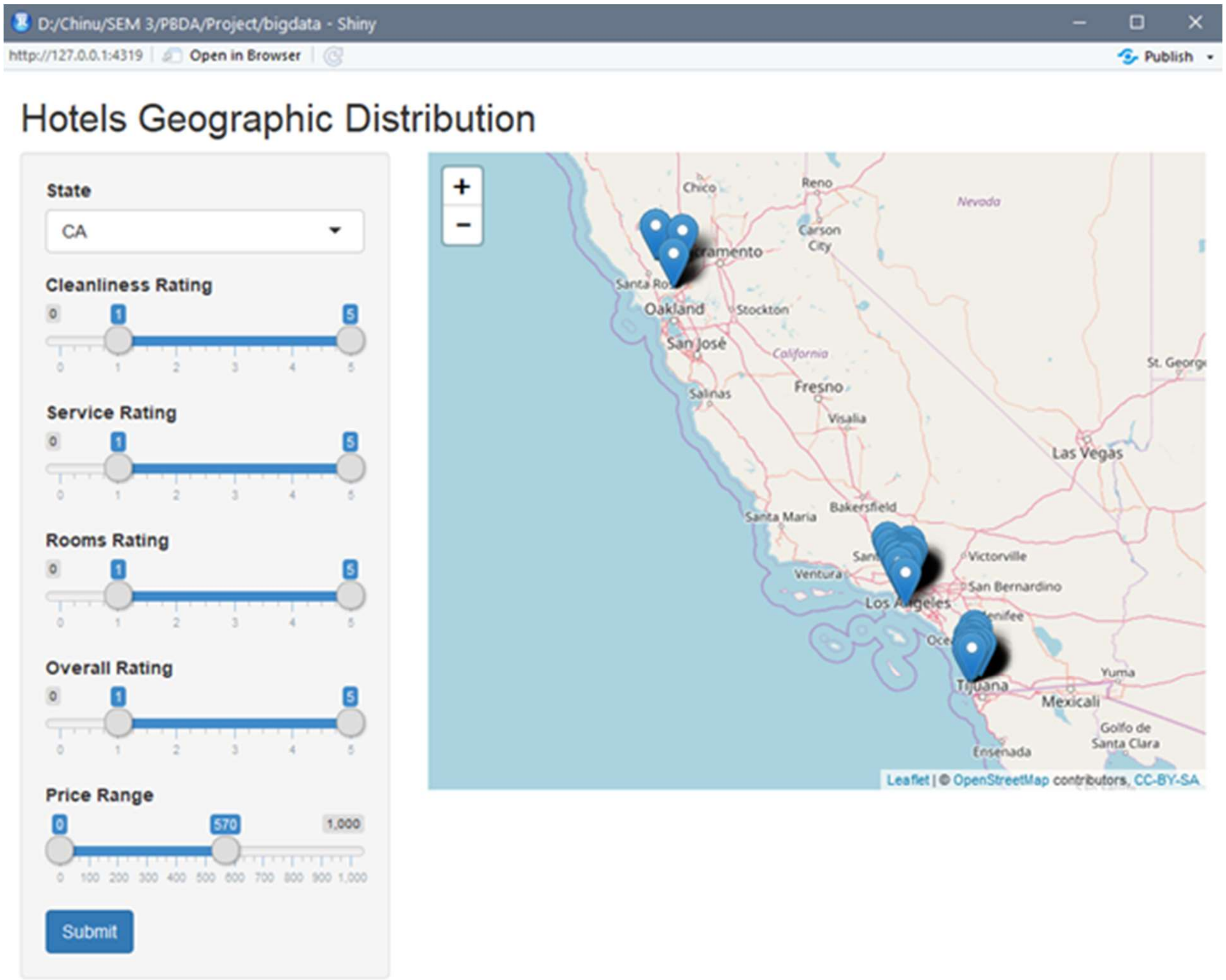
Which Hotel? Napa River Inn at the Historic Napa

3.5 R Shiny Dashboard

Dashboard to filter hotels based on various parameters. The following are the parameters through which the hotels can be filtered in the Dashboard

- State
- Cleanliness Rating(1 to 5)
- Service Rating(1 to 5)
- Rooms Rating(1 to 5)
- Overall Rating(1 to 5)
- Price Range

Clicking submit after selecting the parameters would plot markers on the hotel location. User can get more information about the hotel by clicking on the markers.



Hotels Geographic Distribution

State: CA

Cleanliness Rating: 1 – 5 (range 0–5)

Service Rating: 1 – 5 (range 0–5)

Rooms Rating: 1 – 5 (range 0–5)

Overall Rating: 1 – 5 (range 0–5)

Price Range: 0 – 570 (range 0–1,000)

The above screenshot clearly illustrates the dashboard implemented in RShiny.

4. Code and References

4.1 Code

The is code is present in the Pyspark and Rshiny folders in the submission

4.2 References

- http://times.cs.uiuc.edu/~wang296/Data/
- https://www.citibikenyc.com/system-data
- http://spark.apache.org/docs/2.1.0/api/python/pyspark.sql.html
- http://spark.apache.org/docs/2.1.0/api/python/pyspark.html
- http://stackoverflow.com/
- https://shiny.rstudio.com/tutorial/
- https://github.com/plotly/plotly.py
- https://github.com/mwaskom/seaborn