Finding the Best Hotel in Paris

Tristan Renaud

November 2nd, 2020

Table of Contents

Table of Figures

1 INTRODUCTION

1.1 ASSIGNMENT

This report is part of my capstone project, which is the final requirement to complete my IBM Data Science Professional certification.

Project Information: https://www.coursera.org/learn/applied-data-science-capstone

Certification Information: https://www.coursera.org/professional-certificates/ibm-data-science

1.2 TARGET AUDIENCE

The target audiences are travel websites and agencies. While these companies may utilize personal experience and algorithms to recommend hotels, they do not necessarily consider what is in the immediate vicinity of every hotel in a city. Traditional neighborhoods can be used to generalize what is in the vicinity of a hotel, but neighborhood borders can be vague.

By integrating a customer's interests into the recommendation algorithm travel companies can improve their recommendations and ultimately improve customer satisfaction and retention.

1.3 SCENARIO

Consider the following scenario, which is used throughout the report:

> *I would like to take a trip to Paris and will need to book a hotel. I would like a hotel that is well rated, moderately priced, and near venues I would like to visit.*
>
> *Furthermore, I want to be within walking distance to cafés, nightlife, and entertainment.*

A generalization of the Scenario is,

> **"What is the optimal hotel in Paris given a traveler's budget and travel interests?"**

I decided to look to data science to solve this problem. The solution I developed may be customized to fit anyone's personal travel preferences.

1.4 APPROACH

This problem was solved using data science methods, utilizing rich location data from Foursquare, Yelp, and Nominatim.

The solution generates threes (3) best rated hotels in Paris that match someone's travel interests and budget.

All work for this report can be found in this Jupyter Notebook (coded in Python 3):
https://nbviewer.jupyter.org/github/renautri/best-hotel-in-paris/blob/main/Finding%20the%20Best%20Hotel%20in%20Paris%20-%20Notebook%20-%20Tristan%20S%20Renaud.ipynb

2 DATA

2.1 DATA SUMMARY

The data falls into three categories:

1. Geographical coordinates of Paris.
2. Hotel data.
3. Data for venues in proximity to each hotel.

The data was extracted from the following sources:

Data Source	Data Collected
Nominatim (GeoPy)	Geographical coordinates of Paris.
Foursquare API	Hotel data AND data for venues in proximity to each hotel
Yelp Fusion API	Hotel data

Figure 1. Data collected by data source.

2.2 DATA DETAILS

2.2.1 Latitude and longitude value of Paris.

The geographical coordinates of Paris will be used to center the maps generated using Folium.

Map example:

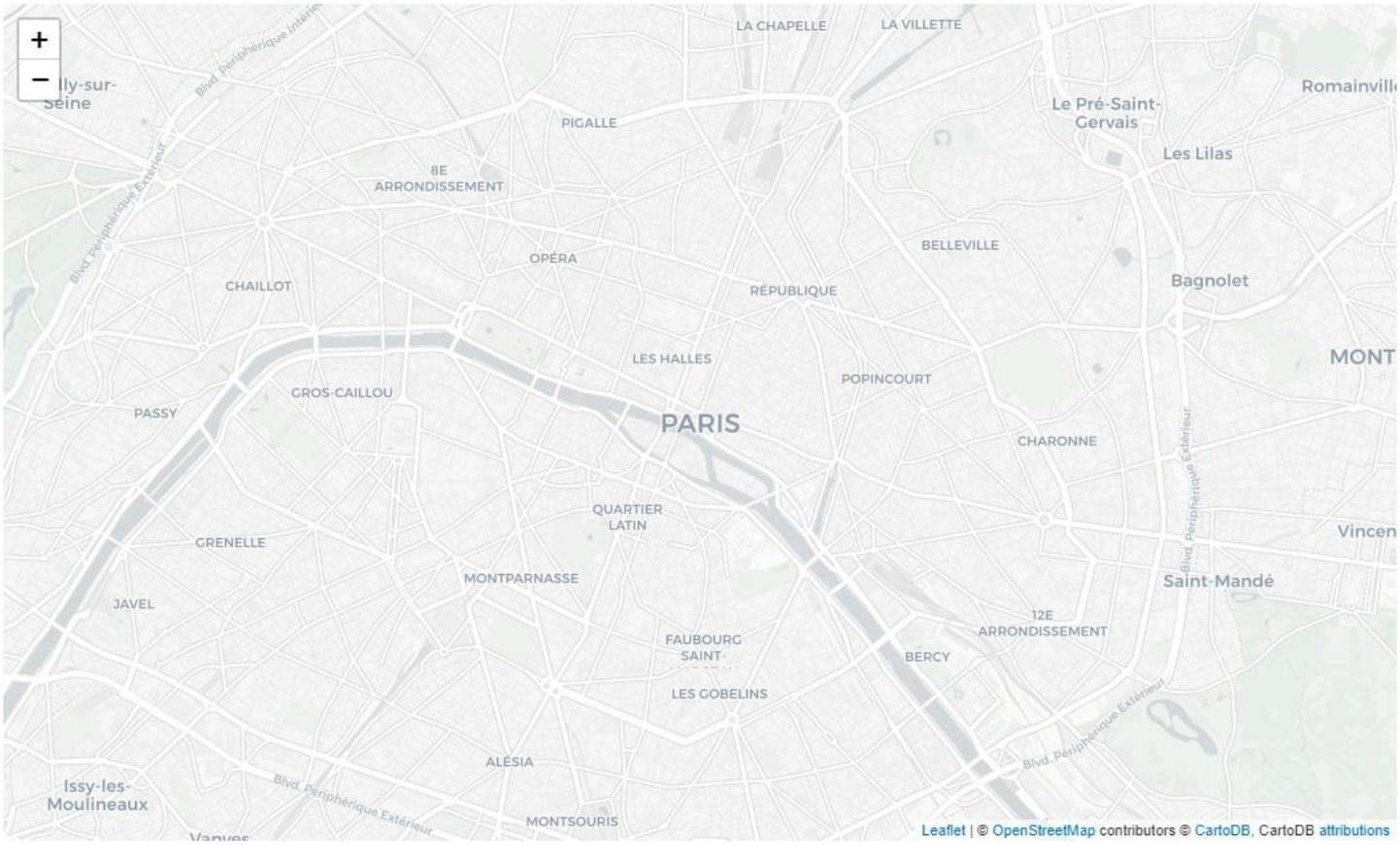


Figure 2. Map of Paris (generated using Folium).

2.2.2 Parisian hotel data

Hotel data is sourced from Foursquare and Yelp. I chose to include Yelp in this analysis because it has pricing information on hotels whereas Foursquare does not.

A combination of hotel information is included, such as ID (unique identifier), address, phone number and URL. This information will keep the data clean and allow others to easily research the hotel and make a reservation.

Summary of Hotel Data:

Source	Attributes	Metrics
Foursquare	id, name, address, city, state, cc, formattedAddress, formattedPhone, url	latitude, longitude, rating, ratingSignals
Yelp	Yelp.id, Yelp.name	Yelp.price, Yelp.rating, Yelp.review_count

Figure 3. Hotel attributes and metrics by source.

Sample of hotel data collected:

	0	1
id	4adcda02f964a520953121e3	4adcd9fff964a520af3021e3
name	Hôtel Le Notre-Dame	Hôtel Les Rives de Notre-Dame
latitude	48.853	48.8533
longitude	2.3465	2.34577
formattedAddress	[1 quai Saint-Michel, 75005 Paris, France]	[15 quai St Michel, 75005 Paris, France]
city	Paris	Paris
state	Île-de-France	Île-de-France
cc	FR	FR
rating	6.9	7.3
ratingSignals	45	8
address	1 quai Saint-Michel	15 quai St Michel
formattedPhone	+33 1 43 54 20 43	+33 1 43 54 81 16
url	http://www.hotelnotredameparis.com	http://www.rivesdenotredame.com
yelp.id	2i5m01j1Sbq9SW8mxN3IVQ	8jT-Pc7mpVaiK6Cpt5LDtg
yelp.name	Hôtel Notre-Dame	Hôtel les Rives de Notre-Dame
yelp.price	2	2
yelp.rating	4	4
yelp.review_count	14	3
yelp.is_claimed	False	True

Figure 4. Sample of hotel data collected from Foursquare and Yelp.

2.2.3 Venues within "walking distance" to each hotel.

Up to 100 venues within 500m (0.31mi) of each hotel were extracted using the Foursquare Places API. Each row represents a venue near a specific hotel.

Sample of venues near hotel:

	Hotel Name	Hotel Latitude	Hotel Longitude	Venue Name	Venue Latitude	Venue Longitude	Venue Category
0	Hôtel Le Notre-Dame	48.85304	2.346504	Shakespeare & Company	48.852568	2.347096	Bookstore
1	Hôtel Le Notre-Dame	48.85304	2.346504	Shiso Burger	48.853376	2.345302	Burger Joint
2	Hôtel Le Notre-Dame	48.85304	2.346504	Place Saint-Michel	48.853345	2.343875	Plaza
3	Hôtel Le Notre-Dame	48.85304	2.346504	Cathédrale Notre-Dame de Paris	48.853124	2.349561	Church
4	Hôtel Le Notre-Dame	48.85304	2.346504	Amorino	48.853012	2.345771	Ice Cream Shop

Figure 5. Sample of venues near each hotel.

2.3 DATA CLEANING

The data was cleaned throughout data collection and transformation. This strategy reduced processing time and kept API calls under quota.

The starting dataset contained 433 venues identified by Foursquare as hotels in Paris.

The following criteria was applied to clean the hotel data:

1. Drop hotels that were not in Paris.
2. Drop hotels without a Yelp match.
3. Drop hotels without a Yelp price.
4. Dedupe for repeating addresses and/or Yelp ids, keeping the first result returned by Foursquare.
5. Drop hotels missing a Foursquare rating and/or a Yelp rating.

The final dataset contained 245 hotels.

3 METHODOLOGY

3.1 *K*-MEANS MACHINE LEARNING

3.1.1 Preparation

After aggregating the venues near each hotel, the data was one-hot encoded to run *k*-means clustering. In addition, it allows manipulations to list the 10 most common venues near each hotel:

	Hotel Name	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	1K Hotel	French Restaurant	Hotel	Bistro	Coffee Shop	Wine Bar	Sandwich Place	Bar	Art Gallery	Restaurant	Clothing Store
1	7 Eiffel Hotel****	French Restaurant	Hotel	Italian Restaurant	Coffee Shop	Plaza	Café	History Museum	Japanese Restaurant	Ice Cream Shop	Cocktail Bar
2	AC Hotel by Marriott Paris Porte Maillot	Italian Restaurant	French Restaurant	Hotel	Japanese Restaurant	Bakery	American Restaurant	Chinese Restaurant	Café	Bus Stop	Lounge
3	Art Hotel Congres	French Restaurant	Restaurant	Bakery	Pizza Place	Comic Shop	Bar	Garden	Gym / Fitness Center	Café	Turkish Restaurant
4	Artus Hotel	French Restaurant	Italian Restaurant	Hotel	Café	Plaza	Pastry Shop	Cocktail Bar	Wine Bar	Clothing Store	Chocolate Shop

Figure 6. Most common nearby venues by hotel.

3.1.2 Finding ideal cluster count (k)

The number of k clusters should be around 10-20 clusters, so it is digestible to the user. If there are too few clusters, there will not be enough options, whereas too many may be difficult for someone to digest.

The distortion score and silhouette score elbow points were used to assist in determining the cluster count.

A cluster count of 11 is selected, based on the silhouette score elbow point.

Distortion score elbow: No elbow point was found.

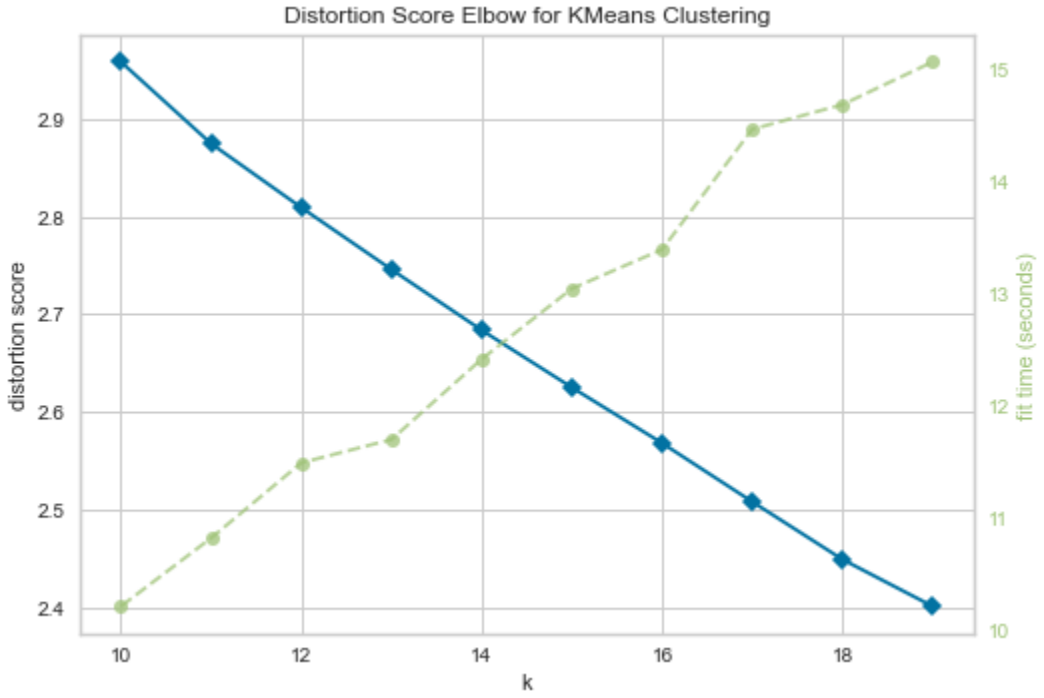


Figure 7. Distortion score and fit time by cluster count, *k*.

Silhouette score elbow: An elbow is found at k=11:

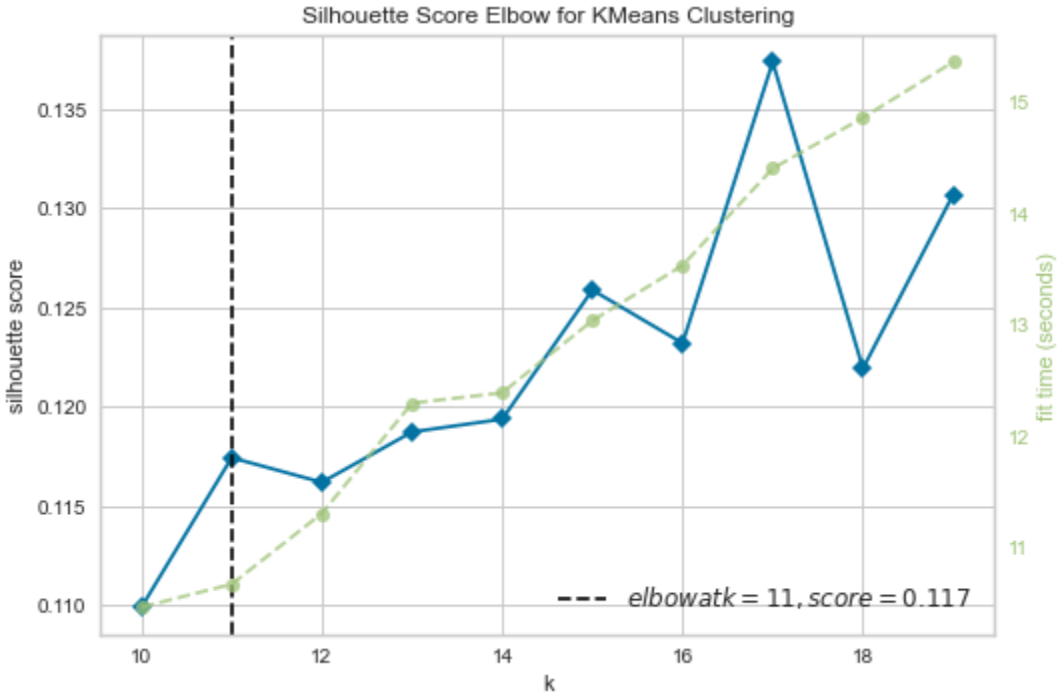


Figure 8. Silhouette score and fit time by cluster count, *k*.

3.1.3 Clustering hotels using *k*-means

Running *k*-means results in the following breakdown of hotels by cluster:

Cluster	Hotel Count
0	31
1	27
2	27
3	25
4	36
5	36
6	6
7	15
8	1
9	11
10	30

Figure 9. Number of hotels in each cluster.

Plotting the hotels on the map by cluster resulted in this map:

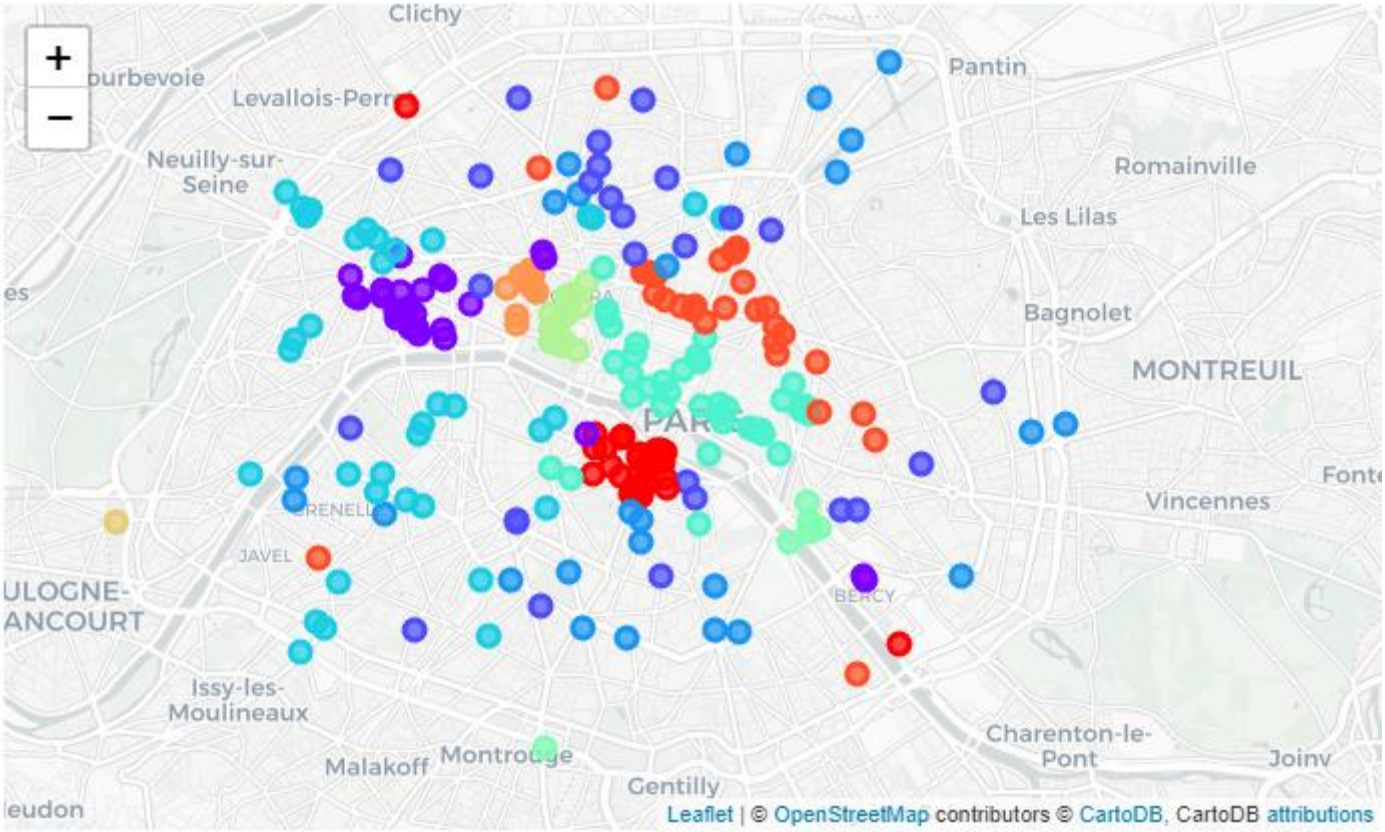


Figure 10. Map of hotels in Paris, each color representing a different cluster.

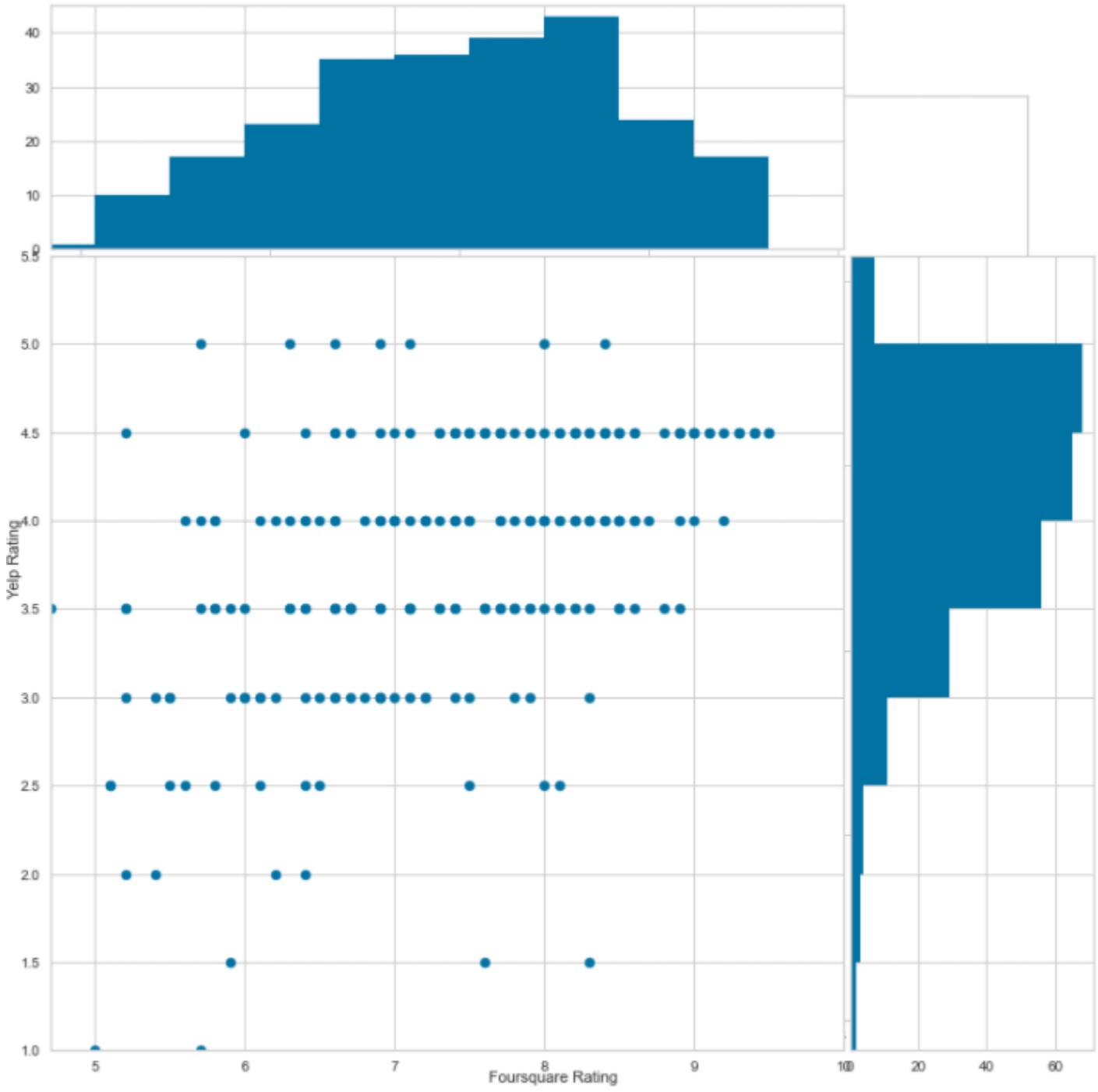
3.2 RANKING HOTELS

After a cluster is selected, there needs to be a way to rank the hotels within that cluster such that the best hotels are returned in the recommendation.

The visual below plots the Foursquare and Yelp ratings against each other for each hotel. In addition, the top of the plot represents the Foursquare rating distribution, and the right represents the Yelp rating distribution of ratings.

Notice how the distributions are shaped differently. The foursquare distribution is more evenly distributed and symmetrical than the Yelp distribution.

Figure 11. Paris hotels plotted by Yelp and Foursquare rating. Plot includes independent distributions of Yelp and Foursquare ratings.



Since the ratings are distributed differently, they need to be transformed in such a way that they can be averaged together and be used to rank the hotels.

To do this, a uniform quantile transformation is applied to the Foursquare and Yelp ratings independently. Then, an average is applied, weighed by the number of ratings on each platform, to calculate final scores.

Now, the top results within a cluster can be generated by filtering on price and cluster.

Top 5 hotels by combined rating:

	Cluster Labels	Hotel Name	foursquare.rating	foursquare.ratingCount	yelp.rating	yelp.reviewCount	foursquare.rating.fit	yelp.rating.fit	combined.rating
195	1	Hôtel Four Seasons George V	9.5	913.0	4.5	104.0	1.000000	0.833333	0.982956
130	9	Hôtel de Crillon	9.5	104.0	4.5	12.0	1.000000	0.833333	0.982759
84	10	The Hoxton Paris	9.4	224.0	4.5	14.0	0.989899	0.833333	0.980689
217	1	The Peninsula Paris	9.4	482.0	4.5	41.0	0.989899	0.833333	0.977625
176	1	Hôtel Plaza	9.4	500.0	4.5	43.0	0.989899	0.833333	0.977501

Figure 12. Top 5 hotels by combined rating.

4 RESULTS

As a reminder, the following scenario will be used to assess results:

> *I would like to take a trip to Paris and will need to book a hotel. I would like a hotel that is well rated, moderately priced, and near venues I would like to visit.*
>
> *Furthermore, I want to be within walking distance to cafés, nightlife, and entertainment.*

Hotel recommendations will be generated using the three constraints highlighted in the scenario. Each will be approached as follows:

1. Well rated – recommendation will output highest rated hotels
2. Moderately priced – recommendation will consider hotels with a price point of 2 out of 4 (with 4 being the highest price)
3. Near venues I would like to visit – the cluster that best matches the scenario's preference will be used to narrow down hotels.

Looking at the top 10 nearby venues by cluster below, which cluster seems most fitting for the scenario?

	0	1	2	3	4	5	6	7	8	9	10
Cluster Labels	0	1	2	3	4	5	6	7	8	9	10
1st Most Common Venue	French Restaurant	French Restaurant	French Restaurant	Hotel	French Restaurant	French Restaurant	Hotel	Hotel	Tennis Court	Hotel	French Restaurant
2nd Most Common Venue	Hotel	Hotel	Hotel	French Restaurant	Hotel	Hotel	French Restaurant	French Restaurant	French Restaurant	Boutique	Bar
3rd Most Common Venue	Plaza	Italian Restaurant	Bar	Bar	Italian Restaurant	Italian Restaurant	Sandwich Place	Japanese Restaurant	Plaza	French Restaurant	Hotel
4th Most Common Venue	Bookstore	Boutique	Italian Restaurant	Italian Restaurant	Bakery	Coffee Shop	Coffee Shop	Boutique	Soccer Stadium	Women's Store	Bistro
5th Most Common Venue	Italian Restaurant	Clothing Store	Bakery	Café	Japanese Restaurant	Bakery	Bakery	Plaza	Sporting Goods Shop	Gourmet Shop	Wine Bar
6th Most Common Venue	Indie Movie Theater	Steakhouse	Bistro	Supermarket	Bistro	Plaza	Nightclub	Jewelry Store	Garden	Clothing Store	Cocktail Bar
7th Most Common Venue	Pub	Café	Japanese Restaurant	Bakery	Plaza	Café	Pizza Place	Chocolate Shop	Italian Restaurant	Department Store	Coffee Shop
8th Most Common Venue	Coffee Shop	Japanese Restaurant	Café	Bistro	Pizza Place	Clothing Store	Cocktail Bar	Bookstore	Supermarket	Sandwich Place	Pizza Place
9th Most Common Venue	Café	Pastry Shop	Pizza Place	Japanese Restaurant	Coffee Shop	Japanese Restaurant	Convenience Store	Clothing Store	Lounge	Bar	Italian Restaurant
10th Most Common Venue	Cocktail Bar	Spa	Restaurant	Pizza Place	Café	Art Gallery	Hotel Bar	Tea Room	Office	Lounge	Restaurant

Figure 13. Top 10 venue categories by cluster.

Cluster 0 looks like a good choice, satisfying the requirements of cafés ('Coffee Shop', 'Café'), nightlife ('Pub', 'Cocktail Bar') and entertainment ('Indie Movie Theater').

The following hotel recommendations are generated after filtering on price and cluster from the hotel ranking:

```
**HOTEL RECOMMENDATIONS**
```

	Hotel Name	address	url	combined.rating
24	Hôtel Dauphine St Germain	36 rue Dauphine	http://hotel@dauphine-st-germain.com	0.732917
36	Hotel Odéon Saint Germain	13 rue Saint-Sulpice	http://www.paris-hotel-odeon.com	0.641274
38	Artus Hotel	34 Rue de Buci	http://www.hotelsmauricehurand.com	0.619895

Figure 14. Hotels recommended in report's scenario.

Recommended hotels on the map:

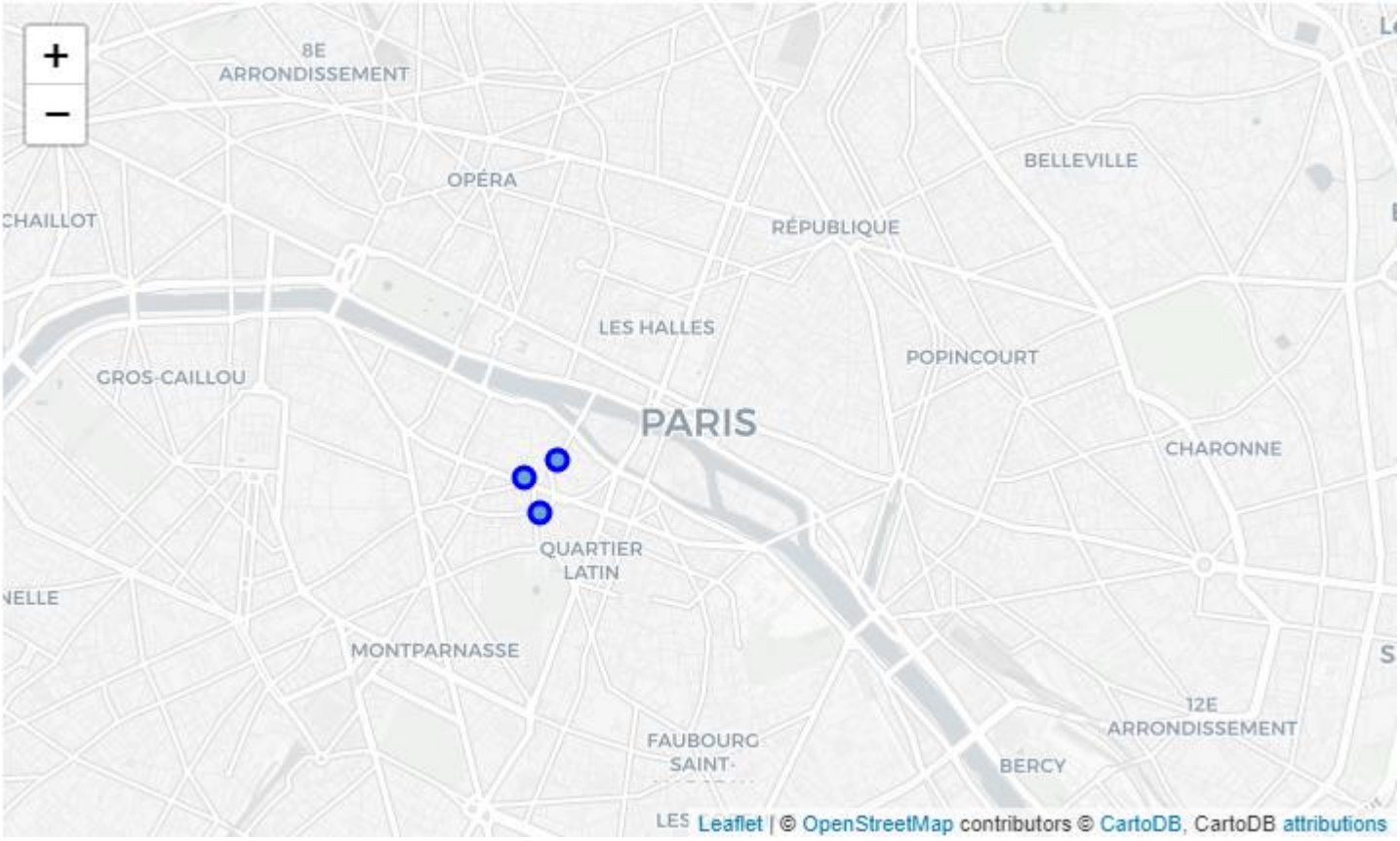


Figure 15. Map of recommended hotels.

5 DISCUSSION

5.1 OBSERVATIONS

1. **Not all clusters are created equal** – As seen in **Figure. 9**, cluster size varies from 1 to 36 hotels. If small clusters were selected, there may not be 3 hotels to return as a recommendation or the hotels recommended are less likely to be well rated.

2. **Not all clusters form geographical neighborhoods** – If they all created geographical neighborhoods, hotel selection based on existing defined neighborhoods would be the way to go.

3. **"Hotel" and "French Restaurant" are the most common venues** – This tends to make sense, given that Paris is a city and in France. There was concern that including these venues would skew the results. However, the decision was made to keep these venues because they do, in fact, impact what can be found around each hotel in this report. If all hotels and French restaurant venues were removed, then the k-means algorithm may find a hotel that originally did not have any hotels or French restaurants nearby as similar to one that had a lot of hotels and French restaurants nearby.

5.2 Potential Future Exploration

1. **Application to other cities** – Is the analysis in this report transferrable to other cities? If so, how well? Are manipulations required besides the initial information (coordinates of the city of interest and name of city)?

2. **Expand beyond nearby venues** – Consider additional parameters such as proximity to public transportation, distance to city center, distance to specific points of interests (Louvre, Eiffel tower, etc.). Do they improve the recommendation?

3. **Evaluation, Deployment and Feedback** – This report omits the Evaluation, Deployment and Feedback steps found in the Foundational Methodology for Data Science. Unfortunately, they are out of the scope of this report as it would require physically visiting several hotels and identifying if the nearby venues match the original interests of a tourist.

5.3 Constraints and Shortfalls

The following constraints and shortfalls are worth acknowledging. With this information, this report's methods can be improved upon, given more time and resources.

1. **Covid-19 impact** – This report was produced during the Covid-19 epidemic. Consumer and business behavior were irregular and impacted the signals Foursquare uses to provide location data. This data may provide different results once the epidemic is over. Further information on changes in behavior: https://enterprise.foursquare.com/intersections/article/understanding-the-impact-of-covid-19/

2. **Free-tier API limits** – One big constraint with this project was sticking to a low volume of API calls. With higher limits, the following improvements can be achieved:

    a. **More hotels** – a greater pool of hotels gives more choice and accuracy potential.

    b. **More nearby venues** – Why stop at 100 nearby venues? Expanding to 200 could help paint a better picture of what is around each hotel.

    c. **More hotel information** – additional useful information can be passed to the user, such as tips, photos, and even events happening at the hotel.

6 Conclusion

This report illustrates how data science methods can be used to improve travel recommendations for someone looking for a hotel in a particular city that is within their budget and is in proximity to venues that interest them.

By considering hotels in Paris, along with venues nearby each one, clusters of hotels can be generated, which can help narrow down which immediate neighborhood a tourist would like around their hotel.

Applying a modified hotel rating system and filtering by price range on to the hotels in the selected cluster allows for the best hotels to be returned as a recommendation.

Interested in trying this out for yourself? Check out the GitHub in the references below!

7 REFERENCES

1. GitHub Repository: https://github.com/renautri/best-hotel-in-paris
2. Foursquare Places API: https://developer.foursquare.com/places
3. Yelp Fusion API: https://www.yelp.com/fusion
4. Nominatim (OpenStreetMap): https://nominatim.openstreetmap.org/
5. CARTO ("Positron with labels" basemap): https://carto.com/help/building-maps/basemap-list/
6. Python Libraries: Pandas, NumPy, Folium, Yellowbrick, Scikit-learn, Matplotlib, Json, Requests, GeoPy and Math