Day Tour Recommendation Service

2020

APRIL 4

Authored by: Sabari Girish Parampoor

Contents

Highlights

Introduction

Background

We all love our weekends and holidays, and a lot of us like to make the most of them by exploring new places in the city based on our unique interests. Wouldn't it be nice if we can quickly get some recommendations on the things to do in a city, based on our interests and preferences? Day tour recommendation service aims to provide recommendations about the places you can visit in a day, near a given starting location and within a desired distance radius. The service takes into account your things of interests like food, sight-seeing, outdoor activities, entertainment etc. and recommends places you can visit on a given day.

The service will then group the different places of interest and plot each group on a map for better visualization and informed decision-making.

Problem

This exercise aims to explore San Francisco (SF) for recommendations on things to do based on given interests and preferences.

Problem Statement

I am on an official trip to San Francisco to attend a conference, staying at JW Marriot. I have a day off and I want to go on a day tour to explore SF. Being a foodie and a nature lover, I want to explore nature attractions and have some good food along the way. **Recommend me places to visit in SF within a radius of 50Kms from where I stay.**

My Interests and Preferences:

1. I am interested in visiting one the following nature attractions:
   a. Scenic lookout
   b. Waterfall
   c. Lake
   d. Beach
2. I am interested in trying one of these cuisines:
   a. Indian
   b. Italian
   c. Mexican
   d. Ethiopian
3. I am staying at JW Marriot, recommend me places within 50kms of my stay

Target Audience

This service aims to target **day trippers** who are looking to explore a city during the day time and be back by the evening.

Data Analysis

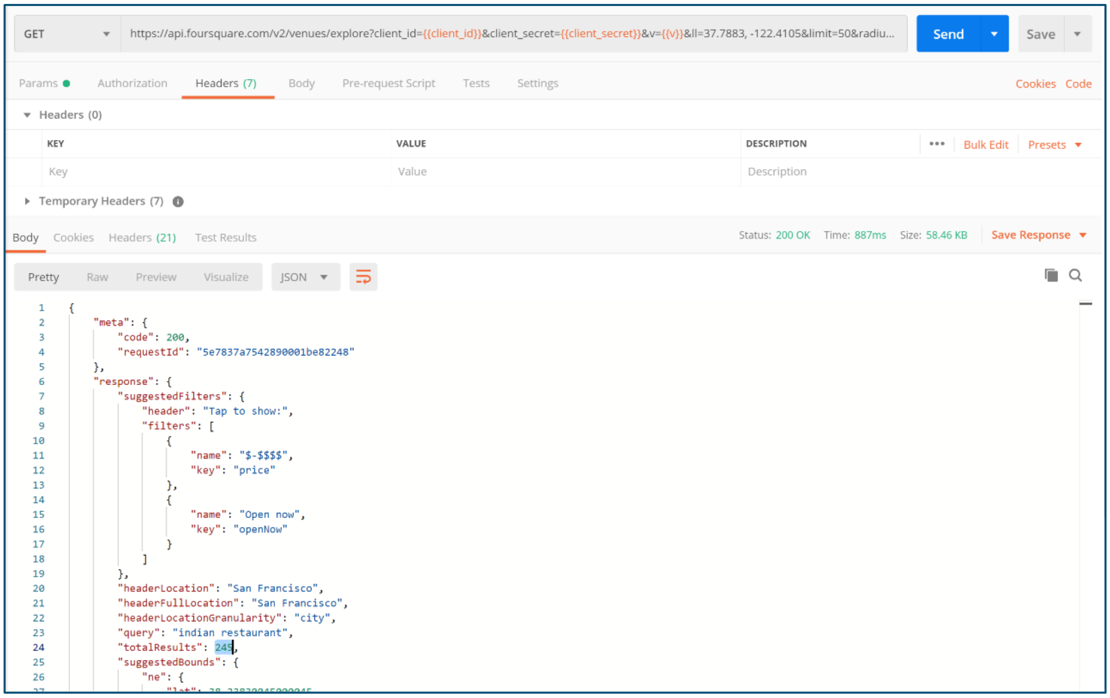
Data Source and Considerations

We will use Foursquare API as our primary source of data. We will rely on the "Get Venue Recommendations" API to fetch relevant data. Following are the important parameters we will be passing to the API:

1. **ll** --> This is the coordinates of the starting location (in this case 37.7883, -122.4105 for JW Marriot)
2. **radius** --> This is the maximum distance from the starting location (in this case 50kms)
3. **categoryId or query** --> Comma-separated list of interested categories of places to visit (food, outdoors etc.) or specific text to search within places (Indian restaurant etc.)

Following are some considerations while working with the Foursquare API:

1. Separate API calls to fetch data for each cuisine of interest
   We will be making separate calls to the API for each cuisine of interest as otherwise the data is mixed up and data set for each cuisine is drastically reduced. We will aim to get as much relevant data as possible. For example, to fetch Indian Restaurants near JW Marriot, we will use query parameter ('Indian Restaurant') instead of categoryId for food ('4d4b7105d754a06374d81259').

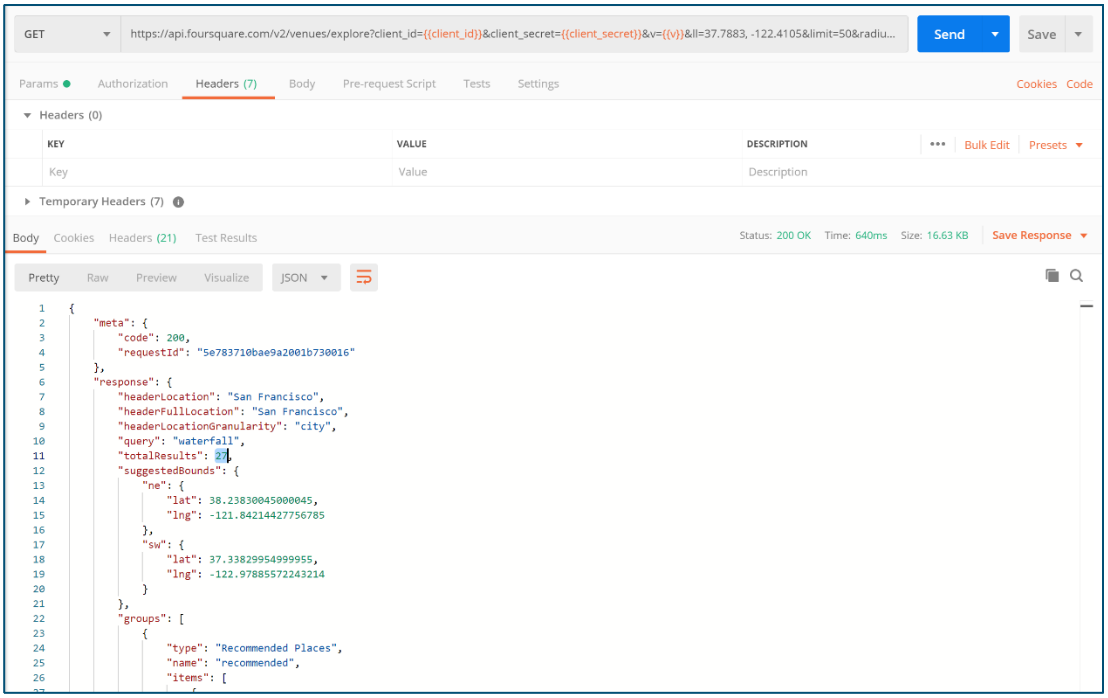


GET https://api.foursquare.com/v2/venues/explore?client_id={{client_id}}&client_secret={{client_secret}}&v={{v}}&ll=37.7883, -122.4105&limit=50&radiu...

Status: 200 OK Time: 887ms Size: 58.46 KB

```
{
    "meta": {
        "code": 200,
        "requestId": "5e7837a7542890001be82248"
    },
    "response": {
        "suggestedFilters": {
            "header": "Tap to show:",
            "filters": [
                {
                    "name": "$-$$$$",
                    "key": "price"
                },
                {
                    "name": "Open now",
                    "key": "openNow"
                }
            ]
        },
        "headerLocation": "San Francisco",
        "headerFullLocation": "San Francisco",
        "headerLocationGranularity": "city",
        "query": "indian restaurant",
        "totalResults": 245,
        "suggestedBounds": {
            "ne": {
```

2. Separate API calls to fetch data for each nature attraction of interest
   Similar to cuisines/ food, we will be making separate calls to the API for each nature attraction of interest. For example, to fetch Waterfalls near JW Marriot, we will use query parameter ('Waterfall') instead of categoryId for outdoors ('4d4b7105d754a06377d81259').



GET https://api.foursquare.com/v2/venues/explore?client_id={{client_id}}&client_secret={{client_secret}}&v={{v}}&ll=37.7883, -122.4105&limit=50&radiu...

Headers (0)

KEY	VALUE	DESCRIPTION
Key	Value	Description

Temporary Headers (7)

Status: 200 OK Time: 640ms Size: 16.63 KB

```
{
    "meta": {
        "code": 200,
        "requestId": "5e783710bae9a2001b730016"
    },
    "response": {
        "headerLocation": "San Francisco",
        "headerFullLocation": "San Francisco",
        "headerLocationGranularity": "city",
        "query": "waterfall",
        "totalResults": 27,
        "suggestedBounds": {
            "ne": {
                "lat": 38.23830045000045,
                "lng": -121.84214427756785
            },
            "sw": {
                "lat": 37.33829954999955,
                "lng": -122.97885572243214
            }
        },
        "groups": [
            {
                "type": "Recommended Places",
                "name": "recommended",
                "items": [
```

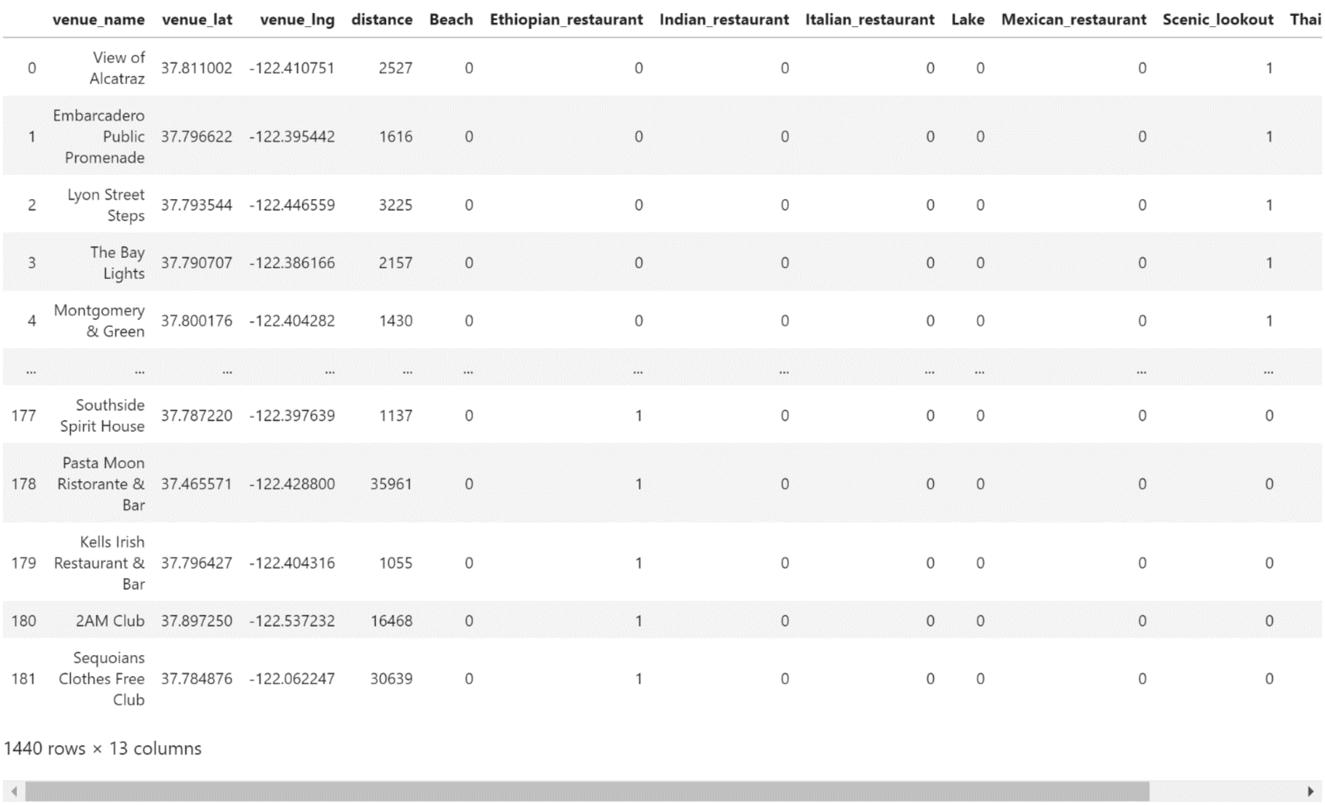
3. Distance from starting location and pagination
   Each of the above API calls will pass 'radius' parameter to ensure only things of interest within a specified distance are considered. Each API call will also handle pagination for fetching multiple pages of the result.

Feature Selection and Data Structure

As we can see from sample responses, there a few attributes that need to be ignored from our processing. Below table lists the attributes of Venue we will extract from the response; rest will be ignored:

Venue Attribute (Feature)	**Path**	**Explanation**
name	response.groups.items.venue.name	Name of the venue
lat	response.groups.items.venue.location.lat	Latitude of the venue; required for plotting venue on the map
lng	response.groups.items.venue.location.lng	Longitude of the venue; required for plotting venue on the map
distance	response.groups.items.venue.location.distance	Distance of the venue from starting point; required for grouping venues

We will store these attributes in a Python DataFrame having below columns:



	venue_name	venue_lat	venue_lng	distance	Beach	Ethiopian_restaurant	Indian_restaurant	Italian_restaurant	Lake	Mexican_restaurant	Scenic_lookout	Thai
0	View of Alcatraz	37.811002	-122.410751	2527	0	0	0	0	0	0	1	
1	Embarcadero Public Promenade	37.796622	-122.395442	1616	0	0	0	0	0	0	1	
2	Lyon Street Steps	37.793544	-122.446559	3225	0	0	0	0	0	0	1	
3	The Bay Lights	37.790707	-122.386166	2157	0	0	0	0	0	0	1	
4	Montgomery & Green	37.800176	-122.404282	1430	0	0	0	0	0	0	1	
...	...	...	...	...	...	...	...	...	...	...	...	
177	Southside Spirit House	37.787220	-122.397639	1137	0	1	0	0	0	0	0	
178	Pasta Moon Ristorante & Bar	37.465571	-122.428800	35961	0	1	0	0	0	0	0	
179	Kells Irish Restaurant & Bar	37.796427	-122.404316	1055	0	1	0	0	0	0	0	
180	2AM Club	37.897250	-122.537232	16468	0	1	0	0	0	0	0	
181	Sequoians Clothes Free Club	37.784876	-122.062247	30639	0	1	0	0	0	0	0	

1440 rows × 13 columns

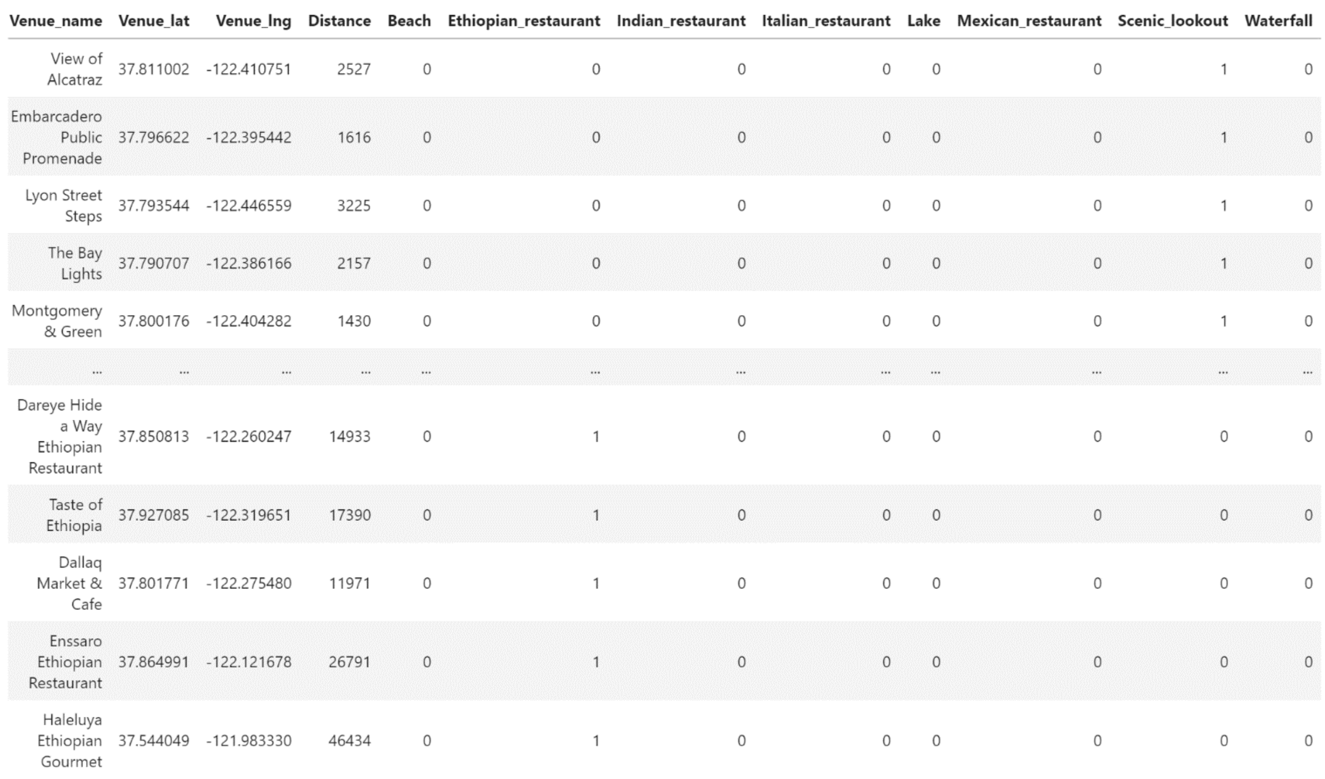
Note: We will use indicator variables (like Scenic_lookout, Waterfall etc.) instead of a category variable (like venue_type)

Methodology

Exploratory Analysis

We will start by acquiring required relevant data using Foursquare API. We will need to clean the data and prepare it before it can used for clustering. Below steps are carried out towards data preparation:

1. **Extract relevant attributes** of venue from Foursquare API responses
2. **Merge different data sets** to consolidate data for all venue types
3. **Create indicator values** for categorical columns



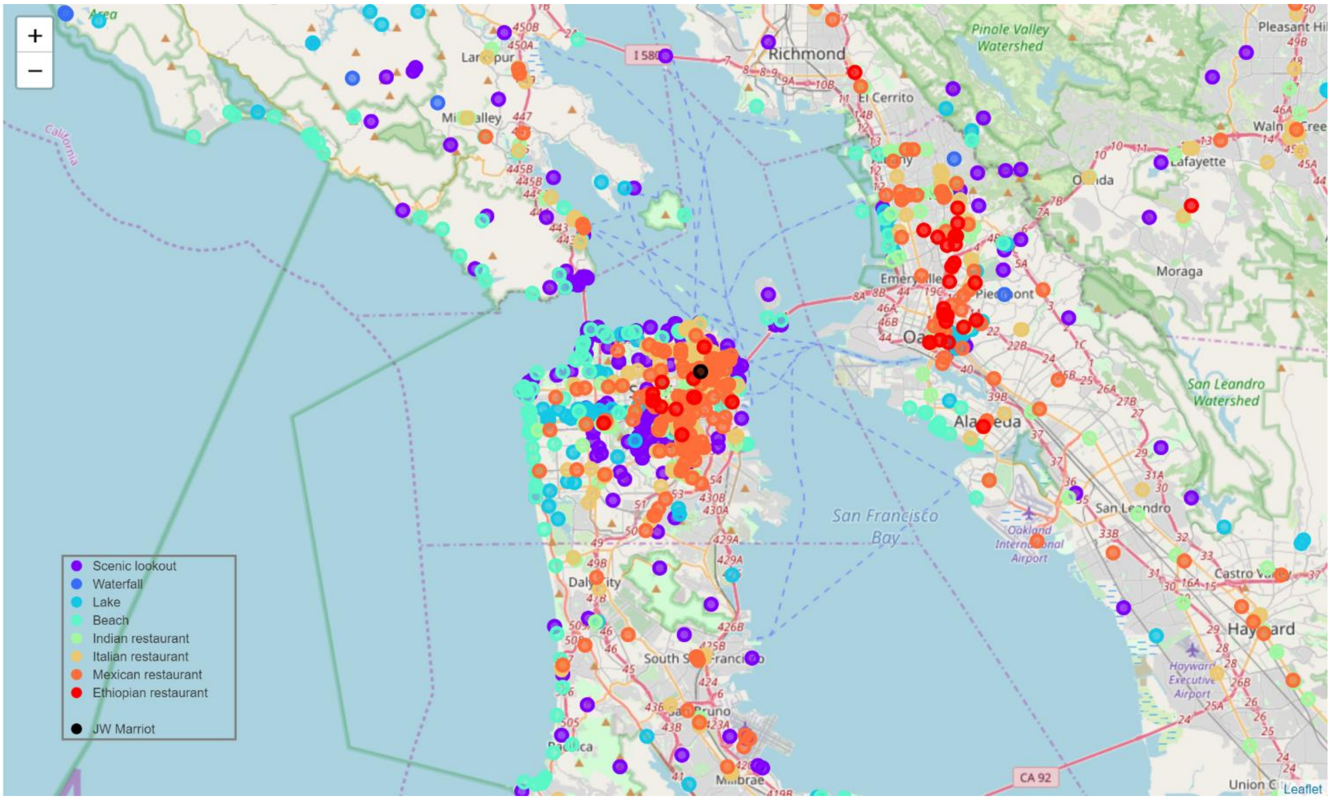
Venue_name	Venue_lat	Venue_lng	Distance	Beach	Ethiopian_restaurant	Indian_restaurant	Italian_restaurant	Lake	Mexican_restaurant	Scenic_lookout	Waterfall
View of Alcatraz	37.811002	-122.410751	2527	0	0	0	0	0	0	1	0
Embarcadero Public Promenade	37.796622	-122.395442	1616	0	0	0	0	0	0	1	0
Lyon Street Steps	37.793544	-122.446559	3225	0	0	0	0	0	0	1	0
The Bay Lights	37.790707	-122.386166	2157	0	0	0	0	0	0	1	0
Montgomery & Green	37.800176	-122.404282	1430	0	0	0	0	0	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...
Dareye Hide a Way Ethiopian Restaurant	37.850813	-122.260247	14933	0	1	0	0	0	0	0	0
Taste of Ethiopia	37.927085	-122.319651	17390	0	1	0	0	0	0	0	0
Dallaq Market & Cafe	37.801771	-122.275480	11971	0	1	0	0	0	0	0	0
Enssaro Ethiopian Restaurant	37.864991	-122.121678	26791	0	1	0	0	0	0	0	0
Haleluya Ethiopian Gourmet	37.544049	-121.983330	46434	0	1	0	0	0	0	0	0

Let’s extract some data from the above dataset and do some initial analysis.

	Total	Min Distance (Km)	Max Distance (Km)
Beaches	179	1.0	48.5
Lakes	115	3.6	49.7
Scenic lookouts	212	0.6	49.4
Waterfalls	18	0.8	42.4
Ethiopian Restaurants	42	0.5	46.4
Indian Restaurants	245	0.3	49.7
Italian Restaurants	276	0.1	49.2
Mexican Restaurants	248	0.1	49.1

As we can see from the data above, JW Marriot has plenty of attractions around it and **most of the venue types can be reached within a radius of 3.6 Kms**. Also, there are a good number of options for various cuisines of interest, with Italian being the most available cuisine while Ethiopian having fewer options.

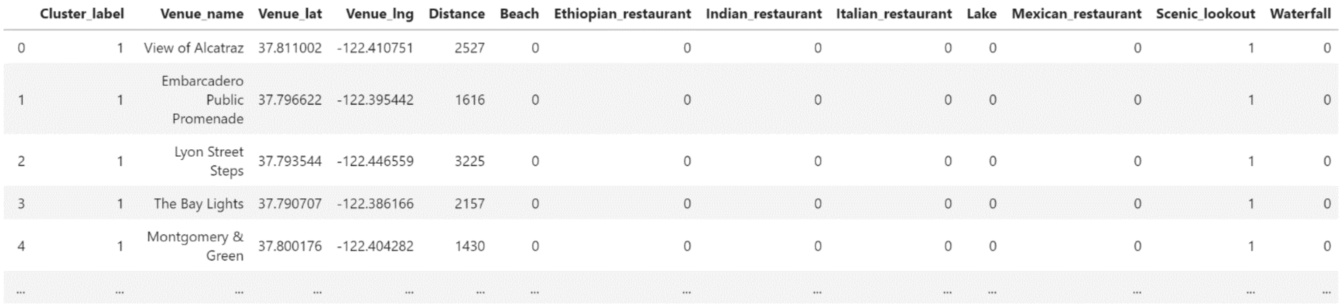
Now let’s plot these points on a map to understand their proximities from our starting point (JW Marriot).



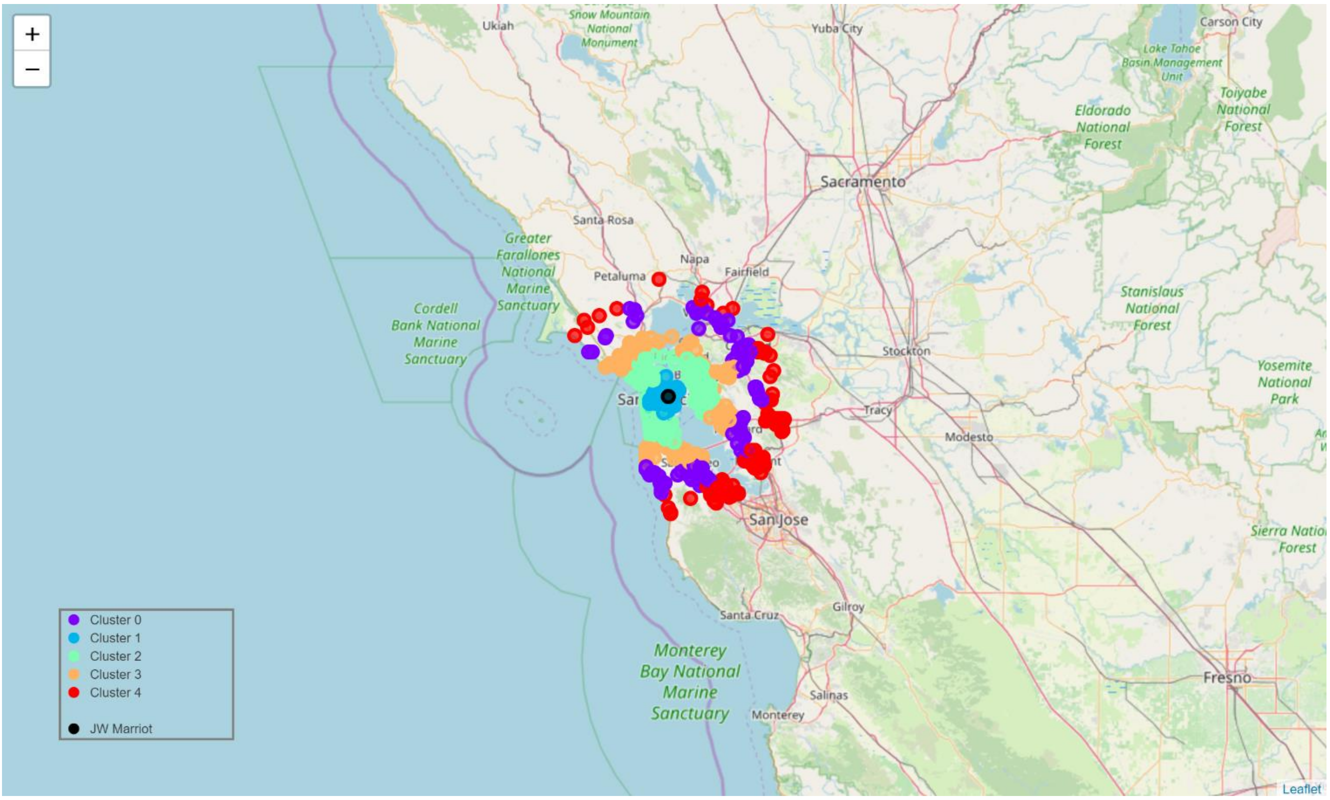
As we can see, the map just confirms the data points we discussed above. **JW Marriot has numerous restaurants of different cuisines immediately surrounding it**, and as we start traveling, we have plenty of natural attractions like beaches, lakes and scenic lookouts. Comparatively, **there are fewer waterfalls in the surroundings, and one needs to travel to neighboring cities to enjoy them**.

Clustering using K-Means

Now the immediate question is whether we can group these points based on their distances from the starting location and see if each group consists of a good mix of these points of interests for the user to choose. Let's look at clustering. **We will use K-Means to split the places of interest into 5 clusters**, below is the resulting cluster labels and map:



	Cluster_label	Venue_name	Venue_lat	Venue_lng	Distance	Beach	Ethiopian_restaurant	Indian_restaurant	Italian_restaurant	Lake	Mexican_restaurant	Scenic_lookout	Waterfall
0	1	View of Alcatraz	37.811002	-122.410751	2527	0	0	0	0	0	0	1	0
1	1	Embarcadero Public Promenade	37.796622	-122.395442	1616	0	0	0	0	0	0	1	0
2	1	Lyon Street Steps	37.793544	-122.446559	3225	0	0	0	0	0	0	1	0
3	1	The Bay Lights	37.790707	-122.386166	2157	0	0	0	0	0	0	1	0
4	1	Montgomery & Green	37.800176	-122.404282	1430	0	0	0	0	0	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...



As we can see, the map exhibits a nice **pattern of concurrent circles, keeping the starting location (black marker) at the center**. We have tried to create five clusters primarily based on distances, now let's see if we are able to find a good mix of points of interests in each of these clusters.

Results and Discussion

	Distance (Kms)	Scenic	Waterfalls	Lakes	Beaches	Indian Food	Italian Food	Mexican Food	Ethiopian Food
Cluster 1	0.1 - 8.6	108	7	30	47	78	123	122	13
Cluster 2	8.6 – 19.0	46	3	33	60	47	52	61	27
Cluster 3	19.1 - 29.4	34	2	18	37	27	26	18	1
Cluster 0	30.1 - 40.0	14	4	14	27	29	45	23	0
Cluster 4	40.5 – 49.7	10	2	20	8	64	30	24	1

As we can see, each group contains a relatively **good mix of natural attractions and restaurants of interest**. Through our Clustering model, we have tried to spread most of the attractions and restaurants evenly between the clusters. One exception being Ethiopian restaurants, as they are spread between Cluster 1 and Cluster 2, due to concentration of them being located between 0.1 – 19.0 Kms. From the summary above, we are able to present a clear picture of where the places/ restaurants of interest are, making it an easy decision for the user to pick a cluster and go on a day-tour. **Cluster 1 is clearly my choice for a day trip! What's yours?**

Conclusion

In this exercise we set out to recommend places of interest for a day-tour based on certain preferences. Data Science is all about telling a compelling story with data using various tools (visualization, machine learning etc.), allowing stakeholders to make a decision. **Effectiveness of the Data Science methodology can be judged by the clarity in the story telling and the ease in which a decision can be made. Here using exploratory analysis and clustering, we are able to present data with utmost clarity, leaving the user to make an easy decision. As can be seen from the summary table, Cluster 1 is the most straightforward choice from JW Marriot for the given preferences**, due to the following reasons:

1. It has an excellent mix of places and restaurants of interest
2. It is the nearest cluster from JW Marriot, and the user has to travel a radius of just 8.6 Kms to enjoy these attractions

Cluster 1 is followed by Cluster 2 and can be preferred by users who like beaches more, and it also has a greater number of Ethiopian restaurants to choose from. Cluster 1 and Cluster 2 are the clear top choices.

Future Possibilities

The Day Tour Recommendation Service can be easily extended to **cover more places of interest**, and when it comes to restaurants, **price-based and ratings-based filtering** become a possibility too. It can be **made generic to support any city in the world** and has the potential to become a well-rounded service. Possibilities about **integrating other location-based APIs** can also be explored to offer wide array of preferences.

When it comes to usability, the service can be **integrated with a Maps platform** (Google Maps etc.) and can help users navigate to each of the attractions inside a cluster of choice. The service can be further expanded to include **social collaboration features** for increased user engagement. **Possibilities exist in offering users a truly end-to-end product that they will love to use and recommend.**