Capstone Project

Using Foursquare API and Clustering Algorithms for Trip Planning

By Furqan Tariq

Version 2.0 (Week 5)

Contents

1. Introduction 3
2. Data Description 3
    2.1. User Input Data 3
    2.2. Data Fetched from Foursquare 3
    2.3. Output Data 3
3. Methodology 4
    3.1. Project Methodology 4
    3.2. Exploratory Analysis and Data Cleaning 4
    3.3. Model Building 6
    3.4. Evaluation 6
4. Results 7
5. Discussion 11
6. Conclusion 12

1. Introduction

The project is inspired by my personal planned vacation to Italy in the summers of 2020. For anyone that has travelled abroad, they will know that there is frantic planning of places to see, things to do, experiences to try – all in a specified number of days. Planning usually involves scouring Google and finding recommendations but this can take a lot of time and effort.

The idea behind this project is to develop a methodology that would be easily replicable and scalable for any such trip, and provide the foundational information for trip planning in a minimal amount of time. Given some inputs (discussed in later sections), the program provides visualization of the itinerary on the map of the target city.

The project not only can appeal to travellers, but it can also be the foundation for development of any trip planning application. Right now, the scope of the project is limited to just clustering the places to see but in the future this can be aggregated with transportation data, budget planning tools, and operating hours of target places to provide a holistic experience to end users.

In terms of Capstone Project, this is not the typical scenario of comparing neighbourhoods. However, the techniques learned in the course are all applicable here and therefore this project falls in the scope of the Capstone Project.

2. Data Description

2.1. User Input Data

Data required from the user for the project is:

- Target city e.g. Florence, Italy
- Number of days to spend in target city e.g. 4 days
- Venue categories e.g. museums, landmarks, parks etc.
- Foursquare credentials

2.2. Data Fetched from Foursquare

Data that is fetched from Foursquare is following:

- Venues of the specified categories for the specified target city. Refer to the following article for details of this search: https://developer.foursquare.com/docs/api/venues/search
- Venue details (specifically their rating) for each venue from the above search. Refer to the following article for the details of this search: https://developer.foursquare.com/docs/api/venues/details

2.3. Output Data

The output data of the project is:

- Cluster labels for all the landmarks/places-to-see (each cluster label refers to the day of the vacation) and visualization of all landmarks/places-to-see with their respective colour differentiation on a map
- Itinerary listing all the landmarks/places-to-see for each day and sorted by their ratings

3. Methodology

This section describes the steps that were taken to achieve the objectives highlighted in the Introductory section.

3.1. Project Methodology

The methodology followed for the project was:

1) Define objectives (Introductory section of this report)
2) Identify data sources (Introductory section of this report)
3) Exploratory analysis
4) Data cleaning/pre-processing
5) Model building
6) Evaluation

3.2. Exploratory Analysis and Data Cleaning

1) First the Foursquare API was used for searching venues. Following parameters were defined for the request URL:
    a. Foursquare Client ID
    b. Foursquare Client Secret
    c. Longitude and Latitude
        i. This is extracted using ***geopy*** library. The library provides the latitude and longitudes of the target city i.e. Florence, Italy.
    d. Foursquare Version
    e. Radius of the search
        i. This was set at 1km as for me (as a tourist) I will only be comfortable to see venues in this vicinity
    f. Limit
        i. This was set at 100 as a safe number. Obviously, the larger the number the more hectic the schedule. But I would filter the venues at a later stage rather than at this point in time.
    g. Categories
        i. The categories for the venues I am interested in are Museums and Landmarks.
        ii. Their codes are '4bf58dd8d48988d181941735,4bf58dd8d48988d12d941735'
        iii. These codes for categories can be seen from: https://developer.foursquare.com/docs/resources/categories

    The URL for this request can be seen in the notebook.

2) Once the response was received it was converted to a dataframe using json_normalize. The shape of the dataframe was found to showcase how many venues were originally fetched from Foursquare

3) The dataframe was then pre-processed:
    a. The categories column of the dataframe was split and the 'name' portion of that variable was extracted
    b. The cleaned column of categories was further analysed by using value_counts(). This highlighted the number of sub-categories of venues in the dataset; at this point in time, one can refine the categories selection in the API call if they are not happy with the kind of results in the dataset
    c. Only columns of interest were kept in the dataset i.e. venue_id, venue_name, categories, latitude and longitude of the venue.
    d. Column names were renamed where there was 'location.' in the name of the columns
4) Once the venues were fetched and the data cleaned, the next call to Foursquare was for the details of each venue in the dataset. The purpose was to extract the rating of that venue which would aid in deciding which venues to go to. The parameters for that call were:
    a. Foursquare Client ID
    b. Foursquare Client Secret
    c. Foursquare Version
    d. Venue ID
        i. This was extracted from the dataset that was in the first call.
5) For every call, the 'ratings' parameter was extracted from the response. In some cases, there was no ratings for a venue. The rating was then substituted with np.nan
6) Since the details of the venue is a premium call, the program has an additional check:
    a. The very first time, the quota is not exceeded, and hence the dataset is extracted from Foursquare. The program merges the ratings information with the original dataset. This merged dataset is then exported to a CSV file. The CSV file can be referred to in case the quota for the day has exceeded and the user wants to still test different scenarios
    b. In case the user is running the program multiple times and the quota is exceeded, then the program loads the CSV file (as mentioned above).
7) The merged dataset (with ratings) goes through further pre-processing
    a. Venues where there is no rating provided are removed from the dataset – it's a risky proposition to go to a place where there have been no ratings.
    b. The shape of this pre-processed dataset is analysed to see how many venues remain
8) The next step in pre-processing/data cleaning is setting a ratings threshold. The user can set the threshold so that all those venues that satisfy that rating are included. For instance, a user may only want to visit venues that are at least 8.0 out of 10.
9) The next step in the exploratory phase is to visualize these final venues on a map using Folium. This gives the user a sense of potential places to see in actuality.
10) The final step carried out is a scatter plot of the longitudes and latitudes of the venues to provide a visualization of how closely clustered these venues are geospatially.

3.3. Model Building

Two machine learning algorithms have been used for the purpose of clustering:

1) K-means
2) Gaussian Mixture

For K-means, the following steps were taken:

1) Made a copy of the dataset with only latitudes and longitudes as clustering only works on numerical data
2) Sklearn was used to preprocess and transform the dataset
3) The number of clusters was defined as the target number of days in that city – this is a user input. The notebook has this variable set at 4 days in Florence.
4) K-means algorithm is run on the transformed coordinates data and the labels are generated
5) The labels are appended to the original dataframe
6) Visualization of the venues is done on a map using Folium and ColorMap

Based on the results of K-means, it was decided to use Gaussian Mixture algorithm (the results section will detail why). Gaussian Mixture has higher efficacy than K-means. The reader is encouraged to read: https://towardsdatascience.com/gaussian-mixture-models-d13a5e915c8e

For Gaussian Mixture, the following steps were taken:

1) Made a copy of the dataset with labels (so that new labels could be compared)
2) Sklearn was used to set up the classifier
3) The number of components was defined as the target number of days in that city – this is a user input. The notebook has this variable set at 4 days in Florence.
4) The labels from this classifier were appended to the dataset
5) The value_counts was used to see how the labels differed between the two algorithms
6) Visualization of the venues is done on a map using Folium and ColorMap

The Gaussian Mixture was picked as the ML algorithm and based on its labels, a final itinerary is generated that lists all the venues in order of the day of vacation (label) and their ratings – so in case a person is unable to actually see all those venues, they would have their priorities set straight based on the ratings of the venues.

3.4. Evaluation

This section discussed the methodology of evaluation; the actual results are discussed in the section of 'Results'.

The evaluation of the algorithms was based on value_counts as well as a sanity check on whether it's feasible to follow a certain itinerary or not. Since this is unsupervised learning, it is difficult to judge the recommendations of the system unlike in the case of supervised learning (such as regression of classification).

4. Results

This section showcases the results of the program (their details and coding can be seen in the notebook).

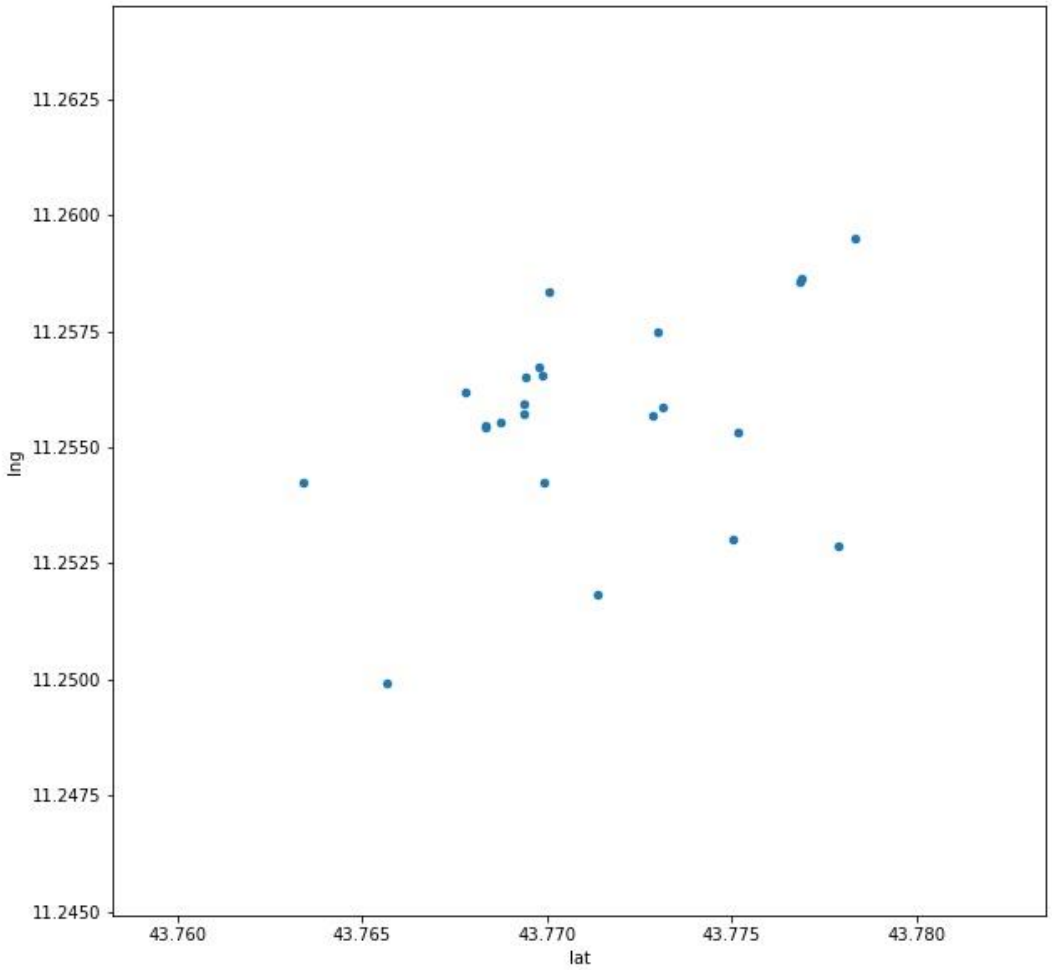
The order of the results is in sequence with the methodology highlighted above. The results below are based on the inputs of:

City: Florence, Italy
Categories: Museums and Landmarks
Days: 4
Rating Threshold: 8.0

1) Number of venues in the raw dataset from Foursquare: **31**
2) Number of venues in the dataset that had ratings: **29**
3) Number of venues that had rating greater than threshold: **23**
4) Visualization of venues (without labels):

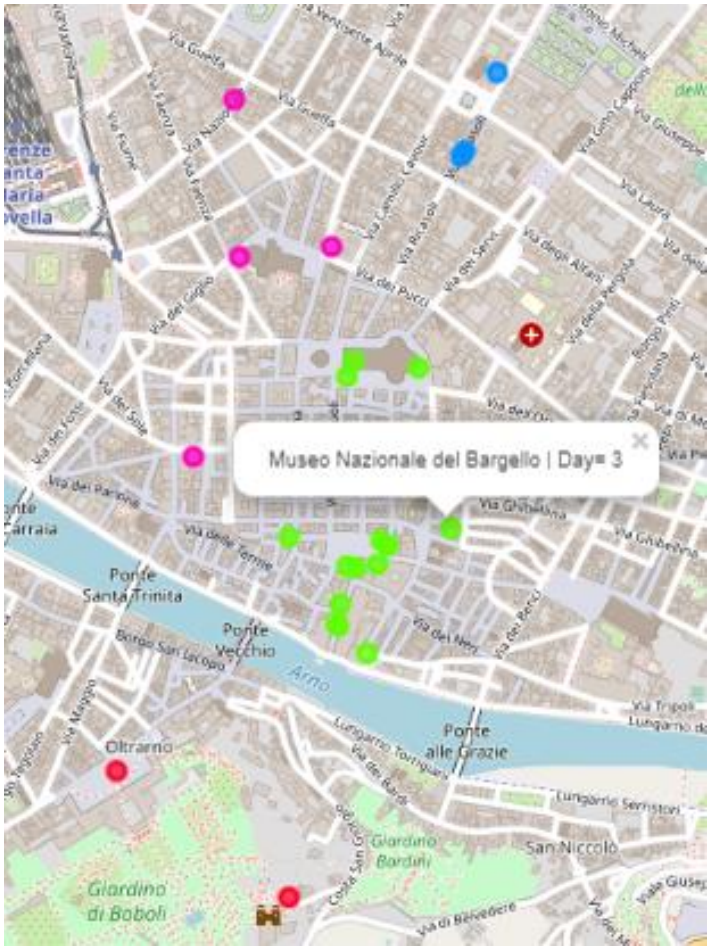
The label shows 'name | category'

5) Following scatter plot shows the coordinates of the venues:



The scatter plot shows how the venues are spread and even an overview of it gives an idea of the approximate clusters of venues.

6) Venues visualization based on k-means clustering:



The plot shows 4 clusters, the blue and red are only a couple of venues as they are far away from the rest of the venues. The majority of the venues are in the middle and they have been split into 2 clusters.

Cluster Color	Day of Vacation	Number of Venues
Pink Clusters	Day 1	4
Red Clusters	Day 2	2
Green Clusters	Day 3	14
Blue Clusters	Day 4	3

We can see that there is a very large skewness in number of venues for the 3rd day which makes this itinerary practically unfeasible

7) Venues visualization using Guassian Mixture

Cluster Color	Day of Vacation	Number of Venues
Pink Clusters	Day 1	3
Red Clusters	Day 2	11
Green Clusters	Day 3	2
Blue Clusters	Day 4	7

The clustering from Gaussian Mixture shows a more reasonably planned itinerary. Two heavy days and two light days (still a challenge but better than K-means)

8) Final Itinerary:

	id	name	categories	lat	lng	ratings	labels
0	4bd432ff41b9ef3b1a9001e6	David di Michelangelo	Monument / Landmark	43.776835	11.258575	9.3	0
1	4bc95f61cc8cd13acb9fbbcf	Museo di San Marco	Museum	43.778341	11.259498	8.9	0
2	4bcd9011fb84c9b64524223e	Galleria dell'Accademia	Art Museum	43.776907	11.258654	8.8	0
3	55c9d3b7498e89eef0daee5b	Birth of Venus - Botticelli	Art Gallery	43.768319	11.255487	9.5	1
4	5778e577498ea36b51b46890	Leonardo - Galleria Degli Uffizi	History Museum	43.768344	11.255441	9.4	1
5	51191cdfb0ed67c8fff5610b	Galleria degli Uffizi	Art Museum	43.768721	11.255554	9.1	1
6	4bd02aaa046076b0d6716f71	Forte di Belvedere	Scenic Lookout	43.763388	11.254258	8.9	1
7	53124421498e0a4ae3b41807	Torre del Palazzo Vecchio	Monument / Landmark	43.769361	11.255942	8.8	1
8	4bc6f24092b376b0efc94e3a	Loggia dei Lanzi	Sculpture Garden	43.769382	11.255714	8.7	1
9	4bd01a269854d13a1620f74d	Museo Galileo - Istituto e Museo di Storia della Scienza	Science Museum	43.767799	11.256202	8.5	1
10	4e8026c577c8c61e00ca31c5	Gucci Museo	Museum	43.769762	11.256747	8.3	1
11	5a759d5f8e886a73e9d0f995	Gucci Garden	Museum	43.769851	11.256550	8.2	1
12	53524fb4498eb5c71458096a	Museo di Palazzo Vecchio (Museo Civico)	Art Museum	43.769422	11.256502	8.1	1
13	4d03d7c226adb1f701eece70	Il Porcellino	Fountain	43.769906	11.254237	8.0	1
14	4c64255a79d1e21efe4bda15	Palazzo Strozzi	Art Museum	43.771379	11.251829	9.2	2
15	4bc8c9d7af07a593d4aa812d	Palazzo Pitti	Art Museum	43.765658	11.249919	8.9	2
16	4bd00cdb046076b00a576f71	Cattedrale di Santa Maria del Fiore	Church	43.773108	11.255879	9.5	3
17	4bc97c2dfb84c9b65a301b3e	Palazzo Medici-Riccardi	Museum	43.775182	11.255335	9.0	3
18	4b49cd73f964a520d87326e3	Campanile di Giotto	Monument / Landmark	43.772837	11.255692	9.0	3
19	4bd00aa8b221c9b6c80cd3d0	Museo dell'Opera del Duomo	Museum	43.772988	11.257492	8.9	3
20	4bf911c6b182c9b62261785a	Cappelle Medicee	Museum	43.775013	11.253010	8.9	3
21	4b49d545f964a520507426e3	Museo Nazionale del Bargello	Art Museum	43.770058	11.258340	8.9	3
22	4d682054709bb60c7b63b214	Le Fonticine	Italian Restaurant	43.777860	11.252890	8.6	3

The itinerary shows the venues, their ratings and labels (days). The venues are sorted based on labels (days) and their ratings.

5. Discussion

The results of the program are fairly intuitive – clustering of venues that are nearby is a logical approach that any person manually planning would do. However, the speed with which this program can do that is way faster than a manual process – *I have done this exercise manually for my Italy trip and I can assure the reader of this report I had not found all these venues by searching on Google. Looking at it now, my original itinerary needs revision.*

It is also highly repeatable. By re-running the program, the itinerary can be generated for multiple cities. Furthermore, the categories can be expanded – someone may be interested in nightclubs and bars whereas someone could be interested in adventures. One could also run the program to set up their itineraries for their day and night activities (based on different categories).

The report, however, does recognize there are shortcomings of the current program:

1) In reality a major constraint in trip planning is budgeting. All of these venues will have certain costs associated with them. The program could incorporate costing information (either through online sources or by manually entering them in a csv file), and find the best combinations of venues that don't break the budget.
2) Every venue would have associated working hours and different times that need to spent there for the perfect touristic experience. The recommended time spent could be scoured from Foursquare by going into venue details and the tips, and applying some Natural Language Processing or could simply be input manually in a csv file. Constraints on the dataset could be applied to find the best combination of venues
3) K-means and Gaussian Matrix both require an input of number of clusters. In this program, it was set as the number of days pre-decided by the user. A good thing would be for the

program to recommend the number of days to spend based on the venue-timings constraints and the user-budget constraints.

4) K-means and Gaussian Matrix do not recognize outliers. In reality, a landmark may be just too far away to be in a cluster yet the algorithm will still assign that to a cluster. Alternative machine learning algorithms like DBSCAN can be tested though that algorithm could make too many clusters (beyond the time limit of the vacation).

The current program provides a good basis for future work:

1) An application that takes input of user budget, preferences and links that with location data, transportation data and venue working hours' data to provide a holistic experience for any traveller
2) Incorporation of multiple travellers and their individual preferences to provide an itinerary that provides optimal 'fun' for all the travellers.

6. Conclusion

Foursquare API, machine learning algorithms and Python provide a powerful trifecta of tools that can be effectively used to solve problems – be it a business problem or a personal need. This program showcases that a problem/need as ubiquitous as travel planning can be solved (or at least attempted to be solved) by these techniques.

Further work and incorporation of advanced techniques/learnings can improve the usability of this program.

Signing off from the Capstone Project.