Final report for the peer-graded assignment of capstone project

Identifying the best places to open a French restaurant in Nagoya

by Jonhathan Pinon

15/07/2019

Contents

List of Figures

1 Introduction

1.1 Background

Nagoya is one of the largest cities in Japan. It has a population of over 2 million and is situated in the middle-eastern part of Japan. In Japan, French cuisine is relatively popular. According to Foursquare data, there are about a hundred of French restaurants in Nagoya. Supposing that a chain of restaurants specialized in French cooking want to open new French restaurants in Nagoya, where would be the best places?

1.2 Problem

The best place is the place that would attract the most customers. In this report, we define popularity as the number of customers. Ideally, we would need to identify, and ideally quantify, which factors are beneficial for attracting customers, and which are not. For example, we can expect the proximity of subway station to substantially increase the popularity. On the other hand, we can expect that similar restaurants would decrease the popularity. However, the lack of data make quantifying the factors impossible. We will still attempt to estimate the best places using other means.

1.3 Interest

Such a map would obviously be very interesting for any group who wishes to open a restaurant. A restaurant popularity depends on its quality, but the location is still a major factor. To maximize potential income, choosing the place that would potentially attract the most customers is very important.

2 Data description

2.1 Data sources

Foursquare location data is the only data source. Foursquare provides the position of most restaurants, train stations, subway stations, and other venues in Nagoya. The location of all venues in Nagoya listed by Foursquare, along with their category, will be our main data during this study. We could further increase the amount of data using other API such as Google Maps's API, but we judged it would create more inconsistently than benefits due to the different nature of the data (for example, the categories are not the same). Henceforth, we limited our source to Foursquare.

In addition, Foursquare provides stats about each venue such as the number of visitors currently at the specified venue. By collecting the number of users at a specific venue over the course of several weeks, we could get an estimate of its popularity. Unfortunately, in the scope of this course, we did not have enough time to collect enough data. Foursquare allowed developers to get the total number of visits of any venue by getting the details of that venue, thanks to the response field *checkinsCount*, but since April 2018, this no longer possible (reference 1).

2.2 Dataframes

Almost all data are stocked in dataframes, objects of the library pandas. Some data were stocked temporary in lists and Series for the sake of coding. All relevant data from Foursquare were stocked into just two dataframes. A dataframe containing all French restaurants in Nagoya (Figure 1), and a dataframe containing all venues in Nagoya (Figure 2).

2.3 Data cleaning

In the case of the dataframe containing the French restaurants, *french_venues*, directly after the construction of the dataframe from Foursquare data, some venues are not classified as French restaurants. This is because in Foursquare, venues can have several categories. When searching for all venues belonging to a certain category, Foursquare will pick all venues that contains that specific category among all their categories. Then, in our script, for each venue, we select its first category as its category which will be later stocked

```
[33]: french_venues
```

[33]:

	Venue ID	Venue	Venue Latitude	Venue Longitude	Venue Category
0	4d8b45a94757721e05f5c769	La Pêche	35.157840	136.906874	French Restaurant
1	58ca83f5ce593d315f7efa2a	THE GATEHOUSE (ゲートハウス 名古屋)	35.172402	136.882937	French Restaurant
2	4d2ec79eb97cb1f7fa359248	ビストロ横丁	35.168726	136.889152	French Restaurant
3	4c0b876f7e3fc928b267f582	ガーデンレストラン 徳川園	35.184579	136.932372	French Restaurant
4	4ec7752c93ad41338ccd0ac5	ビストロ ダイア	35.169817	136.924363	French Restaurant
5	4ba98533f964a5202f2c3ae3	Innover (イノーヴェ)	35.175126	136.922906	French Restaurant
6	52e783cf498ec421fe53cfb6	WINE & FOOD ワイン渡辺。	35.169359	136.912575	French Restaurant
7	59b1e5d86a8d860b72cf886b	BOUL'ANGE	35.171385	136.882461	French Restaurant
8	57e00b74498e42f046425207	La Bobine Galette Cafe (ラ ボビン ガレットカフェ)	35.169579	136.886761	French Restaurant
9	4c99e80bb8e9224ba0d8483d	BREIZH Cafe Creperie 名古屋タワーズプラザ店	35.171269	136.884098	French Restaurant
10	4c73a70e6b91b7138fc4fb20	Contemporary Dining Crown & Teppanyaki (クラウン)	35.185337	136.895959	French Restaurant
11	4c3320e13896e21e3f76e990	Brasserie Effort (ブラッセリー エフォール)	35.175810	136.897726	French Restaurant
12	4dd12ab4d22deadedd92f08b	Absinthe (アブサン)	35.176586	136.908785	French Restaurant
13	4f1a3031e4b064e65ab807d6	Neo Bistro Hondo Mondo	35.175803	136.908899	French Restaurant
14	4dafb8176a23d0da7ea63264	Matsuura (四間道レストラン マツウラ)	35.174537	136.892549	French Restaurant
15	4fa27fcae4b0b4fd555cff95	日仏食堂 en	35.175246	136.891003	French Restaurant
16	4ce34907ef2db60cd9e7bf5b	西洋食房 飯島屋 名駅店	35.173574	136.893035	French Restaurant
17	4f6d9988e4b068929fa0fe37	Nature Vert (ナチュール ヴェール)	35.175056	136.909899	French Restaurant
18	4c91a2821adc3704900232d1	レストラン ツキダテ	35.171478	136.901134	French Restaurant
19	4e40abe7483b72d779d92213	Dubonnet (旧春田邸 デュボネ)	35.181169	136.915869	French Restaurant
20	4ceb61efd99f721e05a8c073	タワーレストラン NAGOYA	35.172072	136.908360	French Restaurant
21	4ba1a25cf964a52021c537e3	Brasserie GLOUTON	35.170235	136.898439	French Restaurant
22	4ddf0b3bd22d728b20bf7a40	La Grande Table de KITAMURA (ラ・グランターブル・ドゥ・キタムラ)	35.180693	136.917371	French Restaurant

Figure 1 – *First part of the dataframe listing all French restaurants in Nagoya*

in the dataframe. As that first category may not necessarily be 'French restaurant', we manually reassign the category 'French restaurant' to all venues in *french_venues*.

In the case of the dataframe containing all venues around French restaurants in Nagoya, due to how we collected the data, there are thousands of duplicates. We remove them by dropping the duplicate rows from the dataframe. The construction of that dataframe, *nagoya_venues*, is explained in details in the following section.

3 Methodology

Principe

Our final goal is to get a list of the best places to open a French restaurant in Nagoya. As mentioned above, we cannot measure the popularity of a restaurant using Foursquare data. Consequently, it is nearly impossible to find out and measure what factors, related to its geographical position, make a restaurant popular.

Let us think from another perspective. Let us assume the owners of the already existing French restaurants thought thoroughly where they should open a French restaurant. Let us assume that the previous French restaurants were opened at *a good place*.

```
[35]: nagoya_venues
```

[35]:

	Part	Venue ID	Venue	Venue Latitude	Venue Longitude	Venue Category
0	167	5a28b892838e5955d0d3dee8	Yomoda Soba (よもだそば)	35.169870	136.883432	Soba Restaurant
1	197	5a28b892838e5955d0d3dee8	Yomoda Soba (よもだそば)	35.169870	136.883432	Soba Restaurant
2	167	4b64f138f964a5204edb2ae3	Soup Stock Tokyo	35.169965	136.883473	Soup Place
3	197	4dedd28718385379de4d6b65	Estmare (エストマーレ)	35.170098	136.883549	Hotel Bar
4	167	4dedd28718385379de4d6b65	Estmare (エストマーレ)	35.170098	136.883549	Hotel Bar
5	197	4b761274f964a520c23a2ee3	Yabaton (矢場とん)	35.169801	136.884170	Tonkatsu Restaurant
6	167	4b761274f964a520c23a2ee3	Yabaton (矢場とん)	35.169801	136.884170	Tonkatsu Restaurant
7	167	4c3be7f15418897cb4330803	名駅前宝くじチャンスセンター	35.169932	136.884134	Lottery Retailer
8	167	53367b0011d2a2ad52660a3c	Starbucks (Starbucks Coffee)	35.169875	136.884277	Coffee Shop
9	197	53367b0011d2a2ad52660a3c	Starbucks (Starbucks Coffee)	35.169875	136.884277	Coffee Shop
10	167	4def2a63c65bf3f03e8fdb3d	Sumiyoshi (住よし)	35.170139	136.882778	Udon Restaurant
11	197	4b59625ff964a520af8628e3	Tokyu Hands (東急ハンズ 名古屋店)	35.170388	136.883361	Stationery Store
12	167	4b59625ff964a520af8628e3	Tokyu Hands (東急ハンズ 名古屋店)	35.170388	136.883361	Stationery Store
13	167	4b679514f964a5202a562be3	Sumiyoshi (住よし)	35.170104	136.882688	Udon Restaurant
14	167	4b66f887f964a520e8322be3	TOWER RECORDS	35.169265	136.884450	Record Shop
15	197	4b66f887f964a520e8322be3	TOWER RECORDS	35.169265	136.884450	Record Shop
16	167	4b630a99f964a5205b5f2ae3	Marriott Nagoya Associa (名古屋マリオットアソシアホテル)	35.170446	136.882980	Hotel
17	197	4c452d0edcd61b8d8b707c56	Salon de Moncher ミッドランドスクエア店	35.169937	136.884590	Café
18	167	4c452d0edcd61b8d8b707c56	Salon de Moncher ミッドランドスクエア店	35.169937	136.884590	Café
19	167	4cc556b082388cfa29a57435	DEAN & DELUCA Express Cafe	35.169841	136.884638	Café
20	197	4cc556b082388cfa29a57435	DEAN & DELUCA Express Cafe	35.169841	136.884638	Café
21	167	4d41ecd6f0dba1cd022b2e49	Sumiyoshi (住よし)	35.169981	136.882354	Udon Restaurant
22	197	4db69b7a93a017099e02ffe6	MUJI (無印良品)	35.168687	136.884053	Clothing Store
23	167	4db69b7a93a017099e02ffe6	MUJI (無印良品)	35.168687	136.884053	Clothing Store
24	167	5530e5c6498ee29a12373085	Café & Meal MUJI	35.168623	136.883890	Café

Figure 2 – *First part of the dataframe listing all venues around French restaurants in Nagoya*

Then by analyzing the geographical details of each French restaurant, we can estimate in what kind of places French restaurants tend to be opened. In other words, which are the best places to open a French restaurant. By *geographical details*, we mean the venues surrounding the French restaurants. For example, if we discover that French restaurants tend to be close to subway stations and far away from other restaurants, then we find the best places in Nagoya by looking at which points the distances from subway stations are minimized while the distances from other restaurants are maximized.

This is the general idea of our methodology. Due to the limiting computing power, we cannot look at each possible point (i.e: set of longitude and latitude) in Nagoya. We decided to limit our study to a grid of 30*30 points in an area which contains all French restaurants in Nagoya. There are thus 900 points in the grid. One part of the grid is shown on Figure 3. The red circle is the area studied, which was defined so as to contain all French restaurants in Nagoya and their surrounding venues. The meaning of the blue circles is explained in the next subsection.

For each point in the grid, we will calculate a *recommendation index*. The higher it is, the more recommended it is to open a restaurant at that place. This *recommendation index* is the weighted sum of a *similarity index* and a *penalty index*. For each point, the *similarity index* represents how the venues around that point are similar to the venues that

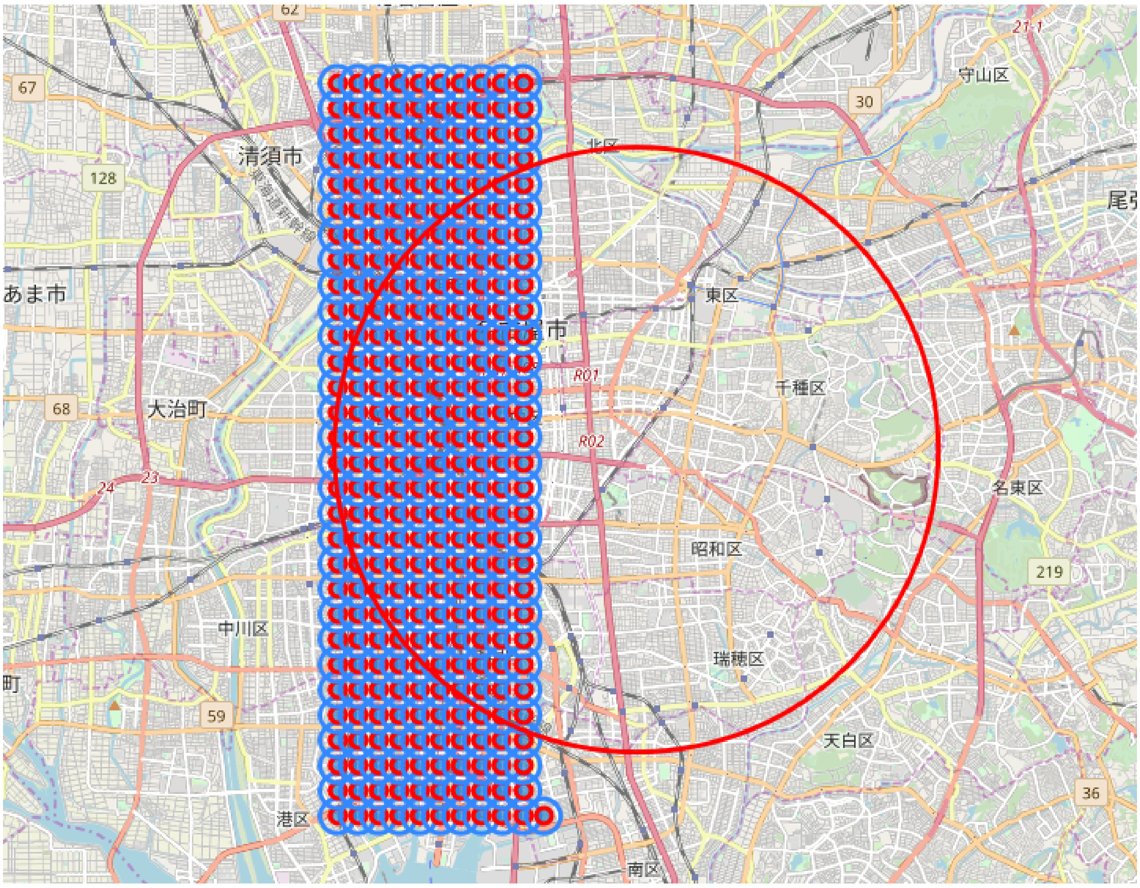

Figure 3 – *Division of Nagoya into a 30*30 grid*

tend to be found around French restaurants. The *penalty index* represents the proximity of French restaurants. We assume that it is better to open a French restaurant away from other French restaurants due to competition.

$$\mathbf{recommandation}_i = (1 - c) * \mathbf{similarity}_i - c * \mathbf{penalty}_i \tag{1}$$

c is a coefficient used to weight the importance of the similarity versus the penalty. It is calculated later.

3.1 Preparation of the data

The collection of all French restaurants in Nagoya is straightforward. We run a single request to Foursquare API to get the list of all venues that are classified as *French restaurant*. The latitude and the longitude of the center of the circle where the venues are explored are obtained through the python libray *geopy*.

The collection of all venues, which are NOT French restaurants, is more complex. One of the limits of Foursquare API is that a single request, to explore the venues around a place, only gives back at most 100 venues. However, there are certainly more than 100 venues in Nagoya. To get as many venues as possible, we split Nagoya, or more precisely the area around the French restaurants, into 900 parts, in a 30*30 grid. The grid is the same grid referenced in Figure 3. Then for each area, we make a call and get the list of all venues in that area. The blue circles in the Figure represent the size of each area. Let us notice that all area inside the red circle, which contains all French restaurants in Nagoya, is covered by the blue circles. Thanks to this division of Nagoya, we can get most of the venues around French restaurants by making 900 calls to Foursquare API.

Of course, collecting the venues this way will generate a lot of duplicates. Indeed, each point inside the red circle is covered by several blue circles. To clean our data, we drop the duplicates in *nagoya_ venues*.

The calculation of the *similarity index* will involve millions of mathematical operations. The distance between each French restaurant and each venue in Nagoya will be calculated. We can calculate the distance between two points on Earth using the Haversine formula, but this formula involves trigonometric functions which would make the computational time far too long (more than one day). To accelerate as much as possible the calculation of the *similarity index*, we will first calculate the position of all venues (including French restaurants) in a local Cartesian system. The center of this coordinate system will the center of the red circle, shown on Figure 3. To transform the latitude and longitude into Cartesian coordinates, we use the formula described in reference 2. This formula is derived from the Equirectangular approximation. The formula is as follow:

$$x = R * (lgt - lgt_0) * cos(lat_0) \tag{2}$$

$$y = R * (lat - lat_0) \tag{3}$$

where:

- R is the radius of Earth (6373 km)
- lgt is the longitude in radians of the given point
- lat is the latitude in radians of the given point
- lgt_0 is the longitude in radians of the origin
- lat_0 is the latitude in radians of the origin

The Equirectangular approximation is a good enough approximation considering the distances involved. We verified this hypothesis by comparing the distances calculated using this approximation to the distances calculated using the Haversine formula.

All venues in *french_ venues* and *nagoya_ venues* have now their coordinates in the local Cartesian system.

3.2 Calculation of the recommendation index

The *recommendation index* is the weighted sum of the *similarity index* and the *penalty index*. The *similarity index* is calculated as follows:

Step 1 : 100 closest venues for each French restaurant For each French restaurant in *french_venues*, get the 100 closest venues (which are not French restaurants). The venues are obtained from *nagoya_venues*.The distances are calculated using the Cartesian coordinates. The 100 closest venues along with the distances to the French restaurant are saved into a dataframe. The list of dataframes is called *french_100venues_list*. We now have a list of dataframes which show what venues are found near French restaurants. More importantly, we have the distance of each French restaurant to their closest venues.

Step 2 : construction of the super french restaurant model Concatenation of all dataframes in *french_100venues_list* into a super dataframe, called *model_fr*. What matter are the distance and the venue category of each row. All venues inside that dataframe will be compared to the 100 closest venues of each point in the grid. In a way, the super dataframe is like the average venues found near a French restaurant.

Step 3 : 100 closest venues for each point in the grid Like it was done with the French restaurants, we create a list of dataframe, called *nagoya_100venues_grid*. Each dataframe in the list contains the 100 closest venues to a specific point of the grid. After the dataframes are created, we calculate the minimum distance of the maximum distance found in each dataframe. In other words, we get the minimum distance among the distance between the 100 farthest venue and the respective point. This value, called *max_distance_normalization* is an important part of the model. The meaning of this value will be explained shortly.

Step 4 : extrapolate each dataframe of the grid to compare it to *model_fr* Each point of the grid is associated with 100 venues, yet *model_fr* have $100 * number\ of\ French\ restaurants$ venues. To compare the two dataframes, we extrapolate the dataframe in *nagoya_100venues_grid*. For each dataframe, we duplicate X times each venue inside that dataframe. X is the number of restaurants.

Step 5 : extrapolate each dataframe of the grid to compare it to *model_fr*

4 Results

5 Discussion

6 Conclusion

References

1. https://developer.foursquare.com/docs/announcements#start-up-tier-launch

2. http://www.movable-type.co.uk/scripts/latlong.html