Final report for the peer-graded assignment of capstone project

Identifying the best places to open a French restaurant in Nagoya

by Jonhathan Pinon

10/09/2019

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 millions 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 stations 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

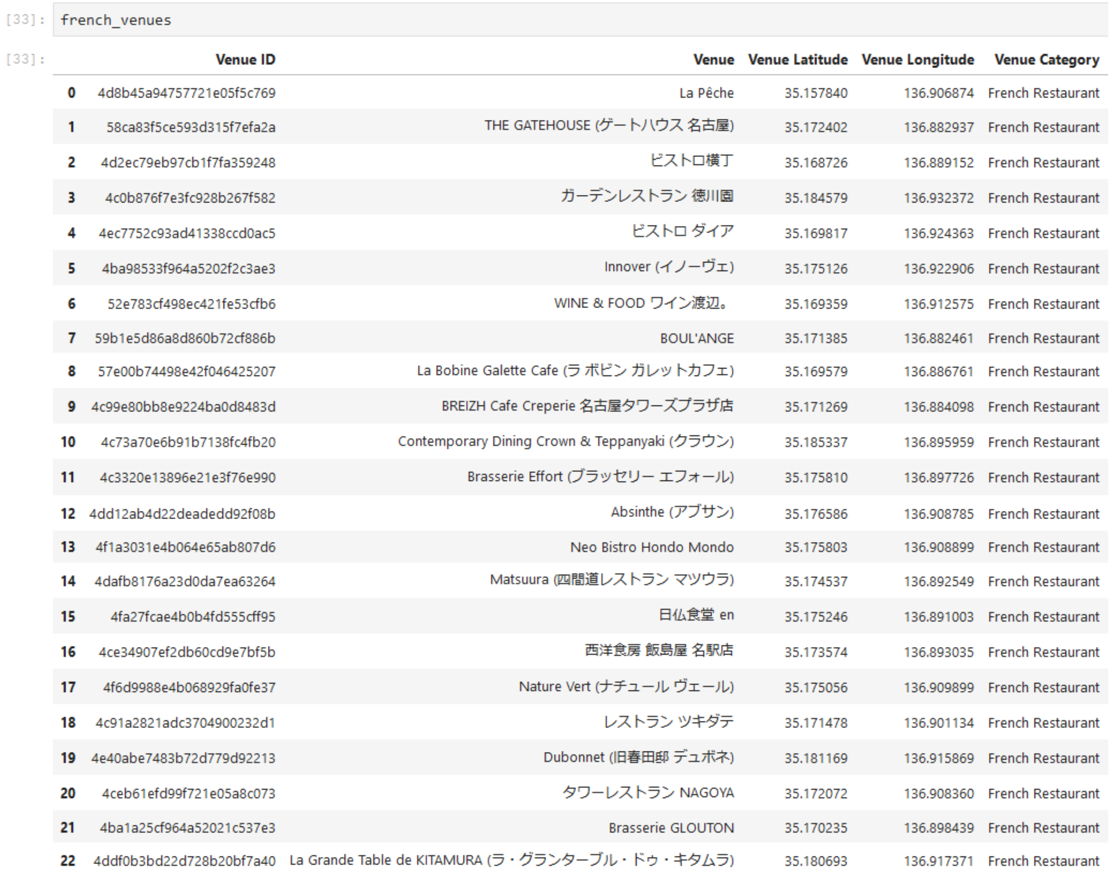
Foursquare location data is the only data source. Foursquare provides the position of most restaurants, train stations, subway stations, and other venues in Nagoya. The location of all venues in Nagoya listed by Foursquare, along with their category, will be our main data during this study. We could 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 python library pandas. Some data were stocked temporary in lists and series for the sake of coding. All relevant data from Foursquare were stocked into dataframes. The dataframe that were used to stock the data are:

- french_venues that contains all French restaurants in Nagoya (Figure 1).
- nagoya_venues_food that contains all restaurants in Nagoya (except French restaurants).
- nagoya_venues_transport that contains all transport venues in Nagoya.
- nagoya_venues_longstay that contains all workplace and residence related venues in Nagoya.
- nagoya_venues_shortstay that contains all shops, recreation, hobbies, park, art related venues in Nagoya (Figure 2).

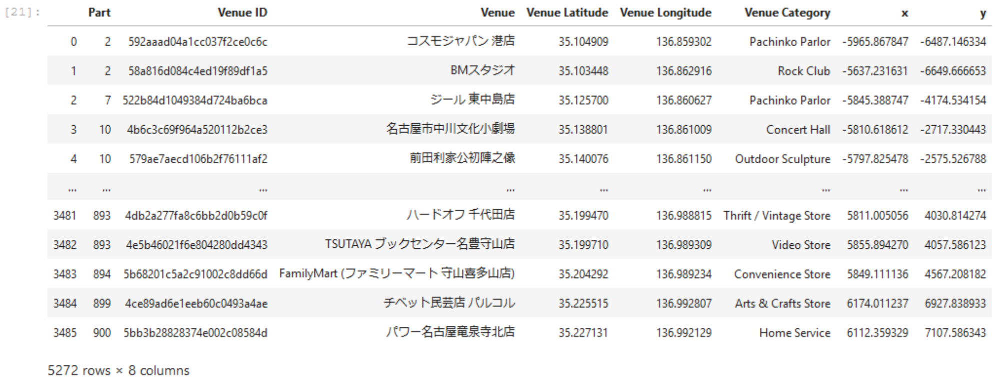


```
[33]: french_venues
```

[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*



[21]:

	Part	Venue ID	Venue	Venue Latitude	Venue Longitude	Venue Category	x	y
0	2	592aaad04a1cc037f2ce0c6c	コスモジャパン 港店	35.104909	136.859302	Pachinko Parlor	-5965.867847	-6487.146334
1	2	58a816d084c4ed19f89df1a5	BMスタジオ	35.103448	136.862916	Rock Club	-5637.231631	-6649.666653
2	7	522b84d1049384d724ba6bca	ジール 東中島店	35.125700	136.860627	Pachinko Parlor	-5845.388747	-4174.534154
3	10	4b6c3c69f964a520112b2ce3	名古屋市中川文化小劇場	35.138801	136.861009	Concert Hall	-5810.618612	-2717.330443
4	10	579ae7aecd106b2f76111af2	前田利家公初陣之像	35.140076	136.861150	Outdoor Sculpture	-5797.825478	-2575.526788
...	...	...	...	...	...	...	...	...
3481	893	4db2a277fa8c6bb2d0b59c0f	ハードオフ 千代田店	35.199470	136.988815	Thrift / Vintage Store	5811.005056	4030.814274
3482	893	4e5b46021f6e804280dd4343	TSUTAYA ブックセンター名豊守山店	35.199710	136.989309	Video Store	5855.894270	4057.586123
3483	894	5b68201c5a2c91002c8dd66d	FamilyMart (ファミリーマート 守山喜多山店)	35.204292	136.989234	Convenience Store	5849.111136	4567.208182
3484	899	4ce89ad6e1eeb60c0493a4ae	チベット民芸店 パルコル	35.225515	136.992807	Arts & Crafts Store	6174.011237	6927.838933
3485	900	5bb3b28828374e002c08584d	パワー名古屋竜泉寺北店	35.227131	136.992129	Home Service	6112.359329	7107.586343

5272 rows × 8 columns

Figure 2 – *First part of the dataframe listing all 'short stay' venues in Nagoya*

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 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 dataframes containing the other kind of 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. We remove all French restaurants in those dataframes.

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 *good places*. Then by analyzing the geographical details of each French restaurant, we can estimate what kind of places French restaurants tend to be opened. In other words, what 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 limited 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. 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.

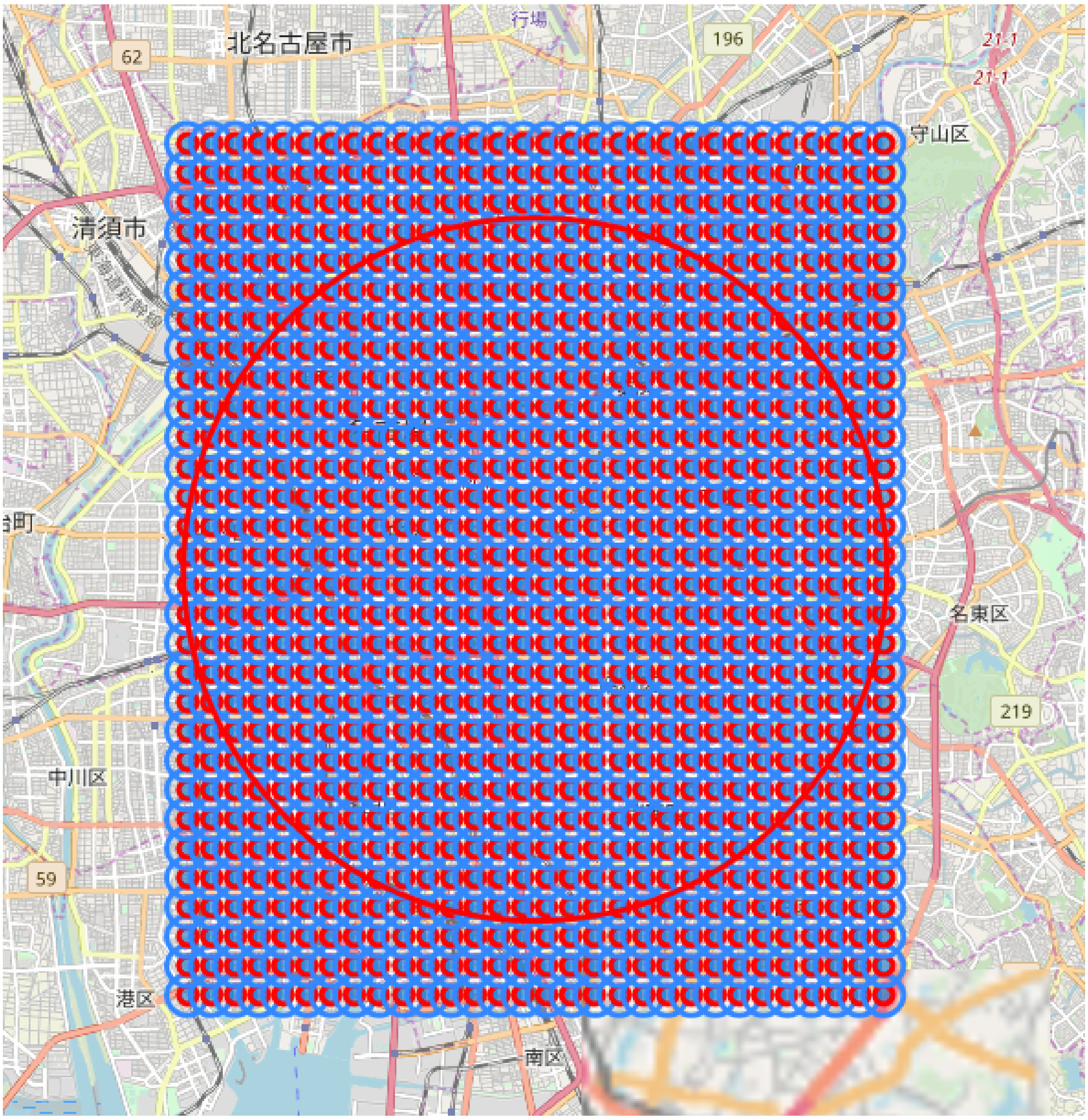


Figure 3 – *Division of Nagoya into a 30*30 grid*

For each point in the grid, we will calculate a *similarity distance*. The lower it is, the more this area is similar to the other areas with French restaurants. In other words, the more recommended it is to open a restaurant at that place. This *similarity distance* is the the weighted sum of several 'distances'. They measure the similarity of the area with the other areas with French restaurants, with regards to the numbers of restaurants, short stay venues, long stay venues and transport venues.

$$\begin{aligned}\textbf{similarity distance}_i = \Sigma_j \quad & a_1(pfood_{fr,j} - pfood_{a,i})^2 \\ &+ a_2(ptrans_{fr,j} - ptrans_{a,i})^2 + a_3(plgstay_{fr,j} - plgstay_{a,i})^2 \\ &+ a_4(pshstay_{fr,j} - pshstay_{a,i})^2 + a_5(nvenues_{fr,j} - nvenues_{a,i})^2 \quad (1)\end{aligned}$$

- **similarity distance**$_i$ is the similarity distance of the area i, belonging to the 30*30 grid (lower = more similar).
- a_1 is a coefficient that weights the distance relative to restaurant venues. It will be adjusted later.
- $pfood_{fr,j}$ is the proportion of restaurant venues compared to all other kind of venues, in a 150 meters around the French restaurant j.
- $pfood_{a,i}$ is the proportion of restaurant venues compared to all other kind of venues, in a 150 meters around the area i.
- a_2 is a coefficient that weights the distance relative to transport venues. It will be adjusted later.
- $ptrans_{fr,j}$ is the proportion of transport venues compared to all other kind of venues, in a 150 meters around the French restaurant j.
- $ptrans_{a,i}$ is the proportion of transport venues compared to all other kind of venues, in a 150 meters around the area i.
- a_3 is a coefficient that weights the distance relative to long stay venues. It will be adjusted later.
- $plgstay_{fr,j}$ is the proportion of long stay venues compared to all other kind of venues, in a 150 meters around the French restaurant j.
- $plgstay_{a,i}$ is the proportion of long stay venues compared to all other kind of venues, in a 150 meters around the area i.
- a_4 is a coefficient that weights the distance relative to short stay venues. It will be adjusted later.
- $shstay_{fr,j}$ is the proportion of short stay venues compared to all other kind of venues, in a 150 meters around the French restaurant j.
- $shstay_{a,i}$ is the proportion of short stay venues compared to all other kind of venues, in a 150 meters around the area i.

- a_5 is a coefficient that weights the distance relative to the number of all venues. It will be adjusted later.
- $nvenues_{fr,j}$ is the number of all venues in a 150 meters around the French restaurant j.
- $nvenues_{a,i}$ is the number of all venues in a 150 meters around the area i.

We choose 150 meters as the limit because we suppose that beyond 150 meters, the impact of other venues become negligible compared to the impact of the closer venues .

Intuitively, the impact of the nearby transports seems larger than the impact of short stay and long stay venues. The number of shops (short stay) should impact the popularity of French restaurant differently than the number of offices (long stay) nearby too. This is why we separated the 3 above types of venues, with 3 coefficients a_2, a_3 and a_4.

The presence of nearby restaurants may not be beneficial because of competition. Hence, we used a different coefficient, a_1, for food-related venues.

The proportion of a certain kind of venues versus the number of all venues may matter more than the absolute number of that kind of venues. For example, an area with 10 restaurants and 20 transport venues seems more attractive than an area with 20 restaurants and 10 transport venues. This is why we are comparing proportions rather than absolute numbers. This way, we are less biased against areas with a small amount of venues. To keep all information, we still compare the absolute number of venues, weighted by a_5.

We will use an optimization algorithm to find the best coefficients.

After we calculate the similarly distance for every area in the grid, we will rank the areas according to the distance. The areas with the lowest distance are the most recommended areas.

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 the surface 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 distance* will involve hundreds of thousands 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. 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 the other dataframes have now their coordinates in the local Cartesian system.

3.2 Calculation of the similarity distance and the F1 score

Let us describe step by step how we calculate the similarity distance and then the f1 score:

1. Make the dataframe containing all french restaurants in Nagoya
2. Make the dataframe containing all restaurants / transport venues / long stay venues / short venues in Nagoya. Take out all French restaurants.

3. Clean the dataframes : delete duplicates. Change the category of each venue to the first category listed (because one venue can have several categories).

4. Calculate the x and y local Cartesian coordinates to greatly accelerate the calculations later.

5. For each french restaurant in the French dataframe, calculate the total number of venues, the proportion of other restaurants, transport venues, short stay venues and long stay venues in a 150 meter radius.

6. For each area in the grid, calculate the total number of venues, and the aforementioned proportions in a 150 meter radius. Those data are stocked in the dataframe nagoya_grid_df. Delete rows that have NaN values (which happens if there is no restaurant / transport / other venues in the 150 meter radius).

7. Calculate for each area the similarity distance using the formula above.

8. For each area in the grid, check if there is a French restaurant. Get the number, *nb_ areas_ with_fr* of different areas with at least one French restaurant. This number is smaller than the total number of French restaurants because there can be more than one French restaurants per area.

9. Sort each area by their similarity distance. The *nb_ areas_ with_fr* first areas with the smallest distance will be marked as having a 'predicted' French restaurant.

10. Check if the areas that have predicted French restaurants also have real French restaurants (true positive). Do the same for true negative, false positive and false negative. Calculate the F1 score (calculated automatically using the function f1_score from the sklearn library).

11. The calculation of the similarity distance was done with one set of $a_k (1 < k <= 5)$. To try to get the best results, reiterate the steps from the similarly distance calculation with different a_k. The a_k will be chosen automatically by the optimization algorithm. Here, we decided to use the least square method and the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm. The goal is to get the highest F1 score.

4 Results

4.1 Preliminary results

Before applying the model, we explored the data with pandas. We plotted the histograms of the distributions of:

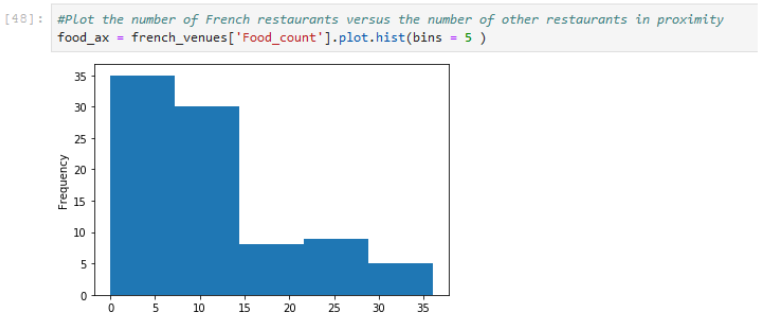
- Other restaurants in a 150 meter radius around French restaurant : Figure 4.
- Transport venues in a 150 meter radius around French restaurant : Figure 5.
- Long stay venues in a 150 meter radius around French restaurant : Figure 6.

- Short stay venues in a 150 meter radius around French restaurant : Figure 7.

For the four distributions, we see French restaurants tend to be in areas with few other venues. In fact, the more venues there are, the less likely there are French restaurants ... Or at least, this is what we could believe if we just look at the plots. The truth is more complicated because the number of low venue populated areas outnumber greatly the number of high venue populated areas. Hence, it is expected that we find more French restaurants in low venues populated areas, just because there are much more low populated areas.

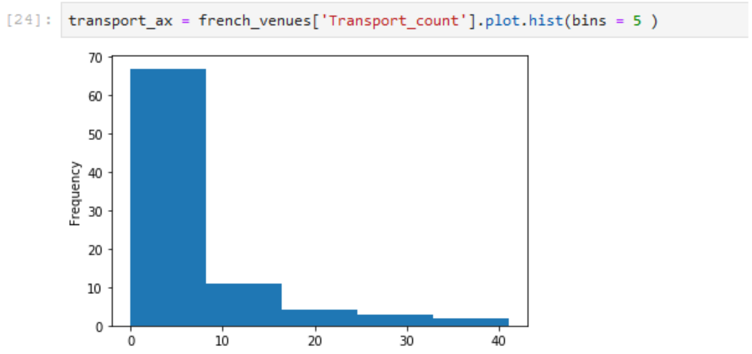
The distribution of restaurants is clearly different from the other distributions. It is not as left-sided. It shows that the presence of French restaurants is more strongly correlated with the presence of other restaurants. Competition does not seem to matter. We can explain this by the fact that an area with a lot of restaurants is very attractive for a French restaurant, like any restaurant. Otherwise, there would not be a lot of restaurants in that area in the first place.

Considering the distributions of transport, long stay and short stay related venues are similar, it may be more efficient to group them together in the similarly distance equation (i.e: same coefficient). But for the sake of completeness, we decided to keep them separated.



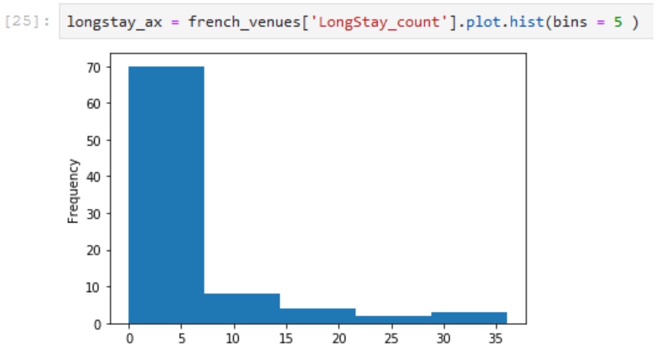
```
[48]: #Plot the number of French restaurants versus the number of other restaurants in proximity
      food_ax = french_venues['Food_count'].plot.hist(bins = 5 )
```

Figure 4 – *Distribution of the number of restaurants around French restaurants*



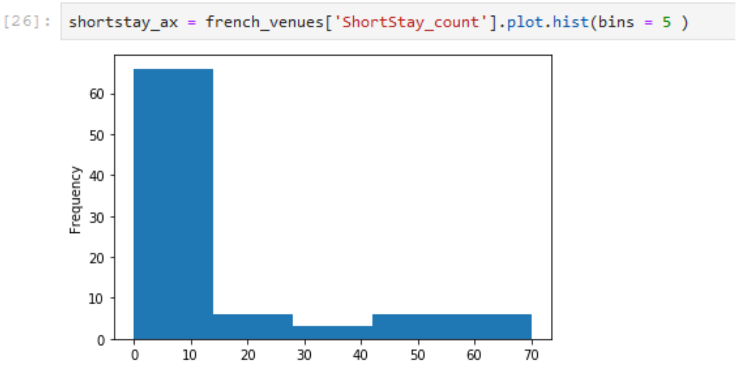
```
[24]: transport_ax = french_venues['Transport_count'].plot.hist(bins = 5 )
```

Figure 5 – *Distribution of the number of transport venues around French restaurants*



```
[25]: longstay_ax = french_venues['LongStay_count'].plot.hist(bins = 5 )
```

Figure 6 – *Distribution of the number of long stay venues around French restaurants*



```
[26]: shortstay_ax = french_venues['ShortStay_count'].plot.hist(bins = 5 )
```

Figure 7 – *Distribution of the number of short stay venues around French restaurants*

4.2 Model results

To test the accuracy of our model, we divided the dataframe containing the French restaurants into two parts. One part for training the model (finding the coefficients a_k giving the highest f1 score) and one part to test the model.

We used the least square algorithm and the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm from the scipy library to optimizes the model.

The initial vector was:

- $a_1 = 1$
- $a_2 = 1$
- $a_3 = 1$
- $a_4 = 1$
- $a_5 = 0.01$

We supposed the impact of food-related venues was the same as the impacts of the other kinds of venues. We assigned a coefficient 100 times lower for the impact of the absolute number of venues, because in the case of the proportions, the distance $((plgstay_{fr,j} - plgstay_{a,i})^2)$ should be between 0 and 1. Whereas, in the case of the absolute number of nearby venues, the distance $(nvenues_{fr,j} - nvenues_{a,i})^2$ should be between 0 and 10,000 (in average, between 100 and 1000).

Despite trying both algorithm, we failed to find a better vector than the one above. The two algorithms stopped (successfully) at the first iteration. We tried several different initial vectors, such as [1,1,1,1,0.0001], but no matter what, the algorithms stopped at the first iteration. The f1 score was slightly worse when we tried other initial vectors (0.375 instead of 0.406 for example). The function that gives the f1 score, which we tried to optimize, is non-linear and extremely complex. There are about 80*5 (French restaurants) + 900*5 (grid of Nagoya) independent variables. This is why the algorithm failed to find a better vector.

The f1 score on the training set was 0.40625.

The f1 score on the test set was 0.1212.

The f1 score on the complete (training + test) set was 0.3875. The worst score is due to the fact there can be more than one French restaurant per area.

The maps showing the true positives (in green), true negatives (in blue), false positives (in red) and false negatives (in yellow) in the three cases are shown in Figure 8 (training set), Figure 9 (test set) and Figure 10 (complete set).

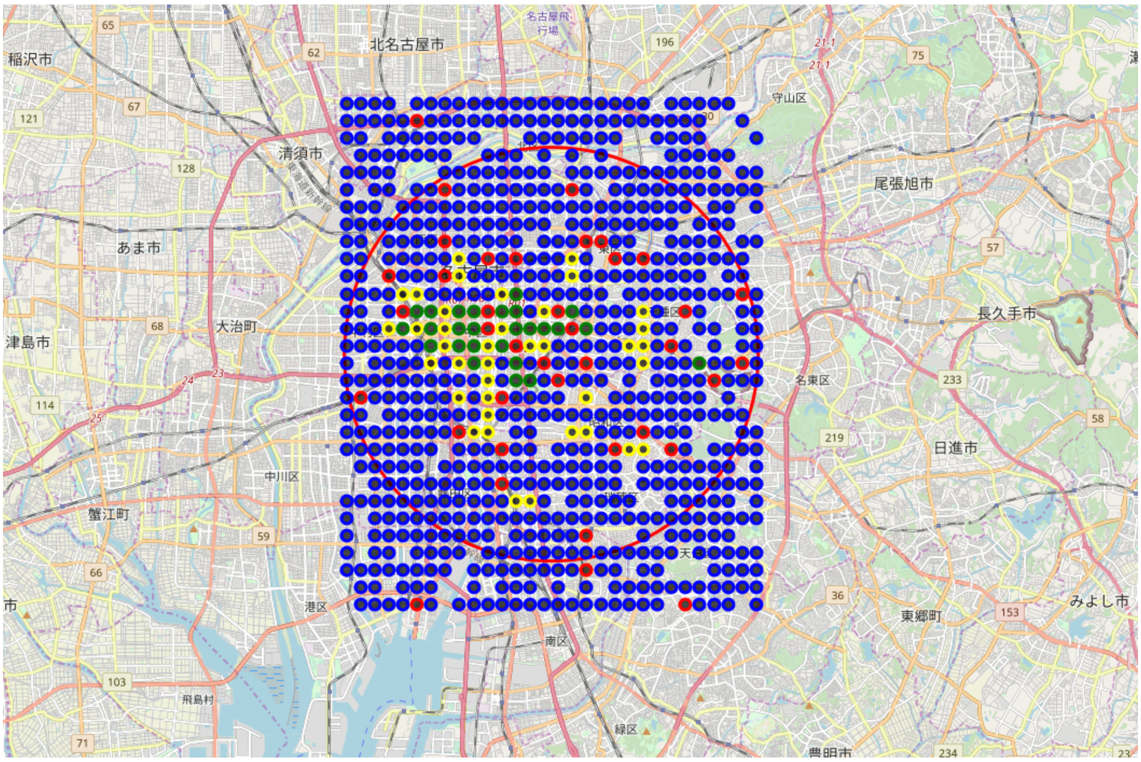


Figure 8 – *Grid of Nagoya showing true positives, true negatives, false positives and false negatives on the training set*

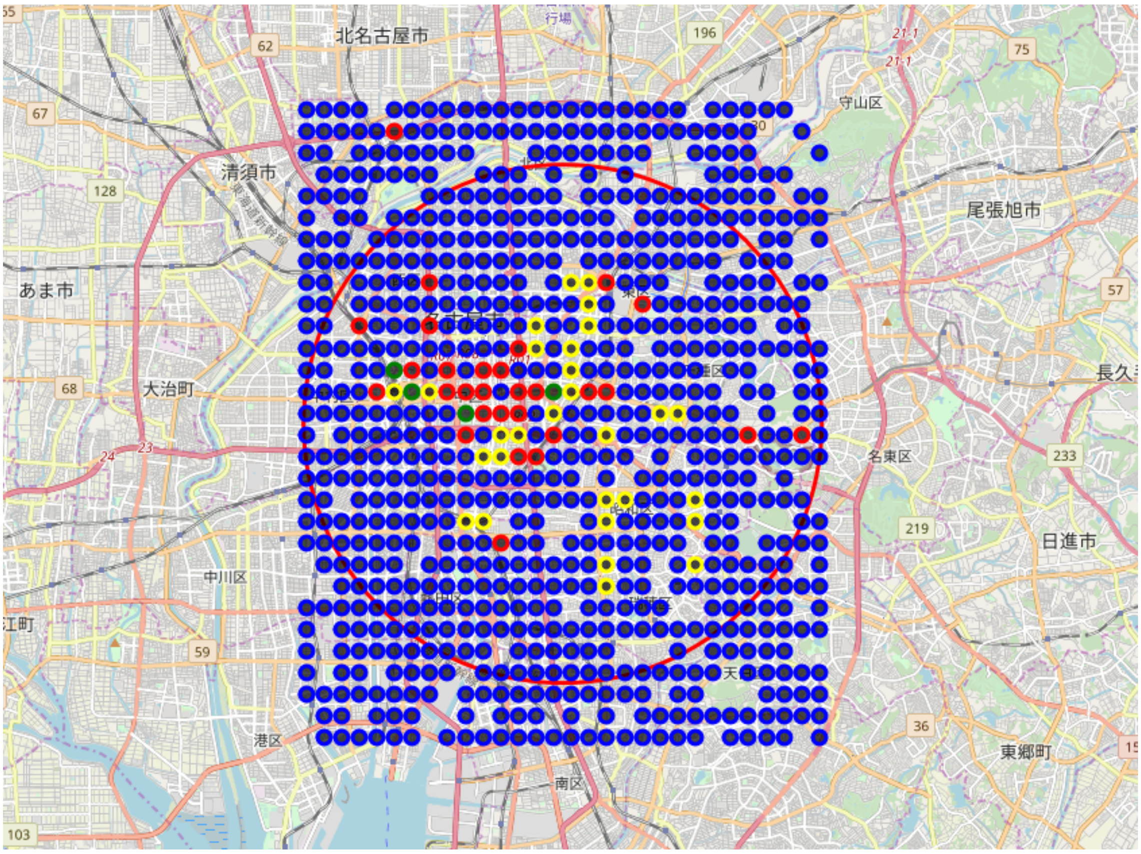


Figure 9 – *Grid of Nagoya showing true positives, true negatives, false positives and false negatives on the test set*

5 Discussion

5.1 Accuracy of the model

The f1 score is bad (0.40). It would be even worse if there was not a certain bias in the model. We calculated the similarity distance for each area. Then we took the *nb_ areas_ with_ fr* areas with the smallest distance. *nb_ areas_ with_ fr* is the number of areas with at least one (true) French restaurant. This means that no matter what, the model is guaranteed to predict as many areas with French restaurants as there are in reality:

$$true\ positive + false\ positive = real\ number\ of\ areas\ with\ French\ restaurant \tag{4}$$

$$true\ negative + false\ negative = real\ number\ of\ areas\ without\ French\ restaurant \tag{5}$$

Despite the low score, the model is better than random predictions. We calculated the average f1 score if the list of predicted areas with French restaurants was

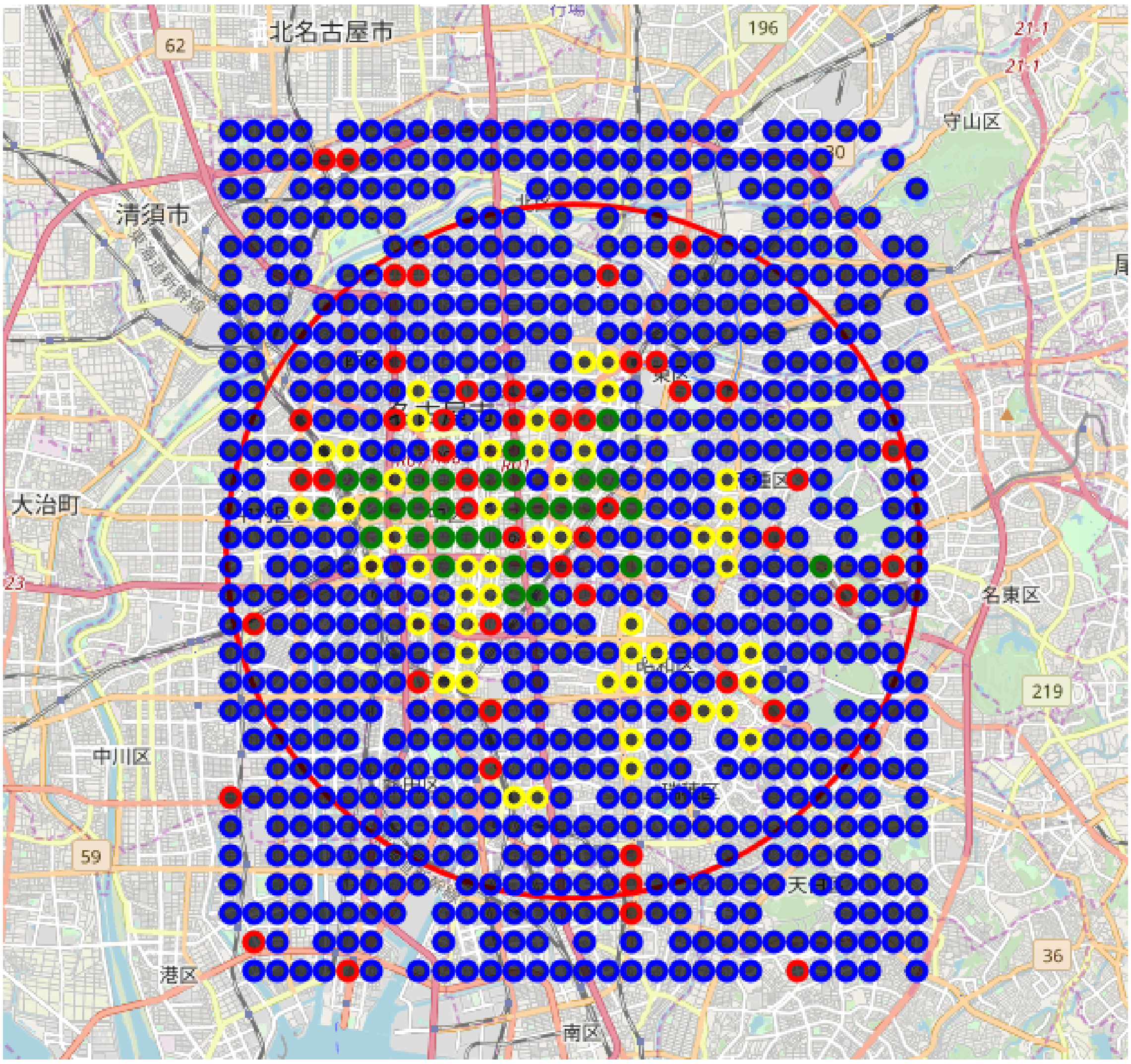


Figure 10 – *Grid of Nagoya showing true positives, true negatives, false positives and false negatives on the complete set*

generated totally randomly, instead of being the list of the areas with the smallest similarity distance. We ran 2000 simulations and averaged the f1 score.

The f1 score obtained was about 0.10. Although the f1 score of our model is bad, it is still 4 times higher than the random f1 score. The model is considerably better than random guesses.

Let us go back to the original question : where would be the best places to open a French restaurant? We need to look at the false positives (red, Figure 10) to answer this question. The false positives at the edge of the map seem to be outlier and thus should be ignored. The areas with several consecutive red points look promising, especially if there are no true or false negatives nearby. Although the model is bad, it is not random. It proposes areas that are similar to areas with French restaurants. If it proposes several areas without yet existing restaurants, it means there are potentially a market with few competition in those areas.

5.2 Limit of the study

The absence of data on the popularity of the venues was a major problem. If we could get the number of visits of every venues in Nagoya, then we would be able to check what factors impact the popularity of French restaurants. Does the number of nearby restaurants increase in average the number of visits? Does the presence of nearby French restaurants decrease in average the number of visits? Such questions could have been answered if we had access to those data.

Foursquare does not list ALL venues in Nagoya. In fact, based on my experience in Nagoya, I would stay about 20% at most are shown. Besides, Foursquare's API is inconsistent . Concretely, let us say we find all French restaurants inside a 1 kilometer circle. Let us say we get 80 results. Then, if we search all French and Italian restaurants inside that same area, by calling API one time, we may find just 60 French restaurants instead of 80!

6 Conclusion

The goal of the study is to find the best places to open a French restaurant in Nagoya. This question is very important for any chain or individual who wishes to open a French restaurant in Nagoya. To answer this problem, we used exclusively Foursquare API. Unfortunately, the lack of data regarding the number of visits for each venue forced us to devise indirect, and thus very complex, ways to predict the best areas. The underlining principle is that previous shops owners opened their restaurants in the best places. Hence, looking for areas similar to the current areas with French restaurants should show us good areas to open a new restaurant.

To find those areas, we first defined a 'similarity distance'. This distance represents how similar an area is compared to the average area with French restaurants with regards to the proportion of restaurants, transports, long stay venues, short stay venues and total number of nearby venues. The result showed the model was inaccurate. The f1 score on the complete set was just 0.4. Furthermore, the machine algorithms we tried failed to find a better initial vector due to the complexity of the model. Even so, we demonstrated it was better than random guesses, and thus could be used to estimate where to open a French restaurant.

Much more useful models could be built if we had access to the number of visits. Even with the data we collected, we believe better models could be built. For example, we chose to count all restaurants, shops and so on around French restaurants in a 150 meter radius. This limit, 150, should have considerable impact on the result. We also could find a better vector. More importantly, we could build a new model based on a new equation, hopefully simpler and more intuitive.

The results are unsatisfactory from a business point of view. But considering the scope of this project (one person, 30 hours, application of data science), I personally believe this study demonstrates the skills learned from the courses of IBM Data Science Professional Certificate.

References

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

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