

Country restaurants around the world

1- Business Approach.

Description of the problem / opportunity

1.1. OBJECTIVE:

The objective of the project is to create a graphical tool (based on Folium) that allows for any city to analyze:

- a- the restaurants of which nationality predominate in each area
- b- the distribution of restaurants of the desired country, grouped by area.

Initially, the objective also included analyzing the relationship between the immigrants of each country and the restaurants of each of those nationalities in the main cities of that country. However, I finally focused the objective on the previous point; so possibly at best that immigration analysis will narrow it down to a big city in Canada (Toronto for example).

1.2. AUDIENCE.

The groups interested in this project are among others.

- People interested in opening a national restaurant in a city. It helps them decide how the competition is distributed.
- Analyze the correlation between the national restaurants the immigrants of that country who live in that city. It can be seen that nationalities are more likely to open national restaurants. This, for example, can be used by a bank to offer to certain people from those countries facilities to open a restaurant.
- The first part of previous point can be repeated for tourism.
- Sociologists. There are national restaurants whose importance does not seem to be correlated with immigration (entrances and exits) or tourism (entrances and exits). This happens for example with Italy, Greece or Mexico that have more restaurants than they would be due for migratory or tourist movements.

This analysis is a first step to look for what other factors influence (cultural, historical, marketing, ??)

- Foursquare. Looking at the conclusions of this analysis could.

. Allow new types of searches not by circles, but also by square or rectangle which would facilitate the management of overlaps in the areas if we need to divide the search into smaller areas for whatever reason.

. Improve the precision of the searches and when someone put in SEARCH a country name they would translate it into a category and search for it (it is the approach that I use). This can be extrapolated to other categories of not only 'FOOD' where I focussed.

1.3. MAIN DESIGN CRITERIA:

1- Flexible to use directly with any city.

The manual intervention will be restricted to the introduction of parameters that allow to improve the analysis.

Therefore, the city is divided geometrically into circles, NOT INTO neighborhoods, postal codes or other types of divisions that may depend on the city, which would avoid being able to do "universal" searches of any city.

2- Minimize the number of calls to Foursquare and rework.

To avoid exceeding daily limits. Even so, the solution is designed so that, if necessary (eg: a very low area granularity is desired), the data obtained from foursquare can be saved and the process restarted the next day without having to repeat most of the searches.

3. View the output data on a map

4. Minimize rework.

The data of the analyzed cities are saved in files.

One for each city and the other that serves as a catalog and in this it is saved for each selected city its chosen parameters.

Thus for a city already processed the user can choose to retrieve the data from it (avoiding access to Foursquare) or to reprocess and update these.

1.4. DETAIL OF THE FUNCTIONALITIES OFFERED BY THE TOOL:

For a city chosen by name (whichever you want).

- It shows the city map including.
 - a. The limits you decided
 - b. The original "center" obtained from Geolocator.
 - c. One big circle used to search directly restaurants of countries that have less than 50 restaurants in the whole city.
 - d. City divided in small circles, within each one it shows which country has more restaurants in that area and how many.
Each country is painted with a certain colour.
- You can choose any country and it will shows you how many restaurants there are in each area.

Only for countries with more than 50 restaurants throughout the country. This is a constrain imposed only to save access to Foursquare, but it can be modified if needed

1.5. PARAMETERS THAT THE USER CAN CHOOSE:

- The city to be viewed (and if it has already been processed, if you want to retrieve your data and view or reprocess it).
- The city limits to analyze: north, south, east and west from the geometric coordinates returned by Geolocator.
- The radius of the circular areas that divide the city: when smaller is the radius greater is the granularity and precision, but with a greater consumption of calls to Foursquare. After the initial analysis it has been seen that between 20 and 40 is a good number of areas to divide the city (which is equivalent a radio of the areas between 1000 and 1500 m)
- Country from which you want to view the distribution of its restaurants in that city.

2. Required data. Origin, description and manipulation.

2.1. DATA ORIGIN:

Basically we need 2 types of data:

1- RESTAURANTS X COUNTRY.

The restaurants of the city within the circular areas in which we divide the city. For this we will use the foursquare API (with the SEARCH endpoint), although previously we will have to homogenize the catalog of categories using the CATEGORIES endpoint.

2- "OFFICIAL" LIST OF COUNTRIES. to match the nationalities of the restaurants and the names of the countries. For this, the same list that we used to obtain the immigrants from Canada that had all the countries was used, and thus I already have the number of immigrants per country in case it gives me time to analyze later the correlation between restaurants in a country and immigrants from That country.

2.2. DATA. STRUCTURE AND OBTAINING METHOD

To find the restaurants of each country in a city, the first thing that occurs to you is to call the foursquare API "query" = name of the country to choose and for each of these, paint them based on their latitude and longitude.

However, this has several drawbacks:

1- Lack of precision.

The foursquare API when searching for a restaurant of a nationality using the command "query" (eg query = 'Italian') does NOT give all Italian restaurants.

If we want to get all the Italian restaurants we must use the category, which makes the process quite complicated, for several reasons:

1.1. Daily calls to Foursquare limited .

When analyzing the structure of the venues categories on Foursquare, it has been seen that in total there are 943 different categories that are grouped into 5 levels.

In the first level there are 10 values of which one is FOOD (its id category is '4d4b7105d754a06374d81259' and it is the one we will focus on. Additionally, at level 5 there are only Brazilian restaurants of different types.

Restaurants in different countries have a different grouping. Thus, for example, all Afghan restaurants are grouped into a level 2 category, while Ethiopians are all grouped into a single level 3 category and Brazilians are all under a level 4 category that has a breakdown by type in Level 5.

This has been solved by creating a dataframe with as many rows as FOOD categories (373) and the 5 levels of categories in the columns.

So for each FOOD category you look at what level it is and look for the upper categories on which it depends until reaching level 1 (FOOD).

Let me remember that for this project I am only interested in restaurants that can be associated to specific countries and NOT all types of restaurants

1.2. Non-coded country names.

The names of the categories of restaurants in the different countries are not coded in order to 'easy match' with a list of countries.

This led to a long process of analysis and modification of the names of countries to be able to match with the restaurants of their nationality.

Some of the most important tasks to achieve that were:

- Non-significant names were removed from the official list of countries. For example: of, states, Republic, People, of, the, ...

Leaving the most significant at the beginning of the name.

- For each country, the first 6 letters were taken as a code to search into the restaurant category list. (This achieved matching of 57 countries).

The search was repeated with only the first 5 letters of this new significant name of the country and then repeated with the first 4 letters.

- Still there were countries that did not match their type of cuisine.

For example: France with French and others that did not have their own kitchen, but grouped (thus Norway, Sweden, Denmark were assigned Scandinavia to be able to fit in with Scandinavian restaurant or Jamaica was assigned directly to cajun cuisine).

At the end of this process, a matching of 68 countries was achieved, among which I made sure that they were the main recipients of immigration in Canada (in case I later wanted to make the correlation between immigration and country restaurants).

All this required an important initial work to standardize levels (previous point) and name of countries.

2. The SEARCH endpoint has a limitation of 950 search per day and does NOT return more than 50 restaurants with each call.

However, there are many large cities that have many more restaurants. Therefore, attempts have been made to optimize this search to the maximum in several ways:

a- The user delimits the city limits as he wants (choosing the north, south, east and west limits). This is essential, since the center of the original search (the latitude and longitude provided by Geolocator) does not have to be in the center.

b- The city is divided into circles with a default radius of 1,500m (but can be modified by the user). This value has been chosen to, on the one hand, minimize the number of calls to foursquare and, on the other, ensure that there are not many more than 50 restaurants in that area (otherwise we would lose them).

c- For each country, it is sought how many restaurants there are in the city of that nationality and if they are less than 50, the distribution by area of those restaurants is no longer sought (to save calls to foursquare with countries with insignificant numbers of restaurants).

3. Foursquare only allows you to search circular areas (from the radius) and NOT in rectangles that, for my need, would have been ideal.

This could have been solved using only the internal square of the circle sought, but with this, it would be wasting searches for restaurants that were outside that internal square and this goes against the optimization of calls to Foursquare that I was looking for, in addition to complicating more the project (although visually a grid would have been better than overlapping circles).

When looking for circles you have to overlap them to make sure that we minimize the number of restaurants. not assigned.

This forces to use later the 'UNIQUE' function so that in the total count of restaurants of a nationality for the whole city there will be no duplicates. That's because a restaurant could be assigned twice to different circles if it is place in the overlapping area of these circles.