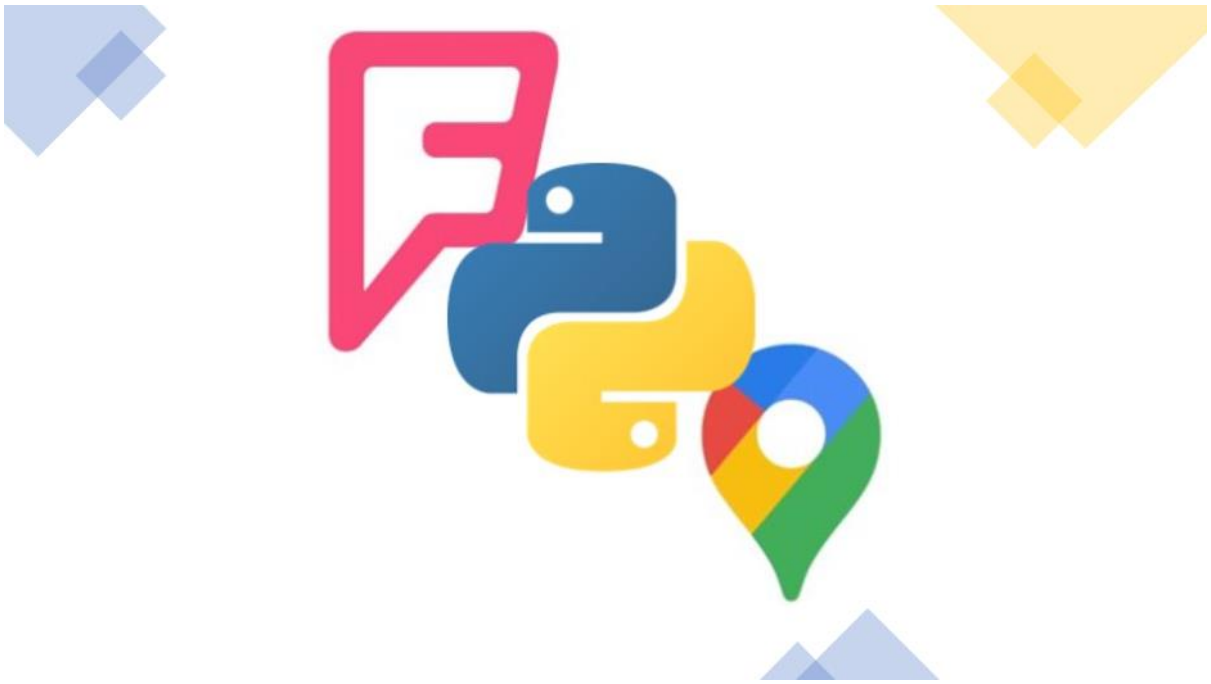
Smart travel with Python, Foursquare and Google

Final project Data Science Courses

1. Introduction

Whether you are on vacation in a foreign country, or traveling in your native land, it is convenient to use tools to narrow down and select our places of interest.

Most of us at some time in our lives have faced the situation of needing to know certain places, either for leisure or business. This text describes how to use libraries and simple Python code to automatically gather mass opinions from multiple users of the Foursquare intelligence platform and represent these results on a free Google Maps map. Let's go!



Three platforms available for our most common technology needs.

"Waiting for the times of normality and contact with people to return, to travel and know the world"

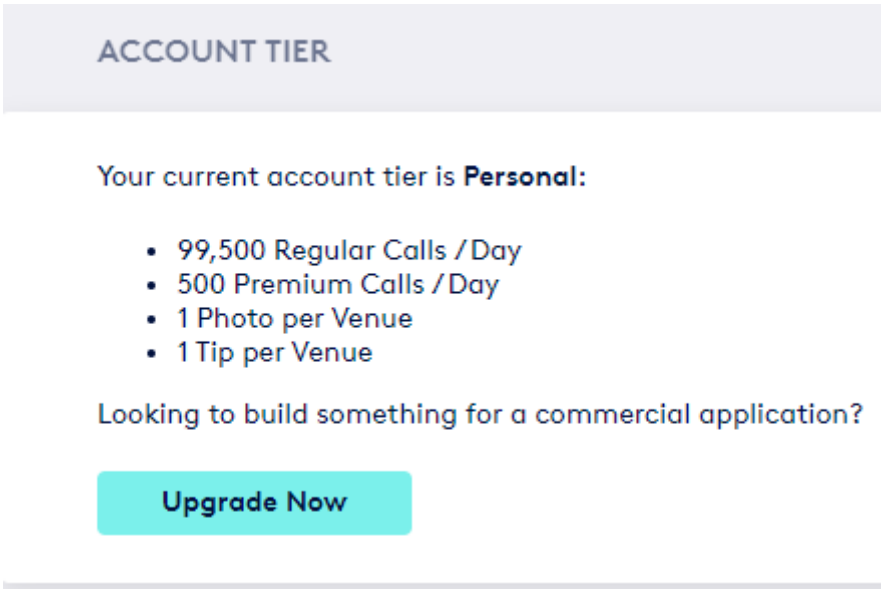
2. Traveler's data

The objective of analyzing large amounts of data is to offer a route proposal to the user according to his preferences. The base data analysis is provided by the Foursquare location intelligence platform, which is used by companies like Spotify and Uber to power their own businesses.

In its City Guide portal, it is possible to carry out quick and advanced searches for places of interest differentiated by categories. Within each result it is possible to identify data provided by the community such as comments, scores, location, schedules, etc.

Foursquare web portal for avenue search.

Foursquare has developer tools and plenty of documentation. We have created a free account where we can make code queries using the preferred language, in this case Python.



For each place of interest, although we have categorical and other free text data, we are particularly interested in those that serve to apply a preference criterion, for example distance, user scores, number of scores, average cost.

A quick query via API to Foursquare's database reveals all of this information in a structured manner in JSON format files, which we will use our loyal friend Python to build these queries and apply hundreds of queries depending on the number of locations and options to compare by the user. Go ahead Jupyter Notebook!


```
{
  "meta": {
    "code": 200,
    "requestId": "59a45921351e3d43b07028b5"
  },
  "response": {
    "venue": {
      "id": "412d2800f964a520df0c1fe3",
      "name": "Central Park",
      "contact": {
        "phone": "2123106600",
        "formattedPhone": "(212) 310-6600",
        "twitter": "centralparknyc",
        "instagram": "centralparknyc",
        "facebook": "37965424481",
        "facebookUsername": "centralparknyc",
        "facebookName": "Central Park"
      },
      "location": {
        "address": "59th St to 110th St",
        "crossStreet": "5th Ave to Central Park West",
        "lat": 40.78408342593807,
        "lng": -73.96485328674316,
        "postalCode": "10028",
        "cc": "US",
        "city": "New York",
        "state": "NY",
        "country": "United States",
        "formattedAddress": [
          "59th St to 110th St (5th Ave to Central Park West)",
          "New York, NY 10028",
```


JSON response structure with data dictionaries by avenue Corresponds to a Premium type query.

The development considers a first section for data capture by the user, where the simple use of "*input*" and "*print*" functions is preferred over the construction of a GUI with libraries such as *Tkinter*. Not a marketable idea, right?

A second section that applies all the versatility of Python to apply query loops and comparison algorithms to select the places of interest that best suit the user's preferences. Finally, Python gives us a preference box which we can track geographically on Google Maps, thanks to the latitude and longitude data Foursquare gives us for each search.

You can access the full code in the following GitHub repository.

vertexavier35/Coursera_Capstone
Project Capstone Datascience Cousera Foursquera categories list, more info…
github.com

3. Data in action

The development is applicable anywhere on the globe, of course it is better to use it in places with high traffic or where it is estimated that there is a vast influx of people with diverse interests.

As an example, we helped a lost traveler staying at the Hotel X in Toronto, Canada. We enter test values that the code summarizes as follows.

```
The journey will start at the following address: 111 Princes, Toronto, Canadá

The journey will end at the following address: 111 Princes, Toronto, Canadá

Summary of selected activities

Arts & Entertainment at 10:00:00 according to rating criteria
Food at 14:00:00 according to distance criteria
Event at 15:30:00 according to rating criteria
Arts & Entertainment at 18:00:00 according to tips.count criteria
Nightlife Spot at 22:00:00 according to price.currency criteria
```

Data entered by user using the Python input function.

The journey that starts in the direction of the hotel and ends in the same place. Additionally, the traveler wishes to carry out activities related to art and entertainment, food, events and nightlife at specific times. It is the mission of the algorithm to find the best places and make a recommendation.

As an additional configuration, a comparison number is established between 4 places of the same category, and within a search radius of 5 km.

```
latitude = location.latitude
longitude = location.longitude
```

```
Enter the number of places or attractions among which to generate a comparison according to the defin
ed criteria: 4
Enter the radius (m) of opportunity search: 5000
```

Data entered by user using the Python input function.

We make use of the versatile “Pandas” library to obtain as a result a DataFrame that contains the recommendations for each of the 5 categories indicated by the user. This table contains location and contact information, as well as the available criteria.

	id	name	categories	address	crossStreet	lat	lng	distance	postalCode	cc	...
0	4ad94f83f964a520b91921e3	Union Station	Train Station	65 Front St W	btwn Bay & York St	43.645167	-79.380641	3279	M5J 1E6	CA	...
0	4adb6a84f964a520332721e3	Loblaws	Grocery Store	10 Lower Jarvis St	at Queens Quay	43.645427	-79.369789	874	M5E 1Z2	CA	...
0	4c253f69f1272d7f488084c5	Prince Edward Viaduct	Bridge	1 Danforth Ave	Bloor	43.676776	-79.358652	3603	NaN	CA	...

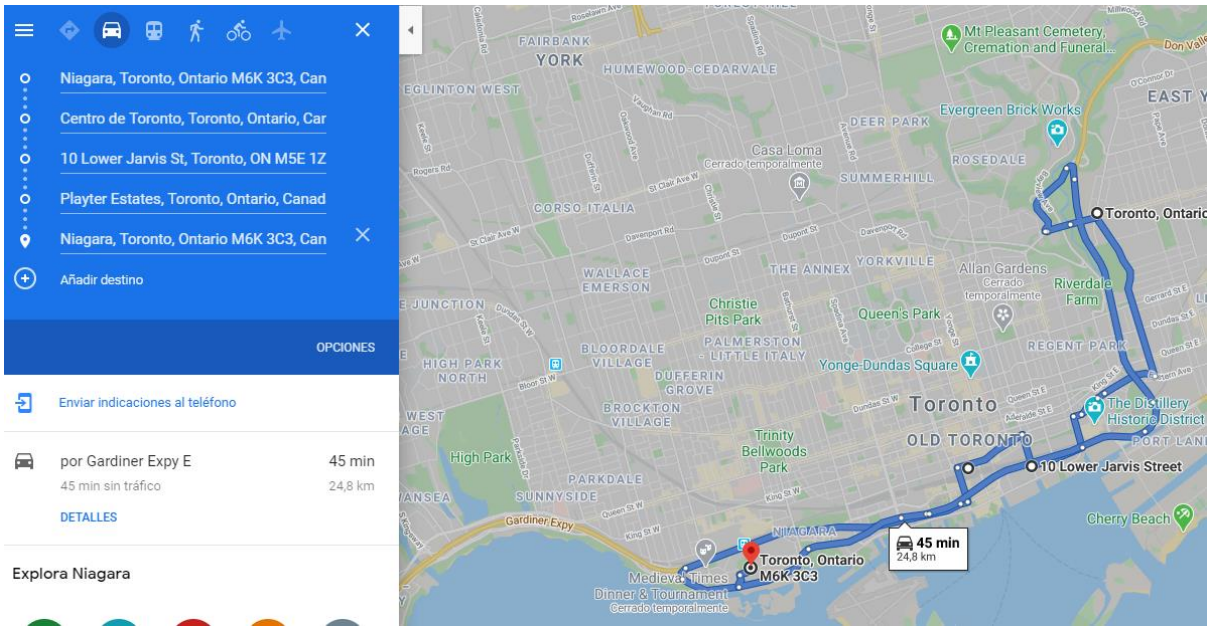
3 rows × 22 columns

DataFrame with final results for each category selected by the user, name section and location data 22 columns in total.

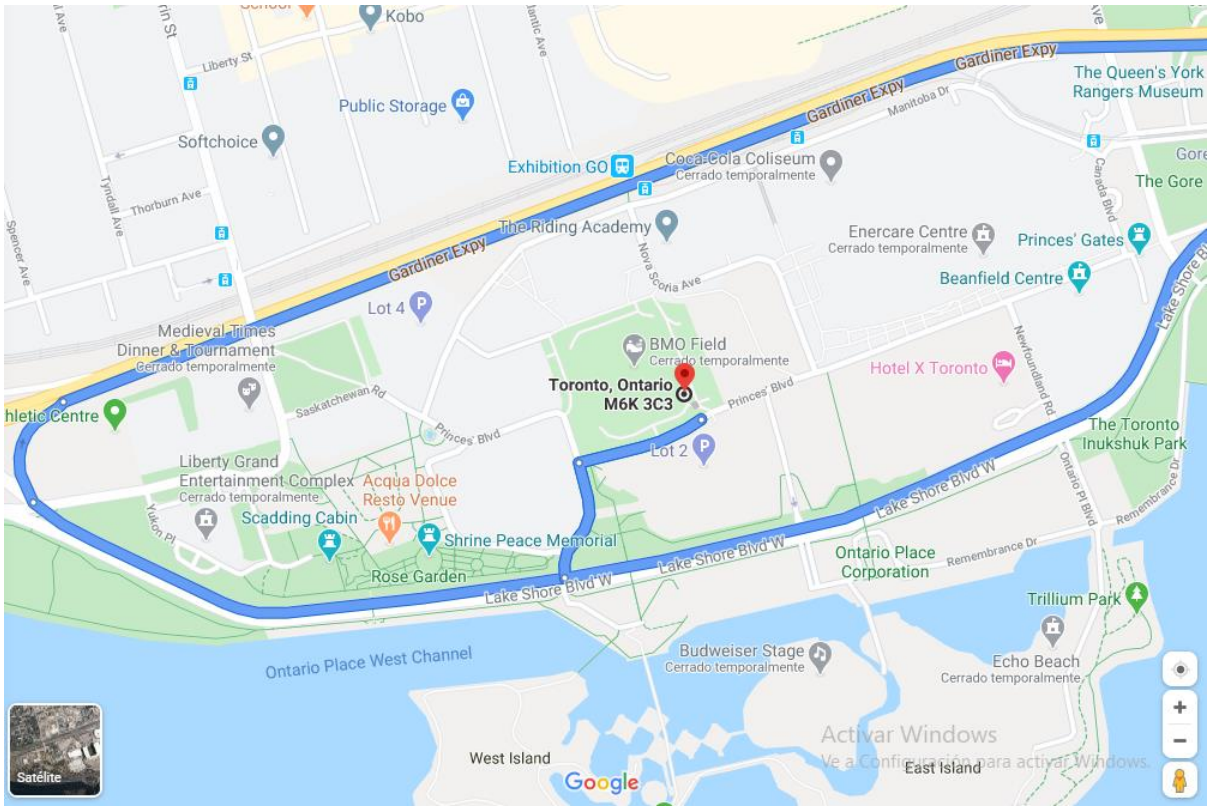
contact.phone	contact.formattedphone	rating	ratingSignals	tips.count	url	price.currency
8884386646	NaN	6.4	1425.0	208.0	http://www.torontounion.ca	NaN
4163040611	NaN	7.2	201.0	24.0	https://www.loblaws.ca/store-locator/?storeId=...	NaN
NaN	NaN	NaN	NaN	8.0	NaN	NaN

DataFrame con resultados finales por cada categoría seleccionada por el usuario, sección de criterios.

Then the magic part, take the delivered location data and build the API query to build in Google Maps the proposed route that starts and ends in the direction of Hotel X, and runs in the order selected by the user the best places identified by the algorithm, ah! and using car.



Map generated via API in Google Maps for locations calculated via Python.

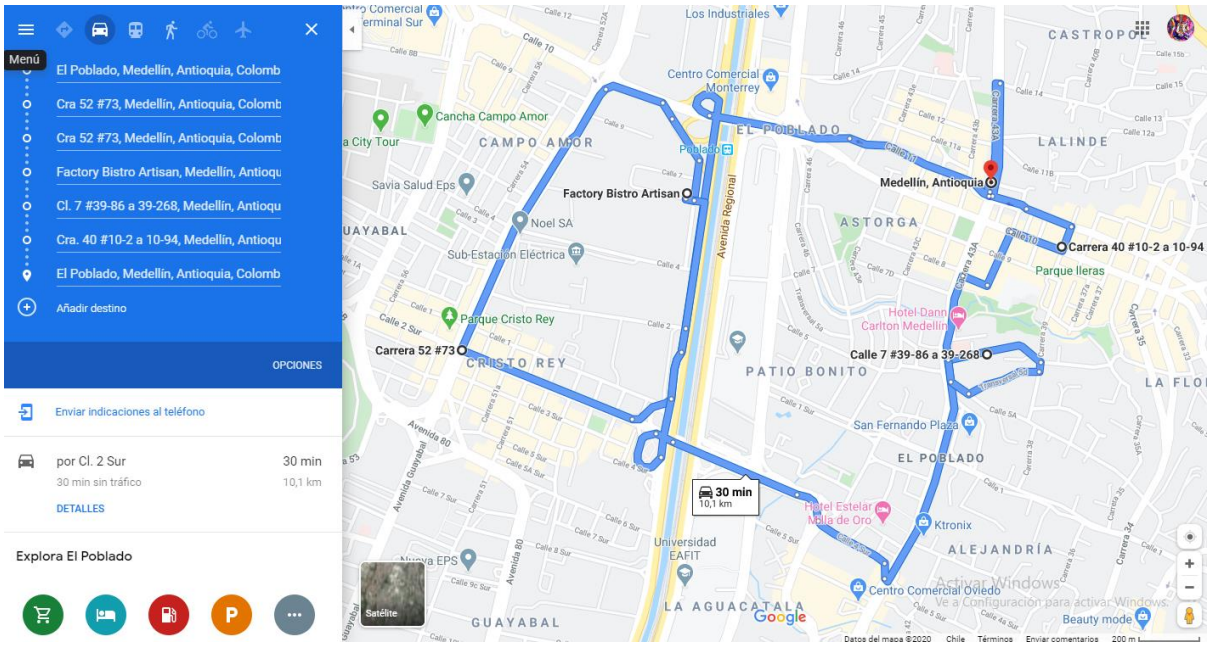


Map generated via API in Google Maps. The start and end of the trip are located near the Hotel X, Toronto.

Useful, isn't it? It is not a recommendation to trust blindly, but a good basis on which to start, which should be refined with information from acquaintances, hotel workers :), avoid dangerous places.

Did you see that the total path is 24.8 km? and the search radius was set at 5 km. This is because each new category search is performed from the previous point where the calculated recommendation was selected.

A more complicated circuit? in another city in the world?
A recommendation in the center of Medellin, Colombia, with 5 activities to be categorized by the algorithm Here is the result map.



Results for a more complex search with 5 required categories per user, in total 7 locations to be generated in Google Maps

	id	name	categories	crossStreet	lat	lng	distance	cc	city	state	...	adc
0	4e7b640dd164401b7fef1251	Parque Cristo Rey	Plaza	Guayabal	6.206695	-75.586056	1847	CO	Medellín	Antioquia	...	
0	4e7b640dd164401b7fef1251	Parque Cristo Rey	Plaza	Guayabal	6.206695	-75.586056	0	CO	Medellín	Antioquia	...	
0	54400244498e9cf3770022ad	Artisano	Italian Restaurant	NaN	6.211388	-75.579280	913	CO	NaN	NaN	...	
0	5374f143498e69e7fda4a685	Medical	Doctor's Office	NaN	6.206571	-75.570279	1131	CO	NaN	NaN	...	
0	4f9f5091e4b0ad70a8b49e3d	Lo Doy Por Que Quiero	Bar	NaN	6.209807	-75.568025	438	CO	Medellín	Antioquia	...	Ca 40

5 rows × 22 columns

Results for a more complex search with 5 required categories per user, in total 7 locations to be generated in Google Maps

4. Discussion

The results obtained, although they can be improved in many aspects, show that we can access and make use of free tools to obtain conclusions that are in any case valid and result from the opinions of the community over time, that is, we base ourselves on the average or most probable value of the criteria we selected. Improvements? of course:

- Comprehensive error and exception handling. Elimination of duplicate recommendations con zero distance.
- Better graphical interface, patience with mockups :(.
- Inclusion of subcategories of places to visit. This is possible because Foursquare has them correctly indexed in its databases.
- Better accuracy in the results? With a paid Foursquare account, it is possible. For 5 places of interest, only 100 comparisons per location per day can be accessed before reaching the daily consultation fee on the free Foursquare account.
- Apply temporariness to results. For example, consider travel times based on traffic and stay times at each location to build a more comprehensive itinerary.
- Generation of map with location (e.g. with the Folium library) of the compared and neglected places. Even the location can be highlighted with a larger or smaller size depending on the value of the criterion contained.
- Cross referencing more than one source of information, e.g. Foursquare + Tripadvisor, government source on places not recommended for a foreigner, among other improvements.

.

5. Conclusion

The data is there, certain companies placed the infrastructure and capital, the community shaped the databases with huge lakes of data. Whether free or paid for, the data is there, and only our curiosity and enthusiasm is needed to harness this potential to solve domestic, technological, business or personal entertainment problems.

This is a clear example of how it is possible to apply simple Python code to exploit the full potential of a location intelligence platform. Today the strange thing is to find businesses or platforms that do not have developer portals or APIS to generate connections, which means that we have at our disposal through an internet connection and a mid-range personal computer, an endless number of tools and data to solve practically any personal or business need.