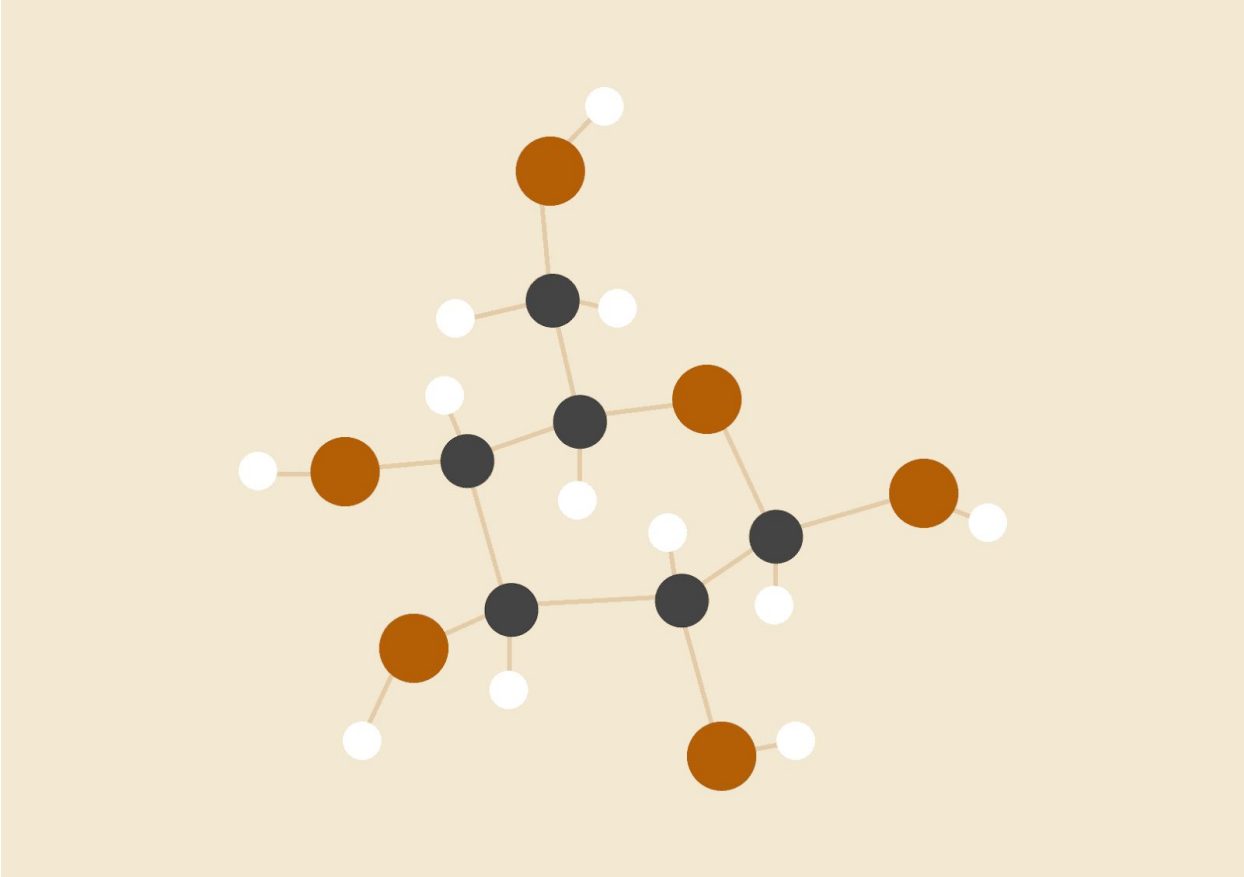
CAPSTONE PROJECT

Applied Data Science Capstone Report



Ruslan Moskalenko

01.26.2020

INTRODUCTION

Modern cities sleep less and less. There is more demand for services available at any time of the day. However, the available search engines tend to focus on “regular” business hours and they make it hard to find out what is available at non typical times. This project was an attempt to provide a tool to do this kind of research.

WHO WOULD NEED IT AND WHY

First, let me start with my own experience. As a frequent traveller I often found myself in unfamiliar places at unusual times. When I tried to use available search engines, they would tell me what is popular at regular hours. Some engines have an option to look for “Nightlife”, but they use it as a generic term meaning mostly Friday night style entertainment, but I couldn’t simply ask what will be open around me at 2 am?

Then I started looking around. First observation was that highways are definitely busy 24 hours. I looked at the people around me and a lot of them have a day schedule different from a typical get up at 7, work 9 to 5 and go to bed before midnight. There could be many reasons, some possible ones I can think of:

- People are trying to beat commute and start earlier and earlier or staying later and later
- Globalization. More people need to communicate with someone in a different time zone, sometimes many hours off
- Business and leisure travel. First, it often means a different time zone. Second, in many cases travellers want to use all the time they’ve got.

That means two things:

1. There should be more demand for people trying to find open services at given location and at given time.
2. There could be some interest to meet that demand. Entrepreneurs and business owners might be interested to analyze what venues are available to identify opportunities.

In both cases, ability to generate a report of available venues at given time and location could be beneficial, so I tried to use the skills I learned from the course to create such reporting engine.

DATA SOURCES

The main data source is the Foursquare database available through its API using a personal account. It provides venue locations in the area (virtually unlimited number of calls). It also has data about the venue hours, but those are available as Premium calls with a 500/day limit.

The secondary source of data is Opencage DB also available through a free API. It is used to resolve an address to geographic coordinates.

METHODOLOGY

Here is how that report is implemented.

User needs to provide the following input data:

- Target Address. It's a US address written as a string.
- Day of the week. 1-7
- Time of the day. 24HH format, for example 0130 means 1:30 am

There are also a couple of optional parameters:

- Target Name. It will be used to generate a label on the map.
- Radius. Default is 1000 meters to keep it walkable.
- The venue categories. Need to provide Foursquare category IDs.

The program will do the following steps:

1. Convert the target address into geographical coordinates
2. Get a list of the venues in the specified area for given categories.
3. Request hours for every venue. This is the most complicated case for a few reasons. First, those are premium calls, so this is the most precious resource. Second, Foursquare has two kinds of schedule data: hours and popular. Hours are the stated hours and popular are the popular times. In some cases both are present, in some are only one of two and in many cases none. So there should be some way to resolve potential conflicts. There could be many different approaches, the one implemented in this program would mark a venue open if either hours or popular shows this venue as open at this time. Also, Foursquare is

using some interesting format. For example, “day: 1, hours: start: 2000 end: +0200” means the venue is open until 2:00 am the next day. That makes parsing data a bit more complicated because we need to check the current and the previous day. At the end of this step we mark every venue’s field Open as either True or False.

4. Filter Open venues. To simplify processing, we leave only open venues. At this point we have all the data we need in a table.
5. “A picture is worth a thousand words”. To make data easier to understand, they are shown on a map using Folium. The target address is placed in the center and the venues are marked around it. Each venue has a label with a name and a category and colored based on the category.

RESULTS

Results are generally positive. The application is working as expected and it produced reasonable reports in a convenient format.

A few examples are included below.

Case 1: Midweek, 22:30. There are 3 venues open within 1000 meters.

Case 2: Midweek, 00:30. The radius had to be increased to 1500 meters to find 2 venues.

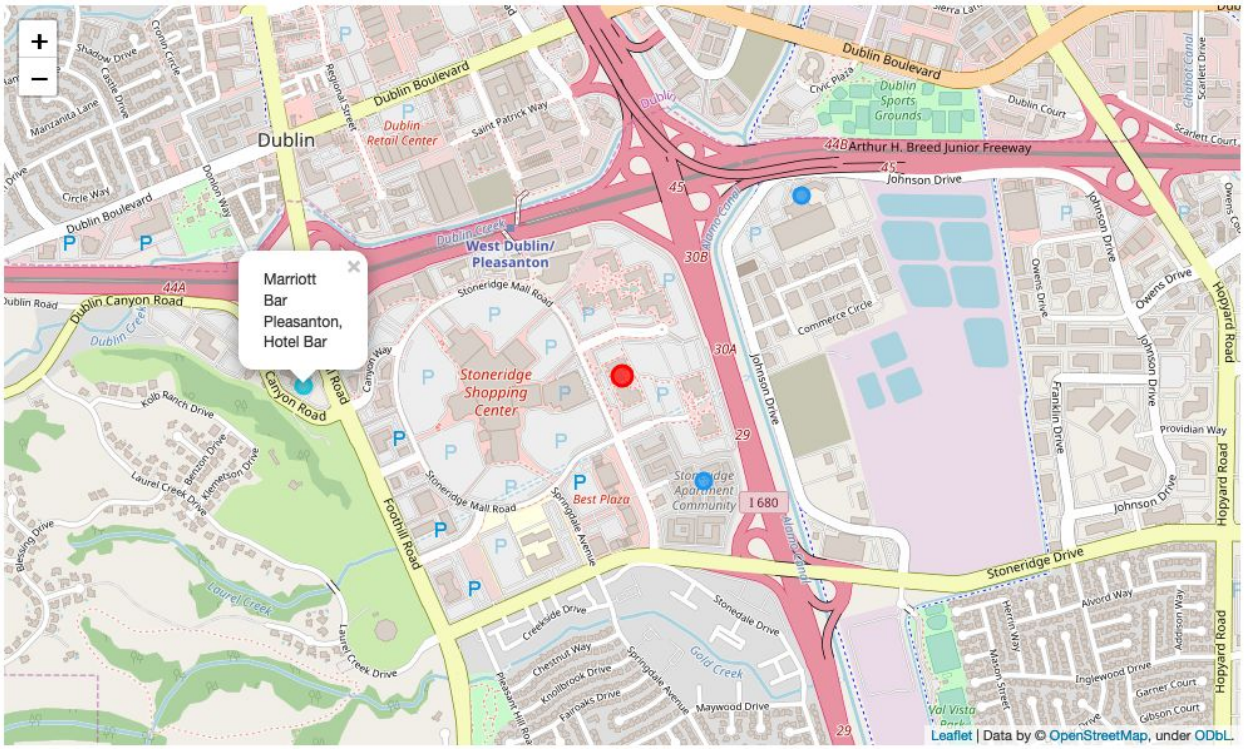
Input 1:

Input data

Target location. Enter a US address. Target day. Day of the week 1-7 Target time. "0000"-"2359" Target radius. In meters

```
target={}
target['address']="6200 Stoneridge Mall Rd, Pleasanton, CA"
target['name']="Office in Pleasanton"
target['day']=3
target['time']="2230"
# These values are optional
target['radius']=1000 # Keep it walkable, about 1 mile
target['latitude']=None # We'll find out later based on the address
target['longtitude']=None # We'll find out later based on the address
```

Output 1:



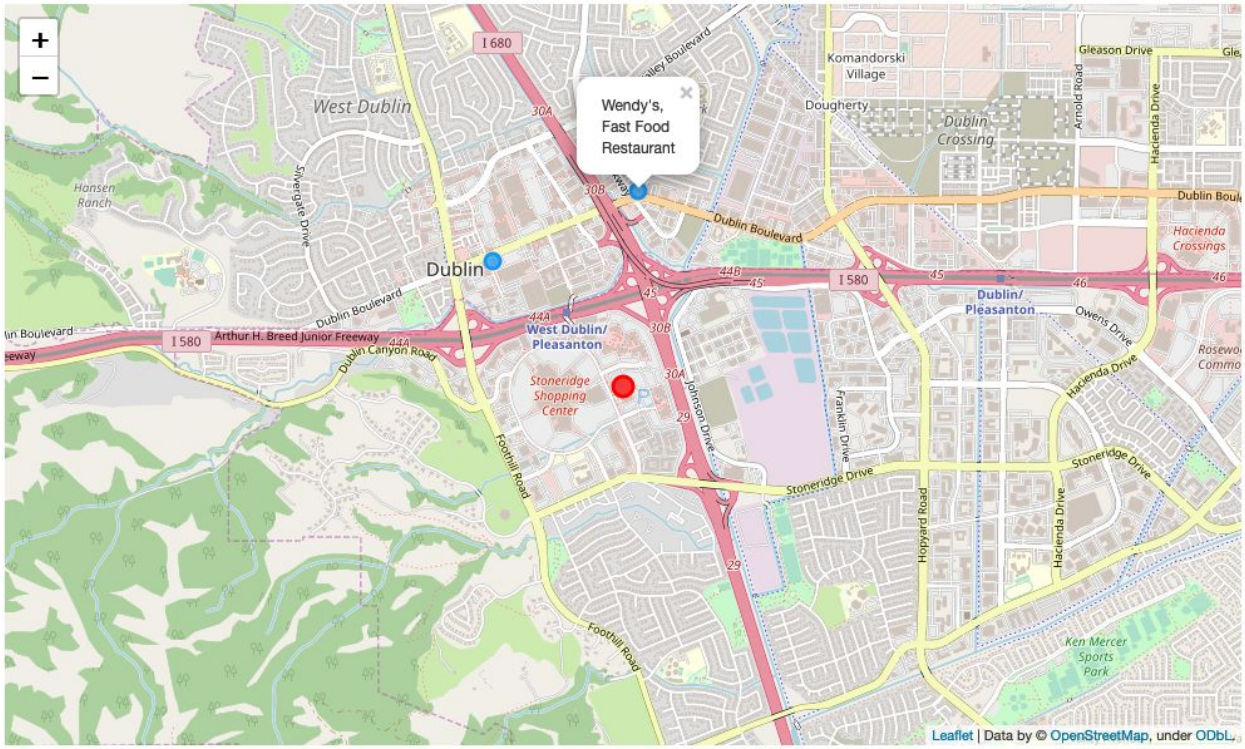
Input 2:

Input data

Target location. Enter a US address. Target day. Day of the week 1-7 Target time. "0000"-"2359" Target radius. In meters

```
target={}
target['address']="6200 Stoneridge Mall Rd, Pleasanton, CA"
target['name']="Office in Pleasanton"
target['day']=3
target['time']="0030"
# These values are optional
target['radius']=1500 # Keep it walkable, about 1 mile
target['latitude']=None # We'll find out later based on the address
target['longtitude']=None # We'll find out later based on the address
```

Output 2:



DISCUSSION

Obviously, due to limited time, this project is more a proof of concept. However it pointed out some interesting observations:

1. Data quality. While Foursquare has very detailed data about venues locations, the hours data are not the same quality. A lot of venues have either missing data or

conflicting data between “hours” and “popular”. Also, they don’t seem to be reflecting holiday schedules either. So I wouldn’t rely on them around the holidays or if the report returns a handful of venues. But the more results are found, the more likely that at least some of them will be correct, so that would be acceptable for the purpose of this project.

2. Data cost. Requesting hours is a premium call. That means if you want to run a series of reports for different times (what is open in this area at every hour of the day?) it would make sense to modify this program to get the venue details first, cache them and then run all reports against that cache.
3. Data extremes. As expected, there is a big difference between large cities that never sleep and small sleepy towns. While this report provides meaningful results in metro areas, it won’t find much in the rural areas - there is simply not much to be found there.

CONCLUSION

Overall, it has been demonstrated that the proposed approach was able to achieve stated goals.

REFERENCES

1. GitHub Link to the notebook; https://github.com/rmoskalenko/Coursera_Capstone/blob/master/Capstone%20project%20Final.ipynb