4/1/2021

Capstone Project

The battle Of Neighborhoods

Evert Johannes Woudstra
CONTACT INFORMATION

Table of Contents

1 INTRODUCTION 2

1.1 Context of the project 2

1.2 Business problem 2

1.3 Audience and stakeholders 2

2 DATA DESCRIPTION 3

2.1 Data sources 3

2.1.1 Location data neighborhoods 3

2.1.2 Venue Data: The Foursquare database 3

2.1.3 Venue Data: Feature selection 4

2.2 Data cleaning 5

2.2.1 Removing unnecessary features 5

2.2.2 Removing duplicate values 5

2.2.3 Assigning venues to correct neighborhood 5

1 Introduction

This project is the final assignment of the IBM Professional Data science Certificate. This certificate consists of a series of 9 courses in which data science skills, including Data Science Methodology, data mining and analysis with python has been studied and applied. The main objectives for this project are the following:

- leverage location data provided by Foursquare and the Data of Amsterdam website
- applying data science skills in machine learning and data visualization

The results will be presented in this report as well as in a blogpost online and a Jupyter notebook published on Github.

1.1 Context of the project

Amsterdam is the capital city of the Netherlands and is famous for its multicultural identity. With 872.922 inhabitants from 177 different nationalities, it belongs to one of the most divers cities of the world.
The city has the largest number of bars and pubs of any city in the Netherlands, although the number is declining over the last decade. Small traditional pubs are transitioning to more fancy venues targeting the market of “hipsters” and the modern millennial customer seeking for experiences and high-quality food.
The opposite trend is visible in the category of restaurants. Since 2010 the number of restaurants is increasing not only limited to the old city center but also surrounding neighborhoods. New forms like food markets as “Rollende Keukens” and concepts restaurants are being developed over the last few years. The growing popularity of Amsterdam worldwide and the annually increasing number of tourists indicates good prospects for opening a new restaurant concept.

1.2 Business problem

Opening a restaurant in a city like Amsterdam is a complex project. Although a good food concept can be a profitable and fulfilling business, reality is that many restaurants fail during their first year, mainly due to a lack of planning and having a sound business plan.
In the business plan many topics like funding, choosing a concept, advertising and logistics and choosing an optimal location has to be carefully considered.
The scope of this project is addressing the question what the optimal location could be for opening an Italian restaurant in the city of Amsterdam with a concept of choice.

1.3 Audience and stakeholders

This project could be potentially interesting to:

- Entrepreneurs who want to start a food business in Amsterdam
- Brokers to advise people on buying properties based on their business plan
- Data analysts interested in GeoPandas for fitting location points into polygon objects

2 Data description

To find an answer to the business problem we will make use of the following two data sources:

- Geographical data provided by the Amsterdam Data Website
- Location data provided by Foursquare

In this chapter these sources are further introduced and explained.

2.1 Data sources

In this chapter data courses are defined and described. Also, some introduction has been made addressing the possibilities and difficulties handling those courses.

2.1.1 Location data neighborhoods

The Amsterdam data website (https://maps.amsterdam.nl) has all the geographical data this project needs. The geographical data of the neighborhoods is downloadable in different formats like .csv and Geo Json.
The advantage of the Geo Json format, which is used in this project, is its precise definition of each neighborhood where the borders are defined by a polygon (fig 1) consisting of al the latitudes and longitudes.

```
In [15]: amsterdam_data.geometry.head(1)

Out[15]: 0    POLYGON ((4.90326 52.37658, 4.90298 52.37668, ...
         Name: geometry, dtype: geometry
```

Fig 1. Example of a Geo Json polygon object.

This list of location coordinates describes the borders of each neighborhood as shown in figure 2.

```
In [8]: p_list = amsterdam_data.geometry[0]
        p_list

Out[8]:
```

Fig 2. Plot of a polygon entry in the geojson dataset amsterdam_data.

2.1.2 Venue Data: The Foursquare database

The second source which is used in this project is the Foursquare database.
Foursquare (https://foursquare.com) is a company which build and maintains a massive dataset of accurate location data. This data is freely available using the RESTful API.
After registration on their website on can do a limited number of queries with different search parameters so called “endpoints” and collect a dataset containing all the interesting venues a location has to offer.

Figure 3 shows what a RESTfull API url would looks like:

```
# create the API request URL
url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'
```

Fig 3. Example of quiry on Foursquare database.

As show in figure 3, one of the limitations of this API is the search method “radius” around a location of interest.
Since neighborhoods in old cities like Amsterdam tends to have irregular forms and sizes there is a need for a more specific definition of the search area.
If the radius has been defined to small, interesting venues could be missed in the neighborhood. On the other side, choosing a to large radius will result in duplicate venues and venues assigned to the wrong neighborhood.

Therefore, a method has to be developed finding most of the important venues in each neighborhood while preventing duplicates or venues wrongly assigned.

2.1.3 Venue Data: Feature selection

To retrieve only venues interesting for this project the following parameters has been user for the query.

Latitude and Longitude: This is the geographical location of the search point
Radius: radius within venue information will retrieved.
CategoryId: Here a sting of categories can be given to include in the search

The categories of interest are:

- Food: Find all venues for venues to assess the indirect competition
- Italian Restaurants: Find all venues for venues to assess the direct competition
- Arts and Entertainment: attractive venues for tourists \ possible business partner
- Nightlife: indicated for popularity for locals and tourists
- Travel and Transport: indication how reachable a potential neighborhood is

After performing the query, the above-mentioned features will be selected, cleaned, venues hot encoded and put into a data frame (fig 4).

	Neighborhood	Longitude	Latitude	Restaurants	Italian restaurants	Arts\entertainment	Nightlife venues	Trasport venues
0	Burgwallen-Oude Zijde	4.896919	52.372084	44.0	23.0	29.0	64.0	28.0
1	Burgwallen-Nieuwe Zijde	4.895055	52.374195	69.0	37.0	17.0	46.0	81.0
2	Grachtengordel-West	4.887844	52.373349	29.0	8.0	13.0	23.0	32.0
3	Grachtengordel-Zuid	4.892044	52.365930	39.0	29.0	31.0	35.0	27.0
4	Nieuwmarkt/Lastage	4.904592	52.371729	31.0	11.0	14.0	16.0	33.0
...	...	...	...	...	...	...	...	...
94	Bijlmer Oost (E,G,K)	4.982963	52.324360	4.0	0.0	3.0	4.0	2.0
95	Nellestein	4.999182	52.307183	1.0	0.0	4.0	0.0	5.0
96	Holendrecht/Reigersbos	4.976040	52.293713	5.0	1.0	3.0	2.0	4.0
97	Gein	4.994089	52.296346	1.0	0.0	0.0	1.0	4.0
98	Driemond	5.011249	52.311792	0.0	0.0	0.0	0.0	0.0

Figure 4: Final data frame containing the features to be analyzed

2.2 Data cleaning

An essential step in Data Science is cleaning of data before using in any analysis.
When data does have flows or is not in the correct format for the tools to be used the results can be sub optimal and results inconclusive.
In this chapter the most important data cleanings will be mentioned and explained.

2.2.1 Removing unnecessary features

After performing the query on the foursquare data base, it appeared there where contaminations of venues types outside of categories provided (fig 5).

Venue Category	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	id	Venue Latitude	Venue Longitude	Venue Category id
Bakery	1	1	1	1	1	1	1	1
Bar	3	3	3	3	3	3	3	3
Bed & Breakfast	2	2	2	2	2	2	2	2
Bistro	1	1	1	1	1	1	1	1
Brasserie	6	6	6	6	6	6	6	6
Burrito Place	1	1	1	1	1	1	1	1
Café	1	1	1	1	1	1	1	1
Chocolate Shop	1	1	1	1	1	1	1	1

Figure 5: Found venues outside of the predefined categories to be removed.

For example, the query in the restaurant category yielded also venues like “Hotel”, “Bar” and grocery shop.
After a short investigation of the unwanted categories, python techniques for data frame dropping have been used to get rid of unwanted venues.

2.2.2 Removing duplicate values

One of the disadvantages of the Foursquare database is the lack of flexibility in defining the search area. A search can only be made by a circular of rectangular area.
Due to overlap of the search areas for small crowded neighborhoods in de city center, there will be a huge number of duplicated venues which have to be removed.

2.2.3 Assigning venues to correct neighborhood

As introduced in the previous chapter. Searching neighborhoods varying in shape and size by an API request with a fixed radius can result in either missing many venues or having duplicated values spread over multiple neighborhoods. To overcome this problem an optimal radius has to be chosen and the dataset will be cleaned of duplicates. Finally, each venue assignment will be checked based on its location and the definition of the neighborhood polygon.