Assignment: Applied Data Science Capstone project

Marc Jellema

December 21, 2020

Abstract

This report is a deliverable for my capstone project *Battle of the neighbourhoods*, part of my graduation for my **IBM Data Science Professional certificate**. The assignment was to define a challenge that was to be solved by using the FourSquare API. Nowadays almost anything you need can be found online, including public data sets to use freely. I tapped into the public domain data sets with key figures from the Netherlands. Combining the FourSquare with the Dutch key figures showed a strong correlation between house prices and the number of restaurants in the neighbourhood. Before this correlation is to be used in other algorithms, I strongly advice to use more detailed data sets to underwrite the conclusion found during my assignment. Feel free to send me an email on mjellema@omnia.nl if you like to discuss this report in more detail.

Introduction

I tasked myself with the assignment to find additional ways to enhance automatic estimation of house prices in the Netherlands. The use of the **FourSquare** API is mandatory for this assignment, part of the IBM Data Science professional certificate [1]. As a tech savvy person myself, I'd heard of FourSquare and I do remember seeing the FourSquare logo on venues in the past, but never came across its footprint in the Netherlands. Wondering if I could use the FourSquare API in finding an answer for a challenge in the Netherlands, I did some research and as it turned out, the FourSquare database was more than comprehensive enough to use for data science challenges in the Netherlands as well. Not knowing a single person actively using FourSquare to let their friends know where they check-in, it puzzled me how FourSquare actually accumulates their data. James D. Walsh wrote a nice article [2] on how FourSquare accomplishes this. Enough about FourSquare for now, let's focus on introducing the question I want to answer with help of the FourSquare API and the data it can provide:

> *Based on **FourSquare** data, can we find a correlation between the average house price, the number of restaurants and the number of inhabitants in the provinces of the Netherlands?*

Although the COVID pandemie is having an impact on the overall economy world wide, it seems house prices have not yet taken a hit due to the pandemic. Startups like **Promodomo** [3] use algorithms to calculate house prices, even when they are not on the market yet. Algorithms like Promodomo's always benefit from additional sources to enrich their calculation. Searching for a correlation based on FourSquare data is just one of the many possibilities to augment the estimation of house prices and this is why I choose to take this challenge and come up with answers in this capstone project of my IBM Data Science Professional certificate.

1 Data

To answer the challenge I needed the following data points:

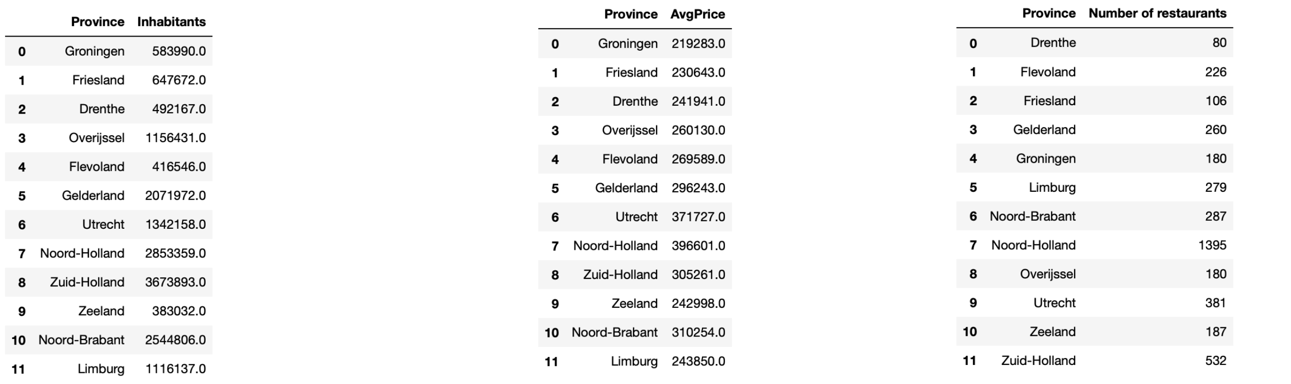
1. Number of inhabitants per province in the Netherlands
2. Average house price per province in the Netherlands
3. Number of restaurants per province in the Netherlands

Ad 1. The number of inhabitants per province in the Netherlands can be downloaded from the Centraal Bureau voor Statistiek **(CBS)**: a government owned, freely accessible web site with tons of statistical datasets (over 4K of them!) to use in all kind of analysis. I did some research on which table would best provide my need for the number of inhabitants per province and its title turned out to be `[Regionale kerncijfers Nederland]`. The resulting data frame, including some data cleaning done like removing substring `[(PV)]` from the province name, is in figure 1a.

Ad 2. The average house price per province in the Netherlands in 2019 also can be downloaded from the CBS, this time based on the table with title `[Bestaande koopwoningen; gemiddelde verkoopprijzen, regio]`. After a bit of cleaning and tweaking (i.e. 'Friesland' ≠ 'Fryslân' and some column renaming had to be adjusted to make the dataset comprehensible for our international readers) the resulting data frame is in figure 1b.

Ad 3. The third dataset, the number of restaurants, came from FourSquare. I needed to find all restaurants within each province of the Netherlands. The FourSquare API had the following two drawbacks that I needed to overcome: the non commercial API limits the returned number of venues per call to max 100 and although the results have a `[state]` field, the API doesn't allow searching per state. The first drawback, the limit of max 100 venues per call, I overcame with the help of the material I found of a fellow course student Guillermo (G.) Bareirro [4]. FourSquare returns max 100 venues per call, but if you make the call specific enough, the total results will not grow above the limit. Using the categories listed bij G. Bareirro, I was able to split querying all restaurants into their separate categories and thereafter grouped and summed them with pandas standard data frame functionality. To overcome the second drawback of not being able to query FourSquare by state, I realised an estimate of the restaurants per province would be sufficient for my analyses so I queried the restaurants (via their respective subcategories) in the capital city of each province. For the province of Noord-Holland I made an exception: the capital of Noord-Holland is Haarlem, but Amsterdam (the capital of the Netherlands) is by far a more important city in the province of Noord-Holland so I took the liberty of using Amsterdam instead of Haarlem as the anchor point for the FourSquare queries. The data frame is in figure 1c.

The three data frames used in the analyses for this assignment are in the images below:



	Province	Inhabitants
0	Groningen	583990.0
1	Friesland	647672.0
2	Drenthe	492167.0
3	Overijssel	1156431.0
4	Flevoland	416546.0
5	Gelderland	2071972.0
6	Utrecht	1342158.0
7	Noord-Holland	2853359.0
8	Zuid-Holland	3673893.0
9	Zeeland	383032.0
10	Noord-Brabant	2544806.0
11	Limburg	1116137.0

(a) Inhabitants from StatLine [5]

	Province	AvgPrice
0	Groningen	219283.0
1	Friesland	230643.0
2	Drenthe	241941.0
3	Overijssel	260130.0
4	Flevoland	269589.0
5	Gelderland	296243.0
6	Utrecht	371727.0
7	Noord-Holland	396601.0
8	Zuid-Holland	305261.0
9	Zeeland	242998.0
10	Noord-Brabant	310254.0
11	Limburg	243850.0

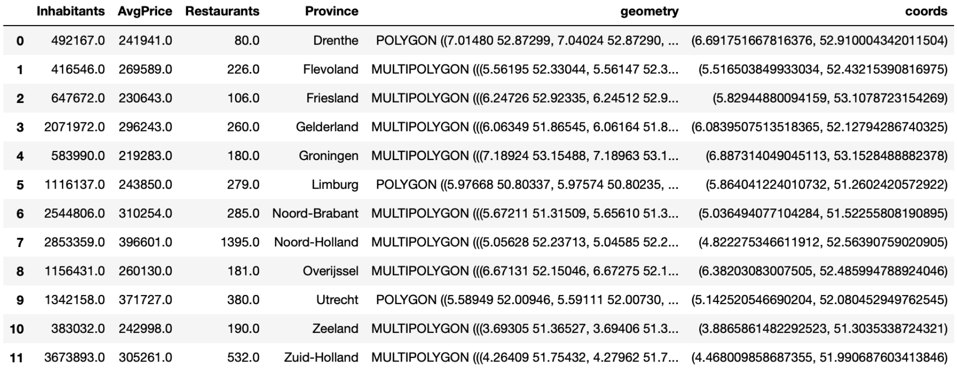
(b) Average Price from StatLine [6]

	Province	Number of restaurants
0	Drenthe	80
1	Flevoland	226
2	Friesland	106
3	Gelderland	260
4	Groningen	180
5	Limburg	279
6	Noord-Brabant	287
7	Noord-Holland	1395
8	Overijssel	180
9	Utrecht	381
10	Zeeland	187
11	Zuid-Holland	532

(c) Restaurants from FourSquare [7]

Figure 1: Cleaned data frames as used in analysis

After submitting this data section for the first part of my graduation I came across a dataset that turned out to be very helpful in showing the final results on a map of the Netherlands and it was too compelling for me to leave this opportunity out of this report. So the fourth part added to the data frames was a **GeoPandas** data frame with the geometries of each province in the Netherlands. Thanks to the excellent blog post of Artem (A.) Zapara [8] I was able to get the final bit of information I needed and added it to the resulting data frame of figure 2.



	Inhabitants	AvgPrice	Restaurants	Province	geometry	coords
0	492167.0	241941.0	80.0	Drenthe	POLYGON ((7.01480 52.87299, 7.04024 52.87290, ...	(6.691751667816376, 52.910004342011504)
1	416546.0	269589.0	226.0	Flevoland	MULTIPOLYGON (((5.56195 52.33044, 5.56147 52.3...	(5.516503849933034, 52.43215390816975)
2	647672.0	230643.0	106.0	Friesland	MULTIPOLYGON (((6.24726 52.92335, 6.24512 52.9...	(5.82944880094159, 53.1078723154269)
3	2071972.0	296243.0	260.0	Gelderland	MULTIPOLYGON (((6.06349 51.86545, 6.06164 51.8...	(6.0839507513518365, 52.12794286740325)
4	583990.0	219283.0	180.0	Groningen	MULTIPOLYGON (((7.18924 53.15488, 7.18963 53.1...	(6.887314049045113, 53.1528488882378)
5	1116137.0	243850.0	279.0	Limburg	POLYGON ((5.97668 50.80337, 5.97574 50.80235, ...	(5.864041224010732, 51.2602420572922)
6	2544806.0	310254.0	285.0	Noord-Brabant	MULTIPOLYGON (((5.67211 51.31509, 5.65610 51.3...	(5.036494077104284, 51.52255808190895)
7	2853359.0	396601.0	1395.0	Noord-Holland	MULTIPOLYGON (((5.05628 52.23713, 5.04585 52.2...	(4.822275346611912, 52.56390759020905)
8	1156431.0	260130.0	181.0	Overijssel	MULTIPOLYGON (((6.67131 52.15046, 6.67275 52.1...	(6.38203083007505, 52.485994788924046)
9	1342158.0	371727.0	380.0	Utrecht	POLYGON ((5.58949 52.00946, 5.59111 52.00730, ...	(5.142520546690204, 52.080452949762545)
10	383032.0	242998.0	190.0	Zeeland	MULTIPOLYGON (((3.69305 51.36527, 3.69406 51.3...	(3.8865861482292523, 51.3035338724321)
11	3673893.0	305261.0	532.0	Zuid-Holland	MULTIPOLYGON (((4.26409 51.75432, 4.27962 51.7...	(4.468009858687355, 51.990687603413846)

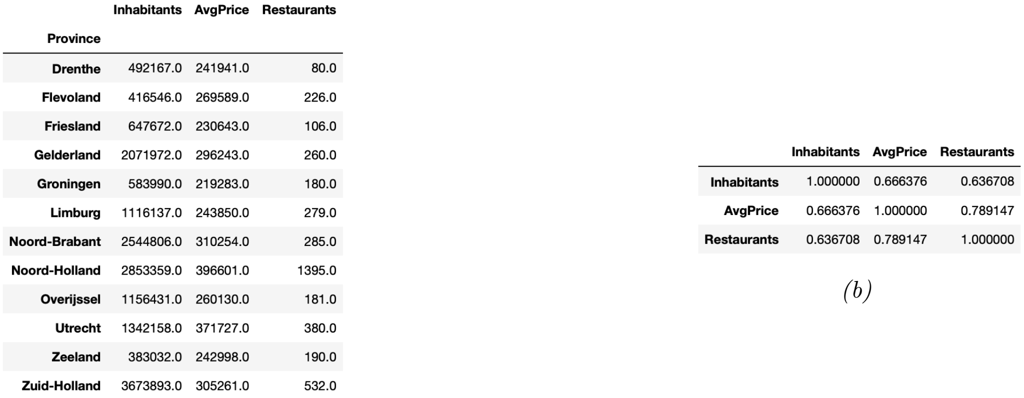
Figure 2: Result set for analyses and reporting

As always I was very impressed with the work of fellow programmers / students: almost any programming question already has been answered (although sometimes in parts) and can be found on Google. If you know the right keywords and have enough time to keep searching, you will find most of the puzzle pieces readably available. You just need to connect the dots so to say. On top of that, Python keeps amazing me for the strength of its coding syntax: import the right libraries and with just a couple of lines of code you can generate, filter and merge data frames into a data frame ready for takeoff.

2 Methodology & Analyses

Next to the data and the master data frame I explained in the previous data section, for this assignment I used a public GitHub repository [9] to store my documentation, images, notebook and report connected to this assignment. Everything was developed and programmed locally on my Macbook pro in a Python 3 / Anaconda setup and committed to the repository with the GitHub desktop client for OSX [10]. I used the regular Python libraries used in many data science assignments like `[pandas]` for handling data frames and `[requests]` for handling data retrieval via URL requests. For charting and graphics I used a combination of the `[matplotlib]` and `[seaborn]` libraries.

Finding correlations can be easy if you have the master data frame well prepared. The `[analyses data frame]` as shown in the data section (see figure 2) only uses one line of code to come up with the correlation matrix. Both the input (`analyses data frame`) and the output (`correlation matrix`) are in the images below:



	Inhabitants	AvgPrice	Restaurants
Province			
Drenthe	492167.0	241941.0	80.0
Flevoland	416546.0	269589.0	226.0
Friesland	647672.0	230643.0	106.0
Gelderland	2071972.0	296243.0	260.0
Groningen	583990.0	219283.0	180.0
Limburg	1116137.0	243850.0	279.0
Noord-Brabant	2544806.0	310254.0	285.0
Noord-Holland	2853359.0	396601.0	1395.0
Overijssel	1156431.0	260130.0	181.0
Utrecht	1342158.0	371727.0	380.0
Zeeland	383032.0	242998.0	190.0
Zuid-Holland	3673893.0	305261.0	532.0

(a)

	Inhabitants	AvgPrice	Restaurants
Inhabitants	1.000000	0.666376	0.636708
AvgPrice	0.666376	1.000000	0.789147
Restaurants	0.636708	0.789147	1.000000

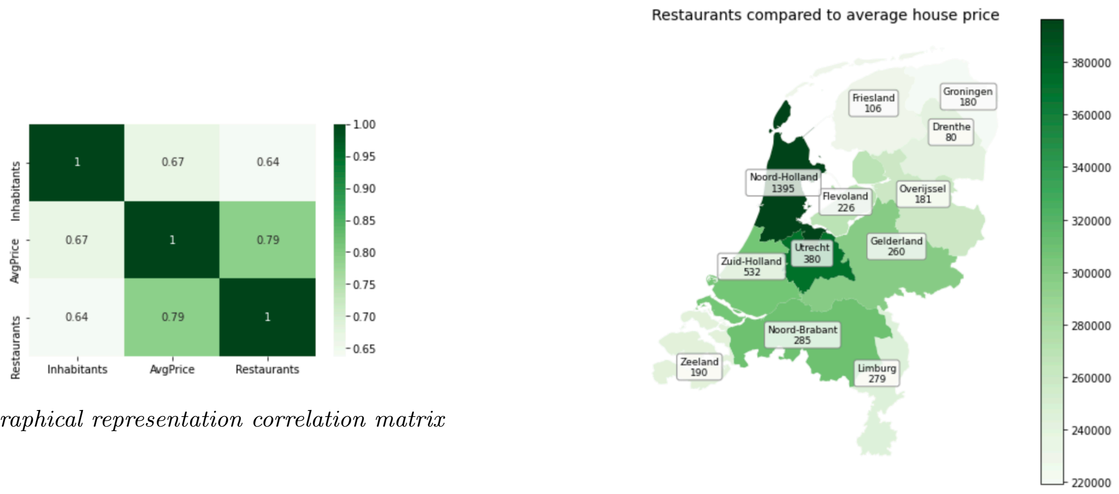
(b)

Figure 3: Analysis data frame and calculates correlation matrix

Based on the calculated correlation coefficients, I was able to do the analysis and come up with the desired results, which I compared to the rule of thumb mentioned on ResearchGate [11] about the value of correlation coefficients. The IBM Data Science Professional course on Coursera covers a lot of model training and learning modules and during the course I had plenty of opportunity to play around with most modern types of analysis and machine learning techniques. The question underlying this report however, doesn't need a lot of computer calculation power to solve. Basically the methodology followed to answer the challenge summarises as follows:

1. Get the venues data from FourSquare
2. Get the demographic data from CBS
3. Get the price data from CBS
4. Get the geo data from Arcgis
5. Combine & clean the data into a result set
6. Use default routines of Pandas to calculate the correlation coefficients
7. Analyse the results
8. Plot the results on a map of the Netherlands

3 Results



(a) Graphical representation correlation matrix

(b) Colormap of average house price and number of restaurants

Figure 4: Analysis results

Looking at the correlation matrix (see figure 4a), the number of restaurants in de province capital city has a **strong** correlation to the average house price in the respective province: the correlation coefficient has a value ≈ .79. This report started with the question if a correlation between the average house price and the number of restaurants exists and the answer based on my analysis is the following:

> *Based on **FourSquare** data, a statistical strong correlation between the average house price and the number of restaurants in the provinces of the Netherlands exists.*

The main question underlying this report raised another question: does a correlation exists between the number of inhabitants and the average house price? Looking at the correlation matrix, this correlation does exist but is less strong than the number of restaurants (.64 compared to .79).

I think the most important note int this results part is that with the right preparation, the right libraries and add-ons installed, performing a correlation analyses in Python is extremely powerful and easy to execute.

4 Discussion

Looking back on the results, one could say that statistically the answers found on the two questions hold, but of course there is a lot to say about the depth of this analysis and if the results are generally applicable in the real world. For example, let's look at the level of detail of the data sets used in the analysis. The data frames used are condensed data sets. Look at the data set with the average house price: this set holds the average price based on thousands of transactions. For more in depth analyses and better understanding of the results, it might be better to use the underlying data set of all these transactions and not the condensed average. The same holds for the set of restaurants. The correlation is now based on the number of restaurants in the main or capital city of a province. Taking just the restaurants of the focal point of the province was done due to the limitations of the free requests possible to FourSquare. If we would use a data set with all venues instead of just a this subset, the correlation could turn out differently. Last but not least, just looking at restaurants being listed in FourSquare, without examining the venues in more detail like looking at reviews and actual use (=number of guests per annum) of these venues, there is a lot of room for improvement on the thoroughness of the analysis.

5 Conclusion

To summarise, with the data sets I used, I've been able to give a statistical answer on both of the questions raised but before the results can safely augment an automatic house estimation algorithm, more work has to be done.

6 Acknowledgements

I would like to take the opportunity to thank Guido van Rossum: the creator of Python. I've been programming since ANSI C was the standard and Turbo Pascal 5.0 was the new kid on the block, but in all those years I never came across a language that uses soo few lines of code to accomplish so much. As mentioned before, adding just a few libraries and you have a very powerful programming toolbox at hand.

Next to Guido, I'd like to thank all the programmers and contributors that share their code, their examples and answer the questions raised by co-programmers on platforms like Stack Overflow, Medium, GitHub and many others: without your explanations my life as a python programmer would have been a lot tougher.

Let me sign off with thanking IBM and Coursera for setting up this course. I was a bit reluctant at first having to complete nine courses to get a single certificate and that much of the reviewing and grading would be done by co students, but you guys have managed to setup an honourable community and the course materials where tough enough to make me proud that I have passed the test.

References

[1] Coursera, *IBM Data Science Professional Certificate.* More info.

[2] J. D. Walsh, *Ten Years On, Foursquare Is Now Checking In to You.* Article link.

[3] Promodomo, *Promodomo website.* Visit website.

[4] G. Bareirro, *Best Cuisines.* Github repo link.

[5] CBS, *Number of inhabitants in 2019 in each province of the Netherlands.* Source on StatLine.

[6] CBS, *Average house price in 2019 in each province of the Netherlands.* Source on StatLine.

[7] FourSquare, *FourSquare developers API - explore endpoint.* Docs on FourSquare.

[8] A. Zapara, *Mapping the COVID-19 outbreak in the Netherlands.* Medium link.

[9] M. Jellema, *GitHub repository.* Github repo link.

[10] GitHub, *GitHub Desktop Client OSX.* Github desktop client OSX.

[11] A. Yussuf, *What is the minimum value of correlation coefficient to prove the existence of the accepted relationship between scores of two of more tests?* Researchgate link.