Project name – Commute or Relocate... that is the question

Introduction - Project description

Problem definition

Johannesburg and Pretoria are situated in the heart of Gauteng, a province in South Africa. These two cities combined have the highest populations[1] and number of job opportunities [2]in South Africa. They are also relatively close to one another (+-50km apart centre to centre).

With the cities being spread over a large geographical area, many people tend to only operate in their familiar region of their own city that they know and trust. There are a few of misconceptions between these two cities caused by this reason and historic prejudices. The effect is that when a job opportunity comes along and is in a different area, or city, you do not relocate, but rather extend your daily commute.

Recently, there has been a warning from the government that this commute could increase considerably in the next few years, due to infrastructure development not being supported by the residents. (Please see: https://www.timeslive.co.za/news/south-africa/2018-11-12-get-ready-for-a-six-hour-commute-from-joburg-to-pretoria/)

Intended audience

The main idea with this project is to analyse the suburbs of Johannesburg and Pretoria, using their municipal regions, house sales information as well as FourSquare venue data to cluster similar neighbourhoods. With this information it would be possible for people moving between the cities, to start considering relocation to similar neighbourhoods, as an alternative to long daily commutes.

Scope

A problem with using clustering is that it is not clear why specific neighbourhoods would be similar, but with manual analysis it can be determined.

For the purpose of this analysis I will simply cluster the different neighbourhoods and display them on a physical map.

I will then conclude with some findings from the analysis.

---

[1] Johannesburg, Pretoria and Soweto are jointly considered in this analysis.
[Source: http://www.geonames.org/ZA/largest-cities-in-south-africa.html]

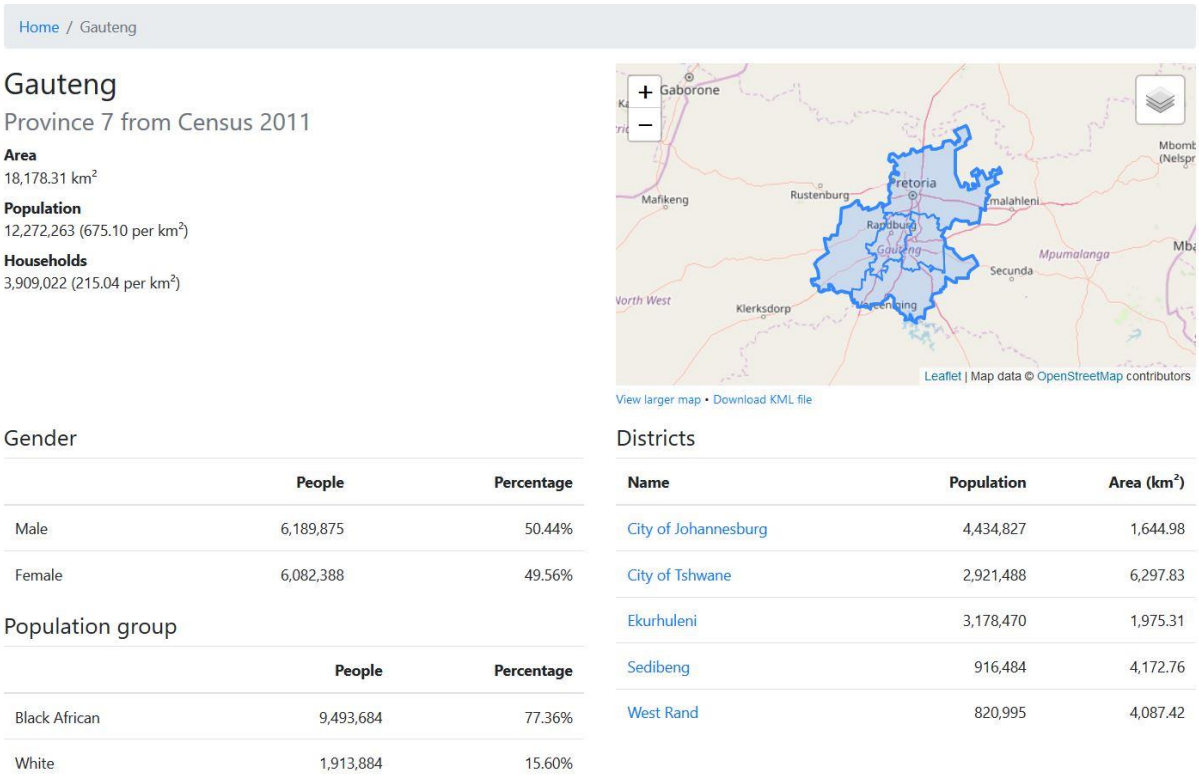
[2] Johannesburg and Pretoria are both considered in this analysis.
[Source: https://www.adzuna.co.za/blog/2019/01/28/here-are-the-best-and-worst-paying-destinations-in-south-africa/]

Data used

Johannesburg and Pretoria municipal regions

There was a census done in the year 2011, where the findings were quite well documented on the below link:

https://census2011.adrianfrith.com/place/7



Home / Gauteng

Gauteng

Province 7 from Census 2011

Area
18,178.31 km²

Population
12,272,263 (675.10 per km²)

Households
3,909,022 (215.04 per km²)

View larger map • Download KML file

Gender

	People	Percentage
Male	6,189,875	50.44%
Female	6,082,388	49.56%

Population group

	People	Percentage
Black African	9,493,684	77.36%
White	1,913,884	15.60%

Districts

Name	Population	Area (km²)
City of Johannesburg	4,434,827	1,644.98
City of Tshwane	2,921,488	6,297.83
Ekurhuleni	3,178,470	1,975.31
Sedibeng	916,484	4,172.76
West Rand	820,995	4,087.42

Figure 1: Census information for Gauteng provice (2011)

In *Figure 1* Johannesburg and Pretoria are listed as City of Johannesburg and City of Tshwane respectively, in the '*Districts*' table. From the link mentioned above, I only extracted the official city names, regions (constituents of each city) and suburbs (constituents of each region) information.

The combination of these three municipal levels, I used to get the specific latitude and longitudes for each suburb. I used the ***HERE geocoder API*** for this information.

House sales information

There are a few good online websites where houses are listed for rent or for sale. I have identified Private Property (see link below) as the website to be used for my data gathering as it has a clean design in the website layout:

https://www.privateproperty.co.za/

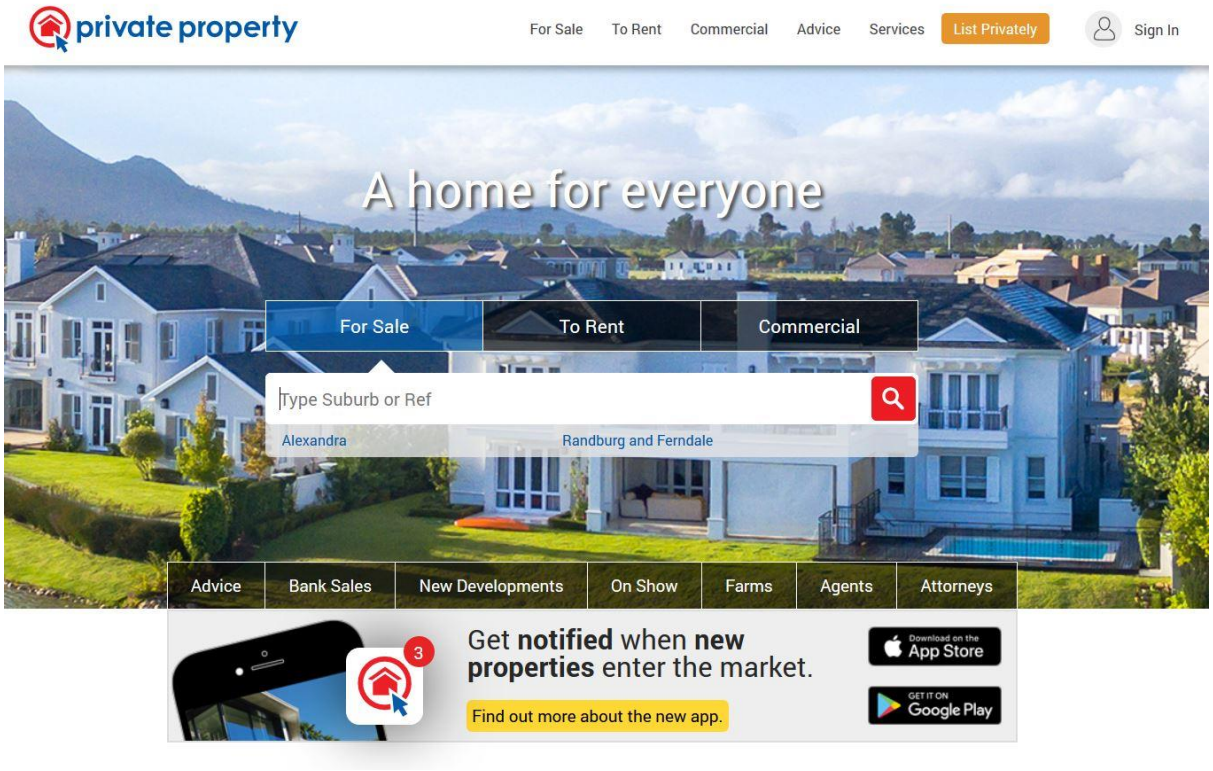


Figure 2: Properties for sale (rentals were not considered) were collected from this site

For the purpose of this project I only considered houses for sale. There is an abundance of data on this site and houses that are uploaded most recently are presented first. In order to level the playing fields between the sites, I sorted the listings from cheapest to most expensive. I only included the first 10 or less houses to be presented on the first page after being sorted (i.e. only consider the cheapest properties for sales in each suburb). The following metrics were extracted:

1. Property for sale prices
2. Number of bed rooms on the property

These two attributes are chosen to ensure the following:

1. The property being considered is for residential use (i.e. a house, townhouse, flat, etc.). The assumption is made that if there is a bedroom, the property would be for residential use.
2. This simulates the lowest barrier to entry for a relocating employee
3. Gives an indication of the size of the cheapest residential properties, and by extension if the area is suited for small or large families or for single employees

FourSquare data

The venue data for the surrounds will be retrieved from the FourSquare API. This data will give good insight into the facilities and businesses that are available in the nearby area for each suburb. I will be using the venue_category from FourSquare to get an indication of these different venues surrounding the suburb.

Methodology

Technologies used

I thought it appropriate to make a list of the technologies used in a central place and not scatter the information through the report. Also, this creates a clean break between ***how*** and ***why*** things were done.

Web scraping

1. ***Selenium*** – This package is used to interact with websites. Its main use is for unit testing for website developers and has a wide range of languages that it is capable of handling (e.g. Java, Javascript, Perl, Ruby, php and Python). I used this package for navigating through both links mentioned above, in the browser (i.e. clicking links and entering area names, etc.).
2. ***BeautifulSoup*** – BeautifulSoup is used to parse the embedded HTML tags. This creates a HTML object (analogues to a dictionary in python) in your development environment and you simply reference the correct element to extract the information.
3. ***Requests*** – Requests is the python package that allows you to interact with RESTful API's. I used requests to interact with both the HERE geocoder as well as FourSquare API.

Data processing

1. ***Pandas*** – this is a very powerful package used by many (if not all) data scientists in the industry. It allows you to create single row-column objects (i.e. dataframes), with many built in functionality to aid in processing data. Dataframes is a familiar construct for data scientist as it strongly resembles relational database tables. The added functionality makes it a pleasure to work with the data in a simplified and effective manner.
2. ***numpy*** – this package is also a very powerful data processing package, creating many of the functionality that is available in Pandas. Pandas and Numpy are standalone packages but are both synonymous to data processing and work in a symbiosis manner.
3. ***sklearn*** – this short for 'scientific kit – machine learning'. As the name suggest, this is a very powerful package where many (if not all) of the popular machine learning capabilities are built in. I specifically used the data pre-processing and the K-Means clustering capabilities in this project.

Data visualization

1. ***matplotlib*** – matplotlib is a very intuitive visualization tool that works well with pandas. There are many built in plots that you can create and all are very dynamic and adjustable in order to visualize data simply. I only used the colour scheme capability of this package to give a bit more meaning to the plots that I created.
2. ***folium*** – folium creates many different and beautiful looking maps for geocoded data. As I produced clusters, I simply plotted each suburb on the map to see how the different clusters are distributed over the area.

Data acquisition

When creating my data set for the clustering methodology, I followed the below process:

1. Created a base from the city, regions and suburbs data from the census site
2. Enhanced the data with the house sales data
3. Added latlong information for each suburb
4. Enhanced the above information with venue data

1. Creating the base

The base off of which the rest of the analysis works is the cities and their constituent regions and suburbs. If these are not based off genuine and correct data the rest of the analysis will not be usable. This is the reason why the Census 2011 (the most recent census to be held in South Africa) was used.

Having said this, there have been some assumptions that were made to use the data:

1. *The area names are spelled correctly* – when using the results in the house sales site I found that there are discrepancies in spelling some names between the sites (e.g. Census: Theresapark vs PrivateProperty: Theresa Park). I have assumed that the Census data has the correct spelling.
2. *All suburbs included under correct regions* – when inspecting results it was found that some suburbs had the same name between the two cities (e.g. Roodepoort). This is addressed when finalising the census data.

Upon investigation it was found that there were some regions that have the same name between the two cities, Pretoria and Johannesburg. I identified the two regions as:

	region	city
84	Rietfontein	2
86	Roodepoort	2

Figure 3: Regions with the same name between the two cities

I assigned the two regions to the following cities by applying the following logic:

1. **Roodepoort** -- This region is strictly a Johannesburg region, and technically is part of the Tshwane municipality as well, but is in a different smaller town (Bronkhorstspruit). I therefore assigned it to the City of Johannesburg.
2. **Rietfontein** -- This region is similar to Roodepoort, but flows the opposite logic, i.e. it is mainly referred to as a Tshwane suburb. The Rietfontein in Johannesburg, is in a different smaller town (Krugersdorp) although it is technically part of the greater Johannesburg. I therefore assigned it to the City of Tshwane.

With this I was confident that my base was solid and I was ready to progress to the next step.

2. Sales data

When considering the region and suburb data, I found that some regions were vastly larger than others. These bigger regions had a lot of constituent suburbs and similarly some of the smaller regions simply had a single suburb. Understanding this, I decided to do search on privatepoperty.co.za for both. The resulted in the following number of beds and price data:

```
City Shape: (59, 4)
Region Shape: (517, 4)
Property Shape: (253, 7)
```

Figure 4: Number of city regions (City Shape) and regional suburbs (Region shape) that returned bed and price data

This presented a problem as I had 117 city regions and 1374 regional suburbs originally, as shown below:

```
City Shape: (117, 2)
Region Shape: (1374, 2)
```

Figure 5: Original number of city regions (City shape) and regional suburbs (Regional shape)

This meant that from my original base I did not have sales data. I decided to only work with the regions that I had sales data. The reasoning behind this was simply that if South Africans did not have data for these regions (privateproperty.co.za), then it can be safe to assume that an international organisation would not have more. Therefore, there would not be any data for these regions to cluster on.

This decision shrank my workable base.

Another complication was that I had two levels of sales data: regional and suburban. I decided to apply the regional data as the average for all the suburbs that did not have sales data. The result from this was that a region would be inflated with the average which would skew the overall cluster creation. To solve for this I collapsed all suburbs that has the average into a single line item.

An example of this logic is shown below on the Akasia region in the City of Tshwane:

	city	region	suburb	beds	price
0	City of Tshwane	Akasia	Amandasig	1.50	382500.00
1	City of Tshwane	Akasia	Chantelle	1.81	561600.00
2	City of Tshwane	Akasia	Clarina	1.00	332454.00
3	City of Tshwane	Akasia	Heatherdale AH	5.66	4309000.00
4	City of Tshwane	Akasia	Akasia	3.00	1031727.00

Figure 6: Result of applying the collapsing logic on the Akasia region Pretoria

At the end of this stage the total number of suburbs, and therefore the number of line items totalled 556, and I was ready to move on to the following stage.

3. Overlaying of geocodes

In order to use the HERE geocoding API I needed to create a search string based off the city, region and suburb names. Below is an extract from the resulting dataset:

	city	region	suburb	beds	price	search_string	latlong
0	City of Tshwane	Akasia	Amandasig	1.50	382500	amandasig akasia pretoria south africa	(-25.67387, 28.10067)
1	City of Tshwane	Akasia	Chantelle	1.81	561600	chantelle akasia pretoria south africa	(-25.66581, 28.0923)
2	City of Tshwane	Akasia	Clarina	1.00	332454	clarina akasia pretoria south africa	(-25.64972, 28.11789)
3	City of Tshwane	Akasia	Heatherdale AH	5.66	4309000	heatherdale ah akasia pretoria south africa	(-25.66769, 28.1204)
4	City of Tshwane	Akasia	Akasia	3.00	1031727	akasia pretoria south africa	(-25.65923, 28.10318)

Figure 7: Dataset with search string and geocode response form HERE

I did a quick check to see if there were any duplicate geocodes between the suburbs. There was a total of 8 duplicate values, affecting 17 places (one pair appeared 3 times). I did not apply any logic or make any changes to these duplications and used the entries as is. This was my reasoning behind this:

1. *Geographical proximity* – the places with the same geocode pair were in any case very close to one another, and therefore would have very similar surrounding venues.
2. *Low number of occurrences* – on the complete set of 556 entries, there were only 9 non-unique geocode pairs

Below are a few of the duplicate locations:

	city	region	suburb	beds	price	search_string	latlong
955	City of Johannesburg	Lakeside	Lakeside	2.66	1659444	lakeside johannesburg south africa	(-26.10003, 28.14939)
821	City of Johannesburg	Johannesburg	Lakeside	2.66	1659444	lakeside johannesburg south africa	(-26.10003, 28.14939)
1227	City of Johannesburg	Sandton	Hurlingham Gardens	3.50	5425000	hurlingham gardens sandton johannesburg south ...	(-26.09095, 28.02361)
1226	City of Johannesburg	Sandton	Hurlingham	2.36	2341726	hurlingham sandton johannesburg south africa	(-26.09095, 28.02361)
1079	City of Johannesburg	Randburg	Johannesburg North	2.18	1190636	north randburg johannesburg south africa	(-26.0285, 27.97593)
1052	City of Johannesburg	Randburg	Bosmont	3.36	932727	bosmont randburg johannesburg south africa	(-26.0285, 27.97593)
1043	City of Johannesburg	Randburg	Randburg	1.00	320000	randburg johannesburg south africa	(-26.0285, 27.97593)
657	City of Johannesburg	Chartwell	Chartwell	4.18	3418181	chartwell johannesburg south africa	(-25.99462, 27.98145)
656	City of Johannesburg	Chartwell	Chartwell AH	6.00	3750000	ah chartwell johannesburg south africa	(-25.99462, 27.98145)

Figure 8: Duplicate geocode information extract

4. Venue data

The geocode data retrieved from the previous step was an input into this part of the project. As stated in the introduction, there is a wealth of data from FourSquare, but the only data of interest was the venue category surrounding the specified geocodes.

An example from the resulting dataset is shown below:

	city	region	suburb	beds	price	lat	lng	venue	venue_lat	venue_lng	venue_category
0	City of Tshwane	Akasia	Amandasig	1.50	382500	-25.67	28.10	None	nan	nan	None
1	City of Tshwane	Akasia	Chantelle	1.81	561600	-25.67	28.09	Wolmer	-25.67	28.09	Neighborhood
2	City of Tshwane	Akasia	Clarina	1.00	332454	-25.65	28.12	None	nan	nan	None
3	City of Tshwane	Akasia	Heatherdale AH	5.66	4309000	-25.67	28.12	Debonairs Pizza	-25.67	28.12	Pizza Place
4	City of Tshwane	Akasia	Akasia	3.00	1031727	-25.66	28.10	Nasi goreng jakarta	-25.66	28.10	Asian Restaurant

Figure 9: Venue data from FourSquare

Even from the above small extract it is evident that the venue data is sparse, but not non-existent, in the South African context. I decided to still keep the suburbs where there is no FourSquare data available, as this is also informative about the specific suburb.

This concluded the data acquisition phase of the project.

Cluster pre-processing

When using data in a clustering model, it is essential to ensure that it does not skew any results due to magnitude differences between variables. This is ***corrected via normalisation*** of the data, where all the variable is transformed to a number between 0 and 1. I followed the min-max-methodology and applied this to both the bed and price fields from the sales data.(For more info on the min-max-methodology please reference the sklearn documentation here).

	city	region	suburb	beds	price	beds_norm	price_norm
0	City of Tshwane	Akasia	Amandasig	1.50	382500	0.03	0.01
1	City of Tshwane	Akasia	Chantelle	1.81	561600	0.04	0.01
2	City of Tshwane	Akasia	Clarina	1.00	332454	0.00	0.01
3	City of Tshwane	Akasia	Heatherdale AH	5.66	4309000	0.25	0.15
4	City of Tshwane	Akasia	Akasia	3.00	1031727	0.11	0.03

Figure 10: Extract of normalised bed and price data (beds_norm and price_norm respectively)

Clusters can also not operate / compute based on categorical variables, such as the venue category data from FourSquare. These need to be ***converted into dummy variables***. There is a powerful capability in Pandas to generate such dummy fields, which is what I used. (For more info on this method please refer to Pandas documentation here.)

	city	region	suburb	Accessories Store	Afghan Restaurant	African Restaurant	Airport	Airport Service	Airport Terminal	American Restaurant	...
0	City of Tshwane	Akasia	Amandasig	0	0	0	0	0	0	0	...
1	City of Tshwane	Akasia	Chantelle	0	0	0	0	0	0	0	...
2	City of Tshwane	Akasia	Clarina	0	0	0	0	0	0	0	...
3	City of Tshwane	Akasia	Heatherdale AH	0	0	0	0	0	0	0	...
4	City of Tshwane	Akasia	Akasia	0	0	0	0	0	0	0	...

Figure 11: Extract of dummy indicators built off the venue category field

Having two datasets when data is visualized, can create complications. I decided to create a single dataframe which is specifically suited for the visualization of the information.

	key	city	region	suburb	beds	price	lat	lng	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	City of Johannesburg-Alexandra-Alexandra	City of Johannesburg	Alexandra	Alexandra	3.00	850000	-26.11	28.10	Afghan Restaurant	Zoo	Fast Food Restaurant	Fruit & Vegetable Store	Frozen Yogurt Shop
1	City of Johannesburg-Alexandra-East Bank	City of Johannesburg	Alexandra	East Bank	3.25	808750	-26.10	28.11	Soccer Field	Zoo	Furniture / Home Store	Frozen Yogurt Shop	Fried Chicken Joint
2	City of Johannesburg-Chartwell-Chartwell	City of Johannesburg	Chartwell	Chartwell	4.18	3418181	-25.99	27.98	Garden Center	Deli / Bodega	Coffee Shop	Farm	Café

Figure 12: Single dataframe for ease of reference when data is visualized

Finally, we can proceed to the clustering of the gathered data.

Clustering

There are a few different clustering methodologies. I decided on KMeans to do the clustering method.

A difficult decision, irrespective of methodology implemented, is how many clusters will be appropriate. It is also a highly subjective matter.

Given the wide geographical area and the very diverse populations being brought into play, I do believe there needs to be a larger than normal number of clusters (normal number of clusters ranging from 3 to 7). In order to make the result simple to display on a map and clear out confusion, it can't be too many either.

I have decided on 10.

*(**Note**: I did test on 20 and 15, however 10 seemed to produce the most clarifying map for visualization)*

Results

From the gathered data I produced two maps:

1. *Mapping of the regions* – this gave perspective on the municipal zoning of the regions and also the relative size of each region in relation to the other.
2. *Mapping of the clusters* – this is a great way of presenting the final clusters produced from the gathered data.

1. Mapping of regions

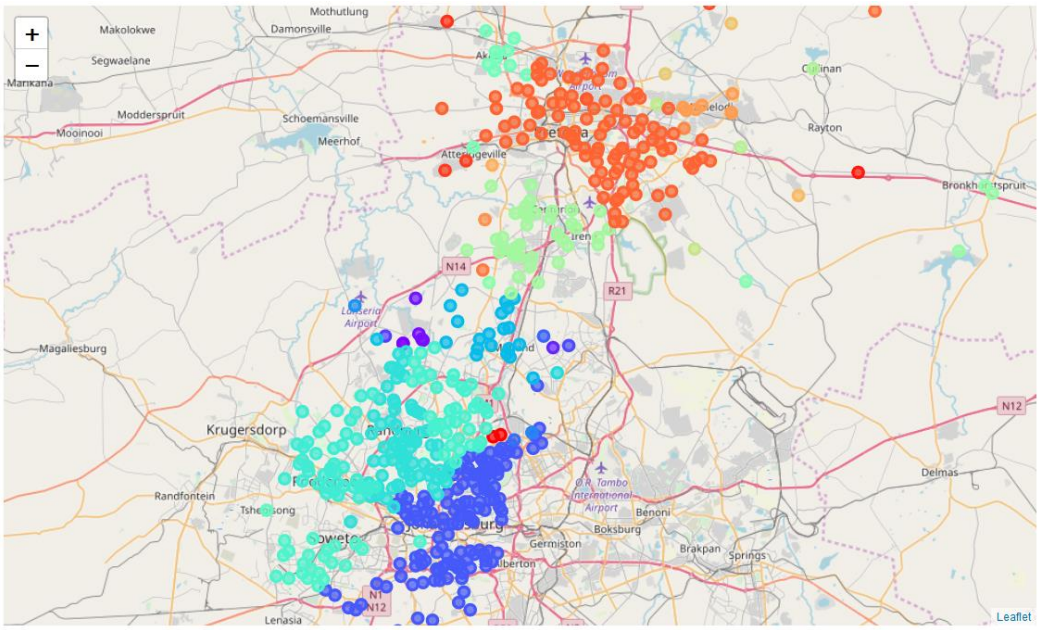


Figure 13: Map of city regions

2. Mapping of clusters

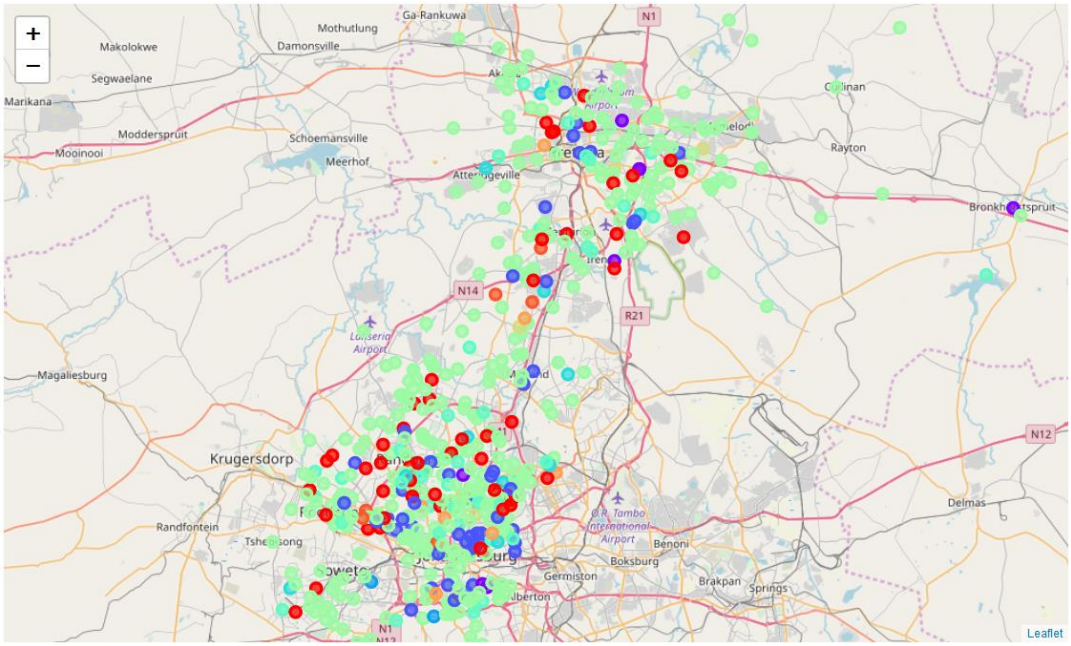


Figure 14: Map of created clusters

Findings

Finally mapping the dataset does do the effort justice. A quick few comments on the end result:

1. Overall there is a *strong similarity over the entire geographical area*. There is an evident majority cluster (cluster = 6), which does make sense as both of these cities are considered to be the heart of Gauteng.
2. When focus is turned closer, there does seem to be a modest difference in the homogeny between Pretoria and Johannesburg*. Johannesburg does seem to be a bit more metropolitan*, in the sense that it carries more diverse clusters over a smaller geographical area.
3. The space between the two cities (can be referred to as Midrand), also has a strong heterogenic appearance. This could be because some of the more *progressive communities have already moved to the centre*, which is closer to either centres.

Conclusion – Audience feedback

From the analysis performed, it appears that more care needs to be taken when considering a move from Pretoria to Johannesburg. This is true only if your objective is to relocate to a similar neighbourhood. The differences between the separate suburbs is more pronounced and happen at a much less gradual scale in Johannesburg when compared to Pretoria.

However, if the goal is to feel part of a greater society, where you feel that you are part of a melting pot of cultures, then moving from Pretoria to Johannesburg may be precisely what you are looking for. If you are considering a more stable and homogenous environment to a move from Johannesburg to Pretoria will cater to your needs.

Ultimately there are more similar areas than not between the two cities. It comes down to the choice of lifestyle that you are wanting to follow:

1. Stable and homogenous – Pretoria
2. Outgoing and divers – Johannesburg

From this analysis, all that remains to say is: Safe stay (when you are happy with your current environment), or happy hopping (when you are looking to still relocate).