Data Science Capstone Project: Analyzing Venues in Johannesburg Suburbs with Machine Learning

Brighton Nkomo

August 2020

1 Introduction

Johannesburg, informally known as Jozi, Joburg, or "The City of Gold", is the largest city in South Africa and one of the 50 largest urban areas in the world [2]. It is the provincial capital and largest city of Gauteng, which is the wealthiest province in South Africa. Johannesburg is the seat of the Constitutional Court, the highest court in South Africa. The city is located in the mineral-rich Witwatersrand range of hills and is the centre of large-scale gold and diamond trade. It was one of the host cities of the official tournament of the 2010 FIFA World Cup.

In this project, I analyzed different kinds of venues using the power of k-means clustering to seek the hidden patterns about the most visited venues in each of the suburbs within the City of Johannesburg municipality.

1.1 Business Problem

Suppose that there is a contractor trying to open a restaurant within the Johannesburg municipality, how can we use the current machine learning techniques to determine the suitable locations? To begin answering the question, it is reasonable to ask what makes a good location to situate a restaurant at? These are some of the key features to consider when opening a new restaurant:

- Visibility: Urban areas tend to have high car and foot traffic. Locating a restaurant around towns would hence be a good choice. However, it could be possible to find places that offer high visibility for the restaurant, but also have high crime rates. Such areas are not best suited for family-style restaurant.
- Parking: Places near parking lots would be another good choice. It would be ideal to have a restaurant with it's own parking lot.

Out[10]:

	Province	District	Local_municipality	Suburb	Metro	Latitude	Longitude	Main place
0	Gauteng	Sedibeng	Midvaal	Brenkondown	Johannesburg	-26.343133	28.073783	Alberton
1	Gauteng	Sedibeng	Lesedi	Masetjhaba View	Johannesburg	-26.388533	28.384250	Duduza
2	Gauteng	Sedibeng	Lesedi	Sonstraal AH	Johannesburg	-26.406613	28.361255	Sonstraal
3	Gauteng	West Rand	Mogale City	Ruimsig Noord	Johannesburg	-26.075359	27.865240	Krugersdorp
4	Gauteng	Ekurhuleni	Ekurhuleni	Germiston Ext 3	Johannesburg	-26.214897	28.181906	Germiston

Figure 1: The Relevant Features

- Accessibility: It could be beneficial to have a restaurant built across a road with a relatively low speed limit and high car traffic. Supposedly around freeway/highway exits. As for foot traffic, a location near urbanized areas would be ideal. Inside shopping malls, an ideal place to have a restaurant would be within or near food courts.

There are many other factors to also consider such as average income and the population of the area of interest [7, 8, 10]. However, the goal of this project is to find out how urbanized an area is by finding out the most popular venues within that area and to seek out hidden patterns that may reveal some additional information about a location.

1.2 The Data Set

For this project, the location data was a geojson file taken from CartoDB [1], which is the world's leading location intelligence platform. After downloading the geojson file from this website and loading the data set in a jupyter notebook, the data set contains information we need such as name of the province, suburb (also known as a neighborhood in Commonwealth countries), main place, local municipality (also known as a borough in some English speaking countries) latitude and longitude coordinates of the locations. There is also other information such as the population of black people, colored people (a term referring to people of mixed race in South Africa) and white people. Although this demographic data may be relevant when it comes to picking out which locations have the people with the highest average income and locations that may offer high foot traffic, however I did feature selection and decided to drop the population data. The table in Figure 1 shows the relevant features after feature selection.

With this location data, I then used the Foursquare API which is 'a local search-and-discovery mobile app developed by Foursquare Labs Inc. The app provides personalized recommendations of places to go near a user's current location based on users' previous browsing history and check-in history.' So basically this app can be used for location detection. As explained from the Foursquare Wikipedia page 'When users opt in to always-on location sharing, Pilgrim determines a user's current location by comparing historical check-in data with the user's current GPS signal, cell tower triangulation, cellular signal strength and surrounding WiFi signals.'

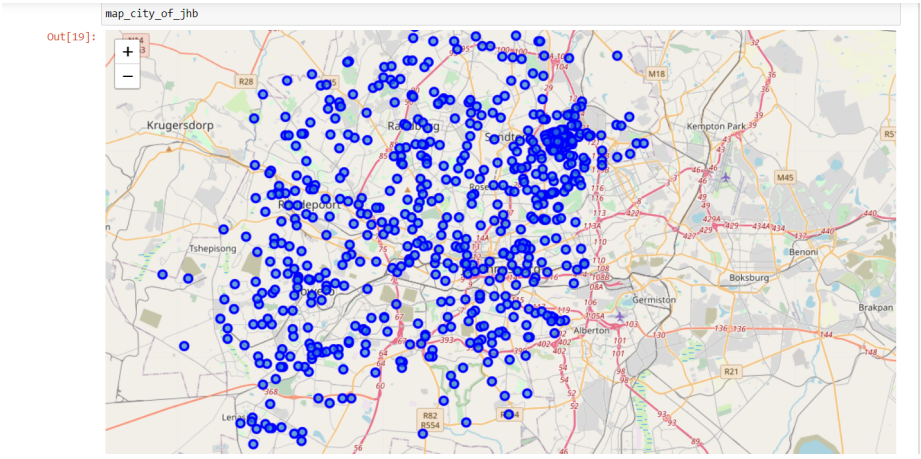


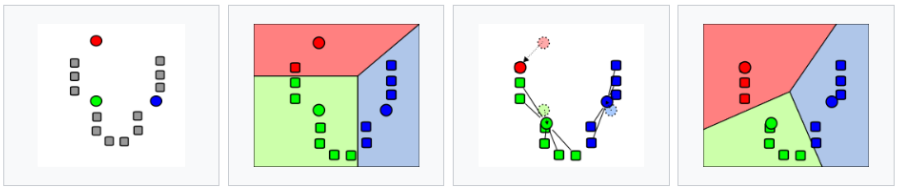
Figure 2: The Locations of the Suburbs within the City of Johannesburg Municipality

So Foursquare uses one's location information and visit frequency to "learn" what the user likes, which aims to improve user-facing recommendations and gauge the popularity of a venue. With the location data I had to make calls to the Foursquare API to find the most common venues per suburb in the Johannesburg, by constructing a URL to send a request to the API to search for a specific type of venues and to explore a geographical location. Also, prior to this I used the visualization library, Folium, to visualize the Suburbs in Johannesburg and find out how big the size of the data set was. In the City of Johannesburg municipality, there were 659 Suburbs as shown on the map in Figure 2 (each of the blue points are a suburb within the local municipality).

Since I was analyzing the most common venues within 500 meters per suburb, I decided to limit the number of the most common venues to just 100 venues.

2 Methodology

Since there are no labels in the data set for this particular problem, unsupervised learning is best suited to solve this problem. In particular clustering algorithms such as k-means clustering are good candidates for dealing with location data. The k-means clustering algorithm creates clusters automatically and takes the mean values of the to determine the cluster centers.



1. *k* initial "means" (in this case *k*=3) are randomly generated within the data domain (shown in color).

2. *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.

3. The centroid of each of the *k* clusters becomes the new mean.

4. Steps 2 and 3 are repeated until convergence has been reached.

Figure 3: Demonstration of the Standard Algorithm

2.1 k-Means

'The k-means algorithm is used to partition a given set of observations into a predefined amount of k clusters.' The algorithm begins with randomly generated set of k centroids or cluster center (μ). The algorithm update step assigns all the observation x to their closest cluster center (see eq. 1). 'In the standard algorithm, only one assignment to one center is possible. If multiple centers have the same distance to the observation, a random one would be chosen.'

$$S_i^{(t)} = \left\{ x_p : \left\| x_p - \mu_i^{(t)} \right\|^2 \leq \left\| x_p - \mu_j^{(t)} \right\|^2 \ \forall j, 1 \leq j \leq k \right\} \tag{1}$$

The cluster centers would then be moved to a new position by calculating the mean of the assigned observations with respect to their cluster centers (eq. 2).

$$\mu_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j \tag{2}$$

This updating process carries on until there are clustering centers that can be shifted/re-positioned and when all observations cannot be assigned to a new cluster center. This is best illustrated in the figure 3 that I took from wikipedia

This means that the k-means algorithm tries to optimize the *objective function* (eq. 3).
Since there's only a finite number of possible assignments for the finite number of cluster centers and observations and also that each iteration is an improved solution, it is guaranteed that the algorithm will stop in a *local minimum*.

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||x_n - \mu_k||^2 \tag{3}$$

$$\text{with } r_{nk} = \begin{cases} 1 & x_n \in S_k \\ 0 & \text{otherwise} \end{cases}$$

The main disadvantage of k-means is its dependency on the initial randomly chosen cluster centers. The cluster centers could end up splitting the the clusters whilst observations that clearly belong to other clusters get grouped together especially when some of the cluster centers are attracted by outliers. In other words, the k-means is not robust to outliers.
The common solution to this problem is to have multiple clusterings with different starting positions. Then after, the most frequently occured clustering is considered as correct. The textbfk-means++ algorithm is also another solution to solving this issue which was proposed by Arthur and Vassilvitskii [3]. The k-means++ tries to distribute the initial chosen cluster centers over the given data so as to minimize the probability of bad clustering outcomes. According to Arthur and Vassilvitskii, the initial cluster centers are set as follows:

1. Take uniformly a random data point from the data X and mark it as centroid c_1
2. Choose another centroid c_i with the probability $\frac{D(x)^2}{\sum_{x \in X} D(x)^2}$ where $D(x)$ denotes the shortest distance from the data point x to its closest, already chosen centroid.
3. Repeat 2. until all k initial centroids are chosen.

After these steps, the standard k-means algorithm as described from the beginning of this section is performed. The authors also showed that with this initialization algorithm, k-means++ approximately can be computed in $O(\log n)$, compared to $O(n^{dk+1} \log n)$ for the standard algorithm. Therefore there k-means++ algorithm is faster that the standard algorithm.

There are detailed papers that give good reviews of k-means clustering [9] while they also provide discussions of the main challenges that clustering algorithms present [4, 11] and where there is some emerging and useful research directions (this includes ensemble clustering, semi-supervised clustering[6], simultaneous feature selection during data clustering and large scale data clustering).

2.2 Data Preparation

- **data cleaning**: In this project it was necessary to clean the data by deleting rows with missing values. If the latitude and longitude coordinates are missing, then the Foursquare API would not be able to retrieve any information about the unknown locations. Perhaps taking the mean of the longitude and latitude coordinates would have also worked because the coordinates do not differ by a large order of magnitude. In fact, the

latitude and longitude coordinates differ by an order of magnitude at most -1. No outliers were removed from the data, because this would have been very tricky to do for all 659 suburbs.

- **feature selection**: Attributes such as the population of different races, total population within a suburb and names of district municipalities were dropped. In general, redundant features were eliminated.

- **feature scaling**: Since the range of values of raw data varies widely, in some machine learning algorithms, initially the objective functions did not work properly without normalization. This was because the k-means standard algorithm uses the Euclidean distance to calculate the distance between two points. If one of the features has a broad range of values, the distance will be governed by this particular feature. Therefore, the range of all features were normalized so that each feature contributes approximately proportionately to the final distance.

3 Results

One way to find the optimal number of clusters is to use the **elbow** method where the mean squared distance between each instance/observation, also know as *inertia*, is plotted against the number of clusters [5, p. 245–258]. Since the elbow method graph did not reveal any clear elbow to show the optimal number of cluster centers k to use as shown in Fig. 4, the alternative approach of plotting the silhouette score vs a range of k centroids was used and the results are depicted in Fig. 5. Note that the range of k is from 2 to 50, because 2 clusters are needed to compute the silhouette score.

Looking at Fig.5, there are four significant peaks, at $k = 2$, at $k = 6$, $k = 11$ and $k = 14$. Choosing $k = 2$ would result in a loss of information (under-segmenting) and choosing $k = 11$ or 14 may result in a loss of interpretability (over-segmenting). However, for this business problem we may want to over-segment, so $k = 11$ would have been a good choice of clusters, but it is has a bad silhouette score when compared to $k = 6$. Therefore, for these reasons, $k = 6$ was use to train the model and also the $k-$means++ initialization algorithm was used for the reasons stated in section 2.1. Afterwards, the clusters were then visualized on a map in Fig.6. Each color is used to distinguish each of the six clusters (cluster 0 to cluster 5).

Table 1 is a list of the most frequently appearing venues and main places for each cluster and the numbers in the brackets are the modes of the respective venue an main place. The Jupyter notebook contains a list of all results. However, the list most significant results are in the '1st Most Common Venue' and '2nd Most Common Venue' columns. Main places contain the suburbs that we may want to look at. In other words, main places are a compact way of grouping suburbs. Also to note, Johannesburg is both a city and a main place.

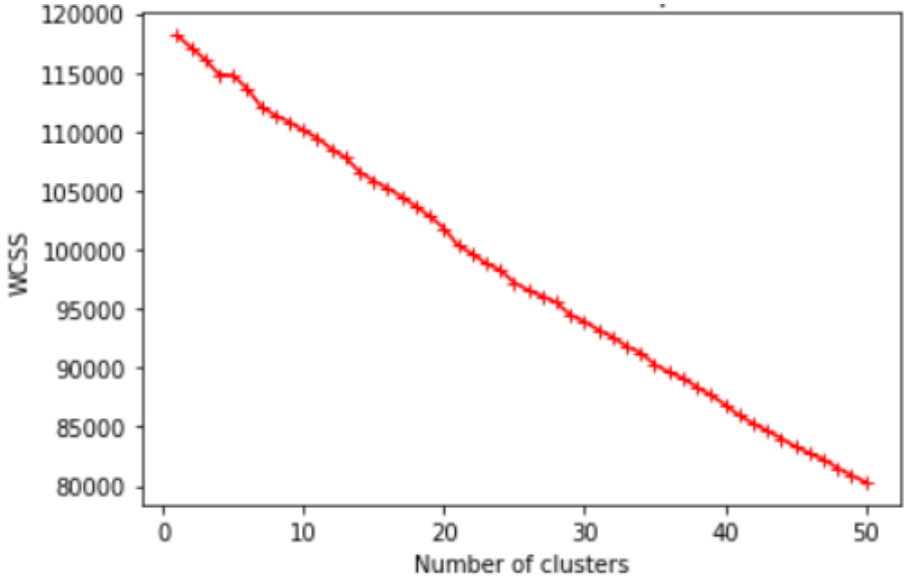


Figure 4: The Elbow Method Graph

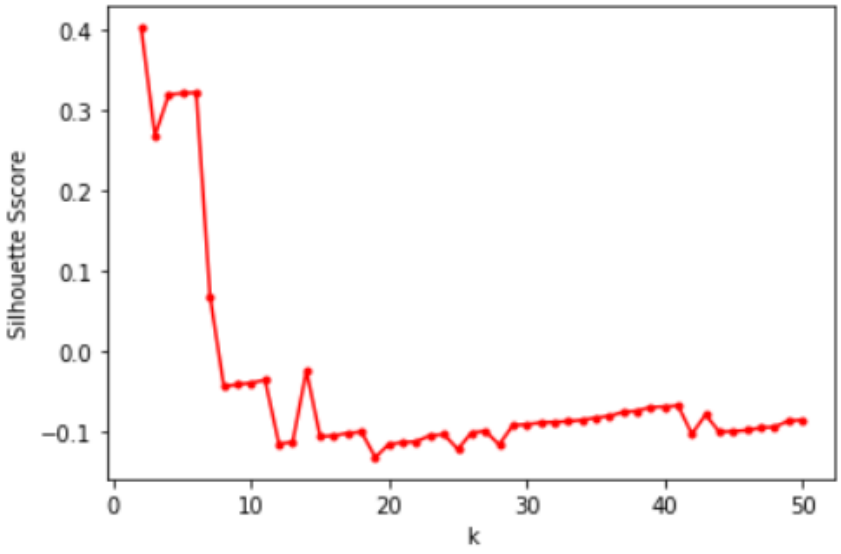


Figure 5: The Silhouette Score vs the Number of Cluster Centers

Out[58]:

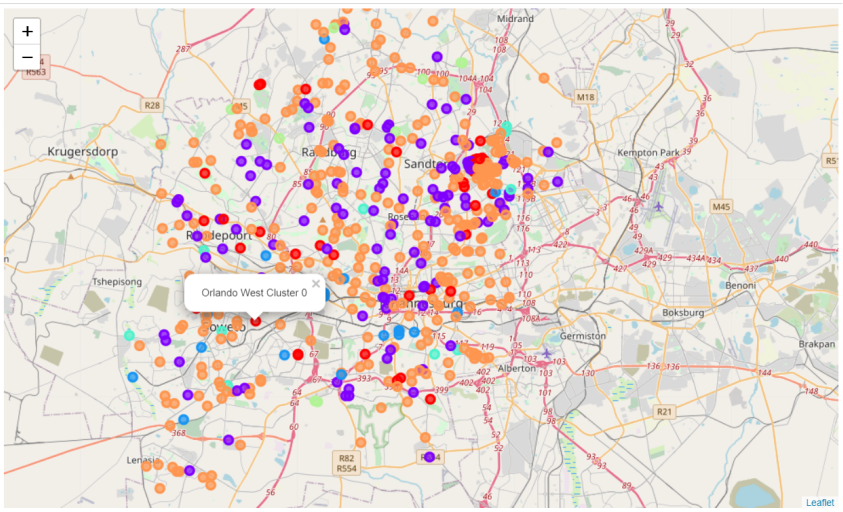


Figure 6: Clusters on the Map

Summary of Results			
Cluster	**1st Most Common Venue**	**2nd Most Common Venue**	**3rd Most Common Venue**
0	Fast Food Restaurant (21)	Fast Food Restaurant (5)	Johannesburg(12)
1	Grocery Store (14)	Yoga Studio (13)	Johannesburg(45)
2	Construction and Landscaping(13)	Construction and Landscaping(10)	Johannesburg (12)
3	Gas Station (11)	Yoga Studio (5)	Soweto(3), Johannesburg(3)
4	Restaurant (9)	Yoga Studio (6)	Sandton (3)
5	African Restaurant(13) Coffee Shop (13)	Yoga Studio (72)	Johannesburg (111)

Table 1: A summary of results. See **Section 5: Analysing the Clusters** in the Jupyter notebook for full results.

4 Discussion

As we can see, cluster 0 is dominated by fast foot restaurants. This seems plausible because there are a lot of such restaurants in the main place location. This cluster suggests that there is a lot of foot traffic for the location belonging to this cluster. Although there is competition, this cluster still offers good locations for opening a restaurant. As an example, food courts in shopping malls have competing restaurants next to each other, but due to the foot traffic, the restaurants are able to survive. Actually since people dislike long queues, there is a good chance that people will go to the next-door less occupied restaurant once the well known one becomes too full or has queues that are too long.

Cluster 1 is dominated by the grocery store and yoga studio venue. Also, the third most common venue is a shopping mall. This would be a good cluster for someone who wants to open a restaurant within a shopping mall that is within the City of Johannesburg Municipality. Yoga studios generally do not occupy many people compared to the grocery store and shopping mall, but since yoga classes in South Africa are somewhat expensive, the yoga studio locations suggest that those locations have high income.

Cluster 2 is largely dominated by the construction and landscaping venues. This seems plausible since there are still some underdeveloped locations called "townships" in some locations of Johannesburg. Some of these locations tend to be closer to wealthy suburbs and such an example is the township called Alexandra which is very close to the Sandton main place suburbs. This possible explains why the third most common venue in this cluster was yoga studio.

Cluster 3 is dominated by the gas station locations. This is,in general, not the best cluster for opening a restaurant although it a suggests a lot of car traffic. In locations such as Soweto, most people rely on public transport for transportation. Minibuses which tend to occupy 15 people and buses which can occupy 60+ people. However due to the high car traffic in this cluster, it may perhaps be great for advertising the restaurant with billboards. However, the restaurant could be built closer to the yoga studios than the gas stations.

Cluster 4 and 5 are somewhat identical to each other. Unlike cluster 1 and 3, the most common venues are restaurants, not a grocery store or gas station. Again, having a restaurant next to competitors is not necessarily a bad idea. Thus cluster 4 and 5 still make good location for opening a restaurant. However, in cluster 4 people like going to restaurants in general while in cluster 5 people like African restaurants.

5 Conclusion

One might rename the six clusters as follows:

1. Cluster 0 - Fast Food Restaurants
2. Cluster 1 - Grocery Stores and/or Yoga Studios
3. Cluster 2 - Construction and Landscaping
4. Cluster 3 - Gas Stations
5. Cluster 4 - Restaurants
6. Cluster 5 - African Restaurants, Coffee Shops and/or Yoga Studios

Cluster 0 , 4 and 5 make great locations for opening a restaurant. On the map, these are suburbs highlighted in red, green and orange respectively. These locations on the map would then need to be checked for the visibility and the accessibility of the restaurant when it's built. Furthermore, since locations near the highway exits can make good locations for a new restaurant, such locations can now be found on the map.

Also, although there is no cluster clearly dominated by football stadiums, Looking at the maximum capacity of the stadiums and the fact that concerts can also take place, it seems like a good idea to have a restaurant either advertised or built not faraway a major stadium. Due to the foot and car traffic that may occur on the weekdays and special occasions.

6 Future Work

It would be interesting to see how many and what kind of clusters the DBSCAN algorithm will produce, given the right values of the hyper-parameters. The DBSCAN algorithm is also well suited for being used on location data and it does not require us to guess the number of clustering centers k beforehand. This project may also be altered to solve other types of business problems such as where in Johannesburg to open an office. However, I believe that a different dataset may have to be used to determine a suitable office location. Specifically, using the speed profiles, traffic density and statistics data sets for the City of Johannesburg municipality may help solve the problem this business problem.

References

[1] Carto.com.

[2] Principal agglomerations of the world.

[3] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

[4] Tobias Brunsch and Heiko Röglin. A bad instance for k-means++. *Theoretical Computer Science*, 505:19–26, 2013.

[5] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.* O'Reilly Media, 2019.

[6] Nizar Grira, Michel Crucianu, and Nozha Boujemaa. Unsupervised and semi-supervised clustering: a brief survey. *A review of machine learning techniques for processing multimedia content*, 1:9–16, 2004.

[7] Haden Jeff. How to find the perfect location for a new restaurant: 6 ways.

[8] Haden Jeff. What makes for a great restaurant location?

[9] Trupti M Kodinariya and Prashant R Makwana. Review on determining number of cluster in k-means clustering. *International Journal*, 1(6):90–95, 2013.

[10] Mealey Lorri. 4 important factors when choosing a location to open a restaurant.

[11] Michael Steinbach, Levent Ertöz, and Vipin Kumar. The challenges of clustering high dimensional data. In *New directions in statistical physics*, pages 273–309. Springer, 2004.