Johannesburg Urban Tourism Data Analysis Report

Morrie Yu
2020.4.27
Johannesburg, South Africa

Introduction

In 2017, the Gauteng province in South Africa experienced a sizable drop in tourism revenue (Gauteng Tourism Authority, 2019). It was down by 16.6% than the previous year (Gauteng Tourism Authority, 2019). A preliminary research showed that visitors of Gauteng mainly came to visit friends and families (Gauteng Tourism Authority, 2019). According to this visitors' profiling, and the fact that there are two major urban areas in the province, it is sensible to focus on the in-city entertainment to turn the table.

Johannesburg has great potential in developing urban tourism. Not only it is the capital of the Gauteng province, the most populated city in South Africa, it is also one of the metropolises in Africa. It offers a broad and heterogeneous range of cultural, architectural, technological, social and natural experiences and products for leisure and business (UNWTO).

It is hard to develop urban tourism without knowing where the city stands now. So, what is the city offering now? And how can tourists find these spots?

The Gauteng Tourism Authority might benefit from a city tourism spot database and recommendation tool to provide tailor-made information services to visitors. Accurate and non-biased travel information can save the cost of travelers (Wei, 2012). A well-tested tourism spot database can provide comprehensive information inquiry services for tourists, whether they want to better understand the destination, or to pinpoint a desired venue, or to design a route.

There is a natural bond between geographic data analysis and tourism, for tourism has a strong geographical attribute (Wei, 2012). Functions of data collection, processing, spatial analysis are particularly relational to the tourism industry (Wei, 2012). It can play a role in tourism information management and thematic mapping (Wei, 2012).

Tourism has become increasingly popular as people's standard of living continues to improve. Africa has attracted a fair amount of attention from travelers all over the world. Investing in developing a tourism geographic information system is a step toward attracting more visitors, and encouraging longer stays and more spending.

This research aims to utilise Foursquare venue database to provide non-biased tourism knowledges on Johannesburg. The first step this research took was to produce a tourism spot database inside the city, and then by querying the database, this research sketched up an urban tourism landscape of the city, such as a list of top 20 most diverse suburbs, and outdoor activities. Lastly, this research clustered the suburbs in the city according to commonality of venues' categories. The first step provided a database, and the combination of step two and three provided an easy-to-use recommendation tool for tourists, such as where to stay and where to go based on their interests.

Data description

This research utilised Foursquare venue database to provide non-biased tourism advices on Johannesburg for visitors. Firstly, this research collected all the suburbs names in the city from the following websites:

http://www.sapostalcodes.info/queryPostal/Johannesburg

Then geocoding was used to covert the suburbs into geographic coordinates (like latitude and longitude) and added to the suburbs dataframe. By now, there is a dataframe which contains suburb name, latitude and longitude information.

Using the coordinates as location attributes, this research then queried Foursquare venue database to collect venue information including venue name and venue category. This concluded the data collection. In the end the dataframe had columns of suburb name, latitude, longitude, venue, and venue categories, and was ready for analysis.

Methodology

The methodology discussion was divided into three parts. First part was the methodology used for getting the dataframe ready. Second part was the various dataframe querying attempts used in data analysis. Last part was the machine learning algorithms in clustering.

Getting the dataframe

The values of the dataframe JHB-venues consisted of suburb names, coordinates and venues. As introduced in the Data description section, the dataframe started with getting suburbs in the city. BeautifulSoup library was used to scrape for suburbs names from a website. The website had two ready-made tables, and one of them had suburb names, cities, streets and post boxes information. It was easy to covert the table into a dataframe.

Before geocoding, a decision was made to remove suburbs which contained “ext” and “uit” in their names. Because the geocoding was not sensitive enough to differentiate between the extensions nor the units of the same suburb. For example, Alex ext 1 and Alex ext 2 yielded same coordinates.

Nominatim was used for geocoding. Nominatim is a search engine for OpenStreetMap data. A Nominatim function was created and then looped through suburb names. The result of Nominatim was appended to the list of suburbs. Then the list was converted into a dataframe which contained suburb names, latitudes and longitudes. These suburbs were then mapped by Folium.

Parameters of latitudes and longitudes were then passed to locate the suburbs when making the Foursquare “explore” requests. A random suburb was chosen to test the API and then a function was created to loop through all the suburbs to get venue and venue category information. Venue name and venue category were chosen because these could reflect the amount of venues and the diversity of the venues for analysis in the next step. The radius was set at 1000 meters because there were places where a radius of 500 meters was not wide enough to find anything.

Querying the dataframe

The dataframe of JHB-venue was then grouped and counted by venue and unique venue categories to show the busiest and most diverse suburbs. A few querying methods were used as shown here:

df_D=JHB_venues

```
m=df_D.groupby('Suburb')['Venue Category'].nunique()

df_B=JHB_venues.groupby(by='Suburb').agg('count').sort_values(by=["Venue"],ascending=False)

JHB_VC=JHB_venues.groupby(["Venue
Category"]).Suburb.agg("count").to_frame("Counts").reset_index()

JHB_VC1=JHB_venues.groupby(["Venue"]).Suburb.agg("count").to_frame("Counts").reset_index()
```

Bar charts from matplotlib was used to show the results of the querying as shown here:

```
df_D=JHB_venues
m=df_D.groupby('Suburb')['Venue Category'].nunique()
n=pd.DataFrame(m)
o=n.sort_values(by=['Venue Category'],ascending = False).head(20)
p=o.sort_values(by=['Venue Category'],ascending =True)
p.plot(kind="barh",figsize=(10,6))
plt.title("Top 20 most diverse suburbs in Johannesburg")
plt.xlabel("Count of places'categories in each suburb")
plt.ylabel("Suburbs")
plt.show()
```

Folium was used show the results of the querying as shown here:

```
map_jhb_busy = folium.Map(location=[latitude, longitude], zoom_start=11)


for lat, lng, suburb in zip(jhb_busy['Suburb Latitude'], jhb_busy['Suburb Longitude'],
jhb_busy['Suburb']):
    label = '{}'.format(suburb)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='red',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_jhb_busy)

for lat, lng, suburb in zip(jhb_notbusy['Suburb Latitude'], jhb_notbusy['Suburb Longitude'],
jhb_notbusy['Suburb']):
    label = '{}'.format(suburb)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=1,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
```

```
        parse_html=False).add_to(map_jhb_busy)

map_jhb_busy
```

Fuzzywuzzy library was used for string matching to create an outdoor list to showcase the outdoor activities available in Johannesburg. The keywords could be “Restaurant”, “Shop” or any other category of interests. The outdoor was an example to show the capability of Fuzzywuzzy. The codes were:

```
def get_ratio(row):
    name = row['Venue Category']
    return fuzz.partial_ratio(name, "Park")

Park = df_data_1[df_data_1.apply(get_ratio, axis=1) > 90]
Park.shape

def get_ratio(row):
    name = row['Venue Category']
    return fuzz.partial_ratio(name, "Golf")

Golf = df_data_1[df_data_1.apply(get_ratio, axis=1) > 60]
Golf.shape

def get_ratio(row):
    name = row['Venue Category']
    return fuzz.partial_ratio(name, "Trail Zoo")

Trail = df_data_1[df_data_1.apply(get_ratio, axis=1) > 80]
Trail.shape

Outdoor=[Park, Golf, Trail]
Outdoor_result = pd.concat(Outdoor)
Outdoor_result
```

DBSCAN clustering

Density-Based Spatial Clustering of Applications with Noise was used to cluster the suburbs. core samples of high density and expands clusters from them. The eps was chosen to be 0.35 and the min samples were 4. Experiments were made to determine eps and min samples. These experiments always came to a result of 14 clusters with one noise list.

The codes were as shown here:

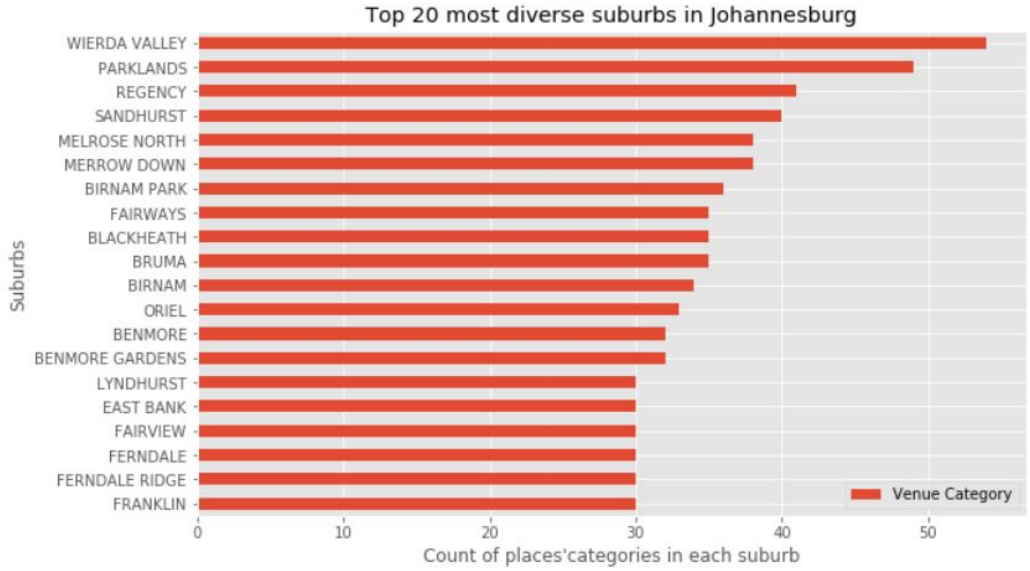
```
from sklearn.cluster import DBSCAN
db = DBSCAN(eps=0.35, min_samples=4).fit(JHB_grouped_clustering)
core_samples_mask = np.zeros_like(db.labels_, dtype=bool)
core_samples_mask[db.core_sample_indices_] = True
labels = db.labels_
JHB_grouped_clustering["Clus_Db"]=labels

realClusterNum=len(set(labels)) - (1 if -1 in labels else 0)
clusterNum = len(set(labels))
set(labels)
```

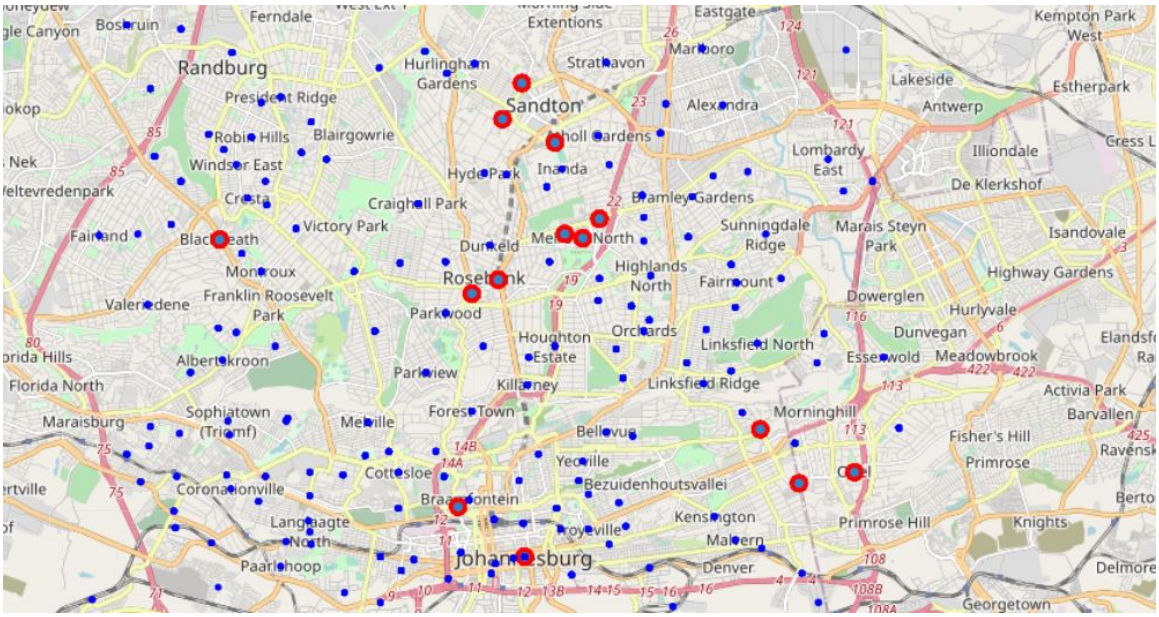
Results

The research found 1562 suburbs in Johannesburg, 519 geocoding-recognisable suburbs, 318 pairs of coordinates, 5024 venues and 218 unique venue categories.

The top 20 most diverse suburbs were shown in the following bar chart:

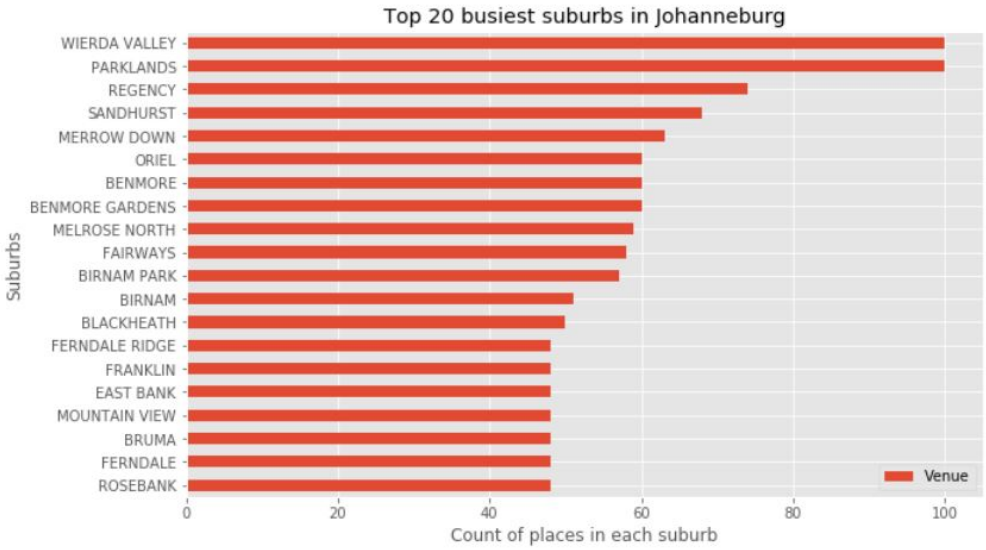


They were marked as red dots in the map:

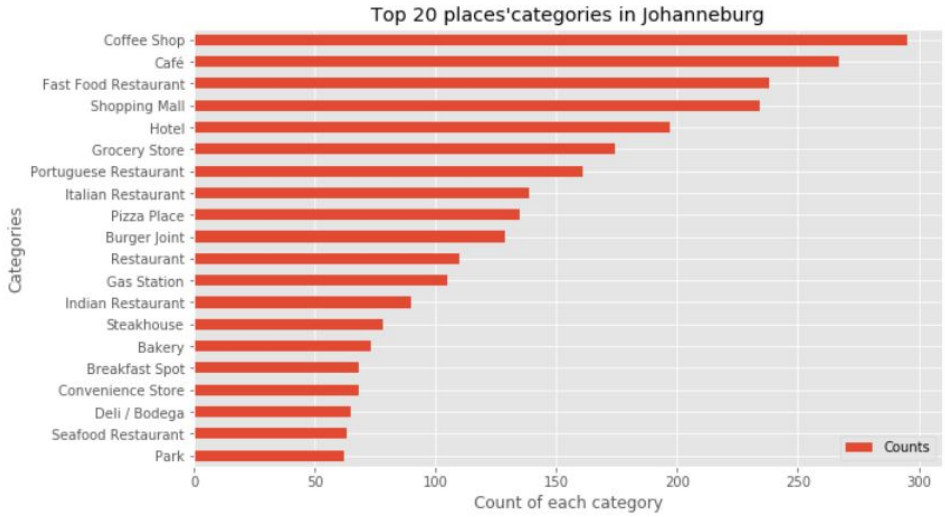


The top 20 busiest suburbs were shown in the following list:

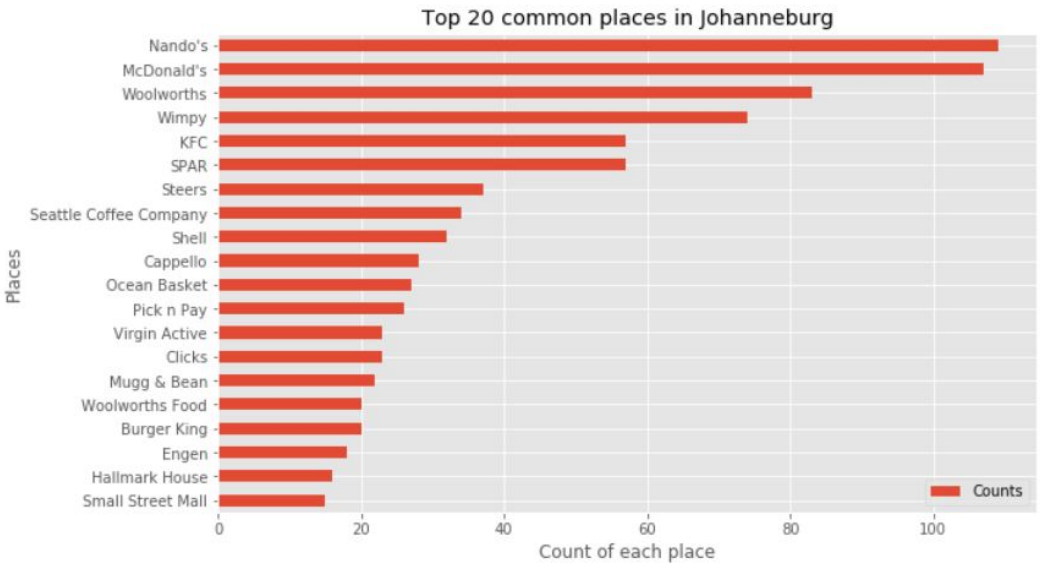
I



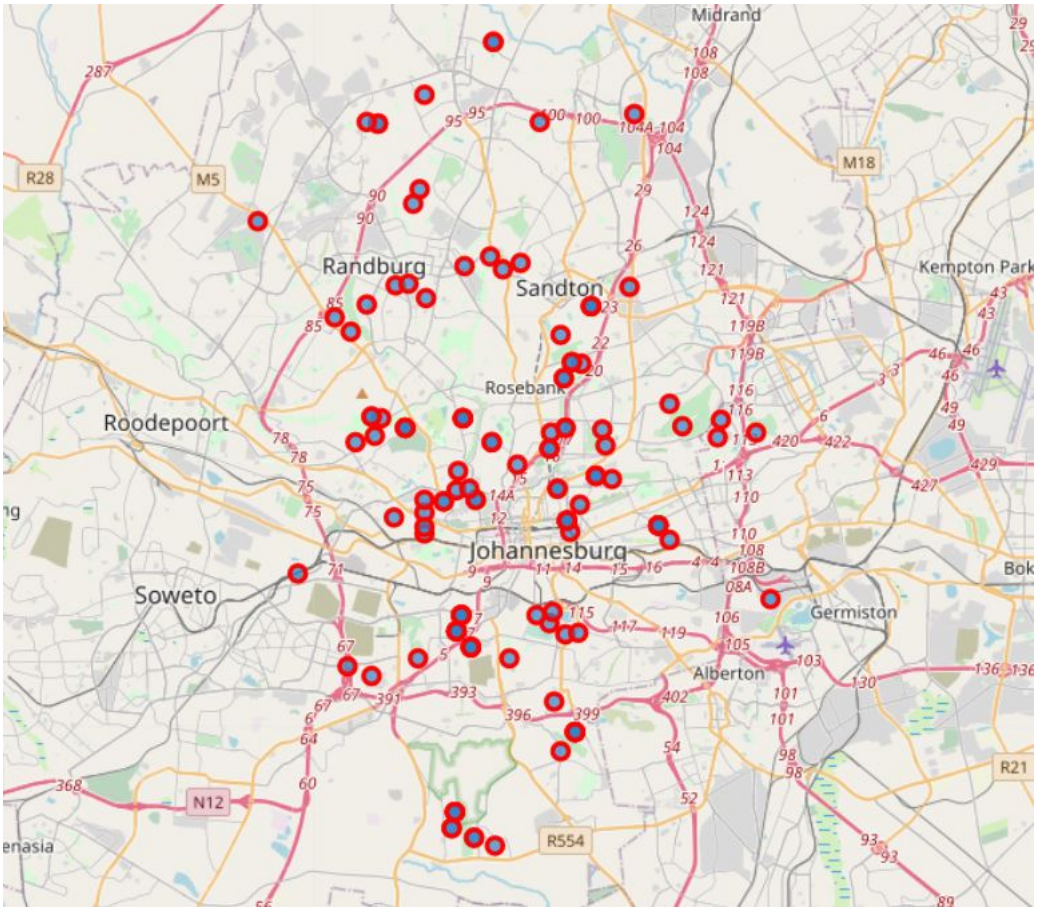
The top 20 most popular venue categories were:



The top 20 most popular venues were:



The in city outdoor activities were located as shown here:



DBSCAN clustering yielded 15 clusters.

Clusters

Cluster labels	Feature
Cluster -1 Noises	Noises
Cluster 0	Common everyday areas, with coffee shops, restaurants, and malls
Cluster 1	Trails
Cluster 2	Bellevue
Cluster 3	Theme parks
Cluster 4	Portuguese restaurants
Cluster 5	Bryanston
Cluster 6	Delis
Cluster 7	Doornfontein
Cluster 8	Coffee shops
Cluster 9	Indian restaurants
Cluster 10	Shopping malls and Chinese restaurants
Cluster 11	Cocktail bars
Cluster 12	Sunninghill
Cluster 13	Rock climbing gyms

Discussion

Folium mapping showed that there were parts of the city which was not included in the research due to lack of information in Nominatim. Google Maps has geocoding and places API which might yield better results. However, they were not free.

The recommendation tool should be interactive where a user should be able to type in a keyword to search for activities or venues.

The DBSCAN did have a lot of noises than expected. Kmeans clustering was considered but the best K was difficult to find. An elbow graph was created to find the best K, but there was no obvious point. Without knowing the K it's hard to cluster.

The categories might not be the best categories to describe venues. The biggest problem was that some of them were essentially the same, for example, shopping malls and malls.

As the city grew, there should be more venues in Foursquare database, hence the tourism database should be updated accordingly.

Conclusion

With an urban tourism themed database, and recommendation tools, it was easy to see that Johannesburg had a lot to offer. As a very developed urban area, visitors could easily choose to stay in the top 20 most diverse suburbs to enjoy themselves.

Foursquare was handy to map out the hot spots in the city.

References

Gauteng Tourism Authority. (2019). *2018-2019 Annual Report*
https://www.gauteng.net/pages/industry-information

UNWTO. Urban Tourism
https://www.unwto.org/urban-tourism

Wei,W. (2012). Research on the Application of Geographic Information System in Tourism Management. Procedia Environmental Sciences, 12(2012), 1104–1109.
https://www.sciencedirect.com/science/article/pii/S1878029612003957