The Battle of Neighborhoods:
Best location for new Gym and Fitness center

Julija Pirogova

Table of contents

Introduction: Business Problem

The City of Riga, is the capital and most populous city of Latvia. Almost half population of Latvia live or work in Riga. Nowadays in most cases work requires seating for multiple hours a day, which brings very unhealthy static tension to the body. Therefore, various sport venues became more and more popular among student and young workers.
It is quite nice choice for new business but where will be better choice to start?
Our goal to building a recommendation system for finding best suitable Gym/Fitness Center for "active-age" population (young people from 15- 49) based on certain criteria is valuable analytical problem that perfectly fits into Clustering type of Data Science problems which could be solved by unsupervised learning algorithms.

Target Audience: Group of activists and investors who want to start "gym/fitness business" in Riga.

Success Criteria: The success criteria of the project will be a good recommendation of Neighborhood in Riga based on possible lack of Gyms /Fitness Center.

Data

The data that will be required will be a combination of CSV files that have been downloaded for the purposes of the analysis from Central Statistical Bureau of Latvia , the Geographical location of the neighbourhoods (via Geocoder package) and Venue data pertaining to Gym/Fitness Centres in Riga (via Foursquare). The Venue data will help find which neighbourhood is best suitable to open new Gym/Fitness Center.

- **Riga City Neighborhood & Population.**

  Data source:

  http://data1.csb.gov.lv/pxweb/en/iedz/iedz__tautassk__riga__tsk2011/TSG11-R02.px/?rxid=0cebbc5d-dfba-43d1-903e-e18eb1de0eaf
  Description: This data from Central Statistical Bureau of Latvia contains the required information. It is not very new, but still can give us some insights. And we will use this data set to explore various neighborhoods of Riga.

- **GeoJSON file for Riga's Neighborhoods.**

  Data source: **https://github.com/art-licis/riga-geojson-neighbourhoods**
  Description: By using this json file we can create map of Riga with correct neighborhoods borders.

- **Gyms/Fitness Center in each neighborhood of Riga.**

  Data source: **Foursquare API**
  Description: By using this API we will get all the venues in each neighborhood.

Methodology

Collect Inspection Data

After importing the necessary libraries, we download the data from the website of *Central Statistical Bureau* of Latvia. Before using data, we will have to explore and understand it.

Explore and Understand Data

When getting the data from *Central Statistical Bureau*, we need to sum population of age 15-24 and 24-49 to get total of "active fitness age" population. As result we have a table with 58 Neighborhood in Riga, its total population and its "active age" population. First five row of this table looks like this:

[114]:

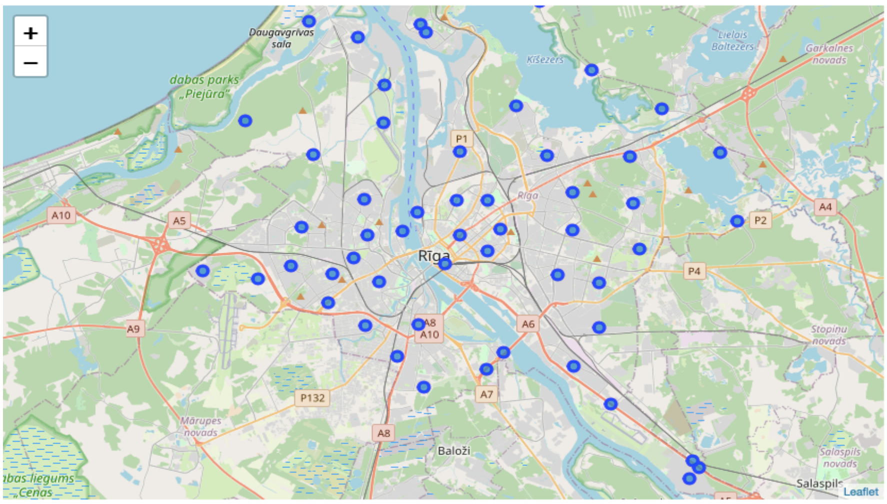
	Neighborhood	Total Population	Age 15-49
0	Sarkandaugava	18095	8588
1	Ziepniekkalns	33614	16801
2	Vecmilgravis	22667	10649
3	Daugavgriva	9015	4430
4	Kundzinsala	399	185
5	Ciekurkalns	7794	3942
6	Bisumuiza	2354	1164
7	Tornakalns	6947	3699
8	Agenskalns	26841	13123
9	Dzirciems	12079	5831

Then using Geocoder we get Latitude and Longitude for Riga and each of Neighborhood. And merge all these data into one data frame called "riga_data".

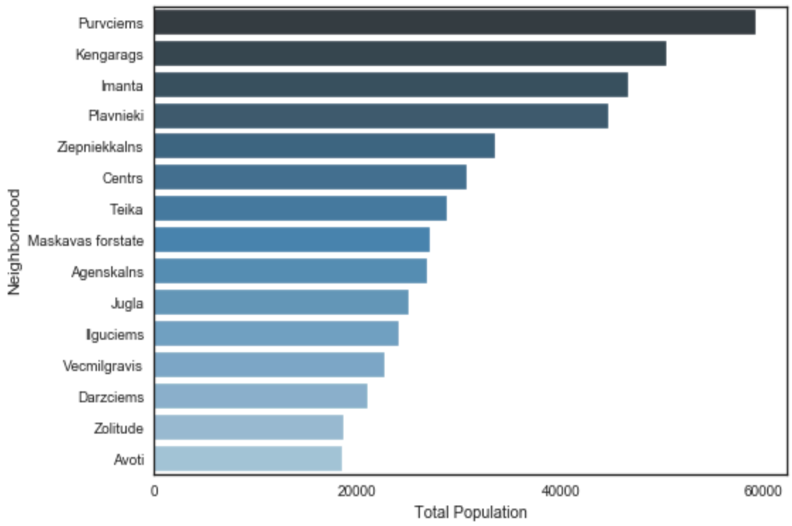
[7]:

	Neighborhood	Total Population	Age 15-49	latitude	longitude
0	Sarkandaugava	18095	8588	56.98857	24.12164
1	Ziepniekkalns	33614	16801	56.89951	24.09832
2	Vecmilgravis	22667	10649	57.03612	24.09670
3	Daugavgriva	9015	4430	57.03731	24.02443
4	Kundzinsala	399	185	57.16667	24.10556

Now after cleansing the data, we can visualize neighborhood centers on the map:



The next step was to analyze Neighborhood population. First of all, we look at total population.

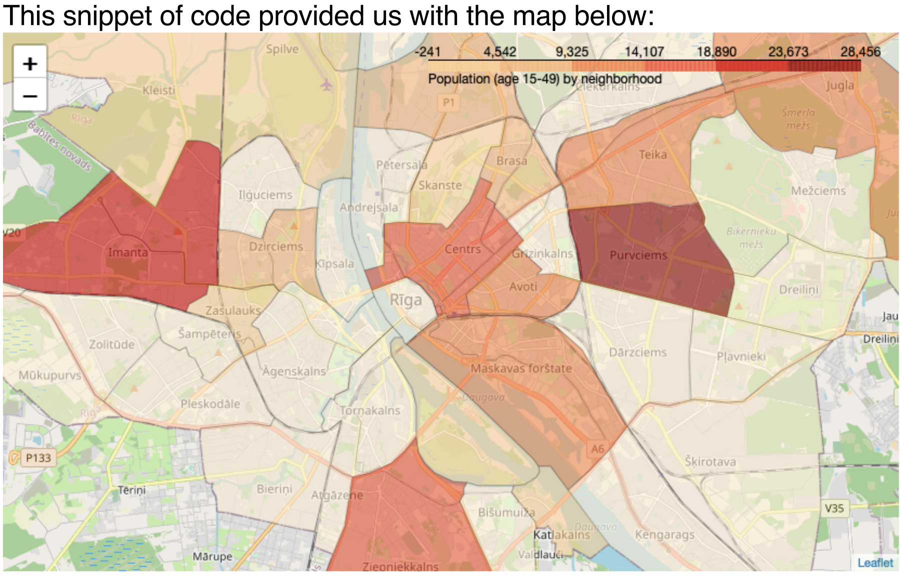


From our data we can see that Purvciems is the most populated neigborhood in Riga.

Then we look at population age 15-49. We created a map using Folium and choropleth each Neighborhood depending on "active fitness age (15-49) population".

```
# Creating map of Riga with population 15-49
riga_map = folium.Map(location=[riga_latitude, riga_longitude], zoom_start=12,tiles='openstreetmap')

riga_map.choropleth(
    geo_data='/Users/julija/IT/Coursera_Capstone/Apkaimes.json',
    data=riga_data,
    columns=['Neighborhood','Age 15-49'],
    key_on='feature.properties.apkaime',
    fill_color='OrRd',
    fill_opacity=0.6,
    line_opacity=0.2,
    legend_name = 'Population (age 15-49) by neighborhood'
)

riga_map
```

This snippet of code provided us with the map below:



Modeling

Using Foursquare API, we can get database of nearby venues for each neighborhood in Riga. Here in this table, we are able to choose 100 popular locations maximum for each neighborhood within a 2 km radius. Totally we get 267 venue categories. We then merged the Foursquare Venue data with the Riga data which then gave us the nearest Venue for each of the Neighborhoods.

```
There are 267 unique venue categories. Some of them are as below:
```

[100]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue ID	Venue Latitude	Venue Longitude	Venue Category
0	Sarkandaugava	56.98857	24.12164	MyFitness Sky&More	4f1ec79be4b0454d62987680	56.986270	24.132861	Gym
1	Sarkandaugava	56.98857	24.12164	Boksa Klubs Latvijas Cimdi	4f9eb6b7e4b0edc5630b92c6	56.990587	24.122617	Boxing Gym
2	Sarkandaugava	56.98857	24.12164	Nāc un ēd	535631b6498eb2d3777d377e	56.987213	24.121677	Bistro
3	Sarkandaugava	56.98857	24.12164	SKY lielveikals	58668b6865e7c70a2f851761	56.986478	24.132768	Grocery Store
4	Sarkandaugava	56.98857	24.12164	Radio SWH	4d67ea96052ea1cdee3da049	56.981202	24.117657	Music Venue

Getting this data was crucial to analyzing the number of Gyms /Fitness Center all over Riga. There was a total of 103 Gym/ Fitness Center in Riga.

Are there any Gyms/Fitness Center in the venues?

```
[25]: "Gym / Fitness Center" in riga_venues['Venue Category'].unique()
```

```
[25]: True
```

```
[26]: riga_venues['Venue Category'].value_counts()['Gym / Fitness Center']
```

```
[26]: 103
```

Next we performed a technique called **One hot encoding**. For each of the neighbourhoods, individual venues were turned into the frequency at how many of those Venues were located in each neighbourhood.

```
[28]: # one hot encoding
riga_onehot = pd.get_dummies(riga_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
riga_onehot['Neighborhoods'] = riga_venues['Neighborhood']

# move neighborhood column to the first column
fixed_columns = [riga_onehot.columns[-1]] + list(riga_onehot.columns[:-1])
riga_onehot = riga_onehot[fixed_columns]

print(riga_onehot.shape)
riga_onehot.head()
```

```
(3183, 268)
```

[28]:

	Neighborhoods	ATM	Accessories Store	Adult Boutique	Airport	American Restaurant	Animal Shelter	Arcade	Argentinian Restaurant	Art Gallery	...	Vietnamese Restaurant	Volleyl Co
0	Sarkandaugava	0	0	0	0	0	0	0	0	0	...	0	
1	Sarkandaugava	0	0	0	0	0	0	0	0	0	...	0	
2	Sarkandaugava	0	0	0	0	0	0	0	0	0	...	0	
3	Sarkandaugava	0	0	0	0	0	0	0	0	0	...	0	
4	Sarkandaugava	0	0	0	0	0	0	0	0	0	...	0	

5 rows × 268 columns

Then we grouped those rows by Neighborhood and by taking the **average** of the frequency of occurrence of each Venue Category.

```
[30]: riga_grouped = riga_onehot.groupby(["Neighborhoods"]).mean().reset_index()

      print(riga_grouped.shape)
      riga_grouped.head()
```

```
(57, 268)
```

[30]:

	Neighborhoods	ATM	Accessories Store	Adult Boutique	Airport	American Restaurant	Animal Shelter	Arcade	Argentinian Restaurant	Art Gallery	...	Vietnamese Restaurant	Voll
0	Agenskalns	0.0	0.0	0.00	0.000000	0.0	0.0	0.0	0.0	0.01	...	0.000000	
1	Atgazene	0.0	0.0	0.00	0.000000	0.0	0.0	0.0	0.0	0.00	...	0.015152	
2	Avoti	0.0	0.0	0.01	0.000000	0.0	0.0	0.0	0.0	0.01	...	0.000000	
3	Beberbeki	0.0	0.0	0.00	0.043478	0.0	0.0	0.0	0.0	0.00	...	0.000000	
4	Bergi	0.0	0.0	0.00	0.000000	0.0	0.0	0.0	0.0	0.00	...	0.000000	

5 rows × 268 columns

After, we created a new data frame that only stored the Neighborhood names as well as the mean frequency of Gyms/Fitness Center in that Neighborhood. This allowed the data to be summarized based on each individual Neighborhood and made the data much simpler to analyze.

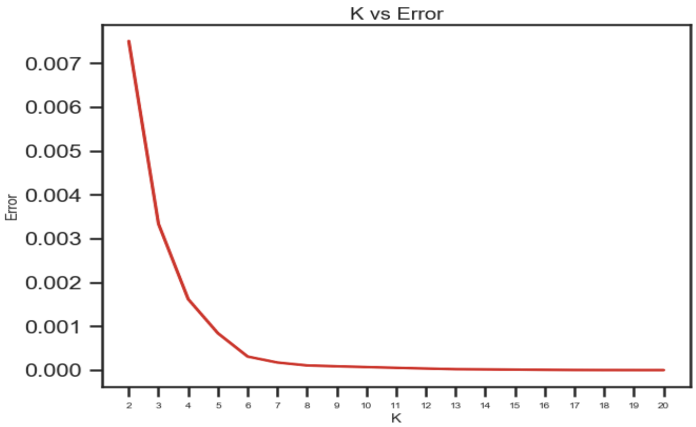
```
[31]: gym_fit = riga_grouped[["Neighborhoods","Gym / Fitness Center"]]
      gym_fit
```

[31]:

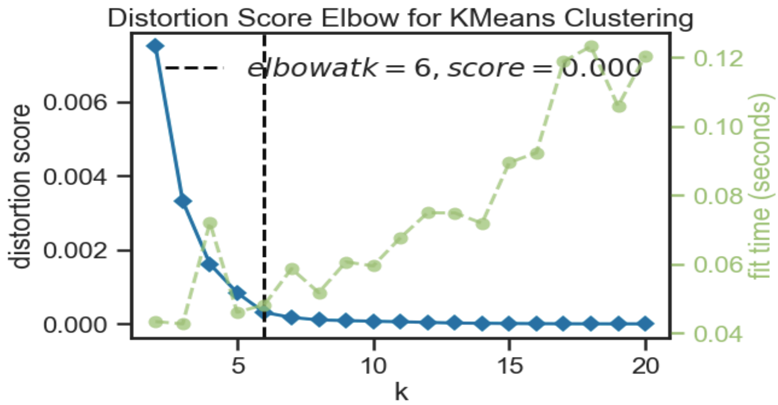
	Neighborhoods	Gym / Fitness Center
0	Agenskalns	0.040000
1	Atgazene	0.045455
2	Avoti	0.030000
3	Beberbeki	0.000000
4	Bergi	0.000000
5	Bierini	0.030000

K-Means Clustering

To make the analysis more interesting, we wanted to cluster the neighbourhoods based on the neighbourhoods that had similar averages of Gyms in that Neighborhood. To do this we used **K-Means** clustering. To get our optimum K value that was neither overfitting or underfitting the model, we used the **Elbow Point** Technique. In this technique, we ran a test with different number of K values and measured the accuracy and then chose the best K value. The best K value is chosen at the point in which the line has the sharpest turn. In our case, we had the Elbow Point at K = 6. That means we will have a total of 6 clusters.



Then we used a model that accurately pointed out the optimum K value. We imported '***KElbowVisualizer***' from the ***Yellowbrick package.*** Then we fit our K-Means model above to the Elbow visualizer. This gave the model below:



We just integrated a model that would fit the error and calculate the distortion score. From the dotted line, we see that the Elbow is at K=6. Moreover, in K-Means clustering, objects that are similar based on a certain variable are put into the same cluster. Neighbourhoods that had a similar mean frequency of Gyms/Fitness Center were divided into 6 clusters. Each of these clusters was labelled from 0 to 5 as the indexing of labels begins with 0.

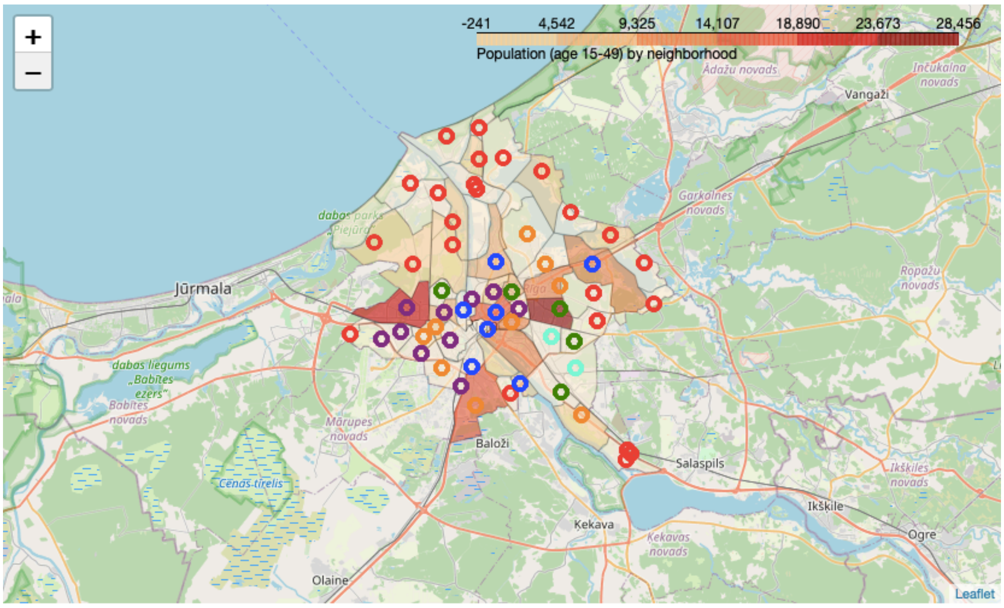
```
[40]: # create a new dataframe that includes the cluster .
      riga_merged = gym_fit.copy()

      # add clustering labels
      riga_merged["Cluster Labels"] = kmeans.labels_
      riga_merged.head()
```

[40]:

	Neighborhood	Gym / Fitness Center	Cluster Labels
0	Agenskalns	0.040000	1
1	Atgazene	0.045455	1
2	Avoti	0.030000	5
3	Beberbeki	0.000000	0
4	Bergi	0.000000	0

After, we merged the venue data with the table above creating a new table which would be the basis for analyzing new opportunities for opening a new Gym/Fitness Center in Riga. Then we

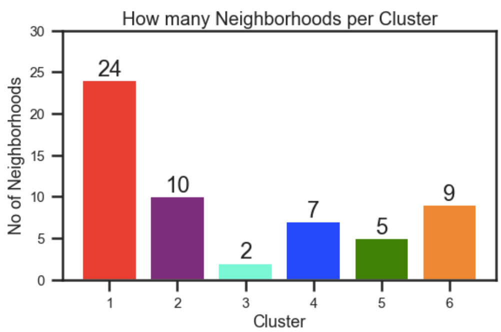
created a map using the Folium package in Python and each neighbourhood was coloured based on the cluster label and population (age 15-49).
Cluster 1 — Red
Cluster 2 — Purple
Cluster 3 — Turquoise/ aquamarine
Cluster 4 — Blue
Cluster 5 —Green
Cluster 6 — Orange

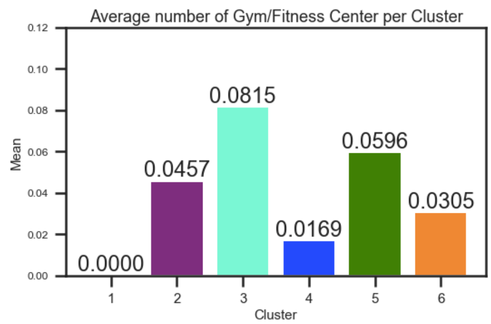


The map above shows Population (age 15-49) by Neighborhood and the different clusters that had a similar mean frequency of Gym/Fitness.

Analysis and Results

We have a total of 6 clusters (0,1,2,3,4,5). Before we analyze them one by one let's check the total amount of neighbourhoods in each cluster and the average Gym/Fitness Center in that cluster. From the bar graph that was made using Matplotlib, we can compare the number of Neighborhoods per Cluster. We see that Cluster 3 has the least neighbourhoods (2) while Cluster 1 has the most (24). Cluster 2 has 10 neighbourhoods, Cluster 6 has 9, Cluster 4 has 7 and Cluster 5 has only 5. Then we compared the average Gym/ Fitness Center per cluster.





Cluster Analysis

As we can see that even though there is only 2 neighbourhood in Cluster 3, it has the highest number of Gym/Fitness Center (0.08) while Cluster 1 has the most neighbourhoods but has the least average of Gym/Fitness Center = zero (0.0000). Also, from the map, we can see that neighbourhoods in Cluster 1 are the most sparsely populated. Now let's analyze the Clusters individually (Note: these are just snippets of the data).

Cluster 1(Red)

```
[134]: df_cluster1.sort_values(["Gym / Fitness Center"], ascending=False)
```

[134]:

	Neighborhood	Gym / Fitness Center	Cluster Labels	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue ID	Venue Latitude	Lon
0	Vecmilgravis	0.0	0	57.036120	24.096700	Bérnu rotaļu laukums pie Daugavas	51811e38498e034fb2fa5b2e	57.028287	24.
351	Spilve	0.0	0	56.999170	24.072390	Voleru Mols	51f4ebb4498ee36a4104be19	56.992860	24.0
365	Voleri	0.0	0	57.013320	24.073020	Rīgas jūras osta "Voleri"	4dd7a5a0c65bee535ab69f3f	57.004158	24.0

```
[245]: print('There are {} uniques venues.'.format(len(df_cluster1['Venue'].unique())))
       "Gym / Fitness Center" in df_cluster1['Venue Category'].unique()

       There are 362 uniques venues.
[245]: False
```

There was a most of neighbourhoods (24), 362 different venues and NO Gym/Fitness Center at all. May be because of Neighborhoods in Cluster 1 are the most sparsely populated in Riga, mainly located in Riga suburb, far from historical and office center.

Cluster 2 (Purple):

```
[112]: df_cluster2.sort_values(["Gym / Fitness Center"], ascending=False).head()
```

[112]:

	Neighborhood	Gym / Fitness Center	Cluster Labels	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue ID	Venue Latitude	Venue Longitude
245	Pleskodale	0.05	1	56.93146	24.03649	Route88	548b3608498e2d4a132f509a	56.926791	24.040669
619	Petersala-Andrejsala	0.05	1	56.96556	24.09472	Esplanāde	4bbf8cff4cdfc9b608759121	56.954746	24.114754
608	Petersala-Andrejsala	0.05	1	56.96556	24.09472	CrossFit Rīdzene	52ee91e1498eb46f763dcc78	56.974720	24.104693
609	Petersala-Andrejsala	0.05	1	56.96556	24.09472	Klubs "Nauda"	4fb81f58e4b0f914e0bc22f5	56.964770	24.078040
610	Petersala-Andrejsala	0.05	1	56.96556	24.09472	Pagalms	559418f4498ebeb409dde02c	56.954616	24.104573

```
[244]: print('There are {} uniques venues.'.format(len(df_cluster2['Venue'].unique())))
       print("Gym / Fitness Center" in df_cluster2['Venue Category'].unique())
       print('There are {} Gym/Fitness Center.'.format(df_cluster2['Venue Category'].value_counts(ascending=False)[

       There are 543 uniques venues.
       True
       There are 38 Gym/Fitness Center.
```

Second place by Neighborhood count (10) and 543 different venues. It also has most Gym/Fitness centers - 38. But the average amount of Gym/Fitness Centers that were 0.0457. In the map, we can see that neighborhoods of Cluster 2 also sparsely populated clusters (except Imanta – third most populated Neighborhood in Riga).

Cluster 3 (Turquoise):

Last Saved: 2020-06-25 19:10
Last Checkpoint: 2020-06-19 11:15

```
[114]: cluster3 = riga_merged.loc[riga_merged['Cluster Labels'] == 2]
       df_cluster3 = pd.merge(df_new, cluster3, on='Neighborhood')
       df_cluster3.head()
```

[114]:

	Neighborhood	Gym / Fitness Center	Cluster Labels	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue ID	Venue Latitude	Venue Longitude
0	Darzciems	0.08	2	56.94213	24.18494	Titan	4d00fedf4f56b60c67dea437	56.932124	24.207546
1	Darzciems	0.08	2	56.94213	24.18494	AURA | Salons, frizētava	50afa79ce4b00d9395e9dd98	56.944470	24.170429
2	Darzciems	0.08	2	56.94213	24.18494	Rimi Hipermārkets [Stirnu]	4c546eb96a4bb71326f43227	56.958416	24.185908
3	Darzciems	0.08	2	56.94213	24.18494	T/C Green	5752b619498e75528b0da79f	56.944964	24.173156
4	Darzciems	0.08	2	56.94213	24.18494	Sushi Shop	50d9c270e4b05cc5ef0bd9b8	56.945617	24.208752

```
[247]: print('There are {} uniques venues.'.format(len(df_cluster3['Venue'].unique())))
       print("Gym / Fitness Center" in df_cluster3['Venue Category'].unique())
       print('There are {} Gym/Fitness Center.'.format(df_cluster3['Venue Category'].value_counts(ascending=False)[

       There are 103 uniques venues.
       True
       There are 11 Gym/Fitness Center.
```

Cluster 3 located in the east of Riga .There was a total of 103 unique venues and out of those 11 were Gym / Fitness Centers. It has the highest average of Gym / Fitness Centers (0.0815).
The reason why the AVERAGE of Gym / Fitness Centers is the highest is that all these Gyms are in two neighbourhoods, Darzciems and Skirotava.

Cluster 4 (Blue):

Cluster 4 - BLUE

```
[116]: cluster4 = riga_merged.loc[riga_merged['Cluster Labels'] == 3]
       df_cluster4 = pd.merge(df_new, cluster4, on='Neighborhood')
       df_cluster4.head()
```

[116]:

	Neighborhood	Gym / Fitness Center	Cluster Labels	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue ID	Venue Latitude	Lo
0	Sarkandaugava	0.014286	3	56.98857	24.12164	Latvijas Finieris atgriezumi	4f7558bae4b06c81e0a07d65	56.981858	24
1	Sarkandaugava	0.014286	3	56.98857	24.12164	Tik Tak	504902a8e4b06b90916c37c3	56.976028	24.
2	Sarkandaugava	0.014286	3	56.98857	24.12164	LDz | Dzelzceļa stacija "Sarkandaugava" (SARKA...	502006bee4b09c1a7062b1c9	56.993061	24
3	Sarkandaugava	0.014286	3	56.98857	24.12164	Čarlstons Express	508906c3e4b0db195ec061a3	56.977782	24.
4	Sarkandaugava	0.014286	3	56.98857	24.12164	Jūrnieks Hotel Riga	4bc87c77dc55eee1917ee8ac	56.999351	24

```
[249]: 'There are {} uniques venues.'.format(len(df_cluster4['Venue'].unique())))
       'Gym / Fitness Center" in df_cluster4['Venue Category'].unique())
       'There are {} Gym/Fitness Center.'.format(df_cluster4['Venue Category'].value_counts(ascending=False)['Gym

       There are 477 uniques venues.
       True
       There are 10 Gym/Fitness Center.
```

Cluster 4 was mainly located in the historical & office center of Riga . There were a total of 477 unique Venues in Cluster 4 with 10 Gym/Fitness Center. But the average of Gym / Fitness Centers was rather low - 0.0169.

Cluster 5 (Green):

Cluster 5 - GREEN

[118]:
```
cluster5 = riga_merged.loc[riga_merged['Cluster Labels'] == 4]
df_cluster5 = pd.merge(df_new, cluster5, on='Neighborhood')
df_cluster5.head()
```

[118]:

	Neighborhood	Gym / Fitness Center	Cluster Labels	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue ID	Venue Latitude	Venue Longitude
0	Ilguciems	0.055556	4	56.97035	24.05994	Ziedu Ekspresis	4d849c2b61676dcbe5a275e4	56.974705	24.040855
1	Ilguciems	0.055556	4	56.97035	24.05994	Andrejsala / Siena	518130e8498e79c2cb6c9b8e	56.970926	24.092477
2	Ilguciems	0.055556	4	56.97035	24.05994	Balasta Dambis	4d7273ced976236a1b9a0679	56.958509	24.084736
3	Ilguciems	0.055556	4	56.97035	24.05994	Nellijas pirts	5051b59ae4b06a9490cd4036	56.963427	24.061836
4	Ilguciems	0.055556	4	56.97035	24.05994	Aviācijas Muzejs	4ff2fd48e4b00cc51ba841b6	56.976149	24.072461

[250]:
```
print('There are {} uniques venues.'.format(len(df_cluster5['Venue'].unique())))
print("Gym / Fitness Center" in df_cluster5['Venue Category'].unique())
print('There are {} Gym/Fitness Center.'.format(df_cluster5['Venue Category'].value_counts(ascending=False)[
```

```
There are 316 uniques venues.
True
There are 22 Gym/Fitness Center.
```

Neighborhoods of Cluster 5 are mainly residential sleeping areas. . 316 unique venues and 22 Gym/Fitness Center. There are 5 rather populated Neighborhoods including Purvciems (most populated Neighborhood in Riga) .Therefore second-highest average of Gym/ Fitness Centers in that cluster which was approximately 0.0596.

Cluster 6 (Orange):

Cluster 6 - ORANGE

[140]:
```
cluster6 = riga_merged.loc[riga_merged['Cluster Labels'] == 5]
df_cluster6 = pd.merge(df_new, cluster6, on='Neighborhood')
df_cluster6.head()
```

[140]:

	Neighborhood	Gym / Fitness Center	Cluster Labels	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue ID	Venue Latitude	Venue Longitude
0	Ziepniekkalns	0.03125	5	56.89951	24.09832	VANS kafejnica	4e34438db0fb59390e91dfe4	56.899562	24.084457
1	Ziepniekkalns	0.03125	5	56.89951	24.09832	BK Kolibri/47.vsk Sporta Zāle	528aeb0e11d2ed2fd8ae1379	56.915933	24.103235
2	Ziepniekkalns	0.03125	5	56.89951	24.09832	Pica Lulū picērija	4cdb1669c409b60c18a8d21a	56.907834	24.082500
3	Ziepniekkalns	0.03125	5	56.89951	24.09832	McDonald's	4f678d4e7b0cfc7ca4ba1059	56.907041	24.082914
4	Ziepniekkalns	0.03125	5	56.89951	24.09832	Turība | Café "Turība"	50446f06e4b072dc97417d9c	56.910085	24.081150

[251]:
```
print('There are {} uniques venues.'.format(len(df_cluster6['Venue'].unique())))
print("Gym / Fitness Center" in df_cluster6['Venue Category'].unique())
print('There are {} Gym/Fitness Center.'.format(df_cluster6['Venue Category'].value_counts(ascending=False)[
```

```
There are 503 uniques venues.
True
There are 22 Gym/Fitness Center.
```

Cluster 6 located in not very expensive but extremely developing areas. Teika, Avoti, Ciekurkalns (in general 9 neighborhood per this cluster) there are many new office centers and young potential customers. Cluster 6 has 503 different venues and 22 Gym/Fitness Center. Average of Gyms 0.0305.

Therefore, the ordering of the average Gym / Fitness Centers in each cluster goes as follows:
1. Cluster 3 (≈0.0815)
2. Cluster 5 (≈0.0596)
3. Cluster 2 (≈0.0457)
4. Cluster 6 (≈0.0305)
5. Cluster 4 (≈0.0169)
6. Cluster 1 (≈0.0000)

Discussion

Most of the Gym/Fitness Center are in cluster 3 represented by the turquoise clusters. The Neighborhoods located in the 2 residential areas in Riga (but the not most populated) that have the highest average of Gym/Fitness Center. Even though there is a huge number of Neighborhoods in Cluster 1, there is no Gym/Fitness Centers. We see that in the most populated neighborhood Purvciems and rather populated (age 15-49) Brasa (cluster 5) has the second last average of Gym/Fitness Center.
Looking at the nearby venues, the optimum place to put a new Gym/Fitness Center in Cluster 4 as there are many news offices but little to no Gym/Fitness Center, therefore, eliminating any competition. The second-best Neighborhoods that have a great opportunity would be in areas such as Maskavas forstate. There is no Gym/Fitness Center at all, but it is rather populated and neighborhoods with many young people (potential clients) such as Avoti or Center are not far away. Having 24 neighbourhoods in the area with no Gym/Fitness Center gives a good opportunity for opening a new one. Some of the drawbacks of this analysis are — the clustering is completely based on data obtained from the Foursquare API. Also, the analysis does not take into consideration of the income level of population across neighbourhoods as this can play a huge factor while choosing which place to open a new Gym/ Fitness Center. This concludes the optimal findings for this project and recommends the entrepreneur to open an Gym/ Fitness Center in these locations with little to no competition.

Conclusion

In conclusion, to end off this project, we had an opportunity on a business problem, and it was tackled in a way that it was similar to how a genuine data scientist would do. We utilized numerous Python libraries to fetch the information, control the content and break down and visualize those datasets. We have utilized Foursquare API to investigate the settings in neighbourhoods of Riga. We also visualized utilizing different plots present in seaborn and Matplotlib libraries. Similarly, we applied AI strategy to anticipate the error given the information and utilized Folium to picture it on a map.
Places that have room for improvement or certain drawbacks give us that this project can be additionally improved with the assistance of more information and distinctive Machine Learning strategies. Additionally, we can utilize this venture to investigate any situation, for example, opening an Restaurant or Cafe and so forth. Ideally, this task acts as an initial direction to tackle more complex real-life problems using data science.