Analysis and K-Means Clustering of Restaurants in Greater Helsinki Area

COURSERA DATA SCIENCE CAPSTONE PROJECT

MARCH 2019

ANSSI HAKKARAINEN

Introduction

- The purpose of this project was to explore neighbourhoods in greater Helsinki area (Finland) to find the best places for a new restaurant. The target audience for this report is anyone looking to setup a certain type new restaurant in the area. The analysis will shed light into questions like: how many restaurants of each type already exists in different areas, how many people live there and how many restaurants there is per capita.

- The data analysis was carried out using Jupyter notebook with Python programming language and was published in Github. Python's pandas library was used for the analysis and findings will be visualised using Matplotlib and Folium libraries.

Getting neighbourhood data

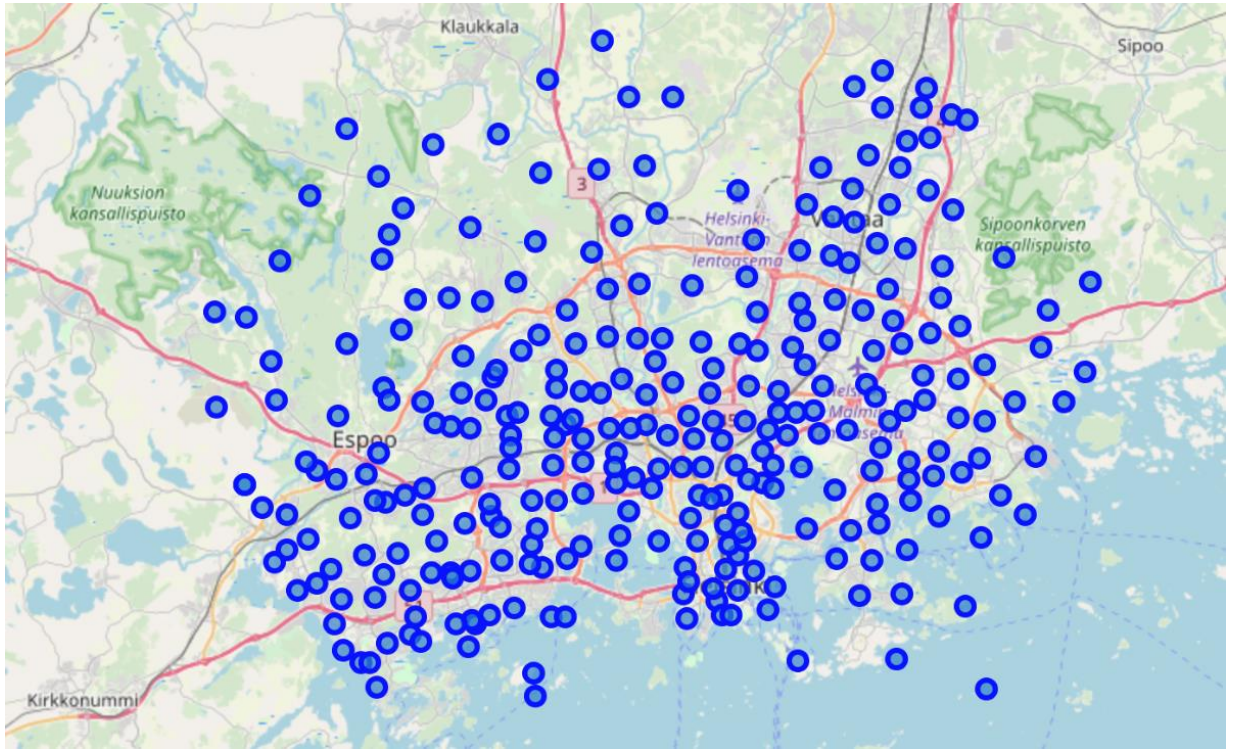
- Greater Helsinki neighbourhood data was gathered from Helsinki Region Infoshare, https://hri.fi/en_gb/ , which is a great resource for a lot of interesting regional data. It currently boasts over 600 different datasets.
- The data I was interested in was 1) list of neighbourhoods 2) their coordinates and 3) their population. For this a combines two datasets which was easy enough with Python's pandas library:

```
#Merge dataframe with the one that has income data
dfmerged = pd.merge(df_data_1,df, on=['name'])
dfmerged[["lat", "loc"]] = dfmerged[["lat", "loc"]].apply(pd.to_numeric)
dfmerged.drop_duplicates(subset="name", inplace=True)
#Just take columns we need, "2019" is the population figure for 2019
dfmerged=dfmerged[["name","lat","loc","2019"]]
dfmerged.head()
```

	name	lat	loc	2019
0	Kruununhaka	60.172643	24.964175	7470
1	Kluuvi	60.173882	24.942115	649
2	Katajanokka	60.166575	24.980714	4874
3	Kaartinkaupunki	60.165507	24.950692	1056
4	Punavuori	60.161397	24.935337	9417

Greater Helsinki neighbourhoods

- Visualizing the different neighbourhoods was done with Folium library:



Getting info about restaurants

- The project instructions mandated use of Foursquare API for venue information.
- Luckily the data in fairly comprehensive, although it later became evident it is far from perfect what comes to Helsinki area restaurants.
- With Python's requests librarty it is pretty straightforward to use Foursquare's REST API

```
        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.for
mat(
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

        # make the GET request
        results = requests.get(url).json()["response"]['groups'][0]['items']
```

First look at restaurant data

- Total of 2833 venues were found

```
#How many venues we got
print(venues.shape)
venues.head()
```

```
(2833, 7)
```

- But just 387 were restaurants so others were removed

```
# We only are interested in restaurants so let's drop others
venues = venues[venues['Venue Category'].str.contains("Restaurant")]
print(venues.shape)
venues.head()
```

```
(387, 7)
```

Restaurants vs. population

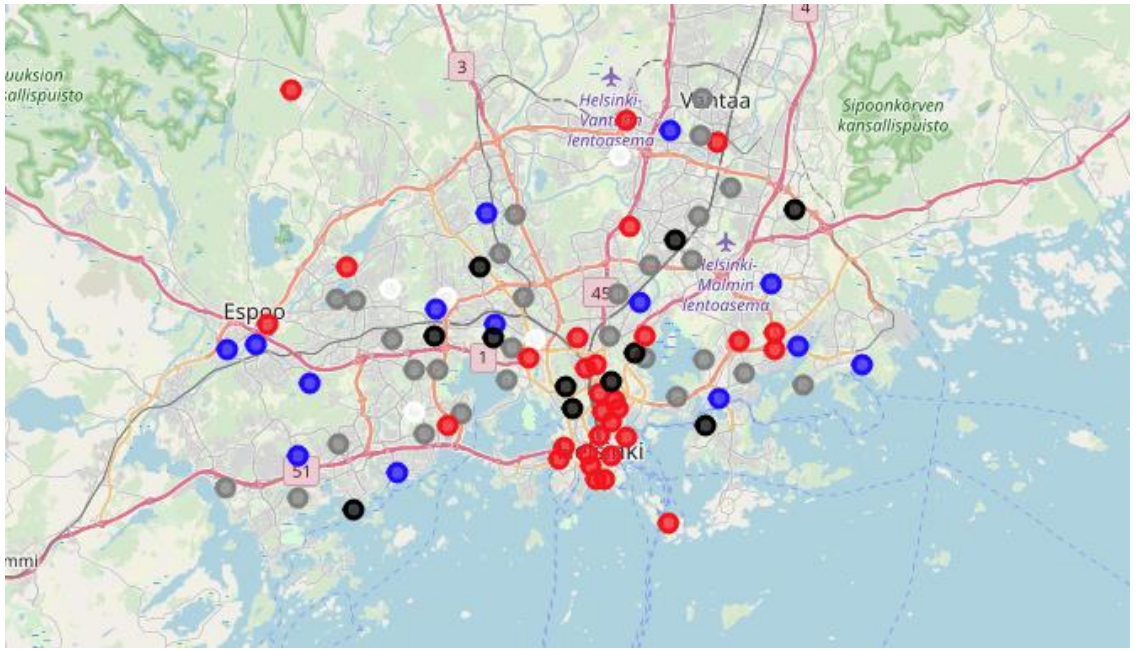
- I was interested to find out how many restaurants there is per capita (or per 1000 capita) within each neighbourhood.

	Neighborhood	Restaurants	Population	Rest per 1000 capita	lat	loc
0	Ala-Malmi	1	6481	0.154297	60.244835	25.017064
1	Alppila	6	4602	1.303781	60.190469	24.941519
2	Arabianranta	1	7550	0.132450	60.205128	24.978939
3	Aurinkolahti	3	8003	0.374859	60.202623	25.155513
4	Eestinmalmi	1	3145	0.317965	60.165339	24.695261
5	Eira	9	1108	8.122744	60.155592	24.939001
6	Etelä-Haaga	1	12488	0.080077	60.212566	24.889656
7	Etelä-Leppävaara	7	7425	0.942761	60.213938	24.806510
8	Harju	14	7566	1.850383	60.188088	24.954459
9	Haukilahti	2	5677	0.352299	60.158607	24.776324
10	Herttoniemenranta	5	9297	0.537808	60.188674	25.038892

- As expected this varied a lot as rural neighbourhoods has hardly any whereas some areas had tens of restaurants per 1000 inhabitants.

Restaurants vs. population cont'd

- The density was visualized with colours white-grey-blue-black-red: white being the least dense (less than 0.1 restaurants per 1000 people) and red the most dense (minimum 1.0 0.1 restaurants per 1000 people)

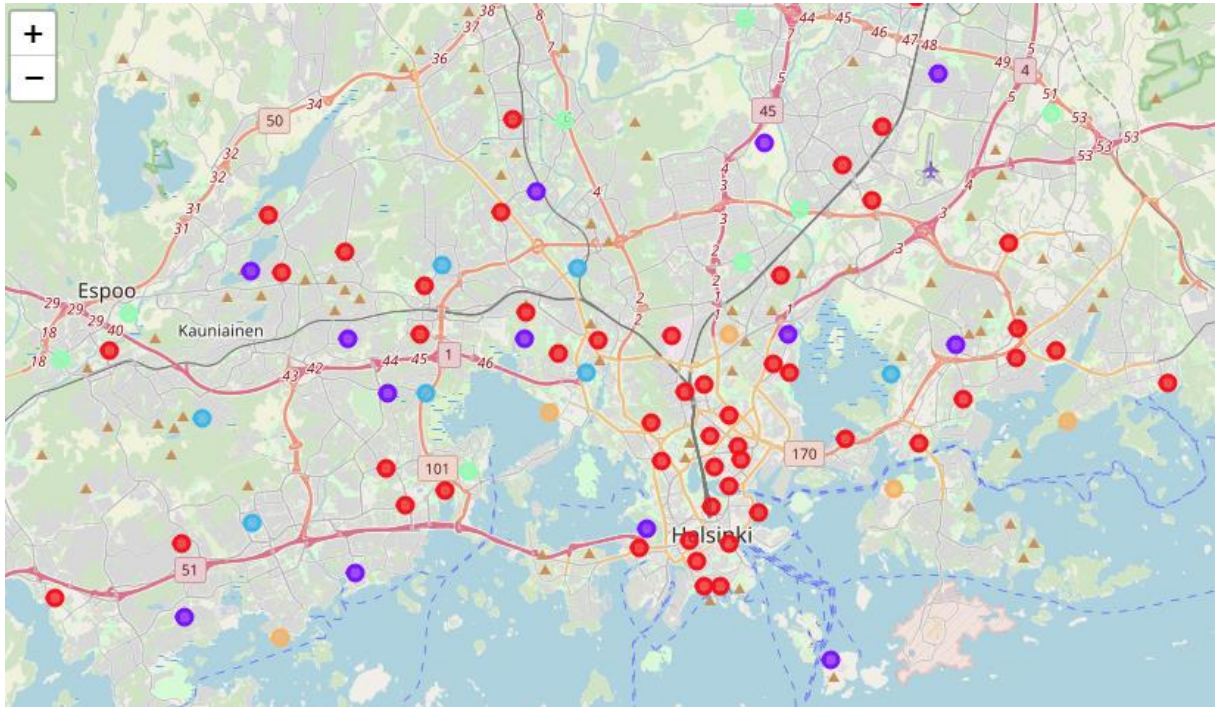


Enter data science, K-means clustering

- Finding, preparing and dealing with peculiarities of pandas dataframes was the most time consuming part of the project.
- However, by far the most intriguing for me was the most *data scientific* part of the study: the clustering.
- I did the clustering of the neighbourhoods with **K-means algorithm** using Python's Scikit learn library.
- K-means clustering is one of the simplest and popular unsupervised machine learning algorithms.
- the objective of K-means is simple: group similar data points together and discover underlying patterns. To achieve this objective, K-means looks for a fixed number (*k*) of clusters in a dataset."
- A cluster refers to a collection of data points aggregated together because of certain similarities.
- In this case the similarities of neighbourhoods were based on which types of restaurants were most common.

Clusters visualized

- I chose number of clusters to be 5 (that's the K in K-means), the end result shown here.



Conclusion

- For me this project was a great learning experience. I have used tools as such as Matlab and R before but Python's data science libraries were new to me.
- How about usefulness of the study for the original business question: "what would be the best places for a new restaurant in Greater Helsinki area"?
- I think there definitely are useful pieces of data such as the restaurant density information and for sure the clusters (i.e. which neighbourhoods are similar based on most common types of restaurants). However, anyone really looking into setting up a new restaurant need to do their part of the puzzle, e.g. by carrying out market research.