Capstone project -- Opening a restaurant

1. Introduction and business problem

Background

Nowadays, it is difficult to imagine a city without a restaurant or a venue for food where people can have a meal or drink. The city of my choice is Taganrog that is the leading historic, cultural and industrial center in the South of Russia. Local industry and businesses are represented by aerospace, machine-building, military, iron and steel industry, farming, food, theaters, museums and one of the major ports of Azov Sea. That means there are a lot of business opportunities for restaurant business what leads to high competition.

To survive in such competitive market it is very important to find right place and take into account many other important factors such as:

- City population
- Sport and Entertainment zones
- Food markets with products of local farmers
- Local competitors and their ratings
- etc

Problem

In order to reduce the risks and avoid loss of money, the analysis of all accessible data should be carried out carefully in order to choose a suitable place or location. In my opinion, even an amazing idea or ingenious concept will not make your restaurant business successful without suitable place for it.

Obviously, this project will be interesting for a big company as well as anyone who wants to open a new restaurant in Taganrog city.

2. Data description

The city of **Taganrog** (located in the South of Russia) will be analyzed in this project.

To solve the problem of finding the right location, we should find all existing businesses in the city of interest, explore them carefully to understand what we already have and plot our venues on map to gain insights, possible patterns or clusters.

For further analysis we will use the following data sources:

1. Wikipedia page for city population (https://en.wikipedia.org/wiki/Taganrog)
2. Nominatim search engine for OpenStreetMap data to get the bounding box of the city (https://nominatim.openstreetmap.org/search?format=json&q=Taganrog&polygon_geojson=1)
3. Foursquare API

Foursquare will be used as the main data source for analysis. We will retrieve both geographical coordinates and additional information about each venue using Foursquare API.

The following attributes for each venue will be collected:

- Id -- venue id (in order to remove duplicates)

- Venue -- venue name
- Category -- venue category
- Location -- venue address
- Latitude -- venue latitude
- Longitude -- venue longitude
- Rating -- numerical rating of the venue (0 through 10)
- Tips -- total count of tips
- Likes -- the count of users who have liked this venue

Because of Taganrog city has no neighborhood division like cities in the United States, we will build a coordinate grid that will cover the entire city by cells (or squares) of size 0.005x0.005 or 700x700 meter approximately. South-west (sw) and north-east (ne) corners of cells will be utilized as input for the **search** endpoint of Foursquare API.

Example of coordinate grid

Example of data from Foursquare

The first five venues within 700 meters bounding box of city center are below:

	Venue	Latitude	Longitude	Category	Id
0	Площадь перед администрацией города	47.215733	38.928230	Plaza	5368f4ad498ea0cb80cef632
1	Культ вина	47.215510	38.929310	Wine Bar	5c74142e60255e002c1aefbc
2	Театр имени А. П. Чехова	47.216325	38.928217	Theater	4dcbe98a1f6ea1401d49d12a
3	Администрация Таганрога	47.215517	38.928420	City Hall	4da693d90cb66f658708dafc
4	Л'Этуаль	47.215416	38.929266	Cosmetics Shop	4f83002ee4b0b2237e8a6cb1

When data collected, the general approach to solution is to cluster venues in the city and identify what cluster fit best of all. It may either be a cluster with the most popular venues for food or a cluster having similar parameters but the least venues for food.

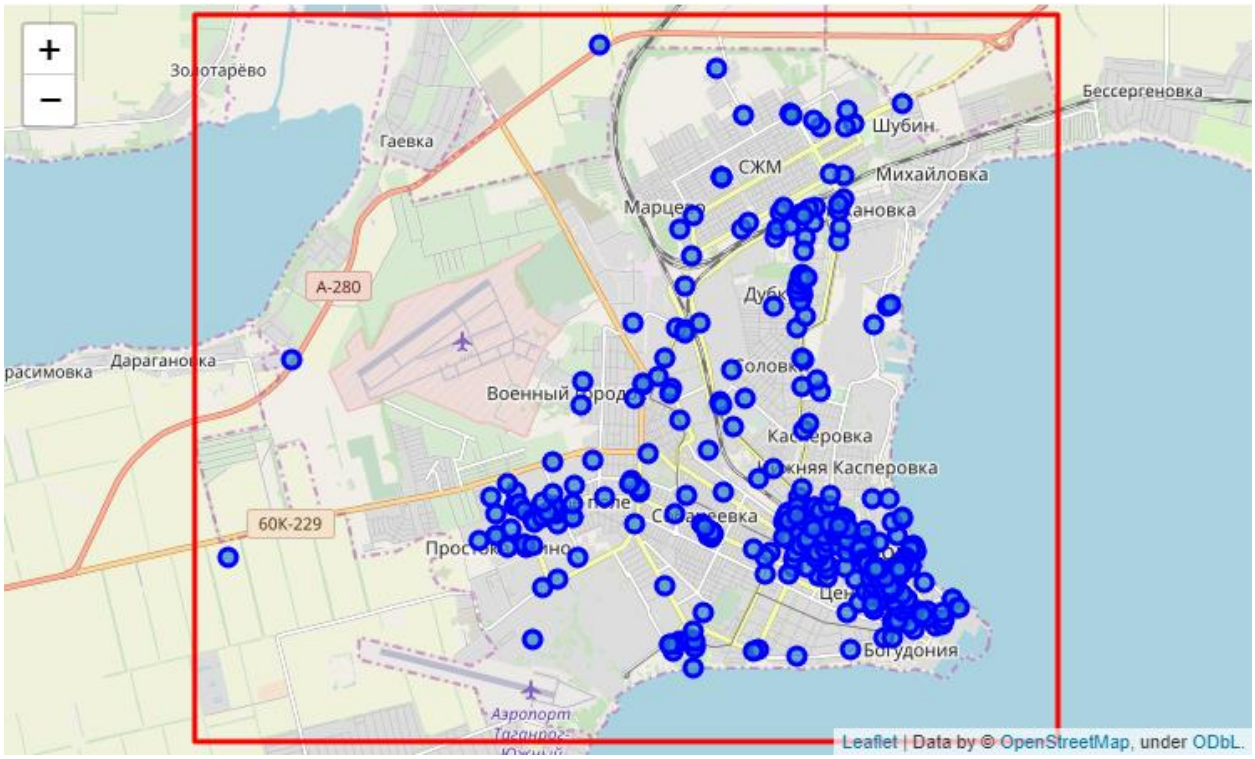
Data collection

This section covers data acquisition steps:

- Get the city bounding box using Nominatim search engine for OpenStreetMap data;
- Build a coordinate grid that covers the entire city by cells;
- Get the list of all venues in the city iterating over cells of coordinate grid;
- Get all possible categories from Foursquare using its API endpoint **categories** in order to determine venues for food among all city venues;

	Venue	Latitude	Longitude	Category	Food_venue
0	Место Разворота Маршруток	47.202091	38.854572	Bus Line	0
1	Ресторан Пирамила	47.217564	38.855255	Eastern European Restaurant	1
2	штрафстоянка	47.221715	38.831101	Parking	0
3	ДокАвто	47.223408	38.858902	Gas Station	0
4	Евролюкс	47.224610	38.844511	General Travel	0

- Display venues on map with the bounding box



- Get detailed information about each venue and put it in a new dataset;

	Id	Rating	Likes	Tips	PostalCode	Price_tier	createdAt	Category_id
0	55a0e60c498e873de699ccd1	0.0	7.0	0.0	347900	0.0	1.436608e+09	4bf58dd8d48988d132951735
1	58401718e9233e42ab37615e	5.8	1.0	0.0	347900	0.0	1.480595e+09	52f2ab2ebcbc57f1066b8b46
2	51ecf0f2498edbbbb8c6c29d	0.0	0.0	0.0	347900	0.0	1.374483e+09	4bf58dd8d48988d1e2941735
3	53f917c4498e47b1dc40249e	0.0	1.0	0.0	347900	0.0	1.408833e+09	4bf58dd8d48988d1de941735
4	516166d9e4b0d19a95e6b214	0.0	0.0	0.0	347900	0.0	1.365338e+09	4bf58dd8d48988d12b951735

- Join two datasets into a new one using venue id;

Category	Id	Rating	Likes	Tips	PostalCode	Price_tier	createdAt	Category_id
Hotel Pool	55a0e60c498e873de699ccd1	0.0	7.0	0.0	347900	0.0	1.436608e+09	4bf58dd8d48988d132951735
Supermarket	58401718e9233e42ab37615e	5.8	1.0	0.0	347900	0.0	1.480595e+09	52f2ab2ebcbc57f1066b8b46
Beach	51ecf0f2498edbbbb8c6c29d	0.0	0.0	0.0	347900	0.0	1.374483e+09	4bf58dd8d48988d1e2941735
Vineyard	53f917c4498e47b1dc40249e	0.0	1.0	0.0	347900	0.0	1.408833e+09	4bf58dd8d48988d1de941735
Bus Line	516166d9e4b0d19a95e6b214	0.0	0.0	0.0	347900	0.0	1.365338e+09	4bf58dd8d48988d12b951735

Now we have got the dataset with 3525 rows. Each row corresponds to a venue.

3. Methodology

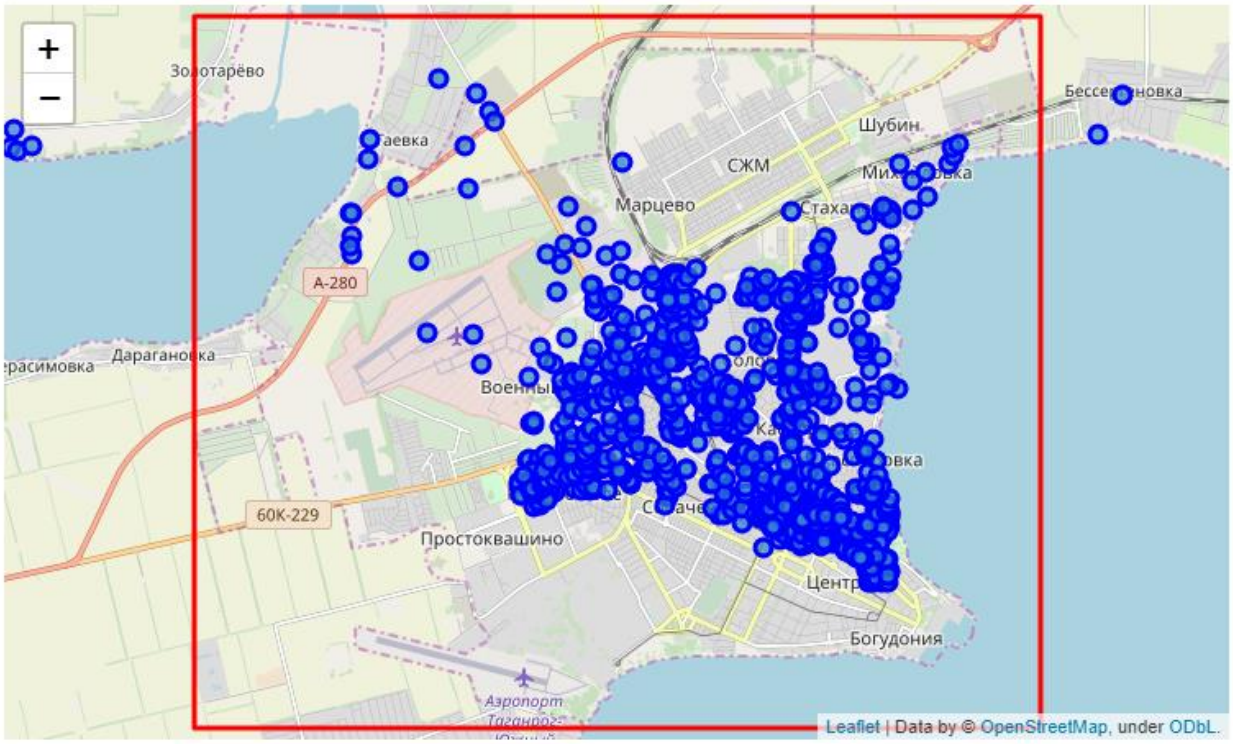
This section focuses on data understanding and data preparation of previously collected data, exploratory analysis, and clustering.

- Understanding and preparation: the data needs to be cleaned and prepared: fields renamed, NaN's fixed;
- Exploratory analysis includes check field distributions, detect any outliers, exploring correlation between fields and discarding erroneous or not needed data;
- Clustering -- the core Machine Learning methodology and a very popular clustering algorithm, called k-means is used in this project. The major focus of the project is to determine the optimum number of clusters and finding suitable cluster for a new restaurant.

Understanding and preparation

Display collected venues on map.

```
# for some reason, the max number of venues the folium can display is 1500
plot_venues_with_rect(city_venues.iloc[1500:3000], sw=city_sw, ne=city_ne)
```



We can see outliers on map i.e. there are a number of venues out of city bounding box. Remove them.

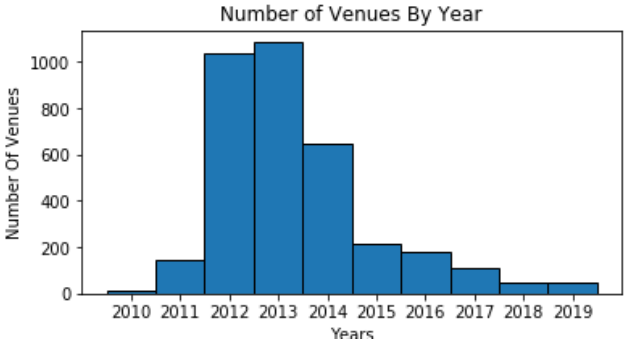
Exploratory Analysis

Let's examine the **createdAt** field. This field contains amount of seconds since epoch when the venue was created.

Let's convert timestamp in seconds to years and find a range of years and add it as columns named **Venue_age** and **Venue_year**.

Explore the **Created_year** field and plot its histogram.

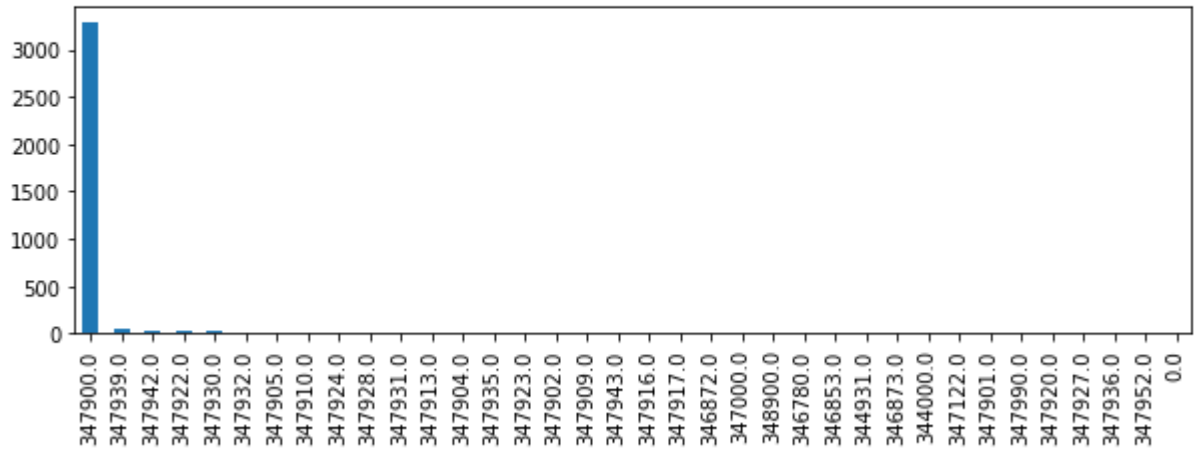
```
bins = np.arange(2010, 2021)
ax = city_venues.Created_year.plot(kind='hist', figsize=(6, 3), bins=bins, xticks=bins[:-1], edgecolor="k"

plt.title('Number of Venues By Year') # add a title to the histogram
plt.ylabel('Number Of Venues') # add y-label
plt.xlabel('Years') # add x-label

plt.show()
```



Now, let's examine the **PostalCode** field.

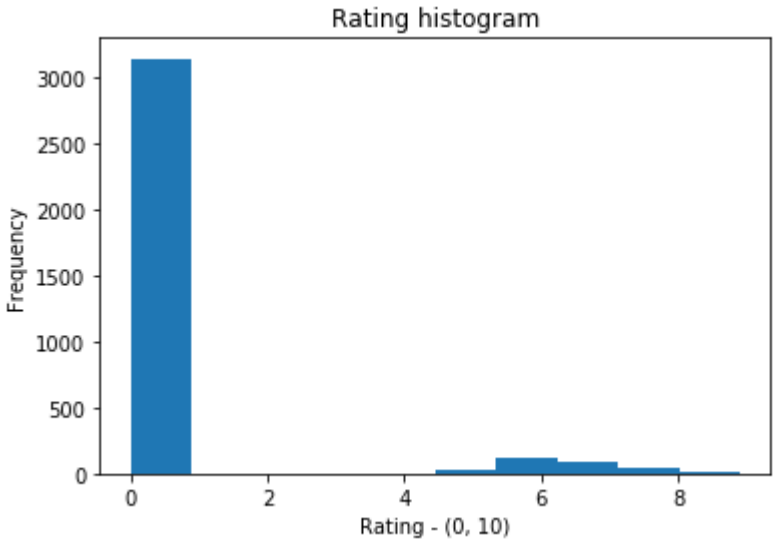
```
city_venues['PostalCode'].value_counts().plot(kind='bar', figsize=(10,3))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f138852ec8>
```



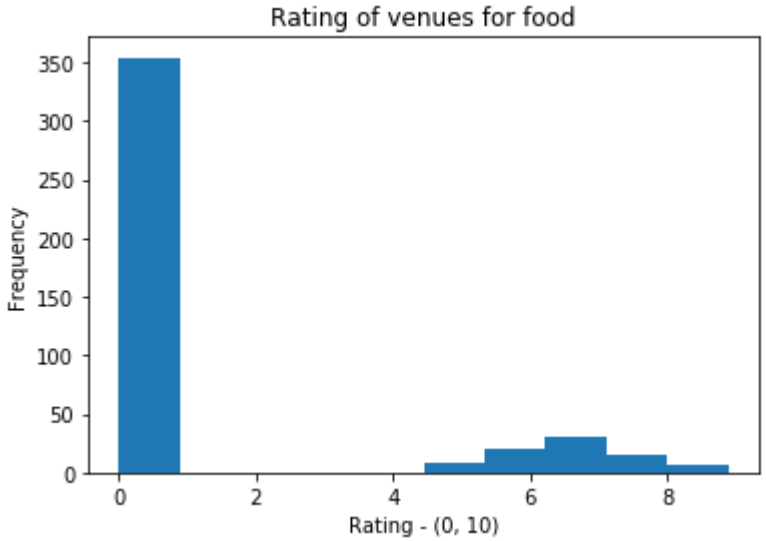
It is obvious that this column is useless because majority venues have postal code equal to 347900. Drop this column as well.

Next explore the **Rating** field and plot its histagram.



Here we see two seperate sets of venues -- one set with rating equal to 0 and second one greater than 0.

My hypothesis is that the second set consists of venues for food. Let's check it out and plot histogram of venues for food.



Correlation between fields:

```
city_venues.corr()
```

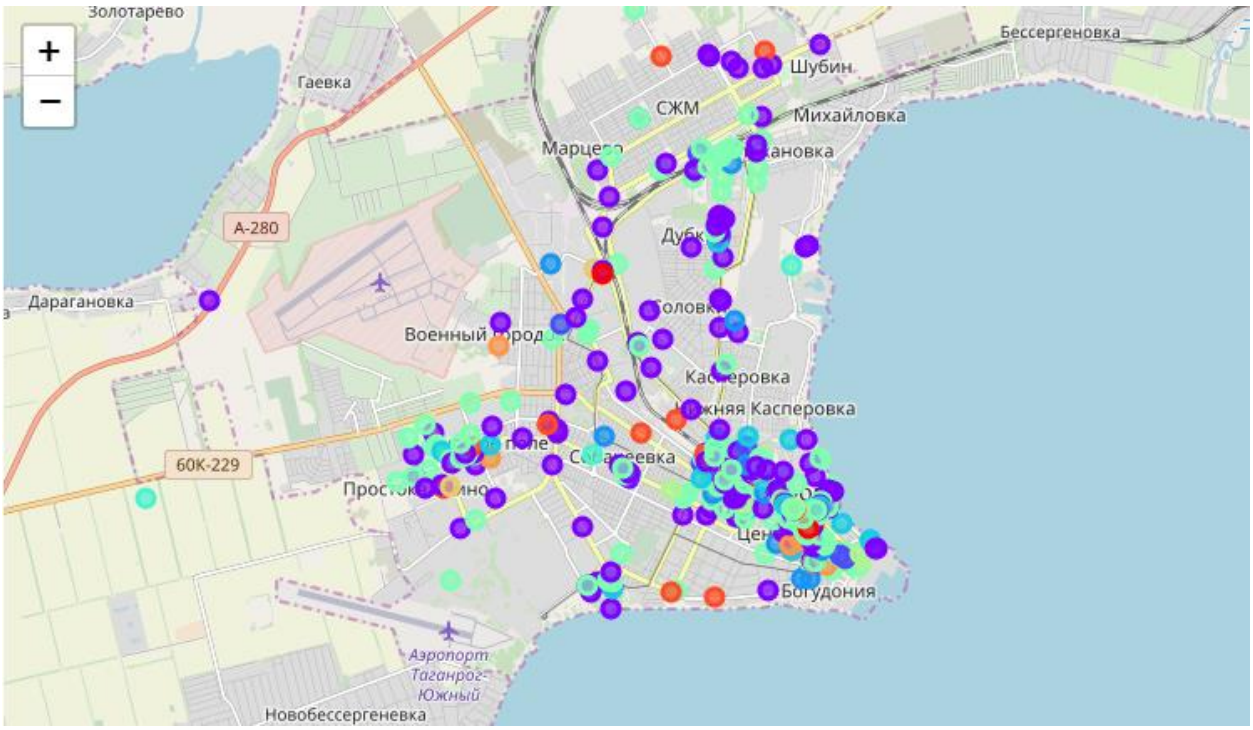
	Latitude	Longitude	Rating	Likes	Tips	Price_tier	Food_venue	Created_year	Venue_age
Latitude	1.000000	-0.115304	-0.045754	-0.061645	-0.062015	-0.054484	-0.076316	-0.026311	0.026311
Longitude	-0.115304	1.000000	0.055542	0.068591	0.080413	0.087981	0.088966	0.025289	-0.025289
Rating	-0.045754	0.055542	1.000000	0.495428	0.435916	0.153763	0.142334	-0.151191	0.151191
Likes	-0.061645	0.068591	0.495428	1.000000	0.871930	0.144708	0.128512	-0.128767	0.128767
Tips	-0.062015	0.080413	0.435916	0.871930	1.000000	0.203122	0.182537	-0.117383	0.117383
Price_tier	-0.054484	0.087981	0.153763	0.144708	0.203122	1.000000	0.673304	0.046388	-0.046388
Food_venue	-0.076316	0.088966	0.142334	0.128512	0.182537	0.673304	1.000000	0.095791	-0.095791
Created_year	-0.026311	0.025289	-0.151191	-0.128767	-0.117383	0.046388	0.095791	1.000000	-1.000000
Venue_age	0.026311	-0.025289	0.151191	0.128767	0.117383	-0.046388	-0.095791	-1.000000	1.000000

Summary. Almost all variables demonstrate very weak positive and negative correlations except those that obviously related to each other. For example Tips, Likes and Rating correlate between each other as well as Food_venue and Food_tier.

Clustering

The core of the analytic method is clustering of all venues in Taganrog city by its fields, determining the optimum number of clusters and find suitable cluster for a new restaurant.

1. Transform categorical field Category to dummies
2. Normalize the dataset
3. Run k-means to cluster all venues into 11 clusters
4. Assign labels to venues
5. Display our clustered venues on map



Now let's create the new dataframe **clusters_df** with the top 5 venue's category for each cluster.

```
clusters_df = createClustersRating(food_venues, num_clusters)
clusters_df
```

	Label	1st	2nd	3rd	4th	5th	Total
0	0	Dessert Shop	None	None	None	None	7
1	1	Restaurant	Sushi Restaurant	Pizza Place	Snack Place	BBQ Joint	162
2	2	Fast Food Restaurant	None	None	None	None	18
3	3	Diner	None	None	None	None	12
4	4	Eastern European Restaurant	Italian Restaurant	Restaurant	Japanese Restaurant	Café	27
5	5	Modern European Restaurant	None	None	None	None	5
6	6	Café	Cafeteria	Bistro	Caucasian Restaurant	Pastry Shop	146
7	7	Coffee Shop	None	None	None	None	23
8	8	Sandwich Place	None	None	None	None	9
9	9	Burger Joint	None	None	None	None	10
10	10	Bakery	None	None	None	None	15

4. Results

This section summarizes the results of clustering. Empirically we identified 11 clusters of venues; let us first see what these clusters are.

To help with interpreting the data, let's sort the clusters by a meaningful attribute. A good candidate for such attribute is total number of venues in each cluster.

```
clusters_df.sort_values('Total', ascending=False)
```

	Label	1st	2nd	3rd	4th	5th	Total
1	1	Restaurant	Sushi Restaurant	Pizza Place	Snack Place	BBQ Joint	162
6	6	Café	Cafeteria	Bistro	Caucasian Restaurant	Pastry Shop	146
4	4	Eastern European Restaurant	Italian Restaurant	Restaurant	Japanese Restaurant	Café	27
7	7	Coffee Shop	None	None	None	None	23
2	2	Fast Food Restaurant	None	None	None	None	18
10	10	Bakery	None	None	None	None	15
3	3	Diner	None	None	None	None	12
9	9	Burger Joint	None	None	None	None	10
8	8	Sandwich Place	None	None	None	None	9
0	0	Dessert Shop	None	None	None	None	7
5	5	Modern European Restaurant	None	None	None	None	5

In the table above the data is sorted in descending order, with the highest total number at the top.

Clusters 1 and 6 are the biggest among others and represented by such venue categories as Cafe, Cafeteria, Sushi, Pizza, Snack ect. It shows that the number of restaurants is high and it is risky to open a restaurant similar to clusters 1 and 6.

Cluster 4 contains of cuisines of different countries: Eastern European, Italian, Japanese.

The rest clusters have a small number of venues and introduced by a single category.

5. Discussion

One interesting observation is that small restaurants (such as Cafe, Cafeteria, Sushi, Pizza, etc.) are over big restaurants.

There is small number of restaurants of different cuisines and a recommendation might be to open a restaurant of national cuisine (for instance Indian Restaurant).

6. Conclusion

The project analyzes venue similarity based on data obtained from Foursquare.

The data needed cleaning as it contained nonexistent venues and missing data. We can observe a lack of restaurants of national cuisines and suitable cluster for opening a restaurant.

However, we wish Foursquare had returned more venues as the data was very sparse and difficult to use for clustering. In future work, it may also be useful to include data pertaining to schools, universities, and such.