Coursera Capstone

Opening a new sushi bar in Saint Petersburg, Russia

By: Pavel Kravetskiy

June 2019

Introduction/Business Problem

Sushi is traditional Japanese food is gaining great popularity for many people especially in **Saint Petersburg, Russia**. Saint Petersburg is Russia's second-largest city after Moscow, with about 5,4 million inhabitants in 2019 and with lots of business opportunities and business friendly environment. Since the number of sushi bars in Saint Petersburg is rather small, customers may be interested in opening additional sushi bars in the most favorable neighborhoods of the city. However, any new business venture or expansion in the country needs to be reviewed carefully and strategically targeted so that the return on investment will be sustainably reasonable and more importantly the investment can be considerably less risky. Particularly, the location of the sushi bar is one of the most important decisions that will determine whether the bar will be a success or a failure.

The objective of this capstone project is to analyze and select the best locations in the city of Saint Petersburg, Russia to open a new sushi bar. Using data science methodology and machine learning techniques like clustering, this project aims to provide solutions to answer the following business question: **Which neighborhoods would be a good choice for opening a new sushi bar in Saint Petersburg, Russia?**

Data description

To solve this problem, we will need the following data:

- List of neighbourhoods in Saint Petersburg.
- Latitude and longitude coordinates of those neighbourhoods. This is required in order to plot the map and also to get the venue data.
- Venue data, particularly data related to sushi bars. We will use this data to perform clustering on the neighbourhoods.

Unfortunately, the Saint Petersburg neighborhood data is not widely available on the Internet in the structured format, hence we need to scrap it through an existing Wikipedia page (https://en.wikipedia.org/wiki/Category:Districts_of_Saint_Petersburg) that has all the information we need to explore and cluster the neighborhoods in Saint Petersburg.

Then we will get the geographical coordinates of the neighbourhoods using Python Geocoder package which will give us the latitude and longitude coordinates of the neighbourhoods.

And after that, we will use Foursquare API to get the venue data for those neighbourhoods. Since Foursquare has one of the largest database used by many developers around the world, we will use it to get information about *Sushi Restaurant* category of the venue data in order to help us to solve the business problem put forward.

The data before feature engineering step looks like as follows:

	Neighborhood	Latitude	Longitude	VenueName	VenueLatitude	VenueLongitude	VenueCategory
0	Admiralteysky District	59.92659	30.3056	Булочная Ф. Вольчека	59.926702	30.307921	Bakery
1	Admiralteysky District	59.92659	30.3056	Chao, mama!	59.926993	30.308474	Hotel
2	Admiralteysky District	59.92659	30.3056	CUP IN CUP	59.928074	30.302705	Coffee Shop
3	Admiralteysky District	59.92659	30.3056	ЛУУК	59.926154	30.310403	Clothing Store
4	Admiralteysky District	59.92659	30.3056	Pacman	59.923537	30.307985	Hookah Bar

Methodology

1. Data preprocessing

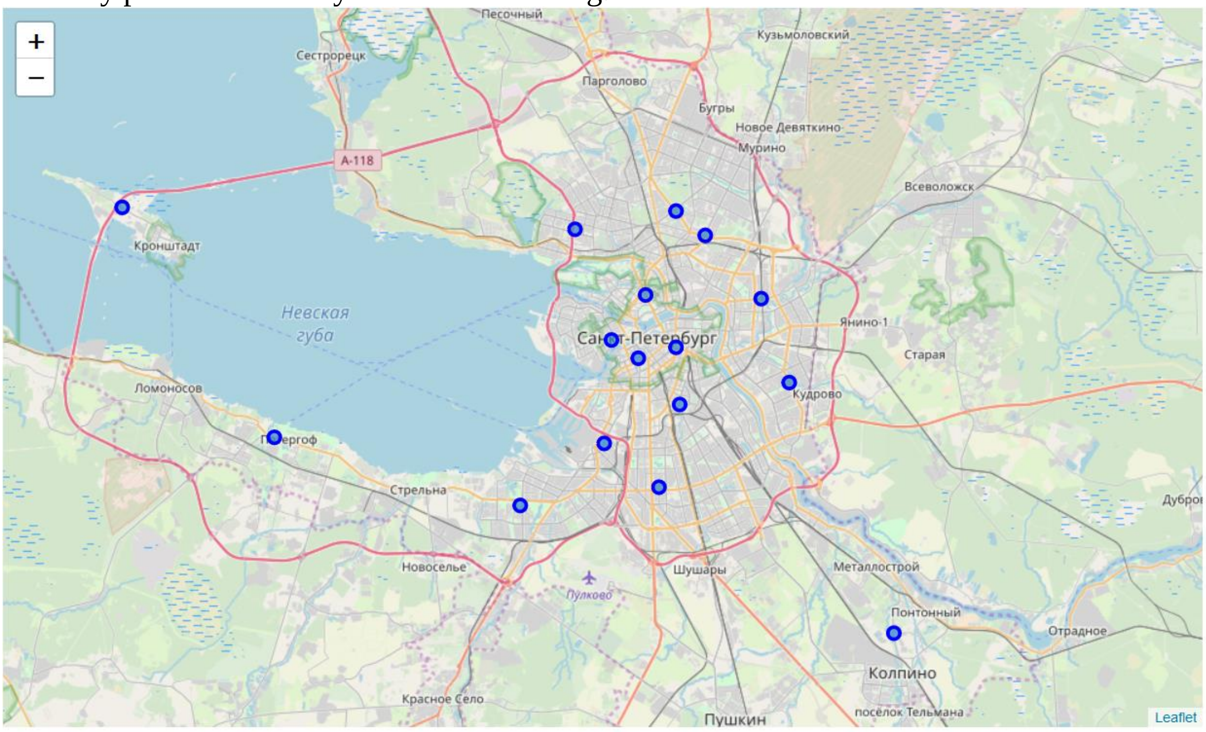
Firstly, we need to get the list of neighbourhoods in Saint Petersbug. The list is available in the Wikipedia page (https://en.wikipedia.org/wiki/Category:Districts_of_Saint_Petersburg). To extract the list of neighbourhoods we use Python *requests* and *beautifulsoup* packages. However, this is just a list of names. We sanitize the neighbourhood names by removing redundant data and striping unnecessary characters. After that we get the following list of neighbourhoods in Saint Petersburg:

	Neighborhood
0	Admiralteysky District
1	Frunzensky District
2	Kalininsky District
3	Kirovsky District
4	Krasnogvardeysky District
5	Krasnoselsky District
6	Kurortny District
7	Moskovsky District
8	Nevsky District
9	Petrodvortsovy District
10	Petrogradsky District
11	Primorsky District
12	Pushkinsky District
13	Tsentralny District
14	Vasileostrovsky District
15	Kolpinsky District
16	Kronshtadtsky District
17	Vyborgsky District

We need now to get the geographical coordinates in the form of latitude and longitude to be able to use Foursquare API. To do so, we will use the wonderful Geocoder package that will allow us to convert address into geographical coordinates in the form of latitude and longitude.

	Neighborhood	Latitude	Longitude
0	Admiralteysky District	59.92659	30.30560
1	Frunzensky District	59.90066	30.35211
2	Kalininsky District	59.99628	30.38081
3	Kirovsky District	59.87876	30.26721
4	Krasnogvardeysky District	59.96040	30.44418
5	Krasnoselsky District	59.84321	30.17219
6	Kurortny District	60.14784	30.01070
7	Moskovsky District	59.85352	30.32980
8	Nevsky District	59.91309	30.47637
9	Petrodvortsovy District	59.88201	29.89546
10	Petrogradsky District	59.96273	30.31452
11	Primorsky District	59.99968	30.23428
12	Pushkinsky District	59.71229	30.31000
13	Tsentralny District	59.93268	30.34810
14	Vasileostrovsky District	59.93703	30.27570
15	Kolpinsky District	59.77076	30.59402
16	Kronshtadtsky District	60.01211	29.72333
17	Vyborgsky District	60.01015	30.34806

After gathering the data, we will populate the data into a pandas DataFrame and then visualize the neighbourhoods in a map using Folium package. This allows us to perform a sanity check to make sure that the geographical coordinates data returned by Geocoder are correctly plotted in the city of Saint Petersburg.



Next, we will use Foursquare API to get the top 100 venues that are within a radius of 2000 meters. Foursquare will return the venue data in JSON format and we will extract the venue name, venue latitude, venue longitude and venue category so that our data looks like follows:

	Neighborhood	Latitude	Longitude	VenueName	VenueLatitude	VenueLongitude	VenueCategory
0	Admiralteysky District	59.92659	30.3056	Булочная Ф. Вольчека	59.926702	30.307921	Bakery
1	Admiralteysky District	59.92659	30.3056	Chao, mama!	59.926993	30.308474	Hotel
2	Admiralteysky District	59.92659	30.3056	CUP IN CUP	59.928074	30.302705	Coffee Shop
3	Admiralteysky District	59.92659	30.3056	ЛУУК	59.926154	30.310403	Clothing Store
4	Admiralteysky District	59.92659	30.3056	Pacman	59.923537	30.307985	Hookah Bar

2. Exploratory analysis

With the data, we can check how many venues were returned for each neighbourhood and examine how many unique categories can be curated from all the returned venues.

	Latitude	Longitude	VenueName	VenueLatitude	VenueLongitude	VenueCategory
Neighborhood						
Admiralteysky District	100	100	100	100	100	100
Frunzensky District	96	96	96	96	96	96
Kalininsky District	100	100	100	100	100	100
Kirovsky District	96	96	96	96	96	96
Krasnogvardeysky District	100	100	100	100	100	100
Krasnoselsky District	100	100	100	100	100	100
Kurortny District	18	18	18	18	18	18
Moskovsky District	100	100	100	100	100	100
Nevsky District	100	100	100	100	100	100
Petrodvortsovy District	100	100	100	100	100	100
Petrogradsky District	100	100	100	100	100	100
Primorsky District	100	100	100	100	100	100
Pushkinsky District	8	8	8	8	8	8
Tsentralny District	100	100	100	100	100	100
Vasileostrovsky District	100	100	100	100	100	100
Kolpinsky District	6	6	6	6	6	6
Kronshtadtsky District	15	15	15	15	15	15
Vyborgsky District	100	100	100	100	100	100

```
print('There are {} uniques categories.'.format(len(venues_df['VenueCategory'].unique())))
```

```
There are 251 uniques categories.
```

```
# print out the list of categories
venues_df['VenueCategory'].unique()[:20]
```

```
array(['Bakery', 'Hotel', 'Coffee Shop', 'Clothing Store', 'Hookah Bar',
       'Café', 'Bar', 'Opera House', 'Garden', 'Palace', 'Pizza Place',
       'Arcade', 'Park', 'Italian Restaurant', 'Plaza', 'Hostel',
       'Restaurant', 'Concert Hall', 'Music Venue', 'Historic Site'],
      dtype=object)
```

3. Clustering

Then, we will analyse each neighbourhood by grouping the rows by neighbourhood and taking the mean of the frequency of occurrence of each venue category. By doing so, we are also preparing the data for use in clustering. Since we are analysing the "Sushi Restaurant" data, we will filter the "Sushi Restaurant" as venue category for the neighbourhoods.

	Neighborhoods	Sushi Restaurant
0	Admiralteysky District	0.000000
1	Frunzensky District	0.010417
2	Kalininsky District	0.010000
3	Kirovsky District	0.020833
4	Krasnogvardeysky District	0.010000
5	Krasnoselsky District	0.020000
6	Kurortny District	0.000000
7	Moskovsky District	0.020000
8	Nevsky District	0.010000
9	Petrodvortsovy District	0.000000
10	Petrogradsky District	0.000000
11	Primorsky District	0.020000
12	Pushkinsky District	0.000000
13	Tsentralny District	0.010000
14	Vasileostrovsky District	0.000000
15	Kolpinsky District	0.000000
16	Kronshtadtsky District	0.000000
17	Vyborgsky District	0.010000

Lastly, we will perform clustering on the data by using k-means clustering. K-means clustering algorithm identifies k number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible. It is one of the simplest and popular unsupervised machine learning algorithms and is particularly suited to solve the problem for this project. We will cluster the neighbourhoods into 3 clusters based on their frequency of occurrence for "Sushi Restaurant".

The results will allow us to identify which neighbourhoods have higher concentration of sushi bars while which neighbourhoods have fewer number of sushi bars. Based on the occurrence of sushi bars in different neighbourhoods, it will help us to answer the question as to which neighbourhoods are most suitable to open a new sushi bar.

Results

After clustering the Saint Petersburg neighborhoods based on the results from the Foursquare API data, we were able to separate our dataset into 3 distinct clusters, and then from our target cluster pick the best candidates to open a new sushi bar.

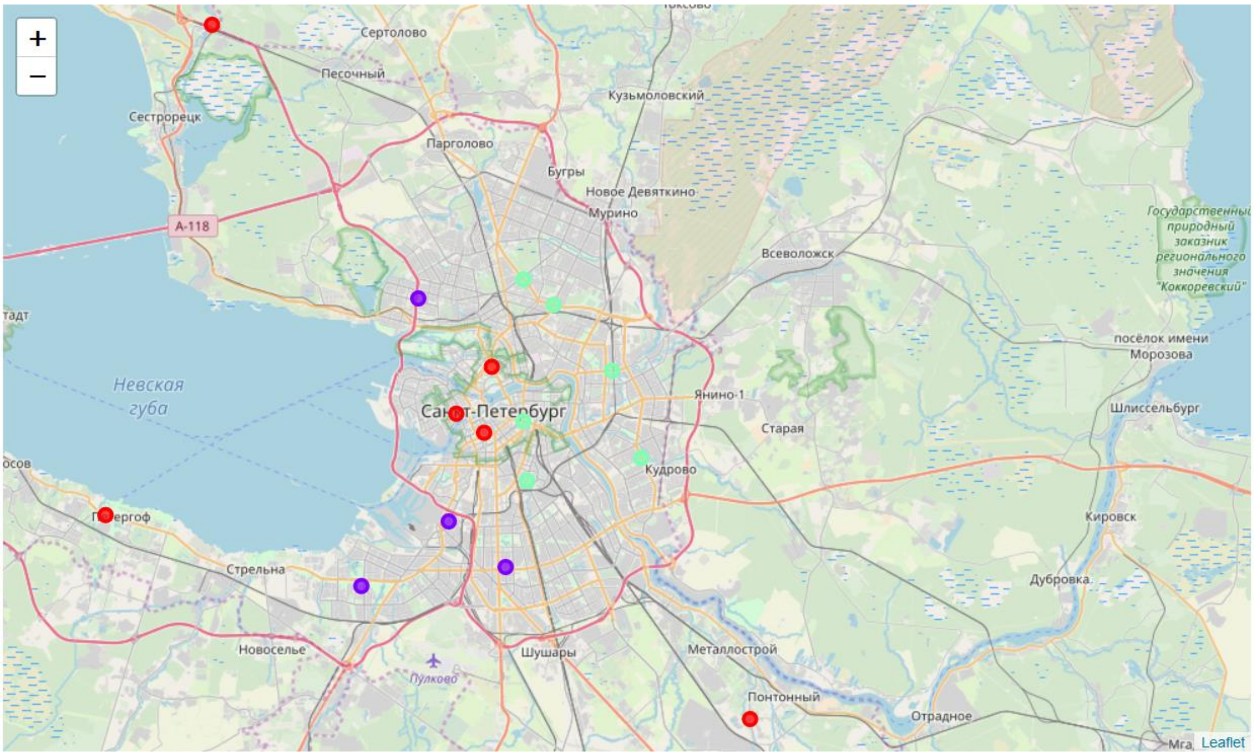
- Cluster 0: Neighbourhoods with low number to no existence of sushi bars
- Cluster 1: Neighbourhoods moderate number of sushi bars
- Cluster 2: Neighbourhoods with relatively high concentration of sushi bars

	Neighborhood	Latitude	Longitude	Sushi Restaurant	Cluster Labels
0	Admiralteysky District	59.92659	30.30560	0.000000	0
15	Kolpinsky District	59.77076	30.59402	0.000000	0
14	Vasileostrovsky District	59.93703	30.27570	0.000000	0
6	Kurortny District	60.14784	30.01070	0.000000	0
16	Kronshtadtsky District	60.01211	29.72333	0.000000	0
9	Petrodvortsovy District	59.88201	29.89546	0.000000	0
10	Petrogradsky District	59.96273	30.31452	0.000000	0
12	Pushkinsky District	59.71229	30.31000	0.000000	0
3	Kirovsky District	59.87876	30.26721	0.020833	1
5	Krasnoselsky District	59.84321	30.17219	0.020000	1
7	Moskovsky District	59.85352	30.32980	0.020000	1
11	Primorsky District	59.99968	30.23428	0.020000	1
13	Tsentralny District	59.93268	30.34810	0.010000	2
8	Nevsky District	59.91309	30.47637	0.010000	2
2	Kalininsky District	59.99628	30.38081	0.010000	2
1	Frunzensky District	59.90066	30.35211	0.010417	2
4	Krasnogvardeysky District	59.96040	30.44418	0.010000	2
17	Vyborgsky District	60.01015	30.34806	0.010000	2

According our clustering the best candidates are:

- Kolpinsky District
- Vasileostrovsky District
- Kurortny District
- Kronshtadtsky District
- Petrodvortsovy District
- Petrogradsky District
- Pushkinsky District

We can also visualize the clusters on the map where the green points correspond to the Cluster 0, the red points correspond to the Cluster 1 and the blue ones to the Cluster 2.



Discussion

As observations noted from the map in the Results section, most of the sushi bars are concentrated around the central area of Saint Petersburg city, with the highest number in cluster 2 and moderate number in cluster 1. On the other hand, cluster 0 has very low number to no sushi bars in the neighbourhoods. This represents a great opportunity and high potential areas to open a new sushi bar as there is very little to no competition from existing bars. Therefore, this project recommends property developers to capitalize on these findings to open new sushi bars in neighbourhoods in cluster 0 with little to no competition. You can also open in neighbourhoods in cluster 1 with moderate competition if you have unique selling propositions to stand out from the competition.

However, for a real-life project, probably additional metrics should be added to create a more robust clustering.

Conclusion

In this project, we have gone through the process of identifying the business problem, specifying the data required, extracting and preparing the data, performing machine learning by clustering the data into 3 clusters based on their similarities, and lastly providing recommendations to the relevant stakeholders i.e. property developers and investors regarding the best locations to open a new sushi bar. The findings of this project will help the relevant stakeholders to capitalize on the opportunities on high potential locations while avoiding overcrowded areas in their decisions to open a new sushi bar in Saint Petersburg, Russia.

For future projects with similar characteristics, it should be considered to expand the amount of data available (for example, using the premium features of the Foursquare API) and other clustering algorithms such as DBSCAN.