Capstone Project. New place of residence

Table Of Contents:

1. Introduction: Business Problem

In this project, we will try to find the best place to live for a young person who has decided to change jobs and move to Moscow from St. Petersburg. This report will be of interest to anyone who wants to change their place of residence to Moscow.

We will assume that the current place of residence suits the person in the part of the various venues around it and he wants the new place of residence to be as similar as possible to the current one. He also wants the new location will be within 500m of any metro station. Venues that are important to him and that he takes into account when evaluating a new place of residence:

- Restaurants
- Bars
- Cafes
- Fitness centers
- Clubs
- Pharmacies
- Movie theaters
- Pools
- Shopping Malls
- Supermarkets
- Various stores

Using data science, we will determine the most suitable place of residence for a young person, taking into account the above criteria.

2. Description of data sources

The introductory information for our project will be the address of a young person in St. Petersburg, as well as the address of his new place of work in Moscow. We will use the Google Maps API to get the coordinates of these addresses. And using the Foursquare API, we will get a list of the necessary venues around the current place of residence.

To solve this problem, we will need a list of metro stations with their coordinates. We will take it here list of metro stations. Using the Foursquare API, we will get a list of the necessary venues within a radius of 500m around each metro station.

We will search for the area most similar to the current place of residence. All other things being equal, the area closest to the address of the place of work will be finally recommended.

3. Methodology

3.1 Data Collection, Understanding, Preparation

The input data for the project are:

- the address of a young person in St. Petersburg
- the address of his new place of work in Moscow

If the project will be embedded somewhere, you can get this data using the input () function. But here we'll just set them as string variables.

1. The address in St. Petersburg
   "проспект Стачек, 92к2, Санкт-Петербург, Россия"

2. The address in Moscow
   "Ленинский проспект, 69, Москва, Россия"

For the project, we need a list of metro stations. In Moscow, in addition to the metro, there is the Moscow Central Circle (MCC). It is equivalent to the metro, because trains run at the same frequency as the metro. In addition, there are pedestrian crossings between the stations, and one ticket is valid. We will take the data from Wikipedia List of Moscow Metro stations. There are several tables there. We will need:

- the Stations of the Moscow Metro
- the Platforms of the Moscow Central Ring

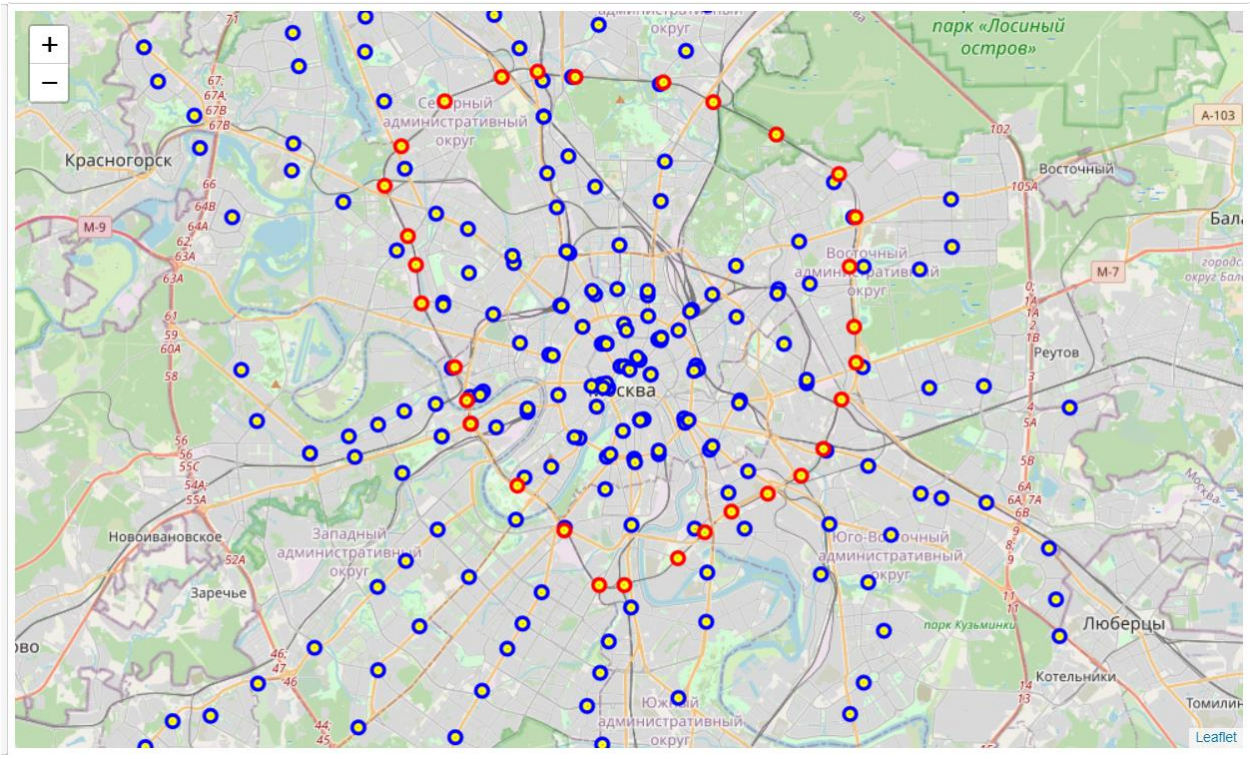
Load the data into two dataframes. Since the address bar contains Russian characters, we need to pre-process the URL using the function. Example of uploaded data:

	Unnamed: 0	Название станцииПрежние названия	Датаоткрытия	Пере-садки	Глубина,м[2]	Тип конструкции	Координаты	Вид станции
0	NaN	Бульвар РокоссовскогоУлица Подбельского (до 2014)	1 августа 1990	NaN	−8	колоннаямелкого заложениятрёхпролётная	55°48′53″ с. ш. 37°44′03″ в. д.НЯ	NaN
1	NaN	Черкизовская	1 августа 1990	NaN	−9	односводчатаямелкого заложения	55°48′14″ с. ш. 37°44′41″ в. д.НЯ	NaN
2	NaN	Преображенская площадь	31 декабря 1965	NaN	−8	колоннаямелкого заложениятрёхпролётная	55°47′47″ с. ш. 37°42′54″ в. д.НЯ	NaN
3	NaN	Сокольники	15 мая 1935	NaN	−9	колоннаямелкого заложениятрёхпролётная	55°47′20″ с. ш. 37°40′49″ в. д.НЯ	NaN
4	NaN	Красносельская	15 мая 1935	NaN	−8	колоннаямелкого заложениядвухпролётная	55°46′48″ с. ш. 37°40′02″ в. д.НЯ	NaN

Some metro stations are now closed. We need to delete these records. In the uploaded data, the coordinates of the stations are represented as degrees, hours, and minutes. We need to convert them to decimal coordinates. To do this, we will use the function, write the results to the new columns of the dataframe, and convert the data to the type "float". If a metro station has changed its name, its name is loaded in the "New nameOld name" format. So we have to cut off the old name using the function. Next, we need to collect the necessary attributes in a single dataframe and delete the missing data. To solve our problem, we will use clustering. Therefore, we will also add a location in St. Petersburg to the dataframe. But at first We need use the Google Maps API to get the coordinates of the address. Since the names of the stations can be duplicated, we will assign each station its own ID. After rebuilding the index, we got this dataframe

	ID_Station	Type	Metro_name	Location	Latitude	Longitude
0	0	Metro	Бульвар Рокоссовского	[55.81472222222222, 37.73416666666667]	55.814722	37.734167
1	1	Metro	Черкизовская	[55.803888888888885, 37.74472222222222]	55.803889	37.744722
2	2	Metro	Преображенская площадь	[55.796388888888885, 37.715]	55.796389	37.715000
3	3	Metro	Сокольники	[55.788888888888884, 37.680277777777775]	55.788889	37.680278
4	4	Metro	Красносельская	[55.78, 37.66722222222222]	55.780000	37.667222

Now we can display the Moscow metro and MCC stations on the map. We will display metro stations in blue, and MCC stations in red. We will not display the location in St. Petersburg. The center of the map will be the center of Moscow.



Now let's use the Foursquare API to get information about venues within a 500m radius of metro stations. We collected preliminary data from Foursquare in a dataframe metro_venues_prev. We are interested in the following categories of venues:

- Restaurants
- Bars
- Cafes
- Fitness centers
- Clubs
- Pharmacies
- Movie theaters
- Pools
- Shopping Malls
- Supermarkets
- Various stores

We will collect the necessary venues in the dataframe metro_venues. We will add entries for each condition.

1. To get restaurants, we will filter the categories that contain "restaurant" and "Restaurant"
2. To get bars, we will filter the categories that contain whole word "Bar"
3. To get Cafes, we will filter the categories that contain "Cafe", "Bakery" or "Coffee"
4. To get Fitness centers, we will filter the categories that contain "Gym"
5. To get Clubs, we will filter the categories that contain "Club"
6. To get Pharmacies, we will filter the categories that contain "Drugstore" or "Pharmacy"
7. To get Movie theaters, we will filter the categories that contain "Movie"
8. To get pools, it is necessary that the category is equal "Pool"
9. To get Shopping Malls, we will filter the categories that contain "Mall"
10. To get Supermarkets, we will filter the categories that contain "Supermarket"
11. To get Various stores, we will filter the categories that contain "store" or "Store"

Now we have collected the final information for our analysis:

	ID_Station	categories	shortName	lat	lng	name
3	5	Varenyky restaurant	Restaurant	55.774982	37.656780	Вареничная №1
2	90	Varenyky restaurant	Restaurant	55.774982	37.656780	Вареничная №1
3	0	American Restaurant	Restaurant	55.814026	37.733659	Бургер кинг
4	0	Vietnamese Restaurant	Restaurant	55.815955	37.736421	Фо & Ролл
6	0	Tex-Mex Restaurant	Restaurant	55.815503	37.737244	El Taco

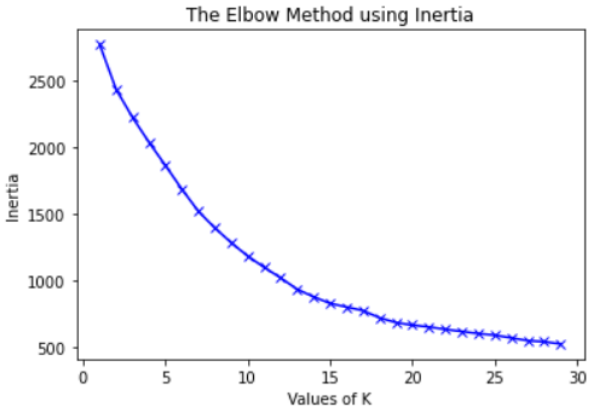
3.2 Modeling and Deployment: Analyze Each Metro Station

Let's make categorical columns using the function get_dummies, group rows by ID Metro Station and take the mean of the frequency of occurrence of each category

	ID_Station	Bar	Cafe	Club	Fitness center	Mall	Movie	Pharmacy	Pool	Restaurant	Store	Supermarket
0	0	0.000000	0.071429	0.000000	0.000000	0.000000	0.0	0.142857	0.000000	0.428571	0.357143	0.000000
1	1	0.272727	0.090909	0.090909	0.181818	0.000000	0.0	0.000000	0.000000	0.090909	0.181818	0.090909
2	2	0.000000	0.217391	0.000000	0.130435	0.000000	0.0	0.173913	0.000000	0.130435	0.304348	0.043478
3	3	0.125000	0.000000	0.000000	0.125000	0.000000	0.0	0.000000	0.000000	0.250000	0.500000	0.000000
4	4	0.125000	0.250000	0.000000	0.125000	0.000000	0.0	0.125000	0.000000	0.125000	0.125000	0.125000
...	...	...	...	...	...	...	...	...	...	...	...	...
247	256	0.000000	0.200000	0.000000	0.400000	0.000000	0.0	0.000000	0.000000	0.000000	0.400000	0.000000
248	257	0.000000	0.156863	0.000000	0.019608	0.039216	0.0	0.019608	0.000000	0.137255	0.607843	0.019608
249	258	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.200000	0.000000	0.200000	0.200000	0.400000
250	259	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.333333	0.333333	0.000000	0.333333	0.000000
251	260	0.000000	0.250000	0.000000	0.000000	0.000000	0.0	0.125000	0.000000	0.125000	0.500000	0.000000

Cluster Metro Stations

We will try to find the most similar place to stay when moving from St. Petersburg. So let's try segmenting the areas around metro stations using K-Means, using the elbow method, to find the Areas most similar to the area in St. Petersburg. To determine the optimal number of clusters, we will use the elbow visualization.

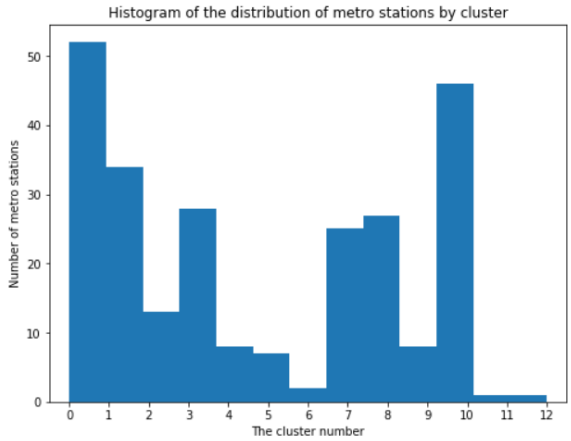


The graph shows that the optimal number of clusters is 13. Since a further increase in the number of clusters does not lead to a significant improvement in the accuracy of the model. Then we will add the cluster to the data frame with the average frequency of occurrence of each category and to a copy of the data frame with the coordinates of the metro stations.

	ID_Station	Type	Metro_name	Location	Latitude	Longitude	Cluster Labels
0	0	Metro	Бульвар Рокоссовского	[55.81472222222222, 37.73416666666667]	55.814722	37.734167	0
1	1	Metro	Черкизовская	[55.803888888888885, 37.74472222222222]	55.803889	37.744722	2
2	2	Metro	Преображенская площадь	[55.796388888888885, 37.715]	55.796389	37.715000	1
3	3	Metro	Сокольники	[55.788888888888884, 37.680277777777775]	55.788889	37.680278	10
4	4	Metro	Красносельская	[55.78, 37.66722222222222]	55.780000	37.667222	3
...	...	...	...	...	...	...	...
247	256	MCC	Стрешнево	[55.81361111111111, 37.48694444444445]	55.813611	37.486944	2
248	257	MCC	Балтийская	[55.825833333333335, 37.49611111111111]	55.825833	37.496111	10
249	258	MCC	Коптево	[55.83972222222222, 37.519999999999996]	55.839722	37.520000	9
250	259	MCC	Лихоборы	[55.84722222222222, 37.55138888888889]	55.847222	37.551389	1
251	260	S-Pt	St. Petersburg	[59.862909, 30.2602144]	59.862909	30.260214	10

Visualize the resulting clusters

Let's display the distribution of metro stations by cluster on the histogram.



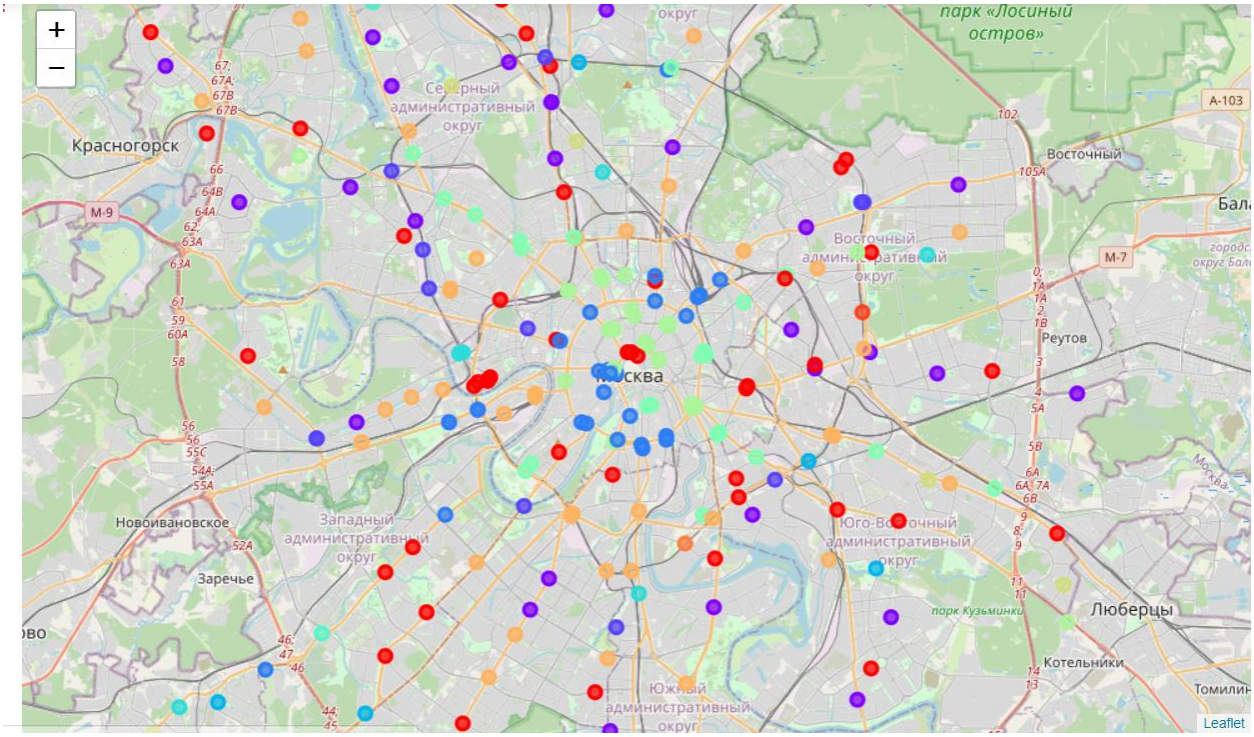
The histogram shows that the distribution of metro stations to clusters is uneven. Let's now look at this distribution on the map.

Saint-Petersburg map

Let's display which cluster the address in St. Petersburg is in.

Moscow map

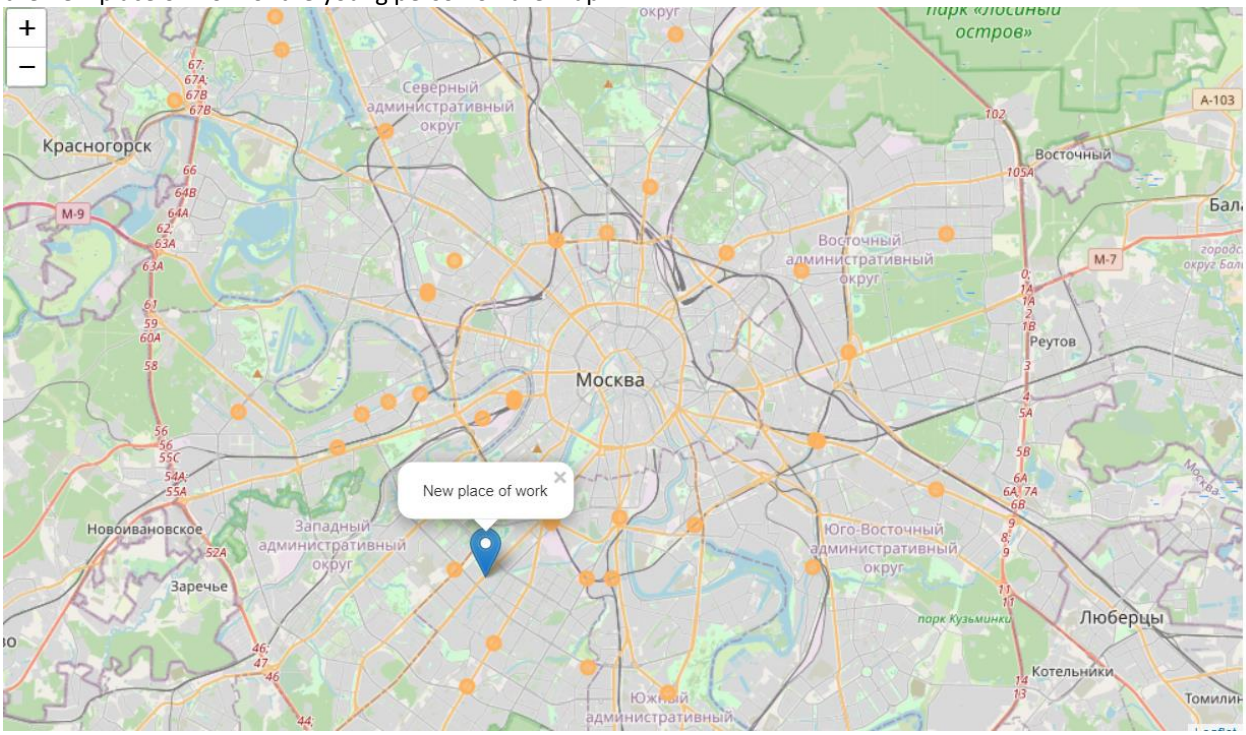
Let's display which clusters the Moscow metro stations are located in.



4. Results

Now we can find which locations are most similar to a person's current place of residence. The cluster that was defined for the address in St. Petersburg is 10.

Let's display on the map of Moscow the metro stations that have been identified in cluster 10. We will also display the new place of work of the young person on the map.



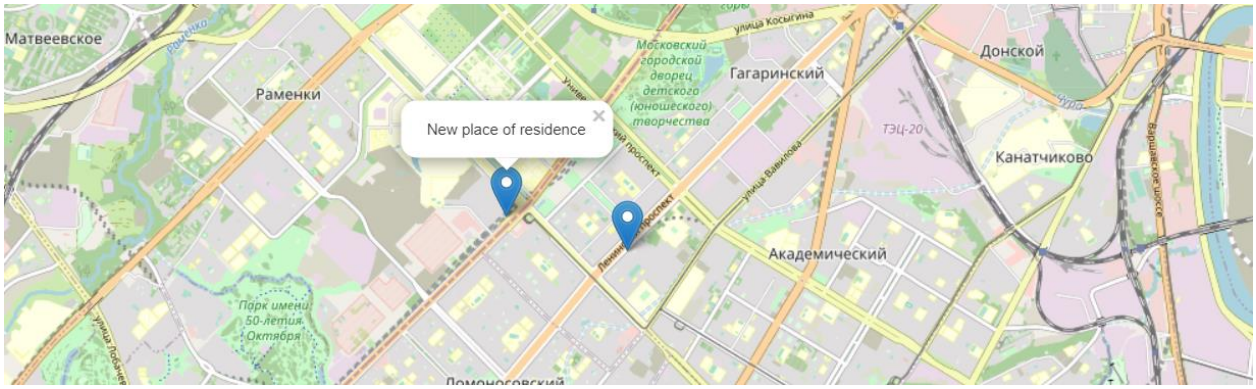
Now, from the selected locations, we need to select the most suitable one. The criterion will be the proximity to the new place of work. To do this, we need to calculate the distance from each metro station the new place of work. Let's assemble the metro stations from cluster 10 into a separate dataframe

	ID_Station	Type	Metro_name	Location	Latitude	Longitude	Cluster Labels
0	3	Metro	Сокольники	[55.788888888888884, 37.680277777777775]	55.788889	37.680278	10
1	16	Metro	Университет	[55.692499999999995, 37.53333333333333]	55.692500	37.533333	10
2	43	Metro	Каширская	[55.655, 37.64861111111111]	55.655000	37.648611	10
3	47	Metro	Домодедовская	[55.61083333333333, 37.718611111111116]	55.610833	37.718611	10
4	52	Metro	Волоколамская	[55.83527777777778, 37.382222222222225]	55.835278	37.382222	10
5	56	Metro	Молодёжная	[55.740833333333335, 37.416666666666664]	55.740833	37.416667	10
6	58	Metro	Славянский бульвар	[55.72972222222222, 37.470555555555556]	55.729722	37.470556	10
7	60	Metro	Киевская	[55.7444444444445, 37.565555555555555]	55.744444	37.565556	10

In order to calculate the distance from the new place of work to the metro station, we first need to translate the longitude and latitude into XY-coordinates, find the coordinates of a new place of work in Moscow. And then calculate the distance. Adding XY-coordinates and the distance to a separate columns of the new dataframe. Let's find the metro station with the shortest distance to the young person's new place of work:

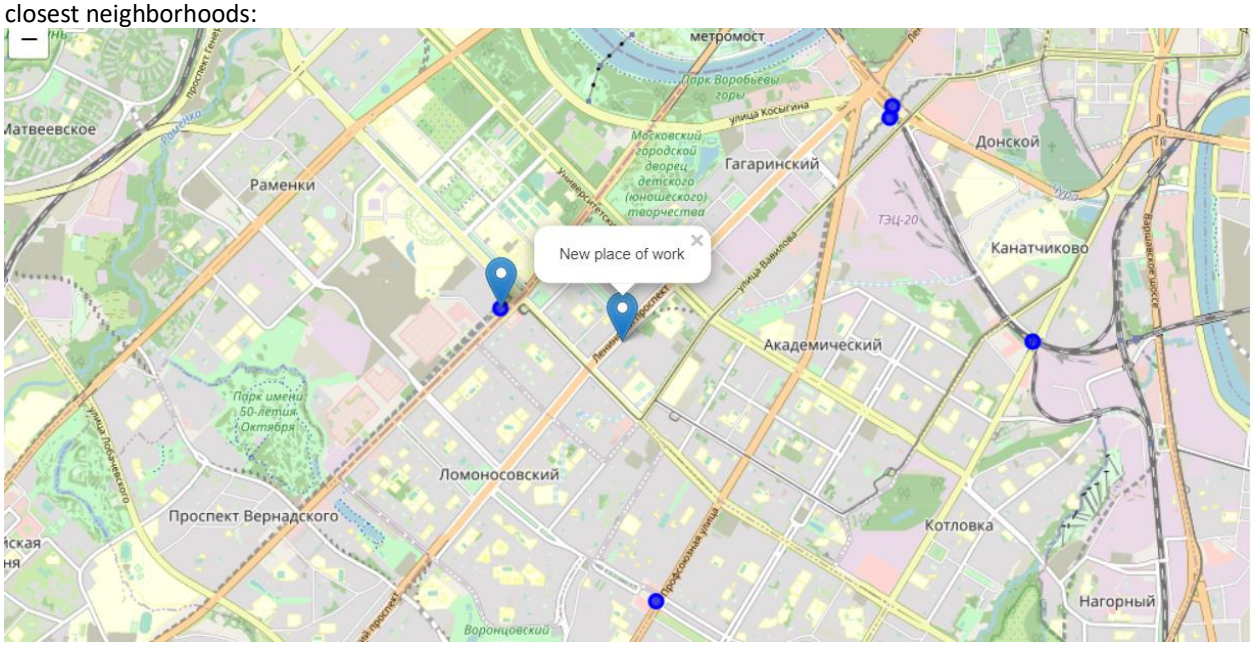
	ID_Station	Type	Metro_name	Location	Latitude	Longitude	Cluster Labels	XY	dist
1	16	Metro	Университет	[55.692499999999995, 37.53333333333333]	55.6925	37.533333	10	[1902364.2229705716, 6404546.192221226]	1094.192002

Let's display this answer on the map:



5. Discussion

Let's look at what other stations are close to the new place of work and suitable for living. Let's display the 5 closest neighborhoods:



You can see that the 2nd, 3rd, 4th, and 5th districts are about the same distance from the new place of work. Therefore, we can recommend them if for some reason the 1st option is not suitable. In total, the appropriate metro stations are:

- University
- New Cheryomushki
- Gagarin Square
- Leninsky Prospekt
- Crimean

This project could be complicated in terms of the selection criteria. For example, you could add type of venues that do not exist near the current place of residence of a young person in St. Petersburg, but that he would like to have in Moscow near the new place of residence.

6. Conclusion

In this project, we tried to find the most suitable place of residence for a young person who decided to change jobs and move from St. Petersburg to Moscow. The selection criteria were:

- similarity of the new area to the current place of residence
- proximity to the metro station. The similarity criteria were the presence of the following venues within walking distance:
- Restaurants
- Bars
- Cafes
- Fitness centers
- Clubs
- Pharmacies
- Movie theaters
- Pools
- Shopping Malls
- Supermarkets
- Various stores

The most suitable new place of residence for a young person is the metro area "University". The distance from this metro station to the new place of work is about 1 km.