Capstone Project
IBM Applied Data Science Specialization

The Battle of Neighborhoods in Europe: Choosing A City For Relocation

By Sergei Perfilyev
June 2021

1. INTRODUCTION

This project aims to solve the following problem:

A hypothetical family from Russia is planning to relocate to some place in the European Union (EU). They are aiming at broadening their career opportunities (as for the parents) and, most importantly, getting access to high-quality education for the children, so that the kids could study at top ranked European universities. In addition, to feel less uncomfortable while adjusting to living abroad and getting accustomed to foreign culture as well as potential changes of work/life balance, they would prefer to live in a city that resembles their hometown.

The family currently resides in Novosibirsk, which is the third-largest city in Russia with a population of over 1.5 million people. The city is one of the major business and transport hubs. A substantial number of large companies' headquarters are located here as well as an international airport, a river port and a railway station. Novosibirsk has many parks, open spaces, and squares throughout the city. Attractions and entertainment facilities include museums, theaters, concert halls, a water park and several large movie theaters. Novosibirsk is known for its shopping malls, numerous restaurants, bars and coffee shops. So, our friends are willing to know which of European cities tend to be similar to their own hometown in everyday life.

In addition, the family would like to include in the analysis another Russian city, Irkutsk, which is located in Western Siberia. With a population of 600 thousand people, it is more provincial and cozy, whereas it is also a major cultural center with some famous museums, theaters, concert halls and natural attractions. Our friends used to live in Irkutsk ten years ago and consider it their "second hometown".

Certainly, it would be a very complex task for them to gather all the information manually and perform such a comparison for each target city with both hometowns. Therefore, we will apply data science techniques to analyze data on European cities and help our friends compare all available options in order to choose the most appropriate and comfortable city for their relocation.

The outcome of this project may be interesting (with some adjustments, perhaps) to any person or family, who is considering options to relocate from their hometown to a new place either within their native country or in a foreign country or another part of the world. Therefore, we argue that lots of people in today's globalized world actually form target audience for this problem.

2. DATA

We intend to use the following data in order to implement this project as well as for learning purpose:

2.1. Eurostat open data website https://ec.europa.eu/eurostat

Eurostat is the statistical office of the EU whose mission is to provide high quality statistics and data on Europe. For example, we can obtain current, up-to-date information on member countries of the EU from this page:

https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Glossary:Country_codes

Member States of the European Union (EU) and other countries have been assigned a two-letter **country code**, always written in capital letters, and often used as an abbreviation in statistical analyses, tables, figures or maps.

The **protocol order** in which countries are often listed is based on the alphabetical list of countries in their national language for EU and EFTA Member States and for candidate countries; for potential candidates, it is based on the alphabetical order of their country code.

EU Member States come first, followed by European Free Trade Association (EFTA) Member States, candidate countries for EU membership, potential candidates and, finally, other countries. The order in the tables below is first column down, then second column down, etc..

European Union (EU)

Belgium	(BE)	Greece	(EL)	Lithuania	(LT)	Portugal	(PT)
Bulgaria	(BG)	Spain	(ES)	Luxembourg	(LU)	Romania	(RO)
Czechia	(CZ)	France	(FR)	Hungary	(HU)	Slovenia	(SI)
Denmark	(DK)	Croatia	(HR)	Malta	(MT)	Slovakia	(SK)
Germany	(DE)	Italy	(IT)	Netherlands	(NL)	Finland	(FI)
Estonia	(EE)	Cyprus	(CY)	Austria	(AT)	Sweden	(SE)
Ireland	(IE)	Latvia	(LV)	Poland	(PL)		

Specifically, we will parse the table of the EU member countries and create a dataset containing country names and their two-letter codes. The list of countries will define the scope of our analysis: we will select data on cities and universities located in those countries only.

2.2. Times Higher Education website https://www.timeshighereducation.com

This site contains a lot of statistics, articles and other useful data on higher education in the world including various university rankings. In particular, we will be using the `Best Universities in Europe` ranking:

https://www.timeshighereducation.com/student/best-universities/best-universities-europe

The page contains summary in a tabular format, and hyperlinks to more detailed pages for each institution, where we can find and parse all information necessary for the project, including location data with complete address of the university. Below is an excerpt from the webpage:

Best universities in Europe 2021: the results in full

Click on each institution to view its World University Rankings 2021 result

World University Rank 2021	Europe Rank 2021	University	Country
1	1	University of Oxford	United Kingdom
6	2	University of Cambridge	United Kingdom
11	3	Imperial College London	United Kingdom
14	4	ETH Zurich	Switzerland
16	5	UCL	United Kingdom
27	6	London School of Economics and Political Science	United Kingdom
30	7	University of Edinburgh	United Kingdom
32	8	LMU Munich	Germany
35	9	King's College London	United Kingdom
=36	10	Karolinska Institute	Sweden

In order to create a list of the universities, we will parse this webpage, then also download and parse all the detailed pages for each universities. The resulting dataset should contain the following columns:

- Name of the university;
- Rank of the university according to the Times Higher Education list;
- Country;
- Complete address (will use it later to obtain geographical coordinates).

For further analysis, we will join this dataset with the list of EU member countries and then select only the highest-ranked universities.

2.3. Wikipedia

In order to analyze cities in the EU countries, we will parse the Wikipedia page `List of cities in the European Union by population within city limits`:

https://en.wikipedia.org/wiki/List_of_cities_in_the_European_Union_by_population_within_city_limits

The tabular data from the webpage will help us create a list of potential places to relocate, and then rank them by population and link to the previously obtained data on educational institutions.

Cities by population within the city boundary [edit]

Cities in bold are capital cities of their respective countries.

Rank	City	Member State	Official population	Date of census	Reference	Photography
1	**Berlin**	Germany	3,669,495	31 December 2019	[1]	
2	**Madrid**	Spain	3,348,536	1 February 2020	[2]	
3	**Rome**	Italy	2,856,133	31 December 2018	[3]	
4	**Bucharest**	Romania	2,155,240	1 July 2020	[4]	

Our dataset of cities will contain the following columns:

- City;
- Country;
- Population.

2.4. Foursquare API

We have created a developer account on https://developer.foursquare.com and will be using the API to collect data on recommended venues in the cities of interest. Then we will analyze distribution of the venues in order to cluster the cities into groups and find the most suitable ones for relocation. We will request data from the Foursquare platform via the `explore` endpoint of the API. It will return a list of recommended venues near each city we want to analyze. The endpoint accepts a set of input parameters, below is the subset appropriate for our task:

Name	Example	Description
ll	40.74224, -73.99386	Latitude and longitude of the user's location.
radius	250	Radius to search within, in meters.
limit	10	Number of results to return, up to 50.
offset	20	Used to page through results, up to 50.

The endpoint will return a JSON object containing numerous fields, so we will parse it and extract data on venues recommendations and venues' attributes meaningful for our project:

Field	Description
totalResults	Total number of recommended venues.
groups	An array of objects representing groups of recommendations. Each group contains a `type` such as "recommended" a human-readable (eventually localized) `name` such as "Recommended Places," and an array `items` of recommendation objects.
venue.name	The best known name for this venue.
venue.location	An object containing none, some, or all of `address` (street address), `crossStreet`, `city`, `state`, `postalCode`, `country`, `lat`, `lng`, and `distance`. All fields are strings, except for `lat`, `lng`, and `distance`.
venue.categories	An array, possibly empty, of categories that have been applied to this venue.

Listed below is an example of a data segment for the city of Lisbon, which was parsed as `response['groups'][0]['items']`:


```
[{'reasons': {'count': 0,
   'items': [{'summary': 'This spot is popular',
     'type': 'general',
     'reasonName': 'globalInteractionReason'}]},
  'venue': {'id': '54fcbf1a498ec6204cad12b7',
   'name': 'Hotel H10 Duque de Loulé',
   'location': {'address': 'Av. Duque de Loulé, 81',
    'lat': 38.726364248881325,
    'lng': -9.147197881306065,
    'labeledLatLngs': [{'label': 'display',
      'lat': 38.726364248881325,
      'lng': -9.147197881306065}],
    'distance': 199,
    'postalCode': '1050-088',
    'cc': 'PT',
    'city': 'Lisboa',
    'state': 'Lisboa',
    'country': 'Portugal',
    'formattedAddress': ['Av. Duque de Loulé, 81',
     '1050-088 Lisboa',
     'Portugal']},
   'categories': [{'id': '4bf58dd8d48988d1fa931735',
     'name': 'Hotel',
     'pluralName': 'Hotels',
     'shortName': 'Hotel',
     'primary': True}],
   'photos': {'count': 0, 'groups': []}},
  'referralId': 'e-0-54fcbf1a498ec6204cad12b7-0'},
 {'reasons': {'count': 0,
   'items': [{'summary': 'This spot is popular',
     'type': 'general',
     'reasonName': 'globalInteractionReason'}]},
  'venue': {'id': '4b634318f964a5204d6e2ae3',
   'name': 'Marquês de Pombal',
   'location': {'address': 'Pç. do Marquês de Pombal',
    'lat': 38.7249136994271,
    'lng': -9.149396142959395,
    'distance': 164,
    'postalCode': '1250',
    'cc': 'PT',
    'city': 'Lisboa',
    'state': 'Lisboa',
    'country': 'Portugal',
    'formattedAddress': ['Pç. do Marquês de Pombal',
     '1250 Lisboa',
     'Portugal']},
   'categories': [{'id': '4bf58dd8d48988d164941735',
     'name': 'Plaza',
     'pluralName': 'Plazas',
     'shortName': 'Plaza',
     'primary': True}],
   'photos': {'count': 0, 'groups': []}},
  'referralId': 'e-0-4b634318f964a5204d6e2ae3-1'},
...
...
```


This structure of Foursquare's data will help us create a dataset of venues for all the cities and label all the venues by their categories.

2.5. HERE.com API

In order to obtain geographical coordinates of various objects (cities, venues, etc.) we have created a developer account on https://developer.here.com and received credentials for accessing HERE Location Services REST APIs.

In particular, we will be using HERE Geocoding service to get latitude/longitude coordinates, since this service has proved to be robust and accurate (in contrast to OSM). We will call the service's API from Python via GeoPy library.

The following tasks will be solved by means of geocoding:

1. A dataset of cities for clustering will be formed as a subset of the list of European cities. We will specify a radius of `neighborhood` (in kilometers), then for each city we will find the number of universities located in the neighborhood of this city, and store those numbers as an additional column of the cities dataset. This column will allows us to select only the cities with desired number of universities nearby, so that we can narrow our choices of a place for relocation.
2. The cities will be visualized on a map of Europe, their populations and cluster labels will be shown as different marker colors and sizes.

3. METHODOLOGY

3.1. Background on cluster analysis

Our research for this project falls in a category of machine learning (ML) tasks called *unsupervised learning*, since we are dealing with data that has not been labelled, classified or categorized. Instead of responding to feedback, we are to identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A branch of unsupervised learning that studies algorithms designed for solving tasks of grouping objects in this manner is called *cluster analysis* or *clustering*.

In general, cluster analysis performs grouping a set of objects in such a way that objects in the same group (called *a cluster*) are more similar (in some sense) to each other than to those in other groups (clusters). It can be achieved by various algorithms that differ significantly in their understanding of what constitutes a cluster and how to efficiently find them. The appropriate clustering algorithm and parameter settings depend on the individual data set and intended use of the results. Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and failure. Clustering can be helpful as a data analysis activity in order to learn more about the problem domain, so-called pattern discovery or knowledge discovery.

The `scikit-learn` library provides a suite of different clustering algorithms to choose from, the most popular among them are: K-Means, Affinity Propagation, DBSCAN, Mean Shift. Each algorithm offers a different approach to the challenge of discovering natural groups in data. There is no best clustering algorithm, and no easy way to find the best algorithm for your data without using controlled experiments.

Probably, the most well-known clustering algorithm is *K-Means*. It is easy to understand and implement in code. Also, it is relatively fast since its computational complexity is minimum (i.e. linear O(n)). In scikit-learn, it is implemented via the `KMeans` class and the main configuration to tune is the `n_clusters` hyperparameter set to the estimated number of clusters in the data.

3.2. Exploratory data analysis

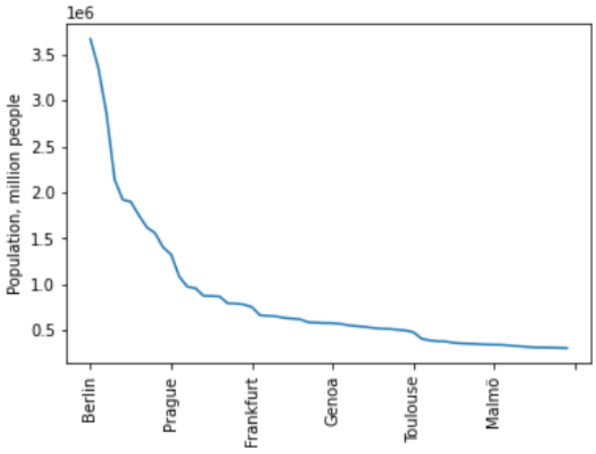
Now that we have parsed our data sources and extracted necessary information on cities and venues, countries and universities, our dataset of the cities of interest is ready for clustering. Before we start applying ML algorithms, let us explore the data in order to define the problem of our project more precisely.

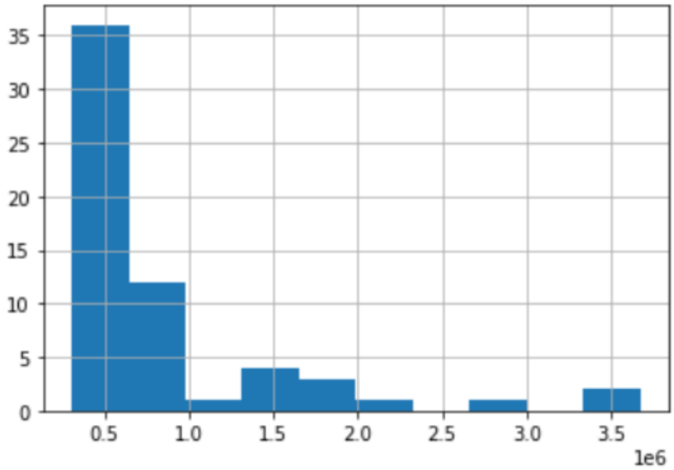
First, take a look the complete table of the cities sorted by population: the dataset contains 60 objects (cities) with population in range from 306.7 thousand to 3.67 million people. These are the cities having one or more top-ranked universities in close neighborhood and located in EU member countries:

	City	Country	Population	Lat	Lon	HasUnivs
0	Berlin	Germany	3669495	52.51605	13.37691	5
1	Madrid	Spain	3348536	40.41956	-3.69196	3
2	Rome	Italy	2856133	41.90323	12.49566	2
3	Paris	France	2140526	48.85718	2.34141	14
4	Vienna	Austria	1921153	48.20263	16.36843	3
5	Hamburg	Germany	1899160	53.55562	9.98746	2
6	Budapest	Hungary	1752286	47.49973	19.05508	1
7	Barcelona	Spain	1620343	41.38804	2.17001	3
8	Munich	Germany	1558395	48.13642	11.57755	2
9	Milan	Italy	1404239	45.46796	9.18178	6
10	Prague	Czech Republic	1324277	50.07913	14.43303	1
11	Cologne	Germany	1085664	50.94168	6.95517	2
12	Stockholm	Sweden	974073	59.33258	18.06683	3
13	Naples	Italy	959188	40.84016	14.25222	2
14	Turin	Italy	875698	45.06236	7.67994	2
15	Amsterdam	Netherlands	873289	52.36994	4.90788	4
16	Marseille	France	868277	43.29338	5.37132	1
17	Copenhagen	Denmark	794128	55.67567	12.56756	3
18	Valencia	Spain	791413	39.46895	-0.37686	1
19	Kraków	Poland	780981	50.06045	19.93243	1
20	Frankfurt	Germany	753056	50.11208	8.68342	3
21	Athens	Greece	664046	37.97614	23.73640	3
22	Helsinki	Finland	657674	60.17116	24.93266	3
23	Rotterdam	Netherlands	651870	51.91439	4.48717	4
24	Stuttgart	Germany	635911	48.76779	9.17203	3

	City	Country	Population	Lat	Lon	HasUnivs
25	Riga	Latvia	627487	56.94599	24.11487	1
26	Düsseldorf	Germany	619294	51.21564	6.77662	3
27	Dortmund	Germany	587010	51.51661	7.45830	3
28	Essen	Germany	583393	51.45183	7.01109	3
29	Gothenburg	Sweden	579281	57.70068	11.96823	2
30	Genoa	Italy	578000	44.41048	8.93917	1
31	Bremen	Germany	569352	53.07537	8.80454	2
32	Dresden	Germany	554649	51.05364	13.74082	1
33	The Hague	Netherlands	545273	52.08409	4.31732	4
34	Hanover	Germany	538068	52.37228	9.73816	1
35	Antwerp	Belgium	525935	51.22213	4.39769	4
36	Nuremberg	Germany	518365	49.45435	11.07350	1
37	Lyon	France	515695	45.75917	4.82966	2
38	Lisbon	Portugal	506654	38.72639	-9.14949	3
39	Duisburg	Germany	498590	51.43148	6.76356	3
40	Toulouse	France	479638	43.60579	1.44864	1
41	Palma de Mallorca	Spain	409661	39.57149	2.64694	1
42	Bologna	Italy	390636	44.50485	11.34507	3
43	Brno	Czech Republic	381346	49.19728	16.60368	1
44	Florence	Italy	378839	43.78238	11.25502	1
45	Bochum	Germany	364628	51.48800	7.21399	3
46	Utrecht	Netherlands	357676	52.08979	5.11415	6
47	Wuppertal	Germany	354382	51.27165	7.19678	4
48	Aarhus	Denmark	349977	56.15302	10.20487	1
49	Bilbao	Spain	345821	43.26890	-2.94530	1
50	Malmö	Sweden	344166	55.59670	13.00110	3
51	Nice	France	342637	43.70029	7.27766	1
52	Bielefeld	Germany	333786	52.01548	8.53232	1
53	Bonn	Germany	327258	50.73243	7.10187	2
54	Bari	Italy	320862	41.12588	16.86666	2
55	Münster	Germany	314319	51.96302	7.61782	1
56	Karlsruhe	Germany	313092	49.01094	8.40846	1
57	Catania	Italy	311584	37.51136	15.06752	1
58	Mannheim	Germany	309370	49.48651	8.46679	2
59	Nantes	France	306694	47.21812	-1.55306	3

A simple plot and a histogram below show the distribution of those cities by population:





Also, we can see how the cities are located on a map of Europe:

3.3. Hypothesis

Our hypothesis is that we can apply the K-means algorithm to cluster cities according to their commonalities and differences based on the most frequent categories of venues located in each city. Thus, we can add the two Russian hometowns of our clients to the above dataset and perform clustering so that the algorithm will enable us to choose a partition of EU cities similar to Novosibirsk and another partition of EU cities similar to Irkutsk.

For each city in the dataset, we will engineer a set of features that correspond to the frequencies of particular categories of recommended venues for that city. Then we will cluster the cities based on these features.

To make our problem more interesting and illustrative, we will split it into two subproblems:

- **Subproblem #1: Find a group of large European cities similar to Novosibirsk**

We create a new dataset of "large cities" as follows: select all cities with population >= 800 thousand people, add Novosibirsk to them. Then apply clustering and select a group of large cities containing Novosibirsk.

- **Subproblem #2: Find a group of medium European cities similar to Irkutsk**

We create a new dataset of "medium cities" as follows: select all cities with population <= one million people, add Irkutsk to them. Then apply clustering and select a group of medium cities containing Irkutsk.

3.4. Solution

For each city in the initial dataset, we have explored the city's venues via Foursquare API and therefore we have created a new dataset as follows:

Venue Category	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude
Airport Lounge	Amsterdam	52.36994	4.90788	Privium ClubLounge	52.309003	4.765398
Airport Lounge	Frankfurt	50.11208	8.68342	Lufthansa First Class Terminal	50.049840	8.564382
Airport Lounge	Frankfurt	50.11208	8.68342	Lufthansa First Class Lounge B	50.047246	8.572217
Airport Lounge	Gothenburg	57.70068	11.96823	SAS Lounge	57.668075	12.293854
Airport Lounge	Gothenburg	57.70068	11.96823	Vinga Lounge by Menzies Aviation	57.668057	12.293839
Airport Lounge	Lyon	45.75917	4.82966	Montblanc Lounge	45.717086	5.078384
Airport Lounge	Lyon	45.75917	4.82966	Salon Air France	45.722105	5.081130
Airport Lounge	Lyon	45.75917	4.82966	Salon Confluence	45.716308	5.078130
Airport Lounge	Novosibirsk	55.03977	82.91017	S7 Business Lounge	55.009708	82.666600
Airport Service	Amsterdam	52.36994	4.90788	Sky Priority Check-In	52.309406	4.763865
Airport Service	Gothenburg	57.70068	11.96823	SAS Check In	57.667941	12.294907
Airport Service	Lyon	45.75917	4.82966	Security Check	45.717549	5.076735
Airport Service	Novosibirsk	55.03977	82.91017	Взлётно-посадочная полоса	55.010181	82.667421
Airport Service	Novosibirsk	55.03977	82.91017	Паспортный контроль / Passport Control	55.009589	82.671067
Airport Service	Novosibirsk	55.03977	82.91017	Зона досмотра пассажиров / Security Control (3...	55.009491	82.667545
Arcade	Munich	48.13642	11.57755	Chaos Computer Club	48.153618	11.560834
Arcade	Prague	50.07913	14.43303	ArcadeHry	50.073157	14.164236
Arcade	Kraków	50.06045	19.93243	Kraków Pinball Museum	50.052748	19.939833
Arcade	Bologna	44.50485	11.34507	Piscina Junior	44.416583	11.349241
Arcade	Novosibirsk	55.03977	82.91017	Кёрлинг клуб Пингвин	54.999850	82.750312
Asian Restaurant	Rome	41.90323	12.49566	Thien Kim Ristorante Vietnamita	41.971203	12.433639

- **For large cities:**

The dataset contains 18 cities:

> *Amsterdam, Barcelona, Berlin, Budapest, Cologne, Hamburg, Madrid, Marseille, Milan, Munich, Naples, Novosibirsk, Paris, Prague, Rome, Stockholm, Turin, Vienna.*

The Foursquare API returns 1557 venues in 50 categories.

We use one-hot encoding and feature normalization, and the prepared data looks as follows:

Neighborhood	Airport Lounge	Airport Service	Arcade	Asian Restaurant	BBQ Joint	Bar	Beer Bar	Beer Store	Big Box Store	Brewery	Burger Joint	Café	Cocktail Bar	Coffee Shop	Cosmet Sh
Amsterdam	0.009524	0.009524	0.000000	0.009524	0.000000	0.057143	0.038095	0.009524	0.000000	0.028571	0.009524	0.123810	0.009524	0.152381	0.000
Barcelona	0.000000	0.000000	0.000000	0.000000	0.019608	0.049020	0.029412	0.019608	0.000000	0.019608	0.058824	0.058824	0.058824	0.078431	0.000
Berlin	0.000000	0.000000	0.000000	0.000000	0.013158	0.039474	0.026316	0.013158	0.039474	0.026316	0.013158	0.144737	0.052632	0.157895	0.000
Budapest	0.000000	0.000000	0.000000	0.000000	0.009615	0.028846	0.028846	0.009615	0.000000	0.009615	0.076923	0.038462	0.019231	0.163462	0.000
Cologne	0.000000	0.000000	0.000000	0.014493	0.000000	0.028986	0.000000	0.000000	0.028986	0.014493	0.014493	0.188406	0.043478	0.043478	0.000
Hamburg	0.000000	0.000000	0.000000	0.037037	0.012346	0.012346	0.000000	0.024691	0.000000	0.012346	0.012346	0.222222	0.037037	0.086420	0.000
Madrid	0.000000	0.000000	0.000000	0.000000	0.021277	0.031915	0.000000	0.021277	0.000000	0.021277	0.074468	0.063830	0.021277	0.085106	0.000
Marseille	0.000000	0.000000	0.000000	0.000000	0.000000	0.119048	0.000000	0.000000	0.000000	0.000000	0.023810	0.023810	0.000000	0.071429	0.047
Milan	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.021053	0.000000	0.000000	0.042105	0.031579	0.073684	0.042105	0.021053	0.000
Munich	0.000000	0.000000	0.012346	0.012346	0.012346	0.024691	0.012346	0.012346	0.000000	0.037037	0.012346	0.234568	0.049383	0.024691	0.000
Naples	0.000000	0.000000	0.000000	0.000000	0.000000	0.009009	0.000000	0.000000	0.000000	0.009009	0.045045	0.198198	0.045045	0.036036	0.009
Novosibirsk	0.011765	0.035294	0.011765	0.011765	0.035294	0.011765	0.011765	0.011765	0.023529	0.023529	0.011765	0.011765	0.023529	0.105882	0.011
Paris	0.000000	0.000000	0.000000	0.016949	0.000000	0.033898	0.016949	0.016949	0.000000	0.000000	0.000000	0.050847	0.000000	0.033898	0.050
Prague	0.000000	0.000000	0.009259	0.037037	0.000000	0.037037	0.018519	0.000000	0.000000	0.018519	0.027778	0.259259	0.027778	0.092593	0.000

Now, we are ready to apply K-Means algorithm for the large cities.

In order to obtain stable and interpretable clusters, we have performed a few experiments varying the `n_clusters` hyperparameter.

For our practical purpose, the optimal value turned out to be `n_clusters=5`, so that the typical cluster size is in range 4 to 5.

- **For medium cities:**

The dataset to analyze contains 49 cities:

> *Aarhus, Amsterdam, Antwerp, Athens, Bari, Bielefeld, Bilbao, Bochum, Bologna, Bonn, Bremen, Brno, Catania, Copenhagen, Dortmund, Dresden, Duisburg, Dusseldorf, Essen, Florence, Frankfurt, Genoa, Gothenburg, Hanover, Helsinki, Irkutsk, Karlsruhe, Krakow, Lisbon, Lyon, Malmo, Mannheim, Marseille, Munster, Nantes, Naples, Nice, Nuremberg, Palma de Mallorca, Riga, Rotterdam, Stockholm, Stuttgart, The Hague, Toulouse, Turin, Utrecht, Valencia, Wuppertal.*

The Foursquare API returns 2440 venues in 34 categories.

After one-hot encoding and feature normalization, the data looks as follows:

Neighborhood	Art Gallery	Australian Restaurant	Beer Store	Bookstore	Café	Cocktail Bar	Coffee Shop	Cosmetics Shop	Cupcake Shop	Dessert Shop	Farm	Food Truck	Garden Center	Gastropub
Aarhus	0.000000	0.000000	0.000000	0.000000	0.137931	0.034483	0.172414	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
Amsterdam	0.000000	0.013889	0.013889	0.041667	0.180556	0.013889	0.222222	0.000000	0.000000	0.041667	0.000000	0.000000	0.000000	0.041667
Antwerp	0.000000	0.000000	0.000000	0.031250	0.046875	0.046875	0.265625	0.031250	0.015625	0.015625	0.046875	0.000000	0.000000	0.015625
Athens	0.011628	0.000000	0.023256	0.023256	0.174419	0.046512	0.220930	0.000000	0.023256	0.151163	0.000000	0.000000	0.011628	0.000000
Bari	0.000000	0.000000	0.000000	0.047619	0.238095	0.071429	0.047619	0.000000	0.000000	0.071429	0.000000	0.000000	0.000000	0.023810
Bielefeld	0.000000	0.000000	0.000000	0.000000	0.250000	0.041667	0.041667	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.083333
Bilbao	0.000000	0.000000	0.000000	0.055556	0.166667	0.000000	0.000000	0.000000	0.000000	0.055556	0.000000	0.000000	0.000000	0.111111
Bochum	0.000000	0.000000	0.011765	0.023529	0.270588	0.011765	0.023529	0.000000	0.000000	0.000000	0.011765	0.023529	0.011765	0.011765
Bologna	0.000000	0.000000	0.000000	0.021739	0.119565	0.021739	0.010870	0.000000	0.043478	0.032609	0.010870	0.000000	0.000000	0.000000
Bonn	0.000000	0.000000	0.030303	0.000000	0.151515	0.060606	0.030303	0.000000	0.000000	0.000000	0.000000	0.000000	0.030303	0.000000
Bremen	0.000000	0.000000	0.000000	0.000000	0.266667	0.033333	0.066667	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.066667
Brno	0.000000	0.000000	0.010870	0.000000	0.326087	0.032609	0.076087	0.010870	0.000000	0.043478	0.000000	0.010870	0.010870	0.043478
Catania	0.000000	0.000000	0.020000	0.020000	0.300000	0.080000	0.000000	0.040000	0.000000	0.080000	0.000000	0.000000	0.000000	0.020000

Finally, we are ready to apply K-Means algorithm for the medium cities.

In order to obtain stable and interpretable clusters, we have performed a few experiments adjusting the `n_clusters` hyperparameter.

For our practical purpose, the optimal value turned out to be `n_clusters=10`, so that the typical cluster size is in range 6 to 7.

4. RESULTS

Below are the results of our cluster analysis for both subproblems:

- **For large cities:**

Our clients' hometown Novosibirsk appears in the following cluster:
['Barcelona', 'Budapest', 'Madrid', 'Novosibirsk']

The most frequent venue categories for the cities in that cluster are:

	Cluster Labels	City	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue	Country	Population
10	3	Madrid	Restaurant	Park	Italian Restaurant	Tapas Restaurant	Coffee Shop	Burger Joint	Café	Theater	Bar	Grocery Store	Spain	3348536
11	3	Novosibirsk	Coffee Shop	Pub	Park	Theater	Airport Service	BBQ Joint	Grocery Store	Flower Shop	Department Store	Gaming Cafe	Russia	1620000
12	3	Barcelona	Tapas Restaurant	Park	Restaurant	Pizza Place	Coffee Shop	Cocktail Bar	Café	Burger Joint	Wine Bar	Bar	Spain	1620343
13	3	Budapest	Coffee Shop	Park	Dessert Shop	Pizza Place	Burger Joint	Gym / Fitness Center	Restaurant	Italian Restaurant	Café	Wine Bar	Hungary	1752286

At first glance, the categories we could describe as commonalities of those cities are: Parks, Coffee Shops, Bars, etc.

- **For meduim cities:**

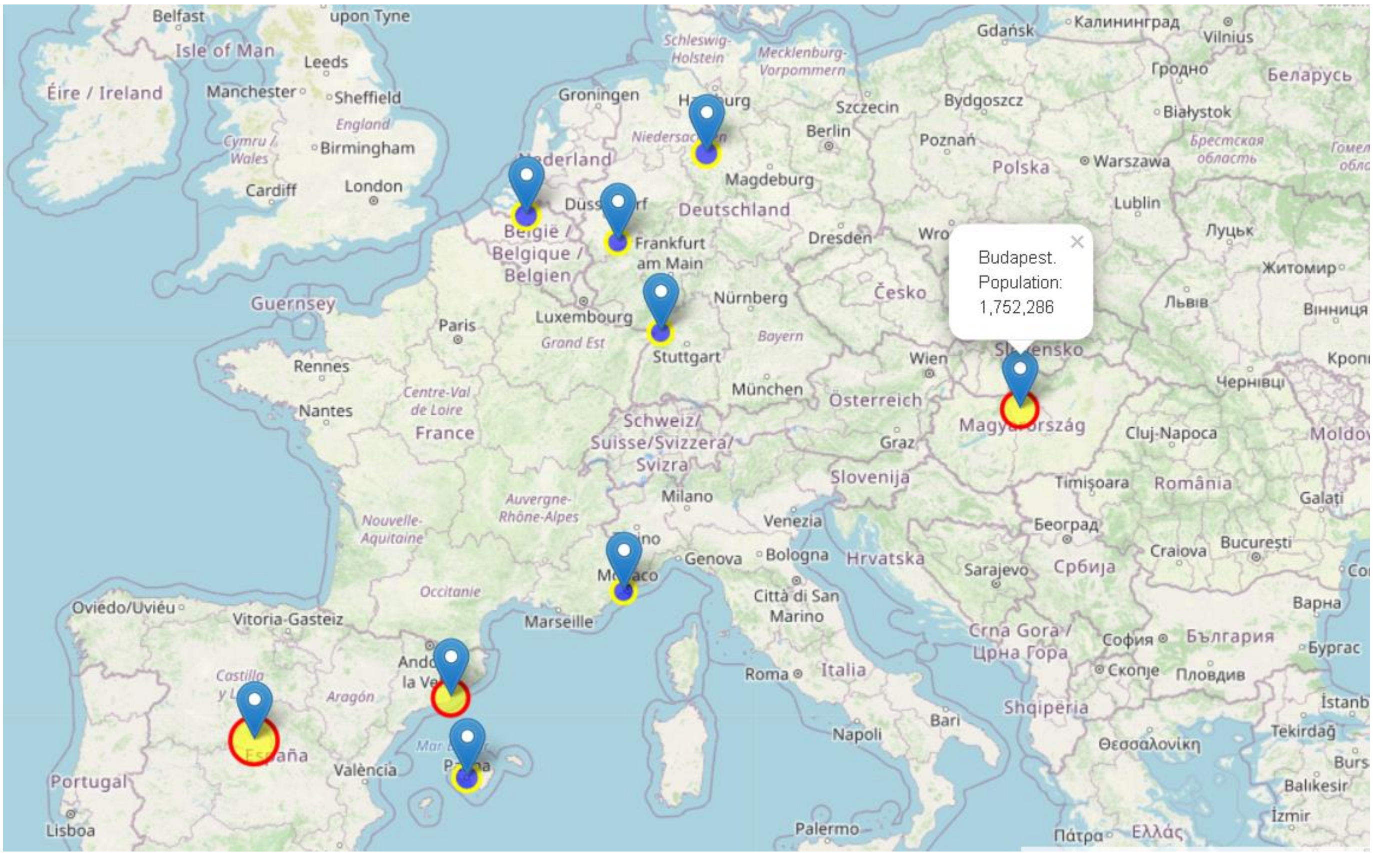
Our clients' hometown Irkutsk appears in the following cluster:
['Antwerp', 'Bonn', 'Hanover', 'Irkutsk', 'Karlsruhe', 'Nice', 'Palma de Mallorca']

The most frequent venue categories for the cities in that cluster are:

	Cluster Labels	City	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue	Country	Population
42	9	Bonn	Italian Restaurant	Café	Park	Pedestrian Plaza	History Museum	Steakhouse	Cocktail Bar	Mountain	Pub	Coffee Shop	Germany	327258
43	9	Nice	Italian Restaurant	Café	Scenic Lookout	Train Station	Park	Cocktail Bar	Pedestrian Plaza	Theater	Steakhouse	Cupcake Shop	France	342637
44	9	Hanover	Italian Restaurant	Park	Coffee Shop	Café	Steakhouse	Cocktail Bar	Gastropub	Gym / Fitness Center	History Museum	Train Station	Germany	538068
45	9	Irkutsk	Café	Coffee Shop	Cocktail Bar	Pedestrian Plaza	Garden Center	Pub	Wine Bar	Dessert Shop	Gastropub	Food Truck	Russia	617000
46	9	Karlsruhe	Café	Italian Restaurant	Coffee Shop	Train Station	Gastropub	Park	History Museum	Scenic Lookout	Pub	Cocktail Bar	Germany	313092
47	9	Antwerp	Coffee Shop	Italian Restaurant	Park	Gym / Fitness Center	Cocktail Bar	Farm	Café	Steakhouse	Bookstore	Cosmetics Shop	Belgium	525935
48	9	Palma de Mallorca	Café	Coffee Shop	Italian Restaurant	Cocktail Bar	Park	Steakhouse	Scenic Lookout	Wine Bar	Dessert Shop	History Museum	Spain	409661

At first glance, the categories we could describe as commonalities of those cities are: Cafe, Pedestrian Plaza, Cocktail Bar, Wine Bar, Scenic Lookout, etc.

This is how the EU cities belonging to the above clusters look on the map of Europe:



The recommended large cities are marked with red circles with yellow filling, while the recommended medium cities are marked with yellow circles with blue filling.

Relative sizes of the markers (circles) correspond to the populations of the marked cities.

5. DISCUSSION

Based on the cluster analysis we have performed, we could make the following recommendations:

- Our clients should consider **Barcelona, Budapest or Madrid** as their primary choices of relocation, if they are willing to move to a large EU city similar to Novosibirsk;
- The clients might consider **Antwerp, Bonn, Hanover, Karlsruhe, Nice or Palma de Mallorca**, if they prefer to move to a moderate-size city similar to Irkutsk.

Interestingly, the algorithm finds that the majority of the "target" cities for our clients' relocation belong to Spain and Germany. Although, this might be related to some specific preferences of Russian, Spanish and German users of the Foursquare platform rather than to actual similarities of cities in those countries.

The following adjustments may be applied in order to improve the solution and increase its practical value for potential clients:

- The dataset of the cities may be filtered more thoroughly based on the number of universities nearby or on their fields of study, e.g. if we are interested in a specific area of study for our clients' children;
- Some additional characteristics of the cities can be added to the dataset as extra features, such as crime rate, quality of healthcare, proximity to seas or lakes, etc.;
- Some venue categories may be considered irrelevant for some clients, hence we can discard them or assign least significant weighs to least significant categories (e.g. bars, airports, history museums, if we are not interested in them).

In addition, it may be useful to experiment with different clustering algorithms, for example:

Mean-Shift Clustering. In contrast to K-means clustering, there is no need to select the number of clusters as mean-shift automatically discovers this. That's a massive advantage. The fact that the cluster centers converge towards the points of maximum density is also quite desirable as it is quite intuitive to understand and fits well in a naturally data-driven sense.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN). DBSCAN is a density-based clustered algorithm similar to mean-shift, but with a couple of notable advantages. Firstly, it does not require a pe-set number of clusters at all. It also identifies outliers as noises, unlike mean-shift which simply throws them into a cluster even if the data point is very different.

6. CONCLUSION

Our analysis has proven that even simple data science techniques enable people and businesses to make important decisions based on facts and historical data rather than intuition. In particular, well-known and widely accepted unsupervised learning algorithms can generate a solution that might be easily interpreted and applied to real-world problems.

We can utilize ready-to-use libraries or modules, such as pandas, scikit-learn, matplotlib and folium to extract, parse and process significant amounts of data and visualize results of our computations.

Various external APIs (e.g. Foursquare, Here, OSM) may be valuable sources of information to help enrich our datasets with new features, such as geographical coordinates or users' recommendations from social networks.