Capstone Project report

IBM Data Science Professional Certificate

Recommending Rental Properties

By Aron Shirazi

August 2020

Please send your inquiries to aron.shirazi (a) gmail.com

Table of Contents

Introduction

Many people struggle in the time of renting a property which is followed by a huge amount of search, calls, inspections and more importantly a great deal of time and money. The traditional arrangement in the real estate market has developed agency profession in which people are referred to a local expert to find an appropriate rental property. Search engines could evolve the market in the way that information made accessible for all prospective tenants no matter of current location. However, the problem was finding a suitable option from thousands which made practically online renting impossible especially in populated cities. Hopefully, by the advent of artificial intelligence, the idea of rental recommending systems seemed possible.

In this project, Aron is assumed as an ordinary person who got some criteria in renting a property. The story takes him to the development of a rental property recommender system. Aron is a data scientist who has decided to enrol a professional degree in one of the well-known universities in Boston. He is an international student who needs to explore many secrets in the real-estate sector of Boston city. However, he has no time to do much field exploration just to land on a suitable property for renting. He will arrive just one day before the semester start date. So, he is thinking of finding a comprehensive and accurate solution.

After consulting with several of his friends he got two options. Asking an agent to find a suitable apartment which does not ensure cost-effectiveness and efficiency. The other option which he is much into developing a code which takes criteria on one side and suggests best available options. He knows how to use API services of Foursquare and Realtor API. He came up with a methodology which is described in the methodology section. This project takes Aron as an example of an audience who is going to find an apartment close to campus. But this project can benefit a wide range of people as far as they come with some criteria in the first hand. However, covering a wide range of users demands developing a user-friendly online interface which is beyond the scope of this project.

The problem in the realm of data science can be described as the following:

> *Having a city name and the desired range of a specified location, a recommending system is needed to explore specifications of neighbourhoods and search for available rental properties to match them with the given criteria. Main parameters are the distance from the desired location, herein a university campus, and important venues which define the appropriateness of each rental option.*

This code can save time and cost in the way that a user can explore options easily. Not only it ensures having an optimised solution but also lets the user train his understanding of given criteria. For example, Aron might reach to a conclusion that one set of criteria is beyond his budget and he needs to compromise the criteria. So, instead of failing in rental search, this project makes him capable of making realistic decisions without losing time and money.

Data sources

There are four data sources used in this project:

1. List of neighbourhoods in the desired city, Boston and Cambridge which are retrieved from two public reports.
2. Coordinates of neighbourhoods extracted by Geopy from Nominatim
3. Foursquare used for retrieving data of venues including their name, location, and category. In this project, only location and categories are going to be used.
4. Realtor API which is going to utilised for exploring available rental options in nominated neighbourhoods

For example, Aron decided to pursue his studies at MIT which is in Cambridge neighbourhood. Although the idea choice would be within Cambridge, Aron suspects that it would be costly to live there. So, he is interested to consider surrounding neighbourhoods as well. So, in the first step, a list of neighbourhoods is retrieved. Then several neighbourhoods are nominated based on the distance for further exploration. In the next step, coordinates of these neighbourhoods are found by use of Geopy. Having those coordinates, Foursquare API can be utilised to find venues accompanying their categories. Each category is scored by Aron which contributes to the overall score of rental options. In the next stage, available apartments for rent are found by use of Realtor API. Having coordinates of each option, it is possible to calculate the distances from desirable neighbourhoods aligned with other features like price. At last, options are clustered by use of Kmeans which makes Aron capable of selecting one of those clusters and go through each included option.

Methodology

In this section, the methodical framework of the developed recommender system is described. It is comprised of 10 parts and each explained separately.

1. **Downloading the neighbourhoods' report**
   To find an appropriate property close to the campus which Aron has decided to pursue his studies, Boston and Cambridge neighbourhoods should be explored. This section accounts to define the searching domain. For doing so, neighbourhoods' lists are needed, and two reports used:
   - http://www.bostonplans.org/getattachment/6f48c617-cf23-4c9f-b54b-35c8a954091c
   - https://www.cambridgema.gov/CDD/externallinks/Profiles/neighborhoodprofile
2. **Importing, cleaning, and forming the dataset**

in this section, the imported datasets are taken into cleaning and dropping unnecessary features. Then data types are converted to the desired state for further analysis. Finally, a dataset formed by combining two regions neighbourhoods for further study.

The downloaded pdf files have many pages including demographic data. Page 5 of the first source and page 65 of the second file include population distribution of Boston and Cambridge. The pdf file is read by use of tabula library. Then, the desired columns are selected, and the data is converted into float type.

3. **Finding the coordinates of neighbourhoods**

   For exploring properties, we need to have some coordinates to go around it, analyse and identify options. In this project, geocode is used for this purpose. Important parts are as follows:

   - using Geopy, Nominatim
   - passing GeocoderTimedOut for avoiding errors of timing out
   - setting a search limit for a neighbourhood
   - using sleep of 1 sec for avoiding server runtime limit block
   - passing a random symbolic password
   - random ordering of address

   Finally, because some neighbourhoods' coordinates not found, they are handled manually.

4. **Plotting location of neighbourhoods**

   in this step, found coordinates of neighbourhoods are plotted along with their attached names. The initial zoom command in folium is not used, instead, a more efficient method of fit_bound has been utilised.

5. **Filtering neighbourhoods by distance**

   Aron has decided to find an apartment as close as possible to the campus. His ideal choice would be less than 3 km from the MIT campus. In this step, the selected neighbourhoods are filtered by the calculated distances by use of the Geodesic method from Geopy library.

6. **Finding venues in the selected neighbourhoods**

   After taking out coordinates of neighbourhoods, it is time to extract the specifications of registered venues. To do so, the Foursquare API service is used. There are 4 corresponding steps introduced in the following:

   - defining API credentials by using dot env. in this method, credentials are saved in a .env file which set to be ignored by Github in the time of publication in .gitignore file.
   - defining two main functions: the first function finds venues around a specified location by passing latitude/longitude. The limit is set to 100 and the radius is 3000m by default. The second function, extract venues specification stored in the retrieved JSON file.
   - exploring neighbourhoods' venues by running two functions along with all extracted coordinates in the former step. A new dataset is generated here which stores specifications of venues.
   - analysing the venues dataset which starts by finding how many venues found per neighbourhood. Then the number of unique venues is calculated as well as their categories.
   - plotting found venues imposed on neighbourhoods' plot to see the distribution of them.

7. **Analysing each neighbourhood by found venues**

   In this step, neighbourhoods are analysed by classifying their venues into interested groups of Restaurant, Bar, Sport, Coffee, and Gym. Then the corresponding frequencies are calculated after and based on given scores of classes, neighbourhoods are ranked. this part consists of four steps:

   - establishing one-hot dataset

- grouping the dataset by its neighbourhood
- grouping categories into 5 desired classes
- frequency calculation

8. **Developing scoring frame and ranking neighbourhoods**

   At some point, a recommender system needs to use a scorecard to rank options. The reason for doing so is, it would be quite useless to consider all neighbourhoods. But the challenge is the scoring system needs to have a logical background. At this point, the easiest way is by using user preferences. For example, Aron has put these scores out of 10 for each class of venue:

   - Gym: 10 / 10
   - Coffee: 8 / 10
   - Restaurant: 5 / 10
   - Bar: 5 / 10

   Taking these scores, it is possible to rate neighbourhoods and select top 2.

9. **Finding, filtering, and plotting rental properties**

   In this section, rental properties are identified by the use of realtor API. The JSON file coming from the request method is flattened and a pandas dataframe is developed which includes interested parameters. This section consists of the following parts:

   - exploring the rental properties in Boston by use of realtor API
   - conversion of JSON file to pandas dataframe and selection of interested columns
   - finding the distance of properties from the campus and selected neighbourhoods, and filtering them
   - visualisation of found properties in folium

10. **Clustering and visualisation of results**

    This is the last stage where filtered properties are classified and plotted. This process is accomplished through three steps:

    - clustering properties by use of Kmeans method which is an unsupervised machine learning method. The clustering is applied on several features, distance from the centre of neighbourhoods, number of bathrooms and bedrooms, more importantly price.
    - creating a dataset for passing to plotting section. it includes main columns of features.
    - plotting the properties using folium library, in which each cluster is colour coded.

    There are several important points about clustering rental properties:

    - Features: many features could be used as input to the Kmean algorithm. However, to make it practical and meaningful, it is needed to consider features which are independents. As a result, the number of full bath and baths are merged into a single parameter.
    - Distances: it is meaningless to include distances from all neighbourhoods. Each property has a distance from each neighbourhood and close distance to one equates to a long distance to another one. As a result, minimum distances from all neighbourhoods are considered.
    - Number of clusters: number of clusters should be investigated to enhance the Kmeans algorithm performance. One way of doing so is plotting inertia versus the number of clusters. The elbow point can give the minimum number of clusters for our study. However, trial tests are conducted to diverge classes if they include dissimilar properties. The elbow point detected to be 5 and decided cluster number is 7.

Results

There are many intermediary tables and plots in the project which can be viewed on the Jupyter notebook by use of nbviewer.

- Jupyter notebook link:

  https://github.com/Aron-XXV/Coursera_Capstone/blob/master/Boston%20Neighbourhoods/Recommending%20rental%20properties.ipynb

- Nbviewer link:
  https://nbviewer.jupyter.org/

Disregarding many of them, only important results are presented here.

1. Map of considered neighbourhoods
   This map is generated by combining the list of neighbourhoods in both Boston and Cambridge. The location is found through Geopy library and map is plotted by folium.

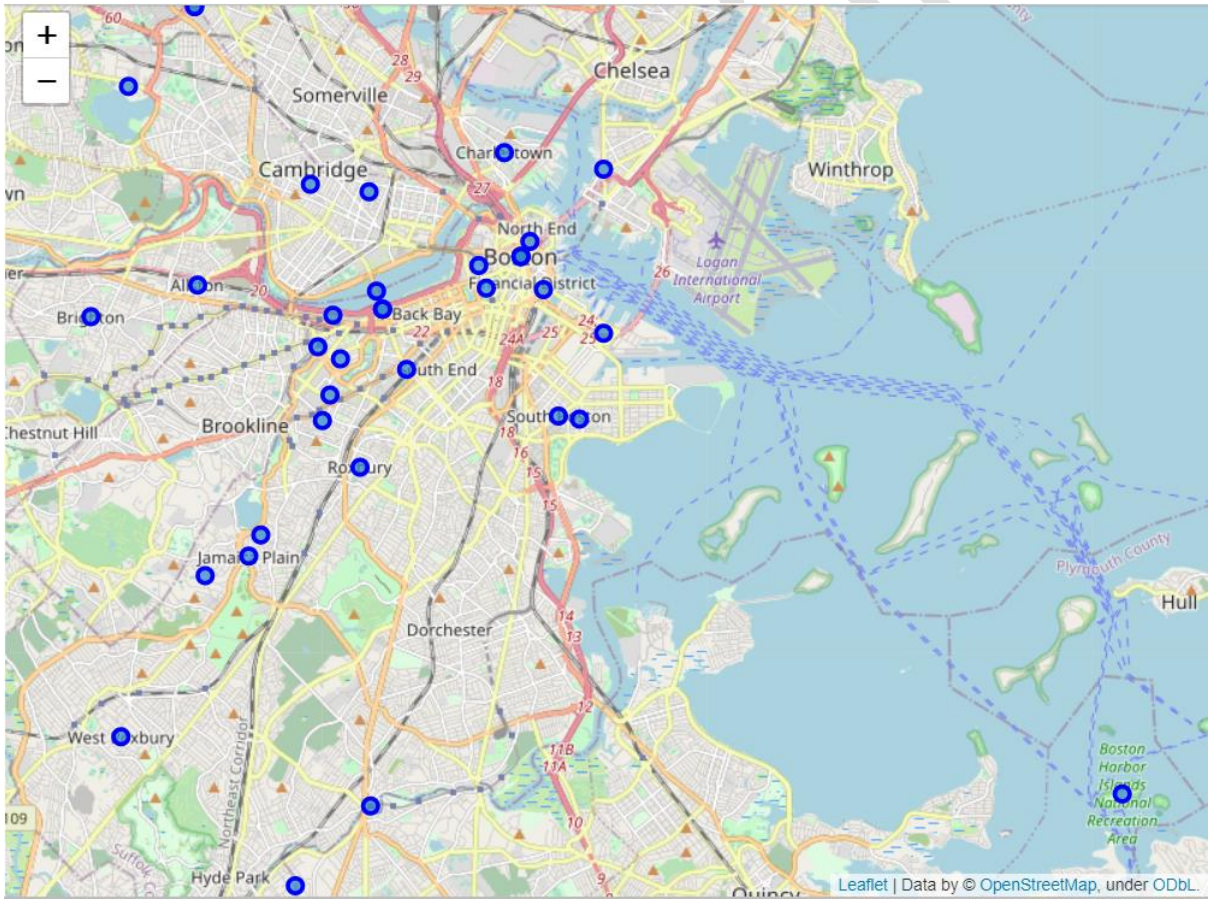


Figure 1. Map of all neighbourhoods

2. Map of selected neighbourhoods
   This map shows the selected neighbourhoods after implementing a 3 km distance limitation. The important point is Geopy location outputs for two neighbourhoods of Cambridge Port and East Cambridge are the same.

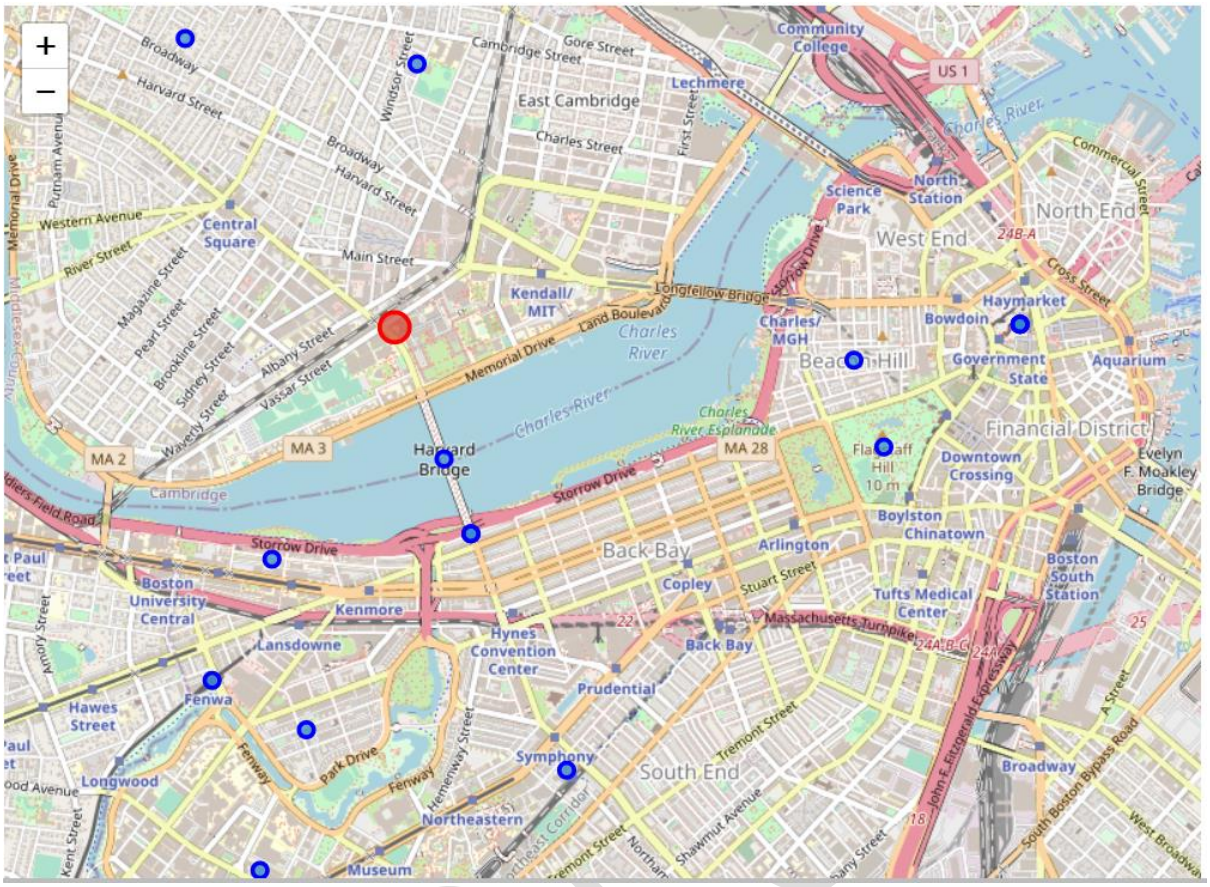


Figure 2. Filtered neighbourhoods

3. List of venues

   After using Foursquare API, 1500 venues are found, 100 for each of the selected 15 neighbourhoods. However, it does not mean that all of them are unique, this list only includes 369 unique venues with 139 unique categories.

Table 1. List of found venues

	name	categories	lat	lng	neighbourhood
0	Charles River Esplanade	Trail	42.351128	-71.100407	Back Bay
1	Island Creek Oyster Bar	Seafood Restaurant	42.348838	-71.095280	Back Bay
2	Fenway Park	Baseball Stadium	42.346282	-71.097535	Back Bay
3	Fenway Beer Shop	Liquor Store	42.344928	-71.099908	Back Bay
4	Mei Mei	Chinese Restaurant	42.347481	-71.105949	Back Bay
...	...	...	...	...	...
1495	USS Constitution	Boat or Ferry	42.372450	-71.056510	Strawberry Hill
1496	Tatte Bakery & Cafe	Bakery	42.351966	-71.043246	Strawberry Hill
1497	Residence Inn by Marriott Boston Downtown/Seaport	Hotel	42.350179	-71.047857	Strawberry Hill
1498	Whole Foods Market	Grocery Store	42.345304	-71.063061	Strawberry Hill
1499	Thinking Cup	Coffee Shop	42.351653	-71.074884	Strawberry Hill

1500 rows × 5 columns

4. Map of venues

   The location of identified venues is depicted in this plot.

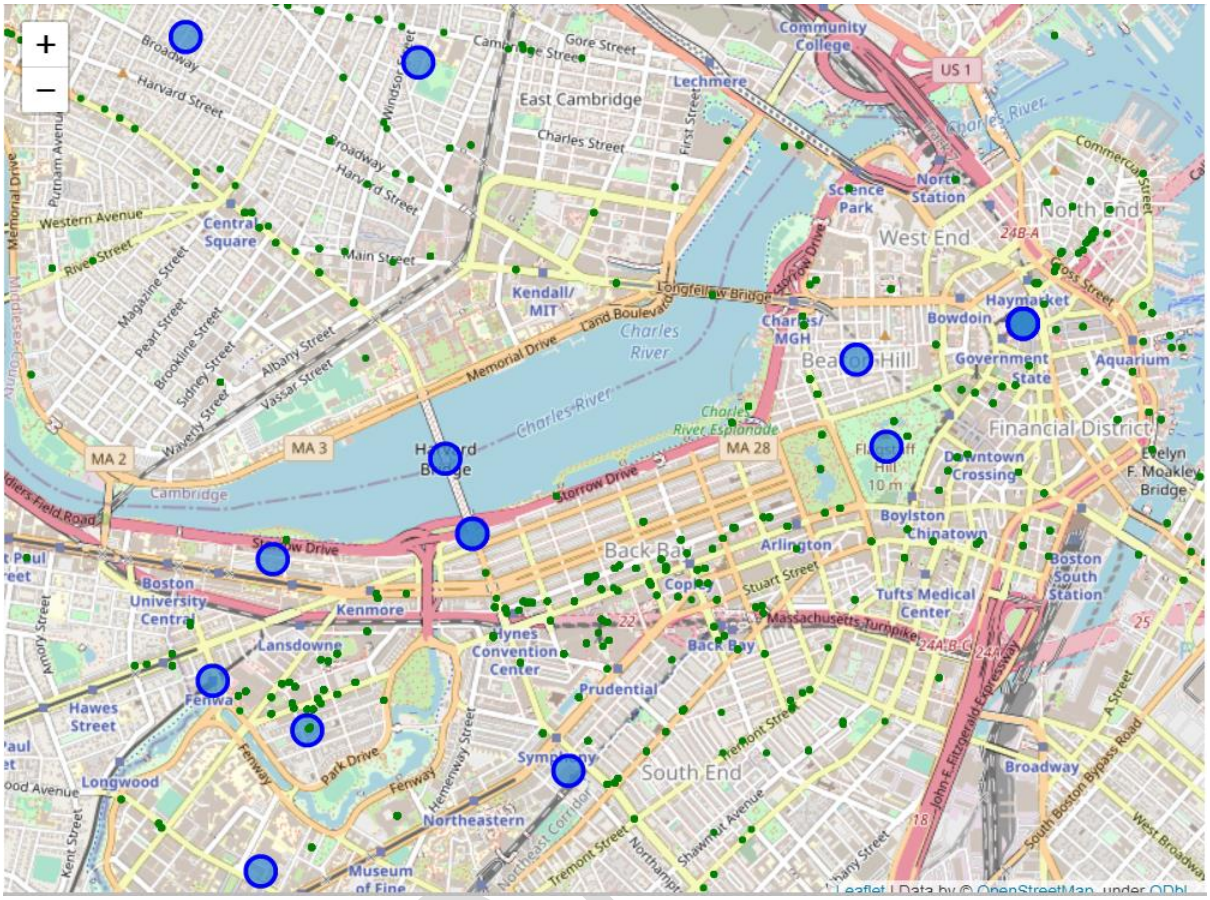


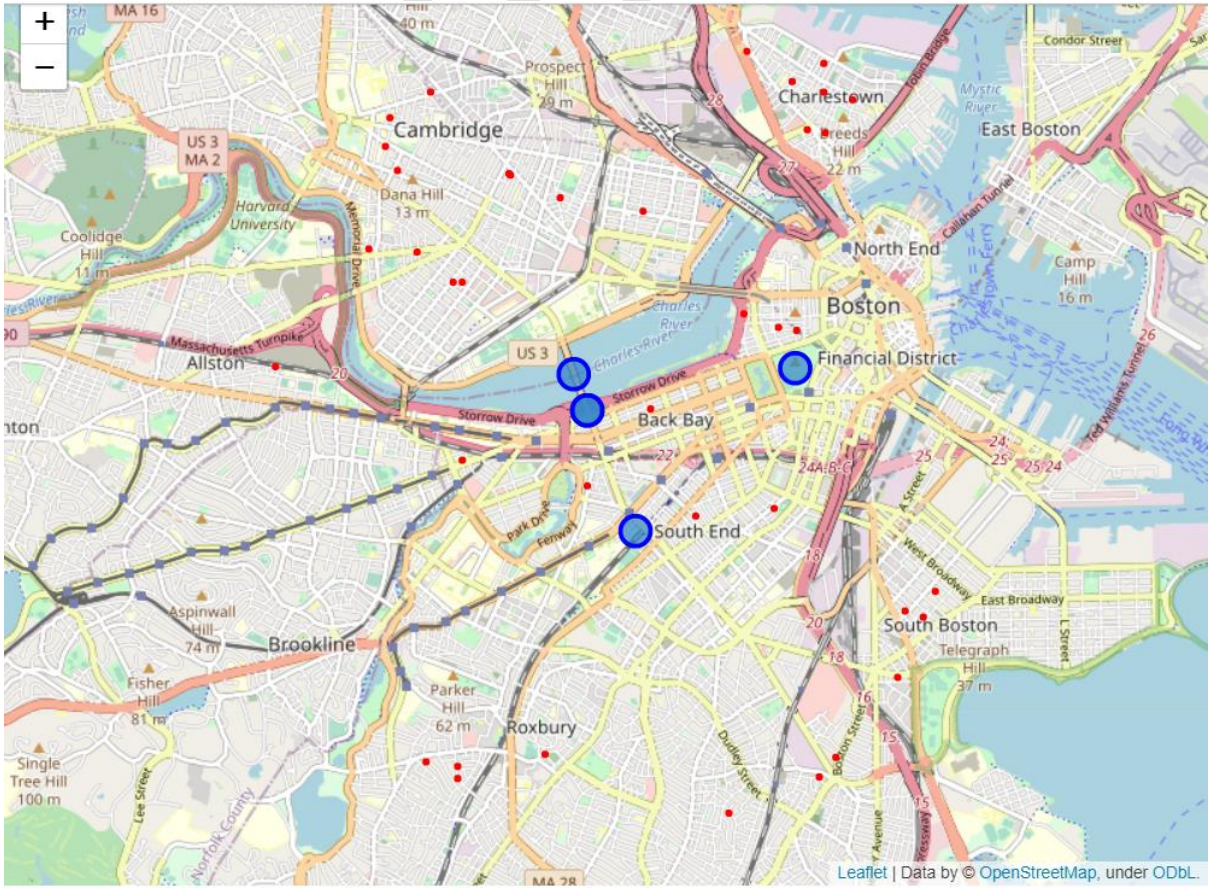
Figure 3. explored venues map from Foursquare API

5. Scoring tables
   After implementing the scoring framework, each neighbourhood has ended to a score coming from its nearby venues. Then this list is sorted and first 5 neighbourhoods are selected which are Downtown, South End, East Cambridge, MIT, and Cambridge Port.

Table 2. sorted scored list of neighbourhoods

neighbourhood	Restaurant	Bar	Coffee	Gym	Score
Cambridgeport	0.320000	0.483871	0.653061	0.847458	2.304390
East Cambridge	0.320000	0.483871	0.653061	0.847458	2.304390
Downtown	0.280000	0.645161	0.489796	0.847458	2.262415
South End	0.440000	0.645161	0.489796	0.677966	2.252923
MIT	0.320000	0.322581	0.489796	0.847458	1.979834
Wellington-Harrington	0.386667	0.645161	0.326531	0.508475	1.866833
Longwood	0.373333	0.322581	0.489796	0.677966	1.863676
Mid-Cambridge	0.453333	0.483871	0.326531	0.508475	1.772209
Riverside	0.266667	0.161290	0.816327	0.508475	1.752758
Strawberry Hill	0.266667	0.161290	0.816327	0.508475	1.752758
West End	0.266667	0.161290	0.816327	0.508475	1.752758
Neighborhood Nine	0.360000	0.161290	0.489796	0.677966	1.689052
Beacon Hill	0.266667	0.000000	0.489796	0.847458	1.603920
Fenway	0.320000	0.161290	0.326531	0.677966	1.485787
Back Bay	0.360000	0.161290	0.326531	0.508475	1.356296

6. Selected properties
   Realator API provides only 63 results for each search. For more results, it should be subscribed to remove this limitation. However, these 63 options still work for this project and out of 63, 39 property is selected considering distance criteria from neighbourhoods.



7. Finding elbow point

   After the selection of features, it is necessary to explore an appropriate number of clusters. To do so, Kmeans inertia needs to be plotted to detect the elbow point. The visible pattern of the following figure tells that elbow point happens are cluster number of 5. However, for enhancing the clustering accuracy, some random tests of properties are conducted. The experienced performance has proved that the optimal number of clusters is 7.

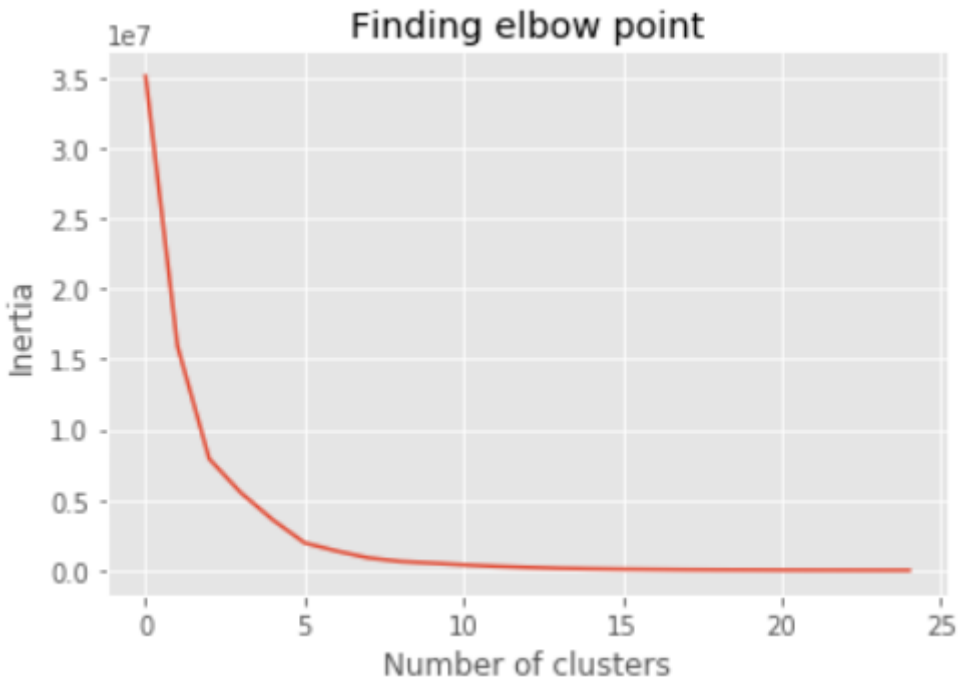


Figure 4. Inertia versus the number of clusters

8. Results of Kmeans clustering

   In this table, a list of clustered properties is provided. It includes important features needed for further decision making or even detailed study of properties. At this point, Aron can decide which class he should take for visit or contact.

Table 3. List of clustered properties

No.	Cluster Labels	price	beds	bath	distance	address.line	address.lat	address.lon	address.line
0	0	1695.0	0	1.0	1.262326	856 Beacon St	42.347276	-71.104035	856 Beacon St
1	0	2000.0	1	1.0	1.419777	94 Pearl St	42.362262	-71.105042	94 Pearl St
2	0	2000.0	2	1.0	2.759420	36 Mozart St	42.321511	-71.104501	36 Mozart St
3	0	2050.0	1	1.0	2.949377	10 Sumner Rd	42.376035	-71.112334	10 Sumner Rd
4	0	2200.0	1	1.0	2.491432	147 W 8th St Unit 147	42.334595	-71.053620	147 W 8th St Unit 147
5	0	2250.0	1	1.0	0.586108	274 Marlborough St # #ph	42.351547	-71.082650	274 Marlborough St # #ph
6	0	2350.0	2	1.0	2.590577	5 Badger Pl Unit 1	42.378234	-71.062758	5 Badger Pl Unit 1

7	1	6800.0	2	1.0	0.411751	87 MT Vernon Unit Carriageh	42.358455	-71.067997	87 MT Vernon Unit Carriageh
8	1	6995.0	5	2.0	1.855454	158 Western Ave	42.364736	-71.109149	158 Western Ave
9	1	7500.0	2	2.0	0.582007	149 West Newton St	42.342501	-71.077431	149 West Newton St
10	1	7600.0	6	2.0	2.463433	226 W 5th St	42.336252	-71.050169	226 W 5th St
11	2	4600.0	3	1.0	1.947019	45 Tremont St	42.371224	-71.098578	45 Tremont St
12	2	4600.0	3	2.0	2.555949	12 Lexington Ave	42.377544	-71.059577	12 Lexington Ave
13	2	4700.0	4	3.0	1.644588	9 Hamlin St	42.369303	-71.092936	9 Hamlin St
14	2	5000.0	3	2.0	1.646613	185 Charles St	42.368173	-71.083430	185 Charles St
15	3	6000.0	5	3.0	2.512167	365 Harvard St	42.371572	-71.111376	365 Harvard St
16	3	6000.0	7	3.0	2.788076	11 Hartford St	42.317518	-71.073618	11 Hartford St
17	3	6500.0	3	2.0	2.206961	24 Pleasant St Unit 24	42.374755	-71.062725	24 Pleasant St Unit 24
18	4	2450.0	2	1.0	0.621358	51 Hemenway St Apt 1	42.345150	-71.089823	51 Hemenway St Apt 1
19	4	2500.0	1	1.0	2.253069	22 Hingham St Unit Cottage	42.365009	-71.114732	22 Hingham St Unit Cottage
20	4	2500.0	3	1.0	2.915346	11 Forbes St	42.321905	-71.108102	11 Forbes St
21	4	2550.0	2	1.0	0.691124	141A Revere St	42.359529	-71.071861	141A Revere St
22	4	2600.0	2	1.0	2.836399	223 Boston St Unit House	42.322284	-71.061392	223 Boston St Unit House
23	4	2600.0	2	1.0	2.955105	93 Kirkland St	42.378228	-71.107649	93 Kirkland St
24	4	2600.0	3	1.0	2.880514	21 Elder St Unit 1	42.320580	-71.063353	21 Elder St Unit 1
25	4	2650.0	3	2.0	2.860958	63 Mozart St	42.320423	-71.104588	63 Mozart St
26	4	2800.0	1	1.0	1.960426	12 Murdock St Unit 12	42.371300	-71.098788	12 Murdock St Unit 12
27	4	2900.0	3	2.0	2.857780	2 Pearl Street Pl	42.380660	-71.062880	2 Pearl Street Pl
28	4	2950.0	2	1.0	2.672002	114 Bartlett St	42.379094	-71.066384	114 Bartlett St
29	4	3000.0	2	2.0	2.823994	NaN	42.328977	-71.054366	NaN
30	4	3200.0	2	1.0	1.325639	258 Shawmut Ave	42.343205	-71.068428	258 Shawmut Ave
31	5	9800.0	7	4.0	2.762905	25 Ware St	42.373627	-71.112819	25 Ware St
32	6	3500.0	2	1.0	1.343464	11 Watson St	42.362163	-71.103970	11 Watson St
33	6	3575.0	4	1.0	2.620255	7 Grimes St Unit 2	42.334083	-71.051457	7 Grimes St Unit 2
34	6	3700.0	3	1.0	2.216910	12 Austin St	42.374972	-71.064670	12 Austin St

35	6	3750.0	2	2.0	0.344067	34 Mount Vernon St # #ph	42.358136	-71.065899	34 Mount Vernon St # #ph
36	6	3950.0	3	2.0	2.983811	442 Main St	42.381594	-71.071524	442 Main St
37	6	4000.0	5	2.0	2.800970	30 Wadsworth St	42.355079	-71.125291	30 Wadsworth St
38	6	4100.0	5	2.0	2.258699	NaN	42.322459	-71.094523	NaN

9. Map of clusters

   In this figure, clustered rental properties are plotted. The bigger red radius is the location of the MIT campus, and smaller blue circles are selected neighbourhoods.

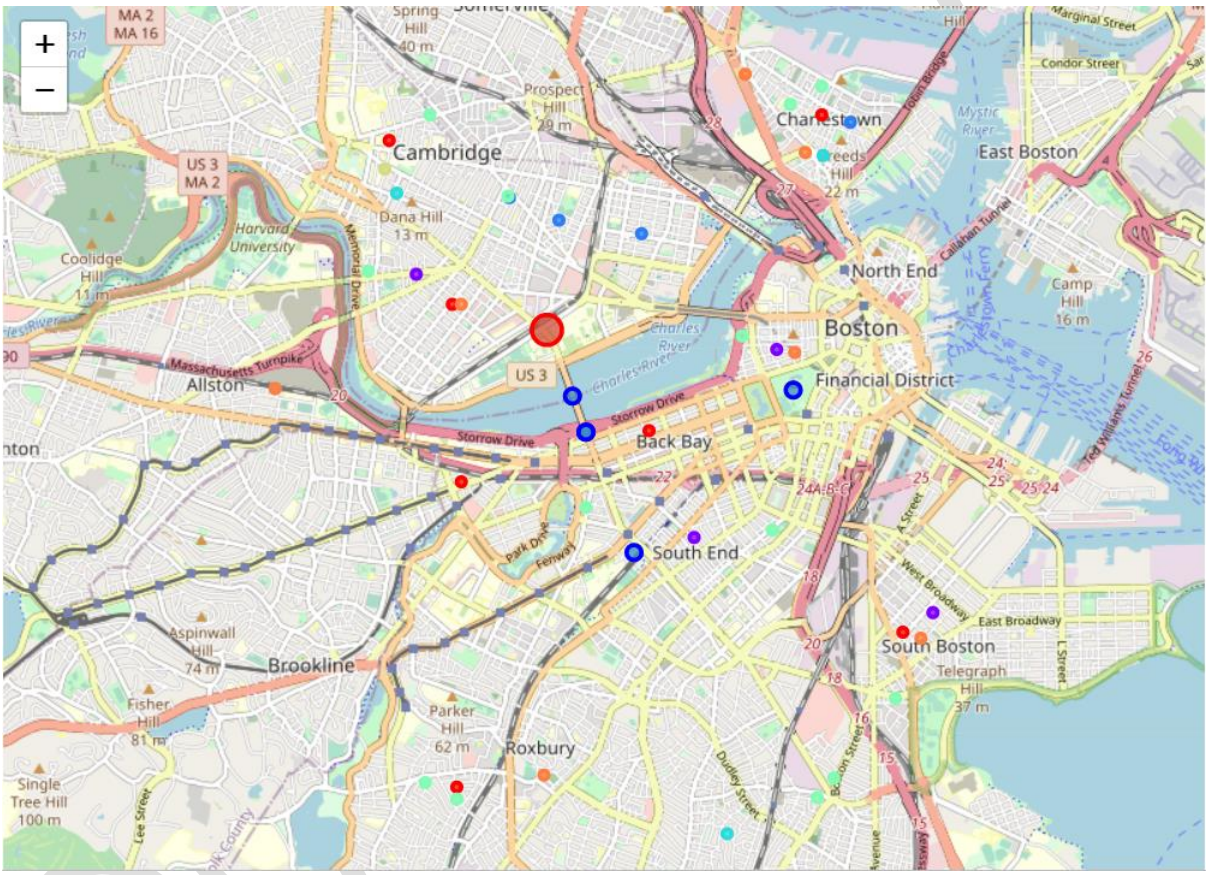


Figure 5. map of clustered rental properties

Conclusion

The purpose of this project was the development of a rental property recommender system which took a case of finding a property for a student in Boston. The list of neighbourhoods retrieved from public official reports and their coordinate found. Then venues of each neighbourhood are explored and used as a basis for ranking them. In the next step, rental properties are searched by use of Realtor API and Kmeans algorithm applied for clustering of renting options.

The final result of the project, a list of clustered properties and related map, is quite promising as it could effectively classify a very wide range of properties into similar groups. At this point, a user can easily compare clusters to select his desirable one and investigate included property cases for detailed study or even contact. As a result, a great deal of time and money wasted through discovery and search in the market is saved and ended in several minutes of running code and having an appropriate short-list. Another advantage of this project is the user gets trained through the process. It means that

the user can broaden his knowledge of the market and makes a better decision in the time of trade-offs between criteria and compromising one feature.

At last, the outcome of this project can be enhanced if the limitation of API calls is removed and more rental properties could be analysed. Another aspect of future improvements is implementing the project through an online user-friendly interface.

Please send your inquiries to aron.shirazi (a) gmail.com