|  |
| --- |
| Capstone Project - The Battle of Neighborhoods |
|  |

|  |  |
| --- | --- |
| 07-10-2018 | Live closer to work |

|  |
| --- |
| Utilizing Foursquare data to recommend place to live in New York. |

Capstone Project - The Battle of Neighborhoods

Live closer to work

# **Introduction/Business Problem**

According to New York Times magazine, New Yorkers spend in average 35.9 min each way on commute to work every day. That’s the longest commute compared to all cities in USA. New York is also in the top of other inglorious competition – hours spent in congestion per year. Average New Yorker spends almost 90 hours per year in crowded, stuffy trains and subway platforms. Only habitants of Los Angeles have it worse with average of 104.1 hours per year.

It’s a known fact that living closer to your workplace can save you stress, free up time for leisure and help increase your overall happiness. Why don’t people move closer to their work?

Choosing a right place to live is a complex optimization task in which you need to find a balance between low commute time and a very vague “general feeling” of the neighborhood. People don’t like change, and if they need to move to another place, wouldn’t it be comforting to know that the new neighborhood will be similar to the one that you’re leaving behind?

Business case for the analysis is creation of a tool, which for given current home address and work address will create suggestions of neighborhoods which resemble the one which customer is currently living in, but are located closer to the place they commute to every day. Leveraging Foursquare data, we want to capture the “general feeling” on the neighborhood and look for similar ones with lower commute time.

Figure 1 One-way commute time and hours per year in congestion. Source: New York Times.

Possible further development of the tool but out of scope for this project, would be directing customer to relevant apartment listings. There are many apartment listings sites, where customer can filter listings by features of apartments, but what they are missing is capturing “general feeling” of the neighborhood.

# **DATA**

There are 3 main data sources which will be used to solve the problem:

1. Dataset of New York Boroughs and Neighborhoods with geographical coordinates, which can be accessed freely under link: <https://geo.nyu.edu/catalog/nyu_2451_34572>.
2. Foursquare data regarding venues in close location of neighborhoods.
3. Forge geolocator – to get geographical coordinates of given addresses.

Example to illustrate how data will be used:

Mark currently lives in 1030 Neil Avenue, Bronx, NY and commutes every day to work at 545 5th Ave, Manhattan, NY. First, using Foursquare API get venues near 1030 Neil Avenue and create a profile for the neighborhood. After that, create similar profiles for all neighborhoods from the 1st data set. Find neighborhoods which are within e.g. ~ 1km range, ~ 1-3 km range and ~ 3-5 km range from 545 5th Ave and in each sector recommend a neighborhood which profile resembles surroundings of 1030 Neil Avenue the most. Mark will be left with three options, for which he can explore e.g. apartment listings to make a final decision.

# Methodology

There are three main methods used in this project. Finding a neighborhood which is “close” to our home neighborhood in geographical sense, capturing “general feeling of a neighborhood” and finding neighborhoods which are “similar” - which calls out for defining a measure of distance between neighborhoods characteristics.

## Geographical distance

In order to find neighborhoods close to the workplace, we need a function which for given geographical coordinates of two points on a map will return distance between them in km. A solution easy to implement and well-conditioned down to distances as small as a few meters on the earth’s surface is Spherical Law of Cosines. It gives a 1-line formula for calculating d - distance between two points on a map:

d = arcos (sin φ1 ⋅ sin φ2 + cos φ1 ⋅ cos φ2 ⋅ cos Δλ) ⋅ R

Where:

R is radius of earth in km,

φ1, φ2 are latitudes,

and Δλ = difference between longitudes.

Important note: in this equation both longitude and latitude are given in radians instead of degrees. Luckily Python library *math* provides a function radians() which converts angle in degrees to radians.

## Neighborhood similarity

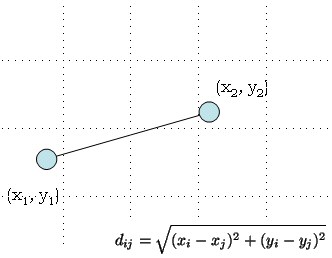
Even though k-means algorithm is not used in this project, reasoning behind grouping points in multidimensional space into clusters is borrowed and used to compare neighborhoods. Let’s assume for now that we can represent each neighborhood as a point in a n-dimensional space. We can think of a point in a n-dimensional space as a vector of its coordinates (of length n). E.g. point in 2-dimensional space can be represented as: (0,1) and using a common axis naming it is a point placed at 0 on X axis, 1 on Y axis. Most common and natural method of measuring distance between two points in n-dimensional space in Euclidean distance. This is the same way we measure distance in every day, e.g. when we want to check if a new wardrobe will fit into our bedroom. A simplistic example on a figure above can be easily generalized to n-dimensional space:

Figure 2 Euclidean distance between two points in 2-dimensional space. Source: bigsnarf WordPress



Python *scipy* library provides spatial.distance.euclidean(u, v) function, which calculates distance between two vectors in a NumPy array format.

Knowing how to measure distance between points in n-dimensional space, we need to find a way to represent each neighborhood in the same manner.

## Capture neighborhood characteristics

Each dimension in a multidimensional space represents characteristic of an object. E.g. in 3-dimensional space we speak about length, width and height. For capturing “general feeling” of a neighborhood we will use the same concept but with different characteristics. Venues located nearby is a key characteristic of the neighborhood as they well represent the surroundings. We will use them to find each neighborhoods representation as a point in n-dimensional space.

Foursquare API (<https://developer.foursquare.com/>) provides and easy way to explore neighborhoods all over the world. Given geographical coordinates it returns venues located close by with their category (Coffee Shop, Museum, Park, bus station etc.). You can also find rating of venues and read comments left by other users.

In this project we will use only number of venues in each category. Venues categories will be used as axis names, and a coordinate will be a number from [0,1] representing percentage of particular category among all venues located nearby given neighborhood. Example: in Lincoln Centre 10% of all venues are Art Galleries.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Neighborhood** | **Yoga Studio** | **Cofee Shop** | **American Restaurant** | **Italian Restaurant** | **Art Gallery** | **Art Museum** | **Arts & Crafts Store** | **…** |
| Midtown | 0.01 | 0.05 | 0.02 | 0.02 | 0 | 0.07 | 0.1 | **…** |
| Lincoln Centre | 0 | 0.1 | 0.3 | 0.1 | 0.1 | 0.1 | 0 | **…** |
| Greenpoint | 0 | 0.03 | 0.1 | 0.08 | 0 | 0 | 0.03 | **…** |

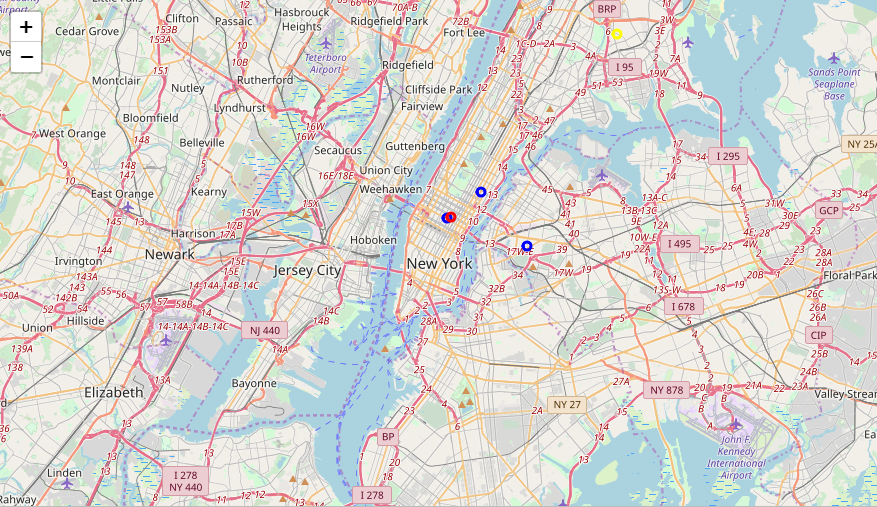
Having this representation, you can calculate Euclidean distance between neighborhoods. The smaller the distance the more similar neighborhoods are.

# RESULTS

Function *recommend()* is implemented in workbook: <https://github.com/warstwowa/Coursera_Capstone/blob/master/Capstone-Project.ipynb>

Function is well defined for all workplaces located in New York and home addresses from all around the world, however as Foursquare data quality might vary depending on location, we don’t recommend to use it for home addresses located outside New York. Map below displays results of applying function to example defined in Introduction/Business Problem section of this document.

Yellow circle marker located in northern New York (Bronx) is a current home address. Red circle marker represents work place, and blue markers represent recommended neighborhoods. It’s clearly visible that tree recommendations vary in distance from the workplace.



# DISCUSSION

Methodology presented to solve problem of recommending neighborhood closer to the workplace makes some assumption which could be challenged. It might be interesting to explore some enhancements.

## Characteristics

In this project the only characteristics used was number and category of nearby venues. One might consider use of additional characteristics, e.g. population density, average cost of rental. This would require using additional data sources.

## Irrelevant variables

In presented model we use all venues categories we received from Foursquare. A possible improvement to the model would be analyzing variables and removing ones of low relevance (e.g. highly correlated).

## Distance vs Commute time

In the project straight line distance in km is used to measure distance from home to work, however this is not necessarily directly correlated with commute time, which is our main focus in this exercise. A place located closer to the workplace might have a longer commute, e.g. because it’s far away from the subway station, and the only possible commute is bus, which would be stuck in traffic during rush hours. On the other hand, public transportation and especially subway network is very well developed in New York, and thus straight line might still be a good approximation.

## Zones

Zones in which model is recommending place to live are set ad-hoc on 1, 3 and 5 km radius. The is a possibility to explore other zone definitions.

# CONCLUSION

During the project I have created well-conditioned function *recommend()* which returns logical recommendations of places to live. The Discussion section has presented possible enhancements of the tool which could lead to receiving better results. This project shows how easy it is to use machine learning to recommend places to live, which in my view is the future of real-estate marketing. Everybody would agree that the place that we live in has a huge impact on our life, but searching for perfect place to live using standard filtering on apartment listings can be exhausting. Why not let the machine do the hard work for us? Next step would be inventing robots who would also do packing and moving for us.

Thanks for reading!