Neighborhood recommendation, based on people and homes density, average wage and nearby services

Francisco Antunes

04/2020

1. INTRODUCTION

I’s very common nowadays people change their jobs more frequently, mainly in big cities, where there are many job opportunities. Moving from one neighborhood to another it’s not always easy, due to many factors. One big factor is that we doesn't know how the neighborhoods nearby the new job area are. The solution, usually, is open the city map online and search one by one neighborhood information, or even searching on Google: “Is it good live in XXXX?”.

1.1 Problem

People usually have to move due to new job opportunities. Sometimes it gets in a neighborhood that is far away from their current place. So, they need to find a neighborhood that is closer to the new job, but still have similar services to the ones they have nearby. It’s also interesting use other data to get the most similar neighborhoods.

1.2 Target

The target are the people that are moving to a different neighborhood in the same city, but want to keep similar characteristics to the ones that they are used to have. This algorithm could be used in a recommendation system in a house rental websites.

2. DATA REQUIREMENTS

2.1 Data sources

Wikipedia: As an example, I’ll use the wikipedia page from my city neighborhoods (Curitiba, Brazil). https://pt.wikipedia.org/wiki/Lista_de_bairros_de_Curitiba. It includes all neighborhoods name, population, average wage and area.

geopy.geocoders library : based in neighborhood name, I’ll get the coordinates using the API.

Foursquare API: Foursquare API will be used to get venues nearby each neighborhoods.

2.2 Data cleaning/formatting

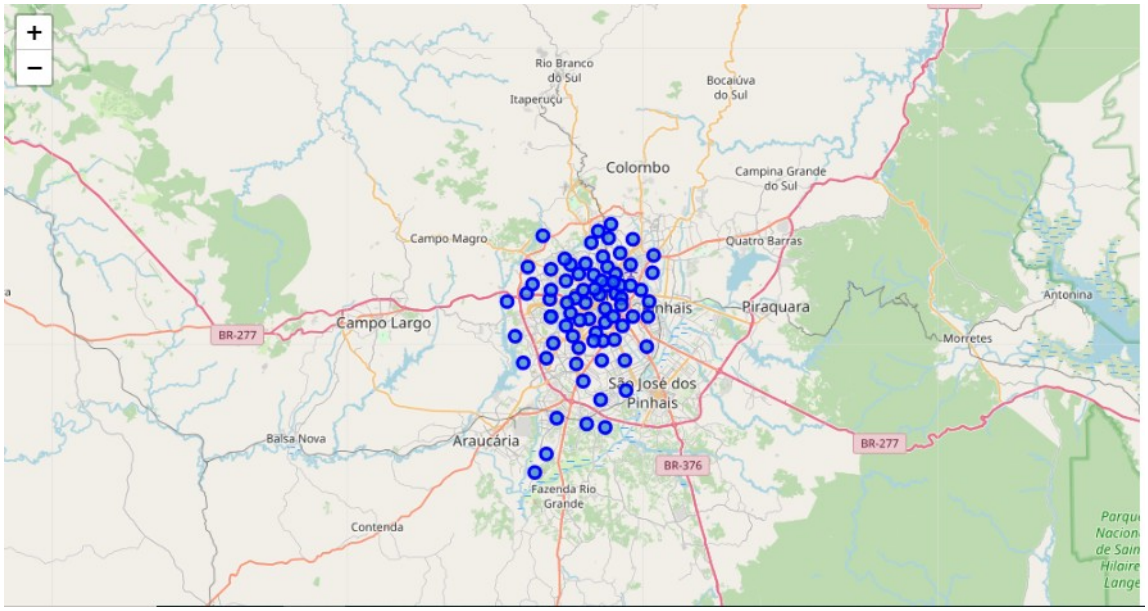
The raw data was collected from tables inside a Wikipedia html page by a *pandas* function called *"read_html()"*. This function returns a list of *Dataframe* objects analysing html plain text, looking for <td> and <tr> tags that are used in html to format tables. A little bit fuzzy issue was that I had to remove from html code a " " symbol, after I noticed problems in data format. After that, I could iterate over the dataframe list and append all tables (removing those headers) data into one panda dataframe. Another important thing is to remove some rows and columns with NaN (not a number) value, because the response from read_html function generate NaN from merged cells in html tables.

Another important thing to do is to format column types from 'object' to 'numeric' and create two new columns: people_per_area and homes_per_area. These two columns are created from another existing columns and they are really important to demonstrate the population and home density in each neighborhood. Some adjustments are also made in specific neighborhoods name to get them ready to be used in geopy.geocoder library. After fixing these issues, I had to adjust some average wage column values for inflation from 2000 to 2010, to get all average wage values in the same "value". It was necessary because some data was from 2000 and others from 2010.

3 Methodology

3.1 Data Visualization

When we are treating with locations, the best tool to visualize the data is to put them on a map, so we can get the scope of analysis more clearly. To get a good visualization, we've used *folium.Map* object. Below, we demonstrate in a blue circle the location of each neighborhood.



The coordinates from each neighborhood were been caught from geopy.geocoder library, requesting the function with neigborhood name concatenated with city name and state initials.

3.2 Venue Data Analysis

At this point, we have a clear dataframe with all neighborhoods from Curitiba and data that represents population and home density, average wage from the house owner and the latitude and longitude from each neighborhood.

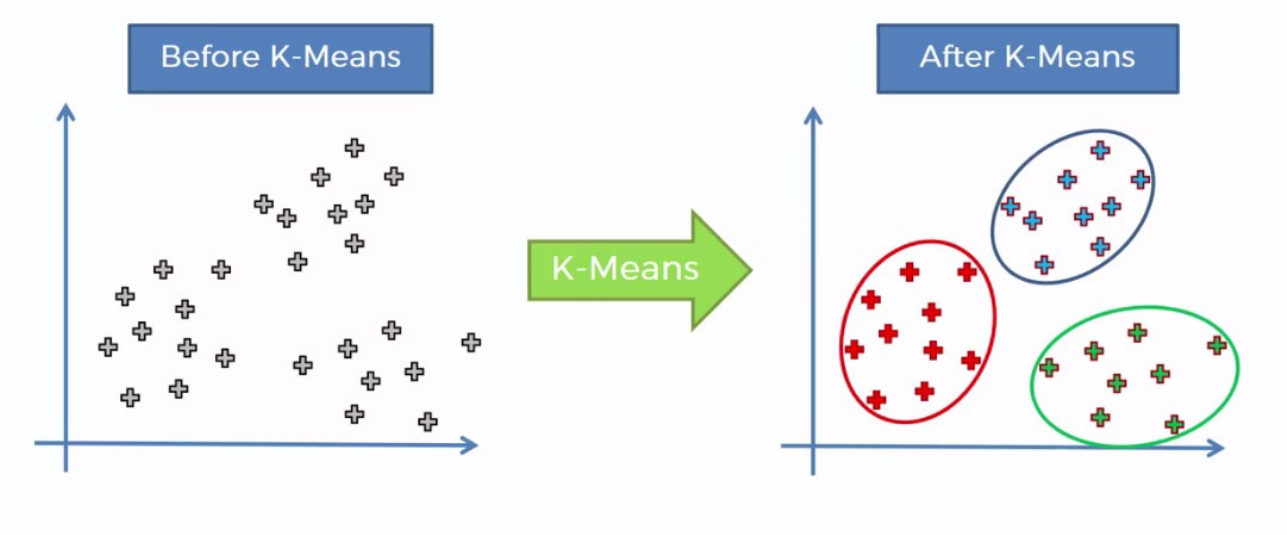
For using venue category data in a clustering algorithm, it's necessary format them to categorical data. First, I've used pandas gey_dummies() function to transform venue categories in categorical columns. After that, I've calculated the frequency of each category in the neighborhoods, so I could get the 10th most frequent category for each neighborhood.

Finally, we've got so far a dataframe with all column that will be used in clustering algorithm.

3.3 Clustering Algorithm

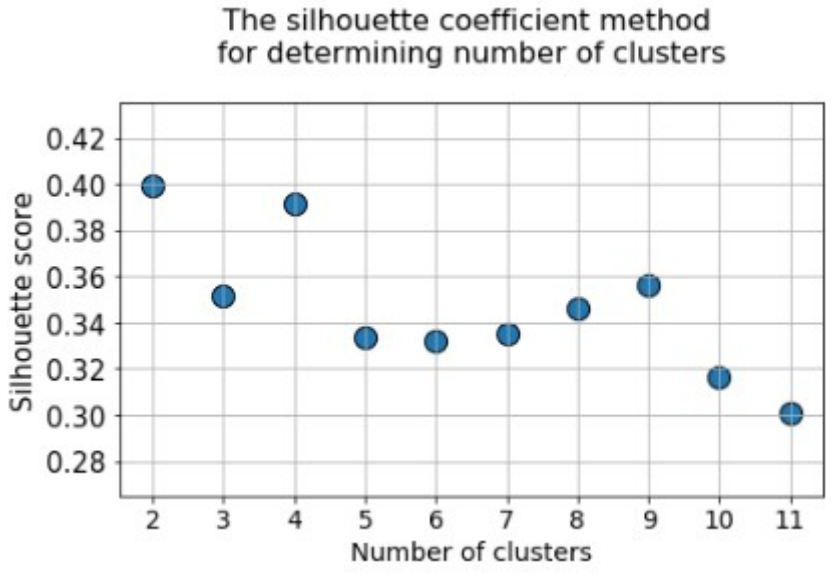
A good technique to group data that have similar features is the clustering algorithm. They are unsupervised algorithm that create clusters that have similar characteristics and group data into. As we intend to make a recommendation system, it's a perfect strategy.

For the purpose of the project, I decided to use k-means algorithm. This algorithm is really good for clustering data and simple to understand. Basically, the algorithm tries to find the minimum distance for intra-cluster data and maximize distance for inter-cluster data.



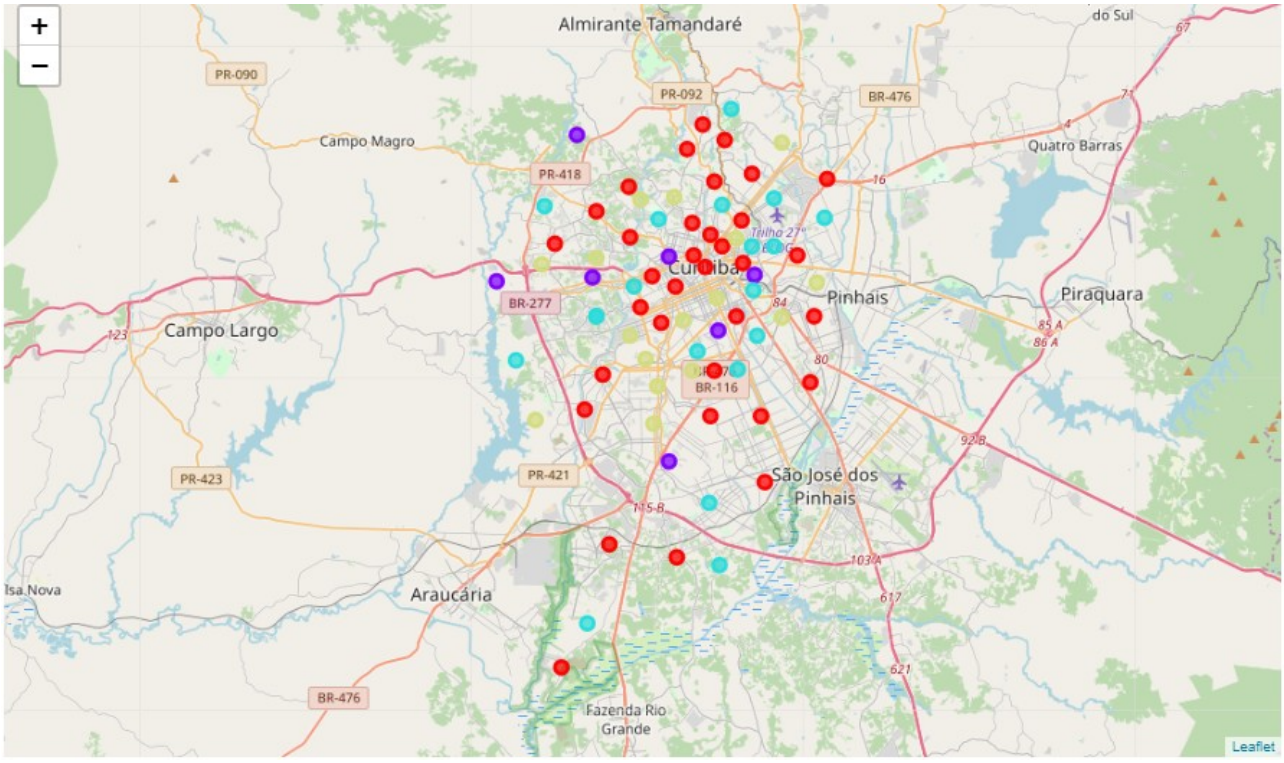
One important definition for the k-means algorithm is the definition ok *k:* the number of cluster to be generated. There are many methods to calculate the best *k.* **Elbow method** is one of the most famous, but also one of the simplest one. I decided to use ***silhouette method***. Silhouette method, calculate a score, that considers intra-cluster and inter-cluster distance to determine the best k. So, I

made a loop to run k-mens algorithm with many *k* values (from 2 to 12), and calculate silhouette score. The result is shown in the chart below:



As a result, k=2 is the best choice, but considering the business rules, dividing neighborhoods in just 2 groups is not interesting, so I decided to used the second best choice, k=4.

The result from the clustering algorithm is well shown in the chart below:



4 Results

Going back to the problem to be solved, to get the recommended neighborhoods that are similar from the one I live (same cluster) and also closer to my new job neighborhood, I had to calculate distance between each neighborhood and the new job neighborhood.

As a result from the recommendation system, I”ve got the closest neighborhood that are in the same cluster as my current home neigborhood.

0	Parolin
1	Cristo Rei
2	Mercês
3	Mossunguê
4	Pinheirinho
5	Riviera
6	Lamenha Pequena

5 Discussion

Well, the result is really good, considering that is really hard to compare regions in the same city, as we know that people have different perception of the environment. It’s important to consider that same category venue doesn't mean that the venue have similar quality, for example, or even the same target people. But this kind of “score” is very subjective, what makes difficult to be 100% assertive. To improve the algorithm, another external data could be considered, like security information, or even some health information.

6 Conclusion

As a conclusion, I can say that I could use machine learning skills to answer a real world problem, even though there’s room for improvement. It means that data analysis is an awesome method of understand the world and even help us to solve problems.