Battle of Neighborhoods

August 10, 2019

This is a data science project to find the best neighborhood to live in a faraway city. This project deals with unlabeled data, due to that reason I used an unsupervised algorithm, namely the K-mean algorithm. This project is an excellent example of how effective the K-mean algorithm clustering unlabeled data.

Contents

1 Introduction - Five Ws

1.1 What is the problem?

I work in downtown Memphis, TN. In general, I live 30 mins away from work, when there is no traffic. However, since I am commuting when most of the people are commuting to there work, there is almost always traffic on the roads. So it is easily 45-60 mins one-way trip. Being optimistic, considering one-way is 45 mins, it is 90 mins for a round trip. For a week with 5 workdays, it is 7.5 hours, for a month it is 30 hours, for a year it is 16 days. So for a given year, I am wasting full 16 days counting days and nights riding my car wasting my time.

I am planning to move St. Louis, Missouri for a new job offer. While I have a friend already living there, I would like to do my analysis and find out where I can move. But driving time is something I strongly want to reduce.

So the problem is, where I should move to save some time from driving but still have good amenities such as restaurants, cafes, parks, shopping, etc within reachable distance.

Also, one of the jobs I was being interviewed is in St. Louis, Illinois. I want to do a similar analyze there to find out if I can find a good neighborhood to live.

1.2 Where is this?

It is St. Louis, Missouri, where birds sing and elephants bath, just kidding. But it sure looks like a fantastic place to live. There are tons of things to do around there. The population was roughly 300 k and rising. The job market seams great too.

1.3 When is this applicable?

I know this is a changing world! The time will change everything. The time of this analysis is August 2019. So don't blame me if you decided to move based on this data analysis in 2050. But the good thing is, I developed the program to pull the latest data. So if you re-run the program in 2050, you should be (may be…) fine?

1.4 Why do we do this?

It is to primarily to save time when I move. I am currently spending so much time on the road, 16 full days per year! just to commute. People say time is money. So it is to save me some money. I am sure if you are in the same boat, following this, you might able to save some money with this. Who doesn't like saving money for next cruise trip? Wait, is someone paying me when I save my own time? Ney.. I will use this saved time to play with my daughters. Not everything is money. I think I have bipolar disorder.

1.5 Who cares?

Do you even here me? It is to save money (really the time) while finding the best neighborhood to live. If you are someone who cares about saving money (time), you should read. If you have plenty of those lying around that you don't know what to do, this is not for you. You should spend some money buying a boat and traveling the world instead of reading this.

2 Data

We need data to do our analysis. This section will gather all the required data and do the clean-up job so that the data are usable. I am hoping to gather neighborhood names from Wikipedia (use web scraping with BeautifulSoup package) and use FourSquare to obtain point of interest around the selected neighborhoods.

1. St. Louis Neighborhoods - The names of the St. Louis neighborhoods will be obtained from the Wikipedia page https://en.wikipedia.org/wiki/List_of_neighborhoods_of_St._Louis. This page has neighborhoods of the St. Louis along with some demographic data. This will be great for met to get started. All I really need is the names of the neighborhoos so that I can find the address, latitude, longitude, nearby point of interest details using FourSquare module.
2. As mentioned before, I am going to use FourSquare to obtain point of interest data.

2.1 Load the required libraries.

I will start by importing some libraries. These libraries are not necessary to use in this section. But to keep it clean, I always like to have all my libraries are loaded at the top of the program. That way I know which modules I have used in this project.I have used the following libraries in this project.

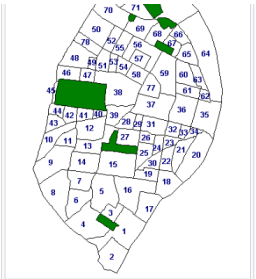
1. BeautifulSoup - To scoop data from web
2. Geocoders - To obtain the address of neighborhoods
3. Folium - To obtain the information of venues around specific point of interest
4. pandas - To manage and manipulate data frames
5. requests - To download web pages
6. numpy - To do various calculations
7. matplotlib - To do various plotting applications
8. json - To decode web request data
9. sklearn - To do machine learning algorithms

2.2 Get the neighborhood names

We are going to analyze neighborhoods of St. Louis, Missouri. There could be multiple sources that I could get this information about St. Louis, but I decided to go with this Wikipedia page. I used Beautiful Soup to scrape the data out from this page. That way, if the neighborhood list is updated in the future, I can still re-run the scripts to find the latest data. The following figure shows a screen shot of the sample of the data table.

Demographics [edit]

Neighborhood	Population	White	Black	Hispanic/Latino[2]	AIAN[1]	Asian	Mixed Race	Corridor
Academy	3,006	16.9	54.7	20.5	1.52	4.3	3.5	North
Baden	7,268	6.3	91.8	0.5	0.1	0	1.3	North
Benton Park	3,532	68.2	25.1	3.2	0.3	1.2	3.8	South
Benton Park West	4,404	28.0	59.6	10.5	0	1.9	5.1	South
Bevo Mill	12,654	74.2	13.8	7.5	0.4	4.6	3.9	South
Botanical Heights	1,037	20.3	74.4	2.1	0.2	1.7	2.6	Central
Boulevard Heights	8,708	89.5	3.6	3.5	0.3	3.6	2.0	South
Carondelet	8,661	57.3	33.8	7.1	0.6	1.3	3.7	South
Carr Square	2,774	0.5	98.0	0.5	0.3	0	0.9	North
Central West End	14,473	58.0	28.0	2.7	0.2	11.1	2.2	Central
Cheltenham	620	67.3	15.0	3.7	0.2	10.6	5.6	Central
Clayton-Tamm	2,251	89.0	6.0	2.6	0.2	2.0	2.1	Central
Clifton Heights	3,074	90.1	3.9	3.0	0.5	1.9	2.4	South
College Hill	1,870	3.7	92.7	1.2	0.5	0.6	2.3	North
Columbus Square	1,869	4.1	92.9	0.9	0.2	0.6	1.9	North
Compton Heights	1,315	71.0	21.3	2.0	0.2	3.9	3.3	South
DeBaliviere Place	3,466	59.0	29.4	3.0	0.2	7.8	2.5	Central
Downtown	3,721	53.5	37.1	2.9	0.5	5.4	2.3	Central
Downtown West	3,940	56.3	36.9	2.6	0.3	3.7	2.9	Central
Dutchtown	15,770	35.6	50.8	9.0	0.3	6.0	3.8	South
Ellendale	1,575	80.6	11.9	7.3	0.4	0.8	3.5	South
Fairground	1,793	1.7	97.1	0.5	0.1	0	1.0	North



Map of the 79 neighborhoods of St. Louis, Missouri

Red brick homes in Gravois Park

A part of Wikipedia page that scoop neighborhood data

After downloading the data, the I created a pandas data frame. and the data frame looked like below. Notice that this is only showing the first five rows of the data for the illustrative purposes. Numerical data are converted into int and float data types rather than having object types. Comparing the wikipedia sample above and pandas data frame below, one can identify that all the required data were downloaded properly using BeautifulSoup.

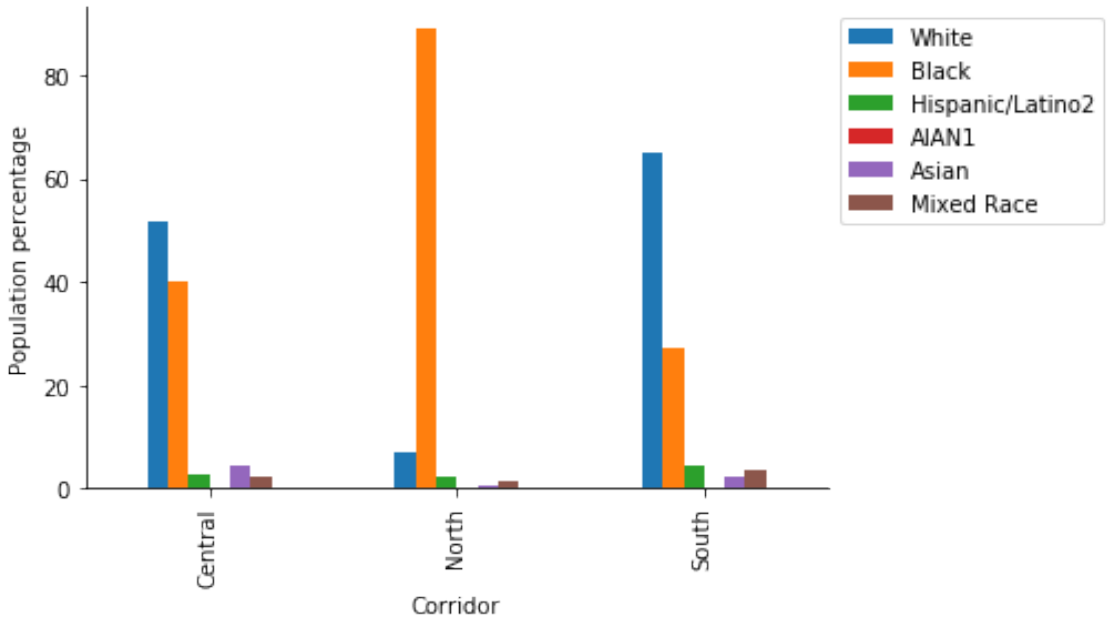
	Neighborhood	Population	White	Black	Hispanic/Latino2	AIAN1	Asian	Mixed Race	Corridor
0	Academy	3,006	16.9	54.7	20.5	1.52	4.3	3.5	North
1	Baden	7,268	6.3	91.8	0.5	0.1	0	1.3	North
2	Benton Park	3,532	68.2	25.1	3.2	0.3	1.2	3.8	South
3	Benton Park West	4,404	28.0	59.6	10.5	0	1.9	5.1	South
4	Bevo Mill	12,654	74.2	13.8	7.5	0.4	4.6	3.9	South

Notice that according to the table above, St. Louis has three regions, north, south, and central. Since the demographic data are there, it made sense to do an initial analysis of those data to find demographic concentration on each of these regions. I decided to find the average percentages of different ethnic groups living in each region mentioned above.

Following data table shows the average percentages in each corridor/region. It looks like most white people live in South side has more white people, north side is dominated by black people, Asian tends to go in the Central region.

	White	Black	Hispanic/Latino2	AIAN1	Asian	Mixed Race
Corridor						
Central	51.820000	40.145000	2.645000	0.210000	4.640000	2.500000
North	7.148387	88.912903	2.306452	0.239355	0.435484	1.470968
South	64.967857	27.175000	4.625000	0.292857	2.389286	3.403571

This tabular data can be nicely visualize using a bar chart as shown below. With that graph it is much clear that white people dominate in north while the black people dominate in the south. The central region is almost equally distributed by both ethnicity. Other ethnic groups are not dominating in any region and create very small contributions.



2.3 Obtain the latitude, longitude, address, and miles to work for each neighborhood

My next step was to find additional data for each neighborhood. For this section, I found Latitude, Longitude, Address, and Distance to the workplace. For this, I created a python function that will give me a pandas data frame with all the information mentioned. Following is a sample of the data frame generated using five random rows of data.

	Neighborhood	Population	White	Black	Hispanic/Latino2	AIAN1	Asian	Mixed Race	Corridor	Latitude	Longitude	Address	miles_to_work
24	Fox Park	2632	32.3	61.2	4.7	0.4	1.2	2.5	South	38.6084	-90.2259	Fox Park, Saint Louis, City of Saint Louis, Mi...	1.92652
51	North Hampton	7892	75.8	15.2	4.1	0.3	4.8	2.5	South	38.5972	-90.281	North Hampton, Saint Louis, City of Saint Loui...	4.87109
21	Fairground	1793	1.7	97.1	0.5	0.1	0.0	1.0	North	38.6669	-90.213	Fairground, Saint Louis, City of Saint Louis, ...	2.85123
63	Soulard	3440	82.6	13.3	2.7	0.2	1.0	2.3	South	38.6045	-90.2093	Soulard, Saint Louis, City of Saint Louis, Mis...	1.64338
5	Botanical Heights	1037	20.3	74.4	2.1	0.2	1.7	2.6	Central	38.6211	-90.2501	Botanical Heights, Saint Louis, City of Saint ...	2.77056

I noticed that Nominatim doesn't always give a valid address for some neighborhoods. This behavior can be improved by including the city and the state with the search query. For the case of St. Louis, I got all the address. But in case incomplete data are there, I decided to drop those lines with incomplete data rather than trying to search for the address. Because the number of neighborhoods without the addresses may not be many and additional information may not not worth the time spend for this case.

I also noticed that some times there were several neighborhoods was identified by the Nomitatim is not even belongs to the state I am searching. I am not sure the reason for these, but I think somehow nominatim is not registering the city and the state of the search query. One way to take care of that find those and remove them individually. However, since I am not interested in any neighborhood more than 30 miles, I can just get rid of those, which should automatically take care of out of the state neighborhoods.

After both of the above processes, the number of neighborhoods still stayed 77. That means Nominatim actually did a wonderful job of searching the addresses of our neighborhoods.

2.4 Obtain venues around each neighborhood

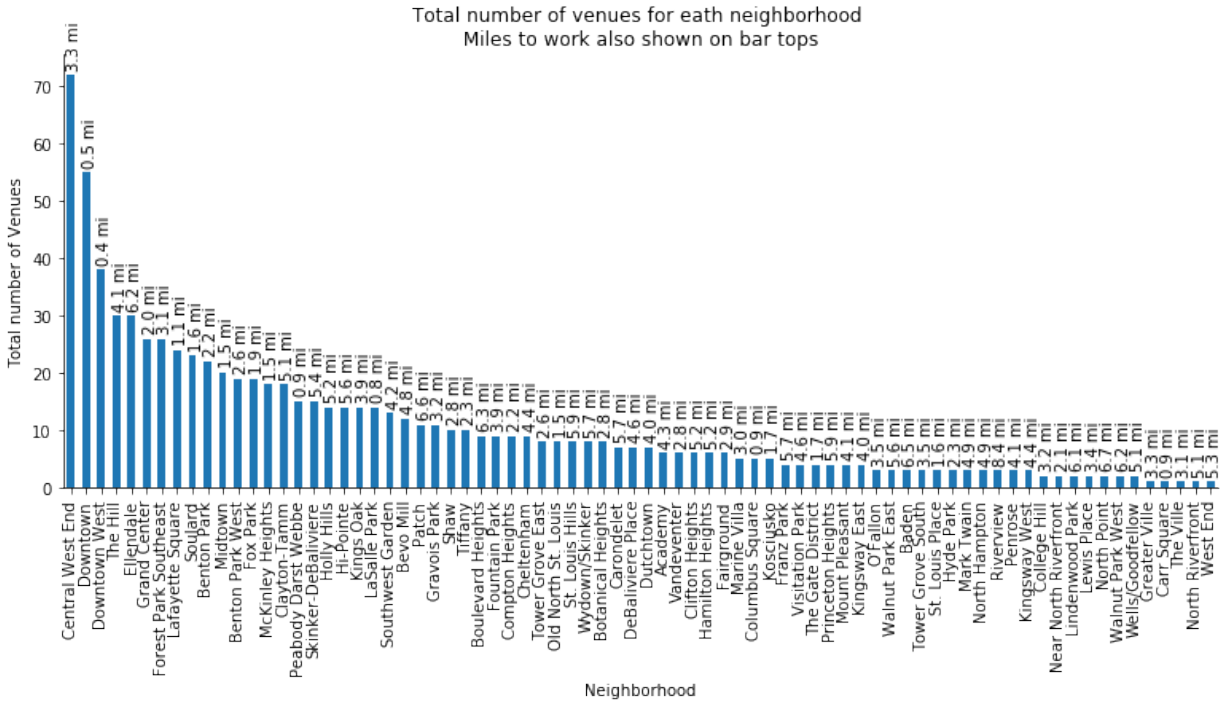
After finding the coordinates, I wanted to find good amenities around each neighborhood. I was using Foursquare api for that. I have a free account with them which allow me to make 100 k free requests per day. I definitely acknowledge their service. I crated another important function accomplish this. What this function does is produce a data frame that contains the information nearby amenities, up to a 100 of those. Following data frame is the result of this function.

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Academy	38.658421	-90.267226	Walgreens	38.660054	-90.272582	Pharmacy
1	Academy	38.658421	-90.267226	Redbox	38.660150	-90.272126	Video Store
2	Academy	38.658421	-90.267226	Union & Page Ave	38.659903	-90.263602	Outdoors & Recreation
3	Academy	38.658421	-90.267226	solls	38.657974	-90.262276	Grocery Store
4	Academy	38.658421	-90.267226	West End Market	38.658640	-90.272224	Convenience Store

It looks like the function returned information about 811 venues while above table is only showing 5 venues. But it is interesting to know how many venues for each neighborhood have. I created new data frame using pandas group by function to above data frame by grouping with the Neighborhood column. This found the total number of amenities around each neighborhood. Since I also like to know the distance from each neighborhood, I added that information to a data frame first.

	Neighborhood	Venue Category	miles_to_work
0	Central West End	72	3.343809
1	Downtown	55	0.505269
2	Downtown West	38	0.438926
3	The Hill	30	4.120770
4	Ellendale	30	6.186064

I plotted data to and see how the results look like. I created bar graph of Neighborhood vs. Total number of Venues. Since I am interested about the distance to work, I displayed miles to work on top of each bar.



According to the graph, it is clear that the top place for a lot of amenities is Central West End which is only 3.3 miles away. Then there is Downtown, Downtown West, etc. But the distance is not the only thing I am looking at. I found that we had these venues belongs to 177 different unique categories.

2.5 Analyze Each Neighborhood to find how many unique catogories are belongs to each neighborhood

My goal was to find the best neighborhood that has so many different amenities close by. I was planning to use the k-means algorithm. For that, I wanted to create another data frame that has all our neighborhoods in the first column and frequency of different categories of amenities in the next columns. So I should have columns equal to the number of different categories I found from the previous section + 1 (for the Neighborhood name column). Let's creae a dummy data frame and then we will fill the fields with the mean.

It turns out that different neighborhoods have different strength as the top comment venue for each neighborhood is different. For example, if you are a person who likes Music you might want to go to Belleair, but if you are someone you want to access to the gym every day, you might find Bunker Hill is better. Did I lose you for a moment? Ok, let me show what I mean. Let me show you what are the best 10 common venues near each neighborhood. See below.

	Neighborhood	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
0	Academy	Pharmacy	Chinese Restaurant	Convenience Store	Grocery Store	Video Store	Outdoors & Recreation	Fast Food Restaurant	Farmers Market	Farm	Falafel Restaurant
1	Baden	Food	Fast Food Restaurant	Bar	Yoga Studio	Food Truck	Filipino Restaurant	Festival	Farmers Market	Farm	Falafel Restaurant
2	Benton Park	Brewery	Dive Bar	Beer Garden	Bakery	New American Restaurant	Pizza Place	Cocktail Bar	Massage Studio	Sandwich Place	Café
3	Benton Park West	Mexican Restaurant	Pizza Place	Intersection	Bakery	Locksmith	Convenience Store	Art Gallery	Music Venue	Restaurant	Taco Place
4	Bevo Mill	Bar	Restaurant	Lounge	Mexican Restaurant	Arcade	Discount Store	Bed & Breakfast	German Restaurant	Food	Taco Place

] I think by looking at the table above you might understand that different neighborhoods have different amenities strength. The best neighborhood depends on which kind of amenities are around you and what your choises are.

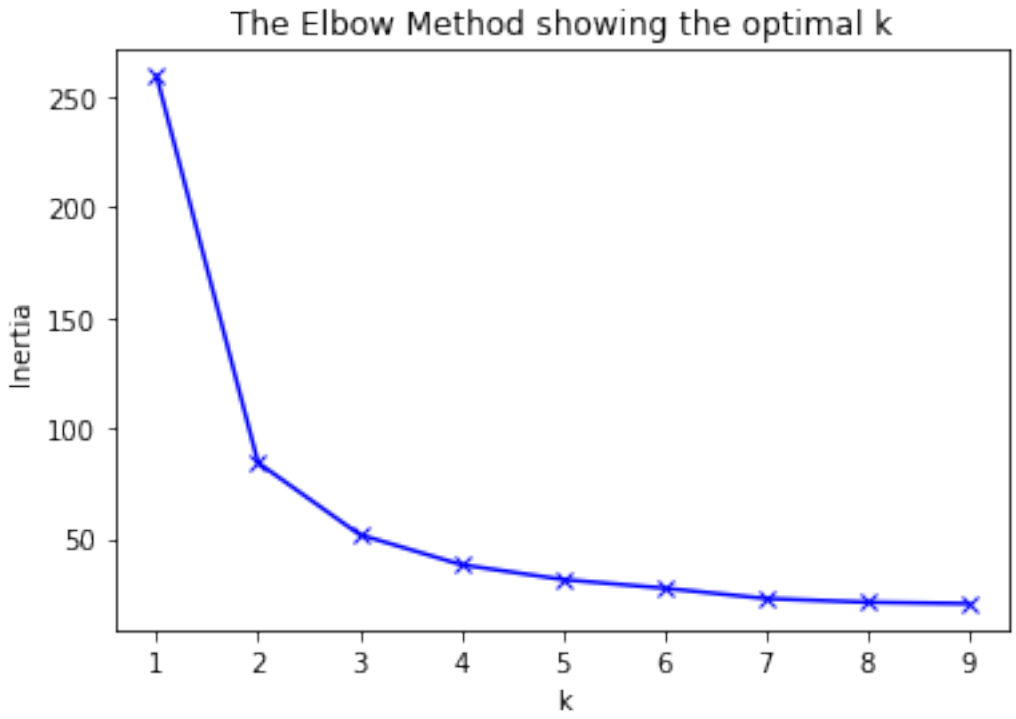
With this, I am done with my data preperation and preliminary analysis of the data. Let's move on to real machine learning.

3 Methodology - K-means Clustering

For this study, I used K-means clustering algorithm. Since we have unlabeled data, I think this will be a very good starting point to do an unsupervised algorithm. However, I was not 100 % sure how many clusters to choose. I needed to do some calculations to find the best number of clusters

3.1 Choosing the best cluster number using elbow method

One way to find the right number of for k is use of elbow method. I started from K = 1 and go up to K = 10 and did k-mean clustering for each K value. To find the best K value, I compared the resulted inertia or within-cluster sum-of-squares value. The following figure shows the Inertia vs k value plot.



As the number of clusters increases, the centroids becomes closer to their clusters. This makes the distortion decrease as you increase K. You will get he minimum distortion when the number of clusters is exactly equal to the number of data points. At that case the distortion becomes exactly zero. But notice the graph has a sudden variation of the slope at k = 4. Therefore, according to the Elbow method, we will consider this point as the best number of clusters as beyond this point the improvement of distortion is minimal.

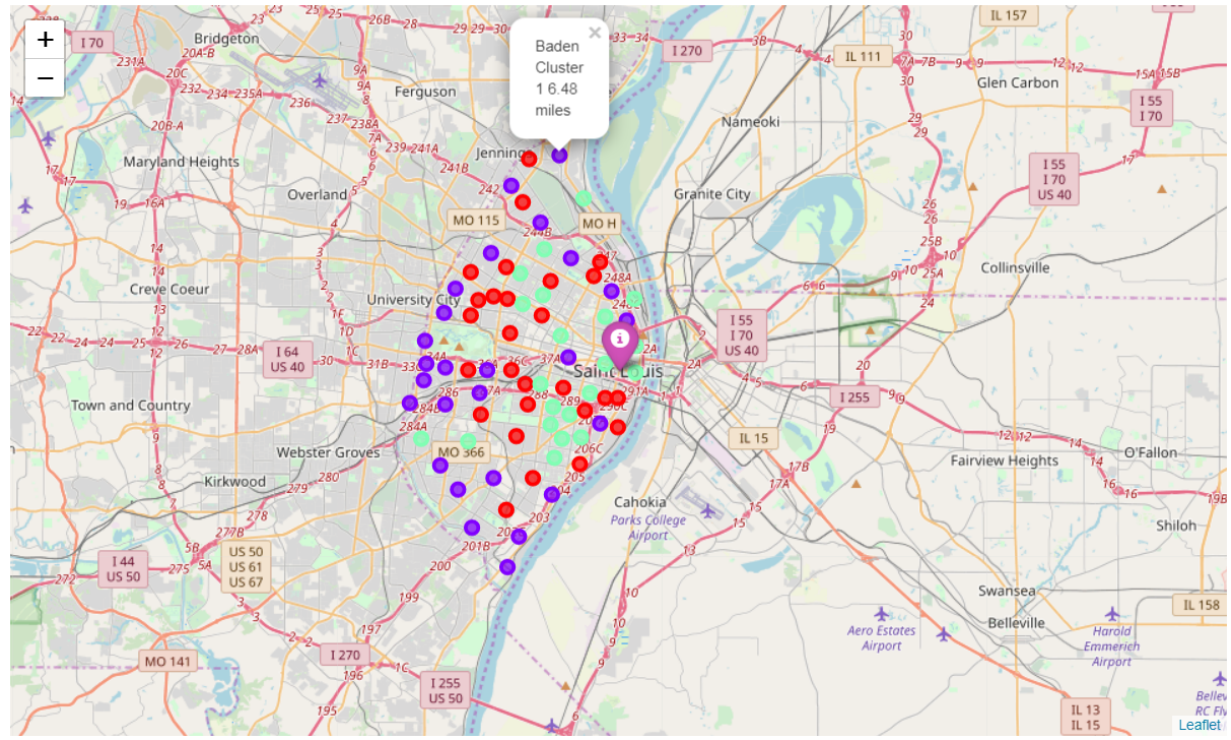
3.2 Reanalyze data with best K

Let's rerun the K-mean method with k = 3. According to the graph, one can argue that k = 2, is more accurate. However, since the Distortion change is significant from k = 2 to k = 3, I choose k = 3. I found that resulted clustering labels for each cluster. The labels are nicely randomly

distributed. This was actually a good sign that K-means was appropriate and working well for this problem. This can be further confirmed by plotting all data points in the map. Just wait for it.

3.3 Creating an interactive map to show the neighborhoods and their cluster numbers

Nothing beats to the interactive map of the different neighborhoods. Folium is great at producing an interactive map. Following figure showing an image of the map produced. Different color dots in the map represent the location of neighborhoods analyzed. The colors correspond to different k-mean groups. The balloon icon close to the middle of the map is showing us the location of my workplace.



The map showing above was highly interactive. One can zoom the map to see more details around a particular neighborhood. If you click on a colored dot, it can tell you the information about that particular point. I programmed so that when you click on a point it will show us the name of the neighborhood, cluster id, and distance to work. In the figure above, I clicked on the neighborhood 'Baden'. This neighborhood belongs to category 1 and situated in 6.48 miles from my prospective workplace.

4 Results

The original problem was to find the best neighborhood to have less commute time and easy access to close by venues. From a Wikipedia page, we found that there are 79 neighborhoods around St. Louis, Illinois. The initial analyses of the data suggested that there are 3 main regions of St. Louis and different ethnic groups dominated in different areas. North populated with black and the south is mainly occupied by white. It is not unusual to see that the central region is roughly populated by both groups. Other ethnic groups are scattered across all areas.

When I try to find the locations of the neighborhoods using geolocator app, it gave me no results for two neighborhoods. I could have obtained the locations for those two neighborhoods manually. Since it is just less than 3 % of the total data, I decided that not to put time on that. On the other hand, I like to use this learning algorithm for other towns. Therefore, it might be wise to stick to programmable solutions.

Then I obtained the amenities around each neighborhood. I found that the total number of amenities around each neighborhood ranges from 1 to ~70. According to the plot I created, the Central West End neighborhood has the best amenities around it with only 3.3 miles away from the workplace. There were a total of 811 venues covering 177 unique categories.

Then I found the frequency of each venue type belongs to each neighborhood created a new data frame. After adding the miles to work data column to this data frame, I used to use this as the input for the K-means algorithm.

I decided to use the K-means algorithm because of it very simple and fast for finding similar features in unlabeled data. I search through different values of Ks. Using the elbow method, I found that K = 3 would give us the optimum results. All 77 neighborhoods were divided into 3 categories by K-means.

4.1 4.1 Average cluster results

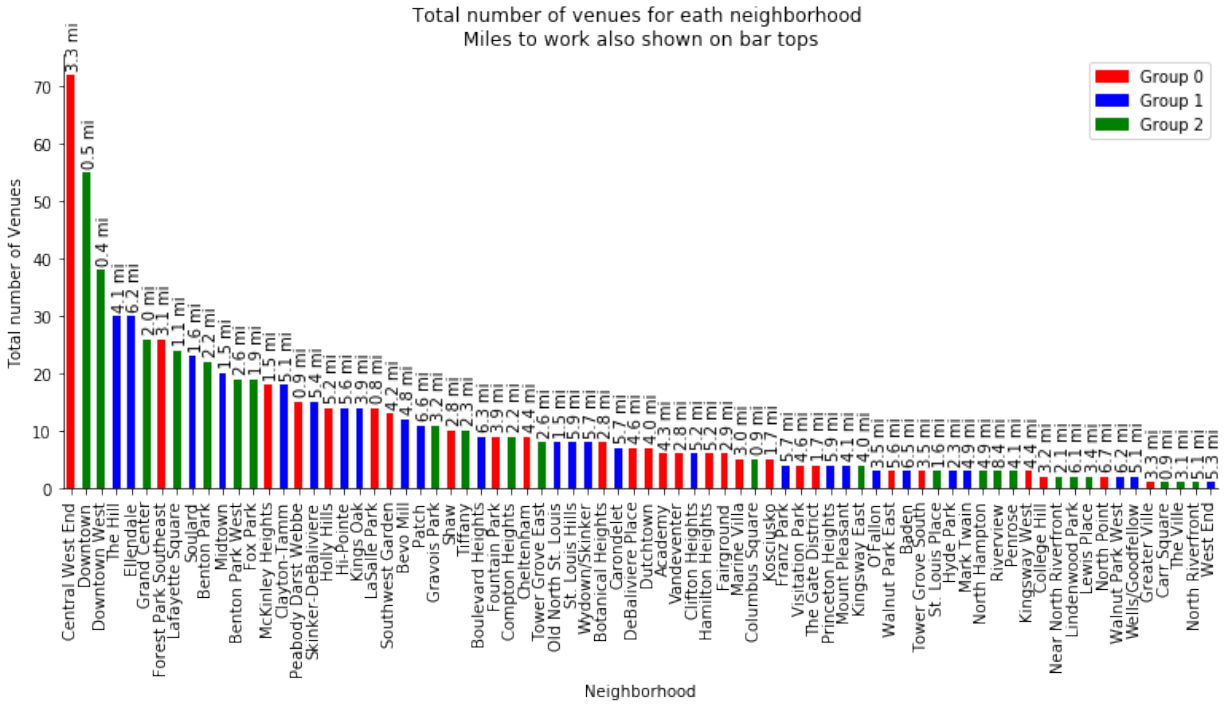
Let's investigate these three categories (cluster labels) in more details by averaging different numbers of data.

Cluster Labels	Venue Category	miles_to_work	Population	White	Black
0.0	10.296296	3.495589	4076.481481	27.970370	64.496296
1.0	10.076923	4.788743	4542.884615	55.926923	37.973077
2.0	11.782609	2.851308	3371.565217	33.813043	60.439130

Notice that, I did not give the demographic data as input. The reason I didn't do it because original data suggested that different ethnic groups are already concentrated into some parts of the town, north -black, south -white, center -mixed. But it very interesting to notice that both K group 0 and 3 are black-dominated, K Group 1 is black and white roughly equally distributed. Notice that K groups are nicely scattered around the map. If any, one can expect that all ethnics should have mixed numbers. Yet, different K-clustering have their own identical demographic variations, which is definitely unexpected and very interesting. In terms of the miles to work, K group 2 have an edge. On the other hand, the total number of venues are about the same for all three categories. **According to these results, my personal choice is a K - 2 group due to the closer distance to work and more close by venues.**

4.2 Graphical representation of best Neighborhoods

Let's try to visualize this graphically. I created a bar graph, similar to the one I created during the data section of documents. But this time, other than showing the miles to work on top of each bar, I like to color the bars to show the different groups, as shown in the figure bellow.



Since I like group 2 and closer distance to work, I would choose Downtown as my top spot which has 50+ closeby venues and only half a mile away from the work. If I do not like it, then I go for Downtown West, Forest Park South East, etc.

5 Discussion

The K-means algorithm was successfully able to suggest the best neighborhood to live around St. Louis. The best neighborhoods I choose are showing in green in the figure above. My choise of best neighborhood based on the number of close-by venues and closer distance to the workplace.

While demographic data were available for each neighborhood, I have not used those data in this study. However, one might consider that it a deciding factor and should be included in this study. There could be multiple other things I didn't consider. Some of those things could be access to a good public school, the ratings of the closest public school, distance to a bus station, a train station, or an airport, crime rates around the neighborhood, etc. If were to include those data, a more completed model can be developed.

Also, in the east of St. Louis there is the Illinois -Missouri border. The neighborhoods towards east which are close to St. Louis but not belongs to Missuiri are not included. I think these neighborhood can be included in the study.

6 Conclusion

Using publicly available data and K-mean machine learning algorithm, I was able to find several prospective neighborhoods. It is incredible that without visiting a faraway town, we can do a detailed analysis.

I choose location data of venues nearby every neighborhood in St. Louis to figure out the best place to live. One of the requirements I have is to live close by to save some driving time. However, there is a huge room to improve this algorithm by including additional data like school ratings, crime information, distance to public transportation, etc. Not only adding data but also eliminating redundant data is also important by investigating each features more carefully.

All in all, I am very comfortable that this machine learning K-clustering was very successful at providing me the best neighborhood to live in a foreign city.

7 Acknowledgment

This project was not possible without instructors from IBM - Coursera, therefore I thank all the IBM instructors who teach the courses in Data Science Professional Certificate. I also thank free data providers, Wikipedia and foursquare. This application was initially developed using IBM cloud and Watson free account and later completed in Google Colab project. I thank both of these services.

8 References

1. Main data obtained from the site:
   'https://en.wikipedia.org/wiki/List_of_neighborhoods_of_St._Louis'
2. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011
3. Map data provider: https://foursquare.com/