

Contents

[Introduction 2](#_heading=h.gjdgxs)

[Target Audience 2](#_heading=h.30j0zll)

[Use Case Example 3](#_heading=h.1fob9te)

[Data 3](#_heading=h.3znysh7)

[Postal Code Data Sources 3](#_heading=h.2et92p0)

[Methodology 3](#_heading=h.tyjcwt)

[Loading in the Data 3](#_heading=h.3dy6vkm)

[Cleaning the Data 3](#_heading=h.1t3h5sf)

[Visualizing the Neighbourhoods 4](#_heading=h.4d34og8)

[Calgary 4](#_heading=h.2s8eyo1)

[Ottawa 4](#_heading=h.17dp8vu)

[Montreal 5](#_heading=h.3rdcrjn)

[Vancouver 5](#_heading=h.26in1rg)

[Venue Data 6](#_heading=h.lnxbz9)

[KMeans Clustering 6](#_heading=h.35nkun2)

[K=10 7](#_heading=h.1ksv4uv)

[K = 15 8](#_heading=h.44sinio)

[Getting Similar Neighbourhoods 8](#_heading=h.z337ya)

[Results 8](#_heading=h.3j2qqm3)

[Using the Model 11](#_heading=h.1y810tw)

[Discussion 12](#_heading=h.4i7ojhp)

[Conclusion 12](#_heading=h.2xcytpi)

# Introduction

When moving from one city to another, it’s often difficult to know what each neighbourhood is like and house-hunting can feel like you're going in blind. This is particularly relevant for people who need to relocate for work but have never been to the city they're relocating to. In this project I want to analyze the neighbourhoods of 4 Canadian metropolitan cities (Vancouver, Montreal, Ottawa, and Calgary) to find which neighbourhoods in different cities are most similar based on venues.

### Target Audience

The target audience for this solution is anyone who is moving or thinking about moving from one of these cities to another, but I think the people who would find it most helpful are those who have never been to the city they are moving to. For this project I chose 4 metropolitan cities because I think the largest audience for this solution are people who live in urban areas, and I chose Canadian cities because I'm most familiar with them personally, but a complete solution would involve analysis of every city in the world.

### Use Case Example

Let's say that I am really like the Church and Wellesly neighbourhood from Toronto and I have to move to Vancouver from a job, but I really don't know much about what area I'd like to live in there. This solution will tell me which neighbourhoods in Vancouver are most similar to Church and Wellesly so that I have a better idea at what real estate to look at.

# Data

I found a website online that contains all the neighbourhood names, postal codes, boroughs, and cities for all of British Columbia, all of Alberta, all of Ontario, and all of Quebec. I have scraped those pages and then dropped all rows that refer to cities other than Vancouver, Calgary, Ottawa, or Montreal. I will also be using foursquare to extract the most common venue types in each neighbourhood, which will be used as a feature to help determine similarity when training my model.

I had initially planned to also use demographic data from the 2016 Census, which is available for download from Stats Canada, but I discovered that this data isn’t split up by neighbourhood (except for the Toronto Census data) and so I’m unable to use it for this project. I also tried to find weather data but, again, it wasn’t separated by neighbourhood so it ended up being irrelevant to this project.

### Postal Code Data Sources

Ottawa: <http://www.geonames.org/postal-codes/CA/ON/ontario.html>

Vancouver: <http://www.geonames.org/postal-codes/CA/BC/british-columbia.html>

Montreal: <http://www.geonames.org/postal-codes/CA/QC/quebec.html>

Calgary: <http://www.geonames.org/postal-codes/CA/AB/alberta.html>

# Methodology

## Loading in the Data

I had initially planned to scrape the webpages using BeautifulSoup but the formatting didn’t work out so instead I manually copy and pasted each table into an excel document, saved them as csvs, and read the csv data into a dataframe using read\_csv.

## Cleaning the Data

First, I created a list for each city that contained all postal codes corresponding to that city, then I wrote a function that would drop all rows with invalid postal codes, rename the column “Place” to “Neighbourhood”, and print out how many neighbourhoods were found for each city. I also had to make sure to use the encoding=’latin1’ parameter when reading the csv files, because otherwise French accent characters that were in the names of Montreal neighbourhoods weren’t getting read properly.

## Visualizing the Neighbourhoods

Next, I created a function to map all the neighbourhoods for each city using geopy and folium.

### Calgary

Map

Description automatically generated

### Ottawa

Map

Description automatically generated

### Montreal

Chart, map

Description automatically generated

### Vancouver

Map

Description automatically generated

## Venue Data

Using foursquare I was able to get all of the most common venues for each neighbourhood. I then used onehot encoding to normalize the values and prepare them for analysis. Here’s an example of the Ottawa dataframe:

A screenshot of a computer

Description automatically generated with medium confidence

Afterwards I added a column called ‘Metro Area’ to each dataframe that specified which city the dataframe was referring to, and then I merged all 4 dataframes using pd.concat.

## KMeans Clustering

For my model, I decided to use Kmeans clustering to determine similarity. If this were a full-length project, I would have wrote something more intricate somehow involving Euclidean distance to determine similarity, but after doing research about this it seems like Kmeans is the best built-in way to achieve close to what I was looking for.

I tried using the elbow method to determine the optimal k, and this is the graph I found.

Chart, line chart

Description automatically generated

Based on this, it’s hard to determine the optimal k, so I decided to run analysis for k=10 and k=15. Visualizing the maps of the neighbourhoods, coloured by cluster, this is what I received.

### K=10Map Description automatically generated

Chart

Description automatically generatedMap

Description automatically generatedMap

Description automatically generated

### K = 15Map Description automatically generatedMap Description automatically generated

### Map Description automatically generated

Map

Description automatically generated

The problem I ran into is that with a large number of clusters, most clusters only contained data from one city, which doesn’t work for my use case. For that reason I decided to run kmeans with only 6 clusters, despite the results from the elbow method for determining k.

## Getting Similar Neighbourhoods

Finally, I wrote a function that, when the name of any valid neighbourhood is input, will return all other neighbourhoods in the cluster. I also added an option to exclude all results from the cluster that were in the same city as the given neighbourhood, to tailor it more to the use case of someone who is moving from one city to another.

# Results

As discussed in the previous section, I decided to create my final kmeans model with k=6. Below are maps of each city with neighbourhoods coloured by clusters from my final model.

Montreal:

Map

Description automatically generated

Ottawa:

Map

Description automatically generated

Calgary:

Diagram

Description automatically generated

Vancouver

Map

Description automatically generated

## Using the Model

Let’s say that I live in the “Nepean East” neighbourhood of Ottawa and I would like to know what neighbourhoods in other major Canadian cities are similar to where I live. Using this code:



I can receive a list of all similar neighbourhoods along with their 10 most common venues.

A screenshot of a computer

Description automatically generated with medium confidence

Now let’s say that I really want to move to a neighbourhood like Petite-Patrie Northeast in Montreal, I don’t mind if I also see results in Montreal, but I only want a list of similar neighbourhoods, no venues. In that case, I can use this code:



And I will receive a list of similar neighbourhoods, including neighbourhoods in Montreal:Text

Description automatically generated

# Discussion

Based on the map visualizations and through looking at the dataframes for each cluster, we can see that neighbourhoods in Ottawa and Calgary are very similar to each other. We can also see that North Vancouver is very different than South Vancouver in terms of venues, and that the centre cores of every city have some similarities. The major downfall of this method is that not every cluster has neighbourhoods from each city. 2 of my clusters, for example, contain only one neighbourhood each and therefore wouldn’t be helpful at all if either of those were the neighbourhoods a user wanted to look at.

# Conclusion

In conclusion, when given the name of a neighbourhood we are able to get a list of neighbourhoods either in the 3 cities other than the current city of that neighbourhood or in any of the 4. There are issues with the model, such as 2 clusters only having one neighbourhood and another cluster being almost entirely made up of one city, but all in all I think it’s a good first step. The next steps for this model would be to create a function that uses Euclidean distance to find the mostsimilar neighbourhood given certain criteria (like given the city a user wants to move to as criteria). This way, even neighbourhoods that aren’t similar to any other neighbourhood, such as those that are alone in their clusters, will still show the *most* similar result.