# Toronto Battle of Neighbourhoods

## Tri Juhari

### February 10, 2021

**1. Introduction**

**1.1 Background**

Toronto is the capital city of the Canadian province of Ontario. With a recorded population of 2,731,571 in 2016, it is the most populous city in Canada and the fourth most populous city in North America.

Toronto encompasses a geographical area formerly administered by many separate municipalities. These municipalities have each developed a distinct history and identity over the years. Former municipalities include East York, Etobicoke, Forest Hill, Mimico, North York, Parkdalse, Scarborough, Swansea, Weston and York. Throughout the city there exist hundreds of small neighbourhoods and some larger neighbourhoods voering a few square kilometers.

**1.2 Problem**

I am currently living in a small neighborhood in the borough of West Toronto, called Dufferin, Dovercourt Village. I love where I live because of all the amenities it has to offer me, such as restaurants, gyms, parks and so on.

I work as an operations manager in a successful company located in the same neighborhood that I am living in. The problem comes when they offer me a “dream” promotion which the job location of it is not the same as the actual. The job neighborhood is in Scarborough, and it’s called Cedarbrae.

This neighborhood is approximately 19.58km from Dufferin, my actual neighborhood. What it means is that I should move if I want to take this job opportunity. Therefore I want to make a data driven decision and find the optimal neighborhood between two parameters: nearness with the job location and similarity in relation to my actual neighborhood.

**1.3 Interest**

The personal interest of getting a promotion where I work and also can move to a neighborhood which is near the job location and is similar to my actual neighborhood, which I love.

**2. Data acquisition and cleaning**

**2.1 Data Sources**

Different data sources compose the final dataset. Starting from Wikipedia’s article: [List of postal codes of Canada: M](https://en.wikipedia.org/wiki/Postal_codes_in_Canada), getting the data via parsing the HTML code, and where the features name of neighborhood, postal code, borough were extracted.

From there, using the geolocator in the package *pygeo*, and the postal code of each neighborhood, we could obtain the latitude and longitude belonging to each respective neighborhood. Also were deleted the wrong values with NaN type assigned. The neighborhood distributions following the geolocation looks like the next image:

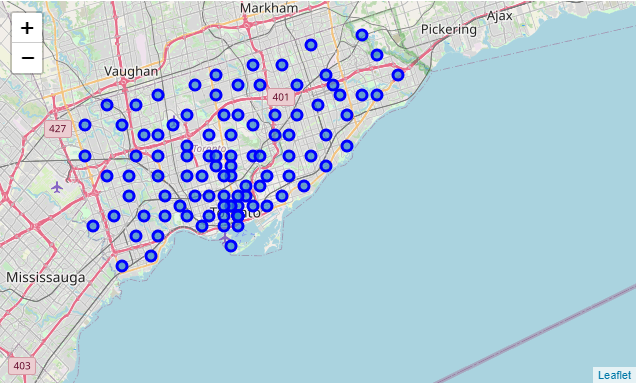


Figure 1.1: Neighbourhoods distribution in Toronto City

Applying the jaccard distance provided by *geopy*, and passing as parameters the longitude and latitude of the neighbourhoods, we got the distance between the job neighborhood and each single neighborhood in kilometers. In example, the neighborhood Parkwoods, located in North York, is 7.58 kilometers from Cedarbrae.

Already with the neighbourhood's location, the next step was to get the first 100 venues in a radius of 500 meters with respect to each neighborhood. This was achieved with the [Foursquare API](https://foursquare.com/) which provides a different useful information in relation with venues. From these data we extracted the following features: venue category, venue name, venue address, venue id (API id), category id (API id), venue latitude and venue longitude.

**2.2 Data cleaning**

All the data was merged into a single dataset. The data extracted from Wikipedia HTML table had a lot of insignificant places with values “not assigned”, so all of these values were removed.

Then, when extracting the location of the neighborhoods, some of the postal code query returns NaN values, so they also were removed after analysing the impact of these in the data.

Finally, visualizing the neighborhoods segmenting by borough, we could see the wrong value corresponding to Business reply mail Processing Centre, South Central Letter Processing Plant Toronto. As we can see in the map (look at figure 2.1), between the light green neighborhoods (Scarborough borough) there is a cercle with other colors.

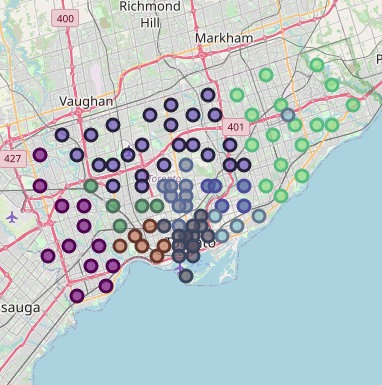


Figure 2.1: Wrong neighborhood assignation.

Business reply mail Processing Centre, South Central Letter Processing Plant is not even a neighborhood, but a mailing service, so also was removed from the dataset.

**2.3 Feature selection**

The following features were removed: venue id, category id, venue latitude, venue longitude and venue address. This is because these features are useful to extract more and more data (for example the venue id is useful to extract more data from Foursquare API) which is out of this scope.

Then, using the One Hot Encoder technique we got the boolean values of the category type by neighbourhood. This allows us to do Exploratory Data Analysis on the frequency of venues categories in the neighbourhood.

We also kept the distance between the job neighbourhood and each neighborhood with the motivation of discard options that are further away than Dufferin, the current neighborhood.

**3. Exploratory Data Analysis**

**3.1 Understanding distances**

In the image of the map (see figure 3.1) we can notice the distance between our current neighborhood (Dufferin, Dovercourt Village) which is in color red and the neighbourhood where the job is (Cedarbrae), in color green.

Also we can remark the neighbourhoods near Cedarbrae in color yellow. These neighbourhoods belong to the same borough as Cedarbrae, Scarborough.

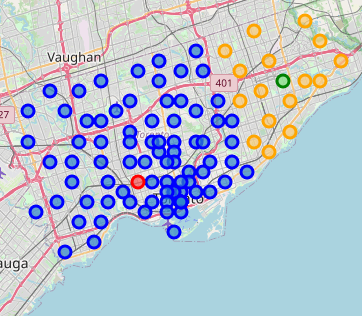


Figure 3.1: in red Dufferin and in green Cedarbrae.

**3.2 Frequency of venues categories in Dufferin**

One of the two parameters to find the “optimal” neighbourhood was the types of venues in it. So it’s important to know which kind of venues are more frequent in the current neighbourhood, Dufferin. To achieve this we can obtain the mean of the One Hot Encoder values of the category name feature, and plot it in a barchart.

In the chart (see figure 3.2) we can notice that the most frequent venue category in Dufferin is the kind Bakery, with a frequency approximately to 0.12. Then also we can remark that the other 9 categories have the same frequency, it says that are the same number of venues for each category.

The frequency value of the Bakery category is twice the other nines top categories. This means that the Bakery category will have more weight at the time of decision.

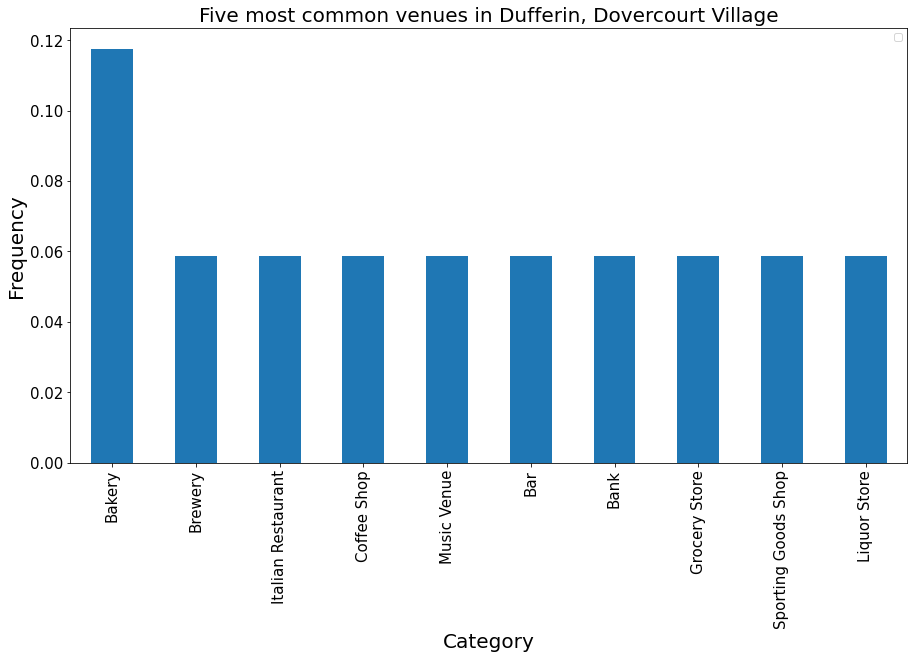


Figure 3.2: venue category type frequency in Dufferin

**3.3 Data Points distribution**

To visualize the distribution of the data points, we should reduce the number of features because at this point it is 250. So to accomplish it, we use the Principal Component Analysis (PCA) to reduce the dimensionality of the dataset (only to plot the data points into a 3d chart).

In the figure 3.3 we can notice that the points are very near each one with each other. This will mean that when we are clustering there are a lot of data points belonging to just one cluster, and the others with less data points.

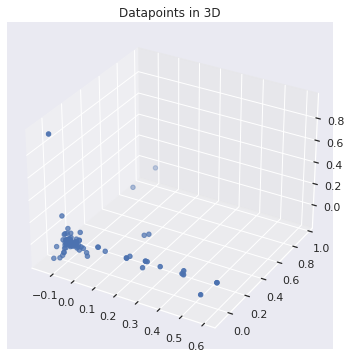


Figure 3.3: datapoints 3d representation

**4. Predictive Modeling**

**4.1. Defining the type of modeling**

The idea is to classify the neighbourhoods according to their venues similarity with unlabeled data. So this problem can be solved through clustering techniques. As we saw in the EDA section, we could notice that the majority of the data points are close to each other, this means that one cluster probably is gonna have more elements belonging to it than the others.

In this case it is going to be used a partitioning clustering technique called K-Means, which will return us *k* number of different clusters. In the next subsections we are going to determine which value of k returns the best score, it says which number of clusters fit better with our solution.

**4.2. Metrics measurement**

As we don’t have an external evaluation to measure the performance of our model, we have to appeal to internal validation to check the quality of the K-Means technique. For this case I decided to use the silhouette score. This score ranges from -1 to 1, where 1 is the best score.

The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). It is calculated by the formula shown below (look at figure 4.1) where is the current cluster, a single data point belonging to and depicts the distance between the data points and .



Figure 4.1: silhouette score formula

**4.3 Finding the best number of clusters**

In this part the idea is to find the optimal number of clusters or groups to solve our problem. This number will mean the quantity of different groups that the neighbourhoods will be assigned and separated.

To achieve this, it is going to iterate over a range of values predefined (in this case I choose between 2 inclusive to 9 exclusive), train the model with that different value in each iteration and save the silhouette score to then compare it.

After this, we got 7 scores respective to each *k* value, where the highest one was when we used 4 differents clusters, as we can see in the chart below (see figure 4.2).

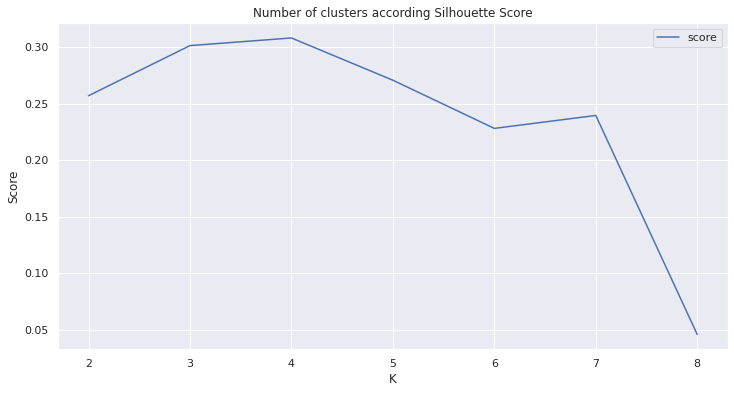


Figure 4.2: score given by the number of clusters

**4.4 Dimensionality Reduction**

At this point, the number of features are two hundred and fifty. This is not only operationally expensive, but also can impact in the loss of the model. Then, to improve the performance of our cluster technique it is going to reduce the number of features through Principal Component Analysis (PCA) method, which will return new features from the original ones.

To determine the number of the features that PCA returns, we can evaluate it and see where the variance of it is between 95-99%. In this way, we can watch how much is the optimal number of features. In this case, was 45 new features (see figure 4.3)

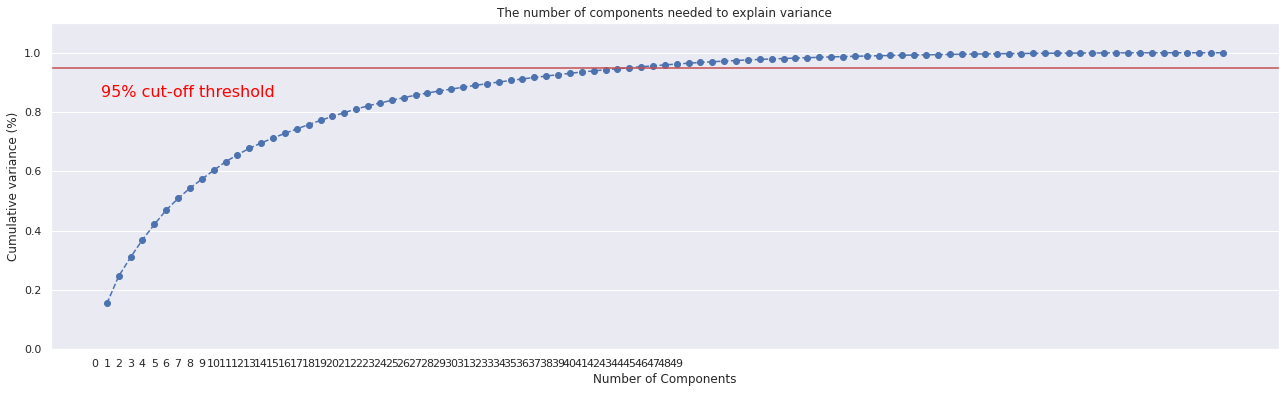


Figure 4.3: variance of PCA in relation with the number of features.

Finally, with the final 45 new features, we trained again the K-Mean algorithm with different values of *k* and we got new scores values, where again 4 was the best option to set the number of clusters (see figure 4.4).

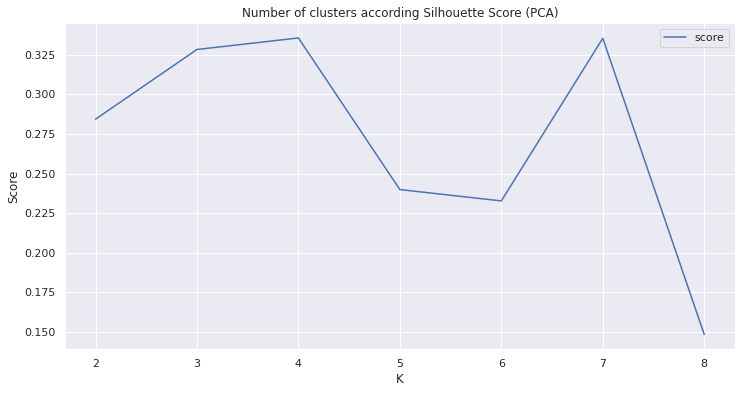


Figure 4.4: silhouette score after PCA transformation

**4.5 Labeled clusters**

After the model training, now we assign to each neighbourhood the respective label given by the clustering returns. As a result we got that Dufferin, my current neighbourhood, belongs to a different cluster than the job neighbourhood, Cedarbrae. Then we have to discard this neighbourhood as a possibility to move in because it does not have complex similarity with Dufferin.

The clusters can be watched in the map at figure 4.5, where each color represents a different cluster. Dufferin is with the color red, and Cedarbrae with light blue.

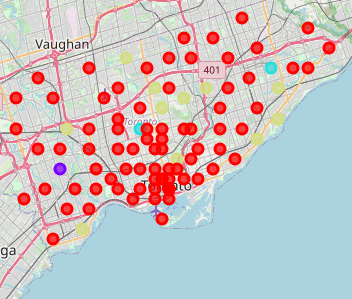


Figure 4.5:neighbourhoods by clusters

**4.6 Cutting distances**

As we are moving because of the distance between the new job and my current neighbourhood, it’s logical to eliminate from the options the neighbourhoods that are further than 19.58km (the actual distance between Dufferin and Cedarbrae). But we also can put a threshold of radius from the job neighbourhood where we discard the options that are further than that radius. In this case a threshold of 12.57km. The final results were a total of 29 neighbourhoods (see figure 4.6).

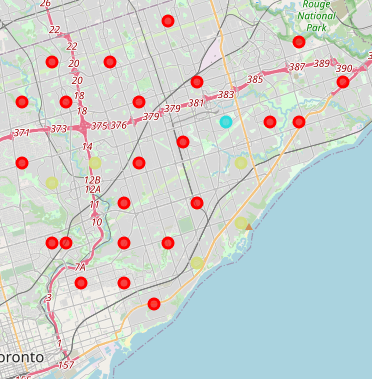


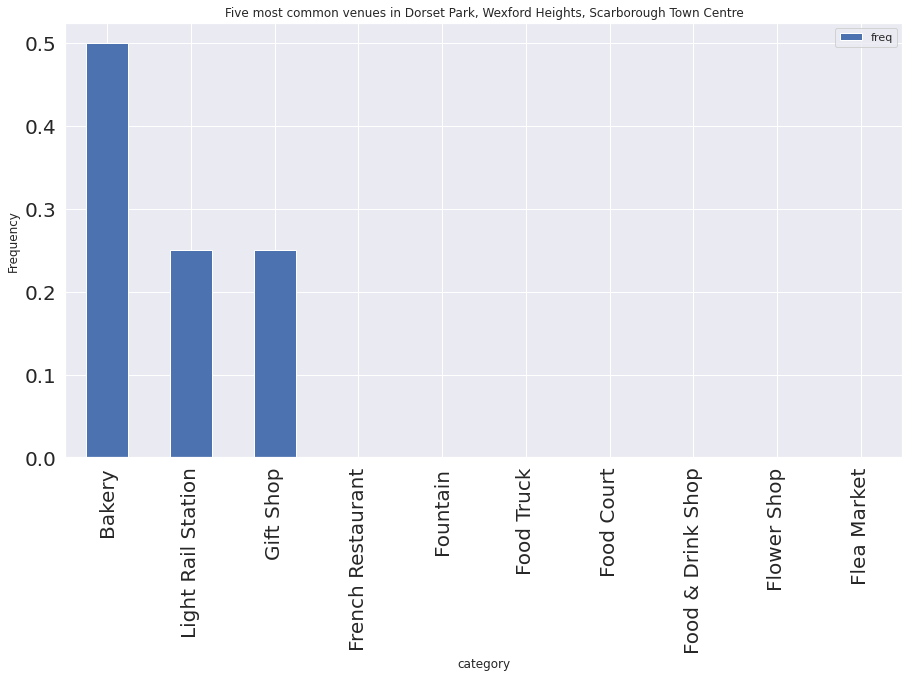
Figure 4.6: neighbourhoods between a radius of 12.57km

**4.7 Selecting the best neighbourhood**

We already have the different neighbourhoods clusters, and also we can notice that near to Cedarbrae there are various neighbourhoods of the same cluster than Dufferin. Now it remains to select which are the best options. The three neighborhoods closest to work that also belong to the cluster of Dufferin are:

* Dorset Park, Wexford Heights, Scarborough Town Centre
* Woburn
* Agincourt

For example, for Dorset Park, the most common venues are depicted in the following chart:



It makes sense that this neighborhood belongs to the same cluster of Dufferin because as we saw in earliest sections, the most common venue (twice more than the other most common) in Dufferin was also Bakery.

**5. Conclusion and Summary**

With the data extracted from trusted sources, we could extract insights from it that they allowed us to understand the venues and neighbourhoods in Toronto City. With this data cleaned and processed we train a clustering model using the most common venues in each neighbourhood and it returns us 4 different clusters to assign to the neighbourhoods.

Finally we notice that are several different neighbourhoods with the parameters we were looking for:

1. Short distance with job location
2. Similarity with actual neighbourhood