Detecting Potential Business Improvement Areas in Toronto with Machine Learning

Diego Caballero Loza

July 2020

1 Introduction

1.1 Background

According to the City of Toronto, a Business Improvement Area (BIA) is an "association of commercial property owners and tenants within a defined area who work in partnership with the City to create thriving, competitive, and safe business areas that attract shoppers, diners, tourists, and new businesses". By working together as a BIA, local businesses have the capacity to improve the quality of life in their local neighbourhood and the city as a whole.

Currently there are 80 BIAs across the city, which is the largest number of BIAs of any urban centre in the world. In conjunction, they generate more than $34 million in funding towards civic improvement. From 2001 to 2018 the city has increased its number of BIAs by 38, which marks the success of the BIA program and its continued growth.

Proven by the success of the program, a BIA can revitalize and enhance a neighbourhood and help owners improve their businesses. Therefore, it is advantageous for business owners to have a tool that can aid them in their decision of becoming a BIA.

1.2 Business Problem

The city of Toronto has multiple data sets that can help determine the volume of people walking through a neighbourhood, and the geographical boundaries of neighbourhoods and BIAs. This, in addition with venue data provided by Foursquare Places, can help determine which areas of the city might benefit from having a BIA in their local neighbourhood.

The goal of this project is to leverage geographical and venue data to aid commercial property owners and tenants with the process of defining new BIAs based on the location and type of businesses in the city. To achieve this goal, I used DBSCAN clustering to cluster venues outside BIAs and then trained an Autoencoder to see if any of the clusters have the potential of being part of a BIA.

2 Data

For this project, three data sets from the City of Toronto's Open Data Portal were used to leverage the Foursquare Places venue data. The first data set contains information about pedestrian and vehicle volume across intersections throughout the city. The second data set contains geographical data for neighbourhood profiles in the city. Lastly, the third data set contains the geographical data for the Business Improvement Areas in the city.

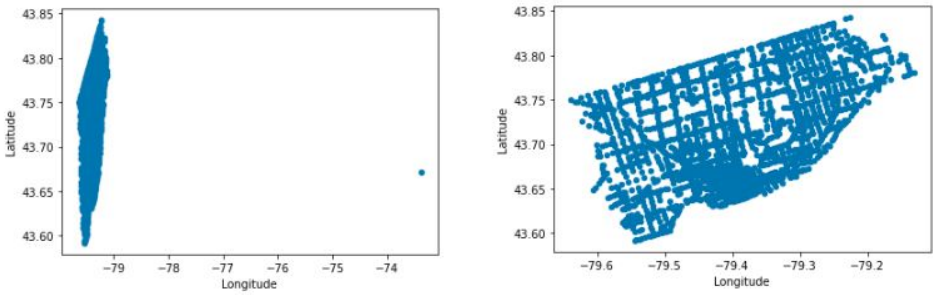
2.1 Pedestrian Volume in the City

2.1.1 Source

This data set contains the most recent eight-hour peak pedestrian volume counts collected throughout intersections in the city that have traffic signals. The data was typically collected between the hours of 7:30 a.m. and 6:00 p.m. The data was obtained from the Open Data Portal and can be found here.

2.1.2 Cleaning

There are certain intersections that have zero pedestrian volume which are likely to be highway ramps or other intersections were pedestrians have no access. These entries were removed since I am only interested in areas of high pedestrian volume. Furthermore, one of the intersections with a high pedestrian volume had a corrupt value for its geographical location. Figure 1a shows a scatter plot of the latitude and longitude values of each intersection. This scatter plot should resemble the shape of the city, but you can see the corrupt point on the right of the plot. I fixed this issue by finding the correct location and updating the entry accordingly. Figure 1b shows the scatter plot after correcting the data point.



(a) Corrupt Intersection Location (b) Corrected Intersection Location

Figure 1: Cleaning the Pedestrian Volume Data

2.1.3 Feature Selection

The vehicle volume was dropped since most people use their vehicles to commute and not to visit venues. That is, someone driving in the city is not likely to stop at a venue while they are on their way to somewhere else, especially in the busiest areas of the city. The opposite is more likely for people walking since they might see a store or a cafe that they are interested in, and might stop to visit that venue. The vehicle volume would skew the process of selecting the busiest neighbourhoods in the context of what is the likelihood of someone visiting a business.

2.2 Neighbourhood Boundaries

2.2.1 Source

The boundaries of neighbourhoods in the city were used in conjunction with the pedestrian volume counts in order to determine the busiest area of the city. I was interested in finding the busiest area of the city since that is the area with the most venues and highest pedestrian volume. The data can be found here and figure 2 displays the boundaries of the neighbourhoods over a map of the city.

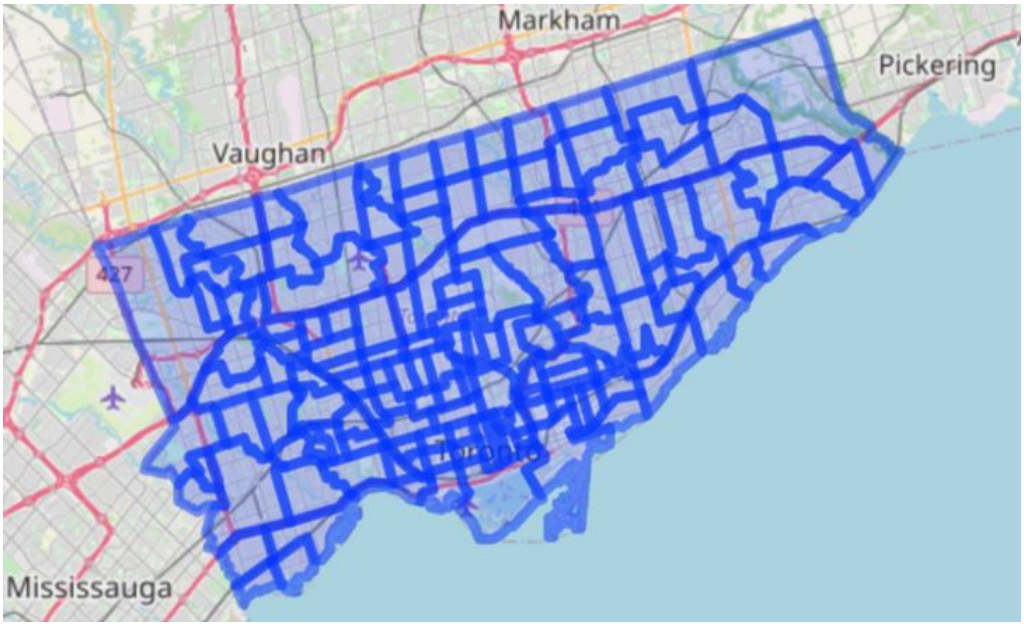


Figure 2: Neighbourhood Boundaries

2.3 BIA Boundaries

2.3.1 Source

The BIA boundaries data set was used to extend the area given by the busiest neighbourhoods to include the BIAs that fall within, or cross the boundaries, of that area. Furthermore, it was used to assign venues to their corresponding BIAs which allowed me to calculate aggregate metrics on the venue data of each BIA. The data can be found here. Figure 3 displays the BIA boundaries over a map of the city.

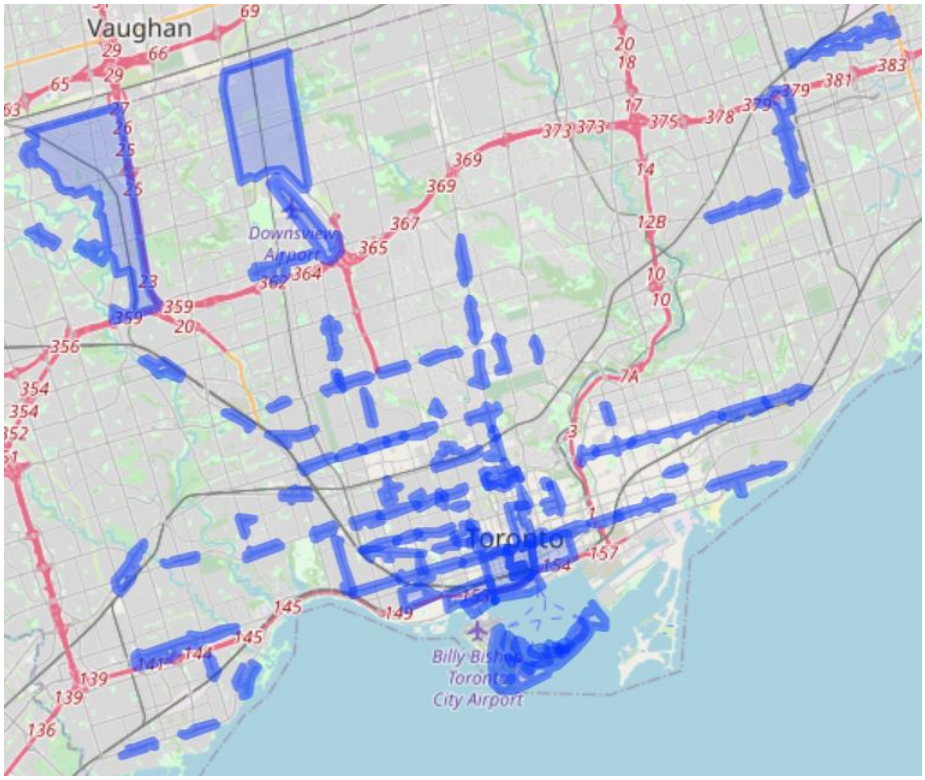


Figure 3: BIA Boundaries

2.4 Foursquare Location Data

2.4.1 Source

The Foursquare Places API allows real-time access to a global database of venue data. The API was used to retrieve venue data across the city. More information about the API and its uses can be found on their docs page.

2.4.2 Cleaning

Through the process of acquiring this data, there exists the possibility of retrieving duplicate venues. Therefore, I had to remove any duplicates that resulted from calling the API on areas that overlapped geographically. Furthermore, there are areas where the API returns no venues (e.g. residential neighbourhoods, parks, the lake, etc.) which creates an entry with missing data and thus requires removing. Finally, the API sometimes returns error messages due to maintenance of their servers so these entries had to be removed.

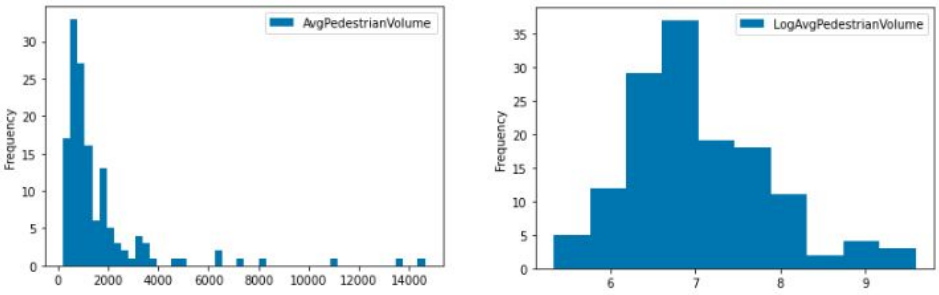
2.4.3 Feature Selection

The venue data contains not only the name and location of the venue, but the category as well (e.g. coffee shop or clothing store). However, the list of categories is very granular which resulted in over 583 unique categories. Venue categories have a parent category, children categories, or both. Therefore, to solve the problem of high category granularity I created an algorithm to retrieve the parent category of each venue and was left with less than 150 unique categories. Additionally, I retrieved the uppermost parent category of each venue to group them into six unique categories.

3 Methodology

3.1 Defining the Busiest Area in the City

The pedestrian volume data set was used in conjunction with the neighbourhood profile data set in order to determine the busiest neighbourhoods in the city. I designed an algorithm that takes a groups of points and polygons, and determines which points belong to which polygons. This allowed me to determine the neighbourhoods that each intersection belongs to. Once the intersections were assigned to their corresponding neighbourhoods, I was able to calculate the average pedestrian volume of each neighbourhood.



(a) Average Pedestrian Volume (b) Log Average Pedestrian Volume

Figure 4: Transforming the Average Pedestrian Volume Across Toronto Neighbourhoods

Figure 4a shows the distribution of the average pedestrian volume across all neighbourhoods. The distribution has a long tail, and most of the values are concentrated below 2,000. In order to better visualize the busiest neighbourhoods on a map (Figure 5), I decided to instead use the log of the average pedestrian volume.

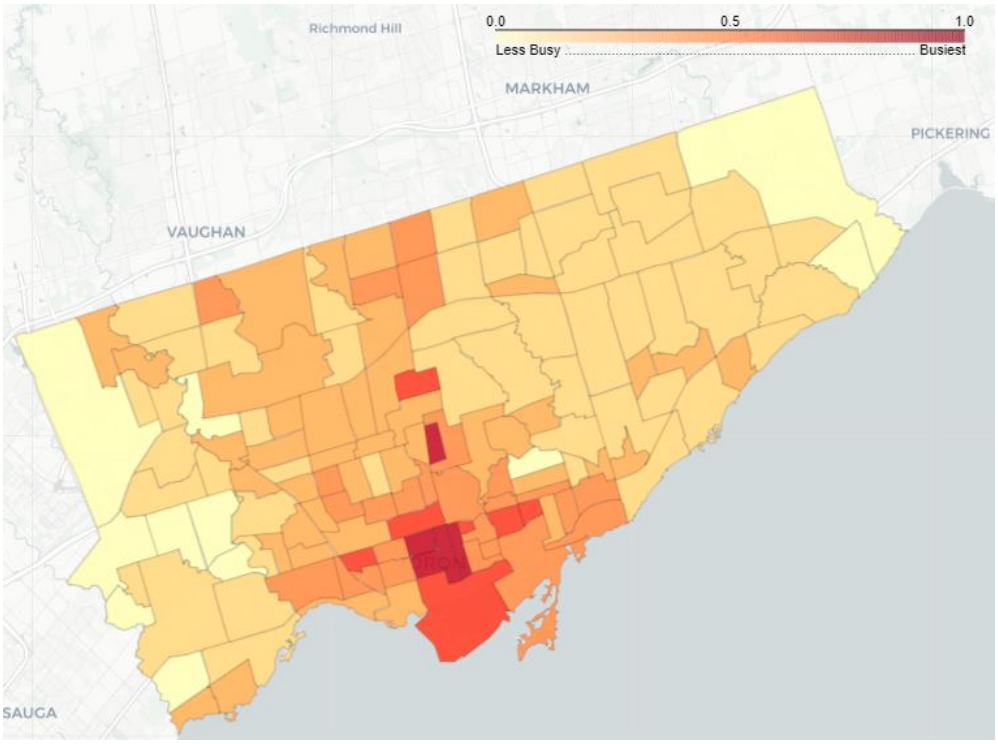


Figure 5: Busiest Toronto Neighbourhoods by Pedestrian Volume

You can see that the busiest neighbourhoods are located near the south of the city (Figure 5). Given that this is the downtown area of the city where the financial district is located, it makes sense that these neighbourhoods would get the highest pedestrian volume. Using this information, I chose the ten busiest neighbourhoods and defined a rectangular area given by the boundaries of the outermost neighbourhoods (Figure 6). The next step was to look at the BIAs that fall within or cross this rectangular area, and adjust it to fully include those BIAs.

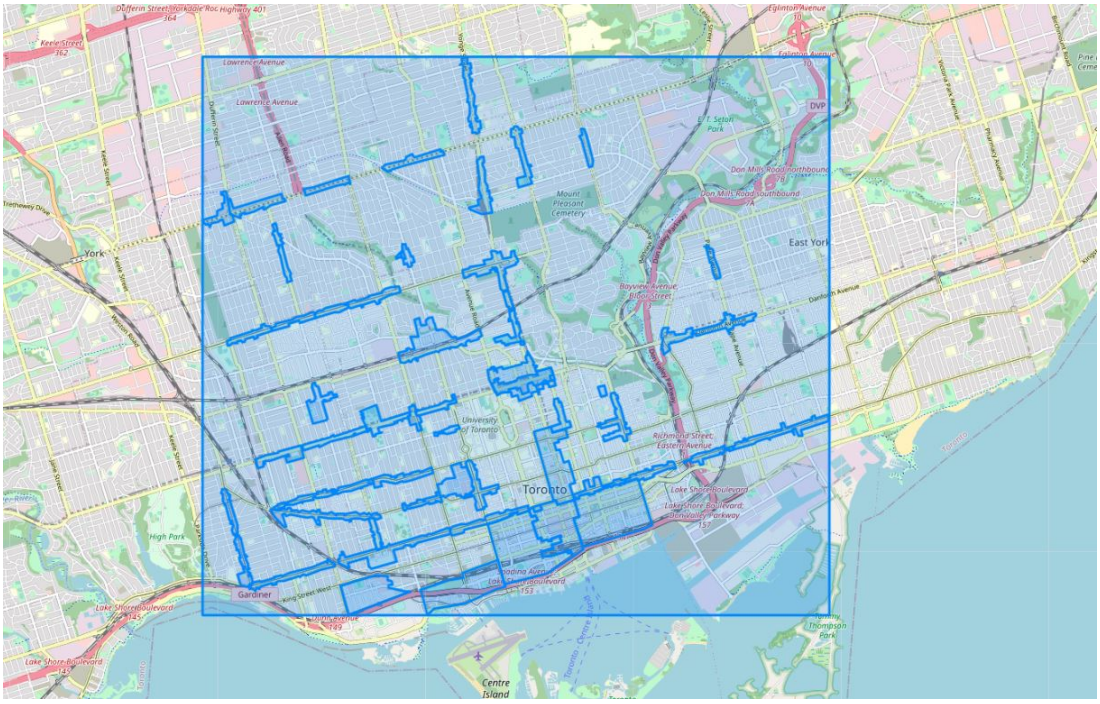


Figure 6: Area of Interest with BIAs

3.2 Using *k*-Center Clustering to Retrieve Venue Data

The k-Center Clustering algorithm covers the points of a polygon with k balls of maximum radius. It finds k points in a polygon, such that the maximum distance of a point in the polygon to the closest point in a cluster is minimized. I implemented this algorithm from section 4.2 of the paper Geometric Approximation Algorithms by Sariel Har-Peled of the University of Illinois.

In order to find such k points, the algorithm has to iterate through points inside the polygon. In this case, the polygon is the rectangular area I defined. However, this area contains no points so I had to solve this issue by filling it with a set of uniformly distributed points (Figure 7).

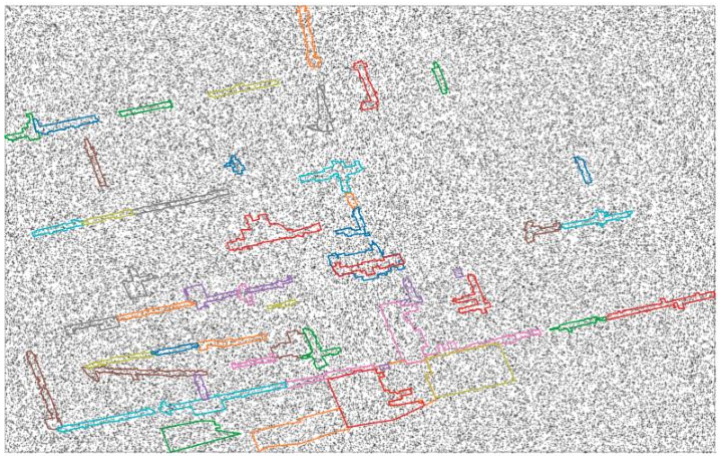


Figure 7: Area of Interest with BIAs and Uniformly Distributed Points

The k points define the center of circles of a fixed radius passed to the algorithm as a parameter. I decided that each circle will have an area determined by a radius of 110 metres. This is enough to capture most if not all of the venues inside the given area. Additionally, the API only allows up to 5,000 calls per hour and the chosen radius accounts for this limitation. Lastly, venues that aren't captured by one API call might get captured by the API call of an adjacent, overlapping circle. Figure 8 shows the *Trinity-Bellwoods* neighbourhood with its corresponding circles. There are clear gaps between the circles but this is a consequence of the limit of API calls I was able to make.

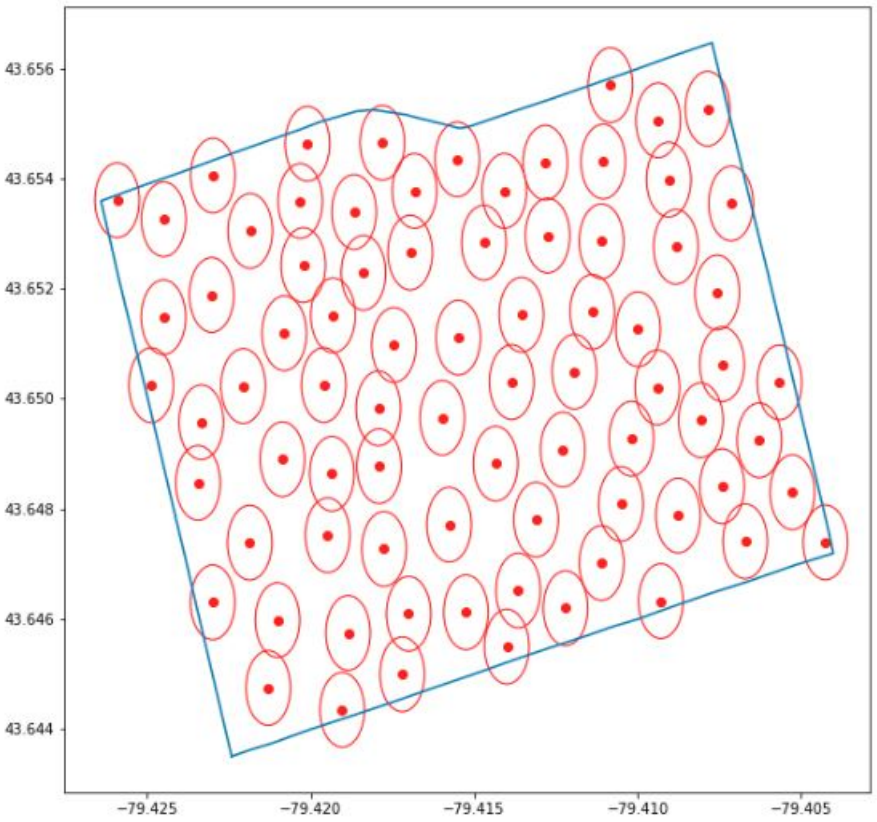


Figure 8: Visualization of the k-Center Clustering Algorithm

3.3 Foursquare Venue Data

With the circles defined within the rectangular area, I made an API call for each circle and retrieved the venues that fell within those localized areas. Each API call takes just under half a second but with almost 5000 API calls total, this would have taken much longer than necessary. In order to get past this issue, I parallelized the API calls and was able to make all the calls in less than a minute.

After cleaning up and processing the data, I had over 32,000 venues within the rectangular area. The next step was to assign each venue to its corresponding BIA (if it is inside a BIA) in order to compare the venue distributions inside and outside BIAs. Figure 9 shows the distributions of venues by category, inside and outside BIAs. From the figure it is clear that there is a difference in their distributions.

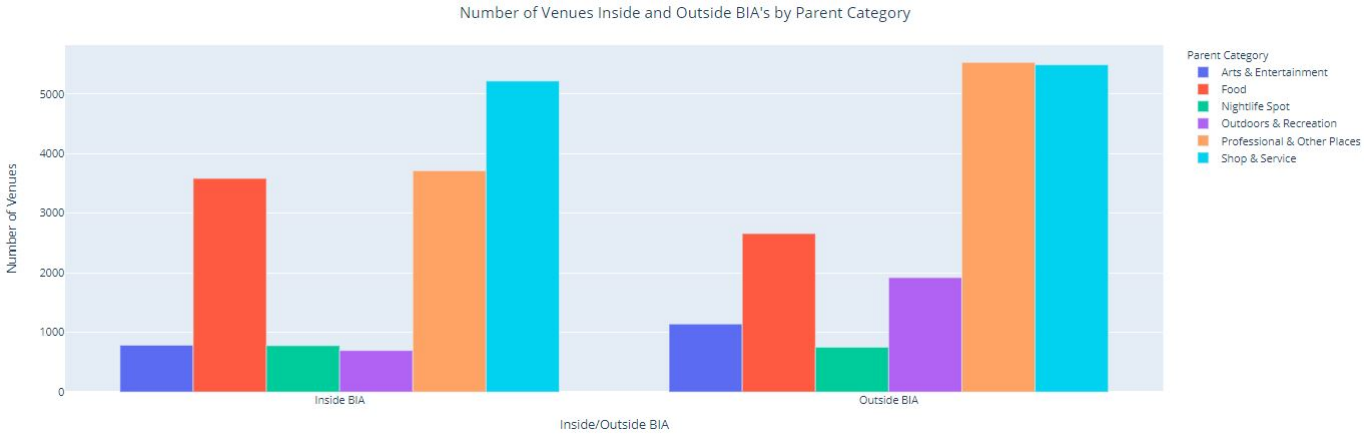


Figure 9: Distribution of Venues by Category Inside and Outside BIAs

The figure above shows some interesting differences in the number of venues inside and outside BIAs. First of all, it is worth noting that there are 15.55% more venues outside of BIAs than inside. There is a similar number of 'Shop & Service' venues inside and outside BIAs; but, since there is a higher number of venues outside the BIAs this is an indication that when designating BIAs business owners are interested in including these types of venues. This also applies to venues in the 'Nightlife Spot' category. The venues in the 'Food' category are also of high interest as shown by the fact that there are actually more food venues inside BIAs than outside BIAs. It is interesting to see that even though BIAs cover a lower percentage of the area of interest, there are more food venues inside them than outside of them. The rest of the venues don't differ much, except for those in the 'Outdoors & Recreation' category. However, given that BIAs cover a lower percentage of land in addition to most green spaces being outside BIAs, it is expected to see this difference in the number of venues of that category.

3.4 Clustering Venues Outside BIAs Using DBSCAN Clustering

To detect areas with potential of being in a BIA, I used the DBSCAN clustering algorithm to cluster venues outside BIAs. This would allow me to compare such clusters to the venues inside existing BIAs.

The DBSCAN algorithm has a hyper-parameter called *eps* of which its optimal value was found using a k Nearest Neighbour algorithm. After performing the DBSCAN on the set of venues, I filtered out the clusters that didn't have a satisfactory number of venues in them (Figure 10).

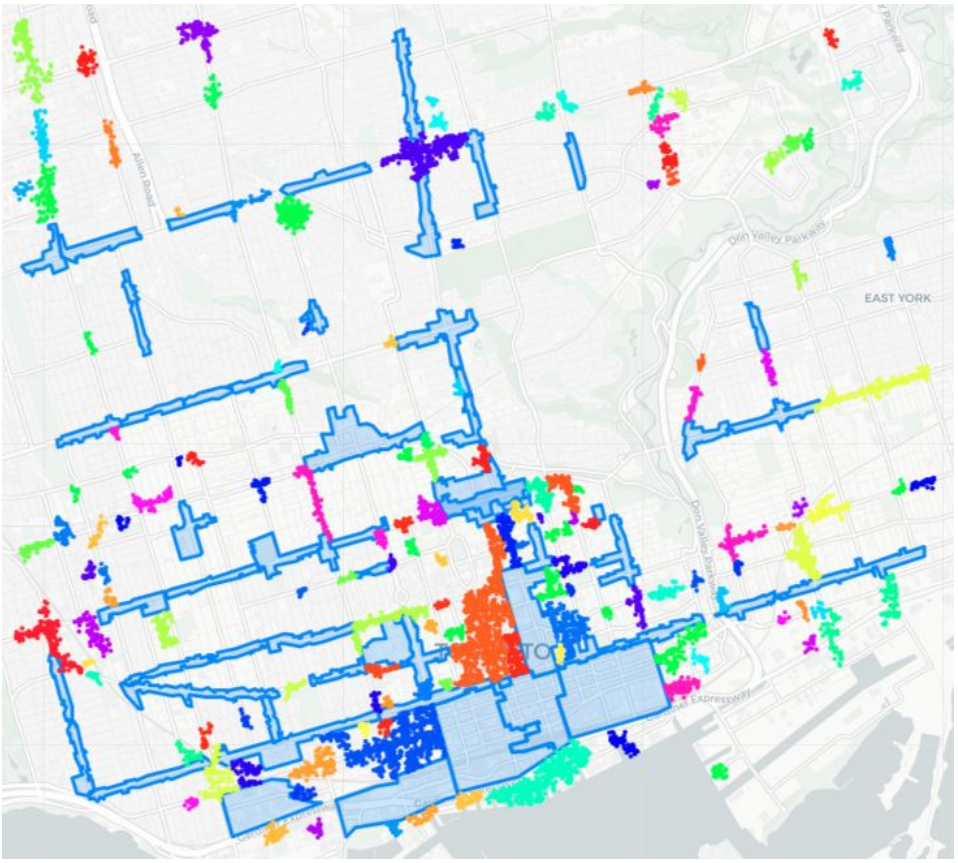


Figure 10: Clusters of Venues Outside BIAs

These are the clusters that will be passed through the trained Autoencoder to determine which ones are most similar to existing BIAs. In the next section, I show the results of using the Autoencoder to achieve this goal.

3.5 Autoencoder to Detect Potential BIAs

The set of existing BIAs was used to train an Autoencoder using PyTorch. The Autoencoder has two hidden layers: one for encoding and one for decoding the data. The activation function used in each node is the ReLu function. The total number of features used as an input was eight: the log average pedestrian volume, the percentage of venues by category in the categories shown in figure 9, and the total number of venues in the BIA. The encoder layer compressed the features to six new, encoded features. A MinMax scaler was used to normalize the data, the Adam optimization algorithm was used as the optimizer, and the Mean Standard Loss was used to calculate the reconstruction loss during training.

The algorithm was trained for 120 epochs until the training loss converged (Figure 11).

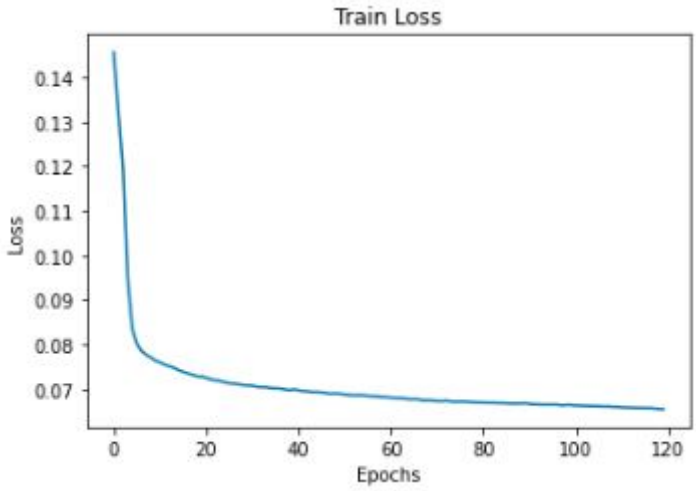


Figure 11: Epochs vs Reconstruction Loss

After training the Autoencoder, the set of clusters found using DBSCAN can be passed through the network and their reconstruction losses calculated. In the next section I show the reconstruction loss distribution for all the clusters, as well as a map containing the ten clusters with the lowest reconstruction loss.

4 Results

A low reconstruction loss means that the cluster is similar to the set of existing BIAs, thus indicating that the set of venues could exist within a BIA. Figure 12 shows the distribution of the reconstruction loss across all clusters. As you can see, multiple clusters have a low reconstruction loss while some of them have a higher loss indicating that they may not be suitable to be in a BIA. Figure 13 shows the map of city with the existing BIAs as well as the ten clusters with the lowest reconstruction loss.

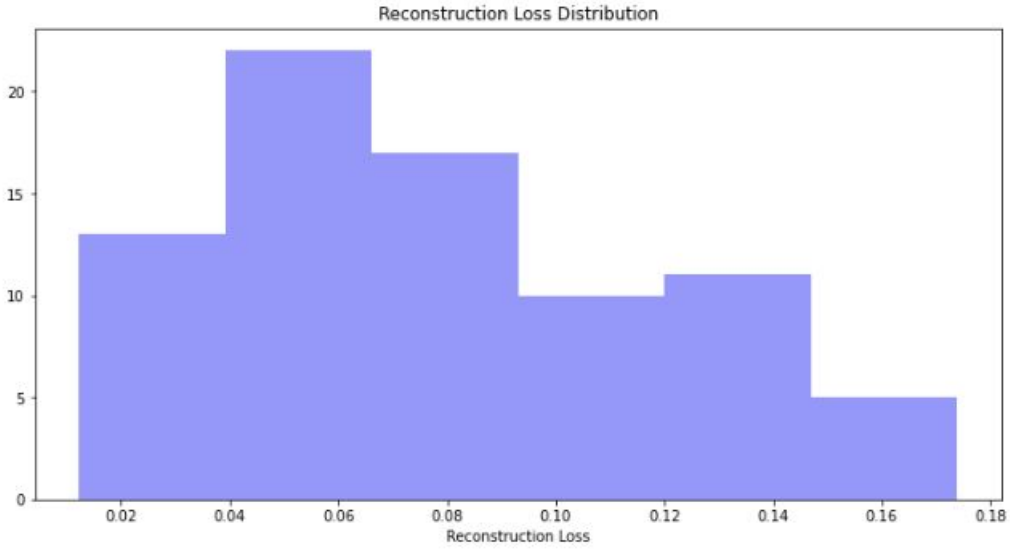


Figure 12: Reconstruction Loss Distribution

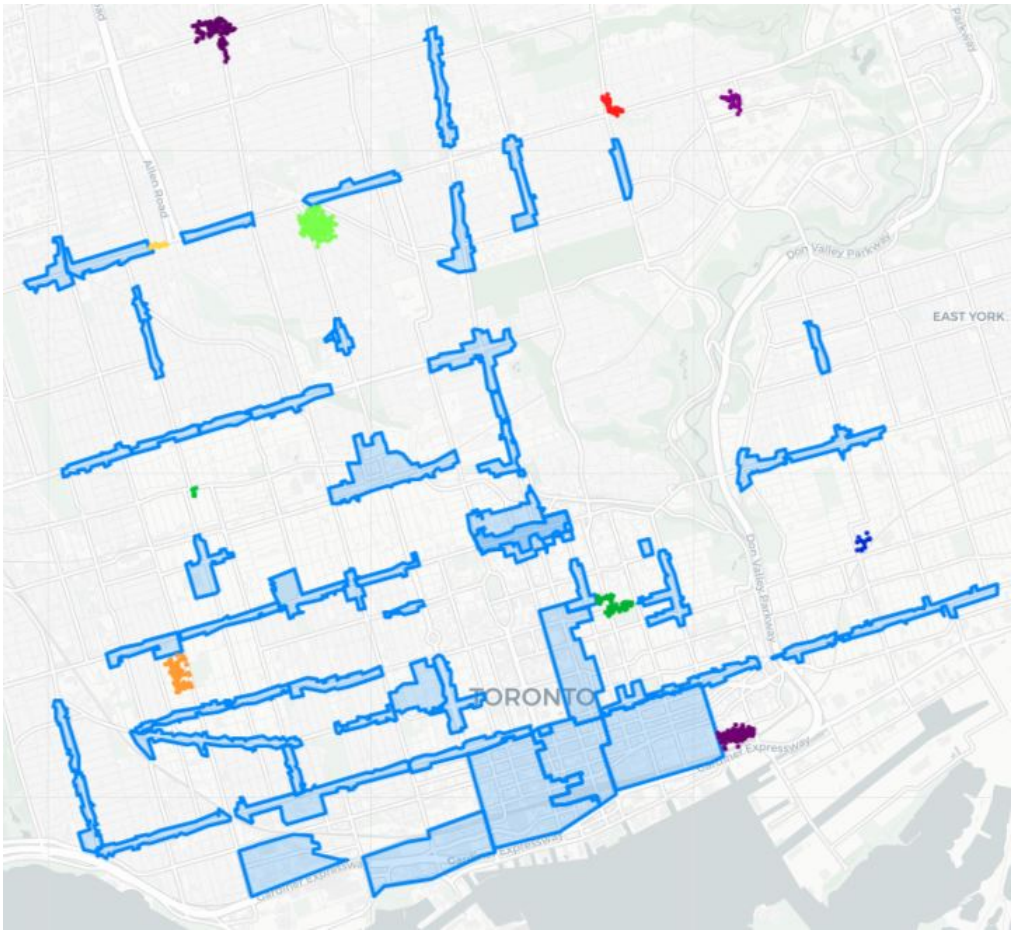


Figure 13: Reconstruction Loss Distribution

From the map, you can see that some of the clusters are not surrounded by existing BIAs while some are adjacent to them. The ones that are adjacent to existing BIAs could potentially become additions to them while the standalone ones could be part of their own BIA.

5 Discussion

In this project, I was able to explore the types of venues within the BIAs of Toronto. It was interesting to see the distribution of venues inside BIAs compared to the distribution of venues outside BIAs. A particular piece of information that I found surprising was the fact that even though BIAs cover less area, they contain almost the same number of venues dedicated to shopping and nightlife and more food venues than areas outside BIAs.

Trying to detect venues with the potential of being in a BIA was both ambitious and challenging, and at times it really felt like that! I went through many ideas, failed multiple times, but finally found something that I thought was worthwhile exploring. After exploring different methods such as Isolation Forests and One-Class SVMs, I found the Autoencoder to be the best match to my problem. However, using the Autoencoder to achieve my goal came with some limitations.

The first limitation I encountered was the fact that I didn't have too many samples to train the algorithm on which can lead to over-fitting the training data. This limitation came from the fact that I wasn't able to include all the BIAs in the city given the cap on the number of calls I could

make to the API; with more sparse areas to make API calls, I wouldn't have been confident that I captured most of the venues in any given area. The second limitation was the absence of negative data (that is, data that I could confidently flag as not a BIA) which transformed my problem into a one-class classification problem. One-class classification isn't as well developed as other classification methods with still a lot of research going on in this field, and I spent multiple hours reading through papers trying to figure out what the best course of action was. With the inclusion of negative data I could turn this into a binary classification problem and use more traditional and accessible methods.

6 Conclusion

The Business Improvement Area program in the City of Toronto has proven to be successful in improving neighbourhoods in the city and helping new and existing businesses be more successful. In this study I analyzed the BIAs in the busiest area of the city and used geographical and venue data to determine if other neighbourhoods in the city could host new BIAs. I used multiple clustering algorithms to help me retrieve data using the Foursquare Places API and to cluster such data into groups of venues. Finally, I implemented an Autoencoder to see whether groups of venues could be included within a BIA and detected the ten areas with the highest potential of being a part of a new or existing BIA. I hope that this solution can help business owners in their decisions of extending existing BIAs or creating new ones to further improve the city and its neighbourhoods.

7 Future Improvements

There is still a lot of work to do in order to improve the performance of the Autoencoder. For example, it would be useful to extend the solution to include all the venues and BIAs in the city. Furthermore, creating negative data (i.e. areas in the city that are guaranteed to not be a BIA) from residential areas, suburbs, parks, etc. would be useful to increase the number of training samples. Creating negative data would also allow me to transform the problem from a one-class classification problem to a binary classification problem. Then I could use more common and better developed model performance metrics. I hope to come back and improve this solution.