2. Data Acquisition and Cleaning

2.1. Data sources

The data used in this project was obtained from the below sources-

- List of postal codes, boroughs and neighborhoods in Toronto, Canada obtained by scraping **Wikipedia** at the below URL: https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M
- Latitude and Longitude of these neighbourhoods from **Geospacial_Coordinates.csv**: http://cocl.us/Geospatial_data/Geospatial_Coordinates.csv
- Venues data of these neighborhoods using the **Foursquare API**. Foursquare API provides access to an enormous database consisting of venues from all around the world including their categories, addresses, tips, photos and comments. List of end-points: https://developer.foursquare.com/docs/places-api/endpoints/.

2.2. Data cleaning

Postal code table data was scraped from Wikipedia to create a dataframe consisting of Postal Code, Borough and Neighborhood.

However, some data cleansing was required as it had a few values under the column 'Borough' that were 'Not Assigned'. To correct this, I dropped any rows with a 'Not Assigned' value in the 'Borough' column. Another problem was that there were a few rows that had a borough but a 'Not Assigned' value in the "Neighborhood" column. For these cases, I set 'Neighborhood' to be the same as 'Borough'. To ensure there are no duplicate rows, I combined rows with the same 'PostalCode' by concatenating neighborhoods by a comma. There were also new line characters in the Postal Code column that had to be removed.

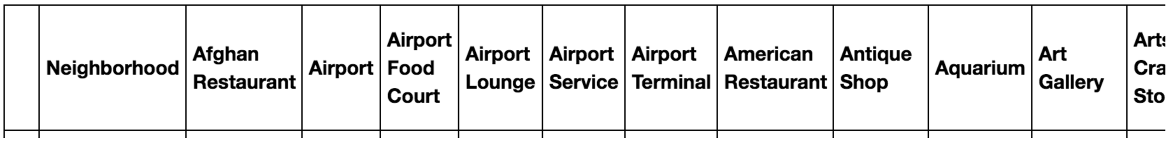
After fixing these problems, I verified the data to ensure that all the cells had correct values and formats.

2.3. Feature selection

After data cleansing, I combined the neighborhoods dataframe (PostalCode, Borough, Neighborhood) with the geospatial dataframe to add Latitude and Longitude columns. Since the primary focus was Toronto city, I filtered the data to work with only boroughs containing the word 'Toronto'.

The next step was to get venues information for all the neighborhoods using the Foursquare API. After signing up for a Foursquare developer account, using the Client ID and Client Secret, I made API requests to retrieve venue information. From the output received, I selected Venue name, Venue Latitude, Venue Longitude and Venue Category.

I then performed a One-Hot encoding on ‘Venue Category’ and grouped the rows by ‘Neighborhood’. The result was a dataframe that had 235 features (Neighborhood, Venue Categories as columns)



	Neighborhood	Afghan Restaurant	Airport	Airport Food Court	Airport Lounge	Airport Service	Airport Terminal	American Restaurant	Antique Shop	Aquarium	Art Gallery	Arts Cra Sto