COURSERA BATTLE OF THE NEIGHBOURHOODS/BOROUGHS

INTRODUCTION

PROBLEM DEFINITION

Whenever someone is trying to move to a different state/neighborhood they always have to decide on where to move to depending on what preferences they have and which places fulfill those preferences. The problem however is knowing whether your preferences will most likely be fulfilled in the neighborhood. And given that you might have never been there it becomes a bit tricky to decide on where to move to.

For example, person A prefers a neighborhood with a gym and a restaurant and person B prefers one with a book store and a coffee shop. Now, assume A and B have never been outside their state for example, or are just randomly searching for places that have the highest chance of fulfilling their preferences. Going through all neighborhoods/states/districts/boroughs might be hard and cumbersome.

Similar to people trying to set up businesses, for example, depending on whether they are bringing in something that wasn't already there or are going with the trend and are setting up the business they see more people are venturing into or are already setup. They have to go through a lot of data if they choose to search place by place.

OVERVIEW

What this project intends to achieve is create a way, for those who want to start businesses/shops and those looking for places to move to, to find places that most fit their preferences or target by collecting all available data on places and returning those that will most likely fit their preferences.

From the result they can now narrow down and search/thoroughly go through the data of places that will most likely have shops/centers for those travelling/moving and business for those looking to start one. It can also help those who want to start business/centers find places that have more complementary business, e.g., a snack shop can move to a place with many coffee shops.

DATA

We will use the foursquare api to get data on various place. To begin with we will compare New York and Toronto, their boroughs and the neighborhood in one of their boroughs.

Using foursquare api we will get data on centers in New York and Toronto and what category they belong to e.g., a coffee shop or an Italian restaurant.

We will first get the geocodes of both Toronto and New York and use them to set boundaries for our search area, then return everything within that search area. The catch is we set the limit of returned results to 100 just for simplicity but you can set the limit to larger numbers to get as much data as possible.

You can also use geocodes of places you prefer, like other states/cities, and compare the results.

We will aggregate this data and get the count of all categories available in the area, e.g., how many coffee shops or movie shops are in the area (basically get the count). Using this we get the probability of finding that category in that specified area.

These areas can either be a comparison of boroughs in a town/state or of neighborhoods in a borough. The result will be a chat of the areas with the most likelihood of having what you prefer.

Needed libraries

```
import pandas as pd
import numpy as np
from bs4 import BeautifulSoup
import urllib.request,urllib.parse , requests
import re
import geocoder
import folium
import json
import matplotlib
import matplotlib.pyplot as plt
import matplotlib.cm as cm
from geopy.geocoders import Nominatim
```

METHODOLOGY

1. We scrap data on the cities/states from the relevant sites and get their geocodes of neighborhoods. This can be from any site in our case we used Toronto borough and neighborhood data from Wikipedia, Toronto geodata csv and New York data json

Toronto Dataframe

```
toronto_neighborhoods.head()
```

	PostalCode	Borough	Neighborhood	Latitude	Longitude
0	M1B	Scarborough	Malvern,Rouge	43.806686	-79.194353
1	M1C	Scarborough	Rouge Hill,Port Union,Highland Creek	43.784535	-79.160497
2	M1E	Scarborough	Guildwood,Morningside,West Hill	43.763573	-79.188711
3	M1G	Scarborough	Woburn	43.770992	-79.216917
4	M1H	Scarborough	Cedarbrae	43.773136	-79.239476

New York Dataframe

```
newyork_neighborhoods.head()
```

	Borough	Neighborhood	Latitude	Longitude
0	Bronx	Wakefield	40.894705	-73.847201
1	Bronx	Co-op City	40.874294	-73.829939
2	Bronx	Eastchester	40.887556	-73.827806
3	Bronx	Fieldston	40.895437	-73.905643
4	Bronx	Riverdale	40.890834	-73.912585

2. Get the latitude and longitudes of Toronto and New York using the following function, and view the neighborhoods.

```
def view_neighborhoods(user_agent,town_name,df):
    geolocator = Nominatim(user_agent=user_agent)
    location = geolocator.geocode(town_name)
    latitude_ ,longitude_ = location.latitude, location.longitude
```

3. Now we set up our foursquare credentials and use them to get data for both Toronto and New York. The results look as shown below

Toronto

```
toronto_venue_data = get_venue_data(toronto_latitude,toronto_longitude,toronto_neighborhoods)
toronto_venue_data.head()
```

	Borough	Neighborhood	name	category	latitude	longitude
0	Scarborough	Malvern,Rouge	Downtown Toronto	Neighborhood	43.653232	-79.385296
1	Scarborough	Malvern,Rouge	Nathan Phillips Square	Plaza	43.652270	-79.383516
2	Scarborough	Malvern,Rouge	Poke Guys	Poke Place	43.654895	-79.385052
3	Scarborough	Malvern,Rouge	Textile Museum of Canada	Art Museum	43.654396	-79.386500
4	Scarborough	Malvern,Rouge	Old City Hall	Monument / Landmark	43.652009	-79.381744

New York

```
newyork_venue_data = get_venue_data(newyork_latitude, newyork_longitude, newyork_neighborhoods)
newyork_venue_data.head()
```

	Borough	Neighborhood	name	category	latitude	longitude
0	Bronx	Wakefield	The Bar Room at Temple Court	Hotel Bar	40.711448	-74.006802
1	Bronx	Wakefield	The Beekman, A Thompson Hotel	Hotel	40.711173	-74.006702
2	Bronx	Wakefield	Alba Dry Cleaner & Tailor	Laundry Service	40.711434	-74.006272
3	Bronx	Wakefield	City Hall Park	Park	40.711893	-74.007792
4	Bronx	Wakefield	Gibney Dance Center Downtown	Dance Studio	40.713923	-74.005661

4. Now we plot the all the top categories from both states/towns individually and combined to see how each is performing with regards to presence(count).

5. Now that we have a table of the counts of all available categories from the scrap/data mine, we can get the probabilities
   (Let us use coffee shops and Movie shops as examples of categories, but the same idea can be applied across the board)
   - Probability of a coffee shop occurring in a borough in Toronto is given:

$$\frac{count\ of\ coffee\ shops\ in\ borough(x)}{count\ of\ coffee\ shops\ in\ Tororno}$$

   - We group by category and divide all categories(columns) by their sum, this will return a dataframe of probabilities of that category being in that neighborhood/borough
   - **Case 1: Only one preference**
     - Here we just return a graph of the column/preference since one preference could be something like, likelihood of having a coffee shop in the area is just the column with coffee shop probabilities in the dataframe with probabilities.
   - **Case 2: More than one preference**
     - We consider the chance of joint, disjoint and conditional probability

   Say: A = Coffee shop, B= Movie shop , C = Italian Restaurant

   A) <u>Joint Probability</u>

$$p(A\ and\ B\ and\ C) = p(A) * p(B) * p(C)$$

   In this case you are looking for a place with all shops/centers you want with no preference for any of the shops/center, but you want all of them to be there.

B) **Disjoint Probability**

$$p(A\ or\ B\ or\ C\ ) = p(A) + p(B) + p(C)$$

In this case you are looking for a place with all shops/centers you want with no preference for any of the shops/center, and you are okay with at least one of them being there. E.g., A big American restaurant, an American restaurant and a New American, maybe you just want a restaurant regardless of which it is

C) Conditional Probability

Here the order of selection matters. The assumption is that, the order of selection is also the order of preference.

$$p(B|A) = \frac{p(A\ and\ B)}{p(A)}$$

Here we assume A came before B in the selection therefore there is more preference for A over B. Therefore implying what is being looked for is the probability that B is in the area given A is in the area. Then we set $p(B|A)$ as the probability of B.
If there are 3 selections then the probability of the third becomes the probability of the third given the first and second; $p(C|A\ and\ B)$, and so on:

Therefore:

$$p(X_n|X_0\ and\ X_1\ and\ \dots X_{n-1})$$

That is the general idea.

Eventually Bayes probability theorem is used for the conditional probability.

$$p(B|A) = \frac{p(A|B)p(B)}{p(A|B^-)p(B^-) + p(A|B)p(B)}$$

Here: $p(B^-) + p(B) = 1$
$p(B^-)$ is probability of the next choice not being in that location.

For every row(in this case location), we run a loop to determine what is the probability of finding that combination in that place. The idea behind this is we are trying to get the probability of the next preference being satisfied given the previous preference have been satisfied.
Let Joint probabilities of those before be = $PN = X_0\ and\ X_1\ and\ \dots X_{n-1}$
So, X_n it will be:

$$p(X_n|PN) = \frac{p(PN|B)p(B)}{p(PN|B^-)p(B^-) + p(PN|B)p(B)}$$

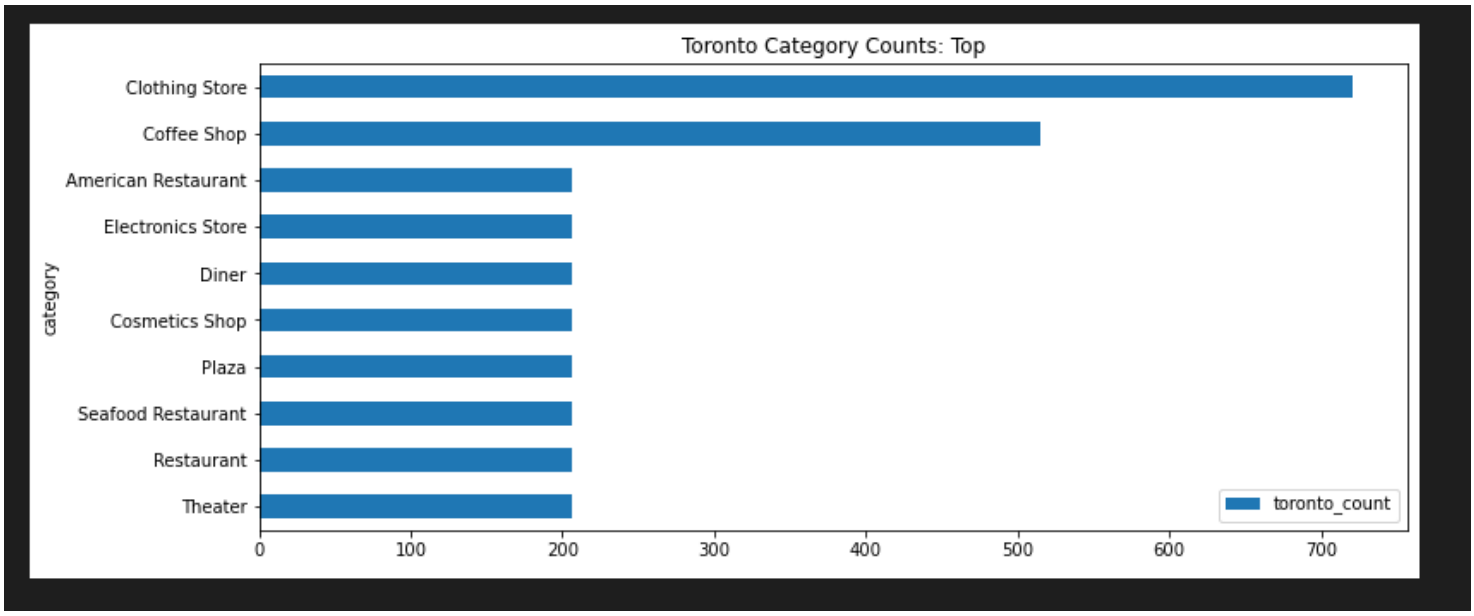
Where, X_n is the probability of choice **n** being in that location given choices **1** to **n-1** are already in that location.

The probability $p(X_n|PN)$ will be our desired probability for that combination since it caters for all other/previous conditions.

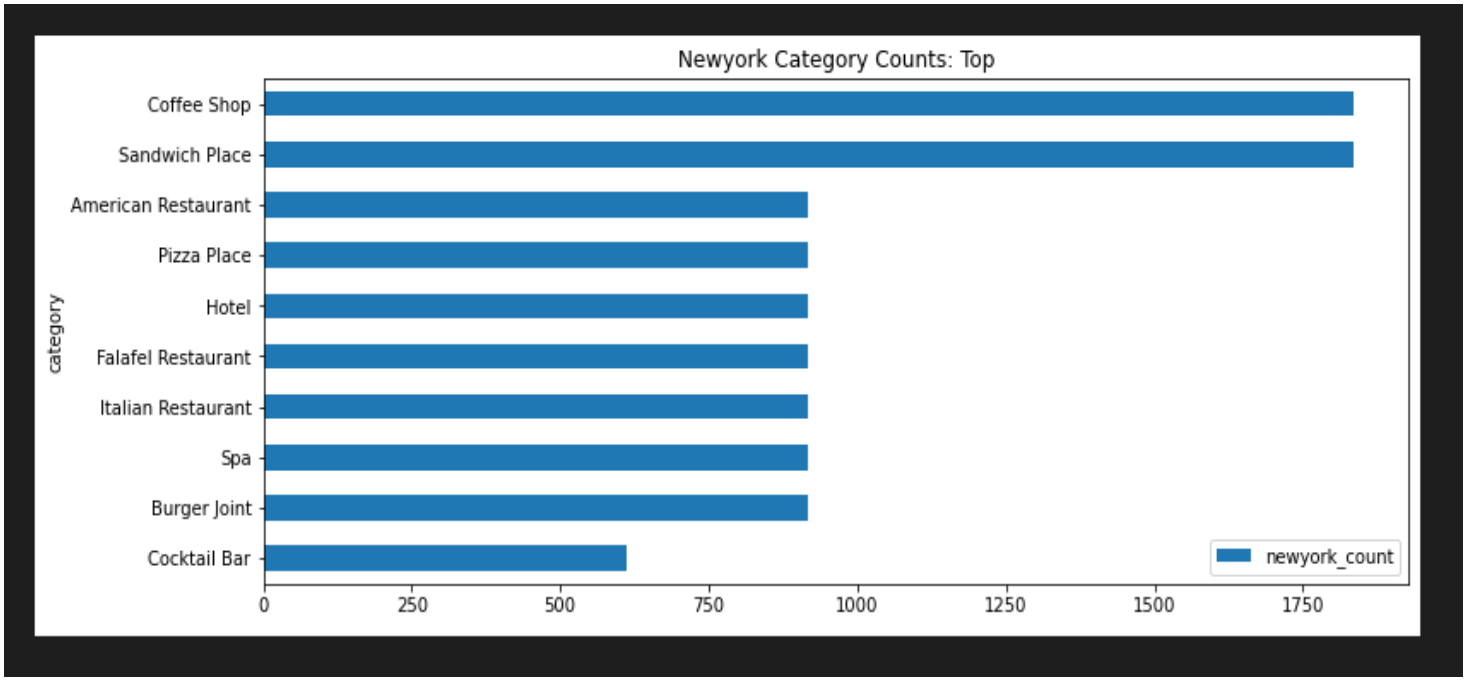
We repeat the same for all locations/Neighborhoods/Boroughs and get the results sort them descending order and plot it.

RESULTS

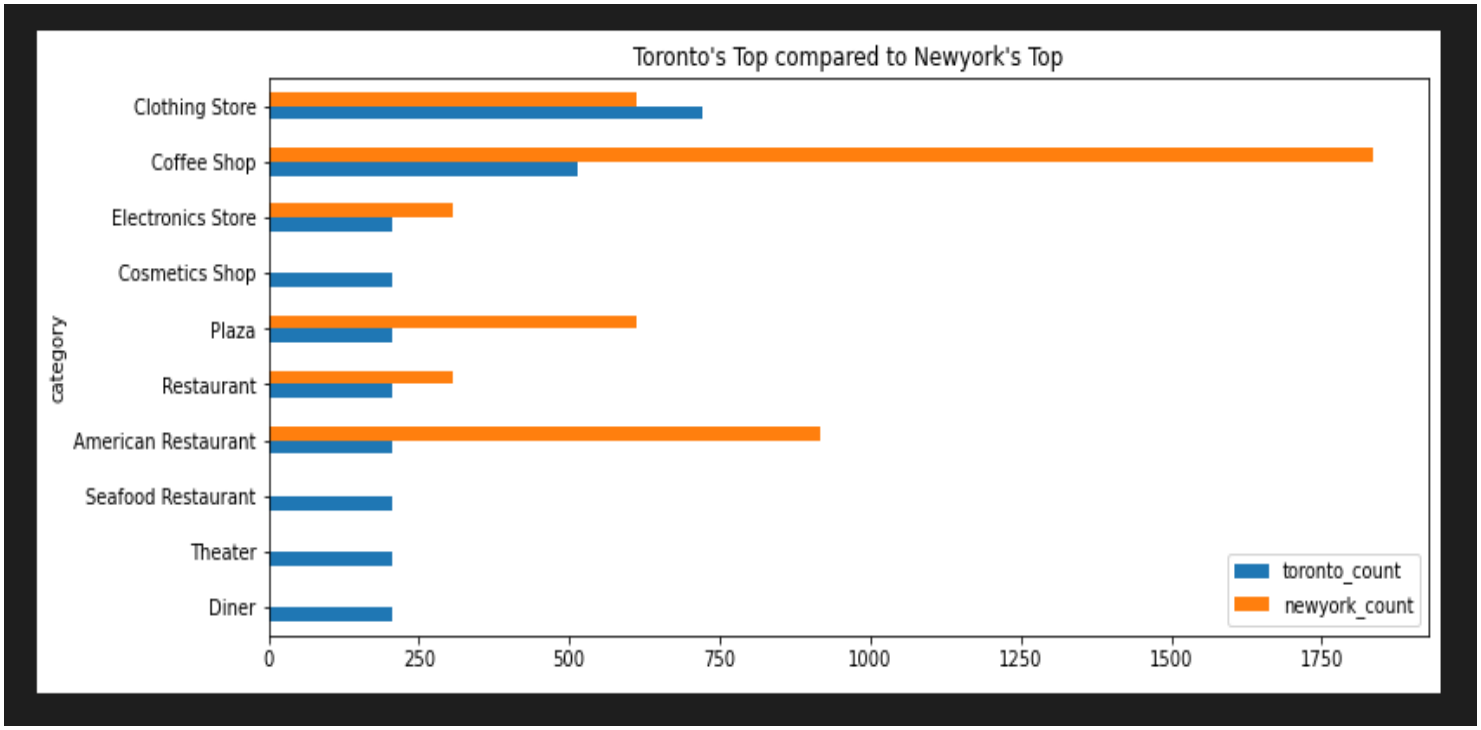
Top categories in Toronto



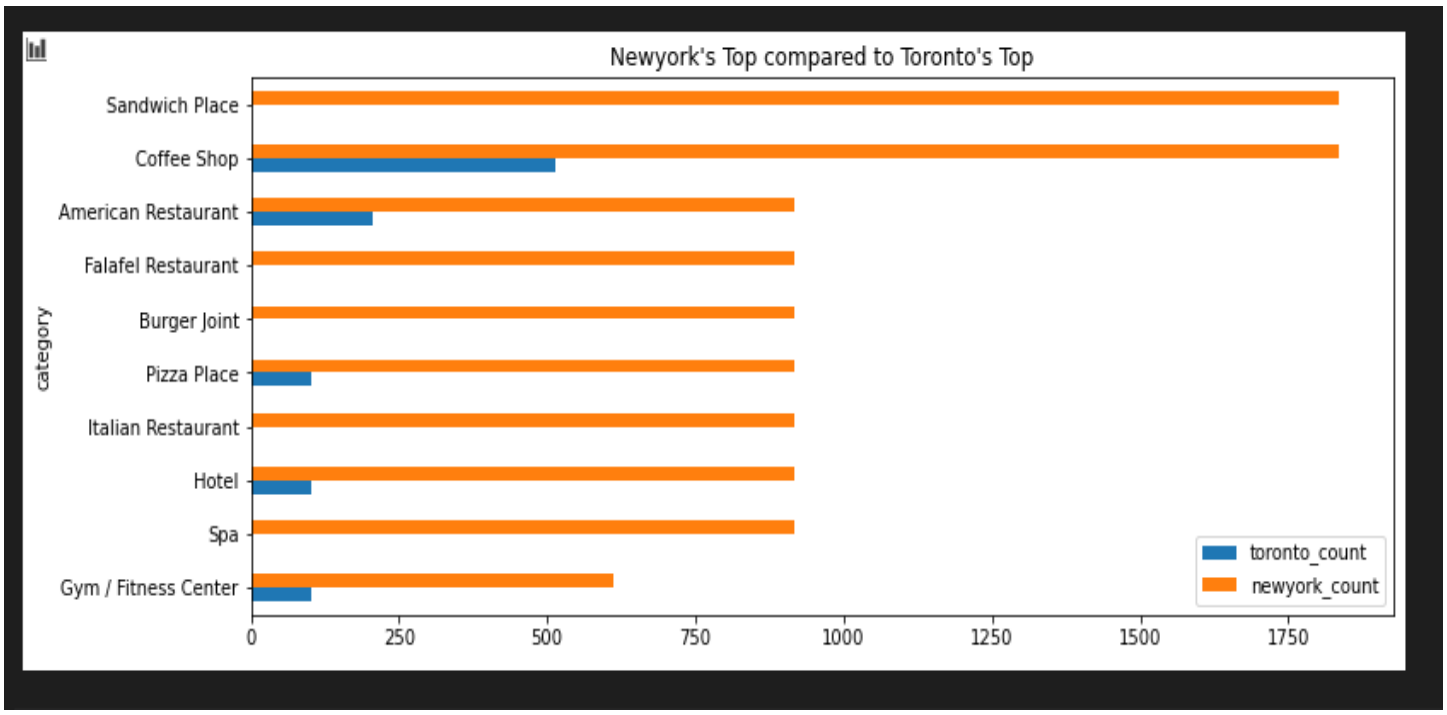
Top categories in New York



Top categories in Toronto vs Top categories in New York

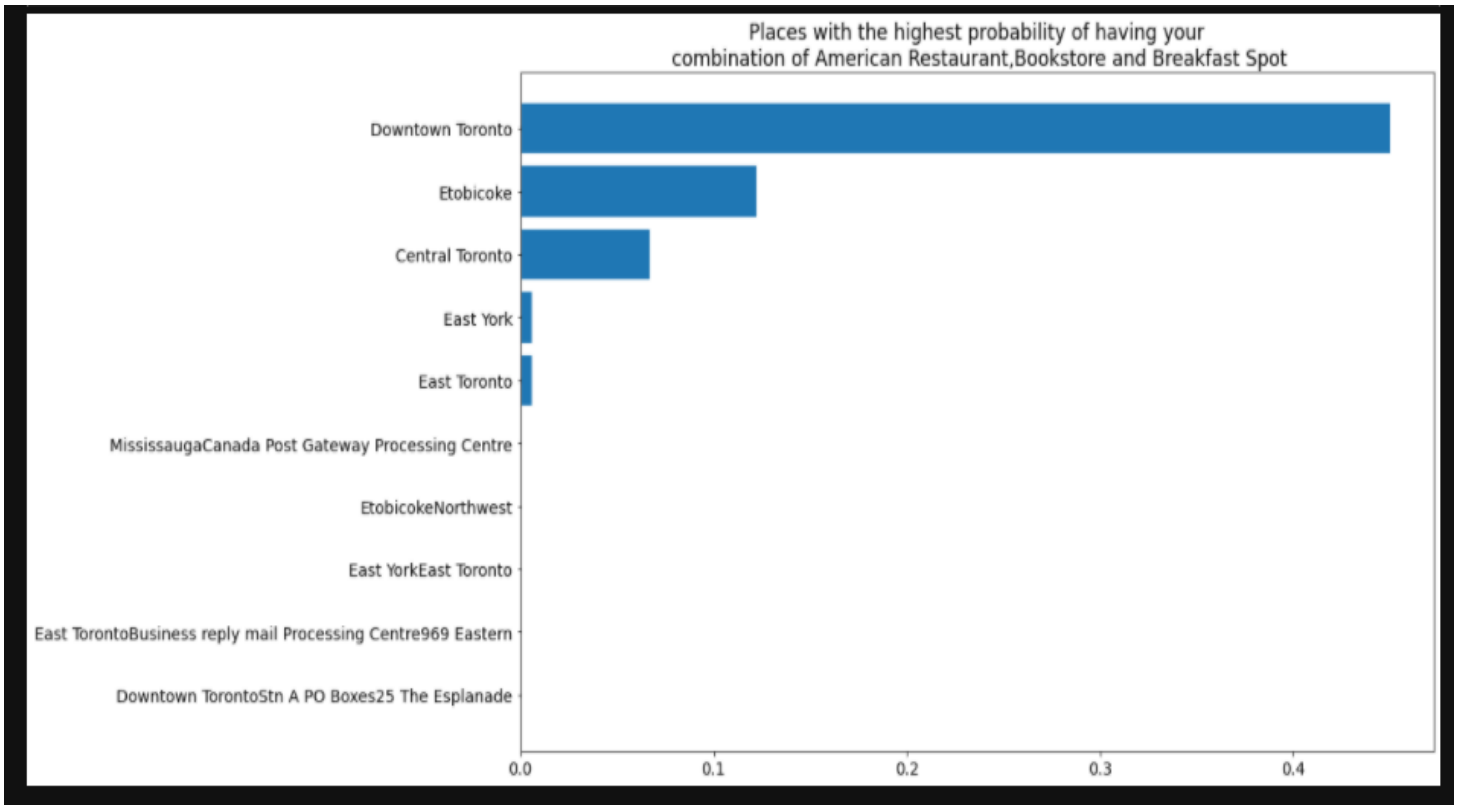


Top categories in New York vs Top categories in Toronto

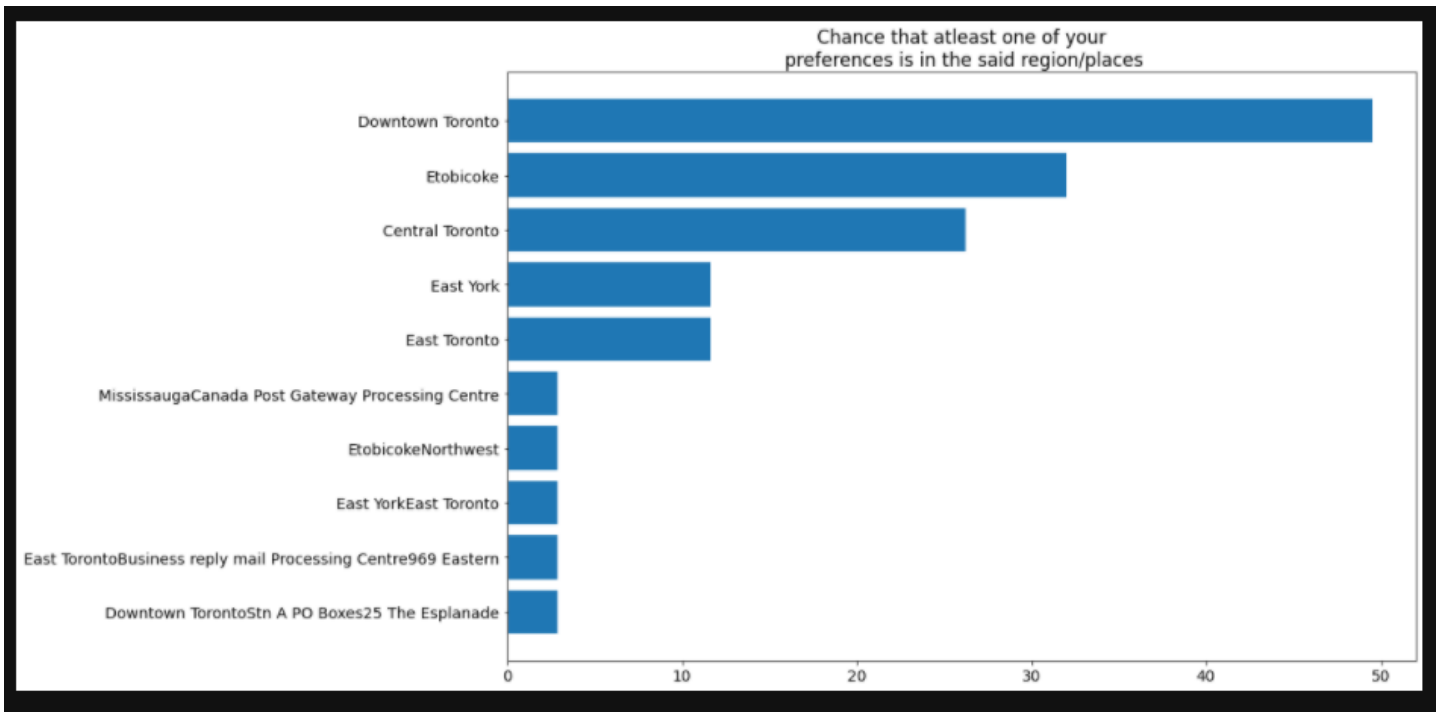


We will use Toronto as a test view for results:

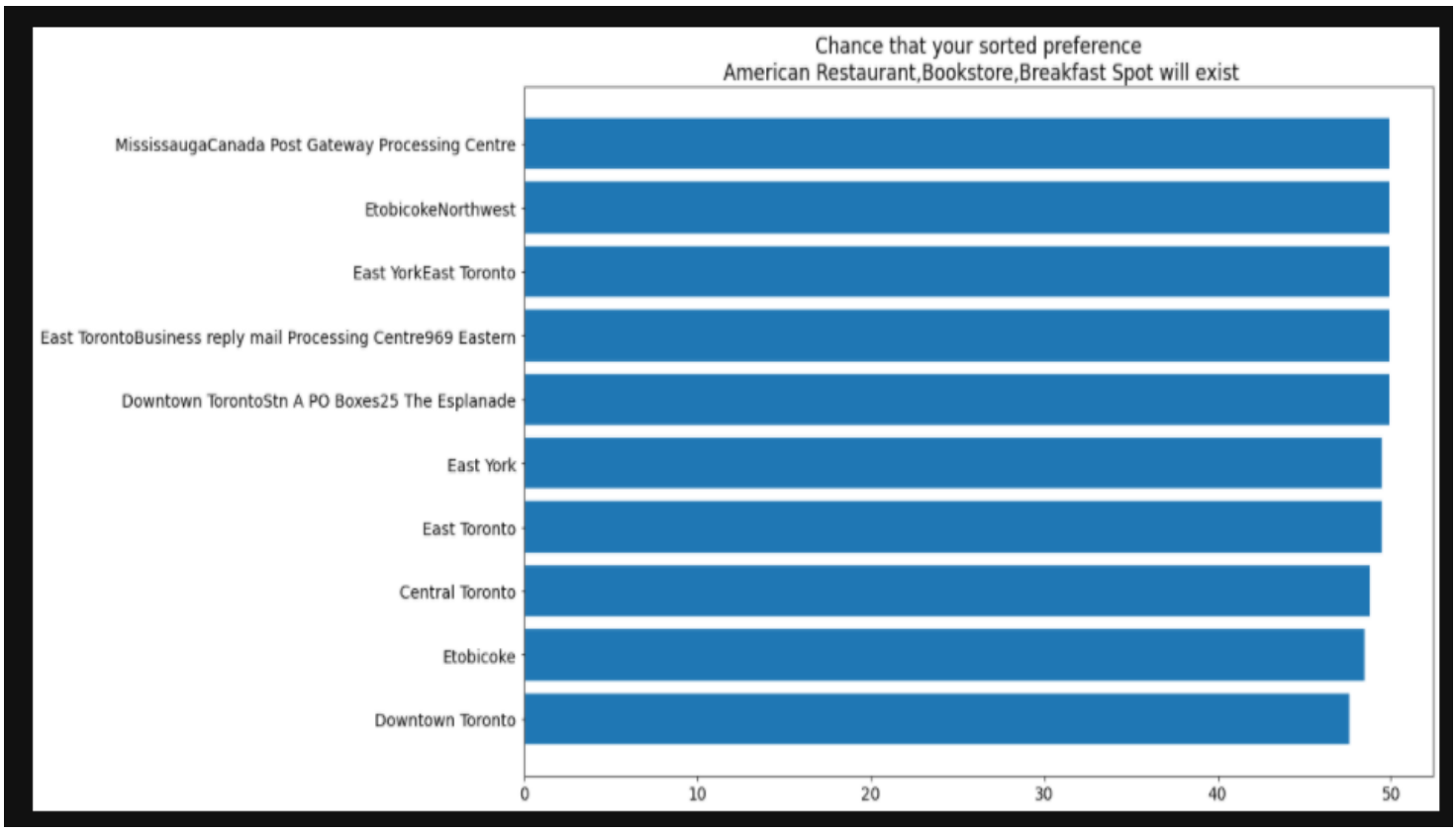
Joint Probability



Disjoint Probability



Conditional probability



DISCUSSION

The final result was multiplied by 100 for visibility i.e., the X axis is the percentage probability.

Categories at the top are fairly within the same line of business.

Categories at the bottom are mostly categories the other town/city/state doesn't have.

The use of mutual exclusivity might downplay other factors in play

ASSUMPTIONS

We assume mutual exclusivity of existence of businesses/shops, which might not be the case as some are complementary to each other therefore influencing the effect of each other.

No movement to nearby neighborhoods to get what you want i.e., the you are looking for entire combination to fit a neighborhood, so even in a case where you live close to another neighborhood and their coffee shop is closer, we assume you are going to the one in your neighborhood.

RECOMMENDATIONS

Improvements by giving a way to factor in the fact that the businesses/shops' existence isn't exactly mutually exclusive.

Find a way to allow combination of multiple probabilities e.g. If someone wants a place that has a gym given that it has restaurant and also have a coffee shop and maybe a bookstore or cinema. The first 2 will be conditional then joint then the last 2 will be disjoint.

CONCLUSION

Hopefully with the given recommendations the model can be improved and provide more accurate results/likelihoods. I believe this will come a long way to help people trying to move to new towns for business/house hunting/travel.

Finally, I'd like to thank:

- Coursera for the opportunity for the opportunity to take their course and for the challenge and data they have provided
- My peers whom we've journeyed through the course together