Quantifying Cities to Compare their Similarity

Mohammed Hussain

April 2023

1 Introduction

I like the city of Toronto, having grown up here. What if I wanted to visit a city with the same 'vibe' as Toronto? What even contributes to a city's vibe, and how would I be able to quanitfy these variables? This is the topic my project covers. Given sixty cities (Toronto, Vancouver, Montreal, New York, Ottawa, Denver, Los Angeles, San Diego, Sacramento, San Francisco, Portland, Seattle, Spokane, Houston, Dallas, Austin, San Antonio, Miami, Jacksonville, Orlando, Atlanta, Cleveland, Columbus, Cincinnati, Louisville, Chicago, Milwaukee, Detroit, Grand Rapids, St. Louis, New Orleans, Nashville, Memphis, Indianapolis, Pittsburgh, Philadelphia, Buffalo, Boston, Madison, New Haven, Little Rock, Oklahoma City, Kansas City, Washington D.C., Charlotte, Raleigh, Las Vegas, Tampa Bay, El Paso, Minneapolis, Phoenix, Tucson, Virginia Beach, Baltimore, Oakland, Charleston, Reno, Newark, Salt Lake City), **is it possible to quanitfy each city into a single score, and determine what other cities it is the most similar to?**

I had to make my own csv document to answer this question. Thus I had to decide early on which variables I wanted to include. I settled on population (in millions), % of population that is white, % Black population, % Hispanic population, % Asian population, GDP (in billions), average temperature, number of sunny days, % of population below 40, number of tourists (in millions), population density, crime rate per 100k people, accessibility by water, % of population college educated, and % of population with no religion. Of course the number of variables here could be endless, but I thought these were the most important. I got all the data for this csv file from the U.S. census bureau website and Statistics Canada.

2 Computational Overview and Instructions

Graphs are the datatype that I used to illustrate my findings. In my graph, each city is a node, and edges connect cities with scores within 1.0 of each other. My code accomplishes two things: it reads a csv file and assigns each city a score for each variable. It then constructs a graph using this information. One major dilemna I had was how to weight each variable. Ultimately I decided to give

the population size and demographics more weight than the other variables. In csv_demographic_scorer, I gave each city a score from 0 - 12, where the more non-white a city was, the higher of a score it got. However, this alone does not provide an accurate summary of a city's demographics, since two cities that are for example 70% Hispanic and 70% Asian may receive the same score but have very different demographics. Thus this function also has another variable, even_split_score. Cities with a perfectly even balance of demographics (each race making up 25% of the population) would receive a max score of 6.75, while those dominated by one ethnicity would get a score closer to 0. I was able to create this by doing an absolute value subtraction of 0.25 with that racial group's share of the population, dividing this value by 0.25, and multiplying it by 3 for each race. These two scores would be combined together to create a dictionary where the key is a city, and the associated value is the overall demographic score.

I repeated this same process for all my other variables, where I created a dictionary mapping each city to a score. To create a score for a city's GDP for example, I divided each city's GDP by the city with the maximum GDP (New York), and then multiplied this value by 10. In this category, New York as a result achieved the max score of 10. I found the maximum value in each column using a helper function, csv_max_at_specified_index. In terms of weighting, population size, GDP, temperature, number of sunny days, and density were given scores up to 10, college education was given a score up to 8, number of tourists up to 7, number of young people up to 6, and water access, religiousity, and crime rate up to 5 (higher numbers in all were given higher scores, except for religiousity and crime rate, where a lower rate was given a higher score). Also included in personalprject.py is sort_dictionary, which uses a lambda function to sort a dictionary taken in as the input according to it's associated values. There is also dict_of_neighbours, which creates a new dictionary where keys are cities, and associated values are lists of cities with scores within an input value of the key's city score (see doctest for dict_of_neighbours for more info).

In create_graph.py, the graph itself is created using dict_of_neighbours from personalprject.py. I also find the city with maximum number of edges here. It is in main.py however that most of the work happens. I create twelve different dictionaries here, one for each variable. Then using numerous for loops, I folded them all together into one dictionary where the scores are all added together. Running the file should also result in the creation of a graph as seen below.

Creating this graph required the extensive use of networkx and to a lesser extent, matplotlib.pyplot. Along with being used to create the graph, I used networkx to make my graph easier to understand. Specifically, I used the spring_layout function which repulses nodes from each other to make sure they aren't located too closely together. It also bring two nodes with an edge between them closer together. Another function in networkx I used was degree_centrality. This returns a dictionary where the key is a node, and the associated value are the number of edges it has. I used this to return the city with the maximum number of edges. Finally, matplotlib was used to display the graph.

The dataset I am using has been uploaded to MarkUs under the name city-dataset_csv. In terms of what the output of my program will look like, I have

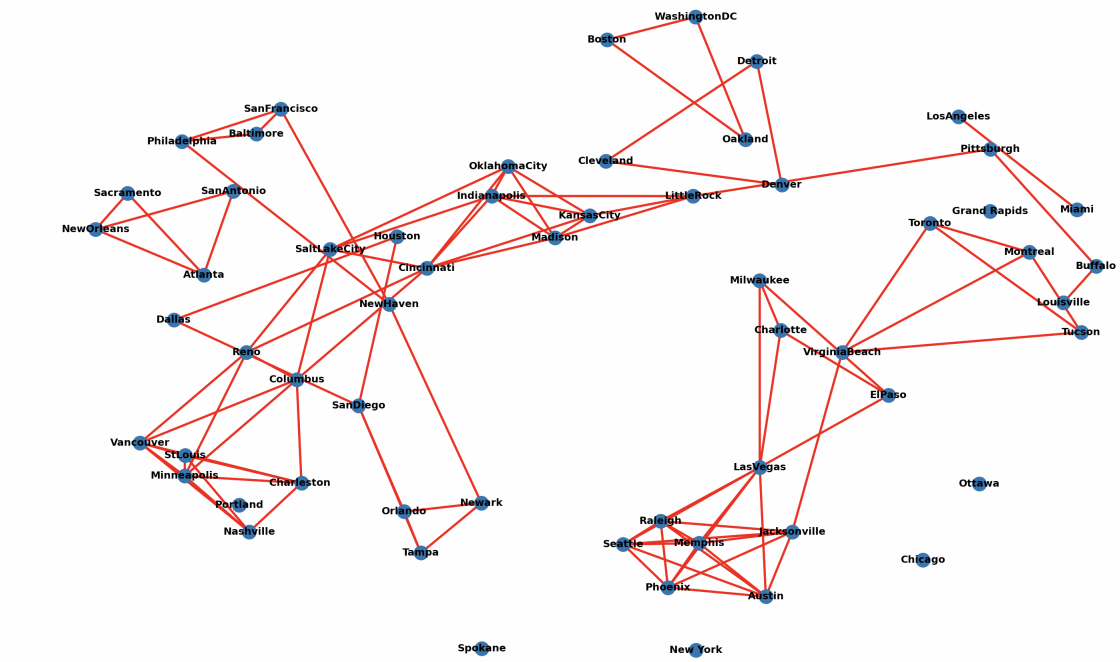

Figure 1: Graph of cities given a score from 0 - 89.75, where connected cities have scores within 1.0

already included a picture of the graph. It should also produce this:

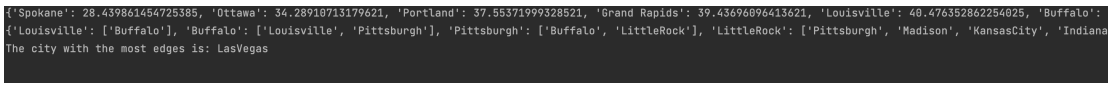

Figure 2: Sample output

where the first line is a sorted dictionary of all sixty cities according to their overall score. The second line is another dictionary where cities with similar scores are displayed. The final line displays the city with largest amount of edges.

3 Changes from Project Proposal

My original project proposal was to use a csv file from 2013 on U.S. flight data to construct a graph where nodes were airports and edges would be flights that occurred between airports. I would have weighted edges to represent the number of flights between airports. I decided not to proceed on this idea as the csv file had four million entries and trying to construct a graph with less entries would not have produced an accurate summary of the data. Additionally, my graph would have had hundreds of nodes, making it impossible to understand. This is why I decided to limit this project to sixty cities.

4 Discussion

There are so many interesting takeaways from this data. First off, its important to understand my analysis was very flawed, since there were numerous variables I left out to simplify this project, such % multiracial population, surrounding geography, and cost of living. I also had difficulty quantifying the climate of a city. The way I did this was looking at the number of sunny days and average temperature, but cities can receive identical scores on these metrics and still have wildly divergent weather patterns. The way I weighted all the variables could be considered quite arbitrary as well. That being said, the first thing that stood out to me was that the city with the most edges was Las Vegas. Reviewing the graph and the dictionary of neighbours, the results mostly seem to track. It is very conceivable to me that Buffalo and Pittsburgh, two northeastern industrial cities are similar to each other. Or that Los Angeles and Miami, two coastal cities with a large Hispanic population are similar. Looking at Toronto's associated values, the cities that my program said it was the most similar to were Montreal, Tucson, and Virginia Beach. This brings up another issue I had: how do I verify my results are correct? Having never visited any of these cities it's hard to confirm if Toronto and Virginia Beach are indeed similar. One thing I found interesting was how some cities in close geographical proximity to each other were not considered similar. Newark is just a twenty minute drive from

New York, yet it is more similar to cities in Florida than NYC. Oakland in the Bay Area was found to be more similar to Washington DC and Boston than San Francisco. Related to New York, it was one of the few cities in the graph not to have any edges. In fact in my dictionary, it received the highest score out of any city. In contrast, Spokane in Washington State received the lowest score by far. Looking at the graph, it is clear to me that cities in the middle of the United States had the most edges.

Another observation I had was how cities seemed to be composites of the cities in their associated values. I found this fascinating. For instance Austin's associated values were Jacksonville, Phoenix, Raleigh, Memphis, Seattle, and Las Vegas. Austin shares the same population as Jacksonville, but is also heavily Hispanic like Phoenix and Seattle, and has a relatively educated workforce, like Seattle.

Looking back on my project, I wonder if I haven't just made a city desirability calculator, rather than finding cities that are similar to each other. Miami and Boston may be considered desirable places to live, but that doesn't mean they have much in common with each other. I think I was able to somewhat meet my goal of finding cities similar to each other, but several of the results don't make sense to me. I find it hard to believe that Charleston, South Carolina has much in common with Vancouver. If I were to keep refining this project, I would a) introduce more variables such as those mentioned above and b) use a different method to compute city similarity.

5 References

Bureau, U. S. C. (2023, March 31). Census.gov. Retrieved April 4, 2023, from https://www.census.gov/

Government of Canada, S. C. (2023, March 10). Data. Government of Canada, Statistics Canada. Retrieved April 5, 2023, from https://www150.statcan.gc.ca/n1/en/type/data

Tutorial. Tutorial - NetworkX 3.1 documentation. (n.d.). Retrieved April 7, 2023, from https://networkx.org/documentation/stable/tutorial.html