**Introduction**

My first job was working at a pizza place, spinning dough, laying down toppings and packing fresh baked pizza into boxes to feed the hungry. It's a noble profession, and a repressed dream of mine to own and run a successful pizza shop. If you're looking for good pizza around here in Miami, not the $5 soggy cardboard kind, your options are few and far between.

Over half of all restaurants fail in their first year, and poor location choice is cited as the primary reason. This begs the question, where is a good location for a pizza place?

A data-driven approach to answering this question could involve profiling a number of popular, long running pizza shops to use as models. We could collect and chart up data points on their areas' business and residential composition, demographics, and even behavioral data such as check-ins and mentions to run through a decision tree model and determine what factors contribute to our chosen pizza place's success.

As may often be the case, that involves a lot of data I don't have, and for the purposes of this project, I limited myself to what data I could gleam off free, standard access to the Foursquare API.

I went with my pizza-loving gut on this one. I set my own criteria:

1) A lot of people nearby (Density)
2) Greater than 5000 people (Total population)
3) A lot of restaurants nearby (Suggesting a hungry populace)
4) Not a lot of pizza options nearby

I collected the relevant data and built a simple recommender system to score each Miami Dade neighborhood on how qualified they'd be to host my very good, very successful, pizza place.

**Data**

There were 3 sources for my data

Primarily, Foursquare - the 2008 social app that pioneered the formerly questionable idea of public check-ins, the kind that broadcasted your Friday night Cold Stone addiction. They are surprisingly still around, and with an impressive store of public location data for aspiring data scientists to have a go at.

The Expore endpoint from the foursquare API returns up to 50 venues near any single location queried. Each venue result comes tagged with the venue category and the geo coordinates.


Wikipedia has a page with a long list of the communities in Miami Dade, along with their 2010 population counts in a table form.

| # | Incorporated Community | Designation | Date incorporated | 2010 Population |
|---|---|---|---|---|
| 2 | Aventura | City | November 7, 1995 | 35,762 |
| 7 | Bal Harbour | Village | June 16, 1947 | 2,513 |
| 8 | Bay Harbor Islands | Town | April 1947 | 5,628 |
| 11 | Biscayne Park | Village | 1933 | 3,055 |

Each neighborhood has a link to their own Wikipedia page, where the population density is available on the right rail of every page.

| | |
|---|---|
| **Elevation** | 10 ft (2.8 m) |
| **Population** (2010) | |
| · **City** | 46,780 |
| · **Estimate** (2019)[6] | 49,700 |
| · **Density** | 3,844.67/sq mi (1,484.43/km$^2$) |
| · **Metro** | 5,422,200 |
| **Time zone** | UTC-5 (EST) |

**Methodology**

Using what I've learned from the Coursera track about Python and Pandas, I first went about scraping the Wikipedia page for the full list of Miami Dade Communities and their populations. I used the BeautifulSoup library to read the html code, and select the parts I needed.

| | Neighborhood | Population |
|---|---|---|
| 0 | Brownsville | 15313 |
| 1 | Coral Terrace | 24376 |
| 2 | Country Club | 47105 |
| 3 | Country Walk | 15997 |
| 4 | Fisher Island | 132 |

Each community in the table links to their own wiki page, where the population density value is available. I wrote a for loop to iterate over each link in the table, get the page, and find the density value to store into my dataframe.

| | Density | Link | Neighborhood |
|---|---|---|---|
| 0 | [13,983.74/sq mi (5,398.59/km, [2], )] | https://en.wikipedia.org/wiki/Aventura,_Florida | Aventura |
| 1 | [7,731.07/sq mi (2,982.07/km, [2], )] | https://en.wikipedia.org/wiki/Bal_Harbour,_Flo... | Bal Harbour |
| 2 | [14,628.79/sq mi (5,642.92/km, [2], )] | https://en.wikipedia.org/wiki/Bay_Harbor_Islan... | Bay Harbor Islands |
| 3 | [4,943.55/sq mi (1,909.08/km, [2], )] | https://en.wikipedia.org/wiki/Biscayne_Park,_F... | Biscayne Park |

To cast the density values to numbers, I removed the unit data and kept the values as miles for the analysis.

| | Neighborhood | Density_num | Population |
|---|---|---|---|
| 0 | Aventura | 13983.74 | 35762 |
| 1 | Bal Harbour | 7731.07 | 2513 |
| 2 | Bay Harbor Islands | 14628.79 | 5628 |
| 3 | Biscayne Park | 4943.55 | 3055 |
| 4 | Coral Gables | 3844.67 | 46780 |

Using the Foursquare API, I created a table of the neighborhoods' venues and their categories. Most neighborhoods returned the full expected 50, but over a couple dozen neighborhoods returned significantly less.

| Neighborhood | Venue | Venue Latitude | Venue Longitude | Venue Category |
|---|---|---|---|---|
| Aventura | 50 | 50 | 50 | 50 |
| Bal Harbour | 50 | 50 | 50 | 50 |
| Bay Harbor Islands | 22 | 22 | 22 | 22 |
| Biscayne Park | 12 | 12 | 12 | 12 |

I created dummy columns for every category of venue, and filtered for columns containing the words 'Restaurant', 'Steakhouse', or 'Place', as these were the words used to identify food service venues.

I summed the count of restaurants and created the column, 'Restaurants to People' for my third criterion point

For my ‘Pizza places to Restuarants’ column values (Fourth criterion), I divided the number of pizza places by the total number of other restaurants.

| | Neighborhood | Density_num | Population | Restuarants to People | Pizza Places to Restuarants |
|---|---|---|---|---|---|
| 0 | Aventura | 13983.74 | 35762 | 0.001001 | 0.071429 |
| 1 | Bal Harbour | 7731.07 | 2513 | 0.001035 | 0.000000 |
| 2 | Bay Harbor Islands | 14628.79 | 5628 | 0.000684 | 0.100000 |
| 3 | Biscayne Park | 4943.55 | 3055 | 0.000809 | 0.000000 |
| 4 | Coral Gables | 3844.67 | 46780 | 0.005462 | 0.095238 |
| 5 | Cutler Bay | 4421.77 | 40286 | 0.002262 | 0.000000 |
| 6 | Doral | 4757.63 | 45704 | 0.004624 | 0.045455 |
| 7 | El Portal | 5778.31 | 2325 | 0.000346 | 0.000000 |
| 8 | Florida City | 1965.11 | 11245 | 0.005598 | 0.090909 |
| 9 | Golden Beach | 2861.96 | 919 | 0.000000 | NaN |
| 10 | Hialeah | 10812.24 | 224669 | 0.001757 | 0.052632 |

I changed NaN values to 0, as they were a result of trying to divide by 0 (0 restaurants in the above case)

I dropped neighborhoods with less than 5000 people. These smaller neighborhoods with high density ended up being deceptively high scoring later on.

There was a large range between population numbers and the ratios. I planned to score against the criteria, and that would require positive normalized values. Using scikit-learn’s MinMaxScaler class, I brought all the numbers down to a similar range between 0 and 1.

| | Neighborhood | Density_num | Population | Restuarants to People | Pizza Places to Restuarants |
|---|---|---|---|---|---|
| 0 | Aventura | 0.617920 | 0.077394 | 0.081149 | 0.142857 |
| 2 | Bay Harbor Islands | 0.649550 | 0.000951 | 0.050038 | 0.200000 |
| 4 | Coral Gables | 0.120748 | 0.105344 | 0.518153 | 0.190476 |
| 5 | Cutler Bay | 0.149046 | 0.088870 | 0.204618 | 0.000000 |
| 6 | Doral | 0.165515 | 0.102614 | 0.436065 | 0.090909 |

To each criterion, I assigned a weight. This weight represented the importance of the criterion in relation to all the others. The weights I assigned each point were:

- 40% of the decision will be based on the area's population density,
- 10% on the area's overall population
- 30% on the area's restaurant to people ratio,
- 20% on the area's pizza place to restaurant ratio, a lower ratio being more favorable.

The actual weight values assigned were 4,1,3,-2, respectively. The -2 was used for the Pizza place to Restaurant' column, because a lower ratio was favorable. The normalized column's values were multiplied by their weights to score, and the sum of those scores were used to determine the area's favorability.

**Results**

The top 6 locations for a new pizza place were:
-drumroll-

| | Density_num | Population | Restuarants to People | Pizza Places to Restuarants | Neighborhood | scores |
|---|---|---|---|---|---|---|
| **29** | 3.959024 | 0.039520 | 0.139993 | -0.000000 | Sunny Isles Beach | 4.138538 |
| **16** | 2.278811 | 1.000000 | 0.265701 | -0.000000 | Miami | 3.544512 |
| **22** | 4.000000 | 0.004779 | 0.124667 | -0.615385 | North Bay Village | 3.514062 |
| **62** | 0.087891 | 0.024845 | 3.000000 | -0.210526 | Three Lakes | 2.902210 |
| **10** | 1.849619 | 0.556605 | 0.465655 | -0.210526 | Hialeah | 2.661353 |
| **0** | 2.471680 | 0.077394 | 0.243446 | -0.285714 | Aventura | 2.506806 |

The worst locations were

| | | | | | | |
|---|---|---|---|---|---|---|
| **25** | 0.452221 | 0.025281 | 1.064903 | -1.142857 | Opa-locka | 0.399548 |
| **57** | 1.070734 | 0.077227 | 0.121051 | -1.000000 | South Miami Heights | 0.269012 |
| **65** | 0.721132 | 0.010672 | 0.588253 | -1.090909 | West Perrine | 0.229149 |
| **20** | 0.546345 | 0.013293 | 0.936439 | -1.428571 | Miami Shores | 0.067506 |
| **51** | 1.067557 | 0.000000 | 0.078399 | -1.333333 | Palm Springs North | -0.187378 |
| **41** | 0.389438 | 0.012303 | 0.123750 | -2.000000 | Goulds | -1.474508 |

These bottom locations were negatively impacted by the number of pizza places already there. I double checked on Google maps, and they were indeed pizza rich neighborhoods.

**Discussion**

I hadn't expected Sunny Isles to be the top qualifier for a pizza shop. Thinking back at my visits, I recall many condos out there. Sure enough, the density scores for Sunny Isle corroborated my impressions.

The next highest score was Miami city proper, which covers a very large area, and I can see that no pizza places were returned for Miami. That simply is not accurate.

Which brings me to a discussion on the limitations of my work.

- My Foursquare access limited the venues I could count to 50 per area. This becomes a problem when the area is as big as a large city. I assumed the 50 venues returned, a representative sample of the business composition of the area.
- A neighborhood recommendation is a start, but a more practical recommendation would be at the physical commercial property level. This would be possible using county data, or a list of available properties.
- While all the skills from the Coursera Data Science track culminated into this report, there is no actual machine 'learning' used here. My scoring system is based off a recommender system, similar to how Netflix might suggest movies you'd like, except the learning part involved in the analysis of what a viewer likes is missing. It's been replaced by the weights I manually assigned to the criteria.
- I've learned that census data is relatively difficult to work with. The latest population figures I could find for Miami Dade neighborhoods were from 2010. It seems a similar, comprehensive and updated list may not be available until after the 2020 census.

**Conclusion**

I was disappointed to see there was no pizza place required anywhere closer to my neighborhood, but was reminded it's because I already have 2 great pizza places by me in South Miami, the data confirms it.

This project was completed as my capstone project for the Coursera IBM Data Science Certificate. It took me 10 months to complete the track, but I'm proud to say I stuck it out the whole way. It's been a great way to learn what is, and how to 'do' data science. You'll get out what you put in to it. I recommend anyone interested in learning more about data and Python to consider Coursera.