New York Pizza Store Map Project

Course: Programming For Data Science
Prepared by: Asena Derin Cengiz, Ksenia Saenko, Preet Gandhi, Stella Sun(ss8955)
Term: Fall 2017

1. INTRODUCTION

The goal of this project was to create an interactive, color-coded visualization of pizza store locations in NYC. We wanted to produce two maps: one for real-time visualization of nearby pizza restaurants in Manhattan that would allow a user to make an informed choice of a restaurant. This map would include price and rating information for each restaurant.
The second deliverable would be a heat map of pizza restaurants in Manhattan. It would be useful for such applications as competitive analysis for other pizza owners and rental selection based on the apartment proximity to pizza stores for consumers.

2. METHODOLOGY

All code was written in Python 3 using Jupyter Notebooks.

We identified four key parts for designing the project:

1) Obtaining New York restaurant locations using Google Places API
2) Calculating distances density of restaurants in each zip-code area
3) Creating a simple scatter plot map that display all pizza restaurants in Manhattan
4) Creating a heat map to show distance density on zip-code level
5) Creating an interactive map of Pizza restaurants in New York

2.1 Obtaining restaurant locations from Google Places API.

Packages and Functions: urllib, json, pandas

The first step of the project was to find and process information on pizza restaurant locations in Manhattan. To accomplish this task, we used Google Places API. The API allows to query for place information on a variety of categories and returns a list of places along with summary information about each place. We obtained a unique API key, which allowed us to create query using the keyword “pizza”.

In order to obtain restaurants in Manhattan, we used four sets of latitude/longitude coordinates corresponding to uptown, central park area, midtown, and downtown areas of the city. Providing more coordinates for the API would return even more granular results.

We created a function to query Google API using our criteria. The API returns only 20 results per page, hence we had to use token created by the previous query result in order to read more pages.

We used urllib.request to query API. The output of Google API is in JSON format, so we used json and Pandas to write data into a dataframe for processing.

Processing of the data included removing duplicates from the query using Pandas. We created a new field “borough” indicating if each restaurant is actually located in Manhattan and removed restaurants in other locations. We split the field “geometry” into two fields: latitude and longitude in order to be able to map the restaurants.

Finally, we wrote the output into a CSV file called “pizza_locations_google.csv”. The file included such as information as name of the restaurant, geographic location, price level, and rating.

2.2 Obtaining geographical data and transforming coordinates

Packages and Functions: pandas, pyproj, mapnik

We got the geographical/land use data from PLUTO which is maintained by the city agencies. Relevant columns were extracted from the csv and null rows were dropped for data cleaning. The coordinated present in the PLUTO data are local coordinates which are of New York- Long Island plane. They need to be converted to the uniform longitude/latitude data. New York is EPSG:2263 , according to the standard coordinate transform convention. It has to be converted from [Projected Bounds: 909126.0155, 110626.2880, 1610215.3590, 424498.0529] to [WGS84 Bounds: -74.2700, 40.4700, -71.7500, 41.3100]. Using pyproj, first I create a projection layer using the fields found on the spatial reference website for NYC. Then the layer is applied to the PLUTO data with the inverse transform field set to True as we are converting from local to global coordinate system. Then the data is outputted to a csv file ‘manhattan_lot_locations.csv’ and a unique lot id is assigned for differentiation.

2.3 Calculating Distances and Density of Restaurants in Each Zip-code Area

Packages and Functions: pandas, numpy, math, matplotlib

We used the latitude - longitude records of each pizza restaurant as a link to the Manhattan Lot Locations dataframe. To calculate the distance between two points on New York given their latitudes and longitudes, we used Haversine Formula, the mathematical expression of the distance between two points on a spheroid. By calculating the distances of pizza restaurants to Manhattan’s lots, we were able to plot each restaurant on the map.

Then, we improved our code to calculate the density of pizza restaurants within each zip code area by grouping pizza location data by zip code, getting the average latitude-longitude data for zip codes, and masking the dataframe that shows the distance of each pizza place to the zip code centers by distance. Therefore we were able to create a visualization for the density of the zip-code areas in terms of pizza restaurant numbers. We have also written a function that connects any restaurant to the closest zip code center.

Important Note:

list_distance function calculates each pizza restaurant’s distance to the lots in Manhattan and writes them into a csv file - **if you are running this file on a time constraint, you do not have to run this function again**. **The csv is also included (therefore the function is commented out), and the code will read it back into a DataFrame.**

2.4 Creating a simple scatter plot map that display all Manhattan pizza restaurants

By reading the pizza_locations csv file, we were able to plot all pizza stores with its coordinated on the map to show how pizza stores distributed in Manhattan.

2.5 Creating a heat map to show distance density on zip-code level

Packages: plotly, json, pandas, numpy, mapbox, numpy

Data:

- Distance Density Dataframe that included the distance density (between 0 and 1) for each zip-code area
- ManhattanZipCode geojsom data downloaded from www1.nyc.gov/ that contained the boundary coordinates for each zip-code area

Methodology:

What we were going to do was to extract the original geojson data, add Distance Density info for each zip-code, and generate new geojson data for each zip-code.

We first filtered the json data with Manhattan zip-code area only, since our pizza stores were in Manhattan only. After reading the Distance Density dataframe, we generated a new geojson file for each zip-code by adding the distance density information to the original json data. We ended up getting 52 geojson files for 52 zip-code area in Manhattan, with its default features such as geometry (boundary coordinates), and new info we added, distance density.

To make a heatmap, we used the package called “plotly” that implemented mapbox, an open map source, so that we could display the mapbox map in Python. In order to show the difference density, we set a color scale from 0 to 1, in step of 0.1, and the darker color, the higher distance density in that zip-code area.

With the given distance density for each zip-code area, we assigned the color to each json file, and make a layer to plot on the map. The way to generate the layer was based on the documentation: https://plot.ly/python/reference/#heatmap.

2.6 Creating an interactive map of Pizza restaurants in New York

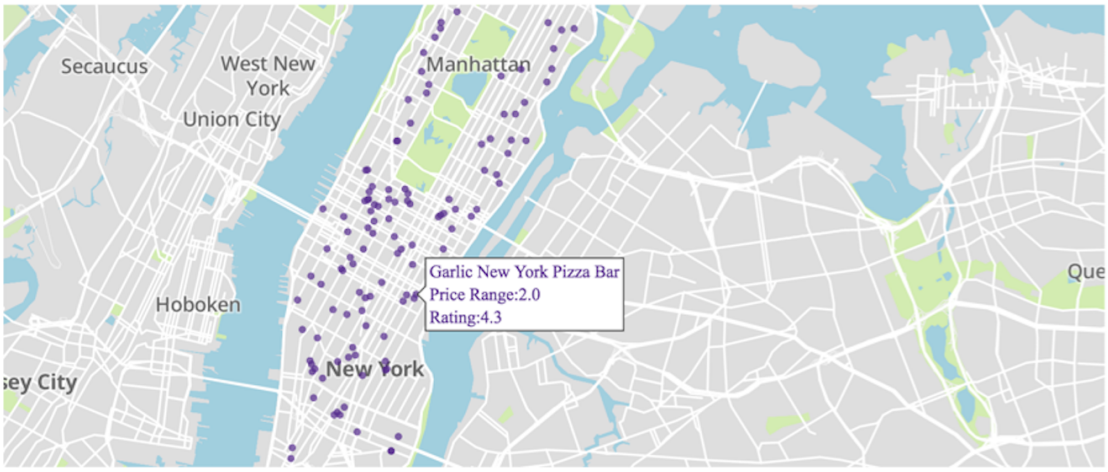
With the same methodology, we created a more interactive map visualization to show users all the pizza stores that open now.

By running the file, we called the API to get the pizza store information and only keep those that were open at the time running the file. And then we just plotted on the map to show users all the open pizza stores.

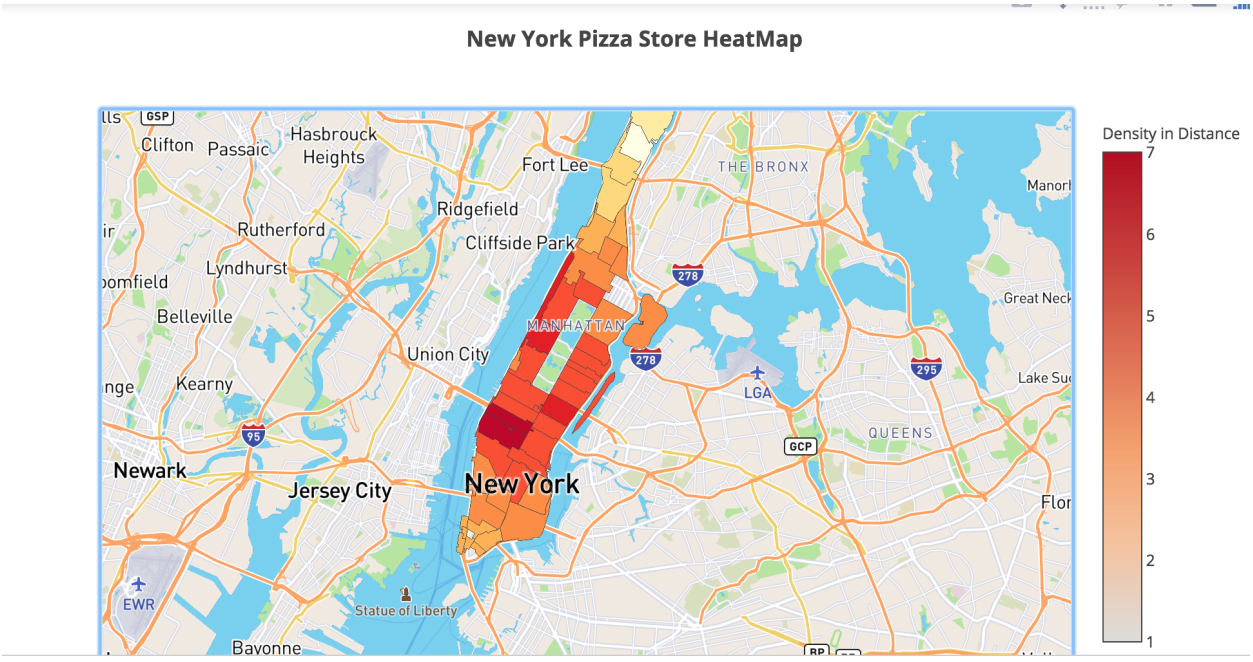
3. RESULTS

We visualized locations of pizza restaurants in Manhattan using interactive maps.

3.1. We produced a map showing pizza store locations in Manhattan. Hovering over each restaurant allows to see more useful information about it, such as its price level and rating.



3.2 We created a heat map of pizza restaurants in Manhattan. This allows to visualize which neighborhoods are more friendly for pizza consumers, as well as understand competition for pizza business owners.



A. Discussion

Here you discuss the results.

4. CONCLUSION

We found the areas with the highest density of pizza restaurants in Manhattan. Hell's Kitchen, or area near Times Square/Port Authority Bus Terminal has the higher number of pizza stores. Probably many commuters as well as tourists prefer to grab a quick inexpensive meal in that area, explaining the high density. Financial district had the lowest number of pizza stores.

Our code also allows the user to visualize pizza restaurants that are open at the time of search. Hence a user can find restaurant near her that are open and see their rating and average price.

Our technique could be easily applied for searching and understanding Manhattan presence of other restaurants as well.