AN EXPLORATION OF ATLANTA NEIGHBORHOOD BOUNDARIES USING KNN CLASSIFIERS

KATIE FULLERTON

1. Introduction

1.1. **Background.** Atlanta is a major city in the south-eastern portion of the United States originally founded at the intersection of multiple railroad lines. As a city it has grown extensively in the last 50 years. Much of the planning efforts for that growth have been left to commercial developers, and consequently the city is somewhat a jumble of neighborhoods. Residents often complain that neighborhood boundaries are poorly-defined and have little connection with the actual usage patterns of the space. The city of Atlanta is divided into a number of Neighborhood Planning Units (NPUs). Each NPU is administered by a citizen and city employee council. These councils are empowered in various ways across the city, and have varying degrees of neighborhood buy-in. However, most NPUs do exert influence in the city zoning office, and thus can effect the type of development allowed within their borders.

1.2. **Problem.** This system of NPUs was designed in the late 1970's. The past 50 years have seen dramatic shifts in population density, usage, and demographics in the Atlanta area. The purpose of this project is to compare the NPU boundaries designed in the 1970's to actual location usage patterns and determine if those boundaries still represent meaningful delineations. This analysis would be of use to the city's planning board. If NPUs truly reflected the ways that citizens use their space, they would be more likely to get involved in the administration of those neighborhoods. Increased civic engagement has benefits both for the citizens and the city. This data might offer commercial value to developers looking to appeal to a specific consumer type, or to place new developments in areas of high likely usage.

2. Data

2.1. **Data Sources.** In order to perform the analysis, we will need two separate data sets. The first required data set is the geospatial boundary data for each NPU, as designed in the 1970's. The second required data set must capture current usage patterns of the spaces in those geospatial regions. For the first dataset, we will access the City of Atlanta's GIS System via their website. The second dataset will be collected through the Foursquare API.

It is assumed that Foursquare data represents real-time, up to date information about the way people live, work, and play in their neighborhoods.

2.1.1. *Geospatial Data.* The City of Atlanta offers a useful API explorer at `https://dcp-coaplangis.opendata.arcgis.com/datasets/npu/geoservice` . The resulting json file contains the information below. The fields of interest for this investigation are the NAME and geometry fields.

```
{ ”attributes”: {
           ”OBJECTID”: 260,
           ”LOCALID”: null,
           ”NAME”: ”K”,
           ”GEOTYPE”: ”NPU”,
           ”FULLFIPS”: null,
           ”LEGALAREA”: null,
           ”ACRES”: 1528.29,
           ”SQMILES”: 2.39,
           ”OLDNAME”: null,
           ”NPU”: null
        },
        ”geometry”: {
           ”rings”: [
              [
                 [
                    −84.4173772073577,
                    33.772197013770004
                 ],
```

Using this tool, we generated a request URL `https://gis.atlantaga.gov/dpcd/rest/services/OpenDataService/FeatureServer/4/query?where=1%3D1&outFields=NAME&outSR=4326&f=json` to create a simplified output object with only the fields of interest.

2.1.2. *Location Usage Data.* In order to collect a sufficient amount of data for the large geographical area covered, we created a latitude and longitude search grid. This grid was set to contain 10 steps between the minimum and maximum latitude and longitude values present in the GIS data. The resulting search grid can be seen in Figure 1. For each coordinate in Figure 1, a url was generated to query the Foursquare API. An anonymized sample URL is: `https://api.foursquare.com/v2/venues/explore?client_id=XXXXXXXX&client_secret=XXXXXXX&ll=33.886869733912235,-84.28962468321286&v=20180604&limit=50&radius=4500`. The results of each API call were compiled into a single large dataframe for cleaning.

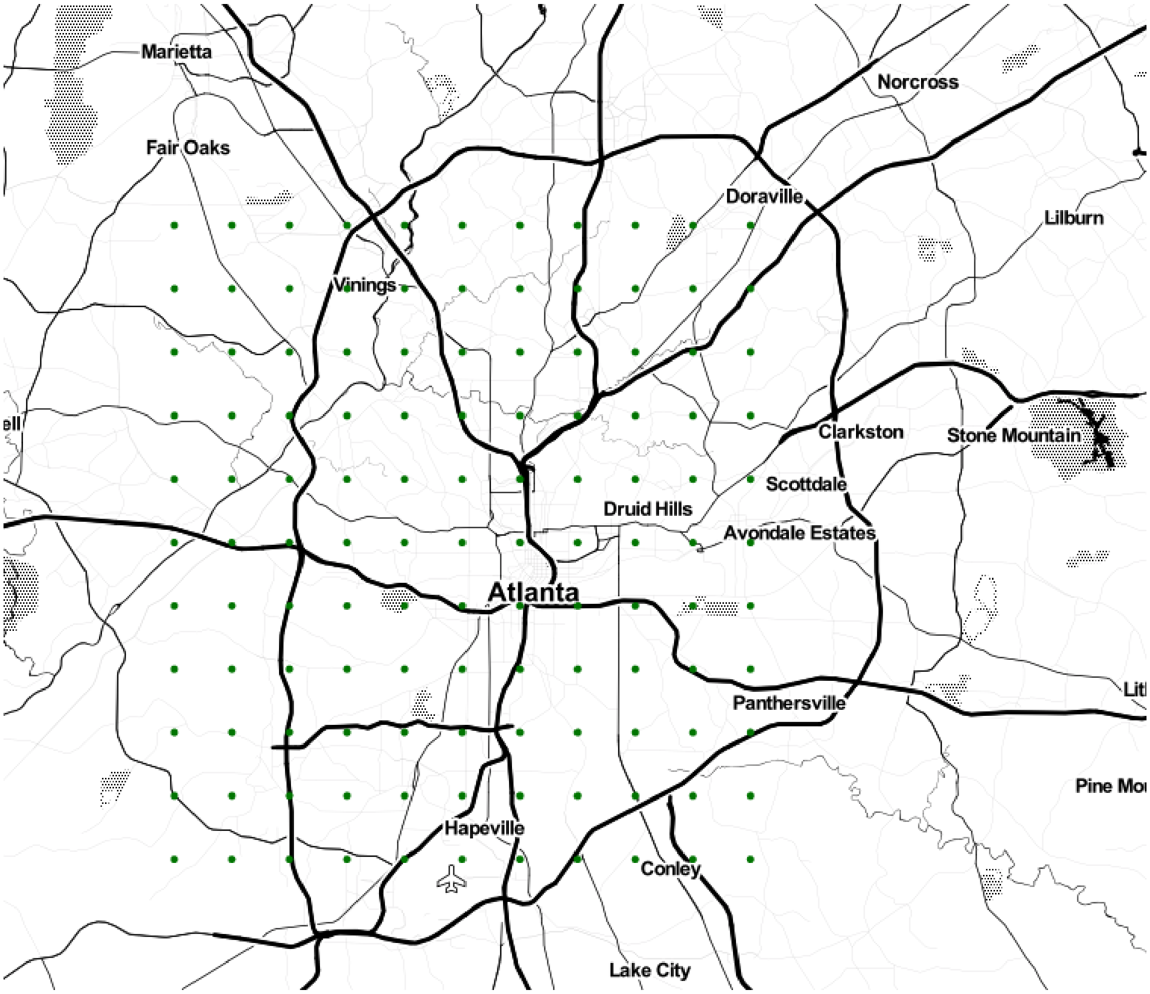


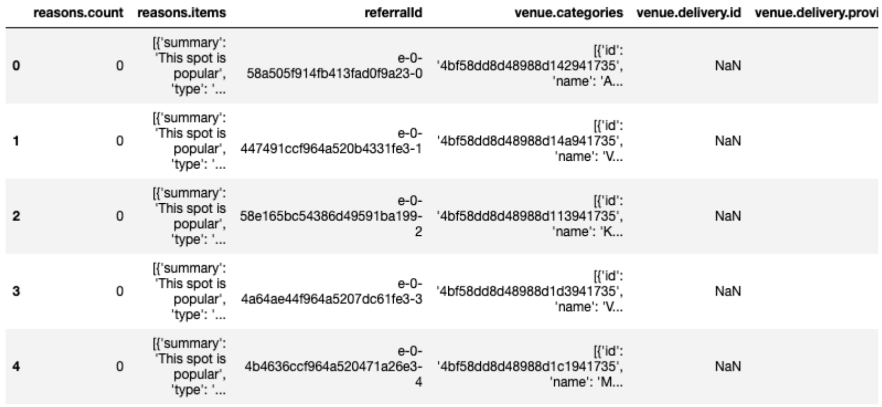
Figure 1. Latitude and Longitude Search Grid

2.2. **Data Cleaning.**

2.2.1. *Cleaning Geospatial Data.*

2.2.2. *Cleaning Location Usage Data.* Once retrieved, the location usage data contains significantly more data than needed for this analysis. The initial output format can be seen in Table 2

The name, latitude, and longitude attributes could be directly extracted. The category field contained additional structure, so a function was written to extract the category string from that structure, as shown below. Initially, it appeared that the Foursquare data



	reasons.count	reasons.items	referralId	venue.categories	venue.delivery.id	venue.delivery.provi
0	0	[{'summary': 'This spot is popular', 'type': '...	e-0-58a505f914fb413fad0f9a23-0	[{'id': '4bf58dd8d48988d142941735', 'name': 'A...	NaN	
1	0	[{'summary': 'This spot is popular', 'type': '...	e-0-447491ccf964a520b4331fe3-1	[{'id': '4bf58dd8d48988d14a941735', 'name': 'V...	NaN	
2	0	[{'summary': 'This spot is popular', 'type': '...	e-0-58e165bc54386d49591ba199-2	[{'id': '4bf58dd8d48988d113941735', 'name': 'K...	NaN	
3	0	[{'summary': 'This spot is popular', 'type': '...	e-0-4a64ae44f964a5207dc61fe3-3	[{'id': '4bf58dd8d48988d1d3941735', 'name': 'V...	NaN	
4	0	[{'summary': 'This spot is popular', 'type': '...	e-0-4b4636ccf964a520471a26e3-4	[{'id': '4bf58dd8d48988d1c1941735', 'name': 'M...	NaN	

FIGURE 2. Raw data output from Foursquare

did include a neighborhood categorization. However, as the analysis continued, it became apparent that this field is either 'NaN' or left out of the data set entirely. While processing, a conditional clause was created, as seen below, to facilitate the compilation of the data. In the final model, this data was determined to offer no additional value to the analysis.

LISTING 1. Category Extraction Function

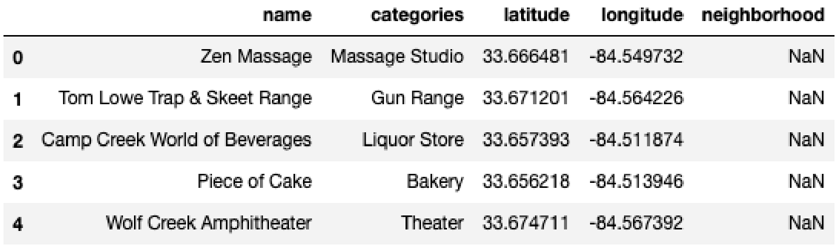
```
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

    if len(categories_list) == 0:
        return None
    else:
        return categories_list[0]['name']
```

LISTING 2. Neighborhood Data Screening Clause

```
if 'venue.location.neighborhood' in list(dataframe.columns.values):
        data['neighborhood'] = dataframe['venue.location.neighborhood']
    else:
        data['neighborhood'] = np.nan
```

The search grid was constructed orthogonally, while the Foursquare interface assumes a circular search radius. Consequently, there are likely to be a number of locations that are captured in more than on API call. Duplicate data does not provide additional information for this analysis, and so duplicates were removed. The initial search resulted in 5970 total



	name	categories	latitude	longitude	neighborhood
0	Zen Massage	Massage Studio	33.666481	-84.549732	NaN
1	Tom Lowe Trap & Skeet Range	Gun Range	33.671201	-84.564226	NaN
2	Camp Creek World of Beverages	Liquor Store	33.657393	-84.511874	NaN
3	Piece of Cake	Bakery	33.656218	-84.513946	NaN
4	Wolf Creek Amphitheater	Theater	33.674711	-84.567392	NaN

Figure 3. Cleaned Location Dataframe

location listings, with only 1913 unique values. It was determined that this is a sufficient number for training and testing of a model. However, correcting the discrepancy between square and circular searches could increase the number of unique data points for future work.

2.3. **Exploratory Data Analysis.**

2.3.1. *NPU Boundaries.* The NPU boundaries were described by a series of coordinates for each boundary.

2.3.2. *Foursquare Venue Data.*

3. Methodology

3.1. **NPU Assignment.**

3.2. **KNN Classifier Construction.**

3.3. **Cross Validation.**

4. Results

4.1. **Scoring.**

4.2. **Visual Analysis.**

5. Discussion

6. Conclusion

6.1. **Future Work.**

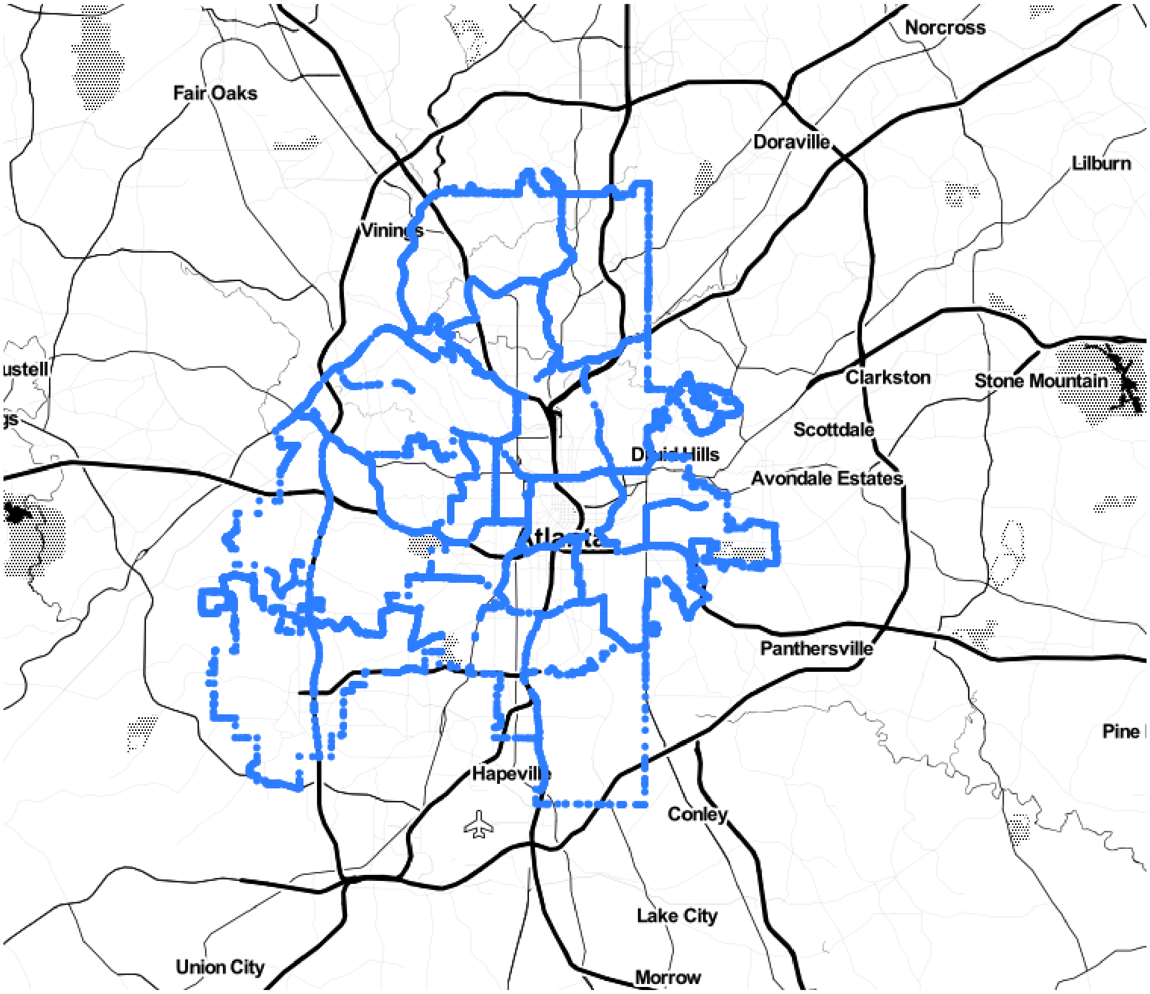


Figure 4. NPU Boundaries in the City of Atlanta

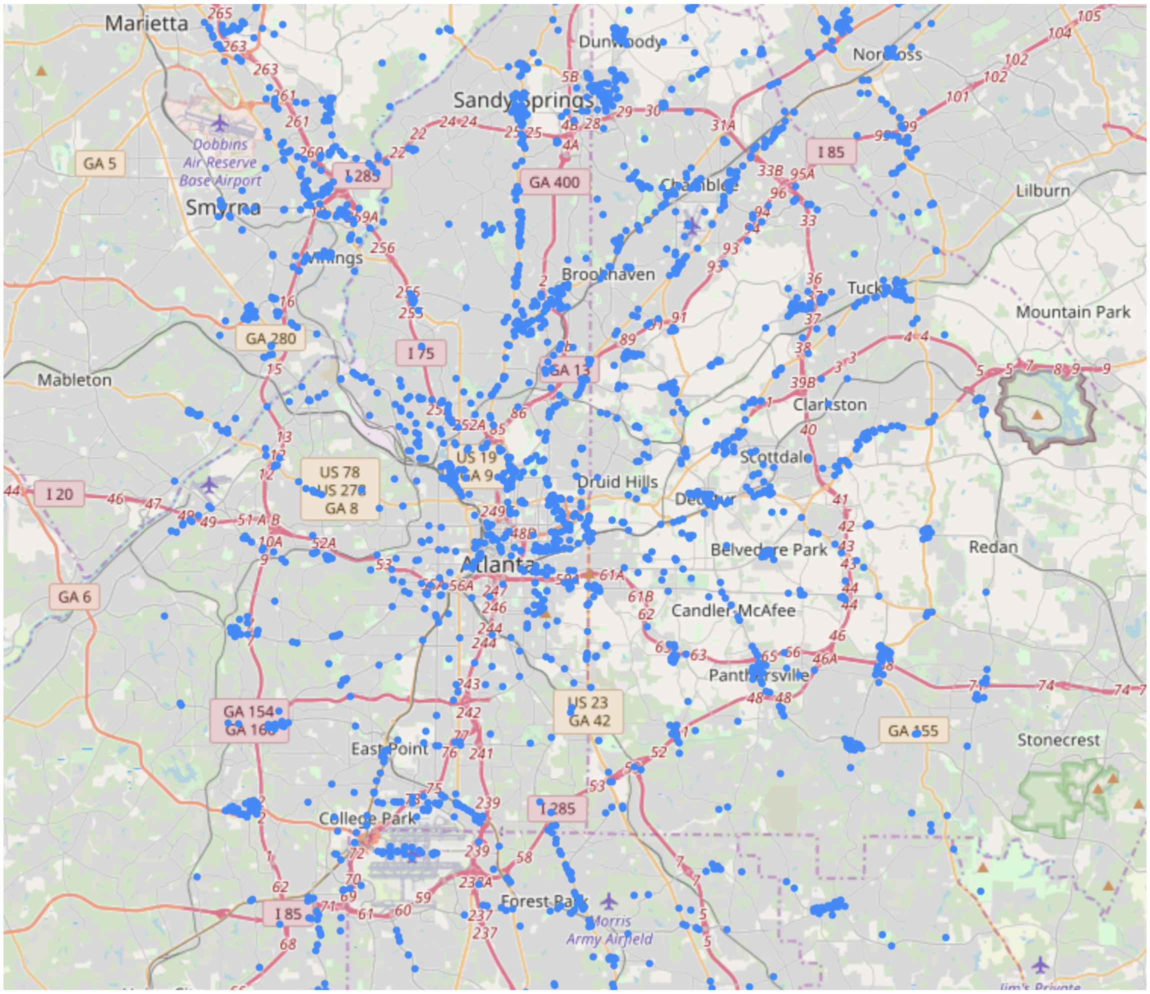


Figure 5. Results of Foursquare API call