Data Science Project: Business Location Optimization

Adam Davis

June 1, 2020

"There are three things that matter in property: *location, location, location*."
--Lord Harold Samuel, British real estate tycoon.

Introduction

Businesses can succeed or fail based on their location. This project will attempt to find an optimum location based on the presence of beneficial current businesses (traffic drivers) and existing competitors. This project will attempt to find a good neighborhood in the city of Baltimore, Maryland for a nighttime live music venue. To be successful, the location should have a sufficiently robust nightlife to support another nighttime business, but hopefully not too many competitors with nighttime entertainment. We will look for neighborhoods with restaurants, bars, and cafes, but with fewer live music venues. If a neighborhood has no restaurants or bars, it probably isn't a place that people frequent at night for entertainment, and it seems less likely that people out for the evening would want to make a special trip just for one venue. On the other hand, with sufficient bars and restaurants, the neighborhood would be enhanced by an additional nighttime venue and help add to the overall business traffic. If an area can be found within a reasonable distance of two nightlife centers, that might be optimal, since it could help bridge the two and benefit from the traffic at both. This data would be valuable to a business owner who wanted to start a new music venue or real estate owners trying to establish the possible value or marketing of their property.

Data

The geographic region searched will be limited to the formal legislated Baltimore city limits as published by the city government.[1]

We will use Foursquare venue location data to find clusters of bars, restaurants, clubs, music, and nighttime entertainment venues within the city and determine where they are and where they are missing. We will try to find ideal locations amongst other existing nightlife destinations that aren't already served by music venues.

[1] The administrative border of Baltimore city can be exported in the form of longitude, latitude tuples from the city website: https://data.baltimorecity.gov/Geographic/Baltimore-City-Line/rz8b-wbi9

Within the city limits, we will collect two sets of Foursquare data. Foursquare provides properties of each venue returned. The ones relevant to this project are street address, postal code, latitude, longitude and the business categorized into one or more categories.

The first set are venues that tend to drive traffic to a neighborhood by using the Foursquare category of Food. This category includes a wide spectrum of restaurant types, sit down, fast food, ethnic restaurants, cafes, etc.[2] It does not include supermarkets or food stores, which have their own category under Food and Drink Shop.

The second set of venue data is the list of competitors. By reviewing the list of Foursquare categories, we selected a group of Competitor venue types that could be considered to be competition for a nighttime live music venue.

Initially we considered the Arts & Entertainment category, but upon inspection, decided this group was too broad, as it included venues such as Historic Site, Memorial Site, Museums, and Public Art. Based on the criteria of venues that might host nighttime entertainment, the categories were reduced to the following specific subset within the broader Arts & Entertainment group: Music Venue, Amphitheater, Circus, Comedy Club, Concert Hall, Country Dance Club, Music Venue (jazz/piano bar/rock club), Performing Arts Venue, Nightclub, Other Nightlife, and Strip Club.

Methodology

The first issue was determining how to limit Foursquare to a set area. Foursquare supports searching for venues by zip codes. It is simple enough to find a list of Baltimore City postal (zip) codes.[3] But hopes of using zip codes were dashed by another quick internet search that shows the zip codes crossing the city border.[4] Zip codes also seemed ill advised because they are only supported by when searching for Foursquare venues with intent=match, and then there is a bit of a disclaimer in the Foursquare docs that zip codes are not required, and a note about sensitivity to the parameter, which seems to invite greater error.

[2] The full list of Foursquare venue categories is available here: https://developer.foursquare.com/docs/build-with-foursquare/categories/

[3] A quick internet search finds Baltimore city zip codes: http://www.ciclt.net/sn/clt/capitolimpact/gw_ziplist.aspx?ClientCode=capitolimpact&State=md&StName=Maryland&StFIPS=&FIPS=24510

[4] https://www.zipdatamaps.com/zipcodes-baltimore-md

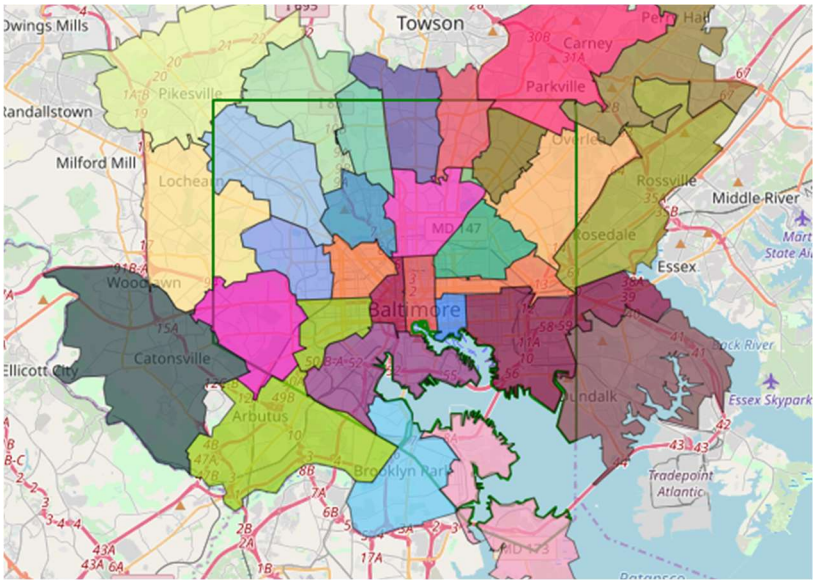


Figure 1 Zip code map. Each colored region is a zip code showing how they extend over the city border in green.

Querying Foursquare involves using a parameter called intent, which has several values: match, browse, checkin, etc. Another shortcoming of zip codes when combined with intent=match is that it seems to be created for those searching for venues matching a set of parameters. We really are looking to get all the venues in an area. For this intent=browse seems better suited. The browse intent also supports a bounding box and appears to supports returning all venues.

Using the latitude/longitude coordinates of the official published city boundaries on a city website we draw the set of boundaries shown on the maps below.[5] Since the city of Baltimore is largely rectangular with straight sides, it seemed a simple matter to divide the 15.51 km x 19.5km city into a 30 x 38 grid of squares approximately 500m x 500m each. We determine the latitude and longitude of each square's vertex using the latitude and longitude of the corners of the city box and using the numpy.linspace function to divide them into 31 points east to west and 39 points north to south. Numbering each square's Folium popup with its index allowed us to identify the indexes of the ones that extended more than 50% over the city line and remove them.[6] We remove 239 squares from the original 1,140 and are left with 901 squares as shown in Figure 2. Using squares 500 meters on a side also allowed us to stay well within the Foursquare stricture that limits responses to 50 venues per call. We save the pandas.dataframe of squares and the coordinates of each corner for later use.

[5] https://data.baltimorecity.gov/browse?category=Geographic&provenance=official

[6] See the method GetSquaresToDiscard() in ClubWorthy.ipynb

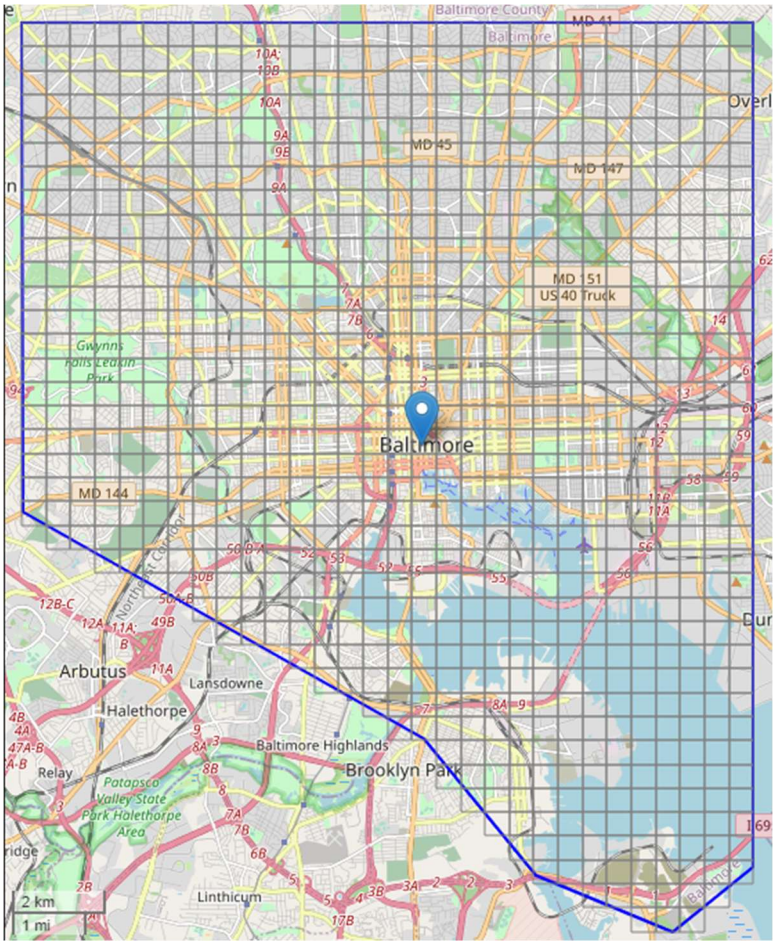


Figure 2 Map with 500m x 500m showing squares remaining after discarding ones outside the border.

Using the northeast and southwest corners of each square as a bounding box, and with intent=browse, we query Foursquare for traffic driving venues and competitor venues and save these as two separate data frames. We map these to see the raw data. Foursquare returns 1539 traffic driving venues, and 234 competitor venues. We save the results in two dataframes, traffic drivers in one and competitors in the other.

Figure 3 shows the traffic drivers in blue and the competitors in red. The circles are centered in the center of each square and the radius is proportional to the number of venues found within the square. The center of Baltimore stands out for the high number of blue food venues, extending east and north. It is quickly clear that using the centers of the squares is somewhat misleading and produces an unwanted effect of forcing the data points into predetermined locales.

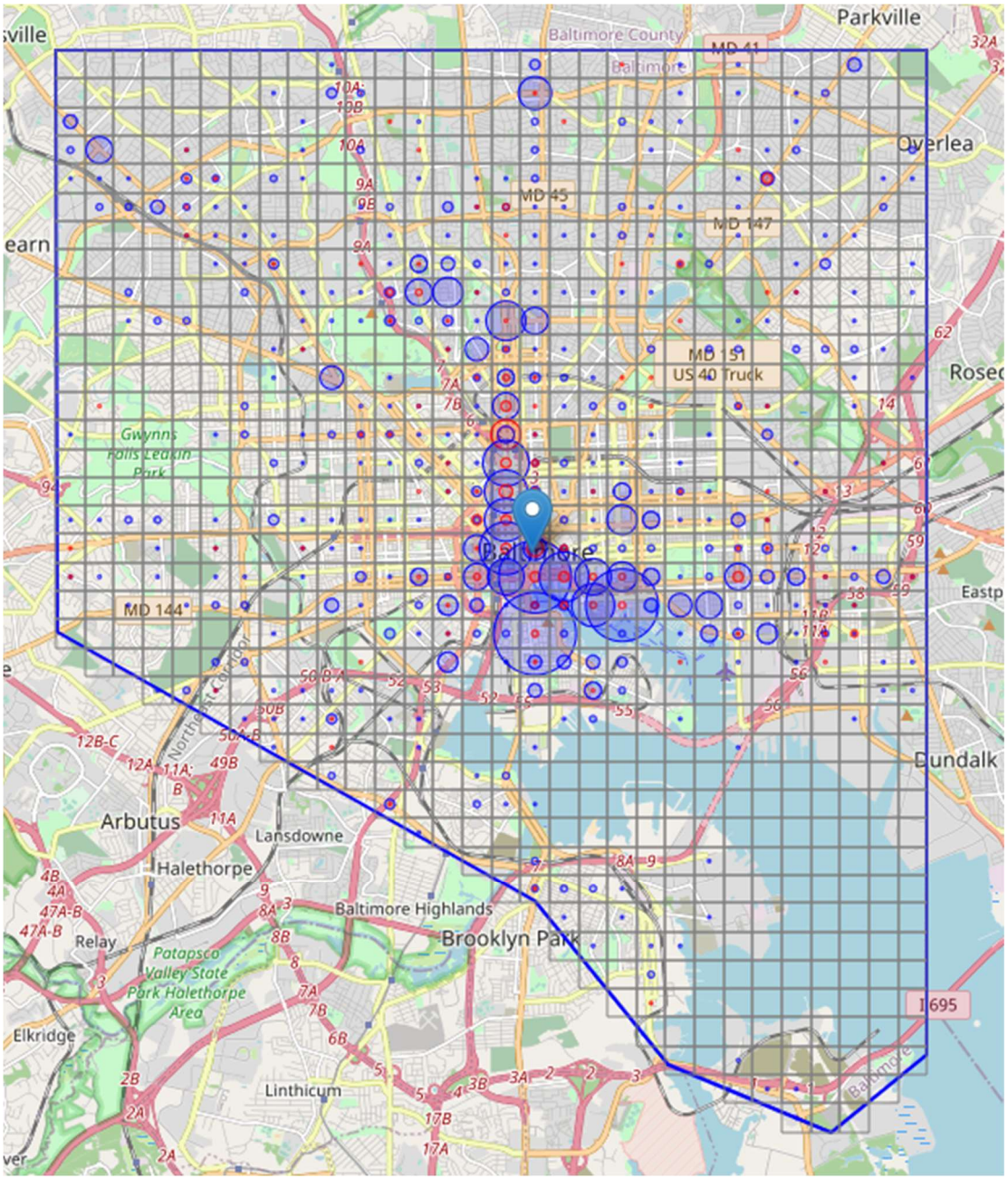


Figure 3 Map showing traffic drivers with proportional circles in traffic drivers in blue and competitors in red.

A more satisfactory depiction is a heatmap with traffic drivers shown in their actual locations in blue and competitors in a yellow-red gradient. Hotter or darker areas are areas with more traffic drivers or competitors.

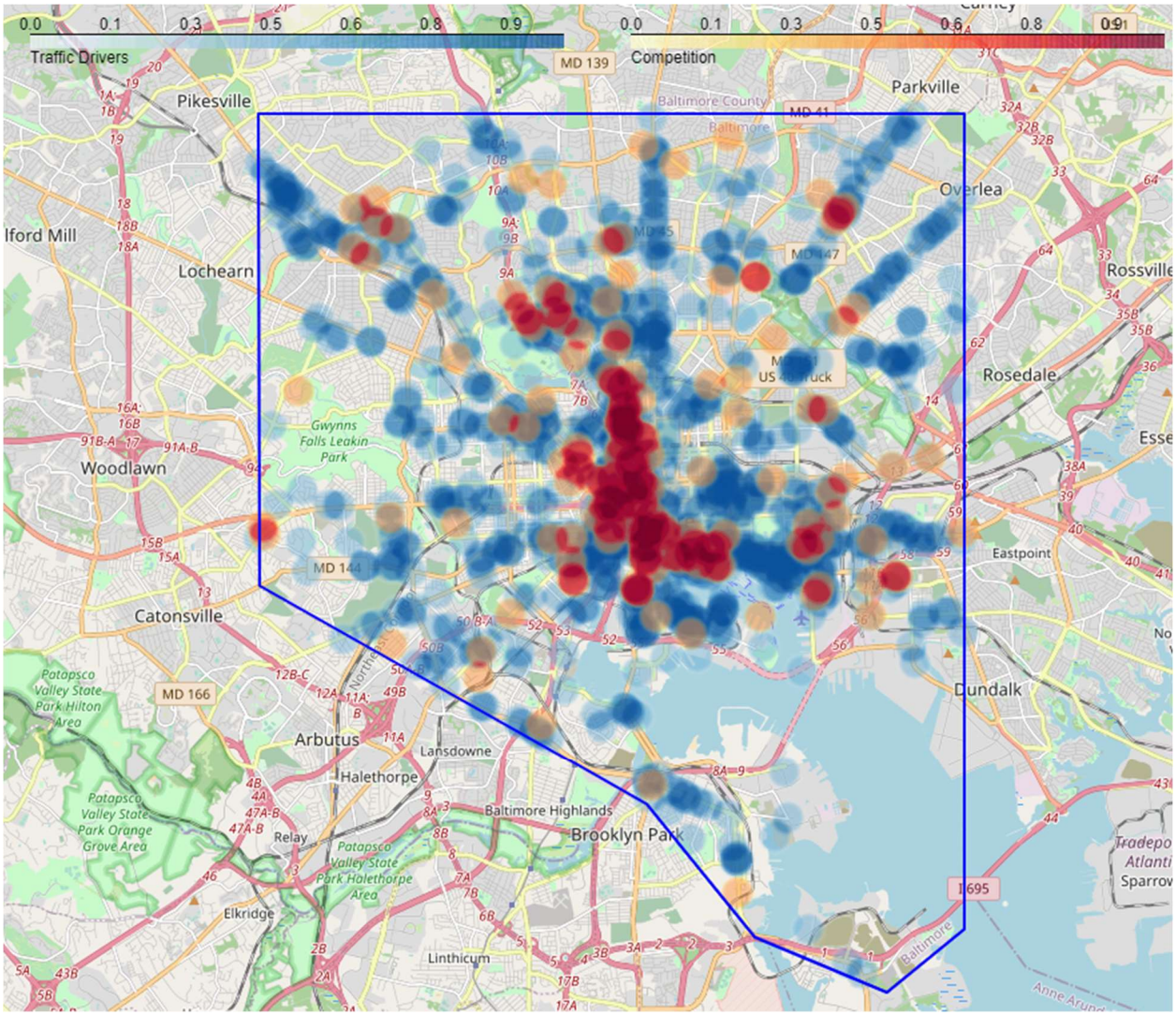


Figure 4 Heatmap of traffic drivers (blue) and competitors (yellow-red).

To quantitatively identify viable candidate locations within the city, we measure the distance from the center of each square to traffic drivers and competition. For our initial attempt to find good locations, we choose parameters of 3 or more traffic drivers within a half kilometer, and zero or one competitor within 1 km. These are shown in green. These parameters can be adjusted later as we further explore the data. We use the Haversine algorithm to determine distances between points defined by latitude and longitude, first converting our digital longitude/latitude coordinates to radians.[7]

We find 370 city 500m x 500m squares with three or more restaurants within .5km, 583 locations with one or zero competitors within 1km, and 159 locations with both conditions met, and draw them below in green.

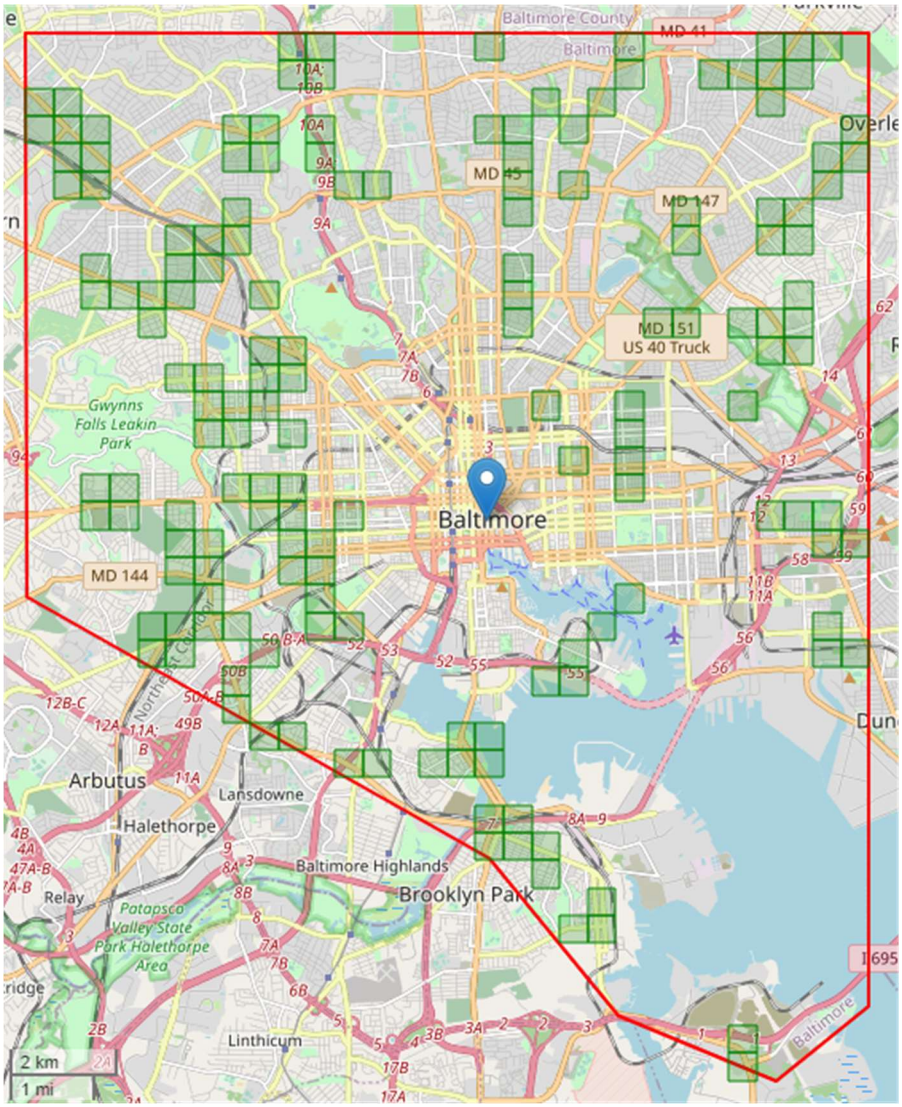
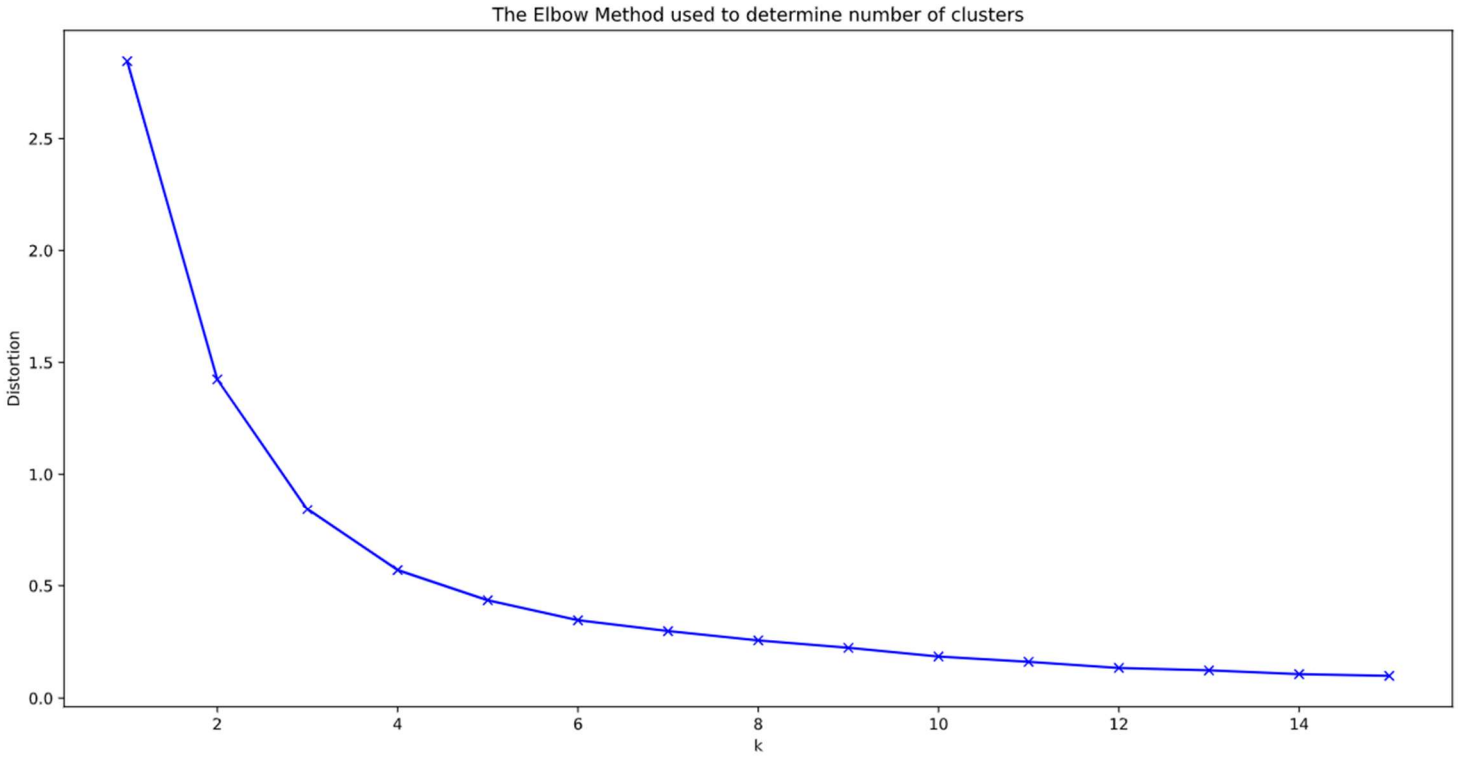


Figure 5 Locales with >= 3 traffic drivers and competitors <= 1.

[7] https://www.geeksforgeeks.org/program-distance-two-points-earth/

To determine the number of clusters in the data, we plot the number of clusters (k) vs. distortion to find the elbow and determine the elbow is at 6 clusters.



The clusters are color coded in Figure 6. The centroid of each cluster is shown in black.

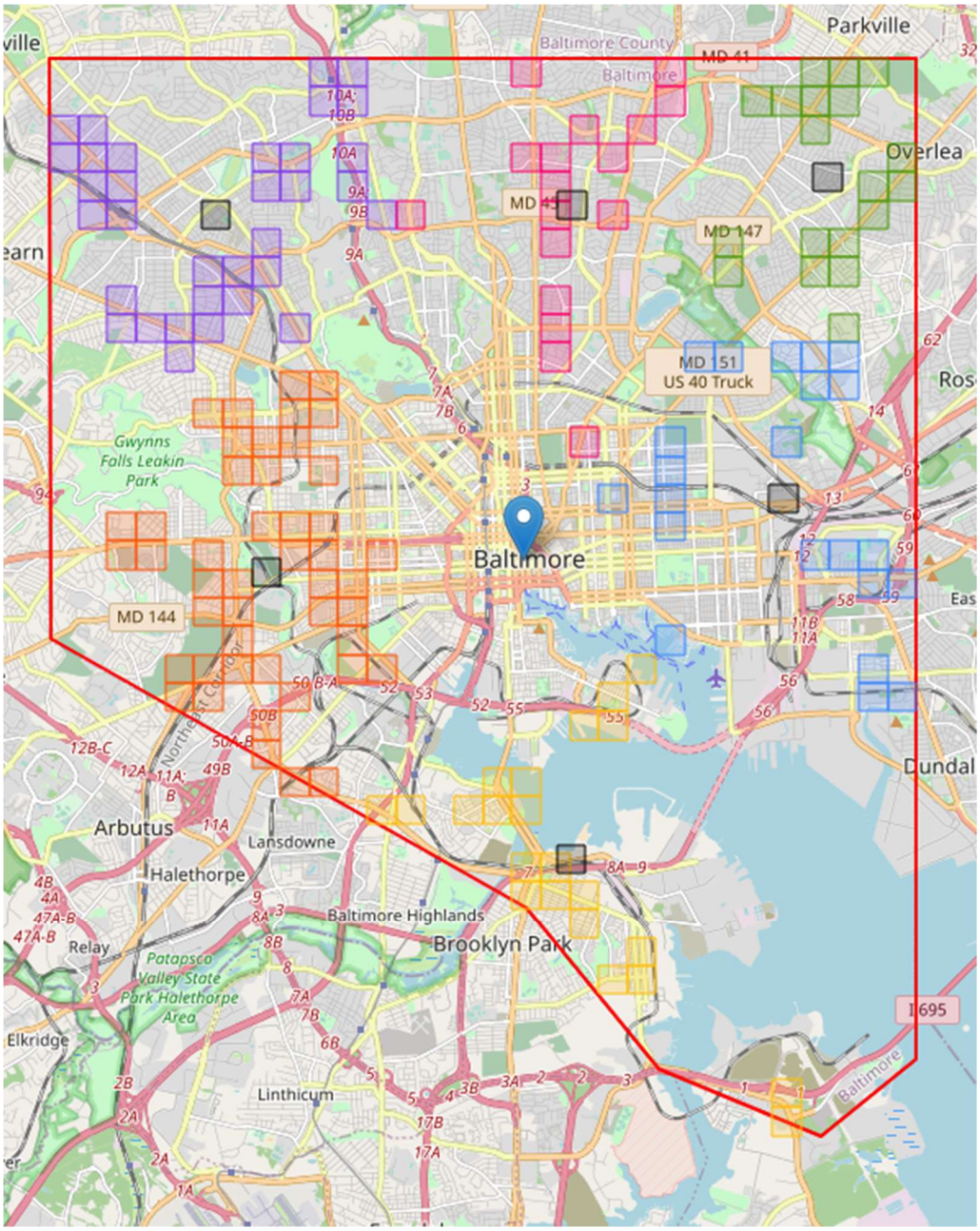


Figure 6 Viable areas shown in color coded clusters

This is unsatisfying. No clear area stands out as being particularly suitable for a music venue, though all match our criteria. There are many restaurants that stand alone and are probably distorting the picture with their noise.

If we plot the traffic driver locations in colored clusters and competitors in grey clusters, we get Figure 7. The small circles showing each venue are drawn with 500-meter diameters. The larger colored circles are the center of the respective clusters. The large grey clusters are the centers of the competitor clusters.

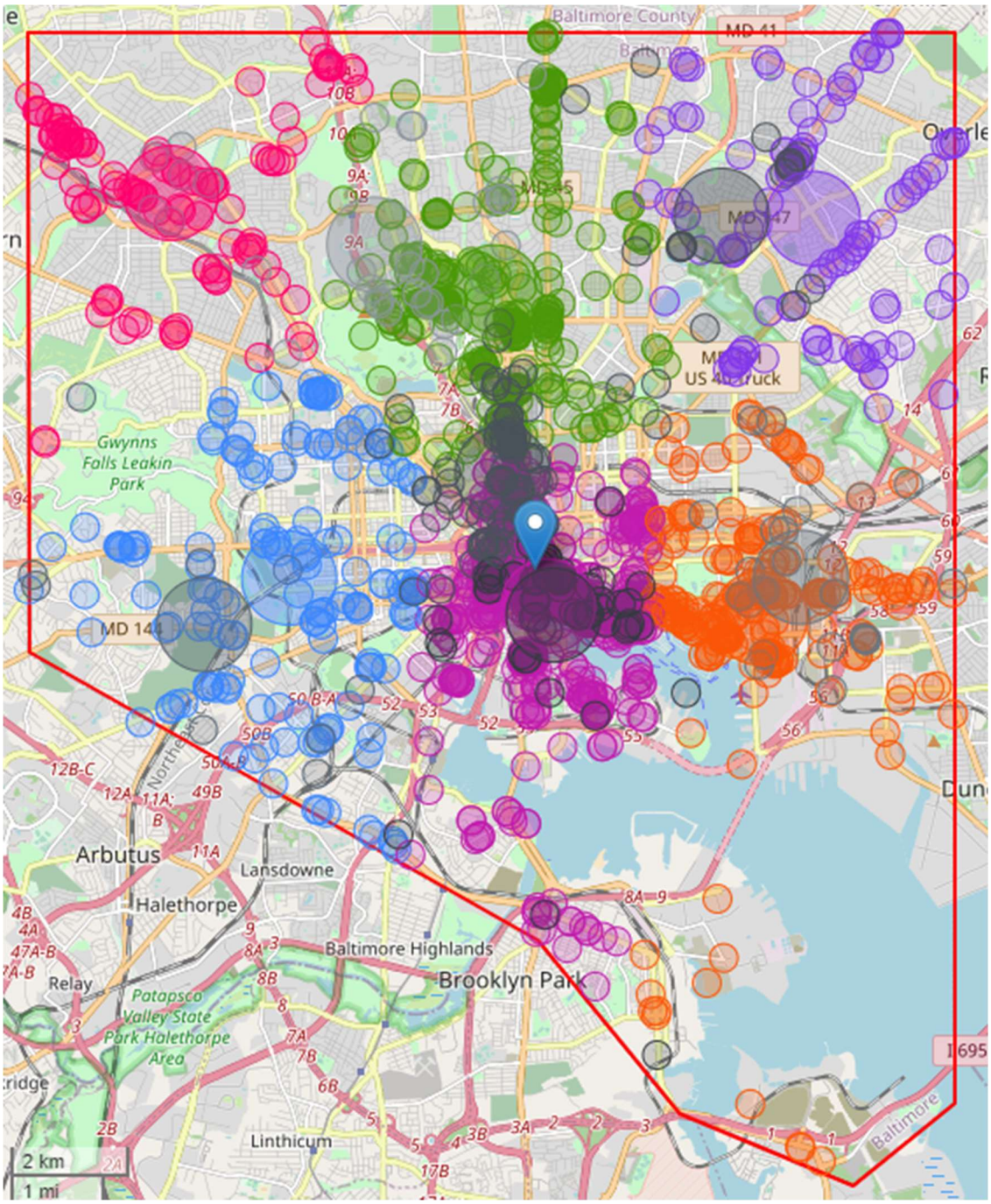


Figure 7 Clustered traffic drivers and competitors.

Figure 7 shows clearly that there are many venues adding noise and distorting our attempts to cluster traffic drivers into nightlife generating clusters.

We can measure the distance between each traffic driver and its other traffic driving compatriots. By drawing a radius around each traffic driving venue, we can count the number of traffic drivers that fall inside. For Figure 8, we used a 250m radius/500m diameter. It shows that 149 venues have no neighbors within a 250m radius, 122 venues have 1 neighbor, etc. (See the Jupyter notebook for the exact figures.) Summing the number of businesses with 0-5 neighbors in a 250m radius, we get 714 businesses, a bit less than half (46%) of the 1539 food venues we initially found in Foursquare's data. Let's remove these from the data set and see if we get a clearer picture of Baltimore City nightlife centers.

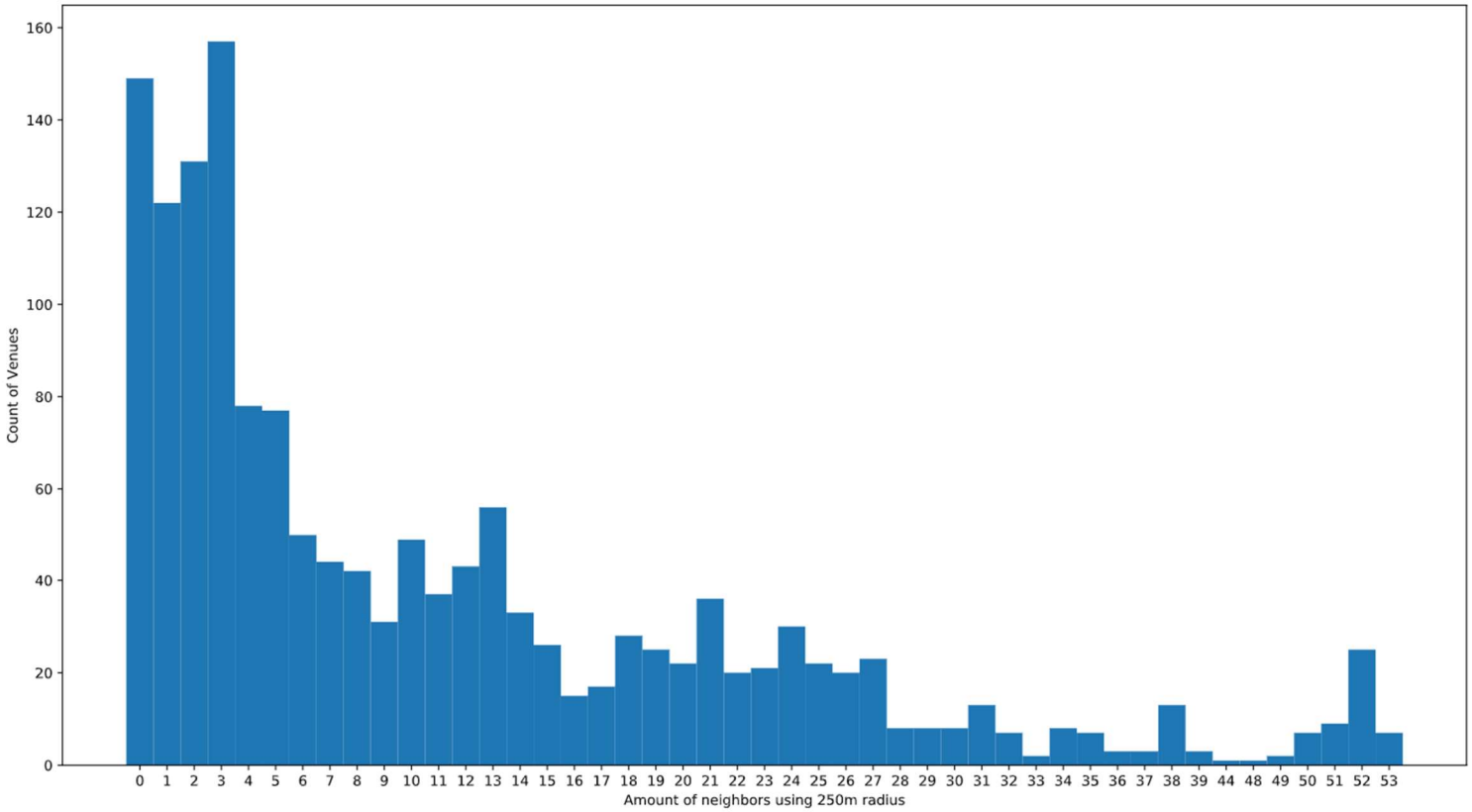


Figure 8 Traffic drivers within 250m radius (500m diameter).

On reflection, we do not need to determine if the competition is clustered together, so we remove the large grey competition centroids from the map. They will not return. We derive the k-means clusters again and redraw the map with the remaining 714 businesses and without the grey competition centroids and get Figure 9.

Again, we have 6 clusters, but their centroids have moved considerably. Comparing Figure 9 with Figure 7 above, we find interesting changes. Notably, the northwest cluster has moved further west, and the northeast cluster has moved northwest, to nearly the northern border of the city.

We observe some interesting anomalies. The purple cluster in Figure 9 clearly shows it has three components: the large set of restaurants where the centroid falls in Fells Point (a well-

established nightlife spot), further north is another purple set of restaurants benefiting from proximity to Johns Hopkins Hospital, and most unusual is the purple grouping of a number of restaurants south of the water along Key Highway. As the crow flies, these may meet our proximity criteria, but no one I know would go to dinner in Fells Point and consider a drive to the other side of the the Inner Harbor to be visiting the same nightlife locus.

Observe two other anomalies: the orange cluster in the northwest corner is near a cluster of restaurants, but somehow has claimed about three restaurants directly west of the city center into its cohort. Similarly, the light blue cluster in the northeastern corner has a centroid that seems to straddle two groups of restaurants with little relationship to each other.

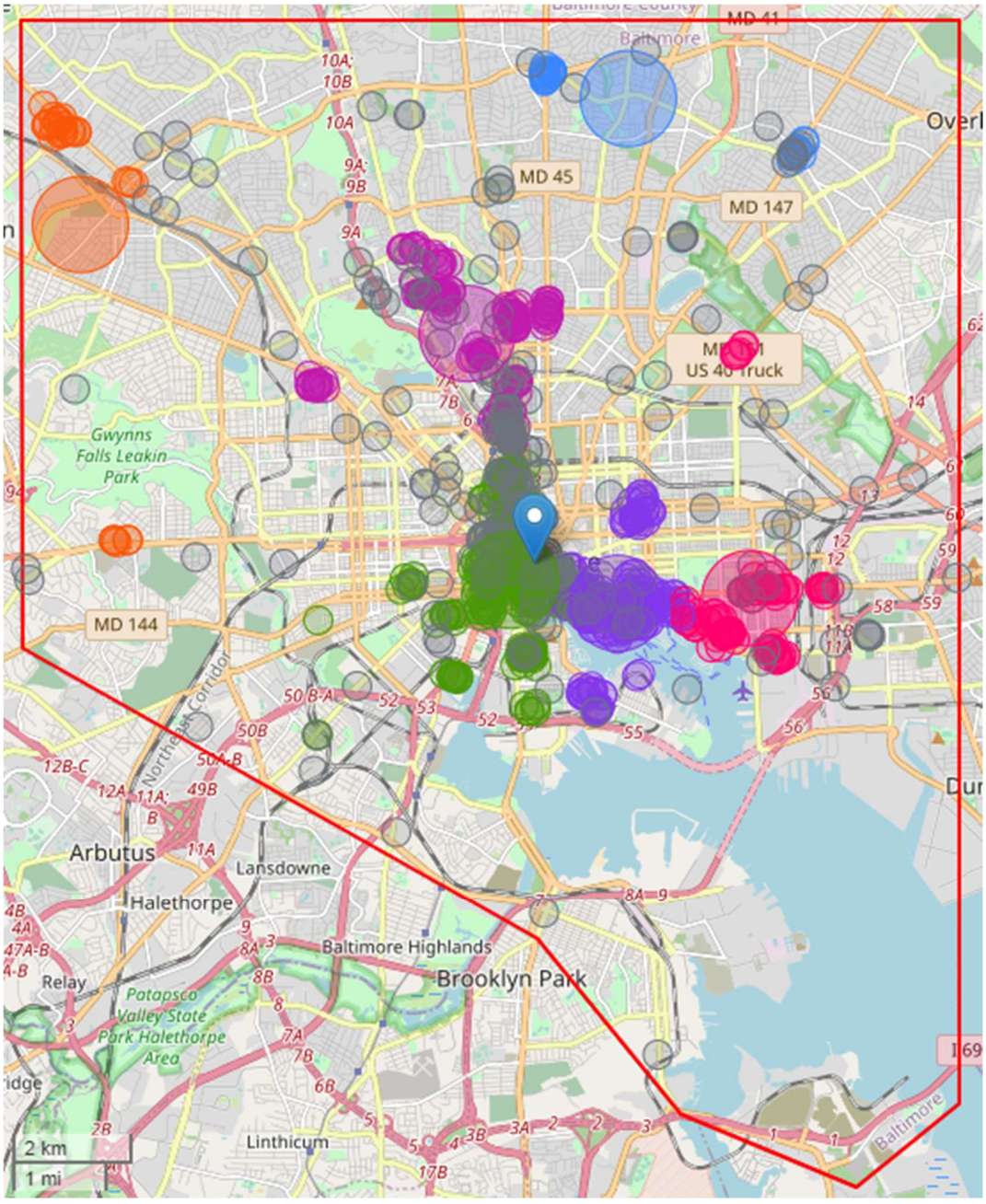


Figure 9 Clustered traffic drivers and competitors with reduced noise

We remain unsatisfied and research a bit more deeply into other clustering techniques. K-means is a relatively good clustering mechanism for unweighted data, but looks like it would be less useful for world spanning latitude/longitude datasets, both because its Euclidean distance algorithm would become nonlinear at higher latitudes as distances between longitudinal values decrease, and because it would fail at the 180 degree boundary. There are workarounds for these issues, but other algorithms may be better.

With some research into the various clustering methods available, we find DBSCAN (density-based spatial clustering of applications with noise). Here's Wikipedia's description: "a density-based clustering non-parametric algorithm: given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away). DBSCAN is one of the most common clustering algorithms and also most cited in scientific literature. In 2014, the algorithm was awarded the test of time award (an award given to algorithms which have received substantial attention in theory and practice) at the leading data mining conference, ACM SIGKDD."[8] This seems to be a good fit for our problem. DBSCAN can handle arbitrary distance functions, which we need for our latitudes and longitudes.

DBSCAN will require two parameters: a distance parameter usually called epsilon and a minimum number of samples needed to determine a cluster. "The combination of min_samples and eps amounts to a choice of density and the clustering only finds clusters at or above that density; if your data has variable density clusters then DBSCAN is either going to miss them, split them up, or lump some of them together depending on your parameter choices."[9] A number of sources describe finding the correct value of epsilon as difficult.

Going one step further, we find that HDBSCAN is a revision of the original DBSCAN algorithm, written by the same authors, with a goal to allow varying density clusters. It supports the Haversine algorithm, has dispensed with the difficult to determine epsilon parameter, and only requires a minimum cluster size variable.

Both DBSCAN and HDBSCAN do not result in a centroid value. They do however return a normalized measure of strength of membership in the cluster, and members with a value of 1.0 are considered "core" members. The Python hdbscan package returns this as a list called "probabilities_. " We redraw our map using the HDBSCAN clustering method in Figure 10 and find that the anomalies noted above have been ameliorated.

[8] https://en.wikipedia.org/wiki/DBSCAN

[9] https://hdbscan.readthedocs.io/en/latest/comparing_clustering_algorithms.html#dbscan

The core members are shown with filled circles, other members are shown as hollow circles. Data points identified by HDBSCAN as noise are shown as black circles. The competitors are gray. HDBSCAN identifies 12 clusters based on a minimum cluster size of 10, which seems to be enough traffic drivers to be considered a nightlife center, though opinions may differ.

The anomalies observed above have been removed. The strange orange western outliers in Figure 9 are now identified as noise, as are the northeastern grouping (light blue in Figure 9, black in Figure 10) that previously was clustered with the northern group (light blue in Figure 9, pink in Figure 10). We are gratified to see that what was a single purple Fells Point /Johns Hopkins Hospital/Key Highway cluster is now identified as 3 distinct clusters in bright green, mint green, and blue.

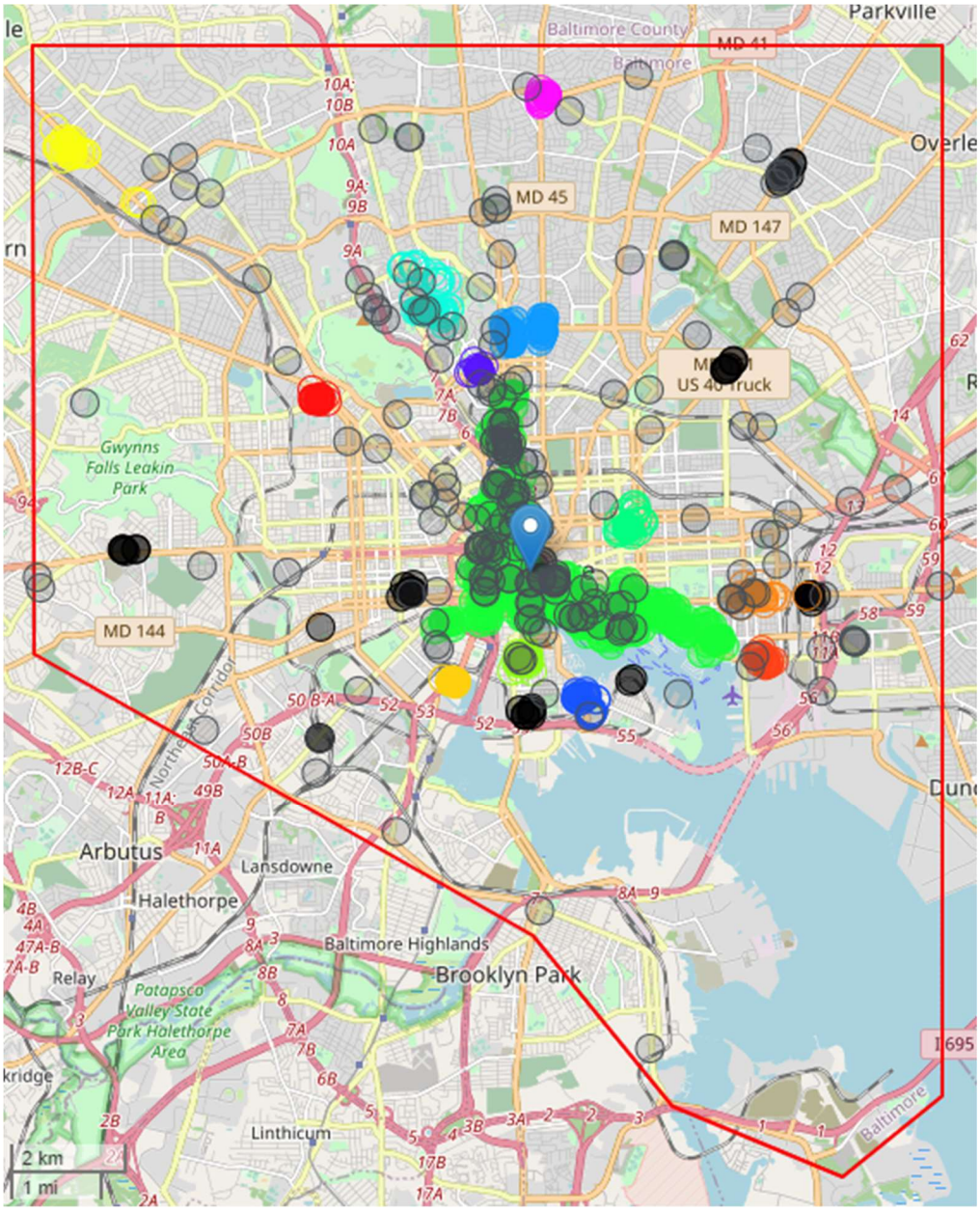


Figure 10 Thirteen clusters generated using HDBMeans and epsilon of .5km.

Now we reexamine our competitors and measure their distance to the center of the clusters. Since there are no defined centroids defined with the HDBSCAN algorithm, we measured the distance from each competitor to each core member of the traffic driving cluster and saved the minimum distance as a column in the dataset. When done, each row in the city squares dataset contains the minimum distance to each cluster (i.e. its nearest core member), and the number of competitors within 1 kilometer.

We select squares from the dataset that have a distance to any cluster less than ½ kilometer and less than 2 competitors within 1 kilometer. The 25 squares revealed are colored according to the cluster they are near, and we are delighted to find several squares that straddle the gaps between nightlife cores, shown as split squares with half of each filled in the respective clusters' colors. In Figure 11, the noisy venues in black have been removed, competitors are grey circles.

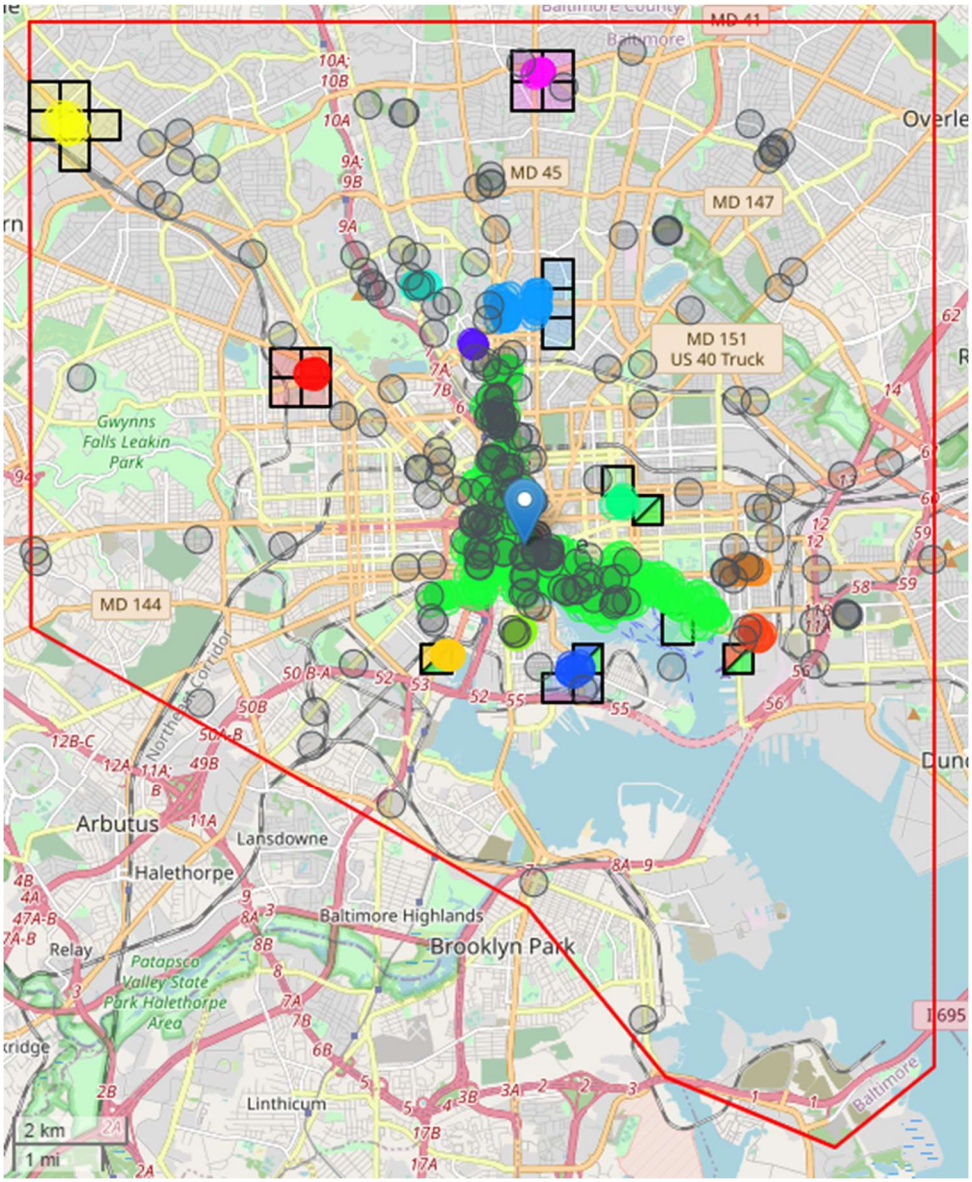


Figure 11 Nightlife centers and candidate square locations for new music venues.

We can reverse geocode the center of each square and get 25 addresses as the starting point of our search:

1) 6854 Reisterstown Rd, Baltimore, MD 21215
2) 6633 Vincent Ln, Baltimore, MD 21215
3) 3737 Clarks Ln, Baltimore, MD 21215
4) 6418 Reisterstown Rd, Baltimore, MD 21215
5) 264 Wabash Ave, Baltimore, MD 21215
6) 3813 W Strathmore Ave, Baltimore, MD 21215
7) 3309 Burleith Ave, Baltimore, MD 21215
8) 2325 Evergreen St, Baltimore, MD 21216
9) 2402 Liberty Heights Ave, Baltimore, MD 21215
10) Frederick Douglass High School, Baltimore, MD 21216
11) 1712 MD-295, Baltimore, MD 21230
12) 422 Howil Terrace, Baltimore, MD 21212
13) 5809 Bellona Ave, Baltimore, MD 21212
14) 811 Dartmouth Rd, Baltimore, MD 21212
15) 613 St Dunstans Rd, Baltimore, MD 21212
16) 720 McKewin Ave, Baltimore, MD 21218
17) 715 E 33rd St, Baltimore, MD 21218
18) 752 E 30th St, Baltimore, MD 21218
19) 1831 Belt St, Baltimore, MD 21230
20) 1449 Key Hwy, Baltimore, MD 21230
21) 1631 Whetstone Way, Baltimore, MD 21230
22) 1720 Ashland Ave, Baltimore, MD 21205
23) 420 N Duncan St, Baltimore, MD 21231
24) 1246 Dockside Cir, Baltimore, MD 21224
25) 1635 S Clinton St, Baltimore, MD 21224

Results

We have achieved our goal of finding candidate locations for possible new music venues near existing nightlife centers and evaluating them with respect to quantity and distance of competition. We began by dividing the city into 500m x 500m squares and finding all traffic drivers and competitors. We evaluated the data using the elbow method to determine that the data contained 6 clusters and clustered traffic drivers using k-means clustering. Each square was than evaluated for its proximity to traffic drivers clusters and scarcity or distance from competing venues. Our initial clustering results shown in Figure 6 didn't produce a clear direction for our location search.

We identified restaurants with few neighbors, as they seemed too isolated to legitimately be considered members of nightlife centers. This left us with about 54% of our original restaurants. We reanalyzed the elbow and found we continued to have six clusters. We mapped the new clusters, and saw the reduced noise, but the anomalies of the clusters produced via k-means clustering were even more noticeable and we remained unsatisfied.

Looking at other clustering methods, we settled on HDBSCAN as the best option. It had fewer requirements than DBSCAN and had the added benefit of being able to identify datapoints it considered noisy. This provided excellent results as shown in Figure 10, with none of the anomalies noted earlier, and gave clear indication of additional noise that could be removed. Using the 13 nightlife centers that were identified, we again evaluated each square for its proximity to traffic drivers and competition and found a set of locations that seem quite sensible when evaluated with the traffic driver and competition data. We even found 4 squares that straddle the space between two identified nightlife clusters.

Discussion

The resulting list of addresses form a starting point for venue searches. There may be many reasons to remove them from consideration and additional research should be done: the neighborhood might not actually have much nighttime dinner traffic, despite the presence of many restaurants, or it may be found to have no parking or troublesome traffic patterns, or the restaurants may not be drawing consumers who also frequent live music venues. If the expectation is that people might park their car and walk to dinner and then to the music venue, there should be safe and well-lit sidewalks.

Our results were especially interesting where they identified locales with no existing music venue, such as the yellow cluster in the northwest corner of the city. This is a neighborhood that has experienced a bit of a rebirth recently, perhaps it has now matured to the point where it is ready for music. We also were pleased to find candidates that straddled two nightlife clusters. Perhaps they could bridge the two areas. These locations may be more valuable if they can draw from two sets of restaurant goers, which may provide justification for high real estate prices, or indicate underpricing and a potential bargain.

It is notable that one of the city squares identified as straddling two clusters, the blue/green one shown under the southcentral blue circle in Figure 11 and on the left edge of Figure 12 does not truly join two nightlife clusters, because the bay falls between it and the green cluster. This is an artifact of our distance measurement, which is performed in straight lines, regardless of terrain or ocean conditions.

Data issues have been observed, but no attempt to remedy them was done for this exercise. For example, several food venues found in the Foursquare data are clearly miscategorized and appear to be food stores, not restaurants. In addition, we see that some food venues are private or school cafeterias. With more time, it might be possible to restrict food venues to publicly available nighttime venues. More data cleansing would be warranted if time allowed.

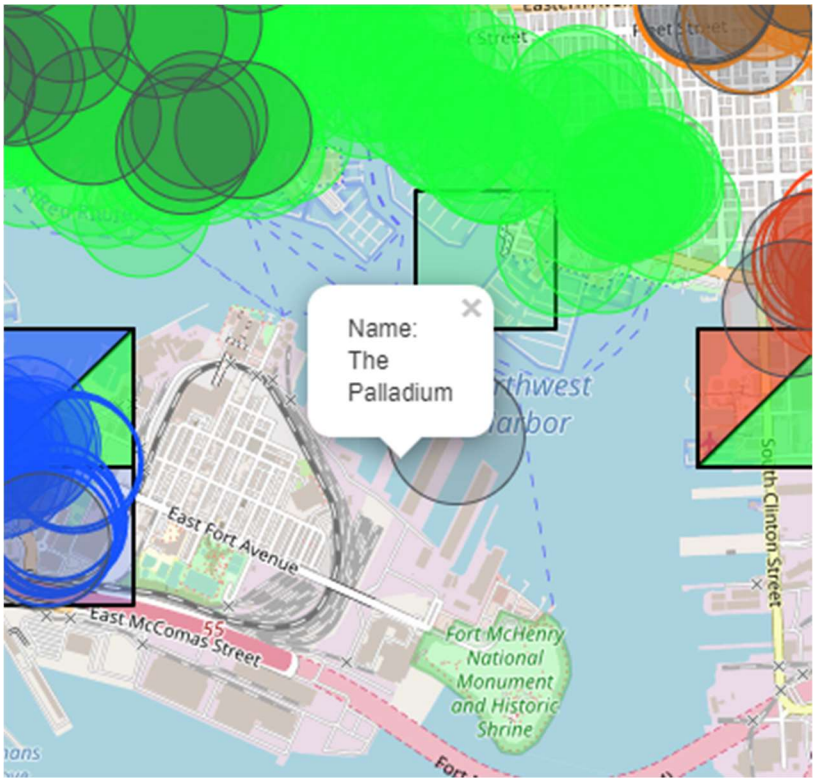


Figure 12 The Palladium doesn't pose much competition to other music venues

The Foursquare data for competitors can also use refinement. Again, high school or college theatres might be removed. Notably, there is a competitor identified as The Palladium shown at the end of the pier near Fort McHenry. I wasn't familiar with this venue and failed to find it in a Google search. After consulting Foursquare I found that it is a nightclub located on the Royal Caribbean International cruise ship Grandeur of the Seas, both a private club and a moveable one.[10] Since we were using a competitor count less than or equal to 2, we were very sensitive to competitors and even a single misidentified competitor would cause us to remove the city square from consideration.

Additional factors to consider might result in adjusting the parameters of our searches and queries. What distance is considered reasonable when parking, walking to dinner, and then walking to hear music? We used one half or one kilometer for most of our queries, or perhaps some people are happy to drive across town. The number of restaurants that form a nightlife center is debatable--we used 10--and the final determination depends on a number of factors, including the population of the city and number of restaurant goers with sufficient disposable income. How many music venues can a neighborhood sustain? It would be easy to change any of these parameters and rerun the analysis. Can Baltimore as a city sustain more music venues? It may already be at its limit. These questions could form the basis for future analysis and might point to a predicted music venue density based on comparative city demographics.

[10] https://foursquare.com/v/the-palladium/55b216a2498e102b5ee6d9dc

Conclusion

Our goal was to find possible locations for a new music venue in the city of Baltimore. We were able to find 25 locations to start our search based on proximity to restaurant clusters and distance from possible competition. Suggestions for cleaning the data were made, which would likely improve the quality of the search. were made. The parameters used to refine our analysis could be debated or adjusted as needed and might reveal additional candidate locations. Ultimately, any location would require further research and should be visited to fully determine its suitability for a new business.