MAPPING PROGRESS IN WASHINGTON DC

Georgetown CCPE Data Science Capstone

DECEMBER 16, 2017

TEAM DATA EXTRACTORS
TONY, JAY HUANG, KEN SHUART & JASON COFFEY

Table of Contents

Abstract

The growth in Washington DC can be described as a microcosm of U.S. urban neighborhood development and gentrification where poor and crime-ridden neighborhoods suddenly become prosperous hubs of hip culture and expensive condos. Is that really the case? Based on a study conducted by Next City in Philadelphia PA (see Reference below), our team would like to apply the same approach to analyze 5 years of trending data in Washington DC and come to conclusions about growth, decline and progress in the district. Furthermore, our team will improve this process to include applying machine learning models that include monthly data and add additional features to implement a more thorough, interactive, and customizable data product that can be openly accessed for various purposes.

The team started by analyzing 5 categories of data: Crime, Mean Household Income, Population, Poverty and Home Sale Prices. We applied the same methodology as the Next City study to take the overall mean of the each category and calculated the percent growth from the first year to last. That process was used to then set category scores for each neighborhood in D.C. and then an overall score for each neighborhood. For our data science implementation, we applied an unsupervised clustering machine learning process on the same data, including adding additional features such as demographic distributions, violent crime, theft crime, rent prices, poverty range, mean income, and education rates.

Hypotheses

Initially, our hypotheses was to predict growth and decline between neighborhoods in Washington D.C. However, due to the random types of the data we were comparing and the process of comparing apples to oranges, we decided to use an unsupervised classification model to show movement and separability between clusters. We hypothesize that this process will better show growth and decline over time than the Next City study. Our instances would be each neighborhood's location and monthly data over time.

Applications

Our team's vision for this data product would allow city planners, businesses and individual uses to dynamically plug in their particular interests (because we understand, every user will have different perspectives about growth) and run our model over the last 5 to 10 years of data to show the trajectory of progress in Washington D.C.

Data Ingestion

For the process of Ingestion our team used several sources to obtain the essential data sets. Data was obtained from 4 different sources:

1. The first was Census.gov. Census.gov provided us with a central repository of data that was really beneficial and was a great starting point for us. It allowed us to come up with a good idea of what information was publicly accessible and therefore provided direction into what our data application would be comprised of. Our Household Income,

Population and Poverty data were all pulled from the American Community Survey and had a range of year 2011 – 2015.

2. Opendata.dc.gov was used to scrape criminal data. Open data is a really cool website if you've never used it because it allows you to discover, analyze and download data in any format you need. We also adopted neighborhood cluster shape files from opendata.dc.gov as our primary grouping mechanism.

3. Our third source was Zillow.com. Zillow is the leading real estate marketplace and has an excellent database to obtain home values throughout Washington D.C.

4. Our final data source was USBoundary.com. US Boundary provided us with latitude and longitude information to group our data into neighborhood shape files.

Below is a brief description of the metrics of five of the most important data sets in determining growth and decline thought out the district:

- Household income & Sale price were a dollar value.
- Population data included categories such as race, age, gender, native born and education
- Crime data encapsulated total crimes and was condensed to a ratio of violent crimes vs theft
- Poverty data is calculated by comparing pre-tax cash income against a threshold.

Data sources were exported into CSV format and scraped using Beautiful soup before they were imported into Postgres. We chose Postgres as our open source database and created a total of 18 tables. We also used Amazon Web Services Relational Database Services in order to set up, operate, and scale our relational database in the cloud.

<u>Data Wrangling</u>

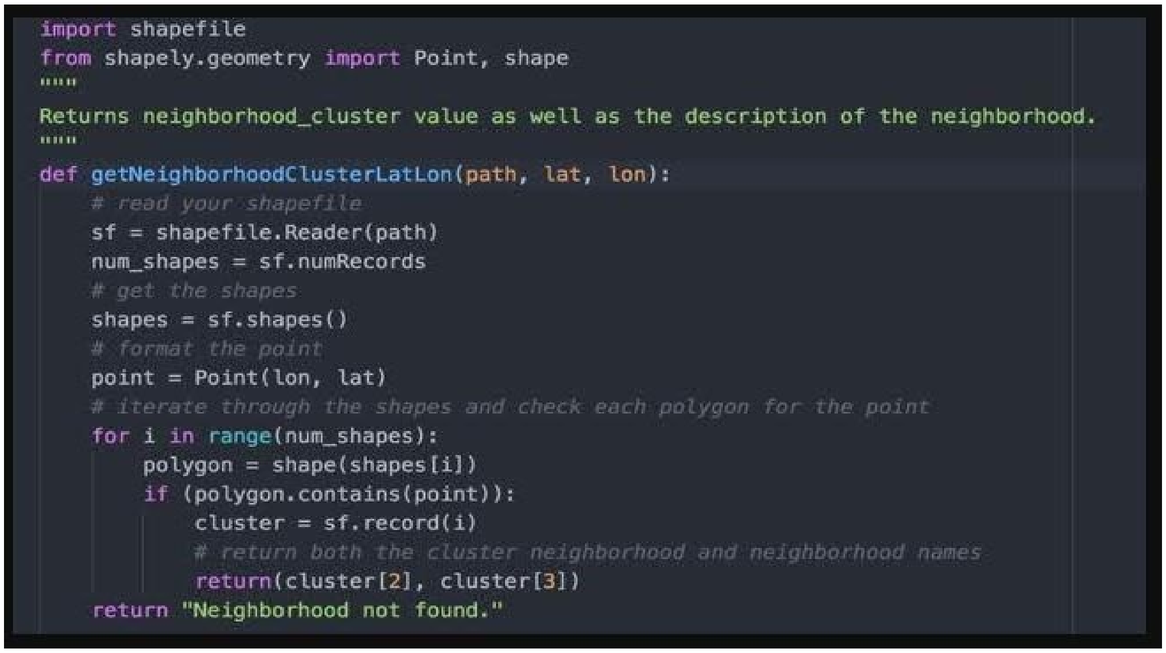
We had 2 key objectives with our wrangling efforts:

- Aggregate Data into 46 Neighborhoods
- Format time period instances into data frame

Neighborhood Aggregation:

Our study was predicated on the analysis of data points by named DC neighborhoods (NBH). We obtained geospatial boundaries of 46 neighborhoods from opendata.dc.gov in the form of a shape file that included labels (neighborhood name) that exist inside these boundaries.
Our population demographic, median income, and some median sales price data originated from Census.gov. It was segmented into 170+ census tracts within the DC city boundaries.
Crime statistics included incident locations with Latitude/Longitude. Zillow home price were identified by a neighborhood label.

These disparate data sources needed to be aggregated and transformed to fit within spatial boundaries as defined by the shape file. To do this we wrote a python function. See Figure 1 below.



```python
import shapefile
from shapely.geometry import Point, shape
"""
Returns neighborhood_cluster value as well as the description of the neighborhood.
"""
def getNeighborhoodClusterLatLon(path, lat, lon):
    # read your shapefile
    sf = shapefile.Reader(path)
    num_shapes = sf.numRecords
    # get the shapes
    shapes = sf.shapes()
    # format the point
    point = Point(lon, lat)
    # iterate through the shapes and check each polygon for the point
    for i in range(num_shapes):
        polygon = shape(shapes[i])
        if (polygon.contains(point)):
            cluster = sf.record(i)
            # return both the cluster neighborhood and neighborhood names
            return(cluster[2], cluster[3])
    return "Neighborhood not found."
```

Figure 1 Returns specific neighborhood cluster based on Lat/Lon input

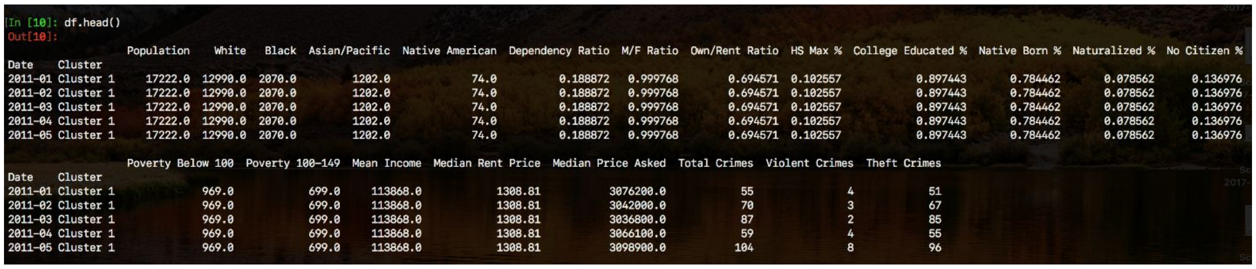
The script above will accept Lat/Lon coordinates and determine when a point is inside a given geospatial shape. It purpose is to return the neighborhood in which that point falls (opendata.dc.gov used the term ‘cluster’ as a way to label an area that contains more than one named neighborhood.; example ‘Cluster 35’ = Fairfax Village; Naylor Gardens; Hillcrest; Summit Park).

The script enabled us to achieve the transformation of census tracts and incident locations into our 46 neighborhoods. Zillow data included neighborhood names. This was a simple name matching exercise. Transforming 170+ census tracts into 46 NBH meant we had more than one census tract inside a NBH boundary. To compensate we used the mean value(s) among the group of tracts that mapped to a single NBH. Mean was used because the data was already a median value.

Data Frame Aggregation:

For each NBH, we had 60 instances (5 years X 12 months). Our total number of instances was 46 NBH x 60 = 2760. We dropped 2 NBH due to sparse data which brought down our total number to 2640. The NBH we dropped from our study were (National Mall/Potomac River) and (Arboretum/Anacostia River). Both areas have a small permanent population.
Our feature data consisted of home sales price, median income, demographic ratio, and frequency of occurrence(s). We were able to wrangle nearly 40 features but ultimately settled

on 21 that were distinct and independent in nature. The resulting data frame is shown in Figure 2.

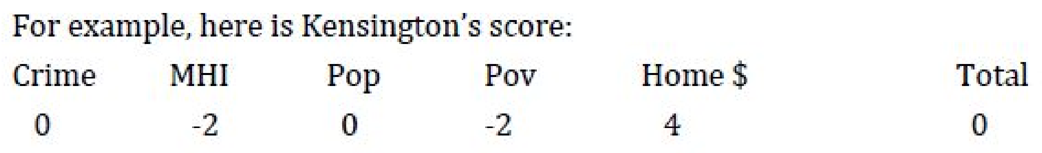


```
In [10]: df.head()
Out[10]:
                    Population    White   Black  Asian/Pacific  Native American  Dependency Ratio  M/F Ratio  Own/Rent Ratio  HS Max %  College Educated %  Native Born %  Naturalized %  No Citizen %
Date    Cluster
2011-01 Cluster 1      17222.0  12990.0  2070.0         1202.0             74.0          0.188872   0.999768        0.694571  0.102557            0.897443       0.784462       0.078562      0.136976
2011-02 Cluster 1      17222.0  12990.0  2070.0         1202.0             74.0          0.188872   0.999768        0.694571  0.102557            0.897443       0.784462       0.078562      0.136976
2011-03 Cluster 1      17222.0  12990.0  2070.0         1202.0             74.0          0.188872   0.999768        0.694571  0.102557            0.897443       0.784462       0.078562      0.136976
2011-04 Cluster 1      17222.0  12990.0  2070.0         1202.0             74.0          0.188872   0.999768        0.694571  0.102557            0.897443       0.784462       0.078562      0.136976
2011-05 Cluster 1      17222.0  12990.0  2070.0         1202.0             74.0          0.188872   0.999768        0.694571  0.102557            0.897443       0.784462       0.078562      0.136976

                    Poverty Below 100  Poverty 100-149  Mean Income  Median Rent Price  Median Price Asked  Total Crimes  Violent Crimes  Theft Crimes
Date    Cluster
2011-01 Cluster 1               969.0            699.0     113868.0            1308.81           3076200.0            55               4            51
2011-02 Cluster 1               969.0            699.0     113868.0            1308.81           3042000.0            70               3            67
2011-03 Cluster 1               969.0            699.0     113868.0            1308.81           3036800.0            87               2            85
2011-04 Cluster 1               969.0            699.0     113868.0            1308.81           3066100.0            59               4            55
2011-05 Cluster 1               969.0            699.0     113868.0            1308.81           3098900.0           104               8            96
```

Figure 2 Final Data Frame

Comparison Calculations:

The goal of our study was to make a comparison between a previous statistical analysis of NBH with 5 features against our machine learning analysis with 21 features.
The statistical study computed the median for each feature and measured its distance from a city wide median over the same time period. Then assigned a numeric value to each feature based on a -4 to 4 range. The sum of each feature score generated a total value which was the score assigned to the NBH. See Figure 3.



For example, here is Kensington's score:

Crime	MHI	Pop	Pov	Home $	Total
0	-2	0	-2	4	0

Figure 3 Next City Scoring

We, therefore, had to make a similar calculation on our data so that we could compare this scoring methodology to our results from machine learning. With these scores computed, a side by side comparison could be made.

<u>Computation and Analyses</u>

We wanted to briefly touch on the computation and analysis of our project by highlighting the growth of our 5 most important categories.

•Crime growth from 2011-2015 was 10% and to give you an idea of the metrics, there were almost 3,000 violent crimes in DC in 2015.
•House Price growth from 2011-2015 to was 4.3%. The median Washington D.C. home value in 2015 was right around $500,000.
•Mean household income growth from 2011-2015 to was 13%. In 2011 the DC mean income average was 89,000 and in 2015 grew to $101,000.
•Population 8.81%. Washington DC population is currently just shy of 700,000.

•Poverty 8.26%. Poverty is measured by comparing a pre-tax income to a city wide threshold. See Figure 4.

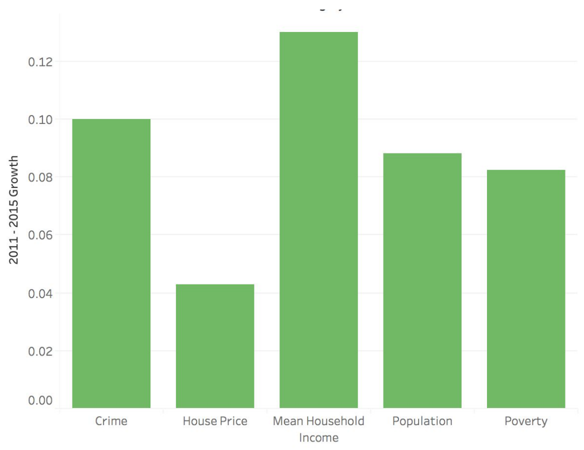


Figure 4: Summary Statistics

Modeling and Application

The machine learning portion of our Capstone project was organized into three distinct steps. First, we clustered our data using unsupervised machine learning. Since our data was unlabeled, we used unsupervised clustering algorithms to group neighborhood instances with similar features into clusters. The first step of our machine learning pipeline was standardizing our data using StandardScaler from scikit-learn. See Figure 5.

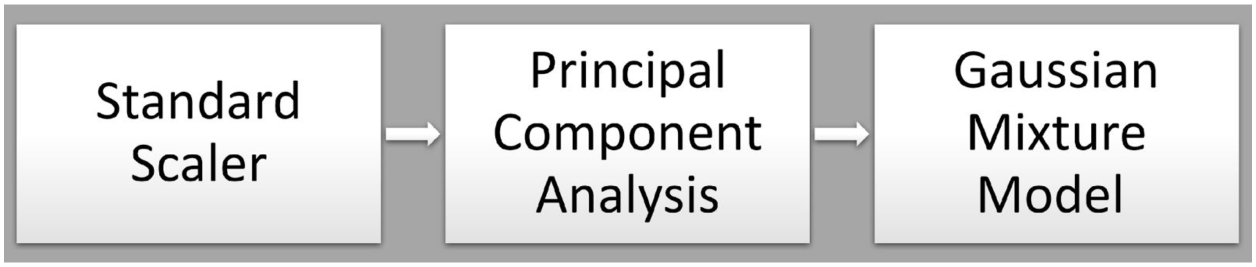


Figure 5: Machine Learning Pipeline

We chose to standardize our data because our features consisted of different units and had different orders of magnitude. We then reduced the standardized data using Principal Component Analysis. We chose to use PCA from scikit-learn and reduce the dimensions of our feature space to two dimensions in order to reduce complexity and prevent overfitting of our model. Initially, we reduced our data to five components, which had a cumulative explained variance of 90%, but we found that we were not getting robust movement of instances between clusters and were unable to adequately measure neighborhood growth. We decided on reducing our data to two components in order to get robust movement of instances between clusters. We were also comfortable with retaining 75% cumulative explained variance of the original data set. See Figure 6.

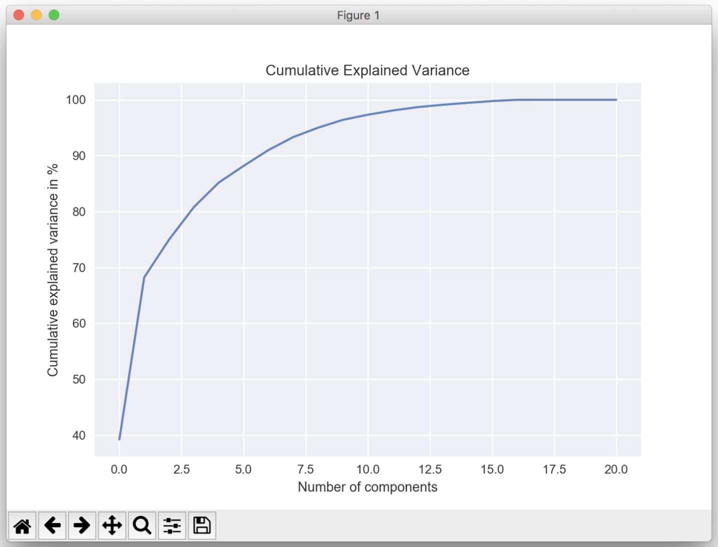


Figure 6: Cumulative Explained Variance

Looking at the coefficient heat map of the principal components, we see that the heat map exhibits correlations that are in line with what we would expect. For example, the first component has a correlation between the frequency of people with a high school diploma as their highest form of education and the frequency of people under the poverty line. The second component shows a correlation between population and total crimes. See Figure 7

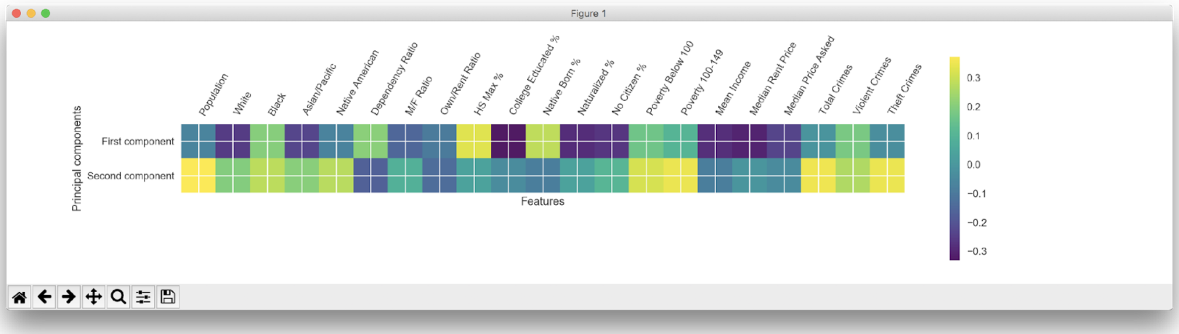


Figure 7: Heat Map of First and Second Principal Components

In order to determine the number of clusters, we looked at the elbow method graph and saw that there was a general elbow at around 20 to 30 clusters before gradually declining. We needed to choose a number of clusters that had a high silhouette score and at the same time allowed for movement of instances between clusters. Too few clusters and the instances would be stagnant and never change clusters. Too many clusters and there would be too much movement, leading to growth results that do not make sense. We decide to use 30 clusters because it had a fairly high silhouette score and allowed for robust movement of instances between clusters. See Figure 8.

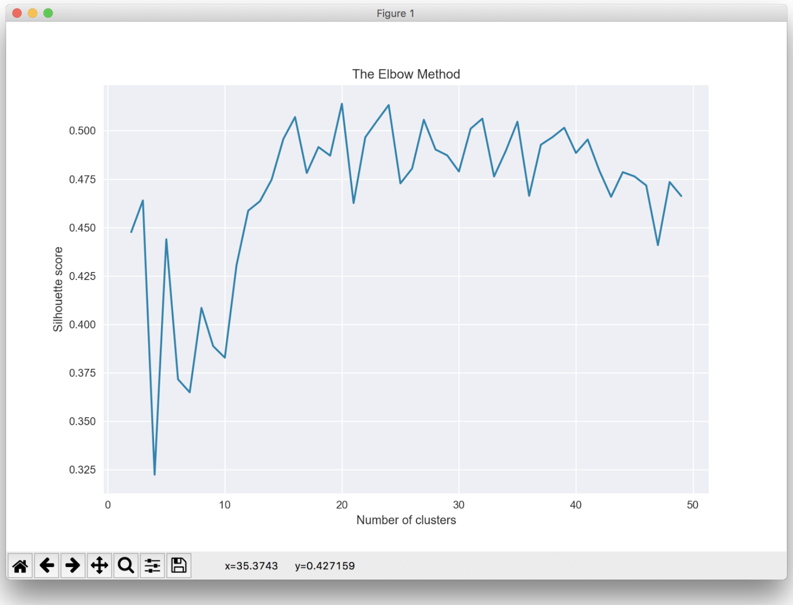


Figure 8: Elbow Method

Finally, we clustered our data using the Gaussian Mixture Model algorithm from scikit-learn. We first eliminated the clustering algorithms that did not allow us to specify the numbers of clusters as a parameter. We decided on GMM over other algorithms such as K-Means because GMM provides flexibility in covariance and produces non-convex clusters, unlike K-Means which only produces spherical clusters. GMM also will not bias the cluster sizes to have specific structures that might not apply to our data.

Looking at the silhouette plot, we see that all of the clusters contain instances that have silhouette scores greater than the mean, and some of the clusters have a majority of instances with silhouette scores greater than the mean, resulting in acceptable cohesion and separability of clusters. Looking at the visualization of the clustered data, we see that clusters located around the corners of the graph have good cohesion and separability, while the clusters in the center of the graph tend to have less separability. The clusters in close proximity with each other are where neighborhood growth is taking place. See Figure 9.

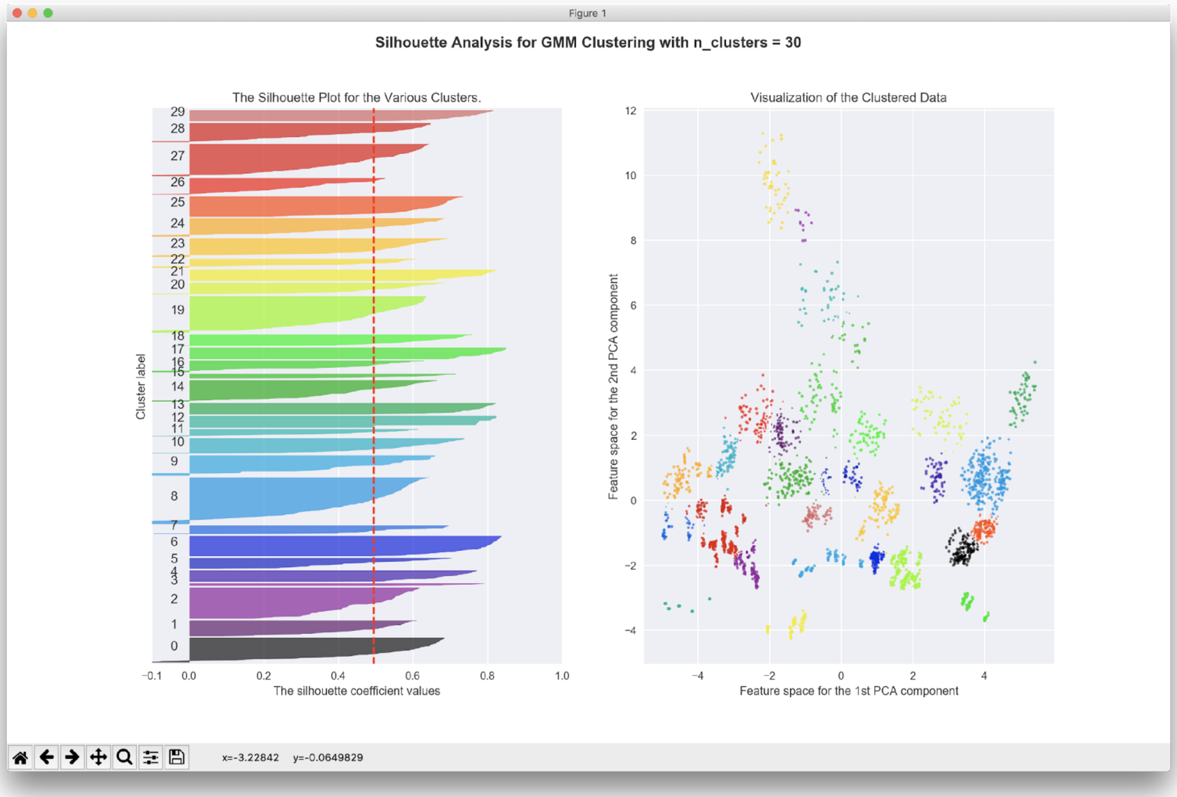
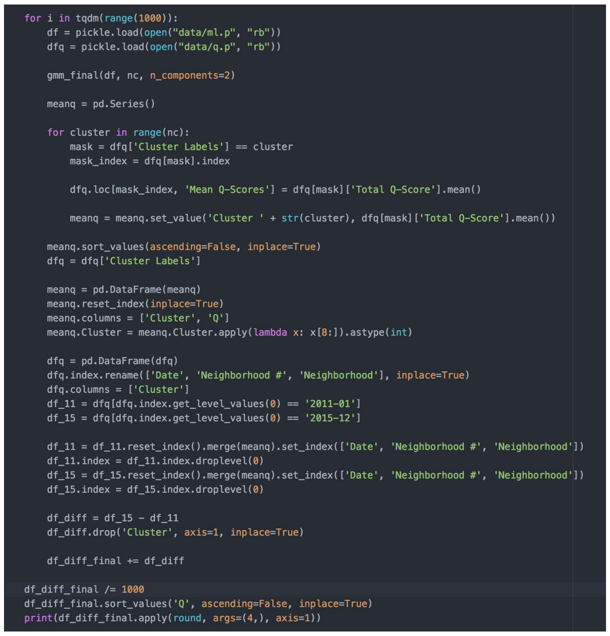


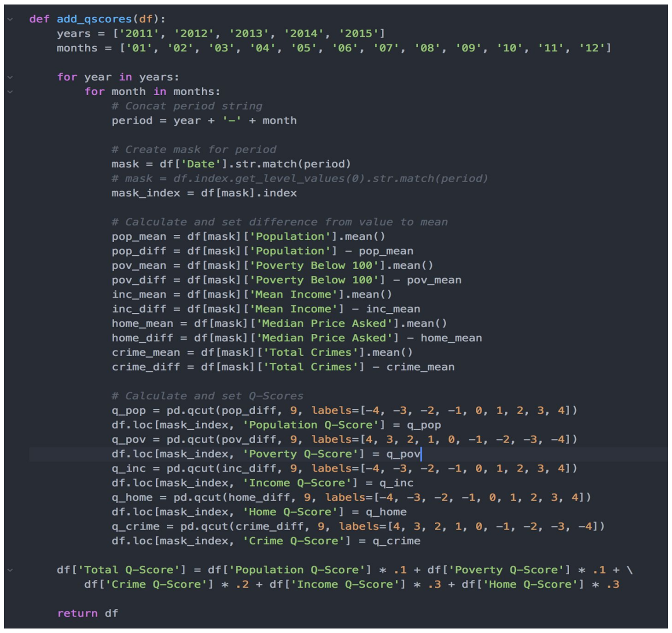
Figure 9: Silhouette Plot and Visualization of Clustered Data

After clustering our data, we then labeled the clusters according to socioeconomic characteristics of the neighborhood instances inside a given cluster. This was accomplished by determining, for every year/month period, the distance between a given neighborhood's socioeconomic metric (population, poverty, income, real estate value, and crime) from the citywide DC mean of the socioeconomic metric. A weighted average representing the total socioeconomic score of a neighborhood instance was then calculated using the values representing the five main socioeconomic categories. Finally, a cluster was labeled by taking the mean of the instances' total socioeconomic scores that are inside the cluster.
See Figures 10 and 11.



```python
for i in tqdm(range(1000)):
    df = pickle.load(open("data/ml.p", "rb"))
    dfq = pickle.load(open("data/q.p", "rb"))

    gmm_final(df, nc, n_components=2)

    meanq = pd.Series()

    for cluster in range(nc):
        mask = dfq['Cluster Labels'] == cluster
        mask_index = dfq[mask].index

        dfq.loc[mask_index, 'Mean Q-Scores'] = dfq[mask]['Total Q-Score'].mean()

        meanq = meanq.set_value('Cluster ' + str(cluster), dfq[mask]['Total Q-Score'].mean())

    meanq.sort_values(ascending=False, inplace=True)
    dfq = dfq['Cluster Labels']

    meanq = pd.DataFrame(meanq)
    meanq.reset_index(inplace=True)
    meanq.columns = ['Cluster', 'Q']
    meanq.Cluster = meanq.Cluster.apply(lambda x: x[8:]).astype(int)

    dfq = pd.DataFrame(dfq)
    dfq.index.rename(['Date', 'Neighborhood #', 'Neighborhood'], inplace=True)
    dfq.columns = ['Cluster']
    df_11 = dfq[dfq.index.get_level_values(0) == '2011-01']
    df_15 = dfq[dfq.index.get_level_values(0) == '2015-12']

    df_11 = df_11.reset_index().merge(meanq).set_index(['Date', 'Neighborhood #', 'Neighborhood'])
    df_11.index = df_11.index.droplevel(0)
    df_15 = df_15.reset_index().merge(meanq).set_index(['Date', 'Neighborhood #', 'Neighborhood'])
    df_15.index = df_15.index.droplevel(0)

    df_diff = df_15 - df_11
    df_diff.drop('Cluster', axis=1, inplace=True)

    df_diff_final += df_diff

df_diff_final /= 1000
df_diff_final.sort_values('Q', ascending=False, inplace=True)
print(df_diff_final.apply(round, args=(4,), axis=1))
```

Figure 10: Calculating Neighborhood Growth

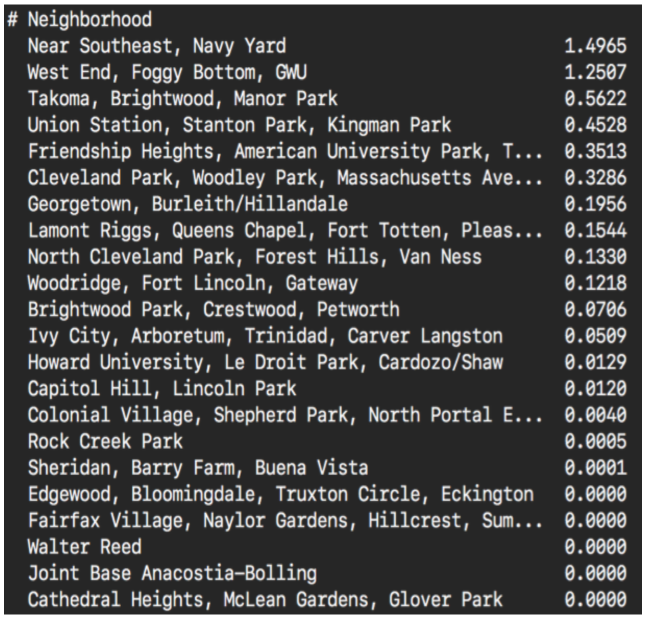


```python
def add_qscores(df):
    years = ['2011', '2012', '2013', '2014', '2015']
    months = ['01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12']

    for year in years:
        for month in months:
            # Concat period string
            period = year + '-' + month

            # Create mask for period
            mask = df['Date'].str.match(period)
            # mask = df.index.get_level_values(0).str.match(period)
            mask_index = df[mask].index

            # Calculate and set difference from value to mean
            pop_mean = df[mask]['Population'].mean()
            pop_diff = df[mask]['Population'] - pop_mean
            pov_mean = df[mask]['Poverty Below 100'].mean()
            pov_diff = df[mask]['Poverty Below 100'] - pov_mean
            inc_mean = df[mask]['Mean Income'].mean()
            inc_diff = df[mask]['Mean Income'] - inc_mean
            home_mean = df[mask]['Median Price Asked'].mean()
            home_diff = df[mask]['Median Price Asked'] - home_mean
            crime_mean = df[mask]['Total Crimes'].mean()
            crime_diff = df[mask]['Total Crimes'] - crime_mean

            # Calculate and set Q-Scores
            q_pop = pd.qcut(pop_diff, 9, labels=[-4, -3, -2, -1, 0, 1, 2, 3, 4])
            df.loc[mask_index, 'Population Q-Score'] = q_pop
            q_pov = pd.qcut(pov_diff, 9, labels=[4, 3, 2, 1, 0, -1, -2, -3, -4])
            df.loc[mask_index, 'Poverty Q-Score'] = q_pov
            q_inc = pd.qcut(inc_diff, 9, labels=[-4, -3, -2, -1, 0, 1, 2, 3, 4])
            df.loc[mask_index, 'Income Q-Score'] = q_inc
            q_home = pd.qcut(home_diff, 9, labels=[-4, -3, -2, -1, 0, 1, 2, 3, 4])
            df.loc[mask_index, 'Home Q-Score'] = q_home
            q_crime = pd.qcut(crime_diff, 9, labels=[4, 3, 2, 1, 0, -1, -2, -3, -4])
            df.loc[mask_index, 'Crime Q-Score'] = q_crime

    df['Total Q-Score'] = df['Population Q-Score'] * .1 + df['Poverty Q-Score'] * .1 + \
        df['Crime Q-Score'] * .2 + df['Income Q-Score'] * .3 + df['Home Q-Score'] * .3

    return df
```

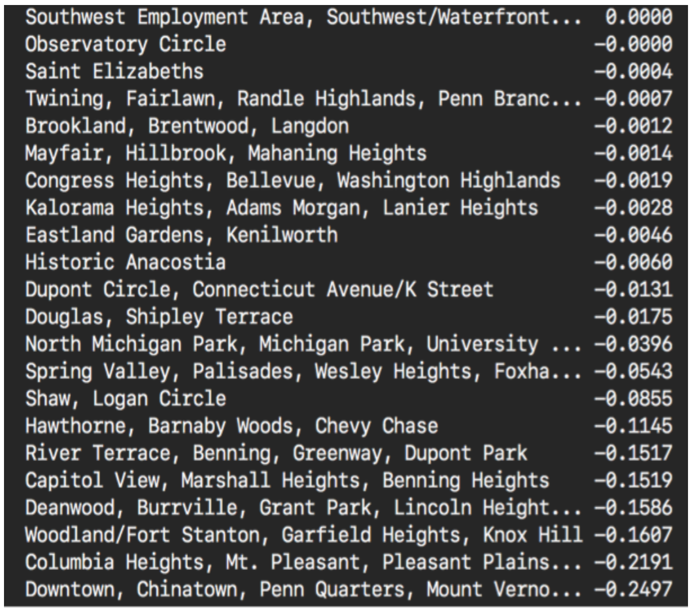
Figure 11: Calculating Scores for Instances

Finally, we measured the socioeconomic growth of a neighborhood over a period of time using our labeled clusters. Specifically, this was accomplished by measuring the movement of a neighborhood between the labeled clusters from January 2011 to December 2015. This resulted in a DataFrame with 44 instances, one instance for each neighborhood, and a score that represented the socioeconomic growth of the neighborhood from 2011 to 2015.

Because GMM clustering does not converge on an optimal solution, we iterated the pipeline a thousand times and took the average of the iterations to produce our final results. We see that some of the results are in line with our expectations: Navy Yard, an up and coming neighborhood that has experienced significant growth over the past years, has the highest score. See Figure 12.



```
# Neighborhood
  Near Southeast, Navy Yard                                 1.4965
  West End, Foggy Bottom, GWU                               1.2507
  Takoma, Brightwood, Manor Park                            0.5622
  Union Station, Stanton Park, Kingman Park                 0.4528
  Friendship Heights, American University Park, T...        0.3513
  Cleveland Park, Woodley Park, Massachusetts Ave...        0.3286
  Georgetown, Burleith/Hillandale                           0.1956
  Lamont Riggs, Queens Chapel, Fort Totten, Pleas...        0.1544
  North Cleveland Park, Forest Hills, Van Ness              0.1330
  Woodridge, Fort Lincoln, Gateway                          0.1218
  Brightwood Park, Crestwood, Petworth                      0.0706
  Ivy City, Arboretum, Trinidad, Carver Langston            0.0509
  Howard University, Le Droit Park, Cardozo/Shaw            0.0129
  Capitol Hill, Lincoln Park                                0.0120
  Colonial Village, Shepherd Park, North Portal E...        0.0040
  Rock Creek Park                                           0.0005
  Sheridan, Barry Farm, Buena Vista                         0.0001
  Edgewood, Bloomingdale, Truxton Circle, Eckington         0.0000
  Fairfax Village, Naylor Gardens, Hillcrest, Sum...        0.0000
  Walter Reed                                               0.0000
  Joint Base Anacostia-Bolling                              0.0000
  Cathedral Heights, McLean Gardens, Glover Park            0.0000
```



```
Southwest Employment Area, Southwest/Waterfront...   0.0000
Observatory Circle                                  -0.0000
Saint Elizabeths                                    -0.0004
Twining, Fairlawn, Randle Highlands, Penn Branc...  -0.0007
Brookland, Brentwood, Langdon                       -0.0012
Mayfair, Hillbrook, Mahaning Heights                -0.0014
Congress Heights, Bellevue, Washington Highlands    -0.0019
Kalorama Heights, Adams Morgan, Lanier Heights      -0.0028
Eastland Gardens, Kenilworth                        -0.0046
Historic Anacostia                                  -0.0060
Dupont Circle, Connecticut Avenue/K Street          -0.0131
Douglas, Shipley Terrace                            -0.0175
North Michigan Park, Michigan Park, University ...  -0.0396
Spring Valley, Palisades, Wesley Heights, Foxha...  -0.0543
Shaw, Logan Circle                                  -0.0855
Hawthorne, Barnaby Woods, Chevy Chase               -0.1145
River Terrace, Benning, Greenway, Dupont Park       -0.1517
Capitol View, Marshall Heights, Benning Heights     -0.1519
Deanwood, Burrville, Grant Park, Lincoln Height...  -0.1586
Woodland/Fort Stanton, Garfield Heights, Knox Hill  -0.1607
Columbia Heights, Mt. Pleasant, Pleasant Plains...  -0.2191
Downtown, Chinatown, Penn Quarters, Mount Verno...  -0.2497
```

Figure 12: Measure of Growth of Neighborhoods from 2011 to 2015

__**Reporting & Visualization**__

After running our data through both models, side-by-side comparison weren't enough to conclude results. Viewing the results through ARCGIS dramatically improved our interpretations of the data and helped explain much of the nuances between each map. The map hot spots below show advancing neighborhood and the light blue shows regression.

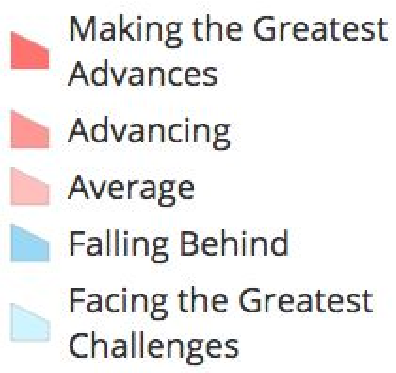


Figure 13 Map Legend

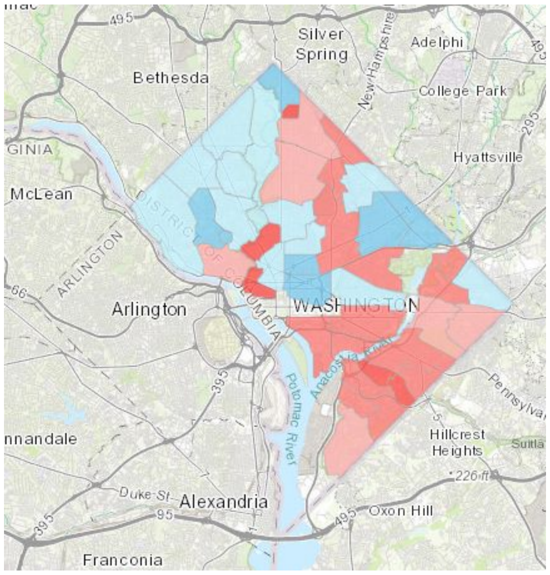


Figure 14: Next City Heat Map

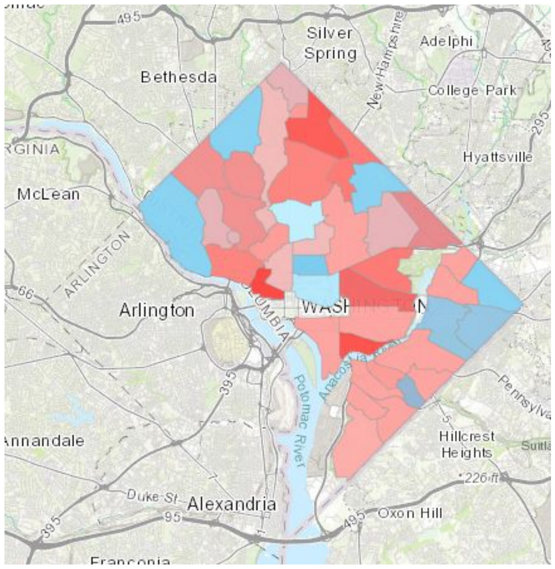


Figure 15 Data Extractors Heat Map

Since the primary factor of each neighborhood score was based on distance from the DC Mean Growth over the 5-year period, one might conclude that the results in the Next City implementation were pretty accurate. However, the heavily advancing growth in the SE and the Falling behind in the NW don't indicate what is really happening.

Our implementation model is more equally distributed. Since we factored every single month into our analysis and added variables such as: Rent, Violent Crime, Demographics, there's more detail in the outcomes. For instance, the first principle component surrounding age dependency discussed in the modeling portion shows a wide range of variance and it affected overall accuracy of our model.

Hot areas such as Navy Yard, makes since thinking about the growth around the ballpark, Union Station had higher scores. Foggy Bottom and Columbia Heights Scored high and low respectively in both cases which was primarily due to high crime in Columbia Heights and very high real estate and Income in Foggy Bottom.

<u>Key Insights</u>

At first glance after EDA, what was initially expected didn't happen. With additional resources, money, growth in housing and jobs, we expected a decline in Poverty and Crime. They both went up. Perhaps this was due to an increase in population and a decrease in affordable resources. As far as crime, perhaps there were just more targets for criminals.

Overall, the interpretations of why or why not certain areas grew and those that didn't is a subjective process. However, the machine learning clustering models doesn't lie, it shows movement between clusters and allowed us to measure growth and essentially compare apples to oranges.

A drawback of the Next City model was that it didn't factor middle year or monthly detail. Their scoring process was based on distance from the citywide mean and a subjective ranking system.

Conclusions and Lesson Learned

We concluded that applying richer data sources and a greater number of instances would result in more accurate results. Because of the distribution of our data and the variance pick up in our machine learning process, we concluded that our model was more informative than theirs and could be improved further with additional features and instances. Additional conclusions are as follows:

1. We spent a lot of time ensuring our data was clean, in fact 80% of our time was spent cleaning, and adjusting our data to numerical types we could use for machine learning.

2. Clustering involved a lot of labeling. Fortunately, the scoring we accomplished during the implementation of the Next City model, helped us label our clusters for the unsupervised clustering.

3. Given the chance to run this project over, we would use Census Tracts shapefiles over OpenData's neighborhood cluster shapefiles. This is because we discovered errors in our Latitude/Longitude remap/grouping process. This was not because of a poor algorithm but because we used the CensutTract center points to map to the neighborhoods shapfile. We chose neighborhoods because they were already labeled and were visually more informative, but the error caused overlap and some of the data crossing into other neighborhoods making the output map not 100% accurate geospatially.

4. Our team had a healthy disagreement about applying more effort to adjust our final dataset from month to month to year to year thus reducing the duplication for certain categories like population and poverty. We started on this process by re-wrangling our data yearly numbers and reduced the duplication, but we ran out of time to convert all of our data to census tracts.

5. Given the availability, we would choose a wider year range such as a full set of census data covering a 10 year period. We would also apply a greater range of features such as: Education, Unemployment, New Construction, Foreclosures, Park and Infrastructure funding, Jobs and businesses added and transportation access.

6. Our ideal product application derived from this project would be to have a user customizable app that allows the consumer to add and remove the data features to see a real time visual of growth and decline. Providing a feedback mechanism to feed into the result would make this application a true data product.

Github

https://github.com/aosanchez/TheDataExtractors

References

https://nextcity.org/features/view/philadelphia-neighborhoods-gentrification-mapping-growth

https://mpdc.dc.gov/page/district-crime-data-glance

https://planning.dc.gov/page/tables

http://opendata.dc.gov/datasets/affordable-housing

https://www.arcgis.com/home/item.html

https://www.yelp.com/developers/documentation/v2/search_api

http://opendata.dc.gov/datasets/neighborhood-clusters