ISyE 6740 – Summer 2022
Final Project
July 31, 2022

Team Member Names: Brendan Danyliuk, Luolin Shao, Nikolos Lahanis

Project Title: U.S Real Estate Home Price Valuation Tool

PROBLEM STATEMENT

As a group, we have decided on the Real Estate industry as our topic of interest. The goal of our project was to create a real estate valuation tool that properly assessed home values as well as a given home's unique features in the city of Atlanta, GA. We will be looking to cluster homes in Atlanta to create groups of housing with similar traits (i.e. home type, number of floors, proximity to city center) in order to create more accurate valuations through segmentation. In addition to our main deliverable of clustering home types, we also expect our model to identify for us the most important features of a house to look at when valuing a home.

DATA SOURCE

Property Features

One benefit to using the Real Estate industry as our topic of interest is that there is a plethora of readily available data surrounding both the general housing market and individual property information, usually in the form of accessible APIs. Our primary source of property data is pulled from Realty Mole Property's API.

Now knowing where we are sourcing our data from, the next question becomes identifying how many total individual properties we should pull to establish the proper sample size for our model to return reliable results. To obtain the specific sample size of the dataframe we are looking for, we first established a baseline of what the total size of the population we are investigating is. In this case, the population we are investigating is the number of housing units in Atlanta, GA. According to the US census, there are 250,000 housing units in the Atlanta, GA Metro area. Knowing that our total population size is 250,000, we can determine what a proper sample size is by either using the necessary equations, an online calculator, or even interpolating from this example table below:

	Confidence level = 95%			Confidence level = 99%		
	Margin of error			Margin of error		
Population size	5%	2,5%	1%	5%	2,5%	1%
100	80	94	99	87	96	99
500	217	377	475	285	421	485
1.000	278	606	906	399	727	943
10.000	370	1.332	4.899	622	2.098	6.239
100.000	383	1.513	8.762	659	2.585	14.227
500.000	384	1.532	9.423	663	2.640	16.055
1.000.000	384	1.534	9.512	663	2.647	16.317

Table 1: Sample Size Calculations

Knowing that we want our sample size to have a confidence interval of +-2.5, and a confidence level of 95%, we determined that a proper sample size for this project needs to be at least 1600 data points. 1600 data points allow us to be confident that the output values of our models will be accurate within a range of +-2.5%, with 95% certainty that this range itself is also accurate.

The Realty Mole Property API contains a total of 90,000 data points for households in Atlanta, GA. Knowing that each request to the API returns 50 households at any given time, we created a request script that randomly iterates through the API and pulls rows out from the dataset until we reach the necessary number of datapoints for our dataset. We used the following query to make the necessary pulls:

```
querystring = {"state":"GA", "City":"Atlanta", "offset":offset}
```

Figure 1: API Request String

We output this to a Pandas Dataframe with the following shape and features columns:

```
1604 rows × 27 columns
```

```
Index(['Unnamed: 0', 'addressLine1', 'city', 'state', 'zipCode',
       'formattedAddress', 'assessorID', 'bedrooms', 'county',
       'legalDescription', 'ownerOccupied', 'squareFootage', 'subdivision',
       'yearBuilt', 'bathrooms', 'lotSize', 'propertyType', 'features',
       'lastSalePrice', 'lastSaleDate', 'owner', 'taxAssessment',
       'propertyTaxes', 'id', 'longitude', 'latitude', 'addressLine2'],
      dtype='object')
```

Figure 2: Initial Dataframe Columns

In this DataFrame, ‘taxAssessment’ is the tax-assessed value of the home, and will act as the response variable for our pipeline, in combination with ‘lastSalePrice’.

Past Home Values

Because the above dataset also contains historical tax assessment and home sale records, we can use a mix of imputation and backwards forecasting to create our past home value forecasts for all the properties that are pulled from our dataset. Our tax assessment values represent the years 2019, 2020 and 2021. Compared with ‘taxAssessment’ of 2021 and ‘lastSalePrice’, we used the maximum of the two to create a new feature named prediction. Prediction will be the response utilized in the following clustering and prediction.

```
taxAssessment
```

```
{'2019': {'value': 69000, 'land': 15000, 'improvements': 54000}, '2020': {'value': 68640, 'land': 15000, 'improvements': 53640}}
```

Figure 3: Before Imputation

2021	lastSalePrice	prediction
159813.2	160000	160000
125440	105000	125440
125120	11569148	11569148
175440	100500	175440
144912.2	123700	144912.2
166800	169400	169400
162514	129700	162514
162514	124700	162514
116972.8	117500	117500
119934.4	159000	159000
144912.2	116400	144912.2
151120	124400	151120
157578	138600	157578
141652.6	141500	141652.6

Table 2: After Imputation

`

Using ‘taxAssessment’ and ‘lastSalePrice’ as a base, our newly created ‘prediction’ column now acts as the response variable for our future analysis.

Initial Data Visualizations

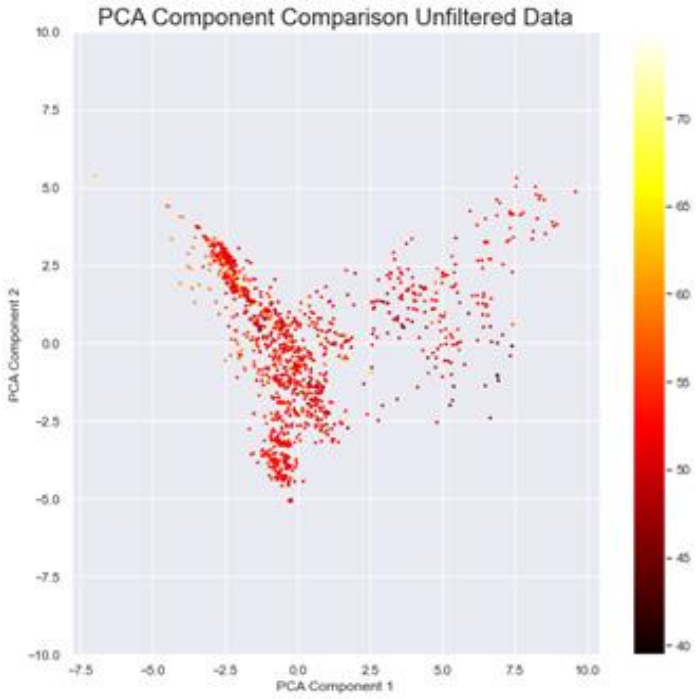

Figure 4: PCA Visualization of Uncleaned Data

In figure 5 above, a visualization of the uncleaned data was done using PCA in order to see if there were any notable patterns or trends that may be noted before starting our analysis. As of now, the PCA visualization didn't seem to show any particularly strong insights into what the dataset might be able to uncover for us quite yet.

METHODOLOGY

Feature/Row Cleaning

For this project, the next step after obtaining our initial dataset was to clean and prepare it; this is done to ensure data consistency. At this step, we cleaned up data types, decided how to handle missing values, and removed any data that was not pertinent to the project. Depending on the breadth of missing data, we used imputation to handle missing data points. Also, dummy values were extensively used to transform text information to digital information for future calculations.

After the data was cleaned and NaN, null and missing values were handled, we began feature selection, to eliminate redundant or uninformative features, as well as handle outlier values in the remaining features. These steps were done to improve model performance and informativeness. Outlier values can skew model results and we want to ensure that we do not overfit the model. Feature selection and data cleaning also helped us to reduce model run times, while keeping model integrity.

The first step we took for feature selection was to understand how correlated our features were. To evaluate the correlation between features, a correlation matrix was used. In our analysis, we chose to use a coefficient threshold of ±0.6. Any features that fell above this threshold were removed to prevent multicollinearity. However, in analyzing the correlation matrix, we decided to include two features (Tax Assessment values in 2019 and 2020) despite a high correlation with 2021 Tax Assessment values, as these features were considered informative; we thought the over-time change in assessment value would be worth keeping those features in.

To begin with, after data cleaning we were left with 1335 rows and 124 features. The correlation matrix for all of these is provided below in figure 5:

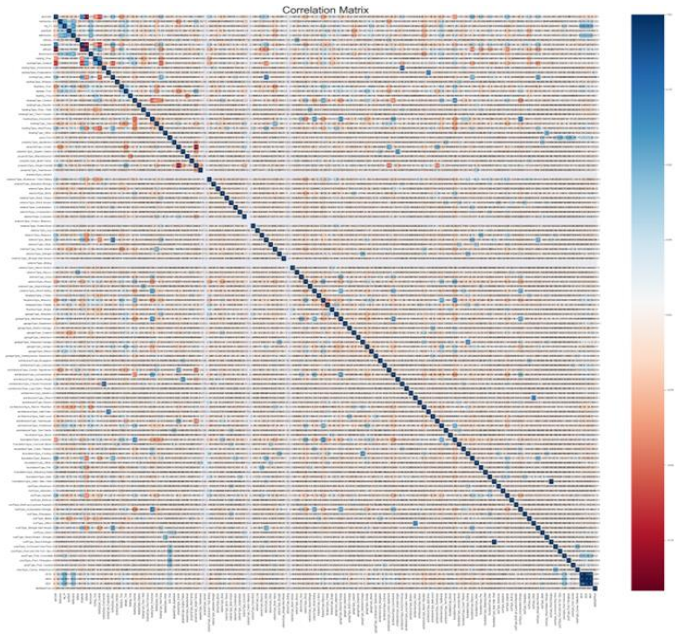

Figure 5: Correlation Matrix for Initial Dataset

After looking at the correlation between features, our next concern was outliers. We chose to utilize the statistical method of Z-score for outlier analysis. We computed the Z-scores for each feature; this gave us a measure of how many standard deviations above the mean values were. This is important, as outlier values can skew a model, and cause poor performance. Examples of how this affects features can be seen below; box and whisker plots were computed, with pre and post filtering by z-score for each feature were plotted to showcase the difference between the two; there are clear changes in values pre and post filtering. We chose to use 3 standard deviations above or below the mean as our cut-off point; this roughly corresponds to 99% of the data, if it were normally distributed.

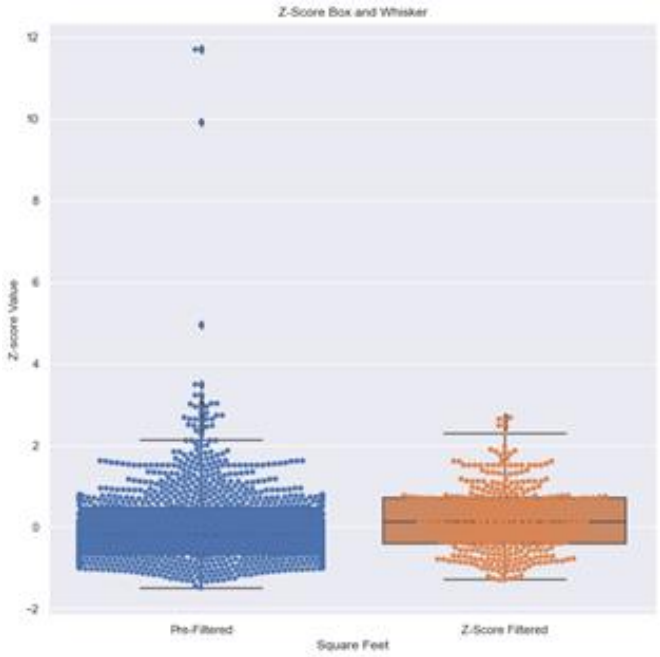

Figure 6: Box Plots for 'Square Feet' Feature

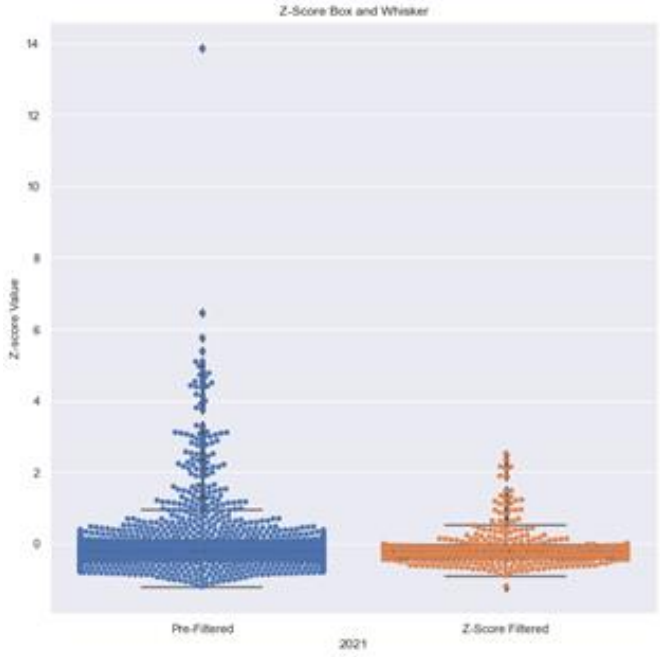

Figure 7: Box Plots for '2021' Feature

For our final step in feature selection, we used LASSO with 10-fold cross validation to help further reduce dimensionality. GridSearchCV was used to test different alpha values. From this, we chose an alpha value of 1,000; this balanced out to a reasonable amount of remaining features; we ended up with 16 features (including the two previous features we chose to include, Tax assessment for 2019 and 2020) to train our model with.

0	bedrooms
1	Sq_Ft
2	yearBuilt
3	bathrooms
4	latitude
5	floorCount
6	fireplace_True
7	heatingType_Central
8	heatingType_Heat Pump
9	exteriorType_Brick
10	garageType_Attached Garage
11	architectureType_Traditional
12	2021
13	lastSalePrice
14	2019
15	2020

Figure 8: Column Names Post-Feature Selection

After completing the feature selection steps, Principal Component Analysis (PCA) was run on the remaining data (384 rows and 16 features). A comparison is provided below:

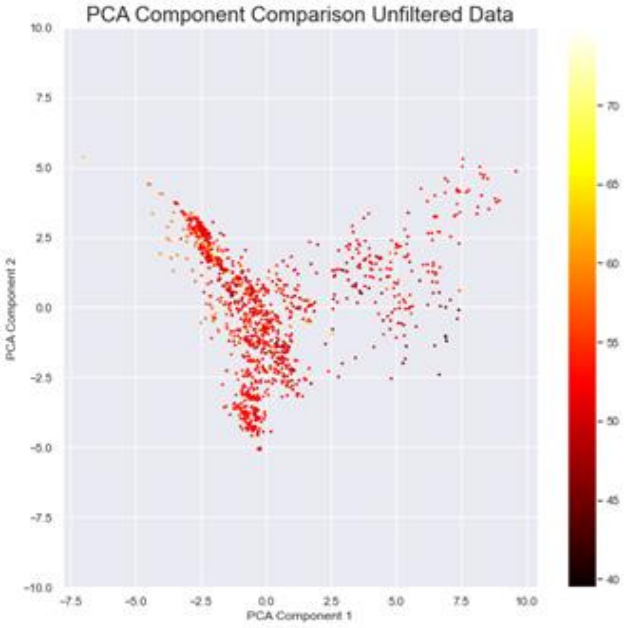

Figure 9: Uncleaned PCA

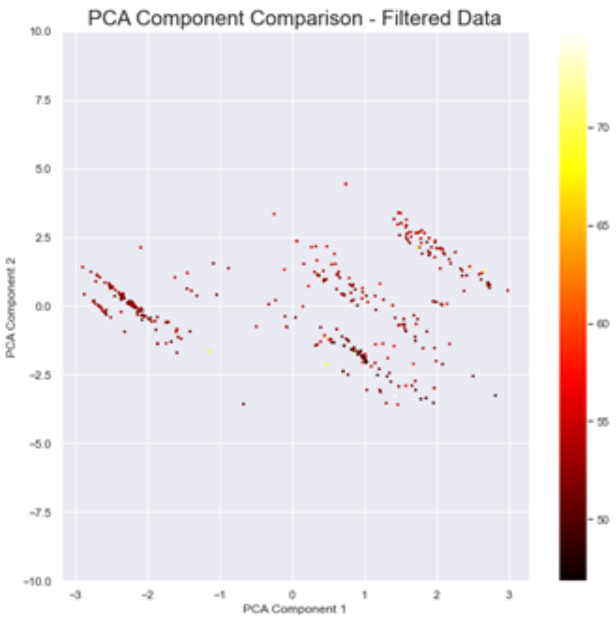

Figure 10: Cleaned PCA

Compared to the unfiltered data, the filtered PCA components showcase the data in more distinguishable groups.

On this final set of data, a Correlation Matrix was run to showcase the difference in features:

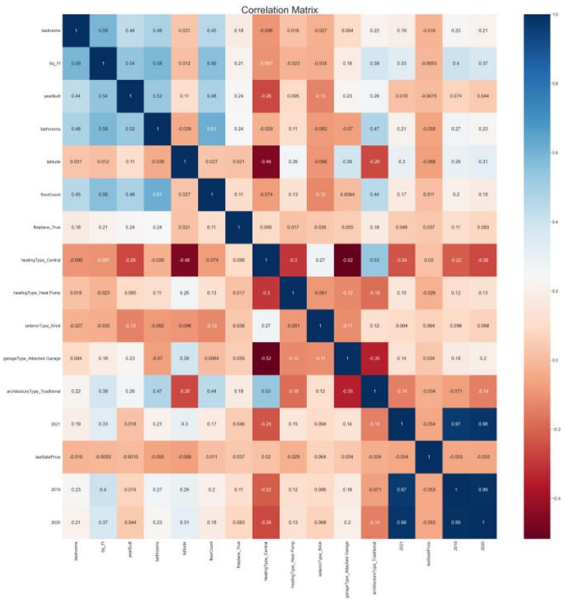

Figure 11: Cleaned Correlation Matrix

As seen above, outside of Tax Assessments from 2019 and 2020, the variables all fall outside of ±0.6

Clustering Models

Now that our dataset has been sourced and cleaned, we can begin to run some ML models to generate some further insights. In the case of our project, we decided to focus on clustering our dataset to see if we could segment home types in Atlanta, GA with similar traits. In order to do this, we ran 3 separate clustering models and compared their performance: K-means, Spectral Clustering, and Gaussian Mixture.

Before running the optimal version of each of these models to group the homes in our dataset, We first have to answer the question: "How many total clusters should we try to label our homes to?" The ideal number of groups for our dataset is different for each model, but can be found for each of our models using the following techniques.

For K-means and Gaussian Mixture, we used the elbow method to compare the overall error of a given model with the total number of clusters that were generated (k). The optimal number of clusters for each model is where the graph of each respective model type decides to "bend". These bending points are indicated in figure 13 for K-means, and figure 15 for Gaussian Mixture. The optimal cluster size for K-means is 6, while the optimal cluster size for Gaussian Mixture is 5.

For spectral clustering, identifying optimal cluster size uses a slightly different technique. By finding the largest eigengap between different eigenvalues at different cluster sizes from an adjacency matrix of data points labeled from spectral clustering, we can determine what spectral clustering deems to be the optimal cluster size for our dataset. The location of the largest eigengap value is indicated by the vertical red line in figure 14, which determined the cluster size to be 35.

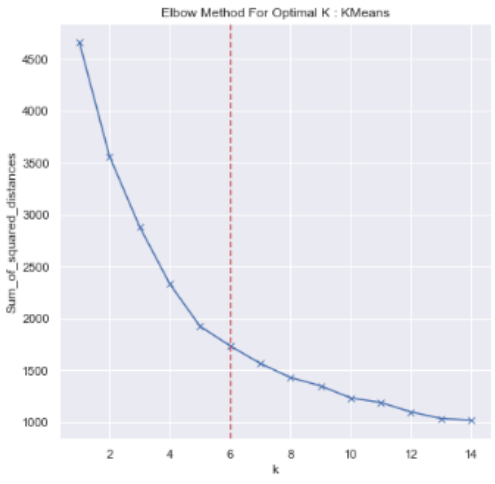

Figure 12: K-Means Elbow Method

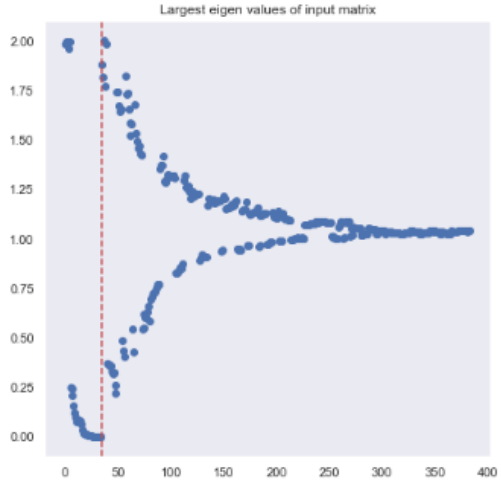

Figure 13: Spectral Clustering Eigengap Heuristic

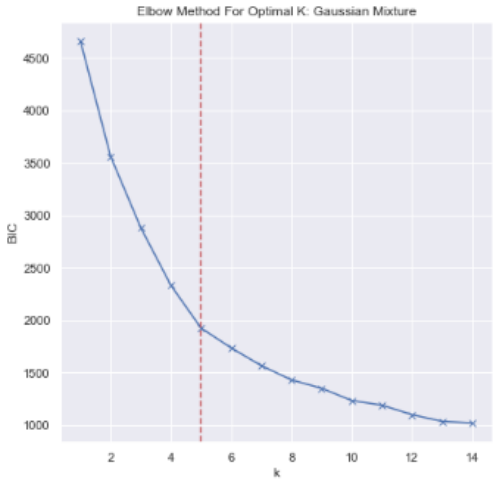

Figure 14: GMM Elbow Method

After determining the optimal cluster size for each of our model types, we can now run these on our cleaned dataset, and see how each of these models decide to go through the labeling process. In order to visualize this cluster labeling, we created the following figures on the next page to visualize how each clustering model performed using PCA. Based on the results of our visualizations, it was clear to us that the Gaussian Mixture Model returned the most accurate, clearly separable groups for the goal of segmenting home types.

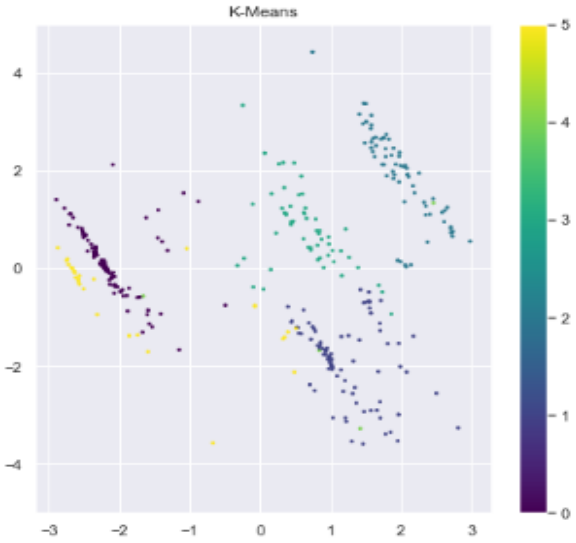

Figure 15: K-Means Visualization

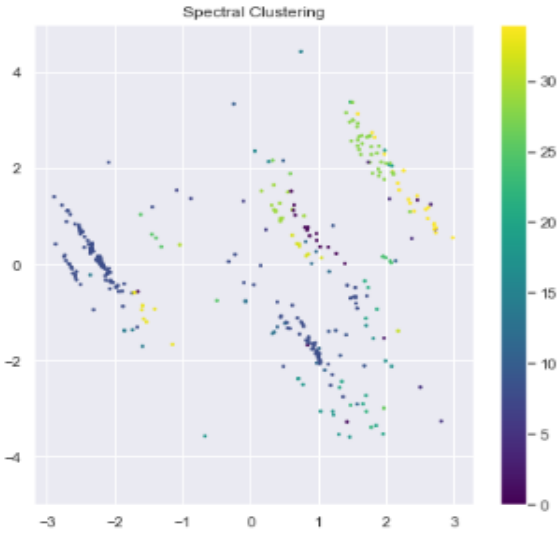

Figure 16: Spectral Clustering Visualization

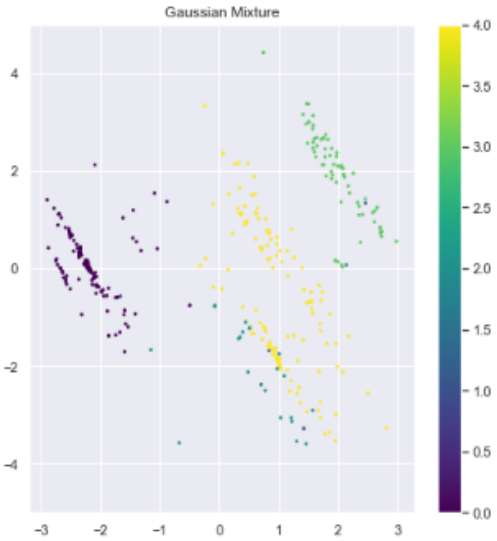

Figure 17: GMM Visualization

EVALUATION AND FINAL RESULTS

Gaussian Mixture Model Results

Table 3 summarizes the average values for the 15 key features from each cluster:

	0	1	2	3	4
bedrooms	4.00	3.00	3.00	4.00	3.00
Sq_Ft	2446.00	2121.00	1550.00	2353.00	1679.00
yearBuilt	1998.00	1990.00	1968.00	2001.00	1984.00
bathrooms	3.00	2.00	2.00	2.50	2.50
latitude	33.75	33.75	33.75	33.88	33.78
floorCount	2.00	2.00	1.00	2.00	1.00
fireplace_True	0.99	1.00	0.96	0.96	0.86
heatingType_Central	0.95	0.75	1.00	0.07	0.50
heatingType_Heat Pump	0.00	0.00	0.00	0.00	0.15
exteriorType_Brick	0.17	0.25	0.88	0.03	0.00
garageType_Attached Garage	0.02	0.25	0.04	1.00	0.00
architectureType_Traditional	1.00	0.25	0.08	0.00	0.00
2021	154402.07	125046.72	198524.70	186195.13	167603.32
2019	127078.71	100700.00	150753.25	151582.39	124243.09
2020	137140.79	111466.31	171344.51	174308.41	144519.62

Table 3: Average Feature Values for Each Cluster

Of the features seen above, the largest differentiators would appear to be square footage, year built, and the tax assessment values. All of these features vary between features in an easily observed manner. Bathrooms and bedrooms show less variation (While these features, in general do not have as much variance, we see clustering around 3-4 bedrooms and 2-3 bathrooms that was not necessarily expected).

Brick houses were an unexpected surprise; a brick house in Atlanta appears to be an indication of increased value, but does not correlate to the year the house was built, although we can see that cluster 2 (the oldest average age cluster) has the highest average for brick houses and year built. When comparing cluster 0 and cluster 2, we see that despite some overlap in area, the assessment values are quite varied, despite having higher square footage and newer builds.

Clusters 2 and 3 seem to have the highest property values, while clusters 0 and 1 seem to have significantly lower property values as compared to other groups. Another interesting insight is that even though cluster 2 has the highest home values, it also has the lowest square footage.

Visualizations

After the clustering, the data was classified into 5 groups(0, 1, 2, 3, 4). We used Plotly to draw the distribution of labeled houses on the map. Plotly enabled us with an interface visualization to zoom in and out the map easily and provided the latitude, longitude and label of each house when the mouse hover over the scattered dots on the map.

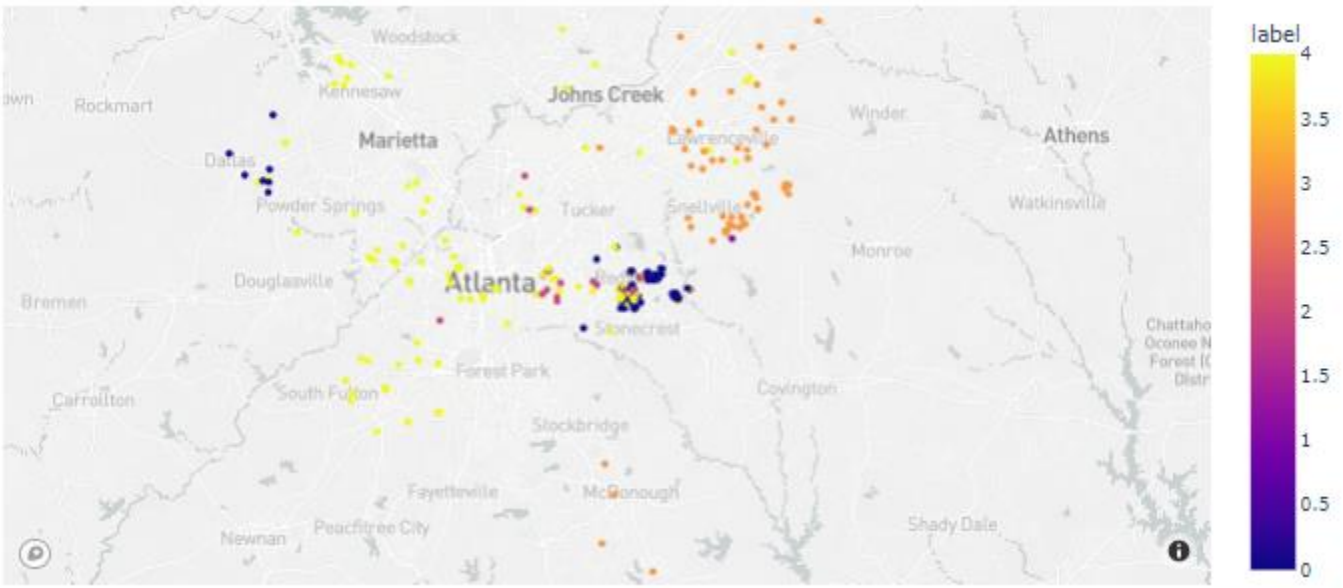

Figure 18: Clusters on Map of Atlanta, GA

Conclusions

In summary, we were able to come up with some interesting insights following the completion of our pipeline.

First, properties in Atlanta can be clustered into 5 independent groups by unique property features, and for any given new property, the property can be classified into one of the 5 groups given information about the property regarding these features.

Next we determined the following properties features to be significant when valuing a home in Atlanta, Georgia: number of bedrooms, square footage, year built, number of bathrooms, latitude, number of floors, whether there is a fireplace, whether central heating is used in the property, whether a heat pump is used in the property, whether the exterior of the property is brick, whether the garage is attached, whether the architecture of the property is traditional, and the tax assessed values of the property in 2019, 2020 and 2021.

Third, among all significant features, Sq_Ft, yearBuilt and the Tax Assessment are the most key features indicating the segmentation. Also, properties constructed with brick are more desirable in Atlanta than properties not made with brick.

We believe that our work provides insight not only on how to cluster property types in the Atlanta, GA metro area, but also identify property features that are most significant to the valuation process. This project also leaves space for improvement if we can monitor the property sale price trend of these 5 groups in the future. Going forward we could build on our analysis by using historical housing price data and macroeconomic information. The addition of these pieces of information would allow us to create a tool for future home value forecasting, and subsequently determine what we expect the future home price of a given house in Atlanta, GA to be.

Appendix

Team Responsibilities

Name	Owner	Description
Proposal	Brendan, Luolin, Nikolos	Initiate research problem and approaches
Data Preparation	Nikolos, Luolin	Data collection from API and data cleaning
Feature Selection & Correlation Matrix	Brendan	Implement variable selection like LASSO and Z-test. Investigate the correlation among variables
Clustering Models	Nikolos	Implement clustering models like Kmeans, Spectral Clustering and Gaussian Mixture to explore the meaningful property features and provide property segmentation.
Map Visualization	Luolin	Visualize segmented data on map of Atlanta
Final Report	Nikolos, Brendan, Luolin	Compile final results into final written report

Works Cited

- *Realty mole property API: How to use the API with free api key*. RapidAPI. (n.d.). Retrieved July 14, 2022, from https://rapidapi.com/realtymole/api/realty-mole-property-api/details
- Bureau, U. S. C. (2022, March 8). *American Community survey 5-year data (2009-2020)*. Census.gov. Retrieved July 21, 2022, from https://www.census.gov/data/developers/data-sets/acs-5year.html
- Dessel, G. V. (2020, February 5). *How to determine population and survey sample size?* CheckMarket. Retrieved July 21, 2022, from https://www.checkmarket.com/blog/how-to-estimate-your-population-and-survey-sample-size/
- Ciortan, M. (2019, January 7). *Spectral graph clustering and optimal number of clusters estimation*. Medium. Retrieved July 26, 2022, from https://towardsdatascience.com/spectral-graph-clustering-and-optimal-number-of-clusters-estimation-32704189afbe