Predicting Gentrification Using Machine Learning in a Post Covid-19 World

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Eric Guan

Spring, 2021

Technical Project Team Members

Jonathan Wen

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Predicting Gentrification Using Machine Learning in a Post Covid-19 World

Proposing a New Tool

Eric Guan
Computer Science
University of Virginia
Charlottesville, Virginia
eg5jn@virginia.edu

Jonathan Wen
Computer Science
University of Virginia
Charlottesville, Virginia
jsw2dg@virginia.edu

ABSTRACT

In 2020, the world was struck by a global pandemic. Many countries - the US especially - had to cease normal function of life and business. As a result, millions of Americans lost their jobs and thousands of businesses closed. The US economy saw drops in GDP unheard of since The Great Depression. In lieu of all this, the housing market has actually steadily been going up, showing more pronounced growth among both ends of income distribution. This market reaction is dissimilar to the previous recession a decade ago, indicating that there are other factors influencing this increased demand. In addition, looking at data that has been released on human migration patterns, there has been a clear indication of large groups leaving urban areas [1]. While in specific instances this was due to a want for less crowded areas with lower risk of transmission, it is important to note that a significant number of people were also displaced due to negative economic effects [2]. As current data is not all in yet, we propose a machine learning tool that could be used in the future to analyze housing prices for Low and Middle Income (LMI) areas and formulate predictions, which could be used to inform the containment of gentrification in these zones. This paper highlights the preliminary work done towards that end and compares the various methods used to formulate a recommendation for future predictive tools.

1 Introduction

This project focused on the lower spectrum of income distribution, due to alarming statistics on how Covid-19 has disproportionately impacted minorities and people within that financial bracket [3]. Due to 2020 being a special year, the proposed tool is one utilizing demographic and housing market data to create a baseline model which can be used in the future for families in these vulnerable communities. Specifically, using openly available data for housing prices in cities from Kaggle, we formulated insights into how housing prices will change in a volatile market. We used Python's scikit-learn and JupyterLab to implement 4 different machine learning techniques to find whether or not affordable housing is decreasing for lower income residents, as well as see if rising housing prices are spreading to such neighborhoods outside urban centers. These insights can be used to inform future economy forecasts and for homeowners in respective neighborhoods to better understand possible changes in their property value in the coming future.

1.1 Background

Gentrification has been a well documented process occurring in certain cities and areas for quite some time now. In general, when the prices of houses begin increasing in areas it is an indication that particular neighborhood is "gentrifying"[4], meaning that the area is changed economically through real estate investment and new higher-income residents moving in. This usually means a change in the demographic level as low-income minority residents with little education are displaced and replaced with higher income residents who can afford the rising cost of living [5]. While gentrification is hailed by some as key

to revitalizing cities and impoverished areas through an influx of money and new real estate developments, there are several key problems that it causes. These include cultural displacement, homelessness, and increased crime [6].

Upon first glance, gentrification does in fact remove poverty from a specific area within a city. However, that poverty does not simply disappear; it expands into the suburbs [7]. In fact, due to gentrification and general migration patterns [8], poverty is expanding outwards, while also leaving areas of extreme poverty. With the global pandemic Covid-19 affecting both human migration patterns [1] and the economy [9], it became clear that how communities change would also be impacted. As the housing market soared [2], minority and lower income communities were hit especially hard [11]. When looking at these communities, it becomes increasingly clear that they lack investment in infrastructure and development [11]. The problem is when investment does come, these people are (meaning they are often forced into areas that struggle from similar lack of investment) trapping them within a cycle of poverty [12]. These negative effects are of course not the intended purpose of redevelopment, meaning that a better understanding needs to be gained of how gentrification is progressing in these urban centers. For the purposes of creating a proof of concept for the proposed tool, we used 1990 census housing data from California [13]. The most pertinent data was not readily available to the public for research, however the 1990 data was still sufficient, as it contained valuable information on housing valuations and the incomes of those who lived there.

1.2 Prior Research Analysis

The California housing dataset is not one that is unfamiliar. In the University of Virginia course CS 4774: Machine Learning, this dataset was used to form a k-means clustering model based on geographical location and median house value. The data was not cleaned as thoroughly as described in section 2.1; however, the ocean proximity column had to be changed to numerical values, since a machine learning model can not be run on text. A LabelEncoder was used to convert the text describing distance to ocean to values ranging from 0 to 4.

```
encoder = LabelEncoder()
encoder.fit(["INLAND", "<1H OCEAN", "NEAR OCEAN", "NEAR BAY", "ISLAND"])
df.ocean_proximity = encoder.transform(df.ocean_proximity)


df.info()
df.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   longitude           20640 non-null  float64
 1   latitude            20640 non-null  float64
 2   housing_median_age  20640 non-null  float64
 3   total_rooms         20640 non-null  float64
 4   total_bedrooms      20433 non-null  float64
 5   population          20640 non-null  float64
 6   households          20640 non-null  float64
 7   median_income       20640 non-null  float64
 8   median_house_value  20640 non-null  float64
 9   ocean_proximity     20640 non-null  int64
dtypes: float64(9), int64(1)
memory usage: 1.6 MB
```

Figure 1: Using a LabelEncoder to ensure uniformity of data. All data types are now suitable for analysis

A k-means clustering algorithm was then applied to the data. This algorithm works by selecting k number of means for a particular variable, and then forming clusters for data points close to each mean. The means are then recalculated for each cluster and the points are shifted around clusters based on how close it is to the mean determined by the formula for Euclidean distance provided by $D(x,y) = \sqrt{\Sigma \quad |x_i - y_i|^2}$. The goal is that eventually, the clusters will normalize and reach a point at which the points no longer shift. K-means clustering is especially useful for large datasets because it is able to adapt to new data points quickly, the many clusters are able to respond to different shapes, sizes, and relationships between data points. Using the elbow method of identifying k by observing the graph of mean distances vs the number of clusters, it was determined that 5 was the optimal number of clusters, and the resulting cluster map is shown below.

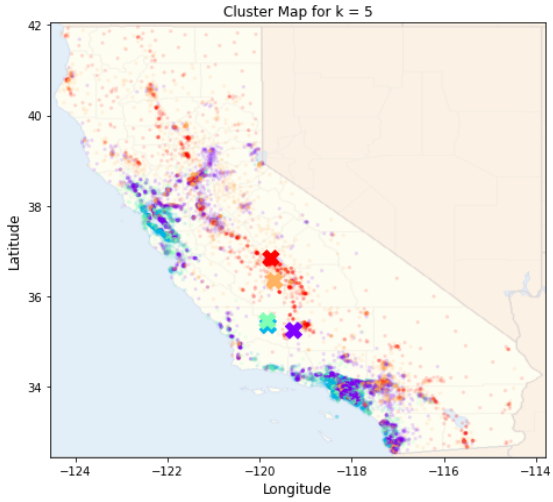


Figure 2: California housing cluster map for k = 5 based on median house value. Please refer to Appendix A for elbow graph and cluster maps k = 2 through k = 4

Even though the map shows that much of the data is clustered in the Los Angeles and San Francisco areas, the clusters appear to be in the middle of the state, which we believe to be a conflict of two large data spots resulting in means almost equidistant to both places. Although the central positioning of the clusters does not provide any clear cut groups within the data, the high density of expensive homes closer to 2 specific urban areas shows that the prices of homes increase by a sizable margin when specific desirable traits - such as ocean proximity - are present. However, since the clustering was done based on home value means, it did not provide much information on median income and by extension gentrification. That being said, this prior research helped to inform the work seen in the following section by illuminating geospatial relationships between data, and contributed to the conclusions formulated in section 4.

2 Tool Architecture

To better explain the progression of our research, the following sections will be divided into four sections. We will start with how we prepared our data for the different methods of manipulation. We will then talk about two different methods of regression we used and move onto a newer algorithm, XGBoost that has recently gained popularity.

2.1 Data Cleaning

As mentioned in section 1.1, the data used is California housing data from the 1990 Census provided on Kaggle [4]. This data includes attributes such as geographical location (longitude and latitude), statistics per street block (number of rooms, number of bedrooms, median age, number of households, median income measured in tens of thousands of USD, median house value measured in USD), and proximity to the ocean. All of these attributes have some correlation to housing prices, but we specifically wanted to compare median household income with the other attributes. To clean the data the null values in each column would first have to be replaced, as those would cause errors down the road. For columns containing null data points, these points were replaced using the mean of that respective column. The data then had to be standardized so that it's easier to perform these analyses. Using Scikit-learn's StandardScalar which calculates the standard score of value x as $z = \frac{x-\mu}{s}$ with μbeing the average of the column [14]. Once this scalar object is created, the data can then be fitted onto it. The data now follows this format:

population	households	median_income	median_house_value
-0.974429	-0.977033	2.344766	2.129631
0.861439	1.669961	2.332238	1.314156
-0.820777	-0.843637	1.782699	1.258693
-0.766028	-0.733781	0.932968	1.165100
-0.759847	-0.629157	-0.012881	1.172900

Figure 3: Standardized subset of 1990 California housing Census data

Now that the data is standardized, we can move forward checking the columns for any relations and prepare the data for the different types of regression we want to perform.

2.2 Linear Regression

As mentioned in section 2.1, the data can now be checked for any relationships we want to consider. The Python library Matplotlib was used to create plot diagrams that will

compare each of the columns of data with the median house value column (refer to Appendix B).

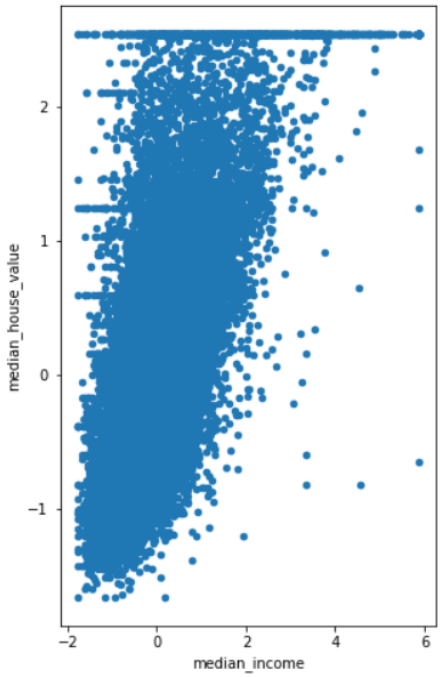


Figure 4: Median Income vs. Median House Value

From the plot shown in Figure 4, there is a linear relationship that can be seen between median income and median house value . The data can then be split into X (input) and Y (output) data points from our datasets. The Y set in this case will be the median house value, while the input X data will be the rest of the columns. The input and output data was split so that the dependency of the data could be determined. Before any kind of regression can actually be performed, the data needs to be split into a training set and a test set. This is done in order to prevent overfitting while developing the tool. The dataset covers an extremely vast region so even though the dataset is large, it cannot be assumed that every single property has been included. When analyzing this data we only want to find correlations between attributes without having the tool memorize training data. Thus, a training set was created so that the model can be tuned, before working with the test set at the end. The dataset can be split into training and testing sets by using *test_train_split* from *sklearn.model_selection* so that we get *trainX, testX, trainY, testY* initialized. The test dataset size was set to 0.2/ 20% of the data which leaves 80% to be the training dataset, which is a common ratio drawing from the Pareto principle.

Because median income was the only plot that showed linearity, we decided to isolate that column from the training and test sets to create alternate sets for X. We did this to see if the linear relationship indicated that median income affected median house prices with lower error. Given that there is one independent variable (median income) and one dependent variable (median house price), it is appropriate to apply a linear regression model to this data. Linear regression requires one independent variable and one dependent variable, which is satisfied by the selection of data we want to correlate. We made a variable *linear_reg = LinearRegression()* from the LinearRegression model imported from sklearn. Using the built in fit() method from *LinearRegression* the X training set was fit with the Y training set. The Y can then be predicted by declaring *predictY = linear_reg.predict(income_x_test)* which can then be used to compute the root mean square error and r^2values. Root mean square error is important in order to measure the error of each model relative to each other. RMSE gives a heuristic view on the "distance between the vector of predicted values and the vector of observed values" [15]. The formula for RMSE is provided by $RMSE = \sqrt{\frac{\Sigma(predicted - observed)^2}{n}}$with *n* being the total number of elements in the dataset, *predicted* being *predictY* and *observed* representing the actual Y values. Using sklearn, we can square root the value of *mean_square_error(testY, predictY)* to get the RMSE. The calculated RMSE was 0.721259591424315. Next we need to calculate our r^2 value, or coefficient of determination. r^2 is used as a statistical measure of how closely the data fits the regression line. $r^2$is determined on a scale of 0% to 100%, with 0% meaning that the model does not explain any variability in our data and 100% meaning that the model explains all variability of our data around the mean. As with the RMSE calculation, the sklearn library was used to produce our r^2value using *predictY and testY.* The r^2 value produced was 0.47190835934467723, which indicates that linear regression moderately fits our data. As mentioned earlier in this section, ideally the regression model is able to match the pattern of the data without

overfitting. An r^2 of ~0.47 indicates a moderate recognition of this pattern however is not as accurate as we could be, as explained in the next section.

2.3 Random Forest Regression

Random forest regression expands upon the decision tree algorithm by picking random values from the training set, building a decision tree for these random points, then choosing a number *n* trees to perform the prior 2 steps and average the values across all trees. Decision trees are known to cause overfitting issues when applied to data, through prioritizing direct better choices over better long term choices (in essence, a greedy algorithm). Random forest regression limits this issue while remaining fast and robust when compared to other methods of regression.

```
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,
                      max_features='auto', max_leaf_nodes=None,
                      min_impurity_decrease=0.0, min_impurity_split=None,
                      min_samples_leaf=1, min_samples_split=2,
                      min_weight_fraction_leaf=0.0, n_estimators=20,
                      n_jobs=None, oob_score=False, random_state=None,
                      verbose=0, warm_start=False)
```

Figure 5: Parameters for our Random Forest Regressor

As shown in Figure 3, a random forest regressor was created using *sklearn.ensemble* with 20 trees in the forest (n_estimators). The random forest regressor was used to fit our *trainX* and *trainY* data, then calculated *predictY2* using the provided *.predict()* method on *testX*. We As with the linear regressor, the RMSE and r^2 values were calculated to be 0.4271544001636753 and 0.8147765970827132, respectively. From this we were able to implement random forest regression, where we could see how close our model was to the data and how accurately the model can predict median housing prices off of median income. When comparing the results from the random forest regressor to the linear regressor, we can see that the RMSE is significantly lower, indicating that the random forest regressor produced less error. Additionally, the r^2 value was significantly higher, showing that the random forest regressor fit the data more closely.

2.4 XGBoost

To further our understanding of technologies pertaining to the topic, we explored an algorithm called extreme gradient boosting (XGBoost) that has gained a lot of popularity for tabular data. Gradient boosting refers to an algorithm used to optimize weak learning machines (one whose performance is at least slightly better than random chance) [16]. To perform gradient boosting, a loss function is needed, which in our case is RMSE that needs to be minimized, and a weak learner which for us was decision trees which use greedy methods of making choices. We add more trees to our algorithm such that the other existing trees are not modified while a gradient descent algorithm is applied to minimize the RMSE. In parameterizing the tree it now lowers loss, hence following the gradient. This method (also known as gradient descent) does reduce the RMSE value, however can lead to overfitting. XGBoosting follows the same principle of gradient boosting, however uses a more regularized formalization model to increase performance while giving more control on overfitting. We import the *xgboost* library and initialize the regressor using the parameters shown in Figure 6.

```
import xgboost as xgb
xgbReg = xgb.XGBRegressor(objective ='reg:squarederror', colsample_bytree = 1,
eta=0.3, learning_rate = 0.1, max_depth = 5, alpha = 10, n_estimators = 1000)
```

Figure 6: XGBoost Regressor initialization

We chose to use 1,000 decision trees here which helps the gradient descent reduce the RMSE. We fit our *trainX and trainY* from Sections 2.2/2.3 then initialize *predictY3* so the RMSE and r^2 values can be calculated and compared from the XGBoost regressor with those from the linear and random forest regressors. The calculated RMSE value was 0.3890693997547225, which shows to be the lowest error calculated so far. This reinforces the idea that using gradient descent can be used with decision trees to further reduce error. The calculated r^2 value was 0.8454214762936871 which was actually slightly higher than the r^2 from random forest regression.

3 Limitations and Considerations

As mentioned in section 1.1, we were limited in our creation of an effective machine learning model due to the limited availability of open source data. The ideal data source for the proposed tool in this paper is the American Housing Survey Metro Internal Use File, a file created every 2 years jointly by the Census Bureau and the United

States Department of Housing and Urban. This file has unfiltered census response data from many large urban centers in the US, including important columns for income, race, and geography. There is a Public Use File which is released as well, however many of these key variables are masked to ensure confidentiality of the responders [17]. Without this data, it is difficult to construct a machine learning model that is applicable in the larger context of tracking and predicting gentrification.

Gaining access to the Internal Use File is no easy task. Only researchers associated with the secure Research Data Center (RDC) can gain access, which is limited to 30 locations around the country. The closest location to the University of Virginia is in Georgetown. Since there was no Census Bureau employee near the University and the fact that a Special Sworn Status was required even for approved researchers, the proposed tool in this paper was built using an alternative dataset from 1990 using specific California housing data. This dataset was lacking in much of the demographic data that the Internal Use File would have, therefore it is important to note that conclusions formulated by our tool are limited by these confounding variables.

3.1 Risk Analysis

The risks associated with this tool are not many, since the dataset used to construct it is a publicly available open source set from 30 years ago. Even though the data is pulled from United States Census data, many features which could have been used to identify responders have been masked. As discussed in section 3.1, this extra security was the tradeoff for slightly worse performance for the tool. For future research into tools of this nature using the recommended file from section 3, researchers will have to publish their results with hesitation towards releasing their code, especially pertaining to data cleaning. Most importantly, the datasets used for these tools must not be given to anyone without approval from the Census Bureau. If the appropriate measures are followed concerning confidentiality of census participants, there is no reason to believe that this tool will fail to meet its needs.

4 Conclusions

The machine learning models applied to the 1990 California Housing Dataset serve as a proof of concept for the tool proposed by this paper. Using data containing key information such as median housing value, median yearly income, number of bedrooms, and ocean proximity, we were able to use 3 methods to accurately model how housing prices will change in accordance with annual income.

4.1 Interpreting Results

As mentioned in Section 2, the root mean square error (RMSE) describes the standard deviation of prediction errors and indicates how well the data surrounds the fits made from regression. RMSE is inversely related to the coefficient of determination (r^2) which describes how well our regression fits the pattern of our data. In other words, smaller prediction errors will produce a better fit. We scaled both our RMSE and r^2 on a scale of 0-1. Between the three methods used, linear regression is the most common as it can be applied as a basic fit for data. From using linear regression on median income and median house prices, we got an RMSE of 0.721259591424315 and r^2 of 0.47190835934467723. This means that linear regression somewhat fits the data, however due to a notably higher RMSE (compared to later regressions used), we can conclude that linear regression will not provide the best fit for the data. Using random forest regression, we lowered our RMSE to 0.4271544001636753 and increased r^2 to 0.8147765970827132. Because random forest regression libraries will adjust for overfitting we can conclude that random forest regression will provide a much better fit for the data compared to a linear model. Beyond median income, random forest regression would still be a better fit for the data compared to linear regression as more of the independent attributes get added because it can discover more dependencies with Y (median house price). With gradient descent and the optimized XGBoost algorithm we were able to slightly lower our RMSE to 0.3890693997547225 while our regressor still closely fit the data, as shown by an r^2 of 0.8454214762936871, which is 0.0306448792109739 higher than random forest regression. As mentioned in section 2.4, the difference in r^2 between random forest regression and XGBoost is much smaller than that of either of these to linear regression. The algorithm for gradient descent prevents larger increases in our loss function (which in our case is the RMSE). As mentioned earlier, a lower RMSE will typically relate to a higher r^2 thus explaining why the slightly lower RMSE in

our XGBoost resulted in this higher r^2. We hypothesized that there would be a strong, direct linear relationship between median income and housing prices, but were surprised to see that the data was more logarithmic than expected. The linear regression showed higher RMSE, which can be explained by the data tapering off at higher income levels.

4.2 Future Implementations

The tool described in this paper is a partial implementation. There are still many other applications of machine learning that could be used to model the housing data, such as Support Vector Machine Regression or Deep Learning. A large consideration was placed on standardizing the data as well as removing null values so that the analyses would be clearer, however looking at the results there is still more that could be done in terms of outlier removal. We recommend that future research into this kind of tool spend adequate time filtering outliers to ensure that the spread of data is more normal. In addition, the tool could be adapted to run the models on other variables/data should it become available in the future. To that end, using the American Housing Survey Metro Internal Use File to be described in section 3.1 is our top recommendation for creating the most comprehensive machine learning tool to analyze gentrification. Since this dataset has many more variables, it will include information about renting, which is especially important in very condensed urban areas. The United States Department of Housing and Urban Development defines housing as affordable when it consumes no more than 30 percent of a household's annual income [18]. As described in section 1.1, housing becoming too expensive in less affluent areas is a key characteristic of gentrification. Having data about rent paid per month as well as income of each responder would allow a new boolean column to be constructed determining if each respective person was in a situation where their housing could be described as affordable. This column could be used in tandem with similar analyses that this paper conducted to generate predictions for trends that would better inform gentrification studies.

4.3 Broader Impact

The pandemic of 2021 marks a new era for society moving forward. Beyond many everyday tasks and jobs becoming virtual, human migration patterns have been greatly impacted [1]. In addition, studies over the past 10 years have examined the dangers of redeveloping cities without consideration for already existing populations [5][6][7][9][10]. These dangers highlight the need for a better understanding for how gentrification is already happening, as well as better predictive models for how gentrification progresses to suburbs and other urban areas. Many current models have limited data for specific metropolitan areas, utilizing census data collected every 10 years. The proposed tool would be more effective because the ideal file described in section 4.2 is published every 2 years, and allows for extrapolation concerning affordable housing, a key metric for how likely a community is to gentrify. Having a tool that can provide predictions every 2 years also allows for quicker comparisons to past models, so the tool could be adjusted for confounding variables, meaning the accuracy would greatly improve over time. These results could inform city planners so that urban developments continuing into the 21st century can be more sustainable and safer for the communities they affect.

ACKNOWLEDGMENTS

Thank you Professor Anil Vullikanti and Professor Kevin Sullivan for reviewing this report. Thank you Professor Aaron Bloomfield for guiding the overall capstone process.

REFERENCES

[1] Cynthia Paez Bowman. 2021. Coronavirus Moving Study Shows More Than 15.9 Million People Moved During COVID-19. (February 2021). Retrieved March 20, 2021 from https://www.mymove.com/moving/covid-19/coronavirus-moving-trends/

[2] Courtenay Brown. 2020. More evictions could be looming for America's renters. (June 2020). Retrieved March 19, 2021 from https://www.axios.com/eviction-crisis-coronavirus-351bb693-a04f-4ea1-a27d-dceb5163af14.html

[3] Solomon Greene and Alanna McCargo. 2020. New Data Suggest COVID-19 is Widening Housing Disparities by Race and Income. (June 2020). Retrieved March 25, 2021 from https://www.urban.org/urban-wire/new-data-suggest-covid-19-widening-housing-disparities-race-and-income

[4] Frank Olito. 2019. 7 signs your neighborhood is gentrifying. (September 2019). Retrieved March 15, 2021 from https://www.businessinsider.com/signs-your-neighborhood-is-gentrifying

[5] Michael Maciag. 2021. Gentrification in America Report. (April 2021). Retrieved March 20, 2021 from https://www.governing.com/archive/gentrification-in-cities-governing-report.html

[6] Emily Chong. 2017. Examining the Negative Impacts of Gentrification. (September 2017). Retrieved March 20, 2021 from

http://www.law.georgetown.edu/poverty-journal/blog/examining-the-negative-impacts-of-gentrification/.

[7] Daniel C. Vock. 2021. Suburbs Struggle to Aid the Sprawling Poor. (April 2021). Retrieved March 21, 2021 from https://www.governing.com/archive/gov-suburban-poverty-gentrification-series.html

[8] Reid Wilson. 2020. Americans move less, but those doing so are leaving urban centers. (December 2020). Retrieved March 24, 2021 from https://thehill.com/homenews/state-watch/530515-americans-move-less-but-those-doing-so-are-leaving-urban-centers?rl=1

[9] Aaron Klein and Ember Smith. 2021. Explaining the economic impact of COVID-19: Core industries and the Hispanic workforce. (February 2021). Retrieved March 14, 2021 from https://www.brookings.edu/research/explaining-the-economic-impact-of-covid-19-core-industries-and-the-hispanic-workforce/#cancel

[10] Kovie Biakolo. 2020. The Cities Where Gentrification and Covid-19 Collide. (September 2020). Retrieved March 26, 2021 from https://zora.medium.com/the-cities-where-gentrification-and-covid-19-collide-f672f44f0a9b

[11] Jason Richardson, Bruce Mitchell, and Juan Franco. 2020. Shifting Neighborhoods: Gentrification and Cultural Displacement in American Cities " NCRC. (November 2020). Retrieved April 26, 2021 from https://ncrc.org/gentrification/

[12] Institute for Children, Poverty, and Homelessness. 2018. The Process of Poverty Destabilization: How Gentrification is Reshaping Upper Manhattan and the Bronx and Increasing Homelessness in New York City. (September 2018). Retrieved March 31, 2021 from https://www.icphusa.org/reports/the-process-of-poverty-destabilization/

[13] Cam Nugent. 2017. California Housing Prices. (November 2017). Retrieved March 21, 2021 from https://www.kaggle.com/camnugent/california-housing-prices

[14] Anon. sklearn.preprocessing.StandardScaler. Retrieved April 10, 2021 from https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[15] James Moody. 2019. What does RMSE really mean? (September 2019). Retrieved April 15, 2021 from https://towardsdatascience.com/what-does-rmse-really-mean-806b65f2e48e

[16] Jason Brownlee. 2020. A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning. (August 2020). Retrieved April 17, 2021 from https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/

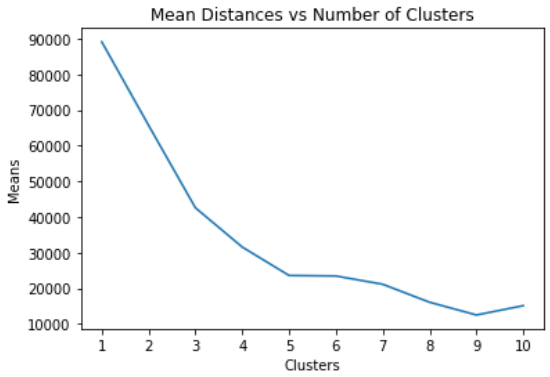
[17] U.S. DEPARTMENT OF HOUSING AND URBAN DEVELOPMENT, U.S. DEPARTMENT OF COMMERCE, and U.S. CENSUS BUREAU. 2019. Using the Internal Use File (IUF) . (March 2019). Retrieved March 16, 2021 from https://www.census.gov/content/dam/Census/programs-surveys/ahs/tech-documentation/American_Housing_Survey_AHS_Using_the_Internal_User_File_IUF.pdf

[18] Matthew Yglesias. 2014. Everything you need to know about the affordable housing debate. (April 2014). Retrieved March 23, 2021 from https://www.vox.com/2014/4/10/18076868/affordable-housing-explained

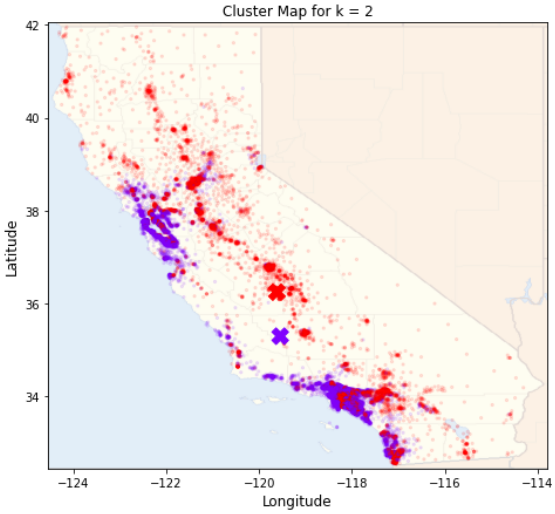
APPENDIX

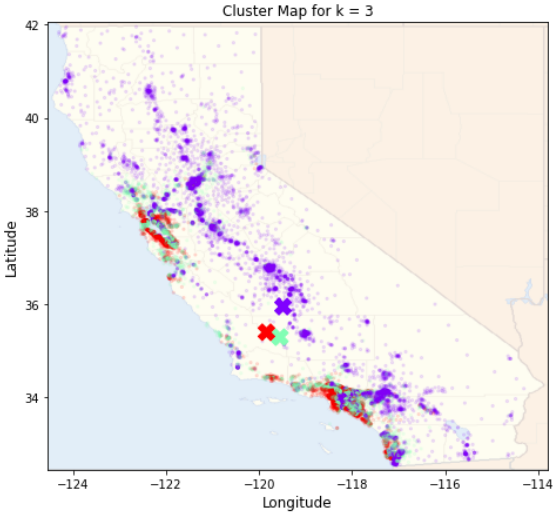
A. PRIOR RESEARCH ANALYSIS

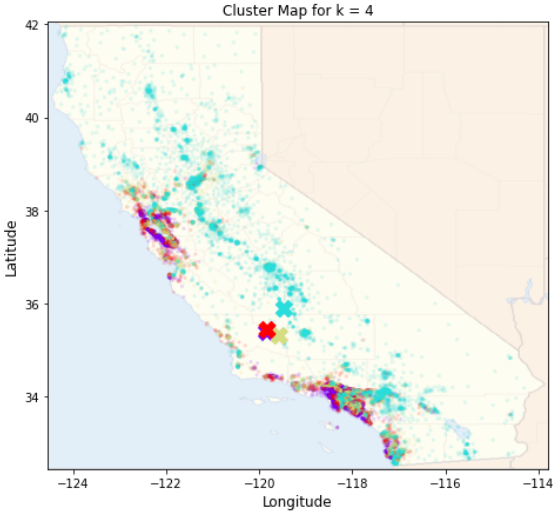
A.1 K-means Elbow Plot



A.2 K-means Cluster Plots for k=2 through k=4

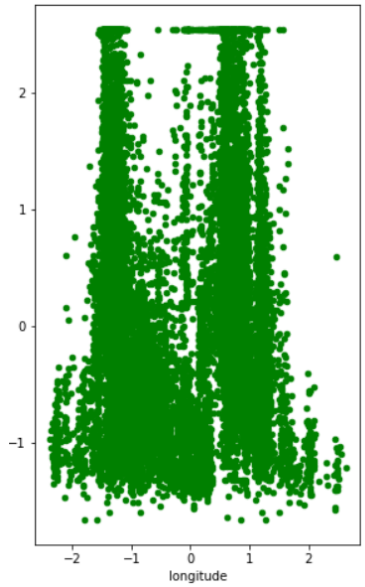




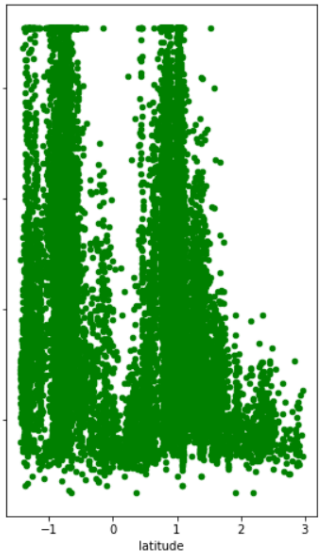


B. TOOL ARCHITECTURE

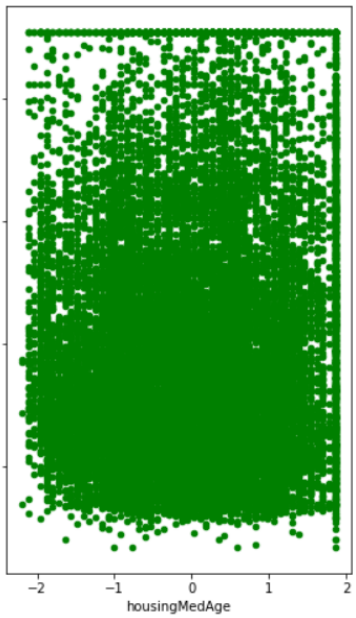
B.1 Matplotlib Plot Chart for Mean House value vs. Longitude



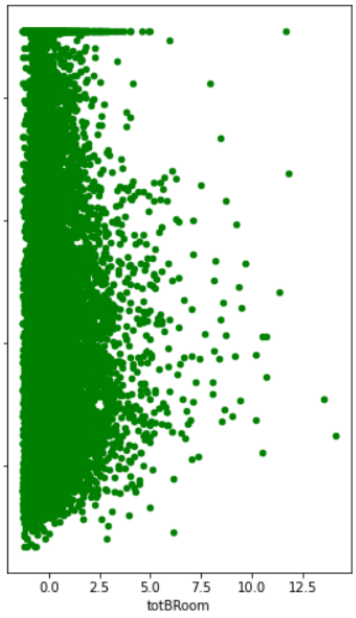
B2. Matplotlib Plot Chart for Mean House value vs. Latitude



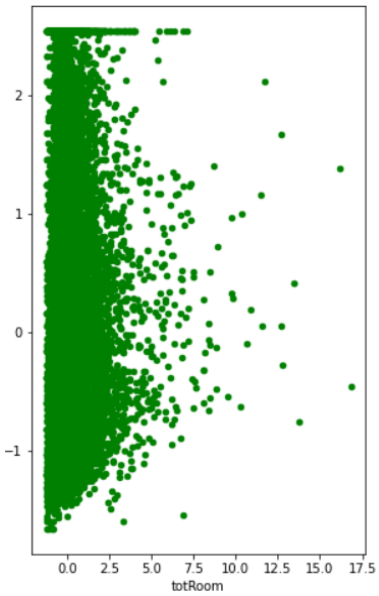
B3. Matplotlib Plot Chart for Mean House value vs. House Median Age per Block



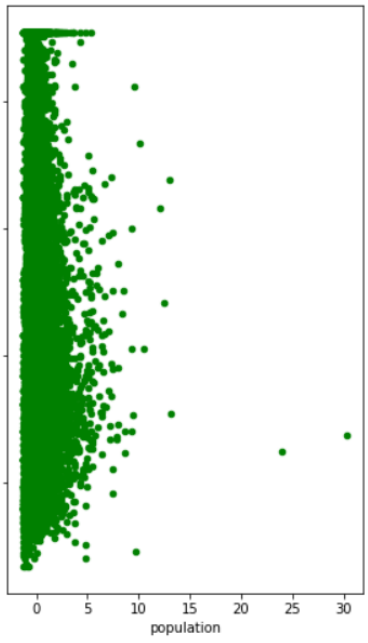
B4. Matplotlib Plot Chart for Mean House value vs. Total Rooms per Block



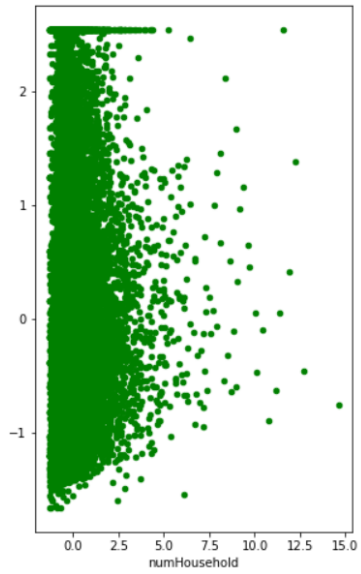
B6. Matplotlib Plot Chart for Mean House value vs. Population per Block



B5. Matplotlib Plot Chart for Mean House value vs. Total Bedrooms per Block



B7. Matplotlib Plot Chart for Mean House value vs. Number of Households per Block



B8. Matplotlib Plot Chart for Mean House value vs. Proximity to Ocean