Using Machine Learning K-Means Clustering to Comprehend California Housing Prices

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

Anh Nguyen

Spring, 2023

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

Briana Morrison, Department of Computer Science

Using Machine Learning K-Means Clustering to Comprehend California Housing Prices

CS4991 Capstone Report, 2023

Anh Nguyen
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
aqn9yv@virginia.edu

Abstract

California is known for its high housing prices, and as interest to live there increases, many people have to try to predict the pricing trend of their future home in order to make sure they are making a good investment. To address this issue, I developed a program that clusters housing regions in California to help the people interested in living there better understand the pricing trends. The program uses the K-means machine learning algorithm on different characteristics from a California housing dataset. The K-means clustering algorithm is better fitted for this problem because it can group the houses by all the different characteristics. This algorithm also helps with comprehension, either because the clusters are based on grouping characteristics relevant to a user or visually presenting graphing correlations to a user. This application only uses one out of many machine learning algorithms. Future steps for development can be using this machine learning algorithm to predict California housing prices. Other future work can also be developing an app that uses these algorithms and expands predictions to other states.

1. Introduction

California has been hub for big technology companies. Specifically, San Francisco and Silicon Valley are known to have one of the highest densities of technology companies [6]. Recently, many young software developers have been looking to live in California, within the hub of these companies. Many people also consider living in California for the weather. All these factors make California an attractive place to live. The downside of living in there is the exorbitant housing prices. Whether it be because workplaces went online and people wanted office space or because people wanted personal space during quarantine, the housing prices increased a lot due to the high demand and limited supply in housing during Covid-19 [5]. This makes clustering housing in California a useful tool when buying a home or investing in property, especially in places like California where the housing prices are already high. This California housing clustering program can also help people who may not understand what characteristics contribute to housing prices. It would give insight into what characteristics are at play and how those characteristics affect housing prices in California.

2. Related Works

Machine learning has been around for some time, and there have been previous studies on using machine learning algorithms to predict housing prices. There are not too many recent studies in the last few years on using K-means to cluster California housing. Previous studies use different models for predicting like the random forest, gradient boosting, or regression models (Calainho et al., 2022). Some studies also focus on housing in different areas other than California, like China (Ho et al., 2021). Studies in the past have shown that machine learning models were able to provide better predictions for housing prices compared to regression models because the models were able to take into account many different characteristics and relate them to each other (Calainho et al., 2022).

Previous studies do not really address using K-means as a model. Other studies that are not housing price predictions have used K-means (Truong et al., 2020). They mention how K-means can cluster data well and accurately produce predictions (Gupta & Chandra, 2021). This study is using K-means in IoT applications and not for predicting housing prices. Overall, it seems that K-means clustering works well when it comes to pattern matching and image analysis [4]. Studies also go into the implementation of the algorithm and using different distance metrics to adjust the clusters. K-means is a form of unsupervised learning, so it would work better in real-world applications that have to do with finding patterns (Chandra, 2020). A common example is using the algorithm to group foods together so customers of a grocery store will spend more money and give the grocery store a bigger profit.

This would require more datasets and the app would also have to take into consideration events relevant during the time periods. More steps should also be taken to make sure the data collection is ethical and the data itself is unbiased.

3. Program Design

This program utilizes the California Housing dataset from Kaggle. After the data is preprocessed, the program splits the data into a training dataset and a testing dataset. Then, the program makes a model based on applying the implemented K-means algorithm to the training dataset. It uses various numbers of clusters to find the one that best groups the data. Lastly, the sum of squares errors (SSE) for each different number of clusters are calculated to analyze the overall performance of the clustering algorithm.

3.1 Review of K-means Clustering

The K-means clustering algorithm is an unsupervised machine learning algorithm that groups data into a number of K clusters based on characteristics in the data. It can be broken up into four steps: assigning the number of clusters, calculating the centroids, grouping the data points, and recalculating the centroids. First, the number of clusters is assigned. The number of clusters is usually decided after preprocessing the data and looking for correlations throughout the dataset. Second, the algorithm calculates the centroids. The centroids can be random, or the developer can choose specific locations for the centroids based on the results from preprocessing the data. Third, the algorithm groups or clusters the data points. It goes through all the points in the dataset and measures the distance between the point and the centroid. Whichever centroid the point is closest to is the centroid it will be clustered into. Fourth, the algorithm will recalculate the values of each centroid. The new value of a centroid is the mean of all the points in its group. Once the four steps are completed, the data points get regrouped again. If any of the datapoints are grouped to another centroid, the recalculating the centroids and grouping the data points steps are repeated. Those two steps repeat until all the points do not change which centroid they are grouped to.

3.2 Program Components

The first component of this program is the California Housing dataset from Kaggle. Kaggle is a well-known and popular platform that many data scientists use to publish or find datasets. The second component of this program is that it is written in Jupyter Notebook which utilizes Python. The Python packages utilized are: pandas, sklearn, numpy, os, matplotlib, and random. The Minkovski Distance is the distance metric used in this program.

3.3 Program Implementation

The program implementation can be broken up into four parts: loading the dataset, preprocessing the data, implementing the K-means algorithm, and clustering the housing into regions.

3.3.1 Loading the Dataset

The first part is the loading the dataset part. This is where the dataset is loaded and other variables are initialized that will be useful for the program in the later parts. All the packages are also imported in this step. In order to load the California housing dataset, it must be downloaded from Kaggle. Once downloaded, it is loaded using the read_csv method from the pandas package which saves the data into a pandas DataFrame. The numpy random seed is initialized to 42 in this step, and the size of the labels for the graph in the later parts is also initialized in this step.

3.3.2 Preprocessing the Data

The second part is the preprocessing the data part. Data entries that had missing values were dropped from the DataFrame. The categorical variables were preprocessed in order to facilitate the prediction process. The categorical variable ocean_proximity was encoded. It was the only categorical variable, and it was encoded with one hot encoding. This process started with finding the different values of the variable. The values were ocean, inland, island, near bay, and near ocean.

The OrdinalEncoder class and the OneHotEncoder class was used for the one hot encoding process. The OrdinalEncoder made a categorical variable list for the OneHotEncoder that had all the different values for ocean_proximity. These classes made a binary array for each data entry where a 1 was in the index that corresponded to the value of the categorical variable and a 0 in all the other indices. The numerical variables were preprocessed using the Pipeline class in the sklearn package. It took in the SimpleImputer class and StandardScaler class to process the numerical variables. Lastly, the preprocessed variables were put through a final pipeline using the ColumnTransformer class from the sklearn package to fit and transform the numerical and categorical variables into one training DataFrame.

3.3.3 Implementing the K-means Algorithm

The third part is implementing the K-means algorithm. There will be one K-means method that has the algorithm, but there will be helper methods within the overarching method to help with calculations. Initializing the centroids is the first step. The number of centroids used will be from 2 to 10. The first centroids are randomly initialized in a helper method based on the dimensions of the preprocessed training data and using the choice method from the random class in the numpy package. After that helper method runs, the clusters are initialized inside a while loop where the condition is a condition variable set to True. The initialization of the clusters is done in another helper method within the while loop. The helper method goes through all the points in the data and each centroid, finding the distance between each and storing all these values. The distance metric used in this program is the Minkovski Distance. The centroid with the minimum distance away from the point is the cluster the point is assigned to. At the end of the method, all the clusters are returned and each cluster contains their respective data points. After, new centroids are calculated by finding the average of all the data points in each cluster. If any of the new centroids are not the same as the current centroids, the condition variable for the while loop is set to False. Otherwise, the while loop starts again and the clusters are recalculated using the cluster helper method and the new centroids. At the end of the K-means method, the centroids are returned with their respective data points.

3.4 Program Analysis Implementation

The analysis for the program will be based on the SSEs for each of the K clusters, and the analysis will also be based off graphing each of the K clusters. The clusters graphing will be done using the matplotlib package from Python. The SSEs will be calculated for each number of clusters. The average and standard deviations for the characteristics of each cluster will also be calculated.

4. Results

The program explored clustering California housing based on a number of centroids from 2 to 10. The centroids and the way the housing regions were grouped for each number of centroids is graphed in Figures 1-4 below. Only four clusters were graphed because the other number of clusters displayed similar results to the following figures despite the different number of clusters. The SSEs of each cluster is displayed in Table 1 below.

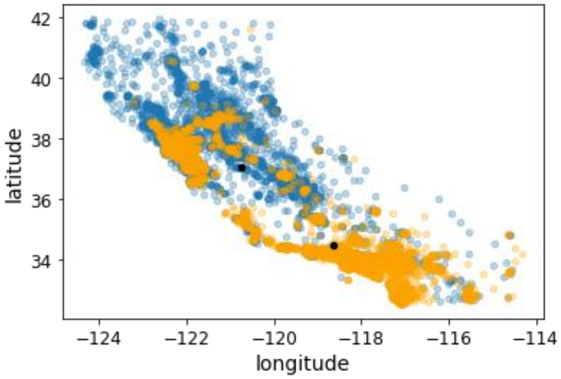

Figure 1: Graph of California Housing Regions Based on 2 Centroids

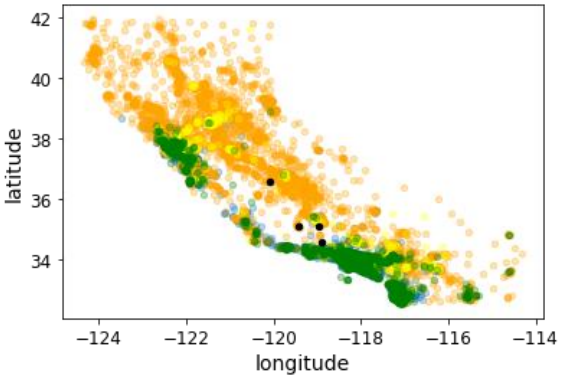

Figure 2: Graph of California Housing Regions Based on 4 Centroids

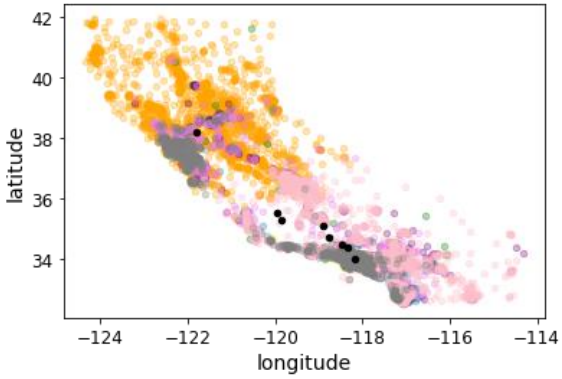

Figure 3: Graph of California Housing Regions Based on 8 Centroids

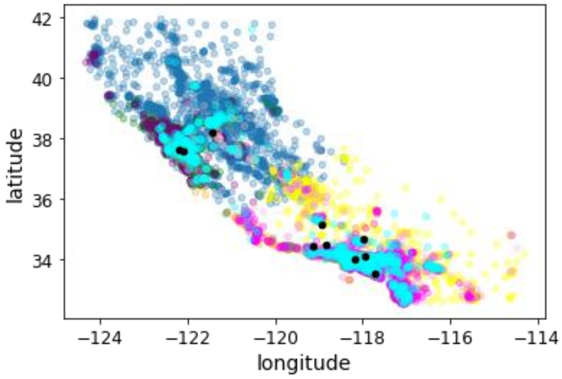

Figure 4: Graph of California Housing Regions Based on 10 Centroids

Clusters	SSE	Clusters	SSE
2	166394.640	7	104461.636
3	147419.066	8	91054.301
4	124759.251	9	87766.803
5	123630.566	10	81253.312
6	103847.319		

Table 1: SSEs for Each Number of Clusters

Based on the figures, it can be seen what characteristics influence the clustering. The centroids that were chosen are represented by the black dots in the graph. These centroids can explain what characteristics go into grouping the housing. When the clusters were 2 or 3, the grouping focused mainly on the location of the house. This was based on inland or bay characteristics. 4 clusters is when more characteristics are looked at. These clusters still look at location, but more characteristics are being taken in like number of rooms and population. With 8 and 10 clusters, it can be seen how the houses are grouped very diversely. With more clusters, the characteristics like price or house value, median income, households, and characteristics from previous clusters that were already accounted for. With more clusters, the more fitted the model will be. Based on the table, it can also be seen that with more clusters, the better the model's performance. Grouping by more clusters was avoided to prevent overfitting the model. A line to the number of clusters must be drawn, otherwise there could be a cluster for every data point which would defeat the purpose of trying to see housing trends in different areas of California.

5. Conclusion

This project helped with further understanding the machine learning K-means algorithm and what factors affect its performance. It was also able to show how clustering algorithms work well with grouping housing data. The clustering housing regions in California program gave insight on characteristics of houses, and the correlation of these characteristics can be used to predict housing prices or further understand the housing market in California. Machine learning algorithms are used in applications every day, and understanding what contexts these algorithms produce the best results in can benefit future developments in this field. California is a popular place to live, and understanding what factors go into pricing houses is useful for people who wish to live there or invest in properties there.

6. Future Work

The next step to improve this program would be using the K-means clustering algorithm to predict housing prices in California. This would require training the model and dividing the dataset into training and testing sets, instead of just clustering the data. Another aspect to consider with this program is changing the dataset to one that includes characteristics like national parks or other landmarks that could affect the price of the house. Lastly, this clustering program can be applied to larger datasets and the number of clusters for those can be adjusted accordingly.

7. Acknowledgements

This project was completed at the University of Virginia in the Department of Computer Science's Machine Learning or CS 4774 class taught by Dr. Rich Nguyen.

References

[1] Calainho, F. D., van de Minne, A. M., & Francke, M. K. (2022). A Machine Learning Approach to Price Indices: Applications in Commercial Real Estate. The Journal of Real Estate Finance and Economics. https://doi.org/10.1007/s11146-022-09893-1

[2] Chandra, A. (2020, July 19). Machine Learning with real world examples. Analytics Vidhya. https://medium.com/analytics-vidhya/machine-learning-with-real-world-examples-3e79877d08b3

[3] Ho, W. K. O., Tang, B.-S., & Wong, S. W. (2021). Predicting property prices with machine learning algorithms. Journal of Property Research, 38(1), 48–70. doi:10.1080/09599916.2020.1832558

[4] Gupta, M. K., & Chandra, P. (2021). Effects of similarity/distance metrics on k-means algorithm with respect to its applications in IoT and multimedia: A review. Multimedia Tools and Applications. https://doi.org/10.1007/s11042-021-11255-7

[5] Riquier, A., & Witkowski, R. (2022). Why are houses so expensive? Forbes Advisor. https://www.forbes.com/advisor/mortgages/real-estate/why-houses-are-expensive/

[6] Rees, A. (2022). Top 10 best cities for tech jobs. Bloom Institute of Technology. https://www.bloomtech.com/article/top-10-best-cities-for-tech-jobs#:~:text=San%20Francisco%20has%20the%20largest,remains%20a%20top%20innovation%20hub.

[7] Truong, Q., Nguyen, M., Dang, H., & Mei, B. (2020). Housing price prediction via improved machine learning techniques. Procedia Computer Science, 174, 433–442. https://doi.org/10.1016/j.procs.2020.06.111