Capstone_final_report

December 17, 2020

1 Harris County Appraisal District House Value Prediction

by: Rafael Pinto Ortiz

2 Introduction

I had the privilege of owning a house. Part of the responsibilities of owing property in Texas is to pay property taxes. In the Harris county, the property tax system has four primary stages:

1. Valuing the taxable property: The Harris County Appraisal District (HCAD) determines the current year value (appraised value) of your property.
2. Protesting the appraised value: The owner is given an opportunity to protest and provide evidence of a suspect appraised value.
3. The officials adopt the present year tax rates.
4. Taxing units collect the taxes.

This work focuses on the first two stages of this process. I present my experience with the system, and a hindsight data driven solution to determine if the first step of the process produced an appraised property value that is fair. Then, I show how to calculate a reasonable appraised value using machine learning.

3 Problem Identification

In 2016 I received the appraised value of my property from HCAD. It totaled $292,707. At first, I was misleadingly happy because I thought my property had drastically appreciated from purchase value, but then I realized it could all be a mistake, and if so, my property taxes for the year would be much higher, since they depend on the appraised value.

As any other home owner would have done, I set up to find properties similar to mine, the same-model houses in the same subdivision, to understand how they were appraised. I gathered these data using a web tool provided by HCAD named parcelview. The following table is the result of that investigation:

House	Bedrooms	Baths	Half Baths	Impr Sq ft	Appraised Value
My House	4	3	1	3,256	$292,707
Comp 1	4	2	1	2,832	$268,084
Comp 2	4	2	1	2,574	$252,786
Comp 3	4	2	1	2,586	$257,056

House	Bedrooms	Baths	Half Baths	Impr Sq ft	Appraised Value
Comp 4	4	2	1	2,587	$259,443
Comp 5	4	2	1	2,598	$257,804
Comp 6	4	2	1	2,659	$253,696
Comp 7	4	2	1	2,619	$256,539
Comp 8	4	3	1	2,668	$259,716

There are a couple of interesting facts in this table:

1. My house received the highest appraised value, and it is also has the largest improvement area (Impr Sq ft column). It is $24,623 more expensive than the next most expensive house (Comp 1), and it has 424 square feet more than the next largest house, also Comp 1.
2. The rest of house features are more or less the same. This suggest that the appraised value driver is the improvement area.

With this information I felt sufficiently armed with evidence that the house was not fairly appraised. How is it possible that these other same-model houses were appraised at least $24,623 less than my own? This was all the information I provided in my protest. The friendly receiving appraisal agent reviewed my protest and quickly realized a problem with the data, and my overly appraised house value problem was immediately resolved. In the spirit of preventing spoilers, I'll discuss what the agent saw in the final part of this report.

At the time I made a note to myself. I had the data, and a way to get more data if needed through the parcelview portal. I wanted to automate the fairness estimation of the appraised value, and I wanted to add technical rigor to the calculations. However, I did not had the knowledge nor the time to keep working this problem, and since my appraised value was back to normality, this matter stayed dormant, waiting for future me to dig it up and solve it in a more elegant fashion.

Fast-forward to 2020. The year I started my data science journey. Despite of being trained in STEM, statistics, like linear algebra, is one of those subjects I have had to study and re-study every time I'm trying to solve a problem where they are needed. This year I re-learned how to perform statistical tests, and enhanced my model building skills with those offered in my Springboard class.

As part of my curriculum, I have to complete a couple of capstone projects, so I selected my HCAD appraised value problem from 2016, to test my knowledge and programing skills. In sum, this project tries to answer two questions:

1. **Was my house fairly appraised in 2016**
2. **If not, then what is a fair value?**

I'll answer the first question with a hypothesis test, and the second with a machine learning model.

4 Data Wrangling

Let's start by saying that I'll be using more than eight houses as my data. I knew HCAD exposes the appraised value data on the parcelview portal, so I started digging to find if they had an API or a database with the same data, but machine friendly. They do. It is on their public data site.

I built this project's repository (hcad_pred) with reproducibility in mind. I use `invoke` to run scripts from the command line and `papermill` to run the final Jupyter notebooks. Going forward

in this document, I'll provided the invoke commands to complete the task at hand.

4.1 Data Download

I downloaded the data with the scrip: `/scr/data/download_hcad_data.py` This can be run with the invoke command: `invoke download-data`. This script fetches all the 2016 data in HCAD and saves them to the project directory: `data/external/2016`

4.2 Data selection

The downloaded data is numerous and comes in multiple files, representing different kinds of data. The data preparation is documented in the suite of notebooks `notebooks/01_Exploratory/1.[0-6]`. In the first five notebooks, I inspect, decode, and clean the columns, and then I run a summary statistics for each. The resulting cleaned version of the data is saved to `data/raw/2016` directory. The sixth notebook on this series is for joining the results of the previous five into a single dataframe: `data/interim/2016/comps.pickle`

Here is a summary of what each data selection notebook does:

1.0: Input data: `Real_building_land/building_res.txt`. This file contains some of the properties descriptions like areas, date erected, quality description, property use code, and percent built. Also, this file contains the HCAD account number associated to each property. In this notebook, I selected the features that represent physical properties of the houses, and filter the properties to contain only Residential single-family homes. I also exported these single-family home account numbers as a stand-alone file that I will use to filter the accounts on load in subsequent notebooks.

1.1: Input data: `Real_building_land/fixtures.txt`. This file contains property features like number of bedrooms, full baths, half baths, and more. It comes as a melted table, so we need to use the pivot_table method on the dataframe instance to shape it to a table with one observation per row (HCAD account number). I selected the ten most common features in the properties to prevent columns mostly filled with missing values.

1.2: Input data: `Real_acct_owner/real_acct.txt`. This file contains property information like total appraised value (the target on this exercise), neighborhood, school district, economic group, land value, and more.

1.3: Input data: `Real_building_land/extra_features.txt`. This file contains property information like number and quality of pools, detached garages, outbuildings, canopies, and more. Similar to the fixtures file, I selected the 15 most common features to minimize the number of missing values.

1.4: Input data: `Real_building_land/exterior.txt`. This file contains property information on the areas of the sections of the improved area (base area pri, base area upr, open fram porch pri…). Similar to the fixtures file, I selected the 10 most common features to minimize the number of missing values.

1.5: Input data: `Real_building_land/structural_elem1.txt`. This file contains property information about the building data, like foundation type, exterior wall composition, Heating/AC, and more. Similar to the fixtures file, I selected the 7 most common features to minimize the number of missing values.

1.6: This file joins the output of the last six notebooks into a single dataframe, and exports it to `data/interim/2016/comps.pickle`.

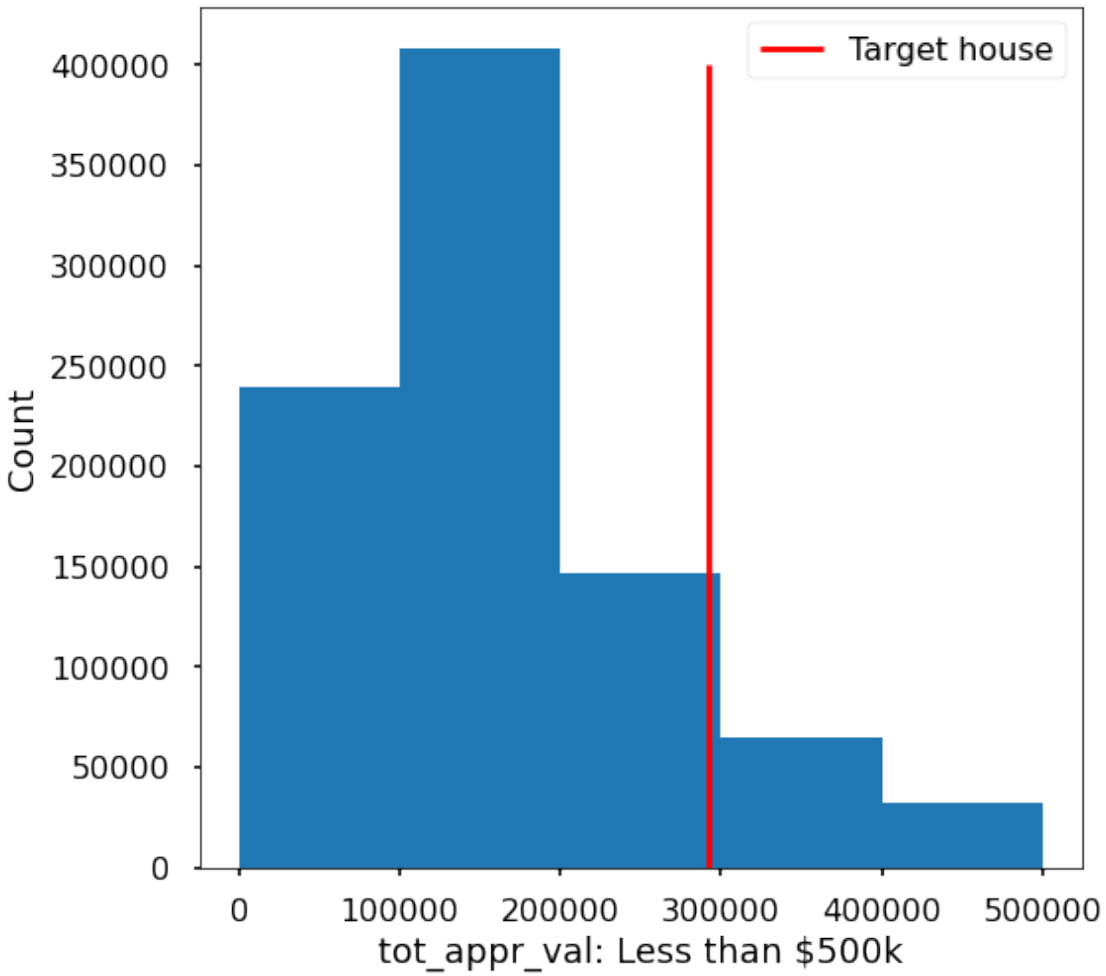
The resulting merged dataframe has 957686 properties with 87 features.

5 Exploratory Data Analysis

Notebook **2.2** shows the Exploratory Data Analysis that I performed. The first step was to look at the distribution of appraised values:

Descriptor	Value
count	9.559880e+05
mean	2.191061e+05
std	2.800815e+05
min	1.000000e+02
25%	1.000000e+05
50%	1.501055e+05
75%	2.342272e+05
max	1.726682e+07

For the Harris county single-property homes, the mean value is close to USD 219K, and the highest valued house is roughly USD 17M. This large variation obfuscate the distribution when presented as a histogram, so I'll zoom in the plot on the houses valued less than USD 500K.



The target house is on the high tail of the distribution of all single-family houses in HCAD. There is a wide range of property values given their location (neighborhood), physical condition, renovation, and other factors. Let's calculate the year-to-year percent change in property value, in an attempt to minimize the effect of the property's appraisal value magnitude.

`yy_pct_change` = (`tot_appr_val` - `prior_tot_appr_val`) / `prior_tot_appr_val`

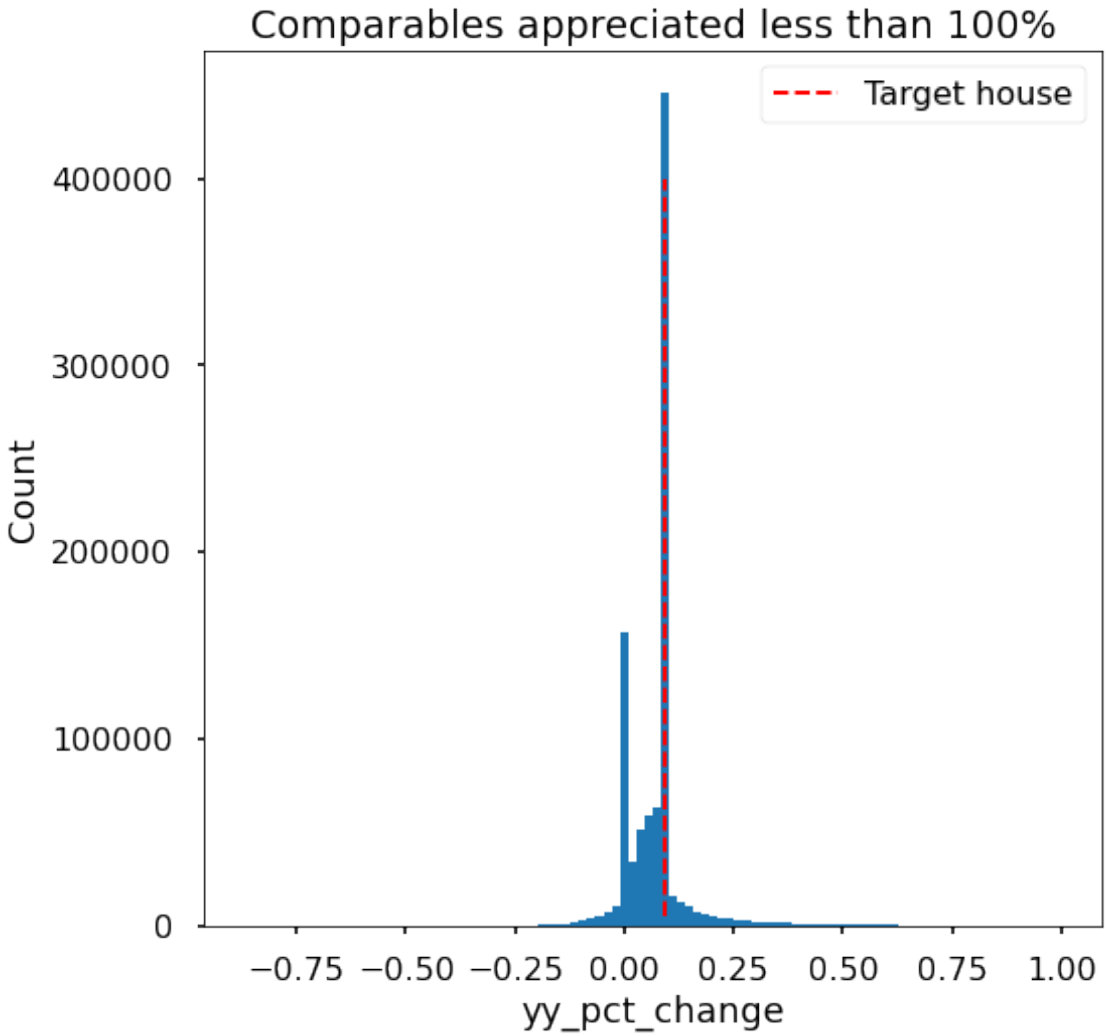
Descriptor	Value
count	955988.000000
mean	0.165985
std	8.581099
min	-0.870447
25%	0.030658
50%	0.099989
75%	0.099999
max	7151.941176

There are valid reasons for a property value appreciation from year-to-year to be multiples of the

initial value, for example, when a new house is constructed on an existing lot, the previous year value will be the value of the land alone, and the current year value will be the value of the land plus the value of the improvement (built house). Similarly, when the existing house in the prior year is a tear-down, we can expect the value of the property to increase significantly if the owner has removed the old house and built a new one on its place. This is fairly common on neighborhoods with new constructions restrictions, like The Heights.

Also, we can expect a large value increase if there has been a major remodeling that either increased the number of rooms or baths (fixtures), common on fixer-uppers, but this work generally increases the value of a property by a fraction, and not whole multipliers. Finally, I think these large (7151%) appreciation values could be errors in the data, or properties that sold under the 10% cap benefits in very expensive neighborhoods. After the sale the cap no longer applies, and thus the property is assessed to its true market value.

The `prior_tot_appr_val` feature is accessible from the 2016 HCAD data. This appraised value corresponds to HCAD's appraised value for the property in 2015. However, there is no feature in the 2016 data that tells us about the percent completion of the property, in case it was still being built. Ideally, we would like to compare the appraised value of houses that were already built in 2015, so we only look at value changes mostly due to market changes, but for this we would need to download and process the 2015 data separately, which is out of the scope for this project. For this reason, the current `yy_pct_change` calculation contains relatively high changes in value, that I clipped to 100% in the following figure.



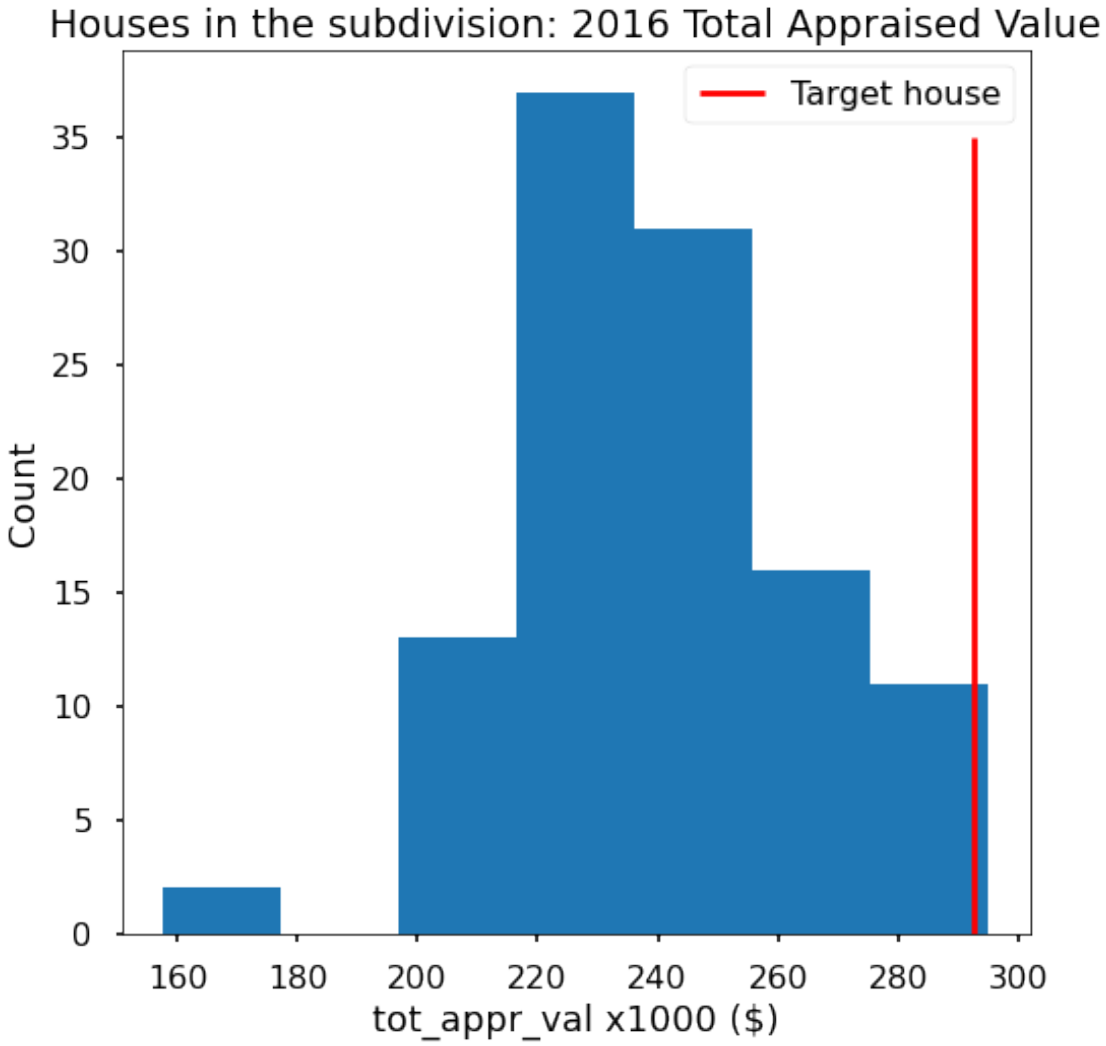
This is interesting. Most properties appreciated from 2015 to 2016, with the vast majority gaining about 10% in value. There is a spike in the 10% value increase mark, which is possibly related to capped appraisal values, where the appraisal district will cap the value of an existing property if the current year appraised value is more than 10% of last year's. Also, the property has to be homestead exempt for the current and prior year to be eligible for this benefit. We can see in the histogram that about 445000 properties qualified for this capping rule. Neat! There is a lesser spike near 0 % value increase, but its origin is less clear.

5.1 Select the subdivision properties

The histogram on year-to-tear percent change appraise value above shows that most properties appreciated up to 40%, and depreciated down to 25%. While this is insightful, the spread is too broad as it accounts for almost all properties in the district, with all sorts of conditions, year built, neighborhood, and many other variables, and as a result, the target house is well within this distribution.

I selected similar properties to my target house by counting only properties in the same subdivision to narrow down the variability in the home values. By selecting these houses, there

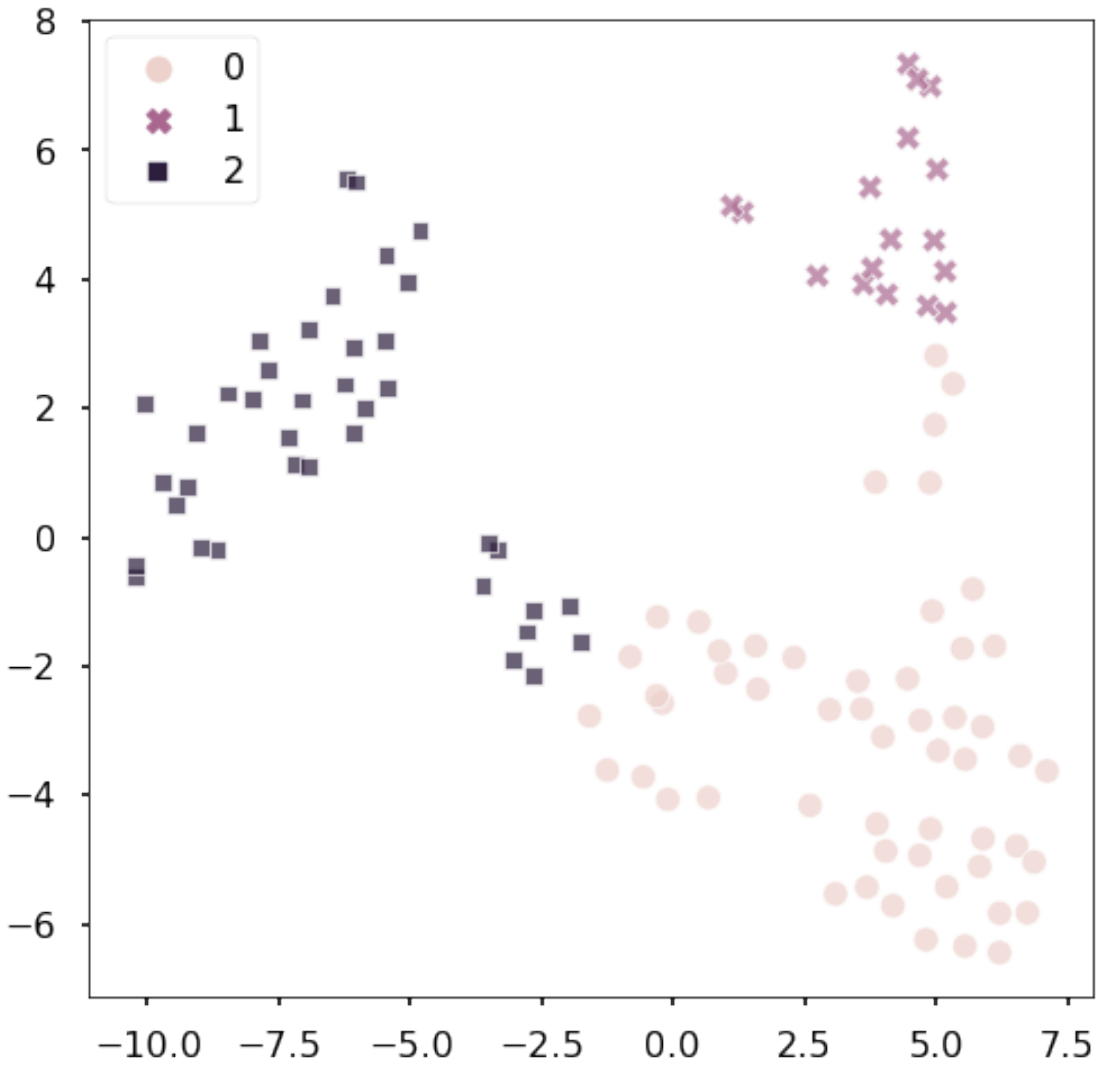
are a lot of categorical columns in this subset that are single valued (e.g. `neighborhood_code`, `market_area_1_dscr`, etc.), so I removed those as well. As a result, the subset of subdivision houses has 111 samples, with 32 features. The histogram of these houses' appraised property values is below.



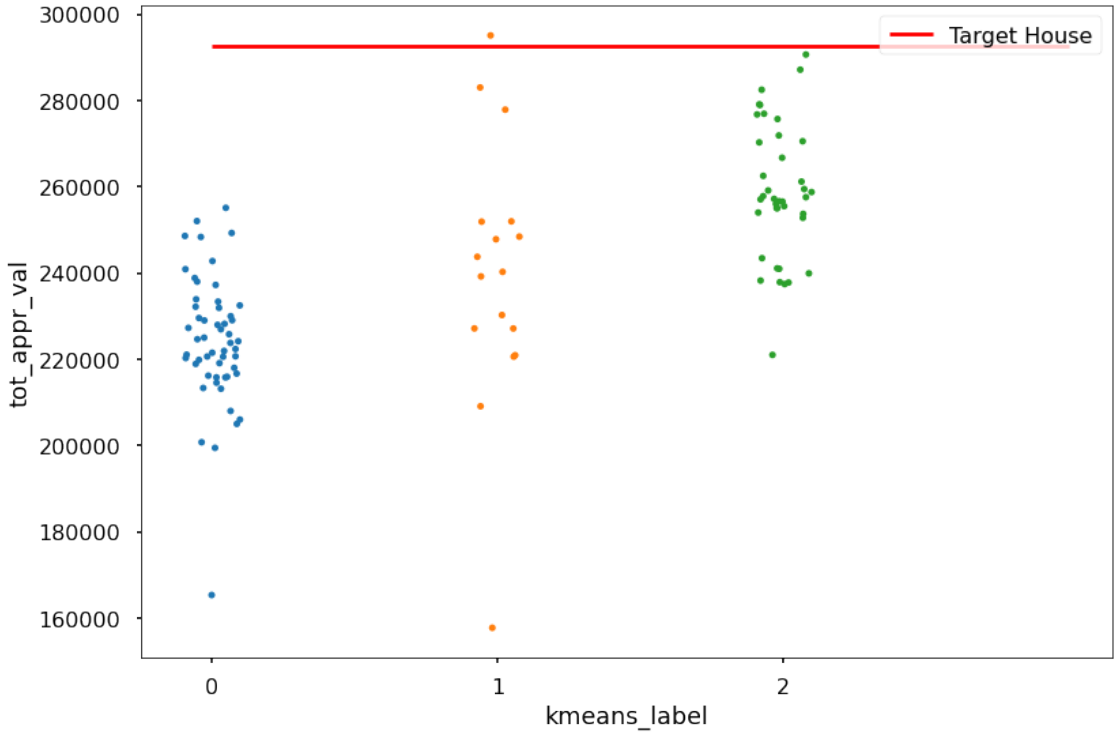
5.2 Find subdivision comparables

In the last figure, we can see that the target house sits on the high end of the distribution of appraised values for the houses in the subdivision. To make a fair comparison, we should find the properties with similar characteristics to the target property, i.e. the comparables. I used the KMeans method (unsupervised learning) to find the houses grouping.

For K-Means to work, in theory, we should remove the non-numerical features. In addition, I removed the features that represent monetary value ($) in such way to build the grouping based only on the physical characteristics (areas, number of rooms, baths, half-baths…) of the properties. After some trial and error, I found that the number of clusters (k) is best set at 3, so every cluster has at least 17 samples. Then, I plotted the selected labels using TSNE to find if the properties are well separated using KMeans.

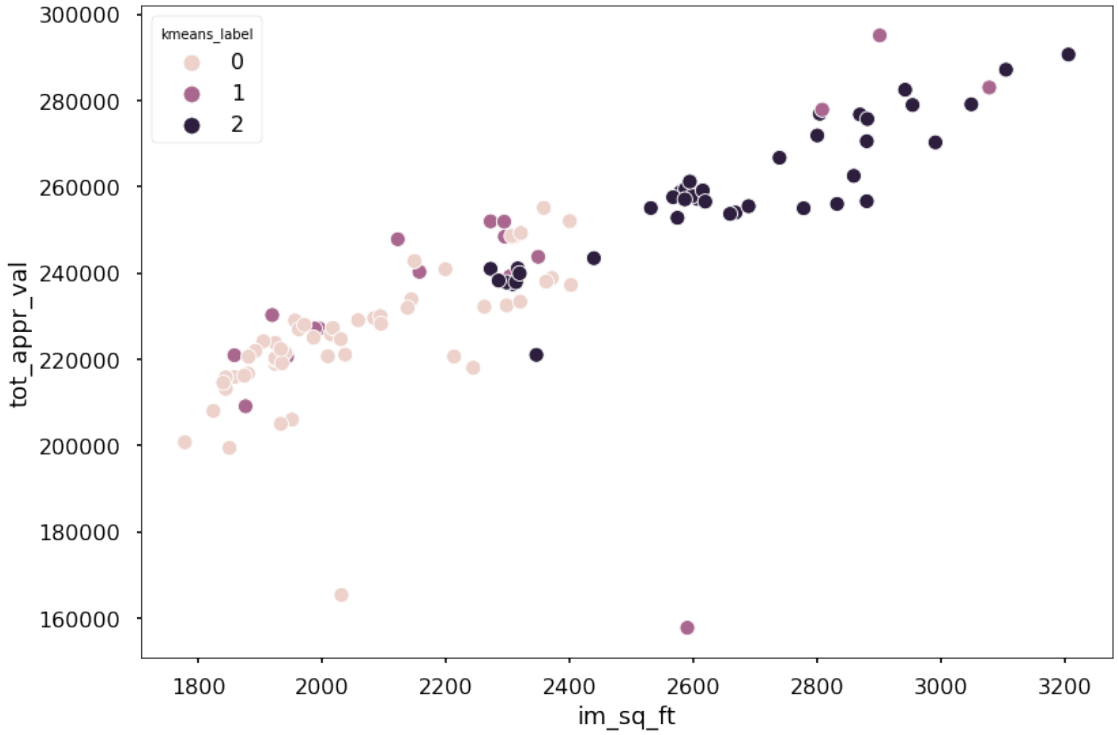


This figure shows that the clusters are well separated indicating that these groups are meaningful. Next, I show the distribution of appraised property values split by comparable houses.



This figure indicates that the target house is the most expensive of its group (group 2), and the second most expensive in the neighborhood. This is in contrast to my home buying experience, where my property was the third largest (and less expensive) property model in the subdivision. This is hinting that there is a problem with the appraised value, or the data used to calculate it.

By plotting the improvement area (`im_sq_ft`) vs. the total appraised value we can see that the k-means groups 0 and 2 represent houses with large area and value, and houses with low area and value, respectively. Also this figure shows the strong correlation between home area (improvement) and value.



5.3 Null-hypothesis test

Group 2 represents the houses that are most similar to my target house, based on the properties' physical characteristics. There are 39 houses in this group. With this data, let's define the null-hypothesis test as follows:

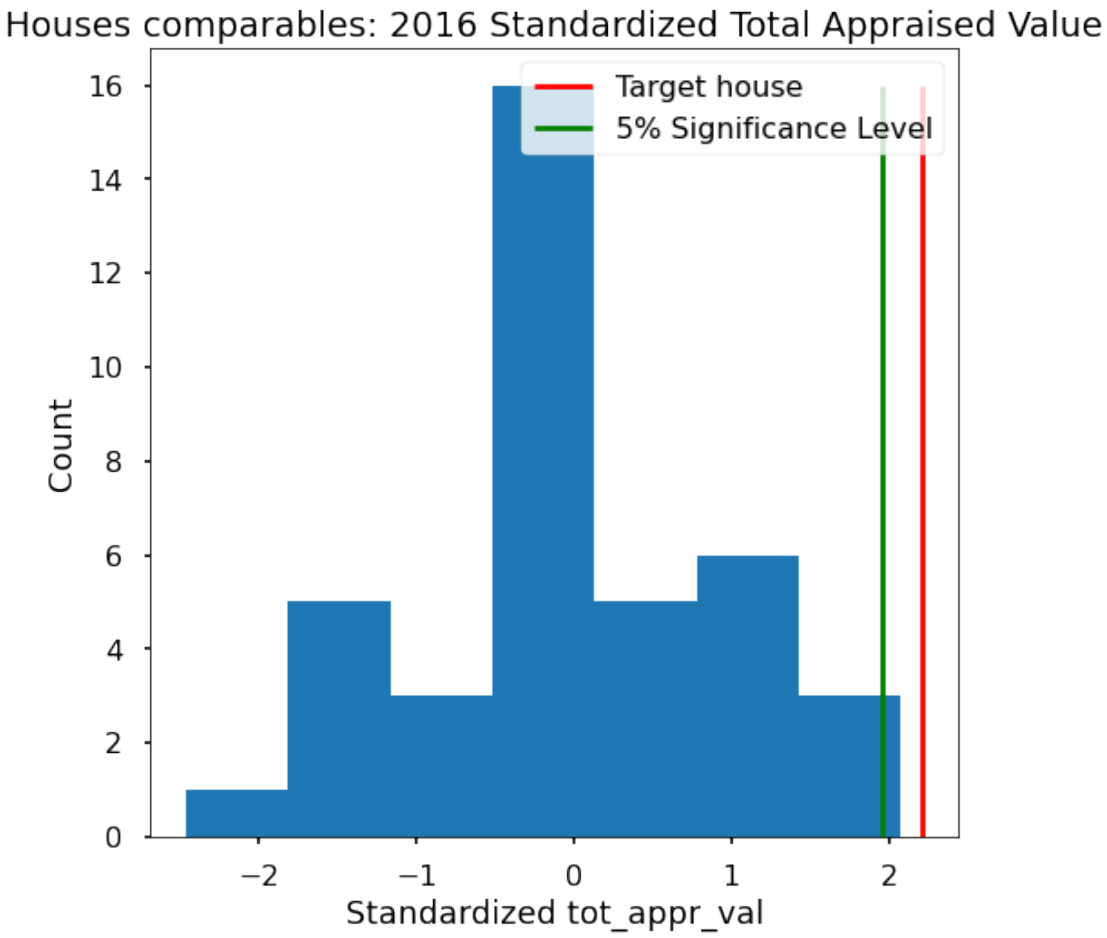
Population: Comparable houses (group 2) appraised by HCAD in the target subdivision. **Significance level**: 5%.

H_0: The 2016 appraised value for the target house was fair relative to its comparables.

H_a: The 2016 appraised value was unfair, or it doesn't belong to the comparables distribution.

The population mean for group 2 is USD 258,729 with a standard deviation of USD 15,374. The appraised value I received from HCAD was USD 292,707, which corresponds to a z-score of 2.21, and a p-value of 0.014. Since this is below the selected significance level (0.05) we can say there is reasonable statistical evidence to reject the the null hypothesis, and thus we have statistical evidence to say that the property appraisal was unfair relative to its comparables.

In other words, if the null hypothesis is true, there is a 1.4% probability of getting an appraised value as extreme as USD 292,707 or lager. Since this 1.4% is below the 5% significance level (z=1.96), we reject the null hypothesis. These values are best visualized on the standardized distribution of appraised valuesfor properties in group 2:



Now that we have found that there is something wrong with the appraisal value of the target property, let's turn our attention to predicting what would have been a better value, based on the subdivision houses subset.

6 Pre-processing, Training Data Development, and Modeling

Notebook **3.1** shows the preprocessing steps, training data development, and modeling approach that I applied to the HCAD subdivision data. In this case, I'm using the 110 houses found in the subdivision, not including my target property. The first step was to remove any feature associated with the previous year value, since several houses were still being built in 2015, leading to inflated percent changes in value. the impacted columns are: `yy_pct_change`, `prior_land_val`, and `prior_tot_appr_val` (we discussed this at the beginning of the EDA section). I also removed the `new_own_dt` (latest purchase date) and `lgl_1` (legal house lot name), as I didn't think these features will impact the appraised value.

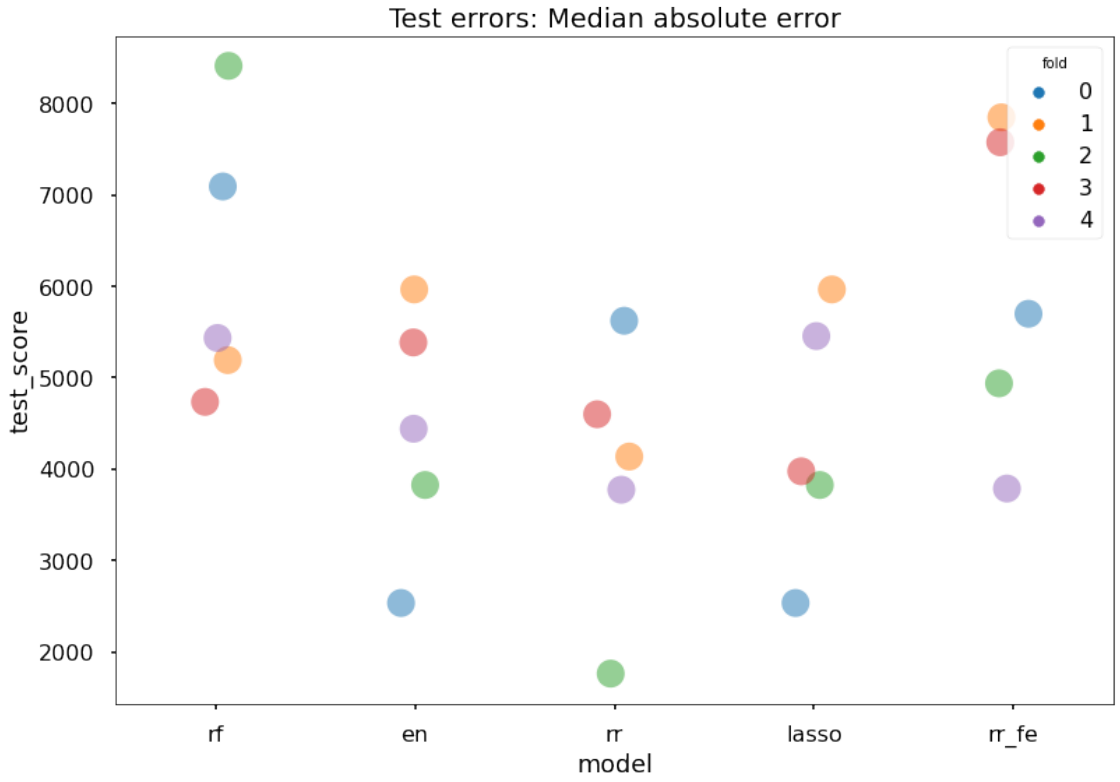
Next, I created the dummy features for categorical variables. Since I need to apply this step to the target house after modeling, I decided to use sklearn's `OneHotEncoder`. This allows to save the encoding parameters from the training data, and apply them on new observations. I followed this same principle in the data standardization step, using `StandardScaler`.

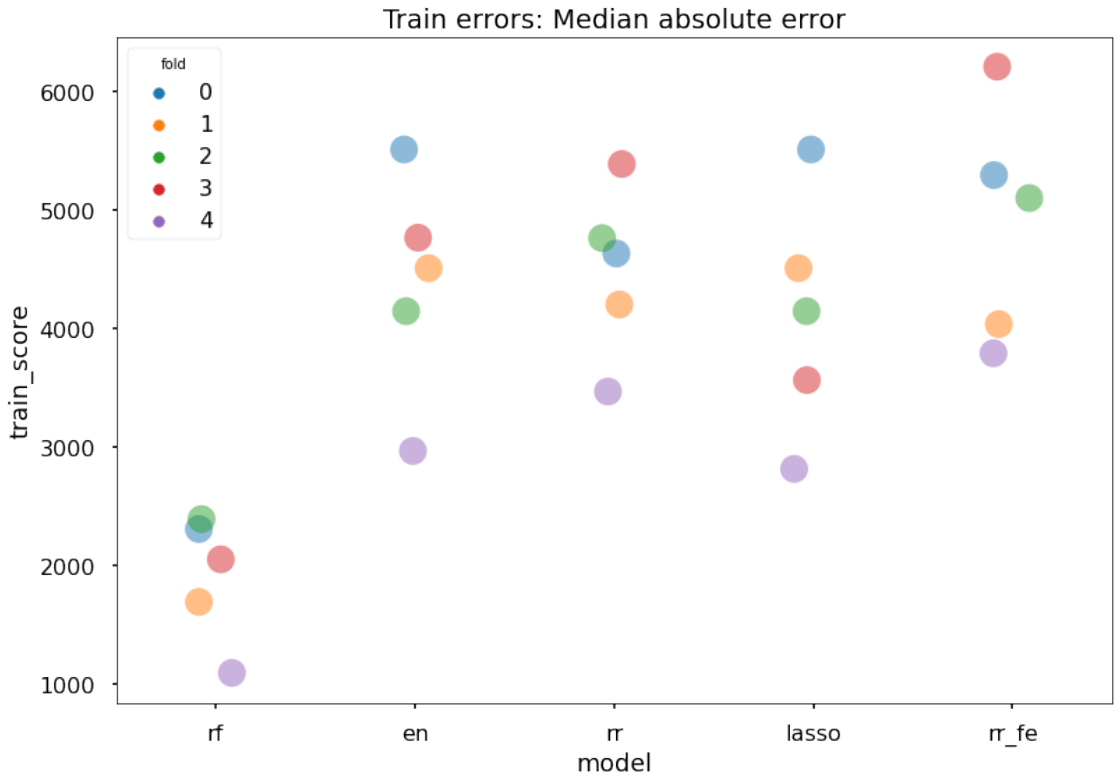
6.1 5-fold cross-validation, feature selection and modeling

I'll follow a variant of the steps on page 245 of Hastie, Tibshirani, and Friedman (2009). The idea is to perform all data processing steps that depend on all the samples within each fold in the cross-validation. This way, we minimize the possibility of information leaking into the training set, which could cause artificially low test errors. A great explanation of the step-by-step procedure is presented in Raschka 2016 blog post.

In this case, I first split the data and use the training data (70% of all the data) in cross-validation to: 1. Fit a random forest estimator (rf) 2. Standardize the variables 3. Fit an ElasticNet estimator (en) 4. Fit a ridge regression estimator (rr) 5. Fit a LASSO estimator for feature elimination (lasso) 6. Fit a ridge regression estimator with features passed by LASSO (rr_fe)

By summarizing the model scores across fold (cross-validation error) we can form an idea of the prediction error on unseen data, and the stability of the hyperparameters for each model. I chose to as a scoring function the median absolute error, since I suspected a few outliers to be present in the data. the cross-validation errors are presented in the following figures.

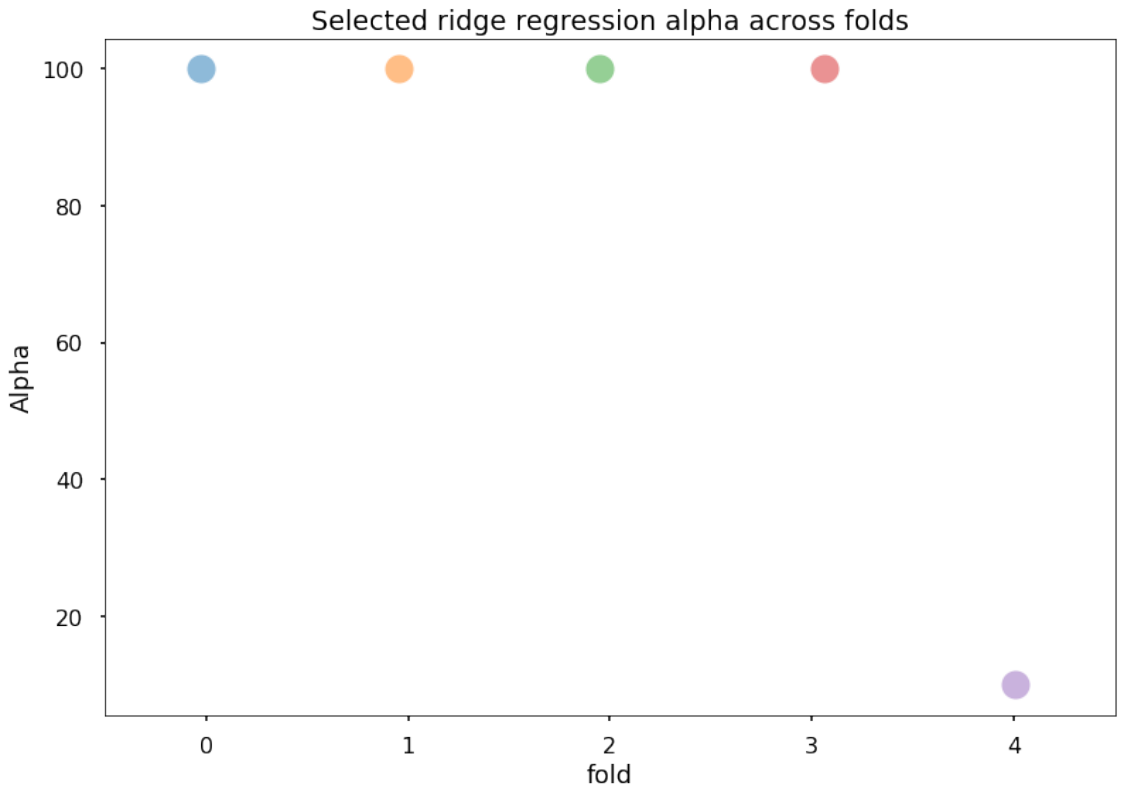




Ridge regression produces the lowest test errors. Overall, LASSO, ElasticNet, and ridge regression have a similar cross-validation test errors, while random forest and ridge regression with feature elimination are the worst performers. Note that the random forest estimator has the lowest train score and the highest test scores suggesting that is overfitting. Base on these plots I selected the ridge regression estimator, as it produces the smallest maximum and the smallest minimum cross-validation error.

6.1.1 Hyper-parameter tuning

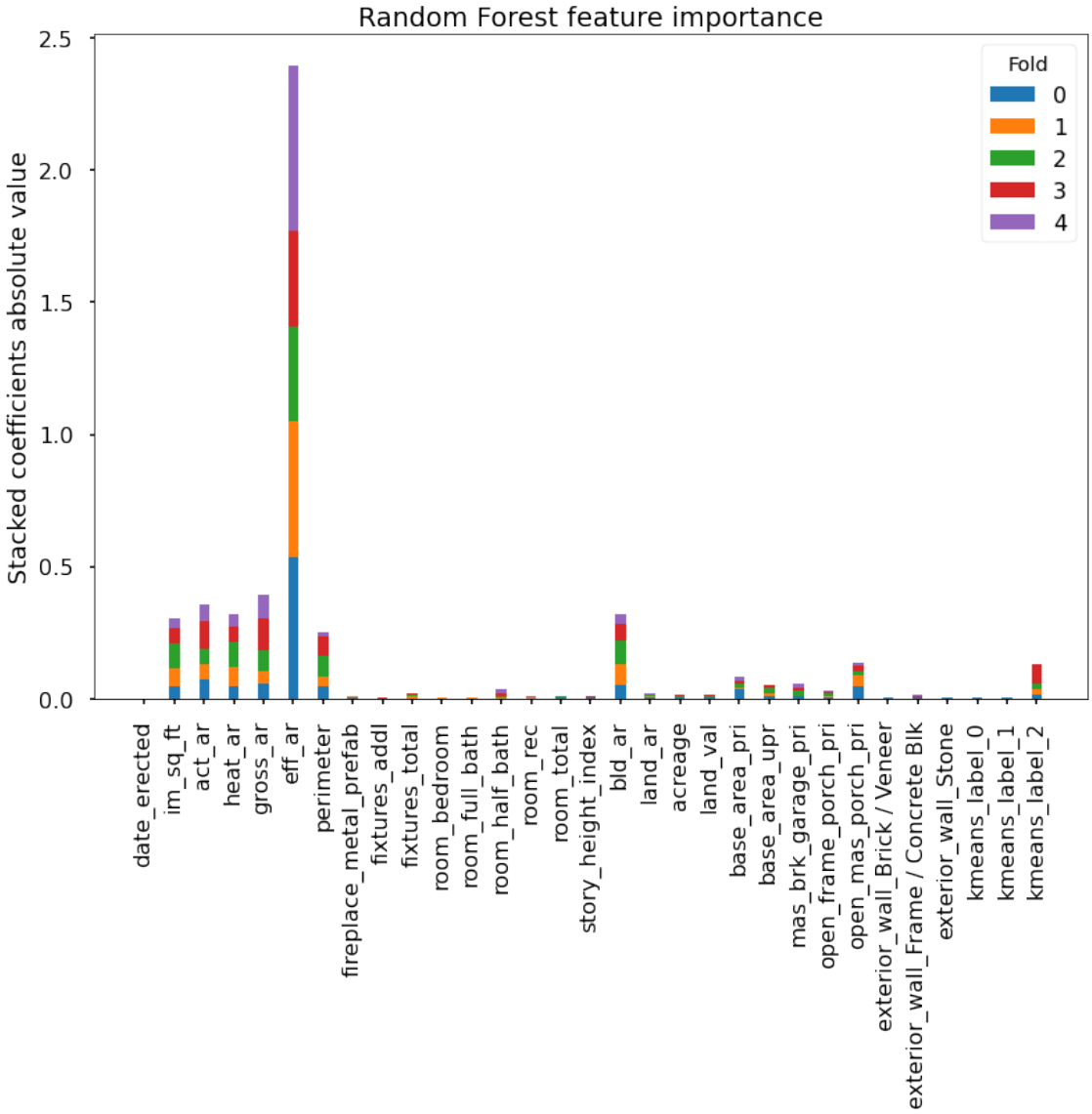
Now, let's look at the stability of the ridge regression parameter alpha.



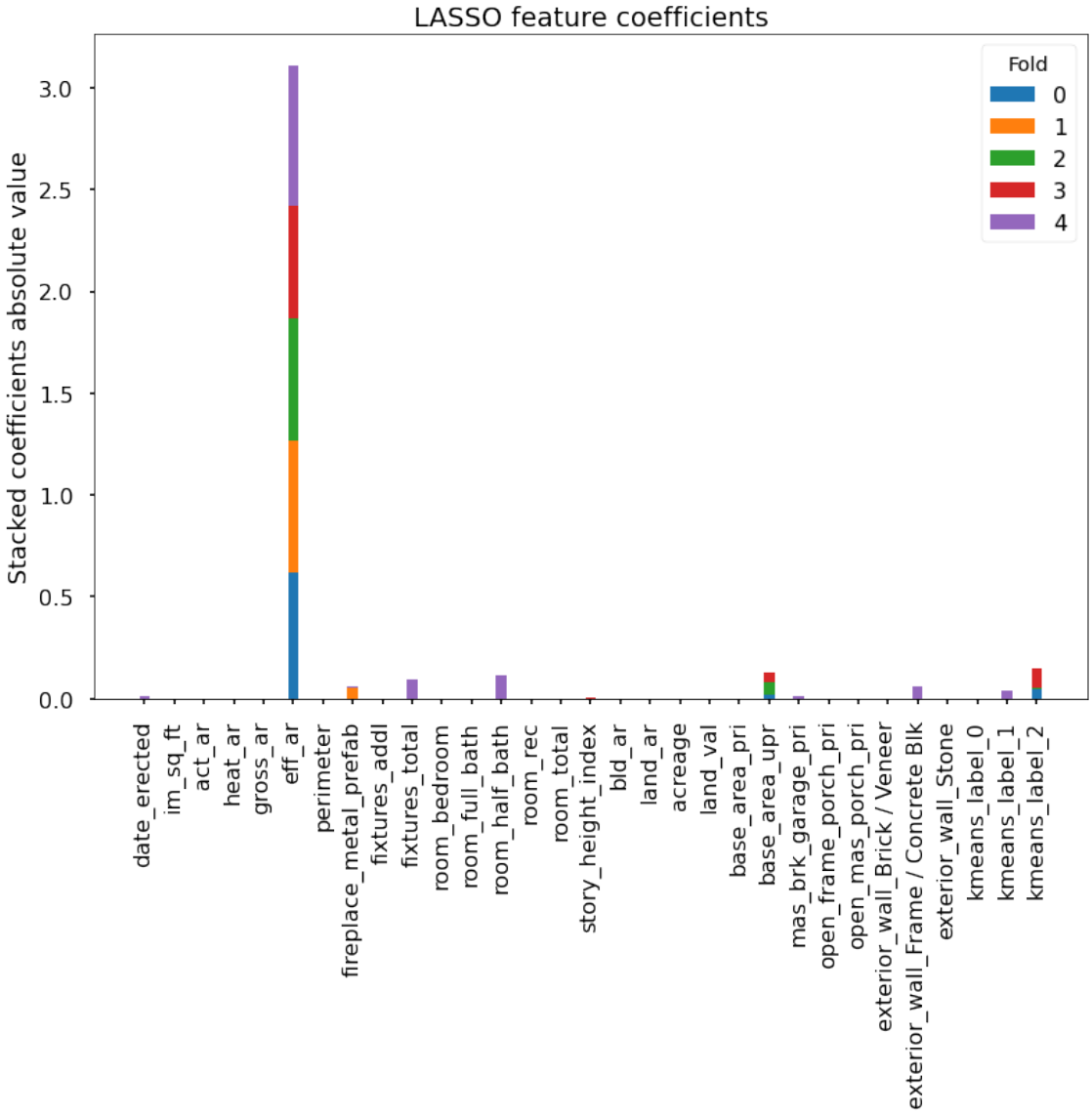
These alpha hyper-parameters were selected using the cross-validation version of the ridge regression estimator (`RidgeCV`) inside each fold, effectively running a nested cross-validation scheme. The plot above suggest that an alpha of 100 is appropriate for 80% of the data (four folds out-out-five), so I set alpha to 100 in the subsequent model training.

6.1.2 Feature elimination

Let's look at the feature importance returned by the random forest estimator and compare them to the LASSO features coefficients.



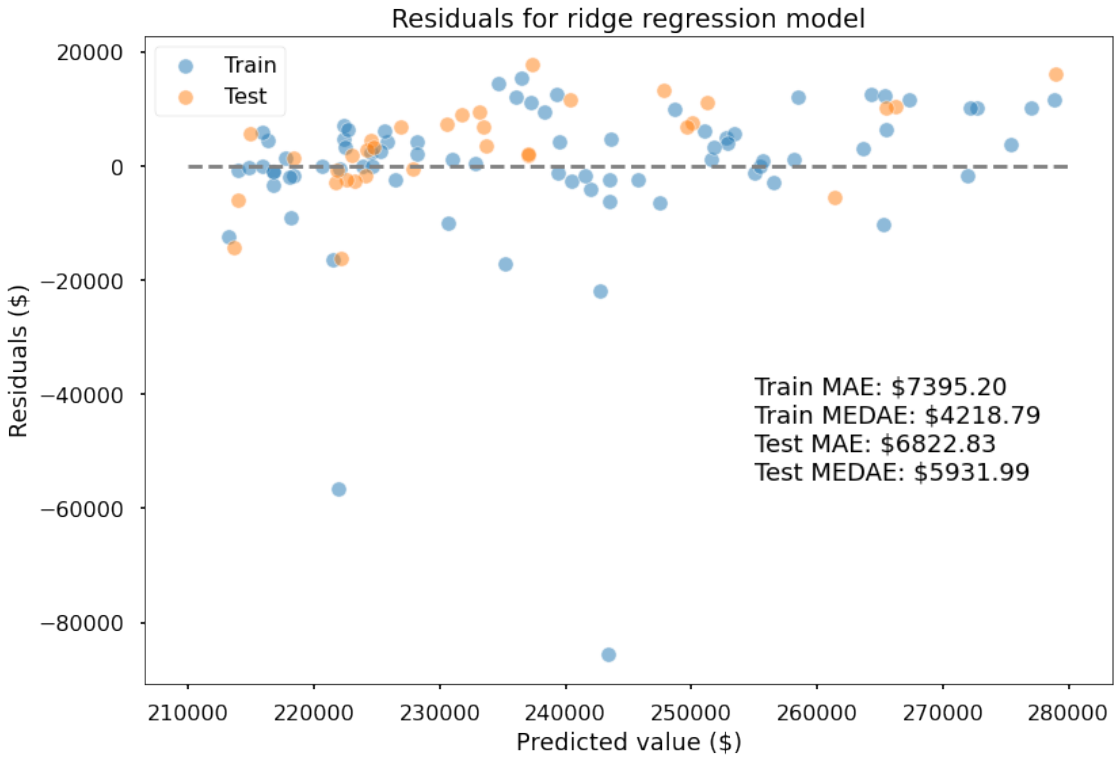
By far, the most significant feature is the effective area (`eff_ar`). The other area related features have a significant contribution each, except for the land area.



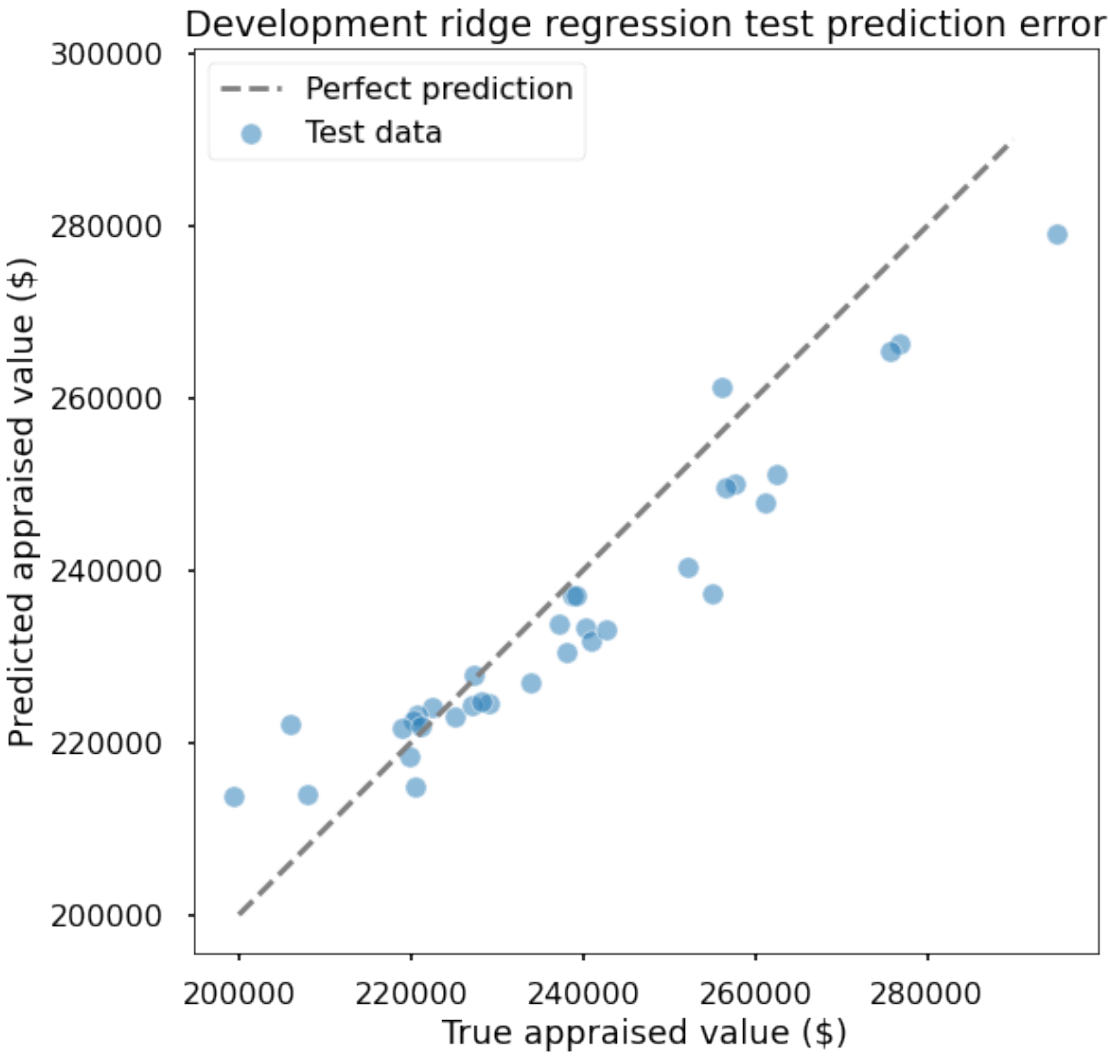
As in the case with random forest, the effective area is the dominant feature in the LASSO estimator. Nonetheless, the ridge regression estimator after feature elimination is out performed by the ridge regression with all the features, suggesting that even though the contribution from the other variables is small, it helps reduce the test error.

6.2 Fit development ridge regression

I determined that ridge regression was the best estimator, and that the penalty term alpha value of 100 was preferred by 4 out of the 5 folds in cross-validation. Next, I built a ridge regression estimator using alpha equal to 100 and all the data on the train set, and then scored this model with the test set (30% of the subdivision houses), unseen yet.



On the test data, the majority of the residuals are positive, indicating that the model tends to underpredict the property value, USD 5,932 on average according to the test set MedAE score.



Like we saw on the residuals plot, the prediction error plot above suggest that most of the properties values are underpredicted.

6.3 Fit production ridge regression

Finally, I built the production model (ridge regression, alpha set to 100) using all the data (110 subdivision houses) to predict the target house appraised value. After dropping the same features and applying the same standardization scheme to the target house data, the predicted appraised value is: **USD 258,696**

7 Conclusions

The intent of this project was to determine if the 2016 appraised value of my target property (USD 292,707) was fair. By performing a hypothesis test using the target house comparables, I showed that the HCAD appraised value was well above the distribution with a p-value of 0.014. Next, I found a data driven property value of USD 258,696 by building a ridge regression model using all the properties in the neighborhood.

In 2016, I protested the appraised value of my house using 8 neighboring properties from the same builder model as a value reference. The HCAD appraisal agent quickly realized that there was a mistake on their database, where my property showed an improvement area of 3,256 sq. ft, but it should have been 2697 sq. ft. She promptly explained that these areas are automatically estimated with aerial photography, and that this could have been the source of the error.

After this fix, the updated HCAD appraised value was settled at USD 263,704. The difference between the settle amount and my model prediction is USD 5008, which is remarkably close to the expected USD 5930 of median absolute error for the test set on the development model. As expected, my model slightly under-predicted the actual value.

Thus, the first simple check to apply if you suspect that your property has been over-valued relative to your neighbors is to verify that the property's area hasn't changed from last year. Even in this case, the workflow developed in this study could be use to predict an independent property value, that can be filed in the protest to HCAD.