Zillow prize: Zillow's Home Value Prediction

BST 691 Final Project Paper

Ye Cai

Abstract

The aims of this study is to predict the log-error between Zillow's Zestimate and the actual sale price, given all the features of a home. But in the study, in order to better prove the model best fitting the dataset, I will train the model based on dataset which included house information sold in 2016, then I predict the log-error of dataset including house information in 2017. Finally, I compare the predicted logerror with the actual logerror in 2017 dataset. RMSE, Root Mean Squared Error, is used to compare the performance of the predicted logerror and the actual one in 2017. The features related to a house, from 2016 to 2017, has been annually documented by Zillow Research website. Lower RMSE means that the model fits the dataset better same as Zillow's data scientist. Machine learning algorithm, eXtreme Gradient Boosting is used to predicte the logerror in 2017. After tuning the parameter, this machine learning works well when fitting the test data set, which achieves relatively low RMSE, equal to 0.09894484.

Introduction

Zillow research has always periodically collected all data related to real estate transitions in different states in the United States, and then produce published data in more comprehensive, reliable, timely order. And people could use those published datasets to do some re4search.

Catalogue

Dataset Explanation

The dataset I downloaded are from Kaggle competition:
https://www.kaggle.com/c/zillow-prize-1
I combined useful features extracted from "properties_2016.csv" and "train_2016_v2.csv" those two files as my training dataset. There is a common unique feature, "parcelid" in those two datasets, and I will use this common column ass an index to combine those two data frames together for feature engineering and modeling. In the "properties_2016.csv", there are all the properties with their home features for 2016, like the number of bathrooms, the number of bedroom and so on while there are specific transaction date and logerror on "train_2016_v2.csv" file. In more specific, there are in three different countries in the United State, Los Angeles, Orange and Ventra, California data in 2016. And in order to make further diagnosis of the accuracy of a model, I will use "properties_2017.csv" and "train_2017.csv" as my test dataset. I will use the model trained by the training dataset to predict the logerror between Zillow's Zestimate and the actual sale price, and then compare the predicted logerror with the actual one.

All statistical analyses will be performed in R. There are multiple features in the "properties_2016.csv" file, in order to better understand the feature, there are brief introduction about some important variables which are selected by feature engineering:

a. basedmentsqft: finished living area below or partially below ground level;
b. bathroomcnt: number of bathrooms in home including fractional bathrooms;
c. finishedsquarefeet6: base unfinished and finished area;
d. finishedsquarefeet12: finished living area;
e. finishedsquarefeet13: perimeter living area;
f. finishedsquarefeet15: total area;
g. fireplacecnt: number of fireplaces in a home(if any).

Data Preprocessing

Firstly, I preprocess the variable in "train_2016_v2.csv" file. Because all observations are from different months on 2016, I divide all observations into twelve months. Then when taking a glimpse of "properties_2016.csv" file, I found the two columns, "fireplaceflag" and "taxdelinquencyflag" are Boolean type with a factor type. In order to make machine learning algorithm works, I change them into binary numeric type.

Features Engineering

In this part, my goal is to transforma raw data into some understandable format. Firstly, after preprocessing the variable in "transactions" data frame, I plot the trend of the sum of transaction on 2016 grouped by months. From the Figure 1, there are many successful transactions from Mar to Sep on 2016, but there is an decreasing trend in the last three months.

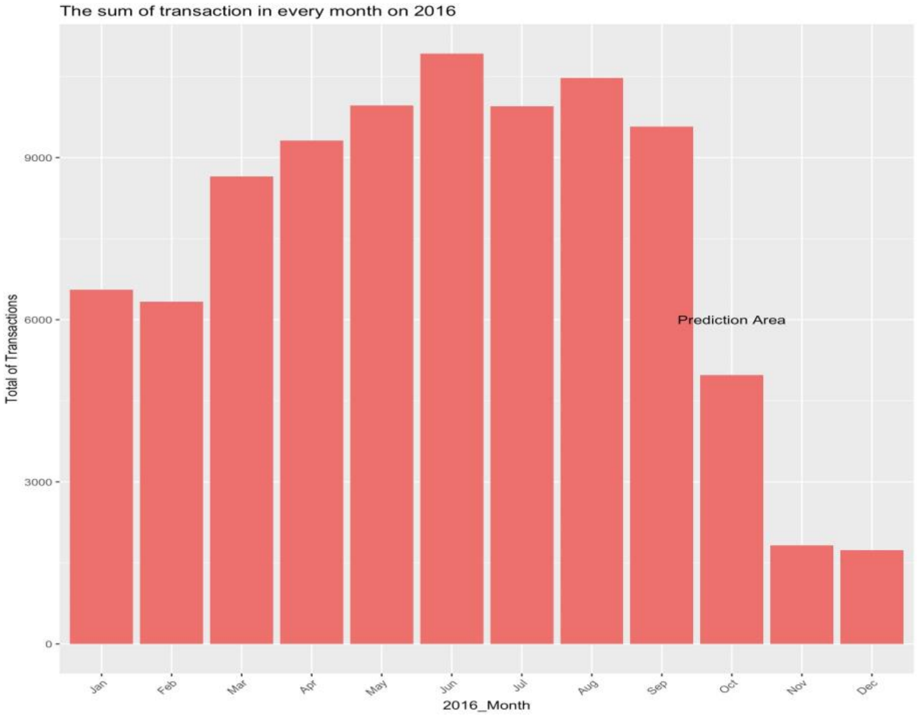

Figure 1

Before I train my own model, it is better to access the performance of Zillow's prediction on the actual price on 2016. When plotting the distribution of the difference of logerror in "transaction", logerror close to zero means prediction is good. To be more specific, a positive logerror means Zestimate overestimated the value of the house while the negative one underestimated. From Figure 2, I could see that most logerror is relatively near to zero, which proves that Zillow has good data scientist.

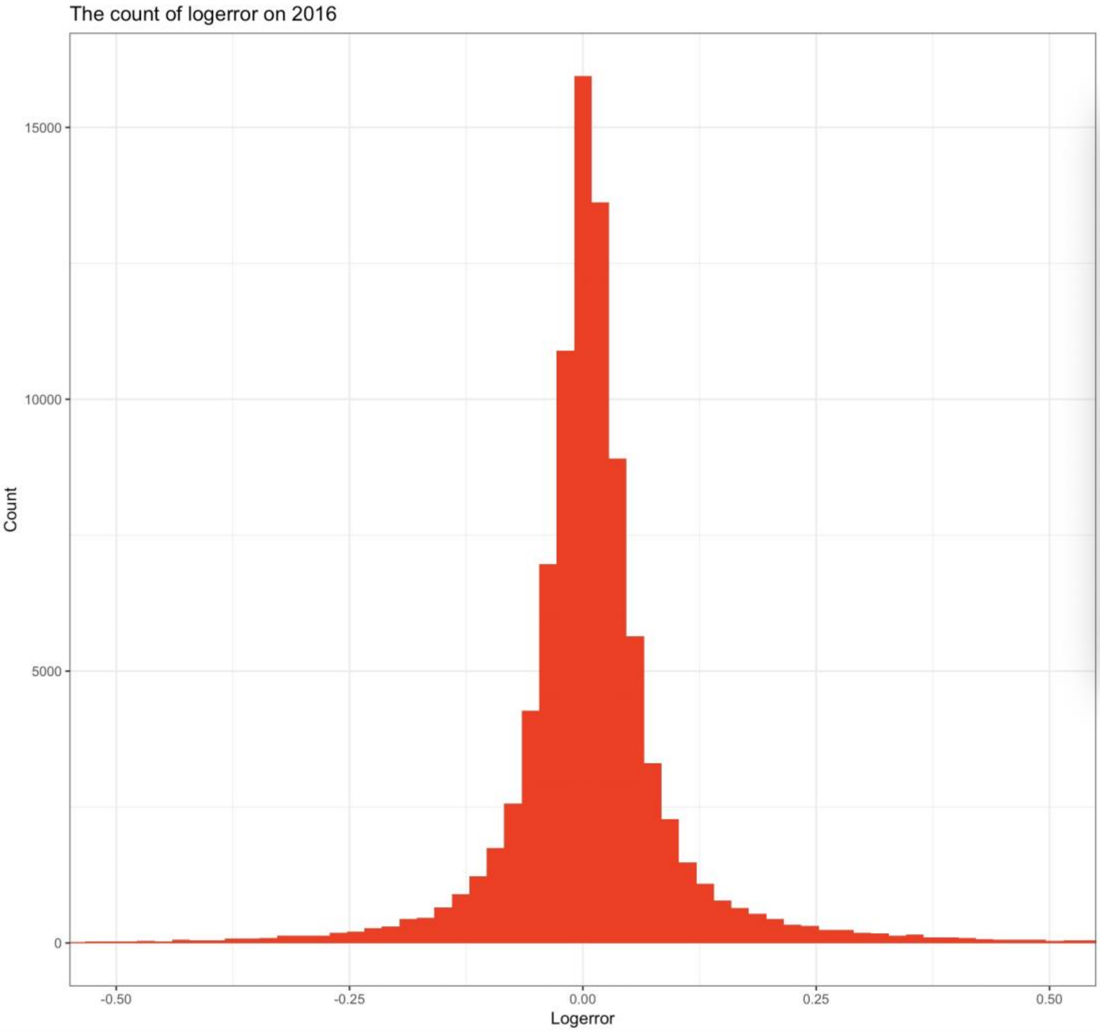

Figure 2

Now looking at the changing trend of absolute logerror, we could conclude that Zillow achieved a relatively good prediction on almost 80 percent of transaction happened on 2016 from Figure 3.

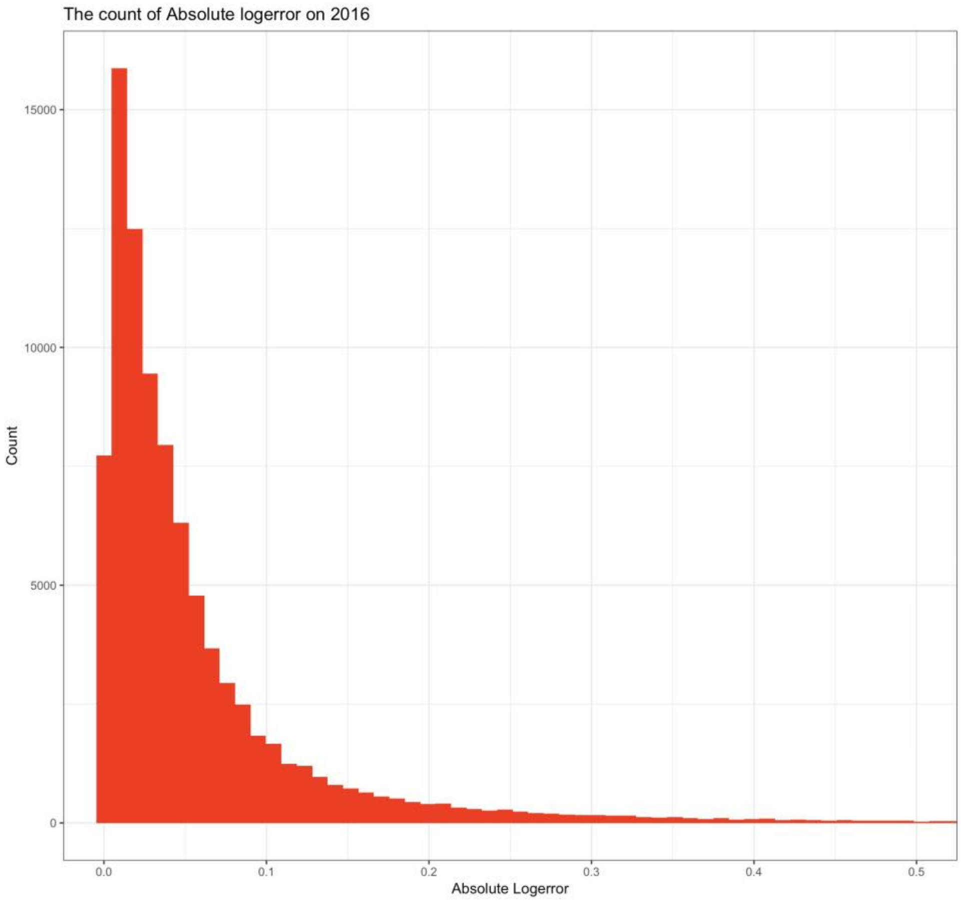

Figure 3

Besides the whole trend, I pay attention to which months Zillow predicted better. That is to say, in order to find out more useful and detailed trend behind the raw data in the whole year, I also plot the distribution of mean logerror, and the distribution of the mean of absolute logerror changed over time. As we can see below, Zillow done a good prediction on the actual house price from Apr to Jun, but have a relatively worse performance from the beginning and last two months. In more details, when looking through the distribution of the mean of absolute logerror ranging from Jan to Dec, it easily shows that Zillow's made a bad decision in the beginning and the last.

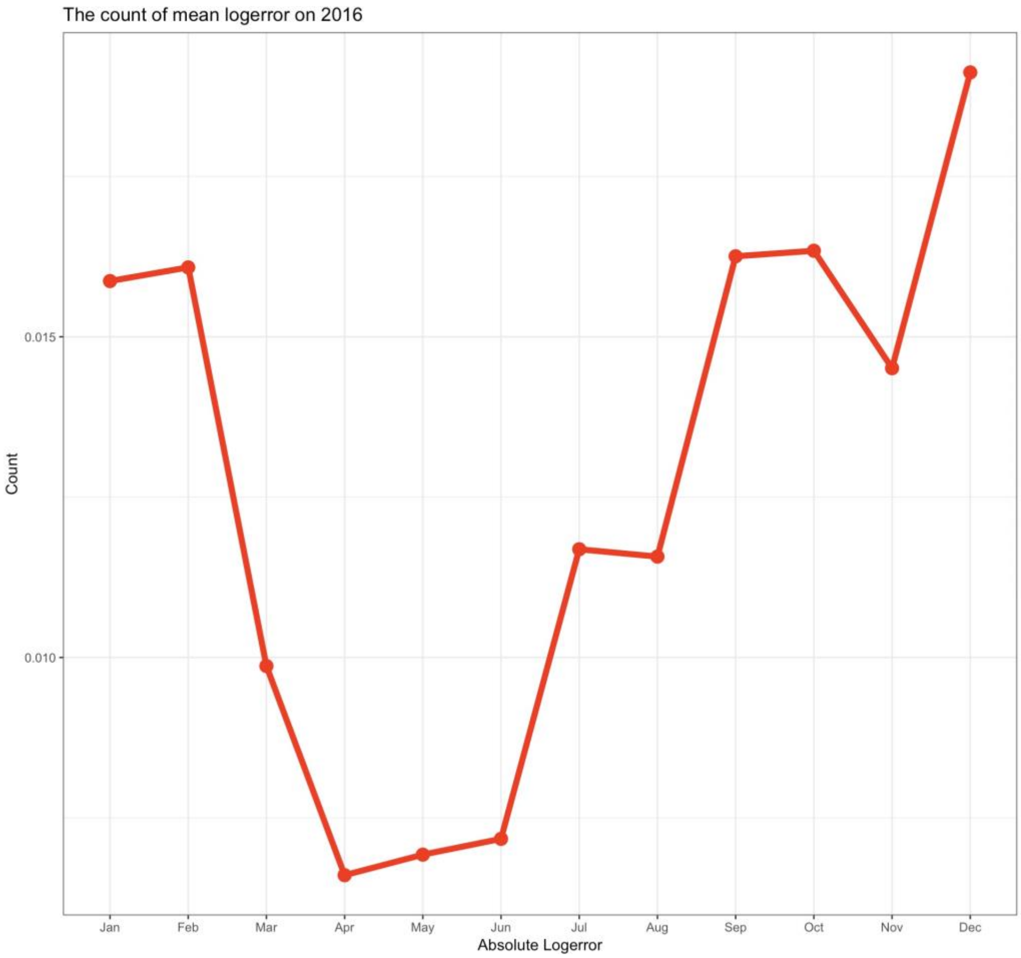
The count of mean logerror on 2016
Count
0.015
0.010
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Absolute Logerror

Figure 4

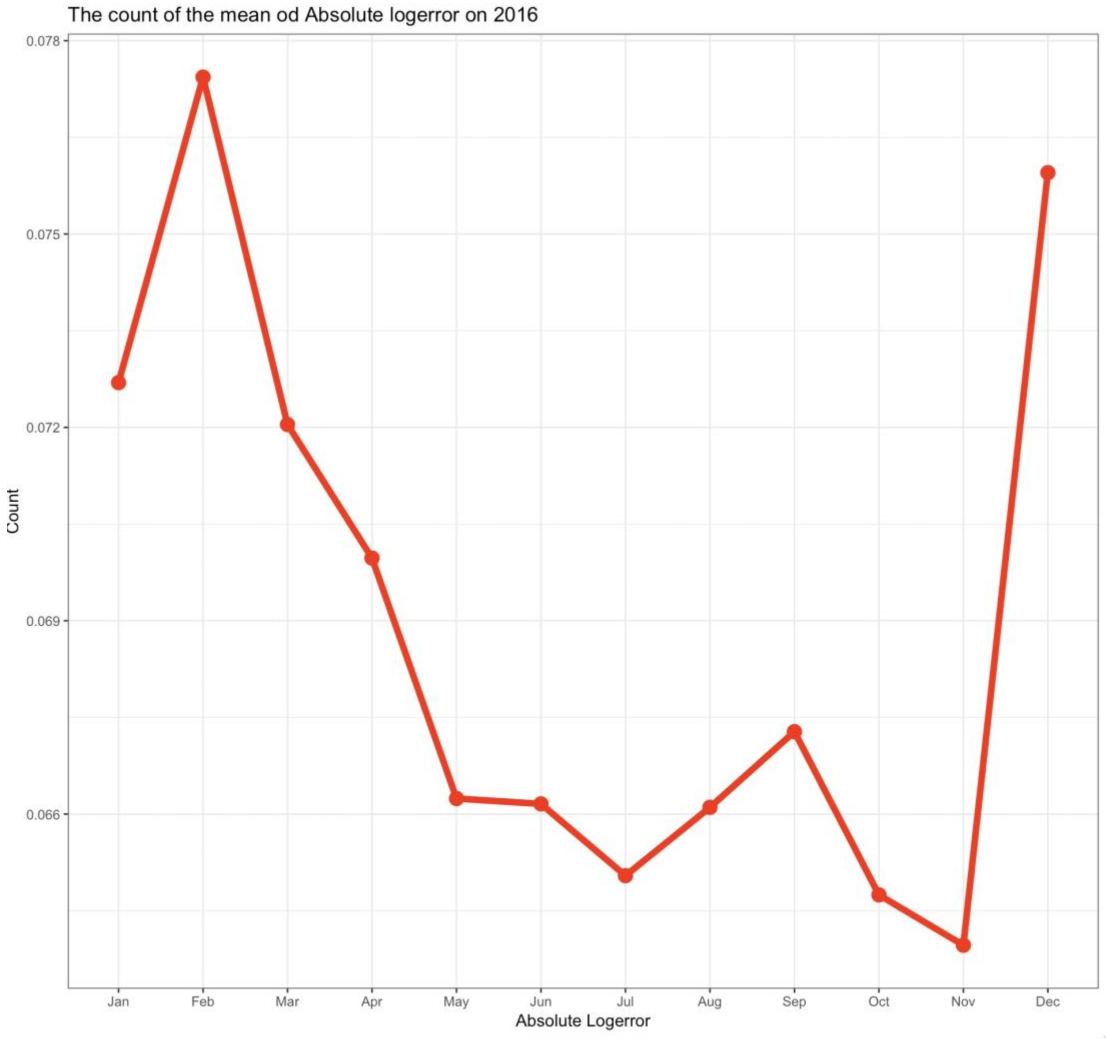

Figure 5

Now, after knowing the whole trend of transactions on 2016, I did some data visualization to see the relationship between the response and different variables or variables themselves, so that I could have a better understanding of the dataset. It will be very help to conduct feature engineering for modeling finally. Firstly, I checked the missing values of each column.

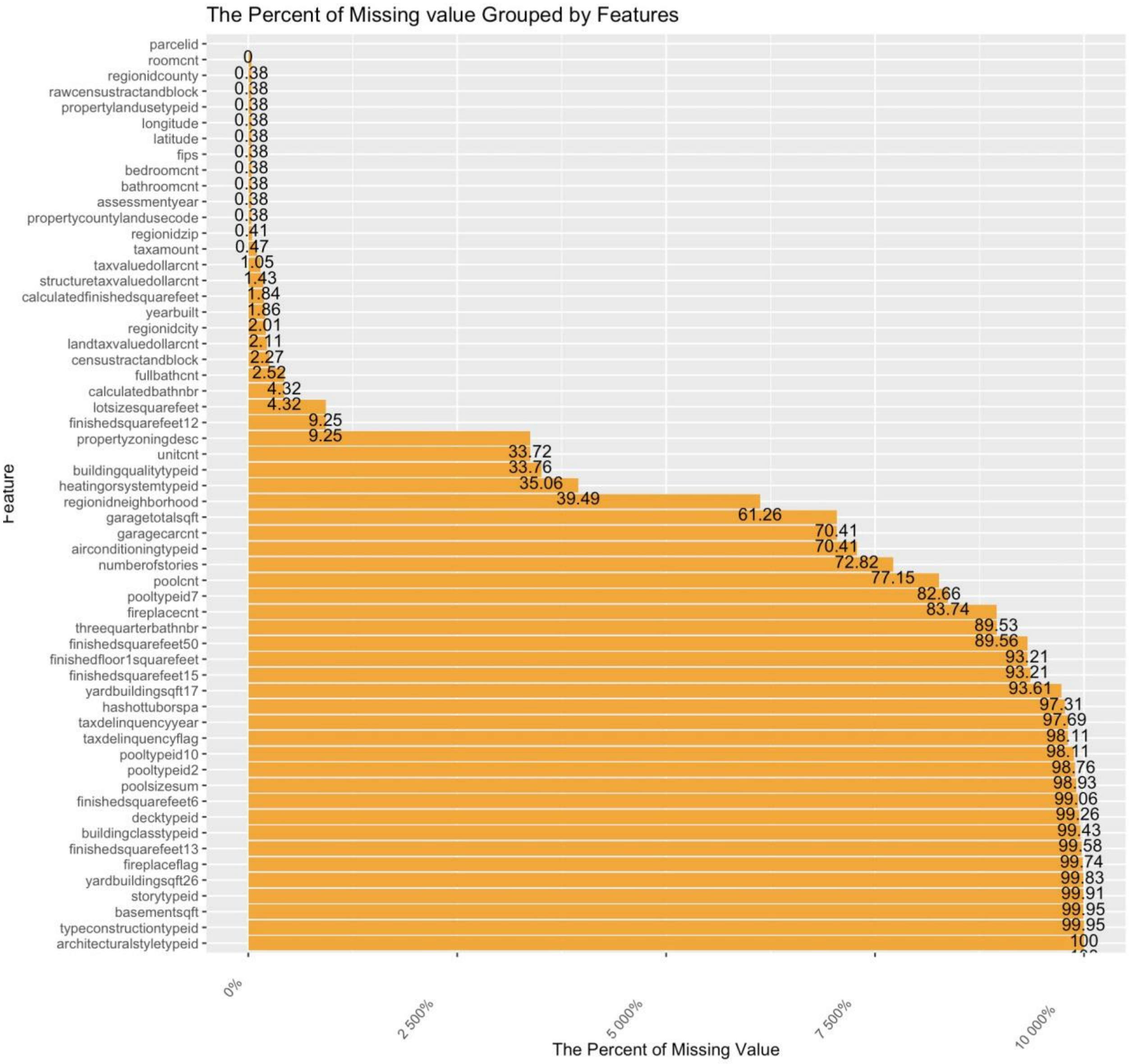

Figure 6

From Figure 6, it shows that we have many missing values in more than 10 columns. In order to better fit the model by reducing the dimension of the final dataset, those columns having many missing values will not provide useful information in the final model. I decide to filter all variables by setting a threshold and I will just keep those columns which have a percent of missing value is less than 75%.

Since the final goal is to identify whether the model could perform well on the "logerror", the response, I left join "properties" with useful columns with "train" data frames while mutating a new column which is the absolute value of logerror, named "abs_logerror" based on "logerror" column.

In order to figure out which columns have better correlation with the response, "logerror", I chose numeric variables from the combined data frame. There are 33 numeric features out of 63 columns. In order to have a better understating of dataset from different aspects, I divided those 33 columns into three groups, like the number of different room type, the geographic information and the tax information of the house. And then I plot the correlation plot of "abs_logerror" and "logerror" with those different groups. From the below figures, we could conclude that there does exist a correlation between those variables and the response.

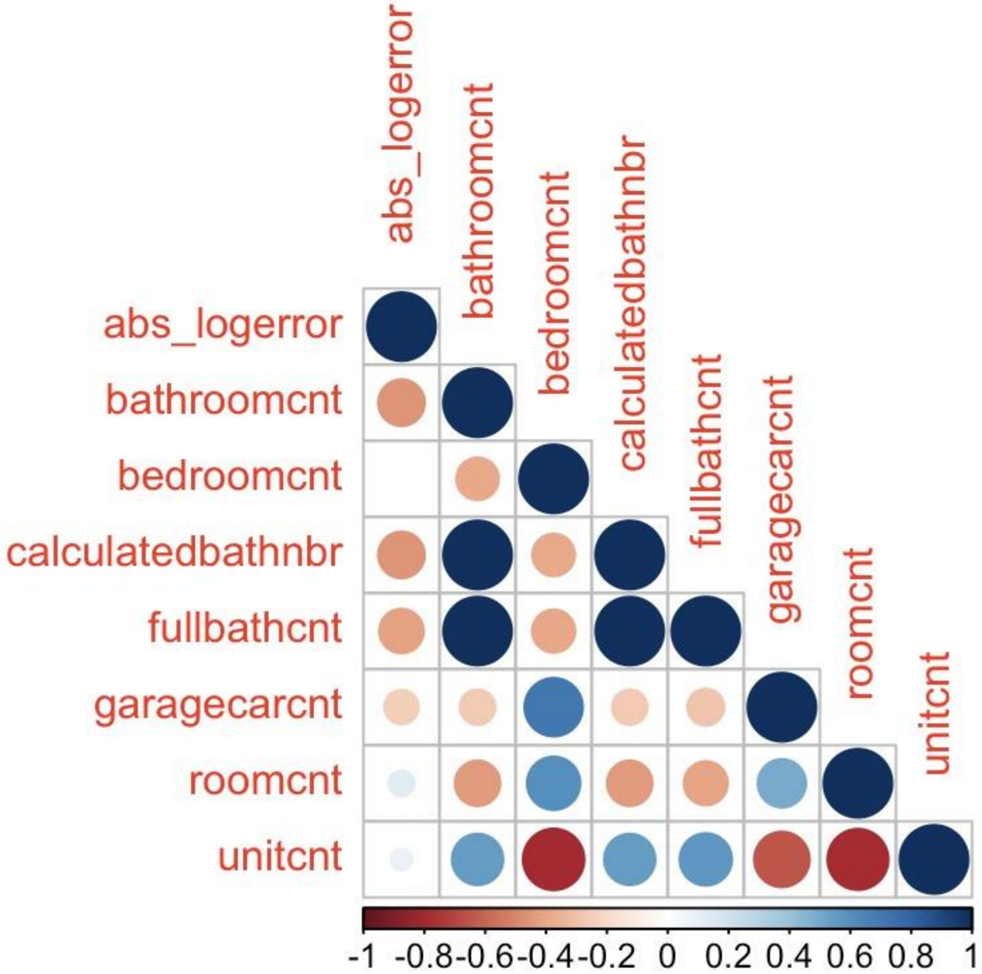

Figure 7

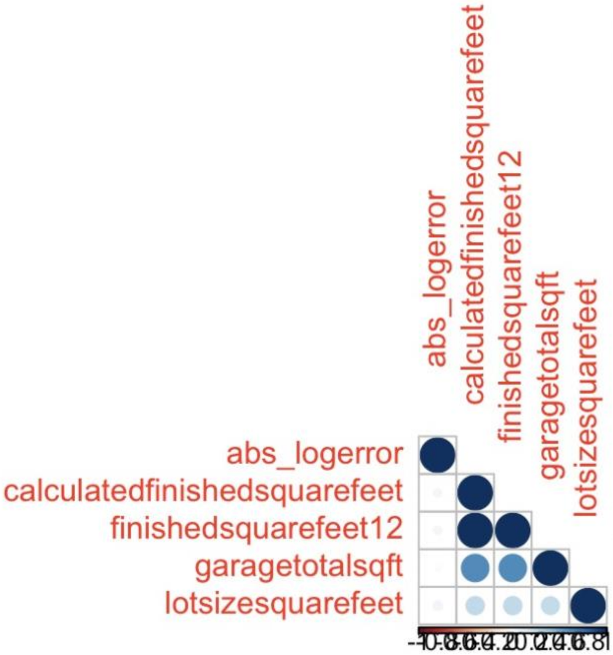

Figure 8

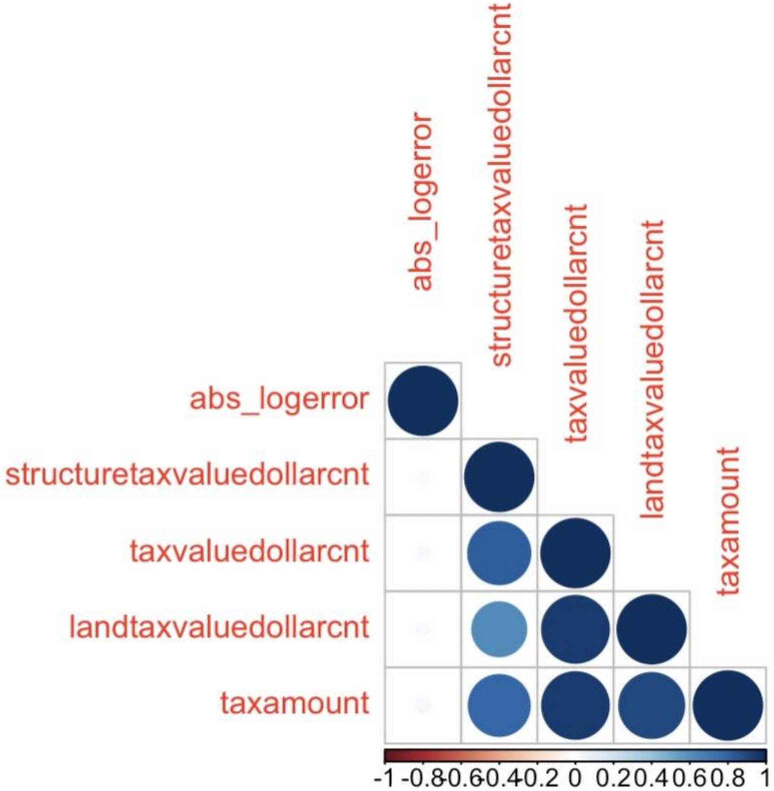

figure 9

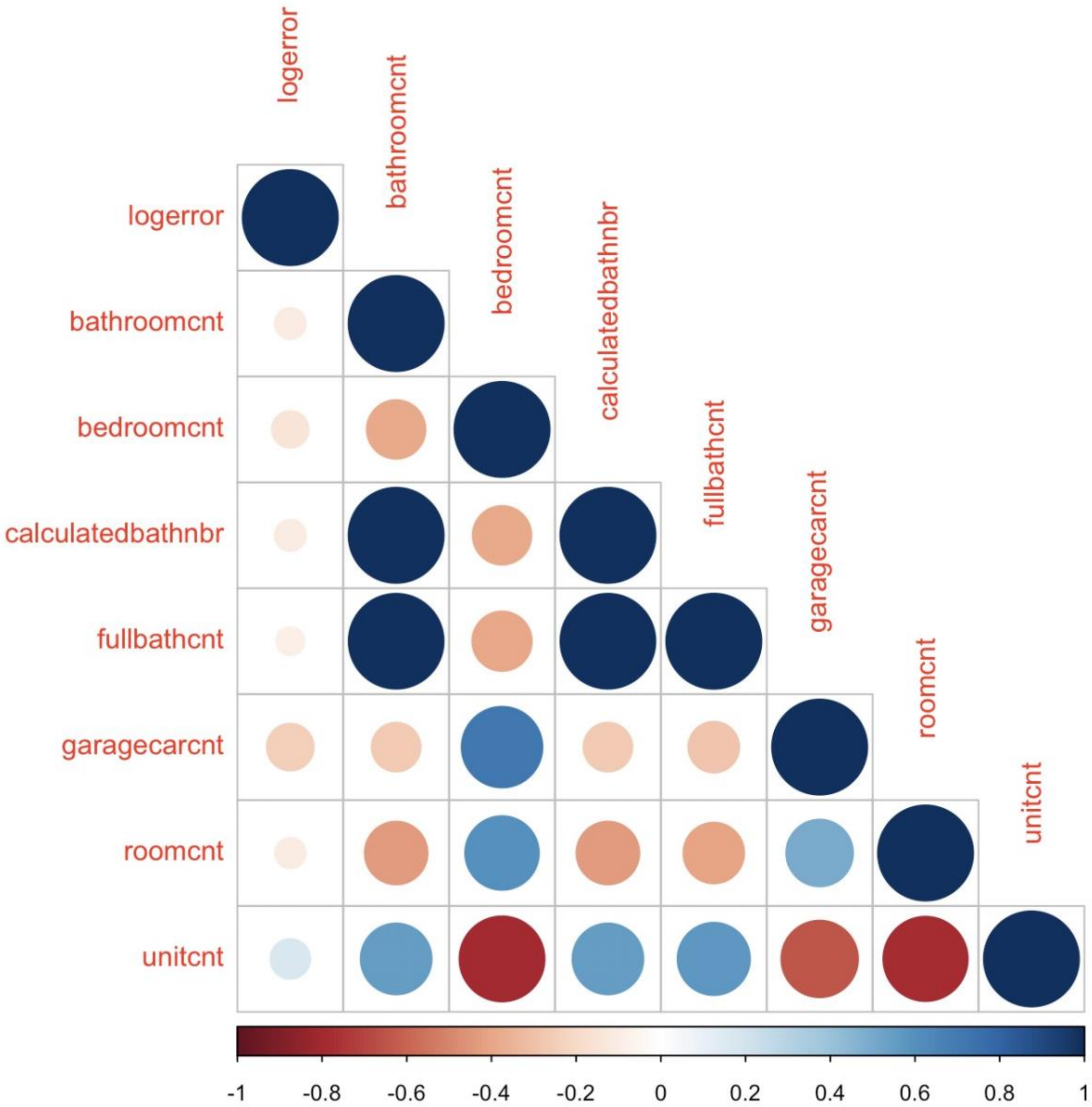
logerror
bathroomcnt
bedroomcnt
calculatedbathnbr
fullbathcnt
garagecarcnt
roomcnt
unitcnt
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

Figure 10

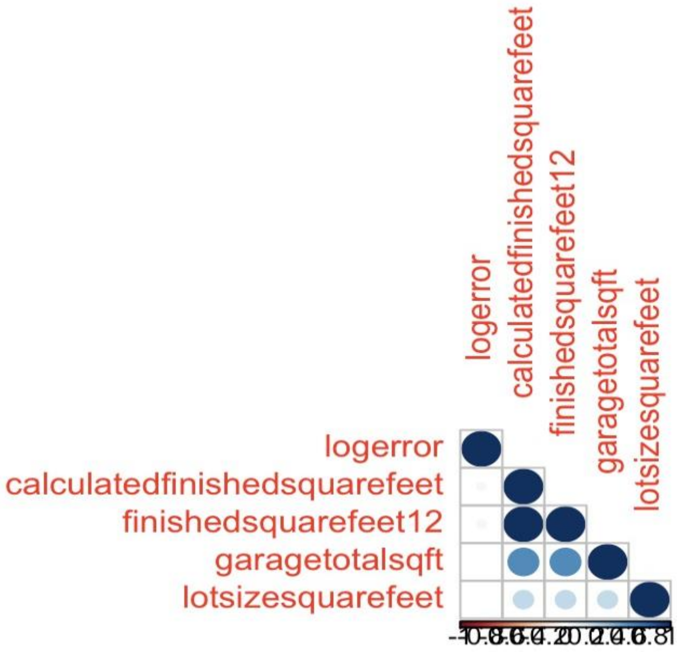

Figure 11

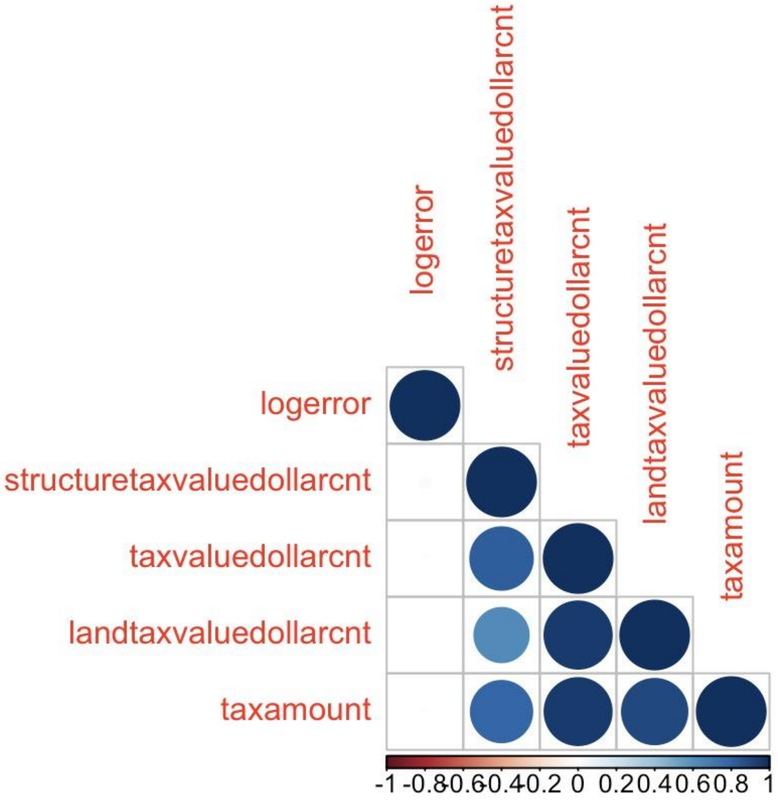

Figure 12

Now, looking through the distribution of house itself, like which year had more house built. From Figure 13, there was a house built peak in 1950. And analyzing the trend of absolute logerror changed by the year of house built combined with the density plot of house built, I found that the prediction done by Zillow about the house built in 1950 was good and improved a lot about those houses built after 1950.

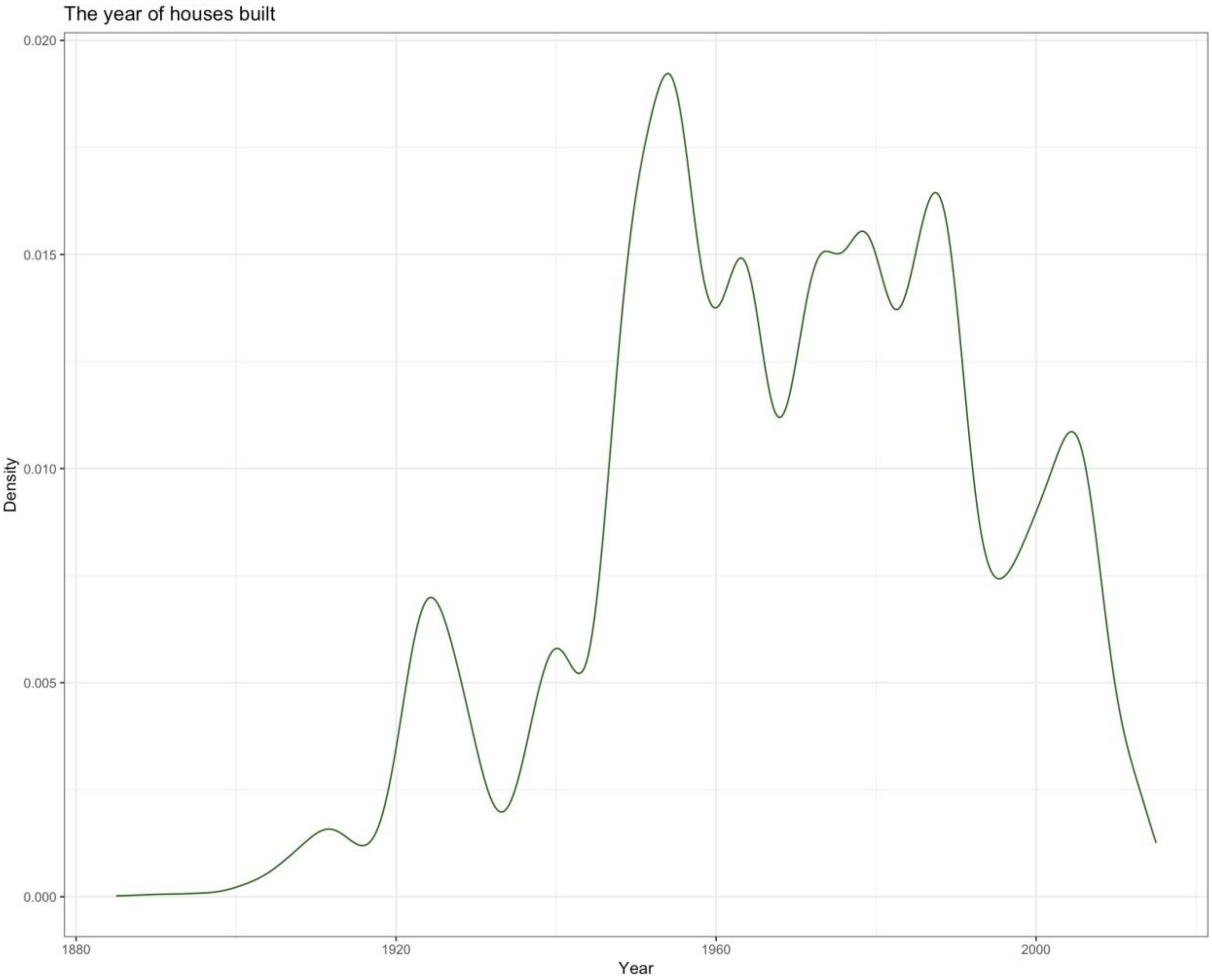

Figure 13

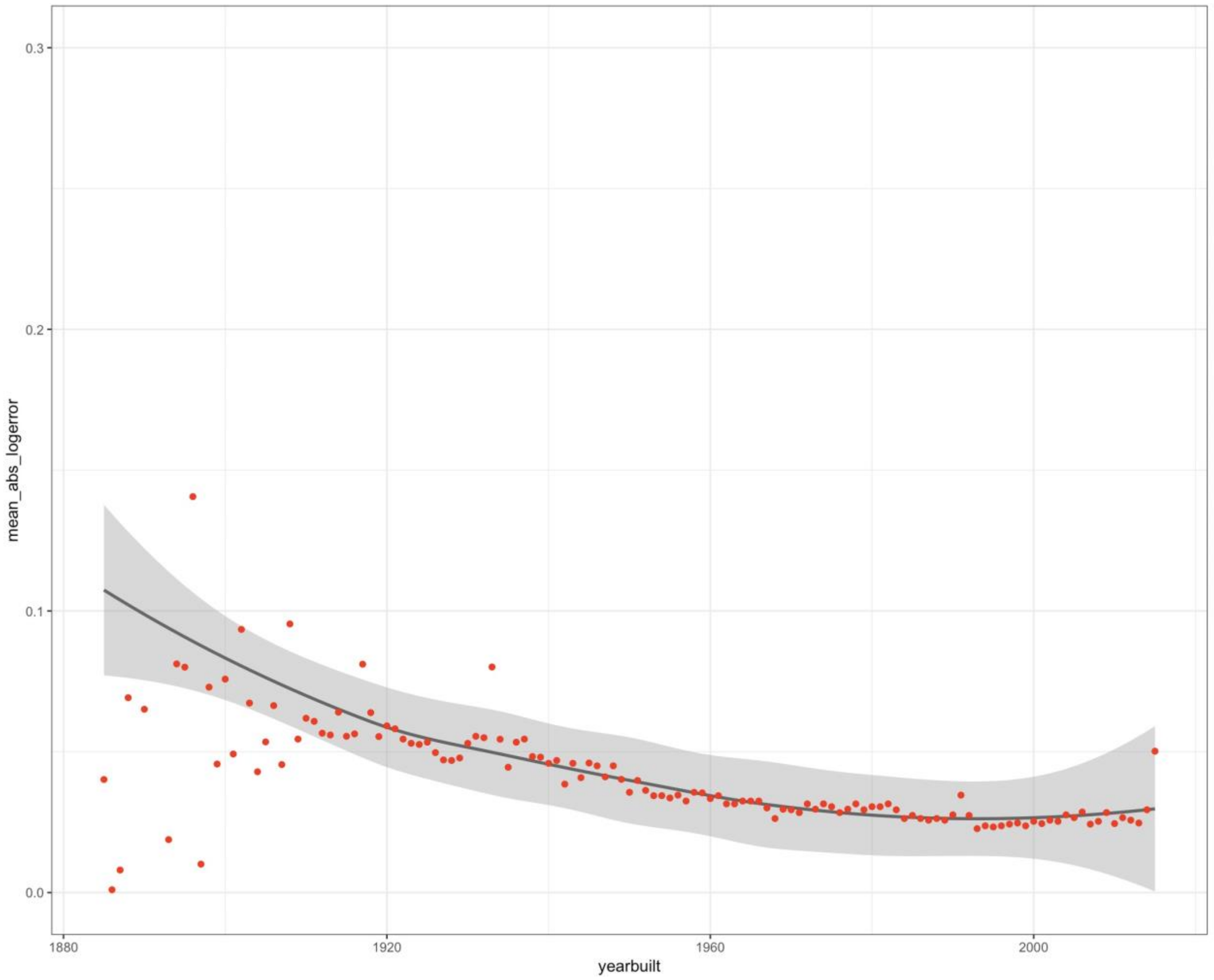

Figure 14

Because it is also significant to know which one or more variables has a relatively stronger relationship with the response, no matter a good prediction, bad prediction or typical one (not so good but also not so bad). I divided the absolute value of “logerror” into five classes by using “quantile” function, and then chose three classes, “best_prediction”, “worst_prediction” and “typical_prediction”. And correspond values of percentile of those three are 1,5 and 3. Firstly, plotting the relationship between the total finished area and those three different classes. And I concluded that those three different prediction followed a relatively similar trend based on total finished area in a large extent. Especially, they all reached a peak when total finished area equals to 2000 square feet. And taking a closer look at the distribution of absolute logerror and total finished areas, it also shows there existed the smallest shrinkage in 2000 sqaure feet.

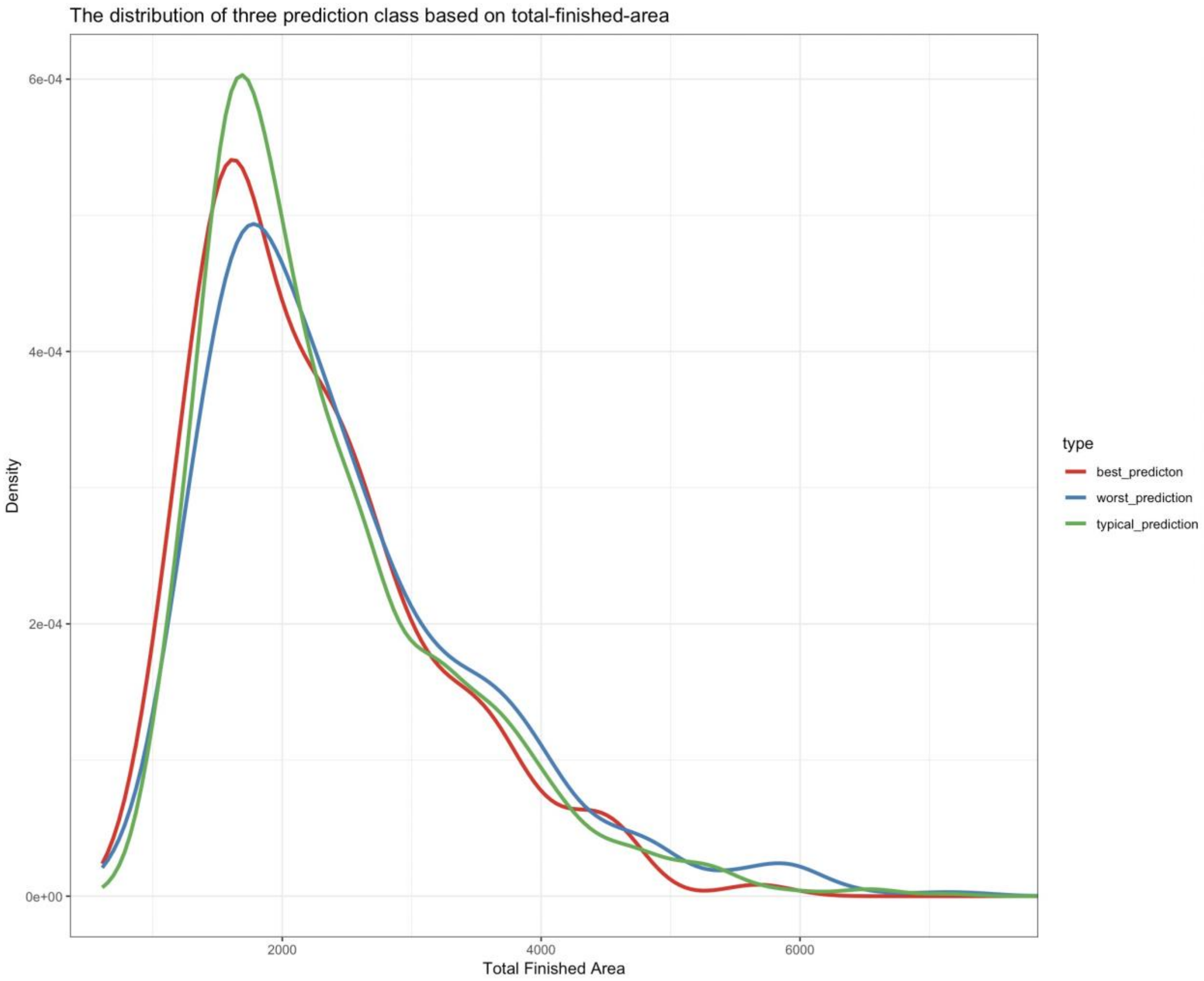
The distribution of three prediction class based on total-finished-area
6e-04
4e-04
2e-04
0e+00
Density
2000
4000
6000
Total Finished Area
type
best_predicton
worst_prediction
typical_prediction

Figure 15

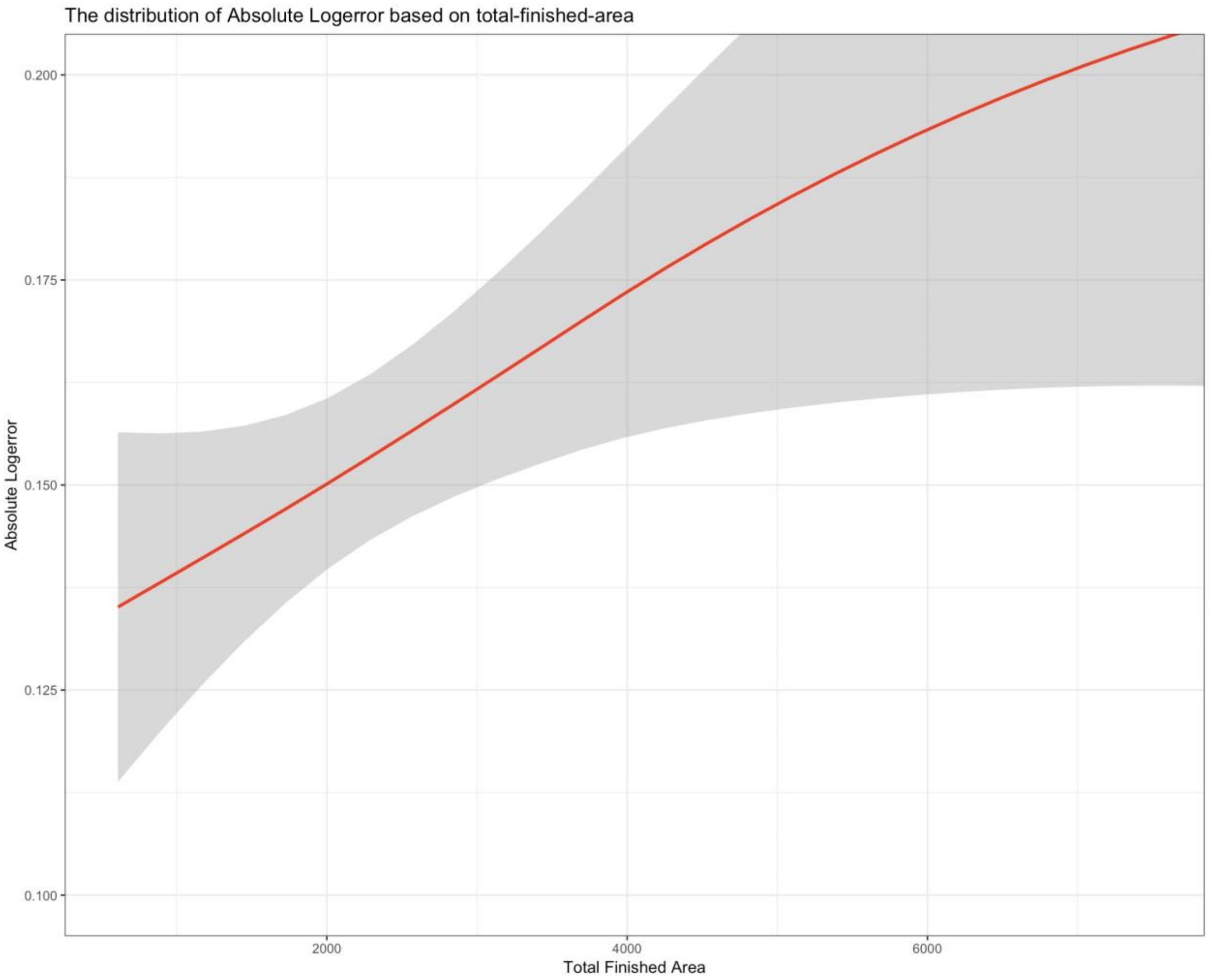

Figure 16

When focused on the relationship between the finished living area with those three different logerror, it showed similar trend as that based on total finished area from Figure 17.

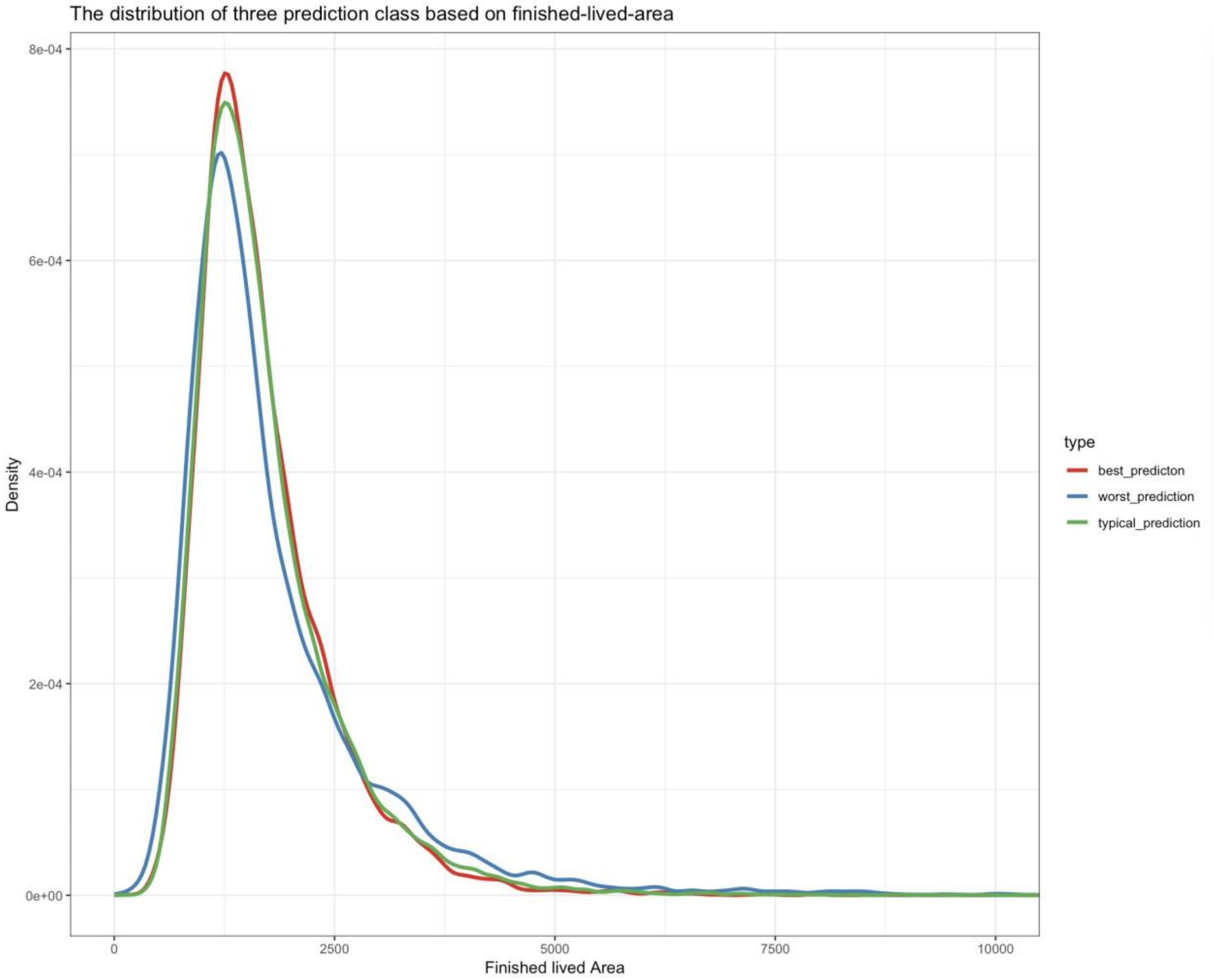

Figure 17

Then, I also took a glimpse of the relationship between those three different predictions with the number of rooms, the number of units, different house built year, the total sum of tax, and the assessed value of the house. From all below figures, I concluded that all those features have a relationship with the response, “logerror”.

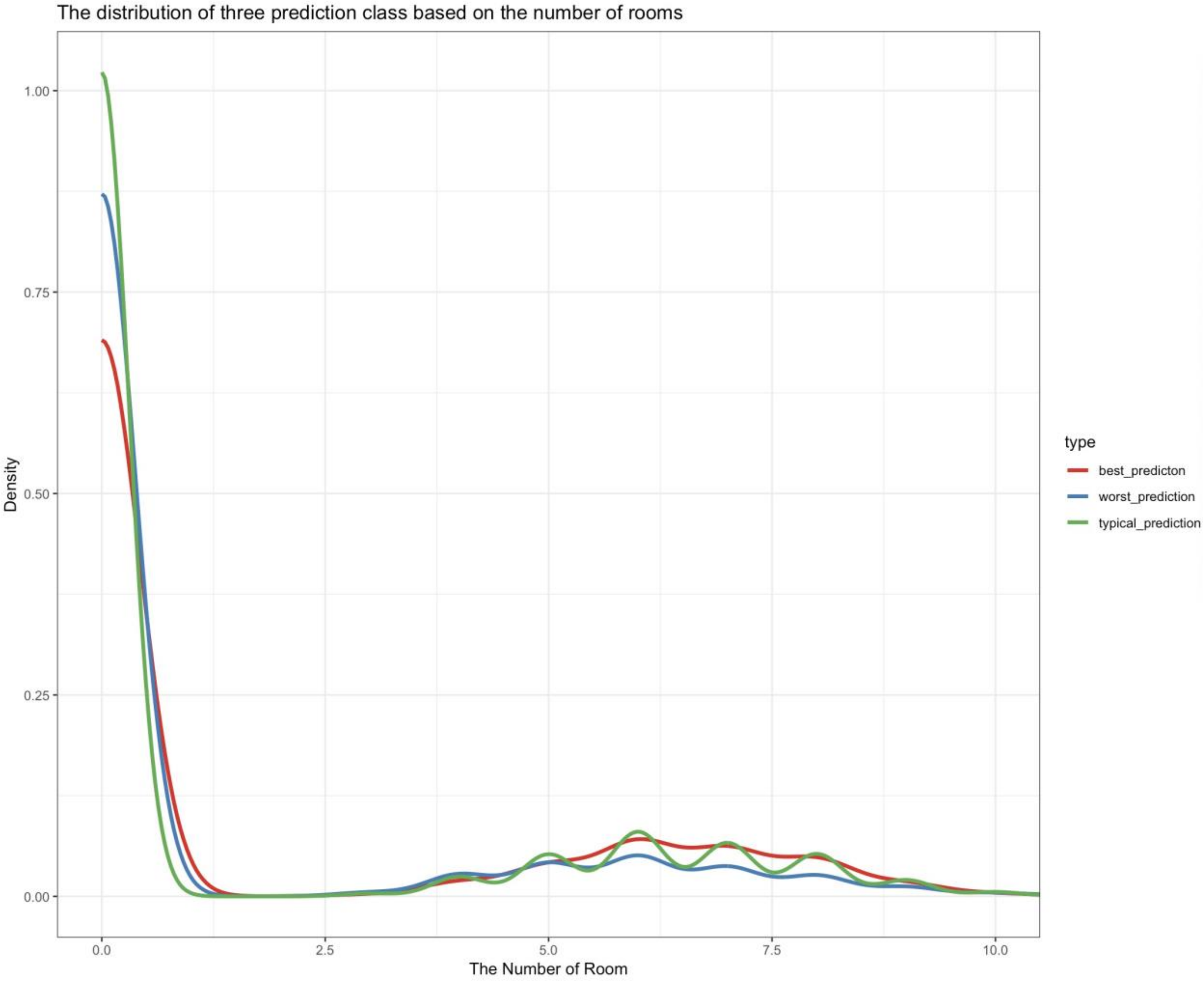
The distribution of three prediction class based on the number of rooms
Density
1.00
0.75
0.50
0.25
0.00
0.0
2.5
5.0
7.5
10.0
The Number of Room
type
best_predicton
worst_prediction
typical_prediction

Figure 18

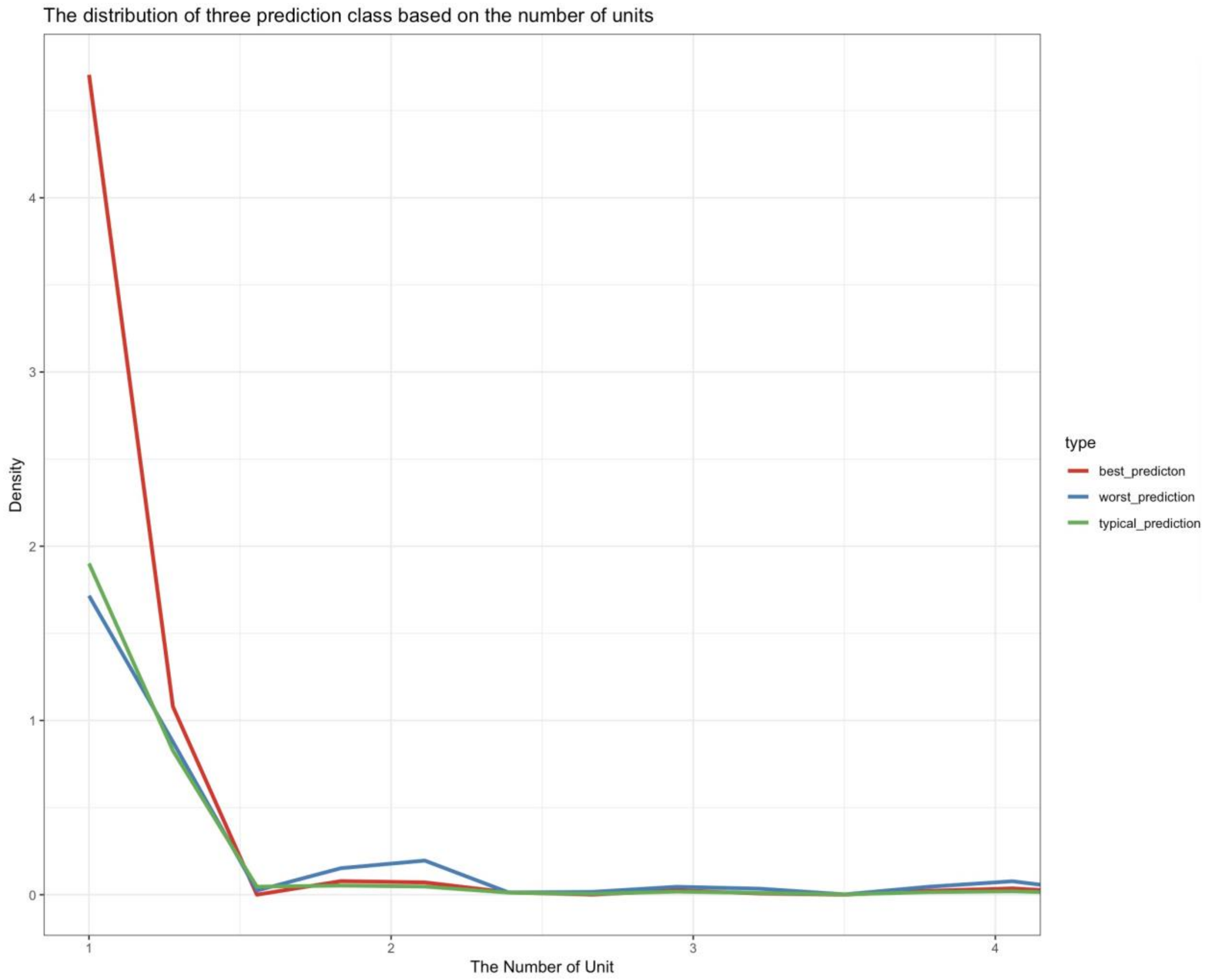
The distribution of three prediction class based on the number of units
Density
4
3
2
1
0
1
2
3
4
The Number of Unit
type
best_predicton
worst_prediction
typical_prediction

Figure 19

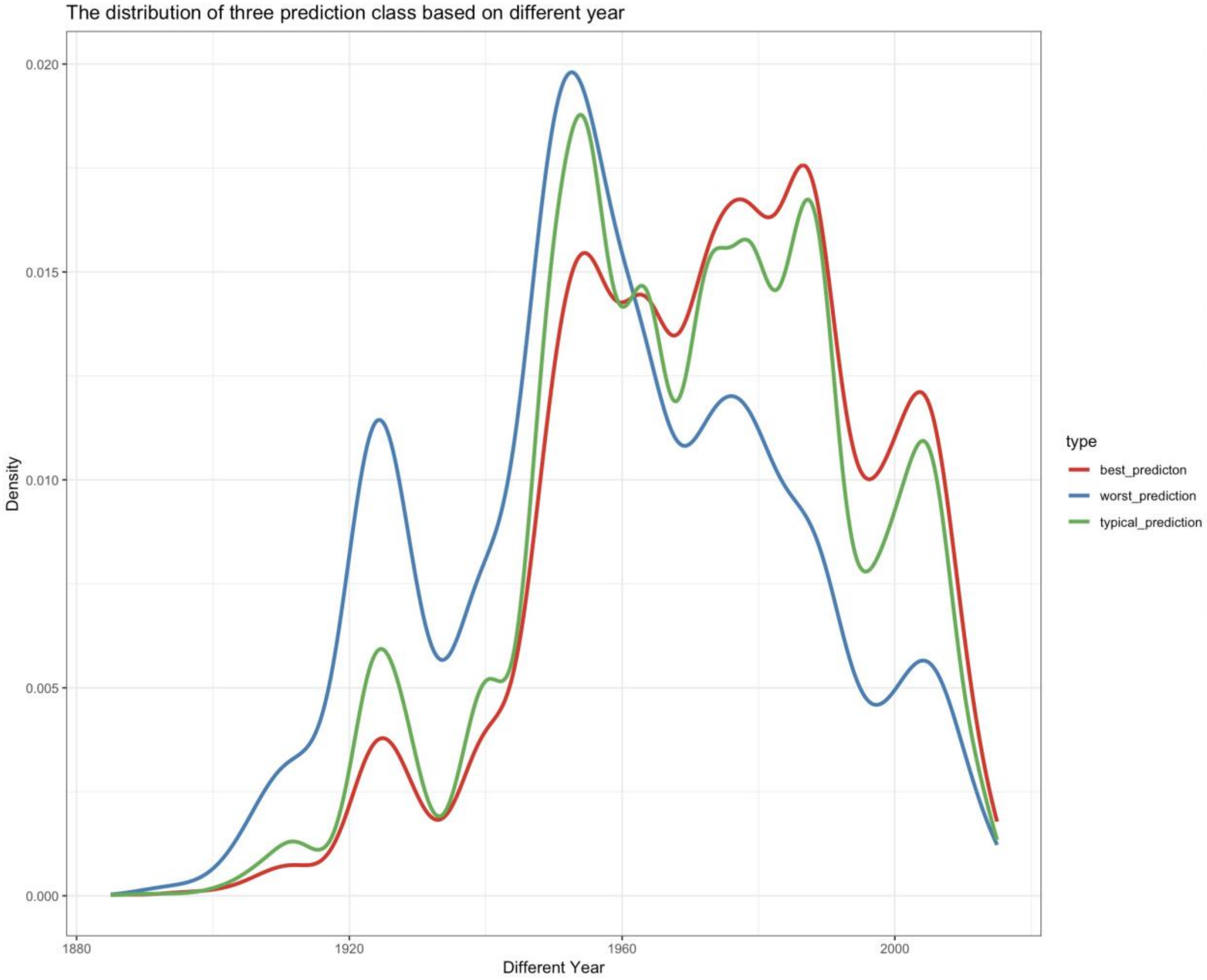
The distribution of three prediction class based on different year
Density
0.000
0.005
0.010
0.015
0.020
1880
1920
1960
2000
Different Year
type
best_predicton
worst_prediction
typical_prediction

Figure 20

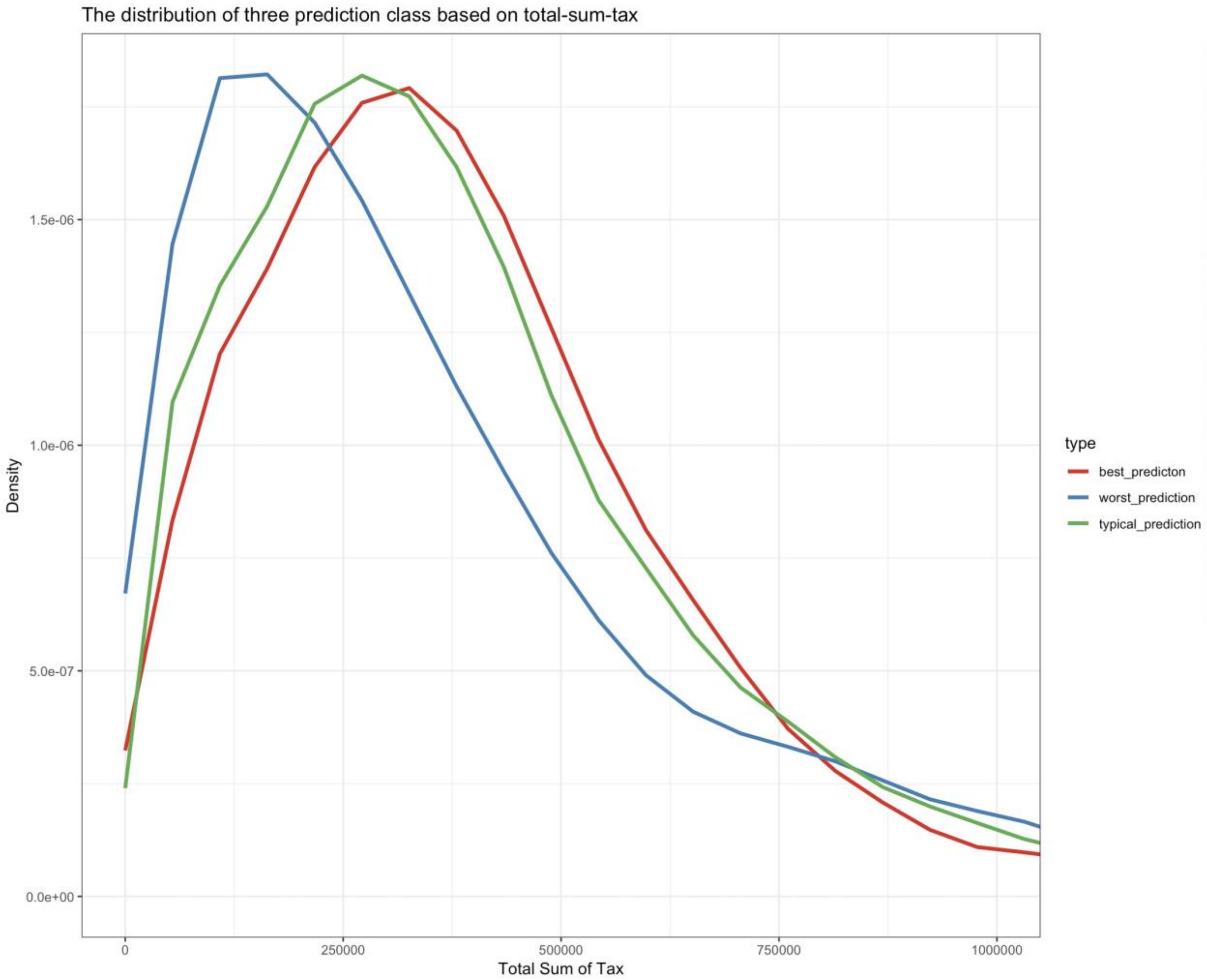
The distribution of three prediction class based on total-sum-tax
Density
1.5e-06
1.0e-06
5.0e-07
0.0e+00
0
250000
500000
750000
1000000
Total Sum of Tax
type
best_predicton
worst_prediction
typical_prediction

Figure 21

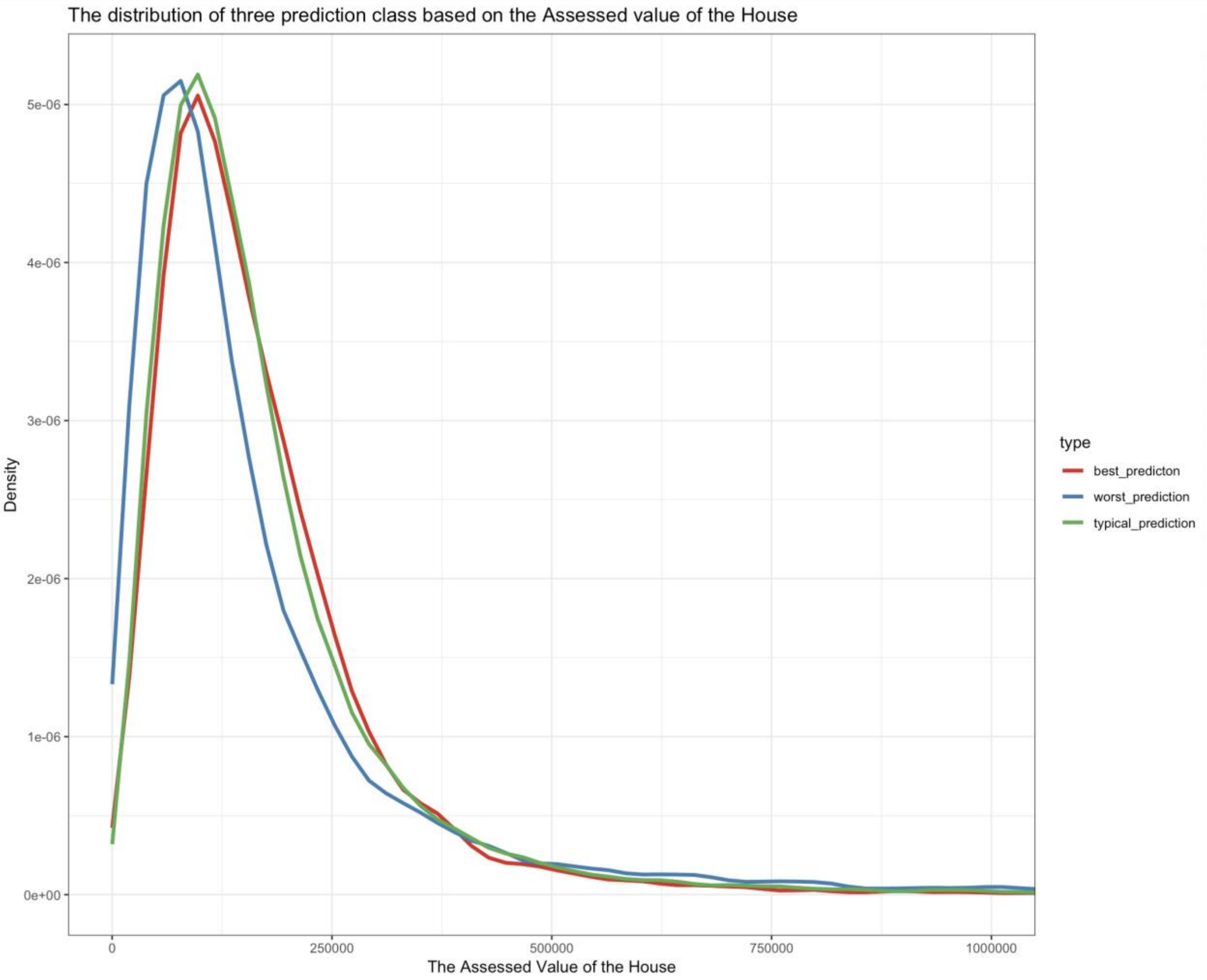

Figure 22

As I mentioned before, if logerror < 0 , it is underestimation, otherwise. And I also paid attention to which areas Zillow overestimated the house and which areas they underestimated. Limited by space and computer memory, I just randomly selected 2000 samples observations from the merging data frame to make a map. From Figure 24, I found there were many overlapped points in the Los Angeles, which means that specific location does not matter the changing trend of the response – "logerror".

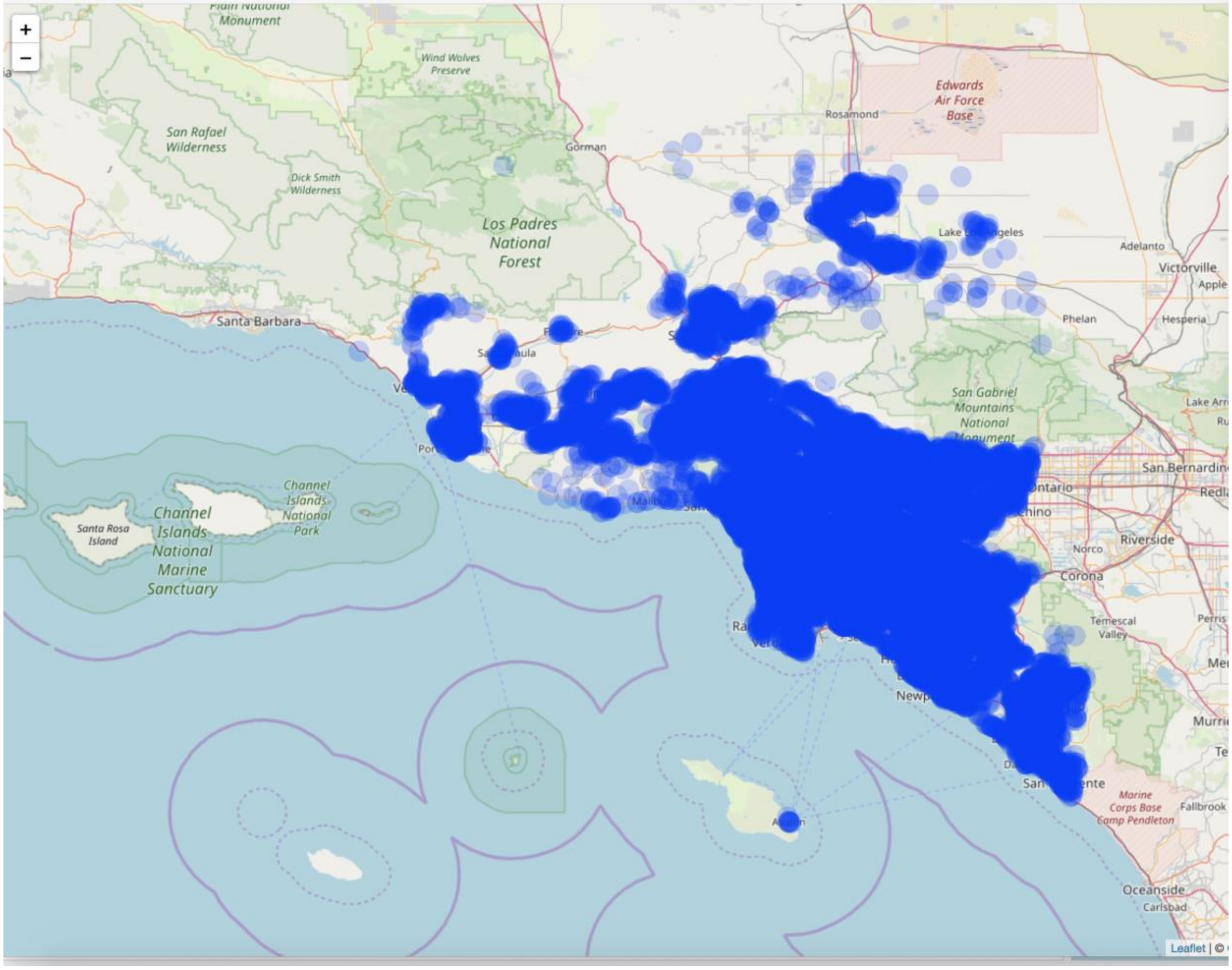
Monument
Wind Wolves Preserve
Edwards Air Force Base
Rosamond
San Rafael Wilderness
Dick Smith Wilderness
Gorman
Los Padres National Forest
Adelanto
Victorville
Hesperia
Phelan
Santa Barbara
San Gabriel Mountains National Monument
Channel Islands National Park
Santa Rosa Island
Channel Islands National Marine Sanctuary
Ontario
Chino
San Bernardino
Riverside
Norco
Corona
Temescal Valley
Marine Corps Base Camp Pendleton
Fallbrook
Oceanside
Carlsbad
Leaflet

Figure 23

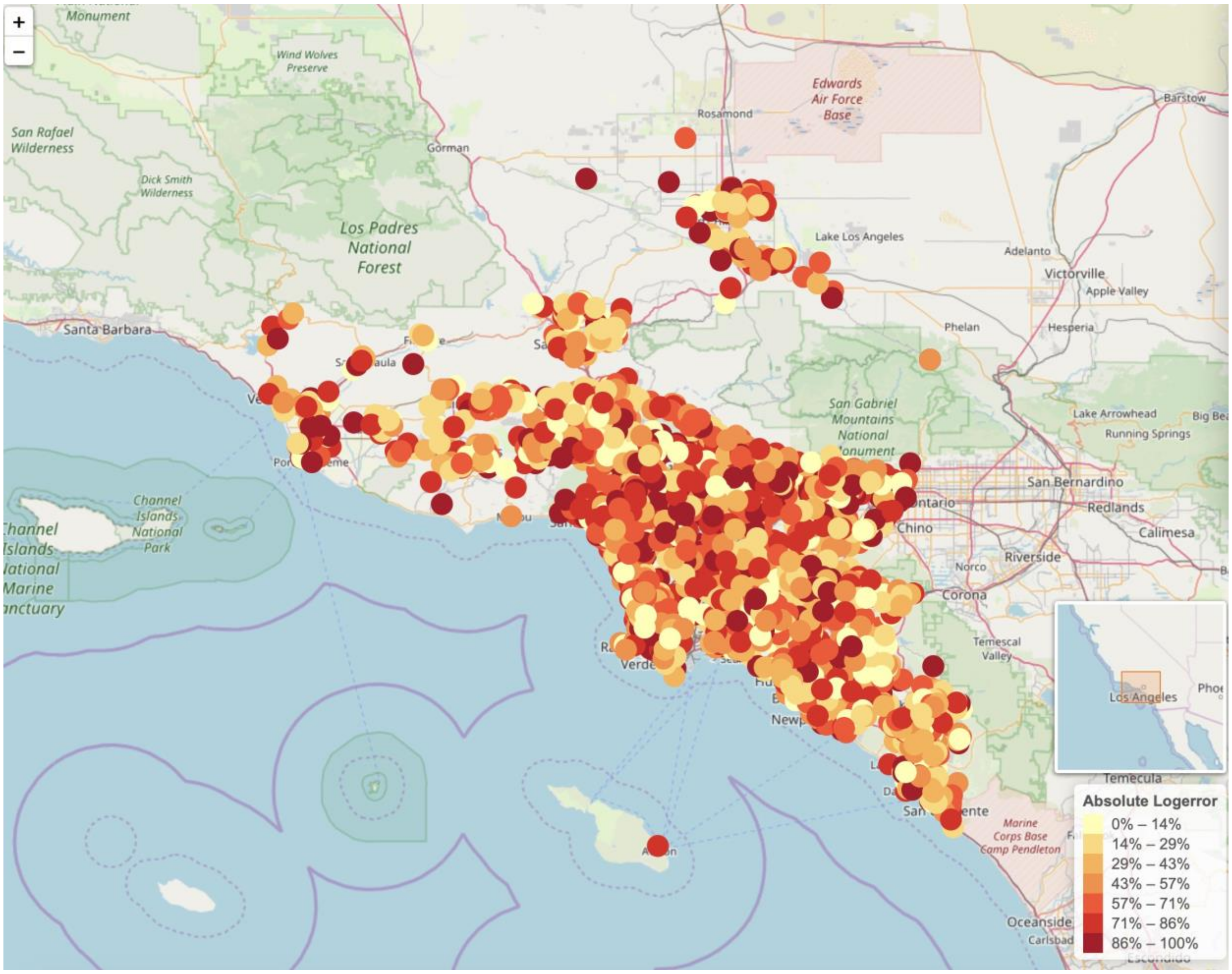

Figure 24

After carefully looking through those columns, I relatively fully understood which variables will play a significant role in final model. However, there are four factor columns in all good features, "fips", ”regionidcountry“, “regionidneighborhood","regionidzip","regionidcity". Before training the model, I needed to deal with those factor in order to make machine learning works. Since XGBoost could only handle with numeric variables, I chose OneHotEnocing or dummy variables to further preprocess those four features. Taking the dimensionality of final dataset and expensive computation into account, I deleted "regionidneighborhood", "regionidzip", "regionidcity" those three columns which all have more than 10 unique observations. And using OneHotEnocing to deal with “fips” and “regionidcounty”.

Building Matrix

Making decision which variables should be included in my final model and how to handle with some factors, I also conducted same data preprocessing on the test dataset ranging from selecting appropriate variables to washing some wrangling features. Since the target is “logerror”, and variables related to date were not longer useful index for modeling. I dropped off "parcelid", "logerror", "Year_Month", "abs_logerror" from training dataset. Finally, building two XGBoost matrix based on cleaned training dataset and test dataset.

Modeling Fitting

Compare with the results calculated by random forest and bagging algorithms, I finaly chose a model combined with eXtreme Gradient Boosting algorithm to train my model. Because my goal is to predict the logerror based on all features related to the house instead of classification in multiple classes. I made some changes on model parameters by redefine “amm_mae”. And then I got the predicted logerror on 2017 and merge columns, the predicted logerror and the actual one together by left-join function. Finally, comparison was conducted based on a formula:

$$\textbf{RMSE} = \frac{1}{n} * \sum_{i=1}^{n}\left(y_i - y_i^{\Delta}\right)\text{\textasciicircum}2$$

Result and Conclusion

RMSE is equal to 0.09894484. It is relatively low which means the model fits the test dataset well. But there still exists some problems which need answer. The questions are as follows:

a. From Figure 1, we could see that in the last three months, there are a downtrend. There may exist some reasons which is worthy to be investigated;
b. Why three different prediction classes all go up until reaching a same threshold and then go down, no matter which variables they are related to;
c. RMSE could be further reduced by tuning the parameter;