**Overview**

The Project is to build a model to improve the Zestimate residual error.

*logerror=log(Zestimate)−log(SalePrice)*

“Zestimates” are Zillow's estimated home values. The model is to predict the difference between the Zillow's estimated home value, Zestimate, and the actual sale price.

**Client**

The client could be Zillow. Zillow can improve its algorithm with the model which would predict where zestimates will do good or bad. When we want to improve existing model, modeling errors can be a good way to find areas to improve the existing model.

**Data**

The data used in the project has been provided from Zillow through Kaggle.com.

The following files were used in the project. We will use data in 2016 as a train data and 2017 data as a test data. Data have 60 columns which have features for homes.

* Train Data: Two dataset where one dataset contains explanary variable and the other dataset has target variable, log error are merged to produce a train data

1. **properties\_2016.csv**: a full list of real estate properties in three counties (Los Angeles, Orange and Ventura, California) data in 2016.
2. **train\_2016.csv**: all the transactions before October 15, 2016, plus some of the transactions after October 15, 2016. It contains parcel ID , transaction date and calculated log error .

- Test Data:

1. **properties\_2017.csv**: a full list of real estate properties in three counties (Los Angeles, Orange and Ventura, California) data in 2017.
2. **train\_2017.csv**: all the transactions from Jan 1, 2017 to Sep 25, 2017. It can be used as a test dataset.

**Approach to solving this problem**

First of all, find out any variables which relate to the target variable, logerror. Then, apply regression model or any statistical model which can be applicable.

**Deliverables**

This would include code and a report.

**Data Wrangling**

**Data Cleaning**

- Duplication : I explored training data. 125 duplicated data for 2016 and 199 duplicated data for 2017 data were found. However, it meant they were trasacted for more than twice for a year. So, I didn’t delete any duplication.

- Negative values: Also, I checked if there were any negative numbers for each column. Two columns, logerror and longitude, have negative values which are reasonable to have for them.

- Unusual Object : We have 5 columns which are objective type. Each column does not have unusual values, for example “?” , “$”

**Missing Values**

Let's check how many missing value each column has. I found that 47 columns out of 60 columns have missing values and 18 columns among them have more than 95% of missing values.

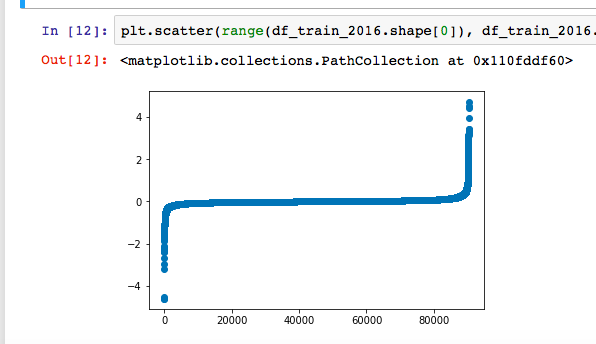
Let's explore how missing values were treated.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Column name | Describtion | Missing | Missing Values |
| 1 | buildingclasstypeid | The building framing type | 99.98% | Deleted the column because only 16 cells out of 90275 cells are not missing and all with the same value 4. Rest of data, 90259 are missing for random. |
| 2 | finishedsquarefeet13 | Perimeter living area | 99.96% | Delete the column because every build must have living area and most of them are missing. |
| 3 | basementsqft | Finished living area below ground level | 99.95% | Filled with 0 because every building does not have living area below ground leve, missing could mean building does not have partial living room. |
| 4 | storytypeid | Type of floors in a multi-story house | 99.95% | Deleted the column because every building should have a type of floor and most of them are missing . |
| 5 | yardbuildingsqft26 | Storage shed/building in yard | 99.89% | Filled with 0 because missing value can mean it doesn't have storage in yard |
| 6 | fireplaceflag | Is a fireplace present in this home | 99.75% | Filled with False because missing value means it does not have a fireplace. |
| 7 | architecturalstyletypeid | Architectural style of the home | 99.71% | Deleted the column because every buildling has its architectural style and most of them are missing. |
| 8 | typeconstructiontypeid | type of construction used to construct the home | 99.67% | Deleted the column because every building has its type of construction material and most of them are missing |
| 9 | finishedsquarefeet6 | Base unfinished and finished area | 99.53% | Deleted the column because everyg home should have base area and most of them are missing |
| 10 | decktypeid | Type of deck present on parcel | 99.27% | Deleted the column because non-missing cells have the same value, 66 and most of them are missing |
| 11 | poolsizesum | Total square footage of all pools on property | 98.93% | Deleted the column because it is missing randomly and most of them are missing |
| 12 | pooltypeid10 | Spa or Hot Tub | 98.71% | Deleted the column because it is missing randomly and most of them are missing |
| 13 | pooltypeid2 | Pool with Spa/Hot Tub | 98.67% | Deleted the column because it is missing randomly and most of them are missing |
| 14 | taxdelinquencyflag | Property taxes for this parcel are past due as of 2015 | 98.02% | Filled with Y because all non-missing values are "N" |
| 15 | taxdelinquencyyear | Year for which the unpaid propert taxes were due | 98.02% | Filled with 0 because the missing cells are the same as the previoius column. |
| 16 | hashottuborspa | Does the home have a hot tub or spa | 97.38% | Filled with False because all non-missing cells are "True" |
| 17 | yardbuildingsqft17 | Patio in yard | 97.07% | Filled with 0 because not every building has a patio in yard. |
| 18 | finishedsquarefeet15 | Total area | 96.05% | Deleted the column because every place should have total area and most of them are missing |
| 19 | finishedfloor1squarefeet | Size of the finished living area on the first floor of the home | 92.41% | Deleted the column because most place has living area and most of cells are missing |
| 20 | finishedsquarefeet50 | Size of the finished living area on the first floor of the home | 92.41% | Deleted the column because it is the repeat of the previous column. |
| 21 | fireplacecnt | Number of fireplaces in a home (if any) | 89.36% | Filled with 0 because not every building has fireplace. |
| 22 | threequarterbathnbr | Number of 3/4 bathrooms in house (shower + sink + toilet) | 86.70% | Filed with 0 because no all home has 3/4 bathrooms |
| 23 | pooltypeid7 | Pool without hot tub | 81.50% | Deleted the column because not every home has a pool and most of them are missing. |
| 24 | poolcnt | Number of pools on the lot (if any) | 80.17% | Filled with 0 because no every home has pool. |
| 25 | numberofstories | Number of stories or levels the home has | 77.21% | Deleted the column because every home should have a number of levels and most of them are missing. |
| 26 | airconditioningtypeid | Type of cooling system present in the home (if any) | 68.12% | Filled with 0 because not all home have a cooling system. |
| 27 | garagetotalsqft | Total number of square feet of all garages on lot including an attached garage | 66.84% | Deleted the column. Missing might mean no garage, but there are non-missing cells with 0. |
| 28 | garagecarcnt | Total number of garages on the lot including an attached garage | 66.84% | Deleted the column with the same reason with the previous. |
| 29 | regionidneighborhood | Neighborhood in which the property is located | 60.11% | Deleted the column because it is missing randoly. |
| 30 | heatingorsystemtypeid | Type of home heating system | 37.88% | Filed with 0 because not all home have heating system. |
| 31 | buildingqualitytypeid | Overall assessment of condition of the building | 36.46% | Filled with mean because all home have overall assessment. |
| 32 | propertyzoningdesc | Description of the allowed land uses (zoning) for that property | 35.41% | Filled with "Missing" to treat missing values as another class |
| 33 | unitcnt | Number of units the structure is built | 35.36% | Filled with 1 because 1 is the most frequent value. |
| 34 | lotsizesquarefeet | Area of the lot in square feet | 11.24% | Filled with mean because all home has are of the lot. |
| 35 | finishedsquarefeet12 | Finished living area | 5.18% | Filled with mean because all home has living area. |
| 36 | regionidcity | City in which the property is located (if any) | 2.00% | Filled with the most frequent value because all home is located in city. |
| 37 | fullbathcnt | Number of full bathrooms present in home | 1.31% | Filed with 0 because missing might mean home does not have full bathroom. |
| 38 | calculatedbathnbr | Number of bathrooms in home including fractional bathroom | 1.31% | Filed with 0 because missing might mean home does not have full bathroom. |
| 39 | yearbuilt | The Year the principal residence was built | 0.84% | Filled with mean because all home have the year built in. |
| 40 | calculatedfinishedsquarefeet | Calculated total finished living area of the home | 0.73% | Filled with mean because most home have living room |
| 41 | censustractandblock | Census tract and block ID combined | 0.67% | Filled with most frequent value because every home has it's value |
| 42 | structuretaxvaluedollarcnt | The assessed value of the built structure on the parcel | 0.42% | Filled with mean because every home has the assessed value |
| 43 | regionidzip | Zip code in which the property is located | 0.04% | Filled with the most frequent value because every home has zip code. |
| 44 | taxamount | The total property tax assessed for that assessment year | 0.01% | Filed with mean because every home has property tax. |
| 45 | taxvaluedollarcnt | The total tax assessed value of the parcel | 0.00% | Filed with mean because every home has property tax. |
| 46 | landtaxvaluedollarcnt | The assessed value of the land area of the parcel | 0.00% | Filled with mean because every home have assesed value. |
| 47 | propertycountylandusecode | County land use code i.e. it's zoning at the county level | 0.00% | Filled with the most frequent value. |

**Outliers**

Let's draw a scatter plot on "logerror", then we can find that there are some outliers at the end of both sides.

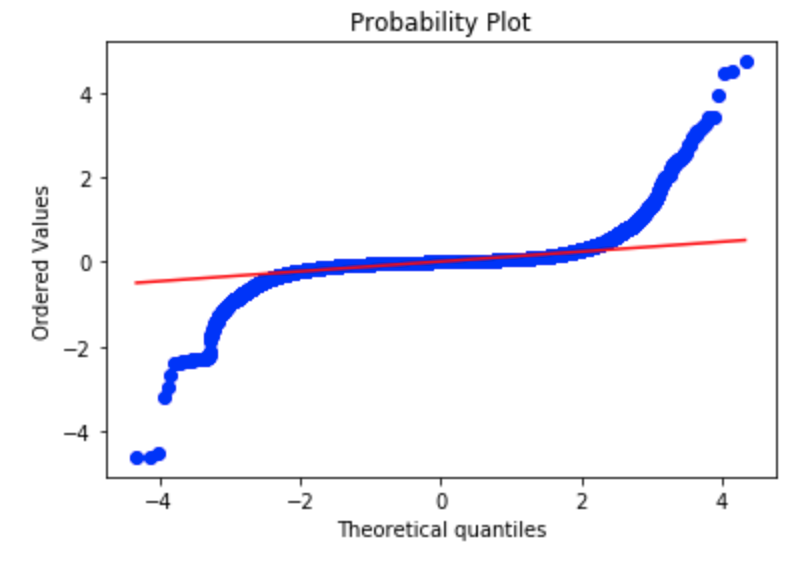
Our task in the project is to find where the zillow algorithm fails. These outliers means where the zillow algorithm fails the most. Thus, I will leave outliers just like that.

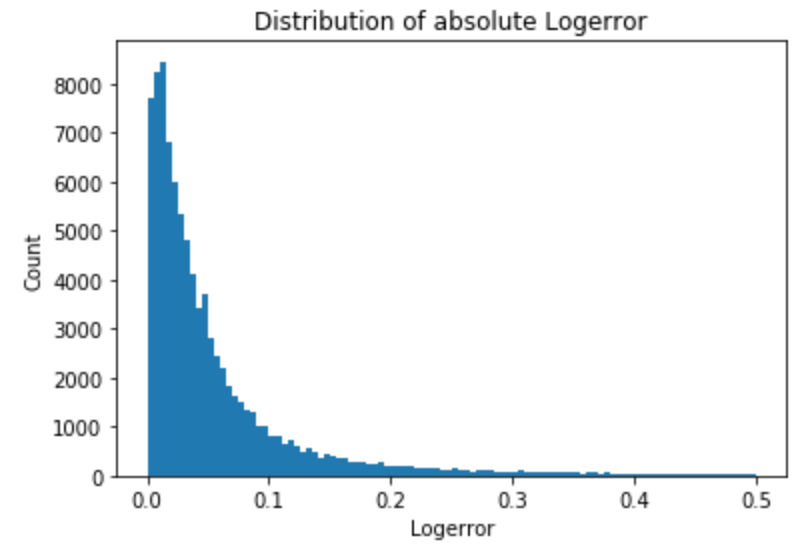
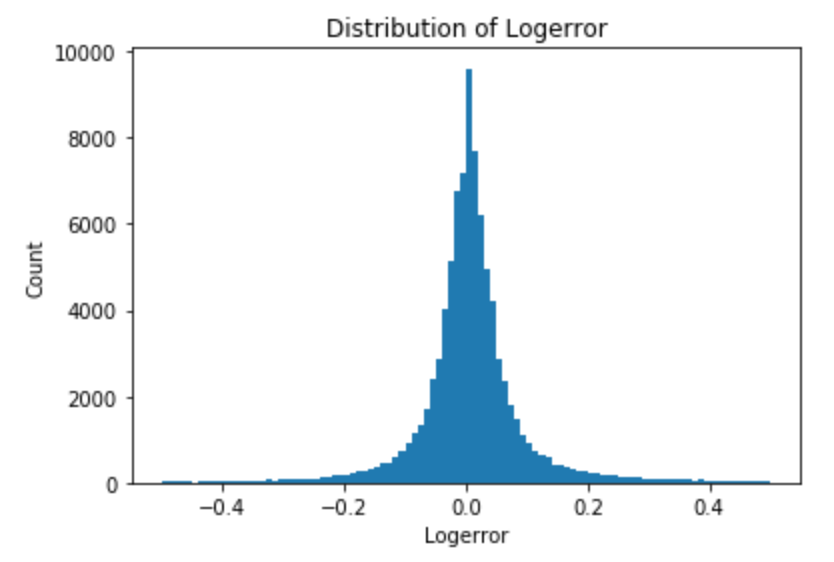


**Data Storytelling**

**Distribution of Logerror**

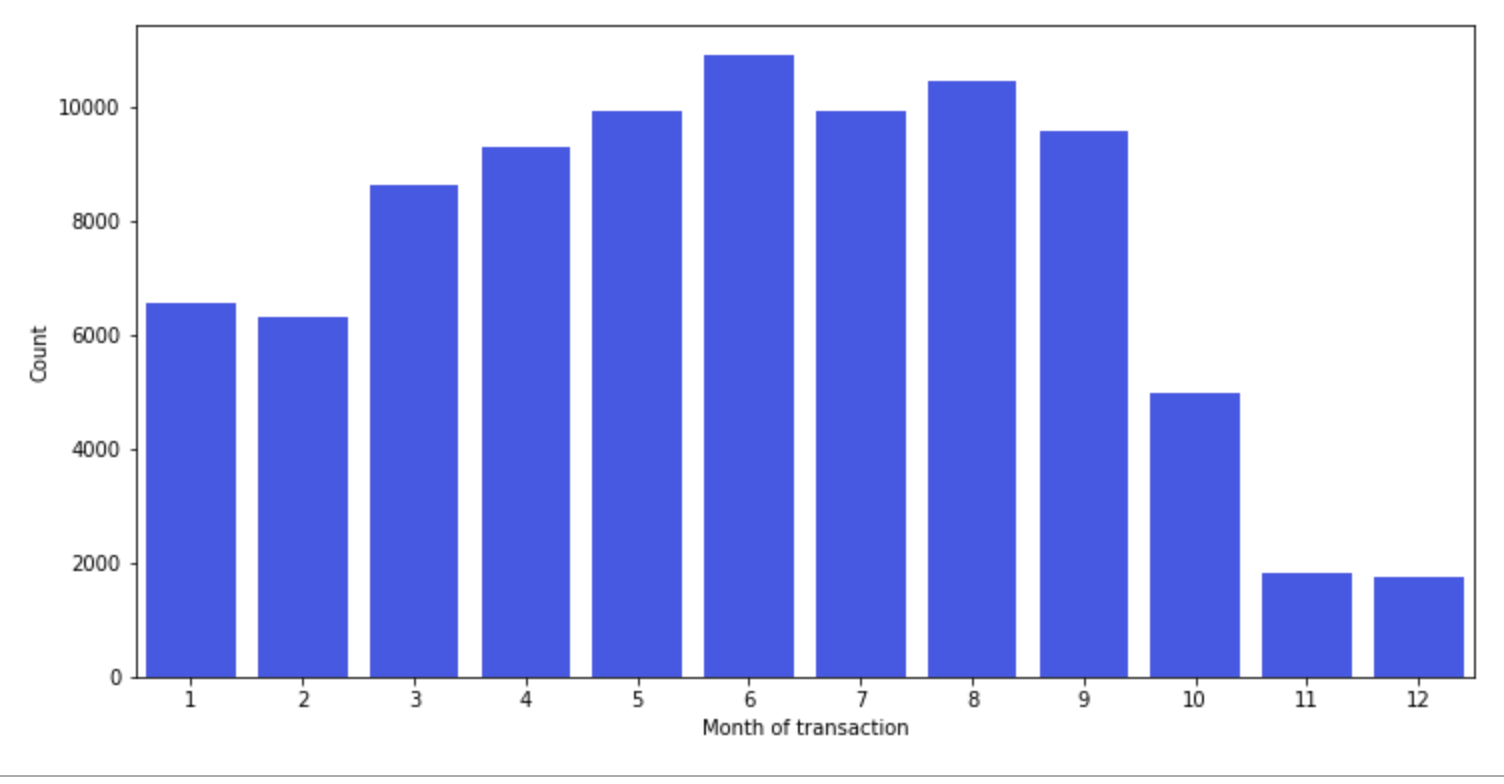
We would check both logerror and absolute value of logerror. Logerror indicates wheather estimated house values has been underestimated or overestimated while absolute logerror tells us that how estimated house value is close to an actual house value. It seems like the distribution of logerror follows a normal distribution by checking QQ plot.



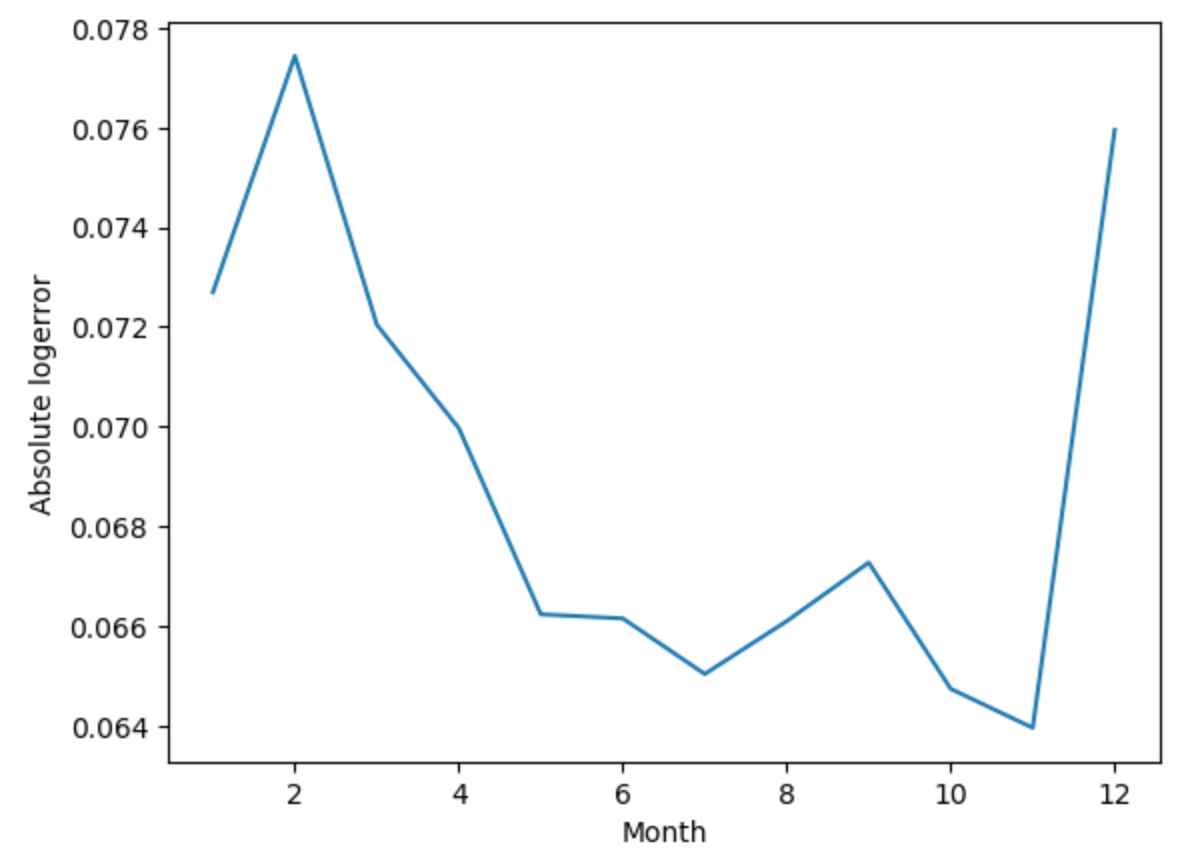


**Transaction Dates**

Let's check the distribution of transaction dates, there are fewer transactions after October

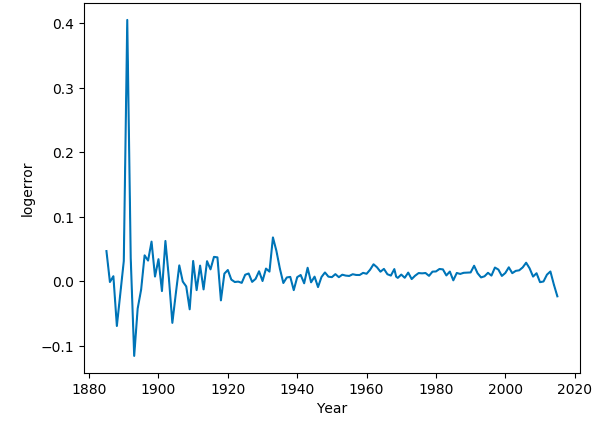
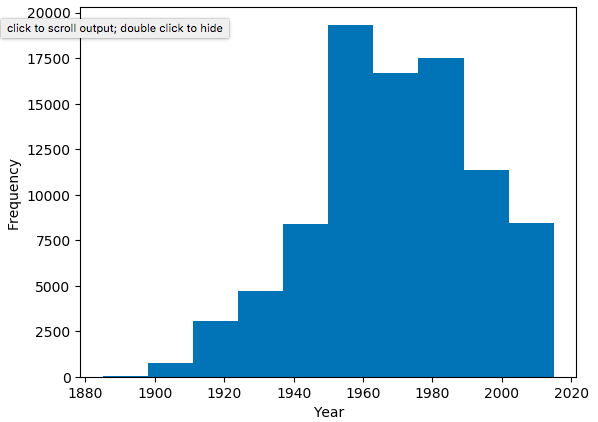


Let's see how absolute logerror change over time. We can see that logerror is getting better over time. The large logerror on December is because there were few transaction at that time. Small quantity of data leaded to large mean of logerror.



**Built year**

After observing plot for density of built year, we can find that most houses, 59.37%, are built between 1950 and 1990. Logerror is getting smaller with newer houses. Zestimate predicts home value with newer homes.

****

**Inferential Statistics**

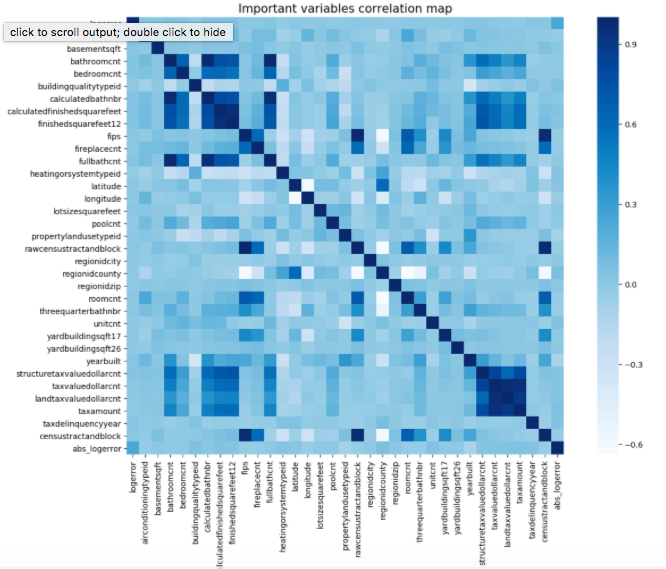
Let's check correlations of each variables to "logerror" to see how variables are related.

Correlation between target variable, logerror, and dependent variables are all weak. They are between 0.237380 and -0.018009.

|  |  |
| --- | --- |
| Name | Coeff |
| abs\_logerror | 0.23738 |
| finishedsquarefeet12 | 0.039248 |
| calculatedfinishedsquarefeet | 0.038341 |
| calculatedbathnbr | 0.028788 |
| bathroomcnt | 0.027889 |
| fullbathcnt | 0.027571 |
| bedroomcnt | 0.025467 |
| structuretaxvaluedollarcnt | 0.021935 |
| taxdelinquencyyear | 0.018107 |
| yearbuilt | 0.017089 |
| basementsqft | 0.009019 |
| rawcensustractandblock | 0.008376 |
| fips | 0.008363 |
| fireplacecnt | 0.007746 |
| taxvaluedollarcnt | 0.006508 |
| roomcnt | 0.00576 |
| threequarterbathnbr | 0.00549 |
| airconditioningtypeid | 0.005404 |
| latitude | 0.004915 |
| lotsizesquarefeet | 0.004612 |
| censustractandblock | 0.004495 |
| yardbuildingsqft17 | 0.002497 |
| propertylandusetypeid | 0.001003 |
| regionidcounty | 0.000341 |
| yardbuildingsqft26 | -0.000846 |
| regionidcity | -0.002342 |
| landtaxvaluedollarcnt | -0.003051 |
| longitude | -0.003432 |
| unitcnt | -0.003447 |
| regionidzip | -0.006487 |
| taxamount | -0.006671 |
| buildingqualitytypeid | -0.00788 |
| poolcnt | -0.008983 |
| heatingorsystemtypeid | -0.018009 |

Let's check correlations between pairs of independent variables. We can find that there are 2 clusters on the heat map below. The first cluster on the top left shows that variables about sizes of houses such as bathroom size or bedroom size and total square feets are stronly related. The second cluster on the bottom right tells us that variables about taxes are related to each other. Alse, we can observe from the right top cluster that variables about sizes of houses are weakly related to variables about taxes. It is reasonable because the bigger a house is, the more expensive the property is resulting the more taxes. However, the price of house is not only resulted from the size of house. So correlation is not strong among them.

There are not variables which can be particularly significant in terms of predicting logerror based on correlation. Also, there are strong multicollinearity between dependent variables. Therefore, a linear regression is not suitable for the model because of multicollinearity.



**Machine Learning**

**Random Forest**

As we see on the above, a linear regression is not a good choice for a model because of multicollinearity. I first tried a random forest as multicolinearity is not important factor for random forest.

To find the best fitted randome forest model, grid search is used. possible combination of options were applied to find the better model. From grid search, the model with max\_depth of 5, min\_samples\_spli of 20 and n\_estimators of 30 was selected.

The RSME for the model was 0.0011. So random forest is a good to predict logerror.

**Lasso**

Let's try Lasso. To find the alpha for Lasso try many possible variables, 0,0.0001,0.001, 0.01,0.1,0.5,1,2,3,4 for alpha, then choose the suitable variables. Also, R squared is too low.

|  |  |  |
| --- | --- | --- |
| Alpha | RMSE | R squared |
| 0 | 0.0258 | 0.0063 |
| 0.0001 | 0.0258 | 0.0062 |
| 0.001 | 0.0258 | 0.0055 |
| 0.01 | 0.0258 | 0.0047 |
| 0.1 | 0.0258 | 0.0046 |
| 0.5 | 0.0258 | 0.0045 |
| 1 | 0.0258 | 0.0044 |
| 2 | 0.0258 | 0.0040 |
| 3 | 0.0259 | 0.0034 |
| 4 | 0.0259 | 0.0029 |

11 variables were chosen for Lasso, but coefficients for each chosen variables are low. Therefore Lasso is not a good model to predict logerror

|  |  |
| --- | --- |
| coeff | name |
| 7.13E-08 | taxvaluedollarcnt |
| 4.43E-08 | rawcensustractandblock |
| 4.46E-09 | lotsizesquarefeet |
| 3.76E-09 | latitude |
| 1.29E-09 | longitude |
| -4.12E-14 | censustractandblock |
| -4.78E-09 | regionidcity |
| -1.34E-08 | regionidzip |
| -1.40E-08 | structuretaxvaluedollarcnt |
| -5.63E-08 | landtaxvaluedollarcnt |
| -2.22E-06 | taxamount |

**Support Vector Machines**

Support Vector Machine is not significant for a model.