Zestimate Error Prediction Project Final Report

Problem Statement

In 2017, Zillow released “Zillow Prize,” a competition challenging participants to beat Zillow’s *Zestimate*, a predictor of home future sale prices. The competition data is still open to the public although it finished some time ago.

Homes are typically the most expensive purchases any individual will make, and at the time of the competition, Zillow was actively predicting the sale price of over 100 million homes in the United States. Zillow started this competition in hopes of gaining insights to improve its Zestimate algorithm and reduce the log error of actual sale price compared to the Zestimate. An accurate algorithm would help it compete with other real estate marketplaces like Redfin.

Data Wrangling

I used data that Zillow provided on nearly 3 million homes in LA, Orange and Ventura counties in 2016. The data included over 50 features of homes, such as square footage, the number of rooms, the existence of a pool, etc. There was also a separate table that contained records on each home that was sold during 2016, including a transaction date and the log error of its price from the Zestimate. I was able to join the two tables via a Parcel ID.

The data was accompanied by a dictionary that helped to explain the column names, but I renamed several of the columns to make them easier to decipher. I also removed duplicate or redundant columns. For example, there were originally four columns related to the number of bathrooms in each home, and I was able to retain the critical information using only one column.

Many of the features had null values, and there were several categorical features, including some that appeared numerical at first, such as land use codes. I identified each categorical column and kept a record of them to prepare for one-hot

encoding. In some cases, I needed to reduce the number of unique values in a column in order to avoid having excessive columns after one-hot econding. For example, the data included nearly 2000 “zoning codes” but fewer than 600 of those codes appeared more than 10 times. In most cases, I tried to reduce to number of unique values in each of my categorical columns to less than 200.

Many of the features had null values, some as high as 98% null. For categorical features, I typically set null values to ‘not given.’ For numerical features that had only a few missing values, I often imputed the median value. In some cases, imputing zero made more sense. For example, garage square footage and garage car count had null values that I determined were due to the fact that property did not have a garage, so I set those null values to zero. I needed to a drop a few columns, such as Entry Floor Square Feet, because it was over 95% null values.

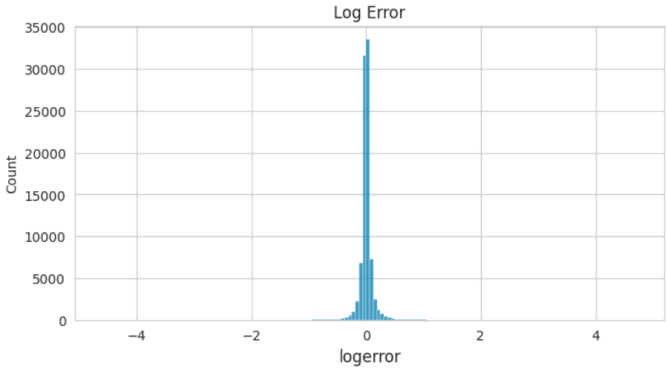
There were some numerical columns related to pools and patios that had a high percentage of null values but I felt that the data was important, when present. I decided to convert these columns into categorical bins and I included a ‘not given’ category.

After data wrangling, there were 19 numerical columns, 30 categorical columns, and over 90,000 records.

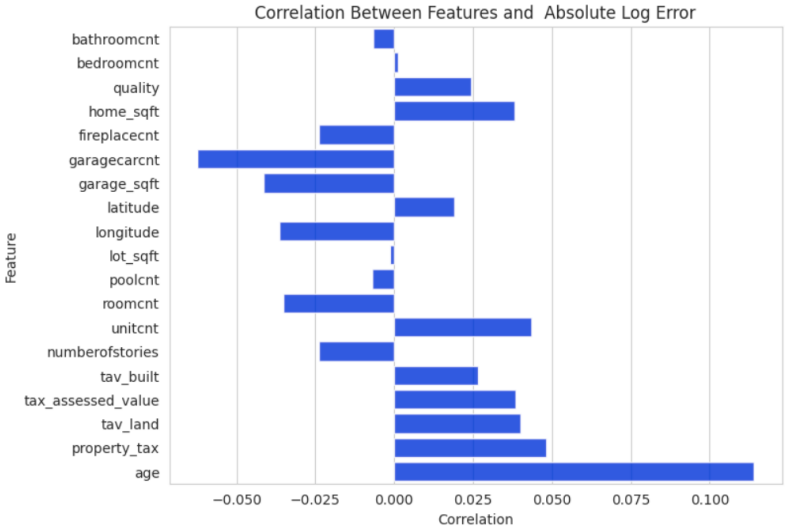
Exploratory Data Analysis

During exploratory data analysis I checked the distribution of ‘log error’ which Zillow had already calculated as *log error = log(Zestimate) - log(Sale Price)*. The log error was a very tight bell curve centered at zero.

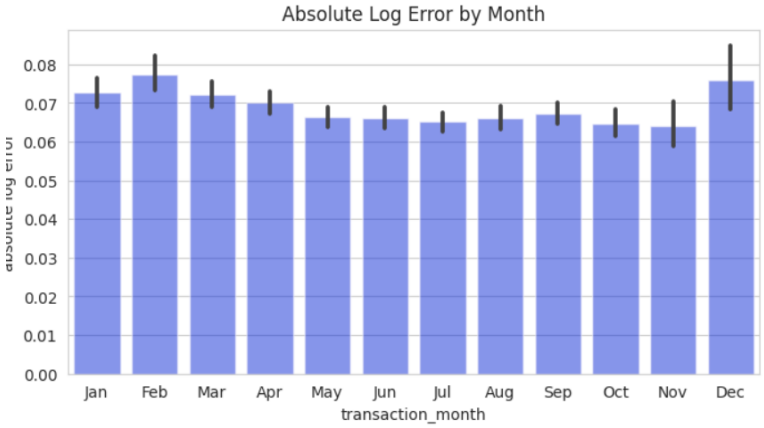
I also calculated the absolute value of the log error to help identify any features that were correlated with inaccurate Zestimates.



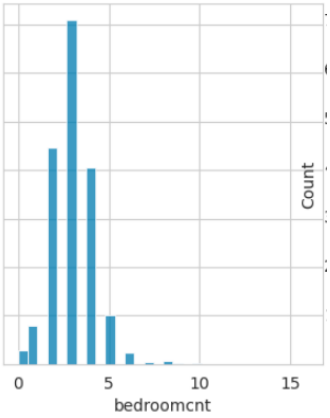
Linear correlations between absolute log error and the features were small, but a few features such as age and property tax value stood out.



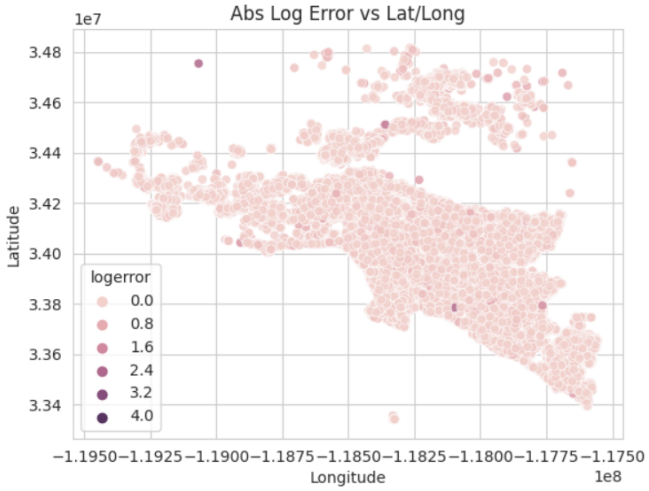
There also appeared to be seasonality to the data, although we only had data from the year 2016. Typically, the Winter has fewer residential real estate transactions. Absolute log error was highest in the Winter months.



Most of the features had distributions that were skewed to the right. For example, most homes had around 3 bedrooms, but there were a few homes with over 15.



I also had data on the latitude and longitude of each property, but there was no obvious correlation between location and absolute log error. The data was from three counties near Los Angelos, so a plot of latitude and longitude closely resembled the southern California coastline.



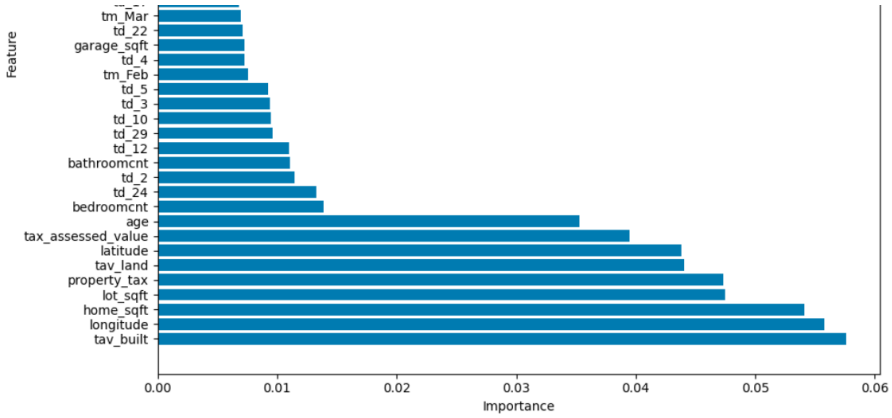
Modeling

I one-hot encoded the categorical columns and created a scaled version of the data that I could use for distanced-based models.

I tested four different types of regression models: Linear Regression, K-Nearest Neighbors, Random Forest, and XG Boost. I tried out different versions of each model, using different features or tuning the hyperparameters differently. The Random Forest Model performed the best after tuning, but interestingly, it was not much more accurate than a basic Linear Regression that used only the original numerical columns.

Model Name	RMSE	MAE
Linear Regression with dummies	15042246901	177931586.5
Linear Regression Numeric Only	0.15452	0.06718
KNN Regression	0.16868	0.08334
KNN Regression Numeric Only	0.16721	0.08199
Random Forest Default	0.15860	0.07277
Random Forest Tuned	0.15445	0.06706
XG Boost	0.15546	0.07051
XG Boost DMatrix	0.15755	0.07142
XG Boost Tuned	0.15504	0.06798

Plotting the feature importance showed that there was a steep drop off in feature importance, and that none of the columns created by one_hot_encoded were a part of the most relevant features.



Future Research

It would be interesting to see if the Random Forest and XG Boost models perform better when only using the most important features. Many of the most important dummy variables were related to the date of the house sale, so it would be worth exploring alternative ways to provide the models with that data. In addition, more background information about what the different land use or zoning codes mean could help to group them into categories that are more helpful to the model.