Kaggle Username: kimkaminsky
Kaggle Display Name: Kimberly Kaminsky

Kimberly Kaminsky
Predict 413 Section 55
Midterm – Zillow Kaggle Competition

Zillow Kaggle Competition

Problem Definition and Significance

Zillow presented a competition on the Kaggle website which challenged participants to help Zillow improve the accuracy of its home price predictions. On it's website, Zillow produces "Zestimates" that are estimated home values. Since a home is often the largest and most expensive purchase a person makes, ensuring that homeowners have a trusted way to monitor this asset is incredibly important. The Zestimate was created to give consumers as much information as possible about homes and the housing market. In this competition, the challenge is to build a model to improve the Zestimate residual error. Determining the predictors of log error should enable Zillow to pinpoint the problems in their models, which will then help them improve their models. For each property provided the log error was predicted for each of six time points: October 2016, November 2016, December 2016, October 2017, November 2017, and December 2017.

Data Exploration

There were four datasets provided for this competition: properties_2016, properties_2017, train_2016, and train_2027. The properties_2016 and properties_2017 datasets each consisted of 2,985,217 properties with 58 predictor variables for each of the properties. The train datasets tracked the errors associated with each property sale and consisted of a parcel id, a log error, and a transaction date. The train_2016 dataset had 90,275 observations and the train_2017 dataset had 77,613 observations. Some of the variables had unclear names and so the variables were renamed to provide clarity and consistency. Appendix A provides an overview of the variables in the properties datasets.

In order to train the data, the properties and train datasets were merged on the parcel id creating two files with 90,275 and 77,613 rows. An initial inspection of the data indicated that in the 2016 dataset 42 of the 59 variables were missing at least some of the values. In the 2017 dataset, 52 out of the 59 variables were missing some values. If more than half the values were missing then a decision was made to excluded them from the dataset. This excluded 24 of the 59 potential predictor variables. This created two new datasets, Train and Train17.

An initial analysis of the correlations between the predictor variables and the response variable were conducted at this point. Correlation matrices were constructed to provide a visualization of these relationships. For the correlation matrix, the logerror was converted to absolute log error, abs_logerror, to simplify the analysis. Figure 1, below, displays the correlation matrix for the counting variables, or those variables starting with num_ and for the area variables, or those variables starting with area_. All rows with missing variables were excluded for this analysis. Num_bedroom, num_bedroom_calc, num_bath, and num_garage look promising as predictors of abs_logerror. Num_room, num_unit, area_lot, area_total_calc, and area_live_finished also show some correlation, but just not as strong.

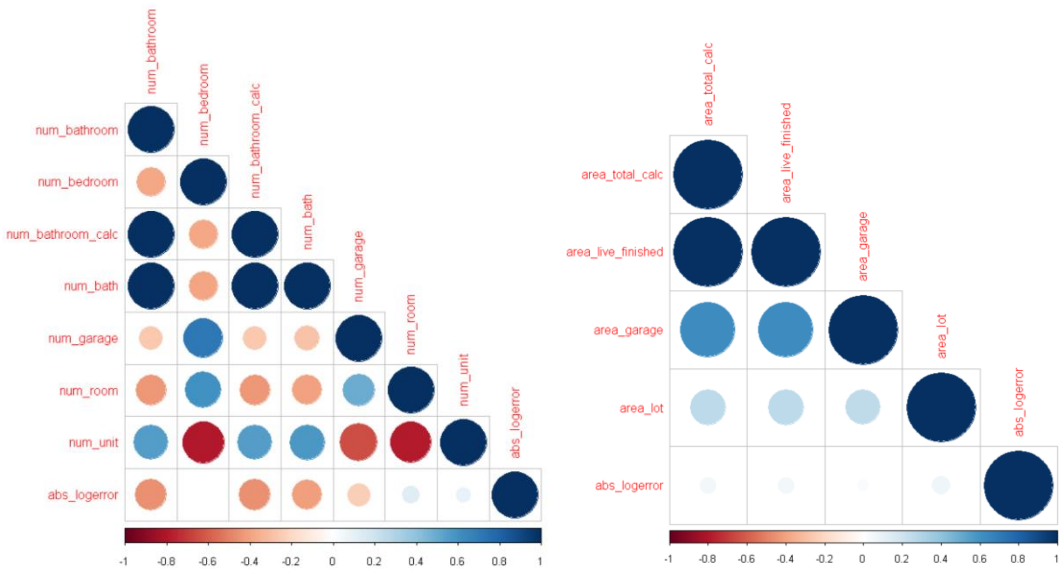

Figure 1:Correlation Matrix for num_ and area_ variables and abs_logerror

Next, the remaining variables with missing values were imputed utilizing decision trees with a few exceptions. Exceptions include region_zip which was imputed using a geocode locator, in the ggmap library, using the longitude and latitude values. Region_city was imputed using a lookup based on current values of the zip code. Censustractandblock was imputed utilizing the rawcensustractandblock and just appended zeros to the end. Num_bath was imputed as the floor of num_bathroom. Longitude and latitude used the most frequently occurring values to impute the value. Train17 had a number of variables, including tax_land and num_bathroom, that had missing values where Train did not have missing values. Since the data often didn't exist to use the decision tree imputation, the median value of the variable was taken to impute the value. Some variables had values that didn't make sense, such as zero number of bedrooms. In these cases, a decision tree was used to impute a better value.

Computed variables were added to the dataset. One variable, houseAge, was created utilizing the transaction date less the year built. Tax_AgeDeliquency was created using the year of the transaction date minus the tax delinquency year. After creating these variables, the build_year and tax_delinquency_year variables were deleted. A month, MO, and day of the month, Day, variable was created by parsing out the values from the transaction date field from the train datasets. Since the object of this competition was to predict the log error of the estimates of the housing values, it seemed logical that missing values could be important predictors. Therefore, for every variable that was dropped, a missing variable indicator flag variable was created. These all used the same naming convention, which was m_ prepended to the original variable name.

Once the data was cleansed, correlation matrices were constructed again. Figure 2 shows the correlation matrix for those variables starting with num_ and area_. Unlike the previous correlation matrix, there are no strong correlations with abs_logerror for any of the original variables. However, the variables that show some correlation are m_num_bath, m_num_bathroom_calc, m_num_unit, m_num_garage, m_area_live_finished, m_area_total_finished, and m_area_garage. This seems to indicate that the missing variable indicators not only have some predictive ability, but that the subsets of rows of data where the variables that do not have missing values may have different predictive ability than the rows that do have missing values.

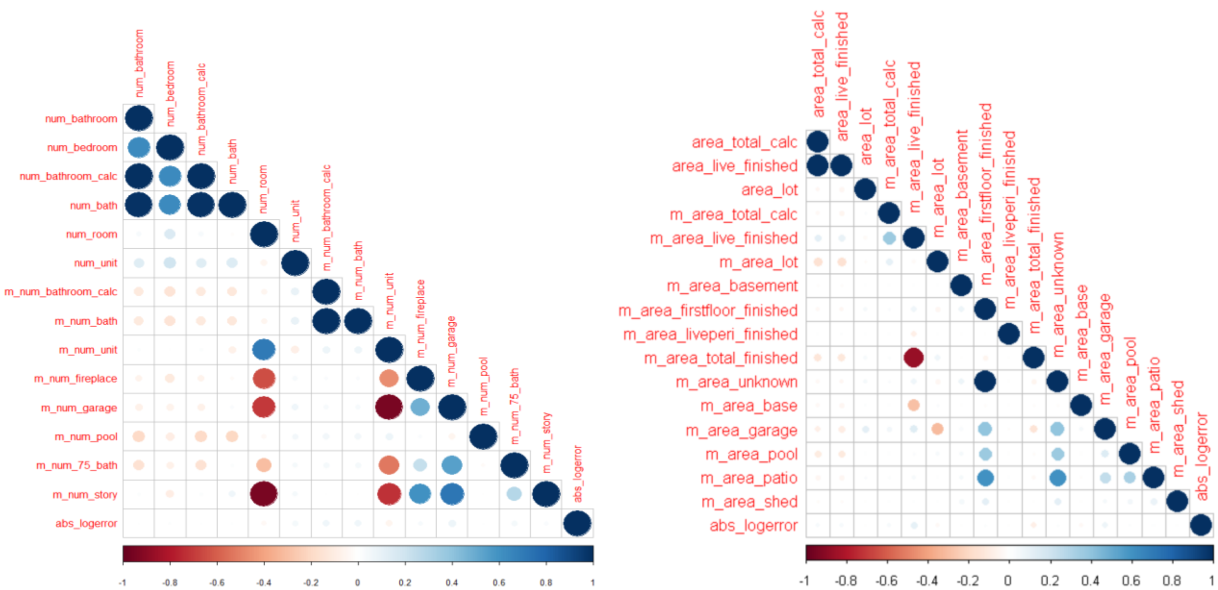

Figure 2: Correlation matrix for num_ and area_ variables with all missing values imputed

The categorical variables were analyzed utilizing box plots ordered by mean value. Figure 3 below shows the box plots for the absolute log error by region_zip. Some zip codes have much larger outliers than other zip codes. Also, as can be seen from the blue bars, there is some variance in the interquartile range between zip codes. This indicates that this variable may be a promising predictor variable.

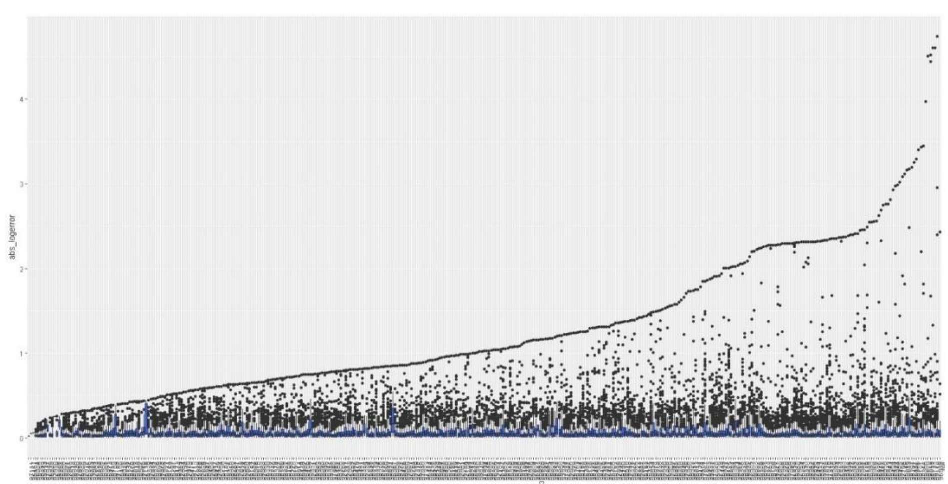

Figure 3: Box plots for zip codes by absolute log error

Model Selection\Implementation

Five types of models were developed to produce predictions on the Zillow data. All work was done in the R programming language. Since boosted regression trees have been successfully applied for mass appraisal of residential properties in Malaysia as a tool to set taxation valuation (McCluskey et al. 2014), generalized boosted regression modeling was utilized by applying the gbm algorithm from the gbm library in R. Goerss found that multiple linear regression was useful in predicting forecast error (Goerss et al. 2014). Also, Yang, Liu, Xu and Zhao also found that semi-supervised regression was helpful in predicting housing prices (Yang et al. 2016). An MLR model was created based on these findings although the semi-supervised approach was not used at this time. Limsombunchai discovered that artificial neural networks, ANN, are superior to hedonic regression models when predicting house prices so an ANN model was constructed (Limsombunchai 2004). Guo found that an ARMA model creates an excellent forecast for short term housing prices (Guo 2012). In order to add in the property attributes, an Arima model, with xreg components, was created. Finally, many Kaggle winners have used xgboost so a model was built using this methodology.

For each method, a model was built utilizing information gleaned from the correlation matrices to do variable selections. Further tweaks to the variables were applied based on the results depending on model results. Naturally, different sets of variables were used for the Train and Train17 datasets, which produced two models for each method. Also, some models were built utilizing variable selection based on the variable importance as determined by the Angoss software tool. The models trained using the Train data were applied to the Oct, Nov, and Dec 2016 time points for the final test set submitted to Kaggle. Likewise, the Train17 data was applied to the Oct, Nov, and Dec 2017 time points for the final test set.

Model Performance

The Kaggle competition evaluated the results based on Mean Absolute Error, MAE, between the predicted log error and the actual log error of the property sale price. Table 1 compares the MAEs between the best model produced using each of the different modeling methods for the Train 2016 dataset, the Train 2017 dataset and the test data set submitted to the Kaggle competition. Although, both the ARIMa and the XGBoost models appear to have better performance according the MAE, the Boosted Regression model performed the best on the test data that was submitted to Kaggle.

Modeling Method	Train 2016	Train 2017	Kaggle
MLR	0.0683	0.0707	0.0651912
Boosted Regression	0.0681	0.0703	0.0648929
ARIMA	0.0121	0.0149	0.0667389
ANN	0.0684	0.0708	0.0657513
XGBoost	0.06666	0.0699	0.0653673

Table 1: Model Comparison of Mean Absolute Error against training dataset and final Kaggle result

Model Limitations

Each of the models above have their limitations. MLR provides and intuitive and easy to understand model, but requires a great deal of feature engineering to get the best results. As can be seen by the results above, it was the 2nd to worst performer. ARIMA is also straightforward but it's training results did not provide a good indicator of its test performance. Also, it provided the same log error prediction for every property, because there wasn't a way to differentiate the properties since each property doesn't have a time series and so aggregation must be performed to use this model. ANN performed badly and it is a black box so it is difficult to understand what is going on. Boosted regression and XGBoost can provide an output to explain the models but it unwieldy and difficult to understand. However, these two models provided the best performance.

Future Work

The first thing that can be done to improve these models is to break out each of the training datasets into train\test and do some cross validation before creating the test file submitted to Kaggle. This would probably improve the performance, or at least the predictability of the results of all the models, but especially the ARIMA model where the results didn't come close to what was submitted to Kaggle. All the models could be improved by better feature engineering. Most of the categorical variables could not be included because of memory and time constraints when running the models. Collapsing these variables into fewer categories would allow these to be used in the models. Another idea is to create some interactions terms. As was noted earlier, the missing variable flags became extremely important as predictors once all missing values were imputed. It is likely there are some interactions there that can be used. A more careful look at the data to determine normality and identify outliers would identify opportunities for variable transformation. Further, no dimension reduction was applied to this data and this would, also, likely provide some benefit. One thing that was tried was tuning the model parameters on the boosted regression. This improved the performance of the model. However, the tuning was more of a guess and a more careful analysis of the tool and what the parameters mean could yield meaningful improvements. This could be applied to the ANN, Boosted Regression and the XGBoost models. Finally, it would be worthwhile to try out an ensemble approach that combines all the feature of these models.

Conclusions

The object of this competition was to build a model that minimized the mean absolute error in predicting the log error of the Zestimate of the property sale price. Some basic data cleanup which included imputing missing values using decision trees and trying out a variety of models yielded a decent result, which placed 2485 in the Kaggle competition. This beat out 1400 other competitors. The model that provided the best results on the test set submitted to Kaggle was the Boosted Regression model which makes sense as this type of model seeks to minimize error through building a series of weak trees and this allows it to produce a good model without too much feature engineering. XGBoost, which often produces good results in competitions, and performed better than the Boosted Regression in the training data did not provide the best results on the test data. This may be due to overfitting. Overall, all the models did a good job with an MAE ranging from about .066 to .6489. Which means there isn't a great deal of difference between the models. All the models can likely be improved either through feature engineering, better training, or parameter tuning. In order to make a definitive conclusion about which type of model performs best on this type of problem, a great deal more rigor and experimentation would be needed.

References

Goerss, James S. & Sampson, Charles R. (2014). Prediction of Consensus Tropical Cyclone Intensity Forecast Error. *Weather and Forecasting*, Vol 29, 750-762.

Guo, Jianhua. (2012). Housing Price Forecasting based on StochasticTime Series Model. *Int.J.Buss.Mgt.Eco.Res*., Vol 3(2), 498-505.

Limsombunchai, Visit. (2004). House Price Prediction: Hedonic Price Model vs. Artificial Neural Network. *American Journal of Applied Sciences*, Vol 1(3), 193-201.

Yang , Yi., Liu, Jiping., Xu, Shenghua., and Zhao, Yangyang. (2016). An Extended Semi-Supervised Regression Approach with Co-Training and Geographical Weighted Regression: A Case Study of Housing Prices in Beijing. *International Journal of Geo-Information*, Vol 5(1), 4.

McCluskey, William J., Daud , Dzurllkanian Zulkarnain & Kamarudin, Norhaya. (2014). Boosted regression trees: An application for the mass appraisal of residential property in Malaysia. *Journal of Financial Management of Property and Construction,* Vol. 19(2), 152-167.

Kimberly Kaminsky

Appendix A

Feature	Description
aircon	Type of cooling system present in the home (if any)
architectural_style	Architectural style of the home (i.e. ranch, colonial, split-level, etc…)
area_basement	Finished living area below or partially below ground level
num_bathroom	Number of bathrooms in home including fractional bathrooms
num_bedroom	Number of bedrooms in home
Quality	Overall assessment of condition of the building from best (lowest) to worst (highest)
Framing	The building framing type (steel frame, wood frame, concrete/brick)
num_bathroom_calc	Number of bathrooms in home including fractional bathroom
Deck	Type of deck (if any) present on parcel
num_75_bath	Number of 3/4 bathrooms in house (shower + sink + toilet)
area_firstfloor_finished	Size of the finished living area on the first (entry) floor of the home
area_total_calc	Calculated total finished living area of the home
area_base	Base unfinished and finished area
area_lived_finished	Finished living area
area_liveperi_finished	Perimeter living area
area_total_finished	Total area
area_unknown	Size of the finished living area on the first (entry) floor of the home
Fips	Federal Information Processing Standard code
num_fireplace	Number of fireplaces in a home (if any)
flag_fireplace	Is a fireplace present in this home
num_bath	Number of full bathrooms (sink, shower + bathtub, and toilet) present in home
num_garage	Total number of garages on the lot including an attached garage
area_garage	Total number of square feet of all garages on lot including an attached garage
flag_tub	Does the home have a hot tub or spa
Heating	Type of home heating system
latitude	Latitude of the middle of the parcel multiplied by 10e6
longitude	Longitude of the middle of the parcel multiplied by 10e6
area_lot	Area of the lot in square feet
num_story	Number of stories or levels the home has
parcel_id	Unique identifier for parcels (lots)
num_pool	Number of pools on the lot (if any)
area_pool	Total square footage of all pools on property
pooltypeid10	Spa or Hot Tub
pooltypeid2	Pool with Spa/Hot Tub
pooltypeid7	Pool without hot tub
zoning_landuse_county	County land use code i.e. it's zoning at the county level
zoning_landuse	Type of land use the property is zoned for
zoning_property	Description of the allowed land uses (zoning) for that property
rawcensustractandblock	Census tract and block ID combined - also contains blockgroup assignment by extension
censustractandblock	Census tract and block ID combined - also contains blockgroup assignment by extension
region_county	County in which the property is located
region_city	City in which the property is located (if any)
region_zip	Zip code in which the property is located
region_neighborhood	Neighborhood in which the property is located
num_room	Total number of rooms in the principal residence
Story	Type of floors in a multi-story house (i.e. basement and main level, split-level, attic, etc.)
material	What type of construction material was used to construct the home
num_unit	Number of units the structure is built into (i.e. 2 = duplex, 3 = triplex, etc…)
area_patio	Patio in yard
area_shed	Storage shed/building in yard
build_year	The Year the principal residence was built
tax_total	The total tax assessed value of the parcel
tax_building	The assessed value of the built structure on the parcel
tax_land	The assessed value of the land area of the parcel
tax_property	The total property tax assessed for that assessment year
tax_year	The year of the property tax assessment
tax_delinquency	Property taxes for this parcel are past due as of 2015
tax_delinquency_year	Year for which the unpaid property taxes were due