PRICE PREDICTION OF ZILLOW HOUSING DATA

Jinu Kingcy Sebastin

STAT/OR 568 Applied Predictive Analytics

Professor. Jie Xu

GEORGE MASON UNIVERSITY

Background: Zillow

Zillow Group, or simply Zillow, is an online real estate database company that was founded in 2006, and was created by Rich Barton and Lloyd Frink, former Microsoft executives and founders of Microsoft spin-off Expedia.

"Zestimates" are estimated home values based on 7.5 million statistical and machine learning models that analyze hundreds of data points on each property. And, by continually improving the median margin of error (from 14% at the onset to 5% today), Zillow has since become established as one of the largest, most trusted marketplaces for real estate information in the U.S. and a leading example of impactful machine learning.

Motivation:

Can we predict housing prices based off the predictive analysis? If yes, which method gives us the best RMSE/ R^2 values? What are the various factors affecting housing prices in different counties?

Dataset before pre-processing:

Our dataset contains 5.99 million rows and 58 columns, and shows the different features of various houses like number of bathrooms, number of fireplaces etc. The dataset has 57 predictor variables and 1 response variable. The variables in our dataset are as follows:

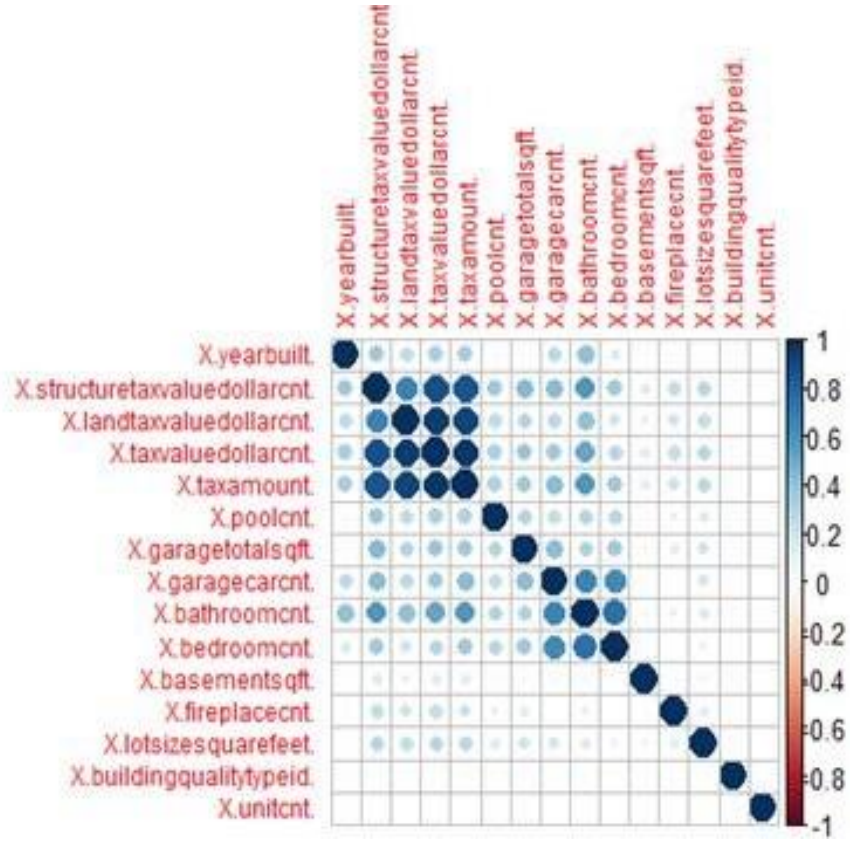
airconditioningtypeid	architecturalstyletypeid	basementsqft	bathroomcnt	bedroomcnt	buildingqualitytypeid
buildingclasstypeid	calculatedbathnbr	decktypeid	threequarterbathnbr	finishedfloor1squarefeet	calculatedfinishedsquarefeet
finishedsquarefeet6	finishedsquarefeet12	fips	finishedsquarefeet15	finishedsquarefeet50	finishedsquarefeet13
fireplacecnt	fireplaceflag	fullbathcnt	garagecarcnt	garagetotalsqft	hashottuborspa
heatingorsystemtypeid	latitude	longitude	lotsizesquarefeet	numberofstories	parcelid
poolcnt	poolsizesum	pooltypeid10	pooltypeid2	pooltypeid7	propertycountylandusecode
propertylandusetypeid	propertyzoningdesc	storytypeid	censustractandblock	regionidcounty	regionidcity
regionidzip	regionidneighborhood	roomcnt	rawcensustractandblock	typeconstructiontypeid	unitcnt
yardbuildingsqft17	yardbuildingsqft26	yearbuilt	taxdelinquencyyear	structuretaxvaluedollarcnt	landtaxvaluedollarcnt
assessmentyear	taxdelinquencyflag	taxamount	<u>taxvaluedollarcnt</u>		

The underlined variable in the above table, namely taxvaluedollarcnt, is the response variable. The dataset contains 18 variables with greater than 90% values with NAs, which we will subsequently deal with during pre-processing.

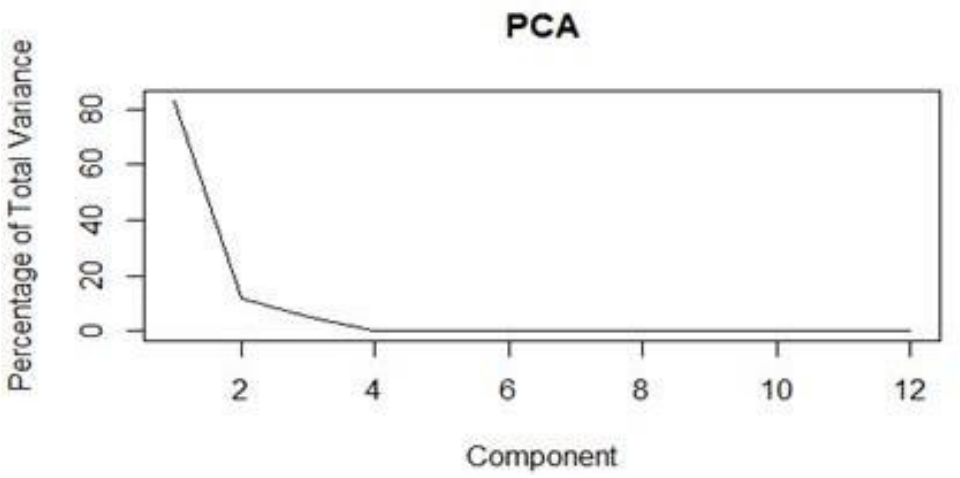
Pre-processing:

First, all the columns with over 90% missing values were removed. This led to the removal of 18 variables from our dataset. Then, the rows were deleted for the variables which had less than 1% missing values. This led to the deletion of about 100,000 rows. The dataset was then split into 3, depending on the Fips code, which divided the dataset based on the county that the houses belong to.

Now, we had 3 datasets, one each for houses in Ventura County, Orange County and LA County in California. We performed subsequent pre-processing steps on each of the 3 datasets separately. The below figure depicts the correlation plot for the dataset.



The variables that have near zero variance for each of the 3 datasets were removed. For variables with high correlation, the modeling was done with all of them, one at a time, and the variable that gave us the best answers was chosen, whereas the other variables were removed. PCA analysis, centering and scaling were performed on the rest of the dataset wherever applicable. The below figure depicts the Scree plot for the dataset.



Dataset after Pre-processing:

The datasets after pre-processing are as follows: The dataset for Ventura county has 430 thousand rows and 15 variables. The dataset for Orange county has 1.4 million rows and 15 variables. The dataset for LA county has 4.4 million rows and 15 variables. The aforementioned datasets were split into training and testing datasets in 4:1 ratio. 10 fold cross-validation was performed on these datasets.

Models:

Several models were used to evaluate the dataset. The Linear models used are Ordinary Linear Regression, Robust Linear Regression and Partial Least Squares method. Tree models used are Boosted Regression Tree and Random Forest. Non-linear models used are K-Nearest Neighbours and Neural Network.

Ordinary Linear Regression Model

Linear models are simple and therefore are preferred first for model evaluation. The objective of ordinary least squares linear regression is to find the plane that minimizes the sum of squared errors between the observed and predicted response. The predictors were centered and scaled prior to modeling to have same units. The linear model was resampled using 10-fold cross validation.

	RMSE	R^2
Ventura county	44935	0.874
Orange county	41525	0.917
Los Angeles county	49682	0.861

It is clear that the RMSE values are too high. This is due to sensitivity of ordinary linear regression models towards any outliers in the dataset. The value of RMSE increases, considering all the values in the datasets.

Robust Linear Regression Model

The high values of RMSE can be avoided using the robust linear regression model as it is insensitive to outliers. The robust linear model function (rlm) from the MASS package is used, which employs Huber approach by default. The training dataset is preprocessed using the pca function. Then, 10-fold cross validation is applied on the dataset that produces reasonable estimates of the model performance.

	RMSE	R^2
Ventura county	44621	0.871
Orange county	39814	0.903
Los Angeles county	46584	0.859

The computation time of robust linear regression model is very high as it performs pca prior to modeling. This achieves best results compared to the ordinary least squares regression models.

Partial Least Squares

If the correlation among predictors is high, then the ordinary least squares solution for the multiple linear regression models will have high variability and will become unstable. Using PLS is recommended when there are correlated predictors and a linear-regression type solution is required. PLS finds linear combinations of predictors, commonly known as components. PLS finds components that maximally summarize the variance of the predictors while simultaneously requiring these components to have maximum correlation with the response.

	RMSE	R^2
Ventura county	43911	0.855
Orange county	36191	0.891
Los Angeles county	45738	0.861

The predictors were centered and scaled prior to modeling. The tuning parameter in this model is number of components. 10-fold cross validation was used to determine the optimal number of PLS components to retain that minimizes the RMSE. The optimal value of tuning parameter is 13. The computation cost of PLS is less compared to the computation cost of RLM, and obtains better results. As a result, the output of PLS is the best of all the Linear models used.

Boosted Regression Trees

Tree-based methods are one of the simplest means to predict and interpret classification and regression problems. With respect to our dataset, we use Boosted regression tree and Random Forest regression trees to predict the tax value amount for each of the three counties - Ventura county, Orange county and Los Angeles county.

Boosting works similar to bagging, except that each of the trees is grown by utilizing the information from the trees that are grown prior to it and are not sensitive to training samples like bagging.

Boosted trees uses three tuning parameters namely, number of trees, shrinkage and interaction depth. The number of trees must be carefully chosen because using a large value for ntree could potentially overfit the boosting model. The shrinkage parameter λ is generally a very small positive number that controls the learning rate of boosting. A small value of λ and a large value for number of trees can achieve good performance. Finally, the interaction depth keeps control of the interaction order for the boosting model. In order to predict the best performance of the model, we have set the tuning parameters as ntree = 1000, shrinkage λ = 0.1 and depth d=7.

We have implemented the boosting tree model on all three counties in R using the package 'gbm' which includes all the functions needed for the Gradient Boosting Machine. The summary provides information about the variable importance and the most important predictor using boosting model. For Ventura county, the figure below predicts the most important predictors as bathroomcnt followed by yearbuilt with RMSE and R^2 value as 40065 and 0.86 respectively.

```
> summary(Boost_ventura)
                                     var     rel.inf
bathroomcnt                   bathroomcnt 37.75854241
yearbuilt                       yearbuilt 32.76807734
parcelid                         parcelid 10.20839618
garagetotalsqft           garagetotalsqft  9.00585466
regionidzip                   regionidzip  3.58221749
bedroomcnt                     bedroomcnt  2.11454960
poolcnt                           poolcnt  2.07056415
lotsizesquarefeet       lotsizesquarefeet  1.67220591
fireplacecnt                 fireplacecnt  0.70217893
basementsqft                 basementsqft  0.09865144
airconditioningtypeid airconditioningtypeid  0.01876190
buildingqualitytypeid buildingqualitytypeid  0.00000000
unitcnt                           unitcnt  0.00000000
```

For Orange county, the figure below predicts the most important predictors as yearbuilt followed by bathroomcnt with RMSE and R^2 value as 37296 and 0.84 respectively.

```
> summary(Boost_orange)
                                     var    rel.inf
yearbuilt                       yearbuilt 49.5604223
bathroomcnt                   bathroomcnt 33.0270666
parcelid                         parcelid  5.8384000
bedroomcnt                     bedroomcnt  4.7265384
regionidzip                   regionidzip  2.5155399
poolcnt                           poolcnt  1.9261702
garagetotalsqft           garagetotalsqft  0.8769987
airconditioningtypeid airconditioningtypeid  0.5691992
fireplacecnt                 fireplacecnt  0.5633279
lotsizesquarefeet       lotsizesquarefeet  0.3963367
basementsqft                 basementsqft  0.0000000
buildingqualitytypeid buildingqualitytypeid  0.0000000
unitcnt                           unitcnt  0.0000000
```

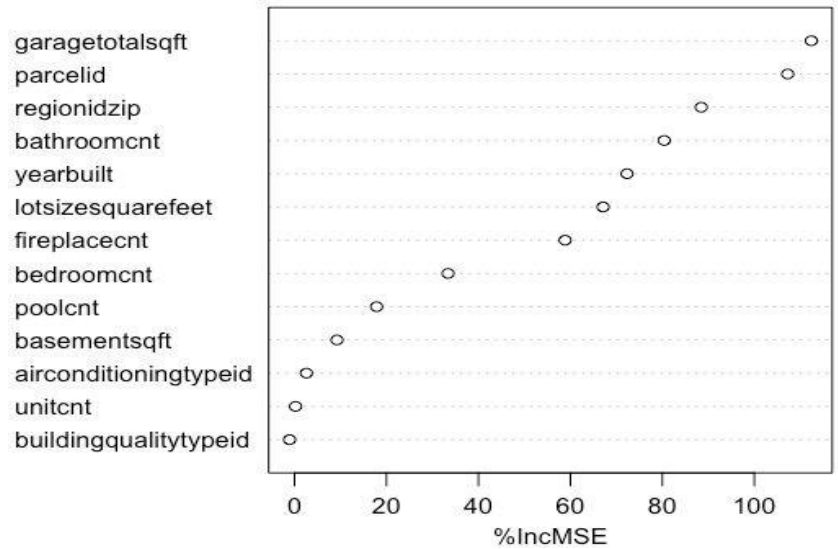
For Los Angeles county, the figure below predicts the most important predictors as bathroomcnt followed by regionidzip with RMSE and R^2 value as 42818 and 0.82 respectively.

```
> summary(Boost_LA)
                                     var     rel.inf
bathroomcnt                   bathroomcnt 34.03879469
regionidzip                   regionidzip 23.45689978
buildingqualitytypeid buildingqualitytypeid 14.32716606
yearbuilt                       yearbuilt 11.23287647
parcelid                         parcelid  7.74507206
lotsizesquarefeet       lotsizesquarefeet  3.69811316
bedroomcnt                     bedroomcnt  2.13217934
airconditioningtypeid airconditioningtypeid  1.92853370
poolcnt                           poolcnt  1.36856337
unitcnt                           unitcnt  0.07180137
basementsqft                 basementsqft  0.00000000
fireplacecnt                 fireplacecnt  0.00000000
garagetotalsqft           garagetotalsqft  0.00000000
```

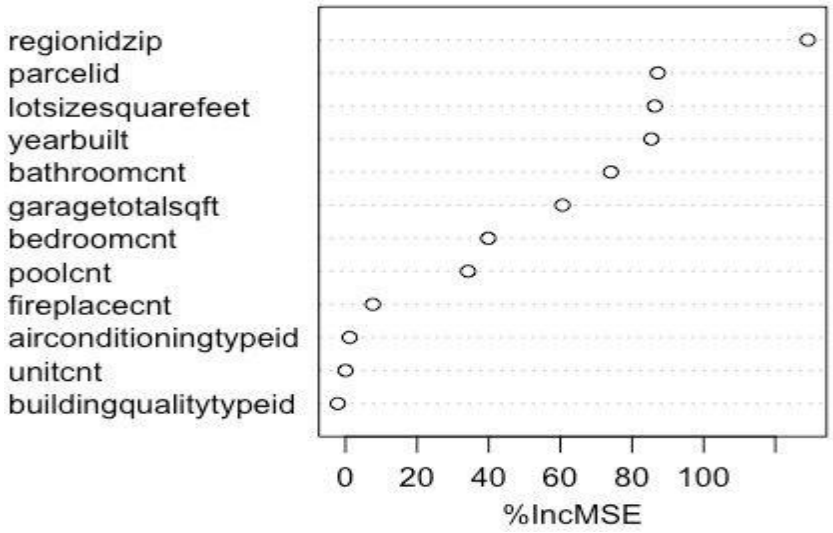
Random Forest

Random forest uses a tuning parameter that is the number of randomly selected predictors, k, to choose from each split and is commonly referred to as mtry. Mtry is random forests' tuning parameter. Mtry is the predictor count or number that are randomly selected (k) from each split. Mtry is recommended to be one-third (1/3) of the total number of predictors. The recommended number of trees to build (m), should be at least 1000. An advantage of the random forest is that it is protected from overfitting, making models unaffected by a large number of trees. However, in this project we are not using a more massive than the average number of trees because of the level of the computational burden this has on our training model. In predicting predictors for home sale prices, random forest reduces variance by selecting the sophisticated and robust learners that show low bias. This technique enables the model to improve its error rates.

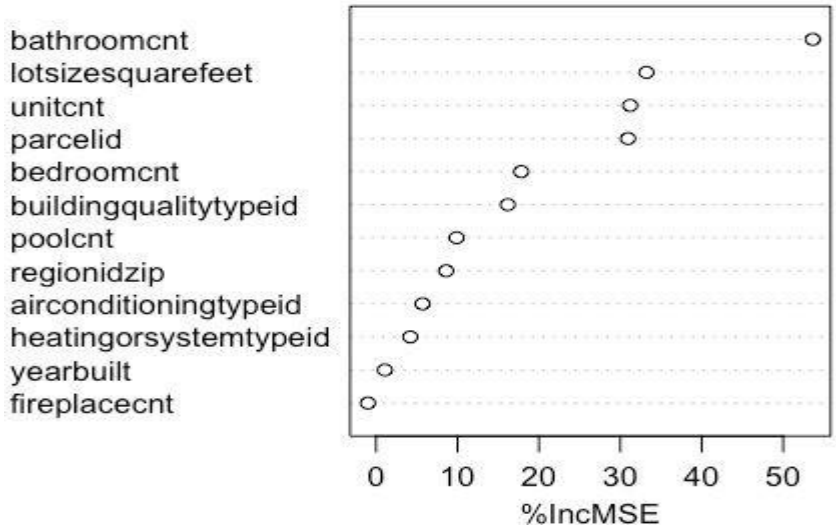
For our project, we used 10-fold cross-validation and out-of-bag validation, to train our model on the Zillow home sale data. The mtry parameter for the model is going to evaluate at twelve(12) values. The figures below displays random forest variable importance values for the predictors of the sale price in Ventura, Orange, and Los Angeles County.



For the Ventura county, garagetotalsqft, parceled, regionidzip, and bathroomcnt are top of the important predictors with RMSE and R^2 value as 38970 and 0.89 respectively.



For Orange county, regionidzip, parcelid, lotsizesquarefeet, and yearbuilt are top of the important predictors with RMSE and R^2 value as 30098 and 0.88 respectively.



For Los Angeles county, bathroomcnt, lotsizesquarefeet, unitcnt, and parcelid are top of the important predictors with RMSE and R^2 value as 41367 and 0.86 respectively. Although Boosting compared to Random Forest is more computationally efficient, for the Zillow house sales prediction dataset, Random Forest regression tree model provided us with better results.

Neural Network

We want to implement a couple of non-linear models to run on the dataset and neural networks were one of the models we picked. The value of this type of model comes from the complexity and its propensity to give accurate predictions. Although there is value in using this model, there is complexity

and also comes with some negatives. Given the "black box" nature of the model there can't be too much to interpret from the actual fitting of the model when ran against the data.

When running the model through the data a couple of issues arose. When using the 'train' function in the 'caret' package model fitting took a long time. This is probably due to numerous factors such as the size of the dataset, complexity of the model, hardware, and that the caret package is more computationally intensive than just the normal neural network model out of R. Due to the time it took to train the model further preprocessing was done using the 'findcorrelation' and 'nearzerovar' functions the already preprocessed predictors were trimmed further to 11 variables. Scaling and centering the data was done manually instead of it being ran within the train function. Due to high training time, tuning was not able to be done for this model. Each training iteration took around 25-40 minutes with minor tweaks of the parameters each time to try to get the most accurate fit.

Results below show the RMSE and R^2 of the neural network ran on the Ventura, Orange County, and LA datasets. Although the RMSE results are good compared to the other models tested and the R^2 values are decent for this problem we found that Neural Networks weren't exactly a great fit for predicting housing prices. One of the main reasons is the computation time. Fitting a neural network to the dataset took a decent amount of time. The R^2 values are good but they are normally overly optimistic for neural net models. The over complexity of neural network models make it hard to take anything away from them after fitting. If there is further work to be done on this model it would be interesting to see if the current fit of the model on this specific years data is reflected and viable over the next couple years.

	RMSE	R^2
Ventura County	32269	0.88
Orange County	35789	0.81
Los Angeles	37433	0.78

K-Nearest Neighbors (KNN)

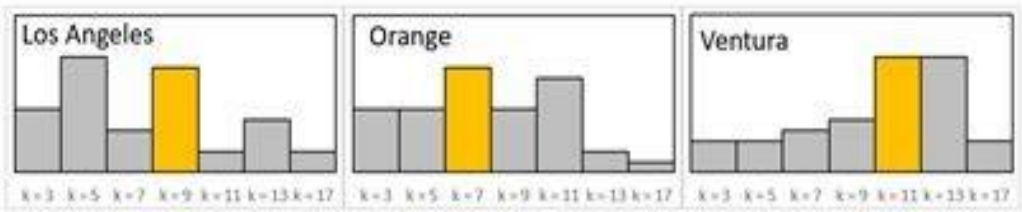
KNN is a unique prediction model compared to most applied in this analysis. The concept of KNN is to calculate the distance of an unknown record to each of a group of known records and estimate the unknown response based on the average of k nearest known responses (or most frequent class in k nearest responses, in the case of a classification problem). KNN model is uniquely suited for some predictions, while other predictions easily accommodated by other models might not be well suited for the model. Examples of common use cases are content retrieval, recommender systems, and commerce valuation (i.e. the house price prediction application in this study). This section will discuss some of the key considerations for implementation of KNN and the process of exploration, tuning, assessing, and running excursions in support of this analysis.

Firstly, KNN is a non-parametric, lazy model. A non-parametric model is a model that does not assume a prior distribution for predictor variables (unlike LDA, for example, which is a parametric model assuming a normal distribution of all predictors). A lazy model is one that does not require a training phase. In fact, the training phase of KNN is used solely for tuning the parameter 'k.' In many cases, the analyst may be able to select a value of k without tuning the model just by understanding the dataset and the noise between records.

KNN also assumes all data is in metric space because the distance between one value of a predictor and another much be able to be calculated. In some cases, this means predictors like zip code can be helpful to a KNN model, where they could be detrimental so some other models. Zip codes work in this scenario because zip codes that are near each other are typically near geographically as well. It is a reasonable assumption that geographically near zip codes could add value in the prediction of house price. A challenge of using KNN is its ineffectiveness in high dimensions. Distance calculations (especially Euclidean distance, used in this model) are susceptible to high computational cost and a phenomenon where, under high dimensions, the distance from a fixed point to the nearest and furthest record are approximately the same. This means the key method of prediction (closest known records) becomes useless. Preprocessing to a reasonable dimensionality before implementing KNN is extremely important.

Predicting the house prices in Zillow data for this analysis began with exploring the data and approximating the performance of the model. In exploration, the key determination was that the full dataset could not be used due to computational cost (1.5M records with 25 predictors). Instead, the predictor space was reduced to 15 dimensions of the most important variables. Then an algorithm was introduced to repeatedly resample the training dataset, so the value of k could be tuned.

Next, the tuning phase was implemented to determine the best value of k. Because of the resampling technique used, the model was trained 30 times for each county with k values ranging from 3 to 17 at intervals of 2. Then the most frequently selected k value was chosen for the assessment of the model, unless a different value of k proved to be more stable (as in the Los Angeles County dataset). Histograms of the k value selected are shown below.



The selected k value was then used to predict the testing set with 20 different samples of the training set. The results are below.

	k_value	Iterations	Avg RMSE	SD RMSE	Avg r2
LA	9	20	39032	1226	0.90
VC	11	20	39073	834	0.91
OC	7	20	39902	889	0.91

The prediction result is promising because the RMSE performance appears to be very stable, which contrasts with other models evaluated. Also, the R^2 value is consistently the best of the models used and the computational cost was very low due to the resampling algorithm used. In conclusion, KNN is the best model yet.

Conclusion

There were a number of lessons learned and conclusions drawn in the process of conducting the analysis in this report. Preprocessing was the most time consuming and significant process to solving this problem. The team found a lot of data can be collected regarding a topic, but not all of it is useful and collected data is never perfect. The decisions made regarding imputation, removal, encoding, and predictor transformations have a large impact on the ability to implement predictor models and receive a valuable result. The team learned predictors will have different levels of importance depending on the predictor model structure. Preprocessing decisions can be well informed many times by subject matter experts that have an intuitive understanding of the data set. Additionally, the exploratory phase of model development is valuable to manage computational cost and can allow the modeler to implement algorithms to make the tuning and evaluation more efficient later in the analysis.

Root mean squared error (RMSE) was used as the primary evaluation parameter in the analysis. R^2 was the secondary parameter and was used to confirm or qualify findings. RMSE is a common model evaluation parameter and only encounters issues when there is a disagreement in units used for different models. All models in this analysis had consistent units.

The team concludes KNN is the most beneficial model evaluated. With implementation of a few new algorithms, the KNN model became the most computationally efficient with near equal evaluation parameters (RMSE and R^2). Neural net is a close second with a result that is slightly more accurate, but less stable and with more computing power required.

Source:

The dataset for the project is obtained from Kaggle. https://www.kaggle.com/c/zillow-prize-1
Data is collected from two large datasets with the size of each file being approximately 600MB.

References:

James, G., Witten, D., Hastie, T.,Tibshirani, R. (2017). An Introduction to Statistical Learning with Applications in R.
Kuhn, M., Johnson. K. (2013). Applied Predictive Modeling.