Zillow's Home Value Prediction: Midway Report

Harish Pullagurla
Department of ECE
North Carolina State University
hpullag@ncsu.edu

Pooja Mehta
Department of ECE
North Carolina State University
pmehta@ncsu.edu

Ratika Kapoor
Department of Mathematics
North Carolina State University
rkapoor3@ncsu.edu

1 Background and Introduction

1.1 Problem Description

A house is the largest and perhaps the most expensive purchase a person makes in her or her lifetime. The goal of this project is to provide an estimated prediction of the value of a house for future sale.

The accurate prediction of a house price is key to prospective homeowners, developers, investors and other real estate participants such as mortgage lenders and insurers. Local government use market values to set real estate taxes.

A property's appraised value also plays an important role in several real estate transactions such as sales, loans and its marketability. Conventional method for determining property prices involved professional appraisers. This method had the disadvantage of the appraiser having a vested interest from the seller, buyer, mortgage broker or lender leading to a biased estimate of the house price. This led to the creation of automated house prediction systems, independent of the bias and providing a more accurate and less biased estimate of the house price.

Zestimate is one such house price prediction model. It was created by Zillow to give consumers as much information as possible about homes and the housing market.It is computed using a proprietary formula and is a starting point in determining a home's value. It is based on public and user-submitted data and also takes into account special features, location, and market conditions.

To further improve the accuracy of prediction, Zillow introduced an open competition on Kaggle (1). In this competition, Zillow provided full list of real estate properties in three counties (Los Angeles, Orange and Ventura, California) from 2016 and 2017. The goal of this competition was to minimize the log error between their Zestimate and the actual sale price. Error modeling is considered to be a powerful method to find areas to improve in a good existing model like Zestimate.

$$logerror = log(Zestimate) - log(SalePrice)$$

For this competition, Zillow provided all the properties with their home features for 2016 and 2017 from 3 counties. Each home feature had a 58 dimensional vector talking about features like year built, number of rooms, location, tax paid etc. Data set was provided with a list of 170,000 values, each having the log error and the date on which the transaction took place. Log errors corresponding to 6 time points for each of the property are to be predicted at the end.

Steps towards building a regression model to improve Zestimate residual error are provided in the following subsections.

1.2 Literature Survey

With a large amount of unstructured resources available, the Real Estate industry has become a highly competitive business. The data mining process in such an industry provides an advantage to the developers by processing this data, forecasting future selling prices and thus, helping people make knowledge-driven decisions.

Submitted as a Project Report for CSC522 Automated Learning and Data Analysis.

In the area of house price prediction, many attempts were made using different methods.Park et.al (2) conducted an analysis of 5359 townhouses in FairFax County, VA and used machine learning to develop house prediction models. They concluded that RIPPER outperformed other house prediction models such as C4.5, Bayesian and AdaBoost in all the tests. In their conclusion, they underlined the fact that housing markets can be influenced by macro-economic variables and future research should consider these variables along with environmental amenities variables as input to house prediction models.

Wang et.al (3) performed real estate forecasting based on SVM optimized by PSO. Support Vector Machine (SVM) is a type of learning machine that has helped in solving the problems of limited sample learning, nonlinear regression and in overcoming the "curse of dimensionality". To determine the parameters of SVM, which determine its learning and generalization; particle swarm optimization (PSO) was chosen. Their experimental results showed that PSO-SVM had a high forecasting accuracy compared to other models.

Extreme Gradient Boosting is one of the most extensively used algorithms in the Kaggle competition. Proposed by Tianqi Chen and Carlos Guestrin, it is a sparsity-aware algorithm for sparse data and weighted quantile sketch for approximate tree learning[7]. It uses far fewer resources than the existing algorithms, scaling billions of examples.

2 Method

2.1 Lasso Regression

Simple Linear regression is a method of fitting an optimal line/ plane in a feature space by reducing the least squares(OLS) error of the prediction with respect to the actual value. It is a method of modeling the relationship (using β) between the dependent variable y (here log error) in terms of explanatory independent variables x (like location , home description).

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i \tag{1}$$

Lasso (6) Regression is an extension of simple linear regression. When there are many $\beta's$ to be estimated the variance of the estimated coefficients gets large. To curtail the problem Lasso includes a penalty term that constrains the size of the estimated coefficients similar to ridge regression. That is $\sum^N |\beta| < t$, where t is the pre specified hyper parameter that determines the amount of regularization

$$lasso =_\beta ||Y - \beta||_2^2 + \lambda||\beta||_1^2 \tag{2}$$

Lasso is a shrinkage estimator which outputs coefficients, to generate a best linear fit for the given feature space, with a constrain on magnitude of coefficients that are biased to be small. In this, as the penalty term increases, lasso sets more coefficients to zero. This process would help to identify which features to keep and which to ignore based on β values being 0's, sparsifying the β vector. Thus the lasso estimator is a smaller model, with fewer predictors, having inherent dimensionality reduction nature.
This model would be a good method to this experiment because of two reasons. First, the data is of very high dimension and second, the relative importance of features is not known in prior, which the algorithm automatically handles.

2.2 Random Forest Algorithm

Random forest is a method which uses ensemble of multiple decision trees with each treee designed using algorithms like information gain or gini index at each node. Here each of the tree is generated based on samples and attributes that are randomly sampled from input data set. Thus each tree is build on a different set of attributes. The regression output is the average of outputs from each tree grown.
Random Forests are a preferred method where we have a large data-set with high dimensionality. Also, this technique is relevant here as they are know to handle missing values without decrease in the accuracy. They have a slight disadvantage in regression tasks as they handle it by quantizing outputs, whose smoothness depends on the complexity of the tree.

2.3 Extreme Gradient Boosting

XGBoost is a tree ensembles supervised learning method. It is a set of classification and regression trees(CART). It classifies the members of a family into different leaves, and assign them the score on the corresponding leaf. A CART is a bit different from decision trees, where the leaf only contains decision values. In CART, a real score is associated with each of the leaves, which gives richer interpretations that go beyond classification. This also makes the unified optimization step easier.

Algorithm:

1. Add a new tree in each iteration
2. Beginning of each iteration, calculate $g_i = \partial_{\overline{y}_i^{t-1} l(y_i, \overline{y}_i^{t-1})}$, $h_i = \partial^2_{\overline{y}_i^{t-1}} l(y_i, \overline{y}_i^{t-1})$
3. Use the statistics to greedily grow a tree $f_t(x)$
$$obj = -0.5(\sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T)$$
4. Add to the model, $\overline{y}_i^{(t)} = \overline{y}_i^{(t-1)} + f_t(x_i)$, where ϵ, is called step-size or shrinkage, usually set around 0.1
   This means we do not do full optimization in each step and reserve chance for future rounds, it helps prevent over-fitting.

XGBoost works faster than random forest as it reaches optimum solution in much lesser number of trees. It has better accuracy in prediction than most of the regression techniques because it reduces bias and variance in data by combining outputs. Here the most important that was tuned was eta, which is step size shrinkage used in update to prevents overfitting. After each boosting step, one can directly get the weights of new features. and eta actually shrinks the feature weights to make the boosting process more conservative.

3 Plan and Experiment

Figure [1] shows the flow of the experiments.

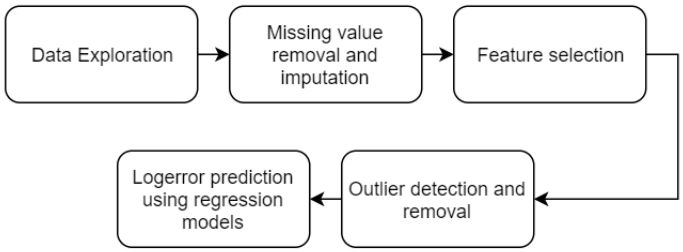

Figure 1: Block Diagram

3.1 Data Set Description and Exploration

Data set provided has 3 million house properties information. From these, 170,000 samples, each 58 dimensional describing the house attributes were provided with the log error which is to estimated. Figure : 2 shows the density vs. feature value plot for few features to understand if the training set follows a similar trend as the complete properties data provided. As similar trend was seen to be followed by both, deductions made on training data would be valid on other data as well. A Train - Test split of 70:30 was done on the data, giving 120,000 and 50,000 values respectively to work on.

Figure:3 A, displays the trends in log error values on the training data set. It was seen to follow a normal distribution centered at zero, implying that Zestimate was able to accurately predict most of the values. Figure:3 B, shows the mapping of one of the attributes with log error. It can be noted that there is no clear linear relationship between them.

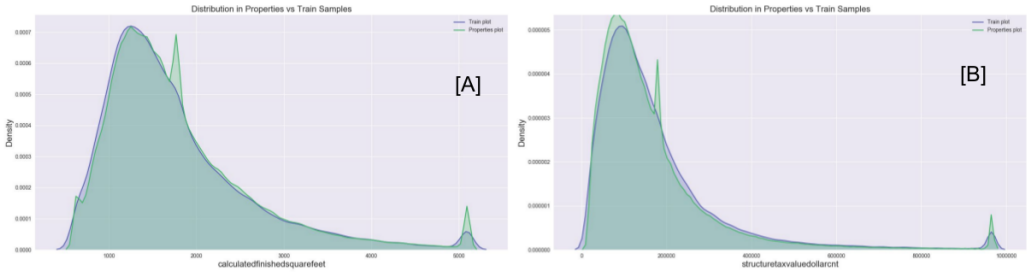

Figure 2: Comparison of Distribution between Training and Properties data

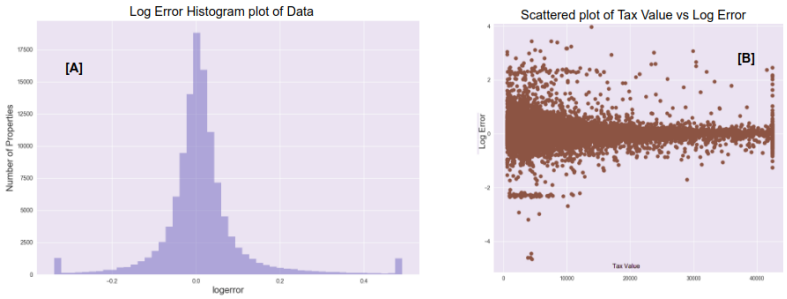

Figure 3: A. Histogram plot of log error values from training data , B. Log error vs Tax amount plot

The 58 dimensions of dataset could be split into 3 major categories, based on the information they are providing. 38 dimensions described about the physical properties of the house. Features like area of plot, number of rooms, type of A/C correspond to this category. 13 dimensions described about the location of the house, example, latitude, longitude, region zip code. 7 features talked about the tax paid in different categories for this property.

3.2 Missing Value Imputation

A large proportions of the 58 dimensional vectors were missing. Figure : 4 shows the histogram plot of percentages of missing values for each of the features. To impute extreme missing values would indirectly mean creating artificial data, hence imputation for these extreme cases was not considered to be appropriate. Thus, properties with missing values greater than 90 percent were dropped from further processing. Out of the initial 58 features provided, 20 of them had extreme missing values in this data, hence leaving 38 features to work with.

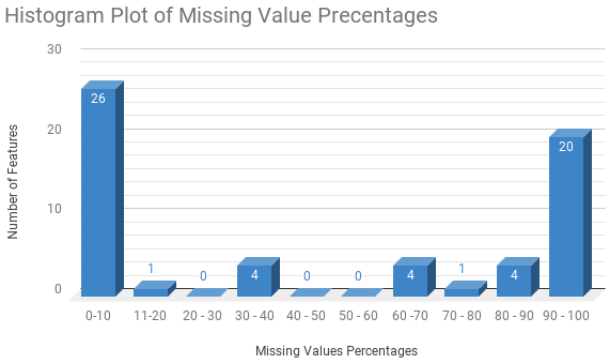

Figure 4: Histogram Plot of Missing value Percentages

Few features were dropped based on duplication or conveying the same information. For example, features such as 'FIPS' and 'region zip' captured the same information and hence one was kept while the other removed. 3 other such duplicate entries were removed bringing the feature set size to 35. In addition, features with near zero variance like 'assessment year' were also removed as they wouldn't add any information to the machine learning models.

The 34 features available, were understood in detail to come up with methods to impute. These features could be broadly divided into multiple categories based on the information being conveyed. Features giving number count describing the house like number of bedrooms, bathrooms etc constituted one third of the feature set. This category of features were imputed based on mode representing the most frequently spotted count, with exceptions for a few.Other major categories included area of different portions of the house and tax amount payed in different fractions. These were imputed with the mean values computed ignoring the outliers.

For a few attributes data was understood from other values and imputed based on those types. For examples, 'number of floors' was imputed with 'unit count' which described something similar.
A new method of imputing values based on physical location in which they were present was experimented. Nearest Neighbours based classification was used on latitude and longitude to get the house closest to the missing value house and data was imputed with it, with an assumption that properties closest to each other are mostly similar.

3.3 Feature Selection

Random Forests is one of the most common methods used to find the importance of different features. The tree-based strategies used by random forests naturally ranks by how well they improve the purity of the node. This means a decrease in impurity over all trees (called gini impurity). Nodes with the greatest decrease in impurity happen at the start of the trees, while notes with the least decrease in impurity occur at the end of trees. Thus, by pruning trees below a particular node, a subset of the most important features can be created. Figure:5 shows the relative importance of each of the features provided, which acted as a initial queue for selection of parameters for the machine learning model towards final log error prediction. 24 features were available after dropping features that were not important.

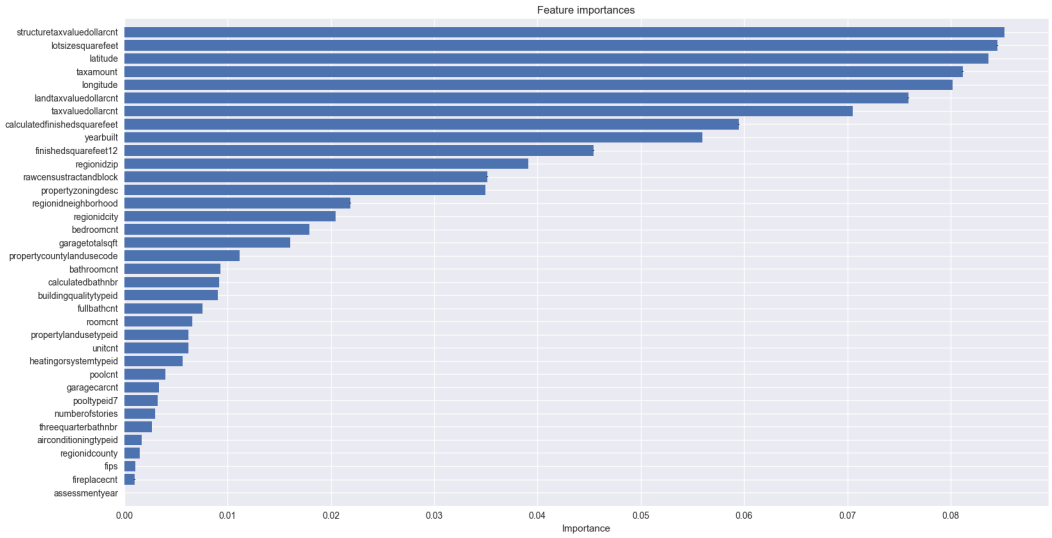

Figure 5: Feature Importance Scores using Random Forests

3.4 New Features

Feature engineering played an important role in data pre-processing. All the given features were analyzed and understood to come up with new features that could serve to be an important input for the models for predicting the house value. For example, the age of a house is considered to be

an important aspect when valuing its price. The age of each house was derived from the property describing the year it was built in. Similarly, the actual living area of a house relative to the total lot size also plays a key role in determining the worth of a house. This could be calculated as a ratio of the actual finished square feet area and the total lot size of the house. Both, the actual finished square feet area and the total lot size were properties given in the properties data set.
The aforementioned features, along with a few other features were created as part of feature engineering to increase the accuracy of our model's prediction. Adding 4 features, 28 features were left from the initial 58 features, for the machine learning model to model the data on.

3.5 Outlier detection and removal

Many of the features contained some extreme values which were clearly noise in the data and would have affected the prediction of logerror. Hence it became necessary to remove all such outliers before performing regression on the dataset. Boxplot was used to visualize how the data was spread and also the outliers. As can be seen from Figure (), the black dots are the outliers and there are quite a few of them for many features. 10th and 90th percentile for each feature was calculated and all values beyond these percentiles were removed, which improved the accuracy of the prediction.

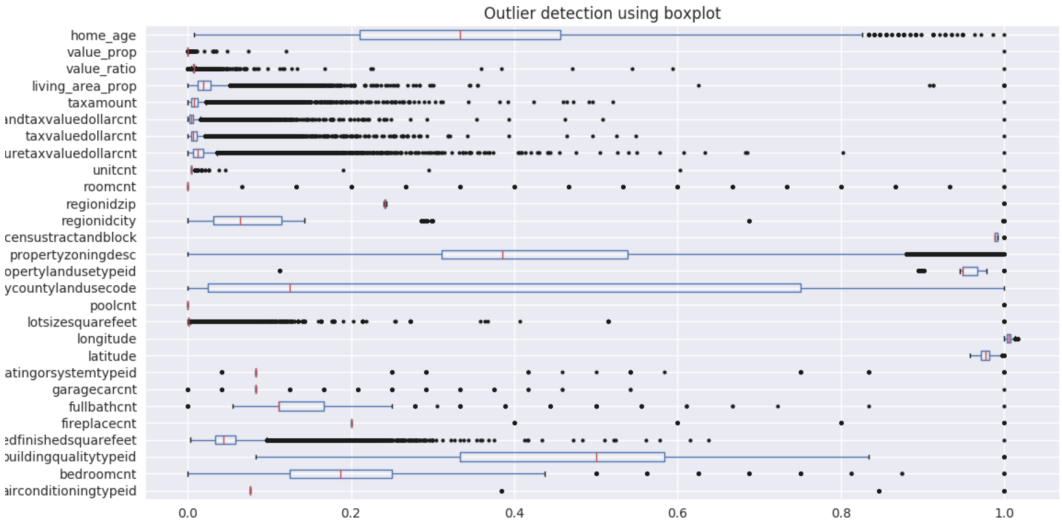

Figure 6: Visualization of Outliers

3.6 Machine learning regression models

On the cleaned and imputed data, three machine learning models were trained and tested, namely, lasso regression, random forest and XGBoost. All the algorithms were tuned to reduce the mean absolute error of the final predicted logerror values.

A 5 fold cross validation method was used to tune for various values of λ, the weight factor for L1 norm of coefficients, in the range or 0.0001 to 0.5, for lasso regression to minimize the mean absolute error. A λ value of 0.01 was seen to produce the minimum MAE.

Main tuning parameters for random forest were number of trees, maximum depth in each of them and threshold for making a new split.

XGBoost was mainly tuned for different values of eta and depth of tree. Training data was split into Train and Validation sets in 80 : 20 ratio. eta = 0.0475 with depth = 4 gave the best R-squared and MAE scores.

4 Results

The R-squared and the Mean Absolute Error (MAE) were chosen as the performance metrics for this regression task.

$$R - squared = \frac{SSR}{SST} = \frac{\Sigma(\hat{y} - y_i)^2}{\Sigma(y_i - \overline{y})^2}$$

$$MeanAbsoluteError = \frac{1}{n}\sum_{i=1}^{n} |y_j - \hat{y_j}|$$

R-squared, also known as the coefficient of determination is a statistical measure that determines how close the data is to the fitted regression line. It is calculated as a fraction of the regression sum of squares (SSR) and the total sum of squares (SST). Regression sum of squares quantifies how far the estimated sloped line is from the sample mean. Total sum of squares (SST) quantifies how much data points vary around their mean. Since R-squared is a proportion, it is always a number between 0 and 1. In general, the higher the R-squared, the better the predictability of the model.

Mean Absolute Error (MAE) was also considered to be important in this project for forecasting accuracy. MAE is the average of all absolute error, ie. the difference between the measured value and the true value.

The table below provides the results of R-squared and MAE based on the three models: Lasso Regression, Random Forest Regression and XGBoost used for predicting values.

Machine Learning Model	R-squared	Mean Absolute Error
Lasso Regression	1.00001917	0.074863
Random Forests	1.0000163	0.0707384
XGBoost	1.00	0.0706256

It was inferred from the results that XGBoost outperformed the other two models in minimizing the log error. This model provided an R-squared of 1 and the minimum Mean Absolute Error. The MAE obtained was comparable to the leader board values from the competition page, where the best values were around 0.068

An R-squared of 1 indicates near-perfect prediction. In other words, all data points fall on the regression line and the predictor variables account for all the variation in the variable to be predicted.

5 Conclusion

Considering the large amount of data provided by Zillow for this project, it was concluded that data exploration and pre-processing was an important step towards exploring any machine learning algorithm for house price prediction.

Imputation of missing values with the right technique also plays an important role as too much of missing data can affect the performance of any model. It was also observed that feature selection for understanding their relative importance and contribution for predicting a house price will be key to any model's forecast.

6 Project Learning's and Future Work

The log error value also depends on the date during which the transaction takes place. Temporal variation of the predicted value is to be considered for further improvement in accuracy. New methods for data imputation could be experimented, by deriving values from other features instead of using mean or mode to impute the values. Creation of new features is also the be experimented further.

It is also understood from readings that cascading of multiple machine learning models each doing some part of the prediction tasks helps in boosting the final output. Such methods could be experimented in future.

The project acted as a great experience giving an introduction to the exiting world of data science. It helped us understand real world data and how it could be used to predict useful values. It also made us realize that machine learning is not just about the final algorithm but a complete pipeline. The tasks of handling imputation of missing values and importance of parameter tunning was a major learning from the project.

7 GitHub Repository

The code for the project was developed using python 2.7, using several open-source statistical learning toolboxes like scikit learn, pandas and other basic toolboxes like numpy, matplotlib were also use, The code is free to use for any further development.

The code could be accessed at https://github.com/pharish93/kaggle_zillow .

8 References

[1] Kaggle.com. (2017). Zillow Prize: Zillow's Home Value Prediction (Zestimate) [online] Available at: https://www.kaggle.com/c/zillow-prize-1 [Accessed 07 Nov 2017].

[2] Park.B & Jae.K, Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data,(2016) *Expert Systems with Applications*

[3] Wang.X, Wen.J, Zhang.Y & Wang.Y, Real estate price forecasting based on SVM optimized by PSO, *Optik*

[4] Gan.V, Agarwal.B & Kim.B, Data Mining Analysis and Prediction of Real Estate Prices, *Issues in Information Systems*

[5] Jirong.G, Mingcang.Z & Liuguangyan.J, Housing price forecasting based on genetic algorithm and support vector machine, *Expert Systems with Applications*

[6] Tibshirani, R. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, Series B, Vol 58, No. 1, pp. 267–288, 1996.

[7] Tianqi Chen, Carlos Guestrin, XGBoost: A Scalable Tree Boosting System, Proceeding KDD '16 Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining Pages 785-794