Homework 6

Aruna Kumaraswamy[1]*

Abstract
House Price features and related pricing is published in Kaggle for prediction competition. In this paper, we will discuss the steps leading towards to the selection of the best model to predict the sale price of a house based on the features. We will discuss the data, influential features, feature engineering and model tuning. Linear regression, regression tree, gradient boosting machine, neural network and random forest algorithms are used to analyze a better prediction model. Best performing model is chosen based on the validation. Using the best model and the test data we will predict the sale price. Results are submitted to Kaggle for ranking.

Keywords
House Price - Prediction - Mining - Regression

[1] *Computer Science, School of Informatics and Computing, Indiana University, Bloomington, IN, USA*

Contents

1. Problem and Data Description

House price data set from Kaggle has 1460 entries in training and 1459 entries for test. There are 80 features related to house, its location and surrounding facilities that can impact the sale price. Our objective is to engineer the features to better derive its impact on the sale price and also choose an algorithm that can help predict the sale price of a house with less error.

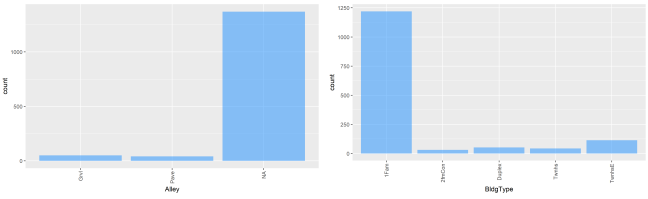
Following are discrete variables (43): ”MSZoning” ”Street” ”Alley” ”LotShape” ”LandContour” , ”Utilities” ”LotConfig” ”LandSlope” ”Neighborhood” ”Condition1” ”Condition2” ”BldgType” ”HouseStyle” ”RoofStyle” ”RoofMatl” ”Exterior1st” ”Exterior2nd” ”MasVnrType” ”ExterQual” ”ExterCond” ”Foundation” ”BsmtQual” ”BsmtCond” ”BsmtExposure” ”BsmtFinType1” ”BsmtFinType2” ”Heating” ”HeatingQC” ”CentralAir” ”Electrical” ”KitchenQual” ”Functional” ”FireplaceQu” ”GarageT ”GarageFinish” ”GarageQual” ”GarageCond” ”PavedDrive” ”PoolQC” ”Fence” ”MiscFeature” ”SaleType” ”SaleCondition”

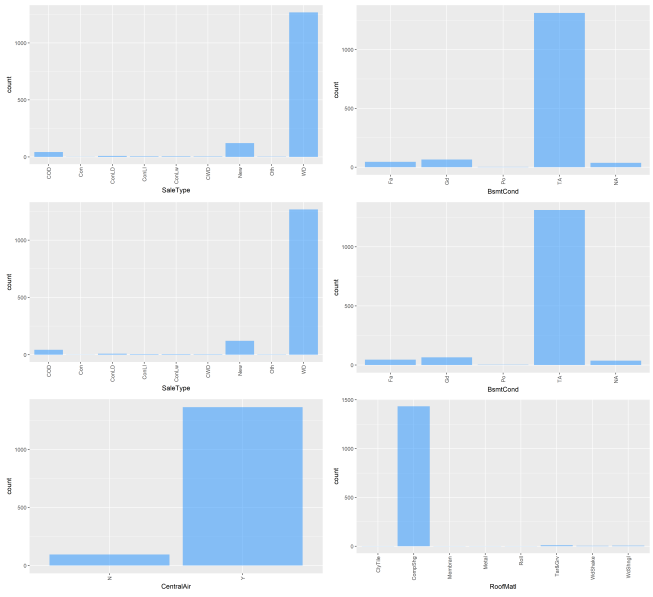
Following are continuous variables (38): ”Id” ”MSSubClass” ”LotFrontage” ”LotArea” ”OverallQual” ”OverallCond” ”YearBuilt” ”YearRemodAdd” ”MasVnrArea” ”BsmtFinSF1” ”BsmtFinSF2” ”BsmtUnfSF” ”TotalBsmtSF” ”X1stFlrSF” ”X2ndFlrSF” ”LowQualFinSF” ”GrLivArea” ”BsmtFullBath” ”BsmtHalfBath” ”FullBath” ”HalfBath” ”BedroomAbvGr” ”KitchenAbvGr” ”TotRmsAbvGrd” ”Fireplaces” ”GarageYrBlt” ”GarageCars” ”GarageArea” ”WoodDeckSF” ”OpenPorchSF” ”EnclosedPorch” ”X3SsnPorch” ”ScreenPorch” ”PoolArea” ”MiscVal” ”MoSold” ”YrSold” ”SalePrice”

Several attributes have missing values. In training data, missing value information is as follows: LotFrontage(259), Alley (1369),MasVnrArea(8), BsmtQual(37), BsmtCond(37), BsmtExposure(38), Electrical(1), BsmtFinType1(37), BsmtFinType2(38), FireplaceQu(690), GarageType(81), GarageYrBlt(81), GarageFinish(81), GarageQual(81), GarageCond(81), PoolQC(1453),Fence(1179), MiscFeature(1406)

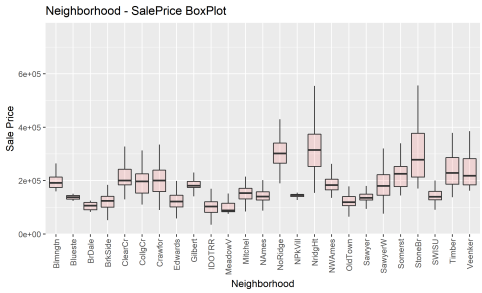
There are 6965 missing values in training data. In test data, we have 7000 missing values. When tested with manyNAs function for entries with NAs exceeding 20 percent threshold, no rows are returned in training and test. We have records in training and test data with 10 percent columns missing values. For this project, we will use the default 20 percent and hence will not remove any entries from both training and test data.

Following diagrams are the distribution of some discrete attributes that exhibit high bias. We do not observe a normal distribution of values across the 43 categorical variables.

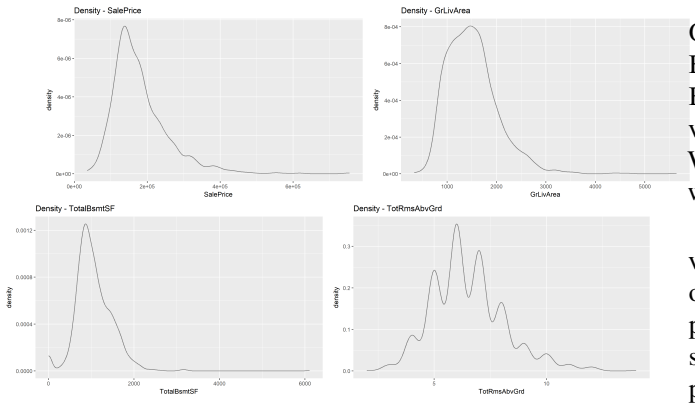


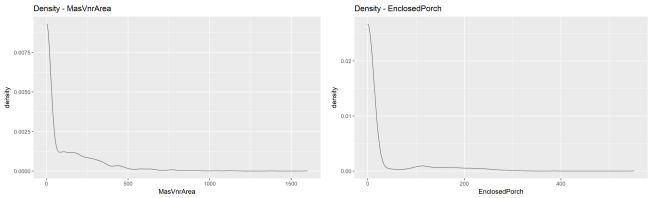


Following conditional boxplot shows the distribution of Neighborhood and sale price. We observe that neighborhood does not have a normal distribution. Within the neighborhood we see that NridgeHt and StoneBr have listed higher sale prices. IDOTRR has listed lower sale price. Distribution of price however is predominantly within 350000, prices above this value are either outliers or under represented in the data.



Following diagram are the distribution of some continuous variable. We have some variables such as SalePrice, GrLivArea, TotalBsmtSF that exhibits normal distribution for most part, but have outliers. Most continuous variable do not have normal distribution of values. They have either high bias or several outliers.





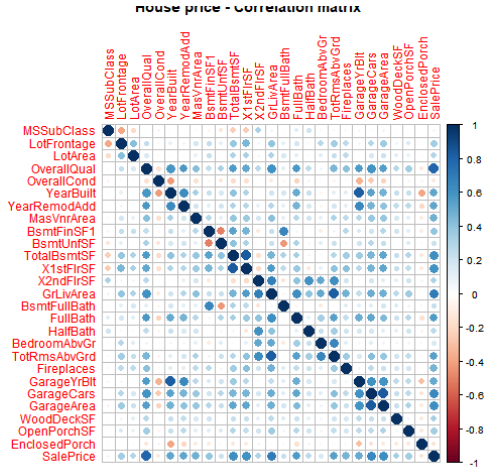
2. Data Preprocessing & Exploratory Data Analysis

2.1 Handling Missing Values

Both training and test data has missing values. I chose central imputation to fill the missing values. Attributes such as Alley, Fence, MiscFeature, PoolQC that have a much higher NA values, switched to values that have a higher frequency. This will lead to a high bias. I removed them to study the impact to the model, no significant change was observed. So I have left them in the data.

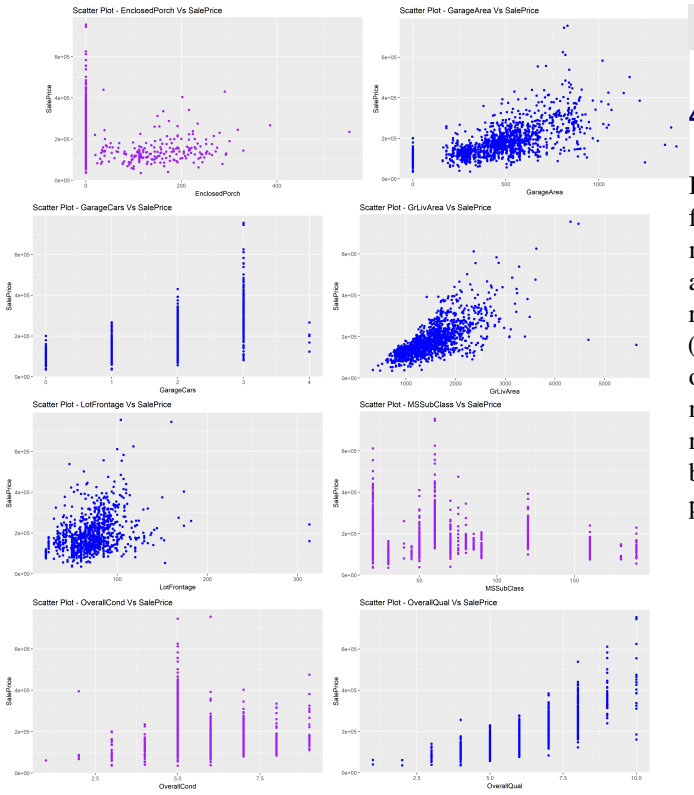
2.2 Exploratory Data Analysis

Following diagram shows the correlation of attributes to the sale price. We observe that the following attributes contribute to a higher correlation with sale price.



Based on the diagram, we observe that LotFrontage, LotArea, OverallQual, YearBuilt ,YearRemodAdd, MasVnrArea, BsmtFinSF1, BsmtUnfSF, TotalBsmtSF , X1stFlrSF, X2ndFlrSF, GrLivArea BsmtFullBath, FullBath, HalfBath, BedroomAbvGr, TotRmsAbvGrd, Fireplaces, GarageYrBlt, GarageCars, GarageArea, WoodDeckSF and OpenPorchSF have a higher correlation with sale price.

In the following scatter plots, we also see that there are values which are not linearly correlation. For example, we observe that for enclosed porch, when the value is 0, the sale price does increase. This indicates, that the linear regression model is multi-variate and does not have a pre-dominant predictor variable.



As part of feature engineering, I have categorized the GrLivArea and TotalBsmtSF, YearBuilt. If year of re-modeling is more recent than the year built, I use it as year built. Exterior 1st and 2nd had no impact to the model, so these features are removed from training and test.

Also, I tried 'One Hot Encoding'. This expanded the categorical variable into a separate feature, resulting in 288 variables. However, I did not see any improvement in model performance as a result of this. So I decided to not use one hot encoding.

3. Algorithm and Methodology

After feature engineering, the cleaned data is split into training and test data sets again. We have 78 variables including the dependent variable. We use the cleaned training data to train the model to learn predicting the sale price. Feature engineering and model tuning is a iterative process.

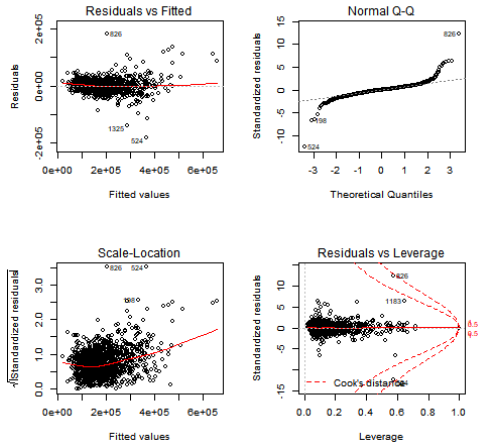
We split the training data into half for validation (730 entries). For each model, we train with the dependent variable. Use the validation data on the trained model to observe the error. When the model results are satisfactory, I used the test data (1459 entries) to predict results and save as CSV for Kaggle submission.

Algorithms chosen for this experiment are Linear regression, Regression tree, Random Forest, Gradient Boosting Machine and Neural Network.

4. Experiments and Results

4.1 Linear Regression

Based on the correlation plot, we observe that there are a few attributes that are good predictor variables. So I tried 2 models, one with the selected predictors from cor function and other with all the features in the training data. Linear regression model with all features has good R-squared values (91.9) compared to choosing specific predictors (80.6) based on the corrplot. So this indicates that the variables that do not have a high correlation are quite useful, in creating the necessary variance in the prediction process. The p-value for both models are very low, supporting the hypothesis that the predictor variables have significance.

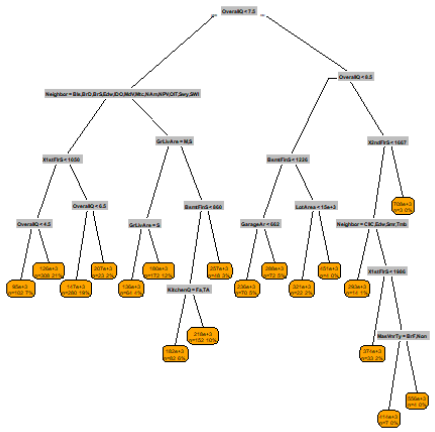


Residual Vs Fitted value shows that we have both positive and negative residual value. As most sale prices are within, 350K, we can see a pattern affecting the homogeneity of our sample. This indicates that our model will have a bias towards predicting in this range and treats anything above the range as outliers. QQPlot also describes the same behavior, we have outliers in either end of the tail and in between the sample has normal distribution.

Mean absolute error 12945.50, Root Mean square error 20436.62 and Root Mean square log error 0.1068

4.2 Regression Tree

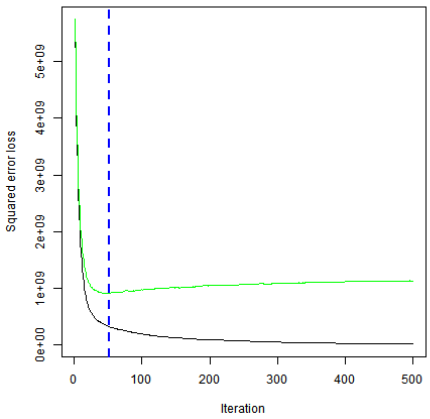
Based on the following tree, we can see that this algorithm uses - OverallQual, Neighborhood, GrLivArea, X2ndFlrSF, BsmtFinSF1, MasVnrArea and GarageYrBlt



Mean absolute error 29584.26, Root Mean square error 44614.86 and Root Mean square log error 0.2275

4.3 Gradient Boosting Machine

GBM is an ensemble of models that combines the base learner, such as a regression tree. The combined model should perform better than the individual base learner. Gradient boosting is a boosting algorithm based on the idea of steepest descent minimization. In this iterative model, a weak learner is added in each step to minimize the error of current ensemble. In this experiment, I started with a training of 500 trees, depth of 15 and a 10 fold cross validation. Best number of trees recommended by gbm based on the loss function is 51. So this is used to predict the test.



Mean absolute error 17347.51, Root Mean square error 28881.99, Root Mean square log error 0.1392

4.4 Random Forest

It is a ensemble of regression tree models. Unlike GBM, it is not a weak learner, it grows full trees in parallel and it reduces the error by decreasing the variance. GBM, on the other hand, decreases the error by decreasing bias. in this experiment, we grow 500 full decision trees to train the model. This model resulted in the lowest error of all models in comparison. As it functions by decreasing the variance, it appeared more stable during the feature engineering exercise where I added and removed many features to study the impact.

Mean absolute error 6514.58, Root Mean square error 11347.432, Root Mean square log error 0.0597

4.5 Neural Network

In this experiment, we predict using regression function on Gaussian distribution of sale price. In NN, there is 1 input layer, 3 hidden layers and 1 output layer. We use H2O default activation function (rectifier). I have specified 10 epochs and a 5 fold cross validation. Neural network issued warnings that it notices values for SaleType and RoofStyle that were not observed during training.

RMSE: 30479.13, MAE: 19393.9 and RMSLE: 0.1538

5. Summary and Conclusions

Regression tree did not perform well in this experiment. I had the highest error. RandomForest yields a stable and better performing model in predicting the sale price of the house. Linear regression is a basic model, but resulted in a much better performance compared to GBM and neural network. As GBM operates by reducing bias, we have to focus in feature engineering to remove features that have high bias and see if this can yield a better performance.

Random Forest is the best model based on this experiment. I would recommend additional feature engineering efforts to fine tune this model.

I submitted, random forest (0.1485), linear regression, GBM (0.15) and Neural network results to Kaggle. Neural network resulted in the best RMSLE (0.14061) compared to other submissions.

Acknowledgments

https://www.kaggle.com/notaapple/detailed-exploratory-data-analysis-using-r/notebook
https://www.kaggle.com/skirmer/fun-with-real-estate-data/notebook
Data Mining with R - Luis Torgo