MEF UNIVERSITY

FORECASTING HOUSE PRICES BY REGRESSION AND CLASSIFICATION MODELS

BDA 551 – Model Building and Validation Final Project

Bengisu Öniz

Cihan Tektunalı

Duygu Can

Feray Ece Topçu

İsmetcan Hergünşen

İSTANBUL, 2018

ABSTRACT

FORECASTING HOUSE PRICES BY
REGRESSION AND CLASSIFICATON MODELS

Bengisu Öniz

Cihan Tektunalı

Duygu Can

Feray Ece Topçu

İsmetcan Hergünşen

AUGUST, 2018

Figuring out the best model is one of the main jobs that machine learning industry deals with, in predictive modeling. Data preprocessing steps, tuning the parameters and even the comparison of the performance scores of different models becomes a tremendous job, there is a wide spectrum of techniques that can be employed. In this study, we propose several regression and classification models to forecast house prices in the Ames Housing Dataset which is a popular data source. The major steps of the study involve data cleaning, feature selection, model building, parameter tuning, and model comparison based on evaluation score. As regressors, we choose to work with *Linear Regression* and *Extreme Gradient Boosting* algorithms on a subset of strong predictor variables. Alternatively, we also used three different algorithms to build classification models, such as *Logistic Regression*, *Decision Tree*s, and *Random Forest*. Another subset of numerical and categorical data is included in the classification models to make two-class predictions (below or over the median sale price). For classification models, we used F1-score as a performance metric and AUC for model comparison. For regression models, we used Mean Squared Error (MSE) and Adjusted R^2. The results show that including categorical variables together with numerical variables resulted with a better model. We observed that the classification models outperforms the others.

Keywords: Forecasting, predictive modeling, regression, classification, price forecasting

TABLE OF CONTENTS

1. INTRODUCTION

In some stage of life a majority of the people are cursed to be in a house hunt. Most of us dream to find a perfect place to live and to transform a concrete and iron mass into a sweet home. However, as our expectations from a home become bigger and bigger, finding a modest place to live becomes harder and harder. Making a pros and cons list, hunting houses within a budget can be a frustrating job. One should decide which features of the houses are really important and worth for paying the price. Thanks to Ames Housing dataset, every detail of a potential home can be analyzed and used for price estimates.

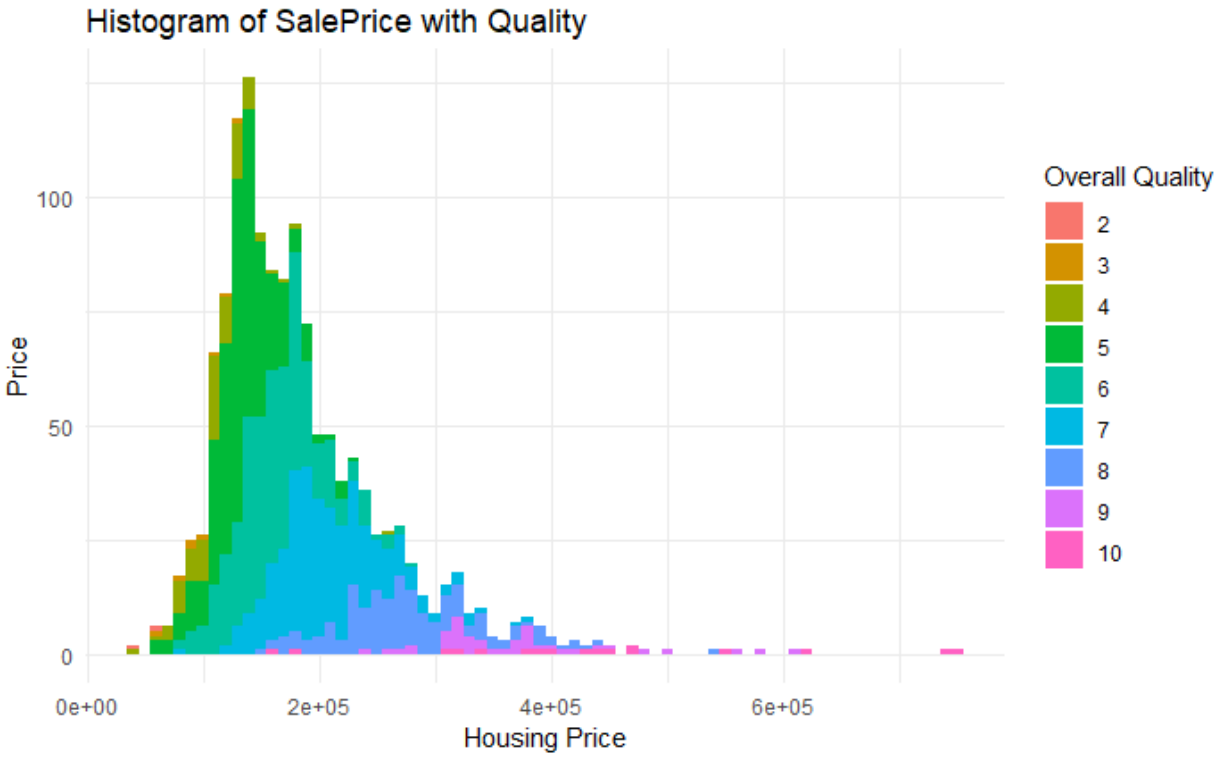


Figure 1: Change in price distribution with house quality.

Ames Housing dataset is collected by Dean De Cock and there is an ongoing Kaggle competition on it [1]. The price range of the houses in Ames, Iowa lies between $35311 and $75500 with a median value of $168500. As it can be seen from the histogram above the sale price is positively skewed. There are lots of features present in the dataset representing every

aspect of a residential house. These features affect the price behaviour. For example, as seen from Figure 1 the utmost quality houses can only bought at high prices.

The main aim of this research project is to make a valid estimate of the house prices by using as few features as possible, including only the necessary ones in the model. The problem of interest can be approached in both ways: either with a predictive regressor or reframing the research question into a classification problem. According to the precision and strength of the features selected in models, we believe that either of these methods is able to work well. We preferred to use only numerical variables rather than categorical ones in our regression models (incorporating Linear Regression and Extreme Gradient Boosting algorithms) to explore the prediction capacity of simpler models. Because the number of observations in the dataset was very small compared with the number of features, we picked the best numerical predictors and built the models by using them. Our aim was to find the model that minimizes MSE and maximizes Adjusted R^2.

Our classifier separates cheap and expensive houses from each other. As a threshold SalePrice median value is used, since it is not affected from the outliers. Prices above that value are considered as expensive and below that value as cheap. In our first classification model that uses Logistic Regression algorithm, we continued to use only numeric predictors selected by best subset selection method in order to make comparison with regression models. Afterwards, we decided to add one hot encoded categorical features to source data to get more valuable results from Logistic Regression. Output of this model was not better than the first logistic regression. Thus, we seek alternative classification models (Decision Tree and Random Forest) to reach a better outcome than previous models. The main performance metric we used for classification was F1 score. Additionally, confusion matrix, AUC and lastly accuracy were used as evaluation metrics for classification models. As we expected, high performing models were created by including both numerical and categorical predictors generate the most reliable results.

2. METHODOLOGY

Exploratory data analysis in this study started with observation of features, their types and distributions. To prepare data for modeling, further steps were applied such as data type conversion, exclusion of highly correlated features, transformation of categorical features and deskewing-normalization. After the initial data preprocessing stage, best subset selection method was used to acquire the strongest features in numerical data and narrow down the predictor set. At the modeling stage, regression and classification models were formed. We compared model performances by training set cross-validation errors and test set errors.

2.1. Exploratory Data Analysis - Data preprocessing

The Ames Housing dataset consists of 1460 instances with 81 features, including 38 numerical columns. At the first step of data cleaning, six columns with missing values more than 10% were completely removed since filling those with median or mean values might be misleading. When we observe the rest, count of the missing values does not exceed 5% of the size so they are dropped too. This brutal move shrinked our dataset only by 8.4%, which is tolerable at first sight. Thus, we proceeded with the 6 columns dropped and 122 rows deleted version of the original dataset. Later on in the evaluation step we could only noticed that we also lost one of the levels of one of our most dominant predictor, OverallQuall.

Three columns representing categorical features of buildings were also reformatted as factor variables. These are MSSubClass (type of dwelling), OverallQual (general material condition of building) and OverallCond (general condition of building). We generated a new feature by subtracting the build years from current year and replaced YearBuilt column. The correlations between numerical features were investigated (Figure 2). Not to train our models with collinear features and avoid model bias, we decided to detect collinear features and dropping the one which is less correlated with the target variable, SalePrice. According to our definition, the features which had an absolute correlation score more than 0.8 were accepted as collinear. According to our analysis such multicollinear features to be dropped were GarageArea,

TotalBsmtSF, GarageYrBlt and TotRmsAbvGrd. Furthermore, by looking at the distributions of PoolArea and MoSold columns, we decided that they are not informative. So we removed them.

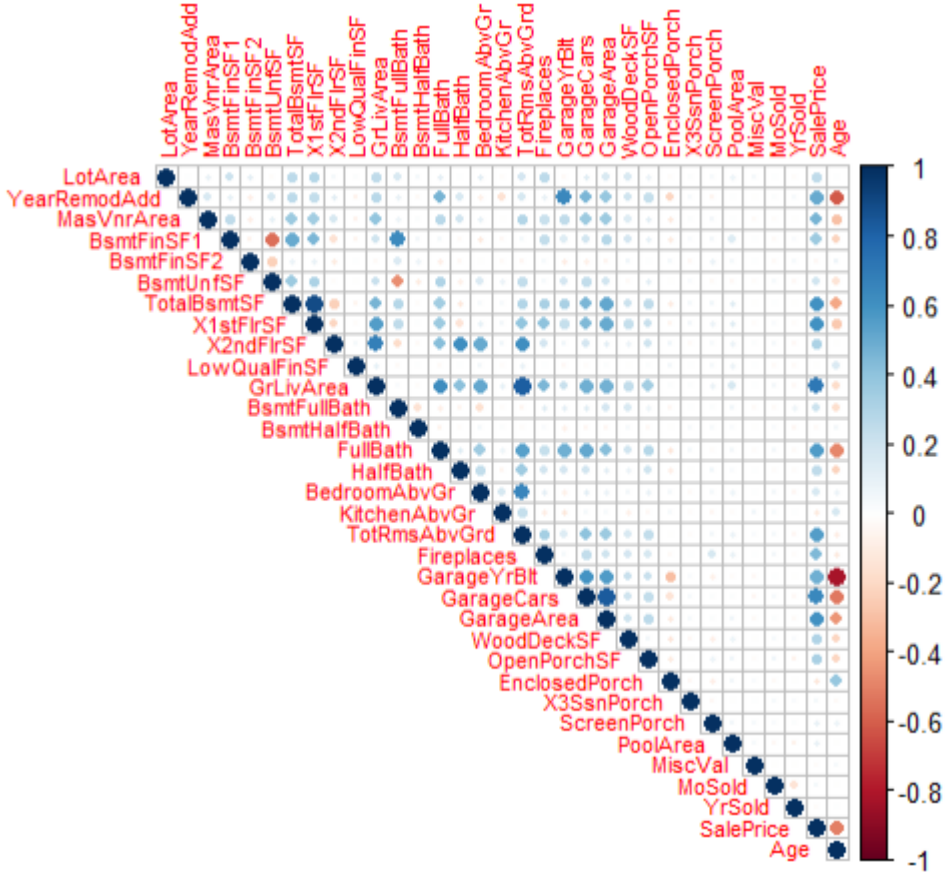
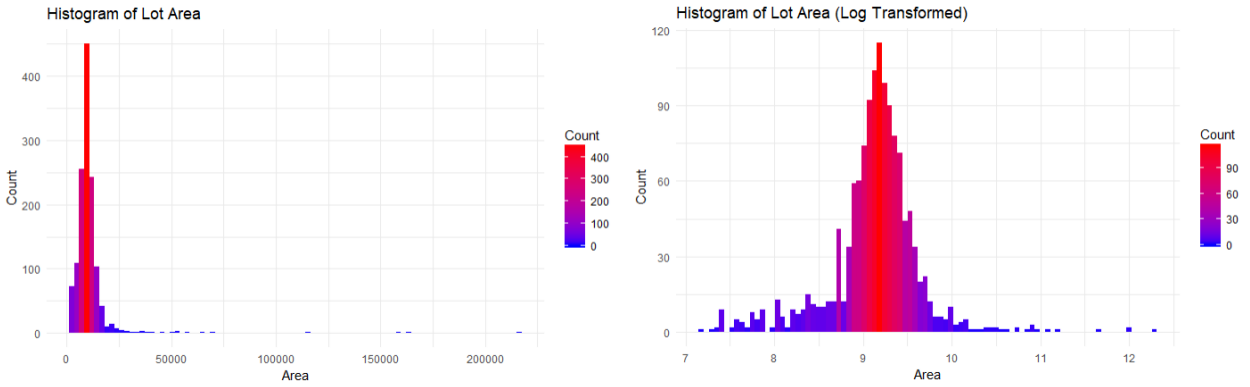


Figure 2: Correlation matrix of numerical features.

At data transformation stage, we focused on numerical features again, which have positively skewed distributions (skewness greater than 0.5) in general. To solve this problem, we applied logarithmic transformation. The divergence problem for zero values was handled by shifting each value by 1. Figure 3 shows the results of this transformation for LotArea column. Notice that the positively skewed distribution becomes to the normal distribution after the transformation.



(a) (b)

Figure 3: Lot Area distribution before (a) and after (b) log transformation.

We also standardized to all numerical columns by subtracting mean value from each observation and dividing by standard deviation. In this way all numerical columns had zero mean and unit variance.

2.2. Feature Selection

After normalizing the source data for regression models, *best subset selection* was executed to figure out the most important independent variables. As James, Witten, Hastie & Tibshirani (2013) mentioned, this method first fits a different least squares regression for every combination of given variables [2]. Then it selects the models having the least residual sum of squares (RSS) value. Final set of the strongest predictors is selected from the model with the lowest cross-validated prediction error or adjusted R^2.

All of 27 numerical features were given as input to algorithm. The ideal number of features that maximize adjusted R^2 value (0.831) is found to be 16 (Figure 4). Schneider, Hommel & Blettner (2010) point out that in linear regression analysis with multiple independent variables, the number of samples should be more than 20 times the number of features as a general rule [3]. So our data is large enough to train models with 16 variables. The residual sum of squares (RSS) value for 16 variables is calculated as 223.11. With the exclusion of one more variable (BsmtFinSF2) which we thought to have no use, the following 15 features are decided to

be used in regression models: LotArea, YearRemodAdd, BsmtFinSF1, X2ndFlrSF, LowQualFinSF, GrLivArea, BsmtFullBath, BedroomAbvGr, KitchenAbvGr, Fireplaces, GarageCars, WoodDeckSF, EnclosedPorch, ScreenPorch, Age.

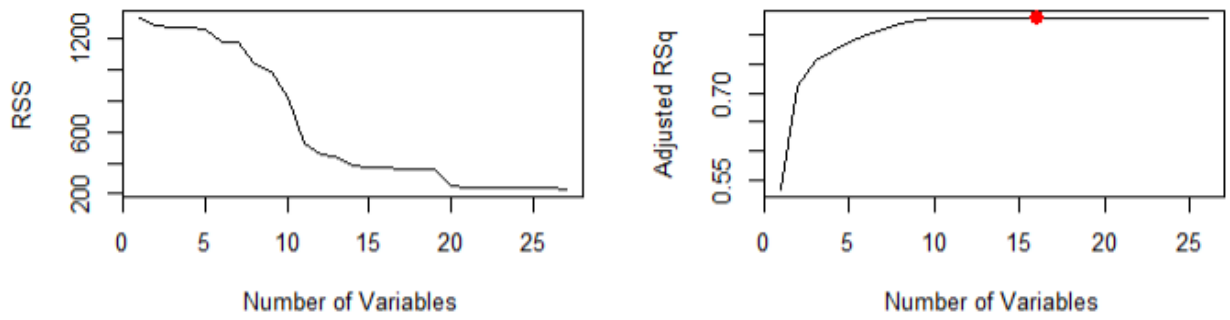


Figure 4: Best subset selection results from numerical features.

2.3. Linear Regression

We fitted our linear regression model by using 15 features selected. The adjusted 10-fold cross-validation error we got is ≈ 0.164. Test set Adjusted R^2 value is ≈ 0.846 while Mean Squared Error (MSE) is computed as ≈ 0.193. Table 1 displays the effects of independent variables and their significance values below. It is observed that 3 variables in this model do not have statistically significant importance (LowQualFinSF, WoodDeckSF, EnclosedPorch).

	Estimate	**Std. Error**	**t value**	**Pr(>|t|)**
(Intercept)	0.010468	0.013220	0.792	0.42865
LotArea	0.127560	0.015659	8.146	1.25e-15 ***
YearRemodAdd	0.162809	0.018837	8.643	< 2e-16 ***
BsmtFinSF1	0.093613	0.016780	5.579	3.21e-08 ***
X2ndFlrSF	-0.130132	0.018597	-6.997	5.09e-12 ***
LowQualFinSF	-0.023161	0.012293	-1.884	0.05988 .
GrLivArea	0.590122	0.026029	22.672	< 2e-16 ***
BsmtFullBath	0.035514	0.016884	2.103	0.03571 *
BedroomAbvGr	-0.042631	0.017375	-2.454	0.01433 *
KitchenAbvGr	-0.091298	0.014219	-6.421	2.19e-10 ***

Fireplaces	0.078179	0.016198	4.826	1.63e-06 ***
GarageCars	0.118318	0.019137	6.183	9.54e-10 ***
WoodDeckSF	0.020351	0.014403	1.413	0.15801
EnclosedPorch	0.006221	0.014898	0.418	0.67637
ScreenPorch	0.041992	0.013863	3.029	0.00252 **
Age	-0.241659	0.021880	-11.045	< 2e-16 ***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ‘ 1

Table 1: Linear Regression Coefficients.

2.4. Extreme Gradient Boosting

For Extreme Gradient Boosting regression, we converted training and test data into matrix format. Then 10-fold cross-validation is applied to training data. Root Mean Squared Error (RMSE) is used as evaluation metric. Training set Root Mean Squared Error value for best iteration is computed as 0.395 while mean test set Root Mean Squared Error is 0.4. After we fitted the model with parameter values from the best iteration results, Mean Squared Error gained from test set is ≈0.193. Both models yielded approximately the same test error. Therefore, we applied grid search to Extreme Gradient Boosting model for extra bit of performance. The feature importance graph acquired from this model is displayed in Figure 5.

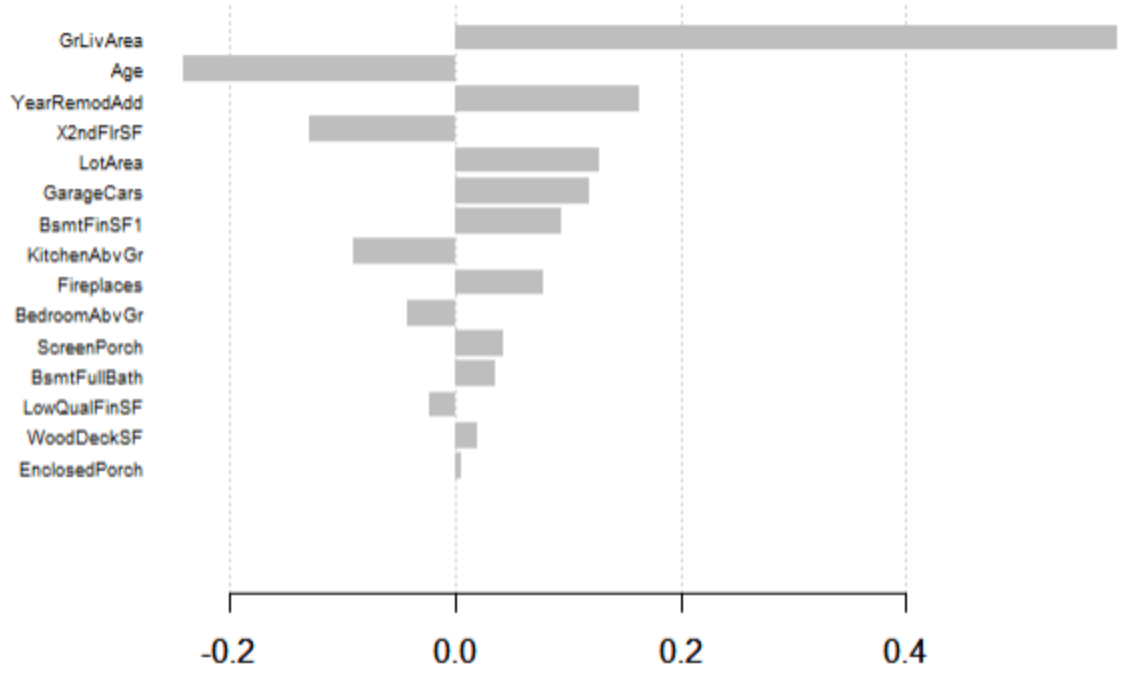


Figure 5: Feature importance graph from trained Extreme Gradient Boosting model.

As it is seen from feature importances, GrLivArea is by far the most influencing predictor, followed by Age and YearRemodAdd.

Extreme Gradient Boosting model is further optimized by grid search with the following *Parameters:*

Number of rounds: 67, 68, 69
Maximum Tree Depth: 2, 3, 4
Eta: 0.2
Gamma: 0
Col Sample By Tree: 0.9
Min Child Weight: 0.5, 0.6
Subsample: 0.8

Number of boosting rounds has been narrowed down from a broader range to 67 – 69 after several trials. Maximum tree depth values were selected smaller than default value of 6 to prevent overfitting, as stated in XGBoost documentation [4]. Weight shrinkage ratio (eta) has been decreased from the default value of 0.3 to 0.2. Default value (0) of minimum loss reduction

parameter (gamma) is used. Column subsample ratio was lowered to 0.9 as it yielded lower Root Mean Squared Error. The minimum child weight alternatives in each node were 0.5 and 0.6, which caused a less conservative algorithm compared with default value of 1. Subsample ratio was selected as 0.8 rather than default value of 1, which helps prevent overfitting. 10-fold repeated cross-validation is used in every boosting iteration. Search grid is visualized in Figure 6.

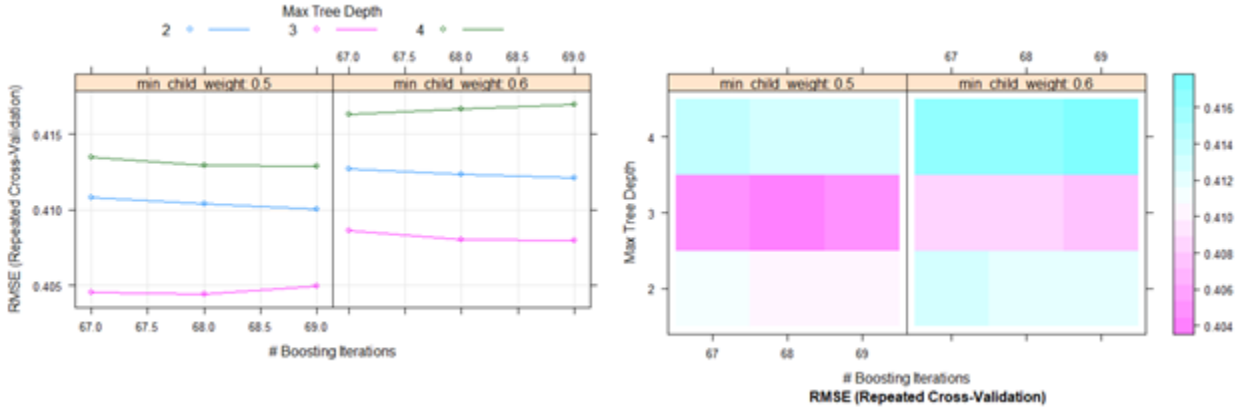


Figure 6: Root Mean Squared Errors in Extreme Gradient Boosting Model's Grid Search.

Consequently, the best model selected from grid has R^2 value of 0.844 and RMSE value of 0.404 which approximately equals to Mean Squared Error value of 0.163. Extreme Gradient Boosting model tuned with grid search showed an improvement over Linear Regression model.

2.5. Logistic Regression Classification

Predicting discrete dependent variable, classification, requires the categorical version of SalePrice. Therefore, continuous SalePrice column is converted into two categories. Prices over median value are tagged as 1 and prices lower than the median are labeled as 0. After this conversion, the target variable becomes categorical and classification algorithms can be used on prediction. Additionally, cleaned and transformed dataset is split as train and test dataset. Train dataset has 915 rows and 68 columns and test dataset has 423 rows and 68 columns.

As the first algorithm for discrete dependent variable, logistic regression is used with different input features.

2.5.1. Logistic Regression with Best Subset Selection

To begin with, logistic regression executed with numeric features which are offered by best subset selection. These columns can be ordered as LotArea, YearRemodAdd, BsmtFinSF1, X2ndFlrSF, LowQualFinSF, GrLivArea, BsmtFullBath, BedroomAbvGr, KitchenAbvGr, Fireplaces, GarageCars, WoodDeckSF, EnclosedPorch, ScreenPorch, Age. Logistic regression is fitted with train data and after fitting, SalePrice column is predicted with test data. Accuracy on test dataset is 0.884 for this model (Table 2). Additionally, AUC is 0.884 and F1 Score of the model is 0.88305.

	y_pred	
y_true	0	1
0	185	27
1	22	189

Table 2: Confusion Matrix of Logistic Regression

2.5.2. Logistic Regression after One Hot Encoding

Using categorical features in logistic regression requires one hot encoding method. Therefore, categorical features are converted into numerical version of them by applying one hot encoding method on cleaned data. One hot encoding method generates more than 200 new features due to categorical independent variables and number of numeric features become 289 to use in logistic regression. After data transformation; logistic regression fitted with all these numeric columns of train dataset.

Accuracy of test dataset of this model is 0.834 (Table 3). In addition, AUC is 0.8345 and F1 Score is 0.8317. Although there are more features, evaluation metrics of logistic regression decreases. The reason of this situation is that after one hot encoding, one of the category of categorical column should be dropped for each converted categorical variable. However, this step is skipped on one hot coding part and logistic regression after one hot encoding model become insignificant.

	y_pred	
y_true	0	1
0	173	39
1	31	180

Table 3: Confusion Matrix of Logistic Regression

2.6. Decision Tree Classification

2.6.1. Decision Tree before Pruning

Decision tree classifier is more appropriate algorithm for this dataset because there are both numerical and categorical features. The train data is used as input of decision tree classifier without any transformation such that one hot encoding. According to summary of decision tree; the number of terminal nodes is 14, residual mean deviance is 0.3598, misclassification error rate is 0.06667 and accuracy on test dataset is 0.8676 (Table 4). In addition; due to the result of prediction on test dataset, AUC is 0.868 and F1 Score is 0.866 for decision tree even before pruning.

tree.pred	0	1
0	181	25
1	31	186

Table 4: Confusion Matrix of Decision Tree

According to summary of decision tree, variables actually used in tree construction can be ordered as "FullBath", "Neighborhood", "GrLivArea", "BsmtUnfSF", "MSSubClass" "X1stFlrSF", "Exterior2nd" and decision tree fitted on test dataset can be examined on Figure 6 below.

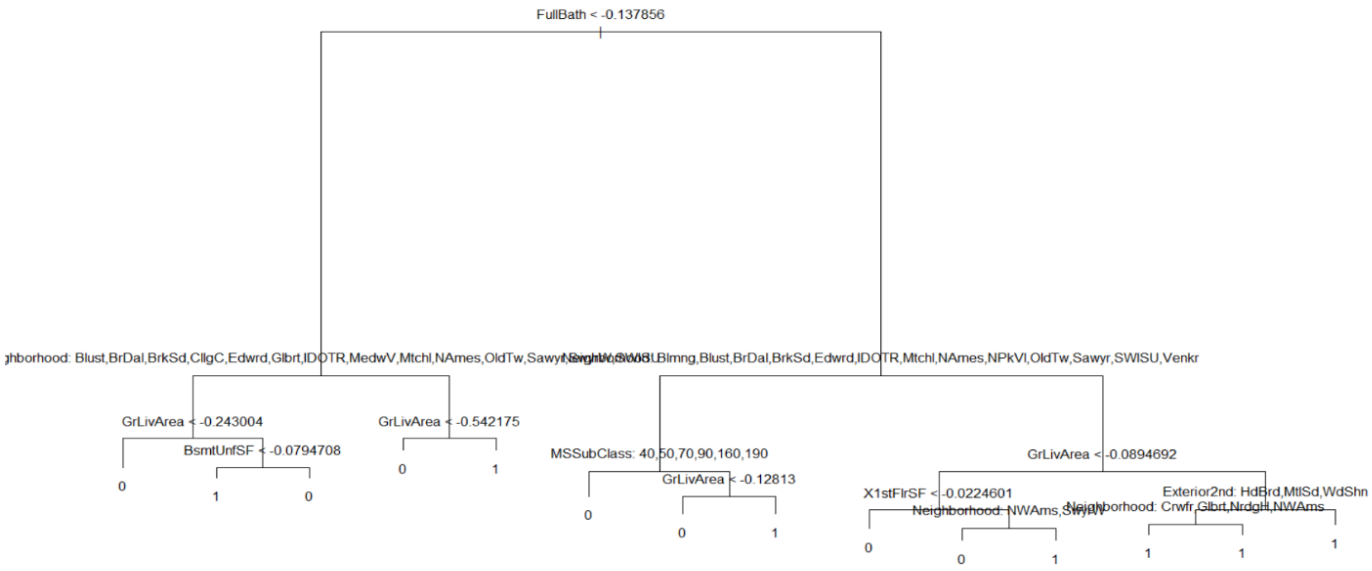


Figure 7: Constructed Decision Tree before Pruning

2.6.2. Decision Tree after Pruning

2.6.2.1. Cross Validation for Pruning

10-Fold cross validation method from tree library is used for pruning. As it can be seen on Figure 8, output of cross validation shows that 9 is the best value for number of terminal nodes.

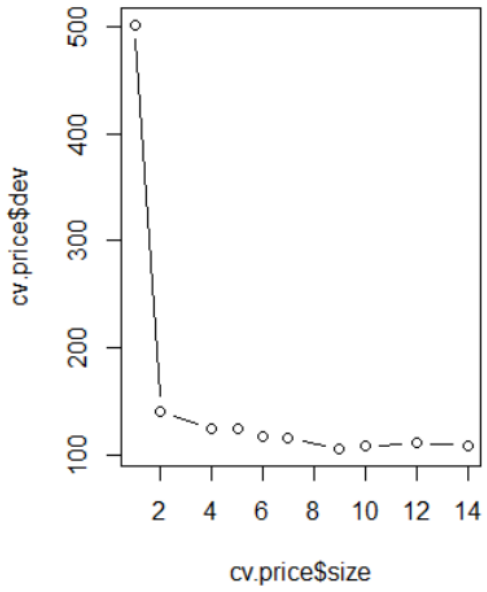


Figure 8: Result of Cross Validation for Pruning

2.6.2.2. Decision Tree after Pruning

After the result of cross validation, decision tree algorithm is fitted on train data with the best parameter value: 9 (Figure 9). It means that decision tree algorithm generates a tree with 9 terminal nodes.

The results after pruning has become more reliable. Accuracy on test dataset is 0.884, residual mean deviance is 0.4725 (Table 5). Misclassification error rate is 0.07104. In addition, due to the result of prediction on test dataset, AUC is 0.884 and F1 Score is 0.883 for decision tree.

Besides, variables actually used in tree construction can be ordered as "FullBath", "Neighborhood", "GrLivArea", "MSSubClass" and "X1stFlrSF". Variables used in tree construction were reduces. Values of evaluation metrics such as accuracy, AUC and F1 Score decreased after pruning.

	y_pred	
y_true	0	1
0	185	27
1	22	189

Table 5: Confusion Matrix of pruned Decision Tree

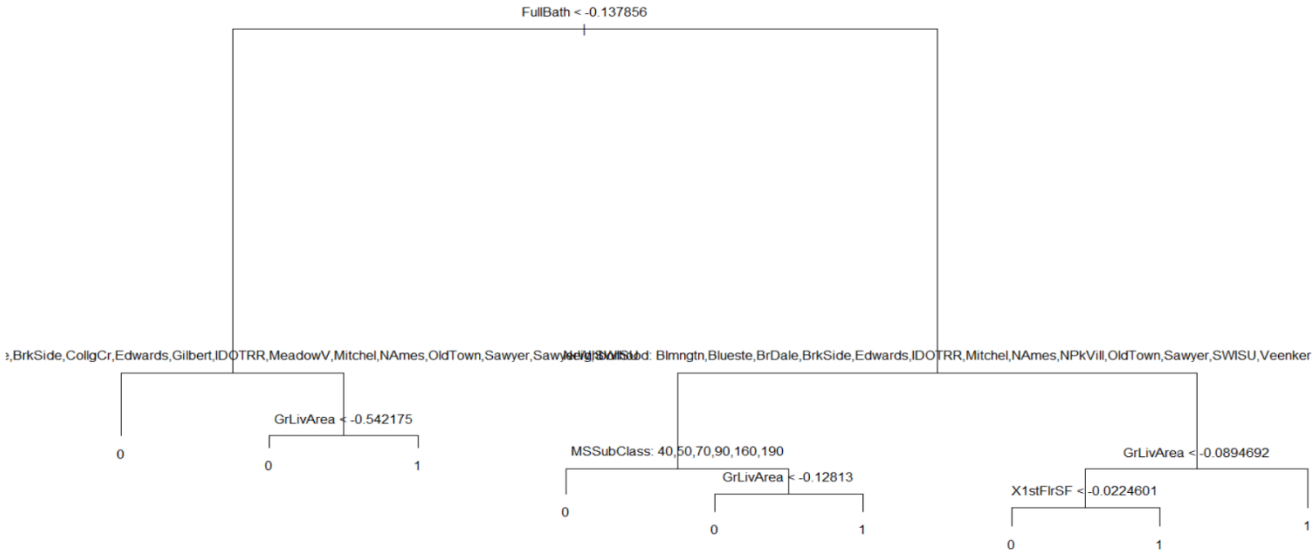


Figure 9: Constructed Decision Tree after Pruning

2.7. Random Forest Classification

Random forest is the third algorithm to predict discrete target variable. Random forest does not suppose any data transformation for categorical features like logistic regression. Thus, cleaned and scaled train data is used as input data of Random Forest. Random Forest has 2 parameters: the first parameter is the same as bagging (the number of trees) and the second parameter (unique to Random Forests) is mtry which is how many features to search over to find the best feature.

Although different mtry values are tried when ntree=500, Random Forest yields the most valuable results when mtry is 7. Accuracy on test data is 0.9219, AUC is 0.9219 and F1 Score is 0.923 (Table 6).

	y_pred	
y_true	0	1
0	185	27
1	22	189

Table 6: Confusion Matrix of Random Forest

The most important variables are shown on below graph. GrlivArea, Neighborhood, OverAllQual and FullBath are the most important features according to MeanDecreaseGini graph. Random Forest explore the real significant features while decision tree focused on local important feature such as FullBath.

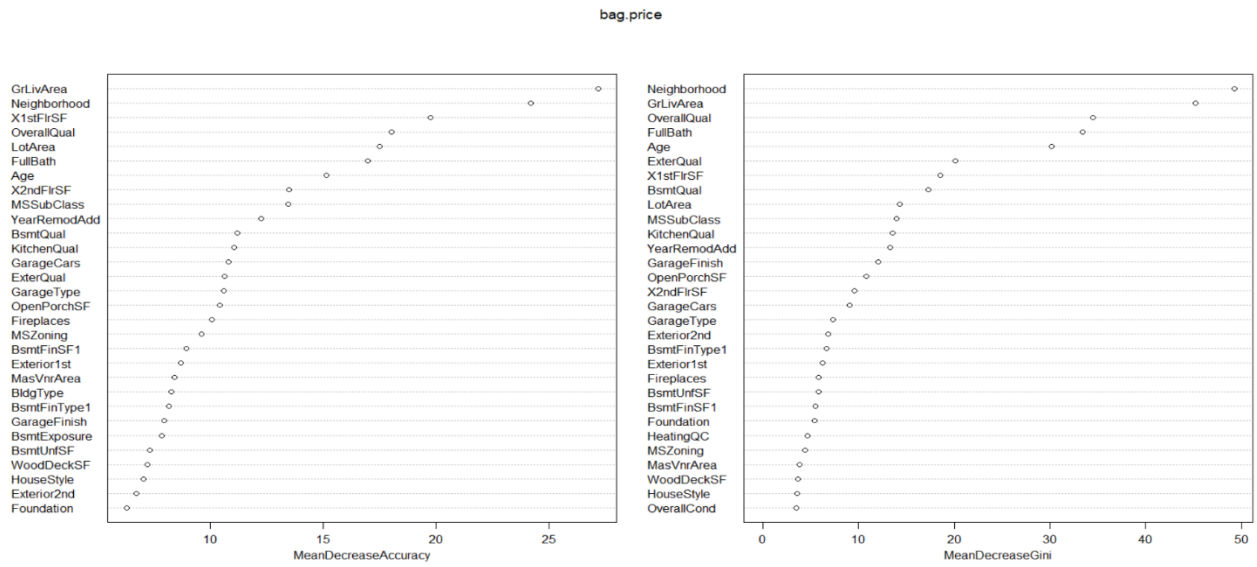


Figure 10: Importance of Features due to Random Forest

In conclusion, the summary of classification models can be examined on Table 7. The results imply that Random Forest on cleaned and transformed dataset generates the most reliable model while logistic regression does not fit very well on given dataset.

	Accuracy	AUC	F1-Score
Logistic Reg with Subset Selection	0.884	0.884	0.8831
Logistic Reg after One Hot Encoding	0.834	0.8345	0.8317
Decision Tree before Pruning	0.8676	0.868	0.866
Decision Tree after Pruning	0.884	0.884	0.883
Random Forest	0.9219	0.9219	0.923

Table 7: Summary of Classification Models

3. CONCLUSION

As a conclusion, we compare the classification (Logistic Regression, Decision Trees, and Random Forest) and regression models (Linear Regression and Extreme Gradient Boosting) with each other. In classification models, we interpreted the F1 score as prior and single number evaluation metric. Random Forest has the higher F1-score with 0.923 and AUC value with 0.923. The accuracy, AUC and F1-Score of Logistic Reg with Subset Selection and Pruned Decision Tree are the same.

When we compare the importance features of Random Forest Classification Model and Extreme Gradient Boosting model, they are not exactly the same. As a result, GrLivArea, Age, Neigborhood, X1nsFirSF, OverallQual, LotArea are the most important features. The feature of FullBath is the dominant splitter in Decision Tree model.

The best model selected from grid has R^2 value of 0.832 and RMSE value of 0.419 which approximately equals to Mean Squared Error value of 0.176. Extreme Gradient Boosting model tuned with grid search showed an improvement over Linear Regression model.

We fitted our Linear Regression model by using 15 features selected. The adjusted 10-fold cross-validation error we got is ≈ 0.164. Test set Adjusted R^2 value is ≈ 0.846 while Mean Squared Error (MSE) is computed as ≈ 0.193

We used RMSE and Adjusted R^2 as an evaluation metric for regression models. The Adjusted R^2 of the linear regression model is 0.846 and RMSE is 0.439. The R^2 of the Extreme Gradient Boosting Model is 0. 844 and RMSE is 0.404. The number of features in both Extreme Gradient Boosting and Linear Regression models is 15. Therefore, the comparison of the R^2 of the model results will not be illogical. Both R^2 (0.849) and RMSE scores of the Linear Regression model is higher than that of Extreme Gradient Boosting model. We conclude that Extreme Gradient Boosting predicts better on the new instances compared to Linear Regression model.

4. REFERENCES

[1] House Prices: Advanced Regression Techniques | Kaggle. (n.d.). Retrieved August 17, 2018, from https://www.kaggle.com/c/house-prices-advanced-regression-techniques

[2] James, G., Witten, D., Hastie, T., Tibshirani, R. (2013). An Introduction to Statistical Learning: with Applications in R. Available from: https://www-bcf.usc.edu/~gareth/ISL/ISLR%20First%20Printing.pdf

[3] Schneider, A., Hommel, G., Blettner, M. Linear regression analysis—part 14 of a series on evaluation of scientific publications. Deutsches Ärzteblatt International, 2010, 107(44): 776–82. DOI: 10.3238/arztebl.2010.0776 Retrieved from: https://www.nki.nl/media/837524/m776.pdf

[4] XGBoost Documentation - Parameters. (n.d.) Retrieved from https://xgboost.readthedocs.io/en/latest/parameter.html

APPENDIX