December 2018

Housing Affordability

STATS 101C Lec2 201

Eugene Ji lotusjy@ucla.edu
Guangkuo Liu guangkuoliu@g.ucla.edu
Yuxiao Liu lyxhshenqing@163.com
Enjie Ma jack666@g.ucla.edu

Abstract

This document is intended for statisticians who would like to try out classification techniques such as logistic regression, KNN, and random forest on the prediction of the affordability of housing in the U.S. Our objective is to predict the affordability of 1,500 houses in Ames, Iowa using 3,500 housing data with 78 variables. We have conducted various classification techniques including LDA, QDA, logistic regression, KNN, random forest, and PCA. The final decision is to use the random forest method that produces the highest accuracy of 98.888% on 750 houses and 98.000% on the other 750 houses.

1 Introduction

Ask a home buyer to describe their dream house, and they probably won't begin with the height of the basement ceiling or the proximity to an east-west railroad. However, there are much more influences price negotiations than the number of bedrooms or the availability of a white-picket fence. We acknowledge the compilation of the Ames Housing dataset by Dean De Cock for use in data science education. It's an incredible alternative for data scientists looking for a modernized and expanded version of the often cited Boston Housing dataset. The 78 explanatory variables describe almost every aspect of residential homes in Ames, Iowa. Our objective is to predict the affordability of 1,500 houses in Ames, Iowa using 3,500 housing data with 78 variables. We have conducted various classification techniques including LDA, QDA, logistic regression, KNN, random forest, and PCA. The final decision is to use the random forest method that produces the highest accuracy of 98.888% on 750 houses and 98.000% on the other 750 houses.

2 Methods and Procedures

2.1 Data cleaning

The training dataset has 3500 observations and 80 variables. The first variable is the observation serial number and the last variable is the affordability, which represents whether the price of each house in Ames, Iowa is above the median. The other 78 variables are potential predictors of the affordability. We started with data cleaning to ensure all values we use to model the relationship between predictors and the response variable are potentially meaningful. We eliminated all observations with the affordability unknown or not answered (2 observations) because they cannot be used to estimate the predictors' effect. We then identified a massive amount of missing values (NAs) in our dataset, which may create great leverages and interfere with the reliability of the model.

Since we want to incorporate as many observations as possible in our model for better predictive power and for a more representative sample group, we need to find methods to fill in these missing values instead of removing all observations with NAs. First, we relied on the data description file that provides detailed explanations for the meaning of each predictor. From the description, we believed that the first column (MSSubClass) is a categorical variable, so we transformed it from numeric to factor. In addition, some of the predictors have strong correlations with others; they are describing an identical aspect of houses with different measurements and perspectives. For example, we found columns from 30 to 38 are various measurements of "basement" and columns from 58 to 64 are all about "garage". As the

construction of an assumption that an observation should be regarded as “no such thing” (in this case, “no basement” or “no garage”) once this observation has an NA in one of the predictors in a certain cluster; however, there are some paradoxes in this dataset. Sometimes, an NA shows first but comes with other categories and numbers in the related predictors. So, we changed the values of those predictors to NAs to conform to the meaning of these observations.

Second, we noticed NAs are meaningful for some predictors as described in the file; they indicate “None” instead of “Missing Value”. For predictors satisfying this situation, we processed slightly different operations according to their types. To be more specific, we created a new level “None” and assigned “NA” to it for categorical variables. While, for numerical variables, we changed “NA” into 0 to better capture the meaning of the observations. (As a reference to the code, we operated categorical variables (column 6, 25, 30, 31, 32, 33, 35, 57, 58, 60, 63, 64, 72, 73, 74) and numerical variables (column 26, 34, 36, 37, 38, 47, 48, 61, 62)).

After that, there are still some observations which contain NAs in the training dataset and we dealt with them separately. Due to the paucity of relevant information, we classified NAs into the most common class for each categorical variables. We understood that limitations may be caused in this way because we altered the class distribution of variables. But thanks to the small proportion of its amount compared to the total number of observations, it seems not necessary to split NAs as the proportion of different classes and assign different classes into them. However, for numerical variables, we tried a more reasonable way. By using the best subset selection with criteria BIC, we chose the optimized predictors for our target variables from all of the numerical variables and predicted those NAs via Multiple Linear Regression. We believed it worked better than filling NAs by the column median or the column mean because it will not significantly impact the distribution of each variable and each multiple linear regression model has an adjusted R-square greater than 70%, which suggested the models are effective and the predictions are valid. (As a reference to the code, we operated categorical variables (column 2, 9, 42, 53, 55) and numerical variables (column 3, 59)).

So far, we completely cleaned all the NAs. Furthermore, inspired by Professor Almohalwas, our next step is variable transformation. According to the cleaned data, we found out that there is only one class (AllPub) in the ninth variable (Utilities), so we deleted it because it has no effect on predicting affordability. We also mutated column “YearRemodAdd” to a categorical variable with level “Yes/No” by checking whether it has the same year number as column “YearBuilt”. In this way, we redefined this variable although the column name remains unchanged. We then redefined “YearBuilt” to “Age of House” (column name unchanged) by subtracting “2010” (The latest year of built) from each observation.

We repeated the same procedures to clean the test data. However, we also unified the levels and types of each predictor with the training data.

2.2 Model Exploration

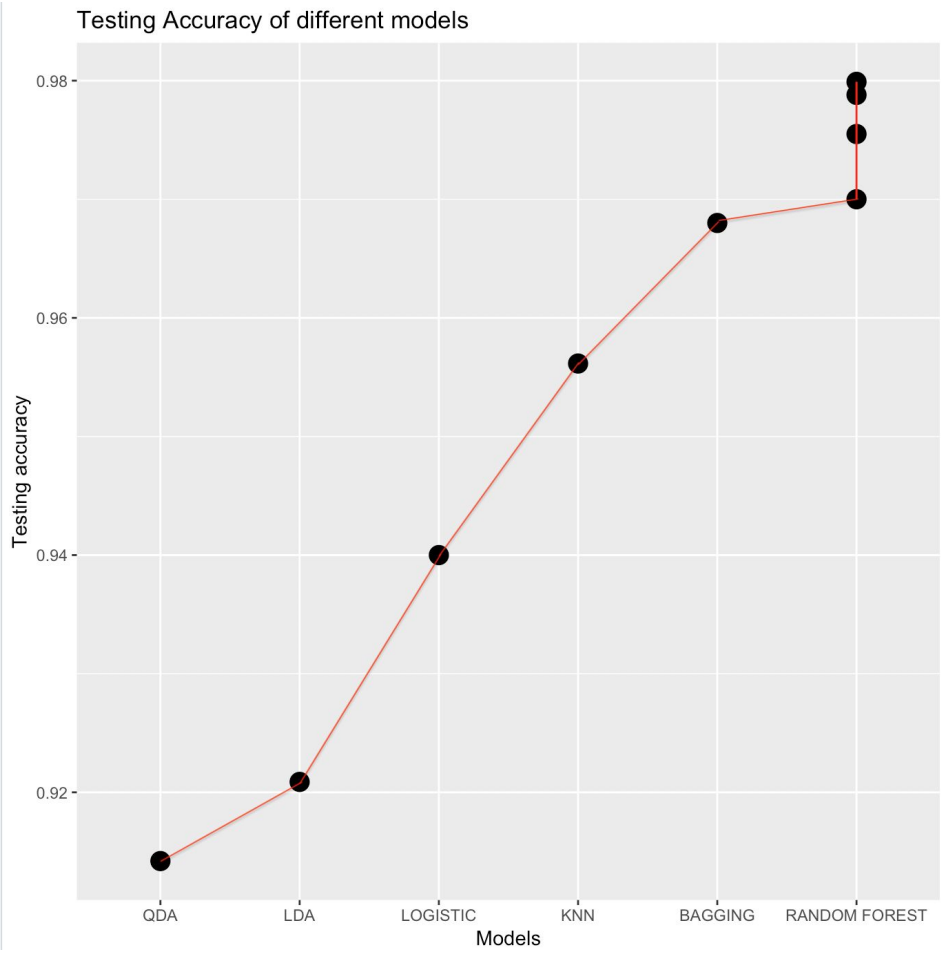
First, we tested how different models perform on the full dataset. We split our full dataset into training, which is 70 percent of the full dataset, and testing, which is the remaining 30 percent. We evaluated the performance of each model by comparing the testing misclassification rate based on the model derived from the training dataset. To ensure that we can compare the misclassification rate between different models, we used the same seed when splitting the data so that for each model, our training dataset, from which we make the model, and testing dataset, from which we calculate the misclassification rate, are the same.

Our models include Logistic regression, Linear Discriminant Analysis, Quadratic Discriminant analysis, and K-Nearest Neighbor. We did not first include Random forest model because we have not learned this model until the end of the quarter. When it comes to the K-Nearest Neighbor, we made sure that we scaled the predictors to make the model valid, since the KNN is an unsupervised model based on distance. In addition, we used a for loop to find the k that gives us the best misclassification rate and our result showed that the optimal k is 1. For the full dataset, KNN outperformed the other models with the lowest testing misclassification rate of 5.33 percent.

After running different models based on the full dataset, we tried to improve our variable selection and to see how much better each of our models can perform on an improved variable selection. When it comes to the variable selection, we concentrated mainly on the collinearity between each predictor. Including highly correlated can result in higher variance in the estimated coefficient, which makes our model inaccurate on the testing dataset. First of all, we detected perfect collinearity by using the function “alias” in R. This function finds predictors that are the linear combinations of other predictors. The result showed that the predictor “TotalBsmtSF” is the sum of “BsmtFinSF1”, “BsmtFinSF2”, and “BsmtUnfSF” and the predictor “GrLivArea” is the sum of “X1stFlrSF”, “X2ndFlrSF”, and “LowQualFinSF”. We removed these two predictors to remove perfect collinearity. We ran into an obstacle when we tried to find the correlation between the remaining predictors because we could not directly find the correlation between categorical predictors and numerical predictors. Nevertheless, it was feasible to find the correlation between predictors for either numeric or categorical. Hence, we divided our predictors into numerical predictors and categorical predictors, and examined the correlations between numeric predictors and between categorical predictors.

For numeric predictors, we first used the “findcorrelation” function from the package “caret”. This function looks at the mean absolute correlation of each variable and removes the variable with the largest mean absolute correlation. In other words, it first identifies highly correlated pairs and then chooses one of the two variables in the pair based on how correlated it is with all the other variables. For the correlation cutoff option in this function, we ran a for loop to decide which correlation cutoff gave us the best testing error rate and we ended up with a cutoff of 0.8. Unlike the regular “correlation” function, which only identifies pair-wise correlation and can be ineffective in models with multiple predictors, this function not only compares pair-wise correlation but also the correlation of one variable against the remaining variables. Besides using the function “findcorrelation”, we also used the function “VIF” in R to examine the variance inflation factor of each numeric variables. Based on the combined result of function “findcorrelation” and “vif”, we finalized our decision of highly correlated numeric predictors to be predictors “YearBuilt” and “GarageCars”. Then we continue to improve our variable selection by using the “StepAIC” function that gave us the variable selection with the lowest AIC score, as a model with the lowest AIC score gives us the best fit. For categorical predictors, we also conducted the Stepwise AIC variable selection. When it comes to investigating the correlation between categorical predictors, we decided to perform a Chi-square test to test the independence between categorical variables. However, we did not come up with a criterion to decide which categorical variables should be removed due to high correlation. Hence we kept all the categorical predictors.

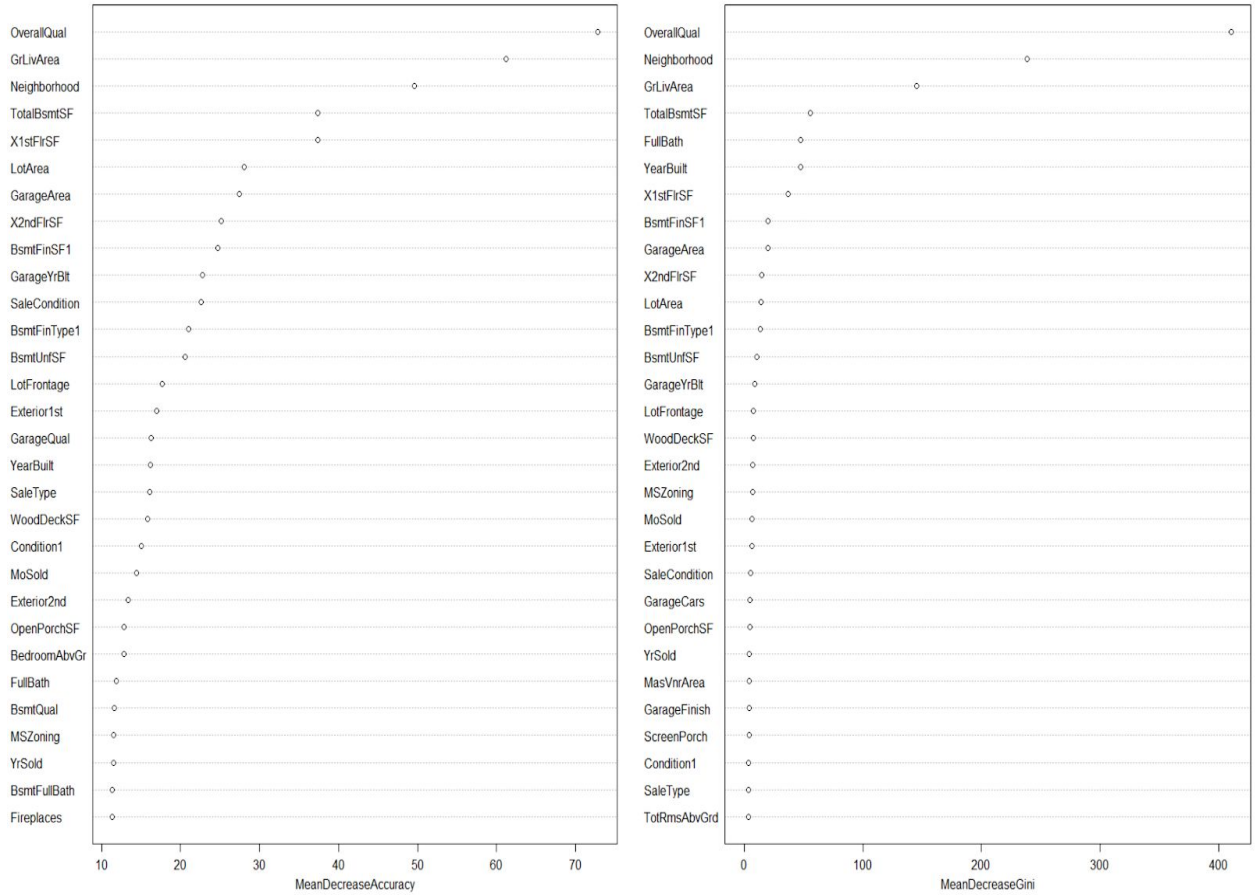
After variable selection, we combined the new numeric and categorical data and ran models through the newly combined dataset. The resulting testing accuracy did not improve a lot, and it seemed that we had reached the limit of the models themselves. Consequently, as we learned new models in class, we tried to apply them on our data to see if we could improve the testing accuracy.



2.3 Finalizing Model Selection

We also fit the tree model into our training data. Unfortunately, both tree and pruned version show error rates that are quite high comparing to other models we have used and they suffer very high variance, although they have very good interpretability. In order to solve the problem of having high variance, we further explored bagging and random forest. We first fit the bagging model with all the variables which shows a quite high accuracy. However, after checking the importance plot, we think some of the predictors are very strong that would make the bagged trees look similar. Hence, the predictions from all the bagged trees will be highly correlated. Based on our observations, we make two improvements: first use random forest that chooses a random sample of m predictors each time a split in a tree is considered and second abridge the 78 predictors into only 10 predictors. The accuracy of this improved model is higher than the original full bagging model and all the other models we have explored, and thus we decided to continue to enhance the random forest model with only 10 predictors. As to choose which ten

variables, we take both “mean decrease accuracy” and “mean decrease gini” into consideration and choose the 10 best predictors accordingly.



For our final model with the lowest misclassification rate, we choose these 10 predictors: OverallQual, Neighborhood, GrLivArea, TotalBsmtSF, X1stFlrSF, YearBuilt, FullBath, X2ndFlrSF, BsmtFinSF1, and GarageArea.

By using the package “randomforest”, two parameters “mtry” and “ntree” are of importance and can be manipulated. For “mtry”, we used a function “tuneRF” in R that can test which value of “mtry” can produce the lowest out of bag error rate. We also used plots to see around how many trees would the model approach a steady rate. After using a couple of different “mtry” and “ntree” the final accuracy rate fluctuates a little for each of them and “mtry” of 4 and “ntree” of 300 gives the best accuracy.

3 Limitations and Recommendations

3.1 Data cleaning

Due to the paucity of relevant information, we classified NAs into the most common class for each categorical variables. We understood that limitations may be caused in this way because we altered the classes distribution of variables. But thanks to the small proportion of its amount compared to the total number of observations, it seems not necessary to split NAs as the proportion of different classes and assign different classes into them. However, in the future projects, we should replace the NAs in the categorical variables with their level proportions to maintain the holistic structure.

3.2 Model Limitation

The random forest method produces the best result, but the method itself has an intrinsic disadvantage which is the poor interpretability. Also, the random forest method tends to favor categorical variables with more levels, so the scores from the function importance in R are biased in some sense. We should continue to explore other powerful categorical methods that do not have such disadvantages.

3.3 Variable Selection

The detailed procedures of our variable selection are mentioned in the second section of the paper. The preliminary attempt of using the PCA analysis in our random forest method failed to improve our result, but we believe the PCA analysis is a good tool to include as much as the variation of the predictors using simple combinations of them. We should keep trying the PCA analysis if we are given more time.

3.4 Investigation of misclassified data

We focused on trying different model selection and variable selection to improve our ability to successfully classify each observation, but we failed to isolate and investigate the misclassified cases that might be hard to classified by statistical model and yet share some similarities. If we were able to investigate the misclassified cases in each model, we could find the common characteristics of the misclassified cases and make manual adjustment to our classification process. These adjustment might allow us to surpass the limit of statistical model.

3.5 Ensemble

After the final presentation, we learned that the ensemble is a very powerful way to integrate our methods we studied this quarter. If we have tried to compare the predictions among some of our best random forest models, we will get a much better final prediction that relying on only one of our best models. What is more, if we apply XGBoost, clustering, and some other good

classification techniques to our data and integrate these methods using an ensemble, our final result will be more desirable.

4 Conclusions

The random forest method is the best classification technique that we experienced with so far. This method with 10 selected variables produces a decent prediction of the affordability of houses in Ames, Iowa with an accuracy of 98.888% on a half of the data and 98.000% on the other half.

The real-world application of our project is to understand the influencing factors of the affordability, i.e., the price of houses, in Ames, Iowa, and we can extend this result to the housing in the United States. The most important influencing factors of the affordability are the "OverallQual (rates the overall material and finish of the house)," "Neighborhood (physical locations within Ames city limits," "GrLivArea (above grade (ground) living area square feet)," and "TotalBsmtSF (total square feet of basement area)." These four factors coincide with our intuition of the most important pricing factors of houses. The least important influencing factor of the affordability is "Fireplaces (number of fireplaces)." In today's houses, fireplaces indeed become more like decorations instead of their practical purposes.

5 References

Andy Liaw., & Matthew Wiener (2018). *Breiman and Cutler's Random Forests for Classification and Regression*. https://cran.r-project.org/web/packages/randomForest/randomForest.pdf

Seni, G., & Elder, J. F. (2010). *Ensemble methods in data mining improving accuracy through combining predictions.* San Rafael, Calif. (1537 Fourth Street, San Rafael, CA 94901 USA): Morgan & Claypool.

Rodriguez-Galiano, V., Ghimire, B., Rogan, J., Chica-Olmo, M., & Rigol-Sanchez, J. (2012). An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 67, 93-104. doi:10.1016/j.isprsjprs.2011.11.002