Albert Na
Kathy Fu
Tiffaney Pi

STATS 101C LEC 2 - "Team KAT" - Final Report

Abstract

In machine learning and statistics, classification is the problem of identifying to which of a set of categories a new observation belongs on the basis of a training set of data in which its category membership is known. In this project, advanced classification techniques were applied to predict the affordability of houses in Ames, Iowa. An analysis of each explanatory variable in the data set was performed to amend or remove data that was incorrect, incomplete, or improperly formatted. Afterwards, several classification techniques were applied using the cleaned data to assess which model would achieve the highest prediction accuracy of the affordability of houses. In the end, a Random Forest model achieved the highest prediction accuracy of 98.89% on the training data and 97.90% on the testing data, resulting in a rank of 7 out of 22 groups in the Kaggle competition.

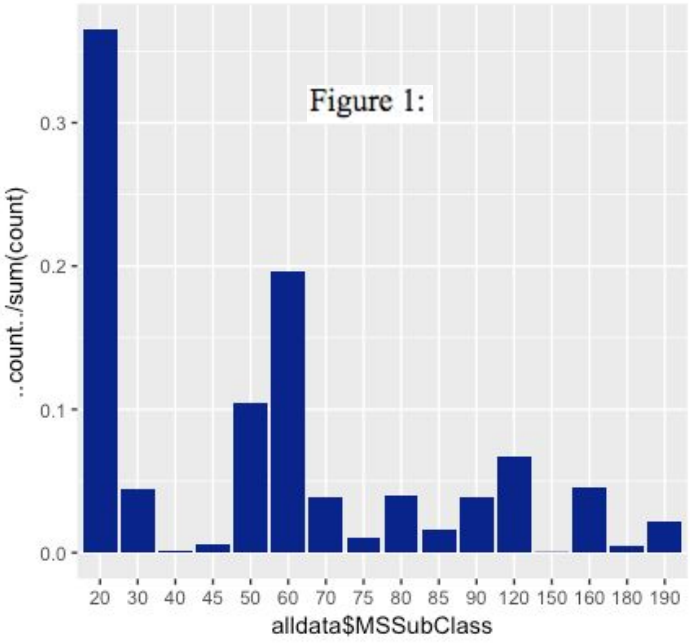
Introduction

Machine learning algorithms have completely transformed the way we can perform exploratory data analysis across a wide variety of fields and for many purposes. One application is the prediction of housing affordability using data from a Kaggle competition, which was our main task of this project. The data consisted of 5,000 homes (3,500 in the training data) in Ames, Iowa evaluated across eighty features such as: zoning classification, lot area, overall quality, number of bathrooms, etc. Our goal is to accurately predict a home as "Affordable" or "Unaffordable" using the best supervised machine learning technique. However, we first needed to determine which of the eighty features could and could not help accurately label the home's affordability status.

Albert Na
Kathy Fu
Tiffaney Pi

Methodology

Before even applying any form of variable selection or model, we took a deeper look into what the actual data sets were comprised of and how they each fit into the context of housing affordability. When analyzing the training and testing data, we combined the two data sets into a new data frame that we will refer to as 'alldata'. Prior to data cleaning, alldata had the dimensions (4998 x 81). The first step to our data cleaning process was to perform type conversion for two cases -- one to coerce variables with character types into factors, and second to coerce some seemingly numeric variables into factors because they were actually categorical.



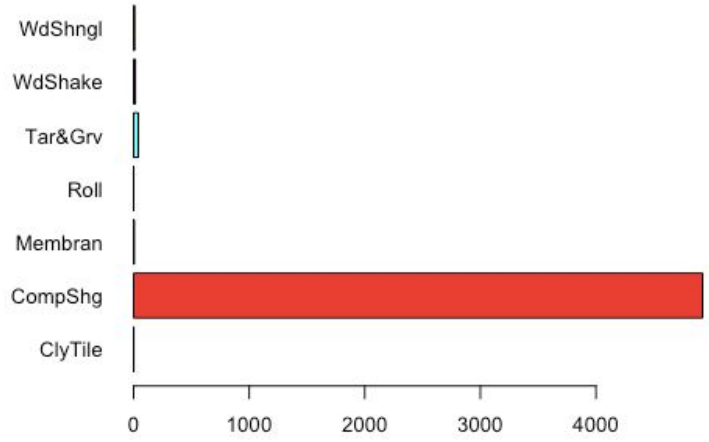
For example, the MSSubClass variable had numerical values, but each number actually represented a type of dwelling instead, so it was actually a categorical variable. From the bar plot in Figure 1, we see that we have the distribution numbers now saved as levels like the left side as opposed to having the values represent numbers on a continuous scale. Another example would be a variable like MoSold, which lists numbers 1 through 12 to represent the month the house was sold, but because these are fixed numbers it was better to code them as factor levels instead.

After type conversion, the next and most important step was to find the amount of NA values in each of the variables and manage them appropriately. We deployed four approaches to

handle the NA values. Upon reading the data short description provided on the Kaggle website, we noticed that many of the categorical variables listed NA as a level labeled “None”, so we turned those NA values into a new factor level called “None”. For example, variable Alley, an NA value meant that there was no alley present for the house. The bulk of our experimentation came from data cleaning. Then the remaining variables that still had NA values, we determined that it would be best to impute either the median, mode, or zero. This was dependent on the variable itself. We imputed zero for the NA values of variables that were dependent on the presence or absence of another variable. For instance, we reasonably concluded that if the MasVnrArea (masonry veneer area) had an NA value, it meant that the MasVnrType (masonry veneer type) was set to be “None”. For other categorical variables, we used imputed the most frequently appeared level, or the mode, for the NA values in our original model, but we are aware that this could skew the proportions of the level distributions. While we wanted to preserve the ratio of these levels, we felt it was still acceptable to proceed with this same imputation process because there were under five NA values in any of the given variables, and would not have negligibly affected our variable proportions. Then for one of the remaining numerical variables, LotFrontage, we imputed the median value for the NA values to ensure that the values inputted were not greatly affected by outliers.

After tabling the variables, we noticed that many of the categorical ones had infrequently occurring levels, and most of the values in the variable were categorized under one level. To identify variables from our model that had minimal variation, we used the caret library’s function nearZeroVar(), and removed all of the 24 variables that were tagged as such. With the remaining categorical variables, we removed some other variables with a lot of rare levels and where their

Figure 2:



most frequently observed level was true for 4000+ observations. A visualization of one such variable as shown in Figure 2, RoofMatl (roof material) shows that the CompShg level is the observed level for almost all of the observations, while others hardly occur. We concluded that variables like this would not be good predictors of affordability.

To simplify our model, we also made sure to combine variables to create new variables because some of the information was redundant. The three new variables created were AgeofHouse, Bath and BsmtBath. Initially the AgeofHouse we chose was defined as YearRemodAdd - YearBuilt, but we later modified that variable to be defined by 2010 - YearBuilt, where 2010 was the latest year in the YearBuilt category. Next, we combined the full and half bathrooms for the basement then again for the bathrooms not in the basement. BsmtBath was defined by BsmtFullBath + .5*BsmtHalfBath, and similarly Bath was defined by FullBath + .5*HalfBath. After creating these new variables, we discarded the used variables (YearBuilt, BsmtFullBath, BsmtHalfBath, FullBath, HalfBath) from the alldata data frame.

With so many variables, we thought there would bound to be correlated variables. Before our type conversion step, we evaluated the correlation matrix in Figure 3 for numerical variables, where a dark opaque red indicates a 1.0 correlation and a dark opaque blue indicates -1.0 correlation.

Albert Na
Kathy Fu
Tiffaney Pi

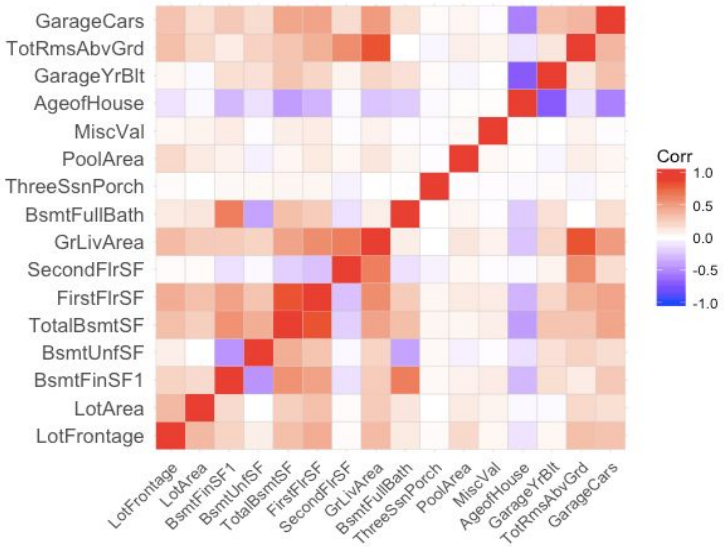


Figure 3:

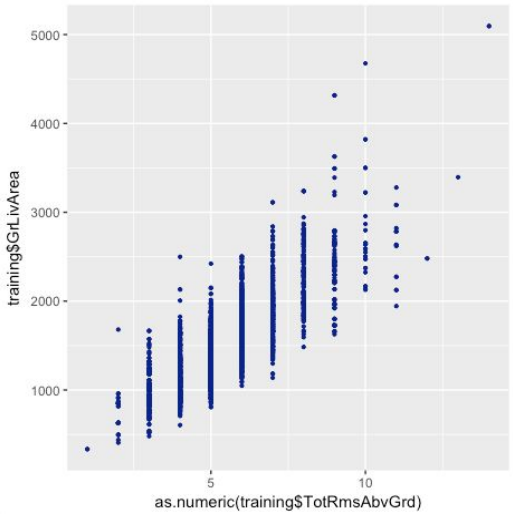


Figure 4:

We found that these particular variable pairs had correlations above r = 0.7 or below r = -0.7: GrLivArea & TotRmsAbvGrd, AgeofHouse & GarageYrBuilt, FirstFlrSF & TotalBsmtSF, and GarageArea & GarageCars. We removed one of each of these pairings trying our best to keep the numerical variables. On the scatter plot in Figure 4, we see the relationship between the variable pairing TotRmsAbvGrd (total rooms above grade) and GrLivArea (above grade living area) on the training data. These two variables have a positive correlation of 0.81, which is indicative from the graph.

For the remaining numerical variables, we also considered their density plots to investigate if the affordable versus unaffordable distributions would overlap or not. Some notable variables that presented a clear division between affordable and unaffordable houses were SecondFlrArea. From its density plot in Figure 5 we see that GrLivArea would be a great

predictor because the distributions for the affordable versus unaffordable observations can be divided easily. On the other hand, variables like LotFrontage and LotArea had affordability distributions that were practically the same in Figure 6, so we determined that they were bad predictors in predicting affordability and removed them from the model.

Figure 5:

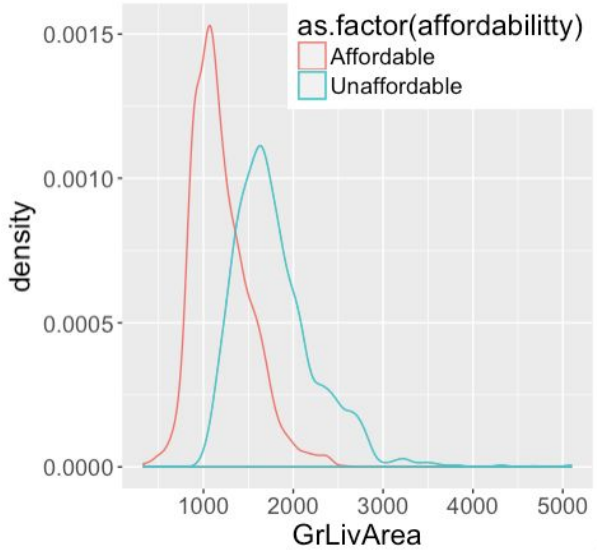
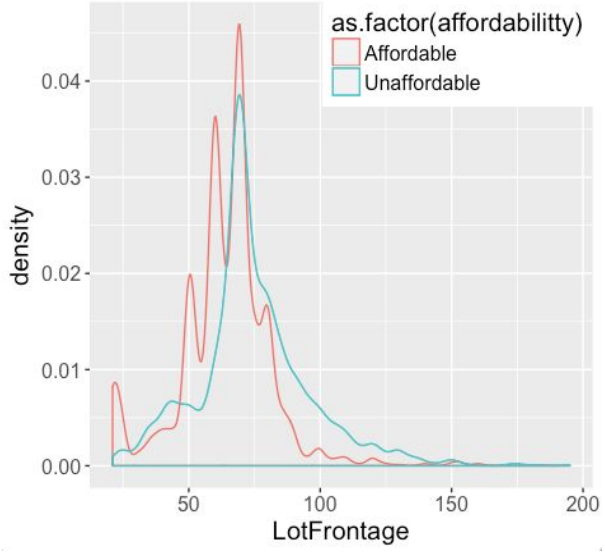


Figure 6:



Lastly, we omitted some other miscellaneous predictors. We made sure to omit the variable Obs because it wasn't a real predictor for affordability. Then we omitted a couple of other categorical variables which we felt still had very infrequently occurring levels even after applying the nearZeroVariance function from earlier. We also removed the variable Neighborhood because there were so many factor levels that we felt it would negatively affect our model because randomForest tends to favor predictors with a lot of levels. However, a possible solution to this which is listed in our recommendations section too, was to group levels together to bypass this problem.

Albert Na
Kathy Fu
Tiffaney Pi

Model & Main Results

After cleaning the data, several different classification methods were used to determine a model that would minimize the prediction misclassification rate. The first model tested was a logistic regression model. Forward stepwise selection was used to find the most significant variables to be included in the model, resulting in 37 predictors. Backwards stepwise selection was also tested resulting in 41 predictors, however the model using the variables selected from forward selection resulted in a higher prediction accuracy. Using the 37 predictors for forward selection, a logistic regression model was fitted to obtain a 93.11% prediction accuracy for affordability of houses.

The next models tested were lasso/ridge logistic regression to check if the results would be improved if we tested for collinearity. In order to convert our categorical variables into numeric values that can be used for lasso/ridge regression, we used the model.matrix function to expand our factors into a set of dummy variables. We then used cross validation to find the best value of lambda that would minimize MSE and fit lasso/ridge regression using each respective best lambda. For the lasso model the prediction accuracy came out to 95.33% while the ridge regression model came out to 96.44% prediction accuracy.

Next, a random forest model with mtry = 8 was fitted to obtain a prediction accuracy of 98.89% We tried using importance plots in Figure 8 to find the most significant variables and fitting a new model using only the most significant variables, but the models that included only the top 10 or top 20 most significant variables failed to outperform the model that included all the variables.

Albert Na
Kathy Fu
Tiffaney Pi

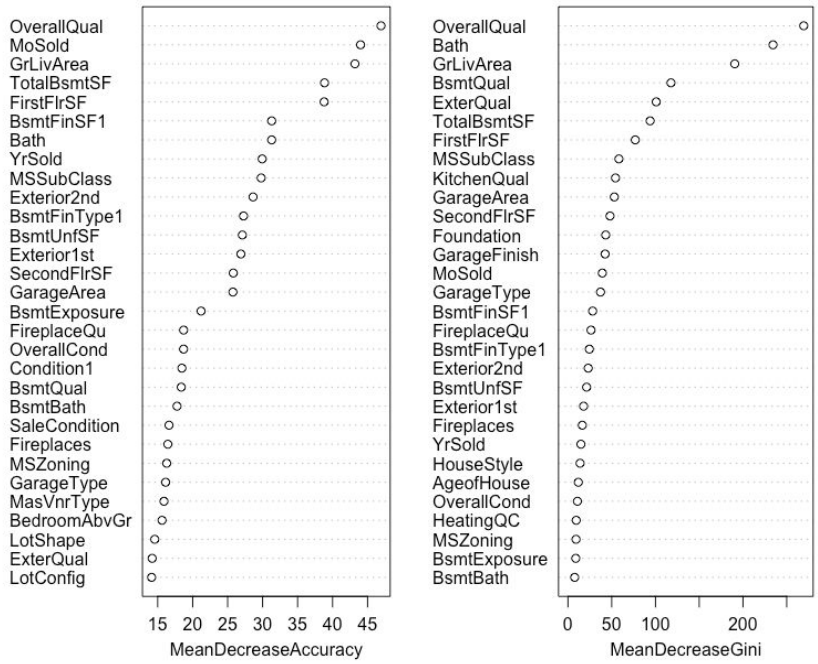
Lastly, we checked whether boosting would reduce the bias and variance of the model. Using a bernoulli distribution with 1000 trees, the prediction accuracy came out to 98.66%, which was slightly worse than our best model.

The off-diagonals of the confusion matrix shown in Figure 7 represent the number of misclassified homes in the training data set after our optimal Random Forest model was run. The misclassification rate is 2.12%.

Figure 7:

	Affordable	Unaffordable
Affordable	1705	34
Unaffordable	40	1719

Figure 8:



Limitations

One limitation of the project was that the public leaderboard was not representative of the rankings on the private leaderboard. Since only 3 submissions a day were allowed on Kaggle, our group split the training data 70/30 to test our model accuracy through testing on the training data. If the models we were testing failed to beat the accuracy of our best model, we would give

up on that model and start from scratch. We realize now however that although the models we were testing may have had a worse prediction rate on the training (public) data, it could have outperformed our best model on the testing (private) data. Another limitation is that randomForests are hard to interpret with large sample sizes and we don't know if our sample size is too large that it is affecting the accuracy of the randomForest model. It remains to be tested whether removing more observations would increase the prediction accuracy of our best model.

Conclusion and Recommendations:

After data cleaning, our final model was run on our training data set with 37 predictors and 3498 observations. Using this subset of data, we fit several classification techniques such as logistic regression, lasso/ridge logistic regression, XGBoost, tree models, and random forest, only to find that the random forest model resulted in the lowest misclassification rate (0.011). For future studies and with more time we should consider bringing in outside data, creating more variables from the original data, and perform ensembling by combining several models together. For Neighborhood, create different buckets for the levels, possibly five, so that the randomForest model won't favor this variable as much in MeanDecreaseAccuracy. We also could have added a Total Square Footage variable since the size of a house is a widely regarded top determining factor of housing prices and affordability. This new variable would have simplified our model instead of spreading out square footage into separate variables. We also could perform a deeper analysis into the misclassified observations and explore their commonalities to make more informed future predictions.