Kaggle Report

**Introduction & Description of Original Set**:

In this competition the goal was to predict the housing prices of homes in Ames, Iowa using machine learning techniques. The data set includes 79 explanatory variables which deal with almost every aspect of a house including square footage, number of bedrooms, and the condition of the house. The original dataset of course had 79 variables and many of them were missing values. It was obvious that a lot of work would have to go into cleaning the data and making sure that there were no NA values present in order to ensure the models would run correctly.

**Cleaning**:

In my work I had 3 distinct data cleaning phases. The first of which was to just make sure that I could run a basic OLS regression using the data set as unmodified as possible. So, to start I imported both the train and the test data into R and used stringsAsFactors = F which would turn out to cause problems later on. For the first regression it didn’t matter however, and I went about ensuring that no NA values were left in the data so the model could run. I made far too many changes to go through each one individually, so I’ll just go through some key changes and the rationale behind them. For NA values that were integers/numeric I imputed the mean value of the column. For NA values that were characters I generally would take the most common type in that variable and replace the NA with it because most of the variables in the data set had disproportionate representation in the variables which would lead to one being far more common than the other. I removed the Utilities variable in my first round of cleaning because there were only 2 variable outcomes and one of them had only 1 instance so it didn’t make sense to keep the column. In the KitchenQual variable there was an instance where a rating was given despite no kitchen being present, so I replaced that instance with “No Kitchen”. For the Exterior variables in the test set I replaced the NA’s with plywood because in every case where the roof style was flat and the material was tar and gravel the exterior was always plywood. For the variable KitchenQual in the test set I used TA as it was the average rating but later decided to remove the variable because it was too much of a headache. After all the changes (some of them ended up being useless because I later removed the variables or found a better way to do the job) I could run my baseline OLS regression. I got a pretty bad score with that which will be shown later in this report.

I then began my second round of cleaning where I changed some values that were integers to characters (I should have changed them to factors which would become a problem later). I also found two interesting cases where MasVnrArea = 0 yet MasVnrType != “None”. I thought that if the area equaled zero then there was no veneer present and it should be recoded to none in those two circumstances. I then got rid of two variables I found to be redundant because they added up to another one (I made this decision based on the ACT example we saw in the admissions data set). After looking into documents on t-learn I saw that some of the integers should be recoded to numeric values, so I applied that to my data set. The last change I made during this round of cleaning was to remove columns that, based on my own theory, did not seem like they would have any significant bearing on the price of the home. This level of cleanliness in the data set would allow me to run the OLS regression again, forward/backwards/hybrid feature selection, and Ridge/LASSO regression. When I reached SVM my decision to make most of my categorical variables characters came back to haunt me as the difference in feature levels between the train and test set didn’t allow me to run SVM models. This was a big headache and I didn’t figure out the problem for quite a while. Eventually I did my last round of cleaning and changed all characters to factors and equalized the factor levels in both the train and the test set which allowed me to run SVM and random forests with no problems.

**Description of Final Data Set**:

After all the modifications I made to the data set it ended up being 57 variables (excluding the response variable). I kept all the rows in the original dataset because I didn’t want to throw away a whole row of data points. The data frames in the final data set are comprised of numerics, integers, and factors.

**Methods Used & Results**:

* Baseline OLS (As unmodified as possible – Score: 0.47079
* Baseline OLS (Clean data set) – Score: 0.17405
* Backwards Feature Selection – Score: 0.15531
* Forwards Feature Selection – Score: 0.17405
* Hybrid Feature Selection – Score: 0.15531
* Ridge Regression – Score: 0.14639
* LASSO Regression – Score: 0.14638
* SVM Regression (Polynomial Kernel) – Score: 0.32288
* SVM Regression (Radial Kernel) – Score: 0.17727
* SVM Regression (Linear Kernel) – Score: **0.14039**
* Random Forest (All Variables) – Score: 0.16263
* Random Forest (Random 26 Variables) – Score: 0.16057
* Random Forest (Random 26 w/ Importance) – Score: 0.16063
* Boosted Random Forest – Score: 0.19719

**Discussion of Results**:

To begin, the first regression I ran was terrible because the data had yet to be cleaned. After cleaning the data and running the OLS regression again my results improved remarkably. Out of the three feature selection methods the best two were backwards and hybrid as they returned the same RMSLE (which was even better than OLS). My results improved again when I used Ridge and LASSO regression, with LASSO being just a small amount better than Ridge. Out of the 3 SVM models I ran the best was the model using a linear kernel which returned my best result, which was **0.14039**. Lastly, I ran four different random forest models with one using boosting. I was surprised to find that none of the four random forest models outperformed the linear kernel SVM model.

**Conclusion**:

Overall, I would say that I learned a ton throughout the course of this project. I encountered so many errors in my code and had to fix all of them which gave me a better understanding of how the functions work. I learned when dealing with regression it is typically easier to have factors rather than characters. Some of the obstacles I encountered included forgetting a few NA values in my data set (which took hours to figure out because the error said nothing about NA values), models not working because of different factor levels, and some weird matrix stuff that I avoided using the caret package for my ridge and lasso. I enjoyed the project thoroughly and I’m proud of the work that I did. Next time I’ll definitely make sure StringsAsFactors = T.