Predicting Housing Prices in Des Moines, Iowa

Nicolas Pacheco Soliz

ID: 704821096

Stats 101A Lecture 1 Professor Almohalwas

Abstract:

In this research study, I attempted to predict the housing prices of sales prices in Des Moines Iowa through a Kaggle competition for STATS 101A. After a lot of data exploration, imputation, transformation and manipulation, while keeping in mind problems such as overfitting with the training dataset, I decided on my best model. Using the data set given, I created a training model that resulted in an R^2 of .9024.

After examining relationships between the columns of data and sales prices with my testing model, I acquired an R^2 of .9189, given a final rank result of 44 in my respective lecture. The model I eventually decided to use had 6 predictors, including transformations and interactions to strengthen the model as well as overcome violations in linear regression assumptions.

Introduction:

The initial dataset utilized for this model was compiled of 81 columns and 2500 rows, with each row representing a house in Des Moines, Iowa, and each column representing a quality of that corresponding house. Using a multiple linear regression model, there were many possibilities for which predictors to use. My goal was to create a model based off this training dataset with 81 columns and 2500 rows, and test my model using the testing dataset with 80 columns and 1500 rows. The training dataset contained the column SalePrice, which was the response variable in my multiple linear regression model, and the testing dataset contained the same variables except for the SalePrice variable. I also created several variables in the pursuit of strengthening my model, given me a final training dataset with 85 columns and 2500 rows. I added the columns YearsOld, ExterQual_Num,KitchenQual_Num, neighbor_rank, giving the four extra columns. YearsOld was the years old rather the actual year of the house, and the remaining three were quantified versions of three originally categorical predictors. However, none of these resulted in any improvement in my R squared, nor improved my model in a significant way. However, it is worth noting that these were attempted, as they all lead me to my final model. With the model I created, I only ended up using 6 of the columns as I found many

issues with multicollinearity using too many predictors or ended up with models that resulted in overfitting.

Methodology:

To create the best model for the testing dataset, I first investigated a couple of predictors based of my intuition. I knew right away that some variables would have a much higher influence than others. For example, I knew that the number of fireplaces could not be compared with the age of a house. In general, most people tend to look heavily at how old a house before looking at the number of fireplaces. This was how I initially created my model.

The next steps consisted of eliminated NA values. For quantitative predictors, I did this my replacing the NA values with the median of my dataset. For qualitative predictors, I simply looked at the scale given for that predictor, and found which predictor was "the most average". I only did this once, however. For the most part, I replaced the missing values of numeric predictors.

After discovering a couple of predictors that I knew would most likely help predict the SalePrice, I began to test the assumptions of a linear regression model. The first thing I did was check the assumptions of multicollinearity. The model would be invalid if had any predictors with VIF greater than 5, so I made to sure to eliminate any variables, one by one, if they violated this assumption. Ultimately, besides using my intuition, this is how I created my initial valid model.

Next, I focused my efforts on transforming my response and predictor variables. When I checked the diagnostic plots for my model, I noticed some of the assumptions such and normality and linearity were violated. To relieve my model of these violation, I used transformations. Using a box cox transformation, I knew which predictors I had to transform to help my model become more valid.

The three steps above were the main framework as to how I ended up choosing my main model for submission. Choosing my predictors, checking for multicollinearity, then transforming my predictors, in this order, was how I ultimately chose my best model.

To find the best possible R squared, I looked at individual relationships between certain predictors and the SalePrice. For example, I created boxplots for specific categorical predictors and SalePrice. If I found a categorical predictor showed boxplots a large variation in means, I included them in my main model. If this were the case, I would add the predictor my model, then check my new R-squared and check for multicollinearity and violation of other assumptions.

Another method I used was quantifying my categorical variables. This was useful with categorical variables with plenty of categories, such as the "Neighborhood" predictor. This predictor had 25 levels, and while it helped my R squared significantly, I knew I could utilize a predictor in a better way by quantifying it. So, I used dplyr to get an estimate of the all the median SalePrice per each neighborhood, then I ranked each neighborhood accordingly to its

median SalePrice. Once I did that, I basically narrowed down "Neighborhood" into one predictor, which still helped explain the variation in y. Though I did not end up using this in my final model, it helped me get a better sense of what I would use in my final model.

While I was eliminating certain variables in my data because of a violation in multicollinearity, I realized I could instead use some of these variables to much advantage. Using my intuition and logic, I checked to see for interaction between any predictors. As a result, I was able to increase my R squared with some interactions, as shown the following section. By taking the listed steps above, I was able to find my best model.

Results:

As a result, I ended up going with my model:

```
tkaggle_regression1<- lm(log(SalePrice) ~ OverallQual*Neighborhood+ I((GarageArea)^.5)+
KitchenQual+TotalBsmtSF*GrLivArea , data= kaggledata.yvalues1)
```

A short summary to Show the R squared and the statistical significance of the model:

```
Residual standard error: 0.1238 on 2443 degrees of freedom
Multiple R-squared:  0.9024,   Adjusted R-squared:  0.9002
F-statistic: 403.5 on 56 and 2443 DF,  p-value: < 2.2e-16
```

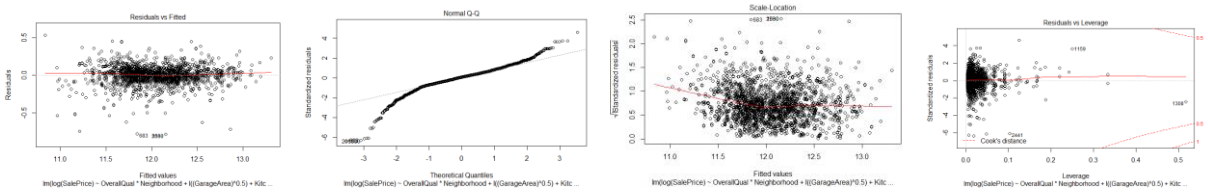
Proof of Noncollinearity:

```
kaggle_regression1_no_interaction <-lm(SalePrice ~ OverallQual+Neighborhood+
GarageArea+KitchenQual+TotalBsmtSF+GrLivArea , data= kaggledata.yvalues1)

vif(kaggle_regression1_no_interaction)
```

	GVIF	Df	GVIF^(1/(2*Df))
OverallQual	3.533258	1	1.879696
Neighborhood	4.809556	24	1.033262
GarageArea	1.999515	1	1.414042
KitchenQual	2.984079	3	1.199872
TotalBsmtSF	1.833554	1	1.354088
GrLivArea	1.928581	1	1.388733

Diagnostic Plots:



Discussion:

From above, we can see that I used predictors OverallQual, Neighborhood, GarageArea, KitchenQual, TotalBsmSF and GrLivArea and its followed transformations and interactions. Checking my VIF output before any transformations or interactions, we can see there was no violations with the multicollinearity assumption. The only interactions I used were OverallQual*Neighborhood and TotalBsmtSF*GrLivArea, as I found these interactions significantly improved my R^2.

Checking the diagnostic plots, however, we can still see there is still an existence of outliers. While the transformations helped my model significantly, some things I would explore given more time would be the existence of these outliers, and how I could use them to either strengthen my model, or perhaps see if I could eliminate all together. Also, while I applied the transformation to overcome nonlinearity and non-normality, it did not fix the issue completely. So, once again, given more time, I would have explored what to do with these outliers and investigated ways I could have overcome the violations of these assumptions.

While I used the same methodology throughout my entire thought process in this project, I found a few models that were all likely contenders for my strongest model. The first model included neighborhood which gave me the highest R squared(which was .9277 reported on kaggle), though I later realized it was invalid after checking my VIFs again. I began eliminating variables so that I did not violate multicollinearity conditions, giving me the next highest R squared, as well as model with the least predictors. However, I still wanted to test some of the other predictors, so I quantified neighborhood, and added some predictors, giving me the next highest R squared. However, this did not give me the highest R squared, and I had too many predictors with this model. This is known as **model 2** in my R code, while **model 1** was the actual model I decided on suing. Even though I quantified some categorical predictors, such as neighborhood, I was not able to get a better R squared and ended up using even more predictors with this model. So, I ended up not going with **model 2.**

Overall, I chose the model that gave me the highest R squared but was also still valid and gave the most simplicity. The only complex thing about my model was the fact that it had Neighborhood, which had more levels. Some levels were not significant, but this did not seem to worsen my model. At first, using this predictor seemed like it would cause potential problems with overfitting, though as I explain below, it actually had the opposite effect. The addition of this predictor did not have multicollinearity issues, and though it contained some insignificant levels, it strengthened my model and had an even better R squared in the testing dataset.

Limitations and Conclusions:

As noted above, my model is limited in a few ways. There is still the existence of outliers, and I could have done something about these outliers to make a stronger model. It is also evident that while the transformations I applied to my model relieved some normality issues, they did not completely fix the problem as we can see from one of the diagnostic plots. This is another limitation that my model had that with more time, could have been further investigated and fixed. That being said, my model did not have any issues with overfitting or multicollinearity, making it an overall strong model. It still predicted about 91% of the variation in y. Another note to make is that while my training data set had a .9024 R-squared, by testing dataset had a .9181 R squared. While it may seem like Neighborhood might create overfitting in my model, it had the opposite effect. I did not get an R squared that was lower for my training model than my testing. Ultimately, the model does an excellent job at predicting SalePrice, but could do a better job at predicting sales prices of houses that are considered outliers.