Scott Jue
MSDS 410
2/12/2023

Modeling Assignment #5: Finalizing the Model – Variable Selection Procedures and Validation

Introduction

For this analysis, I will be finalizing a linear regression model to predict SalePrice from the Ames, IA housing data. The focus of this analysis will be feature selection and model validation using a train/test split. Building off of previous work, I will be using the same sample population definition which are observations that are in residential zones, of single-family homes type, have normal sale conditions, have a total square foot less than 2,800, have less than or equal to 9 total rooms above ground, have all public utilities, has a basement that is no bigger than 2,000 square feet, and have at least a 1 car garage, at least 1 full bath, at least 1 kitchen above ground, and at least 1 bedroom above ground. These filters and drop conditions have been applied to the original dataset so that we can only model observations that meet the criteria of the sample population of interest. From my previous exploratory analysis, it was discovered that the distribution of SalePrice is right-skewed. Therefore, I will be using logSalePrice as my response variable (Y) in order to obtain a normal distribution of the response variable. I will then perform a back-transformation on the predicted values to get them back in the original scale for interpretation purposes.

Previous modeling work only involved continuous variables, so for this final model I will be expanding my variable selection and will also be considering categorical variables in addition to the continuous variables. The first categorical variable I was interested in was HouseStyle since the type of house style (1 story, 2 story, etc.) might be related to the home's value. Fitting a simple linear regression model using HouseStyle resulted in a R-Squared value of 0.15. I considered this R-Squared value to be too low for this variable to be considered for my model. Next, I decided to look at Neighborhood since home values tend to fluctuate depending on certain neighborhoods or locations. Using a side-by-side boxplot (Appendix A) of each Neighborhood, I compared SalePrice values between each Neighborhood and determined there was a difference in means which can be seen in Appendix B along with other summary statistics for each Neighborhood. The "Blmngtn" neighborhood only had 1 observation, so I removed this observation as there are not enough data points. Additionally, to simplify this variable and improve interpretability I have decided to split the neighborhoods into 3 different groups (Low, Mid, Higher) based on the mean sale price of the neighborhood. The Low group is Neighborhoods that had a mean SalePrice less than $150K. The Mid group has a mean SalePrice between $150K and $225K. The High group has a mean SalePrice of more than $225K. A table of the Neighborhoods and their respective group can be found in Appendix C. Additionally, the summary statistics by Neighborhood Type can be seen in Appendix D. We can see a difference in means which could indicate we may be dealing with unequal slopes. I then created dummy codes for these newly created Neighborhood types. Fitting a regression model using the dummy codes for Neighborhood Type, resulted in a R-squared of 0.35. Since the Neighborhood Type can explain more than one-third of the variance observed in logSalePrice, I will consider these dummy variables in the final model.

Results

1) To assess model performance, I split the sample data into a 70/30 train/test split using a uniform random number. Using this train/test split, I can then perform a cross-validation analysis of the model. Below is a table of the observation count for the train/test split.

Train Observation Count	1246
Test Observation Count	535
Total Observation Count	1781

2) Next, I have selected the below variables as potential predictor variable candidates. I then used automated variable selection to select the best variables to use as predictor variables in the final model. The automated variable selection approaches I used was forward, backward, and stepwise from the MASS library in R. In addition to these three approaches for variable selection, I will also compare a "junk" model which contains variables that were not selected to be potential predictor variable candidates. After implementing these variable selection approaches, I then calculated the VIF values for the variables selected for each approach to check for any collinearity which can be seen below.

Forward

TotalSqftCalc	TotalFloorSF	BsmtUnfSF	TotalBsmtSF	GarageCars	GarageArea
195.665013	94.465255	77.717421	57.160765	4.019563	3.723480
TotRmsAbvGrd	Neighborhood_typeLow	Neighborhood_typeMid	HouseAge	BedroomAbvGr	LotArea
3.507327	3.007471	2.347568	2.050208	1.819369	1.235405
QualityIndex	LotFrontage	KitchenAbvGr			
1.222475	1.216152	1.026274			

Backward

GarageCars	GarageArea	TotalFloorSF	TotRmsAbvGrd	HouseAge	Neighborhood_typeLow
4.019563	3.723480	3.561029	3.507327	2.050208	1.963456
BedroomAbvGr	TotalBsmtSF	Neighborhood_typeHigh	LotArea	QualityIndex	LotFrontage
1.819369	1.605872	1.311294	1.235405	1.222475	1.216152
BsmtUnfSF	LowQualFinSF	KitchenAbvGr			
1.201832	1.052883	1.026274			

Stepwise

TotalFloorSF	TotalSqftCalc	GarageCars	GarageArea	TotRmsAbvGrd	Neighborhood_typeLow
6.691355	5.484006	4.017784	3.721959	3.409698	3.003412
BsmtUnfSF	Neighborhood_typeMid	HouseAge	BedroomAbvGr	LotArea	QualityIndex
2.352747	2.345636	2.024817	1.819368	1.234926	1.219748
LotFrontage	KitchenAbvGr				
1.210495	1.025627				

Junk

QualityIndex	OverallQual	OverallCond	GrLivArea	TotalSqftCalc
46.241590	28.016995	27.871758	2.616352	2.161088

We can see that there are quite a few variables with high VIF values (greater than 5) that indicate collinearity. Therefore, I removed some of the variables that displayed high VIF values and repeated the automated variable selection process. This resulted in all three approaches selecting the same variables which all had VIF values under, so I was satisfied with the variables that were selected. The variables and VIF values can be seen below for the three variable selection procedures as well as the junk model.

Forward, Backward, and Stepwise

TotalSqftCalc	TotRmsAbvGrd	HouseAge	Neighborhood_typeLow	BedroomAbvGr	BsmtUnfSF
2.851801	2.688450	1.923467	1.894264	1.769305	1.740417
GarageArea	Neighborhood_typeHigh	LotArea	LotFrontage	QualityIndex	LowQualFinSF
1.569460	1.308194	1.232604	1.211415	1.199052	1.040142
KitchenAbvGr					
1.025193					

Junk

GrLivArea	TotalSqftCalc	QualityIndex
2.273491	2.136077	1.130781

Using the variables from the variable selection above, I fit regression models on the training data to see which model produced the highest predictive accuracy. A summary table can be seen below of the four different models.

Model	Adj_R_Squared	AIC	BIC	MSE	RMSE	MAE
Forward	0.9038057	-2243.2640	-2166.349	0.0094445	0.0971826	0.0728835
Backward	0.9038057	-2243.2640	-2166.349	0.0094445	0.0971826	0.0728835
Stepwise	0.9038057	-2243.2640	-2166.349	0.0094445	0.0971826	0.0728835
Junk	0.7147137	-898.6364	-872.998	0.0282371	0.1680389	0.1293672

As expected, the Forward, Backward, and Stepwise model produced the same results since the same variables were selected. All three of these models outperformed the Junk model which was also expected. We can see that the models using the automated variable selection process resulted in an adjusted R-Squared of 0.9 which was impressive, while the junk model resulted in a 0.71 adjusted R-Squared. Therefore, the Forward, Backward, and Stepwise models are ranked higher than the junk model.

3) Next, I analyzed how the models performed on the unseen test data. Below is a table showing the Mean Squared Error (MSE) and the Mean Absolute Error (MAE) for the four models using the test sample data.

Model	Test_MSE	Test_MAE
Forward	0.0114076	0.0790053
Backward	0.0114076	0.0790053
Stepwise	0.0114076	0.0790053
Junk	0.0334064	0.1425474

Again, as expected the Forward, Backward, and Stepwise model all produced the same results. Based on the lower MSE and MAE these models fit better than the junk model. This ranking is the same as the ranking of the model performance on the in-sample predictions. In terms of MSE or MAE preference, the MSE metric penalizes larger errors more than smaller errors, so that can be more preferred in cases where being really wrong is not good. However, MAE can be also preferred since it is more interpretable when it comes to model evaluation. So, it can depend on context and use case of the model, so there is not one metric that is always preferred over the other. Therefore, it is important to look at both metrics when assessing model accuracy. We also need to compare the test MSE and MAE to the training MSE and MAE. If a model has better predictive accuracy on in-

sample data than it does on out-of-sample data than there is a risk that the model has been over-fitted to the in-sample training data and has difficulty generalizing to unseen data.

4) Next, I validated these models from a business perspective by checking to see what percentage of the model predictions accuracies met certain thresholds or cut off points. To do this I created a new variable called PredictionGrade which considers the predicted value to be 'Grade 1' if it is within ten percent of the actual value, 'Grade 2' if it is not Grade 1 but within fifteen percent of the actual value, Grade 3 if it is not Grade 2 but within twenty-five percent of the actual value, and 'Grade 4' otherwise. Below are the results for the distribution of prediction grades of each model for both the training data and the test data.

Prediction Grades:

```
forward_PredictionGrade
Grade 1: [0.0.10]
               1
forward_testPredictionGrade
Grade 1: [0.0.10]
               1

step_PredictionGrade
Grade 1: [0.0.10]
               1
step_testPredictionGrade
Grade 1: [0.0.10]
               1
```

```
backward_PredictionGrade
Grade 1: [0.0.10]
               1
backward_testPredictionGrade
Grade 1: [0.0.10]
               1

junk_PredictionGrade
Grade 1: [0.0.10]
               1
junk_testPredictionGrade
Grade 1: [0.0.10]
               1
```

We can see that for each model 100% of the predictions fell within 10% of the actual value which would indicate that all of the models are of "underwriting quality" as defined by the GSEs rate (accurate to within ten percent more than fifty percent of the time). However, these results are questionable as this degree of accuracy seems too good to be true, especially since the junk model performed just as well as the other models despite having lower predictive accuracy metrics. I suspect that performing a log transformation on the response variable has something to do wit this. As such, I would not place any meaningful value into the above results. Once a final model has been selected, I will back-transform the response variable and repeat this analysis to see if the distribution of PredictionGrades changes and re-evaluate the models from a business context and underwriting quality.

5) As previously mentioned, the Forward, Backward, and Stepwise models are the same model; therefore, they are all the "best" fitting model. I will use the variables selected by these models and refine the model further by removing certain variables that may be difficult to interpret, display collinearity, and/or little predictive value. Due to the large sample size, we run the risk of having too high of statistical power which can lead to an overfit model. Therefore, I will methodically remove variables from the model to reduce the number of variables to guard against overfitting.

   First, I removed BedroomsAbvGr since it is correlated to TotRmsAbvGrd. Then, I removed LotArea since the coefficient is near 0, so this variable has little impact on the

predicted value. I also dropped LowQualFinSF since this coefficient was also near 0 and has a small t-value. Next, I dropped KitchenAbvGr, since the coefficient is negative which logically does not make sense. Having more kitchens above ground theoretically should not decrease home value, so it would be difficult to interpret a negative coefficient. After removing these variables, I am left with TotalSqftCalc, BsmtUnfSF, HouseAge, QualityIndex, GarageArea, TotRmsAbvGrd, LotFrontage, Neighborhood_typeLow, Neighborhood_typeHigh as the predictor variables for logSalePrice. Fitting a regression model using these variables on the training data resulted in an R-Squared value of 0.9. A summary of the model coefficients for these variables can be seen below.

```
Call:
lm(formula = logSalePrice ~ TotalSqftCalc + BsmtUnfSF + HouseAge +
    QualityIndex + GarageArea + TotRmsAbvGrd + LotFrontage +
    Neighborhood_typeLow + Neighborhood_typeHigh, data = train.clean)

Residuals:
     Min       1Q   Median       3Q      Max
-0.63037 -0.05707  0.00015  0.05931  0.50908

Coefficients:
                       Estimate   Std. Error t value             Pr(>|t|)
(Intercept)          10.958448439  0.022894130 478.658 < 0.0000000000000002 ***
TotalSqftCalc         0.000270389  0.000007558  35.777 < 0.0000000000000002 ***
BsmtUnfSF             0.000168151  0.000009634  17.454 < 0.0000000000000002 ***
HouseAge             -0.003038195  0.000138641 -21.914 < 0.0000000000000002 ***
QualityIndex          0.009406950  0.000362841  25.926 < 0.0000000000000002 ***
GarageArea            0.000164509  0.000020232   8.131  0.00000000000000102 ***
TotRmsAbvGrd          0.016503154  0.003163750   5.216  0.00000021381465599 ***
LotFrontage           0.001151514  0.000179826   6.404  0.00000000021525295 ***
Neighborhood_typeLow -0.066581576  0.007799358  -8.537 < 0.0000000000000002 ***
Neighborhood_typeHigh 0.030381593  0.008819811   3.445              0.000591 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1007 on 1236 degrees of freedom
Multiple R-squared:  0.8986,    Adjusted R-squared:  0.8979
F-statistic:  1218 on 9 and 1236 DF,  p-value: < 0.00000000000000022
```

Next, I will go through and examine the impact of removing each variable from the final model, one at a time and compare the change in R-Squared to determine the predictive value of the variable when included in the model. Below is a table that shows the resulting R-Squared of the model as each variable is dropped until there is only the TotalSqftCalc variable left.

Dropped Variable	**New R-Squared**
TotRmsAbvGrd	0.8964
LotFrontage	0.8931
GarageArea	0.887
BsmtUnfSF	0.8335
HouseAge	0.7605
QualityIndex	0.7014
Neighborhood Type (dummy variables)	0.5686

We can see by removing TotRmsAbvGrd, LotFrontage, and GarageArea, the model’s R-squared is relatively unchanged, only decreasing by 0.01. However, when we remove BsmtUnfSF, HouseAge,

QualityIndex, and Neighborhood Type, there was a larger change in R-Squared. Therefore, for the final model I will remove TotRmsAbvGrd, LotFrontage, and GarageArea.

Since the dummy variables for the categorical variable Neighborhood Type is included in the final model, I then proceeded to check if adding interaction variables would improve the model fit. Evaluating the impact of interaction variables is important because this allows us to test for unequal slopes. First, I added interaction variables for Neighborhood Type and TotalSqftCalc and refitted the model. The R-Squared increased from 0.887 to 0.8886. Using a nested F-test, the F-value of 8.6 is higher than the critical F-value of 3.003 which suggests the interaction variables are statistically significant. However, this improvement is minimal; therefore, I decided to not include these interaction variables to help with model interpretability. Next, I refitted the model using interaction variables for Neighborhood Type and QualityIndex which resulted in no change in R-Squared and a nest F-test value of 0.13, so these were also not included in the model. Lastly, I refitted the model using interaction variables for Neighborhood Type and HouseAge the R-Squared increased minimally to 0.8906 and a had nested F-test value of 20.1. Since the R-Squared value of this full model is approximately the same as the reduced model (~0.89), I have chosen to exclude these interaction variables from the final model and opt for a simpler model to help with interpretability. The coefficient table and ANOVA table of the final model are below.

Final Model Coefficient Summary Table:

```
Call:
lm(formula = logSalePrice ~ TotalSqftCalc + BsmtUnfSF + HouseAge +
    QualityIndex + Neighborhood_typeLow + Neighborhood_typeHigh,
    data = train.clean)

Residuals:
     Min       1Q   Median       3Q      Max
-0.64588 -0.05773 -0.00090  0.06256  0.56071

Coefficients:
                          Estimate   Std. Error t value             Pr(>|t|)
(Intercept)           11.113427378  0.018786127 591.576 < 0.0000000000000002 ***
TotalSqftCalc          0.000317208  0.000006105  51.955 < 0.0000000000000002 ***
BsmtUnfSF              0.000212224  0.000008762  24.221 < 0.0000000000000002 ***
HouseAge              -0.003436949  0.000138627 -24.793 < 0.0000000000000002 ***
QualityIndex           0.009649076  0.000381065  25.321 < 0.0000000000000002 ***
Neighborhood_typeLow  -0.064633140  0.008082194  -7.997   0.0000000000000029 ***
Neighborhood_typeHigh  0.038336157  0.009205990   4.164   0.0000333982085672 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1062 on 1239 degrees of freedom
Multiple R-squared:  0.887,     Adjusted R-squared:  0.8865
F-statistic:  1621 on 6 and 1239 DF,  p-value: < 0.00000000000000022
```

Final Model ANOVA Table:

```
Analysis of Variance Table

Response: logSalePrice
                     Df Sum Sq Mean Sq  F value                Pr(>F)
TotalSqftCalc         1 70.296  70.296 6235.455 < 0.00000000000000022 ***
BsmtUnfSF             1 17.941  17.941 1591.466 < 0.00000000000000022 ***
HouseAge              1 11.408  11.408 1011.897 < 0.00000000000000022 ***
QualityIndex          1  8.748   8.748  775.944 < 0.00000000000000022 ***
Neighborhood_typeLow  1  1.069   1.069   94.812 < 0.00000000000000022 ***
Neighborhood_typeHigh 1  0.195   0.195   17.341            0.0000334 ***
Residuals          1239 13.968   0.011
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Final Model Equation:

Y_hat = 11.113427378 + 0.000317208*TotalSqftCalc + 0.000212224*BsmtUnfSF - 0.003436949*HouseAge + 0.009649076*QualityIndex - 0.064633140*Neighborhood_typeLow + 0.038336157*Neighborhood_typeHigh

The final model uses TotalSqftCalc, BsmtUnfSF, HouseAge, QualityIndex, and Neighborhood type dummy variables as the explanatory variables. These variables account for approximately 89% of the variability in the response variable logSalePrice. Since the response variable has been transformed this makes interpretation of this model challenging as the units have been converted to logSalePrice. However, there are some takeaways that we can generalize from this transformed version of the model. First, the use of dummy variables requires us to drop one of the dummy variables. In this case, the dummy variable associated with "Mid" Neighborhood Type has been removed. As such, this is the basis of interpretation of the model. That is if Neighborhood_typeLow and Neighborhood_typeHigh is 0 than the model defaults to predicting the value of a home in the "Mid" Neighborhood type. A negative coefficient for Neighborhood_typeLow means that it lowers the estimated value from the Mid type when the home is in a neighborhood classified as "Low". The positive coefficient for the Neighborhood_typeHigh means that the predicted value increases from the Mid type when the home is in a neighborhood classified as "High". Additionally, the negative coefficient for HouseAge tells us that the older the house is there is greater decrease in predicted value.

Hypothesis Testing and Underlying Assumptions:

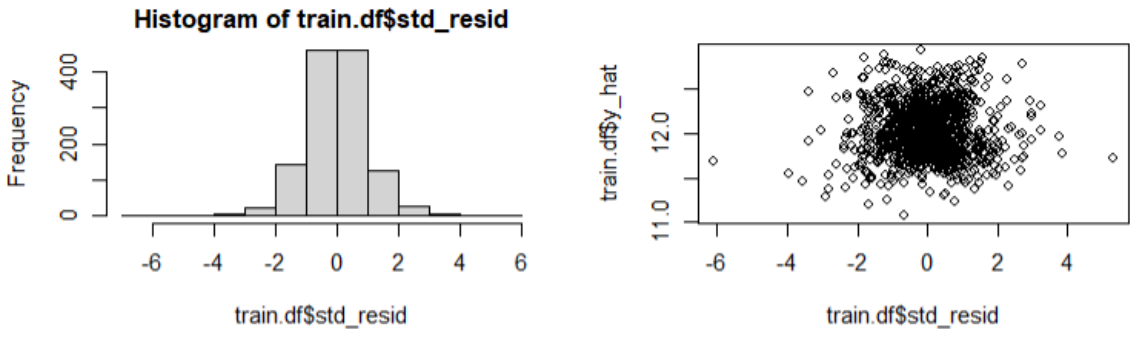
For hypothesis testing, I will perform an Overall Omnibus F-Test. I will use a critical of 2.1059 which is the value associated with degrees of freedom of 6 and 1239, at the 0.05 significance level.

Null hypothesis → H_0: $\beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = \beta_6 = 0$
Alternative hypotheses → H_A: at least one $\beta_i \neq 0$

From the Coefficient Summary Table, we can see that the F-statistic of 1621 is greater than the critical F-value of 2.1059. Therefore, we should reject the null hypothesis and conclude that at least one $\beta_i \neq 0$. This indicates that there is a significant relationship between the independent variables and the response variable.

Next, I checked to see if the underlying assumption of the hypothesis tests have been met by looking at the distribution of the standardized residuals which can be seen below in the histogram and scatterplot.

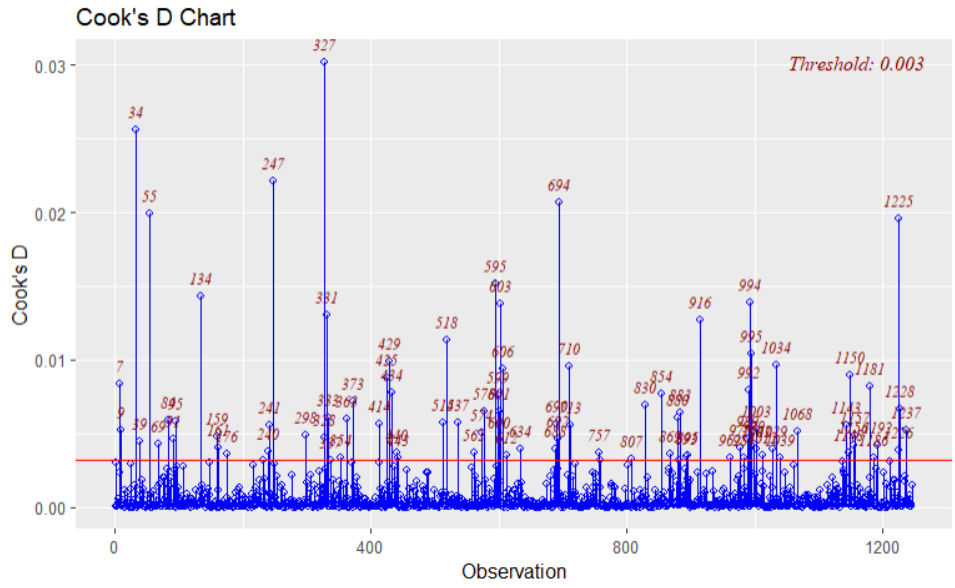
Histogram and Scatterplot of Standardized Residuals:



The histogram shows that the distribution of the standardized residuals is normally distributed. Additionally, the scatterplot of the residuals against the predicted values show a homoscedasticity pattern, as there are equal variances the residuals. Therefore, we can conclude that the underlying assumptions have been met and the hypothesis tests can provide meaningful information about the model's goodness of fit.

Influential Observations:

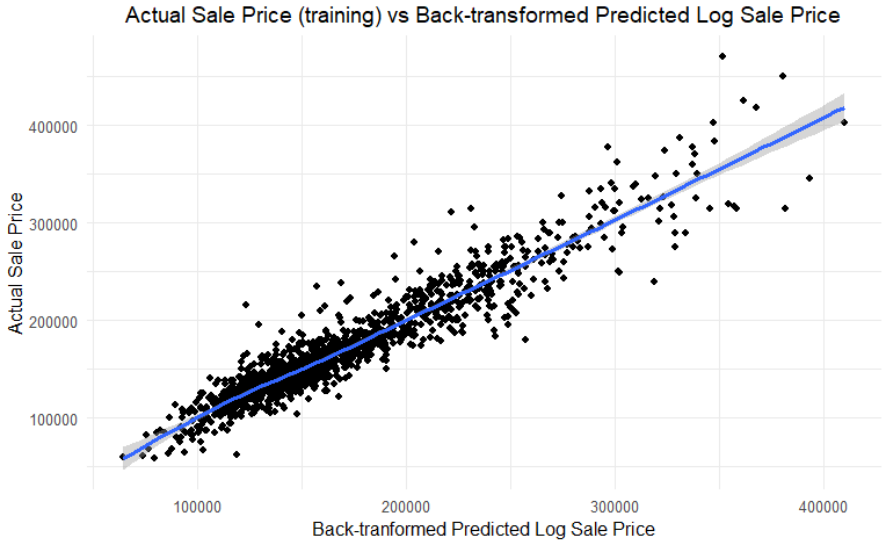
Next, I created the below Cook's Distance Chart to visualize the influential observations of the model. This chart shows that there are 97 potential outliers based on a Cook's Distance threshold of 0.003. We should be concerned that these observations may pull the regression model too far in one direction, thereby decreasing the overall fitness of the model. We could potentially remove these observations to increase the model performance and fit. However, these observations could also be valid representations of the sample population. So, further analysis should be done on these influential observations before removing them and refitting the model.

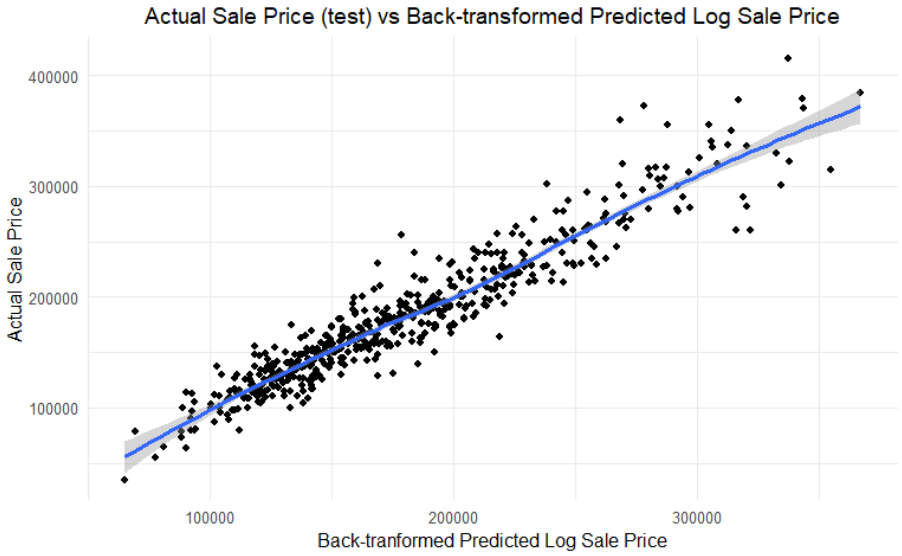


Back-transformation:

For better interpretation of the model predictions, I performed back-transformation of the response variable, logSalePrice, back to the original scale of SalePrice. Then I re-evaluated the model based on comparing the back-transformed predicted value to the actual SalePrice. We can see in the below scatterplots that the back-transformed predicted value tracks very closely to the actual SalePrice in a linear pattern in both the training and test sample data. This suggests that the model does a reasonably good job of predicting sale price when using the log transformation of SalePrice as the response variable for the regression model. Additionally, I recalculated the RMSE and MAE using the residual between actual SalePrice and the back-transformed predicted value. This resulted in a RMSE of $19,063 and MAE of $13,670 for the training data and a RMSE of $19,623 and MAE of $14,320 for the test data. This is much easier to interpret than the previous metrics using logSalePrice. We can interpret the MAE as that on average the back-transformed predicted log SalePrice is within $13,670 of the actual sale price in the training data and within $14,320 of the actual sale price in the test data.

Scatterplots of Actual SalePrice vs Back-transformed Predict logSalePrice:





Furthermore, as discussed earlier, the Prediction Grades results were dubious in my opinion as all the models produced prediction values within 10% of the actual value. While this is in theory possible, I believe that the log transformation was impacting the results. As such, I have repeated the Prediction Grades using the back-transformed values which yielded more feasible results. Since the Forward, Backwards, and Stepwise are essentially the same model, I only re-tested the Forward model. Additionally, I also did not re-test the junk model as there was no added value at this point, since I already made my final model selection. Thus, the prediction grade for the Forward model can be seen below using the back-transformed prediction values.

```
forward_PredictionGrade_bt
   Grade 1: [0.0.10] Grade 2: (0.10,0.15] Grade 3: (0.15,0.25]     Grade 4: (0.25+]
          0.71027287           0.15569823           0.10192616           0.03210273
forward_testPredictionGrade_bt
   Grade 1: [0.0.10] Grade 2: (0.10,0.15] Grade 3: (0.15,0.25]     Grade 4: (0.25+]
          0.69532710           0.16822430           0.10280374           0.03364486
```

We can see that for the training data, approximately 71% of the predictions fell within 10% of the actual value and for the test data, approximately 70% of the predictions fell within 10% of the actual. This indicates that the model is of underwriting quality. More importantly, these results appear to be more trustworthy and interpretable than the previous prediction grades using the log transformation of SalePrice.

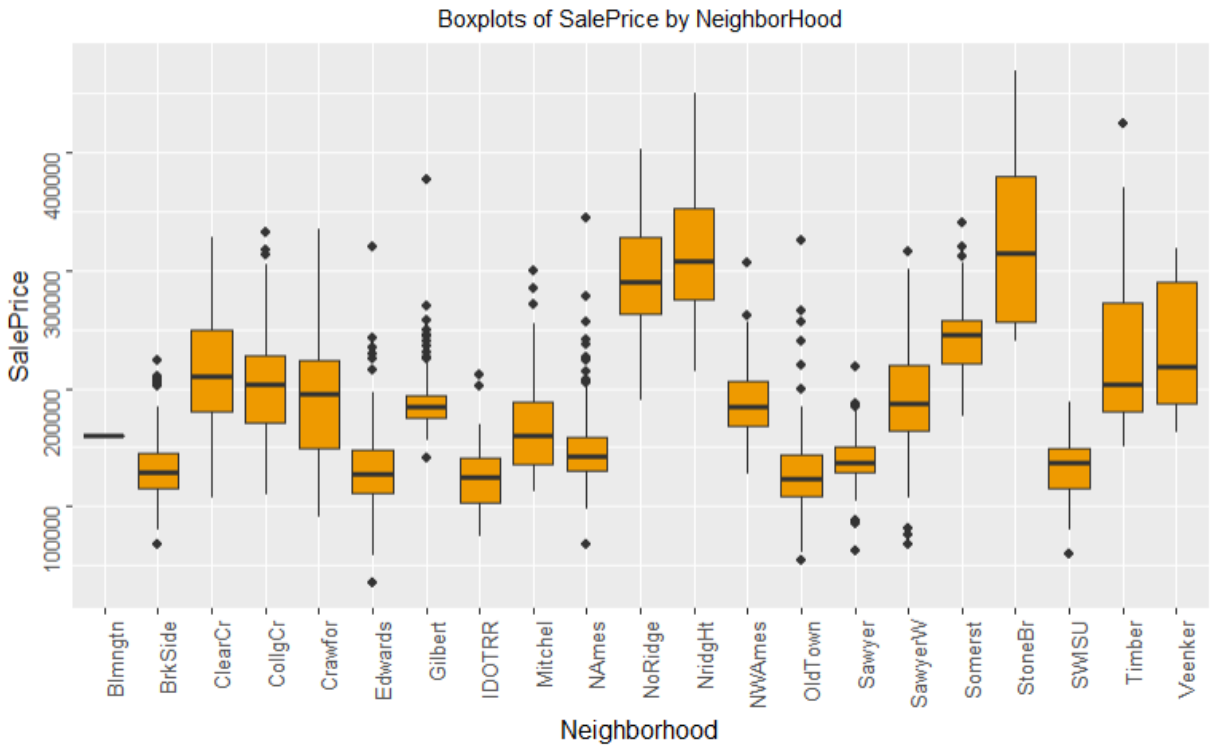
Conclusion/Reflections

From this analysis, I was able to build off of previous work and course learnings to develop a final multiple regression model that uses both continuous and categorical variables to explain approximately 89% of the response variable. From a business context, this model can predict sale prices of typical homes in Ames, IA between 2006 and 2010 within 10% of the actual sale price about 70% of the time. Overall, I am pretty impressed with the accuracy of the final model all things considered. However, arriving at this final model was no easy task and I have a greater understanding and appreciation of the regression modeling process.

In developing this final model, I learned how to assess categorical variables as potential predictor variables and perform automated feature selection using both the continuous variables and categorical variables via dummy encoded variables. I got practice in validating models using a train/test split cross-validation approach by training on the model on 70% of the sample data and testing on the remaining unseen 30% of the sample data. Furthermore, I learned how to further refine the feature selection process by testing for multicollinearity and also testing the predictive value of individual variables. I would previously rely on the automated variable selection process without much thought, but I have learned that large sample sizes may also come with too much statistical power. Therefore, it is important to thoroughly review the selected variables from automated feature selection procedures and remove any variables that are not useful for predicting the response variable or that may over-complicate the model. Often, a simpler model is preferred over a complicated max-fit model. Complicated models can sometimes overfit the data and therefore, make it difficult to generalize the model. Additionally, models with many variables can be difficult to interpret. Furthermore, I learned how to increase interpretability of the model by creating prediction grades. This can help the interpretation of the model from a business sense. It is not always practical to report technical statistical metrics in a business context, so using certain cut-off thresholds can be a more appropriate way to report the model results. Lastly, I think one of the biggest challenges of working with this dataset was that the response variable is not normally distributed which ultimately led to heteroscedasticity patterns in the residuals of the models that I did not perform a log transformation. As such, I used a log transformation of the SalePrice variable the response variable in my final model to improve my regression modeling performance. However, in doing so this decreases the interpretability of the model and the predicted values. For this reason, I found that performing a back-transformation on the predicted values to return these values to their original scale helped in interpreting the model as we can now compare predicted SalePrice to actual SalePrice which is more meaningful than comparing the log

transformed values. Another challenge to working with this dataset is the presence of outliers and influential observations. Keeping these observations in the data can skew the regression model and decrease the overall fit. However, removing these observations can be tricky as these could be valid outliers, so these need to be handled carefully. Removing too many of these influential observations could lead to over-fitting. However, at the same time if these outliers do not share similar characteristics of the defined “typical” home, then removing them might be appropriate and would result in improved prediction accuracy of the final model.

Appendix A: Side-by-Side Boxplot of SalePrice by Neighborhood



Appendix B: Summary Statistics of SalePrice by Neighborhood

Neighborhood	n	Mean	Std_Dev	Min	Q1	Median	Q3	Max
IDOTRR	41	125779.3	30346.99	75200	103000.0	125500.0	141000.0	212300
OldTown	149	128327.0	37750.16	55000	109000.0	123500.0	144000.0	325000
SWISU	29	132239.7	29716.99	60000	115000.0	136500.0	149000.0	189000
BrkSide	82	133755.8	30727.46	68500	115000.0	129250.0	144750.0	223500
Edwards	100	135375.0	40936.98	35000	110750.0	127750.0	148475.0	320000
Sawyer	113	139212.9	20916.20	62383	129000.0	137000.0	150000.0	219000
NAmes	342	147159.0	28216.50	68000	129900.0	142112.5	159000.0	345000
Blmngtn	1	159895.0	NA	159895	159895.0	159895.0	159895.0	159895
Mitchel	77	169438.3	42034.83	113000	136000.0	160000.0	188000.0	300000
NWAmes	106	189416.0	30806.91	127000	168175.0	184750.0	205750.0	306000
Gilbert	125	189879.5	28027.80	141000	175000.0	184100.0	194500.0	377500
SawyerW	78	190398.1	43827.70	67500	164750.0	187000.0	219500.0	316600
Crawfor	73	195125.6	53726.76	90350	149000.0	195000.0	224000.0	335000
CollgCr	211	200499.4	45692.27	110000	171500.0	203000.0	227500.0	332000
ClearCr	31	216854.8	49668.10	107500	181000.0	211000.0	250000.0	328000
Timber	44	229758.3	63496.66	150000	180000.0	204000.0	272803.2	425000
Veenker	13	231103.8	55795.60	162500	187000.0	218000.0	290000.0	318750
Somerst	65	244195.3	31733.79	176000	221500.0	245000.0	257500.0	340000
NoRidge	50	297309.8	45228.24	190000	263137.5	290000.0	328750.0	403000
NridgHt	44	312272.7	54012.61	214000	274975.0	307500.0	352500.0	450000
StoneBr	8	326540.8	83146.09	240000	256975.0	315000.0	379819.5	470000

Appendix C: Summary Table of Neighborhood Type

Neighborhood	Value type
IDOTRR	Low
OldTown	Low
SWISU	Low
BrkSid	Low
Edwards	Low
Sawyer	Low
NAmes	Low
Mitchel	Mid
Gilbert	Mid
NWAmes	Mid
SawyerW	Mid
Crawfor	Mid
CollgCr	Mid
ClearCr	Mid
Timber	High
Veenker	High
Somerst	High
NoRidge	High
NridgHt	High
StoneBr	High

Appendix D: Summary Statistics of SalePrice by Neighborhood Type

Neighborhood_type	n	Mean	Std_Dev	Min	Q1	Median	Q3	Max
Low	774	139159.6	32338.90	35000	121125	136950	154300	345000
Mid	701	192557.6	42679.58	67500	165000	188000	217500	377500
High	306	232588.9	80881.90	68500	165625	240000	286375	470000