# Automated Variable Selection, Multicollinearity, and Predictive Modeling

Serra Uzun, MSDS_410 FALL 2020
11/01/2020

**Introduction**

Automated variable selection regression models are an effective and efficient way to predict response variables within a dataset with numerous predictor variables in the most accurate manner. We will be conducting Forward, Backward, and Stepwise Variable Selection models on the Ames Iowa Housing dataset, analyzing, comparing, and assessing the outputs on these models' performance and accuracy and then conducting model validation. Our study begins by cleaning and modifying the dataset to prep for train/test split for further modeling, followed by various model assessments. We will choose the Automated Selection Model that is most likely to perform best with the Ames dataset.

## 1. Sample Definition and Data Split

### 1.1 Sample Definition

The drop conditions we have created to obtain our sample group were related to building type (BldgType), sale condition (SaleCondition), street condition (Street), year the house was built (YearBuilt), total basement square footage (TotalBsmtSF), ground floor living area square footage (GrLivArea) and utilities (Utilities). With the aim of creating a sample dataset, we set conditions for these variables as below:

- BldgType: Must be Single Family Residence (SFR)
- SaleCondition: Must have Normal Sale Condition
- Street: Must be located on a paved street
- YearBuilt: Must be built after the year 1950
- TotalBsmtSF: Must have a basement
- GrLivArea: Must have a ground floor living area greater than 800 square feet
- Utilities: Must have all utilities included

Per these droplist conditions, the raw dataset with 2,930 observations has dropped to 1,469 observations. The conditions that resulted in the drop of over 90% of the observations from raw dataset to sample dataset were Building Type, Sale Condition, and Year Built. The details are shown in the droplist counts waterfall table below:

| | | |
|---|---|---|
| | Ames Housing Raw Dataset | 2,930 |
| Variable | Drop Condition | |
| BldgType | Not SFR | -505 |
| SaleCondition | Non-Normal Sale | -423 |
| Street | Not Paved | -6 |
| YearBuilt | Prior to 1950 | -489 |
| TotalBsmtSF | No Basement | -28 |
| GrLivArea | Less Than 800 SF | -9 |
| Utilities | No Public Utilities | -1 |
| | Total Eligible Samples | 1,469 |

**Table 1: Waterfall Table of Droplist Counts**

### 1.2 The Train/Test Split

Prior to beginning our predictive model, we will perform a 70/30 split on our dataset that only consists of eligible 1,469 observations. This split aims to 'train' our model with 70 percent of the sample data and then 'test' it with the test dataset, which is 30 percent of the sample data. With the 70/30 split, we continue our model with a train dataset of 1,037 observations and a test dataset of 432 observations. While we will mainly use the train data for our models, we will be tapping back to the test data for cross-validation of our models in the following sections.

## 2. Model Identification and In-Sample Model Fit

A mix of continuous, discrete, and categorical (converted to factor) variables that are most likely to be the optimal predictors of our response variable, SalePrice, were selected for further examination. Overall, these variables are relevant to lot/land size and shape, square footages of different house areas, and counts of rooms (bathrooms and bedrooms). The list of the selected predictor variables is as below:

- Lot Area
- Lot Configuration
- Land Slope
- Overall Quality
- Overall Condition
- Year Built
- Total Basement SF
- First Floor SF
- Second Floor SF
- Ground Floor Living Room Area
- Full Baths
- Half Baths
- Bedrooms Above Ground
- Total Rooms Above Ground
- Fireplaces
- Car Spaces in the Garage
- Garage Area
- Wood Desk SF
- Open Porch SF
- Total SF Calc

The correlation plot presented below uses the selected predictor variables above and helps us to detect multicollinearity and the absence of correlation of any of the variables. The correlation plot shows us that none of the predictor variables have alarmingly high correlations with other variables. In general, we have a well variety of correlations between our selected pool of predictor variables. While lot and outdoor space-related variables present to be the variables

with the least overall correlation with other predictor variables, and Overall Condition is the only variable with negative correlation throughout. Per the correlation plot presented in Figure 1, we are not expecting to see high VIF (Variance Inflation Factor) on any of our selected variables.

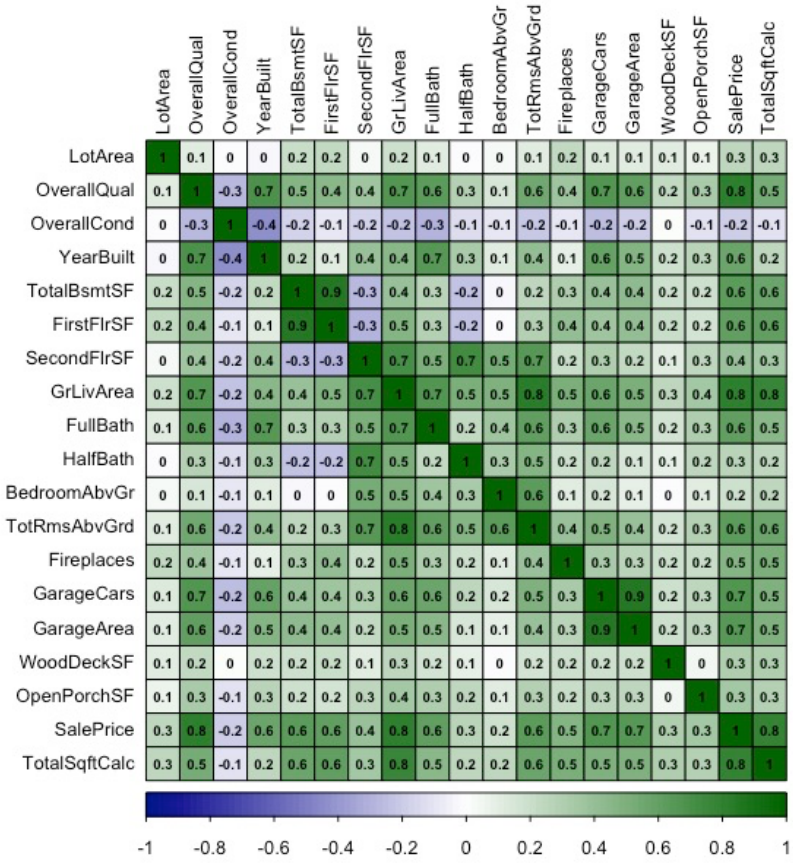


**Figure 1: Correlation Plot of Selected Predictor Variables**

Moving forward, we will be conducting our models with the 20 predictor variables listed above and exclude all other variables in the Ames Dataset. Prior to conducting the forward, backward, and stepwise linear regression models, we set lower and upper limits to be used in these models. The lower limit is set to be 1 and the upper limit to include all variables in our selected predictor variable pool. Also, for the stepwise model, we set the initialization point to be SalePrice ~ TotalSqftCalc.

### 2.1 Forward Variable Selection

The forward variables selection model (forward.lm) begins with an empty model and adds variables from our selected pool of predictor variables one by one. We obtained the forward.lm results through Ames dataset selected predictor and response variable present adjusted R-squared of 0.89 and p-value below alpha level, 0.05. The 0.89 adjusted R-squared shows that almost 90% of the data variance can be explained through the forward.lm. LandSlopeMod, LandSlopeSev and SecondFlrSF are the only variables that have p-value above 0.05. (Please see Appendix for a detailed model summary)

The Mean Absolute Error (MAE) we obtained for our forward variable selection model is 17055.66, which suggests that there is a 17055.66 difference between the original and predicted values extracted by averaged the absolute difference over the train dataset.

| | Df | Sum of Sq | RSS | AIC |
|---|---|---|---|---|
| | | | 6.11E+11 | 20975 |
| SecondFlrSF | 1 | 1.18E+09 | 6.12E+11 | 20975 |
| LandSlope | 2 | 2.77E+09 | 6.14E+11 | 20976 |
| TotRmsAbvGrd | 1 | 2.38E+09 | 6.13E+11 | 20977 |
| WoodDeckSF | 1 | 3.03E+09 | 6.14E+11 | 20978 |
| HalfBath | 1 | 4.61E+09 | 6.16E+11 | 20981 |
| FullBath | 1 | 9.06E+09 | 6.20E+11 | 20989 |
| BedroomAbvGr | 1 | 1.86E+10 | 6.30E+11 | 21004 |
| LotArea | 1 | 2.70E+10 | 6.38E+11 | 21018 |
| GarageArea | 1 | 2.81E+10 | 6.39E+11 | 21020 |
| OverallCond | 1 | 3.05E+10 | 6.41E+11 | 21024 |
| TotalBsmtSF | 1 | 3.29E+10 | 6.44E+11 | 21028 |
| GrLivArea | 1 | 4.33E+10 | 6.54E+11 | 21044 |
| YearBuilt | 1 | 4.95E+10 | 6.61E+11 | 21054 |
| TotalSqftCalc | 1 | 7.63E+10 | 6.87E+11 | 21096 |
| OverallQual | 1 | 1.16E+11 | 7.27E+11 | 21153 |

**Table 2: Forward Variable Selection Model AIC Values**

Upon reviewing Table 2, we see that the starting AIC of the forward.lm is 20975.41, which then follows the iterations as the model adds variables to become 21153. SecondFlrSF and LandSlope are the variables with the lowest AIC value, which indicates that they are the variables that would cause the least amount of information loss if we were to remove them from the model. This result is parallel with our forward.lm summary, where SecondFlrSF and LandSlope both had high p-values. On the other hand, Overall Quality has the highest AIC value compared to all variables included in the model, with 178 additional AIC values above the starting AIC. Per the forward.lm AIC results, we also see that the model has chosen the most significant predictor variables and not included some other variables, such as GarageCars and LotConfig, that we had in our selected pool of predictor variables.

### 2.2 Backward Variable Selection

Opposite to the Forward Variable Selection approach, the Backward Variable Selection model starts off the model by having all variables included and then removes it as it moves forward. This approach allows us to see the effect of eliminating a variable. Another difference between the forward and backward variable selection models in our case is that backward.lm starts with including all variables in our selected pool of predictor variables, whereas forward.lm add variables as it goes along the model and not all variables as they have very little to no impact on the outcome.

The summary of the backward.lm shows that the adjusted R-squared is 0.89, and the model p-value is below 0.05, while there are multiple variables in the model that have an individual p-value of above 0.05 alpha level. Even though we have more variables with higher p-values, we see only a 0.01 difference in adjusted R-squared, which suggests that backward.lm can explain 89% of the same dataset variance.lm adjusted R-squared. (Please see Appendix for a detailed model summary)

The Mean Absolute Error we obtained for the backward.lm is 16996.63, suggesting that the averaged absolute difference between the original and predicted mean is 16,996.63.

| | Df | Sum of Sq | RSS | AIC |
|---|---|---|---|---|
| | | | 6.11E+11 | 20976 |
| GarageCars | 1 | 1.29E+09 | 6.06E+11 | 20976 |
| LotConfig | 4 | 4.82E+09 | 6.10E+11 | 20976 |
| SecondFlrSF | 1 | 1.37E+09 | 6.07E+11 | 20976 |
| LandSlope | 2 | 2.68E+09 | 6.08E+11 | 20976 |
| TotRmsAbvGrd | 1 | 2.03E+09 | 6.07E+11 | 20977 |
| WoodDeckSF | 1 | 3.20E+09 | 6.08E+11 | 20979 |
| HalfBath | 1 | 5.21E+09 | 6.10E+11 | 20982 |
| GarageArea | 1 | 7.04E+09 | 6.12E+11 | 20986 |
| FullBath | 1 | 9.98E+09 | 6.15E+11 | 20990 |
| BedroomAbvGr | 1 | 1.80E+10 | 6.23E+11 | 21004 |
| LotArea | 1 | 2.96E+10 | 6.35E+11 | 21023 |
| OverallCond | 1 | 3.15E+10 | 6.37E+11 | 21026 |
| TotalBsmtSF | 1 | 3.23E+10 | 6.37E+11 | 21028 |
| GrLivArea | 1 | 4.19E+10 | 6.47E+11 | 21043 |
| YearBuilt | 1 | 4.88E+10 | 6.54E+11 | 21054 |
| TotalSqftCalc | 1 | 7.78E+10 | 6.83E+11 | 21099 |
| OverallQual | 1 | 1.12E+11 | 7.18E+11 | 21150 |

**Table 3: Backward Variable Selection Model AIC Values**

The backward.lm variable AIC values show that GarageCars, LotConfig, SeconfFlrSF, and LandSlope are the variables with the lowest AIC value of 20976. In addition to SeconfFlrSF and LandSlope, which were the variables with the least AIC values, we see that GarageCars and LotConfig are also shown the same outcome as these variables and a slightly higher start AIC. The variables with the highest AIC values, which indicate that in case of elimination of these variables, the amount of information loss would be maximum compared to other variables, are Overall Quality, Total SF Calc, and Year Built. These are the same as the AIC results we obtained through the forward.lm stepAIC.

### 2.3 Stepwise Variable Selection

The Stepwise Variable Selection combined both Forward Variable Selection and Backward Variable Selection that we modeled in Sections 2.1 and 2.2, respectively. The Stepwise model will add and remove variables simultaneously to optimize the model and its results. The linear model summary of the stepwise shows us that the stepwise model with both forward and backward approach have chosen the same variables as the forward.lm did. The Adjusted R-squared is the same as both previous models, which is 0.89, and the p-value of the model is less than 0.05. These, like previous models, indicate statistical significance and 89% explanation of the variance in the dataset through the stepwise model. In addition to these results, the stepwise model's MAE is identical with the forward.lm, which is 17055.66.

The stepwise.lm takes start-to-finish 15 steps to add and remove variables, decreasing each forward's AIC value. When we run the stepwise.lm in the stepAIC() function, the results obtained show the same output as the forward.lm model.

| | Df | Sum of Sq | RSS | AIC |
|---|---|---|---|---|
| | | | 6.11E+11 | 20975 |
| SecondFlrSF | 1 | 1.18E+09 | 6.12E+11 | 20975 |
| LandSlope | 2 | 2.77E+09 | 6.14E+11 | 20976 |
| TotRmsAbvGrd | 1 | 2.38E+09 | 6.13E+11 | 20977 |
| WoodDeckSF | 1 | 3.03E+09 | 6.14E+11 | 20978 |
| HalfBath | 1 | 4.61E+09 | 6.16E+11 | 20981 |
| FullBath | 1 | 9.06E+09 | 6.20E+11 | 20989 |
| BedroomAbvGr | 1 | 1.86E+10 | 6.30E+11 | 21004 |
| LotArea | 1 | 2.70E+10 | 6.38E+11 | 21018 |
| GarageArea | 1 | 2.81E+10 | 6.39E+11 | 21020 |
| OverallCond | 1 | 3.05E+10 | 6.41E+11 | 21024 |
| TotalBsmtSF | 1 | 3.29E+10 | 6.44E+11 | 21028 |
| GrLivArea | 1 | 4.33E+10 | 6.54E+11 | 21044 |
| YearBuilt | 1 | 4.95E+10 | 6.61E+11 | 21054 |
| TotalSqftCalc | 1 | 7.63E+10 | 6.87E+11 | 21096 |
| OverallQual | 1 | 1.16E+11 | 7.27E+11 | 21153 |

**Table 4: Stepwise Variable Selection Model AIC Values**

The stepwise.lm variables and AIC values of the stepwise.lm are the same as the forward.lm output. This suggests that even when the stepwise model has the option to pick whether or not to add and remove variables, the steps it has taken gave the same output as the forward variable selection model.

### 2.4 Model Comparison

Firstly, we will look at each model's VIF scores to see if there was an occurrence of multicollinearity with any of the variables that adding and removing predictors have caused in any of the models described in Section 2.1 2.2 and 2.3.

| | Forward Variable Selection VIF | Backward Variable Selection VIF | Stepwise Variable Selection VIF |
|---|---|---|---|
| OverallQual | 3.452 | 3.493 | 3.452 |
| TotalSqftCalc | 3.604 | 3.635 | 3.604 |
| GarageArea | 1.831 | 4.178 | 1.831 |
| GrLivArea | 11.690 | 11.747 | 11.690 |
| TotalBsmtSF | 4.330 | 4.358 | 4.330 |
| BedroomAbvGr | 2.085 | 2.097 | 2.085 |
| YearBuilt | 3.699 | 3.800 | 3.699 |
| OverallCond | 1.278 | 1.290 | 1.278 |
| LotArea | 1.413 | 1.495 | 1.413 |
| FullBath | 3.144 | 3.229 | 3.144 |
| HalfBath | 2.569 | 2.611 | 2.569 |
| WoodDeckSF | 1.203 | 1.209 | 1.203 |
| TotRmsAbvGrd | 3.887 | 3.931 | 3.887 |
| LandSlope | 1.316 | 1.336 | 1.316 |
| SecondFlrSF | 7.817 | 7.898 | 7.817 |
| LotConfig | | 1.167 | |
| GarageCars | | 4.967 | |

**Table 5: VIF Scores of All Variables in All Models**

The VIF scores obtained for all variables used in all previous models do not present any concerning multicollinearity. The variable that has the highest VIF score amongst all models is Ground Floor Living Room Area (GrdLivArea), followed by Second Floor SF (SecondFlrSF) and Total Basement SF (TotalBsmtSF). While the VIF score of these variables is higher than the selected pool's remaining predictor variables, they are not at a concerning level that would indicate unsatisfactory model performance. In addition to VIF, the below table presents the Adjusted R-squared, AIC, BIC, Mean Squared Error (MSE), and Mean Absolute Error (MAE) amongst all models for comparison.

| | FORWARD VARIABLE SELECTION | BACKWARD VARIABLE SELECTION | STEPWISE VARIABLE SELECTION |
|---|---|---|---|
| ADJUSTED R-SQUARED | 0.89 | 0.89 | 0.89 |
| AIC | 23920.29 | 23920.35 | 23920.29 |
| BIC | 24009.29 | 24034.06 | 24009.29 |
| ROOT OF MEAN SQUARE ERROR (RMSE) | 24272.84 | 24156.74 | 24272.84 |
| MEAN ABSOLUTE ERROR (MAE) | 17055.66 | 16996.63 | 17055.66 |

**Table 6: Model Comparison**

Upon reviewing the results from all models presented in Table 6, we can see that overall all three models present quite similar outputs. Forward Variable Selection and Stepwise Variable Selection models have identical results throughout all these parameters presented above.

All three models have an Adjusted R-squared of 0.89, which as mentioned earlier, indicates that 89% of the variance in the data can be explained through the model. When we look at the AIC and BIC values, we see that the Backward Variable Selection model has higher AIC and BIC than the Forward and Stepwise Selection Models. This suggests that Forward and Stepwise models are better and more likely to be the true model in comparison to Backward Selection. As we review the MSE and MAE, we can see that the Backward Variable Selection model has slightly lower MSE and MAE than both Forward and Stepwise Selection.

The MSE and MAE results and AIC and BIC values show a contradicting output. Ideally, we would like to select a variable selection model with the lowest AIC and BIC, as well as the lowest MSE and MAE. Yet none of the models clearly give us this output, therefore through the results presented in Table 6, we prioritize the AIC and BIC results over RMSE and MAE output and select Forward Variable Selection (or Stepwise Variable Selection as they have the identical results) model as the optimal option. The reason for this selection is (a) AIC and BIC indicate a truer model, which we will assess this theory through the test data, and (b) the difference between the Forward Selection Model and Backward Selection Model RMSE and MAE is not as significant to cause concerns. So, while the Backward Variable Selection model has fewer errors overall, the Forward Variable Selection model and Stepwise Variable Selection Model are the ones likely to best predict the future values.

### 3. Predictive Accuracy

To assess the accuracy of our Forward, Backward, and Stepwise Variable Selection models the test dataset, which is the 30% of the original dataset that we initially split, and evaluate the RMSE and MAE result to understand how our models perform with a set of data that it hasn't seen before.

| | Train Data | | | Test Data | | |
|---|---|---|---|---|---|---|
| | Forward.lm | Backward.lm | Stepwise.lm | Forward.lm | Backward.lm | Stepwise.lm |
| Root Mean Square Error (RMSE) | 24272.84 | 24156.74 | 24272.84 | 24428.81 | 24735.95 | 24428.81 |
| Mean Absolute Error (MAE) | 17055.66 | 16996.63 | 17055.66 | 16582.5 | 16740.54 | 16582.5 |

**Table 7: RMSE and MAE Comparison of All Models with Train and Test Data**

By the results presented in Table 7, we can observe that from the train dataset to the test dataset RMSE has increased while MAE has decreased. This pattern suggests that the models that ran with test data have more spread out residuals, meaning the models is decreased performance regarding fit than it was with the train data. On the other hand, the average of differences observed between the actual and predicted values, MAE, had decreased when models were conducted using the test data instead of train data.

Through Table 7, we are clearly able to see that the best forming model with train data, backward variable selection, is not the best performer when conducted with test data. Forward and Stepwise Selection models, when modeled with test data, are lower in RMSE and MAE results

with a more significant difference than the Backward Selection model was when modeled with train data in Section 2. So as Forward and Stepwise Models perform better with test data, they are better performing models and are more likely to give a model with a better fit and better predictions. Therefore, we are able to remain consistent with our model selection and conclusion in Section 2.4. and maintain our stand by choosing the Forward (or Stepwise) model as the best model.

## 4. Operation Validation

The operation validation is a crucial part of the analysis where the model is assessed through the business application. In the case of the Ames dataset, we form four prediction grades that are Grade 1 if the prediction is within 10% of actual value, Grade 2 if it within 10 – 15% of actual value, Grade 3 if it within 15 – 25% of actual value and Grade 4 otherwise.

| | Train Data | | | Test Data | | |
|---|---|---|---|---|---|---|
| | Forward.lm | Backward.lm | Stepwise.lm | Forward.lm | Backward.lm | Stepwise.lm |
| Grade 1 (0 - 10%) | 0.65 | 0.66 | 0.65 | 0.67 | 0.68 | 0.67 |
| Grade 2 (10-15%) | 0.17 | 0.16 | 0.17 | 0.16 | 0.15 | 0.16 |
| Grade 3 (15-25%) | 0.14 | 0.14 | 0.14 | 0.14 | 0.14 | 0.14 |
| Grade 4 (25% +) | 0.04 | 0.04 | 0.04 | 0.03 | 0.03 | 0.03 |

**Table 8: Distribution of Train and Test Data under each Prediction Grade for All Models**

Per the results presented in Table 8, all Variable Selection Models show they can predict 65 - 68 % of the predicted values within the 10% of the actual value, followed by 15 - 17% within the 10 - 15%, 14% within 15 – 25% and 3 – 4% pass 25% of the actual value. Like the prior results we obtained, the output for each model's prediction accuracy is considerably close to each other, while Backward Variable Selection Model is the model with the highest percentage of Grade 1 predictions. The variance between models regarding the distribution of predictive accuracy is minimal and could be considered to show parallel output with the AIC, BIC, RMSE, and MAE values. The model ranking remained the same, yet our model selection was not affected by the operation validation. Therefore, we can conclude that our model selection of Forward and Stepwise Variable Selection Models are valid, show consistent and accurate results on their performance and reliability throughout our study.

## APPENDIX

**Forward LM**

| | *Dependent variable:* |
|---|---|
| | SalePrice |
| | Forward LM |
| Constant | -1,489,929.00*** |
| | (151,180.00) |
| OverallQual | 15,530.27*** |
| | (1,117.18) |
| TotalSqftCalc | 23.50*** |
| | (2.08) |
| GarageArea | 39.28*** |
| | (5.73) |
| GrLivArea | 45.71*** |
| | (5.37) |
| TotalBsmtSF | 31.61*** |
| | (4.26) |
| BedroomAbvGr | -9,384.66*** |
| | (1,684.75) |
| YearBuilt | 702.49*** |
| | (77.24) |
| OverallCond | 6,481.45*** |
| | (908.04) |
| LotArea | 1.07*** |
| | (0.16) |
| FullBath | -9,792.31*** |
| | (2,518.26) |
| HalfBath | -6,751.67*** |
| | (2,434.58) |
| WoodDeckSF | 13.72** |
| | (6.10) |
| TotRmsAbvGrd | 2,205.65** |
| | (1,105.88) |
| LandSlopeMod | 5,805.50 |
| | (3,840.14) |
| LandSlopeSev | -16,639.51 |
| | (12,165.85) |
| SecondFlrSF | 6.69 |
| | (4.76) |
| Observations | 1,037 |
| $R^2$ | 0.89 |
| Adjusted $R^2$ | 0.89 |
| Residual Std. Error | 24,474.28 (df = 1020) |
| F Statistic | 514.33*** (df = 16; 1020) |
| *Note:* | *p<0.1; **p<0.05; ***p<0.01 |

**Backward LM**

| | *Dependent variable:* |
|---|---|
| | SalePrice |
| | Backward LM |
| Constant | -1,498,638.00*** |
| | (152,937.70) |
| LotArea | 1.15*** |
| | (0.16) |
| LotConfigCulDSac | -3,971.71 |
| | (3,107.80) |
| LotConfigFR2 | -7,953.75 |
| | (4,886.09) |
| LotConfigFR3 | -20,502.66* |
| | (11,179.82) |
| LotConfigInside | 63.64 |
| | (2,093.18) |
| LandSlopeMod | 5,000.05 |
| | (3,850.76) |
| LandSlopeSev | -18,559.74 |
| | (12,171.09) |
| OverallQual | 15,396.20*** |
| | (1,121.13) |
| OverallCond | 6,620.61*** |
| | (910.13) |
| YearBuilt | 706.67*** |
| | (78.10) |
| TotalBsmtSF | 31.43*** |
| | (4.27) |
| SecondFlrSF | 7.23 |
| | (4.78) |
| GrLivArea | 45.08*** |
| | (5.37) |
| FullBath | -10,418.21*** |
| | (2,546.42) |
| HalfBath | -7,237.05*** |
| | (2,448.94) |
| BedroomAbvGr | -9,257.82*** |
| | (1,685.43) |
| TotRmsAbvGrd | 2,046.37* |
| | (1,109.52) |
| GarageCars | 3,874.66 |
| | (2,636.94) |
| GarageArea | 29.69*** |
| | (8.64) |
| WoodDeckSF | 14.13** |
| | (6.10) |
| TotalSqftCalc | 23.83*** |
| | (2.09) |
| Observations | 1,037 |
| $R^2$ | 0.89 |
| Adjusted $R^2$ | 0.89 |
| Residual Std. Error | 24,417.13 (df = 1015) |
| F Statistic | 394.18*** (df = 21; 1015) |
| *Note:* | *p<0.1; **p<0.05; ***p<0.01 |

**Stepwise LM**

| | *Dependent variable:* |
|---|---|
| | SalePrice |
| | Steowise LM |
| Constant | -1,489,929.00*** |
| | (151,180.00) |
| TotalSqftCalc | 23.50*** |
| | (2.08) |
| OverallQual | 15,530.27*** |
| | (1,117.18) |
| GarageArea | 39.28*** |
| | (5.73) |
| GrLivArea | 45.71*** |
| | (5.37) |
| TotalBsmtSF | 31.61*** |
| | (4.26) |
| BedroomAbvGr | -9,384.66*** |
| | (1,684.75) |
| YearBuilt | 702.49*** |
| | (77.24) |
| OverallCond | 6,481.45*** |
| | (908.04) |
| LotArea | 1.07*** |
| | (0.16) |
| FullBath | -9,792.31*** |
| | (2,518.26) |
| HalfBath | -6,751.67*** |
| | (2,434.58) |
| WoodDeckSF | 13.72** |
| | (6.10) |
| TotRmsAbvGrd | 2,205.65** |
| | (1,105.88) |
| LandSlopeMod | 5,805.50 |
| | (3,840.14) |
| LandSlopeSev | -16,639.51 |
| | (12,165.85) |
| SecondFlrSF | 6.69 |
| | (4.76) |
| Observations | 1,037 |
| $R^2$ | 0.89 |
| Adjusted $R^2$ | 0.89 |
| Residual Std. Error | 24,474.28 (df = 1020) |
| F Statistic | 514.33*** (df = 16; 1020) |
| *Note:* | *p<0.1; **p<0.05; ***p<0.01 |

## The R code for Assignment #7

```
# Serra Uzun
# MSDS_410 Supervised Learning Methods_FALL 2020
# 11.01.2020
# Assignment_07

getwd()
setwd("/Users/serrauzun/Desktop/MSDS_410_Supervised/Assignment #7")

ames.df <- readRDS("Ames_eligible_sample.Rdata")
dim(ames.df)
colnames(ames.df)
str(ames.df)

unique(ames.df$LotConfig)

ames.df$LandSlope <- as.factor(ames.df$LandSlope)
ames.df$LotConfig <- as.factor(ames.df$LotConfig)
colnames(ames.df)
str(ames.df)
sapply(ames.df, function(x) sum(is.na(x)))

# Set the seed on the random number generator so you get the same split every time that # you run the
code.
set.seed(123)
ames.df$u <- runif(n=dim(ames.df)[1],min=0,max=1)
# Define these two variables for later use;
ames.df$QualityIndex <- ames.df$OverallQual * ames.df$OverallCond
ames.df$TotalSqftCalc <- ames.df$BsmtFinSF1 + ames.df$BsmtFinSF2 + ames.df$GrLivArea
# Create train/test split;
train.df <- subset(ames.df, u<0.70)
test.df <- subset(ames.df, u>=0.70)
# Check your data split. The sum of the parts should equal the whole. # Do your totals add up?
dim(ames.df)[1]
dim(train.df)[1]
dim(test.df)[1]
dim(train.df)[1] + dim(test.df)[1]

colnames(ames.df)

##
ames.df.small <- ames.df[,c(6, 12:13, 19:21, 40, 45:46, 48, 51:53, 56, 58, 63:64, 68:69, 82,86)]
colnames(ames.df.small)
var_pool <- as.data.frame(colnames(ames.df.small))
str(ames.df.small)

ames.small_cor <- cor(ames.df.small[sapply(ames.df.small, is.numeric)])
```

```
ames.small_cor

par(mfrow=c(1,1))
corrplot(ames.small_cor, method = "color", outline = T, addrect = 4, rect.col = "black", rect.lwd =
3,cl.pos = "b", tl.col = "black", tl.cex = 0.8, cl.cex = 0.8, addCoef.col = "black", number.digits = 1,
number.cex = 0.60, col = colorRampPalette(c("darkblue","white","darkgreen"))(100))

train.clean <-train.df[,(names(ames.df.small))]
colnames(train.clean)
dim(train.clean )

sapply(train.clean, function(x) sum(is.na(x)))

# Define the upper model as the FULL model
upper.lm <- lm(SalePrice ~ ., data=train.clean)
summary(upper.lm)
# Define the lower model as the Intercept model
lower.lm <- lm(SalePrice ~ 1, data=train.clean)
# Need a SLR to initialize stepwise selection
sqft.lm <- lm(SalePrice ~ TotalSqftCalc, data=train.clean);
summary(sqft.lm)


# Note: There is only one function for classical model selection in R - stepAIC();
# stepAIC() is part of the MASS library.
# The MASS library comes with the BASE R distribution, but you still need to load it;
library(stargazer)
library(MASS)
install.packages("Metrics")
library(Metrics)
# Call stepAIC() for variable selection
forward.lm <- stepAIC(object=lower.lm,scope=list(upper=formula(upper.lm),lower=~1),
                      direction=c('forward'))
summary(forward.lm)
mse.fwd <- mse(train.clean$SalePrice, predict(forward.lm, train.clean))
mae.fwd <- mae(predict(forward.lm), train.clean$SalePrice)

sqrt(mse.fwd)
mae.fwd

stepAIC(forward.lm)
out.path = "/Users/serrauzun/Desktop/MSDS_410_Supervised/Assignment #7"
frwd_lm <- 'Forward.html';
stargazer(forward.lm, type=c('html'),out=paste(out.path,frwd_lm,sep=''),
          title=c('Forward LM'),
          align=TRUE, digits=2, digits.extra=2, initial.zero=TRUE,
          column.labels=c('Forward LM'), intercept.bottom=FALSE)
```

```
backward.lm <- stepAIC(object=upper.lm,direction=c('backward'))
summary(backward.lm)

mse.bkwd <- mse(train.clean$SalePrice, predict(backward.lm, train.clean))
mae.bkwd <- mae(predict(backward.lm), train.clean$SalePrice)

sqrt(mse.bkwd)
mae.bkwd

stepAIC(backward.lm)
out.path = "/Users/serrauzun/Desktop/MSDS_410_Supervised/Assignment #7"
bckwrd_lm <- 'Backward.html';
stargazer(backward.lm, type=c('html'),out=paste(out.path,bckwrd_lm,sep=''),
          title=c('Backward LM'),
          align=TRUE, digits=2, digits.extra=2, initial.zero=TRUE,
          column.labels=c('Backward LM'), intercept.bottom=FALSE)


stepwise.lm <- stepAIC(object=sqft.lm,scope=list(upper=formula(upper.lm),lower=~1),
direction=c('both'));
summary(stepwise.lm)

mse.step <- mse(train.clean$SalePrice, predict(stepwise.lm, train.clean))
mae.step <- mae(predict(stepwise.lm), train.clean$SalePrice)

sqrt(mse.step)
mae.step

stepAIC(stepwise.lm)
out.path = "/Users/serrauzun/Desktop/MSDS_410_Supervised/Assignment #7"
stpws_lm <- 'Stepwise.html';
stargazer(stepwise.lm, type=c('html'),out=paste(out.path,stpws_lm,sep=''),
          title=c('Stepwise LM'),
          align=TRUE, digits=2, digits.extra=2, initial.zero=TRUE,
          column.labels=c('Steowise LM'), intercept.bottom=FALSE)


junk.lm <- lm(SalePrice ~ OverallQual + OverallCond + QualityIndex + GrLivArea + TotalSqftCalc,
data=train.df)
summary(junk.lm)
stepAIC(junk.lm)
AIC(junk.lm)

# Compute the VIF values
library(car)
```

```
sort(vif(forward.lm),decreasing=TRUE)
sort(vif(backward.lm),decreasing=TRUE)
sort(vif(stepwise.lm),decreasing=TRUE)
sort(vif(junk.lm),decreasing=TRUE)

AIC(forward.lm)
AIC(backward.lm)
AIC(stepwise.lm)

BIC(forward.lm)
BIC(backward.lm)
BIC(stepwise.lm)



###Predict
forward.test <- predict(forward.lm,newdata=test.df)
backward.test <- predict(backward.lm,newdata=test.df)
stepwise.test <- predict(stepwise.lm,newdata=test.df)
#junk.test <- predict(junk.lm,newdata=test.df)

summary(forward.test)

mse.fwd_test <- mse(test.df$SalePrice, predict(forward.lm,newdata=test.df))
mae.fwd_test <- mae(predict(forward.lm,newdata=test.df), test.df$SalePrice)
sqrt(mse.fwd_test)
mae.fwd_test

mse.bck_test <- mse(test.df$SalePrice, predict(backward.lm,newdata=test.df))
mae.bck_test <- mae(predict(backward.lm,newdata=test.df), test.df$SalePrice)
sqrt(mse.bck_test)
mae.bck_test

mse.stp_test <- mse(test.df$SalePrice, predict(stepwise.lm,newdata=test.df))
mae.stp_test <- mae(predict(stepwise.lm,newdata=test.df), test.df$SalePrice)
sqrt(mse.stp_test)
mae.stp_test


# Training Data
# Abs Pct Error
forward.pct <- abs(forward.lm$residuals)/train.clean$SalePrice
backward.pct <- abs(backward.lm$residuals)/train.clean$SalePrice
stepwise.pct <- abs(stepwise.lm$residuals)/train.clean$SalePrice
# Assign Prediction Grades;
forward.PredictionGrade <- ifelse(forward.pct<=0.10,'Grade 1: [0,0.10]',
                                  ifelse(forward.pct<=0.15,'Grade 2: (0.10,0.15]',
```

```
                                ifelse(forward.pct<=0.25,'Grade 3: (0.15,0.25]',
                                    'Grade 4: (0.25+]')))
backward.PredictionGrade <- ifelse(backward.pct<=0.10,'Grade 1: [0,0.10]',
                            ifelse(backward.pct<=0.15,'Grade 2: (0.10,0.15]',
                                ifelse(backward.pct<=0.25,'Grade 3: (0.15,0.25]',
                                    'Grade 4: (0.25+]')))
stepwise.PredictionGrade <- ifelse(stepwise.pct<=0.10,'Grade 1: [0,0.10]',
                            ifelse(stepwise.pct<=0.15,'Grade 2: (0.10,0.15]',
                                ifelse(stepwise.pct<=0.25,'Grade 3: (0.15,0.25]',
                                    'Grade 4: (0.25+]')))

forward.trainTable <- table(forward.PredictionGrade)
backward.trainTable <- table(backward.PredictionGrade)
stepwise.trainTable <- table(stepwise.PredictionGrade)

forward.trainTable/sum(forward.trainTable)
backward.trainTable/sum(backward.trainTable)
stepwise.trainTable/sum(stepwise.trainTable)

# Test Data
# Abs Pct Error
forward.testPCT <- abs(test.df$SalePrice-forward.test)/test.df$SalePrice
backward.testPCT <- abs(test.df$SalePrice-backward.test)/test.df$SalePrice
stepwise.testPCT <- abs(test.df$SalePrice-stepwise.test)/test.df$SalePrice


# Assign Prediction Grades;
forward.testPredictionGrade <- ifelse(forward.testPCT<=0.10,'Grade 1: [0.0.10]',
                            ifelse(forward.testPCT<=0.15,'Grade 2: (0.10,0.15]',
                                ifelse(forward.testPCT<=0.25,'Grade 3: (0.15,0.25]', 'Grade 4: (0.25+]')
                            ))
backward.testPredictionGrade <- ifelse(backward.testPCT<=0.10,'Grade 1: [0.0.10]',
                            ifelse(backward.testPCT<=0.15,'Grade 2: (0.10,0.15]',
                                ifelse(backward.testPCT<=0.25,'Grade 3: (0.15,0.25]', 'Grade 4: (0.25+]')
                            ))
stepwise.testPredictionGrade <- ifelse(stepwise.testPCT<=0.10,'Grade 1: [0.0.10]',
                            ifelse(stepwise.testPCT<=0.15,'Grade 2: (0.10,0.15]',
                                ifelse(stepwise.testPCT<=0.25,'Grade 3: (0.15,0.25]', 'Grade 4: (0.25+]')
                            ))

forward.testTable <-table(forward.testPredictionGrade)
backward.testTable <-table(backward.testPredictionGrade)
stepwise.testTable <-table(stepwise.testPredictionGrade)

forward.testTable/sum(forward.testTable)
backward.testTable/sum(backward.testTable)
stepwise.testTable/sum(stepwise.testTable)
```