Ames Housing – Property Value Prediction Report

Automated Variable Selection

Singh, Gurjeet

This report contains analysis done while building regression models for the for the home sale price on Ames housing data to understand and provide estimates of home values for typical homes in Ames, Iowa.

Table of Contents

Table of Figures

Introduction:

The purpose of this assignment is to build statistical models i.e. regression model to predict the value of a property or home. In this assignment, we will set up a predictive modeling framework, explore the use of automated variable selection techniques for model identification, assess the predictive accuracy of our model using cross-validation, and compare and contrast the difference between statistical model validation and an application (or business) model validation. There will be three different models namely forward variable selection (forward.lm), backward variable selection (backward.lm), stepwise variable selection (stepwise.lm), and one additional model i.e. junk model (junk.lm). Each of the first three models is discussed in their own section. Junk model is compared and discussed in section 2.4, Model Comparison.

For this assignment, we use the data set that contains information from the Ames Assessor's Office used in computing assessed values for individual residential properties sold in Ames, IA from 2016 to 2010. The data set contains 2930 observations and 82 explanatory variables which include 23 nominal, 23 ordinal, 14 discrete, and 20 continuous variables, and 2 additional observation identifiers.

Now that we have some context for our analysis and dataset, let's look at the results in the next section.

Section 1: Sample Definition and Data Split

Section 1.1: Sample Definition

For the purposes of the end goal, we have created a sample population. This eligible sample excludes the following kind of properties using the drop conditions. Figure 1 shows the drop conditions with the total count. Each of the drop conditions used to eliminate a property is explained below.

- Building type is not a single family.
- It not a normal sale
- Street are not paved
- Any house built prior to 1950
- There is no basement
- Living room square feet is below 800 and above 4000.
- There is no bedroom
- There is no kitchen
- There is no full bath
- There is no public utilities

Figure 1: Drop Conditions with count

	Total Count
01: Not SFR	505
02: Non-Normal Sale	423
03: Street Not Paved	6
04: Built Pre-1950	489
05: No Basement	28
06: LT 800 SqFt	9
07: LT 4000 SqFt	1
08: No Bedroom	4
09: No Kitchen	1
11: Not Public Utilities	1
99: Eligible Sample	1463

Next, I deleted any observations with any missing values after creating a list of predictor variables. Lastly, I created some discrete and indicator variables and saved the sample population as an .RData data object for the later use for the remainder of the assignment.

In this sample population, we have 1132 observations and 56 explanatory variables.

Section 1.2: The Train/Test Split

The sample population that we created in section 1.1, we will use that to split the sample into a 70/30 train/test split using the uniform random number. The 70/30 training/test split that we are using is the most basic form of cross-validation. With the train/test split, we now have two data sets: one for in-sample model development and the other one for out-of-sample model assessment. We will use train data set for our in-sample model development and test data set for our out-of-sample model assessment. The train data set was used in section 2 to develop the three automated selection models i.e. forward, backward, and stepwise. The test data set was used in section 3 to assess the predictive accuracy of the same models.

Figure 2: Train/Test data Partition

	Train_DF	Test_DF
Observation	802	330

Section 2: Model Identification and In-Sample Model Fit

This section explains the model identification of the three models i.e. Forward Variable Selection, Backward Variable Selection, and Stepwise Variable Selection. At the end, we perform the model comparison. For these model identification, we created a new data frame that only contains our response variable and the predictor variables that we include as our pool of predictor variables. Figure 3 shows the list of all the variables (Response, Predictor, and Indicator variables) that we used in our model development process.

Figure 3: Response, Predictor and Indicator variables

	Field_Names
1	LotFrontage
2	LotArea
3	BedroomAbvGr
4	TotRmsAbvGrd
5	Fireplaces
6	GarageCars
7	GarageArea
8	WoodDeckSF
9	OpenPorchSF
10	EnclosedPorch
11	ThreeSsnPorch
12	ScreenPorch
13	SalePrice
14	TotalSqftCalc
15	TotalBathCalc
16	CornerLotInd
17	CentralAirInd
18	BrickInd
19	VinylSidingInd
20	PoolInd
21	WoodDeckInd
22	PorchInd
23	QualityIndex

Section 2.1: Forward Variable Selection

In the forward variable selection approach, we start with no regressors and continue to add terms until adding another term makes the criterion of interest worse i.e. increase AIC. Figure 4 shows the last step at which the function stopped because adding any of the five remaining predictors (LotFrontage, TotalBathCalc, ScreenPorch, ThreeSsnPorch, and CentralAirInd) would increase the AIC.

Figure 4: forward.lm model: Last Step

```
Step:  AIC=16511.91
SalePrice ~ TotalSqftCalc + GarageCars + QualityIndex + VinylSidingInd +
    TotRmsAbvGrd + BedroomAbvGr + LotArea + GarageArea + PoolInd +
    OpenPorchSF + CornerLotInd + WoodDeckInd + Fireplaces + BrickInd +
    EnclosedPorch + WoodDeckSF

                Df  Sum of Sq          RSS   AIC
<none>                        671739547728 16512
+ LotFrontage    1 1196772312 670542775416 16513
+ TotalBathCalc  1  493610235 671245937493 16513
+ ScreenPorch    1  490773684 671248774044 16513
+ ThreeSsnPorch  1  414397756 671325149973 16513
+ CentralAirInd  1  133167295 671606380433 16514
```

Figure 5 shows the summary output of the forward selection model i.e. forward.lm. We look at the R-Squared statistic to measure how well our model is fitting the actual data. The R-Squared measure the linear relationship between our predictor variables and our response variable (SalePrice). The R-Squared we got is **0.8437** which is roughly 84% (approx.) of the variance found in the response variable (SalePrice) can be explained by the predictor variables.

Figure 5: forward.lm model: Output

```
Call:
lm(formula = SalePrice ~ TotalSqftCalc + GarageCars + QualityIndex +
    VinylSidingInd + TotRmsAbvGrd + BedroomAbvGr + LotArea +
    GarageArea + PoolInd + OpenPorchSF + CornerLotInd + WoodDeckInd +
    Fireplaces + BrickInd + EnclosedPorch + WoodDeckSF, data = train.clean)

Residuals:
    Min      1Q  Median      3Q     Max
-105858  -17726   -1486   16045  187230

Coefficients:
                Estimate Std. Error t value             Pr(>|t|)
(Intercept)    -86053.549   7612.039 -11.305 < 0.0000000000000002 ***
TotalSqftCalc      40.791      2.206  18.491 < 0.0000000000000002 ***
GarageCars      16278.894   3357.294   4.849       0.00000149759766 ***
QualityIndex     2065.796    172.808  11.954 < 0.0000000000000002 ***
VinylSidingInd  16865.987   2413.119   6.989       0.00000000000591 ***
TotRmsAbvGrd    11430.989   1309.495   8.729 < 0.0000000000000002 ***
BedroomAbvGr   -13577.072   2211.228  -6.140       0.00000000130914 ***
LotArea             1.610      0.292   5.514       0.00000004762055 ***
GarageArea         52.410     11.912   4.400       0.00001234415980 ***
PoolInd         45317.547  11651.206   3.890             0.000109 ***
OpenPorchSF        59.662     18.913   3.155             0.001669 **
CornerLotInd    -6295.554   2829.401  -2.225             0.026361 *
WoodDeckInd      8704.135   3425.116   2.541             0.011236 *
Fireplaces       3238.958   1969.306   1.645             0.100428
BrickInd         9776.671   6455.831   1.514             0.130328
EnclosedPorch     -28.082     18.146  -1.548             0.122129
WoodDeckSF        -19.490     13.182  -1.479             0.139663
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 29250 on 785 degrees of freedom
Multiple R-squared:  0.8437,    Adjusted R-squared:  0.8405
F-statistic: 264.8 on 16 and 785 DF,  p-value: < 0.00000000000000022
```

Figure 6 shows the Variance Inflation Factor (VIF) values of the forward.lm model for the predictor variables. The VIF values of the predictor variables indicate the strength of the linear relationship between the variable and remaining predictor variables. A good rule of thumb is that VIF values greater than 10 give some cause for concern. A low VIF value means that there are no high correlations among some or all predictor variables. In Figure 6, we see that all the VIF values are below 5. Hence, we can conclude that in this model that multicollinearity is not a problem. In section 2.4, we do the side-by-side comparison of these values.

Figure 6: forward.lm model: VIF values

	VIF_Values
GarageCars	4.673355
GarageArea	4.214643
TotRmsAbvGrd	2.792751
WoodDeckInd	2.717133
WoodDeckSF	2.660281
TotalSqftCalc	2.157398
BedroomAbvGr	1.693272
Fireplaces	1.590113
QualityIndex	1.450879
VinylSidingInd	1.361003
OpenPorchSF	1.210311
LotArea	1.176445
EnclosedPorch	1.111708
PoolInd	1.100782
BrickInd	1.088088
CornerLotInd	1.025362

Section 2.2: Backward Variable Selection

In the backward variable selection approach, we start with all the regressors in the model and continue to remove terms until removing another term makes the criterion of interest worse i.e. increase AIC. Figure 7 shows the last step at which the function stopped because removing any of the sixteen remaining predictors would increase the AIC. Hence, selected this model as our backward selection.

Figure 7: Backward.lm model: Last Step

```
Step:  AIC=16511.91
SalePrice ~ LotArea + BedroomAbvGr + TotRmsAbvGrd + Fireplaces +
    GarageCars + GarageArea + WoodDeckSF + OpenPorchSF + EnclosedPorch +
    TotalSqftCalc + CornerLotInd + BrickInd + VinylSidingInd +
    PoolInd + WoodDeckInd + QualityIndex

                 Df    Sum of Sq          RSS   AIC
<none>                             671739547728 16512
- WoodDeckSF      1   1870685634 673610233363 16512
- BrickInd        1   1962496999 673702044727 16512
- EnclosedPorch   1   2049425977 673788973705 16512
- Fireplaces      1   2314809472 674054357200 16513
- CornerLotInd    1   4236528232 675976075960 16515
- WoodDeckInd     1   5526268384 677265816112 16517
- OpenPorchSF     1   8515251286 680254799014 16520
- PoolInd         1  12945602893 684685150621 16525
- GarageArea      1  16563508471 688303056200 16529
- GarageCars      1  20118805075 691858352803 16534
- LotArea         1  26017518699 697757066427 16540
- BedroomAbvGr    1  32260885681 704000433410 16548
- VinylSidingInd  1  41802020309 713541568037 16558
- TotRmsAbvGrd    1  65206572366 736946120094 16584
- QualityIndex    1 122286638244 794026185973 16644
- TotalSqftCalc   1 292569667386 964309215115 16800
```

Figure 8 shows the summary output of the backward selection model i.e. backward.lm. We look at the R-Squared statistic to measure how well our model is fitting the actual data. The R-Squared measure the linear relationship between our predictor variables and our response variable (SalePrice). The R-Squared we got is **0.8437** which is roughly 84% (approx.) of the variance found in the response variable (SalePrice) can be explained by the predictor variables.

The reason we have the same R-Squared as of forward.lm model is because of both of the models, forward.lm and backward.lm, have selected the same predictor variables. Hence, our models are the same.

Figure 8: backward.lm model: Output

```
Call:
lm(formula = SalePrice ~ LotArea + BedroomAbvGr + TotRmsAbvGrd +
    Fireplaces + GarageCars + GarageArea + WoodDeckSF + OpenPorchSF +
    EnclosedPorch + TotalSqftCalc + CornerLotInd + BrickInd +
    VinylSidingInd + PoolInd + WoodDeckInd + QualityIndex, data = train.clean)

Residuals:
    Min      1Q  Median      3Q     Max
-105858  -17726   -1486   16045  187230

Coefficients:
                 Estimate Std. Error t value            Pr(>|t|)
(Intercept)    -86053.549   7612.039 -11.305 < 0.0000000000000002 ***
LotArea             1.610      0.292   5.514     0.00000004762055 ***
BedroomAbvGr   -13577.072   2211.228  -6.140     0.00000000130914 ***
TotRmsAbvGrd    11430.989   1309.495   8.729 < 0.0000000000000002 ***
Fireplaces       3238.958   1969.306   1.645             0.100428
GarageCars      16278.894   3357.294   4.849     0.00000149759766 ***
GarageArea         52.410     11.912   4.400     0.00001234415980 ***
WoodDeckSF        -19.490     13.182  -1.479             0.139663
OpenPorchSF        59.662     18.913   3.155             0.001669 **
EnclosedPorch     -28.082     18.146  -1.548             0.122129
TotalSqftCalc      40.791      2.206  18.491 < 0.0000000000000002 ***
CornerLotInd    -6295.554   2829.401  -2.225             0.026361 *
BrickInd         9776.671   6455.831   1.514             0.130328
VinylSidingInd  16865.987   2413.119   6.989     0.00000000000591 ***
PoolInd         45317.547  11651.206   3.890             0.000109 ***
WoodDeckInd      8704.135   3425.116   2.541             0.011236 *
QualityIndex     2065.796    172.808  11.954 < 0.0000000000000002 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 29250 on 785 degrees of freedom
Multiple R-squared:  0.8437,    Adjusted R-squared:  0.8405
F-statistic: 264.8 on 16 and 785 DF,  p-value: < 0.00000000000000022
```

Figure 9 shows the Variance Inflation Factor (VIF) values of the backward.lm model for the predictor variables. The VIF values of the predictor variables indicate the strength of the linear relationship between the variable and remaining predictor variables. A good rule of thumb is that VIF values greater than 10 give some cause for concern. A low VIF value means that there are no high correlations among some or all predictor variables. In Figure 9, we see that all the VIF values are again below 5. Hence, we can conclude that in this model that multicollinearity is not a problem.

If we compare the backward.lm VIF values (Figure 9) with forward.lm (Figure 6), we will notice the values are the same. This is because both of these models have selected the same predictor variables. Hence, they are the same models. Therefore, we got the same VIF values. In section 2.4, we do the side-by-side comparison of these values.

Figure 9: backward.lm model: VIF values

	VIF_Values
GarageCars	4.673355
GarageArea	4.214643
TotRmsAbvGrd	2.792751
WoodDeckInd	2.717133
WoodDeckSF	2.660281
TotalSqftCalc	2.157398
BedroomAbvGr	1.693272
Fireplaces	1.590113
QualityIndex	1.450879
VinylSidingInd	1.361003
OpenPorchSF	1.210311
LotArea	1.176445
EnclosedPorch	1.111708
PoolInd	1.100782
BrickInd	1.088088
CornerLotInd	1.025362

Section 2.3: Stepwise Variable Selection

In the stepwise variable selection approach, we start with one regressor in the model and continue to remove or add terms until removing or adding another term makes the criterion of interest worse i.e. increase AIC. Figure 10 shows the last step at which the function stopped because removing or adding any of the twenty-one remaining predictors would increase the AIC. Hence, selected this model as our stepwise selection.

Figure 10: Stepwise.lm model: Last Step

```
Step:  AIC=16511.91
SalePrice ~ TotalSqftCalc + GarageCars + QualityIndex + VinylSidingInd +
    TotRmsAbvGrd + BedroomAbvGr + LotArea + GarageArea + PoolInd +
    OpenPorchSF + CornerLotInd + WoodDeckInd + Fireplaces + BrickInd +
    EnclosedPorch + WoodDeckSF

                 Df   Sum of Sq          RSS   AIC
<none>                          671739547728 16512
- WoodDeckSF      1  1870685634 673610233363 16512
- BrickInd        1  1962496999 673702044727 16512
- EnclosedPorch   1  2049425977 673788973705 16512
+ LotFrontage     1  1196772312 670542775416 16513
- Fireplaces      1  2314809472 674054357200 16513
+ TotalBathCalc   1   493610235 671245937493 16513
+ ScreenPorch     1   490773684 671248774044 16513
+ ThreeSsnPorch   1   414397756 671325149973 16513
+ CentralAirInd   1   133167295 671606380433 16514
- CornerLotInd    1  4236528232 675976075960 16515
- WoodDeckInd     1  5526268384 677265816112 16517
- OpenPorchSF     1  8515251286 680254799014 16520
- PoolInd         1 12945602893 684685150621 16525
- GarageArea      1 16563508471 688303056200 16529
- GarageCars      1 20118805075 691858352803 16534
- LotArea         1 26017518699 697757066427 16540
- BedroomAbvGr    1 32260885681 704000433410 16548
- VinylSidingInd  1 41802020309 713541568037 16558
- TotRmsAbvGrd    1 65206572366 736946120094 16584
- QualityIndex    1 122286638244 794026185973 16644
- TotalSqftCalc   1 292569667386 964309215115 16800
```

Figure 11 shows the summary output of the stepwise selection model i.e. forward.lm. We look at the R-Squared statistic to measure how well our model is fitting the actual data. The R-Squared measure the linear relationship between our predictor variables and our response variable (SalePrice). The R-Squared we got is **0.8437** which is roughly 84% (approx.) of the variance found in the response variable (SalePrice) can be explained by the predictor variables.

The reason we have the same R-Squared as of forward.lm and backward models is because of all three models, forward.lm, backward.lm, and stepwise.lm, have selected the same predictor variables. Hence, our models are the same as well as our R-Squared values.

Figure 11: stepwise.lm model: Output

```
Call:
lm(formula = SalePrice ~ TotalSqftCalc + GarageCars + QualityIndex +
    VinylSidingInd + TotRmsAbvGrd + BedroomAbvGr + LotArea +
    GarageArea + PoolInd + OpenPorchSF + CornerLotInd + WoodDeckInd +
    Fireplaces + BrickInd + EnclosedPorch + WoodDeckSF, data = train.clean)

Residuals:
    Min      1Q  Median      3Q     Max
-105858  -17726   -1486   16045  187230

Coefficients:
                 Estimate Std. Error t value             Pr(>|t|)
(Intercept)    -86053.549   7612.039 -11.305 < 0.0000000000000002 ***
TotalSqftCalc      40.791      2.206  18.491 < 0.0000000000000002 ***
GarageCars      16278.894   3357.294   4.849      0.00000149759766 ***
QualityIndex     2065.796    172.808  11.954 < 0.0000000000000002 ***
VinylSidingInd  16865.987   2413.119   6.989      0.00000000000591 ***
TotRmsAbvGrd    11430.989   1309.495   8.729 < 0.0000000000000002 ***
BedroomAbvGr   -13577.072   2211.228  -6.140      0.00000000130914 ***
LotArea             1.610      0.292   5.514      0.00000004762055 ***
GarageArea         52.410     11.912   4.400      0.00001234415980 ***
PoolInd         45317.547  11651.206   3.890              0.000109 ***
OpenPorchSF        59.662     18.913   3.155              0.001669 **
CornerLotInd    -6295.554   2829.401  -2.225              0.026361 *
WoodDeckInd      8704.135   3425.116   2.541              0.011236 *
Fireplaces       3238.958   1969.306   1.645              0.100428
BrickInd         9776.671   6455.831   1.514              0.130328
EnclosedPorch     -28.082     18.146  -1.548              0.122129
WoodDeckSF        -19.490     13.182  -1.479              0.139663
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 29250 on 785 degrees of freedom
Multiple R-squared:  0.8437,    Adjusted R-squared:  0.8405
F-statistic: 264.8 on 16 and 785 DF,  p-value: < 0.00000000000000022
```

Figure 12 shows the Variance Inflation Factor (VIF) values of the stepwise.lm model for the predictor variables. The VIF values of the predictor variables indicate the strength of the linear relationship between the variable and remaining predictor variables. A good rule of thumb is that VIF values greater than 10 give some cause for concern. A low VIF value means that there are no high correlations among some or all predictor variables. In Figure 12, we see that all the VIF values are again below 5. Hence, we can conclude that in this model that multicollinearity is not a problem.

If we compare the stepwise.lm VIF values (Figure 12) with backward.lm (Figure 9) and forward.lm (Figure 6), we will notice the VIF values are the same for each of these models. This is because all three of these models have selected the same predictor variables. Hence, they are the same models. Therefore, we got the same VIF values. In section 2.4, we do the side-by-side comparison of these values.

Figure 12: stepwise.lm model: VIF values

	VIF_Values
GarageCars	4.673355
GarageArea	4.214643
TotRmsAbvGrd	2.792751
WoodDeckInd	2.717133
WoodDeckSF	2.660281
TotalSqftCalc	2.157398
BedroomAbvGr	1.693272
Fireplaces	1.590113
QualityIndex	1.450879
VinylSidingInd	1.361003
OpenPorchSF	1.210311
LotArea	1.176445
EnclosedPorch	1.111708
PoolInd	1.100782
BrickInd	1.088088
CornerLotInd	1.025362

Section 2.4: Model Comparison

Figure 13 shows each of our four models i.e. forward selection, backward selection, stepwise selection, and junk model. We have created a fourth model i.e. junk mode for model comparison purposes. When comparing our models, the junk model here outperformed all the other three models by every measure. However, there's no guarantee that the junk model will be predictive. Hence, we check for the multicollinearity for each to the models to make sure the predictor variables used in these models have no high correlation among themselves.

In Figure 13, we see that our forward, backward, and stepwise selection have selected the same model. Hence, the metrics in these models are the same as seen in section 2.1, section 2.2, and section 2.3.

Figure 13: Four Models: Forward, Backward, Stepwise, and Junk

```
> forward_selection_model
lm(formula = SalePrice ~ TotalSqftCalc + GarageCars + QualityIndex +
    VinylSidingInd + TotRmsAbvGrd + BedroomAbvGr + LotArea +
    GarageArea + PoolInd + OpenPorchSF + CornerLotInd + WoodDeckInd +
    Fireplaces + BrickInd + EnclosedPorch + WoodDeckSF, data = train.clean)
>
>
> backward_selection_model
lm(formula = SalePrice ~ LotArea + BedroomAbvGr + TotRmsAbvGrd +
    Fireplaces + GarageCars + GarageArea + WoodDeckSF + OpenPorchSF +
    EnclosedPorch + TotalSqftCalc + CornerLotInd + BrickInd +
    VinylSidingInd + PoolInd + WoodDeckInd + QualityIndex, data = train.clean)
>
>
> stepwise_selection_model
lm(formula = SalePrice ~ TotalSqftCalc + GarageCars + QualityIndex +
    VinylSidingInd + TotRmsAbvGrd + BedroomAbvGr + LotArea +
    GarageArea + PoolInd + OpenPorchSF + CornerLotInd + WoodDeckInd +
    Fireplaces + BrickInd + EnclosedPorch + WoodDeckSF, data = train.clean)
>
>
> junk_selection_model
lm(formula = SalePrice ~ OverallQual + OverallCond + QualityIndex +
    GrLivArea + TotalSqftCalc, data = train.df)
```

Figure 14 shows the Variance Inflation Factor (VIF) values of all the models for the predictor variables. The VIF values of the predictor variables indicate the strength of the linear relationship between the variable and remaining predictor variables. As stated before a good rule of thumb is that VIF values greater than 10 give some cause for concern. A low VIF value means that there are no high correlations among some or all predictor variables. In Figure 14, we see that all the VIF values are below 5 for forward, backward, and stepwise selection. However, we see large VIF values i.e. greater than 10, for three of the five predictor variables of the junk mode. Therefore, multicollinearity is not a problem for forward, backward, and stepwise selection models but it is for junk model. Hence, we call the junk model junk because it leads to unreliable and unstable estimates of regression coefficients.

We should not be concerned with VIF values for indicator variables because if they are not considered to be important as compared to the other variables, they will be dropped automatically.

Figure 14: VIF Values: Forward, Backward, Stepwise, and Junk

Forward, Backward, and Stepwise Models

	forward.VIF	backward.VIF	stepwise.VIF
GarageCars	4.673355	4.673355	4.673355
GarageArea	4.214643	4.214643	4.214643
TotRmsAbvGrd	2.792751	2.792751	2.792751
WoodDeckInd	2.717133	2.717133	2.717133
WoodDeckSF	2.660281	2.660281	2.660281
TotalSqftCalc	2.157398	2.157398	2.157398
BedroomAbvGr	1.693272	1.693272	1.693272
Fireplaces	1.590113	1.590113	1.590113
QualityIndex	1.450879	1.450879	1.450879
VinylSidingInd	1.361003	1.361003	1.361003
OpenPorchSF	1.210311	1.210311	1.210311
LotArea	1.176445	1.176445	1.176445
EnclosedPorch	1.111708	1.111708	1.111708
PoolInd	1.100782	1.100782	1.100782
BrickInd	1.088088	1.088088	1.088088
CornerLotInd	1.025362	1.025362	1.025362

Junk Model

	junk.VIF
QualityIndex	67.082864
OverallQual	60.469459
OverallCond	33.396955
GrLivArea	3.109841
TotalSqftCalc	2.320355

Figure 15 shows the adjusted R-Squared, AIC, BIC, mean squared error (MSE), and the mean absolute error (MSE) for each of these models i.e. junk, forward, backward, and stepwise. In addition, Figure 15 provides the rank for each model based on the metric values. Since my forward, backward, and stepwise models selected the same predictor variables, all these models fell into rank 2 because each metric values were the same for those three models. Even though junk model is unreliable, for the purposes of ranking, it takes the rank one due to metric values. I expected each metric to give the same ranking of model fit.

Figure 15: Metrics: Forward, Backward, Stepwise, and Junk

	Rank	Adjusted_R_Squared	AIC_Values	BIC_Values	MSE_Values	MAE_Values
junk.lm	1	0.8538010	18709.21	18742.02	778488283	20795.79
Forward.lm	2	0.8404994	18789.89	18874.26	837580483	21221.65
backward.lm	2	0.8404994	18789.89	18874.26	837580483	21221.65
stepwise.lm	2	0.8404994	18789.89	18874.26	837580483	21221.65

Section 3: Predictive Accuracy

Next, we test the predictive accuracy of our model in the out-of-sample population. Figure 16 shows the Mean Squared Error (MSE) and the Mean Absolute Error (MAE) for each of the four models in the test sample. Based on the criteria in Figure 16, we see that the junk model fits the best. However, since we know that junk model is unstable, we consider other models to evaluate the fitness based on the criteria. All three of the models i.e. forward, backward, and stepwise, fit the best on these criteria. The reason these all fit because they have selected the same predictor variables.

Based on the output, I noticed that the junk model fit best in-sample predicted the best out-of-sample as well. If we compare the MAE values in Figure 15 (in-sample) and Figure 16 (out-of-sample), we notice that in Figure 16 (out-of-sample) MAE value is less than the Figure 15 (in-sample). However, this might be pure chance and luck due to the out-of-sample data. For the purposes of discussion, let's ignore junk model for now. We notice that remaining three models did not predict the best out-of-sample as compared to in-sample since the MAE and MSE values are higher in out-of-sample. However, the difference is not that high. Looking at the values in Figure 16 and Figure 15, we only have the difference of $2,433 (approx.) which is not the best nor the worst, in my opinion. My own preference is always to use MAE for comparison and explanation because its unit is same as of response variable. Therefore, it helps to convey the message and make the point across easily. Since we know that we have a difference in MAE, our models have better predictive accuracy in-sample then out-of-sample. Hence, it means that our model is slightly overfitting. I say slightly here because the difference is not a lot.

Figure 16: Out-of-Sample Metrics

	MSE_Values	MAE_Values
junk.lm.test	877239140	19827.30
Forward.lm.test	1287709268	23665.08
backward.lm.test	1287709268	23665.08
stepwise.lm.test	1287709268	23665.08

Section 4: Operational Validation

So far we have validated our models in the statistical sense. In this section, we validate these models in the business sense as well. For our business rule, we state the same policy that GSEs use to rate an AVM model as 'underwriting quality' i.e. we need our model to be accurate to within ten percent (10%) more than fifty percent (50%) of the time. In order to test our model against the business rule, we categorize the predicted value into different grades: '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' over twenty-five percent of the actual value.

Figure 17 shows the table in distribution form of the prediction grades for the in-samples training data and the out-of-sample test data. Columns that have ".train.result" are the results from in-sample training data and columns that have ".test.results" are the results from out-of-sample test data.

Based on the results in Figure 17, we see that both in-sample and out-sample pass the policy of our business rule. We see that our models i.e. forward, backward, and stepwise selection, are accurate within ten percent more than fifty percent of the time. Hence, we see the majority falls under Grade 1. When comparing the results to our predictive accuracy results in section 3, we see the trend remains the same. The models predicted out-of-sample results lower than the in-sample results. However, our models passed the underwriting quality in out-of-sample test data.

Figure 17: Prediction Grades

	forward.train.result	forward.test.result	backward.train.result	backward.test.result	stepwise.train.result	stepwise.test.result	junk.train.result	junk.test.result
Grade 1: [0,0.10]	0.53865337	0.5181818	0.53865337	0.5181818	0.53865337	0.5181818	0.55112219	0.58787879
Grade 2: (0.10,0.15]	0.19825436	0.1969697	0.19825436	0.1969697	0.19825436	0.1969697	0.19576060	0.21515152
Grade 3: (0.15,0.25]	0.17331671	0.1848485	0.17331671	0.1848485	0.17331671	0.1848485	0.17082294	0.13636364
Grade 4: (0.25+]	0.08977556	0.1000000	0.08977556	0.1000000	0.08977556	0.1000000	0.08229426	0.06060606

Conclusion:

In conclusion, I would like to state that even the junk model here outperformed all the other three models by every measure. However, there's no guarantee that the junk model will be predictive. It did not pass the multicollinearity test. Even though multicollinearity does not affect prediction by much, it is always good to check for it.

Our three models i.e. forward, backward, and stepwise selections, selected the same predictor variables. Hence, the metrics were the same for each of these. When tested these models for predictive accuracy, we came to know that these models were overfitting. However, the difference of approx. $2,433 was not that significant in terms of the sale price of the house.

Lastly, all three of my automated variable selection models are accurate to within ten percent more than fifty percent of the time. Hence, models are of 'underwriting quality' as per the GSEs rating.