Assignment # 5: Automated Variable Selection, Multicollinearity, and Predictive Modeling

Setu Madhavi Namburu

05/06/2018

Introduction

In this assignment, we will develop predictive models for predicting home prices using Ames Iowa housing dataset. In section 1, the sample data considered along with drop conditions are explained. In section 2, framework for predictive modeling is established by diving the data into training and testing sets. Model identification is performed using different automated variable selection methods along with junk model (where predictors are randomly chosen) in section 3 and different fit metrics are compared between the models. In section 4, predictive accuracy on testing data is compared between the models followed by operational validation in section 5 for business policy development. Diagnostic tests are performed on "best" model (in terms of predictive accuracy) in section 6 to check for any violations of OLS assumptions. The report is finally concluded with a summary containing reflections and recommendations.

Section 1. Sample Definition

The dataset contains variables used in assessing values of individual properties sold in Ames, IA from 2006 to 2010. It has 80 variables (23 nominal, 23 ordinal, 14 discrete, and 20 continuous) and two additional observation identifiers for 2930 properties (number of observations) resulting in 82 columns and 2930 rows.

The problem statement is, Can we build a linear regression model to accurately predict the home values based on historical data. The first step in model building is data survey & data sampling followed by exploratory data analysis (In this assignment, predictors will be selected based on EDA done in last assignments). The data given has several explanatory variables which represent physical attributes of the properties in assessing home values. Since the goal is to provide 'typical' home values, we need to make sure we have appropriate sample of the data to represent typical population of the properties. Assuming typical home buyers give most preference to type of the property (BldgType where individual properties like family homes can be treated different from other types of properties), type of sale(not foreclosure type of sales) and zoning(prefer residential zones) associated with it, we will select sample data using drop conditions on these three variables. Below table shows number of samples dropped with conditions on each of these variables and remaining samples after these drop conditions which will be used as the sample data for the remainder of the analysis. Due to lack of knowledge in the subject or not having means to validate outliers or erroneous data, only 4 samples (where saleprices

are validated as unusual data points) recommended by the author of the data are deleted from the dataset resulting in **2232 observations** for further analysis.

Waterfall drop conditions

DropCondition	Freq
01-Non single family	505
02-Non-residential	26
03-Abnormal sale	163
RemainingSamples	2236
Sum	2930

Section 2: The Predictive Modeling Framework

A typical way to assess performance of a predictive model is using out-of-sample data. 70/30 training test split is most commonly used basic form of cross-validation method to assess the model performance. The sample data of 2232 observations is randomized using uniform distribution and 70% of the data is identified as training set and remaining 30% as test set. The count of observations in each split are shown below for training (building) and testing the models.

Observation counts of train-test splits

	x
train.count	1568
test.count	664
total.count	2232

Section 3: Model Identification by Automated Variable Selection

Model specification starts with choice of predictor variables. While there is no standard strategy for selecting predictor variables, the model building process guides the analyst to make informed decisions about choice of the raw variables/calculating new variables/transformations if any needed and it is an iterative process until final model is determined to put into practice. Exploratory data analysis (EDA) is basic building block for visually observing the data so the analyst can make judgment along with statistics generated at each step in model building process. In this assignment 6 variables are calculated based on subject matter expertise or based on learnings from last 3 assignments. Also the categorical variable Neighborhood is recoded into 3 sub-groups based on mean sale prices from all neighborhoods (based on assignment 3).

The calculated variables are:

```
ames$TotalFloorSF <- ames$FirstFlrSF + ames$SecondFlrSF
```

ames$HouseAge <- ames$YrSold - ames$YearBuilt

ames$QualityIndex <- ames$OverallQual * ames$OverallCond

ames$HouseArea <- ames$GrLivArea + ames$TotalBsmtSF

ames$TotalSqftCalc <- ames$BsmtFinSF1+ames$BsmtFinSF2+ames$GrLivArea

ames$price_sqft <- ames$SalePrice/ames$TotalFloorSF

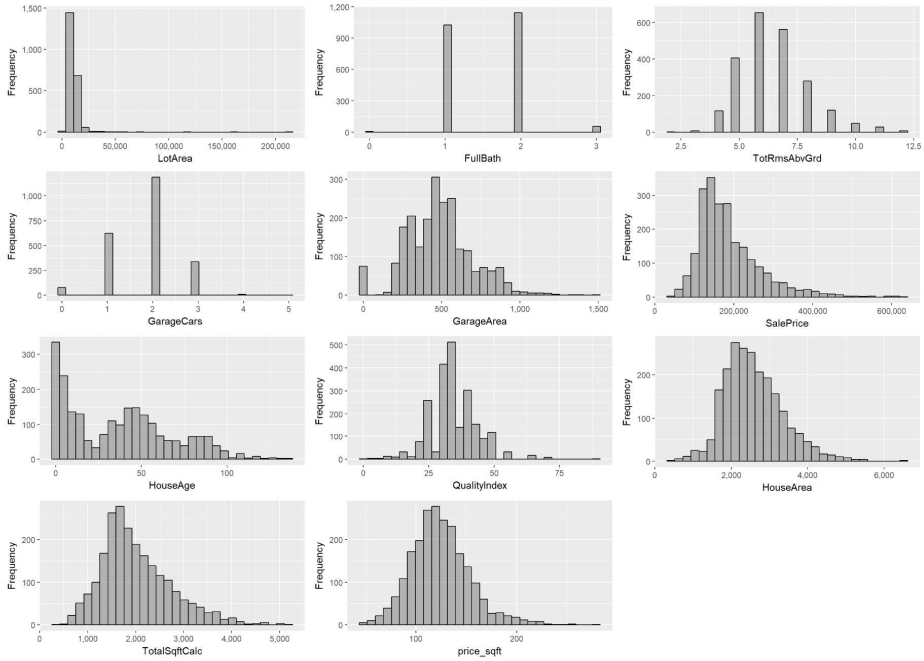
From all raw/calculated/regrouped variables 18 variables are chosen as pool of candidate predictor variables for this assignment. These variables are selected based on past assignments EDA/subject matter expertise (if I were to buy a home). The below table presents the names of 18 variables, where the 10 variables highlighted in red are numerical/continuous variables and 8 variables highlighted in green are categorical variables. The SalePrice is the response variable throughout this assignment. To keep the data further clean the rows with missing values in any of these variables are deleted from the dataset resulting in **1538 samples for training/model building and 650 samples for testing.**

Predictors of Interest

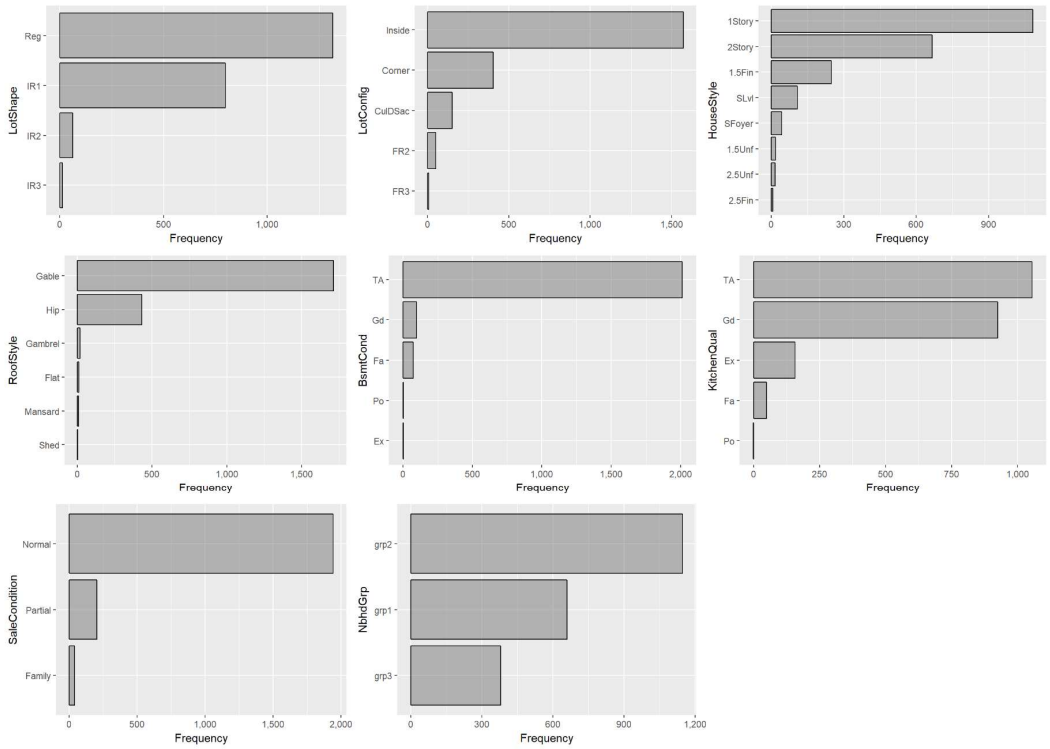
GarageCars	TotalSqftCalc	LotShape
GarageArea	HouseAge	SaleCondition
LotArea	HouseArea	RoofStyle
FullBath	QualityIndex	LotConfig
TotRmsAbvGrd	NbhdGrp	KitchenQual
price_sqft	HouseStyle	BsmtCond

Below histogram and bar plots give glimpse of selected continuous and categorical variables visually. Based on the visual observations (skewed distributions, imbalanced categories etc), there may be transformations/regroupings warranted on some of the variables but the goal of the assignment is to automatically select the variables for better predictive assessment, so no other changes are made to the variables at this point and model building process is continued using automated variable selection methods.

Continuous Features (Histogram)



Discrete Features (Bar Chart)



When number of variables are very high or the analyst has no prior subject matter expertise, automated variable selection methods can come very handy in selecting

potential predictor variables where the variables are selected based on model fit statistics (AIC, BIC, Mallows Cp, Adj. R-sq, t-values) at each iteration of search criteria and the model with minimum AIC value is chosen.

3.1 Forward Selection

In the forward variable selection method, the model starts with intercept model and one variable is added at a time until optimal model is determined based on fit statistics. The 70% data is fit using forward selection algorithm where 18 predictor variables are used one at a time (R automatically converts categorical variables into dummy variables and the reference category is arbitrary, so the model need to be interpreted relative to baseline intercept). The final model with minimum AIC value for predicting the Saleprice is determined as (forward.lm): "SalePrice = (-141616.04) + (52.76)*HouseArea + (1353.15)*price_sqft + (-17024.5)*HouseStyle1.5Unf + (-24903.94)*HouseStyle1Story + (-8047.24)*HouseStyle2.5Fin + (2961.76)*HouseStyle2.5Unf + (11575.18)*HouseStyle2Story + (-29471.12)*HouseStyleSFoyer + (-11231.31)*HouseStyleSLvl + (-15581.34)*KitchenQualFa + (-23027.75)*KitchenQualGd + (-12595.97)*KitchenQualPo + (-27058.44)*KitchenQualTA + (4564.63)*TotRmsAbvGrd + (8.69)*TotalSqftCalc + (0.23)*LotArea + (263.22)*QualityIndex + (3053.73)*GarageCars + (-3761.94)*SaleConditionNormal + (3196.42)*SaleConditionPartial + (-4614.32)*NbhdGrpgrp2 + (-9620.7)*NbhdGrpgrp3 + (-2843.81)*RoofStyleGable + (-1950.01)*RoofStyleGambrel + (866.38)*RoofStyleHip + (12419.48)*RoofStyleMansard + (22259.53)*RoofStyleShed + (5035.33)*LotConfigCulDSac + (-4335.14)*LotConfigFR2 + (2873.82)*LotConfigFR3 + (-850.58)*LotConfigInside + (1945.87)*FullBath"

The overall F-statistic indicates the model is significant. The model achieved Adjusted R-sq value of 95.42% which is very near to R-sq value of 95.51% indicating that the variables selected are capturing almost all possible variation in the model. The residual standard error indicates a possible predictive error in home sale prices is +/- $17250. Most of the predictors are statistically significant based on their t-values, except some predictor variables where it may be possible to re-group them.

```
##
## Call:
## lm(formula = SalePrice ~ HouseArea + price_sqft + HouseStyle +
##     KitchenQual + TotRmsAbvGrd + TotalSqftCalc + LotArea + QualityIndex +
##     GarageCars + SaleCondition + NbhdGrp + RoofStyle + LotConfig +
##     FullBath, data = train.clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -106461   -8833   -1341    6304  133666
##
## Coefficients:
```

```
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -1.416e+05  9.615e+03 -14.728  < 2e-16 ***
## HouseArea             5.276e+01  1.525e+00  34.595  < 2e-16 ***
## price_sqft            1.353e+03  3.810e+01  35.520  < 2e-16 ***
## HouseStyle1.5Unf     -1.702e+04  5.076e+03  -3.354 0.000817 ***
## HouseStyle1Story     -2.490e+04  1.775e+03 -14.034  < 2e-16 ***
## HouseStyle2.5Fin     -8.047e+03  8.006e+03  -1.005 0.314985
## HouseStyle2.5Unf      2.962e+03  5.980e+03   0.495 0.620457
## HouseStyle2Story      1.158e+04  1.702e+03   6.801 1.49e-11 ***
## HouseStyleSFoyer     -2.947e+04  3.546e+03  -8.310  < 2e-16 ***
## HouseStyleSLvl       -1.123e+04  2.492e+03  -4.506 7.10e-06 ***
## KitchenQualFa        -1.558e+04  3.971e+03  -3.924 9.09e-05 ***
## KitchenQualGd        -2.303e+04  2.197e+03 -10.480  < 2e-16 ***
## KitchenQualPo        -1.260e+04  1.754e+04  -0.718 0.472830
## KitchenQualTA        -2.706e+04  2.505e+03 -10.801  < 2e-16 ***
## TotRmsAbvGrd          4.565e+03  5.634e+02   8.102 1.11e-15 ***
## TotalSqftCalc         8.689e+00  1.203e+00   7.223 8.06e-13 ***
## LotArea               2.273e-01  5.628e-02   4.039 5.64e-05 ***
## QualityIndex          2.632e+02  6.185e+01   4.256 2.21e-05 ***
## GarageCars            3.054e+03  9.002e+02   3.392 0.000711 ***
## SaleConditionNormal  -3.762e+03  3.385e+03  -1.111 0.266643
## SaleConditionPartial  3.196e+03  3.771e+03   0.848 0.396832
## NbhdGrpgrp2          -4.614e+03  1.532e+03  -3.013 0.002632 **
## NbhdGrpgrp3          -9.621e+03  2.831e+03  -3.399 0.000694 ***
## RoofStyleGable       -2.844e+03  6.629e+03  -0.429 0.667972
## RoofStyleGambrel     -1.950e+03  8.116e+03  -0.240 0.810162
## RoofStyleHip          8.664e+02  6.667e+03   0.130 0.896623
## RoofStyleMansard      1.242e+04  9.718e+03   1.278 0.201431
## RoofStyleShed         2.226e+04  1.197e+04   1.860 0.063086 .
## LotConfigCulDSac      5.035e+03  2.033e+03   2.477 0.013371 *
## LotConfigFR2         -4.335e+03  2.877e+03  -1.507 0.132096
## LotConfigFR3          2.874e+03  7.170e+03   0.401 0.688624
## LotConfigInside      -8.506e+02  1.176e+03  -0.723 0.469545
## FullBath              1.946e+03  1.199e+03   1.623 0.104884
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17250 on 1505 degrees of freedom
## Multiple R-squared:  0.9551, Adjusted R-squared:  0.9542
## F-statistic:  1001 on 32 and 1505 DF,  p-value: < 2.2e-16
```

3.2 Backward Elimination

In the backward variable elimination method, the model starts with all potential predictor variables and one variable is dropped at a time until optimal model is determined based on fit statistics. The 70% data is fit using backward elimination algorithm (R automatically converts categorical variables into dummy variables, so the model need to be interpreted relative to baseline intercept). The final model with minimum AIC value for predicting the

Saleprice is determined as (backward.lm): "SalePrice = (-141616.04) + (0.23)*LotArea + (5035.33)*LotConfigCulDSac + (-4335.14)*LotConfigFR2 + (2873.82)*LotConfigFR3 + (-850.58)*LotConfigInside + (-17024.5)*HouseStyle1.5Unf + (-24903.94)*HouseStyle1Story + (-8047.24)*HouseStyle2.5Fin + (2961.76)*HouseStyle2.5Unf + (11575.18)*HouseStyle2Story + (-29471.12)*HouseStyleSFoyer + (-11231.31)*HouseStyleSLvl + (-2843.81)*RoofStyleGable + (-1950.01)*RoofStyleGambrel + (866.38)*RoofStyleHip + (12419.48)*RoofStyleMansard + (22259.53)*RoofStyleShed + (1945.87)*FullBath + (-15581.34)*KitchenQualFa + (-23027.75)*KitchenQualGd + (-12595.97)*KitchenQualPo + (-27058.44)*KitchenQualTA + (4564.63)*TotRmsAbvGrd + (3053.73)*GarageCars + (-3761.94)*SaleConditionNormal + (3196.42)*SaleConditionPartial + (263.22)*QualityIndex + (52.76)*HouseArea + (8.69)*TotalSqftCalc + (1353.15)*price_sqft + (-4614.32)*NbhdGrpgrp2 + (-9620.7)*NbhdGrpgrp3". This model is same as the model selected using forward variable selection method.

```
## Call:
## lm(formula = SalePrice ~ LotArea + LotConfig + HouseStyle + RoofStyle + 
##     FullBath + KitchenQual + TotRmsAbvGrd + GarageCars + SaleCondition + 
##     QualityIndex + HouseArea + TotalSqftCalc + price_sqft + NbhdGrp, 
##     data = train.clean)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -106461   -8833   -1341    6304  133666 
## 
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)    
## (Intercept)          -1.416e+05  9.615e+03 -14.728  < 2e-16 ***
## LotArea               2.273e-01  5.628e-02   4.039 5.64e-05 ***
## LotConfigCulDSac      5.035e+03  2.033e+03   2.477 0.013371 *  
## LotConfigFR2         -4.335e+03  2.877e+03  -1.507 0.132096    
## LotConfigFR3          2.874e+03  7.170e+03   0.401 0.688624    
## LotConfigInside      -8.506e+02  1.176e+03  -0.723 0.469545    
## HouseStyle1.5Unf     -1.702e+04  5.076e+03  -3.354 0.000817 ***
## HouseStyle1Story     -2.490e+04  1.775e+03 -14.034  < 2e-16 ***
## HouseStyle2.5Fin     -8.047e+03  8.006e+03  -1.005 0.314985    
## HouseStyle2.5Unf      2.962e+03  5.980e+03   0.495 0.620457    
## HouseStyle2Story      1.158e+04  1.702e+03   6.801 1.49e-11 ***
## HouseStyleSFoyer     -2.947e+04  3.546e+03  -8.310  < 2e-16 ***
## HouseStyleSLvl       -1.123e+04  2.492e+03  -4.506 7.10e-06 ***
## RoofStyleGable       -2.844e+03  6.629e+03  -0.429 0.667972    
## RoofStyleGambrel     -1.950e+03  8.116e+03  -0.240 0.810162    
## RoofStyleHip          8.664e+02  6.667e+03   0.130 0.896623    
## RoofStyleMansard      1.242e+04  9.718e+03   1.278 0.201431    
## RoofStyleShed         2.226e+04  1.197e+04   1.860 0.063086 .  
## FullBath              1.946e+03  1.199e+03   1.623 0.104884    
## KitchenQualFa        -1.558e+04  3.971e+03  -3.924 9.09e-05 ***
```

```
## KitchenQualGd        -2.303e+04  2.197e+03 -10.480  < 2e-16 ***
## KitchenQualPo        -1.260e+04  1.754e+04  -0.718 0.472830
## KitchenQualTA        -2.706e+04  2.505e+03 -10.801  < 2e-16 ***
## TotRmsAbvGrd          4.565e+03  5.634e+02   8.102 1.11e-15 ***
## GarageCars            3.054e+03  9.002e+02   3.392 0.000711 ***
## SaleConditionNormal  -3.762e+03  3.385e+03  -1.111 0.266643
## SaleConditionPartial  3.196e+03  3.771e+03   0.848 0.396832
## QualityIndex          2.632e+02  6.185e+01   4.256 2.21e-05 ***
## HouseArea             5.276e+01  1.525e+00  34.595  < 2e-16 ***
## TotalSqftCalc         8.689e+00  1.203e+00   7.223 8.06e-13 ***
## price_sqft            1.353e+03  3.810e+01  35.520  < 2e-16 ***
## NbhdGrpgrp2          -4.614e+03  1.532e+03  -3.013 0.002632 **
## NbhdGrpgrp3          -9.621e+03  2.831e+03  -3.399 0.000694 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17250 on 1505 degrees of freedom
## Multiple R-squared:  0.9551, Adjusted R-squared:  0.9542
## F-statistic:  1001 on 32 and 1505 DF,  p-value: < 2.2e-16
```

3.3 Stepwise selection

Stepwise variable selection method is essentially same as forward selection method with an additional provision of being able to delete the variable selected in the previous steps. The final model with minimum AIC value for predicting the Saleprice is determined using stepwise selection method as (stepwise.lm): "SalePrice = (-141616.04) + (8.69)*TotalSqftCalc + (-15581.34)*KitchenQualFa + (-23027.75)*KitchenQualGd + (-12595.97)*KitchenQualPo + (-27058.44)*KitchenQualTA + (52.76)*HouseArea + (1353.15)*price_sqft + (-17024.5)*HouseStyle1.5Unf + (-24903.94)*HouseStyle1Story + (-8047.24)*HouseStyle2.5Fin + (2961.76)*HouseStyle2.5Unf + (11575.18)*HouseStyle2Story + (-29471.12)*HouseStyleSFoyer + (-11231.31)*HouseStyleSLvl + (4564.63)*TotRmsAbvGrd + (0.23)*LotArea + (263.22)*QualityIndex + (3053.73)*GarageCars + (-3761.94)*SaleConditionNormal + (3196.42)*SaleConditionPartial + (-4614.32)*NbhdGrpgrp2 + (-9620.7)*NbhdGrpgrp3 + (-2843.81)*RoofStyleGable + (-1950.01)*RoofStyleGambrel + (866.38)*RoofStyleHip + (12419.48)*RoofStyleMansard + (22259.53)*RoofStyleShed + (5035.33)*LotConfigCulDSac + (-4335.14)*LotConfigFR2 + (2873.82)*LotConfigFR3 + (-850.58)*LotConfigInside + (1945.87)*FullBath". The model is same as the model selected using both forward selection and backward elimination methods.

```
##
## Call:
## lm(formula = SalePrice ~ TotalSqftCalc + KitchenQual + HouseArea +
##     price_sqft + HouseStyle + TotRmsAbvGrd + LotArea + QualityIndex +
```

```
##     GarageCars + SaleCondition + NbhdGrp + RoofStyle + LotConfig +
##     FullBath, data = train.clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -106461   -8833   -1341    6304  133666
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -1.416e+05  9.615e+03 -14.728  < 2e-16 ***
## TotalSqftCalc        8.689e+00  1.203e+00   7.223 8.06e-13 ***
## KitchenQualFa       -1.558e+04  3.971e+03  -3.924 9.09e-05 ***
## KitchenQualGd       -2.303e+04  2.197e+03 -10.480  < 2e-16 ***
## KitchenQualPo       -1.260e+04  1.754e+04  -0.718 0.472830
## KitchenQualTA       -2.706e+04  2.505e+03 -10.801  < 2e-16 ***
## HouseArea            5.276e+01  1.525e+00  34.595  < 2e-16 ***
## price_sqft           1.353e+03  3.810e+01  35.520  < 2e-16 ***
## HouseStyle1.5Unf    -1.702e+04  5.076e+03  -3.354 0.000817 ***
## HouseStyle1Story    -2.490e+04  1.775e+03 -14.034  < 2e-16 ***
## HouseStyle2.5Fin    -8.047e+03  8.006e+03  -1.005 0.314985
## HouseStyle2.5Unf     2.962e+03  5.980e+03   0.495 0.620457
## HouseStyle2Story     1.158e+04  1.702e+03   6.801 1.49e-11 ***
## HouseStyleSFoyer    -2.947e+04  3.546e+03  -8.310  < 2e-16 ***
## HouseStyleSLvl      -1.123e+04  2.492e+03  -4.506 7.10e-06 ***
## TotRmsAbvGrd         4.565e+03  5.634e+02   8.102 1.11e-15 ***
## LotArea              2.273e-01  5.628e-02   4.039 5.64e-05 ***
## QualityIndex         2.632e+02  6.185e+01   4.256 2.21e-05 ***
## GarageCars           3.054e+03  9.002e+02   3.392 0.000711 ***
## SaleConditionNormal -3.762e+03  3.385e+03  -1.111 0.266643
## SaleConditionPartial 3.196e+03  3.771e+03   0.848 0.396832
## NbhdGrpgrp2         -4.614e+03  1.532e+03  -3.013 0.002632 **
## NbhdGrpgrp3         -9.621e+03  2.831e+03  -3.399 0.000694 ***
## RoofStyleGable      -2.844e+03  6.629e+03  -0.429 0.667972
## RoofStyleGambrel    -1.950e+03  8.116e+03  -0.240 0.810162
## RoofStyleHip         8.664e+02  6.667e+03   0.130 0.896623
## RoofStyleMansard     1.242e+04  9.718e+03   1.278 0.201431
## RoofStyleShed        2.226e+04  1.197e+04   1.860 0.063086 .
## LotConfigCulDSac     5.035e+03  2.033e+03   2.477 0.013371 *
## LotConfigFR2        -4.335e+03  2.877e+03  -1.507 0.132096
## LotConfigFR3         2.874e+03  7.170e+03   0.401 0.688624
## LotConfigInside     -8.506e+02  1.176e+03  -0.723 0.469545
## FullBath             1.946e+03  1.199e+03   1.623 0.104884
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17250 on 1505 degrees of freedom
## Multiple R-squared:  0.9551, Adjusted R-squared:  0.9542
## F-statistic:  1001 on 32 and 1505 DF,  p-value: < 2.2e-16
```

3.4 Junk model

Another model is fit using some random pool of 5 continuous predictor variables and let us call this a junk model(junk.lm) to compare it with other models selected using automated variable selection methods. The model is fitted using 70% of training data as: "SalePrice = (-208724.42) + (46646.33)*OverallQual + (17813.12)*OverallCond + (-3351.38)*QualityIndex + (30.59)*GrLivArea + (39.18)*TotalSqftCalc". The model captured 82.69% total variation in the sale prices with residual standard error of $33610.

```
##
## Call:
## lm(formula = SalePrice ~ OverallQual + OverallCond + QualityIndex +
##     GrLivArea + TotalSqftCalc, data = train.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -172119  -19524   -1854   15962  236699
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.087e+05  1.809e+04 -11.539  < 2e-16 ***
## OverallQual    4.665e+04  3.003e+03  15.531  < 2e-16 ***
## OverallCond    1.781e+04  3.289e+03   5.416 7.06e-08 ***
## QualityIndex  -3.351e+03  5.597e+02  -5.988 2.63e-09 ***
## GrLivArea      3.059e+01  2.918e+00  10.482  < 2e-16 ***
## TotalSqftCalc  3.918e+01  1.874e+00  20.910  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33610 on 1562 degrees of freedom
## Multiple R-squared:  0.8275, Adjusted R-squared:  0.8269
## F-statistic:  1498 on 5 and 1562 DF,  p-value: < 2.2e-16
```

3.5 Collinearity

One major area of concern in multiple variable regression is correlation/interactions between predictor variables (called collinearity) and if it is not addressed properly it can potentially lead to inaccurate/invalid model for practical purposes. Variation inflation factor (VIF) is a statistical metric calculated using each predictor as a response. If VIF of any predictor variable is high (>10 typically) that can indicate the variable is correlated with other predictor variables. The below table shows VIF values for junk model, and very high values for OverallQual, OverallCond, QualityIndex indicate that these variables are potentially interacting with other predictors in the model which indicates that the junk model would be inaccurate in predicting home price values.

VIF values for Junk model

	x
OverallQual	24.195808
OverallCond	18.778186
QualityIndex	34.822226
GrLivArea	2.892621
TotalSqftCalc	2.501365

3.5.1 VIF values for indicator variables

We do not need to be concerned about VIF values for indicator variables as we do with continuous variables as indicator variables only impact the intercept not the slope (the collinearity among continuous variables can hugely impact how the betas are estimated hence it need to be addressed seriously). High values of VIF on indicator variables may indicate potential possibility to regroup them or may suggest to build separate models for each sub category which may provoke us to revisit EDA using categorical variables. In R, since the categorical variables are automatically converted into dummy variables, the VIF values are shown at each categorical variable name level unless we manually specify the dummy variables. The below table shows VIF values calculated for forward.lm model when original Neighborhood variable (with several categories) is directly used in the model. The VIF value 30 may indicate if we sub-group some of the neighborhoods the model will be much precise and variation can be reduced (the model is fitted with original variable along with other predictors here to verify that).

VIF values for Forward selection model

	GVIF	Df	GVIF^(1/(2*Df))
HouseArea	7.440607	1	2.727748
price_sqft	3.731072	1	1.931598
HouseStyle	5.992676	7	1.136434
Neighborhood	30.002132	20	1.088752
KitchenQual	3.645786	4	1.175503
TotRmsAbvGrd	3.556237	1	1.885799
TotalSqftCalc	4.113970	1	2.028293
LotArea	1.300763	1	1.140510
QualityIndex	1.752741	1	1.323911
GarageCars	2.365609	1	1.538053
RoofStyle	1.714060	5	1.055365
SaleCondition	1.479854	2	1.102947
FullBath	2.624008	1	1.619879
LotConfig	1.246008	4	1.027875

Based on previous assignment findings the Neighborhood variable is re categorized into 3 subgroups based on price/sqft mean values in those areas. Accordingly now the VIF value for NbhdGrp is now 5.19 which is below our threshold value as shown below.

The three tables below show the VIF values for the variables picked using the three automated variable selection methods. None of them are above 10, so we are good to proceed with these models further (indicating no collinearity among the selected predictors).

VIF values for Forward selection model

	GVIF	Df	GVIF^(1/(2*Df))
HouseArea	6.611498	1	2.571283
price_sqft	6.379036	1	2.525675
HouseStyle	3.608022	7	1.095986
KitchenQual	2.615214	4	1.127687
TotRmsAbvGrd	3.437969	1	1.854176
TotalSqftCalc	3.776219	1	1.943250
LotArea	1.153050	1	1.073802
QualityIndex	1.527498	1	1.235920
GarageCars	2.246175	1	1.498724
SaleCondition	1.376004	2	1.083066
NbhdGrp	5.197597	2	1.509909
RoofStyle	1.386392	5	1.033210
LotConfig	1.143388	4	1.016890
FullBath	2.223430	1	1.491117

VIF values for Backward elimination model

	GVIF	Df	GVIF^(1/(2*Df))
LotArea	1.153050	1	1.073802
LotConfig	1.143388	4	1.016890
HouseStyle	3.608022	7	1.095986
RoofStyle	1.386392	5	1.033210
FullBath	2.223430	1	1.491117
KitchenQual	2.615214	4	1.127687
TotRmsAbvGrd	3.437969	1	1.854176
GarageCars	2.246175	1	1.498724
SaleCondition	1.376004	2	1.083066
QualityIndex	1.527498	1	1.235920
HouseArea	6.611498	1	2.571283
TotalSqftCalc	3.776219	1	1.943250
price_sqft	6.379036	1	2.525675
NbhdGrp	5.197597	2	1.509909

VIF values for Stepwise selection model

	GVIF	Df	GVIF^(1/(2*Df))
TotalSqftCalc	3.776219	1	1.943250
KitchenQual	2.615214	4	1.127687
HouseArea	6.611498	1	2.571283
price_sqft	6.379036	1	2.525675
HouseStyle	3.608022	7	1.095986
TotRmsAbvGrd	3.437969	1	1.854176
LotArea	1.153050	1	1.073802
QualityIndex	1.527498	1	1.235920
GarageCars	2.246175	1	1.498724
SaleCondition	1.376004	2	1.083066
NbhdGrp	5.197597	2	1.509909
RoofStyle	1.386392	5	1.033210
LotConfig	1.143388	4	1.016890
FullBath	2.223430	1	1.491117

The three automated variable selection methods selected same models and the estimated models are as shown below (1.forward.lm, 2.backward.lm, 3. stepwise.lm).

```
##
## Comparison of models selected through different variable selection methods
##
================================================================================
##                                                Dependent variable:
##                                    ---------------------------------------
--
##                                                      SalePrice
##                                         (1)             (2)             (3)
## ------------------------------------------------------------------------------
--
## HouseArea                                52.764          52.764          52.764
##                                         (1.525)         (1.525)         (1.525)
##                                        t = 34.595      t = 34.595      t = 34.595
##                                       p = 0.000***    p = 0.000***    p =
0.000***
##
## price_sqft                             1,353.153       1,353.153       1,353.153
##                                         (38.096)        (38.096)        (38.096)
##                                        t = 35.520      t = 35.520      t = 35.520
##                                       p = 0.000***    p = 0.000***    p =
0.000***
##
```

```
## HouseStyle1.5Unf                -17,024.500    -17,024.500    -17,024.500
##                                 (5,076.363)    (5,076.363)    (5,076.363)
##                                  t = -3.354     t = -3.354     t = -3.354
##                                 p = 0.001***   p = 0.001***   p =
0.001***
##
## HouseStyle1Story                -24,903.940    -24,903.940    -24,903.940
##                                 (1,774.517)    (1,774.517)    (1,774.517)
##                                 t = -14.034    t = -14.034    t = -14.034
##                                 p = 0.000***   p = 0.000***   p =
0.000***
##
## HouseStyle2.5Fin                 -8,047.236     -8,047.236     -8,047.236
##                                 (8,005.991)    (8,005.991)    (8,005.991)
##                                  t = -1.005     t = -1.005     t = -1.005
##                                  p = 0.315      p = 0.315      p = 0.315
##
## HouseStyle2.5Unf                 2,961.762      2,961.762      2,961.762
##                                 (5,979.676)    (5,979.676)    (5,979.676)
##                                  t = 0.495      t = 0.495      t = 0.495
##                                  p = 0.621      p = 0.621      p = 0.621
##
## HouseStyle2Story                 11,575.180     11,575.180     11,575.180
##                                 (1,701.920)    (1,701.920)    (1,701.920)
##                                  t = 6.801      t = 6.801      t = 6.801
##                                 p = 0.000***   p = 0.000***   p =
0.000***
##
## HouseStyleSFoyer                -29,471.120    -29,471.120    -29,471.120
##                                 (3,546.423)    (3,546.423)    (3,546.423)
##                                  t = -8.310     t = -8.310     t = -8.310
##                                 p = 0.000***   p = 0.000***   p =
0.000***
##
## HouseStyleSLvl                  -11,231.310    -11,231.310    -11,231.310
##                                 (2,492.299)    (2,492.299)    (2,492.299)
##                                  t = -4.506     t = -4.506     t = -4.506
##                                p = 0.00001*** p = 0.00001*** p =
0.00001***
##
## KitchenQualFa                   -15,581.340    -15,581.340    -15,581.340
##                                 (3,970.580)    (3,970.580)    (3,970.580)
##                                  t = -3.924     t = -3.924     t = -3.924
##                                 p = 0.0001***  p = 0.0001***  p =
0.0001***
##
## KitchenQualGd                   -23,027.740    -23,027.740    -23,027.740
##                                 (2,197.389)    (2,197.389)    (2,197.389)
##                                 t = -10.480    t = -10.480    t = -10.480
##                                 p = 0.000***   p = 0.000***   p =
```

```
0.000***
##
## KitchenQualPo                    -12,595.970    -12,595.970    -12,595.970
##                                  (17,541.590)   (17,541.590)
(17,541.590)
##                                   t = -0.718     t = -0.718     t = -0.718
##                                   p = 0.473      p = 0.473      p = 0.473
##
## KitchenQualTA                    -27,058.440    -27,058.440    -27,058.440
##                                  (2,505.247)    (2,505.247)    (2,505.247)
##                                  t = -10.801    t = -10.801    t = -10.801
##                                  p = 0.000***   p = 0.000***   p =
0.000***
##
## TotRmsAbvGrd                      4,564.630      4,564.630      4,564.630
##                                   (563.403)      (563.403)      (563.403)
##                                   t = 8.102      t = 8.102      t = 8.102
##                                  p = 0.000***   p = 0.000***   p =
0.000***
##
## TotalSqftCalc                        8.689          8.689          8.689
##                                     (1.203)        (1.203)        (1.203)
##                                     t = 7.223      t = 7.223      t = 7.223
##                                   p = 0.000***   p = 0.000***   p =
0.000***
##
## LotArea                              0.227          0.227          0.227
##                                     (0.056)        (0.056)        (0.056)
##                                     t = 4.039      t = 4.039      t = 4.039
##                                  p = 0.0001***  p = 0.0001***  p =
0.0001***
##
## QualityIndex                       263.225        263.225        263.225
##                                    (61.854)       (61.854)       (61.854)
##                                    t = 4.256      t = 4.256      t = 4.256
##                                 p = 0.00003*** p = 0.00003*** p =
0.00003***
##
## GarageCars                         3,053.725      3,053.725      3,053.725
##                                    (900.151)      (900.151)      (900.151)
##                                    t = 3.392      t = 3.392      t = 3.392
##                                   p = 0.001***   p = 0.001***   p =
0.001***
##
## SaleConditionNormal                -3,761.936     -3,761.936     -3,761.936
##                                   (3,385.351)    (3,385.351)    (3,385.351)
##                                    t = -1.111     t = -1.111     t = -1.111
##                                    p = 0.267      p = 0.267      p = 0.267
##
## SaleConditionPartial               3,196.422      3,196.422      3,196.422
```

```
##                                (3,771.436)    (3,771.436)    (3,771.436)
##                                 t = 0.848      t = 0.848      t = 0.848
##                                 p = 0.397      p = 0.397      p = 0.397
##
## NbhdGrpgrp2                     -4,614.324     -4,614.324     -4,614.324
##                                (1,531.576)    (1,531.576)    (1,531.576)
##                                 t = -3.013     t = -3.013     t = -3.013
##                                p = 0.003***   p = 0.003***   p =
0.003***
##
## NbhdGrpgrp3                     -9,620.696     -9,620.696     -9,620.696
##                                (2,830.606)    (2,830.606)    (2,830.606)
##                                 t = -3.399     t = -3.399     t = -3.399
##                                p = 0.001***   p = 0.001***   p =
0.001***
##
## RoofStyleGable                  -2,843.809     -2,843.809     -2,843.809
##                                (6,628.650)    (6,628.650)    (6,628.650)
##                                 t = -0.429     t = -0.429     t = -0.429
##                                 p = 0.668      p = 0.668      p = 0.668
##
## RoofStyleGambrel                -1,950.011     -1,950.011     -1,950.011
##                                (8,116.293)    (8,116.293)    (8,116.293)
##                                 t = -0.240     t = -0.240     t = -0.240
##                                 p = 0.811      p = 0.811      p = 0.811
##
## RoofStyleHip                     866.381        866.381        866.381
##                                (6,666.987)    (6,666.987)    (6,666.987)
##                                 t = 0.130      t = 0.130      t = 0.130
##                                 p = 0.897      p = 0.897      p = 0.897
##
## RoofStyleMansard                12,419.480     12,419.480     12,419.480
##                                (9,717.573)    (9,717.573)    (9,717.573)
##                                 t = 1.278      t = 1.278      t = 1.278
##                                 p = 0.202      p = 0.202      p = 0.202
##
## RoofStyleShed                   22,259.530     22,259.530     22,259.530
##                                (11,967.740)   (11,967.740)
(11,967.740)
##                                 t = 1.860      t = 1.860      t = 1.860
##                                 p = 0.064*     p = 0.064*     p = 0.064*
##
## LotConfigCulDSac                 5,035.329      5,035.329      5,035.329
##                                (2,033.121)    (2,033.121)    (2,033.121)
##                                 t = 2.477      t = 2.477      t = 2.477
##                                p = 0.014**    p = 0.014**    p = 0.014**
##
## LotConfigFR2                    -4,335.139     -4,335.139     -4,335.139
##                                (2,877.227)    (2,877.227)    (2,877.227)
##                                 t = -1.507     t = -1.507     t = -1.507
```

```
##                                    p = 0.133      p = 0.133      p = 0.133
##
## LotConfigFR3                       2,873.822      2,873.822      2,873.822
##                                   (7,170.201)    (7,170.201)    (7,170.201)
##                                    t = 0.401      t = 0.401      t = 0.401
##                                    p = 0.689      p = 0.689      p = 0.689
##
## LotConfigInside                     -850.582       -850.582       -850.582
##                                   (1,175.807)    (1,175.807)    (1,175.807)
##                                    t = -0.723     t = -0.723     t = -0.723
##                                    p = 0.470      p = 0.470      p = 0.470
##
## FullBath                           1,945.873      1,945.873      1,945.873
##                                   (1,199.229)    (1,199.229)    (1,199.229)
##                                    t = 1.623      t = 1.623      t = 1.623
##                                    p = 0.105      p = 0.105      p = 0.105
##
## Constant                          -141,616.000   -141,616.000   -
141,616.000
##                                   (9,615.115)    (9,615.115)    (9,615.115)
##                                   t = -14.728    t = -14.728    t = -14.728
##                                   p = 0.000***   p = 0.000***   p =
0.000***
##
## ------------------------------------------------------------------------
--
## Observations                         1,538          1,538          1,538
## R2                                   0.955          0.955          0.955
## Adjusted R2                          0.954          0.954          0.954
## Residual Std. Error (df = 1505)   17,251.600     17,251.600     17,251.600
## F Statistic (df = 32; 1505)      1,000.730***   1,000.730***
1,000.730***
##
========================================================================
## Note:                                          *p<0.1; **p<0.05;
***p<0.01
```

3.6 Model Comparison

As shown above, the four final models are compared against each other in terms of in-sample model fit (1st step in model validation is making sure in-sample fit metrics make sense before proceeding with out of sample predictions) and predictive accuracy (Mean squared error(MSE) and Mean absolute errors(MAE) are more intuitive metrics to measure how good the model is predicting home-price values, where as adjusted r-sq, AIC, BIC metrics address capturing as much variation as possible there by measuring model fit with simplest model possible – principle of parsimony to balance between accuracy and precision.).

The below table presents these metrics for all the four models. The mean absolute error of junk model is $23784 which means the home values could be predicted with an error of +/- $23784 on average. The MAE of other three models is $11493 which seems pretty good on average without acknowledging practical requirement at this time. The Adjusted R-sq value for junk model is 82.69% but as we saw through VIF values it has some collinearity issues which need to be addressed. The adjusted r-sq values for all the other three models is 95.42% which indicates most of the variation in sale prices in training data is captured by these models. The models are also ranked based on each individual metrics as shown in the following table. Junk model is ranked 2 in all the 5 metrics and other three are ranked 1 as they all converged to the same model. Each metric may not necessarily give same rank depending on the model selected by each method (depending on the data sample also each metric can perform differently in each method).

Model Fit Metrics Comparison

Model	adj.r.squared	AIC	BIC	MSE	MAE
Junk	0.8269	37143.00	37180.50	1125317620	23784.26
Forward Selection	0.9542	34407.71	34589.21	291231763	11493.46
Backward Elimination	0.9542	34407.71	34589.21	291231763	11493.46
Stepwise	0.9542	34407.71	34589.21	291231763	11493.46

Model Fit Metrics Ranks

Model	Rank.adjR	Rank.AIC	Rank.BIC	Rank.MSE	Rank.MAE
Junk	2	2	2	2	2
Forward Selection	1	1	1	1	1
Backward Elimination	1	1	1	1	1
Stepwise	1	1	1	1	1

Section 4: Predictive Accuracy

MAE and MSE are two commonly used metrics to measure models predictive accuracy. MAE measures average magnitude of errors in predictions without considering their direction (absolute). MAE is more relevant when all individual predictive differences have equal weight whereas MSE is more useful when large errors are not desirable (as the errors are squared before they are averaged). In our case, variance in the errors is of more relevant meaning how close the predictions are to the actual value unless there is a pattern observed in residuals, so MAE is of more interest here. The predictive model performance is evaluated by testing the models on out of sample data using cross-validation techniques. If the model has better in-sample predictive accuracy than out of sample means there may be issues with out of sample data (meaning it may not belong to the population sample data that is used to fit the model or right sample is not used in fitting the model). In general in-sample accuracy will be better than out of sample as the model has seen all the in-sample data while fitting the model where the errors are minimized. Out of sample data may not exactly fall into the operating region of in sample data thereby possibility of predicting the response with more error.

30% test data is used to predict HomeSale Prices using all 4 models and in sample (70% data) and out of sample (30% data) MAE and MSE are presented below. The MAE for junk model if $23967 which is not much different from in sample mean absolute error and out of sample MAE for other models is $11630 which is also very close to in sample error. Similarly the MSE is not much different between both the samples between models. In reality this may not be the case depending on the nature of the data. Here we only performed cross validation using one 70-30 split but if we performed n-fold cross validation or other types of sampling these may differ. So exhaustive testing is very important as part of validation before putting models into practice.

Prediction metrics Comparision between models

Model	MAE.In.Sample	MAE.Out.of.Sample	MSE.In.Sample	MSE.Out.of.Sample
Junk	23784.26	23967.28	1125317620	1172647802
Forward	11493.46	11630.56	291231763	304344213
Backward	11493.46	11630.56	291231763	304344213
Stepwise	11493.46	11630.56	291231763	304344213

Section 5: Operational Validation

The errors in statistical sense indicate that the model selected using automated variable selection is best model compared to junk model (since all three methods converged to the same model we will focus only on junk model and forward selection model in this section). The MAE of $11630 for predictions seems pretty good relative to the typical home prices in Iowa but in order to put model into practice we need to translate these errors into business thresholds and make sure business requirement is met before saying that the model is valid. As discussed above the errors in statistical sense do not give us fair idea about the

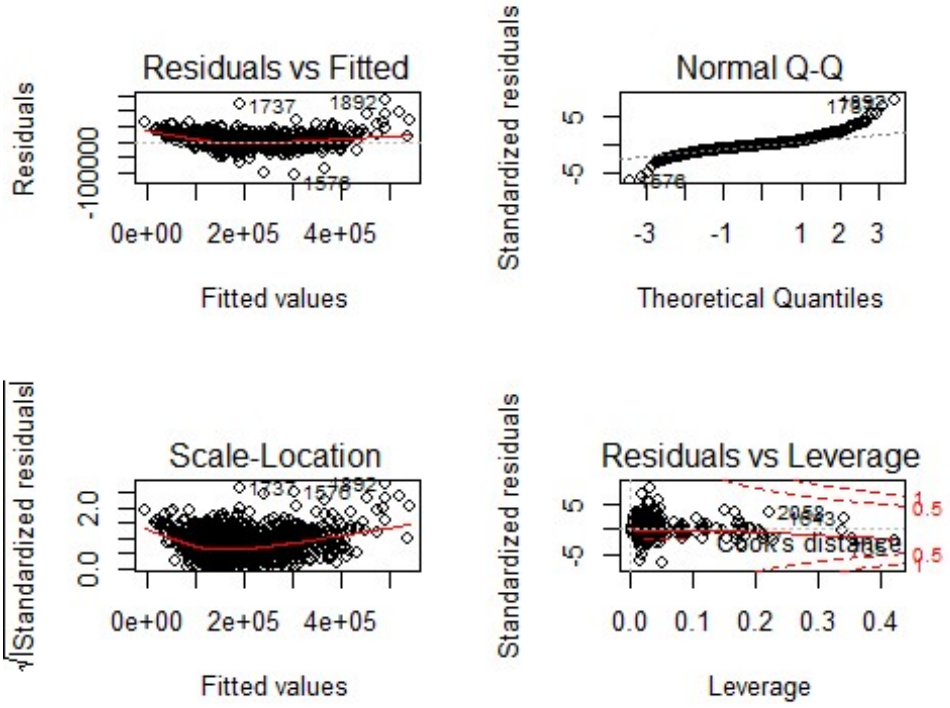
distribution of errors. The predictive error rates with respect to sale prices are divided into 4 grades (Grade 1: if error is 0 to 10% of sale price, Grade 2: if error is 10-15% of sale price, Grade 3: if error is 15-25% of sale price, Grade 4: if error is >25% of sale price). GSEs rate an AVM model as 'underwriting quality' if the model is accurate to within 10% more than 50% of the time. The below table shows performances of forward and junk models in terms of percentage of observations in all 4 grades for both in-sample and out of sample data. Junk model is within 10% only 49-50% of the time using in sample or out of sample data. The forward model's error is within 10% of sale price around 80% of times both within and out of sample data. Hence we can conclude that the model selected using any automated variable selection method is 'best' model in our case for predicting home prices in Iowa. The model ranking remained same as predictive accuracy results, i.e., junk model performed poorly in terms of both metrics.

Operational Validation

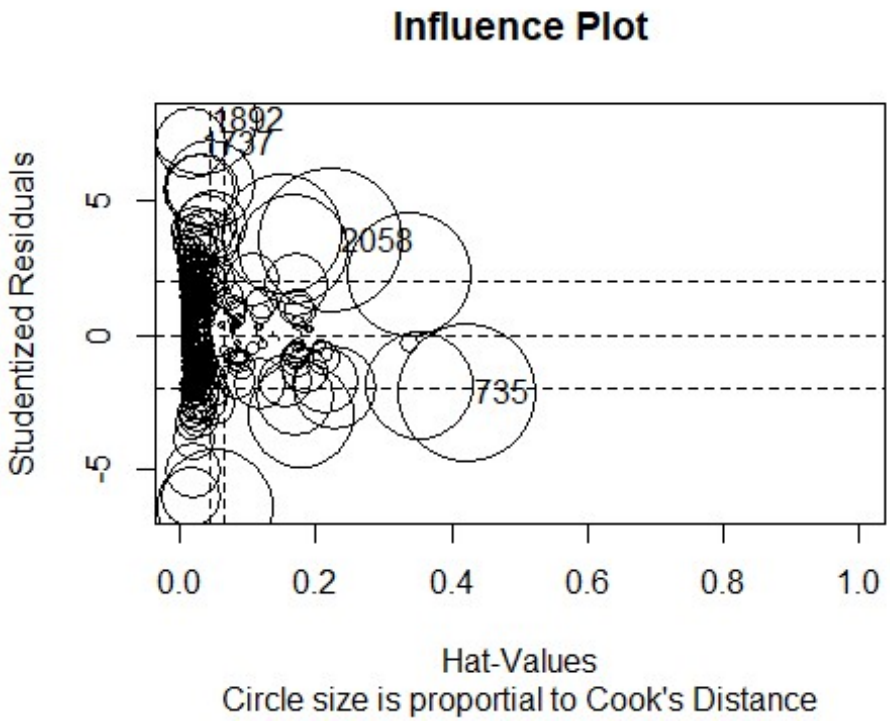
Grade	forward.train.grade	forward.test.grade	junk.train.grade	junk.test.grade
Grade 1: [0.0, 0.1]	81%	80%	49%	50%
Grade 2: [0.10,0.15]	9%	9%	18%	19%
Grade 3: [0.15, 0.25]	7%	6%	19%	17%
Grade 4: [0.25 +]	3%	5%	14%	14%

Section 6: Best Model

Based on model fit statistics and predictive accuracy metrics using 70-30 train test splits, we concluded that the model selected using any of the automated variable selection methods is the best model (as all three converged to the same model). It is time to re-visit the model to perform diagnostics on residuals to make sure that the OLS assumptions are holding true (1. Residuals are normal, 2.Residuals are homoscedastic). The below diagnostic plots indicate that the residuals do not seem to have constant variation throughout the range of SalePrices (somewhat u shaped pattern is observed) and also q-q plot doesn't look quite normal at the ends. These may indicate the model is not predicting low and high priced homes as accurately as the middle range home values are predicted.



Given the patterns in residuals, we can also suspect that there are outliers in the sample data (which may not belong to the typical home values) which if removed can improve the model performance tremendously. The influence plot below shows that there are several influential points represented by big circles in our data, if these are removed based on DFFITs statistics, it may have huge positive impact on the model.



After observing the diagnostic plots we can conclude that the model is not satisfying all OLS assumptions and need to be revisited where the variables may need transformation or some outliers may need to be removed or more EDA need to be conducted to select appropriate variables/regroup categorical variables/calculate new continuous variables etc. This cycle can continue until the point OLS assumptions and predictive accuracy requirements are met.

In our case, the goal is to be able to predict home sale prices within 10% error more than 50% of the times based on 70-30% cross-validation scheme and as for predictive purposes we can live with slight violations in OLS assumptions, so we conclude that the model developed using automated variable selection methods is the 'best' model in our case– this forward.lm model captured 95.42% variation in saleprices and achieved within 10% of saleprice error more than 80% of time using both train and test data. The final 'best' model is reported with pride below:

SalePrice = (-141616.04) + (52.76)*HouseArea + (1353.15)*price_sqft + (-17024.5)*HouseStyle1.5Unf + (-24903.94)*HouseStyle1Story + (-8047.24)*HouseStyle2.5Fin + (2961.76)*HouseStyle2.5Unf + (11575.18)*HouseStyle2Story + (-29471.12)*HouseStyleSFoyer + (-11231.31)*HouseStyleSLvl + (-15581.34)*KitchenQualFa + (-23027.75)*KitchenQualGd + (-12595.97)*KitchenQualPo + (-27058.44)*KitchenQualTA + (4564.63)*TotRmsAbvGrd + (8.69)*TotalSqftCalc + (0.23)*LotArea + (263.22)*QualityIndex + (3053.73)*GarageCars + (-3761.94)*SaleConditionNormal + (3196.42)*SaleConditionPartial + (-

4614.32)*NbhdGrpgrp2 + (-9620.7)*NbhdGrpgrp3 + (-2843.81)*RoofStyleGable + (-1950.01)*RoofStyleGambrel + (866.38)*RoofStyleHip + (12419.48)*RoofStyleMansard + (22259.53)*RoofStyleShed + (5035.33)*LotConfigCulDSac + (-4335.14)*LotConfigFR2 + (2873.82)*LotConfigFR3 + (-850.58)*LotConfigInside + (1945.87)*FullBath

Conclusion:

There are several take-aways from this analysis based on all 4 assignments.

Challenges presented by the data:

- The data has 80 predictors to choose from which has been quite challenging given the skewed distributions of continuous variables including the sale price response variable and imbalanced categories in categorical variables
- The sample definition is the key which depended on subject matter expertise to define the population
- There are so many ways to perform EDA and depending on analyst's subject matter expertise, judgement and visual cues it can lead to selection of predictor variables which may or may not be the right variables, while the statistical plots/metrics can provide guidance to the analyst practical feasibility or significance of the predictors is equally important in selecting the variables
- There is no rule of thumb for specifying a model, it is an iterative process where EDA need to be revisited at every step of model building process (from sample definition to EDA to model specification to model adequacy checking to validation to practical implementation)
- Interpretation of the models need to be done carefully when categorical variables are present. Including categorical variables is advantageous as they can guide us to decide if we need more than one model for solution
- When transformations are used, we need to retransform the model outcome to make business sense otherwise the interpretations using transformed quantities wouldn't make sense (e.g., log(saleprice))
- While the automated variable selection methods are very handy, there are several caveats associated with it which may not be obvious until diagnostics are performed using residual plots
- Model validation need to be done at every step of model building process
- Understanding the business requirement, data constraints, model constraints is very important for the analyst to come up with feasible model while addressing the caveats as much as possible
- Ideally transformed variables, calculated variables, raw variables all can to be used to come up with best solution but the principle of parsimony says that the model

need to be as simple as possible while capturing as much variation as possible (striking right balance between accuracy and precision). So using predictors in their raw form is most preferable way.

Recommendations for improving predictive accuracy:

- In our case, the practical requirement of within 10% error more than 50% of the time is met with model selected using automated variable selection methods but as observed through diagnostic plots, OLS assumptions are violated so the model need to be revisited – SalePrice response variable can be transformed using log transform to address heteroscedasticity, outliers need to be investigated and removed if possible, the model looks complex with number of variables selected, so we can look for ways to simplify it using different predictors (transformed, regrouped, recalculated) which warrants revisiting EDA. Where to draw line depends on what violations we can live with and what is the practical requirement of accuracy.

- Even though OLS assumptions are violated, the predictive accuracy of our model is very high using 70-30% cross-validation, so we may be able to live with the violation as long as the practical requirement is met (i.e., is high errors in low and high value homes ok?). The model need to be validated exhaustively using n-fold cross validation or different sampling techniques to make sure that same performance holds in every fold, before putting the model into practice.