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

Harini K. Anand

1. Introduction

Models for accurate prediction of housing prices are in high demand by homeowners, home sellers, mortgage lenders, building developers, tax assessment offices, insurance companies, etc. It is because home buying is a costly investment.

In this report, we present the results of the model identification done for the prediction of home sale prices with the use of automated variable selection methods on the Ames, Iowa Housing dataset from DeCock (2011). The purpose of the assignment is to demonstrate the utilization of the computational exploratory tools for model identification when the datasets are large both in terms number of observations and number of columns (dimensions). The analysis was conducted by following the predictive modeling framework to create in-sample and out-of-sample data partitions through random sampling of the data set. The report is structured to present the analysis of the sample creation followed by the generation of the train and test data partitions and then the results of models selected by the automatic variable selection methods (methods used were forward, backward, and stepwise variable selection). After that, the results of the model comparison using model-fit, predictive accuracy, and application-specific predictive accuracy metrics conducted using both the in-sample and the out-of-sample data are presented. As part of the analysis, the presence of multicollinearity in the data was studied along with its effects on the models.

2. Data

The dataset used contains the assessed values for individual residential properties sold in Ames, IA from 2006 to 2010. The data dictionary for the dataset is available at this location: http://www.amstat.org/publications/jse/v19n3/decock/DataDocumentation.txt. Per the data dictionary, the dataset has 2,930 observations. It contains 82 variables (23 nominal, 23 ordinal, 14 discrete, and 20 continuous variables, and 2 observation identifiers).

3. Sample Definition and Data Split

3.1. Sample Definition

The goal of the assignment was to build and evaluate linear regression models for the prediction of the price of a ‘typical’ home. Towards that goal, we have chosen to exclude any observations in the population of apartments, condominiums, commercial properties, farm properties, and industrial properties (the properties that are not in the residential zones) as these properties present huge variations in the sale prices. For similar reasons, observations that are not “normal” sales (such as sales between family members, abnormal sales, or partial sales) and building types that are not single-family homes (such as townhomes, duplex, 2 family conversions) were excluded.

We also excluded any observations of ‘atypical’ homes such as the ones with no paved street access, basements (the vast majority of houses in the mid-west have basements to protect against freeze damage) or no water or just ELO (Electricity only) (since without these utilities, homes would be uninhabitable) or with miscellaneous features (such as an elevator, tennis court, shed, etc). Due to the housing boom in the 1950s, huge differences are noted in the sale prices of homes built before 1950 compared to those built after 1950. So, for this assignment, the regression models were developed to predict the prices of houses built only after 1950. Consequently, the observations of houses built before 1950 are excluded from the sample. The excluded observations do not reflect a 'typical' home, and the variations in the data they present could achieve poor results for the model.

Additionally, we also examined the special notes in the data dictionary which states that houses with more than 4000 square feet should be removed from the observations due to the sale price not representing the actual market values and due to those observations being from unusual sales. These observations were excluded as they are outliers and would affect the model.

Drop conditions

Based on the above discussion of the appropriate sample population for the problem to predict the value of a house in Ames, Iowa, the below exclusion rule set (drop conditions) is defined:

a. ***01: Not Residential:*** Excluded observations that are not in residential zones (observations with MS Zoning values A, C, I, or FV)

b. ***02: Non-Normal Sale:*** Excluded observations with any sale condition value that is other than "Normal" (observations with Sale Condition that is not 'Normal')

c. ***03: Not Single-Family Home:*** Excluded observations with building types that are other than "Single-family" (observations with Bldg Type values '2FmCon', 'Duplx', 'TwnhsE', or 'TwnhsI')

d. ***04: Street Not Paved:*** Excluded observations that do not have "paved" road access to property (observations with Street value 'Gravel')

e. ***05: No Water:*** Excluded observations with utilities that do not include water (observations with Utilities values 'NoSeWa' or 'ELO')

f. ***06: Built Pre-1950:*** Excluded observations of houses built before the year 1950 (observations with Year Built value < 1950)

g. ***07: No Basement:*** Excluded observations with no basements (observations with BsmtFin Typ1 1 value 'No Basement')

h. ***08: LT 800 OR GT 4000 SqFt***: Excluded observations with ground living area of less than 800 sq.ft or more than 4000 sq.ft (observations with Gr Liv Area with value < 800 sq. ft or > 4000 sq.ft)

i. ***09: MiscVal > 0:*** Excluded houses with miscellaneous features (observations with MiscVal > 0)

Waterfall Drop Condition	Count
01: Not Residential	168
02: Non-Normal Sale	457
03: Not Single Family Home	362
04: Street Not Paved	2
06: Built Pre-1950	480
07: No Basement	27
08: LT 800 OR GT 4000 SqFt	10
09: MiscVal>0	65
99: Eligible Sample	1359

Table 01: Distribution table of the records excluded from the eligible sample based on the drop conditions. Also included is the size of the eligible Sample.

From Table 01, we can note that a total of 1571 observations were excluded from the original dataset. There is one observation in the dataset with Utilities value of ‘NoSeWa’ (No Sewage or Water). However, this observation is also for a Non-Normal sale. As a result, it was excluded as part of the second condition. After the exclusions using the waterfall drop conditions were applied, the eligible sample reduces to 1359 observations.

Code Snippet 01 in section 8.1 contains the R code for the creation of the eligible sample population.

3.2. The Train/Test Split

The model results in this report are generated by following the predictive modeling framework. As part of that, the sample generated using the drop conditions in section 3.1 are split using random sampling into two data sets – one for in-sample model development and one for out-of-sample model assessment. We perform a 70/30 training/test split which is the basic form of cross-validation. Cross-validation testing helps to evaluate predictive models by determining the test error and identify the problems like overfitting.

Prior to the split, any observations with missing values in the variables of interest were removed. This reduced the sample size by 314 observations from 1359 down to 1045 observations.

Table 02 shows the number of observations in the training and test data partitions obtained after splitting the sample (with missing values removed).

Data Partition	Number of observations in the partition	Percentage of the total observations
Training	742	71.01%
Test	303	28.99%

Table 02: Observation counts of the 70/30 training/test data partition of the sample.

Code Snippet 02 in section 8.2 (page 28) contains the R code for the creation of the training/test data split from the sample.

Note: Additional continuous variables and the indicator/dummy variables for the discrete variables for use in the model identification were added to the sample prior to performing the train/test data split.

4. Model Identification and In-Sample Model Fit

In this selection, we present the results of the automated variable selection (using the methods of forward variable selection, backward variable selection, and stepwise variable selection) in the identification of a model for the prediction of the price of a house.

For the variable selection, we first picked out 20 candidate predictor variables. Specific attention was given to ensure all the variables in the candidate list are not orthogonal. The candidate pool was picked to contain some similar variables such TotRmsAbvGrd and BedRoomAbvGr which have a correlation coefficient of 0.627. Additionally, we included some variables such as ThreeSsnPorch, Fireplaces, and PoolArea that are intuitively not good predictor variables of SalePrice.

There are 14 continuous (or approximately continuous) and 6 discrete variables in the candidate pool list. The 6 discrete variables were coded into a group of indicator variables. The categories created for each of the 6 discrete variables were

Fireplaces:

Data for this variable ranges from 0 to 4 fireplaces. We defined 3 categories for this variable:

Category 1: Houses with 0 fireplaces (Base Category)
Category 2: Houses with 1 or 2 fireplaces
Category 3: Houses with 3 or more fireplaces

YearBuilt:

Due to the Drop condition for the eligible population, the observations contain only houses built between 1950 to 2010. We defined 3 categories for this variable. This is based on the distribution table where we noticed an increase in the number of houses built from 1954 to 1969 followed a decrease in the number of houses built in the 1970s and 1980s, and an increase in the number of houses built from 1992 to 2008.

Category 1: Houses built between 1950 and 1969 (Base Category)
Category 2: Houses built between 1970 and 1989
Category 3: Houses built between 1990 and 2010

GarageCars:

Data for this variable ranges from 0 to 4 cars in garage capacity. We defined 3 categories for this variable:

Category 1: Houses with 0 car capacity in the garage (Base Category)
Category 2: Houses with 1 or 2 car capacity in the garage
Category 3: Houses with 3 or more car capacity in the garage

TotRmsAbvGrd:

Data for this variable ranges from 3 to 12 rooms above ground. We defined 3 categories for this variable:

Category 1: Houses with 3 or 4 rooms (Base Category)
Category 2: Houses with 5 to 9 rooms
Category 3: Houses with 10 or more rooms

MoSold:

Data for this variable ranges from 1 (Jan) to 12 (Dec). We defined 3 categories for this variable:

Category 1: Houses sold in Jan or Feb (Base Category)
Category 2: Houses sold between Mar and Aug
Category 3: Houses sold between Sept and Dec

BedroomAbvGr:

Data for this variable ranges from 0 to 5. We defined 3 categories for this variable:

Category 1: Houses with 0 or 1 bedrooms (Base Category)
Category 2: Houses with 2 to 4 bedrooms
Category 3: Houses with more than 5 bedrooms

Table 03 lists the 20 variables in the candidate pool used in the variable selection along with their data type and the description.

No.	Variable Name	Data type	Description
1	LotFrontage	Continuous	Linear feet of street connected to property
2	LotArea	Continuous	Lot size in square feet
3	TotalBsmtSF	Continuous	Total square feet of basement area
4	GarageArea	Continuous	Size of garage in square feet
5	ScreenPorch	Continuous	Screen porch area in square feet
6	MasVnrArea	Continuous	Masonry veneer area in square feet
7	WoodDeckSF	Continuous	Wood deck area in square feet
8	OpenPorchSF	Continuous	Open porch area in square feet
9	EnclosedPorch	Continuous	Enclosed porch area in square feet
10	PoolArea	Continuous	Pool area in square feet
11	ThreeSsnPorch	Continuous	Three season porch area in square feet
12	TotalSqftCalc	Continuous	Computed as the sum of the variables BsmtFinSF1 + BsmtFinSF2 + GrLivArea The total square footage of the house.
13	HomeSize	Continuous	Computed as the sum of the variables FirstFlrSF + SecondFlrSF The sum of the first and second floor areas.
14	QualityIndex	Approximately Continuous	Computed as the product of the OverallQual (Overall quality of the house) and the OverallCond (Overall condition of the house) OverallQual * OverallCond
15	Fireplaces	Discrete	Number of fireplaces Created 3 indicator variables for the categories described above.
16	YearBuilt	Discrete	Original construction date Created 3 indicator variables for the categories described above
17	GarageCars	Discrete	Size of garage in car capacity Created 3 indicator variables for the categories described above
18	TotRmsAbvGrd	Discrete	Total rooms above grade (does not include bathrooms) Created 3 indicator variables for the categories described above
19	MoSold	Discrete	Month Sold (MM) Created 3 indicator variables for the categories described above
20	BedroomAbvGr	Discrete	Bedrooms above grade (does NOT include basement bedrooms) Created 3 indicator variables for the categories described above

Table 03: List of candidate predictor variables for evaluation using the automated variable selection methods.

To perform the variable selection effectively and easily, any unwanted variables and base category indicator variables were removed from the training and test data partitions.

Code Snippet 02 in section 8.2 (pages 27, 28) contains the R code for the creation of the three continuous variables TotalSqftCalc, QualityIndex, HomeSize, and for the creation of the indicator/dummy variables for the discrete variables.

4.1. Forward Variable Selection

For the forward variable selection, we utilized the stepAIC function in R with “forward” direction and a starter regression equation containing no predictor variables (Intercept-only model). To ensure the search is a full or exhaustive search, we defined the scope of the search to include the full model containing every predictor variable in the candidate pool and also the intercept-only model. The stepAIC function performed 18 iterations based on the AIC value. It picked out a regression equation with 18 terms (17 predictor variables and an intercept term).

The fitted regression equation for the model selected by the forward variable selection is

SalePrice = - 21,116.750 + 28.730 * HomeSize + 31.900 * TotalBsmtSF + 31,627.120 * GarageCarsCat3 + 1,662.643 * QualityIndex + 24.601 * TotalSqftCalc + 32,816.850 * YearBuiltCat3 + 1.954 * LotArea + 36,675.590 * TotRmsAbvGrdCat3 + 34.252 * MasVnrArea -30,849.140 * BedroomAbvGrCat2 -37,407.890 * BedroomAbvGrCat3 + 43.557 * ScreenPorch + 4,307.927 * YearBuiltCat2 + 10.798 * GarageArea + 10.991 * WoodDeckSF + 25.143 * OpenPorchSF + 2,932.676 * MoSoldCat2

On the predictor variables picked out by stepAIC function using forward variable selection, we then computed the VIF values (Variance Inflation Factors) to identify if there is any multicollinearity among the predictor variables. Table 04 shows the VIFs computed for the 17 predictor variables in the descending order of the value.

Predictor Variable Name	VIF value
HomeSize	4.102371
TotalSqftCalc	3.650919
GarageCarsCat3	2.481769
GarageArea	2.446537
YearBuiltCat3	2.035002
TotalBsmtSF	1.851932
BedroomAbvGrCat3	1.731164
BedroomAbvGrCat2	1.675785
MasVnrArea	1.592965
QualityIndex	1.458150
YearBuiltCat2	1.356379
TotRmsAbvGrdCat3	1.321570
LotArea	1.293311
WoodDeckSF	1.289029
OpenPorchSF	1.233304
ScreenPorch	1.096863
MoSoldCat2	1.021262

Table 04: List of VIFs values in descending order for the predictor variables identified by using the forward selection

None of the computed VIF values listed in Table 04 are large values (in excess of 20) indicating collinearity is not a problem among the variables selected by the forward selection algorithm**. *As a result, none of the predictor variables are removed.***

	Dependent Variable - SalePrice
Constant	-21,116.750** (9,433.428)
HomeSize	28.730*** (3.603)
TotalBsmtSF	31.900*** (2.986)
GarageCarsCat3	31,627.120*** (3,636.783)
QualityIndex	1,662.643*** (146.026)
TotalSqftCalc	24.601*** (2.266)
YearBuiltCat3	32,816.850*** (2,476.583)
LotArea	1.954*** (0.211)
TotRmsAbvGrdCat3	36,675.590*** (5,321.358)
MasVnrArea	34.252*** (5.645)
BedroomAbvGrCat2	-30,849.140*** (6,474.304)
BedroomAbvGrCat3	-37,407.890*** (11,043.030)
ScreenPorch	43.557*** (15.154)
YearBuiltCat2	4,307.927 (2,760.384)
GarageArea	10.798 (7.462)
WoodDeckSF	10.991 (7.283)
OpenPorchSF	25.143 (16.922)
MoSoldCat2	2,932.676 (1,983.580)

Observations	742
R2	0.914
Adjusted R2	0.912
Residual Std. Error	23,610.780 (df = 724)
F Statistic	450.635*** (df = 17; 724)

Note: *p<0.1; **p<0.05; ***p<0.01

Table 05: Regression output table using stargazer for the final estimated model using the Forward Variable Selection (the coefficient and the standard error (in blue) are listed against each variable)

Based on the output in Table 05, we conclude that at a level of significance of 0.001, the overall regression is strongly statistically significant with a p-value < 2.2e-16 (with a F-statistic value of 450.635).

So, at a level of significance of 0.001, we can reject the joint null hypothesis that all the predictor variables have a zero coefficient and have no effect on the regression.

Among the strongly statistically significant variables, as can be noted from Table 05, GarageCarCat3 (houses with 3 or more car capacity), YearBuiltCat2 (house built between 1970 and 1989), YearBuiltCat3 (house built between 1990 and 2010), TotRmsAbvGrdCat3 (category for houses with 10 or more total rooms), BedroomAbvGrCat2 (category for houses with 2 to 4 bedrooms) and BedroomAbvGrCat3 (category for houses with more than 5 bedrooms) have large standard errors (> 1000) for the coefficients.

From Table 05, we can note that given all else in the model is held constant, a unit increase in the HomeSize (sum of FirstFlrSF and SecondFlSF) will result in an increase of $28.730 in the Sale Price. A unit increase in TotalBsmtSF results in an increase of $31.90 in the Sale Price given everything else about the model is held constant. Similarly, while holding other variables fixed, a house with a 3 or more car capacity garage results in an increase of $31,627.120 in the Sale Price than the base category (with no garage).

We also gather from Table 05 that a house built between 1990 and 2010 will result in an increase of $32,816.850 compared to the base category (a house built between 1950 and 1969). A unit increase in the QualityIndex results in an increase of $ 1,662.643 in the Sale Price with everything else held constant. A unit increase of TotalSqftCalc and LotArea result in an increase of $24.601 and $1.954 in the Sale price respectively given all else in the model is held constant. Also, a house with 10 or more rooms results in an increase of $36,675.590 in Sale Price compared to the base category (house with 3 or 4 rooms) when everything is fixed. A unit increase of MasVnrArea increases the SalePrice by $34.252 under the condition all else in the model is constant. A unit increase in ScreenPorch area will provide an increase of $43.557 to the SalePrice given all else in the model is held constant.

The intercept shows the expected mean of the SalePrice is negative (-21,116.750) when all the predictor variables are zero. Interestingly, the regression coefficients for BedroomAbvGrCat2 and BedroomAbvGrCat3 were negative.

The coefficient of determination (Adjusted R-squared) value of 0.912 indicates approximately 91.2% of the variation in the SalePrice can be explained by variation in the predictor variables listed in Table 05.

From Table 05 we can also note that the following predictor variables in the candidate pool were not selected by the forward variable selection algorithm:

- LotFrontage - Linear feet of street connected to property
- EnclosedPorch - Enclosed porch area in square feet
- ThreeSsnPorch - Three season porch area in square feet
- PoolArea - Pool area in square feet
- Fireplaces – Number of Fireplaces
- GarageCarsCat2 – category of the GarageCars discrete variable representing 1 or 2 car garages
- TotRmsAbvGrdCat2 – category of the TotRmsAbvGrd discrete variable representing 5 to 9 rooms

- MoSoldCat3 – category of the MoSold discrete variable representing the months Sept thru Dec regarding when the house was sold

Term	t value	Pr(>|t|)
(Intercept)	-2.239	0.025491 *
HomeSize	7.974	6.01e-15 ***
TotalBsmtSF	10.685	< 2e-16 ***
GarageCarsCat3	8.696	< 2e-16 ***
QualityIndex	11.386	< 2e-16 ***
TotalSqftCalc	10.855	< 2e-16 ***
YearBuiltCat3	13.251	< 2e-16 ***
LotArea	9.281	< 2e-16 ***
TotRmsAbvGrdCat3	6.892	1.20e-11 ***
MasVnrArea	6.068	2.09e-09 ***
BedroomAbvGrCat2	-4.765	2.28e-06 ***
BedroomAbvGrCat3	-3.387	0.000744 ***
ScreenPorch	2.874	0.004169 **
YearBuiltCat2	1.561	0.119049
GarageArea	1.447	0.148313
WoodDeckSF	1.509	0.131726
OpenPorchSF	1.486	0.137757
MoSoldCat2	1.478	0.139715

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Table 06: Table shows the t-values and p-values for the individual coefficient tests on the terms in the model selected by Forward Variable Selection

In Table 06 we noted that, among the coefficient T-test results, only the coefficients for the predictor variables HomeSize, TotalBsmtSF, GarageCarsCat3, QualityIndex, TotalSqftCalc, YearBuiltCat3, LotArea, TotRmsAbvGrdCat3, MasVnrArea, BedroomAbvGrCat2, and BedroomAbvGrCat3 are strongly statistically significant at a level of significance of 0.001. The coefficient for ScreenPorch is only significant at a level of significance of 0.01. The intercept is significant at a level of significance of 0.05. The remaining variables YearBuiltCat2, GarageArea, WoodDeckSF, OpenPorchSF, and MoSoldCat2 are NOT statistically significant.

Code Snippet 03 in section 8.3 contains the R code for the model selection using the forward variable selection on the cleaned train data partition.

4.2. Backward Variable Selection

For the backward variable selection, we utilized the stepAIC function in R with "backward" direction and a starter regression equation containing all the predictor variables in the candidate pool list (Table 03, the base category variables were not included for the discrete variables). The stepAIC function performed 10 iterations based on the AIC value. It selected a regression equation with 18 terms (17 predictor variables and an intercept term).

The fitted regression equation selected by the backward variable selection is

SalePrice = -21,116.750 + 1.954 * LotArea+ 34.252 * MasVnrArea+ 31.900 * TotalBsmtSF+ 10.798 * GarageArea + 10.991 * WoodDeckSF + 25.143 * OpenPorchSF + 43.557 * ScreenPorch+ 1,662.643 * QualityIndex + 24.601 * TotalSqftCalc + 28.730 * HomeSize + 4,307.927 * YearBuiltCat2 + 32,816.850 * YearBuiltCat3 + 31,627.120 * GarageCarsCat3 + 36,675.590 * TotRmsAbvGrdCat3 + 2,932.676 * MoSoldCat2-30,849.140 * BedroomAbvGrCat2 -37,407.890 * BedroomAbvGrCat3

Note: The backward variable selection and the forward variable selection picked out the same model.

On the predictor variables picked out by the stepAIC function using the backward variable selection, we computed the VIF values (Variance Inflation Factors) to determine if any of the predictor variables are collinear. VIF values greater than 20 indicate the presence of multicollinearity among the predictor variables. One way to address the multicollinearity is by removing the variables with VIF value > 20 and re-fitting the model. Table 07 shows the VIFs computed for the 17 predictor variables in the descending order of the value.

Predictor Variable Name	VIF value
HomeSize	4.102371
TotalSqftCalc	3.650919
GarageCarsCat3	2.481769
GarageArea	2.446537
YearBuiltCat3	2.035002
TotalBsmtSF	1.851932
BedroomAbvGrCat3	1.731164
BedroomAbvGrCat2	1.675785
MasVnrArea	1.592965
QualityIndex	1.458150
YearBuiltCat2	1.356379
TotRmsAbvGrdCat3	1.321570
LotArea	1.293311
WoodDeckSF	1.289029
OpenPorchSF	1.233304
ScreenPorch	1.096863
MoSoldCat2	1.021262

Table 07: List of VIFs values in descending order for the predictor variables identified by using the backward variable selection

None of the VIF values listed in Table 07 are greater than 20, indicating multicollinearity is not a problem among the variables selected by the backward selection algorithm. ***As a result, none of the variables are removed and re-fitting is not attempted.***

	Dependent Variable - SalePrice
Constant	-21,116.750** (9,433.428)
LotArea	1.954*** (0.211)
MasVnrArea	34.252*** (5.645)
TotalBsmtSF	31.900*** (2.986)
GarageArea	10.798 (7.462)
WoodDeckSF	10.991 (7.283)
OpenPorchSF	25.143 (16.922)
ScreenPorch	43.557*** (15.154)
QualityIndex	1,662.643*** (146.026)
TotalSqftCalc	24.601*** (2.266)
HomeSize	28.730*** (3.603)
YearBuiltCat2	4,307.927 (2,760.384)
YearBuiltCat3	32,816.850*** (2,476.583)
GarageCarsCat3	31,627.120*** (3,636.783)
TotRmsAbvGrdCat3	36,675.590*** (5,321.358)
MoSoldCat2	2,932.676 (1,983.580)
BedroomAbvGrCat2	-30,849.140*** (6,474.304)
BedroomAbvGrCat3	-37,407.890*** (11,043.030)

Observations	742
R2	0.914
Adjusted R2	0.912
Residual Std. Error	23,610.780 (df = 724)
F Statistic	450.635*** (df = 17; 724)

Note: *p<0.1; **p<0.05; ***p<0.01

Table 08: Regression output using stargazer for the final estimated model using the Backward Variable Selection (the coefficient and the standard error (in blue) are listed against each variable) The content of this table matches Table 05 (regression output of model selected by forward variable selection).

In Table 08, the coefficients and the corresponding standard errors are exactly the same as in Table 05 (Regression output for the model selected by Forward Variable Selection). Therefore, as with the forward variable selection model, at a level of significance of 0.001, the overall regression is strongly statistically significant with a p-value < 2.2e-16 (with a F-statistic value of 450.635). So, at a level of significance of 0.001, we can reject the null joint hypothesis that all the predictor variables have a zero coefficient and have no effect on the regression.

Among the strongly statistically significant variables, GarageCarCat3 (houses with 3 or more car capacity), YearBuiltCat2 (house built between 1970 and 1989), YearBuiltCat3 (house built between 1990 and 2010), TotRmsAbvGrdCat3 (category for houses with 10 or more total rooms), BedroomAbvGrCat2 (category for houses with 2 to 4 bedrooms) and BedroomAbvGrCat3 (category for houses with more than 5 bedrooms) have large standard errors (in excess of 1000) for the coefficients.

Like with the model selected by the forward variable selection, given all else in the model is held constant, a unit increase in the HomeSize (sum of FirstFlrSF and SecondFlSF) will result in an increase of $28.730 in the Sale Price. A unit increase in TotalBsmtSF results in an increase of $31.90 in the Sale Price given everything else about the model is held constant. Similarly, while holding other variables fixed, a house with 3 or cars garage results in an increase of $31,627.120 in the Sale Price than the base category (with no garage).

We also gather from the model provided by backward variable selection that a house built between 1990 and 2010 will result in an increase of $32,816.850 compared to the base category (a house built between 1950 and 1969). A unit increase in the QualityIndex results in an increase of $ 1,662.643 in the Sale Price with everything else held constant. A unit increase of TotalSqftCalc results in an increase of $24.601 in the Sale Price when everything is fixed. Separately, a unit increase in LotArea results in an increase of $1.954 in the Sale price given all else in the model is held constant. Also, a house with 10 or more rooms results in an increase of $36,675.590 in Sale Price compared to the base category (house with 3 or 4 rooms) when everything is fixed. A unit increase of MasVnrArea increases the SalePrice by $34.252 under the condition all else in the model is constant. A unit increase in ScreenPorch area will provide an increase of $43.557 to the SalePrice given all else in the model is held constant.

The intercept shows the expected mean of the SalePrice is negative (-21,116.750) when all the predictor variables are zero. In addition to the intercept, there are two other coefficients with negative values. A house with 2 to 4 bedrooms results in a decrease of $30,849.140 in SalePrice compared to base category of 0 or 1 bedrooms when all is held constant. Compared to this, a house with more than 5 bedrooms results in a decrease of $37,407.890 in SalePrice compared to a house of 0 or 1 bedrooms when everything in the model is fixed. Interestingly, this does not support intuition.

The coefficient of determination (Adjusted R-squared) value of 0.912 indicates approximately 91.2% of the variation in the SalePrice can be explained by variation in the predictor variables listed in Table 08.

From Table 08, we noted that the following predictor variables were also not selected by the backward variable selection algorithm:

- LotFrontage - Linear feet of street connected to property
- EnclosedPorch - Enclosed porch area in square feet
- ThreeSsnPorch - Three season porch area in square feet

- PoolArea - Pool area in square feet
- Fireplaces – Number of Fireplaces
- GarageCarsCat2 – category of the GarageCars discrete variable representing 1 or 2 car garages
- TotRmsAbvGrdCat2 – category of the TotRmsAbvGrd discrete variable representing 5 to 8 rooms
- MoSoldCat3 – category of the MoSold discrete variable representing the months Sept thru Dec when the house was sold

Term	t value	Pr(>|t|)
(Intercept)	-2.239	0.025491 *
LotArea	9.281	< 2e-16 ***
MasVnrArea	6.068	2.09e-09 ***
TotalBsmtSF	10.685	< 2e-16 ***
GarageArea	1.447	0.148313
WoodDeckSF	1.509	0.131726
OpenPorchSF	1.486	0.137757
ScreenPorch	2.874	0.004169 **
QualityIndex	11.386	< 2e-16 ***
TotalSqftCalc	10.855	< 2e-16 ***
HomeSize	7.974	6.01e-15 ***
YearBuiltCat2	1.561	0.119049
YearBuiltCat3	13.251	< 2e-16 ***
GarageCarsCat3	8.696	< 2e-16 ***
TotRmsAbvGrdCat3	6.892	1.20e-11 ***
MoSoldCat2	1.478	0.139715
BedroomAbvGrCat2	-4.765	2.28e-06 ***
BedroomAbvGrCat3	-3.387	0.000744 ***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Table 09: Table shows the t-values and p-values for the individual coefficient tests on the terms in the model selected by backward Variable Selection

As shown in Table 09, the coefficients for the predictor variables HomeSize, TotalBsmtSF, GarageCarsCat3, QualityIndex, TotalSqftCalc, YearBuiltCat3, LotArea, TotRmsAbvGrdCat3, MasVnrArea, BedroomAbvGrCat2, BedroomAbvGrCat3 are strongly statistically significant at a level of significance of 0.001. The coefficient for ScreenPorch is only significant at a higher level of significance of 0.01. Lastly, the intercept is significant at a level of significance of 0.05. The coefficients for the variables YearBuiltCat2, GarageArea, WoodDeckSF, OpenPorchSF, MoSoldCat2 are NOT statistically significant.

Code Snippet 03 in section 8.3 contains the R code for the model selection using the backward variable selection method on the cleaned train data partition.

4.3. Stepwise Variable Selection

For the Stepwise variable selection, we invoked the stepAIC function in R with “both” direction and a starter regression equation containing a simple linear regression model with TotalSqftCalc as the predictor variable and SalePrice as the response variable. Additionally, to ensure an exhaustive search is performed, we provided a scope variable to the stepAIC containing the full model with all the predictor variables in the candidate pool and the intercept-only model.

The stepAIC function performed 17 iterations. It picked out a regression equation with 18 terms (17 predictor variables and an intercept term). The selection of a variable is performed based on the AIC value. Though theoretically, it is possible for the stepwise variable algorithm to eliminate a previously added variable, in our execution, we did not notice any added variable dropped at a later step.

The fitted regression equation selected by the stepwise variable selection is

SalePrice = - 21,116.750 + 24.601 * TotalSqftCalc + 31,627.120 * GarageCarsCat3 + 32,816.850 * YearBuiltCat3 + 1,662.643 * QualityIndex + 1.954 * LotArea + 31.900 * TotalBsmtSF + 28.730 * HomeSize + 36,675.590 * TotRmsAbvGrdCat3 + 34.252 * MasVnrArea-30,849.140 * BedroomAbvGrCat2 -37,407.890 * BedroomAbvGrCat3 + 43.557 * ScreenPorch + 4,307.927 * YearBuiltCat2 + 10.798 * GarageArea + 10.991 * WodDeckSF+ 25.143 * OpenPorchSF+ 2,932.676 * MoSoldCat2

Note: The Stepwise variable selection method picked out the same model as the backward variable selection and the forward variable selection methods.

On the predictor variables picked out by the stepAIC function using the stepwise variable selection, we then computed the VIF values (Variance Inflation Factors) to determine if any of the predictor variables are collinear. VIF values greater than 20 indicate multicollinearity.

Predictor Variable Name	VIF value
HomeSize	4.102371
TotalSqftCalc	3.650919
GarageCarsCat3	2.481769
GarageArea	2.446537
YearBuiltCat3	2.035002
TotalBsmtSF	1.851932
BedroomAbvGrCat3	1.731164
BedroomAbvGrCat2	1.675785
MasVnrArea	1.592965
QualityIndex	1.458150
YearBuiltCat2	1.356379
TotRmsAbvGrdCat3	1.321570
LotArea	1.293311
WoodDeckSF	1.289029
OpenPorchSF	1.233304
ScreenPorch	1.096863
MoSoldCat2	1.021262

Table 10: List of VIFs values in descending order for the predictor variables identified by using the Stepwise variable selection

None of the VIF values listed in Table 10 are greater than 20 indicating multicollinearity is not a problem among the variables selected by the stepwise selection algorithm. ***As a result, none of the variables are removed.***

	Dependent Variable - SalePrice
Constant	-21,116.750** (9,433.428)
TotalSqftCalc	24.601*** (2.266)
GarageCarsCat3	31,627.120*** (3,636.783)
YearBuiltCat3	32,816.850*** (2,476.583)
QualityIndex	1,662.643*** (146.026)
LotArea	1.954*** (0.211)
TotalBsmtSF	31.900*** (2.986)
HomeSize	28.730*** (3.603)
TotRmsAbvGrdCat3	36,675.590*** (5,321.358)
MasVnrArea	34.252*** (5.645)
BedroomAbvGrCat2	-30,849.140*** (6,474.304)
BedroomAbvGrCat3	-37,407.890*** (11,043.030)
ScreenPorch	43.557*** (15.154)
YearBuiltCat2	4,307.927 (2,760.384)
GarageArea	10.798 (7.462)
WoodDeckSF	10.991 (7.283)
OpenPorchSF	25.143 (16.922)
MoSoldCat2	2,932.676 (1,983.580)

Observations	742
R2	0.914
Adjusted R2	0.912
Residual Std. Error	23,610.780 (df = 724)
F Statistic	450.635*** (df = 17; 724)

Note: *p<0.1; **p<0.05; ***p<0.01

Table 11: Regression output using stargazer for the final estimated model using Stepwise Variable Selection (the coefficient and the standard error (in blue) are listed against each variable)

In the Table 11, the coefficients and the corresponding standard errors are exactly the same as the Regression output for the model selected by Forward Variable Selection and Backward Variable Selection. So, as with the forward and backward variable selection models, we conclude that at a level of significance of 0.001, the overall regression is strongly statistically significant with a p-value < 2.2e-16 (with F-statistic value 450.635). So, at a level of significance of 0.001, we can reject the joint null hypothesis that all the predictor variables have a zero coefficient and have no effect on the regression.

As with the forward and backward variable selection models, TotRmsAbvGrdCat3 (category for house with 10 or more total rooms), TotRmsAbvGrdCat2 (category for house with 5 to 8 rooms), GarageCarCat3 (houses with 3 or more car capacity garage), YearBuiltCat2 (house built between 1970 and 1989), an dYearBuiltCat3 (house built between 1990 and 2010) have large standard errors in the range of 2000-6000 among the strongly statistically significant variables.

We again note that given all else in the model is held constant, a unit increase in the HomeSize (sum of FirstFlrSF and SecondFlSF) will result in an increase of $28.730 in the Sale Price. Similarly, when all else is held constant, a house with 3 or car capacity garage results in an increase of $31,627.120 in Sale Price than the base category (with no garage). Likewise, given all else in the model is held constant, a house built between 1990 and 2010 results in an increase of $32,816.850 in Sale Price than the base category (a house built between 1950 and 1969). In comparison, under the same conditions, a house built between 1970 and 1989 results only in an increase of $4,307.927 in the Sale price in comparison with the base category (a house built between 1950 and 1969) when all else is held the constant. Also, a unit increase in the QualityIndex results in an increase of $1,662.643 in the Sale Price with everything else held constant. A house with 10 or more rooms results in an increase of $36,675.590 in comparison to the base category (a house with 3-4 rooms).

Also, interestingly the variables – BedroomAbvGrCat2 and BedroomAbvGrCat3 (representing a house with 2 to 4 bedrooms and a house with more than 5 bedrooms respectively) have negative coefficients (result in a decrease of $30,849.140 and $37,407.890 respectively in comparison with the base category for these variables which is a house with 0 or 1 bedrooms when all else in the model is constant).

The intercept is also negative indicating the expected Sale price is -$21,116.750 when all the variables are 0.

The coefficient of determination (Adjusted R-squared) value of 0.912 indicates approximately 91.2% of the variation in the SalePrice can be explained by variation in the predictor variables listed in Table 11.

From the Table 11, we can notice that the following predictor variables were not selected by the stepwise variable selection algorithm:

- LotFrontage - Linear feet of street connected to property
- EnclosedPorch - Enclosed porch area in square feet
- ThreeSsnPorch - Three season porch area in square feet
- PoolArea - Pool area in square feet
- Fireplaces – Number of Fireplaces
- GarageCarsCat2 – category of the GarageCars discrete variable representing 1 or 2 car garages
- TotRmsAbvGrdCat2 – category of the TotRmsAbvGrd discrete variable representing 5 to 8 rooms

- MoSoldCat3 – category of the MoSold discrete variable representing the months Sept thru Dec when the house was sold

Term	t value	Pr(>|t|)
(Intercept)	-2.239	0.025491 *
TotalSqftCalc	10.855	< 2e-16 ***
GarageCarsCat3	8.696	< 2e-16 ***
YearBuiltCat3	13.251	< 2e-16 ***
QualityIndex	11.386	< 2e-16 ***
LotArea	9.281	< 2e-16 ***
TotalBsmtSF	10.685	< 2e-16 ***
HomeSize	7.974	6.01e-15 ***
TotRmsAbvGrdCat3	6.892	1.20e-11 ***
MasVnrArea	6.068	2.09e-09 ***
BedroomAbvGrCat2	-4.765	2.28e-06 ***
BedroomAbvGrCat3	-3.387	0.000744 ***
ScreenPorch	2.874	0.004169 **
YearBuiltCat2	1.561	0.119049
GarageArea	1.447	0.148313
WoodDeckSF	1.509	0.131726
OpenPorchSF	1.486	0.137757
MoSoldCat2	1.478	0.139715

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Table 12: Table shows the t-values and p-values for the individual coefficient tests on the terms in the model selected by Stepwise Variable Selection

In Table 12, among the coefficient T-test results, we determine only the coefficients for the predictor variables TotalSqftCalc, GarageCarsCat3, YearBuiltCat3, QualityIndex, LotArea, TotalBsmtSF, HomeSize, TotRmsAbvGrdCat3, MasVnrArea, BedroomAbvGrCat2, and BedroomAbvGrCat3 are strongly statistically significant at a level of significant of 0.001. The coefficient for the variable ScreenPorch is significant only at a slightly higher level of significance of 0.01. The intercept is significant at a level of significance of 0.05. The remaining variables GarageArea, MoSoldCat2, OpenPorchSF, WoodDeckSF are not significant.

Code Snippet 03 in section 8.3 contains the R code for the model selection using the stepwise variable selection method on the cleaned train data partition.

4.4. Junk Model

For the purpose of model comparison, a fourth model is created. The predictor variables used to fit the model are strongly interrelated. Regression models that are fit to data by the method of least squares when strong multicollinearity is present are notoriously poor (unstable) and the values of the coefficients are often very sensitive to the data in the particular sample collected. However, multicollinearity affects statistical inference primarily. On the other hand, a model with multicollinearity could perform well-enough to use in prediction (this is shown to be the case with this model in the next section). Hence, the model is called the junk model. Thus, the model picked out by the automatic variable selection methods will be compared to the junk model to determine which one fits better.

Fitting a multiple linear regression model to the observations of OverallQual, OverallCond, QualityIndex, GrLivArea, and TotalSqftCalc to predict SalePrice using OLS results in:

$$\textbf{SalePrice} = -358{,}190.800 + 68{,}852.890 * \textbf{OverallQual} + 43{,}947.330 * \textbf{OverallCond} - 7{,}406.182 * \textbf{QualityIndex} + 27.028 * \textbf{GrLivArea} + 43.716 * \textbf{TotalSqftCalc}$$

Table 13 shows the VIF (Variance Inflation Factors) values computed for the 5 variables to identify if any of the predictor variables are collinear. OverallCond, OverallQual, and QualityIndex have large VIF values in excess of 20 indicating very strong multicollinearity. This is because QualityIndex is the product of OverallCond and OverallQual.

Predictor Variable Name	VIF value
QualityIndex	60.882305
OverallQual	56.700119
OverallCond	32.169719
GrLivArea	3.354516
TotalSqftcalc	2.588014

Table 13: List of VIFs values in descending order for the predictor variables in the junk model

	Dependent variable: SalePrice
Constant	- 358,190.800** (37,318.030)
OverallQual	68,852.890*** (68,852.890)
OverallCond	43,947.330*** (6,897.452)
QualityIndex	- 7,406.182*** (1,191.276)
GrLivArea	27.028*** (4.113)
TotalSqftCalc	43.716*** (2.409)

Observations	742
R2	0.860
Adjusted R2	0.859
Residual Std. Error	29,808.990 (df = 736)
F Statistic	904.880*** (df = 5; 736)

Note *p<0.1; **p<0.05; ***p<0.01

Table 14: Regression output using stargazer for the fitted junk model with the coefficient and the standard error (in blue) are listed against each variable

From Table 14, the overall F-test shows at a level of significance of 0.001, we can reject the null hypothesis that all the predictor variables have a zero coefficient and have no effect on the regression. The p-value for the overall F-statistic (value of 904.880) is < 2.2e-16.

Due to the presence of collinearity among some predictor variables, it severely limits the use of the regression equation for inference and forecasting. The regression coefficients cannot be interpreted in the usual manner. But the model may still serve well for the purpose of prediction.

The coefficient of determination (Adjusted R-squared) value of 0.859 indicates approximately 85.9% of the variation in the SalePrice can be explained by variation in the predictor variables.

From Table 14, we noted that the intercept, the variables OverallCond, OverallQual, and QualityIndex have large standard error value (greater than 1000).

Term	t value	Pr(>|t|)
(Intercept)	-9.598	< 2e-16 ***
OverallQual	10.9	< 2e-16 ***
OverallCond	6.372	3.30e-10 ***
QualityIndex	-6.217	8.49e-10 ***
GrLivArea	6.571	9.43e-11 ***
TotalSqftCalc	18.147	< 2e-16 ***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Table 15: Table shows the t-values and p-values for the individual coefficient tests on the terms in the junk model

From Table 15, the individual regression coefficients for all the variables and the intercept are statistically significant at a level of significance of 0.001. So, at 0.001 level of significance, we can reject the null hypothesis that the individual regression coefficient is 0.

Code Snippet 03 in section 8.3 contains the R code for the junk model fitted using the training data partition.

4.5. Model Comparison

In this section, we present the results of the model comparison based on the metrics of in-sample fit and predictive accuracy. The metrics were computed using the train data partition (in-sample data). Table 16 shows the Adjusted-R-squared, AIC, BIC, MSE (Mean Squared Error), and MAE (Mean Absolute Error) for the four models. MSE and MAE measure the predictive accuracy. Adjusted-R-squared, AIC, BIC measure the in-sample fit. Table 17 shows the four models side-by-side with the regression coefficients, standard error values, overall F-Statistic etc. Code Snippet 03 in section 8.3 has the R code for the model comparison.

Model	Adjusted R-Squared	Akaike Information Criterion (AIC)	Bayesian Information Criterion (BIC)	Mean Squared Error (MSE)	Mean Absolute Error (MAE)	Overall Model Rank
forward.lm	**0.912**	**17068.56**	**17156.14**	**543945509**	**16320.05**	**1**
backward.lm	**0.912**	**17068.56**	**17156.14**	**543945509**	**16320.05**	**1**
stepwise.lm	**0.912**	**17068.56**	**17156.14**	**543945509**	**16320.05**	**1**
junk.lm	0.859	17402.69	17434.95	881390650	21860.88	2

Table 16: Table shows the Adjusted-R^2, AIC, BIC, MSE, and MAE for the four models – forward variable selection model (forward.lm), backward variable selection (backward.lm), stepwise variable selection (stepwise.lm), junk model (junk.lm). Shown in blue for each column (metric) are the model(s) that performed better.

The forward, backward, and stepwise variable selection methods picked out the same model. This model is compared to the junk model using the metrics that measure the in-sample fit and predictive accuracy.

Adjusted-R-squared, the coefficient of determination is defined as the proportion of the total variability in the response variable that is explained by the regression model. The model with larger Adjusted-R-squared value is preferred. The model selected by the automatic variable selection methods has a higher Adjusted-R-squared value (of 91.2%) in comparison to the junk model (85.9%). Highlighted in blue is the higher value for Adjusted-R-squared.

AIC is used to rank the models based on the twin criteria of goodness-of-fit and simplicity (number of parameters). The model with the lower AIC is preferred. To compare the AIC values between models, they are computed using the same set of observations. In our case, the model selected by the automatic variable selection methods has a better AIC value than the junk model. Highlighted in blue is the minimum value of AIC.

A variation of AIC is the BIC which imposes a more severe penalty on the number of parameters (complexity) to control the overfitting. The penalty is more severe in BIC when the number of observations is more than 8 which is the case here. The model with the lower BIC is preferred. Here too the model selected by the automatic variable selection methods has a better BIC value than the junk model. Highlighted in blue is the minimum value of BIC.

For predictive accuracy, we want the mean squared error (MSE) and mean absolute value (MAE) to be lower. From Table 16, we note, in comparison to the junk model, the model selected by the automated variable selection has smaller MSE and MAE values. Highlighted in blue are the minimum values of MSE and MAE.

The model selected by the automatic variable selection methods (forward, backward, and stepwise) has the higher rank in all the metrics compared to the junk model.

	forward.lm (1)	backward.lm (2)	stepwise.lm (3)	junk.lm (4)
Constant	-21,116.750** (9,433.428)	-21,116.750** (9,433.428)	-21,116.750** (9,433.428)	-358,190.800*** (37,318.030)
HomeSize	28.730*** (3.603)	28.730*** (3.603)	28.730*** (3.603)	
TotalBsmtSF	31.900*** (2.986)	31.900*** (2.986)	31.900*** (2.986)	
GarageCarsCat3	31,627.120*** (3,636.783)	31,627.120*** (3,636.783)	31,627.120*** (3,636.783)	
OverallQual				68,852.890*** (6,316.625)
OverallCond				43,947.330*** (6,897.452)
QualityIndex	1,662.643*** (146.026)	1,662.643*** (146.026)	1,662.643*** (146.026)	-7,406.182*** (1,191.276)
GrLivArea				27.028*** (4.113)
TotalSqftCalc	24.601*** (2.266)	24.601*** (2.266)	24.601*** (2.266)	43.716*** (2.409)
YearBuiltCat3	32,816.850*** (2,476.583)	32,816.850*** (2,476.583)	32,816.850*** (2,476.583)	
LotArea	1.954*** (0.211)	1.954*** (0.211)	1.954*** (0.211)	
TotRmsAbvGrdCat3	36,675.590*** (5,321.358)	36,675.590*** (5,321.358)	36,675.590*** (5,321.358)	
MasVnrArea	34.252*** (5.645)	34.252*** (5.645)	34.252*** (5.645)	
BedroomAbvGrCat2	-30,849.140*** (6,474.304)	-30,849.140*** (6,474.304)	-30,849.140*** (6,474.304)	
BedroomAbvGrCat3	-37,407.890*** (11,043.030)	-37,407.890*** (11,043.030)	-37,407.890*** (11,043.030)	
ScreenPorch	43.557*** (15.154)	43.557*** (15.154)	43.557*** (15.154)	
YearBuiltCat2	4,307.927 (2,760.384)	4,307.927 (2,760.384)	4,307.927 (2,760.384)	
GarageArea	10.798 (7.462)	10.798 (7.462)	10.798 (7.462)	
WoodDeckSF	10.991 (7.283)	10.991 (7.283)	10.991 (7.283)	
OpenPorchSF	25.143 (16.922)	25.143 (16.922)	25.143 (16.922)	
MoSoldCat2	2,932.676 (1,983.580)	2,932.676 (1,983.580)	2,932.676 (1,983.580)	
Observations	742	742	742	742
R^2	0.914	0.914	0.914	0.860
Adjusted R^2	0.912	0.912	0.912	0.859
Residual Std. Error	23,610.780 (df = 724)	23,610.780 (df = 724)	23,610.780 (df = 724)	29,808.990 (df = 736)
F Statistic	450.635*** (df = 17; 724)	450.635*** (df = 17; 724)	450.635*** (df = 17; 724)	904.880*** (df = 5; 736)

Table 17: Table shows comparison of the four models with regression coefficients, standard error (in blue color), R-squared, adjusted-R-squared, F-Statistic values.

5. Predictive Accuracy

In this section, we present the predictive accuracy metrics computed using the out-of-sample (test) data. This is to understand how the model would behave with data different from what the model has trained with. These metrics measure the error of the predicted values.

MAE measures the average magnitude of the errors in the predictions, without considering the direction. MSE is the average of squared differences between prediction and observed values. These prediction accuracy metrics express the prediction error in the units of the response variable. Lower values are preferred.

Model	Mean Squared Error (MSE)	Mean Absolute Error (MAE)	Model Rank (based on MAE)
forward.lm	764580041	**17219.03**	1
backward.lm	764580041	**17219.03**	1
stepwise.lm	764580041	**17219.03**	1
junk.lm	**709529319**	20209.85	2

Table 18 shows the MSE and MAE values computed using test data partition for the four models. Shown in blue for each column (metric) are the model(s) that performed better based on that metric.

From Table 18, it can be noted that the model selected by the automatic variable selection methods has **larger** MSE value than the junk model. In contrast, the MAE value for the model selected by the automatic variable selection methods **is lower than** the junk model.

Since we have different models picked out by MSE and MAE values, we needed to determine whether we prefer the model that has the better MSE or MAE values. The decision of whether to use MSE or MAE depends on whether outliers are present in the predicted values generated by the model or not. MAE is generally more robust to outliers. The MSE will be much larger in the presence of outliers.

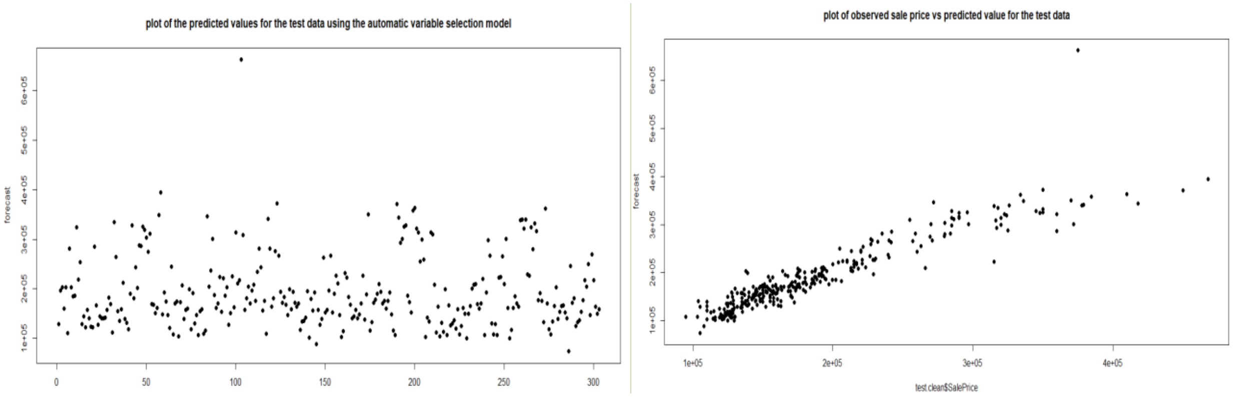

FIG 01 a) Scatter plot of the predicted values based on index for the test data using the model picked out by the automatic variable selection methods and b) plot of observed sale price vs predicted value for the test data using the model picked out by the automatic variable selection methods (the outliers (> 600,000) are can be seen towards to the top of the plots)

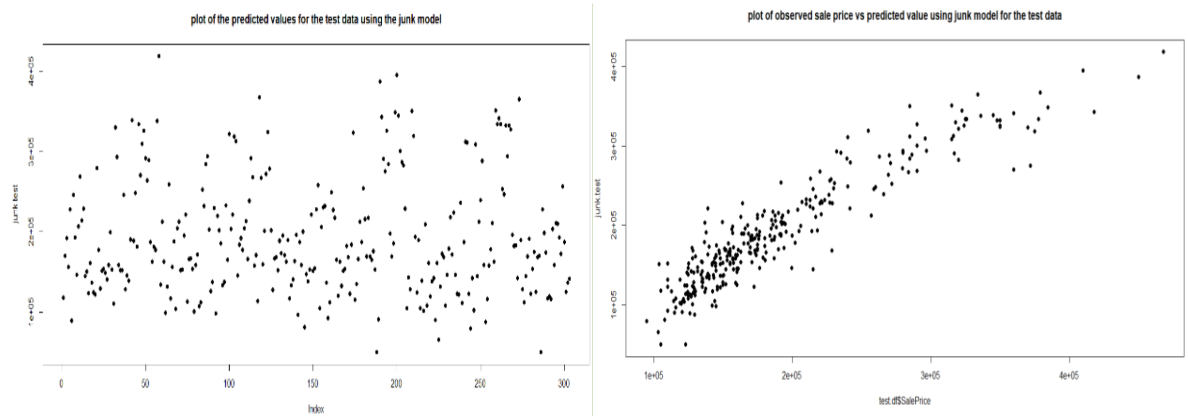

FIG 02 a) Scatter plot of the predicted values based on index for the test data using the junk model and b) plot of observed sale price vs predicted value for the test data using the junk model

From FIG 01 and FIG 02, we can see there are outliers present in the predicted values, using the model picked out by the automatic variable selection methods, which could result in very large MSE values for that model. **Therefore, we preferred to use MAE value instead of MSE in assessing the model performance. MAE are more robust to outliers.** Based on the MAE value, the model selected by the automatic variable selection methods is determined to be have the better predictive performance.

We also noted that the MSE and MAE values are much larger for the test data (out-of-sample data) compared to the values obtained using the training data (in-sample data). **Having a model that has better predictive accuracy in-sample than it does out-of-sample indicates the model is overfitted to the training data.**

Code Snippet 04 in section 8.4 contains the R code for the computing the predictive accuracy metrics using the test data partition.

6. Operational Validation

In this section, we present the results of the operational validation using an application-specific predictive accuracy metric called Prediction Grade. The Prediction Grade is determined for each predicted value by computing the ratio of the absolute value of the residual to the observed value. Then a policy is defined such that

a. If the ratio is between 0 and 0.10, the predicted value is assigned Grade 1.
b. If the ratio is between 0.10 and 0.15, the predicted value is assigned Grade 2.
c. If the ratio is between 0.15 and 0.25, the predicted value is assigned Grade 3.
d. If the ratio is 0.25 or more, the predicted value is assigned Grade 4.

We computed the prediction grades using both the training and the test data. Prediction grade is used in the rating of AVM models.

The forward, backward, and stepwise variable selection methods picked out the same model. This model is compared to the junk model using the prediction grade metric computed using train and test data.

Prediction grades for the model selected by the automatic variable selection methods and the junk model using the in-sample (train) data set:

By comparing the prediction grades in Table 19 and 20, we can determine that model selected by the automatic variable selection methods is more accurate than the junk model since more predicted values (14% more) are of Grade 1. The model selected by the automatic variable selection methods has additionally has fewer predicted values that are of Grade 4 (6.4% fewer) than the junk model. The percent of predicted values of Grade 2 and Grade 3 are also fewer in comparison to the junk model.

So, using the in-sample data and the new definition of predictive accuracy, the model selected by the automatic variable selection methods is still the better model compared to the junk model.

Grade1: [0,0.10]	Grade2: [0.10,0.15]	Grade3: [0.15, 0.25]	Grade4: (0.25+]
0.679	0.163	0.136	0.022

Table 19: the distribution table of the prediction grades computed for the model selected by the automatic variable selection methods using the in-sample data

Grade1: [0,0.10]	Grade2: [0.10,0.15]	Grade3: [0.15, 0.25]	Grade4: (0.25+]
0.539	0.201	0.174	0.086

Table 20: the distribution table of the prediction grades computed for the junk model using the in-sample data

Prediction grades for the model selected by the automatic variable selection methods and the junk model using the out-of-sample (test) data set:

Using the out-of-sample prediction grades computed in Table 21 and 22, we determined that the junk model under performs compared to the model selected by the automatic variable selection methods. The model selected by the automatic variable selection methods has 10.2% more predicted values of Grade 1 than the junk model and 6.6% fewer predicted values of Grade 4. The Grade 2 and Grade 3 predicted values are also fewer than obtained with the junk model.

One thing to note is that the grades obtained with the out-of-sample data are slightly worse than the ones obtained using the in-sample data.

But the out-of-sample prediction grades do not change the ranking of the model. The model selected by the automatic variable selection methods is the better model compared to the junk model.

Grade1: [0,0.10]	Grade2: [0.10,0.15]	Grade3: [0.15, 0.25]	Grade4: (0.25+]
0.650	0.172	0.149	0.030

Table 21: The distribution table of the prediction grades computed for the model selected by the automatic variable selection methods using out-of-sample data

Grade1: [0,0.10]	Grade2: [0.10,0.15]	Grade3: [0.15, 0.25]	Grade4: (0.25+]
0.548	0.195	0.162	0.096

Table 22: The distribution table of the prediction grades computed for the junk model using out-of-sample data

Additionally, since the predicted values of the model are accurate to within 10 % for more than 50% of the data (this is true with both the in-sample (Grade 1 values are 67.9% of the total) and the out-of-sample data (Grade 1 values are 65% of the total)), the model selected by the automatic variable selection methods (forward, backward, and stepwise) is of the 'underwriting quality'.

Code Snippet 05 in section 8.5 contains the R code for the operational validation (application-specific predictive accuracy assessment).

7. Summary

We began the analysis by creating a sample from the population. Drop conditions were created based on the definition of a 'typical' house and also based on reading the data dictionary. By applying the drop conditions, a sample of 1359 observations was created from the population.

For the model identification, a candidate pool of 20 variables was created with a mixture of continuous and discrete variables. The candidate pool has three continuous variables (TotalSqftCalc, QualityIndex, and HomeSize) that were created based on the existing predictor variables for use in the model selection. Additionally, for the discrete variables, several indicator/dummy variables were created.

From the sample data, any observations with missing values were removed, and the data was split using random sampling into two sets – one for in-sample model development and one for out-of-sample model assessment - following the predictive modeling framework (70/30 training/test split).

After that, model selection was performed using automatic variable selection methods (forward variable selection, backward variable selection, and stepwise variable selection). In our case, all three methods selected the same model. VIF values were computed to determine the presence of any multicollinearity among the predictor variables selected by the model. None of the VIF values for the predictor variables were greater than 20. Hence, no refitting was performed. This model was compared with the junk model (a model we fitted with highly correlated predictor variables) using metrics of in-sample fit (adjusted-R-squared, AIC, BIC) and in-sample predictive accuracy metrics (MSE, MAE). The model picked out by the automatic variable selection methods ranked better compared to the junk model among all the metrics computed with the in-sample data.

Next, we compared the model selected by the automatic variable selection methods and the junk model using the predictive accuracy metrics computed on the out-of-sample data. In this case, we had the MSE and MAE metrics pick out different models, but since the out-of-sample data produced some outliers in the predicted values, we preferred the MAE metric. Outliers generate large MSE values. Based on the MAE metric, the model selected by the automatic variable selection evaluated better compared to the junk model.

Lastly, we evaluated the model selected by the automatic variable selection methods and the junk model using an application-specific predictive accuracy metric called prediction grade which is used in the AVM models. Here too, the model picked out the automatic variable selection models ranked better than the junk model. Additionally, with both the in-sample and the out-of-sample data, it turned out the model picked out by the automatic variable selection methods was accurate to within 10% for more 65% of the time. Therefore, the model picked out by the automatic variable selection methods in our case is of the "underwriting quality".

8. Code

8.1. Sample Definition

```
# Read in csv file for Ames housing data;
path.name <- 'C:\\Users\\harinikanand\\Desktop\\NorthwesternUniversity\\MSDS-410-DL\\Week5\\Assignment5\\Code\\';
file.name <- paste(path.name,'ames_housing_data.csv',sep='');

# Read in the csv file into an R data frame;
ames.df <- read.csv(file.name,header=TRUE,stringsAsFactors=FALSE);

# Create a waterfall of drop conditions to define a sample population
ames.df$dropCondition <- ifelse((ames.df$Zoning!='RH' & ames.df$Zoning!='RL' & ames.df$Zoning!='RP' &
ames.df$Zoning!='RM'),'01: Not Residential',
  ifelse(ames.df$SaleCondition!='Normal','02: Non-Normal Sale',
  ifelse(ames.df$BldgType!='1Fam','03: Not Single Family Home',
  ifelse(ames.df$Street!='Pave','04: Street Not Paved',
  ifelse((ames.df$Utilities!='AllPub' & ames.df$Utilities!='NoSewr'),'05: No Water',
  ifelse(ames.df$YearBuilt <1950,'06: Built Pre-1950',
  ifelse(ames.df$TotalBsmtSF <1,'07: No Basement',
  ifelse((ames.df$GrLivArea <800 | ames.df$GrLivArea >4000),'08: LT 800 OR GT 4000 SqFt',
    ifelse((ames.df$MiscVal!=0),'09: MiscVal>0','99: Eligible Sample'
  )))))))))

# Save the counts of the drop condition rules
waterfall <- table(ames.df$dropCondition);

# Convert waterfall table as a matrix
as.matrix(waterfall,10,1)

# Eliminate all observations that are not part of the eligible sample population;
eligible.population <- subset(ames.df,dropCondition=='99: Eligible Sample');

# Check that all remaining observations are eligible and obtain the count of the eligible population
table(eligible.population$dropCondition)

# Save the eligible.population as RDS file
saveRDS(eligible.population,file='C:\\Users\\harinikanand\\Desktop\\NorthwesternUniversity\\MSDS-410-DL\\Week5\\Assignment5\\Code\\sample_population.rds')
```

Code Snippet 01: R code for creation of the eligible population sample

8.2. Variable creation and Train/Test Split

```
# Read the eligible.population from RData file
my.data <- readRDS('C:\\Users\\harinikanand\\Desktop\\NorthwesternUniversity\\MSDS-410-
DL\\Week5\\Assignment5\\Code\\sample_population.rds')

##########################################################################################
# Define new continuous variables and indicator/dummy variables for the discrete variables          #
##########################################################################################
# Create a variable for the overall quality of the house
my.data$QualityIndex <- my.data$OverallQual * my.data$OverallCond
# Create a variable for the total square footage of the house
my.data$TotalSqftCalc <- my.data$BsmtFinSF1 + my.data$BsmtFinSF2 + my.data$GrLivArea
# Create a variable for the 1st and 2nd floor sizes
my.data$HomeSize <- my.data$FirstFlrSF + my.data$SecondFlrSF

# Create 3 indicator variables for Fireplaces Category
# Category 1 is category with 0 Fireplaces
my.data$FireplacesCat1 <- ifelse((my.data$Fireplaces < 1),1,0)
# Category 2 includes Fireplaces with 1 or 2 cars
my.data$FireplacesCat2 <- ifelse(((my.data$Fireplaces == 1) | (my.data$Fireplaces == 2)),1,0)
# Category 3 includes Fireplaces with 3 or 4 cars
my.data$FireplacesCat3 <- ifelse((my.data$Fireplaces >= 3),1,0)

# Create 3 indicator variables for YearBuilt Category
# Category 1 contains houses built between 1950 and 1969
my.data$YearBuiltCat1 <- ifelse((my.data$YearBuilt >=1950 & my.data$YearBuilt <=1969),1,0)
# Category 2 contains houses built between 1970 and 1989
my.data$YearBuiltCat2 <- ifelse((my.data$YearBuilt >=1970 & my.data$YearBuilt <=1989),1,0)
# Category 3 contains houses built between 1990 and 2010
my.data$YearBuiltCat3 <- ifelse((my.data$YearBuilt >=1990 & my.data$YearBuilt <=2010),1,0)

# Create 3 indicator variables for GarageCars Category
# Category 1 includes houses with 0 car capacity in the garage
my.data$GarageCarsCat1 <- ifelse((my.data$GarageCars < 1),1,0)
# Category 2 contains houses with 1 or 2 car capacity in the garage
my.data$GarageCarsCat2 <- ifelse(((my.data$GarageCars == 1) | (my.data$GarageCars == 2)),1,0)
# Category 3 contains houses with 3 or more car capacity in the garage
my.data$GarageCarsCat3 <- ifelse((my.data$GarageCars >= 3),1,0)

# Create 3 indicator variables for TotRmsAbvGrd Category
# Category 1 includes houses with 3 or 4 rooms
my.data$TotRmsAbvGrdCat1 <- ifelse(((my.data$TotRmsAbvGrd == 3) | (my.data$TotRmsAbvGrd == 4)),1,0)
# Category 2 contains houses with 5 to 8 rooms
my.data$TotRmsAbvGrdCat2 <- ifelse(((my.data$TotRmsAbvGrd >= 5) & (my.data$TotRmsAbvGrd <= 9)),1,0)
# Category 3 contains houses with 11 or more rooms
my.data$TotRmsAbvGrdCat3 <- ifelse(((my.data$TotRmsAbvGrd >= 10)),1,0)

# Create 3 indicator variables for MoSold Category
# Category 1 includes houses sold in Jan or Feb
my.data$MoSoldCat1 <- ifelse((my.data$MoSold == 1 | my.data$MoSold == 2),1,0)
# Category 2 contains houses sold between Mar and Aug
my.data$MoSoldCat2 <- ifelse(((my.data$MoSold >= 3) & (my.data$MoSold <= 8)),1,0)
# Category 3 contains houses sold between Sept and Dec
my.data$MoSoldCat3 <- ifelse((my.data$MoSold >= 9 & my.data$MoSold <=12),1,0)

# Create 3 indicator variables for BedroomAbvGrd Category
# Category 1 includes houses with 0 or 1 BedroomsAbvGr
```

```
my.data$BedroomAbvGrCat1 <- ifelse((my.data$BedroomAbvGr == 0 | my.data$BedroomAbvGr == 1),1,0)
# Category 2 contains houses with 2 to 4 BedroomsAbvGr
my.data$BedroomAbvGrCat2 <- ifelse(((my.data$BedroomAbvGr >= 2) & (my.data$BedroomAbvGr <= 4)),1,0)
# Category 3 contains houses with 5 or more BedroomsAbvGr
my.data$BedroomAbvGrCat3 <- ifelse((my.data$BedroomAbvGr >= 5),1,0)

# Create a list of unwanted variables
drop.list <-
c('SID','PID','SubClass','Zoning','Street','Alley','LotShape','LandContour','Utilities','LotConfig','LandSlope','Neighborhood',
'Condition1','Conditionq','BldgType','HouseStyle','YearRemodel','RoofStyle','RoofMat','Exterior1','Exterior2','MasVnrTyp
e','ExterQual','ExterCond','FoundationBsmtQual','BsmtCond','BsmtExposure','BsmtFinType1','BsmtFinType2','BsmtUnfS
F','Heating','Electrical','FirstFlrSF','SecondFlrSF','LowQualFinSF','BsmtFullBath','BsmtHalfBath','FullBath','HalfBath','Bedr
oomAbvGr','KitchenAbvGr','KitchenQual','Functional','GarageType','GarageYrBlt','GarageFinish','GarageCars','Condition
2','Foundation','BsmtQual','GarageYrBlt','PoolQC','Fence','MiscFeature','MiscVal','YrSold','SaleType','SaleCondition','dro
pCondition','Fireplaces','YearBuilt','GarageCars','TotRmsAbvGrd','TotBathRms','HeatingQC','CentralAir','FireplaceQu','Ga
rageQual','GarageCond','PavedDrive','MoSold','BedroomAbvGrd','BsmtFinSF1','BsmtFinSF2')

# Create a data frame by removing the unwanted variables from the eligible.population
skinny.df <- my.data[,!(names(my.data) %in% drop.list)];

# Use the structure command to view the contents of the data frame;
str(skinny.df)

# Remove the observations with missing values
sample.df <- na.omit(skinny.df);

# Check the change in dimension;
dim(skinny.df)
dim(sample.df)
dim(skinny.df)-dim(sample.df)

##########################################################################################
# Create train/test data partitions from the sample data                                 #
##########################################################################################

# Set the seed on the random number generator
set.seed(123)
sample.df$u <- runif(n=dim(sample.df)[1],min=0,max=1)

# Create train/test split
train.df <- subset(sample.df,u<0.70);
test.df <- subset(sample.df,u>=0.70);

# create a second list of unwanted variables for the automatic variable selection
drop.list2 <-
c('BsmtFinSF1','BsmtFinSF2','GrLivArea','OverallCond','OverallQual','u','FireplacesCat1','YearBuiltCat1','GarageCarsCat1',
'TotRmsAbvGrdCat1','MoSoldCat1','BedroomAbvGrCat1')

# Create train.clean by shedding the unwanted attributes
train.clean <- train.df[,!(names(train.df) %in% drop.list2)]

# Create test.clean by shedding the unwanted attributes
test.clean <- test.df[,!(names(test.df) %in% drop.list2)]
```

Code Snippet 02: R code for a.) the creation of the additional continuous variables, b.) for the creation of the indicator/dummy variables, and c.) creation of train/test data partitions using predictive modeling framework

8.3. Model Identification using Automatic Variable Selection

```
# Include the MASS library
library(MASS)

# Include the car library
library(car)

# Include the stargazer library
library(stargazer)

# location to store the stargazer html tables
out.path <- 'C:\\Users\\harinikanand\\Desktop\\NorthwesternUniversity\\MSDS-410-
DL\\Week5\\Assignment5\\Code\\stargazer\\';

# 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)
summary(lower.lm)

# Create a SLR to initialize the stepwise selection
sqft.lm <- lm(SalePrice ~ TotalSqftCalc, data=train.clean)
summary(sqft.lm)

######################################################################
#            Forward Variable Selection using AIC                    #
######################################################################
forward.lm <- stepAIC(object=lower.lm,scope=list(upper=formula(upper.lm),lower=~1), direction=c('forward'))
summary(forward.lm)

# Compute the VIFs value for the predictor variables in forward.lm
sort(vif(forward.lm),decreasing=TRUE)

# Create a stargazer table with the regression output of forward.lm
file.name <- 'forward_lm.html';
stargazer(forward.lm, type=c('html'),out=paste(out.path,file.name,sep=''),
      title=c('Table XX: Regression output of forward.lm'),
      align=TRUE, digits=3, digits.extra=3, initial.zero=TRUE, intercept.bottom=FALSE)

######################################################################
#             Backward Elimination using AIC                         #
######################################################################
backward.lm <- stepAIC(object=upper.lm,direction=c('backward'))
summary(backward.lm)

# Compute the VIFs value for the predictor variables in backward.lm
sort(vif(backward.lm), decreasing=TRUE)

# Create a stargazer table with the regression output of backward.lm
file.name <- 'backward_lm.html';
stargazer(backward.lm, type=c('html'),out=paste(out.path,file.name,sep=''),
      title=c('Table XX: Regression output of backward_lm'),
      align=TRUE, digits=3, digits.extra=3, initial.zero=TRUE, intercept.bottom=FALSE)
```

```
####################################################################
#              Stepwise Variable Selection using AIC                                  #
####################################################################
stepwise.lm <- stepAIC(object=sqft.lm,scope=list(upper=formula(upper.lm),lower=~1), direction=c('both'))
summary(stepwise.lm)

# Compute the VIFs value for the predictor variables in stepwise.lm
sort(vif(stepwise.lm), decreasing=TRUE)

# Create a table with the regression output of stepwise.lm
file.name <- 'stepwise_lm.html';
stargazer(stepwise.lm, type=c('html'),out=paste(out.path,file.name,sep=''),
       title=c('Table XX: Regression output of stepwise_lm'),
       align=TRUE, digits=3, digits.extra=3, initial.zero=TRUE, intercept.bottom=FALSE)

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

# Compute the VIFs value for the predictor variables in junk.lm
sort(vif(junk.lm), decreasing=TRUE)

# Create a table with the regression output of junk.lm
file.name <- 'junk_lm.html';
stargazer(junk.lm, type=c('html'),out=paste(out.path,file.name,sep=''),
       title=c('Table XX: Regression output of junk_lm'),
       align=TRUE, digits=3, digits.extra=3, initial.zero=TRUE, intercept.bottom=FALSE)

###############################################################################
#    Model Comparison using metrics of in-sample fit and predictive accuracy (train data)        #
###############################################################################

# Create a side-by-side table with all four models regression output
file.name <- 'model_comparison.html';
stargazer(forward.lm, backward.lm,stepwise.lm,junk.lm, type=c('html'),out=paste(out.path,file.name,sep=''),
       title=c('Table XX: Comparison of forward.lm, backward.lm, stepwise.lm and junk.lm'),
       align=TRUE, digits=3, digits.extra=2, initial.zero=TRUE,
       column.labels=c('forward.lm','backward.lm','stepwise.lm','junk.lm'), intercept.bottom=FALSE )

# Compute the AIC, BIC, MSE and MAE for the forward.lm model using in-sample data
aic.forward <- AIC(forward.lm)
bic.forward <- BIC(forward.lm)
mse.forward <- mean(forward.lm$residuals^2)
mae.forward <- mean(abs(forward.lm$residuals))

# Compute the AIC, BIC, MSE and MAE for the backward.lm model using in-sample data
aic.backward <- AIC(backward.lm)
bic.backward <- BIC(backward.lm)
mse.backward <- mean(backward.lm$residuals^2)
mae.backward <- mean(abs(backward.lm$residuals))

# Compute the AIC, BIC, MSE and MAE for the stepwise.lm model using in-sample data
aic.stepwise <- AIC(stepwise.lm)
bic.stepwise <- BIC(stepwise.lm)
mse.stepwise <- mean(stepwise.lm$residuals^2)
mae.stepwise <- mean(abs(stepwise.lm$residuals))
```

```
# Compute the AIC, BIC, MSE and MAE for the junk.lm model using in-sample data
aic.junk <- AIC(junk.lm)
bic.junk <- BIC(junk.lm)
mse.junk <- mean(junk.lm$residuals^2)
mae.junk <- mean(abs(junk.lm$residuals))
```

Code Snippet 03: R code for model identification using a) forward variable selection, b) backward variable selection, c) stepwise variable selection. It also contains R code for junk model fitting and model comparison using in-sample fit and predictive accuracy metrics.

8.4. Predictive Accuracy using the out-of-sample data

```
####################################################################################
#           Predictive Accuracy using out-of-sample data                           #
####################################################################################

# Compute the MSE and MAE for the forward.lm model
mse.test.forward <- mean((test.clean$SalePrice - predict(forward.lm, newdata=test.clean))^2)
mae.test.forward <- mean(abs(test.clean$SalePrice - predict(forward.lm, newdata=test.clean)))

# Compute the MSE and MAE for the backward.lm model
mse.test.backward <- mean((test.clean$SalePrice - predict(backward.lm, newdata=test.clean))^2)
mae.test.backward <- mean(abs(test.clean$SalePrice - predict(backward.lm, newdata=test.clean)))

# Compute the MSE and MAE for the stepwise.lm model
mse.test.stepwise <- mean((test.clean$SalePrice - predict(stepwise.lm, newdata=test.clean))^2)
mae.test.stepwise <- mean(abs(test.clean$SalePrice - predict(stepwise.lm, newdata=test.clean)))

# Compute the MSE and MAE for the junk.lm model
mse.test.junk <- mean((test.df$SalePrice - predict(junk.lm, newdata=test.df))^2)
mae.test.junk <- mean(abs(test.df$SalePrice - predict(junk.lm, newdata=test.df)))

# plot of the predicted values for the test data using the automatic variable selection model
# Note forward.lm, backward.lm and stepwise.lm are all the same model
forecast <- predict(forward.lm, newdata=test.clean)
plot(forecast,pch=16)
title('plot of the predicted values for the test data using the automatic variable selection model')

# plot of the sale price vs predicted values for the test data using the automatic variable selection model
plot(test.clean$SalePrice, forecast,pch=16)
title('plot of observed sale price vs predicted value for the test data')

# plot of the predicted values for the test data using the junk model
plot(junk.test,pch=19)
title('plot of the predicted values for the test data using the junk model')

# plot of the sale price vs predicted values for the test data using the junk model
plot(test.df$SalePrice, junk.test,pch=19)
title('plot of observed sale price vs predicted value using junk model for the test data')
```

Code Snippet 04: R code for predictive accuracy using out-of-sample (test) data partition

8.5. Operational Validation

```
######################################################################################
#       Application Specific Predictive Accuracy for forward.lm and junk.lm models using train data      #
######################################################################################

# Compute production grade for the model selected by the automatic variable selection using train data
# Abs Pct Error
forward.pct <- abs(forward.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+]')
                               )
                             )

# Create a distribution table with the forward.lm prediction grade values from train data
forward.trainTable <- table(forward.PredictionGrade)
forward.trainTable/sum(forward.trainTable)

# Compute the production grade for the junk model using train data
# Abs Pct Error
junk.pct <- abs(junk.lm$residuals)/train.df$SalePrice

# Assign Prediction Grades
junk.PredictionGrade <- ifelse(junk.pct<=0.10, 'Grade 1: [0,0.10]',
                           ifelse(junk.pct<=0.15, 'Grade 2: (0.10,0.15]',
                             ifelse(junk.pct<=0.25,'Grade 3: (0.15,0.25]',
                              'Grade 4: (0.25+]')
                           )
                         )

# Create a distribution table with the junk.lm prediction grade values from train data
junk.trainTable <- table(junk.PredictionGrade)
junk.trainTable/sum(junk.trainTable)

######################################################################################
#       Application Specific Predictive Accuracy for forward.lm and junk.lm models using test data        #
######################################################################################

# Compute the production grade for the model selected by the automatic variable selection using test data
# Abs Pct Error
forward.test <- predict(forward.lm, newdata=test.clean)
forward.testPCT <- abs(test.clean$SalePrice-forward.test)/test.clean$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+]')
                                   )
                                 )
# Create a distribution table with the forward.lm prediction grade values from test data
forward.testTable <- table(forward.testPredictionGrade)
forward.testTable/sum(forward.testTable)
```

```
# Compute the production grade for the junk model using test data
# Abs Pct Error
junk.test <- predict(junk.lm, newdata=test.df)
junk.testPCT <- abs(test.df$SalePrice-junk.test)/test.df$SalePrice

# Assign Prediction Grades
junk.testPredictionGrade <- ifelse(junk.testPCT<=0.10, 'Grade 1: [0,0.10]',
                              ifelse(junk.testPCT<=0.15, 'Grade 2: (0.10,0.15]',
                                ifelse(junk.testPCT<=0.25,'Grade 3: (0.15,0.25]',
                                 'Grade 4: (0.25+]')
                                )
                              )
# Create a distribution table with the junk.lm prediction grade values from test data
junk.testTable <- table(junk.testPredictionGrade)
junk.testTable/sum(junk.testTable)
```

Code Snippet 05: R code for operational validation (application-specific predictive accuracy)