Assignment #5

Nate Bitting

Introduction

The objective of this assignment is to perform automated variable selection techniques for identifying the "best" regression model for predicting sale price for homes in the Ames, Iowa area. The first phase includes the assessment of which predictor variables, based on common sense and business justification, made sense to include in a predictive model. After conducting some preliminary exploratory data analysis (EDA), it was concluded that the following predictor variable candidates would be considered in the model:

- **X1:** YearBuilt – year the home was built
- **X2:** total_SF – total square footage in the home
- **X3:** total_baths – total number of bathrooms in the home
- **X4:** good_kitchen – indicator variable to determine the condition of the kitchen
- **X5:** good_fireplace – indicator variable to determine the condition of the fireplace
- **X6:** good_exterior – indicator variable to determine the condition of the exterior of the home
- **X7:** quality_index – measure of Overall Condition * Overall Quality
- **X8:** central_air – indicator variable to determine if the home has central air or not
- **X9:** fireplace_ind – indicator variable to determine if there were any fireplaces or not
- **X10:** garage_ind – indicator variable to determine if there was a garage or not
- **X11:** good_basement_ind – indicator variable to determine the condition of the basement

After we assess which of the eleven predictor variables should be included in our model, we then will conduct an assessment of extreme observations, or outliers. The approach we will use for outlier detection is by outputting the studentized residuals for a model that includes all eleven variables. We will then remove any observation that has an absolute value of studentized residual exceeding 2. The next step will be to create a training and test dataset that could be used for building the model. The training set will be used to build the regression models and the test set will be used to test the predictive accuracy of the final selected model. Lastly, we will assign a prediction grade to each observation based on how accurate our predictions on the test set compare against the observed values. This will be the basis of our final assessment of how well the model will perform in a real-world setting using new observations.

Results

Model Identification by Automated Variable Selection

After performing our preliminary EDA and outlier removal process, the next step was to perform various automated variable selection methods in order to identify the "best" regression model. The automated variable selection methods used on this analysis included the following:

- Adjusted R-squared (Model_AdjR2)
- Maximum R-squared (Model_MaxR)
- Mallow's Cp (Model_MCp)
- Forward selection (Model_F)
- Backward selection (Model_B)
- Stepwise selection (Model_S)

In the next section we will go through each variable selection method and assess the final model and discuss each step in the variable selection process and provide an assessment of the results.

Adjusted R-Squared Model (Model_AdjR2)

The results of the adjusted R-squared variable selection method determined that all predictor variable candidates (X1-X11) should be included in the final model. The adjusted R-squared of the final model was 0.8887, which was far superior when a subset of the predictor variables were included in the model. In Table 1 below, you can see the summary output of the variable selection process with the final result in the first row.

Table 1 – Adjusted R-squared variable selection summary

Number in Model	Adjusted R-Square	R-Square	Variables in Model
11	0.8887	0.8898	YearBuilt total_SF total_baths good_kitchen good_fireplace good_exterior quality_index central_air fireplace_ind garage_ind good_basement_ind
10	0.8885	0.8894	YearBuilt total_SF total_baths good_kitchen good_fireplace good_exterior quality_index central_air fireplace_ind good_basement_ind
10	0.8885	0.8894	YearBuilt total_SF total_baths good_kitchen good_fireplace good_exterior quality_index central_air fireplace_ind garage_ind
9	0.8883	0.8891	YearBuilt total_SF total_baths good_kitchen good_fireplace good_exterior quality_index central_air fireplace_ind
10	0.8879	0.8888	YearBuilt total_SF total_baths good_fireplace good_exterior quality_index central_air fireplace_ind garage_ind good_basement_ind
10	0.8877	0.8887	YearBuilt total_SF total_baths good_kitchen good_fireplace good_exterior quality_index fireplace_ind garage_ind good_basement_ind
10	0.8877	0.8887	YearBuilt total_SF total_baths good_kitchen good_exterior quality_index central_air fireplace_ind garage_ind good_basement_ind
9	0.8877	0.8886	YearBuilt total_SF total_baths good_fireplace good_exterior quality_index central_air fireplace_ind good_basement_ind
9	0.8877	0.8885	YearBuilt total_SF total_baths good_kitchen good_fireplace good_exterior quality_index fireplace_ind good_basement_ind
9	0.8876	0.8884	YearBuilt total_SF total_baths good_fireplace good_exterior quality_index central_air fireplace_ind garage_ind
9	0.8875	0.8884	YearBuilt total_SF total_baths good_kitchen good_exterior quality_index central_air fireplace_ind garage_ind
9	0.8875	0.8883	YearBuilt total_SF total_baths good_kitchen good_exterior quality_index central_air fireplace_ind good_basement_ind
8	0.8875	0.8882	YearBuilt total_SF total_baths good_fireplace good_exterior quality_index central_air fireplace_ind

NOTE: Not all models included due to limited space

Based on these results, the adjusted R-squared method recommends we include all candidate predictor variables we originally considered for the model (X1-X11) to yield the highest possible adjusted R-squared value.

Maximum R-Squared Model (Model_MaxR)

Just like the adjusted R-squared variable selection method, the results of the maximum R-squared variable selection method also determined that all predictor variable candidates (X1-X11) should be included in the final model. The R-squared of the final model was 0.8898, which was far superior when a subset of the predictor variables were included in the model. In Table 2 below, we can see the ANOVA and parameter estimates for the final suggested model from the maximum R-squared variable selection method.

Table 2 – Maximum R-squared variable selection summary

Maximum R-Square Improvement: Step 11

Variable garage_ind Entered: R-Square = 0.8898 and C(p) = 12.0000

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	11	3.708006E12	3.370915E11	850.35	<.0001
Error	1159	4.594433E11	396413516		
Corrected Total	1170	4.167449E12			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	-898145	79758	50267949163	126.81	<.0001
YearBuilt	448.06847	41.11992	47068764087	118.74	<.0001
total_SF	47.94454	1.24588	5.870452E11	1480.89	<.0001
total_baths	7137.00060	1098.01401	16748046366	42.25	<.0001
good_kitchen	5114.75146	1628.58766	3909986365	9.86	0.0017
good_fireplace	5587.92777	1662.48892	4478499220	11.30	0.0008
good_exterior	10620	1916.40496	12172904617	30.71	<.0001
quality_index	1441.45165	82.73133	1.203396E11	303.57	<.0001
central_air	-11873	3541.64889	4455031344	11.24	0.0008
fireplace_ind	8325.14250	1504.36520	12140183362	30.63	<.0001
garage_ind	7249.84653	3883.73858	1381357538	3.48	0.0622
good_basement_ind	3557.07044	1869.02846	1435825141	3.62	0.0573

Bounds on condition number: 3.1103, 228.45

The above model is the best 11-variable model found.

No further improvement in R-Square is possible.

Based on these results, the maximum R-squared method recommends we include all candidate predictor variables we originally considered for the model (X1-X11) that would yield the highest R-squared value.

Mallow's Cp Model (Model_MCp)

As with the previous two models, the results of the Mallow's Cp variable selection method also determined that all predictor variable candidates (X1-X11) should be included in the final model. The Mallow's Cp of the final model was 12.0, which was the smallest result from all other subset model alternatives. In Table 3 below, we can see some of the output from the Mallow's Cp variable selection process. The best model is shown in the first row of the table.

Table 3 – Mallow's Cp variable selection summary

Number in Model	C(p)	R-Square	Variables in Model
11	12.0000	0.8898	YearBuilt total_SF total_baths good_kitchen good_fireplace good_exterior quality_index central_air fireplace_ind garage_ind good_basement_ind
10	13.4846	0.8894	YearBuilt total_SF total_baths good_kitchen good_fireplace good_exterior quality_index central_air fireplace_ind good_basement_ind
10	13.6220	0.8894	YearBuilt total_SF total_baths good_kitchen good_fireplace good_exterior quality_index central_air fireplace_ind garage_ind
9	14.7703	0.8891	YearBuilt total_SF total_baths good_kitchen good_fireplace good_exterior quality_index central_air fireplace_ind
10	19.8634	0.8888	YearBuilt total_SF total_baths good_fireplace good_exterior quality_index central_air fireplace_ind garage_ind good_basement_ind
9	20.5344	0.8886	YearBuilt total_SF total_baths good_fireplace good_exterior quality_index central_air fireplace_ind good_basement_ind
9	20.9530	0.8885	YearBuilt total_SF total_baths good_kitchen good_fireplace good_exterior quality_index fireplace_ind good_basement_ind
10	21.2383	0.8887	YearBuilt total_SF total_baths good_kitchen good_fireplace good_exterior quality_index fireplace_ind garage_ind good_basement_ind
10	21.2975	0.8887	YearBuilt total_SF total_baths good_kitchen good_exterior quality_index central_air fireplace_ind garage_ind good_basement_ind
9	21.7200	0.8884	YearBuilt total_SF total_baths good_fireplace good_exterior quality_index central_air fireplace_ind garage_ind
8	22.0781	0.8882	YearBuilt total_SF total_baths good_fireplace good_exterior quality_index central_air fireplace_ind
9	22.6109	0.8884	YearBuilt total_SF total_baths good_kitchen good_exterior quality_index central_air fireplace_ind garage_ind
9	22.7771	0.8883	YearBuilt total_SF total_baths good_kitchen good_exterior quality_index central_air fireplace_ind good_basement_ind
8	22.9991	0.8881	YearBuilt total_SF total_baths good_kitchen good_fireplace good_exterior quality_index fireplace_ind

Based on these results, the Mallow's Cp method recommends we include all candidate predictor variables we originally considered for the model (X1-X11) that would yield the lowest Mallow's Cp value.

Forward Selection Model (Model_F)

As with all proceeding variable selection methods, the results of the forward variable selection method also determined that all predictor variable candidates (X1-X11) should be included in the final model. In Table 4 below, we can see the summary of the forward selection method and you'll notice that the p-value from the nested F-tests did not increase until the 7th variable was entered into the model. For this method, we chose a *slentry* value of 0.15 as our threshold for variables to be allowed to enter into the model.

<u>Table 4 – Forward variable selection summary</u>

Summary of Forward Selection							
Step	Variable Entered	Number Vars In	Partial R-Square	Model R-Square	C(p)	F Value	Pr > F
1	total_SF	1	0.7605	0.7605	1350.89	3711.90	<.0001
2	YearBuilt	2	0.0571	0.8176	752.324	365.85	<.0001
3	quality_index	3	0.0498	0.8675	230.301	438.91	<.0001
4	good_exterior	4	0.0073	0.8748	155.436	68.08	<.0001
5	fireplace_ind	5	0.0064	0.8812	89.7125	63.18	<.0001
6	total_baths	6	0.0048	0.8860	41.7044	48.56	<.0001
7	good_fireplace	7	0.0011	0.8871	31.6481	11.82	0.0006
8	central_air	8	0.0011	0.8882	22.0781	11.44	0.0007
9	good_kitchen	9	0.0009	0.8891	14.7703	9.27	0.0024
10	good_basement_ind	10	0.0003	0.8894	13.4846	3.28	0.0704
11	garage_ind	11	0.0003	0.8898	12.0000	3.48	0.0622

Based on these results, the forward selection method recommends we include all candidate predictor variables we originally considered for the model (X1-X11).

Backward Selection Model (Model_F)

The backward variable selection method actually did not iterate through any removal steps as all predictor variable were kept in the model. This result is exactly the same as all previous selection methods. A p-value of 0.15 was used for the *slstay* option, which resulted in none of the variables being removed from the model.

Stepwise Selection Model (Model_S)

The stepwise selection method was the final option used for model selection. As with all previous methods, the stepwise selection method indicated that all predictor variable candidates should remain in the model. The stepwise variable selection summary is shown in Table 5 below. You can see that all variables were entered into the model at each step with the Model R-Square increasing and Cp decreasing with the addition of each new variable. The final "best" model from this method indicates that all variables should remain in the model.

Table 5 – Forward variable selection summary

Summary of Stepwise Selection								
Step	Variable Entered	Variable Removed	Number Vars In	Partial R-Square	Model R-Square	C(p)	F Value	Pr > F
1	total_SF		1	0.7605	0.7605	1350.89	3711.90	<.0001
2	YearBuilt		2	0.0571	0.8176	752.324	365.85	<.0001
3	quality_index		3	0.0498	0.8675	230.301	438.91	<.0001
4	good_exterior		4	0.0073	0.8748	155.436	68.08	<.0001
5	fireplace_ind		5	0.0064	0.8812	89.7125	63.18	<.0001
6	total_baths		6	0.0048	0.8860	41.7044	48.56	<.0001
7	good_fireplace		7	0.0011	0.8871	31.6481	11.82	0.0006
8	central_air		8	0.0011	0.8882	22.0781	11.44	0.0007
9	good_kitchen		9	0.0009	0.8891	14.7703	9.27	0.0024
10	good_basement_ind		10	0.0003	0.8894	13.4846	3.28	0.0704
11	garage_ind		11	0.0003	0.8898	12.0000	3.48	0.0622

Model Comparison

Given all variable selection methods resulted in the same model that included all eleven predictor variable candidates, the model fit criteria is exactly the same for all models. Table 6 below shows the model fit criteria from the models created using the training sample. In the next couple sections we will explore if any multicollinearity exists in the model as well as the operational accuracy of the final model that includes all predictor variables using the test dataset (out-of-sample).

Table 6 – Model Comparison from Training and Test samples

		Model_AdjR2	Model_MaxR	Model_MCp	Model_F	Model_B	Model_S
Training Sample	Predictor(s) Selected	X1 - X11	X1 - X11	X1 - X11	X1 - X11	X1 - X11	X1 - X11
	Adjusted R2	0.8887	0.8887	0.8887	0.8887	0.8887	0.8887
	AIC	23195.3592	23195.3592	23195.3592	23195.3592	23195.3592	23195.3592
	BIC	23197.6074	23197.6074	23197.6074	23197.6074	23197.6074	23197.6074
	Mallow's Cp	12.0000	12.0000	12.0000	12.0000	12.0000	12.0000
	MSE	396413516	396413516	396413516	396413516	396413516	396413516
	MAE	15372.37	15372.37	15372.37	15372.37	15372.37	15372.37
Test Sample	MSE	402854448	402854448	402854448	402854448	402854448	402854448
	MAE	15488.47	15488.47	15488.47	15488.47	15488.47	15488.47

Based on the results in Table 6, we can see that the predictive ability of the final model built from the training sample performed very well with the test sample data.

Multicollinearity Assessment

Given the end result for all variable selection methods determined all predictor variable candidates should be included in the final model, we performed an assessment for multicollinearity on the final model only. To asses multicollinearity we leverage the variable inflation factor (VIF) statistic as shown in Table 7 below.

Table 7 – Parameter Estimates with Variable Inflation Factors (VIF)

Parameter Estimates						
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Variance Inflation
Intercept	1	-898145	79758	-11.26	<.0001	0
YearBuilt	1	448.06847	41.11992	10.90	<.0001	3.11033
total_SF	1	47.94454	1.24588	38.48	<.0001	2.18016
total_baths	1	7137.00060	1098.01401	6.50	<.0001	1.83676
good_kitchen	1	5114.75146	1628.58766	3.14	0.0017	1.92785
good_fireplace	1	5587.92777	1662.48892	3.36	0.0008	1.46337
good_exterior	1	10620	1916.40496	5.54	<.0001	2.29912
quality_index	1	1441.45165	82.73133	17.42	<.0001	1.38250
central_air	1	-11873	3541.64889	-3.35	0.0008	1.22244
fireplace_ind	1	8325.14250	1504.36520	5.53	<.0001	1.66208
garage_ind	1	7249.84653	3883.73858	1.87	0.0622	1.11225
good_basement_ind	1	3557.07044	1869.02846	1.90	0.0573	2.57132

Common practical experience suggests that if any VIFs exceeds 5 or 10, it is an indication that multicollinearity exists in the model. Based on the VIF values for each parameter in Table 7, we conclude that no multicollinearity exists among the predictor variables given the fact that none of the VIFs exceeds 5.

Operational Validation

By leveraging the eleven predictor variable candidates suggested from the variable selection and performing model adequacy checking, the final model resulted strong predictive accuracy. To assess the operational accuracy of the final model, we placed the predictive scores, absolute value for each observation's actual vs predicted value for the response variable SalePrice, into three categories: Grade 1 (within 10% of the observed value), Grade 2 (between 10-15% of the observed value), Grade 3 (everything else).

Table 8 – Frequency Table for Prediction Grade Categories

Prediction_Grade	Frequency	Percent	Cumulative Frequency	Cumulative Percent
01: Grade 1	344	64.66	344	64.66
02: Grade 2	101	18.98	445	83.65
03: Grade 3	87	16.35	532	100.00

Based on the results of the prediction scores using the test sample dataset, we can see in Table 8 that 83.7% of the predicted values were within 15% of the observed value. Depending on the success criteria from management, this could be considered a usable model for predicting the sale price for homes in the Ames, Iowa area.

Conclusion

One can understand how automated variable selection methods are extremely useful in the model selection process, however, in this particular exercise all methods suggested the same “best” model. Given the fact that we had a biased selection of the original candidate predictor variables, we could further explore the incorporation of additional variables from the dataset. After working with the Ames housing dataset for the past several weeks, we have learned the importance of proper exploratory data analysis leveraging both statistics and visual aids, such as scatter plots, histograms, or correlation matrices. The final model selection process of actually calculating the predictive accuracy by using out-of-sample observations on the model was the most beneficial takeaway from all assignments to date. This was the final step in determining how well the models would perform in the final intended environment.

Our final model yielded a fairly high Adjusted R-squared of 0.8887, which shows extremely good fit to the data. However, I think if we wanted to explore ways of further improving predictive accuracy of our model, we should partner with subject matter experts in real estate in the Ames, Iowa area. By partnering with a realtor in the area, they could further inform us which predictor variables could be important when determining sale price for homes in the area. This would of course introduce bias to the model, but we need to ensure we include variables that are not only statistically significant, but also those that are practical from a business perspective.

SAS Code Output

```
*-----------------------------------------------------------------------------------------
*  Nate Bitting
*  Assignment 5
*-----------------------------------------------------------------------------------------;

* Code used to get the data into my library;
ods graphics on;
libname mydata '/courses/d6fc9ae5ba27fe300/c_3505/SAS_Data/' access=readonly;
proc datasets library=mydata; run; quit;

*-----------------------------------------------------------------------------------------
*  Create original dataset by filtering out unnecessary data and adding new categorical
   features to the dataset
*-----------------------------------------------------------------------------------------;

* Smart data formats for sale price and square footage;
proc format;
value price_sfmt
    . = '10: Missing'
    1 -< 100000 = '01: [1; 100,000)'
    100000 -< 150000 = '02: [100,000; 150,000)'
    150000 -< 200000 = '03: [150,000; 200,000)'
    200000 -< 250000 = '04: [200,000; 250,000)'
    250000 -< 300000 = '05: [250,000; 300,000)'
    300000 -< 350000 = '06: [300,000; 350,000)'
    350000 -< 400000 = '07: [350,000; 400,000)'
    400000 - high = '08: [400,000+]'
    other = '09: Invalid Value'
    ; * use a semi-colon to end each format in the proc format statement;
    * Note on how we use the < to create open intervals.
    * The dash - will create closed intervals, and
    hence the dash should only be used with discrete values;
value sqft_sfmt
    . = '08: Missing'
    1 - 1000 = '01: [1; 1,000]'
    1000 <- 1500 = '02: (1,000; 1,500]'
    1500 <- 2000 = '03: (1,500; 2,000]'
    2000 <- 2500 = '04: (2,000; 2,500]'
    2500 <- 3000 = '05: (2,500; 3,000]'
    3000 - high = '06: (3,000+]'
    other = '07: Invalid Value'
    ;
run;

* Dataset before removing outliers;
Data building;
    SET mydata.ames_housing_data;

    * filter on only single family homes that meet specific criteria;
    if (SaleCondition = 'Normal');
    if (BldgType = '1Fam'); * single family homes only;
    if (Zoning in ('RH','RL','RP','RM')); * residential zones only;
    if (Street='Pave'); * paved streets only;
    if (Utilities='AllPub'); * only homes with public utilities;

    log_price = log(SalePrice); *create a variable for the natural log of SalePrice;

    * create new variables by combining multiple variables in the housing dataset;
    total_SF = max(GrLivArea,0) + max(TotalBsmtSF,0);
    total_baths = max(FullBath,0) + max(BsmtFullBath,0);
    total_halfbaths = max(HalfBath,0) + max(BsmtHalfBath,0);
    total_baths_calc = total_baths + total_halfbaths;

    * Neighborhood dummy variables;
    if (Neighborhood = 'Blmngtn') then Blmngtn=1; else Blmngtn=0;
    if (Neighborhood = 'Blueste') then Blueste=1; else Blueste=0;
    if (Neighborhood = 'BrDale') then BrDale=1; else BrDale=0;
    if (Neighborhood = 'BrkSide') then BrkSide=1; else BrkSide=0;
    if (Neighborhood = 'ClearCr') then ClearCr=1; else ClearCr=0;
    if (Neighborhood = 'CollgCr') then CollgCr=1; else CollgCr=0;
    if (Neighborhood = 'Crawfor') then Crawfor=1; else Crawfor=0;
    if (Neighborhood = 'Edwards') then Edwards=1; else Edwards=0;
    if (Neighborhood = 'Gilbert') then Gilbert=1; else Gilbert=0;
    if (Neighborhood = 'Greens') then Greens=1; else Greens=0;
    if (Neighborhood = 'GrnHill') then GrnHill=1; else GrnHill=0;
    if (Neighborhood = 'IDOTRR') then IDOTRR=1; else IDOTRR=0;
    if (Neighborhood = 'Landmrk') then Landmrk=1; else Landmrk=0;
    if (Neighborhood = 'MeadowV') then MeadowV=1; else MeadowV=0;
    if (Neighborhood = 'Mitchel') then Mitchel=1; else Mitchel=0;
    if (Neighborhood = 'NAmes') then NAmes=1; else NAmes=0;
```

```
    if (Neighborhood = 'NridgHt') then NridgHt=1; else NridgHt=0;
    if (Neighborhood = 'OldTown') then OldTown=1; else OldTown=0;
    if (Neighborhood = 'SWISU') then SWISU=1; else SWISU=0;
    if (Neighborhood = 'Sawyer') then Sawyer=1; else Sawyer=0;
    if (Neighborhood = 'SawyerW') then SawyerW=1; else SawyerW=0;
    if (Neighborhood = 'Somerst') then Somerst=1; else Somerst=0;
    if (Neighborhood = 'StoneBr') then StoneBr=1; else StoneBr=0;
    if (Neighborhood = 'Timber') then Timber=1; else Timber=0;
    if (Neighborhood = 'Veenker') then Veenker=1; else Veenker=0;

    * KitchenQual dummy variable;
    if (KitchenQual in ('Ex', 'Gd')) then good_kitchen=1; else good_kitchen=0;

    * FireplaceQu dummy variable;
    if (FireplaceQu in ('Ex', 'Gd')) then good_fireplace=1; else good_fireplace=0;

    * ExterQual dummy variable;
    if (ExterQual in ('Ex', 'Gd')) then good_exterior=1; else good_exterior=0;

    * Foundation dummy variables;
    if (Foundation = 'BrkTil') then Foundation_BrkTil=1; else Foundation_BrkTil=0;
    if (Foundation = 'CBlock') then Foundation_CBlock=1; else Foundation_CBlock=0;
    if (Foundation = 'PConc') then Foundation_PConc=1; else Foundation_PConc=0;
    if (Foundation = 'Slab') then Foundation_Slab=1; else Foundation_Slab=0;
    if (Foundation = 'Stone') then Foundation_Stone=1; else Foundation_Stone=0;
    if (Foundation = 'Wood') then Foundation_Wood=1; else Foundation_Wood=0;

    * MoSold dummy variables;
    if (MoSold = 1) then jan_sold=1; else jan_sold=0;
    if (MoSold = 2) then feb_sold=1; else feb_sold=0;
    if (MoSold = 3) then mar_sold=1; else mar_sold=0;
    if (MoSold = 4) then apr_sold=1; else apr_sold=0;
    if (MoSold = 5) then may_sold=1; else may_sold=0;
    if (MoSold = 6) then jun_sold=1; else jun_sold=0;
    if (MoSold = 7) then jul_sold=1; else jul_sold=0;
    if (MoSold = 8) then aug_sold=1; else aug_sold=0;
    if (MoSold = 9) then sep_sold=1; else sep_sold=0;
    if (MoSold = 10) then oct_sold=1; else oct_sold=0;
    if (MoSold = 11) then nov_sold=1; else nov_sold=0;
    if (MoSold = 12) then dec_sold=1; else dec_sold=0;

    * Construct a composite quality index;
    quality_index = OverallCond*OverallQual;

    * Central Air Indicator;
    if (CentralAir='Y') then central_air=1; else central_air=0;
    * Fireplace Indicator;
    if (Fireplaces>0) then fireplace_ind=1; else fireplace_ind=0;
    * Garage Indicator;
    if (GarageCars>0) then garage_ind=1; else garage_ind=0;
    * Good Basement Indicator;
    if (BsmtQual in ('Ex','Gd')) or (BsmtCond in ('Ex','Gd'))
    then good_basement_ind=1;
    else good_basement_ind=0;

    *apply the put function to create the categorical variables for the various scales of price and sqft;
    price_cat = put(SalePrice,price_sfmt.);
    sqft_cat = put(total_SF,sqft_sfmt.);

    if (YearBuilt>1920);
run; quit;


* Review new dataset to ensure no missing values based on filtering;
proc contents data=building;
run; quit;


*---------------------------------------------------------------------------------------
*  Include all predictor variable candidates and identify outliers using the Studentized
*  Residuals
*---------------------------------------------------------------------------------------;

proc reg data=building;
    model SalePrice = YearBuilt total_SF total_baths good_kitchen good_fireplace good_exterior
                        quality_index central_air fireplace_ind garage_ind good_basement_ind;
    output out=outdata (keep = SalePrice YearBuilt total_SF total_baths good_kitchen
                                good_fireplace good_exterior quality_index central_air fireplace_ind
                                garage_ind good_basement_ind studentr) rstudent=studentr;
run;
```

```
* Assess the plot of the studentized residuals to see how many exceed an absolute value of 2;
proc univariate data=outdata plot;
    var studentr;
run;

* Remove any observations with an absolute studentized residual greater than 2;
data clean_data;
    SET outdata;
    if abs(studentr) > 2 then delete;
run;

data clean_data_2;
    set clean_data;
    * generate a uniform(0,1) random variable with seed set to 123;
    u = uniform(123);
    if (u < 0.70) then train = 1;
    else train = 0;
    if (train=1) then train_response=SalePrice;
    else train_response=.;
    if(train=0) then test_response=SalePrice;
    else test_response=.;
run;

*-----------------------------------------------------------------------------------------
*  Build a model using the adjusted r-squared variable selection: Model_AdjR2
*-----------------------------------------------------------------------------------------;

proc reg data=clean_data_2;
    model train_response = YearBuilt total_SF total_baths good_kitchen good_fireplace good_exterior
            quality_index central_air fireplace_ind garage_ind good_basement_ind /
    selection = adjrsq vif;
run;
* The Adjusted R2 selection method suggested to use all 11 predictor variables;
*-----------------------------------------------------------------------------------------;


*-----------------------------------------------------------------------------------------
*  Build a model using the adjusted r-squared variable selection: Model_MaxR
*-----------------------------------------------------------------------------------------;

proc reg data=clean_data_2;
    model train_response = YearBuilt total_SF total_baths good_kitchen good_fireplace good_exterior
            quality_index central_air fireplace_ind garage_ind good_basement_ind /
    selection =  MAXR vif;
run;
* The MaxR selection method suggested to use all 11 predictor variables;
*-----------------------------------------------------------------------------------------;


*-----------------------------------------------------------------------------------------
*  Build a model using the adjusted r-squared variable selection: Model_MCp
*-----------------------------------------------------------------------------------------;

proc reg data=clean_data_2;
    model train_response = YearBuilt total_SF total_baths good_kitchen good_fireplace
            good_exterior quality_index central_air fireplace_ind garage_ind good_basement_ind /
    selection = cp;
run;
* The Mallow's Cp selection method suggested to use all 11 predictor variables;


*-----------------------------------------------------------------------------------------
*  Build a model using the adjusted r-squared variable selection: Model_F
*-----------------------------------------------------------------------------------------;

proc reg data=clean_data_2;
    model train_response = YearBuilt total_SF total_baths good_kitchen good_fireplace good_exterior
            quality_index central_air fireplace_ind garage_ind good_basement_ind /
    selection = FORWARD slentry=.15;
run;
* The Forward selection method suggested to use all 11 predictor variables;
*-----------------------------------------------------------------------------------------;


*-----------------------------------------------------------------------------------------
*  Build a model using the adjusted r-squared variable selection: Model_B
*-----------------------------------------------------------------------------------------;

proc reg data=clean_data_2;
    model train_response = YearBuilt total_SF total_baths good_kitchen good_fireplace good_exterior
            quality_index central_air fireplace_ind garage_ind good_basement_ind /
```

```
    selection = BACKWARD slstay=.15;
run;
* The Backward selection method suggested to use all 11 predictor variables;
*-------------------------------------------------------------------------------------;


*-------------------------------------------------------------------------------------
*  Build a model using the adjusted r-squared variable selection: Model_S
*-------------------------------------------------------------------------------------;

proc reg data=clean_data_2;
    model train_response = YearBuilt total_SF total_baths good_kitchen good_fireplace good_exterior
            quality_index central_air fireplace_ind garage_ind good_basement_ind /
    selection = STEPWISE aic bic mse adjrsq slentry=.15 slstay=.15;
run;
* The Stepwise selection method suggested to use all 11 predictor variablse;
*-------------------------------------------------------------------------------------;

*-------------------------------------------------------------------------------------
*  Calculate the AIC, BIC, Adjusted R-squared, Mallow's Cp, MSE for the training sample
*-------------------------------------------------------------------------------------;
proc reg data=clean_data_2;
    model train_response = YearBuilt total_SF total_baths good_kitchen good_fireplace good_exterior
            quality_index central_air fireplace_ind garage_ind good_basement_ind /
    selection = adjrsq aic bic cp mse;
run;

*-------------------------------------------------------------------------------------
*  Calculate the Mean Absolute Error for the training sample
*-------------------------------------------------------------------------------------;
proc reg data=clean_data_2;
    model train_response = YearBuilt total_SF total_baths good_kitchen good_fireplace good_exterior
            quality_index central_air fireplace_ind garage_ind good_basement_ind;
    output out=residuals_final (keep = resid_final) r=resid_final;
run;

data abs_resid;
    set residuals_final;
    abs_resid = abs(resid_final);
run;

proc means data=abs_resid mean;
    var abs_resid;
run;

*-------------------------------------------------------------------------------------
*  Use the suggested model that includes all 11 predictor candidates and
*  perform cross validation by outputting the estimated values from the test set
*-------------------------------------------------------------------------------------;
proc reg data=clean_data_2 outest=RegOut;
    SalePrice_Hat: model train_response = YearBuilt total_SF total_baths good_kitchen
                            good_fireplace good_exterior quality_index central_air fireplace_ind
                            garage_ind good_basement_ind;
    title "Model with all 11 Predictor Candidates";
run;

proc print data=RegOut;
      title2 'OUTEST= Data Set from PROC REG';
run;

* calcualte the estimated values using the test dataset;
proc score data=clean_data_2 score=RegOut out=RScoreP type=parms;
      var YearBuilt total_SF total_baths good_kitchen good_fireplace good_exterior quality_index
                                            central_air fireplace_ind garage_ind good_basement_ind;
run;

proc score data=clean_data_2 score=RegOut out=RScoreR type=parms;
      var train_response YearBuilt total_SF total_baths good_kitchen good_fireplace good_exterior
                                            quality_index central_air fireplace_ind garage_ind good_basement_ir
run;

* Output the scores using the model built with the training dataset against the test dataset;
proc score data=test_dataset score=RegOut out=NewPred type=parms
              nostd predict;
      var train_response YearBuilt total_SF total_baths good_kitchen good_fireplace good_exterior
                                            quality_index central_air fireplace_ind garage_ind good_basement_ir
run;

* Smart data formats for sale price and square footage;
proc format;
value pred_acc_sfmt
```

```
    0 -< .1 = '01: Grade 1'
    .1 -< .15 = '02: Grade 2'
    other = '03: Grade 3'
    ;
run;

*create a new dataset that contains the test set data and the predictive scores;
data prediction_Data;
    set NEWPRED;
    if (train_response = null);
    pred_score = abs(test_response / SalePrice_Hat - 1);
    abs_resid = abs(test_response - SalePrice_Hat);
    error_term = test_response - SalePrice_Hat;
    sq_error = error_term**2;
    Prediction_Grade = put(pred_score,pred_acc_sfmt.);
run;

* calculate the MAE and MSE from the test sample;
proc means data=prediction_Data mean;
    var abs_resid sq_error;
run;

* create a frequency table to show the operational accuracy of the model;
proc freq data=prediction_Data;
    TABLES Prediction_Grade;
run;
```