*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-GrLivArea

X2-LotArea

X3-AgeAtSale

X4-TotalBsmtSF

X5-total\_baths\_calc

X6-TotRmsAbvGrd

X7-highend\_ind Neighborhood grouping

X8-midend\_ind Neighborhood grouping

X9-good\_heating

X10-excl\_kitchen kitchen quality

X11-central\_air

X12-fireplace\_ind

X13-garage\_ind

X14-good\_basement\_ind

X15-concr\_foundation

X16-quality\_index

X17-brick\_exterior

X18-lot\_frontage

X19-new\_bldg

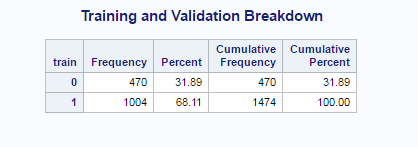
X20-old\_bldg

X21-pos\_cond

X22-recent\_remodel

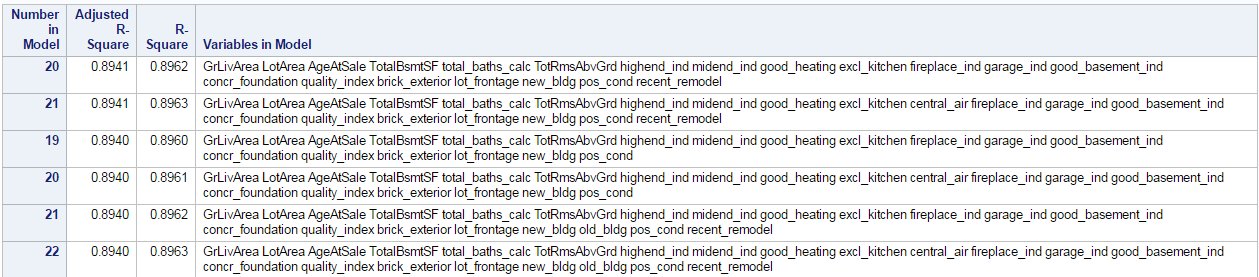
*Training data set*

The data set has been split in test/train data set as shown below. We will test the model accuracy by training the model on the 70% of the data set and validating its accuracy on remaining 30%.



*Adjusted R-Squared Model (Model\_AdjR2)*

The results of the adjusted R-squared variable selection method determined that not all predictor variables candidates (X1-X22) should be included in the final model. The adjusted R-squared of the final model was 0.8941. Table 1 below, you can see the summary output of the variable selection process with the final result in the first row.



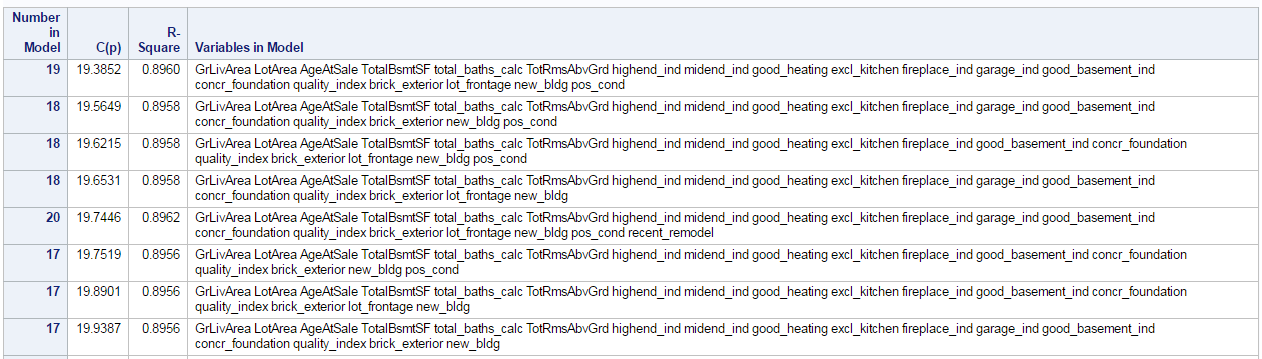
The adjusted R-squared model suggest to exclude variables central\_air(X11) and old\_bldg(X20).

*Maximum R-Squared Model (Model\_MaxR)*

Like the adjusted R-squared variable selection method, the results of the maximum R-squared variable selection method determined that not all predictor variable candidates (X1-X22) should be included in the final model. The R-squared of the final model was 0.8963. In Table below, we can see the ANOVA and parameter estimates for the final suggested model from the maximum R-squared variable selection method. The model suggests excluding variables central\_air(X11) and old\_bldg(X20) as none of them is statistically significant.



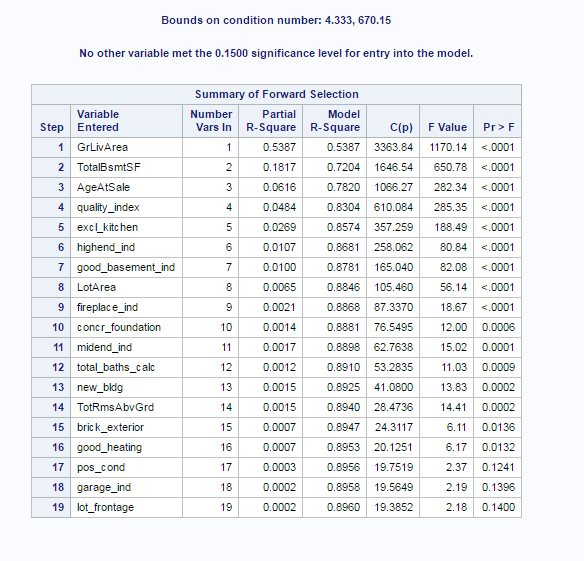
*Mallow’s Cp Model (Model\_MCp)*

As with the previous two models, the results of the Mallow’s Cp variable selection method also determined that not all predictor variable candidates (X1-X22) should be included in the final model. The Mallow’s Cp of the final model was 19.3852. 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. 

Apart from old\_bldg(X20) and central\_air(X11) the model suggest to exclude recent\_remodel(X22) variable.

*Forward Selection Model (Model\_F)*

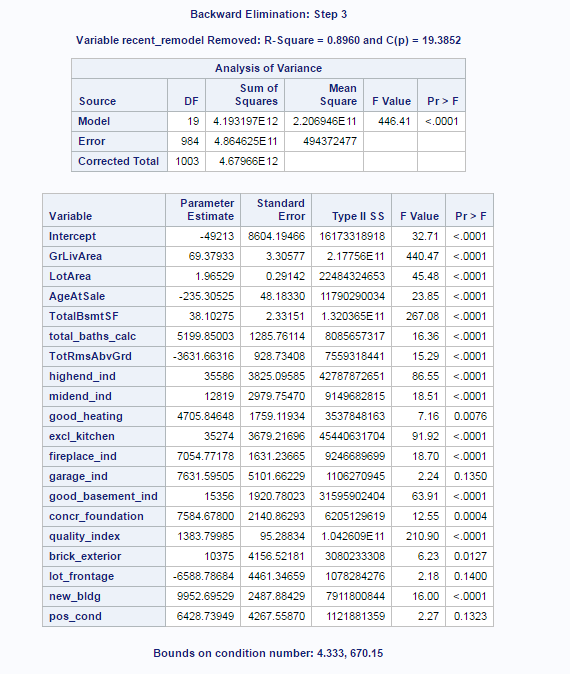
Like proceeding variable selection methods, the results of the forward variable selection method also determined that not all predictor variable candidates (X1-X22) should be included in the final model. In table below, we can see the summary of the forward selection method and you’ll notice that the p-value from the nested F-tests did not increase until the 10th 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.

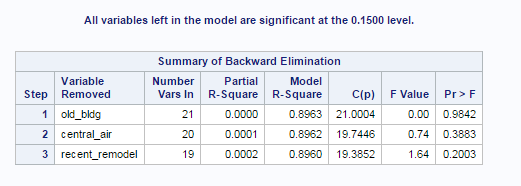


The model suggest exclusion of central\_air(X11), old\_bldg(X20), recent\_remodel(X22) variables

*Backward Model (Model\_B)*

Also this time the results of the backward variable selection method determined that not all predictor variable candidates (X1-X22) should be included in the final model. In table below, we can see the summary of the beckward elimination method. For this method, we chose a *slentry* value of 0.15 as our threshold for variables to be allowed to enter into the model.

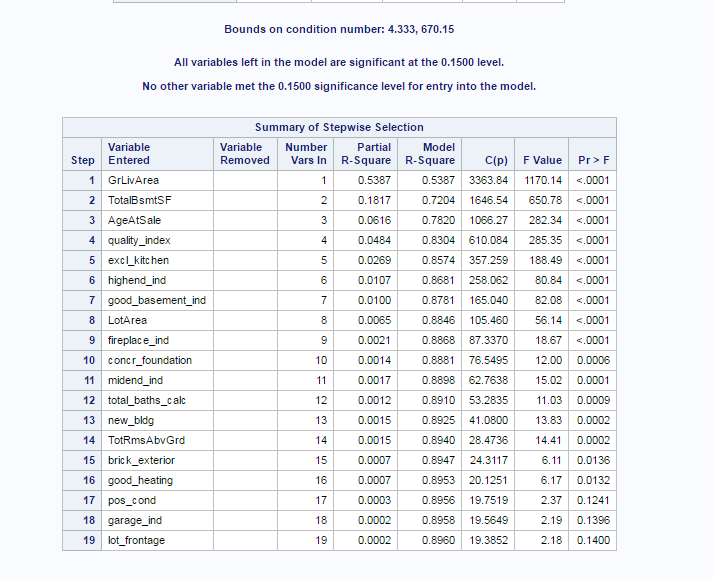




The model suggest exclusion of central\_air(X11), old\_bldg(X20), recent\_remodel(X22) variables

*Stepwise Selection Model (Model\_S)*

The stepwise selection method was the final option used for model selection. As with previous methods, the stepwise selection method indicated that not all predictor variable candidates should remain in the model. The stepwise variable selection summary is shown below.



At the 0.1500 level variable X20-X22 were excluded from the model.

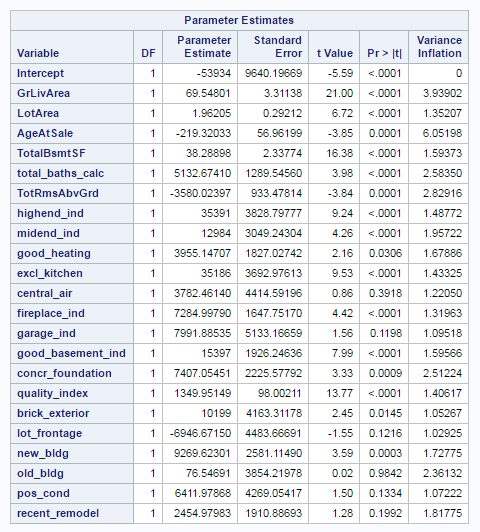
*Model Comparison*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Model\_AdjR2 | Model\_MaxR | Model\_MCp | Model\_F | Model\_B | Model\_S |
| Train | Predictors  excluded | X11,X20 | X11,X20 | X11,X20,X22 | X11,X20,X22 | X11,X20,X22 | X11,X20,X22 |
| Train | Adjusted R2 | 0.8941 | 0.8940 | 0.8940 | 0.8940 | 0.8940 | 0.8940 |
| Train | AIC | 20118.9967 | 20122.23 | 20118.67 | 20118.67 | 20118.67 | 20118.67 |
| Train | BIC | 20121.9478 | 20125.31 | 20121.51 | 20121.51 | 20121.51 | 20121.51 |
| Train | Mallow’s Cp | 19.7446 | 23 | 19.3852 | 19.3852 | 19.3852 | 19.3852 |
| Train | MAE | 15879.15 | 15866.46 | 15907.54 | 15907.54 | 15907.54 | 15907.54 |
| Train | MSE | 483716094 | 483349211 | 484524420 | 484524420 | 484524420 | 484524420 |
| Test | MAE | 15052.86 | 15016.78 | 15107.78 | 15107.78 | 15107.78 | 15107.78 |
| Test | MSE | 426320079 | 424752379 | 430209443 | 430209443 | 430209443 | 430209443 |

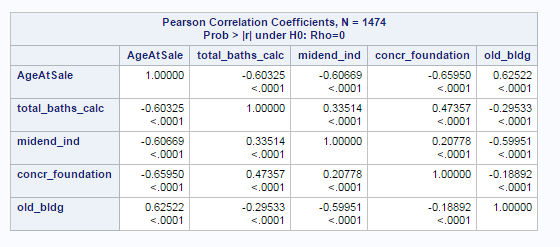
There has been slight variation in variable selection between models . Table below shows model fit criteria from the models created using the training sample. Based on the results, we can see that the predictive ability of the final model built from the training sample performed very well with the test sample data.

*Multicollinearity*

Values 10 or above for variance inflation analysis may cause serious problem for data analysis. It seems that there are no immediate risk for multicollinearity even though the value for AgeAtSale is a little bit high.

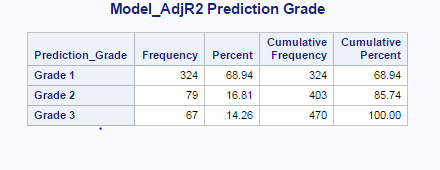


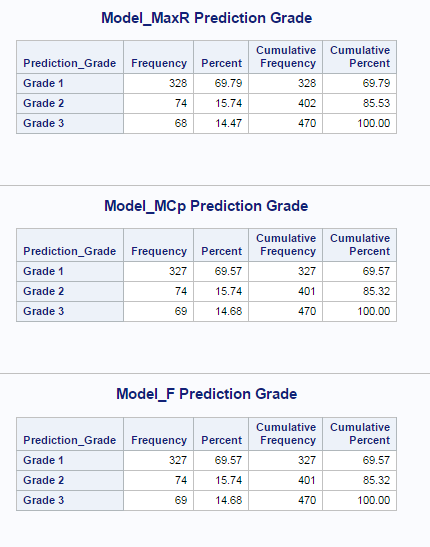
Pearson Correlation analysis reveals that correlation between AgeAtSale and other variables is within -0.7 and 0.7. The highest value of 0.65 is between AgeAtSale and concr\_foundation. The table below shows variables with highest Pearson Correlation.

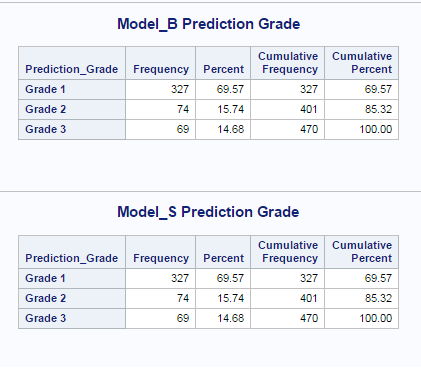


*Operational validation*

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). Below the results for each one of the models.







Based on the results of the prediction scores above, we can see that 85% of the predicted values were within 15% of the observed value.

*Conclusion*

In order to see effects of the automated variable selection I had to choose quite a few variables. The model might be hard to interpret for the business users, although it performs quite well. It has been proven that simpler models repeatedly run on larger data sets perform better in the long run.

It is quite clear how automated variable selection can be valuable when designing a predictive model. That should be combined with domain knowledge of stakeholders in order to achieve the best results.

libname mydata '/libname mydata '/scs/crb519/PREDICT\_410/SAS\_Data/' access=readonly;

ods graphics on;

ods noproctitle;

title 'Assignment 5';

\*data set A method b for defining variables without YearBuilt funkar utan yb 10 variabler

\* create water drop condition to establish sample;

data building;

set mydata.ames\_housing\_data;

format drop\_condition $40.;

if (BldgType ne '1Fam') then drop\_condition='01: Not a Single Family';

else if (Zoning not in ('RH','RL','RM','FV')) then drop\_condition='02: Non-Residential Zoning';

else if (GrLivArea > 4000 and SalePrice < 200000) then drop\_condition='03: Large Area and Low Sale Price';

else if (FullBath < 1) then drop\_condition='04: No Bath';

else if (LotArea > 20000) then drop\_condition='05: Lot Area Over 20,000sqf';

else if (SalePrice > 700000) then drop\_condition='06: Sale Price Over $700,000';

else if (SalePrice < 15000) then drop\_condition='07: Sale Price Below $15,000';

else if (GrLivArea < 1000) then drop\_condition='08: Living Area Below 1000sqf';

else if (GrLivArea > 3000) then drop\_condition='09: Living Area Above 3000sqf';

else if (AgeAtSale > 100) then drop\_condition='10: Age Over 100';

else if (Functional ne 'Typ') then drop\_condition='11: Non-Typcial Function';

else if (SaleCondition ne 'Normal') then drop\_condition='12: Non-Normal Sale';

else drop\_condition='99: Sample Population';

AgeAtSale = YrSold - YearBuilt;

logSalePrice = log(SalePrice);

run;

proc freq data=building;

tables drop\_condition;

title 'Sample Waterfall';

run;

data train;

set building

(where = (drop\_condition = '99: Sample Population'));

u = uniform(123);

if (u < 0.70) then train = 1; else train = 0;

if (train=1) then train\_response=SalePrice; else train\_response=.;

\* create derived variables;

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;

\*Kitchen categorical variables;

if (KitchenQual = 'Ex') then excl\_kitchen = 1; else excl\_kitchen = 0;

\*Fireplace;

if (Fireplaces>0) then fireplace\_ind=1; else fireplace\_ind=0;

\*Exterior;

if (ExterQual in ('Ex', 'Gd')) then good\_exterior=1; else good\_exterior=0;

\*Neighborhood variables;

if (Neighborhood in ('StoneBr','NridgHt','Greens','GrnHill')) then

highend\_ind = 1;

else

highend\_ind = 0;

if (Neighborhood in ('BrDale','IDOTRR','MeadowV','OldTown','SWISU')) then

midend\_ind = 0;

else

midend\_ind = 1;

\*Foundation;

if (Foundation = 'PConc') then concr\_foundation = 1; else concr\_foundation = 0;

quality\_index = OverallCond\*OverallQual;

if(CentralAir='Y') then central\_air=1; else central\_air=0;

if (GarageCars>0) then garage\_ind=1; else garage\_ind=0;

if (BsmtQual in ('Ex', 'Gd')) or (BmstCond in ('Ex', 'Gd'))

then good\_basement\_ind=1;

else good\_basement\_ind=0;

if (BsmtFinType1 = 'GLQ') then good\_basement\_ind=1; else good\_basement\_ind=0;

if (HeatingQC = 'Ex') then good\_heating = 1; else good\_heating = 0;

if (Exterior1 in ('BrkComm','BrkFace')) or (Exterior2 in ('BrkComm','BrkFace'))

then brick\_exterior=1; else brick\_exterior=0;

if (LotConfig in ('FR2','FR3')) then lot\_frontage=1; else lot\_frontage=0;

if (AgeAtSale < 5) then new\_bldg = 1; else new\_bldg = 0;

if (AgeAtSale > 85) then old\_bldg = 1; else old\_bldg = 0;

if (Condition1 in ('PosA','PosN') or Condition2 in ('PosA','PosN')) then pos\_cond = 1; else pos\_cond = 0;

YearsSinceRemodel = YrSold - YearRemodel;

if (YearsSinceRemodel >= 0 and YearsSinceRemodel <= 10) then recent\_remodel = 1; else recent\_remodel = 0;

run;

\* Model\_AdjR2: use automatic variable selection with adjusted R squre ;

proc reg data = train plots = diagnostics(unpack) outest = Model\_AdjR2\_est;

Model\_AdjR2: model train\_response =

GrLivArea LotArea AgeAtSale TotalBsmtSF total\_baths\_calc

TotRmsAbvGrd highend\_ind midend\_ind good\_heating excl\_kitchen

central\_air fireplace\_ind garage\_ind good\_basement\_ind concr\_foundation

quality\_index brick\_exterior lot\_frontage new\_bldg old\_bldg pos\_cond recent\_remodel

/ selection=adjrsq best=1 aic bic cp vif;

output out=Model\_AdjR2\_out (keep=train SalePrice yhat) predicted=yhat;

title 'Model\_AdjR2';

run;

\* Model\_MaxR: use automatic variable selection with MaxR ;

proc reg data = train plots = diagnostics(unpack) outest = Model\_MaxR\_est;

Model\_MaxR: model train\_response =

GrLivArea LotArea AgeAtSale TotalBsmtSF total\_baths\_calc

TotRmsAbvGrd highend\_ind midend\_ind good\_heating excl\_kitchen

central\_air fireplace\_ind garage\_ind good\_basement\_ind concr\_foundation

quality\_index brick\_exterior lot\_frontage new\_bldg old\_bldg pos\_cond recent\_remodel

/ best=1 selection=maxr aic bic cp vif;

output out=Model\_MaxR\_out (keep=train SalePrice yhat) predicted=yhat;

title 'Model\_MaxR';

run;

\*-------------------------------------------------------------------------------------------

\* Modle\_MCp

\*-------------------------------------------------------------------------------------------;

\* Model\_MCp: use automatic variable selection with Minimal Mallow's Cp ;

proc reg data = train plots = diagnostics(unpack) outest = Model\_MCp\_est;

Model\_MCp: model train\_response =

GrLivArea LotArea AgeAtSale TotalBsmtSF total\_baths\_calc

TotRmsAbvGrd highend\_ind midend\_ind good\_heating excl\_kitchen

central\_air fireplace\_ind garage\_ind good\_basement\_ind concr\_foundation

quality\_index brick\_exterior lot\_frontage new\_bldg old\_bldg pos\_cond recent\_remodel

/ best=1 selection=cp best=1 aic bic cp vif;

output out=Model\_MCp\_out (keep=train SalePrice yhat) predicted=yhat;

title 'Model\_MCp';

run;

\* Model\_F: use automatic variable selection with forward selection ;

proc reg data = train plots = diagnostics(unpack) outest = Model\_F\_est;

Model\_F: model train\_response =

GrLivArea LotArea AgeAtSale TotalBsmtSF total\_baths\_calc

TotRmsAbvGrd highend\_ind midend\_ind good\_heating excl\_kitchen

central\_air fireplace\_ind garage\_ind good\_basement\_ind concr\_foundation

quality\_index brick\_exterior lot\_frontage new\_bldg old\_bldg pos\_cond recent\_remodel

/ best=1 selection=forward slentry=0.15 aic bic cp vif;

output out=Model\_F\_out (keep=train SalePrice yhat) predicted=yhat;

title 'Model\_F';

run;

\* Model\_B: use automatic variable selection with backward selection ;

proc reg data = train plots = diagnostics(unpack) outest = Model\_B\_est;

Model\_B: model train\_response =

GrLivArea LotArea AgeAtSale TotalBsmtSF total\_baths\_calc

TotRmsAbvGrd highend\_ind midend\_ind good\_heating excl\_kitchen

central\_air fireplace\_ind garage\_ind good\_basement\_ind concr\_foundation

quality\_index brick\_exterior lot\_frontage new\_bldg old\_bldg pos\_cond recent\_remodel

/ selection=backward slstay=0.15 best=1 aic bic cp vif;

output out=Model\_B\_out (keep=train SalePrice yhat) predicted=yhat;

title 'Model\_B';

run;

\* Model\_S: use automatic variable selection with stepwise selection ;

proc reg data = train plots = diagnostics(unpack) outest = Model\_S\_est;

Model\_S: model train\_response =

GrLivArea LotArea AgeAtSale TotalBsmtSF total\_baths\_calc

TotRmsAbvGrd highend\_ind midend\_ind good\_heating excl\_kitchen

central\_air fireplace\_ind garage\_ind good\_basement\_ind concr\_foundation

quality\_index brick\_exterior lot\_frontage new\_bldg old\_bldg pos\_cond recent\_remodel

/ best=1 selection=stepwise slentry=0.15 slstay=0.15 aic bic cp vif;

output out=Model\_S\_out (keep=train SalePrice yhat) predicted=yhat;

title 'Model\_S';

run;

\* create macros for printing the estimators and calculating MSE and MAE;

%macro print\_est(indata,mytitle);

proc print data=&indata.;

title &mytitle.;

run; quit;

%mend;

%macro data\_validation(indata,outdata);

data &outdata.;

set &indata.;

err = SalePrice - yhat;

abserr = abs(SalePrice - yhat);

sqerr = (SalePrice - yhat) \*\* 2;

if (abserr <= SalePrice \* 0.1) then

Prediction\_Grade = 'Grade 1';

else if (abserr <= SalePrice \* 0.15) then

Prediction\_Grade = 'Grade 2';

else

Prediction\_Grade = 'Grade 3';

run; quit;

%mend;

%macro print\_grade(indata,mytitle);

proc freq data=&indata.;

tables Prediction\_Grade;

where train=0;

title &mytitle.;

run; quit;

%mend;

%macro print\_mse\_mae(indata,mytitle);

proc means data=&indata. mean min max median sum;

var abserr sqerr;

class train;

title &mytitle;

run; quit;

%mend;

\* print model estimates for all six models ;

%print\_est(indata=%str(Model\_AdjR2\_est),mytitle=%str('Model\_AdjR2 estimator'));

%print\_est(indata=%str(Model\_MaxR\_est),mytitle=%str('Model\_MaxR estimator'));

%print\_est(indata=%str(Model\_MCp\_est),mytitle=%str('Model\_MCp estimator'));

%print\_est(indata=%str(Model\_F\_est),mytitle=%str('Model\_F estimator'));

%print\_est(indata=%str(Model\_B\_est),mytitle=%str('Model\_B estimator'));

%print\_est(indata=%str(Model\_S\_est),mytitle=%str('Model\_S estimator'));

\* run model in-sample and out-of-sample validation and print results ;

%data\_validation(indata=%str(Model\_AdjR2\_out),outdata=%str(Model\_AdjR2\_validation));

%print\_mse\_mae(indata=%str(Model\_AdjR2\_validation),mytitle=%str('Model\_AdjR2 MSE MAE'));

%data\_validation(indata=%str(Model\_MaxR\_out),outdata=%str(Model\_MaxR\_validation));

%print\_mse\_mae(indata=%str(Model\_MaxR\_validation),mytitle=%str('Model\_MaxR MSE MAE'));

%data\_validation(indata=%str(Model\_MCp\_out),outdata=%str(Model\_MCp\_validation));

%print\_mse\_mae(indata=%str(Model\_MCp\_validation),mytitle=%str('Model\_MCp MSE MAE'));

%data\_validation(indata=%str(Model\_F\_out),outdata=%str(Model\_F\_validation));

%print\_mse\_mae(indata=%str(Model\_F\_validation),mytitle=%str('Model\_F MSE MAE'));

%data\_validation(indata=%str(Model\_B\_out),outdata=%str(Model\_B\_validation));

%print\_mse\_mae(indata=%str(Model\_B\_validation),mytitle=%str('Model\_B MSE MAE'));

%data\_validation(indata=%str(Model\_S\_out),outdata=%str(Model\_S\_validation));

%print\_mse\_mae(indata=%str(Model\_S\_validation),mytitle=%str('Model\_S MSE MAE'));

\* print model operational grades ;

%print\_grade(indata=%str(Model\_AdjR2\_validation),mytitle=%str('Model\_AdjR2 Prediction Grade'));

%print\_grade(indata=%str(Model\_MaxR\_validation),mytitle=%str('Model\_MaxR Prediction Grade'));

%print\_grade(indata=%str(Model\_MCp\_validation),mytitle=%str('Model\_MCp Prediction Grade'));

%print\_grade(indata=%str(Model\_F\_validation),mytitle=%str('Model\_F Prediction Grade'));

%print\_grade(indata=%str(Model\_B\_validation),mytitle=%str('Model\_B Prediction Grade'));

%print\_grade(indata=%str(Model\_S\_validation),mytitle=%str('Model\_S Prediction Grade'));

\* run a final operationa validation of the model produced by F/S/B;

proc reg data = train plots = diagnostics(unpack) outest = Model\_Final\_est;

Model\_Final: model train\_response =

GrLivArea LotArea AgeAtSale TotalBsmtSF total\_baths\_calc

TotRmsAbvGrd highend\_ind midend\_ind good\_heating excl\_kitchen

fireplace\_ind good\_basement\_ind concr\_foundation quality\_index

brick\_exterior new\_bldg;

output out=Model\_Final\_out (keep=train SalePrice YrSold yhat) predicted=yhat;

title 'Model\_Final';

run;

%data\_validation(indata=%str(Model\_Final\_out),outdata=%str(Model\_Final\_validation));

\*Multicollinearity;

proc reg data=train;

model train\_response =

GrLivArea LotArea AgeAtSale TotalBsmtSF total\_baths\_calc

TotRmsAbvGrd highend\_ind midend\_ind good\_heating excl\_kitchen

central\_air fireplace\_ind garage\_ind good\_basement\_ind concr\_foundation

quality\_index brick\_exterior lot\_frontage new\_bldg old\_bldg pos\_cond recent\_remodel / corrb vif collin;

ods graphics on;

title 'Correlation matrix ';

proc corr data=train plots=scatter(alpha=.20 .30);

var GrLivArea LotArea AgeAtSale TotalBsmtSF total\_baths\_calc

TotRmsAbvGrd highend\_ind midend\_ind good\_heating excl\_kitchen

central\_air fireplace\_ind garage\_ind good\_basement\_ind concr\_foundation

quality\_index brick\_exterior lot\_frontage new\_bldg old\_bldg pos\_cond recent\_remodel ;

run;

ods graphics off;

ods graphics on;

title 'Correlation matrix ';

proc corr data=train plots=scatter(alpha=.20 .30);

var AgeAtSale total\_baths\_calc

midend\_ind

concr\_foundation

old\_bldg ;

run;

ods graphics off;scs/crb519/PREDICT\_410/SAS\_Data/' access=readonly;

ods graphics on;

ods noproctitle;

title 'Assignment 5';

\*data set A method b for defining variables without YearBuilt funkar utan yb 10 variabler

\* create water drop condition to establish sample;

data building;

set mydata.ames\_housing\_data;

format drop\_condition $40.;

if (BldgType ne '1Fam') then drop\_condition='01: Not a Single Family';

else if (Zoning not in ('RH','RL','RM','FV')) then drop\_condition='02: Non-Residential Zoning';

else if (GrLivArea > 4000 and SalePrice < 200000) then drop\_condition='03: Large Area and Low Sale Price';

else if (FullBath < 1) then drop\_condition='04: No Bath';

else if (LotArea > 20000) then drop\_condition='05: Lot Area Over 20,000sqf';

else if (SalePrice > 700000) then drop\_condition='06: Sale Price Over $700,000';

else if (SalePrice < 15000) then drop\_condition='07: Sale Price Below $15,000';

else if (GrLivArea < 1000) then drop\_condition='08: Living Area Below 1000sqf';

else if (GrLivArea > 3000) then drop\_condition='09: Living Area Above 3000sqf';

else if (AgeAtSale > 100) then drop\_condition='10: Age Over 100';

else if (Functional ne 'Typ') then drop\_condition='11: Non-Typcial Function';

else if (SaleCondition ne 'Normal') then drop\_condition='12: Non-Normal Sale';

else drop\_condition='99: Sample Population';

AgeAtSale = YrSold - YearBuilt;

logSalePrice = log(SalePrice);

run;

proc freq data=building;

tables drop\_condition;

title 'Sample Waterfall';

run;

data train;

set building

(where = (drop\_condition = '99: Sample Population'));

u = uniform(123);

if (u < 0.70) then train = 1; else train = 0;

if (train=1) then train\_response=SalePrice; else train\_response=.;

\* create derived variables;

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;

\*Kitchen categorical variables;

if (KitchenQual = 'Ex') then excl\_kitchen = 1; else excl\_kitchen = 0;

\*Fireplace;

if (Fireplaces>0) then fireplace\_ind=1; else fireplace\_ind=0;

\*Exterior;

if (ExterQual in ('Ex', 'Gd')) then good\_exterior=1; else good\_exterior=0;

\*Neighborhood variables;

if (Neighborhood in ('StoneBr','NridgHt','Greens','GrnHill')) then

highend\_ind = 1;

else

highend\_ind = 0;

if (Neighborhood in ('BrDale','IDOTRR','MeadowV','OldTown','SWISU')) then

midend\_ind = 0;

else

midend\_ind = 1;

\*Foundation;

if (Foundation = 'PConc') then concr\_foundation = 1; else concr\_foundation = 0;

quality\_index = OverallCond\*OverallQual;

if(CentralAir='Y') then central\_air=1; else central\_air=0;

if (GarageCars>0) then garage\_ind=1; else garage\_ind=0;

if (BsmtQual in ('Ex', 'Gd')) or (BmstCond in ('Ex', 'Gd'))

then good\_basement\_ind=1;

else good\_basement\_ind=0;

if (BsmtFinType1 = 'GLQ') then good\_basement\_ind=1; else good\_basement\_ind=0;

if (HeatingQC = 'Ex') then good\_heating = 1; else good\_heating = 0;

if (Exterior1 in ('BrkComm','BrkFace')) or (Exterior2 in ('BrkComm','BrkFace'))

then brick\_exterior=1; else brick\_exterior=0;

if (LotConfig in ('FR2','FR3')) then lot\_frontage=1; else lot\_frontage=0;

if (AgeAtSale < 5) then new\_bldg = 1; else new\_bldg = 0;

if (AgeAtSale > 85) then old\_bldg = 1; else old\_bldg = 0;

if (Condition1 in ('PosA','PosN') or Condition2 in ('PosA','PosN')) then pos\_cond = 1; else pos\_cond = 0;

YearsSinceRemodel = YrSold - YearRemodel;

if (YearsSinceRemodel >= 0 and YearsSinceRemodel <= 10) then recent\_remodel = 1; else recent\_remodel = 0;

run;

\* Model\_AdjR2: use automatic variable selection with adjusted R squre ;

proc reg data = train plots = diagnostics(unpack) outest = Model\_AdjR2\_est;

Model\_AdjR2: model train\_response =

GrLivArea LotArea AgeAtSale TotalBsmtSF total\_baths\_calc

TotRmsAbvGrd highend\_ind midend\_ind good\_heating excl\_kitchen

central\_air fireplace\_ind garage\_ind good\_basement\_ind concr\_foundation

quality\_index brick\_exterior lot\_frontage new\_bldg old\_bldg pos\_cond recent\_remodel

/ selection=adjrsq best=1 aic bic cp vif;

output out=Model\_AdjR2\_out (keep=train SalePrice yhat) predicted=yhat;

title 'Model\_AdjR2';

run;

\* Model\_MaxR: use automatic variable selection with MaxR ;

proc reg data = train plots = diagnostics(unpack) outest = Model\_MaxR\_est;

Model\_MaxR: model train\_response =

GrLivArea LotArea AgeAtSale TotalBsmtSF total\_baths\_calc

TotRmsAbvGrd highend\_ind midend\_ind good\_heating excl\_kitchen

central\_air fireplace\_ind garage\_ind good\_basement\_ind concr\_foundation

quality\_index brick\_exterior lot\_frontage new\_bldg old\_bldg pos\_cond recent\_remodel

/ best=1 selection=maxr aic bic cp vif;

output out=Model\_MaxR\_out (keep=train SalePrice yhat) predicted=yhat;

title 'Model\_MaxR';

run;

\*-------------------------------------------------------------------------------------------

\* Modle\_MCp

\*-------------------------------------------------------------------------------------------;

\* Model\_MCp: use automatic variable selection with Minimal Mallow's Cp ;

proc reg data = train plots = diagnostics(unpack) outest = Model\_MCp\_est;

Model\_MCp: model train\_response =

GrLivArea LotArea AgeAtSale TotalBsmtSF total\_baths\_calc

TotRmsAbvGrd highend\_ind midend\_ind good\_heating excl\_kitchen

central\_air fireplace\_ind garage\_ind good\_basement\_ind concr\_foundation

quality\_index brick\_exterior lot\_frontage new\_bldg old\_bldg pos\_cond recent\_remodel

/ best=1 selection=cp best=1 aic bic cp vif;

output out=Model\_MCp\_out (keep=train SalePrice yhat) predicted=yhat;

title 'Model\_MCp';

run;

\* Model\_F: use automatic variable selection with forward selection ;

proc reg data = train plots = diagnostics(unpack) outest = Model\_F\_est;

Model\_F: model train\_response =

GrLivArea LotArea AgeAtSale TotalBsmtSF total\_baths\_calc

TotRmsAbvGrd highend\_ind midend\_ind good\_heating excl\_kitchen

central\_air fireplace\_ind garage\_ind good\_basement\_ind concr\_foundation

quality\_index brick\_exterior lot\_frontage new\_bldg old\_bldg pos\_cond recent\_remodel

/ best=1 selection=forward slentry=0.15 aic bic cp vif;

output out=Model\_F\_out (keep=train SalePrice yhat) predicted=yhat;

title 'Model\_F';

run;

\* Model\_B: use automatic variable selection with backward selection ;

proc reg data = train plots = diagnostics(unpack) outest = Model\_B\_est;

Model\_B: model train\_response =

GrLivArea LotArea AgeAtSale TotalBsmtSF total\_baths\_calc

TotRmsAbvGrd highend\_ind midend\_ind good\_heating excl\_kitchen

central\_air fireplace\_ind garage\_ind good\_basement\_ind concr\_foundation

quality\_index brick\_exterior lot\_frontage new\_bldg old\_bldg pos\_cond recent\_remodel

/ selection=backward slstay=0.15 best=1 aic bic cp vif;

output out=Model\_B\_out (keep=train SalePrice yhat) predicted=yhat;

title 'Model\_B';

run;

\* Model\_S: use automatic variable selection with stepwise selection ;

proc reg data = train plots = diagnostics(unpack) outest = Model\_S\_est;

Model\_S: model train\_response =

GrLivArea LotArea AgeAtSale TotalBsmtSF total\_baths\_calc

TotRmsAbvGrd highend\_ind midend\_ind good\_heating excl\_kitchen

central\_air fireplace\_ind garage\_ind good\_basement\_ind concr\_foundation

quality\_index brick\_exterior lot\_frontage new\_bldg old\_bldg pos\_cond recent\_remodel

/ best=1 selection=stepwise slentry=0.15 slstay=0.15 aic bic cp vif;

output out=Model\_S\_out (keep=train SalePrice yhat) predicted=yhat;

title 'Model\_S';

run;

\* create macros for printing the estimators and calculating MSE and MAE;

%macro print\_est(indata,mytitle);

proc print data=&indata.;

title &mytitle.;

run; quit;

%mend;

%macro data\_validation(indata,outdata);

data &outdata.;

set &indata.;

err = SalePrice - yhat;

abserr = abs(SalePrice - yhat);

sqerr = (SalePrice - yhat) \*\* 2;

if (abserr <= SalePrice \* 0.1) then

Prediction\_Grade = 'Grade 1';

else if (abserr <= SalePrice \* 0.15) then

Prediction\_Grade = 'Grade 2';

else

Prediction\_Grade = 'Grade 3';

run; quit;

%mend;

%macro print\_grade(indata,mytitle);

proc freq data=&indata.;

tables Prediction\_Grade;

where train=0;

title &mytitle.;

run; quit;

%mend;

%macro print\_mse\_mae(indata,mytitle);

proc means data=&indata. mean min max median sum;

var abserr sqerr;

class train;

title &mytitle;

run; quit;

%mend;

\* print model estimates for all six models ;

%print\_est(indata=%str(Model\_AdjR2\_est),mytitle=%str('Model\_AdjR2 estimator'));

%print\_est(indata=%str(Model\_MaxR\_est),mytitle=%str('Model\_MaxR estimator'));

%print\_est(indata=%str(Model\_MCp\_est),mytitle=%str('Model\_MCp estimator'));

%print\_est(indata=%str(Model\_F\_est),mytitle=%str('Model\_F estimator'));

%print\_est(indata=%str(Model\_B\_est),mytitle=%str('Model\_B estimator'));

%print\_est(indata=%str(Model\_S\_est),mytitle=%str('Model\_S estimator'));

\* run model in-sample and out-of-sample validation and print results ;

%data\_validation(indata=%str(Model\_AdjR2\_out),outdata=%str(Model\_AdjR2\_validation));

%print\_mse\_mae(indata=%str(Model\_AdjR2\_validation),mytitle=%str('Model\_AdjR2 MSE MAE'));

%data\_validation(indata=%str(Model\_MaxR\_out),outdata=%str(Model\_MaxR\_validation));

%print\_mse\_mae(indata=%str(Model\_MaxR\_validation),mytitle=%str('Model\_MaxR MSE MAE'));

%data\_validation(indata=%str(Model\_MCp\_out),outdata=%str(Model\_MCp\_validation));

%print\_mse\_mae(indata=%str(Model\_MCp\_validation),mytitle=%str('Model\_MCp MSE MAE'));

%data\_validation(indata=%str(Model\_F\_out),outdata=%str(Model\_F\_validation));

%print\_mse\_mae(indata=%str(Model\_F\_validation),mytitle=%str('Model\_F MSE MAE'));

%data\_validation(indata=%str(Model\_B\_out),outdata=%str(Model\_B\_validation));

%print\_mse\_mae(indata=%str(Model\_B\_validation),mytitle=%str('Model\_B MSE MAE'));

%data\_validation(indata=%str(Model\_S\_out),outdata=%str(Model\_S\_validation));

%print\_mse\_mae(indata=%str(Model\_S\_validation),mytitle=%str('Model\_S MSE MAE'));

\* print model operational grades ;

%print\_grade(indata=%str(Model\_AdjR2\_validation),mytitle=%str('Model\_AdjR2 Prediction Grade'));

%print\_grade(indata=%str(Model\_MaxR\_validation),mytitle=%str('Model\_MaxR Prediction Grade'));

%print\_grade(indata=%str(Model\_MCp\_validation),mytitle=%str('Model\_MCp Prediction Grade'));

%print\_grade(indata=%str(Model\_F\_validation),mytitle=%str('Model\_F Prediction Grade'));

%print\_grade(indata=%str(Model\_B\_validation),mytitle=%str('Model\_B Prediction Grade'));

%print\_grade(indata=%str(Model\_S\_validation),mytitle=%str('Model\_S Prediction Grade'));

\* run a final operationa validation of the model produced by F/S/B;

proc reg data = train plots = diagnostics(unpack) outest = Model\_Final\_est;

Model\_Final: model train\_response =

GrLivArea LotArea AgeAtSale TotalBsmtSF total\_baths\_calc

TotRmsAbvGrd highend\_ind midend\_ind good\_heating excl\_kitchen

fireplace\_ind good\_basement\_ind concr\_foundation quality\_index

brick\_exterior new\_bldg;

output out=Model\_Final\_out (keep=train SalePrice YrSold yhat) predicted=yhat;

title 'Model\_Final';

run;

%data\_validation(indata=%str(Model\_Final\_out),outdata=%str(Model\_Final\_validation));

\*Multicollinearity;

proc reg data=train;

model train\_response =

GrLivArea LotArea AgeAtSale TotalBsmtSF total\_baths\_calc

TotRmsAbvGrd highend\_ind midend\_ind good\_heating excl\_kitchen

central\_air fireplace\_ind garage\_ind good\_basement\_ind concr\_foundation

quality\_index brick\_exterior lot\_frontage new\_bldg old\_bldg pos\_cond recent\_remodel / corrb vif collin;

ods graphics on;

title 'Correlation matrix ';

proc corr data=train plots=scatter(alpha=.20 .30);

var GrLivArea LotArea AgeAtSale TotalBsmtSF total\_baths\_calc

TotRmsAbvGrd highend\_ind midend\_ind good\_heating excl\_kitchen

central\_air fireplace\_ind garage\_ind good\_basement\_ind concr\_foundation

quality\_index brick\_exterior lot\_frontage new\_bldg old\_bldg pos\_cond recent\_remodel ;

run;

ods graphics off;

ods graphics on;

title 'Correlation matrix ';

proc corr data=train plots=scatter(alpha=.20 .30);

var AgeAtSale total\_baths\_calc

midend\_ind

concr\_foundation

old\_bldg ;

run;

ods graphics off;