Assignment #5

Andrew Knight

Introduction:

The goal of this assignment is to bring together all of the concepts and analysis from the previous three assignments to choose the best model. The final model should be chosen and validated according to the methods practiced to this point in the course. The sections that follow contain the analysis of different models based on the now familiar Ames Housing Data.

Data File: ames_housing_data.csv

Calculations File: Knight_Assignment5_code.R

Assignment Tasks

1: Define the Sample Population

For my sample population I have started with the full list of observations listed in the datafile and performed the following modifications. I've applied drop conditions to restrict the complete list of 2900 observations to only those records that are zoned as Residential and only those records with GrLivArea less than 4500 square feet. Accordingly, records where Zoning is C (all) or I (all) were dropped and records where the above ground square footage was above 4500 were dropped. This removed 27 records not listed as Residential and dropped 3 records with excessive square footage. I also took into account the fact that the three records with very high square footage values also showed SalePrices much lower than expected indicating some other factor was at play. This left a reduced sample of exactly 2900 observations, which represents approximately 99% of the original dataset. I saved this sample subset as 'sampledat' and will be using this for my model analysis below.

2: The Predictive Modeling Framework

<u>New Variables Created</u>

I created a few new variables in the sample dataset derived from existing variables. These included the following: **TotalSqFtCalc** which gives sum of GrLivArea + BsmtFinSF1 + BsmtFinSF2 intended to represent the total finished living area in square feet, **HouseAge** which gives the age of the house from the YrSold minus the YrBuilt in years, **QualityIndex** which gives a measure of quality of the property from combining the ordinal scores of OverallQual and OverallCond, and **TotalPorchSF** which gives the total area of all screenPorch + OpenPorchSF + WoodDeckSF in square feet.

In addition to the numeric variables above I created discrete dummy variables for the categorical variables **BldgType**, **Neighborhood**, **Condition2** , **KitchenQual** and **CentralAir** to provide additional predictive power. The selection of these categorical variables was based on my current and past EDA analysis of categorical variables available in the dataset.

Creation of Train & Test Data

I also created a 70/30 train/test set. The total observation count is 2900. The split for the observations contained in each is given below. This was done after all new variables were created and analyzed. I also experimented with a 80/20 train/test split to see how prediction grades would change. Interestingly, the split affected everything from my model predictor selection, to VIF to the final prediction grades achieved. I've included the highest VIF, highest R2 value and highest Prediction Grade achieved as a comparison in the table below. It shows that for the same predictive model, using a different training set size can give a different predictive outcome. Of course, the actual outcomes are based on the specific observations included in training/test set are determined by random assignment so these number would be expected to fluctuate if a different seed was used each time.

Ultimately, I decided it was better to be conservative on the training set used to build the model to avoid overfitting and risk not having enough test data to validate. If I had more records to deal with in the original data set, I may have used a different split. The highest prediction grade (PG) value for the greater than 25% error on the test set was also noted for each split choice. The goal was to observe how splitting the dataset differently would influence the results for the model and for the final prediction

	Train Obs #	Test Obs #	Highest VIF	Highest R^2	Highest Test PG @ .25+
70/30 Split Used	2032	868	6.2	.92	0.0426
Also Tested 80/20 split	2327	573	13.5	.93	0.0366

3: Model Identification by Automated Variable Selection

Using the variables I created along with the existing good predictors, I created a keepcol.list that will allow me to focus on the relevant variables for my automated variable selection process. I started with a list of 20 predictors and SalePrice as the response. Significance calculations for each given in the screenshot below. Next, I defined the upper (full) model including each of the 20 predictors identified above.

```
Call:
lm(formula = SalePrice ~ ., data = train.clean)

Residuals:
    Min      1Q  Median      3Q     Max
-116454  -11690    -431   10969  210247

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    5.262e+04  2.314e+04   2.274  0.02309 *
LotArea        3.031e-01  7.273e-02   4.168 3.20e-05 ***
TotalBsmtSF    1.872e+01  1.693e+00  11.059  < 2e-16 ***
GrLivArea      2.621e+01  2.298e+00  11.409  < 2e-16 ***
BedroomAbvGr  -3.957e+03  8.347e+02  -4.740 2.29e-06 ***
GarageArea     1.773e+01  3.166e+00   5.600 2.44e-08 ***
HouseAge      -3.546e+02  2.526e+01 -14.040  < 2e-16 ***
QualityIndex   8.105e+02  7.292e+01  11.115  < 2e-16 ***
TotalSqFtCalc  5.237e+01  1.741e+00  30.074  < 2e-16 ***
TotalPorchSF   5.567e+00  3.714e+00   1.499  0.13407
BldgTypeGrp1   4.996e+03  1.891e+03   2.643  0.00829 **
BldgTypeGrp2  -9.228e+02  2.676e+03  -0.345  0.73024
NbhdGrp1      -6.140e+04  1.985e+03 -30.930  < 2e-16 ***
NbhdGrp2      -3.292e+04  1.578e+03 -20.870  < 2e-16 ***
CondPos        6.414e+04  1.337e+04   4.797 1.73e-06 ***
CondNeg       -1.083e+04  6.174e+03  -1.754  0.07957 .
CentralAir.Y  -6.635e+03  2.363e+03  -2.808  0.00503 **
kq_ex          2.260e+04  2.315e+04   0.976  0.32916
kq_gd         -1.751e+04  2.302e+04  -0.760  0.44707
kq_ta         -1.956e+04  2.298e+04  -0.852  0.39458
kq_fa         -1.256e+04  2.315e+04  -0.543  0.58746
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 22840 on 2009 degrees of freedom
Multiple R-squared:  0.9223,    Adjusted R-squared:  0.9216
F-statistic:  1193 on 20 and 2009 DF,  p-value: < 2.2e-16
```

I continued on to the automated variable selection process and defined a good model. Later, however I came back and restricted my predictor variable list further to see if I could improve on the AIC/BIC/AdjR^2 values further. The resulting models from the auto variable selection process improved; thus I decided to stick with the final list defined here:

Final Variables Selected (18 total including **SalePrice** as response):

 # col list test3 - final used, 18 total predictors

keepcol.list <- c('SalePrice', 'TotalSqFtCalc', 'GrLivArea', 'QualityIndex', 'HouseAge', 'BedroomAbvGr', 'TotalBsmtSF', 'GarageArea', 'CentralAir.Y', 'LotArea', 'BldgTypeGrp1', 'BldgTypeGrp2', 'kq_ex', 'kq_gd', 'kq_ta', 'NbhdGrp1', 'NbhdGrp2', 'CondPos', 'CondNeg')

A couple of things to note about the above final predictor list, I have included dummy variables derived from BldgType, Neighborhood, Condition2, and KitchenQual among the other continuous predictors.

Test Model Definition

My approach was to start with the forward selection, backward and stepwise AIC process and verify resulting models with AIC, BIC and Adjusted R-squared values for each. Reviewing each step of the stepAIC automated results, I verified that AIC values obtained as each variable was added or removed made sense and that the resulting final model did indeed improve on the metrics over the 'unfinished' models obtained. I did this by essentially spot checking RSE, Adj R-squared and AIC values for a few models that were listed as non-optimal that were printed while the automated model selection function was running. By reviewing the AIC values for each predictor, we can be confident that the right optimal model was obtained without blindly accepting the result. A junk, or bogus model is also a good way to verify the selection process function is working as expected.

I arrived at the following good models based on the forward, backward and stepwise tests.

Good Model 1

good1 <- lm(SalePrice ~ TotalSqFtCalc + NbhdGrp3 + NbhdGrp1 + HouseAge + QualityIndex + TotalBsmtSF + GrLivArea + GarageArea + BedroomAbvGr + CondPos + CentralAir.Y + BldgTypeGrp1 + LotArea + CondNeg, data=train.clean)

This model was the first and least best of my final model selections using a slightly different set of possible predictor variables defined by c('SalePrice', 'TotalSqFtCalc', 'GrLivArea', 'QualityIndex', 'HouseAge', 'BedroomAbvGr', 'TotalBsmtSF', 'GarageArea', 'CentralAir.Y', 'TotalPorchSF', 'LotArea', 'BldgTypeGrp1', 'BldgTypeGrp2', 'NbhdGrp1', 'NbhdGrp2','NbhdGrp3', 'CondPos', 'CondNeg').

As a result, I actually went back and finalized the list of the predictor variables to the list shown at the top of this section. The final predictor variable list was used to narrow in on the final models shown in the table.

Good Model 2 - from forward (same as stepwise)

good2 <- lm(SalePrice ~ TotalSqFtCalc + NbhdGrp1 + NbhdGrp2 + kq_ex + HouseAge + QualityIndex + TotalBsmtSF + GrLivArea + GarageArea + CondPos + LotArea + BedroomAbvGr + BldgTypeGrp1 + CentralAir.Y + kq_ta + CondNeg, data=train.clean)

This model provided a better AIC, BIC and Adjusted R-squared model over Model 1. Based on the backward and stepwise test, I arrived at the variables listed for model 2. After reviewing AIC value for

similar models the best AIC, BIC and Adjusted R-squared values obtained using Model 2 are listed in the table below.

Good Model 3 - from backward

good3 <- lm(SalePrice ~ LotArea + TotalBsmtSF + GrLivArea + BedroomAbvGr + GarageArea + HouseAge + QualityIndex + TotalSqFtCalc + BldgTypeGrp1 + NbhdGrp1 + NbhdGrp2 + CondPos + CondNeg + CentralAir.Y + kq_ex + kq_ta, data=train.clean)

Turns out that while the predictors listed and the order are different, the backward model arrived at almost identical metrics as shown in table below.

Good Model 4 - simplified

good4 <- lm(SalePrice ~ TotalSqFtCalc + NbhdGrp1 + NbhdGrp2 + kq_ex + HouseAge + QualityIndex + TotalBsmtSF + GrLivArea + GarageArea + CondPos + LotArea + BedroomAbvGr + BldgTypeGrp1 + CentralAir.Y + kq_ta, data=train.clean)

I also tried this slightly model with just one predictor difference from Model #2 above where the last (and least significant predictor was omitted). This attempt of slightly modifying the forward optimized model minus the last predictor was named simplified (Good4).

Model	**AIC**	**BIC**	**Adj R-Squared**	**RSE**
Good1 - init	46807	46897	0.9099	24,480
Good2 - forwd	46527	46628	0.9216	22,840
Good3 - bckwd	46527	46628	0.9216	22,840
Good4 - simpl	46528	46623	0.9215	22,850

It turns out that the initial (Good1) model did not perform as well as expected after reviewing the AIC/BIC and residual standard error. This test of the initial model resulting in modifying my original set of 'keep predictors' by removing TotalPorchSF and changing dummy variables used.

Predictor Correlation Test using VIF

The next step before reviewing the predictive accuracy on the proposed final models is to check predictor variables to make sure they are not too highly correlated. The VIF test was used; I wanted to verify that the values obtained di not exceed 10 as a reasonable threshold for highly correlated predictors. In all proposed models the highest VIF value I obtained was 6.19 with most values falling between 1 and 5 for each predictor. This assuaged my concerns about multicollinearity.

4: Predictive Accuracy

The predictive accuracy given by the mean squared error (MSE) and the mean absolute error (MAE) from the predictors. These are used along with the metrics in the model comparison table above to assess the error in the difference between the overall mean and the predicted values for each variable.

Model	MSE	MAE
Good1 - init	594964355	16735
Good2 - forwd	517225876	15340
Good3 - bckwd	517225876	15340
Good4 - simpl	518028981	15378

Because the models Good2 (based on the forward best) and Good3 (based on the backward best) gave identical results for MSE and MAE, and nearly identical results for AIC/BIC/AdjR^2, I decided to take Good2 and the final model to validate.

5: Operational Validation

Because statistical metrics do not always translate to actionable business metrics, we need a grading system that we can use to determine the performance of the models relative to the business goals. In this case our goal is to come up with a model that can be used to predict future sale prices with some level confidence. Because we want to establish the confidence in the predictive ability, I'll use a grading system lists the performance of our models with between 0 and 10%, 10 to 15%, 15 to 25%, and more than 25% error respectively. To do this we compare the absolute values of the residuals for each divided by the SalePrice. We are interested in two aspects of this metric. We want to see 'most' of our observations fall within the lowest margin of error possible and we want to make sure our test set results do not deviate too much from our training set results. The table below give the complete comparison for each percentage accuracy grade obtained.

Model	Grade1 <10%	Grade2 – 10-15%	Grade3- 15-25%	Grade4 > 25%
Good1	0.631 / 0.621	0.153 / 0.166	0.147 / 0.133	0.069 / 0.080
Good2	0.670 / 0.664	0.150 / 0.152	0.127 / 0.115	0.053 / 0.068
Good3	0.670 / 0.664	0.150 / 0.152	0.127 / 0.115	0.053 / 0.068
Good4	0.6699 / 0.6655	0.149 / 0.149	0.127 / 0.116	0.053 / 0.068

Each listing above shows the training set value followed by the test set value for each prediction grade. These values are actually all quite close in values, in fact I had to include additional decimal places to note differences as shown in the last row. The point is, all models perform relatively well in that over 65% give a probability of correct values with less than 10% error. The other promising thing to note is that training and test results are close. The actual values between train and test will vary depending on the randomization of the split function. However, we should not see major deviations between train and test. Based on these results, my two best models Good2 and Good3 appear to perform strong accuracy in both train and test data. Also, these numbers appear to be good enough for 'underwriting quality' designation.

6: Best Model

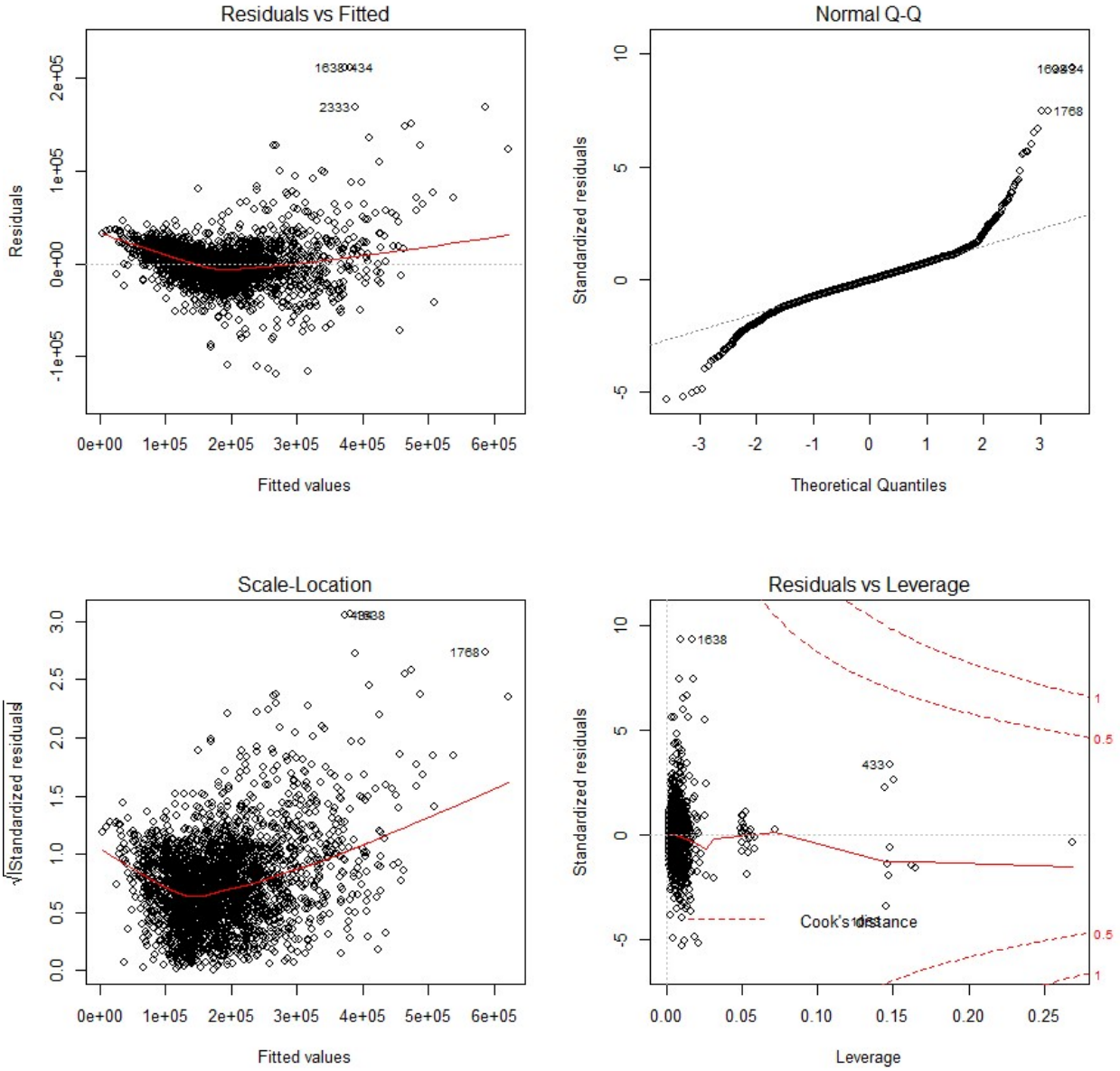
My final 'best' model is given by Good 2, which was obtained using the forward (and stepwise) selection process to arrive at the best AIC, BIC, and Adj R^2 values.

My Final Selected Model is:

lm(SalePrice ~ TotalSqFtCalc + NbhdGrp1 + NbhdGrp2 + kq_ex + HouseAge + QualityIndex + TotalBsmtSF + GrLivArea + GarageArea + CondPos + LotArea + BedroomAbvGr + BldgTypeGrp1 + CentralAir.Y + kq_ta + CondNeg, data=train.clean)

This model was selected on the basis of being the best fit for both training data and test data. I feel it strikes the right balance of number of predictors and ideal prediction accuracy.

I revisited the assumptions for the multiple linear model and conducted one final review. The plot of the result when running the model on the full sample set is shown below.



As before, we notice the heteroscedasticity and linearity issues in the plot which challenge the assumptions of the model we've formed. Again, a transformation would help with these issues as we saw in previous work, however we are restricting our final model to non-transformed interpretations only for this one.

Another thing to recheck is the dffits threshold to find any influential points that may be adversely affecting my model. Based on my calculated dffits threshold of 0.162, I did not observe any outliers which were far enough away from the standard residual to merit being removed. A quick check of dffits values confirmed this in my calculations.

Based on the results in the Q-Q Plot, additional outliers in one or more of my predictors may be candidates for removal especially on the right hand (positive) side of the quantile spectrum. However, given the number of predictors requiring review a possible transformation may be a better solution. I intentionally limited the number of drop conditions at the start to maintain as many observations as possible. This was important to me when running this analysis of predictors because the dataset needed to be split into train and test sets. As such, I was willing to let the assumptions lax a bit to focus on predictive robustness for future out-of-sample performance of the final model.

Reflections & Conclusion

In summary, I enjoyed working through the automated variable selection process to arrive at a best model. My best final model gives a decent prediction grade while utilizing a range of predictor variables that satisfy my intuitions about what drives sale prices for residential properties. It was interesting to see how changes to variables (and observations) included in the various models changed the significance and predictive power of other variables. Without a disciplined process and automated tools within R packages, this would be very time consuming. It also occurred to me that while we worked with a dataset with 80+ possible predictor variables, there may be other similar scenarios where predicting housing data could have many more possible predictors. The Ames data is only in one geographical area and excluded other possible items like school district rating, state/local tax laws, walkability scores, air/water quality scores, population density, health of macro economy and other factors.

The overall predictive ability of this final measure can likely be improved some but with a final estimated R-squared value of 92% and a Residual Standard Error of 22,840 without transformation, I'm happy with the results. Perhaps a really good model is never complete. As more observations become available we can continue to modify as needed. The important measure of any model we intend to use is if it meets the needs of the original business problem we started out to solve. As the business needs evolve, so too must the analysis of the model.