Lab3

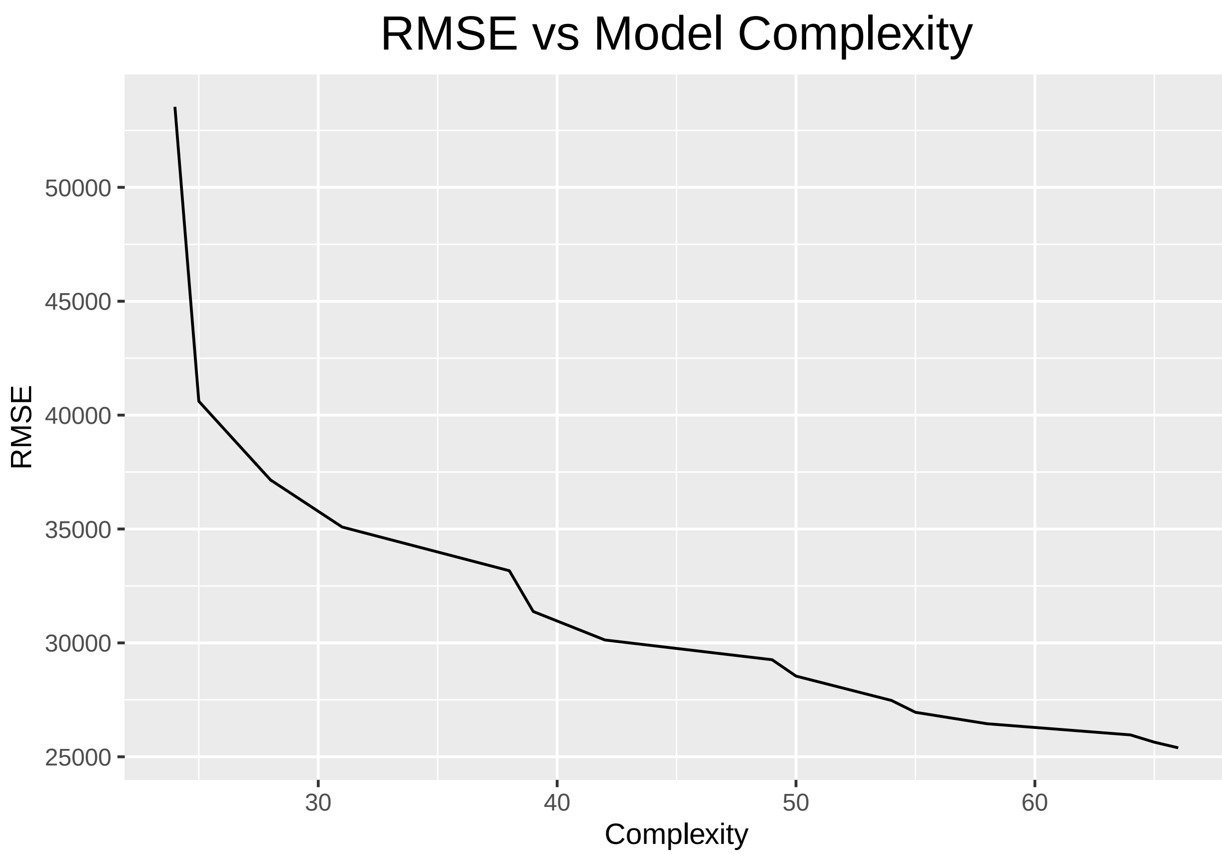
Group 17

2/11/2020

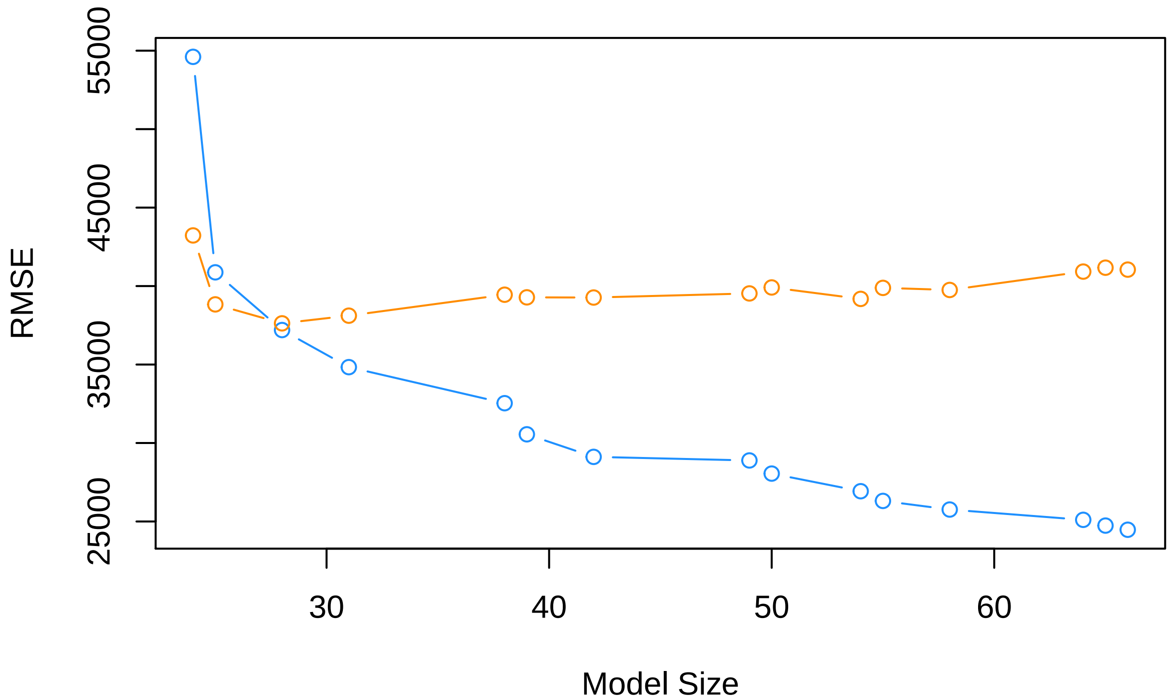
Exercise 1

The code below is 15 models we created based on a forward selection process. Criteria for adding the next parameter was which additional varaible results a lower RSS determined by the function step(). In addition we created a plot comparing the complexity of models to its Root Mean Squared Error. Complexity was measure by the number of predictors in a model. It is important to note that for categorical data, each dummy variable acts as a predictor. Threfore, a variable like Neighborhood with 25 levels would result in a complexity of 24.

0. nullModel <- lm(SalePrice ~ 1, data = ames_t)
1. lm(SalePrice ~ Neighborhood, data = ames_t)
2. lm(SalePrice ~ Neighborhood + GrLivArea, data = ames_t)
3. lm(SalePrice ~ Neighborhood + GrLivArea + KitchenQual, data = ames_t)
4. lm(SalePrice ~ Neighborhood + GrLivArea + KitchenQual + BsmtExposure, data = ames_t)
5. lm(SalePrice ~ Neighborhood + GrLivArea + KitchenQual + BsmtExposure + RoofMatl, data = ames_t)
6. lm(SalePrice ~ Neighborhood + GrLivArea + KitchenQual + BsmtExposure + RoofMatl + TotalBsmtSF, data = ames_t)
7. lm(SalePrice ~ BsmtQual + Neighborhood + GrLivArea + KitchenQual + BsmtExposure + RoofMatl + TotalBsmtSF, data = ames_t)
8. lm(SalePrice ~ Condition2 + BsmtQual + Neighborhood + GrLivArea + KitchenQual + BsmtExposure + RoofMatl + TotalBsmtSF, data = ames_t)
9. lm(SalePrice ~ BsmtFinSF1 + Condition2 + BsmtQual + Neighborhood + GrLivArea + KitchenQual + BsmtExposure + RoofMatl + TotalBsmtSF, data = ames_t)
10. lm(SalePrice ~ BldgType + BsmtFinSF1 + Condition2 + BsmtQual + Neighborhood + GrLivArea + KitchenQual + BsmtExposure + RoofMatl + TotalBsmtSF, data = ames_t)
11. lm(SalePrice ~ YearBuilt + BldgType + BsmtFinSF1 + Condition2 + BsmtQual + Neighborhood + GrLivArea + KitchenQual + BsmtExposure + RoofMatl + TotalBsmtSF, data = ames_t)
12. lm(SalePrice ~ ExterQual + YearBuilt + BldgType + BsmtFinSF1 + Condition2 + BsmtQual + Neighborhood + GrLivArea + KitchenQual + BsmtExposure + RoofMatl + TotalBsmtSF, data = ames_t)
13. lm(SalePrice ~ Functional + ExterQual + YearBuilt + BldgType + BsmtFinSF1 + Condition2 + BsmtQual + Neighborhood + GrLivArea + KitchenQual + BsmtExposure + RoofMatl + TotalBsmtSF, data = ames_t)
14. lm(SalePrice ~ LotArea + Functional + ExterQual + YearBuilt + BldgType + BsmtFinSF1 + Condition2 + BsmtQual + Neighborhood + GrLivArea + KitchenQual + BsmtExposure + RoofMatl + TotalBsmtSF, data = ames_t)
15. lm(SalePrice ~ YearRemodAdd + LotArea + Functional + ExterQual + YearBuilt + BldgType + BsmtFinSF1 + Condition2 + BsmtQual + Neighborhood + GrLivArea + KitchenQual + BsmtExposure + RoofMatl + TotalBsmtSF, data = ames_t)



As we can see from the graph above, as the model gets more complex, the Root Mean Squared Error will decrease. However, just because the the mean squared error decreases does not mean we should use the full model. The RMSE is the measure by taking sqrt(mean(actual - predicted)^2), therfore, when we add more predictors to our model, it is going to fit it better and of course redcuce the RMSE. However, when making models, we are not looking to fit our sample with the best model possible, but find the real relationship of something. The full model will often overfit the data causing greater RMSE in the actual population. ##Ex-



ercise 2

```
## [1] 37625.61
```

```
## [1] 33014.77
```

```
## [1] 31692.69
```

```
##
## Call:
## lm(formula = SalePrice ~ GrLivArea + GrLivArea * GrLivArea +
##     ExterQual + BsmtQual + GarageCars + BsmtFinSF1 + KitchenQual +
##     MSSubClass + BsmtExposure + YearBuilt + Fireplaces + Functional +
##     Condition1 + LotShape + LandContour + KitchenAbvGr + YearRemodAdd +
##     MasVnrArea + MSZoning + LotFrontage + BedroomAbvGr, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -379511  -15793     108   13227  243595
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -3.151e+05  2.178e+05  -1.447 0.148365
## GrLivArea         6.703e+01  4.633e+00  14.469  < 2e-16 ***
## ExterQualFa      -4.398e+04  1.801e+04  -2.441 0.014884 *
## ExterQualGd      -2.848e+04  9.694e+03  -2.938 0.003418 **
## ExterQualTA      -4.375e+04  1.060e+04  -4.129 4.10e-05 ***
```

```
## BsmtQualFa      -5.415e+04  1.112e+04  -4.870 1.39e-06 ***
## BsmtQualGd      -3.894e+04  6.140e+03  -6.341 4.15e-10 ***
## BsmtQualTA      -4.801e+04  7.408e+03  -6.481 1.74e-10 ***
## GarageCars       1.347e+04  2.417e+03   5.574 3.59e-08 ***
## BsmtFinSF1       7.104e+00  3.263e+00   2.177 0.029810 *
## KitchenQualFa   -3.082e+04  1.298e+04  -2.376 0.017798 *
## KitchenQualGd   -2.905e+04  6.981e+03  -4.162 3.57e-05 ***
## KitchenQualTA   -3.460e+04  7.968e+03  -4.342 1.63e-05 ***
## MSSubClass      -2.179e+02  3.733e+01  -5.838 8.17e-09 ***
## BsmtExposureGd   1.375e+04  5.643e+03   2.437 0.015073 *
## BsmtExposureMn  -1.232e+04  5.737e+03  -2.147 0.032117 *
## BsmtExposureNo  -1.458e+04  3.993e+03  -3.652 0.000280 ***
## YearBuilt        2.845e+01  8.056e+01   0.353 0.724079
## Fireplaces       9.418e+03  2.400e+03   3.924 9.59e-05 ***
## FunctionalMaj2  -5.490e+03  3.194e+04  -0.172 0.863581
## FunctionalMin1   2.797e+03  2.240e+04   0.125 0.900646
## FunctionalMin2  -6.742e+03  2.238e+04  -0.301 0.763374
## FunctionalMod    1.490e+04  2.367e+04   0.630 0.529112
## FunctionalSev   -7.754e+04  4.025e+04  -1.926 0.054490 .
## FunctionalTyp    2.385e+04  2.076e+04   1.149 0.251059
## Condition1Feedr -1.296e+03  8.819e+03  -0.147 0.883225
## Condition1Norm   1.517e+04  7.167e+03   2.117 0.034612 *
## Condition1PosA   4.714e+03  1.725e+04   0.273 0.784757
## Condition1PosN   3.882e+03  1.267e+04   0.306 0.759347
## Condition1RRAe  -4.326e+03  1.432e+04  -0.302 0.762624
## Condition1RRAn   2.878e+04  1.489e+04   1.932 0.053731 .
## Condition1RRNe  -5.684e+03  3.517e+04  -0.162 0.871662
## Condition1RRNn  -1.546e+04  3.614e+04  -0.428 0.669046
## LotShapeIR2      1.224e+03  7.857e+03   0.156 0.876297
## LotShapeIR3     -6.715e+04  1.592e+04  -4.218 2.80e-05 ***
## LotShapeReg     -4.368e+03  3.024e+03  -1.444 0.149088
## LandContourHLS   2.625e+04  9.557e+03   2.746 0.006182 **
## LandContourLow   1.366e+04  1.099e+04   1.243 0.214347
## LandContourLvl   9.453e+03  7.250e+03   1.304 0.192712
## KitchenAbvGr    -2.268e+04  6.659e+03  -3.406 0.000698 ***
## YearRemodAdd     1.975e+02  9.093e+01   2.172 0.030229 *
## MasVnrArea       1.764e+01  8.852e+00   1.993 0.046623 *
## MSZoningFV       5.315e+04  1.751e+04   3.036 0.002488 **
## MSZoningRH       2.991e+04  1.982e+04   1.509 0.131791
## MSZoningRL       4.295e+04  1.624e+04   2.646 0.008345 **
## MSZoningRM       2.981e+04  1.633e+04   1.825 0.068464 .
## LotFrontage     -1.868e+02  6.894e+01  -2.710 0.006891 **
## BedroomAbvGr    -1.470e+03  2.246e+03  -0.654 0.513202
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34160 on 682 degrees of freedom
## Multiple R-squared:  0.8334, Adjusted R-squared:  0.8219
## F-statistic: 72.57 on 47 and 682 DF,  p-value: < 2.2e-16
```

```
## Distribution not specified, assuming gaussian ...
```

```
##                       var     rel.inf
## GrLivArea       GrLivArea 26.31479454
```

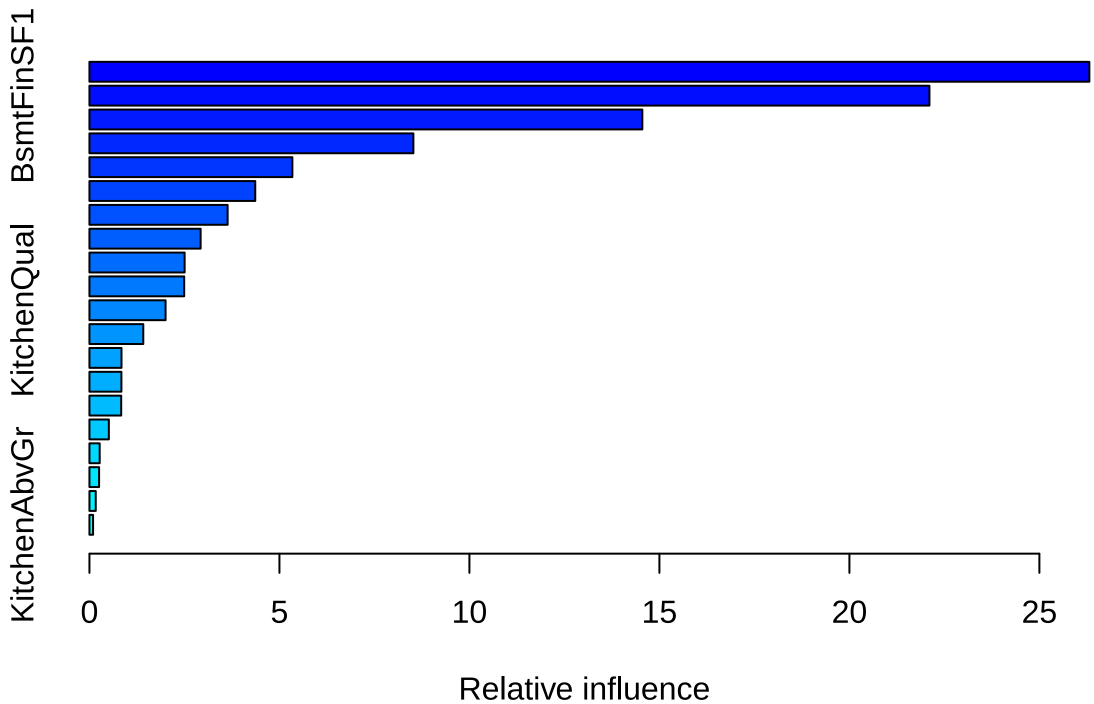
```
## BsmtFinSF1     BsmtFinSF1 22.10627883
## LotFrontage   LotFrontage 14.55072438
## ExterQual       ExterQual  8.52501167
## MasVnrArea     MasVnrArea  5.34079447
## YearBuilt       YearBuilt  4.36390668
## LandContour   LandContour  3.63688926
## GarageCars     GarageCars  2.92586994
## YearRemodAdd YearRemodAdd  2.50488467
## BsmtQual         BsmtQual  2.49346155
## KitchenQual   KitchenQual  2.00346916
## LotShape         LotShape  1.41702761
## Fireplaces     Fireplaces  0.84374332
## Condition1     Condition1  0.84129084
## BsmtExposure BsmtExposure  0.83569041
## MSZoning         MSZoning  0.51112766
## MSSubClass     MSSubClass  0.26941679
## BedroomAbvGr BedroomAbvGr  0.25422648
## Functional     Functional  0.16584926
## KitchenAbvGr KitchenAbvGr  0.09554246
```

```
## Using 216 trees...
```

```
## [1] 29923.25
```

```
## Using 216 trees...
```

```
## [1] 30539.99
```




```
##                       var     rel.inf
## GrLivArea       GrLivArea 26.31479454
## BsmtFinSF1     BsmtFinSF1 22.10627883
## LotFrontage   LotFrontage 14.55072438
## ExterQual       ExterQual  8.52501167
## MasVnrArea     MasVnrArea  5.34079447
## YearBuilt       YearBuilt  4.36390668
## LandContour   LandContour  3.63688926
## GarageCars     GarageCars  2.92586994
## YearRemodAdd YearRemodAdd  2.50488467
## BsmtQual         BsmtQual  2.49346155
## KitchenQual   KitchenQual  2.00346916
## LotShape         LotShape  1.41702761
## Fireplaces     Fireplaces  0.84374332
## Condition1     Condition1  0.84129084
## BsmtExposure BsmtExposure  0.83569041
## MSZoning         MSZoning  0.51112766
## MSSubClass     MSSubClass  0.26941679
## BedroomAbvGr BedroomAbvGr  0.25422648
## Functional     Functional  0.16584926
## KitchenAbvGr KitchenAbvGr  0.09554246
```

The best mean squared error we were able to achieve was 30539.99. To get this we used the best features from the data as found by the feed forward model. We removed roof type and neighborhood as they seemed to cause overfitting. Using GrLivArea squared increased the performance of the model probably due to an obeservable dimishing return to square footage in homes. Using a gradiant boosted tree regression through the gbm package also allowed us to perform slightly better than the linear regression. The test and train rmse are close which we took to be a good sign that we hadn't over or underfitted the data too much. We think our model overall will perform somewhat in the middle of the pack in the class.