Lab3

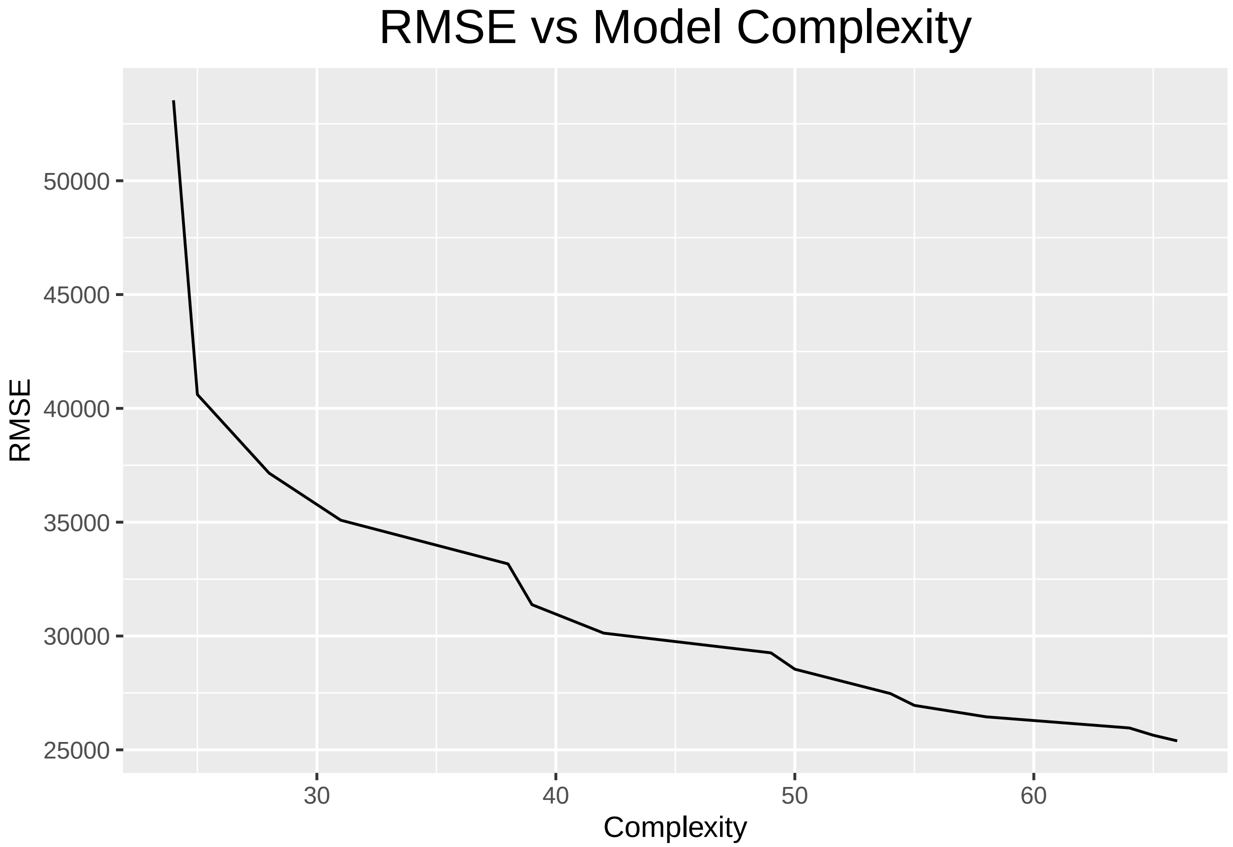
Group 17

2/11/2020

Exersize 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.

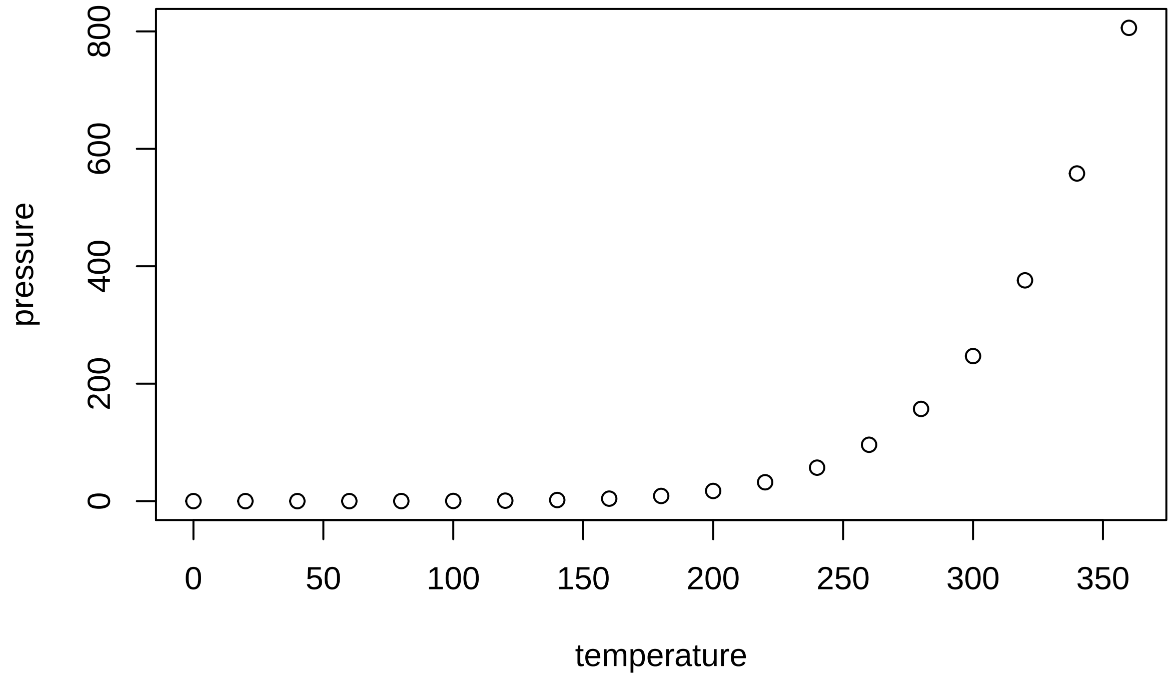
0. nullModel <- lm(SalePrice ~ 1, data = ames_t)
1. lm(SalePrice ~ Neighborhood, data = ames_t)
2. lm(SalePrice ~ Neighborhood + GrLivArea, data = ames_t)
3. lm(SalePrice ~ Neighborhood + GrLivArea + KitchenQual, data = ames_t)
4. lm(SalePrice ~ Neighborhood + GrLivArea + KitchenQual + BsmtExposure, 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)ˆ2), 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.

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.