Using Decision Trees and Random Forest to Predict MPG for 2019 Vehicles

I am going to use regression, decision trees, and the random forest algorithm to predict combined miles per gallon for all 2019 motor vehicles.  The raw data is located on the [EPA government site](https://www.fueleconomy.gov/feg/download.shtml)  
  
After preliminary diagnostics, exploration and cleaning I am going to start with a multiple linear regression model.  
  
The variables/features I am using for the models are: Engine displacement (size), number of cylinders, transmission type, number of gears, air inspired method, regenerative braking type, battery capacity Ah, drivetrain, fuel type, cylinder deactivate, and variable valve.   
  
There are 1253 vehicles in the dataset (does not include pure electric vehicles) summarized below.

fuel\_economy\_combined eng\_disp num\_cyl transmission

Min. :11.00 Min. :1.000 Min. : 3.000 A :301

1st Qu.:19.00 1st Qu.:2.000 1st Qu.: 4.000 AM : 46

Median :23.00 Median :3.000 Median : 6.000 AMS: 87

Mean :23.32 Mean :3.063 Mean : 5.533 CVT: 50

3rd Qu.:26.00 3rd Qu.:3.600 3rd Qu.: 6.000 M :148

Max. :58.00 Max. :8.000 Max. :16.000 SA :555

SCV: 66

num\_gears air\_aspired\_method

Min. : 1.000 Naturally Aspirated :523

1st Qu.: 6.000 Other : 5

Median : 7.000 Supercharged : 55

Mean : 7.111 Turbocharged :663

3rd Qu.: 8.000 Turbocharged+Supercharged: 7

Max. :10.000

regen\_brake batt\_capacity\_ah

No :1194 Min. : 0.0000

Electrical Regen Brake: 57 1st Qu.: 0.0000

Hydraulic Regen Brake : 2 Median : 0.0000

Mean : 0.3618

3rd Qu.: 0.0000

Max. :20.0000

drive cyl\_deactivate

2-Wheel Drive, Front :345 Y: 172

2-Wheel Drive, Rear :345 N:1081

4-Wheel Drive :174

All Wheel Drive :349

Part-time 4-Wheel Drive: 40

fuel\_type

Diesel, ultra low sulfur (15 ppm, maximum): 28

Gasoline (Mid Grade Unleaded Recommended) : 16

Gasoline (Premium Unleaded Recommended) :298

Gasoline (Premium Unleaded Required) :320

Gasoline (Regular Unleaded Recommended) :591

variable\_valve

N: 38

Y:1215

Call:

lm(formula = fuel\_economy\_combined ~ eng\_disp + transmission +

num\_gears + air\_aspired\_method + regen\_brake + batt\_capacity\_ah +

drive + fuel\_type + cyl\_deactivate + variable\_valve, data = cars\_19)

Residuals:

Min 1Q Median 3Q Max

-12.7880 -1.6012 0.1102 1.6116 17.3181

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 36.05642 0.82585 43.660 < 2e-16 \*\*\*

eng\_disp -2.79257 0.08579 -32.550 < 2e-16 \*\*\*

transmissionAM 2.74053 0.44727 6.127 1.20e-09 \*\*\*

transmissionAMS 0.73943 0.34554 2.140 0.032560 \*

transmissionCVT 6.83932 0.62652 10.916 < 2e-16 \*\*\*

transmissionM 1.08359 0.31706 3.418 0.000652 \*\*\*

transmissionSA 0.63231 0.22435 2.818 0.004903 \*\*

transmissionSCV 2.73768 0.40176 6.814 1.48e-11 \*\*\*

num\_gears 0.21496 0.07389 2.909 0.003691 \*\*

air\_aspired\_methodOther -2.70781 1.99491 -1.357 0.174916

air\_aspired\_methodSupercharged -1.62171 0.42210 -3.842 0.000128 \*\*\*

air\_aspired\_methodTurbocharged -1.79047 0.22084 -8.107 1.24e-15 \*\*\*

air\_aspired\_methodTurbocharged+Supercharged -1.68028 1.04031 -1.615 0.106532

regen\_brakeElectrical Regen Brake 12.59523 0.90030 13.990 < 2e-16 \*\*\*

regen\_brakeHydraulic Regen Brake 6.69040 1.94379 3.442 0.000597 \*\*\*

batt\_capacity\_ah -0.47689 0.11838 -4.028 5.96e-05 \*\*\*

drive2-Wheel Drive, Rear -2.54806 0.24756 -10.293 < 2e-16 \*\*\*

drive4-Wheel Drive -3.14862 0.29649 -10.620 < 2e-16 \*\*\*

driveAll Wheel Drive -3.12875 0.22300 -14.030 < 2e-16 \*\*\*

drivePart-time 4-Wheel Drive -3.94765 0.46909 -8.415 < 2e-16 \*\*\*

fuel\_typeGasoline (Mid Grade Unleaded Recommended) -5.54594 0.97450 -5.691 1.58e-08 \*\*\*

fuel\_typeGasoline (Premium Unleaded Recommended) -5.44412 0.70009 -7.776 1.57e-14 \*\*\*

fuel\_typeGasoline (Premium Unleaded Required) -6.01955 0.70542 -8.533 < 2e-16 \*\*\*

fuel\_typeGasoline (Regular Unleaded Recommended) -6.43743 0.68767 -9.361 < 2e-16 \*\*\*

cyl\_deactivateY 0.52100 0.27109 1.922 0.054851 .

variable\_valveY 2.00533 0.59508 3.370 0.000775 \*\*\*

---

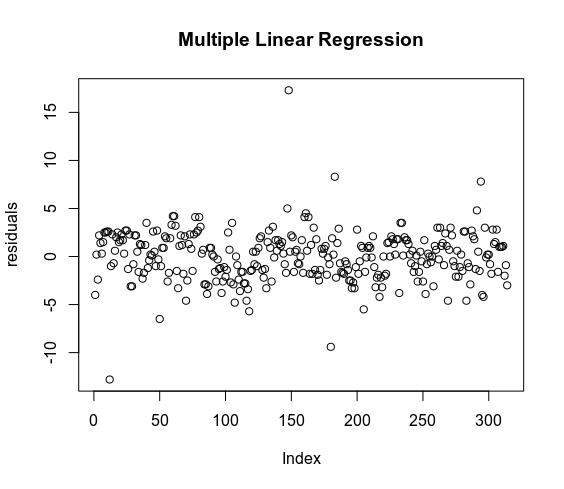
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

standard error: 2.608 on 1227 degrees of freedom

Multiple R-squared: 0.8104, Adjusted R-squared: 0.8066

F-statistic: 209.8 on 25 and 1227 DF, p-value: < 2.2e-16

The fitted MSE is 6.8 and predicted MSE of 6.83.  Some of the below residuals are too large.  The extreme large residual is a Hyundai Ioniq which none of the models predict very well as it is unique vehicle (versus the other data points).

[](https://s3-us-west-1.amazonaws.com/alpha-analysis.com/Pictures/MPG/mlr_residuals.png)

Let's try a decision tree regression model.

#regression tree full

m\_reg\_tree\_full <- rpart(formula = fuel\_economy\_combined ~ .,

data = train,

method = "anova",)

#regression tree tuned

m\_reg\_tree\_trimmed <- rpart(

formula = fuel\_economy\_combined ~ .,

data = train,

method = "anova",

control = list(minsplit = 10, cp = .0005)

)

#rpart.plot(m\_reg\_tree\_full)

plotcp(m\_reg\_tree\_full)

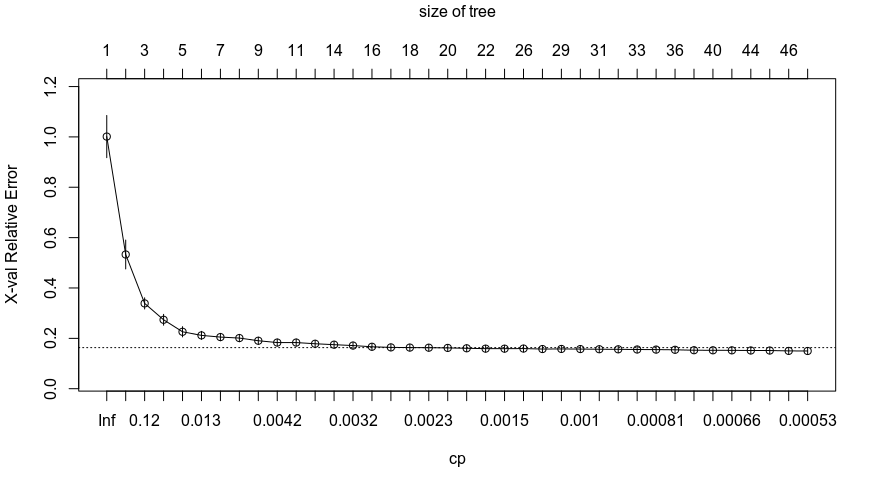
pred\_decision\_tree\_full <- predict(m\_reg\_tree\_full, newdata = test)

mse\_tree\_full <- RMSE(pred = pred\_decision\_tree\_full, obs = test$fuel\_economy\_combined) ^2

pred\_decision\_tree\_trimmed <- predict(m\_reg\_tree\_trimmed, newdata = test)

mse\_tree\_trimmed <- RMSE(pred = pred\_decision\_tree\_trimmed, obs = test$fuel\_economy\_combined) ^2

plotcp(m\_reg\_tree\_trimmed)

[](https://s3-us-west-1.amazonaws.com/alpha-analysis.com/Pictures/MPG/decision_tree.png)

After tuning the decision tree the predicted MSE is 6.20 which is better than the regression model.  
  
Finally let's try a random forest model.  The random forest should produce the best model as it will attempt to remove some of the correlation within the decision tree structure.

#random forest

m\_random\_forest\_full <-randomForest(formula = fuel\_economy\_combined ~ ., data = train)

predict\_random\_forest\_full <- predict(m\_random\_forest\_full, newdata = test)

mse\_random\_forest\_full <- RMSE(pred = predict\_random\_forest\_full, obs = test$fuel\_economy\_combined) ^ 2

which.min(m\_random\_forest\_full$mse)

#random forest tuned

m\_random\_forest <- randomForest(formula = fuel\_economy\_combined ~ ., data = train, ntree = 250)

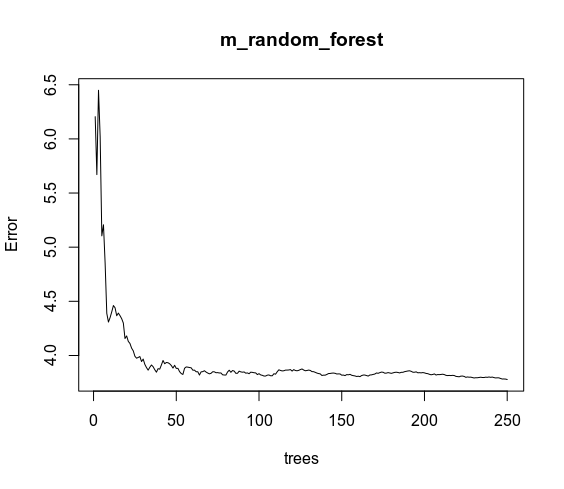
plot(m\_random\_forest)

predict\_random\_forest <- predict(m\_random\_forest, newdata = test)

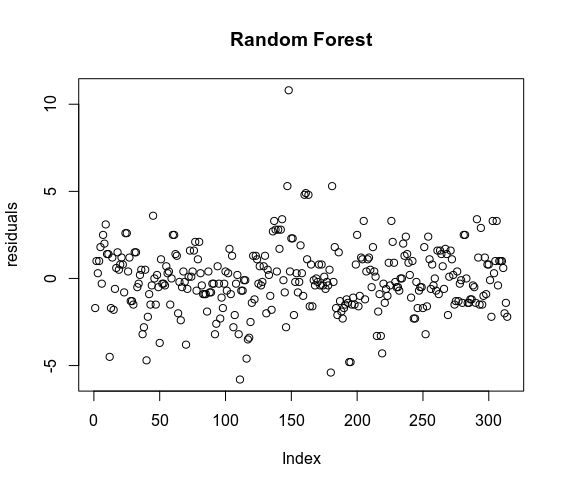
mse\_random\_forest <- RMSE(pred = predict\_random\_forest, obs = test$fuel\_economy\_combined) ^ 2

plot(tmp$test.fuel\_economy\_combined - tmp$r.predict\_random\_forrest., ylab = "residuals",main = "Random Forest")

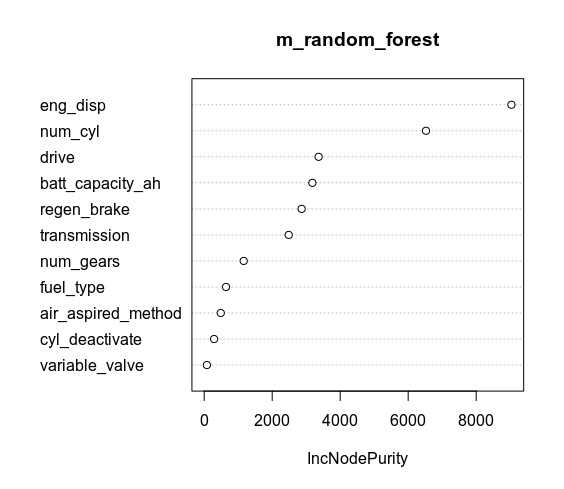
varImpPlot(m\_random\_forest)

[](https://s3-us-west-1.amazonaws.com/alpha-analysis.com/Pictures/MPG/random_forest_error.png)

The error stabilizes at 250 trees.  randomForest() by default uses 500 trees which is unnecessary.

[](https://s3-us-west-1.amazonaws.com/alpha-analysis.com/Pictures/MPG/random_forest_residuals.png)

After tuning the random forest the model has the lowest fitted and predicted MSE of 3.67 which is substantially better than the MSE of the decision tree 6.2  
  
The random forest also has an r-squared of .9

[](https://s3-us-west-1.amazonaws.com/alpha-analysis.com/Pictures/MPG/varimp.png)

Engine size, number of cylinders, and transmission type are the largest contributors to accuracy.

|  |  |
| --- | --- |
|  | #raw data |
|  | # https://www.fueleconomy.gov/feg/epadata/19data.zip |
|  |  |
|  | library(rpart) # regression trees |
|  | library(rpart.plot) # plotting regression trees |
|  | library(ipred) # bagging |
|  | library(caret) # bagging |
|  | library(e1071) |
|  | library(randomForest) #random forests |
|  |  |
|  | r <- function(data) {round(data, 1)} |
|  |  |
|  | set.seed(123) |
|  | indices <- sample(nrow(cars\_19), size = .75 \* nrow(cars\_19)) |
|  |  |
|  | train <- cars\_19[indices,] |
|  | test <- cars\_19[-indices,] |
|  | names <- cars[-indices, c(3, 4)] |
|  |  |
|  | #multiple linear regression model |
|  | lm1 <- step(lm(fuel\_economy\_combined ~ ., data = cars\_19)) |
|  | pred\_lm1 <- predict(lm1, newdata = test) |
|  | mse\_lm1 <- RMSE(pred = pred\_lm1, obs = test$fuel\_economy\_combined) ^ 2 |
|  |  |
|  | plot(pred\_lm1$residuals) |
|  |  |
|  | #regression tree full |
|  | m\_reg\_tree\_full <- rpart(formula = fuel\_economy\_combined ~ ., |
|  | data = train, |
|  | method = "anova",) |
|  |  |
|  | #regression tree tuned |
|  | m\_reg\_tree\_trimmed <- rpart( |
|  | formula = fuel\_economy\_combined ~ ., |
|  | data = train, |
|  | method = "anova", |
|  | control = list(minsplit = 10, cp = .0005) |
|  | ) |
|  |  |
|  | #rpart.plot(m\_reg\_tree\_full) |
|  | plotcp(m\_reg\_tree\_full) |
|  |  |
|  | pred\_decision\_tree\_full <- predict(m\_reg\_tree\_full, newdata = test) |
|  | mse\_tree\_full <- RMSE(pred = pred\_decision\_tree\_full, obs = test$fuel\_economy\_combined) ^2 |
|  |  |
|  | pred\_decision\_tree\_trimmed <- predict(m\_reg\_tree\_trimmed, newdata = test) |
|  | mse\_tree\_trimmed <- RMSE(pred = pred\_decision\_tree\_trimmed, obs = test$fuel\_economy\_combined) ^2 |
|  | plotcp(m\_reg\_tree\_trimmed) |
|  |  |
|  | #random forest |
|  | m\_random\_forest\_full <-randomForest(formula = fuel\_economy\_combined ~ ., data = train) |
|  | predict\_random\_forest\_full <- predict(m\_random\_forest\_full, newdata = test) |
|  | mse\_random\_forest\_full <- RMSE(pred = predict\_random\_forest\_full, obs = test$fuel\_economy\_combined) ^ 2 |
|  |  |
|  | which.min(m\_random\_forest\_full$mse) |
|  |  |
|  | #random forest tuned |
|  | m\_random\_forest <- randomForest(formula = fuel\_economy\_combined ~ ., data = train, ntree = 250) |
|  | plot(m\_random\_forest) |
|  | predict\_random\_forest <- predict(m\_random\_forest, newdata = test) |
|  | mse\_random\_forest <- RMSE(pred = predict\_random\_forest, obs = test$fuel\_economy\_combined) ^ 2 |
|  |  |
|  | plot(tmp$test.fuel\_economy\_combined - tmp$r.predict\_random\_forrest., ylab = "residuals",main = "Random Forest") |
|  |  |
|  | varImpPlot(m\_random\_forest) |
|  |  |
|  | tmp <-data.frame(names, test$fuel\_economy\_combined, r(pred\_lm1), r(pred\_decision\_tree\_trimmed),r(predict\_random\_forrest)) |
|  | tmp <- tmp[order(tmp$Division, tmp$Carline), ] |
|  |  |

Continuing on the below post, I am going to use a gradient boosted machine model to predict combined miles per gallon for all 2019 motor vehicles.

The raw data is located on the [EPA government site](https://www.fueleconomy.gov/feg/download.shtml)

The variables/features I am using for the models are: Engine displacement (size), number of cylinders, transmission type, number of gears, air inspired method, regenerative braking type, battery capacity Ah, drivetrain, fuel type, cylinder deactivate, and variable valve.

There are 1253 vehicles in the dataset (does not include pure electric vehicles) summarized below.

fuel\_economy\_combined eng\_disp num\_cyl transmission  
 Min. :11.00 Min. :1.000 Min. : 3.000 A :301   
 1st Qu.:19.00 1st Qu.:2.000 1st Qu.: 4.000 AM : 46   
 Median :23.00 Median :3.000 Median : 6.000 AMS: 87   
 Mean :23.32 Mean :3.063 Mean : 5.533 CVT: 50   
 3rd Qu.:26.00 3rd Qu.:3.600 3rd Qu.: 6.000 M :148   
 Max. :58.00 Max. :8.000 Max. :16.000 SA :555   
 SCV: 66   
 num\_gears air\_aspired\_method  
 Min. : 1.000 Naturally Aspirated :523   
 1st Qu.: 6.000 Other : 5   
 Median : 7.000 Supercharged : 55   
 Mean : 7.111 Turbocharged :663   
 3rd Qu.: 8.000 Turbocharged+Supercharged: 7   
 Max. :10.000   
   
 regen\_brake batt\_capacity\_ah   
 No :1194 Min. : 0.0000   
 Electrical Regen Brake: 57 1st Qu.: 0.0000   
 Hydraulic Regen Brake : 2 Median : 0.0000   
 Mean : 0.3618   
 3rd Qu.: 0.0000   
 Max. :20.0000   
   
 drive cyl\_deactivate  
 2-Wheel Drive, Front :345 Y: 172  
 2-Wheel Drive, Rear :345 N:1081  
 4-Wheel Drive :174   
 All Wheel Drive :349   
 Part-time 4-Wheel Drive: 40   
   
   
 fuel\_type   
 Diesel, ultra low sulfur (15 ppm, maximum): 28   
 Gasoline (Mid Grade Unleaded Recommended) : 16   
 Gasoline (Premium Unleaded Recommended) :298   
 Gasoline (Premium Unleaded Required) :320   
 Gasoline (Regular Unleaded Recommended) :591   
   
   
 variable\_valve  
 N: 38   
 Y:1215

Starting with an untuned base model:

trees <- 1200  
m\_boosted\_reg\_untuned <- gbm(  
 formula = fuel\_economy\_combined ~ .,  
 data = train,  
 n.trees = trees,  
 distribution = "gaussian"  
)

> summary(m\_boosted\_reg\_untuned)  
 var rel.inf  
eng\_disp eng\_disp 41.26273684  
batt\_capacity\_ah batt\_capacity\_ah 24.53458898  
transmission transmission 11.33253784  
drive drive 8.59300859  
regen\_brake regen\_brake 8.17877824  
air\_aspired\_method air\_aspired\_method 2.11397865  
num\_gears num\_gears 1.90999021  
fuel\_type fuel\_type 1.65692562  
num\_cyl num\_cyl 0.22260369  
variable\_valve variable\_valve 0.11043532  
cyl\_deactivate cyl\_deactivate 0.08441602  
> boosted\_stats\_untuned  
 RMSE Rsquared MAE   
2.4262643 0.8350367 1.7513331

The untuned GBM model performs better than the multiple linear regression model, but worse than the random forest.

I am going to tune the GBM by running a grid search:

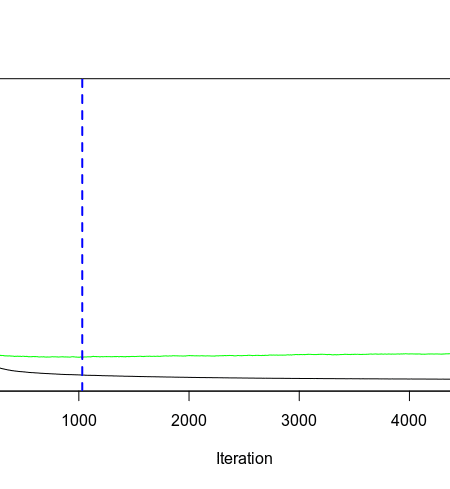
#create hyperparameter grid  
hyper\_grid <- expand.grid(  
 shrinkage = seq(.07, .12, .01),  
 interaction.depth = 1:7,  
 optimal\_trees = 0,  
 min\_RMSE = 0  
)  
  
#grid search  
for (i in 1:nrow(hyper\_grid)) {  
 set.seed(123)  
 gbm.tune <- gbm(  
 formula = fuel\_economy\_combined ~ .,  
 data = train\_random,  
 distribution = "gaussian",  
 n.trees = 5000,  
 interaction.depth = hyper\_grid$interaction.depth[i],  
 shrinkage = hyper\_grid$shrinkage[i],  
 )  
   
 hyper\_grid$optimal\_trees[i] <- which.min(gbm.tune$train.error)  
 hyper\_grid$min\_RMSE[i] <- sqrt(min(gbm.tune$train.error))  
   
 cat(i, "\n")  
}

The hyper grid is 42 rows which is all combinations of shrinkage and interaction depths specified above.

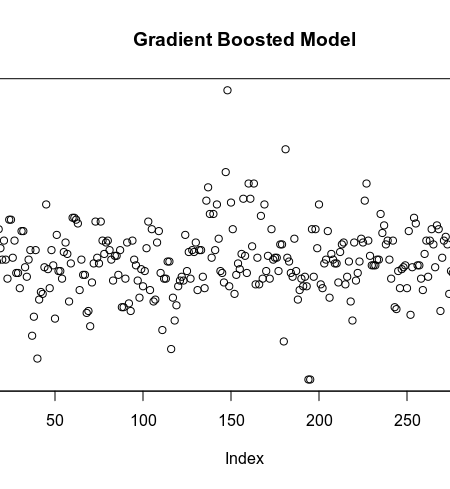
> head(hyper\_grid)  
 shrinkage interaction.depth optimal\_trees min\_RMSE  
1 0.07 1 0 0  
2 0.08 1 0 0  
3 0.09 1 0 0  
4 0.10 1 0 0  
5 0.11 1 0 0  
6 0.12 1 0 0

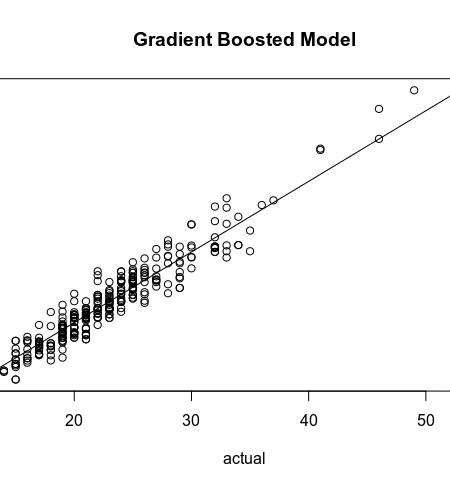
After running the grid search, it is apparent that there is overfitting. This is something to be very careful about.  I am going to run a 5 fold cross validation to estimate out of bag error vs MSE.  After running the 5 fold CV, this is the best model that does not overfit:

> m\_boosted\_reg <- gbm(  
 formula = fuel\_economy\_combined ~ .,  
 data = train,  
 n.trees = trees,  
 distribution = "gaussian",  
 shrinkage = .09,  
 cv.folds = 5,  
 interaction.depth = 5  
)  
  
best.iter <- gbm.perf(m\_boosted\_reg, method = "cv")  
pred\_boosted\_reg\_ <- predict(m\_boosted\_reg,n.trees=1183, newdata = test)  
mse\_boosted\_reg\_ <- RMSE(pred = pred\_boosted\_reg, obs = test$fuel\_economy\_combined) ^2  
boosted\_stats<-postResample(pred\_boosted\_reg,test$fuel\_economy\_combined)

[](https://i1.wp.com/s3-us-west-1.amazonaws.com/alpha-analysis.com/Pictures/GBM/mse_vs_oob_5000_boost.png?ssl=1)

The fitted black curve above is MSE and the fitted green curve is the out of bag estimated error.  1183 is the optimal amount of iterations.

[](https://i0.wp.com/s3-us-west-1.amazonaws.com/alpha-analysis.com/Pictures/GBM/residuals_boosted.png?ssl=1)

[](https://i1.wp.com/s3-us-west-1.amazonaws.com/alpha-analysis.com/Pictures/GBM/predict_vs_actual_boosted.png?ssl=1)

> pred\_boosted\_reg <- predict(m\_boosted\_reg,n.trees=1183, newdata = test)  
> mse\_boosted\_reg <- RMSE(pred = pred\_boosted\_reg, obs = test$fuel\_economy\_combined) ^2  
> boosted\_stats<-postResample(pred\_boosted\_reg,test$fuel\_economy\_combined)  
> boosted\_stats  
 RMSE Rsquared MAE   
1.8018793 0.9092727 1.3334459   
> mse\_boosted\_reg  
3.246769

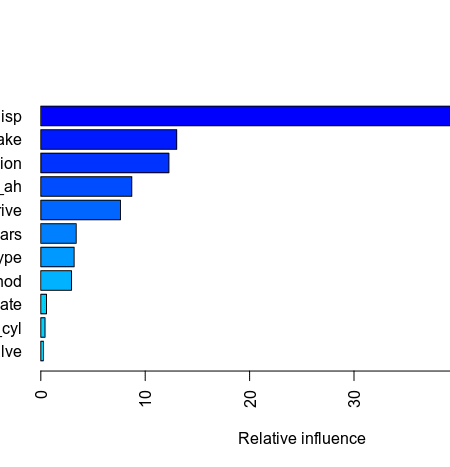
The tuned gradient boosted model performs better than the random forest with a MSE of 3.25 vs 3.67 for the random forest.

> summary(res)  
 Min. 1st Qu. Median Mean 3rd Qu. Max.   
-5.40000 -0.90000 0.00000 0.07643 1.10000 9.10000

50% of the predictions are within 1 MPG of the EPA Government Estimate.

The largest residuals are exotics and a hybrid which are the more unique data points in the dataset.

> tmp[which(abs(res) > boosted\_stats[1] \* 3), ]   
 Division Carline fuel\_economy\_combined pred\_boosted\_reg  
642 HYUNDAI MOTOR COMPANY Ioniq Blue 58 48.5  
482 KIA MOTORS CORPORATION Forte FE 35 28.7  
39 Lamborghini Aventador Coupe 11 17.2  
40 Lamborghini Aventador Roadster 11 17.2

[](https://i0.wp.com/s3-us-west-1.amazonaws.com/alpha-analysis.com/Pictures/GBM/varImp_boosted.png?ssl=1)