MATH4050-Final

Thomas Powell

2022-08-04

#Determine meaningful predictors  
data.lm = lm(price~., data = data)  
summary(data.lm)

##   
## Call:  
## lm(formula = price ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -21376.3 -592.4 -183.4 376.5 10694.3   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2183.404 408.175 5.349 8.87e-08 \*\*\*  
## carat 11257.127 48.626 231.504 < 2e-16 \*\*\*  
## cutGood 579.792 33.591 17.260 < 2e-16 \*\*\*  
## cutIdeal 832.975 33.406 24.935 < 2e-16 \*\*\*  
## cutPremium 762.161 32.227 23.650 < 2e-16 \*\*\*  
## cutVery Good 726.771 32.240 22.543 < 2e-16 \*\*\*  
## colorE -209.237 17.892 -11.694 < 2e-16 \*\*\*  
## colorF -272.877 18.092 -15.083 < 2e-16 \*\*\*  
## colorG -482.046 17.716 -27.210 < 2e-16 \*\*\*  
## colorH -980.271 18.835 -52.044 < 2e-16 \*\*\*  
## colorI -1466.251 21.162 -69.287 < 2e-16 \*\*\*  
## colorJ -2369.401 26.130 -90.676 < 2e-16 \*\*\*  
## clarityIF 5345.101 51.023 104.759 < 2e-16 \*\*\*  
## claritySI1 3665.451 43.633 84.006 < 2e-16 \*\*\*  
## claritySI2 2702.611 43.818 61.679 < 2e-16 \*\*\*  
## clarityVS1 4578.415 44.545 102.782 < 2e-16 \*\*\*  
## clarityVS2 4267.181 43.852 97.308 < 2e-16 \*\*\*  
## clarityVVS1 5007.771 47.159 106.190 < 2e-16 \*\*\*  
## clarityVVS2 4950.832 45.854 107.970 < 2e-16 \*\*\*  
## depth -63.790 4.534 -14.068 < 2e-16 \*\*\*  
## table -26.469 2.911 -9.092 < 2e-16 \*\*\*  
## x -1008.321 32.897 -30.651 < 2e-16 \*\*\*  
## y 9.606 19.332 0.497 0.619   
## z -50.123 33.486 -1.497 0.134   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1130 on 53919 degrees of freedom  
## Multiple R-squared: 0.9198, Adjusted R-squared: 0.9198   
## F-statistic: 2.688e+04 on 23 and 53919 DF, p-value: < 2.2e-16

confint(data.lm)

## 2.5 % 97.5 %  
## (Intercept) 1383.37748 2983.42956  
## carat 11161.81936 11352.43417  
## cutGood 513.95321 645.63173  
## cutIdeal 767.49867 898.45208  
## cutPremium 698.99618 825.32557  
## cutVery Good 663.58070 789.96100  
## colorE -244.30507 -174.16832  
## colorF -308.33746 -237.41708  
## colorG -516.76859 -447.32265  
## colorH -1017.18877 -943.35359  
## colorI -1507.72855 -1424.77365  
## colorJ -2420.61677 -2318.18555  
## clarityIF 5245.09570 5445.10637  
## claritySI1 3579.92985 3750.97170  
## claritySI2 2616.72883 2788.49415  
## clarityVS1 4491.10688 4665.72405  
## clarityVS2 4181.22978 4353.13206  
## clarityVVS1 4915.33936 5100.20206  
## clarityVVS2 4860.95821 5040.70562  
## depth -72.67694 -54.90212  
## table -32.17541 -20.76283  
## x -1072.79904 -943.84263  
## y -28.28589 47.49769  
## z -115.75533 15.50862

data = subset(data, select= -c(y,z))  
  
data.lm = lm(price~., data=data)  
summary(data.lm)

##   
## Call:  
## lm(formula = price ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -21385.4 -592.4 -183.8 376.4 10694.6   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2365.026 390.329 6.059 1.38e-09 \*\*\*  
## carat 11257.116 48.598 231.637 < 2e-16 \*\*\*  
## cutGood 580.281 33.572 17.285 < 2e-16 \*\*\*  
## cutIdeal 833.324 33.395 24.954 < 2e-16 \*\*\*  
## cutPremium 762.775 32.224 23.671 < 2e-16 \*\*\*  
## cutVery Good 726.808 32.211 22.564 < 2e-16 \*\*\*  
## colorE -209.356 17.892 -11.701 < 2e-16 \*\*\*  
## colorF -272.858 18.092 -15.082 < 2e-16 \*\*\*  
## colorG -481.950 17.716 -27.205 < 2e-16 \*\*\*  
## colorH -980.126 18.835 -52.037 < 2e-16 \*\*\*  
## colorI -1466.188 21.162 -69.284 < 2e-16 \*\*\*  
## colorJ -2369.507 26.130 -90.680 < 2e-16 \*\*\*  
## clarityIF 5344.337 51.014 104.763 < 2e-16 \*\*\*  
## claritySI1 3664.884 43.626 84.007 < 2e-16 \*\*\*  
## claritySI2 2702.102 43.811 61.676 < 2e-16 \*\*\*  
## clarityVS1 4577.607 44.534 102.788 < 2e-16 \*\*\*  
## clarityVS2 4266.569 43.846 97.309 < 2e-16 \*\*\*  
## clarityVVS1 5007.073 47.151 106.192 < 2e-16 \*\*\*  
## clarityVVS2 4950.186 45.846 107.975 < 2e-16 \*\*\*  
## depth -66.753 4.091 -16.319 < 2e-16 \*\*\*  
## table -26.452 2.911 -9.088 < 2e-16 \*\*\*  
## x -1029.543 20.549 -50.102 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1130 on 53921 degrees of freedom  
## Multiple R-squared: 0.9198, Adjusted R-squared: 0.9198   
## F-statistic: 2.944e+04 on 21 and 53921 DF, p-value: < 2.2e-16

confint(data.lm)

## 2.5 % 97.5 %  
## (Intercept) 1599.97767 3130.07351  
## carat 11161.86339 11352.36909  
## cutGood 514.48085 646.08185  
## cutIdeal 767.86988 898.77746  
## cutPremium 699.61634 825.93459  
## cutVery Good 663.67496 789.94125  
## colorE -244.42361 -174.28777  
## colorF -308.31789 -237.39742  
## colorG -516.67251 -447.22682  
## colorH -1017.04356 -943.20910  
## colorI -1507.66559 -1424.71075  
## colorJ -2420.72245 -2318.29128  
## clarityIF 5244.34966 5444.32369  
## claritySI1 3579.37635 3750.39128  
## claritySI2 2616.23219 2787.97202  
## clarityVS1 4490.31901 4664.89413  
## clarityVS2 4180.63129 4352.50659  
## clarityVVS1 4914.65645 5099.48879  
## clarityVVS2 4860.32767 5040.04365  
## depth -74.77043 -58.73552  
## table -32.15735 -20.74736  
## x -1069.81904 -989.26744

#Determine co-linearity  
vif(data.lm)

## GVIF Df GVIF^(1/(2\*Df))  
## carat 22.413427 1 4.734282  
## cut 1.928137 4 1.085531  
## color 1.178115 6 1.013753  
## clarity 1.346615 7 1.021484  
## depth 1.450563 1 1.204393  
## table 1.786854 1 1.336733  
## x 22.442074 1 4.737307

data = subset(data, select = -x)  
  
data.lm = lm(price~., data=data)  
summary(data.lm)

##   
## Call:  
## lm(formula = price ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -16828.8 -678.6 -199.4 464.6 10341.3   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -4557.249 373.456 -12.203 < 2e-16 \*\*\*  
## carat 8895.192 12.079 736.406 < 2e-16 \*\*\*  
## cutGood 614.481 34.337 17.896 < 2e-16 \*\*\*  
## cutIdeal 877.661 34.151 25.700 < 2e-16 \*\*\*  
## cutPremium 806.047 32.953 24.460 < 2e-16 \*\*\*  
## cutVery Good 778.426 32.935 23.635 < 2e-16 \*\*\*  
## colorE -211.002 18.303 -11.528 < 2e-16 \*\*\*  
## colorF -304.321 18.497 -16.453 < 2e-16 \*\*\*  
## colorG -506.975 18.116 -27.985 < 2e-16 \*\*\*  
## colorH -977.979 19.268 -50.755 < 2e-16 \*\*\*  
## colorI -1438.284 21.641 -66.461 < 2e-16 \*\*\*  
## colorJ -2322.566 26.714 -86.941 < 2e-16 \*\*\*  
## clarityIF 5404.240 52.173 103.584 < 2e-16 \*\*\*  
## claritySI1 3567.753 44.586 80.020 < 2e-16 \*\*\*  
## claritySI2 2619.029 44.787 58.478 < 2e-16 \*\*\*  
## clarityVS1 4525.418 45.546 99.359 < 2e-16 \*\*\*  
## clarityVS2 4210.140 44.839 93.894 < 2e-16 \*\*\*  
## clarityVVS1 5061.754 48.223 104.966 < 2e-16 \*\*\*  
## clarityVVS2 4957.333 46.900 105.700 < 2e-16 \*\*\*  
## depth -20.998 4.079 -5.148 2.64e-07 \*\*\*  
## table -24.794 2.977 -8.327 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1156 on 53922 degrees of freedom  
## Multiple R-squared: 0.9161, Adjusted R-squared: 0.916   
## F-statistic: 2.942e+04 on 20 and 53922 DF, p-value: < 2.2e-16

confint(data.lm)

## 2.5 % 97.5 %  
## (Intercept) -5289.22548 -3825.27196  
## carat 8871.51659 8918.86724  
## cutGood 547.18100 681.78136  
## cutIdeal 810.72481 944.59660  
## cutPremium 741.45826 870.63576  
## cutVery Good 713.87323 842.97792  
## colorE -246.87641 -175.12737  
## colorF -340.57498 -268.06686  
## colorG -542.48197 -471.46707  
## colorH -1015.74540 -940.21273  
## colorI -1480.70062 -1395.86698  
## colorJ -2374.92563 -2270.20563  
## clarityIF 5301.98119 5506.49899  
## claritySI1 3480.36516 3655.14109  
## claritySI2 2531.24714 2706.81160  
## clarityVS1 4436.14723 4614.68922  
## clarityVS2 4122.25464 4298.02554  
## clarityVVS1 4967.23700 5156.27035  
## clarityVVS2 4865.40893 5049.25800  
## depth -28.99304 -13.00332  
## table -30.63027 -18.95857

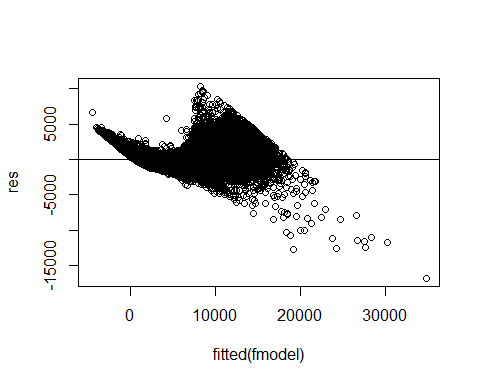
vif(data.lm)

## GVIF Df GVIF^(1/(2\*Df))  
## carat 1.323094 1 1.150258  
## cut 1.925927 4 1.085375  
## color 1.169324 6 1.013121  
## clarity 1.303869 7 1.019133  
## depth 1.378263 1 1.173995  
## table 1.786623 1 1.336646

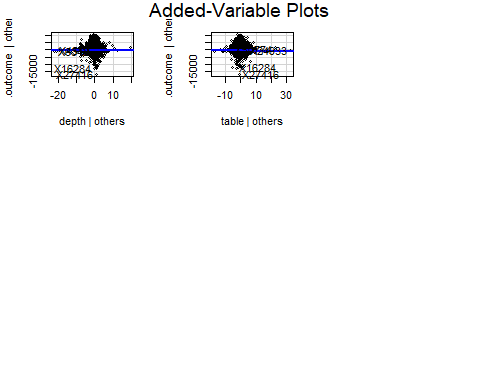
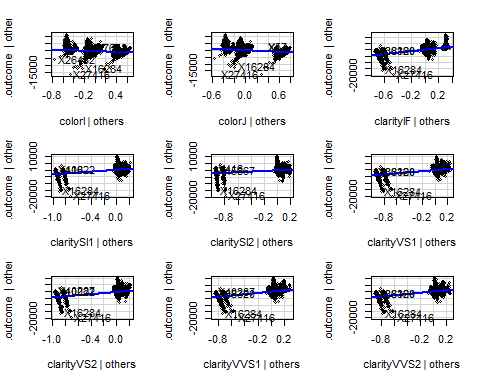
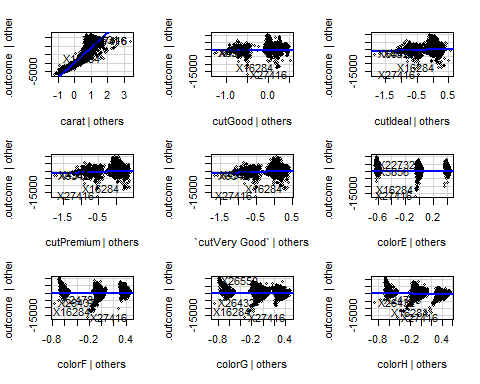
View(data)  
  
#Apply k-fold cross validation to model  
ctrl = trainControl(method = "cv", number = 5)  
data.lm = train(price ~., data = data, method = "lm", trControl = ctrl)  
  
print(data.lm)

## Linear Regression   
##   
## 53943 samples  
## 6 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 43154, 43154, 43154, 43155, 43155   
## Resampling results:  
##   
## RMSE Rsquared MAE   
## 1156.737 0.9160301 803.5233  
##   
## Tuning parameter 'intercept' was held constant at a value of TRUE

#Graphs  
fmodel = data.lm$finalModel  
res = resid(fmodel)  
plot(fitted(fmodel), res)  
abline(0,0)



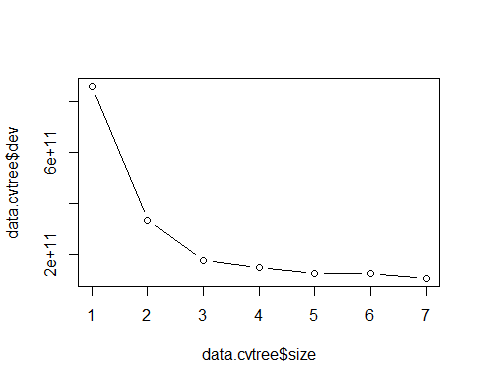
avPlots(fmodel)



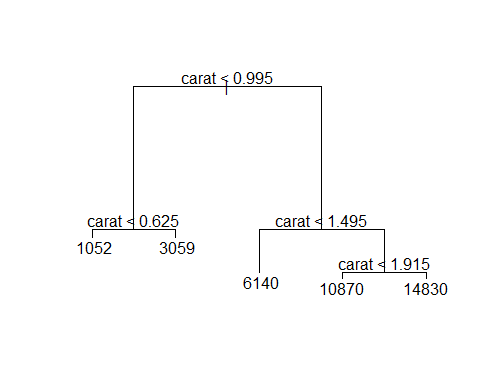
#Encode categorical variables  
dmy = dummyVars(" ~.", data = data, fullRank = T)  
data.dmy = data.frame(predict(dmy, newdata = data))  
  
head(data.dmy)

## carat cutGood cutIdeal cutPremium cutVery.Good colorE colorF colorG colorH  
## 1 0.23 0 1 0 0 1 0 0 0  
## 2 0.21 0 0 1 0 1 0 0 0  
## 3 0.23 1 0 0 0 1 0 0 0  
## 4 0.29 0 0 1 0 0 0 0 0  
## 5 0.31 1 0 0 0 0 0 0 0  
## 6 0.24 0 0 0 1 0 0 0 0  
## colorI colorJ clarityIF claritySI1 claritySI2 clarityVS1 clarityVS2  
## 1 0 0 0 0 1 0 0  
## 2 0 0 0 1 0 0 0  
## 3 0 0 0 0 0 1 0  
## 4 1 0 0 0 0 0 1  
## 5 0 1 0 0 1 0 0  
## 6 0 1 0 0 0 0 0  
## clarityVVS1 clarityVVS2 depth table price  
## 1 0 0 61.5 55 326  
## 2 0 0 59.8 61 326  
## 3 0 0 56.9 65 327  
## 4 0 0 62.4 58 334  
## 5 0 0 63.3 58 335  
## 6 0 1 62.8 57 336

#Implement decision tree  
data.tree = tree(price ~., data = data.dmy)  
data.cvtree = cv.tree(data.tree)  
plot(data.cvtree$size, data.cvtree$dev, type = 'b')



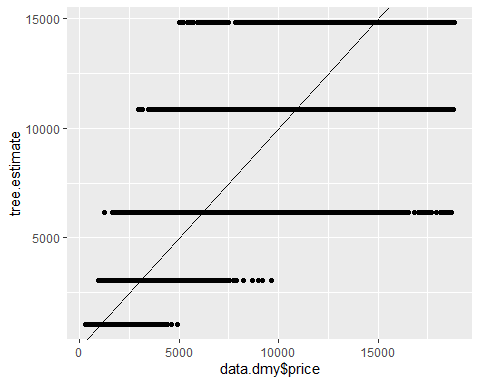
#Prune tree  
data.prune = prune.tree(data.tree, best = 5)  
plot(data.prune)  
text(data.prune, pretty = 0)



#Find RMSE  
tree.estimate = predict(data.prune, newdata = data.dmy)  
MSE = mean((tree.estimate - data$price)^2)  
RMSE = sqrt(MSE)  
RMSE

## [1] 1539.806

ggplot() +   
 geom\_point(aes(x = data.dmy$price, y = tree.estimate)) +  
 geom\_abline()



#Create a Random Forest  
rf.model = train(price~., data = data, method = "ranger", trControl = ctrl, importance = "impurity")  
rf.model

## Random Forest   
##   
## 53943 samples  
## 6 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 43155, 43154, 43154, 43155, 43154   
## Resampling results across tuning parameters:  
##   
## mtry splitrule RMSE Rsquared MAE   
## 2 variance 2308.3276 0.8602532 1607.1573  
## 2 extratrees 3114.6099 0.7139948 2307.8530  
## 11 variance 595.9875 0.9777345 296.5488  
## 11 extratrees 580.2361 0.9792208 297.7312  
## 20 variance 603.4472 0.9771453 298.0529  
## 20 extratrees 568.2139 0.9797341 289.3223  
##   
## Tuning parameter 'min.node.size' was held constant at a value of 5  
## RMSE was used to select the optimal model using the smallest value.  
## The final values used for the model were mtry = 20, splitrule = extratrees  
## and min.node.size = 5.

plot(rf.model)

