

Q41-WilliamKennedy-300015367

William Kennedy

2023-12-01

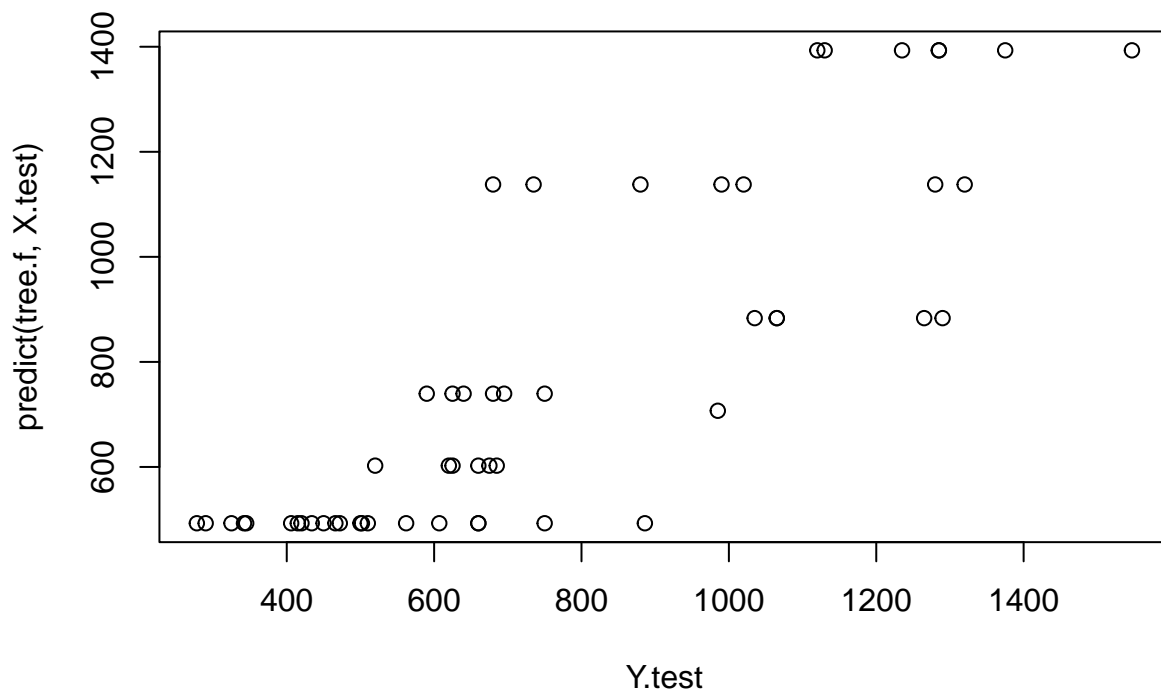
```
wine = read.csv("wine.csv")
str(wine)
```

```
## 'data.frame':  178 obs. of  14 variables:
## $ Class          : int  1 1 1 1 1 1 1 1 1 1 ...
## $ Alcohol         : num 14.2 13.2 13.2 14.4 13.2 ...
## $ Malic.acid      : num  1.71 1.78 2.36 1.95 2.59 1.76 1.87 2.15 1.64 1.35 ...
## $ Ash             : num  2.43 2.14 2.67 2.5 2.87 2.45 2.45 2.61 2.17 2.27 ...
## $ Alcalinity.of.ash : num 15.6 11.2 18.6 16.8 21 15.2 14.6 17.6 14 16 ...
## $ Magnesium       : int 127 100 101 113 118 112 96 121 97 98 ...
## $ Total.phenols    : num  2.8 2.65 2.8 3.85 2.8 3.27 2.5 2.6 2.8 2.98 ...
## $ Flavanoids       : num  3.06 2.76 3.24 3.49 2.69 3.39 2.52 2.51 2.98 3.15 ...
## $ Nonflavanoid.phenols: num  0.28 0.26 0.3 0.24 0.39 0.34 0.3 0.31 0.29 0.22 ...
## $ Proanthocyanins  : num  2.29 1.28 2.81 2.18 1.82 1.97 1.98 1.25 1.98 1.85 ...
## $ Colour.intensity : num  5.64 4.38 5.68 7.8 4.32 6.75 5.25 5.05 5.2 7.22 ...
## $ Hue             : num  1.04 1.05 1.03 0.86 1.04 1.05 1.02 1.06 1.08 1.01 ...
## $ OD280.OD315     : num  3.92 3.4 3.17 3.45 2.93 2.85 3.58 3.58 2.85 3.55 ...
## $ Proline         : int 1065 1050 1185 1480 735 1450 1290 1295 1045 1045 ...
```

```
set.seed(1234)
ind = sample(1:nrow(wine), size = 125)
wine.tr = wine[ind,]
wine.te = wine[-ind,]

lambda = 0.005
X = wine.tr[,1:13] # predictors
r = wine.tr[,14] # response
X.test = wine.te[,1:13]
Y.test = wine.te[,14]

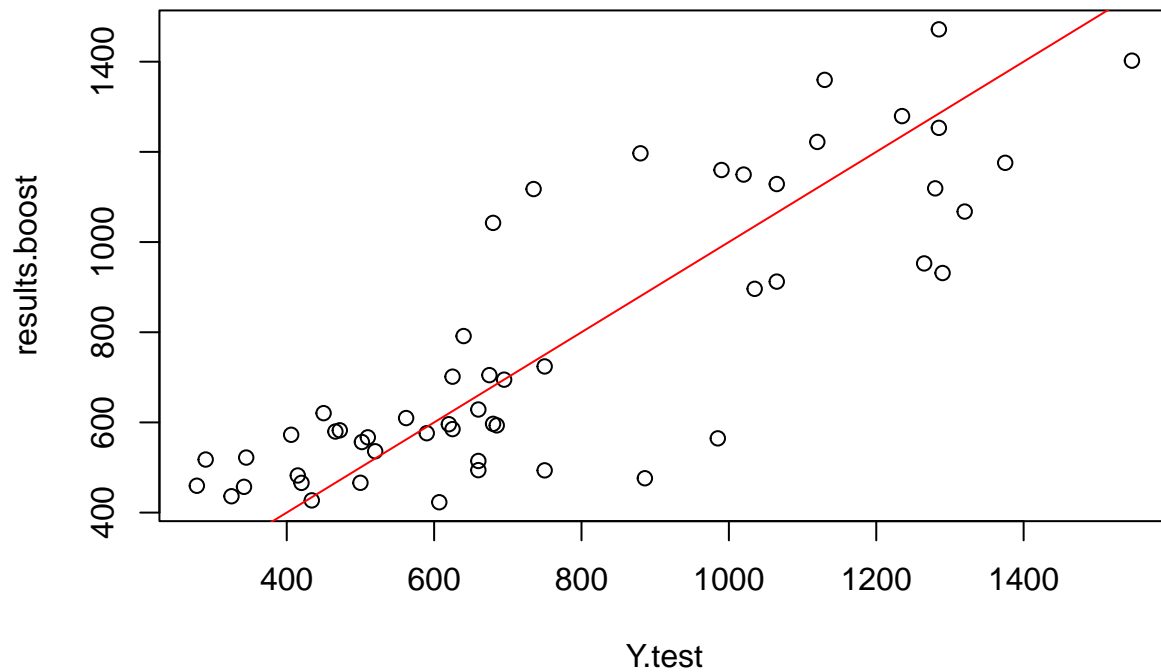
tree.f <- tree::tree(r ~ ., data = data.frame(cbind(X,r)),
na.action=na.pass)
r <- r - lambda * predict(tree.f, X)
plot(Y.test,predict(tree.f, X.test))
abline(0,1400,col="red")
```



```
cor(Y.test,predict(tree.f, X.test))
```

```
## [1] 0.8443184
```

```
results.boost <- 0*Y.test
B=2000
tree.full <- c()
tree.snipped <- c()
for(b in 1:B){
  tree.full[[b]] <- tree::tree(r ~ .,
    data = data.frame(cbind(X,r)), na.action=na.pass)
  tree.snipped[[b]] <- tree::tree(r ~ .,
    data = data.frame(cbind(X,r)), na.action=na.pass)
  r <- r - lambda * predict(tree.snipped[[b]], X)
  results.boost = results.boost + lambda *
    predict(tree.snipped[[b]], X.test)
}
plot(Y.test,results.boost)
abline(0,1,col="red")
```



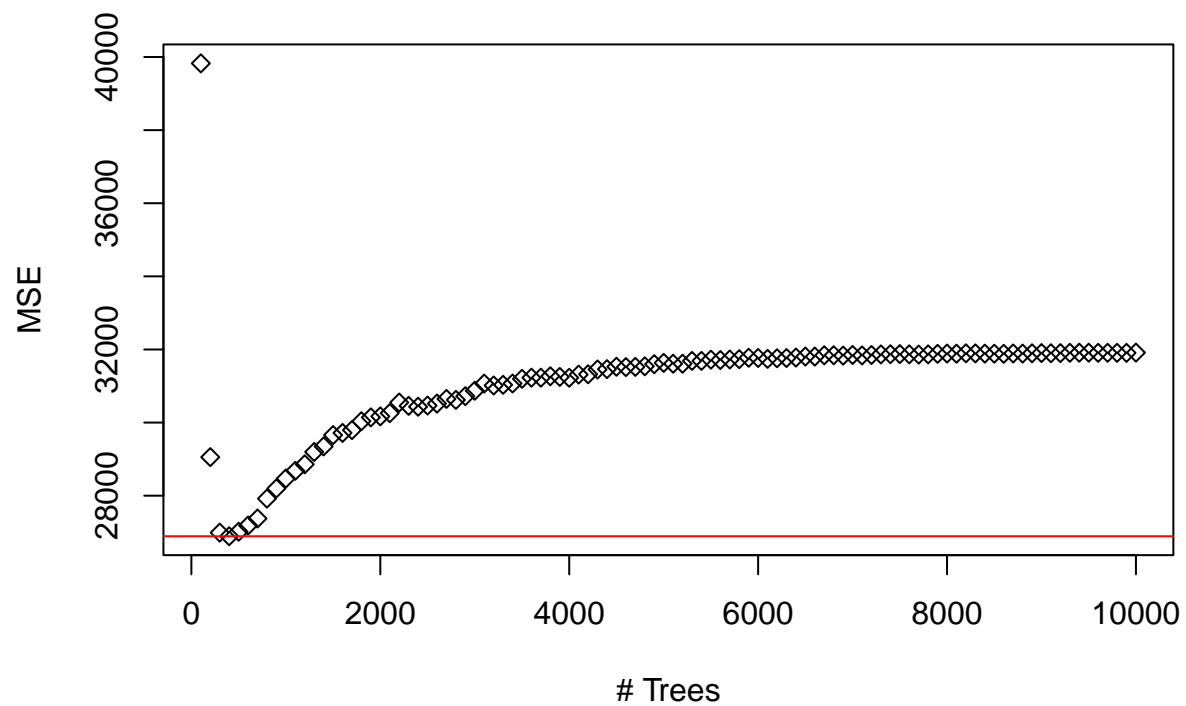
```
cor(Y.test,results.boost)
```

```
## [1] 0.8425456
```

```
boost.wine = gbm::gbm(Proline~., data = wine.tr,
distribution = "gaussian", n.trees = 10000,
shrinkage = 0.01, interaction.depth = 4)
```

```
n.trees = seq(from = 100, to = 10000, by = 100)
predmat = predict(boost.wine, newdata = wine.te,
n.trees = n.trees)
boost.err = with(wine.te,
apply( (predmat - Proline)^2, 2, mean) )
plot(n.trees, boost.err, pch = 23, xlab = "# Trees",
ylab = "MSE", main = "Boosting Test Error")
abline(h = min(boost.err), col = "red")
```

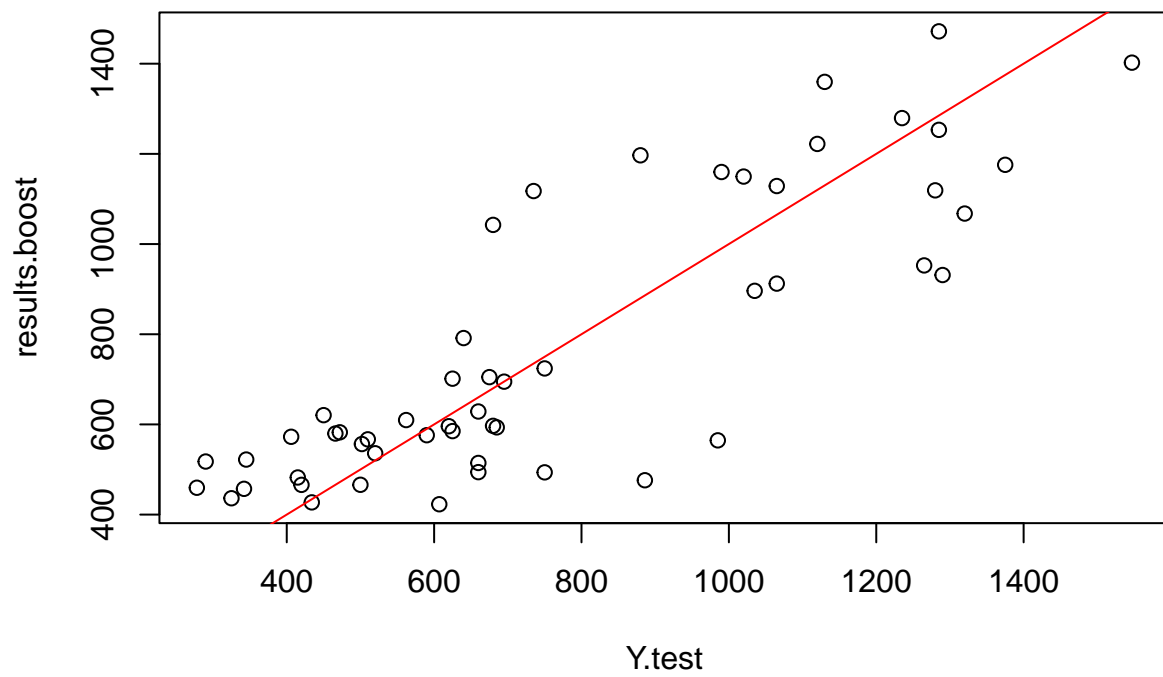
Boosting Test Error



```
which(boost.err==min(boost.err))
```

```
## 400  
## 4
```

```
results.boost.gbm = predict(boost.wine,  
newdata = wine.te, n.trees = 1200)  
plot(Y.test,results.boost)  
abline(0,1,col="red")
```



```
cor(Y.test,results.boost)
```

```
## [1] 0.8425456
```