BMI 651

Joshua Burkhart

February 22, 2016

HW4: APPENDIX

```
spam.df <- read.table("~/SoftwareProjects/StatisticalMethodsInCompBio/HW4/spam.data")
```

1

```
spam.df %>% dim()
```

```
[1] 4601   58
```

```
spam.df[,58] %>% table()
```

```
.
   0    1
2788 1813
```

```
spam.df[,58] %>% table() %>% as.vector() %>% .[1]/spam.df %>% nrow()
```

```
[1] 0.6059552
```

```
spam.df[,58] %>% table() %>% as.vector() %>% .[2]/spam.df %>% nrow()
```

```
[1] 0.3940448
```

2

2A

```
index <- sample(1:nrow(spam.df),round(0.5*nrow(spam.df)))
spam.df_train <- spam.df[index,]
spam.df_test <- spam.df[-index,]
```

2B

```r
duplicated(spam.df) %>% table()
```

```
.
FALSE  TRUE
 4210   391
```

```r
spam.df_unique <- unique(spam.df)

spam.df_unique %>% dim()
```

```
[1] 4210   58
```

```r
spam.df_unique[,58] %>% table()
```

```
.
   0    1
2531 1679
```

```r
spam.df_unique[,58] %>% table() %>% as.vector() %>% .[1]/spam.df_unique %>% nrow()
```

```
[1] 0.6011876
```

```r
spam.df_unique[,58] %>% table() %>% as.vector() %>% .[2]/spam.df_unique %>% nrow()
```

```
[1] 0.3988124
```

```r
index <- sample(1:nrow(spam.df_unique),round(0.5*nrow(spam.df_unique)))
spam.df_train <- spam.df_unique[index,]
spam.df_test <- spam.df_unique[-index,]

intersect(spam.df_train,spam.df_test) %>% nrow()
```

```
[1] 0
```

2C

```r
spam.df_train %>% dim()
```

```
[1] 2105   58
```

```r
spam.df_test %>% dim()
```

```
[1] 2105   58
```

2D

```
# training set
spam.df_train[,58] %>% table()
```

```
.
   0    1 
1267  838 
```

```
spam.df_train[,58] %>% table() %>% as.vector() %>% .[1]/spam.df_train %>% nrow()
```

```
[1] 0.6019002
```

```
spam.df_train[,58] %>% table() %>% as.vector() %>% .[2]/spam.df_train %>% nrow()
```

```
[1] 0.3980998
```

```
# test set
spam.df_test[,58] %>% table()
```

```
.
   0    1 
1264  841 
```

```
spam.df_test[,58] %>% table() %>% as.vector() %>% .[1]/spam.df_test %>% nrow()
```

```
[1] 0.6004751
```

```
spam.df_test[,58] %>% table() %>% as.vector() %>% .[2]/spam.df_test %>% nrow()
```

```
[1] 0.3995249
```

3

3A

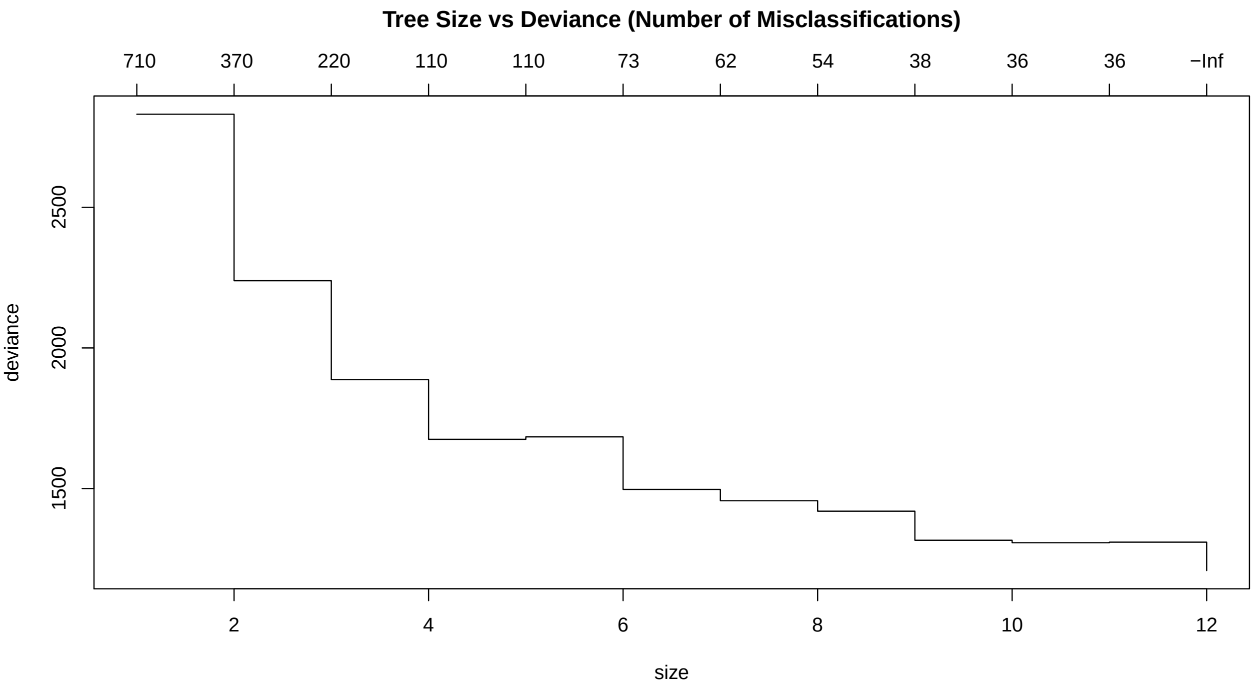
```
# applying factor() to the response forces production of a classification tree
# (Brian, A., & Ripley, M. B. (2016). Package " tree .")
spam.tree <- tree(factor(spam.df_train$V58)~.,data=spam.df_train)

# perform K=10 fold cross validation
spam.tree_cv <- cv.tree(spam.tree,K=10)

# prune the tree, allowing 12 terminal nodes
spam.tree_pruned <- prune.tree(spam.tree,best=12)
```

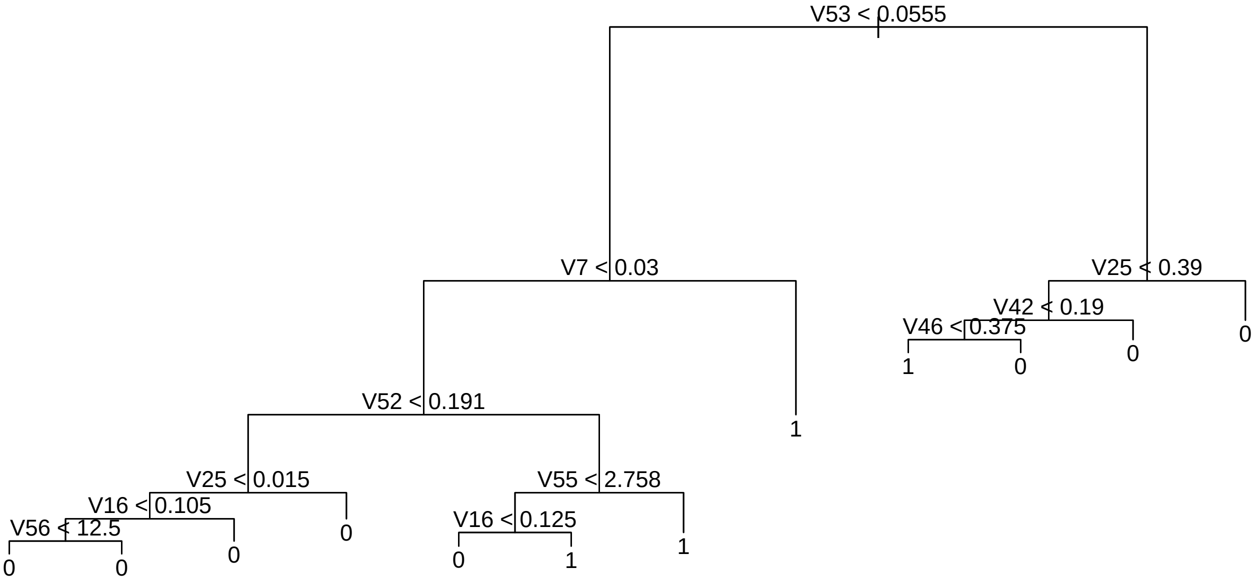
3B

```r
plot(spam.tree_cv,main="Tree Size vs Deviance (Number of Misclassifications)\n\n")
```



3C

```r
plot(spam.tree_pruned,main="'Pruned' Tree\n")
text(spam.tree_pruned)
```



3D

```r
# perform prediction on test dataset
spam.test_pred <- predict(spam.tree_pruned,spam.df_test[,-58],type="class")

# calculate misclassification percentage
table(as.integer(as.character(spam.test_pred)) == spam.df_test[,58]) %>% .[1] / length(spam.test_pred)
```

```
    FALSE
0.1054632
```

3E

```r
spam.tree_pruned %>% summary()
```

```
Classification tree:
tree(formula = factor(spam.df_train$V58) ~ ., data = spam.df_train)
Variables actually used in tree construction:
[1] "V53" "V7"  "V52" "V25" "V16" "V56" "V55" "V42" "V46"
Number of terminal nodes:  12
Residual mean deviance:  0.4819 = 1009 / 2093
Misclassification error rate: 0.08409 = 177 / 2105
```

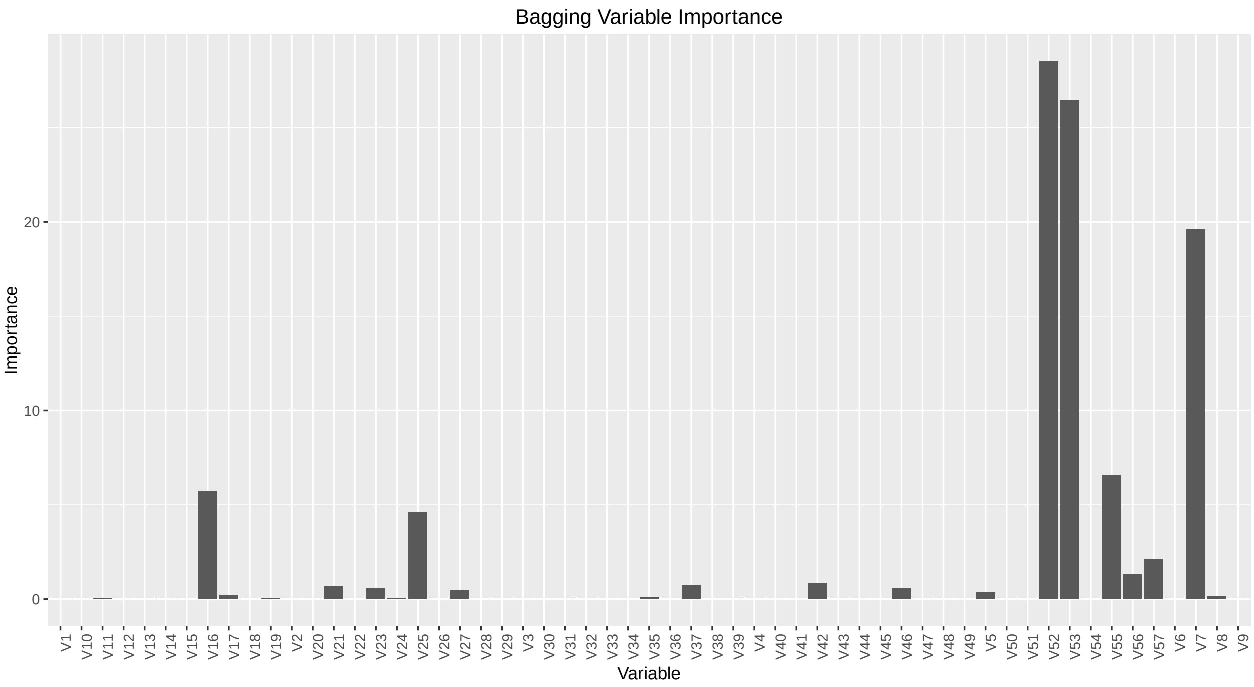
4

4A

```r
spam.df_train$V58 <- factor(spam.df_train$V58)
spam.bagging <- bagging(V58~.,data=spam.df_train,mfinal=100)
```

4B

```r
imp <- data.frame(spam.bagging$importance)
ggplot(data=imp, aes(x=rownames(imp),y=imp$spam.bagging.importance)) +
  geom_bar(stat="identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  xlab("Variable") +
  ylab("Importance") +
  ggtitle("Bagging Variable Importance")
```



4C

```
spam.df_test$V58 <- factor(spam.df_test$V58)
spam.bagging_test_pred <- predict(spam.bagging,spam.df_test,type="class")
spam.bagging_test_pred$error
```

```
[1] 0.0935867
```

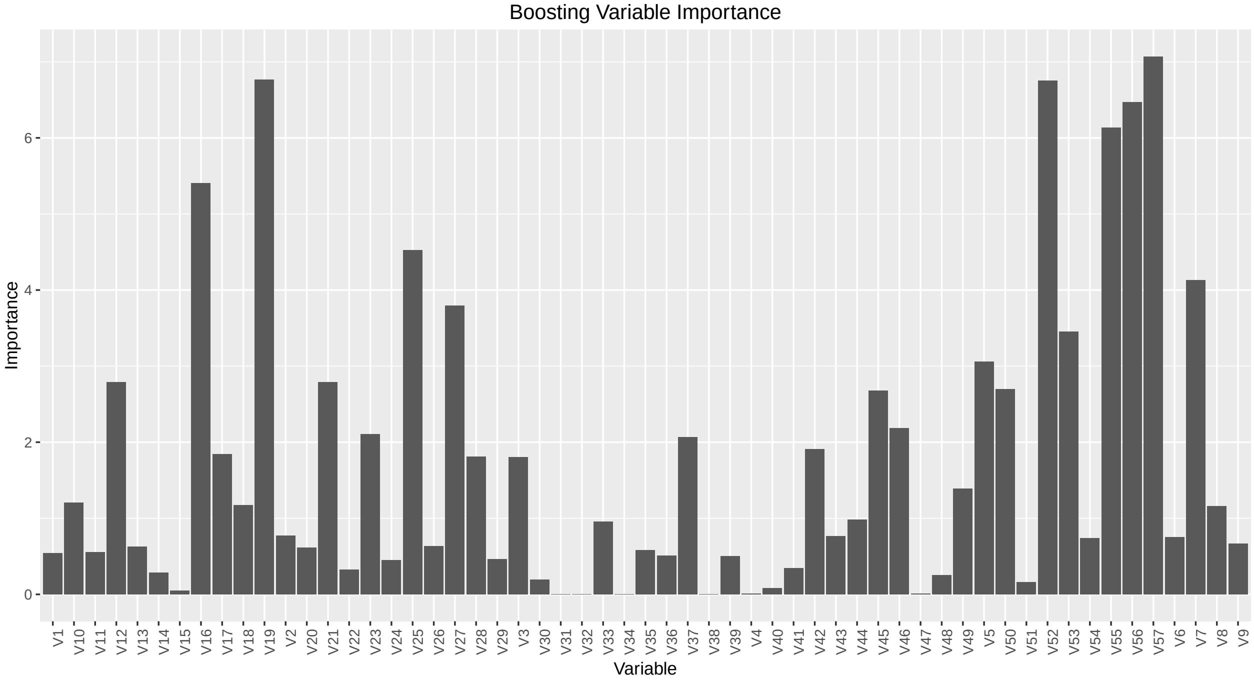
5

5A

```
spam.boosting <- boosting(V58~.,data=spam.df_train,mfinal=100)
```

5B

```
imp <- data.frame(spam.boosting$importance)
ggplot(data=imp, aes(x=rownames(imp),y=imp$spam.boosting.importance)) +
  geom_bar(stat="identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  xlab("Variable") +
  ylab("Importance") +
  ggtitle("Boosting Variable Importance")
```



5C

```
spam.boosting_test_pred <- predict(spam.boosting,spam.df_test,type="class")
spam.boosting_test_pred$error
```

```
[1] 0.05130641
```