**BUSINESS DATA MINING**

**(IDS 572)**

**Solutions to Homework 3**

**Group Members**

* Amey Pophali (apopha2@uic.edu)
* Karthik Varanasi (vvaran3@uic.edu)
* Mrinal Dhawan (mdhawa3@uic.edu)

**Problem 1 –**

**To see whether a classifier is actually working, we should compare it to a constant classifier**

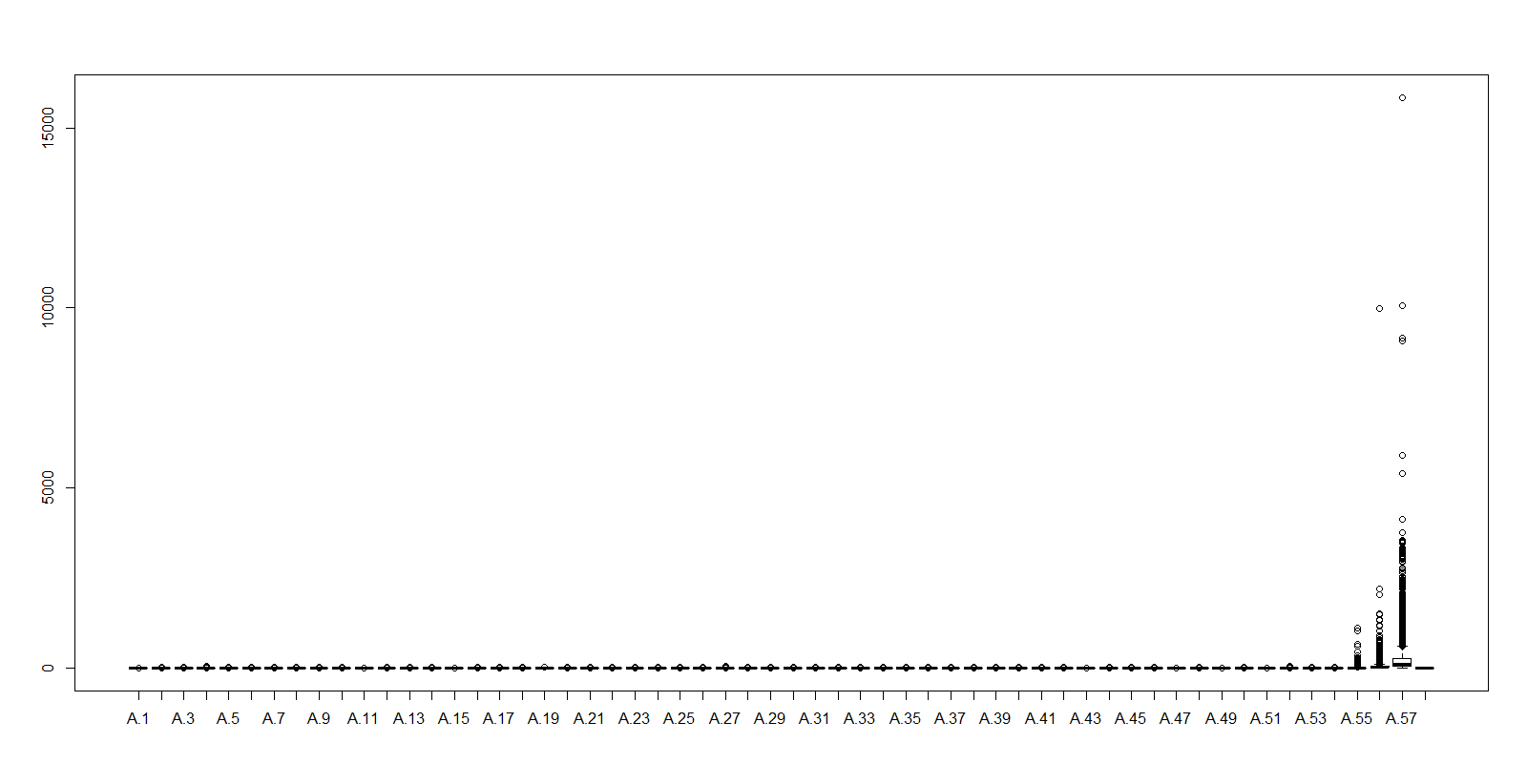
**which always predicts the same class, no matter what the input features actually are.**

Checking the missing values and outliers in ‘spam’ data.

sum(is.na(spam))

[1] 0

Hence there are no missing values in spam.



There are outliers in the data but these outliers are required for classification, we keep the data as is.

* **What fraction of the e-mails are actually spam?**

0.3940448

vari = table(spam$spam)

varip <- ((vari)/nrow(spam))\*100

varip

email spam

60.59552 39.40448

* **What should the constant classifier predict?**A constant classifier predicts the same target value irrespective of the input. Here The constant classifier should predict that all the values fall into a single category “E-mail”.
* **What is the error rate of the constant classifier?**

The error rate of the constant classifier can be found by comparing the predicted value to the actual value of the variable. The error is 39.40448%.

**Problem 2 –**

**Divide the data set at random into a training set of 2301 rows and a testing set of 2300**

**rows. Check that the two halves do not overlap (use intersect () function), and that they have the right number of rows. What fraction of each half is spam? (Do not hand in a list of 2301 row numbers.)**

This can be done using the mentioned commands –

#Divide the data into 2300 and 2301 rows

train.rows = sample(1:nrow(spam),2301,replace=FALSE)

training.data = spam[train.rows,]

testing.data = spam[-train.rows,]

#Check the result of the commands

dim(training.data)

[1] 2301 58

dim(testing.data)

[1] 2300 58

#Check intersection

intersect(rownames(training.data),rownames(testing.data))

character(0)

sum(training.data$spam=="spam")/2301

[1] 0.3920035

sum(testing.data$spam=="spam")/2300

[1] 0.396087

**Problem 3 –**

**Remember to show your work by including your code.**

* **Fit a classification tree to the training data. Prune the tree by cross-validation (see below).**

**Include a plot of the CV error versus tree size, a plot of the best tree, and its error rate on the**

**testing data. Which variables appear in the tree?**

**We plot the tree using the following commands –**

#Create tree

r\_tree = tree(spam~., training.data)

#Summary of the tree created

summary(r\_tree)

Classification tree:

tree(formula = spam ~ ., data = training.data)

Variables actually used in tree construction:

[1] "A.52" "A.7" "A.27" "A.53" "A.25" "A.56" "A.5" "A.55" "A.46" "A.21" "A.49"

Number of terminal nodes: 16

Residual mean deviance: 0.4598 = 1051 / 2285

Misclassification error rate: 0.08953 = 206 / 2301

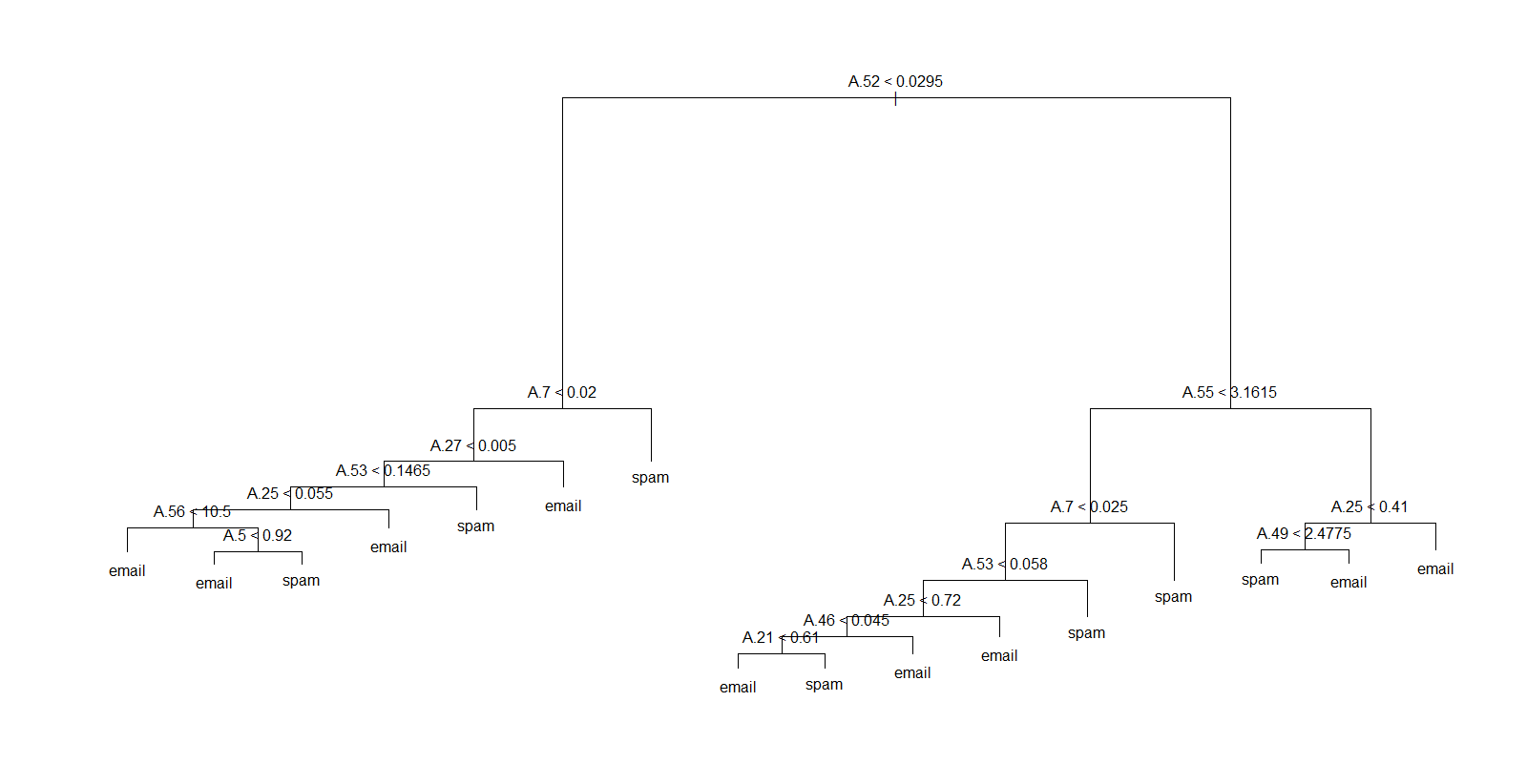
The tree is made using the attributes are A.52, A.7, A.27, A.53, A.25, A.56, A.5, A.55, A.46, A.21 and A.49.

The misclassification rate as seen from the summary is 0.08953.

Plotting the tree using the command –

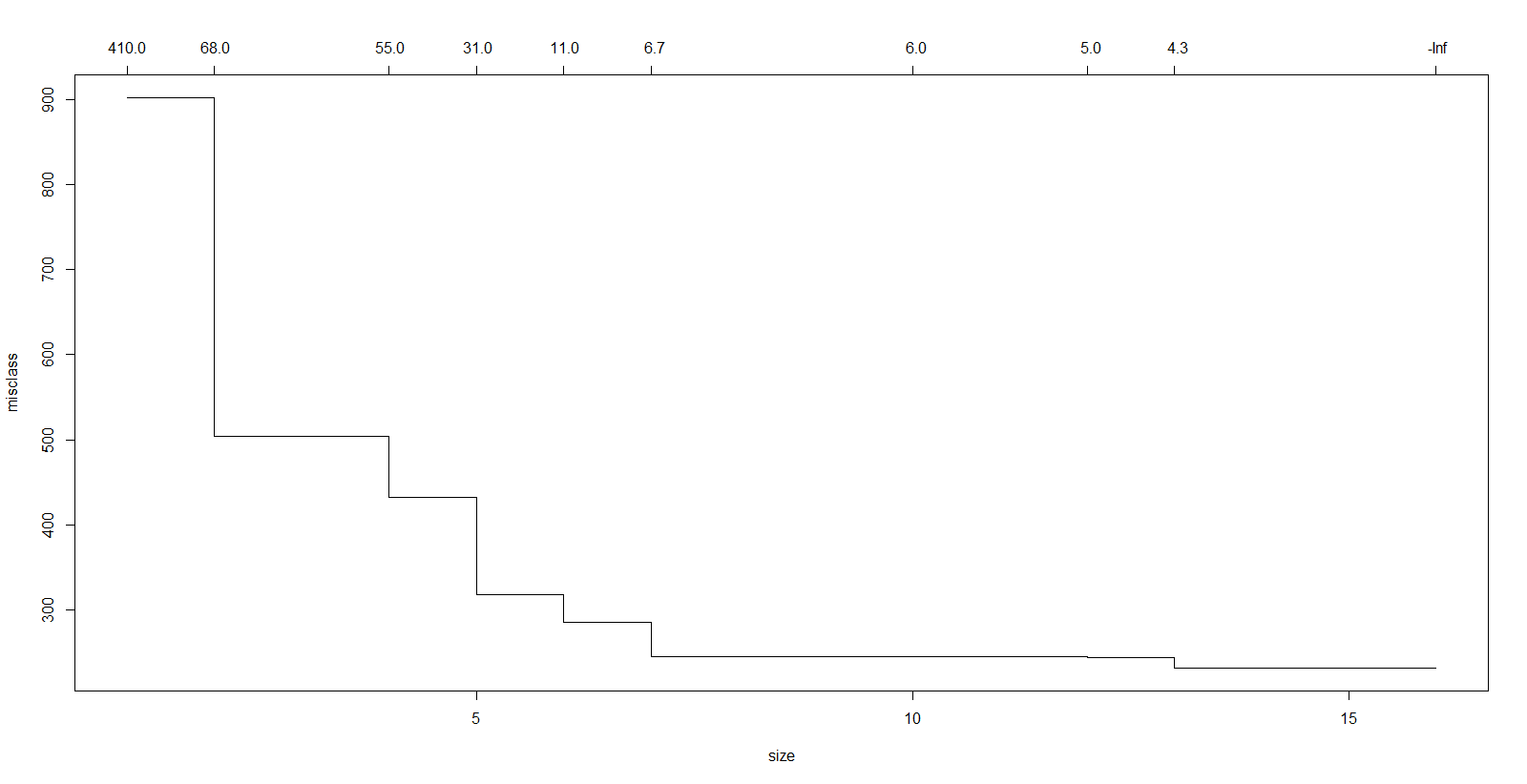
plot(r\_tree)

text(r\_tree, use.n=T, xpd=T)

#Pruning the tree with cross validation

r\_tree.cv = cv.tree(r\_tree,method="misclass")

plot(spam.cv)



* **Use bagging to fit an ensemble of 100 trees to the training data. Report the error rate of the**

**ensemble on the testing data. Include a plot of the importance of the variables, according to**

**the ensemble.**

This can be done by using the following commands –

install.packages("adabag")

library(adabag)

#Create

hw\_bag = bagging(spam ~ ., data = Train.data, minsplit = 1, cp = 0)

summary(hw\_bag)

summary(hw\_bag)

Length Class Mode

formula 3 formula call

trees 100 -none- list

votes 4602 -none- numeric

prob 4602 -none- numeric

class 2301 -none- character

samples 230100 -none- numeric

importance 57 -none- numeric

terms 3 terms call

call 5 -none- call

sum(hw\_bag$class != training.data$spam)/nrow(training.data)

[1] 0.3928727

predict(hw\_bag,newdata= testing.data)$error

[1] 0.08869565

The error rates are 39.2% and 8.8%.

* **Which (if any) of these methods out-performs the constant classifier?**

**Problem 4 –**

**Pick the prediction method from the previous problem with the lowest error rate.**

* **What fraction of the spam e-mails in the training set did it not classify as spam?**
* **What fraction of the genuine e-mails in the testing set did it classify as spam?**
* **What fraction of e-mails it classified as spam were actually spam?**