# Random Forests Worksheet

### 

### Cleaning data and using the randomForest package in R

In this worksheet, we will predict income of individuals, similar to the rpart worksheet, but with a more powerful algorithm, randomForest. The first 2 questions do not require coding.

1. Given the following iris flower data, consider a single tree within a randomForest.

Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
#1 5.5 3.5 1.3 0.2 setosa  
#2 5.2 3.4 1.4 0.2 setosa  
#3 6.3 2.9 5.6 1.8 virginica  
#4 7.7 2.8 6.7 2.0 virginica  
#5 5.1 3.8 1.5 0.3 setosa  
#6 6.5 2.8 4.6 1.5 versicolor  
#7 4.9 2.5 4.5 1.7 virginica  
#8 4.6 3.1 1.5 0.2 setosa  
#9 6.0 2.9 4.5 1.5 versicolor

Give an example of which rows in the iris flower data above could be used for

1) creating the tree and

2) the OOB sample for that tree.

3) In addition, give an example of the set of variables that could be used at a single split in this tree given mtry= 2 and we are predicting 'Species'.

1. Consider variable j. The Variable Importance Score =

OOB error with permuted j - OOB error without permuted j

What does a high variable importance score imply? One sentence only.

**Aim: Clean income.csv dataset and use randomForest algorithm to predict income class**

1. Set your working directory (where you have stored your datafile) using either setwd() or the 'Session' menu

setwd("C:/Users/Tim/OneDrive/SSA\_bigdata\_course/trees")

1. Install relevant packages: randomForest

install.packages('randomForest')

1. Load packages:

library('randomForest')

library('ggplot2')  
library('tidyr') #we will use gather()

1. Load dataset and save as 'a'. Delete first column at the same time

a <- read.csv('income.csv')[,-1]

1. We should do some cleaning: neaten up some of the variables Let's look at table summaries for three variables: occupation, education, marital status, workclass

table(a$education, a$class)

table(a$occupation, a$class)

table(\_\_\_\_\_\_\_?, a$class)

table(\_\_\_\_\_\_\_?, a$class)

Are any of the category levels representing a similar idea and have similar stats with respect to the class we're predicting? Do any of the variable levels have a very low count that could be combined into another level? Give an example of one level in one variable like this and why:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_?

1. Creating clear distinct levels within a variable helps with the final prediction. Before we can combine levels, first convert factor variables to character variables - this allows us to change the names of the levels

a$occupation = as.character(a$occupation)  
a$education = as.character(\_\_\_\_\_\_\_\_\_?)  
a$\_\_\_\_\_\_\_\_\_? = as.character(\_\_\_\_\_\_\_\_\_?)  
\_\_\_\_\_\_\_\_\_\_\_? = as.character(\_\_\_\_\_\_\_\_\_?)

1. Replace variable levels with other labels. Target levels with low numbers or similar meanings/stats

We'll be using the function gsub, look it up under help.

?gsub

**Occupation:**

table(a$occupation, a$class)

#combine Priv-house-serv and Other-service  
a$occupation<- gsub("Priv-house-serv", "Service", a$occupation)  
a$occupation<- gsub(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_?, "Service", \_\_\_\_\_\_\_\_\_\_\_\_\_?)

**Education:**

table(a$education, a$class)

#create highschool dropout level - "HS.dropout", check for similar

stats. Include all levels which imply the student dropped out in highschool.  
# | means 'or'

a$education<- gsub("10th|11th|12th|9th", "HS.dropout", \_\_\_\_\_\_?)  
  
#create primary school dropout level and include preschool - "PS.dropout"  
a$education<- gsub(\_\_\_\_insert 4 levels\_\_\_\_?, \_\_\_\_\_\_?,\_\_\_\_\_?)  
  
#meld 'some college' with HS-grad  
a$\_\_\_\_\_\_? <- gsub("Some-college", \_\_\_\_?, \_\_\_\_\_\_?)  
  
#merge Assoc-acdm and Assoc-voc. Similar stats  
a$\_\_\_\_\_\_?<- \_\_\_\_\_?(\_\_\_insert the 2 levels\_\_\_?, "Associates", \_\_\_?)

**marital.status:**

table(a$marital.status, a$class)

#combine married-AF-spouse (tiny) and Married-civ-spouse.   
a$marital.status<- \_\_\_?(\_\_\_\_\_2 levels\_\_\_?, "Married", \_\_\_\_?)  
  
# combine Divorced, Married-spouse-absent and Separated b/c similar meaning  
\_\_\_\_\_? <- \_\_\_?(\_\_3 levels\_\_?, "Prev.Married", \_\_\_?)

**workclass:**

table(a$workclass, a$class)

#remove class 'Without-pay' - class is obvious  
a<- a[!(a$workclass == 'Without-pay'),]  
  
# combine Loval-gov and State-gov to non-fed government called ‘lower.gov’  
a$workclass<- \_\_?(\_\_\_?, \_\_?, \_\_\_?)

Convert back to factor:

a$occupation = as.factor(a$occupation)  
a$education = as.factor(\_\_\_?)  
a$marital.status = \_\_\_\_?(\_\_\_?)   
a$workclass = \_\_\_?(\_\_\_?)

View cleaned data:

summary(a)

1. Note the ratio of cases which are <=50K and >50k. What could this mean for the predictive power of the model?

We won't worry about doing anything about it in this worksheet.

1. Create test and train set.

set.seed(100)  
sample<- sample(1:nrow(a), 2/3\*nrow(a), replace=F)  
train<- a[sample,]  
test<- a[-sample,]

1. Build a first-run random forest model with train data. We will tune mtry next, set at sqrt(number of variables) for now. Set number of trees = 1000. Set test dataframe as test data

# look up which arguments you can use in randomForest  
?randomForest  
#examples are at the bottom of the help file

We will use the arguments: formula, data, mtry, ntree, importance, xtest, ytest

rf.fit<- randomForest(class~., data=?, mtry= ?, ntree=?, xtest=test[,1:9],  
 ytest=test[,10])  
  
#view results  
rf.fit

# OOB error for classification is err.rate of all trees up to i-th tree. Which row are we most interested in?  
rf.fit$err.rate[?]

print result below:

\_\_\_\_\_\_\_\_\_?

1. Manually tune mtry parameter with a for-loop. Build 6 trees using mtry = 1,2,3,4,5 and plot OOB error and test error vs. mtry. Similar to tuning cp in single trees.

#set up two NULL vectors to contain the OOB error rate and test error rate  
oob.err<-double(5)  
test.err<-double(5)  
  
for(i in 1:5){  
 rf.fit<-randomForest(formula=?, data=?, mtry=?, ntree=?, xtest=?  
 ytest=?  
 oob.err[i]<-rf.fit$err.rate[1000]  
 test.err[i]<- rf.fit$test$err.rate[1000]  
   
 #print progress of each i while computing  
 cat(i," ")  
}

#view two original vectors  
oob.err

test.err

#plot error vectors against mtry  
#first create data frame with all data  
plot.df<- data.frame('mtry' = c(1:5), 'oob.err'=oob.err, 'test.err' = test.err)  
print(plot.df)

# ‘gathering’ is often required to work well with ggplot grammar of graphics

# to understand how the function works, type:

?gather

plot.df<- gather(plot.df, key=variable, value =value, -mtry)  
print(plot.df)  
  
#plot values of OOB error and test error vs mtry using ggplot. Colour lines by either OOB or test. Each column variable should be used in aes()  
ggplot(data= plot.df, aes(x=?, y=value, colour=? )) +  
 geom\_line(size =1) +  
 geom\_point(size=3)+  
 labs(y='error')

Which mtry is optimal?

1. Make final forest and look at test set confusion matrix, print below. Better than single tree?

#set importance = TRUE  
rf.fit<- randomForest(formula?, data=?, mtry=?, ntree=?, xtest=test[,1:9],  
 ytest=test[,10], importance = TRUE)  
  
#view results  
rf.fit

1. Look at variable importances

importance(rf.fit)

varImpPlot(rf.fit, scale=FALSE)

1. How are the variable importances different for the two different methods?

Note: in reality, randomForest variable importances are biased esp. towards correlated variables and variables with many categories. Should use cforest() in 'party' package for purpose of variable importance.

Do you think we would have correlated variables?

1. Optional: tune mtry parameter using tuneRF() function. Google to help if needed.

?tuneRF

## End