Assignment 2

In this assignment, you will have the chance to practice building a random forest model. Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees. Random forests are bagged decision tree models that split on a subset of features on each split.

Below is an example of how random forest models can be trained using the caret package. In this example, we use “PimaIndiansDiabetes” dataset that is part of the mlbench library. This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

The dataset consists of several medical predictor variables and one target variable, Outcome. Predictor variables includes the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

Pregnancies:Number of times pregnant

Glucose:Plasma glucose concentration a 2 hours in an oral glucose tolerance test

BloodPressure:Diastolic blood pressure (mm Hg)

SkinThickness: Triceps skin fold thickness (mm)

Insulin: 2-Hour serum insulin (mu U/ml)

BMI: Body mass index (weight in kg/(height in m)^2)

Age:Age (years)

Diabetes : The target (outcome) variable with positive and negative levels.

library(mlbench)

## Warning: package 'mlbench' was built under R version 3.4.4

#install using the following command if the library is not installed  
#install.packages('mlbench')  
library(caret)

## Warning: package 'caret' was built under R version 3.4.4

## Loading required package: lattice

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.4.4

# This will load the PimaIndiansDiabetes dataset  
data(PimaIndiansDiabetes)  
  
#check the levels  
levels(PimaIndiansDiabetes$diabetes)

## [1] "neg" "pos"

set.seed(2)  
intrain <- createDataPartition(y = PimaIndiansDiabetes$diabetes, p= 0.7, list = FALSE)  
training <- PimaIndiansDiabetes[intrain,]  
testing <- PimaIndiansDiabetes[-intrain,]  
  
rf.model<-train(diabetes~.,data=training,method='rf')  
rf.model

## Random Forest   
##   
## 538 samples  
## 8 predictor  
## 2 classes: 'neg', 'pos'   
##   
## No pre-processing  
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 538, 538, 538, 538, 538, 538, ...   
## Resampling results across tuning parameters:  
##   
## mtry Accuracy Kappa   
## 2 0.7556950 0.4418766  
## 5 0.7502580 0.4338101  
## 8 0.7434803 0.4195052  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was mtry = 2.

# You can define the exact values of mtry parameters to be tested by caret  
#Lets say we want to check 2,3 and 9 as values for mtry  
# Forgot what mtry was? it's the number of variables available for splitting at each tree node!  
  
Grid\_Serach <- expand.grid(.mtry=c(2,3,8))  
  
rf.model2<-train(diabetes~.,  
 data=training,  
 method='rf',  
 tuneGrid=Grid\_Serach)  
rf.model2

## Random Forest   
##   
## 538 samples  
## 8 predictor  
## 2 classes: 'neg', 'pos'   
##   
## No pre-processing  
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 538, 538, 538, 538, 538, 538, ...   
## Resampling results across tuning parameters:  
##   
## mtry Accuracy Kappa   
## 2 0.7533980 0.4450398  
## 3 0.7517710 0.4435371  
## 8 0.7392433 0.4188302  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was mtry = 2.

probs <-predict(rf.model2,testing,type="prob")  
head(probs)

## neg pos  
## 2 0.856 0.144  
## 3 0.428 0.572  
## 4 0.976 0.024  
## 15 0.438 0.562  
## 16 0.604 0.396  
## 25 0.456 0.544

pred\_class <-predict(rf.model2,testing,type="raw")  
head(pred\_class)

## [1] neg pos neg pos neg pos  
## Levels: neg pos

**Your Task**

Your task is to use the above example to build a random forest classification model to predict the type of a breast tumor (benign or malignant).

The data can be loaded using the mlbench library using

library(mlbench)  
data(BreastCancer)

A data frame with 699 observations on 11 variables, one being a character variable, 9 being ordered or nominal, and 1 target class.

[,1] Id Sample code number

[,2] Cl.thickness Clump Thickness

[,3] Cell.size Uniformity of Cell Size

[,4] Cell.shape Uniformity of Cell Shape

[,5] Marg.adhesion Marginal Adhesion

[,6] Epith.c.size Single Epithelial Cell Size

[,7] Bare.nuclei Bare Nuclei

[,8] Bl.cromatin Bland Chromatin

[,9] Normal.nucleoli Normal Nucleoli

[,10] Mitoses Mitoses

[,11] Class Class

Submission:

Submit a pdf of your r markdown with the results shown in the file. Similar to my example above (you can alternatively include screen shots- however your submission should be one single pdf). In your submission:

1. Try examine the following values for the mtry variable 2 6 8
2. Show the confusion matrix of predictions on the test data (in the above example, I have the raw (class) predictions. You need to compare them with the ground truth values (i.e. testing$Class). Hint: table( ) function