Quiz Week3

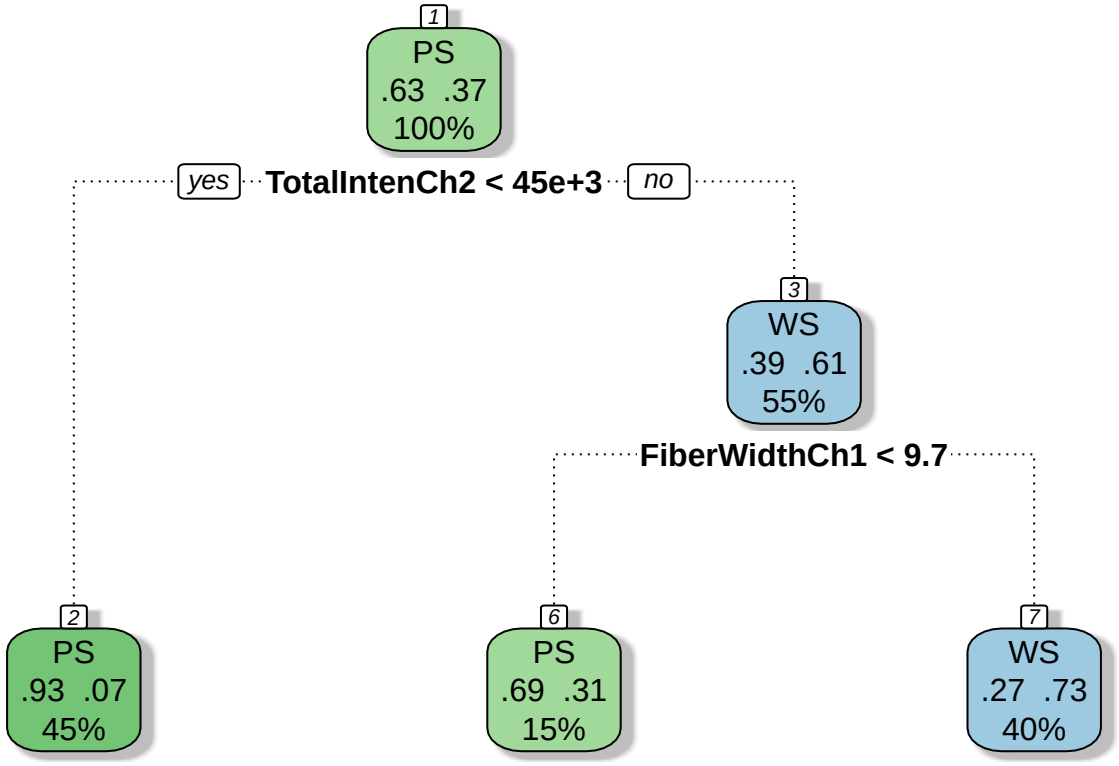
Question 1

1. Subset the data to a training set and testing set based on the Case variable in the data set.
2. Set the seed to 125 and fit a CART model with the rpart method using all predictor variables and default caret settings.
3. In the final model what would be the final model prediction for cases with the following variable values:

```
library(AppliedPredictiveModeling)
data(segmentationOriginal)
library(caret)
trainIndex <- segmentationOriginal$Case=="Train"

table(segmentationOriginal$Case)
```

```
##
##  Test Train
##  1010  1009
```

```
train <- segmentationOriginal[trainIndex,-2]
test <- segmentationOriginal[!trainIndex,-2]

set.seed(125)
fit <- train(Class~.,method = "rpart",data = train)

library(partykit);library(rattle)
# Plot graph and check visually
fancyRpartPlot(fit$finalModel,sub="")
```



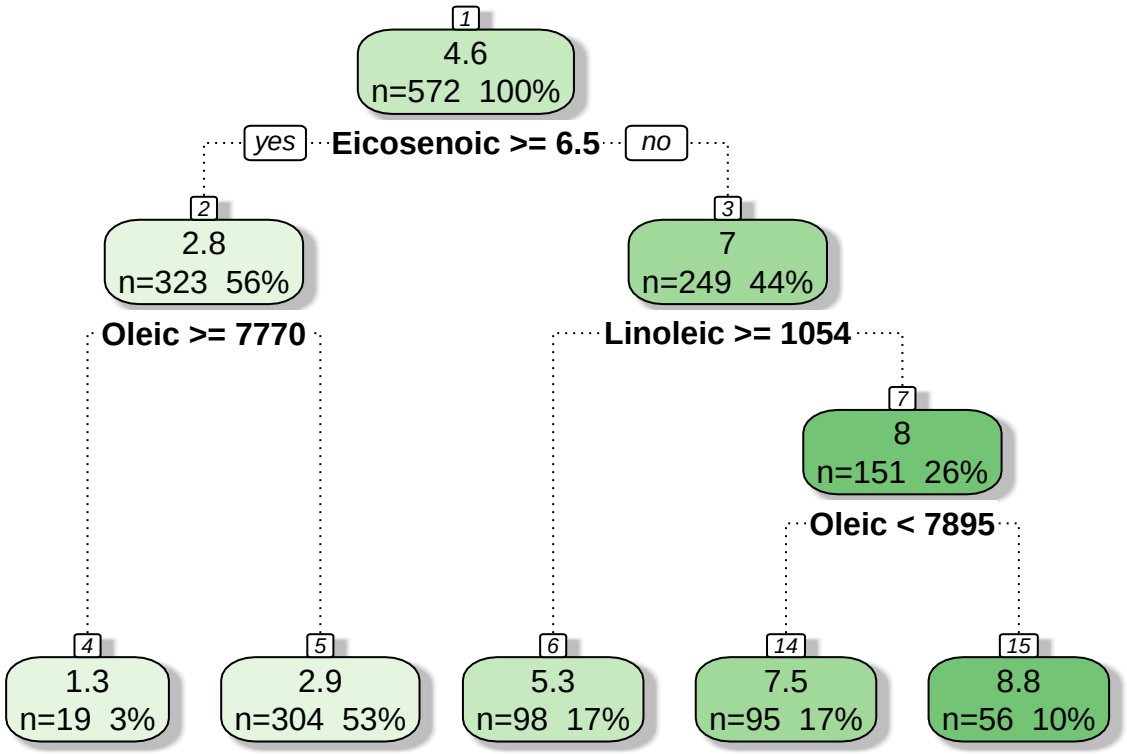

```
# Add data to use model to check plus a test sample (a WS sample (form graph))
values <-list(c(TotalIntenCh2 = 23000, FiberWidthCh1 = 10, PerimStatusCh1 = 2),
              c(TotalIntenCh2 = 50000, FiberWidthCh1 = 10, VarIntenCh4 = 100),
              c(TotalIntenCh2 = 57000, FiberWidthCh1 = 8, VarIntenCh4 = 100),
              c(FiberWidthCh1 = 8, VarIntenCh4 = 100, PerimStatusCh1 = 2 ),
              c(TotalIntenCh2 = 53000,FiberWidthCh1 =12, VarIntenCh4 = 100, PerimStatusCh1 = 2 ))

OutPut <- vector()
for (i in seq(1:length(values))){
      inPuts <- as.data.frame(t(values[[i]]))
      #Drop Case and New Variables
      DataNa <- train[1,-c(2,match(colnames(inPuts),colnames(train)))]
      DataNa[1,]<-NA
      DataNa <- cbind(DataNa,inPuts)
      fitPred <- predict(fit,newdata = DataNa,,na.action = na.pass )
      OutPut[i] <- as.vector(fitPred)
      }
print(OutPut)
```

```
## [1] "PS" "WS" "PS" "PS" "WS"
```

Question 3

hese data contain information on 572 different Italian olive oils from multiple regions in Italy. Fit a classification tree where Area is the outcome variable. Then predict the value of area for the following data frame using the tree command with all defaults

```
library(caret)
library(pgmm)
data(olive)

library(rpart)
fitTree <- rpart(Area~.,data=olive)
library(partykit);library(rattle)
fancyRpartPlot(fitTree,sub="")
```




```
newdata = as.data.frame(t(colMeans(olive)))

predict(fitTree,newdata)
```

```
##     1
## 2.875
```

Question 4

Set the seed to 13234 and fit a logistic regression model (method="glm", be sure to specify family="binomial") with Coronary Heart Disease (chd) as the outcome and age at onset, current alcohol consumption, obesity levels, cumulative tabacco, type-A behavior, and low density lipoprotein cholesterol as predictors. Calculate the misclassification rate for your model using this function and a prediction on the "response" scale:

```
missClass = function(values,prediction){sum(((prediction > 0.5)*1) != values)/length(values)}
```

```
library(ElemStatLearn)
library(caret)
data(SAheart)
set.seed(8484)
train = sample(1:dim(SAheart)[1],size=dim(SAheart)[1]/2,replace=F)
trainSA = SAheart[train,]
testSA = SAheart[-train,]

set.seed(13234)
fitGLM <- train(chd~age+alcohol+obesity+tobacco+typea+ldl,method="glm",family="binomial",data =trainSA )

missClass <- function(values,prediction){sum(((prediction > 0.5)*1) != values)/length(values)}


prediction <- predict(fitGLM,trainSA)
values <- trainSA$chd
trainset <- missClass(values,prediction)

prediction <- predict(fitGLM,testSA)
values <- testSA$chd
testset <- missClass(values,prediction)

c(testset,trainset)
```

```
## [1] 0.3116883 0.2727273
```

Question 5

Set the variable y to be a factor variable in both the training and test set. Then set the seed to 33833. Fit a random forest predictor relating the factor variable y to the remaining variables. Read about variable importance in random forests here: http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#ooberr The caret package uses by defualt the Gini importance. Calculate the variable importance using the varImp function in the caret package. What is the order of variable importance?

```
library(ElemStatLearn)
data(vowel.train)
data(vowel.test)
library(caret)
vowel.train$y <- as.factor(vowel.train$y)
vowel.test$y <- as.factor(vowel.test$y )
set.seed(33833)

fitTrain <- train(y~.,method="rf",data=vowel.train)
```

```
## Loading required package: randomForest
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
varImp(fitTrain)
```

```
## rf variable importance
##
##      Overall
## x.1  100.000
## x.2   99.895
## x.5   45.292
## x.6   30.265
## x.8   25.512
## x.4   10.568
## x.3   10.133
## x.9    9.524
## x.7    6.514
## x.10   0.000
```

```
set.seed(33833)
fitTrain2 <- randomForest(y~.,method="rf",data=vowel.train)

cbind(order(fitTrain2$importance,decreasing = T),sort(fitTrain2$importance,decreasing = T))
```

```
##       [,1]     [,2]
##  [1,]    2 91.24009
##  [2,]    1 89.12864
##  [3,]    5 50.25539
##  [4,]    6 43.33148
##  [5,]    8 42.92470
##  [6,]    4 34.24433
##  [7,]    9 33.37031
##  [8,]    3 33.08111
##  [9,]    7 31.88132
## [10,]   10 29.59956
```