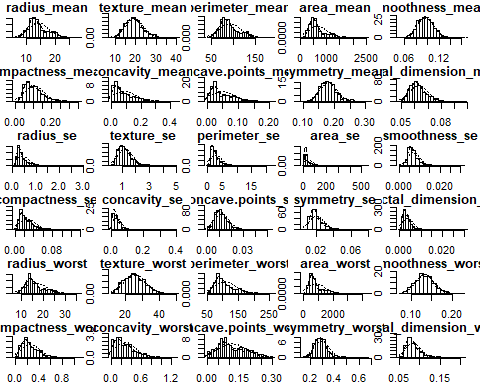
Decision Trees

Brian Seggebruch

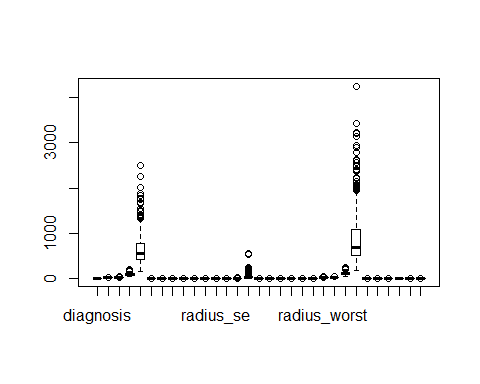
March 2, 2019

Data prep (output not shown in knitted document for the sake of space):

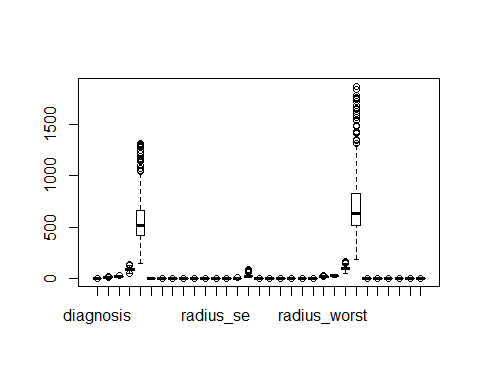
# Data cleaning  
breast\_cancer<-read.csv("wisconsin\_breast\_cancer.csv", header = TRUE)  
breast\_cancer<-breast\_cancer[2:32]  
breast\_cancer\_varnames<-read.csv("variable\_names.csv", header = TRUE)  
breast\_cancer  
head(breast\_cancer)  
names(breast\_cancer)  
  
is.na(breast\_cancer)  
breast\_cancer[!complete.cases(breast\_cancer),]  
  
# Histogram distribution  
multi.hist(breast\_cancer[,sapply(breast\_cancer, is.numeric)])



# Identify and remove outliers  
breast\_cancer.cat<-breast\_cancer[1]  
breast\_cancer.num<-breast\_cancer[2:31]  
remove\_outliers <- function(x, na.rm = TRUE, ...) {  
 qnt <- quantile(x, probs=c(.25, .75), na.rm = na.rm, ...)  
 H <- 1.5 \* IQR(x, na.rm = na.rm)  
 y <- x  
 y[x < (qnt[1] - H)] <- NA  
 y[x > (qnt[2] + H)] <- NA  
 y  
}  
breast\_cancer.noout.ma <- apply(breast\_cancer[2:31], 2, remove\_outliers)  
breast\_cancer.noout.num<-data.frame(breast\_cancer.noout.ma)  
breast\_cancer.noout<-cbind(breast\_cancer.cat,breast\_cancer.noout.num)  
breast\_cancer.noout.nona<-na.omit(breast\_cancer.noout)  
boxplot(breast\_cancer)



boxplot(breast\_cancer.noout.nona)



Our dataset contains diagnostic data for 579 cancer-screenings, digitized from images of a “fine needle aspirate of mass” procedure. The data is provided by the University of Wisconsin and is intended to be used to help predict whether a mass of cells is malignant or benign. There are ten real-valued variables measured for each record. They are, (from the dataset documentation):

1. radius (mean of distances from center to points on the perimeter)
2. texture (standard deviation of gray-scale values)
3. perimeter
4. area
5. smoothness (local variation in radius lengths)
6. compactness (perimeter^2 / area - 1.0)
7. concavity (severity of concave portions of the contour)
8. concave points (number of concave portions of the contour)
9. symmetry
10. fractal dimension (“coastline approximation” - 1)

Each of these measures is a taken from the cell nuclei present in the image generated from the procedure. The mean, standard error, and “worst” (largest) value are calculated for each image and recorded. Therefore, we have 30 variables (10 real-value measurements \* 3 statistically derived values). Each variable is recorded with four significant digits. There are 357 benign classifications and 212 malignant classifications.

We want to understand our data, so we preview it with a few R functions.

names(breast\_cancer)  
head(breast\_cancer)  
summary(breast\_cancer)

Splitting into test/train:

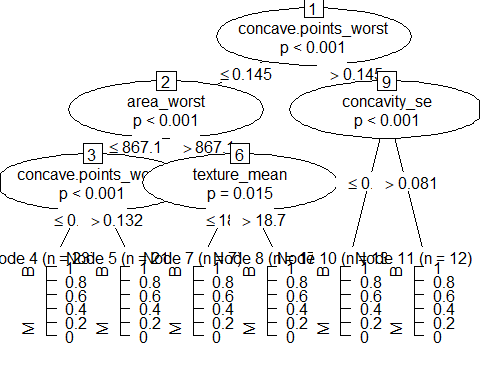
set.seed(231654)  
  
## 75% of the sample size  
sample\_size <- floor(0.75 \* nrow(breast\_cancer))  
  
set.seed(2356498)  
train\_index <- sample(seq\_len(nrow(breast\_cancer)), size = sample\_size)  
  
train <- breast\_cancer[train\_index, ]  
test <- breast\_cancer[-train\_index, ]

Here we do some minor data cleansing, removing fields we don’t want in our model:

train$id <- NULL  
train$X <- NULL  
test$id <- NULL  
test$X <- NULL

Fitting our model. The Gini Impurity algorithm helps us choose splits that build our tree to be the most accuracte and most direct. The actual formual can be written as: , where is the fraction of items labeled with class and is the number of classes we can choose from:

set.seed(3212)  
fit <- ctree(diagnosis ~ ., data = train)

Now we plot our model to see what splits were made and with what values: 

Looking at one specific path from the tree that was created, we can follow the logic and see that our output given these filtering parameters is more favorble for predicting a classification:

train[train$concave.points\_worst<=0.142 & train$area\_worst>947.9,1]

## [1] B M M B M M M M M M M M M M M M M  
## Levels: B M

Measuring accuracy:

testoutput <- as.matrix(as.character(predict(fit, newdata = test)))  
(model\_accuracy <- mean(testoutput == test$diagnosis))

## [1] 0.951049

Showing the side-by-side of our predicted vs. the actual classifications:

sidebyside <- as.data.frame(as.character(test$diagnosis))  
sidebyside$predicted <- testoutput  
names(sidebyside) <- c('observed', 'predicted')  
sidebyside

Testing the hypothesis: Our explanatory variables have a significant impact on the outcome of classification…:

probSuccesss <- summary(test$diagnosis)[1]/sum(summary(test$diagnosis))  
  
randomClass\_B <- rbinom(10000, 89, probSuccesss)  
randomClass\_M <- rbinom(10000, 143-89, 1-probSuccesss)  
randomClass <- randomClass\_B + randomClass\_M  
randomClass <- as.data.frame(randomClass)  
randomClass$accuracy <- randomClass$randomClass/143  
  
length(filter(randomClass, accuracy >= model\_accuracy)[,1])

## [1] 0

mu <- mean(randomClass$randomClass)  
stdev <- sd(randomClass$randomClass)  
qnorm(0.95, mean = mu, sd = stdev)

## [1] 87.00274