Yvonne Lee: MSDS 664 Week 4

Assignment 4: Using the Pima Indians diabetes data set, I predict the class of diabetes by using ensemble methods.

Some EDA.

```
library(mlbench)
```

```
## Warning: package 'mlbench' was built under R version 4.0.5
```

```
data("PimaIndiansDiabetes2")
```

```
head(PimaIndiansDiabetes2)
```

	pregnant	glucose	pressure	triceps	insulin	mass	pedigree	age	diabetes
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<fct>
1	6	148	72	35	NA	33.6	0.627	50	pos
2	1	85	66	29	NA	26.6	0.351	31	neg
3	8	183	64	NA	NA	23.3	0.672	32	pos
4	1	89	66	23	94	28.1	0.167	21	neg
5	0	137	40	35	168	43.1	2.288	33	pos
6	5	116	74	NA	NA	25.6	0.201	30	neg

6 rows

```
tail(PimaIndiansDiabetes2)
```

	pregnant	glucose	pressure	triceps	insulin	mass	pedigree	age	diabetes
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<fct>
763	9	89	62	NA	NA	22.5	0.142	33	neg
764	10	101	76	48	180	32.9	0.171	63	neg
765	2	122	70	27	NA	36.8	0.340	27	neg
766	5	121	72	23	112	26.2	0.245	30	neg
767	1	126	60	NA	NA	30.1	0.349	47	pos
768	1	93	70	31	NA	30.4	0.315	23	neg

6 rows

```
str(PimaIndiansDiabetes2)
```

```
## 'data.frame':    768 obs. of  9 variables:
##  $ pregnant: num  6 1 8 1 0 5 3 10 2 8 ...
##  $ glucose : num  148 85 183 89 137 116 78 115 197 125 ...
##  $ pressure: num  72 66 64 66 40 74 50 NA 70 96 ...
##  $ triceps : num  35 29 NA 23 35 NA 32 NA 45 NA ...
##  $ insulin : num  NA NA NA 94 168 NA 88 NA 543 NA ...
##  $ mass    : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 NA ...
##  $ pedigree: num  0.627 0.351 0.672 0.167 2.288 ...
##  $ age     : num  50 31 32 21 33 30 26 29 53 54 ...
##  $ diabetes: Factor w/ 2 levels "neg","pos": 2 1 2 1 2 1 2 1 2 2 ...
```

```
PimaIndiansDiabetes2[is.na(PimaIndiansDiabetes2)] <- 0
```

All NAs have been change to 0.

```
str(PimaIndiansDiabetes2)
```

```
## 'data.frame':    768 obs. of  9 variables:
##  $ pregnant: num  6 1 8 1 0 5 3 10 2 8 ...
##  $ glucose : num  148 85 183 89 137 116 78 115 197 125 ...
##  $ pressure: num  72 66 64 66 40 74 50 0 70 96 ...
##  $ triceps : num  35 29 0 23 35 0 32 0 45 0 ...
##  $ insulin : num  0 0 0 94 168 0 88 0 543 0 ...
##  $ mass    : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
##  $ pedigree: num  0.627 0.351 0.672 0.167 2.288 ...
##  $ age     : num  50 31 32 21 33 30 26 29 53 54 ...
##  $ diabetes: Factor w/ 2 levels "neg","pos": 2 1 2 1 2 1 2 1 2 2 ...
```

```
summary(PimaIndiansDiabetes2)
```

```
##     pregnant         glucose         pressure         triceps     
##  Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00  
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00  
##  Median : 3.000   Median :117.0   Median : 72.00   Median :23.00  
##  Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54  
##  3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00  
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00  
##     insulin           mass          pedigree           age        diabetes
##  Min.   :  0.0   Min.   : 0.00   Min.   :0.0780   Min.   :21.00   neg:500 
##  1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437   1st Qu.:24.00   pos:268 
##  Median : 30.5   Median :32.00   Median :0.3725   Median :29.00           
##  Mean   : 79.8   Mean   :31.99   Mean   :0.4719   Mean   :33.24           
##  3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00           
##  Max.   :846.0   Max.   :67.10   Max.   :2.4200   Max.   :81.00           
```

```
library(corrplot)
```

```
## corrplot 0.90 loaded
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
library(RColorBrewer)
```

```
## Warning: package 'RColorBrewer' was built under R version 4.0.3
```

```
library(dplyr)
```

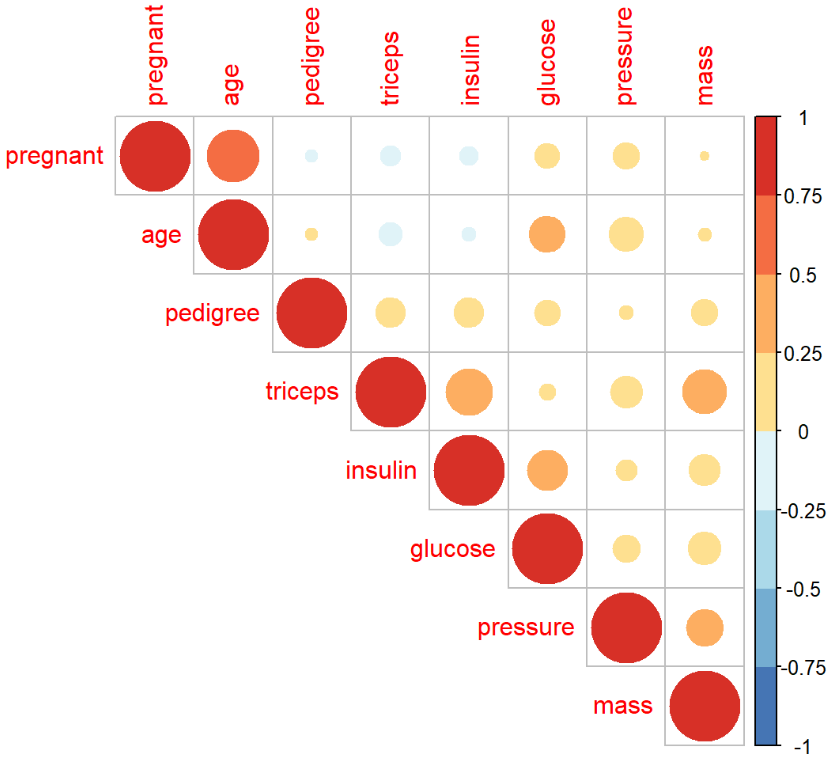
```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
diabetes_predictors = subset(PimaIndiansDiabetes2, select = -c(diabetes))
correlations <-cor(diabetes_predictors, use = "pairwise.complete.obs")
corrplot(correlations, type="upper", order="hclust",
         col=rev(brewer.pal(n=8, name="RdYlBu")))
```



## None of the variables are significantly correlated.

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.5
```

```
## Loading required package: lattice
```

```
## Warning: package 'lattice' was built under R version 4.0.5
```

```
# Splitting the data.
set.seed(123)
trainIndex <- createDataPartition(PimaIndiansDiabetes2$diabetes, p=0.8, list=FALSE)
trainData <- PimaIndiansDiabetes2[trainIndex,]
testData <- PimaIndiansDiabetes2[trainIndex,]
```

Bagging

```
library(ipred)
```

```
## Warning: package 'ipred' was built under R version 4.0.5
```

```
baggedM1 <- bagging(formula=diabetes ~ ., data=PimaIndiansDiabetes2, coob=TRUE)
baggedM1
```

```
## 
## Bagging classification trees with 25 bootstrap replications 
## 
## Call: bagging.data.frame(formula = diabetes ~ ., data = PimaIndiansDiabetes2, 
##     coob = TRUE)
## 
## Out-of-bag estimate of misclassification error:  0.276
```

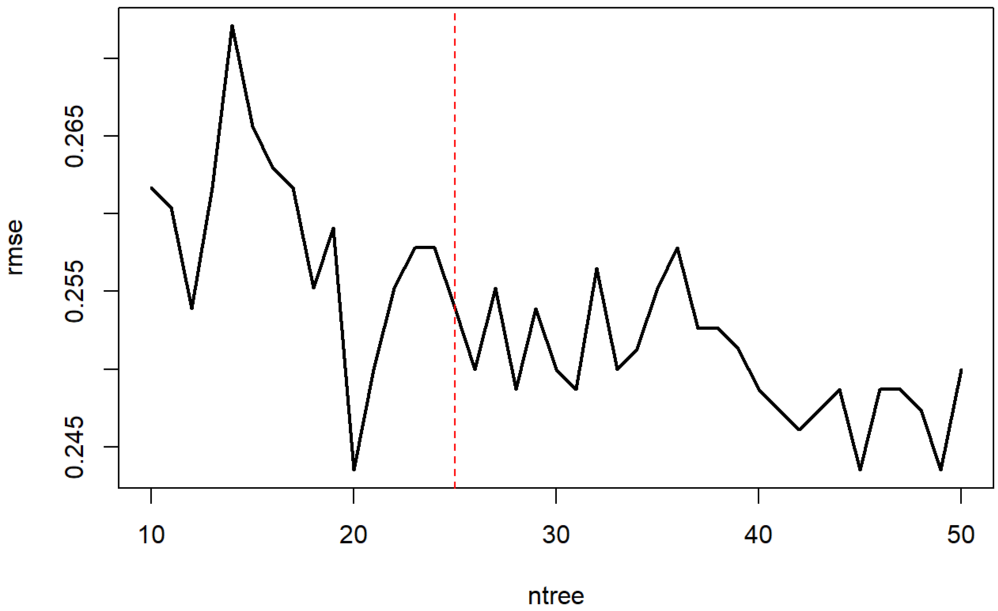
```
# assess 10-50 bagged trees
ntree <- 10:50

# create empty vector to store OOB RMSE values
rmse <- vector(mode = "numeric", length = length(ntree))

for (i in seq_along(ntree)) {
  # reproducibility
  set.seed(123)

  # perform bagged model
  model <- bagging(
  formula = diabetes ~ .,
  data    = PimaIndiansDiabetes2,
  coob    = TRUE,
  nbagg   = ntree[i]
)
  # get OOB error
  rmse[i] <- model$err
}

plot(ntree, rmse, type = 'l', lwd = 2)
abline(v = 25, col = "red", lty = "dashed")
```



### It seems 25 trees is where the error stabilizing.

```
pred <- predict(baggedM1, testData)
result <- confusionMatrix(pred, testData$diabetes)
result
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction neg pos
##        neg 399   4
##        pos   1 211
##                                           
##                Accuracy : 0.9919          
##                  95% CI : (0.9811, 0.9974)
##     No Information Rate : 0.6504          
##     P-Value [Acc > NIR] : <2e-16          
##                                           
##                   Kappa : 0.9821          
##                                           
##  Mcnemar's Test P-Value : 0.3711          
##                                           
##             Sensitivity : 0.9975          
##             Specificity : 0.9814          
##          Pos Pred Value : 0.9901          
##          Neg Pred Value : 0.9953          
##              Prevalence : 0.6504          
##          Detection Rate : 0.6488          
##    Detection Prevalence : 0.6553          
##       Balanced Accuracy : 0.9894          
##                                           
##        'Positive' Class : neg             
## 
```

```
library(plotrix)
```

```
## Warning: package 'plotrix' was built under R version 4.0.5
```

```
pred <- as.integer(pred)
std.error(pred)
```

```
## [1] 0.01918055
```

Confusion matrix shows 99% accuracy and the standard error is 0.02.

```
library(adabag)
```

```
## Warning: package 'adabag' was built under R version 4.0.5
```

```
## Loading required package: rpart
```

```
## Loading required package: foreach
```

```
## Warning: package 'foreach' was built under R version 4.0.5
```

```
## Loading required package: doParallel
```

```
## Warning: package 'doParallel' was built under R version 4.0.5
```

```
## Loading required package: iterators
```

```
## Warning: package 'iterators' was built under R version 4.0.5
```

```
## Loading required package: parallel
```

```
##
## Attaching package: 'adabag'
```

```
## The following object is masked from 'package:ipred':
##
##     bagging
```

```
library(rpart)
pima.baggingcv = bagging.cv(diabetes ~ ., v=10, data=trainData, mfinal=10)
```

```
pima.baggingcv
```

```
## $class
##   [1] "pos" "neg" "neg" "neg" "neg" "neg" "pos" "neg" "neg" "pos" "neg" "pos"
##  [13] "pos" "neg" "neg" "neg" "neg" "neg" "neg" "pos" "neg" "neg" "neg" "neg"
##  [25] "neg" "pos" "pos" "neg" "neg" "pos" "neg" "neg" "pos" "neg" "neg" "pos"
##  [37] "pos" "pos" "pos" "neg" "neg" "neg" "neg" "neg" "pos" "neg" "pos" "pos"
##  [49] "neg" "neg" "neg" "neg" "neg" "pos" "neg" "neg" "neg" "neg" "neg" "neg"
##  [61] "neg" "pos" "neg" "neg" "neg" "neg" "neg" "neg" "pos" "neg" "neg" "neg"
##  [73] "neg" "neg" "neg" "neg" "pos" "neg" "neg" "neg" "neg" "pos" "neg" "pos"
##  [85] "pos" "neg" "neg" "pos" "neg" "neg" "neg" "pos" "neg" "neg" "neg" "neg"
##  [97] "neg" "neg" "neg" "neg" "pos" "neg" "neg" "neg" "neg" "neg" "pos" "neg"
## [109] "neg" "neg" "neg" "pos" "neg" "neg" "neg" "neg" "neg" "pos" "neg" "pos"
## [121] "neg" "neg" "pos" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg"
## [133] "neg" "neg" "neg" "neg" "neg" "neg" "pos" "pos" "neg" "neg" "neg" "neg"
## [145] "neg" "pos" "pos" "pos" "neg" "neg" "pos" "pos" "pos" "neg" "neg" "neg"
## [157] "neg" "neg" "neg" "neg" "neg" "pos" "pos" "pos" "neg" "pos" "pos" "neg"
## [169] "neg" "pos" "neg" "neg" "neg" "pos" "pos" "neg" "neg" "neg" "neg" "pos"
## [181] "neg" "pos" "neg" "neg" "pos" "pos" "pos" "neg" "neg" "neg" "neg" "pos"
## [193] "pos" "pos" "neg" "pos" "neg" "neg" "neg" "neg" "neg" "neg" "pos" "pos"
## [205] "pos" "neg" "pos" "neg" "pos" "pos" "neg" "pos" "neg" "neg" "neg" "neg"
## [217] "neg" "neg" "neg" "neg" "neg" "pos" "neg" "neg" "pos" "neg" "neg" "neg"
## [229] "pos" "neg" "pos" "neg" "neg" "pos" "neg" "neg" "neg" "neg" "neg" "pos"
## [241] "neg" "neg" "neg" "neg" "neg" "neg" "pos" "neg" "neg" "pos" "neg" "neg"
## [253] "neg" "neg" "neg" "neg" "pos" "neg" "pos" "neg" "neg" "neg" "pos" "neg"
## [265] "pos" "pos" "neg" "pos" "neg" "neg" "neg" "pos" "neg" "pos" "neg" "neg"
## [277] "neg" "neg" "neg" "neg" "neg" "neg" "neg" "pos" "neg" "pos" "neg" "pos"
## [289] "pos" "pos" "neg" "pos" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg"
## [301] "neg" "pos" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg"
## [313] "neg" "neg" "pos" "neg" "neg" "pos" "neg" "neg" "neg" "neg" "pos" "neg"
## [325] "neg" "neg" "pos" "pos" "neg" "neg" "neg" "neg" "neg" "pos" "neg" "pos"
## [337] "neg" "neg" "neg" "neg" "neg" "pos" "pos" "neg" "pos" "pos" "neg" "neg"
## [349] "neg" "neg" "neg" "neg" "neg" "pos" "neg" "pos" "pos" "neg" "neg" "pos"
## [361] "neg" "neg" "neg" "neg" "neg" "neg" "neg" "pos" "neg" "neg" "pos" "neg"
## [373] "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "pos" "neg" "neg"
## [385] "neg" "neg" "neg" "neg" "neg" "pos" "neg" "neg" "neg" "pos" "neg" "neg"
## [397] "pos" "neg" "pos" "neg" "neg" "pos" "neg" "pos" "neg" "neg" "neg" "neg"
## [409] "neg" "neg" "neg" "neg" "pos" "neg" "neg" "neg" "neg" "neg" "neg" "neg"
## [421] "pos" "pos" "neg" "neg" "neg" "pos" "neg" "neg" "neg" "neg" "neg" "neg"
## [433] "neg" "neg" "neg" "neg" "neg" "pos" "neg" "neg" "neg" "neg" "neg" "neg"
## [445] "pos" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "pos" "neg"
## [457] "neg" "neg" "neg" "neg" "neg" "pos" "neg" "neg" "neg" "neg" "neg" "neg"
## [469] "neg" "pos" "neg" "neg" "neg" "neg" "pos" "neg" "pos" "pos" "neg" "neg"
## [481] "neg" "neg" "pos" "neg" "neg" "pos" "neg" "neg" "neg" "neg" "pos" "neg"
## [493] "pos" "neg" "neg" "neg" "pos" "pos" "neg" "neg" "neg" "neg" "neg" "neg"
## [505] "pos" "neg" "neg" "neg" "pos" "neg" "neg" "neg" "neg" "neg" "neg" "neg"
## [517] "neg" "neg" "neg" "neg" "neg" "pos" "pos" "neg" "neg" "neg" "neg" "neg"
## [529] "neg" "pos" "neg" "pos" "pos" "neg" "neg" "neg" "neg" "neg" "pos" "pos"
## [541] "neg" "neg" "neg" "pos" "neg" "neg" "neg" "neg" "pos" "neg" "neg" "neg"
## [553] "neg" "neg" "neg" "neg" "pos" "neg" "neg" "neg" "pos" "neg" "neg" "neg"
## [565] "neg" "pos" "neg" "neg" "neg" "pos" "neg" "pos" "pos" "neg" "pos" "pos"
## [577] "neg" "neg" "neg" "neg" "pos" "pos" "neg" "neg" "neg" "neg" "neg" "neg"
## [589] "neg" "pos" "neg" "neg" "neg" "neg" "pos" "neg" "neg" "pos" "pos" "pos"
## [601] "neg" "pos" "neg" "neg" "neg" "pos" "pos" "pos" "pos" "neg" "neg" "pos"
## [613] "neg" "neg" "neg"
## 
## $confusion
##                Observed Class
## Predicted Class neg pos
##             neg 348 106
##             pos  52 109
## 
## $error
## [1] 0.2569106
```

The 10-fold cross validation method shows a larger standard error of 0.25.

Boosting

```
library(gbm)
```

```
## Warning: package 'gbm' was built under R version 4.0.5
```

```
## Loaded gbm 2.1.8
```

```
gbm.fit <- gbm(
  formula = diabetes ~ .,
  distribution = "gaussian",
  data = PimaIndiansDiabetes2,
  n.trees = 10000,
  interaction.depth = 1,
  shrinkage = 0.001,
  cv.folds = 5,
  n.cores = NULL,
  verbose = FALSE
  )
gbm
```

```
## function (formula = formula(data), distribution = "bernoulli",
##     data = list(), weights, var.monotone = NULL, n.trees = 100,
##     interaction.depth = 1, n.minobsinnode = 10, shrinkage = 0.1,
##     bag.fraction = 0.5, train.fraction = 1, cv.folds = 0, keep.data = TRUE,
##     verbose = FALSE, class.stratify.cv = NULL, n.cores = NULL)
## {
##     mcall <- match.call()
##     lVerbose <- if (!is.logical(verbose)) {
##         FALSE
##     }
##     else {
##         verbose
##     }
##     mf <- match.call(expand.dots = FALSE)
##     m <- match(c("formula", "data", "weights", "offset"), names(mf),
##         0)
##     mf <- mf[c(1, m)]
##     mf$drop.unused.levels <- TRUE
##     mf$na.action <- na.pass
##     mf[[1]] <- as.name("model.frame")
##     m <- mf
##     mf <- eval(mf, parent.frame())
##     Terms <- attr(mf, "terms")
##     w <- model.weights(mf)
##     offset <- model.offset(mf)
##     y <- model.response(mf)
##     if (missing(distribution)) {
##         distribution <- guessDist(y)
##     }
##     if (is.character(distribution)) {
##         distribution <- list(name = distribution)
##     }
##     if (!is.element(distribution$name, getAvailableDistributions())) {
##         stop("Distribution ", distribution$name, " is not supported.")
##     }
##     if (distribution$name == "multinomial") {
##         warning("Setting `distribution = \"multinomial\"` is ill-advised as it is ",
##             "currently broken. It exists only for backwards compatibility. ",
##             "Use at your own risk.", call. = FALSE)
##     }
##     var.names <- attributes(Terms)$term.labels
##     x <- model.frame(terms(reformulate(var.names)), data = data,
##         na.action = na.pass)
##     response.name <- as.character(formula[[2L]])
##     class.stratify.cv <- getStratify(class.stratify.cv, d = distribution)
##     group <- NULL
##     num.groups <- 0
##     if (distribution$name != "pairwise") {
##         nTrain <- floor(train.fraction * nrow(x))
##     }
##     else {
##         distribution.group <- distribution[["group"]]
##         if (is.null(distribution.group)) {
##             stop(paste("For pairwise regression, `distribution` must be a list of",
##                 "the form `list(name = \"pairwise\", group = c(\"date\",",
##                 "\"session\", \"category\", \"keywords\"))`."))
##         }
##         i <- match(distribution.group, colnames(data))
##         if (any(is.na(i))) {
##             stop("Group column does not occur in data: ", distribution.group[is.na(i)],
##                 ".")
##         }
##         group <- factor(do.call(paste, c(data[, distribution.group,
##             drop = FALSE], sep = ":")))
##         if ((!missing(weights)) && (!is.null(weights))) {
##             w.min <- tapply(w, INDEX = group, FUN = min)
##             w.max <- tapply(w, INDEX = group, FUN = max)
##             if (any(w.min != w.max)) {
```

```
##                  stop("For `distribution = \"pairwise\"`, all instances for the same ",
##                    "group must have the same weight.")
##              }
##              w <- w * length(w.min)/sum(w.min)
##          }
##          perm.levels <- levels(group)[sample(1:nlevels(group))]
##          group <- factor(group, levels = perm.levels)
##          ord.group <- order(group, -y)
##          group <- group[ord.group]
##          y <- y[ord.group]
##          x <- x[ord.group, , drop = FALSE]
##          w <- w[ord.group]
##          num.groups.train <- max(1, round(train.fraction * nlevels(group)))
##          nTrain <- max(which(group == levels(group)[num.groups.train]))
##          Misc <- group
##      }
##      cv.error <- NULL
##      if (cv.folds == 1) {
##          cv.folds <- 0
##      }
##      if (cv.folds > 1) {
##          cv.results <- gbmCrossVal(cv.folds = cv.folds, nTrain = nTrain,
##              n.cores = n.cores, class.stratify.cv = class.stratify.cv,
##              data = data, x = x, y = y, offset = offset, distribution = distribution,
##              w = w, var.monotone = var.monotone, n.trees = n.trees,
##              interaction.depth = interaction.depth, n.minobsinnode = n.minobsinnode,
##              shrinkage = shrinkage, bag.fraction = bag.fraction,
##              var.names = var.names, response.name = response.name,
##              group = group)
##          cv.error <- cv.results$error
##          p <- cv.results$predictions
##      }
##      gbm.obj <- gbm.fit(x = x, y = y, offset = offset, distribution = distribution,
##          w = w, var.monotone = var.monotone, n.trees = n.trees,
##          interaction.depth = interaction.depth, n.minobsinnode = n.minobsinnode,
##          shrinkage = shrinkage, bag.fraction = bag.fraction, nTrain = nTrain,
##          keep.data = keep.data, verbose = lVerbose, var.names = var.names,
##          response.name = response.name, group = group)
##      gbm.obj$train.fraction <- train.fraction
##      gbm.obj$Terms <- Terms
##      gbm.obj$cv.error <- cv.error
##      gbm.obj$cv.folds <- cv.folds
##      gbm.obj$call <- mcall
##      gbm.obj$m <- m
##      if (cv.folds > 1) {
##          gbm.obj$cv.fitted <- p
##      }
##      if (distribution$name == "pairwise") {
##          gbm.obj$ord.group <- ord.group
##          gbm.obj$fit <- gbm.obj$fit[order(ord.group)]
##      }
##      gbm.obj
## }
## <bytecode: 0x000000002d453e80>
## <environment: namespace:gbm>
```

```
train.gbm <- train(as.factor(diabetes) ~ .,
                   data=PimaIndiansDiabetes2,
                   method="gbm",
                   verbose=F)
train.gbm
```

```
## Stochastic Gradient Boosting
##
## 768 samples
##   8 predictor
##   2 classes: 'neg', 'pos'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 768, 768, 768, 768, 768, 768, ...
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy   Kappa
##   1                   50      0.7669688  0.4553161
##   1                  100      0.7654558  0.4578834
##   1                  150      0.7600540  0.4492392
##   2                   50      0.7610085  0.4484794
##   2                  100      0.7562524  0.4428651
##   2                  150      0.7522818  0.4371831
##   3                   50      0.7619646  0.4565980
##   3                  100      0.7528556  0.4384531
##   3                  150      0.7485702  0.4315635
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were n.trees = 50, interaction.depth =
##  1, shrinkage = 0.1 and n.minobsinnode = 10.
```

```
gbm.classTest <-  predict(train.gbm,
                         newdata = testData,
                          type="raw")
head(gbm.classTest)
```

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

```
confusionMatrix(testData$diabetes, gbm.classTest)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction neg pos
##        neg 361  39
##        pos  99 116
##                                           
##                Accuracy : 0.7756          
##                  95% CI : (0.7405, 0.808) 
##     No Information Rate : 0.748           
##     P-Value [Acc > NIR] : 0.06135         
##                                           
##                   Kappa : 0.4725          
##                                           
##  Mcnemar's Test P-Value : 5.103e-07       
##                                           
##             Sensitivity : 0.7848          
##             Specificity : 0.7484          
##          Pos Pred Value : 0.9025          
##          Neg Pred Value : 0.5395          
##              Prevalence : 0.7480          
##          Detection Rate : 0.5870          
##    Detection Prevalence : 0.6504          
##       Balanced Accuracy : 0.7666          
##                                           
##        'Positive' Class : neg             
## 
```

```
gbm.classTest <- as.integer(gbm.classTest)
std.error(gbm.classTest)
```

```
## [1] 0.01752207
```

For this gradient boosting method, the accuracy is 79% and standard error is 0.018.

```
pima.boostingcv = boosting.cv(diabetes ~ ., v=10, data=trainData, mfinal=10)
```

```
## i:  1 Mon Nov 15 21:16:14 2021 
## i:  2 Mon Nov 15 21:16:14 2021 
## i:  3 Mon Nov 15 21:16:14 2021 
## i:  4 Mon Nov 15 21:16:15 2021 
## i:  5 Mon Nov 15 21:16:15 2021 
## i:  6 Mon Nov 15 21:16:16 2021 
## i:  7 Mon Nov 15 21:16:16 2021 
## i:  8 Mon Nov 15 21:16:17 2021 
## i:  9 Mon Nov 15 21:16:17 2021 
## i:  10 Mon Nov 15 21:16:18 2021
```

```
pima.boostingcv
```

```
## $class
##   [1] "pos" "pos" "neg" "neg" "neg" "pos" "pos" "neg" "neg" "pos" "neg" "pos"
##  [13] "neg" "neg" "pos" "pos" "neg" "neg" "neg" "pos" "neg" "pos" "neg" "neg"
##  [25] "neg" "pos" "pos" "neg" "pos" "pos" "neg" "neg" "pos" "pos" "neg" "pos"
##  [37] "pos" "pos" "neg" "neg" "pos" "neg" "neg" "neg" "pos" "neg" "neg" "pos"
##  [49] "neg" "neg" "pos" "neg" "neg" "neg" "neg" "pos" "neg" "pos" "neg" "neg"
##  [61] "neg" "pos" "neg" "neg" "neg" "neg" "pos" "neg" "pos" "neg" "pos" "neg"
##  [73] "neg" "neg" "neg" "pos" "pos" "neg" "neg" "neg" "neg" "pos" "neg" "pos"
##  [85] "neg" "neg" "neg" "pos" "neg" "neg" "neg" "pos" "neg" "neg" "neg" "neg"
##  [97] "neg" "neg" "neg" "neg" "pos" "pos" "neg" "pos" "neg" "neg" "pos" "neg"
## [109] "neg" "neg" "neg" "pos" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "pos"
## [121] "neg" "neg" "pos" "pos" "neg" "neg" "neg" "pos" "neg" "pos" "pos" "neg"
## [133] "neg" "neg" "neg" "neg" "neg" "pos" "pos" "pos" "neg" "neg" "neg" "neg"
## [145] "neg" "pos" "pos" "neg" "pos" "neg" "pos" "pos" "neg" "neg" "neg" "neg"
## [157] "neg" "neg" "neg" "neg" "pos" "pos" "pos" "pos" "neg" "pos" "pos" "pos"
## [169] "neg" "pos" "pos" "neg" "neg" "pos" "pos" "neg" "neg" "neg" "neg" "pos"
## [181] "neg" "neg" "pos" "neg" "pos" "pos" "pos" "neg" "neg" "pos" "pos" "pos"
## [193] "pos" "neg" "neg" "pos" "neg" "neg" "neg" "neg" "pos" "neg" "neg" "pos"
## [205] "pos" "neg" "pos" "pos" "pos" "neg" "neg" "neg" "neg" "neg" "pos" "neg"
## [217] "neg" "neg" "neg" "pos" "neg" "pos" "neg" "neg" "pos" "neg" "neg" "neg"
## [229] "pos" "neg" "pos" "neg" "neg" "pos" "neg" "neg" "neg" "neg" "neg" "pos"
## [241] "neg" "neg" "pos" "neg" "neg" "neg" "pos" "neg" "neg" "pos" "pos" "neg"
## [253] "pos" "neg" "neg" "neg" "neg" "pos" "pos" "neg" "pos" "neg" "pos" "neg"
## [265] "pos" "pos" "pos" "pos" "neg" "neg" "neg" "pos" "neg" "neg" "neg" "neg"
## [277] "neg" "neg" "neg" "pos" "neg" "neg" "neg" "pos" "pos" "pos" "neg" "pos"
## [289] "pos" "pos" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "pos" "neg" "neg"
## [301] "pos" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "pos" "pos" "neg" "neg"
## [313] "neg" "neg" "pos" "neg" "pos" "pos" "neg" "pos" "neg" "neg" "pos" "pos"
## [325] "neg" "neg" "pos" "pos" "pos" "neg" "neg" "neg" "neg" "neg" "neg" "pos"
## [337] "neg" "neg" "neg" "neg" "neg" "pos" "pos" "neg" "pos" "neg" "neg" "neg"
## [349] "neg" "neg" "neg" "neg" "pos" "pos" "neg" "neg" "pos" "neg" "neg" "pos"
## [361] "neg" "neg" "neg" "neg" "neg" "neg" "neg" "pos" "pos" "neg" "pos" "neg"
## [373] "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "pos" "pos" "pos"
## [385] "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "pos" "neg" "neg"
## [397] "pos" "neg" "pos" "neg" "neg" "pos" "neg" "pos" "neg" "pos" "neg" "neg"
## [409] "neg" "pos" "neg" "neg" "pos" "neg" "neg" "neg" "neg" "neg" "neg" "neg"
## [421] "pos" "pos" "neg" "neg" "neg" "pos" "neg" "neg" "neg" "neg" "neg" "neg"
## [433] "neg" "neg" "pos" "pos" "pos" "neg" "neg" "neg" "neg" "neg" "neg" "pos"
## [445] "pos" "neg" "neg" "neg" "neg" "neg" "pos" "neg" "pos" "neg" "pos" "neg"
## [457] "neg" "neg" "neg" "neg" "neg" "pos" "pos" "neg" "neg" "neg" "neg" "pos"
## [469] "pos" "pos" "neg" "neg" "neg" "neg" "neg" "neg" "pos" "pos" "neg" "neg"
## [481] "neg" "pos" "pos" "neg" "neg" "pos" "neg" "neg" "neg" "neg" "pos" "neg"
## [493] "pos" "neg" "neg" "neg" "pos" "pos" "pos" "neg" "neg" "neg" "neg" "neg"
## [505] "pos" "neg" "neg" "neg" "pos" "neg" "neg" "neg" "neg" "neg" "neg" "neg"
## [517] "neg" "neg" "neg" "neg" "neg" "neg" "pos" "neg" "neg" "neg" "neg" "neg"
## [529] "neg" "pos" "pos" "pos" "pos" "neg" "neg" "neg" "neg" "neg" "pos" "pos"
## [541] "neg" "neg" "neg" "pos" "pos" "neg" "neg" "neg" "pos" "neg" "neg" "neg"
## [553] "neg" "neg" "pos" "neg" "pos" "neg" "neg" "pos" "pos" "neg" "pos" "neg"
## [565] "neg" "pos" "neg" "neg" "neg" "pos" "neg" "pos" "pos" "neg" "pos" "pos"
## [577] "neg" "neg" "neg" "neg" "neg" "pos" "neg" "neg" "neg" "neg" "neg" "neg"
## [589] "neg" "pos" "neg" "pos" "neg" "neg" "neg" "neg" "neg" "pos" "pos" "neg"
## [601] "neg" "pos" "neg" "neg" "neg" "pos" "neg" "neg" "pos" "neg" "neg" "neg"
## [613] "neg" "neg" "pos"
## 
## $confusion
##                Observed Class
## Predicted Class neg pos
##             neg 315  97
##             pos  85 118
## 
## $error
## [1] 0.295935
```

The standard error for the 10-fold cross validation for this boosting method is 0.27 which is higher than the gradient boosting standard error.

It seems that doing a 10-fold cross validation increases the standard of error for both the bagging and boosting methods.

The bagging method had an accuracy of 99% which is extremely high compared to the boosting method with an accuracy of 79%. Both methods have pros and cons and choosing which one to use depends on the need of the user. Both are great for reducing variance and providing higher stability. However, Bagging is great for combining predictions that belong to the same type while boosting is for combining predictions of different types. Bagging aims to decrease variance while boosting aims to decrease bias.

References

Brownlee, J. (2020, August 15). How to estimate model accuracy in R using the Caret package. Machine Learning Mastery. Retrieved November 16, 2021, from https://machinelearningmastery.com/how-to-estimate-model-accuracy-in-r-using-the-caret-package/ (https://machinelearningmastery.com/how-to-estimate-model-accuracy-in-r-using-the-caret-package/).

Ensemble Methods - Pennsylvania State University. (n.d.). Retrieved November 16, 2021, from https://quantdev.ssri.psu.edu/sites/qdev/files/09_EnsembleMethods_2017_1127.html (https://quantdev.ssri.psu.edu/sites/qdev/files/09_EnsembleMethods_2017_1127.html).

Gradient Boosting Machines. Gradient Boosting Machines · UC Business Analytics R Programming Guide. (n.d.). Retrieved November 16, 2021, from http://uc-r.github.io/gbm_regression (http://uc-r.github.io/gbm_regression).

Regression trees. Regression Trees · UC Business Analytics R Programming Guide. (n.d.). Retrieved November 16, 2021, from http://uc-r.github.io/regression_trees#bag (http://uc-r.github.io/regression_trees#bag).