```
library(rpart)
library(ROCR)



# function to build a partition tree

fit.build = function(formula, data) {

  rpart(formula = formula, data = data, method = 'class')
}



# function to build a partition tree with Xtreme parameter values

fit.buildX = function(formula, data) {

  rpart(formula = formula, data = data, method = 'class', cp = -0, minsplit = 1, minbucket = 1, maxdepth =
30)
}



# function to build a partition tree with Xtreme parameter values

fit.buildP = function(formula, data, cp = 0.01, minsplit = 20, minbucket = round(minsplit/3), maxdepth = 30)
{

  rpart(formula = formula, data = data, method = 'class', cp = cp, minsplit = minsplit, minbucket =
minbucket, maxdepth = maxdepth)
}



# function to prune the partition tree

fit.prune = function(fit, cp) {

  prune(fit, cp = cp)
}



# function to plot a partition tree with labels

fit.plot = function(fit) {

  plot(fit, margin = 0.01, uniform = T)
  text(fit, use.n = T, xpd = T, minlength = 6)
}



# function to plot and print complexity parameter information

fit.plotcp = function(fit) {

  plotcp(fit)
  printcp(fit)
}



# function that returns the complexity parameter that corresponds to the min cross validation error

fit.mincvcp = function(fit) {

  fit$cptable[which.min(fit$cptable[, "xerror"]), "CP"]
}
```

```
# function to make predition, and to make human readable

fit.predict = function(fit, newdata) {

  predict(fit, newdata = newdata)
}



# function to print the performance of the prediction

fit.performance = function(fit, pred, val, pred.labels) {

  outcome = rep(pred.labels[1], nrow(pred))
  outcome[pred[, 1] > 0.5] = pred.labels[2]

  prop.table(xtabs(~ val$result + outcome), 1)
}



# function to plot the analysis of the fit

fit.analysis = function(fit, pred, val, pred.labels) {

  p.scores = prediction(pred[,1], val$result, label.ordering = pred.labels)
  p.perf   = performance(p.scores, "tpr", "fpr")
  p.auc    = performance(p.scores, "auc")

  plot(p.perf, col = "blue", lwd = 2, xlab = '1-Specificity (FPR)', ylab = 'Sensitivity (TPR)')
  abline(a = 0, b = 1, col = 'red', lty = 3, lwd = 2)
  text(0.1, 1, round(p.auc@y.values[[1]], 3), col = 'red')
}
```