Try this for class predictions:

pred <- predict(model\_ctree, newdata=test)

library(caret)

confusionMatrix(pred, test$response)

Try this for class probabilities:

probs <- treeresponse(model\_ctree, newdata=test)

pred <- do.call(rbind, pred)

summary(pred)

Try this for a roc curve:

library(ROCR)

roc\_pred <- prediction(pred[,1], test$response)

plot(performance(roc\_pred, measure="tpr", x.measure="fpr"), colorize=TRUE)

Try this for a lift curve:

plot(performance(roc\_pred, measure="lift", x.measure="rpp"), colorize=TRUE)

Sensitivity/specificity curve and precision/recall curve:

plot(performance(roc\_pred, measure="sens", x.measure="spec"), colorize=TRUE)

plot(performance(roc\_pred, measure="prec", x.measure="rec"), colorize=TRUE)

More info:

?ctree

?confusionMatrix

?performance

Also, you should check out the caret package if you're building predictive models in R. It implements a number of out-of-sample evaluation schemes, including bootstrap sampling, cross-validation, and multiple train/test splits. caret is really nice because it provides a unified interface to all the models, so you don't have to remember, e.g., that treeresponse is the function to get class probabilities from a ctree model. Here's an example of using 10-fold cross-validation to evaluation your model, which is much better than a single train/test split:

model <- train(response ~ x1 + .. xn , data = train, method='ctree', tuneLength=10,

trControl=trainControl(

method='cv', number=10, classProbs=TRUE, summaryFunction=twoClassSummary))

model

plot(model)