```r
tree.app <- function(Xapp, zapp, Xtst) {
    zapp = factor(zapp)
    ctrl = rpart.control(cp=0.0001)
    tree = rpart(zapp~., data = Xapp, control = ctrl)
    treeOptimal <- prune(tree,cp=tree$cptable[which.min(tree$cptable[,4]),1])
    pred = predict(treeOptimal, Xtst, type = "class")

    # for tree.partition afin de voir les frontières de décision
    tree_package_tree = tree(zapp~., Xapp, control = tree.control(nobs=dim(Xapp)[1],mindev=0.0001))

    out <- NULL
    out$pred <- factor(pred)
    out$fullTree = tree
    out$optimalTree = treeOptimal
    out$treeForDecisionBorder = tree_package_tree
    return(out)
}


random.forest.app <- function(Xapp, zapp, Xtst, nbTrees = 1000) {
    out <- NULL
    random_forest <- randomForest(as.factor(sample$zapp) ~ .,
                          data=sample$Xapp,
                          importance=TRUE,
                          ntree=nbTrees)
    out$random_forest = random_forest
    out$pred <- predict(random_forest, sample$Xtst)
    return(out)
}
```