```r
##Test dataset
#Load packages
library(randomForest)
library(mlbench)
library(caret)
#Load dataset
data("Sonar")
dataset=Sonar
x=dataset[,1:60]
y=dataset[,61]
##Test Algorithms
#Create model with default parameters
trainControl=trainControl(method="repeatedcv", number=10, repeats=3)
seed=7
metric="Accuracy"
set.seed(seed)
mtry=sqrt(ncol(x))
tunegrid=expand.grid(.mtry=mtry)
rfDefault=train(Class~., data=dataset, method="rf", metric=metric,
tuneGrid=tunegrid, trControl=trainControl)
print(rfDefault)

##Tune Using Caret
#Random search
trainControl=trainControl(method="repeatedcv", number=10, repeats=3,
search="random")
set.seed(seed)
mtry=sqrt(ncol(x))
rfRandom=train(Class~., data=dataset, method="rf",metric=metric,
tuneLength=15, trControl=trainControl)
print(rfRandom)
plot(rfRandom)
#Grid Search
trainControl=trainControl(method="repeatedcv", number=10, repeats=3,
search="grid")
set.seed(seed)
tunegrid=expand.grid(.mtry=c(1:15))
rfGrid=train(Class~., data=dataset, method="rf", metric=metric,
tuneGrid=tunegrid, trControl=trainControl)
print(rfGrid)
plot(rfGrid)

##Tune using algorithm tools
#Algorithm Tune (tuneRF)
set.seed(seed)
bestmtry=tuneRF(x,y, stepFactor = 1.5, improve = 1e-5, ntree=500)
print(bestmtry)
##Craft your own parameter search
#Manual search
trainControl=trainControl(method="repeatecv", number=10, repeats=3,
search="grid")
tunegrid=expand.grid(.mtry=c(sqrt(ncol(x))))
modellist=list()
for (ntree in c(1000, 1500, 2000,2500)){
  set.seed(seed)
  fit=train(Class~., data=dataset, method="rf", metric=metric,
tuneGrid=tunegrid, tControl=trainControl, ntree=ntree)
  key=toString(ntree)
  modellist[[key]]=fit
}
#Compare results
results=resample(modellist)
summary(results)
dotplot(results)
```