```
##Combine model predictions into ensemble predictions
##Test dataset
#Load packages
library(mlbench)
library(caret)
library(caretEnsemble)
#Load the dataset
data("Ionosphere")
dataset=Ionosphere
dataset=dataset[,-2]
dataset$V1=as.numeric(as.character(dataset$V1))
##Boosting algorithms
#Boosting algorithms
trainControl=trainControl(method="repeatedcv", number=10, repeats=3)
seed=7
metric="Accuracy"
#C5.0
library(C50)
set.seed(seed)
fit.c50=train(Class~., data=dataset, method="C5.0", metric=metric,
trControl=trainControl)
#Stochastic Gradient Boosting
set.seed(seed)
fit.gbm=train(Class~., data=dataset, method="C5.0", metric=metric,
trControl=trainControl,verbose=FALSE)
#Summarize results
boostingResults=resamples(list(c5.0=fit.c50, gbm=fit.gbm))
summary(boostingResults)
dotplot(boostingResults)

##Test dataset
#Load packages
library(mlbench)
library(caret)
library(caretEnsemble)
#Load the dataset
data("Ionosphere")
dataset=Ionosphere
dataset=dataset[,-2]
dataset$V1=as.numeric(as.character(dataset$V1))
##Bagging algorithms
#Example of bagging algorithms
trainControl=trainControl(method="repeatedcv", number=10, repeats=3)
seed=7
metric="Accuracy"
#Bagged CART
set.seed(seed)
fit.treebag=train(Class~., data=dataset,
method="treebag",metric=metric, trControl=trainControl)
#Random forest
```

```
set.seed(seed)
fit.rf=train(Class~., data=dataset, method="rf",metric=metric,
trControl=trainControl)
#Summarize results
baggingResults=resamples(list(treebag=fit.treebag, rf=fit.rf))
summary(baggingResults)
dotplot(baggingResults)

##Combine model predictions into ensemble predictions
##Test dataset
#Load packages
library(mlbench)
library(caret)
library(caretEnsemble)
library(MASS)
library(rpart)
library(kernlab)
#Load the dataset
data("Ionosphere")
dataset=Ionosphere
dataset=dataset[,-2]
dataset$V1=as.numeric(as.character(dataset$V1))
##Stacking algorithms
#Create submodls:LDA, CART, Logistic Regression, KNN, SVM
trainControl=trainControl(method="repeatedcv", number=10, repeats=3,
savePredictions=TRUE, classProb=TRUE)
algorithmList=c('lda','rpart','glm','knn','svmRadial')
set.seed(seed)
models=caretList(Class~., data=dataset,
trControl=trainControl,methodList = algorithmList)
results=resamples(models)
summary(results)
dotplot(results)
#Corelation between results
modelCor(results)
splom(results)
#Combine the predictions of the classifiers using a simple linear
model
stackControl=trainControl(method="repeatedcv", number=10, repeats=3,
savePredictions=TRUE, classProb=TRUE)
set.seed(seed)
stack.glm=caretStack(models, method="glm", metric="Accuracy",
trControl=stackControl)
print(stack.glm)
#Combine the predictions of the classifiers using a random forest
set.seed(seed)
stack.rf=caretStack(models, method="rf",
metric="Accuracy",trControl=stackControl)
print(stack.rf)
```