GRADIENT BOOSTING MACHINE(GBM)

```
library(h2o)

h2o.init(nthreads = -1)

train.hex <- h2o.importFile("C:/datasets/multidata.csv")

head(train.hex)

c <- as.factor(train.hex[,])

splits <- h2o.splitFrame(c, 0.8, seed=1234)

train_file <- splits[[1]]

test_file <- splits[[2]]

############## TARGET Y1 ############################

dl1 <- h2o.gbm(x=1:95, y="Y1",
training_frame=train_file,validation_frame=test_file,balance_classes=TRUE,se
ed = 1234)

dl1

print(h2o.logloss(dl1, valid = TRUE))

hyper_params <- list( balance_classes = c(TRUE, FALSE) )

grid <- h2o.grid(x =1:95, y = "Y1", training_frame = train_file,

            validation_frame = test_file, algorithm = "gbm", grid_id =
"covtype_grid", hyper_params = hyper_params,search_criteria = list(strategy =
"Cartesian"), seed = 1234)

sortedGrid <- h2o.getGrid("covtype_grid", sort_by = "logloss", decreasing =
FALSE)

sortedGrid


############## TARGET Y2 ############################


dl2 <- h2o.gbm(x=1:95, y="Y2",
training_frame=train_file,validation_frame=test_file,balance_classes=TRUE,se
ed = 1234)

dl2
```

```
print(h2o.logloss(dl2, valid = TRUE))

hyper_params <- list( balance_classes = c(TRUE, FALSE) )

grid <- h2o.grid(x =1:95 , y = "Y2", training_frame = train_file,

        validation_frame = test_file,

        algorithm = "gbm", grid_id = "covtype_grid", hyper_params =
hyper_params,search_criteria = list(strategy = "Cartesian"), seed =1234)

sortedGrid <- h2o.getGrid("covtype_grid", sort_by = "logloss", decreasing =
FALSE)

sortedGrid

############## TARGET Y3 ############################


dl3 <- h2o.gbm(x=1:95, y="Y3",
training_frame=train_file,validation_frame=test_file,balance_classes=TRUE,se
ed = 1234)

dl3

print(h2o.logloss(dl3, valid = TRUE))

hyper_params <- list( balance_classes = c(TRUE, FALSE) )

grid <- h2o.grid(x =1:95 , y = "Y3", training_frame = train_file,

        validation_frame = test_file,

        algorithm = "gbm", grid_id = "covtype_grid", hyper_params =
hyper_params,search_criteria = list(strategy = "Cartesian"), seed =1234)

sortedGrid <- h2o.getGrid("covtype_grid", sort_by = "logloss", decreasing =
FALSE)

sortedGrid


############## TARGET Y4 ############################

dl4 <- h2o.gbm(x=1:95, y="Y4",
training_frame=train_file,validation_frame=test_file,balance_classes=TRUE,se
ed = 1234)

dl4
```

```
print(h2o.logloss(dl4, valid = TRUE))

hyper_params <- list( balance_classes = c(TRUE, FALSE) )

grid <- h2o.grid(x =1:95 , y = "Y4", training_frame = train_file,

        validation_frame = test_file,

        algorithm = "gbm", grid_id = "covtype_grid", hyper_params =
hyper_params,search_criteria = list(strategy = "Cartesian"), seed =1234)

sortedGrid <- h2o.getGrid("covtype_grid", sort_by = "logloss", decreasing =
FALSE)

sortedGrid
```

RANDOM FOREST(RF)

```
library(h2o)

library(caret)

h2o.init(nthreads = -1)

train.hex <- h2o.importFile("C:/datasets/multidata.csv")

head(train.hex)

c <- as.factor(train.hex[,])

splits <- h2o.splitFrame(c, 0.8, seed=1234)

train_file <- splits[[1]]

test_file <- splits[[2]]

wtrain=as.h2o(train_file)

wtest=as.h2o(test_file)

############## TARGET Y1 ############################

#Balanced + Not stratified

dl1 <- h2o.randomForest(x=1:95, y="Y1",
training_frame=train_file,validation_frame=test_file,balance_classes=TRUE,se
ed = 1234)

dl1

#Balanced + stratified
```

```
dl2 <- h2o.randomForest(x=1:95, y="Y1",
training_frame=train_file,validation_frame=test_file,nfolds=10,fold_assignment
= "Stratified",balance_classes=TRUE,seed = 1234)

dl2

#UnBalanced + stratified

dl3 <- h2o.randomForest(x=1:95, y="Y1",
training_frame=train_file,validation_frame=test_file,nfolds=10,fold_assignment
= "Stratified",balance_classes=FALSE,seed = 1234)

dl3

#Unbalanced + Not stratified

dl4 <- h2o.randomForest(x=1:95, y="Y1",
training_frame=train_file,validation_frame=test_file,balance_classes=FALSE,s
eed = 1234)

dl4

v1 <- h2o.performance(dl1,wtest)

v1

v2 <- h2o.performance(dl2,wtest)

v2

v3 <- h2o.performance(dl3,wtest)

v3

v4 <- h2o.performance(dl4,wtest)

v4

############### TARGET Y2 ##############################

#Balanced + Not stratified

dl1 <- h2o.randomForest(x=1:95, y="Y2",
training_frame=train_file,validation_frame=test_file,balance_classes=TRUE,se
ed = 1234)

dl1

#Balanced + stratified
```

```
dl2 <- h2o.randomForest(x=1:95, y="Y2",
training_frame=train_file,validation_frame=test_file,nfolds=10,fold_assignment
= "Stratified",balance_classes=TRUE,seed = 1234)

dl2

#UnBalanced + stratified

dl3 <- h2o.randomForest(x=1:95, y="Y2",
training_frame=train_file,validation_frame=test_file,nfolds=10,fold_assignment
= "Stratified",balance_classes=FALSE,seed = 1234)

dl3

#Unbalanced + Not stratified

dl4 <- h2o.randomForest(x=1:95, y="Y2",
training_frame=train_file,validation_frame=test_file,balance_classes=FALSE,s
eed = 1234)

dl4

v1 <- h2o.performance(dl1,wtest)

v1

v2 <- h2o.performance(dl2,wtest)

v2

v3 <- h2o.performance(dl3,wtest)

v3

v4 <- h2o.performance(dl4,wtest)

v4

############### TARGET Y3 ##############################

#Balanced + Not stratified

dl1 <- h2o.randomForest(x=1:95, y="Y3",
training_frame=train_file,validation_frame=test_file,balance_classes=TRUE,se
ed = 1234)

dl1

#Balanced + stratified
```

```
dl2 <- h2o.randomForest(x=1:95, y="Y3",
training_frame=train_file,validation_frame=test_file,nfolds=10,fold_assignment
= "Stratified",balance_classes=TRUE,seed = 1234)

dl2

#UnBalanced + stratified

dl3 <- h2o.randomForest(x=1:95, y="Y3",
training_frame=train_file,validation_frame=test_file,nfolds=10,fold_assignment
= "Stratified",balance_classes=FALSE,seed = 1234)

dl3

#Unbalanced + Not stratified

dl4 <- h2o.randomForest(x=1:95, y="Y3",
training_frame=train_file,validation_frame=test_file,balance_classes=FALSE,s
eed = 1234)

dl4

v1 <- h2o.performance(dl1,wtest)

v1

v2 <- h2o.performance(dl2,wtest)

v2

v3 <- h2o.performance(dl3,wtest)

v3

v4 <- h2o.performance(dl4,wtest)

v4

############### TARGET Y4 ##############################

#Balanced + Not stratified

dl1 <- h2o.randomForest(x=1:95, y="Y4",
training_frame=train_file,validation_frame=test_file,balance_classes=TRUE,se
ed = 1234)

dl1

#Balanced + stratified
```

```
dl2 <- h2o.randomForest(x=1:95, y="Y4",
training_frame=train_file,validation_frame=test_file,nfolds=10,fold_assignment
= "Stratified",balance_classes=TRUE,seed = 1234)

dl2

#UnBalanced + stratified

dl3 <- h2o.randomForest(x=1:95, y="Y4",
training_frame=train_file,validation_frame=test_file,nfolds=10,fold_assignment
= "Stratified",balance_classes=FALSE,seed = 1234)

dl3

#Unbalanced + Not stratified

dl4 <- h2o.randomForest(x=1:95, y="Y4",
training_frame=train_file,validation_frame=test_file,balance_classes=FALSE,s
eed = 1234)

dl4

v1 <- h2o.performance(dl1,wtest)

v1

v2 <- h2o.performance(dl2,wtest)

v2

v3 <- h2o.performance(dl3,wtest)

v3

v4 <- h2o.performance(dl4,wtest)

v4
```