R Code of the Session to be Run Independent of automlr

Thiloshon

Tue Apr 23 22:20:14 2019

R Code of the Session, independent of automlr package

```
## Loading Packages

library(mlr)

# Please pass your tabular data here

data_original <- input_data_goes_here

target <- " Species "

################## Model 1 #################

data <- subset(data_original, subset = !is.na(data_original[target]))

resample <- mlr::makeResampleDesc("Holdout", split = 0.6)


# Task
learning.task <- mlr::makeClassifTask(id = nnet data = data, target = target)


# Learner
learner <- mlr::makeLearner( classif.nnet predict.type = "response", fix.factors.prediction = TRUE)


# Training and Testing
mod = mlr::resample(learner, learning.task, resample, measures = list(mmce, acc, timetrain))


################## Model 2 #################

data <- subset(data_original, subset = !is.na(data_original[target]))

resample <- mlr::makeResampleDesc("Holdout", split = 0.6)


# Task
learning.task <- mlr::makeClassifTask(id = ksvm data = data, target = target)


# Learner
learner <- mlr::makeLearner( classif.ksvm predict.type = "response", fix.factors.prediction = TRUE)


# Training and Testing
mod = mlr::resample(learner, learning.task, resample, measures = list(mmce, acc, timetrain))
```

```
################## Model 3 ##################

data <- subset(data_original, subset = !is.na(data_original[target]))

resample <- mlr::makeResampleDesc("Holdout", split = 0.6)


# Task
learning.task <- mlr::makeClassifTask(id = extraTrees data = data, target = target)


# Learner
learner <- mlr::makeLearner( classif.extraTrees predict.type = "response", fix.factors.prediction = TRU


# Training and Testing
mod = mlr::resample(learner, learning.task, resample, measures = list(mmce, acc, timetrain))


################## Model 4 ##################

data <- subset(data_original, subset = !is.na(data_original[target]))

resample <- mlr::makeResampleDesc("Holdout", split = 0.6)


# Task
learning.task <- mlr::makeClassifTask(id = fnn data = data, target = target)


# Learner
learner <- mlr::makeLearner( classif.fnn predict.type = "response", fix.factors.prediction = TRUE)


# Training and Testing
mod = mlr::resample(learner, learning.task, resample, measures = list(mmce, acc, timetrain))
```