Package: kwb.ml (via r-universe)

October 12, 2024

Title R Package with Functions, Workflows and Tutorials for Machine Learning at KWB

Version 0.1.0

Description R Package with Functions, Workflows and Tutorials for Machine Learning at KWB.

License MIT + file LICENSE

URL https://github.com/KWB-R/kwb.ml

BugReports https://github.com/KWB-R/kwb.ml/issues

Imports caret, doParallel, kwb.utils, magrittr, raster, randomForest

Suggests covr, dplyr, fs, knitr, kwb.nextcloud, rmarkdown, rgdal

Remotes github::kwb-r/kwb.nextcloud, github::kwb-r/kwb.utils

VignetteBuilder knitr

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

Repository https://kwb-r.r-universe.dev

RemoteUrl https://github.com/KWB-R/kwb.ml

RemoteRef HEAD

RemoteSha 60bd58d51602533928d57b84827f758663069de2

Contents

buildClassMod	*build image classification model (random forest)*

Description

Wrapper function for fitting a random forest using a multi-band image with the purpose of classi-
fying pixels into roof, street, and pervious (green areas, water surface), etc. These categories are
defined by the user in the ground truth data. Training and testing are done using repeated cross-
validation with package caret

Usage

```
buildClassMod(
  rawdir,
  image,
  groundTruth,
  groundTruthValues = list(roof = 1, street = 2, pervious = 3, shadow = 4, water = 5),
  overlayExists = FALSE,
  spectrSigName,
  modelName,
  nCores = parallel::detectCores() - 1,
  mtryGrd,
  ntreeGrd,
  nfolds = 3,
  nodesize = 3,
  cvrepeats = 2
)
```

Arguments

rawdir	path to directory containing the image to be classified and the ground truth data.
image	The image to be classified. Supported formats are given in the raster package's brick function.
groundTruth	shapefile containing polygons indicating the observed classes of a sample of pixels. These classes must be contained in a column named 'cover' in the shape-file's attribute table. The table may contain further columns.
groundTruthValues	
	list with key value pairs (default: list('roof' = 1, 'street' = 2, 'pervious' = 3, 'shadow' = 4, 'water' = 5))
overlayExists	If FALSE, the function overlays the ground truth data and the image (time con-suming) and saves an R object containing the former's spectral signatures with name spectrSigName (overlay object). If TRUE, the function will skip this over-lay operation and read an existing overlay object with name 'spectrSigName'. (default: FALSE)
spectrSigName	File name of overlay object, either for saving a new or load an existing file.

modelName	File name for saving the fitted random forest model
nCores	no. of cores for running in parallel mode (uses library 'doParallel'), (default: parallel::detectCores() - 1)
mtryGrd	Number of trees to grow. In the random forests literature, this is referred to as the ntree parameter. Larger number of trees produce more stable models and covariate importance estimates, but require more memory and a longer run time. For small datasets, 50 trees may be sufficient. For larger datasets, 500 or more may be required. Please consult the random forests literature for extensive discussion of this parameter (e.g. Cutler et al., 2007; Strobl et al., 2007; Strobl et al., 2008).
ntreeGrd	Number of variables available for splitting at each tree node. In the random forests literature, this is referred to as the mtry parameter. There is extensive discussion in the literature about the influence of mtry. Cutler et al. (2007) reported that different values of mtry did not affect the correct classification rates of their model and that other performance metrics (sensitivity, specificity, kappa, and ROC AUC) were stable under different values of mtry. On the other hand, Strobl et al. (2008) reported that mtry had a strong influence on predictor variable importance estimates.
nfolds	number of folds in repeated cross validation (caret), (default: 3)
nodesize	a single value (not included in grid search), (default: 3)
cvrepeats	number of repeats in repeated cross validation (caret), (default: 2)

Value

Writes spectralSignatures (if overlayExists is FALSE) and fitted random forest model with name modelName.

---

predictSurfClass	*apply model to predict surface type (roof, street, ...)*

Description

apply model to predict surface type (roof, street, ...)

Usage

```
predictSurfClass(rawdir, modelName, image, predName)
```

Arguments

rawdir	path to raw data directory
modelName	Name of fitted random forest model saved by buildClassMod
image	Image to be classified
predName	Name of output raster file (classified image)

Value

Writes raster in "rawdir" with file name "predName"

Index