EnMAP-Box Algorithms Documentation

Release 0.13.0

Andreas Rabe

Mar 02, 2020

GeoAlgorithms:

GeoAlgorithms

1.1 Accuracy Assessment

1.1.1 Classification Performance

Assesses the performance of a classification.

Prediction

Specify classification raster be evaluated

Reference

Specify reference classification raster (i.e. ground truth).

HTML Report

Specify output path for HTML report file (.html).

1.1.2 Clustering Performance

Assesses the performance of a clusterer.

Prediction

Specify clustering raster to be evaluated.

Reference

Specify reference clustering raster (i.e. ground truth).

HTML Report

Specify output path for HTML report file (.html).

1.1.3 Fraction Performance

Assesses the performance of class fractions in terms of AUC and ROC curves.

Prediction

Specify class fraction raster to be evaluated.

Reference

Specify reference classification raster (i.e. ground truth).

HTML Report

Specify output path for HTML report file (.html).

1.1.4 Regression Performance

Assesses the performance of a regression.

Prediction

Specify regression raster to be evaluated.

Reference

Specify reference regression raster (i.e. ground truth).

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True. Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Invert Mask

Whether or not to invert the selected mask.

HTML Report

Specify output path for HTML report file (.html).

1.2 Auxilliary

1.2.1 ClassDefinition from Raster

Creates a Class Definition string from a classification input raster for the usage in other EnMAP-Box algorithms (e.g. 'Classification from Vector'). See Log window for result.

Raster

Specify raster with defined class definition, e.g. classification or class fraction raster

1.2.2 Create additional Testdata

Based on the testdata additional datasets will be created using existing EnMAP-Box algorithms with predefined settings.

Create 30 m maps

Whether or not to create 30 m classification and fraction/regression maps.

Create 3.6 m maps

Whether or not to create 3.6 m classification and fraction/regression maps.

Create labeled Library

Whether or not to create spectral library with classification and fraction/regression labels at 30 m scale.

LandCover Classification for 6 classes at 30 m

Specify output path for classification raster.

LandCover Fraction for 6 classes at 30 m

Specify output path for fraction raster.

LandCover Classification for 6 classes at 3.6 m

Specify output path for classification raster.

LandCover Fraction for 6 classes at 3.6 m

Specify output path for fraction raster.

Library with 30 m profiles and classification/fraction labels

Specify output path for ENVI Spectral Library file.

1.2.3 Import Library

Import Library profiles and labels as Raster.

Library

Select path to an ENVI (e.g. .sli or .esl).

Import Profiles

undocumented parameter

Import Classification Labels (by classification scheme name)

undocumented parameter

Import Regression Labels (by output names)

undocumented parameter

Import Fraction Labels (by output names)

undocumented parameter

Output Raster

Specify output path for raster.

Output Classification

Specify output path for classification raster.

Output Regression

Specify output path for regression raster.

Output Fraction

Specify output path for fraction raster.

1.2.4 Open Test Maps

Opens testdata into current QGIS project (LandCov_BerlinUrbanGradient.shp, HighResolution_BerlinUrbanGradient.bsq, EnMAP_BerlinUrbanGradient.bsq, SpecLib_BerlinUrbanGradient.sli).

EnMAP (30m; 177 bands)

File name: EnMAP_BerlinUrbanGradient.bsq

Simulated EnMAP data (based on 3.6m HyMap imagery) acquired in August 2009 over south eastern part of Berlin covering an area of 4.32 km^2 (2.4 x 1.8 km). It has a spectral resolution of 177 bands and a spatial resolution of 30m.

HyMap (3.6m; Blue, Green, Red, NIR bands)

File name: HighResolution_BerlinUrbanGradient.bsq

HyMap image acquired in August 2009 over south eastern part of Berlin covering an area of 4.32 km^2 (2.4 x 1.8 km). This dataset was reduced to 4 bands (0.483, 0.558, 0.646 and 0.804 micrometers). The spatial resolution is 3.6m.

LandCover Layer

File name: LandCov_BerlinUrbanGradient.shp

Polygon shapefile containing land cover information on two classification levels. Derived from very high resolution aerial imagery and cadastral datasets.

Level 1 classes: Impervious; Other; Vegetation; Soil

Level 2 classes: Roof; Low vegetation; Other; Pavement; Tree; Soil

Library as Raster

File name: SpecLib_BerlinUrbanGradient.sli

Spectral library with 75 spectra (material level, level 2 and level 3 class information)

1.2.5 Open Test Library

1.2.6 Unique Values from Raster Band

This algorithm returns unique values from a raster band as a list. The output will be shown in the log window and can the copied from there accordingly.

Raster

Specify input raster.

Band

Specify input raster band.

1.2.7 View Raster Metadata

Prints all Raster metadata to log.

Raster

Specify input raster.

1.2.8 Unique Values from Vector Attribute

This algorithm returns unique values from vector attributes as a list, which is also usable as Class Definition in other algorithms. The output will be shown in the log window and can the copied from there accordingly.

Vector

Specify input vector.

Field

Specify field of vector layer for which unique values should be derived.

1.3 Classification

1.3.1 Fit GaussianProcessClassifier

Fits Gaussian Process Classifier. See Gaussian Processes for further information.

Raster

Specify input raster.

Classification

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. See GaussianProcessClassifier for information on different parameters.

Output Classifier

Specifiy output path for the classifier (.pkl). This file can be used for applying the classifier to an image using 'Classification -> Predict Classification' and 'Classification -> Predict ClassFraction'.

1.3.2 Fit LinearSVC

Fits a linear Support Vector Classification. Input data will be scaled and grid search is used for model selection.

Raster

Specify input raster.

Classification

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. For information on different parameters have a look at LinearSVC. See GridSearchCV for information on grid search and StandardScaler for scaling.

Output Classifier

Specifiy output path for the classifier (.pkl). This file can be used for applying the classifier to an image using 'Classification -> Predict Classification' and 'Classification -> Predict ClassFraction'.

1.3.3 Fit RandomForestClassifier

Fits a Random Forest Classifier

Raster

Specify input raster.

Classification

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. See RandomForestClassifier for information on different parameters. If this code is not altered, scikit-learn default settings will be used. 'Hint: you might want to alter e.g. the n_estimators value (number of trees), as the default is 10. So the line of code might be altered to 'estimator = RandomForestClassifier(n_estimators=100).'

Output Classifier

Specifiy output path for the classifier (.pkl). This file can be used for applying the classifier to an image using 'Classification -> Predict Classification' and 'Classification -> Predict ClassFraction'.

1.3.4 Fit SVC

Fits a Support Vector Classification. Input data will be scaled and grid search is used for model selection.

Raster

Specify input raster.

Classification

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. For information on different parameters have a look at SVC. See GridSearchCV for information on grid search and StandardScaler for scaling.

Output Classifier

Specifiy output path for the classifier (.pkl). This file can be used for applying the classifier to an image using 'Classification -> Predict Classification' and 'Classification -> Predict ClassFraction'.

1.3.5 Predict Classification

Applies a classifier to a raster.

Raster

Select raster file which should be classified.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Classifier

Select path to a classifier file (.pkl).

Output Classification

Specify output path for classification raster.

1.3.6 Predict Class Probability

Applies a classifier to a raster.

Raster

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True. Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Classifier

Select path to a classifier file (.pkl).

Probability

Specify output path for fraction raster.

1.4 Clustering

1.4.1 Fit AffinityPropagation

Fits a Affinity Propagation clusterer (input data will be scaled).

Raster

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. For information on different parameters have a look at AffinityPropagation. See Standard-Scaler for information on scaling

Output Clusterer

Specifiy output path for the clusterer (.pkl). This file can be used for applying the clusterer to an image using 'Clustering -> Predict Clustering'.

1.4.2 Fit Birch

Fits a Birch clusterer (input data will be scaled).

Raster

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. For information on different parameters have a look at Birch. See StandardScaler for information on scaling

Output Clusterer

Specifiy output path for the clusterer (.pkl). This file can be used for applying the clusterer to an image using 'Clustering -> Predict Clustering'.

1.4.3 Fit KMeans

Fits a KMeans clusterer (input data will be scaled).

Raster

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. For information on different parameters have a look at KMeans. See StandardScaler for information on scaling

Output Clusterer

Specifiy output path for the clusterer (.pkl). This file can be used for applying the clusterer to an image using 'Clustering -> Predict Clustering'.

1.4.4 Fit MeanShift

Fits a MeanShift clusterer (input data will be scaled).

Raster

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. For information on different parameters have a look at MeanShift. See StandardScaler for information on scaling

Output Clusterer

Specifiy output path for the clusterer (.pkl). This file can be used for applying the clusterer to an image using 'Clustering -> Predict Clustering'.

1.4.5 Predict Clustering

Applies a clusterer to a raster.

Raster

Select raster file which should be clustered.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Clusterer

Select path to a clusterer file (.pkl).

Clustering

Specify output path for classification raster.

1.5 Convolution, Morphology and Filtering

1.5.1 Spatial Convolution AiryDisk2DKernel

Applies AiryDisk2DKernel to image.

Raster

Specify input raster.

Code

Python code. See astropy.convolution.AiryDisk2DKernel for information on different parameters.

Output Raster

Specify output path for raster.

1.5.2 Spatial Convolution Box2DKernel

Applies Box2DKernel to image.

Raster

Specify input raster.

Code

Python code. See astropy.convolution.Box2DKernel for information on different parameters.

Output Raster

Specify output path for raster.

1.5.3 Spatial Convolution Gaussian2DKernel

Applies Gaussian2DKernel to image.

Raster

Specify input raster.

Code

Python code. See astropy.convolution.Gaussian2DKernel for information on different parameters.

Output Raster

Specify output path for raster.

1.5.4 Spatial Convolution MexicanHat2DKernel

Applies MexicanHat2DKernel to image.

Raster

Specify input raster.

Code

Python code. See astropy.convolution.MexicanHat2DKernel for information on different parameters.

Output Raster

Specify output path for raster.

1.5.5 Spatial Convolution Moffat2DKernel

Applies Moffat2DKernel to image.

Raster

Specify input raster.

Code

Python code. See astropy.convolution.Moffat2DKernel for information on different parameters.

Output Raster

Specify output path for raster.

1.5.6 Spatial Convolution Ring2DKernel

Applies Ring2DKernel to image.

Raster

Specify input raster.

Code

Python code. See astropy.convolution.Ring2DKernel for information on different parameters.

Output Raster

Specify output path for raster.

1.5.7 Spatial Convolution Tophat2DKernel

Applies Tophat2DKernel to image.

Raster

Specify input raster.

Code

Python code. See astropy.convolution.Tophat2DKernel for information on different parameters.

Output Raster

Specify output path for raster.

1.5.8 Spatial Convolution TrapezoidDisk2DKernel

Applies TrapezoidDisk2DKernel to image.

Raster

Specify input raster.

Code

Python code. See astropy.convolution.TrapezoidDisk2DKernel for information on different parameters.

Output Raster

Specify output path for raster.

1.5.9 Spectral Convolution Box1DKernel

Applies Box1DKernel.

Raster

Specify input raster.

Code

Python code. See astropy.convolution.Box1DKernel for information on different parameters.

Output Raster

Specify output path for raster.

1.5.10 Spectral Convolution Gaussian1DKernel

Applies Gaussian1DKernel.

Raster

Specify input raster.

Code

Python code. See astropy.convolution.Gaussian1DKernel for information on different parameters.

Output Raster

Specify output path for raster.

1.5.11 Spectral Convolution MexicanHat1DKernel

Applies MexicanHat1DKernel.

Raster

Specify input raster.

Code

Python code. See astropy.convolution.MexicanHat1DKernel for information on different parameters.

Output Raster

Specify output path for raster.

1.5.12 Spectral Convolution SavitzkyGolay1DKernel

Applies Savitzki Golay Filter.

Raster

Specify input raster.

Code

Python code. See scipy.signal.savgol_coeffs for information on different parameters.

Output Raster

Specify output path for raster.

1.5.13 Spectral Convolution Trapezoid1DKernel

Applies Trapezoid1DKernel.

Raster

Specify input raster.

Code

Python code. See astropy.convolution.Trapezoid1DKernel for information on different parameters.

Output Raster

Specify output path for raster.

1.5.14 Spatial Morphological Binary Closing

Applies binary_closing morphology filter to image. See Wikipedia for general information about closing morphology

Raster

Specify input raster.

Code

Python code. See scipy.ndimage.binary_closing for information on different parameters. At first, the structuring element will be defined (scipy.ndimage.generate_binary_structure). By default, its dimensions are always equal to 3. The connectivity parameter defines the type of neighborhood. In order create a bigger structuring element, the parameters in iterate_structure have to be altered (e.g. iterations=2 will increase the size to 5). Alternatively, a custom numpy array can be used as structural element.

Output Raster

Specify output path for raster.

1.5.15 Spatial Morphological Binary Dilation

Applies binary_dilation morphology filter to image. See Wikipedia for general information about dilation morphology

Raster

Specify input raster.

Code

Python code. See scipy.ndimage.binary_dilation for information on different parameters. At first, the structuring element will be defined (scipy.ndimage.generate_binary_structure). By default, its dimensions are always equal to 3. The connectivity parameter defines the type of neighborhood. In order create a bigger structuring element, the parameters in iterate_structure have to be altered (e.g. iterations=2 will increase the size to 5). Alternatively, a custom numpy array can be used as structural element.

Output Raster

Specify output path for raster.

1.5.16 Spatial Morphological Binary Erosion

Applies binary_erosion morphology filter to image. See Wikipedia for general information about erosion morphology

Raster

Specify input raster.

Code

Python code. See scipy.ndimage.binary_erosion for information on different parameters. At first, the structuring element will be defined (scipy.ndimage.generate_binary_structure). By default, its dimensions are always equal to 3. The connectivity parameter defines the type of neighborhood. In order create a bigger structuring element, the parameters in iterate_structure have to be altered (e.g. iterations=2 will increase the size to 5). Alternatively, a custom numpy array can be used as structural element.

Output Raster

Specify output path for raster.

1.5.17 Spatial Morphological Binary Fill Holes

Applies binary_fill_holes morphology filter to image.

Raster

Specify input raster.

Code

Python code. See scipy.ndimage.binary_fill_holes for information on different parameters. At first, the structuring element will be defined (scipy.ndimage.generate_binary_structure). By default, its dimensions are always equal to 3. The connectivity parameter defines the type of neighborhood. In order create a bigger structuring element, the parameters in iterate_structure have to be altered (e.g. iterations=2 will increase the size to 5). Alternatively, a custom numpy array can be used as structural element.

Output Raster

Specify output path for raster.

1.5.18 Spatial Morphological Binary Opening

Applies binary_opening morphology filter to image. See Wikipedia for general information about opening morphology

Raster

Specify input raster.

Code

Python code. See scipy.ndimage.binary_opening for information on different parameters. At first, the structuring element will be defined (scipy.ndimage.generate_binary_structure). By default, its dimensions are always equal to 3. The connectivity parameter defines the type of neighborhood. In order create a bigger structuring element, the parameters in iterate_structure have to be altered (e.g. iterations=2 will increase the size to 5). Alternatively, a custom numpy array can be used as structural element.

Output Raster

Specify output path for raster.

1.5.19 Spatial Morphological Binary Propagation

Applies binary_propagation morphology filter to image.

Raster

Specify input raster.

Code

Python code. See scipy.ndimage.binary_propagation for information on different parameters. At first, the structuring element will be defined (scipy.ndimage.generate_binary_structure). By default, its dimensions are always equal to 3. The connectivity parameter defines the type of neighborhood. In order create a bigger structuring element, the parameters in iterate_structure have to be altered (e.g. iterations=2 will increase the size to 5). Alternatively, a custom numpy array can be used as structural element.

Output Raster

Specify output path for raster.

1.5.20 Spatial Morphological Black Tophat

Applies black_tophat morphology filter to image. See Wikipedia for more information on top-hat transformation.

Raster

Specify input raster.

Code

Python code. See scipy.ndimage.black_tophat for information on different parameters. At first, the structuring element will be defined (scipy.ndimage.generate_binary_structure). By default, its dimensions are always equal to 3. The connectivity parameter defines the type of neighborhood. In order create a bigger structuring element, the parameters in iterate_structure have to be altered (e.g. iterations=2 will increase the size to 5). Alternatively, a custom numpy array can be used as structural element.

Output Raster

Specify output path for raster.

1.5.21 Spatial Morphological Gradient

Applies morphological_gradient morphology filter to image. See Wikipedia for more information about morphological gradients.

Raster

Specify input raster.

Code

Python code. See scipy.ndimage.morphological_gradient for information on different parameters. At first, the structuring element will be defined (scipy.ndimage.generate_binary_structure). By default, its dimensions are always equal to 3. The connectivity parameter defines the type of neighborhood. In order create a bigger structuring element, the parameters in iterate_structure have to be altered (e.g. iterations=2 will increase the size to 5). Alternatively, a custom numpy array can be used as structural element.

Output Raster

Specify output path for raster.

1.5.22 Spatial Morphological Grey Closing

Applies grey_closing morphology filter to image. See Wikipedia for general information about closing morphology.

Raster

Specify input raster.

Code

Python code. See scipy.ndimage.grey_closing for information on different parameters. At first, the structuring element will be defined (scipy.ndimage.generate_binary_structure). By default, its dimensions are always equal to 3. The connectivity parameter defines the type of neighborhood. In order create a bigger structuring element, the parameters in iterate_structure have to be altered (e.g. iterations=2 will increase the size to 5). Alternatively, a custom numpy array can be used as structural element.

Output Raster

Specify output path for raster.

1.5.23 Spatial Morphological Grey Dilation

Applies grey_dilation morphology filter to image. See Wikipedia for general information about closing morphology.

Raster

Specify input raster.

Code

Python code. See scipy.ndimage.grey_dilation for information on different parameters. At first, the structuring element will be defined (scipy.ndimage.generate_binary_structure). By default, its dimensions are always equal to 3. The connectivity parameter defines the type of neighborhood. In order create a bigger structuring element, the parameters in iterate_structure have to be altered (e.g. iterations=2 will increase the size to 5). Alternatively, a custom numpy array can be used as structural element.

Output Raster

Specify output path for raster.

1.5.24 Spatial Morphological Grey Erosion

Applies grey_erosion morphology filter to image. See Wikipedia for general information about erosion morphology

Raster

Specify input raster.

Code

Python code. See scipy.ndimage.grey_erosion for information on different parameters. At first, the structuring element will be defined (scipy.ndimage.generate_binary_structure). By default, its dimensions are always equal to 3. The connectivity parameter defines the type of neighborhood. In order create a bigger structuring element, the parameters in iterate_structure have to be altered (e.g. iterations=2 will increase the size to 5). Alternatively, a custom numpy array can be used as structural element.

Output Raster

Specify output path for raster.

1.5.25 Spatial Morphological Grey Opening

Applies grey_opening morphology filter to image. See Wikipedia for general information about opening morphology

Raster

Specify input raster.

Code

Python code. See scipy.ndimage.grey_opening for information on different parameters. At first, the structuring element will be defined (scipy.ndimage.generate_binary_structure). By default, its dimensions are always equal to 3. The connectivity parameter defines the type of neighborhood. In order create a bigger structuring element, the parameters in iterate_structure have to be altered (e.g. iterations=2 will increase the size to 5). Alternatively, a custom numpy array can be used as structural element.

Output Raster

Specify output path for raster.

1.5.26 Spatial Morphological Laplace

Applies morphological_laplace filter to image. See Wikipedia for more information on laplace filtering.

Raster

Specify input raster.

Code

Python code. See scipy.ndimage.morphological_laplace for information on different parameters.

Output Raster

Specify output path for raster.

1.5.27 Spatial Morphological White Tophat

Applies white_tophat morphology filter to image. See Wikipedia for more information on top-hat transformation.

Raster

Specify input raster.

Code

Python code. See scipy.ndimage.white_tophat for information on different parameters.

Output Raster

Specify output path for raster.

1.5.28 Spatial Gaussian Gradient Magnitude

Applies gaussian_gradient_magnitude filter to image.

Raster

Specify input raster.

Code

Python code. See scipy.ndimage.gaussian_gradient_magnitude for information on different parameters.

Output Raster

Specify output path for raster.

1.5.29 Spatial Generic Filter

Applies generic_filter to image using a user-specifiable function. This algorithm can perform operations you might know as moving window or focal statistics from some other GIS systems. Mind that depending on the function this algorithms can take some time to process.

Raster

Specify input raster.

Code

Python code. The function argument can take any callable function that expects a 1D array as input and returns a single value. You should alter the preset in the code window and define your own function. See scipy.ndimage.generic_filter for information on different parameters.

Output Raster

Specify output path for raster.

1.5.30 Spatial Laplace

Applies laplace filter to image. See Wikipedia for more information on laplace filtering.

Raster

Specify input raster.

Code

Python code. See scipy.ndimage.laplace for information on different parameters.

Output Raster

Specify output path for raster.

1.5.31 Spatial Maximum Filter

Applies maximum filter to image.

Raster

Specify input raster.

Code

Python code. See scipy.ndimage.maximum for information on different parameters.

Output Raster

Specify output path for raster.

1.5.32 Spatial Median Filter

Applies median filter to image.

Raster

Specify input raster.

Code

Python code. See scipy.ndimage.median for information on different parameters.

Output Raster

Specify output path for raster.

1.5.33 Spatial Minimum Filter

Applies minimum filter to image.

Raster

Specify input raster.

Code

Python code. See scipy.ndimage.minimum for information on different parameters.

Output Raster

Specify output path for raster.

1.5.34 Spatial Percentile Filter

Applies percentile filter to image.

Raster

Specify input raster.

Code

Python code. See scipy.ndimage.percentile_filter for information on different parameters.

Output Raster

Specify output path for raster.

1.5.35 Spatial Prewitt

Applies prewitt filter to image. See Wikipedia for further information on prewitt operators.

Raster

Specify input raster.

Code

Python code. See scipy.ndimage.prewitt for information on different parameters.

Output Raster

Specify output path for raster.

1.5.36 Spatial Sobel

Applies sobel filter to image. See Wikipedia for further information on sobel operators

Raster

Specify input raster.

Code

Python code. See scipy.ndimage.sobel for information on different parameters.

Output Raster

Specify output path for raster.

1.6 Create Raster

1.6.1 Classification from Fraction

Creates classification from class fraction. Winner class is equal to the class with maximum class fraction.

ClassFraction

Specify input raster.

Minimal overall coverage

Mask out all pixels that have an overall coverage less than the specified value. This controls how edges between labeled and no data regions are treated.

Minimal dominant coverage

Mask out all pixels that have a coverage of the predominant class less than the specified value. This controls pixel purity.

Output Classification

Specify output path for classification raster.

1.6.2 Classification from Vector

Creates a classification from a vector field with class ids.

Pixel Grid

Specify input raster.

Vector

Specify input vector.

Class id attribute

Vector field specifying the class ids.

Class Definition

Enter a class definition, e.g.:

ClassDefinition(names=['Urban', 'Forest', 'Water'], colors=['red', '#00FF00', (0, 0, 255)])

For supported named colors see the W3C recognized color keyword names.

Minimal overall coverage

Mask out all pixels that have an overall coverage less than the specified value. This controls how edges between labeled and no data regions are treated.

Minimal dominant coverage

Mask out all pixels that have a coverage of the predominant class less than the specified value. This controls pixel purity.

Oversampling factor

Defines the degree of detail by which the class information given by the vector is rasterized. An oversampling factor of 1 (default) simply rasterizes the vector on the target pixel grid.An oversampling factor of 2 will rasterize the vector on a target pixel grid with resolution twice as fine.An oversampling factor of 3 will rasterize the vector on a target pixel grid with resolution three times as fine, . . . and so on.

Mind that larger values are always better (more accurate), but depending on the inputs, this process can be quite computationally intensive, when a higher factor than 1 is used.

Output Classification

Specify output path for classification raster.

1.6.3 Fraction from Classification

Derive (binarized) class fractions from a classification.

Classification

Specify input raster.

Output Fraction

Specify output path for fraction raster.

1.6.4 Fraction from Vector

Derives class fraction raster from a vector file with sufficient class information.

Pixel Grid

Specify input raster.

Vector

Specify input vector.

Class id attribute

Vector field specifying the class ids.

Class Definition

Enter a class definition, e.g.:

ClassDefinition(names=['Urban', 'Forest', 'Water'], colors=['red', '#00FF00', (0, 0, 255)])

For supported named colors see the W3C recognized color keyword names.

Minimal overall coverage

Mask out all pixels that have an overall coverage less than the specified value. This controls how edges between labeled and no data regions are treated.

Minimal dominant coverage

Mask out all pixels that have a coverage of the predominant class less than the specified value. This controls pixel purity.

Oversampling factor

Defines the degree of detail by which the class information given by the vector is rasterized. An oversampling factor of 1 (default) simply rasterizes the vector on the target pixel grid.An oversampling factor of 2 will rasterize the vector on a target pixel grid with resolution twice as fine.An oversampling factor of 3 will rasterize the vector on a target pixel grid with resolution three times as fine, . . . and so on.

Mind that larger values are always better (more accurate), but depending on the inputs, this process can be quite computationally intensive, when a higher factor than 1 is used.

Output Fraction

Specify output path for fraction raster.

1.6.5 Raster from Vector

Converts vector to raster (using gdal rasterize).

Pixel Grid

Specify input raster.

Vector

Specify input vector.

Init Value

Pre-initialization value for the output raster before burning. Note that this value is not marked as the nodata value in the output raster.

Burn Value

Fixed value to burn into each pixel, which is covered by a feature (point, line or polygon).

Burn Attribute

Specify numeric vector field to use as burn values.

All touched

Enables the ALL_TOUCHED rasterization option so that all pixels touched by lines or polygons will be updated, not just those on the line render path, or whose center point is within the polygon.

Filter SQL

Create SQL based feature selection, so that only selected features will be used for burning.

Example: Level_2 = 'Roof' will only burn geometries where the Level_2 attribute value is equal to 'Roof', others will be ignored. This allows you to subset the vector dataset on-the-fly.

Data Type

Specify output datatype.

No Data Value

Specify output no data value.

Output Raster

Specify output path for raster.

1.7 Create Sample

1.7.1 Create Sample from synthetically mixed Endmembers

Derives a class fraction sample by synthetically mixing (pure) spectra from a classification sample.

Raster

Specify input raster.

Classification

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

n

Total number of samples to be generated.

Likelihood for mixing complexity 2

Specifies the probability of mixing spectra from 2 classes.

Likelihood for mixing complexity 3

Specifies the probability of mixing spectra from 3 classes.

Class likelihoods

Specifies the likelihoods for drawing spectra from individual classes.

In case of 'equalized', all classes have the same likelihhod to be drawn from.

In case of 'proportional', class likelihoods scale with their sizes.

Output Raster

Specify output path for raster.

Output Fraction

Specify output path for fraction raster.

1.7.2 Extract samples from raster and mask

Extract samples from raster and mask.

Raster

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Invert Mask

Whether or not to invert the selected mask.

Output Raster

Specify output path for raster.

1.7.3 Extract classification samples from raster and classification

Extract classification samples from raster and classification.

Raster

Specify input raster.

Classification

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Invert Mask

Whether or not to invert the selected mask.

Output Raster

Specify output path for raster.

Output Classification

Specify output path for classification raster.

1.7.4 Extract regression samples from raster and regression

Extract regression samples from raster and regression.

Raster

Specify input raster.

Regression

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Invert Mask

Whether or not to invert the selected mask.

Output Raster

Specify output path for raster.

Output Regression

Specify output path for regression raster.

1.7.5 Extract fraction samples from raster and fraction

Extract fraction samples from raster and fraction.

Raster

Specify input raster.

ClassFraction

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Invert Mask

Whether or not to invert the selected mask.

Output Raster

Specify output path for raster.

Output Fraction

Specify output path for fraction raster.

1.8 Masking

1.8.1 Build Mask from Raster

Builds a mask from a raster based on user defined values and value ranges.

Raster

Specify input raster.

Foreground values

List of values and ranges that are mapped to True, e.g. [1, 2, 5, range(5, 10)].

Background values

List of values and ranges that are mapped to False, e.g. [-9999, range(-10, 0)].

Output Mask

Specify output path for mask raster.

1.8.2 Apply Mask to Raster

Pixels that are masked out are set to the raster no data value.

Raster

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True. Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Invert Mask

Whether or not to invert the selected mask.

Masked Raster

Specify output path for raster.

1.9 Post-Processing

1.9.1 Fraction as RGB Raster

Creates a RGB representation from given class fractions. The RGB color of a specific pixel is the weighted mean value of the original class colors, where the weights are given by the corresponding class propability.

ClassFraction

Specify input raster.

Output Raster

Specify output path for raster.

1.10 Random

1.10.1 Random Points from Classification

Randomly samples a user defined amount of points/pixels from a classification raster and returns them as a vector dataset.

Classification

Specify input raster.

Number of Points per Class

Has to be a number or a list of numbers. When a single integer number is given (e.g. 100), equalised random sample will be taken, i.e. in this case 100 samples per class. For taking a disproportional random sample, where the amount of samples should differ between classes, provide a list of numbers. This list has to have as many arguments as classes in the classification and has to be ordered according to the classes (e.g. '[100, 70, 90]' in the case of three classes, 100 samples will be taken from the first class, 70 from the second, etc.). For a proportional stratified random sampling provide a float value between 0 and 1 (e.g. 0.3 for randomly drawing 30% of pixels in each class).

Output Vector

Specify output path for the vector.

1.10.2 Random Points from Mask

Randomly draws defined number of points from Mask and returns them as vector dataset.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Invert Mask

Whether or not to invert the selected mask.

Number of Points

Number of points to sample from mask.

Output Vector

Specify output path for the vector.

1.11 Regression

1.11.1 Fit GaussianProcessRegressor

Fits Gaussian Process Regression. See Gaussian Processes for further information.

Raster

Specify input raster.

Regression

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. See GaussianProcessRegressor for information on different parameters.

Output Regressor

Specifiy output path for the regressor (.pkl). This file can be used for applying the regressor to an image using 'Regression -> Predict Regression'.

1.11.2 Fit KernelRidge

Fits a KernelRidge Regression. Click here for additional information.

Raster

Specify input raster.

Regression

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. See KernelRidge for information on different parameters. See GridSearchCV for information on grid search and StandardScaler for scaling.

Output Regressor

Specifiy output path for the regressor (.pkl). This file can be used for applying the regressor to an image using 'Regression -> Predict Regression'.

1.11.3 Fit LinearRegression

Fits a Linear Regression.

Raster

Specify input raster.

Regression

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. See LinearRegression for information on different parameters. See StandardScaler for information on scaling.

Output Regressor

Specifiy output path for the regressor (.pkl). This file can be used for applying the regressor to an image using 'Regression -> Predict Regression'.

1.11.4 Fit LinearSVR

Fits a Linear Support Vector Regression.

Raster

Specify input raster.

Regression

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. See LinearSVR for information on different parameters. See GridSearchCV for information on grid search and StandardScaler for scaling.

Output Regressor

Specifiy output path for the regressor (.pkl). This file can be used for applying the regressor to an image using 'Regression -> Predict Regression'.

1.11.5 Fit RandomForestRegressor

Fits a Random Forest Regression.

Raster

Specify input raster.

Regression

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. See RandomForestRegressor for information on different parameters.

Output Regressor

Specifiy output path for the regressor (.pkl). This file can be used for applying the regressor to an image using 'Regression -> Predict Regression'.

1.11.6 Fit SVR

Fits a Support Vector Regression.

Raster

Specify input raster.

Regression

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. See SVR for information on different parameters. See GridSearchCV for information on grid search and StandardScaler for scaling.

Output Regressor

Specifiy output path for the regressor (.pkl). This file can be used for applying the regressor to an image using 'Regression -> Predict Regression'.

1.11.7 Predict Regression

Applies a regressor to an raster.

Raster

Select raster file which should be regressed.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Regressor

Select path to a regressor file (.pkl).

Output Regression

Specify output path for regression raster.

1.12 Resampling

1.12.1 Spatial Resampling (Raster)

Resamples a Raster into a target grid.

Pixel Grid

Specify input raster.

Raster

Specify input raster.

Resampling Algorithm

Specify resampling algorithm.

Output Raster

Specify output path for raster.

1.12.2 Spatial Resampling (Mask)

Resamples a Mask into a target grid.

Pixel Grid

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Minimal overall coverage

Mask out all pixels that have an overall coverage less than the specified value. This controls how edges between labeled and no data regions are treated.

Output Mask

Specify output path for mask raster.

1.12.3 Spatial Resampling (Classification)

Resamples a Classification into a target grid.

Pixel Grid

Specify input raster.

Classification

Specify input raster.

Minimal overall coverage

Mask out all pixels that have an overall coverage less than the specified value. This controls how edges between labeled and no data regions are treated.

Minimal dominant coverage

Mask out all pixels that have a coverage of the predominant class less than the specified value. This controls pixel purity.

Output Classification

Specify output path for classification raster.

1.12.4 Spatial Resampling (Regression)

Resamples a Regression into a target grid.

Pixel Grid

Specify input raster.

Regression

Specify input raster.

Minimal overall coverage

Mask out all pixels that have an overall coverage less than the specified value. This controls how edges between labeled and no data regions are treated.

Output Regression

Specify output path for regression raster.

1.12.5 Spatial Resampling (Fraction)

Resamples a Fraction into a target grid.

Pixel Grid

Specify input raster.

ClassFraction

Specify input raster.

Minimal overall coverage

Mask out all pixels that have an overall coverage less than the specified value. This controls how edges between labeled and no data regions are treated.

Output Fraction

Specify output path for fraction raster.

1.12.6 Spectral Resampling

Spectrally resample a raster.

Raster

Select raster file which should be resampled.

[Options 1] Spectral characteristic from predefined sensor

undocumented parameter

[Option 2] Spectral characteristic from Raster

Raster with defined wavelength and fwhm

[Option 3] Spectral characteristic from response function files.

Select path to an ENVI (e.g. .sli or .esl).

Resampling Algorithm

undocumented parameter

Output Raster

Specify output path for raster.

1.13 Transformation

1.13.1 Fit FactorAnalysis

Fits a Factor Analysis.

Raster

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. See FactorAnalysis for information on different parameters.

Output Transformer

Specifiy output path for the transformer (.pkl). This file can be used for applying the transformer to an image using 'Transformation -> Transform Raster' and 'Transformation -> InverseTransform Raster'.

1.13.2 Fit FastICA

Fits a FastICA (Independent Component Analysis).

Raster

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. See FastICA for information on different parameters.

Output Transformer

Specifiy output path for the transformer (.pkl). This file can be used for applying the transformer to an image using 'Transformation -> Transform Raster' and 'Transformation -> InverseTransform Raster'.

1.13.3 Fit FeatureAgglomeration

Fits a Feature Agglomeration.

Raster

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. See FeatureAgglomeration for information on different parameters.

Output Transformer

Specifiy output path for the transformer (.pkl). This file can be used for applying the transformer to an image using 'Transformation -> Transform Raster' and 'Transformation -> InverseTransform Raster'.

1.13.4 Fit Imputer

Fits an Imputer (Imputation transformer for completing missing values).

Raster

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. See Imputer for information on different parameters.

Output Transformer

Specifiy output path for the transformer (.pkl). This file can be used for applying the transformer to an image using 'Transformation -> Transform Raster' and 'Transformation -> InverseTransform Raster'.

1.13.5 Fit KernelPCA

Fits a Kernel PCA (Principal Component Analysis).

Raster

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. See KernelPCA for information on different parameters.

Output Transformer

Specifiy output path for the transformer (.pkl). This file can be used for applying the transformer to an image using 'Transformation -> Transform Raster' and 'Transformation -> InverseTransform Raster'.

1.13.6 Fit MaxAbsScaler

Fits a MaxAbsScaler (scale each feature by its maximum absolute value). See also examples for different scaling methods.

Raster

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. See MaxAbsScaler for information on different parameters.

Output Transformer

Specifiy output path for the transformer (.pkl). This file can be used for applying the transformer to an image using 'Transformation -> Transform Raster' and 'Transformation -> InverseTransform Raster'.

1.13.7 Fit MinMaxScaler

Fits a MinMaxScaler (transforms features by scaling each feature to a given range). See also examples for different scaling methods.

Raster

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. See MinMaxScaler for information on different parameters.

Output Transformer

Specifiy output path for the transformer (.pkl). This file can be used for applying the transformer to an image using 'Transformation -> Transform Raster' and 'Transformation -> InverseTransform Raster'.

1.13.8 Fit Normalizer

Fits a Normalizer (normalizes samples individually to unit norm). See also examples for different scaling methods.

Raster

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. See Normalizer for information on different parameters.

Output Transformer

Specifiy output path for the transformer (.pkl). This file can be used for applying the transformer to an image using 'Transformation -> Transform Raster' and 'Transformation -> InverseTransform Raster'.

1.13.9 Fit PCA

Fits a PCA (Principal Component Analysis).

Raster

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. See PCA for information on different parameters.

Output Transformer

Specifiy output path for the transformer (.pkl). This file can be used for applying the transformer to an image using 'Transformation -> Transform Raster' and 'Transformation -> InverseTransform Raster'.

1.13.10 Fit QuantileTransformer

Fits a Quantile Transformer (transforms features using quantiles information). See also examples for different scaling methods

Raster

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. See quantile_transform for information on different parameters.

Output Transformer

Specifiy output path for the transformer (.pkl). This file can be used for applying the transformer to an image using 'Transformation -> Transform Raster' and 'Transformation -> InverseTransform Raster'.

1.13.11 Fit RobustScaler

Fits a Robust Scaler (scales features using statistics that are robust to outliers). Click here for example. See also examples for different scaling methods.

Raster

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. See RobustScaler for information on different parameters.

Output Transformer

Specifiy output path for the transformer (.pkl). This file can be used for applying the transformer to an image using 'Transformation -> Transform Raster' and 'Transformation -> InverseTransform Raster'.

1.13.12 Fit StandardScaler

Fits a Standard Scaler (standardizes features by removing the mean and scaling to unit variance). See also examples for different scaling methods.

Raster

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Code

Scikit-learn python code. See StandardScaler for information on different parameters.

Output Transformer

Specifiy output path for the transformer (.pkl). This file can be used for applying the transformer to an image using 'Transformation -> Transform Raster' and 'Transformation -> InverseTransform Raster'.

1.13.13 Transform Raster

Applies a transformer to an raster.

Raster

Select raster file which should be regressed.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Transformer

Select path to a transformer file (.pkl).

Transformation

Specify output path for raster.

1.13.14 InverseTransform Raster

Performs an inverse transformation on an previously transformed raster (i.e. output of 'Transformation -> Transform Raster'). Works only for transformers that have an 'inverse_transform(X)' method. See scikit-learn documentations.

Raster

Specify input raster.

Mask

Specified vector or raster is interpreted as a boolean mask.

In case of a vector, all pixels covered by features are interpreted as True, all other pixels as False.

In case of a raster, all pixels that are equal to the no data value (default is 0) are interpreted as False, all other pixels as True.Multiband rasters are first evaluated band wise. The final mask for a given pixel is True, if all band wise masks for that pixel are True.

Transformer

Select path to a transformer file (.pkl).

Inverse Transformation

Specify output path for raster.