Package find projections

Comprehensively searches for informative projections (2-d projections) in the data to find boxes which separate out homogeneous data points.

Returns all projection boxes which match search criteria (support, purity).

For discrete output, the algorithm tries to find 2-d projection boxes which can separate out any class of data from the rest with high purity.

For numeric output, the algorithm tries to find 2-d projection boxes which can separate out data points with low variance.

Classes:

Datset *in find_projections.search_projections*

Data structure representing the input and output feature set.

Methods:

Datset(data)

Constructor. Creates Datset instance with numpy 2-d array of floats.

Methods:

setOutputForClassification(output)

Set output array for classification task.

setOutputForRegression(output)

Set output array for regression task.

isValid()

Checks if Datset instance has been populated properly.

Returns boolean.

getSize()

Returns the number of data points.

Search *in find_projections.search_projections*

Class to perform different types of search operations.

Methods:

search_projections(ds, binsize=10, support=50, purity=0.9, mode=0, num_threads=1)

Comprehensively evaluates all possible pairs of 2-d projections in the data

 Returns all projection boxes which match search criteria

Parameters

 ds: Datset instance created with your data

 binsize: No. of data points for binning. Should be a positive integer

support: Minimum number of data points to be present in a projection box for evaluation. Should be a positive integer.

 purity: Minimum purity (class proportion) in a projection box. Should be in the range 0.0 - 1.0

mode: Used for numeric output (regression-based analysis). Valid values are 0, 1, 2.

 num_threads: No. of threads for multi-threaded operation. Should be a positive integer

Returns FeatureMap instance containing all the projection boxes found meeting the search criteria

find_easy_explain_data(ds, validation_size=0.1, binsize=10, support=50, purity=0.9, mode=0, num_threads=1)

Learns decision list of projection boxes for easy-to-explain data (for classification/regression). Returns FeatureMap instance containing all the projection boxes found meeting the search criteria.

Parameters

ds: Datset instance created with your data

 binsize: No. of data points for binning. Should be a positive integer

support: Minimum number of data points to be present in a projection box for evaluation. Should be a positive integer.

 purity: Minimum purity (class proportion) in a projection box. Should be in the range 0.0 - 1.0

mode: Used for numeric output (regression-based analysis). Valid values are 0, 1, 2. 0: Low variance boxes, 1: high mean boxes and 2: low mean boxes.

num_threads: No. of threads for multi-threaded operation. Should be a positive integer

Returns FeatureMap instance containing all the projection boxes found meeting the search criteria

FeatureMap *in find_projections.search_projections*

Container class containing projection boxes found from search operations

Methods:

get_num_projections():

Returns the total number of projection boxes in this container object

get_projection(i)

Retrieve the i'th projection-box

projection *in find_projections*

Represents 2-d subset of data, bounded like a rectangular box. Each projection box is defined by a pair of attributes and a range of values for each of the attributes.

Methods:

get_total()

Returns the total number of data points contained in the projection box.

get_att1()

Returns the index of the first attribute

get_att2()

Returns the index of the second attribute

get_att1_start()

Returns the starting value for the range of the first attribute.

get_att1_end()

Returns the ending value for the range of the first attribute.

get_att2_start()

Returns the starting value for the range of the second attribute.

get_att2_end()

Returns the ending value for the range of the second attribute.

get_projection_metric()

Returns the projection matric. If discrete_projection, returns the class label (majority class).

If numeric_projection, returns the mean of points in the box.

point_lies_in_projection(ds, row)

Returns boolean whether the row'th point in ds data lies inside the projection.

pprojection()

Prints out the contents of the projection.

For a discrete_projection, prints out the box contents in the following order-

Class_label_index, Att1, Att2, Att1_start, Att1_end, Att2_start, Att2_end, Positives, Negatives, Purity of the box.

For a numeric_projection, prints out the box contents in the following order-

Att1, Att2, Att1_start, Att1_end, Att2_start, Att2_end, Total no. of points, mean of output of points and sum-squared-error of points in the box.

discrete_projection (derives from projection class) *in find_projections*

Represents a projection box learnt from data with discrete output (multi-class data).

Methods:

get_class()

Returns the index of the majority class of data points inside the projection box. This will be referred to as positive class and other class points will be referred to as negatives.

get_pos()

Returns the number of positive data points in the projection box.

get_neg()

Returns the number of negative data points in the projection box.

numeric_projection (derives from projection class) *in find_projections*

Represents a projection box learnt from data with numeric output (output for regression).

<u>Methods:</u>

get_mean()

Returns the arithmetic mean of the data points contained within the projection box.

get_sum_sq_error()

Returns the number sum-of-squared-error for the points contained within the projection box.