

---

# **KUHERD Documentation**

***Release 1.0***

**Joseph St.Amand**

**May 08, 2017**



---

## Contents:

---

<b>1</b>	<b>KUHERD package</b>	<b>1</b>
1.1	Subpackages	1
1.1.1	KUHERD.Experiments package	1
1.1.1.1	Submodules	1
1.1.1.2	KUHERD.Experiments.MultiDT module	1
1.1.1.3	KUHERD.Experiments.MultiLR module	1
1.1.1.4	KUHERD.Experiments.MultiNB module	1
1.1.1.5	KUHERD.Experiments.MultiRF module	1
1.1.1.6	KUHERD.Experiments.MultiSVM module	2
1.1.1.7	Module contents	2
1.1.2	KUHERD.Tools package	2
1.1.2.1	Submodules	2
1.1.2.2	KUHERD.Tools.makePredictions module	2
1.1.2.3	KUHERD.Tools.trainModel module	2
1.1.2.4	Module contents	2
1.2	Submodules	2
1.3	KUHERD.FeatureSelector module	2
1.4	KUHERD.HerdVectorizer module	2
1.5	KUHERD.LabelSets module	4
1.6	KUHERD.LabelTransformations module	4
1.7	KUHERD.Models module	5
1.8	KUHERD.MultiFeatureSelector module	6
1.9	Module contents	6
<b>2</b>	<b>Indices and tables</b>	<b>7</b>
	<b>Python Module Index</b>	<b>9</b>
	<b>Index</b>	<b>11</b>



## 1.1 Subpackages

### 1.1.1 KUHERD.Experiments package

#### 1.1.1.1 Submodules

#### 1.1.1.2 KUHERD.Experiments.MultiDT module

`KUHERD.Experiments.MultiDT.MultiDT()`  
Program for running an experiment using Decision Tree classifier.

#### 1.1.1.3 KUHERD.Experiments.MultiLR module

`KUHERD.Experiments.MultiLR.MultiLR()`  
Program for running an experiment using Logistic Regression classifier.

#### 1.1.1.4 KUHERD.Experiments.MultiNB module

`KUHERD.Experiments.MultiNB.MultiNB()`  
Program for running an experiment using Naive Bayes classifier.

#### 1.1.1.5 KUHERD.Experiments.MultiRF module

`KUHERD.Experiments.MultiRF.MultiRF()`  
Program for running an experiment using Random Forest classifier.

### 1.1.1.6 KUHERD.Experiments.MultiSVM module

`KUHERD.Experiments.MultiSVM.MultiSVM()`

Program for running an experiment using Support Vector Machine classifier.

### 1.1.1.7 Module contents

## 1.1.2 KUHERD.Tools package

### 1.1.2.1 Submodules

### 1.1.2.2 KUHERD.Tools.makePredictions module

### 1.1.2.3 KUHERD.Tools.trainModel module

`KUHERD.Tools.trainModel.main()`

### 1.1.2.4 Module contents

## 1.2 Submodules

## 1.3 KUHERD.FeatureSelector module

**class** `KUHERD.FeatureSelector.FeatureSelector` (*scoring\_function*, *kbest*)

Bases: `object`

**fit** (*X*, *Y*, *label\_set*)

Fits the data by training the feature selection model compomnent.

**Args:** *X* (numpy matrix): The data matrix. *y* (integer list): The labels for the data. *label\_set* (str): Either 'purpose' or 'field'.

**Return:** None

**transform** (*X*)

Transforms the data, retaining only features learned in the "fit" process.

**Args:** *X* (numpy matrix): The data matrix. *y* (integer list): The labels for the data.

**Return:** (numpy matrix): Transformed data matrix.

## 1.4 KUHERD.HerdVectorizer module

**class** `KUHERD.HerdVectorizer.HerdVectorizer` (*config*)

Bases: `object`

Main class responsible for vectorization of the text data. This class is extremely configurable, with many options for each preprocessing behavior, bigrams, stemmers, stopwords, and feature selection. Use of this class is done in the following manner: - Set configuration options for `preproc_config`, `bigram_config`, `stemmer`, `stopwords`, and feature selection. - Train the vectorizer on a set of documents and their corresponding labels. - After training is complete, the documents may be given to the transform function to convert to TFIDF form.

**create\_bigram\_index\_map** (*tokenized\_docs*)

Creates a mapping from each bigram to a column index.

**Args:** *tokenized\_docs* (list): A list of documents, each document is represented as a single long string

**Return:** (dictionary): Dictionary where keys are tokens, values are the index into a feature matrix.

**create\_token\_index\_map** (*tokenized\_docs*)

Given the tokenized documents, finds all unique tokens and forms an index map

**Args:** *tokenized\_docs* (list): A list of documents, each document is represented as a single long string

**Return:** (dictionary): Dictionary where keys are tokens, values are the index into a feature matrix.

**static filter\_docs** (*tokenized\_docs*, *tok\_index\_map*)

Filters tokenized documents, removing all tokens which are not recognized by the specified token index map.

**Args:** *tokenized\_docs* (list): A list of documents, each document is represented as a single long string  
*tok\_index\_map* (dictionary): A mapping from tokens to their index in the feature matrix.

**Return:** (list): A list of documents, where each document is a list of (filtered) tokens.

**form\_bigram\_count\_matrix** (*tokenized\_docs*)

Calculates a bigram count matrix from a list of tokenized documents.

**Args:** *tokenized\_docs* (list): A list of documents, each document is represented as a single long string

**Return:** (sparse numpy matrix): A sparse count matrix in COO format.

**form\_count\_matrix** (*tokenized\_docs*)

Forms the count matrix from the tokenized documents

**Args:** *tokenized\_docs* (list): A list of lists representing the tokenized documents. Each document is a list of tokens.

**Return:** (sparse numpy matrix): A sparse count matrix in COO format.

**getConfig** ()

Retrieve the complete configuration needed to build a vectorizer.

Returns the complete configuration needed to build a vectorizer. Note that the config returned may only be used to train a new vectorizer. The config does NOT give model persistence.

**get\_bigram\_config** ()

Retrieve the bigram configuration.

**get\_preproc\_config** ()

Retrieve the preprocessor configuration.

**static lancaster\_stemmer** (*docs*)

Lancaster stemming algorithm

**static porter\_stemmer** (*docs*)

Porter stemming algorithm

**set\_bigram\_config** (*name*, *value*)

Set the bigram configuration.

**set\_bigrams** (*bigrams*, *bigram\_window\_size*, *bigram\_filter\_size*, *bigram\_nbest*)

Set the bigram configuration.

**set\_feature\_selector** (*scoring\_func*, *kbest*, *multi\_type*)

Set the feature selection configuration values.

**set\_preproc\_config** (*name, value*)  
Set the preprocessor configuration value.

**set\_stemmer** (*the\_stemmer*)  
Set the stemmer configuration values.

**static snowball\_stemmer** (*docs*)  
Snowball stemming algorithm

**tokenize\_docs** (*docs*)  
Breaks each document down into a list of words(tokens).

Converts a list of documents(each document is given as a single string) and converts them to their tokenized form in the following manner(some steps may be skipped if configured as such in the configuration settings)

- break document into tokens
- remove punctuation
- stem tokens

**Args:** docs (list): A list of documents, each document is represented as a single long string

**Return:** (list): The tokenized documents as a list of lists, each item of the outer list is a document, which is represented as a list of words.

**train** (*docs, y, label\_set*)  
Takes a list of documents, and the corresponding labels and trains the preprocessor(including feature selection).

**Args:** docs (list): A list of documents, where each document is represented as a string. y (list): A list of integers representing the label for each document. label\_set (str): Specifies the label set so that the input 'y' may be interpreted. Valid entries are either 'purpose' or 'field'.

@param docs The list of documents @param y A vector of labels which correspond to each document

**transform\_data** (*docs*)  
Transforms documents into a sparse matrix.

**Must be called after the preprocessor has been trained on some data. Process is as follows:** -  
tokenize documents -search for bigrams -transform to TFIDF representation -select features

**Args:** docs (list): A list of documents, each document is represented as a string.

**Return:** (sparse numpy matrix): A sparse CSR formatted matrix, each row corresponds to a document, ordering of documents is preserved.

KUHERD.HerdVectorizer.**main**()

## 1.5 KUHERD.LabelSets module

## 1.6 KUHERD.LabelTransformations module

KUHERD.LabelTransformations.**label2mat** (*x, label\_type*)  
Converts a vector of integers to a matrix of zero-one valued columns.

**Args:** label\_vec (list): A vector containing integer values mapping to members of the label\_type. label\_type (str): Specifies label\_type of label\_vec, either 'purpose' and 'field'.



**Return:** (numpy matrix): A matrix of zero-one valued columns.

`KUHERD.LabelTransformations.mat2vec(M)`

Converts a zero-one valued label matrix to an integer valued label vector.

**Args:** M (numpy mat): A zero-one valued label matrix.

`KUHERD.LabelTransformations.vec2mat(x, label_type)`

Converts a vector of integers to a matrix of of zero-one valued columns.

**Args:** x (list): A vector containing integer values mapping to members of the label\_type. label\_type (str): Specifies label\_type of label\_vec, either 'purpose' and 'field'.

**Return:** (numpy matrix): A matrix of zero-one valued columns.

`KUHERD.LabelTransformations.vec2string(label_vec, label_type)`

Converts vector containing integers to a string representation using the label set dictionaries.

**Args:** label\_vec (list): A vector containing integer values mapping to members of the label\_type. label\_type (str): Specifies label\_type of label\_vec, either 'purpose' and 'field'.

**Return:** (list): A list of strings that are members oif the label\_type.

## 1.7 KUHERD.Models module

`class KUHERD.Models.ClassificationModel(config)`

Bases: object

**fit**(X, Y)

Trains the model.

Fitting or “training” must be done before the model is able to make predictions.

**Args:** X (numpy matrix): Training samples. Y (numpy matrix): Training labels.

**Returns:** None: No return value.

**get\_config**()

Returns the configuration used to build this model.

**Returns:** dict: dictionary containing target label set, internal model configuration, and model name.

**predict**(X)

Make predictions.

**Args:** X (numpy matrix): Training samples.

**Returns:** numpy matrix: predicted label values.

`class KUHERD.Models.PurposeFieldModel(config)`

Bases: object

**fit**(abstracts, Y\_purpose, Y\_field)

Trains the model.

Input arguments must all be the same length.

**Args:** abstracts (list): A list of documents, each document is represented as a list of words. Y\_purpose (list): A list of labels of the 'purpose' variety. Y\_field (list): A list of labels of the 'field' variety.

**get\_config**()

Returns the configuration used to build this model.

**Returns:** dict: dictionary containing the following keys, ‘purpose\_vectorizer’, ‘field\_vectorizer’, ‘purpose\_model’, ‘field\_model’. Each entry is the configuration required to build the model.

**predict** (*abstracts*)

Make predictions on the input data.

The list of documents input is vectorized and input to the prediction model, which generates label predictions. This process is done separately for generating both purpose and field label predictions.

**Args:** abstracts (list): A list of documents, each document is represented as a list of words.

**Returns:** dictionary: dictionary containing two lists of predictions, dictionary keys are ‘purpose’ and ‘field’.

## 1.8 KUHERD.MultiFeatureSelector module

```
class KUHERD.MultiFeatureSelector.MultiFeatureSelector(scoring_function, kbest,  
                                                         multi_integrator)
```

Bases: object

**fit** (*X*, *Y*, *label\_set*)

Trains the feature selection process

**Args:** X (numpy matrix): Training samples. Y (numpy matrix): Training Labels. label\_set (str): Denotes if label set is of the ‘purpose’ or ‘field’ type.

**transform** (*X*)

Transforms the data by selecting the features learned in the training or “fit” process.

**Args:** X (numpy matrix): Data samples to run feature selection on.

**Return:** (numpy matrix): Data with only the selected features.

## 1.9 Module contents

## CHAPTER 2

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### k

- KUHERD, [6](#)
- KUHERD.Experiments, [2](#)
- KUHERD.Experiments.MultiDT, [1](#)
- KUHERD.Experiments.MultiLR, [1](#)
- KUHERD.Experiments.MultiNB, [1](#)
- KUHERD.Experiments.MultiRF, [1](#)
- KUHERD.Experiments.MultiSVM, [2](#)
- KUHERD.FeatureSelector, [2](#)
- KUHERD.HerdVectorizer, [2](#)
- KUHERD.LabelSets, [4](#)
- KUHERD.LabelTransformations, [4](#)
- KUHERD.Models, [5](#)
- KUHERD.MultiFeatureSelector, [6](#)
- KUHERD.Tools, [2](#)
- KUHERD.Tools.trainModel, [2](#)



## C

ClassificationModel (class in KUHERD.Models), 5  
 create\_bigram\_index\_map()  
     (KUHERD.HerdVectorizer.HerdVectorizer  
     method), 2  
 create\_token\_index\_map()  
     (KUHERD.HerdVectorizer.HerdVectorizer  
     method), 3

## F

FeatureSelector (class in KUHERD.FeatureSelector), 2  
 filter\_docs() (KUHERD.HerdVectorizer.HerdVectorizer  
     static method), 3  
 fit() (KUHERD.FeatureSelector.FeatureSelector method),  
     2  
 fit() (KUHERD.Models.ClassificationModel method), 5  
 fit() (KUHERD.Models.PurposeFieldModel method), 5  
 fit() (KUHERD.MultiFeatureSelector.MultiFeatureSelector  
     method), 6  
 form\_bigram\_count\_matrix()  
     (KUHERD.HerdVectorizer.HerdVectorizer  
     method), 3  
 form\_count\_matrix() (KUHERD.HerdVectorizer.HerdVectorizer  
     method), 3

## G

get\_bigram\_config() (KUHERD.HerdVectorizer.HerdVectorizer  
     method), 3  
 get\_config() (KUHERD.Models.ClassificationModel  
     method), 5  
 get\_config() (KUHERD.Models.PurposeFieldModel  
     method), 5  
 get\_preproc\_config() (KUHERD.HerdVectorizer.HerdVectorizer  
     method), 3  
 getConfig() (KUHERD.HerdVectorizer.HerdVectorizer  
     method), 3

## H

HerdVectorizer (class in KUHERD.HerdVectorizer), 2

## K

KUHERD (module), 6  
 KUHERD.Experiments (module), 2  
 KUHERD.Experiments.MultiDT (module), 1  
 KUHERD.Experiments.MultiLR (module), 1  
 KUHERD.Experiments.MultiNB (module), 1  
 KUHERD.Experiments.MultiRF (module), 1  
 KUHERD.Experiments.MultiSVM (module), 2  
 KUHERD.FeatureSelector (module), 2  
 KUHERD.HerdVectorizer (module), 2  
 KUHERD.LabelSets (module), 4  
 KUHERD.LabelTransformations (module), 4  
 KUHERD.Models (module), 5  
 KUHERD.MultiFeatureSelector (module), 6  
 KUHERD.Tools (module), 2  
 KUHERD.Tools.trainModel (module), 2

## L

label2mat() (in module  
     KUHERD.LabelTransformations), 4  
 lancaster\_stemmer() (KUHERD.HerdVectorizer.HerdVectorizer  
     static method), 3

## M

main() (in module KUHERD.HerdVectorizer), 4  
 main() (in module KUHERD.Tools.trainModel), 2  
 mat2vec() (in module KUHERD.LabelTransformations),  
     5  
 MultiDT() (in module KUHERD.Experiments.MultiDT),  
     1  
 MultiFeatureSelector (class in  
     KUHERD.MultiFeatureSelector), 6  
 MultiLR() (in module KUHERD.Experiments.MultiLR),  
     1  
 MultiNB() (in module KUHERD.Experiments.MultiNB),  
     1  
 MultiRF() (in module KUHERD.Experiments.MultiRF),  
     1  
 MultiSVM() (in module  
     KUHERD.Experiments.MultiSVM), 2

## P

`porter_stemmer()` (KUHERD.HerdVectorizer.HerdVectorizer static method), 3  
`predict()` (KUHERD.Models.ClassificationModel method), 5  
`predict()` (KUHERD.Models.PurposeFieldModel method), 6  
`PurposeFieldModel` (class in KUHERD.Models), 5

## S

`set_bigram_config()` (KUHERD.HerdVectorizer.HerdVectorizer method), 3  
`set_bigrams()` (KUHERD.HerdVectorizer.HerdVectorizer method), 3  
`set_feature_selector()` (KUHERD.HerdVectorizer.HerdVectorizer method), 3  
`set_preproc_config()` (KUHERD.HerdVectorizer.HerdVectorizer method), 3  
`set_stemmer()` (KUHERD.HerdVectorizer.HerdVectorizer method), 4  
`snowball_stemmer()` (KUHERD.HerdVectorizer.HerdVectorizer static method), 4

## T

`tokenize_docs()` (KUHERD.HerdVectorizer.HerdVectorizer method), 4  
`train()` (KUHERD.HerdVectorizer.HerdVectorizer method), 4  
`transform()` (KUHERD.FeatureSelector.FeatureSelector method), 2  
`transform()` (KUHERD.MultiFeatureSelector.MultiFeatureSelector method), 6  
`transform_data()` (KUHERD.HerdVectorizer.HerdVectorizer method), 4

## V

`vec2mat()` (in module KUHERD.LabelTransformations), 5  
`vec2string()` (in module KUHERD.LabelTransformations), 5