

UM

Generated by Doxygen 1.8.18

1 Namespace Index	1
1.1 Packages	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Namespace Documentation	7
4.1 Build Namespace Reference	7
4.1.1 Function Documentation	7
4.1.1.1 build_tree()	7
4.1.1.2 count_all()	8
4.2 Correctness_of_building Namespace Reference	8
4.2.1 Function Documentation	8
4.2.1.1 correctness_of_building()	8
4.2.1.2 correctness_of_incremental()	9
4.3 Data Namespace Reference	9
4.3.1 Function Documentation	9
4.3.1.1 read_data()	9
4.3.1.2 read_data_second()	10
4.3.1.3 read_data_third()	10
4.4 Data_matching Namespace Reference	10
4.4.1 Function Documentation	10
4.4.1.1 confusion_matrix()	10
4.4.1.2 data_matching()	11
4.4.1.3 find()	11
4.4.1.4 for_basic_tree()	11
4.4.1.5 for_tree_incremental_learning()	12
4.5 Incremental_learning Namespace Reference	12
4.5.1 Function Documentation	12
4.5.1.1 find_tree()	12
4.5.1.2 incremental_learning()	13
4.6 Mesure Namespace Reference	13
4.6.1 Function Documentation	13
4.6.1.1 count()	13
4.6.1.2 gain()	14
4.6.1.3 giny()	14
4.7 Print_tree Namespace Reference	15
4.7.1 Function Documentation	15
4.7.1.1 print_tree()	15
4.8 Split Namespace Reference	15

4.8.1 Function Documentation	15
4.8.1.1 check_split()	15
4.8.1.2 make_split()	16
4.9 Test Namespace Reference	16
4.9.1 Function Documentation	17
4.9.1.1 data_matching_for_basic_tree()	17
4.9.1.2 data_matching_for_tree_incremental_learning()	17
4.9.1.3 test()	17
4.9.1.4 write_dict()	18
4.9.1.5 write_file()	18
4.9.1.6 write_time()	18
5 Class Documentation	19
5.1 Build.Quantity Class Reference	19
5.1.1 Detailed Description	19
5.1.2 Constructor & Destructor Documentation	19
5.1.2.1 __init__()	19
5.1.3 Member Data Documentation	20
5.1.3.1 quantity	20
5.2 Build.Subtree_Values Class Reference	20
5.2.1 Detailed Description	20
5.2.2 Constructor & Destructor Documentation	20
5.2.2.1 __init__()	21
5.2.3 Member Data Documentation	21
5.2.3.1 false_data	21
5.2.3.2 gain	21
5.2.3.3 left_next	21
5.2.3.4 question	21
5.2.3.5 right_next	22
5.2.3.6 true_data	22
6 File Documentation	23
6.1 Build.py File Reference	23
6.2 Correctness_of_building.py File Reference	23
6.3 Data.py File Reference	24
6.4 Data_matching.py File Reference	24
6.5 Incremental_learning.py File Reference	24
6.6 Mesure.py File Reference	25
6.7 Print_tree.py File Reference	25
6.8 Split.py File Reference	25
6.9 Test.py File Reference	25
Index	27

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

Build	7
Correctness_of_building	8
Data	9
Data_matching	10
Incremental_learning	12
Measure	13
Print_tree	15
Split	15
Test	16

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Build.Quantity	
Class with a dictionary in which the number of elements	19
Build.Subtree_Values	
Class with elemnet to save in all knots	20

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Build.py	23
Correctness_of_building.py	23
Data.py	24
Data_matching.py	24
Incremental_learning.py	24
Mesure.py	25
Print_tree.py	25
Split.py	25
Test.py	25

Chapter 4

Namespace Documentation

4.1 Build Namespace Reference

Classes

- class [Quantity](#)
class with a dictionary in which the number of elements
- class [Subtree_Values](#)
class with elemnet to save in all knots

Functions

- def [count_all](#) (training_data)
function to count all elements
- def [build_tree](#) (training_data)
function to build tree

4.1.1 Function Documentation

4.1.1.1 build_tree()

```
def Build.build_tree (  
    training_data )
```

function to build tree

Parameters

<i>training_elements</i>	list of training elements
--------------------------	---------------------------

Returns

object [Subtree_Values](#) for knot
 object [Quantity](#) for knot

4.1.1.2 count_all()

```
def Build.count_all (
    training_data )
```

function to count all elements

Parameters

<i>training_elements</i>	list in which items will be counted
--------------------------	-------------------------------------

Returns

quantity of elements all elements

4.2 Correctness_of_building Namespace Reference**Functions**

- def [correctness_of_building](#) (start, quantity)
function to build and print tree
- def [correctness_of_incremental](#) (start, quantity, tree)
function to incremental learning and print tree

4.2.1 Function Documentation**4.2.1.1 correctness_of_building()**

```
def Correctness_of_building.correctness_of_building (
    start,
    quantity )
```

function to build and print tree

Parameters

<i>start</i>	- the data number in the csv file used to train the tree
<i>quantity</i>	- quantity of data used to train the tree

4.2.1.2 correctness_of_incremental()

```
def Correctness_of_building.correctness_of_incremental (
    start,
    quantity,
    tree )
```

function to incremental learning and print tree

Parameters

<i>start</i>	- the data number in the csv file used to train the tree
<i>quantity</i>	- quantity of data used to train the tree
<i>tree</i>	- old tree

4.3 Data Namespace Reference

Functions

- def [read_data](#) ()
function to read and optimization data y
- def [read_data_second](#) ()
function to read and optimization data agaricus-lepiota
- def [read_data_third](#) ()
function to read and optimization data iris

4.3.1 Function Documentation

4.3.1.1 read_data()

```
def Data.read_data ( )
```

function to read and optimization data y

Returns

data

4.3.1.2 read_data_second()

```
def Data.read_data_second ( )
```

function to read and optimization data agaricus-lepiota

Returns

data

4.3.1.3 read_data_third()

```
def Data.read_data_third ( )
```

function to read and optimization data iris

Returns

data

4.4 Data_matching Namespace Reference

Functions

- def [for_basic_tree](#) (quantity, data)
function to do test for basic tree
- def [for_tree_incremental_learning](#) (quantity_basic_tree, quantity, data)
function to do test for tree with one incremental learning
- def [confusion_matrix](#) (data_test, tree)
function for creating confusion matrix
- def [data_matching](#) (tree, data_test)
function for calculating test data matching
- def [find](#) (tree, dt)
find the list for test data

4.4.1 Function Documentation

4.4.1.1 confusion_matrix()

```
def Data_matching.confusion_matrix (
    data_test,
    tree )
```

function for creating confusion matrix

Parameters

<i>tree</i>	
<i>data</i>	test

Returns

matrix - confusion matrix

4.4.1.2 data_matching()

```
def Data_matching.data_matching (
    tree,
    data_test )
```

function for calculating test data matching

Parameters

<i>tree</i>	
<i>data</i>	test

4.4.1.3 find()

```
def Data_matching.find (
    tree,
    dt )
```

find the list for test data

Parameters

<i>tree</i>	
<i>dt</i>	element for which we are looking for a place

4.4.1.4 for_basic_tree()

```
def Data_matching.for_basic_tree (
    quantity,
    data )
```

function to do test for basic tree

Parameters

<i>quantity</i>	- quantity of data used to tarin dree
-----------------	---------------------------------------

Returns

arrange - basic tree arrange

matrix - confusion matrix

4.4.1.5 for_tree_incremental_learning()

```
def Data_matching.for_tree_incremental_learning (
    quantity_basic_tree,
    quantity,
    data )
```

function to do test for tree with one incremental learning

Parameters

<i>quantity_basic_tree</i>	- quantity of data used to tarin basic tree
<i>quantity</i>	- quantity of data used to incremental learning

Returns

arrange - arrange for incremental learning

matrix - confusion matrix

4.5 Incremental_learning Namespace Reference**Functions**

- def [find_tree](#) (tree, data)
function to find the same tree in old tree
- def [incremental_learning](#) (data, tree)
function for incremental learning

4.5.1 Function Documentation**4.5.1.1 find_tree()**

```
def Incremental_learning.find_tree (
    tree,
    data )
```

function to find the same tree in old tree

Parameters

<i>tree</i>	- old tree
<i>data</i>	current data

4.5.1.2 incremental_learning()

```
def Incremental_learning.incremental_learning (
    data,
    tree )
```

function for incremental learning

Parameters

<i>tree</i>	- old tree
<i>data</i>	current data

Returns

object Subtree_Values for knot
object Quantity for knot

4.6 Mesure Namespace Reference

Functions

- def [count](#) (training_elements)
function to count elements in all category (primary, secondary, ...)
- def [giny](#) (training_data)
function to calculate the gini coefficient
- def [gain](#) (false, true, current)
function to calculate the information gain

4.6.1 Function Documentation

4.6.1.1 count()

```
def Mesure.count (
    training_elements )
```

function to count elements in all category (primary, secondary, ...)

Parameters

<i>training_elements</i>	list in which items will be counted
--------------------------	-------------------------------------

Returns

count_data dictionary with quantity of elements in all category

4.6.1.2 gain()

```
def Measure.gain (
    false,
    true,
    current )
```

function to calculate the information gain

Parameters

<i>false</i>	list of false elements in which the information gain will be counted
<i>true</i>	list of true elements in which the information gain will be counted
<i>current</i>	all list of true elements in which the information gain will be counted

Returns

info_gain information gain for current split

4.6.1.3 giny()

```
def Measure.giny (
    training_data )
```

function to calculate the gini coefficient

Parameters

<i>training_elements</i>	list in which items will be counted
--------------------------	-------------------------------------

Returns

1 - giny_tmp gini coefficient

4.7 Print_tree Namespace Reference

Functions

- def `print_tree` (element, space="")
function to print tree

4.7.1 Function Documentation

4.7.1.1 print_tree()

```
def Print_tree.print_tree (
    element,
    space = " " )
```

function to print tree

Parameters

<i>element</i>	tree
<i>space</i>	

4.8 Split Namespace Reference

Functions

- def `make_split` (training_data)
function to find the best split
- def `check_split` (training_data, question_split)
function to do split for only one question

4.8.1 Function Documentation

4.8.1.1 check_split()

```
def Split.check_split (
    training_data,
    question_split )
```

function to do split for only one question

Parameters

<i>training_elements</i>	list in which items will be split
--------------------------	-----------------------------------

Returns

best_gain_value the best find gain
 best_question_split the best find question to split
 best_true_data the best find list with true data
 best_false_data the best find list with false data

4.8.1.2 make_split()

```
def Split.make_split (
    training_data )
```

function to find the best split

Parameters

<i>training_elements</i>	list in which items will be split
--------------------------	-----------------------------------

Returns

best_gain_value the best find gain
 best_question_split the best find question to split
 best_true_data the best find list with true data
 best_false_data the best find list with false data

4.9 Test Namespace Reference**Functions**

- def [write_file](#) (name, arrange)
- def [write_time](#) (name, time)
- def [write_dict](#) (matrix, name)
- def [data_matching_for_basic_tree](#) (quantity)
function for sending tests
- def [data_matching_for_tree_incremental_learning](#) (quantity_basic_tree, quantity, quantity_of_all)
function for sending tests
- def [test](#) ()
test management function

4.9.1 Function Documentation

4.9.1.1 `data_matching_for_basic_tree()`

```
def Test.data_matching_for_basic_tree (
    quantity )
```

function for sending tests

Parameters

<i>quantity</i>	- quantity of data used to tarin dree
-----------------	---------------------------------------

Returns

arrange - basic tree arrange

4.9.1.2 `data_matching_for_tree_incremental_learning()`

```
def Test.data_matching_for_tree_incremental_learning (
    quantity_basic_tree,
    quantity,
    quantity_of_all )
```

function for sending tests

Parameters

<i>quantity_basic_tree</i>	- quantity of data used to tarin basic tree
<i>quantity</i>	- quantity of data used to incremental learning

Returns

arrange - arrange for incremental learning

4.9.1.3 `test()`

```
def Test.test ( )
```

test management function

4.9.1.4 write_dict()

```
def Test.write_dict (
    matrix,
    name )
```

4.9.1.5 write_file()

```
def Test.write_file (
    name,
    arrange )
```

4.9.1.6 write_time()

```
def Test.write_time (
    name,
    time )
```

Chapter 5

Class Documentation

5.1 Build.Quantity Class Reference

class with a dictionary in which the number of elements

Public Member Functions

- `def __init__ (self, data)`
save information about number of elements

Public Attributes

- `quantity`
number of elements

5.1.1 Detailed Description

class with a dictionary in which the number of elements

5.1.2 Constructor & Destructor Documentation

5.1.2.1 `__init__()`

```
def Build.Quantity.__init__ (  
    self,  
    data )
```

save information about number of elements

5.1.3 Member Data Documentation

5.1.3.1 quantity

`Build.Quantity.quantity`

number of elements

The documentation for this class was generated from the following file:

- [Build.py](#)

5.2 Build.Subtree_Values Class Reference

class with elemnet to save in all knots

Public Member Functions

- `def __init__` (self, [question](#), [right_next](#), [left_next](#), [gain](#), [true_data](#), [false_data](#))
save information about knots

Public Attributes

- [question](#)
question used to divide data
- [right_next](#)
next right knots
- [left_next](#)
next left knots
- [gain](#)
gain of information obtained
- [true_data](#)
list with true data - that met the query
- [false_data](#)
list with false data - which did not match the query

5.2.1 Detailed Description

class with elemnet to save in all knots

5.2.2 Constructor & Destructor Documentation

5.2.2.1 `__init__()`

```
def Build.Subtree_Values.__init__ (
    self,
    question,
    right_next,
    left_next,
    gain,
    true_data,
    false_data )
```

save information about knots

5.2.3 Member Data Documentation

5.2.3.1 `false_data`

`Build.Subtree_Values.false_data`

list with false data - which did not match the query

5.2.3.2 `gain`

`Build.Subtree_Values.gain`

gain of information obtained

5.2.3.3 `left_next`

`Build.Subtree_Values.left_next`

next left knots

5.2.3.4 `question`

`Build.Subtree_Values.question`

question used to divide data

5.2.3.5 right_next

`Build.Subtree_Values.right_next`

next right knots

5.2.3.6 true_data

`Build.Subtree_Values.true_data`

list with true data - that met the query

The documentation for this class was generated from the following file:

- [Build.py](#)

Chapter 6

File Documentation

6.1 Build.py File Reference

Classes

- class [Build.Subtree_Values](#)
class with elemnet to save in all knots
- class [Build.Quantity](#)
class with a dictionary in which the number of elements

Namespaces

- [Build](#)

Functions

- def [Build.count_all](#) (training_data)
function to count all elements
- def [Build.build_tree](#) (training_data)
function to build tree

6.2 Correctness_of_building.py File Reference

Namespaces

- [Correctness_of_building](#)

Functions

- def [Correctness_of_building.correctness_of_building](#) (start, quantity)
function to build and print tree
- def [Correctness_of_building.correctness_of_incremental](#) (start, quantity, tree)
function to incremental learning and print tree

6.3 Data.py File Reference

Namespaces

- [Data](#)

Functions

- def [Data.read_data](#) ()
function to read and optimization data y
- def [Data.read_data_second](#) ()
function to read and optimization data agaricus-lepiota
- def [Data.read_data_third](#) ()
function to read and optimization data iris

6.4 Data_matching.py File Reference

Namespaces

- [Data_matching](#)

Functions

- def [Data_matching.for_basic_tree](#) (quantity, data)
function to do test for basic tree
- def [Data_matching.for_tree_incremental_learning](#) (quantity_basic_tree, quantity, data)
function to do test for tree with one incremental learning
- def [Data_matching.confusion_matrix](#) (data_test, tree)
function for creating confusion matrix
- def [Data_matching.data_matching](#) (tree, data_test)
function for calculating test data matching
- def [Data_matching.find](#) (tree, dt)
find the list for test data

6.5 Incremental_learning.py File Reference

Namespaces

- [Incremental_learning](#)

Functions

- def [Incremental_learning.find_tree](#) (tree, data)
function to find the same tree in old tree
- def [Incremental_learning.incremental_learning](#) (data, tree)
function for incremental learning

6.6 Measure.py File Reference

Namespaces

- [Measure](#)

Functions

- def [Measure.count](#) (training_elements)
function to count elements in all category (primary, secondary, ...)
- def [Measure.giny](#) (training_data)
function to calculate the gini coefficient
- def [Measure.gain](#) (false, true, current)
function to calculate the information gain

6.7 Print_tree.py File Reference

Namespaces

- [Print_tree](#)

Functions

- def [Print_tree.print_tree](#) (element, space="")
function to print tree

6.8 Split.py File Reference

Namespaces

- [Split](#)

Functions

- def [Split.make_split](#) (training_data)
function to find the best split
- def [Split.check_split](#) (training_data, question_split)
function to do split for only one question

6.9 Test.py File Reference

Namespaces

- [Test](#)

Functions

- def [Test.write_file](#) (name, arrange)
- def [Test.write_time](#) (name, time)
- def [Test.write_dict](#) (matrix, name)
- def [Test.data_matching_for_basic_tree](#) (quantity)
function for sending tests
- def [Test.data_matching_for_tree_incremental_learning](#) (quantity_basic_tree, quantity, quantity_of_all)
function for sending tests
- def [Test.test](#) ()
test management function

Index

- `__init__`
 - Build.Quantity, [19](#)
 - Build.Subtree_Values, [20](#)
- Build, [7](#)
 - build_tree, [7](#)
 - count_all, [8](#)
- Build.py, [23](#)
- Build.Quantity, [19](#)
 - `__init__`, [19](#)
 - quantity, [20](#)
- Build.Subtree_Values, [20](#)
 - `__init__`, [20](#)
 - false_data, [21](#)
 - gain, [21](#)
 - left_next, [21](#)
 - question, [21](#)
 - right_next, [21](#)
 - true_data, [22](#)
- build_tree
 - Build, [7](#)
- check_split
 - Split, [15](#)
- confusion_matrix
 - Data_matching, [10](#)
- Correctness_of_building, [8](#)
 - correctness_of_building, [8](#)
 - correctness_of_incremental, [9](#)
- correctness_of_building
 - Correctness_of_building, [8](#)
- Correctness_of_building.py, [23](#)
- correctness_of_incremental
 - Correctness_of_building, [9](#)
- count
 - Mesure, [13](#)
- count_all
 - Build, [8](#)
- Data, [9](#)
 - read_data, [9](#)
 - read_data_second, [9](#)
 - read_data_third, [10](#)
- Data.py, [24](#)
- Data_matching, [10](#)
 - confusion_matrix, [10](#)
 - data_matching, [11](#)
 - find, [11](#)
 - for_basic_tree, [11](#)
 - for_tree_incremental_learning, [12](#)
- data_matching
 - Data_matching, [11](#)
 - Data_matching.py, [24](#)
 - data_matching_for_basic_tree
 - Test, [17](#)
 - data_matching_for_tree_incremental_learning
 - Test, [17](#)
- false_data
 - Build.Subtree_Values, [21](#)
- find
 - Data_matching, [11](#)
- find_tree
 - Incremental_learning, [12](#)
- for_basic_tree
 - Data_matching, [11](#)
- for_tree_incremental_learning
 - Data_matching, [12](#)
- gain
 - Build.Subtree_Values, [21](#)
 - Mesure, [14](#)
- giny
 - Mesure, [14](#)
- Incremental_learning, [12](#)
 - find_tree, [12](#)
 - incremental_learning, [13](#)
- incremental_learning
 - Incremental_learning, [13](#)
- Incremental_learning.py, [24](#)
- left_next
 - Build.Subtree_Values, [21](#)
- make_split
 - Split, [16](#)
- Mesure, [13](#)
 - count, [13](#)
 - gain, [14](#)
 - giny, [14](#)
- Mesure.py, [25](#)
- Print_tree, [15](#)
 - print_tree, [15](#)
- print_tree
 - Print_tree, [15](#)
- Print_tree.py, [25](#)
- quantity
 - Build.Quantity, [20](#)

- question
 - Build.Subtree_Values, [21](#)
- read_data
 - Data, [9](#)
- read_data_second
 - Data, [9](#)
- read_data_third
 - Data, [10](#)
- right_next
 - Build.Subtree_Values, [21](#)
- Split, [15](#)
 - check_split, [15](#)
 - make_split, [16](#)
- Split.py, [25](#)
- Test, [16](#)
 - data_matching_for_basic_tree, [17](#)
 - data_matching_for_tree_incremental_learning, [17](#)
 - test, [17](#)
 - write_dict, [17](#)
 - write_file, [18](#)
 - write_time, [18](#)
- test
 - Test, [17](#)
- Test.py, [25](#)
- true_data
 - Build.Subtree_Values, [22](#)
- write_dict
 - Test, [17](#)
- write_file
 - Test, [18](#)
- write_time
 - Test, [18](#)