

CSDN新首页上线啦，邀请你来立即体验！(http://blog.csdn.net/)

CSDN

博客 (//blog.csdn.net/?ref=toolbar)学院 (//edu.csdn.net/?ref=toolbar)下载 (//download.csdn.net/?ref=toolbar)GitChat (//gitbook.cn/?ref=csdn)更多 ▾

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0



```
1 read_csv(url, delimiter=";", header=True, error_bad_lines=False, comment_char=","
2         escape_char="\\", double_quote=True, quote_char='"', skip_initial_space=True,
3         column_type_hints=None, na_values=['NA'], line_terminator='\n', usecols=[], nrows=None,
4         skiprows=0, verbose=True, nrows_to_infer=100, **kwargs)
5
6 """
```

读入举例：

```
1 # 离线版
2 sf = tc.SFrame.read_csv('../data.csv')
3 # 在线版
4 >>> url = 'https://static.turi.com/datasets/rating_data_example.csv'
5 >>> sf = SFrame.read_csv(url)
```

导出举例：

```
1 # 第一种
2 sf.export_csv('./1.csv')
3 # 第二种
4 sf.save('./data.csv', format='csv')
```

1.2 json格式数据的读入导出

```

1 # 读入
2 >>> !cat input.json
3 [{"a":1,'b':1}, {'a':2,'b':2}, {'a':3,'b':3}]
4 >>> SFrame.read_json('input.json', orient='records')
5 Columns:
6     a      int
7     b      int
8 Rows: 3
9 Data:
10  +---+---+
11  | a | b |
12  +---+---+
13  | 1 | 1 |
14  | 2 | 2 |
15  | 3 | 3 |
16  +---+---+
17
18 # 导出
19 sf.export_json('./1.json')
20 sf.save('1.json', format='json')

```

其中!cat 有查看功能。



24/10155090)

4

机器学习——Python人工智能教程 (http://download.csdn.net/download/weixin_38311456/10129270)

10



奥迪r8二手



人工智能培训

隐形车衣 真火壁炉 短信验证码接口

OA办公系统 在职研究生取消

望京soho oa系统 app开发报价单

商城系统 开源商城系统 app制作

广告

■ 博主专栏



R的数据操作与清洗

(<http://blog.csdn.net/column/details/13587.html>)

 250448

(<http://blog.csdn.net/column/details/13587.html>)

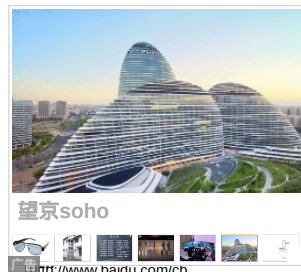


R语言与自然语言处理

(<http://blog.csdn.net/column/details/13670.html>)

149280

(<http://blog.csdn.net/column/details/13670.html>)

[illegible]

1.3 自由格式保存

```
1 sf.save('../mysframedir')
2 sf2 = tc.load_sframe('../mysframedir')
```

有点像R里面的workspace的保存，一股脑保存出去，然后再读进来。

二、SFrame基本操作

2.1 生成SFrame框

借助SArray来生成：

```
1 from turicreate import SArray
2 ids = tc.SArray([1,2,3])
3 vals = tc.SArray(['A','B','C'])
4 #ids = SArray([1,2,3])
5 #vals = SArray(['A','B','C'])
6 sf = SFrame({'id':ids,'val':vals})
7 sf
```

下面也有从dict格式导入、还有从pd.dataframe两种格式导入。

```
1 sf = tc.SFrame({'id':[1,2,3],'val':['A','B','C']}) # 直接导入
2 sf = tc.SFrame(pd.DataFrame({'id':[1,2,3],'val':['A','B','C']})) # pd.dataframe
3 sf
```

2.2 列操作

2.2.1 选中列

通过['name']选中。

```
1 sf = SFrame({'id':[1,2,3],'val':['A','B','C']})
2 sf[['id']]
3 sf[['val','id']]
4 del sf["id"] # 删除列
```

这个跟pd.DataFrame一样

通过一个函数来选择：

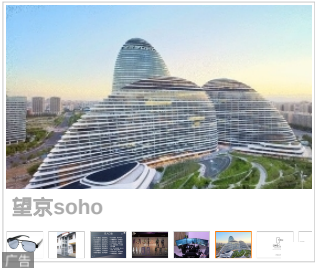
R语言 | 决策树族——随机森林算法 (http://blog.csdn.net/sinat_26917383/article/details/51308061)

R语言 | 文本挖掘——词云wordcloud2包 (http://blog.csdn.net/sinat_26917383/article/details/51620019)

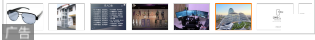
R语言 | 决策树族——随机森林算法 (http://blog.csdn.net/sinat_26917383/article/details/51308061)

R语言数据集合并、数据增减、不等长合并 (http://blog.csdn.net/sinat_26917383/article/details/50676894)

R语言 | 文本挖掘——词云wordcloud2包 (http://blog.csdn.net/sinat_26917383/article/details/51620019)



望京soho



内容举报



返回顶部



内容举报

返回顶部

0

```
1 >>> sf = turicreate.SFrame({'user_id': [1,2,3],
2 ...                         'user_name': ['alice', 'bob', 'charlie'],
3 ...                         'zipcode': [98101, 98102, 98103]
4 ...                         })
5 >>> # This line is equivalent to `sf2 = sf[['user_id', 'zipcode']]`
6 >>> sf2 = sf.select_columns(['user_id', 'zipcode'])
7 >>> sf2
8 +-----+-----+
9 | user_id | zipcode |
10 +-----+-----+
11 | 1 | 98101 |
12 | 2 | 98102 |
13 | 3 | 98103 |
14 +-----+-----+
15 [3 rows x 2 columns]
```



2.2.2 列举列名

```
1 # 找列名以及每一列的type
2 sf = SFrame({'id':[1,2,3], 'val':['A','B','C']})
3 print(sf.column_names())
4 print(sf.column_types())
```

2.2.3 新增列 .add_column ()

```
1 # 新增列
2 >>> sf = turicreate.SFrame({'id': [1, 2, 3], 'val': ['A', 'B', 'C']})
3 >>> sa = turicreate.SArray(['cat', 'dog', 'fossa'])
4 >>> # This line is equivalent to `sf['species'] = sa`
5 >>> res = sf.add_column(sa, 'species')
6 >>> res
7 +----+-----+-----+
8 | id | val | species |
9 +----+-----+-----+
10 | 1 | A | cat |
11 | 2 | B | dog |
12 | 3 | C | fossa |
13 +----+-----+-----+
14 [3 rows x 3 columns]
```

内容举报

返回顶部

2.3 行操作

2.3.1 选中行





```
1 sf[1]
2 sf[1:]
3
4 >>> {'id': 2, 'val': 'B'}
```

切片一样，但是sf[1]代表选中第二行，返回的是dict格式！奇特！

2.3.2 新增行

等同于pd.concat

```
1 # 新增数据：行联合
2 >>> sf = turicreate.SFrame({'id': [4, 6, 8], 'val': ['D', 'F', 'H']})
3 >>> sf2 = turicreate.SFrame({'id': [1, 2, 3], 'val': ['A', 'B', 'C']})
4 >>> sf = sf.append(sf2)
5 >>> sf
6 +----+-----+
7 | id | val |
8 +----+-----+
9 | 4 | D |
10 | 6 | F |
11 | 8 | H |
12 | 1 | A |
13 | 2 | B |
14 | 3 | C |
15 +----+-----+
16 [6 rows x 2 columns]
```

2.4 逻辑选中行列 + 复合选中

```
1 sf[(sf['id'] >= 1) & (sf['id'] <= 2)]
```

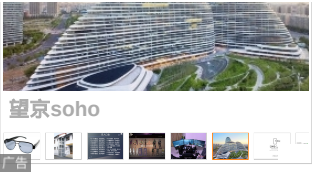
```
1 sf = SFrame({'id':[1,2,3], 'val':['A','B','C'], 'val2':[5,6,7]})
2 sf[2]['id']
```

如何选中'id'列的第三行内容，可以通过这样的方式选中。
(sf.ix[2,1]的方式不可用)

2.5 新增index

SFrame本身不会生成Index，跟pd的不一样。需要额外生成上去。

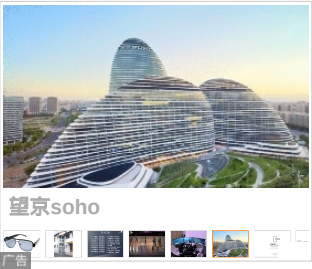
```
1 # 新增Index
2 >>> sf = turicreate.SFrame({'a': [1, None, None], 'b': ['a', 'b', None]})
3 +-----+-----+
4 | a | b |
```



内容举报



返回顶部





```
5 +-----+
6 | 1 | a |
7 | None | b |
8 | None | None |
9 +-----+
10 >>> sf.add_row_number()
11 +-----+
12 | id | a | b |
13 +-----+
14 | 0 | 1 | a |
15 | 1 | None | b |
16 | 2 | None | None |
17 +-----+
18 [3 rows x 3 columns]
```

2.6 缺失值处理

2.6.1 缺失值直接去掉

有以下几种模式：

```
1 # 只保留全部有数字的行
2 >>> sf = turicreate.SFrame({'a': [1, None, None], 'b': ['a', 'b', None]})
3 >>> sf.dropna()
4 +-----+
5 | a | b |
6 +-----+
7 | 1 | a |
8 +-----+
9 [1 rows x 2 columns]
10
11 # 只去掉全部为None的行
12 >>> sf.dropna(any="all")
13 +-----+
14 | a | b |
15 +-----+
16 | 1 | a |
17 | None | b |
18 +-----+
19 [2 rows x 2 columns]
20
21 # 按照a这行进行去缺失值
22 >>> sf.dropna('a', any="all")
23 +-----+
24 | a | b |
25 +-----+
26 | 1 | a |
27 +-----+
28 [1 rows x 2 columns]
```



0



内容举报

返回顶部



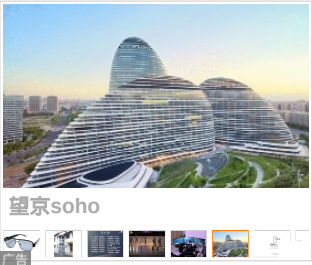
```
29
30 # 把缺失值去掉的也保存下来
31 >>> sf = turicreate.SFrame({'a': [1, None, None], 'b': ['a', 'b', None]})
32 >>> good, bad = sf.dropna_split()
33 >>> good
34 +---+---+
35 | a | b |
36 +---+---+
37 | 1 | a |
38 +---+---+
39 [1 rows x 2 columns]
40 >>> bad
41 +-----+
42 | a | b |
43 +-----+
44 | None | b |
45 | None | None |
46 +-----+
47 [2 rows x 2 columns]
```

2.6.2 缺失值填补

```
1 # 缺失值填补
2 >>> sf = turicreate.SFrame({'a': [1, None, None],
3 ... 'b': ['13.1', '17.2', None]})
4 >>> sf = sf.fillna('a', 0)
5 >>> sf
6 +-----+
7 | a | b |
8 +-----+
9 | 1 | 13.1 |
10 | 0 | 17.2 |
11 | 0 | None |
12 +-----+
13 [3 rows x 2 columns]
```

2.7 数据排序

数据排序有以下几种：



内容举报

返回顶部

内容举报

返回顶部

```

1  # 按某行排序
2  >>> sf = turicreate.SFrame({'a':[1,3,2,1],
3  ...                        'b':['a','c','b','b'],
4  ...                        'c':['x','y','z','y']})
5  >>> sf
6  +---+---+---+
7  | a | b | c |
8  +---+---+---+
9  | 1 | a | x |
10 | 3 | c | y |
11 | 2 | b | z |
12 | 1 | b | y |
13 +---+---+---+
14 [4 rows x 3 columns]
15
16 >>> sf.sort('a')
17 +---+---+---+
18 | a | b | c |
19 +---+---+---+
20 | 1 | a | x |
21 | 1 | b | y |
22 | 2 | b | z |
23 | 3 | c | y |
24 +---+---+---+
25 [4 rows x 3 columns]
26
27 >>> sf.sort('a', ascending = False)
28 +---+---+---+
29 | a | b | c |
30 +---+---+---+
31 | 3 | c | y |
32 | 2 | b | z |
33 | 1 | a | x |
34 | 1 | b | y |
35 +---+---+---+
36 [4 rows x 3 columns]
37
38 >>> sf.sort(['a', 'b'])
39 +---+---+---+
40 | a | b | c |
41 +---+---+---+
42 | 1 | a | x |
43 | 1 | b | y |
44 | 2 | b | z |
45 | 3 | c | y |
46 +---+---+---+
47 [4 rows x 3 columns]

```

2.8 数据合并



内容举报

返回顶部



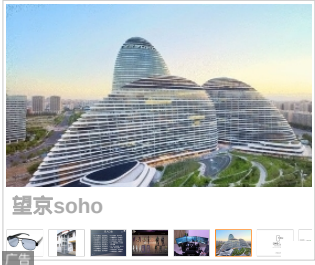
等同于pd.merge



内容举报
返回顶部



```
1 # 数据合并
2 >>> animals = turicreate.SFrame({'id': [1, 2, 3, 4],
3 ...                               'name': ['dog', 'cat', 'sheep', 'cow']})
4 >>> sounds = turicreate.SFrame({'id': [1, 3, 4, 5],
5 ...                             'sound': ['woof', 'baa', 'moo', 'oink']})
6 >>> animals.join(sounds, how='inner')
7 +---+-----+-----+
8 | id | name | sound |
9 +---+-----+-----+
10 | 1 | dog | woof |
11 | 3 | sheep | baa |
12 | 4 | cow | moo |
13 +---+-----+-----+
14 [3 rows x 3 columns]
15
16 >>> animals.join(sounds, on='id', how='left')
17 +---+-----+-----+
18 | id | name | sound |
19 +---+-----+-----+
20 | 1 | dog | woof |
21 | 3 | sheep | baa |
22 | 4 | cow | moo |
```



```

22 | 4 | cow | moo |
23 | 2 | cat | None |
24 +---+-----+
25 [4 rows x 3 columns]
26
27 >>> animals.join(sounds, on='id', how='right')
28 +---+-----+
29 | id | name | sound |
30 +---+-----+
31 | 1 | dog | woof |
32 | 3 | sheep | baa |
33 | 4 | cow | moo |
34 | 5 | None | oink |
35 +---+-----+
36 [4 rows x 3 columns]
37
38 >>> animals.join(sounds, on={'id':'id'}, how='outer')
39 +---+-----+
40 | id | name | sound |
41 +---+-----+
42 | 1 | dog | woof |
43 | 3 | sheep | baa |
44 | 4 | cow | moo |
45 | 5 | None | oink |
46 | 2 | cat | None |
47 +---+-----+
48 [5 rows x 3 columns]

```

2.9 数据去重

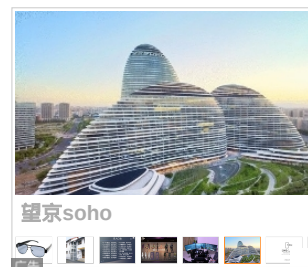
```

1 # 去重
2 >>> sf = turicreate.SFrame({'id':[1,2,3,3,4], 'value':[1,2,3,3,4]})
3 >>> sf
4 +---+-----+
5 | id | value |
6 +---+-----+
7 | 1 | 1 |
8 | 2 | 2 |
9 | 3 | 3 |
10 | 3 | 3 |
11 | 4 | 4 |
12 +---+-----+
13 [5 rows x 2 columns]
14
15 >>> sf.unique()
16 +---+-----+
17 | id | value |
18 +---+-----+
19 | 2 | 2 |
20 | 4 | 4 |
21 | 3 | 3 |
22 | 1 | 1 |

```

内容举报

返回顶部



```
23 +-----+
24 [4 rows x 2 columns]
```

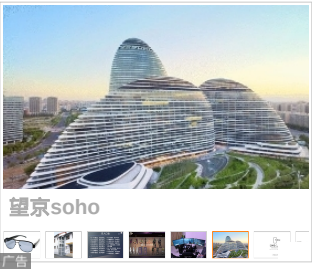
内容举报

返回顶部

三、SFrame数据运行

3.1 使用apply并行

```
1 # 如何使用apply
2 >>> sf = turicreate.SFrame({'user_id': [1, 2, 3], 'movie_id': [3, 3, 6],
3                             'rating': [4, 5, 1]})
4 >>> sf.apply(lambda x: str(x['user_id']) + str(x['movie_id']) + str(x['rating']))
5 dtype: str
6 Rows: 3
7 ['134', '235', '361']
```



3.2 分组汇总sf.groupby

```
1 >>> import turicreate.aggregate as agg
2 >>> url = 'https://static.turi.com/datasets/rating_data_example.csv'
3 >>> sf = turicreate.SFrame.read_csv(url)
```

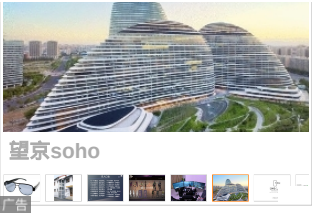


内容举报

返回顶部

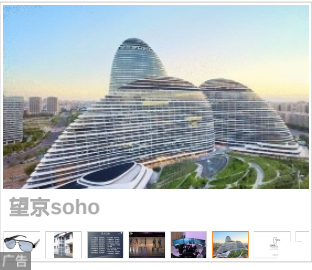


```
4 >>> sf
5 +-----+-----+
6 | user_id | movie_id | rating |
7 +-----+-----+
8 | 25904 | 1663 | 3 |
9 | 25907 | 1663 | 3 |
10 | 25923 | 1663 | 3 |
11 | 25924 | 1663 | 3 |
12 | 25928 | 1663 | 2 |
13 | 25933 | 1663 | 4 |
14 | 25934 | 1663 | 4 |
15 | 25935 | 1663 | 4 |
16 | 25936 | 1663 | 5 |
17 | 25937 | 1663 | 2 |
18 | ... | ... | ... |
19 +-----+-----+
20 [10000 rows x 3 columns]
21
22 # 分组加总
23 >>> user_count = sf.groupby(key_column_names='user_id',
24 ... operations={'count': agg.COUNT()})
25 >>> user_count
26 +-----+-----+
27 | user_id | count |
28 +-----+-----+
29 | 62361 | 1 |
30 | 30727 | 1 |
31 | 40111 | 1 |
32 | 50513 | 1 |
33 | 35140 | 1 |
34 | 42352 | 1 |
35 | 29667 | 1 |
36 | 46242 | 1 |
37 | 58310 | 1 |
38 | 64614 | 1 |
39 | ... | ... |
40 +-----+-----+
41 [9852 rows x 2 columns]
42
43 #
44 >>> user_rating_stats = sf.groupby(key_column_names='user_id',
45 ... operations={
46 ... 'mean_rating': agg.MEAN('rating'),
47 ... 'std_rating': agg.STD('rating')
48 ... })
49 >>> user_rating_stats
50 +-----+-----+
51 | user_id | mean_rating | std_rating |
52 +-----+-----+
53 | 62361 | 5.0 | 0.0 |
54 | 30727 | 4.0 | 0.0 |
55 | 40111 | 2.0 | 0.0 |
56 | 50513 | 4.0 | 0.0 |
```



内容举报

返回顶部



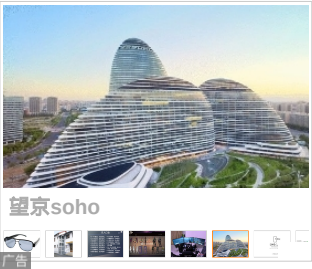


57		35140		4.0		0.0	
58		42352		5.0		0.0	
59		29667		4.0		0.0	
60		46242		5.0		0.0	
61		58310		2.0		0.0	
62		64614		2.0		0.0	
63		
64	+-----+-----+-----+						
65	[9852 rows x 3 columns]						

3.3 数据dataframe框内拆解



1	# 数据dataframe框内拆解						
2	>>> sf = turicreate.SFrame({'topic':[1,2,3,4],						
3	... 'words': [{'a':3, 'cat':2},						
4	... {'a':1, 'the':2},						
5	... {'the':1, 'dog':3},						
6	... {}						
7	... })						
8	+-----+-----+						
9		topic		words			
10	+-----+-----+						
11		1		{ 'a': 3, 'cat': 2 }			
12		2		{ 'a': 1, 'the': 2 }			
13		3		{ 'the': 1, 'dog': 3 }			
14		4		{ }			
15	+-----+-----+						
16	[4 rows x 2 columns]						
17							
18	# 一个拆成两列						
19	>>> sf.stack('words', new_column_name=['word', 'count'])						
20	+-----+-----+						
21		topic		word		count	
22	+-----+-----+						
23		1		a		3	
24		1		cat		2	
25		2		a		1	
26		2		the		2	
27		3		the		1	



```
27 | 3 | true | 1 |
28 | 3 | dog | 3 |
29 | 4 | None | None |
30 +-----+-----+
31 [7 rows x 3 columns]
32
33 # 拆成一列
34 >>> sf = turicreate.SFrame({'topic':[1,2,3],
35 ...                        'friends':[[2,3,4], [5,6],
36 ...                                   [4,5,10,None]]
37 ...                        })
38 >>> sf
39 +-----+-----+
40 | topic | friends |
41 +-----+-----+
42 | 1 | [2, 3, 4] |
43 | 2 | [5, 6] |
44 | 3 | [4, 5, 10, None] |
45 +-----+-----+
46 [3 rows x 2 columns]
```

3.4 格式转换

```
1 # 格式转换
2 SFrame.to_dataframe() Convert this SFrame to pandas.DataFrame.
3 SFrame.to_numpy() Converts this SFrame to a numpy array
4 SFrame.to_sql(conn, table_name[, ...]) Convert an SFrame to a single table in a SQL database.
```

版权声明：本文为博主原创文章，转载请注明来源“素质云博客”，谢谢合作！！微信公众号：素质云笔记





(http://my.csdn.net/weixin_35068028)

相关文章推荐

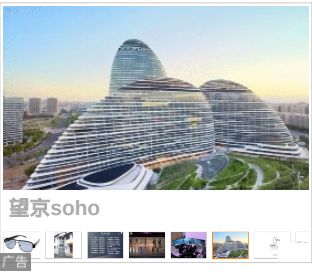
Core ML介绍 (Apple机器学习框架) (<http://blog.csdn.net/chenhaifeng2016/article/details/7...>)

概览 借助 Core ML，您可以将已训练好的机器学习模型，集成到自己的应用当中。 所谓已训练模型 (trained model)，指的是对一组训练数据应用了某个机器学习算法后...

 chenhaifeng2016 (<http://blog.csdn.net/chenhaifeng2016>) 2017年06月08日 11:13  2349

Python与机器学习1——scikit-learn使用的简易框架 (<http://blog.csdn.net/u012824853/articl...>)

Python与机器学习1——scikit-learn使用框架 本系列博客主要参考《利用Python进行数据分析》、《Python数据挖掘入门与实践》、《机器学习》（周志华）。以后两本为主线学习。 第...




 内容举报


 返回顶部

 内容举报

 返回顶部



u012824853 (http://blog.csdn.net/u012824853)2017年03月09日 20:19326



2017年前端报告：程序员薪酬上涨70%！

前端程序员的薪酬曝光，2017年，平均上涨70%，月薪20的人最为常见！以下为详细数据.....

(http://www.baidu.com/cb.php?c=lgF_pyfqHmknj0dP1f0IZ0qnfk9ujYzP1nYPH0k0Aw-5Hc3rHnYnHb0TAq15HfLPWRznb0T1Y4uWl9PAfvm19hujPBnhfz0AwY5HDdnHnYnjcYPHb0lgF_5y9YIZ0IQzq-uZR8mLPbUB48ugfEIAqspynElvNBnHqdlAdxTvqdThP-5yF_UvTkn0KzujYk0AFV5H00TZcq0KdpyfqHRLPjnvnfKEpyfqHc4rj6kP0KWpyfqP1cwrHnz0AqLUWYs0ZK45HcsP



Python机器学习——预测分析核心算法 (http://download.csdn.net/download...

2017年12月12日 12:54 17.27MB 下载





机器学习——Python人工智能教程 (http://download.csdn.net/download/w...

2017年11月22日 22:01 5.55MB 下载

R语言 | 机器学习模型评价指标+（转）模型出错的四大原因及如何纠错 (http://blog.csdn.net/s...

笔者寄语：机器学习中交叉验证的方式是主要的模型评价方法，交叉验证中用到了哪些指标呢？交叉验证将数据分为训练数据集、测试数据集，然后通过训练数据集进行训练，通过测试数据集进行测试，验证集进行验证。 ...

sinat_26917383 (http://blog.csdn.net/sinat_26917383)2016年04月25日 20:037157



一学就会的 WordPress 实战课

学习完本课程可以掌握基本的 WordPress 的开发能力，后续可以根据需要开发适合自己的主题、插件，打造最个性的 WordPress 站点。

(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjfvP1m0IZ0qnfk9ujYzP1f4Pjnz0Aw-5Hc4nj6vPjm0TAq15Hf4rjn1n1b0T1YkuHRzP1fsPHcsnHbkPjn30AwY5HDdnHnYnjcYPWD0lgF_5y9YIZ0IQzqMpgwBUvqoQhP8QviGIAPCmgfEmvq_lyd8Q1N9nHmvnj7hnHPWnjFhPAD1Pyn4uW99ujqdlAdxTvqdThP-5HDknWw9mhcEusKzujYk0AFV5H00TZcq0KdpyfqHRLPjnvnfKEpyfqHnsnj0YnsKWpyfqP1cwrHnz0AqLUWYs0ZK45HcsP6KWThnqPjmvP1D)



Python机器学习——预测分析核心算法 (http://download.csdn.net/download...

2017年10月25日 15:10 22.44MB 下载



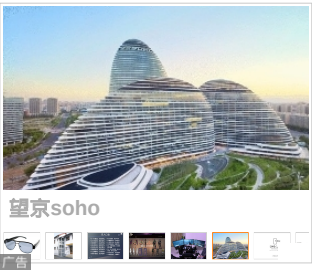
机器学习算法原理与编程实践 郑捷著pdf 下载 (http://download.csdn.net/do...

2017年02月18日 09:38 125.94MB 下载

干货 | 机器学习中防止过拟合的处理方法 (http://blog.csdn.net/GarfieldEr007/article/details/...

我们都知道，在进行数据挖掘或者机器学习模型建立的时候，因为在统计学习中，假设数据满足独立同分布（i.i.d, independently and identically distributed），即当前...

GarfieldEr007 (http://blog.csdn.net/GarfieldEr007)2016年04月01日 11:13591



内容举报

返回顶部





转 | 机器学习算法线上部署方法 (http://blog.csdn.net/sinat_26917383/article/details/548644...)

本文由携程技术中心投递，ID：c triptech。作者：潘鹏举，携程酒店研发BI经理，负责酒店服务相关的业务建模工作，主要研究方向是用机器学习实现业务流程自动化、系统智能化、效率最优化，专注于算法实践...

sinat_26917383 (http://blog.csdn.net/sinat_26917383)2017年02月04日 17:17

893

机器学习 | 非平衡数据处理方式与评估 (http://blog.csdn.net/flyfrommath/article/details/786...)

解决这一问题的基本思路是让正负样本在训练过程中拥有相同的话语权，比如利用采样与加权等方法。为了方便起见，我们把数据集中样本较多的那一类称为"大众类"，样本较少的那一类称为"小众类"。 解决方式分为...

flyfrommath (http://blog.csdn.net/flyfrommath)2017年11月22日 11:50

63

分布式游戏服务器框架sframe（四）—— 消息映射 (http://blog.csdn.net/heyuhang112/articl...)

对于服务器业务开发来说，主要的任务便是处理各种消息——服务间消息、客户端的消息等等。一般的开发模式都是将每一类消息都映射到与其对应的一个处理函数。服务器收到该消息后，找到与其对应的处理函数，再将消息进...

heyuhang112 (http://blog.csdn.net/heyuhang112)2017年03月23日 17:06

270

【机器学习】pandas中Series和DataFrame (http://blog.csdn.net/w417950004/article/detail...)

一，Series 1.Series的定义 Series类似于一个字典，可以通过index参数定义其"key"值。Series使用pandas.Series来定义。如下所示：s = pd.Ser...

w417950004 (http://blog.csdn.net/w417950004)2017年11月27日 17:24

24

Recording | 有价值的各类AI、机器学习比赛心得、经验抄录 (http://blog.csdn.net/sinat_269...)

今年kaggle华人优胜团队很多，所以经验、心得不少，都是干货慢慢收集。一、【干货】Kaggle 数据挖掘比赛经验分享github:https://github.com/ChenglongChen/K...

sinat_26917383 (http://blog.csdn.net/sinat_26917383)2017年07月15日 16:45

1137

R语言 | 常用统计方法包+机器学习包（名称、简介） (http://blog.csdn.net/sinat_26917383/a...)

转载于：http://www.dataguru.cn/thread-116761-1-1.html 时间上有点过期，下面的资料供大家参考 基本的R包已经实现了传统多元统计的很多功能，然而CRNA的许...

sinat_26917383 (http://blog.csdn.net/sinat_26917383)2016年02月11日 15:52

6455

机器学习 | 非平衡数据处理方式与评估 (http://blog.csdn.net/sinat_26917383/article/details/7...)

解决这一问题的基本思路是让正负样本在训练过程中拥有相同的话语权，比如利用采样与加权等方法。为了方便起见，我们把数据集中样本较多的那一类称为"大众类"，样本较少的那一类称为"小众类"。 解决方式分为：...



sinat_26917383 (http://blog.csdn.net/sinat_26917383)2017年07月23日 11:10

1132





R | mlr包挑选最适机器学习模型+变量评估与选择（案例详解）(http://blog.csdn.net/sinat_26...

install.packages("mlr")之后就可以看到R里面有哪些机器学习算法。在哪个包里面。a这个包是听CDA网络课程《R语言与机器学习实战》余文华老师所述，感觉很棒，有待以后深入探讨。以下表...

 sinat_26917383 (http://blog.csdn.net/sinat_26917383) 2016年12月25日 23:10  2988



R | foreach+doParallel并行+联用迭代器优化内存+并行机器学习算法 (http://blog.csdn.net/si...

接着之前写的并行算法parallel包，parallel相比foreach来说，相当于是foreach的进阶版，好多东西封装了。而foreach包更为基础，而且可自定义的内容很多，而且实用性比较强，可...

 sinat_26917383 (http://blog.csdn.net/sinat_26917383) 2016年11月26日 15:11  1710


分布式游戏服务器框架sframe（五）—— 配置管理 (http://blog.csdn.net/heyuhang112/articl...

开发游戏服务器，肯定会有大量的配置数据，什么等级配置、关卡配置等等。对于这些静态数据，我们一般都是采取文件的方式来存储的。策划事先配置好这些数据文件，服务器启动后便将其加载到内存，由配置管理模块统一管...

 heyuhang112 (http://blog.csdn.net/heyuhang112) 2017年03月24日 18:46  406



[人工智能]机器学习的框架偏向于Python原因 (http://blog.csdn.net/BaiHuaXiu123/article/det...

前言主要有以下原因:1. Python是解释语言，这让写程序方便不只一点。 举例来说，在C等编译语言里写一个矩阵乘法，需要自己分配操作数（矩阵）的内存、分配结果的内存、手动对BLAS接口调用gem...

 BaiHuaXiu123 (http://blog.csdn.net/BaiHuaXiu123) 2017年01月11日 22:07  10749

Python 机器学习框架 (http://blog.csdn.net/sinat_30948687/article/details/78732031)

NuPIC: Numenta Platform for Intelligent ComputingNuPIC 是一个基于一种被称为分层式即时记忆（HTM/ Hierarchical Temporal ...

 sinat_30948687 (http://blog.csdn.net/sinat_30948687) 2017年12月06日 16:26  78


内容举报

返回顶部


0





