

レシビ44 Boolean、整数位置、ラベルによる選択(P116)

<ノック>

1.
- movie.csvを読み込み、movie_title をインデックスに
 - content_rating が G かつ imdb_score の点数が4より低い 映画を抽出する
 - 3種類のやり方ができるか？ (Booleanインデックス法、locを使う、ilocを使う)
2.
- 「カラムの型が64ビット整数」の列をlocとilocを使って抽出できるか
3.
- 1の際、カラムを3つに絞った結果を出す (loc,iloc)

- 4章「データからの部分抽出」では、.iloc と .loc インデクサを使って部分的に抽出を行った
 - これらのインデクサは整数位置がラベルで行とカラムを同時に選択できた
 - さらにBooleanインデックス法を使って同じことやってみる
- このレシビでは、.iloc と .loc インデクサを使って、行とカラムをフィルタリングしてみる

(1) movieデータセットを読み込む。題名をインデックスに、内容評価がGでIMDBの点数が4より低い全映画にマッチするBooleanをつくる

```
In [3]: import pandas as pd
movie = pd.read_csv('movie.csv', index_col='movie_title')
c1 = movie['content_rating'] == 'G'
c2 = movie['imdb_score'] < 4
criteria = c1 & c2
```

```
In [4]: # cf.
print(type(criteria))
criteria

<class 'pandas.core.series.Series'>
movie_title
Avatar                                False
Pirates of the Caribbean: At World's End  False
Spectre                                False
The Dark Knight Rises                   False
Star Wars: Episode VII - The Force Awakens  False
...                                     ...
Signed Sealed Delivered                  False
The Following                           False
A Plague So Pleasant                     False
Shanghai Calling                         False
My Date With Drew                        False
Length: 4916, dtype: bool
```

(2) 上記基準をlocに渡し、行をフィルタリング

```
In [4]: movie_loc = movie[criteria]
movie_loc
```

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_2_name	actor_1_facebook_likes	gross	genres
movie_title										
The True Story of Puss'n Boots	Color	J��r��me Deschamps	4.0	80.0	0.0	0.0	Andr�� Wilms	44.0	NaN	Adventure Animation Comedy
Doogal	Color	Dave Borthwick	31.0	77.0	3.0	593.0	Kylie Minogue	787.0	7382993.0	Adventure Animation Comedy Family Fantasy
Thomas and the Magic Railroad	Color	Britt Allcroft	47.0	85.0	2.0	402.0	Colm Feore	1000.0	15911333.0	Adventure Comedy Drama Family Fantasy
Barney's Great Adventure	Color	Steve Gomer	24.0	76.0	9.0	47.0	Kyla Pratt	595.0	11144518.0	Adventure Family
Justin Bieber: Never Say Never	Color	Jon M. Chu	84.0	115.0	209.0	41.0	Sean Kingston	569.0	73000942.0	Documentary Music
Sunday School Musical	Color	Rachel Goldenberg	5.0	93.0	7.0	73.0	Mark Hengst	349.0	NaN	Drama Musical

6 rows x 27 columns

(3) このDataFrameがインデックス演算子で直接作られた場合と同じかどうかをチェックする

```
In [5]: movie_loc.equals(movie[criteria])
```

Out[5]: True

- (4)
- .locインデクサで同じBooleanインデックス法を行ってみると、エラーになる
 - criterialはSeries

```
In [ ]: movie_loc = movie.iloc[criteria]
```

ValueError: iLocation based boolean indexing cannot use an indexable as a mask

- (5)
- インデックスの制約でBooleanのSeriesは直接使えないことがわかった。
 - しかしBooleanの ndarrayであれば使える。配列の抽出にvalues属性を使う

```
In [9]: movie_iloc = movie.iloc[criteria.values]
movie_iloc
```

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_2_name	actor_1_facebook_likes	gross	genres
movie_title										
The True Story of Puss'N Boots	Color	J��r��me Deschamps	4.0	80.0	0.0	0.0	Andr�� Wilms	44.0	NaN	Adventure Animation Comedy
Doogal	Color	Dave Borthwick	31.0	77.0	3.0	593.0	Kylie Minogue	787.0	7382993.0	Adventure Animation Comedy Family Fantasy
Thomas and the Magic Railroad	Color	Britt Allcroft	47.0	85.0	2.0	402.0	Colm Feore	1000.0	15911333.0	Adventure Comedy Drama Family Fantasy
Barney's Great Adventure	Color	Steve Gomer	24.0	76.0	9.0	47.0	Kyla Pratt	595.0	11144518.0	Adventure Family
Justin Bieber: Never Say Never	Color	Jon M. Chu	84.0	115.0	209.0	41.0	Sean Kingston	569.0	73000942.0	Documentary Music
Sunday School Musical	Color	Rachel Goldenberg	5.0	93.0	7.0	73.0	Mark Hengst	349.0	NaN	Drama Musical

6 rows x 27 columns

- (6)
- 普通の使い方ではないが、カラム選択にBooleanインデックス法を使える。
 - 「カラムの型が64ビット整数」を次のように選ぶことができる

```
In [19]: # cf.
movie.dtypes.head()
```

```
Out[19]: color                object
director_name                object
num_critic_for_reviews      float64
duration                    float64
director_facebook_likes     float64
dtype: object
```

- df.dtypesで「カラム⇔型」のSeriesが取得できる

```
In [42]: import numpy as np
# int64型の列をBooleanで取得する
criteria_col = movie.dtypes == np.int64
print(type(criteria_col))
criteria_col
```

```
Out[42]: <class 'pandas.core.series.Series'>
color                False
director_name        False
num_critic_for_reviews  False
duration             False
director_facebook_likes  False
actor_3_facebook_likes  False
actor_2_name         False
actor_1_facebook_likes  False
gross               False
genres              False
actor_1_name         False
num_voted_users      True
cast_total_facebook_likes  True
actor_3_name         False
facenumber_in_poster  False
plot_keywords        False
movie_imdb_link       False
num_user_for_reviews  False
language             False
country              False
content_rating        False
budget              False
title_year           False
actor_2_facebook_likes  False
imdb_score           False
aspect_ratio         False
movie_facebook_likes  True
dtype: bool
```

```
In [24]: # その列のBooleanをセットすれば対象のデータを取得できる
movie.loc[:, criteria_col].head()
```

	num_voted_users	cast_total_facebook_likes	movie_facebook_likes
movie_title			
Avatar	886204	4834	33000
Pirates of the Caribbean: At World's End	471220	48350	0
Spectre	275868	11700	85000
The Dark Knight Rises	1144337	106759	164000
Star Wars: Episode VII - The Force Awakens	8	143	0

- (7)
- criteria_colが「Seriesでインデックスがある」ため
 - .ilocを使うためには基盤 ndarray を使う必要がある
 - 以下 (6) と同じことをやってみる

```
In [29]: movie.iloc[:, criteria_col.values].head()
```

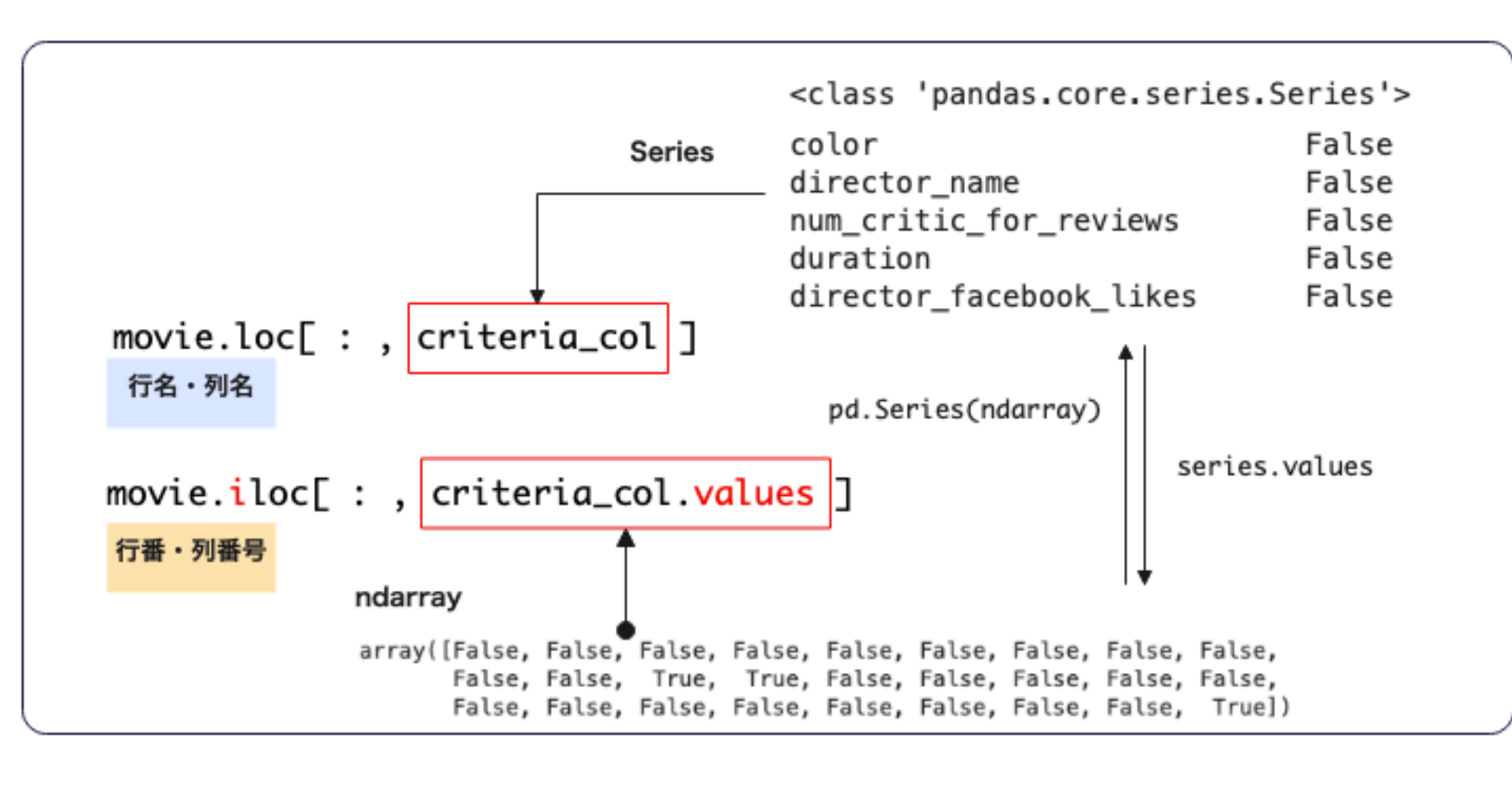
	num_voted_users	cast_total_facebook_likes	movie_facebook_likes
movie_title			
Avatar	886204	4834	33000
Pirates of the Caribbean: At World's End	471220	48350	0
Spectre	275868	11700	85000
The Dark Knight Rises	1144337	106759	164000
Star Wars: Episode VII - The Force Awakens	8	143	0

```
In [31]: # cf.[22]のBoolean情報
criteria_col.values
```

```
Out[31]: array([[False, False, False, False, False, False, False, False, False,
False, False,  True,  True, False, False, False, False, False,
False, False, False, False, False, False, False, False,  True]])
```

少し整理すると

- Booleanインデックス法は
 - .iloc にはそのまま「Series」を渡せないが
 - 「ndarray形式」に変換すれば渡すことできる



- (8)
- BooleanのSeriesを行選択に使い、整数からラベルでカラムを選択する
 - 行基準はcriteriaを使い、列colsを選択する

```
In [33]: cols = ['content_rating', 'imdb_score', 'title_year', 'gross']
movie_loc[criteria, cols].sort_values('imdb_score')
```

	content_rating	imdb_score	title_year	gross
movie_title				
Justin Bieber: Never Say Never	G	1.6	2011.0	73000942.0
Sunday School Musical	G	2.5	2008.0	NaN
Doogal	G	2.8	2006.0	7382993.0
Barney's Great Adventure	G	2.8	1998.0	11144518.0
The True Story of Puss'N Boots	G	2.9	2009.0	NaN
Thomas and the Magic Railroad	G	3.6	2000.0	15911333.0

(9) 同じことを.ilocでもできるが、全カラムの整数位置を取得しなければならない

get_loc

- レシビ29参照

```
In [36]: col_index = [movie.columns.get_loc(col) for col in cols]
col_index
```

Out[36]: [20, 24, 22, 8]

```
In [37]: movie.iloc[criteria.values, col_index]
```

	content_rating	imdb_score	title_year	gross
movie_title				
The True Story of Puss'N Boots	G	2.9	2009.0	NaN
Doogal	G	2.8	2006.0	7382993.0
Thomas and the Magic Railroad	G	3.6	2000.0	15911333.0
Barney's Great Adventure	G	2.8	1998.0	11144518.0
Justin Bieber: Never Say Never	G	1.6	2011.0	73000942.0
Sunday School Musical	G	2.5	2008.0	NaN