

Python

## [Python] Numpy를 통한 정렬하기

2020. 10. 13. 19:01 수정 삭제 공개

# # 정렬 Numpy를 통한 정렬하기

- sort
- axis = 0 (행을 기준으로 정렬)
- axis = 1 (열을 기준으로 정렬)
- inplace = T, F

sort에는 두가지 종류가 있다. np(numpy)를 통한 sort와 해당 값 자체에 대한 sort 가 있다. 값 자체에 하는 sort는 원본 값이 바뀌지만 np를 통한 sort는 결과만을 보여주어서 저장을 하려면 따로 변수로 지정을 해줘야 한다.

## # np(numpy)를 통한 sort

```
In [95]: x = np.array([[4,3,5,7],
                      [1,12,11,9],
                      [2,15,1,14]])
print(np.sort(x))
print('*'*50)
print(np.sort(x,axis=0)) # 각 행에 대한 정렬
print('*'*50)
print(np.sort(x,axis=1)) # 각 열에 대한 정렬

[[ 3  4  5  7]
 [ 1  9 11 12]
 [ 1  2 14 15]]
*****
[[ 1  3  1  7]
 [ 2 12  5  9]
 [ 4 15 11 14]]
*****
[[ 3  4  5  7]
 [ 1  9 11 12]
 [ 1  2 14 15]]
```

np.sort()를 통해 해당 값을 sort한다. axis를 통해 0일 때는 행에 대한 정렬 1일 때는 열에 대한 정렬을 한다.

## # 값에 대한 sort

```
In [99]: x = np.array([[4,3,5,7],
                      [1,12,11,9],
                      [2,15,1,14]])
x.sort(axis=1)
x

Out[99]: array([[ 3,  4,  5,  7],
               [ 1,  9, 11, 12],
               [ 1,  2, 14, 15]])
```

배열 자체에 정렬을 하는 것은 결과가 다르다. 배열 자체에 하는 것은 해당 원본값을 바꿔주고 위에서의 np.sort는 바뀐 값만 보여주고 원본 값은 바뀌지 않는다.

## #argsort

- argsort

```
In [100]: x = np.array([4,3,5,7])
idx = np.argsort(x)
print(idx)
print(x[idx])

[1 0 2 3]
[3 4 5 7]
```

argsort 는 sort에 대한 인덱스 값을 나타내 준다. 즉 오름차순으로 인덱스 값을 가져와서 보여준다.

## 'Python' 카테고리의 다른 글

[Python] Pandas의 이론과 기초적인 사용법

[Python] Numpy를 통한 난수생성, 카운팅, 통계함수 사용법

**[Python] Numpy를 통한 정렬하기**

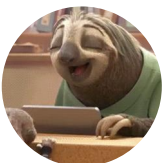
[Python] Numpy 를 통한 최대값, 최소값 , 통계함수 사용하기

[Python] Numpy를 통한 배열 연산

[Python] Numpy의 배열 행 열 삭제

Numpy sort

numpy 정렬



나무늘보스

혼자 끄적끄적하는 블로그 입니다.

