

vncoder.vn

Bài 16: Hàm thống kê

6-8 phút

NumPy có khá nhiều hàm thống kê hữu ích để tìm độ lệch chuẩn và phương sai tối thiểu, tối đa, phần trăm, v.v. từ các phần tử đã cho trong mảng. Các hàm được giải thích như sau:

1. `numpy.amin()` và `numpy.amax()`

Các hàm này trả về giá trị nhỏ nhất và giá trị lớn nhất từ các phần tử trong mảng đã cho dọc theo trục được chỉ định.

Ví dụ :

```
import numpy as np
a = np.array([[3,7,5],[8,4,3],[2,4,9]])

print 'Our array is:'
print a
print '\n'

print 'Applying amin() function:'
print np.amin(a,1)
print '\n'

print 'Applying amin() function again:'
print np.amin(a,0)
print '\n'
```

```
print 'Applying amax() function:'
print np.amax(a)
print '\n'

print 'Applying amax() function again:'
print np.amax(a, axis = 0)
```

Kết quả :

Our array is:

```
[[3 7 5]
 [8 4 3]
 [2 4 9]]
```

Applying amin() function:

```
[3 3 2]
```

Applying amin() function again:

```
[2 4 3]
```

Applying amax() function:

```
9
```

Applying amax() function again:

```
[8 7 9]
```

2. numpy.ptp()

Hàm `numpy.ptp()` trả về phạm vi (tối đa-tối thiểu) của các giá trị dọc theo trục.

```
import numpy as np
a = np.array([[3,7,5],[8,4,3],[2,4,9]])
```

```
print 'Our array is:'
print a
print '\n'

print 'Applying ptp() function:'
print np.ptp(a)
print '\n'

print 'Applying ptp() function along axis 1:'
print np.ptp(a, axis = 1)
print '\n'

print 'Applying ptp() function along axis 0:'
print np.ptp(a, axis = 0)
```

Kết quả :

Our array is:

```
[[3 7 5]
 [8 4 3]
 [2 4 9]]
```

Applying ptp() function:

```
7
```

Applying ptp() function along axis 1:

```
[4 5 7]
```

Applying ptp() function along axis 0:

```
[6 3 6]
```

3. numpy.percentile()

Phần trăm là một thước đo được sử dụng trong thống kê cho

biết giá trị mà dưới đó một phần trăm quan sát nhất định trong một nhóm quan sát giảm xuống. Hàm `numpy.percentile()` nhận các đối số sau.

```
numpy.percentile(a, q, axis)
```

Trong đó :

1. **a** : Mảng đầu vào
2. **q** : Phần trăm để tính toán phải từ 0-100
3. **axis** : Trục dọc theo phần trmaw sẽ được tính

Ví dụ :

```
import numpy as np
a = np.array([[30,40,70],[80,20,10],
              [50,90,60]])

print 'Our array is:'
print a
print '\n'

print 'Applying percentile() function:'
print np.percentile(a,50)
print '\n'

print 'Applying percentile() function along
axis 1:'
print np.percentile(a,50, axis = 1)
print '\n'

print 'Applying percentile() function along
axis 0:'
print np.percentile(a,50, axis = 0)
```

Kết quả :

Our array is:

```
[[30 40 70]
 [80 20 10]
 [50 90 60]]
```

Applying percentile() function:

```
50.0
```

Applying percentile() function along axis 1:

```
[ 40. 20. 60.]
```

Applying percentile() function along axis 0:

```
[ 50. 40. 60.]
```

4. numpy.median()

Giá trị trung bình được định nghĩa là giá trị tách nửa cao hơn của mẫu dữ liệu với nửa dưới. Hàm `numpy.median()` được sử dụng như sau.

Ví dụ :

```
import numpy as np
a = np.array([[30, 65, 70], [80, 95, 10],
              [50, 90, 60]])

print 'Our array is:'
print a
print '\n'

print 'Applying median() function:'
print np.median(a)
```

```
print '\n'

print 'Applying median() function along axis
0:'
print np.median(a, axis = 0)
print '\n'

print 'Applying median() function along axis
1:'
print np.median(a, axis = 1)
```

Kết quả :

Our array is:

```
[[30 65 70]
 [80 95 10]
 [50 90 60]]
```

Applying median() function:

```
65.0
```

Applying median() function along axis 0:

```
[ 50. 90. 60.]
```

Applying median() function along axis 1:

```
[ 65. 80. 60.]
```

5. numpy.mean()

Trung bình là tổng các phần tử dọc theo một trục chia cho tổng số phần tử. Hàm `numpy.mean()` trả về giá trị trung bình cộng của các phần tử trong mảng.

Ví dụ :

```
import numpy as np
a = np.array([[1,2,3],[3,4,5],[4,5,6]])

print 'Our array is:'
print a
print '\n'

print 'Applying mean() function:'
print np.mean(a)
print '\n'

print 'Applying mean() function along axis 0:'
print np.mean(a, axis = 0)
print '\n'

print 'Applying mean() function along axis 1:'
print np.mean(a, axis = 1)
```

Kết quả :

Our array is:

```
[[1 2 3]
 [3 4 5]
 [4 5 6]]
```

Applying mean() function:

```
3.666666666667
```

Applying mean() function along axis 0:

```
[ 2.66666667  3.66666667  4.66666667]
```

Applying mean() function along axis 1:

```
[ 2.  4.  5.]
```

6. numpy.average()

Trung bình cộng là giá trị trung bình kết quả từ việc nhân mỗi thành phần với một yếu tố phản ánh tầm quan trọng của nó.

Hàm `numpy.average()` tính giá trị trung bình có trọng số của các phần tử trong một mảng theo trọng số tương ứng đã cho trong một mảng khác. Hàm có thể có một tham số trục. Nếu trục không được chỉ định, mảng sẽ được làm phẳng.

Xét một mảng `[1,2,3,4]` và các trọng số tương ứng `[4,3,2,1]`, trung bình có trọng số được tính bằng cách cộng tích của các phần tử tương ứng và chia tổng cho tổng các trọng số.

Trung bình có trọng số = $(1 * 4 + 2 * 3 + 3 * 2 + 4 * 1) / (4 + 3 + 2 + 1)$

Ví dụ :

```
import numpy as np
a = np.array([1,2,3,4])

print 'Our array is:'
print a
print '\n'

print 'Applying average() function:'
print np.average(a)
print '\n'

# this is same as mean when weight is not
specified
wts = np.array([4,3,2,1])

print 'Applying average() function again:'
print np.average(a, weights = wts)
```



```
print '\n'

# Returns the sum of weights, if the returned
parameter is set to True.
print 'Sum of weights'
print np.average([1,2,3, 4],weights =
[4,3,2,1], returned = True)
```

Kết quả :

Our array is:

```
[1 2 3 4]
```

Applying average() function:

```
2.5
```

Applying average() function again:

```
2.0
```

Sum of weights

```
(2.0, 10.0)
```

Trong một mảng nhiều chiều, trực để tính toán như sau :

```
import numpy as np
a = np.arange(6).reshape(3,2)

print 'Our array is:'
print a
print '\n'

print 'Modified array:'
wt = np.array([3,5])
print np.average(a, axis = 1, weights = wt)
```

```
print '\n'
```

```
print 'Modified array:'
```

```
print np.average(a, axis = 1, weights = wt,  
returned = True)
```

Kết quả :

Our array is:

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

Modified array:

```
[ 0.625 2.625 4.625]
```

Modified array:

```
(array([ 0.625, 2.625, 4.625]), array([ 8., 8.,  
8.]))
```

7. Độ lệch chuẩn :

Độ lệch chuẩn là căn bậc hai của giá trị trung bình của độ lệch bình phương so với giá trị trung bình. Công thức cho độ lệch chuẩn như sau:

```
std = sqrt(mean(abs(x - x.mean())**2))
```

Nếu mảng là [1, 2, 3, 4], thì giá trị trung bình của nó là 2,5. Do đó, độ lệch bình phương là [2,25, 0,25, 0,25, 2,25] và căn bậc hai của trung bình của nó chia cho 4, tức là $\sqrt{5/4}$ là 1,1180339887498949.

Ví dụ :

```
import numpy as np
```

```
print np.std([1,2,3,4])
```

Kết quả :

8. Phương sai :

Phương sai là giá trị trung bình của độ lệch bình phương, tức là trung bình $(\text{abs}(x - x.\text{mean}()) ** 2)$. Nói cách khác, độ lệch chuẩn là căn bậc hai của phương sai.

Ví dụ :

```
import numpy as np
print np.var([1,2,3,4])
```

Kết quả :