numpy-2-del

January 28, 2024

0.1 Indeksiranje z maskami

Imejmo tabelo

[3]:
```
import numpy as np

a = np.array([5, 8, 3, 1, 10, 5])
```

Pripravimo "masko": tabelo, ki je enako dolga kot `a` in vsebuje `bool`-e.

[4]:
```
m = np.array([True, True, True, False, False, True])
```

Z masko `m` lahko izbiramo elemente `a`, tako da uporabimo `m` kot indeks.

[5]:
```
a[m]
```

[5]:
```
array([5, 8, 3, 5])
```

To nam vrne tiste elemente `a`-ja, pri katerih je istoležni element `m`-ja enak `True`.

To je zelo koristna reč. Tako lahko, recimo, dobimo vse elemente `a`, ki so večji od 5. Pripravimo primerno masko

[6]:
```
m = a > 5

m
```

[6]:
```
array([False,  True, False, False,  True, False])
```

in jo uporabimo:

[7]:
```
a[m]
```

[7]:
```
array([ 8, 10])
```

To seveda navadno naredimo brez posebne spremenljivke, napišemo lahko kar:

[8]:
```
a[a > 5]
```

[8]:
```
array([ 8, 10])
```

0.2 Maske z večdimenzionalnimi tabelami

Recimo, da imamo

[9]:
```
a = np.array([[5, 8, 7, 1, 3],
              [2, 5, 1, 1, 4],
              [7, 6, 1, 9, 2]])
```

Kako bi dobili vse vrstice tabele, ki v stolpcu 1 vsebujejo število manjše od 7?

Zanima nas drugi stolpec (in to vse vrstice tega stolpca), torej

[10]:
```
a[:, 1]
```

[10]:
```
array([8, 5, 6])
```

Konkretno, vedeti hočemo, ali je tam število, ki je manjše od 7.

[11]:
```
a[:, 1] < 7
```

[11]:
```
array([False,  True,  True])
```

In to bomo uporabili za izbor vrstic:

[12]:
```
a[a[:, 1] < 7]
```

[12]:
```
array([[2, 5, 1, 1, 4],
       [7, 6, 1, 9, 2]])
```

Nekaj podobnega boste delali ob reševanju naloge.

Tu pa si oglejmo še nekaj malo bolj zapletenega. Hočemo vrstice, katerih vsota je večja od 20. Vsote vrstic dobimo z

[13]:
```
np.sum(a, axis=1)
```

[13]:
```
array([24, 13, 25])
```

Zanimajo nas vsote, večje od 20, torej

[14]:
```
np.sum(a, axis=1) > 20
```

[14]:
```
array([ True, False,  True])
```

Vidimo, prva in tretja vrstica. Torej

[15]:
```
a[np.sum(a, axis=1) > 20]
```

[15]:
```
array([[5, 8, 7, 1, 3],
       [7, 6, 1, 9, 2]])
```

Kaj pa stolpci, katerih vsota je večja od deset?

[16]:
```
a[:, np.sum(a, axis=0) > 10]
```

[16]:
```
array([[5, 8, 1],
       [2, 5, 1],
       [7, 6, 9]])
```

Ne spreglejte: `np.sum(a, axis=1)` je potrebno uporabiti kot drugi indeks. Prvi indeks pa je `:`, saj želimo vse vrstice; izbir(č)ni smo le glede stolpcev.

0.3 Naloga

Zdaj znamo dovolj, da učinkovito rešimo še drugi del tretje naloge Binary Diagnostic. Tu žal ne bo šlo tako, da bi računali za vse stolpce hkrati. Potrebno bo narediti zanko, s katero bomo šli prek stolpcev in izbirali vrstice, ki ustrezajo kriteriju, dokler ne ostane ena sama vrstica.