

习题解答2.1 ndarray 快速入门

这里是2.1节的习题。我们先在这里 import 所有需要用到的库。

```
import numpy as np
import pandas as pd

arr1 = np.load("data/1-task-arr1.npy")
arr2 = np.load("data/1-task-arr2.npy")

iris_df = pd.read_csv("data/iris.csv")
```

基础练习

1. 请创建形状为 (3, 4, 5) 的全0, 全1, 全2 数组.

```
shape = (3, 4, 5)
arr_0 = np.zeros(shape) # 全0 数组
arr_1 = np.ones(shape) # 全1 数组
arr_2 = np.full(shape, 2) # 全2 数组
```

2. 请创建长度为16的一维数组, 其中数据为 24,22,20,18...,-6.

```
arr = np.arange(24, -8, -2)
arr
```

```
array([24, 22, 20, 18, 16, 14, 12, 10,  8,  6,  4,  2,  0, -2, -4, -6])
```

3. 请获取数组 arr1 的维度、形状与数据类型.

```
print("维度:", arr1.ndim)
print("形状:", arr1.shape)
print("数据类型:", arr1.dtype)
```

```
维度: 5
形状: (1, 2, 3, 4, 5)
数据类型: datetime64[s]
```

4. 请获取 arr2 的简略字符串与详细字符串表示.

```
print("简略字符串:", str(arr2))
print("详细字符串:\n", repr(arr2))
```

简略字符串: `[[1. 2. 3.]`
`[4. 5. 6.]]`
详细字符串:
`array([[1., 2., 3.],`
`[4., 5., 6.]])`

5. 请计算将 `arr2` 中的每个元素加100的平方根后的数组, 并求出它的最大值、均值与标准差.

```
arr = np.sqrt(arr2 + 100)
print("最大值:", np.max(arr))
print("均值:", np.mean(arr))
print("标准差:", np.std(arr))
arr
```

最大值: 10.295630140987
均值: 10.173148676451225
标准差: 0.08394049570182746

```
array([[10.04987562, 10.09950494, 10.14889157],
       [10.19803903, 10.24695077, 10.29563014]])
```

6. 请遍历 `arr2` 中的所有元素并存储到一个集合 `set` 中.

```
result = set()
for i in arr2.flat:
    result.add(i)
result
```

```
{np.float64(1.0),
 np.float64(2.0),
 np.float64(3.0),
 np.float64(4.0),
 np.float64(5.0),
 np.float64(6.0)}
```

7. 请获取一个数值为1~9,形状为 (3, 3)的数组, 将其存储至 `arr3` 中.

```
arr3 = np.arange(1,10).reshape(3,3)
arr3
```

```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

8. 请计算 arr2 与 arr3 在第一个轴上的堆叠, arr2 在 arr3 之前.

```
np.concatenate((arr2, arr3), axis=0)
```

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

实战演练

这里我们使用了经典的鸢尾花数据集 (Iris Dataset)。已知数组 iris_arr 存储的是 iris_df 的特征, 即其中第2-5列的数据(按照顺序), 数组 iris_index 存储的是这四列的名称, iris_target 存储的是目标特征, 即 species 列中的数据。请完成下述问题。

```
iris_arr = iris_df.to_numpy()[ :, 1:5 ].astype(np.float64)
iris_index = iris_df.columns[1:5].to_list().copy()
iris_target = iris_df.to_numpy()[ :, 5 ].copy()
iris_df.head()
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	Unnamed: 0	sepal_length	sepal_width	petal_length	petal_width	species
0	0	5.1	3.5	1.4	0.2	Iris-setosa
1	1	4.9	3.0	1.4	0.2	Iris-setosa
2	2	4.7	3.2	1.3	0.2	Iris-setosa

	Unnamed: 0	sepal_length	sepal_width	petal_length	petal_width	species
3	3	4.6	3.1	1.5	0.2	Iris-setosa
4	4	5.0	3.6	1.4	0.2	Iris-setosa

1. 请计算每个特征的均值和方差

```
means = np.mean(iris_arr, axis=0)
variances = np.var(iris_arr, axis=0)

for col_name, mean, variance in zip(iris_index, means, variances):
    print(f"特征{col_name}的均值为:", mean)
    print(f"特征{col_name}的方差为:", variance)
```

```
特征sepal_length的均值为: 5.843333333333334
特征sepal_length的方差为: 0.6811222222222223
特征sepal_width的均值为: 3.0540000000000003
特征sepal_width的方差为: 0.18675066666666668
特征petal_length的均值为: 3.7586666666666666
特征petal_length的方差为: 3.0924248888888889
特征petal_width的均值为: 1.1986666666666668
特征petal_width的方差为: 0.5785315555555555
```

2. 请将 iris 数据标准化 (均值为0, 方差为1)

```
iris_arr = (iris_arr - means) / np.sqrt(variances)
iris_df.iloc[:, 1:5] = iris_arr
iris_df.head()
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	Unnamed: 0	sepal_length	sepal_width	petal_length	petal_width	species
0	0	-0.900681	1.032057	-1.341272	-1.312977	Iris-setosa
1	1	-1.143017	-0.124958	-1.341272	-1.312977	Iris-setosa
2	2	-1.385353	0.337848	-1.398138	-1.312977	Iris-setosa

	Unnamed: 0	sepal_length	sepal_width	petal_length	petal_width	species
3	3	-1.506521	0.106445	-1.284407	-1.312977	Iris-setosa
4	4	-1.021849	1.263460	-1.341272	-1.312977	Iris-setosa

3. 请计算每个类别的平均特征值

```
unique_classes = np.unique(iris_target)
class_means = {}

for cls in unique_classes:
    class_data = iris_arr[iris_target == cls]
    class_means[cls] = np.mean(class_data, axis=0)

print("每个类别的平均特征值:")
for class_name, mean_values in class_means.items():
    print(f"{class_name}: {mean_values}")
```

每个类别的平均特征值:

```
Iris-setosa: [-1.01457897  0.84230679 -1.30487835 -1.25512862]
Iris-versicolor: [ 0.11228223 -0.65718442  0.28508673  0.16740892]
Iris-virginica: [ 0.90229674 -0.18512237  1.01979162  1.08771971]
```