

계층적 군집 (4) - 다른 유형의 데이터 (MNIST 데이터셋)

#01. 패키지 참조

```
import warnings
warnings.filterwarnings('ignore')

import seaborn as sb
import numpy as np
from matplotlib import pyplot as plt
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.datasets import load_digits
```

#02. 데이터 가져오기

전체 데이터셋 로딩

```
digits = load_digits()
#digits
```

무작위 50개 추출

추출할 이미지 수, 랜덤시드 고정

```
rnd_count = 50
np.random.seed(777)
```

무작위 50개에 대한 인덱스 생성

```
idx = np.random.choice(range(len(digits.data)), rnd_count)
idx
```

```
array([ 103,  815, 1595,  934, 1623,   71, 1693,  639,  116, 1447,  985,
        814, 1530,  792, 1319,  397,  654,   32,  321,  850,  340, 1349,
        577, 1567, 1211, 1127, 1098,  850, 1586, 1342,  967,  365,  836,
        824, 1350,  322,  202,   60,  967,  888, 1664,  819,  186,  642,
       1040,  355,  237, 1651,  171,  384])
```

무작위 인덱스에 대한 표본 추출

```
x = digits.data[idx]
x.shape
```

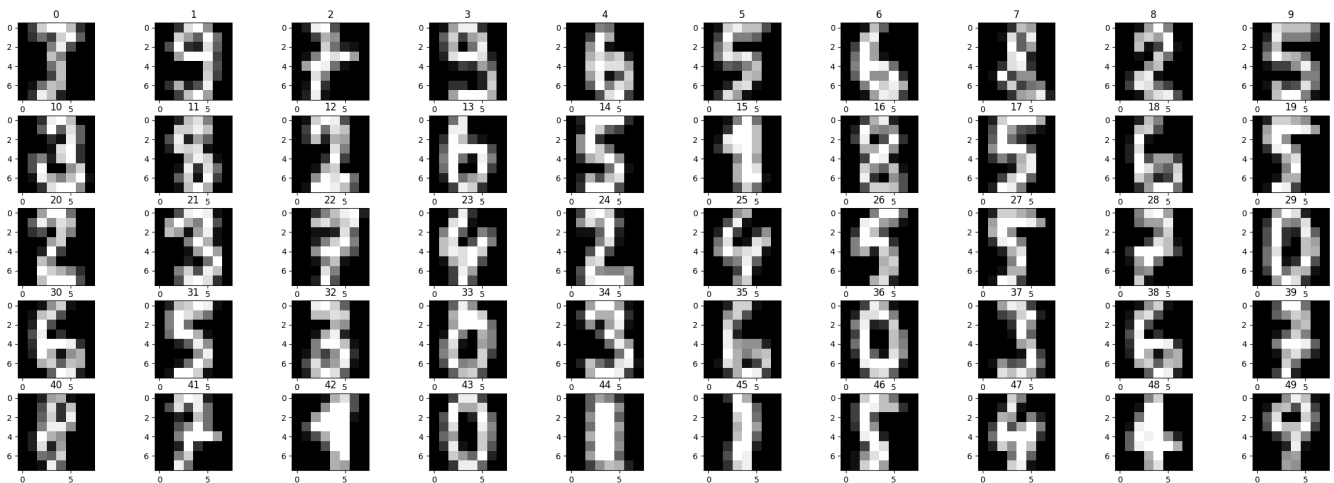
(50, 64)

```
images = digits.images[idx]
images.shape
```

(50, 8, 8)

```
fig, ax = plt.subplots(5, 10, figsize=(30, 10))

idx = 0
for i in range(0, 5):
    for j in range(0, 10):
        #ax[i][j]
        ax[i][j].imshow(images[idx], cmap='gray')
        ax[i][j].set_title(str(idx))
        idx += 1
```



#03. 군집 수행

```
from matplotlib.offsetbox import OffsetImage, AnnotationBbox
```

```
lnk = linkage(x, 'ward')

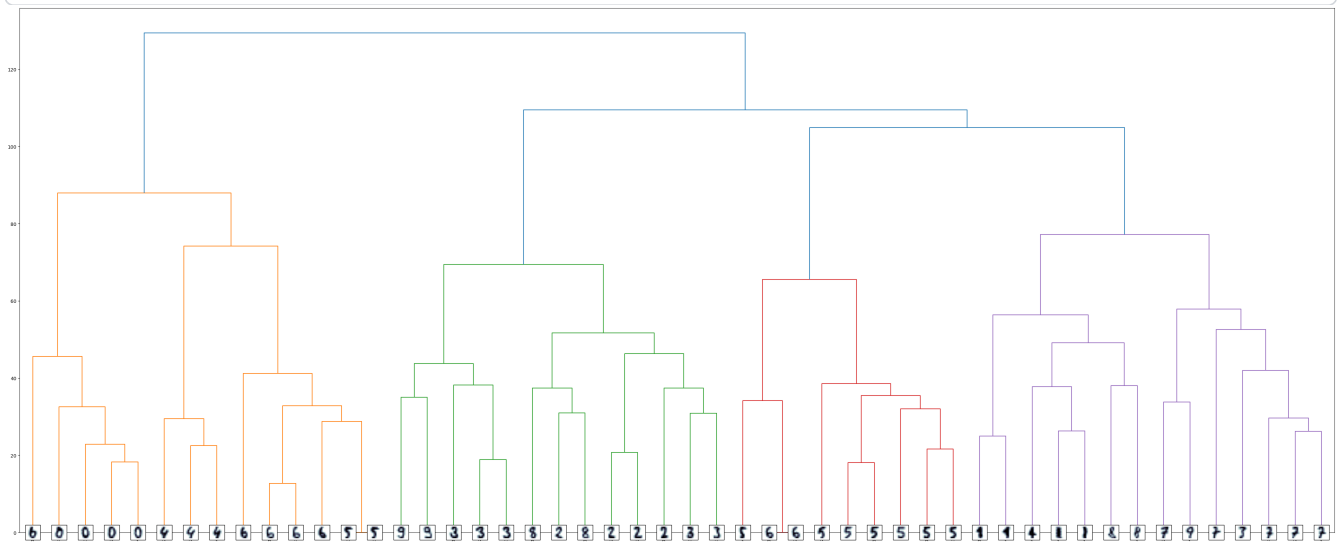
fig, ax = plt.subplots(1, 1, figsize=(50, 20))

r = dendrogram(lnk)

dcoord = np.array(r["dcoord"])
icoord = np.array(r["icoord"])
leaves = np.array(r["leaves"])
idx = np.argsort(dcoord[:, 2])
dcoord = dcoord[idx, :]
icoord = icoord[idx, :]
idx = np.argsort(lnk[:, :2].ravel())
label_pos = icoord[:, 1:3].ravel()[idx][:rnd_count]
```

```
for i in range(rnd_count):
    imagebox = OffsetImage(images[i], cmap=plt.cm.bone_r, interpolation="bilinear", zoom=1.5)
    ab = AnnotationBbox(imagebox, (label_pos[i], 0))
    ax.add_artist(ab)

plt.show()
```



색상이름 목록

k 번째의 항목은 k 번째 링크의 색상

```
print(r['color_list'])
```

```
['C1', 'C1', 'C1', 'C1', 'C1', 'C1', 'C1', 'C1', 'C1', 'C1', 'C1', 'C1', 'C1', 'C2',
```

```
print(r['icoord'])
```

```
[[35.0, 35.0, 45.0, 45.0], [25.0, 25.0, 40.0, 40.0], [15.0, 15.0, 32.5, 32.5], [5.0, 5.0,
```

```
print(r['dcoord'])
```

```
[[0.0, 18.303005217723125, 18.303005217723125, 0.0], [0.0, 22.854612955229264, 22.854612955229264,
```

```
print(r['leaves'])
```

```
[13, 33, 36, 29, 43, 23, 25, 47, 4, 18, 35, 6, 30, 38, 1, 3, 10, 21, 34, 11, 8, 16, 20,
```