

```
1 # -*- coding: utf-8 -*-
2 from sklearn import datasets
3 from sklearn import decomposition
4 import matplotlib.pyplot as plt
5 from mpl_toolkits.mplot3d import Axes3D
6
7 # 手書き数字のデータをロードし、変数digitsに格納
8 digits = datasets.load_digits()
9
10 # 特徴量のセットを変数Xに、ターゲットを変数yに格納
11 X = digits.data
12 y = digits.target
13
14 # 3次元へと次元を減らす主成分分析を定義
15 pca = decomposition.PCA(n_components=3)
16
17 # 主成分分析により、64次元のXを3次元のXrに変換
18 Xr = pca.fit_transform(X)
19
20 # 3次元グラフの領域を準備
21 # '111'は、「縦1枚、横1枚、のグラフエリアの1枚目」を表し、
22 # 表示するグラフが1枚だけであることを意味する
23 fig = plt.figure()
24 ax = fig.add_subplot(111, projection='3d')
25
26 # x, y, zの3つの軸にラベルの設定
27 ax.set_xlabel('X-axis')
28 ax.set_ylabel('Y-axis')
29 ax.set_zlabel('Z-axis')
30
```

64次元 → 3次元

↑

主成分分析

PCA: Principal Component Analysis

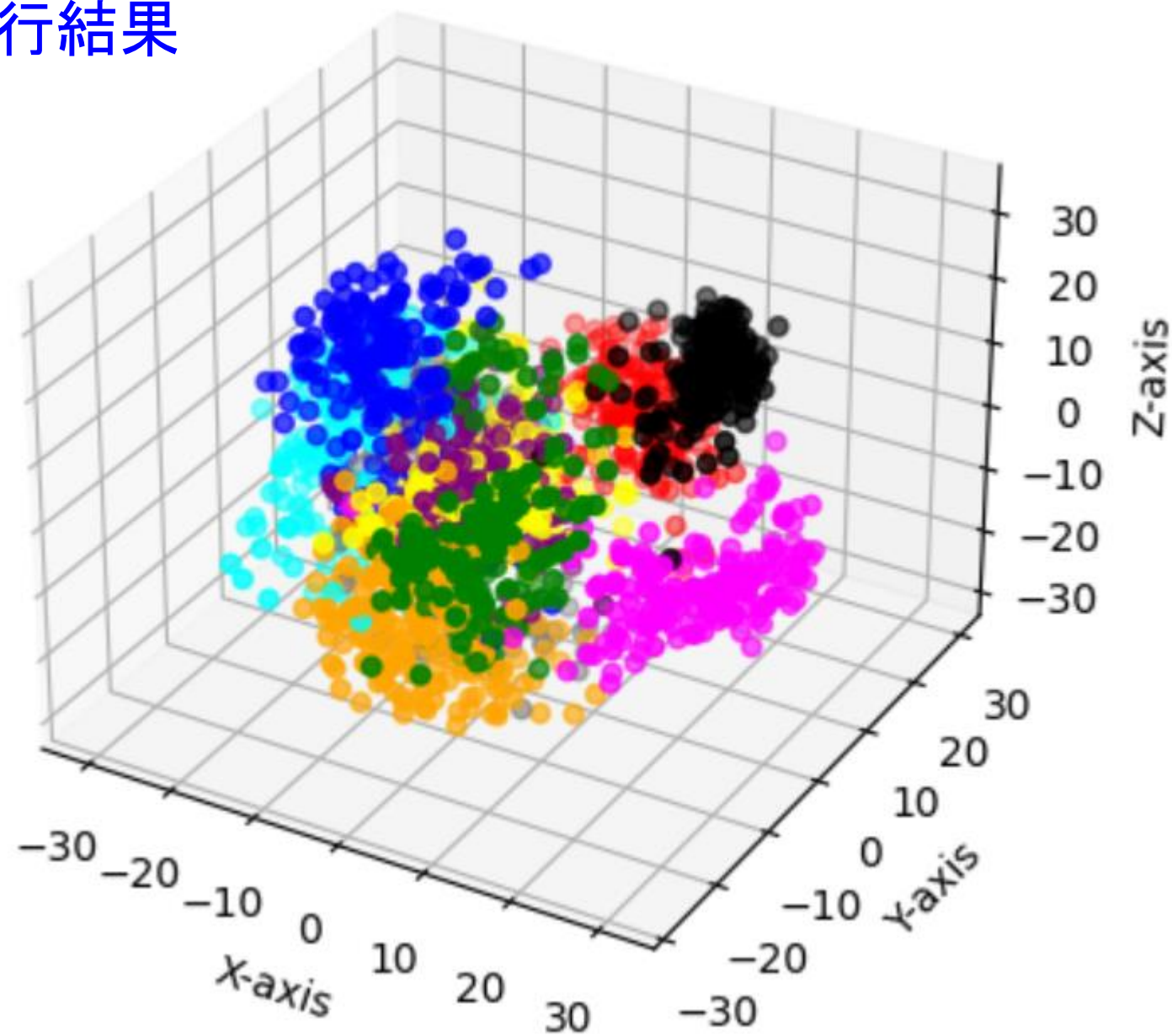
```

31 # 0~9の数字に対する色指定用の関数
32 def getcolor(c):
33     if c==0:
34         return 'red' # 赤
35     elif c==1:
36         return 'green' # 緑
37     elif c==2:
38         return 'blue' # 青
39     elif c==3:
40         return 'cyan' # シアン（水色）
41     elif c==4:
42         return 'magenta' # マゼンタ（ピンク）
43     elif c==5:
44         return 'yellow' # 黄
45     elif c==6:
46         return 'black' # 黒
47     elif c==7:
48         return 'orange' # オレンジ
49     elif c==8:
50         return 'purple' # 紫
51     else:
52         return 'gray' # グレー
53
54 # 正解の数字(y)に対応する色のリストを用意
55 cols = list(map(getcolor, y))
56
57 # 三次元空間へのデータの色付き描画を行う
58 # Xr[:,0] がx軸のデータ
59 # Xr[:,1] がy軸のデータ
60 # Xr[:,2] がz軸のデータ
61 ax.scatter(Xr[:,0], Xr[:,1], Xr[:,2], color=cols)
62
63 # 描画したグラフを表示
64 plt.show()

```

ax.scatter(x座標、y座標、z座標、色)  
指定された色の散布図

## 実行結果



- 0 ← 赤
- 1 ← 緑
- 2 ← 青
- 3 ← シアン(水色)
- 4 ← マゼンダ(ピンク)
- 5 ← 黄
- 6 ← 黒
- 7 ← オレンジ
- 8 ← 紫
- 9 ← グレー