# -\*- coding: utf-8 -\*-

from sklearn import datasets

from sklearn import decomposition

import matplotlib.pyplot as plt

from mpl\_toolkits.mplot3d import Axes3D

# 手書き数字のデータをロードし、変数digitsに格納

digits = datasets.load\_digits()

# 特徴量のセットを変数Xに、ターゲットを変数yに格納

X = digits.data

y = digits.target

# 3次元へと次元を減らす主成分分析を定義

pca = decomposition.PCA(n\_components=3)

# 主成分分析により、64次元のXを3次元のXrに変換

Xr = pca.fit\_transform(X)

# 3次元グラフの領域を準備

# '111'は、「縦１枚、横１枚、のグラフエリアの１枚目」を表し、

# 表示するグラフが１枚だけであることを意味する

fig = plt.figure()

ax = fig.add\_subplot(111, projection='3d')

# x, y, zの3つの軸にラベルの設定

ax.set\_xlabel('X-axis')

ax.set\_ylabel('Y-axis')

ax.set\_zlabel('Z-axis')

# 0～9の数字に対する色指定用の関数

def getcolor(c):

if c==0:

return 'red' # 赤

elif c==1:

return 'green' # 緑

elif c==2:

return 'blue' # 青

elif c==3:

return 'cyan' # シアン（水色）

elif c==4:

return 'magenta' # マゼンタ（ピンク）

elif c==5:

return 'yellow' # 黄

elif c==6:

return 'black' # 黒

elif c==7:

return 'orange' # オレンジ

elif c==8:

return 'purple' # 紫

else:

return 'gray' # グレー

# 正解の数字(y)に対応する色のリストを用意

cols = list(map(getcolor, y))

# 三次元空間へのデータの色付き描画を行う

# Xr[:,0] がx軸のデータ

# Xr[:,1] がy軸のデータ

# Xr[:,2] がz軸のデータ

ax.scatter(Xr[:,0], Xr[:,1], Xr[:,2], color=cols)

# 描画したグラフを表示

plt.show()