Code: LLE

import numpy as np

import matplotlib.pyplot as plt

from itertools import product

from sklearn.manifold import LocallyLinearEmbedding

from sklearn.datasets import fetch\_mldata

from sklearn.utils import shuffle

final\_mnist = fetch\_mldata("MNIST Original")

ourtargets = list(range(0,10))

Xx\_train, Yy\_train = final\_mnist.data / 255., final\_mnist.target

X\_TRAIN=[]

y\_TRAIN=[]

for i, label in enumerate(Yy\_train):

if label in ourtargets:

X\_TRAIN.append(Xx\_train[i])

y\_TRAIN.append(Yy\_train[i])

num\_samples\_to\_plot = 5000

X\_TRAIN, y\_TRAIN = shuffle(X\_TRAIN, y\_TRAIN)

X\_TRAIN, y\_TRAIN = X\_TRAIN[:num\_samples\_to\_plot], y\_TRAIN[:num\_samples\_to\_plot] # lets subsample a bit for a first impression

for digit in ourtargets:

instances=[i for i in y\_TRAIN if i==digit]

print ("Digit",digit,"appears ",len(instances), "times")

transformer = LocallyLinearEmbedding(n\_neighbors = 10, n\_components = 2,

eigen\_solver='auto', method='standard')

fig, plot = plt.subplots()

fig.set\_size\_inches(50, 50)

plt.prism()

X\_transform = transformer.fit\_transform(X\_TRAIN)

plot.scatter(X\_transform[:, 0], X\_transform[:, 1], c=y\_TRAIN)

from sklearn.manifold import Isomap

from sklearn.manifold import MDS

#ISOMAP

data\_y = final\_mnist.data[::30]

target = final\_mnist.target[::30]

K = Isomap(n\_components=2, n\_neighbors=10)

proj = K.fit\_transform(data\_y)

plt.scatter(proj[:, 0], proj[:, 1], c=target, cmap=plt.cm.get\_cmap('jet', 10))

plt.colorbar(ticks=range(10))

plt.clim(-0.5, 9.5)

plt.show()

count=0;

plt.tight\_layout()

plt.suptitle("LLE for MNIST digits ")

for label , x, y in zip(y\_train, X\_transform[:, 0], X\_transform[:, 1]):

if count % 200 == 0:

plt.annotate(str(int(label)),xy=(x,y), color='black', weight='normal',size=10,bbox=dict(boxstyle="round4,pad=.5", fc="0.8"))

count = count + 1

plt.show()