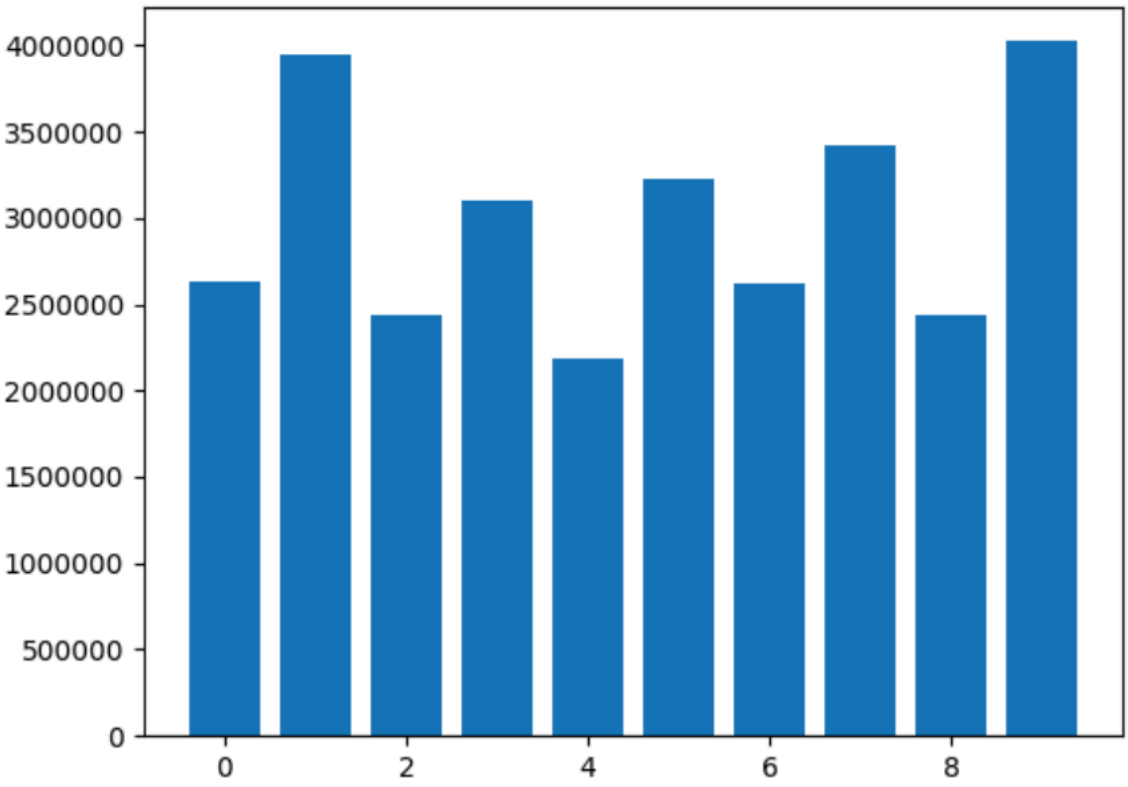
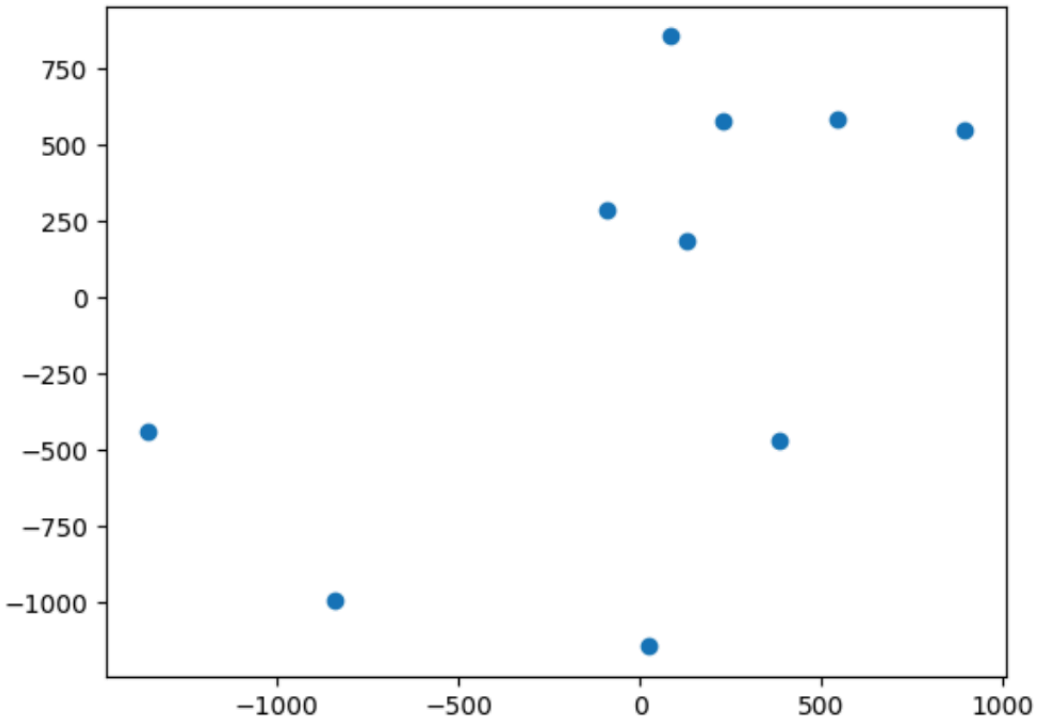
CS498 AML HW4

Yidi Yang (yyang160)
Huiyun Wu (hwu63)

1. MSEs of category 0 ~ 9.
   We use the MSEs between the training data and the reconstructed **training** data. (We personally think this makes more sense.)



2. 2D Map of Principal Coordinates Analysis.



3. Code Screenshots.

```python
from sklearn.decomposition import PCA
from scipy.spatial import distance_matrix
from skbio.stats.ordination import pcoa
import numpy as np
import pickle
from matplotlib import pyplot as plt
from sklearn import manifold


def unpickle(file):
    with open(file, 'rb') as fo:
        dict = pickle.load(fo, encoding='bytes')
    return dict


def pca(data):
    pca = PCA(n_components=20)
    components = pca.fit_transform(data)
    reconstruct = pca.inverse_transform(components)
    return reconstruct


def divide(raw_data, labels, data):                # Separate out the 10 categories
    for i in range(len(labels)):
        if labels[i] not in data.keys():
            data[labels[i]] = [raw_data[i]]
        else:
            data[labels[i]].append(raw_data[i])
    return data


def mse(mat1, mat2):
    return 1/5000* np.sum((mat1-mat2)**2)
```

-----After parsing data------

```python
    data = dict()
    data = divide(raw_data1, labels1, data)
    data = divide(raw_data2, labels2, data)
    data = divide(raw_data3, labels3, data)
    data = divide(raw_data4, labels4, data)
    data = divide(raw_data5, labels5, data)

    error = [0 for i in range(10)]
    meanImages = np.zeros((10, 3072))

    for i in range(10):
        meanImages[i] = np.mean(data[i], axis= 0)
        recon = pca(data[i])
        for j in range(len(data[i])):
            error[i] += mse(data[i][j], recon[j])

    plt.figure(0)
    plt.bar(range(1, len(error)+1), error)
    plt.show()

    # 7.7 (b)
    dis_mat = distance_matrix(meanImages, meanImages)
    plt.figure(1)
    mds = manifold.MDS(dissimilarity='precomputed')
    results = mds.fit(dis_mat)
    coords = results.embedding_
    plt.scatter(coords[:, 0], coords[:, 1])

    plt.show()
```