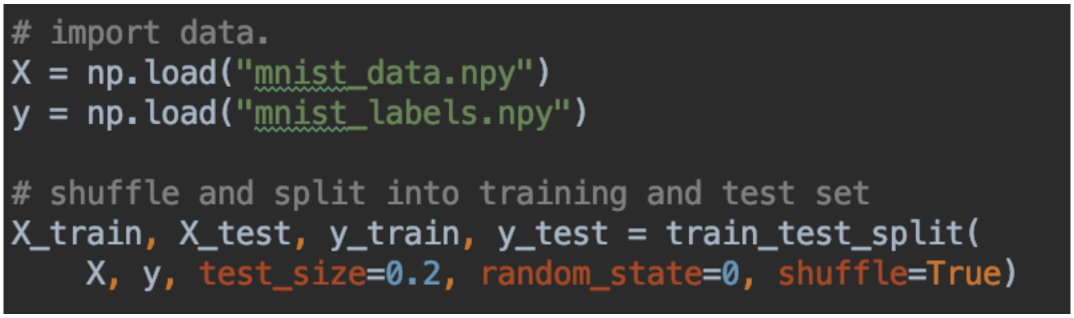
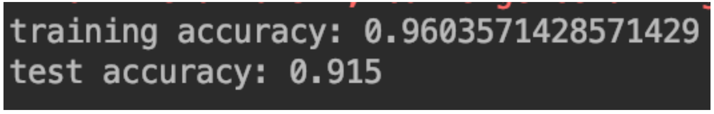
HW 2
Leyang Shen

1. a.
   Load the data and labels by the following. Then shuffle and split into training and test set.



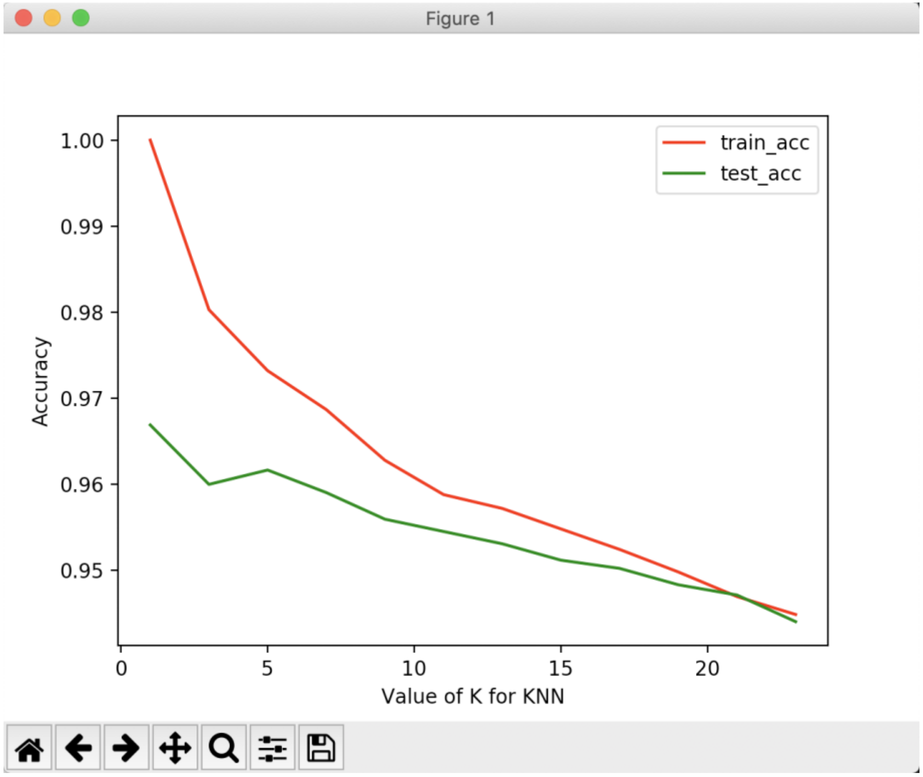
```
# import data.
X = np.load("mnist_data.npy")
y = np.load("mnist_labels.npy")

# shuffle and split into training and test set
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=0, shuffle=True)
```

b. The training acc and test acc are listed as follow:



```
training accuracy: 0.9603571428571429
test accuracy: 0.915
```

c. The training acc and test acc as varying k from 1 to 25 with step size 2 is shown in the graph below.



d. The classification process of knn takes much more time than logistic regression. However, knn supports non-linear solutions where logistic regression supports only linear solution. So the overall accuracy of knn is better than logistic regression.

2. a.

```
scaler = StandardScaler()
# Fit on training set onl
```

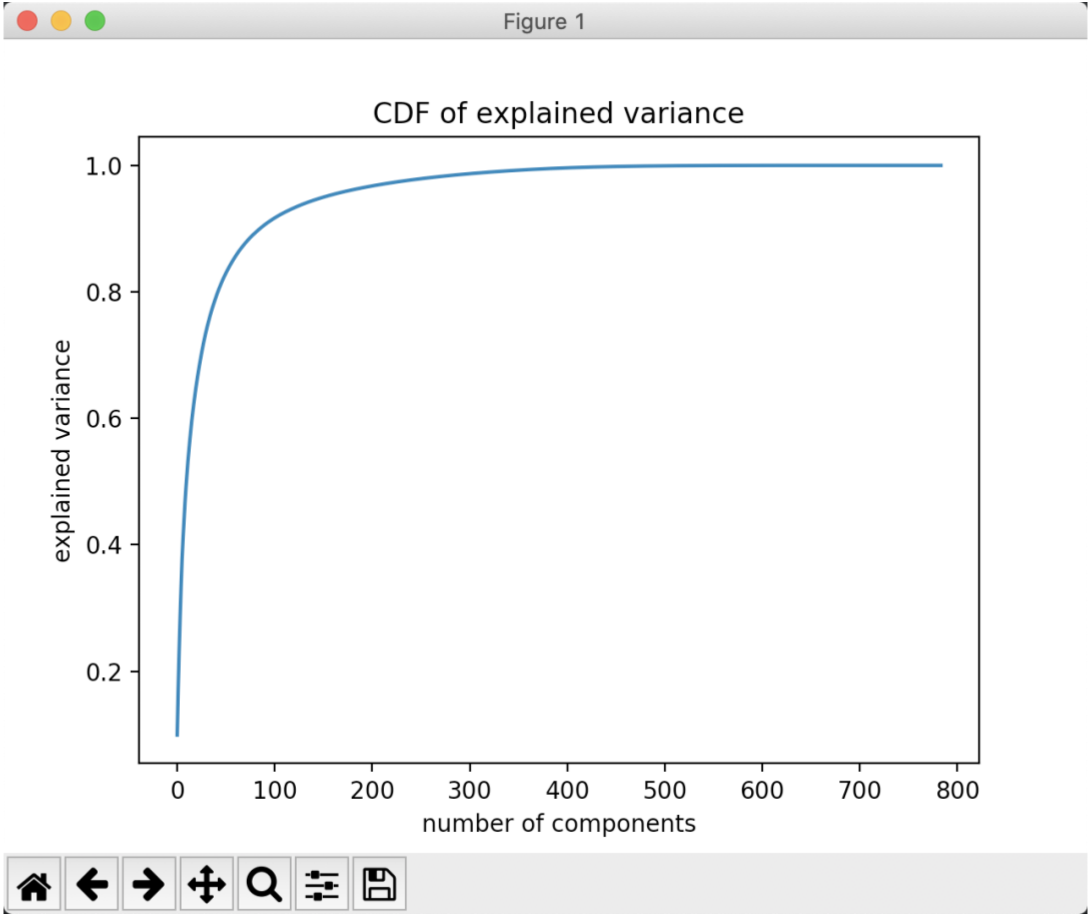
The standardScaler method from sklearn decomposition package contains the method of center-meaning. The standard score of a sample `x` is calculated as:

z = (x - u) / s

where `u` is the mean of the training samples or zero if `with_mean=False`, and `s` is the standard deviation of the training samples or one if `with_std=False`.

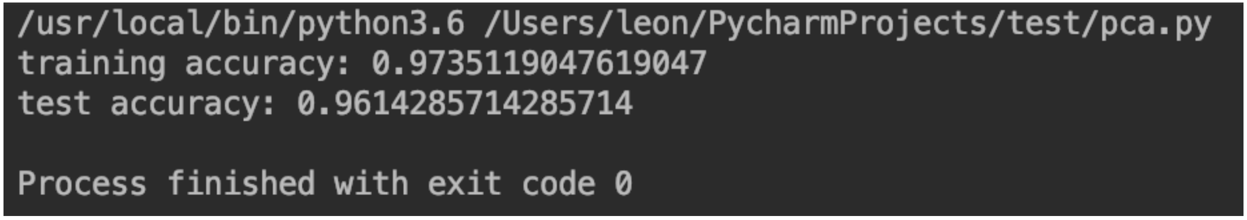
Centering and scaling happen independently on each feature by computing the relevant statistics on the samples in the training set. Mean and standard deviation are then stored to be used on later data using the `transform` method.

b. The CDF of the explained variance is shown as below:



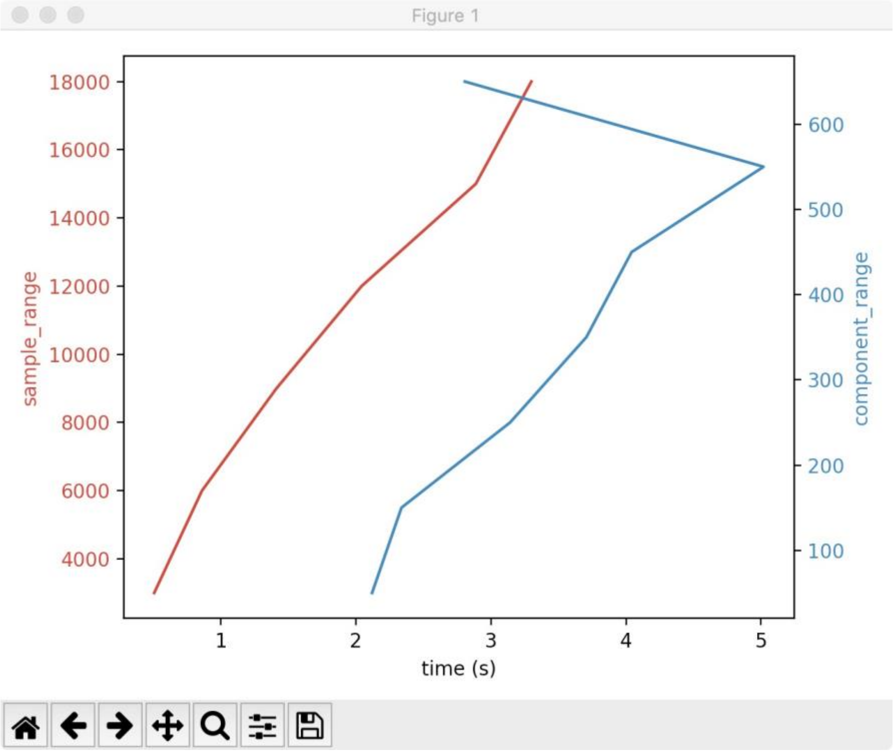
c. I choose 300 as my number of principle components because it is most likely the threshold that explained variance ratio becomes stable.

When k=5, it gives me the best accuracy on test set in problem 1.
After deploying knn, the training and test accuracy are stated below:



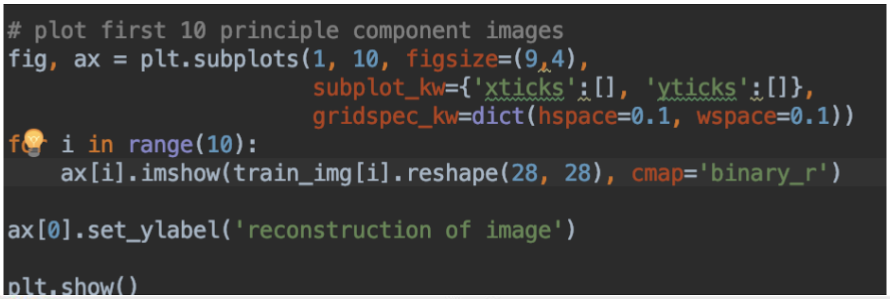
```
/usr/local/bin/python3.6 /Users/leon/PycharmProjects/test/pca.py
training accuracy: 0.9735119047619047
test accuracy: 0.9614285714285714

Process finished with exit code 0
```

d.



For overall both progresses, the time consumed will increase when number of samples and principle components increase. But it occurs that the principle components affect the fitting more because it is less linear.

e. We plot the graph right after making instance and fitting.



```
# plot first 10 principle component images
fig, ax = plt.subplots(1, 10, figsize=(9,4),
                       subplot_kw={'xticks':[], 'yticks':[]},
                       gridspec_kw=dict(hspace=0.1, wspace=0.1))
for i in range(10):
    ax[i].imshow(train_img[i].reshape(28, 28), cmap='binary_r')

ax[0].set_ylabel('reconstruction of image')

plt.show()
```

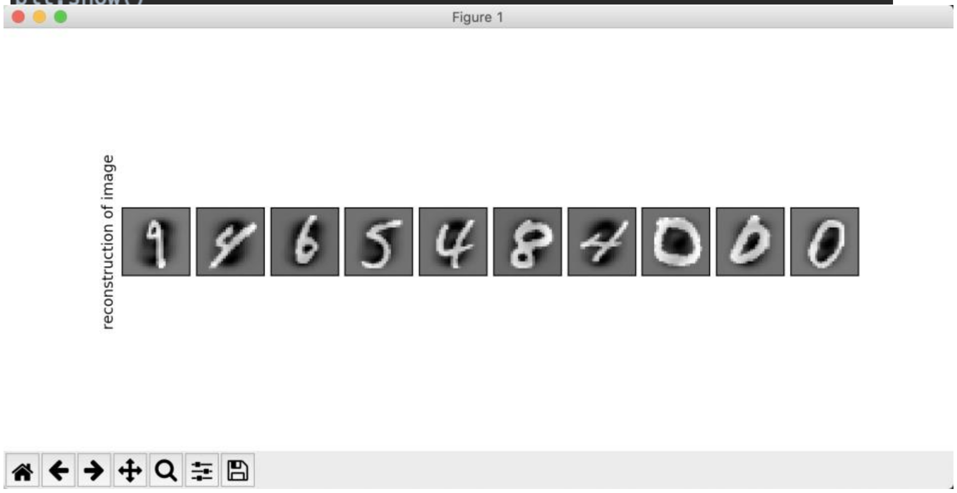


Figure 1
reconstruction of image