Code for problem 4.

```
import numpy as np
import os
os.chdir('/home/tuomas/Documents/DATA.STAT.770/koe')

#%%
iris_data = np.loadtxt('iris-commaseparated.txt', delimiter=',')
iris_data = np.concatenate((np.array([[5.1,3.5,1.4,0.2,0]]), iris_data))

labels = iris_data[:,-1]
iris_data = iris_data[:,0:4]

#%% a)
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
pca.fit(iris_data)

iris_reduced = pca.transform(iris_data)

#%% Plot
import matplotlib.pyplot as plt

iris1 = iris_reduced[(labels==0),:]
iris2 = iris_reduced[(labels==1),:]
iris3 = iris_reduced[(labels==2),:]

plt.plot(iris1[:,0], iris1[:,1], 'bo')
plt.plot(iris2[:,0], iris2[:,1], 'ro')
plt.plot(iris3[:,0], iris3[:,1], 'go')
plt.title('Blue=setosa\n Red=versicolor\n Green=virginica')

#%% b)
from numpy.linalg import norm
prop_of_var = np.sum(pca.explained_variance_ratio_)
print('Proportion of variance explained = {}'.format(prop_of_var))

back_proj = pca.inverse_transform(iris_reduced)
loss = norm((iris_data-back_proj), None)
print('Reconstruction error = {}'.format(loss))


#%% c)
from sklearn.manifold import TSNE
semeion_data = np.loadtxt('semeion-commaseparated.txt', delimiter=',')

labels = semeion_data[:,-1]
semeion_data = semeion_data[:,0:256]

#%%
tsne = TSNE(n_components=2)
semeion_proj = tsne.fit_transform(semeion_data)
```

```
#%% Plot
colors = [(1,0,0),(0,1,0),(0,0,1),
        (1,1,0), (0,1,1), (1,0,1),
        (0.5,0.3,0.2), (0.2,0.5,0.8), (0.8,0.9,0.0)]
for i in range(0,9) :
    data = semeion_proj[(labels==i),:]
    plt.plot(data[:,0], data[:,1], 'o', color=colors[i])

plt.title('T-distributed stochastic neighbourhood embedding')
```