

# Machine Learning

## Homework #2

### Question 4 Supplementary

Daniel Ho Kwan Leung  
104360098  
CSIE, Year 3

#### Question 4:

The re-implementation of Factor Analysis is due to the using of complete module and library in the previous homework. The following code are still completed by using python, code can be viewed in my GitHub:

[https://github.com/DHKLLeung/NTUT\\_Machine\\_Learning/blob/master/HW2\\_Q4.ipynb](https://github.com/DHKLLeung/NTUT_Machine_Learning/blob/master/HW2_Q4.ipynb)

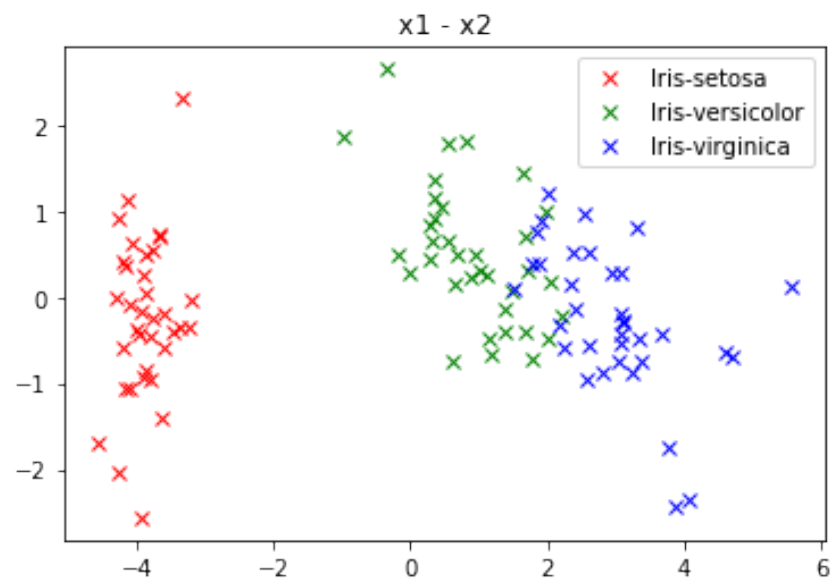
Nearly all the functions used are the same as the functions used in Question 3. The following are specific for this question.

```
"""
Factor Analysis
Return factor matrix
"""
def fa(features, colvar=True):
    features = features.T if not colvar else features
    S = np.cov(features)
    eigen_vals, eigen_vects = np.linalg.eig(S)
    eigen_vals = np.diag(eigen_vals)
    V = np.dot(eigen_vects, np.sqrt(eigen_vals))
    return V

features, labels = load_data(classes)
avg_accuracy = 0.
for i in range(cross_val_times):
    train_features, train_labels, test_features, test_labels = data_preparation(features,
labels)
    V = fa(train_features.T)
    train_features_reduced = np.dot(train_features, V[:, 0:2])
    test_features_reduced = np.dot(test_features, V[:, 0:2])
    predicts = kNN_inference(train_features_reduced, train_labels, test_features_reduced)
    avg_accuracy += np.mean((predicts == test_labels).astype(np.float32)) / cross_val_times
print('Scatter Plots of Original Data: ')
plot_scatter(train_features, train_labels)
print('Scatter Plots of FA-transformed Data: ')
plot_scatter(train_features_reduced, train_labels)
print('Average Accuracy ({} times): {}'.format(cross_val_times, avg_accuracy))
```

The data are standardized before doing Factor Analysis. The Scatter Plots of Original Data are nearly the same as the plots in Question3.

Scatter Plots of FA-transformed Data,



The average accuracy of 10 times is 0.9155555605888367.