# Wangyinan-coursework3

Load Dataset

```
1 from sklearn.datasets import load_iris
  iris = load_iris()
  iris_data = iris['data']
  iris_target = iris['target']
  iris_names = iris['feature_names']
```

Part 1 - PCA

1. Apply PCA projection of the features of IRIS dataset in 2 dimensions

```
2 from sklearn.decomposition import PCA
  pca = PCA(n_components=2)
  pca.fit(iris_data)
```

```
2 PCA(n_components=2)
```

2. Show how much variance ratio is explained by the reduced dimension.

```
3 print(pca.explained_variance_ratio_)
```

```
[0.92461872 0.05306648]
```

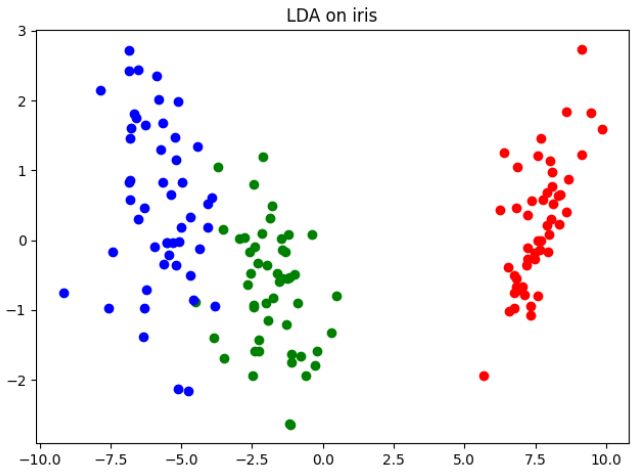
Part 2 - Another Reduction Method

1. Apply another reduction method of your choice on the features of IRIS dataset in 2 dimensions.

```
4 from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
  lda = LDA(n_components=2)
  lda_result=lda.fit_transform(iris.data, iris.target)
```

2. Show scatter plot of the reduced dimension. Use separate color for each class of the data.

```
5 import matplotlib.pyplot as plt
  plt.figure()
  plt.scatter(lda_result[iris.target == 0, 0], lda_result[iris.target == 0, 1], color='r')
  plt.scatter(lda_result[iris.target == 1, 0], lda_result[iris.target == 1, 1], color='g')
  plt.scatter(lda_result[iris.target == 2, 0], lda_result[iris.target == 2, 1], color='b')
  plt.title('LDA on iris')
  plt.show()
```




3. Show how much variance ratio is explained by the reduced dimension.

```
6 print(lda.explained_variance_ratio_)
```

```
[0.9912126 0.0087874]
```

Part 3 Conceptual Question

1. Compare the variance ratio explained by the 2-dimensions of the methods you have used. Which is better?

PCA:[0.92461872 0.05306648]

LDA:[0.9912126 0.0087874]

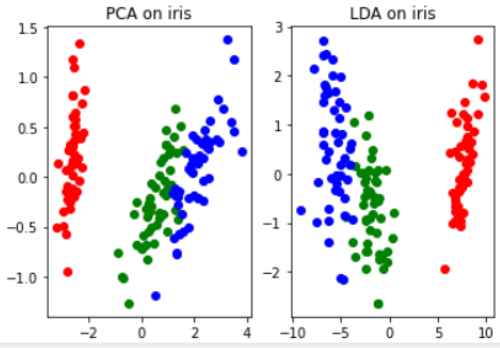
So LDA is better

2. Compare the scatter plot of the two methods after reduction. Which is a better method for separating the dif

The scatter plot is as follows. It shows LDA is a better methos for separating the different classes.

```
8 pca = PCA(n_components=2)
pca_result = pca.fit_transform(iris.data)
plt.figure()
plt.subplot(1, 2, 1)
plt.scatter(pca_result[iris.target == 0, 0], pca_result[iris.target == 0, 1], color='r')
plt.scatter(pca_result[iris.target == 1, 0], pca_result[iris.target == 1, 1], color='g')
plt.scatter(pca_result[iris.target == 2, 0], pca_result[iris.target == 2, 1], color='b')
plt.title('PCA on iris')

plt.subplot(1, 2, 2)
plt.scatter(lda_result[iris.target == 0, 0], lda_result[iris.target == 0, 1], color='r')
plt.scatter(lda_result[iris.target == 1, 0], lda_result[iris.target == 1, 1], color='g')
plt.scatter(lda_result[iris.target == 2, 0], lda_result[iris.target == 2, 1], color='b')
plt.title('LDA on iris')
plt.show()
```



3. What is the primary difference between the two methods? Which method works better in this case and why?

PCA is unsupervised dimensionality reduction, and LDA is supervised dimensionality reduction. PCA hopes that the variance of the data after projection is as large as possible (maximum separability), because it assumes that the more variance, the more information it contains; while LDA hopes that the variance within the same category after projection is small, and the variance between groups The variance is large. In this case, LDA is better. Because LDA can reasonably use label information to make the dimension after projection discriminative, and separate data of different categories as much as possible. Therefore, if there is a label, the data of the label should be used as much as possible