Final Exam ziqiyuan S4669779

**Target:**

I planning to do a classification, base on the given feature predict which Classification that this item belong to.

**Data pre-Processing/clean:**

The raw FoodNutrients contain too much unrelated data and also many columns have too many missing value. So first I simply drop the columns which has over 95% null values, and because the Classification contain too many different labels, also I notice the Classification number which start with three same integer can be grouped into one broad category, if I’m not doing this too many categories and too small sample size will easy lead to the overfitting problem, below is the code for the Data pre-processing:

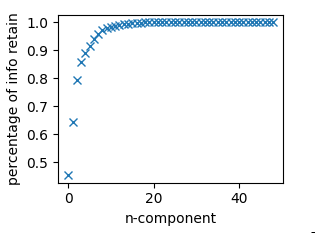
**Normalize the data:**

I realize many units of measurement in the sample features are not uniform. The logic of the PCA is project high dimension data to the low dimension, and keep it variance as large as possible in the low dimension, which means if there is a coloumn have really big value, then apply pca on it without doing normalize it will lead to the projection will try to approximate the largest feature and ignore the features with smaller values. After I normalize the data, it generate many small values that close to 0, so I just use Python to fill in vacant values that converge to zero. Blow is the code that how I doing this:

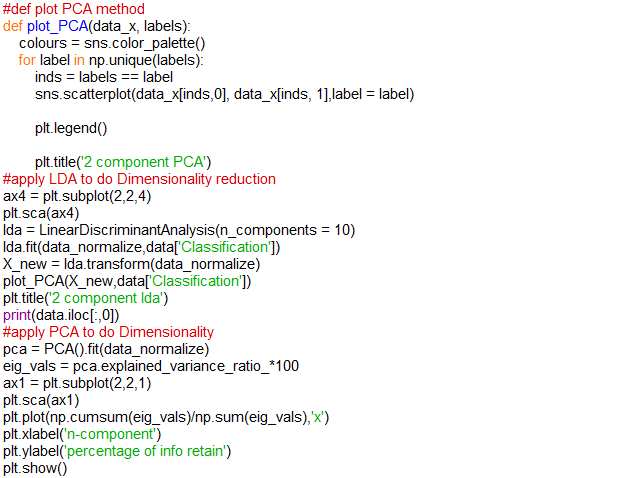


Show how many information that the data after apply PCA keeps:

Below is the graph:



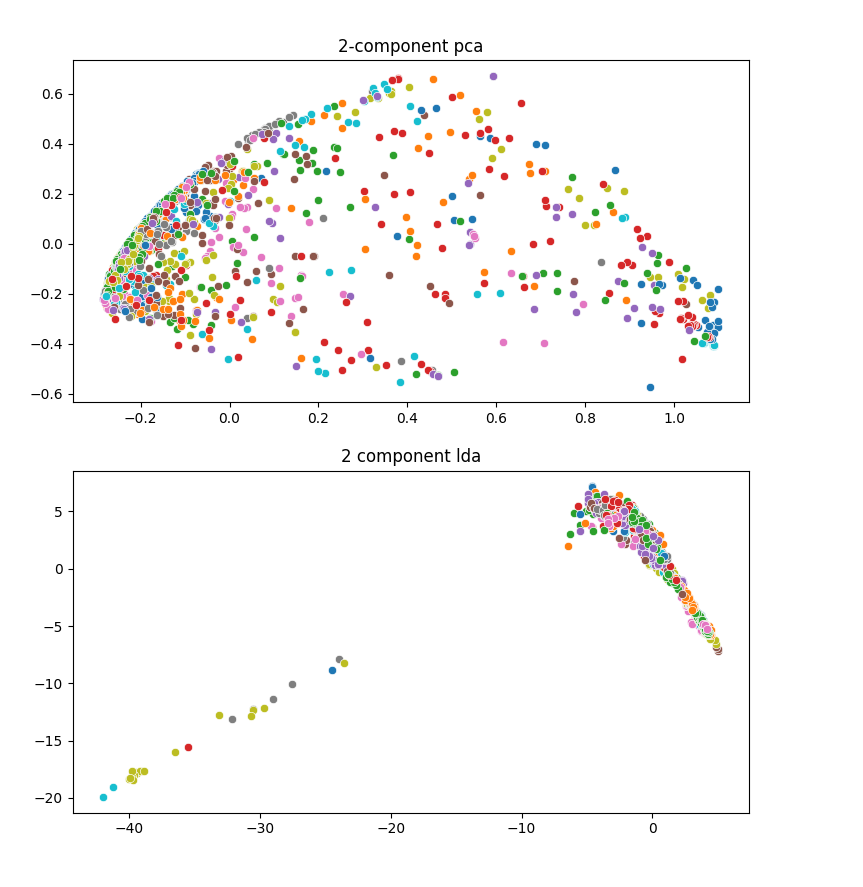
As we see if we keep around 15 components we should have retained more than 95% of the variance divergence information of the original data below is the code that how I doing this:



**Try LDA to do the dimension reduction:**

The reason that using LDA because the given dataset is labelled, and PCA is unsupervised, which means it will not preserve the relationship between classes and features, but LDA can do this, the basic logic of the LDA is get the smallest intra-calss variance and the largest inter-class variance after projection.

I also using both LDA and PCA to project the original data into 2 dimension space aims to visualize it, below is how it looks:

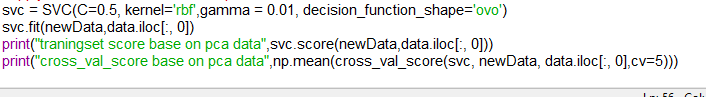


It looks like both of them do not have good performance with only 2 component, it should because the original have too many features, only 2 component do not enough to show the original structure.

But by the two-dimensional projection of this data, we can intuitively guess that the original data should not be linearly separable in the higher dimensional space, so using a linear classifier should be undesirable.

**Using SVM to do the classification base on the PCA:**

First I choose C=0.5(Plenalty Factor) gamma = 0.01, to build the SVM classifier, because high Plenalty Factor(Small tolerance for error) and High gamma will easy to lead overfit problem, that why I choose a relative small C and gamma value at first, then I print the the accuracy of the model on the training set and it’s cross\_val\_score with 5-flod,and the I retain 15 components. Below is the code and the result:



As we see the performance is terrible, it should because the PCA do not respect to the relation between the class and feature, or the model is under fitting because performance on the taining set is also not good, so I try increase the C and gamma to do it again.

1. C=5 gamma = 0.5



1. C=20 gamma = 1



This time overfitting problem occurred, because the score on the trainingset is far more better than the cross\_val\_score.

At the end the performance for this model is not good enough let try SVM on the LDA data next.

**Using SVM to do the classification base on the LDA:**

Still start with C=0.5,gamma = 0.01



It looks much better than PCA, the main reason I think is just because the PCA do not respect to the relation between the class and feature, but LDA does.

Let’s try whether we can get a better performance by changing the hyperparameter

1. c=3 gamma = 0.02



We can see Based on the lda-processed data, the accuracy of the SVM model was significantly higher on line than that of PCA.

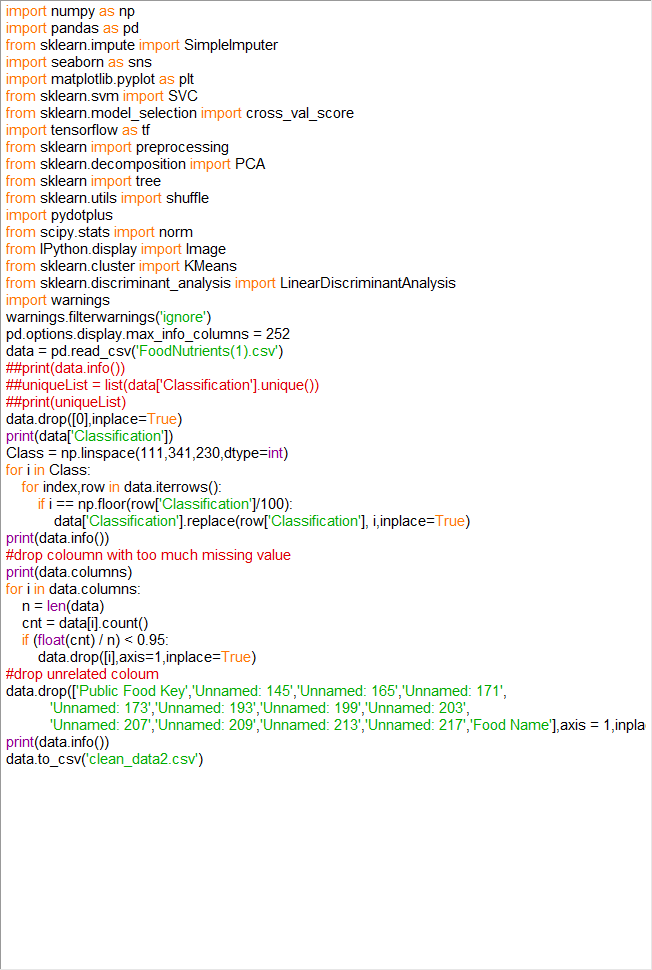
Conclusion:

What I think is if the dataset is labelled LDA is a better dimension reduction method than the PCA.

**Appendix:**

Full code:

Pre-processing:



Dimension reduction and model

