HW6 report 0410024余東儒

Code explanation:

Part 1---Function1: Readfile

**X\_name**: X\_test or X\_train path

**T\_name**:T\_test or T\_train path

**Size**: number of data

**Mode**: output file name will be train.txt or test.txt,it’s for LIBSVM format file

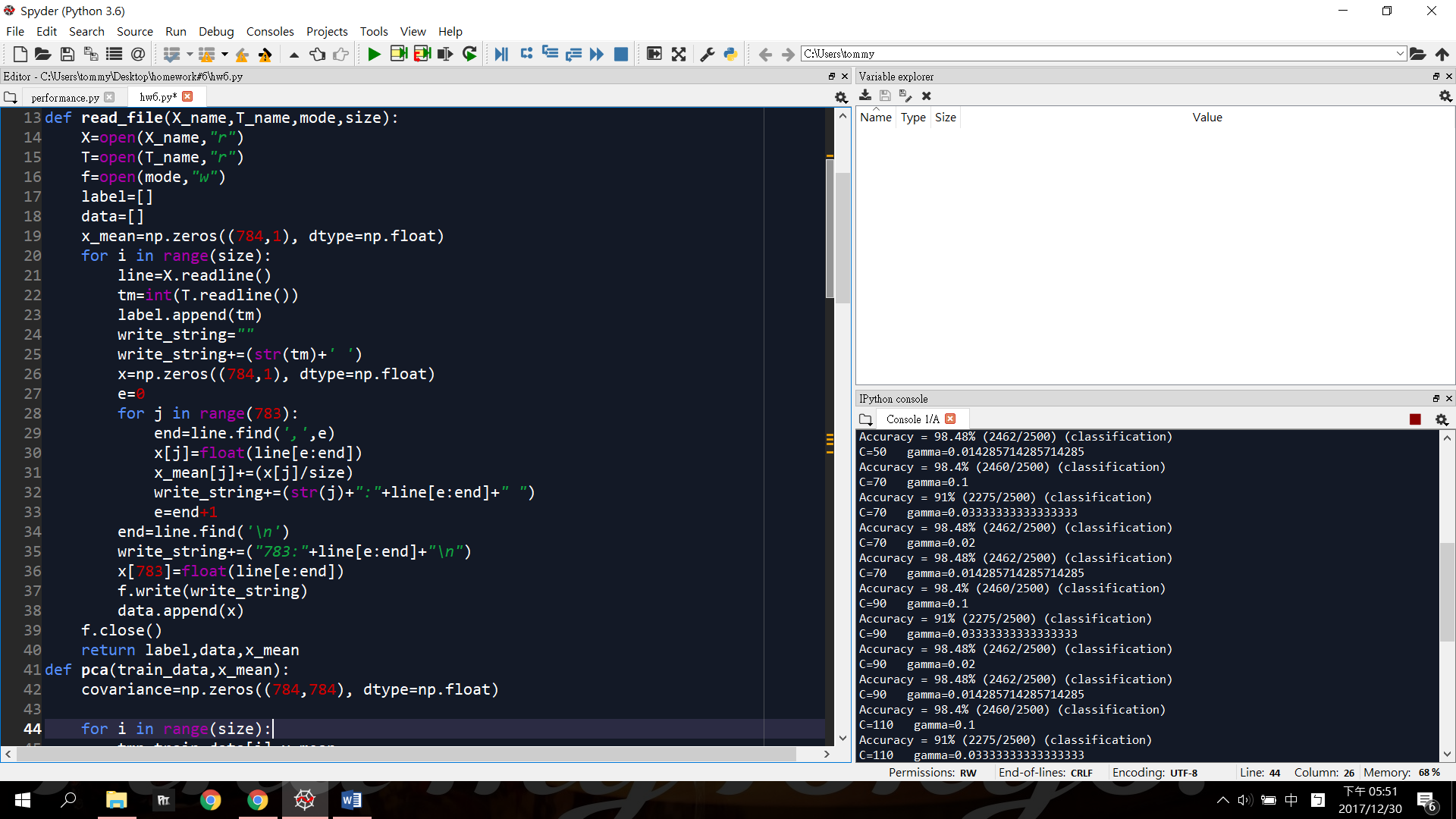
**label**: store all labels from T file(2500,5000)

**data**: store all data from X file(784\*2500,784\*5000)

**x\_mean**: store each pixel’s average ,used for caculating covariance in PCA

outer loop:

1. Read file line by line
2. **line** for X\_file;**tm** for T\_file
3. **write\_string** for writing a line in LIBSVM format file
4. **x**: matrix for storing one data
5. innerloop: Pick every pixel’s value out and dividing **size** storing into **x\_mean**. Also, storing into write\_string
6. write into outputfile; storing x into data



Part2---Function2:PCA

Main purpose: calculating W martrix and projecting traing data to 2D space

Arguments: results from Readfile function,using train\_data and x\_mean

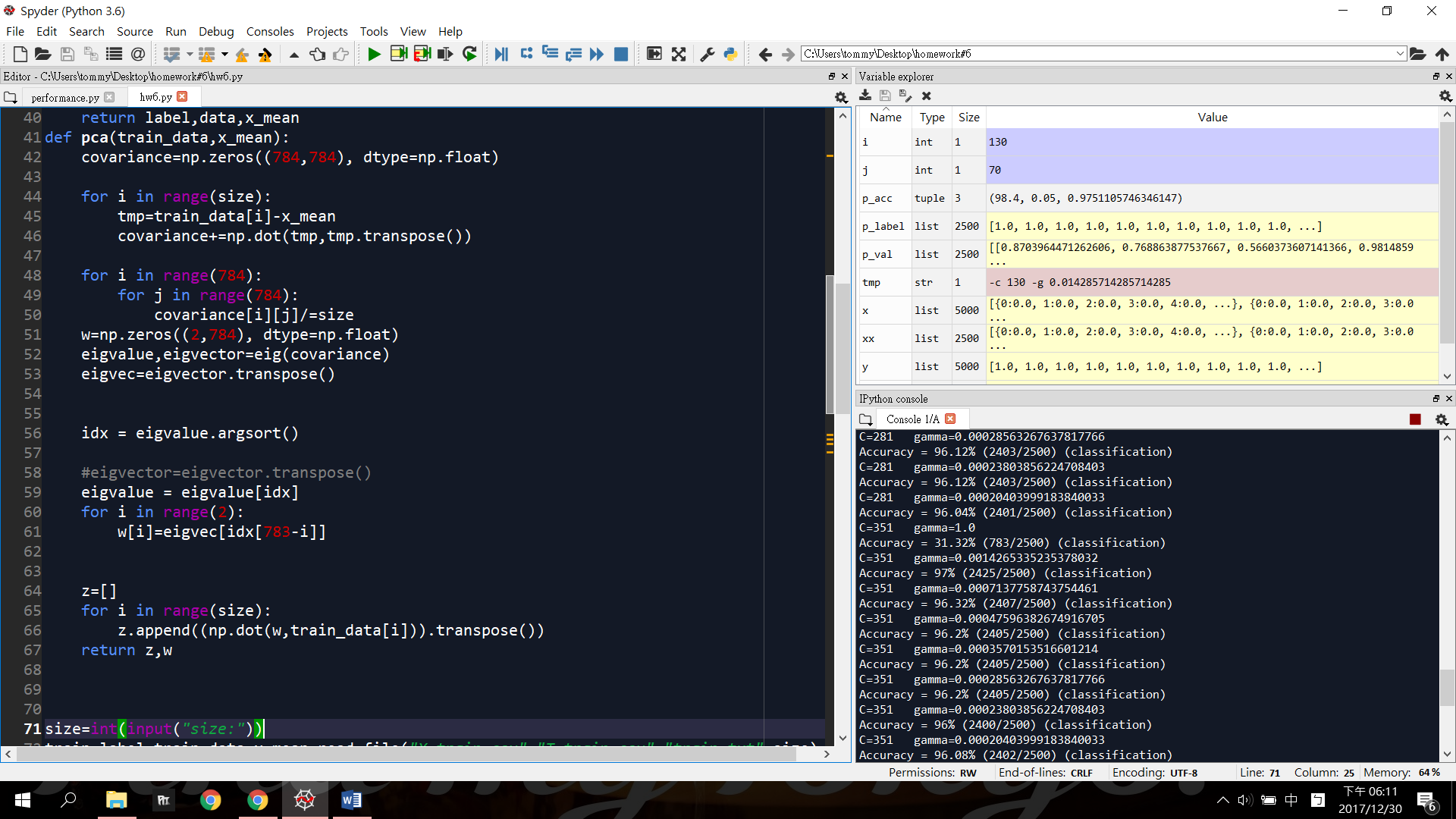
First loop and second loop: Calculating covariance matrix

Covariance to W matrix:

1. Dealing with eigen problem by eig library.
2. Eigvector needs to transpose because original form will be col base,trasposing to row base will be more easy to get
3. Sorting eigenvalue by argsort()
4. Picking first two largest eigenvalue and corresponding eigenvector and put them to **w**

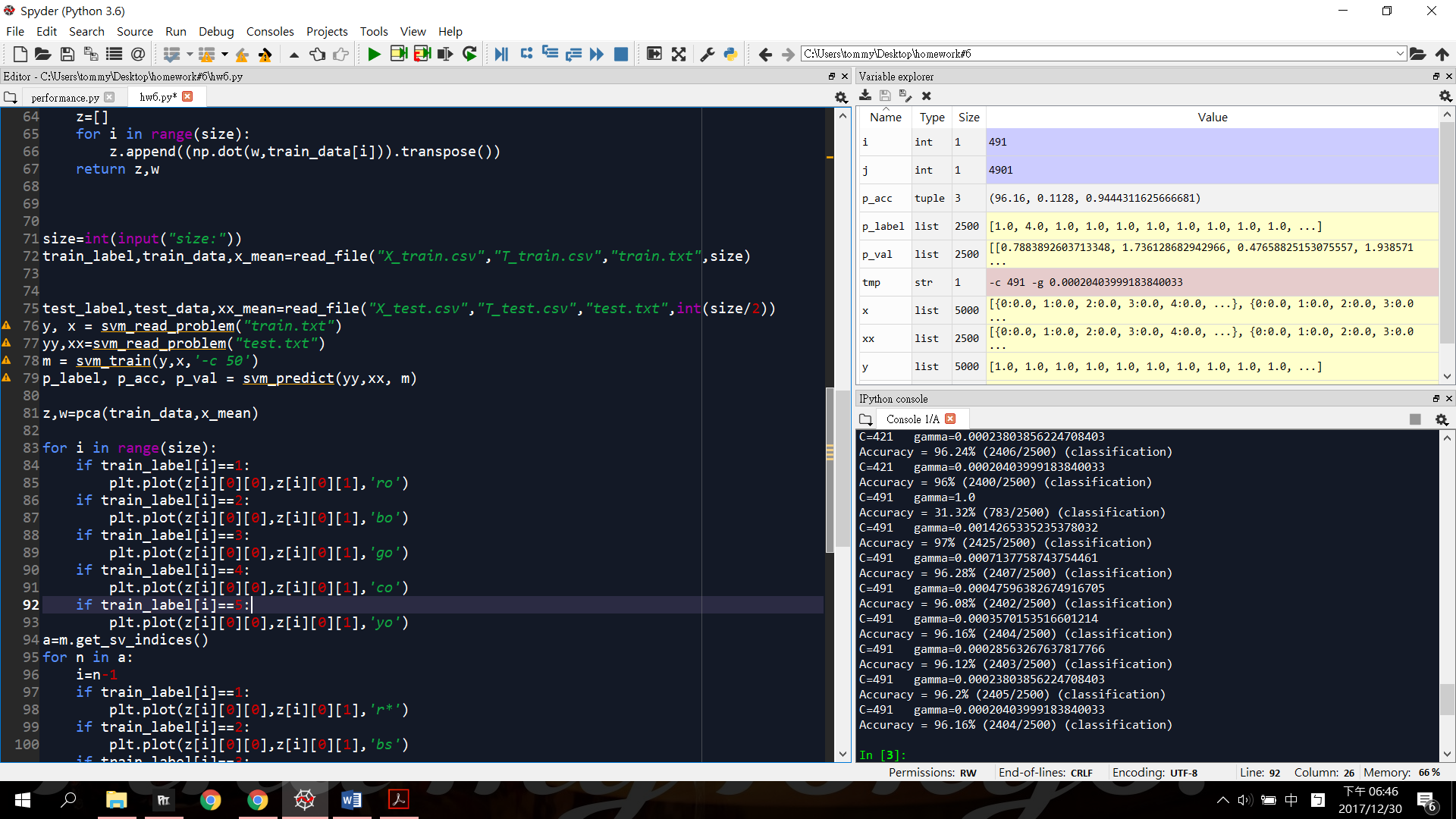
Last loop:

Using **w** matrix to project **train\_data** to **z**(2d sapce)



Main part---training and testing:

1. Read test and train file first
2. Using svm\_read\_problem to get label and data from LIBSVM form file
3. Using training data and label to train model used by svm\_train.Here set option to C=50
4. Using testing data and label to test model used by svm\_predict
5. p\_acc will be the Accuracy rate



Ploting part--- PCA and plot it!!!

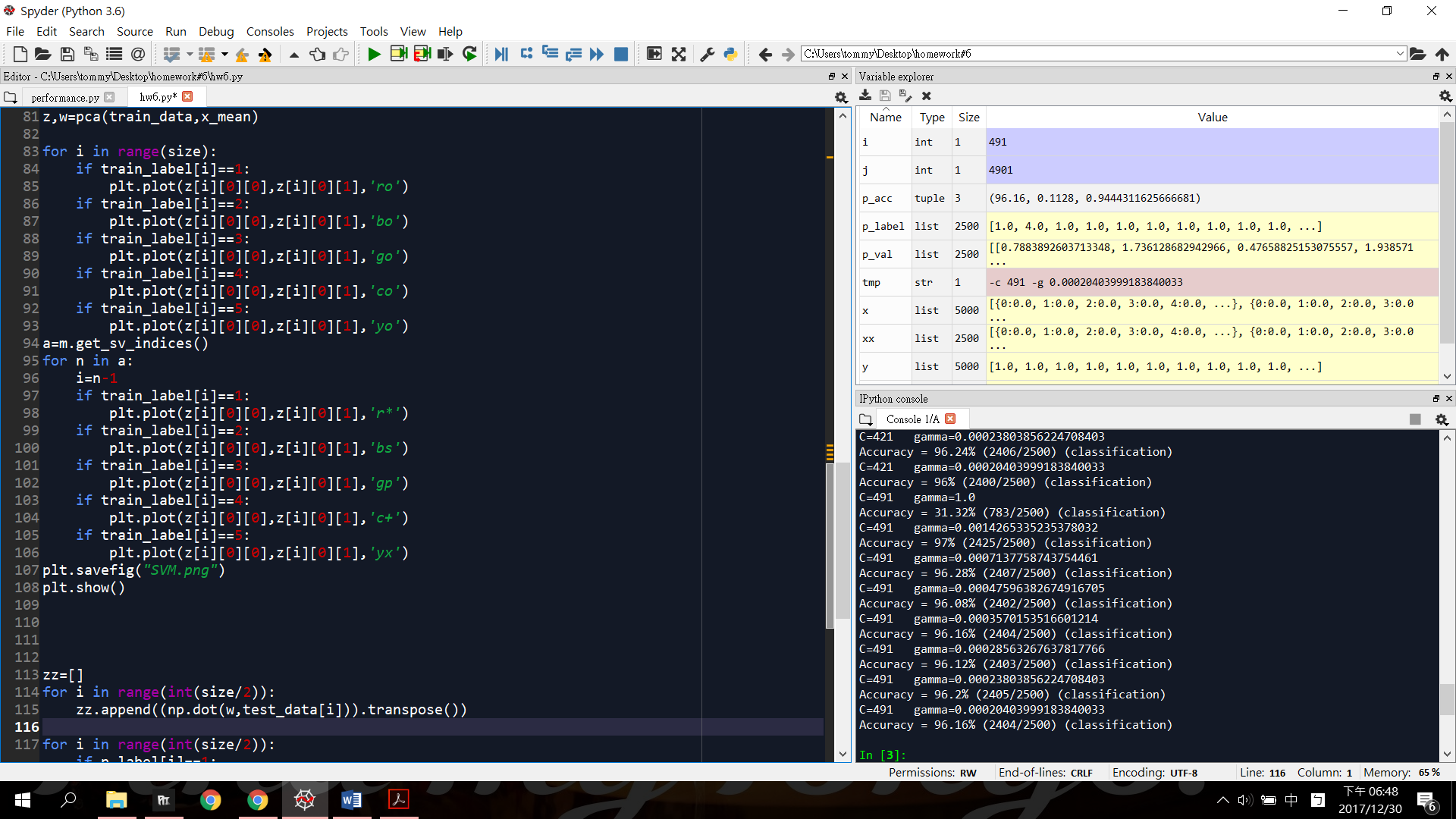
Do the PCA first and get z and w

1.Plot training data

First loop: plot all the training\_data from z depend on which label it is

**a**: get the indices where the support vectors are by get\_sv\_indices

Second loop: Plot all the support vectors with special symbols



2.Plot testing data

First loop:

zz: storing testing data’s projecting data

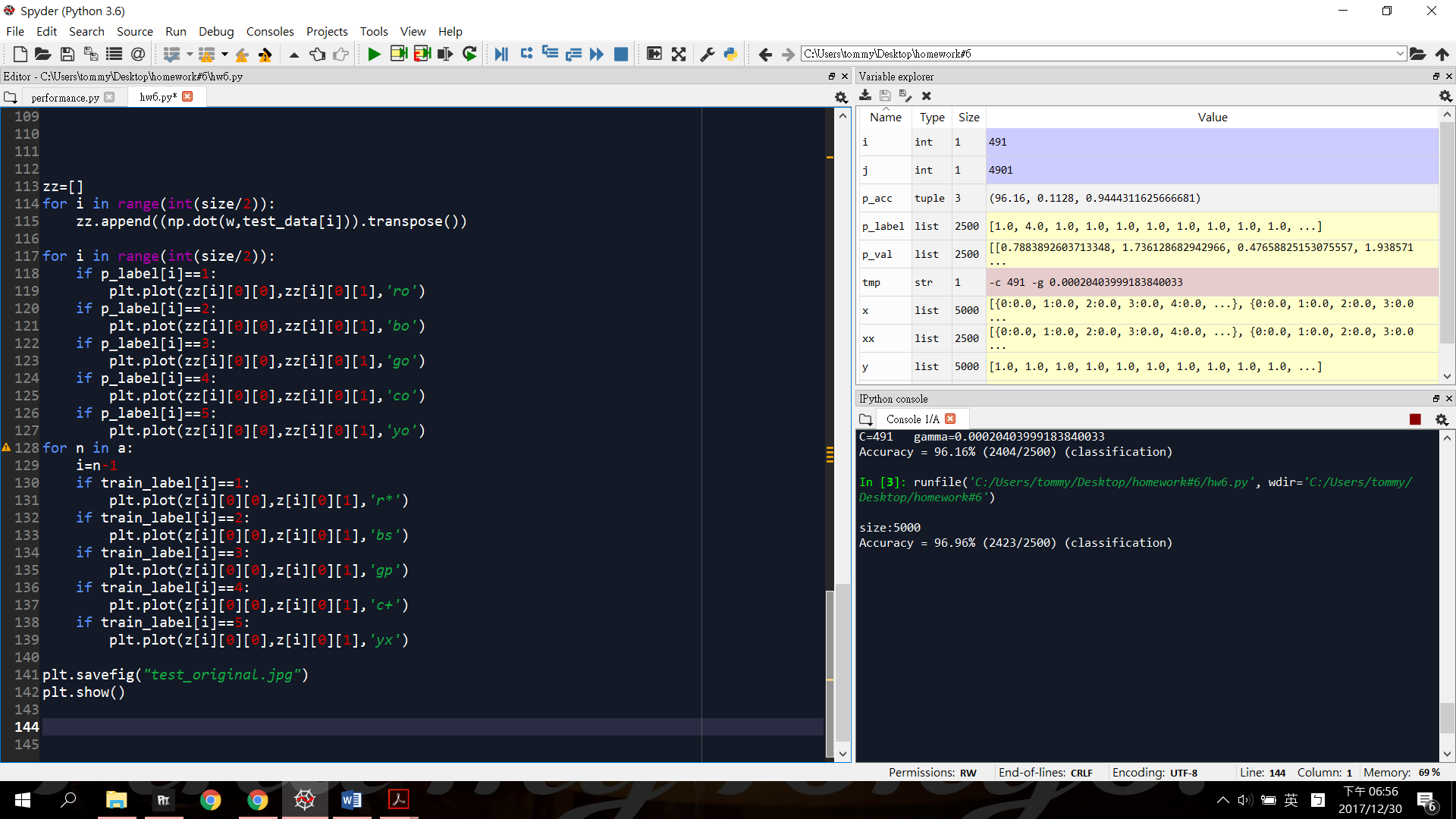
use w to get projecting data

Second loop:

Plot all testing data

Third loop:

Plot all support vectors



Performance:

In previous code, I use C=50 and default RBF. The accuracy rate is 96.96% (2423/2500).

Some discussion on performance (C,gamma) (For bonus):

Chart1:C=1~491 1/gamma=1~4901

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C,1/gamma | 1 | 701 | 1401 | 2101 | 2801 | 3501 | 4201 | 4901 |
| 1 | 30.04 | 95.32 | 94.88 | 94.64 | 94.44 | 94.16 | 93.96 | 93.84 |
| 71 | 31.32 | 97.12 | 96.48 | 96.24 | 96.24 | 96.2 | 96.16 | 96.12 |
| 141 | 31.32 | 97.04 | 96.56 | 96.44 | 96.24 | 96.08 | 96.08 | 96.12 |
| 211 | 31.32 | 97 | 96.44 | 96.36 | 96.32 | 96.24 | 96.08 | 96.08 |
| 281 | 31.32 | 96.96 | 96.4 | 96.32 | 96.2 | 96.12 | 96.12 | 96.04 |
| 351 | 31.32 | 97 | 96.32 | 96.2 | 96.2 | 96.2 | 96 | 96.08 |
| 421 | 31.32 | 97 | 96.32 | 96.08 | 96.12 | 96.2 | 96.24 | 96 |
| 491 | 31.32 | 97 | 96.28 | 96.08 | 96.16 | 96.12 | 96.2 | 96.16 |

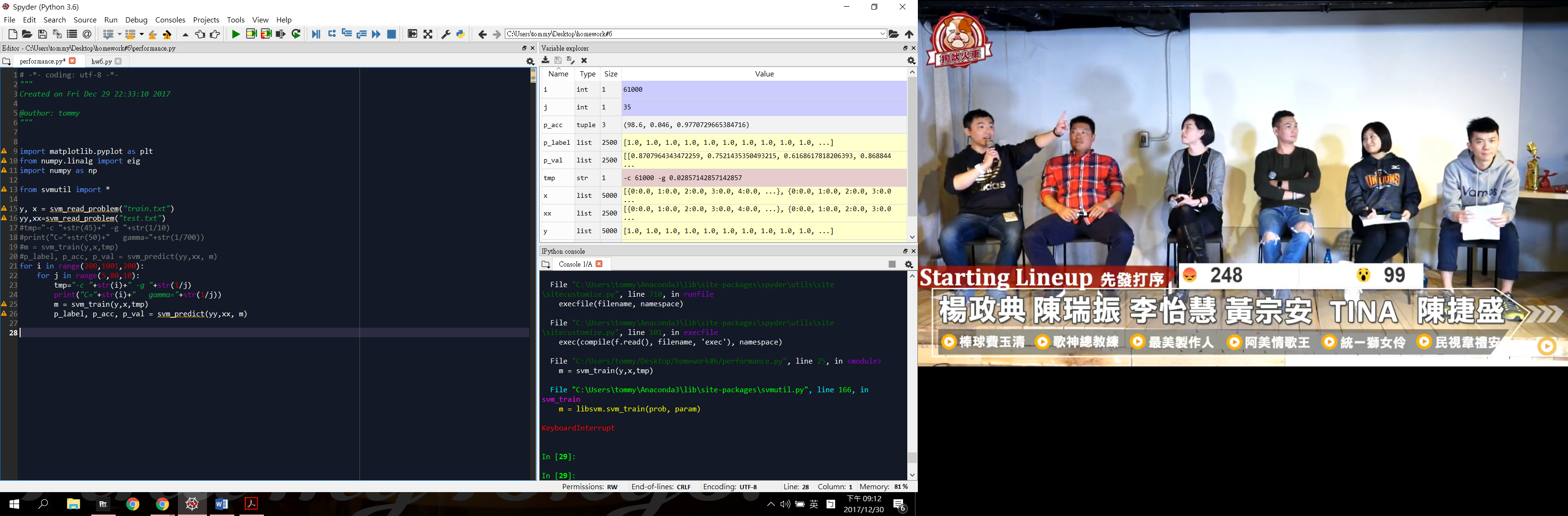
So according to Chart1, we can limit the 1/gamma range below 700. C doesn’t influence much at results.

Chart2:

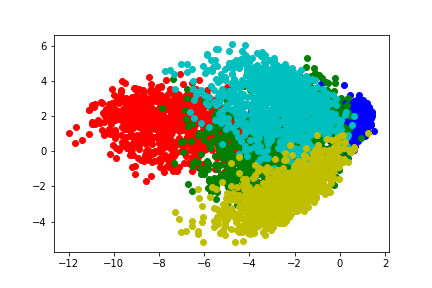
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C,1/gamma | 5 | 15 | 25 | 35 | 45 | 55 | 65 | 75 |
| 5 | 69.36 | 96.96 | 98.44 | 98.6 | 98.52 | 98.48 | 98.4 | 98.28 |
| 200 | 69.36 | 96.96 | 98.44 | 98.6 | 98.52 | 98.48 | 98.4 | 98.28 |
| 400 | 69.36 | 96.96 | 98.44 | 98.6 | 98.52 | 98.48 | 98.4 | 98.28 |
| 600 | 69.36 | 96.96 | 98.44 | 98.6 | 98.52 | 98.48 | 98.4 | 98.28 |
| 800 | 69.36 | 96.96 | 98.44 | 98.6 | 98.52 | 98.48 | 98.4 | 98.28 |
| 1000 | 69.36 | 96.96 | 98.44 | 98.6 | 98.52 | 98.48 | 98.4 | 98.28 |

According to Chart2, results has a peak at 1/gamma=35.

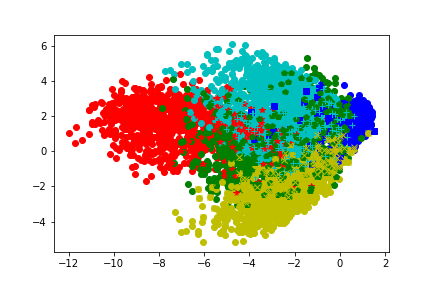
It’s how I get previous charts.



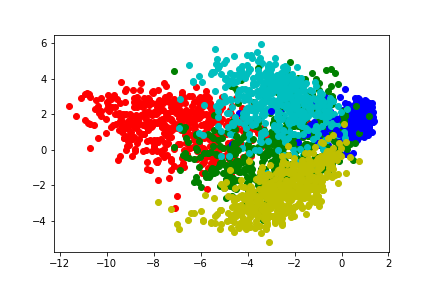
PLOT1:Training data on 2D space



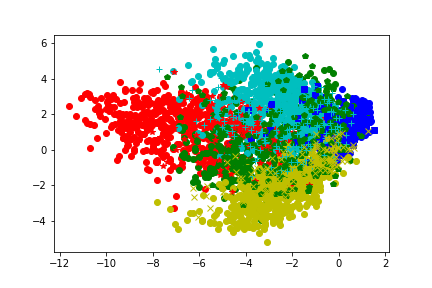
PLOT2:Training data on 2D space with support vectors



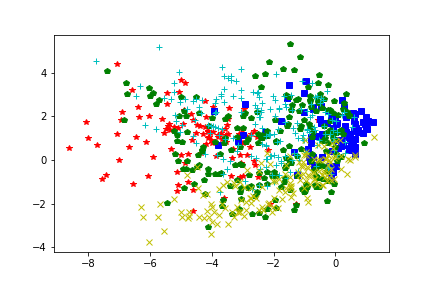
PLOT3:Testing data on 2D space



PLO4:TTesting data on 2D space with support vectors



PLOT5:Support vetors



心得:

之前學的都是上課內容交的公式觀念等等，很難想像這些理論真的能夠做到這些分類，這次把PCA跟SVM實做出來也讓我對這些數學model更加熟悉。 從PCA投影出的圖看的出來，果然同一個class的data會聚集在同一區，不過SVM因為有五個class，並且從784d-2d，比較難看得出gap。