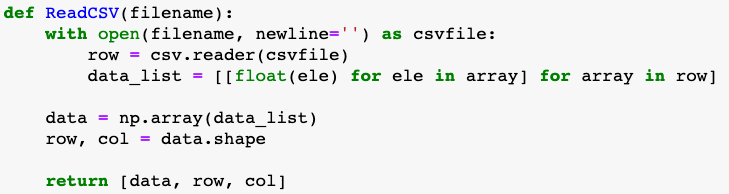
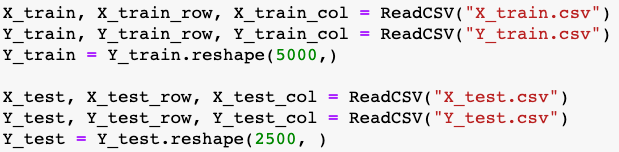
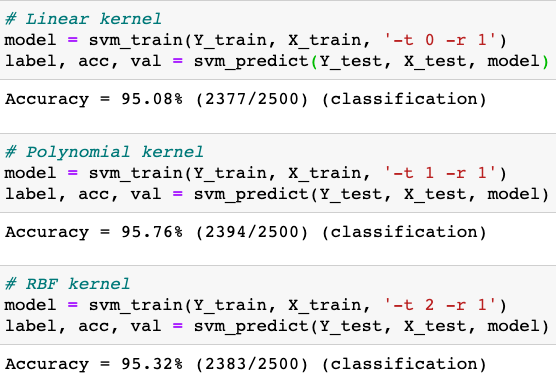
ML HW05-2 – SVM

309553048\_多工所\_陳柏丞

* **Code Explanation：**

1. ReadCSV(filename)  
   
2. Data Prepare  
   
3. Q1.  
     
   -t kernel\_type : set type of kernel function (default 2)

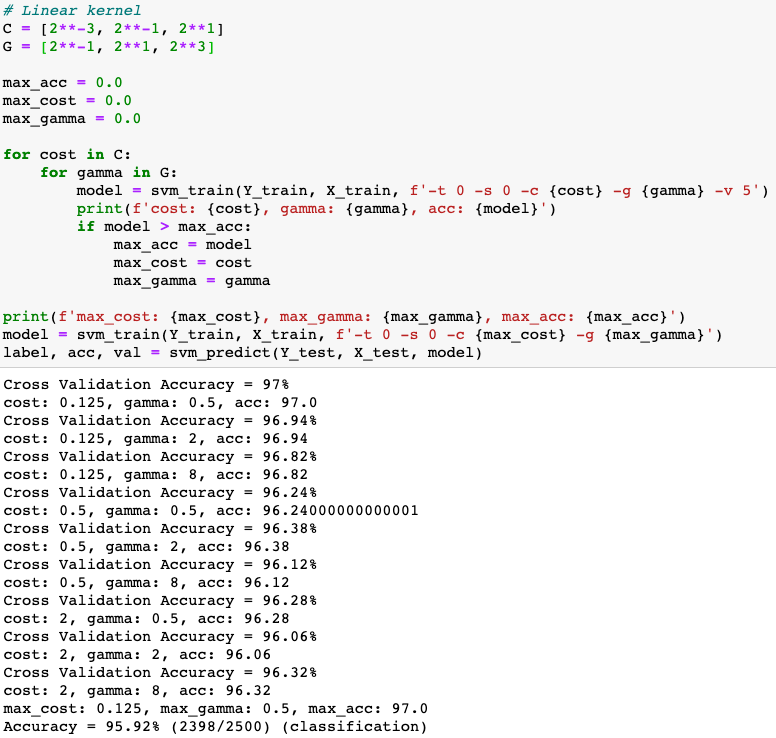
0 -- linear: u'\*v

1 -- polynomial: (gamma\*u'\*v + coef0)^degree

2 -- radial basis function: exp(-gamma\*|u-v|^2)

3 -- sigmoid: tanh(gamma\*u'\*v + coef0)

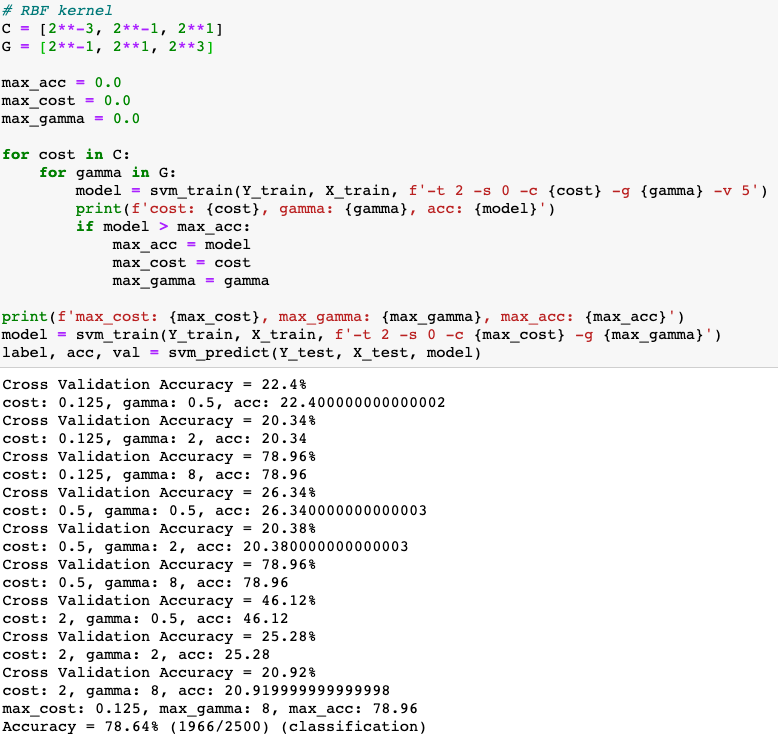
-r coef0 : set coef0 in kernel function (default 0)

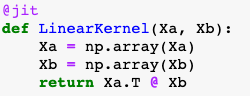
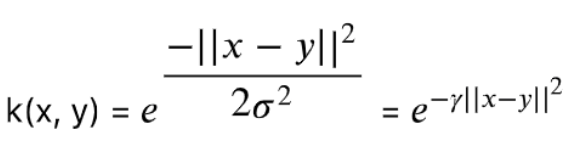
1. Q2  
     
     
   -g gamma : set gamma in kernel function (default 1/num\_features)  
     
   -c cost : set the parameter C of C-SVC, epsilon-SVR, and nu-SVR (default 1)  
     
   -s svm\_type : set type of SVM (default 0)  
   0 -- C-SVC

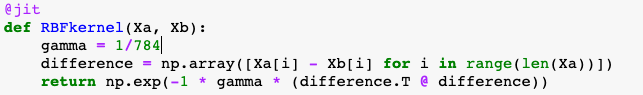
1 -- nu-SVC

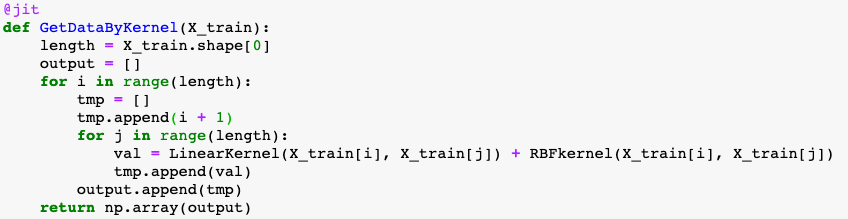
2 -- one-class SVM

3 -- epsilon-SVR

4 -- nu-SVR  


1. LinearKernel  
     
   
2. RBFkernel  
   

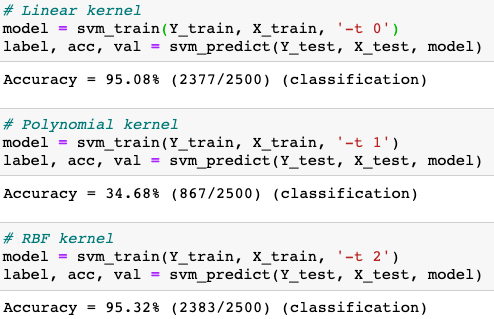


1. GetDataByKernel  
   
2. After kernel  
     
     
   

* **Experiment settings and results：**

1. Q1：Among three kernels, the performance speed：Linear > Polynomial > RBF. I found if I use SVR model, the RBF would be slower, but more accurate.

Result (Accuracy)：  
Linear：95.08%  
Polynomial：95.76%  
RBF：95.32%  
  
if I don’t set “-r”：

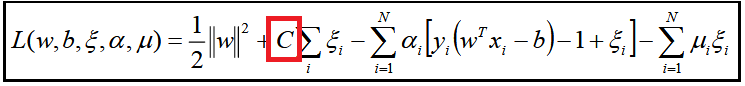
  
the accuracy in polynomial would be worse

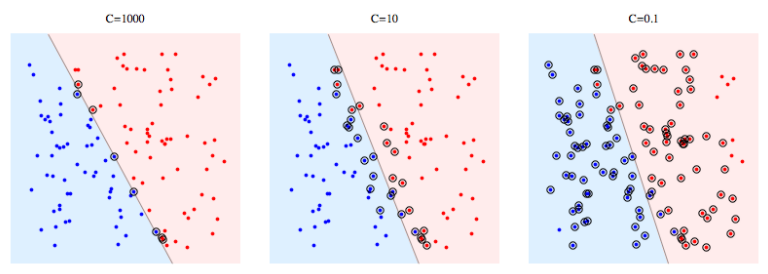
1. Q2：  
   In cost, I set three number：0.125, 0.5, 2  
   In gamma, I set three number：0.5, 2, 8  
   Result (Best parameter)：  
   Linear：max\_cost = 0.125, max\_gamma = 0.5, max\_acc = 97; after fitting, Accuracy = 95%  
   RBF：max\_cost = 0.125, max\_gamma = 8, max\_acc = 78.96; after fitting, Accuracy = 78.64%
2. Q3.  
   I didn’t have result because the error “segmentation fault”. After searching the error, it said the memory was not enough.

* **Observation and discussion：**

In Q1, if I didn’t set “-b”, meaning probability\_estimates: whether to train a SVC or SVR model for probability estimates, 0 or 1 (default 0), then, the accuracy in polynomial would be worse.

In Q2, I didn’t set many numbers to run because the runtime is long. However, if cost is bigger, support vectors is fewer and it’s easy to be overfitting. If cost is smaller, support vectors is more and the margin is bigger.





Gamma = How far the influence of a single training example reaches

If gamma is bigger, the influence of range by data points is closer and it’s easy to be overfitting. If gamma is smaller, the influence of range by data points is more far.

In Q3, I have two questions, one is the performance of GetDataByKernel is bad, the other, I could train the model, but I couldn’t predict the testing data. The error is ”segmentation default”. Thus, I thought the data produced by GetDataByKernel, occupying too much memory to work successfully when predicting the test data. Then, I use library “lamba” to solve the slow performance and its performance is obvious improve.