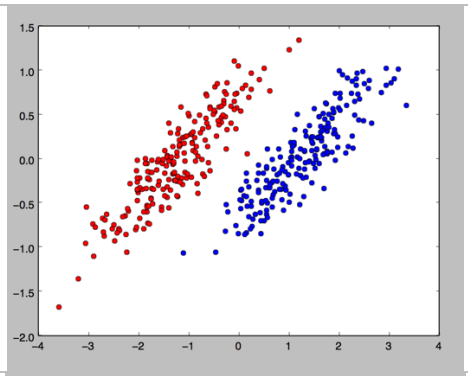
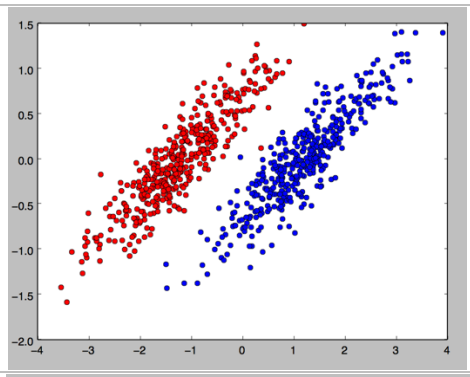
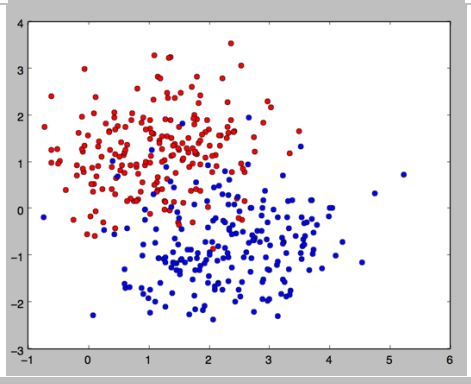
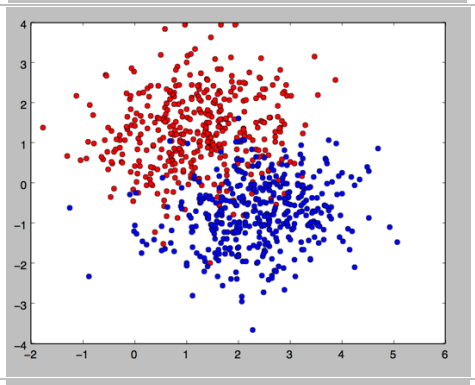
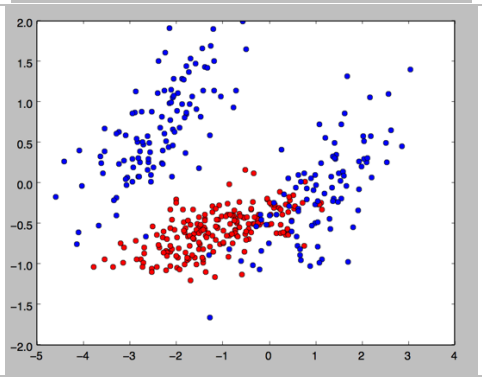
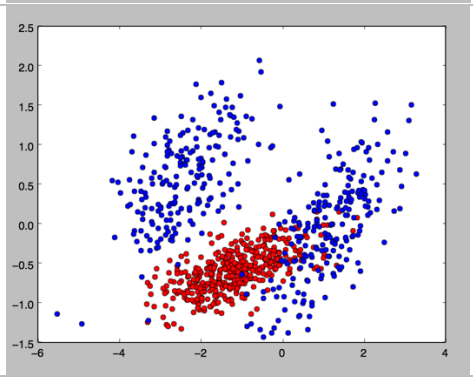
Machine Learning Assignment 2

Liang Shuailong 1000829

Task: SVM and Logistic Regression

The data from three datasets is visualized in the graphs below.

Dataset	Train	Test
A		
B		
C		

1.
When linear SVM is used, the result acquired is shown in table 1.

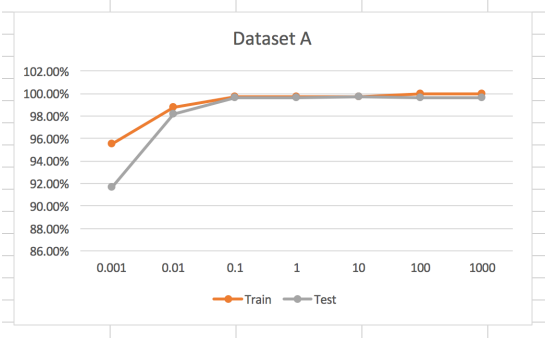
Table 1 Linear SVM primal and dual form result

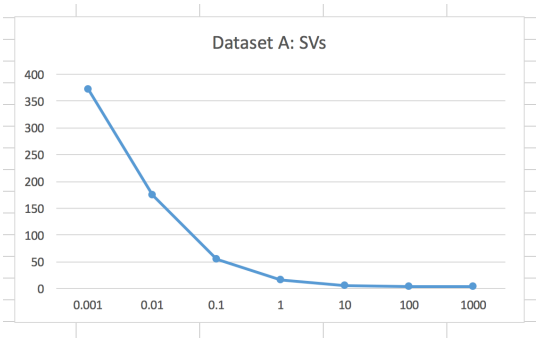
C = 1.0	Dual or Primal	Train accuracy	Test accuracy
A	Primal	99.75%	99.63%
	Dual	99.75%	99.63%
B	Primal	91.25%	92.00%
	Dual	91.25%	91.88%
C	Primal	85.75%	83.50%
	Dual	85.75%	83.63%

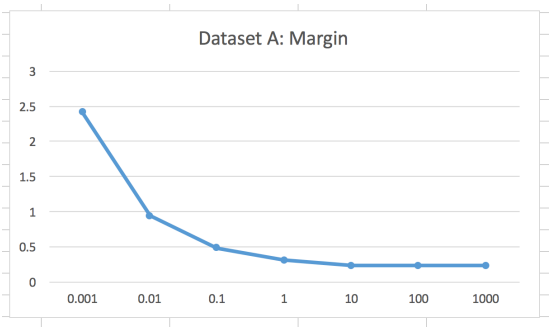
The different between primal form SVM and dual form SVM is very minor, so the implementation is correct. For the experiment result, we can see that since dataset A is linear separable with very small slack variables, the result is pretty good; dataset B has moderate good result since the data points are more scattered; dataset C is not quite linearly separable, so the result is the worst.

2.
The result is shown in the table below. We can expect that as C increases, the model is more prone to over-fitting problem, so the training accuracy will keep increasing and the test accuracy may decease after the optimal C. The number of Support Vectors may decrease or keep constant since the classifier is linear, and the margin will also decrease to make the classifier classify more training points correctly.

Dataset A				
C	Train	Test	SVs	Margin
0.001	95.50%	91.63%	372	**2.4034**
0.01	98.75%	98.13%	174	0.9480
0.1	99.75%	99.63%	55	0.5066
1	99.75%	99.63%	16	0.3097
10	99.75%	**99.75%**	5	0.1916
100	**100.00%**	99.63%	3	0.0865
1000	**100.00%**	99.63%	3	0.0865

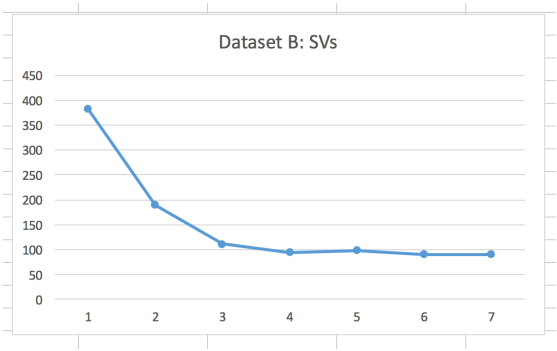


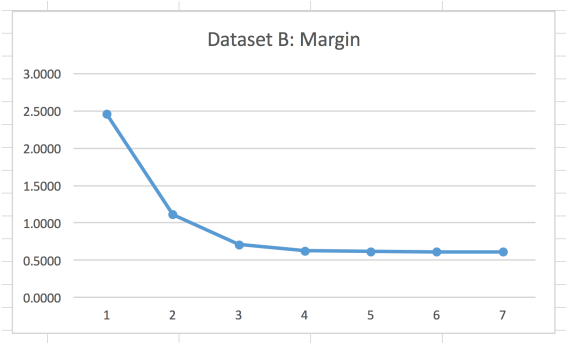




For dataset A, as C increases, training accuracy and test accuracy both increase. The number of support vectors keeps constant in general as C increase. Margin deceases gradually as C increases.

Dataset B				
C	Train	Test	SVs	Margin
0.001	**92.00%**	91.63%	382	**2.4647**
0.01	91.75%	92.00%	190	1.1128
0.1	91.50%	**92.38%**	111	0.7112
1	91.25%	92.00%	94	0.6255
10	91.25%	92.13%	99	0.6176
100	91.25%	92.13%	91	0.6120
1000	91.25%	92.13%	90	0.6120

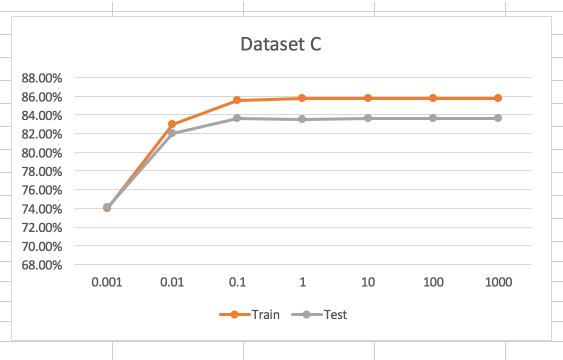


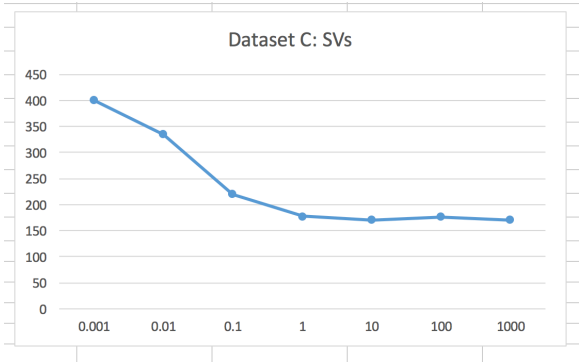


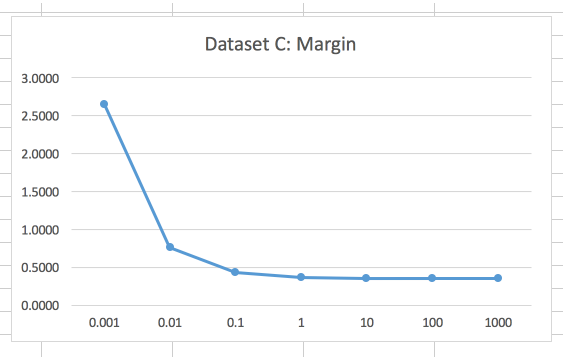
For dataset B,

Dataset C				
C	Train	Test		

0.001	74.00%	74.13%	400	**5.3039**
0.01	83.00%	82.00%	335	0.9751
0.1	85.50%	**83.63%**	220	0.4772
1	**85.75%**	83.50%	177	0.3628
10	**85.75%**	**83.63%**	170	0.3445
100	**85.75%**	**83.63%**	176	0.3445
1000	**85.75%**	**83.63%**	170	0.3445







For dataset C, as C increase, training and test accuracy increase first, and decrease a little bit. The number of support vectors keeps constant in general, and the margin decreases, as C grows.

The experiment results are in line with our expectations generally. The test accuracy decreases very little as C increases, this may be due to that the C is not big enough to cause over-fitting problem.

3.

It is not a good idea to optimize the value of C by maximizing the margin on the training set. This approach may choose the smaller C, but the performance may not be the best. A reasonable approach is to use a cross-validation set. First divide the training set into training set and cv set. The size may be 4:1. Then we choose the C that performs best on the cv set as our optimal C. Using this method, the result is shown in the table below.

	Train	Test	Best C
A	99.75%	99.75%	4.096
B	92.00%	92.00%	0.002
C	85.75%	83.63%	8.192

The result is comparable to the result in problem 1.

4.

In this problem RBF kernel and Polynomial kernel are used. The result is shown in the table below.

kernel='rbf', C=1.0	Train	Test
A	99.50%	99.50%
B	91.50%	92.13%
C	93.25%	92.00%
kernel='poly', C=1.0	Train	Test
A	99.75%	99.88%
B	90.00%	89.63%
C	79.25%	78.00%
kernel='sigmoid', C=1.0, gamma=0.01, coef0=10.0		
A	95.00%	91.38%
B	83.75%	85.38%
C	72.75%	71.50%

From the table we can see that for dataset A, there is little difference using different kernels. However, for dataset C, there is no obvious line between two classes, so it is better to use the RBF kernel instead of linear kernel. The Polynomial kernel is a little worse than linear kernel for dataset C.

5.

The result when learning rate = 0.001 is shown in the tables below.

A	**Learning rate=0.001**		
Lambda	Train accuracy	Test Accuracy	Loss
0.001	**99.75%**	**99.63%**	9.6302
0.01	99.00%	98.75%	30.1770
0.1	97.25%	95.63%	91.1197
1	95.75%	92.38%	193.5409
10	94.50%	91.00%	263.6358
100	93.75%	91.13%	275.5364
1000	48.75%	48.38%	277.2050

B	**Learning rate=0.001**		
Lambda	Train accuracy	Test Accuracy	Loss
0.001	91.75%	**91.88%**	90.1886
0.01	91.75%	**91.88%**	92.6801
0.1	91.75%	**91.88%**	119.3099

1	91.25%	**91.88%**	203.5209
10	**92.00%**	91.38%	264.7552
100	87.50%	86.50%	276.2307
1000	50.50%	49.50%	277.0470

C	**Learning rate=0.001**		
Lambda	Train accuracy	Test Accuracy	Loss
0.001	83.50%	**83.63%**	161.4867
0.01	**83.75%**	**83.63%**	166.7524
0.1	82.75%	83.25%	204.5419
1	79.25%	80.38%	259.2937
10	75.00%	77.38%	275.0398
100	63.75%	69.88%	277.1186
1000	64.25%	65.63%	277.2353

The result when learning rate is 0.0001 is shown in the tables below.

A	**Learning rate=0.0001**		
Lambda	Train accuracy	Test Accuracy	Loss
0.001	**98.75%**	**97.75%**	34.8146
0.01	**98.75%**	**97.75%**	41.8285
0.1	97.25%	95.63%	91.2717
1	95.75%	92.38%	193.5911
10	95.25%	91.50%	263.0578
100	95.25%	91.50%	275.7324
1000	94.25%	90.88%	277.1084

B	**Learning rate=0.0001**		
Lambda	Train accuracy	Test Accuracy	Loss
0.001	90.75%	91.13%	93.4725
0.01	90.75%	91.50%	95.4406
0.1	91.75%	**91.88%**	119.8644
1	**92.00%**	91.63%	204.1643
10	87.50%	87.75%	265.3107
100	90.00%	90.00%	275.9721
1000	89.75%	89.25%	277.1436

C	**Learning rate=0.0001**		
Lambda	Train accuracy	Test Accuracy	Loss
0.001	**84.75%**	83.25%	167.0056
0.01	84.50%	**83.50%**	171.0160
0.1	82.75%	83.25%	204.5782
1	78.25%	80.38%	259.3759

10	74.75%	77.38%	275.1076
100	74.75%	77.75%	277.0477
1000	39.25%	33.63%	277.2479