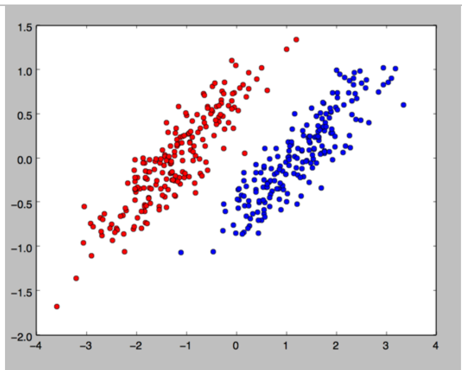
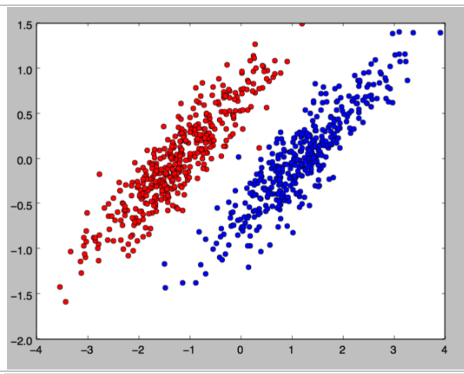
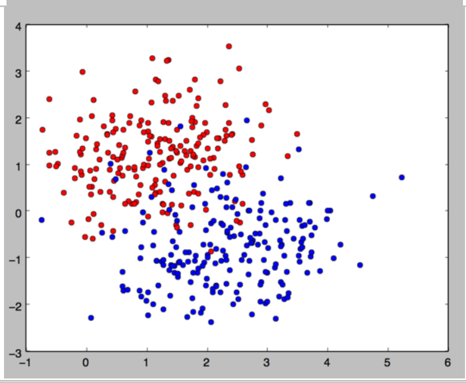
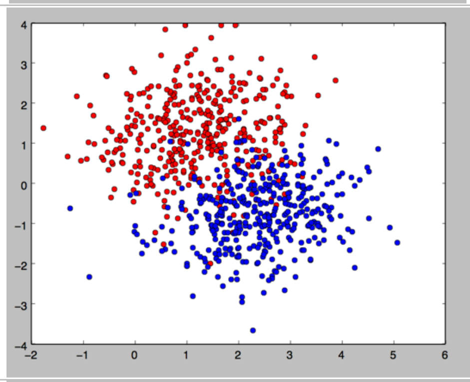
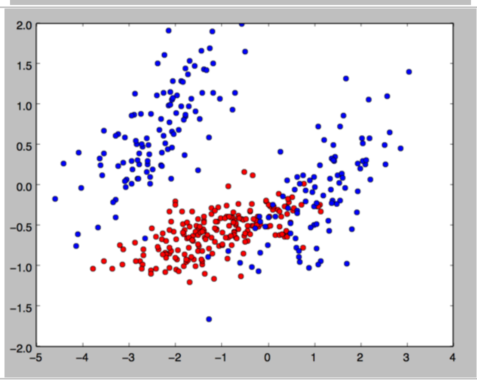
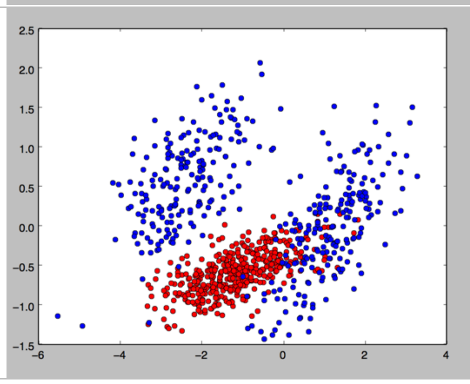
Machine Learning Assignment 2

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%

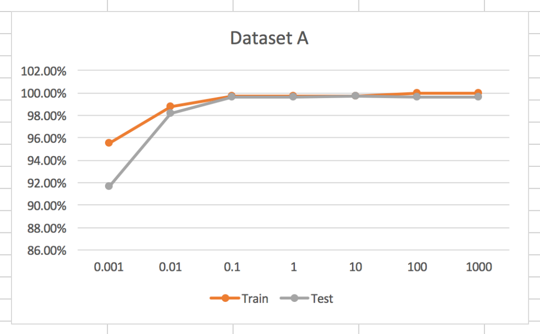
The different between primal form SVM and dual form SVM is very minor, so the implementation is correct. For the experiment result, we can see that since dataset A is linear separable with very small slack variables, the result is pretty good; dataset B has moderate good result since the data points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

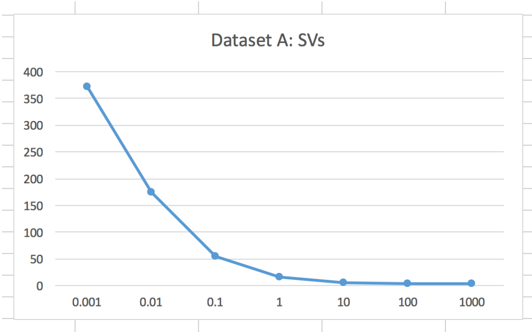
2.
We can expect that as C increases, the model is more prone to over-fitting problem, so the training accuracy will keep increasing and the test accuracy may decease after the optimal C. The margin will decrease to make the classifier classify more training points correctly, and so will the number of Support Vectors lying within the margin.

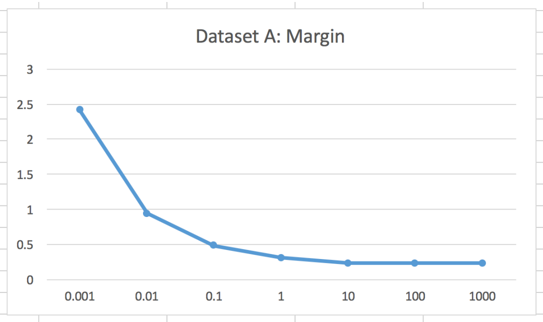
The experiment results are shown in the tables 2-4, and the data are also represented using line charts below.

Table 2 Dataset A with difference C

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



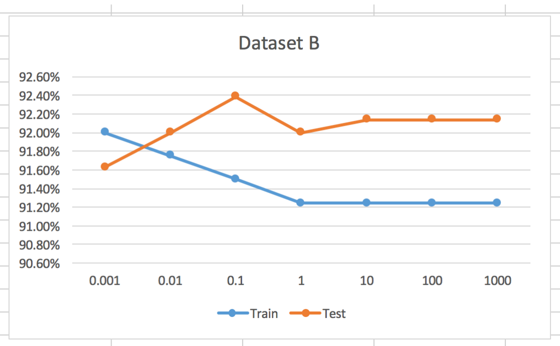


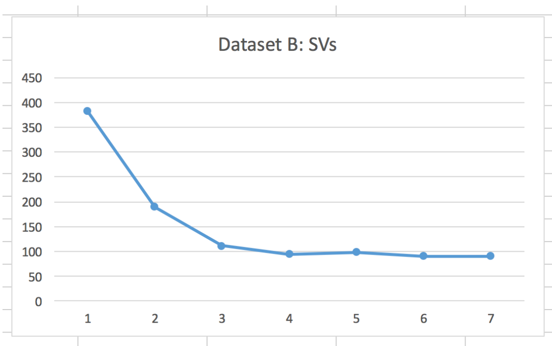


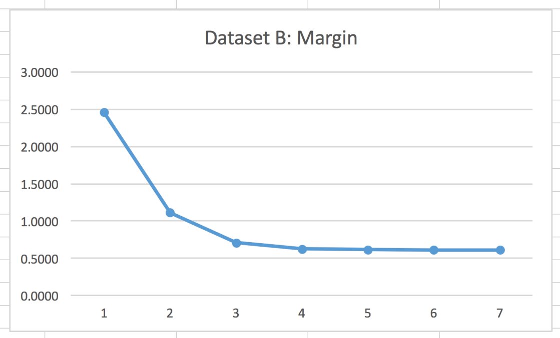
For dataset A, as C increases, training accuracy and test accuracy both increase to a plateau. As for why the test accuracy will not decrease with large C, the reason may be that although the model may overfit, the test data have a similar distribution as the training data, so the performance does not hurt. The number of support vectors and margin deceases gradually as we expect before.

Table 3 Dataset B with different C

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



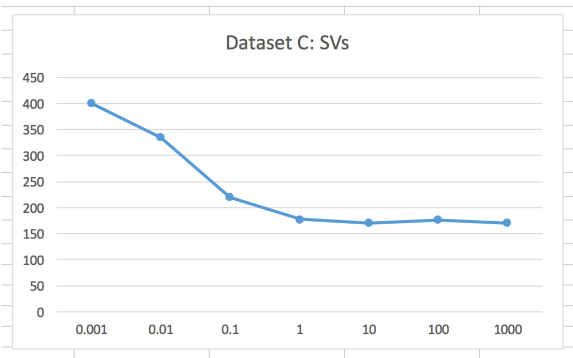


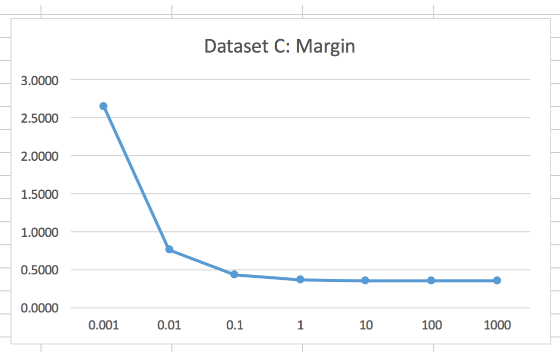


For dataset B, with the increase of C, train accuracy decreases a little, but in general, train accuracy and test accuracy are not changed very much, which means that regularization does not have a significant impact on this dataset. The margin and number of SVs are both decrease gradually as expected.

Table 4 Dataset C with different C

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





Dataset C has similar results with dataset A, which we have analyzed before.

3.

It is not a good idea to optimize the value of C by maximizing the margin on the training set. As we analyzed in problem 2, the smaller the C, the bigger the margin. Therefore, 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 table 5.

Table 5 Optimal C result on three datasets

	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 with C=1.0 in table 1.

4.

For this problem RBF kernel, aside from linear kernel, polynomial kernel and sigmoid kernel are used. The kernel definitions are as follows:

- linear: $\langle x, x' \rangle$.
- polynomial: $(\gamma \langle x, x' \rangle + r)^d$. d is specified by keyword degree, r by coef0.
- rbf: $\exp(-\gamma |x - x'|^2)$. γ is specified by keyword gamma, must be greater than 0.
- sigmoid ($\tanh(\gamma \langle x, x' \rangle + r)$), where r is specified by coef0.

The experiment result is shown in table 6.

Table 6 SVM with different kernels used

kernel='rbf', C=1.0, gamma=0.5	Train	Test
A	99.50%	99.50%
B	91.50%	92.13%
C	93.25%	92.00%
kernel='poly', C=1.0, gamma=0.5, coef0=0, d=3	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 above, we can see that when RBF kernel is used, each of the three datasets can have a better performance compared with the linear kernel, especially for dataset C. RBF kernel can make a good result when the dataset is not quite linear separable. For polynomial kernel, only dataset A has a better result compared with linear SVM. The other two are slightly worse. Sigmoid kernel has a worse result for each of the three datasets. Maybe if we tune the parameters the result will be better.

5.

The result when learning rate = 0.001 is shown in tables 7-9.

Table 7 Learning rate=0.001 Dataset A

A	**Learning rate=0.001**		
Lambda	Train accuracy	Test accuracy	Optimal objective
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

Table 8 Learning rate=0.001 Dataset B

B	**Learning rate=0.001**		
Lambda	Train accuracy	Test accuracy	Optimal objective
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

Table 9 Learning rate=0.001 Dataset C

C	**Learning rate=0.001**		
Lambda	Train accuracy	Test accuracy	Optimal objective
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 tables 10-12.

Table 10 Learning rate=0.0001 Dataset A

A	**Learning rate=0.0001**		
Lambda	Train accuracy	Test accuracy	Optimal objective
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

Table 11 Learning rate=0.001 Dataset B

B	**Learning rate=0.0001**		
Lambda	Train accuracy	Test accuracy	Optimal objective
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

Table 12 Learning rate=0.001 Dataset C

C	**Learning rate=0.0001**		
Lambda	Train accuracy	Test accuracy	Optimal objective
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

From the tables above, for optimal Lambda, logistic regression has similar results to SVM. With the increase of Lambda, the train accuracy and test accuracy both decrease since the model is more prone to underfit.