Rajesh Sakhamuru
7/20/2020

CS6140 – Assignment 4 – Perceptrons, Regularization and SVMs

1.2 – Perceptron and Dual Perceptron

Perceptron Dataset – Single Perceptron:

```
--------------------------------------------------
Results Table:
          Fold #  Train Accuracy  Train Precision  Train Recall  Test Accuracy  Test Precision  Test Recall
0              1             1.0              1.0           1.0       1.000000       1.000000          1.0
1              2             1.0              1.0           1.0       1.000000       1.000000          1.0
2              3             1.0              1.0           1.0       1.000000       1.000000          1.0
3              4             1.0              1.0           1.0       0.990000       0.980769          1.0
4              5             1.0              1.0           1.0       1.000000       1.000000          1.0
5              6             1.0              1.0           1.0       1.000000       1.000000          1.0
6              7             1.0              1.0           1.0       1.000000       1.000000          1.0
7              8             1.0              1.0           1.0       1.000000       1.000000          1.0
8              9             1.0              1.0           1.0       1.000000       1.000000          1.0
9             10             1.0              1.0           1.0       1.000000       1.000000          1.0
10          Mean             1.0              1.0           1.0       0.999000       0.998077          1.0
11  Std Deviation            0.0              0.0           0.0       0.003162       0.006081          0.0

Columns are actual value, Rows are predicted value
Training Confusion Matrix:
      -1     1
-1  4707     0
 1     0  4293

Testing Confusion Matrix:
     -1    1
-1  522    0
 1    1  477
```

Perceptron Dataset – Dual Perceptron:

```
Results Table:
          Fold #  Train Accuracy  Train Precision  Train Recall  Test Accuracy  Test Precision  Test Recall
0              1             1.0              1.0           1.0       1.000000       1.000000          1.0
1              2             1.0              1.0           1.0       1.000000       1.000000          1.0
2              3             1.0              1.0           1.0       1.000000       1.000000          1.0
3              4             1.0              1.0           1.0       1.000000       1.000000          1.0
4              5             1.0              1.0           1.0       1.000000       1.000000          1.0
5              6             1.0              1.0           1.0       0.980000       0.953488          1.0
6              7             1.0              1.0           1.0       1.000000       1.000000          1.0
7              8             1.0              1.0           1.0       1.000000       1.000000          1.0
8              9             1.0              1.0           1.0       1.000000       1.000000          1.0
9             10             1.0              1.0           1.0       1.000000       1.000000          1.0
10          Mean             1.0              1.0           1.0       0.998000       0.995349          1.0
11  Std Deviation            0.0              0.0           0.0       0.006325       0.014708          0.0

Columns are actual value, Rows are predicted value
Training Confusion Matrix:
      -1     1
-1  4707     0
 1     0  4293

Testing Confusion Matrix:
     -1    1
-1  521    0
 1    2  477
```

Both Perceptron and Dual-Perceptron algorithms on the linearly separable dataset perform almost identically as can be seen from the performance data above.

1.3 - Kernelizing Dual Perceptron

Two Spiral Dataset - Dual Perceptron with a Linear Kernel:

```
Results Table:
           Fold #  Train Accuracy  Train Precision  Train Recall  Test Accuracy  Test Precision  Test Recall
0               1        0.673333         0.674833      0.671840       0.640000        0.614035     0.714286
1               2        0.665556         0.661435      0.662921       0.650000        0.678571     0.690909
2               3        0.667778         0.666667      0.668151       0.700000        0.677966     0.784314
3               4        0.655556         0.654709      0.651786       0.680000        0.685185     0.711538
4               5        0.678889         0.671024      0.690583       0.610000        0.674419     0.537037
5               6        0.667778         0.660131      0.679372       0.710000        0.777778     0.648148
6               7        0.674444         0.669643      0.674157       0.610000        0.633333     0.690909
7               8        0.672222         0.674672      0.679121       0.670000        0.636364     0.622222
8               9        0.671111         0.671739      0.680617       0.640000        0.619048     0.565217
9              10        0.660000         0.671082      0.659436       0.710000        0.608696     0.717949
10           Mean        0.668667         0.667593      0.671798       0.662000        0.660539     0.668253
11  Std Deviation        0.006965         0.006736      0.011511       0.037947        0.050630     0.075510

Columns are actual value, Rows are predicted value
Training Confusion Matrix:
      -1     1
-1  2995  1477
 1  1505  3023

Testing Confusion Matrix:
     -1    1
-1  328  166
 1  172  334
```

As we can see here, the data is not linearly seperable.

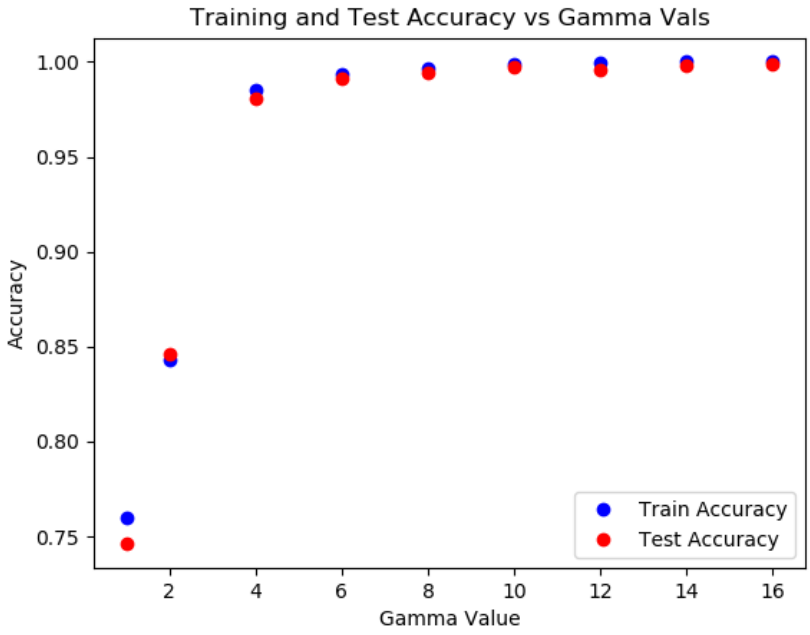
Two Spiral Dataset – Dual Perceptron with Gaussian (RBF) kernel:

```
Results Table:
           Fold #  Train Accuracy  Train Precision  Train Recall  Test Accuracy  Test Precision  Test Recall
0               1        0.998889         0.997788      1.000000       0.990000        0.980000     1.000000
1               2        0.998889         1.000000      0.997792       1.000000        1.000000     1.000000
2               3        0.998889         1.000000      0.997773       1.000000        1.000000     1.000000
3               4        0.997778         1.000000      0.995465       1.000000        1.000000     1.000000
4               5        0.998889         1.000000      0.997821       0.990000        1.000000     0.975610
5               6        0.997778         1.000000      0.995556       1.000000        1.000000     1.000000
6               7        0.997778         0.997807      0.997807       0.980000        0.977273     0.977273
7               8        0.997778         1.000000      0.995455       1.000000        1.000000     1.000000
8               9        0.997778         0.997778      0.997778       0.990000        0.980392     1.000000
9              10        0.996667         0.995575      0.997783       1.000000        1.000000     1.000000
10           Mean        0.998111         0.998895      0.997323       0.995000        0.993766     0.995288
11  Std Deviation        0.000750         0.001564      0.001436       0.007071        0.010069     0.009941

Columns are actual value, Rows are predicted value
Training Confusion Matrix:
      -1     1
-1  4495    12
 1     5  4488

Testing Confusion Matrix:
     -1    1
-1  497    2
 1    3  498
```

I used a gamma value of 9 for the RBF Kernel, where gamma = 1/(2(sigma)^2, and if sigma=.236 that means gamma=9. The way I chose the gamma value is to try various gamma values and do 10-fold cross validation at each gamma value for the accuracy results. As we can see from t he graph to the right, the Gamma values above 9 have very diminishing results and if it's any higher it will probably result in overfitting.



2 – Regularized Logistic Regression

1. The bias term w_0 should not be included in the regularization because since it is a constant value, and by regularizing it we would be moving the predicted plane an arbitrary amount away from the expected/true plane resulting in poor performance of the model.

Logistic Regression vs. Regularized Logistic Regression:
Spambase Dataset:
Without Regularization:

Results Table:

	Fold #	Train Accuracy	Train Precision	Train Recall	Train Log-Loss	Test Accuracy	Test Precision	Test Recall	Test Log-Loss
0	1	0.926346	0.926376	0.883934	0.209763	0.928261	0.903955	0.909091	0.224714
1	2	0.924656	0.925854	0.879363	0.211758	0.928261	0.950920	0.861111	0.203707
2	3	0.924414	0.924697	0.881387	0.215282	0.945652	0.933735	0.917160	0.169919
3	4	0.925863	0.926974	0.877882	0.205683	0.917391	0.929293	0.884615	0.268313
4	5	0.927071	0.929569	0.884662	0.208717	0.902174	0.873333	0.834395	0.242330
5	6	0.927554	0.928389	0.885366	0.204582	0.919565	0.909639	0.872832	0.281934
6	7	0.928037	0.926876	0.887047	0.212621	0.932609	0.937143	0.891304	0.216123
7	8	0.926105	0.925470	0.882026	0.210741	0.923913	0.934426	0.881443	0.218184
8	9	0.928278	0.930968	0.883650	0.210785	0.919565	0.913295	0.877778	0.227300
9	10	0.927795	0.931507	0.880395	0.205091	0.917391	0.913514	0.884817	0.280316
10	Mean	0.926612	0.927668	0.882571	0.209502	0.923478	0.919925	0.881455	0.233284
11	Std Deviation	0.001368	0.002345	0.002867	0.003502	0.011448	0.022017	0.023270	0.035605

Columns are actual value, Rows are predicted value
Training Confusion Matrix:

	0	1
0	23969	1916
1	1123	14402

Testing Confusion Matrix:

	0	1
0	2650	214
1	138	1598

With regularization:

```
Results Table:
          Fold #  Train Accuracy  Train Precision  Train Recall  Train Log-Loss  Test Accuracy  Test Precision  Test Recall  Test Log-Loss
0              1        0.927554         0.928941      0.883292        0.210898       0.919565        0.911111     0.886486       0.219755
1              2        0.929486         0.928526      0.891318        0.203464       0.930435        0.913580     0.891566       0.301689
2              3        0.926829         0.926376      0.885015        0.212437       0.910870        0.936306     0.825843       0.205276
3              4        0.927071         0.927191      0.883907        0.207762       0.930435        0.952663     0.870270       0.241459
4              5        0.924173         0.925422      0.877614        0.215558       0.928261        0.932584     0.887701       0.171003
5              6        0.925622         0.928525      0.878304        0.211724       0.941304        0.925134     0.930108       0.196242
6              7        0.924897         0.926214      0.878993        0.213190       0.932609        0.942529     0.886486       0.190280
7              8        0.926588         0.927036      0.881115        0.208688       0.926087        0.940860     0.883838       0.241689
8              9        0.929486         0.928892      0.889025        0.207012       0.915217        0.908571     0.873626       0.257483
9             10        0.928037         0.927985      0.888687        0.202870       0.917391        0.886076     0.875000       0.284933
10          Mean        0.926974         0.927511      0.883727        0.209360       0.925217        0.924941     0.881093       0.230981
11  Std Deviation        0.001772         0.001243      0.004810        0.004162       0.009291        0.020049     0.025569       0.042275

Columns are actual value, Rows are predicted value
Training Confusion Matrix:
       0      1
0  23965   1897
1   1127  14421

Testing Confusion Matrix:
      0     1
0  2659   215
1   129  1597
```

As you can see for the Spambase dataset, model performance is almost identical, but presumably due to regularization, if given edge-case test data we would get better results because of a reduction in overfitting.

Diabetes Dataset:

Without Regularization:

```
Results Table:
          Fold #  Train Accuracy  Train Precision  Train Recall  Train Log-Loss  Test Accuracy  Test Precision  Test Recall  Test Log-Loss
0              1        0.788712         0.748634      0.578059        0.465777       0.766234        0.809524     0.548387       0.522673
1              2        0.774240         0.724868      0.568465        0.479435       0.844156        0.894737     0.629630       0.401479
2              3        0.774240         0.727273      0.564315        0.479266       0.831169        0.937500     0.555556       0.407770
3              4        0.780029         0.734375      0.582645        0.474592       0.792208        0.727273     0.615385       0.446541
4              5        0.784370         0.747423      0.591837        0.466504       0.740260        0.571429     0.521739       0.516104
5              6        0.797395         0.759358      0.599156        0.453935       0.649351        0.590909     0.419355       0.629090
6              7        0.782923         0.739796      0.594262        0.466597       0.766234        0.650000     0.541667       0.521680
7              8        0.781476         0.737113      0.588477        0.469969       0.779221        0.700000     0.560000       0.484012
8              9        0.780029         0.732620      0.573222        0.475198       0.766234        0.739130     0.586207       0.439485
9             10        0.780664         0.738220      0.580247        0.472507       0.813333        0.739130     0.680000       0.461257
10          Mean        0.782408         0.738968      0.582068        0.470378       0.774840        0.735963     0.565792       0.483009
11  Std Deviation        0.006808         0.010447      0.011428        0.007656       0.054717        0.119317     0.070240       0.067913

Columns are actual value, Rows are predicted value
Training Confusion Matrix:
      0     1
0  4004  1008
1   496  1404

Testing Confusion Matrix:
     0    1
0  444  117
1   56  151
```

With Regularization:

```
Results Table:
           Fold #  Train Accuracy  Train Precision  Train Recall  Train Log-Loss  Test Accuracy  Test Precision  Test Recall  Test Log-Loss
0               1        0.790159         0.754011      0.587500        0.467843       0.740260        0.681818     0.535714       0.508282
1               2        0.784370         0.740741      0.583333        0.473880       0.779221        0.739130     0.607143       0.449302
2               3        0.775687         0.727273      0.587755        0.475747       0.831169        0.777778     0.608696       0.433552
3               4        0.768452         0.717514      0.535865        0.476092       0.805195        0.785714     0.709677       0.440442
4               5        0.791606         0.747368      0.596639        0.457639       0.701299        0.684211     0.433333       0.595645
5               6        0.777135         0.726804      0.582645        0.473158       0.818182        0.800000     0.615385       0.462016
6               7        0.777135         0.728723      0.570833        0.473692       0.805195        0.809524     0.607143       0.457697
7               8        0.780029         0.734694      0.590164        0.472930       0.766234        0.650000     0.541667       0.461712
8               9        0.790159         0.746114      0.600000        0.457820       0.753247        0.680000     0.607143       0.611939
9              10        0.782107         0.741117      0.593496        0.473390       0.786667        0.687500     0.500000       0.450840
10           Mean        0.781684         0.736436      0.582823        0.470219       0.778667        0.729568     0.576590       0.487143
11  Std Deviation        0.007488         0.011352      0.018406        0.006946       0.039513        0.059457     0.076251       0.064757

Columns are actual value, Rows are predicted value
Training Confusion Matrix:
      0     1
0  3997  1006
1   503  1406

Testing Confusion Matrix:
     0    1
0  443  113
1   57  155
```

As you can see for the Diabetes dataset, model performance is almost identical, but presumably due to regularization, if given edge-case test data we would get better results because of a reduction in overfitting.

Breast Cancer Dataset:

Without Regularization:

```
Results Table:
           Fold #  Train Accuracy  Train Precision  Train Recall  Train Log-Loss  Test Accuracy  Test Precision  Test Recall  Test Log-Loss
0               1        0.986328         0.989418      0.973958        0.055813       1.000000        1.000000     1.000000       0.019315
1               2        0.992188         1.000000      0.979058        0.046202       0.947368        1.000000     0.857143       0.112315
2               3        0.988281         0.989305      0.978836        0.049124       0.982456        1.000000     0.956522       0.080132
3               4        0.986328         0.989130      0.973262        0.053831       0.982456        0.961538     1.000000       0.036905
4               5        0.982422         0.978836      0.973684        0.052815       0.982456        1.000000     0.954545       0.067035
5               6        0.988281         0.994681      0.973958        0.052720       0.982456        0.952381     1.000000       0.046828
6               7        0.988281         0.994792      0.974490        0.054317       0.982456        1.000000     0.937500       0.083382
7               8        0.988281         0.989529      0.979275        0.049711       0.964912        0.947368     0.947368       0.078132
8               9        0.986328         0.989362      0.973822        0.050782       0.982456        0.954545     1.000000       0.079369
9              10        0.988304         0.989189      0.978610        0.040813       0.982143        1.000000     0.960000       0.161585
10           Mean        0.987502         0.990424      0.975895        0.050613       0.978916        0.981583     0.961308       0.076500
11  Std Deviation        0.002471         0.005483      0.002647        0.004459       0.013830        0.024017     0.044310       0.040002

Columns are actual value, Rows are predicted value
Training Confusion Matrix:
      0     1
0  3195    46
1    18  1862

Testing Confusion Matrix:
     0    1
0  353    8
1    4  204
```

With Regularization:

Results Table:

	Fold #	Train Accuracy	Train Precision	Train Recall	Train Log-Loss	Test Accuracy	Test Precision	Test Recall	Test Log-Loss
0	1	0.986328	0.989362	0.973822	0.052145	1.000000	1.000000	1.000000	0.026721
1	2	0.986328	0.989362	0.973822	0.053785	0.982456	1.000000	0.952381	0.034017
2	3	0.988281	0.989362	0.978947	0.047943	0.982456	1.000000	0.954545	0.092703
3	4	0.988281	0.994652	0.973822	0.051315	0.982456	0.954545	1.000000	0.040679
4	5	0.986328	0.989071	0.973118	0.052431	1.000000	1.000000	1.000000	0.029170
5	6	0.990234	0.994764	0.979381	0.048877	0.947368	0.894737	0.944444	0.077450
6	7	0.986328	0.984615	0.979592	0.038210	0.982456	1.000000	0.937500	0.179668
7	8	0.990234	0.994792	0.979487	0.046938	0.964912	1.000000	0.882353	0.130350
8	9	0.984375	0.984043	0.973684	0.054357	1.000000	1.000000	1.000000	0.020422
9	10	0.992203	1.000000	0.978261	0.046022	0.946429	0.962963	0.928571	0.103300
10	Mean	0.987892	0.991002	0.976394	0.049202	0.978853	0.981225	0.959980	0.073448
11	Std Deviation	0.002404	0.004987	0.002918	0.004818	0.020084	0.034949	0.039795	0.053107

Columns are actual value, Rows are predicted value

Training Confusion Matrix:

	0	1
0	3196	45
1	17	1863

Testing Confusion Matrix:

	0	1
0	353	8
1	4	204

For all three datasets: Spambase, Diabetes and Breast Cancer, regularization did not make much of a difference in terms of model performance, with almost identical results using 10-fold cross validation. The reason for this, is that regularization is meant to reduce overfitting of the model to the training dataset, but not directly improve performance in accuracy, precision and recall.

2.3 and 2.5 are in other PDF.

3.1 – Determining Model Hyper-parameters

Spambase Dataset:

Linear Kernel optimizing Accuracy:

	OuterFold	TrainAccuracy	TrainPrecision	TrainRecall	TestAccuracy	TestPrecision	TestRecall
0	1	0.932625	0.927990	0.900365	0.950000	0.939024	0.922156
1	2	0.935040	0.930495	0.903893	0.934783	0.937107	0.881657
2	3	0.933349	0.926369	0.902514	0.936957	0.952663	0.884615
3	4	0.936972	0.934045	0.905052	0.917391	0.888235	0.888235
4	5	0.935281	0.933548	0.897643	0.923913	0.927835	0.895522
5	6	0.933108	0.927399	0.900674	0.943478	0.928571	0.928571
6	7	0.935764	0.927861	0.908648	0.928261	0.915663	0.888889
7	8	0.934798	0.929434	0.901912	0.936957	0.931217	0.916667
8	9	0.936730	0.933502	0.903740	0.932609	0.917127	0.912088
9	10	0.937213	0.933504	0.903465	0.910870	0.906250	0.883249
10	Avg	0.935088	0.930415	0.902791	0.931522	0.924369	0.900165
11	StdDev	0.001645	0.002998	0.002976	0.011785	0.018307	0.017847

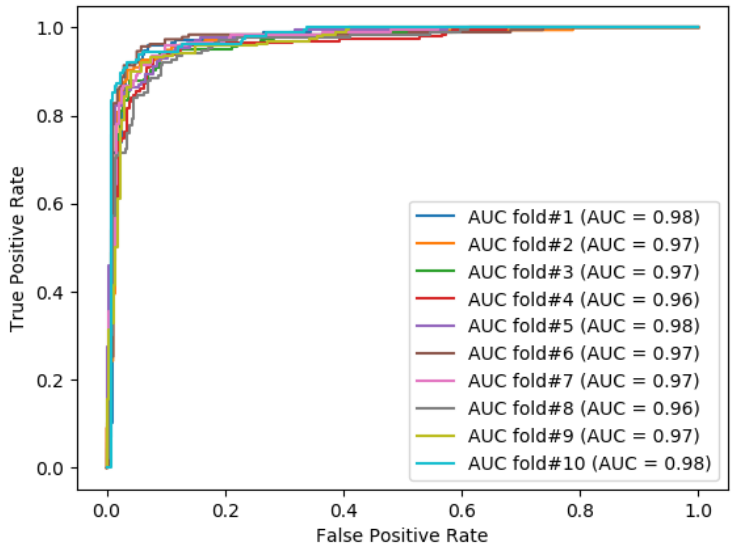
RBF Kernel optimizing Accuracy:

	OuterFold	cVal	gammaVal	TrainAccuracy	TrainPrecision	TrainRecall	TestAccuracy	TestPrecision	TestRecall
0	1	4096.000000	0.000031	0.954359	0.956030	0.927483	0.928261	0.921212	0.883721
1	2	4096.000000	0.000031	0.953876	0.951128	0.930717	0.923913	0.924855	0.879121
2	3	4096.000000	0.000031	0.957498	0.958647	0.932927	0.921739	0.910180	0.878613
3	4	4096.000000	0.000031	0.951702	0.946541	0.929012	0.928261	0.944444	0.880829
4	5	4096.000000	0.000031	0.955808	0.954203	0.932557	0.913043	0.903409	0.873626
5	6	4096.000000	0.000031	0.953151	0.953283	0.926380	0.945652	0.929348	0.934426
6	7	1024.000000	0.000031	0.941801	0.936318	0.913846	0.919565	0.926554	0.872340
7	8	4096.000000	0.000031	0.955808	0.951144	0.936663	0.908696	0.900621	0.847953
8	9	4096.000000	0.000031	0.952910	0.952081	0.926949	0.923913	0.890052	0.923913
9	10	4096.000000	0.000031	0.952185	0.950252	0.926904	0.930435	0.947368	0.875676
10	Avg	3788.800000	0.000031	0.952910	0.950963	0.928344	0.924348	0.919804	0.885022
11	StdDev	971.451697	0.000000	0.004299	0.006114	0.006081	0.010135	0.018695	0.025387

RBF Kernel optimizing AUC:

	OuterFold	cVal	gammaVal	AUC	TrainAccuracy	TrainPrecision	TrainRecall	TestAccuracy	TestPrecision	TestRecall
0	1	1024.000000	0.000031	0.975732	0.941319	0.940025	0.909591	0.936957	0.910615	0.926136
1	2	1024.000000	0.000031	0.970124	0.942284	0.935745	0.916870	0.930435	0.909605	0.909605
2	3	4096.000000	0.000031	0.966459	0.953151	0.950862	0.927194	0.921739	0.942708	0.878641
3	4	1024.000000	0.000031	0.961258	0.946148	0.941176	0.919654	0.908696	0.932203	0.846154
4	5	4096.000000	0.000031	0.975477	0.959430	0.958411	0.938602	0.915217	0.877193	0.892857
5	6	4096.000000	0.000031	0.974600	0.952185	0.948525	0.928703	0.945652	0.949721	0.913978
6	7	1024.000000	0.000031	0.972470	0.946631	0.944236	0.919463	0.926087	0.906977	0.896552
7	8	1024.000000	0.000031	0.961023	0.943975	0.945501	0.910867	0.900000	0.881657	0.851429
8	9	1024.000000	0.000031	0.967146	0.944941	0.938917	0.919236	0.923913	0.938202	0.874346
9	10	1024.000000	0.000031	0.979009	0.942043	0.936764	0.916313	0.939130	0.953333	0.871951
10	Avg	1945.600000	0.000031	0.970330	0.947211	0.944016	0.920649	0.924783	0.920221	0.886165
11	StdDev	1483.916979	0.000000	0.006217	0.005889	0.007053	0.008727	0.014133	0.027199	0.026414

ROC-AUC curve:



Diabetes Dataset:

Linear Kernel optimizing Accuracy:

	OuterFold	TrainAccuracy	TrainPrecision	TrainRecall	TestAccuracy	TestPrecision	TestRecall
0	1	0.775687	0.726316	0.572614	0.792208	0.761905	0.592593
1	2	0.768452	0.730769	0.545082	0.844156	0.833333	0.625000
2	3	0.769899	0.706806	0.567227	0.753247	0.761905	0.533333
3	4	0.774240	0.730769	0.554167	0.766234	0.647059	0.785714
4	5	0.777135	0.719388	0.587500	0.779221	0.823529	0.500000
5	6	0.778582	0.729730	0.567227	0.727273	0.666667	0.600000
6	7	0.781476	0.741758	0.564854	0.831169	0.900000	0.620690
7	8	0.772793	0.716578	0.563025	0.779221	0.809524	0.566667
8	9	0.785818	0.747475	0.601626	0.675325	0.428571	0.409091
9	10	0.766234	0.719388	0.568548	0.826667	0.733333	0.550000
10	Avg	0.775031	0.726898	0.569187	0.777472	0.736583	0.578309
11	StdDev	0.006037	0.012044	0.015885	0.051252	0.132503	0.097408

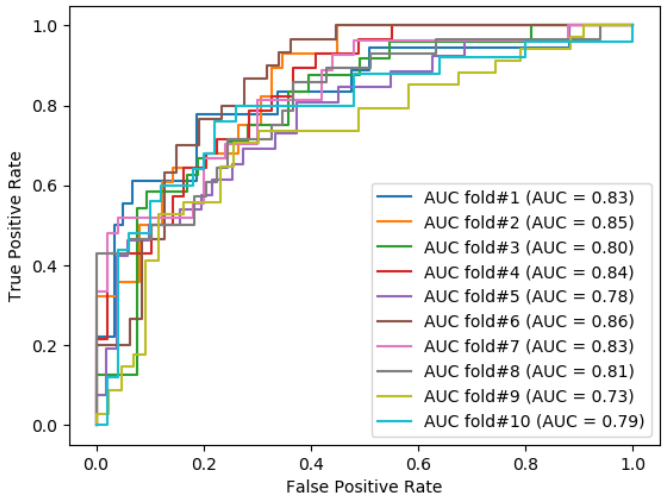
RBF Kernel optimizing Accuracy:

	OuterFold	cVal	gammaVal	TrainAccuracy	TrainPrecision	TrainRecall	TestAccuracy	TestPrecision	TestRecall
0	1	8.000000	0.000031	0.784370	0.773973	0.493450	0.740260	1.000000	0.487179
1	2	2.000000	0.000031	0.780029	0.773810	0.532787	0.753247	0.666667	0.416667
2	3	2.000000	0.000031	0.769899	0.757396	0.520325	0.792208	0.687500	0.500000
3	4	2.000000	0.000031	0.787265	0.782609	0.529412	0.623377	0.529412	0.300000
4	5	0.125000	0.000244	0.769899	0.782313	0.475207	0.740260	0.750000	0.346154
5	6	2.000000	0.000031	0.769899	0.761006	0.500000	0.766234	0.750000	0.461538
6	7	2.000000	0.000031	0.768452	0.739394	0.510460	0.792208	0.842105	0.551724
7	8	2.000000	0.000031	0.775687	0.763975	0.512500	0.792208	0.800000	0.571429
8	9	4.000000	0.000031	0.772793	0.766082	0.528226	0.792208	0.583333	0.700000
9	10	1024.000000	0.000031	0.812410	0.793814	0.631148	0.773333	0.733333	0.458333
10	Avg	104.812500	0.000052	0.779070	0.769437	0.523351	0.756554	0.734235	0.479302
11	StdDev	322.976335	0.000068	0.013431	0.015395	0.041926	0.051357	0.132733	0.114254

RBF Kernel optimizing AUC:

	OuterFold	cVal	gammaVal	AUC	TrainAccuracy	TrainPrecision	TrainRecall	TestAccuracy	TestPrecision	TestRecall
0	1	1024.000000	0.000031	0.975732	0.941319	0.940025	0.909591	0.936957	0.910615	0.926136
1	2	1024.000000	0.000031	0.970124	0.942284	0.935745	0.916870	0.930435	0.909605	0.909605
2	3	4096.000000	0.000031	0.966459	0.953151	0.950862	0.927194	0.921739	0.942708	0.878641
3	4	1024.000000	0.000031	0.961258	0.946148	0.941176	0.919654	0.908696	0.932203	0.846154
4	5	4096.000000	0.000031	0.975477	0.959430	0.958411	0.938602	0.915217	0.877193	0.892857
5	6	4096.000000	0.000031	0.974600	0.952185	0.948525	0.928703	0.945652	0.949721	0.913978
6	7	1024.000000	0.000031	0.972470	0.946631	0.944236	0.919463	0.926087	0.906977	0.896552
7	8	1024.000000	0.000031	0.961023	0.943975	0.945501	0.910867	0.900000	0.881657	0.851429
8	9	1024.000000	0.000031	0.967146	0.944941	0.938917	0.919236	0.923913	0.938202	0.874346
9	10	1024.000000	0.000031	0.979009	0.942043	0.936764	0.916313	0.939130	0.953333	0.871951
10	Avg	1945.600000	0.000031	0.970330	0.947211	0.944016	0.920649	0.924783	0.920221	0.886165
11	StdDev	1483.916979	0.000000	0.006217	0.005889	0.007053	0.008727	0.014133	0.027199	0.026414

ROC-AUC curve:



Breast Cancer Dataset:

Linear Kernel optimizing Accuracy:

	OuterFold	TrainAccuracy	TrainPrecision	TrainRecall	TestAccuracy	TestPrecision	TestRecall
0	1	0.976562	0.973822	0.963731	0.912281	0.888889	0.842105
1	2	0.966797	0.967742	0.942408	0.947368	0.909091	0.952381
2	3	0.964844	0.961957	0.941489	0.982456	1.000000	0.958333
3	4	0.968750	0.973262	0.943005	0.964912	1.000000	0.894737
4	5	0.960938	0.958115	0.938462	1.000000	1.000000	1.000000
5	6	0.964844	0.971910	0.930108	0.982456	0.962963	1.000000
6	7	0.966797	0.962162	0.946809	0.947368	1.000000	0.875000
7	8	0.968750	0.972527	0.941489	0.964912	0.958333	0.958333
8	9	0.962891	0.957672	0.942708	0.982456	1.000000	0.950000
9	10	0.974659	0.968912	0.963918	0.892857	0.875000	0.777778
10	Avg	0.967583	0.966808	0.945413	0.957707	0.959428	0.920867
11	StdDev	0.004897	0.006316	0.010622	0.033686	0.050422	0.071986

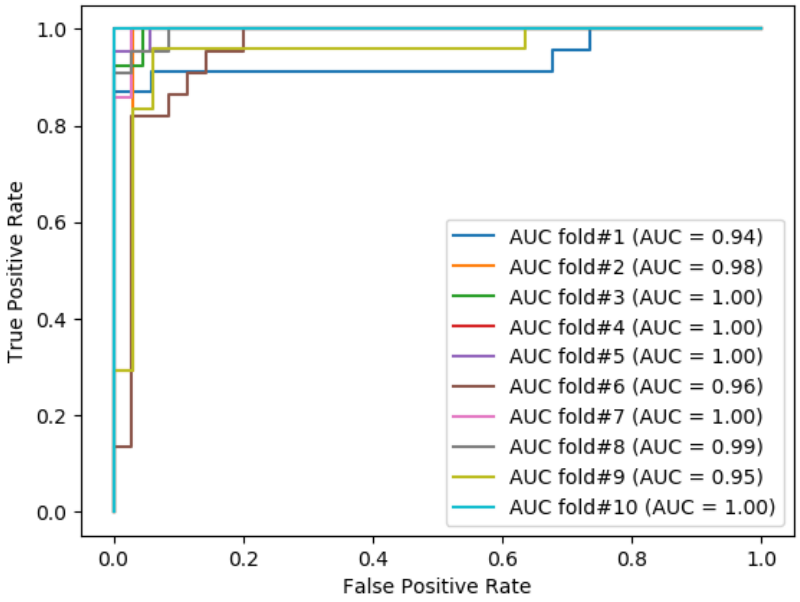
RBF Kernel optimizing Accuracy:

	OuterFold	cVal	gammaVal	TrainAccuracy	TrainPrecision	TrainRecall	TestAccuracy	TestPrecision	TestRecall
0	1	8.000000	0.000031	0.784370	0.773973	0.493450	0.740260	1.000000	0.487179
1	2	2.000000	0.000031	0.780029	0.773810	0.532787	0.753247	0.666667	0.416667
2	3	2.000000	0.000031	0.769899	0.757396	0.520325	0.792208	0.687500	0.500000
3	4	2.000000	0.000031	0.787265	0.782609	0.529412	0.623377	0.529412	0.300000
4	5	0.125000	0.000244	0.769899	0.782313	0.475207	0.740260	0.750000	0.346154
5	6	2.000000	0.000031	0.769899	0.761006	0.500000	0.766234	0.750000	0.461538
6	7	2.000000	0.000031	0.768452	0.739394	0.510460	0.792208	0.842105	0.551724
7	8	2.000000	0.000031	0.775687	0.763975	0.512500	0.792208	0.800000	0.571429
8	9	4.000000	0.000031	0.772793	0.766082	0.528226	0.792208	0.583333	0.700000
9	10	1024.000000	0.000031	0.812410	0.793814	0.631148	0.773333	0.733333	0.458333
10	Avg	104.812500	0.000052	0.779070	0.769437	0.523351	0.756554	0.734235	0.479302
11	StdDev	322.976335	0.000068	0.013431	0.015395	0.041926	0.051357	0.132733	0.114254

RBF Kernel optimizing AUC:

	OuterFold	cVal	gammaVal	AUC	TrainAccuracy	TrainPrecision	TrainRecall	TestAccuracy	TestPrecision	TestRecall
0	1	1024.000000	0.000031	0.975732	0.941319	0.940025	0.909591	0.936957	0.910615	0.926136
1	2	1024.000000	0.000031	0.970124	0.942284	0.935745	0.916870	0.930435	0.909605	0.909605
2	3	4096.000000	0.000031	0.966459	0.953151	0.950862	0.927194	0.921739	0.942708	0.878641
3	4	1024.000000	0.000031	0.961258	0.946148	0.941176	0.919654	0.908696	0.932203	0.846154
4	5	4096.000000	0.000031	0.975477	0.959430	0.958411	0.938602	0.915217	0.877193	0.892857
5	6	4096.000000	0.000031	0.974600	0.952185	0.948525	0.928703	0.945652	0.949721	0.913978
6	7	1024.000000	0.000031	0.972470	0.946631	0.944236	0.919463	0.926087	0.906977	0.896552
7	8	1024.000000	0.000031	0.961023	0.943975	0.945501	0.910867	0.900000	0.881657	0.851429
8	9	1024.000000	0.000031	0.967146	0.944941	0.938917	0.919236	0.923913	0.938202	0.874346
9	10	1024.000000	0.000031	0.979009	0.942043	0.936764	0.916313	0.939130	0.953333	0.871951
10	Avg	1945.600000	0.000031	0.970330	0.947211	0.944016	0.920649	0.924783	0.920221	0.886165
11	StdDev	1483.916979	0.000000	0.006217	0.005889	0.007053	0.008727	0.014133	0.027199	0.026414

ROC-AUC curve:



4 – SVMs vs Multiclass Problems

Wine Dataset:

Linear Kernel:

Class 1 Results:

	OuterFold	cVal	TrainAccuracy	TrainPrecision	TrainRecall	TestAccuracy	TestPrecision	TestRecall
0	1	0.125000	1.000000	1.0	1.000000	1.0	1.0	1.0
1	2	0.031250	0.993750	1.0	0.981481	1.0	1.0	1.0
2	3	0.125000	1.000000	1.0	1.000000	1.0	1.0	1.0
3	4	0.031250	0.993750	1.0	0.980769	1.0	1.0	1.0
4	5	0.125000	1.000000	1.0	1.000000	1.0	1.0	1.0
5	6	0.125000	1.000000	1.0	1.000000	1.0	1.0	1.0
6	7	0.031250	1.000000	1.0	1.000000	1.0	1.0	1.0
7	8	0.125000	1.000000	1.0	1.000000	1.0	1.0	1.0
8	9	0.031250	0.993750	1.0	0.981818	1.0	1.0	1.0
9	10	0.125000	1.000000	1.0	1.000000	1.0	1.0	1.0
10	Avg	0.087500	0.998125	1.0	0.994407	1.0	1.0	1.0
11	StdDev	0.048412	0.003019	0.0	0.009009	0.0	0.0	0.0

Class 2 Results:

	OuterFold	cVal	TrainAccuracy	TrainPrecision	TrainRecall	TestAccuracy	TestPrecision	TestRecall
0	1	0.031250	0.987500	1.000000	0.966667	0.944444	1.000000	0.909091
1	2	0.125000	0.987500	0.983607	0.983607	1.000000	1.000000	1.000000
2	3	0.125000	0.993750	1.000000	0.984615	0.944444	1.000000	0.833333
3	4	0.125000	0.987500	1.000000	0.970588	1.000000	1.000000	1.000000
4	5	0.031250	0.975000	1.000000	0.936508	1.000000	1.000000	1.000000
5	6	0.125000	0.993750	1.000000	0.985714	0.944444	0.500000	1.000000
6	7	0.125000	1.000000	1.000000	1.000000	0.944444	1.000000	0.875000
7	8	0.031250	0.981250	1.000000	0.951613	1.000000	1.000000	1.000000
8	9	0.031250	0.981250	1.000000	0.951613	1.000000	1.000000	1.000000
9	10	0.031250	0.981481	1.000000	0.953846	1.000000	1.000000	1.000000
10	Avg	0.078125	0.986898	0.998361	0.968477	0.977778	0.950000	0.961742
11	StdDev	0.049411	0.007464	0.005184	0.019980	0.028689	0.158114	0.064145

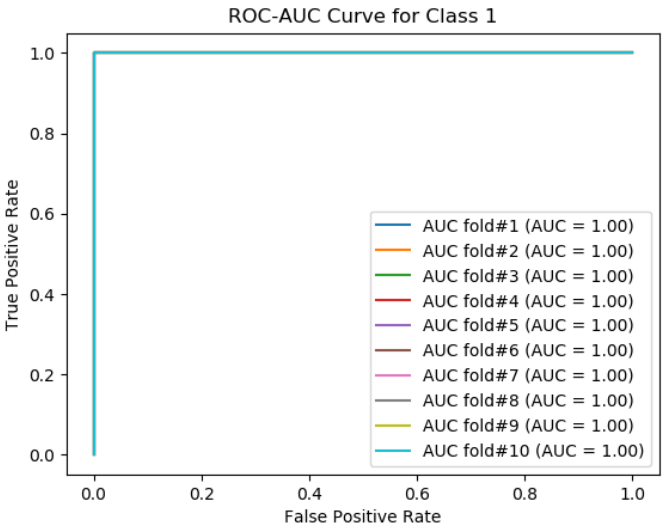
Class 3 Results:

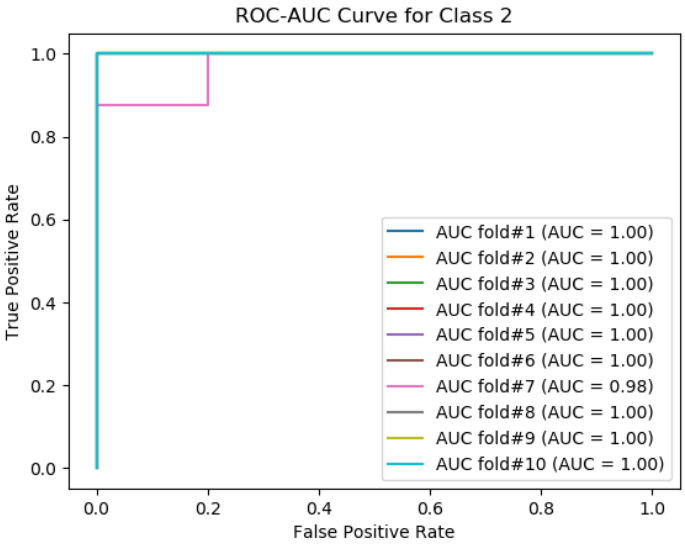
	OuterFold	cVal	TrainAccuracy	TrainPrecision	TrainRecall	TestAccuracy	TestPrecision	TestRecall
0	1	0.031250	0.993750	0.978723	1.0	0.944444	0.666667	1.000000
1	2	2.000000	1.000000	1.000000	1.0	0.944444	1.000000	0.666667
2	3	0.031250	1.000000	1.000000	1.0	0.944444	0.833333	1.000000
3	4	0.031250	0.987500	0.952381	1.0	1.000000	1.000000	1.000000
4	5	0.031250	0.993750	0.977273	1.0	1.000000	1.000000	1.000000
5	6	0.031250	0.993750	0.975000	1.0	1.000000	1.000000	1.000000
6	7	0.031250	0.993750	0.977273	1.0	1.000000	1.000000	1.000000
7	8	0.031250	0.987500	0.957447	1.0	1.000000	1.000000	1.000000
8	9	0.125000	0.993750	0.977273	1.0	1.000000	1.000000	1.000000
9	10	0.125000	1.000000	1.000000	1.0	1.000000	1.000000	1.000000
10	Avg	0.246875	0.994375	0.979537	1.0	0.983333	0.950000	0.966667
11	StdDev	0.617217	0.004612	0.016727	0.0	0.026836	0.112491	0.105409

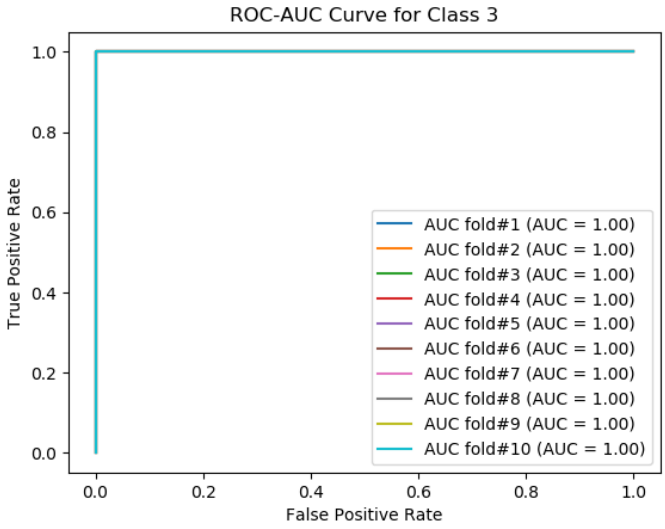
Overall Results (Predictions made using all 3 models concurrently):

	OuterFold	Combined TestAccuracy	Combined TestPrecision	Combined TestRecall
0	1	1.000000	1.000000	1.000000
1	2	0.944444	0.949495	0.944444
2	3	0.944444	0.953704	0.944444
3	4	1.000000	1.000000	1.000000
4	5	1.000000	1.000000	1.000000
5	6	1.000000	1.000000	1.000000
6	7	1.000000	1.000000	1.000000
7	8	1.000000	1.000000	1.000000
8	9	1.000000	1.000000	1.000000
9	10	1.000000	1.000000	1.000000
10	Avg	0.988889	0.990320	0.988889
11	StdDev	0.023424	0.020432	0.023424

ROC-AUC curves for Linear Kernel:







RBF Kernel (for Wine Dataset):

Class 1 Results:

	OuterFold	cVal	gammaVal	TrainAccuracy	TrainPrecision	TrainRecall	TestAccuracy	TestPrecision	TestRecall
0	1	1024.000000	0.000031	1.0	1.0	1.0	1.000000	1.000000	1.000000
1	2	1024.000000	0.000031	1.0	1.0	1.0	0.944444	0.888889	1.000000
2	3	256.000000	0.000244	1.0	1.0	1.0	1.000000	1.000000	1.000000
3	4	2.000000	0.125000	1.0	1.0	1.0	1.000000	1.000000	1.000000
4	5	32.000000	0.001953	1.0	1.0	1.0	1.000000	1.000000	1.000000
5	6	2.000000	0.125000	1.0	1.0	1.0	0.944444	1.000000	0.800000
6	7	256.000000	0.000244	1.0	1.0	1.0	1.000000	1.000000	1.000000
7	8	256.000000	0.000244	1.0	1.0	1.0	1.000000	1.000000	1.000000
8	9	2.000000	0.125000	1.0	1.0	1.0	1.000000	1.000000	1.000000
9	10	2.000000	0.125000	1.0	1.0	1.0	1.000000	1.000000	1.000000
10	Avg	285.600000	0.050275	1.0	1.0	1.0	0.988889	0.988889	0.980000
11	StdDev	405.397802	0.064316	0.0	0.0	0.0	0.023424	0.035136	0.063246

Class 2 Results:

	OuterFold	cVal	gammaVal	TrainAccuracy	TrainPrecision	TrainRecall	TestAccuracy	TestPrecision	TestRecall
0	1	256.000000	0.000244	0.987500	1.0	0.969231	1.000000	1.0	1.000000
1	2	1024.000000	0.000244	1.000000	1.0	1.000000	0.944444	1.0	0.833333
2	3	256.000000	0.000031	0.987500	1.0	0.968254	0.888889	1.0	0.750000
3	4	256.000000	0.000031	0.981250	1.0	0.953125	1.000000	1.0	1.000000
4	5	256.000000	0.000031	0.981250	1.0	0.955224	0.944444	1.0	0.750000
5	6	256.000000	0.000244	0.993750	1.0	0.983607	1.000000	1.0	1.000000
6	7	8.000000	0.000244	0.987500	1.0	0.968750	1.000000	1.0	1.000000
7	8	1024.000000	0.000031	0.987500	1.0	0.969697	1.000000	1.0	1.000000
8	9	1024.000000	0.000031	0.987500	1.0	0.966667	1.000000	1.0	1.000000
9	10	8.000000	0.000244	0.969136	1.0	0.921875	0.875000	1.0	0.714286
10	Avg	436.800000	0.000137	0.986289	1.0	0.965643	0.965278	1.0	0.904762
11	StdDev	417.078943	0.000113	0.008138	0.0	0.020360	0.049539	0.0	0.126363

Class 3 Results:

	OuterFold	cVal	gammaVal	TrainAccuracy	TrainPrecision	TrainRecall	TestAccuracy	TestPrecision	TestRecall
0	1	256.00000	0.000031	0.993750	0.976190	1.000000	1.000000	1.000000	1.0
1	2	256.00000	0.000031	0.987500	0.956522	1.000000	1.000000	1.000000	1.0
2	3	8.00000	0.000244	0.993750	1.000000	0.977778	0.944444	0.750000	1.0
3	4	8.00000	0.000244	0.987500	0.977273	0.977273	1.000000	1.000000	1.0
4	5	8.00000	0.000244	0.993750	0.976190	1.000000	1.000000	1.000000	1.0
5	6	8.00000	0.000244	0.987500	0.977778	0.977778	1.000000	1.000000	1.0
6	7	256.00000	0.000031	0.993750	0.976744	1.000000	1.000000	1.000000	1.0
7	8	1024.00000	0.000031	0.993750	0.976190	1.000000	1.000000	1.000000	1.0
8	9	256.00000	0.000031	0.993750	0.978261	1.000000	1.000000	1.000000	1.0
9	10	8.00000	0.000244	0.993827	0.977778	1.000000	1.000000	1.000000	1.0
10	Avg	208.80000	0.000137	0.991883	0.977293	0.993283	0.994444	0.975000	1.0
11	StdDev	311.81647	0.000113	0.003024	0.010287	0.010817	0.017568	0.079057	0.0

Overall Results (Predictions made using all 3 models concurrently):

	OuterFold	Combined TestAccuracy	Combined TestPrecision	Combined TestRecall
0	1	1.000000	1.000000	1.000000
1	2	0.944444	0.955556	0.944444
2	3	0.944444	0.958333	0.944444
3	4	1.000000	1.000000	1.000000
4	5	1.000000	1.000000	1.000000
5	6	1.000000	1.000000	1.000000
6	7	1.000000	1.000000	1.000000
7	8	1.000000	1.000000	1.000000
8	9	1.000000	1.000000	1.000000
9	10	1.000000	1.000000	1.000000
10	Avg	0.988889	0.991389	0.988889
11	StdDev	0.023424	0.018166	0.023424

ROC-AUC curves for RBF Kernel: