

Need to explain (in this chapter or earlier) where the data set originally came from, what it represents, etc. Also, in earlier chapters describe why this protein family is important.

Chapter 3

Experimental Setup

Use hyphens in 'coarse-level', 'fine-grained', etc.

3.1 Training and Testing Coarse Grain and Fine Grain Classifiers

The bioinformatics dataset is composed of 9 classes as shown in *Figure 1.1* ~~on page 2~~. The coarse level concept is whether or not the protein resides within the mitochondria. The negative case of not residing within the mitochondria is class 0. ~~Is this zero?~~ The positive case of residing within the mitochondria corresponds to any of the 8 target compartment classes, numbered 1 through 8. Since the negative case has no fine grained labels, the fine grained classifier is composed of separate classifiers for each of the fine grained labels. The 8 fine grained classifiers are trained such that only the instances of the class corresponding to that classifier's target compartment are marked as positive, all the others are treated as negative. The coarse level classifier treats all fine grained target compartment instances as members of a single positive class. For all classifiers the non mitochondrial instances are treated as negative or 0 labeled. The totals for each class type is shown in *Table 3.1*. Throughout this experiment a 10 folds cross validation strategy is used, an example partitioning is shown in *Table 3.1b*.

Classes	Count
0	19136
1	13
2	185
3	324
4	190
5	11
6	104
7	59
8	76
Tot All	20098
Tot Coarse	19136
Tot Fine	962
Features	449

(a) Classes

Folds	All	0	1	2	3	4	5	6	7	8
1	2010	1914	1	19	32	19	1	11	6	7
2	2010	1914	1	19	32	19	1	11	6	7
3	2010	1914	1	19	32	19	1	11	5	8
4	2010	1914	1	19	32	19	1	10	6	8
5	2010	1914	1	18	33	19	1	10	6	8
6	2010	1914	1	18	33	19	1	10	6	8
7	2010	1913	2	18	33	19	1	10	6	8
8	2010	1913	2	18	33	19	1	10	6	8
9	2009	1913	2	18	32	19	2	10	6	7
10	2009	1913	1	19	32	19	1	11	6	7
Total	20098	19136	13	185	324	190	11	104	59	76

(b) Folds

Table 3.1: This dataset contains 20098 instances total with 449 features each. An example partitioning is shown, some classes like 1 and 5 contain only 1-2 instances in a given test set. Note there is a heavy class imbalance with approx. 20 negative instances for each positive instance.

Each partition contains a representative portion of each class, the instances are randomly distributed between partitions. The train set is composed of joining 9 of the partitions together holding 1 fold out for the test set. An example of the totals for a Train and Test set is shown on Table 3.2.

Train	All	0	1	2	3	4	5	6	7	8
Total	18088	17222	12	166	292	171	10	93	53	69
Test	All	0	1	2	3	4	5	6	7	8
Total	2010	1914	1	19	32	19	1	11	6	7

Table 3.2: Example of totals for the Train and Test corresponding to when the first fold is held out to be the test set.

Because the experiment will involve running multiple rounds iteratively increasing the number of instances on which the classifiers are trained and tested, a subset was used to tune the parameters of the classifiers. This allowed variations of the classifier parameters to be ~~run~~ ^{run} rapidly and for the class weight parameter to be tuned for various round sizes. The reduced subset contains a randomly chosen group of approximately $1/5^{th}$ of the negatives. The class totals and example partitioning for the reduced subset is shown in Table 3.3.

So, after tuning, did you hold the parameter values fixed and re-run the experiments on a new partitioning?

Classes	Count
0	3827
1	13
2	185
3	324
4	190
5	11
6	104
7	59
8	76
Tot All	4789
Tot Coarse	3827
Tot Fine	962
Features	449

(a) Classes Subset

Folds	All	0	1	2	3	4	5	6	7	8
1	479	383	1	19	32	19	1	11	6	7
2	479	383	1	19	32	19	1	11	6	7
3	479	383	1	19	32	19	1	11	6	7
4	479	383	1	19	32	19	1	11	5	8
5	479	383	1	19	32	19	1	10	6	8
6	479	383	1	18	33	19	1	10	6	8
7	479	383	1	18	33	19	1	10	6	8
8	479	382	2	18	33	19	1	10	6	8
9	479	382	2	18	33	19	1	10	6	8
10	478	382	2	18	32	19	2	10	6	7
Total	4789	3827	13	185	324	190	11	104	59	76

(b) Folds Subset

Table 3.3: The subset of instances used for tuning classifier paramters contains approximately 1/5th and retains all positive instances.

Train	All	0	1	2	3	4	5	6	7	8
Total	4310	3444	12	166	292	171	10	93	53	69
Test	All	0	1	2	3	4	5	6	7	8
Total	479	383	1	19	32	19	1	11	6	7

Table 3.4: Example totals for the train and test set for the subset of data. The subset of data is used for the majority of the parameter search.

Throughout this project the python ~~library~~ sci-kit learn library is used for the implementation of the classification, preprocessing, and evaluation algorithms [2]. The Support Vector Machine (SVM) supervised learning algorithm is used on the un-scaled subset of the data to obtain the base results shown in 3.5. The coarse and the fine algorithm performance is shown for each of the 10 folds along with the average performance across the 10 folds. Also the Reciever Operator Characteristic and Precision-Recall curves are calculated with fine instances weighted according to the number of of instances in the test set divided by the number of positive instances in the test set which is a value of 4.99 for the data subset.

If this chapter is experimental setup, why are you presenting results? Or, is this just to tune parameters?

10

coarse-pr	fine-pr	coarse-roc	fine-roc	coarse-acc	fine-acc	coarse-f1	fine-f1
0.807	0.796	0.779	0.768	0.816	0.802	0.214	0.021
0.848	0.822	0.828	0.790	0.825	0.804	0.263	0.041
0.846	0.821	0.810	0.765	0.818	0.802	0.243	0.021
0.860	0.832	0.826	0.775	0.831	0.802	0.319	0.021
0.859	0.829	0.828	0.783	0.833	0.804	0.298	0.041
0.796	0.763	0.748	0.715	0.816	0.806	0.214	0.061
0.838	0.825	0.797	0.792	0.818	0.800	0.243	0.020
0.836	0.816	0.803	0.770	0.823	0.800	0.309	0.020
0.863	0.845	0.833	0.805	0.829	0.797	0.305	0.000
0.844	0.806	0.806	0.758	0.836	0.807	0.339	0.061
avg 0.840	avg 0.815	avg 0.806	avg 0.772	avg 0.825	avg 0.802	avg 0.275	avg 0.031

Table 3.5: SVM default results without parameter selection or preprocessing. Where Precision Recall area under the curve is (pr), Receiver Operator Characteristic area under the curve is (roc), Accuracy is (acc), F1-measure is (f1).

coarse-tn	fine-tn	coarse-fp	fine-fp	coarse-fn	fine-fn	coarse-tp	fine-tp
379	383	4	0	84	95	12	1
380	383	3	0	81	94	15	2
378	383	5	0	82	95	14	1
379	383	4	0	77	95	19	1
382	383	1	0	79	94	17	2
379	383	4	0	84	93	12	3
378	382	5	1	82	95	14	1
375	382	7	0	78	96	19	1
379	382	3	0	79	97	18	0
379	382	3	0	75	92	20	3
avg 378.8	avg 382.6	avg 3.9	avg 0.1	avg 80.1	avg 94.6	avg 16.0	avg 1.5

Table 3.6: SVM default results confusion matrix. Where True Negatives is (tn), False Positives is (fp), False Negatives (fn), True Positives is (tp).

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.840	0.806	0.825	0.275	(378.8 / 80.1)	(3.9 / 16.0)
fine	0.815	0.772	0.802	0.031	(382.6 / 94.6)	(0.1 / 1.5)

Table 3.7: SVM default condensed view of summary performance metrics, each value is the average of 10 folds.

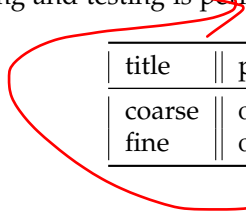
Define PR, ROC, and AUC (perhaps in an earlier chapter).

The primary metric used to make decisions between alternative parameter choices is the PR-AUC and ROC-AUC. The f-measure and accuracy metrics can be shown to be correlated to a chosen point on the ROC or PR curves. As shown in Figure 4.1 on page 29, each point on the ROC curve has an associated chosen accuracy point, both the coarse and fine classifiers have similar sets of accuracy and f-measure

points. The chosen threshold used to output the accuracy, f-measure and confusion matrices varies between the coarse and fine classifier, so at a first glance it appears as if fine out performs coarse in these metrics but an alternative threshold could be selected for the coarse classifier to obtain metrics matching the fine output. Alternatively, the PR-AUC and ROC-AUC compare the correctness of the entire ranking of the instances in the test set by the classifier, and thus eliminate the need to consider the dynamic tuning of the threshold used by the classifier to output a given confusion matrix, accuracy, and f-measure score.

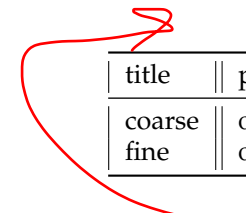
3.1.1 Varying SVM Scaling Methods

Different scaling methods were used to preprocess the data [2]. The standard scaling (std-scaler) strategy centers all features around zero and have variance in the same order, it outputs the features with a mean of zero and a unit variance. The minimum maximum scaling (minmax-scaler) strategy scales features between a minimum and maximum value, which is 0 and 1. The normalization scaling (norm-scaler) strategy scales individual samples to have a unit norm. Each preprocessing strategy is applied on the entire dataset before training and testing is performed. Preprocessing was performed with a radial basis function kernel.



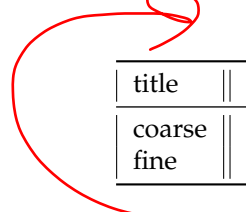
title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.881	0.855	0.799	0.000	(382.7 / 96.1)	(0.0 / 0.0)
fine	0.840	0.810	0.799	0.000	(382.7 / 96.1)	(0.0 / 0.0)

Table 3.8: SVM minmax-scaler results.



title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.801	0.791	0.799	0.000	(382.7 / 96.1)	(0.0 / 0.0)
fine	0.636	0.615	0.799	0.000	(382.7 / 96.1)	(0.0 / 0.0)

Table 3.9: SVM norm-scaler results.



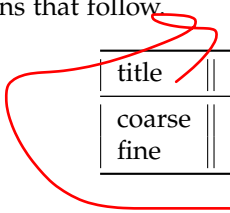
title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.912	0.882	0.881	0.631	(372.7 / 47.1)	(10.0 / 49.0)
fine	0.879	0.848	0.809	0.094	(382.7 / 91.3)	(0.0 / 4.8)

Table 3.10: SVM std-scaler results. This option is chosen.

3.1.2 Varying SVM Kernels

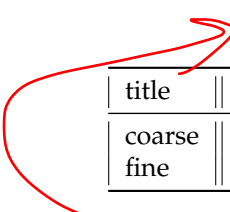
Different kernel functions were used in the SVM classifier including: Radial Basis Function (RBF), Polynomial Degree 3 and 6 (Poly), Linear, and Sigmoid [2]. The chosen preprocessing strategy of std-scaler is

used for these results. As parameter selection is elicited the choices from previous sections are used in any sections that follow



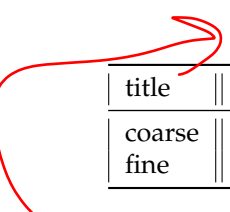
title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.867	0.841	0.853	0.599	(355.5 / 43.4)	(27.2 / 52.7)
fine	0.816	0.789	0.828	0.523	(351.1 / 50.8)	(31.6 / 45.3)

Table 3.11: Linear kernel results.



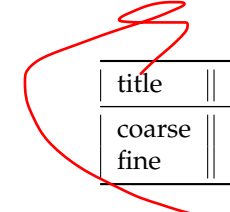
title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.816	0.817	0.807	0.169	(376.9 / 86.7)	(5.8 / 9.4)
fine	0.755	0.743	0.801	0.063	(380.3 / 92.9)	(2.3 / 3.2)

Table 3.12: Poly degree 3 kernel results.



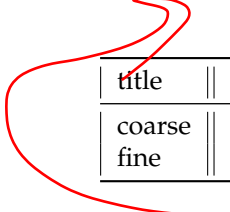
title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.659	0.637	0.797	0.037	(379.5 / 94.2)	(3.2 / 1.9)
fine	0.624	0.584	0.794	0.020	(379.0 / 95.1)	(3.7 / 1.0)

Table 3.13: Poly degree 6 kernel results.



title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.703	0.693	0.773	0.405	(333.0 / 59.0)	(49.7 / 37.1)
fine	0.653	0.622	0.789	0.127	(370.3 / 88.7)	(12.4 / 7.4)

Table 3.14: Sigmoid kernel results.



title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.912	0.882	0.881	0.630	(372.6 / 47.1)	(10.1 / 49.0)
fine	0.879	0.848	0.809	0.094	(382.7 / 91.3)	(0.0 / 4.8)

Table 3.15: RBF kernel results. This option is chosen.

3.1.3 Varying SVM Feature Selection

I tried different feature selection percentages. The Select Percentile library was used from sci-kit learn [2]. This is a univariate feature selection strategy that ranks the features usability for classification according to a statistical measure, then keeps a certain percentage of the features. The 100% example is the option chosen in the previous section.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.907	0.875	0.877	0.623	(370.7 / 47.1)	(12.0 / 49.0)
fine	0.854	0.823	0.806	0.068	(382.7 / 92.7)	(0.0 / 3.4)

Table 3.16: SVM select percentile, keep 25% of features.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.913	0.885	0.879	0.632	(371.3 / 46.4)	(11.4 / 49.7)
fine	0.874	0.842	0.810	0.097	(382.7 / 91.2)	(0.0 / 4.9)

Table 3.17: SVM select percentile, keep 50% of features.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.913	0.883	0.878	0.622	(372.1 / 47.9)	(10.6 / 48.2)
fine	0.880	0.848	0.809	0.089	(382.7 / 91.6)	(0.0 / 4.5)

Table 3.18: SVM select percentile, keep 75% of features. This option is chosen.

Note that leveraging the fine grained labels did not improve classifier performance relative to the coarse classifier. An alternative classifier strategy Logistic Regression (LogReg) is investigated.

3.1.4 Varying Logistic Regression Scaling

Tested out the same options for preprocessing scaling, that were tried for SVM.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.887	0.862	0.867	0.615	(364.1 / 45.0)	(18.6 / 51.1)
fine	0.854	0.837	0.833	0.395	(372.8 / 69.9)	(9.9 / 26.2)

Table 3.19: Logistic Regression - No scaling.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.864	0.849	0.846	0.583	(353.8 / 44.8)	(28.7 / 51.3)
fine	0.833	0.816	0.831	0.471	(362.0 / 60.2)	(20.5 / 36.0)

Table 3.20: Logistic Regression standard scaling.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.790	0.761	0.799	0.000	(382.7 / 96.1)	(0.0 / 0.0)
fine	0.767	0.735	0.799	0.000	(382.7 / 96.1)	(0.0 / 0.0)

Table 3.21: Logistic Regression normalization scaling.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.891	0.867	0.864	0.581	(368.6 / 50.9)	(14.1 / 45.2)
fine	0.888	0.862	0.812	0.130	(382.1 / 89.3)	(0.6 / 6.8)

Table 3.22: Logistic Regression MinMax scaling. This option is chosen.

3.1.5 Varying Logistic Regression Feature Selection

Tested out the same options for feature selection that were tried for SVM.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.872	0.848	0.849	0.497	(370.8 / 60.3)	(11.9 / 35.8)
fine	0.869	0.845	0.804	0.052	(382.2 / 93.5)	(0.5 / 2.6)

Table 3.23: Logistic Regression select percentile 25%.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.875	0.849	0.849	0.497	(370.8 / 60.3)	(11.9 / 35.8)
fine	0.872	0.846	0.803	0.050	(382.2 / 93.6)	(0.5 / 2.5)

Table 3.24: Logistic Regression select percentile 50%.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.871	0.847	0.848	0.493	(370.6 / 60.6)	(12.1 / 35.5)
fine	0.869	0.845	0.803	0.048	(382.0 / 93.7)	(0.7 / 2.4)

Table 3.25: Logistic Regression select percentile 75%.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.891	0.867	0.864	0.581	(368.6 / 50.9)	(14.1 / 45.2)
fine	0.888	0.862	0.812	0.130	(382.1 / 89.3)	(0.6 / 6.8)

Table 3.26: Logistic Regression select percentile 100%. This option is chosen.

3.1.6 Varying Logistic Regression Postive Class Weight and Cost

Since there is a class imbalance in the dataset, see *Table 3.1a* ~~on page 8~~, class weight and cost parameter pairs are varied. The cost default value is 1.0, and the class weight default value is 1.0. The original value for weighting the fine training instance is the number of instances in the train set divided by the number of postive instances, this is 4.977. The negative instance train weight is always 1.0.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.886	0.868	0.787	0.606	(298.7 / 17.9)	(84.0 / 78.2)
fine	0.885	0.862	0.857	0.587	(361.7 / 47.3)	(21.0 / 48.8)

Table 3.27: LogReg weight 4.977, cost 1.0

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.880	0.861	0.755	0.579	(280.7 / 15.4)	(102.0 / 80.7)
fine	0.880	0.856	0.851	0.483	(374.2 / 62.7)	(8.5 / 33.4)

Table 3.28: LogReg weight 4.977, cost 0.1

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.876	0.855	0.793	0.603	(304.6 / 21.1)	(78.1 / 75.0)
fine	0.866	0.842	0.835	0.583	(344.8 / 40.9)	(37.9 / 55.2)

Table 3.29: LogReg weight 4.977, cost 10.0

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.883	0.865	0.690	0.536	(245.4 / 10.9)	(137.3 / 85.2)
fine	0.880	0.859	0.822	0.620	(324.2 / 26.7)	(58.5 / 69.4)

Table 3.30: LogReg weight 10.0, cost 1.0

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.879	0.863	0.609	0.486	(203.6 / 7.9)	(179.1 / 88.2)
fine	0.881	0.859	0.834	0.621	(334.5 / 31.1)	(48.2 / 65.0)

Table 3.31: LogReg weight 10.0, cost 0.1

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.871	0.851	0.723	0.554	(264.1 / 13.9)	(118.6 / 82.2)
fine	0.861	0.837	0.792	0.585	(309.3 / 26.2)	(73.4 / 69.9)

Table 3.32: LogReg weight 10.0, cost 10.0

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.884	0.867	0.734	0.566	(268.8 / 13.5)	(113.9 / 82.6)
fine	0.882	0.861	0.846	0.624	(343.4 / 34.6)	(39.3 / 61.5)

Table 3.33: LogReg weight 7.5, cost 1.0

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.880	0.862	0.668	0.517	(234.8 / 11.1)	(147.9 / 85.0)
fine	0.881	0.858	0.859	0.613	(357.3 / 42.3)	(25.4 / 53.8)

Table 3.34: LogReg weight 7.5, cost 0.1. This option is chosen due to showing advantage for the fine classifier compared to the coarse classifier.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.873	0.852	0.757	0.578	(283.2 / 16.7)	(99.5 / 79.4)
fine	0.863	0.839	0.810	0.588	(323.3 / 31.4)	(59.4 / 64.7)

Table 3.35: LogReg weight 7.5, cost 10.0

3.1.7 Varying Logistic Regression Tune Fine Class Weights

The weight for each of the separate fine classes is tuned by multiplying, the class weight of 7.5, determined in the previous section by a fixed ratio. A weight ratio of 1.0 would output a fine class weight of 7.5. A weight ratio of 0.5 would output a fine class weight of 3.75. Subsections showing the tuning results for each of the 8 fine grained classes follow. The confusion matrices and output metrics for the individual fine class are shown in order to demonstrate how well the classifier is learning that fine grained class. These metrics are the average of 10 folds. The coarse classifier output is not shown as it will not vary or be dependent upon the fine class weight tuning.

3.1.7.1 Fine Tune Class 1 Weights

Add text in each section, including a summary of results.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.881	0.858	0.859	0.613	(357.3 / 42.3)	(25.4 / 53.8)
trainCls-1	0.995	0.999	0.998	0.477	(4297.7 / 7.7)	(0.8 / 4.0)
testCls-1	0.722	0.996	0.997	0.100	(477.4 / 1.2)	(0.1 / 0.1)

Table 3.36: LogReg Class 1 weight ratio 1.0

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.880	0.856	0.859	0.613	(357.4 / 42.3)	(25.3 / 53.8)
trainCls-1	0.994	0.998	0.997	0.142	(4298.5 / 10.8)	(0.0 / 0.9)
testCls-1	0.696	0.995	0.997	0.000	(477.5 / 1.3)	(0.0 / 0.0)

Table 3.37: LogReg Class 1 weight ratio 0.5

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.882	0.860	0.859	0.617	(357.1 / 41.7)	(25.6 / 54.4)
trainCls-1	0.995	1.000	0.999	0.854	(4295.8 / 1.0)	(2.7 / 10.7)
testCls-1	0.722	0.997	0.998	0.400	(477.1 / 0.7)	(0.4 / 0.6)

Table 3.38: LogReg Class 1 weight ratio 3.0. This option is chosen.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.881	0.859	0.860	0.618	(357.0 / 41.5)	(25.7 / 54.6)
trainCls-1	0.995	1.000	0.999	0.850	(4294.3 / 0.0)	(4.2 / 11.7)
testCls-1	0.722	0.997	0.998	0.513	(476.9 / 0.5)	(0.6 / 0.8)

Table 3.39: LogReg Class 1 weight ratio 5.0

3.1.7.2 Fine Tune Class 2 Weights

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.882	0.860	0.859	0.617	(357.1 / 41.7)	(25.6 / 54.4)
trainCls-2	0.800	0.804	0.952	0.200	(4076.9 / 140.5)	(66.8 / 26.0)
testCls-2	0.655	0.689	0.944	0.081	(450.7 / 17.3)	(9.6 / 1.2)

Table 3.40: LogReg Class 2 weight ratio 1.0. This option is chosen.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.882	0.857	0.862	0.618	(359.1 / 42.5)	(23.6 / 53.6)
trainCls-2	0.785	0.787	0.961	0.052	(4139.4 / 161.9)	(4.3 / 4.6)
testCls-2	0.656	0.694	0.960	0.009	(459.4 / 18.4)	(0.9 / 0.1)

Table 3.41: LogReg Class 2 weight ratio 0.5

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.877	0.857	0.855	0.620	(352.5 / 39.3)	(30.2 / 56.8)
trainCls-2	0.806	0.814	0.924	0.263	(3924.1 / 108.1)	(219.6 / 58.4)
testCls-2	0.652	0.684	0.914	0.123	(434.8 / 15.6)	(25.5 / 2.9)

Table 3.42: LogReg Class 2 weight ratio 1.5

3.1.7.3 Fine Tune Class 3 Weights

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.882	0.860	0.859	0.617	(357.1 / 41.7)	(25.6 / 54.4)
trainCls-3	0.846	0.852	0.882	0.401	(3628.6 / 120.7)	(390.0 / 170.9)
testCls-3	0.795	0.803	0.873	0.360	(401.2 / 15.4)	(45.2 / 17.0)

Table 3.43: LogReg Class 3 weight ratio 1.0. This option is chosen.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.870	0.852	0.839	0.445	(370.9 / 65.1)	(11.8 / 31.0)
trainCls-3	0.838	0.838	0.929	0.288	(3942.0 / 229.7)	(76.6 / 61.9)
testCls-3	0.792	0.798	0.925	0.246	(437.2 / 26.5)	(9.2 / 5.9)

Table 3.44: LogReg Class 3 weight ratio 0.5

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.879	0.855	0.832	0.626	(331.2 / 28.9)	(51.5 / 67.2)
trainCls-3	0.849	0.859	0.813	0.351	(3288.4 / 74.5)	(730.2 / 217.1)
testCls-3	0.795	0.805	0.804	0.318	(363.3 / 10.6)	(83.1 / 21.8)

Table 3.45: LogReg Class 3 weight ratio 1.5

3.1.7.4 Fine Tune Class 4 Weights

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.882	0.860	0.859	0.617	(357.1 / 41.7)	(25.6 / 54.4)
trainCls-4	0.937	0.942	0.960	0.531	(4038.6 / 72.9)	(100.6 / 98.1)
testCls-4	0.882	0.902	0.952	0.433	(447.1 / 10.2)	(12.7 / 8.8)

Table 3.46: LogReg Class 4 weight ratio 1.0

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.875	0.852	0.855	0.590	(359.2 / 45.9)	(23.5 / 50.2)
trainCls-4	0.928	0.932	0.965	0.397	(4108.1 / 120.9)	(31.1 / 50.1)
testCls-4	0.878	0.898	0.962	0.320	(456.0 / 14.6)	(3.8 / 4.4)

Table 3.47: LogReg Class 4 weight ratio 0.5

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.883	0.861	0.856	0.624	(352.4 / 38.8)	(30.3 / 57.3)
trainCls-4	0.941	0.947	0.936	0.462	(3918.1 / 53.2)	(221.1 / 117.8)
testCls-4	0.886	0.903	0.926	0.382	(432.5 / 8.0)	(27.3 / 11.0)

Table 3.48: LogReg Class 4 weight ratio 1.5. This option is chosen.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.880	0.859	0.853	0.627	(348.9 / 36.7)	(33.8 / 59.4)
trainCls-4	0.943	0.950	0.917	0.429	(3817.7 / 36.5)	(321.5 / 134.5)
testCls-4	0.886	0.903	0.906	0.352	(421.8 / 6.8)	(38.0 / 12.2)

Table 3.49: LogReg Class 4 weight ratio 2.0

3.1.7.5 Fine Tune Class 5 Weights

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.883	0.861	0.856	0.624	(352.4 / 38.8)	(30.3 / 57.3)
trainCls-5	0.940	0.941	0.998	0.000	(4300.2 / 10.0)	(0.0 / 0.0)
testCls-5	0.393	0.681	0.998	0.000	(477.8 / 1.0)	(0.0 / 0.0)

Table 3.50: LogReg Class 5 weight ratio 1.0

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.883	0.861	0.856	0.624	(352.4 / 38.8)	(30.3 / 57.3)
trainCls-5	0.911	0.912	0.998	0.000	(4300.2 / 10.0)	(0.0 / 0.0)
testCls-5	0.389	0.672	0.998	0.000	(477.8 / 1.0)	(0.0 / 0.0)

Table 3.51: LogReg Class 5 weight ratio 0.5

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.883	0.861	0.856	0.624	(352.4 / 38.8)	(30.3 / 57.3)
trainCls-5	0.957	0.958	0.998	0.000	(4300.2 / 10.0)	(0.0 / 0.0)
testCls-5	0.396	0.687	0.998	0.000	(477.8 / 1.0)	(0.0 / 0.0)

Table 3.52: LogReg Class 5 weight ratio 1.5

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.883	0.861	0.856	0.624	(352.4 / 38.8)	(30.3 / 57.3)
trainCls-5	0.990	0.990	0.998	0.374	(4299.8 / 7.6)	(0.4 / 2.4)
testCls-5	0.401	0.694	0.998	0.000	(477.7 / 1.0)	(0.1 / 0.0)

Table 3.53: LogReg Class 5 weight ratio 5.0

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.883	0.861	0.855	0.623	(352.1 / 38.8)	(30.6 / 57.3)
trainCls-5	0.996	0.997	0.998	0.609	(4293.4 / 2.7)	(6.8 / 7.3)
testCls-5	0.402	0.696	0.996	0.000	(476.8 / 1.0)	(1.0 / 0.0)

Table 3.54: LogReg Class 5 weight ratio 10.0. This option is chosen.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.881	0.860	0.854	0.622	(351.6 / 38.7)	(31.1 / 57.4)
trainCls-5	0.998	0.998	0.992	0.355	(4265.8 / 0.5)	(34.4 / 9.5)
testCls-5	0.381	0.616	0.989	0.000	(473.5 / 1.0)	(4.3 / 0.0)

Table 3.55: LogRegCls5-Wt20

3.1.7.6 Fine Tune Class 6 Weights

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.883	0.861	0.855	0.623	(352.1 / 38.8)	(30.6 / 57.3)
trainCls-6	0.945	0.962	0.976	0.303	(4182.5 / 70.8)	(34.1 / 22.8)
testCls-6	0.892	0.936	0.972	0.191	(463.9 / 8.8)	(4.5 / 1.6)

Table 3.56: LogReg Class 6 weight ratio 1.0

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.882	0.860	0.855	0.622	(352.1 / 38.9)	(30.6 / 57.2)
trainCls-6	0.938	0.956	0.978	0.006	(4216.5 / 93.3)	(0.1 / 0.3)
testCls-6	0.881	0.928	0.978	0.000	(468.3 / 10.4)	(0.1 / 0.0)

Table 3.57: LogReg Class 6 weight ratio 0.5

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.884	0.861	0.855	0.627	(350.8 / 37.6)	(31.9 / 58.5)
trainCls-6	0.950	0.967	0.949	0.380	(4023.8 / 26.4)	(192.8 / 67.2)
testCls-6	0.897	0.939	0.945	0.292	(447.0 / 5.0)	(21.4 / 5.4)

Table 3.58: LogReg Class 6 weight ratio 2.0. This option is chosen.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.884	0.860	0.850	0.629	(346.6 / 35.5)	(36.1 / 60.6)
trainCls-6	0.952	0.969	0.921	0.335	(3885.8 / 8.3)	(330.8 / 85.3)
testCls-6	0.898	0.940	0.915	0.281	(430.5 / 2.6)	(37.9 / 7.8)

Table 3.59: LogReg Class 6 weight ratio 3.0

3.1.7.7 Fine Tune Class 7 Weights

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.884	0.862	0.855	0.628	(350.8 / 37.5)	(31.9 / 58.6)
trainCls-7	0.892	0.893	0.988	0.000	(4257.1 / 53.1)	(0.0 / 0.0)
testCls-7	0.648	0.720	0.988	0.000	(472.9 / 5.9)	(0.0 / 0.0)

Table 3.60: LogReg Class 7 weight ratio 1.0

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.884	0.861	0.855	0.627	(350.8 / 37.6)	(31.9 / 58.5)
trainCls-7	0.859	0.857	0.988	0.000	(4257.1 / 53.1)	(0.0 / 0.0)
testCls-7	0.636	0.708	0.988	0.000	(472.9 / 5.9)	(0.0 / 0.0)

Table 3.61: LogReg Class 7 weight ratio 0.5

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.885	0.863	0.855	0.632	(350.1 / 36.7)	(32.6 / 59.4)
trainCls-7	0.930	0.939	0.986	0.344	(4234.1 / 37.3)	(23.0 / 15.8)
testCls-7	0.667	0.739	0.983	0.105	(470.1 / 5.4)	(2.8 / 0.5)

Table 3.62: LogReg Class 7 weight ratio 3.0. This option is chosen.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.883	0.860	0.847	0.628	(344.0 / 34.4)	(38.7 / 61.7)
trainCls-7	0.941	0.953	0.956	0.265	(4086.0 / 18.9)	(171.1 / 34.2)
testCls-7	0.674	0.744	0.948	0.099	(452.3 / 4.5)	(20.6 / 1.4)

Table 3.63: LogReg Class 7 weight ratio 5.0

3.1.7.8 Fine Tune Class 8 Weights

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.886	0.864	0.855	0.632	(350.1 / 36.6)	(32.6 / 59.5)
trainCls-8	0.967	0.978	0.982	0.453	(4199.8 / 36.1)	(42.0 / 32.3)
testCls-8	0.896	0.952	0.978	0.308	(465.7 / 5.2)	(5.5 / 2.4)

Table 3.64: LogReg Class 8 weight ratio 1.0. This option is chosen.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.885	0.862	0.855	0.630	(350.4 / 37.0)	(32.3 / 59.1)
trainCls-8	0.961	0.972	0.984	0.253	(4229.6 / 56.7)	(12.2 / 11.7)
testCls-8	0.893	0.952	0.982	0.135	(469.5 / 6.8)	(1.7 / 0.8)

Table 3.65: LogReg Class 8 weight ratio 0.5

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
fine	0.886	0.864	0.855	0.632	(349.5 / 36.4)	(33.2 / 59.7)
trainCls-8	0.967	0.980	0.978	0.478	(4169.7 / 24.3)	(72.1 / 44.1)
testCls-8	0.892	0.947	0.973	0.376	(462.2 / 3.9)	(9.0 / 3.7)

Table 3.66: LogReg Class 8 weight ratio 1.5

3.1.8 Varying Logistic Regression Tolerance

There is an **additional Logistic parameter** for determining a tolerance for the stopping criteria. The default tolerance is 0.0001. **Do you discuss logistic regression in an earlier chapter? It's important to define how it works and what its parameters are before discussing how you tuned them.**

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.880	0.863	0.668	0.517	(234.8 / 11.1)	(147.9 / 85.0)
fine	0.886	0.864	0.855	0.632	(350.1 / 36.6)	(32.6 / 59.5)

Table 3.67: LogReg results after fine tuning, effectively had a tolerance of 0.0001

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.880	0.862	0.668	0.518	(234.9 / 11.1)	(147.8 / 85.0)
fine	0.885	0.863	0.855	0.632	(350.1 / 36.7)	(32.6 / 59.4)

Table 3.68: LogReg Tolerance 0.0001, notice that the fine pr and roc decreased by 0.001, and that the coarse roc decreased by 0.001 upon rerunning, there is some statistical variation in these metrics.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.880	0.863	0.668	0.517	(234.7 / 11.1)	(148.0 / 85.0)
fine	0.886	0.864	0.855	0.632	(350.1 / 36.6)	(32.6 / 59.5)

Table 3.69: LogReg Tolerance 0.0001. This option is chosen.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.880	0.862	0.668	0.517	(234.8 / 11.1)	(147.9 / 85.0)
fine	0.885	0.863	0.855	0.632	(350.1 / 36.7)	(32.6 / 59.4)

Table 3.70: LogReg Tolerance 0.000001

3.1.9 Varied Sample Weight On Test Set and Dropping Intermediate ROC Curve Values

The sample weight, as stated previously, weights fine instances in the ROC and PR curves by the ratio of total number of instances in the test set divided by the total number of positives in the test set. This

weighting is performed identically on the coarse and fine classifier. The ROC curve library has a parameter to determine whether or not to drop some suboptimal thresholds which do not appear on a plotted ROC curve [2]. The default setting is to drop intermediate values True, which has the counterintuitive result of a roc curve having on the order of 150 points even though 497 points are passed to the roc curve library method. If drop intermediate values is set to false then the full 497 points are returned in the calculated roc curve.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.880	0.862	0.668	0.517	(234.8 / 11.1)	(147.9 / 85.0)
fine	0.885	0.863	0.855	0.632	(350.1 / 36.7)	(32.6 / 59.4)

Table 3.71: LogReg sample weights, drop intermediate values True. The default option is chosen.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.649	0.862	0.668	0.517	(234.8 / 11.1)	(147.9 / 85.0)
fine	0.663	0.863	0.855	0.632	(350.1 / 36.7)	(32.6 / 59.4)

Table 3.72: LogReg no sample weights, drop intermediate values True

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.880	0.862	0.668	0.517	(234.8 / 11.1)	(147.9 / 85.0)
fine	0.885	0.863	0.855	0.632	(350.1 / 36.7)	(32.6 / 59.4)

Table 3.73: LogReg sample weights, drop intermediate values False

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.649	0.862	0.668	0.517	(234.8 / 11.1)	(147.9 / 85.0)
fine	0.663	0.863	0.855	0.632	(350.1 / 36.7)	(32.6 / 59.4)

Table 3.74: LogReg no sample weights, drop intermediate values False

3.1.10 Varied Logistic Regression Positive Class Weight For Full Dataset

The fine class weight for the subset of data is determined be to 7.5, this value should change and be linearly dependent upon the number of instances in the training set. The weight for the fine class is tuned using all of the data, the original value is the total number of instances in the train set divided by the total number of positives in the train set, which evaluates to 20.887. The previously determined fine class ratios are used in this analysis. The value selected is 23, this value along with 7.5 and the original values of 20.887 and

4.977 for a line with two points that define a function to map a weight original input to a new tuned weight output for all training set sizes.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.867	0.868	0.803	0.280	(1537.6 / 19.2)	(376.0 / 76.9)
fine	0.871	0.868	0.919	0.404	(1792.3 / 41.0)	(121.2 / 55.1)

Table 3.75: LogReg entire dataset, weight 20.887

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.870	0.871	0.787	0.268	(1503.2 / 17.8)	(410.4 / 78.3)
fine	0.875	0.871	0.913	0.403	(1776.5 / 37.3)	(137.1 / 58.8)

Table 3.76: LogReg entire dataset, weight 23.0. This option is chosen.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.867	0.868	0.772	0.256	(1473.0 / 17.3)	(440.6 / 78.8)
fine	0.871	0.868	0.905	0.389	(1758.8 / 35.6)	(154.8 / 60.6)

Table 3.77: LogReg entire dataset, weight 25.0.

3.1.11 Varying SVM Gamma

After the LogReg classifier is tuned with class weights, the SVM is ran again with the class weights determined by the LogReg classifier and a slight advantage for the fine grained classifier is demonstrated with the SVM as well. The SVM parameters for the Radial Basis Function kernel of Cost and Gamma are varied. The Cost is related to a penalty parameter for the error term and Gamma is the kernel coefficient and determines the relative size of the kernel. The default gamma setting is 0.002967 or $(1/\text{num-features})$ or $(1/337)$. Default cost is actually 1.0, and the default class weight is balanced which weights each class by the number of instances it has in the train set, the same fine class weights used in the LogReg classifier are used in the SVM classifier instead of the SVM's default balanced option.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.901	0.874	0.846	0.651	(336.0 / 27.2)	(46.7 / 68.9)
fine	0.896	0.865	0.871	0.598	(371.1 / 50.1)	(11.6 / 46.0)

Table 3.78: SVM Cost 1.0 Gamma 0.0029674

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.903	0.873	0.866	0.672	(348.9 / 30.4)	(33.8 / 65.7)
fine	0.890	0.857	0.865	0.554	(373.8 / 55.8)	(8.9 / 40.3)

Table 3.79: SVM Cost 2.0 Gamma 0.0029674

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.892	0.869	0.664	0.518	(231.5 / 9.8)	(151.2 / 86.3)
fine	0.899	0.870	0.868	0.623	(363.5 / 43.8)	(19.2 / 52.3)

Table 3.80: SVM Cost 0.1 Gamma 0.0029674

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.883	0.860	0.591	0.474	(194.9 / 8.0)	(187.8 / 88.1)
fine	0.884	0.853	0.858	0.544	(370.1 / 55.5)	(12.6 / 40.6)

Table 3.81: SVM Cost 0.05 Gamma 0.0029674

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.896	0.873	0.714	0.553	(257.5 / 11.6)	(125.2 / 84.5)
fine	0.902	0.874	0.871	0.640	(362.1 / 41.1)	(20.6 / 55.0)

Table 3.82: SVM Cost 0.15 Gamma 0.0029674. This cost option is chosen.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.899	0.875	0.755	0.584	(279.6 / 14.0)	(103.1 / 82.1)
fine	0.903	0.875	0.871	0.640	(362.1 / 41.2)	(20.6 / 54.9)

Table 3.83: SVM Cost 0.2 Gamma 0.0029674.

3.1.12 Varying SVM Cost

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.894	0.871	0.706	0.545	(253.6 / 11.7)	(129.1 / 84.4)
fine	0.906	0.877	0.869	0.646	(358.3 / 38.5)	(24.4 / 57.6)

Table 3.84: SVM Cost 0.15 Gamma 0.002. This option for Cost and Gamma is chosen.

title	pr	roc	acc	f1	conf (tn/fn)	conf (fp/tp)
coarse	0.883	0.864	0.664	0.516	(232.1 / 10.3)	(150.6 / 85.8)
fine	0.900	0.872	0.868	0.641	(358.5 / 39.2)	(24.2 / 56.9)

Table 3.85: SVM Cost 0.15 Gamma 0.001