Learning with Limited Minority Class Data

Taghi M. Khoshgoftaar, Chris Seiffert, Jason Van Hulse, Amri Napolitano, Andres Folleco
Florida Atlantic University, Boca Raton, FL, USA
Contact author: taghi@cse.fau.edu

Abstract

A practical problem in data mining and machine learning is the limited availability of data. For example, in a binary classification problem it is often the case that examples of one class are abundant, while examples of the other class are in short supply. Examples from one class, typically the positive class, can be limited due to the financial cost or time required to collect these examples. This work presents a comprehensive empirical study of learning when examples from one class are extremely rare, but examples of the other class(es) are plentiful. Specifically, we address the issue of how many examples from the abundant class should be used when training a classifier on data where one class is very rare. Nearly one million classifiers were built and evaluated to generate the results presented in this work. Our results demonstrate that the often used 'even distribution' is not optimal when dealing with such rare events.

1 Introduction

A common objective in data mining is to train a classification model on previously labeled data with the goal of classifying unlabeled records into one of two or more possible classes. This task becomes substantially more challenging when the classes are unevenly represented in the training data. For example, binary classification models (the only kind considered in this work) tend to favor the overrepresented (majority) class when assigning labels to examples in order to achieve the lowest overall misclassification rate. The result is a high percentage of examples from the underrepresented (minority) class being mislabeled. Such a result is often unacceptable as the minority class is often the class of interest (positive class), and misclassifying a positive class example frequently carries a higher cost than misclassifying an example of the majority (negative) class.

While recent research has addressed the issue of class imbalance, few if any studies address the issue of rare events. That is, if only a few examples of the positive class are available, but examples of the negative class are abundant, what strategy should be employed when building classification models on this data? How many examples from the negative class should be included in the training data set? Researchers of class imbalance frequently use a balanced class distribution (where the number of positive and negative examples are approximately equal), but our results demonstrate that such a strategy is not optimal when dealing with very rare events.

The contribution of this work is a comprehensive study of learning when minority class examples are very limited. Our experiments examine the impact of class distribution on learning performance when training data has as few as five and as many as 40 minority class examples. The class distribution is varied by changing the number of majority class examples while keeping the number of minority class examples constant. Experiments in this work are performed using 10 datasets from different application domains and 11 different learning algorithms. A total of 894,300 classifiers were built and analyzed, making the results presented in this work very comprehensive and statistically significant. We conduct analysis of variance (ANOVA) testing to demonstrate the statistical significance of our conclusions.

The remainder of this paper is organized as follows. Section 2 discusses related work in the area of class distribution, imbalance and rare events. Section 3 provides details about the design of our experiments. Section 4 presents our experimental results with discussion. Conclusions and future work are discussed in Section 5. Finally, the learners used in our experiments are described in the Appendix.

2 Related Work

Although there has been some recent research focusing on the class imbalance problem, few, if any, studies deal with the problem of rare events. For example, Weiss and Provost consider the issue of class distribution for decision tree induction [14] and find that the ideal distribution is often dependent on the performance metric employed. When considering overall classification accuracy, the natural class distribution tends to perform best, but when AUC (the metric used in this work) is used to measure performance, a balanced distribution is recommended. While our work also considers the class distribution issue, our focus is not on learning from imbalanced data, but rather data where ex-

0-7695-3069-9/07 $25.00 © 2007 IEEE
DOI 10.1109/ICMLA.2007.76

IEEE computer society

amples of one class are very rare but examples of the other class are abundant. Subsequently, our results differ from those in [14] even though our research is based on some of the same datasets. In addition, Weiss [13] presents a survey of mining with rarity, but unlike our work does not include a case study.

Much research in the area of class imbalance attempts to improve upon classification performance through data sampling techniques [2, 6, 7, 9]. Van Hulse et al. [12] investigates the issue of class imbalance and provides a thorough investigation of many data sampling techniques. Liu, Wu and Zhou [8] also employ several sampling techniques, and introduce two ensemble based techniques for improving classification performance when data is imbalanced. Again, Zhou [8] tends to favor the use of balanced class distributions which may be sufficient for imbalanced data, but as our work demonstrates this strategy is not optimal when dealing with rare events.

3 Experimental Design

3.1 Evaluation Metric

Binary classifiers make predictions that can be associated into four categories, described in equations 1 through 4. Suppose x_i is an example from a dataset D with class c_j, $j = 0$ or 1. c_1 is the positive class and c_0 is the negative class, while $c(x_i)$ and $\hat{c}(x_i)$ are the actual and predicted classes of x_i. The categories are defined as:

$$x_i \text{ is a } \textit{true positive} \text{ (tpos) if } c(x_i) = c_1 = \hat{c}(x_i). \quad (1)$$

$$x_i \text{ is a } \textit{true negative} \text{ (tneg) if } c(x_i) = c_0 = \hat{c}(x_i). \quad (2)$$

$$x_i \text{ is a } \textit{false positive} \text{ (fpos) if } c(x_i) = c_0 \neq \hat{c}(x_i). \quad (3)$$

$$x_i \text{ is a } \textit{false negative} \text{ (fneg) if } c(x_i) = c_1 \neq \hat{c}(x_i). \quad (4)$$

In our experiments, we use the area under the receiver operating characteristic (ROC) curve (AUC) [10] to evaluate classification performance. The ROC curve graphs the true positive rate on the y-axis versus the false positive rate on the x-axis, and therefore measures the tradeoff between detection rate and false alarm rate. Higher AUC values denote a classifier with generally better performance (in general, a higher AUC implies a higher true positive rate with a lower false positive rate, which is preferred in most applications). AUC values range from zero (worst) to one (best). Two different classifiers can be evaluated by comparing their AUC values.

3.2 Dataset Descriptions

The ten datasets used throughout this study are listed in Table 1 and include four datasets (letter-vowel, pendigits, satimage, phoneme) from the UCI repository [3]. Another five datasets (identified with a '+') have been used extensively in previously published work by our group, and one dataset (mammography) was made available for our study by Dr. Nitesh Chawla [5]. Table 1 provides the name of each dataset in the first column, followed by the number of positive class examples ($\#P$), the number of negative class examples ($\#N$), the percentage of examples belonging to the positive class ($\%P$), and the number of attributes ($\#Attr$).

Dataset	#P	#N	% P	# Attr.
Sp3+	47	3494	1.33	43
Sp4+	92	3886	2.31	43
mammography	260	10923	2.33	7
Sp2+	189	3792	4.75	43
Sp1+	229	3420	6.28	43
pendigits	1055	9935	9.60	17
satimage	626	5809	9.73	37
JM1+	1687	7163	19.06	16
letter-vowel	3878	16122	19.39	17
phoneme	1586	3818	29.35	6

Table 1. Base dataset characteristics

Datasets with more than two classes were transformed to two-class problems, as due to space considerations we could not consider multi-class problems. For satimage, the class attribute originally contained six different values, so we utilized the fourth as the minority class and combined the rest to form the majority class. The pendigits dataset originally had ten different classes, and we used the sixth as the minority class and combined the rest into a single majority class. For letter-vowel, all five vowels (not 'Y') were combined to form the minority class while the rest were combined into the majority class. The phoneme dataset already contained a binary class, and no transformation was performed.

Sp1, Sp2, Sp3, and Sp4 datasets represent four different releases of a very large legacy telecommunication system, written in a proprietary high level language (Protel) similar to Pascal, using the procedural development paradigm and maintained by professional programmers in a large organization. The JM1 project (available from NASA's metrics data program) , written in C, is a real-time ground based system that uses simulation to generate predictions for space missions.

3.3 Experimental Method

From the base datasets described in Section 3.2, we derive new training datasets by varying two parameters, $\#P$ and $\%P$. 5-fold cross validation is used in the dataset derivation process. First, we divide the base dataset into five folds. Using four of the five folds, we randomly sample $\#P$ positive class examples to be used in the training dataset. From these four folds, we also randomly sample $\#N$ negative class examples so that the percentage of positive class examples in the training dataset is $\%P$. The remaining fold is used as test data and is not sampled. Using

$\%P$	$\#P = 5$	$\#P = 10$	$\#P = 20$	$\#P = 40$
65.0	3	5	11	22
50.0	5	10	20	40
35.0	9	19	37	74
25.0	15	30	60	120
20.0	20	40	80	160
15.0	28	57	113	227
12.5	35	70	140	280
10.0	45	90	180	360
9.0	51	101	202	404
7.5	62	123	247	493
5.0	95	190	380	760
3.5	138	276	551	1103
2.0	245	490	980	1960
1.0	495	990	1980	3960

Table 2. Number of negative class examples ($\#N$) in derived datasets

$\%P$	$\#P = 5$	$\#P = 10$	$\#P = 20$	$\#P = 40$
65.0	0.655975	0.693620	0.729204	0.760139
50.0	0.678524	0.716579	0.744065	0.773631
35.0	0.690436	**0.723626**	**0.747896**	**0.777039**
25.0	**0.691485**	0.721731	0.745941	0.776036
20.0	0.689237	0.720245	0.742876	0.773476
15.0	0.686043	0.715384	0.738541	0.768285
12.5	0.683413	0.712483	0.734758	0.765926
10.0	0.680662	0.708663	0.731760	0.763110
9.0	0.678950	0.706852	0.729804	0.761659
7.5	0.674493	0.704264	0.726500	0.759436
5.0	0.668533	0.698445	0.720590	0.754739
3.5	0.663778	0.692865	0.717383	0.749125
2.0	0.656356	0.686054	0.710711	0.741982
1.0	0.647586	0.677158	0.703086	0.767999

Table 3. Mean AUC value for each $\#P$ and $\%P$

the training data, we construct 11 classifiers using the learners described in the Appendix. These classifiers are applied to the test data and performance is judged using AUC as described in Section 3.1. This process is repeated five times, so that each fold is used as test data once.

The above procedure is repeated for each value of $\#P$ and $\%P$ shown in Table 2. Additionally, the entire process is repeated 30 times to prevent any biasing of the results that may occur due to the sampling process. If all combinations of $\#P$ and $\%P$ in Table 2 were possible, 924,000 classifiers would be built and evaluated during the course of this study (30 repetitions × 5 folds × 4 values of $\#P$ × 14 values of $\%P$ × 11 learners × 10 datasets). However, due to the limited sizes of some of the original datasets, not all scenarios listed in Table 2 can be implemented, reducing the actual number of classifiers built to 894,300. All four Software Project datasets (Sp1, Sp2, Sp3, and Sp4) and the 'phenome' dataset had majority size restrictions. For example, the $\#P = 40$ and $\%P = 1\%$ dataset could not be implemented from dataset 'Sp1' since 'Sp1' has only 3420 negative class examples.

By considering the results for a fixed value of $\#P$ and varying $\%P$ we can empirically identify the optimal class distribution for training learners when data contains very rare events. In doing so, we simulate the scenario where there are very few positive examples available, but negative examples are available in abundance. The remainder of this paper addresses the issue of how many negative (majority) class examples should be used when building classifiers on data containing very few positive (minority) examples.

4 Experimental Results

The results presented in this section are the average results across all 11 learners and 10 base datasets. We present these average results to provide the reader with a general understanding of learning from data with rare events. Learner-specific and dataset-specific results cannot be included due to space considerations. The results of our empirical investigation are presented and discussed in Section 4.1. Section 4.2 provides ANOVA analysis, demonstrating the statistical significance of our results.

4.1 Class Distribution and Rare Events

Table 3 presents the mean AUC values for each $\#P$ and $\%P$, averaged across all 11 learners and 10 datasets. For each $\%P$ the best mean AUC value is identified by **bold** print. For three of the four values of $\#P$, we find $\%P = 35$ to be the optimal class distribution. This suggests that when learning from data where very few examples of one class is available, a ratio of 2:1 negative to positive classes is ideal. For the case where $\#P = 5$, the best results are achieved when $\%P = 25$ (a 3:1 ratio of negative to positive examples). These results contradict those in previous research on class imbalance where an even distribution is found to achieve the best performance [14]. This does not imply fault in the existing research, but it does demonstrate the difference between simple class imbalance and truely rare events.

Figure 1 presents this information visually. This figure plots the mean AUC values for each $\#P$ and $\%P$. The vertical dashed line indicates the recommended class distribution of $\%P = 35$. As shown in the figure, as $\%P$ is decreased from 65% to 35%, performance consistently increases. For all but one $\#P$ value, performance reaches its maximum for $\%P = 35$ and decreases as $\%P$ is decreased below 35%. For $\#P = 5$, performance continues to increase slightly when $\%P$ is decreased to 25%, where its maximum performance is achieved. Also, Figure 1 clearly shows that performance is better for larger values of $\#P$, which is not surprising due to the fact that there is more in-

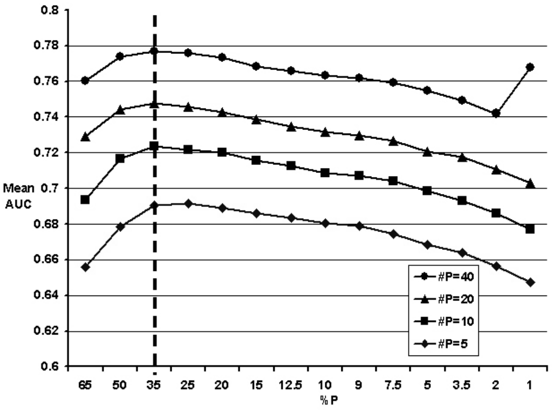

Figure 1. Effect of varying $\%P$ on AUC

Factor	DoF	MS	F	p-value
learner (δ)	10	538.78	5430.17	0.000
$\%P$ (π)	13	29.49	228.62	0.000
$\#P$ (γ)	3	181.27	6089.81	0.000
$\delta \times \pi$	130	49.17	38.12	0.000
$\gamma \times \delta$	30	15.12	50.80	0.000
$\pi \times \gamma$	39	2.76	7.12	0.000
$\delta \times \pi \times \gamma$	390	9.412	2.43	0.000

Table 4. ANOVA Table

#P	N	mean AUC	HSD
40	40260	0.763617	A
20	46200	0.730222	B
10	46200	0.705569	C
5	46200	0.674677	D

Table 5. Main Factor: $\#P$

formation from both the minority and majority classes when $\#P$ is larger. The increase in performance when $\#P = 40$ and $\%P = 1$ is due to the fact this experiment could not be performed on some datasets due to size constraints as discussed in Section 3.3. These datasets resulted in lower AUC values, causing the average AUC to be higher when they were omitted.

The results in the section clearly demonstrate the impact of $\%P$ on classification performance using AUC as the evaluation metric. It is shown that the balanced distribution, which may yield the best performance when data is imbalanced, does not result in optimal performance when data contains very rare events. Instead, a ratio of 2:1 or even 3:1 in favor of the majority class results in superior classification performance when data from one class is very limited. Section 4.2 supports the statistical significance of these findings through ANOVA analysis.

4.2 ANOVA Analysis

In this section, we present a full factorial, three factor analysis of variance (ANOVA) model to demonstrate the statistical significance of the results presented in Section 4.1. A three factor, full factorial ANOVA model can be written as:

$$\psi_{ijkl} = \mu + \delta_i + \pi_j + \gamma_k + (\delta\pi)_{ij} + (\delta\gamma)_{ik} + (\pi\gamma)_{jk} + (\delta\pi\gamma)_{ijk} + \epsilon_{ijkl}$$

where:

- ψ_{ijkl} is the l^{th} value of the response variable ψ (the AUC) for the i^{th} level of δ, j^{th} level of π and k^{th} level of γ.
- μ is the overall mean of the response variable ψ.
- δ_i, π_j, γ_k are the treatment effects of the i^{th}, j^{th}, k^{th} level of the experimental factors: learner, $\%P$ and $\#P$, respectively; (e.g., $i = 1, \ldots, 11$ represents the 11 learners used in this study).
- $(\delta\pi)_{ij}$, $(\delta\gamma)_{ik}$ and $(\pi\gamma)_{jk}$ are second-order cross-effect terms between (δ, π), (δ, γ) and (π, γ), respectively.
- $(\delta\pi\gamma)_{ijk}$ is the third-order cross-effects term amongst the three ANOVA factors δ, π and γ.
- ϵ_{ijkl} is the random error of the statistical model.

ANOVA models can be used to test the hypothesis that the mean AUC for each level of the main factors are equal against the alternative hypothesis that at least one is different. If the alternative hypothesis (i.e., that the mean AUC of at least one level of the factor is different from the other levels) is accepted, numerous procedures can be used to determine which levels of the factor result in mean AUCs that are significantly different from the others. This requires a comparison of the means of two different levels, with the null hypothesis that they are equal. The comparison tests in this work utilize Tukey's Studentized Range (Honestly Significant Difference or HSD) test. All tests of statistical significance discussed use an $\alpha = 5\%$ significance level.

The ANOVA table is presented in Table 4. This first column lists the main experimental factors and their interactions. The remaining columns provide the degrees of freedom (DoF), mean square (MS), F-statistic (F) and the p-value of the F-statistic. As shown in the table, all factors and their interactions are significant, with p-values much less than 5% (all p-values are actually less than 0.001, and hence denoted by 0.000 in the table).

Table 5 shows the analysis of the main factor: $\#P$. The first column identifies the different levels of the factor $\#P$.

$\%P$	N	mean AUC	HSD
35	12870	0.733665	A
25	12870	0.732715	A
20	12870	0.730381	BA
50	12870	0.727035	BC
15	12870	0.726006	C
12.5	12870	0.723074	DC
10	12870	0.719970	DE
9	12870	0.718231	FE
7.5	12870	0.715064	F
5	12870	0.709444	G
65	12870	0.708442	HG
3.5	12870	0.704676	H
2	12870	0.697668	I
1	11550	0.689094	J

Table 6. Main Effect: $\%P$

The second column (N) provides the number of observations at that level of $\#P$. Adding the values in this column results in a total of 178,860 observations, $\frac{1}{5}$ of the total number of classifiers built in our experiment. This is because the results of each fold of the 5-fold cross validation are combined resulting in one observation. N is smaller for $\#P = 40$ due to the dataset size restrictions discussed in Section 3.2. The third column (mean AUC) provides the mean value for the N observations at that level of $\#P$. Finally, the letters in the last column indicate the HSD grouping of the levels of $\#P$. That is, if two levels of $\#P$ have the same letter in the HSD column, their mean AUC values are not significantly different. This analysis supports our conclusion that classification performance is significantly improved by increasing the number of minority examples in the dataset. Again, this result is not surprising since datasets with larger values of $\#P$ contain more information on which classifiers can be trained.

Table 6 shows the analysis of $\%P$. The columns of this table are organized in a similar way to those in Table 5. This analysis demonstrates the main contribution of this paper. That is, it shows the relative performance of the different values of $\%P$ and identifies $\%P = 35$ to be the class distribution that results in the best classification performance. While $\%P = 35$ has the highest mean AUC, there is no significant difference between the mean AUCs for $\%P = \{35, 25, 20\}$, as indicated by the fact that all three of these values of $\%P$ are in group A. $\%P = 35$ and $\%P = 25$ both outperform the commonly used class distribution of $\%P = 50$, which is a member of group B with $\%P = 20$ and group C with $\%P = 15$ and $\%P = 12.5$. Therefore we conclude that selecting a majority:minority ratio of 2:1 or 3:1 significantly outperforms the often used ratio of 1:1 (in research on class imbalance [14, 8]) when training classification models on data containing rare events.

4.3 Threats to Validity

Experimental research commonly includes a discussion of two types of threats to validity [16]: threats to *internal* validity and *external* validity. Threats to internal validity are unaccounted influences that can impact the empirical results. Learners were constructed using a publicly available, high quality, and commonly-used data mining tool called WEKA [15]. ANOVA models were constructed using the SAS GLM procedure [11]. All output was validated for accuracy by members of our research group, giving us confidence in the efficacy of the results.

External validity considers the generalization of the results outside the experimental setting, and what limits, if any, need to be applied. A significant amount of experimentation was performed in this study, with 894,300 learners constructed based on 10 datasets, many of which are publicly available and all of which are frequently used and well understood by our research group. The base datasets were carefully analyzed and selected for use in this study due to their diverse statistical properties, as demonstrated in Table 1.

5 Conclusions

This paper presents an empirical study of learning when very few examples of one class are available for training models. Specifically, we address the issue of how many majority class examples should be included in a training dataset when the minority class is rare. While similar studies have been performed using imbalanced datasets, this work is, to our knowledge, the first to address the issue of class distribution when data contains very rare events. The results presented in this paper clearly demonstrate the difference between the problems of class imbalance and rare events.

The major contribution of this work is to present a comprehensive empirical study of the ideal class distribution when training classifiers on data where examples from one class are very limited. We present a very practical scenario where few (between 5 and 40) positive class examples are available, while negative class examples are available in abundance. Under such circumstances, we address the question of how many negative class examples should be included in the training dataset to achieve the best possible classification performance. The experiments performed in this study use 11 different learners, 10 base datasets, four values of $\#P$ (the number of positive examples in the training data) and 14 values of $\%P$ (class distribution). All experiments were repeated 30 times, resulting in the construction of 894,300 classifiers, making the results presented in this paper statistically significant.

Our findings indicate that the optimal class distribution when learning from data containing rare events is approximately 2:1 (majority:minority). For the most extreme level of rarity ($\#P = 5$) in this study, the optimal ratio was 3:1, although 2:1 resulted in only slightly lower classification performance. These results differ from those presented in studies on class imbalance where a ratio of 1:1 is found to perform the best [14]. While the issues of class imbalance and rare events are related, it is clear that they must be treated differently when attempting to maximize classification performance.

Future work will include additional empirical investigations of learning from data with rare events, including the application of performance enhancing techniques such as cost sensitive classification. Also, similar experiments can be performed on additional datasets to further solidify the results presented in this paper.

References

[1] D. W. Aha. *Lazy learning*. Kluwer Academic Publishers, Norwell, MA, USA, 1997.

[2] R. Barandela, R. M. Valdovinos, J. S. Sanchez, and F. J. Ferri. The imbalanced training sample problem: Under or over sampling? *In Joint IAPR International Workshops on Structural, Syntactic, and Statistical Pattern Recognition (SSPR/SPR'04), Lecture Notes in Computer Science 3138*, (806-814), 2004.

[3] C. Blake and C. Merz. UCI repository of machine learning databases. *http://www.ics.uci.edu/ mlearn/ MLRepository.html*, 1998. Department of Information and Computer Sciences, University of California, Irvine.

[4] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[5] N. V. Chawla, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer. Smote: Synthetic minority oversampling technique. *Journal of Artificial Intelligence Research*, (16):321–357, 2002.

[6] C. Drummond and R. C. Holte. C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on Learning from Imbalanced Data Sets II, International Conference on Machine Learning*, 2003.

[7] H. Han, W. Y. Wang, and B. H. Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In *In International Conference on Intelligent Computing (ICIC'05). Lecture Notes in Computer Science 3644*, pages 878–887. Springer-Verlag, 2005.

[8] X. Liu, J. Wu, and Z. Zhou. Exploratory under-sampling for class-imbalance learning. In *Proceedings of the Sixth International Conference on Data Mining*, 2006.

[9] M. Maloof. Learning when data sets are imbalanced and when costs are unequal and unknown. In *Proceedings of the ICML'03 Workshop on Learning from Imbalanced Data Sets*, 2003.

[10] F. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42:203–231, 2001.

[11] SAS Institute. *SAS/STAT User's Guide*. SAS Institute Inc., 2004.

[12] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano. Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th Annual International Conference on Machine Learning (ICML 2007)*, pages 935–942, Corvalis, OR, June 2007.

[13] G. M. Weiss. Mining with rarity: A unifying framework. *SIGKDD Explorations*, 6(1):7–19, 2004.

[14] G. M. Weiss and F. Provost. Learning when training data are costly: the effect of class distribution on tree induction. *Journal of Artificial Intelligence Research*, 19:315–354, 2003.

[15] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, California, 2nd edition, 2005.

[16] C. Wohlin, P. Runeson, M. Host, M. C. Ohlsson, B. Regnell, and A. Wesslen. *Experimentation in Software Engineering: An Introduction*. Kluwer International Series in Software Engineering. Kluwer Academic Publishers, Boston, MA, 2000.

Appendix

The results presented in this study are based on the average performance of 11 different learners. All learners were implemented in the WEKA tool [15]. Default parameter changes were done only when experimentation showed a general improvement in the classifier performance based on preliminary analysis.

Naive Bayes (NB) utilizes Bayes's rule of conditional probability and is termed 'naive' because it assumes conditional independence of the features. *Logistic regression* (LR) is a statistical regression model for categorical prediction. *RIPPER* (Repeated Incremental Pruning to Produce Error Reduction) is a rule-based learner and is named JRip in WEKA. The default WEKA parameters for these three learners were not changed.

C4.5 is a benchmark decision tree learning algorithm. Two different versions of the C4.5 classifier were used. C4.5 (D) uses the default parameter settings in WEKA, while C4.5 (N) uses no decision-tree pruning and Laplace smoothing [14].The *random forests* (RF) classifier [4] uses bagging and the 'random subspace method' to build an ensemble of randomized decision trees which are combined to produce the final prediction. RF's 'Number of Trees' parameter was changed to 100 from its default value of 10.

K nearest neighbors [1] (kNN) learners were built with changes to two parameters. The 'distanceWeighting' parameter was set to 'Weight by 1/distance'. Two different 'kNN' learners were built using $k = 2$ and $k = 5$ and were denoted '2NN' and '5NN'.

For a *Multilayer perceptrons* (MLP) learner (a type of neural network), the 'hiddenLayers' parameter was changed to '3' to define a network with one hidden layer containing three nodes, and the 'validationSetSize' parameter was changed to '10' to cause the classifier to leave 10% of the training data aside to be used as a validation set to determine when to stop the iterative training process. *Radial basis function networks* (RBF) are another type of artificial neural network. The only parameter change for RBF was to set the parameter 'numClusters' to 10.

The *support vector machine* (SVM) learner called SMO in WEKA had two changes to the default parameters: the complexity constant 'c' was set to 5.0 and 'buildLogisticModels' was set to 'true'. By default, a linear kernel was used.