COGS 118A - Comparing Binary Classifiers

Nicholas Peterzell NPETERZE@UCSD.EDU
Department of Cognitive Science
University of California, San Diego

Abstract

In this paper, I will attempt to partially recreate the study done in the paper *An Empirical Comparison of Supervised Learning Algorithms*, or CNM06, by Rich Caruana and Alexandra Niculescu-Mizil. I will analyze the effectiveness of three of the algorithms used in the Caruana study (logistic regression, decision trees, and random forests) on four datasets that I have selected myself.

Keywords: Binary classification, Machine learning, Linear regression, Decision trees, Random forests

1. Introduction

The paper *An Empirical Comparison of Supervised Learning Algorithms* (referred to as CNM06 from here on) was a study done in 2006 by Rich Caruana and Alexandra Niculescu-Mizil. In it, they compared ten supervised learning methods over eleven datasets in order to find out which one performed the best on average.

The experiment in this paper involves replicating part of the CNMO6 study on a smaller scale using fewer of the algorithms and error metrics present in that study. The algorithms selected for this experiment are linear regression, decision trees, and random forests. They were run on four different sets of data that will be described later, and evaluated using three different metrics: accuracy, area under the ROC curve (ROC), and f1 score. This experiment will not replicate the calibration step of the CNM06 paper.

For the most part, the results from this experiment were the same as those found in CNM06; random forests performed the best by a good margin, and the decision trees and logistic regression algorithms performed more poorly. In fact, random forests outperformed the other algorithms on *every* dataset and metric combination except for the BEAN dataset, where logistic regression was slightly higher. However, the decision trees algorithm in this experiment actually performed *worse* than the logistic regression algorithm, which is the reverse of what is seen in CNM06. This is likely due to my own experiment using different datasets than CNM06.

2. Methods

2.1 Learning Algorithms

The hyperparameters specified for use in the grid search are for the most part the same as what they were in CNM06, save for the decision tree algorithm which has a variation on the hyperparameters used in CNM06. Additionally, for this experiment I have selected extra hyperparameters to include in the search, such as several different solvers for the logistic regression algorithm and varying class weights for the decision trees algorithm.

Logistic regression: The logistic regression algorithm used a variety of solvers for its grid search: namely, the saga, liblinear, newton-cg, lbfgs, and sag solvers. Each solver was accompanied by every one of the following penalties that it supports: L1, L2, elasticnet, and an option for no penalty. The values of C were the same as in CNM06, ranging from 10^{-8} to 10^{4} in steps of a factor of ten.

Decision trees: The decision tree algorithm has had its hyperparameters for the grid search changed from what it was in CNM06. The variation of the splitting criteria and pruning options have remained, but instead of testing different hyperparameters for smoothing and testing different tree models, the max depth was tested with values from 1 to 20, and the class weights were tested with all possible combinations of the values {1, 2, 5, 10}.

Random Forests: Random forests were set up much the same as they were in CNM06, with 1024 estimators, varied splitting criterion, and max feature values of {1, 2, 4, 6, 8, 12, 16, 20}. The only difference is that the Breiman-Cutler and Weka implementations were not considered.

2.2 Performance Metrics

As was said before, the performance metrics used for this experiment are accuracy, area under the ROC curve (ROC), and F1 score. Accuracy is a common metric; however, since it only considers the total number classified correctly out of all the points considered, it can give extremely high or low scores on imbalanced data that has a high percentage of positives or negatives, respectively. The ROC metric measures the area under the ROC curve and uses that as a probability of how likely a classifier is to correctly distinguish between classes; in our case, it is only considering the models' ability to differentiate between two classes. F1 score was included to compensate for the accuracy metric's lack of efficacy on imbalanced datasets, since it is much more effective than accuracy on these kinds of datasets.

2.3 Datasets

There were 4 datasets used in this experiment. The BEAN dataset contains data on the physical attributes of seven different types of dry BEANs. For this experiment, we picked the Seker class of bean as our positive predictor.

The BIKE dataset contains data about bike sharing, a service where you can rent a bike for a period of time; on this dataset we attempt to predict whether a certain day is a workday or not. Furthermore, the instant column was dropped as it was an unnecessary index column, and the "dteday" column was dropped as it contained time information in the YYYY-MM-DD format, and that information is already present in the columns directly after, removing the need to encode this column.

The SHOPPERS dataset contains information about customers shopping online, and we attempt to predict whether or not a customer is a new customer or a returning customer.

The final dataset, STARS, contains astronomical information about stars in space. On this dataset we attempt to predict whether or not a star is a neutron star.

All of the target variables in these four datasets were originally binary; no operations were performed on these columns as they were already in the correct form. In each dataset, columns with nonbinary data were scaled. The training set size for each of these problems follows the guidelines in CNM06; the size of the training set is 5000 samples, with the rest of the samples being made the testing set.

As can be seen in Table 1, the BEAN, SHOPPERS, and STARS datasets are highly negatively imbalanced; the BIKE dataset is less imbalanced, but is still positively skewed.

Table 1: Description of Dataset

DATASET	#ATTR	TRAIN SIZE	TEST SIZE	%POZ
BEAN	16	5000	8611	15%
BIKE	14	5000	12379	68%
SHOPPERS	17	5000	7330	14%
STARS	8	5000	12898	9%

3. Experiment

The algorithms in this experiment are set up to run just as they did in CNM06; each algorithm runs five trials, in each creating a train-test split from the dataset with a training set size of 5000 and the rest of the data being used as the testing data. Then it takes in the training data and conducts a hyperparameter grid search with 5 stratified k-folds on each of these splits in order to find the hyperparameters that result in the best result on each of the error metrics (accuracy, ROC, and f1 score). Then, it fits each of the models with the testing data from each dataset to find the test results for the three error metrics. We do this for each dataset, and then calculate the mean test set performance across trials of each algorithm/dataset combo (Table 2). We also calculate the mean test set performance across trials for each algorithm/metric combo (Table 3).

In the tables that follow, the highest performing algorithm in each column is bolded, and algorithms in the same column with an insignificant difference from the best performing one are labeled with a *. The criterion for significance is a p-value of less than 0.05 when compared to the highest-performing algorithm using an independent t-test. The raw scores used to construct these tables are available in Table 5 in the appendix.

Table 2: Mean test set performance across trials for each algorithm/dataset combination

MODEL	BEAN	BIKE	SHOPPERS	STARS	Mean
Logistic Regression	**0.968**	0.805	0.504*	0.916*	0.798
Decision Tree	0.537	0.665	0.535*	0.514	0.563
Random Forest	0.964*	**1.000**	**0.657**	**0.925**	**0.887**

As was mentioned in the introduction, the random forest classifier outperformed the other classifiers overall. On the BEAN dataset, logistic regression scored slightly higher than random forests, and much higher than decision trees did. Additionally, while there was a significant difference between logistic regression and decision trees on this dataset, this was not the case between logistic regression and random forests.

On the BIKE dataset, random forests scored the highest by a large margin. Logistic regression came in second this time, and decision trees again scored much lower than both of the other classifiers. On this dataset, the results for logistic regression and decision trees were both significantly different from the results of the random forests.

On the SHOPPERS dataset, random forests again scored the highest out of the three, although all the classifiers scored noticeably lower on this dataset than on the previous ones. One thing to note on this dataset is that decision trees scored higher than logistic regression, a break from the pattern seen in the other problems. Both the logistic regression and decision tree algorithms had an insignificant difference in performance from the random forest algorithm on this dataset.

The STARS dataset again saw random forests taking the highest score out of the three. Logistic regression has moved back to second place in this column, and decision trees are again performing much more poorly than the other two classifiers. The difference between random forests and logistic regression is not considered significant on this dataset, and the difference between random forests and decision trees is.

The last column shows the mean score of each algorithm across datasets. As we would expect from the individual dataset results, random forests have the highest score, followed by logistic regression in second and decision trees in last place. The mean scores across datasets for logistic regression and decision trees are both significantly lower than the mean score for random forests.

A collection of the p-values used to determine significance on Table 2 are available in Table 6 in the appendix.

Table 3: Mean test set performance across trials for each algorithm/metric combination

MODEL	ACC	ROC	F1	Mean
Logistic Regression	0.914	0.768	0.713*	0.798
Decision Tree	0.826	0.500	0.362	0.563
Random Forest	**0.959**	**0.878**	**0.822**	**0.887**

In the first column of Table 3, we can see that random forests achieved the highest accuracy score averaged across all 4 datasets. Logistic regression achieved the next highest average accuracy, and although decision trees reached a relatively high accuracy, it is still lower than the other two. Both the difference between random forests and logistic regression and the difference between random forests and decision trees were significant.

This trend continues in the ROC column - random forests give the highest value of this score, with logistic regression again coming in second. Decision trees have a very low ROC score in this case. Again, random forests score significantly higher than both logistic regression and decision trees in this column.

Random forests again give the highest score in the F1 column, and logistic regression again comes in second place. However, decision trees have a markedly low F1 score, and much lower than the F1 scores of the other two metrics. In this column, the difference between random forests and logistic regression was not significant, while the difference between random forests and decision trees were.

When looking at the mean column, we see that random forests achieved the highest average score across metrics, while logistic regression and decision trees predictably came in second and third place respectively. There was also a significant difference between the average random forest score across metrics and the average logistic regression score across metrics, and between the average random forest score across metrics and the average decision tree score across metrics.

A collection of the p-values used to determine significance on Table 3 are available in Table 7 in the appendix.

Comparing the dataset/algorithm testing set performance scores in Table 2 to the their respective training set performance scores (Table 4 in the appendix), we first see that logistic regression performed very well on the BEAN and STARS datasets, and moderately well on the BIKE dataset, while performing very poorly on the SHOPPERS dataset. Looking next at decision trees, we can see that the algorithm performed poorly on every set of training data. Finally, we see that random forests had a perfect accuracy across all training datasets. These results are similar to the scores seen in Table 2; in all except a couple instances, random forests scored the highest, followed by logistic regression, with decision trees scoring the lowest.

4. Discussion

When considering the results across Table 2 and Table 3, we can see that a clear pattern is present between the three algorithms used; random forests almost always score higher than the other two classifiers, logistic regression scores the second highest in all cases except for one, and decision trees perform worse than the other two almost every time.

Looking at the results in this paper, it would seem that this experiment's results partially match the CNM06 paper; firstly, in both sets of results, random forests score higher than the other classifiers involved. However, a difference arises in that the ranking of decision trees and logistic regression are swapped in this experiment. Although this paper found logistic regression to perform a good amount better than decision trees, CNM06 found decision trees to be slightly higher performing than logistic regression. A plausible explanation for this difference can be found in the datasets used; decision trees tend to perform worse when they are used to classify imbalanced data. Looking at the datasets used in this experiment, one of them is moderately imbalanced while the other three are very imbalanced. This would cause the decision tree algorithm to perform worse than usual, and would explain the difference in results between this experiment and CNM06. Thus, the differences between the two can likely be attributed to the differing datasets used by each, and this experiment can be considered a qualified success.

Appendix

Table 4: Mean training set performance across algorithm/dataset combinations

MODEL	BEAN	BIKE	SHOPPERS	STARS	Mean
Logistic Regression	0.967	0.805	0.504	0.916	0.798
Decision Tree	0.537	0.665	0.535	0.514	0.563
Random Forest	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**

Table 5: Raw test set scores

Table is very large and is located here:

https://docs.google.com/spreadsheets/d/1cShWPBbOaDtbvlkqLGktBUMqijbb1b17hkgS7SOrB-0/edit#gid=0

Table 6: P-values for Table 2 (Random forest and logistic regression vs others)

MODEL	Logistic Regression	Decision Tree	Random Forest
Logistic Regression - BEAN	1	3.5^{-7}	0.46
Random Forest - BIKE	4.9^{-10}	1.5^{-10}	1
Random Forest - SHOPPERS	0.11	0.14	1
Random Forest - STARS	0.61	5.0^{-5}	1
Random Forest - Mean	0.01	2.1^{-8}	1

Table 7: P-values for Table 3 (Random forest vs others)

MODEL	Logistic Regression	Decision Tree
Random Forest - Accuracy	0.03	8.09^{-7}
Random Forest - ROC	0.04	6.5^{-14}
Random Forest - F1	0.24	8.03^{-7}
Random Forest - Mean	0.02	2.0^{-13}

References

Caruana, Rich., & Niculescu-Mizil, Alexandru, (2006). An Empirical Comparison of Supervised Learning Algorithms. Department of Computer Science, Cornell University, Ithaca. https://www.cs.cornell.edu/~caruana/ctp/ct.papers/caruana.icml06.pdf

Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

"Classification: Accuracy | Machine Learning Crash Course." *Google*, Google, developers.google.com/machine-learning/crash-course/classification/accuracy.

Narkhede, Sarang. "Understanding AUC - ROC Curve." *Medium*, Towards Data Science, 14 Jan. 2021, towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5.

Huilgol, Purva. "Accuracy vs. F1-Score." *Medium*, Analytics Vidhya, 24 Aug. 2019, medium.com/analytics-vidhya/accuracy-vs-f1-score-6258237beca2#:~:text=Accuracy%20is%20used%20when%20the,and%20False%20Positives%20are%20crucial&text=In%20most%20real%2Dlife%20classification,to%20evaluate%20our%20model%20on.

Brownlee, Jason. "Cost-Sensitive Decision Trees for Imbalanced Classification." Machine Learning Mastery, 20 Aug. 2020, machinelearningmastery.com/cost-sensitive-decision-trees-for-imbalanced-classification/#:~:text=The%20decision%20tree%20algorithm%20is,two%20groups%20with%20minimum%20mixing.&amp;text=How%20the%20standard%20decision%20tree%20algorithm%20does%20not%20support%20imbalanced%20classification.