Timothy Nordahl
A14627185
COGS 118A, A03

A Replication of the Methods From “An Empirical Comparison of Supervised Learning Algorithms”

Abstract

Here in this paper, we will be furthering the research done in “An Empirical Comparison of Supervised Learning Algorithms” by Rich Caruana and Alexandru Niculescu-Mizil by reproducing their methods to produce clear results comparing the outputs and capabilities of logistic regression, decision tree, and random forest classifiers.

Introduction

The research performed and presented by Rich Caruana and Alexandru Niculescu-Mizil in “An Empirical Comparison of Supervised Learning Algorithms” (hereafter referred to as CNM06) has been invaluable and influential in measuring and comparing the performances of algorithms in supervised machine learning. By applying the algorithms they tested to a variety of datasets and performance metrics, they published clear data that shows the strengths and weaknesses of each algorithm as they stand against the rest of the population. This paper is an attempt to replicate the methods and procedures that Caruana and Niculescu-Mizil followed for a subset of the algorithms they tested, including logistic regression, decision trees, and random forests over a range of binary classification datasets. One deviation from the methods of CNM06 and this paper, however, is that we did not further calibrate our results with either Platt Scaling or Isotonic Regression.

What was found was that random forests consistently outperformed both logistic regression and decision trees both in comparing by metric and by dataset. This result echoes what Caruana and Niculescu-Mizil found in their own paper, that decision trees and logistic regression struggle to keep up with random forests in most situations.

Methodology

Learning Algorithms

For each algorithm, a hyperparameter search was conducted across 5 trials over 5000 samples from each dataset to find ideal parameters for the metric and dataset.

Logistic Regression (LOGREG):

We trained models from Python library SciKitLearn’s LogisticRegression class, tuning parameters on the available solvers (newton-cg, lbfgs, liblinear, sag, saga), penalties (l1, l2, none) barring the elastic-net penalty, and regularization strength value by factors of 10 from 10^{-8} to 10^{4}. All datasets were standardized for all features before a logistic regression model was fit to it. No features were one hot encoded in any of the datasets.

Decision Tree (DECTREE):

We trained models from the Python library SciKitLearn's DecisionTreeClassifier class, tuning parameters on the available criteria parameters (gini, entropy), splitter parameters (best, random), and max feature parameters (auto, sqrt, log2, none), max depth following even values from 2 to 12 (including "**None**"), min samples split following even values from 2 to 10, and min samples leaf following even values from 2 to 10.

Random Forest (RF):

We trained models from the Python library SciKitLearn's RandomForestClassifier class, using a forest of 1,024 trees and tuning parameters both with and without a "warm start," on both gini and entropy criteria, and varying the maximum feature values of sqrt, log2, None, and 1, 2, 4, 6, 7.

Performance Metrics

For performance metrics, we examined two "threshold metrics" (as they are referred to in CNM06), Accuracy (ACC) and F1 Score (F1) and one "order/rank metric," area under the receiver operating characteristic curve (ROC AUC).

Data Sets

We use four binary classification datasets, all found on UC Irvine's public data repository. These sets are the "Avila Data Set" (AVILA), the "EEG Eye State Data Set" (EEG), the "Occupancy Detection Data Set" (OCCU), and the "Letter Recognition Data Set" (LETTER). For AVILA, the alphabetic classifiers A through F were designated as the "1" class while the rest were designated as the "0" class. For LETTER, letters A through M were designated as the "1" class and N through Z were designated as the "0" class. For OCCU, as cyclical "datetime" was an included attribute, the date was dropped as it was deemed irrelevant and the time of day was represented as seconds from midnight and transformed into sin and cos attributes to maintain cyclic structure for the models to parse. For EEG, open eyes were designated as the "0" class and closed eyes were designated as the "1" class. See Table 1 for further details on each data set.

Table 1: The Datasets

Problem	**# Attributes**	**Train Size**	**Test Size**	**%Positive**

Avila	11	5,000	20,867	74.8%
EEG Eye	15	5,000	14980	44.9%
Occupancy	8	5,000	8,143	21.2%
Letters	17	5,000	20,000	49.7%

Performance By Metric

Similar to CNM06, over 5 trials we sampled 5,000 instances from each dataset and used 5-fold cross validation on each sample of 5,000. Models for the best performing hyperparameter sets for each algorithm per performance metric, per trial, and per dataset are fit to the whole 5,000 instance sample and used to predict the values of the entire dataset. The scores obtained from these predictions are averaged from the 5 trials and 4 datasets to produce the results seen in Table 2. The Mean column there holds the average score across each metric for the algorithm. The bolded terms represent the highest scoring algorithm for each performance metric and the asterisked terms would represent values that have a non-significant difference between them and the bolded term, however, there were no such values represented in either Table 2 or Table 3. These values would have a p value less than .05 when a t-test is applied to the array of elements whose average makes that value and the array of elements whose average makes the best scoring value in the column.

Across all three metrics, random forest outperforms logistic regression and decision tree, though decision tree similarly stands above logistic regression for all metrics.

Table 2: Scores for each Algorithm by Metric

Model	**Accuracy**	**F1-Score**	**ROC-AUC**	**Mean**
LOGREG	0.785	0.729	0.748	0.754
DECTREE	0.927	0.922	0.897	0.915
RF	**0.969**	**0.968**	**0.966**	**0.968**

Performance by Dataset

As stated above, the scores seen in Table 3 are derived from scores from the best performing hyperparameter sets fit to the 5,000 instance sample from their trial and dataset for each performance, and used to predict the entire dataset. These scores are then averaged over scores across trials and performance metrics to produce those seen in Table 3.

As can be seen below, random forest again outperforms decision tree and logistic regression across each set, and decision tree again outperforms logistic regression for each set as well. What is notable here, however, is how poorly logistic regression performed on EEG such that it would have performed *better* had it's output been inverted before scoring. This implies that EEG is likely not non-linearly separable as logistic regression creates a linear decision boundary and EEG represents an outlier in performance for logistic regression's scores. Conversely, decision tree and random forest do not produce linear decision boundaries and produce scores for it similar to their other results.

Table 3: Scores for Each Algorithm by Dataset

Model	AVILA	EEG	OCCU	LETTERS	Mean
LOGREG	0.840	0.465	0.983	0.727	0.754
Decision Tree	0.955	0.828	0.990	0.889	0.915
Random Forest	**0.981**	**0.932**	**0.996**	**0.962**	**0.968**

Conclusions

Random forest was among the strongest algorithms studied in CNM06 and that notion is solidified here as it was remarkably effective when compared to both logistic regression and decision tree models. The advantage that both random forest and decision tree algorithms have over linear models like logistic regression was also clear as they both performed consistently where logistic regression models could not obtain consistent scores in training sets or any generalizability in test sets. The tradeoff for random forest was also tangible as it's time complexity (with a forest of 1024 trees) meant that it could not support as much hyperparameter probing and nonetheless took much longer to generate fitted models and in turn produce results, a limitation neither Logistic Regression and Decision Trees experienced.

Appendix

Appendix Table 1: Train Scores for Each Algorithm by Dataset

Model	AVILA	EEG	OCCU	LETTERS	Mean
LOGREG	0.842	0.608	0.983	0.730	0.791
Decision Tree	0.977	0.916	0.992	0.949	0.959

Random Forest	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**

Discussion

Comparing the results from *Appendix Table 1* and *Table 2* we can plainly see that the models scored better on the training sets than the test sets across the board. This is further exemplified by random forest's perfect score on all of its training sets for all performance metrics. This is expected though because the model is fitted specifically to the training data so it makes sense why it might score better on it. When a model scores very well on its training model, however, that could be a sign of overfitting to the training data and a lack of generalizability which is why it's important to reserve validation/test data separate from the training data. Though random forest was likely somewhat overfit to the EEG training set and potentially the LETTERS training set when comparing the training set scores to the test set scores, it still performed exceptionally well across testing for both of those datasets, especially in comparison to the other two algorithms.

Appendix Table 2: Raw Test Set Scores

Dataset	A1	A2	A3	A4	A5	E1	E2	E3	E4	E5	O1	O2	O3	O4	O5	L1	L2	L3	L4	L5
LR ACC	.861	.862	.862	.862	.861	.569	.578	.550	.549	.577	.989	.988	.988	.988	.988	.725	.729	.725	.725	.726
LR ROC	.745	.747	.746	.748	.745	.520	.546	.503	.527	.559	.991	.987	.986	.987	.986	.725	.729	.726	.725	.726
LR F1	.913	.913	.914	.914	.913	.229	.356	.130	.385	.406	.973	.973	.973	.973	.973	.729	.734	.727	.724	.721
DT ACC	.972	.972	.972	.959	.972	.849	.839	.859	.814	.859	.996	.993	.994	.994	.996	.901	.894	.901	.904	.904
DT ROC	.923	.930	.919	.914	.898	.822	.814	.814	.803	.822	.989	.992	.987	.984	.989	.863	.870	.870	.865	.870
DT F1	.981	.981	.981	.972	.981	.830	.819	.842	.790	.842	.991	.983	.986	.986	.990	.898	.890	.898	.903	.903
RF ACC	.983	.983	.980	.983	.983	.937	.935	.935	.936	.937	.998	.998	.998	.998	.997	.961	.962	.962	.962	.962
RF ROC	.976	.971	.964	.976	.969	.934	.934	.934	.934	.934	.996	.997	.997	.997	.997	.962	.962	.962	.962	.962
RF F1	.989	.989	.987	.989	.989	.929	.927	.926	.928	.928	.995	.995	.994	.995	.994	.961	.963	.963	.963	.963

Appendix Table 3a: P-Values of Comparisons by metric

Performance Metric (by Algorithm)	**P-Value**
ACC (LOGREG)	9.484476701867612e-06
ROC_AUC (LOGREG)	1.7439128162376142e-06
F1 (LOGREG)	0.000386165792042603
MEAN (LOGREG)	3.699032723330679e-16
ACC (DECTREE)	8.291283065958167e-05
ROC_AUC (DECTREE)	9.47097243970075e-07
F1 (DECTREE)	0.00011958662733499019
MEAN (DECTREE)	3.318955134437652e-10

Appendix Table 3b: P-Values of Comparisons by dataset

Dataset (by Algorithm)	**P-Value**
AVILA (LOGREG)	8.219871082473357e-07
EEG (LOGREG)	1.9707318517189897e-09
OCCU (LOGREG)	5.014367108187927e-07
LETTERS (LOGREG)	1.915883757368666e-28
MEAN (LOGREG)	4.0715571937866547e-78
AVILA (DECTREE)	0.0005406344713840243
EEG (DECTREE)	7.720066362459828e-12
OCCU (DECTREE)	7.112154071582886e-06
LETTERS (DECTREE)	7.6757347712904e-11
MEAN (DECTREE)	1.0302230447189735e-05

References:

1. Caruana, Rich, and Alexandru Niculescu-Mizil. "An empirical comparison of supervised learning algorithms." *Proceedings of the 23rd international conference on Machine learning*. 2006.
2. Oliver Roesler (2013). “EEG Eye State Data Set.” https://archive.ics.uci.edu/ml/datasets/EEG+Eye+State. Irvine, CA: University of California, School of Information and Computer Science.
3. C. DeÂ Stefano, M. Maniaci, F. Fontanella, A. ScottoÂ diÂ Freca, Reliable writer identification in medieval manuscripts through page layout features: The 'Avila' Bible case, Engineering Applications of Artificial Intelligence, Volume 72, 2018, pp. 99-110. https://archive.ics.uci.edu/ml/datasets/Avila Irvine, CA: University of California, School of Information and Computer Science.
4. David J. Slate (1991) “Letter Recognition Data Set” Odesta Corporation; 1890 Maple Ave; Suite 115; Evanston, IL 60201 https://archive.ics.uci.edu/ml/datasets/Letter+Recognition. Irvine, CA: University of California, School of Information and Computer Science.
5. Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models. Luis M. Candanedo, VÃ©ronique Feldheim. Energy and Buildings. Volume 112, 15 January 2016, Pages 28-39. https://archive.ics.uci.edu/ml/datasets/Occupancy+Detection+. Irvine, CA: University of California, School of Information and Computer Science.
6. van Wyk, Andrich. “Encoding Cyclical Features for Deep Learning.” *Kaggle*, Kaggle, 14 Apr. 2018, www.kaggle.com/avanwyk/encoding-cyclical-features-for-deep-learning.
7. “Sklearn.ensemble.RandomForestClassifier¶.” Scikit-Learn, scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html .
8. “Sklearn.tree.DecisionTreeClassifier¶.” *Scikit-Learn*, scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html.
9. “Sklearn.linear_model.LogisticRegression¶.” Scikit, scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html#sklearn.linear_model.LogisticRegression.set_params.
10. “Sklearn.preprocessing.StandardScaler¶.” *Scikit*, scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html.
11. “Sklearn.model_selection.GridSearchCV¶.” *Scikit-Learn*, scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.
12. “Scipy.stats.ttest_rel¶.” *Scipy.stats.ttest_rel - SciPy v1.6.1 Reference Guide*, docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_rel.html.
13. Professor Jason Fleisher, Lecture_19_model_selection, (2021), UCSD_COGS118A/Notebooks, https://github.com/jasongfleischer/UCSD_COGS118A/blob/main/Notebooks/Lecture_19_model_selection.ipynb