COGS 118A: Comparison of Supervised Learning Algorithms: A Replication

COGS 118A

UC San Diego

Winter 2021

Prof. Jason Fleischer

Anwen He

A92147485

Abstract

This paper is a partial replication of the analysis done in the paper by Rich Caruana and Alexandru Niculescu-Mizil in 2006, referred to as CNM06. My version of analysis is conducted using 4 different datasets and they are used to assess the performance of 3 supervised algorithms including logistic regression, random forests and decision trees using various parameters.

Keywords: Binary Classification, Supervised Models, Logistic Regression, Random Forests, Decision Trees

1. Introduction

This paper serves as a report of an attempt to replicate the analysis done by Caruana and Niculescu-Mizil in 2006 in their paper *An Empirical Comparison of Supervised Learning Algorithms (CNM06)*. In order to provide a large scale comparison between some of the most commonly used supervised learning methods, Caruana and Niculescu-Mizil performed over several algorithms on 11 large datasets and checked their performance using a variety of performance criteria while tweaking hyperparameters for each algorithm. The way this assessment is conducted can be immensely helpful in acquiring an understanding of supervised learning algorithms. In our study, we will attempt to replicate CNM06 by using 3 algorithms on 4 large datasets. Noted, we are applying similar settings to each algorithm the way it is done in CNM06 to precisely find the best parameters that achieve maximum performance in each algorithm-and-dataset combination.

2. Method

2.1 Learning Algorithms

In our implementation, we explored three supervised learning algorithms.

We selected **Logistic Regression (LOGREG)** from the linear-based methods. In a similar convention as in CNM06, we trained the method using both unregularized and regularized parameters. The varying parameter C for regularization consists of 10 values: 0 for unregularized and regularized from 10^-4 to 10^4, vary by a factor of 10. We used L2 norm for the penalty parameter, saga as the main solver and put 5000 as the maximum number of iterations in case the algorithm does not converge within the limit.

We used **Random Forests** as a representative of ensemble methods. We applied parameters in the same manner as in CNM06 again, with 1024 as number of estimators/trees, and a varying list of value: 1, 2, 4, 6, 8, 12 as the maximum size of features. We did not include 16 or 20 from the original lists in CNM06, since some of our datasets have fewer than or equal to 16 features; a larger value of maximum feature could result in unwanted errors.

Last but not least, we picked **Decision Trees** as our tree method here. We used two lists of varying parameters, one being the splitting criterion, including gini or entropy; the other list is the max depth of trees, using the value of 1, 2, 3, 4 and 5.

2.2 Performance Metrics

The original CNM06 paper included eight performance metrics; in our adaptation, we will mainly use three metrics as time of execution is relatively limited while we still want to have a decent assessment of the algorithms we are comparing.

All three metrics range from 0 to 1 for clarity, where 0 is the baseline rating and 1 is the highest rating achievable. The first and primary metric is **Accuracy**, calculated by the ratio of sum of true positive cases and true negative cases to the total population. The second metric we use is **F1 Score**, calculated using the equation 2 * (Precision score * Recall score)/(Precision score + Recall score). The third metric we use is the **Area Under ROC Curve**, also known as **ROC AUC**, which provides a measure of performance over all possible classification thresholds, or separability between classes. We will also use the mean between all three metrics to judge the performance of each algorithm.

2.3 Datasets Description

We use 4 different datasets to assess the performance of our algorithms. All 4 can be found on the public UCI repository. Some information about these datasets can be found in Table 1 below.

The **ADULT** dataset contains information related to features used to determine whether a person's annual income is above 50K or not. The dataset originally has 14 features, but most of them are categorical; therefore, we decided to implement one-hot encoding on the dataset which results in 108 final attributes. Income above 50K is our positive class here; 24.1% of the labels are positive, makes our dataset a slightly unbalanced one.

The **COVTYPER** dataset contains information used to predict forest cover type from a number of wilderness area in northern Colorado. There are 54 attributes, most of which are already one-hot encoded when we acquired it from the repository. We used cover type 1 as our positive case, which consists of 36.5% of the dataset.

The 3rd and 4th datasets both came from the **LETTER** recognition dataset, which contains information related to the pixels that form a letter. Both datasets contain 16 feature attributes. Our positive case for **LETTER P1** is the label of letter 'O', and everything else is negative. This makes for an extremely unbalanced dataset, where only 3.8% of the cases are positive. The positive cases for **LETTER P2** is the label of all letters in an alphabetically order from 'A' to 'M'. After cleaning the dataset, we find that 45.7% of the cases are positive, indicating a relatively balanced one.

Table 1. Description of Datasets

NAME	#ATTRIBUTE	TRAIN SIZE	TEST SIZE	POS%
ADULT	14/108	5000	27561	24.1
COVTYPE	54	5000	576012	36.5
LETTER P1	16	5000	15000	3.8
LETTER P2	16	5000	15000	45.7

3. Experiment

In this study, we went through every single combination of dataset and algorithm for 5 separate trials. In each trial, 5000 cases are randomly sampled from the original dataset to be used as training data, while the rest of the dataset become the testing set. We conduct a grid search on these 5000 samples in a manner of 5-fold cross validation, and for each fold of data, we run all of the potential values in the hyperparameter list to find a model with optimal hyperparameters that has the best performance for each of our three performance metrics. We use this model to train all 5000 samples for one last time to make predictions based on our testing set, for which the final performance metrics can be assessed. After that, a trial is concluded.

In conclusion, a total of 3*4 = 12 combination of model and data were tested, and each combination went through 5 trials, each trained using a randomly sampled set. In Table 2 and 3, we rank the testing performance of each algorithm to dataset combo across all trials using either **Accuracy** as the only metric, or the mean of **Accuracy, F1** and **AUC**. The best performance in each column is presented with a bold font. The (!) sign is attached next to a combination if it performed significantly worse to its peers, mainly comparing to the best performer. An asterisk is attached to a combination if it is not performing on a statistically significantly different comparing to the best performer, based on results from the two-sampled t-tests.

Table 2. Test Set Performance Across Trials for Each Algo/Data Combo (Accuracy)

MODEL\DATA	**ADULT**	**COVTYPE**	**LETTER P1**	**LETTER P2**	**MEAN**
LOGREG	0.848*	0.769	0.962	0.735	0.829
RF	**0.851**	**0.837**	**0.988**	**0.951**	**0.907**
DT	0.846*	0.757	0.971	0.771	0.837

Table 3. Test Set Performance Across Trials for Each Algo/Data Combo (Mean over 3 Metrics)

MODEL\DATA	**ADULT**	**COVTYPE**	**LETTER P1**	**LETTER P2**	**MEAN**
LOGREG	0.752	0.730	0.488(!)	0.724	0.674
RF	**0.761**	**0.810**	**0.876**	**0.949**	**0.849**
DT	0.759*	0.753	0.833*	0.817	0.791

One finding is unsurprisingly congruent with the results of CNM06: across all algorithms, **Random Forests** has the best average performance in every metric. **Logistic Regression** performs evenly inferior in all scenarios that we are testing. Although the results in CNM06 suggest that when running on certain datasets, **Logistic Regression** can be one of the best performing models, we cannot conclude the same finding using the results of our study.

We should look through the performance metrics separately to conduct more findings. In Table 4 and 5 below, we found that **Random Forests** achieved a perfect score in all three metrics on our training set, and while their performance all dropped when used to make predictions on the testing set, they still maintained a relatively good performance overall. The **Decision Trees** algorithm showed a boost in performance when ran on the **LETTER** datasets, which could have some correlation to how the data is structured differently compared to the other 2 datasets in our study (perhaps more uniform, integer-based attributes). The **F1** score and **AUC** metrics both failed to assess the combination of **Logistic Regression** and **LETTER P1** dataset, presumably because it is the most unbalanced dataset in our study, with an extremely small number of positive cases. This likely prevented the metric to acquire a functional precision score which is needed to compute an F1 score. The **AUC** metric also dropped significantly on this combination. What we can learn from this result is that we might want to use more performance metrics to assess our algorithms in a more prepared manner when time is sufficient, in case that similar datasets prevent us from having a well-rounded understanding of the algorithm.

Table 4.Mean Test Set Performance Across Trials for Each Algo/Data Combo (Separate Metrics)

Algo/Data Combo	**ACC**	**F1**	**ROC AUC**
LR_ADULT	0.848	0.649	0.758
LR_COVTYPE	0.769	0.675	0.746
LR_LETTER_P1	0.963	0(!)	0.5(!)
LR_LETTER_P2	0.735	0.706	0.732
RF_ADULT	0.851	0.664	0.769
RF_COVTYPE	0.837	0.772	0.821
RF_LETTER_P1	**0.988**	0.813	0.827
RF_LETTER_P2	0.951	**0.946**	0.950
DT_ADULT	0.846	0.632	0.799
DT_COVTYPE	0.757	0.683	0.819
DT_LETTER_P1	0.973	0.634	0.892
DT_LETTER_P2	0.772	0.718	**0.963**

Table 5. Train Set Performance Across Trials for Each Algo/Data Combo (Mean over 3 Metrics)

MODEL\DATA	**ADULT**	**COVTYPE**	**LETTER P1**	**LETTER P2**	**MEAN**
LOGREG	0.766	0.735	0.487(!)	0.722	0.678
RF	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
DT	0.748	0.750	0.769	0.756	0.756

4. Conclusion

Overall, our findings in this study do show some resemblance to the original study – CNM06 by Caruana and Niculescu-Mizil – that we are trying to replicate. It is seen that **Random Forests** have the best performance across all datasets through our metrics. **Logistic Regression** and **Decision Trees** perform on a similar level, both worse than the **Random Forests** algorithm, although **Decision Trees** seems to perform slightly better on certain datasets and are less affected in terms of performance by an extremely unbalanced sample, unlike **Logistic Regression** which had its performance affected severely.

Across all four datasets, there is some significant variability in terms of performance. It is difficult to tell if there is one dataset that all three algorithms performed the worst on.

5. Reference

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

Malik, Usman. *Random Forest Algorithm with Python and Sckit-Learn* https://stackabuse.com/random-forest-algorithm-with-python-and-scikit-learn/

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.

Appendix

Table 6. Raw Training Data Performance Data (Accuracy) by Trial

COMBO\TRIAL	**1**	**2**	**3**	**4**	**5**
ADULT_LR	0.864	0.854	0.852	0.857	0.857
COV_LR	0.763	0.774	0.775	0.773	0.783
L_P1_LR	0.961	0.960	0.963	0.960	0.958
L_P2_LR	0.735	0.732	0.738	0.731	0.728
ADULT_RF	1.0	1.0	1.0	1.0	1.0
COV_RF	1.0	1.0	1.0	1.0	1.0
L_P1_RF	1.0	1.0	1.0	1.0	1.0
L_P2_RF	1.0	1.0	1.0	1.0	1.0
ADULT_DT	0.861	0.853	0.852	0.845	0.853
COV_DT	0.780	0.783	0.774	0.782	0.756
L_P1_DT	0.975	0.980	0.978	0.977	0.971
L_P2_DT	0.793	0.774	0.777	0.771	0.778

Table 7. Raw Testing Data Performance Data (Accuracy) by Trial

COMBO\TRIAL	**1**	**2**	**3**	**4**	**5**
ADULT_LR	0.846	0.850	0.848	0.847	0.848
COV_LR	0.768	0.771	0.771	0.766	0.768
L_P1_LR	0.963	0.963	0.962	0.963	0.964
L_P2_LR	0.735	0.737	0.734	0.732	0.735
ADULT_RF	0.852	0.848	0.852	0.851	0.850
COV_RF	0.837	0.836	0.837	0.837	0.838

L_P1_RF	0.988	0.988	0.988	0.987	0.989
L_P2_RF	0.948	0.952	0.950	0.955	0.951
ADULT_DT	0.842	0.844	0.846	0.845	0.854
COV_DT	0.757	0.765	0.755	0.762	0.749
L_P1_DT	0.971	0.974	0.973	0.975	0.964
L_P2_DT	0.791	0.765	0.758	0.770	0.775

Table 7. P-values for Table 2

MODEL\DATA	**ADULT**	**COVTYPE**	**LETTER**	**LETTER**	**MEAN**
			P1	**P2**	
LR	0.074	2.533e-07	3.530e-07	1.724e-08	0.001
DT	0.085	1.248e-05	0.001	7.577e-06	0.0003

Table 8. P-values for Table 3

MODEL\DATA	**ADULT**	**COVTYPE**	**LETTER**	**LETTER**	**MEAN**
			P1	**P2**	
LR	0.0007	9.620e-13	0.0005	1.54e-19	6.310e-08
DT	0.745	0.0001	0.144	0.0003	6.566e-06