Comparison of Supervised Learning Algorithms

Garvin Mo Zhen GMOZHEN@UCSD.EDU
COGS 118A: Supervised Machine Learning Algorithms
University of California, San Diego

Abstract

This paper describes the replication of Caruana and Niculescu-Mizil's paper on a smaller scale, comparing the performance of three supervised learning algorithms across four different variety of data sets. The three algorithms used are random forests, k-nearest neighbors, and logistic regression. Algorithms are chosen based on bootstrap analysis ranking done by Caruana's paper. The three algorithms consisted one of each from the top-ranked, middle-ranked, and bottom-ranked algorithms. Further on, algorithms will be tuned through gridsearch over multiple trials to yield the best performing hyper-parameters. Performance was measured using three metrics: accuracy, ROC Area and F-score. Each score possibly yields unique hyper parameters, which are then used to recalculate the performance of each score on the testing set, and averaged out over five trials, and four data sets. Caruana and Niculescu-Mizil concluded from their paper that in all cases random forests, bagged trees, boosted decision trees will outperform the other learning algorithms in the bootstrap analysis. Meanwhile, logistic regression and decision trees would be ranked near last.

Keywords: Random Forests, k-Nearest Neighbors, Logistic Regression

1. Introduction

Prior to Caruana and Niculescu-Mizil (CNM6), there existed a few comprehensive empirical studies in which learning algorithms were compared. At the time, STATLOG 1995, was the best known study that was very comprehensive. However, since then newer and better performing algorithms arose that were not used in STATLOG. Newer algorithms such as bagging trees, boosting, SVM, random forests, and many more [1]. Additionally, these algorithms that emerged recently have very good classification power and performance.

In response to these algorithms, CNM6 set forth an entensive empiricla evaluation of these modern machine learning algorithms. In their experiment, CNM6 evaluated the performance of ten different types algorithms: SVMs, neural nets, logistic regression, naive bayes, memory-based learning, random forests, decision trees, bagged trees, boosted trees, and boosted stumps [1]. In each of the eleven problem used, they explored a wide variety of parameters for each algortithm, and calibrated its predictions with Platt Scalling and Isotonic Regression. Additionally, they used a variety of performance metrics to evaluate each algorithm such as accuracy, F-score, Lift, ROC, Area, average precision, precision/recall break-even point, squared error, and cross entropy [1].

CNM6 concluded from their evaluation that calibration with either Platt's method or Isotonic Regression dramatically improved the performance of some algorithms such as booted trees, SVMs, boosted stumps and Naives Bayes [1]. However, for other algorithms either slight improvement or no improvement was noticed. Overall, Boosted trees was the best learning algorithm overall, and random forests the second best. However, the logistic regression, decision trees, boosted stumps, and naive bayes performed the poorest

among all other algorithms. However, they argued that depending on the problem, even the best learning algorithms can perform poorly and average algorithms can perform extremely well. This paper will attempt to recreate CNM6's evaluation of learning algorithms on a smaller scale using three drastically different performing algorithms based on the CNM6's paper. Random forests, KNN and logistic regression will be used. These algorithms range from the best performing to the worst performing. Unlike CNM6, any calibration will not be implemented since it could drastically improve some algorithms. The performances will be scored on three metrics: accuracy, ROC Area, and F-score. Using four data sets, two which were originally used (ADULT and LETTER.0) and two others that were not used in CNM6 (MUSHROOM and WEATHER). With the addition of unevaluated data sets, will help conclude whether if average and bottom-tier algorithms can potetially outperform top-rated performing learning algorithms.

2. Methology

2.1 Learning Algorithms

For each algorithms, most parameters are left default with the scikit-learn library, while exploring different sets of hyper-parameters that differ slightly from CNM6's description. These parameters are tuned in a grid search using 5 fold cross validation for five trials.

k-Nearest Neighbors (KNN): explored 30 different values ok k, which are evenly spaced from a range between 1 and 500. Different weights used are uniform and distance.

Logistic Regression (LR): trained on both unregularized and regularized models, with varying C-values or regularization parameter by factors of 10 from 10^{-8} to 10^{4}.

Random Forests (RF): the numbers of trees in forest trained consisted of 1024 trees and the maximum size of the feature set to be considered at each split consisted of: 1, 2, 4, 6, 8, 12, 16, or 20.

With these choices of algorithms and CNM6's findings, we should expect that random forest should outperform the other two learning algorithms.

2.2 Performance Metrics

The performance metrics that will be used are accuracy (ACC), F-score (FSC), and area under ROC curve (ROC). All these metrics range from 0 to 1, where 0 is the worst and 1 is the best. As a result, we do not need to worry about comparison across metrics, since they are scaled equally.

Accuracy (ACC) computes the percentage of correct predictions, which can be calculated [4]:

$$ACC = \frac{1}{n_{samples}} \sum 1(y_{pred} = y)$$

F-score (FSC) can be interpreted as the weighted average of the precision and recall [5]. This metric can be calculated with the following equation:

$$F1 = \frac{2 * (precision * recall)}{precision + recall}$$

Note: There can be cases such that (true positive + false positive = 0) or (true positive + false negative = 0), in these cases, precision or recall will be undefined, which will give an F-score of 0.

ROC: is a comparison of the true positive rate versus the false positive rate, in other word sensitivity vs. (1 - specificity).

2.3 Data Sets

For the experiment, we used in total 4 different data sets, 2 of the data sets were from the original experiment: ADULT and LETTER.O. Both of these are sourced from the UCI Machine Learning repository. The other two new data sets are MUSHROOM which was also sourced from the UCI repository, and WEATHER which was available online at Kaggle. All data sets contained a binary classification problem.

ADULT binary labels that classified whether adult incomes were greater than 50k or less than equal to 50k. This data set contained both categorical and numerical data. As a result, we had to one hot encode the categorical data and normalize numerical data. Similar process was done for the other data sets. This data set was also imbalanced with only about 7k adults whose income is greater than 50k, while the rest are less than 50k, about 24% positive rate.

LETTER.O same data set used in CNM6 also, only used the case where a letter was classified as an "O" or not. Contained all numerical data, so we did not have to manipulate the data. This data set was also very unbalanced, 753 that are the letter "O", and 19k that are not. This gives us about a 4% postive rate.

MUSHROOM data set was given all categorical features, and in this case we had to classify whether a mushroom was edible or poisonous. This data set was a smaller one among the others, with only 8k data points. Unlike the other data sets, this one is more balanced, with a 52% positive rate.

WEATHER data set was to predict whether it would rain the next day or not given a set of features. This data set contained both categorical and numerical data. Additionally, it also contained the date in the format year-month-day. One-hot-encoding the date feature would create thousands of rows, so a solution was to encode each part separately year, month and day [2]. Additionally, this data set was the largest with 150k data points but contained a lot of missing data, which resulted it to be shrunken down to about 56k data points. Data set was also unbalanced, with a 22% positive rate.
See Table 1 for characteristics of these problems.

Table 1: Description of problems

PROBLEM	#ATTR	TRAIN SIZE	TEST SIZE	%POS
ADULT	14/106	5000	27560	24%
LETTERS.O	16	5000	15000	4%
MUSHROOM	22/117	5000	3123	52%
WEATHER	24/95	5000	51420	22%

3. Experiments & Results

Combining each data set and algorithm, we would randomly sample 5000 data points from the data set for each algorithm independently. Then through a 5 fold cross validation grid search, we will get **three** sets of optimal hyper parameters for every performance metric we are using. Further on, we train our top models once more on the whole training set. Lastly, we would be scoring each model on the entire untouched test set based on the metric that gave us that model. These same steps would be repeated for 5 total trials for each algorithm and data set.

Main results can be seen on Tables 2 and 3. More results can also be seen on the appendix on Tables 4, 5 and 6 to see training set performance, raw trials and statistical testing. In the following tables, algorithms that peform the best in each metric/data set will be **boldfaced**. We performed two independent t-tests on all trials since samples were not consistent throughout algorithms. We set a p-value threshold of $p = 0.05$ and algorithms whose performance is not statistically distinguishable will be marked with an asterisk(*) [1]. In other words, a p-value that is greater than 0.05. The entries that are left unmarked indicate that the algorithm's performed significantly lower than the best model in the specific metric/data set.

3.1 Performance by Metric

Table 2 shows the scores for each algorithm on each of the three metrics used normalized across all four problems presents. Additionally, for the last column, MEAN, is the average score of all three metrics on all four problem sets for the given learning algorithm.

Averaging across all three metrics used, we can see that random forests learning algorithm just barely outperformed KNN but outperformed logictic regression by nearly 10% in performance. Overall, in scoring all three algorithms in accuracy (ACC) and ROC area (ROC), random forests just barely outperformed the other two, where even logistic regression was able to keep up in performance. However, using a third metric, we can see that logistic regression greatly unperformed compared to the other two. See Table 2 below for exact values.

Table 2: Test set performance for each algorithm by metric (average over 4 problems)

MODEL	ACC	FSC	ROC	MEAN
RF	**0.923**	**0.771**	**0.945**	**0.880**
KNN	0.908*	0.731*	0.930*	0.857
LR	0.916*	0.569	0.912*	0.799

3.2 Performance by Problem

Table 3 shows the normalized performance of all metrics for each problem presented individually, whereas the MEAN column remains constant from the previous table.

On the ADULT data set problem, as expected, random forest performed the best overall. However, the other two are also marked with an *, which means that their performances did not differ much from random forests. This data set contained an unbalanced amount

of positive and negative values. Additionally, it also had a relatively big testing set. Surprisingly, all three algorithms performed relatively well.

LETTER.O was the most unbalanced data set of all, with only about 4% positive labels. On this unbalanced data set, k-Nearest Neighbors performed the best, with random forests coming in a close second. Meanwhile, logistic regression significantly under performed on this data set compared to the rest of the algorithms and data sets. This is due to the F-score resulting to 0, as seem on Table 5.2.1 on the Appendix. The case for this is when precision or recall are undefined because there can be cases where we are dividing by 0 [6]. As a result, the normalized performance on logistic regression remained lower than the rest.

The MUSHROOM data set as seen, performed the best out of all other data sets where every algorithm performed equally. This might be due to the fact that the data set was the most balanced out of all the rest. Additionally, this data set also contained the least amount of data points, 8k. Which means that the testing set was about a size of 3k. Not only that, our models would be trained on a training set that was bigger than the test set. As a result, it might have been to easy for the algorithms, where it does not allow us to compare the algorithms performances.

Lastly, the WEATHER data set was a more representative data set that allowed us to compare the algorithms. We can see that using a data set that was not used in this study can has drastically different results. For this data set, we can see that all three algorithms has slightly below average performances. Surprisingly, this is the only data set where logistic regression outperforms better than the other two. However, the other learning algorithms did not perform significantly worse.

Overall, random forests still remained on top of the other two algorithms performance wise. Even when it was outperformed, it was usually just slightly worse than the best for the data set. See Table 3 for scores.

Table 3: Test set performance on each problem (average over 3 metrics)

MODEL	ADULT	LETTER.O	MUSHROOM	WEATHER	MEAN
RF	**0.805**	0.933*	**1.000***	0.782*	**0.880**
KNN	0.785*	**0.956**	**1.000***	0.685*	0.857
LR	0.803*	0.607	**1.000***	**0.786**	0.799

4. Conclusion

On a small scale attempt to recreate Caruana and Niculescu-Mizil's study, we were able to confirm that random forests does indeed rank above KNN and logistic regression. With KNN coming in a close second, and unfortunately, logistic regression still remained last. Our findings were stayed consisted with CNM6, despite the reduced number of problems and metrics. We were able to conclude that the algorithms on average are ranked just as in CNM6, RF > KNN > LR.

However, using 2 complete different data sets that were not used in CNM6, MUSHROOM AND WEATHER. We can infer that, small and balanced data sets such as MUSHROOM, learning algorithms have great performances. This is still not clear given such a small testing set. Not only that, we can see that WEATHER provides a bigger testing set

but with a somewhat unbalanced data. In this case, all algorithms performed mediocre, and even the poor performing algorithms such as logistic regression can outperform the better performing algorithms depending on the specific problem we are looking to solve.

Overall, we can still see that CNM6's findings that even the best algorithms can under perform while average ones can have excellent performance. With the lack of computational power and time, this paper was still capable of replicating CNM6. As even newer and better algorithms emerge, comparisons between those algorithms will be needed. In the near future, this paper can be expanded with more algorithms and more data sets.

Acknowledgments

I would like to acknowledge professor Jason G. Fleisher for giving so much time and help. Since the beginning of the quarter preparing us for this project, with lectures and assignments. And not only that, for having offices hours almost every single day for this past finals week. I would also like to acknowledge my TA Yifan Xu, for all the helpful discussion sections that help me with my assignments and allowed me to succeed in this project. Lastly, I would like to acknowledge my classmates for helping me understand concepts I struggle with on Piazza.

References

[1] 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`

[2] Long, Andrew *Machine Learning with Datetime Feature Engineering: Predicting Healthcare Appointment No-Shows*. Towards Data Science.
`https://towardsdatascience.com/machine-learning-with-datetime`

[3] Can the F1 score be equal to zero?, Stack Exchange
`https://datascience.stackexchange.com/questions/72074/...`

[4] Accuracy score, scikit-learn
`https://scikit-learn.org/stable/modules/model_evaluation.html#accuracy-score`

[5] f1_score, scikit-learn
`https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html`

[6] Receiver operating characteristics (ROC), scikit-learn
`https://scikit-learn.org/stable/modules/model_evaluation.html#roc-metrics`

Appendix

TRAINING SET PERFORMANCE:

Table 4: Training set performance for each algorithm by metric (average over 4 problems)

MODEL	ACC	FSC	ROC	MEAN
RF	1.000	1.000	1.000	1.000
KNN	0.966	0.953	0.994	0.971
LR	0.919	0.641	0.917	0.826

RAW TRIAL SCORES FOR EACH METRIC AND DATASETS:

Table 5.1.1: Test set raw trial accuracy(ACC) on ADULT and LETTER

MODEL	1ADL	2ADL	3ADL	4ADL	5ADL	1LET	2LET	3LET	4LET	5LET
RF	0.851	0.853	0.850	0.852	0.851	0.988	0.988	0.988	0.986	0.989
KNN	0.838	0.837	0.837	0.833	0.836	0.992	0.990	0.991	0.992	0.991
LR	0.847	0.850	0.850	0.847	0.849	0.962	0.963	0.962	0.962	0.962

Table 5.1.2: Test set raw trial accuracy(ACC) on MUSHROOM and WEATHER

MODEL	1MSH	2MSH	3MSH	4MSH	5MSH	1WTH	2WTH	3WTH	4WTH	5WTH
RF	1.000	1.000	1.000	1.000	1.000	0.854	0.854	0.854	0.854	0.853
KNN	1.000	1.000	1.000	1.000	1.000	0.807	0.808	0.806	0.805	0.807
LR	1.000	1.000	1.000	1.000	1.000	0.854	0.853	0.854	0.853	0.853

Table 5.2.1: Test set raw trial F-score(FSC) on ADULT and LETTER

MODEL	1ADL	2ADL	3ADL	4ADL	5ADL	1LET	2LET	3LET	4LET	5LET
RF	0.657	0.669	0.658	0.664	0.661	0.815	0.817	0.819	0.791	0.829
KNN	0.635	0.628	0.617	0.633	0.633	0.896	0.866	0.880	0.894	0.880
LR	0.657	0.662	0.661	0.649	0.651	0.000	0.000	0.000	0.000	0.000

Table 5.2.2: Test set raw trial F-score(FSC) on MUSHROOM and WEATHER

MODEL	1MSH	2MSH	3MSH	4MSH	5MSH	1WTH	2WTH	3WTH	4WTH	5WTH
RF	1.000	1.000	1.000	1.000	1.000	0.610	0.610	0.618	0.599	0.611
KNN	1.000	1.000	1.000	1.000	1.000	0.416	0.417	0.406	0.410	0.416
LR	1.000	1.000	1.000	1.000	1.000	0.625	0.621	0.621	0.617	0.623

Table 5.3.1: Test set raw trial ROC area on ADULT and LETTER

MODEL	1ADL	2ADL	3ADL	4ADL	5ADL	1LET	2LET	3LET	4LET	5LET
RF	0.903	0.901	0.902	0.901	0.902	0.997	0.995	0.997	0.995	0.998
KNN	0.890	0.888	0.890	0.888	0.888	0.991	0.995	0.995	0.994	0.995
LR	0.904	0.904	0.904	0.905	0.905	0.858	0.860	0.857	0.862	0.853

Table 5.3.2: Test set raw trial ROC area on MUSHROOM and WEATHER

MODEL	1MSH	2MSH	3MSH	4MSH	5MSH	1WTH	2WTH	3WTH	4WTH	5WTH
RF	1.000	1.000	1.000	1.000	1.000	0.883	0.883	0.879	0.883	0.881
KNN	1.000	1.000	1.000	1.000	1.000	0.840	0.832	0.840	0.841	0.831
LR	1.000	1.000	1.000	1.000	1.000	0.884	0.885	0.886	0.885	0.884

P-VALUES FOR TABLE 2 & 3:

Table 6.1: P-value for Table 2 (RF compared with KNN/LR)

MODEL/METRIC	p-val KNN	p-val LR
RF/ACC	0.573	0.754
RF/FSC	0.527	0.030
RF/ROC	0.457	0.064

Table 6.2: P-value for Table 3 (RF compared other datasets)

DATASET(MODEL)	p-val KNN
ADULT(KNN)	0.621
ADULT(LR)	0.527
LETTERS.O(KNN)	0.385
LETTERS.0(LR)	0.010
MUSHROOM(KNN)	NO DIFF
MUSHROOM(LR)	NO DIFF
WEATHER(LR)	0.126
WEATHER(LR)	0.916

CODE USED IN PAPER CAN BE SEEN HERE:
https://github.com/garvingit/supervised_algorithm_final