A Comparison of Supervised Learning Algorithms on Binary Classification Accuracy

Enoch Li
Department of Cognitive Science
University of California San Diego
La Jolla, CA 92122 USA
e2li@ucsd.edu

Abstract

Previous research of supervised machine learning algorithms have looked at a comprehension comparison of these models on multiple performance metrics. This report presents a comparison between three supervised machine learning models: k nearest neighbors, logistic regression, and random forests. An accuracy score is used for evaluating each learning model. It is important to note that this paper attempts to replicate a portion of a previously published analysis.

All source code for this paper can be found on GitHub.

1 Introduction

Machine learning algorithms are widely used both in academia and industry, with libraries such as `scikit-learn` [4] optimizing the implementation of these algorithms while also providing tools for model fitting, data preprocessing, and evaluation. While these algorithms have become easier to use, it is still important to evaluate the performance made by these classifiers, as these libraries do not provide the performance metrics on how optimal or sub-optimal a classifier may be.

The goal of this analysis is to evaluate the accuracy performance on three binary classification problems using three supervised machine learning algorithms: K Nearest Neighbors, Logistic Regression, and Random Forests. Accuracy is measured as the number of correctly predicted entries in the testing set. Each algorithm was implemented using the `scikit-learn` library along with helper tools to evaluate the cross-validation and scoring metrics. The intent of this paper is to perform similar analysis done in "An Empirical Comparison of Supervised Learning Algorithms" [1]. Thus, the methodology of this paper will closely resemble that of Caruana, R., & Niculescu-Mizil, A. (2006), henceforth referred to as "Caruana".

2 Methods

Caruana presents a comparison of ten supervised machine learning algorithms on eight performance metrics. However, this paper will only detail the implementation and parameters used for each of the three algorithms on the accuracy performance metric.

2.1 K Nearest Neighbors (KNN)

KNN was implemented using `KNeighborsClassifier` from the `scikit-learn` library using distance weighting. Caruana used 26 values of K ranging from K = 1 to K = |*trainset*|. For this report, only 25 values of K were used, evenly log-spaced ranging from K = 1 to K = 500. All other parameters for **KNN** were set to the default values provided by `KNeighborsClassifier`.

2.2 Logistic Regression (LOGIT)

LOGIT was implemented using `LogisticRegression` from the `scikit-learn` library using both unregularized and regularized models. Unregularized models had no penalty, while the regularized models used l1 and l2 penalties. Similar to Caruana, The regularization parameters were varied by factors of 10 from 10^{-8} to 10^{4}, including regularization parameter = 0, yielding 14 total parameter settings. The `'newton-cg'` and `'saga'` solvers were also used. All other parameters for **LOGIT** were set to the default values provided by `LogisticRegression`.

2.3 Random Forests (RF)

RF was implemented using `RandomForestClassifier` from the `scikit-learn` library using a total of 1024 trees. Similar to Caruana, the size of the feature set for each split are as follows: 1, 2, 4, 6, 8, 12, 16, and 20. All other parameters for **RF** were set to the default values provided by `RandomForestClassifier`.

3 Data Sets

Each of the algorithms were compared on three binary classification problems using data sets from the UCI Repository [2]. Because the data sets contain categorical information, they have been converted to continuous values by transforming each unique attribute into binary values via one-hot encoding (one binary value per unique attribute). The methods for conversion to binary problems are found in Caruana.

3.1 Adult

The **Adult** data set consists of 32561 entries and 14 attributes, with 2399 of these entries containing missing values. Of these 14 attributes, 8 contained categorical values. After transforming the **Adult** data set to a continuous attributes and dropping the missing entries, it consists of 30162 entries and 104 attributes. **Adult** was converted to a binary problem by treating income values $\leq$50K as negative and >50K as positive.

3.2 Covertype (Cover)

The **Cover** data set consists of 581012 entries and 54 attributes, with 0 of these entries containing missing values. Of these 54 attributes, 0 contained categorical values. **Cover** was converted to a binary problem by treating the most common cover type as positive and the remainder as negative.

3.3 Letter Recognition (Letter)

The **Letter** data set consists of 20000 entries and 16 attributes, with 0 of these entries containing missing values. Of these 16 attributes, 0 contained categorical values. **Letter** was converted to a binary problem in two ways, unbalanced (**Letter**$_{p1}$) and balanced (**Letter**$_{p2}$). **Letter**$_{p1}$ treats the letter "O" as positive and the remainder of the letters as negative, while **Letter**$_{p2}$ takes letters "A-M" as positive and the rest of the letters as negative.

4 Experiment

For each algorithm, a total of 3 trials were run on each data set. For each trial, 5000 samples were randomly selected as the training set with the remainder set aside as a testing set. Using `GridSearchCV` from the `scikit-learn` library, a 5-fold cross validation was used on the training set with the specifications mentioned in Section 2. Each algorithm is trained on 4000 samples (4 folds) and evaluated on the final 1000 samples (1 fold) to determine the best parameters. The best classifier is trained again using the entire training data before being tested using the testing set.

Since this paper is focused on accuracy, only the accuracy performance metric will be reported for each classifier. That is, the number of correctly predicted entries out of the total entries in a given set. For this analysis, `accuracy_score` from the `scikit-learn` library was used to calculate accuracy.

Following the reporting from Caruana, the algorithm with the best accuracy performance has been **bolded**. Algorithms with performances that are not significantly different to the best score using an uncorrected two sample t-tests with p = 0.05 are annotated with an * symbol. Algorithm performances that are unmarked indicate an accuracy score that is *significantly lower* than the best algorithm performance.

Table 1: Mean testing accuracy score between algorithms by data set

Algorithm	Adult	Cover	Letter$_{p1}$	Letter$_{p2}$
KNN	0.8258	0.7807	**0.9911**	**0.9568**
LOGIT	0.8453*	0.7567	0.9622	0.7265
RF	**0.8460**	**0.8215**	0.9874*	0.9455

Table 1 describes a comparison of mean testing accuracy scores between algorithms for each data set. (See Table A for raw accuracy scores.) The p-values between algorithms by data set are found in Table B.

Table 2: Mean testing accuracy score between algorithms

Algorithm	Accuracy
KNN	0.8886*
LOGIT	0.8227
RF	**0.9001**

Table 2 describes a comparison of mean testing accuracy scores between algorithms. The p-values between algorithms are found in Table C.

Table 3: Mean training accuracy score between algorithms by data set

Algorithm	Adult	Cover	Letter$_{p1}$	Letter$_{p2}$
KNN	1.0000	1.0000	1.0000	1.0000
LOGIT	0.8498	0.7611	0.9628	0.7309
RF	1.0000	1.0000	1.0000	1.0000

Table 3 describes a comparison of mean training accuracy scores between algorithms for each data set.

5 Results

From Table 1, it appears **KNN** performs the best on both the **Letter**$_{p1}$ and **Letter**$_{p2}$ data sets, while **RF** performed the best on the **Adult** and **Cover** data sets. On the other hand, **LOGIT** did not perform the best for any of the data sets.

Because there are multiple algorithms that performed the best, there does not appear to be a best algorithm over all of the data sets. This conclusion is reflected in Table 2 since, although **RF** had the best mean accuracy score, **KNN** 's accuracy score did not differ significantly. These results are also consistent with the results from Caruana, both for accuracy scores over each data set and overall accuracy scores.

One metric of importance is the accuracy comparison for each of these algorithms on unbalanced (**Letter**$_{p1}$) data versus balanced (**Letter**$_{p2}$) data. Table 1 shows that all three algorithms have excellent

accuracy with the unbalanced data. However, the accuracy scores for all three algorithms decrease when comparing on balanced data, especially for **LOGIT** .

A point of discussion to note is the differences between the training and testing accuracy performance for each of the algorithms used. Table 3 shows that both **KNN** and **RF** had an accuracy score of 1 over all the data sets. This is an indication that these algorithms are most likely overfitting the data they were trained on, which may have lead to the higher accuracy performances on the testing data that were recorded. **KNN** and **RF** are known to overfit data during training, but future analysis could be beneficial to determine how influencial overfitting the training data has on testing performance of these algorithms.

It is also interesting to note that the **LOGIT** training and testing accuracy performances were nearly identical (less than 0.5% difference). This suggests that the testing data set was a representation of the training set. A future analysis could do more trials using **LOGIT** to see if these results are due to random noise or if the data representation does not affect **LOGIT** heavily.

6 Conclusion

It is clear that the applications of machine learning are endless and the field continues to grow and become easier to use. The simplicity of running and implementing these machine learning models was astounding, producing results that were comparable to other research results. This paper demonstrated that both **RF** and **KNN** had the best performance, with **RF** doing insignificantly better, at the cost of overfitting. And while **LOGIT** performed the worst, it was apparent that even poor models can perform well in certain areas. With more computational resources and time, this paper could be extended to explore the performance of other supervised machine learning models, as well as understanding in what situations would one model would perform better than another.

References

[1] Caruana, R., & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning* (pp. 161-168).

[2] 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.

[3] Fleischer, J. (2020). COGS 118A. Supervised Machine Learning Algorithms. University of California San Diego, La Jolla, CA.

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

Appendix A

Table A: Raw test set scores between algorithms by data set

Data Set (Trial #)	KNN	LOGIT	RF
Adult 1	0.8299	0.8461	0.8478
Adult 2	0.8224	0.8454	0.8441
Adult 3	0.8251	0.8445	0.8461
Cover 1	0.7835	0.7548	0.8213
Cover 2	0.7766	0.7540	0.8168
Cover 3	0.7819	0.7613	0.8264
Letter$_{p1}$ 1	0.9911	0.9613	0.9853
Letter$_{p1}$ 2	0.9913	0.9625	0.9893
Letter$_{p1}$ 3	0.9909	0.9629	0.9877
Letter$_{p2}$ 1	0.9553	0.7265	0.9436
Letter$_{p2}$ 2	0.9567	0.7259	0.9499
Letter$_{p2}$ 3	0.9583	0.7270	0.9431

Table B: P-value between algorithms by data set

Algorithms	Adult	Cover	Letter$_{p1}$	Letter$_{p2}$
KNN / LOGIT	0.0100	0.0102	0.0003	0.0000
LOGIT /RF	0.5679	0.0003	0.0011	0.0001
RF / KNN	0.0033	0.0023	0.0822	0.0440

Table C: P-value between algorithms

Algorithms	P-value
KNN / LOGIT	0.0452
LOGIT / RF	0.0116
RF / KNN	0.0911