An Empirical Comparison of Supervised Learning Algorithms for Classification

Dillon Ford, PID: A16092047

***Abstract*—This paper examines multiple supervised machine learning algorithms used to solve classification problems. Here we perform an empirical analysis and comparison between the supervised learning methods: logistic regression, k-nearest neighbors, random forests, and decision trees. Performance is measured across multiple trials and datasets for each classifier.**

I. Introduction

For a classification problem, finding the appropriate classifier with optimal parameters is crucial for success. Many of the ideas for this paper stem from prior work conducted to introduce methodologies and practices in classification [1] that optimized and empirically analyzed the performance of classifiers over many datasets. This paper will examine four different classifiers' performance in three trials across three datasets to solve problems relating to supervised machine learning algorithms. The Mushroom, Heart Disease, and Drug Consumption datasets are obtainable from the UCI Machine Learning Repository [2].

A note on datasets: The instructor provided approval for the use of datasets in this paper.

The Mushroom dataset (MU) contains records drawn from The Audubon Society Field Guide to North American Mushrooms [3]. The goal is to distinguish edible mushrooms from poisonous mushrooms based on 22 features describing the mushroom characteristics such as odor, ring type, and gill size and color.

The Heart Disease dataset (HD) contains records from medical centers in Budapest, Switzerland, and California [4]. The classification operation is to separate patients with heart disease from patients without heart disease utilizing 14 different attributes including, chest pain type, resting blood pressure, and age, to name a few.

The Drug Consumption dataset (DRG) contains records from institutes in the UK, including Rampton Hospital, University of Nottingham, and University of Leicester [5]. Respondents were questioned about their use of 18 legal and illegal drugs and asked about how frequently they used the drug. For this project, a transformation of the dataset into a binary classification problem took place to examine whether or not the participant is a "user" or "non-user" of a particular drug based on 32 attributes such as the big five personality traits. The drug examined in this project is marijuana since it is the most balanced in terms of use and non-use in this dataset.

II. Methods

Four learning algorithms were tested across the datasets consisting of three trials in each. Learning algorithms include: Logistic Regression, K-Nearest Neighbors, Random Forests, and Decision Trees. Algorithm parameters were modeled from the Caruna and Niculescu-Mizil paper [1] as described below. Implementation of Scikit-learn's GridSearchCV [6] is used to find the optimal parameters associated with each algorithm across each trial. Performance of each learning algorithm is evaluated using the performance metrics: accuracy and precision. Scores between all metrics are averaged across the three trials for each dataset and averaged between the three datasets.

The parameter spaces searched for each of the classifiers in each of the datasets are shown below.

Logistic Regression (LOGREG): in this learning algorithm, regularized and unregularized models are implemented. The regularization parameter C in each varies by factors of ten from 10^{-8} to 10^{4}.

K-Nearest Neighbors (KNN): in this learning algorithm, KNN with Euclidean distance is implemented and weighted uniformly and by distance. The number of neighbors is 20, varying by 15, ranging from 1 to 300 for MU and DRG. Neighbor values adjusted to 20 varying by 5, ranging from 1 to 60 for HD due to it being a significantly smaller dataset.

Random Forests (RF): in this learning algorithm, the forests have 1024 trees. The size of the feature set considered at each split is 1,2,4,6,8,12,16, or 20.

Decision Trees (DT): in this learning algorithm, the decision tree classifier CART with the gini criterion is used with a max depth of each ranging from 1 to 10. The best splitting method is used with 2 samples to split.

III. Experiment

For each dataset and learning algorithm, 3 trials are performed. For each trial in the datasets, randomly select 80 percent of the training cases and the remaining 20 percent of cases for testing. A Pipeline GridSearchCV for all learning algorithms is performed with 5-fold cross-validation on the training cases to return the best parameters for the mean over 5-folds. Tuning of GridSearchCV for scoring metrics occurred to refit for accuracy and precision. In each case, precision and accuracy, the model is trained a final time on all the training cases,

selecting optimal parameters, and measuring performance on the test cases.[1]

For each trial and learning algorithm, visualization of training performance is presented through a learning curve, plotting training score against cross-validation score across training examples, a plot for the scalability of the model, showing fit times across training examples, and a plot of model performance, showing score across fit times. Please see the appendix for an example of such graphics.

Table 1 . Algorithm Mean Test Set Performance By Dataset

		KNN	RF	LR	DT
HD	Accuracy	0.809	0.831	0.847	0.781
	Precision	0.789	0.855	0.839	0.794
DRG	Accuracy	0.801	0.791	0.811	0.774
	Precision	0.857	0.845	0.850	0.851
MU	Accuracy	1.000	1.000	0.966	1.000
	Precision	1.000	1.000	0.964	1.000

In the HD dataset, LOGREG performed best in accuracy, and RF showed the highest precision scores. Similarly, in DRG, LOGREG also showed the highest accuracy results, but KNN scored the highest in precision. In contrast, MU, LOGREG did not perform as well as other learning algorithms. Table 1 summarizes the results of each dataset and algorithm accuracy and precision performance. A paired t-test was used between each algorithm across the three datasets to find the p-value, and a calculation of Cohen's d-value to measure effect size; these values are in Appendix 1 along with dataset raw algorithm test scores for the trial. With $p = 0.05$, significant results are shown in DRG, where a comparison between KNN and DT for accuracy resulted in $p = 0.035$ and $d = 2.610$ as well as LOGREG and DT, with $p = 0.007$ and $d = 4.583$. In MUSH significant results are shown between LR and all other classifiers, with $p = 0.01$ and $d = 8.057$ for accuracy, and $p = 0.026$ and $d = 4.978$ for precision.

Table 2 . Algorithm Mean Test Set Performance

	KNN	RF	LR	DT
Accuracy	0.870	0.874	0.875	0.852
Precision	0.882	0.900	0.885	0.881

An analysis of the overall performance of each classification algorithm ensued. In terms of accuracy, LOGREG and RF performed almost equally, followed by KNN and DT. For precision, RF outperformed all other learning algorithms. Table 2 summarizes the mean test set performance for each algorithm. As in Appendix 1, respective p-values and d-values are in Appendix 2. With $p = 0.05$, no significant results were shown between classifiers when averaging test performances across the three datasets and there was no significance in effect size.

Table 3 . Algorithm Mean Optimal Train Set Performance

		KNN	RF	LR	DT
HD	Accuracy	0.814	0.821	0.817	0.749
	Precision	0.856	1.000	0.815	0.864
DRG	Accuracy	0.802	0.807	0.813	0.723
	Precision	0.903	1.000	0.857	0.904
MU	Accuracy	0.999	1.000	0.965	0.998
	Precision	1.000	1.000	0.964	1.000

In comparison to algorithm mean test set performance, as in Table 1, scores in Table 3 reflect the mean train set performance of algorithms adjusted with optimal hyperparameters. In HD, algorithm accuracy performance was similar to test set performance; KNN accuracy slightly increased while the other algorithms slightly decreased. Precision scores improved significantly for HD classifiers when adjusted, notably RF, while LR saw a slight decrease. In DRG, accuracy performance increased somewhat in all classifiers with adjusted parameters except DT, and precision scores improved in all classifiers. In MU, performance with adjusted parameters in the train set remained nearly the same for all classifiers as in their test set performance. Raw scores for dataset algorithm trainset with optimal parameters are in Appendix 3.

IV. Conclusion

Overall, RF showed the most precise results among the four classifiers tested across datasets and had near equal accuracy to that of LR, which showed the highest mean test accuracy performance. As shown in Appendix Table 2, the t-test on average performances between algorithms shows no significant results nor significance in effect size. In retrospect, there is room for improvement; the testing of more classifiers, trials, and metrics would provide insight into the results achieved here. Further assessment of the datasets utilizing a bootstrapping analysis would also prove to be beneficial.

V. Bonus Points

This project deserves bonus points for performing tests on an extra learning algorithm and evaluating the algorithms on an accuracy metric and a precision metric. Furthermore, during each trial, for each algorithm, learning curves are plotted to show performance and other visualizations for each trial algorithm, such as a confusion matrix and ROC curve. Each notebook contains an extensive exploratory data analysis, including comprehensive data visualizations. In HD and MU, perform a preemptive random forest classifier to retrieve the dataset's top features. The top 5 explored extensively before model training.

[1] A .py file containing functions used to perform these calculations and a notebook outlining execution and analysis are in the Appendix; for complete documentation of code and notebooks, please refer to my GitHub, https://github.com/dillon4d/COGS118A_FinalProject

References

[1] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 161–168. [Online]. Available: https://doi.org/10.1145/1143844.1143865

[2] D. Dua and C. Graff, “UCI machine learning repository,” 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[3] G. H. Lincoff, *The Audubon Society Field Guide to North American Mushrooms*. New York: Alfred A. Knopf, 1981.

[4] R. Detrano, A. Janosi, W. Steinbrunn, M. Pfisterer, J.-J. Schmid, S. Sandhu, K. H. Guppy, S. Lee, and V. Froelicher, “International application of a new probability algorithm for the diagnosis of coronary artery disease,” *The American Journal of Cardiology*, vol. 64, no. 5, pp. 304 – 310, 1989. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0002914989905249

[5] E. Fehrman, A. K. Muhammad, E. M. Mirkes, V. Egan, and A. N. Gorban, “The five factor model of personality and evaluation of drug consumption risk,” in *Data Science*, F. Palumbo, A. Montanari, and M. Vichi, Eds. Cham: Springer International Publishing, 2017, pp. 231–242.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

APPENDIX

Below the appendix tables that correspond to the tables mentioned in the paper is a .py file that outlines much of the learning algorithms' calculations (varies slightly by metric and dataset). I have also attached one of the notebooks where data analysis takes place. The notebooks for the datasets as well as each of their py files are found on my GitHub https://github.com/dillon4d/COGS118A_FinalProject

Appendix 1 . p-values and d-values for algorithm comparisons by dataset

Dataset	Metric	KNN & RF	KNN & LR	KNN & DT	RF & DT	RF & LR	LR & DT
HD	Accuracy	p = 0.746	p = 0.463	p = 0.607	p = 0.502	p = 0.814	p = 0.395
		d = 0.289	d = 0.664	d = 0.456	d = 0.608	d = 0.099	d = 0.780
	Precision	p = 0.363	p = 0.297	p = 0.922	p = 0.438	p = 0.827	p = 0.459
		d = 0.907	d = 1.019	d = 0.088	d = 0.708	d = 0.193	d = 0.671
DRG	Accuracy	p = 0.380	p = .291	p = 0.035	p = 0.127	p = 0.097	p = 0.007
		d = 0.807	d = 1.030	d = 2.61	d = 1.601	d = 1.937	d = 4.583
	Precision	p = 0.609	p = 0.679	p = 0.757	p = 0.821	p = .809	p = 0.982
		d = 0.456	d = 0.368	d = 0.272	d = 0.198	d = 0.216	d = 0.020
MU	Accuracy	p = nan	p = 0.010	p = nan	p = nan	p = 0.010	p = 0.010
		d = nan	d = 8.057	d = nan	d = nan	d = 8.057	d = 8.057
	Precision	p = nan	p = 0.026	p = nan	p = nan	p = 0.026	p = 0.026
		d = nan	d = 4.978	d = nan	d = nan	d = 4.978	d = 4.978

Appendix 1.1. Raw Test Set Scores By Dataset

		KNN	RF	LR	DT
HD	Accuracy	Trial 1: 0.787	Trial 1: 0.803	Trial 1: 0.803	Trial 1: 0.771
		Trial 2: 0.771	Trial 2: 0.754	Trial 2: 0.820	Trial 2: 0.721
		Trial 3: 0.869	Trial 3: 0.934	Trial 3: 0.918	Trial 3: 0.853
	Precision	Trial 1: 0.750	Trial 1: 0.815	Trial 1: 0.793	Trial 1: 0.759
		Trial 2: 0.800	Trial 2: 0.784	Trial 2: 0.816	Trial 2: 0.744
		Trial 3: 0.816	Trial 3: 0.967	Trial 3: 0.909	Trial 3: 0.879
DRG	Accuracy	Trial 1: 0.796	Trial 1: 0.801	Trial 1: 0.804	Trial 1: 0.775
		Trial 2: 0.814	Trial 2: 0.796	Trial 2: 0.817	Trial 2: 0.783
		Trial 3: 0.793	Trial 3: 0.777	Trial 3: 0.812	Trial 3: 0.764
	Precision	Trial 1: 0.875	Trial 1: 0.878	Trial 1: 0.859	Trial 1: 0.868
		Trial 2: 0.834	Trial 2: 0.827	Trial 2: 0.835	Trial 2: 0.823
		Trial 3: 0.861	Trial 3: 0.832	Trial 3: 0.857	Trial 3: 0.861
MU	Accuracy	Trial 1: 1.000	Trial 1: 1.000	Trial 1: 0.971	Trial 1: 1.000
		Trial 2: 1.000	Trial 2: 1.000	Trial 2: 0.967	Trial 2: 1.000
		Trial 3: 1.000	Trial 3: 1.000	Trial 3: 0.959	Trial 3: 1.000
	Precision	Trial 1: 1.000	Trial 1: 1.000	Trial 1: 0.972	Trial 1: 1.000
		Trial 2: 1.000	Trial 2: 1.000	Trial 2: 0.968	Trial 2: 1.000
		Trial 3: 1.000	Trial 3: 1.000	Trial 3: 0.953	Trial 3: 1.000

Appendix Table 2. p-values and d-values for algorithm comparisons

	KNN & RF	KNN & LR	KNN & DT	RF & DT	RF & LR	LR & DT
Accuracy	p = 0.935	p = 0.914	p = 0.727	p = 0.678	p = 0.989	p = 0.630
	d = 0.038	d = 0.052	d = 0.167	d = 0.200	d = 0.006	d = 0.232
Precision	p = 0.680	p = 0.942	p = 0.994	p = 0.682	p = 0.687	p = 0.937
	d = 0.198	d = 0.035	d = 0.004	d = 0.197	d = 0.194	d = 0.038

Appendix 3. Raw Optimal Train Set Scores By Dataset

		KNN	RF	LR	DT
HD	**Accuracy**	Trial 1: 0.827 Trial 2: 0.826 Trial 3: 0.789	Trial 1: 0.818 Trial 2: 0.831 Trial 3: 0.814	Trial 1: .831 Trial 2: 0.806 Trial 3: 0.814	Trial 1: 0.736 Trial 2: 0.814 Trial 3: 0.699
	Precision	Trial 1: 0.825 Trial 2: 0.872 Trial 3: 0.871	Trial 1: 1.000 Trial 2: 1.000 Trial 3: 1.000	Trial 1: 0.813 Trial 2: 0.840 Trial 3: 0.791	Trial 1: 0.744 Trial 2: 0.970 Trial 3: 0.879
DRG	**Accuracy**	Trial 1: 0.801 Trial 2: 0.801 Trial 3: 0.804	Trial 1: 0.810 Trial 2: 0.803 Trial 3: 0.808	Trial 1: 0.816 Trial 2: 0.813 Trial 3: 0.812	Trial 1: 0.787 Trial 2: 0.694 Trial 3: 0.688
	Precision	Trial 1: 0.858 Trial 2: 0.852 Trial 3: 1.000	Trial 1: 1.000 Trial 2: 1.000 Trial 3: 1.000	Trial 1: 0.855 Trial 2: 0.860 Trial 3: 0.855	Trial 1: 0.903 Trial 2: 0.909 Trial 3: 0.898
MU	**Accuracy**	Trial 1: 0.999 Trial 2: 0.999 Trial 3: 0.999	Trial 1: 1.000 Trial 2: 1.000 Trial 3: 1.000	Trial 1: 0.962 Trial 2: 0.968 Trial 3: 0.964	Trial 1: 0.998 Trial 2: 1.000 Trial 3: 0.996
	Precision	Trial 1: 1.000 Trial 2: 1.000 Trial 3: 1.000	Trial 1: 1.000 Trial 2: 1.000 Trial 3: 1.000	Trial 1: 0.961 Trial 2: 0.968 Trial 3: 0.963	Trial 1: 1.000 Trial 2: 1.000 Trial 3: 1.000

Example of visualization for an accuracy trial of LR in HD