COGS 118A Final Project

Qixuan Ma
UC San Diego, Cognitive Science Department
9500 Gilman Dr., La Jolla, CA, USA
q5ma@ucsd.edu

Abstract

In this study, we compared the performance of 3 different algorithms (Artificial Neural Network, Logistic Regression, and Random Forest) across 4 different datasets to get three performance metric scores (accuracy, area under ROC, F1 score) using the optimal hyperparameters obtained from grid search with cross validation. Overall, we found that Random Forest algorithm performs the best amongst three models across all three performance metrics. It is also best performing when it comes to handling the HTRU2 dataset and the Adult Income dataset, while coming close to being the best at classifying the Connect 4 dataset and the King-Rook vs. King Dataset.

1 Introduction

A lot studies have been conducted in the topic of performance evaluations comparing different supervised machine learning algorithms, such as "An Empirical Comparison of Supervised Learning Algorithms" by Rich Caruana and Alexandru Niculescu-Mizil at Cornell University. The purpose of this study is to attempt to replicate and reproduce results from previous experiments and studies. In this study, I will compare the performance of 3 different supervised machine learning algorithms across 4 different datasets, performing hyperparameter search with 5-fold cross validation over a set of parameter settings. Afterwards, I will train a model according to optimal hyperparameters and compare the performance metrics generated by applying these models to testing tests.

2 Method

2.1 Learning Algorithms

2.1.1 Artificial Neural Networks (ANN)

I trained ANNs with the Multi-Layer Perceptron Classifier algorithm included in the scikit learn package. We trained the model using hidden_layer_sizes = [1,2,4,8,32,128] and momentum = [0,0.2,0.5,0.9].

2.1.2 Logistic Regerssion (LOGREG)

I trained both unregularized and regularized Logistic Regression models using scikit's native LogisticRegression() function. For both L_2 regularized model and unregularized model, I used regularization parameters that varied from 10^{-8} to 10^4. In order to prevent ConvergenceErrors to take up several pages, I used the newton-cg solver, which did not cause any ConvergenceError during all trials.

2.1.3 Random Forests (RF)

I trained Random Forest Classifiers using the implementation in scikit learn with 1024 trees in the forest. The maximum number of the feature set at each split is 1, 2, 4, 6, 8, the minimum sample needed to split is 2, 5, 10, and the minimum sample on a single leaf is 1, 2, 4.

2.2 Performance Metrics

To evaluate the performance of the different learning algorithms, I used three performance metrics to quantify their performance: accuracy (ACC), area under the ROC curve (ROC), and F-score (FSC).

2.3 Data Sets

To compare the performance of the three algorithms, I used 4 datasets taken from UCI's machine learning database. The task we performed in all three algorithms across four datasets is binary classification (-1 and 1).

2.3.1 HTRU2 Dataset

HTRU2 dataset is collected in the High Time Resolution Universe Survey, which contains information about astronomical bodies that are candidates for Pulsar Stars. This dataset contains 17898 observations each with 9 attributes.

2.3.2 Connect 4 Dataset

The Connect 4 Dataset contains all positions in the 8x8 board where the winner is yet to be decided and the next move is not forced. Each point in the grid is encoded as one of $\{x, o, b\}$, where x and o represents tokens from both sides and b represents blank. The outcome class is the outcome for the first player (win, draw, loss). To make results from this dataset comparable to other binary classification tasks, we will focus on whether the first player is predicted to win $(+1)$ or loss/draw (-1). This dataset contains 67557 observations each with 42 attributes. We performed one hot encoding for the location of the pieces.

2.3.3 King-Rook versus King Dataset

The King-Rook versus King dataset contains a set of board layouts for a chess endgame with white having a King and a Rook while black only have the King. The original purpose of this dataset is to perform multi-class classification, where the class is the depth of win for white (at least how many moves it take white to win), which have the categories of 0,1,...,16,draw. For our purpose of benchmarking algorithms for binary class classification, we will be ignoring the depth of win and just focus on whether white wins (+1) or draws (-1). There are 28056 observations in this dataset, each containing 6 attributes on the location of the pieces on the board.

2.3.4 Adult Census Income Dataset

The Census Income Dataset (also know as "Adult" dataset) contains census data ranging from gender, level of education, job type, etc. The task associated with the dataset is to classify the observations into 2 classes "$\leq$ 50k" and "$>$ 50k" for the income of the subject. There are 48842 observations in this dataset, each with 14 attributes. In order to transform all categorical variables to floating point values, we performed one-hot encoding for each categorical variable.

3 Experiment

The general procedure for this experiment is as follows. We chose 3 different algorithms: Artificial Neural Network (Multi-Layer Perceptron Classifier), Logistic Regression (LogisticRegression), and Random Forest (RandomForestClassifier). We also gathered 4 datasets, as

described above. For each combination of algorithm and dataset, we will perform 5 trials of hyperparameter selection using GridSearch with 5-fold split cross validation. After each trial, we find a optimal set of hyperparameters for each of our metrics (accuracy, area under ROC, F1 score), fit a random sample of the data using the optimal hyperparameter, and storing the output values for our metrics. At the end of all 5 trials, we take average of all metrics for each algorithm-dataset combination. Then we take the average of all algorithm-dataset combo to find which algorithm performs the best in certain tasks.

4 Results

Table 1: Score for Algorithms Averaged by Performance Metrics

Score for each Learning Algorithm by Metric				
Model	Accuracy	AUC	F1 Score	Mean
ANN	0.868	0.895	0.867	0.877
LOGREG	0.878	0.878	0.878	0.878
RF	**0.910**	**0.909**	**0.910**	0.910
Mean	0.885	0.894	0.885	0.888

Table 2: Score for Algorithms Averaged by Different Datasets

Score for each Learning Algorithm by Problem					
Model	HTRU2	Connect 4	KR-K	Adult	Mean
ANN	0.975	**0.824**	**0.994**	0.713	0.877
LOGREG	**0.979**	0.784	0.900	0.850	0.878
RF	**0.979**	0.818	0.986	**0.856**	0.910
Mean	0.978	0.809	0.960	0.806	0.888

5 Discussion

5.1 Interpretation of Results

Looking at Table 1, we can see that the Random Forest model performed the best in all three performance metrics. It was able to achieve $> 90\%$ accuracy on average across 4 datasets, which made this model considerably better than Artificial Neural Network and Logistic Regression. By using a collection of decision trees in classification, Random Forest is able produce much more generalizable results as it is robust against random error and against overfitting.

In Table 2, we observe that despite not the best in all categories, the performance of Random Forest is still strong across multiple tasks. This observation corroborates with previous findings by Caruana and Miculescu-Mizil, which ranked Random Forest as one of the best classifiers for binary classification. It came ahead in average amongst all performance metrics in HTRU2 and Adult Income dataset, while coming at a very close second in the Connect 4 and King-Rool vs. King dataset. In the other 2 classifiers, while the Artificial Neural Network performed extremely well (almost perfect) in classifying the King-Rook vs. King dataset, it fell short in classifying the Adult Income dataset, achieving an average score of merely 0.713. The Logistic Regression classifier matched the performance of Random Forest in the HTRU2 classification task, but performed rather poorly in classifying the Connect 4 dataset.

The No Free Lunch Theorem suggests that there is not one single algorithm that is the best at everything, which is corroborated by the findings described above. While the Artificial Neural Network and Logistic Regression Classifier achieved admirable results in some tasks, they fell short in others. In our analysis, Random Forest seem to come out ahead when it comes to average performance in multiple datasets. However, a sample size of four cannot

Table 3: Time Complexity of Three Algorithms in Question

Algorithm	Training Time Complexity	Testing Time Complexity
ANN (MLP)	$n_{iter} * n$	$O(pnl)$
LOGREG	$O(np)$	$O(p)$
RF	$O(n^2 * p * n_{trees})$	$O(p * n_{trees})$

fully describe the performance of Random Forest Classifier in general. Given more datasets, I am confident that we will be able to find tasks that Random Forest do not excel at. In addition, there are multiple implementations of the Random Forest Classifier which can either improve or reduce the performance of the algorithm in certain tasks.

5.2 Time Complexity Analysis

In this section, I will use p to denote the number of attributes in each observation, and n to denote the number of observations used to train the model.

6 Appendix

6.1 Raw Performance Metric Scores

This section contains the raw scores for different performance metrics for all algorithm-dataset combination (Table 4-7). Each column represent one trial.

Table 4: Raw Score for Performance Metrics for HTRU2 Dataset

ANN	ACCU	0.976	0.973	0.974	0.974	0.974
	ROC	0.974	0.972	0.974	0.976	0.977
	F1	0.975	0.977	0.974	0.974	0.976
LOGREG	ACCU	0.979	0.980	0.978	0.980	0.979
	ROC	0.979	0.980	0.978	0.980	0.978
	F1	0.979	0.980	0.978	0.979	0.979
RF	ACCU	0.982	0.977	0.978	0.980	0.980
	ROC	0.982	0.979	0.978	0.979	0.980
	F1	0.982	0.978	0.977	0.979	0.980

Table 5: Raw Score for Performance Metrics for Connect 4 Dataset

ANN	ACCU	0.976	0.973	0.974	0.974	0.974
	ROC	0.974	0.972	0.974	0.976	0.977
	F1	0.975	0.977	0.974	0.974	0.976
LOGREG	ACCU	0.979	0.980	0.978	0.980	0.979
	ROC	0.979	0.980	0.978	0.980	0.978
	F1	0.979	0.980	0.978	0.979	0.979
RF	ACCU	0.982	0.977	0.978	0.980	0.980
	ROC	0.982	0.979	0.978	0.979	0.980
	F1	0.982	0.978	0.977	0.979	0.980

In the calculation of the average, we just took the arithmetic mean of the raw scores above.

6.2 Training Set Performance

The following table describes the testing accuracy and training accuracy of different dataset-algorithm combinations and the difference between them.

We can see that for most of the algorithm-dataset combinations, we do not observe a

Table 6: Raw Score for Performance Metrics for King-Rook vs. King Dataset

ANN	ACCU	0.976	0.973	0.974	0.974	0.974
	ROC	0.974	0.972	0.974	0.976	0.977
	F1	0.975	0.977	0.974	0.974	0.976
LOGREG	ACCU	0.979	0.980	0.978	0.980	0.979
	ROC	0.979	0.980	0.978	0.980	0.978
	F1	0.979	0.980	0.978	0.979	0.979
RF	ACCU	0.982	0.977	0.978	0.980	0.980
	ROC	0.982	0.979	0.978	0.979	0.980
	F1	0.982	0.978	0.977	0.979	0.980

Table 7: Raw Score for Performance Metrics for Adult Income Dataset

ANN	ACCU	0.976	0.973	0.974	0.974	0.974
	ROC	0.974	0.972	0.974	0.976	0.977
	F1	0.975	0.977	0.974	0.974	0.976
LOGREG	ACCU	0.979	0.980	0.978	0.980	0.979
	ROC	0.979	0.980	0.978	0.980	0.978
	F1	0.979	0.980	0.978	0.979	0.979
RF	ACCU	0.982	0.977	0.978	0.980	0.980
	ROC	0.982	0.979	0.978	0.979	0.980
	F1	0.982	0.978	0.977	0.979	0.980

Table 8: Training Set Performance vs. Testing Set Performance

Dataset	Algorithm	Testing Accu	Training Accu	Difference
HRTU2	ANN	0.975	0.975	0
	LOGREG	0.979	0.980	0.001
	RF	0.979	0.988	0.009
Connect 4	ANN	0.824	0.962	0.151
	LOGREG	0.784	0.792	0.008
	RF	0.995	0.819	0.176
KR-K	ANN	0.994	0.996	0.002
	LOGREG	0.900	0.901	0.001
	RF	0.986	1	0.014
Adult	ANN	0.679	0.780	0.101
	LOGREG	0.850	0.855	0.005
	RF	0.856	0.899	0.043

significant difference between the testing accuracy and training accuracy. This tells us that the specific dataset, algorithms, and hyperparameters we chose were not overfitted and was not erroneous, which further validates our conclusions.

6.3 Code

All code used to generate the data is attached at the end of this document.

7 Acknowledgements

I thank Professor Jason Fleischer for planning and instructing COGS 118A, which provided most of the background knowledge for this paper. I also thank TAs Yifan Xu and Abdullah Albattal for furthering my knowledge in the subject matter during both discussions and office hours. Last but not least I thank every contributor to academic discussion both on the COGS 118A Piazza and Discord for providing clarification and assistance on a lot of logistical problems.

8 Reference

Caruana R., & Niculescu-Mizil, A. (2006). An Empirical Comparison of Supervised Learning Algorithms.
Serpen G., & Gao Z. (2014). Complexity Analysis of Multilayer Perceptron Neural Network Embedded into a Wireless Sensor Network.
Xu Y., et al. (2015). A Fast Learning Method for Multilayer Perceptrons in Automatic Speech Recognition Systems.

9 Extra Credit

I believe I should receive extra credit for doing research on the time complexity analysis on MultiLayer Perceptron models and including multiple secondary analysis.