Supervised Learning Algorithms for Classification

Andrew Hernandez
University of California, San Diego
Cognitive Science

ABSTRACT

The current study evaluates the differences in performance of three supervised-based learning methods on three classification tasks available through the UCI Machine Learning Repository. Logistic regression, k-nearest neighbors, and random forest are all evaluated to determine the differences in performance and time-complexity when used on three distinct data sets.

Keywords: Classification, K-Nearest Neighbors, Logistic Regression, Random Forest

1. INTRODUCTION

Choosing a machine learning model to learn from data involves the important consideration of factors like an algorithms time-complexity and its performance on a given type of data. The current study evaluates the performance of logistic regression, k-nearest neighbors, and random forest on three datasets with accuracy as a single metric. The procedure of this study is motivated by the work of Caruana and Niculescu-Mizil (2006), where an empirical comparison of ten supervised models was conducted with other performance metrics also considered.

The three datasets used within this study come from the UCI repository and vary in sample size and in the number of features used to predict the target class. Variations in sample size across datasets are useful for showing trade-offs between model performance and runtime when training the algorithm and using it to make predictions. Previewing the results of this study, random forest models tend to outperform both logistic regression and k-nearest neighbors, but their runtime increases significantly as the dataset gets larger, so it may be less desirable to implement this sort of model if time-constraints exist.

Each algorithm is evaluated with an accuracy metric that measures the number of accurate predictions over the total number of predictions made. This type of evaluation is not optimal for datasets that are imbalanced since a model can have high accuracy while also performing poorly when predicting the minority class. For these problems, the area under the curve obtained by the receiver operating characteristic may prove to be a more optimal metric.

Hyperparameters are parameters passed into the model by the user and have an important role in determining the model's behavior and performance. The space of parameters for each algorithm was determined for each algorithm and used consistently across all datasets.

2. METHODOLOGY

2.1 Learning Algorithms

Logistic Regression: The optimizer, or solver, of the models varies between a newton method, a coordinate descent algorithm, and one that updates the gradient evaluation by approximating the second derivative matrix. The regularization parameter varies between 10^{-8} and 10^{4} in factors of 10, where the penalty is either the L1 or L2 norm, or none for an unregularized implementation.

KNN: the k-value ranged from 1-26. Distance and uniformly weighted implementations were both used. For uniform weights, all values are weighted equally. Neighbors closer to the point of interest have a greater influence with distance weights.

Random Forest: the number of trees in the forest was constant at 1024 trees. The number of features considered at each split was 1, 2, 4, 6, 8, 12, 16, or 20. The quality of a split was determined by the Gini impurity, or “gini” for short.

2.2: Performance metric

Performance of each algorithm was scored by the accuracy of its predictions. The number of true negatives is summed with the number of true positives and then divided by the total population of predictions in order to get this value.

2.3 Datasets

Three datasets were chosen for binary classification tasks from the UCI machine learning repository. The largest of these is the Forest Cover Type dataset, or Cov_type, with a total of 581,012 samples and 54 input features scaled between 0 and 1. Cov_type was altered by treating the target class that appeared the most as the positive class, and all other target values as the negative class. The Adult dataset was the second largest with 45,222 samples after removing columns with missing values. Dataset Adult had 14 input features, or 104 inputs after hot-encoding categorical features and scaling numerical inputs between 0 and 1. Letter.p2 is a binary version of the Letter dataset that treats the characters A-M as the positive class, and all others as the negative class. All input features for Letter.p2 are numerical and were scaled between 0 and 1. Table 1 below shows the characteristics of these problems.

Problem	#of attributes	Train size	Test Size
Adult	14/104	5000	40222
Cov_type	54	5000	576012
Letter.p2	16	5000	15000

Table 1. Description of datasets

3. EXPERIMENT

For each preprocessed dataset and algorithm, 5000 samples are randomly selected for training, and other samples are held out in order to form a test set. A repeated stratified 5-fold cross validation is used to evaluate the model configurations provided by a parameter-grid during an exhaustive grid search. 5 folds means that for five iterations, a model is trained on 4000 samples and evaluated on a unique holdout set containing the remaining 1000 samples. The number of repeats for the stratified k-fold process is three, so a total of 15 different hold-out sets are used for estimating a model’s performance. An optimal model with the parameters that gave the best performance is then chosen and trained on the entirety of the training set and evaluated on the test set in order to produce one trials worth of results. This process is repeated three times so that three different optimal models are acquired and trained on a set of 5000 samples and then evaluated on the test set. Table 2 shows the mean test set performance across three trials for each algorithm and dataset. This means that the performance of the optimal model on the test set for each dataset and algorithm is averaged across the three trials. The algorithm with the best performance is boldened for each dataset, and those with performances on the same dataset that are not statistically distinguishable from the best algorithm are labeled with an asterisk. Two-sample t-tests across the 3 trials are used to acquire the p-values. Entries not boldened or labeled have performance significantly lower at p = 0.05. Random forest models outperform the others on all datasets but Letter.p2, but its performance is not statistically distinguishable from k-nearest neighbors, which is the best model for Letter.p2.

Table 2. Mean test set performance per algorithm and dataset

	Adult	Cov_type	Letter.p2
Log. Reg	.843833	.756260	.724422
k-NN	.823074	.794162	**.956978**
Rand. Forest	**.848474**	**.823918**	.948089*

The mean test set performance across all trials and datasets was also calculated and is shown in Table 3. This was done by taking the optimal model's performance on the test set for each trial and dataset, which would result in 9 test set performances that can be averaged to produce a mean test set performance for each of the three algorithms. Two-sample t-tests were performed to compare each algorithm across all 9 trials. The algorithm with the best mean test set performance is boldened, and those not statistically distinguishable are labeled with asterisks. The random forest algorithm was the best overall across all datasets and trials, but the k-NN was not significantly distinguishable from it and was also considerably faster to train.

Algorithm	Accuracy
Log_reg	.774838
k-NN	.858071*
Rand_Forest	**.873494**

Table 3. Mean test set performance per algorithm

The cross-validation performance for each algorithm with the optimal hyperparameters was averaged across the 3 trials for each dataset and is shown in Table 4. Logistic regression tends to have the worst cross-validation performance and test set performance when compared to the other two algorithms. The differences between k-nearest neighbors and random forest are not quite so obvious when looking at mean cross-validation or test set scores, but the differences in time-complexity are drastic. Using a smartphone stopwatch, the amount of time to perform the training of the models and acquire a performance measure on the test were acquired for each algorithm only once on each dataset. Letter_p2 was faster for training and testing the algorithms due to the relatively small size of the data. While random forest has shown to have higher cross-validation and test-set performance scores, it takes far longer to train and make predictions compared to k-nearest neighbors, which is usually just slightly behind in performance. The time in seconds for each algorithm to train and acquire a test score is shown in table 5. Surprisingly, the amount of time for random forest to be trained and acquire a test set score was longer on the Adult dataset than the Cov_type dataset, where the latter had the larger number of samples.

Algorithm	Adult	Cov_type	Letter.p2
Log_reg	.852911	.756822	.726178
k-NN	.830978	.786644	.950067
Rand_Forest	.852511	.819400	.937933

Table 4. Mean Cross-Validation score per algorithm and dataset

Algorithm	Adult	Cov_type	Letter.p2
Log_reg	2:20	3:05	:49
k-NN	2:28	4:35	1:18
Rand_Forest	23:20	15:58	10:40

Table 5. Time(min:seconds) to train and acquire a test set score

4. CONCLUSION

This study compared the performance of three supervised learning methods on three unique datasets from the UCI machine learning repository. The results showed that the random forest classifier generally produces the most accurate cross-validation and test set scores when compared to logistic regression and k-nearest neighbors. The k-nearest neighbors performance was similar to that of a random forest, but was also significantly faster to train due to a reduced time-complexity. It would be interesting to see how these models differ when other metrics besides accuracy are considered.

REFERENCES

[1] Caruana, R., & Niculescu-Mizil, A. (2005). An Empirical Comparison of Supervised Learning Algorithms Using Different Performance Metrics

APPENDIX

Table of raw train and test scores for each algorithm

Dataset & trial #	log_reg train	log_reg test	k-NN train	k-NN test	RF train	RF test
Adult trial 1	.850200	.843320	.828533	.823007	.855867	.846477
Adult trial 2	.859467	.842996	.837733	0.823853	0.849467	0.848963
Adult trial 3	.849067	.845184	.826667	0.822361	0.852200	0.849983
Cov_type trial 1	.753467	.757114	.785267	0.794983	0.815867	0.822757
Cov_type trial 2	.760800	.755434	.787600	0.795895	0.820933	0.819636
Cov_type trial 3	.756200	.756231	.787067	0.791607	0.821400	0.829361
Letter.p2 trial 1	.719733	.723267	.944733	0.957133	0.932267	0.954133
Letter.p2 trial 2	.719000	.727467	0.955133	0.955733	0.940733	0.945733
Letter.p2 trial 3	.739800	.722533	0.950333	0.958067	0.940800	0.944400

Table of p-values following t-test comparing algorithm test set performance on each dataset across 3 trials

Input into t-test	Adult, p-values	Cov_type, p-values	Letter.p2 p-values
Log_reg, k-NN	0.000053	0.000332	0.000002
k-NN, RF	0.000403	0.003344	0.093728
Log_reg, RF	0.026395	0.001387	0.000009

Table of p-values following t-test comparing algorithm test set performance across all datasets and trials

Input into t-test	p-values
Log_reg, k-NN	0.016767
k-NN, RF	0.631303
Log_reg, RF	0.001647