.COGS 118A FINAL PROJECT REPORT

Hongbin Miao
UC San Diego
homiao@ucsd.edu

Abstract

In this study, we compared the performance of five different supervised learning algorithms: Logistic Regression, SVMs, Neural Nets, Random Forest, and Gradient Boost. Furthermore, we compared the performance of these five different algorithms on four datasets on binary classification.

1. Introduction

COGS 118A: Supervised Machine Learning Algorithms is a cognitive science curriculum at UC San Diego. This curriculum focuses on the rigorous study and comparison of various supervised machine learning algorithms, including decision stumps, linear regression, logistic regression, and SVMs. Hence, this research project focuses on applying the knowledge to compare various machine learning algorithms.

2. Method

2.1 Learning Algorithms

All the learning algorithms that we use below are based on the implementation by scikit-learn, a Python module for machine learning.

Logistic Regression (LR): we trained both unregularized and regularized models. With penalties in {“None”, “l2”, “l1”, “elasticnet”} and regularization parameter C in {0.1, 1, 10}.

SVMs: we trained support vector machines based on the implementation of libsvm. With kernel in {“linear,” “rbf” } and regularization parameter C in {0.1, 1, 10}.

Neural Nets (NN): we trained several multi-layer perceptron classifiers with hidden layer sizes (in which the ith element represents the number of neurons at the ith hidden layer) in {(50,), (100,), ((100, 50)), ((100, 50, 25))}, and learning rate alpha in {0.0001, 0.001}.

Random Forests (RF): we trained several random forests classifiers with n_estimators in {100, 200} and max_depth in {None, 10, 20}.

Gradient Boost (GB): we trained several gradient boosting classifiers with n_estimators in {100, 200} and learning rate in {0.01, 0.1}.

2.2 Performance Metrics

We choose five metrics to evaluate the model's performance: accuracy, precision, recall, f1, and roc_auc.

2.3 Datasets

Limited by the hardware bandwidth available, the datasets we selected do not contain samples above the scale of thousands. However, we still prepared four representative datasets. The first dataset is the “Iris,” a classic classification dataset with 150 instances and four real number features. The second dataset is “Abalone,” a dataset for predicting abalone age from physical measurements. This dataset contains 4177 instances and eight features (categorical, integer, and real). The third dataset is “Wine,” which uses chemical features to determine the origin of wines. This dataset contains 178 instances and 13 features (integer, real). The fourth dataset is the “Sundanese Twitter Dataset,” which is used for emotion classification for the tweets of the second-largest local language in Indonesia. This dataset contains 2510 instances and one feature (types of emotion represented in string format).

Since our investigation concerned the classifiers’ performance in binary classification problems, we transformed the datasets above. Firstly, for the “Iris” dataset, we relabel the target label “Iris-setosa” as “1” and the rest of the target labels, namely, “Iris-versicolor” and “Iris-virginica,” as “0”. Secondly, for the “Abalone” dataset, we relabel the target labels that are in the following age: 8, 9, 10, and 13 as “1” and the rest as “0”. We pick this specific set of ages as positive labels to maintain a balanced dataset, which now has 2094 positive and 2083 negative labels. Thirdly, for the "Wine” dataset, we relabel targets with the original label “2” as “1” and targets with other original labels as “0”. Fourthly, for the “Sundanese Twitter Dataset,” we performed several text preprocessing steps. First, we remove stopwords (based on the list provided by the dataset) from the training set. Second, we choose the top 100 most occurring words from a random sample of 2094 text samples (the size of the provided training set) from the combined train and test set. Then, we converted every sample in the combined dataset into a bag-of-word vector.

3. Experiment

3.1 Experimental Design

We perform three kinds of train-test split for each dataset: 20/80, 50/50, and 80/20. Then, within each training set from the train-test split, we perform a 50/50 split to get a balanced training and validation set.

For each train, valid, and test set combination for each dataset, we perform a grid search with 5-fold cross-validation and accuracy scoring for finding optimal hyperparameters. Also, we fit the grid search with the training set. Upon finding the best estimator through grid search, we collected the data from the five metrics on the train set, validation set, and test set evaluation on that best estimator.

We repeated the process above for each of the datasets and for each of the classifiers. The hyperparameter being tuned in the experiment has been discussed in the 2.1 Learning Algorithms section.

3.2 Performance by Metric

MODEL	Accuracy	Precision	Recall	F1	Roc_Auc	Mean
LR	0.89 train 0.85 valid 0.85 test	0.89 0.86 0.84	0.89 0.84 0.85	0.89 0.84 0.84	0.92 0.90 0.90	0.90 0.86 0.86
SVM	0.89 0.85 **0.86**	0.88 0.85 0.85	0.90 0.86 **0.86**	0.89 0.85 **0.86**	0.50 0.50 0.50	0.81 0.78 0.79
NN	0.86 0.82 0.82	0.82 0.79 0.77	0.78 0.77 0.77	0.78 0.77 0.75	0.91 0.90 0.89	0.83 0.81 0.80
RF	0.97 0.87 **0.86**	0.96 0.89 **0.88**	0.98 0.85 0.85	0.97 0.86 **0.86**	0.99 0.92 **0.92**	0.97 0.88 **0.87**
GB	0.91 0.84 0.84	0.91 0.83 0.85	0.94 0.86 0.84	0.92 0.84 0.84	0.95 0.88 0.88	0.93 0.85 0.85

This chart represents each model's performance on the five metrics: accuracy, precision, recall, f1, and roc_auc. Each box contains three numbers representing the average from the training, validation, and testing sets, respectively. The result in each box is obtained by averaging the specific metrics on the subset of the four datasets (train, valid, test).

From the result, we can observe that Random Forest has the overall best performance. Surprisingly, Logistic Regression outperformed SVM and Neural Nets on the mean test set. One of the possible reasons for this surprising result is that the complexity of our dataset is relatively low in terms of the quantity and variability of features. Also, our dataset is relatively small (no more than thousands of samples).

3.3 Performance by Problem

MODEL	**Iris**	**Abalone**	**Wine**	**Sundanese Tweet**	**Mean**
LR	1.00 train 1.00 valid **1.00 test**	0.68 0.68 0.67	0.97 0.85 0.85	0.93 0.90 **0.89**	0.90 0.86 0.85
SVM	0.90 0.90 0.90	0.68 0.66 0.65	0.84 0.77 0.79	0.83 0.80 0.80	0.81 0.78 0.79
NN	1.00 1.00 **1.00**	0.72 0.71 **0.70**	0.63 0.64 0.61	0.97 0.89 **0.89**	0.83 0.81 0.80
RF	1.00 1.00 **1.00**	0.95 0.70 0.68	1.00 0.92 **0.93**	0.95 0.89 0.88	0.98 0.88 **0.87**
GB	1.00 1.00 **1.00**	0.76 0.69 0.68	1.00 0.80 0.83	0.94 0.90 **0.89**	0.92 0.85 0.85

This chart represents each model’s performance on the four datasets, respectively. Each box contains three numbers representing the average from the training, validation, and testing sets, respectively. The result in each box is obtained by averaging the five metrics on the specific subset of each dataset (train, valid, test).

Like the result in 3.2, Random Forest has the overall best performance, while Logistic Regression and Gradient Boost have the same mean performance.

4. Conclusions

Our result resembles the finding in “An Empirical Comparison of Supervised Learning Algorithms.” Random Forest is the top-performing method across the two evaluation methods. Neural Nets did not have a significant advantage due to our relatively small training samples. Logistic Regression and Gradient Boost perform relatively less than Random Forest on these four datasets. However, Logistic Regression is a relatively good classifier given a small dataset.

References

Caruana, R., Niculescu-Mizil, A. An Empirical Comparison of Supervised Learning Algorithms.