A Comparison of Supervised Machine Learning Algorithms on Different Training-Testing Partitions of Data

Timothy Taylor
University of California, San Diego
tvtaylor@ucsd.edu

Abstract

The present study empirically analyzes the performance of five different supervised machine learning algorithms across three different partitions of training and testing data. We partition each of three different datasets with binary targets into a 20:80, 50:50, and 80:20 proportion of training to testing data respectively. We then examine the accuracy metrics of cross validated decision trees, random forests, logistic regression classifiers, k-nearest-neighbors classifiers, and two layer neural networks on these sets. Using these metrics, we compare both the performance of the five algorithms to each other, and the performance of the algorithms trained on one partition of data to the same algorithm trained on different partitions.

1 Introduction

In this study, we seek to find the effect of partitioning data into training and testing sets on the performance of an algorithm trained on such sets. Furthermore, we want to compare the performance metrics of each different classifier.

To measure testing performance of each classifier (random forest, logistic regression, KNN, neural network, decision tree), we simply measure the direct accuracy of their predictions on the test set. Considering that having enough data is crucial to training any machine learning algorithm, we naturally expect that a higher proportion of training data to testing data will favor test performance over partitions with a lower amount of training data, which is ultimately supported in our results. We find that the higher the proportion of test data in the partition, the higher the average accuracy of the classifier across the three datasets.

When comparing classifiers, we find that different classifiers perform better on different datasets, most likely due to the very different aspects of each dataset. We choose to use the adult income dataset, which is large with a high number of features; 47,621 samples with 108 features. We also use the breast cancer dataset, and heart disease dataset, which are both smaller with 569 samples with 30 features, and 297 samples with 13 features respectively. Ultimately, we find that logistic regression tends to perform the best on the smaller sets, with the random forest model performing the best on the largest dataset.

2 Methods

2.1 Data processing and selection

We choose all three datasets from the UC Irvine Machine Learning Repository with binary targets. Of the datasets we use, the first is an adult income dataset with 47,621 samples and 108 features. Second is the breast cancer dataset with 569 samples with 30 features, and the last dataset we choose is the heart disease dataset with 297 samples and 13 features. To ensure that our classifiers have the best results, we normalize, numerically encode, and fix missing values in the datasets. We drop samples that are missing values, and use one hot encoding to turn categorical features into numerical features. Moreover, since certain features stretch over a much more broad range of values than others, we apply Z-score normalization to even out the weight of each feature that will be inputted into our models.

2.2 Problem

We use the 1.3.2 release of scikit-learn to implement each of the classifiers in this study. For each parameter that is not specified we use default settings. We perform 3-fold cross validation using an 80:20 split to find the best hyperparameter for each of the following classifiers:
Decision Trees: We test decision trees with a max depth of {1, 2, 3, ..., 20} using scikit-learn's DecisionTreeClassifier.
Random Forests: We test random forests with a max depth of {1, 2, 3, ..., 20} using scikit-learn's RandomForestClassifier.
Logistic Regression: We test logistic regression classifiers with a hyperparameter C of $\{10^{-8}, 10^{-7}, 10^{-6}, \ldots, 1, 10^{1}, \ldots, 10^{4}\}$ using scikit-learn's LogisticRegression.
Neural Networks: We test ANNs with 2 hidden layers of equal size of {1, 2, 4, 8, 16} using scikit-learn's MLPClassifier.
KNN: We test the KNN classifier with a number of neighbors K ranging from 1 to the number of samples in the respective dataset across 26 evenly spaced intervals using scikit-learn's KNeighborsClassifier.

2.3 Performance evaluation

For each classifier, we track the validation accuracy of the model with the best hyperparameters found during cross validation. In order to track the performance of each classifier, we assess its training and testing accuracy for each partition of each dataset, averaged over 3 trials to ensure that we do not obtain an inaccurate result due to the randomness of the algorithms.

In addition to the individual accuracy for each partition, we also track the average accuracy of each classifier for each partition size across all the datasets. Doing so gives us an easily interpretable metric on how the partition size affects the accuracy of each classifier, and also more generally how each classifier compares to each other classifier on each dataset.

To get an overall metric of the best performing classifiers and the consistency in their performance across datasets, we track the number of times across all datasets that each classifier was the top performer, second best, third best, etc.

3 Experiments

To attain all of our initial performance metrics, we train each of the five classifiers on each partition of each dataset, and we use three-fold cross validation across three trials to obtain our performance metrics. In *Table 1, Table 2,* and *Table 3* we display the performance metrics we measure for each dataset; namely, the validation accuracy corresponding to the best hyperparameters, those hyperparameters we obtained, and the training and testing accuracies (both averaged over three trials). To make it clear which model performs the best during testing, we display the row corresponding to the classifier with the highest test accuracy for that dataset in **bold typeface**.

We find that the logistic regression classifier performs the best on the two smaller datasets (the heart disease and breast cancer sets), whereas the random forest classifier performs the best on the largest dataset on adult income. This is a very interpretable result as the random forest classifier is a more complex model than logistic regression, so it naturally performs better on the more complex dataset with a higher amount of samples and features, while logistic regression shines in a simpler setting.

To get a better understanding of the consistency of each model's performance, we evaluate the number of times across all datasets that each classifier was the top performer, second best, third best, etc. in *Table 4*. The results show that random forests and logistic regression tend to perform the best, the former slightly behind the latter, while ANNs are more varied in their results, with KNN classifiers and decision trees tending to perform the worst of the five. Typically, in a higher dimension setting, as is the case with the three datasets (13, 30, and 108 features respectively), similarly classified points are not necessarily close to each other in euclidean space, potentially explaining the worse performance of the KNN classifiers in this study. Furthermore, decision trees are a less complex model than the other models evaluated in this study, so they might tend to perform worse in this higher dimensional setting as well.

However, no clear conclusion can be made on the best or worst classifiers, as the performances of the classifiers are greatly varied, and differ for each dataset. It appears that the context of the data in which a classifier is applied affects the performance metrics more than the classifiers themselves.

In order to more closely observe the effect of our partitions, we calculate and display the average test accuracy for each classifier across all datasets per partition in *Table 5*. From this table it is clear that as the proportion of training data decreases, the performance on the test dataset tends to decrease. This is not surprising, as the more data a model is trained on, the more information there is to guide its fit to the data, leading to better performance. Moreover, the hyperparameters differ as a result of different partitions, showing that the amount of training data directly influences the mathematical algorithms underlying the models.

Table 1. Performance metrics of each classifier by partition on the adult income dataset (47621 samples, 108 features)

Classifier	Train/Test Split	Best Hyperparameter	Validation Accuracy	Training Accuracy	Testing Accuracy
DT	80:20	Max Depth = 9	0.856	0.862	0.853
DT	50:50	Max Depth = 7	0.856	0.861	0.852
DT	20:80	Max Depth = 7	0.850	0.860	0.854
RF	**80:20**	**Max Depth = 18**	**0.862**	**0.896**	**0.862**
RF	50:50	Max Depth = 19	0.861	0.914	0.858
RF	20:80	Max Depth = 18	0.857	0.935	0.859
LR	80:20	C = 1	0.851	0.852	0.847
LR	50:50	C = 0.1	0.852	0.853	0.849
LR	20:80	C = 0.1	0.846	0.851	0.844
ANN	80:20	Layer Size = 4	0.851	0.855	0.847
ANN	50:50	Layer Size = 4	0.848	0.859	0.846
ANN	20:80	Layer Size = 4	0.841	0.863	0.845
KNN	80:20	K = 1905	0.825	0.830	0.827
KNN	50:50	K = 1905	0.807	0.826	0.820
KNN	20:80	K = 1	0.767	1.000	0.784

Table 2. Performance metrics of each classifier by partition on the heart disease dataset (297 samples, 13 features)

Classifier	Train/Test Split	Best Hyperparameter	Validation Accuracy	Training Accuracy	Testing Accuracy
DT	80:20	Max Depth = 4	0.789	0.916	0.772
DT	50:50	Max Depth = 17	0.784	1.000	0.750
DT	20:80	Max Depth = 1	0.695	0.779	0.697
RF	80:20	Max Depth = 18	0.848	1.000	0.819
RF	50:50	Max Depth = 10	0.858	1.000	0.785
RF	20:80	Max Depth = 16	0.847	1.000	0.774
LR	**80:20**	**C = 0.01**	**0.813**	**0.847**	**0.840**
LR	50:50	C = 1	0.858	0.865	0.835
LR	20:80	C = 0.01	0.844	0.857	0.832
ANN	80:20	Layer Size = 16	0.835	0.916	0.799
ANN	50:50	Layer Size = 16	0.825	0.899	0.732
ANN	20:80	Layer Size = 8	0.797	0.834	0.798
KNN	80:20	K = 1	0.770	1.000	0.798
KNN	50:50	K = 1	0.804	1.000	0.785
KNN	20:80	K = 1	0.695	1.000	0.687

Table 3. Performance metrics of each classifier by partition on the breast cancer dataset (569 samples, 30 features)

Classifier	Train/Test Split	Best Hyperparameter	Validation Accuracy	Training Accuracy	Testing Accuracy
DT	80:20	Max Depth = 3	0.929	0.975	0.938
DT	50:50	Max Depth = 3	0.943	1.000	0.919
DT	20:80	Max Depth = 1	0.955	1.000	0.918
RF	80:20	Max Depth = 17	0.956	1.000	0.947
RF	50:50	Max Depth = 10	0.858	1.000	0.942
RF	20:80	Max Depth = 4	0.973	1.000	0.932
LR	**80:20**	**C = 1**	**0.980**	**0.986**	**0.973**
LR	50:50	C = 1	0.968	0.989	0.972
LR	20:80	C = 1	0.955	1.000	0.973
ANN	**80:20**	**Layer Size = 16**	**0.835**	**0.993**	**0.973**
ANN	50:50	Layer Size = 16	0.964	0.996	0.958
ANN	20:80	Layer Size = 8	0.964	0.964	0.951
KNN	80:20	K = 1	0.960	1.000	0.929
KNN	50:50	K = 1	0.964	1.000	0.931
KNN	20:80	K = 1	0.946	1.000	0.929

Table 4. Results of performances for each classifier over the three datasets

Classifier	1st	2nd	3rd	4th	5th
Decision Tree	0	1	0	1	1
Random Forest	1	1	1	0	0
Logistic Regression	2	0	1	0	0
ANN	0	1	1	1	0
KNN	0	0	0	1	2

Table 5. Average testing accuracy by training-testing partition (averaged over three datasets)

Classifier	80:20 Partition	50:50 Partition	20:80 Partition
Decision Tree	0.854	0.840	0.823
Random Forest	0.876	0.862	0.855
Logistic Regression	0.887	0.885	0.883
ANN	0.873	0.845	0.865
KNN	0.851	0.845	0.800

Conclusion

There are many different methods of classification, and machine learning algorithms are becoming increasingly accessible and feasible to implement in the face of new technology and better data collection. However, our study implies that no method is an absolute top performer for any dataset. Different models will perform very differently across different datasets, even with the same data processing methods, as seen with the five models compared in this study. While logistic regression and random forests are generally the top performers, with KNN classifiers and decision trees lagging behind, the results are too varied for an absolute conclusion of a best classifier, and we can only observe that logistic regression and random forests tend to be the most consistent of the five classifiers studied. Looking at logistic regression and random forests, random forests appear to have greater performance on larger, high dimensionality datasets, such as the adult income set while logistic regression is best for smaller datasets. If this study were to be replicated, a greater amount of empirical data (i.e. performance metrics on a greater amount of datasets) would be desired in order to reach a more specific conclusion generalizable outside of the datasets used in this paper. What is certain is that a greater proportion of training data to testing data is desirable, and more training data will lead to a greater accuracy in the face of unseen data.

References

Becker, B. & Kohavi, R. (1996, April 30). *UCI Machine Learning Repository. UCI Machine Learning Repository.* University of California, Irvine. 10.24432/C5XW20

Caruana, R., & Niculescu-Mizil, A. (2006). *An Empirical Comparison of Supervised Learning Algorithms.* International Conference on Machine Learning.

Janosi, A., Steinbrunn, W., Pfisterer, M., Detrano, R. (1998, June 30). *UCI Machine Learning Repository. UCI Machine Learning Repository.* University of California, Irvine. 10.24432/C52P4X

Wolberg, W., Mangasarian, O., Street, N., Street, W. (1998, June 30). *UCI Machine Learning Repository. UCI Machine Learning Repository.* University of California, Irvine. 10.24432/C52P4X