Rodrigo Efraim
December 15, 2018

Test Accuracy of Supervised Classifiers

Abstract

Thorough empirical evaluations of supervised machine learning algorithms have been undertaken by Caruna and Niculescu-Mizil in 2006. The purpose of this project is to try to replicate the work of Caruna and Niculescu-Mizil in a smaller scale by comparing the test accuracies of supervised classification algorithms on a couple of datasets.

Introduction

The research paper written by Caruna and Niculescu-Mizil was done because they were aware that comprehensive empirical evaluations of supervised learning algorithms was lacking in the past decade. They went about their experiment by doing large scale empirical comparisons between ten supervised learning classifiers. They tested these ten classifiers on 11 datasets under eight different metrics. The ten supervised learning methods they chose are SVMs, neural nets, logistic regression, naive bayes, memory-based learning, random forests, decision trees, bagged trees, boosted trees, and boosted stumps. The eight performance metrics used are accuracy, F-score, Left, ROC area, average precision, precision/recall break-even point, squared error, and cross-entropy. One of the main findings was that no one algorithm will outperform all other algorithms on every metric. This finding supports the No Free Lunch Theorem, which suggest there is no universally best learning algorithm. However, there were some algorithms that had excellent performance on all eight metrics. Calibrated boosted trees were the best learning algorithms overall, followed by random forests in second, followed by uncalibrated bagged trees, calibrated SVMs, and un-calibrated neural nets [1]. The models that performed poorest were naïve Bayes, logistic regression, decision trees, and boosted stumps [1].

This experiment is a lighter replication to the work they did, by applying similar methods to new datasets which are not included in their papers. This experiment looks at three of the ten supervised learning algorithms used in the original project. The algorithms selected are SVM, Decision Tree, and Random Forest. These were selected based on the results of the original paper, in which Random Forest has best, SVM has neutral, and Decision Tree has worst performance.

Methods

Supervised Learning Algorithms

This section gives a brief description on each of the classifiers selected for this project. All the algorithms utilized are implementations from the Scikit learn API [ref. 2] [ref. 3] [ref 4]. For each classifier, only one of its respective hyperparameters was tuned. The hyperparameters for each of these algorithms were optimized by passing a CV value of 10 for the grid search cross validation function.

Support Vector Machine (SVM): It's a supervised machine learning algorithm which can be used for both classification or regression problems. But it is usually used for classification. Given 2 or more labeled classes of data, it acts as a discriminative classifier, formally defined by an optimal hyperplane that separates all the classes [ref. 2]. The linear kernel SVM was chosen for this project. The hyper parameter tuned passed into GridSearchCV is C, which is a penalty parameter of the error term. The values passed for this hyperparameter are: [10**-5, 10**-4, 10**-3, 10**-2, 10**-1, 1, 10, 100].

Decision Tree: A non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features [ref. 3]. The decision tree classifier with an entropy criterion was used for this project. The hyper parameter passed into GridSearchCV is max_depth, which represents the maximum depth of the tree. The values passed for the hyperparameter are: [1,2,3,4,5].

Random Forests: A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement if bootstrap=True (default) [ref. 4]. The hyper parameter passed into GridSearchCV is max_depth, which represents the maximum depth of the tree. The values passed for the hyperparameter are: [1,2,3,4,5].

Datasets

The three supervised machine learning algorithms described above were used on three datasets. All of these datasets were obtained from the UCI Machine Learning Repository [ref. 6].

Seeds Data Set: Contains measurements of geometrical properties of kernels belonging to three different varieties of wheat. A soft X-ray technique and GRAINS package were used to construct all seven, real-valued attributes which include; area, perimeter, compactness, length of Kernel, width of Kernel, asymmetry coefficient, and the length of kernel groove. In addition to the feature column, there is the Y column that displays the type of wheat. This dataset contains 211 rows. The shape of this dataset is 211 rows by 8 columns. There are three types of categories in the Y column; in which 71 rows belong to type 1, 70 rows belong to type 2, and 70 rows belong to type 3 . This is a perfect distribution of rows belonging to the 3 types. This data has zero null values. All values from the feature columns are numerical.

Wine Data Set: This data is the result of a chemical analysis of wines grown in the same region in Italy but derived from 3 different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines. The 13 attributes are alcohol, malic acid, ash, alcalinity of ash, magnesium, total phenols, flavanoids, nonflavanoid phenols, proanthocyanins, color intensity, hue, OD280/OD315 of diluted wines, proline. The shape of this dataset is 177 rows by 14 columns. There are three types of categories in the Y column, in which 58 rows belong to type 1, 71 rows belong to type 2, and 48 rows belong to type 3. This data has zero null values. All values from the feature columns are numerical.

EEG Eye State Data Set: The data set consists of 14 EEG values and a value indicating the eye state. Eye state equal to '1' indicates the eye is closed and '0' indicates the eye is open. The shape of this dataset is 14980 rows by 15 columns. There are 2 types of categories in the Y column, in which 8257 rows belong to class 0 (eye open) and 6723 rows belong to to class 1 (eye shut). This data has zero null values. All values from the feature columns are numerical.

Dataset Name	# of Rows	# of Features	# of Labels in Y
Seeds	211	7	3
Wine	177	13	3
EEG Eye State	14980	14	2

Experiment

There are 4 main components to this experiment. The first component are the 3 datasets. The second component are the 3 classifiers. The third component are the 3 partitions. The fourth component is the number of trials, in where each trial is done on a randomized shuffle of the original data. When described in a sequential structure it is as follows; 3 datasets, each undergoing 3 classifiers, each classifier consists of 3 partitions and each partition consists of three runs. In which each of these 3 runs is essentially a shuffle of the original dataset.

The hyperparameters for each of these classifiers were optimized by passing a CV value of 10 for the grid search cross validation function. The grid search implementation used is from Scikit-Learn's GridSearchCV function. The hyperparameters that produced the greatest performance on CV were selected for the models that would be used on the test set. FIG. [1-2] is an example occurring in the 3rd run of a (80% train, 20% test) partition of the Wine data of how a list of max_depth values, max_depth being a hyperparameter for the Random Forest classifier, is passed into the GridSearchCV function. Note that the values passed to this hyperparameter are [1,2,3,4,5], so GridSearchCV gives the results of [FIG. 1], which are the training accuracies and [FIG. 2], which are the validation accuracies, in correspondence to each of the max_depth values that have been passed in.

The original data is split into training and testing sets. There are three ways in how the data is split, these types of partitions are, (80% train, 20% test), (50% train, 50% test), and (20% train, 80% test). The accuracies of each classifier on the three partitions can be seen in the graphs below [FIG. 3-14].

A notable observation that happens with all of the three datasets is that as you decrement the train percentage and increment the testing percentage, the test accuracies for all the classifiers decreases.

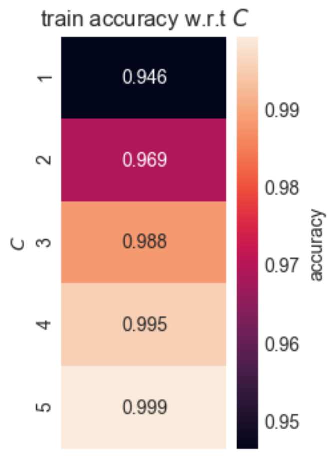
When performing the supervised learning algorithms on the wine dataset (177 rows, 13features, 3 labels) Random Forest turned out to be the best in the (80% train, 20% test) partition with an excellent average test accuracy of 100%. However when looking at the (50% train, 50 test) partition and (20% train, 80% test) partition the SVN beats Random Forest by a small percentage margin of 2.5% . The Decision Tree performed well, but fell just a few percentages behind the other two classifiers. The worst average test accuracy was produced by the Decision Tree classifier in the (20% train, 80% test) partition, scoring 86%. [FIG. 6]

When performing the supervised learning algorithms on the seeds dataset (211 rows, 7 features, 3 labels) SVM turned out to have the better average test accuracy than the other 2 classifiers throughout all 3 partitions. The Decision Tree and Random Forest performed similar to each other, falling slightly behind the average test accuracy percentage of SVM. The worst average test accuracy was produced by the Random Forest classifier in the (20% train, 80% test) partition, scoring 85%. [FIG. 10]

When performing the supervised learning algorithms on the EEG Eye State dataset (211 rows, 7 features, 3 labels) the Random Forest classifier turned out to have the better average test accuracy than the other 2 classifiers throughout all 3 partitions. The SVM and Decision Tree performed similar to each other, falling behind of the Random Forest by an rounded difference of 6%. The worst average test accuracies were produced by SVN and Decision Tree in the (20% train, 80% test) partition, scoring 64%. [FIG. 14]

Overall, the Random Forest classifier got the best test accuracy results in 2 out of 3 datasets. SVM got the best test accuracy in 1 out of 3 datasets.

Overall, out of the 3 datasets, Random Forest had the best test accuracies for 2 of the datasets (Wine) and (EEG Eye State) [FIG. 4 & 12]. SVM had the best test accuracy in one of the datasets (Seeds) [FIG. 8]. Decision Tree was always either last or fell in between the other two classifiers.



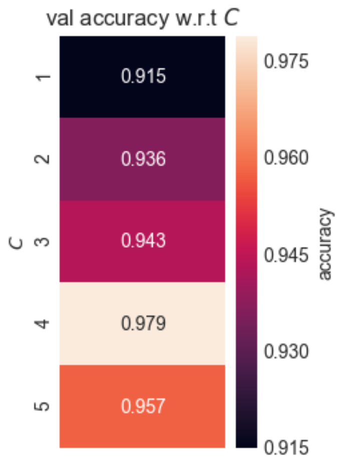


Figure 1 Figure 2

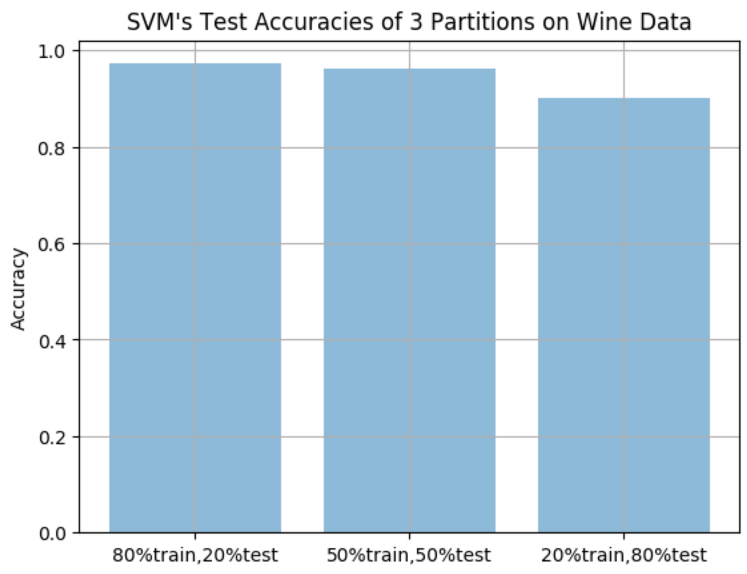


Figure 3

Partitions	1st Run	2nd Run	3rd Run	Average
80% train, 20% test	0.9722	0.9722	0.9722	0.9722
50% train, 50% test	0.9663	0.9551	0.9663	0.9625
20% train, 80% test	0.8592	0.8803	0.9663	0.9019

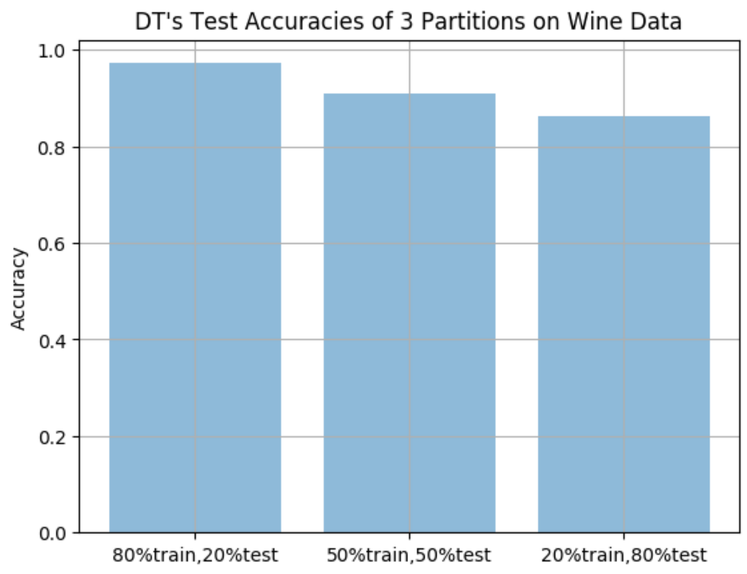


Figure 4

Partitions	1st Run	2nd Run	3rd Run	Average
80% train, 20% test	1	0.9166	1	0.9722
50% train, 50% test	0.8989	0.8764	0.9551	0.9101
20% train, 80% test	0.8310	0.9014	0.8521	0.8615

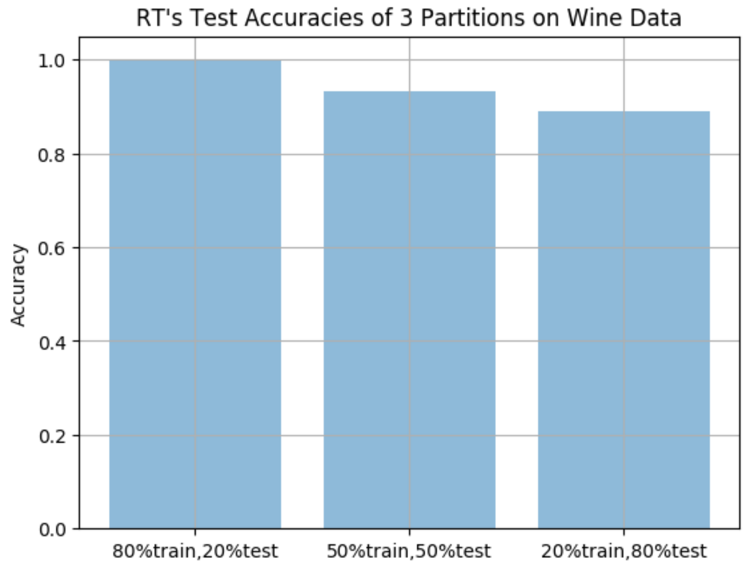


Figure 5

Partitions	1st Run	2nd Run	3rd Run	Average
80% train, 20% test	1	1	1	1
50% train, 50% test	0.9551	0.8764	0.9663	0.9326
20% train, 80% test	0.8943	0.8873	0.8873	0.8897

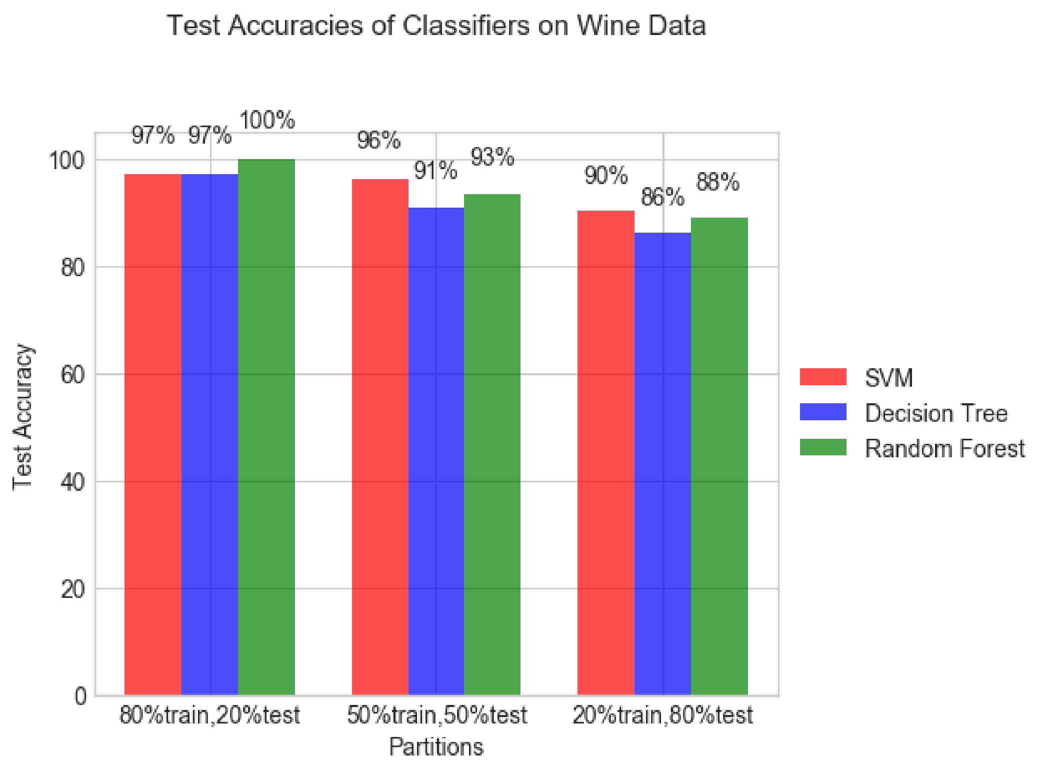


Figure 6

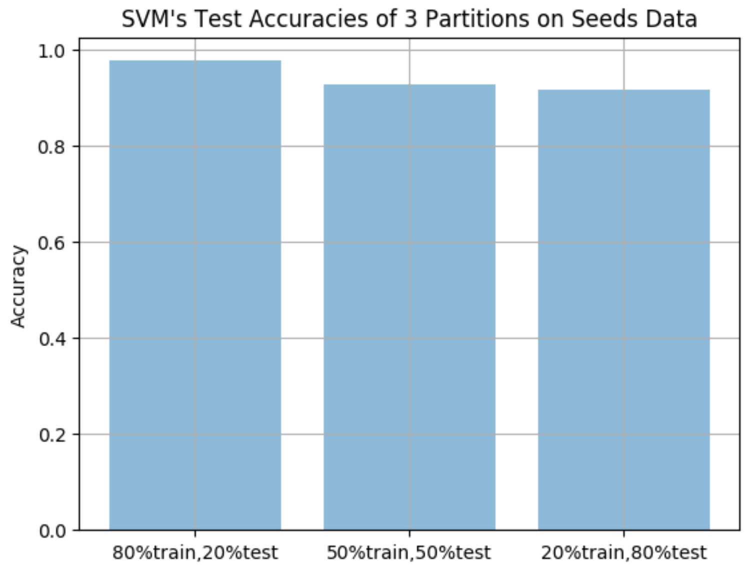


Figure 7

Partitions	1st Run	2nd Run	3rd Run	Average
80% train, 20% test	0.9767	0.9535	1	0.9767
50% train, 50% test	0.9151	0.9339	0.9339	0.9277
20% train, 80% test	0.9467	0.8698	0.9339	0.9168

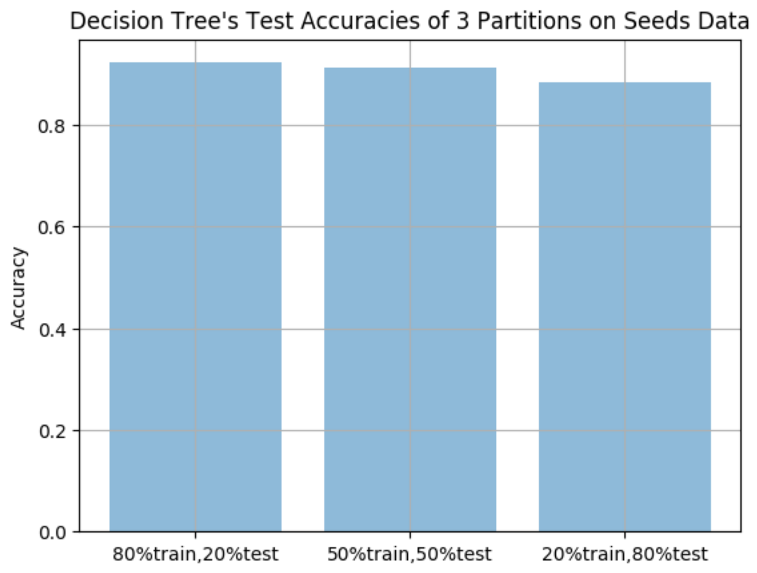


Figure 8

Partitions	1st Run	2nd Run	3rd Run	Average
80% train, 20% test	0.9302	0.9302	0.9069	0.9225
50% train, 50% test	0.9056	0.9245	0.9056	0.9119
20% train, 80% test	0.9053	0.8225	0.9231	0.8836

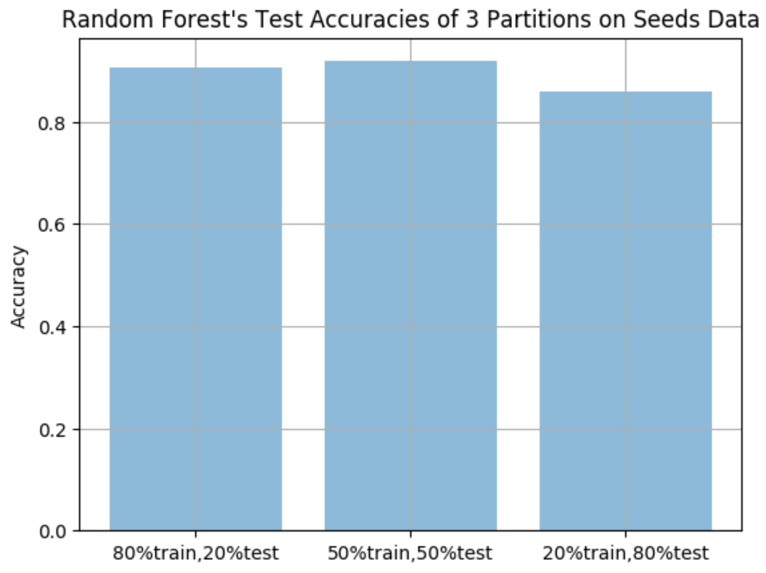


Figure 9

Partitions	1st Run	2nd Run	3rd Run	Average
80% train, 20% test	0.9069	0.8837	0.9302	0.9069
50% train, 50% test	0.9150	0.9339	0.9056	0.9182
20% train, 80% test	0.8521	0.8284	0.8994	0.8599

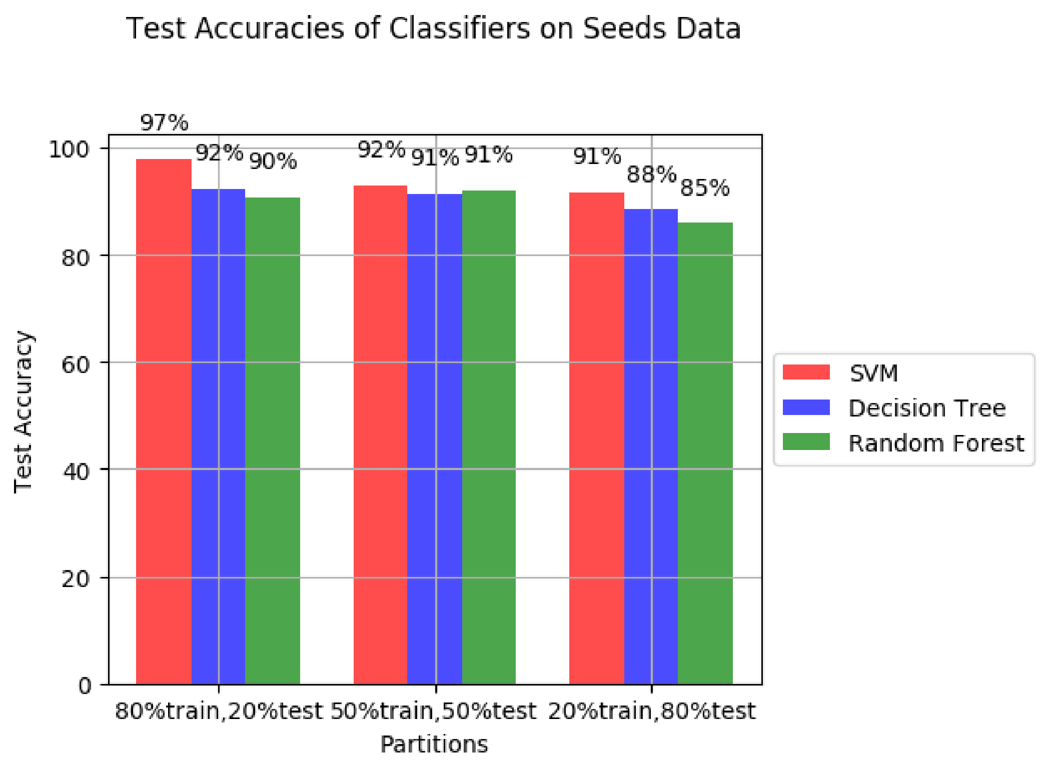


Figure 10

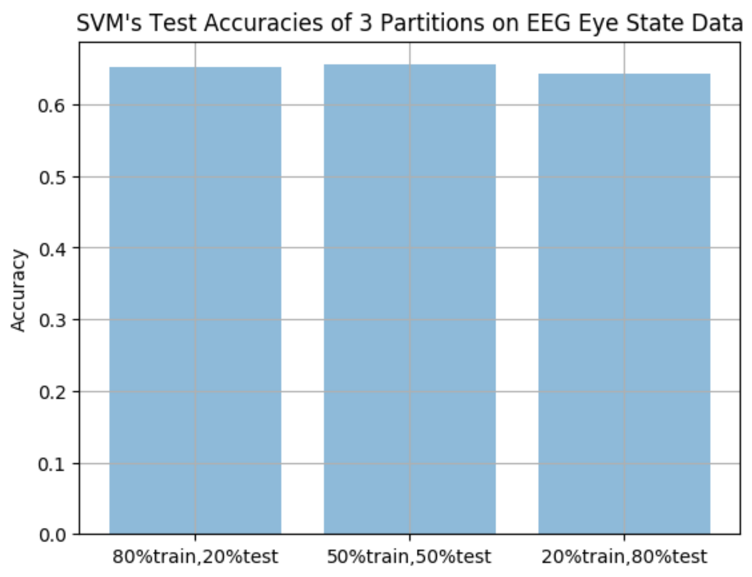


Figure 11

Partitions	1st Run	2nd Run	3rd Run	Average
80% train, 20% test	0.6248	0.6675	0.6622	0.6515
50% train, 50% test	0.6470	0.6652	0.6540	0.6554

20% train, 80% test	0.6441	0.6295	0.6540	0.6425

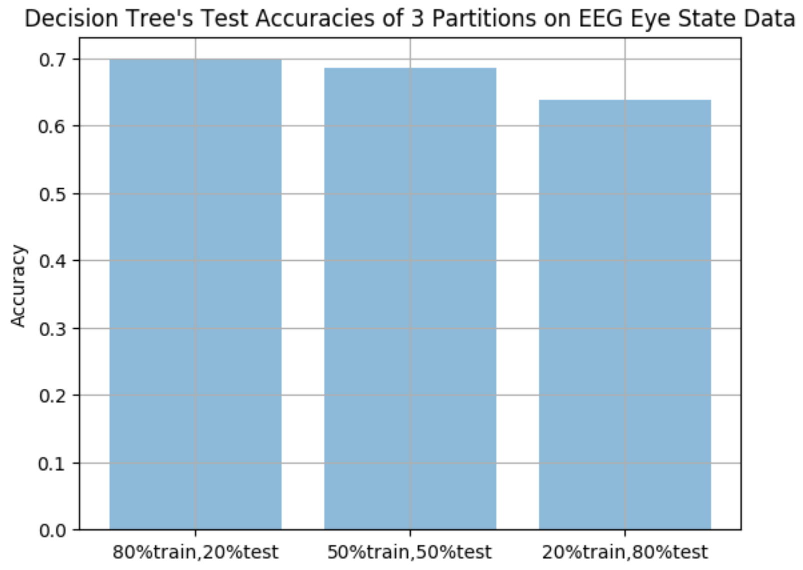


Figure 12

Partitions	1st Run	2nd Run	3rd Run	Average
80% train, 20% test	0.6915	0.6969	0.7022	0.6969
50% train, 50% test	0.7202	0.6593	0.6759	0.6851
20% train, 80% test	0.5921	0.6815	0.6428	0.6388

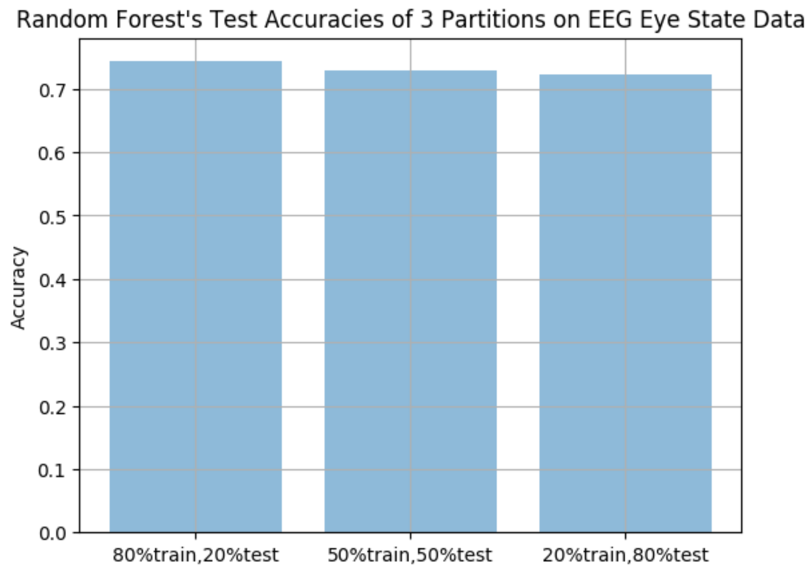


Figure 13

Partitions	1st Run	2nd Run	3rd Run	Average
80% train, 20% test	0.7489	0.7448	0.7356	0.7432
50% train, 50% test	0.7298	0.7367	0.7234	0.7300

20% train, 80% test	0.7386	0.7179	0.7142	0.7236

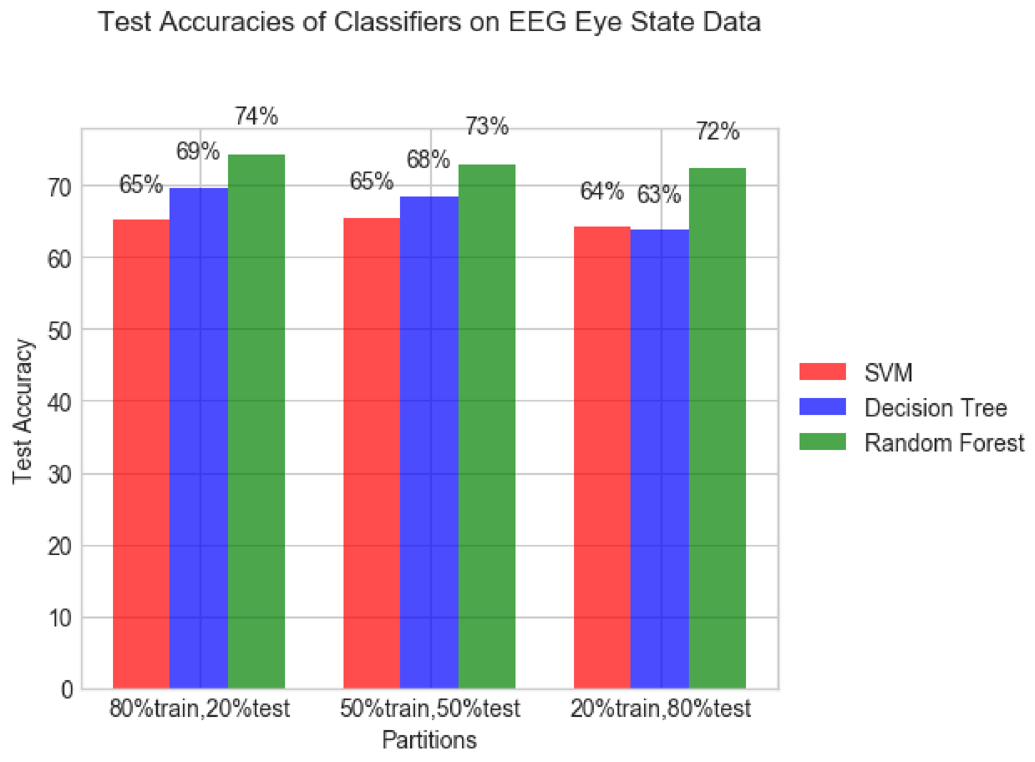


Figure 14

Conclusion

The results demonstrated by Caruna and Niculescu-Mizil show that the Random Forest classifier turned out to have one of the best performances in relation to test accuracy and that Decision Tree classifier was one of the worst. This analysis is consistent with the original research. Results show that the best average test accuracies on 2 out of 3 datasets is from Random Forest classifier. In this analysis, the SVM classifier performs average, by having the best average test accuracy on 1 out of 3 datasets.

References

[1] Caruana R. Niculescu-Mizil A. (2006, June). An empirical comparison of supervised learning algorithms. The Proceedings of the 23rd International Conference on Machine Learning (pp. 161-168). ACM.

[2] sklearn svm SVC

https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC

[3] sklearn Decision Tree Classifier
https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[4] sklearn Random Forest Classifier
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[5] Hyperparameter Tuning the Random Forest in Python
https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74

[6] UCI Machine Learning Repository
https://archive.ics.uci.edu/ml/datasets.html