Performance of Multiple Supervised Machine Learning Models Across Datasets

Broderick Higby [*1]

Abstract

I take an empirical look at how the Random Forest Classifier, Decision Tree Classifier, and Naive Bayes Classifier compare across three data sets provided by the UC Irvine Machine Learning Repository(Dheeru & Karra Taniskidou, 2017). This paper presents an updated analysis of these algorithms over a decade after Caruna (Caruana & Niculescu-mizil, 2006) performed his analysis on them. The model that performed the best was the random forest which is consistent with (Caruana & Niculescu-mizil, 2006). I also varied the size of the dataset and the ratio of the training and testing splits to test the consistency.

1. Introduction

Rich Caruna and Alexandru Niculescu Mizil approached the task of analyzing algorithms in a very thorough and extensive manner in order to see which ones perform the best. The No Free Lunch theorem (Wolpert & Macready, 1997) states that any two optimization algorithms are equivalent when their performance is averaged across all possible problems. The methods used in this project were chosen for their spread in average performance over what (Caruana & Niculescu-mizil, 2006) had in their 2006 paper.

2. Data Description

I originally analyzed three different datasets, that are consistent with (Caruana & Niculescu-mizil, 2006). The datasets come from the UC Irvine repository. The datasets I analyzed are the Adult Data Set (Kohavi & Becker, 1996), Wine (Aeberhard, 1991), Wine Classification Data Set (Cortez, 2009), Letter Recognition (Cortez, 2009), and the Cover type (Blackard & Anderson, 1998) Data Set.

[*]Equal contribution [1]Department of Cognitive Science, University of California, San Diego, USA. Correspondence to: Broderick Higby <bhigby@ucsd.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

2.1. Adult Data Set

The Adult Data Set consists of U.S. census income data that aims to predict whether income exceeds $50,000. Every categorical variable has been converted into dummy variables. It contains 32, 561 with 14 variables six of them are continuous and eight are categorical. When theyre converted to dummy variables it increases to 107.

2.2. Wine Data Set

The Wine Data Set is a Data Set that uses chemical analysis to determine the origin of wines. This data set is extremely small with only 178 instances and 13 attributes. The attributes are continuous real numbers

2.3. Wine Quality Data Set

The Wine Quality Data Set consists of a set of data for both white and red wine. The data set models wine quality from Portugal based on physio chemical tests. I decided to combine the red and white wine data sets in order to have a wider range of data. The data set contains 4898 instances and 12 attributes.

2.4. Letter Recognition Data Set

The Letter Recognition Data Set is a database of character image features with an objective of trying to identify the letter given a black and white image. In the letter dataset the letter O Is treated as positive and the rest of the alphabet is negative, in order to give a more balanced analysis.

2.5. Cover Type Data Set

The Cover type Data Set predicts the forest type from cartographic variables given observations from the U.S. Forest Service. This Data Set was converted to a binary classification problem by treating the most commonly occurring covers as positive and the rest as negative. It contains 581,012 data points with 54 attributes that were used for analysis. The choice to expand from three datasets to five was to see how much variance the machine learning models I chose have when given different types of information. The Wine and Letter Recognition datasets had issues with

	Adult 80/20 Train	Adult 50/50 Train	Adult 20/80 Train
Naive Bayes	0.823	0.824	0.819
Random Forest	0.856	0.86	0.851
Decision Tree	0.805	0.815	0.808
	Wine 80/20 Train	Wine 50/50 Train	Wine 20/80 Train
Naive Bayes	1	0.824	0.819
Random Forest	1	0.86	0.851
Decision Tree	0.944	0.815	0.808
	Wine Classification 80/20 Train	Wine Classification 50/50 Train	Wine Classification 20/80 Train
Naive Bayes	0.152	0.15	0.19
Random Forest	0.951	0.952	0.953
Decision Tree	0.895	0.902	0.89
	Letter Recognition 80/20 Train	Letter Recognition 50/50 Train	Letter Recognition 20/80 Train
Naive Bayes	0.0312	0.034	0.027
Random Forest	0.057	0.057	0.056
Decision Tree	0.049	0.052	0.05
	Forest Covertype 80/20 Train	Forest Covertype 50/50 Train	Forest Covertype 20/80 Train
Naive Bayes	0.605	0.62	0.678
Random Forest	0.822	0.802	0.772
Decision Tree	0.745	0.712	0.669

Table 1. Training dataset cross-validation scores and accuracy across the different datasets using Naive Bayes, Random Forest, and Decision Trees

splitting during cross validation. Each dataset had so few target values when the minimum number of trees needed to be at least 5.

3. Problem Description

In (Caruana & Niculescu-mizil, 2006), the main goal of the paper is to test the robustness of the various algorithms in an effort to provide a more analytical perspective for which algorithm to use for the best results. They explicitly say that some models that perform well on some metrics (such as random forests or boosted trees) will not perform well on all metrics, in some cases such as the Adult dataset, boosted trees (the overall top performer) performed worse compared to bagged trees or random forests. In this paper, I will be analyzing a smaller subset of the models and datasets that (Caruana & Niculescu-mizil, 2006) did using standard libraries that werent available in 2006 when the paper was written. This paper will also be adding some new comparisons for the datasets that will show the generalizability of each algorithm.

4. Methods

The classification methods I chose were written in Python programming language in Anacondas Jupyter Notebook which allows the user to see the results of the code without having to recompile the entire file. The classifiers I used in this paper were the Naive Bayes, Decision Tree, and Random Forest.

4.1. Naive Bayes Classifier

The Nave Bayes Classifier is a very simple probabilistic classifier that loosely follows how humans think. When making a classification decision (i.e. whether a letter is an O or not), it factors in priors. A positive attribute of Nave Bayes classifiers is that theyre highly scalable and require only a number of parameters that coincide linearly with the number of variables. This paper uses a normally distributed Gaussian, the original intention was to use a kernel estimation. However, its accuracy score is nominal across the datasets (not statistically significant) and its implementation increased runtime, so the Gaussian process was substituted in its place.

4.2. Decision Tree Classifier

The Decision Tree Classifier is a non-parametric method that predicts the target variable value by learning simple rules that are inferred from decision features. A good characteristic of decision trees is that they require little data preparation, and its runtime is logarithmic with the number of data points that are used to train the tree. Adjusting the parameters, such as max depth increased our accuracy with our highest accuracy being attained at a max depth of 5.

	Adult 80/20 Test	Adult 50/50 Test	Adult 20/80 Test
Naive Bayes	0.824	0.825	0.835
Random Forest	0.86	0.86	0.868
Decision Tree	0.819	0.815	0.816
	Wine 80/20 Test	Wine 50/50 Test	Wine 20/80 Test
Naive Bayes	0.966	0.941	0.857
Random Forest	0.964	0.95	1
Decision Tree	0.931	0.95	0.875
	Wine Classification 80/20 Test	Wine Classification 50/50 Test	Wine Classification 20/80 Test
Naive Bayes	0.161	0.16	0.222
Random Forest	0.956	0.96	0.966
Decision Tree	0.904	0.899	0.9
	Letter Recognition 80/20 Test	Letter Recognition 50/50 Test	Letter Recognition 20/80 Test
Naive Bayes	0.037	0.031	0.03
Random Forest	0.06	0.061	0.06
Decision Tree	0.055	0.051	0.059
	Forest Covertype 80/20 Test	Forest Covertype 50/50 Test	Forest Covertype 20/80 Test
Naive Bayes	0.638	0.673	0.692
Random Forest	0.831	0.822	0.8
Decision Tree	0.741	0.717	0.705

Table 2. Test results across the different datasets using Naive Bayes, Random Forest, and Decision Trees

4.3. Random Forest Classifier

The Random Forest Classifier is a type of ensemble learning, which like a orchestral ensemble that combines a number of instruments, the random forest classifier combines a number of different algorithms. This classifier constructs a mass of decision trees during training and this outputs the class that is the mode of the classes (in the case of classification). An advantage to random forests is that they offer a very robust option across datasets that produces human readable models. They work on the principal of the wisdom of the crowds which is simply that a lot of heads put together is better than just one when it comes to trying to solve a problem. Deeper trees almost always perform better (with diminishing returns) with respect to requiring more trees for increased performance. This is because they operate on the bias-variance trade off which is that deeper trees reduce bias, while more trees reduce variance. In order to perform more consistently with (Caruana & Niculescu-mizil, 2006), I decided to have 1024 estimators as well.

4.4. Algorithm

The algorithm for evaluating the datasets can be seen in the Algorithm below. In order to save on computation time as well as ensure that I am comparing efficiency, I instantiate the three machine learning models in the same place, then split based on an 80% training and validation split to a 20% test set split, then a 50% training and validation set to 50% test set split, and finally a 20% training and validation set to 80% test set split. I enumerate through the models, fitting them, then immediately getting their 5-fold cross validation. I then predict based off the X test set and report each individual classifier accuracy score.

Algorithm 1 Classification Function

Input: dataset $features, target$
for dataset split based on test set: $.20, .5, .8$ **do**
  split dataset $train{-}xval, test{-}x, train{-}valy, test_y$
  for clf **in** $classifier$ **do**
    fit x_i and y_i
    predict x_i and y_i
    training set co-variance score
    test set co-variance score
  end for
end for

5. Results

The random forest performed with the highest accuracy and the Nave Bayes classifier had consistently the worst performance which is consistent with (Caruana & Niculescu-mizil, 2006). The datasets that had the highest accuracy were the Wine Data Set, then the Wine Classification Data Set. The three classifiers performed worst on the Letter Recognition Data Set. The random forest is the most consistent model with the highest accuracy across the datasets. Increasing the Data Set consistently returns a higher accuracy score which is consistent with (Caruana & Niculescu-mizil, 2006) as well as intuitive.

Discussion

The results were consistent with (Caruana & Niculescu-mizil, 2006). The premise of this paper is a focus on the algorithms with as little bias in each data set as possible, whether its data snooping where a researcher picks a model after seeing the features or picking only relevant features from the dataset while disregarding others. The more features a dataset has, the better its performance. Random forest classifiers performed really well overall which due in part to the wisdom of the crowds. Future work might include the use of other newer machine learning techniques or ones that were not used in the original paper such as Markov models and deep neural networks. The introduction of cloud computing makes testing more models a reasonable proposition.

Extra

This paper extends beyond the datasets used in the paper as well as implements the algorithm for computing the various machine learning models in a computationally efficient manner. The resulting paper is professionally written in a manner consistent with the

References

Aeberhard, Stefan. UCI wine data set, 1991. URL http://archive.ics.uci.edu/datasets/Wine.

Blackard, Jock, Dean Denis and Anderson, Charles. UCI covertype data set, 1998. URL https://archive.ics.uci.edu/ml/datasets/Covertype.

Caruana, Rich and Niculescu-mizil, Alexandru. An empirical comparison of supervised learning algorithms. In *In Proc. 23 rd Intl. Conf. Machine learning (ICML06*, pp. 161–168, 2006.

Cortez, Paulo. UCI wine quality data set, 2009. URL https://archive.ics.uci.edu/ml/datasets/Wine+Quality.

Dheeru, Dua and Karra Taniskidou, Efi. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Kohavi, Ronny and Becker, Barry. UCI adult data set, 1996. URL http://archive.ics.uci.edu/datasets/Adult.

Wolpert, David H. and Macready, William G. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.