Supervised Machine Learning Algorithms on Classification problems

Calixto Calangi
ccalangi@ucsd.edu

Abstract

This paper seeks to extend upon previous studies on supervised machine learning algorithms, with an emphasis on solving classification problems. Work was done utilizing several supervised learning methods, including: decision trees, random forests, bagging classifier, and SVM. Such methods were tried on several datasets that ranged in their features, targets, and sizes. The metrics were how well the algorithms performed in terms of classification accuracy. Results showed some similarity with previous studies, with some variation in the ranking of how well each algorithm did. Some points for future research would be to look into larger datasets consisting of more features as well as a more diverse spread of points.

1. Introduction

This project follows in a similar fashion to an empirical comparison study (Caruana & Niculescu-Mizil 2006), with an emphasis on using the datasets for classification. As stated before, this is accomplished via the use of supervised machine learning algorithms on a series of datasets. The datasets explored in this project were taken from the UC Irvine Machine Learning repository (http://archive.ics.uci.edu/). The four datasets explored include: Glass Identification (German, 1987), Heart Failure Clinical Records (2020), Wireless Indoor Localization (Bhatt, 2017), and Algerian Forest Fires (Faroudja, 2019).

Building upon Caruana & Niculescu-Mizil, this paper reports the classification accuracy across the various datasets and models trained. The main metric, as stated, will be the classification accuracy: the machine learning algorithms' hit rate on each dataset. By focusing on the accuracy, this project tackles the classification problem via various two-class classifiers.

The project evaluates the classification accuracy through the performance of supervised machine learning algorithms. Decision trees, random forests, bagging classifiers, and SVM explore the four data sets, where each are tracked across the experiment to identify how well they perform, i.e. how well they are able to tackle the problem of classification.

2. Methodology

2.1. Algorithms

As aforementioned, four supervised machine learning algorithms are used for the purposes of the classification problem. They were tested and modeled across four datasets, utilizing training data, data partitions, test data to train models, creating the best possible model, and finally, basing the algorithms' performance on the test accuracy across three trials performed for each dataset, each classifier, and partition.

The primary task was binary classification, splitting the datasets' targets into 0's and 1's, and the algorithms' best attempts at correctly classifying being represented via the test accuracy. As for the partitions, various splits (ranging from 80%/20%, 50%/50%, and 20%/80% training-to-test data splits) to look for how well the models perform with more-or-less training data to then evaluate on the remaining test data splits.

Cross-validation was also included, so as to help further subset the data and find the most optimal parameters that would be used on the best model for each algorithm.

The algorithms, collected data, and many of the visualizations were implemented through the use of Python and various libraries including: scikit-learn, seaborn, NumPy Matplotlib, SciPy and pandas.

Decision Tree: Decision trees were used, exploring the max depth and optimal max depth across values of 1,2,3,4, and 5.

Random Forest: Random forests were used, exploring the max depth and optimal max depth across the same values, 1,2,3,4, and 5.

Bagging Classifier: Bagging family classifiers with decision tree as the estimators were used. n=20 estimators were used, exploring across a list of max depth values consisting of 1,2,3,4,5,6,7,8,9,10,11, and 12.

SVM: SVM with RBF kernel was used, exploring gamma parameters across 1e-3, 1e-2, 1e-1, 1, 10 and C, or regularization parameters, parameters 1e-3, 1e-2, 1e-1, 1, 10, 100, 1000.

Across all classifiers, k-fold cross validation was used to best find the optimal hyperparameters across all classifiers.

2.2. Main metric: accuracy

The main metric across the classifiers was the classification accuracy, which was upon the collected test accuracy across the various algorithms' and best models' performance on the datasets. This is the primary answer to the classification problem we have been tackling all quarter long.

To start with, training data was used with cross-validation methods on the classifier to best pick the correct parameters and have the best model assembled for the purposes of training the final test data splits (from the original partitions). The test accuracy was then collected, averaged across each split and an overall mean test accuracy was calculated.

2.3. First dataset: Glass Identification

The glass identification dataset dealt with the binary classification task of identifying float processed glass from non-float processed glass. Originally, the dataset was set with seven labels as the original targets; the glasses were split into various groups. Such groups included: building windows (float processed), building windows (non float processed), vehicle windows (float processed), vehicle windows (non float processed, of which there were none) containers, tableware, and headlamps.

The problem was with binary classification, splitting the data into two groups would be problematic; for instance, splitting into groups 1-4 and 5-7 for the glass would be provide an imbalanced dataset, so float versus non-float classification was the primary approach going forward.

With this re-classification, the dataset consisted of 176 instances (93 float to 70 non-float), and 11 features. The first 10 features were continuous values, consisting of several indexes/relevant information of the glass and glass composition: their refractive index, sodium, magnesium, aluminum, silicon, potassium, calcium, barium, and iron content. The last feature was reconfigured, as it originally labeled the groups into 7 glass groups; it was reconfigured into float (consisting of building windows and vehicle windows) and non-float (the non-float portion of vehicle windows).

The resulting binary classification task was then to identify the float versus non-float.

2.4. Second dataset: Heart Failure Clinical Records

The heart failure clinical record dataset dealt with the binary classification task of identifying of those who experienced heart failure, which patients passed away in the subsequent follow-up period. Unlike the previous dataset, no particular cleaning was conducted, as the balance in the survive versus not survive division did not warrant reconfiguring. As such, the binary classification task proceeded smoothly.

The dataset consisted of 299 instances, with 13 features. The first 12 features were various binary, continuous, and integer features: age (integer), presence of anaemia (binary), level of CPK enyzme in the blood via creatinine phoshpokinase (integer), presence of diabetes (binary), percentage of blood leaving the heart at each contraction (integer), the presence of high blood pressure (binary), platelets (continuous), serum creatinine in the blood (continuous), serum sodium (integer), sex (binary), whether they smoked (binary), and the follow up period (integer).

The last feature was the binary target of the classification task: identifying whether there was a death or no death associated.

2.5. Wireless Indoor Localization

The wireless localization dataset dealt with the binary classification task of identifying, based upon WiFi signal strength of smartphones, where the smartphone was located. Originally, the target was split into four possible locations for each smartphone. This was sorted into a binary classification task via spltting into the possible locations instead being in the first two rooms or not (thus being the other 2 rooms.)

The dataset consisted of 2,000 instances, with eight features. The first 7 all consisted of integer values, looking at the relative WiFi strength across 7 reported signals. The last feature was the binary target, reconfigured into either being in the first two rooms, or not in the first two rooms.

2.6. Algerian Forest Fires

The Algerian forest fires dataset dealt with the binary classification task of identifying whether there had been a forest fire or no forest fire occurrence across two regions in Algeria, Bejaia or Sidi-Bel Abbes. For the purposes of this task, the binary target wasn't set to identify whether it was in either in Bejaia or Sidi-Bel Abbes, but whether a forest fire was present or not.

Some data cleaning was necessary for this, as some of the data was not configured properly; the first two rows were supposed to be the headers and labels but was lost in translation (via the pandas reading csv function); nan values as well as 'classes' being a value for a row; and improper data types that had to resorted to properly proceed.

The dataset consisteed of 244 instances, with 14 features, dropping down to 237 instances and 13 features due to previous error; the 1st feature identified whether it was in Bejaia or Sidi-Bel Abbes, being dropped as it was deemed not relevant (simply wanting to identify the presence or absence of a forest fire). The first 3 features had already labels' year, month, and day, and thus were also removed as they were deemed irrelevant. The other features included

temperature, relative humidity, wind speed, rain, fine fuel moisture code (FFMC), duff mositure code (DMC), drought code (DC) initial spread index (ISI), buildup index (BUI), and fire weather index (FWI).

The binary target of the task was to identify whether there was or wasn't a forest fire.

Across all of the four classifiers and four datasets, 3 trials were conducted per dataset per partition. Averages of the test accuracy across partitions, and average test accuracy results for each partition, as well as the best reported accuracy and associated hyperparameter of the given classifier are as reported in the following section.

3. Results

Average test accuracy		Classifiers			
		Decision Tree	Random Forests	Bagging classifier	SVM w/RBF kernel
Datasets	Glass	**0.729**	**0.815**	**0.872**	**0.713**
	Heart failure	**0.845**	**0.777**	**0.800**	**0.693**
	WiFi local	**0.963**	**0.973**	**0.978**	**0.984**
	Forest fire	**0.976**	**0.980**	**0.990**	**0.938**

Figure 1. Average test accuracy

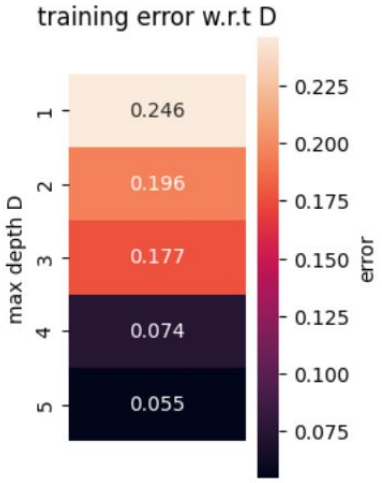
Average test accuracy per partition (80/20) (50/50) (20/80)		Classifiers			
		Decision Tree	Random Forests	Bagging classifier	SVM w/RBF kernel
Datasets	Glass	0.778	0.818	0.869	0.737
		0.691	0.813	0.833	0.707
		0.718	0.814	0.913	0.695
	Heart failure	0.861	0.75	0.739	0.706
		0.864	0.769	0.762	0.684
		0.810	0.813	0.899	0.689
	WiFi local	0.973	0.973	0.978	0.987
		0.964	0.974	0.976	0.984
		0.954	0.971	0.984	0.981
	Forest fire	0.986	0.986	1.0	0.952
		0.973	0.984	0.978	0.934
		0.968	0.969	0.993	0.926

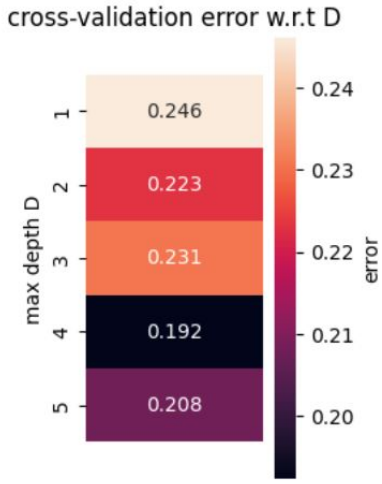
Figure 2. Average test accuracy across partitions

Best reported training accuracy w/ optimal hyperparameter		Classifiers			
		Decision Tree	Random Forests	Bagging classifier	SVM w/RBF kernel
Datasets	Glass	0.96875 D = 2	1.0 D = 4	1.0 D = 11	1.0 C=100 γ = 1
	Heart failure	0.983 D = 4	1.0 D = 5	1.0 D = 5	1.0 C=1 γ = 0.001
	WiFi local	0.99 D = 5	0.995 D = 5	0.999375 D = 9	0.998 C=1 γ = 0.01
	Forest fire	1.0 D = 4	1.0 D = 3	1.0 D = 6	1.0 C = 100 γ = 0.001

Figure 3. Best training accuracy w/ optimal hyperparameter

Looking at table 1, it reports the average test accuracy,






```
Best max depth D: 4
Test error: 0.18181818181818177
Train Accuracy: 0.9230769230769231
Validation Accuracy: 0.8
```

Figure 4. Best training accuracy w/ optimal hyperparameter

across the board on all classifiers and datasets. We can see that the bagging classifier with decision tree performs almost the best across the board, followed by random forests, decision tree, then SVM w/RBF kernel.

When comparing it to Caruana & Niculescu-Mizil, there is obviously variation; we can see how classifiers perform better on some of the data sets (the jump in performance for SVM with RBF kernel on the wireless localization dataset stands out).

There is also a difference in ranking, but overall, it trends in the same fashion regarding random forests and bagging classifier with decision tree as stand-out performers for model accuracy.

Table 2 lays out average test accuracy across all the split partitions. While again, there is variation in performance, a trend that resembles the study is the better pefromance with more training data. Meaning, the 80%/20% splits tend to report better test accuracy scores than in the 20%80%/ splits.

Table 3 reports the best training accuracy of the classifiers with the reported optimal hyperparameter.

Though not pictured in its entirety, for each individual dataset, there exists, for each classifier and partition, the included training error, validation error, and respective heatmaps tracking them as they go across the respective parameters.

4. Discussion

The general trends reiterate some common ground with the study from Caruana & Niculescu-Mizil, as we can see how the average test accuracy scores follow suit of the classifiers. For instance, random forests and bagging classifier with decision trees fare pretty well, with some variation between the decision tree and SVM w/RBF kernel performance across test accuracy scores.

Alongside this, the partition trends towards the same ground, as performance with more training data avaliable appears to indicate better test accuracy scores in the end.

However, one concern would be the in Table 3, as the reported best training accuracy scores across most of the partitions and classifiers trended towards 1.0. Within the classifier training and cross-validation (via kFold or GridSearchCV from scikit-learn), there shouldn't be overfitting in training accuracy as seen; tweaking with the parameters (i.e. setting the regularization parameters for SVM; max depth for decision tree and random forests) in different ranges didn't appear to sway it much without leading to significant error/drastic value changes due to only experiencing a smaller range of parameters.

The performance on the two datasets where most classifiers did superb on is also of question; namely, the wireless localization and forest fire reported test accuracies. Below, it can be seen how already, intuitively, there is some already 'shape' to the way the classes sort themselves, suggesting the data is not particularly challenging.

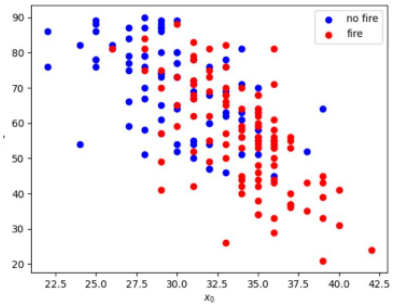


Figure 5. Visualization of datapoints of forest fire

5. Conclusion

The project has shown, to some extent, show reproducibility in some aspects of the performance of classifiers on untested data. General trends of classifiers to perform better than others, and showing variation in some classifiers doing well in some problem datas and less proficient in others.

Overall, in terms of ranking the classifiers, bagging classifier with decision trees perform the best, with random forests in second, decision tree followed by SVM w/RBF kernel.

Some points for future points of reinforcement is to see if such test accuracy scores are reproducible across many more datasets that contain substantially more datapoints, as three of the four datasets were less than 300 instances.

6. Bonus Points

Bonus points are requested for utilizing an extra classifier and dataset.

References

Bhatt, R. (2017). Wireless Indoor Localization [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C51880.

Caruana, R. & Niculescu-Mizil, A. (2006). An Empirical Comparison of Supervised Learning Algorithms. *Proceedings of the 23rd International Conference on Machine Learning*.

Faroudja, A. (2019). Algerian Forest Fires [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5KW4N.

German, B. (1987). Glass Identification [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5WW2P.

Heart Failure Clinical Records [Dataset]. (2020). UCI Machine Learning Repository. https://doi.org/10.24432/C5Z89R.