An Empirical Comparison of Multiclass-Agnostic Machine Learning Algorithms on Binary Tasks

Junwei Chen
Computer Science and Engineering
UC San Diego

Abstract

A number of classic machine learning algorithms have been studied in great detail over the last two decades. This study empirically compares the performance of three such algorithms that can be used for classification tasks but are unaware of the number of classes (multiclass-agnostic): K Nearest Neighbors, decision trees, and random forests. We are particularly interested in how these three classifiers behave if we limit the tasks to binary classification. We compare the classifiers on three classic datasets.

1 Introduction

In machine learning, there are algorithms that are designed for binary classification tasks, such as SVM and logistic regression. These classifiers cannot be generalized to multi-class without employing strategies such as One-vs-Rest (OvR). However, there are also classifiers that can be inherently used for multi-class classification.

In this study, we examine how multi-class agnostic classifiers, such as K Nearest Neighbors (KNN), decision trees, and random forests behave when they are used for binary classification tasks.

2 Methodology

2.1 Learning Algorithms

We explore three classifiers, on three datasets each split into three different ratios of training and testing set size: 20:80, 50:50, and 80:20. We first tune the hyper-parameters of these classifiers using 3-fold cross validation on the training set, then report the performance of the classifiers on the test set.

K Nearest Neighbors (KNN) We use different values of K from $\{1,2,3,5,10,20\}$. We use KNN with Euclidean distance.

Decision Tree (DT) We tune the maximum depth of the decision tree from $\{2,4,6,8,10\}$.

Random Forest (RF) We use the default number of 100 trees and also tune the maximum depth of the decision tree from $\{2,4,6,8,10\}$.

2.2 Dataset

We choose three datasets from the UCI repository [4] of a variety of dataset sizes. We convert multi-class classification problems to binary classification by grouping classes. For categorical fields, we use one-hot encoding to make them suitable for training.

Preprint. Under review.

Table 1: Test/Validation/Training Accuracy with 20:80 split

Algo\Dataset	Wine	Bean	Adult
KNN	0.635/0.646/1.000	0.821/0.811/0.891	0.794/0.790/0.800
DT	0.723/0.743/0.758	0.964/0.961/0.999	0.848/0.851/0.869
RF	**0.759**/0.770/0.946	**0.974**/0.968/0.986	**0.857**/0.856/0.878

Table 2: Test/Validation/Training Accuracy with 50:50 split

Algo\Dataset	Wine	Bean	Adult
KNN	0.697/0.657/1.000	0.871/0.855/0.923	0.800/0.797/0.802
DT	0.728/0.739/0.761	0.971/0.968/0.985	0.857/0.854/0.865
RF	**0.791**/0.787/0.952	**0.976**/0.974/0.995	**0.857**/0.856/0.867

Wine Quality [1] This is a dataset with 4898 instances and 11 features. The "target" field contains the overall quality of the wine, which is an integer between 0 to 10. We group this field into bad quality (score less than or equal to 5) and good (score greater than 5) to form a binary classification problem.

Dry Bean Dataset [2] This is a dataset with 13611 instances and 16 features. The beans can belong to 7 classes: {Seker, Barbunya, Bombay, Cali, Dermosan, Horoz and Sira}. We arbitrarily assign {Seker, Barbunya, Bombay, Cali} to one groups and others to another group to form a binary classification problem.

Adult [3] This is a dataset with 48842 instances and 14 features. It uses features such as age and education to predict whether one's income exceeds 50K. It is already a binary classification problem. The classes are {>50K, <=50K}.

2.3 Evaluation Metrics

We use prediction accuracy on the test set to evaluate the performance of classifiers. For each combination of dataset, split, and classifier, we do 3 experiments in a row and record the average of the prediction accuracy.

3 Experimental Results

Table 1, 2, 3 show the accuracy of each classifier on the three datasets during testing/validation/training. The highest testing accuracy in each dataset in each partition is bolded. KNN is never the best model. In almost all cases, Random Forest achieves the best performance. The only exception is that Decision Tree beats Random Forest on the Adult dataset with 80/20 split, which is hard to explain, but indeed possible. The ratio of training set to testing set increases as we go from Table 1 to 2 to 3. We observe that all corresponding accuracy increases or stays almost the same (there are cases when it drops slightly in accuracy after three decimal points).

Abnormality Explained The KNN in Wine dataset has 1.0 training accuracy only when the hyperparameter K chosen by cross validation is 1. This makes sense since it is just training accuracy.

Table 3: Test/Validation/Training Accuracy with 80:20 split

Algo\Dataset	Wine	Bean	Adult
KNN	0.760/0.688/1.000	0.883/0.873/0.935	0.800/0.800/0.804
DT	0.769/0.749/0.892	0.972/0.973/0.984	**0.864**/0.856/0.872
RF	**0.812**/0.794/0.932	**0.976**/0.977/0,995	0.858/0.857/0.865

4 Conclusion

4.1 Discussion

We draw several lessons from this study. First, for a model on a certain dataset (especially large datasets), once it converges, increasing the training set size will no longer have a significant impact on the accuracy. Second, it is not the case that the deeper the a decision tree is, the better, or the more nearest neighbors we evaluate, the better. Cross validation can give us unexpected results. Therefore, such hyper-parameters need to be tuned. Third, even if random forest, which ensembles multiple decision trees, should never perform worst than random forest in theory, we still see such a situation in practice. This may due to the different implementation and parameter choice of the model.

4.2 Conclusion

In this paper we evaluate three classifiers and three datasets based on an empirical comparison. We choose three classifiers (K Nearest Neighbors, Decision Tree, and Random Forest) that are unaware of the number of classes in the dataset and evaluate their performance on binary classification tasks. Among the three classifiers, Random Forest behaves the best in 8 out of 9 (classifier, dataset) combinations while Decision tree excels in the one that is left. K Nearest Neighbors is often off by a big margin. This suggests that if people want to use such multiclass-agnostic classifiers to do binary classification tasks, they should consider Random Forest over the other two possibilities.

4.3 Future Work

We acknowledge there are several drawbacks of this study. First, all datasets we pick have a low feature dimension (around 10). The results may be different on datasets with hundreds or thousands of feature dimensions. Second, we do not experiment on datasets with too few instances (less than 1,000) or too many instances (more than 100,000). Such datasets can be explored in further studies.

References

[1] Cortez,Paulo, Cerdeira,A., Almeida,F., Matos,T., and Reis,J.. (2009). Wine Quality. UCI Machine Learning Repository. https://doi.org/10.24432/C56S3T.

[2] Dry Bean Dataset. (2020). UCI Machine Learning Repository. https://doi.org/10.24432/C50S4B.

[3] Becker,Barry and Kohavi,Ronny. (1996). Adult. UCI Machine Learning Repository. https://doi.org/10.24432/C5XW20.

[4] Markelle Kelly, Rachel Longjohn, Kolby Nottingham, The UCI Machine Learning Repository, https://archive.ics.uci.edu