A Comparative Analysis of Ensemble and Non-Ensemble Machine Learning Algorithms

Yifei Wang
Department of Mathematics, University of Washington, Seattle, Washington, 98195, U.S.A.

Keywords: Machine Learning, Ensemble Learning, Heart Attack Prediction, Mushroom Classification.

Abstract: The development of machine learning has led to the design of various algorithms to effectively address complex problems. Among these, both ensemble and non-ensemble methods have attracted significant attention due to their unique advantages and applications. This paper compares the performance of ensemble and non-ensemble machine learning algorithms in terms of accuracy, efficiency, and stability, using two classification datasets. This work evaluates six algorithms: three non-ensemble methods, which include support vector classification, decision tree, and k-nearest neighbors; and three ensemble methods, which include random forest, gradient boosting, and voting. The performance is validated on two tasks: heart attack prediction and mushroom classification. The results indicate that ensemble algorithms, particularly random forest, and gradient boosting, generally achieve higher accuracy and greater stability compared to the non-ensemble decision tree algorithm. However, despite the slight accuracy improvement, ensemble methods tend to be much slower during both the training and prediction phases. Support vector classification is efficient on smaller datasets but exhibits slower performance on larger ones. Additionally, the performance of voting algorithms is highly dependent on the selection of base models. These findings highlight the trade-offs between accuracy, efficiency, and stability when choosing appropriate machine learning algorithms for specific tasks.

1 INTRODUCTION

Machine learning (ML) is a class of algorithms that analyze existing data, discover patterns, and make predictions. Machine learning can be used to automate decision-making processes. From healthcare to finance, from autonomous driving to natural language processing, machine learning algorithms have been widely adapted to daily lives.

In the field of machine learning, algorithms can be roughly divided into ensemble and non-ensemble algorithms. Non-ensemble algorithms rely on a single model for prediction. Support vector classification, decision trees, and k-nearest neighbors are several common non-ensemble algorithms. Ensemble machine learning algorithms combine the predictions of several base estimators to obtain a more stable and accurate prediction model. Common ensemble algorithms include random forests, gradient boosting, and voting.

This paper aims to analyze and compare the performance of non-ensemble and ensemble machine learning algorithms on two datasets, including the prediction of heart attack and the classification of mushrooms.

2 MACHINE LEARNING ALGORITHMS

To compare ensemble and non-ensemble machine learning algorithms, the author will use the following 6 algorithms. Support Vector Classification, decision trees, and K-Nearest Neighbors are non-ensemble algorithms, while Random Forest, Gradient Boosting, and, Voting are ensemble algorithms.

2.1 Non-Ensemble Algorithms

Machine learning is aimed at finding a function $y = f(x)$ which can be used to model the data from training data (x, y). Non-ensemble machine learning algorithms are hypothesis spaces containing function f. Training is designed to find out the best function from the hypothesis spaces that match the real-world problem best (Muhamedyev, 2015). Decision tree is

Wang, Y.
A Comparative Analysis of Ensemble and Non-Ensemble Machine Learning Algorithms.
DOI: 10.5220/0013332500004558

In *Proceedings of the 1st International Conference on Modern Logistics and Supply Chain Management (MLSCM 2024)*, pages 382-386
ISBN: 978-989-758-738-2

one of the classic non-ensemble algorithms. And following are non-ensemble algorithms that will be used for compare.

Decision Trees (DT): A decision tree is a supervised learning algorithm for classification and regression tasks (Loh, 2011). It builds a structure similar to a binary tree, dividing the data step by step according to input characteristics. Throughout the process, different options are examined, and the most effective ones are selected at each level, leading to the final decision.

Support Vector Classification (SVC): Support Vector Machines (SVMs) are supervised learning algorithms used for classification and regression tasks (Cortes, 1995). They work by finding an optimal hyperplane that maximizes the margin between different classes in an N-dimensional space. A kernel function is used to transform the data to achieve a better result compared with simpler algorithms like linear regression (Salcedo-Sanz, 2014).

K-Nearest Neighbors (kNN): The k-nearest neighbors algorithm is a supervised learning method frequently applied when solving classification and regression problems. By finding the k-nearest neighbors to a given point, the algorithm gives the output based on choosing the most common category for classification or the mean value for regression tasks. A parameter k is used to control the number of nearest points that participate in the prediction.

2.2 Ensemble Algorithms

Ensemble algorithms are learning algorithms that construct a set of models and make decisions based on the combination of them. Bagging and boosting are common strategies for building ensemble models (Dietterich, 2000). Bagging is a technology first proposed by Breiman (Breiman, 1996). With bagging, a new training set is created by randomly sampling from the original training set with a replacement for each base classifier. Boosting is a strategy proposed by Schapire (Schapire, 1990). It focuses on improving the performance of weak classifiers by sequentially training them, each time focusing more on the instances that previous classifiers misclassified. And following are ensemble algorithms that will be used.

Random Forest (RF): RF is an ensemble learning method that builds multiple decision trees and merges them to get a more accurate and stable prediction (Ho, 1995). In random forests, each tree is built from a bootstrap sample of the training data, and at each split, a random subset of features is considered to find the best split. It may avoid some overfitting problems from the decision tree as it builds multiple decision trees.

Gradient Boosting (GB): Gradient Boosting is an ensemble learning technique that builds a model in a stage-wise fashion from decision trees, which are supervised learning methods used for classification and regression (Friedman, 2002). Gradient Boosting usually combines multiple decision trees to predict the result, which makes it a strong model. It achieves this by finding a loss function and choosing the one with the least error through gradient descent. It iterates many times on the weak learners to build a more precise model.

Voting: Voting is also an ensemble learning technique. The mechanism of it is collecting the results of different models and considering all of them to give an average prediction. In the context of voting, there are two main types: hard voting and soft voting. Hard voting involves taking the majority vote from the predictions of all models, while soft voting averages the predicted probabilities and selects the class with the highest average probability.

2.3 Algorithms Comparison

As Dietterich pointed out, ensemble algorithms may achieve better performance due to three reasons: statistical, computational, and representational (Dietterich, 2000).

Statistical: The hypothesis space that requires searching is too large, however, people typically do not have enough training data to determine the model precisely. Try to learn a model based on them typically caused overfitting. Combining multiple models may offset the errors in each model and avoid overfitting.

Computational: Find the best function in hypothesis space such as a decision tree that could be an NP-Hard problem. So, some heuristics search must be applied to find the function. And therefore, the function may not be the best one. Combining multiple models makes the prediction closer to the optimal solution.

Representational: The hypothesis space may not actually contain the best function. So, the best model based on a specified algorithm may not be the best to represent the real-world problem.

3 EXPERIMENTS AND RESULTS

3.1 Dataset

This work will evaluate these algorithms based on two datasets: (1) The Heart Attack Dataset by Rashik. The dataset contains various fields such as age, sex, and other cardiovascular health indicators for some people. And it aimed to predict if the person has a higher chance of heart attack. The dataset is tiny and contains 303 rows and 14 columns (Rashik, 2021). (2) The Mushroom Dataset for Binary Classification Available at UCI Library. The dataset contains different properties of mushrooms such as color, shape, and size. It aimed to predict if the given mushroom is poisonous or edible. The dataset is much larger, which contains 54035 rows and 9 columns (Joakim, 2023). All these datasets are publicly available on Kaggle.

3.2 Evaluation Metrics

This work applies all 6 algorithms discussed above on these two datasets. When preprocessing the dataset, all categorical columns are one-hot encoded. This work splits the dataset randomly, so 80% rows are used for training, and the remaining 20% rows are used for evaluating.

For each algorithm, this work measures it with these key properties:

Accuracy (Acc.): Indicates the percentage of instances correctly predicted compared to the total in the test set. A higher value means a better outcome for the algorithm. It is one of the most important indicators for evaluating a model.

Training accuracy: Reflects how many instances were accurately predicted relative to the total instances in the training set. A higher value means the model learned more in the training set and may also indicate the overfitting.

Time usage (train time + predict time): Measures the time spent on training the model and generating predictions on the test dataset. A shorter time means more efficient model predictions. The evaluation is executed on the computer with Intel Core i7-8650U.

3.3 Performance Comparison

The evaluated result on the mushroom dataset is shown in Table 1. This work sets specific parameters for each machine learning algorithm. DT is constrained with a maximum depth of 25 and considers up to 28 features. kNN is based on the 5 nearest neighbors, with predictions weighted by the distance to these neighbors. SVC employs a polynomial kernel to transform the data, which can enhance its classification capabilities. RF is configured with a depth limit of 32 and is trained using 100 decision trees. GB operates with a similar depth constraint of 25 and a feature limit of 28, and it constructs the model through 100 iterations. Finally, Vote method combines the predictions from the three non-ensemble algorithms to determine the final classification.

Table 1: Performance comparison on mushroom dataset.

Algorithm	Val Acc	Train Acc	Time
DT	0.9802	0.9982	0.32/0.33
kNN	0.6807	1.0000	0.04/1.85
SVC	0.5564	0.5525	391.7/421.7
RF	0.9899	1.0000	6.55/6.77
GB	0.9892	1.0000	86.02/86.23
Vote	0.8307	0.9997	395.9/429. 0

The evaluate result on the heart attack dataset is shown in Table 2. In this study, DT is capped at a depth of 6 for controlled growth. The kNN uses 7 neighbors in its unweighted predictions. SVC applies a polynomial kernel to enhance data classification. RF, with a depth limit of 5, is trained on 400 decision trees to improve accuracy through diversity. GB is limited to a depth of 3, which undergoes 150 iterations to refine its model. Vote method pools predictions from the DT, kNN, and SVC through hard voting, aiming to consolidate strengths for better accuracy.

Table 2: Performance comparison on heart attack dataset.

Algorithm	Val Acc	Train Acc	Time
DT	0.9802	0.9982	0.32/0.33
kNN	0.6807	1.0000	0.04/1.85
SVC	0.5564	0.5525	391.7/421.7
RF	0.9899	1.0000	6.55/6.77
GB	0.9892	1.0000	86.02/86.23
Vote	0.8307	0.9997	395.9/429.0

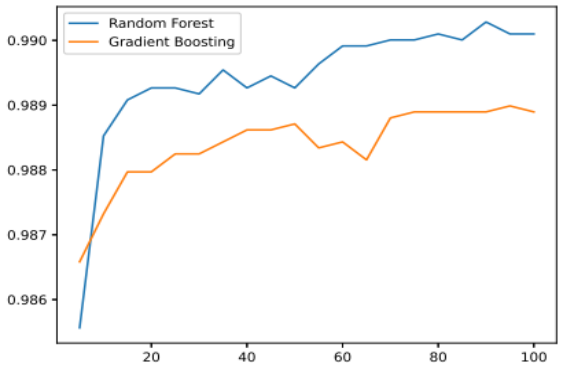

Figure 1: Accuracy of models with given number of estimators (Figure Credits: Original).

This work also interested in the 1.0 training accuracy on the first dataset archived by algorithm random forest and gradient boosting. So, this work trains these two algorithms with different numbers of estimators. The accuracy after the different number of estimators is shown in Figure 1.

4 DISCUSSIONS

4.1 Performance

Accuracy is one of the most important indicators when evaluating models. The author first analyzes the accuracy of given datasets for all these algorithms, and finds out that within the non-ensemble models, random forests have the best accuracy. This may be due to the random forest being more suitable to a given dataset. As it has a much better accuracy, the voting algorithm based on all these 3 algorithms perform worse than the decision tree. However, it could also be noticed that random forest and gradient boosting have similar or better accuracy compare to the decision tree. As these two algorithms are based on the decision tree and have made improvements to it. As distance weighted kNN is applied on mushroom dataset, it suggests that it overfitted the train data with a 1.0 train accuracy. However, since non-weighted kNN is used on the heart attack dataset, the kNN algorithm does not show overfitting. To avoid overfitting, this work limited the depth and features when training decision tree models. By given these parameters, the decision tree has a lower train accuracy but a better performance. It could also be noticed that random forest and gradient boosting have a very high train accuracy. While it may not mean the model is overfitted. As the model is not getting lower accuracy when increasing the number of estimators, aka. underlying decision trees or the number of iterations as shown in Figure 1.

4.2 Time Consumption

Among the three non-ensemble algorithms, SVC has the worst efficiency. It takes much longer time when applied on a large dataset. The kNN algorithm has a lower training time, but a longer predicting time. This is due to the fact that the algorithm is not actually trained into certain model, but use all train data when predicting. The vote algorithm needs to first train all these 3 algorithms. So, its time usage is about to be the sum of the above 3 algorithms. And since this work included the SVC algorithm which is slow on large dataset, it suggests voting have a very poor efficient. Random forest and gradient boosting are all based on decision trees. And there are parameters which could be used to control number of decision trees or number of iterations. So, their efficiency is heavily influenced by the parameter. As a result, they are much slower than decision trees. However, they still show a better efficient when compare to SVC on large datasets.

4.3 Randomness and Stability

The kNN and SVC algorithms are not relied on randomness, so they always have the stable outcome. However, the decision tree needs randomness when splitting nodes. And as the result, the decision tree is not stable. Based on different random seeds, it may have different accuracy.

Random forest is based on decision tree. However, it combines the results from multiple decision trees. So, it has a higher stability. Similar behavior may be observed in gradient boosting. It iterates many times to avoid the unstable introduced by randomness.

5 CONCLUSIONS

This work presents a comprehensive comparison of ensemble and non-ensemble machine learning algorithms, focusing on their performance, efficiency, and stability. The analysis includes decision trees, support vector classification, K-nearest neighbors, random forests, gradient boosting, and voting algorithms.

From the evaluation, it could be observed that ensemble methods, especially random forests and gradient boosting, generally outperform non-ensemble methods in terms of accuracy. This can be attributed to their ability to combine multiple models, thereby reducing overfitting and enhancing generalization. However, the voting algorithm did not perform as well as expected, possibly due to the inclusion of SVC, which exhibits inefficiency on large datasets.

In terms of training and prediction time, non-ensemble methods such as kNN and decision trees exhibit faster training times, but their prediction efficiency varies. kNN, in particular, exhibits longer prediction times due to its reliance on the entire training dataset. Ensemble methods are slower during training due to the complexity of combining multiple models, but are still more efficient than SVC on large datasets.

Stability analysis shows that non-ensemble methods such as kNN and SVC provide consistent

results, while decision trees exhibit variability due to their reliance on randomness. Ensemble methods such as random forests and gradient boosting mitigate this instability by aggregating the results of multiple models, thereby providing greater stability.

Results show that while non-ensemble methods can be efficient and easy to implement, ensemble methods provide better accuracy and stability, making them more suitable for complex datasets. When choosing a machine learning algorithm, it is required to consider factors such as performance and accuracy and choose the right algorithm.

REFERENCES

Breiman, L. 1996. Bagging predictors. *Machine learning*, *24*, 123-140.

Cortes, C. 1995. Support-Vector Networks. *Machine Learning*, 1-25.

Dietterich, T. G. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*. 1-15.

Friedman, J. H. 2002. Stochastic gradient boosting. *Computational statistics & data analysis*, *38*(4), 367-378.

Ho, T. K. 1995. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*. 1, 278-282.

Joakim, A. 2023. Secondary Mushroom Dataset. URL: https://www.kaggle.com/datasets/joebeachcapital/secondary-mushroom-dataset. Last Accessed: 2024/09/13.

Loh, W. Y. 2011. Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, *1*(1), 14-23.

Muhamedyev, R. 2015. Machine learning methods: An overview. *Computer modelling & new technologies*, *19*(6), 14-29.

Rashik, R. 2021. Heart Attack Analysis & Prediction Data set. URL: https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset. Last Accessed: 2024/09/13

Salcedo-Sanz, S., Rojo-Álvarez, J. L., Martínez-Ramón, M., & Camps-Valls, G. 2014. Support vector machines in engineering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *4*(3), 234-267.

Schapire, R. E. 1990. The strength of weak learnability. *Machine learning*, *5*, 197-227.