Analysis of Tree-based Machine Learning Models

Mincong Zhou
UNSW
Sydney, Australia
z5358282@ad.unsw.edu.au

***Abstract*—Use of pruning, bagging, and boosting to improve the performance of tree-based models and analyze how well these methods are and what information can they tell us. In this work, we use these methods to predict the age of abalone. It shows that the shell weight is the most important feature in predicting the age of abalone. And compared to the decision tree without restriction, pruning, bagging and boosting improve the model's performance significantly with about 10% accuracy increased. But it also shows that we can barely improve the performance further by only applying these methods and the best performance has only about 62% accuracy, which is still a poor performance. According to the analysis, the paper suggests using decision tree with pruning if we want to have good interpretation of the model and use Boosting and Random Forest if we want to have better performance and not care too much about the interpretation.**

Keywords—Decision Tree, Pruning, Bagging, Boosting, Prediction

I. INTRODUCTION

Data mining is a process with computer-aided of analyzing data from different perspectives. Data mining tools can forecast the behavior of data and assist to take knowledge-driven decisions. One of the applications is to solve the classification problem and the tree-based model is one of the solutions with data mining.

Tree algorithm has developed 58 years [1]. It is useful to look back and study how the algorithm develops and analyze them by comparing their performance for each other. But due to the large amount of literature, it is impossible and unnecessary to explain and analyze all of them. Therefore, we focus on the major tree-based models which are widely used in different fields. Decision trees are usually used in decision analysis to assist decision makers to identify which strategy has the highest likelihood to reach their goal. But decision trees are also popular in machine learning. In machine learning, they can complete classification, regression, and multioutput tasks. They can usually do an excellent job in fitting complex datasets. And decision trees are the foundation of Random Forests, which is one of the most powerful methods in machine learning nowadays [2].

Tree-based models are white-box models which means we can visualize how a model presents. But the price is that they usually provide worst accuracy than black-box models, like neural network. And decision tree learners often face overfitting problems by constructing over-complex trees. Because of this, they don't have a good generalization from the training data. If dataset contains categorical variables, information gains in the trees will be biased and tends to have more levels of the attributes [3]. To reduce the effect of overfitting, it is advised to prune the decision trees. And the most popular methods are restricting the maximum depth, maximum number of leaf nodes, features, or cases per node/leaf [4]. To improve the performance, it is suggested to implement ensemble learning [5]. A decision tree is a single decision-maker, and it is a wise choice to give a prediction after discussing with many decision-makers. Ensemble learning is the method that tries to apply collective wisdom. Bagging is the process of applying ensemble learning. It selects subsets of training dataset randomly then builds multiple models with the same algorithm [6]. We can apply bagging via Random Forests [7]. Boosting is another ensemble learning method to reduce the effect of overfitting by optimization and regularization. Two of the most commonly used boosting methods are Gradient Boosting and XGBoost.

This paper aims to analyze the advantages and disadvantages of different tree-based models, so that we can use proper models with different requirements.

The rest of the paper is organized as follows. In section 2, it introduces how the dataset preprocessed and gives more details in the models we use. In section 3, it summarizes the results from our methodology. In section 4, we discuss the results. Finally, we give the conclusion of the paper in section 5.

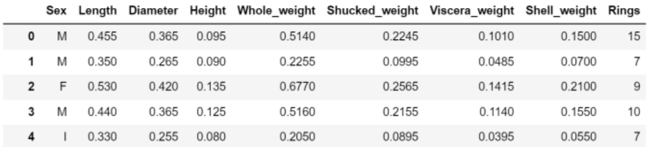
II. METHODOLOGY

A. *Dataset*

The dataset used here records various indicators of abalone [8]. The dataset has 9 variables and 4177 data for each variable. We aim to predict the age of abalone.

The dataset includes 8 features. Rings is the variable to predict [9]. Table 1 shows the first five data of the dataset.

Table 1



	Sex	Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weight	Shell_weight	Rings
0	M	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.1500	15
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.0700	7
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.2100	9
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.1550	10
4	I	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.0550	7

B. Data Processing

Use 'pandas' library in python to load the data and transform 'Sex' variables into integer values by converting M, F, I to 0, 1, 2 respectively.

Note that the response variable 'Rings' is not classified data. To become a classification problem, 'Rings' variable classifies into 4 groups: 0-7 is class 1, 8-10 is class 2, 11-15 is class 3 and the value greater than

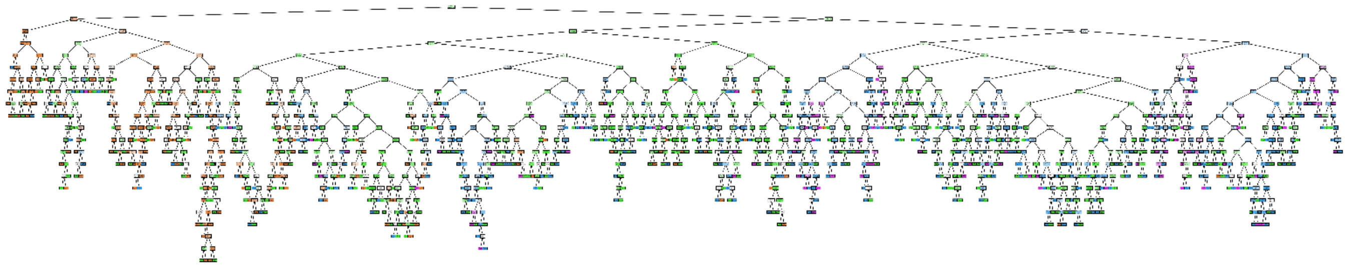
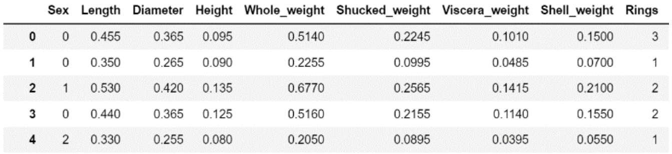


Figure 1 A CART decision tree without pruning, bagging, and boosting

15 is class 4. Table 2 shows the first five data of the dataset after data processing.

Table 2



	Sex	Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weight	Shell_weight	Rings
0	0	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.1500	3
1	0	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.0700	1
2	1	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.2100	2
3	0	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.1550	2
4	2	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.0550	1

Correlation heatmap shows the relationship between each pair of variables.

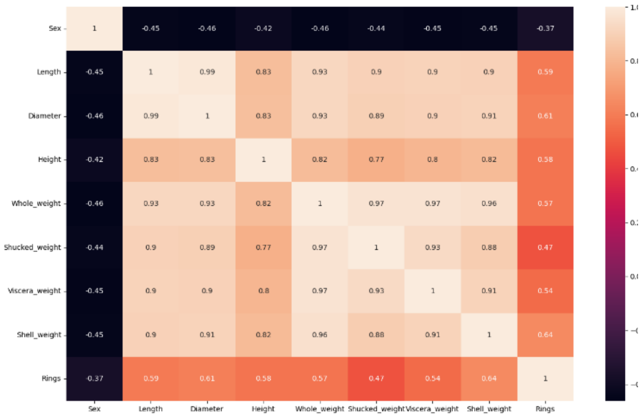


Figure 2. Correlation heatmap

Figure 2 is the correlation heatmap of the dataset. From the heatmap, we observe that most variables have high positive correlation between each other except 'Sex' variable.

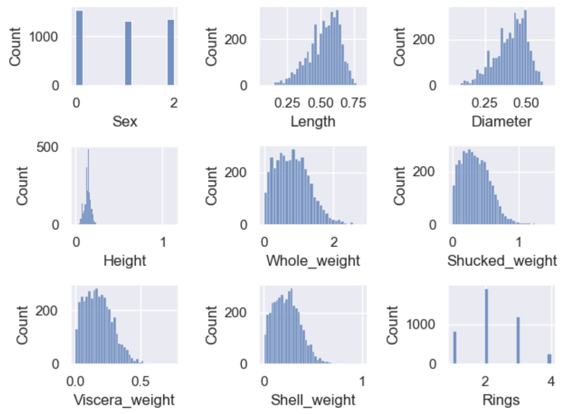


Figure 3. Histograms for all variables

Figure 3 show that 'Sex' variable is uniformly distributed. 'Heigh' has normal distribution. 'Length' and 'Diameter' have distribution with negative skew. And other variables have distribution with positive skew.

The dataset is 60/40 train/test split for all models.

C. *Modelling*

Decision trees are commonly used for solving classification problems. A decision tree takes a set of features to predict and decide a variable needed to be classified [10]. The Classification and Regression Trees (CART) algorithm uses an impurity measure called Gini instead of Entropy. Although using Gini measure can run faster, the problem is that it may not produce more balanced decision compared to the entropy measure. The cost function of CART is to minimize Gini Impurity. The Gini measure is as follow:

$$Gini = 1 - \Sigma_{i=1}^{n}(p_i)^2$$

where p_i is the probability of an object falling to a particular category [11].

In this work, we apply the decision tree with CART algorithm without any restrictions first then do further to improve the model performance.

Pruning is a method to remove insignificant nodes and it can improve prediction accuracy. There are many approaches for pruning a tree. One of the methods we used here is post-pruning with cost complexity pruning. The cost complexity pruning can prevent overfitting by controlling the size of a tree. It is done by choosing the proper cost complexity parameter, alpha [4].

Random Forest is an ensemble method. It uses bagging ensemble learning approach to aggregate the prediction of each predictor. A predictor is a decision tree. In this work, we use Random Forests for bagging the tree and investigate the performance of increasing number of trees in the ensembles. For each Random Forest model with different number of trees in the ensembles, we run the model 10 times and calculate their accuracy score with 95% confidence intervals to obtain the mean. The investigation can give the proper number of trees in the ensembles and rank the features based on their importance.

Boosting is another ensemble method to let weak learners become strong learners. It trains weak learners one by one and puts them into the iterative framework. By doing so, weak learners can correct its predecessor and the model accuracy improves. Gradient Boosting uses residual error from the predictors to boost. XGBoost does further and there is regularization as the trees construction involves pruning. Both methods sum the prediction of trees. In this work, we boost the decision tree by XGBoost and Gradient Boosting. The algorithm can improve accuracy via an iterative framework. The learning rate of these two boosting is set to 0.1.

D. *Software suite*

The paper generally uses sklearn library to model and investigate the models. And XGBoost uses xgboost library for constructing.

III. Results

In this section, we present the results of our proposed approaches.

Figure 1 presents a CART decision tree without pruning, bagging and boosting. We note that there are many nodes and leaves. The accuracy score of the decision tree is 53% which is close to 50%. It means the performance is not good. If-Then rules of some selected nodes are following:

- IF (shell weight ≤ 0.03) THEN (Age is in class 1)
- IF ($0.03 <$ shell weight ≤ 0.05) THEN (Age is in class 3)

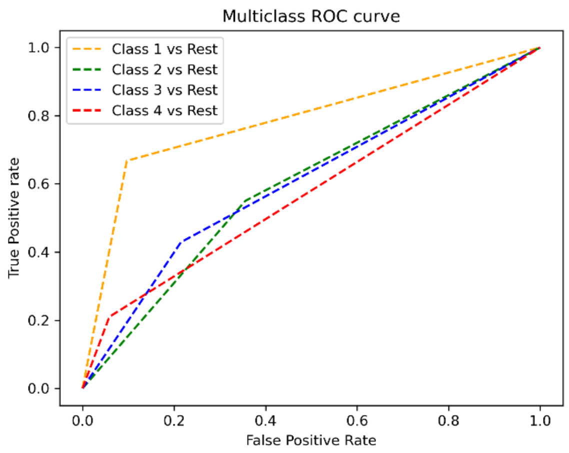


Figure 4 Multiclass ROC curve for DT tree

Figure 4 is the ROC curve classifying each class against others. The AUC of class 1 is 0.79 which means that there is 79% chance that the model can distinguish between class 1 class and other classes. And it is the best performance in these 4 classes. The AUC of class 3 is 0.61 which is the second-best performance. The AUC of class 2 is 0.60 which is the third best performance. And the AUC of class 4 is 0.58 which has the worst performance.

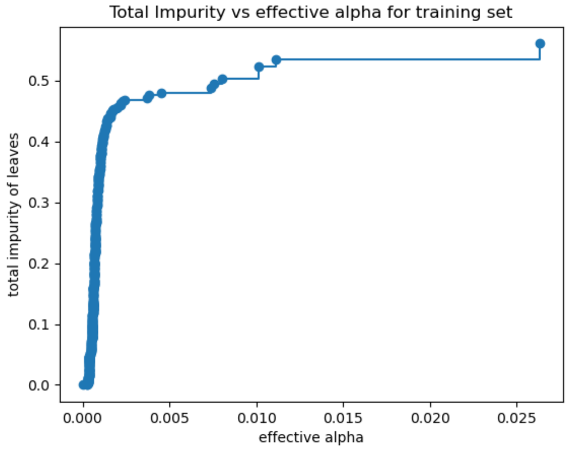


Figure 5 Relationship between alpha and impurity

Figure 5 describes that when the effective alpha becomes larger, the total impurity of its leaves increases. It is because larger alpha means more of tree is pruned. And it eventually flattens.

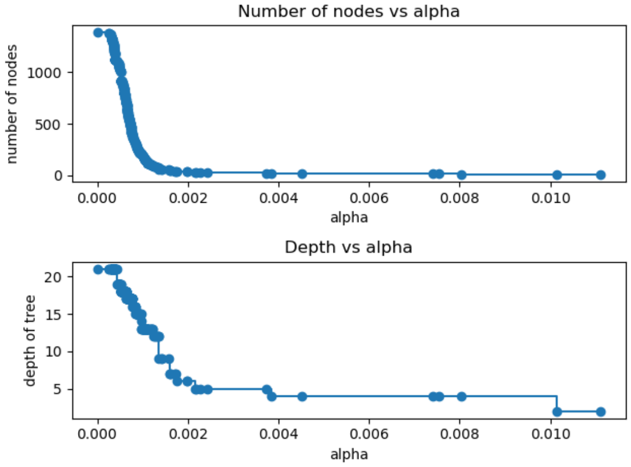


Figure 6 Relationship between alpha and depth and nodes

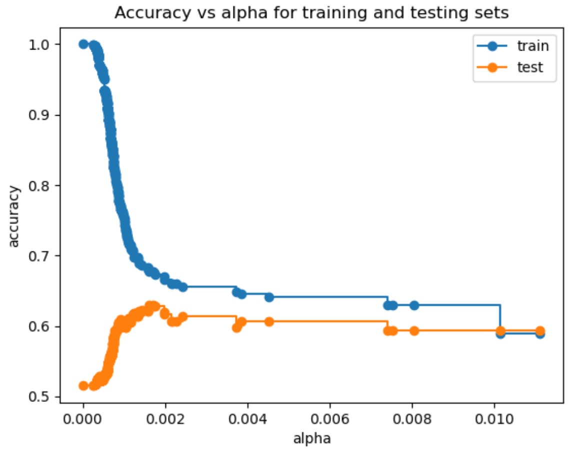


Figure 7 The accuracy of training and testing sets when alpha changes

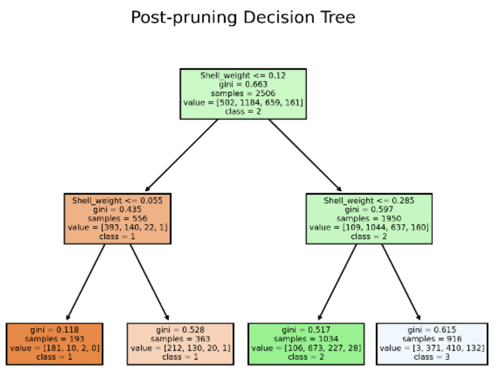


Figure 8 post-pruning decision tree

Figure 6 shows that when alpha rises, the amount of nodes and the depth of the tree reduce. Figure 7 shows that when alpha increases, the accuracy of training set decreases and the accuracy of testing set increases to a certain level then starts to decrease. It indicates that the alpha implies the smallest difference between the accuracy of training and testing sets will be able to give the model the best performance. In this case, the alpha value is 0.01. So, we construct the post-pruning decision tree by setting alpha equal to 0.01. Figure 8 shows the post-pruning decision tree with alpha = 0.01. The accuracy score of the decision tree is 59%, which is 6% higher than the tree without post-pruning. It means the performance is not good. If-Then rules of some selected nodes are following:

- IF (Shell weight ≤ 0.055) THEN (Age is in class 1)
- IF (0.055 < Shell weight ≤ 0.12) THEN (Age is in class 1)
- IF (0.12 < Shell weight ≤ 0.285) THEN (Age is in class 2)
- IF (Shell weight > 0.285) THEN (Age is in class 3)

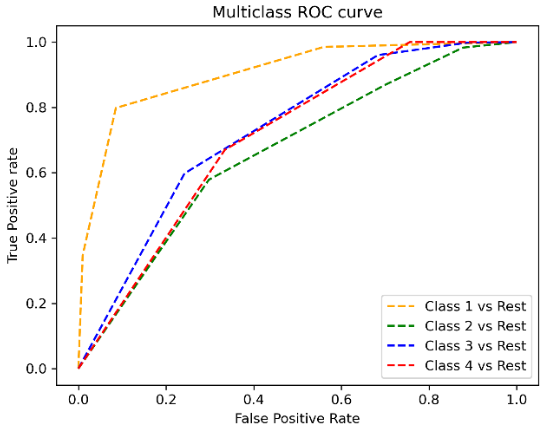


Figure 9 Multiclass ROC curve for DT tree with post-pruning

Figure 9 is the ROC curve of the post-pruning tree classifying each class against others. The AUC of class 1 is 0.90 which means that there is 90% chance that the model can distinguish between class 1 class and other classes. And it is the best performance in these 4 classes. The AUC of class 3 is 0.73 which is the second-best performance. The AUC of class 4 is 0.71 which is the third best performance. And the AUC of class 2 is 0.66 which has the worst performance.

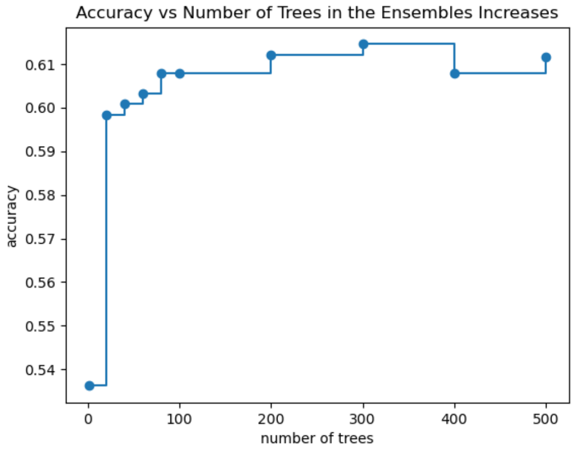


Figure 10 Relationship between accuracy of Random Forests and num of trees in the ensembles

Figure 10 shows when the number of trees in the Random Forests is between 1 and 300, the accuracy of the forest increases. And the accuracy increases sharply when the number of trees is from 1 to 20. Then, when the amount of trees is more than 300, the accuracy seems to flatten and it is around 0.615.

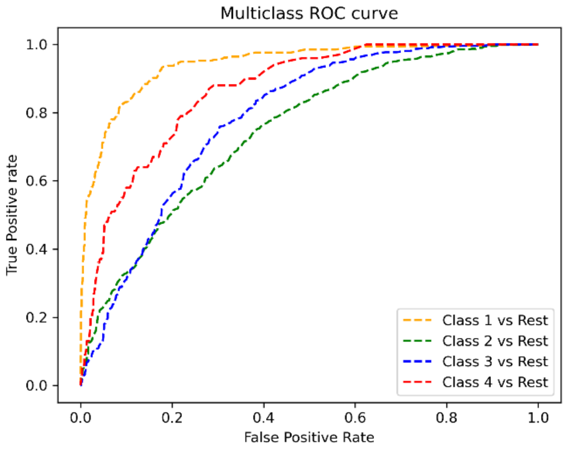


Figure 11 Multiclass ROC of Random Forest with 300 trees in the ensembles

Figure 11 is the ROC curve of the Random Forest classifying each class against others. The AUC of class 1 is 0.94 which means that there is 94% chance that the model can distinguish between class 1 class and other classes. And it is the best performance in these 4 classes. The AUC of class 4 is 0.86 which is the second-best performance. The AUC of class 3 is 0.78 which is the third best performance. And the AUC of class 2 is 0.75 which has the worst performance.

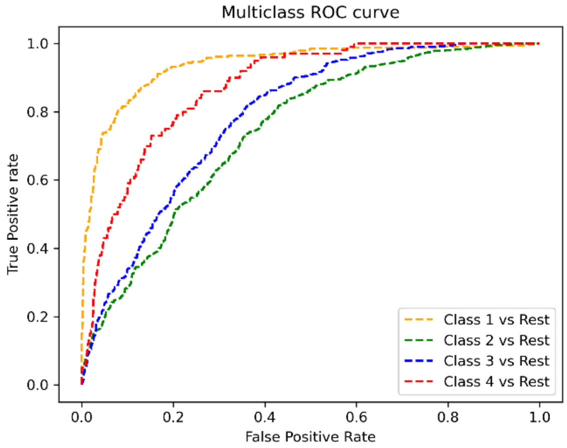


Figure 12 Multiclass ROC of Gradient Boosting

The accuracy performance of Gradient Boosting is 0.62. Figure 12 is the ROC curve of the Random Forest classifying each class against others. The AUC of class 1 is 0.94 which means that there is 94% chance that the model can distinguish between class 1 class and other classes. And it is the best performance in these 4 classes. The AUC of class 4 is 0.87 which is the second-best performance. The AUC of class 3 is 0.79 which is the third best performance. And the AUC of class 2 is 0.74 which has the worst performance.

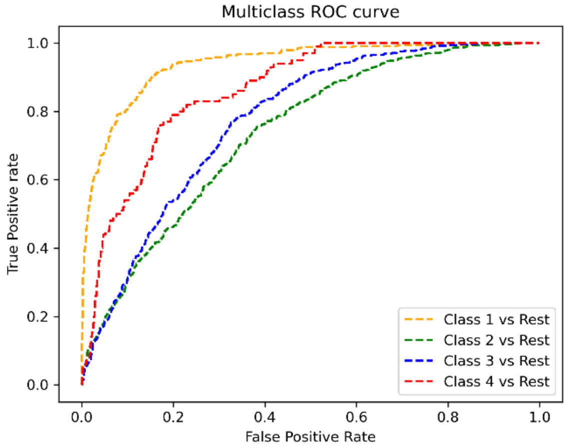


Figure 13 Multiclass ROC of XGBoost

The accuracy performance of Gradient Boosting is 0.612. Figure 13 is the ROC curve of the Random Forest classifying each class against others. The AUC of class 1 is 0.94 which means that there is 94% chance that the model can distinguish between class 1 class and other classes. And it is the best performance in these 4 classes. The AUC of class 4 is 0.86 which is the second-best performance. The AUC of class 3 is 0.77 which is the third best performance. And the AUC of class 2 is 0.74 which has the worst performance.

IV. Discussion

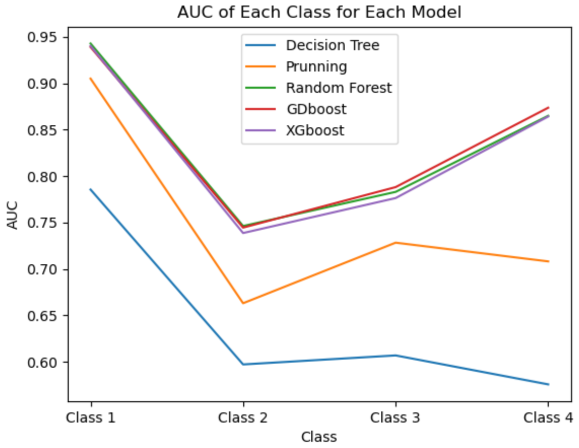


Figure 14 AUC for Each Model

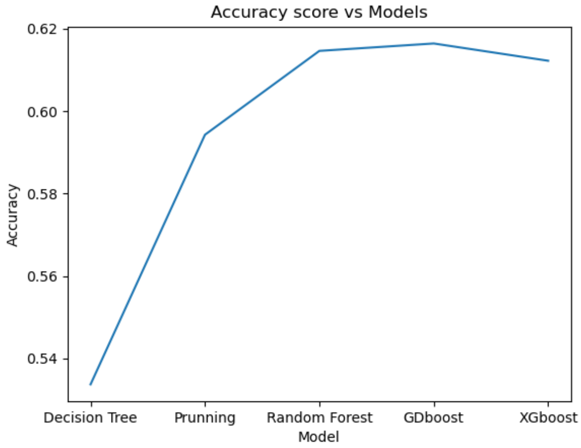


Figure 15 Accuracy score of models

Figure 14 summarize the AUC of each class in all five models. We can see that Decision Tree without modification has the lowest chance to distinguish all 4 classes. In other hands, the AUC curves of Random Forest, Gradient Boosting and XGBoost are almost the same. Figure 15 also proves that Decision Tree without modification has the worst performance and Forests and Boosting models performance similarly. Considering the accuracy score and AUC, it seems that there is no difference for using Forests and boosting models to improve the performance.

Decision Tree without restriction usually has overfitting problems which let the model be unable to generalize well. As a result, it reduces the model performance. The result makes sense because lots of studies have proven this before. And pruning the tree can significantly improve the performance. The other advantage of pruning the tree is that the tree becomes smaller which makes it easier to interpret.

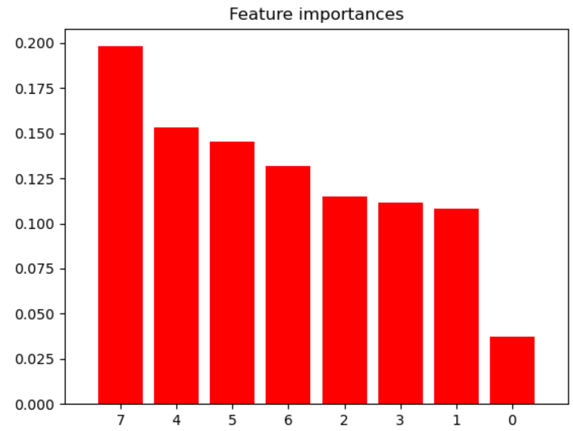


Figure 16 Feature importance ranked by Random Forest

As expected, Random Forest model improves the performance more compared to the pruning tree. And increasing the number of trees in the ensembles usually increases the model performance. However, the performance will not improve anymore when the number of trees reaches a certain level. And the forest with 300 trees is the best forest which also makes it hard to interpret. However, it can rank the features based on their importance and frequency used. Figure 16 suggests that feature 7, which is shell weight, is the most informative feature. And feature 0, which is gender, is the least informative feature.

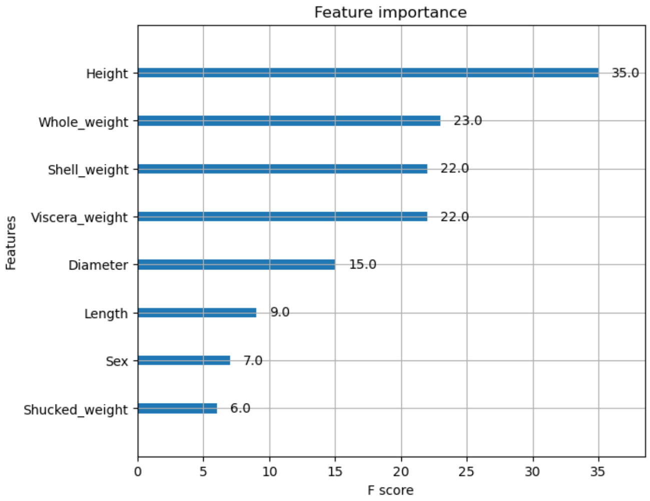


Figure 17 Feature importance ranked by XGBoost

XGBoost is thought to perform better than Gradient boosting because it is more regularized to control over-fitting. But the result is, the performance doesn’t improve, and it is almost the same as using Gradient boosting. However, Gradient boosting is unable to rank the features based on their importance like Random Forest does. But XGBoost can rank them. It gives another way to have Feature Selection as Figure 17 shown. The ranking shows that XGBoost gives height variable the highest informative and Length, gender and shucked weight are the least informative features. Compared to Random Forests, although XGBoost has similar performance, it is more sufficient, and the computer can run the algorithm faster.

V. Conclusion

In general, XGBoost and Random Forest are the ideal model to use in a classification problem if we desire to have better performance instead of easier interpretation, but they can show us two ways to rank features based on their importance that gives us better understanding to the classes and dataset. And decision tree with pruning is the ideal model for interpretation. The performance of decision tree with pruning may improve by setting more restrictions like maximum number of leaves, maximum number of nodes and so on. Decision tree without restriction is overfitting seriously meaning that it is not an ideal way to only use it for prediction or interpretation.

The disadvantages of these ensembles method is that it is not flexible and simple. To get the best performance in our selected methods, we need to execute the methods one by one and manually select the proper method which makes it not sufficient. And for each model, we can’t take the good and discard the bad to improve the performance and at the same time have a easy interpretation. If there is a method can combine the advantage of decision tree with pruning that is easy to interpret and the advantages of XGBoost and Random Forest that are having good performance and able to implement Feature Selection, the world will be a better place.

References

[1] W. Y. Loh, "Fifty years of classification and regression trees," *International Statistical Review,* vol. 82, no. 3, pp. 329-348, 2014.

[2] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2nd Ed.), O'Reilly Media, Inc, 2019, p. 177.

[3] Deng, H., Runger, G., & Tuv, E, Bias of importance measures for multi-valued attributes and solutions. In International conference on artificial neural networks, Springer, Berlin, Heidelberg, 2011, pp. 293-300.

[4] J. R. Quinlan, "Simplifying decision trees," *International journal of man-machine studies,* vol. 27, no. 3, pp. 221-234, 1987.

[5] Sagi, O, & Rokach, L, "Ensemble learning: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery,* vol. 8, no. 4, p. e1249, 2018.

[6] L. Breiman, "Bagging predictors," *Machine learning,* vol. 24, no. 2, pp. 123-140, 1996.

[7] L. Breiman, "Random forests," *Machine learning,* vol. 45, no. 1, pp. 5-32, 2001.

[8] Warwick J Nash, Tracy L Sellers, Simon R Talbot, Andrew J Cawthorn and Wes B Ford, "The Population Biology of Abalone (_Haliotis_ species) in Tasmania. I. Blacklip Abalone (_H. rubra_) from the North Coast and Islands of Bass Strait," Sea Fisheries Division, 1994.

[9] "Abalone Data Set," UCI Machine Learning Repository, [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Abalone.

[10] Quinlan, J. R, "Induction of decision trees," *Machine learning,* vol. 1, pp. 81-106, 1986.

[11] Steinberg, D, & Colla, P, "CART: classification and regression trees," in *The top ten algorithms in data mining,* p. 179.