**Tree and Ensemble Learning, an exploration of Abalone Dataset**

**Abstract**

This paper presents a comprehensive evaluation of machine learning classification techniques on the UC Irvine: Abalone data set [9]. The study aims to evaluate the classification performance of Decision Trees and Random Forests. Furthermore, discussing the benefits and implications of utilizing model modifications (hyperparameter tuning) such as CCP Alpha and N-Estimators. The results illustrate the respective strengths of the models. This paper aims to add to the overall discussion of machine learning models and the understanding of methodological considerations.

**Introduction**

The abalone dataset represents physical measures of Abalone from the north coast and islands of bass strait (Tasmania). The features include Sex, Length, Diameter, Height, Whole Weight, Shucked Weight, Viscera Weight, Shell Weight, and Rings. The feature of interest is Rings. The Rings provide a means of approximating the age of the Abalone +-1.5 in years. However, the process involved in counting the rings is extremely tedious. Through this analysis, there is a hope to utilize the other eight, readily measured, features using tree and ensemble learning to aid in the classification of age process.

Methodology

The core methodology is detailed below. There are some amendments that were made during the analysis because the results prompted further ideas to explore. To keep the methodology section, clear of result discussion, those amendments are detailed in the result section of the paper.

Before modelling can commence there is a requirement to amend two features of the data set. 1) The Sex feature (I: Infant, F: Female, M: Male) is converted to integer values (0,1,2), this primarily assists in the graphing of the features in the exploration phase of the analysis. They are converted in that order because, in general, that is the order of respective sizes. 2) The Rings feature converted into 4 classes:  
Class1: 0-7 Years  
Class2: 8-10 Years  
Class3: 11-15 Years  
Class4: Greater than 15 Years  
This is done to balance the sample set in class group and simplify the resulting model outputs. One of the key benefits of Decision Trees is their readability and interpretability, the model outputs explode when considering 29 (The max ring count in the sample set is 29) different age classes.

With our new classes is important to understand the distribution of class and distribution of features. This primarily to provide context to the interpretation of model results. A correlation matrix was produced to understand what features distributions should be reviewed. The strongest and weakest correlated features with class was reviewed, as well as a distribution of class itself.

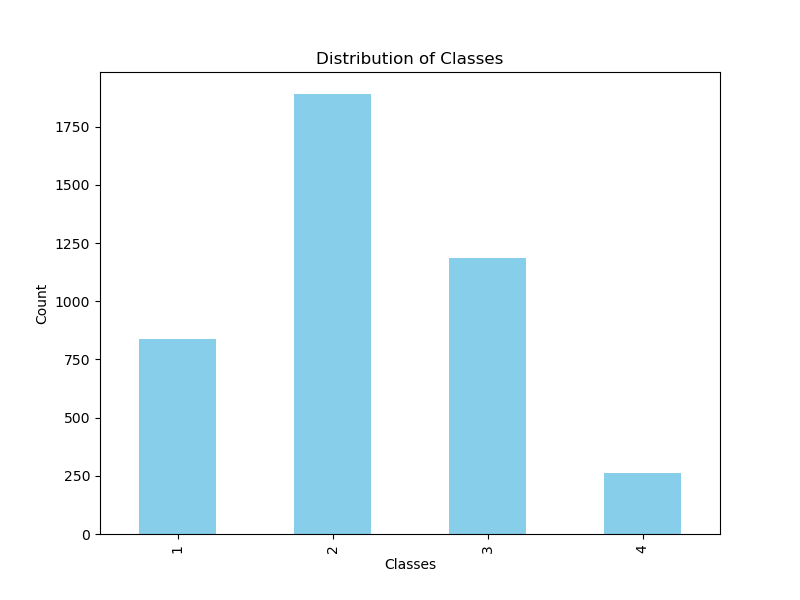
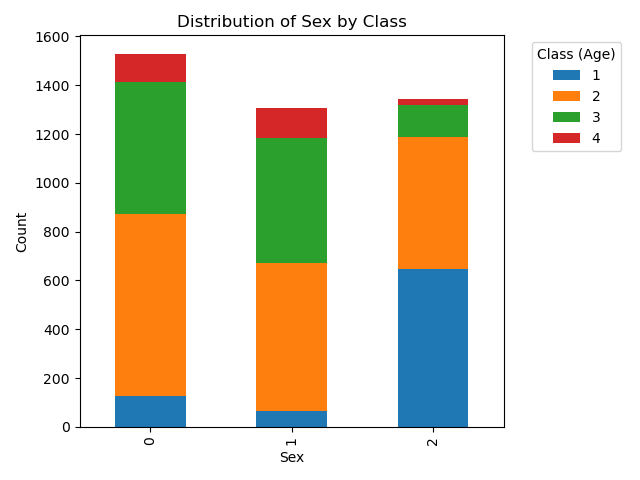
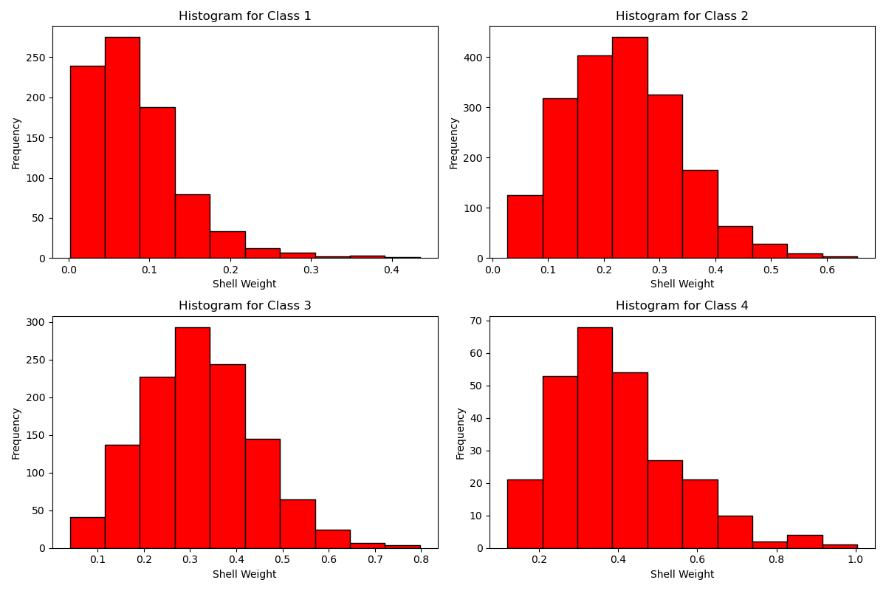
All modelling was produced with a 60-40 train-test split. The core discussion is done with a random\_state = 99 whereas the mean and interval performance measures of each model was done with random\_state = 1 through 10 as means of results replication.

Three models were initially created.

1. A Full Decision Tree not limited in depth, split or CCP Alpha values
2. After analysing the variations of performance from changes in CCP Alpha, a pre-pruned Decision Tree
3. Utilising the bagging of trees via Random Forest

A fourth model was produced from the initial results, and this was tested and discussed in the results

**Results**

**Evaluation of Data Set**  
  
  
From the correlation matrix of the dataset after the reclass, it appears Shell Weight is the most correlated feature, and the most negatively correlated feature is Sex.  
  
  
Looking at the distribution of classes [2], most of the records are in class 2 followed by class 3. Intuitively, this makes sense as the majority will exist within the average adult age whereas less will make it to older age due to dying from predators, statisticians etc.  
  
The highest count [4] is in infant, which following on from the previous discussion about age makes sense (older Abalone dying), although it isn’t heavily distributed towards Infant which suggests that there are other factors that affect sexual maturity in Abalone. This is further supported by looking at the age classes present in the Infant sex. The oldest age class 4 is comparatively represented in Infant as Female and Male.

By observing the distributions of Shell Weight [8], it supports the general idea that the heavier the Shell, the more likely the Abalone is older. With most of Class 1 sitting in the 0.0 – 0.15 range, Class 2 in the 0.15 – 0.35 range, Class 3 in the 0.2 – 0.4 range, and Class 4 in the 0.3 – 0.5 range. However, all the distributions are positively skewed with relatively distributed values which suggests that other factors affect age. This is further supported by the Correlation Matrix. Although Shell Weight was the most strongly correlated, it still only had a correlation value 0.64.

**Output of the Full Decision Tree made of all the features**



The Decision Tree produced with the test data and no limitations is quite large with many branching nodes. The depth of the tree is over 20 and the number of nodes is over 1000. [10][11]

Some IF and THEN statements from the start node of the tree.

IF Shell Weight <= 12  
THEN   
 IF Shell Weight <= 0.07  
 THEN  
 IF Shell Weight <= 0.04

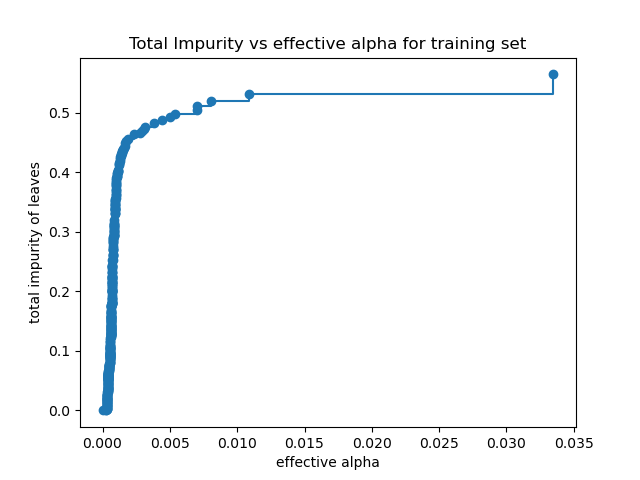
ELSE  
 IF Sex <= 1.5   
ELSE  
 IF Shell Weight <= 0.36  
 THEN   
 IF Shell Weight <= 0.18  
 ELSE  
 Shucked Weight <= 0.6

Shell Weight, the most correlated feature, is the first deciding feature of the data set. There are a few decision splits that are determined off Shell Weight which makes sense. Since it has the strongest correlation, it must therefore have a lot of information to delineate the splits initially.

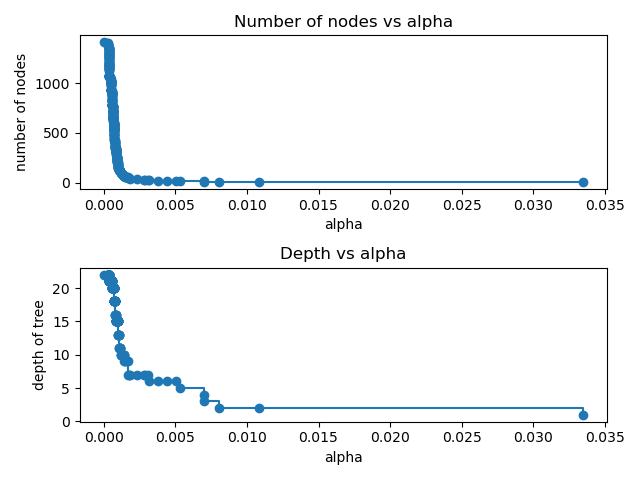
However, one of the core features of a Decision Tree is its ability to be interpreted at this Full Tree is quite complicated for a non-technical user to utilise.

Furthermore, observing the accuracy score of the training data and test data:

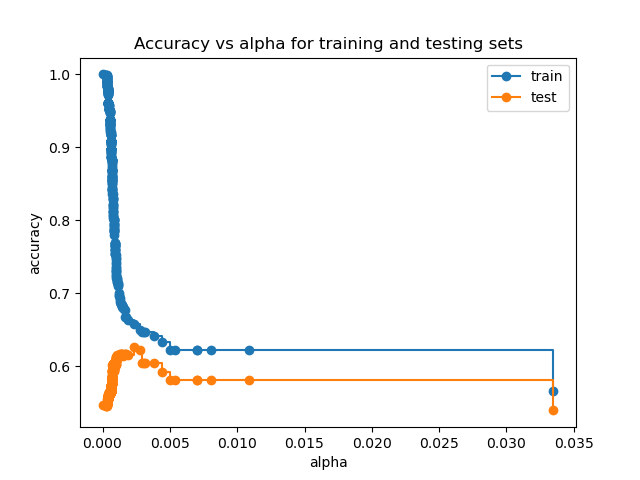
100% Accuracy with the training data but only 55% accuracy with the test data suggests that the model is probably overfitted to the training data.

**Tree Pruning Process**

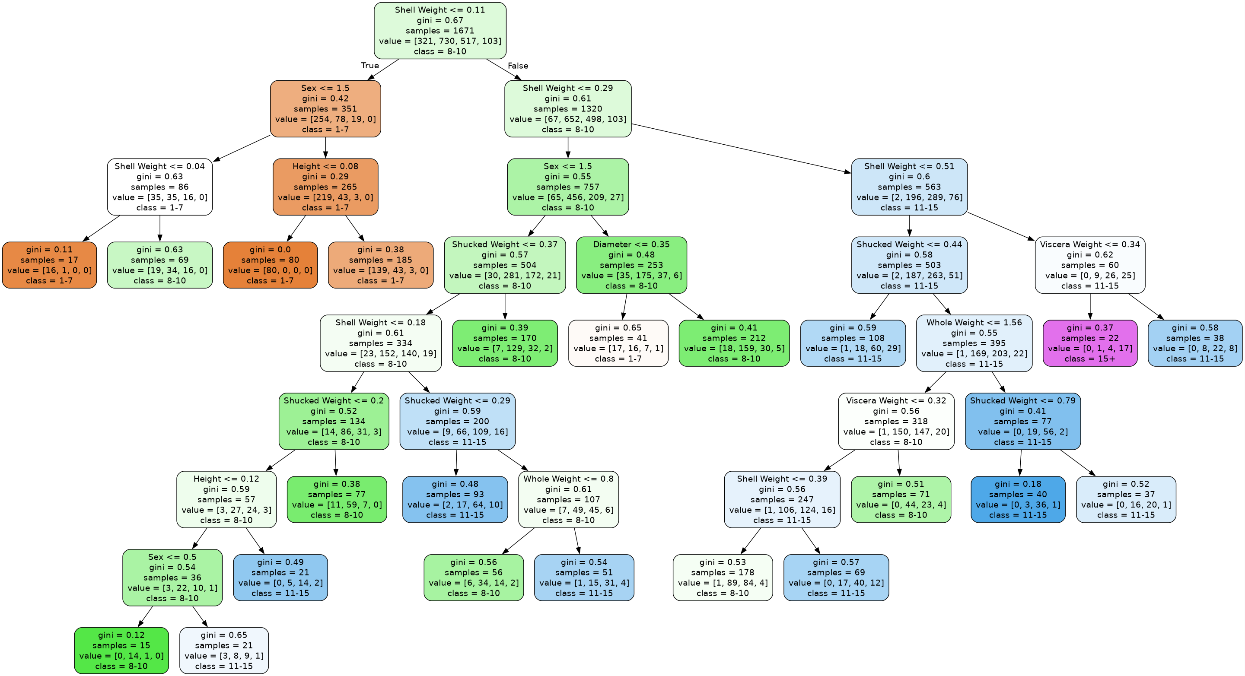
Minimal cost complexity pruning recursively finds the node with the “weakest link”. The weakest link is categorised by an effective alpha value, the nodes with the lowest effective Alpha value are pruned first [6]. In the graph above, as effective alpha increases more of the tree is trimmed but so does the overall “impurity” of the leaves, which means that at those leaves more of a generalisation is being made into what the is the class. At an effective alpha value (ccp\_alpha in scikit-learn) is 0.05 more than 95% of the nodes have been pruned.





Furthermore, as ccp alpha increases to 0.05 the number of nodes decrease from over 1000 to less than 100 and the depth reduces from over 20 to 5. More of the tree leaves are being generalised which will reduce the overall chance of overfitting.

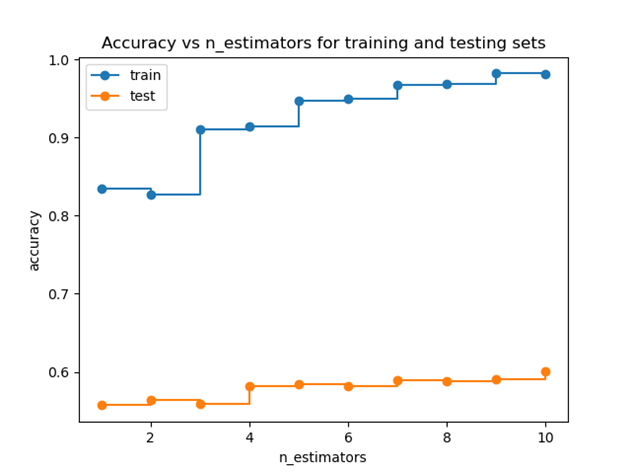
When mapping the accuracy score of the training data and test data as effective alpha increases, the training data sharply reduces accuracy. However, the test data accuracy improves. Although there is a limit to this as when more and more of the tree gets pruned away and the leaves classifications become more and more generalised it starts affecting the accuracy performance for both data sets.

  
From the training – test sample set, the best ccp\_alpha value (which produced the highest testing result) was 0.00235. It produced the following tree:

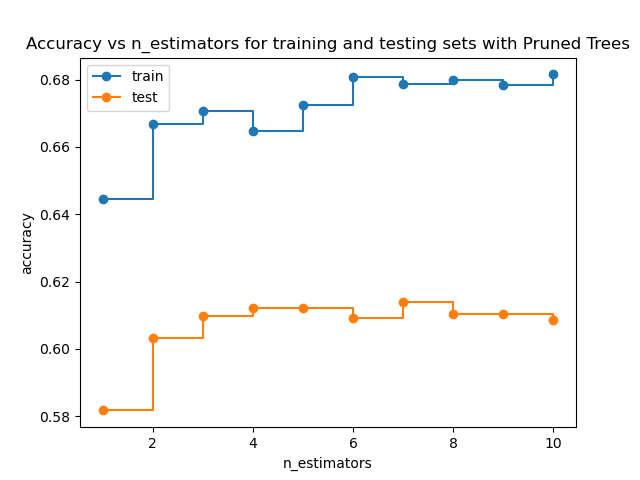
There is significantly less nodes and the depth are also reduced. Besides generalised performance, this kind of tree is more suitable for a regular person to use in the age identification process.

**Random Forest**

Random forests [3] are a modelling technique that utilises mixed sampling of the training set (called bagging) to produce multiple trees. The consensus of the trees is the output classification of the overall model.



Using a similar approach to the optimal ccp\_alpha value, the accuracy score of the train and test data is measured as the number of tree models in the forest go up. In general, as the number of trees models in the forest go up, so do the accuracy score for both training and testing data. The training data approaches 100% whereas the testing data approaches 60% as n -> 10.  
  
However, based on the previous results with the Full Tree and Pruned Tree, perhaps the random forest would benefit from using a ccp\_alpha value on the model especially since a training data accuracy score near 100% suggested that the model was perhaps overfitting.

**Pruned Forest**   
  
Using the best ccp\_alpha value from the pruned tree result, the n\_estimators and accuracy scores were measured for the training and test data sets.

As with the pruned tree, ccp\_alpha limitations drastically reduced the training accuracy score. However, the test data performance exceeded the test data performance in the random forest model without the ccp\_alpha limits.

Testing of performance against test and training data [1] [5] [7]



Iterating through the random\_state values 1-10 for the train\_test\_split, these are the accuracy scores for the training and test data as well as the the best ccp\_alpha value for the pruned tree. The Random Forest and Pruned Random Forest results were produced from models where n\_estimators = 10. From the repeat experiments it appears in general:

1) A pruned tree outperforms a full tree with test data  
2) A random forest model outperformed a full tree with test data and the confidence interval for the test data is smaller  
3) The pruned random forest outperformed the random forest with the test data albeit with a larger confidence interval  
4) The pruned tree outperformed the ensemble models with the test data albeit with a larger confidence interval

The first three points about the general performance of the models against each other is what one might expect based on the classification model literature. However, the biggest takeaway from the results that the random forest benefited from the pruning but was outperformed by the single pruned tree. This result seemed to be explained by looking at the variation of best ccp alpha results. For some samples, the best ccp alpha value was as low as 0.001404 and as high as 0.003694. Reflecting on the node and depth vs ccp alpha graphs in the results show how strongly the affect the increase of ccp alpha value is on the pruning of a tree. These results would suggest that trees in the random forest would benefit from individual best ccp alpha value calculations and the overall best measure overgeneralised some of the trees in the forest. However, the overall model benefitted from some reduction from ccp alpha.

**Conclusion**

None of the models provided strong classifications for the test data set which supports the findings of the UCI Irvine description of the dataset, that ‘further information, such as weather patterns and location (hence food availability) may be required to solve the problem’.  
  
The analysis did provide a strong support for otherwise understood features of the models used: Unrestricted decision trees can overfit, ensemble methods such as random forest reduce variance in results at the cost of interpretability and pruning (and over model generalisation) improves testing performance.

In further testing, it would perhaps be more beneficial to run a best ccp alpha value for the overall random forest like what was done above for the pruned tree model or build the individual trees with specified best ccp alpha values for each.

**References**

[1] “CONFIDENCE function “, support.microsoft.com. <<https://support.microsoft.com/en-au/office/confidence-function-75ccc007-f77c-4343-bc14-673642091ad6>> (accessed Nov 3, 2023)

[2] “matplotlib.pyplot.bar” matplotlib.org. <<https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.bar.html>> (accessed Nov 3, 2023)

[3] “sklearn.ensemble.RandomForestClassifier,” scikit-learn.org. <<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>> (accessed Nov 1, 2023)

[4] “Stacked bar chart in matplotlib” python-charts.com. <<https://python-charts.com/part-whole/stacked-bar-chart-matplotlib/>> (accessed Nov 3, 2023)

[5] A. Sharma *et al*, “Adding Prediction Intervals to Tree-based models”. Medium <<https://medium.com/walmartglobaltech/adding-prediction-intervals-to-tree-based-models-8ea53814a4b9>> (accessed Nov 3, 2023)

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Post pruning decision trees with cost complexity pruning,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011. scikit-learn.org. <<https://scikit-learn.org/stable/auto_examples/tree/plot_cost_complexity_pruning.html>> (accessed Nov 1, 2023)

[7] K. Polimis, A. Rokem, and B. Hazelton, "Confidence Intervals for Random Forests in Python," Journal of Open Source Software, vol. 2, no. 1, 2017. <<https://github.com/scikit-learn-contrib/forest-confidence-interval#readme>>

[8] M. Ginsburg, “how to plot 2 histograms side by side?”. Stack Overflow. <<https://stackoverflow.com/questions/45069828/how-to-plot-2-histograms-side-by-side>> (accessed Nov 3, 2023)

[9] N. Warwick T. Sellers, S. Talbot, A. Cawthorn and W. Ford*, “Abalone”.* UCI Machine Learning Repository.<<https://archive.ics.uci.edu/dataset/1/abalone>> (accessed Oct 27, 2023)

[10] Shubham, “Decision tree” hackerearch.com<<https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/ml-decision-tree/tutorial/>> (accessed Oct 27, 2023)

[11] W. Koehrsen, “Visualize a Decision Tree w/ Python + Scikit-Learn” Kaggle.com.<<https://www.kaggle.com/willkoehrsen/visualize-a-decision-tree-w-python-scikit-learn>> (accessed Oct 27, 2023)