Tree and Ensemble Learning

***Abstrac* —— In this option, explore various machine learning techniques to analyze and predict the multiclass Abalone dataset. The dataset's complexity necessitates the use of advanced algorithms, including decision trees, random forests, ensemble learning, and neural networks. Aim to provide a comprehensive analysis of these techniques, focusing on classification accuracy.**

I. INTRODUCTION

Machine learning techniques play a pivotal role in predictive modeling, especially when applied to complex datasets like the Abalone dataset. This dataset, sourced from the UCI Machine Learning Repository, contains various attributes of Abalone mollusks, presenting a multiclass classification problem. The challenge lies in predicting the age category of Abalone based on physical measurements.

The option focuses on employing diverse machine learning methods, such as decision trees, ensemble techniques like random forests, boosting algorithms (e.g., XGBoost, Gradient Boosting), and neural networks, to address this classification challenge.

Key objectives include exploring the dataset through visualization, analyzing decision tree performance, evaluating pruning techniques, assessing ensemble methods' scalability, and comparing the performance of neural networks with tree-based and ensemble methods.

II. DATA

A. The Dataset

This dataset offers a solution by providing a set of physical measurements of abalones that can be more easily obtained, such as length, diameter, height, and various weights, including whole weight, shucked weight (meat weight), viscera weight (gut weight), and shell weight. These measurements can potentially serve as predictors to estimate the age of abalones. The target variable for prediction is the number of rings, which, when modified by subtracting 1.5, represents the age in years.

B. Pre-processing

- Loading the data file, rename column names, convert classification values to numerical values, convert 'M' and 'F' to 0 and 1, and convert 'I' to 2.
- Convert the 'Rings' feature to categorical class labels as per the given age groups

C. Exploration

- Before modelling it is beneficial to do general data exploration to gain initial insights on the distribution and relationships of the features and response. The distribution of the age groups (classes) is visualised in Figure 1.

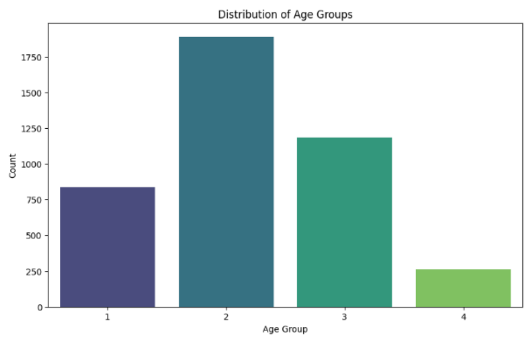

Figure 1: Age groups Distribution

- Visualize the distribution of features using histograms or boxplots

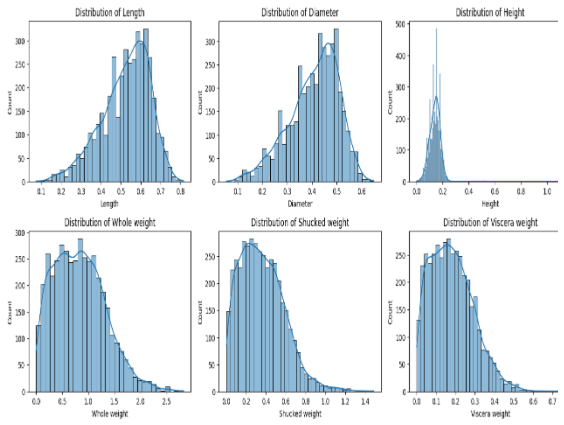

Figure 2:Distribution of feature

III. Model Considerations

A. Decision Tree Model

In this section, application of Decision Trees to the Abalone dataset. Decision Trees are a fundamental machine learning algorithm known for their interpretability and ability to handle both numerical and categorical data.

Step:

1)Split the data into features X and labels y.

2)For each experimental run, perform a new train/test split to ensure different datasets.

3) For each experimental run, create a Decision Tree Classifier.Vary tree depth from 2 to 6.

4)Train and Evaluate the Model, record and store the performance metrics for each run.

5)Repeat steps 1-4 for 30 experimental runs, each with a different train/test split and hyperparameters.

6)Visualize the structure of the best Decision Tree.Translate selected nodes and leaves into IF-THEN rules.

This process ensures robust evaluation of the Decision Tree model with different data splits and hyperparameter settings.

- Visualize the Best Tree:

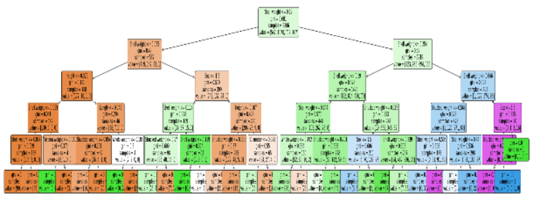

Figure 3: tree_plot

7)Uses cost-complexity pruning to generate a series of Decision Tree classifiers with different alpha values.Then evaluates and plots the testing accuracy for each alpha value. The resulting plot can help you identify the optimal alpha value that balances model complexity and accuracy. The plt.savefig line saves the plot as an image file. The code uses scikit-learn for the Decision Tree classifier and matplotlib for plotting.

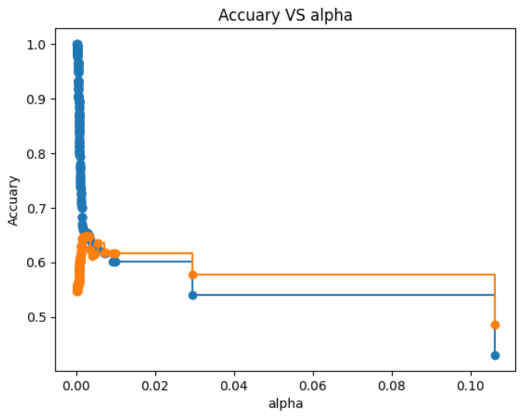

Figure 4: Accuary VS alpha

B. RandomForest Model

Random Forest is an ensemble learning method based on constructing multiple decision trees and combining their results to improve predictive performance.Random Forests often excel in various machine learning tasks, especially when dealing with high-dimensional data and large datasets. By adjusting parameters such as the number of trees (n_estimators) and others, you can effectively improve the model performance.

The modeling process is roughly consistent with the decision tree model mentioned above:

1)Define a Range for n_estimators

2)Train Random Forest with Different n_estimators Values and Record Accuracy

3)Plot the Accuracy Scores vs. the Number of Trees

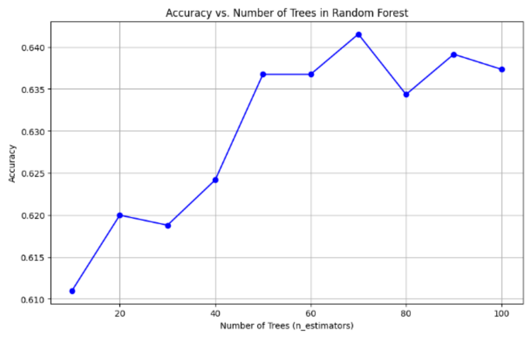

Figure 5: Accuracy vs. Number of Trees

C. *GradientBoosting Model and xgboost Model*

Gradient Boosting is a machine learning technique used for both classification and regression tasks. It is an ensemble learning method that builds a predictive model in the form of an ensemble of weak learners, typically decision trees. The idea behind Gradient Boosting is to iteratively improve the model's predictions by focusing on the mistakes made in previous iterations.

XGBoost is an optimized and efficient implementation of Gradient Boosting. It is particularly popular in machine learning competitions and real-world applications due to its speed and performance. XGBoost builds upon the principles of Gradient Boosting

The steps for modeling the random forest mentioned above are consistent. Train the Gradient Boosting models and XGBoost models and visualize the results.

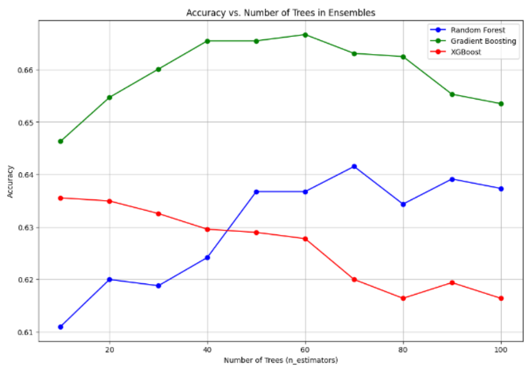

Figure 6: Accuracy vs. Number

D. *Neural networks*

Multi-Layer Perceptron (MLP) is a type of artificial neural network that consists of multiple layers of interconnected nodes, or neurons. It is a feedforward neural network, meaning that information flows through the network in one direction, from the input layer to the output layer. MLPs are versatile and can be applied to various tasks, including classification and regression. They are part of the broader category of deep learning models and have been successfully used in diverse domains such as image recognition, natural language processing, and speech recognition.

Step:

1)Use a loop to iterate over different hidden layer sizes.

2)For each hidden layer size:Create an MLPClassifier with the Adam optimizer and train it on the training data.Record the accuracy score on the test data and append it to the adam_accuracy_scores list.

3)Repeat the process for the SGD optimizer.

4)Create a plot of accuracy scores against the number of hidden layers for both optimizers.

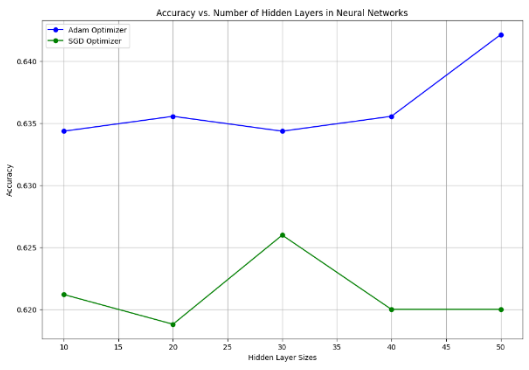

Figure 7: Accuracy vs. Number

E. Reporting Results

From the results of accuracy score and model stability, the performance of the neural network is the best.

(test data)	accuracy	Std
Decision Tree	0.6201	0.0093
post-pruning	0.6231	0.0121
RandomForest	0.6279	0.0093
GradientBoosting	0.6352	0.0116
xgboost	0.6291	0.0096
Neural networks(adam)	0.6423	0.0089

F. Apply the above steps for the Abalone regression problem.

Select the neural network model with the best performance in classification tasks, perform regression training on the dataset, and report the R2 and RMSE of the model. The modeling steps are roughly consistent with neural network classification.

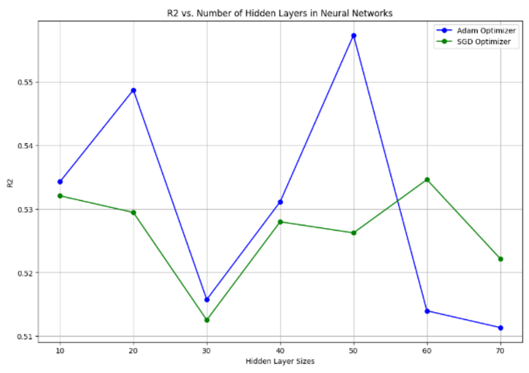

Figure 8: R2 vs. Number

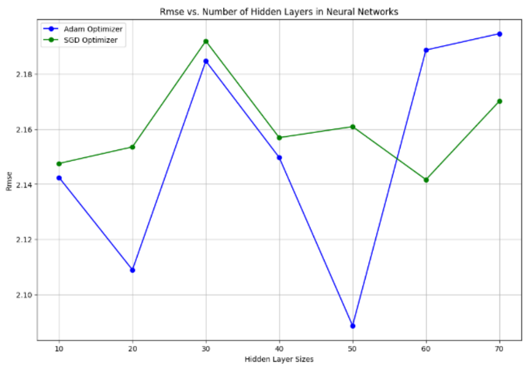

Figure 9: RMSE vs. Number

G. Reporting Results vs linear regression model from earlier Assessment

By comparing the results, it can be seen that in regression prediction, the R2 score of the neural network is higher than that of linear regression, and the RMSE score is lower than that of linear regression, reflecting that the neural network model has higher fitting degree and model interpretability than linear model

linear regression	score	Std	Neural Network	score	Std
Average RMSE (Train)	2.18	0.03	Average RMSE (Train)	2.11	0.05
Average RMSE (Test)	2.23	0.05	Average RMSE (Test)	2.16	0.06
Average R-squared (Train)	0.54	0.01	Average R-squared (Train)	0.57	0.02
Average R-squared (Test)	0.52	0.02	Average R-squared (Test)	0.55	0.02

IV Further development

In general, XGBoost (eXtreme Gradient Boosting) is superior in performance compared to regular Gradient Boosting. But in this option,Gradient Boosting performs better than XGBoost, and this could be due to the following reasons

- Data Scale: For small datasets, the advantages of XGBoost might not be as pronounced. XGBoost typically shows more significant performance improvements on large-scale datasets, leveraging distributed computing and parallel processing to enhance training speed.
- Hyperparameter Tuning: XGBoost has more hyperparameters to tune, and for some specific problems, GB may perform well with fewer hyperparameter adjustments. If proper tuning is not performed, XGBoost might not fully exploit its advantages.
- Problem-specific Structure: For certain problems, especially simpler ones, the complexity of XGBoost may not bring additional benefits to the model. In contrast, GB might be simpler, more straightforward, and more suitable for some problems.

ADAM performs better than SGD in neural network classification,and this could be due to the following reasons

- Adaptive Learning Rates: Adam dynamically adjusts the learning rates for each parameter based on their past gradients. This adaptability helps Adam to converge faster and more effectively, especially in situations where the data has varying scales or when different features have different importance.
- Momentum: Adam incorporates the concept of momentum, which helps the optimization process by considering past gradients. This can be beneficial for overcoming local minima or navigating through flat regions of the loss landscape.
- Bias Correction: Adam includes bias correction terms, which can be particularly useful during the early stages of training. It helps in stabilizing and speeding up the convergence process.
- Combination of Adaptive Methods: Adam combines ideas from both momentum-based methods and RMSprop (Root Mean Square Propagation), making it a powerful optimization algorithm. It addresses some of the limitations of individual methods and offers a balanced approach.