Model Comparison for Multiclass Classification: An Ensemble Learning Study on UCI Datasets

Participant 1
Jie Wu
z5432579

Participant 2
Yirou Li
z5563505

Participant 3
Hanlin Miao
z5501850

***Abstract*—This paper explores multiclass classification by applying and evaluating a range of machine learning models on the UCI Abalone dataset, addressing the challenge of accurately predicting abalone age categories. Models tested include Decision Trees, Random Forests, Gradient Boosting, XGBoost, and Neural Networks with optimizers like SGD and Adam. To optimize model performance, we experiment with various hyperparameters and apply techniques such as pre-pruning and post-pruning in Decision Trees, along with L2 regularization and dropout in Neural Networks. Our primary aim is to compare and identify models that consistently perform well for multiclass classification, based on metrics such as accuracy, F1 score, and ROC-AUC. To verify model consistency and adaptability, we apply the best-performing models to a different dataset, the Contraceptive Method Choice dataset, showcasing cross-domain effectiveness. We further validate model generalizability by selecting an additional UCI dataset for visualization and evaluation, reporting error metrics for train and test sets. Our contributions include a comprehensive performance analysis of ensemble and neural network models for multiclass classification and guidance on model selection. This work enhances the understanding of model suitability for multiclass problems, offering valuable insights for researchers and practitioners in machine learning.**

I. Introduction

Machine learning methods are increasingly being applied to solve classification problems in various fields such as healthcare, finance, and environmental science [1] [2]. Past classification algorithms, such as Decision Trees, Naive Bayes, and k-Nearest Neighbors (k-NN), are known for their simplicity and efficiency. Decision Trees split data using rules, Naive Bayes assumes feature independence and is suited for small datasets [3], and k-NN classifies based on nearest samples [4]. These methods worked well for small datasets and resource-limited settings. In recent years, deep learning and ensemble methods have become popular, enhancing classification for complex data. Deep learning models (e.g., CNNs, RNNs) perform well in image and text classification [5] [6], while ensemble methods improve generalization by combining multiple weak classifiers and fitting large-scale, high-dimensional data effectively [7].

Neural networks and time series problems faced several challenges, especially when handling large datasets and complex patterns. For neural networks, overfitting was a persistent issue as the models tended to fit too closely to training data, reducing their accuracy on new data [8]. Additionally, hyperparameter tuning required significant computational resources and expertise, as even slight adjustments could greatly affect performance [9].

Another issue lies in data preprocessing and feature selection. For high-dimensional data, if appropriate feature engineering is not applied, model complexity may increase, affecting classification performance. Additionally, redundant or irrelevant features can cause fluctuations in model performance. Therefore, it is essential to carefully select and process features to ensure that the model can effectively extract key information from the data.

The motivation for this project arises from the need to identify the most suitable model for multiclass classification tasks, as limited research has focused on systematically comparing various models for this purpose. Although numerous algorithms have demonstrated strong potential in different applications, there is insufficient work exploring how specific model characteristics and parameters affect performance in multiclass settings, especially with complex and high-dimensional datasets. By comparing and analyzing multiple models, we aim to provide insights into selecting and optimizing models that are best suited for robust multiclass classification in practical applications.

In this project, we investigate multiclass classification of abalone age, using physical measurements to classify ages into four groups. We start by analyzing and visualizing the abalone dataset, including class and feature distributions. Using decision trees, we test multiple hyperparameter settings, visualizing the best tree and translating selected nodes into decision rules. We then assess model improvements through pruning methods to optimize performance. Following this, we explore the use of ensemble methods, applying Random Forests to measure how performance changes with varying numbers of trees. We further compare these results with XGBoost and Gradient Boosting models and conduct additional experiments using simple neural networks (Adam/SGD) to compare against ensemble results, testing the effect of L2 regularization and dropout. We report on accuracy, AUC, and F1 scores to capture a comprehensive view of model performance. To assess model generalizability, we apply the two best-performing models to a different dataset from the UCI repository, providing visualizations and reporting key metrics to evaluate model transferability. Finally, we select an Wine dataset from UCI for further visualization and model building, capturing error metrics on both training and test sets for a thorough understanding of model robustness across datasets.

The rest of the paper is organized as follows. In Section 2, we describe the dataset and experimental methodology,

detailing data preprocessing steps and the design of our classification models. Section 3 provides the results of our experiments, including performance metrics and analysis for each model. In Section 4, we discuss the implications of our findings, comparing the effectiveness of different algorithms. Finally, we conclude the paper in Section 5, summarizing key insights and suggesting directions for future research.

II. Methodology

A. Data

- **Part A** In part A, we use the UCI abalone dataset includes 4177 samples, which presents total nine attributes of abalone: sex, length, diameter, height, whole weight, shucked weight, viscera weight, shell weight and the number of rings. We divides the dataset into four classes according to the age of the abalone, which calculated by adding 1.5 to the number of rings.
- **Part B** In part B, the UCI dataset we used is a contraceptive method choice dataset, which includes 1473 samples. The dataset presents total ten attributes: wife's age, wife's education, husband's education, number of children ever born, wife's religion, wife's now working, husband's occupation, standard-of-living index, media exposure and contraceptive method used, which is the class attribute.
- **Part C** In part C, we choose the white wine dataset with 4898 samples from the UCI dataset. The dataset shows twelve attributes: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol and quality, which is the class attribute.

B. Overview of Methods

We use five models in our research: Decision Tree models, Random Forests models, XGBoost models, Gradient Boosting models and Deep Learning models and these models are used for three classification tasks. For the Machine Learning models, we train and test performance for multiple experiences by applying a range of hyperparameters to identify the models with optimal performance. And for the Deep Learning models, we try different weight decay, dropout, and learning rates to optimize the models' performance.

C. Software Suite

We train the Decision Tree models, Random Forests models and Gradient Boosting models by using Python's scikit learn library. For the XGBoost models, we use Extreme Gradient Boosting library. And the Neural Networks are trained by using Pytorch. In the visualization part, we use Seaborn and Matplotlib to visualize the data.

D. Experiment Settings

In part A, the models are used to classify abalones into five classes: 0 - 7 years, 8- 10 years, 11 - 15 years and Greater than 15 years. For the Decision Tree models, we try the tree depths from 2 to 7 to find the best tree depth. And after we build the best model with the highest accuracy, we optimize the model by using post-pruning, which implement by applying multiple values of alphas in order to figure out the best alpha. We use the hyperparameters applied to build the best Decision Tree model and try different tree counts, which also used to train the XGBoost models and Gradient Boosting models, to find out the Random Forests model with the highest accuracy. Then the Deep Learning models we use are Multi-Layer Perceptron models and we introduce two different solver: Stochastic Gradient Descent (SGD) and Adam to optimize the model. We conduct hyperparameter tuning on the neural network, and try different weight decay, dropout, and learning rates to identify the best model. Accuracy scores and F1 scores are the metrics that we use to compare the performance of different models and we applied two of the best models in part B. For part C, we use all models except for the Neural Network model with the SGD optimizer.

III. Results

A. Part A Data Analysis

For the data in Part A, we first generated a correlation heatmap (Figure 1), where the color indicates the magnitude of the correlation coefficient. We observe that the features most correlated with `rings-age` are `Diameter` and `Shell_weight`. Next, we created scatter plots for these two features (Figures 2 and 3). The results indicate that `Diameter` and `Shell_weight` can serve as good predictors of abalone age, but the relationship is not entirely linear, especially in high-value regions, where additional factors may need to be considered for accurate predictions.

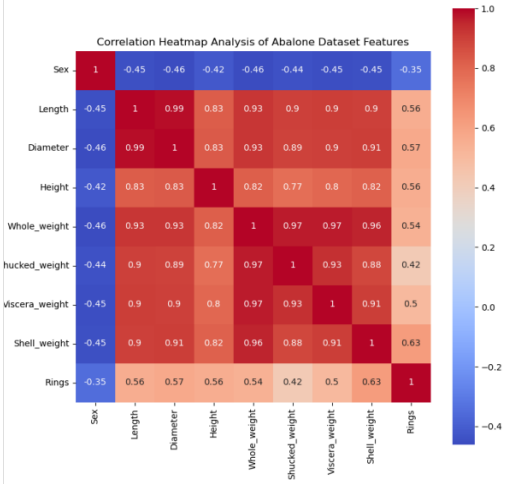

Fig. 1: Correlation Heatmap of Part A Dataset

We then plotted histograms for `Diameter`, `Shell_weight`, and `rings` (Figures 4, 5, and 6). From these histograms, we can see that abalone diameter and age are primarily concentrated in the medium range, while shell weight shows a clear right-skewed distribution. We categorized `rings` into four groups: 0-7, 8-10, 11-15, and above 15, and plotted a pie chart (Figure 7) to observe the distribution. The 0-7 category has the largest proportion, while the above-15 category has the smallest, indicating an imbalance in the data.

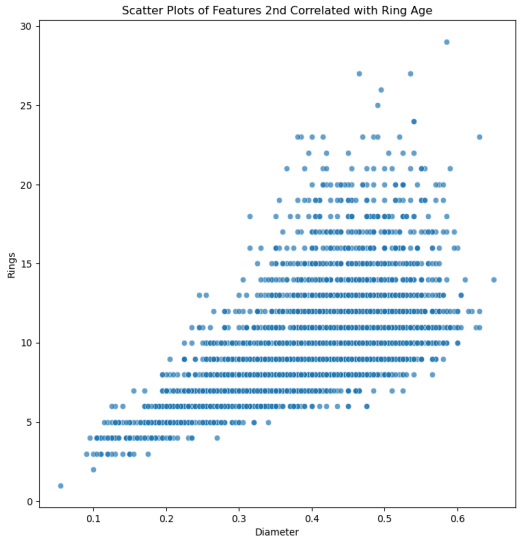

Fig. 2: Scatter Plot of Diameter vs Rings-Age

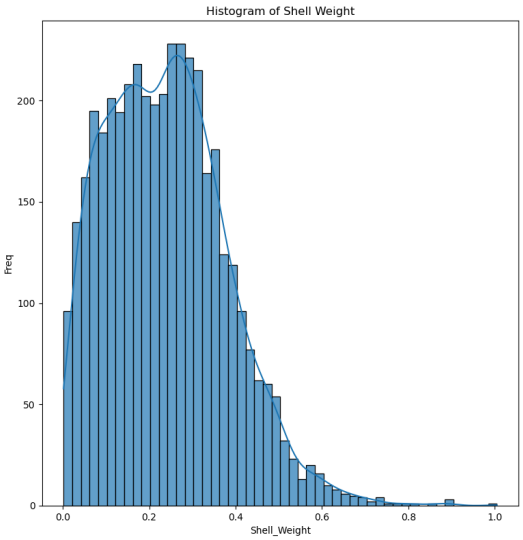

Fig. 5: Histogram of Shell Weight

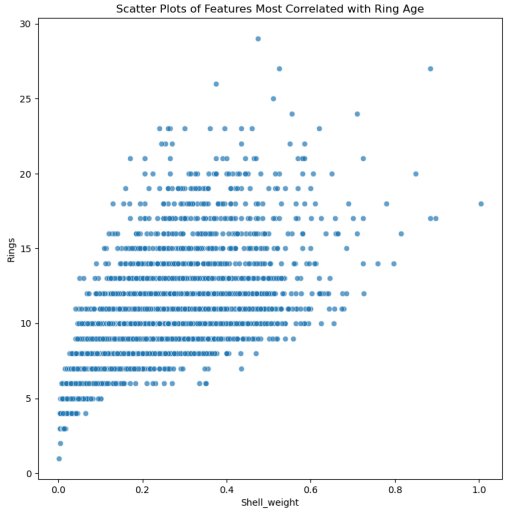

Fig. 3: Scatter Plot of Shell Weight vs Rings-Age

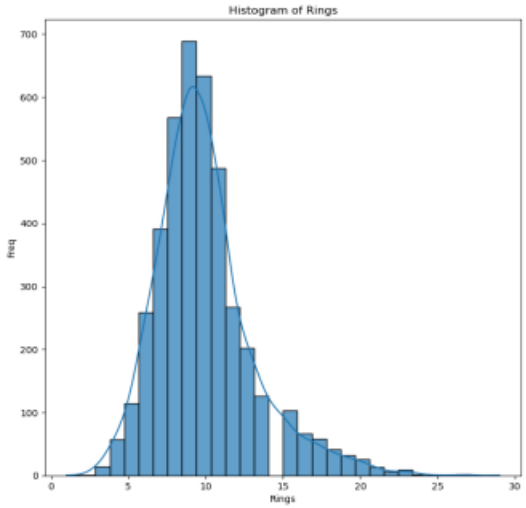

Fig. 6: Histogram of Rings

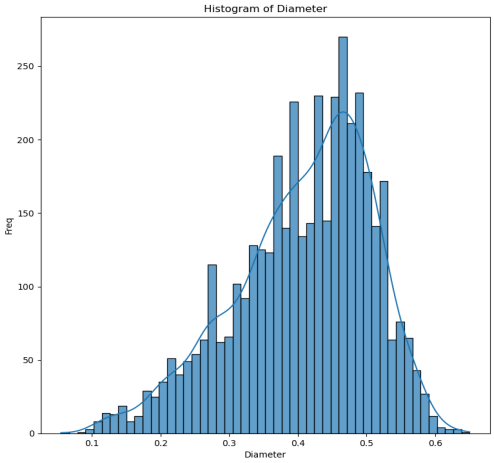

Fig. 4: Histogram of Diameter

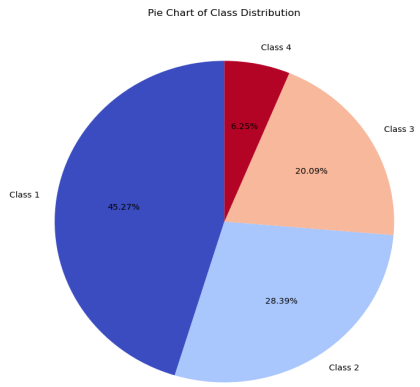

Fig. 7: Pie Chart of Rings-Age Distribution

B. Part B Data Analysis

In Part B, we performed data visualization for the ”Contraceptive Method Choice” dataset. First, we generated a heatmap of the dataset features (Figure 8). In this heatmap, `Wife_Age`, `Wife_Religion`, and `Media_Exposure` are negatively correlated with `contraceptive_method`, while the other features show positive correlations. The two most correlated features are `Wife_Age` and `Wife_Education`.

We displayed the distribution of each feature as a histogram (Figure 9) and created a pie chart for the ”Contraceptive Method Class” distribution (Figure 10). We found that family sizes (number of children) are concentrated in a few specific numbers, and contraceptive method usage is unevenly distributed. Both husband and wife tend to have relatively high education levels, the living index is generally high, and media exposure is limited. The wife’s age and religion are concentrated in certain groups, and the female employment rate is low. The pie chart shows a fairly balanced distribution in terms of contraceptive method choice, with the highest proportion being those not using any method, followed by long-term methods and finally short-term methods.

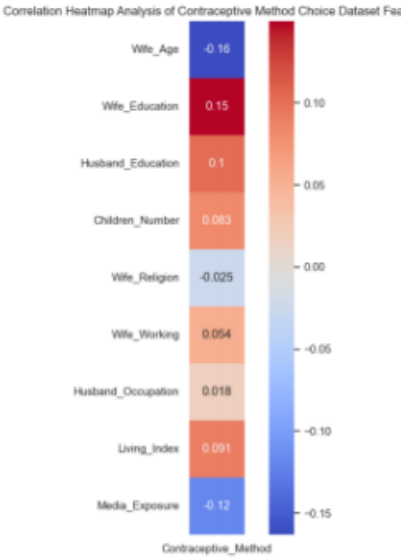

Fig. 8: Heatmap of Contraceptive Method Choice Dataset Features

C. part A Modelling and Predictions

In Part A, we first used the Decision Tree model for training. We experimented with different maximum tree depths to determine the optimal depth for pre-pruning. By plotting the tree depth against accuracy (Figure 11), we found that a tree depth of 6 provided the best model performance. We then generated a visualization of the optimal decision tree and translated a few selected nodes and leaf nodes into IF-THEN rules (Figures 12 and 13). It can be seen that other features have limited differentiation across age groups, so we continued to adjust the `ccp_alpha` parameter to further enhance the differentiation of age groups based on other features. Below is

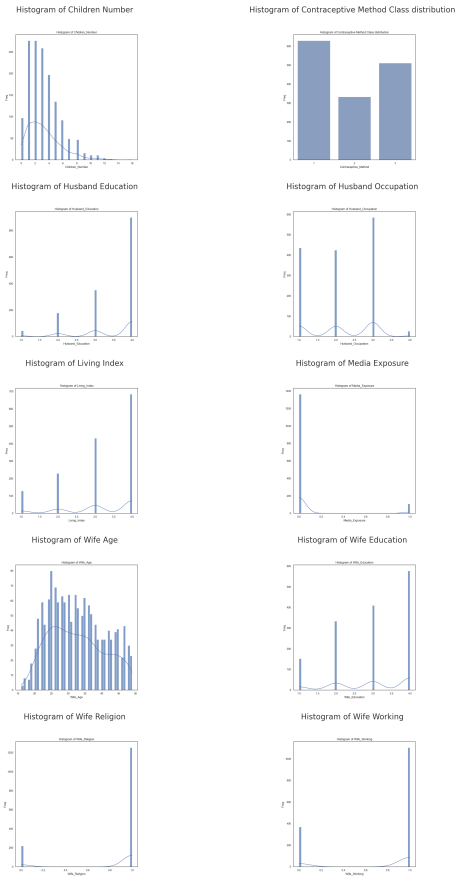

Fig. 9: Histograms of Each Feature in the Contraceptive Dataset

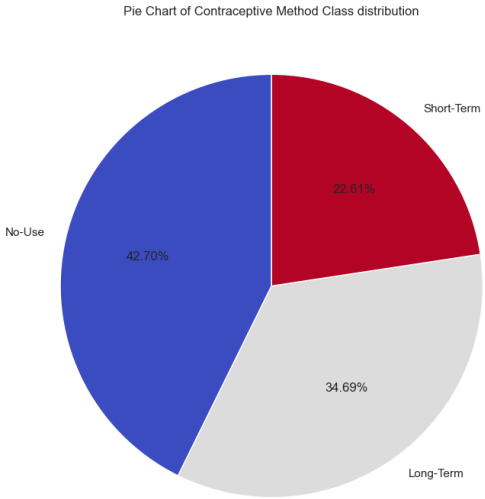

Fig. 10: Pie Chart of Contraceptive Method Class Distribution

the accuracy plot for different alpha values (Figure 14), where we observe that an alpha value around 0.001 yields the best results. The visualization of the optimal decision tree (Figure 15) shows that the distinctions between `rings-age` based on different features are significantly improved. These results indicate that, for this decision tree model, the best `alpha` is 0.001, and the best tree depth is 6.

We then used random forest, XGBoost, and gradient boosting models to train the dataset and plotted the accuracy values of these three models in the range 0-1000 with different number of decision trees as a line chart. It can be seen from the figure that the accuracy of random forest and XGBoost increases rapidly when the number of decision trees is small and maintains a high accuracy when the number of decision trees is large, while the accuracy of gradient boosting decreases significantly after the number of decision trees reaches a certain level.

We then use Adam and SGD simple neural networks to train the dataset and compare the results with the first three models. As shown in (Figure 16), Adam's accuracy is between random forest and XGBoost, while sgd's accuracy is in the lower range.

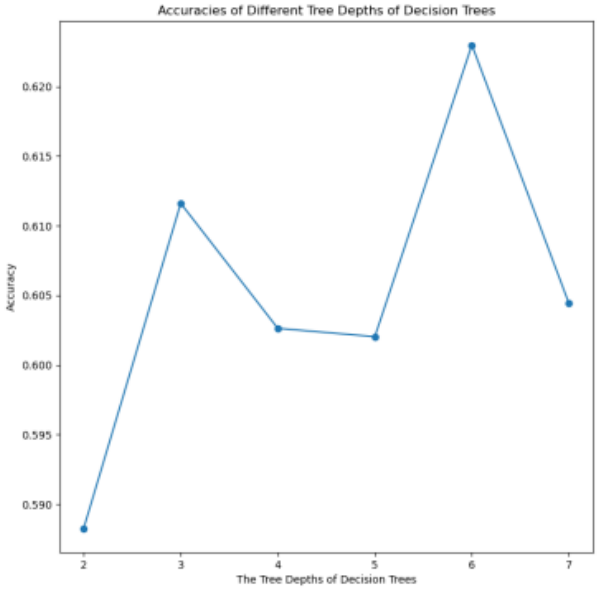

Fig. 11: Tree Depth vs. Accuracy Plot

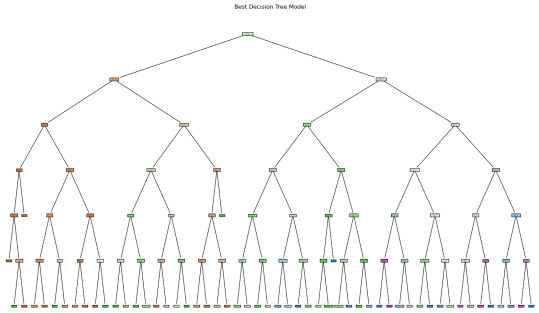

Fig. 12: Visualization of Optimal Decision Tree

Fig. 13: IF-THEN Rules for Selected Nodes

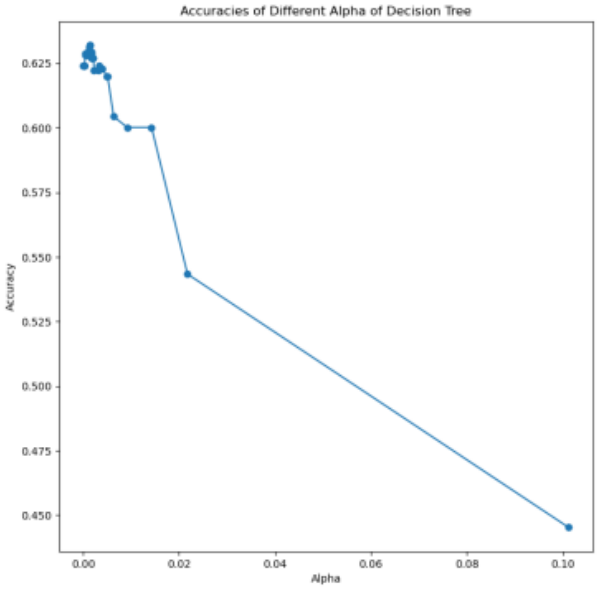

Fig. 14: Accuracy Plot for Different Alpha Values

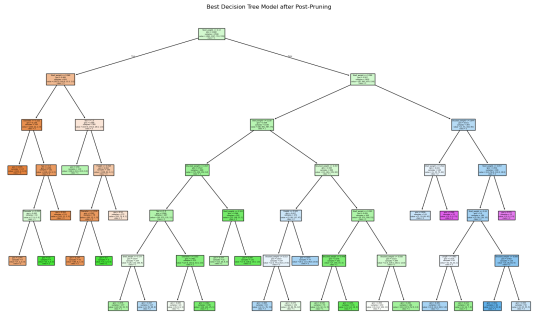

Fig. 15: Visualization of Decision Tree with Optimal Alpha

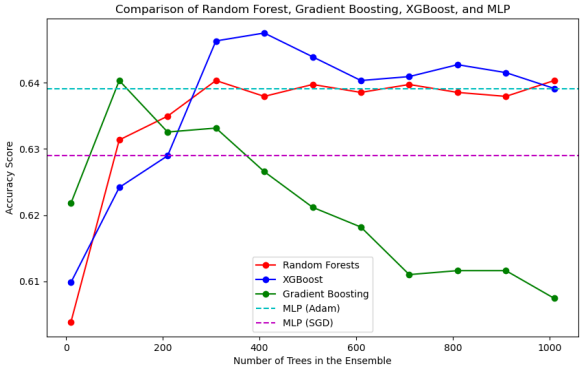

Fig. 16: Comparison of Adam and SGD Neural Networks with Previous Models

Finally, we built an Adam model with three different parameter settings, yielding the following results:

- Dropout Rate: 0.1, Weight Decay: 0.0001, Learning Rate: 0.001, Test Accuracy: 0.65
- Dropout Rate: 0.2, Weight Decay: 0.0001, Learning Rate: 0.001, Test Accuracy: 0.64
- Dropout Rate: 0.3, Weight Decay: 0.001, Learning Rate: 0.005, Test Accuracy: 0.55

D. part B Modelling and Predictions

After performing data visualization, we selected the Decision Tree with a tree depth of 7 and the XGBoost model based on the F1 Scores and Accuracy comparisons from Part A. These models were used to train and predict on the dataset. Additionally, we visualized the Decision Tree model (Figure 17) and obtained the following results:

Decision Tree F1-Score: 0.511 ROC-AUC Score: 0.679

XGBoost with Best Number of Trees F1-Score: 0.477 ROC-AUC Score: 0.691

From the F1 Score and ROC-AUC Score, we observe that the final training results meet expectations, which confirms the validity of the model selection process in Part A.

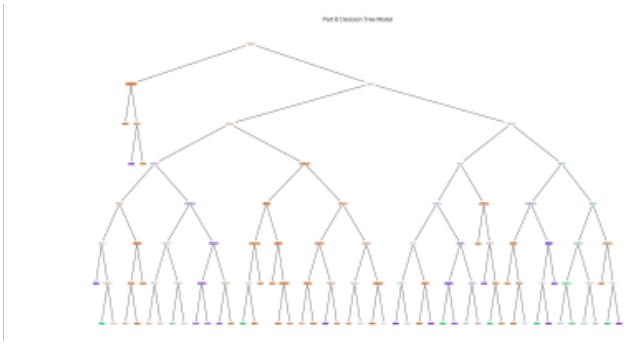

Fig. 17: Visualization of Decision Tree Model

E. Part C Analysis

In Part C, we continued with the previous approach by first generating a correlation heatmap for the dataset features (Figure 18). We found that the two features most correlated with `quality` are `density` and `alcohol`. Therefore, we generated histograms for `quality`, `density`, and `alcohol` (Figure 19), as well as a pie chart for `quality` (Figure 20). From these visualizations, we can see that quality scores and density are relatively concentrated, while alcohol content has a wider, right-skewed distribution.

Next, we trained the dataset using a decision tree model. We first observed the accuracy variations on the training and test sets with different tree depths (Figure 21). We visualized the best-performing decision tree model, which has a depth of 8 (Figure 22). We then plotted the accuracy of the decision tree at different `alpha` values to facilitate post-pruning (Figure 23), and found that the optimal `alpha` is 0.001. We further generated a visualization of the post-pruned optimal decision tree model (Figure 24) and the confusion matrix of the model on the training and test sets (Figure 25 and Figure 26).

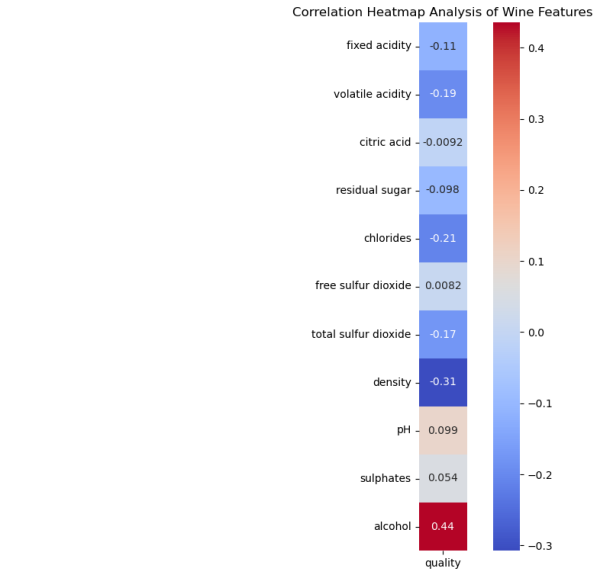

Fig. 18: Correlation Heatmap of Dataset Features

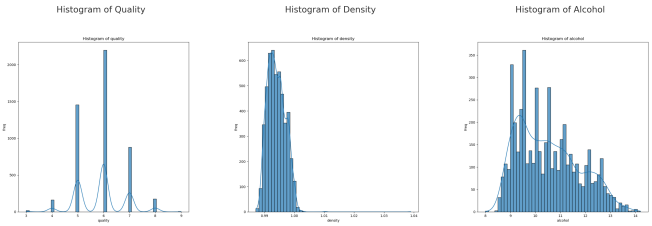

Fig. 19: Histograms of Quality, Density, and Alcohol

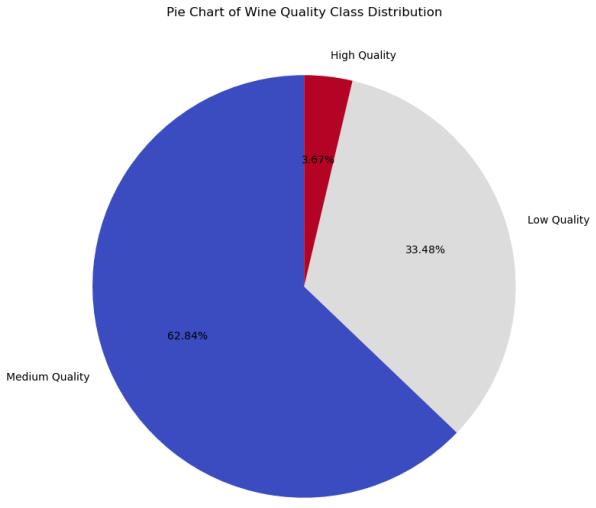

Fig. 20: Pie Chart of Quality Distribution

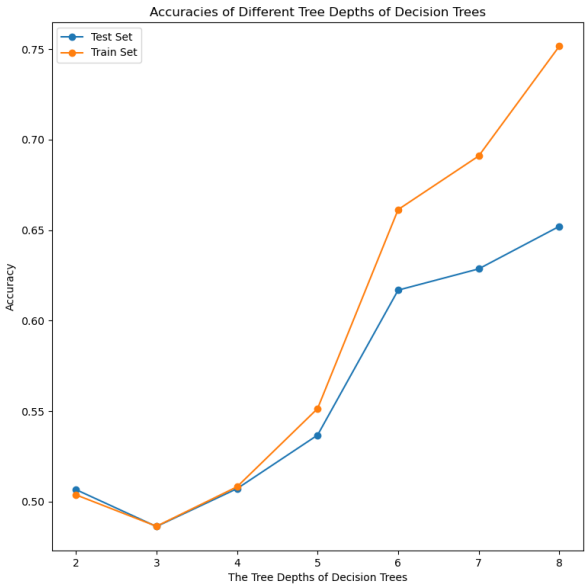

Fig. 21: Accuracy Variation with Different Tree Depths

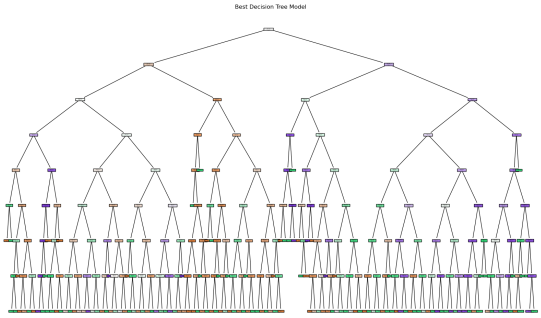

Fig. 22: Visualization of Optimal Decision Tree (Depth=8)

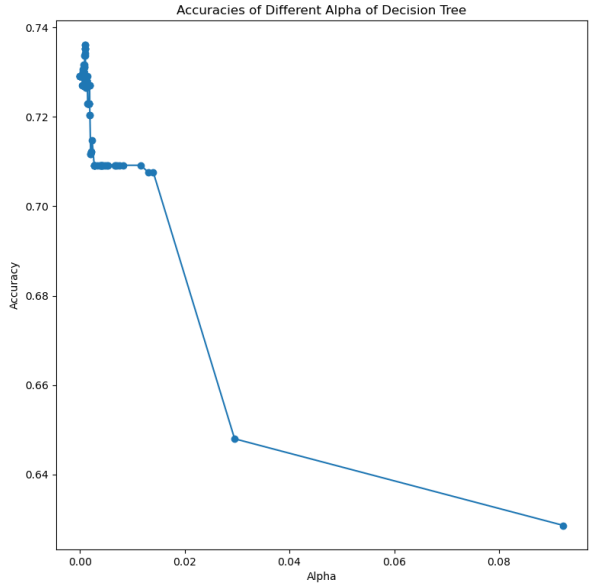

Fig. 23: Accuracy with Different Alpha Values for Post-Pruning

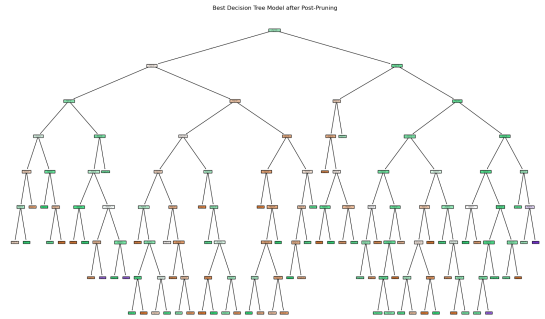

Fig. 24: Visualization of Post-Pruned Decision Tree Model

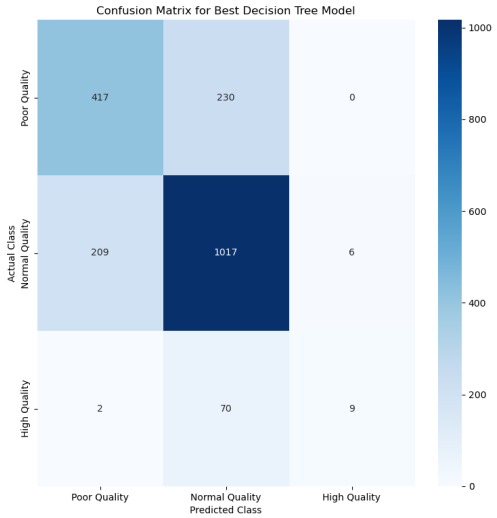

Fig. 25: Confusion Matrix of Wine in Decision Trees (Test Set)

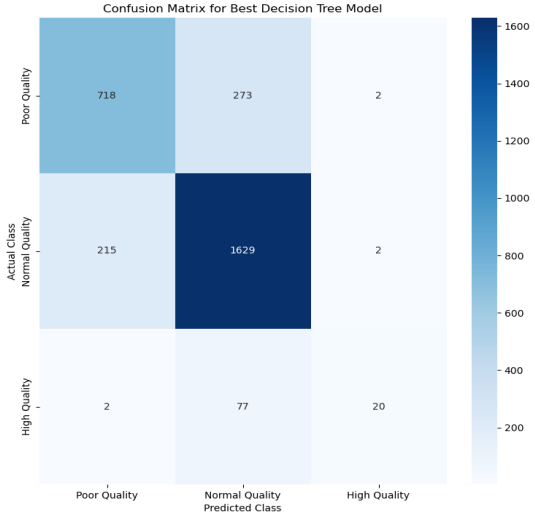

Fig. 26: Confusion Matrix of Wine in Decision Trees (Train Set)

We then used a Random Forest model for training, observing the accuracy variations on the training and test sets with the number of decision trees ranging from 1 to 10 Figure 27. The results indicate that the optimal number of trees is 8. We also generated a confusion matrix for the Random Forest model (Figure 28 and Figure 29) to analyze the model. The results show that the accuracy of the test and training sets stabilizes after adding about 5 trees, and the model performs best in identifying wines of medium quality, with considerable confusion in the distinction between poor and medium quality.

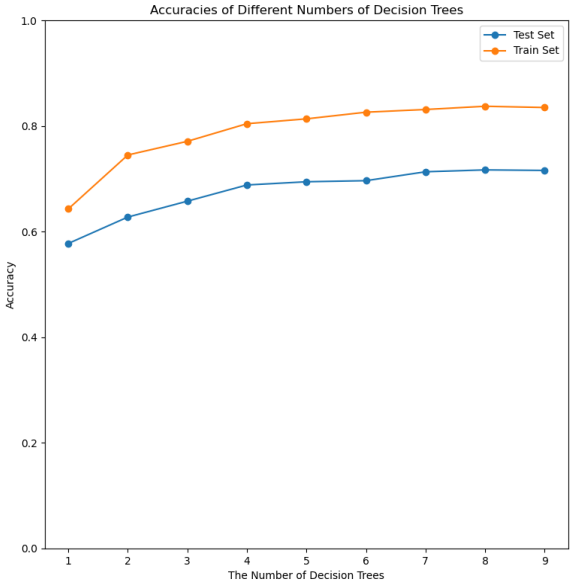

Fig. 27: Accuracy Variation with Different Tree Counts in Random Forest

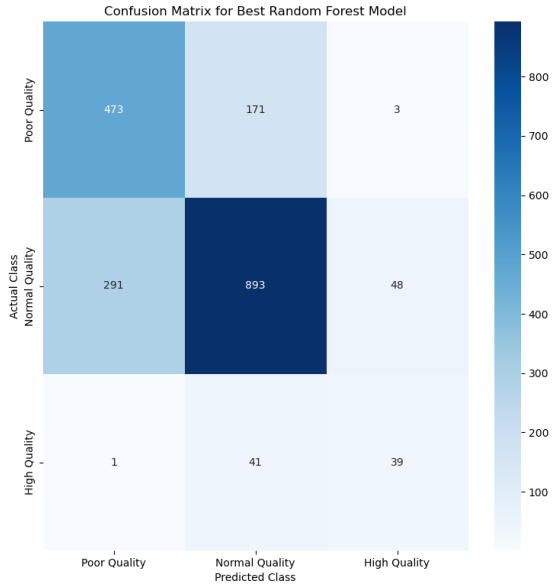

Fig. 28: Confusion Matrix of Random Forest Model (Test Set)

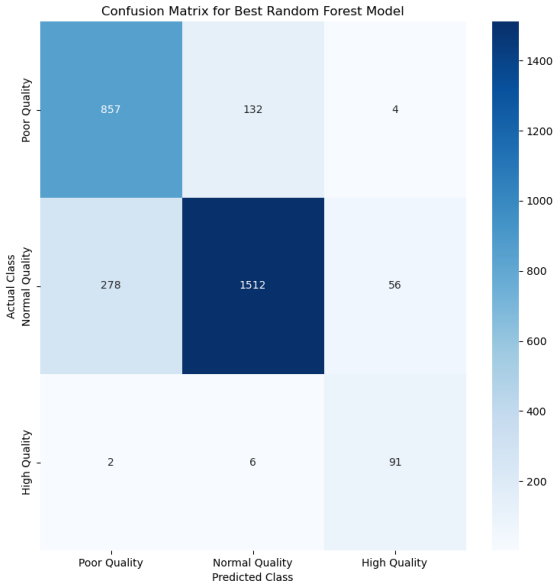

Fig. 29: Confusion Matrix of Random Forest Model (Train Set)

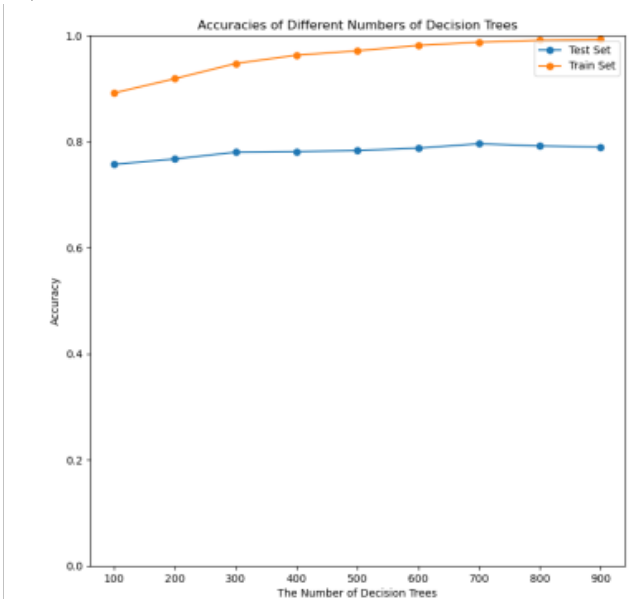

Fig. 30: Accuracy Variation with Different Tree Counts in XGBoost

We then trained the dataset using an XGBoost model, plotting the accuracy variations on the training and test sets with the number of decision trees ranging from 100 to 1000 (Figure 30). The optimal number of trees was found to be 700. A confusion matrix for the XGBoost model was also generated (Figure 31 and 32) to analyze the model. The XGBoost model has a good performance when dealing with this dataset, especially in preventing overfitting. However, it may be necessary to further adapt the model or use different feature selection methods to improve the recognition of high-quality wines.

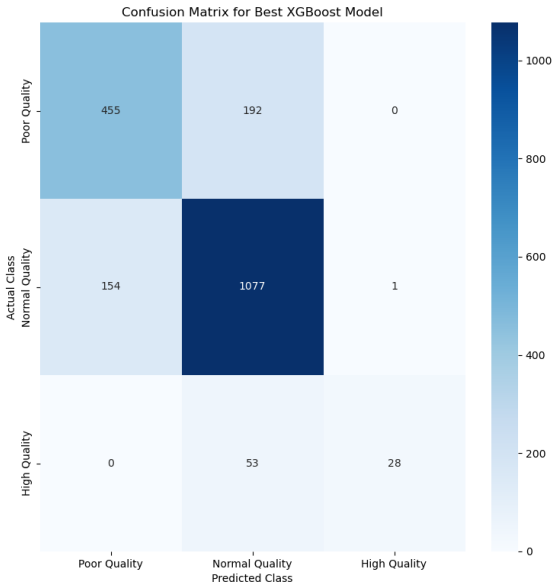

Fig. 31: Confusion Matrix of XGBoost Model (Test Set)

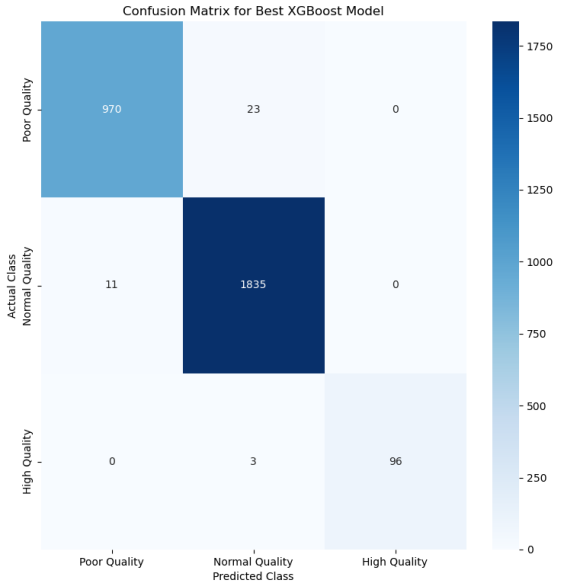

Fig. 32: Confusion Matrix of XGBoost Model (Train Set)

Next, we trained the Gradient Boosting model, examining the accuracy variations on the training and test sets with the number of decision trees from 100 to 1000 (Figure 33). The optimal number of trees was found to be 900. We generated a confusion matrix for the Gradient Boosting model (Figure 34 and Figure 35) to analyze the model, showing that The gradient boosting model shows good classification performance, especially on medium quality wine classification. The recognition accuracy of higher quality wines is lower.

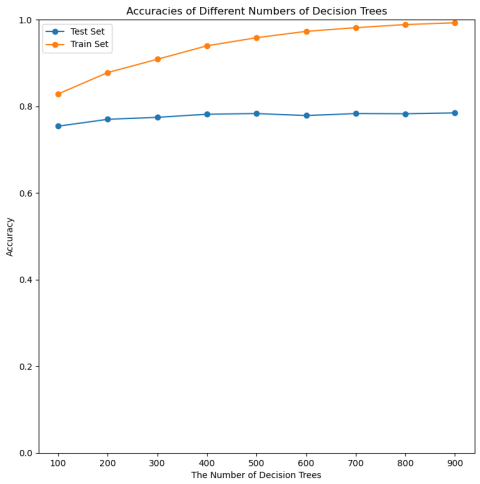

Fig. 33: Accuracy Variation with Different Tree Counts in Gradient Boosting

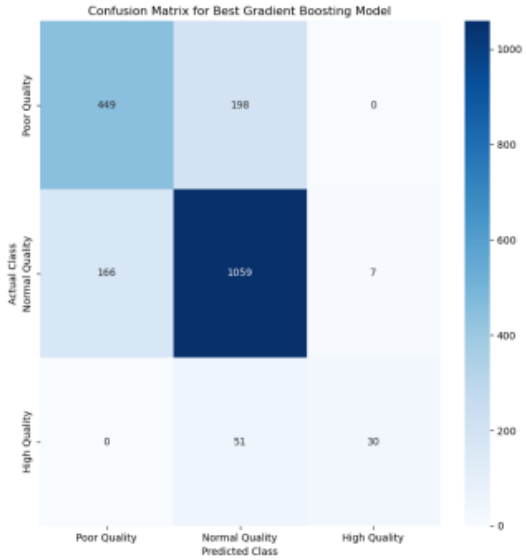

Fig. 34: Confusion Matrix of Gradient Boosting Model

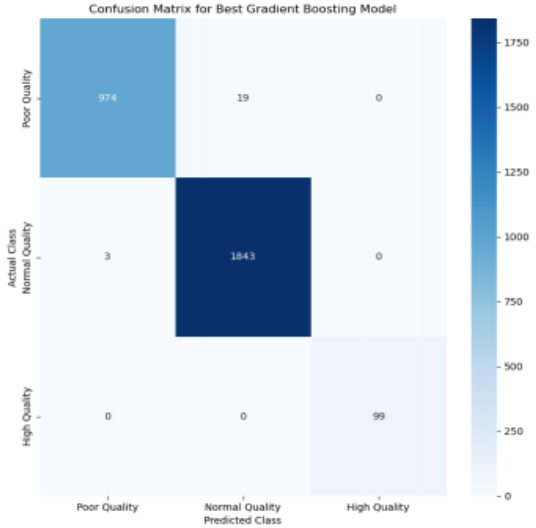

Fig. 35: Confusion Matrix of Gradient Boosting Model

Finally, we trained a simple neural network model using the Adam optimizer. After training for 1000 epochs, we obtained the model's performance metrics and generated plots of the loss variation and accuracy variation on the training and test sets (Figures 36 and 37), as well as a confusion matrix for the Adam model (Figure 38 and 39). From these plots, we can observe that The deep learning model using Adam optimizer performs well on wine quality classification task with stable learning and prediction ability. However, there is room for further optimization, especially in reducing misclassification of medium and premium wines.

Fig. 36: Loss Variation of Adam Model on Training and Test Sets

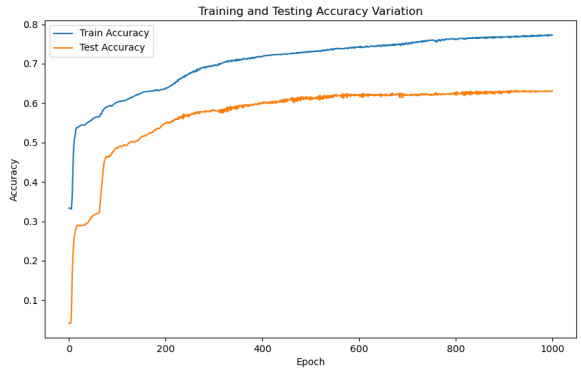

Fig. 37: Accuracy Variation of Adam Model on Training and Test Sets

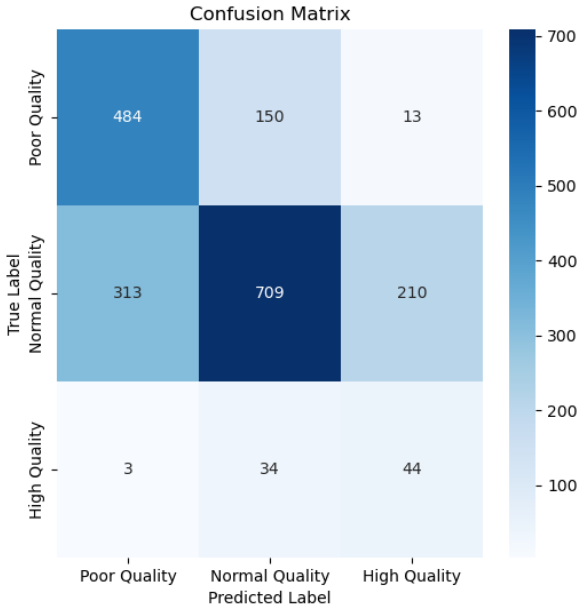

Fig. 38: Confusion Matrix of Adam Model

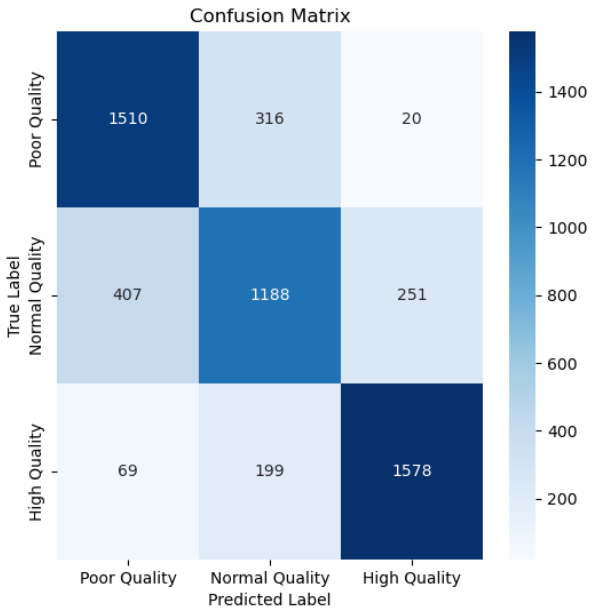

Fig. 39: Confusion Matrix of Adam Model

IV. Discussion

Our experiments aimed to validate the effectiveness of machine learning models on multiple datasets. In Part A, we categorized abalone age ranges and compared decision trees, random forests, XGBoost, and neural networks in multiclass classification. Part B focused on the Contraceptive Method Choice dataset, where we visualized the data and assessed model performance using F1 score and ROC-AUC. Finally, Part C involved the wine quality dataset, where we evaluated model performance on training and test sets. Overall, the results met our expectations and highlighted each model's strengths and limitations across diverse datasets.

A. Summary of Results

- **Data Visualization**
  In the Data Visualization section, we observed through experiments that for Part A, the two features most related to the outcome are shell weight and diameter. Among them, diameter is approximately normally distributed, while shell weight shows a right-skewed distribution. From the related images in the target, rings, we can see that this type has an uneven distribution. For Part B, the heatmap shows that the correlation between features and the outcome is not very strong, and the histogram indicates that these features are discrete numerical data with an uneven distribution. For Part C, the heatmap shows that the two features most related to the outcome are alcohol and density. Among them, density is approximately normally distributed, while alcohol shows a right-skewed distribution. From the images related to the target, Quality, we can see this type has an uneven distribution. This dataset shares similarities with Part A.
- **Modeling**

TABLE I: Comparison of Accuracy Results

Model	PartA	PartB	PartC
Decision Tree	0.631	-	0.652
Post-pruning	0.631	-	0.736
Random forest	0.640	0.524	0.717
XGBoost	0.642	0.471	0.796
Gradient Boosting	0.640	-	0.785
SGD	0.629	-	-
Adam	0.639	-	0.639
Optimized Adam	0.650	-	-

The table shows accuracy results for different models across Parts A, B, and C. In Part A, the highest accuracy was achieved by Optimized Adam at 0.650, followed by XGBoost at 0.642 and Random Forest at 0.640. For Part B, the two best models and their parameters from Part A (XGBoost and Random Forest) were used, resulting in accuracies of 0.524 for Random Forest and 0.471 for XGBoost. In Part C, XGBoost achieved the highest accuracy at 0.796, followed by Gradient Boosting at 0.785 and Post-pruning Decision Tree at 0.736. Overall, XGBoost consistently performed well, particularly in Part C, while Random Forest also showed strong performance in Part A.

B. Implications of the results

- **PartA**
  In Part A, we initially used a random forest model with an accuracy of only 0.631. To improve performance, we applied post-pruning to simplify the model by removing unimportant features and reduce overfitting. However, accuracy did not improve significantly, likely because these pruned features had minimal impact on predictions. They were either redundant or had low influence due to data limitations or imbalanced samples, leading the model to ignore them even before pruning.
  Using the Random Forest model, we applied the depth of the best Decision Tree and adjusted the number of trees to optimize performance. While increasing the number of trees improved stability, accuracy saw little gain due to feature limitations. This suggests that simply adding trees has limited effect, and further improvement would need more feature engineering or a more complex model structure.
  Apart from that, using another ensemble model XGBoost, we tuned the learning rate, selecting 0.01 as the best option. This moderate rate provided stable training and reduced overfitting, leading to higher accuracy. This result suggests that XGBoost handles data with strong feature correlations well, especially with fine-tuning for complex relationships.XGBoost excels with data that has strong feature correlations due to its additive model and forward stepwise algorithm, which iteratively optimizes each tree based on previous residuals, effectively reducing training error. Its built-in regularization helps prevent overfitting, providing stability on high-dimensional, strongly correlated data. Additionally, XGBoost's flexible parameter tuning, such as adjusting learning rate and tree depth, allows for further optimization. These features make XGBoost the best model for this dataset.
  For the optimized Adam, the best parameters are increasing weight decay to 0.001, raising the dropout rate to 0.3, and setting the learning rate to 0.005. The increased weight decay effectively controls model complexity, reducing overfitting and helping the model avoid local minima during training. A higher dropout rate randomly removes neurons in each iteration, preventing over-reliance on specific neurons and enhancing the model's generalization ability. The higher learning rate accelerates training convergence, allowing the model to reach an optimal solution more quickly. Together, these adjustments lead to significant improvements in both training and test performance, offering better stability and prediction accuracy compared to the non-optimized Adam model. Without these optimizations, the model is more prone to overfitting and slower convergence, whereas the optimized Adam performs better across multiple tasks.
- **PartB**

In Part B, we applied the best parameters from Part A directly, using 110 trees for Random Forest and 1010 trees for XGBoost. Random Forest achieved an accuracy of 0.524, an F1 score of 0.491, and a ROC-AUC of 0.699, reflecting moderate discrimination but limited precision and recall. XGBoost produced similar results, with an accuracy of 0.471, F1 score of 0.481, and ROC-AUC of 0.698, indicating stable performance but limited gains despite the large tree count.

Overall, both models showed similar performance, constrained by weak feature correlations. While Part A's parameter settings provided a starting point, further model-specific adjustments and refinements in the code would be beneficial to enhance adaptability and improve performance on this dataset.

- **PartC**

  In Part C, we initially explored a decision tree model with various depths, but accuracy gains were constrained, likely due to redundant features and features with weak predictive power for the target variable. Simply increasing model depth did not enhance performance, as the model began capturing noise and overfitting. To improve generalization, we applied post-pruning, which reduced unnecessary branches and simplified the model. This approach led to an increase in accuracy from 0.652 to 0.736, with the F1 score adjusting slightly droped from 0.561 to 0.547. Post-pruning allowed the tree to retain only branches that offered substantial predictive value, filtering out minor patterns that contributed little to classification accuracy. By focusing on essential features, the pruned model balanced complexity and accuracy, offering better reliability for test data without excessive depth.

  For the random forest model, we applied the best depth from the decision tree model and adjusted the number of estimators to optimize accuracy. Although increasing the number of trees improved model stability, it did not substantially enhance accuracy, suggesting that simply enlarging the ensemble has limited effect without additional feature engineering or complexity adjustments. For XGBoost, we tuned parameters like the learning rate, finding 0.01 to be optimal. This model performed well on the dataset, as it shares similar characteristics with Part A, where feature correlations are quite significant. In summary, XGBoost performed well here for similar reasons as in Part A, leveraging strong feature correlations to achieve high accuracy through its additive and regularization mechanisms.

  For the Adam-optimized neural network model, we initially applied data preprocessing techniques such as SMOTE to address class imbalance and MinMax scaling to normalize feature values. Despite these adjustments, the model's performance remained unsatisfactory, with limited improvements in accuracy. This indicates that preprocessing alone may not be sufficient to enhance model performance in this case. To further improve results, we could explore advanced feature engineering techniques, such as applying PCA to reduce dimensionality and emphasize key patterns in the data. Additionally, experimenting with different neural network architectures, like adding hidden layers or adjusting neuron counts. Another potential approach is to apply L2 regularization, such as weight decay, to penalize large weights, which helps control overfitting and improves generalization by making the model less sensitive to specific features in the training data.

C. Limitations

- **Data Imbalance**

  For Part A and Part C, the target variable distributions were imbalanced, with certain classes significantly more prevalent than others. This imbalance may have led the models to favor the majority classes, impacting the accuracy and recall for minority classes and potentially reducing the model's overall robustness.

  To address this, we experimented with several techniques. First, we applied class-weight='balanced' in models such as Random Forest and XGBoost to automatically adjust weights inversely proportional to class frequencies. Additionally, we implemented the Synthetic Minority Over-sampling Technique (SMOTE) to create a balanced training set. However, this cannot fully address the root cause of the imbalance in data representation. The model still tended to focus more on the majority class in certain cases, which led to reduced accuracy and recall for minority classes.

- **Limited Feature Relevance and Discrete Data Challenges in Part B**

  In Part B, the dataset consists primarily of discrete, categorical features, which may not have been fully leveraged by the models used. Ensemble models like Random Forest and XGBoost generally perform well with categorical data, but the lack of continuous feature information may have restricted the models' ability to capture their relationships. Additionally, the heatmap analysis reveals a lack of strong correlations between the features and the target variable. The highest correlations are only around 0.15 for Wife Education and -0.16 for Wife Age, with other features like Husband Education, Children Number, and Living Index showing even weaker correlations.

  This weak correlation indicates that none of the features strongly influence the target variable, making it challenging for models to identify clear patterns and dependencies. In all, the predictive power of the models is limited, as they rely on strong feature-target relationships for accuracy. These observations suggest a need for additional feature, such as creating new features or combining existing ones, to capture more complex relationships and enhance model performance.

- **Model Complexity and Computational Cost**

  Optimizing hyperparameters for ensemble models like XGBoost and Random Forest required significant com-

putational resources, which limited the extent of tuning we could perform. Although we aimed to find the best possible parameters, additional adjustments might have improved results but were restricted by these computational limitations. For instance, in XGBoost, we experimented with different numbers of estimators, learning rates, and maximum depths. With adequate computational resources, we could also explore parameters like subsample and colsample-bytree to further refine and find more optimal configurations.

V. Conclusions

A. Major Contributions

In our research, the main contribution is that we successfully applied Decision Tree models, Random Forest models, XGBoost models, Grandient Boosting models and Neural Networks in three classification tasks. Random Forest, XGBoost, and Gradient Boosting models are all based on Decision Tree models and we improved model performance by using pre-pruning and post-pruning. We are surprised by the high accuracies that Random Forest models and XGBoost models achieved. In the Neural Network models, we compared two optimizers: SGD and Adam, and we found that Adam outperforms SGD in classification tasks with higher accuracy. Comparing L2 regularisation(weight decay) with dropouts techniques, we found that the model will perform better in a smaller weight decay value, which can effectively control model complexity and prevent overfitting, but with a higher dropouts rate, information loss will be caused, which reduces the model's learning ability and accuracy.

B. Directions for Future Research

For further research, we will focus on the further optimizations that can make the models more adaptable and efficient for practical applications. We could try to find and optimise the best parameter sizesso that the result in future might be better than the currentresult [10]. We also could implementmore advanced regularization and ensemble techniques to solve more complex classification tasks. Finally, we could introduce some more advanced model architectures to better perform on multi-class, multi-label classification tasks.

References

[1] A. Q., S. Rahman, J. Zhou, and J. J. Kang, "A Comprehensive Review on Machine Learning in Healthcare Industry: Classification, Restrictions, Opportunities and Challenges," Sensors, vol. 23, no. 9, pp. 4178, Apr. 2023, doi: 10.3390/s23094178.

[2] N. Nazareth and Y. V. R. Reddy, "Financial applications of machine learning: A literature review," Expert Systems with Applications, vol. 219, 2023, Art. no. 119640, doi: 10.1016/j.eswa.2023.119640.

[3] I. A. P. Banlawe, J. C. Dela Cruz, J. C. P. Gaspar, and E. J. I. Gutierrez, "Decision Tree Learning Algorithm and Naïve Bayes Classifier Algorithm Comparative Classification for Mango Pulp Weevil Mating Activity," 2021 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), Shah Alam, Malaysia, 2021, pp. 317-322, doi: 10.1109/I2CACIS52118.2021.9495863.

[4] G. Guo, X. Ping, and G. Chen, "A Fast Document Classification Algorithm Based on Improved KNN," First International Conference on Innovative Computing, Information and Control - Volume I (ICICIC'06), Beijing, 2006, pp. 186-189, doi: 10.1109/ICICIC.2006.381.

[5] Z. Zhao, "Research on single-character image classification of Tibetan ancient books based on deep learning," 2022 3rd International Conference on Computer Vision, Image and Deep Learning and International Conference on Computer Engineering and Applications, Changchun, China, 2022, pp. 1-5, doi: 10.1109/CVIDLICCEA56201.2022.9824018.

[6] L. Wenzheng and W. Jie, "A YOLOv7 Forest Fire Detection System with Edge Computing," 2023 IEEE 13th International Conference on Electronics Information and Emergency Communication (ICEIEC), Beijing, China, 2023, pp. 223-227, doi: 10.1109/ICEIEC58029.2023.10200044.

[7] U. N. A and K. Dharmarajan, "Diabetes Prediction using Random Forest Classifier with Different Wrapper Methods," 2022 International Conference on Edge Computing and Applications (ICECAA), Tamilnadu, India, 2022, pp. 1705-1710, doi: 10.1109/ICECAA55415.2022.9936172.

[8] P. Thanapol, K. Lavangnananda, P. Bouvry, F. Pinel, and F. Leprévost, "Reducing Overfitting and Improving Generalization in Training Convolutional Neural Network (CNN) under Limited Sample Sizes in Image Recognition," 2020 - 5th International Conference on Information Technology (InCIT), Chonburi, Thailand, 2020, pp. 300-305, doi: 10.1109/InCIT50588.2020.9310787.

[9] F. Sia and N. S. Baco, "Hyperparameter Tuning of Convolutional Neural Network for Fresh and Rotten Fruit Recognition," 2023 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), Kota Kinabalu, Malaysia, 2023, pp. 112-116, doi: 10.1109/IICAIET59451.2023.10291915.

[10] M. F. Misman et al., "Prediction of Abalone Age Using Regression-Based Neural Network," 2019 1st International Conference on Artificial Intelligence and Data Sciences (AiDAS), Ipoh, Malaysia, 2019, pp. 23-28, doi: 10.1109/AiDAS47888.2019.8970983.