## Introduction

Artificial Neural Networks (ANNs) constitute a key branch of machine learning, also referred to as Neural Networks (NNs). These networks are inspired by the structure of the human biological brain, aiming to replicate its functionality. ANNs consist of interconnected nodes or neurons that collaborate to execute specific tasks. Among the various types of ANNs, the Multilayer Perceptron (MLP) is a feedforward neural network capable of approximating any continuous function.

An MLP typically comprises an input layer, one or more hidden layers, and an output layer. The input layer receives information, which is then processed by the hidden layers before reaching the output layer. Depending on the algorithm employed, the output layer performs functions such as classification and regression. MLPs are powerful at addressing problems involving non-linear relationships.

In this study, we leverage an MLP to analyze the physiochemical properties (referred to as features) of red wine, aiming to discern whether they indicate good or bad quality. This classification problem falls under supervised machine learning, where the model's performance is evaluated by comparing its predictions with the actual labels. The outcome of this analysis provides insights into the MLP's effectiveness in assessing wine quality.

## Dataset

The dataset utilized in this study was sourced from Kaggle's website, and you can access it through this link. Originally obtained from the University of California, Irvine Machine Learning Repository, the dataset has undergone preprocessing. It encompasses eleven features, namely fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol. All independent variables are continuous, while the dependent variable (or label) corresponds to the wine quality, categorized into binary classes as either 'good' or 'bad.'

Comprising a total of 1,599 data instances, 80% of these instances are allocated for training the model, leaving the remaining 20% for testing purposes. Notably, the dataset exhibits no missing values. Given the sensitivity of Multilayer Perceptron (MLP) models to feature scaling, the 'StandardScaler' has been applied to normalize features, ensuring a mean of 0 and a standard deviation of 1. Aside from this normalization step, no other preprocessing procedures have been employed in this analysis.

## Analytics

During the training and testing phases of our model, we conducted experiments with three different combinations of model parameters, specifically altering the learning rate (alpha), hidden layers, and activation function. To evaluate the performance of each combination on both the training and test datasets, we employed diverse metrics, including classification accuracy, sensitivity, specificity, harmonic mean (F1 score), and logarithmic loss. The calculations for these metrics were carried out using the Scikit-Learn libraries, while the matplotlib library was utilized to create ROC curves for visualization purposes. Below, we provide a summary of the analysis results and plots for all the parameter combinations.

## Results

### Performance in the Training Dataset

| **Metric** | **Combination 1** (alpha = 0.00001, hidden layers = (5,2), activation = ReLU) | **Combination 2** (alpha = 0.00001, hidden layers = (4,3,2), activation = ReLU) | **Combination 3** (alpha = 10, hidden layers = (5,2), activation = tanh) |
|---|---|---|---|
| Accuracy | 0.7912 | 0.7858 | 0.7873 |
| Sensitivity | 0.7648 | 0.7707 | 0.7811 |
| Specificity | 0.8209 | 0.8027 | 0.7944 |
| F1 Score | 0.7948 | 0.7918 | 0.7952 |
| Log Loss | 0.4494 | 0.4744 | 0.4663 |

Table 1: Metrics for the training dataset

Combination 1 stands out with the highest accuracy, specificity, and the lowest log loss among the three. On the other hand, combination 3 exhibits the highest sensitivity and F1 score. Notably, Combination 2 falls short when considering any of the metrics for the training dataset.

A comprehensive evaluation of each model's performance involved plotting ROC curves for all combinations, and the results are attached below. The examination of the ROC curves, along with the corresponding area under the curve, reveals that Combination 1 (depicted by the blue curve) covers the largest area, indicating superior performance compared to the other combinations.

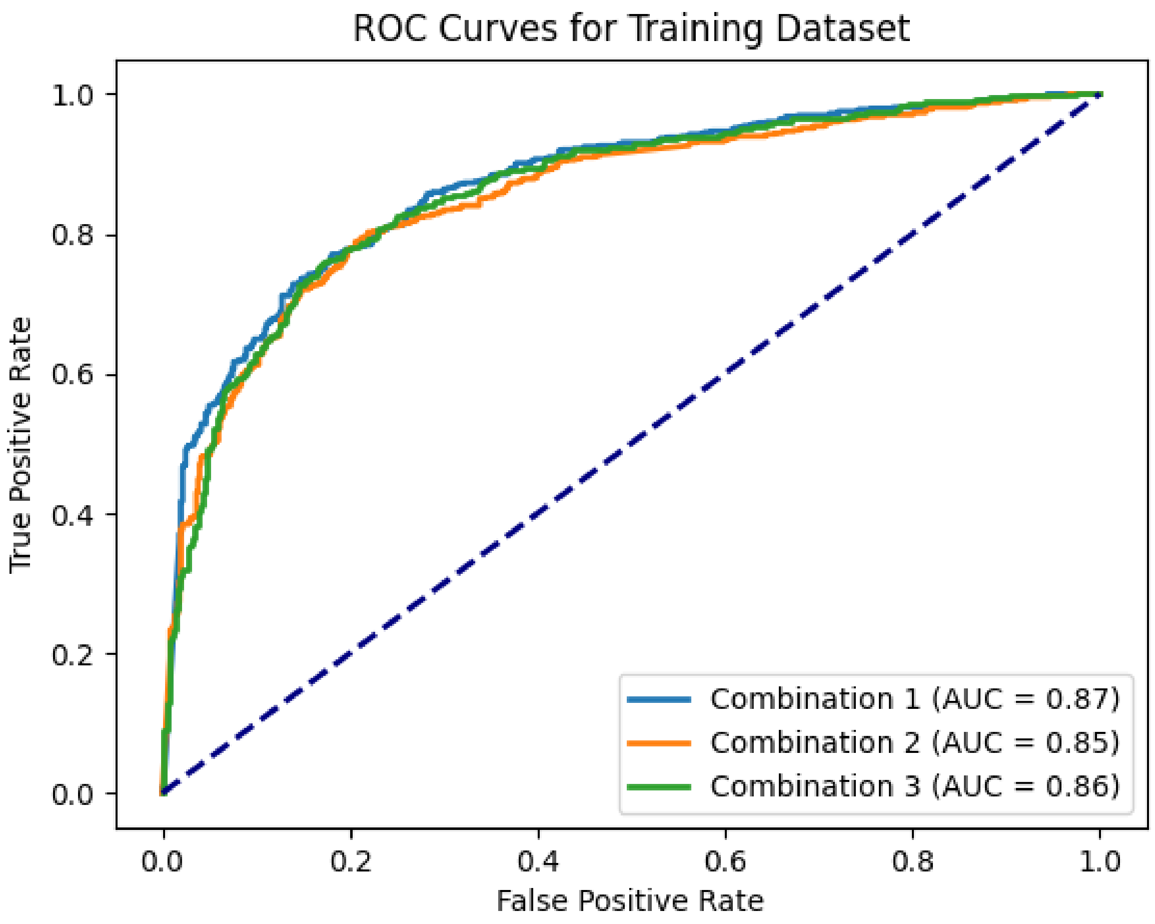

Figure 1: ROC curve for the training dataset

**Performance in the Test Dataset**

| **Metric** | **Combination 1** (alpha = 0.00001, hidden layers = (5,2), activation = ReLU) | **Combination 2** (alpha = 0.00001, hidden layers = (4,3,2), activation = ReLU) | **Combination 3** (alpha = 10, hidden layers = (5,2), activation = tanh) |
|---|---|---|---|
| Accuracy | 0.7250 | 0.7500 | 0.7625 |
| Sensitivity | 0.6927 | 0.7486 | 0.7654 |
| Specificity | 0.7660 | 0.7518 | 0.7589 |
| F1 Score | 0.7381 | 0.7701 | 0.7829 |
| Log Loss | 0.5655 | 0.5127 | 0.4853 |

Table 2: Metrics for the test dataset

Combination 3 demonstrated superior performance compared to the others, boasting the highest accuracy, sensitivity, F1 score, and the lowest log loss. The only metric where other combinations excel is specificity. Based on these results, Combination 3 emerges as the most suitable fit for this dataset, showcasing a commendable ability to generalize well on unseen data.

In alignment with the training dataset evaluation, the plot below illustrates ROC curves for all combinations on the test dataset. Consistent with the metrics, the ROC curve for Combination 3 (depicted by the green curve) stands out, featuring the highest area under the curve and further emphasizing its robust performance compared to the alternative combinations.

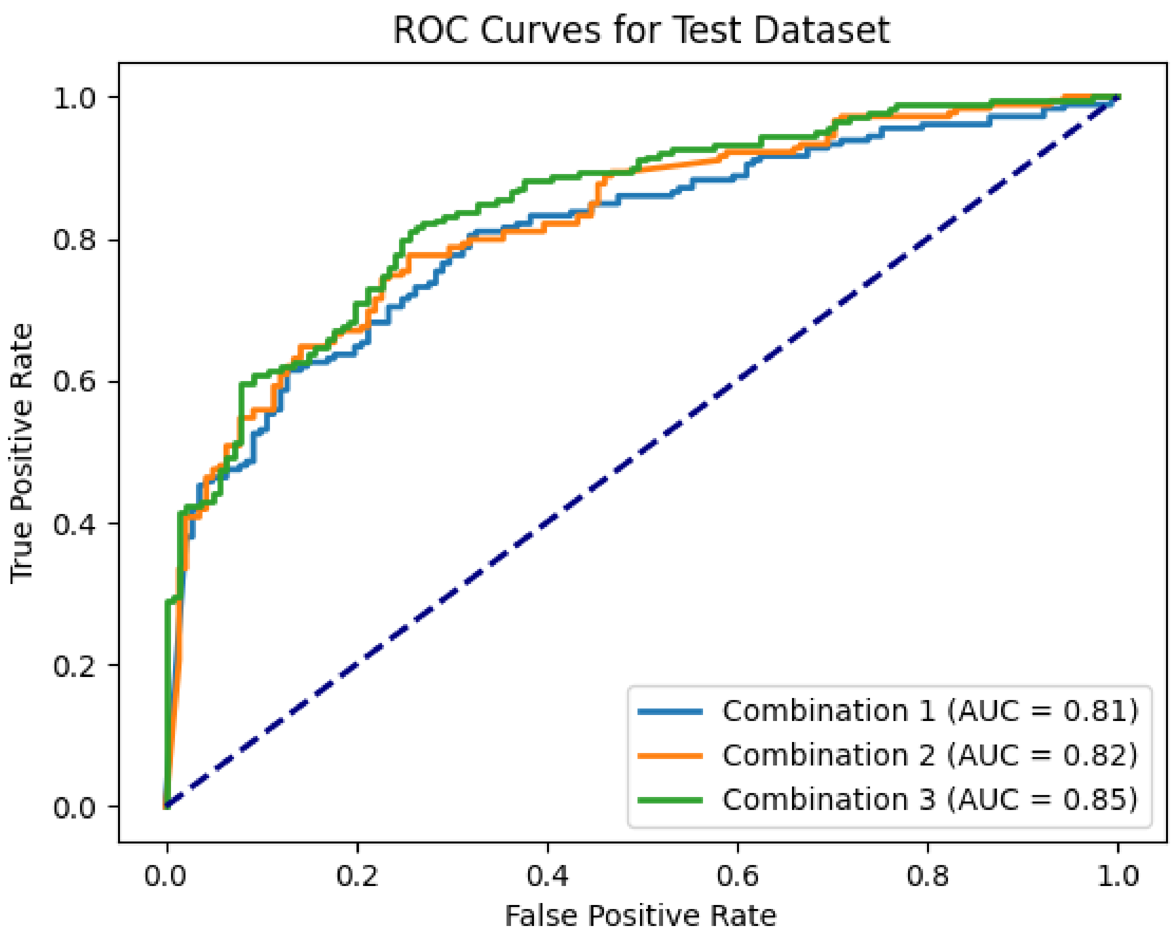

Figure 2: ROC curve for the test dataset