Red Wine Quality Classification

Ulysses Atkeson, Dorothy Mims, Emily Sheehan

ESE 417 Final Project

INTRODUCTION

Machine Learning (ML) is a system or process of using data and algorithms to learn something–such as classification or predication–about a set of data through a gradual process of improvement in its decision making [1]. ML is an application or subcategory of Artificial Intelligence (AI); while both aim to imitate the process by which humans think and learn, drawing conclusions based on data, only AI uses the conclusions to complete an action or actions, while ML simply draws those conclusions [1]. The birth of the field of ML is pinpointed to be upon the mathematical modeling of neural networks by Pitts and McCulloch in 1943 [2]. Since then, ML has expanded to many applications, including email filtering, traffic prediction, image and speech recognition, and medical diagnoses [3].

In this project, we aim to apply different ML algorithms to analyze the red wine ‘Wine Quality’ dataset. This dataset, which is widely available and was obtained from the UCI Machine Learning Repository [4], describes a set of red wine samples of Portuguese “Vinho Verde” wine. The quality as well as various physicochemical testing results are provided for each sample. The red wine dataset contains a total of 1599 instances with 12 attributes each: 11 input and 1 output attributes. The 11 input attributes were collected via objective physicochemical tests and include the following: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol. The output attribute, in contrast, was collected via sensory testing: The median was calculated from evaluations from at least three wine experts rating each wine sample on a scale from 0 to 10 (bad to excellent). A distribution of these outcomes can be seen in Figure 1.1, from which it is clear most samples fall within the middle rankings.

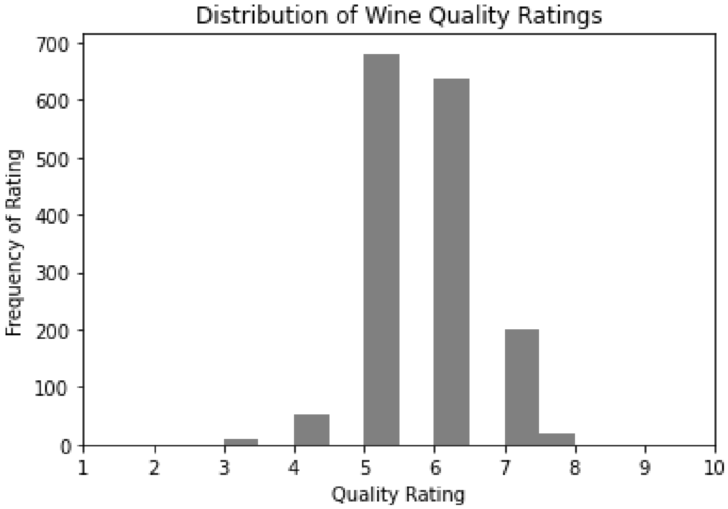


Figure 1.1: Histogram of Wine Quality Dataset Distribution

Our goal for the project is to classify the quality of the wine samples (output attribute) as accurately as possible based on the input attributes. In the process of classifying the samples, we also aim to determine the classification method which will give the best performance. The two methods examined are Random Forest (RF) and Artificial Neural Networks (ANN): The RF model combines multiple decision trees to make predictions [5]. Each decision tree sees a slightly different set of training data and uses that data to search for the best feature from a random sampling of features, then the prediction results from the trees are averaged to provide the final prediction. ANN, on the other hand, combines multiple Perceptron models (hence its alternate name, Multi-Layer Perceptron, or MLP) to make predictions [6]. Each layer uses mathematical processing to learn about the data and these layers are weighted according to their level of influence. Furthermore, the number of hidden layers (layers between the input and output layer) can be optimized for model accuracy. This system of networks is modeled to resemble the biological neural networks present in animal brains.

The performance metric used to compare accuracy of the various models is the F_1 score, which is a combination of two other performance metrics: precision and recall [7]. Precision is a measure of the rate of correct positive classifications (how many positive classifications are positive instances). Recall, or sensitivity, is a measure of the rate of catching positive instances (how many of the positive instances are classified as positive). These two methods, which are defined in Equations 1 and 2, are combined by their harmonic mean to determine the F_1-score, as defined in Equation 3.

$$\text{Precision} = \frac{\text{\# of True Positive Classifications}}{\text{\# of Positive Classifications (True and False)}} \tag{1}$$

$$\text{Recall} = \frac{\text{\# of True Positive Classifications}}{\text{\# of Positive Instances}} \tag{2}$$

$$\text{F}_1\text{score} = \frac{2}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{3}$$

Prior to applying a classification method, the dataset was examined for potential cleaning needed. In addition, Principal Component Analysis (PCA) was used to visualize the dataset. Using RF and ANN, we were able to achieve maximum classification accuracy of 73% and 62%, respectively. To obtain these classification scores, an in-depth search method, GridSearchCV, was used with each method to determine the most important parameters and thereby refine the models to improve accuracy; GridSearch loops through and tests combinations of hyperparameters predefined by the user to choose the best combination.

METHODS

Data cleaning, dimension reduction, classification, and hyper tuning were all performed to achieve high rates of accuracy in our models. First, we obtained the head of the data to examine the contents of the dataset. We dropped any unnecessary columns and checked for null values. Moreover, we used PCA to project the high dimensional wine quality data set onto a 2D feature space.

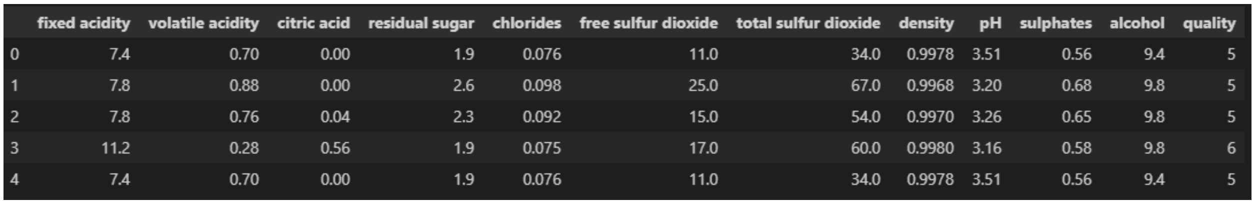
Next, to begin our analysis using the Random Forest model, we ran a baseline accuracy score and classification report using the RandomForestClassifier() class from Sklearn and the otherwise unprocessed data from the dataset. To improve, we began to clean and hyper tune our data to achieve a more accurate model: We determined and dropped the least influential features of the dataset. Our model was then trained using this refined dataset and GridSearch. Using this method, we determined the most important parameters and then trained our model.

For our ANN model, we used the MLPClassifier() class from Sklearn with a logistic activation function and an stochastic gradient descent (sgd) solver to obtain a classification report using the unprocessed data. We then tuned the number of nodes in the hidden layer as well as the number of hidden layers to determine the metrics which will give the best accuracy. Following this, we normalized the data within the dataset using StandardScaler() and again ran our model with the optimal number of hidden layers which we determined above. Finally, in an attempt to improve performance of our model even further, we used GridSearch again, this time to determine the best combination of layer sizes, activation functions, learning rates, and more.

RESULTS AND ANALYSIS

A. Preprocessing and Visualization

The head of the dataset, which was used to help us understand the types and ranges of values within each attribute (column) can be seen in Figure 3.1. We dropped the ‘quality’ column in our dataset because there is very little differentiation between wine types (there were only two values: 5.0 or 6.0). In addition, we found that there are no null values in the dataset.



	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

Figure 3.1: The First 5 Rows of The Dataset

Figure 3.2 shows the results of our PCA. From this, we determined that this classification problem is difficult and not linearly separable.

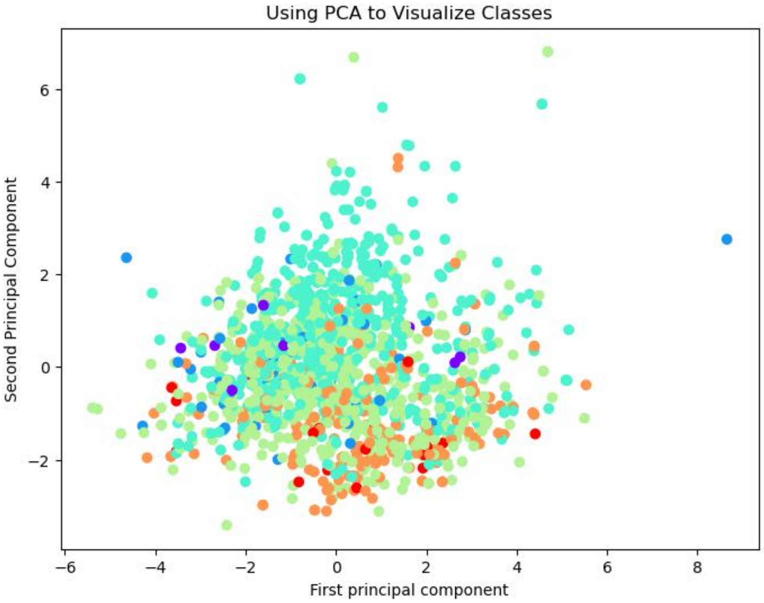


Figure 3.2: 2-Dimensional Projection of Wine Quality Data Using PCA

B. Random Forest Model

For the Random Forest Method, we tested and trained two different models: an unrefined, more primitive model, and a hyper tuned model. Under the primitive model, we did not perform any parameter optimization, and obtained the following classification report (Figure 3.3) with an accuracy score of 0.6875 (69%):

```
==========RANDOM FOREST CLASSIFIER (NO OPTIMIZATION)===========
                   Accuracy:  0.6875
                 Classification Report:
                precision    recall  f1-score   support
             3       0.00      0.00      0.00         1
             4       0.00      0.00      0.00         9
             5       0.75      0.77      0.76       203
             6       0.63      0.72      0.67       197
             7       0.68      0.53      0.60        60
             8       0.00      0.00      0.00        10
```

```
    accuracy                           0.69       480
   macro avg       0.34      0.34      0.34       480
weighted avg       0.66      0.69      0.67       480
```

Figure 3.3: Classification Report for Unoptimized Random Forest Model

We also hyper tuned a model by determining the most optimal parameters. Figure 3.4 shows the results of this parameter analysis; we dropped the least influential features of the dataset: chlorides, pH, and density.

```
                 Feature        Score
6   total sulfur dioxide  2755.557984
5    free sulfur dioxide   161.936036
10               alcohol    46.429892
1       volatile acidity    15.580289
2            citric acid    13.025665
0          fixed acidity    11.260652
9              sulphates     4.558488
3         residual sugar     4.123295
4              chlorides     0.752426
8                     pH     0.154655
7                density     0.000230
```

Figure 3.4: Comparing Feature Importance Across the Dataset

Using Gridsearch, we determined the most important parameters (Figure 3.5), trained our model under the remaining values, and obtained an accuracy score of 0.734375 (73%) when we ran the model (Figure 3.6).

```
          Best Random Forest Parameters -->
RandomForestClassifier(min_samples_split=5, n_estimators=75)
```

Figure 3.5: Best Parameters to Optimize RFClassifier

```
=========RANDOM FOREST CLASSIFIER (PARAMETER OPTIMIZATION)============
                        Accuracy:  0.734375
                      Classification Report:
                 precision    recall  f1-score   support
           3       0.00      0.00      0.00         2
           4       0.00      0.00      0.00        12
           5       0.80      0.86      0.83       136
           6       0.69      0.78      0.73       129
           7       0.63      0.42      0.51        40
           8       0.00      0.00      0.00         1
    accuracy                           0.73       320
   macro avg       0.35      0.34      0.35       320
weighted avg       0.70      0.73      0.71       320
```

Figure 3.6: Classification Report for Optimized Random Forest Model

Comparing the two models, it is clear the hyper tuned model performed better. However, the hyper tuned model was only .05 (5%) more accurate, which is a relatively small difference. Considering the levels to which we attempted to optimize the data in the steps in between the two models, one would predict a greater improvement in accuracy. This could perhaps be attributed to model confusion, as illustrated by the confusion matrix below (Figure 3.7). This diagram also implies that the 5th and 6th choice y-value labels are frequently confused and intermixed whereas the values are more distinct and properly predicted for 3rd and 4th choice values, for example.

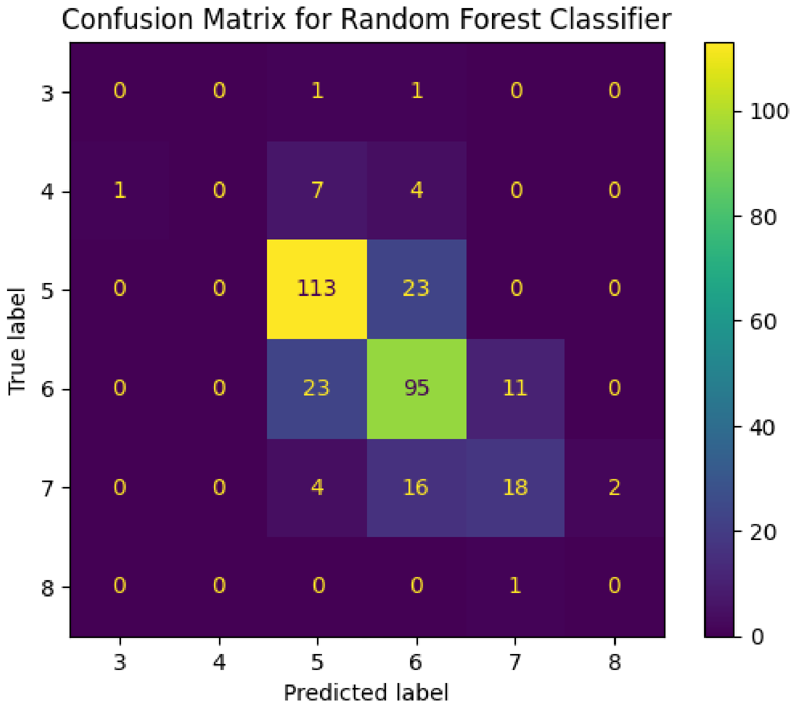


Figure 3.7: Confusion Matrix for Random Forest Classifier

C. Artificial Neural Networks Model

Again, the primary objective of this model was to maximize the model score (accuracy) on the testing data. We began by tuning the number of nodes in the hidden layer and plotting the testing and training accuracy as seen in Figure 3.8.

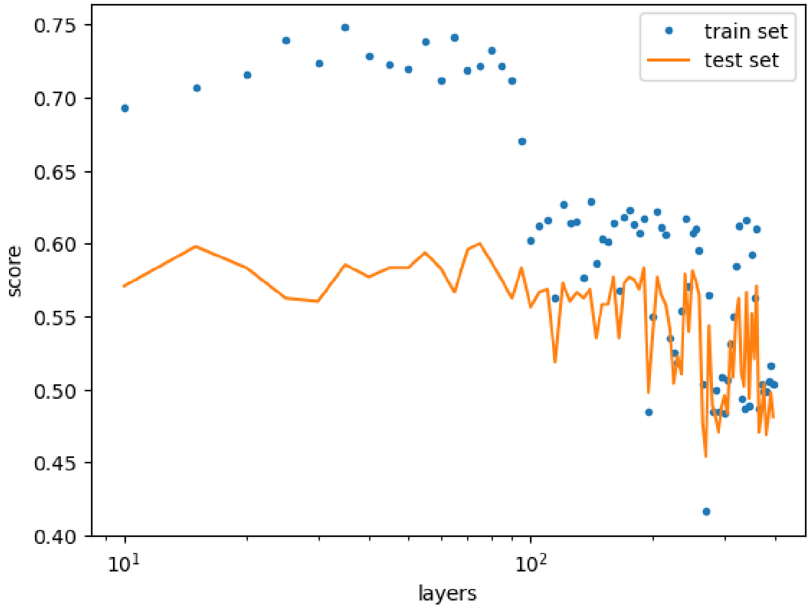


Figure 3.8: Accuracy of Testing and Training Data According to Number of Nodes in Hidden Layer

The maximum accuracy was seen when using 45 hidden nodes (per the most recent run). In addition, after using one hidden layer, adding more hidden layers did not improve our test accuracy. Therefore, we chose to keep only one hidden layer, as more layers and nodes makes the model more prone to overfitting.

We then proceeded to test normalizing the data prior to training. Table 3.1 clearly demonstrates that normalizing the data with StandardScaler() significantly improves our test accuracy.

Table 3.1: Test Accuracy for Different Pre-Processing

Pre-Processing	**F_1 Score (with optimal nodes)**
None	0.433
Standard Normalization	0.619

To ensure the best outcome, we utilized GridSearch to test 1440 different possible combinations of layer sizes, activation functions, learning rates, and more. However, no combination could beat the 0.619 (62%) test accuracy of using one layer with 45 hidden nodes along with a logistic activation and the sgd solver.

CONCLUSIONS

In this project, the red wine ‘Wine Quality’ dataset was analyzed to build a model for classification of wine quality based on physicochemical characteristics. This dataset contains 11 objective physicochemical test results and one subjective tasting result for each sample of red wine, all of which were obtained from the north of Portugal.

Upon our initial analysis of the dataset, we found no null values within. Had there been null values or values within a category not matching the type of the other instances within the category, we would have dealt with them appropriately (removing the entire instance of that sample or filling in the missing/incorrect value with values from similar samples) to ensure the data was not corrupted by those instances. The histogram of the samples (Figure 1.1) showed a very uneven distribution of classes within the dataset, with no samples rated as very bad or very excellent and nearly all samples falling directly between, with a quality of 5 or 6. Because the class imbalance is so high, simple metrics would not sufficiently represent model performance, so we chose to use F_1 score to analyze the classification accuracy of each model. In addition, analysis of PCA results (Figure 3.2) showed that the dataset is complex and nonlinearly separable, and therefore simple classification methods such as a single Perceptron or Decision Tree model would not be sufficient. We chose instead to use classification methods that build upon those simpler models: Random Forest and Artificial Neural Networks.

Using the RandomForestClassifier() and the MLPClassifier() from sklearn to implement our RF and ANN models, we achieved 73% and 62% accuracy, respectively, per the F_1 scores (Figure 3.6 and Table 3.1). From this analysis, we can conclude that RF is a better classification model for predicting red wine quality based on the provided physicochemical properties; even without optimization, RF out-performed ANN with 69% accuracy. In addition, from the analysis on feature importance (Figure 3.4), it was determined that total and free sulfur dioxide content were the most important features for quality of wine and least important were chlorides, pH, and density. (To reduce model complexity these least important features were removed when using RF).

A limitation in our analysis is that we only tested two classification methods to reach this maximal accuracy. Future steps would include implementation of the Support Vector Machine (SVM) classification method, as it is stated by the UCI Machine Learning Repository to give the best results under a regression approach; it would be interesting to see if SVM would also give the best results for a classification approach. Overall, however, we were successful in classifying the Portuguese red wine sampled based on quality from physicochemical tests, with a final 73% accuracy using a Random Forest Classifier.

Our team split up the project as follows: Emily and Ulysses each implemented a model (Emily RF and Ulysses ANN) and documented the results and analysis for their respective models. Emily also explained the methods for RF and completed the data cleaning, while Ulysses performed PCA. Dorothy organized the report and wrote the remaining content (introduction, ANN methods, and conclusions); this included the histogram used in the introduction.

REFERENCES

[1] J. Zhang, "Introduction and Review." ESE417: Machine Learning and Pattern Classification. Washington University in St. Louis. 18 Jan. 2023. Class lecture.

[2] W. Chai, "A Timeline of Machine Learning History," WhatIs.com, Oct. 20, 2020. https://www.techtarget.com/whatis/A-Timeline-of-Machine-Learning-History#:~:text=Machine%20learning%20was%20first%20conceived

[3] "Applications of Machine Learning - Javatpoint," www.javatpoint.com. https://www.javatpoint.com/applications-of-machine-learning.

[4] "UCI Machine Learning Repository: Wine Quality Data Set," archive.ics.uci.edu, 2009. https://archive.ics.uci.edu/ml/datasets/wine+quality.

[5] J. Zhang, "Decision Tree and Random Forest." ESE417: Machine Learning and Pattern Classification. Washington University in St. Louis. 15 Apr. 2023. Class lecture.

[6] J. Zhang, "Artificial Neural Networks." ESE417: Machine Learning and Pattern Classification. Washington University in St. Louis. 27 Mar. 2023. Class lecture.

[7] J. Zhang, "Perceptron and Logistic Regression." ESE417: Machine Learning and Pattern Classification. Washington University in St. Louis. 20 Feb. 2023. Class lecture.

APPENDIX