Analyzing and Experimenting with the Wine Quality Dataset

Daniel Xie
Undergraduate 3rd Year Student
dxie@unc.edu

Rohan Kumar
Undergraduate 4th Year Student
rorod237@ad.unc.edu

Himanshu Kunwar
Undergraduate 4th Year Student
hkunwar@unc.edu

Rohan Ray
Undergraduate 4th Year Student
rohanray@unc.edu

Jeremy Chen
Undergraduate 4th Year Student
ziang1@email.unc.edu

1 Introduction

Wine is an acquired taste. But that is putting it too simply. Some wines are clearly better than others in terms of quality, and the goal of this project is to analyze what parameters and factors have the largest impact on the quality of wine. To do this we first sourced our dataset from Kaggle named the Wine Quality Dataset. Using this dataset, we aim to see how different machine learning algorithms can properly assess this dataset, while also keeping the main goal of the project to predict the quality of a wine given the parameters. The quality metrics range from a value of 3 to 9, therefore to predict quality, we must use multiclass classification.

Multi-class prediction, a process where a model is trained to classify instances into one of three or more classes, passes limitations of binary classification and opens more possibilities across various domains. This methodology not only enhances the capability of machine learning systems in handling large datasets with multiple categories but also aligns more closely with real-world scenarios where classifications often extends past just two classes.

The significance of multi-class prediction lies in its versatility and applicability across diverse fields. From medical diagnostics, where diseases may fall into numerous categories, to natural language processing and image recognition. Moreover, the advent of multi-class prediction has catalyzed advancements in algorithmic approaches and computational techniques. The use of sophisticated algorithms such as neural networks, decision trees, and support vector machines in multi-class prediction scenarios has not only improved the accuracy and efficiency of these models but also contributed to the broader understanding of machine learning dynamics.

This paper aims to explore the intricacies of multi-class prediction and its various uses to see how it can be applied to the dataset of Wine Quality.

2 Data Analysis

The dataset is large given the structure and has 6497 data points each containing the information such as whether it is red or white wine. Categories include wine type, fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol and quality. Using the dataset, we can first analyze the data by doing Principal Component Analysis (PCA). Some data parsing must be done such that red wines are categorized as a 0 and white is a 1 as

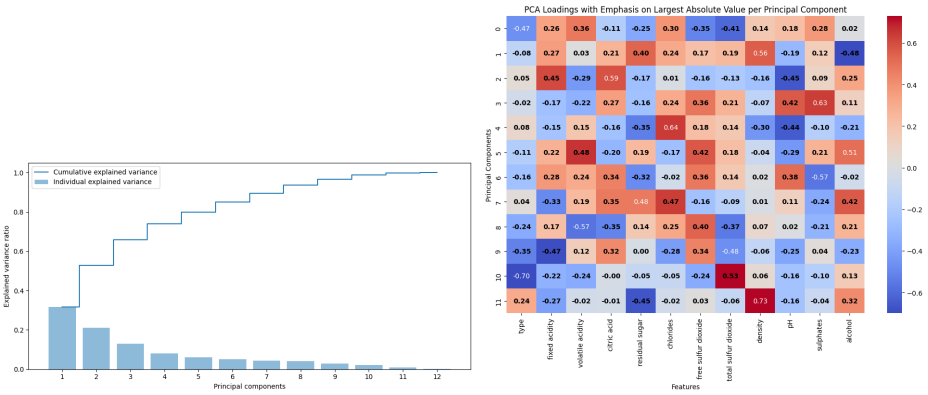

Figure 1: PCA Components and their importance;Depiction of how much each parameter effects each PCA

there are clear distinctions in their taste. Also if a data point is missing data, it was excluded, but there were not many instances of this.

As part of our data preprocessing and exploration, we employ PCA, a statistical technique pivotal in reducing the dimensionality of our dataset while preserving as much variance as possible. PCA will be instrumental in uncovering underlying patterns in the data, simplifying complexities, and enhancing the interpretability without substantial loss of information.

Our dataset, comprising numerous variables, presents a high-dimensional space that might confuse meaningful relationships due to there being too much dimensionality. PCA assists in mitigating this by transforming the original variables into a new set of orthogonal features, known as principal components. These components are linear combinations of the original variables, ordered such that the first few retain most of the variation present in the entire dataset.

Prior to applying PCA, the data will undergo standardization to ensure each variable contributes equally to the analysis, preventing features with larger scales from dominating. Post-standardization, PCA will be executed to extract the principal components. The number of components retained for subsequent analyses will be determined based on the explained variance ratio, ensuring a balance between dimensionality reduction and information retention.

PCA analysis shows the components as seen in Figure 1. However, these are orthogonal elements that can't be easily analyzed such that each component is just simply explained by 1 parameter. The following heatmap in Figure 1 on the right, shows how much each parameter effects the PCA component. As seen, there is no parameter that can be removed as it apperas all of the parameters are necessary. However, this is not a problem, since it is largely numerical data that can be easily trained in the other models even given with the 12 parameters.

2.0.1 Methods

Throughout the semester we learned about many different optimizers. The selection and application of appropriate algorithms are crucial for unraveling complex patterns within data. This section of the paper delves into the comparative analysis of many different machine learning models, each offering unique perspectives and strengths in the exploration of our dataset. Our investigation encompasses Linear Regression, Logistic Regression, Naive Bayes, K-Nearest Neighbors (KNN), Decision Trees, Support Vector Machines (SVM), and Artificial Neural Networks (ANN).

Linear Regression, offers a clear perspective on relationships between variables, ideal for examining continuous data. Logistic Regression, in contrast, extends this analysis to categorical outcomes, providing a probabilistic understanding of classification problems. These regression models, with their simplicity and interpretability, serve as a baseline for our analysis as they are the most simple.

Naive Bayes offers a unique approach grounded in Bayes' theorem, excelling in scenarios where assumptions of independence hold true where it is believed that parameters are largely independant

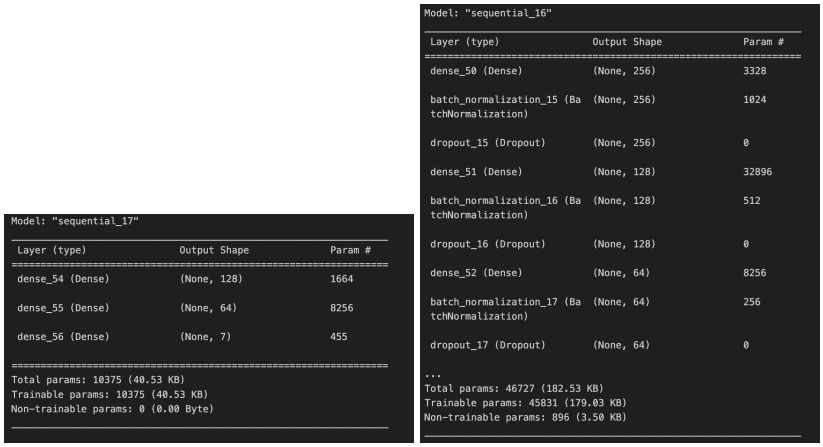

Figure 2: Depiction of how gradients are updated between the local and global models

of each other. This model often finds its strength in text classification and scenarios with high-dimensional data.

The K-Nearest Neighbors (KNN) algorithm relies on the concept of similarity or 'closeness' to make predictions. This intuitive method, while being simple, offers valuable insights, especially in classification problems. Decision Trees provide a more graphical approach, where data is split according to certain criteria enabling us to visualize the decision-making process.

The Support Vector Machine (SVM) focuses on the creation of optimal boundaries between different classes. Known for its effectiveness in high-dimensional spaces, SVM is a robust classifier that performs well in a variety of settings.

Lastly, we explore Artificial Neural Networks (ANN) to model complex, non-linear relationships through layers of interconnected 'neurons' and hidden layers. ANNs are instrumental in tackling tasks that were previously deemed intractable, and potentially becoming the most powerful tool we use in this paper. This section aims to provide a comparative analysis of these models on our dataset.

Linear Regression, Logistic Regression, Naive-Bayes, SVM and Decisions Trees can be created using the sklearn library. However, ANN and KNN requires an adjustment hyperparameters in order to find a most optimal answer. For KNN, we experimented with various values for k, and opted to use k = 15. For the ANNs, we used the following two different Neural Networks in Figure 2. They have largely different architectures, and we opted to analyze both the simple ANN and the complex ANN.

3 Results

3.0.1 Linear Regression, Logistic Regression and Naive Bayes

Linear Regression achieves $MeanSquaredError$: 0.5406592848055256 and R^2Score : 0.29022771380876033. This is clearly very poor as you want to aim for an R^2 value of 1 showing that this linear regression does not accurately depict quality as seen in the leftmost graph of Figure 3. Logistic Regression works with slight improvement is visibile in the center confusion matrix which depicts the results with $Accuracy$: 0.46403712296983757. Naive-Bayes does not work that well and has $Accuracy$: 0.402938901778809 with its confusion matrix on the right of Figure 3. This shows that the parameters are largely related to each other and not independent.

3.1 K-Nearest Neighbors

KNN achieves a higher accuracy of $Accuracy$: 0.588553750966744. To do this we need to experiment with the hyperparameter of how large k should be. Eventually we find an accurate result with $k = 15$. We can see a more accurate Confusion Matrix in Figure 4 compared to the previous results.

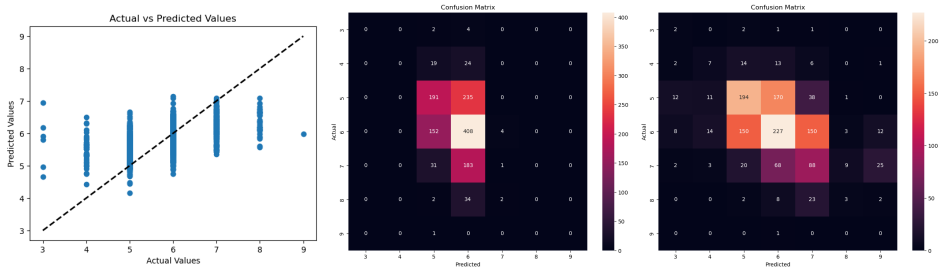

Figure 3: Linear Regression Approximation; Logistic Regression Confusion Matrix; Naive Bayes Confusion Matrix

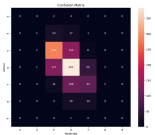

Figure 4: KNN Confusion Matrix

3.2 Decision Tree, and Support Vector Model

SVM works fairly well with $Accuracy$: 0.5815931941221965. The model shows relatively stronger performance in classifying wines of quality ratings 5 and 6, with f1-scores of 0.64 and 0.63, respectively. This suggests that the model is more adept at recognizing patterns or features common in these classes, possibly due to a higher representation in the dataset.

In contrast, the model significantly underperforms in identifying wines of quality ratings 3, 4, 8, and 9, with negligible or zero scores across precision, recall, and f1-score. This poor performance could be attributed to the sparse representation of these classes in the dataset, leading to difficulties in learning distinguishing features.

Decision tree works even better with $Accuracy$: 0.6094354215003867. The model performs reasonably well in predicting medium-quality wines (ratings 5, 6, and 7), with f1-scores of 0.67, 0.64, and 0.55 respectively. This is indicative of the model's effectiveness in discerning features that are more common in wines of these quality ratings. The performance on these classes suggests a decent understanding of the relevant characteristics that define medium-quality wines.

Challenges with Extreme Classes: The model struggles significantly with the extreme classes (particularly ratings 3, 4, 8, and 9), as evidenced by lower precision and recall values. The near-zero scores for the rarest classes (3 and 9) highlight a challenge in identifying unique features of these wines, likely due to their underrepresentation in the dataset.

Unlike the SVC model, the Decision Tree shows some capability in distinguishing wines of quality rating 8, even with a moderate f1-score of 0.38. This difference between the models may stem from the Decision Tree's inherent ability to handle non-linear relationships and complex decision boundaries.

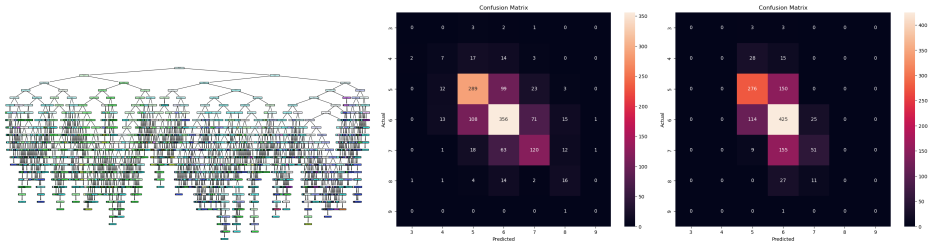

Figure 5: Decision Tree; Decision Tree Confusion Matrix; SVM Confusion Matrix

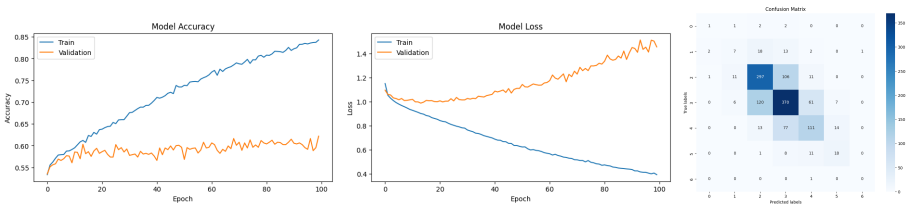

Figure 6: Simple ANN Accuracy, Loss, and Confusion Matrix

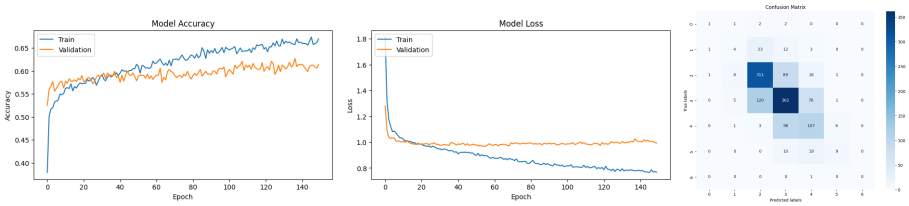

Figure 7: Complex ANN Accuracy, Loss, and Confusion Matrix

3.3 Artificial Neural Networks

The constructed ANNs achieved very different results. The simple model never seemed to be able to decrease the loss and the accuracy validation plateaued at around 55% accuracy which is not optimal which can be seen in Figure 6. It is clear that in this scenario, the model quickly overtrained on the data supplied and did not learn general characteristics. However, the more complex model seemed to become more effective as seen in Figure 7. The validation and training loss and accuracy seemed to generally align although the model slightly over trained, the loss did not increase like the simple model did.

4 Future Elaborations/Conclusion

The models never seemed to be able to train above 65% accuracy. The models consistenly aimed around that area, however, this might be due to the fact the dataset lacked equal representation for every category. The models consistently found patterns for wines of quality 5, 6 and 7, however, they struggled to find accurate depictions for wines of quality 3, 4, 8 and 9. This is due to how few instances of wines of those quality exist. A potential solution is to scale down the wines so that there is equal representation, but there are so few wines with quality 9 and 3 so that the overall dataset of equal representation would be very small. We believe that a dataset with equal representation and large amounts of numerical data will allow the methods we used in this paper to achieve high accuracy.

References

[1] Haddi, Y. (n.d.). Wine Quality Dataset. Kaggle. Retrieved December 10 2023, from https://www.kaggle.com/datasets/yasserh/wine-quality-dataset