Predicting Wine Quality Given Quantitative Characteristics

Josh Connors
Jeffery Bindeman
Josh Kennedy

I. INTRODUCTION

Predicting the success of wines is never an easy task. Traditionally, wine has been rated based on factors like the quality of the grapes, the quality of the land it was produced, and even the expertise of the vintner who made it. However, these are all qualitative attributes that can be next to impossible to replicate, as evidenced by the lack of consistently same wine across years. We want to see if converting those traits into quantitative qualities can give a better insight into the chemical properties of the wine, and if using those properties can correctly determine quality.

The data came from a paper published by Paulo Cortez at the University of Minho in Guimarães, Portugal. [1] It consists of 4,898 white and 1,599 red Portuguese "Vinho Verde" wine. Due to logistic and privacy reasons, there was no data on things like grape type, wine brand, and selling price.

II. CHALLENGES

We encountered a few different problems throughout this project. The dataset consisted of both red wine and white wine samples separated into individual files. This led us to make one of two choices: keep the data separated and make two different models, one for red and one for white wine, or merge both the red and white samples and make one model for the whole dataset. We opted for the former of the two choices, as white and red wines likely have different chemical properties.

III. APPROACH

Three models were ultimately used: a Decision Tree, a Nearest Neighbor classifier, and a Neural Network. This is because we wanted to also see if a specific model would fare better than another. We chose the first two because they would be easy to implement, and the latter because we expected it to have a much higher success rate.

IV. TECHNICAL DETAILS

For all the models, the data was preprocessed in the exact same way. First it was loaded in from 2 csv files with the python numpy library. Then, 100 random samples were pulled out of both the list of white wines and red wines to create the test samples, as we didn’t have our own wines to test. All the wines originally had their qualities labeled in terms of integers between 3 and 9, so to raise the accuracy or our models the labels were grouped into ranges that represented “Bad” (3-4), “Average” (5-7), and “Good” (8-9) wines. This created a slight problem that we will discuss later, where there were significantly more Average wines than there were even Bad and Good combined.

In terms of the Neural Network, we desired a highly non-linear decision boundary due to the skewed nature of the data. We decided on an 11-1000-750-7 network all with sigmoid activation functions to achieve a non-linear decision boundary. For the k-Nearest Neighbors model we used the KNeighborsClassifier from scikit-learn, applying k = 2 as it achieved the highest accuracy without overfitting the data. For the decision tree, we again utilized scikit-learn for the DecisionTreeClassifier module, varying both the maximum depth of the tree and minimum number of samples to split a node on. The white wine model had a maximum depth of 10 and a minimum sample split of 2, while the red wine model had a maximum depth of 7 and minimum sample split of 9.

V. RESULTS

Out of all of the algorithms, the Neural Network performed the worst in terms of accuracy: it achieved 37.4% accuracy on the white wine dataset and only 20% on the red wine dataset. Because of this, we decided to use different metrics to evaluate the performance of the neural network such as the average difference between the correct label and the predicted label rounded up using the numpy.ceil function, and the accuracy for the top 3 guesses. In terms of this metric, the white wine dataset achieved a 1.0 difference lately, and the red wine dataset achieved a 2.0 average difference. The top-3 accuracy was 52.9% and 74.7% for red wine and white wine respectively.

K-Nearest Neighbors and the Decision Tree fared much better. To address the K-Nearest Neighbors models first: the white wine model was able to achieve an 85% success rate, while the red wine model achieved an astounding 95% success rate. As for the Decision Tree, the white wine model achieved a success rate of 92%, while the red wine model also achieved the 95% success rate.

VI. DISCUSSION

One of the largest problems with the data was due to how skewed the data was toward Average wine. Of the 4,898 white wines, 4,535 of them were Average, while only 363 were either Good or Bad. Put another way, only ~7.4% of the wines were not Average. The red wine data had a similar issue, with 1,518 of the 1,599 being average, meaning only ~5.1% of the wine was not Average.

VII. CONCLUSIONS

This dataset is a great window into the real world. It is very possible for datasets to contain problems that can be addressed using techniques learned in college classrooms or through the internet. The purpose of this project was to determine which features had the most profound effect on the quality of a given wine. In this project, we determined that the decision tree was the most helpful in answering this question by visualizing the relationship between features of the wine and their quality.

We also learned how difficult it is to optimize a neural network. While the k-Nearest Neighbor models only had to vary between the number of k used, and the Decision Tree only had to change its maximum depth and number to continue splitting nodes on, the neural network had a variable learning rate, variable number of hidden layers, variable size for each of those layers, and variable activation function. This meant that much more care had to be chosen in what to change in order to find something optimal, and ultimately we feel as though our model is still lacking. This is evident in how much worse it fared compared to the other two classifiers, which by all accounts should have ultimately been worse.

VIII. SOURCES

[1] Cortez, Paulo, et al. “Modeling Wine Preferences by Data Mining from Physicochemical Properties.” *Decision Support Systems*, vol. 47, no. 4, Nov. 2009, pp. 547–553., doi:https://doi.org/10.1016/j.dss.2009.05.016.