Data Exploration of Wine Quality

Zackaria Jandali
Cattien Ngo
Hamood Rana
Department of Information and Computer Science
University of California Irvine

Abstract

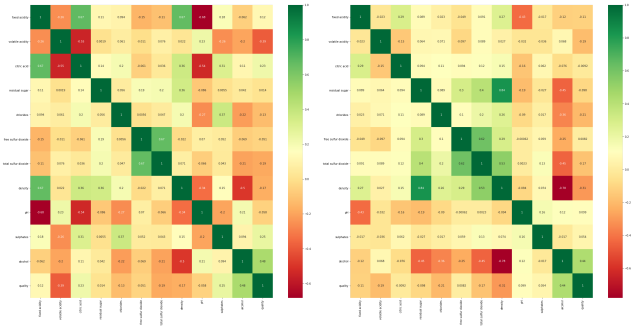
Within this report we detail the exploration of machine learning to classify wine quality based on a given set of data. First, we explore dimensionality reduction by searching for highly correlated features. Next, we use neural networking to train a model that can classify the wine as red or white. Lastly, we utilize random forests to judge the overall wine quality based on the given data. The white wine dataset holds 4898 instances of wine data, while the red wine dataset holds only 1599.

1 Finding Ways to Reduce Redundant Data

When first given the dataset, it is important to find places where features can be removed to reduce noise when training a learner. This also helps reduce model complexity to lower potential overfitting (due to learning features which aren't important and to increase computational efficiency.

1.1 Heatmaps of Red and White Wine

Red **White**



Figures 1 (left) and 2 (right): Heatmaps showing correlation between features in red (left) and white (right) datasets. Green corresponds to high/red to low correlation.

We used heatmaps to try to find correlations between features. We learned how to create these heatmaps from an article called “Feature Selection Techniques in Machine Learning with Python” by Raheel Shaikh (2018). When we plotted the heatmaps for the red wine and white wine datasets, as seen in Figure 1 and 2 respectively, we found that most of the features for both datasets were not strongly correlated. Because most of the features were not strongly correlated, we decided to use all of the features to get the most information.

2 Using Neural Networks to Predict Wine Quality

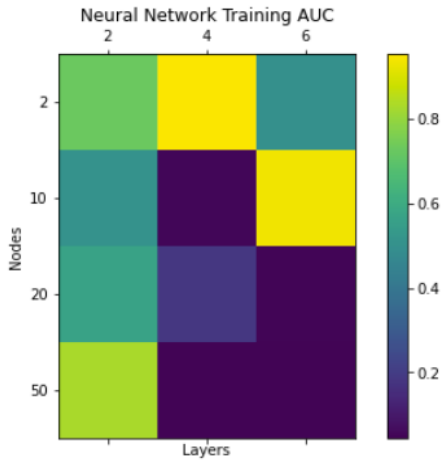
2.1 Why Neural Networks?

Neural networks can learn complex relationships like those found in the real world. Since we were working with data consisting of many features with little correlation with each other, we thought that neural networks were a good option to model the complex relationships between the features in the dataset. The goal was to create a neural network that could accurately predict which wine type it was: red or white. Once we get this information, we move to classifying the wine quality of the data into a decision tree discussed in section 3.

2.2 How We Selected Parameters

Since the goal of this neural network was to classify between red and white, we merged both datasets together and split 80/20. We then assigned each red wine data point a target value of 0 and 1 for white to replace the quality Y value scale of 1-10. To select the parameters that we want to use in our learner, we created two matrices, one for training data and one for validation data, where the columns represented different amounts of layers and the rows represented different amounts of nodes. Each cell in the matrices recorded the area under the curve (AUC) for the learner trained on its respective layer value and node value.

We created a color mapping for the AUC matrix. The x-axis represents the number of layers used in the learner and the y-axis represents the number of nodes used in the learner. Darker colors correspond to a lower AUC and lighter to high. In this case, the ideal learner would have a lighter square highlighted for its (node, layer) value as this represents the probability to successfully classify the data. (Figure 3 & 4 below.)



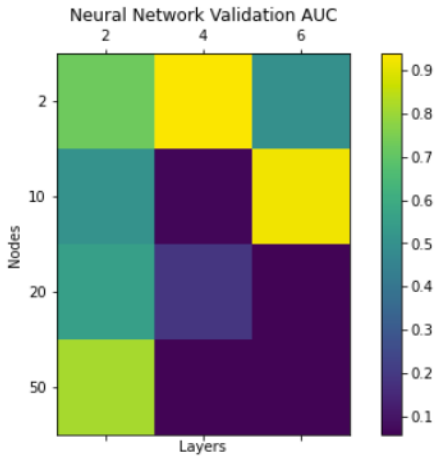


Figure 3 & 4 (left & right): AUC color map for classification between red and white wine on training and validation data respectively.

Figure 3 represents the Training AUC heatmap for the data for both the red and white wine.

Figure 4 represents the Validation AUC heatmap for the data for both the red and white wine. Notice that this graph is very similar to the previous training graph. After careful consideration, we can say that the best (node, layer) pair to select from would be 2 nodes and 4 layers due to its high AUC content.

2.3 Performance of Our Neural Network

After running the neural network a couple of times, we found that the process was very computationally expensive but yielded important results. The main thing to note would be that high node and layer values would often overfit the data. Since there were only 11 features with most of the data points being white wine, the extra layers and nodes were often unnecessary. This caused some of the higher node and layer neural networks to memorize features that "weren't really there" and to overfit on the validation data.

3 Using Random Forests to Predict Wine Quality

3.1 Why Random Forests?

Decision trees inherently choose the best features to split on and since our datasets had many features, we thought that decision trees could be helpful in choosing which features to split on first. Like neural networks, they also have the capability to learn complex functions and, as mentioned before, the wine datasets seemed fairly complex and we decided that decision trees would be a good choice to model these complex datasets.

Random forests have the added benefit of returning the average result of many decision trees. Because they return the average of their decision trees, random forests maintain the low bias of decision trees while also reducing their variance.

3.2 How we Selected Parameters

The parameter we adjusted was the number of trees to use in our random forest. We created a list of different tree values (10, 100, 1000, 2000) and created random forests using those tree values. We also ensured that all random forest learners used the same random seed so that we know that the variations in our results are not due to the randomness inherent in creating random trees.

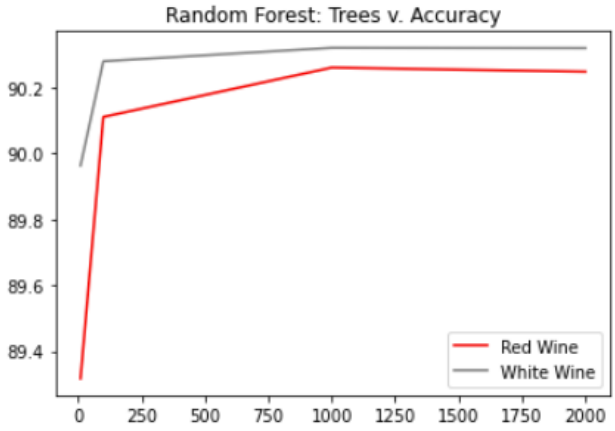


Figure 5: Graph showing the relationship between the number of trees in a random forest and the model's prediction accuracy.

The random forests created for both the red and white wine datasets showed similar results, as seen in figure 5. Below 100 trees, the predictions were fairly accurate, but there is a significant increase in accuracy at 100 trees. A slight increase is seen at 1000 trees. For red wine, the accuracy of predictions started to decrease slightly after 1000 trees. For white wine, the accuracy of predictions remained the same after 1000 trees.

These results led us to choose 1000 trees for both the red wine and white wine random forests. For red wine, accuracy seemed to fall after 1000 trees. For white wine, the accuracy did not improve despite the added computational cost of creating a random forest with more trees.

3.3 Performance of Our Random Forest

Setting aside 75% of the red wine data into a training set and 25% of the data into a validation set, we were able to create a random forest using the training set while checking the random forest's performance using the validation set. Using the parameters that were discussed earlier, we found that the red wine random forest predicted red wine quality with 90.26% accuracy.

We also split the white wine data into training and validation sets in the same way that we did with the red wine data. We created another random forest using the white wine's training set and checked the random forest's performance using the white wine's validation set. We also used the same parameters that were discussed earlier. We found that the white wine random forest predicted white wine quality with 90.32% accuracy.

Both random forests performed this quality prediction task at a level that we thought was acceptable.

Acknowledgements

We want to thank the professor, Stephan Mandt, as well as the Teacher Assistants of this course for their hard work and willingness to accommodate students during the COVID-19 pandemic.

References

[1] Shaikh, R. (2018) “Feature Selection Techniques in Machine Learning with Python”. Towards Data Science.

[2] Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009) Modeling wine preferences by data mining from physicochemical properties. In *Decision Support Systems*. Elsevier.