Classification of the Quality of Wines

Daniel Mason

Abstract—This analysis delves into an investigation associated with red and white variants of Vinho Verde wine from northern Portugal. The primary objective is to model wine quality based on its respective physico-chemical attributes. This was based on the dataset provided by Paulo Cortez, et al [2] which contained the relevant physico-chemical profiles of over 6000 wines . An initial EDA revealed significant differences between the red and white compositions leading to a primary focus on the larger white wine dataset. Both Random Forest and Gradient boosting models were applied to the dataset both having final weighted average F1 scores of 0.79. An encouraging result given the difficulties encountered during the task.

Index Terms—Classification, Wine, Quality, Multi-class, Random Forest, Gradient boosting

1 Introduction

This project involves utilising machine learning techniques to predict the quality of wines based on their physico-chemical attributes. This undertaking holds significant relevance within the realm of viticulture and wine production, offering valuable insight to both manufacturers and consumers alike.

In the wine industry assessment of its products quality has immense influence over the consumer preferences, market positioning, and ultimately, commercial success. Conventional, wine quality evaluation relies on sensory analysis carried out by a Sommelier. This method of assessment is subjective, time-consuming, and often limited in its scale.

This project addresses how one could address the challenge of enhancing and automating the wine quality assessment process. By utilising machine learning algorithms to analyse the key physico-chemical properties (seen in **Table 1)** of high and low quality wines. The overall aim of the project being to predict and categorise wines into distinct quality levels, providing a systematic and efficient alternative to manual evaluation on a large scale.

Feature name	Data type
fixed acidity	Continuous
volatile acidity	Continuous
citric acid	Continuous
residual sugar	Continuous
chlorides	Continuous
free sulfur dioxide	Continuous
total sulfur dioxide	Continuous
density	Continuous
pH	Continuous
sulphates	Continuous
alcohol	Continuous
Colour	Categorical

TABLE 1: Description of features and their data types.

Under Tom Mitchell's definition of a machine learning task, "a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E". [3] My problem can be defined within this framework as follows:

Experience (E): A labelled data-set of physico-chemical properties and wine quality ratings.
Task (T): Predicting and categorising the quality rating of wines based on these properties
Performance Measure (P): The F1 scores of the model

This project aims to enhance the classification of wine quality based on its physical properties. By developing an accurate predictive model, which will enable a more objective assessment of wine quality. Ultimately, this could aid manufacturers in assessing and potentially improving wine quality based on its properties and characteristics. There is further the opportunity to use the model in conjunction with monetary data to analyse the most cost effective high quality wine to produce. This however would require access to more sensitive and secure data which would need to be sourced directly from a manufacturer.

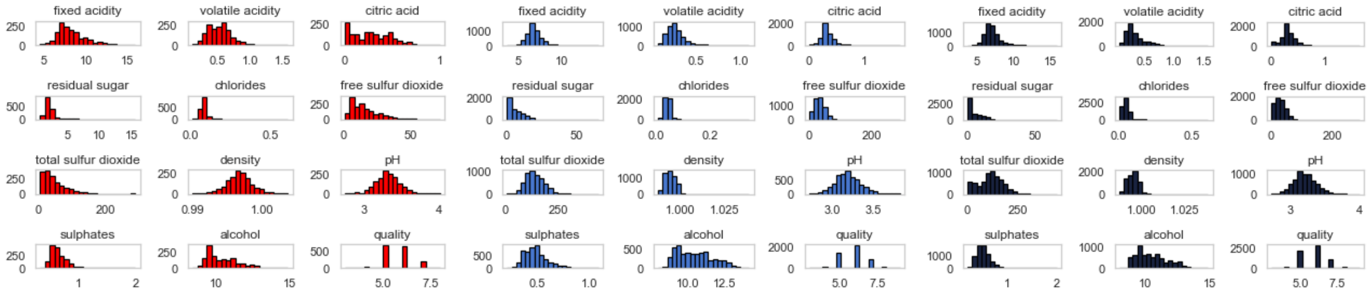
2 Exploratory Data Analysis and Data Preparation

2.1 DATA ANALYSIS AND VISUALISATION

The data set initially comprised of both red and white Vinho Verde wine varieties from northern Portugal. Both red and white wine data were loaded separately as well as the initial combined data set, each of which was subjected to exploratory data analysis (EDA) where histograms were visualised to understand the feature distributions (**Figure 1**) within each wine type.

Upon examination of (Figure 1), it was evident that certain features varied significantly between red and white wines. Statistical measures like the mean, and standard deviations of the features in both red and white wines respectively further highlighted these disparities (**Table 2**).

It was also necessary to check the count distribution of each data set seen in **Figure 2** to analyse the balance of the classes in the data.



(a) Red Wine 1

(b) White Wine 2

(c) Combined Wine 3

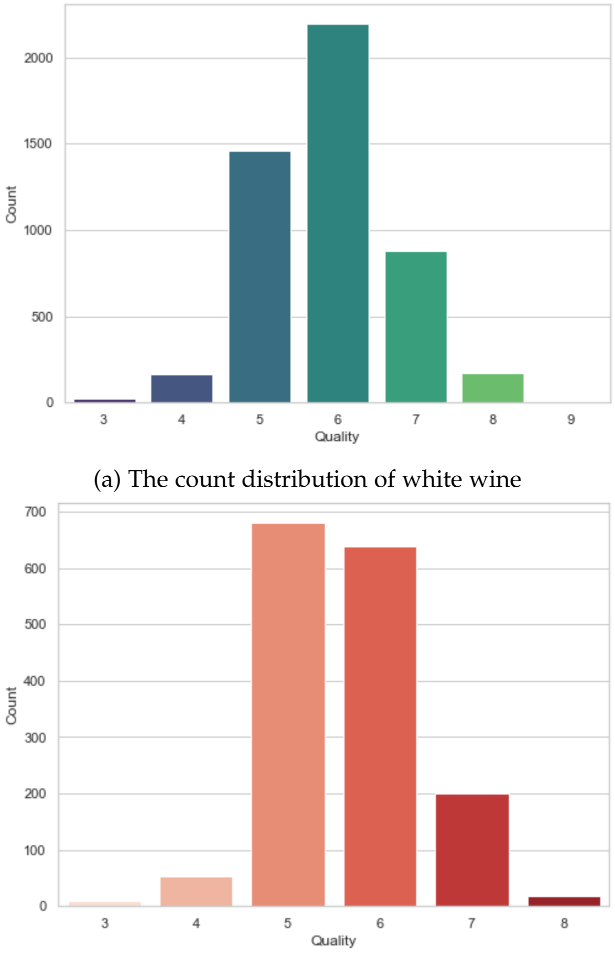
Fig. 1: The Distribution of Features in each Wine Type

White Wine stats	Count	Mean	Std	Min	Max
Fixed Acidity	4898	6.8548	0.8439	3.8	14.2
Volatile Acidity	4898	0.2782	0.1008	0.08	1.1
Citric Acid	4898	0.3342	0.1210	0	1.66
Residual Sugar	4898	6.3914	5.0721	0.6	65.8
Chlorides	4898	0.0458	0.0218	0.009	0.346
Free Sulfur Dioxide	4898	35.3081	17.0071	2	289
Total Sulfur Dioxide	4898	138.3607	42.4981	9	440
Density	4898	0.9940	0.0030	0.9871	1.0389
pH	4898	3.1883	0.1510	2.72	3.82
Sulphates	4898	0.4898	0.1141	0.22	1.08
Alcohol	4898	10.5143	1.2306	8	14.2
Quality	4898	5.8779	0.8856	3	9

Red Wine stats	Count	Mean	Std	Min	Max
Fixed Acidity	1599	8.3196	1.7411	4.6	15.9
Volatile Acidity	1599	0.5278	0.1791	0.12	1.58
Citric Acid	1599	0.2710	0.1948	0	1
Residual Sugar	1599	2.5388	1.4099	0.9	15.5
Chlorides	1599	0.0875	0.0471	0.012	0.611
Free Sulfur Dioxide	1599	15.8749	10.4602	1	72
Total Sulfur Dioxide	1599	46.4678	32.8953	6	289
Density	1599	0.9967	0.0019	0.9901	1.0037
pH	1599	3.3111	0.1544	2.74	4.01
Sulphates	1599	0.6581	0.1695	0.33	2
Alcohol	1599	10.4230	1.0657	8.4	14.9
Quality	1599	5.6360	0.8076	3	8

TABLE 2: Useful Statistics of Both Red and White Wine Attributes

This analysis revealed distinct differences between red and white wines, leading to the decision to evaluate them separately due to their dissimilar characteristics which would confuse a model when trying to determine the quality.



(a) The count distribution of white wine

(b) The count distribution of red wine

Fig. 2: Comparison of wine count distributions

2.2 DATA TRANSFORMATION

After establishing that the two wine types couldn't be evaluated in tandem, the decision to focus primarily on white wine was made. Considering its larger data set deemed it more suitable for effective model training.

Looking at the count distribution within the white wine data set I realised a flaw that would befall the model , in that given it had received 0 input data for the qualities 1,2 or 10 it would never predict a wine to have that quality. To address this I regrouped the data into 4 distinct bins: 'Poor', 'Mediocre' 'Good' and 'Excellent' where each was represented by the qualities '1,2,3' , '4,5' .'6,7' and '8,9,10' respectively. This streamlined representation whilst also making the final result more accessible to a customer who may not notice the subtle differences between a quality 9 and 10 wine, but would appreciate the distinction between a 6 and a 10.

On inspection the data set didn't have even remotely balanced classes of data, if anything it seemed to follow a normal distribution. Even after regrouping them the imbalance remained prominent. Meaning the model would perform with a heavy skew in experience for wines with quality's 'Mediocre' and 'Good' if undealt with.

I began by splitting the test and training sets before trying to address the class imbalance, this ensured that no instance was used in both the training and testing data. The data was separated into a training set which consisted of 80% of the original set and a test set which subsequently was 20% this was carried out in a stratified manner so that the integrity of the data-set's representation was maintained for the examination set.

Following this I used an up-sampling technique provided by `imblearn`. The `RandomOverSampler` code works by targeting non majority classes i.e the underrepresented data, and randomly duplicating instances until the class size is equal to that of the majority class. The randomness introduced in the oversampling process helps avoid overfitting and introduces variability in the synthetic samples. Although given the original limits in the data set this variability itself is quite small.

Due to the limited number of features in the data set, I decided that all of the data should be preserved. As each attribute could hold significant information relating to the wine quality. If more physcio-chemical properties had been available however, it would have been beneficial to undergo feature selection to reduce noise and redundant computations.

3 LEARNING ALGORITHM SELECTION

To choose a viable learning algorithm I had to ensure to choose those which could accommodate the multi-class nature of the dataset. I elected to choose a Random Forest as a baseline model as the ensemble nature of the model helps prevent overfitting of the training data. Further, it is a robust model that handles non-linearity well. The limited initial lack of diversity within the minority classes during training may hinder the model's ability to generalise to them. The proposed model was gradient boosting, also an ensemble method. This method was chosen in preference to Random Forest due its superior ability to handle imbalanced classes, there is also the added benefit of its sequential quality which allows it to adapt to patterns missed by previous trees, hopefully therefore allowing for a greater accuracy. Both of these algorithms were implemented using the `scikit-learn` library. [4]

4 MODEL TRAINING AND EVALUATION

4.1 Training

The data transformation and analysis had a significant involvement in the model training, it was this preprocessing and target definition that laid the foundation for addressing the class imbalance as well as maintaining the model robustness.

Beyond this however both models underwent hyperparameter tuning by utilising `GridSearchCV` to explore the most effective hyperparameter combinations. For the Random Forest this meant evaluating the combinations of:

- `n_estimators`
  - Which controls the number of decision trees in the forest, where generally more trees will improve the performance of the model at the expense of computation time and the risk of overfitting. I elected for a range of 100, 200, 300 and 400 trees.
- `max_depth`
  - Which controls the depth of each decision tree , where deeper trees can observe more complex patterns but can be prone to overfitting. The given choices for the training were; 2, 5 , 10, and 20.
- `min_samples_split`
  - Which helps in the control of overfitting by avoiding small splits which overall makes the model more coarse, and less tailored to the training data. The inspected values were 2, 5 and 10
- `min_samples_leaf`
  - Which again helps with overfitting as a larger leaf node results in a more general decision boundary

For these parameters, the values `max_depth: 20`, `min_samples_leaf: 1`, `min_samples_split: 5`, and `n_estimators: 400` performed best and subsequently were used for fitting the Random Forest model onto the test data.

For gradient boosting the chosen parameters were:

- `n_estimators`, values of 100, 200, 300, and 400 were inspected
- `max_depth`, values of 3, 4, 5, and 6 were reviewed
- `learning_rate`, values of 0.1, 0.01, and 0.001 were tested

where both `n_estimators` and `max_depth` serve the same function as described previously, and `learning_rate` scales the contribution of each tree so that a high learning rate allows trees to have a strong impact on the final predictions at the risk of overfitting. A low learning rate conversely make the model more conservative and encourages a smoother convergence.

For these parameters, the values 400 for `n_estimators`, 6 for `max_depth`, and 0.1 for `learning_rate` performed best and were subsequently used for fitting the Gradient Boosting model to the test data.

4.2 Evaluation and comparison:

Accuracy:

The accuracy and F1 micro scores of the Random Forest and Gradient boosting models original test set were used as the primary analytical metrics. These were chosen as the accuracy is a useful oversight of the prediction success in its entirety while the F1 micro scores reflect the successes in each class. My models achieved an accuracy of 0.81 and 0.79 respectively and both had weighted average F1 scores of 0.79.

Overall they performed very similarly, with the Random Forest having a slight edge. It is seen in the confusion matrices in figures 3 and 4 below that neither model handled the low quality wine prediction well, with neither making a correct prediction. While this is definitely not ideal it isn't drastically concerning in respect to the original brief as it should be expected that manufacturers would want to produce high quality wines and so misclassifying a 'poor' wine as 'mediocre' is not a deterrent for the models' use. The result however does suggest that my efforts to upsample the data to increase its generalised capabilities have fallen short and potentially have caused the models to overfit to the training data for the minority classes.

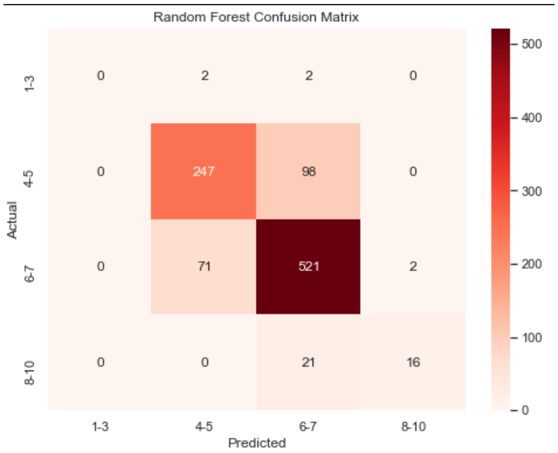


Fig. 3: Random Forest Confusion Matrix

5 CONCLUSION AND SELF REFLECTIONS

This project followed the production and development of two ensemble machine learning models [1] and their capabilities to predict the qualities of wine in the overarching goal of creating a more objective and time efficient assessment of a given wine based on its physico-chemical makeup. The project determined that the composition of white and red wines was significant enough that they could not be compared together. While both models performed incredibly similarly it was the Random Forest classifier that proved most effective overall despite the original belief that the XGB model would be more robust and better suited to the dataset. However, neither model performed perfectly and would require refinement before being used in any commercial setting. Having said this I believe that with access to a greater number of previously established 'poor' and 'excellent' wines, the variety in the data could facilitate more effective learning in the model. This premise could be of utility to producers and consumers of wine alike.

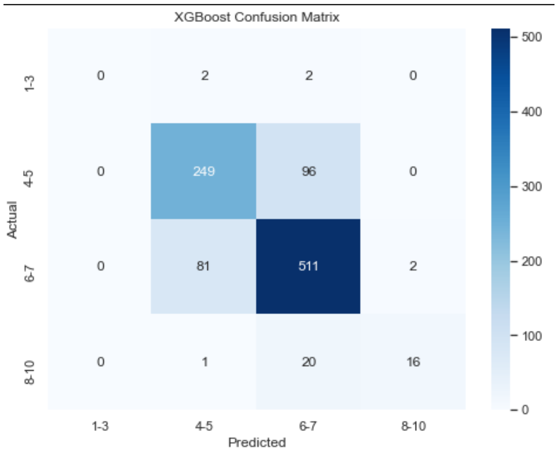


Fig. 4: Gradient Boosting Confusion Matrix

Throughout the process I used the the theoretical knowledge and understanding of machine learning provided by the lectures and have enjoyed applying them to the real world scenario which has only enhanced my understanding and provided insight to the time scale ML takes as well as the iterative nature of the process. As discussed, my biggest difficulty was tackling class imbalance which remains an issue in the model however I attempted as much as I was able to address this and enjoyed trying to improve the results despite this. In reflection it may prove useful to test other ML methods as a follow up to this report as perhaps an ensemble method was not the most suited choice.

REFERENCES

[1] Machine learning module (comp2261). 2022.
[2] Paulo Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Wine quality. *UCI Machine Learning Repository*, 2009.
[3] Tom M Mitchell. *Machine learning*, volume 1. McGraw-hill New York, 1997.
[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.