Classifying Portuguese Wine Quality: A Supervised Learning Approach

Benjamin Berkman (bjb433), Zixiao Chen (zc2194), Stanley Sukanto (ss14358), David Trakhtenberg (dt2229)

DS-GA-1001 | New York University

Project GitHub: https://github.com/dtrakht/1001-Team-Project

**Abstract**: Wine sales have dramatically increased during the coronavirus pandemic. According to the Washington Post, Americans purchased nearly 30 percent more wine in June 2020 than in March 2020[[1]](#footnote-1). Given the increased demand, quick and accurate classification of a wine’s quality is important in order to ensure a positive consumer experience and fair price at market. This paper studies if the physicochemical properties of wine can sufficiently predict its quality through supervised learning approaches. This has the potential to aid wine makers (oenologists), evaluators, and consumers. We show that a high-performing random forest algorithm is able make these classifications, and we discuss its business implications and deployment potential.

**1 Business Understanding**

Wine is a unique good that has objective physicochemical properties yet a somewhat subjective quality rating that is marketed to consumers. These physicochemical properties are readily measured via laboratory tests and include measures of alcohol percentage, pH, and residual sugar, among other features explained in this paper. The median value of three assessors’ blind tests determines a wine’s quality score (Cortez, Cerdeira and Almeida). These scores are marketed to the public, but there is often little transparency in what contributes to a good score. Thus, there exists a need to model the effect of these properties on a wine’s quality score and to be able to predict a wine’s score given its properties.

A machine learning solution has the potential to address several business problems. First, a supervised learning model will be able to identify which physicochemical properties of wine most directly influence its quality. This may allow oenologists to directly increase certain physicochemical properties in the wine making process. For example, residual sugar – one of the properties studied – can be controlled via sugar fermentation in yeasts (Cortez, Cerdeira and Almeida). Additionally, this can serve as a quality control mechanism through the wine scoring process. If the median value of the three assessors’ scores significantly differs from the projected score of the model, it could serve as a reason for re-evaluating the quality. Ultimately, human evaluators may be rendered unnecessary given enough training feedback into the supervised learning model. Moreover, a machine learning solution may help test if wine assessors in training are scoring wine on par with the model. Finally, this paper employs a brief unsupervised clustering approach to group similar wines. This may benefit consumers via targeted marketing. A vendor could employ a recommendation system, recommending wines with similar physicochemical properties given a customer enjoyed a specific wine.

As discussed in the next section, the data in this paper is limited to Portuguese white and red vinho verde wines introduced in 2009[[2]](#footnote-2). However, this opportunity has the potential to be expanded to all wines that have both measured physicochemical properties and evaluation scores. With technological increases in the wine industry, these properties are increasingly feasible to measure. Additionally, as wine is no longer seen exclusively as a luxury good, a variety of consumer-facing applications, such as CellarTracker[[3]](#footnote-3), provide millions of user ratings for millions of wines. There does not appear to exist a database that maps these user ratings to the physicochemical properties of their respective wines, but this would serve as a valuable future research topic.

The state of the art in solving this problem has largely been limited to ad-hoc studies of small datasets of varying physicochemical properties such as the Portuguese dataset in this discussion. For example, Hopfer et al. measured five quality proxies against sensory attributes, volatile compounds, and elemental composition in Californian wine (Hopfer, Nelson and Ebeler). In other cases, hierarchical clustering was applied to both physicochemical and sensory assessments of Brazilian wine (de Castilhosa, Cattelana and Conti-Silva). However, there does not appear to be a large-scale machine learning model in production that predicts wine quality given its physicochemical properties.

**2 Data Understanding**

We acquired the datasets from the UCI Machine Learning Repository. The datasets present red and white wine quality with corresponding physicochemical properties. Given their distinct flavor profiles and use cases, the red and white wines are stored in two separate datasets. This paper also includes separate analysis for red and white wines. The red and white wine datasets have 1,599 and 4,898 rows, respectively. Each row represents a distinct wine therefore the observations are independent. There are 12 variables in each dataset: 11 attributes describing the wines’ physicochemical properties and one target variable, quality (scored from 0 to 10). The 11 features are described in detail in the appendix (Table 2).

Before any data preparation or model building, we conducted exploratory data analysis (EDA) and visualized relevant results. We first analyzed the descriptive statistics of both red wine and white wine datasets, and then analyzed the correlation between each attribute and the target variable. From a correlation heat map, we observed that red wine quality is weakly positively correlated with alcohol, sulphates, citric acid, and fixed acidity, and has a weak negative correlation with other attributes. Similarly, white wine quality is weakly positively correlated to alcohol, sulphates, pH, and free sulfur dioxide, and has a weak negative correlation with other features (see appendix, Figure 2).

We also analyzed feature importance and listed the top five features in regard to both Area under the Receiver Operator Curve (AUC) and Mutual Information (MI) (see appendix, Table 3). Based on AUC, the top five features for red wine are volatile acidity, citric acid, total sulfur dioxide, sulphates, and alcohol (for white: chlorides, total sulfur dioxide, density, pH, and alcohol). A Decision Tree was learned to estimate the MI of each feature. Based on MI, the top five features for red wine are volatile acidity, total sulfur dioxide, density, sulphates, and alcohol (for white: volatile acidity, citric acid, total sulfur dioxide, sulphates, and alcohol). Thus, it stands to reason that the volume percent of alcohol in a wine, for example, will affect the sensory taste. An oenologist using the results of this simple exploratory analysis may opt to increase alcohol content, holding all else equal, in an effort to make higher quality wine.

Clustering, an unsupervised learning technique, served as a useful approach to understand the similarity of the wines in the dataset. Red wines and white wines were clustered separately using both hierarchical clustering and k-means clustering. For this exercise, the quality of the wines was not considered, only the physicochemical properties. The hierarchical clustering demonstrated that the red wines could efficiently be clustered into two clusters, while the white wines fit nicely into three clusters. K-means clustering then separated the red and white wines into two and three clusters, respectively.

As shown in the appendix (Figure 3), when plotted against density and pH, the clusters do form relatively strong boundaries. As demonstrated by Provost and Fawcett in their discussion of whiskey analytics, this clustering can help inform recommendations (Provost and Fawcett). If a customer enjoyed a specific wine, a vendor could recommend a wine with similar properties (and therefore within the same dendrogram node or k-means cluster) which in turn may have a similar taste profile. If a vendor had a limited budget, it could stock only wines within one cluster (to serve as a specialized vendor), or consciously pick wines from each cluster (to ensure a diverse selection).

With a stronger domain knowledge of the features, we then examined feature independence, a necessary assumption for several predictive modeling algorithms (e.g., Logistic Regression). Multicollinearity occurs when one independent variable can be predicted from the other independent variables. When multicollinearity is present, small changes in the data may drastically change the coefficient estimates of the regression model. However, it is important to note that multicollinearity does not affect the predictive power of a model.

One way to detect multicollinearity is to use the variance inflation factor (VIF), which is calculated as follows: where is the value obtained from regressing the predictor on all the other predictors. A VIF higher than 5 or 10 usually indicates a multicollinearity problem. Table 4: VIF shows the VIF of the independent variables that have been standardized and transformed. We observed that for red wine, fixed acidity is the feature with the highest VIF, followed closely by density. A solution to dealing with multicollinearity is to remove several features with the highest VIFs. The appendix shows the VIF of the remaining features after removing “fixed acidity”. All of the remaining features have VIF less than five, indicating that there is no multicollinearity between these features. For white wine, we observed that density is the only feature with VIF greater than five. The appendix shows the VIF of the remaining features after excluding density. There appears to be no multicollinearity among these features.

Since our goal was only to correctly predict the quality of wine given the wine features, the presence of multicollinearity was not a huge problem as we are not concerned with performing statistical inferences on the model (e.g., estimating individual feature importance). A five-fold cross-validation was utilized to determine whether or not fixed acidity and density, for red wine and white wine respectively, should be excluded when building the model. To test this, we compared the performance of two logistic regression models, the first with all of the features available and the second model with all features available except for the feature with the highest VIF. Using AUC as the evaluation metric, the first model performed slightly better for both red (.871 ± .06 vs. .869 ± .06) and white wine (.795 ± .04 vs. .793 ± .04).

**3 Data Preparation**

The dataset arrived largely pre-packaged for modeling, though we did consider several strategic decisions and feature engineering options. The physicochemical properties and qualities were collected from 2004 to 2007 by a Portuguese professional organization focused on improving the excellence and the brand value of vinho verde wines (Cortez, Cerdeira and Almeida). The data was then stored in a database and provided via UCI to machine learning researchers. The data, in its native format, contains no missing values. Both the feature and target variables all contain float or numeric field types.

The various feature variables are, however, on different scales (e.g., among the red wines, volatile acidity ranges from .12 to 1.58 while total sulfur dioxide ranges from 6 to 289). One of the first data preparation steps was thus to use Scikit-learn’s StandardScaler[[4]](#footnote-4) transformer to standardize features by removing the mean and scaling to unit variance. The data is also natively broken out by red and white wines in two separate files. As the data is hosted in .csv format (with ‘;’ delimiters), it can easily be read into a python notebook.

A first consideration involved whether to combine the red and white wine datasets into a single dataset. On one hand, red and white wines have different tasting notes so they arguably should be analyzed separately. On the other hand, a model that can successfully aggregate the two could serve as a more robust prediction system. Ultimately, we did not combine the two datasets with the rationale that the average customer is more likely to search for a “good red wine” instead of simply a “good wine”.

The target variable is defined as the quality of each wine, scored by wine assessors on a scale from 0 (very bad) to 10 (excellent). We decided to frame this business problem as a binary classification example. Instead of developing a system to predict the specific numeric quality of a wine, we instead opted to predict simply if a wine would be “good” or not (i.e., “poor quality”). This approach allowed our team to employ a more diverse set of algorithms. Additionally, classifying a wine as good or of poor quality simplifies the business implications. Still, the wines that the models predict with a very strong A picture containing shape

Description automatically generatedA picture containing shape

Description automatically generated

Figure : Red and white wine target class distribution

probability of being a good wine may be indicative of being a particularly good bottle, and therefore the model should be able to still segregate minimally good versus exceptional bottles. We classified all wines scored with a quality less than 7 to be a not good wine (binary class 0), and those with a quality greater than or equal to 7 to be a good wine (binary class 1). 14% of reds and 22% of white wines are then coded as “good”.

Finally, during EDA, we discovered that many of the features have a right-skewed distribution. This may be due to the chemical nature of the features. With residual sugar, for example, wine has varying levels of sugar after fermentation ends but various instances bunch up around 1 g/L. To normalize the skewed features, a log transformation proved most successful. Other methods may include a square root transform as well as a box-cox transform. Specifically, for our analysis, log-transforms of alcohol, residual sugar, and free sulfur dioxide remedied the adverse effects of the tail region consisting of outliers.

As a final data preparation step, the two datasets were split into training and test sets. Following data splitting standards, two thirds of the data were randomly assigned to the training set; the remaining third was assigned to the test set.

**4 Modeling & Evaluation**

By framing this business scenario as a binary classification problem instead of a regression problem, we enabled the use of a number of classification algorithms. Given the relatively small size of the data, we were not terribly constrained by computational time to train, test, tune, and retest a number of algorithms. With that in mind, we elected initially to test random forest (RF), logistic regression (LR), Gaussian naive Bayes (GNB), support vector machine (SVM), and K-nearest neighbor (K-NN) classifiers. We also separately evaluated the performance of gradient-boosting a decision tree.

We considered the assumptions, benefits, and possible issues with each algorithm. For example, random forests are extremely robust models that perform well in many situations and do not make many assumptions about the underlying data distribution; however, training time may be slow (as is the case for Support Vector Machines). Logistic regression handles class imbalance and small datasets well but requires little multicollinearity and a linear relationship between the logit of the label and each independent feature. The Gaussian Naive Bayes classifier assumes conditional independence and that the independent variables are conditionally normally distributed, but also often performs well; however, it may be unlikely that the physicochemical properties are conditionally independent. K-NN is a simple, interpretable algorithm that performs well on small datasets, and may nicely fit the structured groupings of the physicochemical properties.

Before training or evaluating any models, choosing an appropriate evaluation framework was necessary. Accuracy is not appropriate in this case due to the target class imbalance. We ultimately selected AUC for the following reasons:

* It is base rate invariant: This means we can compare the AUC performance on the white and red wine datasets
* It is easily interpretable and nicely bounded, and thus effective for communicating to business stakeholders
* As it can be analyzed at different true positive/false positive classification thresholds, we can easily interpret which model performs better at a certain threshold
* It is advantageous for ranking predictions. This benefits our business case by allowing a vendor using the model to identify wines that may be exceptionally good, even if they are only rated as “good” by an oenologist. Additionally, a vendor concerned with only stocking good wines can select wines that the model predicts with high probability are of good quality.

In pursuit of an optimal model, we aimed for generalization while avoiding overfitting. Cross-validation helped assess how the models performed against different folds of the data. This method ensured that our models generalize to various holdout sets – not just one test set. To this end, we outlined the following evaluation process:

1. Trained baseline model and evaluated performance using k-fold cross validation.
2. Performed hyperparameter tuning for each model using k-fold grid search with a defined range of possible hyperparameters and recorded respective parameters that maximize AUC.
3. Retrained each model with the optimal hyperparameters and evaluated performance, again using k-fold cross validation.

In selecting a “k” for cross validation, we considered the size of the red and white datasets and the train/test split. With a relatively small dataset, we chose k = 5 which allowed us to maintain a balance between bias (low value of k) and variance (high value of k), and to train and test our models on each fold of the data.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Red Wines | | White Wines | |
| Model | Mean AUC | Mean Standard Deviation | Mean AUC | Mean Standard Deviation |
| RF | 0.909 | 0.06 | .894 | 0.03 |
| LR | 0.871 | 0.06 | .795 | 0.04 |
| GNB | 0.861 | 0.05 | .787 | 0.05 |
| SVM | 0.868 | 0.10 | .835 | 0.02 |
| K-NN | 0.806 | 0.09 | .815 | 0.04 |

We trained the five models with default hyperparameters, evaluated performance via AUC, and plotted the ROC (receiver operating characteristic curve) curve for each model (see appendix, Figure 4). The ROC curve plots true positive rate against false positive rate at different classification thresholds. We then calculated the mean AUC and standard deviation across the 5 folds, shown in Table 1. The Random Forest performed notably better than the other four models, and the ROC curve indicates it outperforms at a majority of classification thresholds.

Table : Baseline model performance

To perform hyperparameter tuning, we employed grid search to take a range of values for each hyperparameter and construct a grid of all possible combinations in an attempt to maximize AUC. For example, with LR, we tuned ‘C’ (the regularization weight), ‘penalty’ (specific norm for penalization), and ‘solver’ (specific algorithm used), which converged to an optimal set at C = 0.1, penalty = ‘L1’, and solver = ‘liblinear’ for red wine and C = 10, penalty = ‘L2’, and solver = ‘saga’ for white wine. An optimal set is defined by the hyperparameters that yield the highest AUC. In defining the range for ‘C’ (and other continuous hyperparameters), we made sure to select values that would span model complexity from low to high (thus varying bias and variance). Applying this approach to each algorithm, we were able to find the optimal set of hyperparameters to retrain the models. A summary of the tested and optimal hyperparameters can be found in the appendix (Table 5).

As a final tuning step, we also evaluated if we could notably improve the random forest’s performance using a Gradient Boosting method. In this algorithm, an initial decision tree is fit, then a decision tree is fit to the error of the prior model. This is repeated until a stopping criterion is met. Despite the strong performance often seen through this approach, the Gradient Boosted model did not improve the AUC of the classification and is not justified given its substantial increase in training and tuning time.

**Works Referenced:**

Cortez, Paulo, et al. "Modeling wine preferences by data mining from physicochemical properties." *Decision Support Systems* (2009): 4800-058.

de Castilhosa, Maurício, et al. "Influence of two different vinification procedures on the physicochemical and sensory properties of Brazilian non-Vitis vinifera red wines." *LWT - Food Science and Technology* (2013): 360-366.

Hopfer, Helen, et al. "Correlating Wine Quality Indicators to Chemical and Sensory Measurements ." *Molecules* (2015): 8453-8483.

Provost, Foster and Tom Fawcett. *Data Science for Business*. O'Reilly, 2013.

**Appendix:**

**Data Understanding:**

Table : Dataset features and descriptions

|  |  |  |
| --- | --- | --- |
| Attribute | Description | Unit |
| Fixed Acidity | Acids enhance the fundamental tastes in wine, such as sourness and tartness. Most acids in wine are fixed acids. | mg/L |
| Volatile Acidity | The steam distillable acids in wine. Usually kept at low level to avoid undesirable aromas. | g/L |
| Citric Acid | A less common acid in wine. Used for stabilization purposes to prevent ferric hazes and should be kept below 1 g/L. | g/L |
| Residual Sugar | The natural sugar leftover after alcoholic fermentation finishes. | g/L |
| Chlorides | The major contributor to saltiness in wine. The common range is from 2 to 4 g/L. | g/L |
| Free Sulfur Dioxide | The amount of free SO2 in wine. Used for preventing microbial growth and oxidation. | mg/L |
| Total Sulfur Dioxide | The portion of free SO2 and bound SO2 to other chemicals. | mg/L |
| Density | The weight per volume of a material. Density of wine is mainly determined by the concentration of alcohol, sugar, and glycerol. | g/mL |
| pH | Measures ripeness versus acidity. The common pH range for wine is from 3.3 to 3.6. The higher the pH, the lower the acidity. | 1-14 scale |
| Sulphates | The amount of salts of sulfuric acid in wine. Used for inhibiting microbial growth. | g/L |
| Alcohol | The percent alcohol content of the wine. | % |
| Quality | A score measures wine quality. | 0-10 scale |

Figure : Correlation Heat Maps

*Red Wine: White Wine:*

Graphical user interface, chart, application, treemap chart, PowerPoint

Description automatically generated Chart, treemap chart

Description automatically generated

Table : Feature Importance

*Red Wine: White Wine:*

*Table

Description automatically generatedTable

Description automatically generated*

Figure : Clustering

*A picture containing icon

Description automatically generated*

*Chart, histogram

Description automatically generated*

A picture containing chart

Description automatically generated Chart, scatter chart

Description automatically generated

Table : VIF

*Red Wine:*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Feature | VIF | | Volatile Acidity | 1.866445 | | Citric Acid | 2.834463 | | Residual Sugar | 1.598921 | | Chlorides | 1.379659 | | Free Sulfur Dioxide | 1.961269 | | Total Sulfur Dioxide | 2.170959 | | Density | 2.747043 | | pH | 1.621783 | | Sulphates | 1.340303 | | Alcohol | 2.316157 |  |  |  | | --- | --- | | Feature | VIF | | Fixed Acidity | 7.814194 | | Volatile Acidity | 1.866593 | | Citric Acid | 3.084088 | | Residual Sugar | 2.002017 | | Chlorides | 1.448900 | | Free Sulfur Dioxide | 1.969745 | | Total Sulfur Dioxide | 2.240458 | | Density | 7.528407 | | pH | 3.324713 | | Sulphates | 1.393500 | | Alcohol | 3.474619 | |  |

*White Wine:*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Feature | VIF | | Fixed Acidity | 1.828567 | | Volatile Acidity | 1.156441 | | Citric Acid | 1.153274 | | Residual Sugar | 4.542915 | | Chlorides | 1.393057 | | Free Sulfur Dioxide | 1.736008 | | Total Sulfur Dioxide | 2.199714 | | Density | 10.357201 | | pH | 1.586563 | | Sulphates | 1.098830 | | Alcohol | 4.587571 | | |  |  | | --- | --- | | Feature | VIF | | Fixed Acidity | 1.348393 | | Volatile Acidity | 1.155812 | | Citric Acid | 1.143687 | | Residual Sugar | 1.401792 | | Chlorides | 1.380040 | | Free Sulfur Dioxide | 1.718938 | | Total Sulfur Dioxide | 2.178811 | | pH | 1.325385 | | Sulphates | 1.068640 | | Alcohol | 1.749781 | |

**Modeling and Evaluation:**

Figure : Baseline model ROC curves

*Red Wines White Wines*

A picture containing diagram

Description automatically generated**A picture containing chart

Description automatically generated**

Table : Grid Search Results

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm** | **Hyperparameter** | **Red Optimal Set** | **White Optimal Set** |
| RF | Criterion | Gini | Entropy |
| Max\_Depth | 50 | None |
| Min\_Samples\_Split | 2 | 2 |
| Min\_Samples\_Leaf | 1 | 1 |
| Class\_Weight | Balanced | Balanced |
| LR | C | 0.1 | 1 |
| Penalty | L2 | L2 |
| Solver | Liblinear | Liblinear |
| GNB | Var\_Smoothing | 46.416 | 0.000534 |
| SVM | C | 1 | 1 |
| Gamma | Scale | Auto |
| Kernel | RBF | RBF |
| K-NN | Algorithm | Auto | Auto |
| Leaf\_Size | 1 | 1 |
| N\_Jobs | -1 | -1 |
| N\_Neighbors | 9 | 9 |
| Weights | Distance | Distance |

1. https://www.washingtonpost.com/lifestyle/food/wine-sales-have-surged-during-the-pandemic--but-not-for-small-producers/2020/06/19/c7c49cba-b0b6-11ea-8f56-63f38c990077\_story.html [↑](#footnote-ref-1)
2. https://archive.ics.uci.edu/ml/datasets/wine+quality [↑](#footnote-ref-2)
3. https://www.cellartracker.com/ [↑](#footnote-ref-3)
4. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html [↑](#footnote-ref-4)