Wine Color and Quality Prediction using Optimization Method and Machine Learning

Julian Sim, Li Jiang, Qikai Jiang, Hongyu Yu, Qinghua Li

Introduction

A natural consequence of graduate school is that students often find themselves exploring wine (n = 1). For many new wine drinkers, understanding what makes a wine "good" becomes an intriguing and sometimes elusive question. While wine appreciation is inherently subjective, there is growing interest in identifying objective qualities that contribute to the overall perception of wine quality. In this context, the Wine Quality dataset, sourced from wines produced in the Vinho Verde region of Portugal, provides a unique opportunity to examine the factors influencing wine quality. Known for its distinctive winemaking characteristics, the dataset, available through the UCI Machine Learning Repository, offers valuable insights into the question: What makes a wine "good"?

We seek to model quality to understand which attributes are the strongest and most accurate predictors, as well as to explore whether these attributes may differ by wine color. The quality score, a continuous variable ranging from 3 to 9, reflects the subjective evaluation of professional tasters who assess each wine based on a combination of sensory factors, such as taste, aroma, appearance, and mouthfeel. These scores are not just numerical ratings but encapsulate expert opinions on a wine's overall drinking experience.

Given this framework, our analysis will explore several key questions. First, which physicochemical properties—such as alcohol content, acidity, sugar levels, and pH—are most strongly associated with wine quality scores? Second, what chemical features are most predictive of a wine's color, specifically distinguishing between red and white wines? Third, do certain attributes influence red and white wines differently, and if so, how do these differences affect perceived quality? By examining these questions, we aim to gain deeper insight into what defines a "good" wine and how wine color may shape the relationship between chemical composition and quality.

The Vinho Verde Wine Quality dataset from the UCI Machine Learning Repository contains 6,497 wine samples, each with 13 different variables. Among these, two variables are treated as outcomes: color and quality, while the remaining eleven are considered as features or predictors. The variables are as follows:

1. Color: A binary variable indicating the type of wine --- red or white.

2. Ordinal Quality: A continuous variable representing the subjective assessment of the wine's overall quality, scored on a scale from 3 to 9, based on sensory evaluations by professional tasters.

3. Binary Quality: For easing the classification, we dichotomize the quality to < 7 and ≥ 7.

The remaining 11 covariates are physicochemical properties of the wine, which are believed to influence its overall quality and can be grouped into six aspects. One important aspect is acids, encompassing fixed acids, volatile acids, and citric acid. Fixed acids remain in the wine when it is boiled and contribute to its tartness and sharpness. Volatile acids, on the other hand, evaporate easily, and if present in excess, can impart a vinegar-like smell that negatively affects the taste and quality of the wine. Citric acid, a naturally occurring organic acid, enhances the wine’s freshness and tartness, playing a key role in its balance and perceived acidity.

Another important aspect is sugar, specifically residual sugar --- the amount remaining in the wine after fermentation—which directly shapes the wine’s sweetness and overall flavor profile. Salts, such as chlorides (primarily sodium chloride), can influence both taste and mouthfeel, with excessive levels potentially introducing off-flavors. Sulphates play a key role in preservation, but when present in high concentrations, they may contribute to undesirable aromas. Acidity, typically measured by pH, also plays a crucial role in determining a wine’s taste, color, and stability; lower pH values indicate higher acidity and sharper notes, while higher pH suggests a softer, rounder profile.

Stabilizing agents play a crucial role in wine quality. Free sulfur dioxide (SO_2) serves as both an antioxidant and antimicrobial agent, helping to preserve freshness and prevent spoilage; however, excessive levels can negatively affect the wine’s aroma and flavor. Total SO_2, the sum of free and bound forms, similarly helps control oxidation and microbial growth but may diminish sensory quality when too high. In terms of alcohol-related properties, density—defined as mass per unit volume—is influenced by alcohol content, residual sugar, and water, and should be interpreted in the context of wine color and fermentation completeness. Alcohol content itself, measured as the percentage by volume, strongly impacts the wine’s body, taste, and mouthfeel, with higher levels typically producing fuller-bodied wines with more intense flavors.

Table 1: Dataset Description (n = 6,497)

Variable name	Role	Type	Mean (SD)	Median (min, max)	Missing values
fixed acidity	Feature	Continuous	7.22 (1.30)	7.00 (3.80, 15.9)	no
volatile acidity	Feature	Continuous	0.34 (0.17)	0.29 (0.08, 1.58)	no
citric acid	Feature	Continuous	0.32 (0.15)	0.31 (0, 1.66)	no
residual sugar	Feature	Continuous	5.44 (4.76)	3 (0.6, 65.8)	no
chlorides	Feature	Continuous	0.06 (0.04)	0.05 (0.01, 0.61)	no
free sulfur dioxide	Feature	Continuous	30.5 (17.7)	29 (1, 289)	no
total sulfur dioxide	Feature	Continuous	116 (56.5)	118 (6, 440)	no
density	Feature	Continuous	1.00 (0.003)	1.00 (0.99, 1.04)	no
pH	Feature	Continuous	3.22 (0.16)	3.21 (2.72, 4.01)	no
sulphates	Feature	Continuous	0.53 (0.15)	0.51 (0.22, 2)	no
alcohol	Feature	Continuous	10.5 (1.19)	10.3 (8, 14.9)	no
quality	Outcome	Continuous	5.82 (0.87)	6 (3, 9)	no
color	Outcome	Binary (red/white)	Red 1,599 (24.6%) White 4,898 (75.4%)		no
binary quality	Outcome	Binary (≥7/<7)	≥7: 1,277 (19.7%) <7: 5,220 (80.3%)		no

Note: n (%) was used for binary variables.

Next, we examine the descriptive statistics presented in Table 1 of the Wine Quality Dataset. All features, or physicochemical properties, are recorded as continuous variables. Total SO_2 shows a wide range (6 to 440), while density has a narrower range (0.99 to 1.04). We will standardize those features to a mean of 0 and a variance of 1. The ordinal outcome for wine quality has a mean of 5.82 and a standard deviation of 0.86, with a distribution close to normal. Binary outcomes are relatively rare, with red wine comprising 24.6% and wines rated with a quality score of 7 making up 19.7%. Notably, there are no missing values in the dataset.

Methods and Results

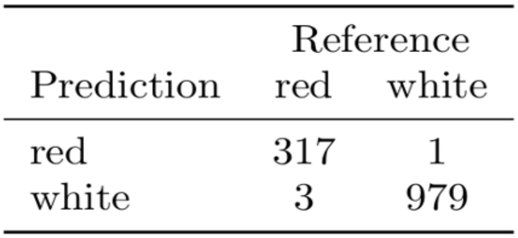
Data-Preprocessing

Data were randomly split 80:20 into training and test sets. Wine quality was dichotomized at a threshold of 7. We examined the correlation matrix to rule out collinearity, then centered and scaled all predictors in both sets to aid convergence and ease the model comparison. Models were trained on the training set using cross-validation and their performance evaluated on the independent test set.

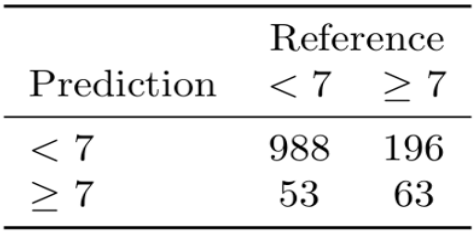
Simple logistic regression

Firstly, we established the simplest classifiers by fitting two independent logistic-regression models using the R built-in “glm” package. In each case, we treated one of our binary outcomes (red vs. white color; “high” vs. “low” quality) as the response and regressed it on the full set of remaining chemical predictors. All continuous inputs had been pre-scaled, and we used the default binomial link to estimate log-odds. Model parameters were estimated via maximum likelihood on the training set, and the performance was assessed via classification accuracy, sensitivity, and specificity on the test data. As a baseline model, we evaluated each classifier’s performance using confusion matrices at a probability threshold of 0.5. The baseline color classifier achieved excellent accuracy on the test set, correctly identifying almost all red and white wines. In contrast, the baseline quality classifier performed poorly: a large proportion of high-quality wines were misclassified as low-quality. (Table 2)

Table 2: Confusion Matrices for Wine Color and Quality using Test set



	Reference	
Prediction	red	white
red	317	1
white	3	979



	Reference	
Prediction	< 7	≥ 7
< 7	988	196
≥ 7	53	63

Bayesian logistic regression

To begin expanding on our simplest classifier of simple linear regression, we introduced a Bayesian component to the model. Bayesian analysis has two primary benefits for modeling: firstly, Bayesian analysis outputs posterior distributions, which not only allow for point estimates as in frequentist analysis, but gives us a distribution which allows for probabilistic statements otherwise impossible. This allows for more natural interpretations, as well as allowing for a more natural incorporation of uncertainty. Secondly, and of primary interest for our analysis, was the potential of including a prior. Informative priors allow for historical data, domain knowledge, or information from different models to be included into the regression, thus stabilizing estimates and improving accuracy.

However, one of the major issues with this stratagem is due to our train/test split. It is highly unlikely that the train data splits utilized in other models from other authors only contained data already present within our training data. Therefore, this causes some problems in interpretation as our prior is indirectly incorporating data within our test split, and therefore, using such an external prior gives our Bayesian analysis some advantage over the remaining models. Utilizing an internal prior is also not ideal, as utilizing our training data simultaneously for our likelihood and our prior is likely to cause problems with overfitting.

Therefore, the most ideal prior to be used is one reliant on domain knowledge. However, as none of our group members were expert sommeliers, we decided to utilize an "inexpert" prior, where we weighted parameters that we thought were predictive of good wine quality highly or a white wine highly, and weighted parameters that we associated with poor wine quality or a red wine negatively. To accomplish this weighting, parameters that we associated positively with high quality wine for that color or parameters we felt were more strongly associated with a specific color were weighted as one, and parameters negatively associated were as negative one. Then, these were scaled by their relative standard deviations, thus equally weighting all parameters.

When choosing the distribution of these priors, we chose to utilize a normal prior as we have a large sample size, and therefore, we somewhat rely on the Bayesian Central Limit Theorem for rough normality of posterior. In order to test this "inexpert" prior, we also fit a non-informative prior in order to test our assumptions about wine quality.

In order to fit this model, we employed Markov Chain Monte Carlo with the Metropolis Hastings algorithm. We choose Metropolis-Hastings over Gibbs sampling due to the ease of access for the conditional distributions. Although the Bayesian Central Limit Theorem does give that the posterior approaches a normal, this does not mean that the conditionals benefit from conjugacy in order to have simple forms. Therefore, as these conditionals do not have easily expressed closed forms, we proceed with Metropolis-Hastings.

One of the primary simulation parameters to fit for Metropolis Hastings is the choice of proposal standard deviation. Allowing for too large jumps leads to missing the correct minimization and therefore leading to ballooning parameter estimates, and only allowing miniscule jumps causes a correspondingly long iteration time. Therefore, in order to fit our proposal deviation, we began by fitting multiple non-informative models and strove to choose one which converged relatively quickly, with acceptance ratios that started off around 15-50% and decayed considerably post a burn-in of around 10,000. This resulted in a proposal of about 0.5 for all of our models. We now move to examining our trace plots for red wine quality, with a non-informative prior on the left, and an informative prior on the right.

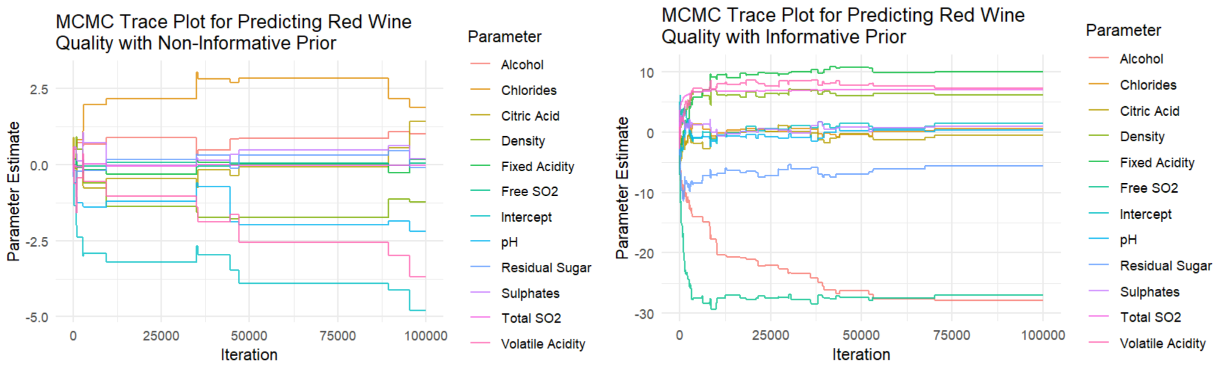
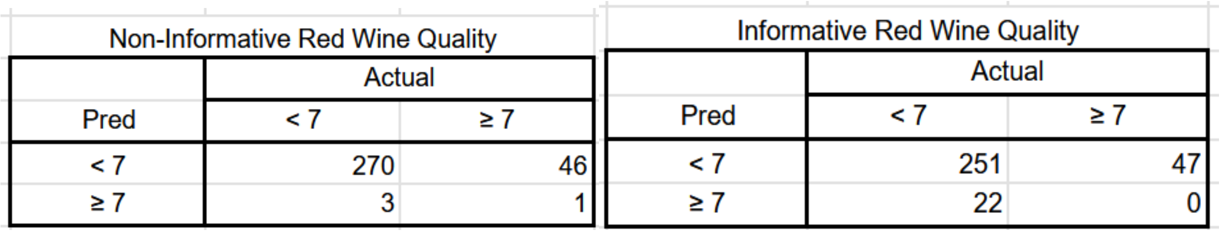


Figure 1: MCMC Trace Plots

We note that given a prior, our model converges relatively quickly to the optimal parameters, and thus stabilizes our parameter results more quickly. However, given a prior, the MCMC chooses to fit larger parameter estimates, but we see that this trend is shown in both positive and negative parameter estimates, and therefore it is probable that these trends counteract each other. To determine this, we assess their performance via a confusion matrix, utilizing a majority vote by the posterior with a burn-in of 25000.

Table 3: Confusion Matrices for Wine Quality by Prior using Test Set



Non-Informative Red Wine Quality

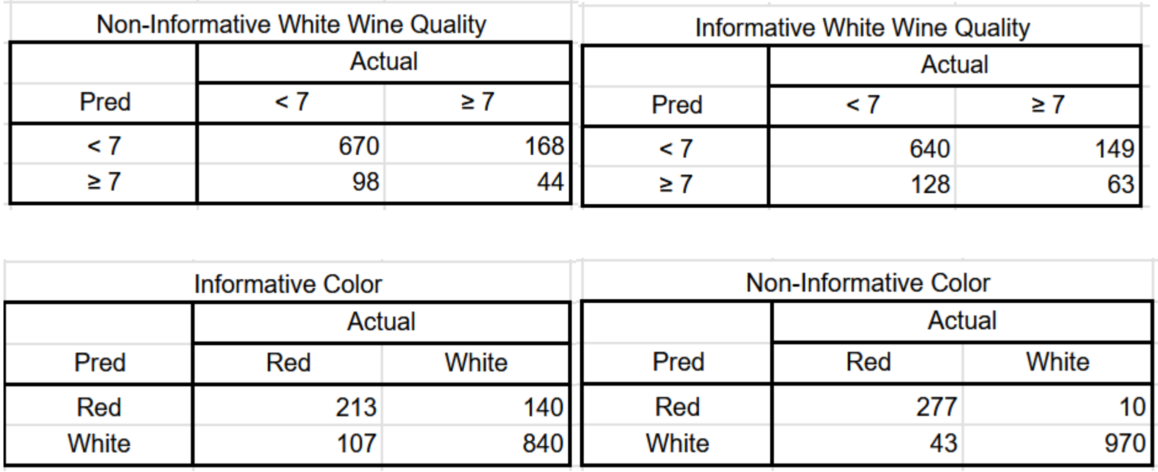
	Actual	
Pred	< 7	≥ 7
< 7	270	46
≥ 7	3	1

Informative Red Wine Quality

	Actual	
Pred	< 7	≥ 7
< 7	251	47
≥ 7	22	0

Here, we see that our informative “inexpert” prior performed worse than the non-informative prior, suggesting that our prior was actively harming the analysis, although not by much. We saw a similar trend within our other analyses with the Bayesian Logistic Regression with inexpert prior, although our white wine prior had somewhat comparable performance instead of being strictly worse.

Table 4: Confusion Matrices for Wine Color and Quality by Prior using Test Set



Non-Informative White Wine Quality

	Actual	
Pred	< 7	≥ 7
< 7	670	168
≥ 7	98	44

Informative White Wine Quality

	Actual	
Pred	< 7	≥ 7
< 7	640	149
≥ 7	128	63

Informative Color

	Actual	
Pred	Red	White
Red	213	140
White	107	840

Non-Informative Color

	Actual	
Pred	Red	White
Red	277	10
White	43	970

This leads us to suspect that our inexpert prior was not a good choice and therefore, allowing for a simple non-informative prior may be a better choice.

Binomial regression & Newton Raphson for quality (now quality being ordinal)

Since the outcome, wine quality, is an integer ranging from 3 to 9, we first subtract all quality by 3 so that the range is between 0 and 6. We model the transformed outcome as a binomial distribution. Specifically, we assume $Y_i \sim \mathrm{Bin}(n, p_i)$ with $g(\mathrm{E}[Y_i]) = g(np_i) = \mathbf{x}_i^\top \boldsymbol{\beta}$. The goal is to estimate $\boldsymbol{\beta}$ given the data. Here we choose to use the canonical link

$$g(y) = \log \frac{y}{n-y}.$$

To derive this canonical link, we first write the likelihood in exponential family form:

$$p(Y_i|p_i) \propto \exp\left(Y_i \log p_i + (n - Y_i)\log(1-p_i)\right) = \exp\left(Y_i \log \frac{p_i}{1-p_i} + n\log(1-p_i)\right),$$

so the canonical parameter is $\theta_i = \mathrm{logit}(p_i)$. Now,

$$g(np_i) = \log \frac{np_i}{n - np_i} = \log \frac{p_i}{1-p_i} = \mathrm{logit}(p_i) = \theta_i,$$

that is, $\theta_i = \mathbf{x}_i^\top \boldsymbol{\beta}$, which shows that g is indeed the canonical link.

To estimate $\boldsymbol{\beta}$ by maximizing the likelihood, we choose to use Newton-Raphson. The joint log-likelihood is $\ell(\boldsymbol{\beta}) = \sum_{i=1}^m Y_i \mathbf{x}_i^\top \boldsymbol{\beta} - n\log(1+\exp(\mathbf{x}_i^\top \boldsymbol{\beta}))$. The first derivative is

$$\partial_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}) = \sum_{i=1}^m \left(Y_i - \frac{n\exp(\mathbf{x}_i^\top \boldsymbol{\beta})}{1+\exp(\mathbf{x}_i^\top \boldsymbol{\beta})}\right)\mathbf{x}_i^\top = \sum_{i=1}^m (Y_i - np_i)\,\mathbf{x}_i^\top = \mathbf{X}^\top \mathbf{S},$$

where $\mathbf{S} \in \mathbb{R}^n$ with $S_i = Y_i - np_i$. The second derivative is

$$\partial^2_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}) = \sum_{i=1}^{m} -n \frac{\exp(\mathbf{x}_i^\top \boldsymbol{\beta})}{(1+\exp(\mathbf{x}_i^\top \boldsymbol{\beta}))^2} \mathbf{x}_i \mathbf{x}_i^\top = -\mathbf{X}^\top \mathbf{V} \mathbf{X},$$

where $\mathbf{V}$ is a diagonal matrix with the ith diagonal element equal to the factor in the summation. Finally, the Newton-Raphson update rule is $\hat{\boldsymbol{\beta}}^{(t+1)} \leftarrow \hat{\boldsymbol{\beta}}^{(t)} + (\mathbf{X}^\top \mathbf{V} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{S}|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}^{(t)}}$ given the tth iteration estimate is $\hat{\boldsymbol{\beta}}^{(t)}$. It remains to choose the initial value for the Newton-Raphson algorithm. We use the estimates from the following transformed linear model as the starting point:

$$\log \frac{\tilde{Y}_i}{n - \tilde{Y}_i} = \mathbf{x}_i^\top \boldsymbol{\beta} + \varepsilon_i, \quad \varepsilon_i \sim_{\text{i.i.d.}} \mathrm{N}(0, \sigma^2),$$

where $\tilde{Y}_i = Y_i$ if $Y_i \notin \{0, n\}$, $\tilde{Y}_i = 0.01$ if $Y_i = 0$, and $\tilde{Y}_i = n - 0.01$ if $Y_i = n$ to avoid dividing by zero or taking log of zero. Note this transformed model has a very similar form as the binomial regression model except the expectations are replaced by the actual values. Thus the maximum likelihood estimates from this model should serve as a good approximate to the binomial regression estimates. Moreover, the linear model has a closed form solution: $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \tilde{\mathbf{Y}}$, so the computational cost of initializing Newton-Raphson is relatively low.

We note that an alternative of binomial regression is ordinal regression, which directly models the probability that the score is below certain values. Binomial regression imposes stronger distributional assumptions and thus might be more restrictive than ordinal regression. However, binomial regression is also easier to compute and might have higher statistical power when the sample size is small.

We also derive the Wald test for testing $H_0 : \beta_i = 0$. In the most general form, the Wald test statistic is

$$R(\hat{\boldsymbol{\beta}})^\top (H(\hat{\boldsymbol{\beta}}) I_n^{-1}(\hat{\boldsymbol{\beta}}) H(\hat{\boldsymbol{\beta}})^\top)^{-1} R(\hat{\boldsymbol{\beta}}) \rightarrow_d \chi^2_r,$$

where R is a set of constraints imposed by the null, and H is the first order derivative of R. In the case of $H_0 : \beta_i = 0$, $R(\boldsymbol{\beta}) = \beta_i$ and thus $H(\boldsymbol{\beta})$ is a zero vector with its ith entry equal to one. The Wald test statistic thus has the following form:

$$\frac{(\hat{\beta}_i)^2}{[(\mathbf{X}^\top \mathbf{V} \mathbf{X}|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}})^{-1}]_{ii}} \rightarrow_d \chi^2_1,$$

and it it asymptotically χ^2_1.

The following figure shows the coefficient estimates from Newton-Raphson at each iteration when fitting the model to the wine data. With the initial value calculated from the transformed model, Newton-Raphson converges in 3 iterations with a convergence threshold of 1e-10. This shows that the algorithm converges quite fast in real data. Moreover, note the initial estimates from the transformed model are already close to the final estimates in magnitude, so the initial value computation procedure we proposed is effective.

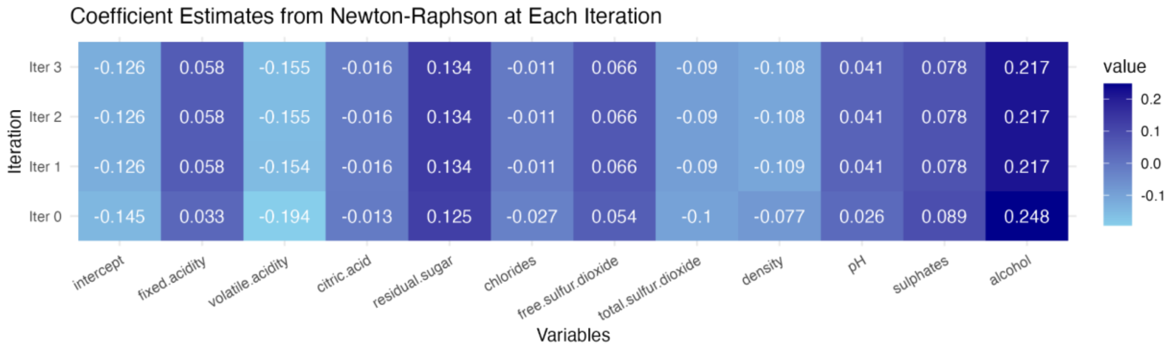


Figure 2: Coefficient Estimates at Each Iteration

We next investigate the predictive power of our binomial regression model. Specifically, we calculate the mean absolute error (MAE) between the predicted wine score and the true wine score on the test dataset. Since the binomial regression outputs a fraction instead of a number, we computed the MAE using both the raw fraction and the rounded fraction (so that the predicted values are also integers). The MAE from the raw fraction is 0.559, and the MAE from the rounded fraction is 0.518. In addition, we plot the MAE for each wine score. As shown below, the binomial regression model tends to predict well for wine qualities equal to 5, 6, or 7 while it has a large error for both very high and very low wine quality scores. We found that the predicted power is negatively correlated with the number of samples in each category. The imbalance of training data leads to the imbalance predictability.

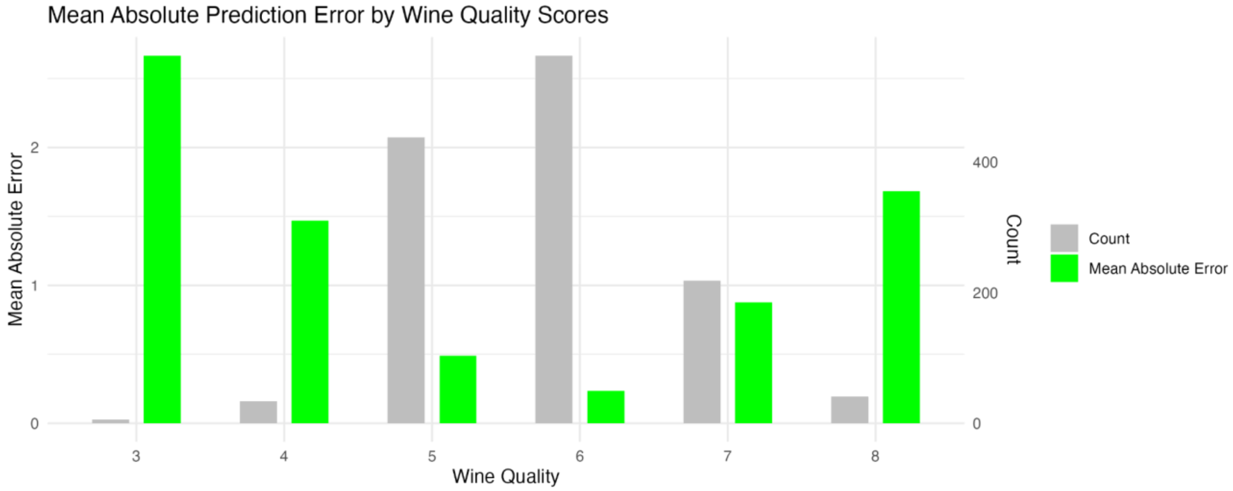


Figure 3: Mean Absolute Prediction Error by Wine Quality Scores

Finally, we perform Wald tests to find statistically significant features. Based on the test result, the top three factors are volatile acidity, sulphates, and alcohol. The two factors with p-values larger than 0.05 are citric acid and chlorides, indicating that they are not predictive in the model.

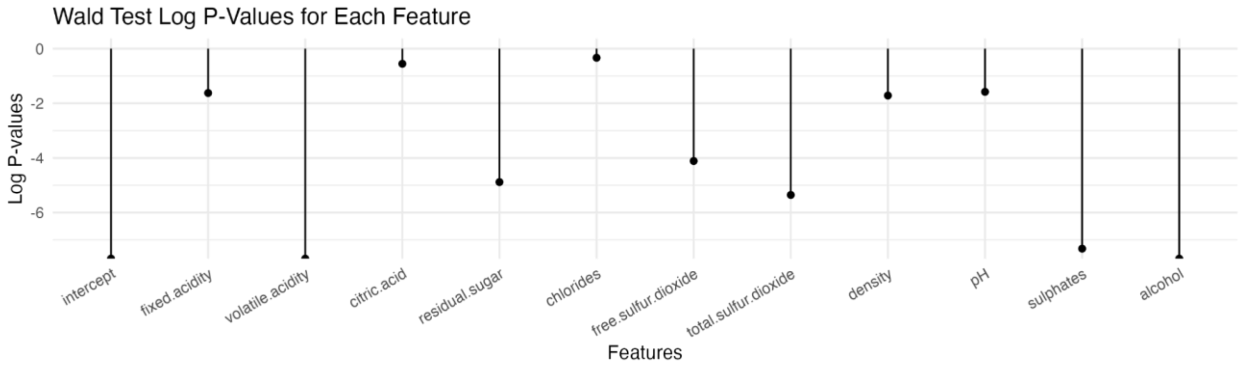


Figure 4: Wald Test Log P-values for Each Feature

Support Vector Machine for Classification (SVC) for both quality & color

SVC with linear kernel and radial kernel were both implemented. In a linear kernel SVM, the classifier tries to find a hyperplane that best separates two classes of data using a linear decision boundary (Figure 1). The kernel function is $K(x_i, x_j) = x_i^T x_j$, and it works in the original linear space using $f(x) = wx - b$. The red line the hyperplane or decision boundary, and it separates the two classes with margin $wx - b = \pm 1$. w is the weight vector which is perpendicular to the decision boundary and b is the bias or intercept which shifts the hyperplane away from the origin, adjusting the hyperplane's position without changing its direction. The data points that lie exactly on the margin lines (on either side) are called support vectors. (Figure 5 left panel)

Radial kernel SVM can be used for classification where the decision boundaries are not linear. $K(x_i, x_j) = exp(-\gamma||x_i - x_j||^2)$, where γ controls the influence of each training point. A small γ means far points are considered similar, leading to smoother boundaries. A large γ results in tighter fits around individual data points. (Figure 5 right panel)

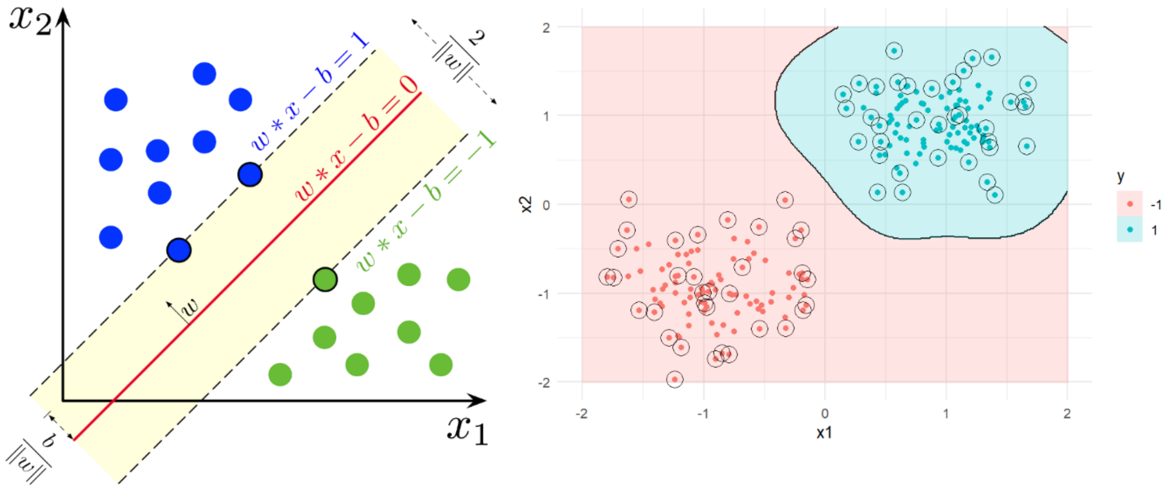


Figure 5: Support vector machine linear kernel (left) and radial kernel (right).

Both linear and radial kernels were applied to the SVM model. Model was trained using the training subset with 10-fold CV, and was evaluated using the test subset. Table 5 summarizes the confusion matrix for each outcome. Figure 6 shows the feature importance for different outcomes using linear and radial kernels. Feature importance was identified by absolute value of weight for linear kernel and feature permutation for radial kernel. Notice that both SVM linear and radial kernels separated the wine color very well, with accuracy 0.9985, precision 1, and recall 0.9938 (Table 7). However, SVM did not separate the wine quality well with either linear kernel or radial kernel.

Table 5: Confusion Matrix for Wine Color and Wine Quality using Test Set

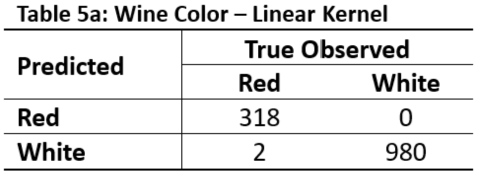


Table 5a: Wine Color – Linear Kernel

Predicted	True Observed	
	Red	White
Red	318	0
White	2	980

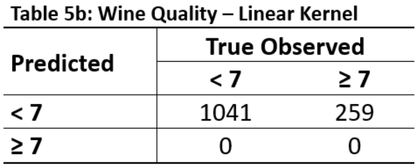


Table 5b: Wine Quality – Linear Kernel

Predicted	True Observed	
	< 7	≥ 7
< 7	1041	259
≥ 7	0	0

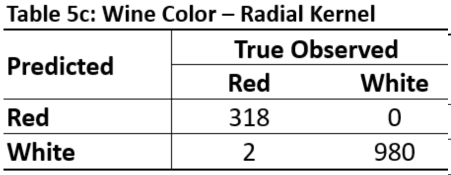


Table 5c: Wine Color – Radial Kernel

Predicted	True Observed	
	Red	White
Red	318	0
White	2	980

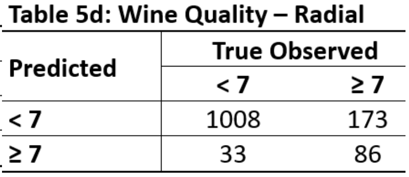


Table 5d: Wine Quality – Radial

Predicted	True Observed	
	< 7	≥ 7
< 7	1008	173
≥ 7	33	86

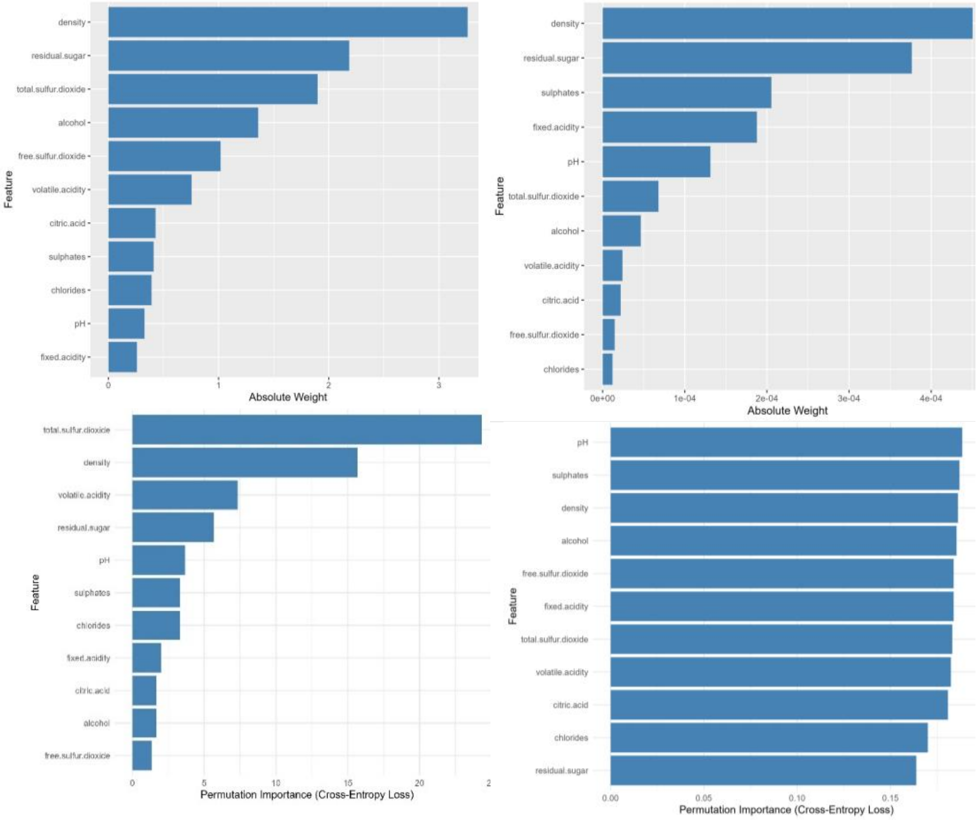


Figure 6: Feature Importance. Top panel: linear kernel for wine color (left) and wine quality (right). Bottom panel – radial kernel for wine color (left) and wine quality (right).

Multi-layer perceptrons for both quality & color

To capture the potential non-linear effects and interactions, we trained multilayer perceptrons (MLPs) with two hidden layers using the “caret” package in R. After numerically encoding color and quality as binary, we performed a grid search over several configurations of first-layer (n=5,4,3) and second-layer (n=3,2,1) neuron counts, and we used five-fold cross-validation to get the best model based on ROC and guard against overfitting. Each candidate network employed logistic activation functions to apply to the weighted combination of the incoming nodes and was trained until convergence within a generous iteration budget. The cross-validated tuning results guided our final choice of architecture, and the performance was then assessed by fitting on the test data.The best color network used a (4, 3) architecture and achieved excellent classification accuracy, while the optimal quality network used a (5, 3) architecture—improving on the baseline logistic model but still misclassifying a notable fraction of high-quality wines as low-quality.

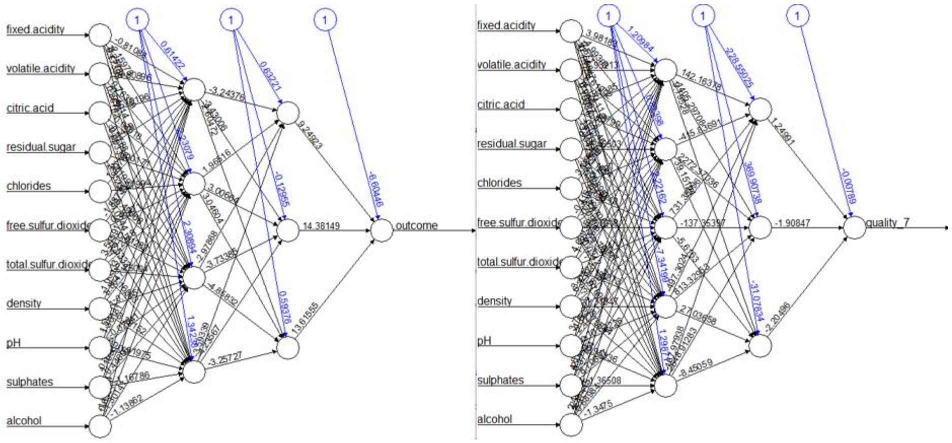
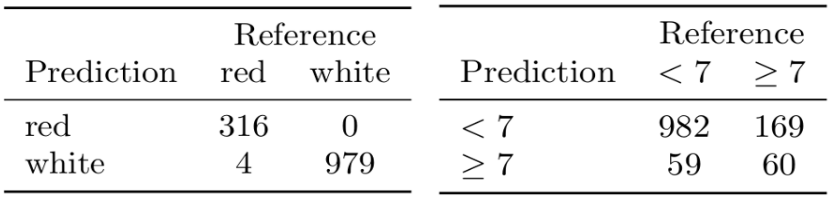


Figure 7: Multilayer Perceptron Architecture for Wine Color and Quality

Table 6: Confusion Matrix using Test Set



	Reference	
Prediction	red	white
red	316	0
white	4	979

	Reference	
Prediction	< 7	≥ 7
< 7	982	169
≥ 7	59	60

Table 7: Model Comparison using Accuracy, Precision, and Recall

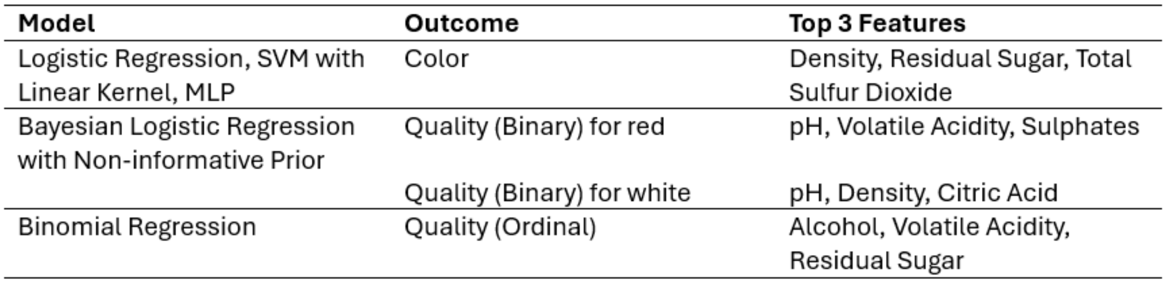
Method	Wine Quality			Wine Color		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Logistic Regression	0.8085	0.5431	0.2432	0.9969	0.9969	0.9906
Bayesian Logistic Regression	Inf: 0.7334 No: 0.7577	Inf: 0.8559 No: 0.9030	Inf: 0.8197 No: 0.8146	Inf: 0.8100 No: 0.9592	Inf: 0.8571 No: 0.9897	Inf: 0.8870 No: 0.9576
SVC	L: 0.8008 R: 0.8415	L: NA R: 0.7227	L: 0 R: 0.3320	L: 0.9985 R: 0.9985	L: 1 R: 1	L: 0.9938 R: 0.9938
MLP	0.8246	0.6040	0.3475	0.9969	1	0.9875

As shown in Table 7, we compared four classification models—Logistic Regression, Support Vector Machines (SVM), Multi-Layer Perceptron (MLP), and Bayesian Logistic Regression—on two binary classification tasks using the Vinho Verde dataset: predicting wine color (red vs. white) and identifying high-quality wines (score ≥ 7). All models performed exceptionally well in predicting wine color, with Logistic Regression, SVM, and MLP each achieving accuracy, precision, and recall above 98%. In contrast, Bayesian Logistic Regression underperformed slightly in this task. This is likely due to the use of a non-informative prior, which can cause over-regularization by shrinking coefficients toward zero and softening the decision boundary, even when the data are clearly separable. Additionally, Bayesian model averaging introduces prediction uncertainty by integrating over a distribution of parameter values, resulting in more conservative estimates.

In predicting whether a wine is high quality (defined as a rating of 7 or above), Bayesian Logistic Regression (BLR) outperformed Logistic Regression, SVM, and MLP in both precision and recall (see Table 7). BLR's advantage lies in its ability to incorporate prior beliefs and average over multiple models, which prevents overconfidence and enhances sensitivity to the minority class --- crucial in imbalanced datasets. In contrast, standard Logistic Regression, relying on maximum likelihood estimation, tends to overfit the majority class due to its use of a single point estimate. MLP models, which require large, balanced datasets and careful hyperparameter tuning, struggled with the small, imbalanced nature of this dataset, resulting in overfitting and poor minority class performance. SVM with a linear kernel underperformed due to the assumption of linear separability, while the RBF kernel, although more flexible, still lacked calibrated probabilities, complicating threshold selection and tending to favor the majority class. Although BLR did have the lowest performance in terms of raw accuracy, it substantially improved on both precision and recall compared to other methods which made it our model of choice as we believe the tradeoff is worth it.

In conclusion, as shown in the table, all models demonstrated strong predictive performance for wine color, with Logistic Regression emerging as the preferred model due to its simplicity and interpretability. In contrast, predicting wine quality proved more challenging; however, Bayesian Logistic Regression showed a notable improvement over other models, particularly in terms of precision and recall. Given its superior performance and ability to handle class imbalance, the Bayesian Logistic Regression with a non-informative prior is the most appropriate model for this task.

Table 8: Important Features from Different Models



Model	Outcome	Top 3 Features
Logistic Regression, SVM with Linear Kernel, MLP	Color	Density, Residual Sugar, Total Sulfur Dioxide
Bayesian Logistic Regression with Non-informative Prior	Quality (Binary) for red	pH, Volatile Acidity, Sulphates
	Quality (Binary) for white	pH, Density, Citric Acid
Binomial Regression	Quality (Ordinal)	Alcohol, Volatile Acidity, Residual Sugar

Finally, let’s examine the physicochemical properties that distinguish high-quality wines and differentiate red from white. To do so, we evaluated the top three features selected using different models (Table 8). For wine color prediction, models such as logistic regression, SVM with a linear kernel, and MLP consistently identified density, residual sugar, and total sulfur dioxide as the most influential features. These results are consistent with chemical differences --- white wines typically have higher residual sugar and density, while red wines often undergo different fermentation and preservation processes, influencing sulfur dioxide levels. For quality prediction, we used Bayesian Logistic Regression with a non-informative prior and analyzed red and white wines separately. In red wines, the most important predictors were pH, volatile acidity, and sulphates, which contribute to the wine’s structure, freshness, and age-worthiness. In contrast, white wine quality was best predicted by pH, density, and citric acid, reflecting their crisp, acidity-driven profiles. Lastly, for ordinal quality prediction, binomial regression identified alcohol, volatile acidity, and residual sugar as key features, aligning with expert knowledge that high-quality wines tend to have higher alcohol content, lower volatility, and well-balanced sweetness.

Reference

P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553. ISSN: 0167-9236.
https://en.wikipedia.org/wiki/Support_vector_machine