Classification of White Wine Quality Based on Logistic Regression

Hongchi Zhou
Department of Applied Mathematics and Statistics
Johns Hopkins University
Baltimore, MD 21218
hzhou66@jh.edu

Qiqing Gao
Department of Applied Mathematics and Statistics
Johns Hopkins University
Baltimore, MD 21218
qgao19@jh.edu

Abstract

The physicochemical properties of white wine directly affect its quality. This study develops a classification model to predict white wine quality based on physicochemical data. The proposed method combines logistic regression with feature selection techniques and Bayesian analysis to address a binary classification problem. Experimental results on a training dataset of 2296 white wine samples and a testing dataset of 984 samples yield prediction accuracies of 72.3% and 71.6%, respectively. This work provides a comparison between Frequency and Bayesian approaches, offering insights for improving the fitting of our Bayesian model.

1 Purpose of the project

1.1 Data Source and Data Collection

In this study, we analyzed a dataset on `vinho verde` white wine, which is collected from UCI machine Learning Library [1]. The dataset was obtained from the `CVRVV`. The data collection process took place between 2004 and 2007, during the certification of wines.

The dataset contains 4,898 samples of white wines, each of which was evaluated on the basis of its physicochemical tests and evaluations by tasting technicians. Physicochemical attributes tested in the dataset included acidity,etc. In addition, sensory evaluations were derived from scores given by expert tasters.The final sensory quality scores were aggregated using the median score for each sample, ensuring robustness in the quality measurement.

One of the main reasons we chose this dataset was its relatively large sample size and data variable refinement. Compare this to other datasets, such as the `Birthwt` dataset in MASS, which often have small sample sizes. This dataset provides an excellent basis for Bayesian statistical modeling and classification tasks.

1.2 Variables in the Dataset

The dataset includes both quantitative and qualitative variables. Below is a description of the key variables:

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

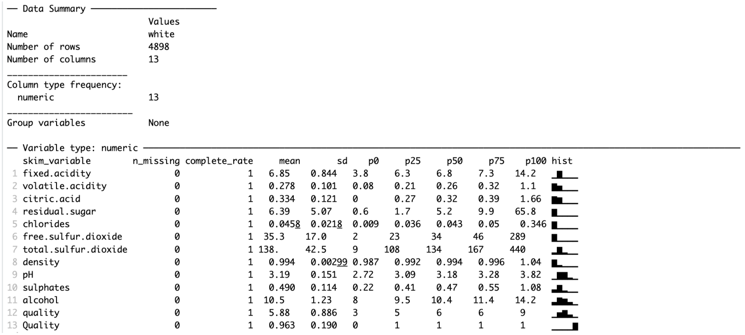
Table 1: Wine Quality Dataset Variables

Variable Name	Type	Description
fixed.acidity	Quantitative	Fixed acids, mainly tartaric acid (g/L)
volatile.acidity	Quantitative	Acetic acid concentration, contributes to vinegar taste (g/L)
citric.acid	Quantitative	Citric acid concentration, adds freshness to wine (g/L)
residual.sugar	Quantitative	Sugar remaining after fermentation (g/L)
chlorides	Quantitative	Salt content in wine (g/L)
free.sulfur.dioxide	Quantitative	Free SO_2, prevents microbial growth (mg/L)
total.sulfur.dioxide	Quantitative	Total SO_2, sum of free and bound forms (mg/L)
density	Quantitative	Density of wine, related to sugar and alcohol (g/cm³)
pH	Quantitative	pH level, acidity of wine
sulphates	Quantitative	Potassium sulfate content, contributes to flavor (g/L)
alcohol	Quantitative	Alcohol content in wine by volume
quality	Qualitative	Quality score, sensory rating (scale 0–10)

1.3 Data Cleaning

The `vinho verde` dataset was checked in R for missing values, etc., and it was finally confirmed that no additional cleaning was required. All observations were complete and the range of variables in the dataset corresponded to the expected characteristics of the physicochemical measurements and sensory evaluations. However, the dataset suffers from a category imbalance and we will undersample the dataset in the subsequent data analysis section to ensure that the dataset can be used for Bayesian statistical modeling and analysis.

1.4 Summary Statistics of Key Variables



```
── Data Summary ────────────────────────
                           Values
Name                       white
Number of rows             4898
Number of columns          13
_______________________
Column type frequency:
  numeric                  13
________________________
Group variables            None

── Variable type: numeric ──────────────────────────────────────────────────────────────────────────────
   skim_variable        n_missing complete_rate    mean      sd    p0      p25     p50     p75    p100 hist
 1 fixed.acidity                0             1   6.85    0.844   3.8      6.3     6.8     7.3    14.2
 2 volatile.acidity             0             1   0.278   0.101   0.08     0.21    0.26    0.32    1.1
 3 citric.acid                  0             1   0.334   0.121   0        0.27    0.32    0.39    1.66
 4 residual.sugar               0             1   6.39    5.07    0.6      1.7     5.2     9.9    65.8
 5 chlorides                    0             1   0.0458  0.0218  0.009    0.036   0.043   0.05    0.346
 6 free.sulfur.dioxide          0             1  35.3    17.0     2       23      34      46     289
 7 total.sulfur.dioxide         0             1 138.     42.5     9      108     134     167     440
 8 density                      0             1   0.994   0.00299 0.987    0.992   0.994   0.996   1.04
 9 pH                           0             1   3.19    0.151   2.72     3.09    3.18    3.28    3.82
10 sulphates                    0             1   0.490   0.114   0.22     0.41    0.47    0.55    1.08
11 alcohol                      0             1  10.5     1.23    8        9.5    10.4    11.4    14.2
12 quality                      0             1   5.88    0.886   3        5       6       6       9
13 Quality                      0             1   0.963   0.190   0        1       1       1       1
```

Figure 1: Summary of Numerical Characteristics

The dataset contains 13 numeric variables, as summarized in Figure 1. Key statistics such as mean, standard deviation, and percentiles provide insights into the variability of these features. For instance, residual sugar has a mean of 6.39 g/L with a standard deviation of 5.07, indicating substantial variability among samples. Similarly, free sulfur dioxide levels range from 2 to 289 mg/L, with a mean of 35.3 mg/L, reflecting diverse preservation practices across wines. Alcohol content, with a mean of 10.5%, highlights its potential influence on wine quality.

1.5 Distribution of Wine Quality Ratings

The wine quality ratings, ranging from 3 to 9, are distributed with a median value of 6, as histogram shown in Figure 1. Most samples are concentrated around scores of 5, 6, and 7, indicating a predominance of mid-quality wines. This distribution will be important for addressing potential class imbalance in the classification model.

1.6 Multicollinearity and Standardization

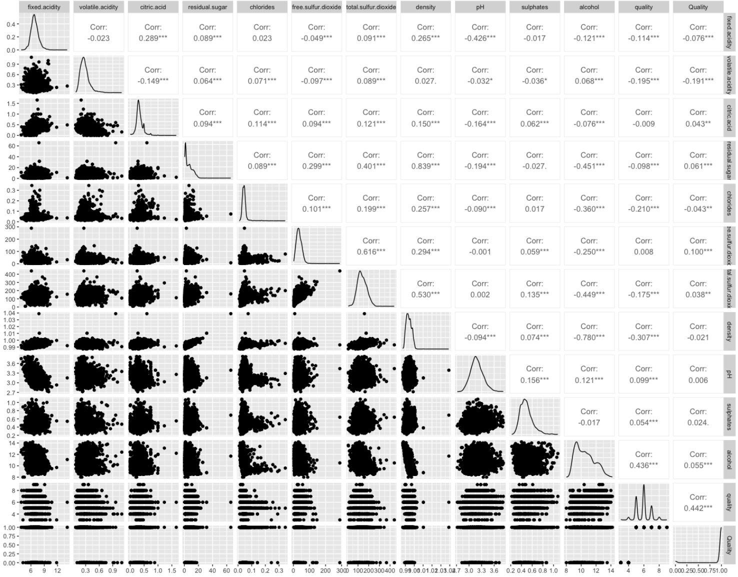


Figure 2: Scatterplot Matrix with Correlation Matrix

Correlation analysis (Figure 2) shows strong relationships between certain variables, which may indicate multicollinearity. For example, alcohol and density exhibit a strong negative correlation (corr = -0.780), suggesting that as alcohol content increases, density decreases significantly. This inverse relationship is not only chemically intuitive but also highlights a potential redundancy between these two variables when included together in a predictive model. Such situation could affect model performance, requiring careful consideration of variable selection.

The scatterplot matrix highlights the necessity of standardizing the dataset. Variables such as residual sugar (ranging from 0.6 to 65.8 g/L) and pH (ranging from 2.72 to 3.82) exist on vastly different scales. Without standardization, features with larger ranges might disproportionately influence the model coefficients, leading to biased results. Additionally, potential outliers are observed in residual sugar and total sulfur dioxide, which may require further investigation to assess their impact on the model.

2 Literature Review

Overview of Logistic Regression Cox's logistic regression model, first proposed in 1958, has long since become a reliable method for binary classification problems.[2] The interpretability and broad applicability of this model allow researchers to understand the relationship between predictor variables and outcomes while maintaining computational efficiency.

Historical Applications Hosmer and Lemeshow (1989) have deepened the understanding and application of logistic regression through their seminal work, Applied Logistic Regression.[4] They emphasize diagnostics and goodness of fit, providing us with usable tools for assessing the reliability of models in subsequent analyses.

Applications in Wine Quality Analysis In wine analysis, Data Collector (2009) used logistic regression to identify physicochemical factors influencing sensory evaluations [5]. Studies categorizing wine quality into two classes highlighted alcohol content and sulphates as key predictors, consistent with our dataset and guiding our analysis.

Two Approaches in Logistic Regression Logistic regression is usually performed using the frequentist and Bayesian paradigms. The frequentist approach obtains point estimates using maximum likelihood estimation (MLE). This method is computationally efficient and the results are easy to interpret, especially when the sample size is large.[2]The Bayesian approach integrates prior knowledge with observed data to estimate posterior parameter distributions, providing a probabilistic framework that accounts for uncertainty. This method gained prominence with Albert and Chib (1993), who utilized Markov Chain Monte Carlo (MCMC) techniques for efficient posterior estimation [7].

3 Proposed Method

3.1 Research Question

The dataset gives the physicochemical properties and quality assessment of different white wine samples. However, we still do not know the specific relationship between physicochemical properties and quality. Therefore, our goal is to build a white wine quality classification model based on 11 features.

3.2 Bayesian Model

3.2.1 Logistic Regression

We define the classification variable Q as

$$Q = \begin{cases} 1, \text{high quality, if quality } \geq 6 \\ 0, \text{low quality, if quality } < 6 \end{cases} \tag{1}$$

Therefore, we use logistic regression for this binary classification problem. We have the logistic regression model as below

$$\Pr[Q] = \begin{cases} \frac{\exp(\mathbf{X}\boldsymbol{\beta})}{1+\exp(\boldsymbol{X}\boldsymbol{\beta})}, & Q = 1 \\ \frac{1}{1+\exp(\boldsymbol{X}\boldsymbol{\beta})}, & Q = 0 \end{cases} \tag{2}$$

where $\mathbf{X}$ is design matrix, $\boldsymbol{\beta}$ is 12 by 1 regression coefficients.

3.2.2 Model Fitting: Metropolis Algorithm

We use Bayesian inference to estimate our regression coefficients $\boldsymbol{\beta}$. The key algorithm of model fitting is Metropolis. Therefore, we need to identify our prior distribution, sampling model, and proposal distribution.

For prior distribution, we do not have enough information about our prior. Therefore, we tend to use uninformative normal prior as our model prior, which is

$$\beta_i \overset{iid}{\sim} N(0, 10^2) \tag{3}$$

where $i = 0, 1, \cdots, 11$.

For the sampling model, we can calculate the sampling model based on the equations of the logistic regression model (2), which is

$$\begin{aligned} p(y_1, \cdots, y_n|\boldsymbol{\beta}, \mathbf{X}) &= \prod_{i=1}^{n} p(y_i|\boldsymbol{\beta}, \mathbf{X}) \\ &= \prod_{i=1}^{n} \left(\frac{\exp(\boldsymbol{X}\boldsymbol{\beta})}{1+\exp(\boldsymbol{X}\boldsymbol{\beta})} \right)^{y_i} \left(\frac{1}{1+\exp(\boldsymbol{X}\boldsymbol{\beta})} \right)^{1-y_i} \\ &= \prod_{i=1}^{n} \frac{\exp(y_i \boldsymbol{X}\boldsymbol{\beta})}{1+\exp(\boldsymbol{X}\boldsymbol{\beta})} \end{aligned} \tag{4}$$

For the proposal distribution, we choose the variance of the proposal distribution as $\delta^2 = 0.025^2$ to make the acceptance ratio lie in $20\% - 50\%$. Therefore, we have our proposal distribution as below

$$J(\boldsymbol{\beta}^*|\boldsymbol{\beta}) \sim MVN(\boldsymbol{\beta}, \delta^2 \mathbf{I}_{12\times12}) = MVN(\boldsymbol{\beta}, 0.025^2 \cdot \mathbf{I}_{12\times12}) \tag{5}$$

Based on prior distribution, sampling model and proposal distribution, our Metropolis sampling procedures are as below

$$\begin{array}{ll} \text{Step 1} & : \text{ Sample a proposal value } \boldsymbol{\beta}^* \sim J(\boldsymbol{\beta}^*|\boldsymbol{\beta}^{(s)}) \sim MVN(\boldsymbol{\beta}^{(s)}, 0.025^2 \cdot \mathbf{I}_{12\times 12}) \\ \text{Step 2} & : \text{ Compute } r = \frac{p(\boldsymbol{\beta}^*|y_1,\cdots,y_n,\mathbf{X})}{p(\boldsymbol{\beta}^{(s)}|y_1,\cdots,y_n,\mathbf{X})} = \frac{p(y_1,\cdots,y_n|\boldsymbol{\beta}^*,\mathbf{X})}{p(y_1,\cdots,y_n|\boldsymbol{\beta}^{(s)},\mathbf{X})} \times \frac{p(\boldsymbol{\beta}^*)}{p(\boldsymbol{\beta}^{(s)})}, \text{ then compute } \log r \\ \text{Step 3} & : \text{ let } u \sim uniform[0,1],\ \boldsymbol{\beta}^{(s+1)} = \begin{cases} \boldsymbol{\beta}^*, \text{ if } \log r > \log u \\ \boldsymbol{\beta}^{(s)}, \text{ if } \log r < \log u \end{cases} \end{array}$$

3.2.3 Model Selection: Metropolis-Hasting Algorithm

According to the Numerical Characteristics above, we notice that there are high correlation coefficients between some features, which indicates that there may exist multicollinearity in our model. Therefore, we need to take model selection to get best model and eliminate multicollinearity.

Same as model fitting, we also use Bayesian inference to select useful features in our model. We will use Metropolis-Hasting algorithm for our model selection. We need to determine our prior distribution and proposal distribution.

For model selection, We rewrite the degree of the exponential term in the equations of the logistic regression model (2) as below

$$\beta_0 + \beta_1\gamma_1 x_{i1} + \beta_2\gamma_2 x_{i2} + \cdots + \beta_{11}\gamma_{11}x_{i,11} \tag{6}$$

where $\gamma_j = \{0, 1\}$, $j = 1, \cdots, 11$. If we drop feature β_j, then $\gamma_j = 0$; if we remain feature β_j, then $\gamma_j = 1$.

For the prior distribution, we still use the prior of β_i in model fitting part as our prior of regression coefficients. And for γ_j, $j = 1, \cdots, 11$, we have the prior as below

$$\gamma_j \overset{iid}{\sim} Bernoulli(\frac{1}{2}),\ j = 1, \cdots, 11 \tag{7}$$

For the proposal distributions, we choose the variance of proposal distribution of β_j as $\sigma^2 = 0.035^2$ to make the acceptance ratio lie in 20% $-$ 50%. Therefore, we have our proposal distributions of γ_j and β_j as below

$$J(\gamma_j^*|\boldsymbol{\gamma}) \overset{iid}{\sim} Bernoulli\left(\frac{1}{11}\right) \tag{8}$$

$$J(\boldsymbol{\beta}^*|\boldsymbol{\beta}) \sim MVN(\boldsymbol{\beta}, \sigma^2\mathbf{I}_{11\times 11}) = MVN(\boldsymbol{\beta}, 0.035^2\mathbf{I}_{11\times 11}) \tag{9}$$

where $\boldsymbol{\gamma} = \{\gamma_1, \cdots, \gamma_{11}\}$, $\boldsymbol{\beta} = \{\beta_1, \cdots, \beta_{11}\}$.

Based on the prior distributions, sampling model and proposal distributions, our Metropolis-Hasting algorithm are as below

$$\begin{array}{lll} \boldsymbol{\gamma}: & \text{Step 1}: & \text{Sample } \gamma_j^* \sim J(\gamma_j^*|\boldsymbol{\gamma}^{(s)}) \sim Bernoulli\left(\frac{1}{11}\right)\ j = 1, \cdots, 11 \\ & \text{Step 2}: & \text{Compute } r = \frac{p(\boldsymbol{\gamma}^*|\boldsymbol{\beta}^{(s)},\mathbf{y},\mathbf{X})}{p(\boldsymbol{\gamma}^{(s)}|\boldsymbol{\beta}^{(s)},\mathbf{y},\mathbf{X})} = \frac{p(\mathbf{y}|\boldsymbol{\beta}^{(s)},\boldsymbol{\gamma}^*,\mathbf{X})}{p(\mathbf{y}|\boldsymbol{\beta}^{(s)},\boldsymbol{\gamma}^{(s)},\mathbf{X})}\frac{p(\boldsymbol{\beta}^{(s)})p(\boldsymbol{\gamma}^*)}{p(\boldsymbol{\beta}^{(s)})p(\boldsymbol{\gamma}^{(s)})}, \text{ compute } \log r \\ & \text{Step 3}: & \text{let } u \sim uniform[0,1],\ \boldsymbol{\gamma}^{(s+1)} = \begin{cases} \boldsymbol{\gamma}^*, \text{ if } \log r > \log u \\ \boldsymbol{\gamma}^{(s)}, \text{ if } \log r < \log u \end{cases} \\ \boldsymbol{\beta}: & \text{Step 1}: & \text{If } \gamma_j^{(s+1)} = 1, \text{ then sample } \beta_j^* \sim J(\beta_j^*|\boldsymbol{\beta}^{(s)}) \sim N(\beta_j^{(s)}, 0.035^2)\ j = 1, \cdots, 11 \\ & \text{Step 2}: & \text{Compute } r = \frac{p(\boldsymbol{\beta}^*|\boldsymbol{\gamma}^{(s+1)},\mathbf{y},\mathbf{X})}{p(\boldsymbol{\beta}^{(s)}|\boldsymbol{\gamma}^{(s+1)},\mathbf{y},\mathbf{X})} = \frac{p(\mathbf{y}|\boldsymbol{\gamma}^{(s+1)},\boldsymbol{\beta}^*,\mathbf{X})}{p(\mathbf{y}|\boldsymbol{\gamma}^{(s+1)},\boldsymbol{\beta}^{(s)},\mathbf{X})}\frac{p(\boldsymbol{\gamma}^{(s+1)})p(\boldsymbol{\beta}^*)}{p(\boldsymbol{\gamma}^{(s+1)})p(\boldsymbol{\beta}^{(s)})}, \text{ compute } \log r \\ & \text{Step 3}: & \text{let } u \sim uniform[0,1],\ \boldsymbol{\beta}^{(s+1)} = \begin{cases} \boldsymbol{\beta}^*, \text{ if } \log r > \log u \\ \boldsymbol{\beta}^{(s)}, \text{ if } \log r < \log u \end{cases} \end{array}$$

where $\boldsymbol{\gamma} = \{\gamma_1, \cdots, \gamma_{11}\}, \boldsymbol{\beta} = \{\beta_1, \cdots, \beta_{11}\}, \mathbf{y} = \{y_1, \cdots, y_n\}$.

4 Data Analysis and Result Interpretation

4.1 Data Rebalancing and Scaling

Based on our analysis in the Summary Statistics of Key Variables, the dataset is imbalanced, with a ratio of $1640 : 3258$ between samples with quality scores below 6 and those above 6. Such imbalance can negatively impact the classification performance of logistic regression. To rebalance the dataset, we resample a new dataset from the original dataset. We remain 1640 samples with quality scores below 6 and randomly select 1640 samples from those with quality scores above 6 as our new dataset with size 3280.

We also notice that some features have large order of magnitude differences. Therefore, we standardize the new dataset to eliminate the impact of different orders of magnitude on model fitting.

In order to make the fitting results more convincing, we divide the new data set into training set and test set in a ratio of $7 : 3$. The data sets used in the following model fitting and model selection are all training sets.

4.2 Bayesian Analysis

4.2.1 Model Fitting

In order to make the Markov chain generated by the Metropolis algorithm reach a stable state, we set the number of iterations $S = 15000$ and obtain a thinning Markov Chain with a sample size of 5000. The sampling acceptance rate of the algorithm is 0.42. At the same time, the traceplot and acf plot of different coefficients also show that the Markov chains are basically stable. We take the posterior mean of the generated samples as our estimate. The traceplot with sample efficient sizes and ACF plots of different coefficients β_i are shown below

Table 2: Estimates with S_{eff} of β_i

β	Estimate	S_{eff}	β	Estimate	S_{eff}
β_0	0.025	322.050	β_6	0.171	181.969
β_1	0.062	36.914	β_7	-0.023	157.400
β_2	-0.744	135.090	β_8	-1.074	8.644
β_3	0.010	188.582	β_9	0.177	41.100
β_4	1.084	18.226	β_{10}	0.233	272.215
β_5	0.048	130.938	β_{11}	0.777	26.379

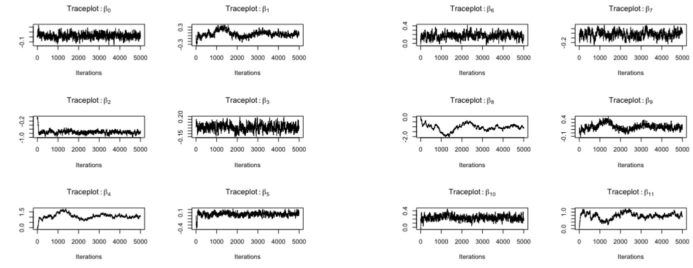


Figure 3: Traceplots of Coefficients β_i

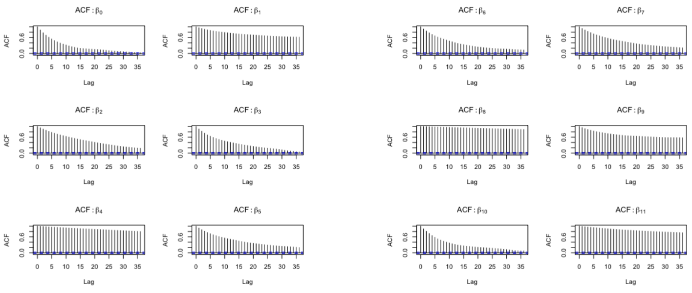


Figure 4: ACF Plots of Coefficients β_i

4.2.2 Model Selection

To ensure that the Markov chain generated by Metropolis-Hastings algorithm reaches a steady state, we set the number of iterations to $S = 8000$. And the sampling acceptance rate of the algorithm of $\boldsymbol{\gamma}, \boldsymbol{\beta}$ are 0.391 and 0.424, respectively. At the same time, the traceplot and acf plot of $\boldsymbol{\beta} \times \boldsymbol{\gamma}$ also show that the Markov chains are basically stable.Similarly as for model fitting, we also use the posterior mean of the generated samples of $\boldsymbol{\gamma}^{(s)}$ and $\boldsymbol{\beta}^{(s)}$ as our estimates. The traceplots with sampling efficient sizes and ACF plots of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are as below

Table 3: Estimates with S_{eff} of $\beta_j \times \gamma_j$

Coefficient	Estimate	$S_{eff}(\beta)$	$S_{eff}(\gamma)$	Coefficient	Estimate	$S_{eff}(\beta)$	$S_{eff}(\gamma)$
$\beta_1 \times \gamma_1$	-0.193	297.426	179.778	$\beta_7 \times \gamma_7$	0	5.974	72.168
$\beta_2 \times \gamma_2$	-0.755	131.366	664.593	$\beta_8 \times \gamma_8$	0	2.352	59.395
$\beta_3 \times \gamma_3$	0	3.566	142.278	$\beta_9 \times \gamma_9$	0	6.598	144.875
$\beta_4 \times \gamma_4$	0.418	164.464	54.773	$\beta_{10} \times \gamma_{10}$	0.147	96.058	84.319
$\beta_5 \times \gamma_5$	0	45.876	85.280	$\beta_{11} \times \gamma_{11}$	1.275	76.862	0
$\beta_6 \times \gamma_6$	0.175	50.474	85.260	β_0	0.018	282	

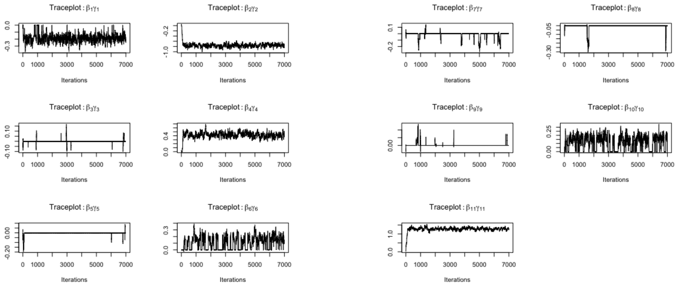


Figure 5: Traceplots of $\beta_j \times \gamma_j$

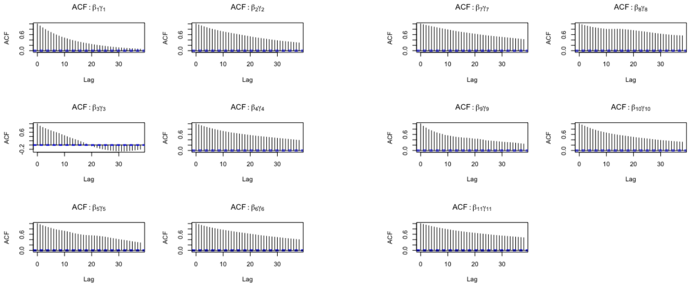


Figure 6: ACF Plots of $\beta_j \times \gamma_j$

We can get $\boldsymbol{\gamma}$ with the highest probability 0.347 of samples is $(1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1)$, which means that we retain features fixed.acidity, volatile.acidity, residual.sugar, free.sulfur.dioxide, sulphates and alcohol as our best model. And the degree of the exponential term in the equations of our final logistic regression model (2) is as below

$$\begin{aligned} \hat{\beta}_0 + \hat{\beta}_1 \text{fixed.acidity} + \hat{\beta}_2 \text{volatile.acidity} + \hat{\beta}_4 \text{residual.sugar} + \\ \hat{\beta}_6 \text{free.sulfur.dioxide} + \hat{\beta}_{10} \text{sulphates} + \hat{\beta}_{11} \text{alcohol} \end{aligned} \tag{10}$$

4.3 Comparison Between Frequency and Bayesian

For the same dataset, we also use Frequency approach to fit logistic regression model. We fit the logistic regression model using the least squares method and perform model selection using stepwise selection, retaining features volatile.acidity, residual.sugar, free.sulfur.dioxide, pH, sulphates and alcohol.

To compare the models based on two approaches, we use confusion matrix and prediction accuracy to measure the effectiveness of model classification. We compare the confusion matrix and accuracy of the two approaches in the training and test sets. And confusion matrices with accuracy of two methods are as below

Table 4: Accuracy of Two Approaches in Different Dataset

Approach	Frequency	Bayesian
Training Set	0.726	0.723
Testing Set	0.714	0.716

$$\text{Confus(Freq,training)} = \begin{bmatrix} & 0 & 1 \\ 0 & 833 & 325 \\ 1 & 304 & 834 \end{bmatrix} \quad \text{Confu(Freq,testing)} = \begin{bmatrix} & 0 & 1 \\ 0 & 362 & 140 \\ 1 & 141 & 341 \end{bmatrix} \tag{11}$$

$$\text{Confus(Bayes,training)} = \begin{bmatrix} & 0 & 1 \\ 0 & 831 & 329 \\ 1 & 306 & 830 \end{bmatrix} \quad \text{Confu(Bayes,testing)} = \begin{bmatrix} & 0 & 1 \\ 0 & 358 & 134 \\ 1 & 145 & 347 \end{bmatrix} \tag{12}$$

From the comparison results above, we notice that

- Frequency: In the case of large samples (2296 samples in the training set), it has higher prediction accuracy; it is more computationally efficient.
- Bayesian: It performs better in small dataset (984 samples in the testing set), but it requires the generation of long Markov chains and thus has a greater computational cost.

5 Model Improvement

5.1 Classification Quality Scores

We define the classification variable $Q = \begin{cases} 1, \text{high quality, if quality } \geq 6 \\ 0, \text{low quality, if quality } < 6 \end{cases}$ previously.

Problem The binary classification standard for high and low quality based on quality scores is subjective. The quality distribution of original dataset is concentrated around 5 and 6, with these scores representing medium-quality white wine and sharing similar physicochemical properties[1]. However, dividing scores 5 and 6 into separate categories (e.g., assigning samples with quality score 5 to high quality $Q = 1$ and those with quality 6 to low quality $Q = 0$) may impact classification results.

Solution Redefine the classification criteria for classification variable Q

$$\text{e.g. } Q = \begin{cases} 1, \text{high quality, if quality } \geq 5 \\ 0, \text{low quality, if quality } < 5 \end{cases} \quad \text{or} \quad Q = \begin{cases} 1, \text{high quality, if quality } \geq 7 \\ 0, \text{low quality, if quality } < 7 \end{cases}$$

i.e. make samples with quality score $5, 6$ be grouped into a same category.

5.2 Choice of Prior Distribution

Problem The choice of prior distribution may not be optimal. Without prior knowledge about the dataset, we use an uninformative normal prior, which may reduce regression accuracy and lead to classification errors.

Solution We can use Jeffrey's Prior $p(\beta_i) \sim \sqrt{\det I(\beta_i)}$, where $I(\beta_i) = -\mathbb{E}\left[\frac{\partial^2 \log P(y|X,\beta_i)}{\partial \beta_i^2}\right]$ or other uninformative prior, such as uniform prior, Laplace prior[8].

References

[1] Cortez, P., Cerdeira, A., Almeida, F., Matos, T. & Reis, J. (2009) Wine Quality [Dataset]. *UCI Machine Learning Repository.* DOI: `https://doi.org/10.24432/C56S3T`.

[2] Cox, D.R. (1958) The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)* **20**(2):215–242.

[3] Walker, S.H. & Duncan, D.B. (1967) Estimation of the probability of an event as a function of several independent variables. *Biometrika* **54**(1–2):167–179.

[4] Hosmer, D.W. & Lemeshow, S. (1989) *Applied Logistic Regression.* Wiley.

[5] Cortez, P. & others. (2009) Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems* **47**:547–553.

[6] Menard, S. (2002) *Applied Logistic Regression Analysis.* Sage Publications, 2nd edition.

[7] Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A. & Rubin, D.B. (2013) *Bayesian Data Analysis.* CRC Press, 3rd edition.

[8] Albert, J.H. & Chib, S. (1993) Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association* **88**(422):669–679.

[9] Genkin, A., Lewis, D.D. & Madigan, D. (2007) Large-scale Bayesian logistic regression for text categorization. *Technometrics.*