Wine Quality Analysis Report

STSCI 3740

Isabella Chen (ic324), Rishika Cherivirala (rrc87), Fiona Huang (xh393), Katie Perlitz (kap256)

1. Introduction

Understanding which attributes most strongly influence wine quality can help winemakers improve their products and optimize them. To assess wine quality, we used the Wine Quality dataset from the UCI Machine Learning Repository. The dataset includes the following predictors: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphases, and type (red or white), along with the response variable quality, which is a score from 0 to 10 assigned by the wine tasters. Our analysis incorporates both regression and classification models to predict the quality and evaluate model performance using metrics like RMSE, AIC, Accuracy, and ROC/AUC. We began our analysis with a linear regression model as a baseline and then explored alternative models, including Lasso Regression, Generalized Additive Models (GAM), K-Nearest Neighbors (KNN), Logistic Regression, and Quadratic Discriminant Analysis (LDA/QDA).

1.1 Data Exploration

Summary Statistics

We began by taking a broad look at each of the 6,497 wines in our dataset (all red and white wines combined). Many predictors cluster tightly around their central values—fixed acidity near 7 g/dm³, volatile acidity around 0.29 g/dm³, and alcohol content close to 10.3% ABV—yet a few features exhibit pronounced right skew behavior: residual sugar (median 3 g/dm³, max 65.8) and sulfur dioxide (*see Figure 1*). Similarly, we also see a comparative trend with features associated with perceived wine quality, with alcohol being notably positively associated, and volatile acidity negatively (*see Figure 2*).

To analyze the distribution of wine quality scores, which will guide our classification modeling, we plotted a histogram (*see Figure 3*). The quality scores exhibit a pronounced peak at 6 and cluster tightly between 5 and 7, with a slight left‑skew as very few wines score below 5 or above 8. This central concentration indicates that most samples are of average quality, with only a handful rated as notably poor or exceptional.

Correlations

We then computed a correlation matrix of all the variables in the dataset and constructed a heatmap accordingly (see *Figure 4*). We noticed that none of the variables have a very high correlation with our response variable “wine quality”, with the highest one being “alcohol” with

a correlation of 0.44. From the heatmap, we observed that most of the predictor pairs in the heatmap not on the main diagonal have a light color, indicating that they have a low inter-predictor relationship. The few exceptions are “total.sulfur.oxide” and “free.sulfur.oxide” having a correlation of 0.72, and “alcohol” and “density” having a correlation of -0.69.

2. Models and Methods

2.1 Pre-Processing

To prepare the data for modeling, we split the dataset into training and test sets using a 70/30 ratio to allow for proper model evaluation. All the models were then trained on the 70% training set and evaluated on the 30% test set. To ensure reproducibility, we also set the random seed to 1 (set.seed(1)) before generating the split. Additionally, all our models were trained on the **combined dataset of both red and white wines**, and we did not conduct separate analyses or build separate models for each wine type. The wine type was instead included as a predictor variable in the models. To account for this categorical variable in non-classification models, we convert it to a dummy variable with *white = 0* and *red = 1*.

2.2 Linear Model

As a baseline, we first fit a linear model using multiple linear regression for all predictors to predict the variable *quality*. We first used a 70/30 training and testing split with the seed set to 1 (to maintain consistency across all methods), and calculated the MSE (mean-squared error) and RMSE (root-mean-squared error) for the model performance. We obtained an RMSE of 0.738247 and an adjusted R-squared of 0.2972. We then performed a 10-fold cross-validation to get an average RMSE for the linear model. This RMSE was **0.7336196**, which served as our baseline for evaluating other methods such as KNN, GAM, and Lasso in other sections.

To quickly evaluate if our linear model could be improved with fewer predictors, we then applied **best subsets selection** using the regsubsets() function. With a maximum of 12 possible predictors, we chose among each of the 12 best subset selection models using cross-validation by calculating each MSE. Our best subset selection indicated predictors like alcohol, volatile acidity, and sulphates being key predictors (*see Figure 5*). However, the model with the minimum MSE using cross-validation was the model with all 12 predictors, so we decided to keep the linear model with all predictors as our baseline.

2.3 Generalized Additive Models

Since our baseline model of linear regression may have missed some non-linear relationships between each predictor and wine quality, we used Generalized Additive Models (GAMs).

Exploratory analysis plotting each predictor with *quality* using loess curves showed several predictors – alcohol, sulphates, chlorides – had slightly curved relationships with the response variable (*see Figure 6*). These nonlinear relationships motivated us to select certain predictors in the GAM to have a basis function of a spline, while others remained as linear predictors.

We evaluated two GAMs.

- **GAM1:** We applied a smoothing spline to all continuous predictors (leaving out "type") to allow for maximum flexibility.
  gam(quality ~ s(fixed.acidity) + s(volatile.acidity) + s(citric.acid) + s(residual.sugar) + s(chlorides) + s(free.sulfur.dioxide) + s(total.sulfur.dioxide) + s(density) + s(sulphates) + s(alcohol) + s(pH))

- **GAM 2:** We applied a smoothing spline only to alcohol, fixed acidity, sulphates, chlorides, and citric acid based on our EDA insights.
  gam(quality ~ fixed.acidity + volatile.acidity + s(citric.acid) + residual.sugar + s(chlorides) + free.sulfur.dioxide + total.sulfur.dioxide + density + s(sulphates) + s(alcohol) + s(pH))

Both models were evaluated using a 70/30 training and testing split with seed set to 1 as specified in Pre-Processing. We then calculated the RMSE for each model and analyzed the summary of each model. GAM1 performed better than GAM2 across all key metrics. The RMSE of **0.722** for GAM1 was lower than GAM2's of **0.735**, and GAM1's AIC of **9806.769** was also lower than GAM2's of **10025.52**. After quickly testing more variations of linear predictors versus using spline smoothers, GAM1 consistently had better predictive performance. This leads us to conclude that adding more flexibility and nonlinear effects to our model helps better capture variability in wine quality. We later compare these GAMs to other non-classification methods.

2.4 KNN Model

Next, we decided to fit a KNN model with the full variables to predict wine quality by classifying new data points based on the similarity to their nearest neighbors in a dataset. Before fitting the model, we first normalized the data using z-score, this step ensures that all predicting variables contribute equally to the distance calculations by rescaling them to have a mean of 0 and a standard deviation of 1. In order to find the optimal value for k, the number of neighbors, we used a 10-fold cross-validation, evaluating values of k ranging from 1 to 20. RMSE was used to select the optimal model using the smallest value. The best-performing model was found with **k = 10**, yielding a **RMSE value of 0.6930709** (*see Figure below*). After fitting the KNN model with this optimal k parameter, we computed a confusion matrix and derived a **test accuracy of approximately 0.5594872**.

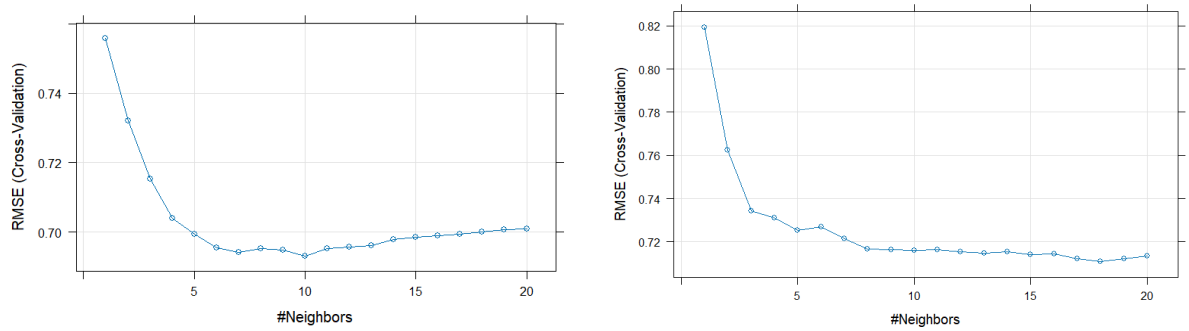


Figure: Number of Neighbors (k) .vs. RMSE
left: model with full data, right: model with PCA-reduced data

To further explore the effect of dimension on the KNN model, we decided to perform a Principal Component Analysis (PCA) before fitting the model. We retained the first 8 principal components, which captured the majority of the variance (around 95%) in the data, and used these components as inputs to the KNN model. As before, we performed a 10-fold cross-validation to select the optimal k value. As a result, the best model with this PCA-reduced data used **k = 18** and a **RMSE value of 0.7109959** (*see Figure above*). The corresponding model **accuracy is 0.5435897**, which is slightly less than the model fitted using simply normalized data, suggesting that reducing the dimensionality of the predictor variables does not achieve a meaningful improvement in the performance of the model. This indicates that the original variable selection retained important structure relevant to prediction, and that reducing dimensionality may lead to information loss in this context.

2.5 Logistic Regression

To classify wine quality based on its chemical properties, we applied both binary logistic regression and multinomial logistic regression models using the full set of predictors.

Method 1: Binary Logistic Regression

We first created a binary outcome variable (quality_binary) by labeling wines with a quality score of **7 or higher as "high" (1)** and those **below 7 as "low" (0),** with 5220 observations in the 0 class and 1277 observations in the 1 class. This transformation allowed us to model the probability that a wine is of high quality using logistic regression. We trained a logistic regression model on the training set using all predictors (excluding the original quality variable, since quality_binary was our new response). The model achieved a **test accuracy of 0.8149**. After obtaining the model summary (*see Figure 7*), we observed that all predictors except **citric.acid** and **type** had statistically significant p-values. To evaluate whether removing these predictors would improve performance or simplify the model, we fit a second logistic regression excluding them. The resulting model yielded a **test accuracy of 0.8128**, which was slightly lower than the full model, suggesting that the exclusion did not lead to meaningful gains in

performance. Overall, the most influential predictors in the binary logistic regression included **alcohol** (positive effect), **volatile acidity** (negative effect), and **sulphates** (positive effect), aligning with findings from earlier exploratory analysis.

Method 2: Multinomial Logistic Regression
To fit the multinomial Logistic Regression, we treated wine quality as a **categorical variable** with classes ranging from **3 to 9**.

```
 3   4    5    6    7   8   9
30 216 2138 2836 1079 193   5
```

Unlike binary logistic regression, this approach allows us to model the probability of each wine belonging to a specific quality level rather than reducing it to a two-class outcome. Each wine was assigned to the class with the highest predicted probability. After fitting the model, we obtained a **test accuracy of 0.5451**. While the accuracy was lower than the binary model, reflecting the increased difficulty of predicting among multiple classes, the model still demonstrated reasonable performance. The confusion matrix (*see Figure 8*) shows that the model often predicted neighboring classes correctly (e.g., predicting a true quality-7 wine as a 6 or 8), which is less problematic than large misclassifications. Unlike the binary model, we also did not perform variable pruning in the multinomial model, as summary outputs for multinomial logistic regression do not provide p-values for individual predictors. Still, consistent with our other models, variables like **alcohol**, **volatile acidity**, and **sulphates** likely contributed strongly to predictions across quality levels, which we can see based on the predictors' high statistical significance.

2.6 LDA and QDA

Similar to logistic regression, we converted wine quality into a binary outcome of “high” (1) when quality is greater than or equal to , versus “low” (0) when quality is less than 7, with the same preprocessing split and seed. Using all predictors (excluding the original quality variable, since quality_binary was our new response), we fitted both Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) models on the training set. Performance was assessed on the held-out test set via confusion matrices and overall metrics.

Test-set performance:

- LDA achieved an overall accuracy of 0.8117, sensitivity of 0.9298, specificity of 0.3290, and balanced accuracy of 0.6294 (*see Figure 9*).
- QDA reached 0.7563 overall accuracy with sensitivity 0.7791, specificity 0.6632, and balanced accuracy of 0.7211 (*see Figure 10*).

LDA yields a higher overall accuracy and strong sensitivity, indicating that it correctly flags the majority class of low-quality wines but tends to overlook the higher-quality class. In contrast, QDA achieves better specificity, making it more adept at correctly identifying high-quality wines, even though its overall accuracy is lower.

To measure the model's stability, we also ran a 10-fold cross-validation on the training set (*see Figure 11*). LDA achieves a higher mean cross-validation accuracy of 0.8206, suggesting that the classes are largely linearly separable. QDA has a cross-validation accuracy of 0.7674 but has a higher Kappa, indicating more agreement beyond chance at some cost to average error. This indicates that most of the separation between the “low” and “high” quality wines can be captured by a linear boundary, shown by LDA’s superior mean accuracy. At the same time, there is enough nonlinear structure in the predictors that a quadratic boundary can use to improve its agreement with true labels beyond chance. In practice, the optimal choice depends on whether the study’s objectives prioritize raw accuracy or balanced detection of wines.

3. Model Comparison

3.1 Model Comparison (Non-classification)

To assess and compare the predictive accuracy of several regression-based approaches for modeling wine quality, we conducted 10-fold cross-validation across five different models: **Linear Regression**, **Generalized Additive Model 1** (GAM1), **Generalized Additive Model 2** (GAM2), **K-Nearest Neighbors** (KNN), and **Lasso Regression**. The evaluation metric used was Root Mean Squared Error (RMSE), averaged across the folds for each model. We used 10-fold cross-validation to test the RMSE across multiple sets of training and testing data, reducing the chance of one model performing better due to overfitting to one random data split.
Our Linear Regression, GAM1, and GAM2 models are outlined in sections 2.2 and 2.3.. For KNN, we used a k value of 10 based on the cross-validation results from section 2.4. For Lasso, we first performed cross-validation to determine the best lambda value, then used this lambda value in the model to generate predictions for each of the 10 iterations in 10-fold cross-validation.

Model <chr>	RMSE <dbl>
Linear	0.7336196
GAM1	0.7149347
GAM2	0.7291629
KNN	0.6903177
Lasso	0.7336021

Figure 3: Root mean-squared-error results for each model after 10-fold cross-validation.

The results of the RMSE for each model show **KNN performing the best**, with an RMSE of **0.690**, which indicates that local similarity may be more effective at predicting wine quality than

more general parametric relationships. This reflects a classic bias-variance tradeoff: KNN has low bias but higher variance, whereas a model like linear regression is more constrained (higher bias, lower variance). In this case, a higher variance of KNN is justified by its improved prediction accuracy. Both Generalized Additive Models performed better than Linear Regression, further supporting that predictors of wine quality include nonlinear effects. Lasso Regression performed almost identically to Linear Regression, showing that regularization and variable selection offered only slight improvement in this case. Overall, KNN appears to be the best choice for reducing RMSE in wine quality prediction, but GAM1 could also be used if a more interpretable model is desired. Thinking of who may use a model like this, having the interpretability in GAM1 may be more useful to winemakers who can improve their wine quality by adjusting specific factors like alcohol or acidity.

3.2 Model Comparison (Classification)

To assess and compare the accuracy of the classification-based approaches, we conducted a 10-fold cross-validation on the training dataset and predicted on the test dataset across four different models: Binary Logistic Regression, Multinomial Logistic Regression, LDA, and QDA. For reference, the logistic models are outlined in section 2.5 and and the LDA/QDA models are outlined in section 2.6. The test-set accuracies were:

- Binary Logistic: 0.8149
- Multinomial Logistic: 0.5451
- LDA: 0.8117
- QDA: 0.7563

The results show that the Binary Logistic Model yields the highest classification accuracy, closely followed by the LDA. This aligns with our previous conclusion of linear separability, where most of our key predictors (e.g., alcohol, volatile acidity) exhibit a roughly linear boundary.

However, it's important to consider the **No Information Rate (NIR)**—the accuracy obtained by always predicting the most frequent class. In our dataset, the most common class for the binary classification models (Binary Logistic, LDA, and QDA) is "low" (quality < 7). The corresponding NIRs were:

- Binary Logistic: 0.8056
- Multinomial Logistic: 0.4415
- LDA: 0.8035
- QDA: 0.8035

This suggests that while the Binary Logistic Model and the LDA Models outperform the NIR slightly, meaning that the models do provide some predictive value beyond random chance, the margins are narrow, and the class imbalance might play a role in increasing the accuracy of the models. In contrast, the Multinomial model's NIR is lower and shows its relatively stronger performance over the random chance baseline, even though its raw accuracy does seem to be quite low compared to the others.

4. Conclusion

In this project, we explored many statistical models to predict wine quality based on its properties. Starting with data exploration, we identified key features like alcohol, volatile acidity, and sulphates as influential across our multiple modeling approaches. These discoveries guided our selection and interpretation of our regression and classification models.

For non-classification methods, K-Nearest Neighbors achieved the lowest RMSE, while models like GAMs may offer a balance between predictive accuracy and interpretability. This is relevant for practical cases of wine quality prediction, such as leading wine-making decisions.

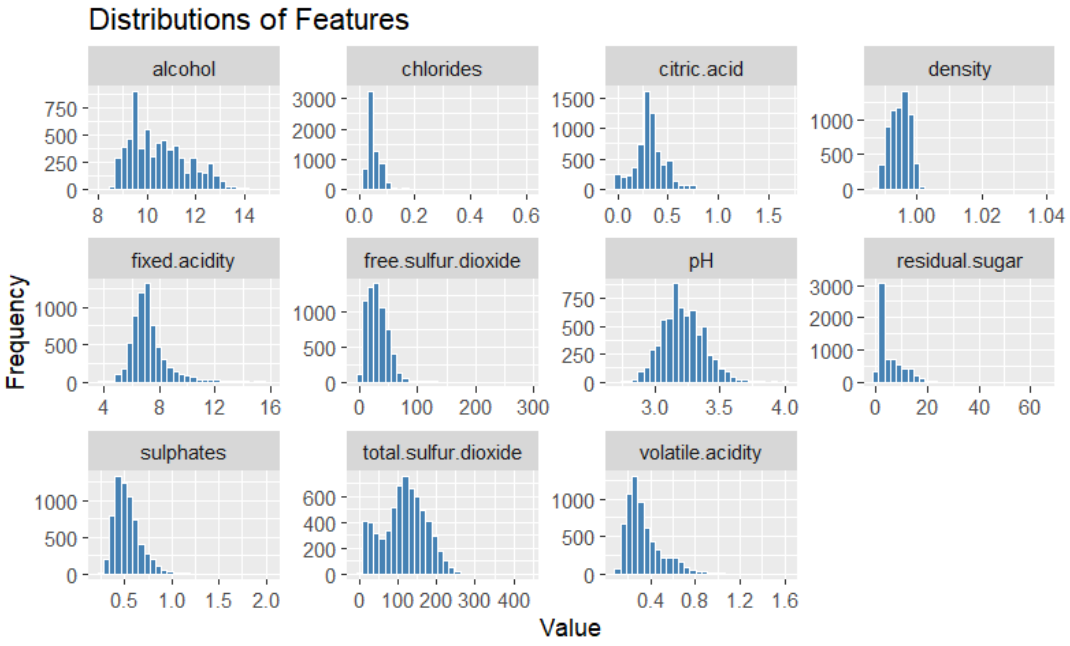
For classification tasks, overall, the binary logistic regression achieved the highest test accuracy rate without taking into account the NIRs. However, when taking the NIR into account, the multinomial logistic regression model achieved the test's highest accuracy with the lowest random chance.

Overall, choosing the best model, both non-classification and classification, might differ depending on the specific constraints and the applications of the model.

5. Appendix

Data set: https://archive.ics.uci.edu/ml/datasets/wine+quality

Github Link for Code: https://github.com/rcherivi/stsci3740final




```
##      type           fixed.acidity    volatile.acidity  citric.acid
##  Length:6497        Min.   : 3.800   Min.   :0.0800   Min.   :0.0000
##  Class :character   1st Qu.: 6.400   1st Qu.:0.2300   1st Qu.:0.2500
##  Mode  :character   Median : 7.000   Median :0.2900   Median :0.3100
##                     Mean   : 7.215   Mean   :0.3397   Mean   :0.3186
##                     3rd Qu.: 7.700   3rd Qu.:0.4000   3rd Qu.:0.3900
##                     Max.   :15.900   Max.   :1.5800   Max.   :1.6600
##  residual.sugar     chlorides       free.sulfur.dioxide total.sulfur.dioxide
##  Min.   : 0.600   Min.   :0.00900   Min.   :  1.00      Min.   :  6.0
##  1st Qu.: 1.800   1st Qu.:0.03800   1st Qu.: 17.00      1st Qu.: 77.0
##  Median : 3.000   Median :0.04700   Median : 29.00      Median :118.0
##  Mean   : 5.443   Mean   :0.05603   Mean   : 30.53      Mean   :115.7
##  3rd Qu.: 8.100   3rd Qu.:0.06500   3rd Qu.: 41.00      3rd Qu.:156.0
##  Max.   :65.800   Max.   :0.61100   Max.   :289.00      Max.   :440.0
##     density             pH          sulphates         alcohol
##  Min.   :0.9871   Min.   :2.720   Min.   :0.2200   Min.   : 8.00
##  1st Qu.:0.9923   1st Qu.:3.110   1st Qu.:0.4300   1st Qu.: 9.50
##  Median :0.9949   Median :3.210   Median :0.5100   Median :10.30
##  Mean   :0.9947   Mean   :3.219   Mean   :0.5313   Mean   :10.49
##  3rd Qu.:0.9970   3rd Qu.:3.320   3rd Qu.:0.6000   3rd Qu.:11.30
##  Max.   :1.0390   Max.   :4.010   Max.   :2.0000   Max.   :14.90
##     quality
##  Min.   :3.000
##  1st Qu.:5.000
##  Median :6.000
##  Mean   :5.818
##  3rd Qu.:6.000
##  Max.   :9.000
```

Figure 1. Distribution of Features and Summary Statistics

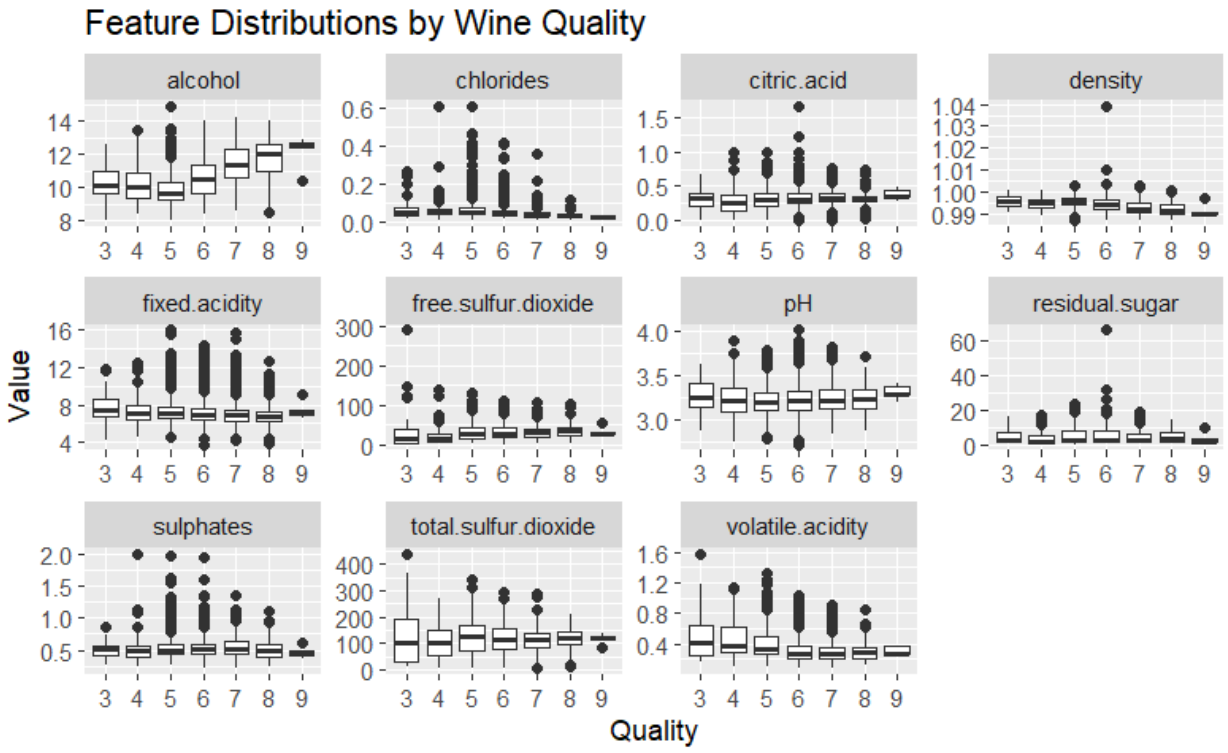


Figure 2. Distribution of Features by Wine Quality

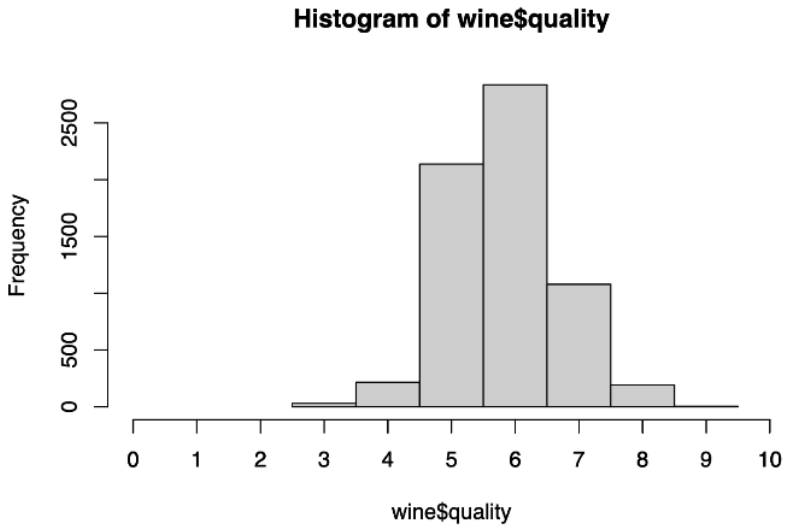


Figure 3. Distribution of Wine Quality

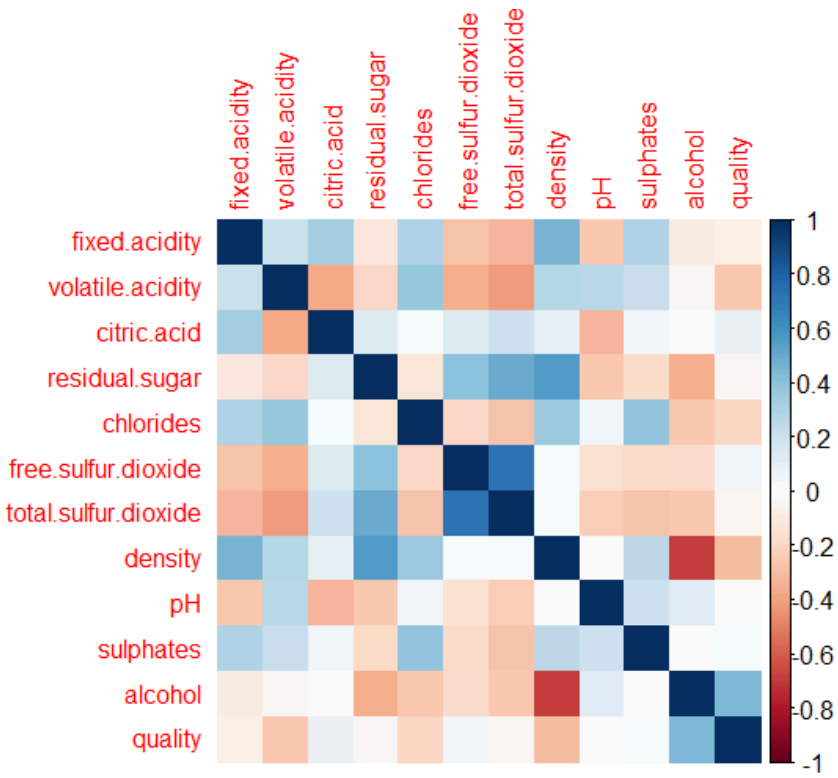


Figure 4. Heatmap of Correlation Matrix

```
Selection Algorithm: exhaustive
          typewhite fixed.acidity volatile.acidity citric.acid
1  ( 1 )  " "       " "           " "              " "
2  ( 1 )  " "       " "           "*"              " "
3  ( 1 )  " "       " "           "*"              " "
4  ( 1 )  " "       " "           "*"              " "
5  ( 1 )  " "       " "           "*"              " "
6  ( 1 )  " "       " "           "*"              " "
7  ( 1 )  " "       " "           "*"              " "
8  ( 1 )  "*"       " "           "*"              " "
9  ( 1 )  "*"       "*"           "*"              " "
10  ( 1 ) "*"       "*"           "*"              " "
11  ( 1 ) "*"       "*"           "*"              " "
12  ( 1 ) "*"       "*"           "*"              "*"
          residual.sugar chlorides free.sulfur.dioxide
1  ( 1 )  " "            " "       " "
2  ( 1 )  " "            " "       " "
3  ( 1 )  " "            " "       " "
4  ( 1 )  "*"            " "       " "
5  ( 1 )  "*"            " "       "*"
6  ( 1 )  "*"            " "       "*"
7  ( 1 )  "*"            "*"       "*"
8  ( 1 )  "*"            " "       "*"
9  ( 1 )  "*"            " "       "*"
10  ( 1 ) "*"            " "       "*"
11  ( 1 ) "*"            "*"       "*"
12  ( 1 ) "*"            "*"       "*"
          total.sulfur.dioxide density pH  sulphates alcohol
1  ( 1 )  " "                  " "     " " " "       "*"
2  ( 1 )  " "                  " "     " " " "       "*"
3  ( 1 )  " "                  " "     " " "*"       "*"
4  ( 1 )  " "                  " "     " " "*"       "*"
5  ( 1 )  "*"                  " "     " " " "       "*"
6  ( 1 )  "*"                  " "     " " "*"       "*"
7  ( 1 )  "*"                  " "     " " "*"       "*"
8  ( 1 )  "*"                  "*"     " " "*"       "*"
9  ( 1 )  " "                  "*"     "*" "*"       "*"
10  ( 1 ) "*"                  "*"     "*" "*"       "*"
11  ( 1 ) "*"                  "*"     "*" "*"       "*"
12  ( 1 ) "*"                  "*"     "*" "*"       "*"
[1] 12
         (Intercept)            typewhite        fixed.acidity
        1.190237e+02        -3.734753e-01         9.607001e-02
    volatile.acidity          citric.acid       residual.sugar
       -1.381747e+00        -1.535991e-02         6.575127e-02
           chlorides  free.sulfur.dioxide total.sulfur.dioxide
       -8.970624e-01         5.661084e-03        -1.625646e-03
             density                   pH            sulphates
       -1.184895e+02         5.766522e-01         6.741088e-01
               alcohol
        2.107583e-01
```

Figure 5: Best subset selection results across 12 predictors. Alcohol is chosen in every model, and as the best singular predictor.

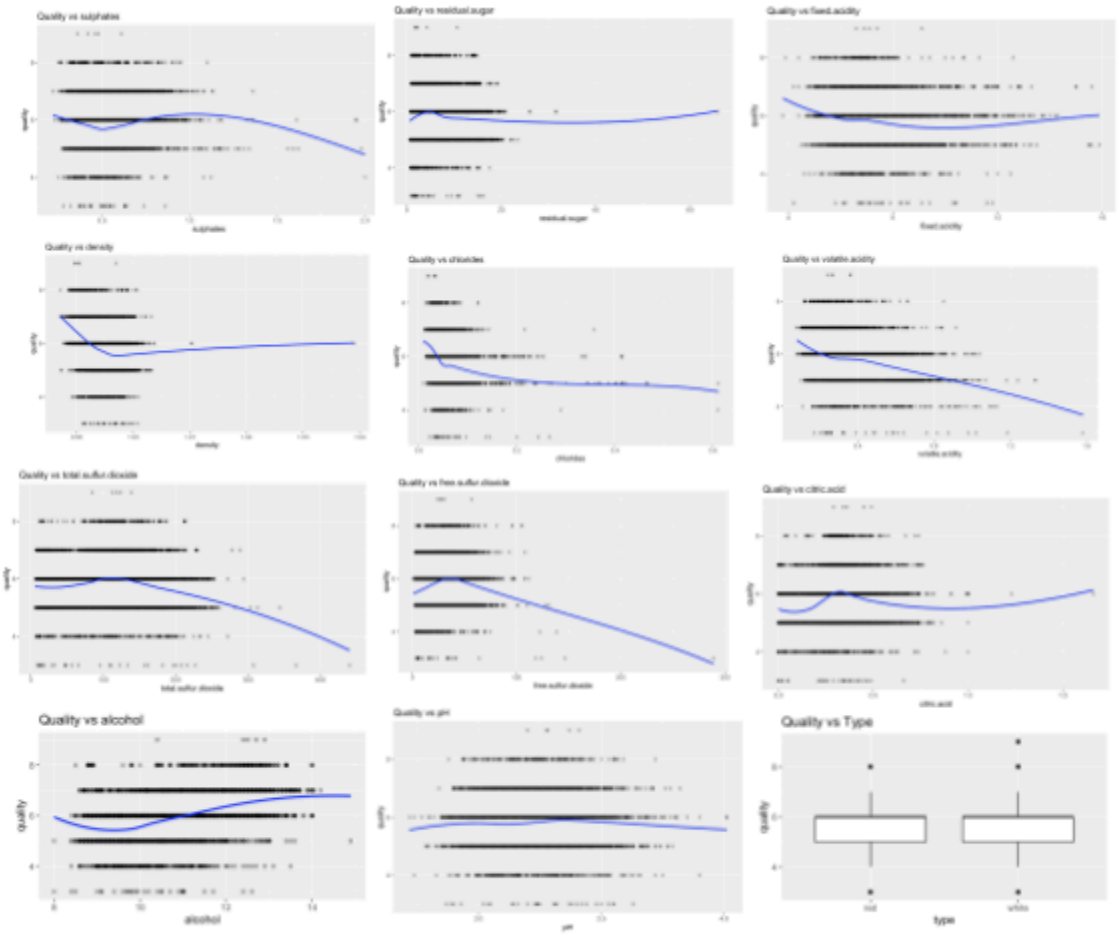


Figure 6: Showing Quality vs Predictor for each variable, with lowess lines.

```
Call:
glm(formula = quality_binary ~ . - quality, family = binomial,
    data = train2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7802  -0.6248  -0.3639  -0.1678   3.0108

Coefficients:
                       Estimate Std. Error z value Pr(>|z|)
(Intercept)           3.291e+02  8.006e+01   4.111 3.94e-05 ***
typewhite            -4.512e-01  3.089e-01  -1.460 0.144182
fixed.acidity         4.660e-01  8.131e-02   5.731 9.98e-09 ***
volatile.acidity     -3.415e+00  4.702e-01  -7.263 3.78e-13 ***
citric.acid          -2.504e-01  4.129e-01  -0.606 0.544203
residual.sugar        1.931e-01  3.183e-02   6.068 1.29e-09 ***
chlorides            -1.057e+01  3.441e+00  -3.071 0.002132 **
free.sulfur.dioxide   1.305e-02  3.633e-03   3.592 0.000328 ***
total.sulfur.dioxide -4.831e-03  1.647e-03  -2.933 0.003352 **
density              -3.502e+02  8.113e+01  -4.316 1.59e-05 ***
pH                    2.576e+00  4.279e-01   6.020 1.74e-09 ***
sulphates             2.193e+00  3.480e-01   6.301 2.97e-10 ***
alcohol               5.474e-01  9.714e-02   5.636 1.75e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 4518.9  on 4546  degrees of freedom
Residual deviance: 3537.4  on 4534  degrees of freedom
AIC: 3563.4

Number of Fisher Scoring iterations: 6

Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1482  272
         1   89  107

               Accuracy : 0.8149
                 95% CI : (0.7969, 0.8319)
    No Information Rate : 0.8056
    P-Value [Acc > NIR] : 0.1583

                  Kappa : 0.2763

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.9433
            Specificity : 0.2823
         Pos Pred Value : 0.8449
         Neg Pred Value : 0.5459
             Prevalence : 0.8056
         Detection Rate : 0.7600
   Detection Prevalence : 0.8995
      Balanced Accuracy : 0.6128

       'Positive' Class : 0
```

Figure 7: Binary Logistic Model Summary and Confusion Matrix

```
# weights:  98 (78 variable)
initial  value 8848.053448
iter  10 value 6112.967244
iter  20 value 5795.162701
iter  30 value 5483.743022
iter  40 value 4942.177481
iter  50 value 4848.170190
iter  60 value 4825.676791
iter  70 value 4814.244712
iter  80 value 4812.501302
iter  90 value 4811.705139
iter 100 value 4810.173610
final  value 4810.173610
stopped after 100 iterations
Confusion Matrix and Statistics

          Reference
Prediction   3   4   5   6   7   8   9
         3   1   0   3   0   0   0   0
         4   0   5   4   0   0   0   0
         5   2  39 393 190  24   6   0
         6   4  22 232 599 228  35   1
         7   0   1   3  71  65  18   2
         8   0   0   0   1   0   0   0
         9   1   0   0   0   0   0   0

Overall Statistics

               Accuracy : 0.5451
                 95% CI : (0.5227, 0.5674)
    No Information Rate : 0.4415
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.2704

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 3 Class: 4 Class: 5 Class: 6 Class: 7  Class: 8  Class: 9
Sensitivity          0.1250000 0.074627   0.6189   0.6957  0.20505 0.0000000 0.0000000
Specificity          0.9984552 0.997876   0.8015   0.5207  0.94182 0.9994712 0.9994864
Pos Pred Value       0.2500000 0.555556   0.6009   0.5343  0.40625 0.0000000 0.0000000
Neg Pred Value       0.9964029 0.968058   0.8133   0.6840  0.85922 0.9697281 0.9984607
Prevalence           0.0041026 0.034359   0.3256   0.4415  0.16256 0.0302564 0.0015385
Detection Rate       0.0005128 0.002564   0.2015   0.3072  0.03333 0.0000000 0.0000000
Detection Prevalence 0.0020513 0.004615   0.3354   0.5749  0.08205 0.0005128 0.0005128
Balanced Accuracy    0.5617276 0.536251   0.7102   0.6082  0.57344 0.4997356 0.4997432
```

Figure 8: Multinomial Logistic Model Summary and Confusion Matrix

```
LDA Model Performance:
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1456  257
         1  110  126

               Accuracy : 0.8117
                 95% CI : (0.7936, 0.8288)
    No Information Rate : 0.8035
    P-Value [Acc > NIR] : 0.1888

                  Kappa : 0.3026

 Mcnemar's Test P-Value : 2.514e-14

            Sensitivity : 0.9298
            Specificity : 0.3290
         Pos Pred Value : 0.8500
         Neg Pred Value : 0.5339
             Prevalence : 0.8035
         Detection Rate : 0.7470
   Detection Prevalence : 0.8789
      Balanced Accuracy : 0.6294
```

Figure 9: Linear Discriminant Analysis Summary and Confusion Matrix

```
QDA Model Performance:
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1220  129
         1  346  254

               Accuracy : 0.7563
                 95% CI : (0.7366, 0.7752)
    No Information Rate : 0.8035
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3643

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.7791
            Specificity : 0.6632
         Pos Pred Value : 0.9044
         Neg Pred Value : 0.4233
             Prevalence : 0.8035
         Detection Rate : 0.6260
   Detection Prevalence : 0.6921
      Balanced Accuracy : 0.7211
```

Figure 10: Quadratic Discriminant Analysis Summary and Confusion Matrix

Model <chr>	CV_Accuracy <dbl>	Test_Accuracy <dbl>
LDA	0.821	0.812
QDA	0.767	0.756

Figure 11: Cross-validation Accuracy and Test Accuracy for LDA and QDA