Project 1 wine quality report

Tingyu Qian

2023-09-14

The dataset is from UCI machine learning database repository https:// archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/. In the red wine dataset, there are 1599 observations, and in the white wine dataset, there are 4898 observations. In each of these two datasets, there are 12 variables, one of them is wine quality (scored between 0 and 10), and the others are chemical attributes (quantitative), which are as follows: Fixed acidity, Volatile acidity, Citric acid, Residual sugar, Chlorides, Free sulfur dioxide, Total sulfur dioxide, Density, PH, Sulphates, and Alcohol. The final goal of this project is to devise and implement a method to determine whether a wine is Red or White given its chemical attributes information.

Part 1

In order to better analyze the data, we need to combine the red wine dataset and the white wine dataset. Therefore, the dataset called all_wine contains 1599 observations of red wine and 4898 observations of white wine. In order to avoid being unable to distinguish which is red wine and which is white wine, we added a column named type. In this column, we use 1 to represent red wine and 0 to represent white wine. The first 10 rows in the dataset all_wine are shown below:

```
#display the first 10 observations
head(all_wine, 10)
```

```
##    fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1            7.4             0.70        0.00            1.9     0.076
## 2            7.8             0.88        0.00            2.6     0.098
## 3            7.8             0.76        0.04            2.3     0.092
## 4           11.2             0.28        0.56            1.9     0.075
## 5            7.4             0.70        0.00            1.9     0.076
## 6            7.4             0.66        0.00            1.8     0.075
## 7            7.9             0.60        0.06            1.6     0.069
## 8            7.3             0.65        0.00            1.2     0.065
## 9            7.8             0.58        0.02            2.0     0.073
## 10           7.5             0.50        0.36            6.1     0.071
##    free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 1                   11                  34  0.9978 3.51      0.56     9.4
## 2                   25                  67  0.9968 3.20      0.68     9.8
## 3                   15                  54  0.9970 3.26      0.65     9.8
## 4                   17                  60  0.9980 3.16      0.58     9.8
## 5                   11                  34  0.9978 3.51      0.56     9.4
## 6                   13                  40  0.9978 3.51      0.56     9.4
## 7                   15                  59  0.9964 3.30      0.46     9.4
## 8                   15                  21  0.9946 3.39      0.47    10.0
## 9                    9                  18  0.9968 3.36      0.57     9.5
## 10                  17                 102  0.9978 3.35      0.80    10.5
```

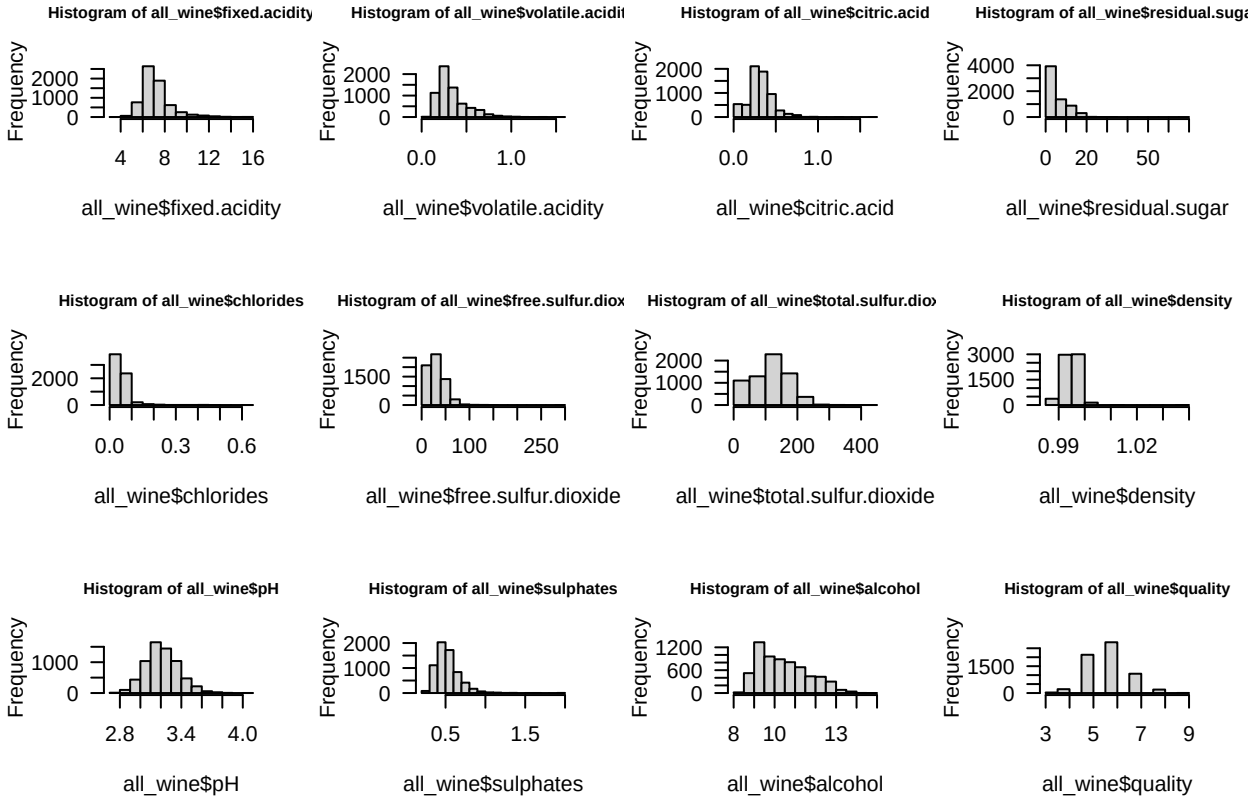
```
##    quality type
## 1        5    1
## 2        5    1
## 3        5    1
## 4        6    1
## 5        5    1
## 6        5    1
## 7        5    1
## 8        7    1
## 9        7    1
## 10       5    1
```

In order to make the prediction results more accurate, we need to check the data first. Check whether the data set contains missing values and extreme values. We need to process the corresponding data if any abnormal data are found. After we checked whether there were missing value, we found that there were no missing values in the dataset all_wine. Now we can check whether the dataset contains extreme values separately.

```
##  fixed.acidity    volatile.acidity  citric.acid     residual.sugar
##  Min.   : 3.800   Min.   :0.0800   Min.   :0.0000   Min.   : 0.600
##  1st Qu.: 6.400   1st Qu.:0.2300   1st Qu.:0.2500   1st Qu.: 1.800
##  Median : 7.000   Median :0.2900   Median :0.3100   Median : 3.000
##  Mean   : 7.215   Mean   :0.3397   Mean   :0.3186   Mean   : 5.443
##  3rd Qu.: 7.700   3rd Qu.:0.4000   3rd Qu.:0.3900   3rd Qu.: 8.100
##  Max.   :15.900   Max.   :1.5800   Max.   :1.6600   Max.   :65.800
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide    density
##  Min.   :0.00900   Min.   :  1.00      Min.   :  6.0        Min.   :0.9871
##  1st Qu.:0.03800   1st Qu.: 17.00      1st Qu.: 77.0        1st Qu.:0.9923
##  Median :0.04700   Median : 29.00      Median :118.0        Median :0.9949
##  Mean   :0.05603   Mean   : 30.53      Mean   :115.7        Mean   :0.9947
##  3rd Qu.:0.06500   3rd Qu.: 41.00      3rd Qu.:156.0        3rd Qu.:0.9970
##  Max.   :0.61100   Max.   :289.00      Max.   :440.0        Max.   :1.0390
##        pH          sulphates         alcohol         quality
##  Min.   :2.720   Min.   :0.2200   Min.   : 8.00   Min.   :3.000
##  1st Qu.:3.110   1st Qu.:0.4300   1st Qu.: 9.50   1st Qu.:5.000
##  Median :3.210   Median :0.5100   Median :10.30   Median :6.000
##  Mean   :3.219   Mean   :0.5313   Mean   :10.49   Mean   :5.818
##  3rd Qu.:3.320   3rd Qu.:0.6000   3rd Qu.:11.30   3rd Qu.:6.000
##  Max.   :4.010   Max.   :2.0000   Max.   :14.90   Max.   :9.000
##       type
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.2461
##  3rd Qu.:0.0000
##  Max.   :1.0000
```

The above data represents the minimum value, first quantile, median, mean, third quantile, and maximum value of each variable in the dataset. The first quantile, often denoted as Q1 or the 25th percentile, is a statistical measure used to divide a dataset into four equal parts. It represents the value below which 25% of the data falls. In other words, it is the 25th percentile of the dataset. The third quantile, often denoted as Q3 or the 75th percentile, is a statistical measure used to divide a dataset into four equal parts. It represents the value below which 75% of the data falls. In other words, it is the 75th percentile of the dataset. From

the values above, we can use Interquartile Range (IQR) method, a statistical technique used for identifying potential outliers in a dataset. It is based on the concept of quartiles, specifically the first quartile (Q1) and the third quartile (Q3). The IQR is the range between Q1 and Q3. Outliers are data points that fall significantly below Q1 or above Q3. We commonly used threshold of 1.5 times the IQR to determine the bounds for potential outliers.



The 12 pictures above are histograms of all variables in the all_wine dataset, and histogram is a graphical representation of the distribution of a dataset. The x-axis of all histograms above represents the range of values of the variable. The y-axis of all histograms above represents the frequency or count of data points that fall into each interval on the x-axis. By looking at the distribution of each variable in the all_wine dataset, we found that although some histograms did not show a normal distribution, some values even more than `1.5*IQR` below Q1 or more than `1.5*IQR` above Q3, (`IQR = Q3-Q1`) which is the occurrence of extreme values. However, since these extreme values are not very different from the normal data, and removing these extreme values will not have a big impact on the final results, we will retain all the data in these datasets here.

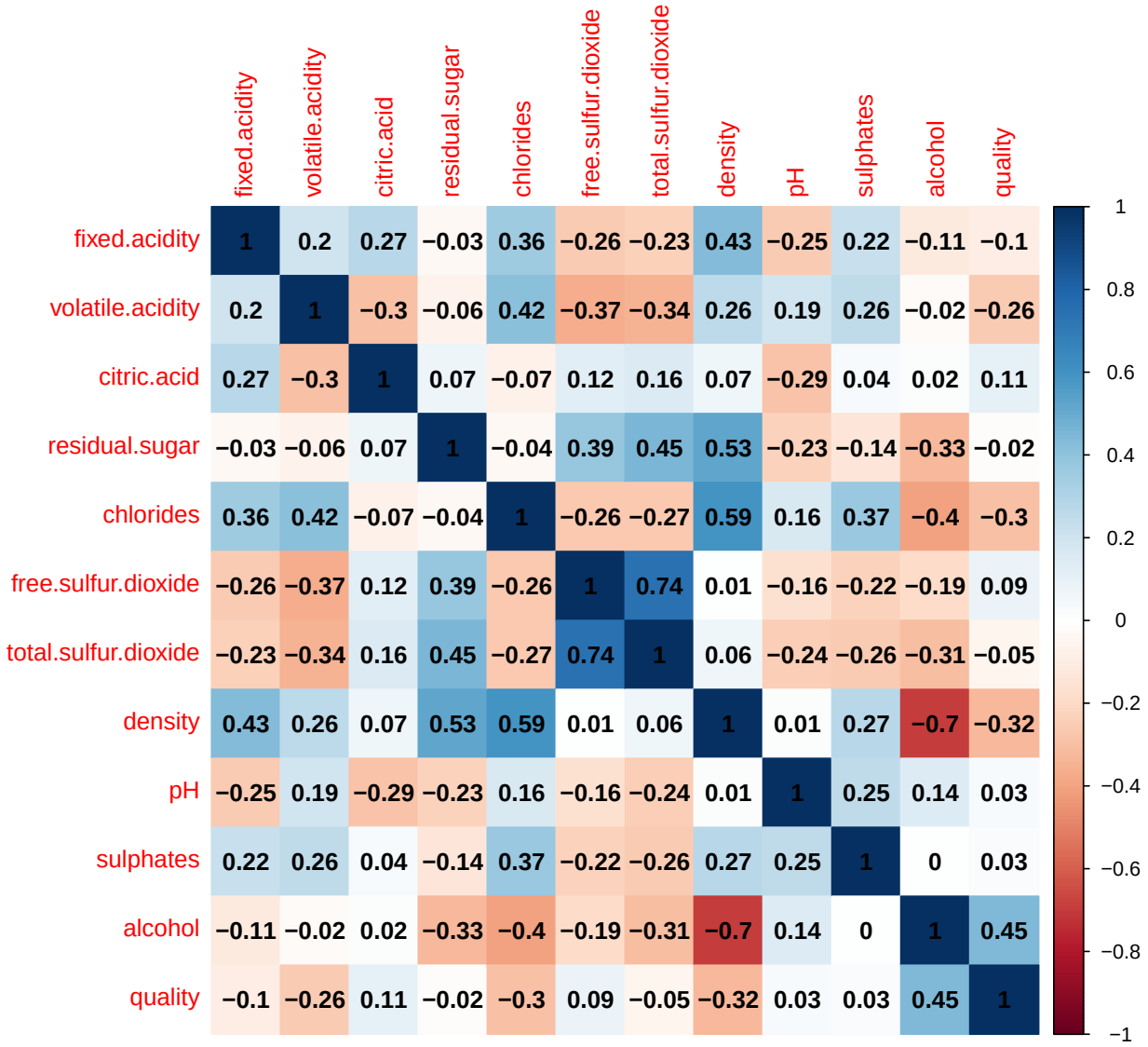
Part 2

(a)

To determine marginal relationships between wine quality and chemical attributes, we chose to use the Spearman rank correlation coefficient. The calculation principle of Spearman rank correlation coefficient is as follows: 1. Rank the values of each variable separately, from lowest to highest, assigning them a rank. If there are ties (i.e., multiple values with the same value), assign the average rank to all tied values. 2. Calculate the difference between the ranks for each pair of data points for both variables. 3. Square these differences and calculate the sum of the squared differences. 4. Use the formula for Spearman's rank

correlation coefficient: $\rho = 1 - \frac{6\sum d^2}{n(n^2-1)}$ Where: ρ is Spearman's rank correlation coefficient. $\sum d^2$ is the sum of the squared rank differences. n is the number of data points.

Spearman's rank correlation coefficient was chosen to look at marginal relationships because it robust to outliers, easy to compute, and because Spearman's rank correlation does not assume that the data follows a specific probability distribution (such as the normal distribution), so it robust to deviations from normality.



	fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide	total.sulfur.dioxide	density	pH	sulphates	alcohol	quality
fixed.acidity	1	0.2	0.27	−0.03	0.36	−0.26	−0.23	0.43	−0.25	0.22	−0.11	−0.1
volatile.acidity	0.2	1	−0.3	−0.06	0.42	−0.37	−0.34	0.26	0.19	0.26	−0.02	−0.26
citric.acid	0.27	−0.3	1	0.07	−0.07	0.12	0.16	0.07	−0.29	0.04	0.02	0.11
residual.sugar	−0.03	−0.06	0.07	1	−0.04	0.39	0.45	0.53	−0.23	−0.14	−0.33	−0.02
chlorides	0.36	0.42	−0.07	−0.04	1	−0.26	−0.27	0.59	0.16	0.37	−0.4	−0.3
free.sulfur.dioxide	−0.26	−0.37	0.12	0.39	−0.26	1	0.74	0.01	−0.16	−0.22	−0.19	0.09
total.sulfur.dioxide	−0.23	−0.34	0.16	0.45	−0.27	0.74	1	0.06	−0.24	−0.26	−0.31	−0.05
density	0.43	0.26	0.07	0.53	0.59	0.01	0.06	1	0.01	0.27	−0.7	−0.32
pH	−0.25	0.19	−0.29	−0.23	0.16	−0.16	−0.24	0.01	1	0.25	0.14	0.03
sulphates	0.22	0.26	0.04	−0.14	0.37	−0.22	−0.26	0.27	0.25	1	0	0.03
alcohol	−0.11	−0.02	0.02	−0.33	−0.4	−0.19	−0.31	−0.7	0.14	0	1	0.45
quality	−0.1	−0.26	0.11	−0.02	−0.3	0.09	−0.05	−0.32	0.03	0.03	0.45	1

The picture above is the correlation matrix, a table that displays the Pearson correlation coefficients between many variables. Each cell in the table shows the correlation between two specific variables. Correlation coefficients measure the strength and direction of a linear relationship between two continuous variables. The value of a correlation coefficient ranges from -1 to 1. A correlation coefficient of 1 indicates a strong positive linear relationship. A correlation coefficient of -1 indicates a strong negative linear relationship. From the correlation matrix above, differnet color represent different levels of correlation. The darker the blue, the stronger the positive correlation between the two variables, and the darker the red, the stronger the negative correlation between the two variables. By looking the correlation matrix, we can find that density

has a strong negative correlation with alcohol, which is -0.7. Total.sulful.dioxide has a strong positive correlation with free.sulfur.dioxide, which is 0.74.

(b)

We can use the wine quality as the response variable, and then use the remaining 11 chemical attributes as explanatory variables and generate a multiple linear regression model. In a multiple linear regression model, we have more than one independent variable (explantory variables) to predict a single dependent (response) variable. The model assumes a linear relationship between the dependent variable and all the independent variables. Equation for multiple linear regression model: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \epsilon$, where $\beta_0, \beta_1, ..., \beta_p$ are coefficients or parameters associated with each independent variable, representing the change in Y for a one-unit change in each corresponding X while holding all other variables constant. $X_1, X_2, \ldots, X_p$ are the independent variables (explantory variables) that influence Y. ϵ is the error term, representing the unexplained or random variation in Y.

```
regression_model_all <- lm(quality ~ fixed.acidity + volatile.acidity +
                           citric.acid + residual.sugar + chlorides +
                           free.sulfur.dioxide + total.sulfur.dioxide +
                           density + pH + sulphates + alcohol, data = all_wine)
summary(regression_model_all)
```

```
##
## Call:
## lm(formula = quality ~ fixed.acidity + volatile.acidity + citric.acid +
##     residual.sugar + chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     density + pH + sulphates + alcohol, data = all_wine)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7569 -0.4597 -0.0412  0.4694  2.9907
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           5.576e+01  1.189e+01   4.688 2.81e-06 ***
## fixed.acidity         6.768e-02  1.557e-02   4.346 1.41e-05 ***
## volatile.acidity     -1.328e+00  7.737e-02 -17.162  < 2e-16 ***
## citric.acid          -1.097e-01  7.962e-02  -1.377    0.168
## residual.sugar        4.356e-02  5.156e-03   8.449  < 2e-16 ***
## chlorides            -4.837e-01  3.327e-01  -1.454    0.146
## free.sulfur.dioxide   5.970e-03  7.511e-04   7.948 2.22e-15 ***
## total.sulfur.dioxide -2.481e-03  2.767e-04  -8.969  < 2e-16 ***
## density              -5.497e+01  1.214e+01  -4.529 6.04e-06 ***
## pH                    4.393e-01  9.037e-02   4.861 1.20e-06 ***
## sulphates             7.683e-01  7.612e-02  10.092  < 2e-16 ***
## alcohol               2.670e-01  1.673e-02  15.963  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7353 on 6485 degrees of freedom
## Multiple R-squared:  0.2921, Adjusted R-squared:  0.2909
## F-statistic: 243.3 on 11 and 6485 DF,  p-value: < 2.2e-16
```

The linear regression model will be

Wine quality = 55.76 + 0.06768*fixed.acidity* - *1.328*volatile.acidity - 0.1097*citric.acid* + *0.04356*residual.sugar - 0.4837*chlorides* + *0.00597*free.sulfur.dioxide - 0.002481*total.sulfur.dioxide* - *50.497*density + 0.4393*pH* + *0.7683*sulphates + 0.267*alcohol

By looking the p-value of each chemical attributes, we can find that variable citric.acid and variable chlorides have bigger p-value (0.168 and 0.146 respectively), and the rest of the variables and the overall F-test p-value are significant ($p<0.05$).

To make the model results more accurate, we can use the Akaike Information Criterion (AIC). The AIC is a tool for comparing the goodness of fit of different statistical models to a given dataset. It is based on the principle of finding a balance between the goodness of fit of the model and the complexity of the model. The formula of AIC is `AIC = -2 * log-likelihood + 2 * k`

log-likelihood is the maximized value of the likelihood function for the model, given the data.

k is the number of parameters in the model.

Our goal is to maximize the `log-likelihood`, and because we have -2 before the `log-likelihood`, so our goal is to minimize the `-2*log-likelihood`. The smaller value of AIC, the better the model fits the data.

AIC has lots of advantages:(1) Model selection:It balances the trade-off between model complexity and goodness of fit, helping us choose the model that best explains the data without overfitting.(2) Generalizability: Models selected using AIC tend to have better generalizability to new, unseen data because they are less likely to be overfitting the training data.

```
AIC_all = step (regression_model_all)
```

```
## Start:  AIC=-3982.79
## quality ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
##     chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     density + pH + sulphates + alcohol
##
##                        Df Sum of Sq    RSS     AIC
## - citric.acid           1     1.026 3507.6 -3982.9
## <none>                              3506.5 -3982.8
## - chlorides             1     1.143 3507.7 -3982.7
## - fixed.acidity         1    10.214 3516.7 -3965.9
## - density               1    11.090 3517.6 -3964.3
## - pH                    1    12.777 3519.3 -3961.2
## - free.sulfur.dioxide   1    34.155 3540.7 -3921.8
## - residual.sugar        1    38.595 3545.1 -3913.7
## - total.sulfur.dioxide  1    43.495 3550.0 -3904.7
## - sulphates             1    55.073 3561.6 -3883.5
## - alcohol               1   137.789 3644.3 -3734.4
## - volatile.acidity      1   159.265 3665.8 -3696.2
##
## Step:  AIC=-3982.89
## quality ~ fixed.acidity + volatile.acidity + residual.sugar +
##     chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     density + pH + sulphates + alcohol
##
##                        Df Sum of Sq    RSS     AIC
## <none>                              3507.6 -3982.9
## - chlorides             1     1.542 3509.1 -3982.0
## - fixed.acidity         1     9.265 3516.8 -3967.7
## - density               1    11.166 3518.7 -3964.2
## - pH                    1    13.364 3520.9 -3960.2
```

```
## - free.sulfur.dioxide   1    34.505 3542.1 -3921.3
## - residual.sugar        1    38.337 3545.9 -3914.3
## - total.sulfur.dioxide  1    47.131 3554.7 -3898.2
## - sulphates             1    54.525 3562.1 -3884.7
## - alcohol               1   136.769 3644.3 -3736.4
## - volatile.acidity      1   175.494 3683.1 -3667.7
```

After we use the AIC, we can see all the printout values, and the started AIC value is -3982.79. The AIC column in the picture represents the how will AIC value change after we drop the corresponding value. From the picture and the results above, we can find that as long as we drop off the variable citric.acid, our AIC value become smaller, which is a good result (the smaller the AIC value, the better the model is). However, if we drop off other variables, our AIC value will increase. So we can only drop citric.acid predictor, and then we are having a small AIC value. We can see the summarized statistic on my final model after the variable selection below:

```
summary(AIC_all)
```

```
##
## Call:
## lm(formula = quality ~ fixed.acidity + volatile.acidity + residual.sugar +
##     chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     density + pH + sulphates + alcohol, data = all_wine)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7408 -0.4614 -0.0380  0.4712  2.9885
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           5.594e+01  1.189e+01   4.703 2.61e-06 ***
## fixed.acidity         6.270e-02  1.515e-02   4.139 3.53e-05 ***
## volatile.acidity     -1.287e+00  7.144e-02 -18.014  < 2e-16 ***
## residual.sugar        4.340e-02  5.155e-03   8.420  < 2e-16 ***
## chlorides            -5.550e-01  3.287e-01  -1.689   0.0913 .
## free.sulfur.dioxide   5.998e-03  7.509e-04   7.988 1.61e-15 ***
## total.sulfur.dioxide -2.546e-03  2.727e-04  -9.336  < 2e-16 ***
## density              -5.515e+01  1.214e+01  -4.544 5.62e-06 ***
## pH                    4.481e-01  9.015e-02   4.971 6.83e-07 ***
## sulphates             7.637e-01  7.606e-02  10.041  < 2e-16 ***
## alcohol               2.649e-01  1.666e-02  15.903  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7354 on 6486 degrees of freedom
## Multiple R-squared:  0.2919, Adjusted R-squared:  0.2908
## F-statistic: 267.4 on 10 and 6486 DF,  p-value: < 2.2e-16
```

By looking the summarized statistic on my final model of all_wine, we can find that 9 variables' (fixed.acidity, volatile.acidity, residual.sugar, free.sulfur.dioxide, total.sulfur.dioxide, density, pH, sulphates, alcohol) p-value are signiftcant, and the variable chlorides's p-value become smaller after we use the Akaike Information Criterion (AIC) method. The overall F test's p-value is significant, too. The new linear regression model becomes:

Wine quality = 55.94 + 0.0627*fixed.acidity* - *1.287*volatile.acidity+ 0.0434*residual.sugar* - *0.555*chlorides + 0.006*free.sulfur.dioxide* - *0.002546*total.sulfur.dioxide - 55.15*density* + *0.4481*pH + 0.7637*sulphates* + *0.249*alcohol

(c)

After we test the relationsip in all_wine dataset to the white_wine dataset and red_wine dataset, we find that the important relationships are not consistent across the wine types.

For the red wine, the started AIC value is -1375.49, and there are 4 variables (density, fixed.acidity, residual.sugar, citric.acid) we need to drop. After we drop these 4 variables, the AIC value becomes -1380.79. By looking the summarized statistic for the red wine find model, we can find that rest of all variables' p-value and over all F test's p-value are significant (P<0.05).

For the white wine, the started AIC value is -2788.44, and there are 3 variables (citric.acid, chlorides, total.sulfur.dioxide) we need to drop. After we drop these 3 variables, the AIC value becomes -2793.63. By looking the summarized statistic for the white wine find model, we can find that rest of all variables' p-value and over all F test's p-value are significant (P<0.05).

Part 3

In order to use chemical attributes information to determine whether a wine is red wine or white wine, we need to use classification in XGBoost. XGBoost (Extreme Gradient Boosting) is a popular machine learning algorithm that is primarily used for classification and regression tasks. It's essential to split the data into a training set and a testing set to evaluate the model's performance. Because we need to use chemical attributes to determine the type of wine, we need to first separate the chemical attributes to be used from the type of wine.

```
all_wine <- rbind(red_wine, white_wine)
cols_to_exclude <- c("fixed.acidity", "volatile.acidity", "citric.acid", "residual.sugar",
                     "chlorides", "free.sulfur.dioxide", "total.sulfur.dioxide",
                     "density", "pH", "sulphates", "alcohol")
X <- all_wine[, names(all_wine) %in% cols_to_exclude]

y <- all_wine$type
```

We divide 70% of the all_wine dataset into the training dataset and 30% into the test dataset.

```
set.seed(123)
trainIndex <- createDataPartition(y, p = 0.7, list = FALSE)
X_train <- X[trainIndex, ]
y_train <- y[trainIndex]
X_test <- X[-trainIndex, ]
y_test <- y[-trainIndex]
```

Next step is to train an XGBoost classifier using the XGBoost function, adjust the parameters like nrounds, objective, and eval_metric.

```
xgb_model <- xgboost(data = as.matrix(X_train),
                     label = y_train,
                     objective = "binary:logistic", # For binary classification
                     eval_metric = "logloss",    # Evaluation metric
                     nrounds = 100) # Number of boosting rounds
```

Objective is binary:logistic, because binary:logistic is a parameter used to specify the objective function for binary classification problems. The logistic regression model is defined by the equation $\mathbf{P}(y = 1|X) = \frac{1}{1+e^{-F(x)}}$. Given an input vector X of features and a binary response variable y (type of the wine, red wine or white wine), the binary logistic regression model can be estimate the probability $\mathbf{P}(y = 1|X)$ that the response variable y is equal to 1 (red wine) based on the input features X. F(x) in the equation ofter expressed as $F(x) = \beta 0 + \beta 1X1 + \beta 2X2 + ... + \beta pXp$, where $\beta 0, \beta 1, ..., \beta p$ are coefficients to be estimated during training.

We choose to use "logloss" in eval_metric parameter. "logloss" is logarithmic loss, which quantifies how well the predicted probabilities match the true class labels. Lower logloss values indicate better model performance. So logarithmic loss use the equation $LogLoss = -(y_i log(p_i) + (1 - y_i) log(1 - p_i)$, where y_i is the binary label (red wine or white white, 0 or 1), and p_i is the predicted probability that the data belongs to 1 (red wine).

nrounds is the number of boosting rounds, which is number of iterations or decision trees that are built during the training process. Each boosting round adds a new decision tree to the ensemble and improves the model's performance. However, it is not that more is better, because if there are too many, it will be overfitting. After training the model, we can make a prediction on the test data and then evaluate the performance.

```
y_pred_probs <- predict(xgb_model, newdata = as.matrix(X_test), type = "prob")
y_pred <- ifelse(y_pred_probs > 0.5, 1, 0)
conf_matrix <- table(y_pred, y_test)
conf_matrix
```

```
##       y_test
## y_pred    0    1
##      0 1463    8
##      1    2  476
```

Above is confusion matrix, also known as an error matrix, a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. The confusion matrix is typically organized in a 2x2 table. The first column is Actual positive and Actual negative, which is 0 and 1, stands for actual white wine and actual red wine. The first row is Predicted positve and Predicted Negative, which is 0 and 1, stands for predicted white wine and predicted red wine. Therefore, in the 2x2 table above, there are 1463 data predict correctly for white wine, and 2 predict wrongly for white wine (actual white wine, but predict to red wine). There are 476 data predict correctly for red wine, and 8 predict wrongly for red wine (actual red wine, but predict to white wine).

```
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
accuracy
```

```
## [1] 0.9948692
```

In order to know the accuracy of the prediction, we can use $\frac{right\ predict\ of\ red\ wine + right\ predict\ of\ white\ wine}{all\ red\ wine + all\ white\ wine}$, and the result is 0.9948692.

There are some limitations with XGBoost. Firstly, XGBoost can be prone to overfitting. XGBoost uses decision trees as weak learners. If the trees are allowed to grow too deep during training, they can capture noise in the data and overfit. In order to mitigate this issue, I choose to use nrounds = 100. Secondly, XGBoost has limited interpretability. XGBoost is a machine learning technique that aggregates the output of numerous decision trees to generate predictions. The amalgamation of these trees can create a remarkably intricate model, making it challenging to discern the specific impact of each individual tree on the ultimate prediction. XGBoost is often considered a "black-box" model due to its absence of clarity regarding the mechanisms underlying its predictions. It doesn't furnish clear-cut rules or equations that elucidate how it reaches its decision-making outcomes.