NORTH CAROLINA STATE UNIVERSITY
DEPARTMENT OF STATISTICS

ST 563: INTRODUCTION TO STATISTICAL LEARNING
Fall 2020

Final Project: Analysis of Wine Quality Data

Group Members: Arjav, Jixin, Yulun, John McDonald

1. Introduction

Our team has been assigned the task of working for a vineyard to identify predictors that impact the perceived quality of red wine from the Vinho Verde region of Portugal. The vineyard has provided a dataset with twelve columns. Eleven of the variables are chemical properties presented as continuous variables, and only the response Quality is categorical. The quality of wine is measured by three different taste testers providing a score, and then taking the median rating. The model built in this paper will relate chemical properties of wine to provide objective quality assessment of taste.

2. Methods of Analysis

In the methods of analysis, we will assess the quality of wine using both regression and classification analysis. The objective is to compare performance of different models for wine quality prediction. An initial attempt was made to compare eight different models. Some of the results are included in the appendix. However, it was decided that quality comparison of a few models with emphasis on understanding the data and proper data preparation would provide better results. For regression analysis, we focus on providing an interpretable model with an estimated regression equation. For classification analysis, we focus primarily on accuracy.

2.1 Theory of Methodologies Used

The methodology for comparison of model performance follows a **two stage comparison**. To do this, it is required to use a training and validation set for cross validation, and then a testing set for overall model performance. We first train each model using the training data. Hyperparameter selection is then evaluated using MSE or accuracy on the validation set. Once the optimal hyperparameters are chosen, the final model is retrained using both validation and training data. Each final model is then evaluated using MSE or accuracy on the testing set.

Linear regression is a basic and commonly used type of predictive analysis. Here, We use multiple linear regression, a technique that contains multiple predictor variables.

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \epsilon$$

where,
y_i =dependent variable
x_i = explanatory variables
β_0 =y-intercept (constant term)
β_p = slope coefficients for each explanatory variable
ϵ = the model's error term (also known as the residuals)

Multiple regressions are based on the assumption that there is a linear relationship between both the dependent and independent variables. We look at the MSE , R^2 value and the adjusted R^2 for testing the accuracy of the model.

Polynomial Regression is a form of linear regression in which the relationship between the predictor variables and the dependent variable y is modelled as an n^{th} degree polynomial.Since it is the predictor variable(x) that is squared and not the beta coefficients it still classifies as a linear model. Polynomial regression is used after examining the relationship between the different variables and our dependent variable. It allows us to introduce a curve in our linear (line) regression.

As with least squares, ridge and **lasso regression** seeks coefficient estimates that fit the data well, by making the RSS small. Ridge and lasso regression will include all p predictors in the final model. The lasso and ridge regression have similar formulations for the penalty, where the one for lasso is $\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}|\beta_j| = RSS + \lambda\sum_{j=1}^{p}|\beta_j|$, where the only difference is that the $|\beta_{j^2}|$ term in the ridge regression penalty has been replaced by $|\beta_j|$. λ is a tuning parameter, and λ ≥ 0, which will be determined separately. The tuning parameter λ serves to control the relative impact of these two terms on the regression coefficient estimates. When λ = 0, the penalty term has no effect, and lasso and ridge regression will produce the least squares estimates. As λ→∞, the impact of the shrinkage penalty grows for both ridge and lasso regression, and the ridge regression coefficient estimates will approach zero while the coefficient estimates of lasso regression will be exactly equal to zero. Hence, much like best subset selection, the lasso performs variable selection. As a result, models generated from the lasso are generally much easier to interpret than those produced by ridge regression.

We require a method to determine which of the models under consideration is best, implementing ridge regression and the lasso requires a method for selecting a value for the tuning parameter λ. Cross-validation provides a simple way to tackle this problem. We choose a grid of λ values, and compute the cross-validation error for each value of λ. We then select the tuning parameter value for which the cross-validation error is smallest. Finally, the model is refitted using all of the available observations and the selected value of the tuning parameter.

A new variable, rating, is created that has binary values for wine quality > 5 = 1 and wine quality < 5 = 0. For these two classes we use **logistic regression** for classification. The logistic regression takes the form:

$$p(X) = e^{\beta_0 + + \beta_1 X} / (1 + e^{\beta_0 + + \beta_1 X})$$

We note the relationship of beta for coefficients to x for our variables. However, the relationship between p(X) and X is not linear for our predictors, as it is related to this log odds equation. When analyzing the beta terms, a negative coefficient will continue to associate with a negative correlation in prediction of our classification. For multiple class logistic classification, discriminant analysis tends to be better at performance. However for our purposes, we have two classes, and logistic regression is chosen as a benchmark performance model.

We usually use **decision** trees to predict the classification, because it is intuitive and easy to explain and interpret, with a nice graphical display of decision-making processes like a checklist. However, they don't have the best accuracy, they are hard to train and unstable in changes in the data. Therefore, by averaging multiple decision trees, **random forest** is used to reduce the bias, variance and correlation of the classification-tree prediction. The algorithm creates several random subsets of the original data (or training set) with randomly chosen predictors, and calculates the classification trees, then the final result is the average of all trees. Due to the random process of splitting the data and creating many trees(or a forest), we get the name random forest.

2.2 Describing The Data

To make better decisions regarding our model selection, we first focus on understanding our data set. We start by reading the dimension of our data and taking a look at the first few rows of our set. We notice that the table includes 1599 observations and 12 variables, where 'quality' is the response. We primarily notice that the data is not short and fat - there are many observations compared to variables, so risks of overfitting in regression are low.

```
> head(wine)
  fixed.acidity volatile.acidity citric.acid residual.sugar chlorides free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol quality
1           7.4             0.70        0.00            1.9     0.076                 11                  34  0.9978 3.51      0.56     9.4       5
2           7.8             0.88        0.00            2.6     0.098                 25                  67  0.9968 3.20      0.68     9.8       5
3           7.8             0.76        0.04            2.3     0.092                 15                  54  0.9970 3.26      0.65     9.8       5
4          11.2             0.28        0.56            1.9     0.075                 17                  60  0.9980 3.16      0.58     9.8       6
5           7.4             0.70        0.00            1.9     0.076                 11                  34  0.9978 3.51      0.56     9.4       5
6           7.4             0.66        0.00            1.8     0.075                 13                  40  0.9978 3.51      0.56     9.4       5
> dim(wine)
[1] 1599   12
```

Figure 1: R output for dimension and first six rows of our dataset

To gain an in-depth understanding of our predictor columns, we choose to plot the histograms. What we can see from the initial analysis of the variables is that some of the predictors are right (positively) skewed. In order to address this issue we standardize the predictors, as detailed in the data preprocessing section.

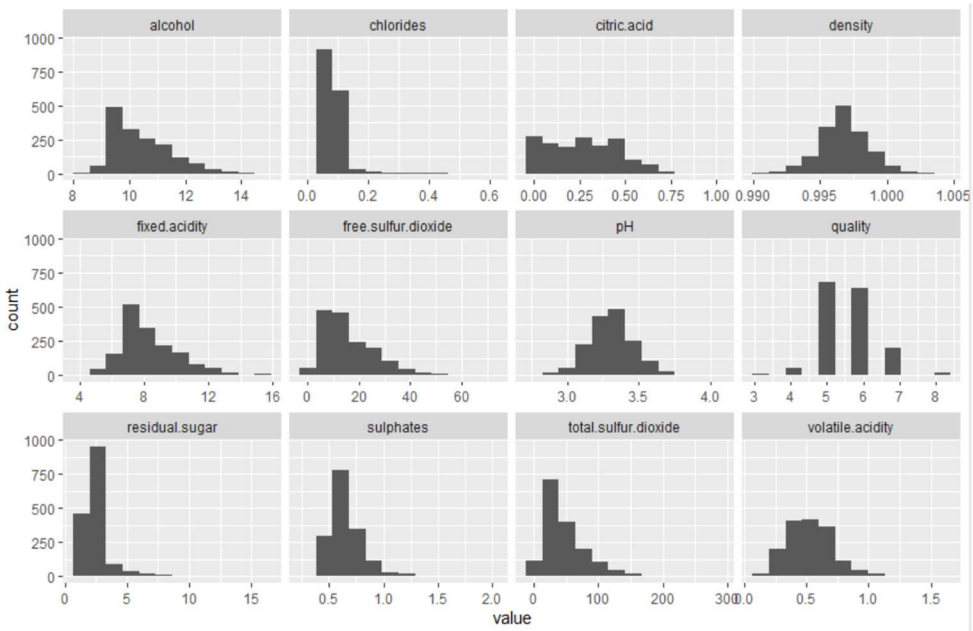


Figure 2: Histograms showing distribution of eleven predictors and categorical response variable 'Quality'

Based on the correlation heat map in figure 3, we can infer from the following variables are highly correlated relationships for the following:

- Free sulphur dioxide and Total sulphur dioxide
- Fixed acidity with Density
- Fixed Acidity with citric acid.

Because we are only interested correlation with quality, we take note of the following relationships:

- Quality moderately increases with alcohol
- Quality moderately decreases with volatile.acidity
- Citric.acid, sulphates, ph, and total.sulfur.dioxide show weak correlation

Note that the significance of these correlations is not yet known. Nevertheless we draw the following conclusions from initial analysis of data:

1. Wine Quality is positively correlated with alcohol and negatively correlated to volatile acidity.
2. pH is negatively correlated with acidity since lower the value in the pH scale means higher acidity
3. We also observe some relationship between density and other variables such as alcohol, residual sugar, citric acid & fixed acidity

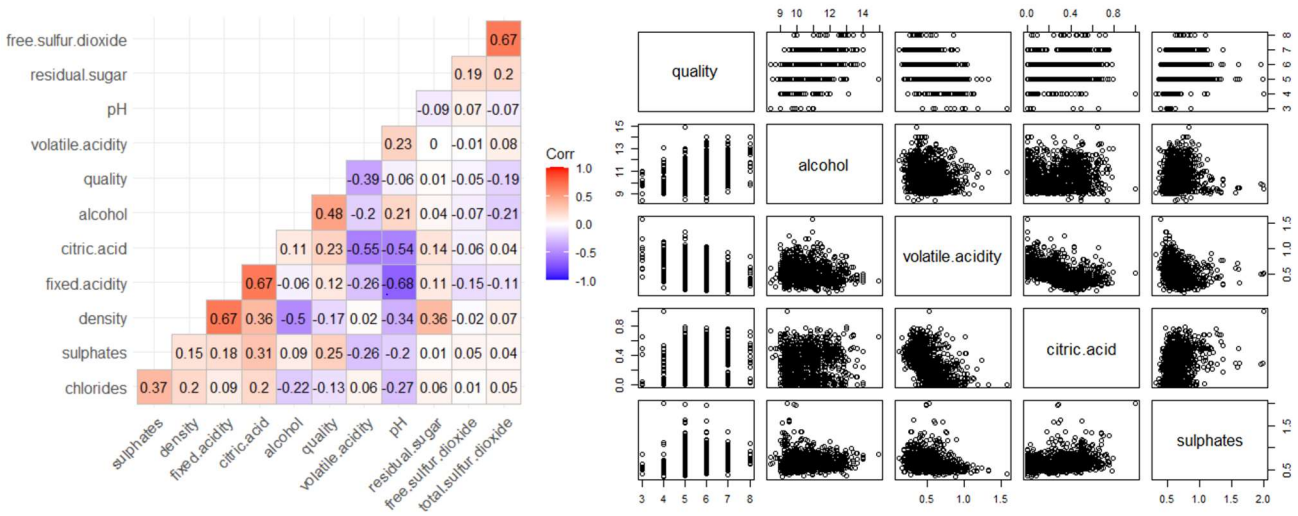


Figure 3: Correlation of all variables on left, with high (above 0.20) pairs correlations to quality shown on right.

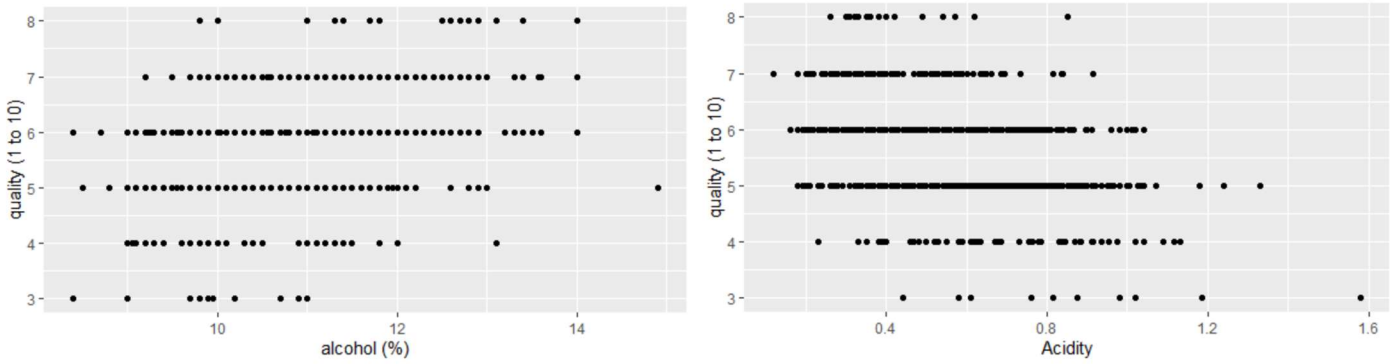


Figure 4: Enlarged bivariate plots for highly correlated predictors to quality response.

Bivariate plots are shown in the pairs plot in figure 3. However, to get a better detail at two potentially significant correlationships, we plot the bivariates in figure 4. We observe that the Alcohol % tends to be higher in the wines with higher quality. The acidity seems to be higher in low quality wines. The plots also show us predictor spread across quality levels. While the correlation in these two predictors is apparent, we can assume that from our dataset, most of the predictors show weak correlation, and only a few are moderately strong.

2.3 Data Preprocessing

Here we detail the data preprocessing steps as required from observations in previous analysis. Note from the histogram spreads we notice that it is important to standardize and center the predictors as skew can impact model performance. Here we use the scale function to perform following transformation:

Variable = Variable - Mean(col) / Sd(col)

```
> head(wine)
  fixed.acidity volatile.acidity citric.acid residual.sugar   chlorides free.sulfur.dioxide total.sulfur.dioxide    density         pH   sulphates    alcohol quality
1    -0.5281944        0.9615758   -1.391037    -0.45307667 -0.24363047         -0.46604672           -0.3790141 0.55809987  1.2882399 -0.57902538 -0.9599458       5
2    -0.2984541        1.9668271   -1.391037     0.04340257  0.22380518          0.87236532            0.6241680 0.02825193 -0.7197081  0.12891007 -0.5845942       5
3    -0.2984541        1.2966596   -1.185699    -0.16937425  0.09632273         -0.08364328            0.2289750 0.13422152 -0.3310730 -0.04807379 -0.5845942       5
4     1.6543385       -1.3840105    1.483689    -0.45307667 -0.26487754          0.10755844            0.4113718 0.66406945 -0.9787982 -0.46103614 -0.5845942       6
5    -0.5281944        0.9615758   -1.391037    -0.45307667 -0.24363047         -0.46604672           -0.3790141 0.55809987  1.2882399 -0.57902538 -0.9599458       5
6    -0.5281944        0.7381867   -1.391037    -0.52400227 -0.26487754         -0.27484500           -0.1966174 0.55809987  1.2882399 -0.57902538 -0.9599458       5
```

Figure 5: R output after standardizing and centering predictors using scale() function

Handling the outliers in our dataset is done carefully. Removing the outliers can improve model performance significantly. However this can only be justified when we are aware of erroneous data collection practices, which is not the case here. Instead, it is recommended to remove leverage points whenever possible, as they can significantly degrade model performance. To do this, we calculate the cook distance of all predictor variables, and use a high cut-off of n/6 to make sure we are only removing high impact outliers. As shown in the plots below, we can see particularly significant improvement at the ends of our regression line.

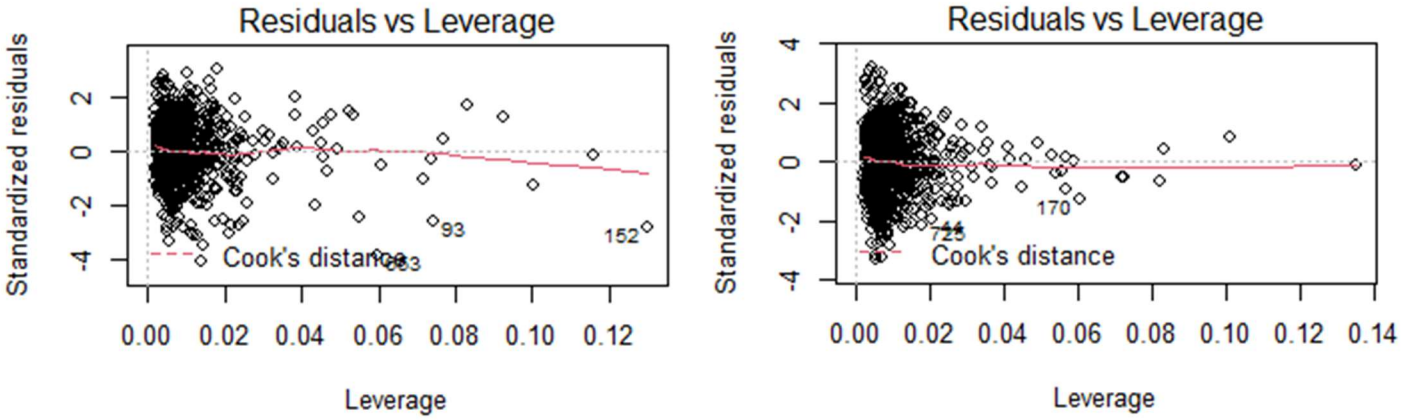


Figure 6: Residual Plot before and after removing leverage points from dataset

For classification models, we collapse the quality column to a new column named ‘rating’. The rating column includes values of ‘low’ if the quality is below 5, and ‘high’ if above five. We then check the class imbalance and see that 53.5% of ratings are ‘high’. This serves as the minimal baseline for classification models.

```
> head(wine)
  fixed.acidity volatile.acidity citric.acid residual.sugar chlorides free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol rating
1           7.4             0.70        0.00            1.9     0.076                  11                   34  0.9978 3.51      0.56     9.4    low
2           7.8             0.88        0.00            2.6     0.098                  25                   67  0.9968 3.20      0.68     9.8    low
3           7.8             0.76        0.04            2.3     0.092                  15                   54  0.9970 3.26      0.65     9.8    low
4          11.2             0.28        0.56            1.9     0.075                  17                   60  0.9980 3.16      0.58     9.8   high
5           7.4             0.70        0.00            1.9     0.076                  11                   34  0.9978 3.51      0.56     9.4    low
6           7.4             0.66        0.00            1.8     0.075                  13                   40  0.9978 3.51      0.56     9.4    low
> # We see 53.47% accuracy as baseline
> prop.table(table(wine$rating))

     high       low
0.5347092 0.4652908
```

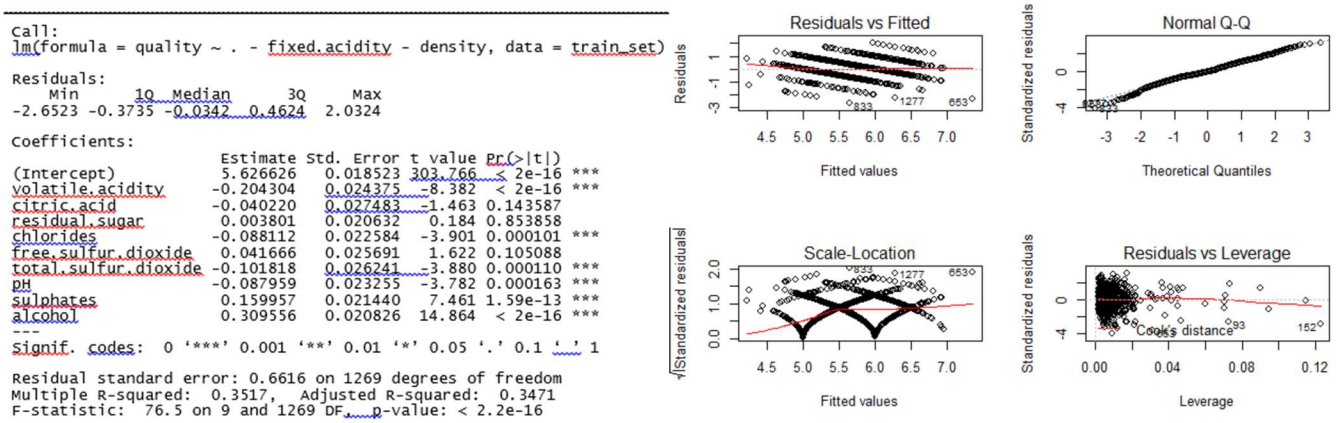
Figure 7: Creating a new column for classification, and class imbalance table.

Final practices include removing null values. It is found that there are no missing values in our original dataset. We also split the wine dataset using an 80/20 train test split. The training dataset is further divided into a validation and training set when implementing cross validation in model analysis.

2.4 Regression analysis

2.4.1 Linear Regression

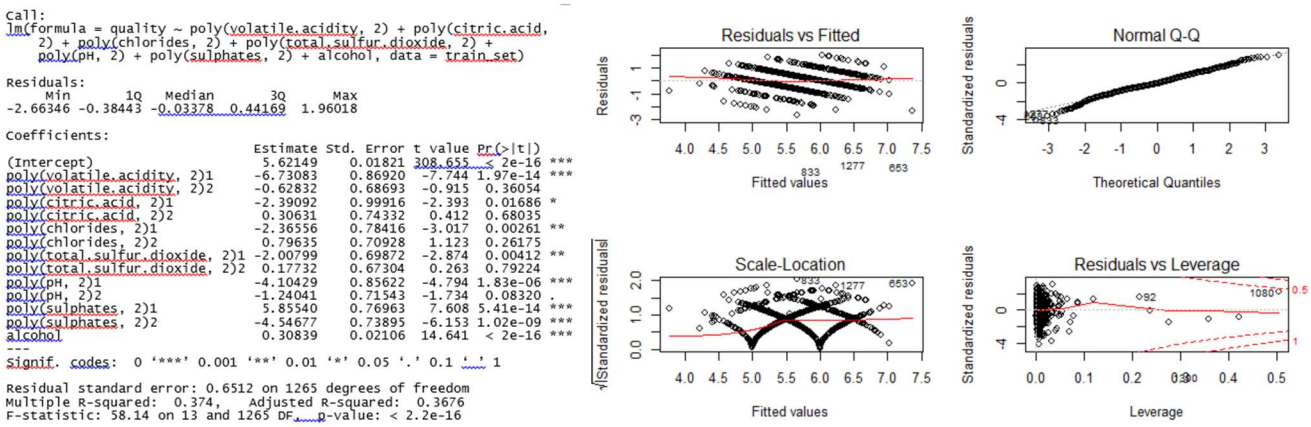
We fit a model with all the predictor variables, However what we see is a lot of the variables are not significant. We then check the VIF factor for all of these variables. We see that the VIF is high for fixed.acidity and density and hence we remove these variables. Variance Inflation Factor (VIF) detects multicollinearity in the regression analysis. Presence of multicollinearity adversely affects our regression result. As a general rule of thumb we consider VIF > 5 as highly correlated.



```
Call:
lm(formula = quality ~ . - fixed.acidity - density, data = train_set)

Residuals:
    Min      1Q  Median      3Q     Max
-2.6523 -0.3735 -0.0342  0.4624  2.0324

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)           5.626626   0.018523 303.766  < 2e-16 ***
volatile.acidity     -0.204304   0.024375  -8.382  < 2e-16 ***
citric.acid          -0.040220   0.027483  -1.463 0.143587
residual.sugar        0.003801   0.020632   0.184 0.853858
chlorides            -0.088112   0.022584  -3.901 0.000101 ***
free.sulfur.dioxide   0.041666   0.025691   1.622 0.105088
total.sulfur.dioxide -0.101818   0.026241  -3.880 0.000110 ***
pH                   -0.087959   0.023255  -3.782 0.000163 ***
sulphates             0.159957   0.021440   7.461 1.59e-13 ***
alcohol               0.309556   0.020826  14.864  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6616 on 1269 degrees of freedom
Multiple R-squared:  0.3517,   Adjusted R-squared:  0.3471
F-statistic:  76.5 on 9 and 1269 DF,  p-value: < 2.2e-16
```


Figure 8: Summary of fit for linear regression with two variables removed from predictors

We notice that on the linear regression model, six predictors are significant: volatile.acidity, chlorides, total.sulfur.dioxide, pH ,sulphates, and alcohol. While the model itself is highly significant, provided the low p-value and high F-statistic, it only accounts for 34.07% of correlation in the response, as described by the adjuster R-squared value. We check the plots for common problems with linear regression. The main issues we notice is an inconsistent magnitude of residuals for our fitted values in the scale-location plot, and a couple leverage points in the Residuals vs Leverage plot. While the issues can be further investigated, we keep the model as is to use as a benchmark for other regression models. In the correlation analysis scatter plot we see that sulphates do not have a linear relationship. Hence, we conclude that there might be a polynomial relationship between sulphates and the dependent variable.



```
Call:
lm(formula = quality ~ poly(volatile.acidity, 2) + poly(citric.acid,
    2) + poly(chlorides, 2) + poly(total.sulfur.dioxide, 2) +
    poly(pH, 2) + poly(sulphates, 2) + alcohol, data = train_set)

Residuals:
    Min      1Q  Median      3Q     Max
-2.66346 -0.38443 -0.03378  0.44169  1.96018

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                      5.62149    0.01821 308.655  < 2e-16 ***
poly(volatile.acidity, 2)1      -6.73083    0.86920  -7.744 1.97e-14 ***
poly(volatile.acidity, 2)2      -0.62832    0.68693  -0.915  0.36054
poly(citric.acid, 2)1           -2.39092    0.99916  -2.393  0.01686 *
poly(citric.acid, 2)2            0.30631    0.74332   0.412  0.68035
poly(chlorides, 2)1             -2.36556    0.78416  -3.017  0.00261 **
poly(chlorides, 2)2              0.79635    0.70928   1.123  0.26175
poly(total.sulfur.dioxide, 2)1 -2.00799    0.69872  -2.874  0.00412 **
poly(total.sulfur.dioxide, 2)2  0.17732    0.67304   0.263  0.79224
poly(pH, 2)1                    -4.10429    0.85622  -4.794 1.83e-06 ***
poly(pH, 2)2                    -1.24041    0.71543  -1.734  0.08320 .
poly(sulphates, 2)1              5.85540    0.76963   7.608 5.41e-14 ***
poly(sulphates, 2)2             -4.54677    0.73895  -6.153 1.02e-09 ***
alcohol                          0.30839    0.02106  14.641  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6512 on 1265 degrees of freedom
Multiple R-squared:  0.374,    Adjusted R-squared:  0.3676
F-statistic: 58.14 on 13 and 1265 DF,  p-value: < 2.2e-16
```


Figure 9: Model summary and model plots for polynomial regression

We can see that the R^2 has increased from our initial multiple regression fit. We can also see that our assumption about sulphates having a polynomial relationship was correct since it is significant, as well as shown in the residuals vs leverage plot.

2.4.2 Lasso Regression

Cross-validation provides a simple way to tackle the tuning parameter λ. We set a grid of λ values first, and compute the cross-validation error for each value of λ. Here we get log10(λ) with its mean squared error. We get the best value of λ =0.0005951615 to let the cross-validation error be smallest. Then we fit the Lasso model with this tuning parameter λ and the train set and the test mean square is 0.3281441. The adjust R square of lasso model is 37.64%, which explains the model more compared with the linear regression model.

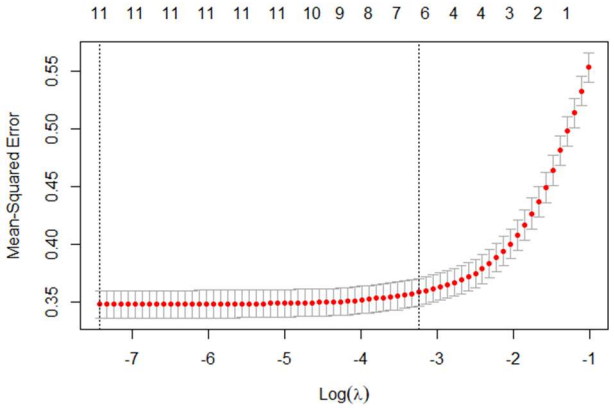


Figure 10: tuning parameter λ with its cross-validation error

2.5 Classification analysis

2.5.1 Logistic Regression

For the logistic regression we train the model based on the training set with cross validation. We check to find the optimal probability threshold for classification between the two classes. As expected, a cutoff of .5 gives the best MSE results, so we use this to train our final model. The ROC curve performs well with AUC = .83.

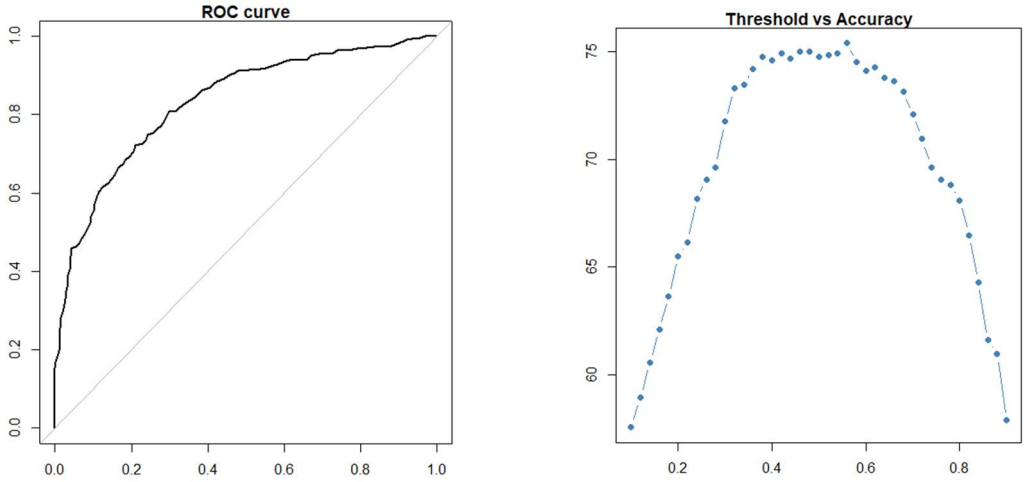


Figure 11: Left plot shows ROC curve. Right plot shows overall accuracy as the threshold is changed.

We run the confusion matrix and model summary on our training set to identify useful metrics. Notably, we see that the accuracy is .74, with 8 significant predictors. The Null deviance is high, and potential further improvements would include removing insignificant predictors from the model. Nevertheless, we get a training error of .26, sensitivity of .72, specificity of .77, which we aim to improve using tree based models.

2.3.4 Classification Tree and Random Forest

We take an overview of the decision tree and the most important predictors(by *gbm()* function). Cross validation gives a decision tree with 7 terminal nodes. We notice that alcohol and sulfates are the most important predictors. The related classification tree is shown below, along with relative influence.

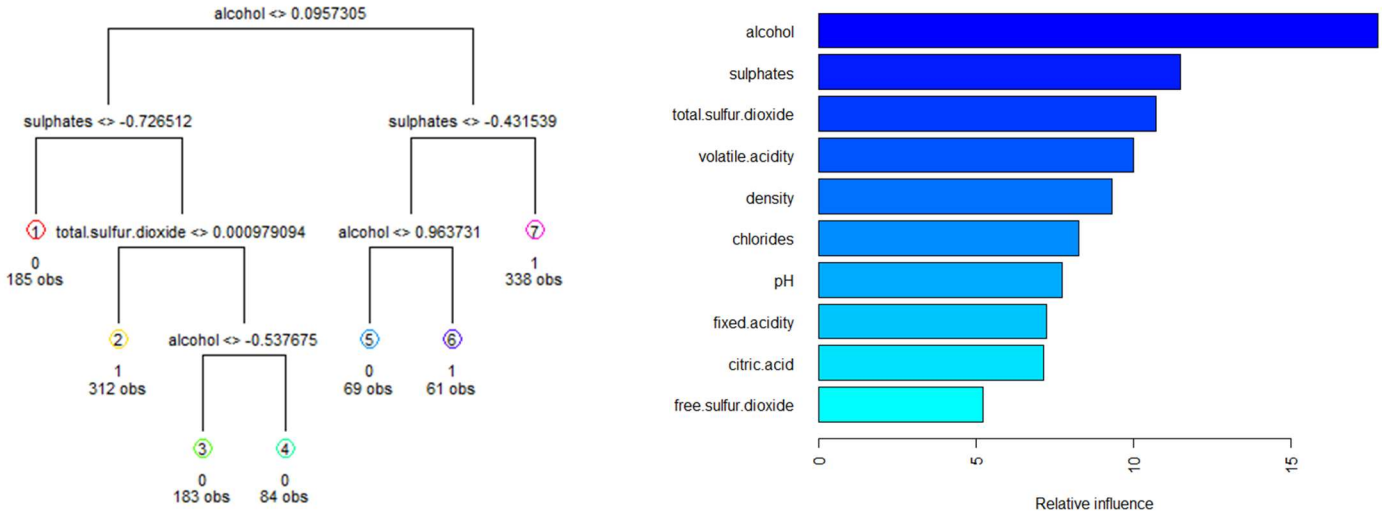


Figure 12: Decision tree with 7 terminal nodes(left); the order of the most important predictors (right)

To improve performance, we extend the analysis to random forest. The random forest includes two hyperparameters, mtry which represents the number of predictors that are considered at each node, as well as trees which represents the number of trees in the forest. We run hyperparameter selection first on mtry, then on trees. Results for hyperparameter sweep are shown below.

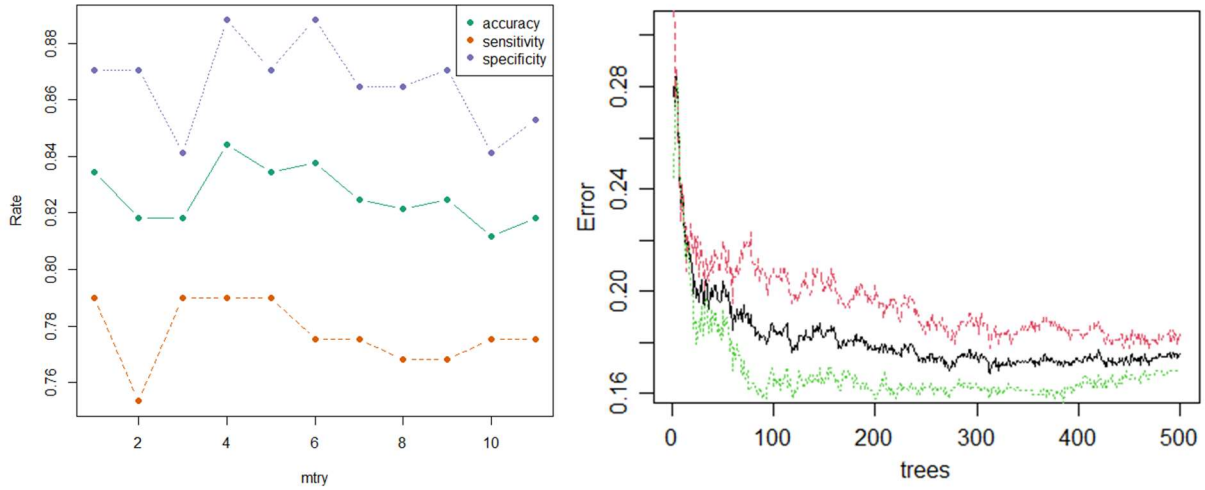


Figure 13: Hyperparameter selection for random forest model, mtry on left and number of trees on right.

Here we can see the 3 most important predictors are alcohol, sulphates and total.sulfur.dioxide. This is aligned with our results for the decision tree. Note alcohol of >0.0963731 gives a high quality. We can see that the highest accuracy takes place at mtry=4, which is the number of predictors sampled for splitting at each node, and we can get the low and stable Out-of-Bag error (black line) from around 350 trees. Results for the model are shown on the next page.

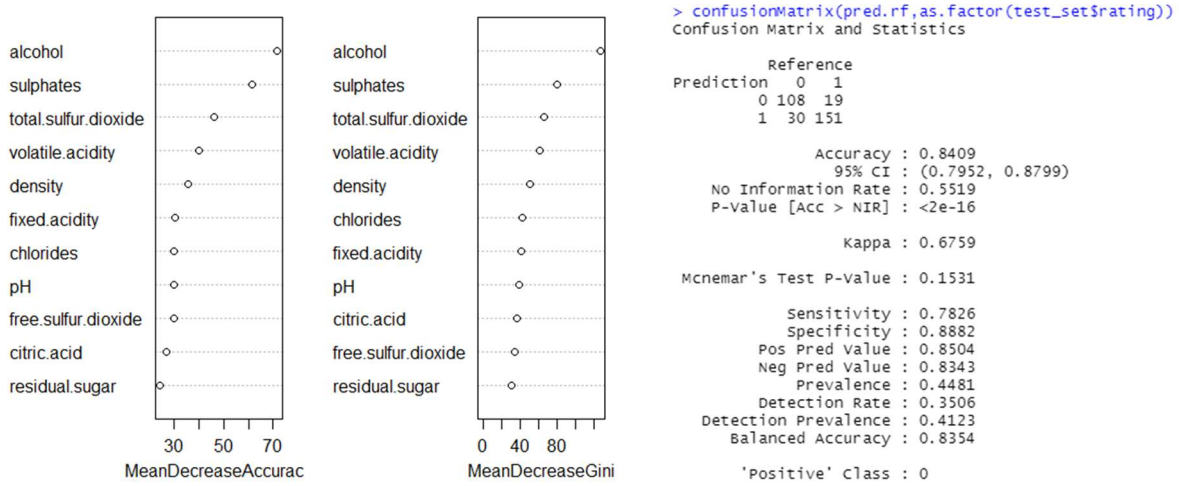


Figure 14: Training model performance, variable importance plot on left (from Random Forest), and confusion matrix on right

Both methods of Boosting (by *gbm()* function) and Random Forest show the 3 most important predictors are alcohol, sulphates and total.sulfur.dioxide, and the Random Forest gives the best fitting performance among all tested classification models, with the accuracy of 0.8409.

3. Relevant Comparisons

3.1 Regression Results

Table 1: Model Performance Comparison for Regression Analysis

Regression Method	Test MSE	Adj-Rsquare
Linear Regression 0	0.3522914	34.7%
Linear Regression 1 removing VIF	0.3499804	34.71%
Polynomial Regression	0.3461992	36.67%
Lasso	0.3281441	37.64%

3.2 Classification Results

Table 2: Model Performance Comparison for Classification Analysis

Classification Method	Accuracy	Specificity	Sensitivity
Logistic Regression	0.7857143	0.7941	0.7754
Decision Tree	0.6720779	0.7352941	0.5942029
Random Forest	0.8409	0.8882	0.7826

4. Conclusion

4.1 Summary and findings

This report uses the red wine datasets from Vinho Verde region in Portugal to predict the quality based on the physicochemical properties. Quality is a subjective measure, given by the median rating of three tasters. Before starting the predictions, this report gives a brief summary of model methodologies, explaining the most common models used in regression/categorical problems, as well as their corresponding bottom-layer mathematical theories.

In the data preparation process, we standardized the predictors to make sure they are using the same dataset for the regression models, and for classification models, we divided quality to High and Low levels by the threshold of 5 as the response. Then we removed the leverages/outliers to improve the model prediction and get the training and testing sets by the ratio of 8:2. In the following process of data exploration and visualization, we look for features that may provide good prediction results. The best predictors have low distribution overlapping area and low correlation among them.

Modeling starts explaining very simple models and gradually moves to more complex ones. In the procedure of model fitting, cross validation and hyperparameter selection are introduced to get the “best-performance model” for each individual model method. In the meanwhile, it comes to be feasible to apply the output of previous models to construct a new type of model, such as the simple linear regression and polynomial regression. The results section presents the modeling results and discusses the model performance. The regression models are evaluated by the MSE and adjusted R^2, and the classification models are evaluated by accuracy, specificity and sensitivity. From the model results, we can conclude that:

- A polynomial relationship exists between sulphates and the dependent variable.
- With cross-validation, Lasso regression has a best prediction performance among the regression methods. It has a lowest MSE of 0.3281441 and highest adjusted R^2 of 37.64%.
- The final model of logistic regression has 8 significant predictors and a good accuracy of 0.7857143.
- In this case, cross-validation gives a decision tree with 7 terminal nodes. It indicates that good quality wines have higher levels of alcohol and sulphates on average.
- Three most important predictors are alcohol, sulphates and total.sulfur.dioxide, given by the methods of decision tree, random forest and boosting.
- In this case, random forest gives a highest accuracy of 0.8409, with mtry of 4 and tress numbers of around 350.

4.2 Issues

The prediction of quality can be impacted by the lack of information about how the dataset was collected and other important variables, such as the production date， accurate review for each taster, composition of grape varieties, and the type of grapes. On the other hand, there are no values of (0, 1, 2) or (9, 10) for the quality response, most of them are 5 and 6. That makes it harder to analyze the data, and looks like the tasters don’t have enough ability to tell which wine is in High/Low quality.

5. References

[1] James, G., Witten, D., Hastie, T. and Tibshirani, R., 2013. *An introduction to statistical learning* (Vol. 112, p. 18). New York: springer.