Modeling to Predict Wine Quality

SDS 323 Final Project

YoungRi Lee (yl29982) & Jihyun Lee (jl64875)

1 Introduction

We used a wine quality dataset from Cortez et al. (2009) [1]. The dataset only includes *vinho verde*, a unique product from the Minho (northwest) region of Portugal. The outcome variable is `wine quality`, measured by a minimum score of three sensory assessors using blind tastes in a scale ranging from 0 (very bad) to 10 (excellent). There are 11 attributes of the wine based on physicochemical tests (Table 1). Initially, two datasets of red wine ($n = 1,599$) and white wine ($n = 4,988$) were available. We merged the two and used for the analyses by creating an additional dummy variable to indicate the wine type, `red`. There is no missing value in this dataset. Figure 1 shows the distribution of wine quality by wine type. Generally, it shows a normal shape distribution and centered around the middle point of the scale. Red wine has fewer observations than white wine.

Our research goals are as follows: (1) to build the best model to predict wine quality, and (2) to examine the most influential set of attributes to predict wine quality. We implemented three techniques: shrinkage approach (i.e., Lasso) and tree-based approaches (i.e., Random Forest and Boosting). Based on the comparison of the alternative methods, we proposed the best model for predicting wine quality and

Table 1: Descriptive statistics of 11 attributes and outcome

	White			Red		
	Min	Max	Mean	Min	Max	Mean
Fixed acidity	3.8	14.2	6.855	4.6	15.9	8.320
Volatile acidity	0.08	1.10	0.278	0.12	1.58	0.528
Citric acid	0	1.66	0.334	0	1.00	0.271
Residual sugar	0.6	65.8	6.391	0.9	15.5	2.539
Chlorides	0.009	0.346	0.046	0.012	0.611	0.087
Free sulfur dioxide	2	289	35.308	1	72	15.875
Total sulfur dioxide	9	440	138.361	6	289	46.468
Density	0.987	1.039	0.994	0.990	1.004	0.997
pH	2.72	3.82	3.188	2.74	4.01	3.311
Sulphates	0.22	1.08	0.490	0.33	2.00	0.658
Alcohol	8.0	14.2	10.514	8.4	14.9	10.423
Wine Quality	3	9	5.878	3	8	5.636

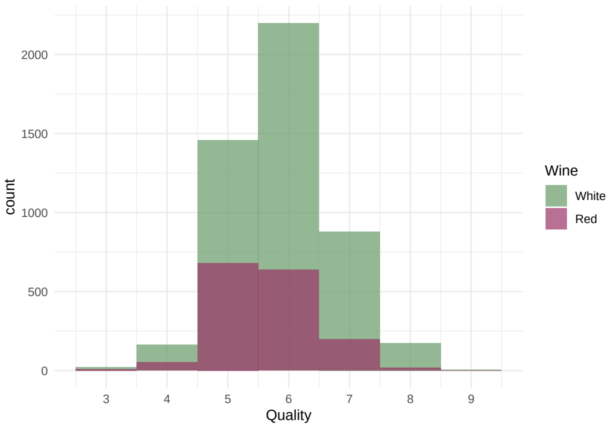


Figure 1: Histrogram of wine quality by wine type

suggest the most influential variables on the prediction.

2 Methods

2.1 Shrinkage Method: Lasso

The two well-known shrinkage methods are *Ridge regression* and the *Lasso*. The benefit of these methods is to reduce the variance by shrinking the relatively unimportant attributes and selecting variables. However, it may lead to the bias of the model by reducing the variance (*bias-variance trade-off*). We choose Lasso over Ridge regression because Lasso forces the coefficients to be equal to zero, and it outperforms ridge when variables are highly correlated. In Lasso, λ is a *tuning parameter*, which controls the impact of shrinkage penalty. We selected λ by the cross-validation.

2.2 Tree-based Methods

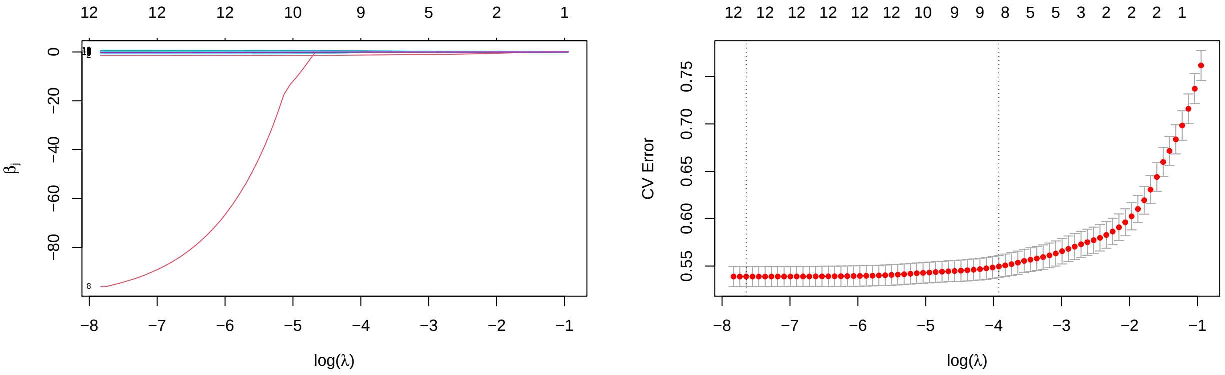
Tree-based methods can be used for regression and classification. This method segments the predictor space into several simple regions, and each rule that split the segment is summarized. We used two of ensemble methods, **random forests** and **boosting**, instead of growing a single tree. The ensemble methods are more powerful and useful than a single tree to build prediction models.

The random forests, is an improved version of *bagged* trees, which build a number of decision trees using bootstrapped training samples. Random forests are similar to bagging but use only a random subset of predictors. This process decorrelates the built trees in bagging and reduces the variance. In this study, we will use $m = \sqrt{p} = \sqrt{12}$ as the number of predictors considered at each split.

In boosting, the trees are grown sequentially, and each tree is fitted on the residuals from the previous tree. The final model in boosting is determined by the sum of all fitted trees. While fitting boosting, we specified three parameters: shrinkage parameter (λ), number of iterations (B), and the size of each new tree (d, the number of splits).

3 Results

3.1 Shrinkage Method: Lasso



(a) Lasso coefficients

(b) CV error

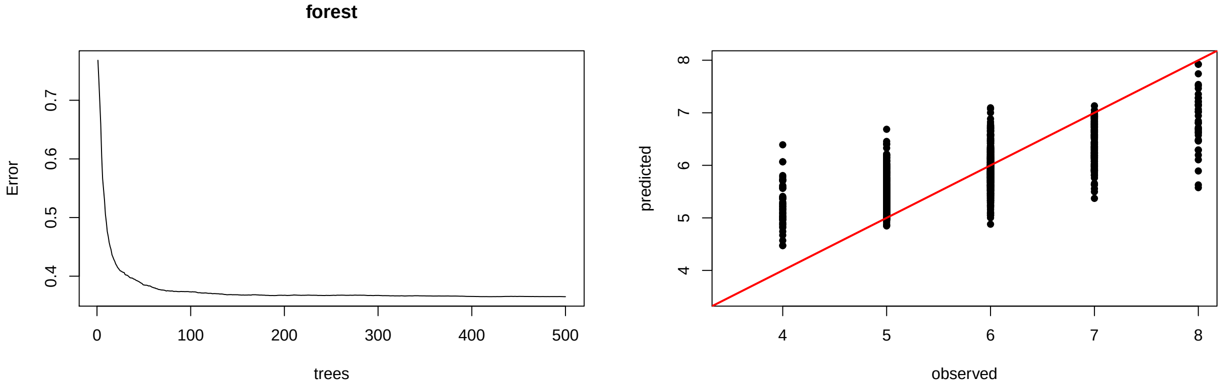
Figure 2: Lasso results

Figure 2(a) shows the lasso coefficients across the $log(\lambda)$. Only one variable, `density`, shows rapid change as $log(\lambda)$ increases and other covariates are close to zero. We use 10-fold cross-validation (CV) to choose the tuning hyperparameter, λ that minimizes the expected out-of-sample prediction error. Figure 2(b) shows CV error as $log(\lambda)$ changes. We chose the optimal model with the minimum CV error, $\hat{\lambda} = 4.7831203 \times 10^{-4}$ (Mean squared error; $MSE = .539$). The optimal model selects all twelve predictors in the model.

3.2 Tree-based Method: Random forests

Figure 3(a) illustrates test (out-of-bag; OOB) error estimation, which suggests OOB error stabilizes before 200 trees. Based on the prediction of random forest, we can see a decent fit of the model in Figure 3(b) with MSE of 0.358.

In the left panel of Figure 4, `%IncMSE` indicates the mean decrease of accuracy in prediction on the out of bag samples when a given variable is excluded from the model. The more considerable reduction indicates that the target variable is important in the model. `IncNodePurity` measures the total decrease



(a) Out-of-bag Error

(b) Prediction

Figure 3: Random forest

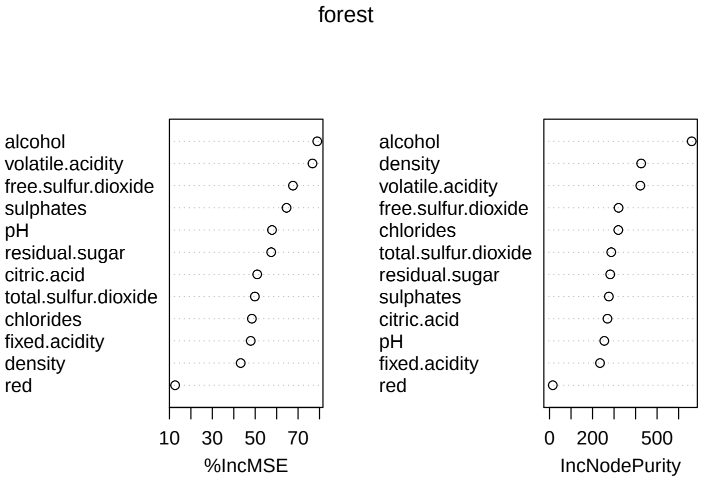


Figure 4: Random Forests: Variable Importance Plot

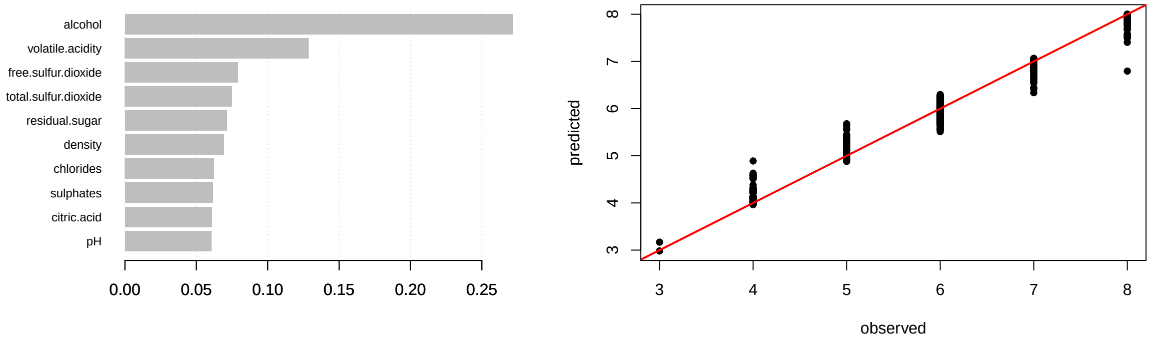
in training RSS (the right panel of Figure 4). Based on the random forests, `Alcohol` seems to be the most important variable and wine type (`red`) seems to be the least important variable in the model.

3.3 Tree-based Method: Boosting

We chose an optimal lambda using the CV approach. Table 2 displays the models with the first five minimum RMSE. The final model is determined based on minimum RMSE (0.602) with $\eta = .05$. Figure 5(a) shows `alcohol` and `volatile.acidity` are important variables, which are similar to that of random forests. The MSE of boosting model is 0.024 [Figure 5(b)].

Table 2: Boosting results

	eta	max_depth	optimal_trees	min_RMSE
14	0.05	7	803	0.602
13	0.01	7	2696	0.606
15	0.15	7	217	0.612
10	0.05	5	1115	0.622
9	0.01	5	4747	0.624



(a) Variable Importance Plot

(b) Prediction

Figure 5: Boosting

4 Conclusion

In this project, we explored the best model to predict wine quality using a real data. We compared lasso, random forests, and boosting, examining each model with the best fit using cross-validation. Overall, tree-based methods showed a better performance than lasso regression (Table 3). Among the tree-based methods, boosting showed a smaller MSE compared to random forests. We concluded that `alcohol` and `volatile.acidity` as the critical variables that influence the wine quality based on the random forests and boosting results. Wine producers will be able to use this information to produce a better wine quality, considering the selected attributes. Also, wine consumers will be able to choose a good quality wine without tasting (e.g., via online shopping) if they are provided the physicochemical information about the wine.

Table 3: MSE Comparison

Lasso	Random Forest	Boosting
0.539	0.358	0.024

References

[1] P. Cortez et al. “Modeling wine preferences by data mining from physicochemical properties”. In: *Decision Support Systems* 47.4 (2009), pp. 547–553.