Predicting Red Wine Quality with Different Models

Shijing Zhou[1]

[1] University of Oregon

https://github.com/shijing-z/EDLD654-Final-Project.git

Predicting Red Wine Quality with Different Models

Research Problem

Wine Quality Data Set is obtained from UCI Machine Learning Repository. The website contains two datasets, which are related to red and while wines sample from vinho verde, which is from the north of Portugal (Cortez et al., 2009). For this project, only data on the red wine samples were used to create models. The aim of the project is to use physicochemical data of wine to predict the quality of wine. Building a model of predicting red wine quality from objective data could potentially not only help to establish wine tasting guideline from the perspective of merchants and consumers, but also help to improve wine production from the perspective of winery as the producer.

Description of the Data

Core features and descriptive statistics

The dataset contains a total of 12 variables. The outcome of interest is wine quality (`quality`). There are also physicochemical measures of red wine samples, including fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol.

```
##                        vars    n  mean    sd  min    max  range   se
## fixed acidity             1 1599  8.32  1.74 4.60  15.90  11.30 0.04
## volatile acidity          2 1599  0.53  0.18 0.12   1.58   1.46 0.00
## citric acid               3 1599  0.27  0.19 0.00   1.00   1.00 0.00
## residual sugar            4 1599  2.54  1.41 0.90  15.50  14.60 0.04
## chlorides                 5 1599  0.09  0.05 0.01   0.61   0.60 0.00
## free sulfur dioxide       6 1599 15.87 10.46 1.00  72.00  71.00 0.26
## total sulfur dioxide      7 1599 46.47 32.90 6.00 289.00 283.00 0.82
## density                   8 1599  1.00  0.00 0.99   1.00   0.01 0.00
```

```
## pH                     9 1599  3.31  0.15 2.74   4.01   1.27 0.00
## sulphates             10 1599  0.66  0.17 0.33   2.00   1.67 0.00
## alcohol               11 1599 10.42  1.07 8.40  14.90   6.50 0.03
## quality               12 1599  5.64  0.81 3.00   8.00   5.00 0.02
```

Missing data check

No missingness was found for the variables in the dataset.

```
##                         n missing_percent
## fixed.acidity        1599             0.0
## volatile.acidity     1599             0.0
## citric.acid          1599             0.0
## residual.sugar       1599             0.0
## chlorides            1599             0.0
## free.sulfur.dioxide  1599             0.0
## total.sulfur.dioxide 1599             0.0
## density              1599             0.0
## pH                   1599             0.0
## sulphates            1599             0.0
## alcohol              1599             0.0
## quality              1599             0.0
```

Outcome transformation

As a consumer, I may consider `quality` as a key binary outcome (i.e., good or bad) for my decision on which wine I should buy. Hence, it makes sense to transform the variable, `quality`, to a categorical variable with binary outcomes (i.e., 1 = Good, 0 = Bad).

```
wine$quality <- I(wine$quality > 6) * 1
```

Description of the models

Three different modeling approaches will be used to predict quality of wine from 11 physicochemical measures of wine, including Logistic Regression, Classification Trees, and Random Forest. Since the aim of the project is to develop a tool that could be used by both consumers, merchants, and winery, it make sense to treat the outcome of interest, `quality`, as binary and run a logistic regression with other continuous physicochemical variables. It is always good to run a generalized linear model (GLM) as a baseline to compare with other more advanced models. For classification tree, it is a advanced tool for outcome prediction. Also, for winery as the producer of wine, decision trees may help them to find and prioritize the most important factors for wine quality during production. Random Forests is a even more advanced tool using bootstrap (i.e., random sample of rows in training dataset with replacement) to predict more unbiased outcomes.

For all models, I am planning to use Area Under the Receiver Operating Curve (AUC or AUROC) and True Positive Rate (TPN) to evaluate those models. For the outcome of interest with different perspectives from winery, merchants, and consumers, it makes the most sense to see how well the model does to predict good quality wine when the wine is really good, because it is related to the profit of winery and merchants, and consumer experience experience.

Model Fits

Preparation

The dataset is split into training and test set with the following code. The training set has 1,000 observations, and the test set has 599 observations. I also prepared a function to easy calculate TNR for each model.

```
set.seed(8)
X <- scale(wine[,1:11])
tst <- 1:599
train <- wine[-tst,]
test <- wine[tst,]

# Function to calculate True Postive Rate (TPR)
TPR <- function(y,yhat)  { sum(y==1 & yhat==1) / sum(y==1) }
```

Model 1: Logistic Regression

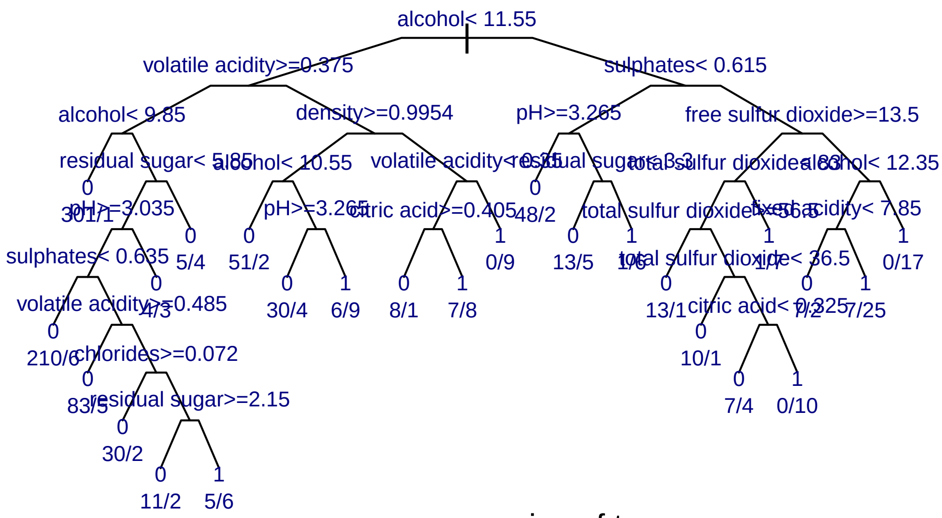
The logistic regression indicated a TRP of 21.33%, and a AUC of 87.22%.

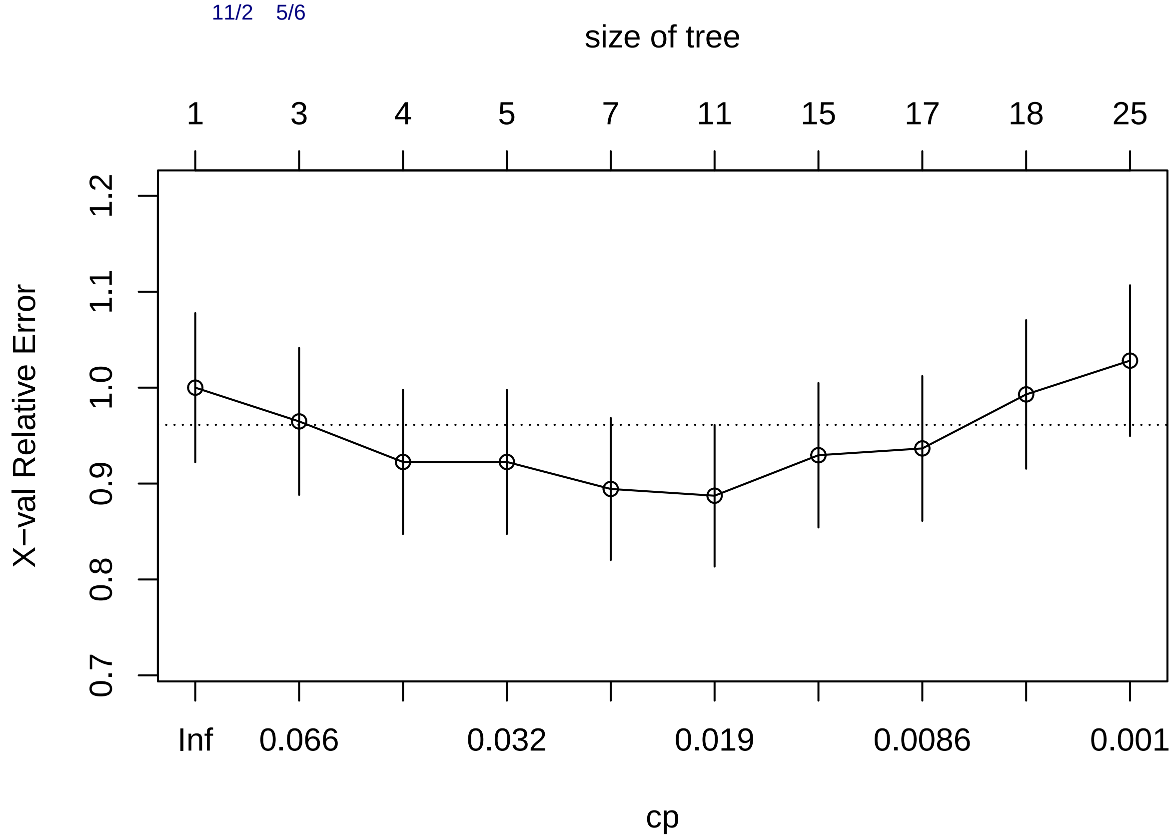
```
##
##     FALSE TRUE
##   0   503   21
##   1    59   16

## [1] 0.2133333

##
## Call:
## roc.default(response = test$quality, predictor = test$yhat.glm,     direction = "<")
##
## Data: test$yhat.glm in 524 controls (test$quality 0) < 75 cases (test$quality 1).
## Area under the curve: 0.8722
```

Model 2: Decision Tree

The classification trees model after pruning indicated a TRP of 55.26%, and a AUC of 79.51%.

Classification trees 1. A explorotory classification trees model

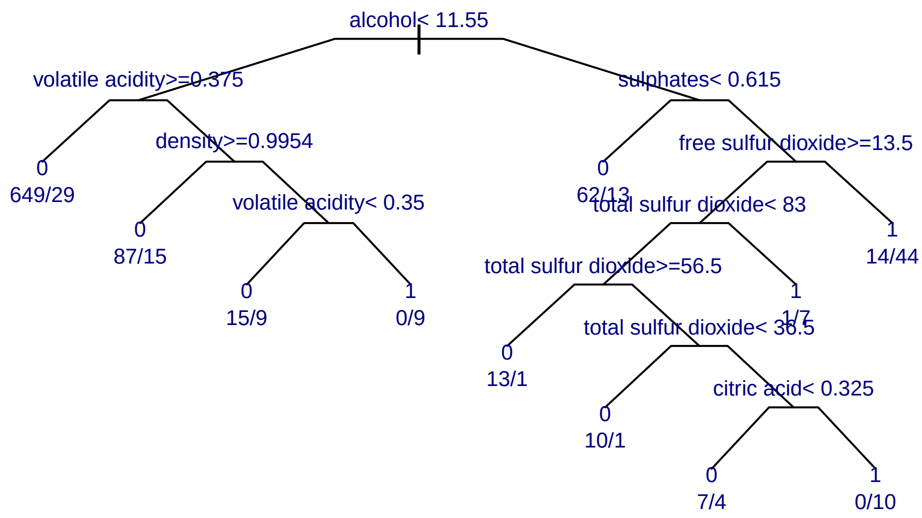





```
##
## Classification tree:
## rpart(formula = form1, data = train, method = "class", cp = 0.001)
##
```

```
## Variables actually used in tree construction:
##  [1] alcohol              chlorides            citric acid
##  [4] density              fixed acidity        free sulfur dioxide
##  [7] pH                   residual sugar       sulphates
## [10] total sulfur dioxide volatile acidity
##
## Root node error: 142/1000 = 0.142
##
## n= 1000
##
##           CP nsplit rel error  xerror     xstd
## 1  0.0774648      0   1.00000 1.00000 0.077732
## 2  0.0563380      2   0.84507 0.96479 0.076573
## 3  0.0422535      3   0.78873 0.92254 0.075138
## 4  0.0246479      4   0.74648 0.92254 0.075138
## 5  0.0211268      6   0.69718 0.89437 0.074152
## 6  0.0176056     10   0.61268 0.88732 0.073901
## 7  0.0105634     14   0.54225 0.92958 0.075380
## 8  0.0070423     16   0.52113 0.93662 0.075622
## 9  0.0010060     17   0.51408 0.99296 0.077503
## 10 0.0010000     24   0.50704 1.02817 0.078635
```

Classification trees 2. A new `cp` value is used in classification tree model 2 based on the classification tree model 1. The new `cp` value for the second tree model is based on the value of relative error, x error, xstd. When nsplit = 10, all error values are at their lowest.




```
##
##          0   1
##   FALSE 507  54
##   TRUE   17  21
```

```
## [1] 0.5526316
```

```
##
## Call:
## roc.default(response = test$quality, predictor = yhat.t2, direction = "<")
##
## Data: yhat.t2 in 524 controls (test$quality 0) < 75 cases (test$quality 1).
## Area under the curve: 0.7915
```

Model 3: Random Forest

The Ramdom Forest model indicated a TRP of 80%, and a AUC of 86.54%.

```
##
## Call:
##  randomForest(x = X, y = Y, ntree = ntree, mtry = mtry, importance = TRUE)
```

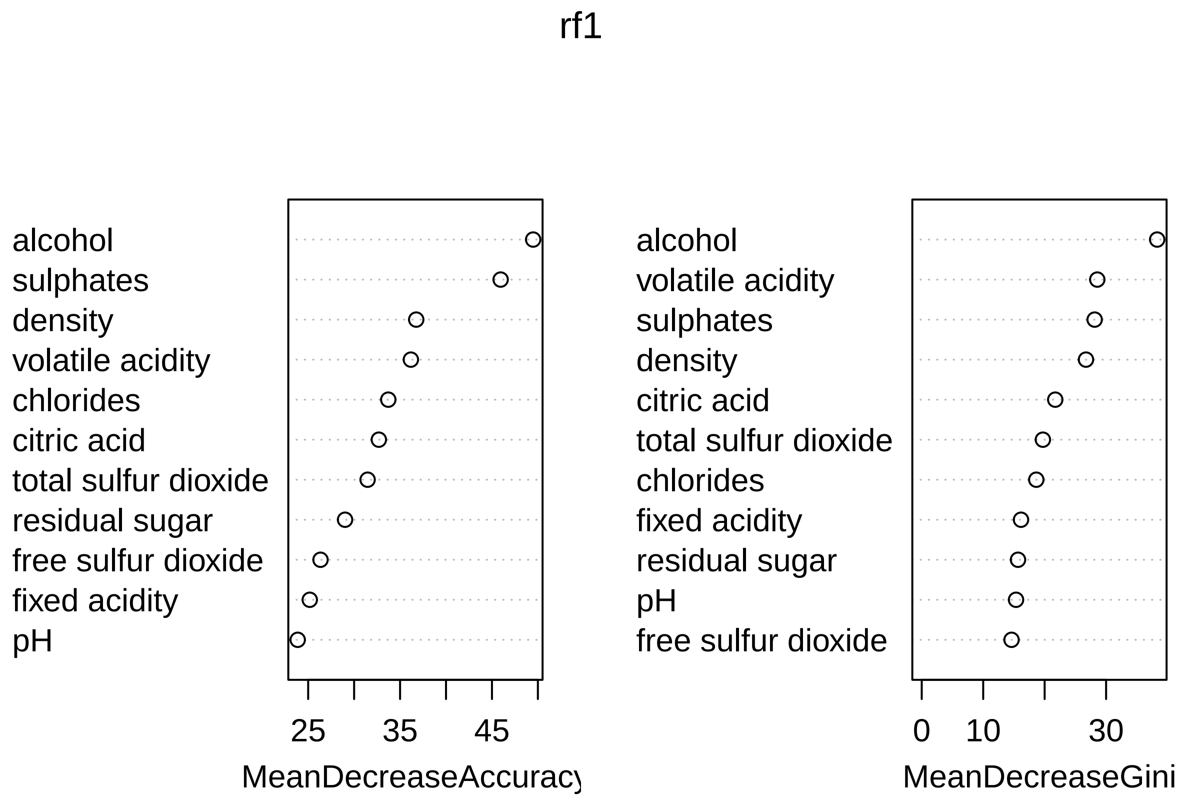
```
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 3
##
##        OOB estimate of  error rate: 8.7%
## Confusion matrix:
##     0  1 class.error
## 0 832 26  0.03030303
## 1  61 81  0.42957746
```

```
##                 Length Class  Mode
## call               6   -none- call
## type               1   -none- character
## predicted       1000   factor numeric
## err.rate        3000   -none- numeric
## confusion          6   -none- numeric
## votes           2000   matrix numeric
## oob.times       1000   -none- numeric
## classes            2   -none- character
## importance        44   -none- numeric
## importanceSD      33   -none- numeric
## localImportance    0   -none- NULL
## proximity          0   -none- NULL
## ntree              1   -none- numeric
## mtry               1   -none- numeric
## forest            14   -none- list
## y               1000   factor numeric
## test               0   -none- NULL
```

```
## inbag              0   -none- NULL
```

```
##  [1] "call"            "type"            "predicted"       "err.rate"
##  [5] "confusion"       "votes"           "oob.times"       "classes"
##  [9] "importance"      "importanceSD"    "localImportance" "proximity"
## [13] "ntree"           "mtry"            "forest"          "y"
## [17] "test"            "inbag"
```

```
##                             0        1 MeanDecreaseAccuracy MeanDecreaseGini
## fixed acidity        16.10640 19.40485             25.17405         16.15569
## volatile acidity     14.67521 38.26160             36.17579         28.56329
## citric acid          14.05525 30.46566             32.68994         21.72569
## residual sugar       20.69442 23.17121             29.01399         15.66553
## chlorides            22.29886 27.02336             33.71803         18.64078
## free sulfur dioxide  18.27226 20.25460             26.34248         14.61407
## total sulfur dioxide 19.05263 34.14284             31.46893         19.71107
## density              22.54147 35.11311             36.75894         26.72439
## pH                   13.35736 22.92480             23.86168         15.34353
## sulphates            16.45267 54.22843             45.94490         28.13688
## alcohol              22.12729 50.92340             49.48475         38.30781
```



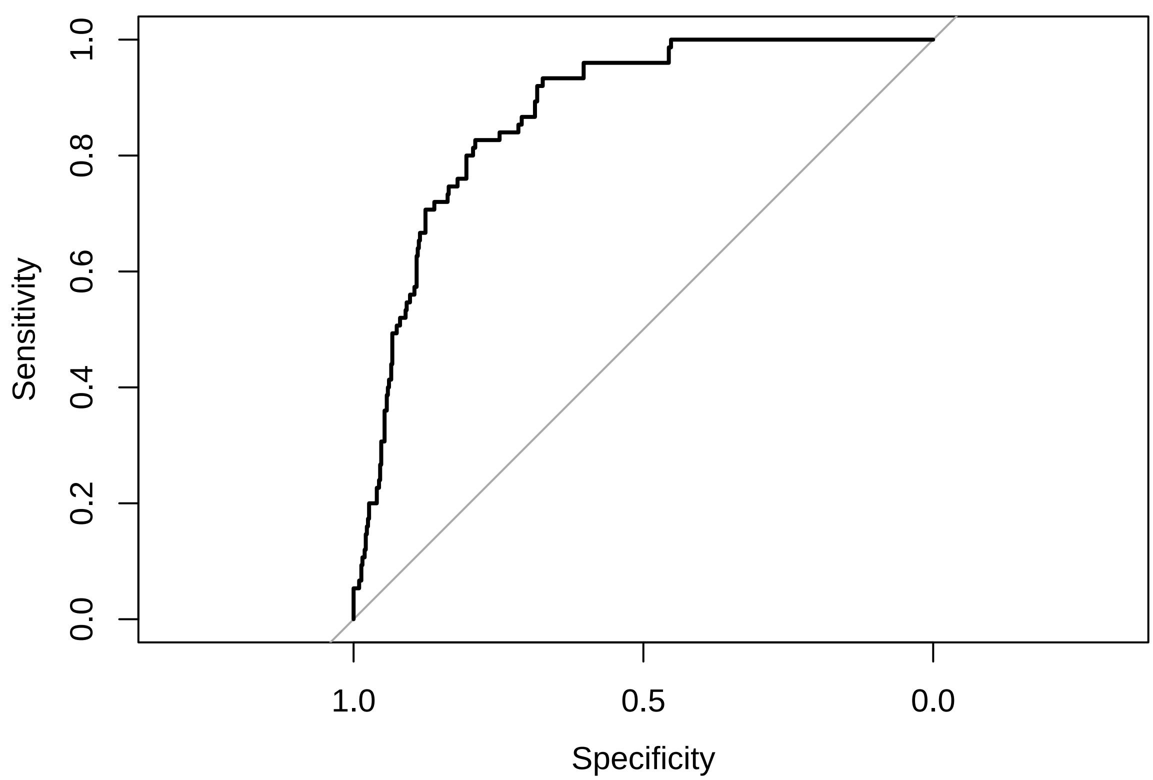

```
##
## pred.rf1   0   1
##        0 519  59
##        1   5  16
```

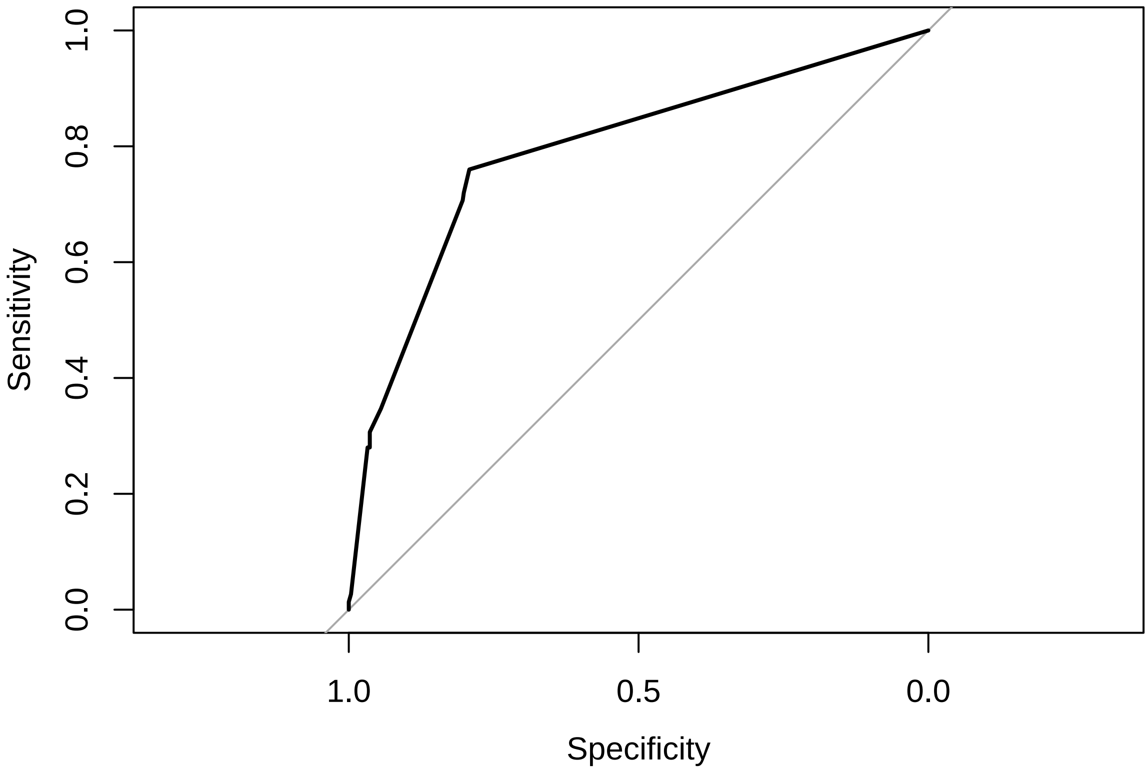
```
## [1] 0.7619048
```

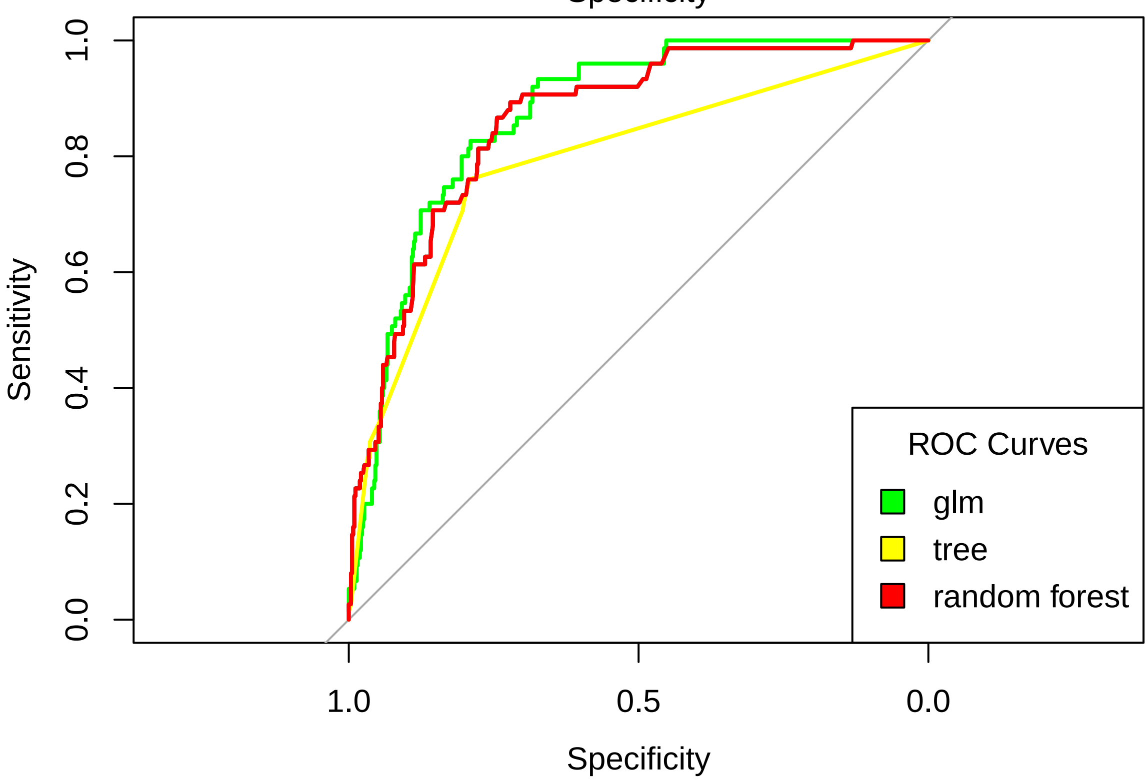
```
##
## Call:
## roc.default(response = test$quality, predictor = yhat.rf1, direction = "<")
##
## Data: yhat.rf1 in 524 controls (test$quality 0) < 75 cases (test$quality 1).
## Area under the curve: 0.8604
```

Plot of Model Comparison

Based on TPR and AUC, it seems that Random Forest model did an outstanding job to predict the outcome. However, based on ROC curves (i.e., trade-off between TPR (sensitivity) and TNR (specificity)), it looks that the Random Forest model performed just slightly better than the logistic regression model, and the worst performed model seems to be classification trees (after pruning) model. Hence, I would choose Random Forest model as the optimal model to predict wine quality from from physicochemical data of wine with the comparisions and results described above. See the following plot for model comparison.




Sensitivity
Specificity
1.0
0.8
0.6
0.4
0.2
0.0
1.0
0.5
0.0


Sensitivity
Specificity
1.0
0.8
0.6
0.4
0.2
0.0
1.0
0.5
0.0
ROC Curves
glm
tree
random forest

Discussion

The three different models definitely gave me different results on predicting powder depending on different method of evaluation. If I only consider TPR and AUC for my model performance, the random forest model is outstanding compared the rest of models, but the random forest model seemed to perform similarly if I also take True Negative Rate (TNR) into consideration.

In the random forest model, it looks like `alcohol` was the most important predictor for the outcome, `quality`. This is certainly surprising for me. I thought factors such as pH levels and residual sugar matter more regarding the taste. However, I realized that wine quality is not all about taste. Color, smell, how wine looks from different angles of glass, and how wine swirls in a glass also matter to wine quality. I think this is very informative, mostly for winery as the producer of wine, to focus on how alcohol plays a role in production to improve their products.

References

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems, 47(4), 547–553. https://doi.org/10.1016/j.dss.2009.05.016