WINE QUALITY PREDICTION MODEL

By: Danny Boudagian, Olivia Desso, Ajdin Kucevic, Lucas McDonnell

CONTENT

- IS THE MODEL APPROPRIATELY MOTIVATED BY THE BUSINESS USE CASE?
- ARE FEATURES TRANSFORMED APPROPRIATELY FOR MODELING?
- DOES THE PRESENTATION SHOW ENOUGH SUMMARY STATISTICS, BACKGROUND INFORMATION, AND FIGURES TO DESCRIBE THE DATA?
- IS THE FIRST MODEL APPROPRIATELY ESTIMATED?
- IS THE SECOND MODEL APPROPRIATELY ESTIMATED?
- DOES THE STUDENT PROPERLY EVALUATE THE PERFORMANCE OF THE MODEL, APPROPRIATELY FOR THE QUESTION AND DATA?
- DOES THE STUDENT COMMENT ON THE MODEL FIT AND MAKE A RECOMMENDATION REGARDING THE MOTIVATION FOR THE BUSINESS USE CASE OF THE ANALYTICS MODEL?
- ARE THE OVERALL AESTHETICS OF THE SLIDES PRESENTABLE TO A BUSINESS ENVIRONMENT?

Motivation/Business Value and Variables

Which variables are most significant in predicting the wine quality?

MSE.

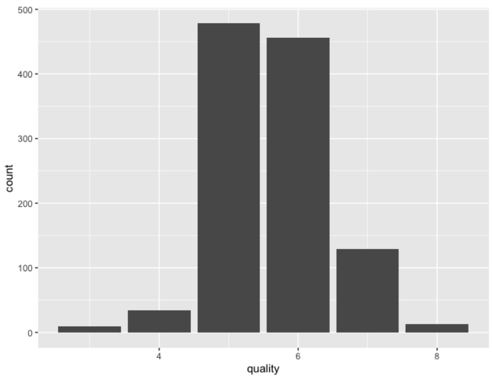
- We want to determine what kinds of factors go into wine quality so that producers will be able to evaluate how to make a higher quality wine
- Wine industry currently valued at **$340 Billion**
- The industry preserves agricultural land, American jobs, attracts tourism, and generates taxes
- Being able to make a high-quality wine can significantly increase profitability
- We are trying to understand and predict the impact that variables have on the quality of wine
- We are using predictors such as pH level, alcohol level, density, and citric acid levels, etc

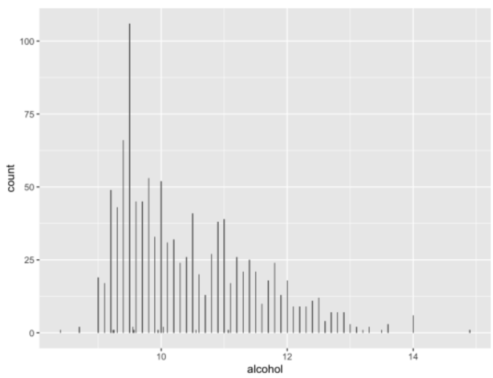
RAW DATA

```
> colSums(is.na(wine))
     fixed.acidity     volatile.acidity          citric.acid       residual.sugar
                 0                    0                    0                    0
         chlorides  free.sulfur.dioxide total.sulfur.dioxide              density
                 0                    0                    0                    0
                pH            sulphates              alcohol              quality
                 0                    0                    0                    0
```

```
## Rows: 1,599
## Columns: 12
## $ fixed.acidity        <dbl> 7.4, 7.8, 7.8, 11.2, 7.4, 7.4, 7.9, 7.3, 7.8, 7.5…
## $ volatile.acidity     <dbl> 0.700, 0.880, 0.760, 0.280, 0.700, 0.660, 0.600, …
## $ citric.acid          <dbl> 0.00, 0.00, 0.04, 0.56, 0.00, 0.00, 0.06, 0.00, 0…
## $ residual.sugar       <dbl> 1.9, 2.6, 2.3, 1.9, 1.9, 1.8, 1.6, 1.2, 2.0, 6.1,…
## $ chlorides            <dbl> 0.076, 0.098, 0.092, 0.075, 0.076, 0.075, 0.069, …
## $ free.sulfur.dioxide  <dbl> 11, 25, 15, 17, 11, 13, 15, 15, 9, 17, 15, 17, 16…
## $ total.sulfur.dioxide <dbl> 34, 67, 54, 60, 34, 40, 59, 21, 18, 102, 65, 102,…
## $ density              <dbl> 0.9978, 0.9968, 0.9970, 0.9980, 0.9978, 0.9978, 0…
## $ pH                   <dbl> 3.51, 3.20, 3.26, 3.16, 3.51, 3.51, 3.30, 3.39, 3…
## $ sulphates            <dbl> 0.56, 0.68, 0.65, 0.58, 0.56, 0.56, 0.46, 0.47, 0…
## $ alcohol              <dbl> 9.4, 9.8, 9.8, 9.8, 9.4, 9.4, 9.4, 10.0, 9.5, 10.…
## $ quality              <int> 5, 5, 5, 6, 5, 5, 5, 7, 7, 5, 5, 5, 5, 5, 5, 7…
```

- 12 columns (11 continuous variables and 1 categorical one)
- 1,599 rows
- No data cleaning required since there were no missing variables in the dataset
- We used 70-30 Training-Test Split for our dataset on all models
- Overall, we believed that these variables seemed to be strong indicators for predicting the quality of wine





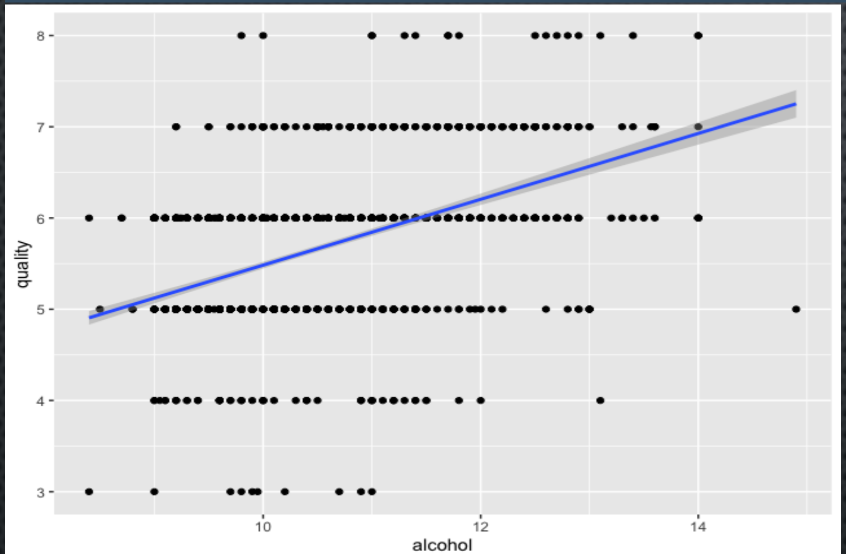

```
##  fixed.acidity   volatile.acidity  citric.acid    residual.sugar
##  Min.   : 4.60   Min.   :0.1200   Min.   :0.000   Min.   : 0.900
##  1st Qu.: 7.10   1st Qu.:0.3900   1st Qu.:0.090   1st Qu.: 1.900
##  Median : 7.90   Median :0.5200   Median :0.260   Median : 2.200
##  Mean   : 8.32   Mean   :0.5278   Mean   :0.271   Mean   : 2.539
##  3rd Qu.: 9.20   3rd Qu.:0.6400   3rd Qu.:0.420   3rd Qu.: 2.600
##  Max.   :15.90   Max.   :1.5800   Max.   :1.000   Max.   :15.500
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide    density
##  Min.   :0.01200   Min.   : 1.00       Min.   :  6.00       Min.   :0.9901
##  1st Qu.:0.07000   1st Qu.: 7.00       1st Qu.: 22.00       1st Qu.:0.9956
##  Median :0.07900   Median :14.00       Median : 38.00       Median :0.9968
##  Mean   :0.08747   Mean   :15.87       Mean   : 46.47       Mean   :0.9967
##  3rd Qu.:0.09000   3rd Qu.:21.00       3rd Qu.: 62.00       3rd Qu.:0.9978
##  Max.   :0.61100   Max.   :72.00       Max.   :289.00       Max.   :1.0037
##        pH          sulphates         alcohol         quality
##  Min.   :2.740   Min.   :0.3300   Min.   : 8.40   Min.   :3.000
##  1st Qu.:3.210   1st Qu.:0.5500   1st Qu.: 9.50   1st Qu.:5.000
##  Median :3.310   Median :0.6200   Median :10.20   Median :6.000
##  Mean   :3.311   Mean   :0.6581   Mean   :10.42   Mean   :5.636
##  3rd Qu.:3.400   3rd Qu.:0.7300   3rd Qu.:11.10   3rd Qu.:6.000
##  Max.   :4.010   Max.   :2.0000   Max.   :14.90   Max.   :8.000
```

Summary Statistics

- The average wine quality can be seen to be closer to 6, with anything above it meaning good wine, while anything below the number 6 is considered bad wine

LINEAR REGRESSION(BASELINE)

```
Residual standard error: 0.648 on 1587 degrees of freedom
Multiple R-squared:  0.3606,    Adjusted R-squared:  0.3561
F-statistic: 81.35 on 11 and 1587 DF,  p-value: < 2.2e-16

>
```



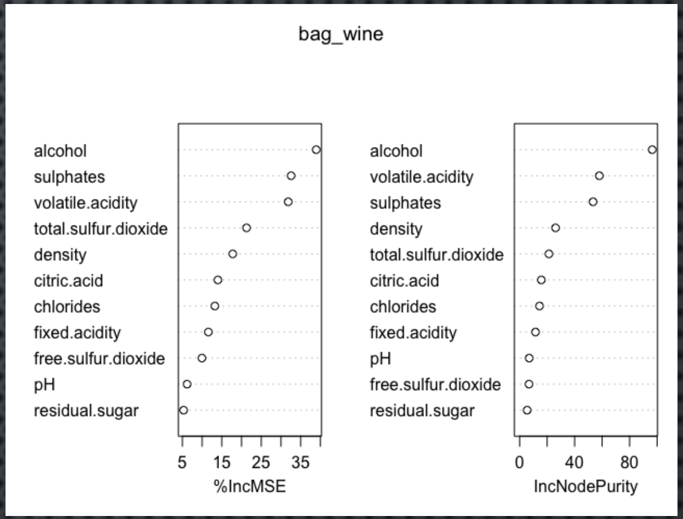

```
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)           2.197e+01  2.119e+01   1.036   0.3002
fixed.acidity         2.499e-02  2.595e-02   0.963   0.3357
volatile.acidity     -1.084e+00  1.211e-01  -8.948  < 2e-16 ***
citric.acid          -1.826e-01  1.472e-01  -1.240   0.2150
residual.sugar        1.633e-02  1.500e-02   1.089   0.2765
chlorides            -1.874e+00  4.193e-01  -4.470 8.37e-06 ***
free.sulfur.dioxide   4.361e-03  2.171e-03   2.009   0.0447 *
total.sulfur.dioxide -3.265e-03  7.287e-04  -4.480 8.00e-06 ***
density              -1.788e+01  2.163e+01  -0.827   0.4086
pH                   -4.137e-01  1.916e-01  -2.159   0.0310 *
sulphates             9.163e-01  1.143e-01   8.014 2.13e-15 ***
alcohol               2.762e-01  2.648e-02  10.429  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

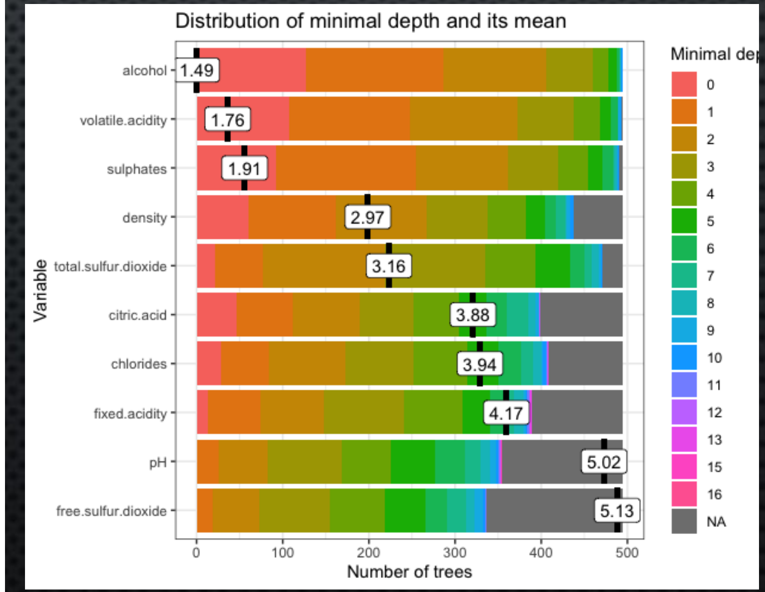
- We use Linear Regression as our baseline model since it's easy to comprehend as well as it being computationally inexpensive
- We find that the most statistically significant variables seem to be both alcohol content and the level of sulfate dioxide gas in the wine bottle that seem to have the biggest impact on the quality of wine at a 100% confidence level
- The Adjusted R^2 is quite low (0.36) meaning that our predictors overall don't do a great job in explaining the variance found in the quality of wine as well not being able to predict the model as well as we thought it would
- Not the start we wanted but we can definitely improve this model with our next two models in both accuracy and prediction in order to help wine makers improve the quality of their product to both regular customers and to wine connoisseurs

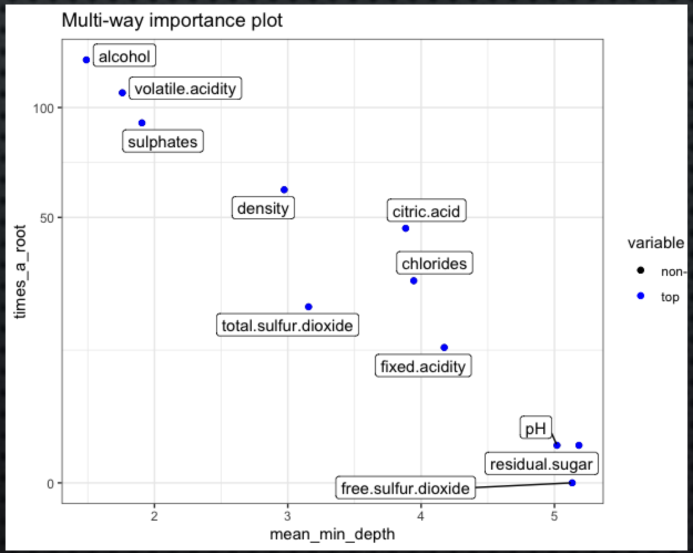
RANDOM FOREST

- We use an ensemble method here in order to stop the overfitting that is occurring in our model since overfitting also leads to high variance
- Both plots (minimal depth distribution and multi-way importance plot) help us determine the importance of each variable
- As seen with the linear regression model, we see that our models that are most significant are alcohol, volatile acidity, and sulphates where this can be concluded since they require the least amount of tree depth which means in a regression tree, they are seen to be at the very top of the tree

```
                     %IncMSE IncNodePurity
fixed.acidity        11.592998     11.550449
volatile.acidity     31.835956     57.982307
citric.acid          14.011371     15.770396
residual.sugar        5.323692      5.462965
chlorides            13.239933     14.464011
free.sulfur.dioxide   9.963321      6.805585
total.sulfur.dioxide 21.267020     21.318083
density              17.772679     26.138066
pH                    6.197876      6.971193
sulphates            32.571684     53.409127
alcohol              38.934625     96.449858
> # importance plot
> varImpPlot(bag_wine)
>
```







RANDOM FOREST

```r
# using random forest function with mtry = p
bag_wine <- randomForest(quality ~ .,
                         data = wine_train,
                         ntree = 500,
                         mtry = 3,
                         nodesize = 120,
                         err.rate = 0.1,
                         importance = TRUE)
```

Before

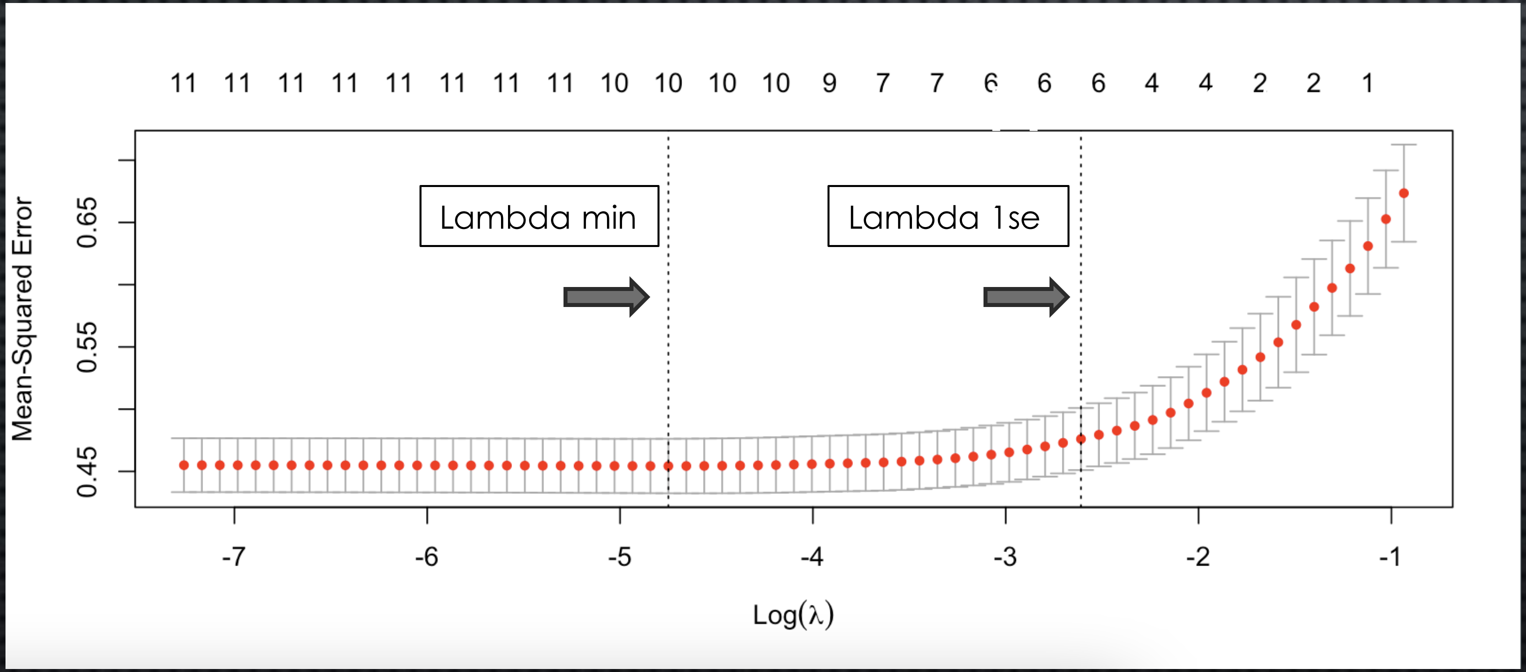
```
Call:
 randomForest(formula = quality ~ ., data = wine_train, ntree = 500,      mtry = 3, importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 3

          Mean of squared residuals: 0.3346744
                    % Var explained: 47.98
> RMSE(predicted_wine_train, wine_train$quality)
[1] 0.262253
> RMSE(predicted_wine_test, wine_test$quality) # RMSE = 2.11, 2.11k in this case
[1] 0.6049808
>
```

After

```
Call:
 randomForest(formula = quality ~ ., data = wine_train, ntree = 500,      mtry = 3, nodesize = 120, err.rate = 0.1, importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 3

          Mean of squared residuals: 0.3956377
                    % Var explained: 38.5
> plot(bag_wine)
> RMSE(predicted_wine_train, wine_train$quality)
[1] 0.5704195
> RMSE(predicted_wine_test, wine_test$quality) # RMSE = 2.11, 2.11k in this case
[1] 0.6514995
>
```

- We use parameters such as nodesize and err.rate to reduce the variance and overfitting of our dataset along with the usual importance, ntree, and mtry parameters that are often used in bagging.
- The importance of nodesize is fairly underestimated as it helped reduce the overfitting massively
- Nodesize- Determines the minimum number of observations in each terminal leaf node
- The higher the nodesize the fewer leaf nodes we have, which reduces the complexity of the model
- When plotting the random forest function, we see that around 100 trees gave use the least amount of error with the error being constant thereafter
- Err.rate - Slight difference, gives us a smaller difference in error size but nothing drastic

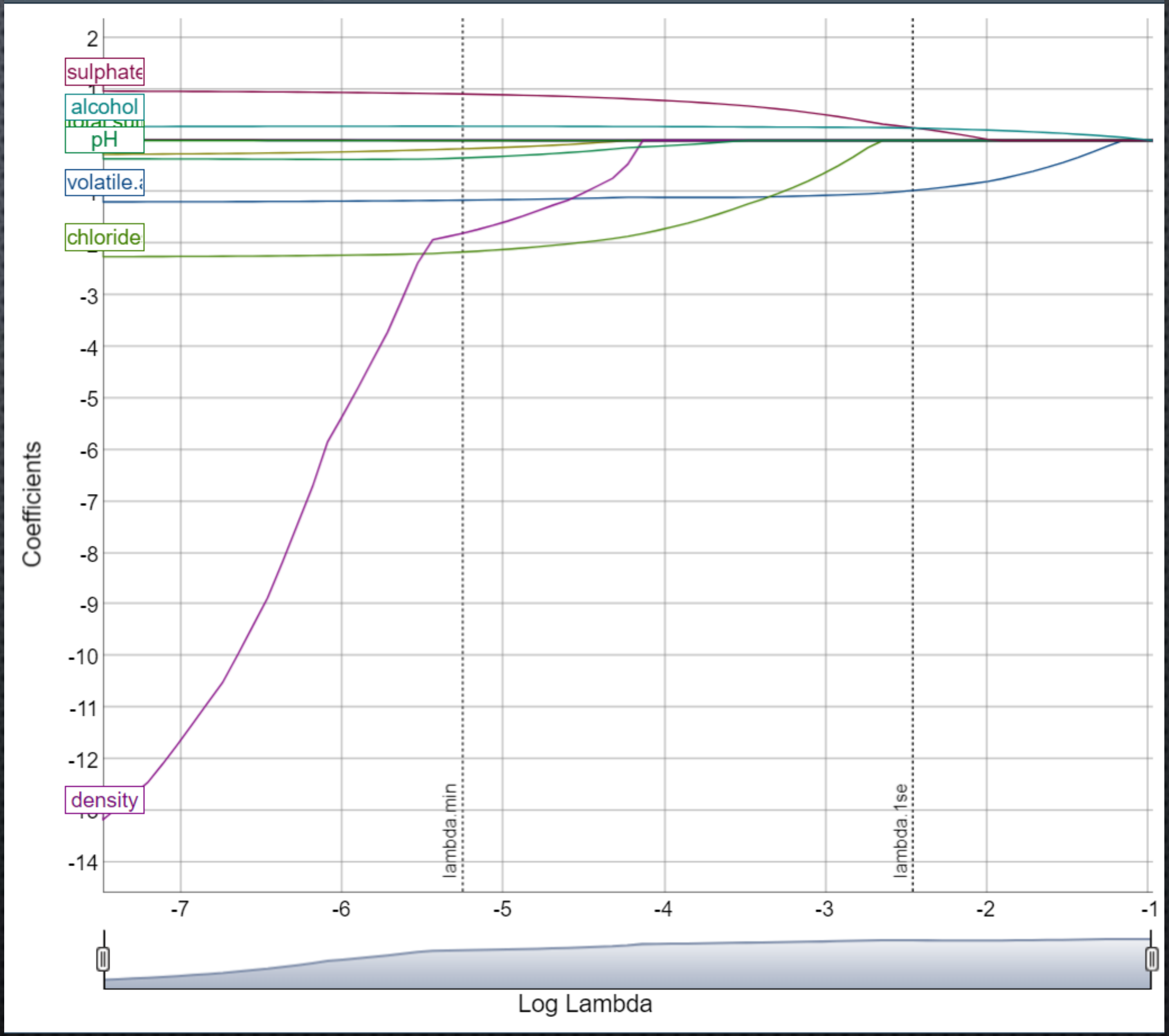
LASSO REGRESSION




```
# Estimating the model

lasso_mod = cv.glmnet(quality ~ .,
                      data = wine_train,
                      alpha = 1)

# Lambda

print(lasso_mod$lamba.min) [1] 0.005272493
print(lasso_mod$lambda.1se) [1] 0.0859285

# Coefficients
coef(lasso_mod, s = lasso_mod$lambda.1se) %>%
  as.matrix() %>%
  as.data.frame() %>%
  round(3)


coef(lasso_mod, s = lasso_mod$lambda.min) %>%
  as.matrix() %>%
  as.data.frame() %>%
  round(3)
```

LASSO REGRESSION

- Similar to the linear model, variables like alcohol, sulphates, and acidity are most significant
- Many variables have minimal significance
- Lambda min more appropriate



COEFFICIENTS

Lambda 1se

	s1
(Intercept)	3.394
fixed.acidity	0.000
volatile.acidity	-0.996
citric.acid	0.000
residual.sugar	0.000
chlorides	0.000
free.sulfur.dioxide	0.000
total.sulfur.dioxide	0.000
density	0.000
pH	0.000
sulphates	0.285
alcohol	0.247

```
lasso_coefs <- data.frame(
  lasso_min = coef(lasso_mod, s = lasso_mod$lambda.min) %>%
    round(3) %>% as.matrix() ,
  lasso_1se = coef(lasso_mod, s = lasso_mod$lambda.1se) %>%
    round(3) %>% as.matrix()
) %>% rename(lasso_min = 1, lasso_1se = 2)

print(lasso_coefs)

lasso_coefs %>%
  select(lasso_min) %>%
  filter(lasso_min != 0) %>%
  nrow()

lasso_coefs %>%
  select(lasso_1se) %>%
  filter(lasso_1se != 0) %>%
  nrow()
```

```
[1] 10
[1] 4
```

10 vs 4 non-zero predictors

Lambda min

	s1
(Intercept)	3.467
fixed.acidity	0.000
volatile.acidity	-1.102
citric.acid	0.000
residual.sugar	0.000
chlorides	-1.742
free.sulfur.dioxide	0.000
total.sulfur.dioxide	-0.002
density	0.000
pH	-0.110
sulphates	0.783
alcohol	0.271

- lasso_1se coefficients have more penalization, are "shrunk" closer to zero
- Therefore, more zero coefficients in the lasso_1se model compared to lasso_min

PREDICTION & EVALUATION

```
#Prediction
predict_train <- predict(lasso_mod, s=lasso_mod$lambda.min,wine_train)
predict_test <- predict(lasso_mod,s=lass0_mod$lambda.min_wine_test)


#Evaluation
results_train <- wine_train %>% mutate(pred=predict_train)
RMSE(results_train$pred,results_train$quality)
[1] 0.6389668
```

- Choosing Lambda:
  - Lasso_min model has smallest value of lambda, therefore lowest training error-->prone to overfitting data
  - Lasso_1se larger value of lambda, higher training error--> may be better for generalization

  - In the context of this Lasso model, an RSME of 0.6389 means that on average, the model's predicted response values are off by approximately 0.6398 units from the actual response values.
    - Pretty good considering the values are relatively small

CONCLUSION

- WE USED RANDOM FOREST AND LASSO REGRESSION MODELS COMPARED TO LINEAR REGRESSION AS OUR BASELINE TO ANALYZE THE DATA AND CONCLUDED THAT ALCOHOL, SULPHATES, AND VOLATILE ACIDITY WERE THE MOST IMPORTANT VARIABLES TO PREDICTING QUALITY OF WINE
- THE USE OF RANDOM FOREST WAS DUE TO ITS EASE OF USE, FLEXIBILITY FOR BOTH CLASSIFICATION AND REGRESSION TREES, AS WELL AS BEING MUCH MORE ACCURATE THAN BAGGING IN THE PREDICTION OF OUR MODELS
- LASSO REGRESSION ALLOWS US TO UNDERSTAND WHICH VARIABLES SHOULD BE CONSIDERED AS SIGNIFICANT TO THE QUALITY OF WINE AND INTERPRET IT EASILY FOR MANAGEMENT OR BUSINESS PURPOSES

• THIS KNOWLEDGE COULD HELP PRODUCERS CREATE HIGHER QUALITY OF WINE AND INCREASE PROFITABILITY FOR THE THEM