Wine Quality Feature Selection & Prediction Project

Autumn Heyman, Bethel Ikejiofor, Erin Weaver, Georgia Miller, Valerie Huston

Here we are taking a look at the quality of Vinho Verde wines within a region and select the more relevant physiochemical features that contribute to wine quality and in which ways. This will be achieved through the use of stepwise binary logistic regression.

Importing Libraries

First we begin by *importing* our libraries.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library("caret")
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library("lmtest")
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library("magrittr")
```

```
##
## Attaching package: 'magrittr'

## The following object is masked from 'package:purrr':
##
##     set_names

## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library("dplyr")
library("tidyr")
library("popbio")
```

```
##
## Attaching package: 'popbio'

## The following object is masked from 'package:caret':
##
##     sensitivity
```

```
library("e1071")
```

Importing our Dataset

Next we *import* our dataset.

```
setwd('/Users/bethelikejiofor/Documents/GitHub/Fab-Five-Final-Project')
wine <- read.csv("./Data/WineQT.csv")

head(wine)
```

```
##   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
##   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
##   quality Id
## 1       5  0
## 2       5  1
## 3       5  2
## 4       6  3
## 5       5  4
## 6       5  5
```

Some Data Wrangling

We beging by reformatting column names so there are no spaces.

```
names(wine) <- str_replace_all(names(wine), c(" "="."))
```

Next, we proceed to drop the ID column since it will not be used in our analysis. We will also take a look again at the head of the dataframe to make sure the wrangling changes took effect.

```
wine = subset(wine, select = -c(Id))
head(wine)
```

```
##   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
##   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
##   quality
## 1       5
## 2       5
## 3       5
## 4       6
## 5       5
## 6       5
```

Assumptions Testing

For this project, rather than taking each of the individual quality levels and doing a logistic regression against them, we will recode the levels so wines either have either ‘good’ or ‘poor’ quality. Wines with a quality between 3 and 5 will fall into the ‘poor’ quality level and those between 6 and 8 will fall into the ‘good’ quality level.

Recoding Wine Quality

```
wine$qualityR <- NA
wine$qualityR[wine$quality==3] <- 0
wine$qualityR[wine$quality==4] <- 0
wine$qualityR[wine$quality==5] <- 0
wine$qualityR[wine$quality==6] <- 1
wine$qualityR[wine$quality==7] <- 1
wine$qualityR[wine$quality==8] <- 1
head(wine)
```

```
##   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
##   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
##   quality qualityR
## 1       5        0
## 2       5        0
## 3       5        0
## 4       6        1
## 5       5        0
## 6       5        0
```

Running the Base Logistic Model

```
mylogit <- glm(qualityR ~ fixed.acidity + volatile.acidity + citric.acid +
                  residual.sugar + chlorides + free.sulfur.dioxide + total.sulfur.dioxide+
                  density + pH + sulphates + alcohol, data = wine, family="binomial")
```

Predicting Wine Quality

```
probabilities <- predict(mylogit, type = "response")
```

Here, I will take the average of the probabilities from the prediction and anything that is above that probability will be classified as a good quality wine and anything below it will be classified as a poor quality wine.

```
avg <- mean(probabilities)
wine$Predicted <- ifelse(probabilities > avg, "good", "poor")
head(wine)
```

```
##   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
##   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
##   quality qualityR Predicted
## 1       5        0      poor
## 2       5        0      poor
## 3       5        0      poor
## 4       6        1      good
## 5       5        0      poor
## 6       5        0      poor
```

Recoding the Predicted Variable

```
wine$PredictedR <- NA
```

```
wine$PredictedR[wine$Predicted=="good"] <- 1
wine$PredictedR[wine$Predicted=="poor"] <- 0
head(wine)
```

```
##   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
##   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
##   quality qualityR Predicted PredictedR
## 1       5        0      poor          0
## 2       5        0      poor          0
## 3       5        0      poor           0
## 4       6        1      good           1
## 5       5        0      poor           0
## 6       5        0      poor           0
```

Converting Variables to Factors

```
wine$PredictedR <- as.factor(wine$PredictedR)
wine$qualityR <- as.factor(wine$qualityR)
head(wine)
```

```
##   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
##   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
##   quality qualityR Predicted PredictedR
## 1       5        0      poor          0
## 2       5        0      poor          0
## 3       5        0      poor          0
## 4       6        1      good          1
## 5       5        0      poor          0
## 6       5        0      poor          0
```

Creating a Confusion Matrix

```
conf_mat <- caret::confusionMatrix(wine$PredictedR, wine$qualityR)
conf_mat
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 410 166
##          1 112 455
##
##                Accuracy : 0.7568
##                  95% CI : (0.7308, 0.7814)
##     No Information Rate : 0.5433
##     P-Value [Acc > NIR] : < 2.2e-16
```

```
##
##                    Kappa : 0.5139
##
##  Mcnemar’s Test P-Value : 0.001479
##
##              Sensitivity : 0.7854
##              Specificity : 0.7327
##           Pos Pred Value : 0.7118
##           Neg Pred Value : 0.8025
##               Prevalence : 0.4567
##           Detection Rate : 0.3587
##     Detection Prevalence : 0.5039
##        Balanced Accuracy : 0.7591
##
##         ’Positive’ Class : 0
##
```

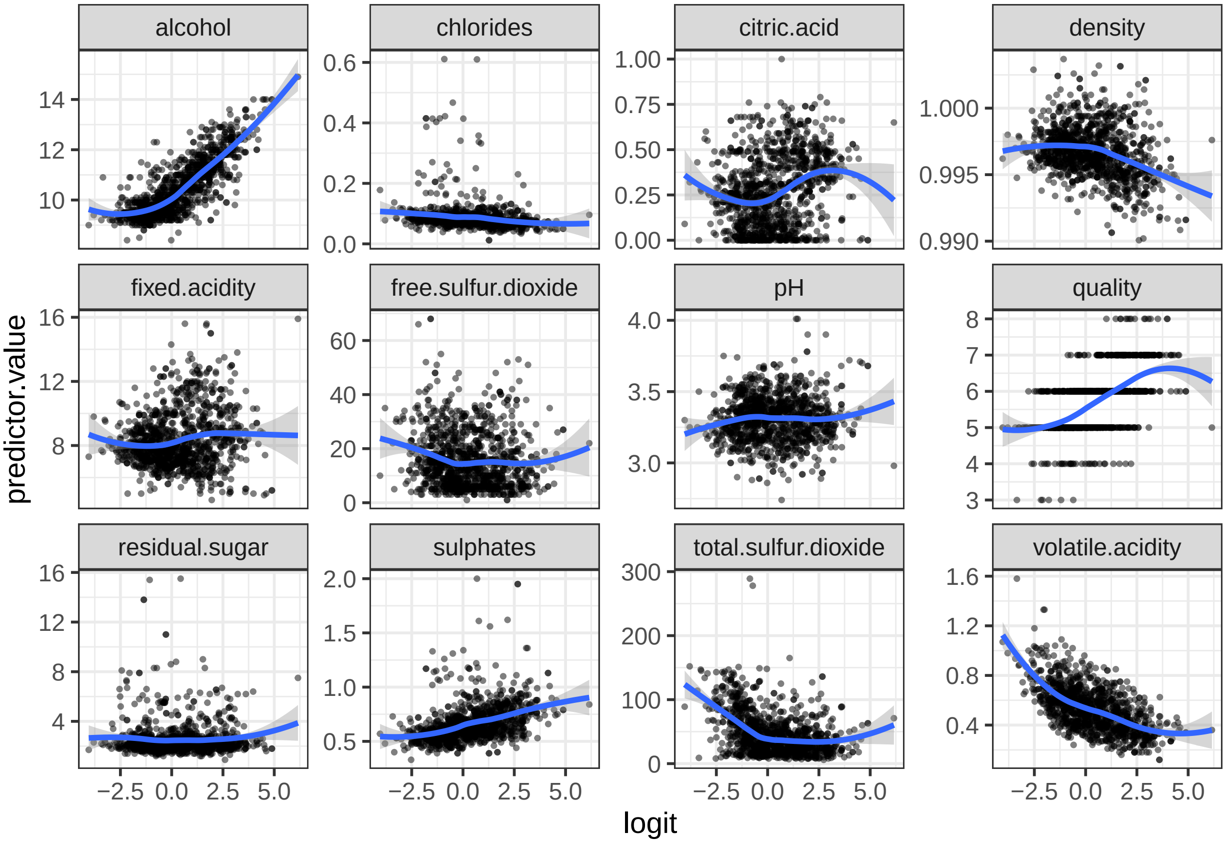
Thankfully, all of the four cells are above 5 so the sample size minimum is met.

Logit Linearity

```
wine1 <- wine %>% dplyr::select_if(is.numeric)
predictors <- colnames(wine1)
wine1 <- wine1 %>% mutate(logit=log(probabilities/(1-probabilities))) %>%
gather(key= "predictors", value="predictor.value", -logit)
```

```
ggplot(wine1, aes(logit, predictor.value))+
geom_point(size=.5, alpha=.5)+
geom_smooth(method= "loess")+
theme_bw()+
facet_wrap(~predictors, scales="free_y")
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



Most of the variables do not seem to have a linear logit relationship with wine quality so the assumption is not met. We will however proceed with our analyses.

Multicollinearity

Insert Valerie's code.