Wine Quality Regression Problem

by Viviane Adohouannon, Kate Alexander, Diana Azbel, Igor Baranov

Abstract We are using a dataset related to red vinho verde wine samples, from the north of Portugal. The goal is to model wine quality based on physicochemical tests. The classes are ordered and not balanced (e.g. there are munch more normal wines than excellent or poor ones). Outlier detection algorithms could be used to detect the few excellent or poor wines. Also, we are not sure if all input variables are relevant. So it could be interesting to test feature selection methods. The method chosen to solve the problem is Linear Regression.

Introduction

Once viewed as a luxury good, nowadays wine is increasingly enjoyed by a all consumers. Portugal is a top ten wine exporting country with 3.17% of the market share in 2005 (FAOSTAT). Exports of its vinho verde wine (from the northwest region) have increased by yearly. To support its growth, the wine industry is investing in new technologies for both wine making and selling processes. Wine certification and quality assessment are key elements within this context. Certification prevents the illegal adulteration of wines (to safeguard human health) and assures quality for the wine market. Quality evaluation is often part of the certification process and can be used to improve wine making (by identifying the most influential factors) and to stratify wines such as premium brands (useful for setting prices). Wine certification is generally assessed by physicochemical and sensory tests (Teranishi et al., 1999). Physicochemical laboratory tests routinely used to characterize wine include determination of density, alcohol or pH values, while sensory tests rely mainly on human experts. It should be stressed that taste is the least understood of the human senses, thus wine classification is a difficult task. Moreover, the relationships between the physicochemical and sensory analysis are complex and still not fully understood (Legin et al., 2003).

Background

The two datasets presented in (UCI Wine Data Set) are related to red and white variants of the Portuguese "Vinho Verde" wine. For more details, consult (Cortez et al., 1998). Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available (e.g. there is no data about grape types, wine brand, wine selling price, etc.).

These datasets can be viewed as classification or regression tasks. The classes are ordered and not balanced (e.g. there are munch more normal wines than excellent or poor ones). Outlier detection algorithms could be used to detect the few excellent or poor wines. Also, we are not sure if all input variables are relevant.

Due to specific purpose of this lab assignment, we are looking at Linear Regression problem only using red wine dataset. Full library of the wine datasets and their description are located here: (UCI Wine Data Set).

Objective

The objective of this article is to provide a reliable and feasible recommendation algorithm to predict wine quality based on physicochemical tests. The target value is a numeric value of wine 'quality', hence the task could be solved by Linear Regression methods. The following 'methodology'check list' standard for a Linear Regression tasks will be applied to the problem at hand:

- Put all relevant variables in the model
- Leave the irrelevant variables out
- Check linearity
- Check regression assumptions:
  - Residuals have a mean of zero
  - Normality of errors
  - Residuals are not autocorrelated
  - Linearity of variables
  - More data than independent variables is used in model building
  - No excessive collinearity

Data understanding

The dataset (UCI Wine Data Set) of red wine quality has 12 attributes and 1599 instances. For more information, read (Cortez et al., 1998). The following is the concept structure of the dataset:

```
Input variables (based on physicochemical tests):

1 - fixed acidity        (FA)
2 - volatile acidity     (VA)
3 - citric acid          (CA)
4 - residual sugar       (RS)
5 - chlorides            (CH)
6 - free sulfur dioxide  (FSD)
7 - total sulfur dioxide (TSD)
8 - density              (DEN)
9 - pH                   (pH)
10 - sulphates           (SUL)
11 - alcohol             (ALC)

Output variable (based on sensory data):
12 - quality (score between 0 and 10) - (QLT)
```

Preparation

To perform the analysis, certain R libraries were used. The code below was used to load and initialize the libraries. The first line invoking seed function was applied to enforce the repeatability of the calculation results.

```
set.seed(42)
library(ggplot2)
library(rpart)
library(rpart.plot)
library(rattle)
library(caret)
```

Reading red wines dataset

The dataset was loaded directly from the site (UCI Wine Data Set) using the R statement below. Note that column names were assigned as the dataset headers had long name making it inconvinient to present it in tables and charts. Corresponence of variable codes to long names is shown in the previous section.

```
library(readr)
wines_red_data <- read.csv(
 "http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv",
 sep=";", header = TRUE,
 col.names = c("FA","VA","CA","RS","CH","FSD","TSD","DEN","pH","SUL","ALC","QLT"))
```

Check for missing values

The dataset has no missing values. Code below calculate number of worw with missing values and checks if there is at list one.

```
any(is.na(wines_red_data))
```

```
#> [1] FALSE
```

Preview of the data

To pretty-print the first 20 rows of the dataset xtable (Dahl, 2016) library was used to generate Table 1.

% latex table generated in R 3.5.1 by xtable 1.8-2 package % Mon Aug 06 23:53:55 2018

	FA	VA	CA	RS	CH	FSD	TSD	DEN	pH	SUL	ALC	QLT
1	7.40	0.70	0.00	1.90	0.08	11.00	34.00	1.00	3.51	0.56	9.40	5
2	7.80	0.88	0.00	2.60	0.10	25.00	67.00	1.00	3.20	0.68	9.80	5
3	7.80	0.76	0.04	2.30	0.09	15.00	54.00	1.00	3.26	0.65	9.80	5
4	11.20	0.28	0.56	1.90	0.07	17.00	60.00	1.00	3.16	0.58	9.80	6
5	7.40	0.70	0.00	1.90	0.08	11.00	34.00	1.00	3.51	0.56	9.40	5
6	7.40	0.66	0.00	1.80	0.07	13.00	40.00	1.00	3.51	0.56	9.40	5
7	7.90	0.60	0.06	1.60	0.07	15.00	59.00	1.00	3.30	0.46	9.40	5
8	7.30	0.65	0.00	1.20	0.06	15.00	21.00	0.99	3.39	0.47	10.00	7
9	7.80	0.58	0.02	2.00	0.07	9.00	18.00	1.00	3.36	0.57	9.50	7
10	7.50	0.50	0.36	6.10	0.07	17.00	102.00	1.00	3.35	0.80	10.50	5
11	6.70	0.58	0.08	1.80	0.10	15.00	65.00	1.00	3.28	0.54	9.20	5
12	7.50	0.50	0.36	6.10	0.07	17.00	102.00	1.00	3.35	0.80	10.50	5
13	5.60	0.61	0.00	1.60	0.09	16.00	59.00	0.99	3.58	0.52	9.90	5
14	7.80	0.61	0.29	1.60	0.11	9.00	29.00	1.00	3.26	1.56	9.10	5
15	8.90	0.62	0.18	3.80	0.18	52.00	145.00	1.00	3.16	0.88	9.20	5
16	8.90	0.62	0.19	3.90	0.17	51.00	148.00	1.00	3.17	0.93	9.20	5
17	8.50	0.28	0.56	1.80	0.09	35.00	103.00	1.00	3.30	0.75	10.50	7
18	8.10	0.56	0.28	1.70	0.37	16.00	56.00	1.00	3.11	1.28	9.30	5
19	7.40	0.59	0.08	4.40	0.09	6.00	29.00	1.00	3.38	0.50	9.00	4
20	7.90	0.32	0.51	1.80	0.34	17.00	56.00	1.00	3.04	1.08	9.20	6

Table 1: `Red Wines Quality Dataset - first 20 rows`

Distribution of target value in the dataset

As we mentione before, the target value QLT of the wine quality is not equaly distributed. The Figure 1 demonstrates the distribution. As we can see, dataset covers medium-quality wines with QLT between 5 and 7 well, low and high quality wines represented poorly.

```
ggplot(data = wines_red_data, mapping = aes(x = QLT)) + geom_bar()
```

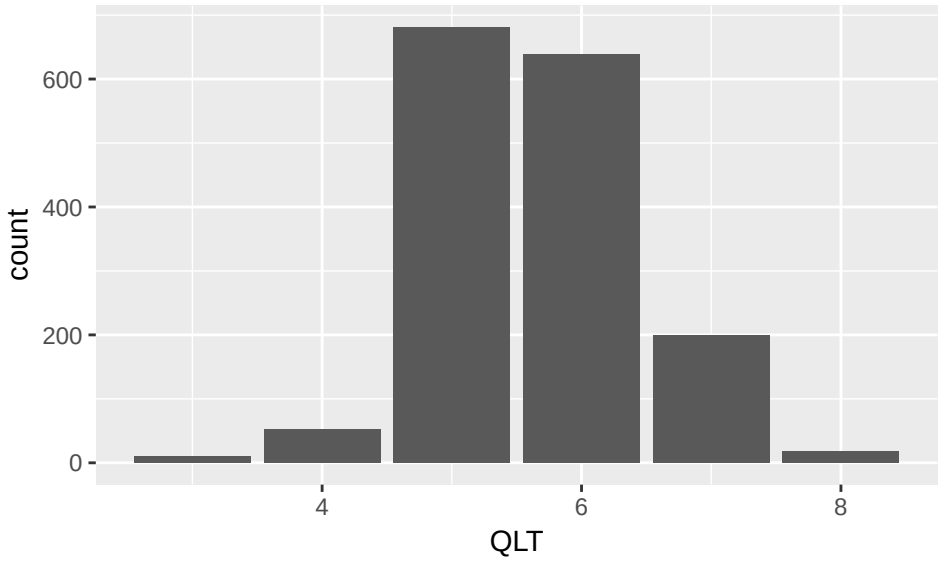

Figure 1: DIstribution of Wine QUality Attribute

OLS Modeling

Default Linear Regression fit

The following code calculates the default OLS model using all the indepedent variables. Results of the calculatiions presented in Table 2.

```
wines_red_data.fit <- lm (QLT ~ ., data=wines_red_data)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	21.9652	21.1946	1.04	0.3002
FA	0.0250	0.0259	0.96	0.3357
VA	-1.0836	0.1211	-8.95	0.0000
CA	-0.1826	0.1472	-1.24	0.2150
RS	0.0163	0.0150	1.09	0.2765
CH	-1.8742	0.4193	-4.47	0.0000
FSD	0.0044	0.0022	2.01	0.0447
TSD	-0.0033	0.0007	-4.48	0.0000
DEN	-17.8812	21.6331	-0.83	0.4086
pH	-0.4137	0.1916	-2.16	0.0310
SUL	0.9163	0.1143	8.01	0.0000
ALC	0.2762	0.0265	10.43	0.0000

Table 2: `Default OLS model - all variables included`

```
summary(wines_red_data.fit)

#>
#> Call:
#> lm(formula = QLT ~ ., data = wines_red_data)
#>
#> Residuals:
#>      Min       1Q   Median       3Q      Max
#> -2.68911 -0.36652 -0.04699  0.45202  2.02498
#>
#> Coefficients:
#>               Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  2.197e+01  2.119e+01   1.036   0.3002
#> FA           2.499e-02  2.595e-02   0.963   0.3357
#> VA          -1.084e+00  1.211e-01  -8.948  < 2e-16 ***
#> CA          -1.826e-01  1.472e-01  -1.240   0.2150
#> RS           1.633e-02  1.500e-02   1.089   0.2765
#> CH          -1.874e+00  4.193e-01  -4.470 8.37e-06 ***
#> FSD          4.361e-03  2.171e-03   2.009   0.0447 *
#> TSD         -3.265e-03  7.287e-04  -4.480 8.00e-06 ***
#> DEN         -1.788e+01  2.163e+01  -0.827   0.4086
#> pH          -4.137e-01  1.916e-01  -2.159   0.0310 *
#> SUL          9.163e-01  1.143e-01   8.014 2.13e-15 ***
#> ALC          2.762e-01  2.648e-02  10.429  < 2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> 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

fit.sum <- summary(wines_red_data.fit)
```

- Multiple R-squared: 0.3605517
- Adjusted R-squared: 0.3561195

More detailed summaries

```
hist(residuals(wines_red_data.fit), xlab = "Residuals", main = "")
```

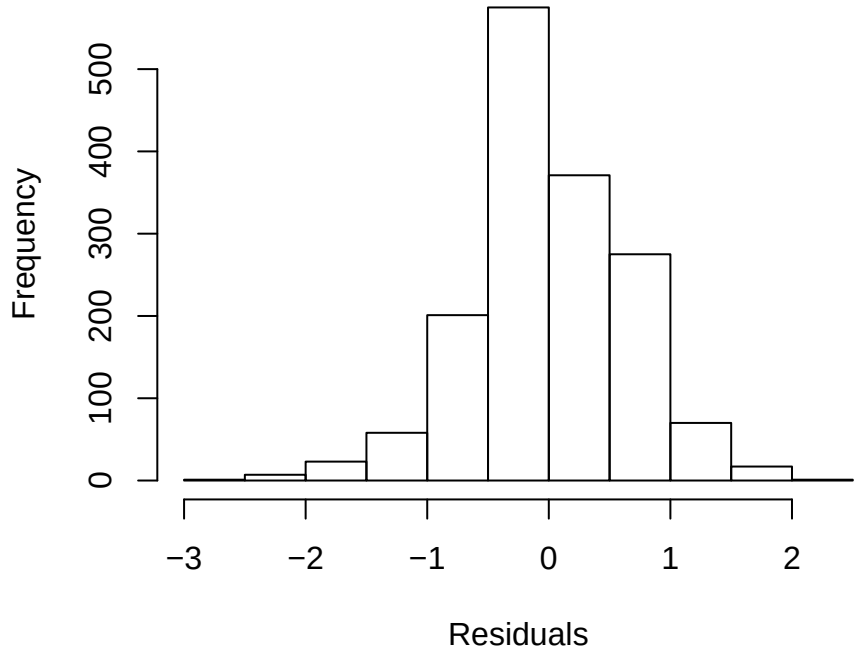

Figure 2: HIstogram of residuals

Adjust the fit removing attributes with p > 0.05

```
wines_red_data.fit1 <- lm (QLT ~ VA + CH + FSD + TSD + pH + SUL + ALC,
                           data=wines_red_data)
summary(wines_red_data.fit1)
```

```
#>
#> Call:
#> lm(formula = QLT ~ VA + CH + FSD + TSD + pH + SUL + ALC, data = wines_red_data)
#>
#> Residuals:
#>      Min       1Q   Median       3Q      Max
#> -2.68918 -0.36757 -0.04653  0.46081  2.02954
#>
#> Coefficients:
#>               Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  4.4300987  0.4029168  10.995  < 2e-16 ***
#> VA          -1.0127527  0.1008429 -10.043  < 2e-16 ***
#> CH          -2.0178138  0.3975417  -5.076 4.31e-07 ***
#> FSD          0.0050774  0.0021255   2.389    0.017 *
#> TSD         -0.0034822  0.0006868  -5.070 4.43e-07 ***
#> pH          -0.4826614  0.1175581  -4.106 4.23e-05 ***
#> SUL          0.8826651  0.1099084   8.031 1.86e-15 ***
#> ALC          0.2893028  0.0167958  17.225  < 2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#>
#> Residual standard error: 0.6477 on 1591 degrees of freedom
#> Multiple R-squared:  0.3595, Adjusted R-squared:  0.3567
#> F-statistic: 127.6 on 7 and 1591 DF,  p-value: < 2.2e-16
```

Stepwise Regression

Find the best model automatically

```
# Stepwise Regression
library(MASS)
fit <- lm(QLT ~ .,data=wines_red_data)
step <- stepAIC(fit, direction="both", trace = FALSE)
step$anova # display results
```

```
#> Stepwise Model Path
#> Analysis of Deviance Table
#>
#> Initial Model:
#> QLT ~ FA + VA + CA + RS + CH + FSD + TSD + DEN + pH + SUL + ALC
#>
#> Final Model:
#> QLT ~ VA + CH + FSD + TSD + pH + SUL + ALC
#>
#>
#>    Step Df  Deviance Resid. Df Resid. Dev       AIC
#> 1                         1587   666.4107 -1375.489
#> 2 - DEN  1 0.2868924      1588   666.6976 -1376.801
#> 3  - FA  1 0.1079824      1589   666.8056 -1378.542
#> 4  - RS  1 0.2566805      1590   667.0623 -1379.926
#> 5  - CA  1 0.4748034      1591   667.5371 -1380.789
```

The model identical to the one found the the previous section

Plot pairwise scatter plots

Pairwise scatter plots 3 to inspect the result for relationships between the independent variable and the numerical dependent variables.

```
attach(wines_red_data)
```

```
#> The following object is masked from package:MASS:
#>
#>     VA
```

```
panel.points<-function(x,y){points(x,y,cex=.1)}
pairs(~QLT + VA + CH + FSD + TSD + pH + SUL + ALC,
      upper.panel=panel.points,lower.panel=panel.points)
```

Checking correlation matrix

Evaluate Nonlinearity component + residual plot

```
library(car)
```

```
#> Loading required package: carData
```

```
crPlots(wines_red_data.fit1, layout = c(4,3), main = "")
```

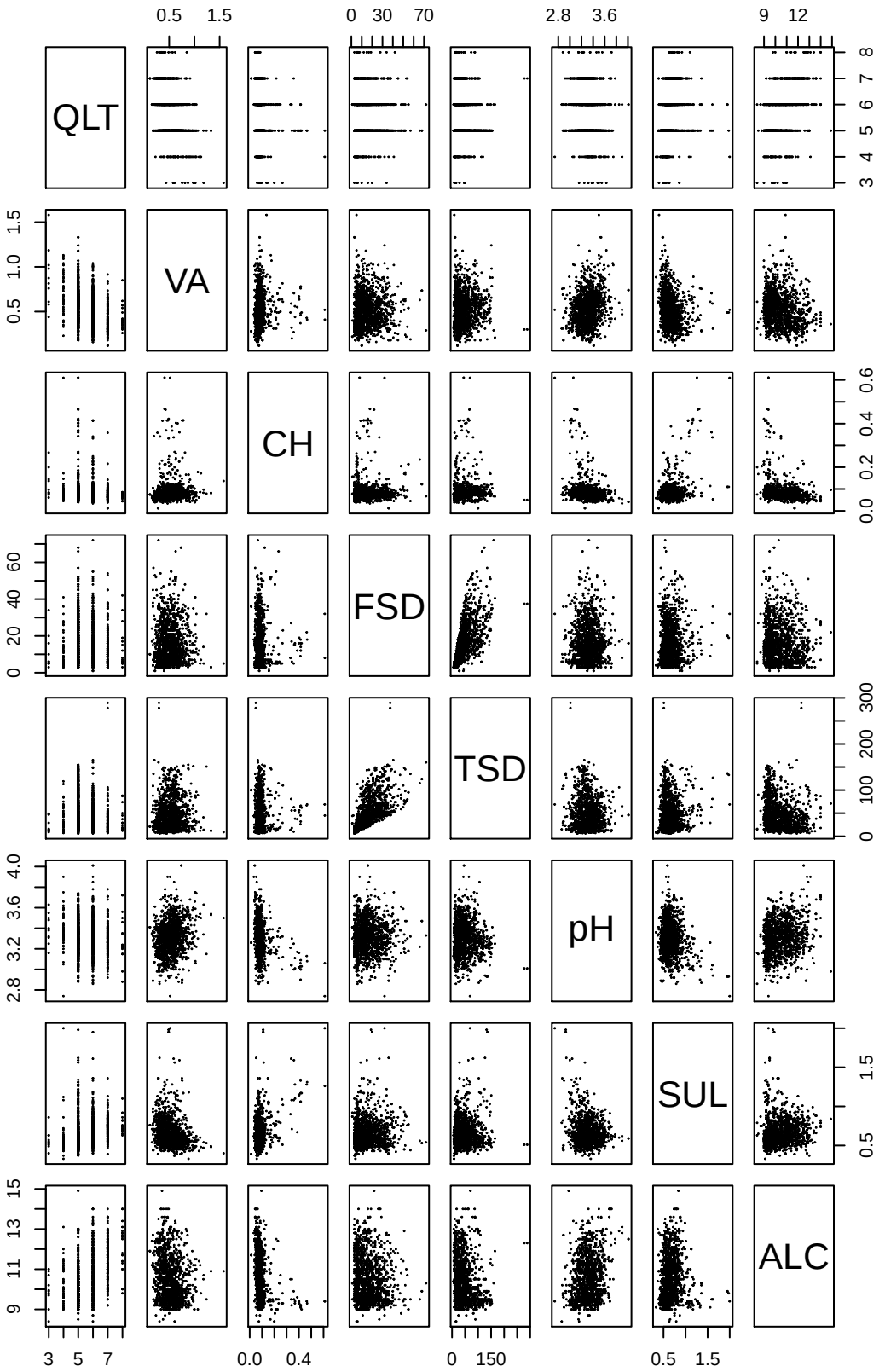

Figure 3: Red Wines - relationships between variables

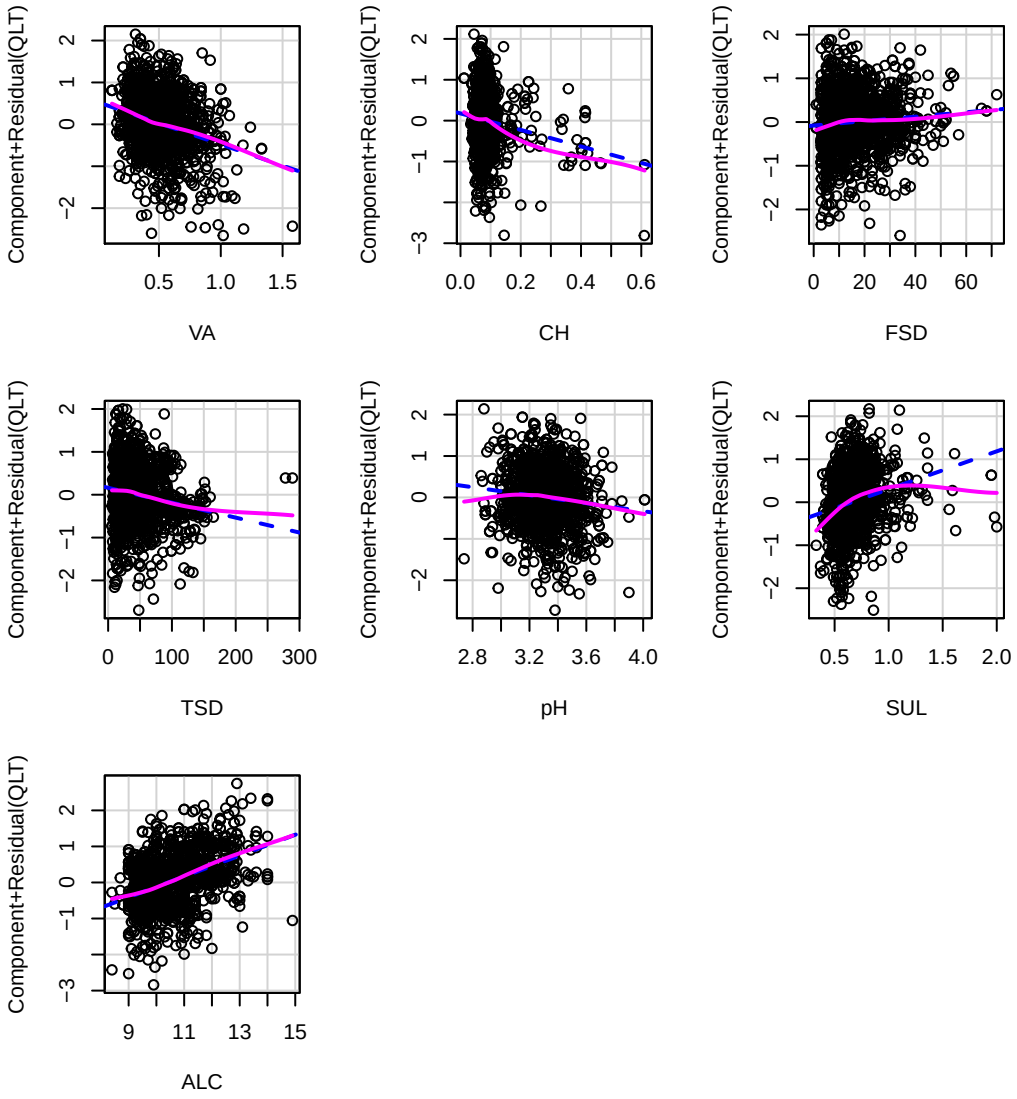

Figure 4: Red Wines - Component + Resudual Plots

	FA	VA	CA	RS	CH	FSD	TSD	DEN	pH	SUL	ALC	QLT
FA	1.00	-0.26	0.67	0.11	0.09	-0.15	-0.11	0.67	-0.68	0.18	-0.06	0.12
VA	-0.26	1.00	-0.55	0.00	0.06	-0.01	0.08	0.02	0.23	-0.26	-0.20	-0.39
CA	0.67	-0.55	1.00	0.14	0.20	-0.06	0.04	0.36	-0.54	0.31	0.11	0.23
RS	0.11	0.00	0.14	1.00	0.06	0.19	0.20	0.36	-0.09	0.01	0.04	0.01
CH	0.09	0.06	0.20	0.06	1.00	0.01	0.05	0.20	-0.27	0.37	-0.22	-0.13
FSD	-0.15	-0.01	-0.06	0.19	0.01	1.00	0.67	-0.02	0.07	0.05	-0.07	-0.05
TSD	-0.11	0.08	0.04	0.20	0.05	0.67	1.00	0.07	-0.07	0.04	-0.21	-0.19
DEN	0.67	0.02	0.36	0.36	0.20	-0.02	0.07	1.00	-0.34	0.15	-0.50	-0.17
pH	-0.68	0.23	-0.54	-0.09	-0.27	0.07	-0.07	-0.34	1.00	-0.20	0.21	-0.06
SUL	0.18	-0.26	0.31	0.01	0.37	0.05	0.04	0.15	-0.20	1.00	0.09	0.25
ALC	-0.06	-0.20	0.11	0.04	-0.22	-0.07	-0.21	-0.50	0.21	0.09	1.00	0.48
QLT	0.12	-0.39	0.23	0.01	-0.13	-0.05	-0.19	-0.17	-0.06	0.25	0.48	1.00

Table 3: Red Wines Quality Dataset Correlation Matrix

Assessing Outliers

```
outlierTest(wines_red_data.fit) # Bonferonni p-value for most extreme obs
```

```
#>      rstudent unadjusted p-value Bonferonni p
#> 833 -4.194088         2.8912e-05     0.046231
```

Both p-value and Bonferroni-corrected p-value are smaller that 0.05, so the model is acceptable.

```
qqPlot(wines_red_data.fit, main="", ylab="Studentized Residuals" )  #qq plot for studentized resid
```

```
#> [1] 653 833
```

A residual plot is a graph that shows the residuals on the vertical axis and the independent variable on the horizontal axis. If the points in a residual plot are randomly dispersed around the horizontal axis, a linear regression model is appropriate for the data; otherwise, a non-linear model is more appropriate. Our model is reasonably dispersed around 0,0 for each of the independent variables

```
# leverage plots
leveragePlots(wines_red_data.fit, layout = c(4,3), main = "")
```

```
# Test for Autocorrelated Errors
durbinWatsonTest(wines_red_data.fit)
```

```
#>  lag Autocorrelation D-W Statistic p-value
#>    1        0.121429       1.75714       0
#>  Alternative hypothesis: rho != 0
```

```
# Global test of model assumptions
library(gvlma)
gvmodel <- gvlma(wines_red_data.fit1)
summary(gvmodel)
```

```
#>
#> Call:
#> lm(formula = QLT ~ VA + CH + FSD + TSD + pH + SUL + ALC, data = wines_red_data)
#>
#> Residuals:
#>      Min       1Q   Median       3Q      Max
#> -2.68918 -0.36757 -0.04653  0.46081  2.02954
#>
#> Coefficients:
#>               Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  4.4300987  0.4029168  10.995  < 2e-16 ***
#> VA          -1.0127527  0.1008429 -10.043  < 2e-16 ***
#> CH          -2.0178138  0.3975417  -5.076 4.31e-07 ***
#> FSD          0.0050774  0.0021255   2.389    0.017 *
#> TSD         -0.0034822  0.0006868  -5.070 4.43e-07 ***
#> pH          -0.4826614  0.1175581  -4.106 4.23e-05 ***
```

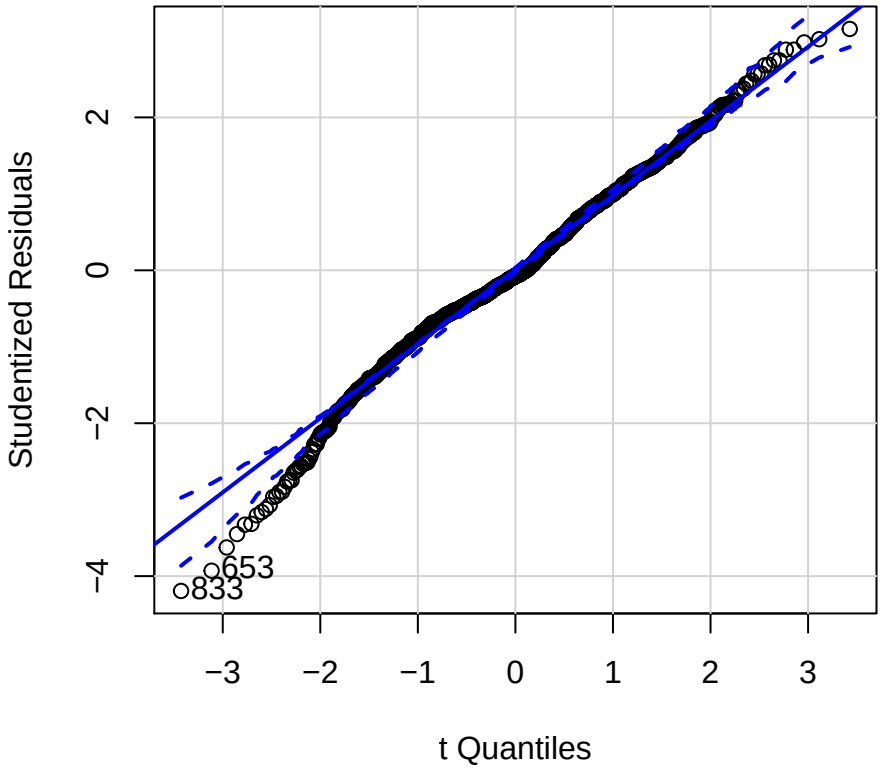

Figure 5: QQ Plot for studentized residuals

```
#> SUL           0.8826651  0.1099084   8.031 1.86e-15 ***
#> ALC           0.2893028  0.0167958  17.225  < 2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.6477 on 1591 degrees of freedom
#> Multiple R-squared:  0.3595, Adjusted R-squared:  0.3567
#> F-statistic: 127.6 on 7 and 1591 DF,  p-value: < 2.2e-16
#>
#>
#> ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
#> USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
#> Level of Significance =  0.05
#>
#> Call:
#>  gvlma(x = wines_red_data.fit1)
#>
#>                      Value   p-value                   Decision
#> Global Stat        37.99895 1.121e-07 Assumptions NOT satisfied!
#> Skewness            6.45922 1.104e-02 Assumptions NOT satisfied!
#> Kurtosis           28.78544 8.086e-08 Assumptions NOT satisfied!
#> Link Function       2.70103 1.003e-01    Assumptions acceptable.
#> Heteroscedasticity  0.05326 8.175e-01    Assumptions acceptable.
```

Addressing Skeweness using log transformation

```
library(car)
summary(wines_red_data.fit2 <- lm (
  bcPower(QLT,1.25)  ~ VA + CH + FSD + TSD + pH + SUL + ALC,
  data=wines_red_data))
```

```
#>
#> Call:
#> lm(formula = bcPower(QLT, 1.25) ~ VA + CH + FSD + TSD + pH +
#>     SUL + ALC, data = wines_red_data)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -3.8872 -0.5845 -0.0839  0.7095  3.2585
#>
#> Coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  4.258050   0.620654   6.861 9.79e-12 ***
#> VA          -1.538767   0.155339  -9.906  < 2e-16 ***
#> CH          -3.112317   0.612374  -5.082 4.17e-07 ***
#> FSD          0.007722   0.003274   2.358   0.0185 *
#> TSD         -0.005436   0.001058  -5.139 3.11e-07 ***
#> pH          -0.751939   0.181087  -4.152 3.46e-05 ***
#> SUL          1.372027   0.169303   8.104 1.05e-15 ***
#> ALC          0.452131   0.025872  17.475  < 2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.9978 on 1591 degrees of freedom
#> Multiple R-squared:  0.3625, Adjusted R-squared:  0.3597
#> F-statistic: 129.2 on 7 and 1591 DF,  p-value: < 2.2e-16
```

```
gvmodel <- gvlma(wines_red_data.fit2)
summary(gvmodel)
```

```
#>
#> Call:
#> lm(formula = bcPower(QLT, 1.25) ~ VA + CH + FSD + TSD + pH +
#>     SUL + ALC, data = wines_red_data)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -3.8872 -0.5845 -0.0839  0.7095  3.2585
#>
#> Coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  4.258050   0.620654   6.861 9.79e-12 ***
#> VA          -1.538767   0.155339  -9.906  < 2e-16 ***
#> CH          -3.112317   0.612374  -5.082 4.17e-07 ***
#> FSD          0.007722   0.003274   2.358   0.0185 *
#> TSD         -0.005436   0.001058  -5.139 3.11e-07 ***
#> pH          -0.751939   0.181087  -4.152 3.46e-05 ***
#> SUL          1.372027   0.169303   8.104 1.05e-15 ***
#> ALC          0.452131   0.025872  17.475  < 2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.9978 on 1591 degrees of freedom
#> Multiple R-squared:  0.3625, Adjusted R-squared:  0.3597
#> F-statistic: 129.2 on 7 and 1591 DF,  p-value: < 2.2e-16
#>
#>
#> ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
#> USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
```

```
#> Level of Significance =  0.05
#>
#> Call:
#>  gvlma(x = wines_red_data.fit2)
#>
#>                      Value   p-value                   Decision
#> Global Stat        24.80148 5.515e-05 Assumptions NOT satisfied!
#> Skewness            0.50847 4.758e-01    Assumptions acceptable.
#> Kurtosis           20.50362 5.952e-06 Assumptions NOT satisfied!
#> Link Function       3.77416 5.205e-02    Assumptions acceptable.
#> Heteroscedasticity  0.01524 9.018e-01    Assumptions acceptable.

hist(residuals(wines_red_data.fit2), xlab = "Residuals", main = "")
```

Tree-based regression methods

Tree-based methods, while simple and useful for interpretation, are typically not as competitive with the best supervised learning approaches such as polynomial regression. However, tree-based methods such as regression tree and random forests make up for this shortfall. By combining a large number of trees instead of one, the model usually results in dramatic improvements in terms of prediction accuracy. This improvement in accuracy comes at the expense of loss in intepretation.

Splitting the dataset into train and test

The dataset has been split in such a way that train and test sets would have the same distribution of the 'QLT' attribute. The reason for this stratification strategy is to focus on the priority on the target value. We used 60:34 split ratio.

```
library(caret)
train.rows<- createDataPartition(y= wines_red_data$QLT, p=0.6, list = FALSE)
train.data<- wines_red_data[train.rows,]
prop.table((table(train.data$QLT)))
```

```
#>
#>           3           4           5           6           7           8
#> 0.005202914 0.029136316 0.430801249 0.398543184 0.124869927 0.011446410
```

```
test.data<- wines_red_data[-train.rows,]
prop.table((table(test.data$QLT)))
```

```
#>
#>           3           4           5           6           7           8
#> 0.007836991 0.039184953 0.418495298 0.399686520 0.123824451 0.010971787
```

Regression tree fit

In a regression tree, the tree arranges or segments observations into regions of a predictor space. For example, using the "Hitters" data set, which contains various statistics on baseball players, a tree might look something like in Figure 8 generated by the code below.

```
library(rpart)
library(rpart.plot)
library(rattle)
library(caret)

reg.tree <- rpart(QLT ~ ., method="anova", data = train.data)
fancyRpartPlot(reg.tree, main="", sub="")


# reg.tree$variable.importance
```

Regression Tree model evaluation

This Decision Tree favours the following attributes in order of their importance for the prediction of the target attribute: health, has_nurs, parents. It does not consider the rest of the attributes as important. Let's apply the model to the test set and evaluate accuracy of the predictions.

```
dtPrediction <- predict(reg.tree, test.data)
cor(dtPrediction,test.data$QLT)
```

```
#> [1] 0.5480833
```

```
plot(jitter(test.data$QLT), dtPrediction,
     pch=20, col=rgb(0.1, 0.2, 0.8, 0.3),
     ylab="Prediction", xlab="Test Values", bty="n" )
```

```
qqPlot(dtPrediction, main="" )
```

```
#> 127 128
#>  44  45
```

Random Forest model fit

```
library(randomForest)
fitRF1 <- randomForest(
  QLT ~ ., method="anova",
  data=train.data, importance=TRUE, ntree=2000)
```

```
varImpPlot(fitRF1, main="")
```

Random Forest model prediction and evaluation

```
PredictionRF1 <- predict(fitRF1, test.data)
cor(PredictionRF1,test.data$QLT)
```

```
#> [1] 0.6748129
```

```
plot(jitter(test.data$QLT), PredictionRF1 ,
     pch=20, col=rgb(0.1, 0.2, 0.8, 0.3),
     ylab="Prediction", xlab="Test Values", bty="n" )
```

```
qqPlot(PredictionRF1, main="")
```

```
#> 199 128
#>  75  45
```

Conclusion

Bibliography

P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 1998. [p1, 2]

D. B. Dahl. *xtable: Export Tables to LaTeX or HTML*, 2016. URL https://CRAN.R-project.org/package=xtable. R package version 1.8-2. [p3]

FAOSTAT. Faostat. URL http://faostat.fao.org/site/535/DesktopDefault.aspx?PageID=535. [p1]

A. Legin, A. Rudnitskaya, L. Lvova, Y. Vlasov, C. Di Natale, and A. D'Amico. Evaluation of Italian wine by the electronic tongue: recognition, quantitative analysis and correlation with human sensory perception. *Analytica Chimica Acta*, 484(1):33–44, May 2003. ISSN 00032670. doi: 10.1016/S0003-2670(03)00301-5. URL http://linkinghub.elsevier.com/retrieve/pii/S0003267003003015. [p1]

R. Teranishi, E. L. Wick, and I. Hornstein, editors. *Flavor chemistry: thirty years of progress*. Kluwer Academic/Plenum Publishers, New York, 1999. ISBN 9780306461996. [p1]

UCI Wine Data Set. Uci machine learning repository: wine quality data set. URL http://archive.ics.uci.edu/ml/datasets/Wine+Quality. [p1, 2]

Note from the Authors

This file was generated using *The R Journal* style article template, additional information on how to prepare articles for submission is here - Instructions for Authors. The article itself is an executable R Markdown file that could be downloaded from Github with all the necessary artifacts.

Viviane Adohouannon
York University School of Continuing Studies

https://learn.continue.yorku.ca/user/view.php?id=21444

Kate Alexander
York University School of Continuing Studies

https://learn.continue.yorku.ca/user/view.php?id=21524

Diana Azbel
York University School of Continuing Studies

https://learn.continue.yorku.ca/user/view.php?id=20687

Igor Baranov
York University School of Continuing Studies

https://learn.continue.yorku.ca/user/profile.php?id=21219

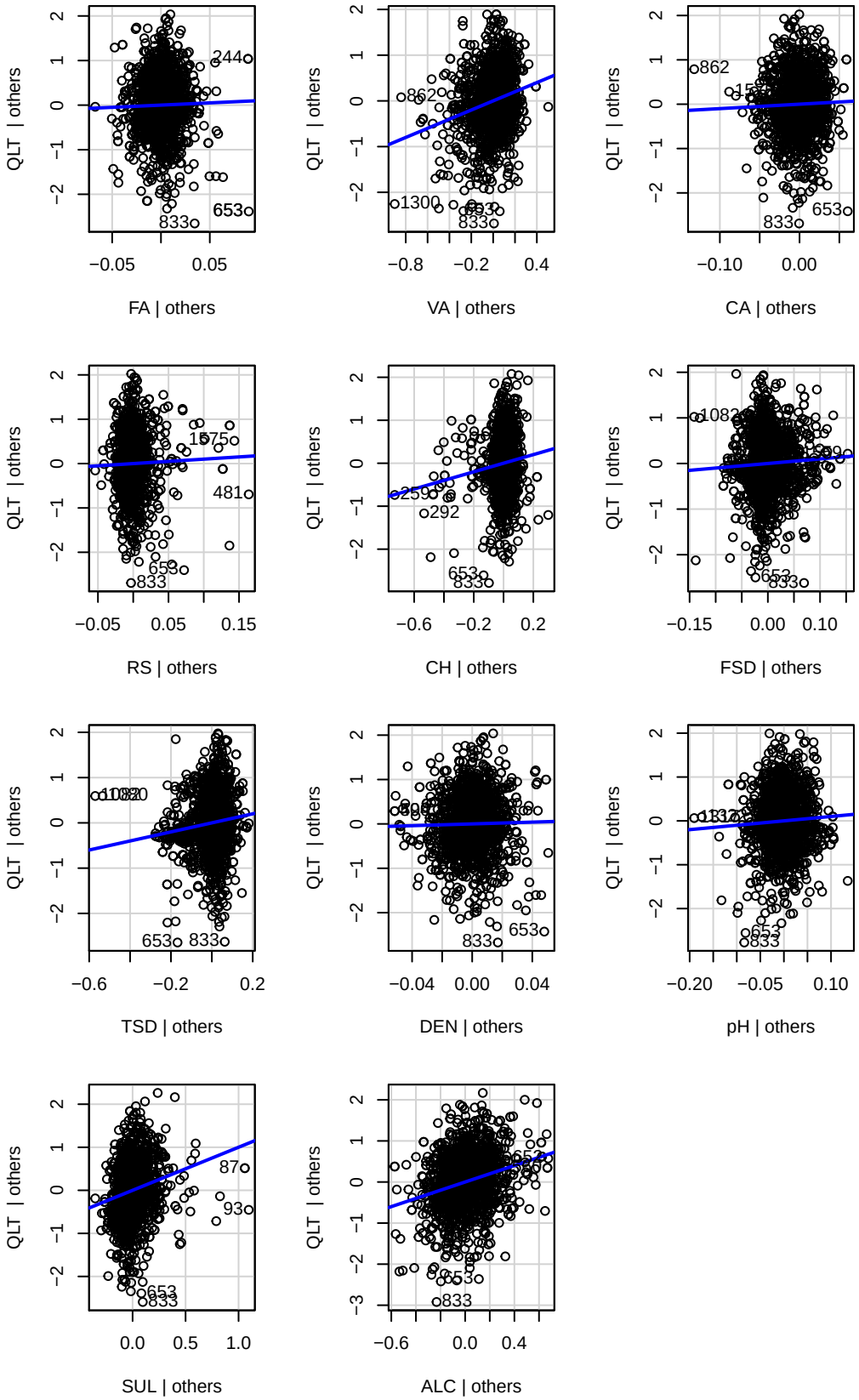

Figure 6: Red Wines - Leverage Plots

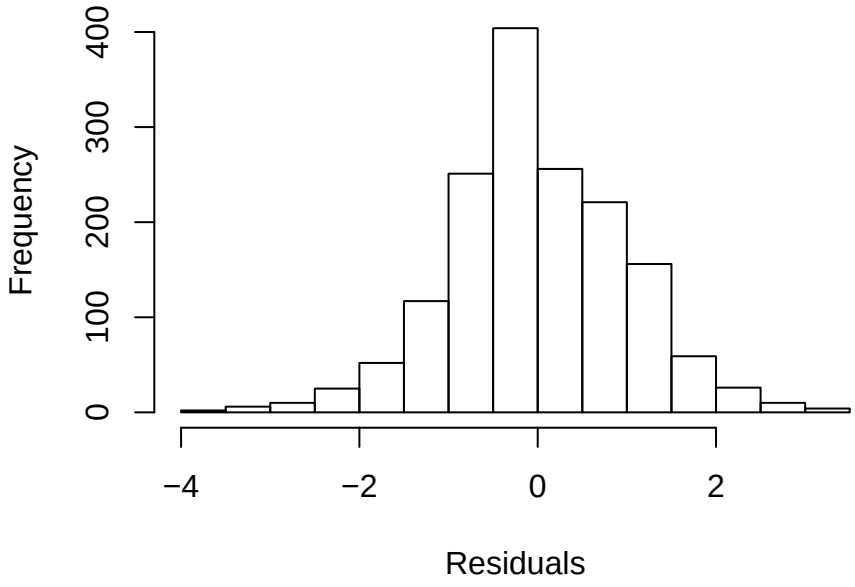

Figure 7: HIstogram of residuals after correcting the Skewness

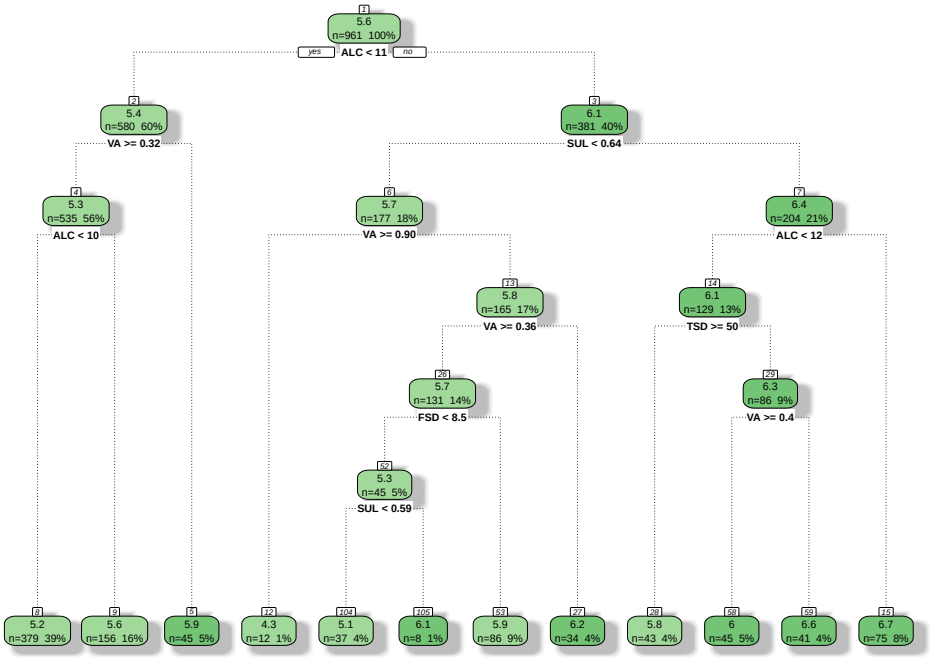

Figure 8: Regression Tree Diagram

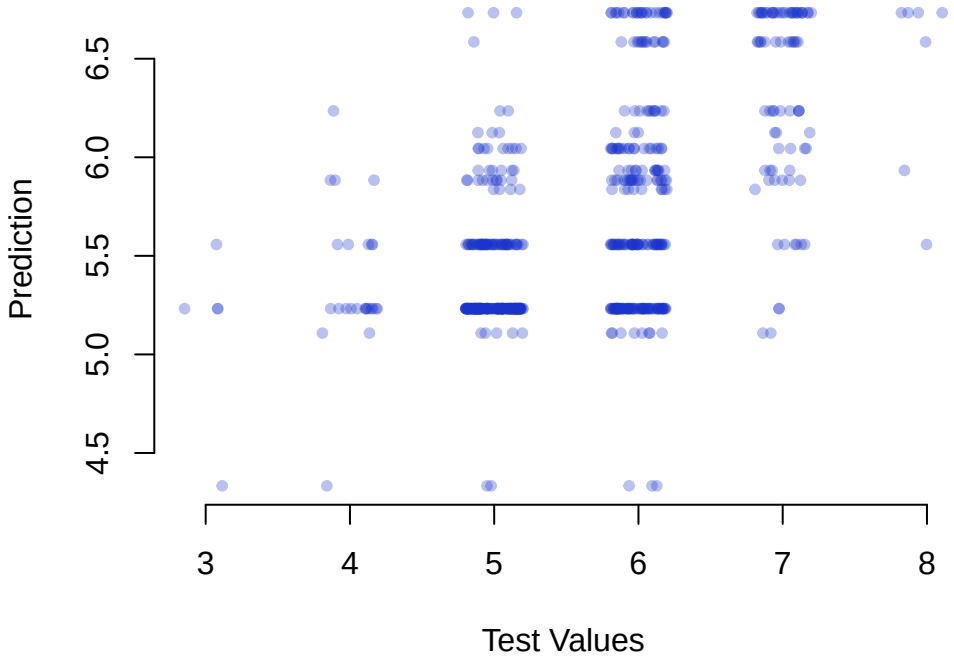

Figure 9: Regression Tree Prediction

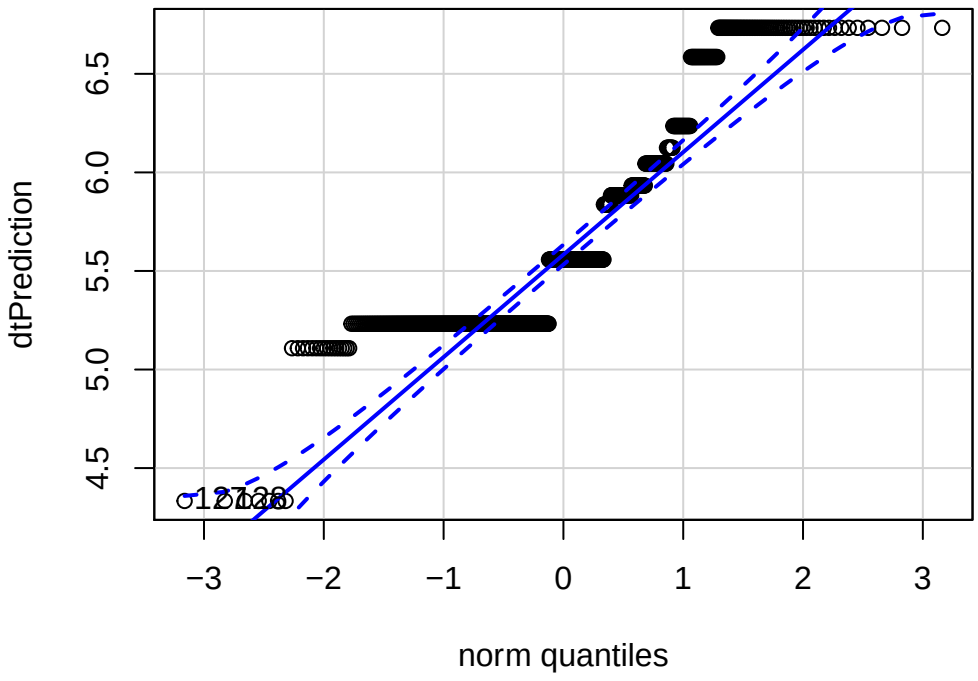

Figure 10: Regression Tree Prediction QQ Plot

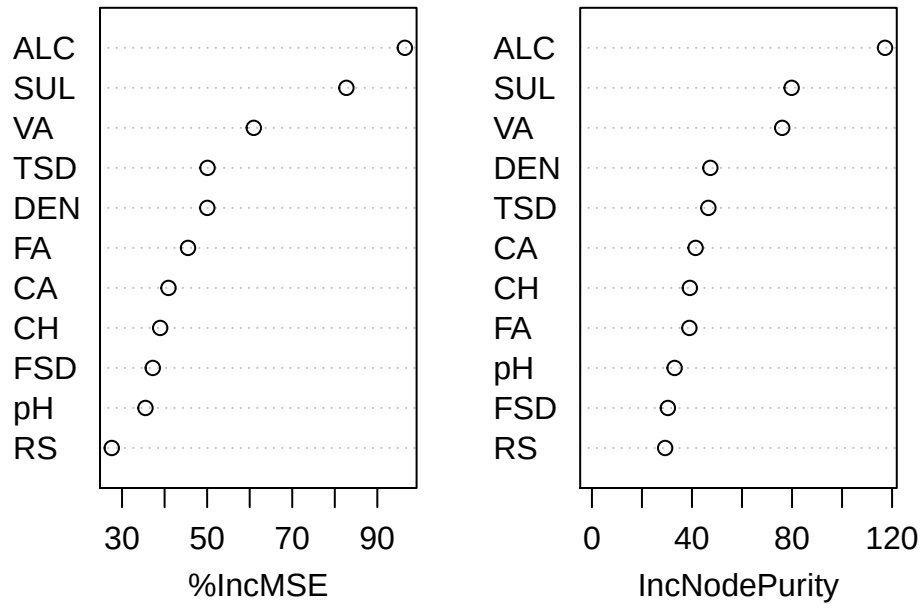

Figure 11: Importance of the dataset attributes for the prediction of the 'class' attribute

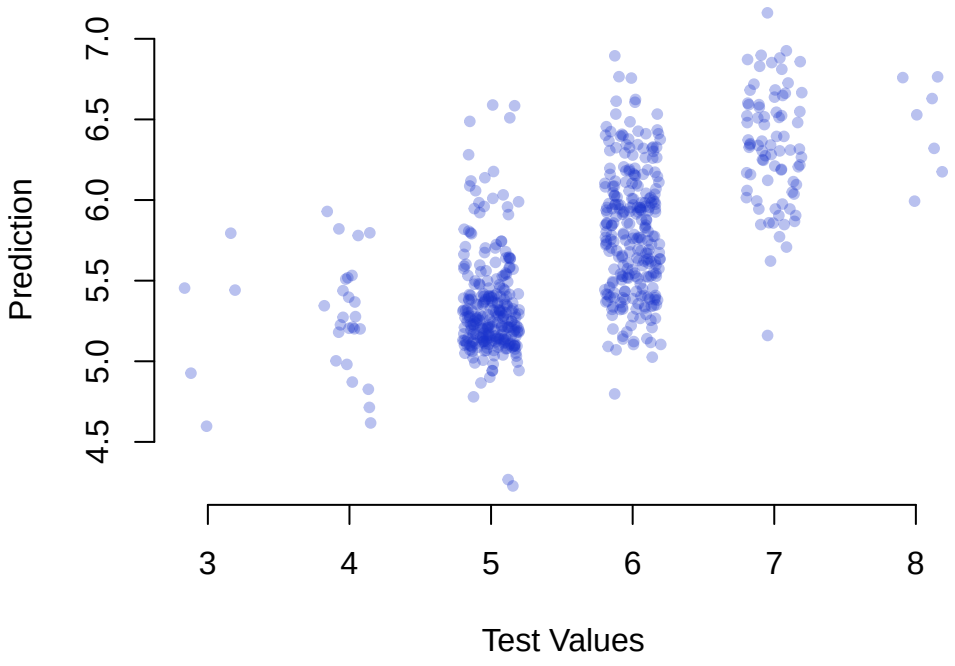

Figure 12: Random Forest Prediction

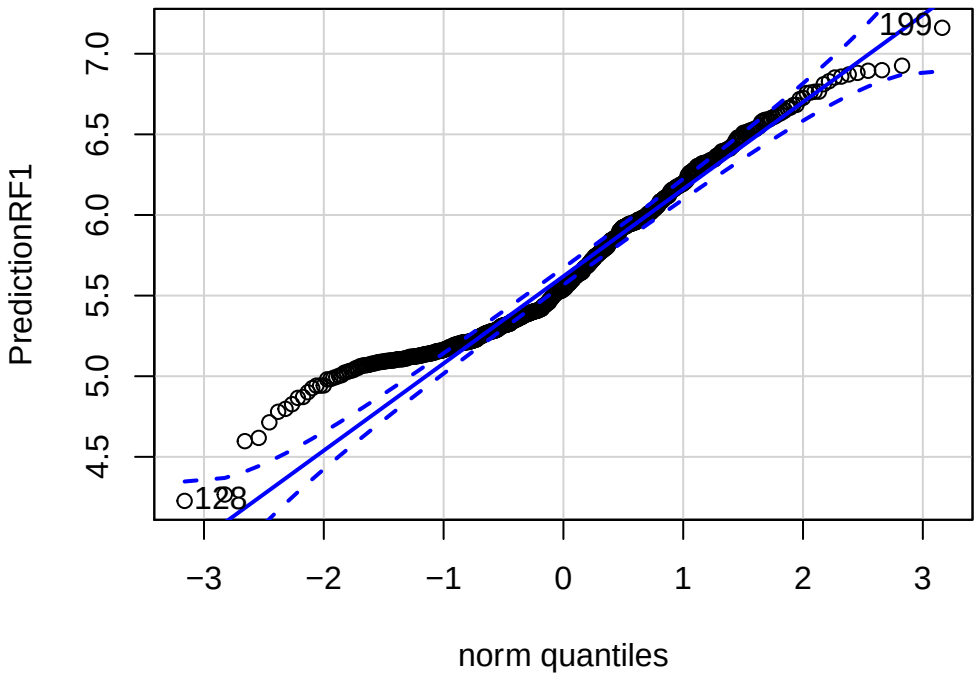

Figure 13: Random Forest Prediction QQ Plot