Assign4

Praveen

5/8/2021

Data Set Information:

The two datasets are related to red and white variants of the Portuguese “Vinho Verde” wine. For more details, consult: [Web Link] or the reference [Cortez et al., 2009]. 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 many 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.

Attribute Information:

For more information, read [Cortez et al., 2009]. Input variables (based on physicochemical tests): 1 - fixed acidity 2 - volatile acidity 3 - citric acid 4 - residual sugar 5 - chlorides 6 - free sulfur dioxide 7 - total sulfur dioxide 8 - density 9 - pH 10 - sulphates 11 - alcohol Output variable (based on sensory data): 12 - quality (score between 0 and 10)

The research was about finding the quality of wine?

The file got stored is CSV format.It is delimited and Comma-separated values the file is binary opned by Ms excel program in general.

data <- read.csv("winequality-white.csv",header = T, sep = ";")  
n = nrow(data); p = ncol(data); dim(data)

## [1] 4898 12

any(is.na(data))

## [1] FALSE

library(psych)  
summary(data)

## fixed.acidity volatile.acidity citric.acid residual.sugar   
## Min. : 3.800 Min. :0.0800 Min. :0.0000 Min. : 0.600   
## 1st Qu.: 6.300 1st Qu.:0.2100 1st Qu.:0.2700 1st Qu.: 1.700   
## Median : 6.800 Median :0.2600 Median :0.3200 Median : 5.200   
## Mean : 6.855 Mean :0.2782 Mean :0.3342 Mean : 6.391   
## 3rd Qu.: 7.300 3rd Qu.:0.3200 3rd Qu.:0.3900 3rd Qu.: 9.900   
## Max. :14.200 Max. :1.1000 Max. :1.6600 Max. :65.800   
## chlorides free.sulfur.dioxide total.sulfur.dioxide density   
## Min. :0.00900 Min. : 2.00 Min. : 9.0 Min. :0.9871   
## 1st Qu.:0.03600 1st Qu.: 23.00 1st Qu.:108.0 1st Qu.:0.9917   
## Median :0.04300 Median : 34.00 Median :134.0 Median :0.9937   
## Mean :0.04577 Mean : 35.31 Mean :138.4 Mean :0.9940   
## 3rd Qu.:0.05000 3rd Qu.: 46.00 3rd Qu.:167.0 3rd Qu.:0.9961   
## Max. :0.34600 Max. :289.00 Max. :440.0 Max. :1.0390   
## pH sulphates alcohol quality   
## Min. :2.720 Min. :0.2200 Min. : 8.00 Min. :3.000   
## 1st Qu.:3.090 1st Qu.:0.4100 1st Qu.: 9.50 1st Qu.:5.000   
## Median :3.180 Median :0.4700 Median :10.40 Median :6.000   
## Mean :3.188 Mean :0.4898 Mean :10.51 Mean :5.878   
## 3rd Qu.:3.280 3rd Qu.:0.5500 3rd Qu.:11.40 3rd Qu.:6.000   
## Max. :3.820 Max. :1.0800 Max. :14.20 Max. :9.000

It seems that the dataset is very clean, with no missing data and clear structure. All variables are numeric. The range of independent variables varies greatly, so when building the model I will normalize them to be within the same range.

colnames(data)

## [1] "fixed.acidity" "volatile.acidity" "citric.acid"   
## [4] "residual.sugar" "chlorides" "free.sulfur.dioxide"   
## [7] "total.sulfur.dioxide" "density" "pH"   
## [10] "sulphates" "alcohol" "quality"

I will split the dataset into a training and testing set, and normalize each set separately.

set.seed(1)  
idx = sample(n, 0.9\*n)  
train = data[idx,]; dim(train)

## [1] 4408 12

test = data[-idx,]; dim(test)

## [1] 490 12

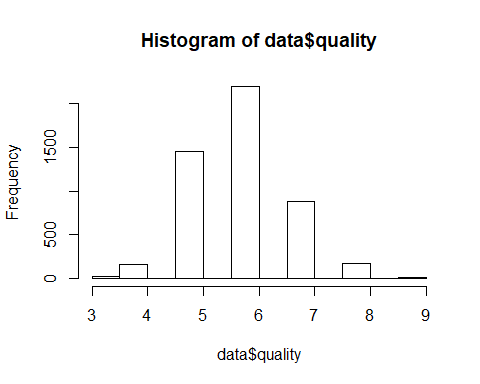
# normalize train set so that the range is 0 ~ 1  
normalize\_train = function(x) (x - min(x))/(max(x) - min(x))  
train.norm = data.frame(apply(train[,-p], 2, normalize\_train),   
 quality = train[,p])  
summary(train.norm)

## fixed.acidity volatile.acidity citric.acid residual.sugar   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.00000   
## 1st Qu.:0.2404 1st Qu.:0.1275 1st Qu.:0.1627 1st Qu.:0.01687   
## Median :0.2885 Median :0.1765 Median :0.1928 Median :0.06902   
## Mean :0.2936 Mean :0.1950 Mean :0.2012 Mean :0.08835   
## 3rd Qu.:0.3365 3rd Qu.:0.2353 3rd Qu.:0.2304 3rd Qu.:0.14110   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.00000   
## chlorides free.sulfur.dioxide total.sulfur.dioxide density   
## Min. :0.00000 Min. :0.00000 Min. :0.0000 Min. :0.00000   
## 1st Qu.:0.08012 1st Qu.:0.07317 1st Qu.:0.2279 1st Qu.:0.08828   
## Median :0.10089 Median :0.11150 Median :0.2884 Median :0.12671   
## Mean :0.10955 Mean :0.11601 Mean :0.2983 Mean :0.13281   
## 3rd Qu.:0.12166 3rd Qu.:0.15331 3rd Qu.:0.3651 3rd Qu.:0.17300   
## Max. :1.00000 Max. :1.00000 Max. :1.0000 Max. :1.00000   
## pH sulphates alcohol quality   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :3.000   
## 1st Qu.:0.3394 1st Qu.:0.2209 1st Qu.:0.2419 1st Qu.:5.000   
## Median :0.4220 Median :0.2907 Median :0.3871 Median :6.000   
## Mean :0.4309 Mean :0.3146 Mean :0.4060 Mean :5.877   
## 3rd Qu.:0.5138 3rd Qu.:0.3837 3rd Qu.:0.5484 3rd Qu.:6.000   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :9.000

# normalize test set using the values from train set to make prediction comparable  
train.min = apply(train[,-p], 2, min)  
train.max = apply(train[,-p], 2, max)  
test.norm = data.frame(sweep(test, 2, c(train.min, 0)) %>%   
 sweep(2, c(train.max-train.min, 1), FUN = "/"))  
summary(test.norm) # test.norm might have data out of range 0~1, since it's normalized against the training set.

## fixed.acidity volatile.acidity citric.acid residual.sugar   
## Min. :0.08654 Min. :0.0000 Min. :0.0000 Min. :0.00000   
## 1st Qu.:0.24038 1st Qu.:0.1275 1st Qu.:0.1627 1st Qu.:0.02090   
## Median :0.28846 Median :0.1765 Median :0.1867 Median :0.08129   
## Mean :0.29507 Mean :0.1886 Mean :0.2028 Mean :0.09309   
## 3rd Qu.:0.33654 3rd Qu.:0.2353 3rd Qu.:0.2349 3rd Qu.:0.14992   
## Max. :0.52885 Max. :0.6471 Max. :0.6024 Max. :0.30828   
## chlorides free.sulfur.dioxide total.sulfur.dioxide  
## Min. :0.01484 Min. :0.003484 Min. :-0.002326   
## 1st Qu.:0.07715 1st Qu.:0.076655 1st Qu.: 0.230233   
## Median :0.09792 Median :0.114983 Median : 0.288372   
## Mean :0.10524 Mean :0.116465 Mean : 0.300384   
## 3rd Qu.:0.12166 3rd Qu.:0.156794 3rd Qu.: 0.369186   
## Max. :0.59050 Max. :0.264808 Max. : 0.776744   
## density pH sulphates alcohol   
## Min. :-0.0003857 Min. :0.07339 Min. :0.01163 Min. :0.08065   
## 1st Qu.: 0.0900675 1st Qu.:0.33028 1st Qu.:0.22093 1st Qu.:0.22581   
## Median : 0.1305689 Median :0.40367 Median :0.30233 Median :0.37097   
## Mean : 0.1349538 Mean :0.41758 Mean :0.30662 Mean :0.40097   
## 3rd Qu.: 0.1758920 3rd Qu.:0.49541 3rd Qu.:0.37209 3rd Qu.:0.54839   
## Max. : 0.2694311 Max. :1.00917 Max. :0.79070 Max. :0.95161   
## quality   
## Min. :4.000   
## 1st Qu.:5.000   
## Median :6.000   
## Mean :5.882   
## 3rd Qu.:6.000   
## Max. :9.000

hist(data$quality)



shapiro.test(data$quality) #Didn't pass normality test, so linear model may have a problem

##   
## Shapiro-Wilk normality test  
##   
## data: data$quality  
## W = 0.88904, p-value < 2.2e-16

The dependent variable doesn’t pass the normality test, so one assumption of linear regression is not met. In addition, as we see from the pairwise plot, the relationship among independent variables and dependent variables are not entirely linear. There is also some collinearity among independent variables. Any of those could sabotage the performance of the linear model.

Then I will apply this linear model to the test set, and visualize the predicted value against the true value. I will also evaluate the model performance based on 3 measures: RMSE (root mean square error), MAE (mean absolute error) and cor (correlation). Smaller RMSE, MAE and larger cor are indicators of a good prediction.

tr.lm = lm(quality~., data = train.norm)  
summary(tr.lm)

##   
## Call:  
## lm(formula = quality ~ ., data = train.norm)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.7925 -0.4919 -0.0371 0.4656 3.1229   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.59161 0.11255 49.681 < 2e-16 \*\*\*  
## fixed.acidity 0.67810 0.22711 2.986 0.00284 \*\*   
## volatile.acidity -1.86362 0.12259 -15.202 < 2e-16 \*\*\*  
## citric.acid 0.04456 0.16881 0.264 0.79181   
## residual.sugar 5.42949 0.51218 10.601 < 2e-16 \*\*\*  
## chlorides -0.07964 0.19115 -0.417 0.67695   
## free.sulfur.dioxide 1.03348 0.25334 4.079 4.60e-05 \*\*\*  
## total.sulfur.dioxide -0.14411 0.17204 -0.838 0.40227   
## density -8.10484 1.02681 -7.893 3.69e-15 \*\*\*  
## pH 0.77261 0.12059 6.407 1.64e-10 \*\*\*  
## sulphates 0.55605 0.09086 6.120 1.02e-09 \*\*\*  
## alcohol 1.13760 0.15612 7.287 3.75e-13 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.7536 on 4396 degrees of freedom  
## Multiple R-squared: 0.281, Adjusted R-squared: 0.2792   
## F-statistic: 156.2 on 11 and 4396 DF, p-value: < 2.2e-16