Projeto 3 - Dataset Qualidade Vinho

Marcos de Morais

31/07/2022

1 - Introdução

Objetivo do projeto

Criar um modelo preditivo capaz de prever a qualidade de um determinado vinho.

Sobre o dataset

O dados foram obtidos a partir de testes feitos em amostras, como o pH por exemplo, e com base na avaliação de experts em vinho para determinar a sua qualidade. Feito com uma base em média de 3 avaliações.

Os dois conjuntos de dados estão relacionados com variantes tinto e branco do vinho português "Vinho Verde". Para mais detalhes, consulte: [Web Link] ou a referência [Cortez et al., 2009]. Por questões de privacidade e logística, apenas variáveis físico-químicas (entradas) e sensoriais (saídas) estão disponíveis (por exemplo, não há dados sobre tipos de uva, marca de vinho, preço de venda do vinho, etc.).

Números de registros

Vinho tinto - 1.599 registros Vinho branco - 4.898 registros

Número de atributos

11 + variável target (output)

Dicionário de Dados

- 1 - fixed acidity(g/dmˆ3): Ácido tartático, confere gosto amargo e encontrado em sendimentos de vinhos
- 2 - volatile acidity(g/dmˆ3): O ácido acético é um ácido orgânico que se forma durante a fermentação alcoólica de forma natural. Característica essencial dos vinhos, contribui decisivamente para o seu sabor, frescura, equilíbrio e capacidade de conservação
- 3 - citric acid(g/dmˆ3): A acidez de um vinho é essencial para identificar o aroma e sabor, contribuindo para sua conservação e envelhecimento
- 4 - residual sugar(g/dmˆ3): Sobra de açucar resultante da fermentação das uvas
- 5 - chlorides(g/dmˆ3): Atua no gosto do vinho, teor muito alto resulta em um gosto mais salgado
- 6 - free sulfur dioxide (g/dmˆ3): Atua na qualidade do vinho para um melhor processo de fermentação aumentando a qualidade geral e longeividade do vinho
- 7 - total sulfur dioxide (g/dmˆ3): Atua na qualidade do vinho para um melhor processo de fermentação aumentando a qualidade geral e longeividade do vinho
- 8 - density (g/dmˆ3): A densidade do vinho está relacionada principalmente ao seu teor alcoólico e de açúcares residuais
- 9 - pH: Os níveis estão ligados ao estilo e qualidade dos vinhos. Geralmente um vinho com níveis de pH mais baixos terá maior longevidade.
- 10 - sulphates (g/dm3): Utilizado como conservante para o vinho.

- 11 - alcohol (% volume): Volume de álccol presente na bebida

Variável target:

12 - quality (0 a 10):

Fonte dos dados

Paulo Cortez, University of Minho, Guimarães, Portugal, http://www3.dsi.uminho.pt/pcortez A. Cerdeira, F. Almeida, T. Matos and J. Reis, Viticulture Commission of the Vinho Verde Region(CVRVV), Porto, Portugal @2009

2 - Carga dos dados

```
# Definindo diretório de trabalho
setwd('C:/FCD/R/UCI/wine_quality')
getwd()
```

```
## [1] "C:/FCD/R/UCI/wine_quality"
```

```
# Carga de Pacotes
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
library(grid)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.1.3
```

```
## corrplot 0.92 loaded
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.3
```

```
## Carregando pacotes exigidos: lattice
```

```
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.1.3
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(nnet)
```

```
## Warning: package 'nnet' was built under R version 4.1.3
```

```
library(rmarkdown)
```

```
## Warning: package 'rmarkdown' was built under R version 4.1.3
```

```
# Carrega dados
wine_red <- read.csv('winequality-red.csv', sep = ";")
wine_white <- read.csv('winequality-white.csv', sep = ";")

# Criar coluna identificando o tipo de vinho
wine_red['color'] = 1 # Red Wine
wine_white['color'] = 0 # White Wine

# Único dataset para os dois vinhos
df_wine <- rbind(wine_red, wine_white)

head(df_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 color
## 1       5     1
## 2       5     1
## 3       5     1
## 4       6     1
## 5       5     1
## 6       5     1
```

3 - Análise Exploratória

```
## Resumo estatístico inicial
summary(df_wine)
```

```
##  fixed.acidity    volatile.acidity  citric.acid     residual.sugar
##  Min.   : 3.800   Min.   :0.0800   Min.   :0.0000   Min.   : 0.600
##  1st Qu.: 6.400   1st Qu.:0.2300   1st Qu.:0.2500   1st Qu.: 1.800
##  Median : 7.000   Median :0.2900   Median :0.3100   Median : 3.000
```

```
##  Mean   : 7.215   Mean   :0.3397   Mean   :0.3186   Mean   : 5.443
##  3rd Qu.: 7.700   3rd Qu.:0.4000   3rd Qu.:0.3900   3rd Qu.: 8.100
##  Max.   :15.900   Max.   :1.5800   Max.   :1.6600   Max.   :65.800
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide    density
##  Min.   :0.00900   Min.   :  1.00      Min.   :  6.0        Min.   :0.9871
##  1st Qu.:0.03800   1st Qu.: 17.00      1st Qu.: 77.0        1st Qu.:0.9923
##  Median :0.04700   Median : 29.00      Median :118.0        Median :0.9949
##  Mean   :0.05603   Mean   : 30.53      Mean   :115.7        Mean   :0.9947
##  3rd Qu.:0.06500   3rd Qu.: 41.00      3rd Qu.:156.0        3rd Qu.:0.9970
##  Max.   :0.61100   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.:3.110   1st Qu.:0.4300   1st Qu.: 9.50   1st Qu.:5.000
##  Median :3.210   Median :0.5100   Median :10.30   Median :6.000
##  Mean   :3.219   Mean   :0.5313   Mean   :10.49   Mean   :5.818
##  3rd Qu.:3.320   3rd Qu.:0.6000   3rd Qu.:11.30   3rd Qu.:6.000
##  Max.   :4.010   Max.   :2.0000   Max.   :14.90   Max.   :9.000
##      color
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.2461
##  3rd Qu.:0.0000
##  Max.   :1.0000
```

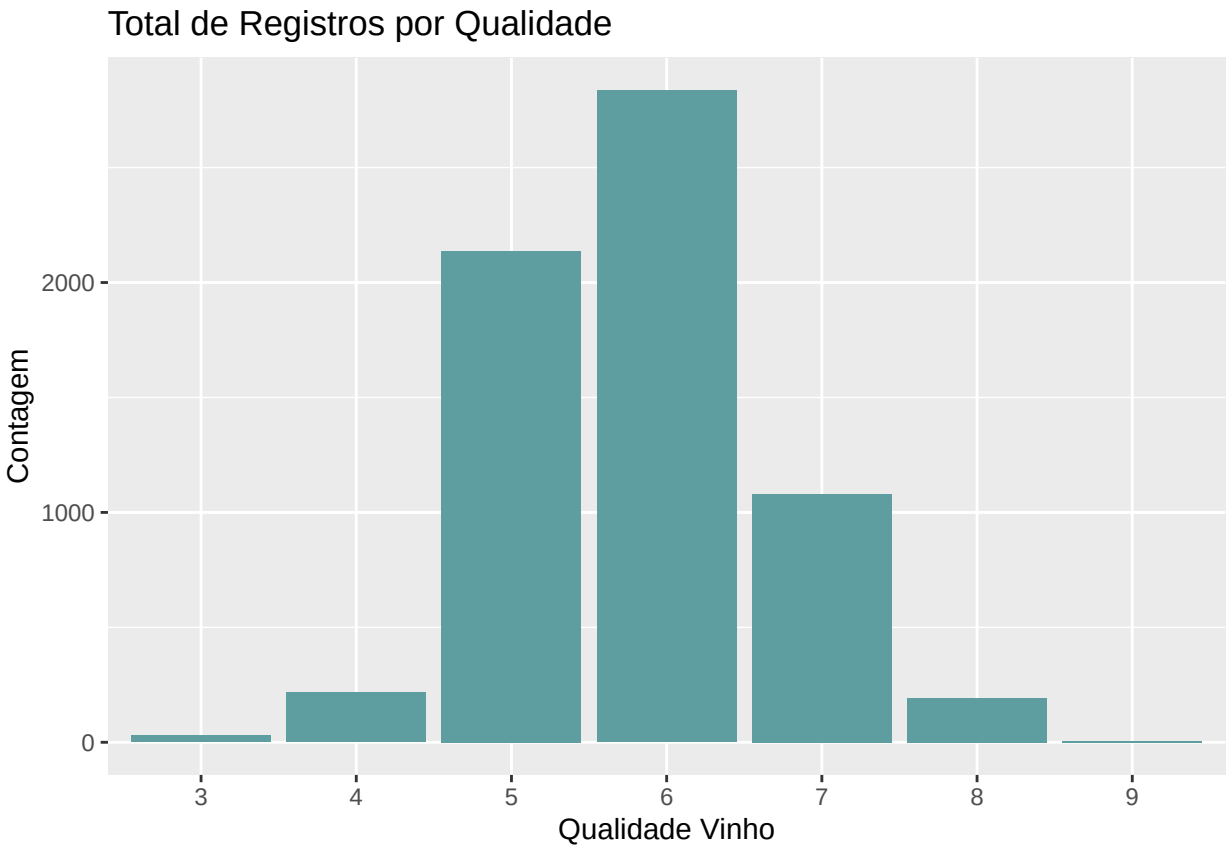
Na análise preliminar, vemos que a média e mediana estão próximas em poucos atributos (ex.: pH, density e alcohol). Nas outras variáveis, temos uma diferença considerável. Isso indica que os dados podem precisar de remoção de outliers.

Pelo dicionário de dados, temos atributos numéricos de diferentes escalas. Por esse motivo, será preciso padronizar os dados.

```
{ r tipodados} # Tipos de dados str(df_wine)
```

Atributo quality e color podem ser classificadas como categóricas. Detalhe para a variável quality porque a mesma é target. Isso implica no tipo de modelo. Para classificação, o interessante é classificar como categoria (factor). Já para regressão, pode-se usar o label enconding.

```
# Gráfico de barras - Wine Quality
ggplot(df_wine, aes(x = as.factor(quality))) +
  geom_bar(fill = 'cadetblue' ) +
  labs(x = 'Qualidade Vinho', y = 'Contagem', title = 'Total de Registros por Qualidade')
```



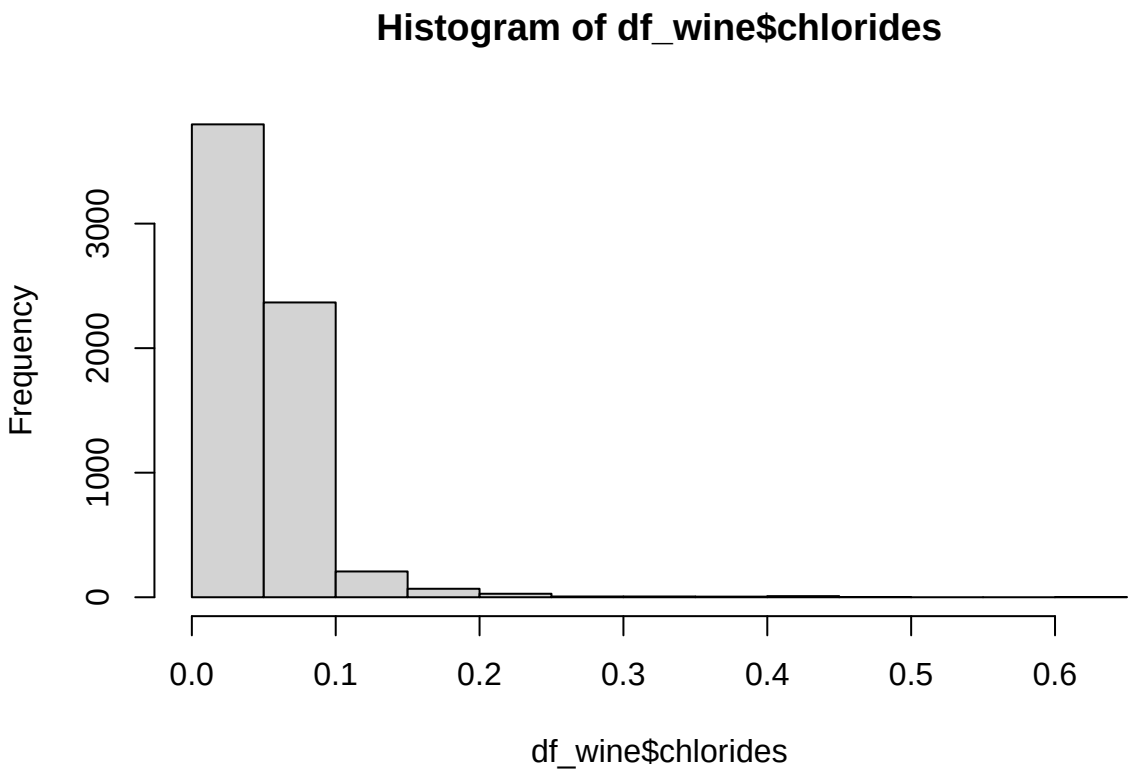
Fica claro que temos a maioria dos registros com qualidade média, o que é esperado. Uma classificação pode criar um modelo com tendência em 6 ou 5, isso pode ser um problema

Análise de correlação

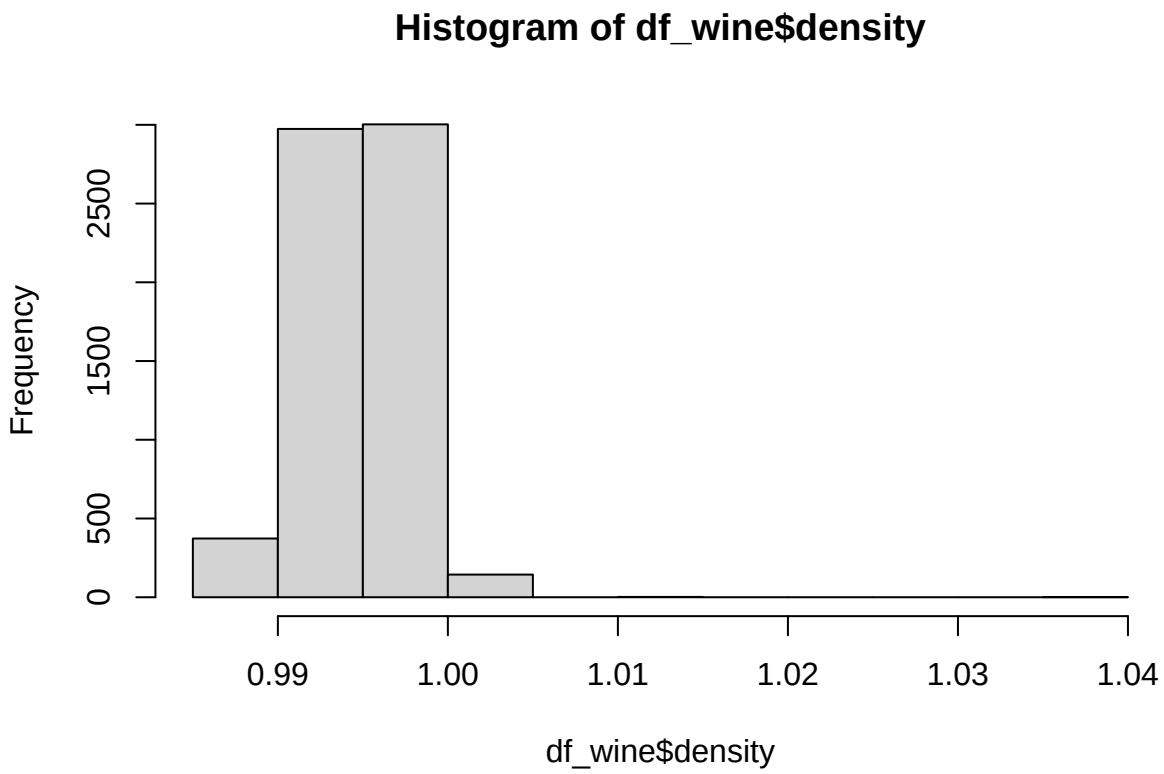
```
{ r correlacao} corrplot(cor(df_wine[,1:12]), type = 'upper')
```

Multicolinearidade entre as variáveis alcohol e density. Além disso, volatile.acidity parece ter uma correlação negativa com quality

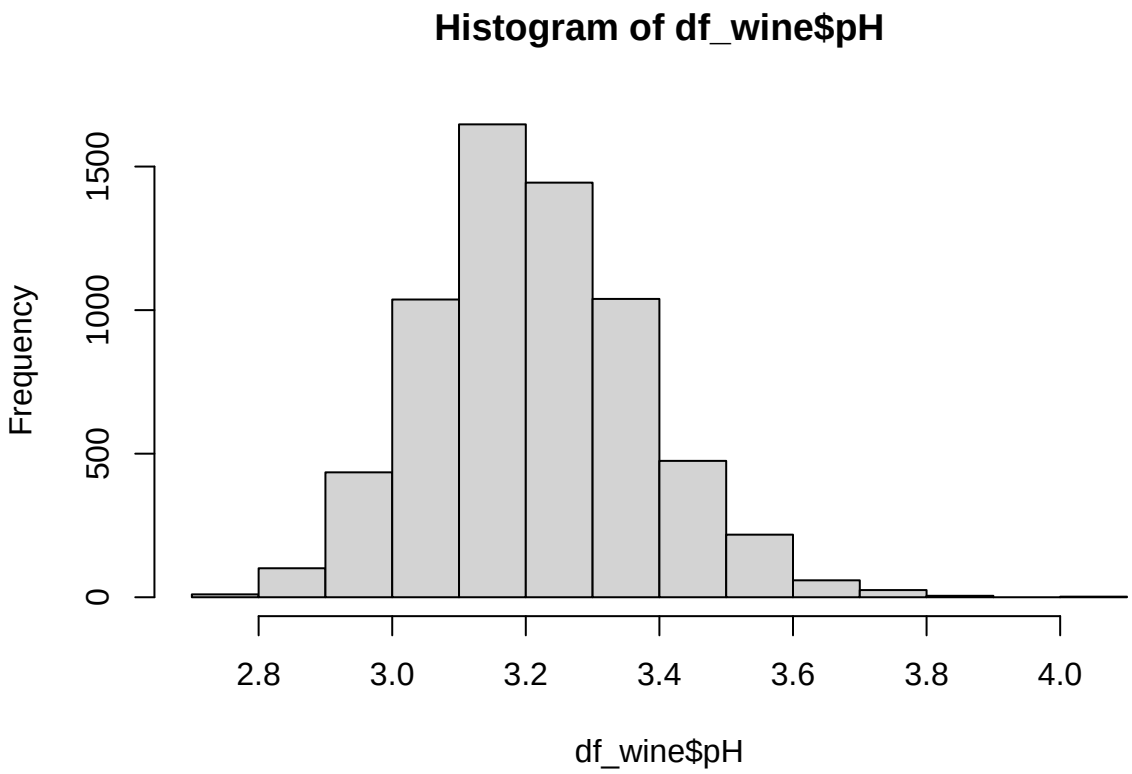
```
# Histogramas
hist(df_wine$chlorides)
```




```
hist(df_wine$density)
```




```
hist(df_wine$pH)
```



Dos histogramas feitos, fica claro que nem todas as variáveis tendem a uma normal. Isso reforça o que foi visto com relação a média e mediana de algumas variáveis. Será preciso uma padronização dos dados

4 - Pré Processamento

Normalização dos dados

```
df <- df_wine[,1:12] # Preserva o dataset original, descartando o tipo de vinho.

names_df <- colnames(df[,1:11])
names_df # Remove a coluna target para normalização
```

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

```
# Função para teste de normalidade
fun.normal <- function(df, variavel){
  for (variavel in variavel){
    a <- shapiro.test(df[1:5000, variavel])
    print(paste('O valor p é: ', a$p.value))

  }
return()
}
```

```
fun.normal(df, names_df)
```

```
## [1] "O valor p é:  9.59518537193713e-52"
## [1] "O valor p é:  1.54570576681314e-50"
## [1] "O valor p é:  8.32653829425776e-31"
## [1] "O valor p é:  3.44251265343529e-62"
## [1] "O valor p é:  4.69134499666567e-74"
## [1] "O valor p é:  9.31527145069941e-38"
## [1] "O valor p é:  4.02539888387008e-29"
## [1] "O valor p é:  3.82275289728067e-34"
## [1] "O valor p é:  1.50313855231543e-15"
## [1] "O valor p é:  9.41991895995202e-50"
## [1] "O valor p é:  1.63115031714687e-38"
```

```
## NULL
```

Como verificado anteriormente, os dados não passaram no teste de Shapiro. Isso indica que asa variáveis tendem a não seguir uma distribuição normal.

```
# Função para normalização dos dados
scale.features <- function(df, variavel){
  for (variavel in variavel) {
    df[[variavel]] <- scale(df[[variavel]], center = T, scale = T)
  }
  return(df)
}

df_normal <- scale.features(df, names_df)
```

Próximo passo é a divisão em treino e teste de forma aleatória afim de garantir a generalização dos modelo.

```
# Dados de treino e teste
amostra_dados <- sample(x = nrow(df_normal),
                        size = 0.8 * nrow(df_normal),
                        replace = FALSE)

# Dados de treino e teste
dados_treino <- df_normal[amostra_dados,]
dados_teste <- df_normal[-amostra_dados,]
```

5 - Modelo de Preditivo

```
modelo_v1 <- lm( quality ~ ., data = dados_treino)
summary(modelo_v1)
```

```
##
## Call:
## lm(formula = quality ~ ., data = dados_treino)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6965 -0.4646 -0.0474  0.4657  2.7388
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)           5.81840    0.01022 569.544  < 2e-16 ***
## fixed.acidity         0.07778    0.02256   3.448 0.000569 ***
## volatile.acidity     -0.22111    0.01437 -15.390  < 2e-16 ***
## citric.acid          -0.01319    0.01319  -1.000 0.317349
## residual.sugar        0.20368    0.02733   7.453 1.06e-13 ***
## chlorides            -0.01811    0.01291  -1.403 0.160777
## free.sulfur.dioxide   0.09898    0.01485   6.665 2.92e-11 ***
## total.sulfur.dioxide -0.13434    0.01750  -7.675 1.96e-14 ***
## density              -0.15792    0.04024  -3.924 8.81e-05 ***
## pH                    0.07358    0.01630   4.515 6.49e-06 ***
## sulphates             0.10911    0.01254   8.702  < 2e-16 ***
## alcohol               0.32833    0.02202  14.913  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7362 on 5185 degrees of freedom
## Multiple R-squared:  0.2998, Adjusted R-squared:  0.2983
## F-statistic: 201.8 on 11 and 5185 DF,  p-value: < 2.2e-16
```

```
# var target com fator
dados_treino2 <- dados_treino
dados_treino2$quality <- as.factor(dados_treino2$quality)

dados_teste2 <- dados_teste
dados_teste2$quality <- as.factor(dados_teste2$quality)

modelo_v2 <- multinom(quality ~ ., data = dados_treino2)
```

```
## # weights:  91 (72 variable)
## initial  value 10112.895045
## iter  10 value 7382.033708
## iter  20 value 6842.054275
## iter  30 value 5952.668107
## iter  40 value 5697.657179
## iter  50 value 5571.220702
## iter  60 value 5536.612546
## iter  70 value 5531.282840
## iter  80 value 5529.078975
## iter  90 value 5527.948389
## iter 100 value 5527.611509
## final  value 5527.611509
## stopped after 100 iterations
```

```
summary(modelo_v2)
```

```
## Call:
## multinom(formula = quality ~ ., data = dados_treino2)
##
## Coefficients:
##   (Intercept) fixed.acidity volatile.acidity citric.acid residual.sugar
## 4    2.636641    -1.3101028       -0.2482087  0.26488733    -0.58789382
## 5    5.264222    -1.5750361       -0.6051884  0.19424533    -0.95742187
## 6    5.813609    -1.5381096       -1.2773333  0.12930864    -0.65351903
## 7    4.345219    -1.0401860       -1.5426195  0.09881413    -0.05211026
## 8    2.416823    -1.1326448       -1.4511745  0.16097482     0.31146609
```

```
## 9  -13.890315     0.9197084       -1.9601198  1.06921604    -0.65846365
##     chlorides free.sulfur.dioxide total.sulfur.dioxide    density         pH
## 4  -0.5158688         -1.76059329           0.41088574  0.5344796 -0.7336597
## 5  -0.4915509         -0.77442744           0.03537153  1.4799896 -0.9750151
## 6  -0.4781570         -0.53615250          -0.32038535  1.4092736 -0.8617025
## 7  -0.8281936         -0.43509102          -0.50028048  0.6636595 -0.5284393
## 8  -0.5787825         -0.21120501          -0.54395204  0.3862126 -0.5461478
## 9 -10.0566394         -0.07078241          -0.37277218 -0.4015041  2.1496198
##   sulphates     alcohol
## 4 0.5514125 -0.09429938
## 5 0.6754091  0.12334822
## 6 0.9049451  1.07294632
## 7 1.2066736  1.48298973
## 8 1.0656734  1.67881692
## 9 0.4988507  3.46692896
##
## Std. Errors:
##   (Intercept) fixed.acidity volatile.acidity citric.acid residual.sugar
## 4   0.4013370     0.3935547        0.1951238   0.2871121      0.5161243
## 5   0.3857528     0.3519045        0.1841129   0.2726282      0.4744148
## 6   0.3851213     0.3505874        0.1887392   0.2734596      0.4744371
## 7   0.3881694     0.3522777        0.1991492   0.2780658      0.4820324
## 8   0.4083775     0.4025542        0.2380650   0.2960539      0.5329905
## 9   5.6546406     0.7631994        1.0359574   0.8346399      2.5456869
##   chlorides free.sulfur.dioxide total.sulfur.dioxide   density        pH
## 4 0.1768728           0.2594635            0.3778894 0.7313034 0.3417022
## 5 0.1504551           0.2159223            0.3605384 0.6618829 0.3173669
## 6 0.1517973           0.2140421            0.3616600 0.6617164 0.3170063
## 7 0.1864298           0.2185298            0.3671493 0.6718512 0.3194701
## 8 0.2582252           0.2315457            0.3945992 0.7664206 0.3428739
## 9 4.2819445           0.8601680            1.5980148 3.2745984 1.1969633
##   sulphates   alcohol
## 4 0.3702200 0.4815416
## 5 0.3522571 0.4464838
## 6 0.3525717 0.4468239
## 7 0.3546354 0.4520751
## 8 0.3661161 0.4898497
## 9 0.9786778 1.8456153
##
## Residual Deviance: 11055.22
## AIC: 11199.22
```

```
previsoes <- predict(modelo_v2, dados_teste)
confusionMatrix(previsoes, dados_teste2$quality)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   3   4   5   6   7   8   9
##          3   0   0   0   0   0   0   0
##          4   0   0   0   0   0   0   0
##          5   2  25 254 146  18   1   0
##          6   3  18 169 415 152  19   1
##          7   1   1   2  24  38   9   0
##          8   0   0   0   0   0   0   0
```

```
##          9   0   0   0   1   0   1   0
##
## Overall Statistics
##
##                Accuracy : 0.5438
##                  95% CI : (0.5163, 0.5712)
##     No Information Rate : 0.4508
##     P-Value [Acc > NIR] : 1.074e-11
##
##                   Kappa : 0.2512
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 3 Class: 4 Class: 5 Class: 6 Class: 7 Class: 8
## Sensitivity          0.000000  0.00000   0.5976   0.7082  0.18269  0.00000
## Specificity          1.000000  1.00000   0.7806   0.4930  0.96612  1.00000
## Pos Pred Value            NaN      NaN   0.5695   0.5341  0.50667      NaN
## Neg Pred Value       0.995385  0.96615   0.7998   0.6730  0.86122  0.97692
## Prevalence           0.004615  0.03385   0.3269   0.4508  0.16000  0.02308
## Detection Rate       0.000000  0.00000   0.1954   0.3192  0.02923  0.00000
## Detection Prevalence 0.000000  0.00000   0.3431   0.5977  0.05769  0.00000
## Balanced Accuracy    0.500000  0.50000   0.6891   0.6006  0.57440  0.50000
##                       Class: 9
## Sensitivity          0.0000000
## Specificity          0.9984604
## Pos Pred Value       0.0000000
## Neg Pred Value       0.9992296
## Prevalence           0.0007692
## Detection Rate       0.0000000
## Detection Prevalence 0.0015385
## Balanced Accuracy    0.4992302
```

Ao analisar os dois modelos, fica claro que o de regressão linear multivariada não teve um bom desempenho (com um R2 de abaixo de 30%). Já o modelo de regressão logística multivariada teve um desempenho melhor (acurácia acima de 50%). Será necessário mais testes com diferentes modelos, mas a princípio uma regressão linear não funcionou muito bem porque a variável target é uma classificação, ou seja, a nota que o especilista determinou para o tipo de vinho. Modelos de classificação tendem a ter um melhor desempenho nesse sentido.

Próximos passos

- Remoção de alguns outliers antes de rodar os modelos.
- Análise de Multicolinearidade
- Modelo com RandomForest
- Modelo com DecisonTree