# Prediction of Wine Quality

## 1.

For task 4, we try to predict of red wine qualities. The data set that we used.

## 2.

For that step, we'll describe the data set in our task in terms of the dimension, variable type and etc. Firstly;

```
winequality.red <- read.csv("winequality-red.csv")
```

After that;

```
str(winequality.red)
```

```
'data.frame':   1599 obs. of  12 variables:
 $ fixed.acidity       : num  7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
 $ volatile.acidity    : num  0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...
 $ citric.acid         : num  0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
 $ residual.sugar      : num  1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...
 $ chlorides           : num  0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.071 ..
 $ free.sulfur.dioxide : num  11 25 15 17 11 13 15 15 9 17 ...
 $ total.sulfur.dioxide: num  34 67 54 60 34 40 59 21 18 102 ...
 $ density             : num  0.998 0.997 0.997 0.998 0.998 ...
 $ pH                  : num  3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...
 $ sulphates           : num  0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...
 $ alcohol             : num  9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
 $ quality             : int  5 5 5 6 5 5 5 7 7 5 ...
```

The data set has 1599 obs. and 12 variables. `quality` is our ***target***. That means our data set have 11 column. In addition to this, all of our ***features*** seems like they are all `numeric`.

### 3.

We are going to use `sample()` function to ***split the data set*** as `test` and `train` set.

```
set.seed(123)
index <- sample(1 : nrow(winequality.red), round(nrow(winequality.red) * 0.80))
train <- winequality.red[index, ]
test  <- winequality.red[-index, ]
```

Then; we can use the `glm()` function to train a logistic regression model.

```
lr_model <- glm(quality ~ ., data = train)
summary(lr_model)
```

```
Call:
glm(formula = quality ~ ., data = train)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.65018  -0.38387  -0.04515   0.45703   2.01377

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)           1.492e+01  2.405e+01   0.620   0.5351
fixed.acidity         1.245e-02  2.998e-02   0.415   0.6780
volatile.acidity     -1.022e+00  1.398e-01  -7.306 4.87e-13 ***
citric.acid          -1.298e-01  1.687e-01  -0.769   0.4419
residual.sugar        6.223e-03  1.749e-02   0.356   0.7220
chlorides            -2.059e+00  4.711e-01  -4.370 1.34e-05 ***
free.sulfur.dioxide   4.125e-03  2.501e-03   1.649   0.0994 .
total.sulfur.dioxide -3.556e-03  8.348e-04  -4.260 2.20e-05 ***
density              -1.026e+01  2.455e+01  -0.418   0.6760
pH                   -5.780e-01  2.193e-01  -2.636   0.0085 **
sulphates             8.652e-01  1.329e-01   6.509 1.09e-10 ***
alcohol               2.907e-01  3.040e-02   9.564  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 0.4401224)
```

```
    Null deviance: 864.63  on 1278  degrees of freedom
Residual deviance: 557.64  on 1267  degrees of freedom
AIC: 2593.9

Number of Fisher Scoring iterations: 2
```

After all of this, the last part of this step is prediction.

```r
predicted_quality <- predict(lr_model, test)
head(predicted_quality)
```

```
       3        7       15       22       23       27
5.226765 5.116426 5.096636 5.368662 5.743560 5.542505
```

## 4.

In that part, We will use the ***Root Mean Square Error (RMSE)*** metric to measure the performance of the regression model.

```r
error <- test$quality - predicted_quality
rmse.model <- sqrt(mean(error^2))
rmse.model
```

```
[1] 0.5859451
```

The performance of the model seems really acceptable because that value is low.

## 5.

Last but not least, we'll check if there is an ***overfitting problem***.

```r
rmse.test <- sqrt(mean((lr_model$residuals)^2))
rmse.model - rmse.test
```

```
[1] -0.07435255
```

Because of the difference which is negative, it can be means that we have the overfitting problem of the model.

### 6.

We will create ***new observations*** for last step of task 4.

```
fixed.acidity <- as.numeric(c(7.1, 8.1, 7.5))
volatile.acidity <- as.numeric(c(0.5, 0.67, 0.89))
citric.acid <- as.numeric(c(0.4, 0.6, 0.58))
residual.sugar <- as.numeric(c(2.1, 2.9, 1.99))
chlorides <- as.numeric(c(0.01, 0.03, 0.055))
free.sulfur.dioxide <- as.numeric(c(18, 17, 26))
total.sulfur.dioxide <- as.numeric(c(37, 41, 51))
density  <- as.numeric(c(0.950, 0.917, 0.940))
pH <- as.numeric(c(3.72, 3.81, 3.1))
sulphates <- as.numeric(c(0.91, 0.89, 0.85))
alcohol <- as.numeric(c(9.91, 9.87, 9.83))

new.observations <- data.frame(fixed.acidity, volatile.acidity, citric.acid , residual.sug
```

```
new.observations
```

```
  fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
1           7.1             0.50        0.40           2.10     0.010
2           8.1             0.67        0.60           2.90     0.030
3           7.5             0.89        0.58           1.99     0.055
  free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
1                  18                   37   0.950 3.72      0.91    9.91
2                  17                   41   0.917 3.81      0.89    9.87
3                  26                   51   0.940 3.10      0.85    9.83
```

Now, we can prediction with new ones.

```
predicted_quality_new <- predict(lr_model, new.observations)
predicted_quality_new
```

```
       1        2        3 
6.150804 6.166856 6.009734 
```