WineQuality (Logistic Regression using R)

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

The objective of this assignment is to learn how to explore, prepare, and analyze a Logistic Regression Model. This dataset is about the properties of Wine and its quality. The objective is to predict the Quality of the wine using its contents such as alcohol, density, and pH level. We will also use statistical analysis to see what attributes are significant/insignificant to the model(s).

Importing Data:

```
wine <- read.csv("/Users/azamrahman/Desktop/CMTH 642 (R)/Assignment 2/winequality-white.csv", header = T
```

Check the datatypes of the attributes.

```
sapply(wine, class)
```

```
##        fixed.acidity     volatile.acidity          citric.acid
##            "numeric"            "numeric"            "numeric"
##       residual.sugar            chlorides  free.sulfur.dioxide
##            "numeric"            "numeric"            "numeric"
## total.sulfur.dioxide              density                   pH
##            "numeric"            "numeric"            "numeric"
##            sulphates              alcohol              quality
##            "numeric"            "numeric"            "integer"
```

```
# All variables are numeric except for quality, which is an integer.
```

Are there any missing values in the dataset?

```
sum(is.na(wine))
```

```
## [1] 0
```

```
# There are no missing values in this dataset.
```

What is the correlation between the attributes other than Quality?

```
library(corrplot)
```

```
## corrplot 0.89 loaded
```

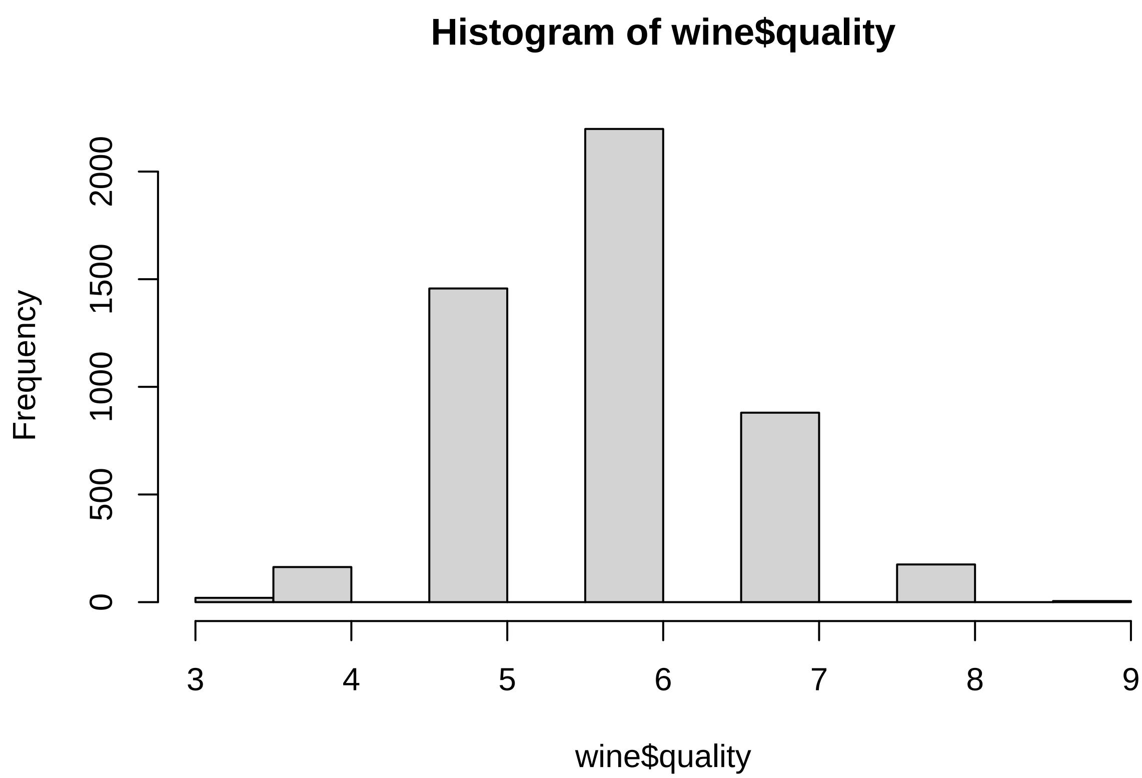
```
winecor <- data.frame(wine$fixed.acidity, wine$volatile.acidity, wine$citric.acid, wine$residual.sugar,

corrplot(cor(winecor), type = "upper", order = "hclust", tl.col = "black",
         tl.srt = 45.)
```


```
# Couple of things to note from this: density vs. alcohol has a strong negative correlation
# Sugar and Density has a strong positive correlation
```

Graph the frequency distribution of wine quality by using Quality.

```
hist(wine$quality)
```




```
# We can see the most frequent wine quality is rated at 6, followed by 5.
```

Reduce the levels of rating for quality to two levels as Pass and Fail. Assign the levels of 3, 4 and 5 to level Fail; and 6, 7, 8 and 9 to level Pass.

```
wine$quality[wine$quality == 3 | wine$quality == 4 | wine$quality == 5 ] = "Fail"
wine$quality[wine$quality == 6 | wine$quality == 7 | wine$quality == 8 | wine$quality == 9 ] = "Pass"
wine$quality <- as.factor(wine$quality)
```

Normalize the data set.

```
norm <- function(x){
  return ((x - min(x)) / (max(x) - min(x)))
}

# Pass = 1, Fail = 0:
wine$quality <- ifelse(wine$quality == "Pass",1,0)

winenorm <- data.frame(sapply(wine, norm))
```

Divide the dataset to training and test sets.

```
# Splitting data (70% Train, 30% Test)
train_index <- sample(1:nrow(winenorm), 0.7 * nrow(winenorm))
train.set <- winenorm[train_index,]
test.set <- winenorm[-train_index,]
```

Use the Logistic Regression algorithm to predict the quality of wine using its attributes.

```r
# Create Logistic Regression:
model <- glm(formula = quality ~., family = "binomial", data = winenorm)

summary(model)
```

```
##
## Call:
## glm(formula = quality ~ ., family = "binomial", data = winenorm)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1731  -0.8946   0.4420   0.7994   2.9466
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -0.1031     0.3732  -0.276 0.782377
## fixed.acidity           0.3794     0.7465   0.508 0.611271
## volatile.acidity       -6.5881     0.4211 -15.646  < 2e-16 ***
## citric.acid             0.1923     0.5029   0.382 0.702219
## residual.sugar         11.0883     1.7627   6.291 3.16e-10 ***
## chlorides               0.2983     0.5632   0.530 0.596379
## free.sulfur.dioxide     2.7555     0.7986   3.451 0.000560 ***
## total.sulfur.dioxide   -0.5746     0.5220  -1.101 0.270982
## density               -14.0503     3.7321  -3.765 0.000167 ***
## pH                      1.1990     0.3980   3.013 0.002590 **
## sulphates               1.5458     0.3092   5.000 5.75e-07 ***
## alcohol                 4.6062     0.5804   7.937 2.08e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6245.4  on 4897  degrees of freedom
## Residual deviance: 4932.6  on 4886  degrees of freedom
## AIC: 4956.6
##
## Number of Fisher Scoring iterations: 5
```

```r
# Predict model:
pdata <- predict(model, test.set, type = "response")

predicted_class <- ifelse(pdata > 0.5, 1, 0)
```

With a significance level of 5%, there are a couple of variables that are not significant to the model as it equals zero. fixed.acidity, citric.acid, chlorides, and total.sulfur.dioxide, all contain p values above 0.05 and therefore we do not reject the null hypothesis for these variables to show that the variables are insignificant to the model.

Display the confusion matrix to evaluate the model performance.

```r
ConfusionMatrix <- table(actual = test.set$quality, predicted = predicted_class)

ConfusionMatrix
```

```
##       predicted
## actual   0   1
##      0 246 270
##      1 107 847
```

Evaluate the model performance by computing Accuracy, Sensitivity and Specificity.

```r
library(caret)
```

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

```
## Loading required package: ggplot2
```

```r
# Accuracy: (relatively a decent accuracy from this model)
Accuracy <-  sum(diag(ConfusionMatrix))/nrow(test.set)
Accuracy
```

```
## [1] 0.7435374
```

```r
# Sensitivity: (relatively an okay sensitivity as it effectively predicts wine quality of Pass)
Sensitivity <- sensitivity(ConfusionMatrix)
Sensitivity
```

```
## [1] 0.6968839
```

```r
# Specificity: (relatively a decent specificity as it effectively predicts wine quality of Fail)
Specificity <- specificity(ConfusionMatrix)
Specificity
```

```
## [1] 0.7582811
```

Extra Learning:

Let's create a model with only the significant variables to see how it would hold up against the complete model:

```r
# Create Selected Logistic Regression:
modelselected <- glm(formula = quality ~ volatile.acidity + residual.sugar + free.sulfur.dioxide + dens

summary(modelselected)
```

```
##
## Call:
## glm(formula = quality ~ volatile.acidity + residual.sugar + free.sulfur.dioxide +
##     density + pH + sulphates + alcohol, family = "binomial",
```

```
##     data = winenorm)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1417  -0.8926   0.4446   0.8018   2.8621
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -0.07664    0.35887  -0.214 0.830898
## volatile.acidity     -6.71394    0.40462 -16.593  < 2e-16 ***
## residual.sugar       10.50308    1.17015   8.976  < 2e-16 ***
## free.sulfur.dioxide   2.25436    0.63849   3.531 0.000414 ***
## density             -13.01885    2.30452  -5.649 1.61e-08 ***
## pH                    1.02214    0.27816   3.675 0.000238 ***
## sulphates             1.48027    0.30130   4.913 8.98e-07 ***
## alcohol               4.79266    0.41576  11.527  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6245.4  on 4897  degrees of freedom
## Residual deviance: 4934.6  on 4890  degrees of freedom
## AIC: 4950.6
##
## Number of Fisher Scoring iterations: 5
```

```r
# Predict model:
pdata <- predict(modelselected, test.set, type = "response")

predicted_class_sel <- ifelse(pdata > 0.5, 1, 0)
```

```r
ConfusionMatrixSel <- table(actual = test.set$quality, predicted = predicted_class_sel)

ConfusionMatrixSel
```

```
##       predicted
## actual   0   1
##      0 242 274
##      1 108 846
```

```r
# Accuracy: (relatively a decent accuracy from this model)
Accuracy <-  sum(diag(ConfusionMatrixSel))/nrow(test.set)
Accuracy
```

```
## [1] 0.7401361
```

```r
# Sensitivity: (relatively an okay sensitivity as it effectively predicts wine quality of Pass)
Sensitivity <- sensitivity(ConfusionMatrixSel)
Sensitivity
```

```
## [1] 0.6914286
```

```r
# Specificity: (relatively a decent specificity as it effectively predicts wine quality of Fail)
Specificity <- specificity(ConfusionMatrixSel)
Specificity
```

```
## [1] 0.7553571
```

With the Selected model, we see a very slight increase in performance measure which may indicate that it may not be necessary to select attributes since it makes an insignificant amount of difference.