FinalProject

Cole Maxwell

4/18/2022

Wine Data Machine Learning Models

These data were obtained from the University of California Irvine's Machine Learning Repository. These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. These models attempt to predict they `Type` of wine (White or Red) and determine the quality of both white and red wines.

Data Pre-Processing

```
library(knitr)
library(keras)
library(tensorflow)
library(ggplot2)
library(ROSE)
```

```
## Loaded ROSE 0.0-4
```

```
#Loading a csv file30
WhiteWineQualityDataFrame <- read.csv("winequality-white.csv", sep = ";")
RedWineQualityDataFrame <- read.csv("winequality-red.csv", sep = ";")
WineQualityDataFrame <- rbind(WhiteWineQualityDataFrame, RedWineQualityDataFrame)


# Various syntax for R dataframes
summary(WineQualityDataFrame) %>% table()
```

```
## .
##  1st Qu.: 1.800    1st Qu.: 17.00    1st Qu.: 6.400     1st Qu.: 77.0
##                 1                 1                 1                 1
##   1st Qu.: 9.50   1st Qu.:0.03800    1st Qu.:0.2300    1st Qu.:0.2500
##                 1                 1                 1                 1
##  1st Qu.:0.4300    1st Qu.:0.9923     1st Qu.:3.110     1st Qu.:5.000
##                 1                 1                 1                 1
##  3rd Qu.: 41.00    3rd Qu.: 7.700    3rd Qu.: 8.100   3rd Qu.:0.06500
##                 1                 1                 1                 1
##  3rd Qu.:0.3900    3rd Qu.:0.4000    3rd Qu.:0.6000    3rd Qu.:0.9970
##                 1                 1                 1                 1
##   3rd Qu.:11.30     3rd Qu.:156.0     3rd Qu.:3.320     3rd Qu.:6.000
##                 1                 1                 1                 1
## Max.   :0.61100    Max.   :1.0390    Max.   :1.5800    Max.   :1.6600
##                 1                 1                 1                 1
```

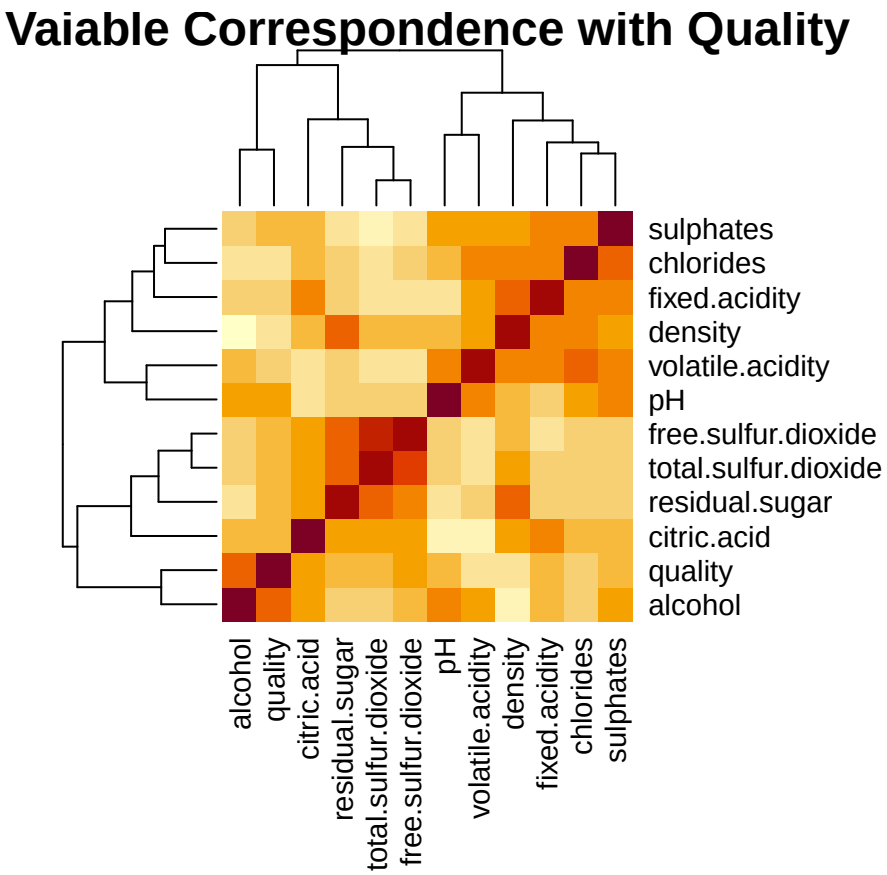
```
##   Max.   :14.90    Max.   :15.900    Max.   :2.0000    Max.   :289.00
##                  1                 1                 1                 1
##   Max.   :4.010     Max.   :440.0    Max.   :65.800     Max.   :9.000
##                  1                 1                 1                 1
##  Mean   : 30.53    Mean   : 5.443    Mean   : 7.215   Mean   :0.05603
##                  1                 1                 1                 1
##  Mean   :0.3186    Mean   :0.3397    Mean   :0.5313    Mean   :0.9947
##                  1                 1                 1                 1
##   Mean   :10.49     Mean   :115.7     Mean   :3.219     Mean   :5.818
##                  1                 1                 1                 1
##  Median : 29.00    Median : 3.000    Median : 7.000   Median :0.04700
##                  1                 1                 1                 1
##  Median :0.2900    Median :0.3100    Median :0.5100    Median :0.9949
##                  1                 1                 1                 1
##   Median :10.30     Median :118.0     Median :3.210     Median :6.000
##                  1                 1                 1                 1
##  Min.   :  1.00     Min.   :  6.0    Min.   : 0.600    Min.   : 3.800
##                  1                 1                 1                 1
##   Min.   : 8.00    Min.   :0.0000   Min.   :0.00900    Min.   :0.0800
##                  1                 1                 1                 1
##  Min.   :0.2200    Min.   :0.9871     Min.   :2.720     Min.   :3.000
##                  1                 1                 1                 1
```

```
head(WineQualityDataFrame)
```

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1           7.0             0.27        0.36           20.7     0.045
## 2           6.3             0.30        0.34            1.6     0.049
## 3           8.1             0.28        0.40            6.9     0.050
## 4           7.2             0.23        0.32            8.5     0.058
## 5           7.2             0.23        0.32            8.5     0.058
## 6           8.1             0.28        0.40            6.9     0.050
##   free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 1                  45                  170  1.0010 3.00      0.45     8.8
## 2                  14                  132  0.9940 3.30      0.49     9.5
## 3                  30                   97  0.9951 3.26      0.44    10.1
## 4                  47                  186  0.9956 3.19      0.40     9.9
## 5                  47                  186  0.9956 3.19      0.40     9.9
## 6                  30                   97  0.9951 3.26      0.44    10.1
##   quality
## 1       6
## 2       6
## 3       6
## 4       6
## 5       6
## 6       6
```

Exploritory Analysis

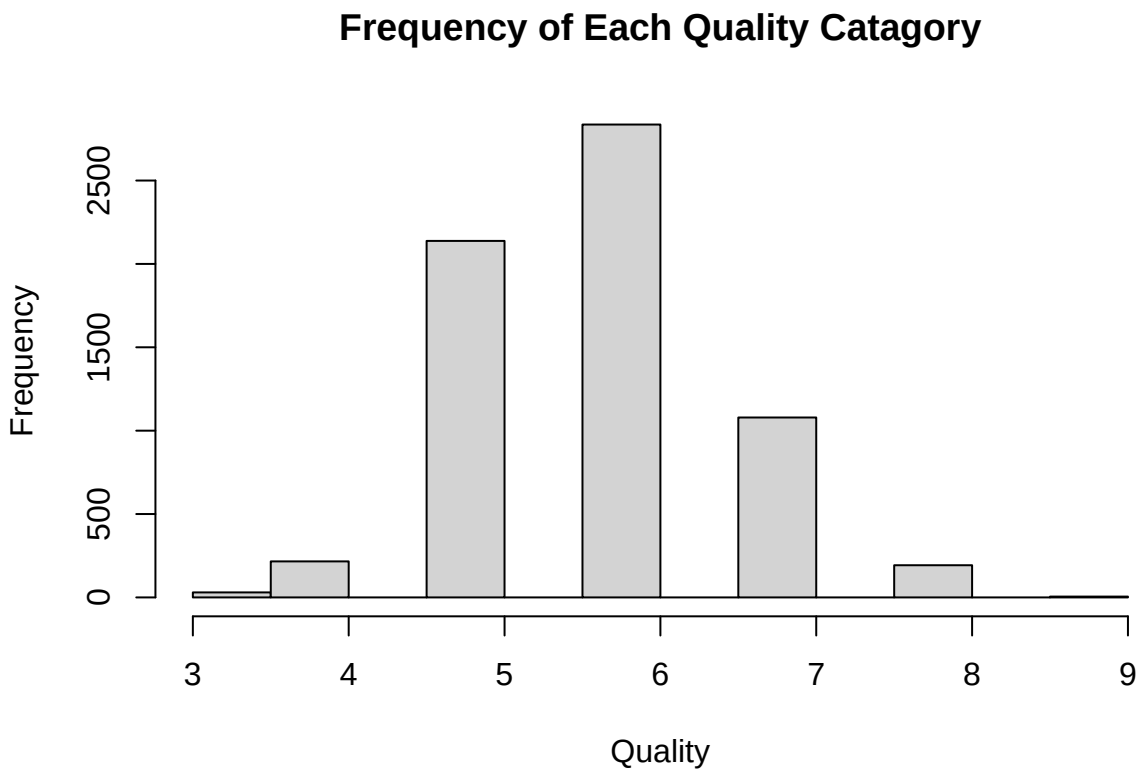
```
# Evaluation of the correlation variables
WineQualityMatrix <- data.matrix(WineQualityDataFrame)
WineQualityCorrelation <- cor(WineQualityMatrix)

# Generate a heatmap of correlations
heatmap(WineQualityCorrelation,
        margins = c(8, 8),
        main = "Vaiable Correspondence with Quality",
        )
```



```
table(WineQualityDataFrame$quality) %>%
  kable(col.names = c("Quality Level", "Frequency"),
        )
```

Quality Level	Frequency
3	30
4	216
5	2138
6	2836
7	1079
8	193
9	5

```
hist(WineQualityDataFrame$quality,
     xlab = "Quality",
     main = "Frequency of Each Quality Catagory"
     )
```



Trim density because no correlation

```
# WineQualityDataFrame <- subset(WineQualityDataFrame, select = -c(density))
```

Splitting data into training and testing

```
# Get 80% of the rows from the data set
sample_size <- round(nrow(WineQualityDataFrame) * 0.8)

# setting random seed to make results repeatable
set.seed(1234)


# Subtract three from the quality column so quality levels start at zero
WineQualityDataFrame$quality <- WineQualityDataFrame$quality -3

# Separate data into training and testing sets
picked <- sample(seq_len(nrow(WineQualityDataFrame)),size = sample_size)
training <- WineQualityDataFrame[picked,]
testing <- WineQualityDataFrame[-picked,]


# Changing y into categorical data (performing one-hot encoding)
yTr <- to_categorical(training$quality, num_classes = 7)
```

```
## Loaded Tensorflow version 2.0.0
```

```
yTest <- to_categorical(testing$quality, num_classes = 7)
```

Neural network for Wine Quality

This is where we tried to improve our accuracy. One way we found that was somewhat successful was to increase the number of epochs. We think Elena chose 20 epochs because she had a smaller data set but since ours was larger we needed more. We found that after 300 we got vastly diminishing returns, so we settled on 300 for the time being. We are still looking at ways of improving accuracy.

initial accuracy -> ~43% drop Id feature -> no improvement in accuracy increase epochs to 100 -> accuracy raised to ~53% increase epochs to 1000 -> accuracy raised to ~67% decrease epochs to 300 -> accuracy lowered to ~60% increase units from 64 to 128 in first layer -> ~56% increase input_shape from 4 to 11 -> ~80% increase first layer units from 64 to 128 -> ~81% increase second layer units from 64 to 128 -> ~88%

increased the sample_size from 120 to 500 -> increased 43% to 57% accuracy Removed two layers and had only one dropout layer at 25% -> accuracy: 0.5941 Removed two layers, removed dropout rate, now only look at 7 features, added regularization to second hidden layer -> 0.6277056 added third hidden layer, reduced first layer from 128 to 7 nodes, added regularization to three hidden layers -> 0.5844156 removed regularization from middle hidden layer -> 0.5454546 removed regularization from first hidden layer -> 0.5887446 add regularization to middle layer -> 0.5627705 7 hidden layer, layer 1 128 nodes, layers 2-7 64 nodes, regularization on layers 4 and 7 -> 0.5021645 add bias of 2.0 to first layer -> 0.5454546 3 hidden layers, regularization on last 2, bias of 2.0 on first layer -> 0.4935065 same as above but reg on hidden layers 1 and 3 -> 0.5324675 increase nodes in first layer back up from 7 to 128 -> 0.6190476 move bias from input to output node -> 0.5584416 add alcohol col as a parameter -> 100% (not the good 100 tho lol) reduce nodes in input layer to 8 from 128 -> 100% L1 = https://www.analyticssteps.com/blogs/l2-and-l1-regularization-machine-learning

Wow!!! Changed loss to binary_crossentropy -> accuracy: 0.8773 No increase in accuracy by changing epochs to 300

```
plot(history)
summary(wineModel)
```

Evaluate the model

```
wineModel %>% evaluate(xTest, yTest)

# Predicting likelihood of all categories:
result <- wineModel %>% predict(xTest)

result

testing[,11]
```