Classification

```r
library(ISLR)
library(ggplot2)
library(reshape2)
library(plyr)
library(dplyr)
```

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

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

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

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

```r
library(class)
```

```r
## Reading df
## setwd("/Users/joaopedro/Documents/MSBA/Classes/BAX 452 - Machine Learning/Assignments/05. Classifica
```

```r
## Reading the data
wine <- read.csv('winequality-red.csv')
head(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
## 1       5
## 2       5
## 3       5
## 4       6
## 5       5
## 6       5
```

```r
## Exploring the data
str(wine)
```

```
## '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 ...
```

Split the wine into three categories (for low, medium, and high quality)

To create three categories, we need to explore the distribution of the 'quality'.

```r
table(wine$quality)
```

```
##
##   3   4   5   6   7   8
##  10  53 681 638 199  18
```

Let's define:

- Low quality: 3, 4, 5
- Medium quality: 6
- High Quality: 7, 8

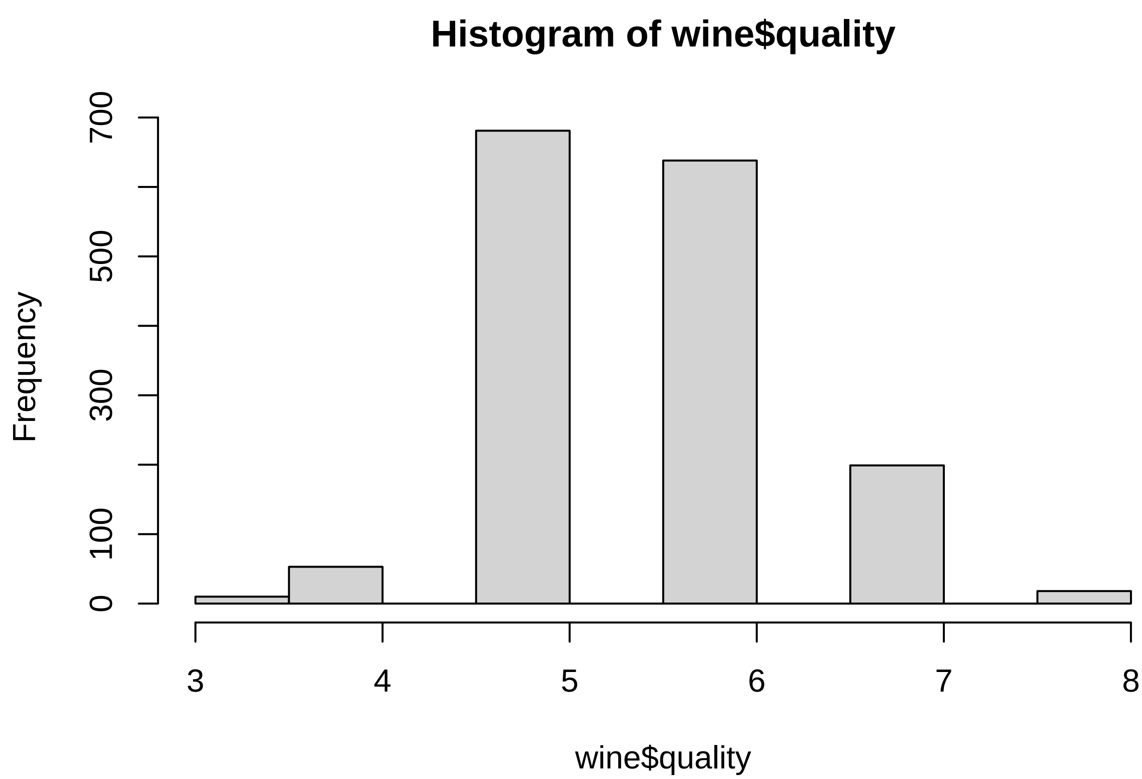
```r
assign_quality <- function(quality) {
                  if (quality < 6) {'low'}
                  else if (quality < 7) {'medium'}
                  else {'high'}
                  }
```

```r
wine['quality_group'] <- apply(X = wine['quality'], FUN = assign_quality, MARGIN = 1)
table(wine$quality_group)
```

```
##
##   high    low medium
##    217    744    638
```

Explore the data

```r
## Distribution of quality
hist(wine$quality)
```



The quality distribution is approximatelly normally distributed, ranging between 3 and 8, with the majority of wines in the 5 and 6 bins.

```r
## Check all possible correlations
cor_matrix <- round(cor(wine[,1:12]),3)
cor_matrix
```

```
##                      fixed.acidity volatile.acidity citric.acid residual.sugar
## fixed.acidity                1.000           -0.256       0.672          0.115
## volatile.acidity            -0.256            1.000      -0.552          0.002
## citric.acid                  0.672           -0.552       1.000          0.144
## residual.sugar               0.115            0.002       0.144          1.000
## chlorides                    0.094            0.061       0.204          0.056
```

```
## free.sulfur.dioxide         -0.154           -0.011      -0.061          0.187
## total.sulfur.dioxide        -0.113            0.076       0.036          0.203
## density                      0.668            0.022       0.365          0.355
## pH                          -0.683            0.235      -0.542         -0.086
## sulphates                    0.183           -0.261       0.313          0.006
## alcohol                     -0.062           -0.202       0.110          0.042
## quality                      0.124           -0.391       0.226          0.014
##                        chlorides free.sulfur.dioxide total.sulfur.dioxide density
## fixed.acidity              0.094              -0.154               -0.113   0.668
## volatile.acidity           0.061              -0.011                0.076   0.022
## citric.acid                0.204              -0.061                0.036   0.365
## residual.sugar             0.056               0.187                0.203   0.355
## chlorides                  1.000               0.006                0.047   0.201
## free.sulfur.dioxide        0.006               1.000                0.668  -0.022
## total.sulfur.dioxide       0.047               0.668                1.000   0.071
## density                    0.201              -0.022                0.071   1.000
## pH                        -0.265               0.070               -0.066  -0.342
## sulphates                  0.371               0.052                0.043   0.149
## alcohol                   -0.221              -0.069               -0.206  -0.496
## quality                   -0.129              -0.051               -0.185  -0.175
##                            pH sulphates alcohol quality
## fixed.acidity          -0.683     0.183  -0.062   0.124
## volatile.acidity        0.235    -0.261  -0.202  -0.391
## citric.acid            -0.542     0.313   0.110   0.226
## residual.sugar         -0.086     0.006   0.042   0.014
## chlorides              -0.265     0.371  -0.221  -0.129
## free.sulfur.dioxide     0.070     0.052  -0.069  -0.051
## total.sulfur.dioxide -0.066     0.043  -0.206  -0.185
## density                -0.342     0.149  -0.496  -0.175
## pH                      1.000    -0.197   0.206  -0.058
## sulphates              -0.197     1.000   0.094   0.251
## alcohol                 0.206     0.094   1.000   0.476
## quality                -0.058     0.251   0.476   1.000
```

The quality variable shows a positive correlations (>0.2) with:

- alcohol: 0.48
- sulphates: 0.25
- citric acid: 0.23

And negative correlation (< -0.2) with:

- Volatile acid: -0.39

Split the data into 80% training and 20% testing.

```
library(caret)
```

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

```r
## Droping quality column
wine <- select(wine, -c(quality))

## Train Test Split
set.seed(123)
train_test <- createDataPartition(y = wine$quality_group, p = 0.8, list = FALSE)
training <- wine[train_test,]
testing <- wine[-train_test,]

## Checking Split
dim(training); dim(testing)
```

```
## [1] 1281   12
```

```
## [1] 318  12
```

```r
## Training the model
trControl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)

set.seed(123)

knn_fit <- train(quality_group ~., data = training, method = "knn",
 trControl=trControl,
 preProcess = c("center", "scale"),
 tuneLength = 10)

## Model Result
knn_fit
```

```
## k-Nearest Neighbors
##
## 1281 samples
##   11 predictor
##    3 classes: 'high', 'low', 'medium'
##
## Pre-processing: centered (11), scaled (11)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 1153, 1152, 1154, 1152, 1153, 1154, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    5  0.5859960  0.3148693
##    7  0.5855275  0.3137430
##    9  0.5951555  0.3289138
##   11  0.6144413  0.3607572
##   13  0.6123922  0.3539459
##   15  0.6079530  0.3453734
##   17  0.6029986  0.3369037
##   19  0.6123723  0.3503460
##   21  0.6105454  0.3451601
##   23  0.6128908  0.3472710
##
```

```
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 11.
```

From the results, we can see that the best k is 11.

```r
## Test prediction
test_pred <- predict(knn_fit, newdata = testing)

## Confusion Matrix
confusionMatrix(test_pred, as.factor(testing$quality_group))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction high low medium
##     high     23   5     12
##     low       5 101     47
##     medium   15  42     68
##
## Overall Statistics
##
##                Accuracy : 0.6038
##                  95% CI : (0.5477, 0.6579)
##     No Information Rate : 0.4654
##     P-Value [Acc > NIR] : 5.037e-07
##
##                   Kappa : 0.3419
##
##  Mcnemar's Test P-Value : 0.8932
##
## Statistics by Class:
##
##                      Class: high Class: low Class: medium
## Sensitivity              0.53488     0.6824        0.5354
## Specificity              0.93818     0.6941        0.7016
## Pos Pred Value           0.57500     0.6601        0.5440
## Neg Pred Value           0.92806     0.7152        0.6943
## Prevalence               0.13522     0.4654        0.3994
## Detection Rate           0.07233     0.3176        0.2138
## Detection Prevalence     0.12579     0.4811        0.3931
## Balanced Accuracy        0.73653     0.6883        0.6185
```

From the confusion matrix we see that our model had 0.6038 accuracy.

Use multinomial logistic regression to classify the same dataset

```r
library(nnet)

# Fitting the multinomial logistic regression
winefit <- multinom(quality_group~., data=training)
```

```
## # weights:  39 (24 variable)
## initial  value 1407.322342
```

```
## iter  10 value 1184.213664
## iter  20 value 979.174992
## iter  30 value 973.994223
## iter  40 value 973.933351
## iter  50 value 971.814088
## final  value 971.814031
## converged
```

```r
summary(winefit)
```

```
## Call:
## multinom(formula = quality_group ~ ., data = training)
##
## Coefficients:
##        (Intercept) fixed.acidity volatile.acidity citric.acid residual.sugar
## low      -281.5893    -0.3182723         4.509583   0.1676551     -0.2713117
## medium   -183.7232    -0.1471474         1.563250  -1.0644794     -0.2164664
##        chlorides free.sulfur.dioxide total.sulfur.dioxide  density           pH
## low    10.559661         -0.04070430           0.03696798 300.9033 -0.397972264
## medium  8.996567         -0.01357061           0.01761101 193.4952  0.006589013
##        sulphates    alcohol
## low    -5.299177 -1.2087676
## medium -3.031606 -0.5028761
##
## Std. Errors:
##        (Intercept) fixed.acidity volatile.acidity citric.acid residual.sugar
## low       2.168026    0.10454196        0.9356539   1.0512518     0.07562658
## medium    1.943305    0.09081303        0.8557318   0.9497005     0.07101414
##        chlorides free.sulfur.dioxide total.sulfur.dioxide density        pH
## low     4.400036          0.01576432          0.006771179 2.11475 1.1051207
## medium  4.265742          0.01429993          0.006412357 1.89933 0.9969161
##        sulphates   alcohol
## low    0.7053950 0.1262062
## medium 0.6132111 0.1036924
##
## Residual Deviance: 1943.628
## AIC: 1991.628
```

We need to convert the coefficients to the exponents to interpret their effects on the odds ratio.

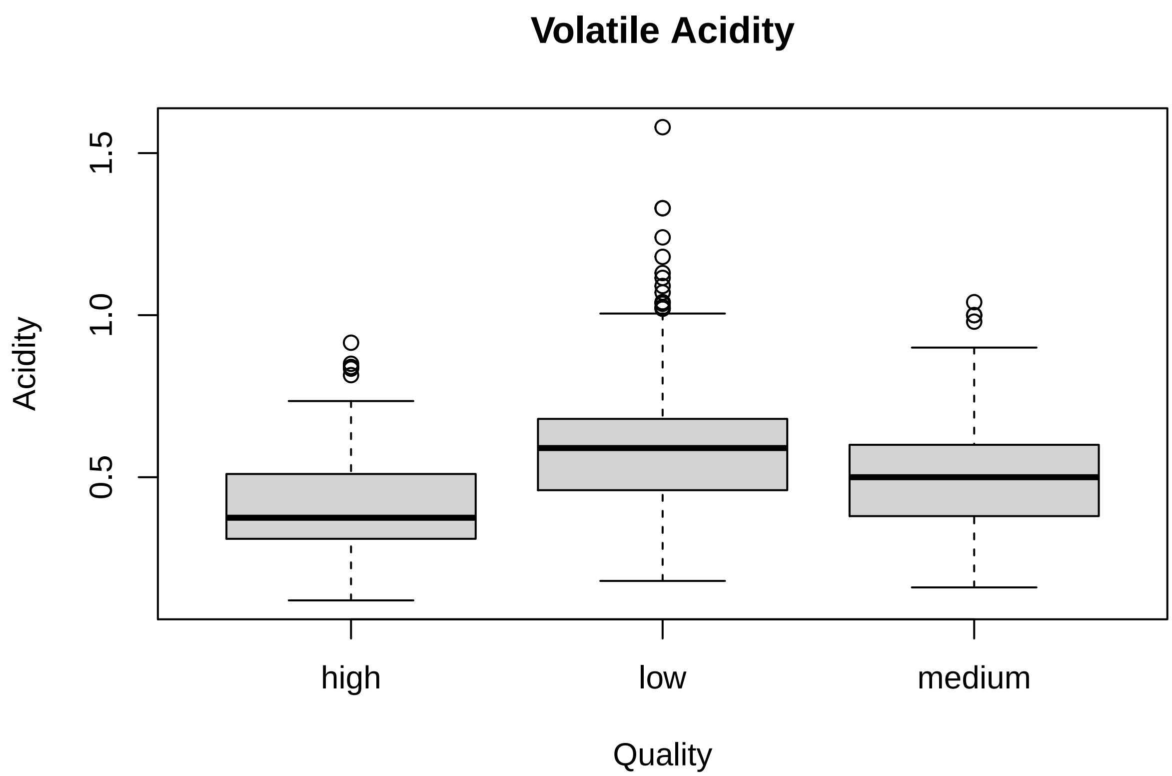
```r
exp(coef(winefit))
```

```
##          (Intercept) fixed.acidity volatile.acidity citric.acid residual.sugar
## low    5.097234e-123     0.7274047        90.883881   1.1825287      0.7623789
## medium  1.621929e-80     0.8631667         4.774313   0.3449074      0.8053596
##        chlorides free.sulfur.dioxide total.sulfur.dioxide       density
## low    38548.055           0.9601130             1.037660 4.793165e+130
## medium  8075.312           0.9865211             1.017767  1.081129e+84
##               pH   sulphates   alcohol
## low    0.6716807 0.004995704 0.2985650
## medium 1.0066108 0.048238108 0.6047887
```

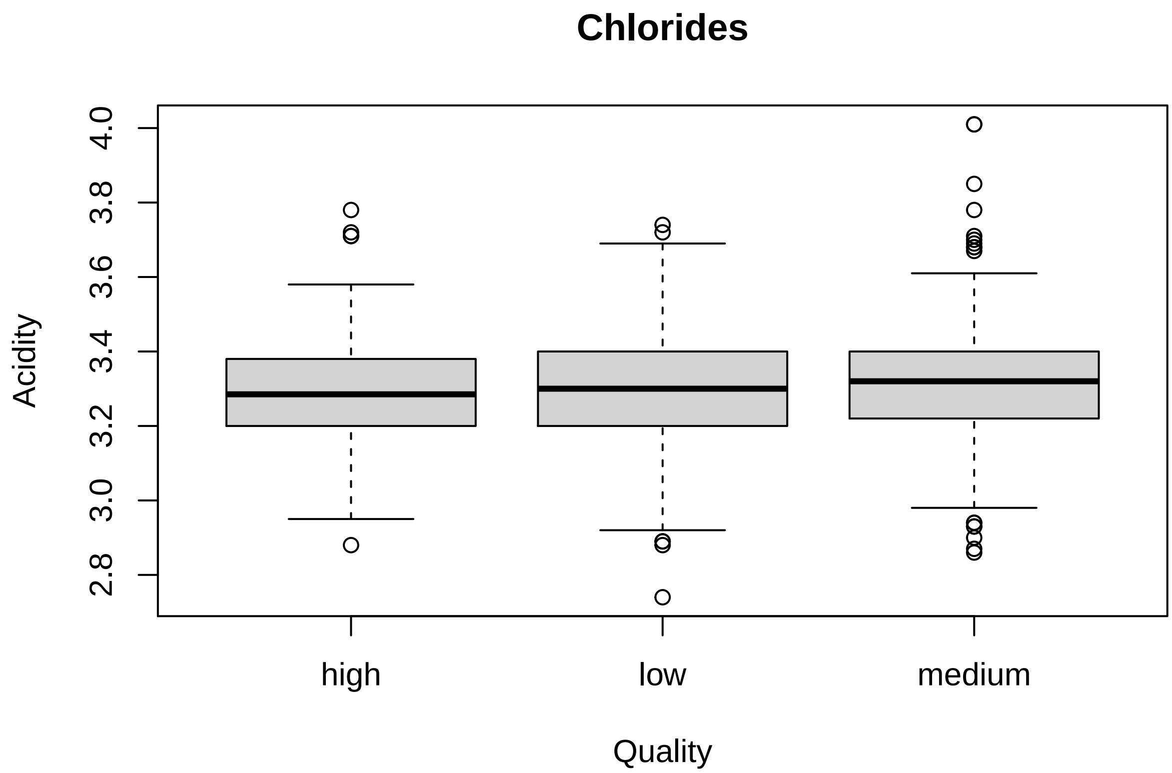
We see few variables having different effects on making it low or medium, especially the volatile acidity and chlorides.

High value of volatile acidity and chlorides increase the odds ratio of it being low by the value shown above. These effects are clearly different for low and medium quality.

```
boxplot(volatile.acidity~quality_group,
        data=training, main="Volatile Acidity",
        xlab="Quality", ylab="Acidity")
```




```
boxplot(pH~quality_group,
        data=training, main="Chlorides",
        xlab="Quality", ylab="Acidity")
```



Even though the multinomial model says, high chlorides correspond to low quality wine, we don't see a direct correlation from the box plot.

There are interactions happening within the variables that we need to look further to fine tune this.

```
test_pred_multinom <- predict(winefit, newdata = testing, "class")

# Building classification table
tab <- table(testing$quality_group, test_pred_multinom)

# Calculating accuracy - sum of diagonal elements divided by total obs
round((sum(diag(tab))/sum(tab))*100,2)
```

```
## [1] 66.04
```

We see the model is 66% accurate in predicting the right quality group based on the features considered on a test dataset.

K-means clustering

To start with, we need to scale the variables before we cluster

```
## scale the data
xtraining <- scale(training[,0:11])
```

Let's cluster using k-means using k=3.

```r
set.seed(23)
wine_quality <- kmeans(xtraining, centers=3, nstart=5)
print(wine_quality$centers)
```

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar   chlorides
## 1   -0.64703121       0.40502695 -0.72542261    -0.21373566 -0.18666483
## 2   -0.08097507       0.08579846  0.07166127     0.38540530  0.04063714
## 3    1.05004399      -0.68420039  1.05281742     0.03112648  0.25390500
##   free.sulfur.dioxide total.sulfur.dioxide    density         pH  sulphates
## 1          -0.2109782           -0.3407575 -0.4744268  0.5965107 -0.2471806
## 2           1.0789255            1.3270346  0.3436003 -0.1274584 -0.1669265
## 3          -0.5043670           -0.4962632  0.4612254 -0.8132255  0.5053197
##       alcohol
## 1  0.08488205
## 2 -0.54050464
## 3  0.28444844
```

Cluster 1 has high volatile acidity, high pH (relatively), least fixed acidity, citric acid, chlorides, density and sulphates. Cluster 3 is exactly on the other end of the spectrum based on the above mentioned characteristics of Cluster 1 and Cluster 2 is in between.

We can't really compare the results of clustering with the supervised approaches because clustering is a unsupervised algorithm and the clusters that are formed aren't necessarily about wine quality as we defined in the supervised cases.

Clusters are somethings that the model came up with and we need to interpret the characteristics of it and come up with a name for each.

Describe the three approaches (knn, multinomial logistic regression, and k-means) and compare/contrast them with each other. KNN refers to K-Nearest Neighbors. The intuition of this model is simple. For each point to be classified we: Look at the classification of K nearest records. Those are the neighbors with similar features. For classification, we find the classification proportion of the closest records and assign the majority class to the new record. For regression, we use the average of the K nearest neighbors and predict this value to the new point KNN is a supervised ML model, so we need a response variable to train the model.

Multinomial Logistic Regression is an extension of the traditional Logistic Regression. It also uses the maximum likelihood estimation (MLE) to estimate the probability of the records belonging to each class, but it includes the possibility of having more than two outcomes.

K-means is a clustering technique used to divide the data into different subgroups. The main objective is to identify meaningful groups of data. K-means do that by minimizing the sum of the squared distance of each point to the mean of its cluster. Unlike KNN, K-means is an unsupervised technique, which means it trains the data without a response variable present

Which approach would you recommend? We see that the k-NN is a naive approach to classify an outcome based on training dataset since it comes with its own limitation of following the majority rule when it's not about the majority and rather about the proximity to the nearest neighbors.

Multinomial logistic regression does relatively better than the k-NN as expected. With more refinements to the model by doing variable selection, we can improve the model and interpret the results better.

Since clustering is an exploratory approach when we don't have an outcome class we are interested in, it is not the best approach for this use case. Even though we understand the innate characteristics about different wines and how they compare with each other, it can be used to interpret and refine the multinomial model further. Hence, we would recommend the best way forward is the multinomial approach as it can help us predict better and interpret the influence of individual variables to the outcome.