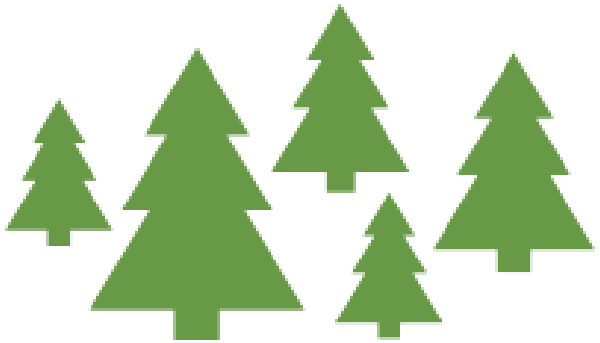
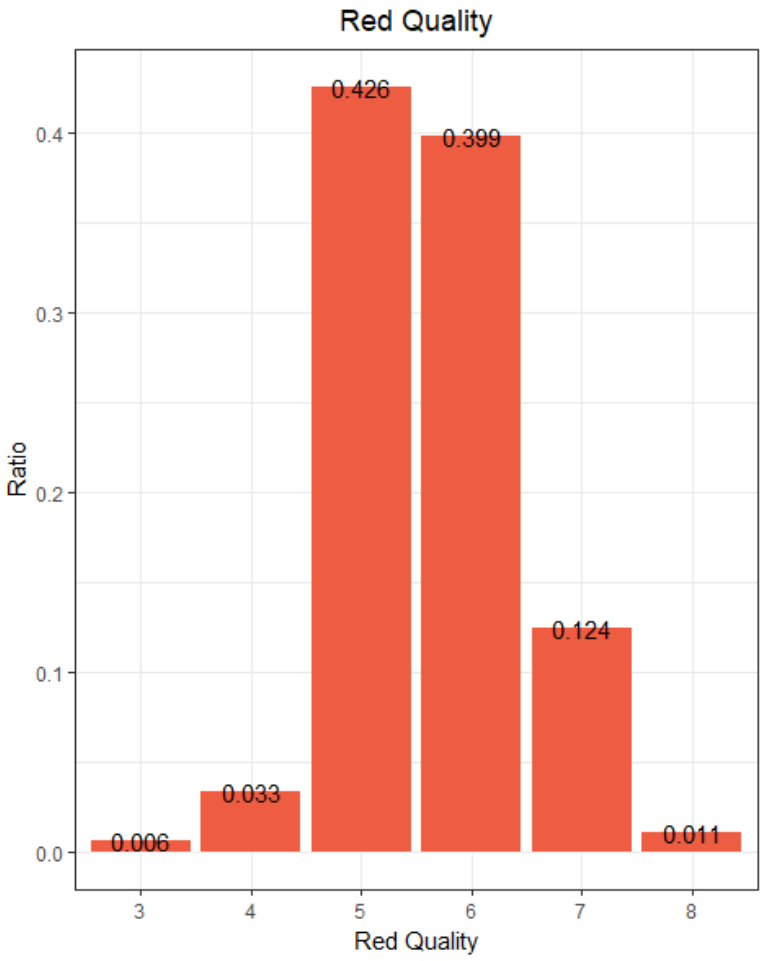
紅白酒分類問題

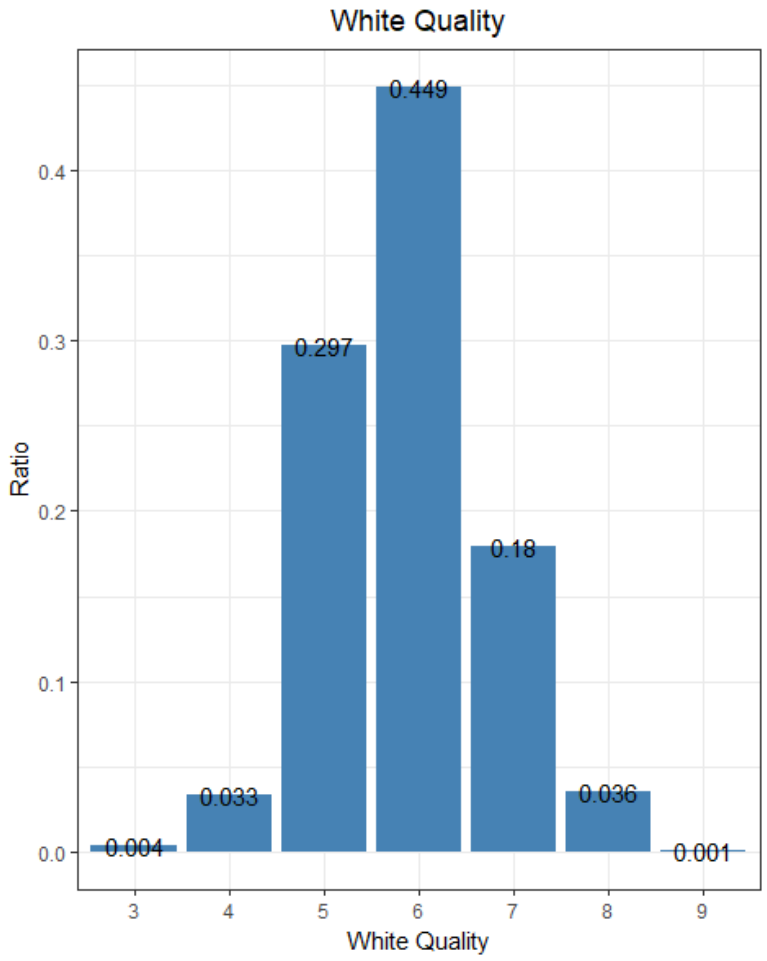
利用隨機森林模型



資管五 郭泰竹 102306037

探索性分析 - 紅白酒品質分布 ➔ 了解品質分布狀況





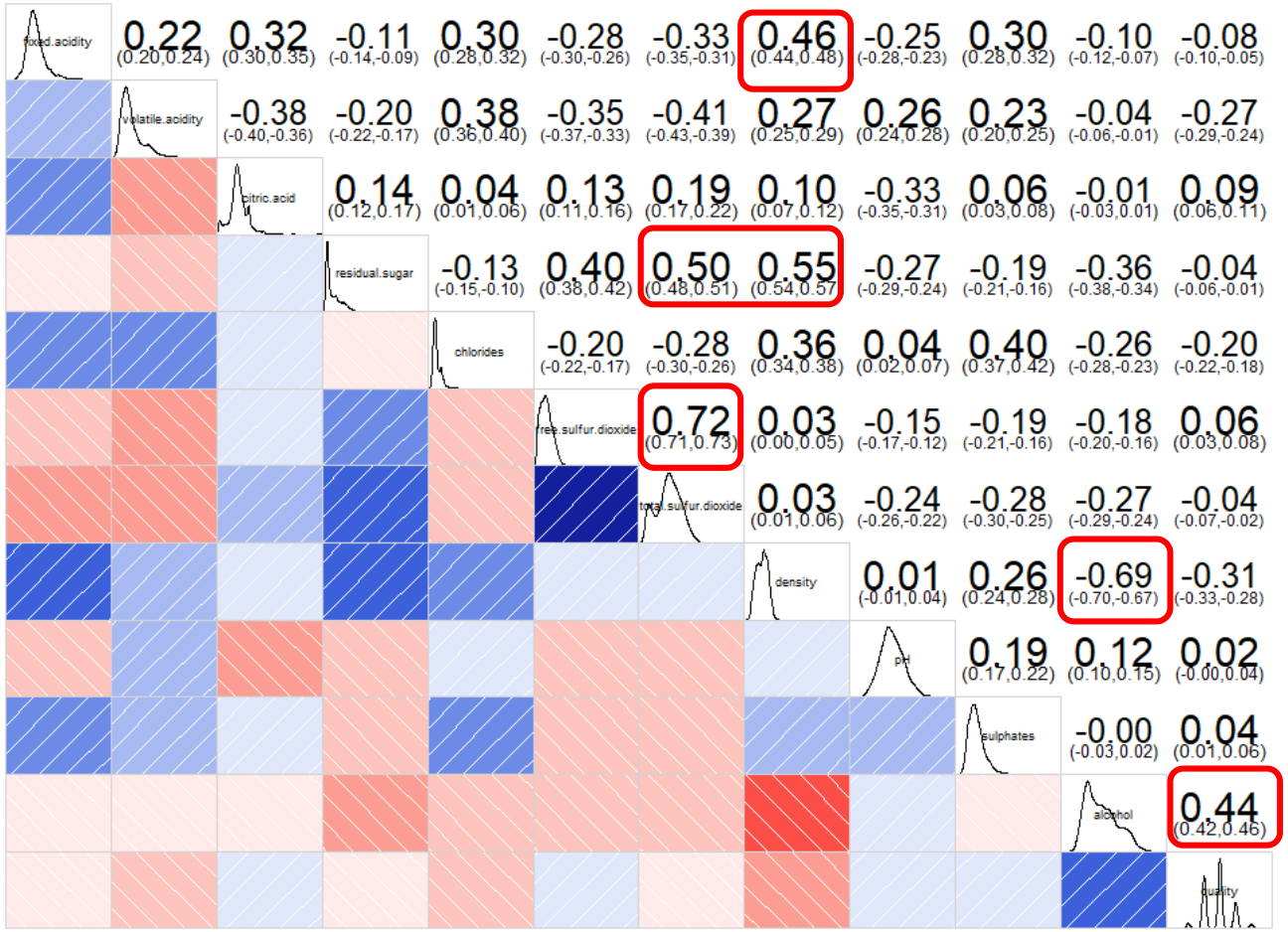
說明

紅酒分布以第五級、第六級最多

白酒分布以第六級為最多

➔等級分布上可能有些微差異

探索性分析 - 相關係數矩陣圖➔觀察變數彼此間影響程度



說明

固定酸度和密度有高度正相關

糖分和密度有高度正相關

總二氧化硫和游離二氧化硫高度正相關

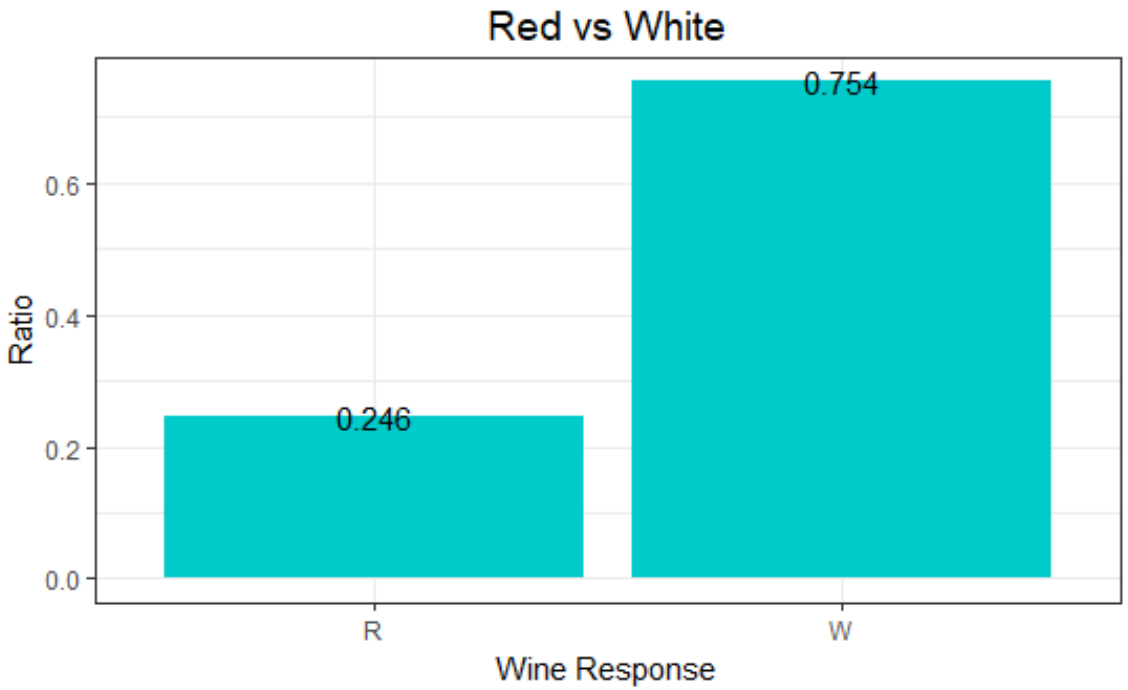
總二氧化硫和糖分高度正相關

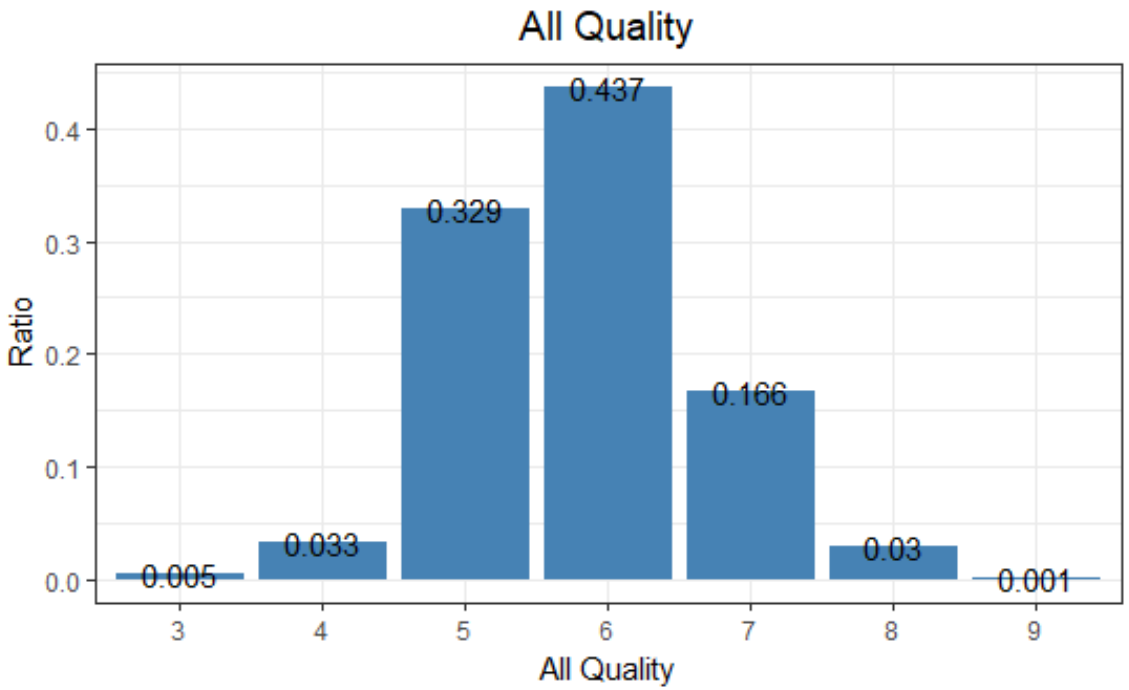
酒精濃度和密度高度負相關

品質和酒精濃度高度正相關

➔密度和許多因素都有相關，可能成為代表性的特徵

合併紅白酒資料 - 探索性分析



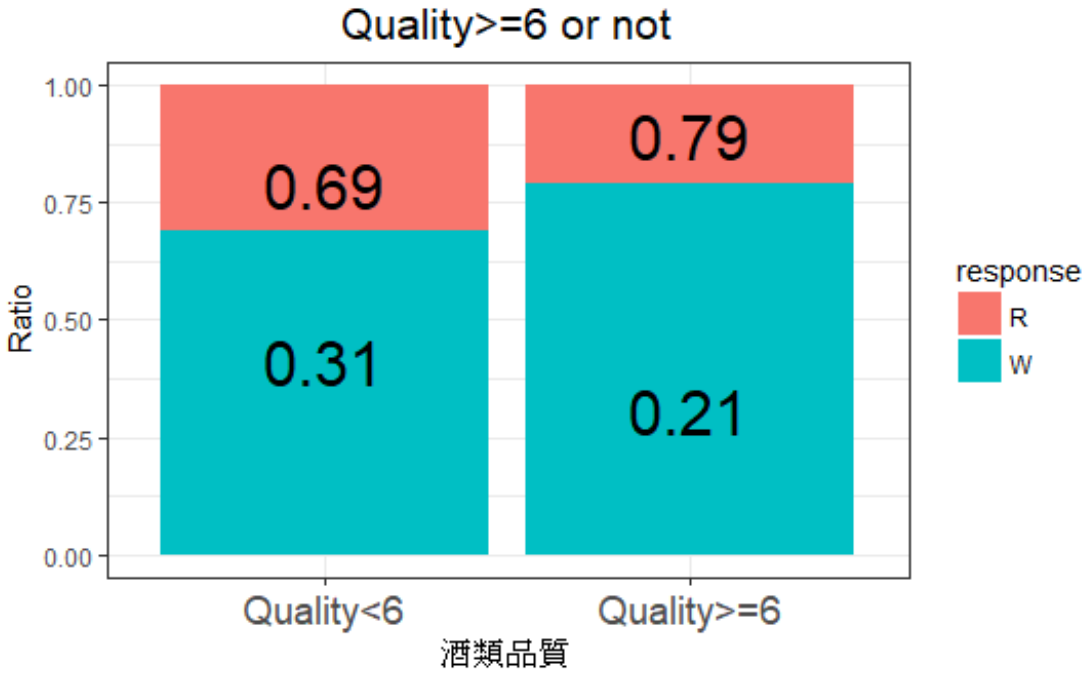


說明

(1) 紅白酒資料合併後，比例上有明顯差異，紅酒對白酒約為1:3，在配模及切割訓練資料時須注意。

(2) 合併後的資料中，Quality以5,6為最多，可能可當作分界點判斷紅白酒。

合併紅白酒資料 - Feature engineering




```
> table(all$GoodQuality,all$response)

      R    W
  0  744 1640
  1  855 3258
```

```
> chisq.test(table(all$GoodQuality,all$response))

        Pearson's Chi-squared test with Yates' continuity
        correction

data:  table(all$GoodQuality, all$response)
X-squared = 87.762, df = 1, p-value < 2.2e-16
```

說明

利用Quality是否大於六作為分界，製作二元的特徵變數，經過卡方檢定確認此特徵對紅白酒有顯著差異。

資料分割 - Train and Test set split

```
#共有6497筆資料，拿6000筆為train，497筆為test
train_red = all[sample(which(all$response=="R"),6000*0.246),]
train_white = all[sample(which(all$response=="W"),6000*0.754),]
train_set = rbind(train_red,train_white)
test_set = all[-as.numeric(rownames(train_set)),]
```

說明

(1) 共有6497筆資料，取6000筆為training data，497筆為testing data。

(2) 分割時按照紅白酒1:3的比例切割，使train 和 test 當中紅白酒的比例都是1:3

資料建模 - Random Forest 隨機森林模型

原理

(1) 用隨機的方式建立決策樹森林，裏頭包含各個決策樹，之間並無關聯。

(2) 利用投票的方式，決定分類的答案，因為樹夠多，能夠減少噪音的產生，並涵蓋所有情況。

(3) 由於每個樹都是隨機獨立的，能夠降低over-fitting的現象(隨機產生樹，每個樹的訓練集都不同)

優點

(1) 能夠處理高維特徵的訓練資料，並無須降維

(2) 無須進行交叉驗證，在生成過程中能夠獲取內部生成誤差

(3) 對於skewed型態的資料有好的處理效果

(4) 能夠知道各變數在分類問題上的重要性

資料建模 - Random Forest (original)

```
Call:
 randomForest(formula = response ~ ., data = train_set[, -14])
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 0.47%
Confusion matrix:
     R    W class.error
R 1454   22  0.01490515
W    6 4518  0.00132626
```

說明

(1) 訓練誤差率➔0.47%

(2) 混淆矩陣 ➔ R失誤率1.49%，W失誤率0.13%

OOB estimate error(Out Of Bag):

隨機森林的好處之一就是無需做交叉驗證，他會在內部取另外約1/3的tree做樣本，並進行分類計算失誤率，最後生成新的樹。

資料建模 - Random Forest (plus new feature)

```
Call:
 randomForest(formula = response ~ ., data = train_set)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 0.48%
Confusion matrix:
     R    W class.error
R 1455   21 0.014227642
W    8 4516 0.001768347
```

說明

(1) 訓練誤差率➔0.48% ，相差不多

(2) 混淆矩陣 ➔ R失誤率1.42%，W失誤率0.17%

結果解釋 - Testing set

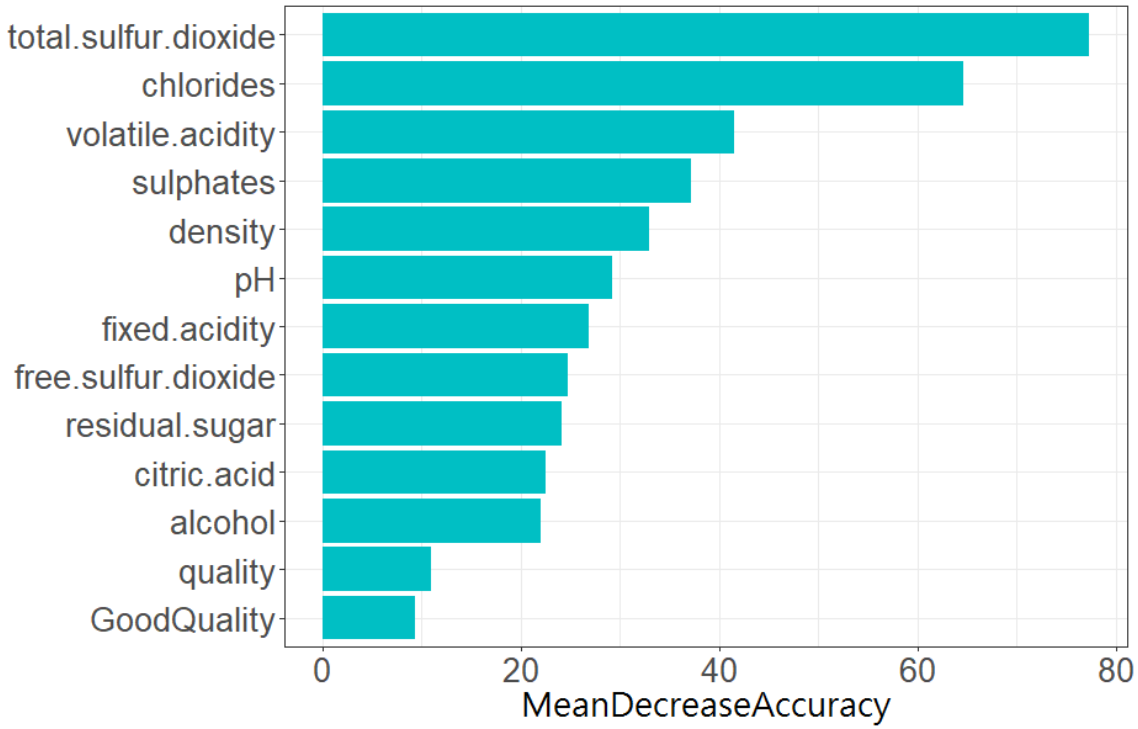
```
> table(test_set$response,test_plus)
   test_plus
      R   W
  R 119   4
  W   0 374
```

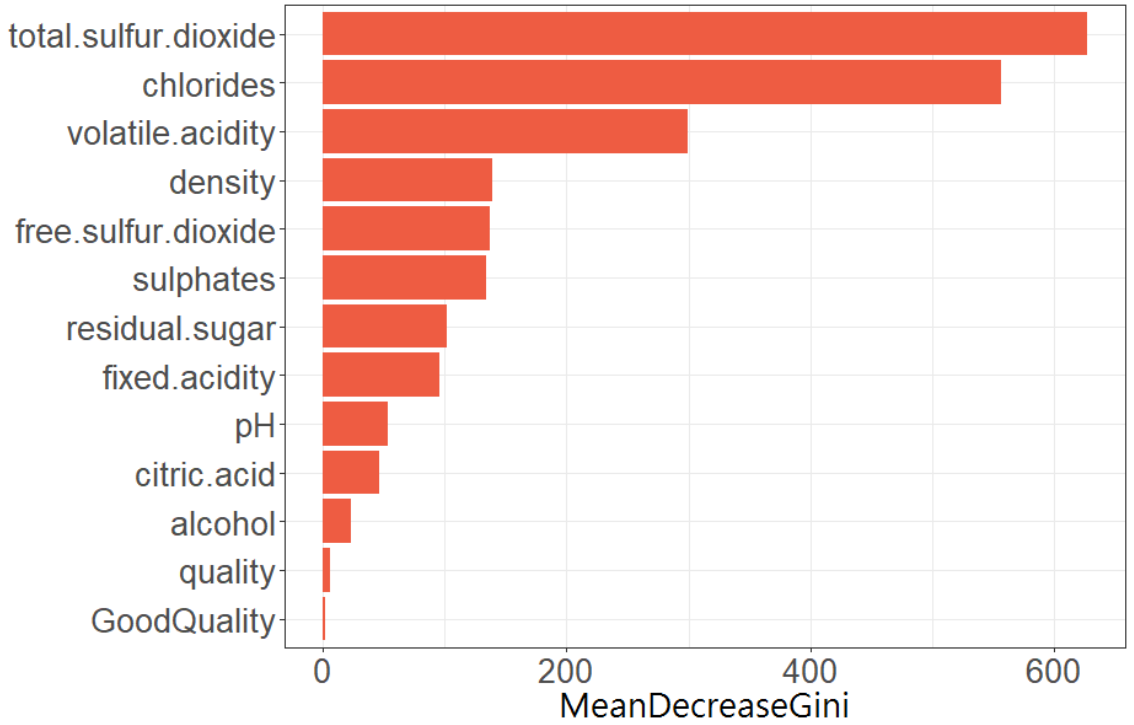
說明

(1) 測試誤差率➔0.6%

(2) 混淆矩陣 ➔ R 失誤率 3.2%，W 失誤率0 %

結果解釋 - Importance plot





說明

下列兩個值愈大，該特徵變數對於該模型的判別影響愈大

(1) MeanDecreaseAccuracy ➔利用permute的方法，單獨對每個特徵的值進行改變，然後對所有樹計算平均差距。

(2) MeanDecreaseGini ➔計算每棵樹的每個特徵在分類上的提升，然後對所有樹計算權重。

簡單來說，Accuracy就是改變這個特徵的值會造成多大影響，Gini就是這個特徵可以提升多少效能。

變數篩選 - Feature Selection

```
[1] "volatile.acidity"     "chlorides"
[3] "total.sulfur.dioxide" "density"
[5] "response"

Call:
 randomForest(formula = response ~ ., data = select_train)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 0.87%
Confusion matrix:
     R    W class.error
R 1440   36 0.024390244
W   16 4508 0.003536693
```

說明

篩選表現前四好的變數，並建立新模型，訓練誤差率也在1%以下。

(1) 測試誤差率➔0.6%

(2) 混淆矩陣 ➔ R 失誤率 3.2%，W 失誤率0 %

```
> table(test_set$response,test_select)
   test_select
      R   W
  R 119   4
  W   0 374
```

Thanks for listening