H34091071 郭哲豪 # HW 2

一、資料簡介

此數據集在探討來自葡萄牙北部的兩個紅和白vinho verde葡萄酒樣品，紅、白酒分別有4898和1599個 instance。 它們都包含相同的attribute，且數據收集的定義明確，並無丟失的數據。

fixed acidity : 固定酸度，在葡萄酒中發現的非揮發性酸為酒石酸、蘋果酸、檸檬酸和琥珀酸。除琥珀酸外，所有這些酸都起源於葡萄，而琥珀酸是在發酵過程中由酵母產生的。

volatile acidity : 揮發性酸度，葡萄酒中的酸性元素，是氣態而不是液態，因此可以感覺到是一種氣味，顯示出一種香氣，而不是在味蕾上發現。

citric acid : 檸檬酸，一種弱有機酸，通常用作食品或飲料的天然防腐劑或添加劑，以增加食品的酸味和新鮮度。

residual sugar : 殘留糖分，發酵停止後殘留的糖量，很少找到少於1克/升的酒，而超過45克/升的酒則被認為是甜酒。

chlorides : 氯化物，酒中鹽的含量。

free sulfur dioxide(SO2) : 遊離二氧化硫，在釀酒過程的所有階段都使用SO2，以防止氧化和微生物生長。 過量的SO2會抑制發酵並引起不良的感官作用。

total sulfur dioxide : 二氧化硫總量，包含 SO2 的遊離形式和結合形式；在低濃度下，葡萄酒中幾乎檢測不到 SO2，但是當遊離 SO2 濃度超過 50 ppm 時，葡萄酒的香氣和口感中會明顯出現 SO2。

density : 密度，葡萄酒的密度根據其酒精和糖分含量的百分比而接近於水的密度。

pH : 釀酒師使用pH值來衡量相對於酸度的成熟度。 低pH值的酒可品味酸澀和爽脆，而高pH值的酒會更容易受到細菌的滋生。 大多數葡萄酒的pH值約為3或4。

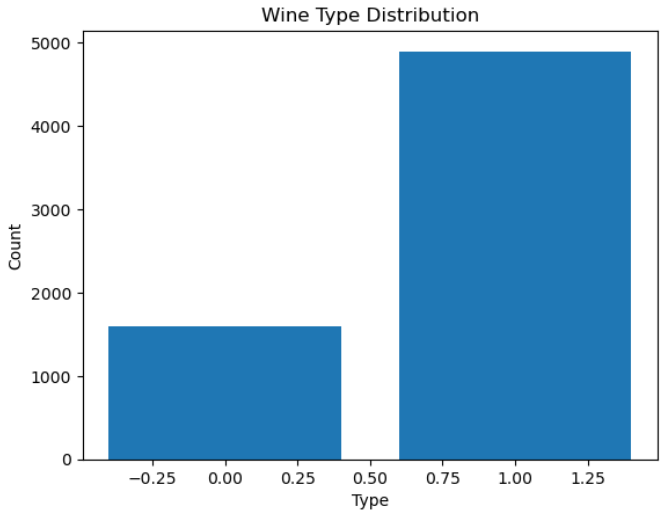
sulphates : 硫酸鹽，一種葡萄酒添加劑，可提高二氧化硫氣體(SO2)的含量，起抗菌劑和抗氧化劑的作用。

alcohol : 葡萄酒中酒精濃度的百分比。

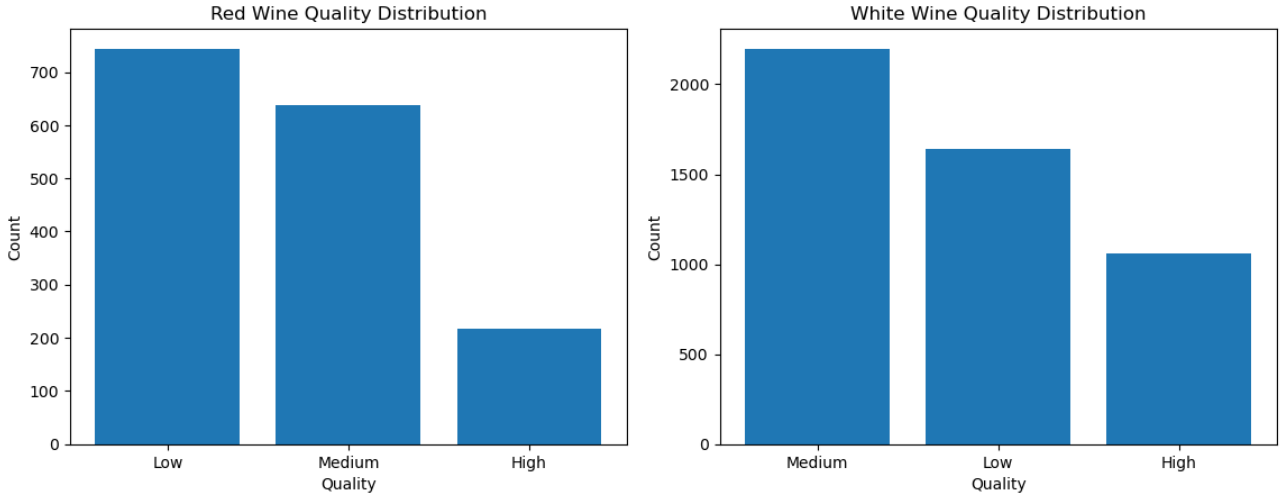
quality : 描述紅白葡萄酒的質量等級。

二、分析概念說明

1. 將紅白酒的資料彙整在一起，加入 type 欄位，分別紀錄其為紅酒還是白酒，針對紅白酒的 binary classification problem，利用 wine 的 attribute 去做 Cross Validation Method，比較 logistic regression model、LDA、QDA、K-NN 以及 random forest model 之間的表現，下圖為紅白酒的 type 資料分布



2. 使用紅、白酒分別的資料，將 quality 分為 Low, Median, High 三個區間，針對紅、白酒的 quality 預測，利用 wine 的 attribute 去做 Cross Validation Method，比較 logistic regression model、LDA、QDA、K-NN 以及 random forest model 之間的表現，下圖為紅、白酒的 quality 資料分布



三、分析結果

1. 利用 python 來做此分析

```
from sklearn.linear_model import LogisticRegression
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score, KFold
from sklearn.datasets import load_iris
import pandas as pd
import matplotlib.pyplot as plt
```

匯入資料集，並將 type 欄中的 red 轉為 0, white 轉為 1

```
# Load the winequality dataset
df = pd.read_csv("winequality.csv")

# Split the data into features (X) and target variable (y)
X = df[['fixed acidity','volatile acidity','citric acid','residual sugar','chlorides','free sulfur dioxide','total s
type_mapping = { 'red' : 0,'white' : 1}
y =  df['type'].map(type_mapping)
```

利用 Logistic Regression model ，做五次 cross-validation 分析，並得到平均的 accuracy

```
# Define the logistic regression model
logreg = LogisticRegression(solver='liblinear', multi_class='ovr')

# Define the number of folds for cross-validation
n_folds = 5

# Define the cross-validation method
cv_method = KFold(n_splits=n_folds, shuffle=True, random_state=42)

# Use cross-validation to evaluate the model
scores = cross_val_score(logreg, X, y, cv=cv_method)

# Print the accuracy scores for each fold
print("Accuracy scores for each fold:\n", scores)

# Calculate and print the mean accuracy score across all folds
print("\nMean accuracy score:", scores.mean())
```

```
Accuracy scores for each fold:
 [0.97461538 0.98615385 0.97690531 0.98614319 0.9830639 ]

Mean accuracy score: 0.9813763249837153
```

利用 LDA model，做五次 cross-validation 分析，並得到平均的 accuracy

```
# Define the LDA model
lda = LinearDiscriminantAnalysis()

# Define the number of folds for cross-validation
n_folds = 5

# Define the cross-validation method
cv_method = KFold(n_splits=n_folds, shuffle=True, random_state=42)

# Use cross-validation to evaluate the model
scores = cross_val_score(lda, X, y, cv=cv_method)

# Print the accuracy scores for each fold
print("Accuracy scores for each fold:\n", scores)

# Calculate and print the mean accuracy score across all folds
print("\nMean accuracy score:", scores.mean())
```

```
Accuracy scores for each fold:
 [0.99384615 0.99615385 0.99461124 0.99615089 0.99230177]

Mean accuracy score: 0.9946127790608159
```

利用 QDA model，做五次 cross-validation 分析，並得到平均的 accuracy

```
# Define the QDA model
qda = QuadraticDiscriminantAnalysis()

# Define the number of folds for cross-validation
n_folds = 5

# Define the cross-validation method
cv_method = KFold(n_splits=n_folds, shuffle=True, random_state=42)

# Use cross-validation to evaluate the model
scores = cross_val_score(qda, X, y, cv=cv_method)

# Print the accuracy scores for each fold
print("Accuracy scores for each fold:\n", scores)

# Calculate and print the mean accuracy score across all folds
print("\nMean accuracy score:", scores.mean())
```

```
Accuracy scores for each fold:
 [0.98538462 0.98769231 0.9830639  0.98460354 0.98537336]

Mean accuracy score: 0.9852235447385563
```

利用 KNN model，做五次 cross-validation 分析，並得到平均的 accuracy

```
# Define the KNN model
knn = KNeighborsClassifier(n_neighbors=5)

# Define the number of folds for cross-validation
n_folds = 5

# Define the cross-validation method
cv_method = KFold(n_splits=n_folds, shuffle=True, random_state=42)

# Use cross-validation to evaluate the model
scores = cross_val_score(knn, X, y, cv=cv_method)

# Print the accuracy scores for each fold
print("Accuracy scores for each fold:\n", scores)

# Calculate and print the mean accuracy score across all folds
print("\nMean accuracy score:", scores.mean())
```

```
Accuracy scores for each fold:
 [0.93615385 0.94153846 0.93148576 0.95150115 0.94842186]

Mean accuracy score: 0.9418202167347663
```

利用 Random Forest model，做五次 cross-validation 分析，並得到平均的 accuracy

```
# Define the random forest model
rf = RandomForestClassifier(n_estimators=100, random_state=42)

# Define the number of folds for cross-validation
n_folds = 5

# Define the cross-validation method
cv_method = KFold(n_splits=n_folds, shuffle=True, random_state=42)

# Use cross-validation to evaluate the model
scores = cross_val_score(rf, X, y, cv=cv_method)

# Print the accuracy scores for each fold
print("Accuracy scores for each fold:\n", scores)

# Calculate and print the mean accuracy score across all folds
print("\nMean accuracy score:", scores.mean())
```

```
Accuracy scores for each fold:
 [0.99692308 0.99384615 0.99538106 0.99692071 0.99384142]

Mean accuracy score: 0.995382483567241
```

2. 將 quality 區分為三群，{3,4,5}為 Low ，{6}為 Median ，{7,8,9}為 High

```
# Load the winequality dataset
df = pd.read_csv("winequality-red.csv")

# Split the data into features (X) and target variable (y)
X = df[['fixed acidity','volatile acidity','citric acid','residual sugar','chlorides','free sulfur dioxide','total s
quality_mapping = { 3 : "Low",4 : "Low",5: "Low",6 : "Medium",7: "High",8 : "High",9 : "High"}
y =  df['quality'].map(quality_mapping)
```

分別利用 Logistic Regression、LDA、QDA、KNN、Random Forest model，對紅酒、白酒做五次 cross-validation 分析，並得到平均的 accuracy

Red Wine:

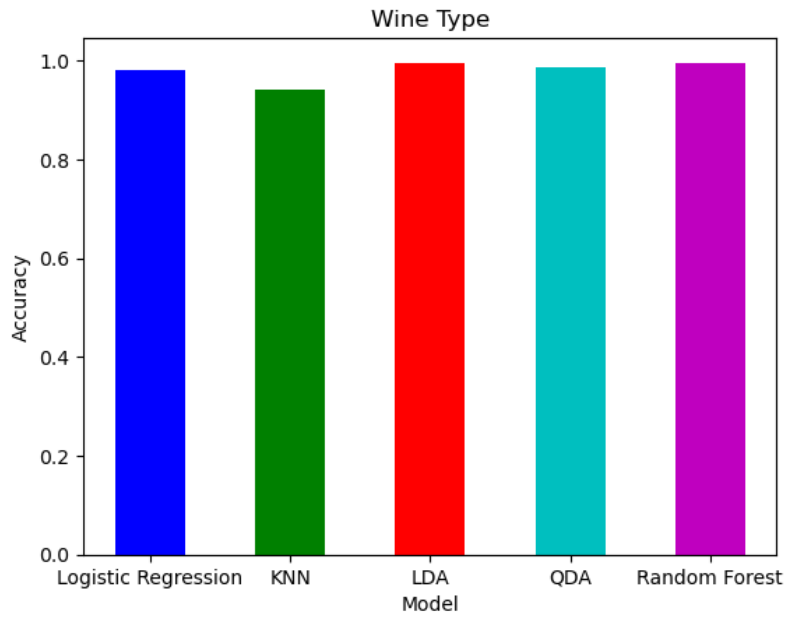
```
Logistic Regression Mean accuracy score: 0.6179447492163008
LDA Mean accuracy score: 0.6285638714733544
QDA Mean accuracy score: 0.6022805642633229
KNN Mean accuracy score: 0.5234541536050157
Random Forest Mean accuracy score: 0.7254760971786833
```

White Wine:

```
Logistic Regression Mean accuracy score: 0.5677888724437681
LDA Mean accuracy score: 0.575138729649163
QDA Mean accuracy score: 0.5291984740780888
KNN Mean accuracy score: 0.5108177857455545
Random Forest Mean accuracy score: 0.7268264159596424
```

四、結論

1. 在分析紅白酒的 binary classification problem 時，各個 model 的 accuracy 差異不會太大，都有維持在 90% 以上的水準，只有 KNN 較為差強人意



2. 在預測紅、白酒的 quality 時，White Wine 的預測準確度整體較 Red Wine 來得高一些，各個 model 的 accuracy 比較為 Random Forest > LDA > Logistic Regression > QDA > KNN，而其中又以 Random Forest 為最穩定的 model ，分別在紅、白酒的預測準確度差異不大