Project：Red Wine Quality

Team 3 40541204S 梁湘梅

1. 資料集名稱: Red Wine Quality Data Set

2. 資料集來源: https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009

3. 資料集特性: Multivariate

4. 資料集說明:

包括來自葡萄牙北部的紅色葡萄酒樣品有關的數據集。目標是根據物理化學(性質)測試對葡萄酒質量進行建模。

5.project 目的:

Random Forest Classifier、Stochastic Gradient Descent Classifier、Support Vector Classifier(SVC)三種分類器的 Accuracy 比較。

6. Attributes Information(12 個)：

[表一]

	資料項目	中文	屬性特性	說明
1.	fixed acidity	固定酸度	Ratio	與酒有關的大多數酸或固定或不揮發的酸（不易蒸發）
2.	volatile acidity	揮發性酸度	Ratio	葡萄酒中乙酸的含量，含量過高會導致令人不快的醋味
3.	citric acid	檸檬酸	Ratio	檸檬酸含量低，可為葡萄酒增添“新鮮度”和風味
4.	residual sugar	殘留糖	Ratio	發酵停止後剩餘的糖量，很少發現少於 1 克/升的葡萄酒，而超過 45 克/升的葡萄酒被認為是甜的
5.	chlorides	氯化物	Ratio	葡萄酒中的鹽含量
6.	free sulfur dioxide	游離二氧化硫	Ratio	SO_2 的游離形式在分子 SO_2（作為溶解氣體）和亞硫酸氫根離子之間處於平衡狀態。它可以防止微生物的生長和葡萄酒的氧化
7.	total sulfur dioxide	總二氧化硫	Ratio	游離和結合形式的 SO_2 的量；在低濃度下，葡萄酒中幾乎檢測不到二氧化硫，但是當游離二氧化硫濃度超過 50 ppm 時，二氧化硫在葡萄酒的香氣和味道中變得明顯
8.	density	密度	Ratio	水的密度比較，具體取決於酒精和糖的含量

9.	pH	pH 值	Ratio	描述葡萄酒的酸性或鹼性程度從 0 到 14；大多數葡萄酒的 pH 值在 3-4 之間
10	sulphates	硫酸鹽	Ratio	一種葡萄酒添加劑，可提高 SO_2 的含量，可作為抗微生物劑和抗氧化劑
11	alcohol	酒精	Ratio	葡萄酒的酒精含量
12	quality	質量	Ordinal	輸出變量（基於感官數據，得分在 0 到 10 之間）

7.資料分析、前處理

- 先看資料檔案的形式。
  =>前 5 筆資料(如下圖所示)。

```
#Let's check how the data is distributed
wine.head()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

[圖一]

- 各欄位資訊。
  =>有 12 個欄位，1599 筆資料。

```
#Information about the data columns
wine.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
fixed acidity           1599 non-null float64
volatile acidity        1599 non-null float64
citric acid             1599 non-null float64
residual sugar          1599 non-null float64
chlorides               1599 non-null float64
free sulfur dioxide     1599 non-null float64
total sulfur dioxide    1599 non-null float64
density                 1599 non-null float64
pH                      1599 non-null float64
sulphates               1599 non-null float64
alcohol                 1599 non-null float64
quality                 1599 non-null int64
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
```

[圖二]

- 是否有 Missing value。
  =>**沒有** missing value。

```
wine.isnull().sum() # no null or Nan values.
```

```
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
dtype: int64
```

[圖三]

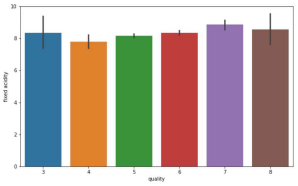
- 各欄位的數據分布。
  =>平均值、標準差...(如下所示)。

```
wine.describe(include='all')
```

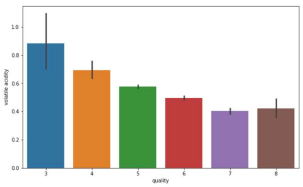
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	5.636023
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	0.807569
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.000000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	5.000000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	6.000000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	6.000000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.000000

[圖四]

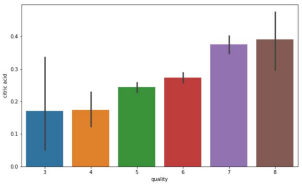
- 觀察各性質與 Quality 之間的關係。



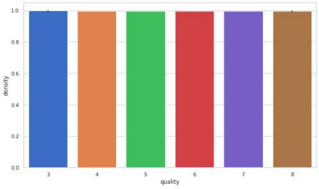
fixed acidity



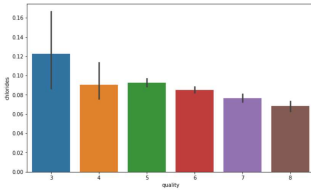
volatile acidity



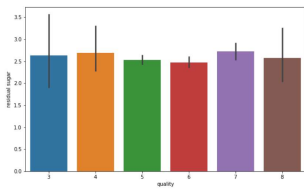
citric acid



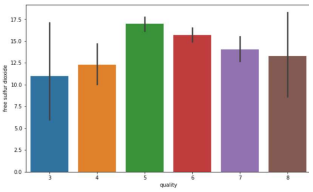
density



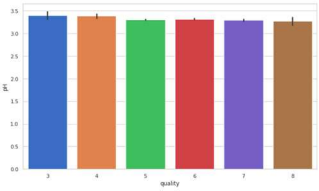
chlorides



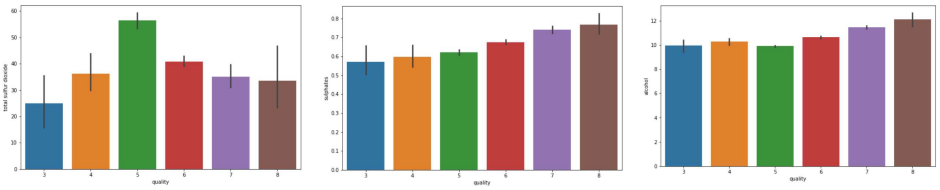
residual sugar



free sulfur dioxide



pH

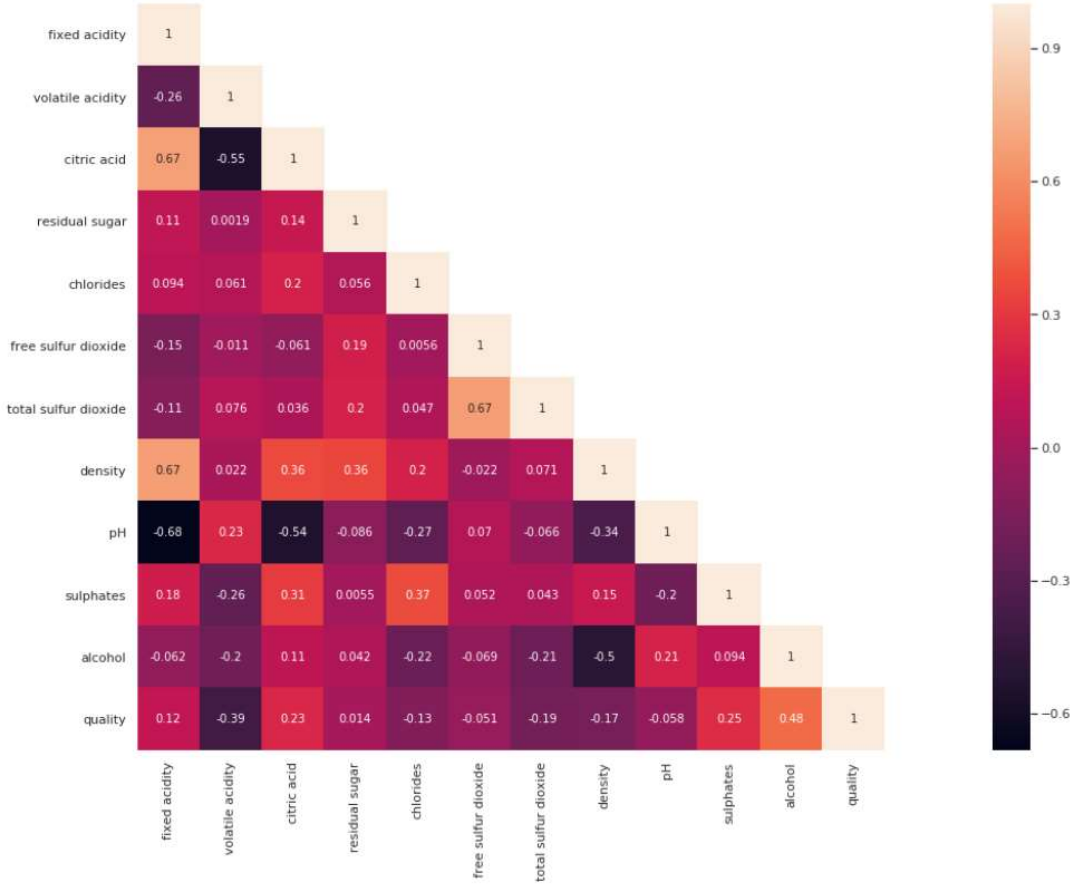


total sulfur dioxide　　sulphates　　alcohol　　[表二]

- 觀察各性質與性質之間的關係。

```
#corelation matrix.
cor_mat= wine.corr()
mask = np.array(cor_mat)
mask[np.tril_indices_from(mask)] = False
fig=plt.gcf()
fig.set_size_inches(30,12)
sns.heatmap(data=cor_mat,mask=mask,square=True,annot=True,cbar=True)
```

[圖五]

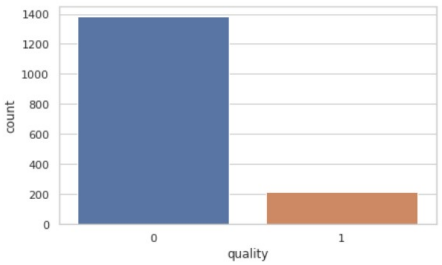


- 根據前面的觀察，我們推測：
  1. 固定的酸度、pH、密度與質量的關係甚小。
  2. 隨著質量的提高，揮發性酸度呈下降趨勢。
  3. 檸檬酸的成分隨著葡萄酒質量的提高而提高。
  4. 隨著葡萄酒質量的提高，氯化物的含量會下降。
  5. 硫酸鹽含量隨葡萄酒的質量而提高。
  6. 隨著葡萄酒質量的提高，酒精含量會升高。
  7. pH 和檸檬酸、固定酸度成反比關係(酸的 pH 值較小)。

- 對質量(應變量)進行二進制分類。根據自訂標準來區分葡萄酒的好壞。
  =>自訂標準：質量在 6.5 以下為 bad(0)；6.5 以上為 good(1)。執行結果如下所示：

```
bins = (2, 6.5, 8)
group_names = ['bad', 'good']
wine['quality'] = pd.cut(wine['quality'], bins = bins, labels = group_names)
label_quality = LabelEncoder()
wine['quality'] = label_quality.fit_transform(wine['quality'])
wine['quality'].value_counts()
```

```
0    1382
1     217
Name: quality, dtype: int64
```



[圖六]

- 將資料隨機分為訓練與測試資料

```
x = wine.drop('quality', axis = 1)
y = wine['quality']
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 42)
```

[圖七]

8.分類器

分類器的選擇上我們使用了 Random Forest Classifier、Stochastic Gradient Descent Classifier、Support Vector Classifier(SVC)三種分類器。

Random Forest Classifier 的優點在於平行化計算的特質，在資料集小或是大時的運算效能都不錯，且以不純度函數來切分樣本，因此不需要歸一化或標準化，簡化了建模時的步驟。下圖為 Random Forest Classifier 的執行結果，Accuracy 為 89%。

```
rfc = RandomForestClassifier(n_estimators=200)
rfc.fit(x_train, y_train)
pred_rfc = rfc.predict(x_test)
print(classification_report(y_test, pred_rfc))
print(confusion_matrix(y_test, pred_rfc))
```

```
              precision    recall  f1-score   support

           0       0.90      0.97      0.94       273
           1       0.70      0.40      0.51        47

    accuracy                           0.89       320
   macro avg       0.80      0.69      0.72       320
weighted avg       0.87      0.89      0.87       320

[[265   8]
 [ 28  19]]
```

[圖八]

Stochastic Gradient Descent Classifier 在使用時要先將資料進行歸一化：

```
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.fit_transform(x_test)
```

[圖九]

經常應用在大規模稀疏機器學習問題上，效率高且容易實現。每次迭代都隨機從訓練集中抽取出 1 個樣本，在樣本量極其大的情況下，可能不用抽取出所有樣本，就可以獲得一個損失值在可接受範圍之內的模型了。下圖為 Stochastic Gradient Descent Classifier 的執行結果，Accuracy 為 82%。

```
sgd = SGDClassifier(penalty=None)
sgd.fit(x_train, y_train)
pred_sgd = sgd.predict(x_test)
print(classification_report(y_test, pred_sgd))
print(confusion_matrix(y_test, pred_sgd))
```

```
              precision    recall  f1-score   support

           0       0.88      0.92      0.90       273
           1       0.34      0.26      0.29        47

    accuracy                           0.82       320
   macro avg       0.61      0.59      0.59       320
weighted avg       0.80      0.82      0.81       320

[[250  23]
 [ 35  12]]
```

[圖十]

Support Vector Classifier(SVC) 為 support vector machine 在分類上的應用。Support vector machine 支援向量在高維或無限維空間中構造超平面或超平面集合，廣泛地應用於分類、回歸、異常點檢測等問題中。透過參數的控制，可以得到非線性的決策邊界。下圖為 Support Vector Classifier (SVC) 的執行結果，Accuracy 為 88%。

```
svc = SVC()
svc.fit(x_train, y_train)
pred_svc = svc.predict(x_test)
print(classification_report(y_test, pred_svc))
print(confusion_matrix(y_test, pred_svc))
```

```
              precision    recall  f1-score   support

           0       0.88      0.98      0.93       273
           1       0.71      0.26      0.37        47

    accuracy                           0.88       320
   macro avg       0.80      0.62      0.65       320
weighted avg       0.86      0.88      0.85       320

[[268   5]
 [ 35  12]]
```

[圖十一]

現在，我們針對 Support Vector Classifier(SVC)的參數進行修正，使我們得到更高的 Accuracy。

在 param 裡放入我們要進行調整的參數。

```
param = {
    'C': [0.1,0.8,0.9,1,1.1,1.2,1.3,1.4],
    'kernel':['linear', 'rbf'],
    'gamma' :[0.1,0.8,0.9,1,1.1,1.2,1.3,1.4]
}
grid_svc = GridSearchCV(svc, param_grid=param, scoring='accuracy', cv=10)
```

```
grid_svc.fit(x_train, y_train)
grid_svc.best_params_
```

```
:
{'C': 1.2, 'gamma': 0.9, 'kernel': 'rbf'}
```

[圖十二]

最後我們得到在參數 C=1.2；gamma=0.9；kernel=rbf 時，我們會得到更高的精確度。將參數輸入後，執行結果如下圖所示，Support Vector Classifier(SVC)的 Accuracy 提高到了 90%。

```
              precision    recall  f1-score   support

           0       0.90      0.99      0.94       273
           1       0.89      0.34      0.49        47

    accuracy                           0.90       320
   macro avg       0.89      0.67      0.72       320
weighted avg       0.90      0.90      0.88       320

[[271   2]
 [ 31  16]]
```

[圖十三]

最後，我們對三個分類器進行交叉驗證，來決定選擇何種分類器。

使用交叉驗證是因為原先我們的資料集並沒有預先切割好「訓練資料(Training data)」和「測試資料(Testing data)」，為避免過度依賴某一特定的訓練和測試資料產生偏差。透過交叉驗證，每一部分的資料都會輪過一遍，成為訓練資料和測試資料，最後將所有的 Accuracy 取平均，我們就可以得到一個沒有偏差的 Accuracy。結果如下：

Random Forest Classifier 的 Accuracy 為 91.2%。

```
rfc_eval = cross_val_score(estimator = rfc, X = x_train, y = y_train, cv = 10)
rfc_eval.mean()
```

```
0.9124446358267717
```

[圖十四]

Stochastic Gradient Descent Classifier 的 Accuracy 為 85.8%。

```
sgd_eval = cross_val_score(estimator = sgd, X = x_train, y = y_train, cv = 10)
sgd_eval.mean()
```

```
0.8584584153543308
```

[圖十五]

Support Vector Classifier(SVC) 的 Accuracy 為 90.8%。

```
svc2_eval = cross_val_score(estimator = svc2, X = x_train, y = y_train, cv = 10)
svc2_eval.mean()
```

```
0.9085383858267717
```

[圖十六]

根據上述結果，我們最後會選用 Random Forest Classifier。

9.結論

在資料的分析與前處理上，我們透過比對各性質與質量和性質與性質間的關係，找出一些特徵，並對質量進行二進制分類。且很幸運的是我們所找到的資料集並沒有 missing value，因此在前處理上不用考慮如何去填補 missing value。接著，因為我們有使用 Stochastic Gradient Descent Classifier 分類器，因此會對資料進行歸一化的處理。最後在分類器的選擇上，透過交叉驗證使我們得到三個分類器無偏差的 Accuracy，而其中 Random Forest Classifier 的 Accuracy 值是最好的，因此我們最後會選用 Random Forest Classifier 進行分類。

10.參考資料

https://ithelp.ithome.com.tw/articles/10191069

https://ithelp.ithome.com.tw/articles/10197461

https://www.kaggle.com/vishalyo990/prediction-of-quality-of-wine

https://www.kaggle.com/rajmehra03/intro-to-parameter-tuning-in-scikit-acc-0-9175/comments