鳶尾花資料集辨識

一、讀取資料與資料視覺化

A. 匯入模塊函數

【01】

```python
from sklearn import datasets
from collections import Counter
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, accuracy_score
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
```

B. 讀取鳶尾花資料

【02】

```python
iris = datasets.load_iris()
print(iris.keys())
```

C. 建立變數儲存特徵、標籤、特徵名稱、標籤名稱

【03】

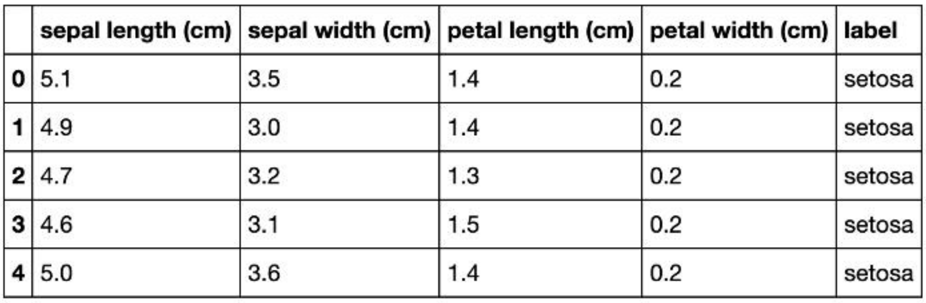
```python
features = iris.data
label = iris.target
features_name = iris.feature_names
label_name = iris.target_names

print("標籤類別數量：", Counter(label))
print("特徵名稱：", features_name)
print("標籤名稱：", label_name)
```

D. 建立 DataFrame，並新增標籤名稱的欄位：

【04】

```python
df = pd.DataFrame(features, columns=features_name)
df['label'] = [label_name[i] for i in label.astype(int)]
df.head()
```



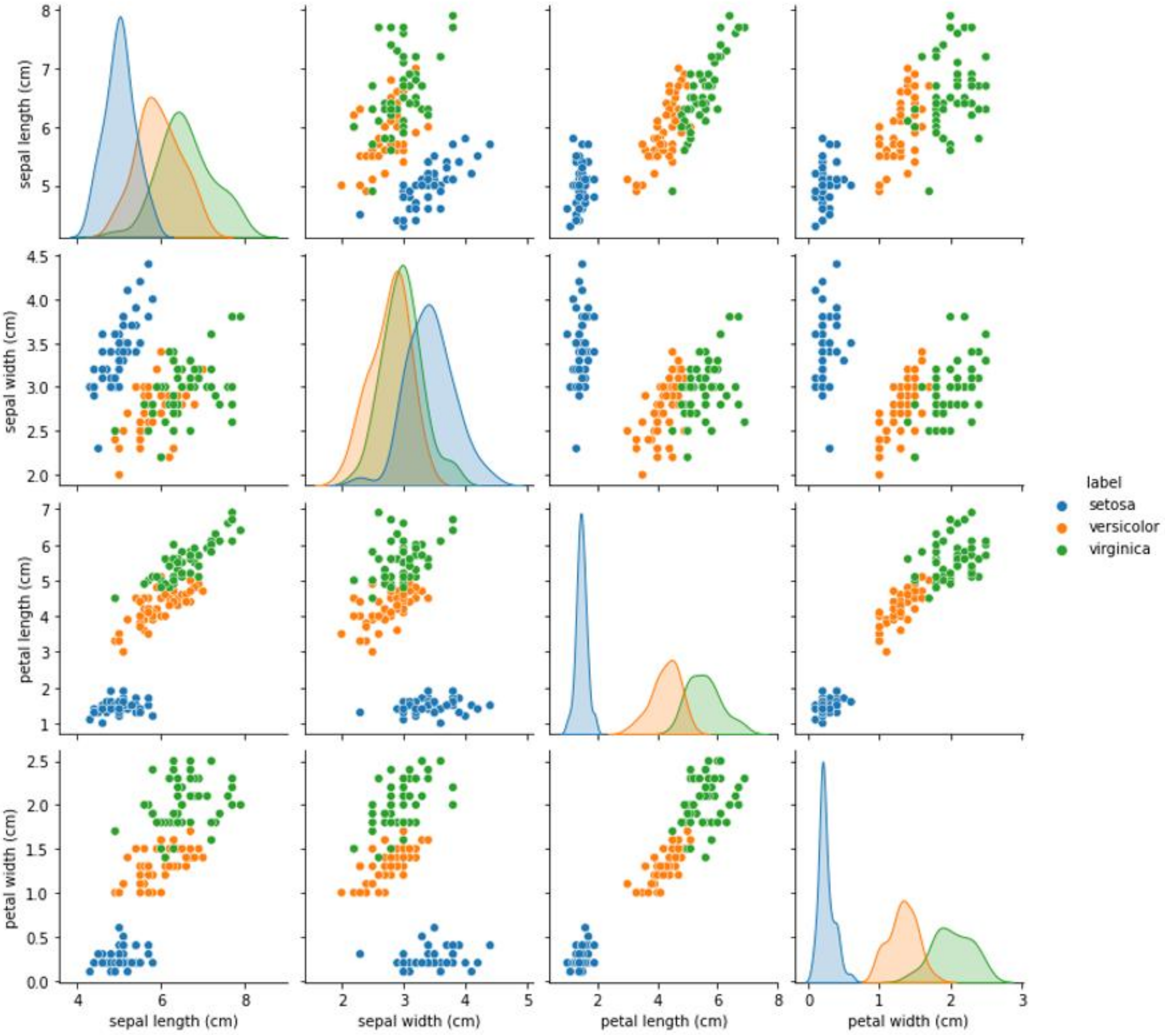
	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	label
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

圖一、建立 DataFrame 並新增標籤名稱

A. 繪製兩變數間的資料視覺化圖

【05】

```
sns.pairplot(df, hue='label')
```



圖二、兩變數間的資料視覺化圖

二、邏輯回歸模型訓練

A. 宣告模型（含超參數數值）與模型訓練

【06】

```
LR_Model = LogisticRegression(random_state=0, max_iter=1000)
LR_Model.fit(features, label)
```

B. 批量預測：回傳最大的預測值與機率值

【07】

```
predict = LR_Model.predict(features)
predict_proba = LR_Model.predict_proba(features)
```

三、驗證指標與模型的資料視覺化

A. 驗證指標：混淆矩陣與準確度

【08】

```
cm = confusion_matrix(label, predict)
acc = accuracy_score(label, predict)
```

```
print("混淆矩陣：\n", cm)
print("準確度：%0.3f" %(acc))
```

B. 模型的權重與偏差

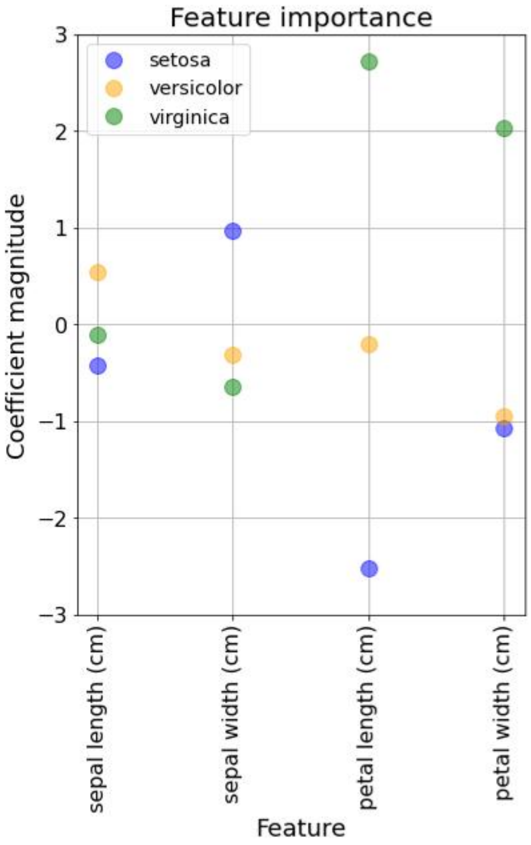
【09】

```
print("模型權重(係數):\n", LR_Model.coef_)
print("模型偏差(截距):", LR_Model.intercept_)
```

C. 繪製模型的權重來顯示特徵重要性圖

【11】

```
plt.figure(figsize=(6, 8))
plt.title('Feature importance',size = 20)
plt.plot(LR_Model.coef_[0].T,'o',markersize=12, color = 'blue', alpha = 0.5)
plt.plot(LR_Model.coef_[1].T,'o',markersize=12, color = 'orange', alpha = 0.5)
plt.plot(LR_Model.coef_[2].T,'o',markersize=12, color = 'green', alpha = 0.5)
plt.xticks(range(len(features_name)), features_name , rotation=90, size = 16)
plt.yticks(rotation=0, size = 16)
plt.legend(loc='upper left', labels=label_name, fontsize = 'x-large')
plt.ylim(-3, 3)
plt.grid()
plt.xlabel("Feature",size = 18)
plt.ylabel("Coefficient magnitude",size = 18)
plt.show()
```



圖三、特徵重要性

四、儲存與讀取模型及模型應用

A. 儲存與讀取模型：

【12】

```
#from sklearn.externals import joblib
import joblib #儲存Model
joblib.dump(LR_Model, 'Iris_LR.pkl')
LR_Model = joblib.load('Iris_LR.pkl')
```

B．應用預測：

【13】

```
test_features = [[9.0, 3.2, 1.1, 0.1]]
predict = LR_Model.predict(test_features)
print("預測為：", label_name[predict])
```