

智慧課服

Wine機器學習HomeWork

駱建宏

Wine機器學習作業流程

1. 匯入資料

匯入檔案: Wine.csv

2. 資料處理

- A. 取得X特徵資料
- B. 建立y資料轉換字典
- C. 取得y資料

3. 建立模型

- A. 建立訓練與測試資料
- B. 建立模型->決策樹
- C. 訓練模型(75%)
- D. 驗證模型(25%)
- E. 模型準確率=0.93

4. 挑選重要特徵

- A. 找出決策樹重要特徵
- B. 挑選重要特徵
- Flavanoids:43%
- Color_intensity:40%
- Proline:12%

5. 資料處理

- A. 取得X(重要特徵)資料
- Flavanoids:0.43
- Color_intensity:0.40
- Proline:0.12
- B. 建立y資料轉換字典
- C. 取得y資料

6. 重新建立模型並匯出

- A. 建立訓練與測試資料
- B. 建立模型->決策樹
- C. 訓練模型(75%)
- D. 驗證模型(25%)
- E. 模型準確率=0.93(與之前模型相同)
- F. 使用模型進行預測
- G. 匯出模型

7. 建立GUI

- A. Load模型
- B. 命令提示字元介面(.py)
- C. 視窗介面(.py)

命令提示字元

```
(base) C:\Windows\system32>python A_wine_pred_0721_2.py
請輸入預測資料:
flavanoids:3.5
color_intensity:8
proline:1480
等級:
flavanoids: 3.5
color_intensity: 8.0
proline: 1480.0
預測等級是: class_0
```

視窗

tk	
flavanoids	2.17
color_intensity	1.9
proline	407
=	
proline	class_1

```

In [1]: import pandas as pd
In [2]: import numpy as np
In [3]: #
In [4]: winedf=pd.read_csv('C:\智慧客服\教材分享_2021-20210715T010257Z-001\教材分享_2021\wine.csv')
In [5]: #檢視winedf dataframe

In [6]: winedf.head(5)
Out[6]:
   alcohol  malic_acid  ash  alcalinity_of_ash  ...  hue  od280/od315_of_diluted_wines  proline  class_label
0    14.23         1.71  2.43              15.6  ...  1.04              3.92      1065.0      class_0
1    13.20         1.78  2.14              11.2  ...  1.05              3.40      1050.0      class_0
2    13.16         2.36  2.67              18.6  ...  1.03              3.17      1185.0      class_0
3    14.37         1.95  2.50              16.8  ...  0.86              3.45      1480.0      class_0
4    13.24         2.59  2.87              21.0  ...  1.04              2.93       735.0      class_0

[5 rows x 14 columns]

In [7]: winedf.shape
Out[7]: (178, 14)

In [8]: winedf.columns
Out[8]:
Index(['alcohol', 'malic_acid', 'ash', 'alcalinity_of_ash', 'magnesium',
      'total_phenols', 'flavanoids', 'nonflavanoid_phenols',
      'proanthocyanins', 'color_intensity', 'hue',
      'od280/od315_of_diluted_wines', 'proline', 'class_label'],
      dtype='object')

In [9]: #載取特徵屬性X
In [10]: X=winedf.loc[:, 'alcohol':'proline']
In [11]: #檢視y的資料型態

In [15]: winedf.class_label.value_counts()
Out[15]:
class_1    71
class_0    59
class_2    48
Name: class_label, dtype: int64

In [16]: type(winedf.class_label.values[0])
Out[16]: str

```

1. 匯入資料

匯入檔案: Wine.csv

2. 資料處理

- A. 取得X特徵資料
- B. 建立y資料轉換字典
- C. 取得y資料

```
In [17]: #建立轉換字典
```

```
In [18]: class_mapping={"class_0":0,"class_1":1,"class_2":2}
```

```
In [19]: winedf['class']=winedf['class_label'].map(class_mapping)
```

```
In [20]: #建立y
```

```
In [21]: y=winedf['class']
```

```
In [22]: #檢視X資料
```

```
In [23]: X.values[5]
```

```
Out[23]:  
array([1.42e+01, 1.76e+00, 2.45e+00, 1.52e+01, 1.12e+02, 3.27e+00,  
       3.39e+00, 3.40e-01, 1.97e+00, 6.75e+00, 1.05e+00, 2.85e+00,  
       1.45e+03])
```

```
In [24]: X.values[:5]
```

```
Out[24]:  
array([[1.423e+01, 1.710e+00, 2.430e+00, 1.560e+01, 1.270e+02, 2.800e+00,  
       3.060e+00, 2.800e-01, 2.290e+00, 5.640e+00, 1.040e+00, 3.920e+00,  
       1.065e+03],  
      [1.320e+01, 1.780e+00, 2.140e+00, 1.120e+01, 1.000e+02, 2.650e+00,  
       2.760e+00, 2.600e-01, 1.280e+00, 4.380e+00, 1.050e+00, 3.400e+00,  
       1.050e+03],  
      [1.316e+01, 2.360e+00, 2.670e+00, 1.860e+01, 1.010e+02, 2.800e+00,  
       3.240e+00, 3.000e-01, 2.810e+00, 5.680e+00, 1.030e+00, 3.170e+00,  
       1.185e+03],  
      [1.437e+01, 1.950e+00, 2.500e+00, 1.680e+01, 1.130e+02, 3.850e+00,  
       3.490e+00, 2.400e-01, 2.180e+00, 7.800e+00, 8.600e-01, 3.450e+00,  
       1.480e+03],  
      [1.324e+01, 2.590e+00, 2.870e+00, 2.100e+01, 1.180e+02, 2.800e+00,  
       2.690e+00, 3.900e-01, 1.820e+00, 4.320e+00, 1.040e+00, 2.930e+00,  
       7.350e+02]])
```

```
In [25]: type(X.values)
```

```
Out[25]: numpy.ndarray
```

```
In [26]: X.values.shape
```

```
Out[26]: (178, 13)
```

```
In [27]: X.values.ndim
```

```
Out[27]: 2
```

2. 資料處理

- A. 取得X特徵資料
- B. 建立y資料轉換字典
- C. 取得y資料

```
In [28]: #檢視y資料
```

```
In [29]: y.values[:5]
```

```
Out[29]: array([0, 0, 0, 0, 0], dtype=int64)
```

```
In [30]: y.values.shape
```

```
Out[30]: (178,)
```

```
In [31]: y.values.ndim
```

```
Out[31]: 1
```

```
In [32]: #建立訓練與測試資料,比例0.75:0.25
```

```
In [33]: from sklearn.model_selection import train_test_split
```

```
In [34]: X_train,X_test,y_train,y_test=train_test_split(  
...: X.values,y.values,random_state=0,test_size=0.25)
```

```
In [35]: #建立模型-->決策樹
```

```
In [36]: from sklearn.tree import DecisionTreeClassifier
```

```
In [37]: tree01=DecisionTreeClassifier(random_state=0)
```

```
In [38]: #訓練模型
```

```
In [39]: treeModel01=tree01.fit(X_train,y_train)
```

```
In [40]: #使用測試資料,驗證決策樹
```

```
In [41]: treeModel01.score(X_test,y_test)
```

```
Out[41]: 0.9333333333333333
```

```
In [42]: #找出決策樹重要特徵
```

```
In [43]: treeModel01.feature_importances_
```

```
Out[43]:  
array([[0.          , 0.01888132, 0.02216502, 0.          ,  
        0.          , 0.43241919, 0.          , 0.          ,  
        0.          , 0.          , 0.12337846])
```

```
In [44]: winedf.columns
```

```
Out[44]:  
Index(['alcohol', 'malic_acid', 'ash', 'alcalinity_of_ash', 'magnesium',  
       'total_phenols', 'flavanoids', 'nonflavanoid_phenols',  
       'proanthocyanins', 'color_intensity', 'hue',  
       'od280/od315_of_diluted_wines', 'proline', 'class_label', 'class'],  
      dtype='object')
```

```
In [45]: #挑選三個重要特徵:'flavanoids':0.43241919,'color_intensity':0.403156,'proline':0.12337846
```

3. 建立模型

- A. 建立訓練與測試資料
- B. 建立模型->決策樹
- C. 訓練模型(75%)
- D. 驗證模型(25%)
- E. 模型準確率=0.93

4. 挑選重要特徵

- A. 找出決策樹重要特徵
 - B. 挑選重要特徵
- Flavanoids:43%
- Color_intensity:40%
- Proline:12%

```
In [50]: newX=winedf.loc[:,['flavanoids','color_intensity','proline']]
```

```
In [51]: #建立y
```

```
In [52]: y=winedf['class']
```

```
In [53]: #檢視newX
```

```
In [54]: newX[:5]
```

```
Out[54]:
```

	flavanoids	color_intensity	proline
0	3.06	5.64	1065.0
1	2.76	4.38	1050.0
2	3.24	5.68	1185.0
3	3.49	7.80	1480.0
4	2.69	4.32	735.0

```
In [55]: newX.values.shape
```

```
Out[55]: (178, 3)
```

```
In [56]: newX.values.ndim
```

```
Out[56]: 2
```

```
In [57]: #檢視y
```

```
In [58]: y.values[:5]
```

```
Out[58]: array([0, 0, 0, 0, 0], dtype=int64)
```

```
In [59]: y.values.shape
```

```
Out[59]: (178,)
```

```
In [60]: y.values.ndim
```

```
Out[60]: 1
```

```
In [65]: tree02=DecisionTreeClassifier(random_state=0)
```

```
In [66]: #訓練模型
```

```
In [69]: #訓練模型
```

```
In [70]: treeModel02=tree02.fit(X_train,y_train)
```

```
In [71]: #使用測試資料,驗證決策樹
```

```
In [72]: treeModel02.score(X_test,y_test)
```

```
Out[72]: 0.9333333333333333
```

5. 資料處理

A. 取得X(重要特徵)資料

Flavanoids:0.43

Color_intensity:0.40

Proline:0.12

B. 建立y資料轉換字典

C. 取得y資料

6. 重新建立模型並匯出

A. 建立訓練與測試資料

B. 建立模型->決策樹

C. 訓練模型(75%)

D. 驗證模型(25%)

E. 模型準確率=0.93(與之前模型相同)

F. 使用模型進行預測

G. 匯出模型


```
In [78]: #將模型dump
In [79]: import joblib
In [80]: #sample測試模型
In [81]: sample_01=[3,5,1520]
In [82]: new_sample01=np.array(sample_01).reshape(1,-1)
In [83]: #以treeModel02預測
In [84]: predict_01=treeModel02.predict(new_sample01)
In [85]: predict_01[0]
Out[85]: 0
In [86]: #將模型dump
In [87]: joblib.dump(treeModel02,'wine_tree02_0721.pkl')
Out[87]: ['wine_tree02_0721.pkl']
```

6. 重新建立模型並匯出

- A. 建立訓練與測試資料
- B. 建立模型->決策樹
- C. 訓練模型(75%)
- D. 驗證模型(25%)
- E. 模型準確率=0.93(與之前模型相同)
- F. 使用模型進行預測
- G. 匯出模型

```

1  # -*- coding: utf-8 -*-
2  """
3  A wine_pred_0721.py
4  Wine預測系統
5  """
6  import numpy as np
7  #import joblib module
8  import joblib
9  loadModel0721=joblib.load('wine_tree02_0721.pkl')
10
11  #=====
12  #使用者輸入介面：
13  print("請輸入預測資料:")
14  flavanoids=float(input("flavanoids:"))
15  color_intensity=float(input("color_intensity:"))
16  proline=float(input("proline:"))
17  input_list=[flavanoids,color_intensity,proline]
18
19  #將使用者輸入轉成(1,3)的numpy.ndarray
20  input_array=np.array(input_list).reshape(1,-1)
21  #以loadModel0721對輸入樣本等級預測
22  pred=loadModel0721.predict(input_array)
23
24  #建立一個輸出字典 "dict_wine"
25  dict_wine={0: 'class_0', 1: 'class_1', 2: 'class_2'}
26
27  #列印預測結果
28  #印出預測等級
29  print("=====")
30  print("等級:")
31  print("flavanoids:",flavanoids," \n",
32        "color_intensity:",color_intensity," \n",
33        "proline:",proline)
34  print("預測等級是:",dict_wine[pred[0]])
35

```

7. 建立GUI

- A. Load模型
- B. 命令提示字元介面(.py)
- C. 視窗介面(.py)

命令提示字元

```

(base) C:\Windows\system32>python A_wine_pred_0721_2.py
請輸入預測資料:
flavanoids:3.5
color_intensity:8
proline:1480
=====
等級:
flavanoids: 3.5
color_intensity: 8.0
proline: 1480.0
預測等級是: class_0

```



```
ch12_010a_event.py x fs717_ch12_011_Entry.py x fs717_ch12_011a_Entry.py x homeWork0722.py x
1  # -*- coding: utf-8 -*-
2  """
3  Created on Thu Jul 22 23:00:22 2021
4
5  @author: User
6  """
7
8  #匯入套件
9  import pandas as pd
10 import numpy as np
11 import joblib
12 import tkinter as tk
13
14 #載入模型
15 loadModel0721=joblib.load('wine_tree02_0721.pkl')
16
17 #建立視窗
18 window01=tk.Tk()
19
20 #設定寫入資料與計算結果
21 num01=tk.DoubleVar()
22 num02=tk.DoubleVar()
23 num03=tk.DoubleVar()
24 result01=tk.StringVar()
25
26 #計算結果函數
27 def class_label():
28     flavanoids=num01.get()
29     color_intensity=num02.get()
30     proline=num03.get()
31     input_list=[flavanoids,color_intensity,proline]
32     input_array=np.array(input_list).reshape(1,-1)
33     pred=loadModel0721.predict(input_array)
34     dict_wine={0: 'class_0', 1: 'class_1', 2: 'class_2'}
35     result01.set(dict_wine[pred[0]])
```

```
36
37 #建立視窗內容
38 tk.Label(window01,width=30,text="flavanoids").grid(row=0,column=0)
39 tk.Entry(window01,width=10,textvariable=num01).grid(row=0,column=1)
40
41 tk.Label(window01,width=30,text="color_intensity").grid(row=1,column=0)
42 tk.Entry(window01,width=10,textvariable=num02).grid(row=1,column=1)
43
44 tk.Label(window01,width=30,text="proline").grid(row=2,column=0)
45 tk.Entry(window01,width=10,textvariable=num03).grid(row=2,column=1)
46
47 tk.Button(window01,width=5,text="=",command=class_label).grid(row=3,column=0)
48
49 tk.Label(window01,width=30,text="proline").grid(row=4,column=0)
50 tk.Entry(window01,width=10,textvariable=result01).grid(row=4,column=1)
51
52 #執行視窗
53 window01.mainloop()
```

7. 建立GUI

- A. Load模型
- B. 命令提示字元介面(.py)
- C. 視窗介面(.py)

