

根據你的學號最後一個尾數除以 6 的餘數，解開所附的 hw03\_data.zip 檔案，選擇對應的資料夾下的資料做以下的事情：

A. 描述資料集的欄位數，資料筆數，以及是否有 missing data。

A: 資料集的欄位數:8，資料筆數:900，missing data 的格數:0，有 missing data 的資料筆數:0。

```
A:
資料集的欄位數:8
資料筆數:900
missing data的格數:0
有missing data的資料筆數:0
```

```
# A---描述資料集的欄位數，資料筆數，以及是否有missing data---
count_t=0  #missing data的格數
f=0
count_p=0  #有missing data的資料筆數
for i in range(len(data)):
    for j in data[i]:
        if j=='?':
            count_t=count_t+1
            f=1
    if f==1:
        count_p=count_p+1
    else:
        #處理missing data by Complete cases analysis
        cleared_data.append(data[i])
    f=0
print(f"A:\n資料集的欄位數:{len(data[0])}\n"
      +f"資料筆數:{len(data)}\n"
      +f"missing data的格數:{count_t}\n"
      +f"有missing data的資料筆數:{count_p}")
#-----
```

B. 分析資料集中最一個欄位(除資料集 4 為 RiskLevel 外,其他為 class)的分佈, 包括

- 列出不同數值的數量與所佔比例
- 根據 a 的分佈計算此欄位的 entropy

A: class 欄位中各數值出現次數: {'Besni': 450, 'Kecimen': 450} ,

class 欄位中各數值出現比例: {'Besni': 0.5, 'Kecimen': 0.5} 。

class 欄位的 entropy:1.0 。

```
B:
class欄位中各數值出現次數: {'Besni': 450, 'Kecimen': 450}
class欄位中各數值出現比例: {'Besni': 0.5, 'Kecimen': 0.5}
class欄位的entropy:1.0
```

```
#-----
# B-- 列出不同數值的數量與所佔比例
class_column=[]
for i in range(len(data)):
    class_column.append(data[i][len(data[i])-1])
value, counts = np.unique(class_column, return_counts=True)
print("B:\n\nclass欄位中各數值出現次數:", dict(zip(value, counts)))
print("class欄位中各數值出現比例:", dict(zip(value, counts/len(data))))
#根據a的分佈計算此欄位的entropy
probs=counts/len(class_column) # numpy陣列可以直接除以一個數值
entropy=sum(-p*log(p,2) if p>0 else 0 for p in probs)
print(f"class欄位的entropy:{entropy}")
#-----
```

C. 從 KNN, Decision Tree, 或是 Naive Bayes 中三選一，建立模型並回報模型的準確度。注意

- 以 0.8/0.2 的比例將資料集分成訓練與測試用資料
- 明確敘述你在建立模型過程中手動設定的所有參數(例如 train\_test\_split 中的 random\_state, KNN 的 K 值等)

A: 建立模型過程中手動設定的所有參數有: 在函式 train\_test\_split 中 test\_size=0.2, random\_state=50。DecisionTreeClassifier(criterion="entropy")是指使用 entropy 作為節點分裂的指標。

Accuracy=0.8

C:  
Accuracy=0.8

```
# C--建立Decision Tree模型並回報模型的準確度

#####將資料切成訓練/測試的地方(以下)#####
#將data、target分開
data_information=[]
data_class=[]
for i in range(len(cleared_data)):
    data_information.append(cleared_data[i][0:-1])
    data_class.append(cleared_data[i][len(cleared_data[i])-1])
#切分訓練、測試資料
x_train,x_test,y_train,y_test=train_test_split(data_information,data_class,test_size=0.2,random_state=50)
#####將資料切成訓練/測試的地方(以上)#####
#####產生模型與訓練模型的的地方(以下)#####
train_accuracy=[]
test_accuracy=[]
dt=DecisionTreeClassifier(criterion="entropy")
dt.fit(x_train,y_train)
#####計算(呼叫套件提供方法)準確度#####
print(f"C:\nAccuracy={dt.score(x_test,y_test)}")

#畫數
tree.plot_tree(dt)
plt.show()
#####產生模型與訓練模型的的地方(以上)#####
```

繳交內容：

1. 你所用的資料檔案
2. 可以重現你的結果的程式碼，並在以下幾個部份以加上註解的方式標注出來(在 python 請在註解前加上井字號#為開頭)
  - 將資料切成訓練/測試的地方
  - 產生模型與訓練模型的的地方
  - 計算(手動或呼叫套件提供方法)準確度的地方
3. 一個 pdf 檔，內容是上述 A,B,以及 C 的部份

將所有內容以 zip 格式壓縮成一個檔案之後上傳。

其他事項：

- 使用 Decision Tree 的同學如果將你所建立出來的樹轉成 png 圖檔輸出並附上來，會有額外加分。

