國立成功大學

資料探勘 Data Mining

Project02

課程教授：高宏宇

學生：葉芯妤

學號: P96074147

目錄

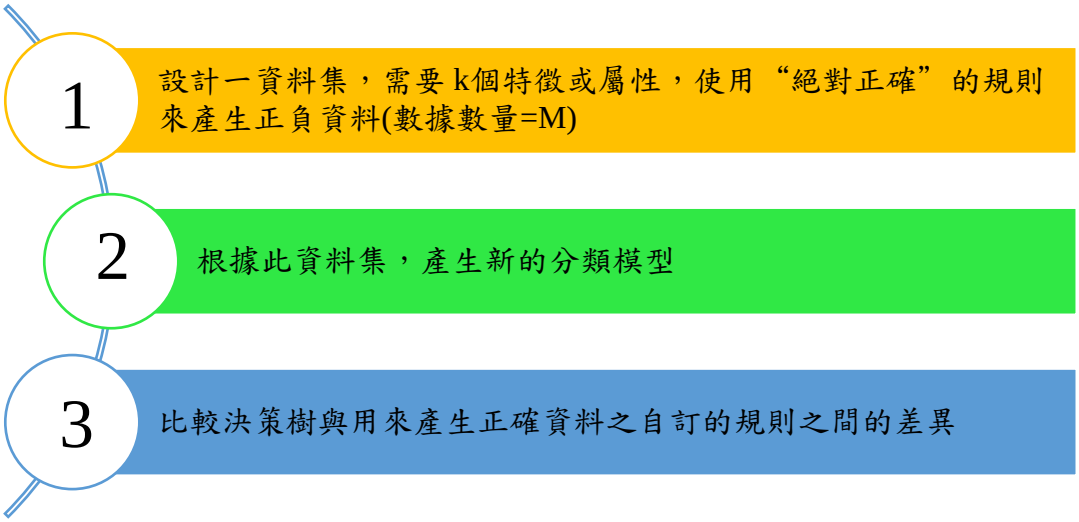
一、目標說明 3

二、資料說明 3

三、實作說明 5

四、分析比較 7

一、目標說明



二、資料說明

以預測研究生的壓力為主軸，建立資料，內容包含是否能準時畢業、論文進度、開會頻率、年級以及修課數量，下表為欄位說明。共有 5 個分類特徵，黃色部分為分類的項目，即為研究生有無壓力。

欄位	說明	內容
on_time	是否能準時畢業	是=1，否=0
schedule	論文進度	未完成一半=0~49，完成一半=50~100
meeting	開會頻率	低=1，中=2，高=3
grade	年級	一年級=1，二年級=2，三年級=3，四年級=4
courses	修課數量	小於等於四門課=0~4，大於四門課=5~8
pressure	研究生壓力	有=1，無=0

利用 excel 建立資料，並用亂數產生 100 筆資料，命名為 data_train.csv，下圖為資料內容：

	A	B	C	D	E	F	G
1	id	on_time	schedule	meeting	grade	courses	pressure
2	1	0	69	2	1	3	0
3	2	0	83	3	3	0	0
4	3	1	52	3	1	2	0
5	4	0	40	2	4	7	1
6	5	1	30	2	3	5	0
7	6	1	2	1	3	0	0
8	7	0	93	1	2	5	1
9	8	1	98	1	4	2	0
10	9	1	26	2	2	6	0
11	10	0	48	3	3	7	0
12	11	0	10	2	2	4	0
13	12	1	66	1	3	8	0
14	13	0	66	3	1	7	0
15	14	0	34	2	4	4	1
16	15	0	55	3	1	6	0
17	16	1	95	2	2	8	0

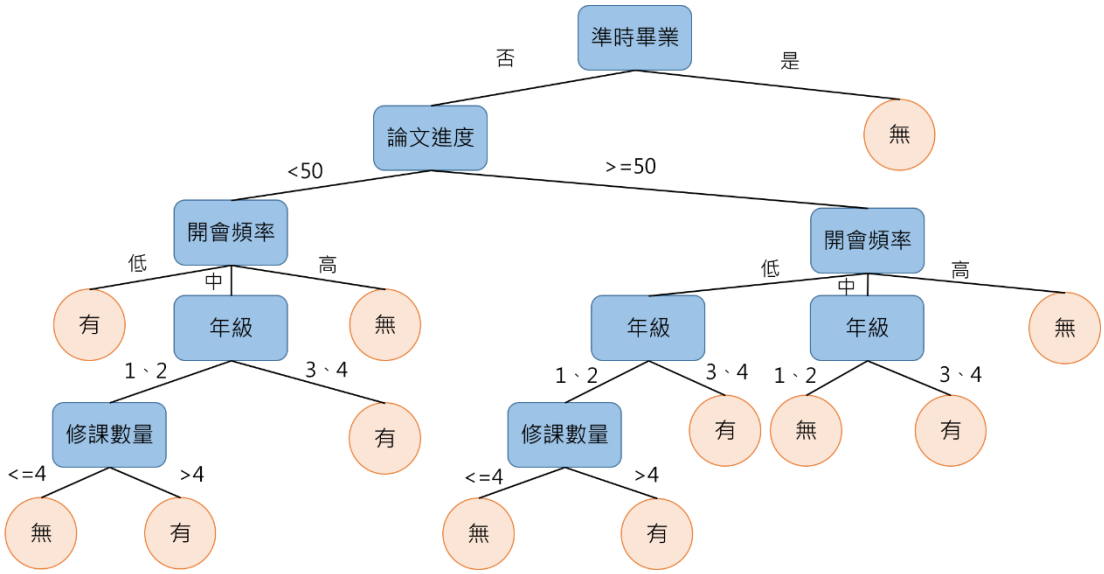
另外建立一測試的資料，同樣利用 excel 建立，並用亂數產生 100 筆資料，命名為 data_test.csv，下圖為資料內容：

	A	B	C	D	E	F
1	id	on_time	schedule	meeting	grade	courses
2	1	1	60	1	1	1
3	2	1	36	1	1	8
4	3	0	64	2	2	3
5	4	1	79	3	2	2
6	5	1	52	3	2	1
7	6	1	38	1	1	0
8	7	0	54	1	3	5
9	8	1	11	3	4	0
10	9	1	20	2	1	0
11	10	0	41	2	3	1
12	11	0	0	3	2	3
13	12	1	12	1	2	5
14	13	0	70	2	3	8
15	14	1	1	3	4	2
16	15	1	91	3	2	0
17	16	1	93	3	4	2

■ 我的預測：

從資料中觀察到，是否能準時畢業為研究生最大的壓力來源，因此，我以是否準時畢業作為考量，認為此關係影響較大，畫出決策樹。下圖為預測的決策樹，規則為：

1. 研究生是否能準時畢業
2. 論文進度是否完成一半
3. 開會頻率多寡
4. 研究生的年級
5. 修課數量是否大於四門課



三、實作說明

利用 Python 撰寫程式碼，步驟為：先將檔案做前處理後，再訓練決策樹，將決策樹視覺化輸出，最後輸出預測模型，下圖為實作步驟及程式碼。

1. 檔案做前處理

```python
import numpy as np
import pandas as pd
import os
import sys
from sklearn import tree
from sklearn import preprocessing
import pydotplus
import collections
from time import gmtime, strftime

def conda_fix(graph):
        path = os.path.join(sys.base_exec_prefix, "Library", "bin", "graphviz")
        paths = ("dot", "twopi", "neato", "circo", "fdp")
        paths = {p: os.path.join(path, "{}.exe".format(p)) for p in paths}
        graph.set_graphviz_executables(paths)

os.chdir('C:\\Users\\P96074147\\Desktop\\P96074147_Project2')

dataFeature = ["on_time", "schedule", "meeting", "grade", "courses"]

trainData = pd.read_csv("data_train.csv")
testData = pd.read_csv("data_test.csv")
```

2. 訓練決策樹

```python
trainer = pd.DataFrame([trainData["on_time"], trainData["schedule"], trainData["meeting"],
                        trainData["grade"], trainData["courses"]]).T
tree_model = tree.DecisionTreeClassifier()
tree_model.fit(X = trainer,y = trainData["pressure"])

tree_model.score(X = trainer, y = trainData["pressure"])
```

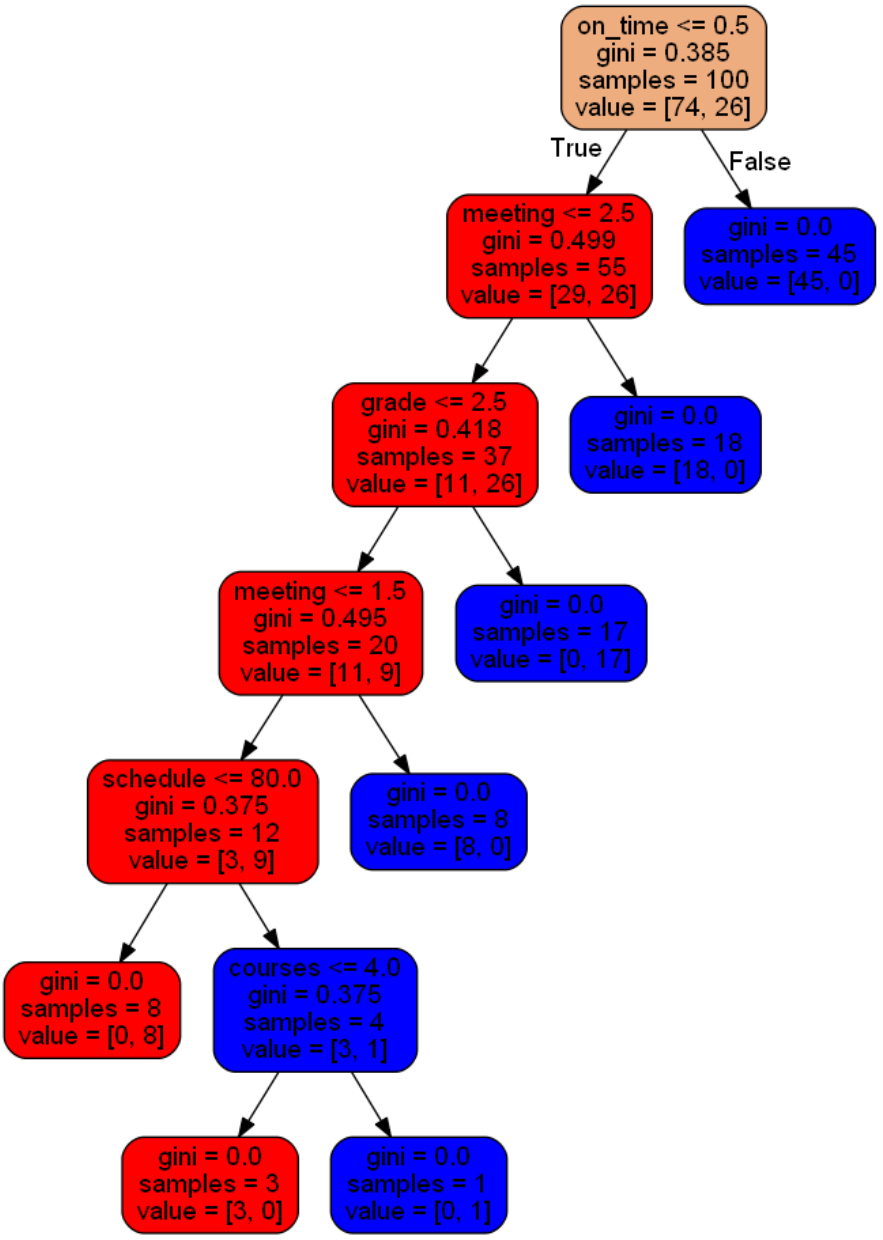
3. 決策樹視覺化輸出

```python
dot_data = tree.export_graphviz(tree_model, feature_names=dataFeature, out_file=None, filled=True, round
graph = pydotplus.graph_from_dot_data(dot_data)

colors = ('red', 'blue')
edges = collections.defaultdict(list)

for edge in graph.get_edge_list():
    edges[edge.get_source()].append(int(edge.get_destination()))

for edge in edges:
    edges[edge].sort()
    for i in range(2):
        dest = graph.get_node(str(edges[edge][i]))[0]
        dest.set_fillcolor(colors[i])

conda_fix(graph)
OUT_PNG_NAME = str(strftime("%Y%m%d%H%M%S", gmtime()))+".png"
graph.write_png(OUT_PNG_NAME)
```

4. 輸出預測模型

```python
test_features = pd.DataFrame([testData["on_time"], testData["schedule"], testData["meeting"],
                              testData["grade"], testData["courses"]]).T
test_preds = tree_model.predict(X=test_features)

reportData = pd.DataFrame(test_preds)
reportData.to_csv("result.csv", index=False)
```

最後輸出的結果命名為 result.csv，將針對產生結果與原先訓練的資料進行比對，而產生的決策樹如下圖所示：



四、分析比較

將 data_test.csv 的訓練結果與原本 data_train.csv 的結果進行比對，另外建立一 excel 檔，命名為 comparison.csv，從全部 100 筆資料中取四分之一出來進行分析，也就是 25 筆資料。

比對結果如下圖所示，欄位 test 為 data_test.csv 的訓練結果，而欄位 train 為原本 data_train.csv 產生的結果，若訓練結果與實際結果是一樣的，即為 0 和 0 或 1 和 1，那麼比較結果，也就是欄位 comparison，則為 1；反之則為 0，代表結果為 0 和 1 或 1 和 0。

	A	B	C
1	test	train	comparison
2	0	0	1
3	0	0	1
4	0	0	1
5	0	1	0
6	0	0	1
7	0	0	1
8	1	1	1
9	0	0	1
10	0	0	1
11	1	0	0
12	0	0	1
13	0	0	1
14	1	0	0
15	0	1	0
16	0	0	1
17	0	0	1

觀察結果發現25筆資料中有7筆比較結果是不同的,也就是欄位comparison為0的，所以推論如果有100筆資料的話，比較結果即為28/100，代表會有28筆比較結果不同，因此得出結論為：兩個資料的相似率大約為72%。但由於決策樹特徵數較少，只有5個，而且仔細觀察的話，當論文進度小於50時，且開會頻率為中的情況下，與論文進度大於等於50 時，且開會頻率為低的情況下，最終結果會是一樣的，因此推測如果特徵數一增加，那麼準確率就會下降。