

## 淺在顧客平板購買預測

### 1. 簡介

透過決策樹從訓練資料學習，預測測試集中的資料中，學生是否會買平板，並且將預測會購買平板的學生聯絡資料載入資料庫

### 2. 原始數據

- 欄位

- 學生名

- 排名: 1-20 名

- 科系:CS(資訊工程系)，EE(electric) (電機工程系)，EE(electronic)(電子工程系)，ME(機械工程系)

- 生活費(income\_per\_month):範圍為 30000 元/月至 60000 元/月

- 是否購買平板：y(有購買平板)，n(未購買平板)

- 原始資料 train\_data.xlsx 由 chatGPT 指定範圍後生成，訓練集與測試集數量比例為 7:3

訓練集(前 14 筆)				
name	rate	subject	income_per_month	buy_tablet_or_not
Olivia	12.00	CS	45890.00	y
Ethan	5.00	CS	51235.00	y
Ava	17.00	EE(Electronic)	36741.00	n
Mason	3.00	ME	40219.00	n
Isabella	8.00	ME	54380.00	y
Liam	6.00	CS	39874.00	n
Sophia	14.00	EE(Electrical )	57302.00	y
Jackson	10.00	ME	32987.00	n
Emma	20.00	EE(Electronic)	47250.00	n
Aiden	2.00	EE(Electrical )	54106.00	y
Mia	1.00	CS	30562.00	n
Lucas	18.00	ME	57643.00	y
Harper	7.00	CS	48795.00	n
Noah	11.00	ME	35321.00	n

測試集(後 6 筆)				
name	rate	subject	income_per_month	buy_tablet_or_not
Abigail	13.00	EE(Electrical )	41908.00	n
Ella	15.00	EE(Electronic)	53674.00	y
Alexander	16.00	ME	38899.00	y
Grace	19.00	EE(Electrical )	45213.00	y
Elijah	4.00	EE(Electronic)	51247.00	y
Scarlett	9.00	CS	34125.00	y

### 3. 資料前處理

數值型資料:用線性壓縮壓縮

```
from sklearn.preprocessing import MinMaxScaler
from sklearn.pipeline import make_pipeline
num_pl=make_pipeline(MinMaxScaler())
numcol_input=num_pl.fit_transform(df_num)
numcol_input
```

✓ 0.0s

處理結果:

	rate	income_per_month
0	0.579	0.566
1	0.211	0.763
2	0.842	0.228
3	0.105	0.357
4	0.368	0.880
5	0.263	0.344
6	0.684	0.987
7	0.474	0.090
8	1.000	0.616
9	0.053	0.869
10	0.000	0.000
11	0.895	1.000
12	0.316	0.673
13	0.526	0.176
14	0.632	0.419
15	0.737	0.853
16	0.789	0.308
17	0.947	0.541
18	0.158	0.764
19	0.421	0.132

## 類別型資料:用 onehot 做讀熱編碼

```
from sklearn.preprocessing import OneHotEncoder

onehotencoder = OneHotEncoder()
cat_pl=make_pipeline(OneHotEncoder(sparse=False))
catcol_input=cat_pl.fit_transform(df_cat)
catcol_input
```

✓ 0.0s

## 處理結果:

	subject_CS	subject_EE(Electrical )	subject_EE(Electronic)	subject_ME
0	1.0	0.0	0.0	0.0
1	1.0	0.0	0.0	0.0
2	0.0	0.0	1.0	0.0
3	0.0	0.0	0.0	1.0
4	0.0	0.0	0.0	1.0
5	1.0	0.0	0.0	0.0
6	0.0	1.0	0.0	0.0
7	0.0	0.0	0.0	1.0
8	0.0	0.0	1.0	0.0
9	0.0	1.0	0.0	0.0
10	1.0	0.0	0.0	0.0
11	0.0	0.0	0.0	1.0
12	1.0	0.0	0.0	0.0
13	0.0	0.0	0.0	1.0
14	0.0	1.0	0.0	0.0
15	0.0	0.0	1.0	0.0
16	0.0	0.0	0.0	1.0
17	0.0	1.0	0.0	0.0
18	0.0	0.0	1.0	0.0
19	1.0	0.0	0.0	0.0

#### 4. 演算法及參數使用

##### I. 演算法:決策樹

##### II. 亂度值標準:entropy

##### III. random\_state:0

#### 5. 預測結果

測試結果					
name	rate	subject	income_per_month	predict	buy_tablet_or_not
Abigail	13.00	EE(Electrical )	41908.00	n	n
Ella	15.00	EE(Electronic)	53674.00	y	y
Alexander	16.00	ME	38899.00	n	y
Grace	19.00	EE(Electrical )	45213.00	y	y
Elijah	4.00	EE(Electronic)	51247.00	y	y
Scarlett	9.00	CS	34125.00	n	y

準確率:0.667

混淆矩陣:

	預測會買	預測不買
實際會買	1	0
實際不買	2	3

因為此處之數據為隨機生成之資料，因此模型

預測準確率只有 0.667

## 6. 載入預測會買的客人至資料庫

- I. 原始顧客資料(資料庫 db\_for\_tablet 的資料表 student，亦透過 chatGPT 生成)，顧客資料中只有測試資料中的學生

	0	Filter	Filter	Filter	Filter
	1	C0015	Abigail	0912345678	A_UNIVERSITY
	2	C0016	Ella	0923456789	B_UNIVERSITY
	3	C0017	Alexander	0934567890	C_UNIVERSITY
	4	C0018	Grace	0945678901	B_UNIVERSITY
	5	C0019	Elijah	0956789012	A_UNIVERSITY
	6	C0020	Scarlett	0967890123	C_UNIVERSITY

- II. 找出測試資料中那些學生可能購買

```
wili_buy=[]
for i in range(len(output)):
    if output[i]=='y':
        wili_buy.append(df['name'][i+14])
wili_buy
[195] ✓ 0.0s
... ['Ella', 'Grace', 'Elijah']
```

- III. 設定資料庫連線

```
import pymysql

db_settings = {
    "host" : "127.0.0.1",
    "port" : 3306,
    "user" : "root" ,
    "password" : "0000" ,
    "db" : "db_for_tablet" ,# 根據你的資料庫名稱
    "charset" : "utf8"
}
```

#### IV. 將資料庫內容放入 tuple 以供使用

```
try:
    # 建立Connection物件
    conn = pymysql.connect(**db_settings)
    print("連線成功")

    with conn.cursor() as cursor:
        for i in range(len(data_insert)):
            cursor.execute("insert into `customer_list` (\
                `customer_ID`,\
                `customer_name`,\
                `phone`,\
                `school` ) values (%s,%s,%s,%s)",\
                (data_insert[i]))

        conn.commit()
        #with statement 完成後，會自動關閉資料庫的連線操作，詳請需要 with 指令
    print("資料新增成功")
except Exception as ex: # 出現意外時印出
    print("出現例外情況")
    print(ex)
```

#### V. 篩出可能購買的學生資料

```
data_insert=[]

for i in range(len(wili_buy)):
    for j in range(len(results)):
        if wili_buy[i]==results[j][1]:
            data_insert.append(results[j])
```

#### VI. 連線資料庫 customer\_list 將可能購買平板的 學生載入資料庫中的資料表

## VII. 至資料庫確認是否順利載入資料

	0	Filter	Filter	Filter	Filter
<input type="checkbox"/>	1	C0016	Ella	0923456789	B_UNIVERSITY
<input type="checkbox"/>	2	C0018	Grace	0945678901	B_UNIVERSITY
<input type="checkbox"/>	3	C0019	Elijah	0956789012	A_UNIVERSITY

## 7. 參考文章

- I. sklearn.tree.DecisionTreeClassifier 官方文件:<https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>
- II. 一行指令學 Python -用機器學習掌握人工智慧，徐聖訓
- III. 使用 Pymysql 套件來在 Python 中執行資料庫:  
[https://theriseofdavid.github.io/2020/11/30/Python/Python\\_Mysql\\_pymysql/#google\\_vignette](https://theriseofdavid.github.io/2020/11/30/Python/Python_Mysql_pymysql/#google_vignette)