金融信用卡
智慧型風險評估系統

指導老師 黃登揚老師、蔡智勇老師

學員 朱漢城 | 楊順翔 | 林宏彥 | 林芷羽 | 黃文怡
葉枝倫 | 王曉雯 | 周盈均 | 魏君豪

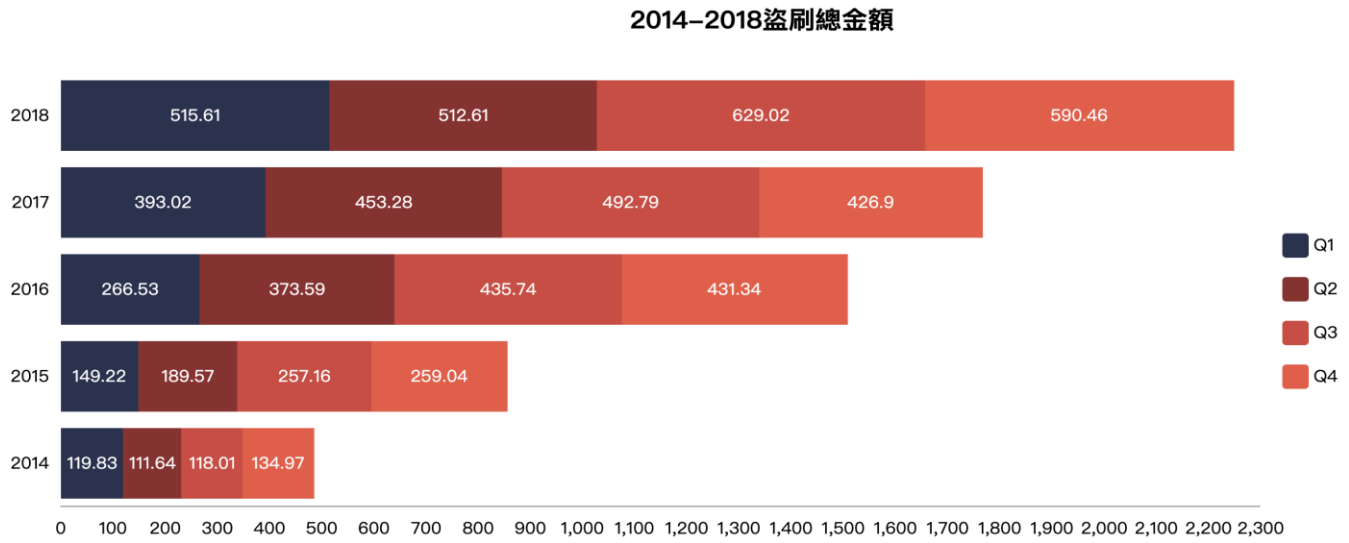
目錄

前言 /INTRODUCTION

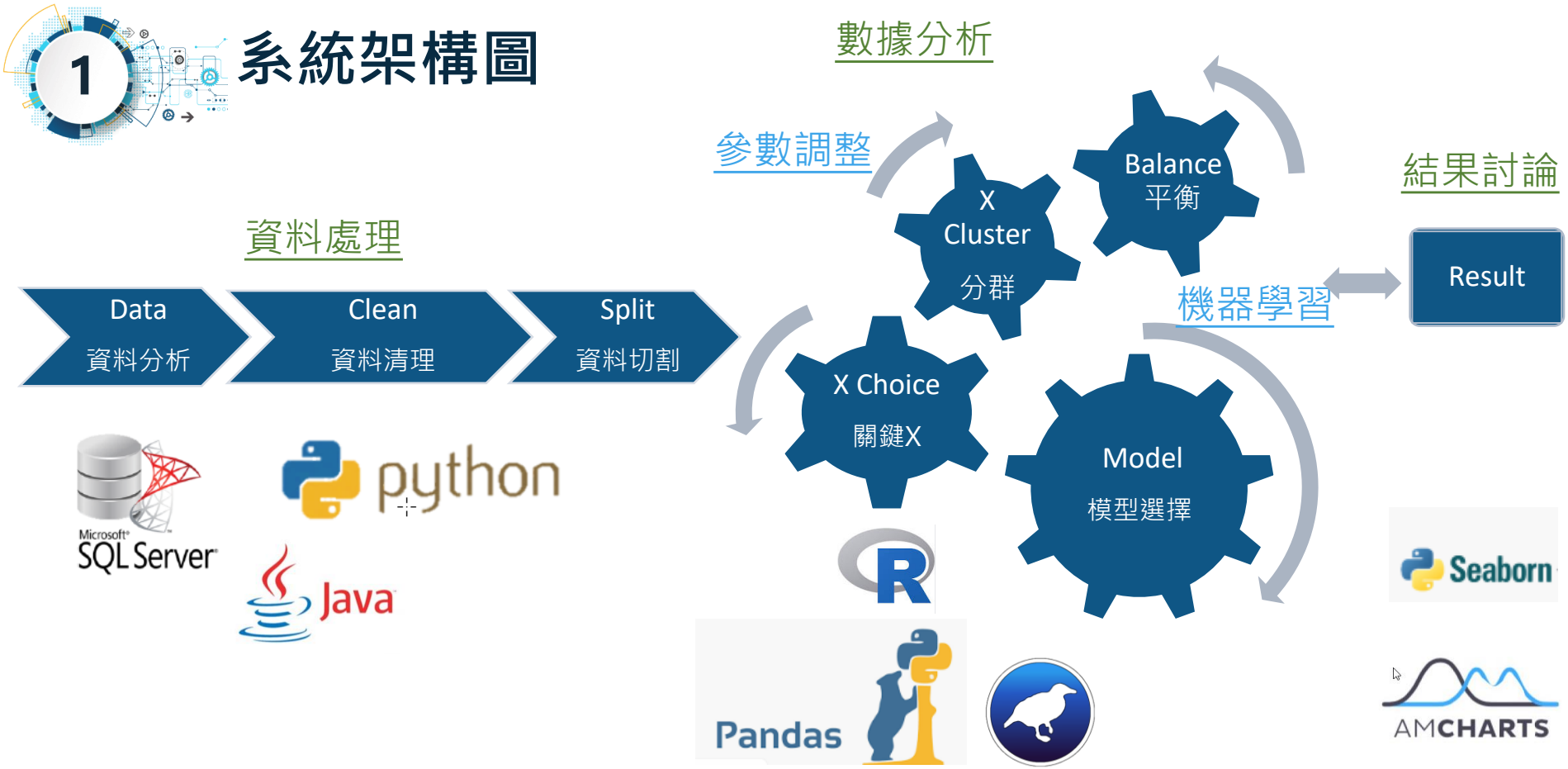
彙報人：黃文怡

信用卡在金流與消費資訊在銀行產業中扮演舉足輕重的腳色，使用率與金額年年提高，在提高信用卡使用率與使用金額的同時，如何降低盜刷與違約欠款等風險狀況顯得格為重要。

本專題以玉山銀行公開競賽的信用卡資料為基礎，輔以網路上搜尋到的相關信用卡公開資料，使用Python & Pandas、Weka、Database & SQL 等工具進行機器學習的模型建立及調校測試，模擬分析出一套針對信用卡盜刷預估系統。



1 系統架構圖



資料分析
- 資料來源
- 欄位說明
- 未處理資料預跑

資料清理
- 欄位刪除
- 資料清除

資料切割
- Train &Test

X Choice
- 欄位刪除
- 資料清除

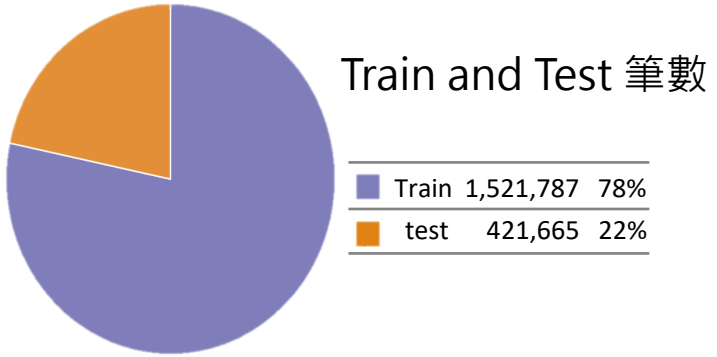
X Cluster
- K Mean

Model
- Tree (J48)
- Tree (RandomForest)
- MulitilayerPerceptron

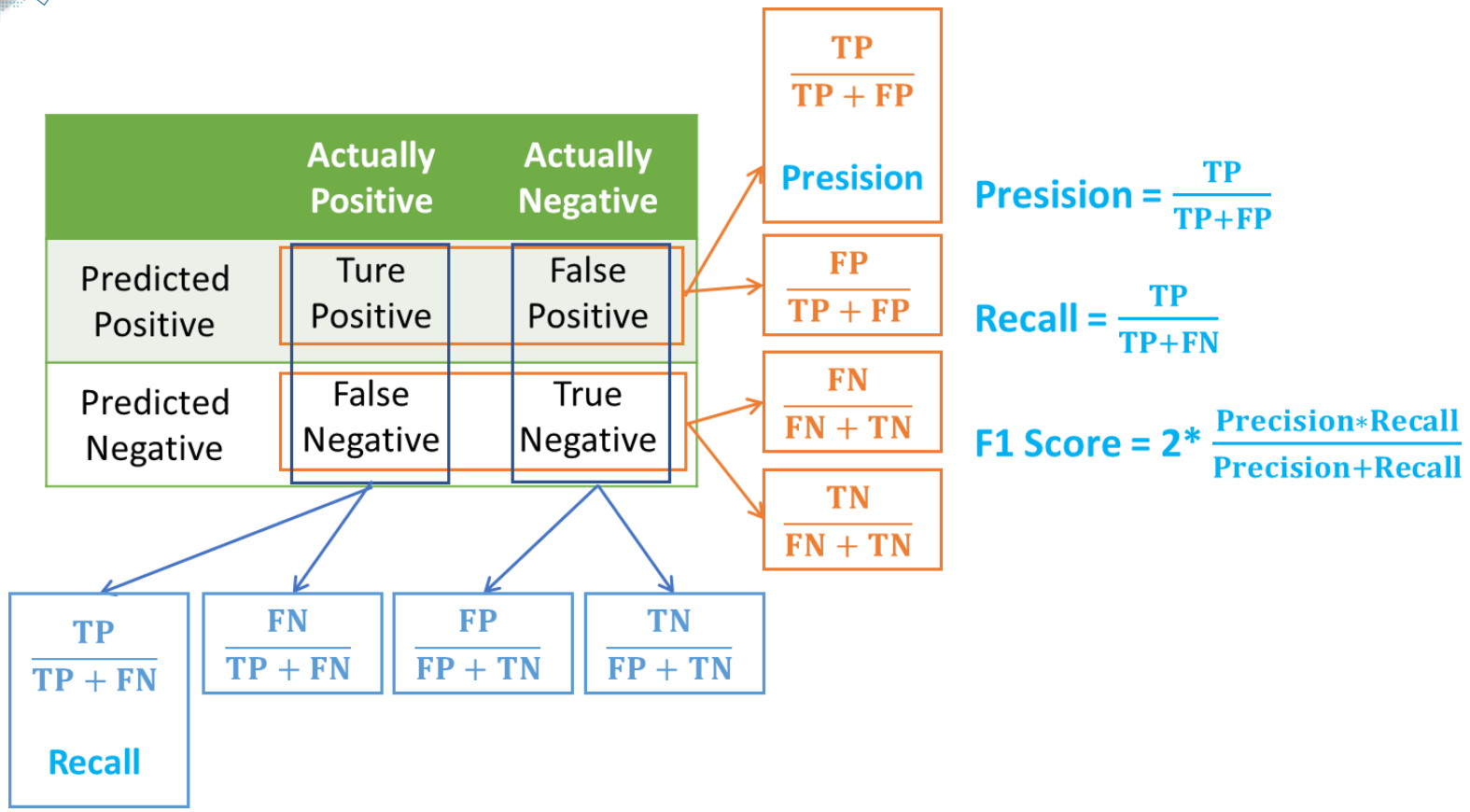
1 資料來源

- 玉山提供150萬筆Train data及46萬筆Test data (無是否盜刷的欄位)
- X : 23 項； Y=1 (盜刷) & Y=0(非盜刷)
- 玉山比賽規則:
  - 評分方式 : F1 Score
  - F1 Score公式 :

$$\text{F1 Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}$$



1 混淆矩陣


Actually Positive
Actually Negative
Predicted Positive
Ture Positive
False Positive
Predicted Negative
False Negative
True Negative
TP / TP + FP
Presision
FP / TP + FP
FN / FN + TN
TN / FN + TN
TP / TP + FN
Recall
FN / TP + FN
FP / FP + TN
TN / FP + TN
Presision = TP / TP+FP
Recall = TP / TP+FN
F1 Score = 2* Precision∗Recall / Precision+Recall

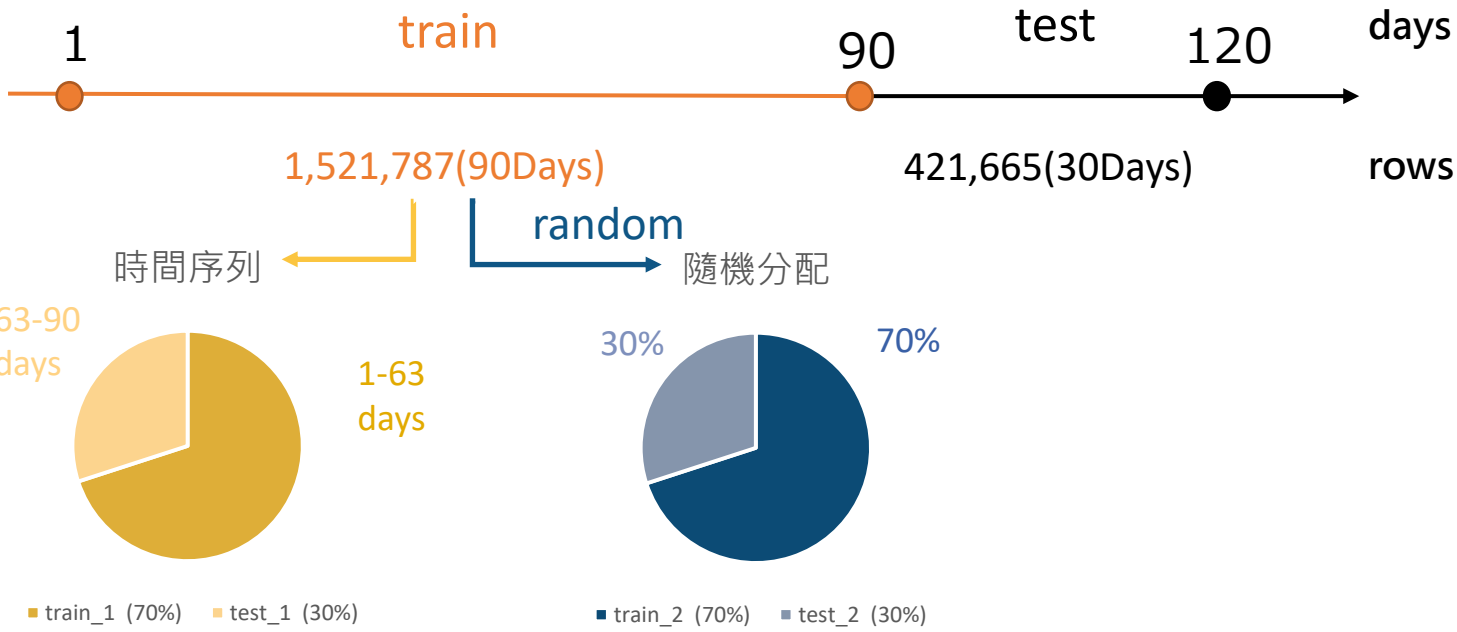
2 資料分析_欄位說明 (原始資料皆經過代碼轉換)

資料欄位、(說明)	資料內容
bacno(歸戶帳號)	跳號
txkey(交易序號)	0到七位數，跳號
locdt(授權日期)	1到90，90天的類別
loctm(授權時間)	六位數(XXXXXX;時分秒)
cano(交易卡號)	跳號
contp(交易類別)	0到6
etymd(交易型態)	0到10
mchno(特店代號)	0到六位數，跳號
acqic(收單行代碼)	0開始依序排序
mcc(MCC_code)	三位數
conam(交易金額-台幣，經過轉換)	含有小數點以下(0.08)的各種位數

資料欄位、(說明)	資料內容
ecfg(網路交易註記)	N/Y兩種類別
insfg(分期交易註記)	N/Y兩種類別
iterm(分期期數)	0到8
stocn(消費地國別)	0到三位數
scity(消費城市)	0到四位數
stscd(狀態碼)	0到4
ovrlt(超額註記碼)	N/Y兩種類別
flbmk(Fallback註記)	N.Y.空值
hcefg(支付型態)	0到9
csmcu(消費地幣別)	0到兩位數
flg_3dsmk(3DS註記)	N.Y.空值
fraud_ind(盜刷註記)	N/Y兩種類別

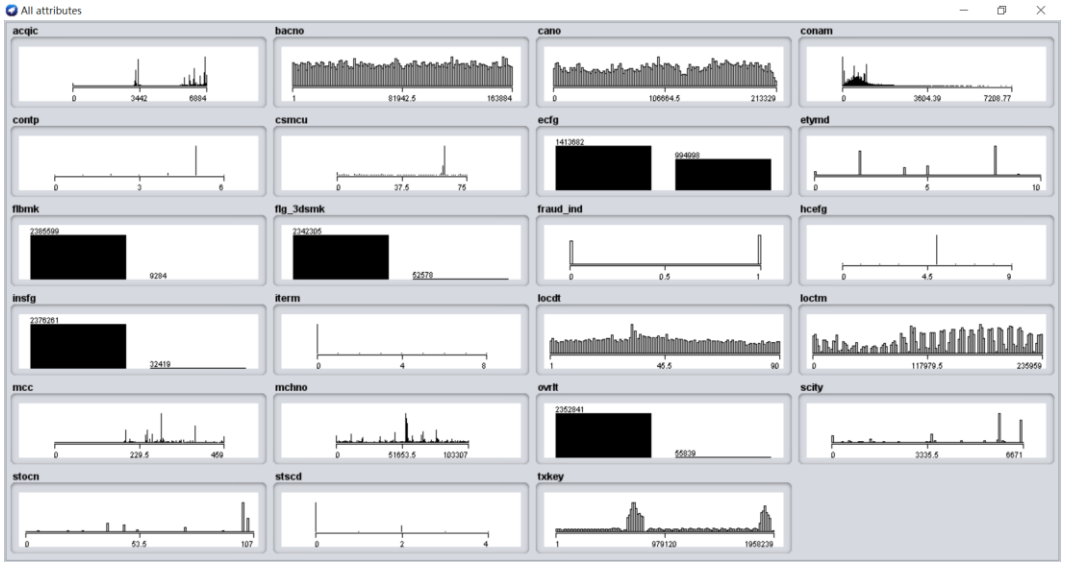
2 資料切割_Train & Test

- 玉山提供data為1~90日為train data, 91~120日為test data
- 將train data資料分割做模型訓練 (train_1: test_1= 7:3)
- 分割方式: 1.依時間切割2.隨機分割→測試結果無明顯差異



2 資料分析_未處理資料預跑

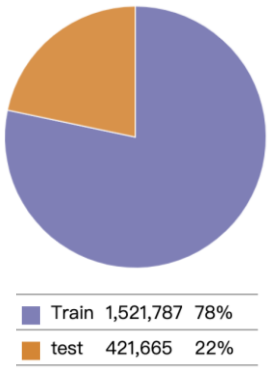
- 使用原X值未經刪除以及類別重新分群的結果

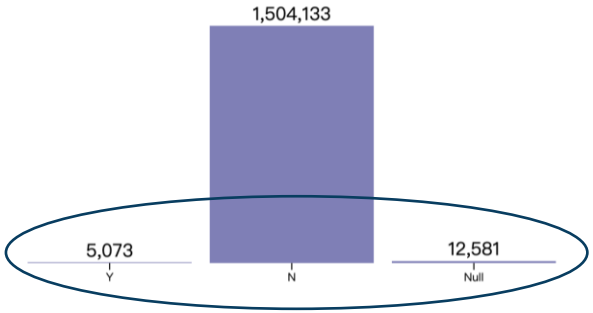


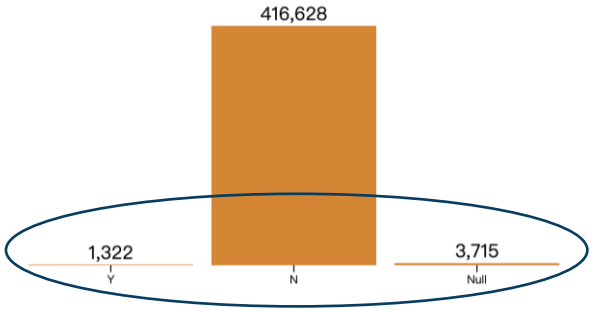
Model	Type	Accuracy	Precision	Recall	F1 Score
J48	Tree	97.2495	0.882	0.411	0.560

2 資料清理

- 空值處理
  - flbmk(Fallback註記)、flg_3dsmk(3DS交易註記)，有大量空格的資料內容
  - 空值比例比盜刷高且Test 數據空值高→不可忽略
- 異常值處理
  - Insfg(分期)、iterm(分期數)，欄位資料內容不符





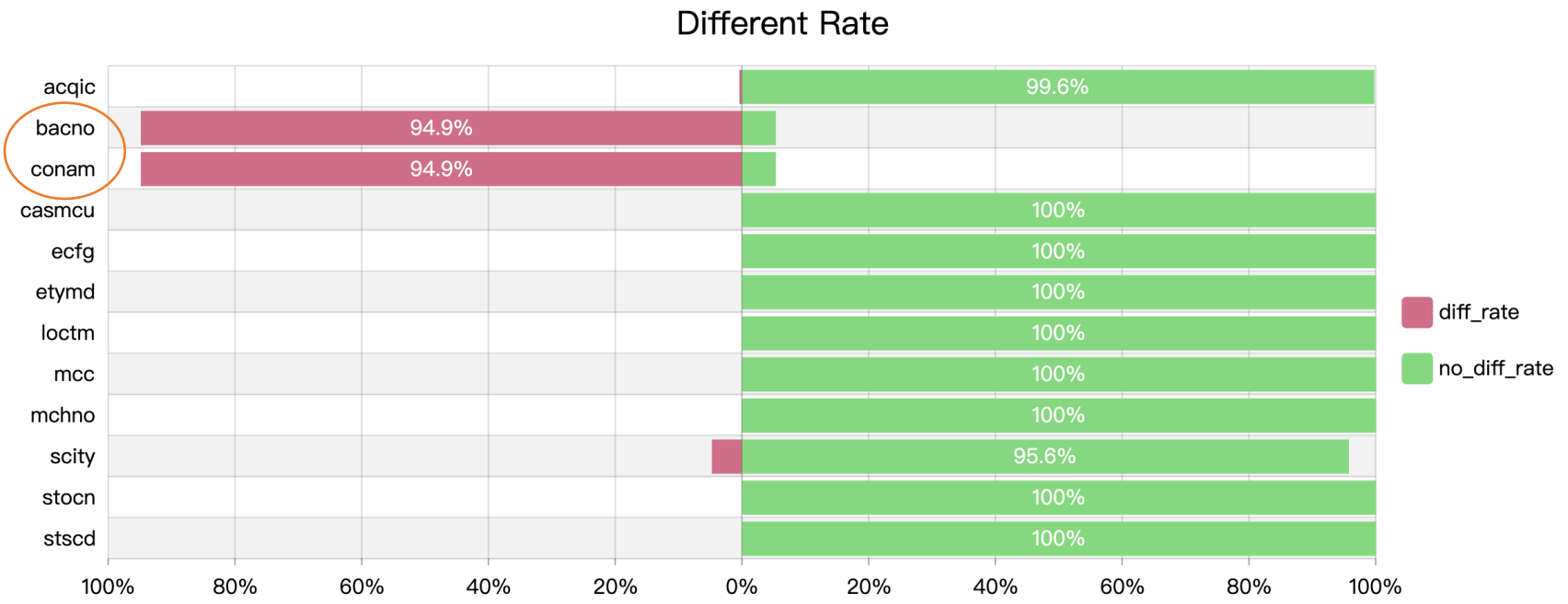


Train	筆數	比例
有盜刷	5073	0.34
沒盜刷	1504133	100
空值	12581	0.84

Test	筆數	比例
有盜刷	1322	0.32
沒盜刷	416628	100
空值	3715	0.89

2 測試集問題

- 由訓練資料與測試資料比對, 發現 “bacno帳戶資料” 及 “cano交易卡號” 兩欄位有94.85% 在test data未出現, 避免overfitting移除2欄位

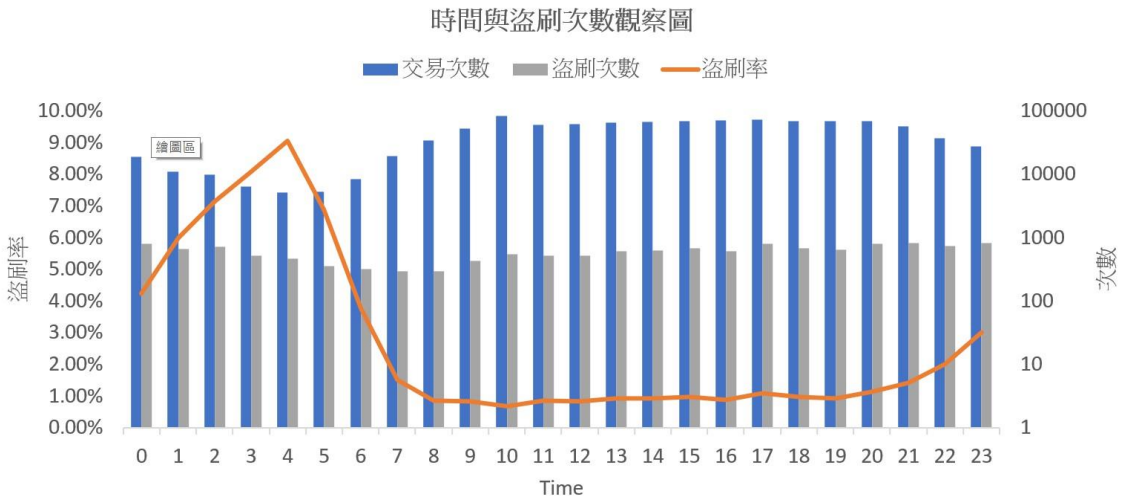


3 資料分析_欄位觀察

時間	沒盜刷	盜刷	刷卡數	盜刷率 (盜刷/刷卡數)
0	17931	795	18726	4.25%
1	10359	660	11019	5.99%
2	9266	713	9979	7.15%
3	5904	519	6423	8.08%
4	4646	463	5109	9.06%
5	4879	361	5240	6.89%
6	8134	318	8452	3.76%
7	18996	293	19289	1.52%
8	33514	291	33805	0.86%
9	51795	435	52230	0.83%
10	81761	556	82317	0.68%
11	59218	515	59733	0.86%
12	61817	517	62334	0.83%
13	64556	610	65166	0.94%
14	66431	631	67062	0.94%
15	68498	677	69175	0.98%
16	69427	617	70044	0.88%
17	71686	790	72476	1.09%
18	68876	672	69548	0.97%
19	68557	640	69197	0.92%
20	68559	797	69356	1.15%
21	56769	815	57584	1.42%
22	36420	747	37167	2.01%
23	26328	816	27144	3.01%

不同時段盜刷頻率

- 樞紐分析 & 類別資料轉換成盜刷率
  - 以loctm為例, 將時間字串列轉換, 抽取小時單位使用樞紐分析
  - 以盜刷筆數除以總刷卡數得到盜刷率
  - 觀察到凌晨時間盜刷頻率最高，盜刷次數與時間無明顯關係
- 直方圖
  - Ex. 以交易筆數來觀察觀察不同時間交易量



資料分析_轉換欄位

Index	ID	time	city	fraud_ind
1	Jade	12	Taoyuan	1
2	Eason	9	Pingtung	0
3	Jade	9	Hsinchu	1
4	Alicia	9	Taipei	1
5	Alicia	22	Hsinchu	0
6	Eason	17	Taipei	0

Index	ID	time	city	fraud_ind
1	100%	100%	100%	1
2	0%	67%	0%	0
3	100%	67%	50%	1
4	50%	67%	50%	1
5	50%	0%	50%	0
6	0%	0%	50%	0

$$\frac{\text{9點盜刷總筆數}}{\text{9點刷卡總筆數}} = \frac{\text{2筆}}{\text{3筆}} = 67\%$$

3 X choice

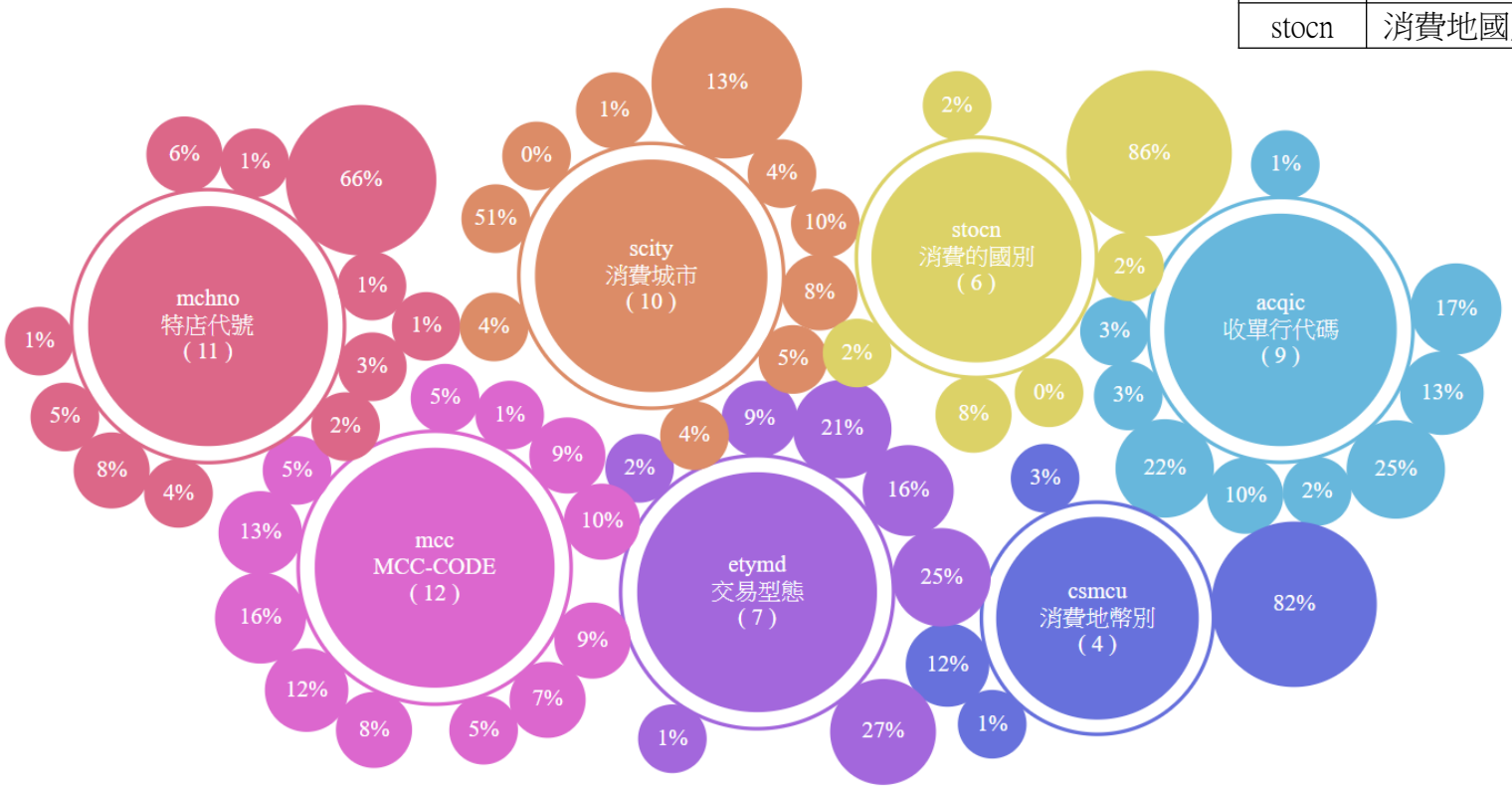
- 運用Logistic regression邏輯回歸
  挑出7組X與Y屬性相關的變數
- R 語言逐步回歸分析檢此7組X均為顯著

Items		Value
acqic	收單行代碼	0.0006706
csmcu	消費地幣別	-0.0009856
etymd	交易型態	-0.0000341
mcc	MCC_CODE	-0.0008198
mchno	特店代號	0.0055315
scity	消費城市	0.002523
stocn	消費地國別	-0.0005085

3 K means

- 運用K means分群方法
  合計7個欄位分別個別處理

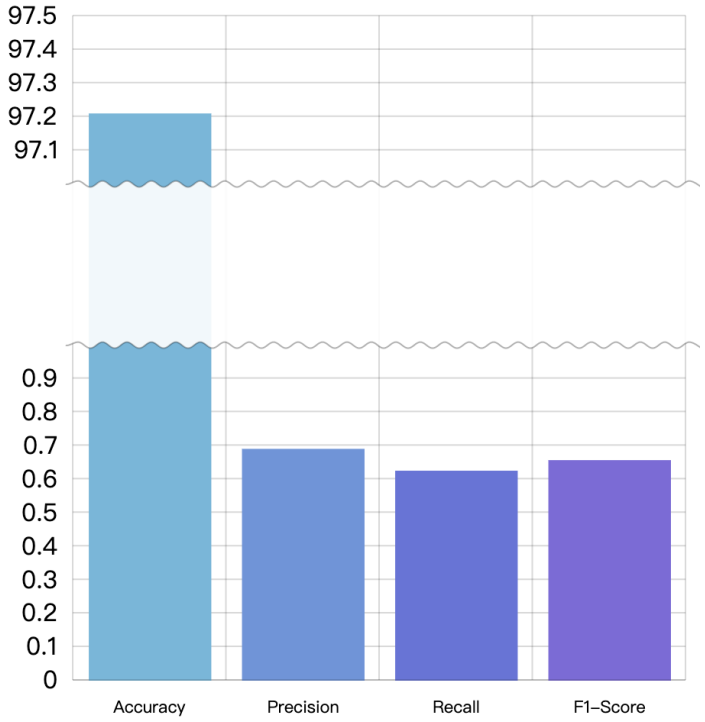
Items		group
acqic	收單行代碼	9
csmcu	消費地幣別	4
etymd	交易型態	7
mcc	MCC_CODE	12
mchno	特店代號	11
scity	消費城市	10
stocn	消費地國別	6



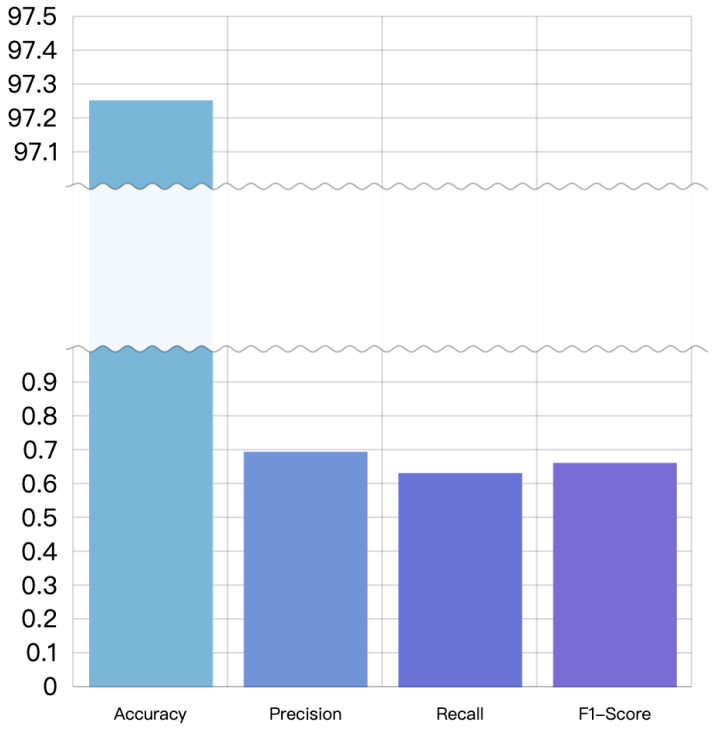
3 比較

Cluster_Type	Module	Type	Columns	Accuracy	Precision	Recall	F1 Score
Self_defintion	J48	Tree	7	97.183	0.687	0.626	0.655
Self_defintion	RandomForest	Tree	7	97.2103	0.691	0.626	0.657
K-mean	J48	Tree	7	96.8068	0.625	0.63	0.627
K-mean	RandomForest	Tree	7	97.2541	0.696	0.633	0.663

同時也比較J48 & RandomForest model 差異



RandomForest



0.657 Self_defintion

K means **0.663**

■ 較高

反覆條件測試

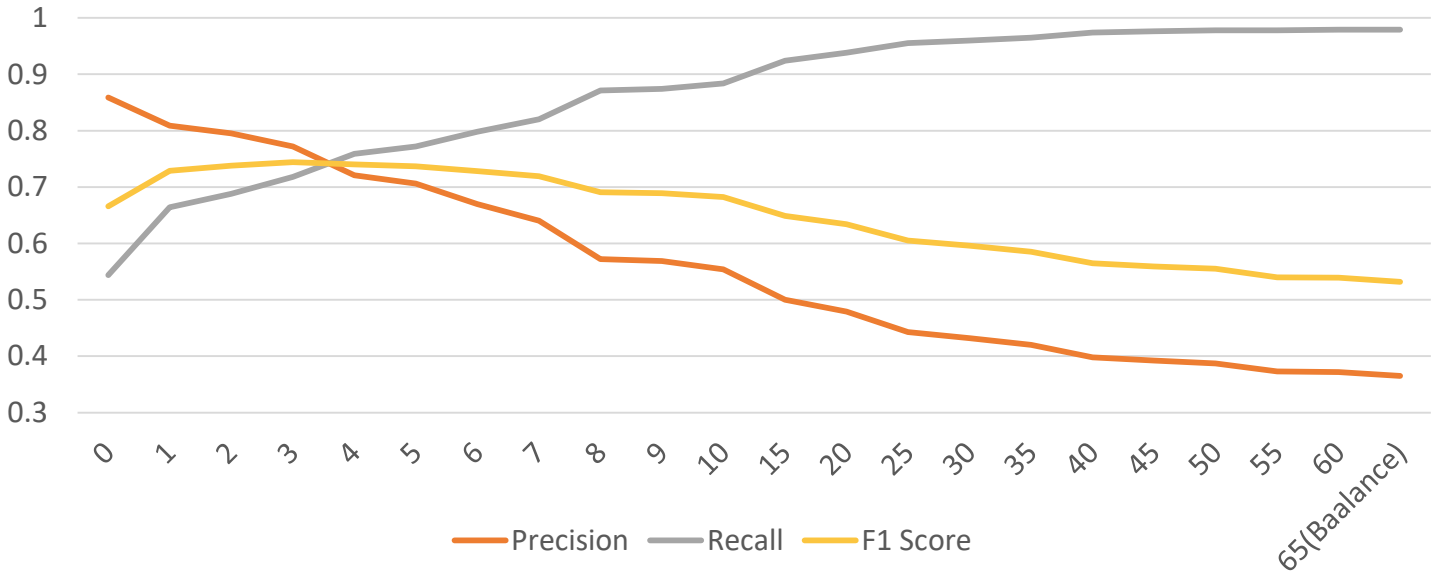
- Tree model: RandomForest
  - Add conam : F1_Score gain 0.069
  - Add loctm : F1_Score gain 0.027
  - Conam+loctm : gain 0.124 (0.757)

- Function model : MulitilayerPercept
  - 類別轉數據data選用不同model測試

Item	Cluster_Type	Model	Type	Columns	Accuracy	Precision	Recall	F1 Score
Original	N/A	J48	Tree	All	97.2495	0.882	0.411	**0.56**
Logistic regression	K-mean	RandomForest	Tree	7	97.2541	0.696	0.633	0.663
add conam	K-mean	RandomForest	Tree	8	97.7559	0.747	0.717	0.732
add loctm	K-mean	RandomForest	Tree	8	97.5575	0.753	0.637	0.69
conam+loctm	K-mean	RandomForest	Tree	9	97.9157	0.753	0.761	**0.757**
function model test	N/A	MulitilayerPerceptron	Function	N/A	98.7955	0.548	0.563	0.555

4 混淆矩陣調和

- F1_Score調和問題:
  - Y1 & Y0 Balance 到1:1，Recall & Precision過高與過低 且 F1_Score偏低
  - F1_Score公式得知最佳化為Precision與Recall相近
  - 調整Y1資料平衡的複製倍數→Copy 4倍 盜刷筆數會出現最高點
  - 且依選取不同X變數, 最高點位置不同



4 總結

- 本組之風險評估系統最終使用Random Forest演算法，K means分群，Precision：0.753，Recall：0.761。
- 網路與非網路的盜刷比例高達17.1倍，若有網路相關的欄位增加，應能增進分析結果。
- 參賽的時程僅兩個禮拜，後續的分析不能帶進比賽系統評估分數，較為可惜。

THANK YOU