金融科技導論期末專案報告
題目：信用卡盜刷偵測
Credit Card Fraud Detection

指導老師：張智星 教授

學生：林晉宇 (R09521606)

劉紹凱 (B06902134)

侯　喆 (B08703093)

林子翔 (B05902131)

蔡淑芬 (R09922A13)

摘要

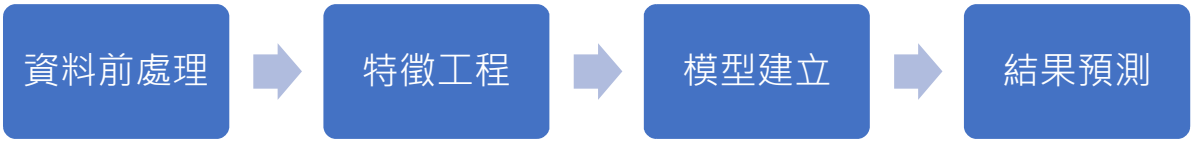
現在信用卡很方便，大家買東西常會刷卡付款，但這也讓不肖人士動了歪念頭，在交易過程中，竊取信用卡資料，盜刷信用卡。面對盜刷，一般民眾除了可以透過經常對帳、防止卡片資訊外洩等方式來避免外，國內外銀行及發卡組織近年也開始運用機器學習演算法找出潛在的盜刷交易，並及早因應。本文提出多種訓練模型來判斷出不符合過去消費習慣的消費紀錄，若能在交易生效前阻止，將有利於銀行、商家與顧客信心消費的局面。本文最後也藉由循序向前選擇法特徵篩選後能選出最少的特徵來快速找到解決這項金融痛點。

關鍵字：信用卡盜刷、機器學習、特徵篩選

A. 問題定義

- 要解決的問題: 我們想要以交易金額、交易類別、消費地城市等資訊來判斷一筆交易是否被盜刷。
- 應用面和重要性：信用卡被盜刷的事件越來越常發生，因應的手段也應該隨之進步。傳統的方法，是將消費行為逐筆用簡訊通知使用者，當使用者發現被盜刷時再打電話給客服，過程太過繁瑣且沒有效率。若是能訓練模型來判斷出當某交易不太符合過去的消費習慣，並在交易生效前阻止，銀行端就可以省下很多時間及金錢成本，而消費者也能對使用信用卡消費產生更多信心。

B. 方法描述



一、資料前處理:

1. 處理缺失值：依照原始資料的眾數來補值。
2. 類別化處理：因為原始資料文字部分是無序的，因此採用 Label Encoding 將文字數據化。
3. 資料特徵縮放：使用 Normalization 的方法，將特徵數據按比例縮放至 0 到 1 的區間。
4. 去除 Outlier：使用 IQR 去除離群值。

二、特徵工程: 使用循序向前選擇法篩選(Forward SFS)選擇適合的特徵。

三、模型建立: 使用決策樹 (Decision Tree)、K-Nearest Neighbors Algorithm (KNN)、Random Forest (RF)、Support Vector Machine (SVM)、ADABoost 及 XGBoost 進行 5-fold Cross Validation 來找出驗證分數最高的模型。

四、結果預測: 使用混淆矩陣 (confusion matrix) 及繪出 ROC 來計算 AUC 以作為預測結果。

由於以前沒有測試集，所以只能利用原始資料切分出驗證集 Validation Set 來進行測試，但我們可以使用真實的測試資料來評估，了解選擇模型是否適合。

C. 資料集

資料來源：

- 玉山人工智慧公開挑戰賽 2019 秋季賽 真相只有一個「信用卡盜刷偵測」（ https://tbrain.trendmicro.com.tw/Competitions/Details/10 ）
- 專案研究--信用卡盜刷偵測業師提供（ http://mirlab.org/jang/courses/finTech/project/2021/ ）

相關特性：

訓練資料共有 1521787 筆資料，測試資料共有 380447 筆資料,訓練資料中沒有盜刷的筆數是 1501432 筆，約有 98.66%，而被盜刷的部分是 20355 筆，占 1.34%。所以是一份 imbalanced 的訓練資料，這也是我們的挑戰點，而測試資料中沒有被盜刷的筆數有 375375 筆，占 98.67%，而被盜刷的部分是 5072 筆，約有 1.33%，如圖一。

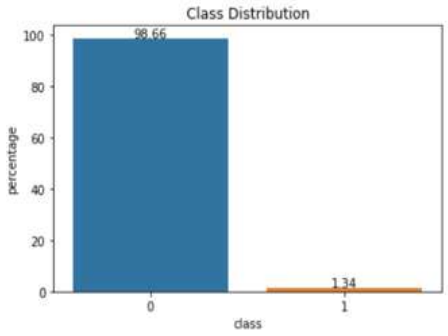


圖 一

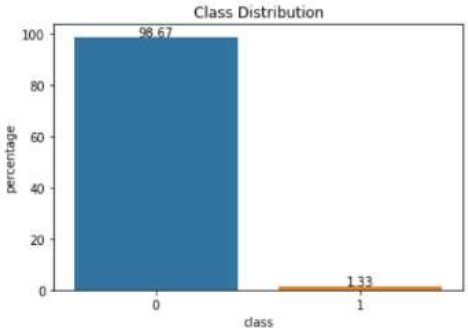


圖 二

而觀察缺失值部分時，訓練資料和測試資料都只有兩組特徵的資料有缺，其中一組是 ***flg_3dsmk***，另一組是 ***flbmk***，訓練資料少了 12581 筆資料，比例約是 0.827%，而測試資料少了 3715 筆，比例約是 0.881%，如表一、表二。

	feature_name	null_total_number	null_ratio
0	flg_3dsmk	3715	0.00881
1	flbmk	3715	0.00881

表一

	feature_name	null_total_number	null_ratio
0	flg_3dsmk	12581	0.00827
1	flbmk	12581	0.00827

表二

相關性的部分，我們計算出兩兩之相關係數，觀察出相關係數最高是 0.63，出現在 ***csmcu*** 及 ***scity*** 之間，因為都在 0.7 以下，非高度相關，所以全部的特徵皆可以交叉使用，如圖三。

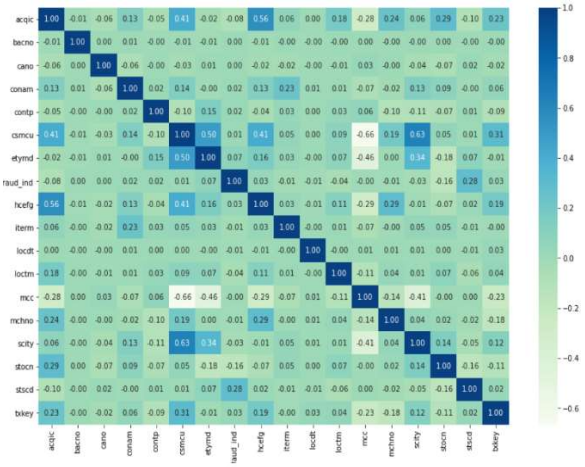


圖 三

所有特徵的總數，利用直方圖及總數資料表格呈現，如圖四及表三，由此也可以觀察出特徵的分布及是否有缺失值。

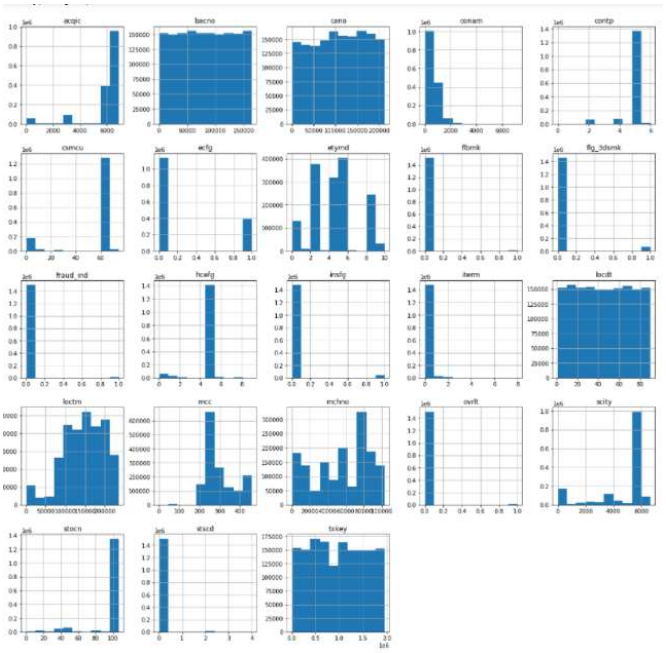


圖 四

	acqic	bacno	cano	conam	contp	csmcu	etymd	fraud_ind	hcefg	iterm
count	1.521787e+06	1.521787e+06	1.521787e+06	1.521787e+06	1.521787e+06	1.521787e+06	1.521787e+06	1.521787e+06	1.521787e+06	1.521787e+06
mean	6.008003e+03	8.209027e+04	1.089170e+05	6.547219e+02	4.829368e+00	5.383324e+01	4.149114e+00	1.337572e-02	4.749410e+00	4.962784e-02
std	1.502420e+03	4.736249e+04	6.090363e+04	4.028078e+02	6.513408e-01	2.072135e+01	2.394259e+00	1.148774e-01	1.111927e+00	3.652493e-01
min	0.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	5.982000e+03	4.140300e+04	5.763500e+04	4.370600e+02	5.000000e+00	6.200000e+01	2.000000e+00	0.000000e+00	5.000000e+00	0.000000e+00
50%	6.716000e+03	8.191200e+04	1.097850e+05	5.919000e+02	5.000000e+00	6.200000e+01	4.000000e+00	0.000000e+00	5.000000e+00	0.000000e+00
75%	6.769000e+03	1.230590e+05	1.618760e+05	8.073400e+02	5.000000e+00	6.200000e+01	5.000000e+00	0.000000e+00	5.000000e+00	0.000000e+00
max	6.884000e+03	1.638840e+05	2.133340e+05	7.208770e+03	6.000000e+00	7.500000e+01	1.000000e+01	1.000000e+00	9.000000e+00	8.000000e+00

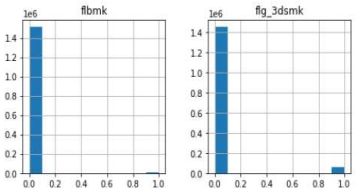
locdt	loctm	mcc	mchno	scity	stocn	stscd	txkey
1.521787e+06	1.521787e+06	1.521787e+06	1.521787e+06	1.521787e+06	1.521787e+06	1.521787e+06	1.521787e+06
4.532732e+01	1.463152e+05	2.978089e+02	5.589022e+04	4.755128e+03	9.565116e+01	2.485499e-02	9.711265e+05
2.601889e+01	5.212107e+04	7.796778e+01	3.082297e+04	1.979815e+03	1.890027e+01	2.216804e-01	5.641322e+05
1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
2.300000e+01	1.109310e+05	2.500000e+02	3.377400e+04	3.795000e+03	1.020000e+02	0.000000e+00	4.869475e+05
4.500000e+01	1.515220e+05	2.640000e+02	5.936000e+04	5.817000e+03	1.020000e+02	0.000000e+00	9.795800e+05
6.800000e+01	1.858270e+05	3.430000e+02	7.920000e+04	5.817000e+03	1.020000e+02	0.000000e+00	1.455200e+06
9.000000e+01	2.359590e+05	4.590000e+02	1.033070e+05	6.671000e+03	1.070000e+02	4.000000e+00	1.958239e+06

表格 三

D. 實驗的設計

一、資料集的清洗和處理

1. 資料集因為有缺失值部分，而觀察出 ***lg_3dsmk*** 及 ***flbmk*** 兩筆資料，明顯發現 NA 的資料較多，所以在缺失值補值部分，我們採用眾數的原則來補數。



2. 而資料內容包含數據型與類別型資料，先使用 LabelEnoder 將文字轉化為離散數值。
3. 再使用 Min Max Normalization 的方法，將特徵數據按比例縮放至 0 到 1 的區間。

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}} \in [0,1]$$

4. 並由盒狀圖得知，如圖五，數值皆落在第一四分位數及第三四分位數之間，所以當我們使用 IQR 方法去除 Outlier 時，資料並沒有改變。

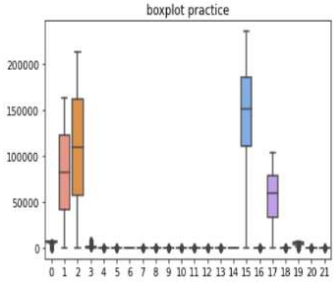


圖 五

二、模型訓練

我們挑選決策樹、KNN、RF、SVC、ADABoost 及 XGBoost，將原訓練資料進行 5-fold Cross Validation 的模型訓練，並且計算正確率，以及利用 confusion matrix 來評估我們的測試資料的結果，可以觀察正確率可以到達 99.17%。

三、特徵選取

我們想利用較少的特徵來測試是否能比較快速達到相同效果，因為有 22 個特徵，如果使用竭盡式特徵選取法 (Exhaustive Feature Selection)來選出最佳的，將需要計算出$2^{22}-1$種組合，這無法達到我們想要的快速的作法，因此我們利用循序向前選擇法篩選 (Forward Sequential Feature Selection) 來當作特徵選取的方法，在篩選的過程中，同樣利用 5-fold Cross Validation 來驗證我們的正確性，發現確實可以用較少的特徵達到接近的效果。例如，用 Decision Tree 分類器做特徵選取的結果，如圖六。

	feature_idx	cv_scores	avg_score
1	(16,)	[0.9882539640817721, 0.9881159686947608, 0.988...	0.988186
2	(3, 16)	[0.9889570834346395, 0.9887533759585752, 0.988...	0.98883
3	(3, 16, 20)	[0.9897489141077284, 0.9896766308097701, 0.989...	0.989644
4	(3, 16, 18, 20)	[0.9896766308097701, 0.9900479041129197, 0.989...	0.989826
5	(2, 3, 16, 18, 20)	[0.9901891851043837, 0.9904651758784063, 0.990...	0.990282
6	(2, 3, 13, 16, 18, 20)	[0.9904093206027113, 0.9906031712654177, 0.990...	0.990377
7	(2, 3, 5, 13, 16, 18, 20)	[0.9906064568698704, 0.9907575946746923, 0.990...	0.990539
8	(2, 3, 5, 13, 15, 16, 18, 20)	[0.9906360273099442, 0.9908857332483457, 0.990...	0.99061
9	(2, 3, 5, 7, 13, 15, 16, 18, 20)	[0.990695168190092, 0.9908857332483457, 0.9908...	0.990691
10	(1, 2, 3, 5, 7, 13, 15, 16, 18, 20)	[0.9906655977500181, 0.9908331635771033, 0.990...	0.99076

圖 六

當選擇[***Cano***、***Conam***、***Mchno***、***Scity***、***Stscd***]，亦即[卡號、交易金額（台幣）、特店名稱、消費地城市、狀態碼]這五個特徵，即可達到正確率 99%以上。

四、其他創意

由於我們的資料集很龐大，不平衡資料過於明顯，所以雖然我們可以從混淆矩陣觀察出盜刷資料的人數比例不高，但正確性仍然很高。

因此，我們想了一個新的解決方法，也就是 BaggingClassifier 的整合分類器，然而這個分類器不能訓練不平衡資料集。當訓練不平衡資料集時，這個分類器將會偏向多數類，從而建立一個有偏差的模型。為了解決這個問題，我們可以使用 imblearn 庫中的 BalancedBaggingClassifier。它允許在訓練整合分類器中的每個子分類器之前，對各個子資料集進行重取樣。重新去訓練決策樹模型後，發現可以從原本找出盜刷比例為 0.49 的狀況下，改善到 0.99，如圖七。

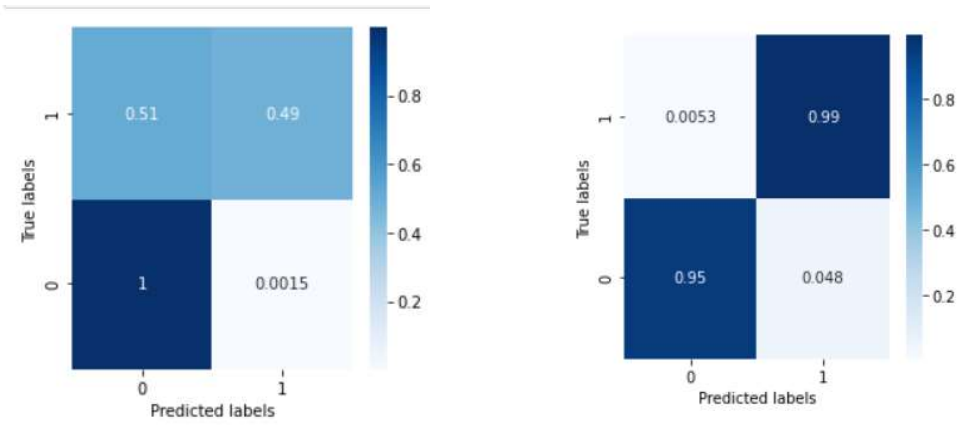


圖 七

E. 實驗結果及討論

以下為各模型代入測試資料的測試結果：

	DT	KNN	RF	ADABoost	XGBoost
Accuracy	0.992	0.986	0.9815	0.988	0.992
Precision	0.811	0.473	0.067	0.583	0.885
Recall	0.488	0.504	0.0417	0.197	0.574
AUC	0.97	0.748	0.517	0.598	0.99

	SVC (SFS)	DT (SFS)	XGBoost (SFS)	DT (BBC)
Accuracy	0.98	0.992	0.993	0.953
Precision	0.058	0.836	0.885	0.219
Recall	0.021	0.462	0.556	0.995
AUC	0.497	0.97	0.99	0.99

所有模型都有 95%以上的 accuracy，然而我們認為 recall 值才是能否真正抓出盜刷者最關鍵的指標（對於 actual negative 即便誤判為盜刷，只要打電話向信用卡持有者確認即可知道為誤判），因此我們選擇 Decision Tree (BBC)及 XGBoost (SFS)做為最終使用的模型，圖八為兩模型的 ROC，其 AUC 均為 0.99。

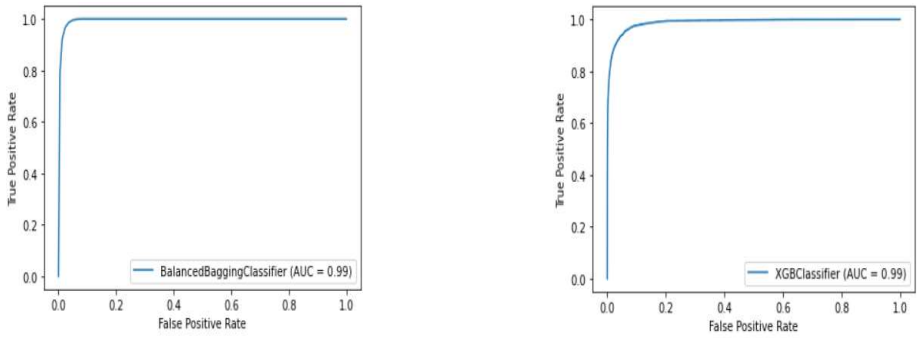


圖 八

F. 結論及展望

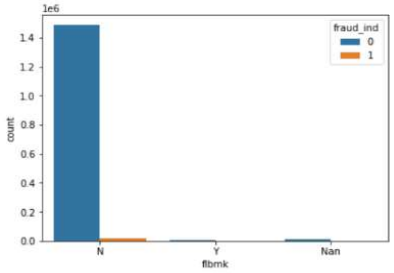
本團隊利用使用者過去消費紀錄，進行刷卡金額、店鋪特性等的分析， 也利用已確認的詐騙紀錄作為樣本，如此更能準確地辨識出盜刷的消費紀錄，也提升整體操作的效率。而我們發現利用交易卡號、交易金額、商店代碼、消費商店國別、以及信用卡狀態碼五個特性進行分析能夠達到最高的準確率，準確率高達 99%。

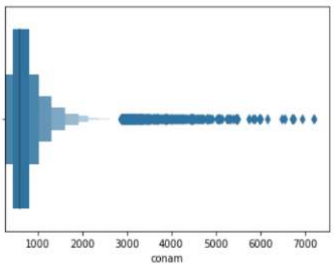
雖然目前現存的盜刷偵測機制也都存在著一定問題，但我們認為在金融科技與機器學習蓬勃發展的情況下，不需要多少年透過科技的進步與數據的準確程度，以及銀行與商家的合作方式都能夠有效地減少盜刷的猖獗。雖然人工智慧金融科技的發展尚未完全成熟，但可以看得出相當有潛力且備受矚目，有朝一日將為商家與顧客帶來雙贏的局面。

G. 附錄

- 參考文獻：
  1. https://scikit-learn.org/stable/index.html
  2. https://toments.com/2487834/
  3. https://www.cnblogs.com/kkkky/p/9389107.html
- 組員執掌 ：
  1. 林晉宇 ： 程式設計、書面資料、投影片、15 分鐘影片彙整
  2. 劉紹凱 ： 3 分鐘廣宣影片、書面資料、投影片
  3. 侯　喆 ： 3 分鐘廣宣影片、書面資料、投影片
  4. 林子翔 ： 程式設計、書面資料、投影片
  5. 蔡淑芬 ： 程式設計、書面資料、投影片、書面報告彙整
- 原始碼：https://github.com/zxfd1100/NTU_FinTech_2021_Spring
- 小組開會記錄：小組會議記錄 - HackMD
- 名詞解釋及資料視覺化:
  - ***flbmk***：Fallback 註記，採用人工授權方式所下的註記，其可能原因為隨機抽樣或是發卡銀行懷疑此筆為不正常交易。

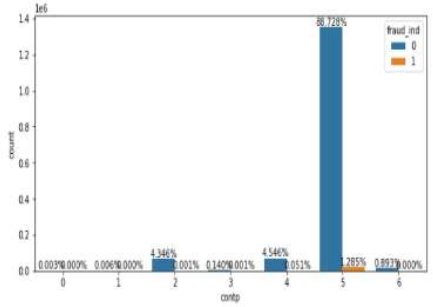
  - ***conam***：交易金額-台幣(經過轉換)

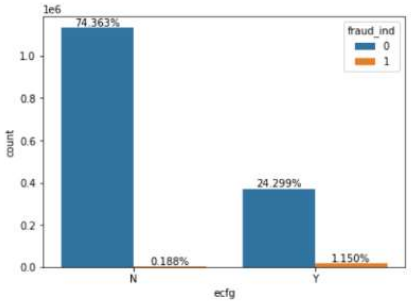




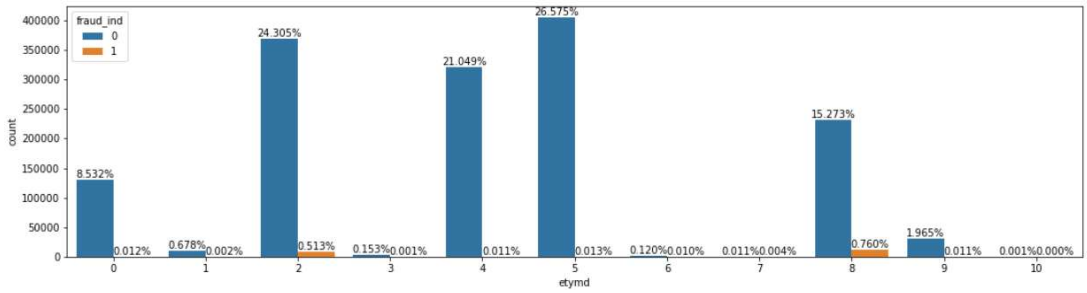
  - ***contp***：交易類別。例如：正向交易、負向交易(刷退)、預借現金等。

  - ***ecfg***：網路交易註記，交易是否在網路上完成。



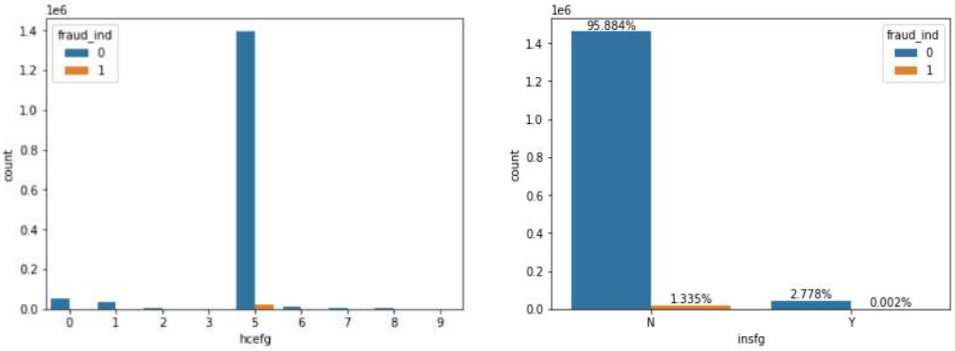


- ***etymd***：交易型態，交易方式。例如：手動輸入卡號、刷磁條機、感應讀取卡號、利用預先儲存的信用卡資料進行交易。



- ***hcefg***：支付形態，虛擬卡交易註記，其型態可能為 Apple Pay、Google Pay、Samsung Pay 等。

- ***insfg***：分期交易註記，交易是否有分期。



- ***stscd***：狀態碼，卡片狀態。例如：停卡、掛失、逾期等。