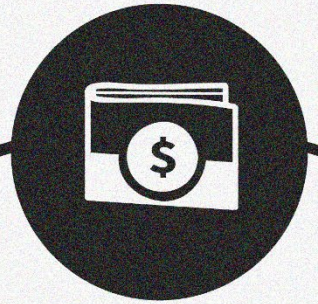


IIE Final Project

MEGA 防護罩

109034523 黃浩銓

Agenda

01 背景簡介

02 問題定義

03 模型建構

04 模型改善

05 結論

01 背景簡介

01 研究背景

- 智慧手機的普及化，互聯網的崛起，電子支付成為近年熱門的話題
- 互聯網是一個開放系統，資安安全有待考慮
- 研究指出2018-2019在網路上交易被詐騙，信用卡被盜刷的金額高達30億元
- 有時用戶被盜刷了無法立即得知
  且往往被盜刷一筆之後就會緊接著被盜刷下一筆款項，造成2次傷害。

01 研究目的

為了解決網路交易信用卡被盜刷卻沒有辦法即時監測的問題

透過5W1H分析法了解問題

使用機器學習的方式，並根據模型分類結果進行訓練與改善

提供給個人用戶或是銀行等金融機構。

02 問題定義

02 5W1H

What 解決信用卡被盜刷卻沒能被準確的偵測出來

When 盜刷信用卡時卻沒有即時偵測出來

Who 平台工程師與金融機構人員

Why 讓使用者信用卡被盜刷時卻無法被即時通知導致無法降低傷害

Where 電子商務

How 建構分類模型，協助金融機構快速且準確的得知信用卡是否有被盜刷或是詐騙

03 模型建構

03 模型架構

資料輸入

資料前處理

模型建構

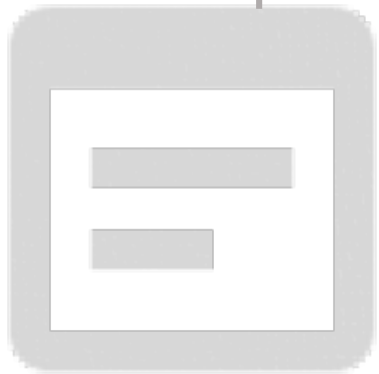
結果輸出

03 資料輸入

- 透過Kaggle線上開放式資料庫所提供之BankSim**銀行**付款模擬器

- 總共有594,643筆資料, 其中有587443筆為正常交易的 7200筆資料為詐騙

  每一筆資料有10個項目



03 資料輸入

(1)step：代表這筆資料是從開始收集經過幾天。總共是0-180，代表6個月

(2)customer：用戶id

(3)age：分類年齡(有8種)

(4)gender：分類性別(有4種)

(5)zipporderOri：郵遞地址

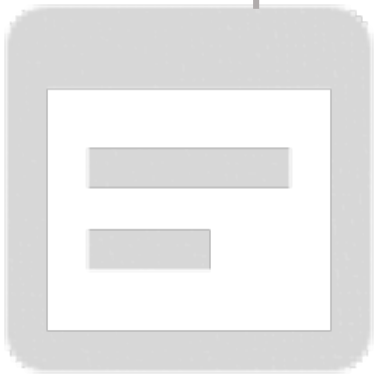
(6)merchant：貿易商id

(7)zipMerchant：貿易商郵遞地址

(8)category：分類消費項目(有15種)

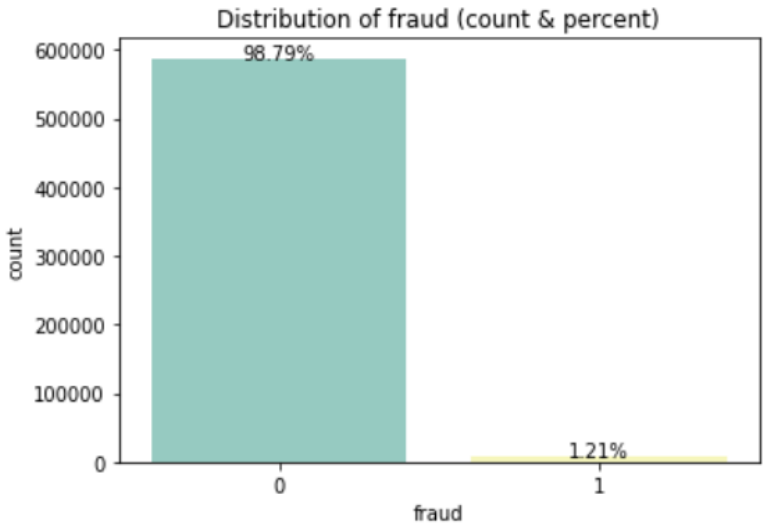
(9)amount：刷卡金額

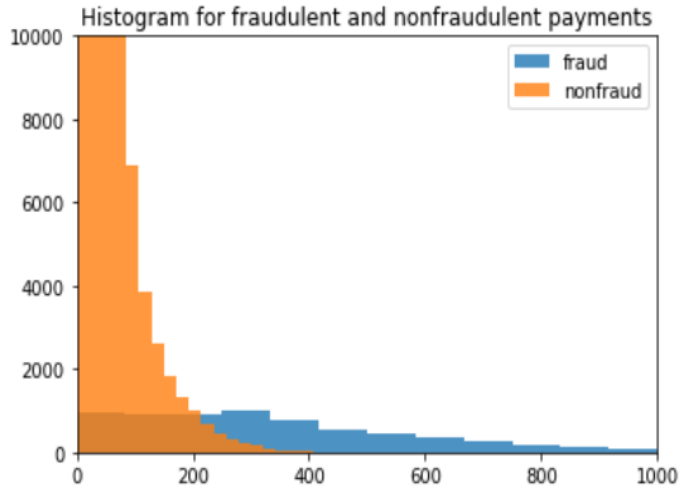
(10)fraud：是否被詐騙



03 資料前處理

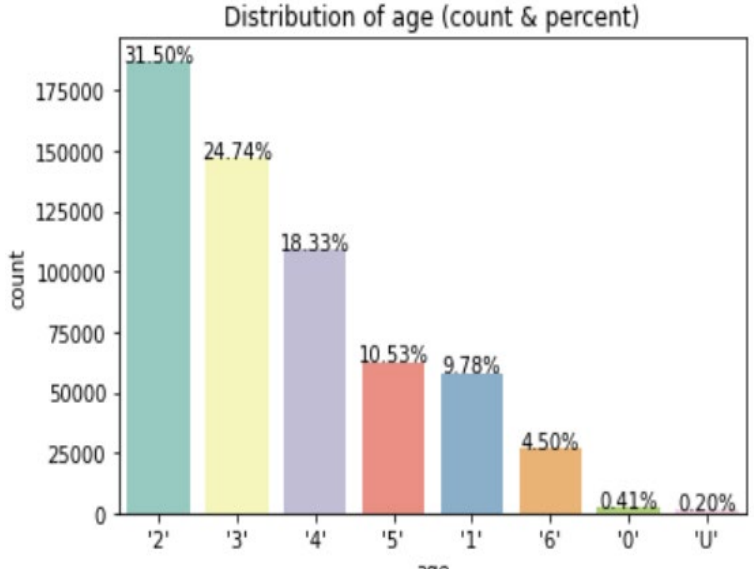
Fraud

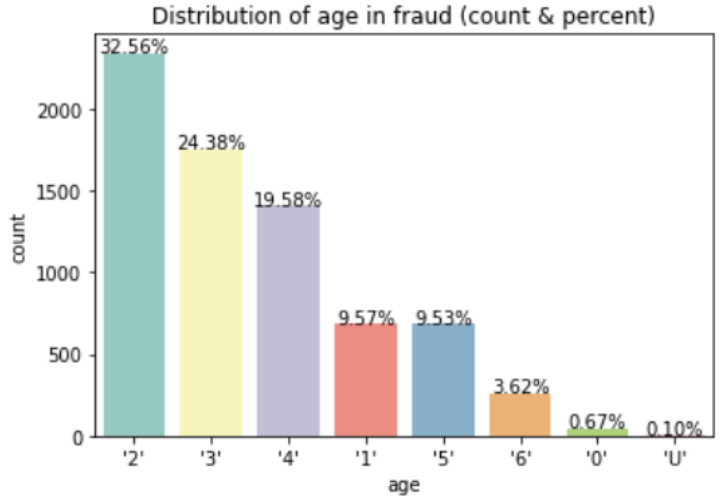


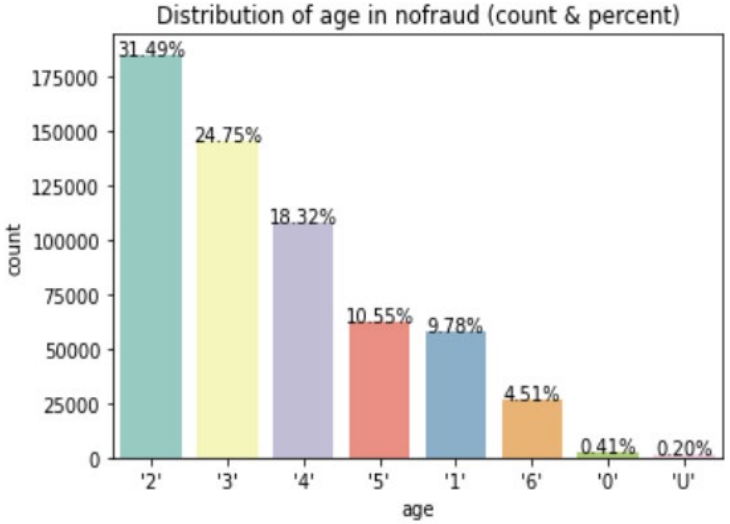


03 資料前處理

Age

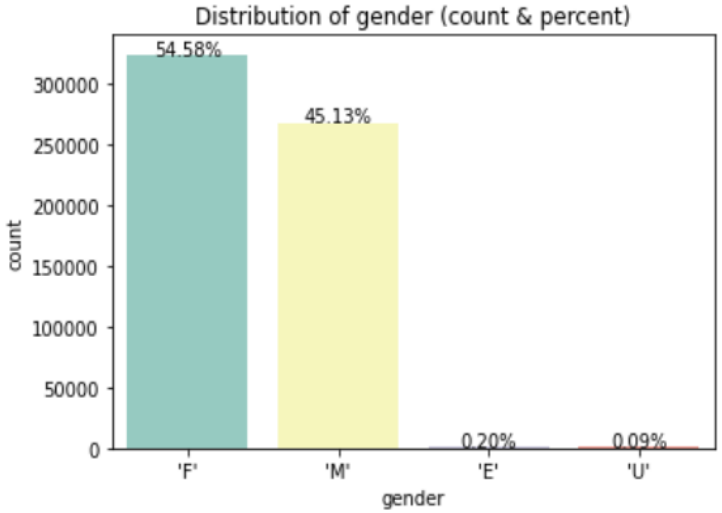

Distribution of age (count & percent)
count
31.50%
24.74%
18.33%
10.53%
9.78%
4.50%
0.41%
0.20%
'2'
'3'
'4'
'5'
'1'
'6'
'0'
'U'

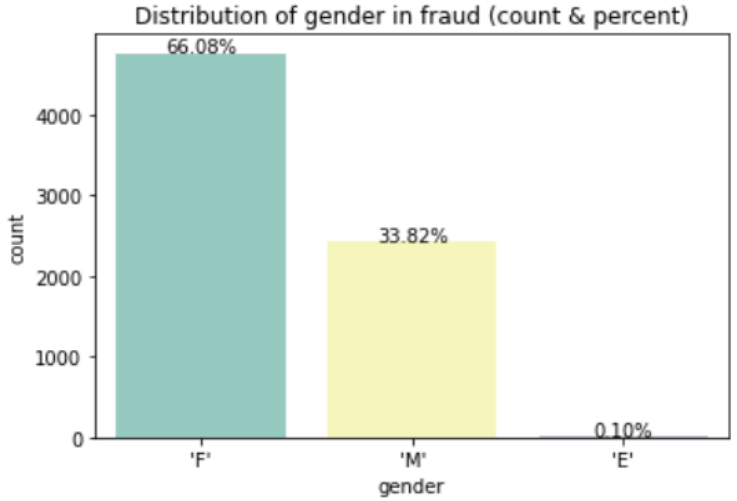

Distribution of age in fraud (count & percent)
count
32.56%
24.38%
19.58%
9.57%
9.53%
3.62%
0.67%
0.10%
'2'
'3'
'4'
'1'
'5'
'6'
'0'
'U'
age

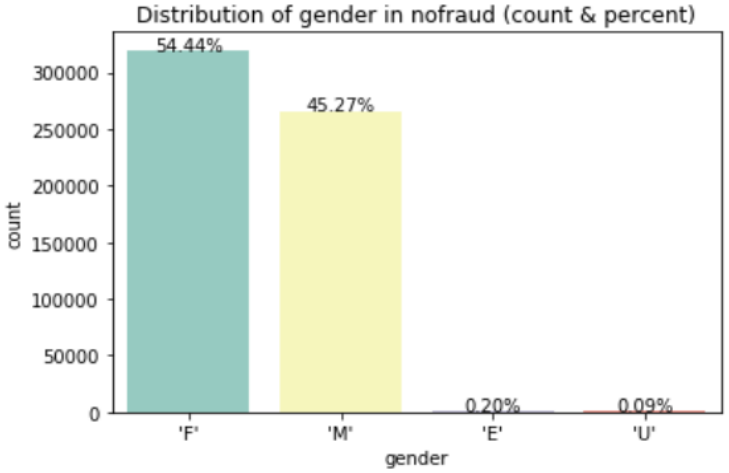

Distribution of age in nofraud (count & percent)
count
31.49%
24.75%
18.32%
10.55%
9.78%
4.51%
0.41%
0.20%
'2'
'3'
'4'
'5'
'1'
'6'
'0'
'U'
age

03 資料前處理

Gender

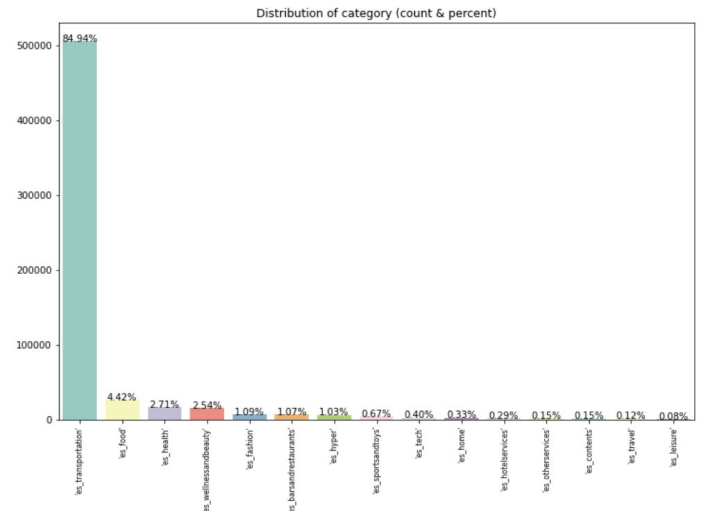


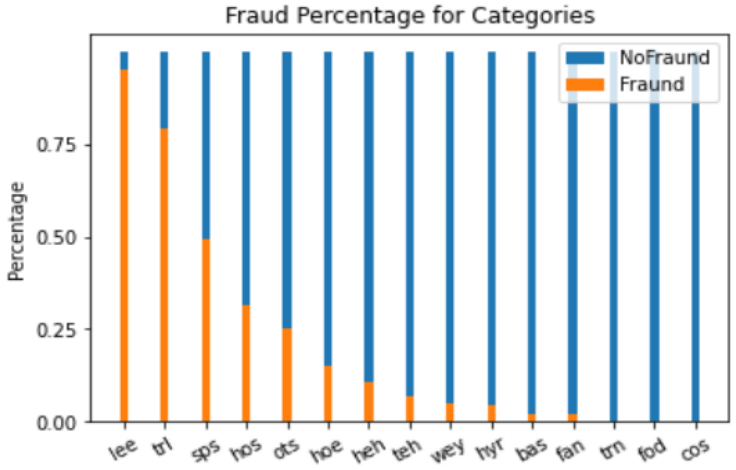




03 資料前處理

Category





03 資料前處理

- 將正常交易做為第 0 類(資料集顯示為0)，詐騙為第 1 類(資料集顯示為0)
- 捨棄掉ziporderOri、zipMerchant的欄位
- 改變欄位中非數值之資料，

  如：Gender (‘M’、’F’ 、’ E’ 、’U’)

  Age(‘0’、’1’、’2’、’3’、’4’、’5’、’6‘ 、’U’)

03 資料前處理

- 首先將 594643筆資料先分出10%做為測試集，共約59465筆資料
- 由於資料集非常不平衡(由fraud那一欄位可得知)，會影響到訓練的準確性，因此本研究首先透用過採樣(SMOTE)的方式，共得到528686筆資料
- 將這528686筆資料，拆分為90%做為訓練集，共約475817 筆
  10%做為驗證集，共約 52869筆

03 模型建構

03 模型建構

RandomForest

本研究針對此模型進行訓練，分析不同超參數模型的分類結果。
並且訓練過程改變樹的深度　，迭代次數，做為模型的超參數。
詳細參數如下：

```
rf_clf  =  RandomForestClassifier(n_estimators=200, max_depth=13, random_state=42,
                                  verbose=1, class_weight="balanced")
```

並得到我們的輸出結果：

Accuracy: 0.988
Precision: 0.491
Recall: 0.927

04 模型改善

04 模型改善

CatBoost

減少了對超參數調優的需求，降低了過度擬合的機會，
這也使得模型變得更加具有通用性。同時可以處理類別型特徵，
並且可以實現特徵組合，使模型效果更快更好。參數如下：

```
clf = CatBoostClassifier(iterations=100,
                         learning_rate=0.01,
                         depth=6,
                         eval_metric='AUC',
                         random_seed = 42,
                         bagging_temperature = 0.2,
                         metric_period = 20
                         )
```

並得到我們的輸出結果：

Accuracy: 0.997
Precision: 0.91
Recall: 0.65

04 模型改善

超參數調整

針對Catboost進行超參數調整，並且加入過擬合的機制。

```
clf = CatBoostClassifier(iterations=200,
                         learning_rate=0.04,
                         depth=12,
                         eval_metric='Recall',
                         random_seed = 42,
                         bagging_temperature = 0.2,
                         od_type='Iter',
                         metric_period = 20,
                         od_wait=25
                         )
```

並得到我們的輸出結果：

Accuracy: 0.997

Precision: 0.94

Recall: 0.82

04 模型改善

改變訓練集，測試集，驗證集大小

我們比較3.2.2和3.2.1的結果可以發現recall卻上升了，
但是卻與原本3.1.6的recall還要小，代表實際是詐騙，
但是卻沒有發現的機率提高了。
因此猜想有可能是因為數量少的類別重複抽取太多次，導致模型發生過擬和

並得到我們的輸出結果：

```
Accuracy: 0.997
Precision: 0.94
Recall: 0.86
```

05 結論

05 研究結果

根據本研究改善後所提出的模型，準確率相比之前有微幅的的提升，但是在精確率可以看到我們提出來的新模型比起原本的模型還要進步許多。雖然召回率的部分下降了一點，但是跟精確率提升的部分相比而言，這點損失我們是承受得起的。

原模型輸出結果：

Accuracy: 0.988
Precision: 0.491
Recall: 0.927

▶

改善後模型的輸出結果：

Accuracy: 0.997
Precision: 0.94
Recall: 0.86

05 未來改善

本研究只測試 Catboost 9種參數，但是實際上可調參數高達102種

因此未來可以研究不同參數所帶來的改變的是什麼，以及分析不

同參數組合所帶來的效果是值得去探討的

Thank you

感謝各位的蒞臨與指教