

網路資料科技應用期末報告

— 信用卡用戶流失預測

組員：日四 A 06122145 魯瑞慈

日四 B 06122247 吳英緩

研究動機

「客戶是否流失」一直是許多公司在探討的問題，然而客戶流失是必然的，因此了解客戶「為何流失」便是重要的課題。在現在信用卡服務較密集的狀況下，如何留住目標客戶是關鍵問題。要想在瞬息萬變的世界下存活，除了分析過去客戶流失的原因以外，能夠事先預測未來用戶的流失並改善將有利於公司未來發展。因此本次報告將利用網路上公開的數據集，進行數據分析，藉由實際操作來探究此數據集的客戶流失特徵及預測，並探討此公司可以如何訂定對策防止客戶流失。

一、資料說明

藉由 kaggle 上的公開數據集「Credit Card customers」預測銀行信用卡流失，此數據集的來源網站為 leaps.analyttica.com，此網站解釋了如何解決特定商業問題。數據集中包含 10,127 筆客戶資料，提供年齡、工資、婚姻狀況、信用卡額度、信用卡類別等，21 個特徵。

資料集名稱：BankChurners.csv

欄位名稱	欄位內容
CLIENTNUM	持有該帳戶的用戶唯一識別編號
Attrition_Flag	客戶是否續約信用卡服務
Customer_Age	用戶年齡
Gender	用戶性別
Dependent_count	家庭人數
Education_Level	教育水平
Marital_Status	婚姻狀況
Income_Category	收入分布
Card_Category	持有卡片類型，分為藍、銀、金、鉑金卡
Months_on_book	每月帳單數量
Total_Relationship_Count	用戶持有銀行業務數量
Months_Inactive_12_mon	用戶不活躍月份數
Contacts_Count_12_mon	過去一年中與銀行接觸次數
Credit_Limit	信用卡額度
Total_Revolving_Bal	循環信用總量
Avg_Open_To_Buy	過去一年平均開放的信貸限額
Total_Amt_Chng_Q4_Q1	第四季度與第一季度間交易金額變化
Total_Trans_Amt	過去一年總交易金額
Total_Trans_Ct	過去一年總交易次數
Total_Ct_Chng_Q4_Q1	第四季度與第一季度間交易數量變化

Avg_Utilization_Ratio	平均卡片利用率
-----------------------	---------

二、匯入套件和資料

首先匯入此次資料分析和預測模型所需要使用的套件。

圖 2-1 匯入套件

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as ex
import plotly.graph_objs as go
import plotly.figure_factory as ff
from plotly.subplots import make_subplots
import plotly.offline as pyo
pyo.init_notebook_mode()
sns.set_style('darkgrid')
from sklearn.decomposition import PCA
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import f1_score as f1
from sklearn.metrics import confusion_matrix
import scikitplot as skplt
plt.rc('figure', figsize=(18,9))
%pip install imbalanced-learn
from imblearn.over_sampling import SMOTE
```

接著匯入此次資料集，由於此原始資料集的倒數 2 欄並非特徵欄末，因此僅查看主要的 21 個特徵欄位。

圖 2-2 匯入資料

```
c_data = pd.read_csv('BankChurners.csv')
c_data = c_data[c_data.columns[:-2]]
c_data
```

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category	Months_on
0	768805383	Existing Customer	45	M	3	High School	Married	60K–80K	Blue	
1	818770008	Existing Customer	49	F	5	Graduate	Single	Less than \$40K	Blue	
2	713982108	Existing Customer	51	M	3	Graduate	Married	80K–120K	Blue	
3	769911858	Existing Customer	40	F	4	High School	Unknown	Less than \$40K	Blue	
4	709106358	Existing Customer	40	M	3	Uneducated	Married	60K–80K	Blue	
...
10122	772366833	Existing Customer	50	M	2	Graduate	Single	40K–60K	Blue	
10123	710638233	Attrited Customer	41	M	2	Unknown	Divorced	40K–60K	Blue	
10124	716506083	Attrited Customer	44	F	1	High School	Married	Less than \$40K	Blue	
10125	717406983	Attrited Customer	30	M	2	Graduate	Unknown	40K–60K	Blue	
10126	714337233	Attrited Customer	43	F	2	Graduate	Married	Less than \$40K	Silver	

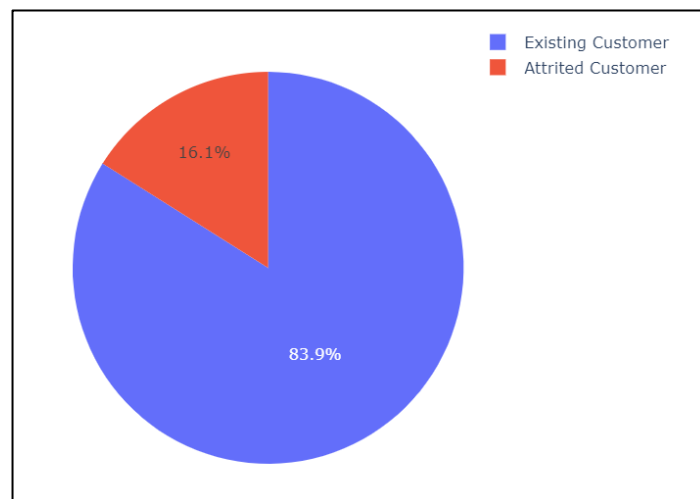
10127 rows × 21 columns

三、數據分析

1、分析目標變數—Attrition_Flag

首先我們先查看此次資料集的目標變數 Attrition_Flag—代表用戶是否續約信用卡服務，繪製圓餅圖後發現在此資料集的用戶中，有 83.9% 的用戶有續約服務，而沒有續約的用戶比例則有 16.1%。

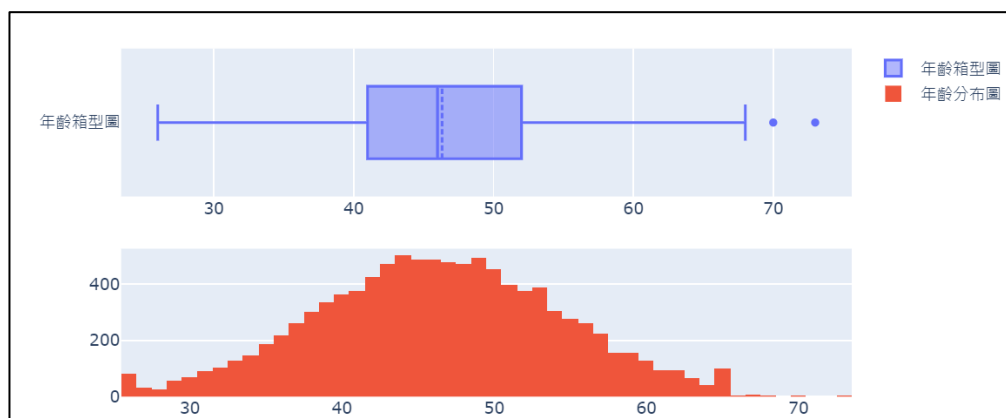
圖 3-1 信用卡用戶流失比率



2、分析特徵變數—Customer_Age

在客戶年齡分布圖形中我們可以發現介於 40-50 歲用戶為集中區段，且 50 歲前後的用戶數量又較多。平均年齡大約為 46 歲，70 歲以上出現離群值。

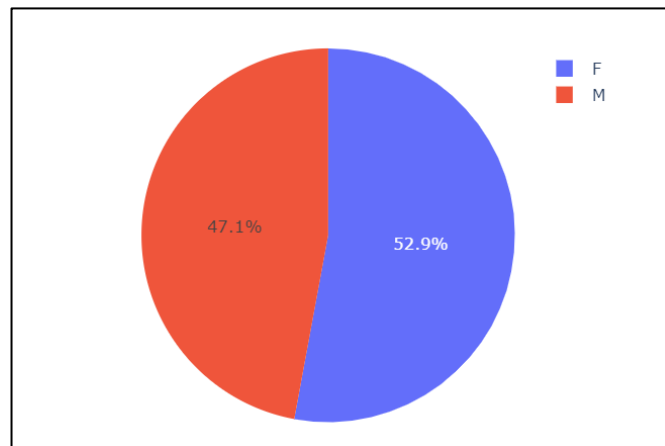
圖 3-2 用戶年齡分布



3、分析特徵變數—Gender

此資料集中我們可以看到男性用戶比例為 47.1% 而女性用戶比例為 52.9%，後者占比較多。

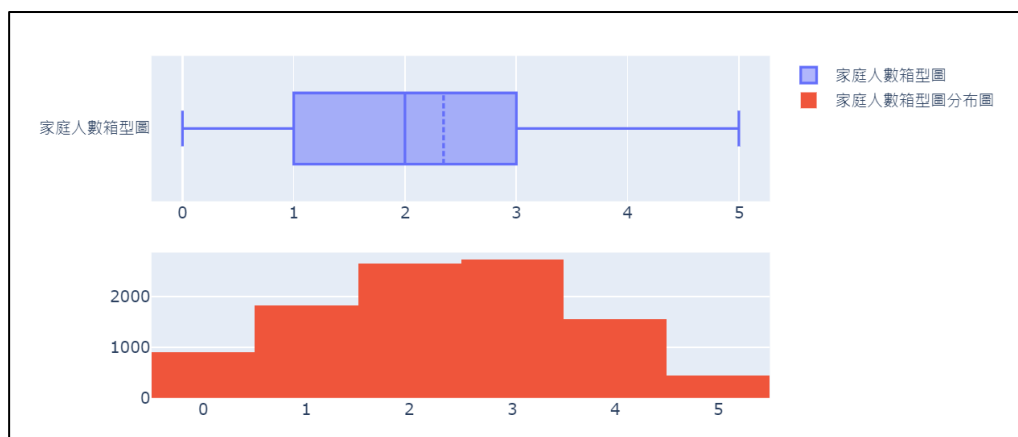
圖 3-3 用戶性別比



4、分析特徵變數—Dependent_count

家庭人數狀況的分布上平均為 2 至 3 人，1 至 3 人的單身貴族或小家庭為集中數，共 3 位家庭成員的用戶數量最高，為常態分佈。

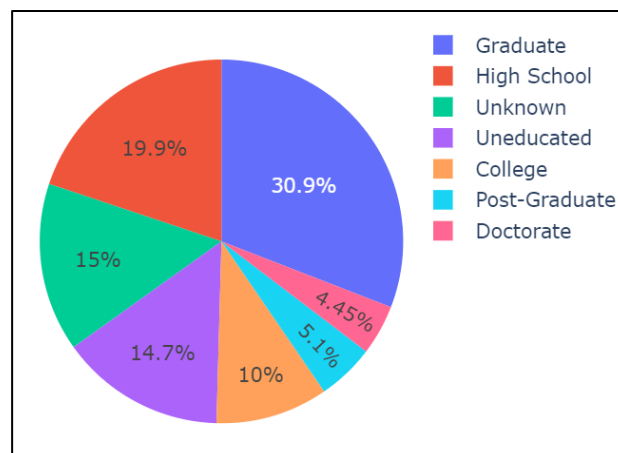
圖 3-4 家庭人數分布



5、分析特徵變數—Education_Level

此圓餅圖中我們可以發現已經畢業的社會人士為大眾用戶，占比為 30.9%。博士學位的用戶則占少數 4.45%。

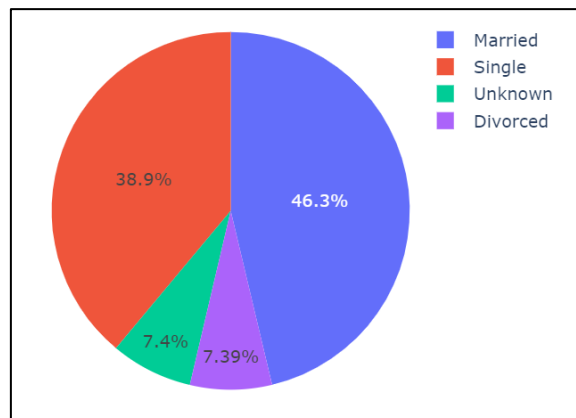
圖 3-5 教育程度分布



6、分析特徵變數—Marital_Status

用戶在申辦時或許在可選擇不填寫的狀況下而出現了無法得知的數據。在有填寫的用戶中佔比最高為結婚狀態 46.3%，最低比例是離婚狀態 7.39%。

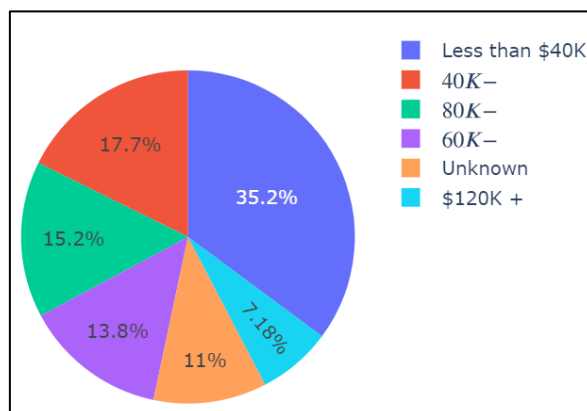
圖 3-6 婚姻狀態分布



7、分析特徵變數—Income_Category

大宗用戶中收入情況小於 4 萬美金的比例為 35.2%，而收入大於 12 萬美金的用戶則為較少的 7.18%。

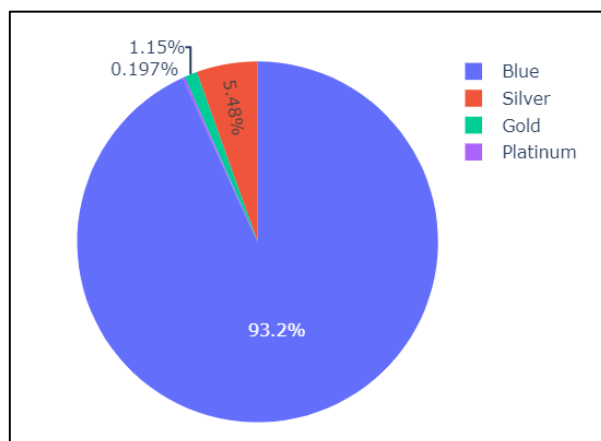
圖 3-7 收入狀態分布



8、分析特徵變數—Card_Category

用戶卡片等級中以普通卡（藍卡）為大眾，占比 93.2%。而高等級的白金卡則占比最少 0.197%。

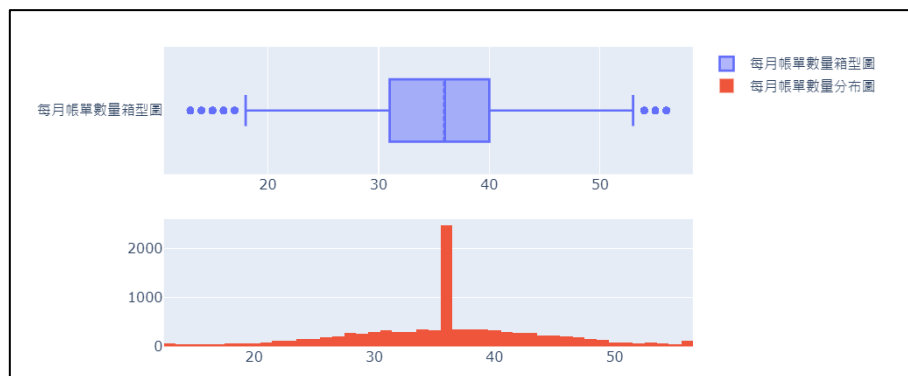
圖 3-8 卡片等級分布



9、分析特徵變數—Months_on_book

統計並視覺化「Months_on_book（每月帳單數量）」後發現，分布圖的峰度極高，代表標準差大，此欄位資料具有集中趨勢，且資料並非常態分布。

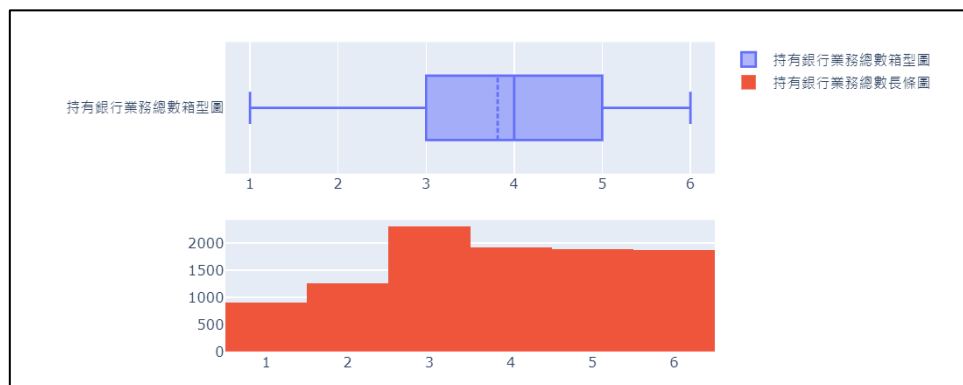
圖 3-9 每月帳單數量



10、分析特徵變數—Total_Relationship_Count

資料集中持有銀行業務總數的平均值為近 4 項，最大值區間介於 3 至 5 項，最多總數用戶持有 3 項銀行業務。

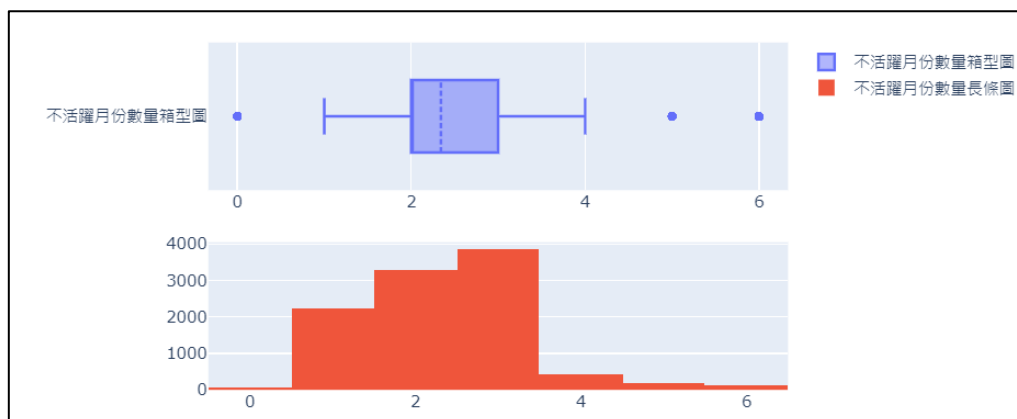
圖 3-10 持有銀行業務總數



11、分析特徵變數—Months_Inactive_12_mon

統計並視覺化後可發現在兩端都出現了離群值，可得知在兩端極值數量較少，平均不活躍月分則為春季 2 至 3 月。

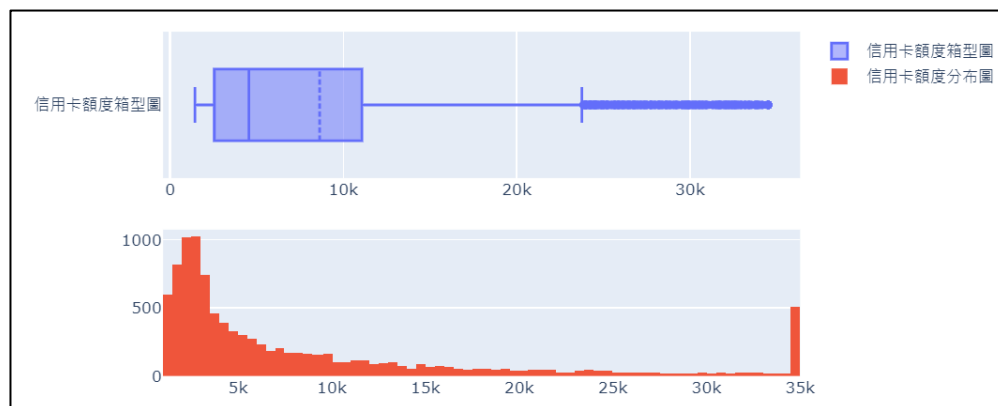
圖 3-11 不活躍月份數量



12、分析特徵變數—Credit_Limit

統計並視覺化我們發現在 2 萬美金額度以上出現了大量的離群值，可以得出此資料非常態分布，但由於大於 3 萬 5 美金額度的用戶數量相對多，即使最大區間的額度都較小但平均數因此被拉升，大約為近 1 萬。

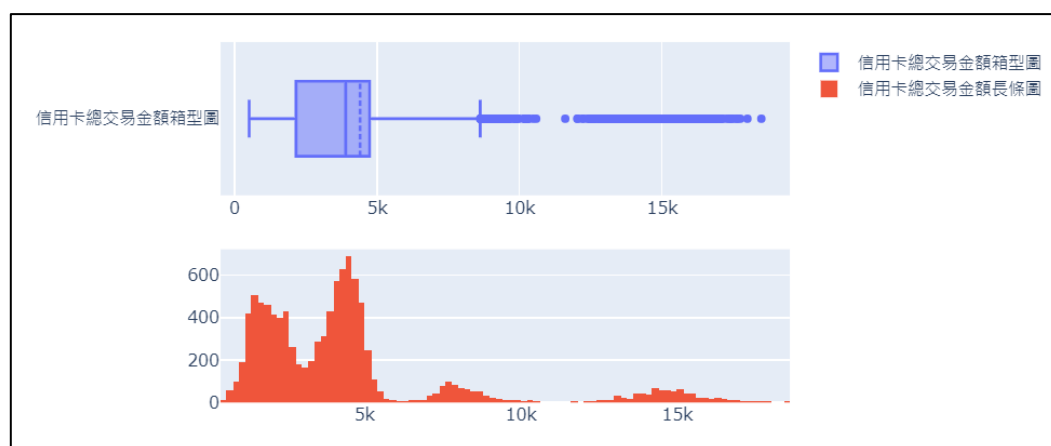
圖 3-12 信用卡額度



13、分析特徵變數—Total_Trans_Amt

此箱型圖及長條圖中可以看到大於 1 萬的總交易金額出現連續離群值，也可以發現在 1 萬出頭和大於 1 萬五的的交易金額出現斷層，平均值則為 5000 美金左右。

圖 3-13 信用卡交易總金額

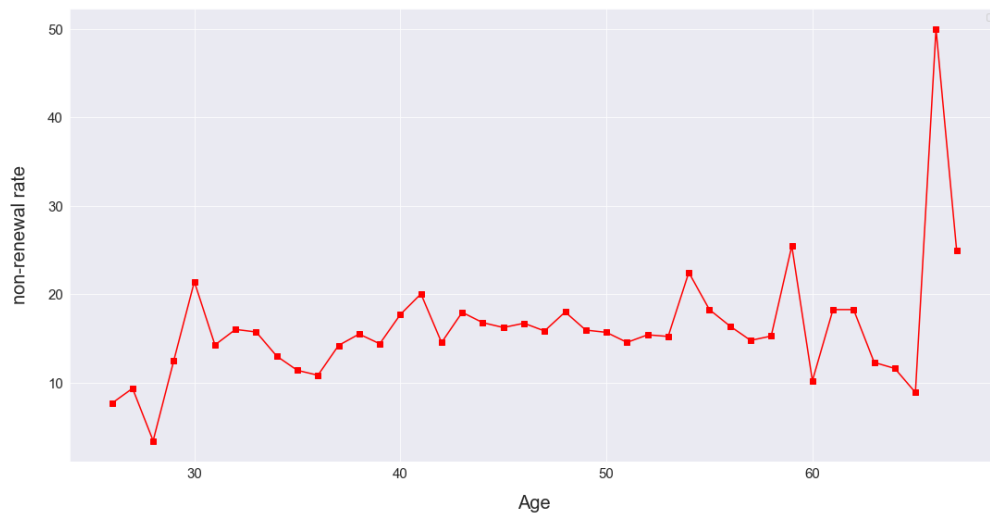


四、交叉分析

1、年齡及不續約比例

利用交叉分析之方式可以得到詳細的年齡、人數、不續約比例數值的表格，因為資料量較多透過折線圖快速獲得大於 60 歲的極端值的用戶不續約的情況最高。因此可以提出一些針對年長者的優惠方案。並對於近 30 歲的用戶重點進行觀察及溝通了解突然不續約的原因，以拉低之後不續約的整體比例。

圖 4-2 年齡不續約比例折線圖



2、性別及不續約比例

在用戶較多為女性的狀況下同樣不續約比例差不多，因此可男女並非太大的影響關聯。

圖 4-3 性別不續約比例表格

	Gender	people	Attrition_Flag	不續約比例
0	F	5358	930	17.357223
1	M	4769	697	14.615223

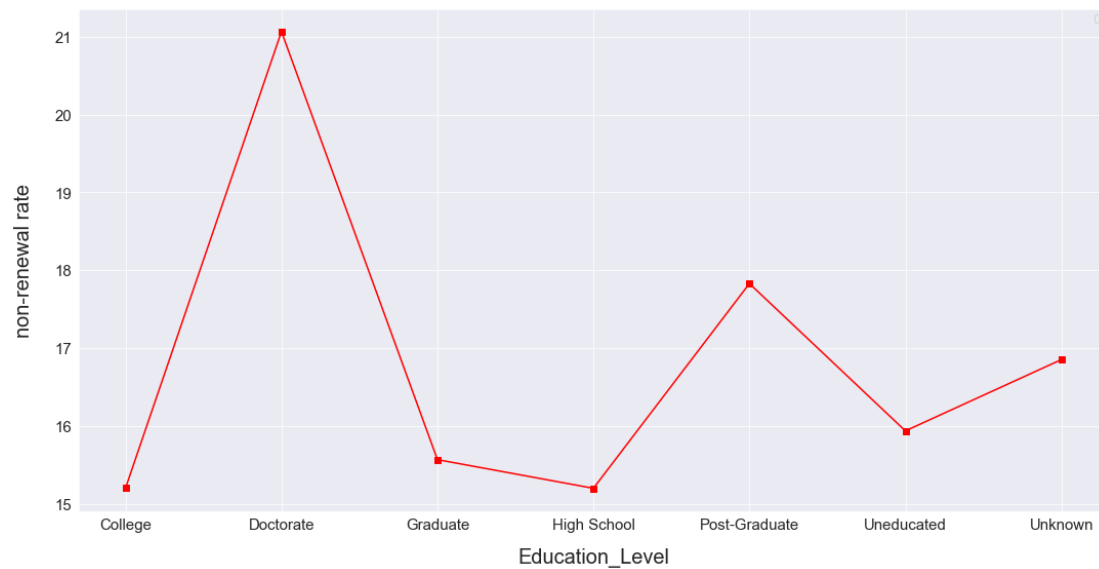
3、教育水平及不續約比例

從此兩張圖我們可以看到博士學位之用戶不續約比例最高而大學學歷及社會人士不續約比例則較低，博士學歷持卡人消費水準較高相對較高的情況下，如何使用者提高單筆簽帳金的消費是可以進行進一步優化的方向。

圖 4-4 教育水平不續約比例表格

	Education_Level	people	Attrition_Flag	不續約比例
0	College	1013	154	15.202369
1	Doctorate	451	95	21.064302
2	Graduate	3128	487	15.569054
3	High School	2013	306	15.201192
4	Post-Graduate	516	92	17.829457
5	Uneducated	1487	237	15.938130
6	Unknown	1519	256	16.853193

圖 4-5 教育水平不續約比例折線圖



4、婚姻狀態及不續約比例

由於此表格顯示的不續約比例相差較少，因此我們也可以推斷婚姻狀態並非用戶不續約之原因。

圖 4-6 婚姻狀態不續約比例表格

	Marital_Status	people	Attrition_Flag	不續約比例
0	Divorced	748	121	16.176471
1	Married	4687	709	15.126947
2	Single	3943	668	16.941415
3	Unknown	749	129	17.222964

5、收入狀況及不續約比例

交叉分析後我們可以看到在收入小於 4 萬美金的最大眾用戶中相對不續約比例也為第二高，針對消費能力或許因為收入較被動的狀況可以提出相關現金回饋或紅利以增加續約之意願。

圖 4-7 收入狀況不續約比例表格

	Income_Category	people	Attrition_Flag	不續約比例
0	\$120K +	727	126	17.331499
1	40K—60K	1790	271	15.139665
2	60K—80K	1402	189	13.480742
3	80K—120K	1535	242	15.765472
4	Less than \$40K	3561	612	17.186184
5	Unknown	1112	187	16.816547

6、卡片等級及不續約比例

交叉分析過後的此表格我們可以看到等級最高的用戶不續約比例也為最高，但因為人數相對小眾的狀況下且等級之間不續約比例的數據相差無幾因此可以針對持有普遍等級信用卡的用戶降低不續約之比例。

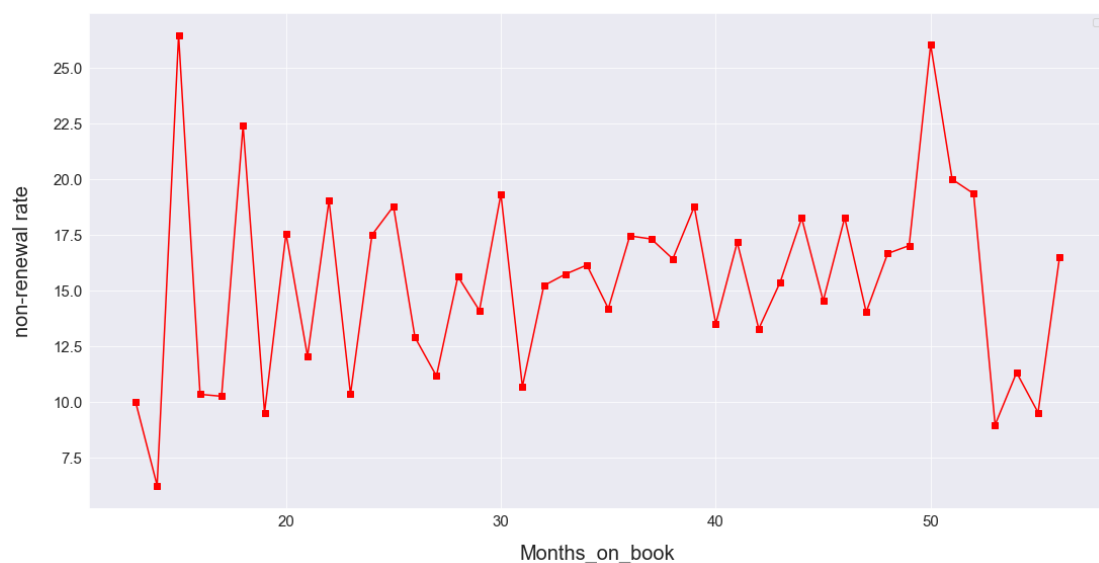
圖 4-8 卡片等級不續約比例表格

Card_Category	people	Attrition_Flag	不續約比例
0	Blue	9436	1519 16.097923
1	Gold	116	21 18.103448
2	Platinum	20	5 25.000000
3	Silver	555	82 14.774775

7、每月帳單數及不續約比例

從此折線圖我們可以看到兩個高峰都接近極值，並且都呈現一個暴增和急遽減少的數據，從此圖形我們可以得出在極值狀況下的每月帳單數都有可能造成不續約的原因。

圖 4-9 每月帳單數不續約比例折線圖



8、持有銀行業務總數及不續約比例

從此分析表格可看出持有銀行業務總數愈高時不續約比例呈遞減趨勢，可以得出當用戶擁有愈多銀行業務總數時依附度可能也愈高，每項銀行業務之間的相關方案愈高的話便能留住用戶繼續持卡。

圖 4-10 持有銀行業務總數不續約比例表格

Total_Relationship_Count	people	Attrition_Flag	不續約比例
0	1	910	233 25.604396
1	2	1243	346 27.835881
2	3	2305	400 17.353579
3	4	1912	225 11.767782
4	5	1891	227 12.004231
5	6	1866	196 10.503751

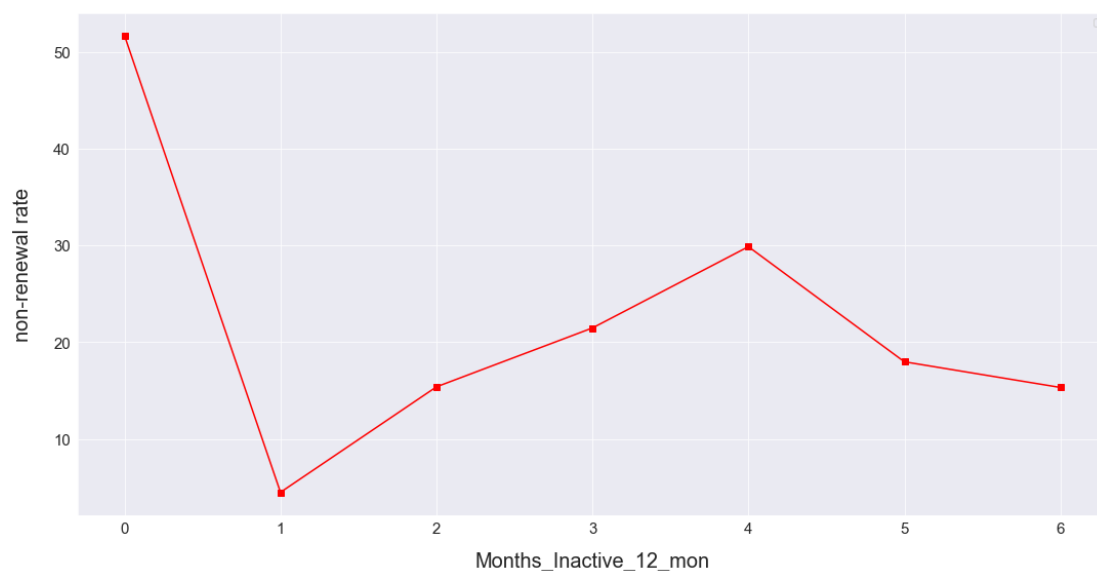
9、不活躍月份及不續約比例

交叉分析表格中可發現其中佔比最高的為未知月分，由於此原資料有數值上的偏差，因此可透過其他用戶資料輔助出重典月份去做改善。

圖 4-11 不活躍月份不續約比例表格

Months_Inactive_12_mon	people	Attrition_Flag	不續約比例	
0	0	29	15	51.724138
1	1	2233	100	4.478280
2	2	3282	505	15.386959
3	3	3846	826	21.476859
4	4	435	130	29.885057
5	5	178	32	17.977528
6	6	124	19	15.322581

圖 4-12 不活躍月份不續約比例折線圖



10、信用卡額度及不續約比例

交差分析完後發現有 6205 筆信用卡額度資料，因此抓取不續約比例最高 100% 的額度共有 449 筆，區間額度介於 1440 至 2994 美金之間。能夠預知先用的額度較低的情況下用戶不續約比例較高，可以鼓勵持卡人頻繁且穩定的使用、期限內還款或是善用分期付款來增加額度。

圖 4-13 信用卡額度不續約比例表格

	Credit_Limit	people	Attrition_Flag	不續約比例
2	1440.0	1	1.0	100.0
4	1442.0	1	1.0	100.0
6	1446.0	1	1.0	100.0
8	1451.0	2	2.0	100.0
10	1454.0	1	1.0	100.0
...
1293	2983.0	1	1.0	100.0
1296	2986.0	1	1.0	100.0
1298	2989.0	1	1.0	100.0
1301	2992.0	1	1.0	100.0
1303	2994.0	1	1.0	100.0

11、信用卡交易總額及不續約比例

交差分析完後發現有 5033 筆信用卡交易總額資料，因此抓取不續約比例最高 100%的額度共有 449 筆，區間介於 510 美金至 2304 美金。因此我們可以大致推斷若是刺激用戶消費則可提高續約比例。

圖 4-14 信用卡交易總額不續約比例表格

	Total_Trans_Amt	people	Attrition_Flag	不續約比例
0	510	1	1.0	100.0
1	530	1	1.0	100.0
2	563	1	1.0	100.0
3	569	1	1.0	100.0
4	594	1	1.0	100.0
...
1243	2286	1	1.0	100.0
1247	2290	1	1.0	100.0
1253	2296	1	1.0	100.0
1258	2302	1	1.0	100.0
1259	2304	1	1.0	100.0

449 rows × 4 columns

五、資料預處理

1. 類別型資料轉換

首先查看此資料集的各個特徵型態，可以發現有 6 個特徵屬於類別型資料，須將他們進行轉換才能進行主成分分析及建置預測模型。

圖 5-1 資料型態

```
c_data.info()
```

<class 'pandas.core.frame.DataFrame'>			
RangeIndex: 10127 entries, 0 to 10126			
Data columns (total 21 columns):			
#	Column	Non-Null Count	Dtype
0	CLIENTNUM	10127 non-null	int64
1	Attrition_Flag	10127 non-null	object
2	Customer_Age	10127 non-null	int64
3	Gender	10127 non-null	object
4	Dependent count	10127 non-null	int64
5	Education_Level	10127 non-null	object
6	Marital_Status	10127 non-null	object
7	Income_Category	10127 non-null	object
8	Card_Category	10127 non-null	object
9	Months_on_book	10127 non-null	int64
10	Total_Relationship_Count	10127 non-null	int64
11	Months_Inactive_12_mon	10127 non-null	int64
12	Contacts_Count_12_mon	10127 non-null	int64
13	Credit_Limit	10127 non-null	float64
14	Total_Revolving_Bal	10127 non-null	int64
15	Avg_Open_To_Buy	10127 non-null	float64
16	Total_Amt_Chng_Q4_Q1	10127 non-null	float64
17	Total_Trans_Amt	10127 non-null	int64
18	Total_Trans_Ct	10127 non-null	int64
19	Total_Ct_Chng_Q4_Q1	10127 non-null	float64
20	Avg_Utilization_Ratio	10127 non-null	float64
dtypes: float64(5), int64(10), object(6)			
memory usage: 1.6+ MB			

將類別型資料以「LabelEncoder」套件進行轉換，轉換後再查看資料集確認，資料都已轉成數值型資料了。

圖 5-2 「LabelEncoder」轉換程式碼

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()

c_data['Attrition_Flag'] = le.fit_transform(c_data['Attrition_Flag'])
c_data['Gender'] = le.fit_transform(c_data['Gender'])
c_data['Education_Level'] = le.fit_transform(c_data['Education_Level'])
c_data['Income_Category'] = le.fit_transform(c_data['Income_Category'])
c_data['Marital_Status'] = le.fit_transform(c_data['Marital_Status'])
c_data['Card_Category'] = le.fit_transform(c_data['Card_Category'])
```

圖 5-3 轉換後的資料集

c_data

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category	Months_on
0	768805383	1	45	1	3	3	1	2	0	
1	818770008	1	49	0	5	2	2	4	0	
2	713982108	1	51	1	3	2	1	3	0	
3	769911858	1	40	0	4	3	3	4	0	
4	709106358	1	40	1	3	5	1	2	0	
...	
10122	772366833	1	50	1	2	2	2	1	0	
10123	710638233	0	41	1	2	6	0	1	0	
10124	716506083	0	44	0	1	3	1	4	0	
10125	717406983	0	30	1	2	2	3	1	0	
10126	714337233	0	43	0	2	2	1	4	3	

10127 rows × 21 columns

2. 分割資料集

將資料集以 7:3 的比例分割為訓練資料集與測試資料集。

```
x, y = c_data[c_data.columns[2:]], c_data[c_data.columns[1]]
#分割資料集
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x, y,
                                                test_size=0.3,
                                                random_state=0,)
```

3. 處理失衡資料

由第三節分析目標變數時可得知，此資料集為一個失衡資料集(續約用戶:未續約用戶比 = 83.9% : 16.1%)，若是使用失衡資料集進行模型建置，將會使模型預測失真，因此導入「SMOTE (Synthetic Minority Oversampling Technique) 一樣本合成方法」套件，利用過取樣使資料達到平衡，而由下圖程式運行結果也可確認資料比例已平衡。

圖 5-4 SMOTE 套件

```
from imblearn.over_sampling import SMOTE

print("Before OverSampling, counts of label '1': {}".format(sum(y==1)))
print("Before OverSampling, counts of label '0': {} \n".format(sum(y==0)))

sm = SMOTE()
x_train, y_train = sm.fit_sample(x_train, y_train.ravel())

print("After OverSampling, counts of label '1': {}".format(sum(y_train==1)))
print("After OverSampling, counts of label '0': {}".format(sum(y_train==0)))

Before OverSampling, counts of label '1': 8500
Before OverSampling, counts of label '0': 1627

After OverSampling, counts of label '1': 5949
After OverSampling, counts of label '0': 5949
```


4. 資料標準化

資料平衡後，再進行標準化。

圖 5-5 標準化

```
#標準化
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
sc.fit(x_train)
X_train_std = sc.transform(x_train)
X_test_std = sc.transform(x_test)
```

5. 資料相關性

因為相關性有正相關與負相關，介於-1~1 之間，因此取絕對值查看各特徵變數與目標變數「Attrition_Flag(用戶是否續約)」之相關係數，並由大至小排序。我們可以發現「Total_Trans_Ct(過去一年總交易次數)」與目標變數的相關係數最高，第二高的為「Total_Ct_Chng_Q4_Q1 (第四季度與第一季度間交易數量變化)」，第三高的則是「Total_Revolving_Bal(循環信用總量)」。

圖 5-6 相關係數

```
#查看相關係數
most_correlated = c_data.corr().abs()['Attrition_Flag'].sort_values(ascending=False)
most_correlated = most_correlated[:15]
print(most_correlated)
```

Attrition_Flag	1.000000
Total_Trans_Ct	0.371403
Total_Ct_Chng_Q4_Q1	0.290054
Total_Revolving_Bal	0.263053
Contacts_Count_12_mon	0.204491
Avg_Utilization_Ratio	0.178410
Total_Trans_Amt	0.168598
Months_Inactive_12_mon	0.152449
Total_Relationship_Count	0.150005
Total_Amt_Chng_Q4_Q1	0.131063
CLIENTNUM	0.046430
Gender	0.037272
Credit_Limit	0.023873
Dependent_count	0.018991
Marital_Status	0.018597

Name: Attrition_Flag, dtype: float64

六、建置預測模型

1. 主成分分析

引入 PCA 套件並繪製圖 5-1，顯示主成分依序保留 1~20 特徵的疊加變異解釋率，並可了解此資料集之主成分降維至 10 時，能夠概括 80%的資料。

因此後續建立模型時，先以「n_components=10」降維，查看各模型的表現。(圖 5-2)

圖 6-1 主成分分析

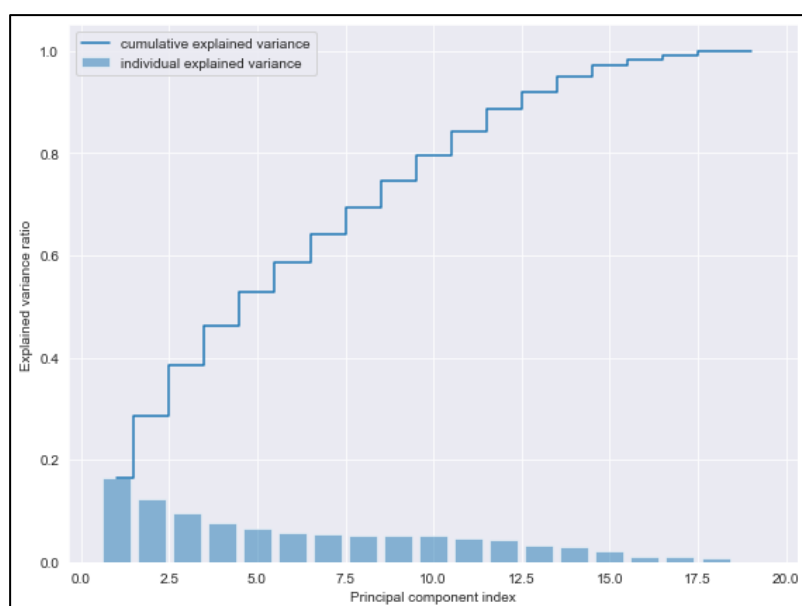


圖 6-2 PCA 套件降至 10 維

```
pca = PCA(n_components=10)
X_train_pca = pca.fit_transform(X_train_std)
X_test_pca = pca.transform(X_test_std)
```

2. 預測模型分析

由於本次目標變數為 2 元分類，採用著名的分類模型「Logistic Regression」、「RandomForest」和「XGBoost」。在主成分降至 10 維後可得到表 5-1 的結果。其中可以發現「XGBoost」的預測表現最佳，F1-score 為 93.2%，而「RandomForest」表現也不錯 F1-score 為 91.4%，雖然在訓練資料集方面 2 者皆已經達到 99% 以上的正確率，但在未知資料的預測分類上面可以得知是「XGBoost」所訓練的模型較佳。而「Logistic Regression」的預測表現之 F1-score 也達到 87.8%，然而訓練資料集之準確率竟低於測試資料集，亦可以得知能夠進行未知資料的預測分類。

表 6-1 預測模型準確率

模型名稱	訓練資料集	測試資料集	F1-score
Logistic Regression	79.9%	80.8%	87.8%
RandomForest(entropy)	99.5%	86%	91.4%
XGBoost	99.2%	88.7%	93.2%

接下來進一步查看重要特徵之權重後發現，兩者雖然特徵權重不同，卻都有極高的正確率。

圖 6-3 隨機森林重要特徵之權重

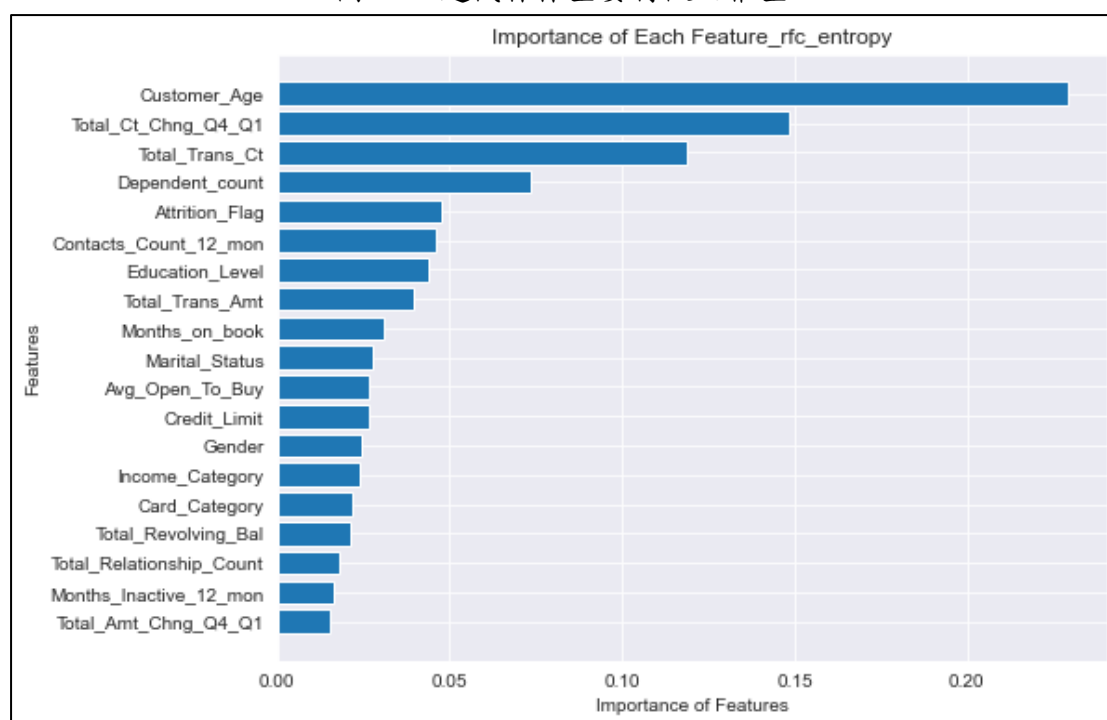
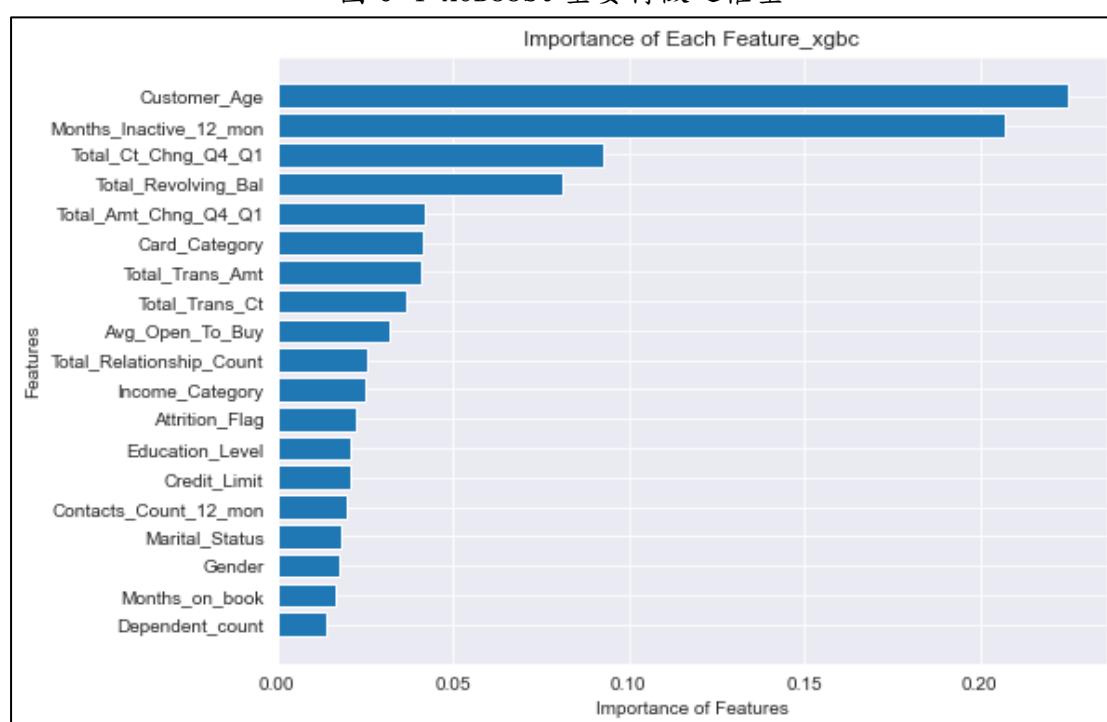


圖 6-4 XGBoost 重要特徵之權重

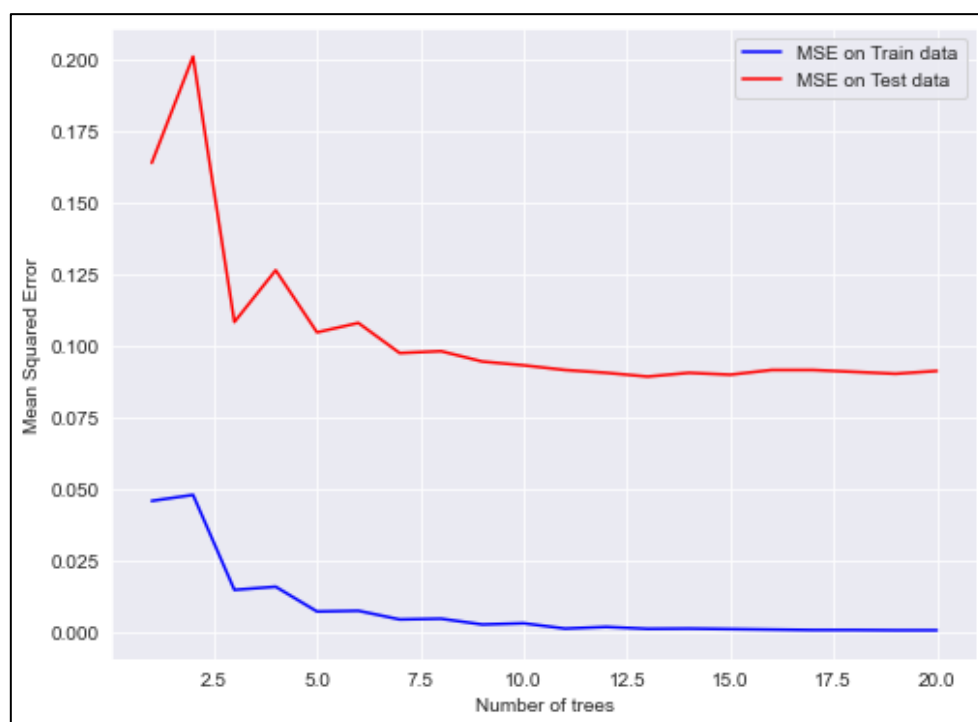


3. 「RandomForest」決策樹參數影響

由於「RandomForest」的參數相較於「XGBoost」較容易進行調整，，因此進一步查看不同決策樹數量對「RandomForest」模型在本次資料集的影響，並尋找最適合的決策樹數量參數。

由程式執行結果可得知，當「n_estimators=10」之後模型便趨於穩定。

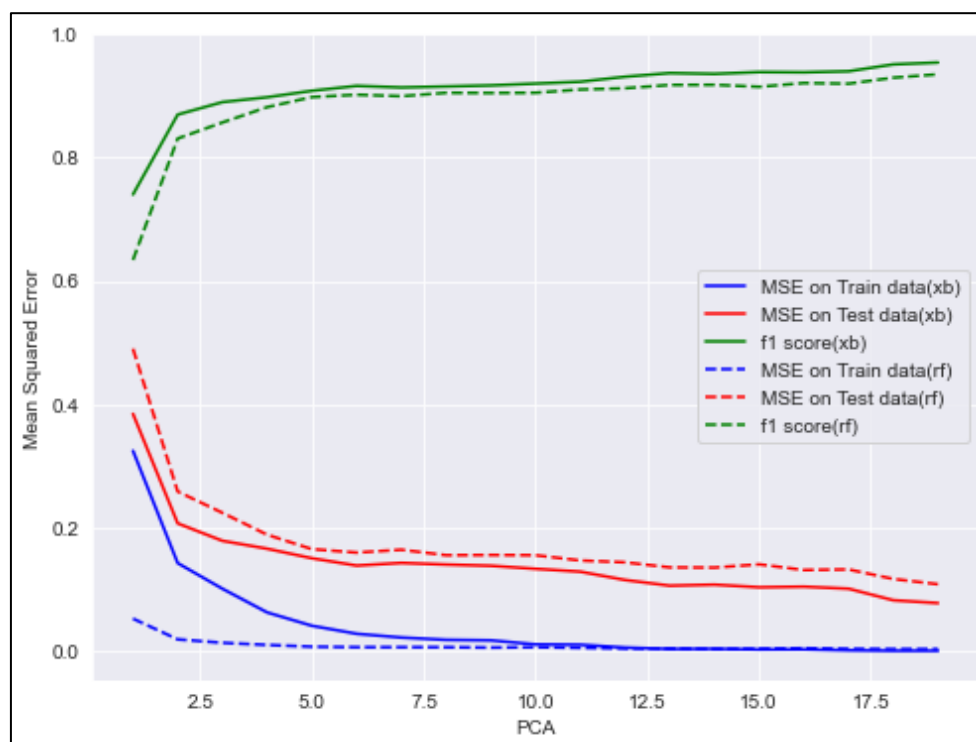
圖 6-5 「RandomForest」決策樹參數影響



4. 主成分疊代之 MSE 及 F1

最後比較在主成分疊代後，「RandomForest」和「XGBoost」的 MSE 和 F1-score 數值，皆是「XGBoost」模型擁有較好的表現，因此若是此資料集需要建置預測模型，可以使用「XGBoost」模型來進行分類預測。

圖 6-6 主成分疊代之兩模型的 MSE 及 F1-score



六、總結

本次期末報告以 Kaggle 網站上之公開資料集進行「信用卡用戶續約預測」，除了資料交叉分析與視覺化實作，亦建置預測分類模型並進行模型分析及評估，藉由實作經驗訓練程式能力，並活用平時所學的理论基礎，累積機器學習的經驗。

七、組員心得及分工

1. 日四 A 06122145 魯瑞慈—資料集說明、數據分析、交叉分析

期末報告的分析方式更多樣的情況下，這次主要著重在圖表的應用，也同時要去理解比較沒有接觸過的預測模型，在大量的資料和練習上有明顯發現自己對於程式邏輯的提升，機器學習真的是一個值得研究的類別，希望之後能夠汲取更多此方面的知識。

2. 日四 B 06122247 吳英緩—預測模型、程式碼校正、書面及簡報排版

我大四上學期的巨資專題報告是「網路數位型廣告的 CTR 分析與預測」，指導老師是李佳蓉老師，由於我是雙主修的學生，當時對機器學習的了解也是在專題製作與同時修習佳蓉老師的機器學習課程邊實做邊學習，因此在之後與機器學習相關的研究或報告中便能學以致用，並且更加熟悉和精進利用 Python 建置預測分類模型來進行分析和資料分類。

這次報告與大四上學期的專題相比，便更加深入分析和探討模型的主成分與重要特徵，對於資料分析師或資料工程師來說如何針對資料挑選適當的模型與進行模型參數調整是十分重要的，未來希望能夠更加提升模型適配與調參的能力。