

別讓精準行銷只是口號！行銷活動利潤模型與最佳決策點分析

當談到精準行銷時，第一個映入眼簾的通常是，根據用戶特點、興趣或以往的購買行為，精準的向「用戶」或「舊有客戶」推薦有興趣的商品或服務，進而讓其轉變成高顧客終身價值的「回頭客」或「新顧客」。在筆者過往的資料科學專案執行經驗中，需要達到好的精準行銷，筆者通常需協同合作企業從自身經營的品牌商城中蒐集、整理與分析顧客的個人資料，再導入機器學習演算法，以數據科學、資料驅動的角度，提供行銷人員、管理人員快速採行適宜的行銷策略方案。

現在，讓筆者來沉浸式體驗以下業界行銷人的情境，請讀者想象自己扮演的是一位電商零售行銷人 – Jasper，而Jasper是一位急於改善電商零售平台銷售的行銷管理者。

Jasper告訴筆者：「精準行銷已經老生常談了啦~！筆者簡單用FB、IG等網路行銷的方式就可以了啊，為什麼還要蒐集舊有的客戶資料？甚至還要將資料導出分析呢？」

相信多數人對此話深信不疑，但真的是如此嗎？

壹、行銷人們似乎只要用FB、IG等網路行銷來做行銷就好？

請讀者仔細回想一下自己行業會「持續」為您創造獲利的客戶是誰？這些客戶通常會以舊有客戶（舊客）為主，哈佛商業評論在《The Value of Keeping the Right Customers》(Gallo, 2014)這篇文章中提過：「開發一個新客的費用是留舊客費用的5到25倍」，這表示留住一個舊客遠比帶進一個新客更划算，如圖1所示。舉例而言，如果一家企業留住舊客所花費的錢是1萬新台幣，則開發新客並讓其成交的成本則會是5到25萬元新台幣。

- 開發一個新客的費用是留舊客費用的5–25倍
- 代表能「精準」的對舊客推薦商品更划算

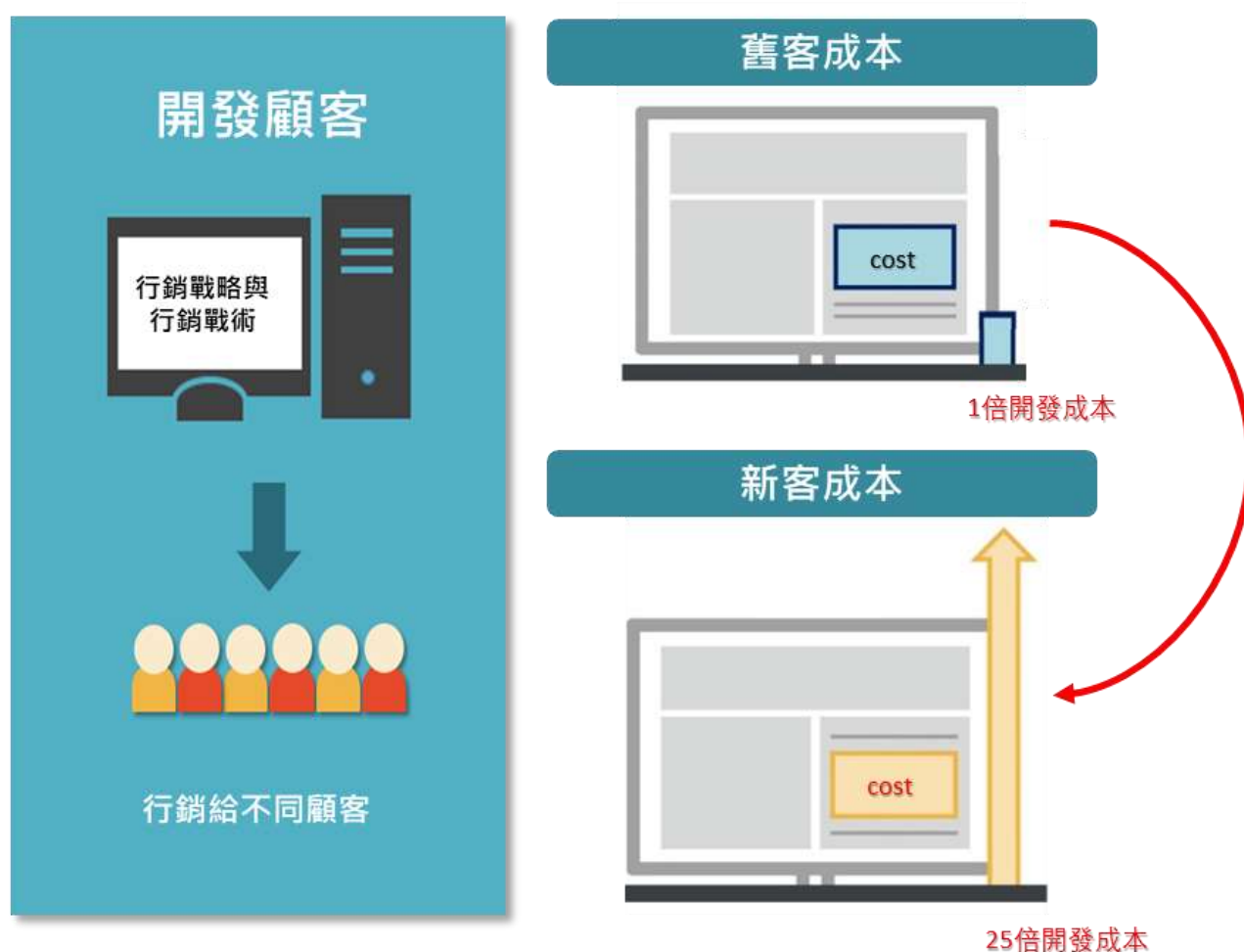


圖 1. 開發一個新客的費用是留舊客費用的 5 到 25 倍

故企業經營的重心應該聚焦在如何讓「舊客回流」。然而，Jasper過往的焦點幾乎都放在「如何以網路行銷的方式吸引新客，而不是想辦法創造舊客回購」。就算可以使用數位行銷方法重新再行銷粉絲頁用戶或已經購買過的客戶，仍不及善用自主擁有的CRM資料來精準投放給真正對產品有需求的顧客，更遑論消費者因為時代趨勢而轉換到其他新興社群平台而讓原社群平台數位行銷成效不佳的損失。

貳、顧客很少且商品不多時，需要蒐集資料嗎？

Jasper這時間：「好！看來客戶的CRM資料真的很重要！不過當筆者客戶很少且商品數量還不多的時候，現階段還不需要CRM資料的管理吧？而且外面CRM廠商的系統都很貴呢！」

筆者則認為，如今在數位轉型的浪潮下，各家企業已開始執行數位化之資料管理，如此競爭態勢下，若未能事前做好資料蒐集的準備，勢必在競爭態勢上會大落人後。反之，若企業經營有成，想要開始擴張，卻發現沒有完善的「舊客資料庫」，使原來可以賺上數十倍盈利的空間，即可能恍然成「空」。所以，企業應該從小就要為自身的客戶資料記錄上下功夫，以自己的主領域知識，一點一滴的建立自己的客戶資料庫。哪怕是以最簡單的Excel sheet做記錄，在這「資料即原油」的時代下，每一筆正確資料對企業都是無上至寶，以奠定後續精準行銷之基石，進而完善CRM資料。

參、萬事起頭難，精準行銷一開始如何實作？

Jasper有了資料蒐集與精準行銷的觀念後，筆者先請Jasper列出該公司的情境，以利各位讀者更好的理解Jasper要分析的情境個案：

公司規模：資本額千萬的中小企業

1. 產品：Jasper公司的A商品，性質係以B2C為主
2. 通路：以數位通路行銷為主，電話、email等自由媒體行銷為輔
3. 產品價格：\$ 2,500元新台幣
4. 產品成本：\$ 1215.5元新台幣
5. 行銷費用：行銷每一位顧客平均花費為\$185.32元新台幣

接著，Jasper興高采烈的跑去向高層主管提報自己習得的新想法，但緊接著又再次遇到了新一輪的問題。於是Jasper接著提問：「向高層主管提出筆者之前所討論的精準行銷想法後，我遇到如下五大問題，我現在該如何是好呢？」

1. 一開始所蒐集到的資料結構應該要如何形塑？
2. 如何用高層、中階主管、執行夥伴聽得懂的話，來實踐精準行銷？
3. 到底哪些才是高購買率的消費者？又如何建構「接地氣」的方案？
4. 如何自動評估精準行銷的財務效果、策略執行與落地方案？
5. 本案能否提高了行銷成效，減少客訴，甚至是開源賺錢與節流省錢

接下來，就讓筆者一步步解答上述Jasper的五大問題吧！

一、一開始所蒐集到的資料結構應該要如何形塑？

首先，第一個面對到的即是「客戶CRM資料要以什麼樣是形式創建？」

先從整理好的CRM資料，表1來回答本疑問，表中每一橫列（Row）代表一位顧客的行銷活動記錄，每一直行（Column）代表每一個變數的記錄。相信讀者可從表中看出一些端倪，這時候可以花幾分鐘想想，該資料表包含了什麼樣的元素？

表 1. 本情境案例之 CRM 模擬資料

| UID | 年齡 | 行業別 | 教育程度 | 是否以信用支付 | 有無逛過我們的網頁 | 聯絡有回應 |
|--------|----|-----|------|---------|-----------|-------|
| U11922 | 45 | 老闆 | 大學 | 否 | 是 | 否 |
| U2519 | 38 | 服務業 | 大學 | 否 | 是 | 否 |
| U28335 | 25 | 藍領 | 高中以下 | 否 | 是 | 否 |
| U9554 | 30 | 老闆 | 大學 | 否 | 是 | 否 |
| U34373 | 37 | 老闆 | 大學 | 否 | 是 | 否 |
| U36574 | 50 | 工程 | 研究所 | 否 | 是 | 否 |
| U22530 | 34 | 藍領 | 高中 | 否 | 是 | 否 |
| U16268 | 29 | 老闆 | 大學 | 否 | 是 | 是 |
| U37093 | 35 | 管理 | 研究所 | 否 | 是 | 是 |
| U34315 | 50 | 未知 | 高中 | 否 | 是 | 是 |
| U8284 | 54 | 家管 | 高中 | 否 | 是 | 否 |
| U23248 | 43 | 服務業 | 大學 | 否 | 是 | 是 |
| U15630 | 61 | 管理 | 研究所 | 否 | 否 | 否 |

| 日 | 月份 | 全部活動跟該客戶連絡次數 | 距離上次有多久時日沒被聯絡 | 前一次行銷活動的銷售結果 | 過往滿意程度 | 買A商品 |
|----|-----|--------------|---------------|--------------|--------|------|
| 13 | apr | 3 | 143 | 失敗 | 3 | 0 |
| 20 | apr | 4 | -1 | 未知 | 3 | 0 |
| 20 | apr | 1 | -1 | 未知 | 3 | 0 |
| 20 | apr | 1 | -1 | 未知 | 3 | 0 |
| 20 | apr | 1 | -1 | 未知 | 3 | 0 |
| 20 | apr | 2 | 340 | 失敗 | 3 | 0 |
| 20 | apr | 2 | 328 | 失敗 | 3 | 0 |
| 20 | apr | 1 | -1 | 未知 | 3 | 0 |
| 20 | apr | 7 | 272 | 其他 | 3 | 0 |
| 20 | apr | 4 | 340 | 成功 | 3 | 1 |
| 20 | apr | 1 | -1 | 未知 | 3 | 0 |
| 20 | apr | 1 | 152 | 失敗 | 3 | 0 |
| 20 | apr | 1 | -1 | 未知 | 3 | 1 |

細看表1，筆者蒐集資料要考量的要素可以歸納出三大項目：

1. 人口變數：每一位客戶購買本商品時，管理者會想要知道的顧客資訊，如表1的客戶年齡、行業別、教育程度等與客戶切身相關的變數。特別要注意的是，這些變數通常難透過後天的行為所改變，所以通常不可變動。
2. 行為變數：該變數通常會由該知識領域的專家進行制定，如表1呈現的全部活動跟該客戶連絡次數、過往滿意程度等可以直接或間接控制的變數。

3. 目標變數：即與公司KPI聯動或直接相關的重要變數，在資料科學中，通常就是被預測的變數，以利公司後續挑選正確的對象來執行精準行銷，如表1的是否「買A商品」之欄位，若「買A商品」之欄位顯示為1，則代表該橫列的顧客有「購買」；若為0，則代表該橫列的顧客「無購買」，若後續在公司顧客尚未購買A商品前即能預測其有高購買意向，即可先行對其做個人化精準行銷。

Jasper這時回問：「哦！那除了上述變數外，是否還有相關的變數可以建立呢？」

其實在客戶資料庫中，亦可包含諸多種類的變數，柯特勒（Philip Kotler）在《行銷管理》(Kotler & Keller, 2016)一書中所提，在進行市場分析時，區隔的變數包括：地理類、人口統計類、心理類、行為類，如圖2所示。若大型企業有實施會員制或平時已經有蒐集客戶個別CRM資料的習慣，不妨可以從圖2再次檢視是否有可增減的重點區隔變數。但以筆者數位轉型輔導的經驗觀察，若中小企業已能將與業務相關的「人口變數」、「行為變數」與「目標變數」共3種變數做好蒐集與建制工作，便已經有相當程度的水準可以進行個人化「精準行銷」分析了。



圖 2. 消費性市場之市場區隔變數（繪圖者:周晏汝）

Jasper繼續追問：「那有沒有什麼經濟實惠的蒐集方式呢？就是低成本的蒐集方法？」

對於大型企業來說，通常則會藉由SAP等相關系統來蒐集與統籌CRM資料；對於中小企業來說，其實要蒐集資料，最經濟實惠的方法即是善用「問卷與訂單」。在顧客購買產品的當下，予以誘因填寫問卷或會員資料，並針對欲蒐集的資料進行提問，蒐集其「人口變數」。再來資訊與行銷人員可共同協作，進而找出顧客過往消費足跡，來記錄客戶的「行為變數」，如：客戶連絡次數、與顧客聯絡方式等，逐步完善自己的客戶資料庫！

在彼此的協助下，Jasper此時已掌握足夠的資料蒐集訣竅，開始帶領團隊四處蒐集資料外，亦綜整了過往四散各地的顧客資料。最終皇天不負苦心人，Jasper統整了39,554位顧客對A商品的實際購買狀況的資料。請讀者特別注意，本案提供的所有資料皆為非真實之模擬資料。Jasper在過程中有蒐集其他不同商品的資料，也利後續商品的精準行銷策略。

蒐集完畢A商品資料後，Jasper再次提出疑問：「呼~！ 千辛萬苦蒐集好資料了！ 接下來該如何分析呢？」

筆者還沒有接話時，Jasper便磨刀霍霍向豬羊的繼續說：「非常期待接下來的如何變魔術，來分析手邊現有資料呢？！」

二、如何用高層、中階主管、執行夥伴聽得懂的話，來實踐精準行銷？

此時，筆者先請Jasper不用著急，筆者還要再請Jasper確認高層、中階主管、執行夥伴從本次專案想要取得的效益是什麼，以再次對齊各層目標，有利後續資料科學專案的實行。

Jasper接著問：「嗯... 之前有確認過總經理的需求啊！就是要找出會買A商品的顧客有誰，又要再次確認，那我究竟還要跟誰確認什麼呢？」

會需要再次確認各層夥伴的目的為，隨著資料蒐集的時間演變，除了確認各層夥伴是否在本案上有其他需求外，還可以根據《商業人工智慧 (Artificial Intelligence for Business)》一書中，所提到的四層「資料金字塔 (Data Pyramid)」(Akerkar, 2019)，如圖3之概念來與第一線人員、中階管理者、高階管理者與策略制定者相互

溝通，對齊各自目標、需求與可交付成果，才不會讓本專案曲高和寡，空有總經理的支持而制度與執行層面無法實際勾稽之狀況。

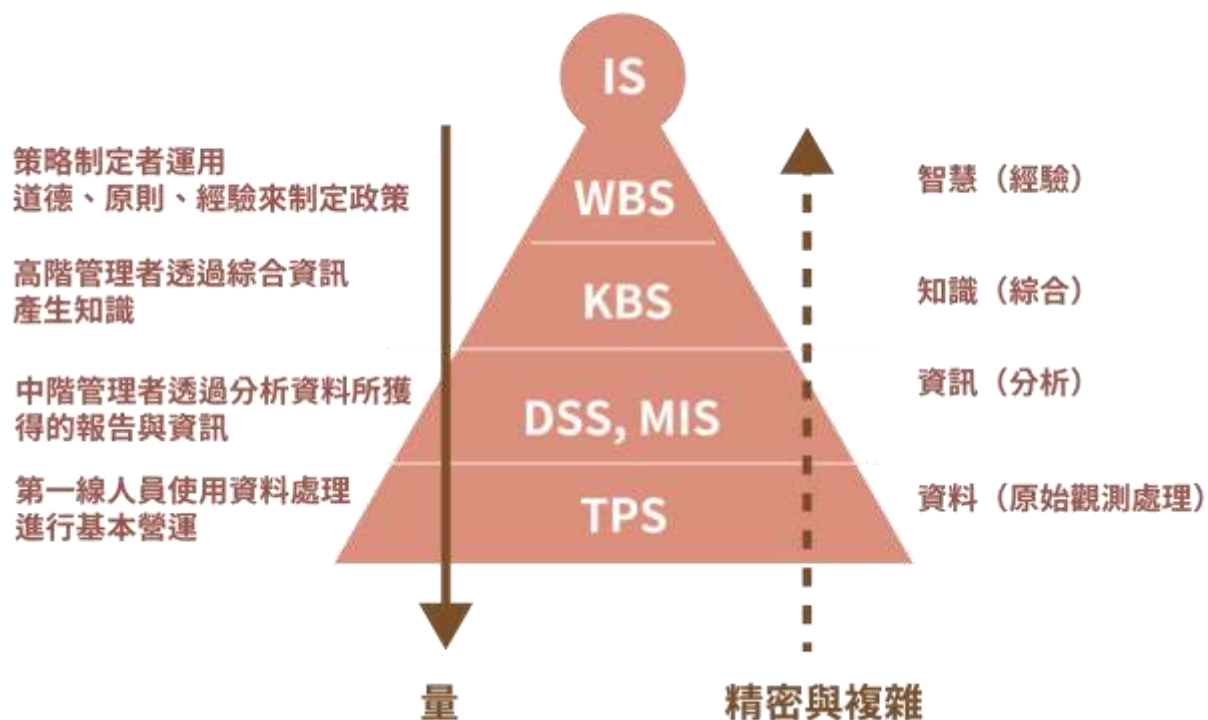


圖 3. 資料金字塔 (繪圖者：黃亭維)

最後，經過Jasper團隊的調查，筆者將資料金字塔的概念描繪出圖4之管理金字塔目標確認圖，並簡要分成概念化能力、人際能力與執行能力階層所需要之目標。

概念化能力部分，代表者為策略制定者與高階管理者，對預測高意願購買A商品的顧客來說，期望能看到開源節流的成效外，亦能善用本專案做中學，培養公司培養資料分析人才，讓公司數位轉型的人才更有價值，同時優化制度，以應對數位轉型時代的到來。

人際能力部分，代表者為中間管理者，其希望能請筆者團隊技轉本案到公司資料分析團隊中，讓未來類似專案一樣能有機會邁向成功。

執行能力部分，代表者為第一線人員，其希望本案能讓他們有機會快、狠、准完成上級交付的任務，達到人機輔助之成效。



圖 4.管理金字塔目標確認圖

Jasper這時候跳出來說：「太好了！看來不管是資料蒐集、各階層目標對齊等工作終於完成了！接下來總可以開始分析了吧！」

三、到底哪些才是高購買率的消費者？又如何建構「接地氣」的方案？

沒錯！完成了資料蒐集及組織溝通的任務後，就可開始資料分析的任務啦！從本節開始，讀者可從下述QR code或從下述github repository鏈接中找到ai_marketing資料夾，將ai_marketing複製（clone）下來，一起隨筆者實作程式碼。

https://github.com/HowardNTUST/Marketing-Data-Science-GenAI-AI-Applicantion/tree/main/ai_marketing

在ai_marketing資料集中會發現有5個檔案，如圖5所示：

1. Results：包含了本章節的所有產出，有興趣的讀者可以先以裡面的檔案配合文章一同查看。

2. ai_marketing.py：為主要的Python執行檔案，請讀者可以直接打開這個檔案跟筆者一起操作與執行。
3. ai_marketing_lib.py：為本案主要自製的Python套件檔案（library）。
4. marketing_ec_data_train.csv：為Jasper蒐集到的39,554筆原始顧客的CRM模擬資料。
5. marketing_ec_data_testing.csv：為Jasper主要用來測試精準行銷準確度用的CRM模擬資料，其含有5,657筆資料。

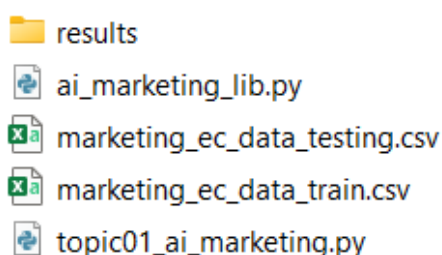


圖 5. ai_marketing 資料集包含的內容

有了上述資料後，首要解決的問題為找出「高購買率的消費者」。請讀者先開啟ai_marketing.py，並讀入程式碼1，並產出其資料形態Excel表格，如程式碼1的產出所示。

程式碼1

```
# %%  
# SECTION - 程式碼1  
from sklearn.metrics import accuracy_score, precision_score, recall_score,  
f1_score, roc_auc_score, roc_curve, auc, confusion_matrix, r2_score  
from ai_marketing_lib import model_profit_fun, profit_linechart  
import matplotlib.pyplot as plt  
from ai_marketing_lib import plot_confusion_matrix  
import plotly.express as px  
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot  
import pandas as pd  
from xgboost import XGBClassifier  
from sklearn.model_selection import train_test_split  
import numpy as np  
from ai_marketing_lib import detect_str_columns, transform_to_category
```

```
# 讀取marketing 資料
data = pd.read_csv('marketing_ec_data_train.csv')
data.info()

# 輸出data.info() --> 將data.dtypes 與 data.count() 合併
data_type = pd.concat(
    [pd.DataFrame(data.dtypes), data.count()], axis=1).reset_index()
data_type.columns = ['欄位名稱', '資料型態', '非空值的資料筆數']
data_type.to_excel('01_資料形態.xlsx', index=False)
#!SECTION - 程式碼1
```

程式碼1的產出

01_資料形態.xlsx

| 欄位名稱 | 資料型態 | 非空值的資料筆數 |
|---------------|--------|----------|
| UID | object | 39554 |
| 年齡 | int64 | 39554 |
| 行業別 | object | 39554 |
| 教育程度 | object | 39554 |
| 是否以信用支付 | object | 39554 |
| 有無逛過我們的網頁 | object | 39554 |
| 聯絡有回應 | object | 39554 |
| 日 | int64 | 39554 |
| 月份 | object | 39554 |
| 全部活動跟該客戶連絡次數 | int64 | 39554 |
| 距離上次有多久時日沒被聯絡 | int64 | 39554 |
| 前一次行銷活動的銷售結果 | object | 39554 |
| 過往滿意程度 | int64 | 39554 |
| 買A商品 | int64 | 39554 |

此時，可以依照需求在Excel表格中多新增一欄「說明」，讓所有為本專案執行資料科學任務的夥伴更加清楚其相關含義，如表2所示。

表 2. 本案資料形態與說明

| 欄位名稱 | 資料型態 | 非空值的資料筆數 | 說明 |
|---------------|--------|----------|---|
| UID | object | 39554 | 顧客獨一無二的ID |
| 年齡 | int64 | 39554 | 顧客的年齡 |
| 行業別 | object | 39554 | 顧客的行業別，含有共12種行業別：SOHO、工程、失業、未知、老闆、服務業、科技、家管、退休、管理、學生、藍領 |
| 教育程度 | object | 39554 | 顧客的教育程度，含有共4種教育程度：大學、研究所、高中、高中以下 |
| 是否以信用支付 | object | 39554 | 顧客過往是否有用信用卡在我們公司付款過 |
| 有無逛過我們的網頁 | object | 39554 | 顧客是否有逛過我們的網頁 |
| 聯絡有回應 | object | 39554 | 公司過往聯絡聯絡顧客時，是否有回應 |
| 日 | int64 | 39554 | 顧客被建檔時間，本處是日期中的「日」 |
| 月份 | object | 39554 | 顧客被建檔時間，本處是日期中的「月份」 |
| 全部活動跟該客戶連絡次數 | int64 | 39554 | 過往跟顧客聯絡次數 |
| 距離上次有多久時日沒被聯絡 | int64 | 39554 | 距離與顧客上次有多久時日沒被聯絡，其中-1為「未知」，這意味過往並未很好的記錄其之前的聯絡日期 |
| 前一次行銷活動的銷售結果 | object | 39554 | 前一次對顧客行銷活動的銷售結果，未知代表沒有記錄到；其他代表有其他相關之業務拓展，如對其他公司活動有興趣參與等 |
| 買A商品 | int64 | 39554 | 為本次最重要的目標變數，即顧客是否有購買本次推薦的A商品 |
| 過往滿意程度 | int64 | 39554 | 付費或免費使用我們產品平台的時候，內部會跳出總體滿意程度讓顧客選擇，分數為1到5分，分數越高，越滿意 |

為了要讓資料能適當的放入機器學習演算法，筆者會適時的將字串欄位或稱字串變數，諸如：行業別、教育程度等表2資料形態為object的欄位轉換成數值形態（numerical data type）或類別形態（categorical data type），以利機器學習模型更好利用這些字串欄位。

本次教學主要將字串欄位轉換成類別形態，如程式碼2所示。接著，筆者再次查看程式碼2的產出成果，除了每位顧客獨有的UID外，其他字串變數均已轉換成類別變數，如此即可執行後續機器學習的分析。

| |
|---|
| <p>程式碼2</p> <pre> # %% # SECTION - 程式碼2 # 偵測有字串的欄位 str_columns = detect_str_columns(data.drop(columns='UID')) dataset = transform_to_category(str_columns, data) # 確認全部都是數字 float, int, uint --> ML dataset.info() # 輸出【轉換後】data.info() --> 將data.dtypes 與 data.count() 合併 data_type = pd.concat([pd.DataFrame(dataset.dtypes), dataset.count()], axis=1).reset_index() data_type.columns = ['欄位名稱', '資料型態', '非空值的資料筆數'] </pre> |
|---|

```
data_type.to_excel('02_【轉換後】資料形態.xlsx', index=False)
#!SECTION - 程式碼2
```

程式碼2的產出

【轉換後】資料形態.xlsx

| 欄位名稱 | 資料型態 | 非空值的資料筆數 |
|---------------|----------|----------|
| UID | object | 39554 |
| 年齡 | int64 | 39554 |
| 行業別 | category | 39554 |
| 教育程度 | category | 39554 |
| 是否以信用支付 | category | 39554 |
| 有無逛過我們的網頁 | category | 39554 |
| 聯絡有回應 | category | 39554 |
| 日 | int64 | 39554 |
| 月份 | category | 39554 |
| 全部活動跟該客戶連絡次數 | int64 | 39554 |
| 距離上次有多久時日沒被聯絡 | int64 | 39554 |
| 前一次行銷活動的銷售結果 | category | 39554 |
| 過往滿意程度 | int64 | 39554 |
| 買A商品 | int64 | 39554 |

接下來，請讀者也將測試資料集的檔案同程式碼3的操作一般，也讀取並轉換其資料集。

程式碼3

```
# %%
# SECTION - 程式碼3
# 將測試資料也同樣讀取進來並執行資料轉換
# 讀取 marketing 資料
data_test = pd.read_csv('marketing_ec_data_testing.csv')
data_test.info()

# 偵測有字串的欄位
str_columns = detect_str_columns(data_test.drop(columns='UID'))
dataset_test = transform_to_category(str_columns, data_test)

# 確認全部都是數字 float, int, uint --> ML
dataset_test.info()
```

#!SECTION - 程式碼3

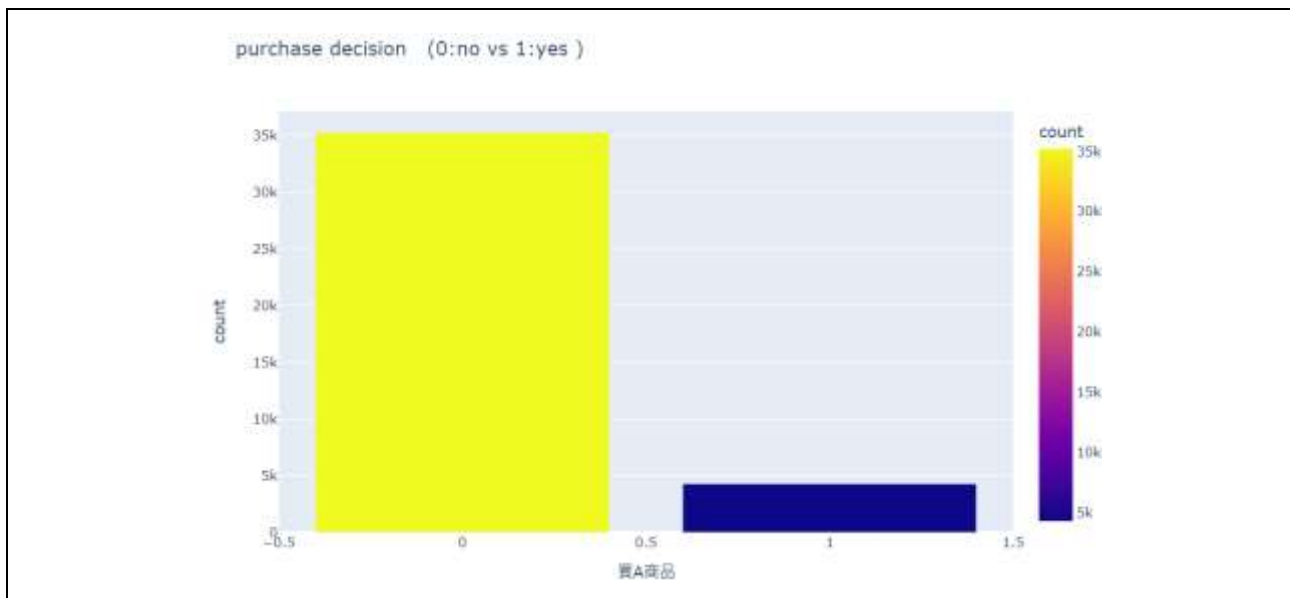
再來可以探查Jasper團隊行銷了39,554位顧客後，買單的比例究竟有多少。透過程式碼4，可在「買A商品」欄位中統計「購買」與「不購買」的佔比分別為多少，從程式碼4的產出結果可知僅有10%的顧客真正會買單A商品，讀者可以試著思考，行銷人需要行銷10位顧客，才有機會換得1位顧客的購買，簡言之，行銷人要判讀這10%真正會消費A商品的顧客是多麼的難，不過在後續卻可透過機器輔助的方式來協助人類判別高意願購買的顧客。

程式碼4

```
# %%  
# SECTION - 程式碼4  
# 繪圖：查看y變數的分佈圖  
# Buy 比例  
print('行銷顧客後卻不購買的比例', round(dataset['買A商品'].value_counts()  
      [0]/len(data) * 100, 2), '%')  
print('行銷顧客成功而購買的比例', round(dataset['買A商品'].value_counts()[  
      1]/len(data) * 100, 2), '%')  
  
# 看看 y 變數的分佈圖  
dataset['count'] = 1  
data_count = dataset.groupby('買A商品', as_index=False)['count'].sum()  
fig = px.bar(data_count, x="買A商品", y="count",  
             color='count',  
             title='purchase decision \n (0:no vs 1:yes )',  
             )  
  
plot(fig, filename='purchase_decision.html', auto_open=False)  
#!SECTION - 程式碼4
```

程式碼4的產出

```
行銷顧客後卻不購買的比例 89.17 %  
行銷顧客成功而購買的比例 10.83 %
```



在進入機器學習模型訓練前的資料前處理部分，筆者必須要先將目標變數「買A商品」或稱y變數與其他特徵變數或稱x變數分割開來，如程式碼5所示，以便筆者後續機器學習時，可使用x變數來對y變數執行訓練，找出高意願購買A商品的顧客。

從程式碼5產出成果來看，得知之前在程式碼4所使用到的count過渡變數不應該要考量進本資料集中，所以即在程式碼5中刪除count變數，由此可知在資料處理完的每一步都建議讀者可以仔細檢閱是否有需要清理的資料。

程式碼5

```
# %%  
# SECTION - 程式碼5  
# 區分訓練資料集的X與y  
X_train = dataset.drop(columns=['買A商品'])  
y_train = dataset['買A商品']  
  
X_test = dataset_test.drop(columns=['買A商品'])  
y_test = dataset_test['買A商品']  
  
# 保留UID  
train_uid = X_train['UID']  
test_uid = X_test['UID']  
  
# 刪除UID  
del X_train['UID']  
del X_test['UID']
```

```

# 再次檢查X_train 與X_test 的資料型態
data_train_type = pd.concat(
    [pd.DataFrame(X_train.dtypes), X_train.count()], axis=1).reset_index()
data_train_type.columns = ['欄位名稱', '資料型態', '非空值的資料筆數']
data_train_type.to_excel('03_【訓練資料集_特徵變數】資料形態.xlsx', index=False)

data_test_type = pd.concat(
    [pd.DataFrame(X_test.dtypes), X_test.count()], axis=1).reset_index()
data_test_type.columns = ['欄位名稱', '資料型態', '非空值的資料筆數']
data_test_type.to_excel('04_【測試資料集_特徵變數】資料形態.xlsx', index=False)

# 刪除不需要的變數
if X_train.filter(regex='count').shape[1] > 0:
    X_train.drop(columns=X_train.filter(regex='count'), inplace=True)

print(X_train.columns)

# !SECTION - 程式碼5

```

程式碼5的產出

03_【訓練資料集_特徵變數】資料形態.xlsx

| 欄位名稱 | 資料型態 | 非空值的資料筆數 |
|---------------|----------|----------|
| 年齡 | int64 | 39554 |
| 行業別 | category | 39554 |
| 教育程度 | category | 39554 |
| 是否以信用支付 | category | 39554 |
| 有無逛過我們的網頁 | category | 39554 |
| 聯絡有回應 | category | 39554 |
| 日 | int64 | 39554 |
| 月份 | category | 39554 |
| 全部活動跟該客戶連絡次數 | int64 | 39554 |
| 距離上次有多久時日沒被聯絡 | int64 | 39554 |
| 前一次行銷活動的銷售結果 | category | 39554 |
| 過往滿意程度 | int64 | 39554 |
| count | int64 | 39554 |

04_【測試資料集_特徵變數】資料形態.xlsx

| 欄位名稱 | 資料型態 | 非空值的資料筆數 |
|---------------|----------|----------|
| 年齡 | int64 | 5657 |
| 行業別 | category | 5657 |
| 教育程度 | category | 5657 |
| 是否以信用支付 | category | 5657 |
| 有無逛過我們的網頁 | category | 5657 |
| 聯絡有回應 | category | 5657 |
| 日 | int64 | 5657 |
| 月份 | category | 5657 |
| 全部活動跟該客戶連絡次數 | int64 | 5657 |
| 距離上次有多久時日沒被聯絡 | int64 | 5657 |
| 前一次行銷活動的銷售結果 | category | 5657 |
| 過往滿意程度 | int64 | 5657 |

為了讓讀者更加瞭解機器學習模型與資料的關係，在此佐以圖6來解釋其關係。讀者可見除了「買A商品」為目標變數外，其餘變數均代表為可能會造成顧客購買A商品的「特徵變數」，故機器學習即是要從茫茫的特徵變數大海中找尋「會影響目標變數的模式」，以更好的精準預測未來會向Jasper公司購買A商品的顧客。



圖 6. 機器學習模型與本案資料的關係

最後，善用程式碼6導入商務分析上經典的XGBoost機器學習模型，輕輕鬆鬆完成程式碼6的產出成果，即顧客精準行銷清單！

程式碼6

```
# %%  
# SECTION - 程式碼6  
# 訓練機器學習模型  
  
# 命名模型物件  
clf = XGBClassifier(n_estimators=500,  
                    tree_method="hist",  
                    enable_categorical=True,  
                    random_state=0, nthread=8,  
                    learning_rate=0.05,  
                    )  
  
# 進行訓練  
model_xgb = clf.fit(X_train, y_train, verbose=True,  
                   eval_metric='aucpr', eval_set=[(X_test, y_test)])  
  
# 預測測試資料集的顧客購買與否  
y_pred = model_xgb.predict(X_test)  
  
# 預測測試資料集的顧客購買機率  
y_pred_prob = model_xgb.predict_proba(X_test)[:, 1]  
  
# 建構精準顧客名單  
consumer_acc_list = pd.DataFrame(y_test.values, columns=['顧客對 A 商品【實際】購買  
狀態'])  
consumer_acc_list['顧客對 A 商品【預測】購買機率'] = y_pred_prob  
  
# 6. 將UID 加回去顧客精準行銷名單  
test_uid = test_uid.reset_index().drop(columns=['index'])  
consumer_acc_list = pd.concat([test_uid, consumer_acc_list], axis=1)  
  
# 7. 將精準顧客名單購買機率由大到小排序  
consumer_acc_list = consumer_acc_list.sort_values(  
    by='顧客對 A 商品【預測】購買機率', ascending=False)
```

```

consumer_acc_list = consumer_acc_list[[
    'UID', '顧客對A商品【預測】購買機率', '顧客對A商品【實際】購買狀態',]]

consumer_acc_list['顧客對A商品【預測】購買機率'] = round(
    consumer_acc_list['顧客對A商品【預測】購買機率'], 3)

# 8. 將精準顧客名單存成 excel 檔
consumer_acc_list.to_excel('05_顧客精準行銷清單.xlsx', index=False)

# !SECTION - 程式碼6

```

程式碼6的產出

05_顧客精準行銷清單.xlsx

| UID | 顧客對A商品【預測】購買機率 | 顧客對A商品【實際】購買狀態 |
|--------|----------------|----------------|
| U16298 | 0.975 | 0 |
| U34709 | 0.972 | 0 |
| U24276 | 0.969 | 1 |
| U9737 | 0.969 | 0 |
| U41521 | 0.966 | 1 |
| U33207 | 0.966 | 0 |
| U26803 | 0.960 | 1 |
| U21483 | 0.955 | 1 |
| U20374 | 0.955 | 1 |
| U4247 | 0.954 | 0 |
| U18893 | 0.952 | 0 |
| U11192 | 0.952 | 1 |
| U2062 | 0.951 | 1 |
| U18110 | 0.951 | 1 |
| U12726 | 0.950 | 1 |
| U40123 | 0.949 | 1 |
| U35736 | 0.945 | 1 |

從顧客精準行銷清單裡，首先可從測試資料集的5,657名顧客中，得知每一位顧客對A商品【預測】購買機率，進而請第一線行銷人員對高意願購買A商品的顧客先行銷售A商品，以求用最低的行銷費用，達到精準行銷之成效，舉例而言：在程式碼6的產出清單中，行銷人員可對U16298、U34709與U24276三位顧客先行行銷，因為三位對A商品的購買機率分別為97.5%、96.2%與96.9%，當行銷人員完成對三者完成行銷後，會將最後顧客「最終」是否成交A商品的狀況記錄在「顧客對A商品

【實際】購買狀態」的欄位中，從表中可見在人機輔助的努力下，其中一者最後果然成交。

Jasper這時問：「經過這一輪的努力後，看起來不一定高機率的顧客最後都有成交耶？」

模型產出的購買機率即是「顧客購買的不確定性」，所以哪怕U16298這位顧客有接近97.5%的購買機率，仍有2.5%的可能性不會購買A商品。此外，模型的表現也是在精準行銷預測上很重要的考量因素。

Jasper繼續追問：「原來如此！但是我還有幾個問題，雖然我現在有看似很強大的精準行銷武器（模型）了，但是我又該如何評估您所說的模型表現，同時讓主管買單呢？」

四、如何自動評估精準行銷帶來的潛在財務效果？同時讓主管們也都聽得懂？

筆者將分別從資料科學與業界的角度來闡述「模型表現」。先從資料科學的角度來看，透過程式碼7的評估，筆者很容易就可以得知本模型在準確度上表現達近83%，即有83%的比例模型成功預測不購買與購買的顧客。

Jasper回應：「哦哦！這樣的命中率看起來不錯哦！但是請問有沒有更直觀一點的表示方式呢？怕老闆與同仁他們還是聽不懂啊！」

| |
|--|
| 程式碼7 |
| <pre># %% # SECTION - 程式碼7 # 資料科學角度評估模型表現 # 準確度 print(accuracy_score(y_test, y_pred)) # !SECTION - 程式碼7</pre> |
| 程式碼6的產出 |
| 0.8278239349478522 |

還記得筆者之前在程式碼4有算過訓練資料集中，行銷所有顧客後，大約僅有10%的顧客會購買筆者的A商品嗎？這次一樣計算一下行銷測試資料集所有人後，會有多少人有意願購買A商品，從程式碼8的產出得知為17.78%。

程式碼8

```
# %%  
# SECTION - 程式碼8  
# 混淆矩陣評估模型表現  
  
# 首先評估測試資料集真正購買A商品顧客佔所有資料的比例  
marketing_all_consumer = round(y_test.value_counts()[1]/len(y_test), 4)  
print('真正購買A商品顧客佔所有資料的比例', marketing_all_consumer * 100, '%')  
# !SECTION - 程式碼8
```

程式碼8的產出

真正購買A商品顧客佔所有資料的比例 17.78 %

筆者採用了機器學習模型後，即對有高機率購買者行銷，相較原先海撒全部顧客行銷方法來說，模型行銷成功的比例可以達到52.27%，增加了近2.94倍的提升成效（lift），如程式碼9的產出所示。

這代表從資料科學的角度來看，本機器學習模型確實有提升原有行銷方法的成效，並且提高近3倍的成效。

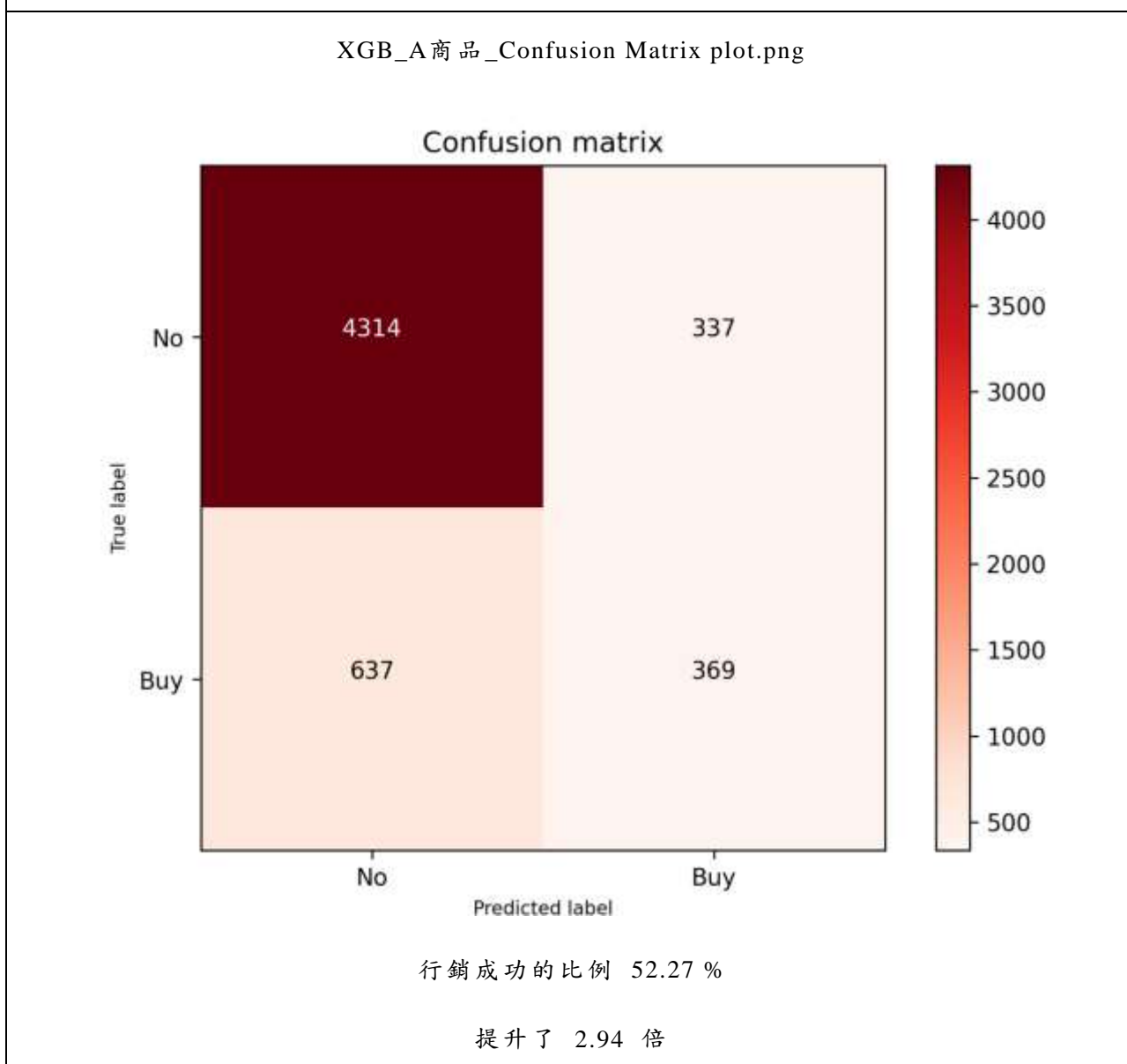
程式碼9

```
# %%  
# SECTION - 程式碼9  
# 先做出混淆矩陣  
  
model_conf = confusion_matrix(y_test, y_pred)  
plot_confusion_matrix(model_conf,  
                        classes=['No', 'Buy'],  
                        title='Confusion matrix',  
                        cmap=plt.cm.Reds  
                        )  
  
# 行銷成功的比例可以達到多少呢？
```

```
ps = precision_score(y_test, y_pred)
print('行銷成功的比例', round(ps, 4) * 100, '%')

# 相較原先海撒全部顧客行銷方法來說，用了模型之後，提升了多少成交比呢？
lift = round(ps / marketing_all_consumer, 2)
print('提升了', lift, '倍')
# !SECTION - 程式碼9
```

程式碼9的產出



至於模型行銷成功的比例如何計算呢？在程式碼9的產出成果中，有一張名為 confusion matrix的圖，橫軸的predicted label分為預測會購買的顧客（Buy）與不會購買的顧客（No），圖中顯示模型預判共706人（369+337）會至少有50%的購買意

願（購買機率）來購買A商品，故請行銷人員對其706人開始行銷。最後在706人中，僅有369人真正買單，其比例為52.27%，意味每行銷2人就有1人會買單，這相較原本僅17.78%，即行銷6人才約1人買單來說，高出近2.94倍，而這也是資料科學界與零售界中鼎鼎大名的提升度（Lift）(Brin et al., 1997)算法。

Jasper繼續回饋：「看起來我知道使用機器學習來做精準行銷的方法比原本的方法還要好約3倍！但有沒有更能讓老闆與同仁都更了解的方法呢？」

接著就來到業界的角度來評估模型表現的環節了！讀者還記得筆者在前面有提及「分析的情境個案」資訊嗎？其中包含了A商品的財務架構，諸如：價格、成本與行銷費用。使用程式碼10即可將本次專案的A商品財務架構考量進機器學習模型中並產出「行銷全部顧客的利潤結果」（XGB全市場行銷利潤矩陣.csv）與「善用模型執行精準行銷的利潤結果」（XGB模型行銷利潤矩陣.csv）。

程式碼10

```
# %%
# SECTION - 程式碼10
# 行銷活動利潤評估

XGB_all_df, XGB_model_profit_df, XGB_y_test_df = model_profit_fun(
    clf=XGBClassifier(n_estimators=500,
                      tree_method="hist",
                      enable_categorical=True,
                      random_state=0, nthread=8,
                      learning_rate=0.05,
                      ), # sklearn 的模型

    X_train=X_train, # 訓練資料集 X : training set (x)
    y_train=y_train, # 訓練資料集 Y : training set (buy)
    X_test=X_test, # 測試資料集 X : testing set (x)
    y_test=y_test, # 測試資料集 Y : testing set (buy)
    test_uid = test_uid, # 測試資料集 UID
    sales_price=2500, # 價格
    marketing_expense=185.32, # 行銷費用或銷貨成本
    product_cost=1215.2, # 產品成本
    plot_name='XGB_A 商品_') # 產出結果的名稱，會存到資料夾

# !SECTION - 程式碼10
```

程式碼10的產出

```

-----模型相對全市場行銷來說【賺錢】-----
模型比全市場行銷賺 $99101.71999999991
-----全市場行銷利潤矩陣-----
      項目      金額  目標對象      小計
0  單品價格    2500  1006  2515000.00
1  單品營業成本 1215.2  1006  1222491.20
2  單品行銷費用 185.32  5657  1048355.24
3      利潤      -      -  244153.56
全市場行銷利潤矩陣.csv saved
-----XGB模型行銷利潤矩陣-----
      項目      金額  目標對象      小計
0  單品價格    2500  369  922500.00
1  單品營業成本 1215.2  369  448408.80
2  單品行銷費用 185.32  706  130835.92
3      利潤      -      -  343255.28
模型行銷利潤矩陣.csv saved

```

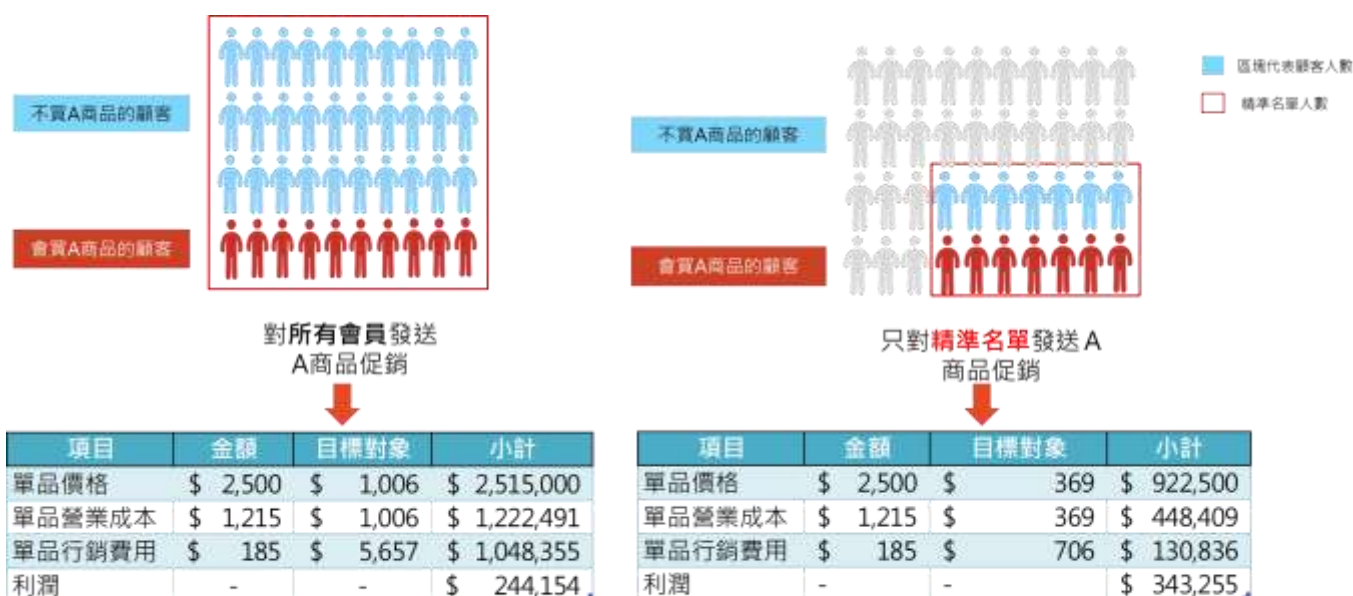
XGB全市場行銷利潤矩陣.csv

| 項目 | 金額 | 目標對象 | 小計 |
|--------|----------|----------|--------------|
| 單品價格 | \$ 2,500 | \$ 1,006 | \$ 2,515,000 |
| 單品營業成本 | \$ 1,215 | \$ 1,006 | \$ 1,222,491 |
| 單品行銷費用 | \$ 185 | \$ 5,657 | \$ 1,048,355 |
| 利潤 | - | - | \$ 244,154 |

XGB模型行銷利潤矩陣.csv

| 項目 | 金額 | 目標對象 | 小計 |
|--------|----------|--------|------------|
| 單品價格 | \$ 2,500 | \$ 369 | \$ 922,500 |
| 單品營業成本 | \$ 1,215 | \$ 369 | \$ 448,409 |
| 單品行銷費用 | \$ 185 | \$ 706 | \$ 130,836 |
| 利潤 | - | - | \$ 343,255 |

從圖7中，筆者可以更清楚知曉在執行5657人的行銷活動下，左圖行銷全顧客的營收達到250萬左右，但其行銷費用卻來到了100萬，雖然廣告投報率（ROAS）達到2.5倍左右，即投入約100萬的行銷費用，創造約250萬的營收，但是總體的利潤率僅有約10%；反觀右圖以模型執行精準行銷的營收雖為92萬左右，但其行銷費用僅約13萬，其ROAS可達到7左右，且總體的利潤率竟有37%。這代表精準行銷方法不但能降低人員行銷費用，同時增加利潤，達到開源賺錢與節流省錢之效果。



用更低的人員行銷費用達到轉換效果。

圖 7. 行銷全顧客（左圖與表）與精準行銷特定顧客（右圖與表）的財務比較表

Jasper接著回饋：「哇！太棒了！這樣才對！老闆最愛錢了！這樣說明老闆才聽得懂筆者做了些什麼嘛！」

Jasper立刻又問到：「這706人挑選出來是很好，不過我怎麼記得您是根據至少有50%的購買A商品意願（機率）的顧客來挑選的？現在既然已經有A商品的單一財務架構，有沒有機會找到一個特定的顧客購買機率，讓利潤最大化呢？」

此時筆者可以根據程式碼6的「顧客精準行銷清單」來比對在哪一顧客對A商品預測購買機率下，實際成交的利潤額最高。接著透過程式碼11窮舉所有利潤最大化的可能性。最後得知在本案中，凡顧客的預測購買機率大於或等於19%，則建議行銷人員要對該顧客行銷，則可獲\$549,314之利潤成效，相較圖7中的右圖僅向50%以上購買機率的顧客行銷來說，其利潤高出了\$206,059（\$549,314-\$343,255），即1.6倍，如程式碼11的產出所示。

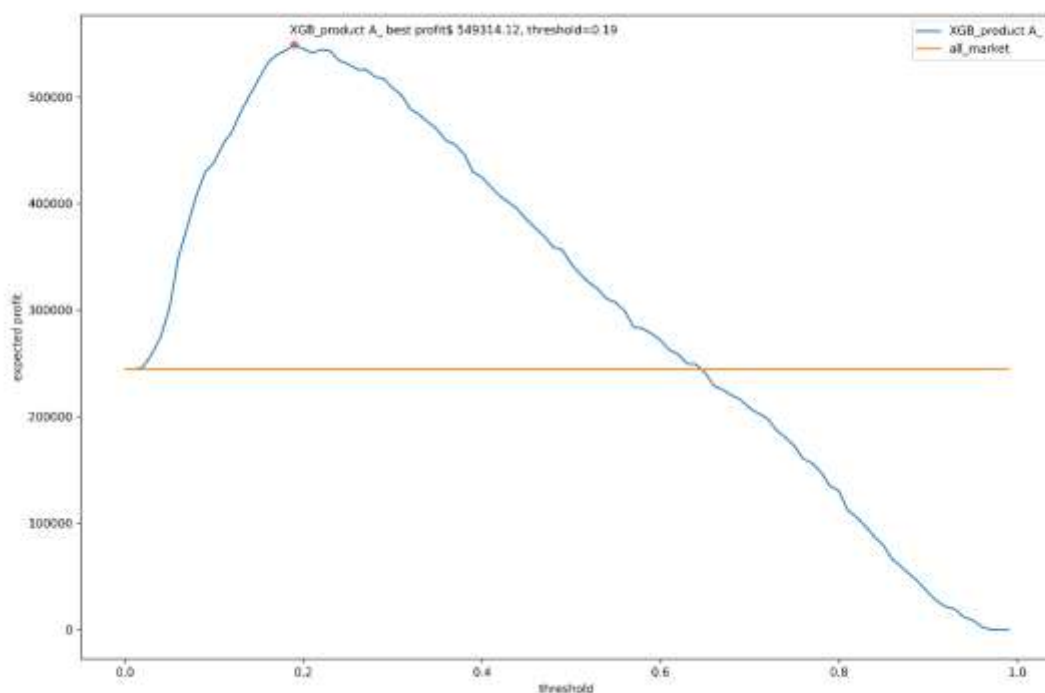
程式碼11

```
# %%
# SECTION - 程式碼11
# 畫出XGB 利潤折線圖
profit_linechart(
    y_test_df=XGB_y_test_df, # 顧客精準行銷清單
    sales_price=2500, # 價格
    marketing_expense=185.32, # 行銷費用
```

```
product_cost=1215.2,  
plot_name='XGB_product A_')  
  
# !SECTION - 程式碼11
```

程式碼11的產出

XGB_product A__預期獲利最佳化模型折線圖.png



Jasper回饋：「原來如此！這樣不但可向主管說明具體成效且又協助公司開源節流，一舉多得！」

五、本案提高了行銷成效，減少客訴，甚至是開源賺錢與節流省錢

接著，讓筆者從高階管理者與策略制定者，如：總經理；以及中間管理者與第一線人員，如：行銷資料家或主管的角度來綜整本案結論。

主管或總經理們的強項在產業知識、經驗與直覺。從系統觀點來看，主管們應該把重點放在「輸入（Input）」與「輸出（Output）」，而非「處理（Process）」。

也就是說，主管應該將重心放在「定義問題」以及「分析結果」上，至於「資料分析」的方式，則交由資料科學家來執行，如圖8所示。

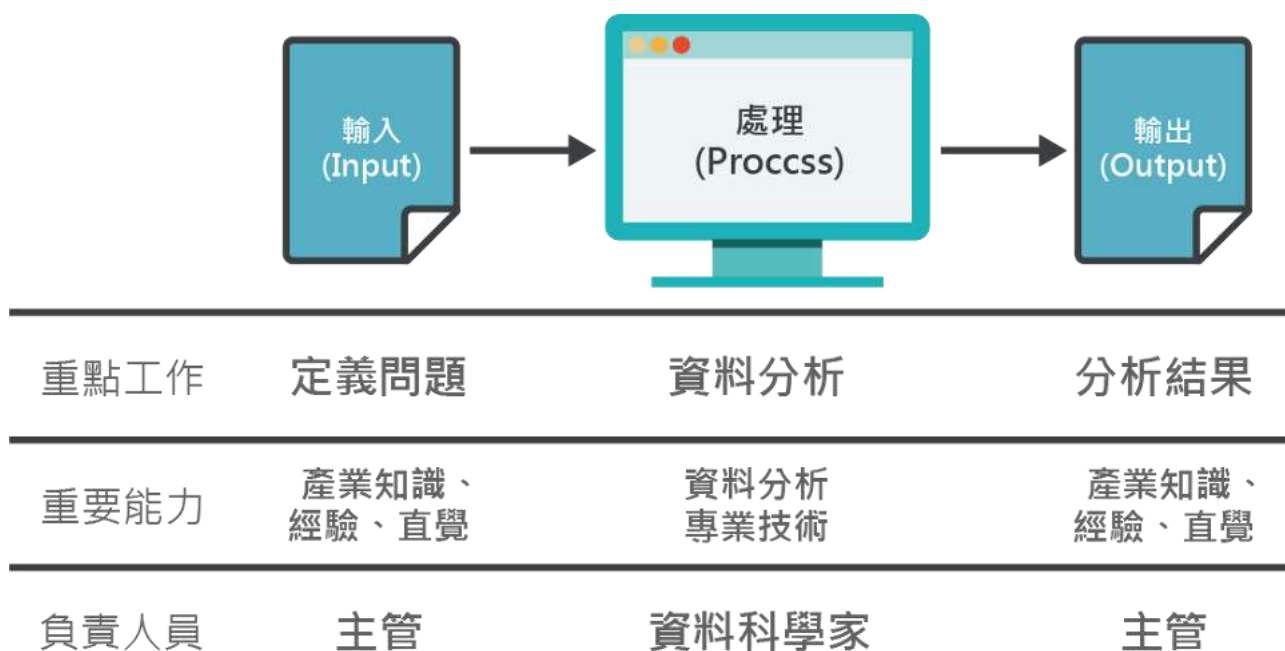


圖 8.資料科學家與主管所扮演的角色（繪圖者:張庭瑄）

對行銷資料科學家而言，要試圖從不同模型中找出最適獲利或組織要求的KPI並時刻回頭檢視自己是否偏離了KPI的範疇，最終能協助找出資料金字塔中，高階管理者所需的「知識」甚至是策略制定者所需的「智慧」，在此例即得出：「只要預測到顧客有19%的意願購買下，就應該要行銷他，就可造成\$549,314元的預期獲利。」的知識，如圖9所示。最後，則要確認在精準行銷的模式下是否遠比之前廣發宣傳的效果還要來得好。

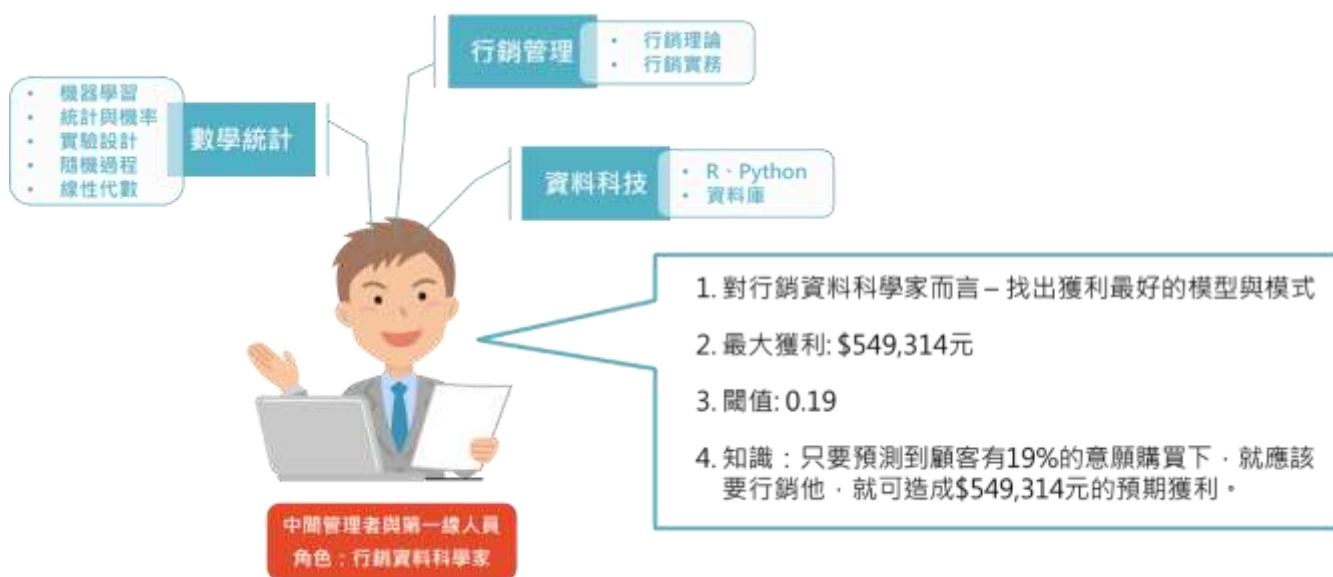


圖 9. 行銷資料科學家在專案協作的範疇（繪圖者：周晏汝、鍾皓軒）

對總經理或管理者而言，資料科學指標並不代表一切，如本例精準行銷的提升度（Lift）具2.94倍，意味用精準行銷的成效是高於對全顧客行銷，但是該指標卻無法涵蓋企業的財務架構，使得給予管理者的決策建議不但不直觀，無法實際接地氣，更遑論與沒有資料科學背景的公司同仁溝通。

湯瑪斯·戴文波特（Thomas H. Davenport）在中文版哈佛商評中2019年的一篇文章——《經理人應該如何運用數據資料》(Davenport, 2019)所道：「經理人的職責是為分析師設定適當的參數，並以具說服力的方式，傳達他們發現的結果」

故管理者為行銷資料科學家或對應的主管設定目標時，除了要瞭解資料科學指標外，更要從獲利或欲達成的KPI來設定「適當的參數」，以更通透的傳達最終所產出的「知識」，如圖10所示。在本案，戰略上，主管可以透過行銷資料科學家所做出來的精準行銷策略來得知\$549,314元的預期獲利且認知到公司商品性質是適合精準行銷方法來推薦商品給顧客的；戰術上，可以請第一線的行銷人員依照「顧客精準行銷清單」來實際執行並產出主管真正要的獲利成果。

有了本次專案經歷後，主管接著則應該將相關專案經歷更落實到人才的「選、訓、用、留」上，為未來的行銷資料科學專案創造更好的良性循環。

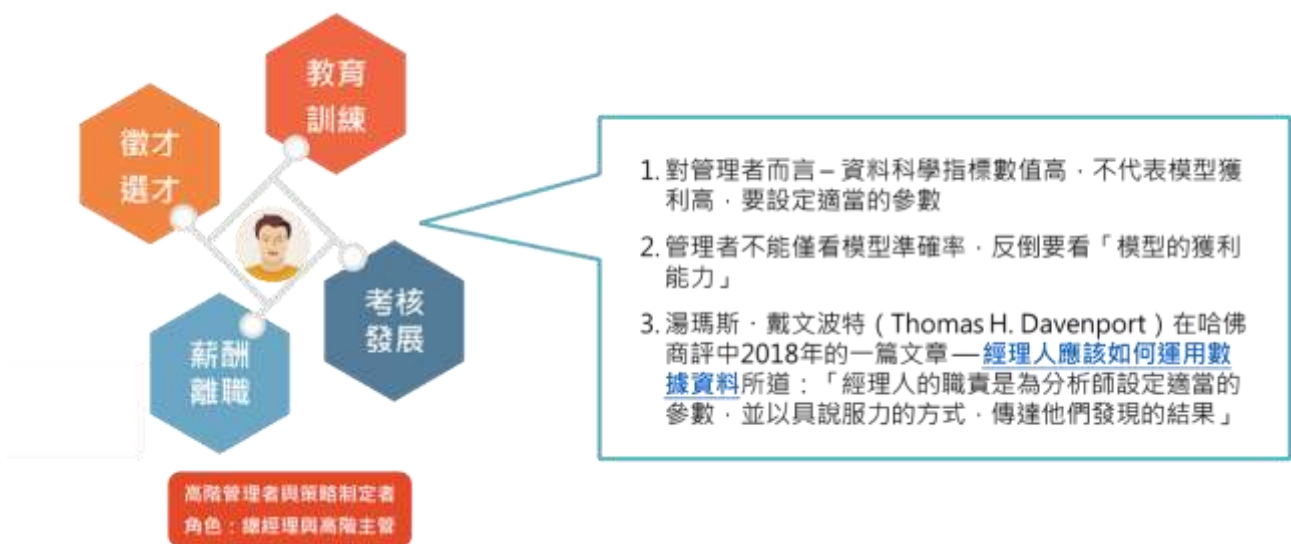


圖 10. 總經理或主管在專案協作的範疇（繪圖者：周晏汝、鍾皓軒、鍾淳育）

最後筆者則建議Jasper團隊持續優化CRM資料，諸如：透過何種行銷管道成功接觸到顧客？顧客的購物車行為為何？等有諸多資料欄位尚可優化，讓精準行銷模型也能不斷優化，增益企業獲利可能性。

肆、後記：低程式碼應用程式展演（Low-code app demo）

當然本案沒有因此結束，因為Jasper不僅有A商品，尚有不同商品欲執行精準行銷，所以便詢問：「本次結果團隊皆大歡喜，那請問是否有相關低程式碼應用工具（low-code app）可以拖拉（drag and drop）資料集並點擊使用呢？讓團隊可以快速知曉其他商品精準行銷怎麼做？」

這是完全沒有問題的！筆者善用Python結合網頁技術，迅速完成了一組low-code app網頁應用，讓Jasper團隊往後只要將如圖11的訓練資料集與測試資料集資料放入對應的位置，並在圖12中指定「目標變數」、「UID」與「財務指標」，設定適當的參數，即可開始執行精準行銷的機器學習訓練任務。



圖 11. 善用 Low-code app，只要拖拉 CRM 資料，就可開始分析

TMR
臺灣行銷研究有限公司

Navigation
Menu
模型訓練

請上傳資料

請輸入目標變數

請輸入目標變數

買A商品

請輸入本資料集的User id

請輸入本資料集的User id

UID

財務指標

售價

2900

商品成本

1215

行銷費用

185

不購買89.17% of the sales_dataset

購買10.83% of the sales_dataset

圖 12. 操作人員可以手動填寫財務等重點指標，為模型設定「適當的參數」

接著，有別於本章使用的XGBoost模型，筆者亦可納入不同的機器學習模型來訓練，藉此找出最適合不同商品的模型，如圖X所示。



圖 13. 準備訓練不同模型，找出不同商品最適合的模式

接著，App就會自動選出最好的模型、最好的預期獲利與最佳的閾值，在圖14中可見到最好的模型為XGBoost並且行銷購買本商品機率超過23%的顧客下，預期獲利為\$459,235元新台幣。



圖 14. App 會自動選出最好的模型、最好的預期獲利與最佳的閾值

除此之外，App還會自動產出Jasper團隊欲與行銷全顧客利潤的精準行銷比較表，如圖15所示，本案可見精準行銷之方法比原方法預期表現還要好上約10萬元新台幣。

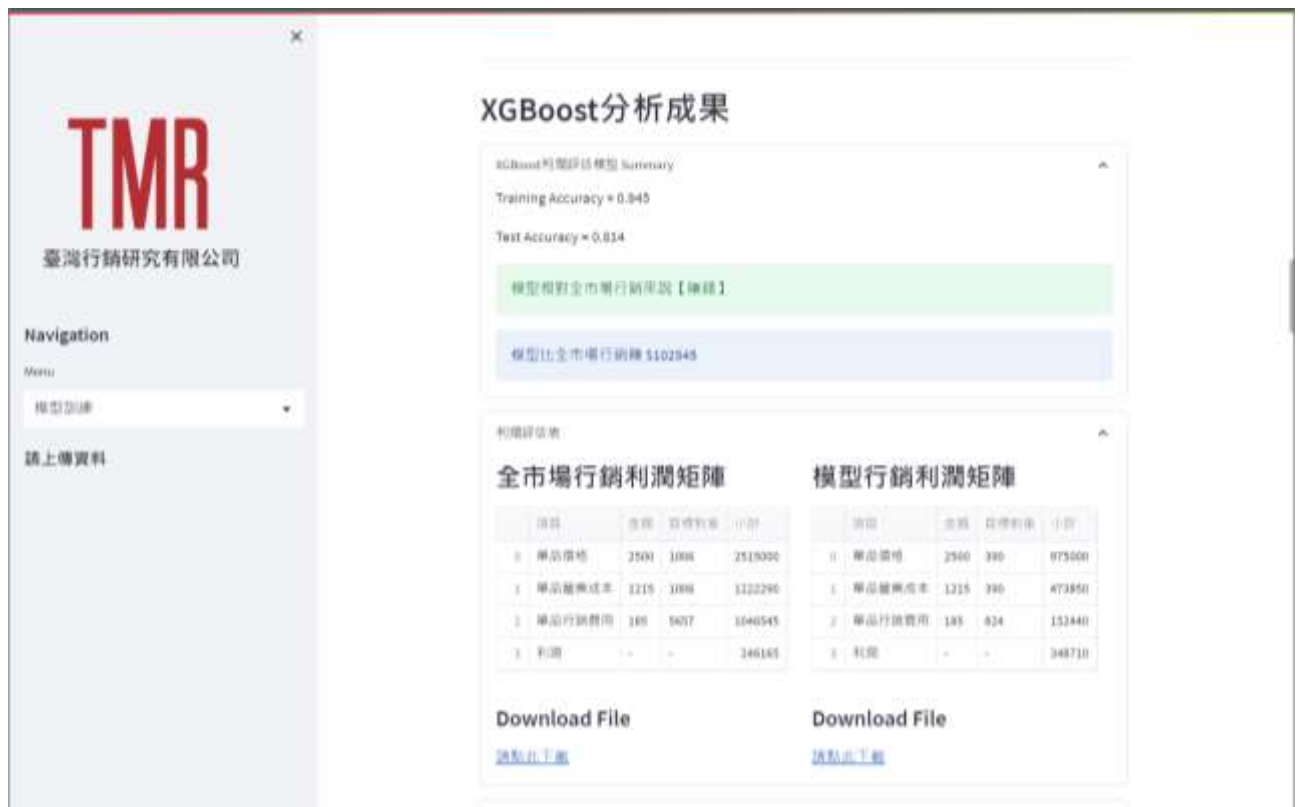


圖 15. 行銷全顧客與精準行銷特定顧客所創造的利潤比較表

不僅如此，當Jasper團隊面對還不知道是否有購買本商品意願的新顧客時，僅需將顧客資料放入圖16的App中，筆者可多善用貝氏統計模型，建構出「顧客精準行銷清單」與「未來的總預期獲利」

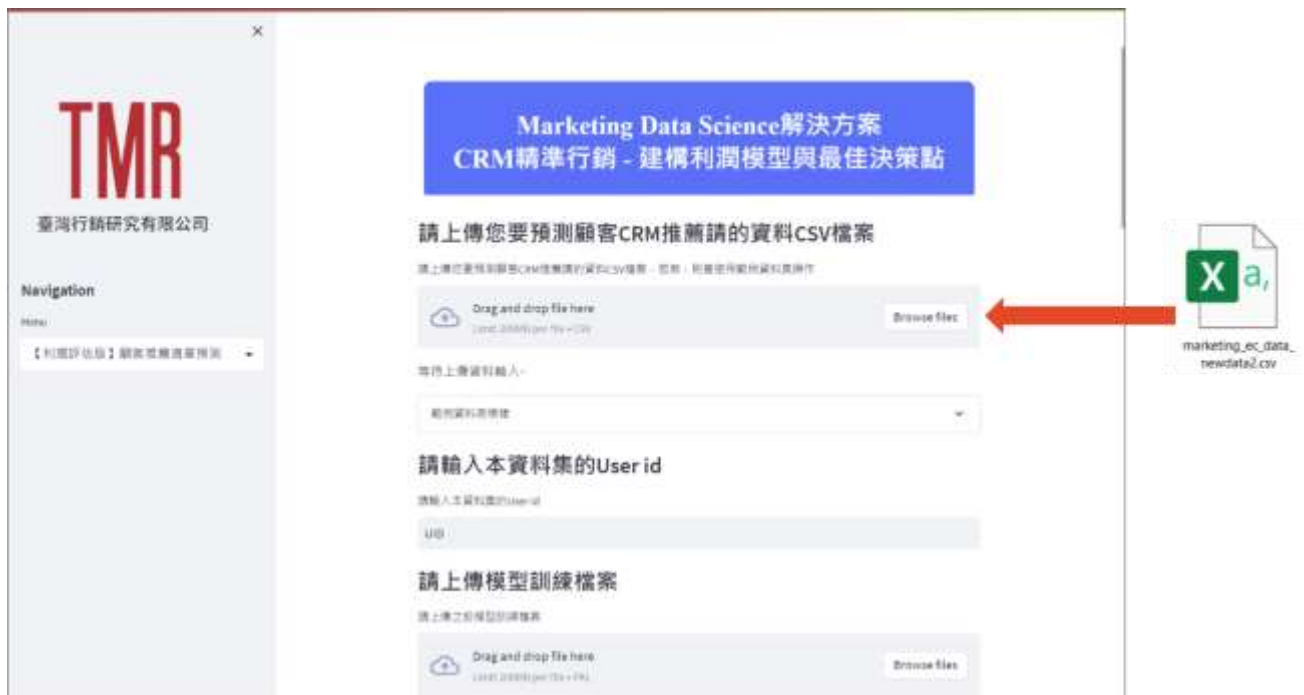


圖 16. 可將新顧客的資料一樣放入 App，作為精準行銷預判的預測資料

以圖17為例，可見在這接近總數5,000人的總資料中，模型幫Jasper挑選出542人，並依照貝氏統計模型的運算，預計會有257人買單本案商品，即造就接近23萬元新台幣之成效，其ROAS為6.4倍，ROI也達近2.3倍。



圖 17. 加入貝氏統計模型，估計新顧客推薦獲利

最後，本App的精準行銷釋例不僅可運用在如Jasper的電商案例，亦可泛用在任何B2C業界，善用精準行銷迎合真正有需求的顧客，進而讓企業的獲利無往不利。

References

- Akerkar, R. (2019). *Artificial intelligence for business*. Springer.
- Brin, S., Motwani, R., Ullman, J. D., & Tsur, S. (1997). Dynamic itemset counting and implication rules for market basket data. *ACM SIGMOD Record*, 26(2), 255-264. <https://doi.org/10.1145/253262.253325>
- Davenport, T. H. (2019). How Managers Should Use Data. *Harvard Business Review*. <https://hbr.org/2013/06/how-managers-should-use-data>
- Gallo, A. (2014). The value of keeping the right customers. *Harvard Business Review*, 29(10), 304-309. <https://hbr.org/2014/10/the-value-of-keeping-the-right-customers>
- Kotler, P., & Keller, K. L. (2016). *Marketing management (15th global ed.)*.