

關鍵績效指標與視覺化

在推展行銷活動或任何行銷工作時，你很可能想知道每個行銷活動的表現如何，並瞭解這些行銷工作的優劣之處。本章將探討常用的關鍵效能指標（KPI），這些指標有助於追蹤行銷工作的表現。更具體地說，我們將討論銷售收入、每次收購成本（CPA, Cost Per Acquisition）、數位行銷 KPI 和網站流量等關鍵績效指標。我們將學習這些 KPI 如何幫助你維持正確方向，確實達成行銷目標。

在探討常用的 KPI 之後，我們將學習如何使用 Python 和 / 或 R 來運算此類 KPI，並將這些 KPI 以視覺化呈現。在本章中，我們將應用一份銀行行銷資料集，這份資料集是一份真實案例，展示金融組織如何進行市場行銷活動。關於 Python 專案，我們將學習如何使用 pandas 和 matplotlib 程式庫來分析資料並構建視覺化呈現。至於 R 專案，我們將介紹 dplyr 和 ggplot2 程式庫，對資料進行分析和運算，並以視覺化呈現。

我們將在本章介紹下列主題：

- 衡量不同行銷績效的 KPI
- 使用 Python 運算並視覺化 KPI
- 使用 R 運算並視覺化 KPI

衡量不同行銷績效的 KPI

每一次進行行銷活動，對組織來說都是一次花費。當你透過電子郵件推展行銷活動，傳送每一封郵件都需要花費成本。當你在社群媒體服務或廣播媒體上進行行銷活動，都需要一筆花費。正因為每一種行銷活動都隱含著金錢成本，檢視行銷成效並追蹤其投資回報率（ROI, Return on Investments）非常重要。本節內容將著重討論如何追蹤銷售收入、CPA 與數位行銷 KPI。

銷售收入

顯而易見，任何行銷工作的目標都是為組織帶來更多銷售收入。沒有一間公司願意在行銷上入不敷出。為了準確回報銷售收入，你必須明確定義如何將銷售額歸因於每一份行銷工作上。某些銷售額可能來自電子郵件行銷活動，另一些銷售額可能來自電視或公共交通上所投放的廣告效益。甚至，某些銷售額可能無法歸因於任何行銷活動，它們可能是自然流量。

為了準確回報各種行銷活動各自帶動多少銷售額，你需要定義將銷售額歸因至各行銷活動的明確規則。舉例來說，如果你服務於一間電商公司，透過電子郵件和電視行銷活動來推廣一些特別優惠，你可能需在電子郵件與電視廣告中分別放置不同的 URL（網址）。這樣一來，你就可以判斷哪些銷售收入來自於電子郵件行銷、哪些來自電視廣告行銷。

根據需求，你可能也希望按時間順序回報銷售收入資料。你就可以利用試算表進行回報，如下圖：

如上列程式碼所示，我們匯入了 pandas 程式庫，並設定別名為 pd，同時使用 read_csv 函數來載入資料。如欲使用除了逗號以外的分隔符，你可以利用 sep 引數自定義使用於 read_csv 函數中的分隔符。

如果你閱讀下載頁面 (<https://archive.ics.uci.edu/ml/datasets/bank+marketing>) 的欄位說明，輸出變數 y 表示用戶是否購買定期存款的資訊，其輸出結果被編碼為 yes 或 no。為了簡化運算過程，我們以 1 表示 yes，以 0 表示 no，以下是程式碼：

```
df['conversion'] = df['y'].apply(lambda x: 1 if x == 'yes' else 0)
```

正如上述程式碼所示，我們為變數 y 套用 apply 函數，以 1 表示 yes，以 0 表示 no，然後將經編碼的資料新增一個新欄位 conversion。在 Jupyter Notebook 中程式碼以及載入資料應顯示如下圖：

```
import pandas as pd

df = pd.read_csv('../data/bank-additional-full.csv', sep=';')

df['conversion'] = df['y'].apply(lambda x: 1 if x == 'yes' else 0)

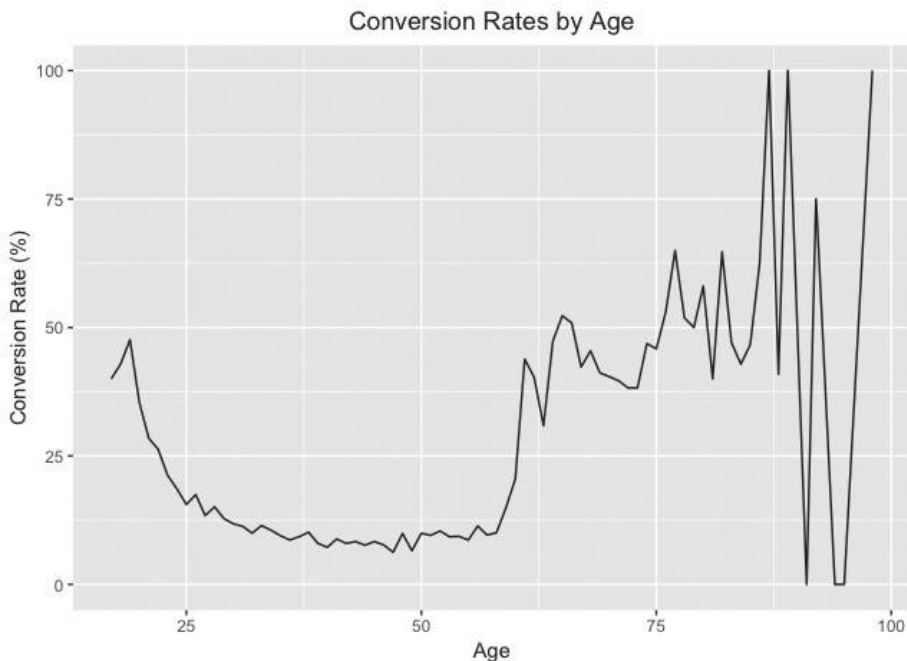
df.head()
```

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	...	pdays	previous	poutcome	emp.var.rate	cons.price.idx
0	56	housemaid	married	basic.4y	no	no	no	telephone	may	mon	999	0	nonexistent	1.1	93.994
1	57	services	married	high.school	unknown	no	no	telephone	may	mon	999	0	nonexistent	1.1	93.994
2	37	services	married	high.school	no	yes	no	telephone	may	mon	999	0	nonexistent	1.1	93.994
3	40	admin.	married	basic.6y	no	no	no	telephone	may	mon	999	0	nonexistent	1.1	93.994
4	56	services	married	high.school	no	no	yes	telephone	may	mon	999	0	nonexistent	1.1	93.994

5 rows x 22 columns

成功將資料讀取至 pandas DataFrame 之後，我們可以使用多種方法與圖表，學習如何分析與視覺化轉換率。

按客戶年齡檢視對應轉換率的另一個作法是繪製線性圖，如下圖所示：



將不同年齡群組的轉換率視覺化呈現的程式碼如下：

```
ax = conversions_by_age.plot(
    grid=True,
    figsize=(10, 7),
    title='Conversion Rates by Age'
)

ax.set_xlabel('age')
ax.set_ylabel('conversion rate (%)')

plt.show()
```

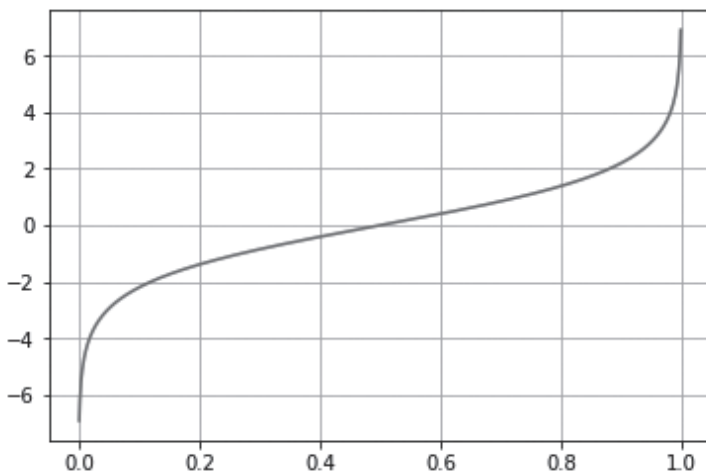
如上所示，我們使用先前建立的 `conversions_by_age` 變數以及 `plot` 函數來繪製線性圖。你可以利用 `figsize` 引數來變更圖表大小，利用 `title` 引數變更圖表標題。如欲變更 x 軸與 y 軸的標籤，可以使用 `set_xlabel` 及 `set_ylabel` 函數。

邏輯迴歸

邏輯迴歸是一種迴歸分析，適用情況為當輸出變數為二元變數時（1 表示結果為正，0 表示結果為負）。與任何其他線性迴歸模型一樣，邏輯迴歸模型透過將特徵變數線性組合來預測輸出結果，唯一差異在於此模型預測的內容，邏輯迴歸模型預測事件發生機率的對數，換句話說，它預測的是正事件和負事件的發生機率的對數。模型公式如下所示：

$$\log\left(\frac{P(y=1)}{1-P(y=1)}\right) = a + b_1 \times X_1 + b_2 \times X_2 + b_3 \times X_3 + \dots$$

位於公式左側的比率是成功機率，表示成功機率和失敗機率之間的比率。對數機率的曲線（也稱為對數曲線）如下所示：



邏輯迴歸模型輸出其實就是對數的倒數，其值的範圍從 0 到 1。本章中將使用迴歸分析來瞭解驅動用戶參與度的因素，用戶是否回應行銷電話是此處的輸出變數。因為輸出是一個二元變數：回應與未回應，因此邏輯迴歸非常適合在這種情境使用。在下列各節中，我們將討論如何在 Python 和 R 中使用和構建邏輯迴歸模型，然後介紹如何解釋迴歸分析結果，以瞭解哪些客戶屬性與高度行銷參與度有密切關聯。

解讀決策樹

有了經過訓練的決策樹模型之後，我們要從中取得洞見。在本節內容中，我們將會使用一個名為 `rattle` 的程式庫：

1. 你可以在 RStudio 使用以下指令來安裝套件

```
install.packages("rattle")
```

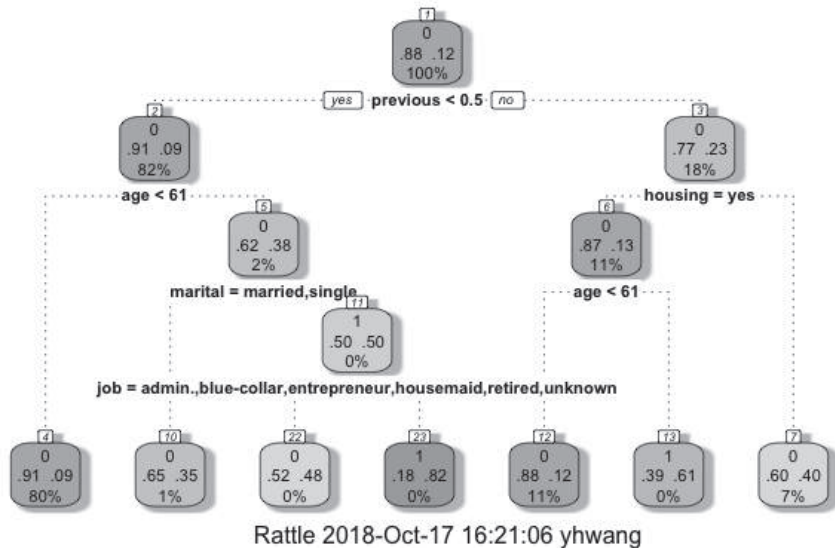
2. 正確安裝後，你應該可以比照下列程式碼，匯入程式庫：

```
library(rattle)
```

3. 為 R 環境新增此程式庫 `rattle` 後，還需要一行程式碼來視覺化呈現這個經過訓練的決策樹模型。請閱讀以下程式碼：

```
fancyRpartPlot(fit)
```

4. `fancyRpartPlot` 函數採用了一個 `rpart` 模型物件。此處的模型物件 `fit` 正是我們在先前步驟中建立的決策樹模型。當你運行這則指令後，將會出現以下的圖表：



產品分析的重要性

產品分析是一種從資料中取得洞察的方法，可瞭解客戶如何與商家所提供的產品進行互動與參與、不同產品之間的效能差異，以及業務中某些可觀察到的弱勢和優勢。不過，產品分析可不僅僅停留在分析資料上。事實上，產品分析的終極目標是建立切實可行的洞察和報告，以便進一步優化和改進產品效能，並根據產品分析結果產生新的行銷或產品創意。

產品分析從追蹤事件開始。這些事件包括網站訪客次數、網頁流覽次數、瀏覽器歷史記錄、購買次數，或者是客戶可以使用你所提供的產品而進行的任何動作。接著，你可以開始分析和視覺化呈現這些事件中的任何可觀察模式，以便建立可行的洞察或報告。產品分析的一些常見目標如下：

- **提升顧客和產品留存率**：仔細分析顧客瀏覽與購買的內容，你可以辨識哪些商品經常被回購，以及哪些顧客屬於回頭客。此外，你也可以找出顧客不會購買的商品與存在流失風險的顧客。分析和瞭解回購商品和回頭客的常見屬性可以幫助你進一步改善使用者留存策略。
- **辨識人氣熱門產品**：身為零售業務的行銷人，清楚掌握人氣產品和熱門產品非常重要。這些暢銷產品為業務帶來主要金流收入，並提供了新的銷售機會，例如交叉銷售或搭配銷售。你可以利用產品分析，輕鬆辨識和追蹤這些人氣熱門產品，並使用這些最暢銷的產品建立全新銷售策略，探索不同的機會。
- **根據顧客和產品的關鍵屬性進行區隔**：掌握顧客資料和產品數據，您可以使用產品分析以特定屬性對顧客群和產品進行區隔，比如獲利率、銷售量、重新訂購量和退款數量。這些區隔可幫助你總結確實可行的洞察，找出在下一階段要鎖定的目標產品或客戶群。

- **制定較高投資報酬率的行銷策略**：產品分析還可用於分析行銷策略的**投資報酬率 (Return on Investment, ROI)**。你可以對推廣某項商品的行銷費用以及這些產品所產生的收入進行分析與評估，瞭解哪些策略確實可行，哪些作法不符效益。利用產品分析進行行銷 ROI 分析，可幫助你建立更高效的行銷策略。

目前為止我們討論了產品分析的內涵，以下的程式設計練習中將探討如何運用零售業務資料，達成上述的產品分析目標。我們將討論如何使用這些資料來分析回頭客的模式及其對總體收入的貢獻。此外，我們也會介紹如何使用產品分析來分析暢銷產品的行為。具體而言，我們將討論如何追蹤流行商品隨時間變化的趨勢，並簡要敘述如何在行銷策略中利用此趨勢產品的資料進行產品推薦。

以 Python 進行產品分析

本節將探討如何使用 Python 的 pandas 和 matplotlib 套件來執行產品分析。想要使用 R 語言的讀者，請跳至下一節內容。首先，我們將分析收入與購買量的時間序列趨勢，以及回頭客的購買模式，最後我們將分析產品的銷售趨勢。

在本練習中，我們會使用 UCI Maching Learning Repository 的公開資料集，你可以在此查看：<http://archive.ics.uci.edu/ml/datasets/online+retail#>。請下載名為 Online Retail.xlsx 的 Microsoft Excel 檔案，下載完成後，請使用下列指令將其載入至 Jupyter Notebook：

```
%matplotlib inline

import matplotlib.pyplot as plt
import pandas as pd

df = pd.read_excel(io='../data/Online Retail.xlsx', sheet_name='Online Retail')
```


正如前幾章所進行的 Python 練習，我們使用 `%matplotlib inline` 指令，在 Jupyter Notebook 中顯示圖表。接著，我們匯入 `pandas` 和 `matplotlib` 套件，以利進行後續的產品分析。這段程式碼中另一個值得注意的地方是我們使用了新的函數，`pandas` 套件的 `read_excel`。此函數可讓你載入任何 Excel 檔案到 `pandas DataFrame` 中。如程式碼所示，我們傳遞了兩個引數到 `read_excel` 函數：描述檔案路徑的 `io` 與欲載入 Excel 工作表的檔案名稱 `sheet_name`。

當你將資料載入至 `pandas DataFrame` 中，應如以下擷取畫面所示：

```
df.shape
(541909, 8)
```

```
df.head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

在進入下一步驟之前，我們必須先進行資料清理。快速看看 `Quantity` 欄位的資料分布情形，使用以下程式碼來視覺化呈現其資料：

```
ax = df['Quantity'].plot.box(
    showliers=False,
    grid=True,
    figsize=(10, 7)
)

ax.set_ylabel('Order Quantity')
ax.set_title('Quantity Distribution')

plt.suptitle("")
plt.show()
```

時間序列趨勢

在檢視產品層級的資料之前，身為一個電子商務的行銷人，掌握收入、訂單量或購買量的整體時間序列趨勢非常有益。如果能掌握某段時間內的整體銷售數字和訂單量，將有助於我們理解業務是否成長或衰退。

首先，我們要檢視一段時間內的訂單量，請閱讀以下程式碼：

```
monthly_orders_df =
df.set_index('InvoiceDate')['InvoiceNo'].resample('M').nunique()
```

這段程式碼中出現了前幾章不曾使用過的 `resample` 和 `nunique` 函數。`resample` 函數對資料重新採樣，並將時間序列資料轉換為我們所希望的頻率。在本範例中，我們將時間序列資料重新採樣為以月為單位的時間序列資料，指定 'M' 為目標頻率，並計算獨立或不重複的訂單數量。這麼一來，我們可以取得每月的不重複購買量或訂單量，而資料框如下圖所示：

InvoiceDate	InvoiceNo
2010-12-31	1629
2011-01-31	1120
2011-02-28	1126
2011-03-31	1531
2011-04-30	1318
2011-05-31	1731
2011-06-30	1576
2011-07-31	1540
2011-08-31	1409
2011-09-30	1896
2011-10-31	2129
2011-11-30	2884
2011-12-31	839

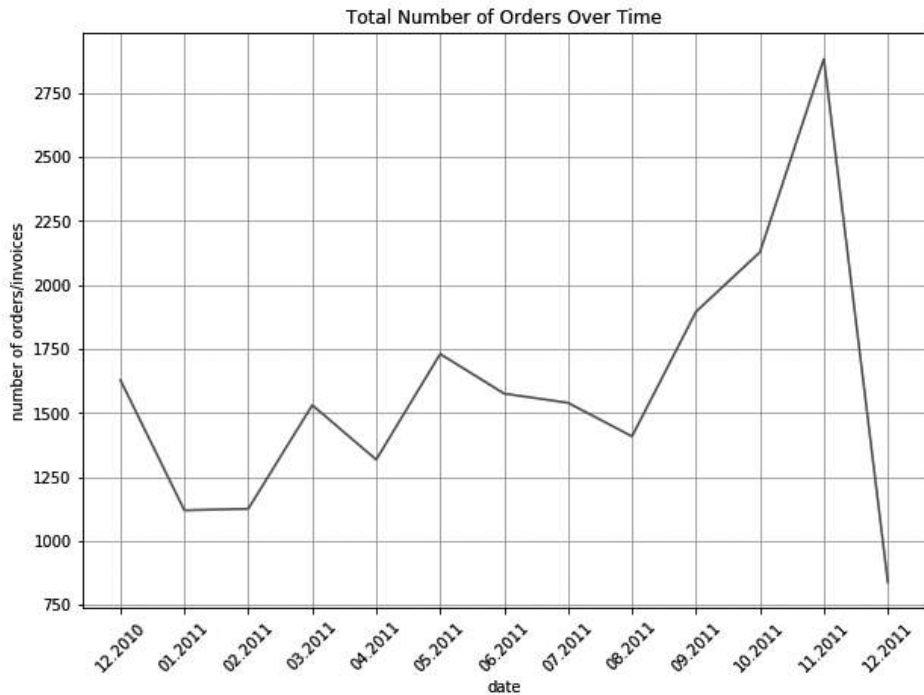
Freq: M, Name: InvoiceNo, dtype: int64

通常，時間序列資料以線圖進行視覺化呈現。讓我們看看以下程式碼，瞭解如何將這份每月資料以線圖呈現：

```
ax = pd.DataFrame(monthly_orders_df.values).plot(
    grid=True,
    figsize=(10,7),
    legend=False)
```

```
)  
  
ax.set_xlabel('date')  
ax.set_ylabel('number of orders/invoices')  
ax.set_title('Total Number of Orders Over Time')  
  
plt.xticks(  
    range(len(monthly_orders_df.index)),  
    [x.strftime('%m.%Y') for x in monthly_orders_df.index],  
    rotation=45  
)  
  
plt.show()
```

我們在 pandas DataFrame 中使用了 plot 函數。然後使用 matplotlib 套件的 xticks 函數，來自訂 X 座標軸的刻度標籤。請先看看下圖：



你可能發現圖中 X 軸的刻度標籤以月和年進行設定。再次閱讀之前的程式碼，我們使用 `x.strftime('%m.%Y')` 來設定格式，此處的 `x` 是 Python `date` 物件，`%m` 是月份值的佔位符，`%Y` 是年份值的佔位符。Python `date` 物件的 `strftime` 函數將日期以給定格式進行設定。

此圖表中值得特別留意的地方是，在 2011 年 12 月時訂單量急遽減少。如果仔細檢視資料，這是因為我們並不具備 2011 年 12 月的完整月份資料，可以透過以下程式碼確認這一點：

```
invoice_dates = df.loc[
    df['InvoiceDate'] >= '2011-12-01',
    'InvoiceDate'
]

print('Min date: %s\nMax date: %s' % (invoice_dates.min(),
    invoice_dates.max()))
```

在這段程式碼中，我們取得從 2011 年 12 月 1 日開始的所有訂單日期，然後我們輸出日期最小值與最大值。當你運行此段程式碼，你將會看到如下輸出：

```
print('Min date: %s\nMax date: %s' % (invoice_dates.min(), invoice_dates.max()))
Min date: 2011-12-01 08:33:00
Max date: 2011-12-09 12:50:00
```

如輸出結果所示，我們只有 2011 年 12 月 1 日至 12 月 9 日的資料。如果我們僅使用這筆資料，將無法代表十二月的銷售量與收入。為了方便後續分析，我們將忽略 2011 年 12 月 1 日以後的資料，你可以使用以下程式碼移除這些資料點：

```
df = df.loc[df['InvoiceDate'] < '2011-12-01']
```