資料分析 HW3-2 Report

F74056085 林哲寬

Dataset: Online Shoppers Purchasing Intention Dataset Data Set

Problem Definition: Revenue(True/False)

說明：

這個 Dataset 收集的是購物網站的使用者的瀏覽記錄，像是使用的瀏覽器、作業系統、點擊了那種類型的網站等等，目的是判斷該名使用者會不會購買。在此次作業中我解決的方式是 **Classifier**，並以 **F1 score** 作為模型能力的量尺。

Method:

Part I: Data Visualization & Preprocessing

1. 觀察整體數據

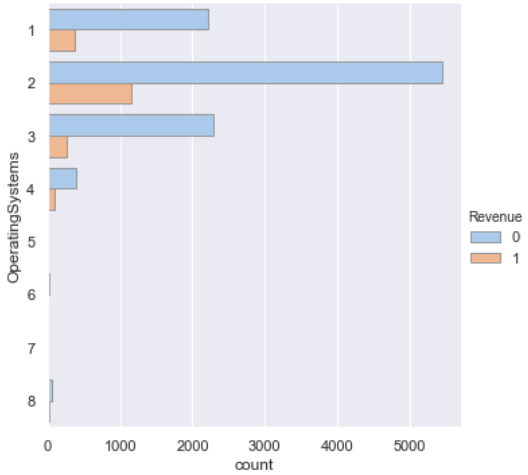
這個 Dataset 所有的欄位都是完整的，因此並不需要進行填補缺失值的處理。但是可以明顯發現負面樣本遠多於正面樣本，這意味著我們必須採取和以往不同的量尺，否則模型將會嚴重失真。

2. 觀察各特徵對結果的影響

為了找出那些特徵最具影響力，以及該使用什麼樣的方式來處理，以下將一一對各個特徵，以視覺化的方式來作分析。

作業系統(OperatingSystems)

欄位說明：這代表的是使用者瀏覽該網頁時使用的作業系統。在資料夾中以代碼的方式呈現。

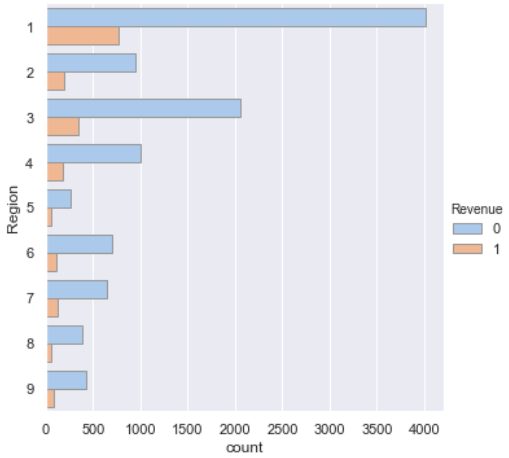


分析結果：從上面的圖表可看出不同的作業系統對 Revenue 的影響沒有顯著的差異。而

且某些 OS 的使用者樣本數過少，應該不是一個很強的特徵。

地區(Region)

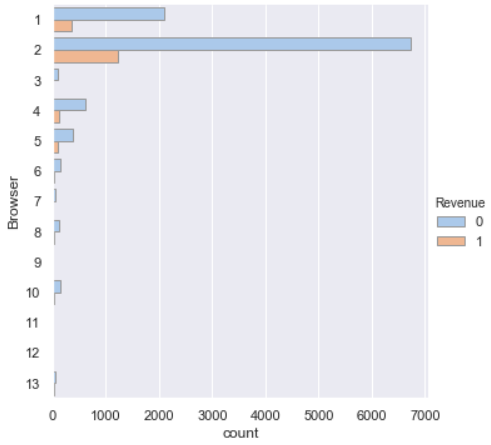
欄位說明：記錄的是潛在的消費者所在的地區。



分析結果：和作業系統一樣，買與不買的比例挺均勻的，沒能給人能單獨決定什麼的感覺。

瀏覽器(Browser)

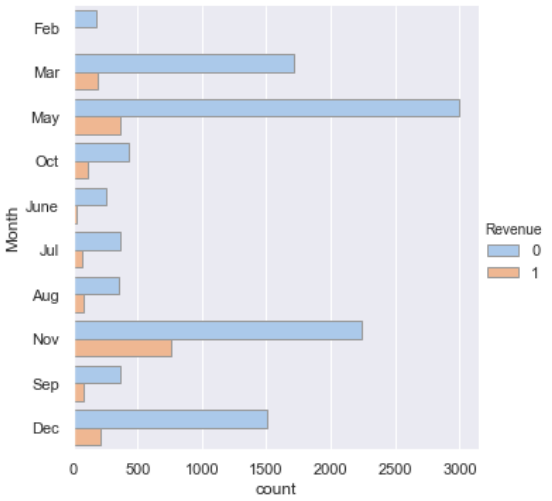
欄位說明：記錄的是使用者使用的瀏覽器。



分析結果：可以注意到有某些瀏覽器的使用者完全沒有消費的紀錄。但那些樣本數本來就比較少，為了避免過於偏頗導致 overfitting，不能當作主要的判斷依據。

消費月份(Month)

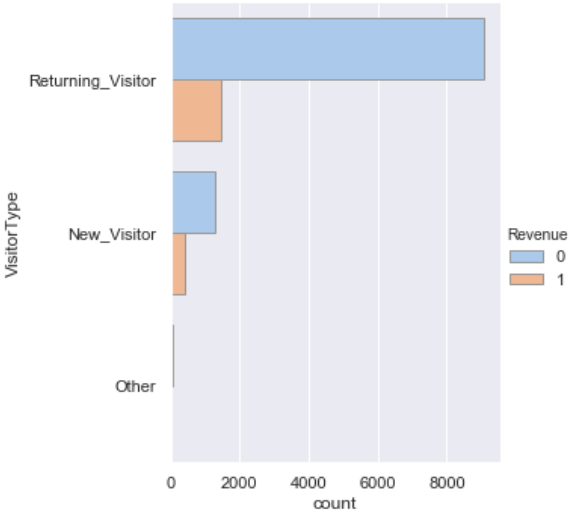
欄位說明：記錄的是使用者留下資料時的月份。



分析結果：五月是擁有最多記錄的，但相對的消費人數卻很低。二月之所以那麼低應該是樣本數太少的緣故。至少我們可以看出十一月的客人消費慾望是比較高的。

訪客類型(VisitorType)

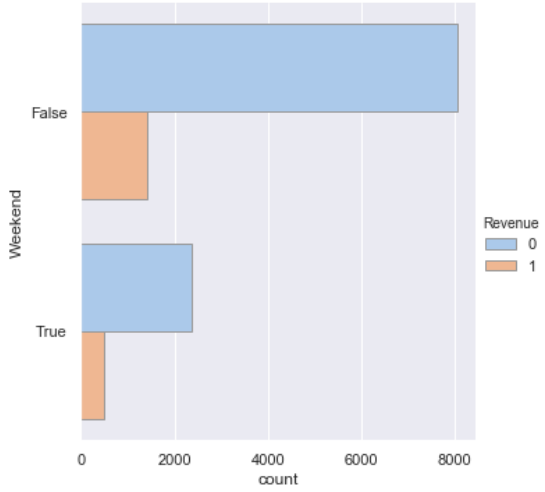
欄位說明：可分為曾經來過的訪客、新的訪客以及其它。



分析結果：理論上再次造訪網站的訪客應該會比較願意掏出錢包，但實際上卻是新的訪客比老客人更有意願購買。推測是彼此之間的個數懸殊所造成的現象。

是否為週末(Weekend)

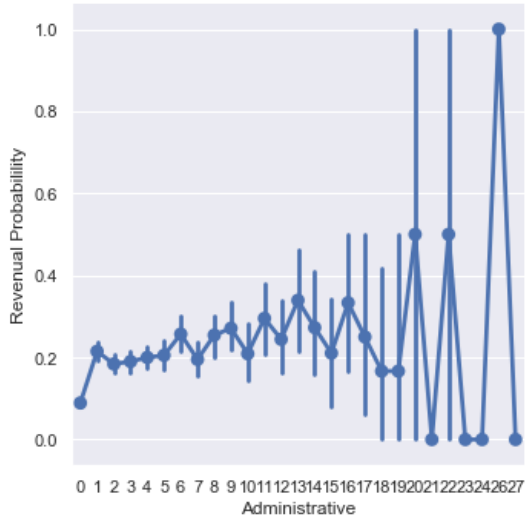
欄位說明：即是否是在週末瀏覽此網站。



分析結果：週末人比較閒，來網站的次數的確比較多，但圖中顯示兩者購買的慾望是差不多的。

行政類網頁瀏覽次數(Administrative)

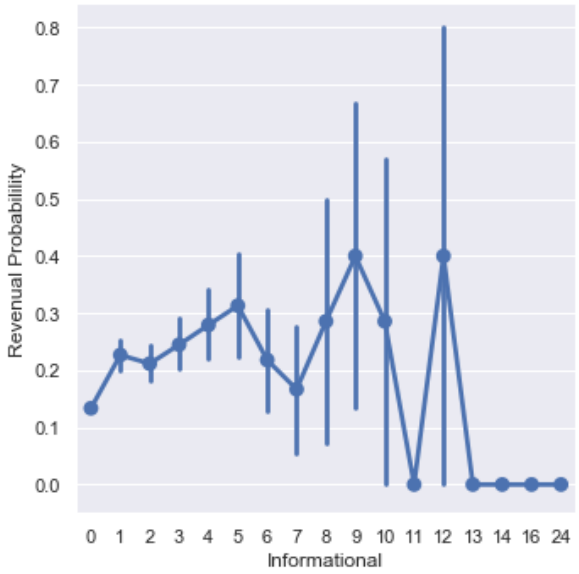
欄位說明：指的是在一個 session 中，該名使用者瀏覽具有管理性質的網頁個數。



分析結果：雖然高於二十次的變化很極端，但如果整合成同一個區間來看的話，會發現分布挺平均的，沒有什麼指標性。

資訊類網頁瀏覽次數(Informational)

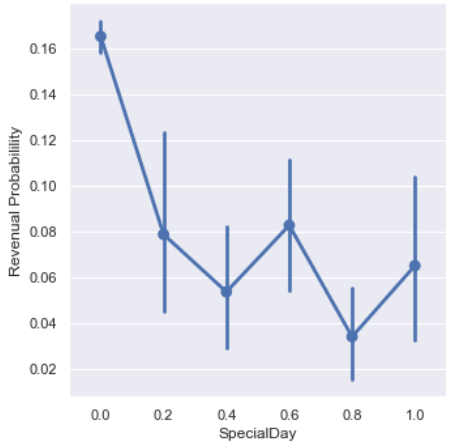
欄位說明：指的是在一個 session 中，該名使用者瀏覽陳列資訊性質的網頁個數。



分析結果：可以明顯看出點擊資訊類網頁超過十二次的顧客意願變得非常的低。應該是可以加以利用的。

特殊節日(SpecialDay)

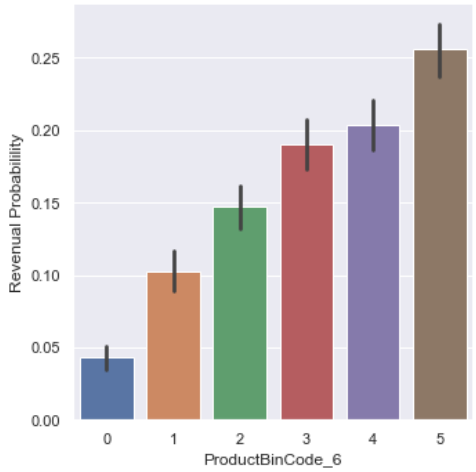
欄位說明：指的是瀏覽的日期與重大節日（例如情人節、聖誕節）的接近程度，相關性從 0 到 1，刻度為 0.2。



分析結果：SpecialDay 的結果有些微妙。它代表的該筆紀錄的日期接近重要節日的程度，如為 1 則代表非常接近。然而下圖顯示出來的趨勢卻是離重要節日越遠購買意願越高的樣子。可能要再觀察一下才把它放進模型的訓練特徵。

產品類網頁瀏覽次數(ProductRelated)

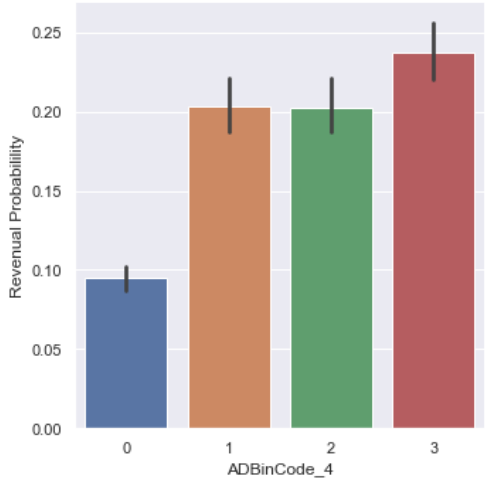
欄位說明：指的是在一個 session 中，該名使用者瀏覽有包含產品介紹的網頁個數。



分析結果：由於 ProductRelated 範圍過廣，直接觀察並不適合，這裡先把它畫分成六個區間。從結果可看出點擊產品相關網頁越多次，購買率的確是有上升的趨勢沒有錯。

行政類網頁瀏覽時間(Administrative_Duration)

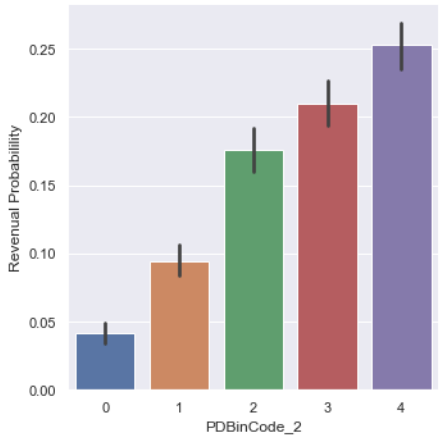
欄位說明：指的是在一個 session 中，該名使用者花多少時間在管理性質的網頁。



分析結果：然後是 Administrative_Duration ，也就是停留在行政網頁的時長和消費意願的關聯性調查。第二到第四區間都差不多，但停留在行政網頁較短的那群人明顯購買的意願更低，或許可以作為參考。

產品類網頁瀏覽時間(ProductRelated_Duration)

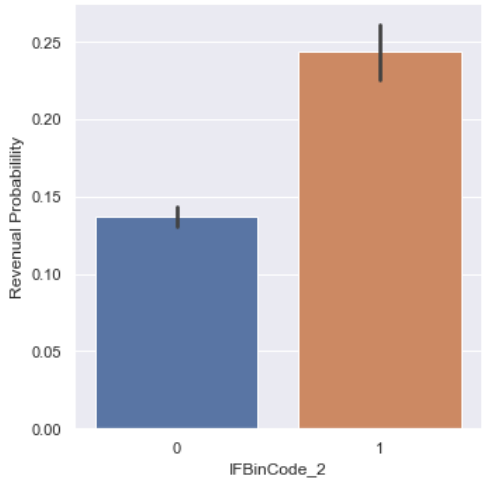
欄位說明：指的是在一個 session 中，該名使用者花多少時間在包含產品的網頁。



分析結果：與 Administrative_Duration 類似，只是區別更為明顯。然而它的分布和上面的 ProductRelated 有點雷同，同時加入訓練特徵的話可能會過於強調。

資訊類網頁瀏覽時間(Informational_Duration)

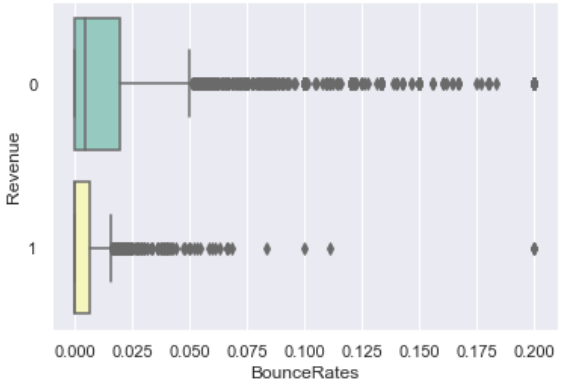
欄位說明：指的是在一個 session 中，該名使用者花多少時間在資訊類的網頁。



分析結果：用二元區間將其分開後，表面上差距並沒有很大，但作完後面的模型測試發閱這種類型的特徵還是需要的。

跳出率(BounceRates)

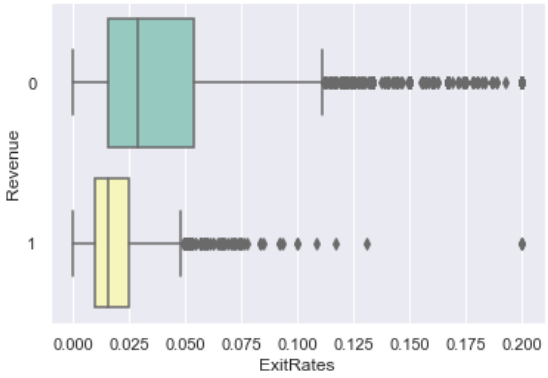
欄位說明：跳出率是指瀏覽該網頁後，沒有瀏覽所屬網站其它網頁就直接離開的比率。可以作為對該網站內容有沒有興趣的指標之一。



分析結果：用盒狀圖分析機率分布後，可以看出低的跳出率雖然不保證購買，但會購買的人跳出率一定很低。

離開率(ExitRates)

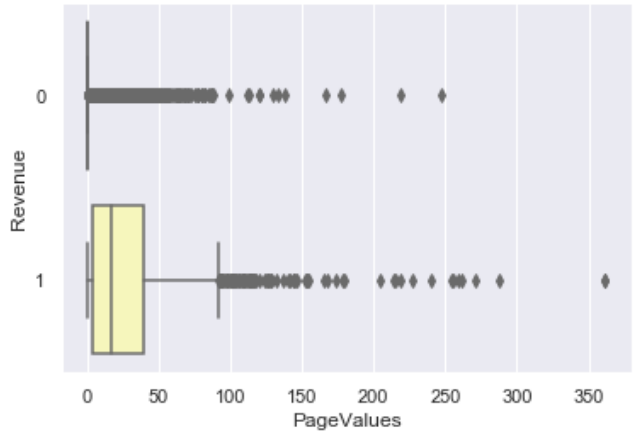
欄位說明：離開率和跳出率一樣是 Google Analytics 提供的數據，指的是訪客在網站上所有的瀏覽過程中，在某一頁結束瀏覽、離開網站的比例。換句話說，訪客在一次完整的造訪過程中，最後離開網站的網頁的比例。



分析結果：離開率的分布則更為明顯。可以看出買與不買是有不同的分布區間的。

網頁值(PageValues)

欄位說明：PageValues 指的是完成線上交易前有瀏覽該網站的比率。



分析結果：象徵不買的那個盒子完全被擠扁了。說明網頁值很低的完全沒有可能購買。這是一個十分強的訊號。

3. 前處理

使用 LabelEncoder 將類別型的資料轉換成數值，並將數值型並連續的資料用 qcut 切成適當的區間，然後再加以編碼。

Part II: Model Training & Improvement

Step 1 : Startup

(訓練準確度 0.51, 測試準確度 0.27)

首先因為沒有提供額外的資料集，我將原本的資料集以 7:3 的比例分割出訓練集及測試集。

一開始先利用 KNN 可以根據最近的點找出答案的特性，建立一個最基本的雛型，並使用 F1 Score 取代原本的 accuracy 計算方式。和前面所說的一樣，這是因為這個資料集的負面樣本遠多於正面樣本，而這個網站購物意願調查的最根本目的又是為了促進客戶消費，Recall 十分重要，一個只能準確判斷負面樣本的模型是不適用的。否則以本資料夾而言全猜 Negative 即可達到至少 0.85 的高準確度。而 F1 Score 的計算方式是 precision 和 recall 的調和平均數，在懲罰把正面樣本當成負面樣本的同時也保證不會一股腦地把所有可能的答案拋出。

起初本來是想把最有可能的特徵當作 baseline，然後再陸續加入其它訊號，但這樣 f1 score 的計算似乎會產生問題的樣子，於是決定先加入所有特徵，然後在一一剔出會造成混淆的屬性來最佳化模型。

第一次嘗試得到的結果還挺慘不忍睹的，不僅是 test accuracy 只有 0.27 這種比亂猜都低很多的分數，就連 train accuracy 也只有 0.5 左右。或許是我們的的確確植入了

過多的噪音，又或者我們挑選的分類器本來就不適用於這種場合。接下來先換其它的分類器試試。

Step 2：更換分類器（Naïve Bayes）

(訓練準確度 0.503，測試準確度 0.524)

本來以為貝氏模型的先驗機率會有很大的幫助，這一次雖然訓練的準確度依然看起來依然沒有什麼起色，但至少 valid accuracy 和 test accuracy 比較正常一點了，這說明了分類性的挑選的確有其重要性。這個結果仍是讓人非常不滿意，還是得再接再厲找尋更強的模型。

Step 3：再次更換分類器（SVM）

(訓練準確度 0.611，測試準確度 0.632)

改用支持向量機後，train accuracy 和 test accuracy 都有很大的成長！原因可能是 SVM 抗雜訊的能力以及各種特徵的組合是高於其它二者的。我們還要再看看有沒有比這個更強的分類器。

Step 4: 最佳化分類器（Decision Tree）

(訓練準確度 0.7，測試準確度 0.645)

再度換成決策樹後，兩項指標都有微幅的成長。看來這樣二元分類的問題是用決策樹的效果不錯。分類器的挑選就到此為止，接下來將剔除不必要的特徵，讓模型達到最高的效能。

Step 5: 剔除冗餘屬性（Region）

(訓練準確度 0.7，測試準確度 0.648)

以下更加仔細地挑選特徵。這裡首先把視覺化時覺得沒什麼關聯性的 Region 剔除後，雖然 train accuracy 降低了一點點，但 test 的結果也變好了一點點。這邊決定將其捨棄掉。

Step 6: 再次剔除冗餘屬性（TrafficType,SpecialDay）

(訓練準確度 0.696，測試準確度 0.652)

仿照上面依序把會造成混淆的 TrafficType 和 SpecialDay 依序丟掉之後，test accuracy 已經來到了 0.652。反覆實驗後發現 **PageValue** 就跟視覺化呈現時發現的一樣，其實是最重要的屬性，少了它了後模型即刻崩潰，甚至比用 KNN 更慘。其它理論上沒什麼關聯性的屬性經實驗後刪除多多少少會讓 test accuracy 反而往下降，這邊決定停手並另闢蹊徑。

Step 7: 新增屬性（PageRatio）

(訓練準確度 0.685，測試準確度 0.654)

視覺化的部分提到可以將 Information 以 12 為界分成二元區間，但實際測試沒有很成功的樣子。由於注意到 PageValue 的重要性，決定建立一個 **PageRatio** 屬性，代表的是平均

點擊每個產品網站花了多少時間。利用它作成數個區間後，test accuracy 勉強來到了 0.654。從一開始的三成都不到，到最後接近有六成五，雖然還有很大的空間可以做得更好，這個模型的確一步步地在改善。最後以 clasification 報表結束這次的分析以及預測。果然負面樣本的準確率已經可以非常高了。如果我們有更多的有效的正面樣本的話，一定能得到更佳的預測結果。

	precision	recall	f1-score	support
0	0.93	0.95	0.94	3120
1	0.71	0.60	0.65	579
accuracy			0.90	3699
macro avg	0.82	0.78	0.80	3699
weighted avg	0.89	0.90	0.90	3699