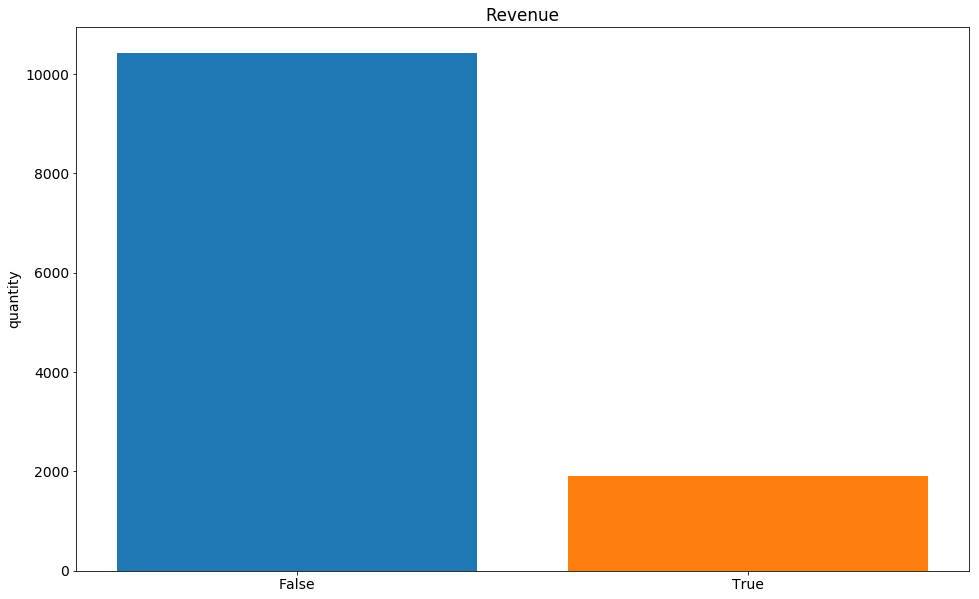
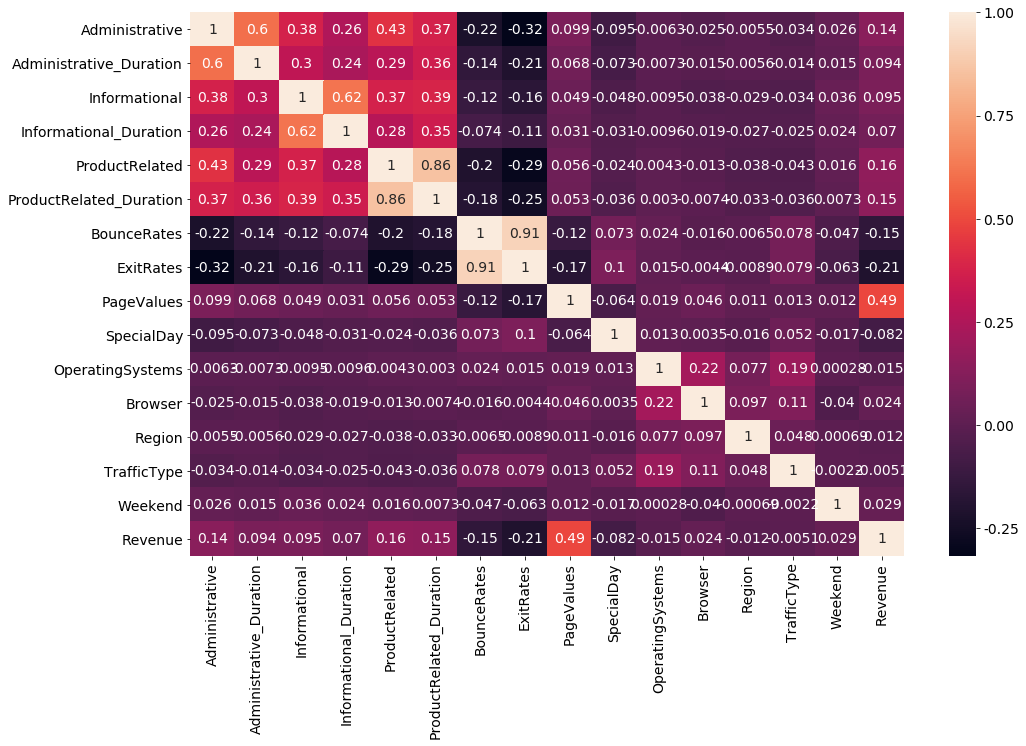
資料: Online Shoppers Purchasing Intention Dataset

目標：從網站瀏覧資料預測用戶是否給公司帶來收益

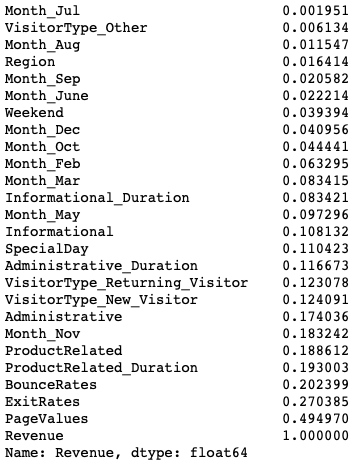
初步分析：

* 有12330筆資料，18項特徵
* 沒有任何空白值
* 二元結果預測
* 從下圖中能觀察出，False的數量比True要多出有5倍，我嘗試過只抽取了部分的False作訓練以得到更好的結果，後來測試後發現不作sampling的結果更好。 
* 觀察heatmap，只看最右邊revenue那行，除了PageValues有明顯相關性外，其他似乎都不太高，相信PageValues對於公司收益起到重要的作用。



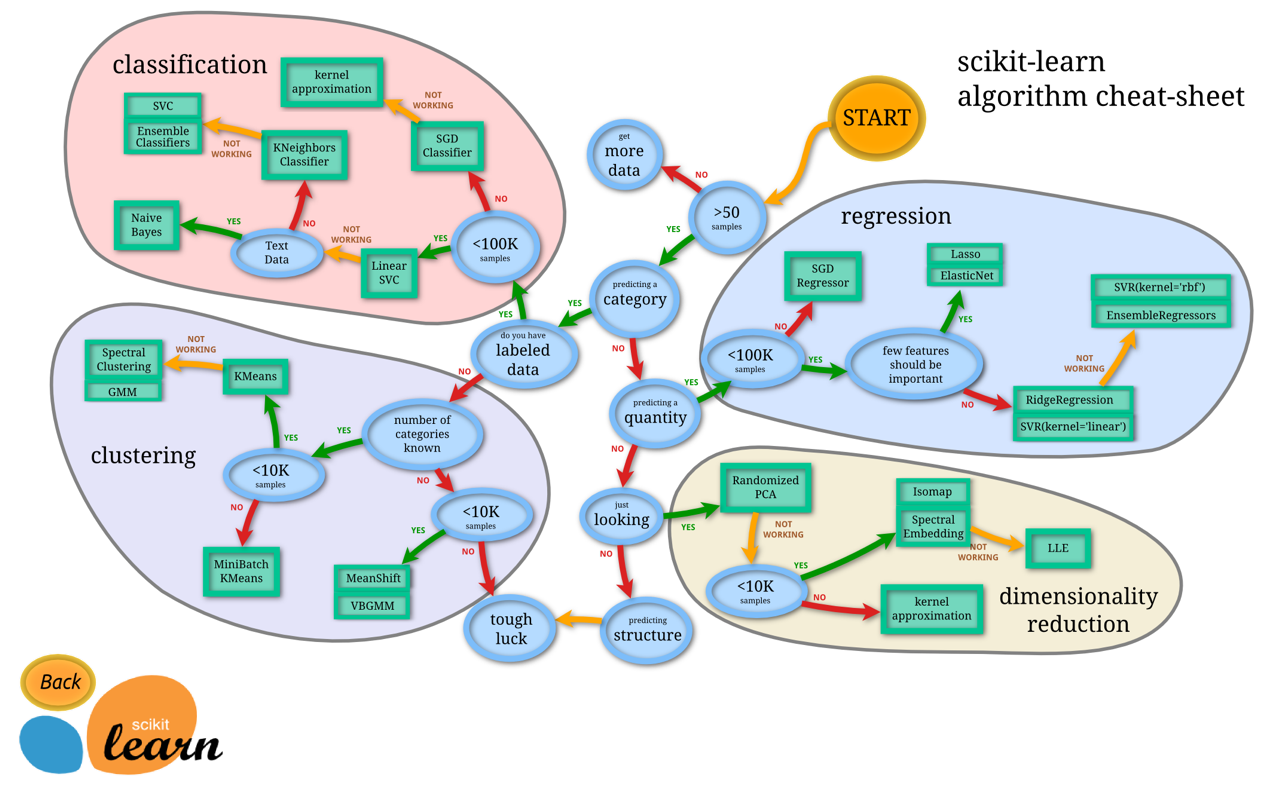
前處理：

* 把"OperatingSystems", "Browser", "TrafficType"三個特徵值拿掉，因為從一般思考來講，這幾項特徵不會對revenue做成影響。
* 對” Month”, “VisitorType”進行one hot encoding，因為”VisitorType”不是數字，需要把它轉換成數字。然後”Month”如果直接改數字的話，因為有12個數字，不是所有數字的距離都相同，會影響結果。
* 把”Weekend”, “Revenue”的True和False換成數字0跟1
* 重新做一次相關性，把相關性絕對值低於0.01的資料拿掉，然後可以觀察到”PageValues”的相關性在資料進行一些後，仍然處於很高的數字。

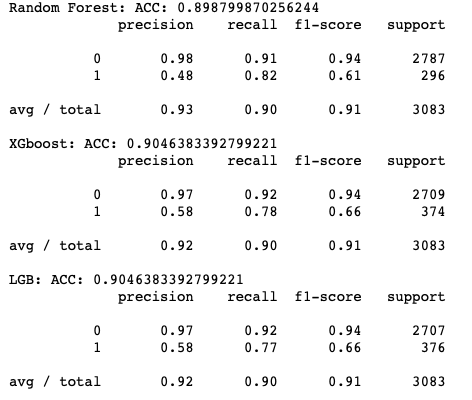


模型訓練：

* 將最後處理完的資料順序打亂
* 以train:test=3:1的比例分割
* 把訓練和測試資料進行標準化，因為有些模型涉及到座標位置的關係，需要把數字統一
* 這次要作的預測屬於classification，有很多模型可以選擇，我根據sklearn官方的cheatsheet先選擇了幾個模型("KNeighbor", "Decision tree", "Random Forest", "SGD", "Logistic Regression", "LinearSVC", "XGboost", "LGB")進行最初步的測試，然後根據F1\_score和accuracy進行選擇，把表現最好最接近的幾個進行下一步測試。



* 上述測試後，觀察到"Random Forest", "XGboost", "LGB”有最好的表現，然後使用gridSearchCV對三個模型以不同的parameter作測試，找出最佳結果的parameter,同時加入K-Fold CV隨機分割以達到更可信的結果。
* 得到最佳parameter後，再把它們代入模型中作一次乾淨版本的比較。
* 下圖為最終結果



改進：

* 我之前曾經遇過結果比例差很多的狀況，一直沒有察覺到，所以我這次一開始就本能反應的看，然後覺得好像差有點多就做了sampling，後來測試後才發現原來不做sampling會更好。
* 忘記做標準化，每次測試結果的落差都很大
* 一開始我是把模試逐個測試，後來才想到可以用迴圈一次過做測試，雖然結果沒有不同，但卻提升了辦事的效率。
* 開頭只在意accuracy，但是precision, recall同樣重要，因為這點又多做了幾次測試。

結論：

從最終結果來看，XGboost和LGB的f1-score和accuracy是接近相同，而Random Forest 在f-score的表現上也是跟其他兩個模型差不多的，雖然accuracy稍為低一點，但是在訓練速度上比其他兩個快不少，所以我認為三個模型都可用。如果是以這次例子的話，資料數不多而且重視準確率的話，我會優先選 LGB，雖然跟XGboost差不多，但是在訓練速度上稍為快一點點。

在選擇模型上，很大程度是需要看你的哪種結果比較重要，舉個例子，像垃圾郵件偵測，寧可偵測不到一部分的垃圾郵件，也不要把正常的郵件誤判，這個時候False-negative比起False-positive應該要更低，這種情況也是判斷的模型的要考量的點。