

91APP

大數據與商業分析

期末專案

第九組

目錄

1

發現問題與預期目標

2

實作 NAPL 分群模型

3

NAPL 轉變與行為資料分析

4

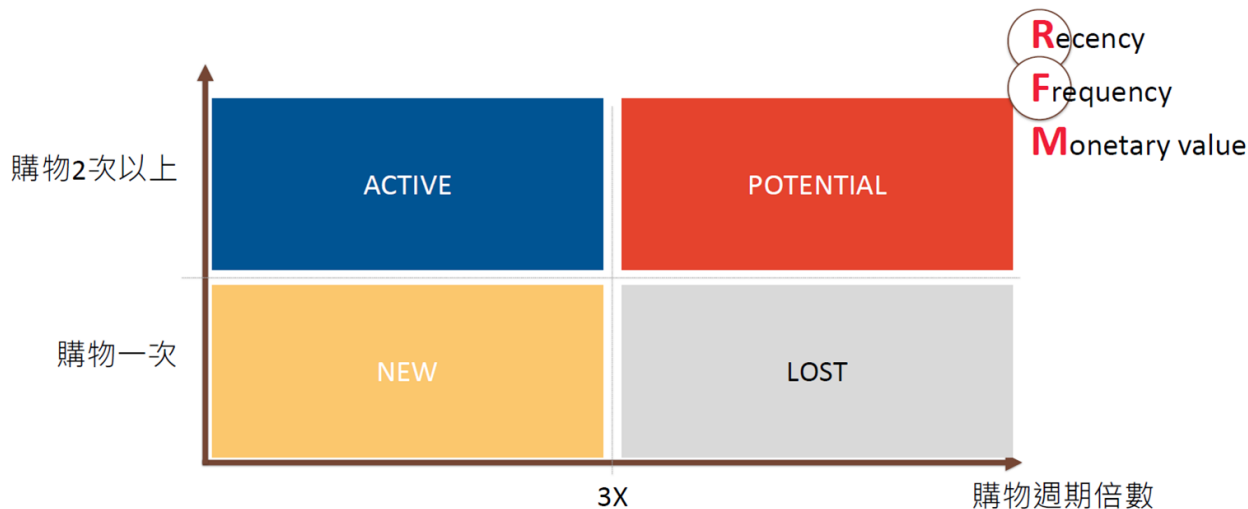
行銷建議

Part 1

發現問題與預期目標

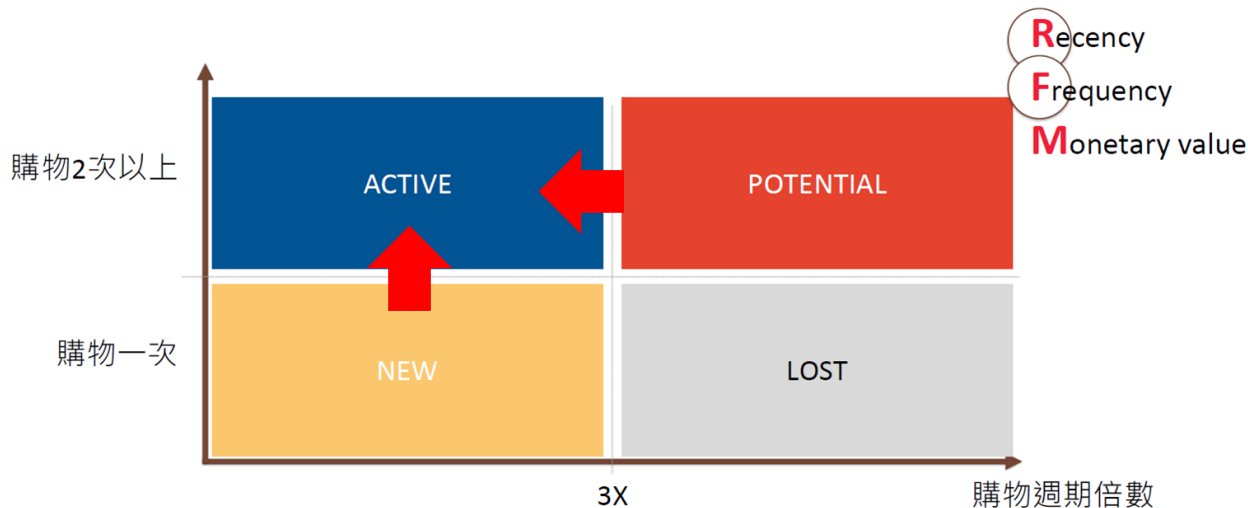
發現問題

- NAPL模型，使用購物週期倍數及購買次數來分群顧客。
- 單純用實際購買行為作為判斷用戶活躍與否的指標，有些不足。
- 族群間的型態轉變，只能觀察到結果，難以推測轉變原因。



預期目標

- 了解用戶在轉換型態之際，會有什麼行為模式的出現。
- 透過行為資料分析，解釋用戶轉換型態的原因。
- 根據分析結果，提出可行建議來幫助、誘使族群間的轉換。



Part 2

實作 NAPL 分群模型

NAPL分群模型 - 目標

- 使用既有的NAPL分群方式。
- 建構出每一天每個id的N、A、P、L狀態。

NAPL分群模型 - 前處理

- 只使用Status = 'Finish' 的 OrderData 。
- 計算全店的購物週期中位數及每個id的購買次數 。

```
purchase_df = purchase_df[df["Status"] == "Finish"]
```


NAPL分群模型 - 購物週期中位數

● function : 計算全店所有購物間隔的中位數。

- 過濾掉只購買一次的顧客，將其他所有購物間隔收集起來並排序。
- 由於是取中位數，過濾掉部分過大或過小的購物週期，不影響結過。

```
def get_median_purchase_period(df):  
    gaps = []  
    member_df = df.groupby(['MemberID'])  
    for memberID, data in member_df:  
        dates = sorted(data['TradesDate'].values)  
        if len(dates) == 1:  
            daysSinceLastOrder = date(2020, 5, 1) - dates[0] #daysSinceLastOrder 計算最後一次下單到現在的週期  
            gaps.append(daysSinceLastOrder.days)  
            continue  
        tmp = [(dates[i+1]-dates[i]).days for i in range(len(dates)-1)]  
        daysSinceLastOrder = (date(2020, 5, 1) - dates[len(dates)-1]).days  
        tmp.append(daysSinceLastOrder)  
        gaps += tmp  
    gaps = [x for x in gaps if x != 0] #移掉同一個人同一天定兩筆訂單的狀況  
    gaps = [x for x in gaps if x <= 365] #移掉超過365天的狀況  
    return statistics.median(gaps)  
# get_median_purchase_period(orderData)
```

NAPL分群模型 - 購買次數

- function : 計算會員每一個時間點從有訂單資料以來的總購物次數。
 - 使用訂單的data計算每一個會員訂了幾筆finished 的訂單。
 - 透過TradesGroupCode計算有完成的訂單筆數。

```
##傳日期可回傳在那個日期當下已前的總購買次數
##如果沒傳某日期以前的話回傳總購買次數
def get_purchase_time(memberID,df,by_time = None):
    member_df = df.groupby("MemberID")
    df = member_df.get_group(memberID).sort_values(by=["TradesDateTime"])
    purchase_df = df[['TradesGroupCode', 'TradesDateTime', "MemberID", "Status"]]
    purchase_df = purchase_df[purchase_df["Status"] == "Finish"]
    #print(purchase_df)
    if by_time:
        purchase_time_by_far = 0
        for date in df["TradesDateTime"]:
            if date.split(" ")[0] <= by_time:
                purchase_time_by_far += 1
        return purchase_time_by_far
    return len(purchase_df) ## return 總購買次數

get_purchase_time("UB%2F0WwXAdW5GhwWmVB036mhSMcUnbK1IA5kXp31h4XQ%3D", df,"2019-09-26")
```

NAPL分群模型 - 實作結果

	%2B%2B44iA8rzzl4lYN3dzParPLqAXetHBSwDFIBUD%2B%2Bt4%3D	%2B%2B%2FsALK1Ndkh585nb%2F9geIrN84FIZZLkD8qf909HvCU%3D
0	X	X
1	X	X
2	X	X
3	X	X
4	X	X
...
2308	N	L
2309	N	L
2310	N	L
2311	N	L
2312	N	L

2313 rows × 181129 columns

橫軸：user_id 縱軸：從開始記錄後 NAPL status 的第幾天

label : N = new/ A = active/ P = potential/ L = lost/ X = no status

Part 3

NAPL 轉變與行為資料分析

NAPL轉變與行為資料分析 - 目標

- 觀察 New/Potential 變成 Active 的行為模式。
- 紀錄轉變前的一個 timewindow (15天)內，在行為資料上有什麼顯著轉變。

NAPL轉變與行為資料分析 - 執行方式

1. 建立好我們要觀察的 feature : 共 21 個
ex. PageViews/ Search times...
2. 建立好從 2018/6/6 => 2020/5/1 的 list (因為從 2018/6/6 才有行為資料)
3. 得到我們要觀察的 user_id (取行為資料最多的前 10000 人)
4. 搜索這些 user 在 user_behavioral 的資料

NAPL轉變與行為資料分析 - 執行方式

5. 針對 feature 計算每日的feature label (這裡以 addToCart 為例)

```
def find_addToCart_1(user_id, csv, get_feature):
    quick_2 = 0
    addToCart_date = []
    for date in date_list:
        count_add = 0
        count_price = 0
        count_items = 0
        # quick_1 = 0
        for row in range(len(user_behave[quick_2:])):
            if(user_behave['time'][row + quick_2] != date):
                break

            elif(user_behave['time'][row + quick_2] == date):
                # quick_1 += 1
                count_add += 1
                if(user_behave['pr1pr'][row + quick_2] >= 0):
                    count_price += user_behave['pr1pr'][row + quick_2]
                if(user_behave['pr1qt'][row + quick_2] >= 0):
                    count_items += user_behave['pr1qt'][row + quick_2]

        quick_2 += count_add
        if(get_feature == 'add'):
            addToCart_date.append()
        elif(get_feature == 'add'):
            addToCart_date.append()
    return addToCart_date
```

NAPL轉變與行為資料分析 - 執行方式

6. 對應到每日的 NAPL status (在這 10000 人內約有 3000 – 4000 人會從 $N \Rightarrow A / P \Rightarrow A$)

```
[194]: len(N_A.groupby("uid")), len(P_A.groupby("uid")), len(N_L.groupby("uid"))
```

```
[194]: (3318, 548, 1452)
```

```
[ ] def GetStateChange(df, init = "N", to = "A"):  
    labels = df.label.values  
    runs = [(labels[0], 0)]  
    seq = [labels[0]]  
    for i in range(1, len(labels)):  
        if labels[i] not in seq:  
            seq.append(labels[i])  
            runs.append((labels[i], i))  
    return runs
```

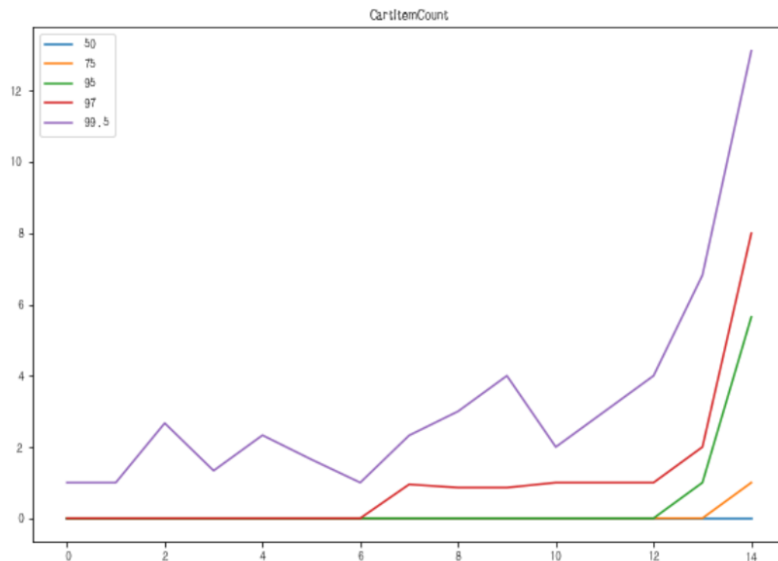

NAPL轉變與行為資料分析 - 執行方式

7. 找出變成 A 的前 15 日在不同 feature 下的變化並觀察

```
def GetLookbackWindow(df, stateChange, uid):  
    lbw = []  
    if len(stateChange) == 1:  
        return lbw  
    global N_A, P_A  
    for ind in range(1, len(stateChange)):  
        if stateChange[ind-1][0] == "N" and stateChange[ind][0] == "A":  
            tmp = df.iloc[ max(state_change[ind][1] - lookback_window, 0): stateChange[ind][1] + 1][:]  
            tmp = pd.DataFrame(tmp)  
            # print(len(tmp))  
            tmp['step'] = [i for i in range(lookback_window-len(tmp), lookback_window)]  
            tmp['uid'] = [uid for i in range(lookback_window-len(tmp), lookback_window)]  
            N_A = pd.concat([N_A, tmp], axis = 0)  
            # break  
        elif stateChange[ind-1][0] == "P" and stateChange[ind][0] == "A":  
            tmp = df.iloc[ max(state_change[ind][1] - lookback_window, 0): stateChange[ind][1]+ 1][:]  
            tmp = pd.DataFrame(tmp)  
            tmp['step'] = [i for i in range(lookback_window-len(tmp), lookback_window)]  
            tmp['uid'] = [uid for i in range(lookback_window-len(tmp), lookback_window)]  
            P_A = pd.concat([P_A, tmp], axis = 0)  
            # break
```

NAPL轉變與行為資料分析 - 執行方式

8. 從每日最多此 feature 行為的各個百分位數標點 (50, 75,)
9. plot: x-時間, y-不同 feature label, 標線為不同百分位



NAPL轉變與行為資料分析 - feature定義

- product, activity, category, main: 瀏覽商品、活動、類別、首頁的次數
- p2a, p2c, p2m ... : 從商品頁跳轉到活動頁的次數、從商品頁跳轉到類別頁的次數，依此類推
- p_stay, a_stay, ... : 商品頁平均停留時間、活動頁平均停留時間
- search_times : 用戶該日有多少 search 行為
- addToCart_times, addToCart_price, addToCart_items : 用戶該日有多少加入購物車行為、該日購物車總價多少、該日有多少商品在購物車
- purchase_times, purchase_price, purchase_items : 用戶該日有多少購物行為、該日購物總價多少、該日購物多少商品

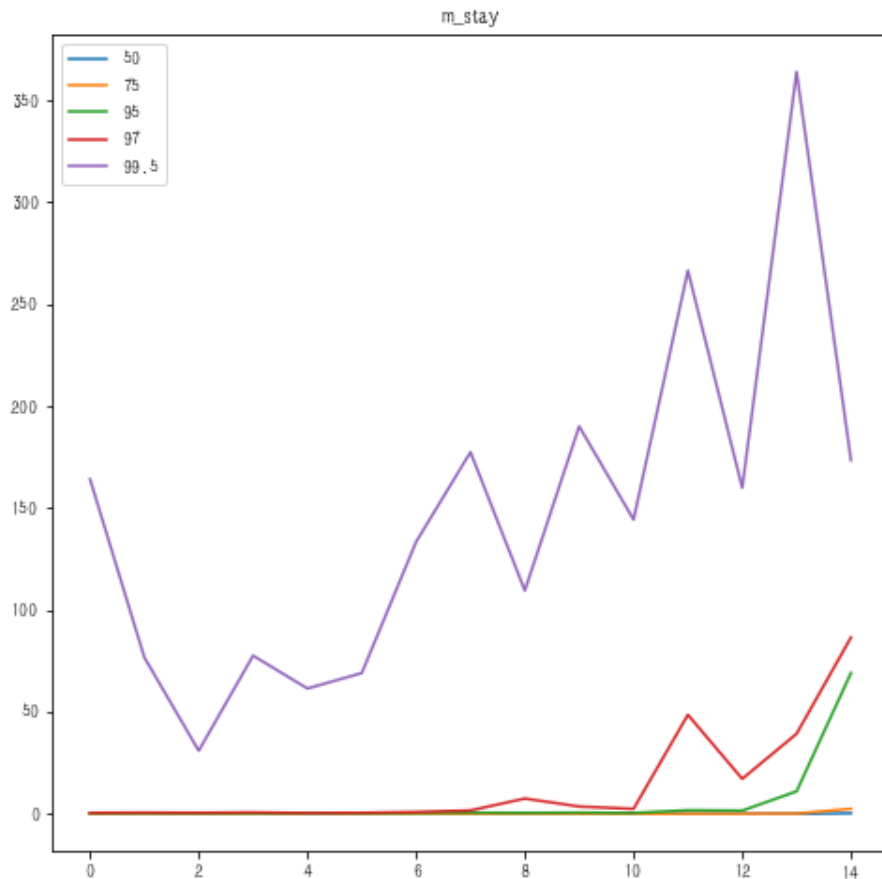
NAPL轉變與行為資料分析 - Potential to Active

- 觀察 NAPL potential status => NAPL active status 有什麼行為發生，舉出幾個較有趣的圖片結果作分析。
- 主要分為：
 - ✓ pageview_stay 相關
 - ✓ pageview1_to_pageview2 相關
 - ✓ pageview_times 相關
 - ✓ 購物車相關

pageview_stay 相關

Main PageView_stay :

大家對於main page沒什麼太大的興趣，除了極活躍的用戶每天逛以外，停留在main page的時間少之又少。

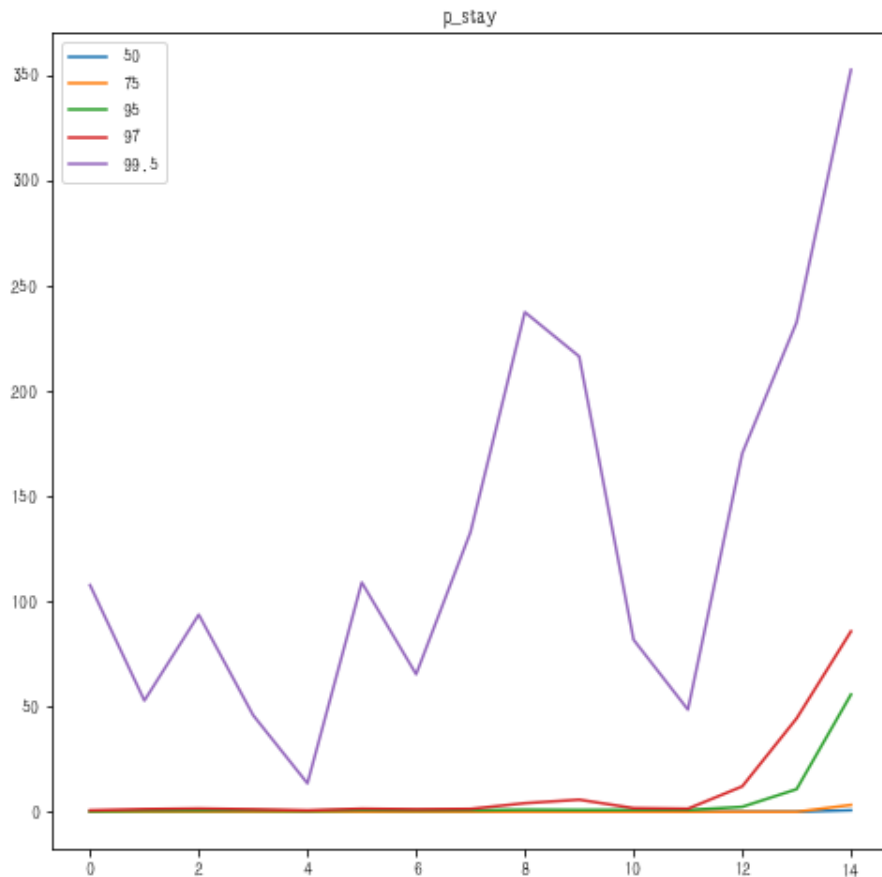


pageview_stay 相關

Product PageView_stay :

意外的是，商品頁本身停留的秒數非常少。除了99.5的用戶以外，

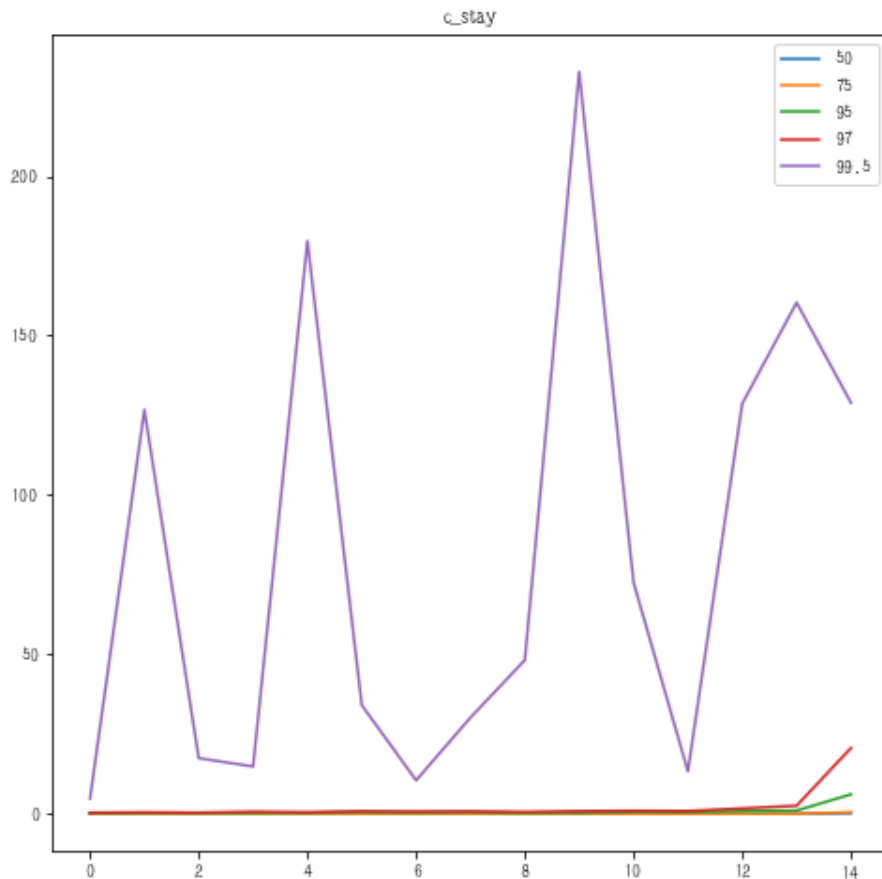
50/75的用戶對於商品頁較無興趣、瀏覽次數也少。



pageview_stay 相關

Category PageView_stay :

最活躍的一批使用者。他們在購買商品、成為A類型會員之前，會時不時地去瀏覽分類頁，代表本身就高度關注各類型商品。

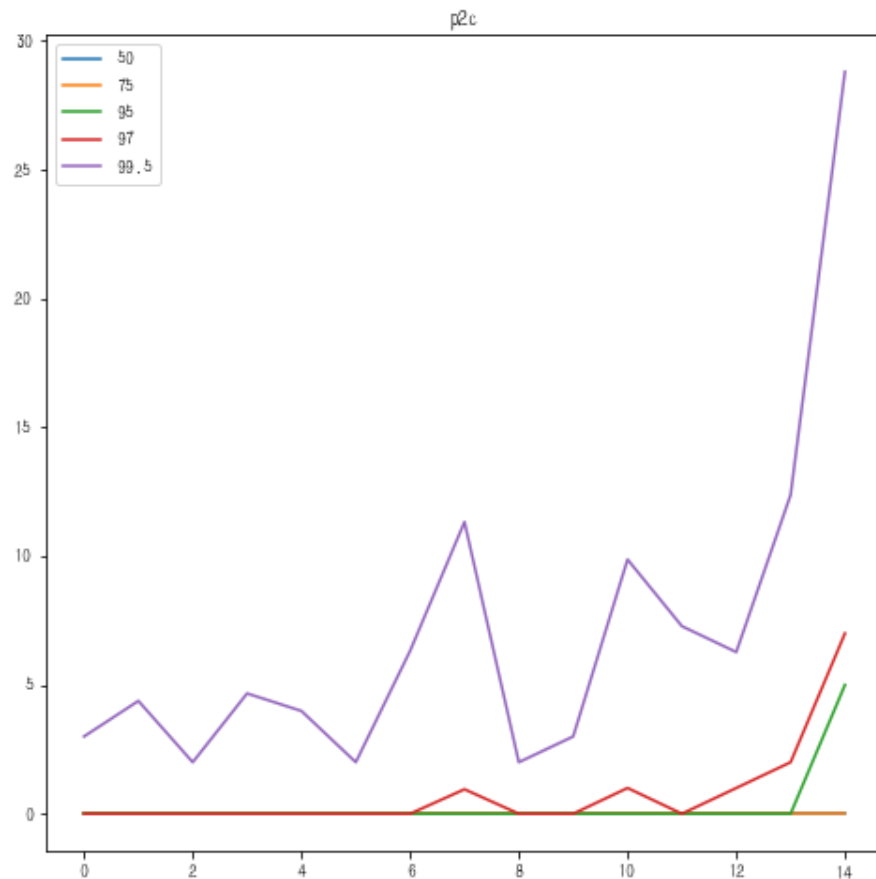


pageview1_to_pageview2 相關

product pageview to category

pageview :

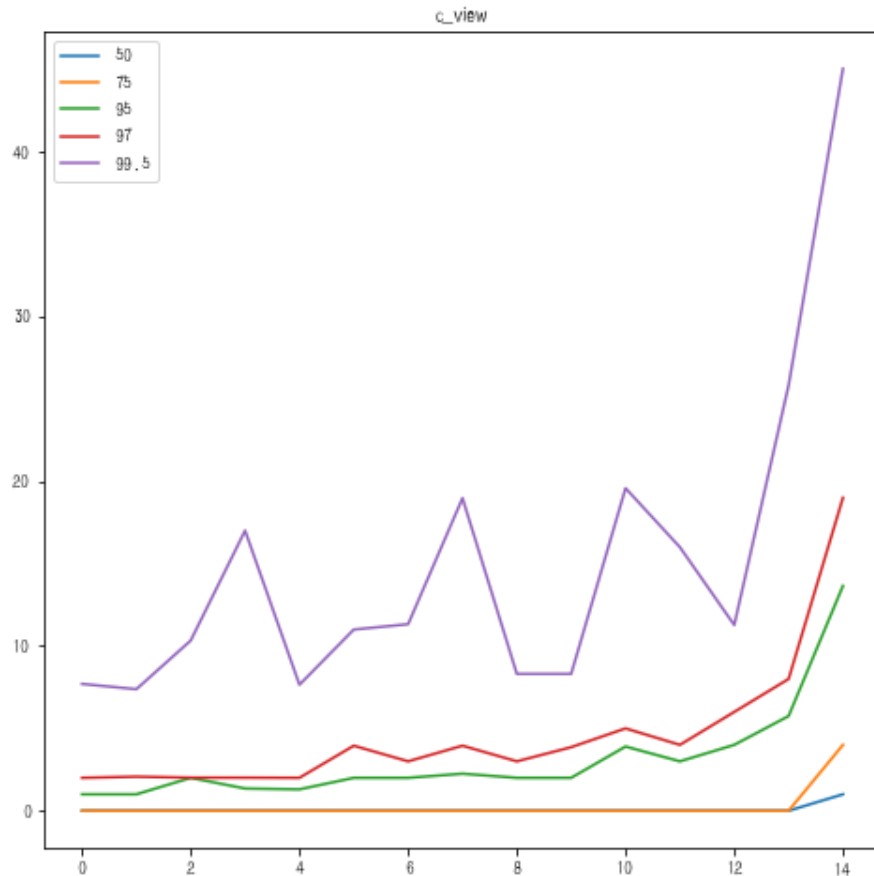
會因為對某個商品品項感興趣，而
去看同類型商品的行為很少。



pageview_times 相關

Category PageView :

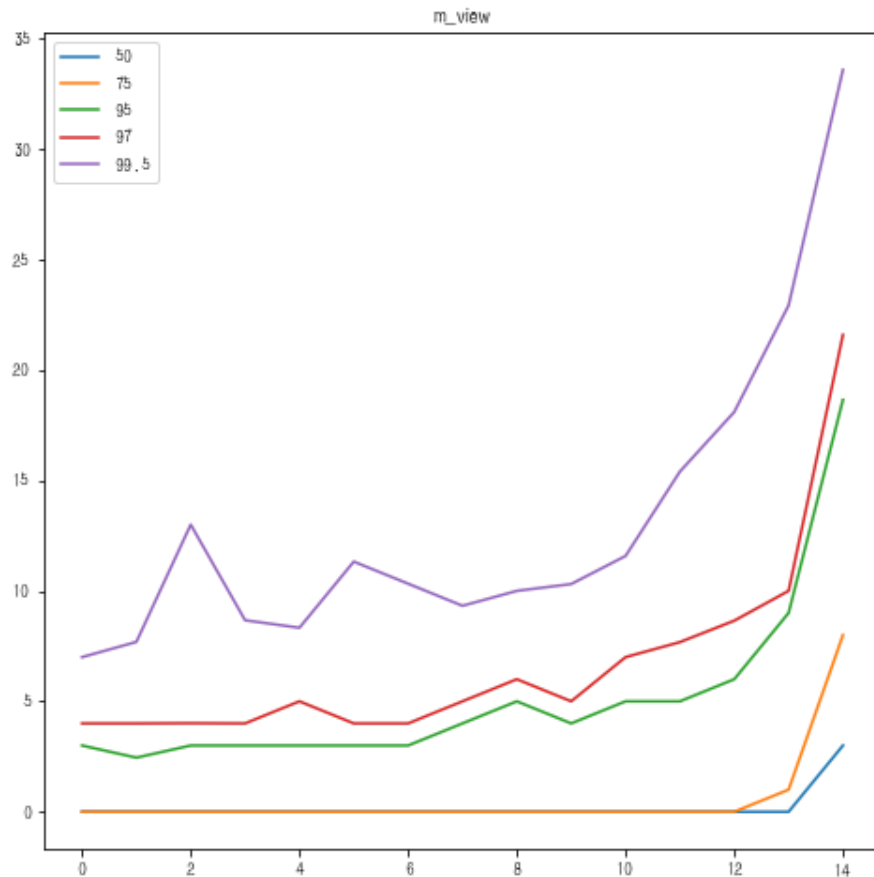
百分位數50/75的用戶在購物週期當中幾乎不瀏覽分類頁面，需要時才進行購買。而90百分位數的會員平時就有不斷在逛各類商品、不斷自我創造需求，燃燒購物慾的興趣，每天瀏覽10-20次左右，這種時候廣告就很重要了，一旦引發購物欲望就可能購買。



pageview_times 相關

Main PageView :

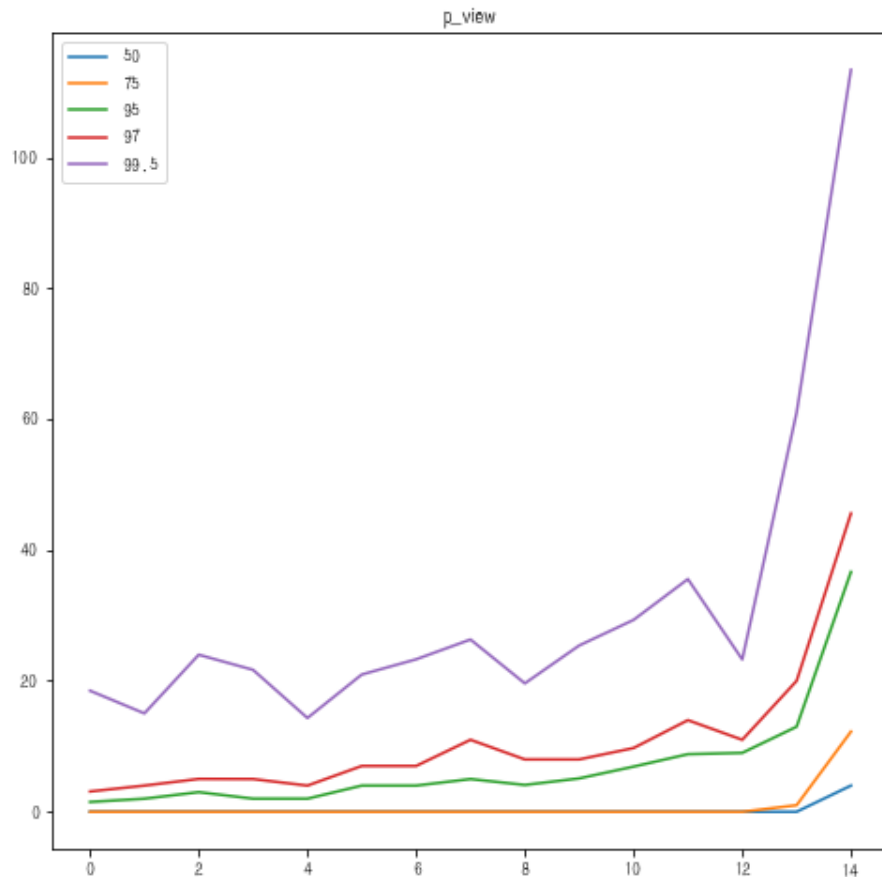
main page的瀏覽次數多，但幾乎都馬上跳走 (觀看的時間少)。可以放一些主題 or 吸引人的廣告頁面來增加顧客進入商品、分類頁面的次數，進而提高購買率。



pageview_times 相關

Product PageView :

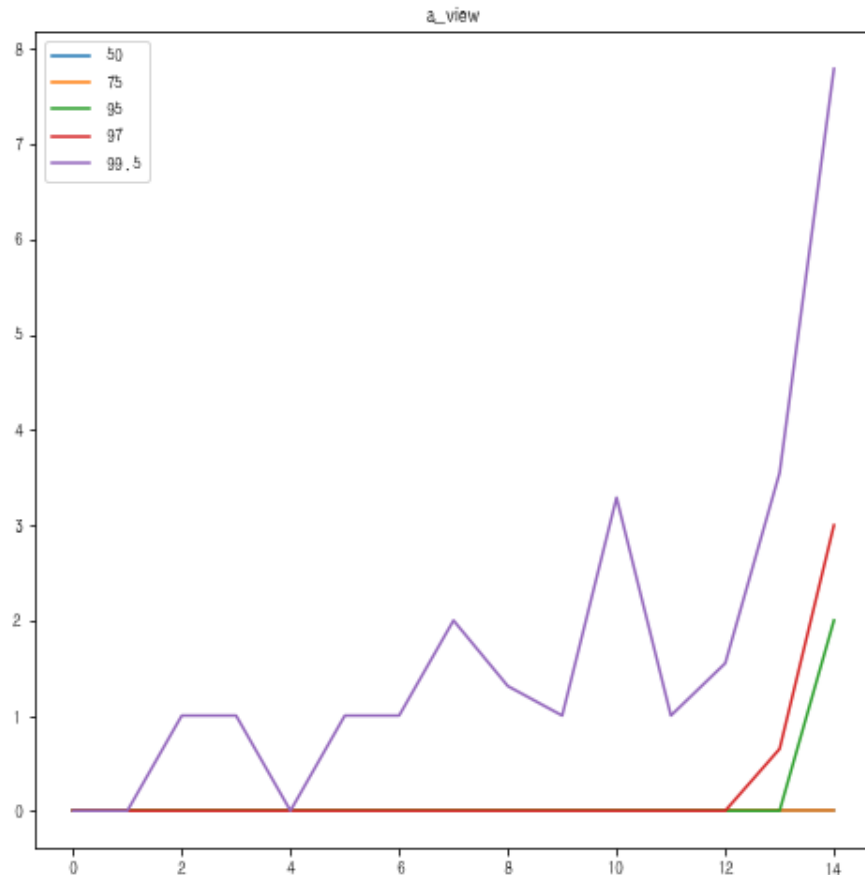
因為準備購買了，所以接近轉變時
瀏覽那個商品本身就會高，不過大
多數的人在購買前不會進入特定商
品頁面，只有忠實粉絲才會去觀察
商品。



pageview_times 相關

Activity PageView :

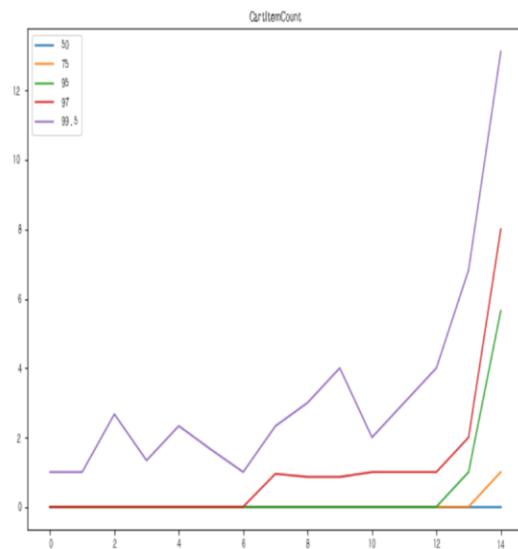
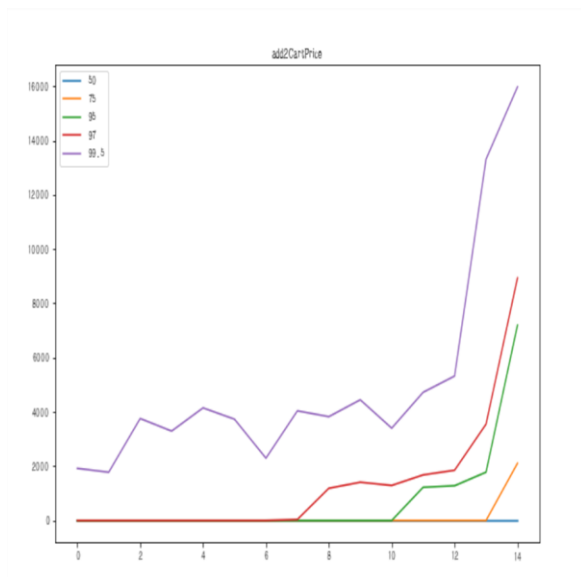
活動頁幾乎吸引不到什麼客戶，可能是商品折扣活動不吸引人或標示不清，也可能是廣告投放策略問題，造成整個廣告、活動沒有太大的造訪率與效益。



購物車相關

add2Cart_Price/ items :

最活躍的用戶，習慣性的在購物時將感興趣的商品加進購物車當中並且反覆瀏覽、調整購物車中的品項。50/75百分位的用戶則是平常沒有一直把商品加入購物車的習慣(把逛網拍當興趣)，僅在需要時購買商品。

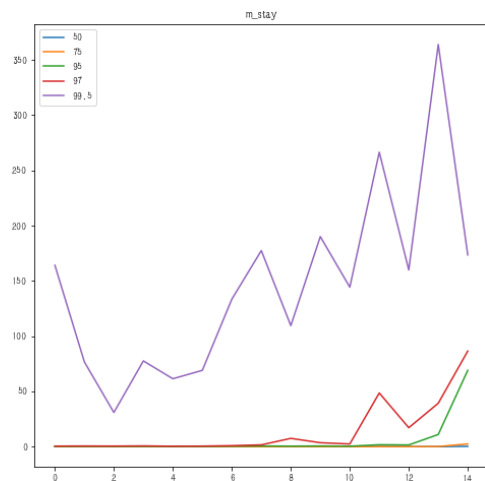
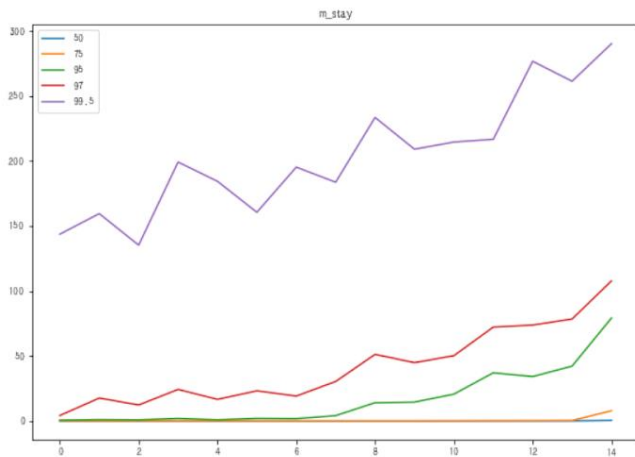


NAPL轉變與行為資料分析 - New to Active

- 觀察 NAPL new status => NAPL active status 有什麼行為發生，舉出幾個較有趣的圖片結果作分析。
- 主要分為：
 - ✓ pageview_stay 相關
 - ✓ pageview1_to_pageview2 相關
 - ✓ pageview_times 相關
 - ✓ 購物車相關

NAPL轉變與行為資料分析 - New to Active

- 此部分只作簡述探討，可以發現趨勢圖與 $P \Rightarrow A$ 的變化差不多，但有趣的事情是 N 用戶在轉變成 A 的過程，他們的 `pageview_times` 與 `stay_times` 大多比 $p \Rightarrow a$ 還要高，推論是因為此店家對它們而言有新鮮感，因此使用時間較多，如何長時間維持吸引力是維持客戶的妙招。(左圖 $n \Rightarrow a$ `main_stay`/ 右圖 $p \Rightarrow a$ `main_stay`)



Part 4

行銷建議

行銷建議 - Potential to Active

- 從行為資料分析結果，發現百分位數90以上的在各種行為模式上較有指標性，能從這些顯著行為中找出一些問題並解決：
 - ✓ 分類頁的停留時間長：針對這些準備回購，卻在眾多商品中難以抉擇的顧客，應試著給予購買指引、推薦商品。
 - ✓ 特定商品頁停留時間長：針對這些有明確目標的顧客，提供其他使用者心得、更多產品介紹、或甚至是特別優惠，促進導購行為。
 - ✓ 活動頁面造訪次數低：這個背後原因很複雜，需要進一步研究。但是可見在廣告投放、活動設計上，需要更精準的打中目標客群。

行銷建議 - New to Active

- 從行為資料分析結果，發現百分位數90以上的在各種行為模式上較有指標性，能從這些顯著行為中找出一些問題並解決：
- ✓ 瀏覽時間與分頁數量較多：這項數據可以說明，new users 對於網站、品牌還不太熟悉，且需要反覆思考、比較。針對此類顧客，提供明確指引與推薦商品，可以快速過濾掉雜訊，直接讓顧客進入考量購買的環節。另外，由於 new users 只有一次購買經驗，適當提供較優惠的回購折價券也是一個方法。

影片連結、完整程式碼、完整行為資料圖

- 影片連結

<https://youtu.be/bw0Zg6RC5wk>

- 完整程式碼

https://drive.google.com/drive/folders/1zXnVMq27H6WoM_A5RWJX6Rr7WDWHpWvM?fbclid=IwAR2U0ZT8bBn2ZHptTA89_eJ3JXrRtFmTSGIMjee5vVfzfaXSHmDbjDTDURk

- 完整行為資料圖

<https://drive.google.com/drive/folders/1sNijszyZv518bijbF4VgLyixRwCnCwnr?fbclid=IwAR3DCFUXxV1FLKQfpQn5NilHy-c-XL2YvEiJIThIPIPU4LiGAqf48EUefsQ>