91App

顧客消費行為預測與折扣優惠影響之探討

B04310036 余京儒
B04704016 林家毅
B04704017 周永堂
R06323061 詹苡晴

Tabel of Contents

Part I

探索式資料分析 (EDA)

探索式資料分析 (EDA) - 資料清理

1. 篩選有效訂單
   - 為主商品
   - 非贈品
   - 已完成的訂單
2. 清除錯誤值
   - 城市："<fo", "1st", "Tao"……
3. 依照下單時間重新排序

原資料集：5,495,276 筆商品訂單資料，有效資料集：4,080,742 筆商品訂單資料

探索式資料分析 (EDA) - 資料清理

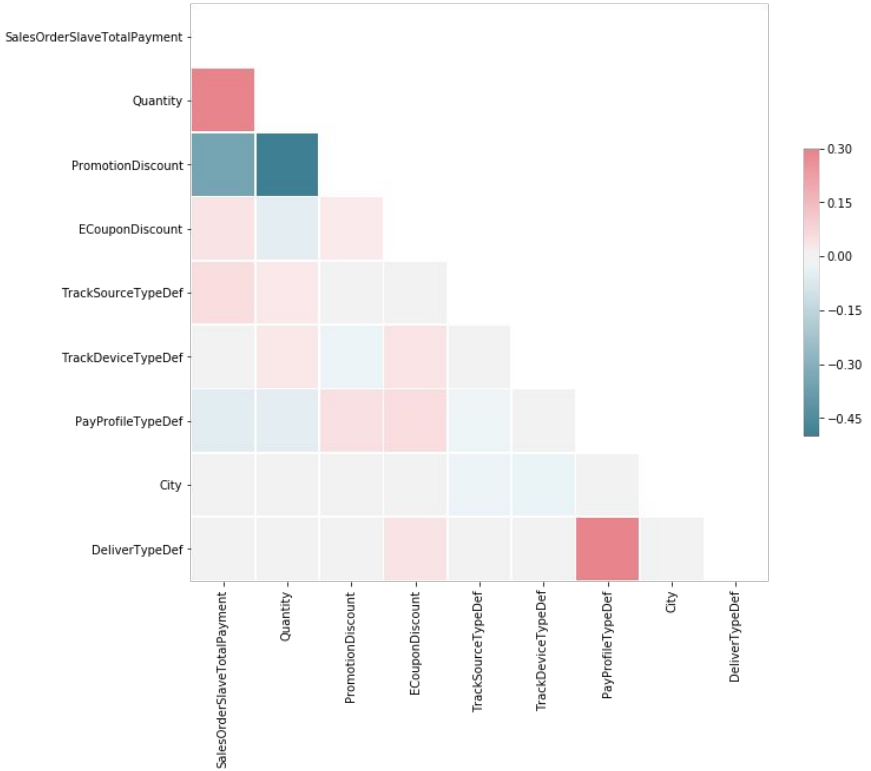
1. 將有效訂單以購物車為單位合併
    - 各商品價格加總
    - 各商品數量加總
    - 紀錄購物車使用的折扣代碼
    - 折扣金額加總
2. 去除極端值
    - 購物車總價格超過 5,000 元
    - 購物車商品數量超過 12 格
3. 依照下單時間重新排序

將 4,080,742 筆有效商品訂單合併為 1,024,634 筆購物車訂單

探索式資料分析 (EDA) - 視覺化

共變異數矩陣

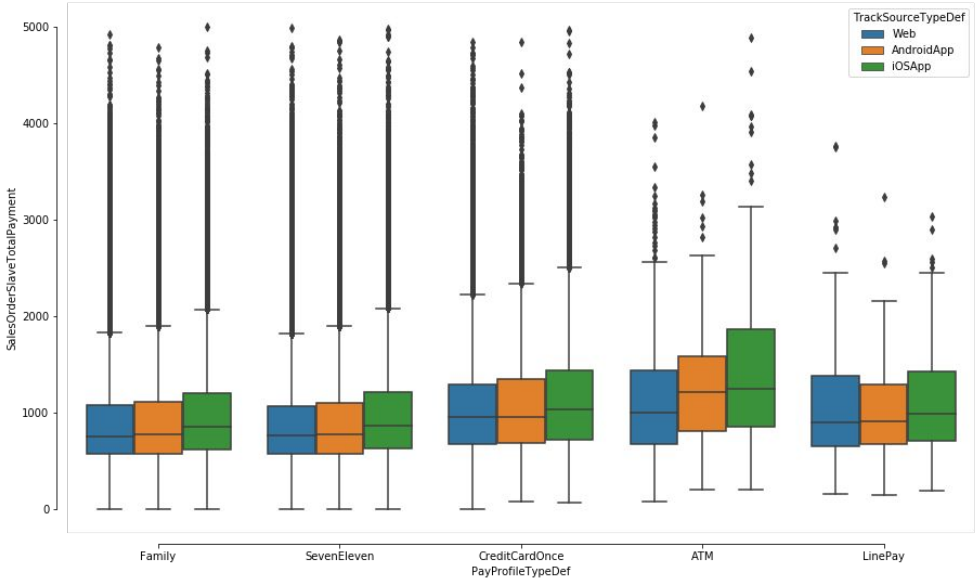
1. 總價格 SalesOrderSlaveTotalPayment
2. 總數量 Quantity
3. 促銷折扣金額 PromotionDiscount
4. 優惠券折扣金額 ECouponDiscount
5. 使用平台 TrackSourceTypeDef
6. 使用裝置 TrackDeviceTypeDef
7. 付款方式 PayProfileTypeDef
8. 城市 City
9. 取貨方式 DeliverTypeDef



探索式資料分析(EDA) - 視覺化

不同付款方式與不同**使用平台**，與**總價格**的關係：

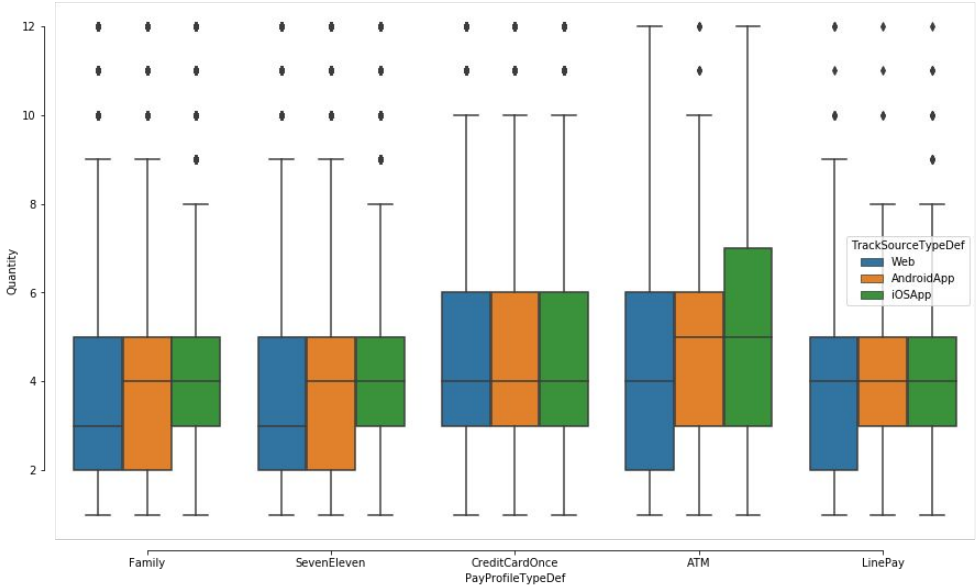
1. 使用iOSApp消費的顧客每次購物的金額大於其他平台使用者，AndroidApp次之，Web居末
2. 高總價的訂單與ATM付款呈現相關性
3. 總價格比較: ATM > CreditCard > LinePay > SevenEleven = Family



探索式資料分析 (EDA) - 視覺化

不同付款方式與不同**使用平台**，與**總數量**的關係：

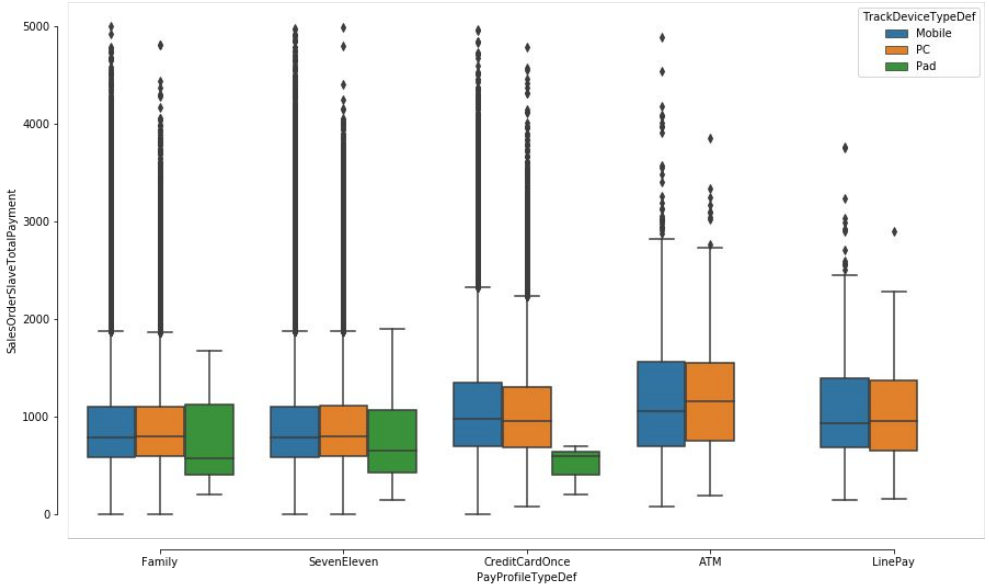
1. 不同平台的顧客在購買商品數量的差異較不明顯，使用 iOSApp消費的顧客購買數量稍多於其他平台使用者
2. 多商品數量的訂單與 ATM付款呈現相關性
3. 總商品數量比較：ATM > CreditCard > LinePay > SevenEleven = Family



探索式資料分析 (EDA) - 視覺化

不同付款方式與不同**使用裝置**，與**總價格**的關係：

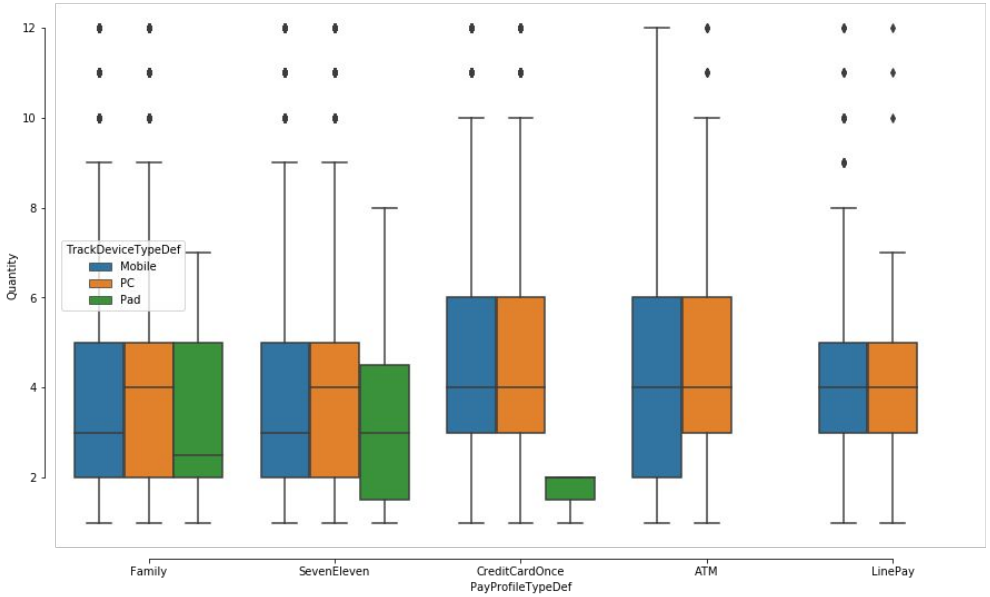
1. 使用Mobile與PC裝置的顧客在購買總價格的差異較不明顯
2. 使用Pad消費的顧客較少、購買總金額也較低
3. 高總價的訂單與ATM付款呈現相關性
4. 總價格比較: ATM > CreditCard > LinePay > SevenEleven = Family



探索式資料分析 (EDA) - 視覺化

不同付款方式與不同**使用裝置**，與**總數量**的關係：

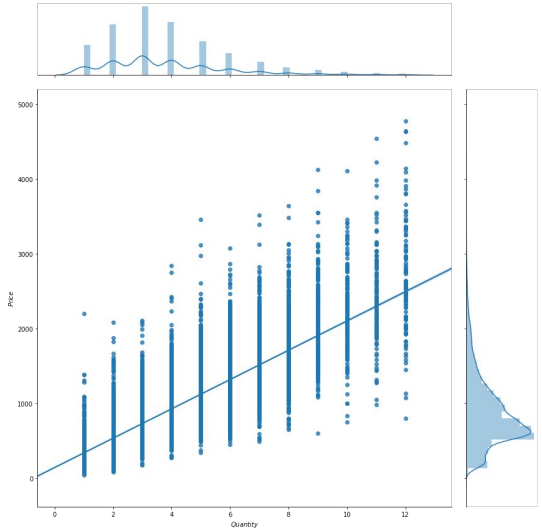
1. 使用PC裝置的顧客在購買商品的總數量略多於使用Mobile裝置的顧客
2. 使用Pad消費的顧客較少、購買總數量也較少
3. 多商品數量的訂單與 CreditCard付款呈現相關性
4. 總商品數量比較：CreditCard > ATM > LinePay > SevenEleven = Family



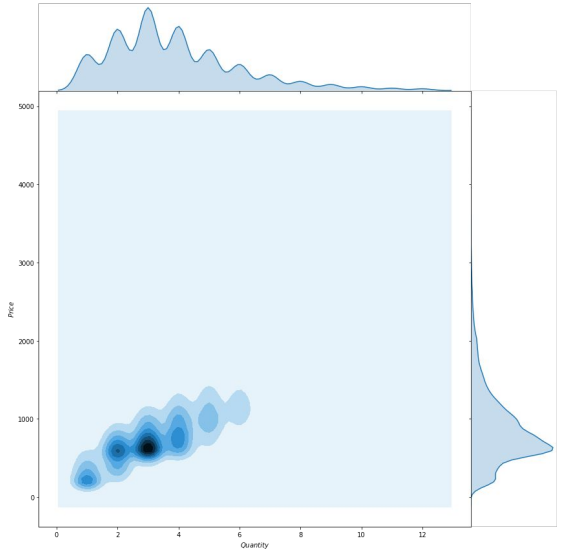
探索式資料分析(EDA) - 視覺化

總價格與**總數量**的關係

1. 依據回歸直線判斷，商品數量每增加一個，約可預期總價格增加 300元



2. 依據分布，大部分訂單的商品數量約為2~4個，總價格約為800元



探索式資料分析 (EDA) - 視覺化

每月總營收

https://plot.ly/~jacky920236/23/revenue-by-month

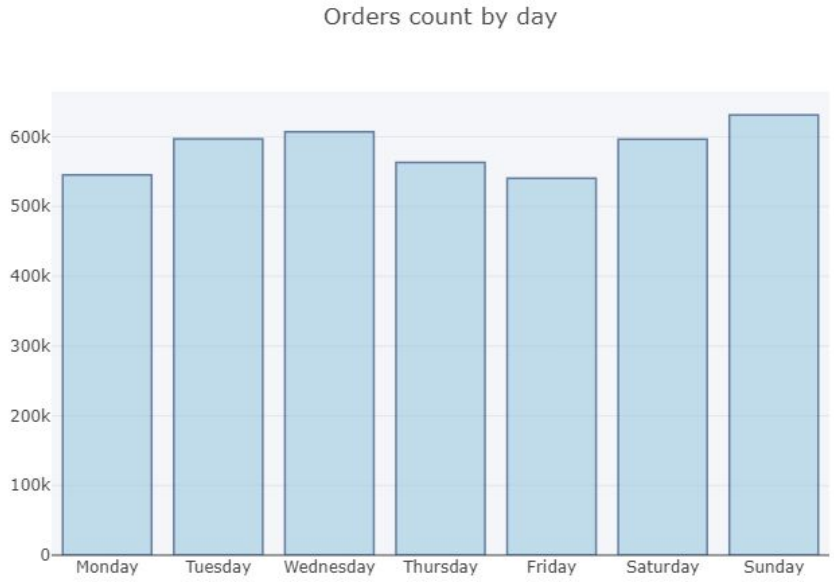
1. 以一年為週期，可以發現每年 4, 5, 11, 12 月是商城旺季，1, 2 月是商城淡季
2. 歷史最高銷售額紀錄分別出現在 2016 年 11 月和 2017 年 11 月

探索式資料分析 (EDA) - 視覺化

週間訂單數

https://plot.ly/~jacky920236/25/orders-count-by-day

1. 週日、週三是一週內訂單數量較多的兩天
2. 週五訂單數量最少



Part II

透過顧客交易紀錄，預測該顧客未來屬性

透過顧客交易紀錄，預測該顧客未來屬性

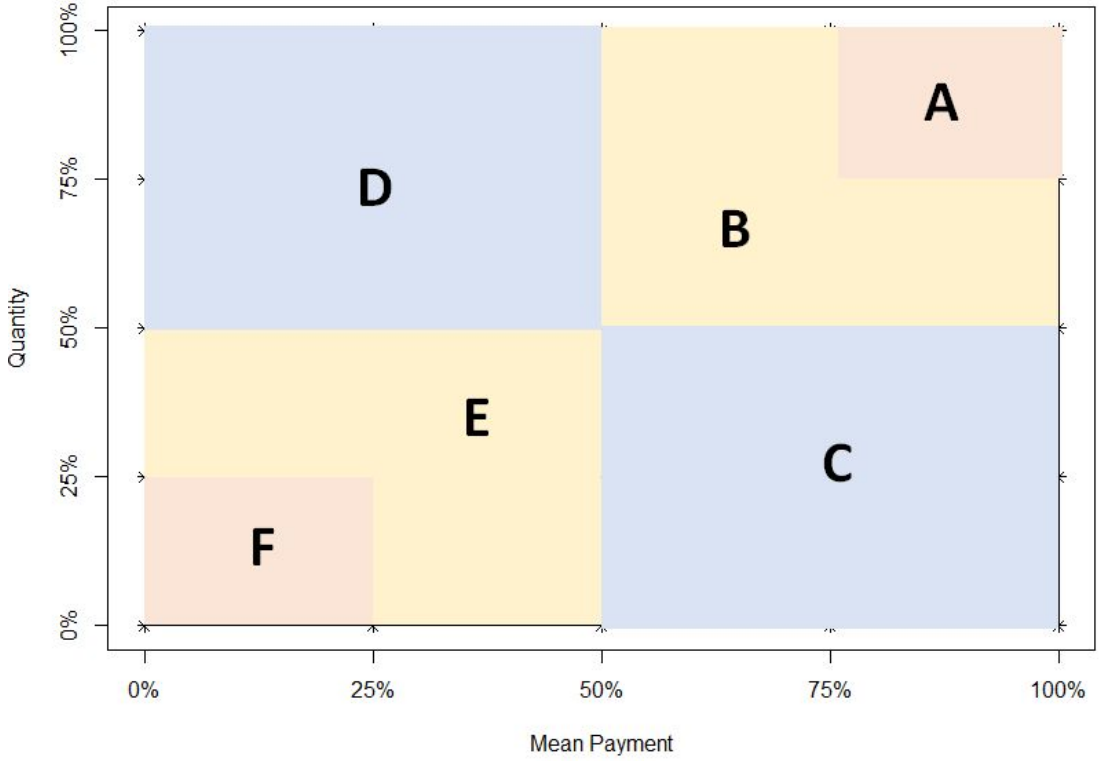
- 若我們能知道顧客的歷史交易紀錄，是否可以進一步了解顧客的分類呢？
- 在這個段落中，我們希望回答以下問題：
  - 根據會員歷史交易紀錄給定一個等級，是否當交易次數越多越逼近該等級？
  - 今日倘若有一個新顧客，是否可以透過其 **第一筆交易紀錄**，了解其未來顧客等級呢？

透過顧客交易紀錄，預測該顧客未來屬性

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4080792 entries, 0 to 5495275
Data columns (total 29 columns):
DateId                          int64
MemberId                        int64
OrderGroupCode                  object
TrackSourceTypeDef              object
TrackDeviceTypeDef              object
PayProfileTypeDef               object
SalesOrderSlaveId               int64
SalePageId                      int64
IsMajor                         bool
IsGift                          bool
IsSalePageGift                  bool
Quantity                        int64
UnitPrice                       float64
PromotionDiscount               float64
ECouponId                       int64
ECouponDiscount                 float64
SalesOrderSlaveTotalPayment     float64
SalesOrderSlaveDateTime         object
SalesOrderReceiverId            int64
City                            object
District                        object
ZipCode                         float64
StoreName                       object
StoreId                         object
DeliverTypeDef                  object
StatusDef                       object
ReturnGoodsDateId               float64
CauseDef                        object
Mean_Payment                    float64
dtypes: bool(3), float64(7), int64(7), object(12)
memory usage: 1012.3+ MB
```

- OrderGroupCode：購物車編號
- IsMajor / IsGift：是否為贈品
- CauseDef：退貨原因
- Quantity：該商品購買數量
- UnitPrice：商品單價
- SalesOrderSlaveTotalPayment：該商品總額

透過顧客交易紀錄，預測該顧客未來屬性



透過顧客交易紀錄，預測該顧客未來屬性

	MemberId	Count	God	1st_rank	2nd_rank	3nd_rank	4th_rank	5th_rank	6th_rank	7th_rank	8th_rank	9th_rank	10th_rank
12	1374116	11	D	D	D	D	E	D	D	D	F	D	D
33	1207723	23	D	D	D	E	D	F	F	E	E	E	E
101	1171712	17	B	B	C	B	B	B	B	C	B	B	A
109	1443639	14	B	C	E	C	C	C	E	E	C	E	E
123	436800	27	B	C	D	C	C	C	C	C	E	C	C
135	1225562	44	D	D	E	E	F	F	D	F	F	E	F
136	1276894	22	D	D	D	E	E	E	D	E	D	E	D
144	932241	10	D	E	E	E	D	F	E	E	E	E	E
146	1281330	10	D	F	F	C	C	F	F	F	F	F	F
181	417910	11	A	C	C	C	C	C	C	C	E	C	C
183	229250	10	D	E	C	F	E	F	F	F	F	E	E
189	1415460	10	A	C	C	C	C	C	C	C	C	C	B
202	1415692	11	B	C	B	B	B	B	C	B	C	C	C
234	1416309	10	D	C	C	C	F	F	E	C	E	F	F
272	214551	20	D	D	D	D	D	D	D	D	D	D	D
277	988063	11	B	B	C	C	E	E	C	C	C	C	E
294	302699	14	D	D	C	D	E	E	E	D	E	D	E
322	1702196	18	D	D	E	F	E	D	D	D	D	D	D
331	1733224	17	D	E	F	D	D	E	F	F	E	E	E
346	1719797	12	B	C	C	C	C	C	C	C	C	C	C
359	1450517	18	B	B	B	C	B	B	B	B	B	B	B
370	1542811	11	A	D	C	E	F	B	D	E	B	E	D

- God：歷史資料給予的顧客等級
- Count：此消費者總共購買的次數
- nth_rank：所有消費者的前N筆紀錄計算出的Rank

透過顧客交易紀錄，預測該顧客未來屬性

- 單純使用歷史資料來看，單筆的消費資料並不能準確預測其真正的等級(God)
- 若僅用前N筆購買的商品價位與數量，命中率僅僅只有25% ~ 30%
- How can we do better?

```
temp = df_all_rank[df_all_rank['10th_rank'].isin('A B C D E F'.split(' '))]
sc = []
for i in range(10):
    temp['temp'] = (temp['God'] == temp.iloc[:,i+2])
    score = round(temp['temp'].sum() / temp.shape[0],4)
    sc.append(score)
sc
```

```
[1.0, 0.2811, 0.292, 0.2674, 0.2809, 0.2631, 0.2729, 0.2692, 0.2804, 0.2804]
```

透過顧客交易紀錄，預測該顧客未來屬性

- 模型選擇：Logistic Regression / Random Forest / XGBoost
- 特徵解釋：
  - Mean_Payment：平均單一商品價格
  - Quantity：前N筆總購買數量
  - SalesOrderSlaveTotalPayment：經折扣後前N筆總價
  - UnitPriceGroup_x：前N-1筆共M件商品的價格分布
  - UnitPriceGroup_y：第N筆共K件商品的價格分布
  - ECoupon：有無使用Ecoupon (Dummy)
  - North ~ DeliverTypeDef：該筆訂單購物資料 (Dummy)

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 491834 entries, 0 to 491833
Data columns (total 26 columns):
God                                491834 non-null object
MemberId                           491834 non-null int32
Mean_Payment                       491834 non-null float64
Quantity                           491834 non-null int64
SalesOrderSlaveTotalPayment        491834 non-null float64
UnitPriceGroup_(0.0, 160.0_        491834 non-null uint8
UnitPriceGroup_(160.0, 320.0_      491834 non-null uint8
UnitPriceGroup_(320.0, 480.0_      491834 non-null uint8
UnitPriceGroup_(480.0, 640.0_      491834 non-null uint8
UnitPriceGroup_(640.0, 3500.0_     491834 non-null uint8
ECoupon                            491834 non-null int32
North                              491834 non-null int32
Mid                                491834 non-null int32
South                              491834 non-null int32
East                               491834 non-null int32
TrackSourceTypeDef_Web             491834 non-null uint8
TrackSourceTypeDef_iOSApp          491834 non-null uint8
TrackDeviceTypeDef_PC              491834 non-null uint8
TrackDeviceTypeDef_Pad             491834 non-null uint8
PayProfileTypeDef_CreditCardOnce   491834 non-null uint8
PayProfileTypeDef_Family           491834 non-null uint8
PayProfileTypeDef_LinePay          491834 non-null uint8
PayProfileTypeDef_SevenEleven      491834 non-null uint8
DeliverTypeDef_Home                491834 non-null uint8
DeliverTypeDef_SevenEleven         491834 non-null uint8
Count                              491834 non-null int64
dtypes: float64(2), int32(6), int64(2), object(1), uint8(15)
memory usage: 40.8+ MB
```

透過顧客交易紀錄，預測該顧客未來屬性

- 使用三種算法得到結果如下：
  - 以精準度言，XGboost 表現最為出色，但演算所需時間也較長
  - 綜合時間與準確度，Random Forest 為此案例中較好的算法
  - 根據下方圖表我們可以發現「頭兩筆資料」可以得到最準的預測

	Model	Score1	Score2	Score3	Score4	Score5	Score6	Score7	Score8	Score9	Score10
2	XGboost Classifier	0.703716	0.756014	0.684862	0.651353	0.618594	0.617642	0.610494	0.598008	0.585342	0.613046
1	Random Forest Classifier	0.630293	0.706697	0.627303	0.597864	0.576133	0.564972	0.553890	0.557453	0.554966	0.553515
0	Logistic Regression	0.601340	0.689882	0.662833	0.637138	0.623174	0.612721	0.597570	0.589826	0.592575	0.601013

透過顧客交易紀錄，預測該顧客未來屬性

- 分類演算法中，若我們能愈肯定某一種解，則此算法效率相對越高
  - 試想若兩種可能性機率分別是 [0.5,0.5]，另外一種是[0.7,0.3]，以分類目的而言後者較佳
  - 我們如何取得此比例呢？計算各自機率的 Variance或許是一個好辦法！
  - Variance越大，代表機率越偏重在某一個等級中，此表現以 Random Forest最佳

	Logreg_accy	RFC_accy	XGB_accy	Logreg_var	RFC_var	XGB_var
1	0.601340	0.630293	0.703716	0.039303	0.083617	0.064479
2	0.689882	0.706697	0.756014	0.054548	0.087755	0.077426
3	0.662833	0.627303	0.684862	0.055859	0.077219	0.067494
4	0.637138	0.597864	0.651353	0.056415	0.080330	0.068033
5	0.623174	0.576133	0.618594	0.054270	0.076945	0.065374
6	0.612721	0.564972	0.617642	0.044359	0.080207	0.058733
7	0.597570	0.553890	0.610494	0.040170	0.073878	0.054145
8	0.589826	0.557453	0.598008	0.039712	0.073219	0.053702
9	0.592575	0.554966	0.585342	0.040968	0.072250	0.052936
10	0.601013	0.553515	0.613046	0.038627	0.074177	0.051204

透過顧客交易紀錄，預測該顧客未來屬性

- 我們來看看頭兩筆交易紀錄表現如何

```
def accuracy_test(dlist):
    global pred_accy, ans_sheet_list, clre_list, feature_importance
    ans_sheet_list = []
    pred_accy = []
    clre_list = []
    feature_importance = []
    for i in dlist:
        pred, accy , prob , rfe = fit_classification_model_with_rfe(RandomForestClassifier(), i,
                                                                    do_prob = False, do_rfe=False)
        test_pred = mod.predict(val[list(val.columns[2:])])
        clre = classification_report(ans, test_pred)
        clre_list.append(clre)

        m = mod.feature_importances_
        feature_importance.append(m)

        ans_sheet = pd.DataFrame(ans)
        ans_sheet['Prediciton'] = test_pred
        ans_sheet.columns = ['God','Prediction']
        ans_sheet['Compare'] = (ans_sheet['God'] == ans_sheet['Prediction'])
        ans_sheet_list.append(ans_sheet)

        accy = ans_sheet['Compare'].sum() / ans_sheet.shape[0]
        pred_accy.append(accy)
```

透過顧客交易紀錄，預測該顧客未來屬性

- 我們來看看頭兩筆交易紀錄表現如何

```
In [96]: ans_sheet_list[1].head()
```

Out[96]:

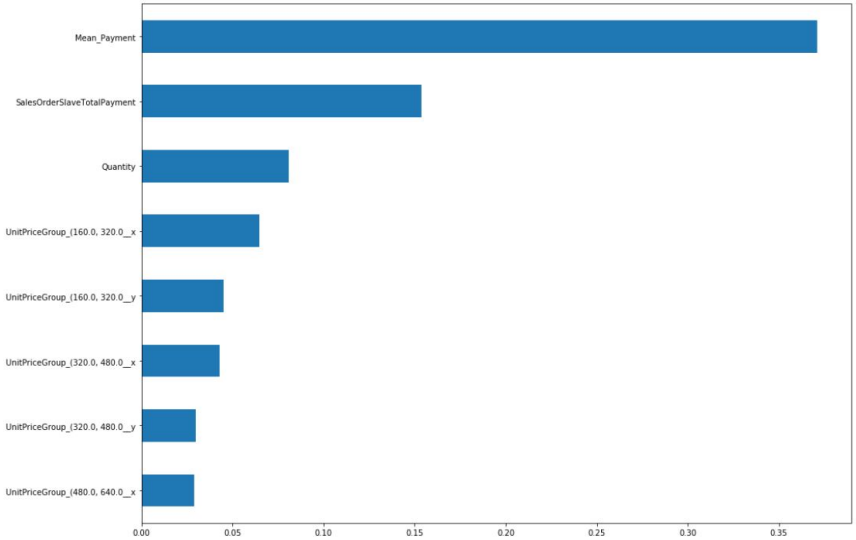
	God	Prediction	Compare
0	D	D	True
1	A	A	True
2	B	B	True
3	D	D	True
4	D	D	True

```
In [95]: ans_sheet_list[1]['Compare'].sum() / ans_sheet_list[1].shape[0]
```

```
Out[95]: 0.6847603683009365
```

透過顧客交易紀錄，預測該顧客未來屬性

- 我們使用RFE來了解Feature的重要性
  - 每個演算法的RFE結果都不同，須注意所挑選的演算法



	Mean_Payment	ECoupon	Quantity
Logreg	20.8	3.200000	13.200000
rfc	1.0	13.900000	5.000000
xgb	1.0	4.100000	5.100000
Mean	7.6	7.066667	7.766667

透過顧客交易紀錄，預測該顧客未來屬性

- Classification Report for df_2nd

	precision	recall	f1-score	support
A	0.43	0.26	0.33	3151
B	0.64	0.62	0.63	16503
C	0.47	0.62	0.54	4498
D	0.82	0.81	0.82	24149
E	0.42	0.51	0.46	2144
F	0.41	0.52	0.46	383
accuracy			0.68	50828
macro avg	0.53	0.56	0.54	50828
weighted avg	0.69	0.68	0.68	50828

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

透過顧客交易紀錄，預測該顧客未來屬性

- Takeaway
  - 根據機器學習結果，當一位顧客來消費第二次時，透過頭兩筆消費可以預測其未來對本公司的總貢獻，精準度達 70%
  - 綜合效率與精準度，Random Forest 會是較佳的演算法
  - 改善方向
    - 用第N筆取代頭N筆 (代表性不足)

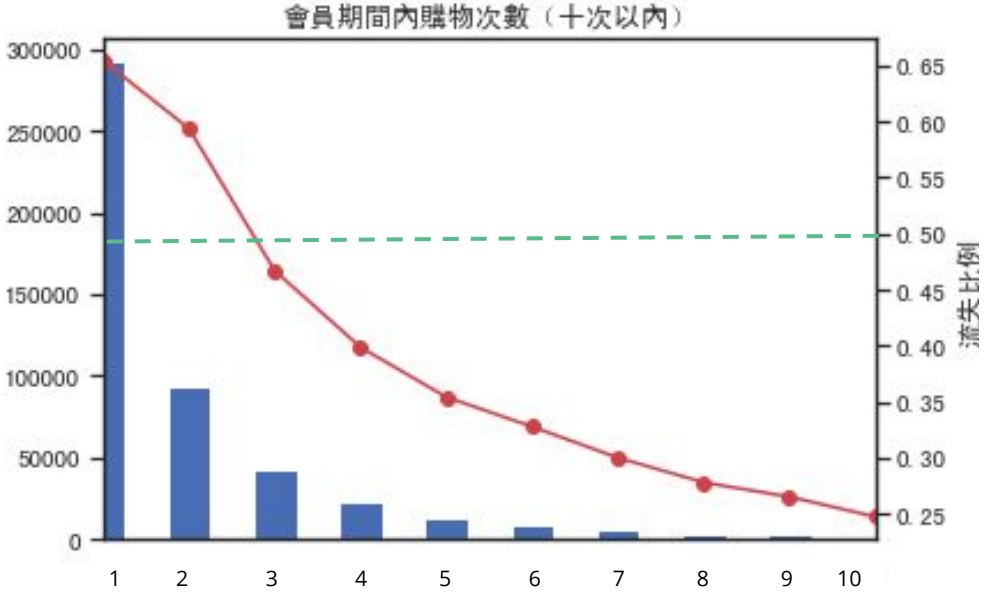
- How to implement in real-life situation?
  - 當顧客進行第二次購買時，可以透過訂單 內容得知其購買媒介、繳費方式、有無使用優惠碼等
  - 同一時間，顧客所購買的商品已經抵定，經過資料清理後即可丟入算法中做預測
  - 若我們精準預測出此顧客等級，則可以針對其顧客等級進行行銷策略投放

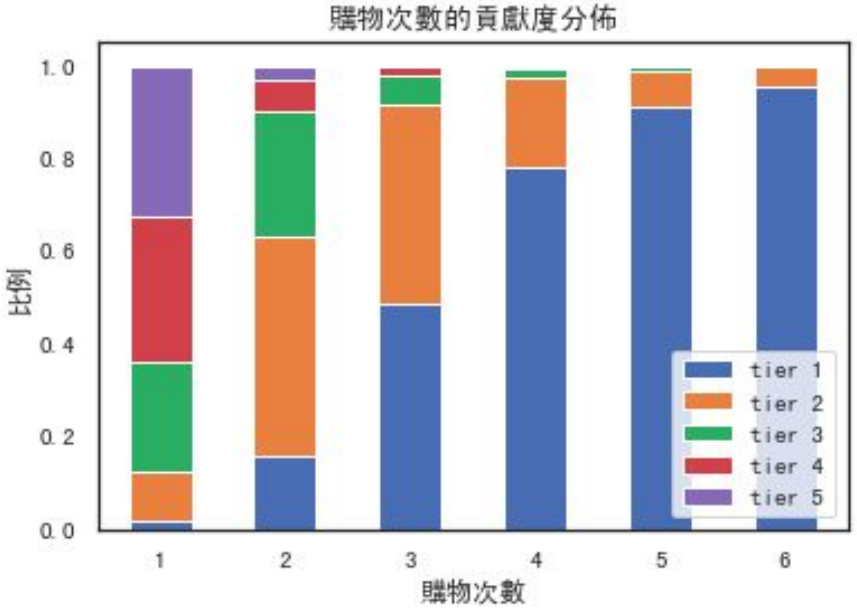
Part III

透過顧客交易紀錄，預測其留存率

會員留存預測

- 將近60%的會員僅消費一次、前三筆消費的流失比例超過一半
- 從貢獻度(以總消費金額衡量)來看，消費三筆以上就屬高貢獻族群
- 目標:我們從頭幾筆交易，判斷該顧客是否為高貢獻族群

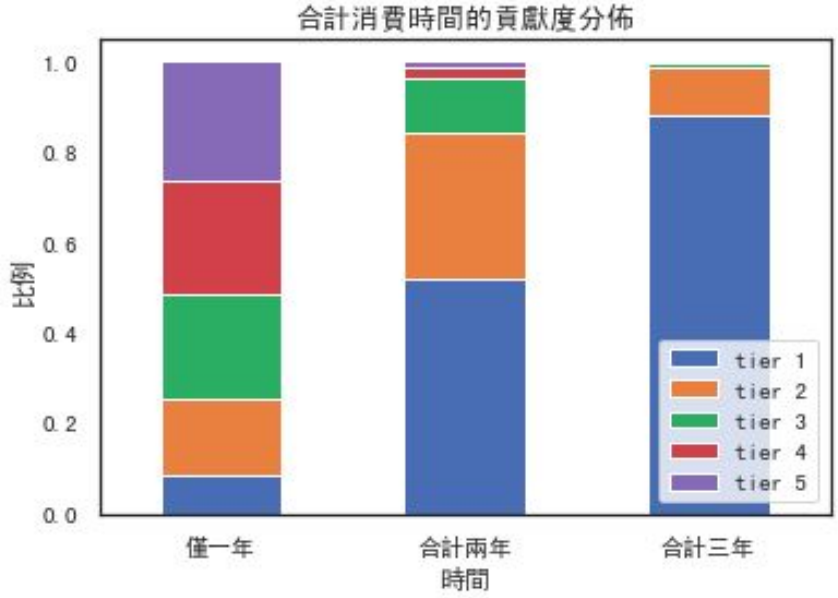




會員留存預測

- 可能問題:沒有納入時間的因素,「會再購 != 近期會再購」
- 比起集中在某一時間區段消費，能夠穩定帶來收入且會在近期消費的會員更重要
- 重新思考:從「前一段時間」的表現，預測「未來一段時間」是否留存

MemberId	year1	year2	year3	ctrbn
3270	tier 4	tier 3	NaN	tier 2
3326	tier 4	tier 2	NaN	tier 1
3672	tier 4	tier 5	NaN	tier 3
1474	NaN	tier 5	NaN	tier 5
1490	tier 1	tier 4	NaN	tier 1
1498	tier 2	NaN	NaN	tier 3
1514	NaN	tier 4	NaN	tier 4
1618	tier 2	NaN	NaN	tier 3
4344	tier 1	tier 2	NaN	tier 1
1384	tier 4	tier 5	tier 4	tier 2



會員留存預測

- 模型選擇：Logistic Regression, Fandom Forest, XGBoosting, Gradient Boosting, Naive Bayes

- 特徵解釋：
  - mean_frequency: 前N筆平均購買週期
  - get_OpenCardPresent: 是否有開卡禮
  - get_LastBirthdayPresent: 是否有生日禮
  - 時間相關的dummies(小時,假日,季節)
  - 除了有無開卡禮/生日禮，全部dummies加總（意義轉換為使用次數）

- 作法：針對前一年有買N筆以上的會員，預測未來一年是否留存 (N=1,2,3,4,5)

會員留存預測

結果:

	LR_accy	RF_accy	XGB_accy	GB_accy	NB_accy
purchase>1	0.755371	0.725954	0.762856	0.760985	0.729405
purchase>2	0.654779	0.622290	0.659891	0.656659	0.622290
purchase>3	0.649666	0.595516	0.656615	0.659630	0.573096
purchase>4	0.668349	0.639576	0.679707	0.691065	0.612569
purchase>5	0.729969	0.688358	0.728198	0.711819	0.676848

	LR_ap	RF_ap	XGB_ap	GB_ap	NB_ap
purchase>1	0.555020	0.438191	0.563912	0.560632	0.461356
purchase>2	0.658409	0.573845	0.666134	0.668292	0.609106
purchase>3	0.734708	0.654606	0.754820	0.750104	0.684371
purchase>4	0.768568	0.742804	0.792965	0.810456	0.756260
purchase>5	0.819297	0.798137	0.837267	0.832980	0.809756

```
top 10 feature of GB model:

season_spring                  1.670847
fre_mean                       0.798535
SalesOrderSlaveTotalPayment    0.731351
season_winter                  0.377458
get_OpenCardPresent            0.365416
get_LastBirthdayPresentYear    0.348498
mean_payment                   0.148715
TrackSourceTypeDef_Web         0.073189
TrackDeviceTypeDef_Mobile      0.070905
season_summer                  0.053060
dtype: float64

top 10 feature of XGB model:

season_spring                  1.537210
season_winter                  0.762382
get_LastBirthdayPresentYear    0.498456
fre_mean                       0.295182
SalesOrderSlaveTotalPayment    0.291555
get_OpenCardPresent            0.241042
season_summer                  0.152745
TrackDeviceTypeDef_Mobile      0.132805
TrackSourceTypeDef_Web         0.099542
ECouponDiscount                0.087226
dtype: float64
```

會員留存預測

- 小結:
  - 模型結合frequency與recency，以rolling-based的方式去預測會員在未來特定時間內的留存率
  - 隨著會員在前段時間的購物次數增加(達到三筆以上)，預測留存度的準度可達75%以上
  - 不同模型中，平均購物週期和是否有開卡禮/生日禮的特徵重要度都在前十名

- 應用:
  - 預測每個會員在未來特定時間內的「預期獲利」
    - 預期獲利 = 留存率 * 預期收益
  - 進而能夠鎖定「預期獲利」最高的族群，結合「顧客屬性預測模型」，針對不同的顧客屬性投放最佳的行銷策略

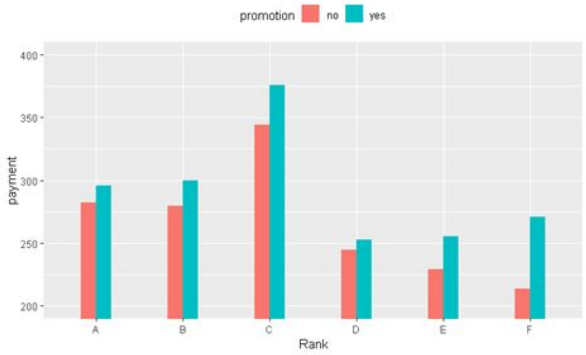
MemberId	prob	pred value	expected value
3385867	0.88	31013.53	27291.91
3287557	0.95	24677.51	23443.63
2841266	0.95	21618.37	20537.45
620298	0.94	21767.10	20461.07
595264	0.96	21226.48	20377.42
3091561	0.94	17713.47	16650.66
1697765	0.96	17300.48	16608.46
1749824	0.93	16244.76	15107.62
2988895	0.95	15592.57	14812.94
2932837	0.92	15997.57	14717.76

Part IV

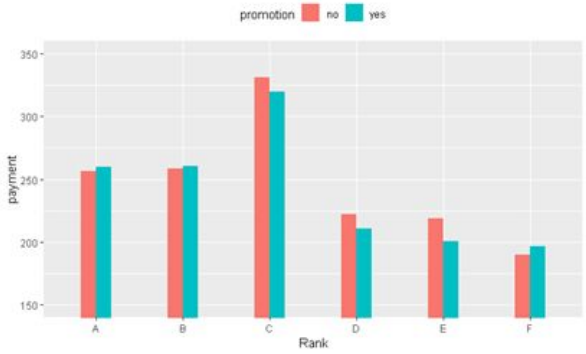
分析不同顧客類別與使用優惠之關聯

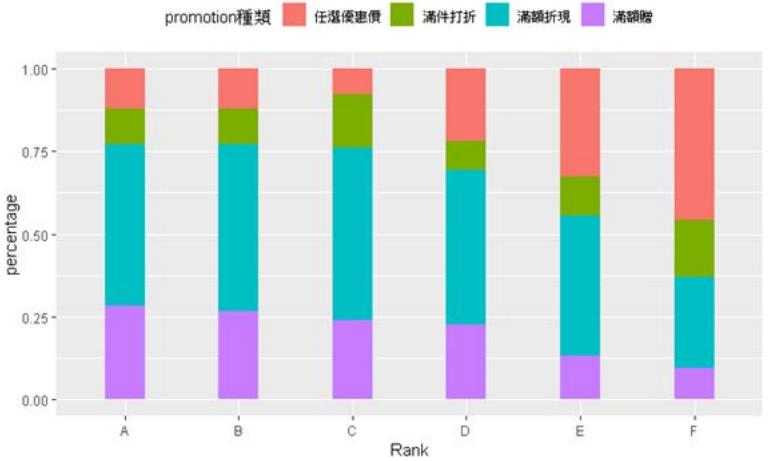
分析不同顧客類別與使用優惠(promotion)之關聯

- 所有組別有使用promotion的購買單價較高



- B組沒有顯著差異
- A F組的折價後平均消費金額依然大於無折價組

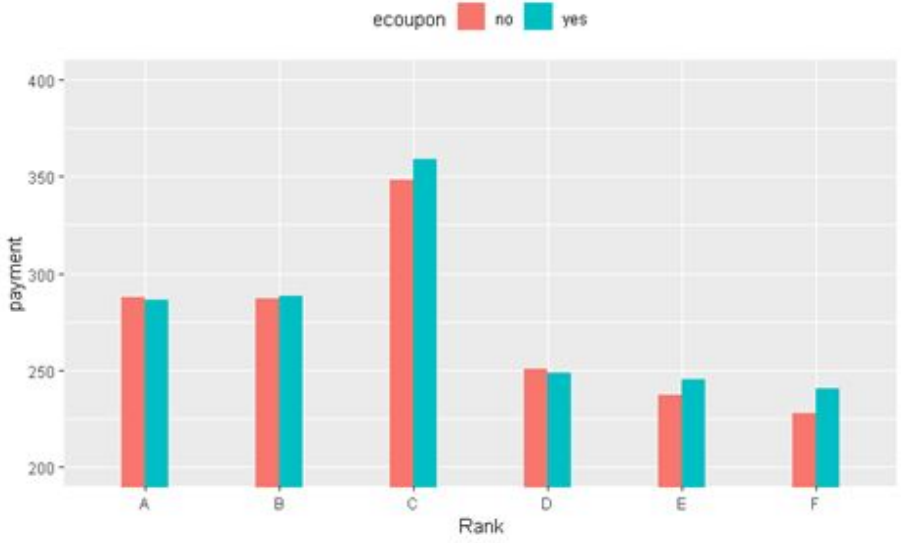


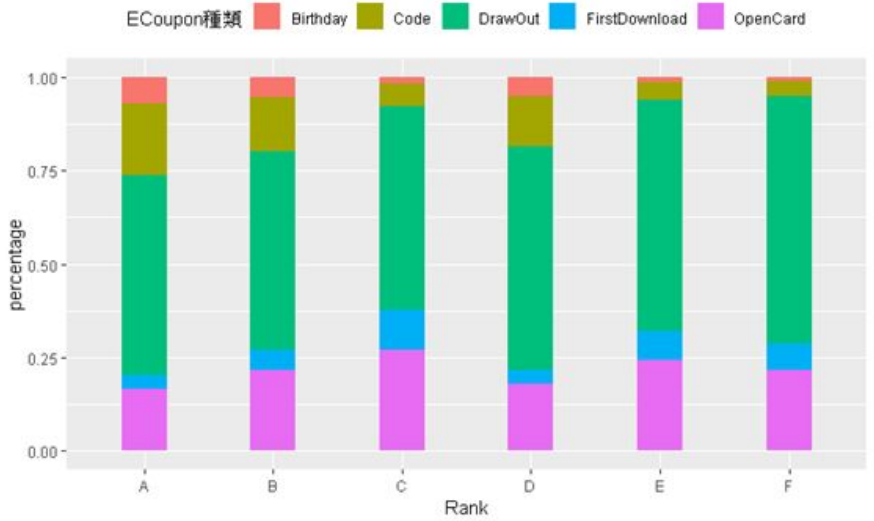


不同組有不同的promotion種類分布
→ 針對不同族群用不同種類的 promotion

分析不同顧客類別與使用優惠(ECoupon)之關聯

- 在A B兩組，有無使用ecoupon的購買金額並無差異
- C E F 組中使用ecoupon的購買金額較大

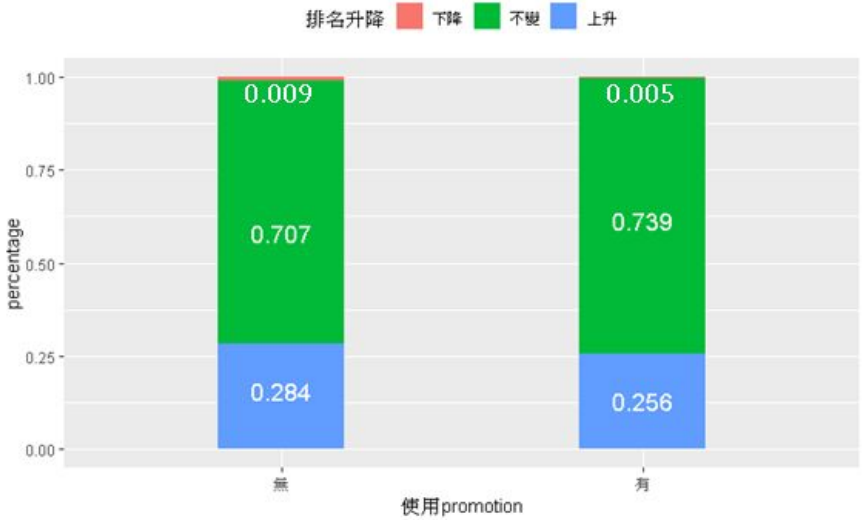


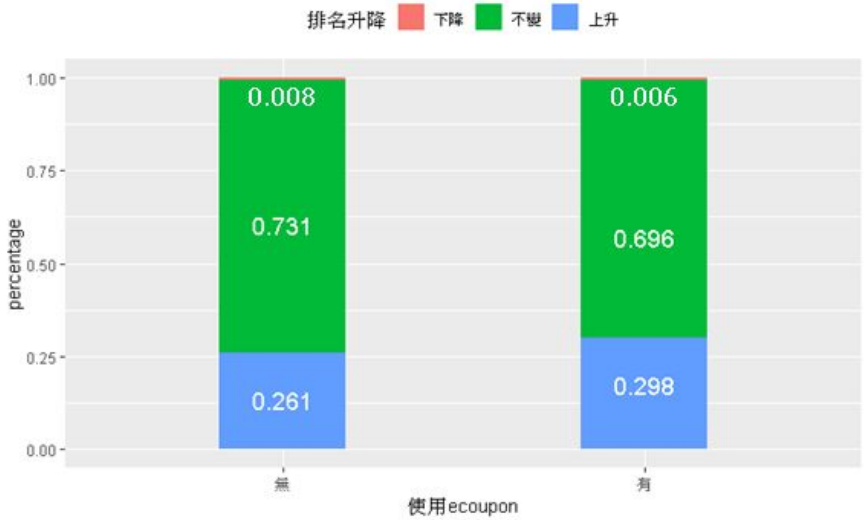


分析顧客首購使用優惠與分群之間的關係

首次消費時

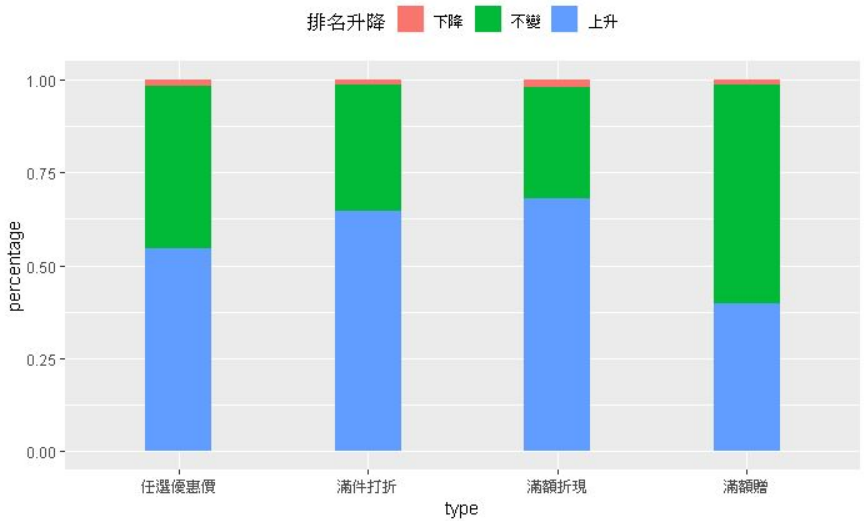
1. 使用ecoupon的人，在真實中排名 上升的比例較大，下降的比例也較小
2. 使用promotion的人，在真實中排名 上升的比例較小，下降的比例則較小





分析顧客首購使用優惠與分群之間的關係

1. 使用滿額折現與滿件打折的顧客，在真實中排名上升的比例較大



未來延伸方向

- 購物籃分析：若得到詳細的商品資訊，分析購物商品的關聯性將帶來更多價值 (Apriori Algorithm)