第一節、資料前處理

一、連結資料表

經過kaggle Olst資料集的JOIN有關訂單的資料，獲得欄位共65欄，資料共99442筆。

二、無效資料處理

我們經過對資料集欄位做敘述統計分析，初步窺探資料分布情形並將極端值刪除後，與小組成員將所有變項逐一理解與討論，將資料分析較無關的欄位刪除，刪除後的剩餘的欄位數量共23個欄位。

三、運用舊有欄位產生新欄位

1.[seller_level]將賣家產品銷售量以Q3、Q1為分界點，將Q3以上的賣家重新編碼為高銷售量賣家[1]；Q1以下的賣家重新編碼為低銷售量賣家[0]，並將其餘Q1-Q3之間的資料刪除，剩餘資料48040筆。

2.將olist平台產品利用單價、交易量進行潛在類別分析(LCA)，參照分析結果將相近的產品進行歸類，產品類型分類由71類別從新分為六大類，分別為時尚配件[1]、3C用品及小型家電[2]、藝文書籍音樂[3]、居家生活及辦公[4]、嬰幼童用品與休閒保健及食品[5]、五金工具及其他[6]。

3.利用RFM分析，R(最近的一次消費)，F(消費頻率)，M(消費金額)來評量客戶潛在價值:

(1) 最近消費日較近，消費頻率高、總消費金額皆高: 4級客戶

(2) 最近消費日較遠，消費頻率低，總消費金額高: 3級客戶

(3) 最近消費日較近，消費頻率高，總消費金額低: 2級客戶

(4) 最近消費日較遠，消費頻率低，總消費金額低: 1級客戶

4. 利用交易量高低、產品單價高低生成出四大商品型態：

(1) 交易量大、單價高：獲利商品:4級商品

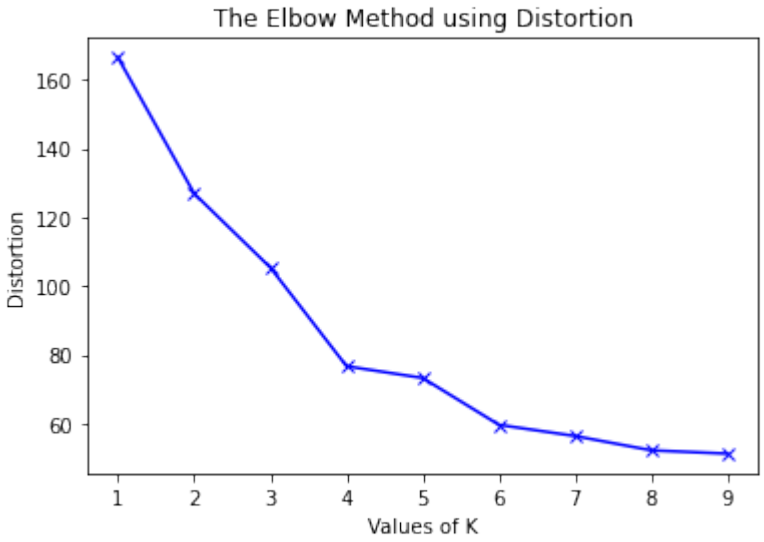
(2) 交易量大、單價低：薄利多銷商品:3級商品

(3) 交易量小、單價高：奢侈商品: : 2級商品

(4) 交易量小、單價低：淘汰商品:1級商品

5. 將[total_item_price]、[total_payment_value]、[total_freight_value] 進行集群處理成新變項[Kmean_value]：

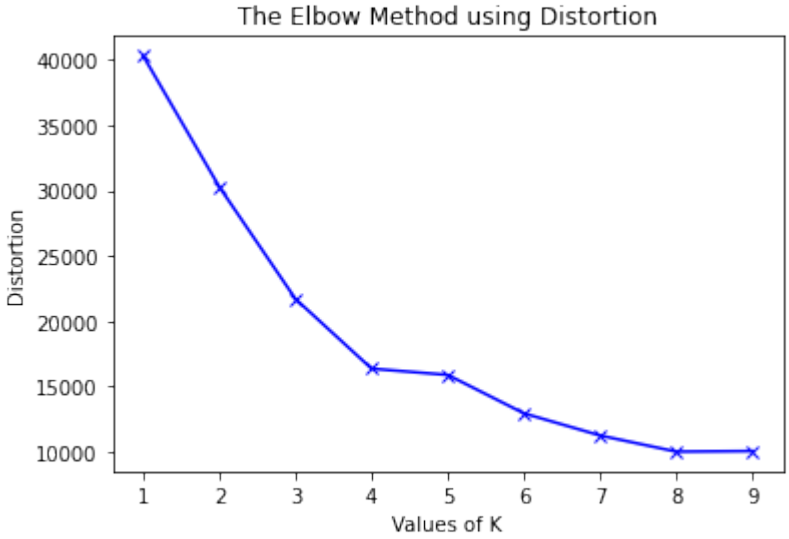
(1)透過Elbow Method 匯出如下圖，發現其值於第4類後的差異趨緩，故選擇分四群。



(2) 透過Kmean集群分析進行非監督式學習獲得新變項[Kmean_value]。

6. 將[total_freight_value]、[total_package_volume]、[total_package_weight_g]、[geo_distance] 進行集群處理成新變項[Kmean_package]：

(1)透過Elbow Method 匯出如下圖，發現其值於第4類後的差異趨緩，故選擇分四群。



(2) 透過Kmean集群分析進行非監督式學習獲得新變項[Kmean_ package]。

四、空值資料處理

1.確認資料是否有 NA 值

```
X = pd.read_csv('order_v17_XY.csv',header=0)
print(np.isnan(X).any())
```

```
item_count                    False
total_item_price              False
total_payment_value           False
total_freight_value           False
review_score                   True
review_answer_waiting_hours    True
```

2.刪除空值

```
X.dropna(inplace=True)
```

```
item_count                    False
total_item_price              False
total_payment_value           False
total_freight_value           False
review_score                  False
review_answer_waiting_hours   False
```

五、資料變數說明

依照欄位內容將資料分為類別型及數值型資料，其變項名稱如下表所示：

類別型資料	數值型資料
item_count	total_item_price
order_purchase_time_day	total_payment_value
is_shipping_delayed	total_freight_value
is_delivered_delayed	review_score
customer_state_region_type	order_purchase_year
seller_state_region_type	order_purchase_month
review_type	order_purchase_day
RFM_type	order_purchase_dayofweek
seller_level	order_purchase_hour
product_type4	until_shipped_waiting_hours
product_category6	until_delivered_waiting_hours
Kmean_price	total_package_volume
Kmean_package	total_package_weight_g
	delivery_efficiency
	total_delivered_waiting_day
	geo_distance
	total_logistics_using_hours
	estimated_logistics_using_hours
	logistics_delay_hours

第二節、機器學習BOOSTING

一、原始模型 ：

一開始將所有的26個欄位轉成類別變項放入模型，得出結果餘下表，模型於XGBOOST 與 CatBOOST 準確率皆有不錯表現，分別為0.799、0.726。但由於變項、資料多導致跑模型測試及訓練需要多一點時間，因此，我們進行特徵工程已減少欄位。

機器學習方法	C4.5 - CART	ExtraTrees	XGBOOST	CatBOOST
測試準確率	0.688	0.702	0.799	0.726

二、特徵工程：

我們藉由「相關係數」將與「賣家評比欄位」[seller_level] 高度相關（>0.7）與完全不相關（<0.1）的欄位剔除，剔除後的欄位從原本的26個欄位縮減成10個欄位，詳細的欄位如下表所示。

▾經特徵工程後剩餘的10個欄位名稱

total_item_price	total_payment_value	total_freight_value	until_shipped_waiting_hours	total_package_weight_g
total_delivered_waiting_day	seller_state_region_type	RFM_type	product_type4	product_category6

在經特徵工程挑選後，我們將結果出現的10個欄位訓練，最後進行機器學習BOOSTING方法發現準確率略微降低。

機器學習方法	C4.5 - CART	ExtraTrees	XGBOOST	CatBOOST
測試準確率	0.689	0.699	0.774	0.72

三、集群分析：

我們透過Elbow Method建議分群、並利用Kmean集群分析產生兩個新欄位，分別為[Kmean_price]、[Kmean_package]，並與特徵工程後剩餘的10個欄位一同進行機器

學習，其結果如下表。

機器學習方法	C4.5 - CART	ExtraTrees	XGBOOST	CatBOOST
測試準確率	0.682	0.709	0.792	0.728

四、比較：

我們透過將原始模型與調整欄位後發現，欄位縮減後，不但可以提升機器學習的分析效率，且準確率也會有上升。

機器學習 方法 模型	C4.5 - CART	ExtraTrees	XGBOOST	CatBOOST
原始模型 準確率	0.688	0.702	0.799	0.726
調整欄位後 模型準確率	0.682	0.709	0.792	0.728