

【資料清洗與前處理】

一、連結資料表

先把 Kaggle Olist 原始資料集中與評價相關的欄位 JOIN 成一張資料表，總共 17 個欄位，資料總數為 79852 筆。

二、欄位新增與刪減

(1) 用舊有欄位產生新欄位

- [total_shipping_hrs] (貨品運送總時長):
[order_delivered_customer_date] - [order_approved_at] (貨品成功送達的日期 - 訂單被同意的日期)
- [approved_waiting_hrs] (買家等待訂單被批准的時長):
[order_approved_at] - [order_purchase_timestamp] (賣家批准訂單的日期 - 買家下訂單的日期)
- [seller_to_logistics_hrs] (賣家出貨給物流的時長):
[order_delivered_carrier_date] - [order_approved_at] (賣家出貨給物流的日期 - 賣家批准訂單的日期)
- [geo_distance] (買家與賣家的距離):
透過 [seller_geolocation_lat]、[customer_geolocation_lat]、[seller_geolocation_lng] - [customer_geolocation_lng] 運算
- [product_volume] (商品體積):
[product_length_cm] x [product_width_cm] x [product_height_cm]

(2) 刪除 [order_delivered_customer_date]、[order_approved_at] [order_purchase_timestamp]、[order_delivered_carrier_date]、 [seller_geolocation_lat]、[customer_geolocation_lat]、[seller_geolocation_lng] - [customer_geolocation_lng]

三、原欄位重新分類

將名目型態欄位 [seller_state]、[product_category_name_english] 重新歸納縮減為 4 大賣家區域 [seller_state_region_type] 與 6 大商品種類 [self_defined_product_category]

四、資料變數說明

重新整理欄位，將新增欄位擷取放入資料集，並剔除無效的原生欄位，整理變項與名稱如下表所示：

Y: self_defined_review_score (評價分數)

| | 擷取新生欄位 (共12) | 中文名稱 | 調教模型最終欄位 (共5) |
|----|-------------------------------|-------------------|----------------------------|
| 1 | seller_state_region_type | 賣家所在區域 | seller_state_region_type |
| 2 | ln_geo_distance | 買家與賣家的距離 | ln_geo_distance |
| 3 | ln_seller_to_logistics_hrs | 賣家出貨給物流的時長 (小時) | ln_seller_to_logistics_hrs |
| 4 | ln_total_shipping_hrs | 貨品運送總時長 (小時) | ln_total_shipping_hrs |
| 5 | | 速度集群 | y2 |
| 6 | self_defined_product_category | 商品種類-歸納為6大類 | |
| 7 | ln_approved_waiting_hrs | 買家等待訂單被批准的時長 (小時) | |
| 8 | ln_item_price | 商品訂價/售價 | |
| 9 | ln_product_length_cm | 商品長度 | |
| 10 | ln_product_height_cm | 商品高度 | |
| 11 | ln_product_width_cm | 商品寬度 | |
| 12 | ln_product_volume | 商品重量 | |
| 13 | ln_product_weight | 商品體積 | |

五、資料平衡化 Normalization、標準化 Standardization 處理

(1) 刪除空值

空值比數如下：

| | |
|-------------------------------|------|
| total_shipping_hrs | 1736 |
| self_defined_product_category | 231 |
| geo_distance | 395 |

空值刪減完筆數：

77503

(2) 刪除不合理負數

[item_price]、[seller_to_logistics_hrs]、[total_shipping_hrs]、
 [approved_waiting_hrs]、[product_length_cm]、[product_height_cm]、
 [product_width_cm]、[geo_distance]、[product_volume]

(3) 對數轉換

- 因所有數值欄位偏度與峰度趨於極端，故取 LN，將數據轉為類常態分佈
- 先將欄位為 0 的數值取代為 0.001

原欄位敘述統計

| | item_price | product_length_cm | product_height_cm | product_width_cm | product_volume | approved_waiting_hours | seller_to_logistics_hours | total_shipping_hours | geo_distance |
|----|------------|-------------------|-------------------|------------------|----------------|------------------------|---------------------------|----------------------|--------------|
| 偏度 | 7.43 | 1.78 | 2.26 | 1.71 | 5.19 | 3.53 | 5.02 | 3.78 | 2.03 |
| 峰度 | 101.85 | 3.81 | 7.52 | 4.55 | 0.39 | 28.21 | 54.00 | 37.53 | 4.87 |

(4) 離群值處理

- 刪除與平均值相差 2.5 個標準差的數值

```
for i in cols:
    maxoutlier=data[i] > data[i].mean() + 2.5* data[i].std()
    minoutlier=data[i] < data[i].mean() - 2.5* data[i].std()
    data1=data[(maxoutlier | minoutlier)]
    data2=data.drop(data1.index)
print(data2)
```

- 經前述處理後，資料總數共有 75,053 筆，共12欄位

(5) 區間值編碼

- 為使所有欄位的權重一致，故把數值欄位的所有值轉為 0.2 ~ 0.8 之間

```
cols_to_norm=['ln_item_price', 'ln_product_length_cm',
'ln_product_height_cm','ln_product_width_cm','ln_product_weight_g','ln_product_volume','ln_approved_waiting_hours','ln_seller_to_logistics_hours','ln_total_shipping_hours','ln_geo_distance']

data[cols_to_norm] = data[cols_to_norm].apply(lambda x: (x - x.min()) / (x.max() - x.min())*0.6+0.2)
```

【機器學習與模型調整】

試跑數值型態機器學習 models (ex.Regression-Logic, SVN, NN, Bayesian, etc.) 與類別型態的機器學習 models (CatBoost, XGBoost, RandomForest, etc.) 做比較，最終選定類別型態的模型做調教優化。

一、為讓模型讀取資料，針對類別型態的欄位做 Label Coding

- 透過 Label Coding 將欄位型態為 'object' → 轉為 'integer' 形式
- 將 seller_state_region_type 的四大區域 CE, NE, NW, SE → 轉為 0,1,2,3 的資料型態

```
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()
data=pd.DataFrame(data)
data['seller_state_region_type'] =
labelencoder.fit_transform(data['seller_state_region_type'])
Data
```

- 將Y欄位 [self_defined_review_score] 的 'boolean' 型態 → 轉為 'integer' 形式
- ```
data['self_defined_review_score'] =
labelencoder.fit_transform(data['self_defined_review_score'])
data
```

### 二、分割訓練 Training\_Set 與測試 Testing\_Set 集資料

1. 原始資料筆數: 75053
2. 分割 Testing\_Set: Traing\_Set = 0.33:0.67 (筆數為 24768:50285)
3. 針對 Training\_Set 的 Y 評價欄位做抽樣平衡處理

### 三、Y [self\_defined\_review\_score] (評價分數) 不平衡處理

- 做完資料前處理後，測試準確率如下表 (按試跑順序排列)

|               | 好評總數  | 負評總數  | 好 / 負評比例 | 回歸準確率 |
|---------------|-------|-------|----------|-------|
| 清洗資料集         | 65798 | 9255  | 7:1      | 88%   |
| 訓練資料集 (無抽樣)   | 44010 | 6275  | 7:1      | 88%   |
| Oversampling  | 44010 | 11000 | 8:2      | 88%   |
| Bootstrapping | 12550 | 12550 | 1:1      | 77.3% |
|               | 18576 | 18576 | 1:1      | 84.8% |

|              |       |       |     |     |
|--------------|-------|-------|-----|-----|
| Oversampling | 44010 | 29340 | 3:2 | 82% |
|              | 44010 | 18861 | 7:3 | 87% |

#### 四、不同模型結果比較

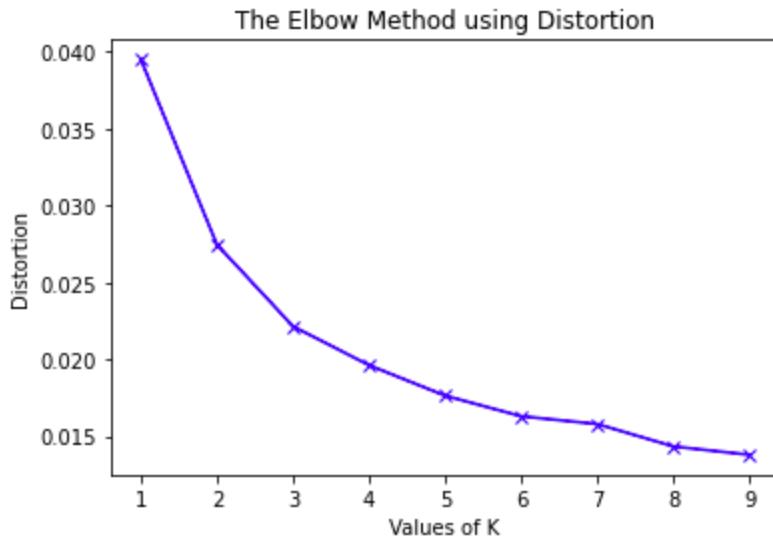
| 類別 Model       | 測試準確率 |
|----------------|-------|
| DNN            | 88.7% |
| C4.5 CART      | 88.6% |
| ExtraTrees     | 89.3% |
| Random Forests | 89.0% |
| AdaBoost       | 88.4% |
| XGBOOST        | 89.1% |
| LightGBM       | 80.9% |
| CatBOOST       | 88.3% |

| 數值 Model             | 測試準確率 |
|----------------------|-------|
| Regression-Logic     | 88.1% |
| DNN                  | 87.2% |
| KNN                  | 88.5% |
| Gaussian Naive Bayes | 79.7% |

#### 五、特徵工程

##### 1. 集群分析

- 利用 Elbow Method 做集群，依建議將 geo\_distance 與 total\_shipping\_hrs 分為 2 群，新增與「速度」相關的欄位 [y2]。



(因 2 群之後的斜率變小，代表各組之間的差異變小，故選擇分兩群)

- 與資料清洗後的12個欄位一同進行機器學習，LightGBM 預測準確度由 0.81 變成 0.89，總共提升 8%

| 類別 Model       | 測試準確率 |
|----------------|-------|
| DNN            | 88.3% |
| C4.5 CART      | 89.0% |
| ExtraTrees     | 88.4% |
| Random Forests | 89.0% |
| AdaBoost       | 88.3% |
| XGBOOST        | 88.5% |
| LightGBM       | 89.1% |
| CatBOOST       | 88.4% |

- 試降維度(與 y2 相關欄位)後，發現準確率不增反減，故保留 ln\_geo\_distance 與 ln\_total\_shipping\_hours 原欄位

|      |                 |                         |                                           |
|------|-----------------|-------------------------|-------------------------------------------|
| 刪除欄位 | ln_geo_distance | ln_total_shipping_hours | ln_geo_distance & ln_total_shipping_hours |
|------|-----------------|-------------------------|-------------------------------------------|

|     |             |             |             |
|-----|-------------|-------------|-------------|
| 準確度 | <b>0.88</b> | <b>0.87</b> | <b>0.86</b> |
|-----|-------------|-------------|-------------|

## 2. 相關係數

- 將與「評價分數」[self\_defined\_review\_score] 高度不相關(< 0.1)的欄位剔除, 但仍舊保留 ln\_geo\_distance 欄位(因準確度微幅下降)
- 從原本的 13 個欄位縮減成 5 個欄位:
  - seller\_state\_region\_type
  - ln\_seller\_to\_logistics\_hrs
  - ln\_total\_shipping\_hrs
  - ln\_geo\_distance
  - y2 (集群欄位)

## 六、最終篩選模型結果

| LightGBM        | precision     | recall | f1-score    | support |
|-----------------|---------------|--------|-------------|---------|
| <b>0:負評</b>     | 0.57 (↑ 45%)  | 0.39   | 0.46        | 2980    |
| <b>1:好評</b>     | 0.92 (↑ 4.3%) | 0.96   | 0.94        | 21788   |
| <b>accuracy</b> |               |        | <b>0.89</b> | 24768   |

原始資料集的好 / 負評比率 **0.877 : 0.123 (65798 筆 : 9255 筆)**

- 隨機猜中 0 (負評) 的可能性為 12%
- 隨機猜中 1 (好評) 的可能性為 88%

經 **LightGBM** 模型調教:

- 預測 0 (負評) 的準確率上升 45%
- 預測 1 (好評) 的準確率上升 4.3%