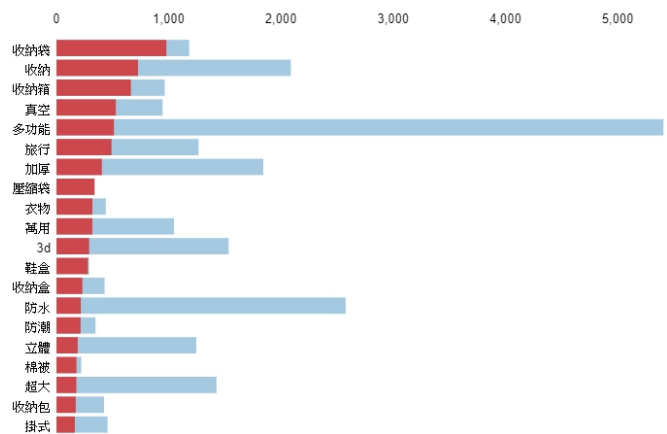
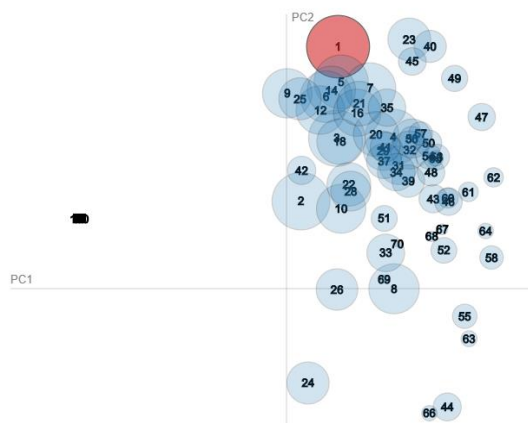
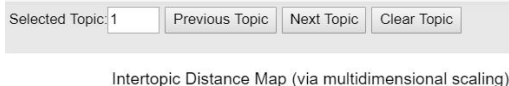
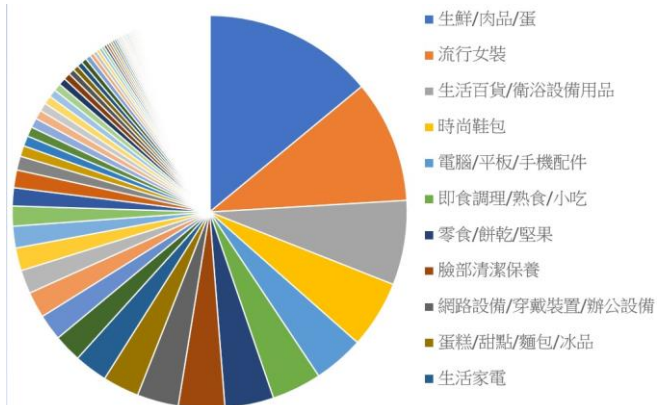
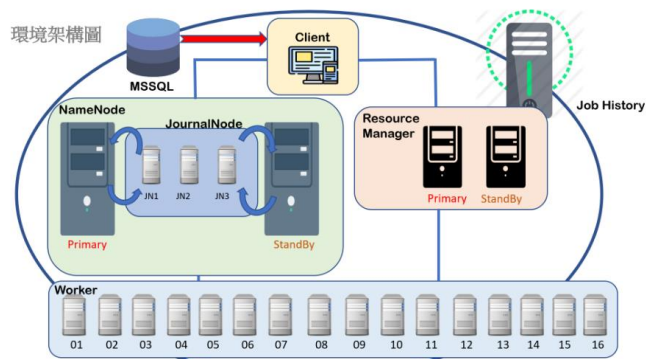


| | |
|------|---|
| 專題名稱 | 電商平台熱銷新品 AI 預測推薦引擎 |
| 小組成員 | <p>組長：陳宏嘉</p> <p>組員：蔡 ○ 周、吳 ○、陳 ○ 安、李 ○ 穎、黃 ○、葉 ○ 豪</p> |
| 主題介紹 | <p>平台電商往往要為了如何做新品選擇評估而傷透腦筋，不是太晚發現爆紅新品而喪失先機，不然就是選品多建立在人為的經驗，耗費大量人力又無法精準評估未來銷售狀況。</p> <p>另外在商品上架時，如何做好商品分類也浪費不少時間成本，且常因人為疏失導致分類錯誤，衍生許多後續行銷與分析層面的問題。</p> <p>本專題預期提供一個解決方案，同時處理新品上架自動分類，以及上架後的銷量預測，並以網路爬蟲主動找尋網路上的新商品，即時評估以掌握銷售先機。</p> |
| 實作歷程 | <p>資料蒐集：</p> <p>本組用於分析與建模的資料是由合作電商平台所提供，包含商品紀錄、行銷紀錄、瀏覽紀錄、交易紀錄等 37 億筆近 3TB 去識別化資料。另外搭配 Python 爬蟲技術，即時取得競業新品資料，從中找出具有潛力的產品。</p> <p>hadoop 環境建置：</p> <p>由於原本龐大的電商資料紀錄於 MSSQL 中，存在分析與運算效能上的問題，故於此專題建立一 hadoop 叢集架構將此巨量資料置於其中，以利使用 Spark 等更高效的方法做資料的篩選清理以及機器學習的運算工作。</p> <p>資料清理：</p> <p>利用 SQL 與 Python 語言進行資料整合、清理、空值填補等作業，並利用機器學習的方式，填補部分產品缺少細項子分類，以避免產品各分類數量不均導致分類模型訓練效果不彰的問題。</p> <p>文字探勘與分類模型建立：</p> <p>使用 Jeiba, Genism 等 NLP 文字探勘工具整理商品信息、建立詞庫並提取關鍵字；將關鍵字分群做為新特徵欄位，建立商品自動分類模型。</p> <p>銷量預測模型建立：</p> <p>利用交易紀錄與商品描述，加入分類模型所產生之精準分類，分析關鍵特徵並進行標準化等特徵工程，並使用 python 嘗試 XGBoost, Random Forest 等不同的機器學習方案建立銷量預測模型。</p> <p>資料呈現：</p> <p>利用上述分類與預測機器學習模型作為核心引擎，實作熱銷新品推薦工具，每日呈現自網路上爬取到的預測熱銷新品，提供銷量預測數據做為電商選品依歸。</p> |
| 價值分析 | <ol style="list-style-type: none"> 1. 加入更客觀的數據導向的選品銷量評估模式，精準預測熱銷商品以搶得銷售先機。 2. 找尋合適的新品與新品上架分類自動化，加速工作流程，節省寶貴的時間與人力。 3. 有利於平台電商的消費者更容易獲得合適的商品。 |

資料視覺化



```
[In [121]: itemName = df['item_name']  
symbols = r'[!@#%&'()*~.-_\\{;:\\}|[{}|~^*+<>`~\"]'  
tmp = [re.sub(symbols," ",str(title)) for title in itemName]  
symbol2 = "[ \\d]+无|[ \\d]+折(起)?|\\\\$ |\\d]+"  
itemNameC = [re.sub(symbol2," ",title) for title in tmp]
```

```
In [130]: itemNameC = itemNameC[:100]
```

```
In [239]: itemToken = [list(filter(lambda x:x.word!="",psge.lcut(i))) for i in itemNameC]  
itemNameToken
```

```
Out[239]: [[pair('桐丝纪', 'nr'),  
            pair('晃', 'n'),  
            pair('胎监素', 'n'),  
            pair('肌肉', 'n'),  
            pair('鲜活', 'v'),  
            pair('身体', 'n'),  
            pair('精華', 'a'),  
            pair('乳', 'n'),  
            pair('18', 'm'),  
            pair('入', 'v')],  
           [pair('albo', 'eng'),  
            pair('領導者', 'n'),  
            pair('立體聲', 'd'),  
            pair('智慧', 'nr'),  
            pair('藍牙', 'ns'),  
            pair('耳機', 'n'),  
            pair('麥克風', 'nr'),  
            pair('系列', 'q'),  
            pair('2', 'm'),  
            pair('入', 'v')]]
```

```
In [218]: itemTokenNl = [list(filter(lambda x:x.flag=="n",i)) for i in itemToken]  
itemNameTokenNl = [list(map(lambda x :x.word,i)) for i in itemTokenNl]
```

```
In [219]: df_codetest = list(np.int32(df.code[:100]))
```

```
In [238]: df_codetest= np.array(df_codetest).reshape(-1,1)  
itemNameTokenNl = np.array(itemNameTokenNl).reshape(-1,1)
```

```
train_x = scaler.transform(train_x)
test_x=scaler.transform(test_x)
```

```
In [28]: train_x.mean(axis=0)

Out[28]: array([-9.98657353e-17,  3.39763719e-16, -2.52827841e-16,  5.76942456e-17,
  3.39911984e-16, -4.15334451e-16,  1.99962041e-16, -3.15876794e-15,
  2.37535777e-15, -2.57062647e-15, -1.630768037e-16, -1.81309937e-16,
  1.72028961e-16,  1.01256815e-15, -9.92676370e-16, -4.37324325e-16,
  -1.71723535e-16,  3.40814243e-15,  1.51571120e-15,  1.63083227e-15,
  -7.7867285e-16,  0.00000000e+00,  5.48035464e-16,  3.91018745e-15,
  -8.52363063e-15,  4.49787404e-15, -1.20873192e-17, -2.81524343e-16,
  -4.62173066e-18,  9.06736719e-16,  4.09944058e-15, -9.98054970e-16,
  -2.22553336e-16])
```

```
In [29]: train_x.std(axis=0)

Out[29]: array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,  
               1., 1., 1., 0., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
```

選所有feature來做線性回歸 並計算 R^2

```
In [30]: #建模
regressor=LinearRegression()
regmodel=regressor.fit(train_x, train_y)
```

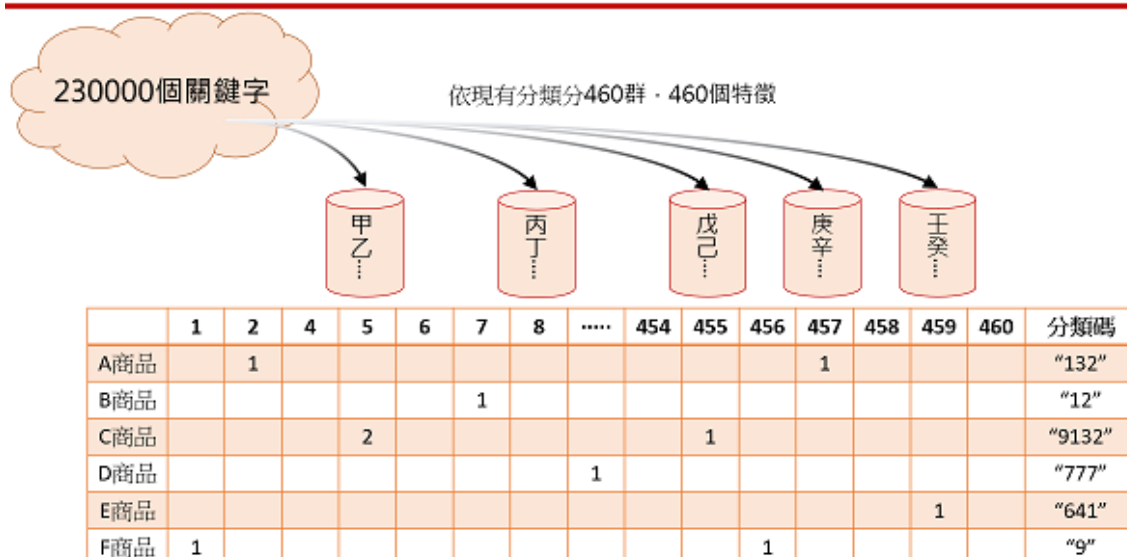
```
In [32]: from sklearn.metrics import r2_score

In [33]: #以 test data 評估模型好壞 (R^2)
pred_y = regmodel.predict(test_x)
r2_score(test_y, pred_y)
```

Out[33]: 0.2955972294117119



分類預測



銷售預測-模型效果比較

