# Term Project Final Report

## - Kaggle - Store Sales - Time Series Forecasting


組別:第 11 組

組長: 何子安 E44065020

組員: 蔡東霖 F74071166

林千祺 N46084036

吳定洋 F74076213

# 一、 任務簡介

針對教授選定的其中一個 Kaggle 資料集題目，對其預測雜貨業的銷售額為本次期末任務之最終目標。在這之中也會對各個資料集進行盡可能全面的資料分析與資料探勘，並於簡報與報告中呈現我們的發現。Kaggle 上共提供了 5 份可以作為輸入資料的的檔案，且說明 Kaggle 競賽上的分數將會以 Root Mean Squared Logarithmic Error (RMSLE)為評估標準。

# 二、 資料概述

Kaggle 上所提供的 5 份資料集分別為 train.csv、transactions.csv、stores.csv、oil.csv、holiday_events.csv。以下將會對資料集個別概述。

1. Train.csv 中含資料筆數 3000888 筆，且沒有缺失值。

| Column | Range/Unique | Description |
|---|---|---|
| id | 3000888 | **索引值** |
| date | 2013.01.01 ~ 2017.08.15 | **日期** |
| store_nbr | **54 間** | **商店編號** |
| family | **33 種商品** | **產品類別** |
| sales | 124717 | **該產品於該商店當日的總銷售額** |

| | | |
|---|---|---|
| onpromotion | 741 | **該產品於該商店當日的總銷售額** |
| **備註** | **發薪日為每月 15 日及月底; 2016.04.16 厄瓜多大地震，地震發生後數週，人們齊心協力捐贈水和其 他急需物資，極大地影響了超市的銷售。** | |

2. Transaction.csv（資料筆數 90936，沒有缺失值）

- 將 date 資料類型改為 datetime
- 將 train 資料根據日期，將各店的銷售量進行加總，產生每日各店銷售量
- 將 transaction.csv 與 train.csv 進行合併，得到新的 column
- 合併後 transaction 欄位會產生 7448 個缺失值

| Column | Range/Unique | Description |
|---|---|---|
| data | 2013.01.01~2017.08.15 | **日期** |
| store_nbr | **54 間** | **商店編號** |
| transaction | 5~8359 | **交易量** |

3. Stores.csv（資料筆數 54 筆，沒有缺失值）

| Column | Range/Unique | Description |
|---|---|---|

| | | |
|---|---|---|
| store_nbr | 54 間 | 資料中涵蓋了 54 間店的其他資料 |
| city | 22 個城市 | 這些店分別在 22 個城市內 |
| state | 16 州 | 這些店分佈在 16 個州中 |
| type | 5 類 | 共 5 種不同類型的商店 |
| cluster | 17 種 | 可分為 17 群相似的商店 |

4. holidays_events.csv(資料筆數 351 筆，沒有缺失值)

| Column | Range/Unique | Description |
|---|---|---|
| date | 2012.03.02~2017.12.26 | 日期 |
| type | 類型，有 6 種 | 分別為 Holiday、Transfer、Additional、Bridge、Work Day、Event |
| locale | 地區類型，有三種 | 分別為 Local、Regional、National |

| | | |
|---|---|---|
| local_name | locale 的詳細名稱 | 有三種可能，如果 locale 為 National，則此欄為 Ecuador；如果為 Local，則此欄為 city name；如果為 Regional，則此欄為 state name |
| description | holiday 的描述 | 為當天 holiday 或 event 的名稱 |
| transferred | True/False | 記錄當天的 holiday 是否有被移動到別天 |

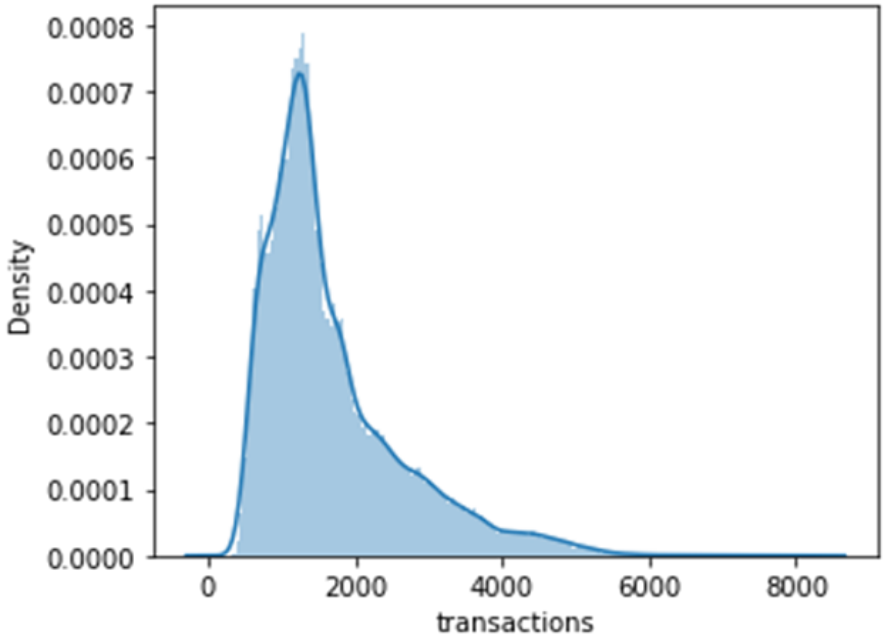
# 三、 資料分析

1. Transcation

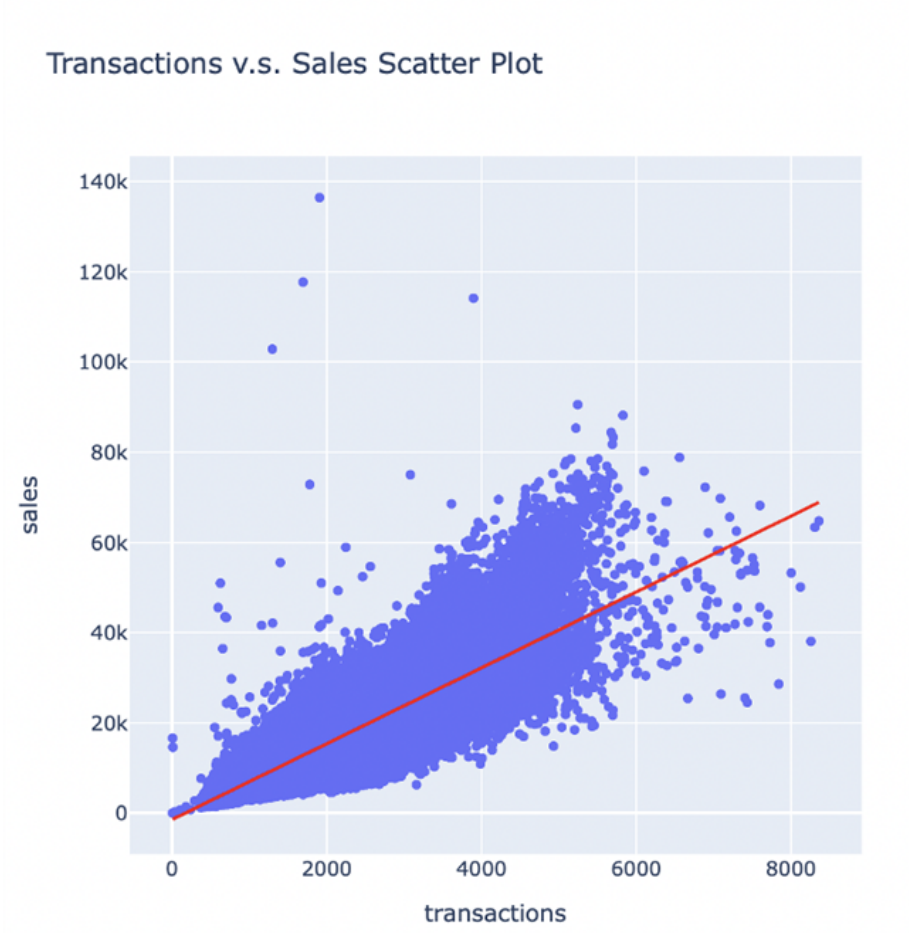
- 將 transaction 與 sales 進行相關性分析，相關係數很高，為 0.8374，資料型態為單峰、右偏，表示偏低的資料較多。而繪製散佈圖可以看到離群值大多分布在偏低的 transaction 資料中，且離群值偏高。
- 原本想將 transaction 作為模型輸入的其中一項屬性，但後來發

現，在官方給予的測試資料 (test data) 中，並沒有 transaction 這項屬性，因此無法作為模型的輸入值。

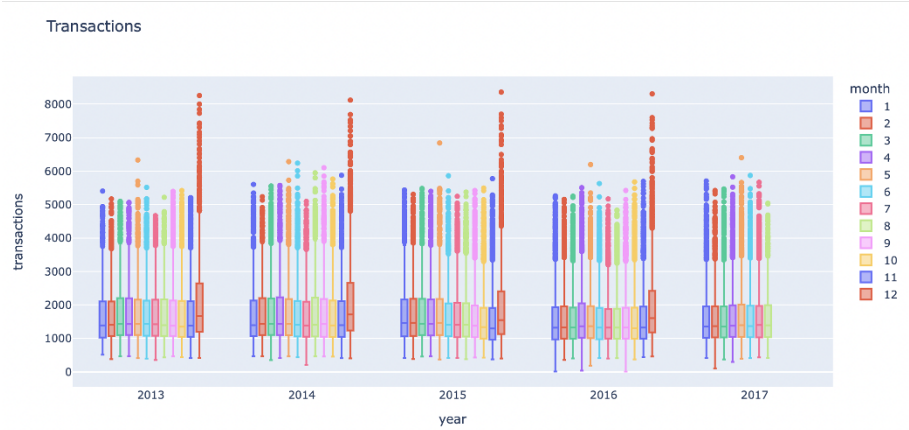
- transaction 與 sales 進行相關性分析



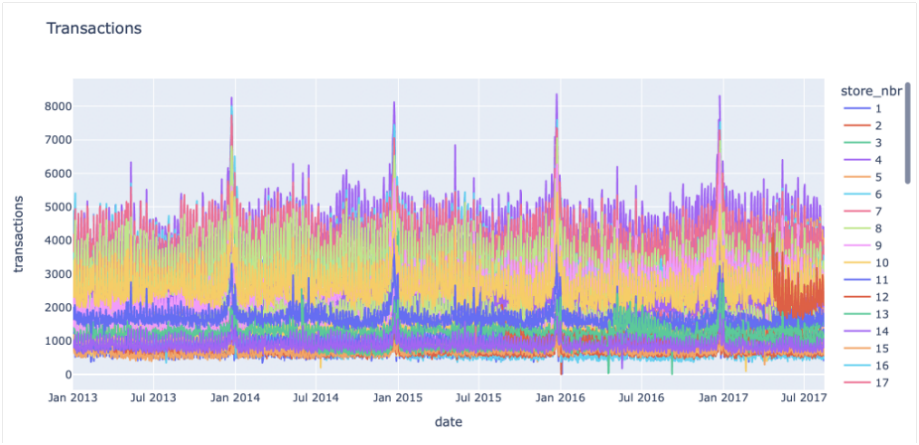

- 散佈圖



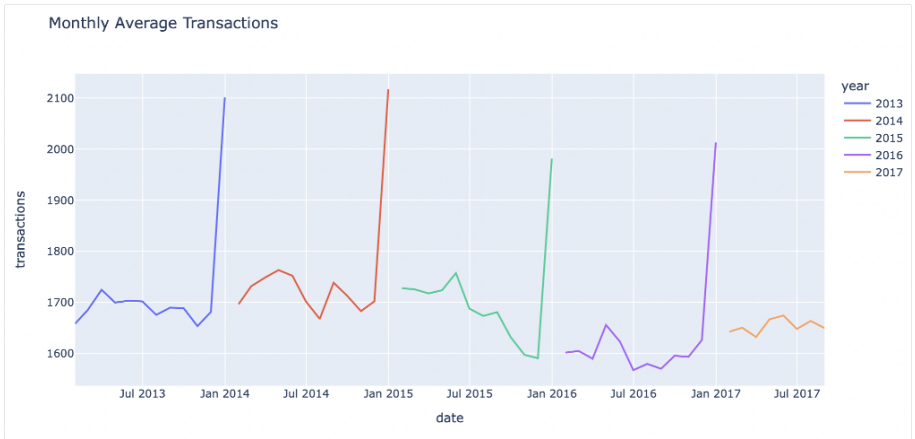

- 繪製箱型圖，可以看到 12 月有最多的離群值，資料較離散，且 12 月的交易量也最大。



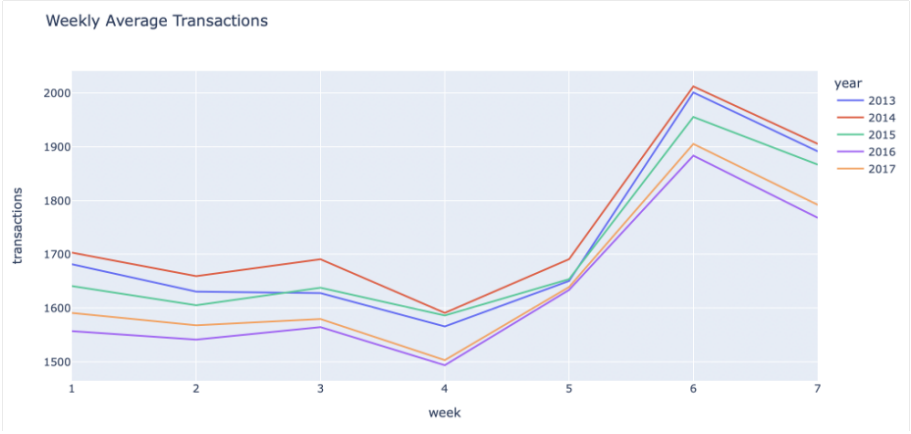
- 繪製各商店交易量隨時間變化的折線圖，可以發現每家店在 12 月時，交易量皆會顯著上升，因此這是一個全域的現象，不受地區影響。



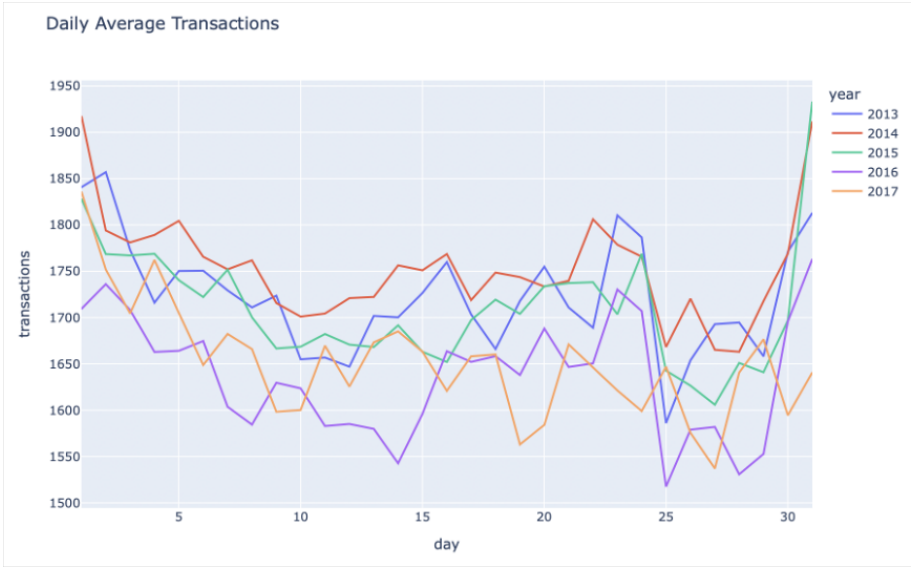

- 繪製每年的月交易量變化折線圖，可以看出每月的交易變化趨勢是很相似的，並且隨著年份緩慢遞降。2016 年的交易量最低，但在 4 月份時有一個小高峰，推測與地震有關。



- 繪製每週交易變化折線圖，可以看到在週末時，交易量上升，其中，週六最高，其次是週日，而週四交易量最低。



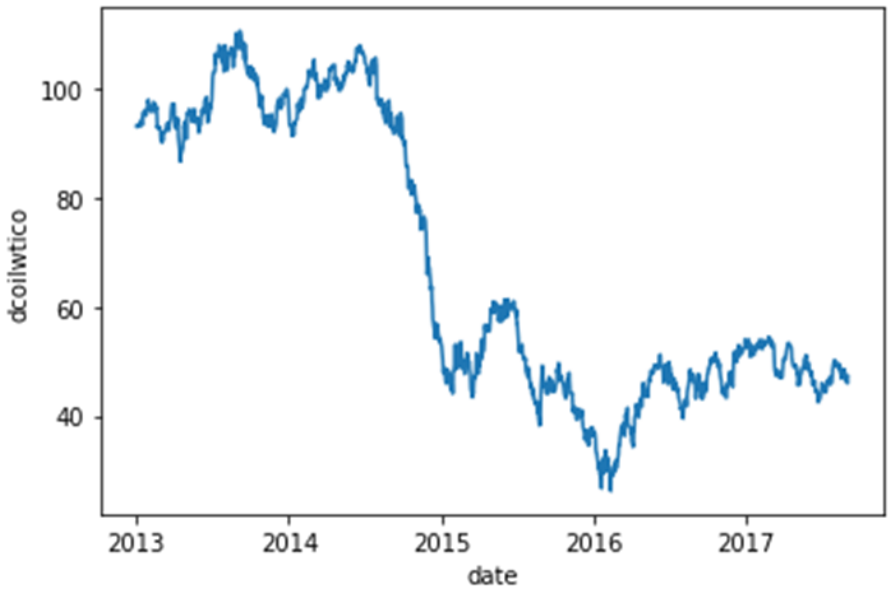

- 繪製單月中的每日平均交易量變化圖，可以看到月初與月末交易量最高，從月初緩慢遞減至 15 日左右，會再小幅度上升，每月 25 日左右是交易量最低的時，推測與發薪的日期有關。



2. Oil

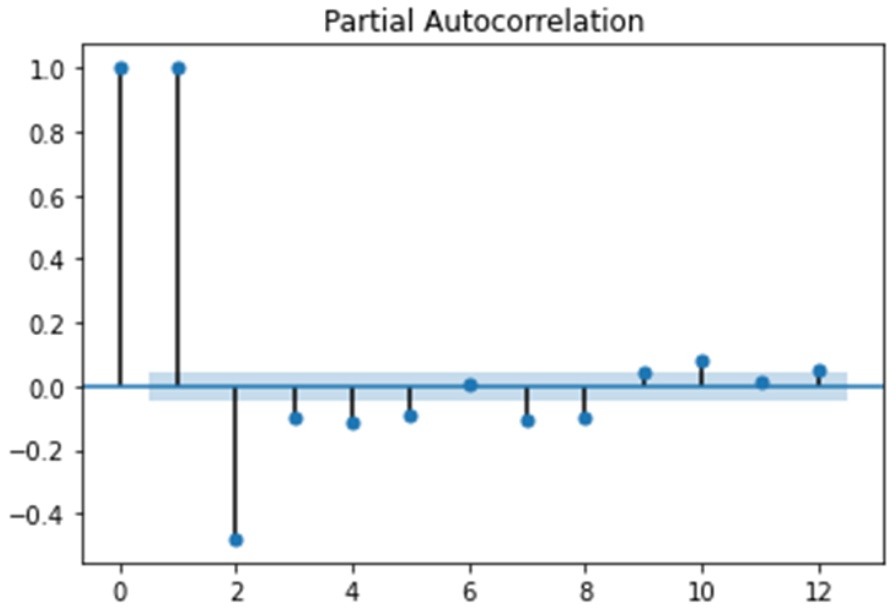
- 油價與時間的關係圖

  油價長期來看是漸低的，2016 是低谷、2013-2014 間是高峰



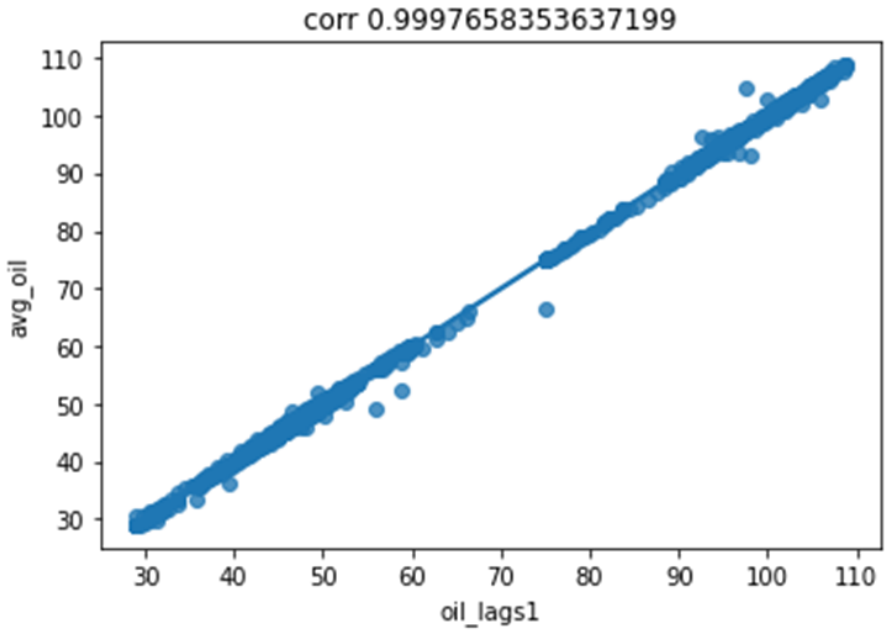

- Oil 的自相關圖-滯後圖

  產生滯後的最大值高達 5



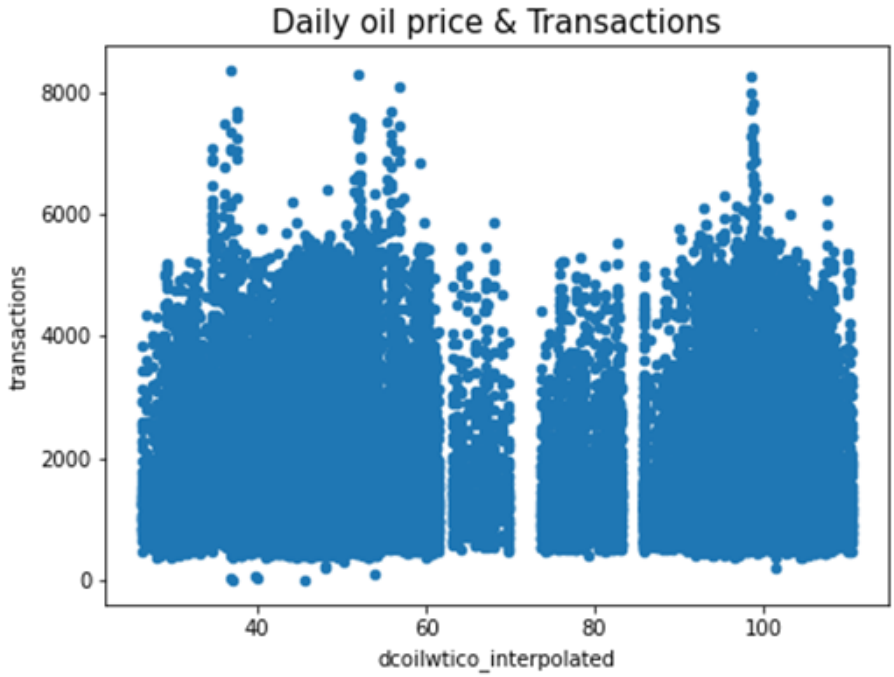
- 滯後平均油價 vs 及時油價的關係

  滯後油價可以預測未來的油價



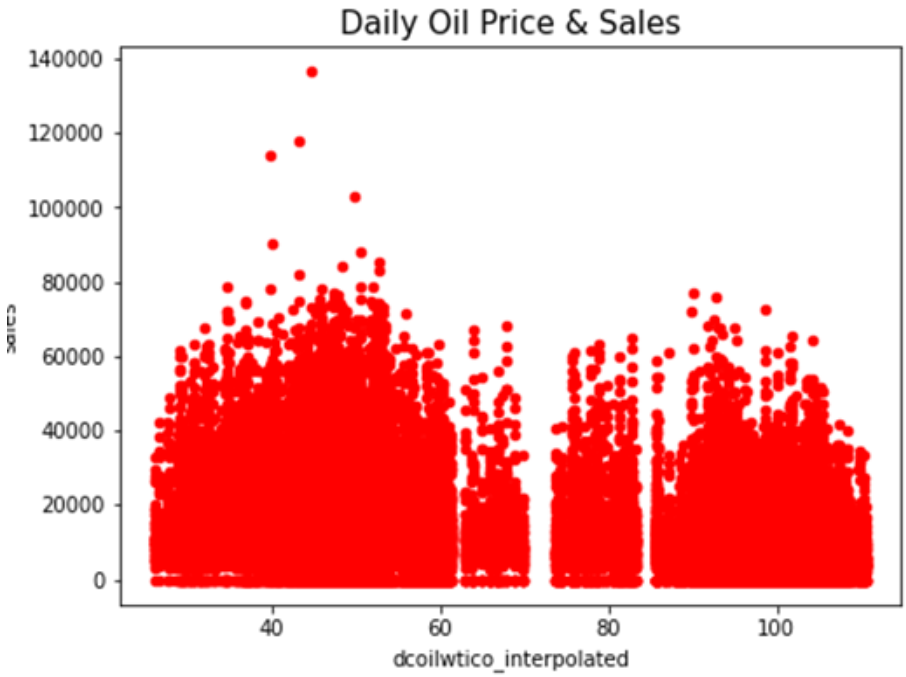

- 油價 vs 交易量

  Correlation: transactions 0.04



- 油價 vs 銷售量

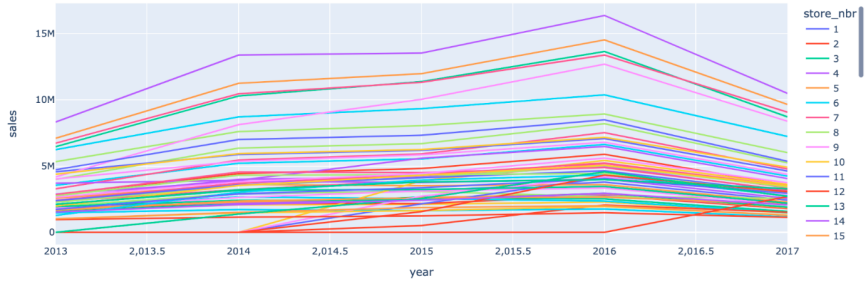
  Correlation: sales  -0.30




3. Stores

- 幾乎所有的店面在 2013 - 2015 年中 sales 都有趨增的情況，

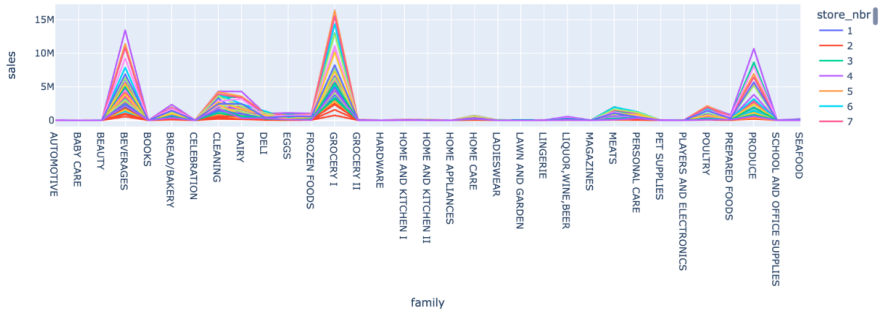
  且都在 2016 年的 sales 有下降的趨勢



- 所有店家的熱賣商品都集中在這一些品項

  GROCERY 1, BEVERAGES, PRODUCE, CLEANING,

  BREAD/BAKERY




## 四、 資料前處理

Holiday & Event

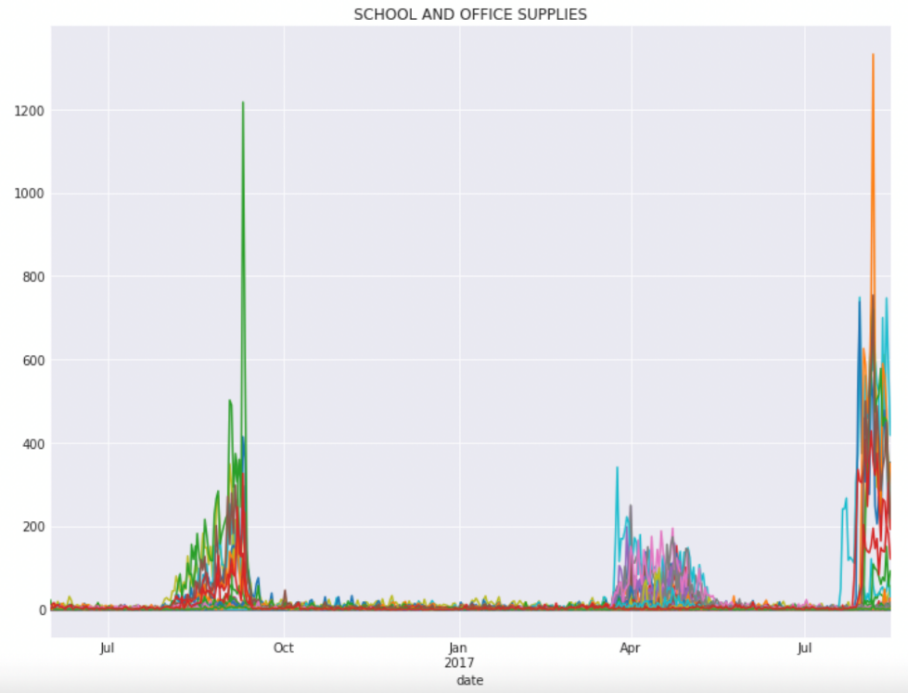
- 將 holidays_event 中所有的 holidays 和 event 用 one hot encoder 列舉出來之後，使用 A/B test 觀察各個 holidays 和 event 的發生對於 sales 變化是否有顯著的影響。將有顯著影響的 holidays 和 event 挑出以提供後續 feature 上的選擇，而沒有通過 A/B test 的 holidays 和 event 在後續要選取 feature 時就完全不用考慮，以節省訓練時間和提高訓練效率與精確度。
- 在 A/B test 中，A group 為有此 holiday，發生那天的所以商店所有商品 sales。B group 為無此 holiday 發生那天的所以商店所有商品 sales。

## 五、 資料挑選

- 挑選 [2017/4/30 - 2017/8/15] 之間的數值
- 平均油價與油價資料的 lag
- Holiday - 各個地區的假日資訊
- School session - 關於各學校的上課資訊
- Blending - 先使用 Linear Regression 對 train data 做一次預測，利用線性回歸得到初步的預測值之後，將預測值加入至原來的 train data 中，以提高模型預測的精準度。幾個 Predictive Features，然後再對這些 Predictive Features 做集合

## 六、 模型介紹

1. 本次競賽我們採用的策略是盡量使用我們熟悉且使用過的預測模型，並在此基礎上嘗試做一些改良。根據我們對於資料的觀察，'SCHOOL AND OFFICE SUPPLIES' 類別的資料分佈與其他特徵相異，如下圖所示，有很強烈的週期性，只在特定月份的銷量上升，因此會對該項目進行特殊處理




2. 當輸入項目有 'SCHOOL AND OFFICE SUPPLIES' 時，使用 ExtraTreesRegressor 及 RandomForestRegressor 等分類樹的機器學習方式作為基礎模型，並將這兩個模型加入 Bagging Regressor 設定每一個模型的訓練次數為 10 次，每一次將從訓練資料中隨機抽取作為子集合來進行預測，並取平均值作為該模型的預測值。最後使用 Voting Regressor 將兩者的預測值作為候選人，也取平均值，得到最終的預測值。

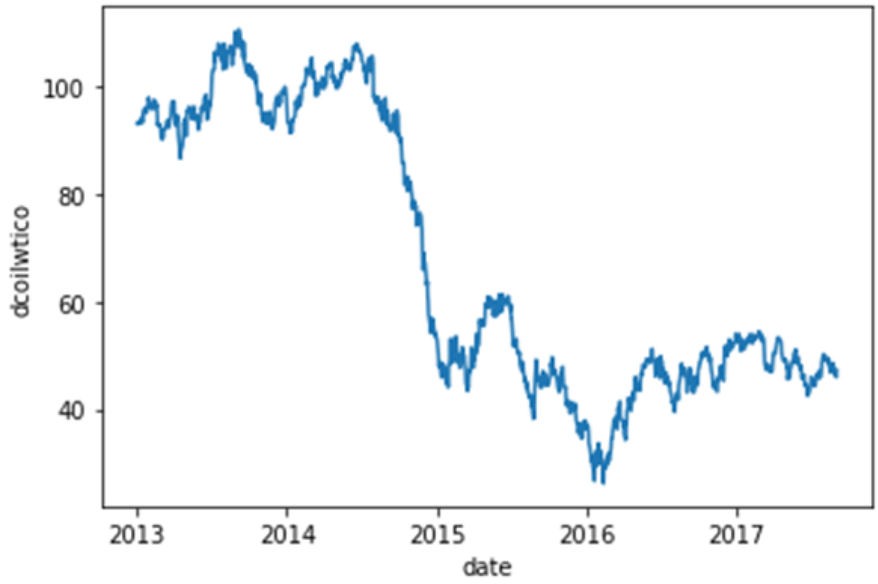
   而其他項目則使用 Ridge 和 SVR 模型來進行預測，同樣也結合 Bagging Regressor 及 Voting Regressor 的方法得到最終的預測值。

   該方式可以減少離散值對於預測所造成的誤差，大致的模型流程圖如下所示：

```
if 'SCHOOL AND OFFICE SUPPLIES' :
    r1 = ExtraTreesRegressor
    r2 = RandomForestRegressor
    b1 = BaggingRegressor(base_estimator=r1, n_estimators=10)
    b2 = BaggingRegressor(base_estimator=r2, n_estimators=10)
    model = VotingRegressor([('et', b1), ('rf', b2)])
else:
    Ridge
    SVR
    Model = VotingRegressor([('ridge', ridge), ('svr', svr)])
```

## 七、 改進方法

1. 嘗試一：在 base case 中我們使用了 2017/4/30 到 2017/8/15 的資料作為 test 資料集，我們嘗試將時間區段拉長，每次都增加一年，想看看是不是有更多的資料進行訓練會有更好的結果，而且將時間拉長到超過一年以上還可以看看是不是有 seasonal 的週期變化，結果發先成績反而越來越差。我們認為可能是我們有一個重要 feature 油價在 2014 年末到 2016 有非常劇烈的變化，導致訓練的模型會因為這段不規律的變動而變差。




2. 嘗試二：在 base case 中的 holiday_event.csv 我們只使用了 locale 為 National 的資料作為 feature 訓練，在這次嘗試中我們也加入了 regional 和 local 的 holiday 進行訓練，但結果反而更差，並沒有讓我們預測結果有所提升。
3. 嘗試三：在 base case 中我們模型中有 SCHOOL AND OFFICE SUPPLIES 這個商品 family 中模型使用了 ExtraTreesRegressor、RandomForestRegressor，並將這兩個 Regressor 使用 BaggingRegressor 行成預測，最後再使用 VotingRegressor 選出有最好的結果的組合作為我們的 model。在這次嘗試中，我們打算使用 XGBRegressor 取代 RandomForestRegressor，因為在前幾次作業中，使用的多種比較基本的 Regressor，基本上都是 XGBRegressor 可以略勝其他 Regressor 一籌，因此結合之前的經驗我們嘗試使用

XGBRegressor，結果確實讓我們的預測結果有所進步，kaggle 排名也往前排爬升了不少。

## 八、 預測結果

經過組員們不懈的努力，我們終於將排行榜上的名次追到了第 21 名，最終的最佳分數為 0.40334，相較我們最差的成績 2.03583，進步了 1.63249 分。

## 九、 結論

1. 關於分數躍進的部分，我們認為有幾件事非常值得我們一提。首先，資料的「乾淨」程度其實會對模型的預測結果好壞十分敏感，且比起模型的預測能力，資料清理或特徵工程的部分才是最能影響模型預測結果的好壞。
2. 在所有訓練資料中，Holiday 資料處理的複雜度最高，因此我們也學習到了很多 dataframe 的操作方式。雖然 Holiday 資料可以進行非常細部的特徵分析，但最後在進行預測時，會發現其實影響到全域的屬性，才是真正可以提高預測準確度的關鍵。因此最後在進行輸入參數的選取時，也只有選擇全國性的節日，區域性的節日會使準確度降低。

3. 在模型選擇上也不一定要一味的追求使用最先新，最複雜的方法，像是我們本次的模型選用的是多種比較簡單 Regressor 結合，且最後得到的成果也挺令人滿意。
4. 有關於時間序列的處理方式，由於第一次接觸，我們也較不熟悉，因此也找了很多相關的資料。其中最為重要的處理方式是利用拉普拉斯轉換，將時間序列解構成多個週期特徵，另外，也會加入很多滯後資料作為特徵。這些處理方式，其實我們在討論過程中，並不能完全理解其中的道理，因此之後可以再多熟悉一些時間序列的相關背景知識。
5. 接下來就是我們也發現在資料科學領域中，是非常依賴經驗的累積與了解任務目標的背景知識，才是資料科學競賽中任務成敗的關鍵。
6. 最後一點要強調的是，站在巨人的肩膀進行任務會使得任務事半功倍，也可以減少對錯誤方法的重蹈覆轍。

## 十、 未來工作

1. 利用 GridSearch 找出最佳的模型組合：在我們的模型中有用到許多的 Regressor，而這些的 Regressor 我們都可以使用 HW3 練習到的 GridSearch 得到更多更好的超參數，例如 n_estimators、learning_rate、max_depth 等等，未來如果要更加優化我們的模型，加入 GridSearch 應該會有不小的提升，但是在原本的 base case 沒有使

用 GridSearch 下，我們訓練一次模型、跑出預測都跑了快 1 個小時，如果加入 GridSearch，所需的時間會增加好幾倍，這也是我們還沒有嘗試使用此方法的原因。

2. 篩選出更具影響力的特徵項：在本次的模型訓練中，我們並沒有將 kaggle 所提供的所有資料都加入 feature 進行訓練，例如 Transaction.csv、store 的 onpromotion 等等，此外我們在 holiday_event 也尚未能夠將每間商店所在地區會擁有的特定節日完全加入到模型訓練中，可能在這些我們沒加入的 feature 還有更有潛力的。未來將全部的 feature 都加入訓練後，可能還可以使用 xgboost 的 plot_importance，將較為重要的 feature，挑出後再進行優化。
3. 嘗試不同 Blending 的作法：我們在本次模型建構中使用了 LinearRegression 作為我們 Blending 各家商店、各項商品形成 specific feature 的方法，我們考慮未來如果要更增進我們的預測，可以將 LinearRegression 改成不同 Regressor 作為嘗試，說不定可以做出更好更準確的 feature 以供使用。

## 十一、 參考資料


- Store Sales TS Forecasting- A Comprehensive Guide | Kaggle
- Store Sales- Time Series Forecasting- Hyperparameters | Kaggle
- https://wizardforcel.gitbooks.io/ntu-hsuantienlin-ml/content/25.html