機器學習期末報告

Predict Future Sales

資工系四年A班 學號 1061443

資工系四年A班 學號 1061416

資工系四年C班 學號 1063337

1. 組員名單與分工(請標註工作佔比)

- 請詳細列出工作項目，以及在專題的工作佔比

|  |  |  |
| --- | --- | --- |
| 學號 | 工作內容 | 專題佔比 |
| 1061443 | Survey paper、LSTM模型實作、製作報告、報告主講、demo影片 | 40% |
| 1061416 | Survey paper、XGBoost模型實作 | 30% |
| 1063337 | Survey paper、Random Forest模型實作 | 30% |

1. 開發動機與目標  
    做銷售量預測，可以對店家有正向的幫助，店家透過預測可以知道有些時段銷售量是高是低，可以根據這個預測去考量進貨的量，減少成本。消費者端也可以透過銷售量預測，或許銷售量低的時候，價格也會促銷，撿到便宜！  
    我們希望我們這個比賽可以達到比賽前15%
2. 參考文獻探討

-此題目是否有相關的paper也是這個題目

Predicting Future Sales of Retail Products using Machine Learning   
[2020-08-18]

-說明文獻上怎麼做，有什麼缺點

這篇論文也是參與相同的比賽，他們的缺點我覺得在於資料前處理還可以做得更好，shop\_id、item\_category\_id這兩個欄位他們並沒有去多做處理，我們自己就有多做這方面的處理，再來就是論文中的lag feature取1-3 months 跟 12 months，我覺得可以再取6 months，以半年來做數據統計也是有意義的。

-用一個表格說明這些文獻與你的方法的差異

|  |  |
| --- | --- |
| 論文的方法 | 我們的方法 |
| shop\_id、item\_category\_id未做處理 | 利用 shop\_id、item\_category\_id 取出 city\_code、type\_code、subtype\_code |
| Lag\_feature 取1-3 months、12 months | Lag\_feature 取1-3 months、6 months、12 months |

1. 解決方案介紹
   * 提出的方法架構

我們使用了三種方法：

1. LSTM
2. XGBoost
3. RandomForest
   * 蒐集的資料  
     使用Kaggle競賽提供的資料：  
     檔案說明：

* **sales\_train.csv** - the training set. Daily historical data from January 2013 to October 2015.
* **test.csv** - the test set. You need to forecast the sales for these shops and products for November 2015.
* **sample\_submission.csv** - a sample submission file in the correct format.
* **items.csv** - supplemental information about the items/products.
* **item\_categories.csv** - supplemental information about the items categories.
* **shops.csv** - supplemental information about the shops.

欄位說明：

* **ID** - an Id that represents a (Shop, Item) tuple within the test set
* **shop\_id** - unique identifier of a shop
* **item\_id** - unique identifier of a product
* **item\_category\_id** - unique identifier of item category
* **item\_cnt\_day** - number of products sold. You are predicting a monthly amount of this measure
* **item\_price** - current price of an item
* **date** - date in format dd/mm/yyyy
* **date\_block\_num** - a consecutive month number, used for convenience. January 2013 is 0, February 2013 is 1,..., October 2015 is 33
* **item\_name** - name of item
* **shop\_name** - name of shop
* **item\_category\_name** - name of item category
  + 資料前處理

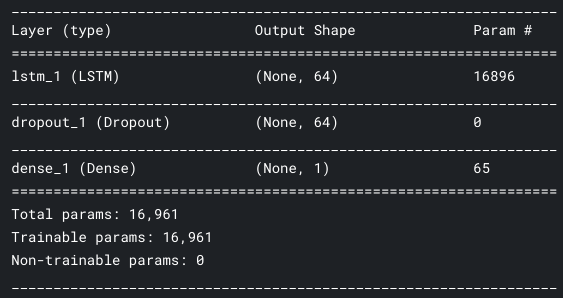
1. 觀察資料數值分佈，去除離峰值。
2. 觀察資料欄位。若發現NA，則根據商店、商品以及日銷售量的欄位取平均值來填補。
3. 將店名相同但商店id不同的資料統一。
4. 根據店家名稱，分析商店所在城市。
5. 根據商品項目，分析主要類型以及次要類型。
6. 對城市、商品主要類型和次要類型做label encoding。
7. 根據商店和商品資料計算商品的月銷售量，並將月銷售量小於0的欄位補0，大於20的欄位補20。(比賽有特別說明月銷售量只會介於0到20之間)
8. 透過mean encoding的方法增加資料欄位。針對日銷售量、商品id、商店id、商品項目id、商品主要類型、商品次要類型的欄位以不同方式組合做mean encoding。
9. 計算商品日銷售額和同商品平均銷售額之間的差異。
10. 計算商店日銷售額和同商店平均銷售額之間的差異。
11. 計算每間商店、每樣商品和前一次售出日日銷售量之間的差異。
12. 計算每樣商品和前一次售出日日銷售量之間的差異。
13. 計算每間商店、每樣商品和售出最少日日銷售量之間的差異。
14. 計算每樣商品和售出最少日日銷售量之間的差異。
15. 增加月份以及當月天數的欄位。

總體來說，資料前處理部分有去除離峰值、填補空缺值、針對有字串的欄位做label encoding、透過mean encoding的方法增加新欄位、以不同方法計算日銷售額的差異以及增加日期天數的欄位。

除此之外，將34個月的資料內容切割成前32個月訓練集，第33個月驗證集和第34個月測試集。由於前3個月在產生新欄位時，欄位內容無可追朔性，所以刪除前3個月的資料。

* + 模型說明

LSTM



* 1 LSTM layer
* 1 Dropout layer
* 1 Dense layer
* Loss: root mean squared error
* Optimizer: adam
* Batch\_size: 4096
* Epochs: 10

XGBoost

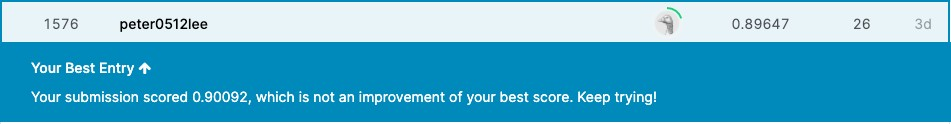
* max\_depth=12
* n\_estimators=300
* min\_child\_weight=162
* colsample\_bytree=0.6
* subsample=0.8
* eta=0.008
* seed=42

Randomforest

* max\_depth=8
* n\_estimators=70
* random\_state=0
* n\_jobs=-1

1. 模型預測結果
   * 參賽結果

於12726名對手中排名第1576名，成績為0.89647。



* + 比較多個模型的預測效果

|  |  |  |  |
| --- | --- | --- | --- |
|  | Kaggle Score | | |
| LSTM | 1.01950 |  |  |
| XGBoost | 0.89647 | 1.02610  (w/o lag feature) | 0.92784  (paper) |
| RandomForest | 0.94685 | 0.93215  (w/o lag feature) |  |

從表格可以得知，XGBoost的效果最好，以及有沒有做lag feature其實對分數是有差距的，有做lag feature不管是在哪個模型，分數都有顯著的提升！

1. 開發最耗時的部份與原因
   * 遇到的問題

這比賽的資料是俄文，像是shop\_name、item\_category\_name，這些欄位其實都是俄文，我們又要取出店名中城市的部分，所以我們就瘋狂翻譯ＸＤ

* + 未來可改善的地方

LSTM我覺得很有機會做得更高，我們光是沒做什麼前處理就有1.02 score，我覺得未來前處理做多一點很有機會超越XGBoost

1. 小組互動照片
   * 合照、討論專題時的照片



1. 專案開發心得

1061443 - 專題開發心得

這應該算是我第一次正式參加完整個kaggle比賽，成績方面我其實算是蠻滿意的，有達到我們當初設下的目標，前15%，可喜可賀。透過這次的比賽其實學到很多工具跟資料前處理的方法，工具的部分我們有使用optuna這個框架去做參數優化，如果我今天沒做這個專案我還真的不知道ＸＤ，這個框架真的很好用，大家可以去用用看！再來就是資料前處理，lag feature也是我第一次接觸，原來lag feature在時間序列分析這方面的問題是這麼有用的，不管是什麼模型，有做lag feature就是硬加個10%，真的很厲害，也讓我體認到模型優化其實不是最重要的，資料前處理也是很重要的一環。最後感謝我的隊友互相分工，感謝各位！

1061416 - 專題開發心得

這是我第一次參加kaggle上的比賽。由於kaggle平台上有些其他人的開源程式碼可以參考，所以我可以透過平台上的開源程式碼，學習到時序性資料的許多處理方式，也可以認識到哪種類型的題目，大家都會選用甚麼樣的模型。在尋找相關資料時，我看到很多人都有使用lag feature和mean encoding的方法來增加資料量，於是我和我的組員也決定實做這個方法。整體的資料前處理，我們採用了平常比較常聽到的label encoding的方法，並去除離峰值、填補缺失值，再加上mean encoding的方法增加新欄位。我們也有比較過有使用mean encoding方法和沒有使用mean encoding方法的差異，發現有使用mean encoding的方法效果真的會好很多。同時，在查詢資料的過程中，我們也有找到optuna這個參數調整的架構。透過xgboost加上optuna，我們可以輕鬆地找出最佳的參數組合，大幅省去我們嘗試不同參數的時間。整體來說，整個比賽過程獲益良多。為了提升準確度，嘗試多種不同的前處理方式，也更加了解模型裡面參數的意思。

1063337 - 專題開發心得

資料收集過程中首次接觸到lag feature與mean encoding，經過比較後發現透過這些feature可以讓模型的準確度大幅提升，這時候才發現自己對於資料前處技術理的視野仍過於狹窄。透過組員介紹而得知Optuna，比較可惜的是我所負責的RandomForest Regression無法透過GPU訓練，難以藉由Optuna來自動優化參數。在kaggle中大致瀏覽過就會發現，在參考其他高手分享的EDA的時候，也要把下方的comment看完，在comment中能看到他人的提問，當我看完原作者的解答後可以快速瞭解原始碼其中的細節。課程中所學的模型大部份都是第一次使用，透過期末專題將不同模型、不同前處理做交叉測試與比較，讓我更加瞭解機器學習這門學問的意義，而lag feature、mean encoding與Optuna是本次專案中最大的收穫。