機器學習 期末報告

玉山 AI

B08901049 張原嘉

B08901010 莊承霖

B08209023 范傑翔

B09901193 林熙哲

一、 題目介紹及動機

本次期末專題我們參加由玉山銀行舉辦的信用卡消費類別推薦，訓練資料為 50 萬名顧客在 23 個月消費的 3 千多萬筆消費紀錄，預測資料為這 50 萬名顧客在第 24 個月消費金額類別前三名，其中類別限定在 16 種類，並且前三名類別不得重複。

每一筆消費紀錄包含消費月份、顧客編號、消費類別、消費次數、消費金額、婚姻狀態、性別代碼、年紀等 54 個欄位。

評分標準採用 NDCG（Normalized Discounted cumulative gain） 進行評分[1]，最後的預測結果會落在 0 到 1 之間，數字越大代表預測結果越精準。

訓練動機為我們可以透過顧客過往的消費紀錄預測顧客真正感興趣的消費類別，因而精準投放不同廣告給不同的消費者，避免消費者一直收到不感興趣的廣告。

二、 資料前處理/特徵工程

原始資料擁有以下特性：

1. 需要預測的人數眾多
2. 總消費次數極大
3. 每人都具有不同的消費次數
4. 消費月份分布不均

因此，我們需要針對個人的消費資訊進行分類與標準化，將流水帳分類成單人在不同月分中的消費情形，保留其月份資訊。之後，我們採用標準化的方式將消費金額資訊轉化成每個月份該使用者各個類別的消費比例。

由於某種類別「在全部 23 個月的消費總次數」對結果造成的影響大於「在全部 23 個月的消費總金額」（詳見第四部分），這樣的處理不僅可以保留重要的資訊，也可以把特定月份造成的影響過濾掉，舉例來說：

某人只在 dt 為 10 時，購買了 type1 的商品，花費 100 萬，但他在每個 dt 都有購買 type2 的商品，花費 1000 元。假如將消費金額化為比例，模型就會預測該消費者在 dt=24 購買 type2 商品的機率較高，符合我們直覺上的預測。

操作步驟：

1. 減少資料量

由於原始檔案有高達 3000 萬筆的資料，我們先針對 23 個月進行檔案拆解，分成 23 個小檔案，避免記憶體瓶頸。

另外，根據先前實驗得到的結果，我們發現檔案中提供的其他資訊，包含副卡、卡片種類編號、客戶來源、職位、國籍、婚姻等，對預測結果的影響都非常小。因此我們僅留下時間、顧客編號、消費種類、消費金額四項資訊，以降低檔案大小。

而題目要求以 dt 為 24 做為目標進行預測，因此我們可以把月份作為最小單位進行整理，無須留存每一筆消費資料與順序。

2. 從時間上的流水帳，轉為對單一顧客之消費者分類帳目

因初始資料為時間上的流水帳，我們需要將資料從「對時間排序」，轉為對「消費者（顧客 id）排序」。我們綜合每位消費者在該月份內的每筆消費資料，將種類與金額轉化為 linked list 的方式儲存。

3. 針對單一顧客在單一月份內的消費金額與種類進行排序

承接上一步驟，我們將同一消費種類的金額進行加總，並進行標準化，得到之資訊為：「某 ID 於某月份消費的金額比例」（該類別無消費則填入 0，總和為 1）。之後，我們再依照 ID 進行排序（後續處理上，可以避免 ID 順序造成的麻煩）。

但在此處，因程式撰寫上的失誤，我們只針對了種類進行排序，並未同時改變消費比例的 index（排序時使用 list，並沒有使用 tuple 綁定消費金額）。雖然原本絕大部分的消費資料都已經根據消費類別進行排序（類別數字大的在後），但若消除這個錯誤，也可稍微改進模型的結果。

經此步驟，我們去除那些「消費金額極大、消費次數極少」的資料對最後預測結果的影響，又同時保留了不同種類消費金額大小的資訊。

4. 月份資料合併

我們對消費者在 23 個月中，總共在幾個月內有消費紀錄進行分群輸出，共分四類。

(1)全部月份皆有消費紀錄: dt_all.csv
(2)共 10 ~ 22 個月份有消費紀錄：dt_10to22.csv
(3)共 2 ~ 10 個月份有消費紀錄：dt_less10.csv
(4)只有一個月份有消費紀錄：dt_only1.csv

5. 資料連續化

將上一步驟產出的(2), (3), (4) 三個檔案，將有消費紀錄的月份往後面放，使其皆緊接於最後一個月份，這樣的連續化使資料得以放入模型預測，示意圖如下圖所示：

dt	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
input		v1								v2					v3			v4	v5				
output																			v1	v2	v3	v4	v5

上圖中，輸入的資料共有五個月份有消費資訊，經由步驟五，將有消費資訊的資料依序排在最後面之月份，方便往後模型針對dt為24進行預測。

經由以上五個步驟，我們成功將資料分為 4 種 case，且符合以下四條要件：

1. 對消費者進行整理與排序

2. 對消費時間，以月作為單位排序

3. 單一月份中個別種類消費金額標準化

4. 所有資料皆密集排列，並接近欲預測之時間點

三、 模型介紹

模型 1：neural network model

第一種模型我們用 neural network 來預測資料[2]，整個 neural network 參考作業三的 CNN 模型，我們取最後兩個月的消費資料，先將 50 萬筆顧客分成三類：

1. 完全沒有消費的顧客，約有 12 萬名顧客。由於我們沒有這些顧客的資料，他們前三名的消費類別預測成 37、15、2，是全部消費紀錄統計下來前三多筆的類別。
2. 需要預測的 16 種類別中有消費至少 3 種類別的顧客，約有 16 萬名顧客。他們前三名的消費類別預測成消費最多金額的前三類別。
3. 需要預測的 16 種類別中有消費紀錄但是未滿 3 種類別的顧客，約有 22 萬名顧客。

我們希望使用 neural network 預測這 22 萬名顧客前三名的消費類別，預測方式如下述，我們利用第二類顧客，及消費最多金額的前三類別當成訓練資料。模型輸入共有九個欄位，分別是婚姻狀態、學歷代碼、行業別、國籍、職位別、客戶來源、正卡信用額度、性別代碼、年紀與正附卡註記，label 為顧客消費最多的類別，用上述資料來訓練 neural network，預測資料為第三類顧客，輸入為和訓練資料相同的九個欄位，輸入為用 neural network 輸入 16 維中前三大的維度。

以下為模型的超參數

num_epoch=30

batch_size=128

valid_ratio=0.12

以下是 neural network 的架構

```
self.fc = nn.Sequential(
    nn.Linear(9, 64),
    nn.ReLU(),
    nn.Linear(64,64),
    nn.ReLU(),
    torch.nn.Dropout(0.5),
    nn.Linear(64, 16)
)
```

模型 2：LSTM 加上 Gradient boosting regression

模型敘述：

狀況 1：對於消費月份數**大於等於 10** 之 id

使用 LSTM 抽取 feature，然後使用線性模型作為 classifier 來輸出 16 維的向量作為 label 進行訓練與預測。

狀況 2：對於消費總月份數**小於 10 且大於 0** 之 id

使用 sklearn 提供的 GradientBoostingRegressor 模型並搭配使用 MultiOutputRegressor 這個函數讓我們可以以 16 維的向量作為 label 進行訓練與預測。

訓練：

狀況 1：對於消費月份數**大於等於 10** 之 id

使用 dt_all 的資料作為訓練集，將 dt=1～10 的資料作為 input feature，dt=11 為 label，共約 10 萬筆訓練資料，將此資料集訓練出之 model，作為 LSTM_model 對消費月份數大於等於 10 之資料進行預測。

狀況 2：對於消費總月份數**小於 10 且大於 0** 之 id

使用 dt_all 資料作為訓練資料集，將 dt=22 作為 input feature，dt=23 作為 label 進行訓練，將其訓練 model 作為 Regression_model，對消費總月份數小於 10 筆之資料進行預測。

預測：

狀況 1：對於消費月份數≥10 之 id 預測模型

使用 LSTM_model，將符合條件之最後 10 筆月份資料(dt=14～23)作為input feature 輸入進 model 內（backbone），得到個別月份的 output 後加總，再丟進 classifier（header）預測結果，而與作業四 model 的差異為最後 output 部分，輸出為 16 維，並且將 16 個資料找到前三高之結果作為預測結果。

狀況 2：對於消費總月份數**小於 10** 之 id

使用 Regression_model，將資料月份數小於 10 之資料，以 dt=23 作為 input 進行預測，所得到的結果會是一個 16 維的向量，其數值代表模型預測的 16 個類別的消費比例，因此我們將這十六維向量中數值最高的三個維度所代表的類別作為預測結果。

合併預測結果：

由於我們是使用兩種不同的模型來應付不同消費頻率的顧客，因此我們最後需要將預測結果進行合併，以得到最後用來繳交的預測。另外，因為有少部分資料並沒有在預測目標的 16 種類別內消費的紀錄，會在預處理時被去除掉，所以無法利用模型解決。對於這些顧客，我們直接利用統計兩個 model 預測次數前三高的種類，作為最後的結果。

四、 實驗及討論

1. 資料型態與處理

(1)我們過 simple baseline 時使用統計方法，將所有消費紀錄中最多金額的前三類別統計出，直接把所有顧客的預測類別前三名成這三個類別，最後得到的 accuracy 為 0.1563，沒有過 simple bseline，然而我將所有消費紀錄中最多次數的前三類別統計出，直接把所有顧客的預測類別前三名成這三個類別，最後得到的 accuracy 為 0.2406，超過 simple bseline。從上述的實驗中我們發現比起總消費金額，總消費次數有更大的影響，舉例而言，有可能發生單筆消費很高（買車），但是往後出現同一消費類別的機率較低(不會每個月都再買車)，然而如果我們消費次數很多（買衣服），我們之後消費同樣類別的機率會比較高。

(2)在第一種模型，我們預測的結果大多都是 37、15、2，這些都是出現比較多次的種類，然而對於出現比較少次的種類，由於輸入的次數比較少，預測到的機率也比較少，另外我們從 valid accuracy 發現只有 20% 的機率能預測出最多可能的種類，如果想要提升最後的 accuracy 我們可以將輸入資料量提升（不僅是最後兩個月，而是全部的資料，但是這樣的時間需求也會提升），另外我們可以去調參數得到更好的 accuracy，最後我們可以去試不同 neural network 的架構，觀察要疊多少層能夠得到好的預測率。

(3)我們處理巨量資料時不可能將全部原始資料放進模型裡面訓練，這樣會花太久時間，而且最後得到的訓練結果也未必符合預期，可能會有 overfitting 的風險，我們必須使用特徵工程將原始資料中較重要的類別以降低計算量。這步驟牽涉到我們對問題的背景知識，這樣才能用可接受的時間得到好的預測結果。

(4)我們處理真實資料時會有極端值、資料欠缺，資料誤植的情況，如果直接丟到模型裡面訓練，這些資料會影響預測結果使其與實際結果產生偏差，因此我們必須做資料前處理，將欠缺的資料補齊，並且將資料做轉換，使得原本極端值的資料轉換後變得較不極端。

2. 模型內容

(1) LSTM 訓練時僅輸入「全部 dt 都有資料的 case」當作訓練資料，而且只使用 dt 為 1～10 當作 input feature，dt 為 11 當作 label，我們目前有想到幾個改進方式都能大幅增加我們用來訓練的資料量，以增進 model 的表現：

(一)固定 input feature 長度，將 input 的時間區間多次平移：

使用相同長度的時間區間，但是起始點不同，例如：先使用 dt 為 1～10 的時間區間作為輸入，將 dt 為 11 當作label；再使用 dt 為 2～11 的時間區間作為輸入，將 dt 為 12 當作 label，以此類推。如此一來可以重複使用訓練資料，以提高資料使用的效率。

(二)使用不固定長度的 input：

雖然我們使用的是 LSTM 模型，但是我們在訓練和預測的時候其實都固定了輸入的區間長度，這麼做失去了 LSTM 模型輸入長度可以不固定的優勢，也限制了我們可以使用的資料數量。

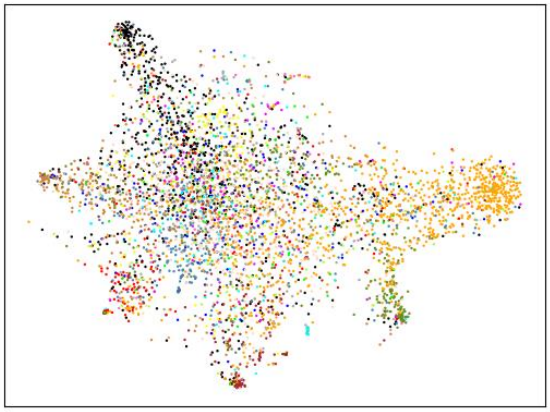
若想要將我們現行的模型改進成可以輸入不固定長度，需要調整將 LSTM 抽取出來的 feature 輸入進 header的方式，目前我們是將輸入 model 後，各個 dt 分別得到的 output 進行加總後輸入 header 進行分類。若將 LSTM 的輸入長度改變，那麼長度越長的 input 總和就會越大，因此我們需要將 LSTM 輸出的 feature 改成加權平均（使 dt 靠後的 output 更重要），或是不選取全部的輸出，只使用最後一個 output（因為將最後一個 dt 輸入LSTM 時，LSTM 的 cell 中已經儲存了先前 dt 的資訊）為 header 的輸入。

在這次比賽的過程中，我們是將顧客依照消費的頻率來分類，然後使用兩種模型（Gradient Boosting Regressor 和 LSTM）針對不同消費頻率的顧客進行預測，在 LSTM 中我們的 header 是使用簡單的線性模型，事實上這個 header 可以改成使用 Gradient Boosting Regressor，再加上如果我們可以將輸入改成不固定長度的，那麼我們就可以把現行的兩個模型直接結合起來，用來預測各種消費頻率的顧客。

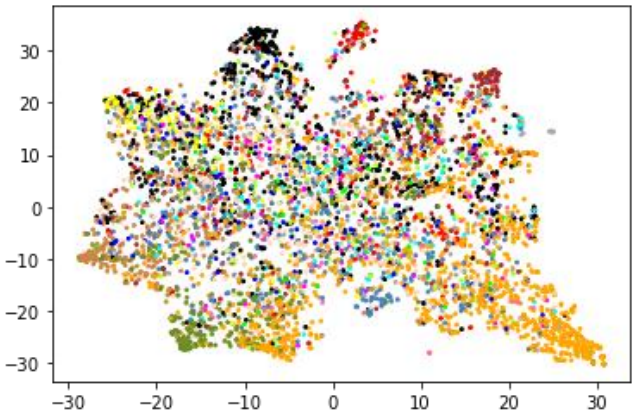
五、 視覺化

視覺化的部分我們是使用 TSNE，我們嘗試使用了兩種不同的 feature 作為輸入來進行視覺化，第一種直接使用資料預處理後得到的消費比例做為輸入，另一種則是使用 LSTM 模型所抽取出來的 feature 來進行視覺化。

受限於 TSNE 的計算速度，我們只有使用所有數據中的 5000 位顧客進行視覺化，並使用第 23 個月的消費比例占最多的類別作為代表該顧客的消費類別（下圖中資料點的顏色）。若是使用消費比例做為輸入，就是將各個顧客在這幾個月的消費比例合併成一個較長的向量後輸入 TSNE；若是使用 LSTM 抽取 feature，則是會像進行預測時類似，將 LSTM 在各個月輸出的結果平均後輸入 TSNE。所得到的結果如下：

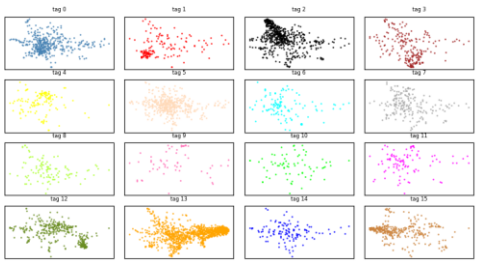


使用消費比例 （圖1）

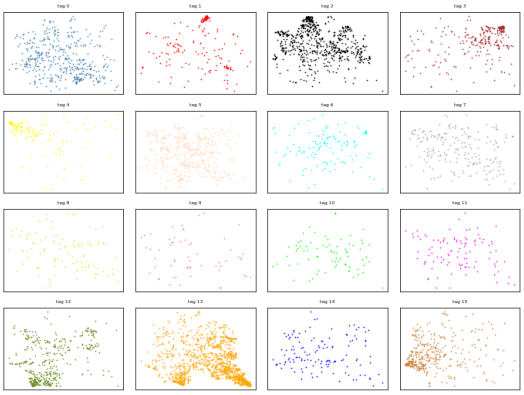


使用 LSTM 抽取 feature（圖2）

可以發現兩種方法所得到的結果都沒有辦法將數據點良好的分開，因此我們將不同消費類別的顧客分開繪製圖表，結果如下：



使用消費比例 （圖3）



使用 LSTM 抽取 feature（圖4）

從個別繪製的圖表中我們觀察到直接使用消費比例來進行視覺化時，在某些消費類別都可以看到資料點的確有聚集在一處的效果，但是許多類別仍然會散布在中央的部分造成最後結果較差。然而使用 LSTM 時，資料點聚集在中央的情況較不明顯，但是各個類別有時不會聚集於一處，而是有兩三個較密集的聚集處，造成最終視覺化的效果不好。

另外，由於我們在視覺化這部分是使用該顧客在第二十三個月時消費金額最高的類別來代表，因此可能會發生該顧客在前兩高的消費金額差距不大的問題，也就是說圖表中的顏色或許沒有辦法良好的代表該顧客的種類，這也是一個造成最後視覺化效果不佳的可能原因。

六、 結論

1. 雖然說我們目前的模型可以通過助教所設定的 baseline，但是在資料的使用效率上、模型的設計上都有許多可以改進的地方。
2. 我深刻體會到了「Garbage in, garbage out」的原則，**資料前處理絕對是整個任務最重要的部分**。為了讓資料能丟進模型，我花了非常大量的時間，查詢非常多 Dataframe 的處理方式。但在解決之後，模型的設計就變得相對簡易了！而且還得到了不錯的結果！
3. 在處理不熟悉的問題時，先使用簡單的統計方法或者是簡單的模型來進行觀察可以提供我們許多對於後續資料處理的方法、模型的設計的直覺。
4. 視覺化的部分可能是因為抽出的 feature 不夠好，或者是因為沒有一個較好的方法將顧客分類來指定最後圖表中資料點的代表顏色，所以造成整體視覺化、分群的效果較差。

七、 參考資料

[1] TBrain 信用卡消費類別推薦－競賽說明 https://tbrain.trendmicro.com.tw/Competitions/Details/18

[2] Pytorch 的 Linear Regression https://ithelp.ithome.com.tw/articles/10276281

[3] [PyTorch] Getting Start: 訓練分類器 —— MNIST https://clay-atlas.com/blog/2019/10/19/pytorch-%E6%95%99%E5%AD%B8-getting-start-%E8%A8%93%E7%B7%B4%E5%88%86%E9%A1%9E%E5%99%A8-mnist/