Product Recommendation on Financial product

NTU Data Mining 2016 Fall - Final Project Report
Group 4 謝友恆，陳威宇，林其政

1. Data & Objective

本次 project 參加了 kaggle 上的比賽[1](Santander product recommendation)，比賽內容是金融產品推薦，主辦方將會提供一間西班牙銀行一年半間的客戶資料，包括年齡、客戶等級、工作狀況、婚姻狀況等等。而我們希望透過這些資訊來預測哪些客戶會對哪些新產品感興趣的同時，分析背後的可能原因。

2. Tool

Xgboost (decision tree library), Keras (neural network library), Cuda (for SPM), R (statistic, visualization)

3. Methodology & Implementation

3.1 Preprocessing

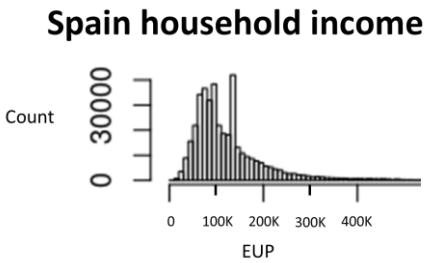
此 Dataset 的每筆資料包含了 Customer, Time , Attributes 與 Products 四個部分，Customer 為每個使用者獨有的 ID，共有 956645 位使用者。Time 是這筆資料出現的月份，範圍是 2015-01~2016-06 共 18 個月，Attributes 是使用者在這個時間點的 24 項屬性，大致上可分為個人資料和銀行相關(如表一)，而 Products 則是要預測的 24 項商品，大致上可分為帳戶、服務、借貸與投資四項(如表二)。

表一: 24 項使用者屬性，粗體為分類，紅字為重要屬性

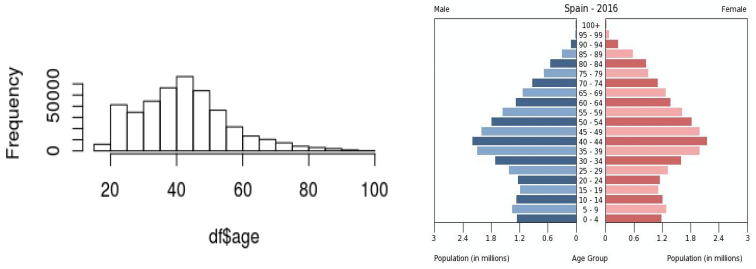
Customer Info		**Bank Relation**	
Identity	**Location**	**Time**	**Specialty**
Customer id	Nationality	Joint date	Active or not (now)
Gender	Province	**Duration of membership**	Active or not (first day in M.)
Age	Province code	New customer or not	Primary or not (ever)
Social status	Residential area	Last date as primary	Primary or not (last day in M.)
Alive or not	Address		Channel used to join
Income	Place of birth		Bank employee or not
			Spouse of employee or not

表二: 24 項商品，紅字為高頻出現，其中 Pensions 有兩項是因為有兩個不同屬性都被解釋成退休金

Account	**Service**	**Loan**
Saving account	Short-term deposits	Loans
Current account	Medium-term deposits	Mortgage
Payroll account	Long-term deposits	Credit Card
Junior account	Payroll	
Más particular Account	Pensions	**Investment**
Particular Account	Pensions2	Funds
Particular Plus Account	Direct Debit	Securities
E-account	Taxes	Derivative Accounts
Home Account	Guarantees	



圖一 資料中的收入分佈，撇開因用平均補缺值造成的突出，是個左偏分佈



圖二 資料中的人口分佈(左)和西班牙人口金字塔(右)，發現兩者呈現相同的趨勢

由於此 Dataset 紀錄了所有使用者在這 18 個月的所有屬性與所有持有的金融商品，因此共有 13647310 筆資料，合計大小 2GB，給分析帶來很大的負擔。然而，大部分使用者都不會經常變動所購的商品，所以會有很多重複的資料。為了輕量化資料，我們使用[2 所實作的方法，先取約 1/3 的使用者，同時只記錄這些使用者每個月多買/少買了那些商品。在化簡過後，總共只剩下 88617 名使用者與 401793 筆資料，大大減少了分析資料的負擔。

此外，有些屬性是有缺失的，如許多人沒有填收入狀況。這裡我們同樣使用[2]的方法，計算各省分地區平均值後。取代各地使用者對應的缺失值。

3.2 Mono attribute analysis

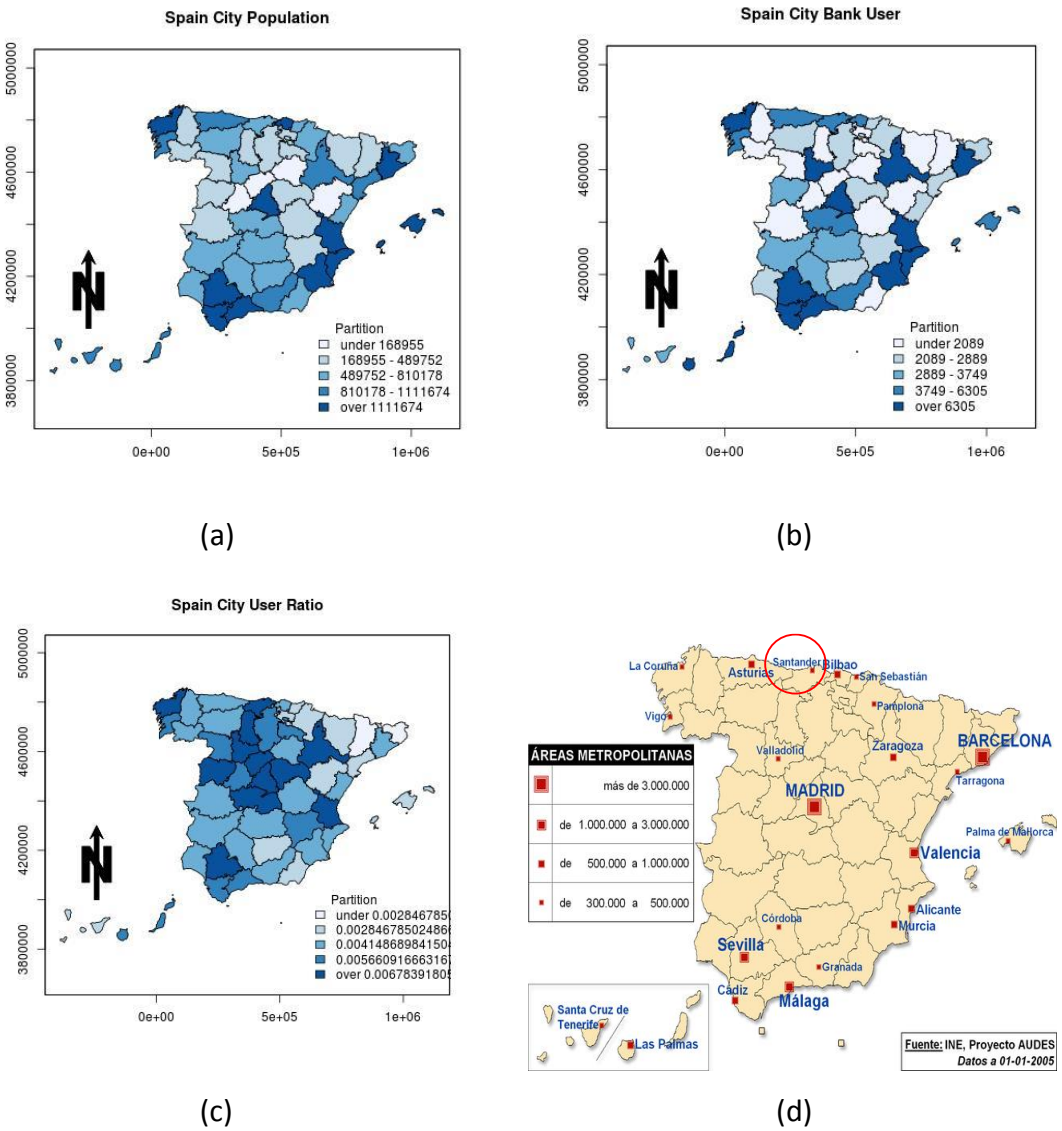
首先，我們先觀察一些比較重要的屬性的分布:

a. Income

我們先統計了所有人的收入分布。如圖一，除了由於缺失值填平均值，造成在 130000 出現了一個不尋常的高峰外，看起來是一個正常的左偏分佈。然而，如果去和西班牙在其他地方的統計數據比較的話，就會發現相當奇怪的事情: 西班牙的每年家庭平均淨收入是兩萬歐元，但是圖表高峰卻在約八萬歐元處。經過仔細查驗後，發現主辦單位為了保護客戶隱私，特別聲明說這些資料不代表西班牙的任何一個人的實際情況，但是考慮到為了讓此資料學出來的東西仍有意義，主辦單位很可能是對其中幾項做了線性調整。也就是說，此收入資料是被線性調整過的，自然也就和真實情況略有差異了

b. Age

接著，我們統計了所有人的年齡分佈。如圖二，可以看出乍看之下有點奇怪的是，此年齡分佈似乎有兩個峰: 主要的峰在中壯年人，但是在年輕人處有另一個峰值。原本認為這可能和不同年齡層是否會買金融商品有關，但實際上查看了西班牙的人口金字塔，發現這其實和真實的人口分布一致。



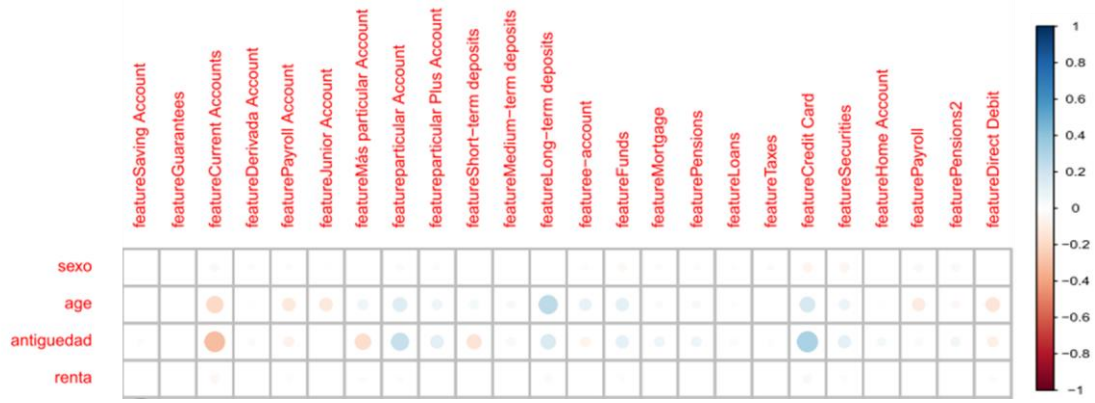
(a) (b)

(c) (d)

圖三 (a) 西班牙人口分布圖，深色為人口密集處，資料來自維基百科 (b) 西班牙銀行顧客分佈圖，可以看出人數多的地方顧客也比較多 (c) 西班牙銀行顧客比例圖，可以看出除了人口密集處外，北方中間的顧客比例也相當高 (d) 西班牙城市圖，發現北方中間的城市即是 Santander (銀行名稱)

c. Population

最後，我們統計了各地的顧客分佈圖。這個步驟其實相當的麻煩，為了能夠使用 R 畫出西班牙地圖並以顏色表示人口密度，我們先參考[3]拿到了西班牙的各省邊界形狀，接著去維基找出了各省的人口數。然而，維基的所用的、地圖上的與銀行資料內的省份名皆有些許落差，因此需要做手動的對應。最後，如圖四(a),(b)畫出了西班牙人口與顧客的分佈圖。能夠看出這兩張圖的趨勢相當一致，但為了進一步分析顧客分佈，我們將(b)的數據除上(a)，得到如(c)的銀行顧客比例分佈。可以發現的是，南方原先人口數多的地區，銀行顧客比例卻不一定那麼高；但在北方中間的部分，卻是原先人口少的地區密度也高了起來。為了解釋這個現象，我們去找了如(d)的西班牙的城市分佈圖，結果發現北方中間的城市正是 Santander – 此家銀行的起源城市。因此可以合理解釋是由於地緣關係，讓這些地方的市占率特別高。



圖四 較重要屬性跟商品間的 correlation，可以看出年齡和辦帳時間(antiguedad)和許多屬性有較大的相關性

d. Other

我們也做了幾項其他屬性的比例分佈，但是發現事實上大部分的屬性不是意義有重複，就是分佈太偏頗(有 90%以上都是其中一項)，刪除掉這些雜項以及難以量化的屬性後(如地址)，我們進入下一階段的屬性與商品分析

3.3 attribute x product analysis

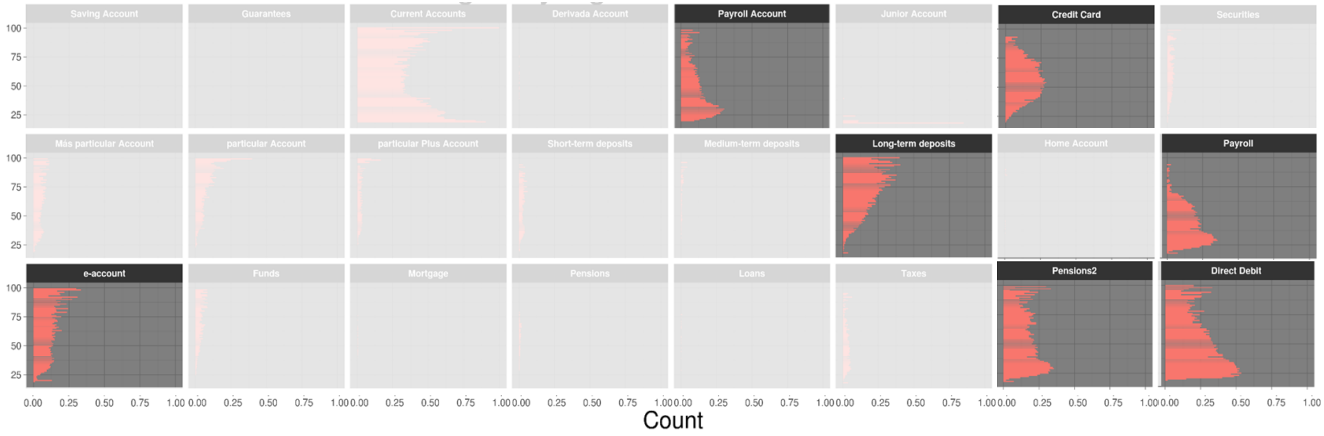
這裡我們觀察各項屬性與產品之間的關係。這邊在處裡時，我們先不考慮時間點的差異，只要這個使用者在這十八個月曾經持有過這項商品，就將其標記為擁有。接著，我們先如圖四，使用 R 的 corrplot library，找出 correlation 觀察各項屬性有哪些和商品較為相關。可以看出，年齡和持卡時間和許多屬性有較大的關係，於是便再細部來看這兩項屬性與各項商品的分佈:

a. 年齡 x 商品

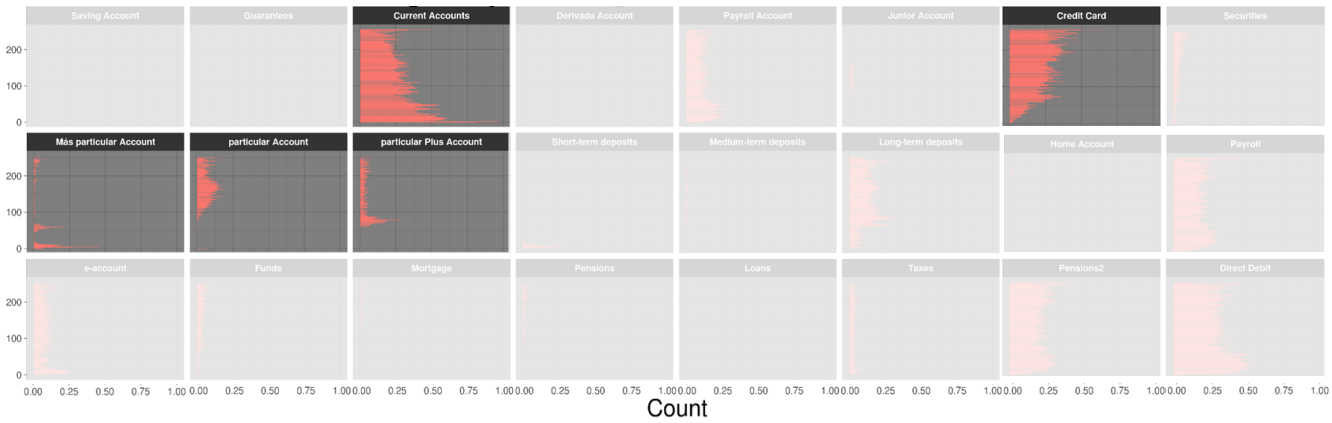
如圖五，可以看出，像是薪資自動匯入帳戶(Payroll account)、薪資匯入(Payroll)、退休金(Pension2)與直接存入帳戶付款(Direct Debit)都和年齡呈負相關。這些都是出現頻率極高的商品，推測是年齡越高越可能已在別家銀行辦過這些項目，就沒有匯薪資跟存退休金等問題。長期置產(Long-term deposit)不太意外的跟年齡呈正比，而電子帳戶(E-account)似乎略呈正相關但原因不明，而信用卡(Credit card)則是集中在中年齡層。

b. 辦帳時間 x 商品

如圖六，可以看出辦帳時間越久就越沒有使用現金帳戶(Current account)，但信用卡的比例卻是逐漸提高，這兩項可能有互補關係? 而有些特別的帳戶是只限在辦帳時間短(Más particular account)/長(Particular account, Particular plus account)的情況，推測是帳戶本身的性質，但是資料並沒有詳細說明這些帳戶的細節



圖五 各商品的年齡比例分佈，將比較沒關係的項目以遮罩蓋住



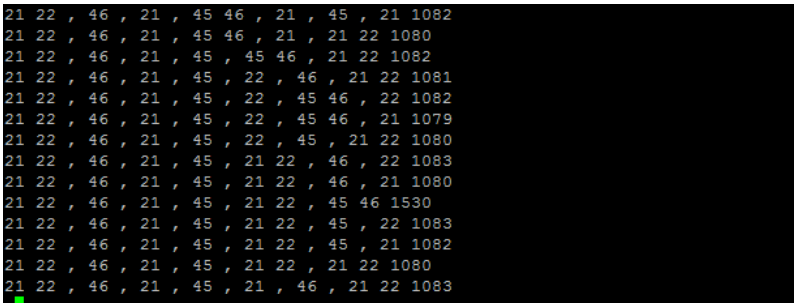
圖六 各商品的辦帳時間(月數)比例分佈，將比較沒關係的項目以遮罩蓋住

3.4 product x product analysis

前面使用的方法雖然能夠看出一些屬性與商品間的相關性，但是只限於兩兩之間的關係，且無法考慮這個時間點購買的商品對下一個時間點造成的影響。為了更進一步找出商品之間的關係，我們實作了諸多 Pattern Mining 演算法中的 Sequential Pattern Mining。

Pattern Mining 很早就被運用在購買行為的分析上。如 Frequent Pattern Mining (FPM)旨在統計多項物品的同時出現的頻率，故可以用來幫助超市決定商品的擺放位置，將會被一起購買的物品放在靠近的區域。另一方面，Sequential Pattern Mining 則在統計跨時間點出現的物品頻率，也因此可以用來分析產品被購買的順序，讓公司更準確地推銷產品。而 project 的資料是銀行在過去十八個月內各個客戶的交易紀錄，因此我們可以得知每個客戶購買商品的順序並以此嘗試分析客戶的購買習性。

實作上，我們自行實作了 GPU 加速的 SPAM algorithm[]。如前面 3.1 所述，我們的資料記錄這些銀行使用者每個月多買/少買了那些商品；因此，所要分析的項目有 24(商品數)x 2(多買/少買) = 48 項。為了方便起見，我們將多買商品對應到編號 0-23，少買商品對應到 24-47，以減少程式計算的負擔。



圖七 實作 SPAM 時所遇到的問題，本身就過於頻繁出現的項目會不停的出現

我們首先試著以 5% support 作為門檻找出高頻率的 sequence。然而，我們很快的發現，絕大部分的結果都充斥著本身出現頻率就很高的項目(如編號 21: 有薪資存入 Payroll)，反而洗掉了個項目之間的關係(如圖七)。為了解決這個問題，我們將出現頻率最高的六項(見前面表二)拿掉，只考慮剩下項目之間的關係。最後，我們得到了幾條的發現:

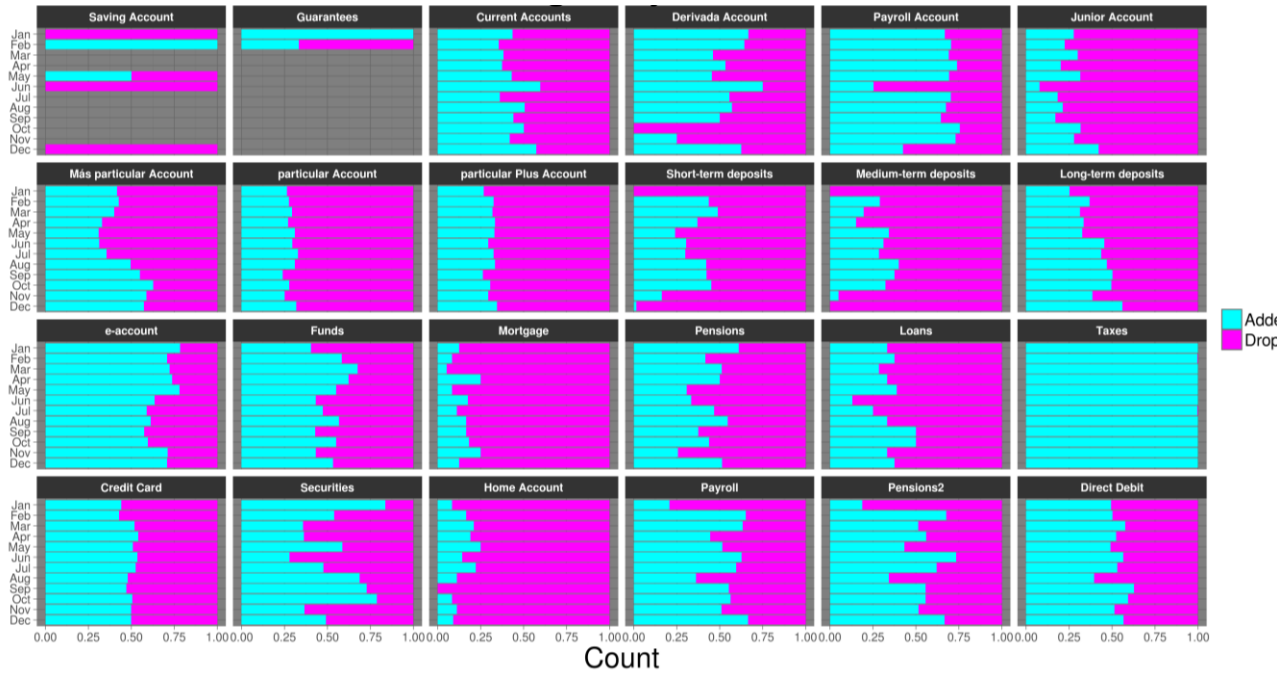
- Pension 跟 Payroll 同時出現機率極高
- 新客戶傾向先辦 Más particular Account 然後換成 E-account
- E account 發展得很好，許多 account 都轉換成 E-account
- 銀行整體存款業務正在衰減
- 使用短期存款的新客戶幾乎都不會繼續使用短期存款。

這些發現其實可以呼應到前面圖表的一些結果。比方說，在 3.3b 中提到，圖六中可看出 Más particular account 只限在辦帳時間短的客戶(即新客戶)，而透過我們演算法的結果發現這些客戶是去換成了 E-account；相對地，有些發現是只有前面 correlation 和圖表看不出來的，如各項存款業務正在衰減。這兩點驗證了我們實作的 SPAM 的可靠性與必要性。

3.5 Product Recommendation

最後，我們要用機器學習的演算法實際預測顧客可能會買的商品。前面雖然找出了許許多多屬性間的關係與趨勢，但是要預測還是需要再重頭開始用機器學習的演算法。比較傷腦筋的是，雖然這是一個有時序關係的資料，但是這個 kaggle 的比賽最後是僅給下一個月的顧客的屬性，而這個使用者可能在之前根本就沒有出現過 - 也就是說，有一部分的顧客我們是沒有辦法拿到他之前的購買紀錄，只能根據其屬性來做判斷。為此，我們將使用者分成了了有出現過和沒出現過兩類。如果是沒出現過的，就只使用屬性訓練出來的模型；如果有出現過，就透過顧客編號找到他之前的紀錄。

要在機器學習中使用不同時間的資料是需要一些額外的心思的，但是在 kaggle 論壇的討論帖[5]中提到，其實購買產品的分佈在不同月份差異很大(如圖八)，而因為題目要求的標的時間點是 2016 年 6 月，故只要使用 2015 年 6 月的資料進行預測就會有不錯的成效了。



圖八 不同月份的商品增減比例分布，可以看出商品隨月份的變化其實是滿劇烈的

以[6]所實作的程式為基礎，我們將 categorical data(如性別、居住地等)轉換為整數，對 ratio data 做 normalization，而資料如前面 3.1 所述是紀錄顧客下個月會多買的商品，且若在一個月中顧客購買複數個產品，則加入複數筆資料。

我們一共實作了三個方法來做辨識:

a. Naïve
   這算是一個 baseline，所有的使用者都相同地以高頻率購買的前幾項作為預測結果，如果低於這個 baseline 則代表學出的東西毫無意義]
b. Decision Tree
   使用 xgboost library 建立 decision tree 模型，並將其中 objective 這個參數設為「multi:softprob」如此一來我們便可得到一 input 對應到各個 output 的機率。
c. Neural Network
   同時我們也嘗試使用現在很流行的 Neural Network 做預測，套件是使用 keras，結構先用最簡單的 2 層，中間為 50 層 hidden layer。

訓練完之後，選擇機率前七高的產品作為預測結果並上傳至 kaggle 評分。kaggle 的評分公式為:

$$\frac{1}{|U|}\sum_{u=1}^{|U|}\frac{1}{min(m,7)}\sum_{k=1}^{min(n,7)}P(k)$$

其中|U|為顧客數量、m 為購買產品數量、n 為預測之購買產品數量、P(k)表示對該產品是否預測成功。觀察此式可知對於每個顧客猜七項產品是不會吃虧的。

表三 不同實作方法在 Kaggle 上的分數(越高越好)與排名(越小越好)

	Naive	Decision Tree	Neural Network
Score	0.014	**0.027**	0.019
Rank (Total: 1735)	1415	**600**	1254

分別將三個方法的預測結果上傳後，得到了如表三的分數與排名。Naïve 的方法自然是最低，這代表我們其他方法學出來的東西是有意義的。然而 Neural Network 的效果較差，我想這一來是因為我們沒有花太多的時間在調整 Neural Network 的架構來得到更好的效果的關係，不過另一方面也是因為使用 NN 的時候，如果本身自己的資料就已經是很高層次的屬性，那麼 NN 就不太能夠發揮其學出良好特徵的能力。而 Decision Tree 的結果還算不錯，和第一名 0.031409 的分數比較起來還算是沒有差太多。

值得一提的是，decision tree 在學的時候，能夠透過看哪些 feature 用來區分的次數較多，來判斷哪個 feature 是在區分上比較重要的。這個值被稱為 fscore。而當我們把所有屬性和商品的 fscore 如表四列出來之後，可以發現和 correlation 的結果略有不同，區分最重要的屬性是 Income(圖四的 Renta)。辦帳時間和年齡也確實是重要的因素，但第四個重要的是 Channel used to join。這項屬性的意義在資料中並沒有說明，其所顯示的值全部都是被以代碼表示的。這個就如同 3.2a 中所提到的，銀行為了保護客戶隱私，選擇性屏蔽了一部分的資料。

表四 所有屬性和商品對應到的 fscore

屬性/商品	fscore	屬性/商品	fscore	屬性/商品	fscore
Income	13359	Active or not (first day in M.)	747	Mortgage	224
Duration of membership	10982	Particular Plus Account	696	Junior account	139
Age	10443	Particular Account	692	Home Account	118
Channel used to join	5268	Active or not (now)	658	Medium-term deposits	60
Social status	1814	Payroll	618	Loans	56
Gender	1547	Securities	571	Nationality	47
Current account	1466	Pensions2	549	Bank employee or not	38
Direct Debit	1332	Place of birth	546	Primary or not (last day in M.)	29
E-account	1289	Funds	437	Residential area	25
Credit Card	1175	Más particular Account	429	Derivative Accounts	21
Payroll account	1069	New customer or not	422	Alive or not	14
Taxes	1005	Short-term deposits	389	Primary or not (ever)	3
Long-term deposits	801	Pensions	336		

這個略有一致但又有所不同的結果，同樣顯示了這個方法一方面是可靠的，一方面又能夠找出用統計的方法所找不出來的訊息。只可惜這只能夠顯現出有什麼可能比較重要，並沒有辦法很明確的敘述這些重要的屬性是對商品預測造成甚麼影響。

4. 結論

在這次的 Project 中，我們使用了多種方法找出使用者屬性與其之前購買的金融商品和之後買的商品的關係。我們先用一般統計方法分析單一屬性分布，並和額外資料做比較；接著用 correlation 找出較有重要關係的屬性後，細部觀察屬性與商品的關係。為了進一步探討多項商品間的關係，我們實作了 SPAM 演算法，最後嘗試多種 Classifier 得到不錯的預測結果，並發現預測中重要的屬性和前面的屬性一定的一致性。我們的貢獻在於整合多種演算法多面向地分析了整個資料庫，並發現了 3.3 與 3.4 中所提到的規則，

Reference

[1] https://www.kaggle.com/c/santander-product-recommendation
[2] https://www.kaggle.com/apryor6/santander-product-recommendation/detailed-cleaning-visualization/notebook
[3] https://procomun.wordpress.com/2012/02/18/maps_with_r_1/
[4] Ayres, Jay, et al. "Sequential pattern mining using a bitmap representation." *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002.
[5] https://www.kaggle.com/c/santander-product-recommendation/forums/t/25579/when-less-is-more
[6] https://www.kaggle.com/sudalairajkumar/santander-product-recommendation/when-less-is-more/output