國立東華大學應用數學系

統計碩士班

碩士論文(口試版)

指導教授：曹振海　博士

Kaggle 資料學習 - Grupo Bimbo inventory demand 與 Rossmann Store sales 資料分析競賽

Learning from Kaggling : Analyses of Grupo Bimbo inventory demand and Rossmann Store sales Competitions.

研究生：林子軒　撰

中 華 民 國 一 〇 六 年 六 月

學位考試委員會審定書

Certificate of Approval of Examination Committee

國立東華大學　應用數學系統計碩士班

研究生 林子軒 君所提之 論文

National Dong Hwa University

The Thesis Graduate Student Proposed

(題目) Title

Kaggle 資料學習 - Grupo Bimbo inventory demand 與 Rossmann Store sales 資料分析競賽

Learning from Kaggling : Analyses of Grupo Bimbo inventory demand and Rossmann Store sales Competitions

經本委員會審查並舉行口試，認為符合 碩士 學位標準。

After evaluation and the oral examination by the committee members, the student complies with the master degree

學位考試委員會召集人 吳章瑩 簽章

The Convener of Examination Committee

委員 張源俊 簽章

Committee Member

委員 曹振海 簽章

Committee Member

委員 吳章瑩 簽章

Committee Member

指導教授 曹振海 簽章

Advising Professor

系主任 (所長) 應用數學系 主任 王昆淙 簽章

The Director of Department

中華民國 106 年 6 月 12 日

ROC Year Month Date

致謝詞

感謝父母提供我經濟上的支助，讓我有機會念研究所。

感謝曹振海老師這兩年耐心的指導，受益良多，教導我許多知識，並在遇到困難時提供我許多意見，對於論文題目，也讓我自由發揮，並無限制我的方向，給我很大的發揮空間，自由對我進行研究方面非常重要，能研究自己喜歡的領域，是很棒的事。感謝吳韋瑩、張源俊與曹振海擔任我的口試委員，並給予我論文上的建議。

感謝本校經濟系張銘仁教授，聘請我擔任研究助理，給我一個學習經濟上資料分析的機會，另外也感謝成功大學林常青教授，對於經濟分析上，提供我計量經濟相關的知識。

感謝趙維雄教授在碩一指導我，對於我的問題也不吝於解答，也提供我擔任助教的機會，藉此培養我上台演講的能力。

感謝 PTT 的版友，提供我許多程式上的協助，熱心回答我的問題，並對我提出許多疑問，讓我了解自己的盲點，所以在論文之餘，我也建立起自己的 github、blog 與個人網站，累積作品，即使論文結束後，我也持續分析不同的 Kaggle 比賽，增加作品數量，並在每個分析結束後，寫下一篇類似論文的 blog 作為介紹。

最後感謝東華提供適合進行研究的環境，這兩年下來，我進步非常多，使我變得更積極，更自動自發學習，未來即使離開學校，我也會持續往我的夢想邁進。

國立東華大學應用數學系統計碩士班 林子軒 謹致

民國 106 年 7 月

摘　要

Kaggle 是一個統計機器學習資料分析競賽的知名平台。 在這個研究中，我們分析先前競賽的 Grupo Bimbo inventory demand 與 Rossmann Store sales 資料。依據私下/競賽後的評分， 在 XGBoost 之上，依據賽後（private leaderboard）分數， 我們建立模型的表現大約屬前 10% 。 Grupo Bimbo inventory demand 這筆資料總共8千萬筆，13個變數。 Rossmann Store sales 總共 一百萬筆，16個變數。 資料事前的處理，變數選擇以及 feature manufacturing 都對模型表現有很大影響， 這方面的經驗，Kaggle（Kernel，Forum）平台使用心得將在本文中分享。

進一步參考：https://github.com/f496328mm

關鍵詞：資料分析，Kaggle，XGBoost。

Abstract

Kaggle is a well-known and very active machine learning analysis competition platform. In this study, we analyze the formerly featured competition, Grupo Bimbo inventory demand and Rossmann Store sales. Built upon XGBoost, our fitted model has achieved top 10% among all competing machines compared with the private leaderboard scores. The preparation and data import, variable selection as well as feature manufacturing all have marked impact on the resultant performance of the fitted model. We will discuss these issues and share our experience in Kaggling.
More at https://github.com/f496328mm

Keywords: Data mining, Kaggle, XGBoost.

目錄

List of Figures

List of Tables

Chapter 1

序論

我們重新分析兩個在 Kaggle 上先前的比賽：Grupo Bimbo Inventory Demand 與 Rossmann Store Sales。 並在最後達到 10% 的排名，在本文中，將介紹我們對於這兩個資料的分析方法。

我們的 Grupo Bimbo 與 Rossmann Store 資料時間長度分別是七個星期與兩年半， 目標是利用過去的資料，分別預測未來兩個星期與48天。 我們將它視爲時間序列問題，先進行資料切割， 分別利用最近一個星期的庫存需求與最近48天的銷售量，當作假的未來(目標)，進行預測建模。

資料中的變數主要都是類別變數，不易直接處理， 故我們將它轉換爲數值變數，藉由不同類別過去的平均表現，來取代類別變數， 進而以此製造特徵變數。 我們認爲過去的平均表現，與未來的目標有高度相關， 又因爲這是時間序列，過去與未來會有一定的相似程度，而我們研究的結論是， 利用過去平均表現作爲變數，有助於提高預測準確率。

由於某些變數可能不重要，而某些變數可能需要額外延伸出新變數。 換句話說，我們進行特徵工程，製造不同於原始資料的變數， 利用類 forward selection 挑選重要變數， 並分別由 Root Mean Squared Logarithmic Error (RMSLE) 與 Root Mean Square Percentage Error (RMSPE) 計算模型誤差，判斷該變數是否重要。 我們使用的模型是 XGBoost 與 elastic net ，主要原因是速度快， 由於 Grupo Bimbo 超過八

千萬筆，一般的機器學習方法，SVM 與 randomforest 速度上不足以進行 forward selection 的多次建模，來挑選重要變數與參數設定。

在預測 Grupo Bimbo 庫存需求上，我們只使用單一模型 XGBoost 進行預測，並利用特徵工程製造14個特徵變數，最後達到不錯的預測。在初始模型上，training RMSLE 與 testing RMSLE 分別是 0.718 與 0.728，而最後進行特徵工程的 fitted model，training RMSLE 與 testing RMSLE 分別是 0.445 與 0.459，降低了將近 40% 的 error。

在預測 Rossmann Store 銷售量上，我們使用 XGBoost 與 elastic net 分別預測，並將預測結果取平均，利用特徵工程製造27個特徵變數，最後也達到不錯的預測。在初始模型上，training RMSPE 與 testing RMSPE 分別是 0.197 與 0.201，而最後進行特徵工程的 fitted model，training RMSPE 與 testing RMSPE 分別是 0.114 與 0.119，也降低了將近 50% 的 error。

我們藉由 Kaggle 學習到資料分析的技巧與程式語法，對於初學者來說，它資料取得方便，提供排名進行比較。當你達到一定的成績以後，它也有提供各種企業，有關資料分析的工作機會。Kaggle 甚至可以累積作品，增加分析不同問題的經驗。

本文後續章節安排，第二章爲資料集，介紹 Grupo Bimbo 與 Rossmann Store 相關資訊。第三章爲結果，介紹資料準備、特徵製造、預測模型、fitted model。第四章爲結論與討論。

Chapter 2

資料集

Kaggle 會提供訓練資料、測試資料與其他相關資料， 其中，訓練資料包含相關變數(x)、目標值(y)，而測試資料只包含變數(x)，我們必須利用測試資料預測目標值(y)。 Kaggle 會提供一個公開的評估準則，來計算得分。

我們將討論兩個資料：Grupo Bimbo Inventory Demand 與 Rossmann Store Sales。 大部分的產業中都會遇到庫存與銷售量的問題，大至 APPLE 公司生產手機，需要預測新產品銷售量如何，小至一般的麵包店， 需要預測每天準備多少麵包才不會產生庫存，造成浪費，降低獲利。 大企業更重視市占率，即使預期產品過多造成庫存，也不希望由於產品不足，造成客户流失。 所以在不同的問題上，注重的目標也不同，重點在於市佔率與庫存成本之間的平衡。

在預測與建模型之前，如果對該領域有相關涉略，你會知道哪些變數是重要的，這將有助於找到相對正確的模型。 所以，接下來我們將介紹，由 Kaggle 提供關於這兩個資料中的相關變數，這將有助於了解問題。

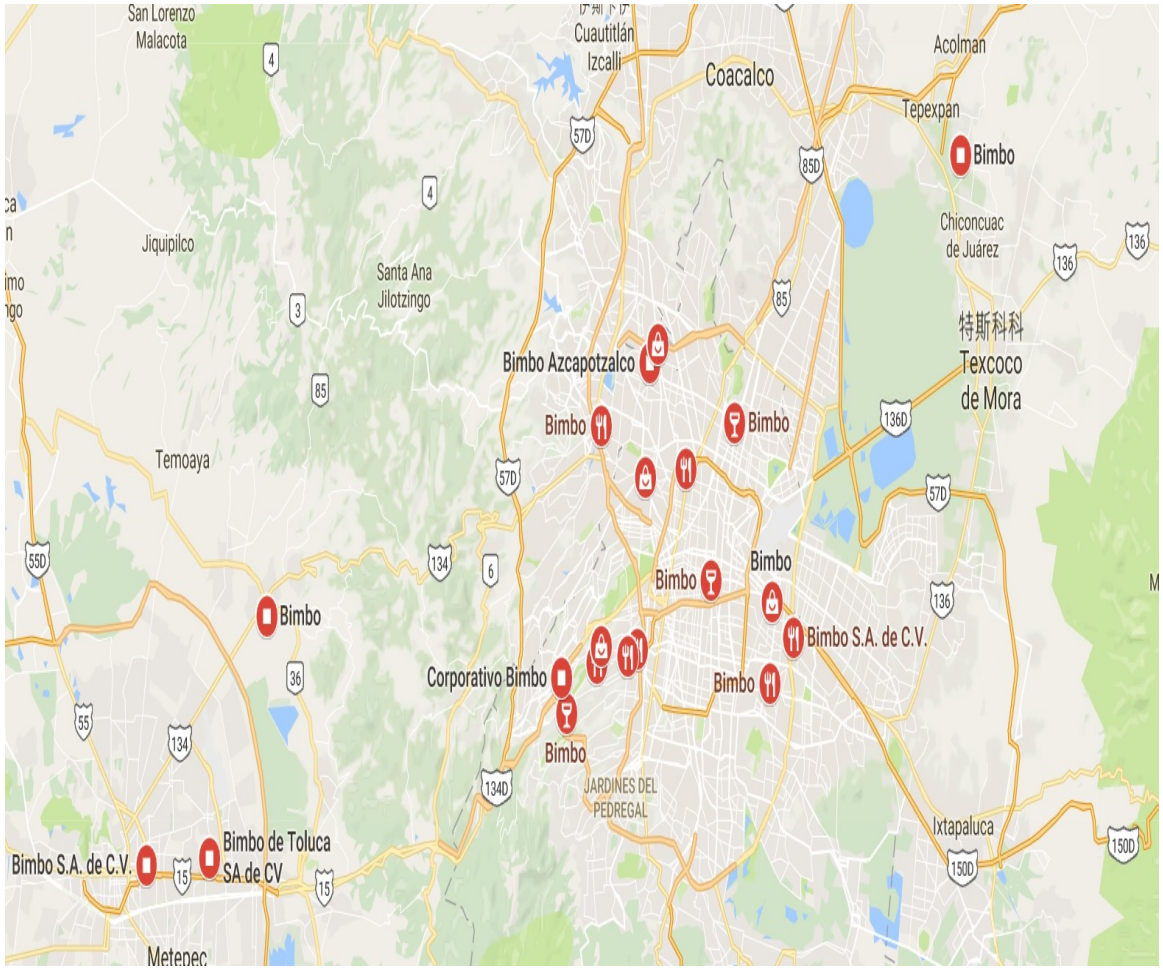

Figure 2.1: Bakery

2.1 Grupo Bimbo Inventory Demand

該庫存需求資料，來自於墨西哥的一家公司 — Grupo Bimbo，這是個麵包連鎖店公司，每個禮拜經由墨西哥四萬五千條路線上，提供相關產品給一百多萬家連鎖店。由於庫存過期，每周損失三萬美金，所以該公司希望預測產品需求，藉由良好的預測，減少成本負擔，進而提升獲利。

Figure 2.1 是 Grupo Bimbo 在墨西哥某一區域的連鎖店分布圖，我們可以看到，每週送貨車經過一些渠道(channel) 與路線(route)運送產品，一部分商店位於郊區，一部分商店位於市區，而由於地區型態等因素，每家商店販賣不同產品。

需要注意的是，因爲有新產品不斷加入銷售，某些產品可能只出現在測試資料(testing data)中，在訓練資料(training data)並不會出現，這是合理的。

接下來我們將介紹，資料中相關的變數。我們的目標變數是 `Demanda_uni_equil`，以下變數來自於 Kaggle - Grupo Bimbo Inventory Demand (2016):

`Demanda_uni_equil`	-	Adjusted Demand (integer) (Target)
`Semana`	-	Week number (From Thursday to Wednesday)
`Agencia_ID`	-	Sales Depot ID
`Canal_ID`	-	Sales Channel ID
`Producto_ID`	-	Product ID
`Ruta_SAK`	-	Route ID (Several routes = Sales Depot)
`Cliente_ID`	-	Client ID
`Venta_uni_hoy`	-	Sales unit this week (integer)
`Dev_uni_proxima`	-	Returns unit next week (integer)

根據我們的變數選擇得知，其他變數是相對不重要的，另外放在 Appendix 5.1。如何找出重要變數的方法，將在稍後的章節介紹。我們期望最大化銷售(sales)，最小化損失(returns)，則成本將會降低，利潤將會提高。評估準則是根據 RMSLE，接下來，我們將介紹另一個資料集。

Root Mean Squared Logarithmic Error (RMSLE)

$$= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(log(p_i+1) - log(a_i+1))^2}$$

p_i is your prediction of demand, and

a_i is the actual demand.

2.2 Rossmann Store Sales

該銷售數據是來自於—Rossmann，一家歐洲藥妝連鎖公司。 而該資料包含超過兩年半的銷售量紀錄， 我們需要預測未來48天的銷售量。 該連鎖商店主要特點是，由於某些店家靠近學校，所以在假日時的銷售量， 與平日會有差異。 舉例來說，在週末、學校假日與特殊節慶， 因爲學生不會來學校，銷售量因此減少。 某些商店可能靠近市區，所以連鎖店密集度特別高， 某些商店靠近郊區，密度比較低。 而過於接近的藥妝店，可能會彼此競爭，影響銷售。 大部分藥妝店在星期天，都是休假日。藥妝店也會有促銷活動，可能會提高銷售量。

我們的目標變數是 `Sales`，以下變數引用於 Rossmann Store Sales (2015) :

原始變數 (Original Variables)

Sales	-	the turnover for any given day (Target)
Store	-	a unique Id for each store
Customers	-	the number of customers on a given day
Open	-	an indicator for whether the store was open: 0 = closed, 1 = open
Date	-	date
DayOfWeek	-	1-7 (Monday - Sunday)
SchoolHoliday	-	indicates if the (Store, Date) was affected by the closure of public schools
StoreType	-	differentiates between 4 different store models: a, b, c, d
Assortment	-	describes an assortment level: a = basic, b = extra, c = extended
CompetitionDistance	-	distance in meters to the nearest competitor store
Promo	-	indicates whether a store is running a promo on that day
Promo2	-	Promo2 is a continuing and consecutive promotion for some stores: 0 = store is not participating, 1 = store is participating

衍生變數 (Derived Variables)

yesterday.open	-	由 Open 變數衍生而成。 若該商店昨天有營業，則 yesterday.open =1 若沒有，則 yesterday.open =0
tomorrow.promo	-	由 Promo 變數衍生而成。 若該商店明天若特價，則 tomorrow.promo =1 若沒有，則 tomorrow.promo =0
open.rate	-	由 Open 變數衍生而成。 它的定義是，該店在 2013/1/1 ∼ 2015/7/3 開門營業之比例。

其他變數是相對不重要的，另外放在 Appendix 5.1。如何找出重要變數的方法，將會在稍後的章節中介紹。該資料的衍生變數，yesterday.open、tomorrow.promo 與 open.rate，理由如下：

yesterday.open 代表該商店昨天是否營業。如果該店昨天並沒有營業，今天銷售量可能會提高，因爲顧客昨天無法在該店進行購買，則今天將會購買更多產品。

tomorrow.promo 代表該商店明天是否將進行促銷活動。如果該店明天即將有促銷活動，則今天銷售量可能會降低，因爲大多數的顧客，希望用更低的價格購買商品，所以他們今天不會購買。

open.rate 代表該商店在這兩年半的資料中，開店營業比例。如果該商店經常不營業，那麼顧客可能會傾向在其他商店購買商品，造成該商店銷售量降低。

在額外製造這些變數前，我們做了以下比較，確保我們的想法正確。以下表格中:

yesterday.open : 1 代表明天營業，0 代表明天不營業，結果與我們的想法相同，昨天營業與否，在平均銷售量之下，差距約 1400。

tomorrow.promo : 1 代表明天將進行促銷活動，0 代表明天無促銷活動，結果與我們的想法不同，但是在平均銷售量之下，差距約 2200，也就是說，該變數在不同狀況下具有不小的差異，我們嘗試將它加入模型中，error 確實降低，因此我們將它加入 fitted model 中。

open.rate : 開店比例。大部分連鎖店在週日都是不營業的，但某部分連鎖店在週日仍然持續營業，即使是國定假日也不休息，該類型商店平均銷售量爲12450，而所有的商店平均銷售量爲6945，因次我們認爲該變數是特徵變數，並把它加入 fitted model 中，也成功降低 error。

yesterday.open	mean.sale	tomorrow.promo	mean.sale
0	8054	0	6164
1	6680	1	8326
open.rate	min	mean	max
-	0.670	0.830	1.000

其他變數，像是明天是否開門、昨天是否特價，我們也曾加入模型中，但是 error 並沒有降低，因此我們不將這些變數加入 fitted model 中。銷售資料所使用的評估準則是根據 RMSPE。

$$\text{Root Mean Square Percentage Error (RMSPE)} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i-\hat{y}_i}{y_i}\right)^2}$$

$\hat{y}_i$ is your prediction of sales, and

y_i is the actual sales.

Chapter 3

方法

本章將介紹三個部分。資料準備、特徵製造與預測模型。

3.1 資料準備

3.1.1 資料大小與輸入資料

該 Grupo Bimbo Inventory Demand 資料大小是 3GB， 包含七個禮拜的相關數據，八千萬筆數據，13個變數，8個類別變數，5個數值變數， 對於大多數資料分析初學者，這是一個相當大的資料。 store sales 資料大小是 36MB， 包含兩年半的相關數據，一百萬筆數據，16個變數，14個類別變數，2個數值變數。

在初步分析 Grupo Bimbo 資料時， 我們使用 `read.csv` 讀取資料， 花費將近 7 分鐘，使用約 13GB 的記憶體。 在記憶體方面的使用量非常驚人， 對於大多數初學者，常見的電腦記憶體容量不超過 16GB。 如果沒有更好讀取資料的方式，那將無法進行分析， 我們也碰到相同的問題，嘗試尋找更有效率的讀取方式。 `Data.table` 套件中的 `fread` 函數， 能夠更有效的輸入資料，相較於 `read.csv` ， 只需要 1.4 分鐘，使用約 4GB 的記憶體， 差距非常驚人。大幅降低記憶體使用量， 對於往後的特徵工程，建立模型，有很大的幫助。

Table 3.1: Grupo Bimbo 資料分割

預測未來 2 周	Week								
眞實情況，要預測 week 10 與 11 的庫存需求	3	4	5	6	7	8	9	10	11
	training data							testing data	
假設 week 11 的 y 與 week 10 相同	3	4	5	6	7	8	9	10	
	training data							testing data	
建立模型，y 是假的 testing data	3	4	5	6	7	8	→	9	
	x							y	
最後預測，時間進行平移		4	5	6	7	8	9	→	10
		x							y

3.1.2 資料分割

在 Grupo Bimbo 資料中， Kaggle 提供第三個禮拜到第九個禮拜的資料， 目標是預測第十與第十一個禮拜的庫存需求。 我們先對問題進行簡化，只預測未來一個禮拜。 對於第十一個禮拜的資料，在時間上假設與第十個禮拜相同， 將問題轉變爲預測未來一個禮拜的庫存需求， 在建模上也會相對簡單。

在以上前提下， 我們將 training data 依照時間， 分割成 week 3 ∼ week 8 與 week 9 兩個資料集， 並假設 week 9 爲 testing data， 使用 week 3 ∼ week 8 進行特徵工程， 藉由預測 week 9 建立模型， 進而建立能夠預測未來一個禮拜的模型。 在未來與過去非常類似的前提下， 當我們能準確預測第九個禮拜時， 相信對於眞正的未來—第十個禮拜的預測，並不會有太大的差異。

實際預測時，資料必須平移。 由於我們的方法，是利用過去六個禮拜製造特徵變數 (x)， 建立模型，所以在預測眞實的 y ， 也必須使用過去六個禮拜的資料， week 4 ∼ week 9 進行特徵工程。 詳細可以參考 Table 3.1。

Table 3.2: Rossmann Store 資料分割

預測未來 48 天	date						
實際 data	2013-01-01	∼	2015-06-13	2015-06-14	2015-07-31	2015-08-01	2015-09-17
	training data					testing data	
建立模型，y 是假的 testing data	2013-01-01	∼	2015-06-13	2015-06-14	2015-07-31		
	x			y			
最後預測，時間進行平移			2013-02-17	∼	2015-07-31	2015-08-01	2015-09-17
			x			y	

在 Rossmann Store 資料中，也是利用相同的概念。Kaggle 提供 2013/1/1 ∼ 2015/7/31 的銷售資料，目標是預測 2015/8/1 ∼ 2015/9/17 的銷售。對於未來48天的銷售量預測，我們進行資料切割，分成 2013/1/1 ∼ 2015/6/13 與 2015/6/14 ∼ 2015/7/31 兩個資料集，模擬預測未來48天，當模擬達到一定的準確率時，實際上預測眞正的未來，2015/8/1 ∼ 2015/9/17，也會達到類似的準確率，詳細切割方法，可以參考 Table 3.2。在這兩個問題上，進行這樣的模擬切割，是不錯的選擇，並得到一定的準確度。

3.1.3 遺失值 (Missing value)

資料缺失在實際問題上，是相當常見的。對於我們分析這兩個資料中，由於特徵工程上的方法，將產生某些遺失值。舉例來說，在 Grupo Bimbo 的特徵工程上，我們使用 銷售站 ID (`Producto_ID`) 與 產品 ID (`Agencia_ID`) 同時製造特徵變數。但是某些產品，過去並沒有在這個銷售站進行販賣，因此它產生遺失值。但是其他銷售站曾經販售該產品，我們可以利用其它的值來進行修補。選擇修補遺失值的方法上，主要考慮到速度與修補的完整性，因此挑選 predictive mean matching 與 logistic regression imputation，這兩個方法。它是藉由 `mice` R package 中的 `mice` 函數來實作。

3.2 特徵製造

3.2.1 特徵生成

在實際資料中，我們需要將變數轉換成特徵。轉換的方法需要進行非常多的測試，進而找出眞正與目標變數相關，並且有效降低 training error 與 testing error 的特徵變數，這就是手動特徵工程 (handcrafted feature engineering)。接下來我們將介紹，我們對於這兩個問題上手動特徵工程的方法。

在 Grupo Bimbo 資料中，我們的目標變數是 Demanda_uni_equil，也就是說，Demanda_uni_equil 是一個最重要的變數。但是，它過於分散，平均高於 Q3，Q3 是 6，平均爲 7.225。由於離群值在整體上佔過多的比例，如果不進行調整，模型將無法進行準確預測。因此，我們將 Demanda_uni_equil 取自然對數(natural log)，轉變成新的變數 — log.due，新變數可以減少離群值造成的影響。另外，由於評分標準是 RMSLE，取自然對數是合理的。

在我們資料中，類別變數占大多數，而常見的方法是使用數值變數來建立模型，因此我們需要對於類別變數進行轉換。方法是在類別變數上，挑選出相同類別的 log.due，再取平均。藉由不同類別，在目標變數上過去的平均表現，取代該類別變數。

舉例來說：

銷售點 ID 這個類別變數是 Agencia_ID = $[1110, 1110...1111, 1111.....2089, 2089...]$，

該變數的種類是 Agencia_ID_class = $[1110, 1111, ...2089]$。

我們創造一個新變數 mean.due.age，代表 log.due 在該類別變數上取平均。以下爲新變數的定義：

$$\texttt{mean.due.age}_k = \frac{\sum_{i=1}^{n} Y_i \cdot I_{\{\Lambda_i = \theta_k\}}}{\sum_{i=1}^{n} I_{\{\Lambda_i = \theta_k\}}}$$

Y_i : log.due [i]。

Λ_i : Agencia_ID [i]。

θ_k : Agencia_ID_class [k]。

for example :

$k = 1$, then $\theta_1 = 1110$, $\texttt{mean.due.age}_1 = \frac{\sum_{i=1}^{n} Y_i \cdot I_{\{\Lambda_i = 1110\}}}{\sum_{i=1}^{n} I_{\{\Lambda_i = 1110\}}} = 2.010$。

我們轉換成以下表格：

Agencia_ID	mean.due.age
1110	2.010
1111	1.643
1112	1.552
⋮	⋮
2089	3.550

該方法是參考 Paulo (2016) 的 kernel 。它的優點是，維度相對較低，一般來說轉換類別變數常見的方法是，指標矩陣(indicator matrix)。例如：該類別變數有三個種類—1、2、3，轉換後變成三維變量— (1, 0, 0)、(0, 1, 0)、(0, 0, 1)。如果類別變數有一百個種類，將會轉換成一百維度的變量，維度變得非常大。但是我們的方法，轉換後只有一個維度。特別的是，指標矩陣認爲不同種類之間，距離相同。舉例來說，使用歐式距離， distance(x,y) = $\sqrt{\sum_{i=1}^{3}(x_i - y_i)^2}$，

distance((1, 0, 0)、(0, 1, 0)) = $\sqrt{2}$。實際情況下，不同種類之間距離可能不相同。而我們的方法，使類別變數不同種類之間，距離不同，它是依據目標變數，來當作距離概念。

另外，我們的方法可以同時針對兩個類別變數進行轉換。

舉例來說 :

兩個類別變數是 銷售站 ID 與 產品 ID ，Agencia_ID 與 Producto_ID.

Producto_ID = $[1212, 1212...1216, 1216.....43201, 43201...]$，

該變數的種類是 Producto_ID_class = $[1212, 1216, ...43201]$，

我們創造一個新變數 mean.due.pa ，代表 log.due 在這兩個類別變數上取平均 ， 以下爲新變數的定義：

$\texttt{mean.due.pa}_k = \frac{\sum_{i=1}^{n} Y_i \cdot I_{\{\Lambda_{i1} = \theta_{k1},\quad \Lambda_{i2} = \theta_{k2}\}}}{\sum_{i=1}^{n} I_{\{\Lambda_{i1} = \theta_{k1},\quad \Lambda_{i2} = \theta_{k2}\}}}$

Y_i : log.due [i] .

Λ_{i1} : Agencia_ID [i] .

Λ_{i2} : Producto_ID [i] .

θ_{k1} : Agencia_ID_class [k].

θ_{k2} : Producto_ID_class [k].

for example,

$k = 1, \theta_{11} = 1110, \theta_{12} = 1212$, then mean.due.pa$_1$ = $\frac{\sum_{i=1}^{n} Y_i \cdot I_{\{\Lambda_{i1}=1110, \quad \Lambda_{i2}=1212\}}}{\sum_{i=1}^{n} I_{\{\Lambda_{i1}=1110, \quad \Lambda_{i2}=1212\}}} = 1.588$

我們轉換成以下表格：

Producto_ID	Agencia_ID	mean.due.pa
1212	1110	1.588
1216	1110	1.426
1238	1110	1.681
⋮	⋮	⋮
43201	25759	1.099

另外在 store sales 資料方面，生成特徵的方法是類似的。我們的目標變數是 Sales，它是最重要的變量，我們一樣對該變數取 natural log 。例如：類別變數是 store ，轉換後的變數爲 mean.sale.store，

store	mean.sale.store
1	8.451
2	8.452
3	8.795
⋮	⋮
1115	8.704

在特徵工程 (feature manufacturing) 之後，我們將進行特徵選擇 (feature selection)，找出重要的特徵，因爲某些特徵可能無法降低誤差。

3.2.2 特徵選擇 (Features Selection)

由於本文中探討的兩個資料中，變數中存在不少遺失值，因此一般常見的 AIC 與 BIC，無法進行變數挑選，所以我們改爲使用類向前選擇 (forward selection) 選擇重要變數。主要是利用 XGBoost 模型的特性，即使資料中存在遺失值，依然能夠進行建模。方法是，將變數加入模型後，測試誤差是否有下降，作爲選擇該變數的標準。

類似的方法還有 backward selection 與 stepwise selection，而我們選擇 forward selection 主要的原因有兩個。第一，速度上的考量，我們的資料約八千萬筆，建模需要花不少時間，而從 forward selection 開始，相對於 backward selection，會節省很多時間。第二，在資料分析中，feature 是額外製造的，我們不可能一次就製造所有的 feature，一般來說，我們是漸進式的製造 feature，可能先製造十個，觀察模型準確度，作爲 baseline，並由此出發，進而找到更多的 feature。根據以上兩點，forward selection 是合理的。

在 Table 3.3 中，是我們分析 Grupo Bimbo 資料時，變數選擇的方法。首先盡可能的製造各種變數，再藉由加入模型中，training RMSLE 與 testing RMSLE 有無下降作爲是否選擇的標準。製造變數的方法如下，對每個類別變數 Agencia_ID、Canal_ID、 Ruta_SAK 與 Cliente_ID 進行轉換，轉換爲數值變數，作爲初始變數(baseline)，基本的模型，接下來額外製造其他特徵變數。

由於初始變數中，只使用一種類別製造特徵變數，接下來將挑選一個以上的類別變數製造特徵。預測的重點是產品 Producto_ID 的庫存需求，而非其他類別變數的需求，因此利用 Producto_ID 與其它類別變數進行結合，創造不同的變數。舉例來說，有100家商店，100種產品，testing data 是預測在a商店的b產品。我們利用[b 產品在這100家商店過去平均庫存需求]、[a 商店過去平均庫存需求]與 [a 商店在 b 產品上的過去平均庫存需求]，分別製造三個特徵變數，將 product id 與其他類別變數進行結合，製造新的特徵變數。

在使用初始變數建立模型時，training RMSLE 與 testing RMSLE 分別是 0.718

與 0.728，當我們進行特徵選擇後，加入的特徵變數有效降低 training RMSLE 與 testing RMSLE。Table 3.3 中，每加入一個特徵變數，RMSLE 便持續下降，最後模型的 training RMSLE 與 testing RMSLE 達到 0.445 與 0.459，與初始變數相比，進步約 40 %。

在 Table 3.4 中，是我們分析 Rossmann Store 資料時，變數選擇的方法，類似於 Grupo Bimbo，藉由將變數加入模型，training RMSPE 與 testing RMSPE 有無下降作爲是否選擇的標準。首先對於 `Store`、`DayOfWeek`、 `Promo`、`month`、`day` 與 `SchoolHoliday` 類別變數轉換爲數值變數，作爲初始變數(baseline)，基本的模型，接下來額外製造其他特徵變數。

在分析 Rossmann Store 資料時，預測的目標是 `Store` 的銷售量，因此利用 `Store` 與其它類別變數進行結合，創造新的變數。初始變數建立模型的 training RMSPE 與 testing RMSPE 分別是 0.194 與 0.201，特徵選擇後，每加入一個特徵變數，RMSPE 也持續下降。值得注意的是，加入 `mean.sale.pi`，error 並沒有降低，但是在加入 `open.rate` 後，RMSPE 大幅降低到 0.168。如果只選擇 `open.rate` 並捨棄 `mean.sale.pi`，error 反而高於 0.168，由此得知，這兩個變數必須同時選擇，才能得到更好的模型。最後模型的 training RMSPE 與 testing RMSPE 達到 0.114 與 0.119，與初始變數相比，進步將近 50%。

在我們的特徵工程上，必須藉由目標變數製造額外的變數，但是在眞正的 testing data 時，目標變數是我們必須預測的。所以在資料分割的章節中提到，我們先進行資料切割，藉由過去的資料進行特徵工程，並沒有使用 testing data 的目標變數。舉例來說，在分析 Grupo Bimbo 資料時，先將資料分割成 week 3 ∼ week 8 與 week 9 兩個資料集，week 9 中的庫存需求，是我們要預測的目標，我們利用 week 3 ∼ week 8 進行特徵工程，預測 week 9 的庫存需求，完全沒有使用到 testing data 中的目標變數。

即使特徵工程的方法，是藉由目標變數所產生，也不影響預測，因爲是利用過去的目標變數，而非未來眞實預測的目標。

Table 3.3: Feature Selection of Grupo Bimbo Inventory Demand Data.

Add Feature	The Feature Meaning	RMSLE of Train	RMSLE of Test
baseline	-	0.718	0.728
+mean.due.pa	the mean of log.due with Producto_ID and Agencia_ID.	0.525	0.536
+mean.due.pr	the mean of log.due with Producto_ID and Ruta_SAK.	0.511	0.525
+mean.due.pcli	the mean of log.due with Producto_ID and Cliente_ID.	0.455	0.467
+mean.due.pcan	the mean of log.due with Producto_ID and Canal_ID.	0.449	0.462
+mean.due.pca	the mean of log.due with Producto_ID, Cliente_ID and Agencia_ID.	0.449	0.461
+mean.vh.age	it is mean of nature log Venta_hoy with Agencia_ID.	0.449	0.461
+sd.due.acrcp	it is standard deviation of log.due with Producto_ID, Cliente_ID, Agencia_ID, Canal_ID and Ruta_SAK	0.446	0.460
+mean.due.acrcp	the mean of log.due with Producto_ID, Cliente_ID, Agencia_ID, Canal_ID and Ruta_SAK.	0.445	0.459

baseline 是使用四個特徵變數 mean.due.Agencia_ID、 mean.due.Canal_ID、 mean.due.Ruta_SAK 與 mean.due.Cliente_ID ，並且利用 XGBoost 建立模型。參數爲 nrounds = 75、eta = 0.1、max_dapth = 8、colsample_bytree = 0.5，隨機取五十萬筆數據，將這些資料以 70-30 比例，分爲訓練資料 (training data) 與測試資料 (testing data) 。之後利用簡單的模型，計算 RMSLE 進行特徵選擇。

Table 3.4: Feature Selection of Rossmann Store Sales Data.

Add Feature	The Feature Meaning	RMSPE of Train	RMSPE of Test
baseline	-	0.194	0.201
+mean.sale.StoreType	the mean of log.sale with StoreType.	0.177	0.183
+mean.sale.Assortment	the mean of log.sale with Assortment.	0.173	0.180
+mean.sale.cd	the mean of log.sale with CompetitionDistance.	0.169	0.175
+mean.sale.promo2	the mean of log.sale with Promo2.	0.170	0.178
+mean.sale.co	the mean of log.sale with CompetitionOpenSinceYear.	0.167	0.175
+mean.sale.p2sy	the mean of log.sale with Promo2SinceYear.	0.167	0.175
+mean.sale.pi	the mean of log.sale with PromoInterval.	0.169	0.177
+open.rate	the mean of log.sale with Store.	0.159	0.168
+mean.sale.store.weekday	the mean of log.sale with Store and DayOfWeek.	0.129	0.135
+mean.sale.store.promo	the mean of log.sale with Store and Promo.	0.125	0.130
+mean.sale.store.month	the mean of log.sale with Store and month.	0.127	0.133
+mean.sale.store.school.h	the mean of log.sale with Store and SchoolHoliday.	0.125	0.130
+mean.sale.store.StoreType	the mean of log.sale with Store and StoreType.	0.123	0.129
+mean.sale.store.Assortment	the mean of log.sale with Store and Assortment.	0.125	0.130
+mean.sale.store.cd	the mean of log.sale with Store and CompetitionDistance.	0.124	0.130
+mean.sale.store.promo2	the mean of log.sale with Store and Promo2.	0.125	0.131
+mean.sale.store.co	the mean of log.sale with Store and CompetitionOpenSinceYear.	0.125	0.130
+mean.sale.sy	the mean of log.sale with Store and year.	0.122	0.128
+mean.sale.swpm	the mean of log.sale with Store, DayOfWeek, Promo and month.	0.113	0.119
+mean.sale.yesop	the mean of log.sale with yesterday.open.	0.115	0.120
+mean.sale.tompro	the mean of log.sale with tomorrow.promo.	0.114	0.119

baseline 是使用六個特徵變數 mean.sale.Store、mean.sale.DayOfWeek、 mean.sale.Promo、mean.sale.month、mean.sale.day 與 mean.sale.SchoolHoliday ，並且利用 XGBoost 建立模型。因爲該資料筆數相對較少，我們取所有的數據進行測試， 參數爲 nrounds = 51、eta = 0.1、max_depth = 10、colsample_bytree = 0.5、subsample = 1 、num_parallel_tree = 9， 將這些資料以 70-30 比例，分爲訓練資料與測試資料。之後利用簡單的模型，計算 RMSPE 進行特徵選擇。

在特徵工程之後， 我們將建立模型，但是機器學習 (machine learning) 有各種模型， 要如何找到適合的模型，是接下來的重點。

3.3 預測模型

3.3.1 模型

準確的模型有助於預測好的結果。例如在迴歸模型中， 我們透過 R^2、MSE 與 p-value 來判斷模型是否適當。 但是在機器學習 (machine learning) 中，並非使用那些準則， 我們希望藉由交叉驗證 (cross validation) 建立好的模型。 好模型意味著，訓練誤差 (training error) 與 測試誤差 (testing error) 非常接近， 且兩者皆足夠小。

接下來我們將介紹新的 boosting 類演算法 XGBoost 。

3.3.2 XGBoost

在機器學習中， random forest 是常用的方法， 但是根據我們在這兩個問題上的經驗， XGBoost 比它更好。 eXtreme Gradient Boosting 是一種 tree 與 boosting 的方法， 它是一種 Gradient Boosting 的方法， 而 Gradient Boosting 主要是透過 最大化減少 loss function 的方向， 進行疊代改善模型， 詳細可以參考 Hastie, et al (2009) 的敘述。

而 XGBoost 優化它的演算法，藉由平行化的方式增加效率， 特別的是，它可以處理稀疏數據 (sparse data)。 這是大部分機器學習中，無法處理的問題， 主要是在樹的分支上，藉由預設方向來解決， 並且利用演算法去找出最好的方向。 詳細可以參考 Chen and Guetrin (2016)。

Table 3.5 中， 我們藉由 Rossmann Store 資料， 進行 XGBoost 與 random forest 比較。 對於我們的分析這兩個問題中，使用 XGBoost 主要是因爲， 速度較快與誤差較低。

Table 3.5: Comparison of time and error.

Rossmann Store Sales dataset, N = 45884				
System	time	training RMSPE	testing RMSPE	parameter
`XGBoost`	0.667 s	0.116	0.120	objective="reg:linear", booster = "gbtree", nrounds=50, eta=0.1, max_depth=5, colsample_bytree=0.45, subsample=1, num_parallel_tree=1
`randomForest`	87.320 s	0.053	0.109	ntree=50, 其他參數使用預設值

速度 (speed)

XGBoost 在速度上優於 random forest。 由於我們在特徵選擇上，使用 forward selection， 必須多次建立模型，比較 error 是否下降， 所以快速的建模預測，對於我們的分析是必須的。

一般來說，R 只使用單核心運算，即使電腦是多核心 CPU ， 但是 XGBoost 的平行化設計，利用所有的核心進行運算， 使它建模速度較原先快上至少一倍。 在 Table3.5 中， 分別利用 XGBoost 與 random forest 進行建模， 資料筆數爲 45884，XGBoost 耗時 0.667 秒， random forest 耗時 87.320 秒。 由此得知，在我們對於 Rossmann Store 資料的比較之下， XGBoost 速度比 random forest 快約 130 倍。 另外，random forest 無法藉由稀疏資料建模， 必須額外花時間修補遺失值。 在 Rossmann Store 資料中，因爲數據量相對少， 所以只花費三秒進行修補， 但是在 Grupo Bimbo 資料中，可能花費超過十分鐘修補。

誤差 (Error)

XGBoost 在預測誤差上優於 random forest 。 由 Table 3.5 可知， 在我們對於 Rossmamm Store 資料進行建模預測時， 雖然 random forest 的 training RMSPE 與 testing RMSPE 低於 XGBoost，但是， training RMSPE 與 testing RMSPE 差距過大。 在 XGBoost 模型中，差距爲 0.004 ， 在 random forest 模型中，差距爲 0.056。 這意味著 random forest 可能更容易過度擬合 (overfitting)， 而實際進行預測上，誤差有可能放大， 造成不良的預測。

根據以上兩點，在本篇探討的兩個問題上， XGBoost 是更好的模型， 它是藉由 `XGBoost` R package 中的 `xgb.train` 函數來實作。

一般來說，只用一個模型可能無法獲得好的結果， 接下來，我們將介紹另一種演算法，在我們的經驗上， 對於 Rossmann Store Sales 的問題，與它結合將得到更精確的預測。

3.3.3 Elastic Net

它是在一般迴歸上，加入懲罰項 (penalty term)， 該數學式子是：

$$\min_{(\beta_0,\beta)} \frac{1}{2N}\sum_{i=1}^{N}(y_i-\beta_0-x_i^T\beta)^2+\lambda[\quad(1-\alpha)\sum_{j=1}^{p}\frac{\beta_j^2}{2}+\alpha\sum_{j=1}^{p}|\beta_j|\quad]$$

β is parameters of the model. α is a adjustment term.
The p is the amount of explanatory variables.
α=1, that is a lasso regression.
α=0, that is a ridge regression.
$0 < \alpha < 1$, that is an elastic net.
The λ is a constant. It controls the penalization term.
The N is the amount of data.

它是一種迴歸所衍生的方法， 藉由 `glmnet` R package 中的 `glmnet` 函數來實作。 在 elastic net 上，如果想獲得更好的預測， 收縮估計 (shrinkage estimator) 可能是一種方法。 在我們的分析上，它達到不錯的結果，能夠降低誤差。 在統計

Table 3.6: 收縮估計前後差異比較

Rossmann Store Sales dataset, N = 45884			
shrinkage estimator	training RMSPE	testing RMSPE	parameter
NO	0.122	0.126	-
YES	0.121	0.125	k = 0.99

學中，收縮估計 (shrinkage estimator) 可能會得到較低的 MSE，所以我們將它應用在 elastic net 上。我們的作法是，在 elastic net 預測後，將預測值乘上一個常數 k，$0 < k < 1$，誤差可能因此降低了。

在 Table 3.6 中，我們進行收縮估計前後比較。收縮估計前，training RMSPE 與 testing RMSPE 分別是 0.122 與 0.126。收縮估計後，training RMSPE 與 testing RMSPE 分別是 0.121 與 0.125 。在收縮估計後，training RMSPE 與 testing error 同時降低，也就是說，對於我們利用 elastic net 進行 Rossmann Store 的預測上，可能有高估的傾向。所以收縮估計可能是降低誤差的方法之一。

而我們選擇該模型的主要原因是，它的方法不同於一般機器學習，而不同的方法，對於預測的結果可能非常不同，在集合預測上，不同的預測有可能組合成更好的結果。我們接下來將介紹集合預測。

3.3.4 集合預測 (Ensemble Prediction)

整合兩個以上的模型進行預測，可能會得到更準確的結果。而在多模型預測中，過於相似的模型，並無法提升準確率，所以我們利用兩種非常不同的演算法，elastic net 與 XGBoost。

在 Table 3.7 中，我們分別利用兩個模型進行預測。首先，glmnet 的 training RMSPE 與 testing RMSPE 分別是，0.122 與 0.126，都低於 XGBoost，但是當我們對於兩個 model 分別進行預測，再把預測值取平均後，得到的 training RMSPE 與 testing RMSPE 同時降低了。也就是說，我們對於 Rossmann Store 的分析預測上，集合預測取得不錯的結果，training RMSPE 與 testing RMSPE 彼此之間的距離也足

Table 3.7: XGBoost and glmnet 集合預測。

Rossmann Store Sales dataset, N = 45884				
System	time	training RMSPE	testing RMSPE	parameter
`glmnet`	1.957 s	0.122	0.126	family = c(”poisson”), alpha = 0.005, nlambda = 5
`XGBoost`	0.667 s	0.116	0.120	objective=”reg:linear”, booster = ”gbtree”, nrounds=50, eta=0.1, max_depth=5, colsample_bytree=0.45, subsample=1, num_parallel_tree=1
glmnet and XGBoost ensemble model	2.624 s	0.113	0.117	(pred_xgb + pred_glmnet)/2

夠靠近，並不會有 overfitting 的問題產生。因此，基於我們的經驗，利用集合預測 (Ensemble Prediction) 建立模型，有可能降低預測誤差。

3.3.5 得分評估(Score Evaluation)

由於每個問題都不相同，因此損失函數 (loss function) 也不同。大多數的損失函數，都是由統計與數學相關的專家設計，將問題的評斷標準，轉換成損失函數。

Kaggle 對於大多數問題提供損失函數，以下是 Grupo Bimbo Inventory Demand 與 Rossmann Store Sales 的損失函數：

RMSLE - Grupo Bimbo Inventory Demand
它是由 MSE 變形而來，對於預測值與實際值取自然對數 (natural log)。

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(log(p_i+1)-log(a_i+1))^2}$$

n : total number of observations in the (public/private) data set,

p_i : prediction of demand, and

a_i : actual demand for i.

RMSLE 對於低估的預測，相較於高估，更重視前者。一個合理的解釋是，它不希望降低市占率，即使庫存過多造成損失，因爲兩者的嚴重性不同。另外該評分準則在 Kaggle 上，許多問題都使用它來計算得分。

RMSPE - Rossmann Store Sales

它所代表的意思是，每個不同的預測值，誤差權重相同，
並不會因爲極端值而產生較大的誤差得分。

$$\text{RMSPE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{y_i}\right)^2}$$

Where y_i denotes the sales of a single store on a single day and $\hat{y}_i$ denotes the corresponding prediction. Any day and store with 0 sales is ignored in scoring.

而在我們的分析中，是利用 XGBoost 內建的 evaluation — RMSE 去逼近 Kaggle 的 evaluation。在 Grupo Bimbo Inventory Demand 與 Rossmann Store Sales 這兩個問題中，我們同樣對目標變數取 natural log，在 RMSLE 的 evaluation 上，是相同的標準。而對於 RMSPE 的 evaluation 上，該逼近方法可能不準，但我們一樣得到不錯的預測準確率。

Chapter 4

結果

4.1 Fitted model

4.1.1 Grupo Bimbo Inventory Demand

首先，我們利用 Data.table 套件中的 fread 函數輸入資料， training data 的時間長度爲 week 3 ～ 9。 第二，進行資料切割。 由於 testing data 是關於 未來 week 10、 week 11 的相關資料， 目標是預測未來兩個禮拜的庫存需求。 我們先簡化問題， 假設 testing data 中，都是對於 week 10 的預測， 將 testing data 中的 week 11 改成 week 10。 第三， 將 week 9 的庫存需求， 假想爲 fitting 的目標，假的 testing data ， 藉由 week 3 ～ 8 的資料進行特徵工程， 生成特徵變數 x ， 搭配 week 9 的庫存需求 y ，建立模型。

在特徵變數挑選上，使用 forward selection 與該問題的誤差計算方式—RMSLE， 計算 error 藉此找到最佳變數，進而最小化 testing RMSLE ， 最後藉由 XGBoost 建立模型。 在模型建立後， 將時間上往後平移成， 使用 week 4 ～ 9 的資料進行特徵工程， 重新製造新的特徵變數。 因爲我們 預測未來一個禮拜， 是利用最近的六個禮拜進行特徵工程， 所以在預測眞實 testing data 上， 時間必須進行平移。

我們也嘗試利用 week 3 ～ 9 建立特徵變數， 好處是減少遺失值數量，由於特徵工程是使用平均的方法， 拉長時間長度，在平均上也會相對準確，結果確實變好。

在 Kaggle 上的 private leaderboard 也持續降低 error ， 達到前 10% 。

fitted model 可以藉由以下 regression 的方法表示：

```
log.due  ~  mean.due.Agencia_ID  +
            mean.due.Agencia_ID  +
             mean.due.Canal_ID   +
             mean.due.Ruta_SAK   +
            mean.due.Cliente_ID  +
                mean.due.pa      +
                mean.due.pr      +
               mean.due.pcli     +
               mean.due.pcan     +
                mean.due.pca     +
                mean.vh.age      +
                sd.due.acrcp     +
              mean.due.acrcp
```

4.1.2 Rossmann Store Sales

方法與 Grupo Bimbo 類似， 首先利用 `Data.table` 套件中的 `fread` 函數輸入資料。 training data 時間長度爲 2013-01-01 ～ 2015-07-31， 第二，進行資料切割。 由於 testing data 是關於 未來48天 (2015-08-01 ～ 2015-09-17) 的資料，目標是預測該段時間的銷售量。

我們將 training data 最近的48天，2015-06-14 ～ 2015-07-31 進行切割， 作爲 fitting 的目標，假的 testing data。 藉由另外一部分資料， 2013-01-01 ～ 2015-06-14 ，進行特徵工程， 製造特徵變數， 搭配假 testing data (2015-06-14 ～ 2015-07-31) 中的銷售量，建立模型。

在 Figure 4.1 中， 是關於 store 1 在不同年份銷售量的折線圖， 我們取前 90 天的銷售量進行繪圖。 x 軸是時間，時間單位是"天"， y 軸則是銷售量。 不同折線代表不同年份， 分別是 2013、2014 與 2015 年， 我們由此圖觀察出， 未來的銷售量與過去高度相關， 搭配 Figure 4.2 的圖， 則此跡象更加明顯。 Figure 3.2 是 store 1 在 2013 與 2014 的全年銷售量折線圖。 未來的銷售量與過去非常相近，也就是說， 我們將它看成時間序列問題， 並進行資料切割，藉由過去預測未來， 是合理的。

而在特徵變數挑選上， 同樣使用 forward selection 與該問題的誤差計算方式—RMSPE ， 計算 error 找到最佳變數。 最後同樣利用 XGBoost 建立模型， 但是結果不夠準確， 在 Kaggle 上無法前入前 10 % 排名。 所以我們另外加入 elastic net 進行集合預測， 並進行 k = 0.968 的收縮估計， 最後成功降低誤差。 而在模型建立後，同樣進行時間平移，重新製造 feature ， 卻發現結果並不好，反而使用原始的 feature，error 比較低。 目前我們並不清楚原因，先略過這個問題。 最終模型在 kaggle 上的 private leaderboard 也成功達到前 10% 排名。

fitted model 可以藉由以下 regression 的方法表示：

$$
\begin{aligned}
\texttt{log.sale} \sim\ & \texttt{mean.sale.Store} + \\
& \texttt{mean.sale.DayOfWeek} + \\
& \texttt{mean.sale.Promo} + \\
& \texttt{mean.sale.month} + \\
& \texttt{mean.sale.day} + \\
& \texttt{mean.sale.SchoolHoliday} + \\
& \vdots + \\
& \texttt{mean.sale.swpm} + \\
& \texttt{mean.sale.yesop} + \\
& \texttt{mean.sale.tompro}
\end{aligned}
$$

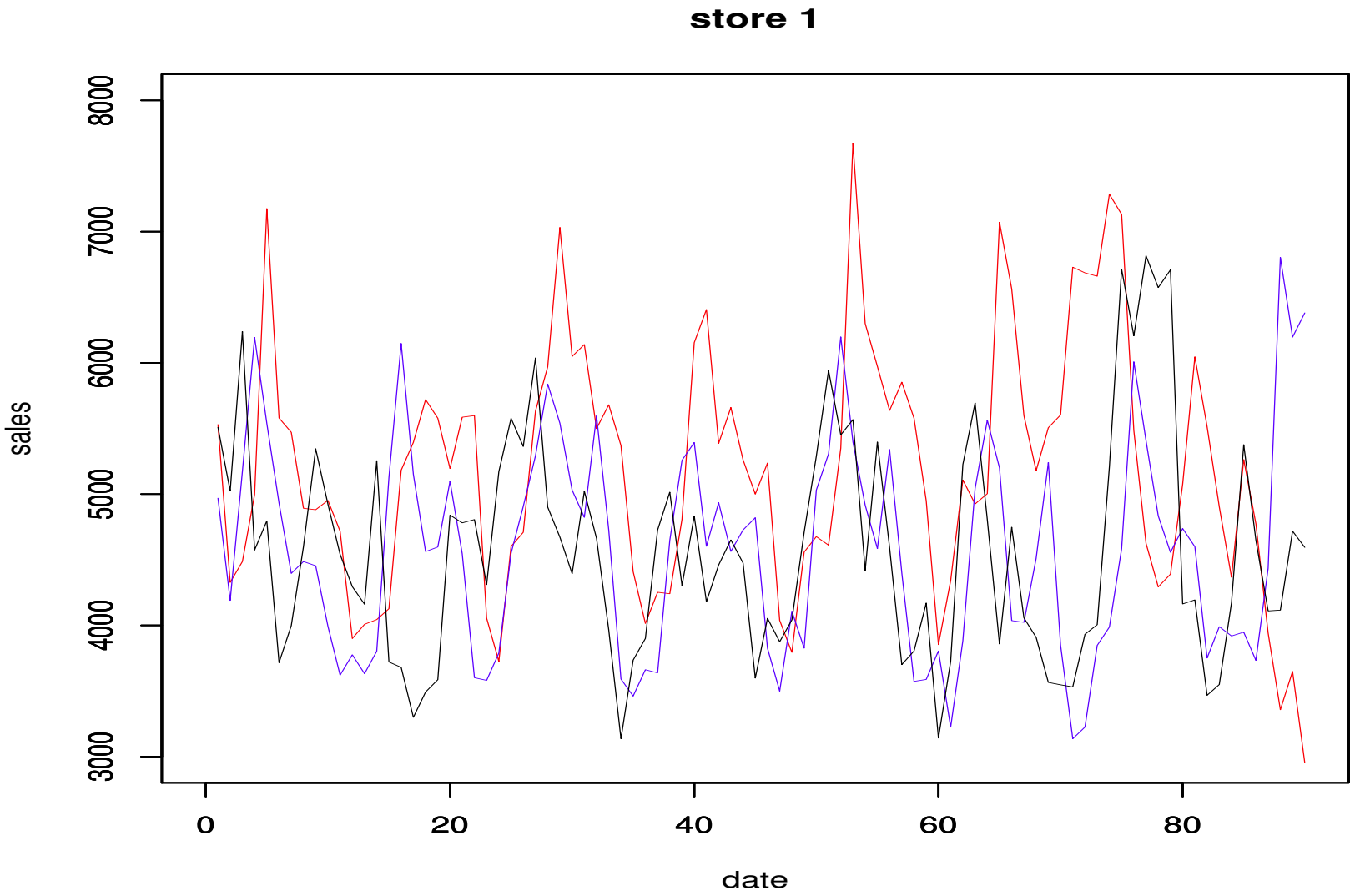

Figure 4.1: Rossmann Store Sales : 商店 1， 2013 ～ 2015 前三個月銷售量，

紅色、藍色、黑色分別代表 2013、2014、2015 的銷售量

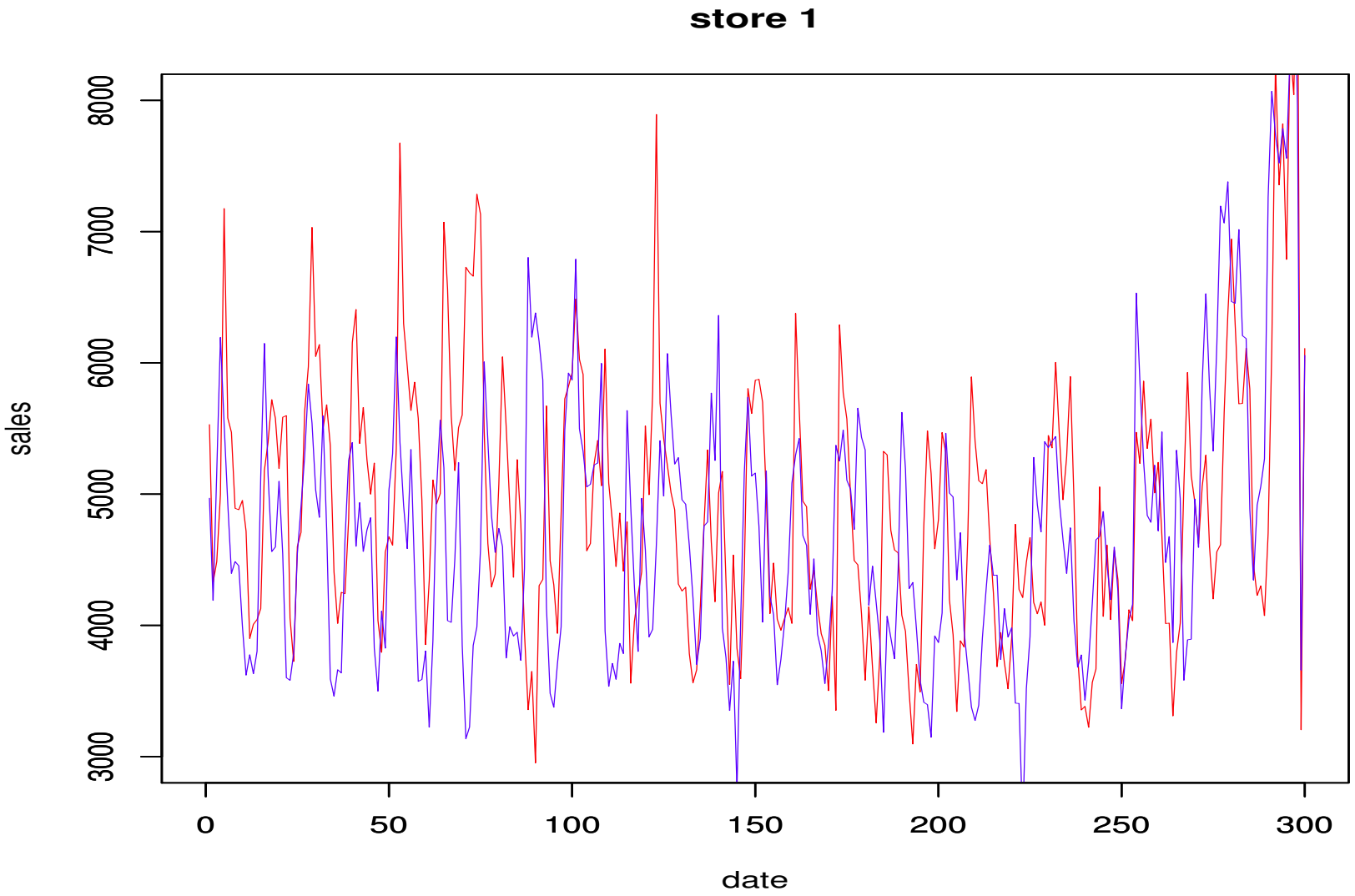

Figure 4.2: Rossmann Store Sales : 商店 1， 2013 ～ 2014 全年銷售量，

紅色、藍色分別代表 2013 與 2014 的銷售量

4.1.3 模型參數

以下是我們對於 Grupo Bimbo 與 Rossmann Store 兩個資料， fitted model 的模型參數設定， 我們藉由觀察 training RMSLE 與 testing RMSLE， 進行調整參數， 在盡可能避免 overfitting 的情況下， 最小化 testing RMSLE。 詳細參數介紹與設定方面， 可以參考相關 packages 介紹與 paper， 並非本篇重點，因此不多作介紹。

dataset	System	parameter
Grupo Bimbo	XGBoost	objective=”reg:linear”, booster = ”gbtree”, nrounds=75, eta=0.1, max_depth=8, colsample_bytree=0.5, 其他使用預設值
Rossmann Store	XGBoost	objective=”reg:linear”, booster = ”gbtree”, nro=51, eta=0.1, md=18, cb=0.45, ss=1, npt=9, 其他使用預設值
	elastic net	family = c(”poisson”) , alpha = 0.005, nlambda = 5, 其他使用預設值

4.2 系統環境

PC : CPU — I7-6700 3.40GHz

RAM — 32 GB

WINDOWS 7 64 bit

R 語言 : 運作環境 — R VERSION 3.3.2

R - PACKAGE : data.table 1.10.4

R - PACKAGE : xgboost 0.6.4

R - PACKAGE : glmnet 2.0.5

R - PACKAGE : dplyr 0.5.0

R - PACKAGE : mice 2.25

Chapter 5

結論與討論

5.1 結論

在分析這兩個資料中，我們的重點可以分爲兩點， 第一，類別變數的轉換方法。第二，XGBoost 與 elastic net 的集合預測。

第一，類別變數的轉換方法。 是類別變數對於目標變數取平均，有效降低維度，不同種類彼此之間的距離也不同。 並且對於兩個以上的類別變數合併，製造額外的特徵變數， 讓我們的變數產生各種組合，衍生出更多特徵變數。

第二，XGBoost 與 elastic net 的集合預測。 快速建立模型，是我們選擇 XGBoost 最主要的因素， 另外它在準確率上的表現也很好。 XGBoost 與 elastic net 進行集合預測，不同模型之間合併預測， 有助於提高準確率。

5.2 討論

Grupo Bimbo 該企業，每個禮拜因爲庫存問題， 造成兩千萬台幣的損失，這不單單只是金錢上的損失，也是資源上的浪費。 而庫存問題，不只發生在該企業上，世界上幾乎所有的企業，都面臨到相同的情況， 總共損失的金錢與資源，是非常驚人的。

根據我們對於 evaluation 的回推，估計出實際損失與庫存誤差，發現即使是 Kaggle 上的第一名，也無法使 Grupo Bimbo 提高獲利。實際上庫存損失確實有降低，但是銷售量也跟著減少，代表低估當周庫存，減少營業額，相互抵銷後，並無法提高獲利。這是可預期的，該企業不可能把所有資料與變數提供給參賽者，只給予一部分，可能是進行人才挑選，或是找出額外新方法。

所以我們在分析類似問題時，evaluation 只是一個參考，最重要的是，該分析是否能眞正應用，解決問題。

Appendix

5.1 Grupo Bimbo Inventory Demand

`Semana`	-	Week number (From Thursday to Wednesday)
`NombreCliente`	-	Client name
`NombreProducto`	-	Product Name
`Venta_uni_hoy`	-	Sales unit this week (integer)
`Venta_hoy`	-	Sales this week (unit: pesos)
`Dev_proxima`	-	Sales this week (unit: pesos)

5.2 Rossmann Store Sales

`Id`	-	an `Id` that represents within the test set
`StateHoliday`	-	indicates a state holiday. Normally all stores, with few exceptions, are closed on state holidays. Note that all schools are closed on public holidays and weekends. a = public holiday, b = Easter holiday, c = Christmas, 0 = None
`CompetitionOpenSince [Month/Year]`	-	gives the approximate year and month of the time the nearest competitor was opened
`Promo2Since[Year/Week]`	-	describes the year and calendar week when the store started participating in Promo2
`PromoInterval`	-	describes the consecutive intervals Promo2 is started, naming the months the promotion is started anew. E.g. ”Feb,May,Aug,Nov” means each round starts in February, May, August, November of any given year for that store

Bibliography

Chen, T. and Guetrin, C. (2016). XGBoost: A Scalable Tree Boosting System. [URL:https://arxiv.org/abs/1603.02754]

Copas, J.B. (1983). Regression, Prediction and Shrinkage.
J.R. Statist. Soc. B. 45, 311–354.
[URL:https://www.jstor.org/stable/2345402?seq=1#page_scan_tab_contents]

Grupo Bimbo Inventory Demand Dataset. (2016).
[URL:https://www.Kaggle.com/c/grupo-bimbo-inventory-demand]

Hastie, Tibshirani and Friedman (2009). The Elements of Statistical Learning: Prediction, Inference and Data Mining 2nd edition. Springer-Verlag, New York.
[URL:http://statweb.stanford.edu/∼tibs/ElemStatLearn/]

Lin, S. (2017). Code for Inventory Demand Dataset.
[URL:https://github.com/f496328mm/Kaggle_Grupo_Bimbo_Inventory_Demand]

Lin, S. (2017). Code for Store Sales Dataset.
[URL:https://github.com/f496328mm/Kaggle_Rossmann_Store_Sales]

Paulo P. Grupo Bimbo Inventory Demand. (2016).
[URL:https://www.kaggle.com/paulorzp/log-mean-plus-lb-0-47000]

Rossmann Store Sales Dataset. (2015).
[URL:https://www.Kaggle.com/c/rossmann-store-sales]