

商學院 大學部專題報告競賽

報告題目：

機器學習應用在文本挖掘以及情感分析 之研究-以某品牌手機為例

報名類別：資訊管理組

指導老師：魏嘉宏 老師

參賽人員：410779057 謝東睿 組長 程式編撰、資料處理

410779015 李家頤 資料收集

410779037 方泓元 資料收集

410779038 林郁儒 文書編輯、資料處理

410779093 劉兆宸 資料收集

410783027 郭庭仔 文書編輯、資料收集

410683039 王侑軒 文書編輯、資料收集

目錄

摘要	3
壹、 相關文獻	3
一、 電商平台之智慧型手機產業	3
二、 網路爬蟲	4
三、 文字探勘	5
(一) TF-IDF 演算法	5
1. TF(term frequency)詞頻	5
2. IDF(inverse document frequency)逆向文件頻率	5
3. TF-IDF	5
(二) 情感分析	6
貳、 研究方法	7
一、 研究架構流程	7
二、 資料收集方法	7
(一) 八爪魚介紹	8
(二) 採集原理	8
(三) 採集方法	8
三、 文字預前處理(Text Pre-Processing).....	10
(一) 使用者資料輪廓	10
(二) TF-IDF 的實現	11
1. 文本清除與彙整	15
2. 關鍵字提取	16
3. TF-IDF 統計結果	16
(三) 詞雲圖	19
四、 情感分析	21
(一) 情感詞庫的蒐集與建立	21
1. 原程式碼之詞庫	21
(二) 程式與分析	24
1. 分析過程	24

2.	原理解釋	25
(三)	分析結果	35
1.	單純貝式分類器之機器學習模型預測	35
2.	情感分析之分數界定	336
3.	評論時間序列可視化	336
參、	結果	40
一、	評估結果	40
(一)	從資料推判評價習慣	40
(二)	不同月份的消費量	40
(三)	蘋果手機在電商平台之行銷價值特徵	43
1.	產品外觀	43
2.	電池容量的擴充以及效能提升	43
3.	物流速度以及運送時之防護	43
4.	定價促銷方案的調整	44
肆、	結論	44
伍、	參考文獻	44

摘要

本研究對電子商務平台網站京東商城進行消費者評論文本統計分析以及機器學習下之情感分析，盡可能利用敘述性統計資料檢視消費者之習性以及將數據進行分類以便做預測。透過八爪魚程式從京東網站攝取資料，並以 Python 語言之 Jieba、SnowNLP、sklearn、ggplot 等套件包，進行關鍵字獲取、情感強度分析、訓練、測試之建模以及學習、數據視覺化之呈現等等，得到得以預測的模型，以及消費者對於蘋果手機的偏好與業者如何改善之建議。

關鍵字：情感分析、文字探勘、機器學習、電商、iPhone。

壹、 相關文獻

一、 電商平台之智慧型手機產業

根據經濟部統計處調查（圖 1），近年來，零售產業在電子購物及郵購的營業額有逐年成長的趨勢，透露出民眾在非店面零售業的消費金額持續增加，也造就更多店家投入新的消費市場進行競爭，目前的電商平台大致可分成通路平台、開店平台與自架平台三種類型，常見的通路平台有 Amazon、淘寶、蝦皮、京東等電商平台；知名的開店平台如 91APP、Cyberbiz；而自架平台則以 Wix、Weebly、WordPress 較為著名。

在電商平台上常見的電子商務型態以 B2C（企業對消費者）、C2C（消費者對消費者）、B2B2C（供應商對平台商對消費者）為主，也因為一般日常生活用品都透過網路輕鬆下訂，所以民眾愈來愈習慣在電商平台上購物的消費模式，而電商平台上的評分、評論機制，也是助長民眾願意在電商平台消費的主因之一，因為透過眾多顧客提供的滿意度評比，店家能夠累積相當程度的信譽，這將解決

民眾對於網路照片的不信任感。

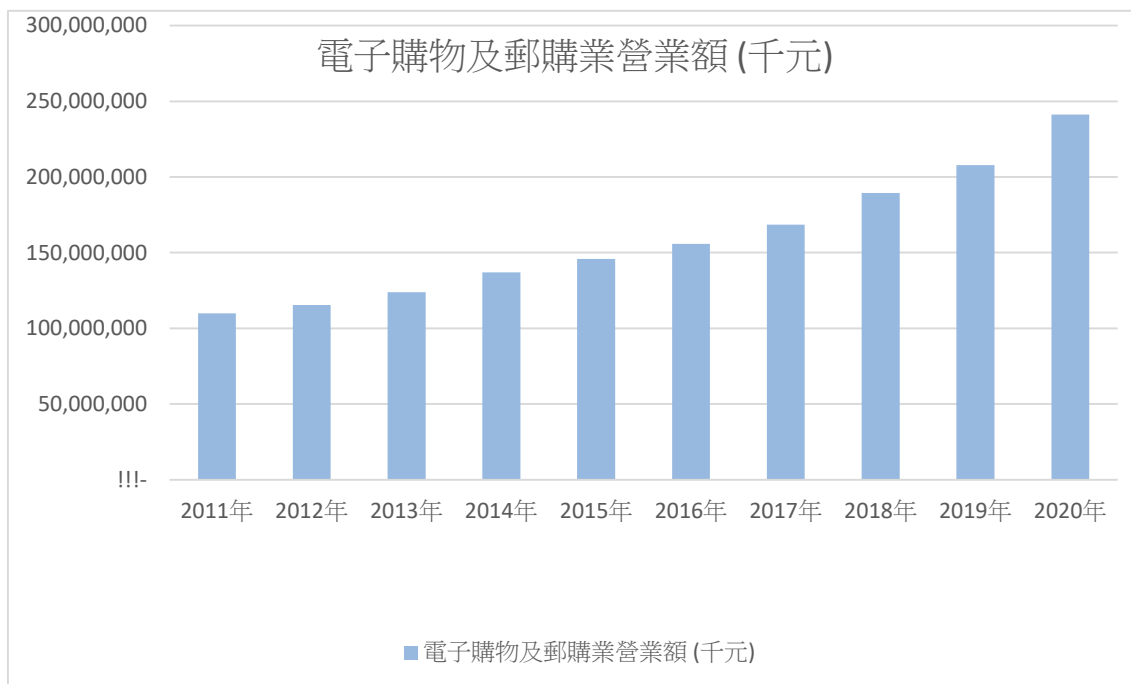


圖1 電子購物及郵購業營業額

資料來源：經濟部統計處

排除民眾對商品品質的質疑，民眾也就更願意嘗試購買高價位的產品，畢竟大多數的產品在電商平台上的販售價格，普遍比傳統店面優惠許多，又民眾還可以免去到傳統店面的交通時間與排隊時間，因此更多高單價的 3C 產品也成為電商平台提供販售的商品之一，其中各家品牌的智慧型手機也紛紛投入電商平台搶占市場。

二、網路爬蟲

網路的基本架構是建立在 HTTP 協定上，也就是從使用者端(User)發送請求(Request)封包給伺服器端(Server)。而傳送後，Server 端則會回覆相對應的 Response 封包。在日常生活中，通常都是由瀏覽器來作為 User 端發送「Request」封包。網路爬蟲則是透過程式來代替瀏覽器發送「Request」給 Server 端，因此可以藉由程式不同的設計，進而搜集客製化的資料，並把蒐集資料的任務交給程式來代理。進行「網路爬蟲」時，需先制定好規則並加以設計程式，即可透過網路爬蟲自動依照規則收集和擷取所需要的資料。本研究研究則是透過爬取「京東商城」iPhone12 整新機的評論進行詞頻重要度統計以及文本的情感分析。

三、 文字探勘

(一) TF-IDF演算法

TF-IDF (Term Frequency-Inverse Document Frequency) 是一種常用於資料處理預資料探勘的加權技術。該技術採用一種統計方法，根據字詞在文件中出現的次數與整個文本中出現的文件頻率來計算一個字詞在整體文本中的重要程度。簡單來說，TF-IDF 可以用來評估「詞」對於「文件」的重要程度。它的優點是能過濾掉一些常見卻無關緊要的詞語，同時保留影響整體文本的重要字詞。該演算法共可以被分為兩部分：詞頻(term frequency, TF)、逆向文件頻率(inverse document frequency, IDF)。

1. TF(term frequency)詞頻

詞頻指的是某一個給定的詞語在該文件中出現的頻率，第 t 個詞在第 d 篇文件的頻率記做 tft,d 。舉例來說，如果文件1共用100個字，而第1個字在文件1出現的次數共有7次，因此 $tft1,1=7/100$ 。藉此，我們可讓各文件內的各個詞彙標準化，使文件與文件間具備可比較性。

2. IDF(inverse document frequency)逆向文件頻率

逆向文字頻率才是用來處理常用字的問題。假設詞彙 t 總共在 dt 篇文章中出現過，則詞彙 t 的IDF定義成 $idft=\log(Ddt)$ 。舉例來說，文字1總共出現在13篇不同的文件則 $idft1=\log(D13)$ 。因此，如果詞彙 t 在多篇文章中皆出現過，代表 dt 很大，此時 $idft$ 就會比較小，也意味著這些「詞」可能為常用詞，如「非常」、「真的」。

3. TF-IDF

一個字對於一篇文件重要性的分數(score)就可以透過TF與IDF兩個指標計算出來。我們將第 t 個詞彙對於第 d 篇文中的TF-IDF權重定義為 $wt,d=tft,d \times idft$ 。如此一來，當辭彙 t 很常出現在文件 d 時，其 tft,d 就會比較大，且辭彙 t 也較少出現在其他篇文中，則 $idft$ 也會較大，使 wt,d 也較大，意味著辭彙 t 對於文件 d 很重要。透過上述方法所算出之TF-IDF矩陣，如（表1）所示。

表1 TF-IDF 未合併之矩陣

詞彙	文件 1	文件 2	...	文件 d
文字 1	0.2	0.12	...	0.01
文字 2	0.06	0.25	...	0.51
文字 3	0.00	0.05	...	0.32
...
文字 t	0.03	0.43	...	0.29

資料來源: 本研究繪製

此外，TF-IDF也常被用作資料檢索。例如說，給予一串指令「文字1+文字7+不在現有辭彙裡面的文字」，則這串指令跟第d篇文件的關聯分數就會為 $tf_{1,d}+tf_{7,d}$ 。

(二) 情感分析

情感分析 (Sentiment Analysis) 是自然語言處理(Natural Language Processing, NLP)中常見的應用方式。於電腦上處理自然語言之相關研究，需對該內文進行特殊處理，使兩者語言得以互通，建立人類與電腦間的溝通管道。而其中之情感分析之技術，可用於商業管理系統來檢視消費者對產品之傾向與建議，讓從業人員得以透過電腦文本的資料，深入顧客評價，進而達到產品優化或萃取行銷價值，逐步堆疊出商業資產。

根據我國國家教育研究院之定義的「情感分析」，意指藉由電腦內部的文字探勘 (Text Mining) 技術，得以將文件中之文字資料，對其進行情感或意見資訊的偵測、萃取與分析。換言之，情感分析能透過一篇文章、一段詞句或一些字詞，對其進行分類，進一步做判斷該結構內容之情緒為正面情緒或負面情緒；抑可瞭解該內容屬於為積極的、消極的、中立性的。此外，若加以細分為更繁多之種類，常見人之情緒可分為七大類別，依序為好、樂、哀、怒、懼、惡、驚。

而本研究於此研究中，採用常見之中文之文本情感分析方法 SnowNLP，探討 iPhone 手機消費者樣本評價之正面與負面情緒。該套件屬於 Python 中處理中文文本內容的類庫，其主要功能包括分詞、詞性標註、情感分析、漢字轉拼音、繁體轉簡體、關鍵詞提取以及文本摘要等等。於三項功能中皆可依據分析者欲進行之文本分析內容修正與新增自訂義文本，本研究將於後續分析步驟中予以解釋。

貳、 研究方法

一、 研究架構流程

本文以京東商城為資料來源建立手機評論之文本庫，先利用 Excel 進行資料庫的統計分析。再者，利用 Jieba 分詞套件找尋文本中之高頻率關鍵詞進行文字預處理。為更精準計算情感強度，透過監督式學習自定義斷詞文本結合 SnowNLP 為資料標註其特徵，並透過貝式分類器計算情感強度。此外，為觀察消費者重視之特徵，引用時間序列，將情感分析強度變化視覺化呈現，觀察其變化程度以及評論情感之穩定度。詳細研究架構流程，如（圖 2）所示。

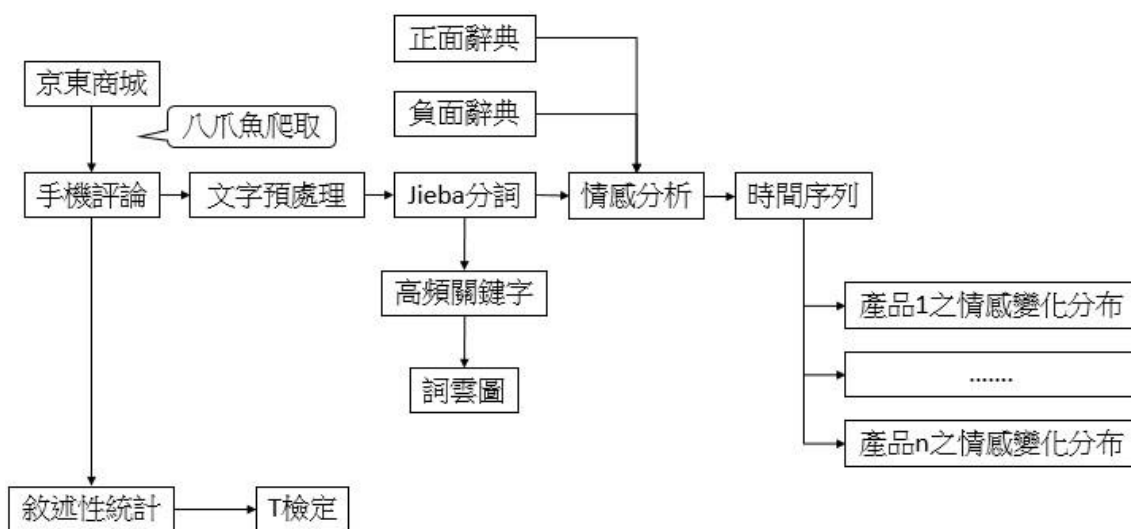


圖2 研究架構流程

資料來源：本研究繪製

二、 資料收集方法

本研究資料來源之電子商務商城網站京東商城屬於須動態爬蟲技術之網站。然其，其網頁代碼屬性標籤為網站所自訂難以定位，故本研究引述第三方軟體八爪魚程式處理京東電商平台之動態爬蟲作業。

(一) 八爪魚介紹

八爪魚是一款通用的網頁數據採集器，能夠採集 98% 的網頁。可簡單快速地将網頁數據轉化為結構化數據，存儲為 Excel 或數據庫等多種形式，並且提供基於雲計算的大數據雲採集解決方案。八爪魚作為一款通用的網頁數據採集器，並不針對於某一網站某一行業的數據進行採集，而是網頁上所能看到或網頁源碼中有的文本信息，幾乎都能採集。

(二) 採集原理

八爪魚採集器的核心原理是：基於 Firefox 內核瀏覽器，通過模擬人瀏覽網頁的行為（如打開網頁，點擊網頁中的某個按鈕等操作），對網頁內容進行全自動提取。內置火狐內核瀏覽器，模擬人瀏覽網頁、複製數據的行為，通過設計工作流程，自動化採集數據。

(三) 採集方法

在八爪魚客戶端中，採集和導出數據主要經過以下 3 個步驟：配置任務、本地採集、導出數據。

任務配置：為本次研究的流程圖，將要採集的網頁打開後，點擊評論內容，進到留言區後開始採集所需的資料，資料包含：使用者姓名、產品名稱、評論星星數、評論主題、評論日期、手機顏色、手機容量、價錢、評論內文、推文按讚數。八爪魚方便的原因是他有循環、判斷條件的用法。如（圖 3）所示。

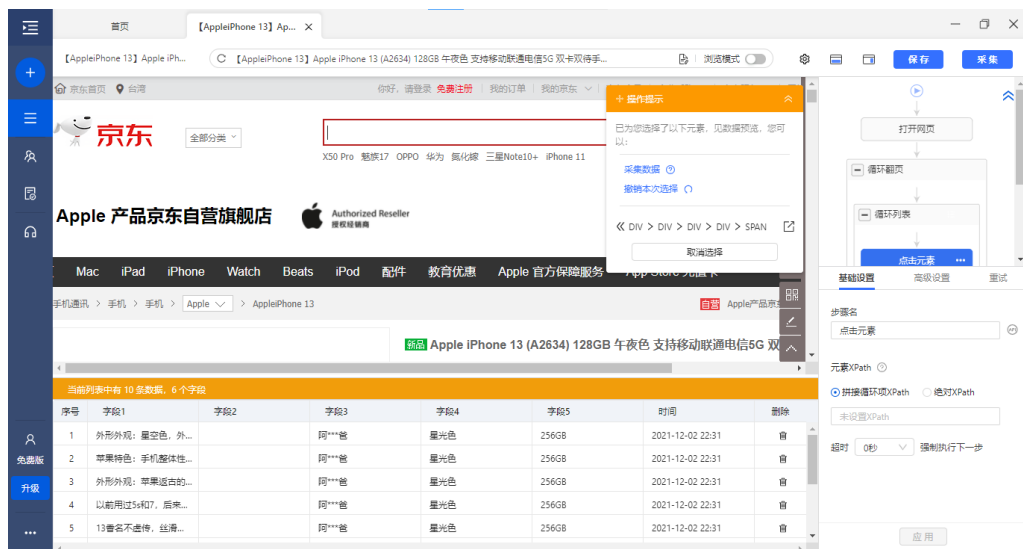


圖3 八爪魚流程圖

資料來源：八爪魚截圖

將所要採集的網址置入循環內，第一項產品採集完後自動會跳入循環的第二項產品。使用判斷條件用法，採集過程中發現第一位留言者評論網頁中會出現下一篇留言者，透過這項發現讓我們運用判斷條件，判斷只要出現「下一頁」字樣，就點擊下一則留言者，自動化本研究的採集速度。

本地採集：八爪魚採集資料有本地採集、雲採集模式，差別在於雲採集能雲端備份採集後的資料，若需修正流程圖就不需要再重複抓取資料。由於本研究未購買雲採集功能，所以採用本地採集，本地採集過程，採集結果如（圖 4）所示。

#	用戶	評論	評論1	信息	信息2	評論4	
1	...***生	一直拖到一两个星期之后才...	2021.11.16	石墨色	256GB	5	详情
2	u***1	就那样像素真的一般般	2021.11.18	远峰蓝色	256GB	2	详情
3	董***迷	快递盒被拆开，不知道东西...	2021.11.20	石墨色	256GB	3	详情
4	****8	摄像头，苹果品控真差，还...	2021.11.22	金色	128GB	2	详情
5	w***1	快递不行，磕成这样屏幕...	2021.11.19	石墨色	512GB	4	详情
6	C***_	这个机子有问题，显示屏抖...	2021.11.20	金色	512GB	2	详情
7	****C	手机包装盒没有塑封吗？	2021.11.12	石墨色	256GB	4	详情
8	眼***梦	保价都不能用？买完就降价	2021.11.05	石墨色	128GB	12	详情
9	****云	今天才看到降价通知，真服...	2021.11.23	远峰蓝色	256GB	2	详情
10	t***m	给不了0量 售后态度不好	2021.11.21	远峰蓝色	128GB	2	详情
11	R***e	手机盒居然是打开的，都是...	2021.11.15	远峰蓝色	256GB	11	详情
12	沙***山	我只能说双十一熬夜抢到的...	2021.11.22	石墨色	256GB	1	详情
13	l***女	双11一过，差价几十块，剧...	2021.11.24	远峰蓝色	128GB	1	详情

圖4 八爪魚爬取資料

資料來源：八爪魚截圖

導出數據:採集完所需資料後可將數據導出成 Excel 格式，導出數據如（表 2）所示。

表2 Excel 格式之數據範例

用戶	評論星數	評論日期	顏色	容量	價格	評論內文	討論度
z***g	5	2021/10/28	星光色	512	8399	可以可以 顏色很正 速度超 快。 外形外 觀：最酷 的世界 屏幕音 效：...	1
胖***乐	5	2021/9/25	粉色	256	6799	超爱 超好 用 粉红 色?yyds 对 我来说 13 刚刚好...	12
刘***焱.	5	2021/9/25	粉色	128	5999	特别特别 特别好， 没有任何 破损的地 方，外观 也好看...	638

資料來源：本研究編撰

本次研究中採集了京東商城的評論，類別分為 i13 mini 系列 1460 則評論，regular 系列 3000 則評論，pro 系列 1780 則評論，pro max 系列 1860 則評論，總共 8104 則評論，在網頁中採集的內容資料包含:用戶、評論星數、評論日期、顏色、容量、價格、評論內文、討論度。

三、 文字預前處理(Text Pre-Processing)

(一) 使用者資料輪廓

本次研究將於此小節對所蒐集到之使用者資料進行敘述性統計描寫及量化分析。由於之後章節的關鍵詞提取及情感分析皆為針對文字領域所進行之分析，故在進入到本研究文字處理之步驟前，應設立撰寫評論之使用者的相關資料。本組於前段說明八爪魚之資料採集，搜集之資料包含使用者之評論帳號、評論星等、評論、評論日期及購買機型之相關資訊。於此小節將使用敘述性統計查看相關使用者輪廓之變數，如：評論星等、購買型號，進行次數統計及百分比分佈檢測。完成該統計步驟，將針對兩種變數進行是否有群組差異之檢測。本組所使用之檢測方式為，透過假設檢定 (Hypothesis Testing) 進行檢驗，去理解並研究此假設是否正確。使用的假設檢定工具包含「獨立樣本 T 檢定」(Independent-Sample t test) 以檢定兩個獨立樣本的平均數差異是否達到顯著水準。於前述之研究假設中，本組分別設定虛無假設 (Null Hypothesis, H_0)，及與其相反之對立假設 (Alternative Hypothesis, H_1)。

於本實驗中，若欲檢測兩組獨立樣本之平均值間是否存在顯著差異，則 T 值是獨立樣本 T 檢定的檢定統計量。

1. 敘述性統計

(1) 各型號之評論星等及樣本數

此節針對研究樣本之評論星等進行描述，其中包含 iPhone 13 mini, iPhone 13 regular, iPhone 13 pro, 以及 iPhone 13 pro max，如 (表3)。

表3 評論統計表

敘述性統計 (n=8104)		
iPhone 13 mini		
變數	次數	百分比
一顆星	96	6
兩顆星	38	2.6
三顆星	16	1.1

四顆星	54	3.7
五顆星	1255	86.6
無法辨認之遺漏值	0	0.0
總和	1459	100
iPhone 13 regular		
變數	次數	百分比
一顆星	160	5.3
兩顆星	112	3.7
三顆星	48	1.6
四顆星	32	1.1
五顆星	2648	88.3
無法辨認之遺漏值	4	0.1
總和	3000	100
iPhone 13 pro		
變數	次數	百分比
一顆星	96	5
兩顆星	50	3
三顆星	16	1

四顆星	18	1
五顆星	1600	90
無法辨認之遺漏值	0	0.0
總和	1780	100
iPhone 13 pro max		
變數	次數	百分比
一顆星	111	6
兩顆星	46	2.5
三顆星	17	1
四顆星	15	0.7
五顆星	1671	89.8
無法辨認之遺漏值	0	0.00
總和	1860	100

資料來源：本研究編製

此項目的研究樣本取於京東商城之評論樣本資料。如下表，在iPhone 13 mini部分，總共採取了1459則評論，有1255位使用者給了五顆星（佔86.6%）；有96位使用者給了一顆星（佔6%）；有38位使用者給了兩顆星（佔2.6%）；有16位使用者給了三顆星（佔1.1%）；有54位使用者給了四顆星（佔3.7%）；無法辨識之遺漏值為0（佔0%）。

在iPhone 13 regular部分，總共採取了3004則評論，有160位使用者給了一顆星（佔5.3%）；有2648位使用者給了五顆星（佔88.3%）；有112位使用者給了兩

顆星（佔3.7%）；有48位使用者給了三顆星（佔1.6%）；有32位使用者給了四顆星（佔1.1%）；無法辨識之遺漏值為4（佔0%）。

在iPhone 13 pro部分，總共採取了1780則評論，有1600位使用者給了五顆星（佔90%）；有96位使用者給了一顆星（佔5%）；有50位使用者給了兩顆星（佔3%）；有16位使用者給了三顆星（佔1%）；有18位使用者給了四顆星（佔1%）；無法辨識之遺漏值為0（佔0%）。

在iPhone 13 pro max部分，總共採取了1860則評論，有1671位使用者給了五顆星（佔89.8%）；有111位使用者給了一顆星（佔6%）；有46位使用者給了兩顆星（佔2.5%）；有17位使用者給了三顆星（佔1%）；有15位使用者給了四顆星（佔0.7%）；無法辨識之遺漏值為0（佔0%）。

2. T 檢定－購買產品之型號會影響使用者之評

本研究如(表4)發現，使用者於網站評分高低，與購入之產品型號有顯著差異。結果顯示，比起購買體積最大的iPhone 13 pro之使用者，購買體積較小的iPhone 13 mini之使用者較滿意所購入之產品，平均星數為4.623，小於iPhone 13 pro的平均星數（4.68），且結果達到統計上之極致顯著意義；同樣的，比起購買體積較大的iPhone 13 regular之使用者，購買體積較小的iPhone 13 mini之使用者較滿意所購入之產品，平均星數為4.623，大於iPhone 13 regular的平均星數（4.384），且結果具統計上之顯著意義。最後，本組也觀察到，規格的升級也會影響使用者之評分高低；可發現比起購買一般型號的iPhone 13 regular之使用者，購買較高規格的iPhone 13 pro之使用者較滿意所購入之產品，平均星數為4.68，大於iPhone 13 regular的平均星數（4.384），且結果達到統計上之極致顯著意義；同樣的，比起購買較高規格的iPhone 13 pro之使用者，購買最高規格的iPhone 13 pro max之使用者較滿意所購入之產品，平均星數為4.638，小於iPhone 13 pro的平均星數（4.68），且結果具統計上之顯著意義。

表4 購買產品之型號影響使用者之推薦星數

品項	n	平均值	標準差	T值
iPhone 13 mini	1459	4.623	1.69	4.04***
iPhone 13 pro	1780	4.68	1.81	

iPhone 13 mini	1459	4.623	1.69	0
iPhone 13 regular	3000	4.384	0.71	
iPhone 13 regular	3000	4.384	0.71	4.21 * * *
iPhone 13 pro	1780	4.68	1.78	
iPhone 13 pro	1780	4.68	1.78	0.001
iPhone 13 pro max	1860	4.638	1.83	
註：p<0.05，統計上為顯著；p<0.001，統計上為極致顯著				

資料來源：本研究編撰

(二) TF-IDF的實現

1. 文本清除與彙整

本研究的資料來源京東電商平台之評消費者評論、網站評分皆為非結構化自然語言形式，進行分析前需先轉為結構化資料，其前置作業包含：文本清除與彙整及關鍵詞提取。

本階段將以人為方式將收集之文本移除各篇之非中文字詞、表情符號，最終將處理過之文本表單重新拼接。由於Python的分詞套件Jieba詞彙量大且持續更新中，其分詞準確、處理速度快，且能支援使用者自行定義辭典，因此本研究將採用Jieba作為中文文本分詞之工具。原始文本資料，如（表5）所示。

表5 斷詞前之原始範例文本

	評論
1	13 香啊！是真的香，没用到大家说的那些不好的地方，第一次用苹果，非常的丝滑，拍照棒极了，淡淡的粉红色非常的少女，喜欢喜欢喜欢！
2	很喜欢用苹果手机拍照拍视频，因为拍出来的效果比其他品牌更真实！这是第三次买苹果手机，也是最贵的一次。从苹果 6 到苹果 11 再到现在的苹果 13，依赖上了苹果 OS 系统。关键是苹果 13 的定价比较厚道，这次大促又有不少

	的优惠，就毫不犹豫入手了，可以说性价比很高。以后使用相信也会有不错的体验，暂时还没拆开使用，等机器拆封体验后再来追评！
3	屏幕音效：很好 运行速度：快 外形外观：好看好看好看

資料來源：本研究編撰

2. 關鍵字提取

本研究的資料來源自於電商網站，為提取較具辨識力之特徵詞，本研究會就各機款加入自定義停用詞，並藉TF-IDF加權計算，為避免Jieba精確模式斷詞影響關鍵字提取，亦會加入自定義詞彙文本。

3. TF-IDF 統計結果

(1) 基於 Jieba 套件實現 TF-IDF 之計算

為結合進一步之詞雲圖、時間序列及情感分析之探討，本研究將以 Jieba 進行 TF-IDF 之計算與統計，決定關鍵詞。(表 6) 為其部分 Python 之程式碼。

表6 TF-IDF 部分之程式碼

```
#encoding=utf-8
import jieba
jieba.set_dictionary("dict_text_big.txt")
jieba.load_userdict("user_dict.txt")

with open("comments.txt","r",encoding="utf-8") as file:
    sentence = file.read()
with open("stop_words.txt","r",encoding="utf-8") as file:
    stopwords = file.read().split("\n")

breakword = jieba.cut(sentence,cut_all=False)
final_words = []
for word in breakword:
    if word not in stopwords:
        final_words.append(word)
print( " ".join(final_words))
```

```

x = str(final_words)

import jieba.analyse
tags = jieba.analyse.extract_tags(x, topK=30, withWeight=True)
for tag in tags:
    print('word:', tag[0], 'tf-idf:', tag[1])

```

資料來源：本研究編撰

(2) 各產品關鍵詞

本階段將電商平台京東商城收集的文本按照產品品項分類，共分為四大類：iPhone 13 mini, iPhone 13 regular, iPhone 13 pro, 以及iPhone 13 pro max，各產品關鍵字提取依據為上述參數的篩選。

此項目為對於iPhone 13研究抽樣於京東商城之1460條mini之評論，TF-IDF加權指數前30名之關鍵詞如（表7）。從表中可發現消費者在mini款項之購買較著重於商品在硬體上面得表現，包含功能類別的「拍照」、「音效」、「續航」等等，以及符合預期可以在mini機款看見的外在特徵如「手感」、「小巧」、「小屏」等等，顯示消費者期待mini款項不同於其他款項之特殊之處。

表7 iPhone 13 mini 前30名關鍵字

排名	關鍵字	TF-IDF	排名	關鍵字	TF-IDF
1	手机	0.1662	16	小巧	0.0674
2	屏幕	0.1258	17	外形	0.0660
3	京东	0.1223	18	效果	0.0595
4	拍照	0.1030	19	运行	0.0541
5	音效	0.0891	20	13mini	0.0527
6	手感	0.0874	21	13	0.0476
7	外观	0.0847	22	好看	0.0467
8	非常	0.0821	23	续航	0.0463
9	待机时间	0.0802	24	小屏	0.0436
10	喜欢	0.0730	25	真的	0.0432
11	Mini	0.0694	26	流畅	0.0366
12	很好	0.0688	27	单手操作	0.0359
13	苹果	0.0683	28	粉色	0.0354
14	速度	0.0680	29	充电	0.0351
15	不错	0.0677	30	客服	0.0342

資料來源：本研究編撰

此項目為對於iPhone 13研究抽樣於京東商城之3000條regular之評論，TF-IDF加權指數前30名之關鍵詞如（表8）。從表中可發現消費者在此款項之購買依然較著重於商品在功能以及外觀上的特徵。比如「拍照」、「續航」、「音效」、「運行速度」等等。

表8 iPhone 13 regular 前30名關鍵字

排名	關鍵字	TF-IDF	排名	關鍵字	TF-IDF
1	拍照	0.1665	16	好看	0.0582
2	屏幕	0.1557	17	很好	0.0543
3	音效	0.1316	18	流畅	0.0518
4	手机	0.1312	19	降价	0.0507
5	待机时间	0.1248	20	充电	0.0445
6	外观	0.1093	21	12	0.0442
7	13	0.1043	22	感觉	0.0399
8	效果	0.0898	23	手感	0.0383
9	苹果	0.0858	24	保价	0.0382
10	速度	0.0852	25	喜欢	0.0379
11	外形	0.0815	26	粉色	0.0366
12	运行	0.0809	27	没有	0.0358
13	京东	0.0788	28	双十	0.0354
14	非常	0.0605	29	11	0.0351
15	不错	0.0593	30	续航	0.0323

資料來源：本研究編撰

此項目為對於iPhone 13 pro研究抽樣於京東商城之1780則之評論，TF-IDF加權指數前30名之關鍵詞如（表9）。從表中可發現消費者在此款項之購買依然較著重於商品在功能以及外觀上的特徵。

表9 iPhone 13 pro前30名關鍵字

排名	關鍵字	TF-IDF	排名	關鍵字	TF-IDF
1	拍照	0.1390	16	13	0.0522
2	京东	0.1368	17	好看	0.0500
3	屏幕	0.1283	18	流畅	0.0497
4	手机	0.1206	19	远峰	0.0470
5	外观	0.0893	20	高刷	0.0458

6	音效	0.0876	21	发货	0.0455
7	待机时间	0.0834	22	很好	0.0446
8	苹果	0.0750	23	手感	0.0408
9	效果	0.0743	24	客服	0.0385
10	非常	0.0731	25	13pro	0.0363
11	速度	0.0691	26	收到	0.0360
12	外形	0.0634	27	续航	0.0338
13	不错	0.0569	28	快递	0.0335
14	运行	0.0561	29	没有	0.0330
15	真的	0.0545	30	问题	0.0327

資料來源：本研究編撰

此項目為對於iPhone 13 pro max研究抽樣於京東商城之1860則之評論，TF-IDF加權指數前30名之關鍵詞如（表10）。從表中可發現消費者在此款項之購買依然較著重於商品在功能以及外觀上的特徵。

表10 iPhone 13 pro max前30名關鍵字

排名	關鍵字	TF-IDF	排名	關鍵字	TF-IDF
1	拍照	0.1501	16	不错	0.0541
2	京东	0.1457	17	续航	0.0528
3	屏幕	0.1456	18	流畅	0.0519
4	手机	0.1385	19	好看	0.0510
5	音效	0.1149	20	13	0.0506
6	外观	0.0947	21	真的	0.0458
7	待机时间	0.0911	22	客服	0.0451
8	效果	0.0864	23	高刷	0.0414
9	速度	0.0781	24	远峰	0.0400
10	非常	0.0744	25	快递	0.0347
11	运行	0.0700	26	手感	0.0343
12	苹果	0.0678	27	没有	0.0325
13	外形	0.0668	28	收到	0.0323
14	很好	0.0631	29	喜欢	0.0303
15	发货	0.0585	30	电池	0.0295

資料來源：本研究編撰

（三） 詞雲圖

詞雲圖使一種展現高頻率關鍵詞的可視覺化表達的一種方法，通過文字、色彩、圖型的搭配，產生有衝擊力的視覺效果，並且傳達有價值的信息。本研究將使用 Python 內之 wordcloud 套件包，採用前階段之 TF-IDF 之結果，將關鍵詞以視覺化方式呈現，文字空間佔比較大者代表較為重要。程式碼如（表 11）所示。以 pro max 系列為例，可發現消費者對於注重的特徵包括「屏幕」、「拍照」、「外觀」、「速度」等等。如（圖 5）所示。

表11 pro max詞雲圖之程式碼

```
from PIL import Image
import matplotlib.pyplot as plt
from wordcloud import WordCloud
import jieba
import numpy as np
from collections import Counter
with open("comments.txt","r",encoding="utf-8") as file:
    text = file.read()
jieba.set_dictionary("dict_text_big.txt")
with open("stop_words.txt","r",encoding="utf-8") as f:
    stops = f.read().split("\n")
terms = []
for t in jieba.cut(text,cut_all=False):
    if t not in stops:
        terms.append(t)
diction = Counter(terms)
font = "C:\\Windows\\Fonts\\simsum.ttc"
wordcloud = WordCloud(background_color = "white",font_path=font)
wordcloud.generate_from_frequencies(diction)
plt.figure(figsize=(6,6))
plt.imshow(wordcloud)
plt.axis("off")
plt.show
```

資料來源：本研究編撰



圖5 pro max詞雲圖

資料來源：本研究繪製

四、情感分析

透過八爪魚採集之 8100 筆評論完成 TF-IDF 之關鍵字萃取，本本研究採用人工撰寫評論之方式，下小節將開始說明詞庫文本

(一) 情感詞庫的蒐集與建立

1. 原程式碼之詞庫

本研究採用之Python套件為中文自然語意處理之SnowNLP，常見功能如前文所提包括分詞、詞性標註、情感分析、漢字轉拼音、繁體轉簡體、關鍵詞提取以及文本摘要等等。由於該套件中的文本是由中國工程師針對網路電商平台之使用者購物後之評論所作的分析模型，是故在判斷簡體中文上理應會來的相當精確，在情感分析的正、負面表態也會因為文本本身是適用於電商平台環境而給予較正確的分析回應。（表12）則是以mini款項直接帶入程式碼之結果，可見其分析來得相當精確，為了方便區分評論為正負面，將情感分數大於0.6設為1，其餘為0。

表12 使用原始碼SnowNLP測試評論之情感分析

```
for i in range(data[len]):
    text = df.comment.iloc[i]
    def get_sentiments(text):
        s = SnowNLP(text)
        return s.sentiments
```

<pre>df["sentiments"] = df.comment.apply(get_sentiments) print(df.head(5))</pre>			
<pre>def snow_result(comment): s = SnowNLP(comment) if s.sentiments >= 0.6: return 1 else: return 0 df["snlp_sentiment_result"] = df.comment.apply(snow_result)</pre>			
user	comment	sentiments	snlp_sentiment_result
1	小手机一直是我的最爱！ 苹果出的几款小型手机都 买完了！国产手机市面好 像已经看不到小屏手机 了。	0.95116472	1
2	我买 iPhone13mini 版是 冲着 5.4 寸小屏来的，买 回后发现很多问题：1、 第二天早上出现无服务持 续 2 小时，当时当地其他 品牌手机是满格信号。 2、使用~语音聊天、当调 至最大音量时，声音严重 失真（声破）。3、~语音 聊天的声音排序混乱，不 按实际发送先后排序，使 人听语音时产生误解...	3.07E-09	0
3	手机小小的好喜欢好喜欢 好喜欢白色好好看好好看 好好看爱死了爱死了	0.99877408	1
4	手机屏幕缝隙非常大，都 能塞进两张纸，要么是组 装不良，要么是手机被拆 解过，联系客服，神回 复，只要手机能使用，不	8.81749E-05	0

	是功能故障是不退的		
5	一如既往的精致，小屏直边手感超好	0.96815934	1

資料來源: 本研究編撰

另外，這邊對於評論評分超過3顆星以及低於3顆星(包含)分別設定為2個標籤(1和0)，定義函式sentiment_star表示利用SnowNLP套件分析文本得到的分數(snlp_sentiment_result欄位)和文本本身真實的分數(sentiment_star欄位)的精度，可以看到精度來到0.9184372858122002。(表13)為文本原始評分以及SnowNLP分析後之比較。(表14)為精度的計算。

表13 原始文本評分以及SnowNLP分析後之比較

user	comment	star	sentiment_star	snlp_sentiment_result
1	小手机一直是我的最爱！苹果出的几款小型手机都买完了！...	5	1	1
2	我买 iPhone13mini 版是冲着 5.4 寸小屏来的，买回后发现很多问题：1、第二天早上出现无服务持续 2 小时，当时当地其他品牌手机是满格信号...	1	0	0
3	手机小小的好喜欢好喜欢好喜欢白色好好看好好看好好看爱死了爱死了	5	1	1
4	手机屏幕缝隙	2	0	0

	非常大，都能塞进两张纸，要么是组装不良，要么是手机被拆解过，联系客户...			
5	一如既往的精致，小屏直边手感超好	5	1	1

資料來源：本研究編撰

表14 利用SnowNLP計算精度

<pre>counts =0 for i in range(len(data)): if data.iloc[i,2]== data.iloc[i,3]: counts+=1 print(counts/len(data))</pre>
result 0.9184372858122002

資料來源：本研究編撰

雖然在開頭已經可以針對研究所收集之評論進行正確的判斷，然而本研究認為仍然可以針對原始碼，透過自定義文本重新訓練分析模組。是故若能修正程式原始碼所運行的文本，便可以根據使用者所欲分析之領域重新設定機器學習的判斷標準。因此，為提高模型判斷正確率，將在下小節說明正、負面文本之詞庫的建立，以及結合模組分詞後，進行貝氏學習的訓練以及預測。

(二) 程式與分析

1. 分析過程

透過閱讀SnowNLP原始程式碼之後的解析，大致上可以了解SnowNLP的情感判斷過程為：

- (1) 讀取已經分好類的文本neg.txt和pos.txt
- (2) 對所有文本進行分詞、去停用詞
- (3) 計算每個詞出現的頻數
- (4) 通過貝氏定理計算正面負面先驗概率 $P(\text{pos})$ 和 $P(\text{neg})$
- (5) 對要進行判斷的文本分詞

(6) 計算每個詞的後驗概率 $p(\text{詞}|\text{neg})$ 和 $p(\text{詞}|\text{pos})$

(7) 選擇計算出的概率較大的類別（正或負）

2. 原理解釋

整個原理即計算每個詞分別在neg.txt和pos.txt出現的詞頻，由此算出待判斷的語句中的詞出現在neg.txt, pos.txt中的概率，選其中概率大的一個類別，而文後將接續說明本研究如何重新設定判斷標準並且訓練模型。

(1) 本研究自訂之正、負面意見詞庫

根據檢視程式原始碼，可發現在該套件中情感分析的處理上，是可以透過train函數修改資料夾中「pos」與「neg」的txt檔內容，重新定義正、面情緒判斷，如（表15）程式碼所示。

表15 SnowNLP中Sentiment函數之原始碼

```
class Sentiment(object):
    def __init__(self):
        self.classifier = Bayes()
    def save(self, fname, iszip=True):
        self.classifier.save(fname, iszip)
    def load(self, fname=data_path, iszip=True):
        self.classifier.load(fname, iszip)
    def handle(self, doc):
        words = seg.seg(doc)
        words = normal.filter_stop(words)
        return words
    def train(self, neg_docs, pos_docs):
        data = []
        for sent in neg_docs:
            data.append([self.handle(sent), 'neg'])
        for sent in pos_docs:
            data.append([self.handle(sent), 'pos'])
        self.classifier.train(data)
    def classify(self, sent):
        ret, prob = self.classifier.classify(self.handle(sent))
        if ret == 'pos':
            return prob
        return 1-prob
```

```
classifier = Sentiment()
classifier.load()
```

資料來源: 本研究編撰

可以看見，在程式碼中定義正負面辭典的分別是”neg”和”pos”檔，是故將人工評論寫入兩個txt檔後，再於程式開發中寫入訓練函數的程式碼，便可以完成自定義正、負面情感分析的判讀模型，訓練新文本之程式碼與詞庫檔如（表16），以mini款項為例。

表16 SnowNLP訓練自定義文本

正面情緒文本
1. "外形外观：和 12mini 基本一致，厚度有所增加
2. 屏幕音效：屏幕很棒，视野更广
3. 运行速度：A15 功耗比确实 nb，低电量模式 yyds
4. "外形外观：外形外观比较亮丽，抢眼，做工一流，材质相当好，一拿上手就有爱不释手，不想放下的感觉。
5. 真不错，过去用过苹果，这次当备用机使用，没想到十分满意，使用感觉很流畅，系统也很干净，大屏、小屏手机各有各的优势，色彩、清晰度也相当满意！小巧又方便携带！
...
1794. 手机外观还 OK 内机还得用用看在做评价，（看到赠送一年 AC 才抢的后来发现就只是一年并不是两年质保，有点可惜了）
負面情緒文本
1. 垃圾的很，正常使用久点就发烫的不行。刷刷~这些就更汤的不行
2. 有这么明显肉眼可见的脏污
3. 打开第一眼就看到上面两个脏污点，体验极差。
4. 13mini 微距拍摄不清晰，屏幕看一会就头晕，眼睛不舒服；我的 12mini 就不会有这种现象！
5. 系统差，充电烫烫烫
...
500. 不是很满意，充电发烫打电话发烫，当备用机用的都达不到，屏幕发黄

資料來源: 本研究編撰

(2) 結合Jieba分詞與SnowNLP

使用下來，SnowNLP在分詞時無法識別否定短語，如下（表17）所示，「样子很好看，很潮不知道耐不耐摔，不过还是很满意。」同樣一句資料，Jieba很清

楚的分成「不知道」，然而SnowNLP卻分成「不」+「知道」兩個詞彙。情感分析時，會因為計算出「不」在neg.txt中出現概率較大，整體偏負面情緒。

表17 Jieba分詞與SnowNLP之seg函式功能之比較

<pre>sentence = “样子很好看，很潮不知道耐不耐摔，不过还是很满意。” s = SnowNLP(sentence) print(s.words) cutted_word = jieba.lcut(“sentence”, cut_all = “False”) print(cutted_word)</pre>
SnowNLP 原套件之分詞結果
['样子', '很', '好看', '很潮', '不', '知道', '耐', '不耐', '摔', '很', '满意']
Jieba 之分詞結果
['样子', '很', '好看', '很潮', '不知道', '耐不耐', '摔', '很', '满意']

資料來源：本研究編撰

本研究實驗在此階段為了提升模型對於繁體中文的判斷能力，是故將進行SnowNLP的情感分析中的handle函數的修正，改以Jieba分詞進行處理，如下（表18）。做此階段處理的主要原因為二：其一是Jieba套件的分詞能力是優於SnowNLP的seg函數；其二是本研究的研究領域是特別針對電商平台，是故前階段處理分詞時，已針對該領域的專業用詞，進行詞彙的新增以及正、負面文本之分類。

表18 將sentimnet(seg)函數改為jieba.lcut之程式碼

<pre>class Sentiment(object): def handle(self, doc): words = jieba.lcut(doc) # 原為 seg(seg) 函數替換為 jieba.lcut words = normal.filter_stop(words) return words def train(self, neg_docs, pos_docs): data = [] for sent in neg_docs: data.append([self.handle(sent), 'neg']) for sent in pos_docs: data.append([self.handle(sent), 'pos']) self.classifier.train(data) def classify(self, sent):</pre>

```
ret, prob = self.classifier.classify(self.handle(sent))
if ret == 'pos':
    return prob
return 1-prob
```

資料來源：本研究編撰

(3) 以單純貝氏分類器(Naïve Bayes Classifier Model)訓練模型

為了提高模型判斷正確率，以及預測消費者評論是否為正面還是負面，本研究利用sklearn模組裡的單純貝氏分類器(naïve_bayes)訓練自定義的模型，將在此小節說明。

a. Jieba 分詞

分詞方式已於上小節闡述，本小節特別針對訓練貝氏學習模型，將不一一贅述。首先，我們要對評論數據進行分詞，中文和英文不一樣，例如：i love dog，就是通過空格來分詞的。中文不一樣，例如：我喜歡狗，我們要分成我/喜歡/狗（通過空格隔開），這個主要是為了後面詞向量做準備。Jieba分詞之程式碼如（表19）所示，分詞結果呈現於（表20）之cutted_comment欄位。

表19 Jieba分詞之程式碼

```
import jieba
def chinese_word_cut(mytext):
    return " ".join(jieba.cut(mytext))
df["cutted_comment"] = df.comment.apply(chinese_word_cut)
print(df.head(3))
```

資料來源：本研究編撰

表20 分詞之結果於cutted_comment欄位

original_comment	cutted_comment
有点失望，才用十天就出现卡顿现象。虽然不是旗舰机，但这好歹是最新出的机器，还有苹果的拍照真的垃圾！！	有点 失望 ， 才 用 十天 就 出现 卡顿 现象 。 虽然 不是 旗舰机 ， 但 这 好歹 是 最新 出 的 机器 ， 还有 苹果 的 拍照 真的 垃圾 ！！
颜色特别喜欢，很经典，功能强大，性价比高！	颜色 特别 喜欢 ， 很 经典 ， 功能 强大 ， 性价比 高 ！
手机很精致，小巧很迷你，当之无愧	手机 很 精致 ， 小巧 很 迷你 ，

是小屏旗舰，电池用过七八个小时， 一点问题没有	当之无愧 是 小屏 旗舰 ， 电 池 用 过 七 八 个 小 时 ， 一 点 问 题 没 有
----------------------------	---

資料來源：本研究繪製

b. 劃分數據集

本組利用sklearn套件裡面的train_test_split模型劃分數據集。分類問題需要x（特徵），和y類別（label）。這裡分詞後的評論為x，情感為y。本組按照8:2的比例切分為訓練集和測試集。另外，這邊的y類別sentiment_star設定為2個標籤(0和1)，分別表示評論評分3分以下(包含3分)以及超過3分，如（表21）以及（表22）所示。

表21 劃分數據集

```
x = df["cut_comment"]
y = df.sentiment_star

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(x,y,test_size = 0.2,
random_state = 3)
```

資料來源：本研究編撰

表22 y類別標籤設定

```
def make_label(star):
    if star > 3:
        return 1
    else:
        return 0
df["sentiments_star"] = df.star.apply(make_label)
```

資料來源：本研究編撰

c. 詞向量(數據處理)

由於電腦是沒法識別文字的，只能識別數字。我們可以通過一個案例來說明詞向量，如（表23）所示。假設下面是我們的文本：“i love the dog”以及“i hate the dog”

表23 詞向量

i	love	hate	the	dog
1	1	0	1	1
1	0	1	1	1

資料來源: 本研究編撰

簡單的說，詞向量就是我們將整個文本出現的單詞一一排列，然後每行數據(評論)去映射到這些列上，出現的就是1，沒出現就是0，這樣，文本數據就轉換成了01稀疏矩陣，如下(表24)以及(圖6)所示。而本組使用sklearn中CountVectorizer方法常用的參數：

(a) **max_df:**

在超過這一比例的文檔中出現的關鍵詞(過於平凡)，去除掉。

(b) **min_df:**

在低於這一數量的文檔中出現的關鍵詞(過於獨特)，去除掉。

(c) **token_pattern:**

主要是通過正則處理掉數字和標點符號。

(d) **stop_words:**

設置停用詞表，這樣的詞我們就不會統計出來。

表24 關鍵字提取部分代碼

```
from sklearn.feature_extraction.text import CountVectorizer

def get_custom_stopwords(stop_words_file):
    with open(stop_words_file, encoding="utf-8") as f:
        stopwords = f.read()
        stopwords_list = stopwords.split('\n')
        custom_stopwords_list = [i for i in stopwords_list]
        return custom_stopwords_list

stop_words_file = 'stop_words.txt'
stopwords = get_custom_stopwords(stop_words_file)

vect = CountVectorizer( max_df = 0.85,
```

```
min_df = 3,
token_pattern=u'(?u)\b[^\d\\W]\\w+\\b',
stop_words=frozenset(stopwords))
```

資料來源：本研究編撰

	AJD	AJE	AJF	AJG	AJH	AJI	AJJ	AJK	AJL	AJM	AJN	AJO
1	非常	非常适合	音效	音质	順手	順滑	順暢	預期	預計	頻率	颜值	顏色
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	1	0	0	0	0	0	0	0	0	0
9	2	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0

圖6 詞向量以稀疏矩陣之部分展示

資料來源：本研究繪製

d. 訓練模型

本組使用的是單純貝式分類器(naïve_bayes)進行貝氏機器學習，而該分類器使用到的定理為貝氏定理，描述在已知某些條件下，計算某個特定事件發生的機率為何。公式如下（圖7）。

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

圖7 貝氏定理

資料來源：擷取網上公式

可以解釋為在事件B發生之前，我們對事件A會有一個基本的機率判斷，因此我們稱P(A)為”事前機率”。而在事件B發生之後，我們會對事件A發生的機率重新評估，因此我們稱P(A|B)為事件A的事後機率。另一方面，在事件A發生之前，我們也會對事件B有一個基本的機率判斷，我們稱P(B)為”事前機率”。同樣的，在事件A發生之後，我們亦會對事件B的發生機率從新評估，因此我們稱P(B|A)為事件B的事後機率。

綜合以上，在機器學習可以把方程式看成如（圖8）所示。在貝氏學習裡，我們想要得到的是，透過貝氏定理計算樣本在不同類別條件下的條件機率，並取得條件機率最大者作為預測類別。也就是要找到一個讓假設資料 $P(\text{假設}|\text{資料})$ 最高的”假設”。

$$P(\text{假設}|\text{資料}) = \frac{P(\text{資料}|\text{假設}) \times P(\text{假設})}{P(\text{資料})}$$

圖8 貝氏定理

資料來源：擷取網上公式

而我們將上述公式轉化成數學式子以配合我們在數據劃分時的x特徵以及y類別標籤。如（圖9）所示。其中 $X_i = \langle x_1, x_2, x_3, \dots, x_n \rangle$ ，代表著各個特徵。 y_i 則代表著所有結果中的其中一種結果，而發生機率最高的 $P(y_i|X_i, D)$ 裡的 y_i ，我們稱之為 y^* ，也就是分類器對該樣本最後判定的類別。其中貝氏分類器模型中假設所有特徵相互獨立。因此在公式的推導下，我們要找到給定特徵下事情發生機率最高的目標，只要計算 $P(y_i)$ 及 $P(x_1, x_2, \dots, x_n|y_i)$ 的相乘。而最後比較各自類別標籤的機率結果。

$$y^* = \arg \max_{y_i \in y} P(y_i|x_1, x_2, \dots, x_n)$$

$$y^* = \arg \max_{y_i \in y} \frac{P(x_1, x_2, \dots, x_n|y_i)P(y_i)}{P(x_1, x_2, \dots, x_n)}$$

$$y^* = \arg \max_{y_i \in y} P(x_1, x_2, \dots, x_n|y_i)P(y_i)$$

圖9 單純貝氏分類器

資料來源：擷取網上公式

另外，該組資料為離散式的特徵參數，因此使用MultinomialNB多項式的貝氏分類器進行學習。訓練模型如（表25）所示。而訓練以及測試模型的精度計算為加權平均，計算方式為最大機率類別標籤除以所有類別標前之相加。可以看到訓練模型精度來到0.9549左右，測試後發現模型精度也來到0.9349，相比直接利用SnowNLP套件的0.9184更為精準。

表25 訓練模型

```
from sklearn.naive_bayes import MultinomialNB

nb =MultinomialNB()
X_train_vect = vect.fit_transform(X_train)
nb.fit(X_train_vect, y_train)
train_score = nb.score(X_train_vect, y_train)
print(train_score)

# 0.95458440445586988 訓練模型

X_test_vect = vect.transform(X_test)
print(nb.score(X_test_vect, y_test))

# 0.934931506849315 測試模型
```

資料來源：本研究編撰

e. 觀察模型效能

利用matplotlib下的heatmap(熱力圖)來顯示混淆矩陣，印出的縱軸為測試資料，橫軸為預測結果。我們可以參考（圖10），True Positive代表預測結果為真，實際值也為真；True Negative代表預測結果為假，實際值也為假；False Positive代表預測結果為真，實際值卻為假；False Negative代表預測值為假，實際值卻為真。因此預測結果當然希望TP以及TN之值越高。混淆矩陣之程式碼如（表26）所示。由此可知，本研究訓練的模型偵測言論的準確率頗為精準，如（圖11）所示(confusion matrix左上角以及右下角)。再利用sklearn.metrics中classification_report觀察檢查報告。很清楚地可以看見實際為0，預測也為0之精度為0.96，實際為1，預測也為1之精度為0.93，如（圖12）所示。

		prediction outcome		
		p	n	total
actual value	p'	True Positive	False Negative	P'
	n'	False Positive	True Negative	N'
total		P	N	

圖10 混淆矩陣

資料來源：擷取網上公式

表26 模型效能報告之程式碼

```

from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

y_predict_test = nb.predict(X_test_vect)
cm = confusion_matrix(y_test, y_predict_test)
sns.heatmap(cm ,cmap = "Blues", annot = True)
plt.show()
print(classification_report(y_test, y_predict_test))

```

資料來源：本研究編撰

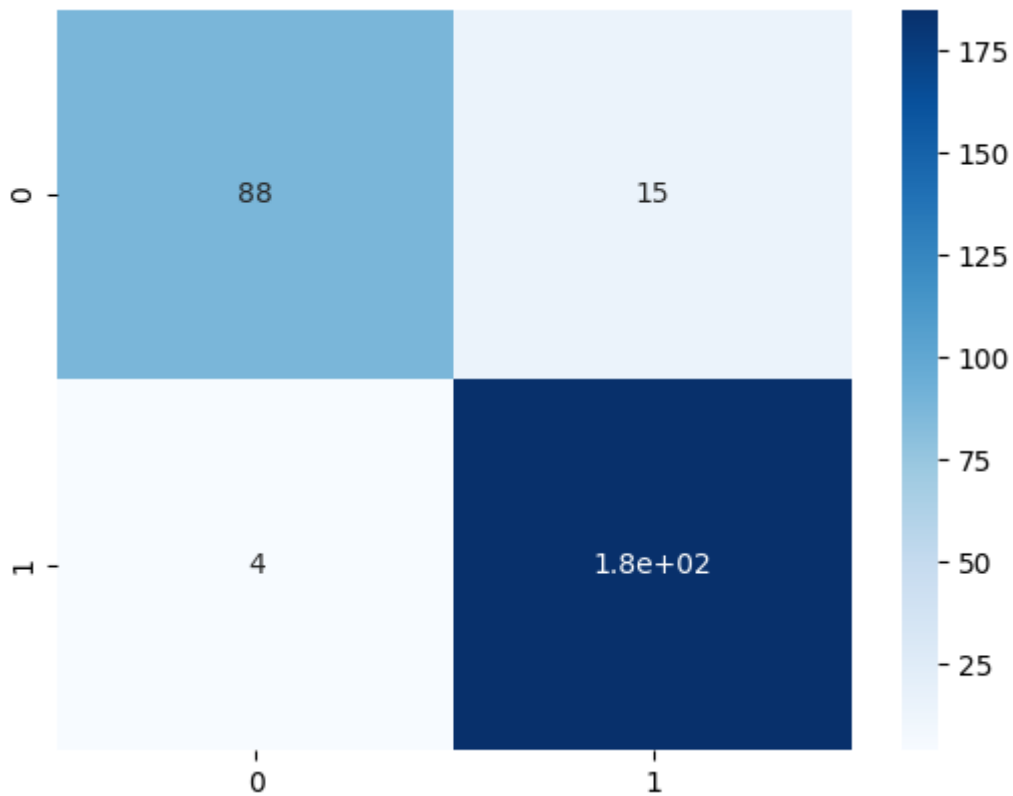


圖11 利用熱力圖印出混淆矩陣

資料來源: 本研究繪製

	precision	recall	f1-score	support
0	0.96	0.85	0.90	103
1	0.93	0.98	0.95	189
accuracy			0.93	292
macro avg	0.94	0.92	0.93	292
weighted avg	0.94	0.93	0.93	292

圖12 檢查報告

資料來源: 本研究繪製

(三) 分析結果

1. 單純貝式分類器之機器學習模型預測

透過上一小節之程式編撰與分析，我們自定義了正、負面文本，結合Jieba和SnowNLP進行分詞，再將資料分成訓練集以及測試集，透過Naïve Bayes Classifier進行模型之訓練、判讀以及預測，最後將結果以視覺化方式呈現(判讀標籤0，也

就是評論星數為3顆以下(含)之精度為0.96，判讀標籤1，也就是評論星數為4顆以上之精度為0.93)。之後便可以利用這個訓練模型重新導入資料，觀察商品評論之時間序列對照之情感分數。並以接下來之繪圖模組呈現。

2. 情感分析之分數界定

根據SnowNLP的Sentiment使用，其所顯示之數值表示文本是屬於正面情緒之機率，也就是說，若該文本之判定數值為0.685，表示有68.5%該段文本屬於正面情緒；若為0.245，則表示僅有24.5%的機率屬於正面情緒。是故本研究在此界定正、負面情緒文本，若分數大於0.7則判定為正面情緒；分數小於0.4，則表列為負面情緒；介於0.4~0.7之間之文本，則認列為中立文本，沒有特別偏向的情緒。

3. 評論時間序列可視化

透過Python套件中ggplot繪圖時間軸與情感分析，可以方便我們檢視在不同期間內，使用者們的情感分數有沒有太大的落差；其次，該套件可以透過使用放大鏡縮小與放大，讓我們在檢視時，也可以特別針對一段期間進行檢視。若情感分數落差大，則圖表會有較密集的塗色，如（圖13），以mini款項為例。11月初至11月底之顏色最為飽和，表示該期間使用者們的情感分數是最不一致的。這時若我們要進一步檢視，則可特別選取該段，放大進一步查看。

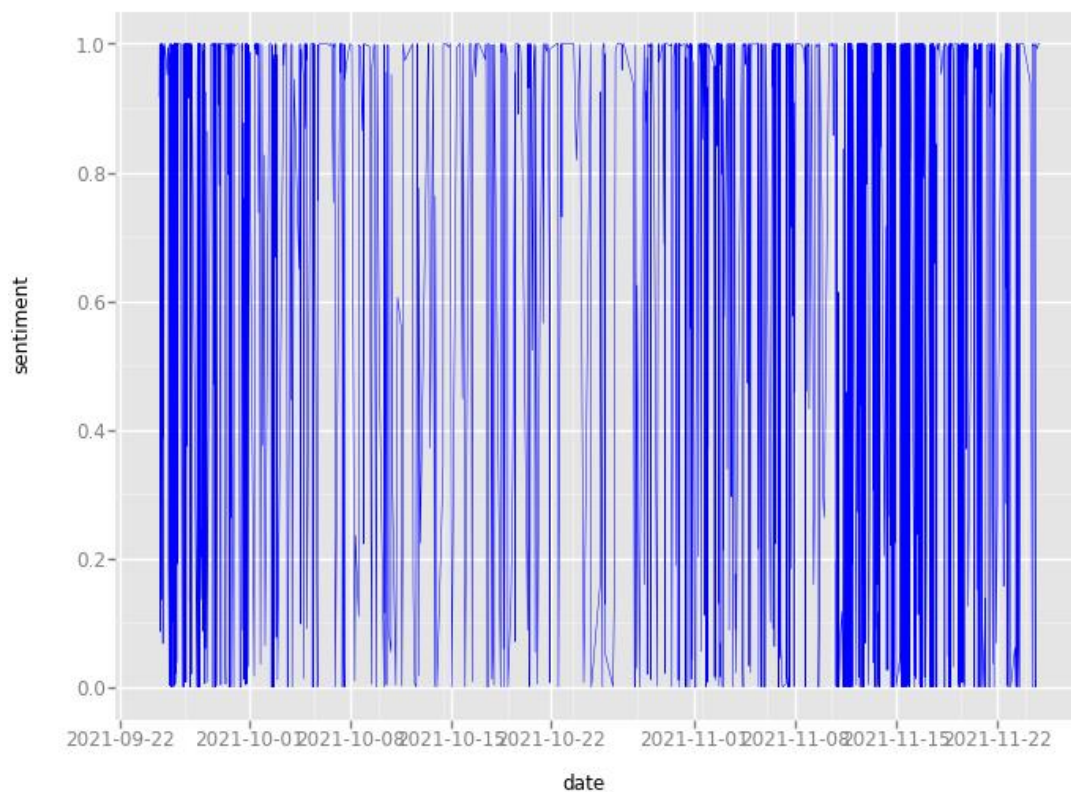


圖13 mini款項手機順時間演進的情感分數

資料來源: 本研究繪製

透過進一步放大，我們可以發現在11月初至11月底這段期間，確實可以看見節點增加代表的購買量上升，情感分數波動也更大。如（圖14）所示。這時可發揮文本情感分析之兩點效益，第一為研究者與從業人員可以透過檢視特定時期的情感分數，更進一步瞭解為什麼使用者會給出負面情緒，第二為可以檢視為什麼使用者在原系統的量化評分可能沒有給出低分，卻在文本顯示出現負面情緒，反之亦然。

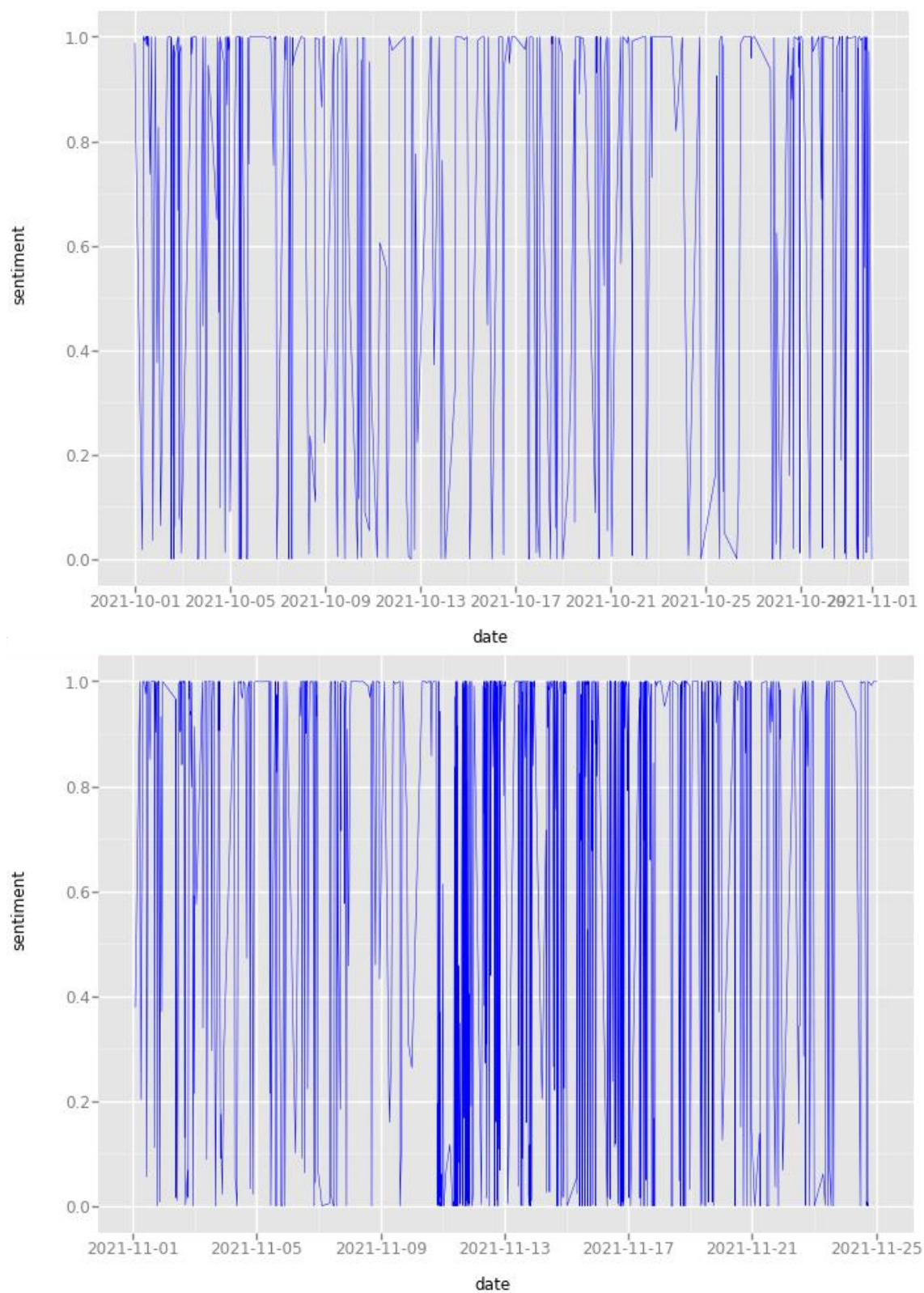


圖14 mini款項11月與其餘時間之情感分數比較

資料來源：本研究繪製

透過鎖定特定時間區塊的評論，我們可以到Excel的資料庫中清楚認識原始資料庫中的評論內容；我們可以在11月初至11月底這段期間發現，當時較多使用者不滿意的產品是特點是其「屏幕」與「電池」，以及商城本身「客服」、「降價」等問題。從（表27）所示，如本研究之mini款項編號442之評論為：「说实话这款13mini用得比较失望，居然会发热！电量挺不了一天…连玩个消消乐都发热…真的醉了，我要买个发热掉电的手机我买小米不是更香吗？而且mini也没有想象中轻薄，手感比老公之前用的7差多了……虽然流畅度还可以，但iPhone在我眼里香了6年，现在不香了……下次一定是国产机了……」雖然其在評分系統上給了看似中立的3顆星(最高為5顆，最低為1顆)，但其評價明顯是給該產品留下不好的使用者經驗，透過sentiment情感分析之分數為0.446352便可更清楚察覺到使用者的不滿意；又如編號439之評論「手感刚刚好，特别的丝滑」之評價同樣為3顆星，但其整體而言是在稱讚該產品的使用效果，是故其情感分數為0.967399。由此可見，若單憑藉系統的量化評分，還是難以察覺到使用者的「負面情緒」，因為使用者習慣會以「保守」的評論方式留下消費者體驗。但即便如此，透過情感分析的方法，可以快速進行量化評分的比對，同時可以透過篩選的方式，更快地去檢選出情感分數較低的資料，進而放大檢視該資料。

表27 mini款項之參照推薦星數與情感分析之分數

編號	推薦星數	評論	情感分數
439	3	手感刚刚好，特别的丝滑	0.967399
440	3	刚到货 开箱就有划痕 充点口有不明粘胶物 手机是真的很小巧 握在手一块玉一样	0.423954
441	3	小巧	0.947674
442	3	说实话这款 13mini 用得比较失望，居然会发热！电量挺不了一天…连玩个消消乐都发热…真的醉了，我要买个发热掉电的手机我买小米不是更香吗？而且 mini 也没有想象中轻薄，手感比老公之前用的 7 差多了……虽然流畅度还可以，但 iPhone 在我眼里香了 6 年，现在不香了……下次一定是国产机了……	0.446352

資料來源：本研究編撰

參、 結果

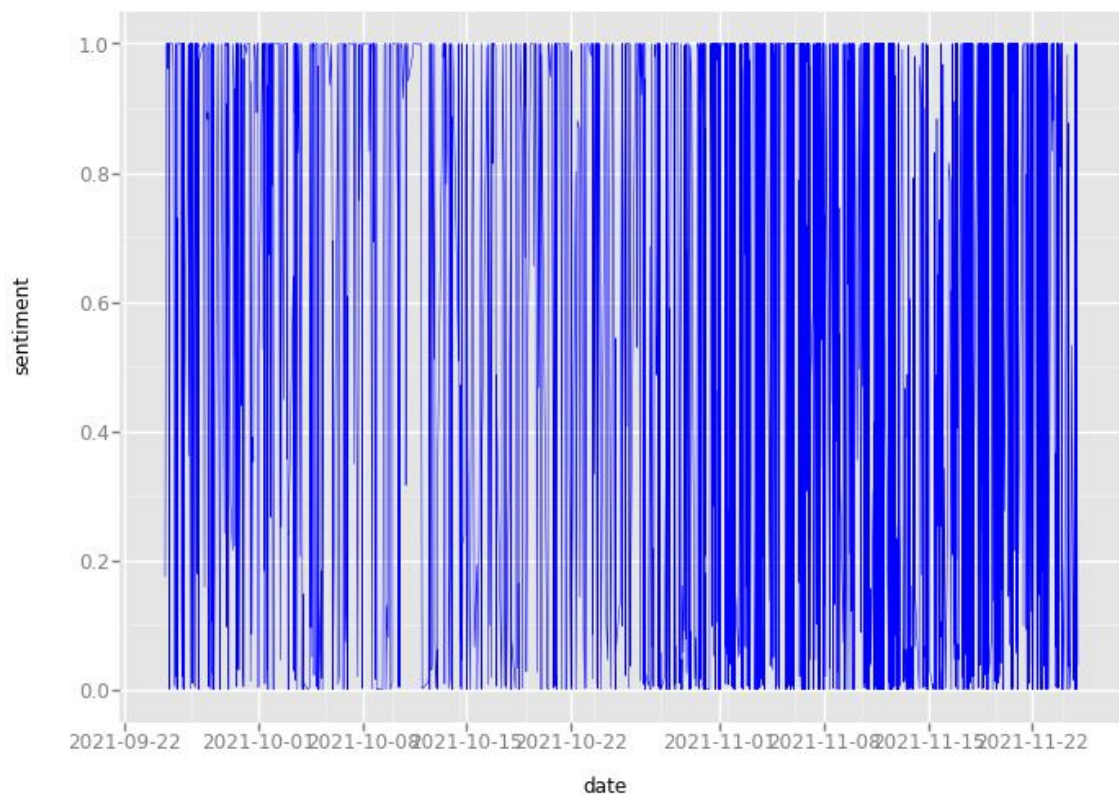
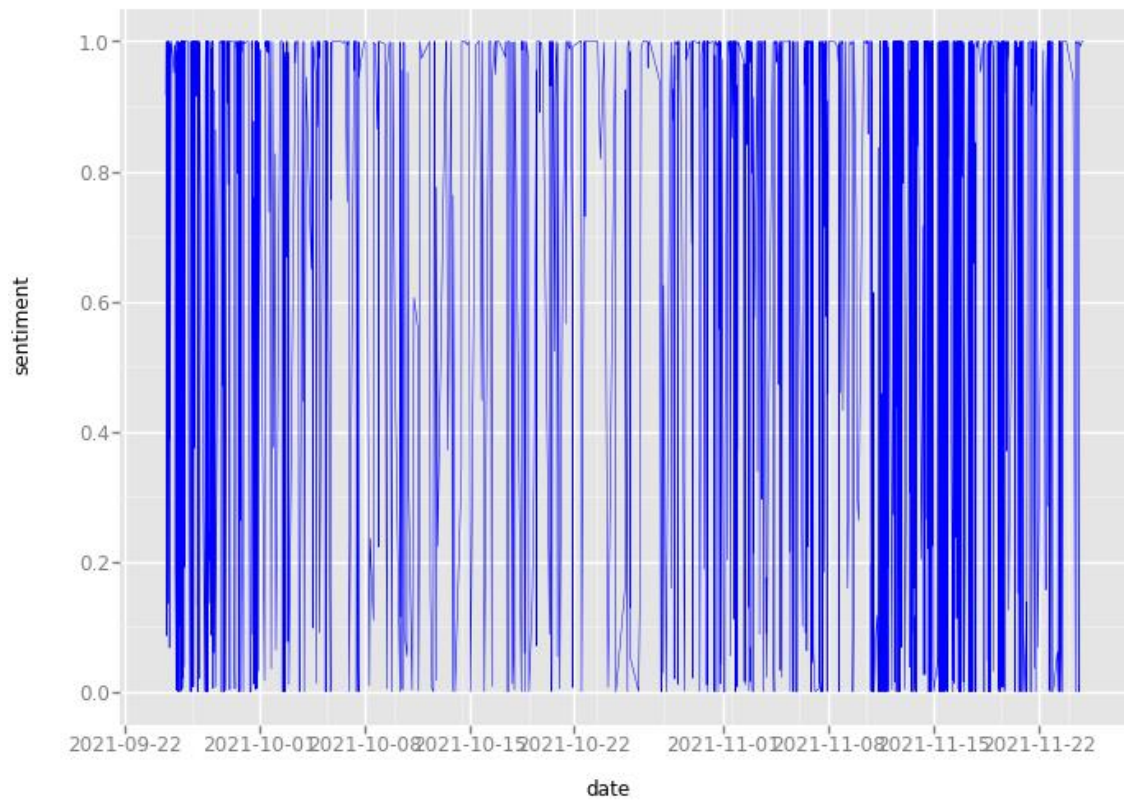
一、 評估結果

(一) 從資料推判評價習慣

從敘述性統計資料，與後半情感分析分數相互參照，研究發現部分使用者即使對產品有負面觀感，也不顯示在系統原有的量化指標。如本研究蒐集之樣本，原系統的分數尺度為 1~5 分，1 分為非常不滿意；5 分為非常滿意；本研究的情感分析之機率分數若大於 0.7，則表示正面情緒；介於 0.7 與 0.4 之間，表示無明顯偏向之中立情緒；低於 0.4 則表示負面情緒。從使用者評分之資料檢視，3 顆星以下(含)之總樣本數共 806 筆，佔總樣本數 9%，然而從情感分析之分數判斷，負面情緒樣本數共有 2285 筆，佔總樣本數 28%。根據結果顯示，用者在量化評分上有明顯的保守趨勢，證明進一步檢視質性文字的重要。

(二) 不同月份的消費量

由（圖 15）所示，綜合四項品項之情感分析以及時間軸，可以發現在兩個時期明顯的看到波動幅度很大，消費量大增。第一個時期為蘋果公司正式推出 iPhone13 系列時，消費者首批預購量大增，短短幾天評論數累積超過 1000 筆，佔總屏論述 13%。第二個時間點為雙 11 購物節促銷時，因為搭配許多促銷方案也導致購買量大增，促銷期間評論數累積達到 2500 筆左右，佔總評論數 30% 左右。兩個時間點在整個時間軸佔了 43% 左右的評論數。



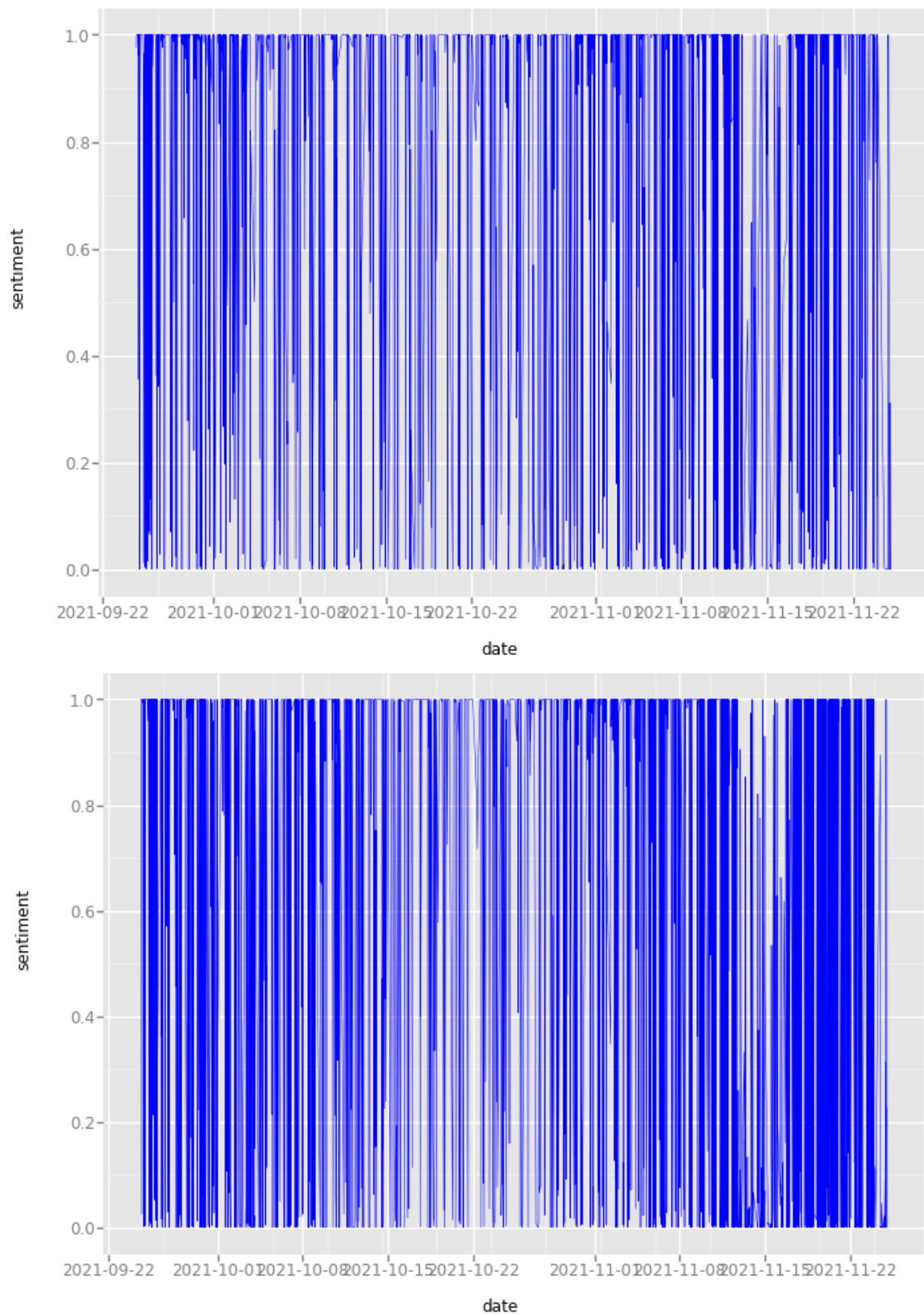


圖15 以時間序列觀察全品項情感分數之比較

資料來源：本研究繪製

(三) 蘋果手機在電商平台之行銷價值特徵

1. 產品外觀

透過TF-IDF之產品外部關鍵詞分析，我們可得知消費者重視的產品外觀包括「屏幕」是否有損壞，或是其大小。特別值得一提的是，從使用者文字發現，mini款項的「小巧」、「小屏」等相關外觀特徵，分別被提及了143次以及63次。相較於pro max系列的「大屏幕」僅僅只被提及了18次。也就說如果依外觀之「屏幕」大小做為標準，消費者理當來說會想要購買mini款項產品不外乎是因為產品本身的輕巧，以及小巧可愛的外型；而pro max款項不外乎是因為其大屏幕展現的大氣。但如統計所示，消費者確實特別在乎mini款項之螢幕大小，卻沒有對於pro max系列之螢幕大小多做評論。證明了消費者其實對於pro max手機尺寸的大小並沒有對於mini款項來的重視。因此廠商可以著重在pro max本身的其他條件上，減少對手機尺寸大小的要求、著墨。

另外值得一提的是，「粉色」在全部品項被提及了足足有231次，比起刻板印象中的黑色或是白色等等中性顏色，壓倒性的在所有機身顏色中脫穎而出。因此廠商亦可以針對此特色發揮更多行銷之價值。

2. 電池容量的擴充以及效能提升

在智慧型手機產業中，電池之續航力必然是每位消費者重視之特徵。本研究中，透過產品內部關鍵詞分析，電池不例外的是每個品項的關鍵字，足足被提及了492次。消費者重視手機的電池健康度以及可使用的時間，以及電池本身是否會因為過熱而導致電池本身的折損。由此可推測，此特徵是廠商可以重視、精進的一環。

另外在「運行」、「速度」、「流暢」等等效能方面之特徵也被提及了超過千次，尤其是此次升級的「A15」晶片。大部分的評論顯示都為正面，只有少部分消費者認為比起原先iPhone12的「A14」晶片，升級較為無感。因此廠商也可以特別針對內部處理器的部分多多著墨。

3. 物流速度以及運送時之防護

於電商平台購買商品本預期會高頻出現「物流」等等相關字眼，查看評論確實可以看見被提及了332次，也可以看見大部分商品送達耗費時長平均為一至兩個禮拜，但消費者在配送速度上面並沒有太多的反彈。原因可能為消費者因為高價格願意等待的時間更長。然而廠商依然可以更加重視物流速度。

另外，「划痕」、「刮痕」等等高機率因物流導致的產品瑕疵雖然並沒有顯示在高頻的關鍵字上，但透過人工觀察幾則負面評論，依然可以看見幾則評論針對產品的損壞而感到不滿意。因此廠商在重視物流速度的同時，也應該確保商品配送之防撞保護機制是否完善。

4. 定價促銷方案的調整

透過人工觀察幾則評論，雖然「價格」並未顯示在高頻的關鍵字中，但仍有幾則評論認為花這筆錢購買此款iPhone13不划算，原因是許多消費者購買此款手機的幾個禮拜之後就碰上雙11購物節，心理上有較大的落差。廠商可以評估非節慶之售價以及逢節慶時之定價的調整，以及促銷方案是否更能符合消費者之預期。

肆、 結論

本研究將相關人口變數之研究做為輔助觀察，主要分析乃是透過機器學習的方式建立新的訓練模型，將評論透過人工的方式編寫為自定義之正、負面文本，再透過結合 Jieba 以及 SnowNLP 進行整理，並將資料分成訓練集以及測試集以單純貝氏分類器進行模型之訓練以及預測，藉此訓練出更精準的情感判讀模型。又根據 TF-IDF 關鍵詞提煉出使用者的重點詞彙，以及利用自定義之文本進行情感分析，使電腦能對評論文字進行情緒之正負解讀，藉此剖析使用者更深層的使用體驗，以及不常反應在量化評分系統上之負面使用想法。透過本研究，著實發現於電商平台領域之詞彙有其特殊性，若欲對其進行文本分析，需透過特定編寫之詞庫以實現之。文末結果也指出，根據此方法可以判讀使用者更真實的正、負面分數。無獨有偶，亦能透過該研究方式，透析某款手機在電子商務市場的獨特行銷價值，以供從業人員與研究者更豐富的視野，瞭解使用者對產品之觀感。

伍、 參考文獻

1. 博、碩士論文

黃怡翔，2012。《應用文本探勘技術於公開來源情報分析》，國立高雄大學資訊管理學系碩士班碩士論文。

- 林育漢，2013。基於文本探勘技術偵測惡意網站程式特徵分析之研究，國立高雄大學資訊管理學系碩士班碩士論文。
- 黃梓瑞，2016。運用文本探勘技術於基於人格特質之遊戲推薦之研究，國立高雄大學資訊管理學系碩士班碩士論文。
- 畢方綺，2017。台灣垂直電商平台經營策略研究，國立中山大學企業管理學系研究所碩士論文。
- 蘇義雄，2017。使用文本探勘在伺服器開發上建立無效的缺陷分類模型，國立台灣科技大學管理研究所博士論文。
- 張玉婷，2019。搜尋性商品與經驗性商品之網路口碑研究—以電商平台亞馬遜為例，東吳大學企業管理學系碩士論文。
- 楊浣而，2019。跨境電商平台的整合行銷溝通對消費者購買意願之影響：信任與知覺價值之角色，國立台北大學企業管理學系碩士論文。
- 蘇怡寧，2019。消費者使用電商平台購物的行為模式-以社群團購群組為例，國立中山大學企業管理學系研究所碩士論文。
- 廖芯榆，2020。探討消費者對品牌社群電商平台之購買意圖，國立中山大學電子商務與商業分析數位學習碩士在職專班碩士論文。
- 陳姿妤，2020。中小企業運用 B2B 跨境電商平台成功因素之研究-以阿里巴巴平台營運為例，國立臺北科技大學工業工程與管理系碩士論文。
- 田心茹，2021。消費者滿意度的文本挖與分析—以某品牌手機為例，中國安徽財經大學應用統計研究所碩士論文。
- 林崇德，2021。基於搜尋引擎與文本探勘之相似文件檢測系統，國立暨南國際大學資訊工程學系碩士論文。

2. 期刊論文

- 林柏宇、謝邦昌、廖佩珊，2016。基於 Python 之文字探勘平臺，*Journal of Data Analysis*，第十一卷第六期，35-61。
- 黃芝璇、馬麗菁，2021。以圖形方式摘要化顧客評論，*資訊管理學報*，第二十八卷第二期，125-153。

楊亨利、林青峰，2020。針對情感商品的推薦機制－以流行音樂為例，*資訊管理學報*，第二十七卷第二期，175-204。

3. 網站文章

Machine Learning-訓練 Naive Bayes Classifier Model，預測 Twitter 發文是正面還是負面[kaggle]。取自：

<https://medium.com/@a626854993/machine-learning-%E8%A8%93%E7%B7%B4naive-bayes-classifier-model-%E9%A0%90%E6%B8%ACtwitter%E7%99%BC%E6%96%87%E6%98%AF%E6%AD%A3%E9%9D%A2%E9%82%84%E6%98%AF%E8%B2%A0%E9%9D%A2-kaggle-ade0c051729f>

[Python]實作單純貝氏分類器(Naive Bayes Classifier)，並應用於垃圾訊息分類。取自：

<https://medium.com/qiubingcheng/%E5%AF%A6%E4%BD%9C%E5%96%AE%E7%B4%94%E8%B2%9D%E6%B0%8F%E5%88%86%E9%A1%9E%E5%99%A8-%E4%B8%A6%E6%87%89%E7%94%A8%E6%96%BC%E5%9E%83%E5%9C%BE%E8%A8%8A%E6%81%AF%E5%88%86%E9%A1%9E-6b26834c4fd8>

【Python 機器學習】102：如何使用 Scikit-Learn 預測器完成數值預測。取自：

<https://medium.com/%E5%B1%95%E9%96%8B%E6%95%B8%E6%93%9A%E4%BA%BA%E7%94%9F/python%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92%E5%85%A5%E9%96%80-%E4%BA%8C-bbc8414d4632>

Python 有趣|中文文本情感分析。取自：

<https://read01.com/6Bz62gk.html#.YZRgBmBBzIX>

文科生数据科学上手指南。取自：

<https://bookdown.org/wshuyi/dive-into-data-science-practically/>