# 國立臺灣大學
# 資訊管理學系暨研究所

# 資訊管理期末專案
# 社群平台之假評論預測模型
# 及假評論寫手行為研究

**第八組**


R09725008 林聖典

R09725009 顏煥勳

R09725011 陳宇鑫

R09725024 彭琮鈺

R09725052 余俊廷


**民國 110 年 1 月 13 日**

# 目錄

# 一、研究動機及專案目標

線上評論在社群平台中是一種熱門的資訊分享形式，使用者能夠依據自身經歷將與店家的消費經驗分享至平台上。在電子商務為消費主流的網路時代，多數消費者在決定是否要購買產品之前會蒐集其他消費者的相關心得評論，若是看到該店家的評論多為正向的將會提升自己的購買意願，反之亦然。然而有些不肖業者為了使自家產品在評論上脫穎而出，便會聘請一些『專業寫手』替該業者撰寫不符合事實的『假評論』，企圖欺騙消費者進而影響消費者的購買意願。

在台灣最知名的便是 2013 年的三星寫手門事件，三星透過公關公司鵬泰操作議題，散播 HTC 的不實使用心得藉此抹黑競爭對手。又或者是中國知名的旅遊網站《馬蜂窩》 被媒體踢爆該網站超過 85%的點評是透過虛假帳號所產生，以上事件皆顯示虛假評論會對於社會存在不良影響。

Yelp 為美國知名的評論社群平台，它不只是美食情報網，其內容也涵蓋購物、食品、美容、運動休閒等類別，使用者可針對特定店家進行評論、上傳照片，或是將評論內容轉發分享他人，以及配合帳號登入進行收藏等操作。然而過去研究指出在 Yelp 平台中有將近 20 % 的評論是由受僱的寫手撰寫出的假評論 ，Yelp 也因為假冒評論情形過於嚴重，曾在 The New York Times 撰文批評與公開他們所發現的不肖業者 。

因此為了提供一個更為可信的評論資訊，本計劃目標在建立出一個假評論分類模型能夠從眾多評論中過濾出虛假評論，協助平台業者 ex.Yelp 過濾虛假評論並重新計算店家評分，使用戶能夠更有參考性的依據評論做出購買決策，讓消費行為不會受到虛假評論而有所影響。

# 二、相關文獻探討

假評論預測的研究有使用監督式學習的方式透過語言學特徵 Linguistic feature 以及用戶行為 User-behavioral feature 來建立分類模型，語言學特徵包含 word unigrams、bigrams、 語文探索與字詞計算 LIWC、詞性分析 POS 等，用戶行為特徵則涵蓋如平均留言長度、評論星星數標準差等，而目前在機器學習方法中常見的分類模型有四種 Logistic Regression、Naive Bayes、Support Vector Machine 以及 XGBoosts，也有以深度學習方法如 LSTM、CNN 建構分類模型，以下將分別介紹 XGBoost 與 LSTM 的方法：

XGBoost：

XGBoost 是基於 Gradient Boosted Decision Tree (GBDT) 改良與延伸，是一種基於決策樹的集成機器學習算法，採用了梯度提升（Gradient Boosting）框架，常被應用於解決監督式學習的問題。在許多資料分析的競賽，例如 Kaggle，都有良好的表現。

長短期記憶模型 LSTM：

與一般的時間序列神經網路相比，長短期記憶模型（LSTM）增加 3 個不同的閘門，輸入閘 （Input gate）、遺忘閘（Forget gate）、輸出閘（Output gate）。LSTM 透過記憶細胞的狀態（Cell state)，可以通過門控狀態來控制傳輸狀態，記住需要連續記憶的，忘記不重要的資訊，目前 LSTM 已經經常用於各種建模與預測問題，例如文字生成、機器翻譯、語音識別、生成影像描述和視訊標記等。

除了上述對於機器學習方法的文獻探討，過往也有許多對於虛假評論偵測的模型，以下分別針對兩篇虛假評論偵測論文進行探討：

論文 1：Fake Review Detection on Yelp Dataset Using Classification Techniques in Machine Learning https://ieeexplore.ieee.org/document/9055644

論文提出了運用四種不同的機器學習方法建立假評論偵測模型，並且比較各個不同方法間假評論偵測模型的差異，但是論文並沒有使用到深度學習方法，以及對於真實評論與寫手之評論做更進一步的探討分析。

論文 2：應用深度學習技術於網路虛假評論偵測

http://jeb.cerps.org.tw/files/JEB2019-008.pdf

論文使用深度神經網路(DNN)、卷積神經網路(CNN)、長短期記憶(LSTM)等深度學習方法建構網路虛假評論偵測模型，並用於台灣知名論壇的虛假評論上，以提供提供網路評論的平台一個監控的機制，可以預警或是篩選有問題的評論，以避免不實評論對廠商或是消費者造成傷害。

# 三、研究模型及設計

## 3.1 研究目標

### 研究目標一：比較真實評論與寫手之評論差異

本計劃研究目標一即在於將 YelpZip Dataset 所提供的資料集，對標記為虛假的評論與真實的進行比較，檢查假評論是否如預期一樣有較短的篇幅、寫手們會透過特定『模板』來產出假評論 、寫手會統一在上班時間(朝九晚五)才會發表評論或是評論是否語意通順，而不是隨意組成，也會將兩者評論做圖表比較，例如透過長條圖、文字雲做進一步分析。

### 研究目標二：假評論預測模型

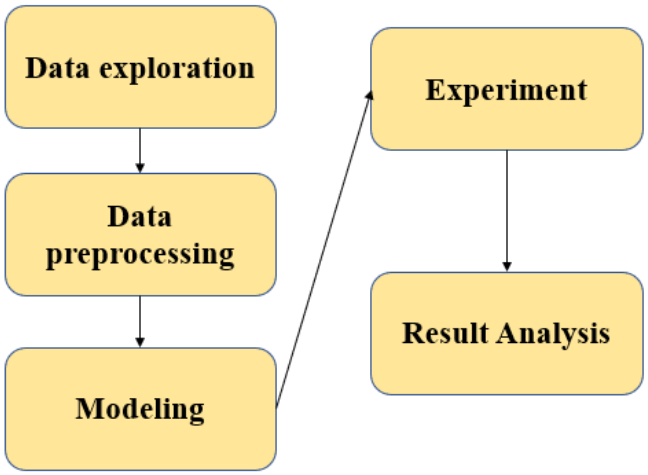
本計劃研究目標二在於觀察研究目標一的結果，增設相關變數，建立假評論分類模型，並將該模型應用於 Yelp Dataset 中的店家評論進行分類，在過濾出可疑評論之後重新計算店家分數，使用戶能夠更有參考性的依據評論做出購買決策 。本研究預計採取數種機器學習方法，並且比較各個不同模型間表現的差異。

### 研究目標三：將模型應用於其他平台之可能性（EX: Google Play）

本計劃研究目標三在於探討將先前建立的預測模型應用於其他平台上的可能性，不同平台評論的特徵差異可能會對於虛假評論的檢測模型造成影響，但若能夠將模型擴大應用於其他平台上，模型的應用層面將會更為廣泛。

## 3.2 研究流程

我們將研究流程大致分為五大步驟，如下圖所示。




[圖一] 研究流程圖。

## 3.3 實驗設計

### Experiment1. Text

考慮真假評論在內容、用字遣詞上可能有所差異，擬將每則評論經過前處理之後轉換至文字向量並作為模型輸入，欲找出高鑑別力的 terms 來提升辨別真假評論的準確率。

### Experiment2. Behavior

試想評論寫手與一般平台用戶因為在撰寫的評論的動機不同而有不同的使用者行為，本團隊可以藉由資料集中的使用者 ID 作為索引連結出自於同一帳號的評論並觀察其在不同則留言間的差異，意即觀察評論寫手與一般用戶在平台上留言的習慣與差異。是故，本團隊擬將透過資料探索，建立數個行為特徵作為模型輸入，藉此進行真假評論的判斷。

### Experiment3. Text + Behavior

結合前兩項實驗所有的特徵，同時考慮了留言內容本身以及留言者的行為特徵來對評論進行分類。

# 四、資料處理、建模及實驗結果分析

## 4.1 Dataset

我們在專案初期蒐集到兩公開資料集，分別為「Yelp Dataset」以及「YelpZip Dataset」，未來可能延伸至自行建構 Google Play 或 Google Map 上的評論資料集作後續相關研究探討。

### Yelp Dataset

https://www.kaggle.com/yelp-dataset/yelp-dataset/

此為 Yelp 官方公開至 Kaggle 平台給有興趣的學術單位作相關研究的資料集，其中包含了橫跨 11 個大都會地區，174,000 間店家資訊以及 5,200,000 的用戶評論。

| Business | | | Review | | User | |
|---|---|---|---|---|---|---|
| **Attribute** | | **Data Type** | **Attribute** | **Data Type** | **Attribute** | **Data Type** |
| business_id | | string | review_id | string | user_id | string |
| name | | string | user_id | string | name | string |
| address | | string | business_id | string | review_count | int |
| city | | string | stars | int | yelping_since | string |
| state | | string | useful | int | useful | int |
| postal_code | | string | funny | int | funny | int |
| latitude | | float | cool | int | cool | int |
| longitude | | float | text | string | elite | string |
| stars | | float | date | string | friends | string |
| review_count | | int | | | fans | int |
| is_open | | int | | | average_stars | float |
| | BusinessAcceptsCreditCards | string | | | compliment_hot | int |
| | BikeParking | string | | | compliment_more | int |
| attributes | GoodForKids | string | | | compliment_profile | int |
| | BusinessParking | string | | | compliment_cute | int |
| | ByAppointmentOnly | string | | | compliment_list | int |
| | RestaurantsPriceRange2 | string | | | compliment_note | int |
| categories | | string | | | compliment_plan | int |
| | Monday | string | | | compliment_cool | int |
| | Tuesday | string | | | compliment_funny | int |
| | Wednesday | string | | | compliment_writer | int |
| hours | Thursday | string | | | compliment_photos | int |
| | Friday | string | | | | |
| | Saturday | string | | | | |
| | Sunday | string | | | | |

[圖二] Yelp Dataset 資料欄位名稱（部分表格）。

## YelpZip Dataset

http://odds.cs.stonybrook.edu/yelpzip-dataset/

此為 Prof. Rayana 在 Stony Brook University 做相關研究時所用的資料集，一共含有來自 260,277 位評論者對 5,044 家餐廳的 608,598 條餐廳評論，各評論也包含了過濾（假）與否（真）的標籤。

Review metadata:

- date (date)
- review ID (text)
- reviewer ID (text)
- business ID (text)
- label (Y/N)
- useful (integer)
- funny (integer)
- cool (integer)
- stars (integer)
- review (text)

[圖三] YelpZip Dataest 資料欄位名稱。

由於 YelpZip Dataset 有包含過濾（假）與否（真）的標籤，符合我們原先預建立假評論分類模型的專案目標，因此最終我們選定該資料集作為之後主要的實驗資料。

## 4.2 Data exploration

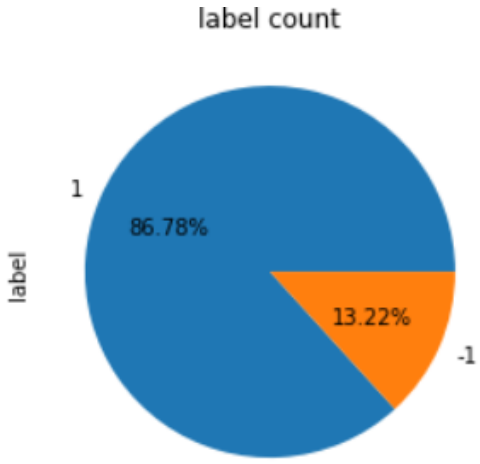
由於一開始我們拿到的 YelpZip Dataset 以類似「關聯式資料庫」的形式將資料分為好幾個資料表儲存(例如含有專門記錄使用者資訊的資料表、專門記錄產品資訊的資料表等)，必須先將這些資料表進行合併，而合併之後的結果如下圖所示。

| | userId | prodId | date | content | content_len | label | rating |
|---|---|---|---|---|---|---|---|
| **0** | 5044 | 0 | 2014-11-16 | Drinks were bad, the hot chocolate was watered... | 36 | -1 | 1.0 |
| **1** | 5045 | 0 | 2014-09-08 | This was the worst experience I've ever had a ... | 248 | -1 | 1.0 |
| **2** | 5046 | 0 | 2013-10-06 | This is located on the site of the old Spruce ... | 50 | -1 | 3.0 |
| **3** | 5047 | 0 | 2014-11-30 | I enjoyed coffee and breakfast twice at Toast ... | 233 | -1 | 5.0 |
| **4** | 5048 | 0 | 2014-08-28 | I love Toast! The food choices are fantastic -... | 152 | -1 | 5.0 |

[圖四] 原始資料合併後示意圖

### 從評論數來看

所有的評論共有 608,598 則評論，在所有評論當中，共有 528,132 筆真實評論，以及 80,466 筆虛假評論。其中這些評論分別由 260,277 為使用者所寫，總共評論了 5,044 個不同的餐廳。




[圖五] 真實評論與虛假評論占比

## 從每一則評論上來看(原始評論資料)

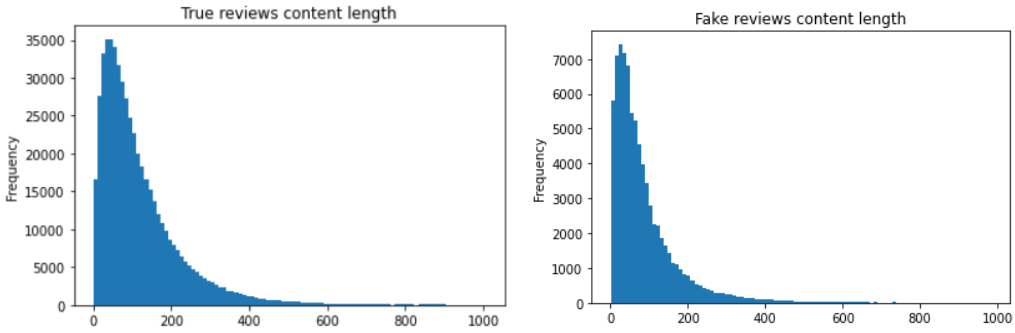
### 真評論與假評論字數的差異

```
count    528132.000000
mean        119.589269
std         107.374770
min           1.000000
25%          47.000000
50%          89.000000
75%         158.000000
max        1004.000000
Name: content_len, dtype: float64
```

```
count    80466.000000
mean        87.095991
std         90.641654
min          1.000000
25%         30.000000
50%         60.000000
75%        112.000000
max        983.000000
Name: content_len, dtype: float64
```

[圖六] 真評論與假評論字數的差異比較

上圖可以看出真評論的字數大多會比假評論還要長，我們覺得可能是因為假評論寫手因為要寫多篇文章，所以較不會認真寫，或是因為假評論寫手並沒有親身體會，所以比較無法寫出更多的具體內容。



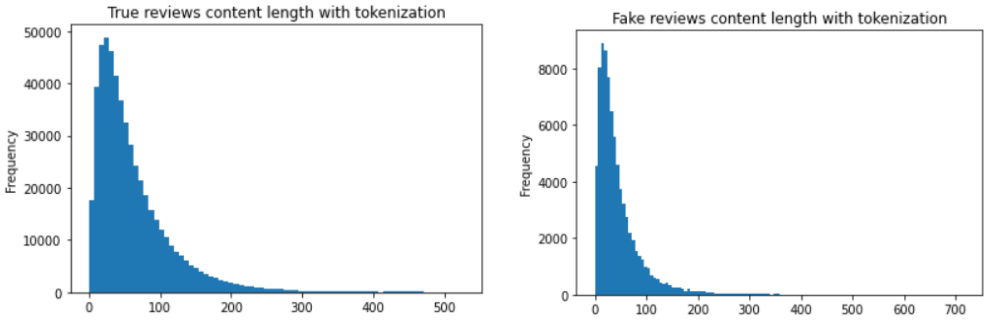

[圖七] 真假評論長度分布

```
count    528132.000000
mean         62.673322
std          55.469286
min           0.000000
25%          25.000000
50%          47.000000
75%          82.000000
max         527.000000
Name: tokenize_content_len, dtype: float64
```

```
count    80466.000000
mean        44.929088
std         45.371750
min          0.000000
25%         17.000000
50%         32.000000
75%         57.000000
max        716.000000
Name: tokenize_content_len, dtype: float64
```

[圖八] 真評論與假評論字數的差異比較(經過 tokenize)

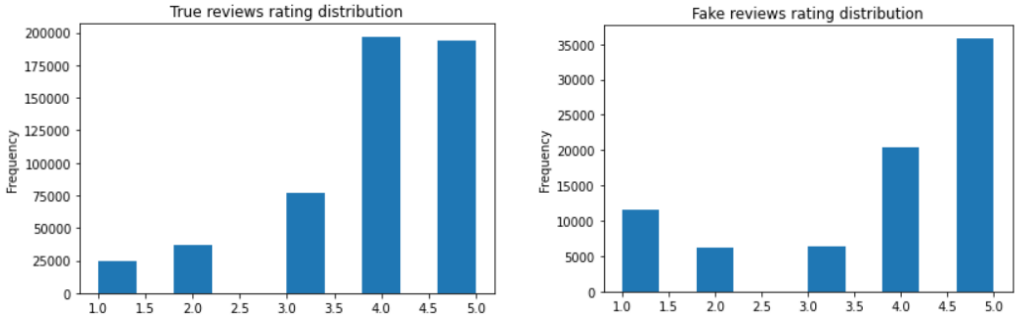
在經過 tokenize 之後，我們去掉了大約一半的字，從上圖可以看出真評論的字數仍然大多比假評論還要長，所以我們在 features 中新增了 tokenize_content_len(經過 tokenization 後的評論長度)作為我們其中一項變數。




[圖九] 真假評論長度分布(經過 tokenize)

**真評論與假評論評分(rating)的差異**

在用戶評論評分方面，我們期望透過使用者給予餐廳的評分找出是否對於真評論或假評論有所差異。



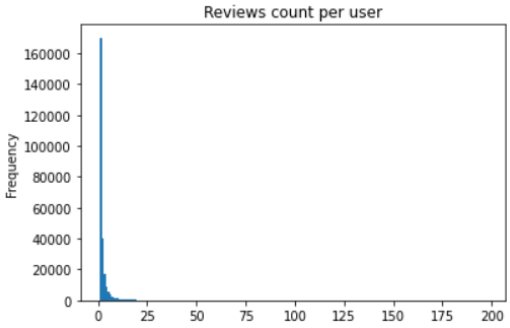

[圖十] 真假評論評分長條圖

從右上圖中可以看出，假評論的評分(rating)分布呈現兩端高中間低的分布，也就是說假評論的寫手傾向給予極端的分數也就是 1 分或是 5 分，而真實評論的使用者，則傾向給予較高的分數(4 分或是 5 分)，並且較少給到 1 分，由於以上的觀察，所以我們在 features 中新增了 rating_deviation(該篇評論的評分和該間商店平均評分差異的絕對值)以及 extreme_rating_ratio(這篇評論的作者的極端評論(1 分或 5 分)的比例)作為我們其中的兩項變數。

## 用戶與評論間的關聯

我們想要了解每位用戶平均的留言數，從下圖中可以看出用戶平均留 2.34 筆留言，並且有超過 50%的使用者只留下一則留言，由於以上的觀察，所以我們在 features 中新增了 review_count (這篇評論的作者的留言數) 以及 review_count_today (這篇評論的作者今天的評論次數) 作為我們其中的兩項變數。

```
count    260277.000000
mean          2.338270
std           4.496138
min           1.000000
25%           1.000000
50%           1.000000
75%           2.000000
max         197.000000
Name: prodId, dtype: float64
```




[圖十一] 用戶評論數

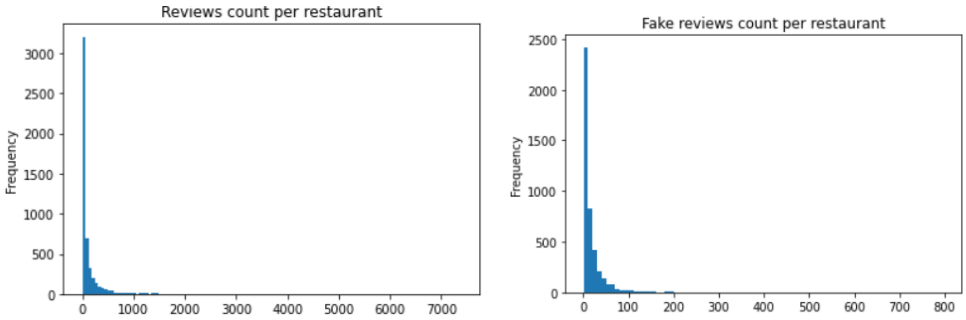
## 餐廳與評論間的關聯

```
count    5044.000000
mean      120.657811
std       300.164451
min         1.000000
25%         8.000000
50%        33.000000
75%       108.000000
max      7378.000000
Name: userId, dtype: float64
```

```
count    4336.000000
mean       18.557657
std        38.630452
min         1.000000
25%         3.000000
50%         9.000000
75%        21.000000
max       797.000000
Name: userId, dtype: float64
```

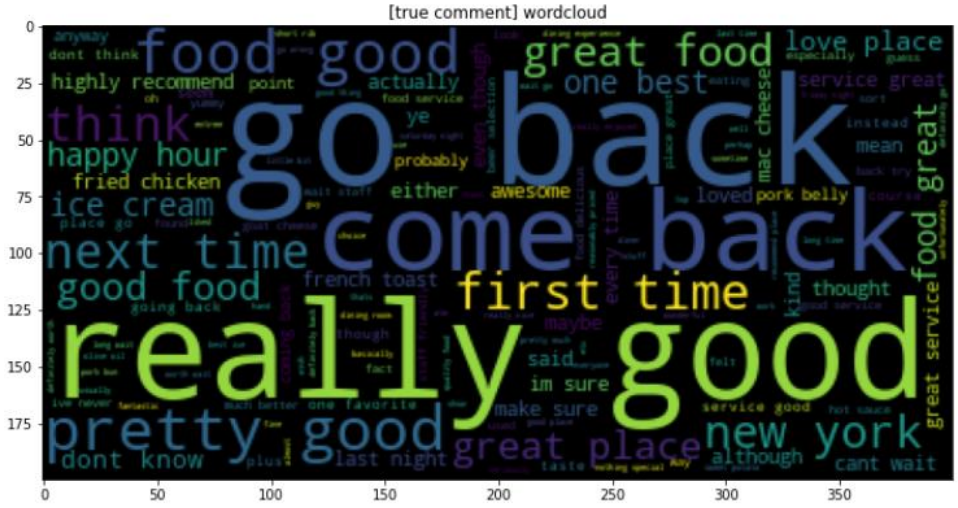
[圖十二] 所有餐廳評論數與假評論數

從上圖中，可以看出每間餐廳平均會有 120 篇評論，其中大約會有 18 筆為假評論，而從下圖的分布中，我們沒有很直接的看出真實評論與假評論的分布是否有差異。

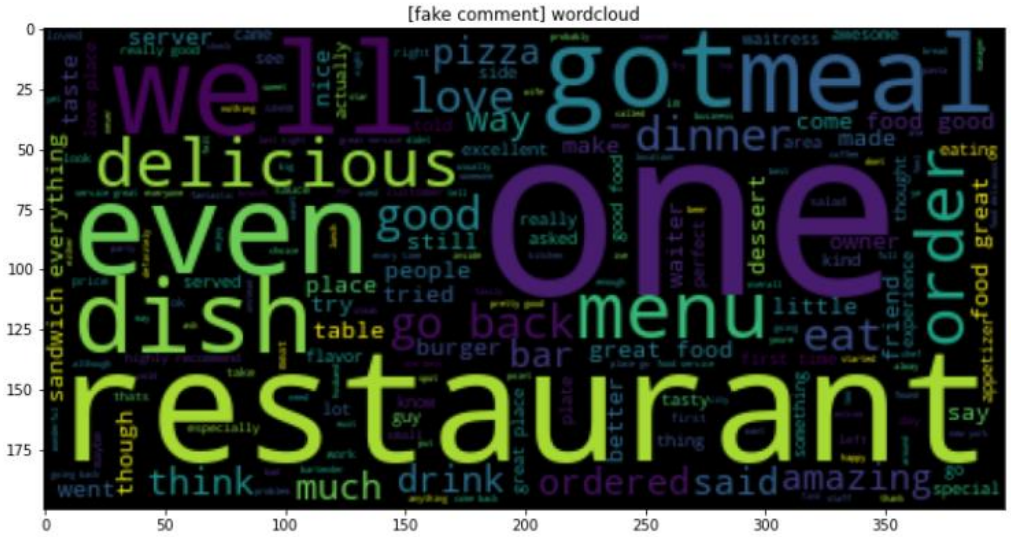



[圖十三] 所有餐廳評論數與假評論數直方圖

## 真評論與假評論文字雲差異




[圖十四] 真實評論文字雲




[圖十五] 假評論文字雲

從上面兩張圖可以分別看出真實評論以及假評論間的用字差異，在真實評論中，大多字是在描述個人的感受，例如說下次會再來、很棒、吃的很開心等等，描述會較為抽象與接近情感的抒發；在假評論中，描述會較為具體，例如說食物很美味，飲料好喝，服務佳等等，這些虛假評論較為客觀且是針對一般人會好奇的項目作評論，所以我們認為虛假評論可能會為了吸引大眾來觀看留言，所以會挑一些人們會感興趣的地方做虛假評論，反而留下真實評論的使用者很多的評論是情感上抒發個人的感受，相較之下較為抽象。

## 4.3 Data preprocessing

### Tokenize review sentences

將每一則評論(如下圖第一行)，依序移除標點符號(下圖第二行)，轉成 tokenize(如下圖第三行)，並且透過 nltk 套件所提供的 stopword 字典移除 stopwords (如下圖第五行)，最後使用 lemmatization 的方法將相同語意的字轉為同一個字(如下圖第六行)。

```
Drinks were bad, the hot chocolate was watered down and the latte had a burnt
taste to it. The food was also poor quality, but the service was the worst pa
rt, their cashier was very rude.

Drinks were bad the hot chocolate was watered down and the latte had a burnt
taste to it The food was also poor quality but the service was the worst part
their cashier was very rude

['Drinks', 'were', 'bad', 'the', 'hot', 'chocolate', 'was', 'watered', 'dow
n', 'and', 'the', 'latte', 'had', 'a', 'burnt', 'taste', 'to', 'it', 'The',
'food', 'was', 'also', 'poor', 'quality', 'but', 'the', 'service', 'was', 'th
e', 'worst', 'part', 'their', 'cashier', 'was', 'very', 'rude']
36

['Drinks', 'were', 'bad', 'the', 'hot', 'chocolate', 'was', 'watered', 'dow
n', 'and', 'the', 'latte', 'had', 'a', 'burnt', 'taste', 'to', 'it', 'The',
'food', 'was', 'also', 'poor', 'quality', 'but', 'the', 'service', 'was', 'th
e', 'worst', 'part', 'their', 'cashier', 'was', 'very', 'rude']
36

['drinks', 'bad', 'hot', 'chocolate', 'watered', 'latte', 'burnt', 'taste',
'food', 'also', 'poor', 'quality', 'service', 'worst', 'part', 'cashier', 'ru
de']
17

['drink', 'bad', 'hot', 'chocolate', 'watered', 'latte', 'burnt', 'taste', 'f
ood', 'also', 'poor', 'quality', 'service', 'worst', 'part', 'cashier', 'rud
e']
17
```

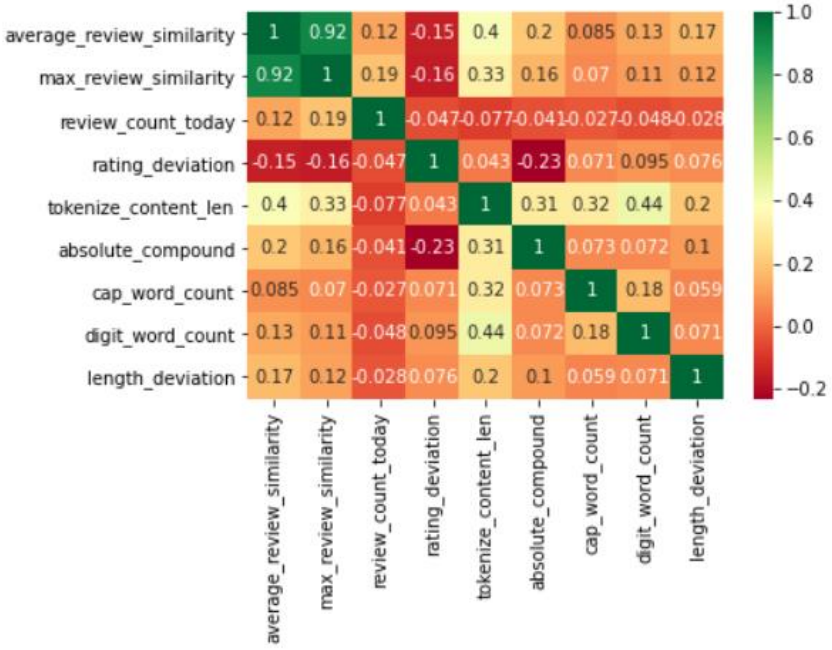
[圖十六] 評論 tokenize 處理

## 建構用戶行為特徵

針對前面所做的 eda 結果，我們建構了 11 個不同的用戶行為特徵。分別為：average_review_similarity 、 max_review_similarity 、 review_count_today 、 rating_deviation、tokenize_content_len、absolute_compound、cap_word_count、digit_word_count、length_deviation、review_count、extreme_rating_ratio，個別特徵所代表的意義會在後面詳加介紹。

## 新特徵的相關係數矩陣

在相關係數矩陣中可以看出，average_review_similarity 和 max_review_similarity 的相關性最高，推測是因為很多的留言都只有一篇或是較少篇，導致在計算時可能無法計算(補為 0)，或是相近，其他的 9 個變數之間的相關係數都不高。




[圖十七] 自建特徵相關係數矩陣

## Class Imbalance

從敘述性統計觀察得知，資料集中真實評論與虛假的評論數量差異懸殊，故本團隊採用 SMOTE 的技巧提升訓練集中虛假評論的數量，以使訓練集中兩類別的評論數量相同。

## 在實際訓練模型前對於各個自建變數的解釋

average_review_similarity：

我們預期如果作者的評論大多相似，則有可能是寫手直接複製貼上相關文章，而不是自己所寫。

max_review_similarity：

我們預期如果作者的評論有文章極度相似，代表可能曾經直接複製貼上自己的留言，我們透過最大值想要看最大的相似度。

review_count_today：

我們預期如果作者在一天內評論了許多間餐廳，則可能是虛假評論。

rating_deviation：

我們預期如果作者的評論的評分和商店平均的評分落差過大，代表作者的行為跟一般大眾行為不相同，所以想特別觀察是否為虛假評論。

tokenize_content_len：

我們預期如果作者是假評論，依照前面 EDA 的結果，評論長度會較真實評論短。

absolute_compound：

我們預期如果是虛假評論，寫手可能會用較為情緒化的字眼吸引大眾的目光，所以我們選用了情緒分析的分數作為特徵，同時因為不確定寫手是為了洗負評或是刷正評，所以我們採用情緒分析分數的絕對值。

cap_word_count：

我們預期如果從評論的大寫字數，可能帶有情感，語調等，可能是使用者所寫的真實評論。

digit_word_count：

我們預期如果從評論的數字數，可能代表價錢、日期等等，可能代表的較為具體、實際的資訊，可能是使用者所寫的真實評論。

length_deviation：

我們預期如果從評論經過前處理(刪除 stopword、標點符號、lemmatization)過後很多字都被刪除了，代表提供了很多沒用的資訊，可能帶有真假評論的相關資訊。

review_count：

我們預期如果從用戶的總評論數來看，如果用戶留過較多言論，只要不是都重複評論，或是在很短的時間內大量留言，可能是真評論居多。

extreme_rating_ratio：

我們預期用戶如果每次評分的行為都較為極端(1 分或 5 分居多)，則可能是虛假評論，因為寫手需要給出較極端的評分來刷分。

## 4.4 Modeling

參考相關文獻之後，最終我們選擇下列三種分類模型：

- 羅吉斯迴歸
- XGBoost
- Gradient Boosting

上述模型皆能使用 Python 的 scikit-learn 套件進行實作。

## 4.5 Experiment

1. Experiment - Behavior：從原始特徵與使用者評論中，做出十一個人工變數，代表使用者行為，詳細內容如前所述。
2. Experiment - Text：15000 個 term，使用 chi-square 特徵選擇法選擇 150 個最重要的 term，來代表這篇評論。
3. Experiment - both ：兩種變數均考慮。

**主要評估指標：False recall、accuracy**

目標是找出所有假評論，因此假評論的召回率(False recall)是我們重視的模型指標。但是單純看假評論的召回率的話，模型只要全部預測為假評論，那 False recall 就是 100%，因此也會檢視該模型是否有一定水準的 accuracy。

附註：所有結果都是有 Oversampling 的

## Experiment 1 - only behavior features

實驗結果：

| | Accuracy | False recall | False precision | False F1 score |
|---|---|---|---|---|
| Logistics regression | 0.68 | 0.80 | 0.27 | 0.40 |
| Gradient Boosting | 0.73 | 0.76 | 0.29 | 0.42 |
| XGBoost | 0.75 | 0.68 | 0.31 | 0.42 |

模型變數重要性：(紅字代表多次出現，**粗體**代表行為變數)

Logistics：**length_deviation**、**digit_word_count**、**max_review_similarity**

Gradient Boosting：**review_count**、**average_review_similarity**、**extreme_rating_ratio**

XGBoost：**review_count**、**extreme_rating_ratio**、**review_count_today**

## 實驗一結論

實驗結果：

False recall 最佳的模型是 Logistics regression，但它的 Accuracy 僅有 0.68，是三種模型中最低的，代表模型傾向預測是假評論，導致整體準確率不佳。XGBoost 則相反，Accuracy 最佳而 False recall 最差，在兩者中取得平衡的話 Gradient Boosting 可能是綜合表現較好的模型，亦即在保證一定水準的準確率下(超過七成)，假評論的召回率還不錯(0.76)。

以實驗結果而言，團隊認為 Gradient Boosting 在只有用戶行為的資料集中，表現得較為穩定而不偏頗。

模型變數重要性：

多次出現的變數有 review_count，代表評論數量有助於判別是否是假評論；extreme_rating_ratio，代表極端評論比例有助於判別是否是假評論，符合我們對該行為變數的假設(即寫手需要為店家刷分或刻意抹黑)。

## Experiment 2 - only text features

| | Accuracy | False recall | False precision | False F1 score |
|---|---|---|---|---|
| Logistics regression | 0.67 | 0.56 | 0.22 | 0.31 |
| Gradient Boosting | 0.69 | 0.54 | 0.22 | 0.31 |
| XGBoost | 0.84 | 0.15 | 0.29 | 0.20 |

模型變數重要性：(紅字代表多次出現，**粗體**代表行為變數)

Logistics：solid、cute、seating、decent、flavorful、space、super、there、liked、sausage

Gradient Boosting：also、pretty、im、little、sauce、come、bit、got、ive、ordered

XGBoost：sauce、there、seating、bit、flavor、pork、cute、solid、tasty

**實驗二結論：**

承兩類別在文字雲的視覺化分析中，我們觀察到假評論傾向給予比較詳細的描述，而真實評論有比較多情緒面的描述。在此階段的實驗結果，從各模型對於變數的重要性之選取中可以發現，各模型所選出的字多是「假評論」對於餐廳客觀描述會使用到的 terms，尤其是對於餐點口味的描述如: flavorful, sauce, pork, tasty 等。但從實驗數據觀察可以發現，相較於上一組實驗，假評論的偵測率顯著下降，而真實評論的偵測率也有微幅下降。造成此現象的發生，本團隊推測，在特徵選取步驟中，因為 data imbalance 的狀況 (真實評論的數量遠高於假評論的數量) ，所使用之 Chi-square method 所選出的強鑑別力特徵，不僅反映出的是假評論的常用字，更應該是真實評論的「不常見字」，意即這些被選出的字為真實評論的「反指標」，故模型在此實驗中，明顯降低了假評論的辨識率，但又因為文字資料有稀疏的特性，使得真實評論的辨識率不增反略減。

實驗結果：

False recall 最佳的模型仍是 Logistics regression，但它的 Accuracy 僅有 0.67，是三種模型中最低的，代表模型傾向預測是假評論，導致整體準確率不佳。XGBoost 則相反，Accuracy 最佳而 False recall 最差，且差異比實驗一更明顯，代表模型幾乎全部預測是真評論，導致不太能偵測出假評論，這與我們的期望相悖，是我們最不樂見的情形。Gradient Boosting 仍然是較平衡的模型，但表現與 Logistics regression 差距不大。

因此，團隊認為 Logistics regression 在只有評論資料的資料集中，表現得較為符合專案目標的期望，亦即較能偵測出假評論(False recall 最高)且保持一定水準的準確率(近七成)。

## Experiment 3 - both text features & behavior features

| | Accuracy | False recall | False precision | False F1 score |
|---|---|---|---|---|
| Logistics regression | 0.71 | 0.75 | 0.28 | 0.41 |
| Gradient Boosting | 0.83 | 0.41 | 0.36 | 0.39 |
| XGBoost | 0.23 | 0.90 | 0.14 | 0.24 |

模型變數重要性：(紅字代表多次出現，**粗體**代表行為變數)

Logistics：**max_review_similarity**、**average_review_similarity**、cute、seating、decent、super、id、youre、space、solid

Gradient Boosting：**review_count**、**max_review_similarity**、**average_review_similarity**、**rating_deviation**、**review_count_today**、**extreme_rating_ratio**、**tokenize_content_len**、**length_deviation**、owner、**absolute_compound**

XGBoost：one、place、**tokenize_content_len**、**rating_deviation**、**max_review_similarity**、**extreme_rating_ratio**、**absolute_compound**、would、**length_deviation**、next

## 實驗三結論：

實驗結果：

False recall 最佳的模型變成 XGBoost，但可以說是犧牲大幅的 Accuracy 換來的，Accuracy 僅 0.23，代表模型幾乎都預測是假評論，導致整體準確率極差。Gradient Boosting 則是另一種極端，Accuracy 最佳而 False recall 最差，代表模型幾乎都預測是真評論，導致不太能偵測出假評論，這與我們的期望相悖，是我們最不樂見的情形。Logistics regression 是較平衡的模型，兩個指標都在七成以上，符合我們的期待。

因此，團隊認為 Logistics regression 在只有評論資料+使用者行為的資料集中，表現得較為符合專案目標的期望，亦即較能偵測出假評論(False recall 七成五)且保持一定水準的準確率(七成左右)。

模型變數重要性：

大部分重要的變數都是行為變數，代表與文字相比，行為變數能給予更多資訊，有助於偵測出假評論。且 max_review_similarity、average_review_similarity、rating_deviation、extreme_rating_ratio、tokenize_content_len、length_deviation、absolute_compound 等行為變數在多個模型中都是前幾重要的，代表本團隊對於這些變數影響真假評論的差異的假設頗合理，的確是偵測假評論的重要變因。

## 三個實驗的最適模型比較

| | Accuracy | False recall | False precision | False F1 score |
|---|---|---|---|---|
| Gradient Boosting (behavior) | 0.73 | 0.76 | 0.29 | 0.42 |
| Logistics regression (text) | 0.67 | 0.56 | 0.22 | 0.31 |
| Logistics regression (behavior+text) | 0.71 | 0.75 | 0.28 | 0.41 |

### 比較結論

只用行為變數的 Gradient Boosting 模型在所有指標都略優，而單純用文字資料的模型效果最差，兩者都採用的模型效果比只用行為變數的模型略遜，但其實差距不大。因此本團隊認為行為變數對於偵測假評論是很重要的，而文字資料可能需要更多不同的處理方法，方能達到更好成效。

### 實驗總結論

1. Accuracy 和 False recall 勢必要作出取捨，模型常常會傾向都預測是真評論(或相反，都預測假評論)，很難同時讓他們穩定上升。
2. 行為變數對於偵測假評論非常重要，可以從評論的特性(評論長短、評分差異、評論的情緒強烈程度)、作者的特性(作者過往評論之間的相似度、當天發布的評論次數)得知該篇評論為假評論的可能性。
3. 文字(評論本身)變數對於偵測假評論助益不大。

# 五、管理意涵及學術貢獻

從實驗結果分析，得知評論者的行為變數相較文字變數對於假評論的辨識更具效果。換句話說，對於評論平台的業者而言，若要抓出假評論，可以從評論者帳號的歷史行為記錄去做判斷，例如以本研究所建立的幾個變數作為判斷；相反的，不論是否為評論寫手，每個人的寫作風格、用字遣詞可能有極高的變異，從評論內容反而難以辨識假評論。

唯從評論內容的大方向來分析，根據文字雲的呈現，真實評論中，大多字是在描述個人的感受，描述會較為抽象與接近情感的抒發；在假評論中，描述會較為具體且針對一般人會好奇的項目作評論。故我們覺得現行多數評論平台的社群作用，也就是提供用者抒發心情，發表意見的作用可能大於實際上資訊交換的功能，反而較多具體客觀的評論可能為虛假評論，畢竟如此更容易吸引讀者的注意。

# 六、結論及未來展望

在本研究中，首先本團隊透過資料的探索性分析，找出真實評論與虛假評論的差異之處，並以此作為衍生變數，納入第二階段的預測模型之中。經過模型的比較我們發現，在進行真實、虛假的評論辨識時，評論者的評論行為相較於其所寫的評論內容更具參考性。

本研究最終所得到的結果與我們一開始的假設與期望大相徑庭，本團隊發現假評論更可能以看似客觀的描述來評價餐廳，藉此博得關注，但內容的真實性我們卻無從驗證。經驗感受的事情本來就因人而異，使用者的留言習慣也不盡相同，唯有切身體驗、親自去感受才是最準確的，在參考評論時但勿盡信之。

未來若本研究在辨識率的精準度上有所突破，以及若能搜集更多不同平台的評論資料，期許得以將建構好的假評論分類模型用於其他平台之上（Google Play, Google Map ...），如果假評論分類模型在其他平台仍然能夠有良好的預測能力，那可以代表本研究所建構的假評論分類模型在多個不同的領域如遊戲、電影等都能夠有準確的預測能力，模型的應用層面將會更為廣泛。

# 附錄

## 參考文獻


- Sihombing and A. C. M. Fong, "Fake Review Detection on Yelp Dataset Using Classification Techniques in Machine Learning," 2019 International Conference on contemporary Computing and Informatics (IC3I), Singapore, Singapore, 2019, pp. 64-68, doi: 10.1109/IC3I46837.2019.9055644.
- 鄭麗珍, 江彥孟, & 游政憲. (2019). 應用深度學習技術於網路虛假評論偵測. 電子商務學報, 21 卷(2 期), 229 – 252. doi:10.6188/JEB.201912_21(2).0004