AirBnb 評論分析

111-1文字探勘初論期末專案第22組

郭旭崴	馬松鐸	陳郁婷	闕蕎蓁
會計三	經濟三	經濟二	經濟二
國立台灣大學	國立台灣大學	國立台灣大學	國立台灣大學
B09702036	B09303131	B10303105	B10303125

一、研究動機

後疫情時代，各國門戶大開，民眾進行報復性旅遊，因此旅遊活動盛行。而作為住宿選擇之一的Airbnb，各房主需要吸引消費者的投宿來創造收入。因此我們希望能運用上課所學，來為房主行銷及改善自己的房子，增加消費者對Airbnb房子的滿意度。

二、專案目標

分析芝加哥Airbnb客戶評論，利用模型將評論根據正負面來區分，再各別對正負面評論進行分群，藉由關鍵字來找出屋主容易有的問題，以及大家最喜歡的點。經由上述資料來為房主分析該如何針對偏好行銷自己的房子，或該針對什麼部分進行改善有助於提升消費者的好感度。

三、程式實作

1. 模型建立

首先，為了要建立一個模型，需要在網路上蒐集IMDB的資料集，裡面有涵蓋各式各樣的電影評論以及被標上的正負面標籤。

藉由這個資料集，我們先做了第一步的處理，即tokenize化，同時，我們也把之中的stopwords還有標點符號和換行符號等無意義的詞彙給刪除來進一步加深模型的精準度，而在做完這些動作以後，我們會得到一堆tokens的資料。

藉由上述產生的資料，我們可以創造出屬於自己模型的字典。這個字典是依照用字頻率，也就是各組字所出現的次數，由高到低排列。如圖一所示：

	words	frequency
0	I	163294
1	’s	121768
2	The	87819
3	movie	83813
4	film	75940
...	...	...
9995	cyborg	65
9996	Lara	65
9997	consisted	65
9998	versa	65
9999	aptly	65

圖一：用字頻率所建立的字典

接著，我們藉由網路上google所釋出的word2vec的檔案，以及上述所創造的字典，製造出我們模型第一層的架構(10000 x300)。

在做完上述處理後，便可以開始架設模型。我們從keras.models函式庫裡面引進Sequential作為我們建模型的依據，建完的模型如下圖二。

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 100, 300)          3000000

flatten_1 (Flatten)          (None, 30000)             0

dense_2 (Dense)              (None, 16)                480016

dense_3 (Dense)              (None, 1)                 17

=================================================================
Total params: 3,480,033
Trainable params: 480,033
Non-trainable params: 3,000,000
```

圖二：模型概述

在建立好模型以後，我們需要回頭去處理訓練資料和測試資料，使資料變成我們模型所需要輸入的模式。

依據我們所創造的字典，我們有對各個常用字進行編號，而我們從IMDB取得的評論共有50000條，我們針對這50000條評論分別去對應字典，看這50000條評論分別用到字典裏面的哪幾號字。

在處理完這些詞彙以後，由於我們規定模型最多吃100個字，因此我們又針對這50000條已經變成使用哪幾號字的句子進行處理，超過100維度只取前100維度，不足的補0，讓句子固定為100維度。如附圖三所示：

```
[  45  307   51  266    2 1268 2994 1531  297  510  194 1831  328  316
   2   74  464   26  389 4524    7   94    5 2157  104  270   16  706
1730 1595 7049   26  189   73  349 2334  316   45 4462  266    8   25
1561    1  132   54    2 1831   15  171  268   51   99 1033 3389  161
 245 2072 3133 1170 1061 5089   10  196 1736 4884  480   63 3688  609
 480  999    1   95 2371 1853   26  133    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0]
```

圖三：其中一條句子之樣式，分別代表用到字典第45號、第307號、第51號……等字的句子

在處理完前述步驟以後，我們把這50000個句子連同其標籤做切割，切割出訓練資料和測試資料，各有25000筆資料，接著我們用訓練的25000筆資料來訓練模型。

在訓練模型時，我們設定模型每過256個資料即要更改一次參數，同時要求奇每個句子都要跑4次，前者為了讓模型更快收斂，後者是為了避免模型過度配飾。如附圖四所示：

```
history = model.fit(train_data, train_labels, epochs=4, batch_size = 256)

Epoch 1/4
98/98 [==============================] - 3s 23ms/step - loss: 0.6063 - acc: 0.6817
Epoch 2/4
98/98 [==============================] - 2s 21ms/step - loss: 0.4874 - acc: 0.7772
Epoch 3/4
98/98 [==============================] - 2s 22ms/step - loss: 0.4179 - acc: 0.8162
Epoch 4/4
98/98 [==============================] - 2s 22ms/step - loss: 0.3544 - acc: 0.8514
```

圖四：模型訓練

而訓練完模型以後，我們把測試資料帶入，確認其精準度。我們得到其精準度約為七十幾趴。如附圖五所示：

```
testing_result = model.evaluate(test_data, test_labels)

782/782 [==============================] - 2s 3ms/step - loss: 0.4687 - acc: 0.7782
```

圖五：測試模型精準度

確認完模型尚可提供一定精準度以後，我們便可以開始處理Airbnb的資料了。

2. 資料帶入

我們一開始先將Inside Airbnb網站的評論資料取出，這次模型特別針對來自Chicago, Illinois, United States的Airbnb的評論，總共約329,026筆資料來當作這次帶入這次模型的資料。由於這些資料是來自世界各地使用者的評論，因此會混雜許多非英語的語言。因此，我們先將不是由英文撰寫的評論去除。

接著，將剩下的英文評論切成單字，並且去掉stop_words、標點符號及
換行符號後存成串列。再來，便可以將剛剛的串列對照一開始創建的字典存成word ID sequence。

接著，利用keras內建的pad_sequence，將句子填充到相同長度，便可將資料放入模型中。如附圖六所示：

```
predict_output = model.predict(airbnb_data)
print(predict_output)
```

```
10283/10283 [==============================] - 24s 2ms/step
[[0.85712296]
 [0.8330933 ]
 [0.42473334]
 ...
 [0.70125806]
 [0.19987772]
 [0.5433807 ]]
```

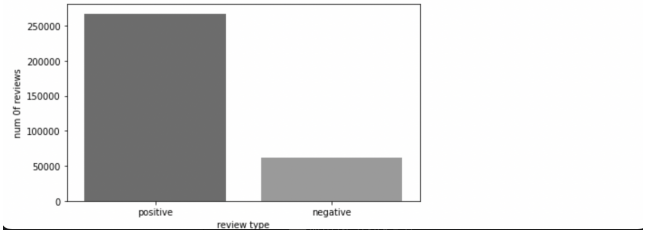
圖六：Airbnb評論資料帶入模型後的結果

3. 模型結果

我們利用0.5當作boundary，將被模型劃分為0.5以上的評論歸為正面評論，而0.5以下的評論歸為負面評論，如附圖七的圖片可看出，正面的評論遠超過負面的評論，正面評論的數量為267,712筆，而負面評論的數量為61,914筆。

```
import seaborn as sns
import matplotlib.pyplot as plt
x = ['positive', 'negative']
y = [len(positive_review), len(negative_review)]
fig, (ax1) = plt.subplots(1, 1, figsize=(7, 4))
sns.barplot(x, y)
ax1.set(xlabel='review type', ylabel='num 0f reviews')
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pas
  warnings.warn(
[Text(0, 0.5, 'num 0f reviews'), Text(0.5, 0, 'review type')]
```



圖七：正面評論及負面評論的數量比較

4. 結果分群

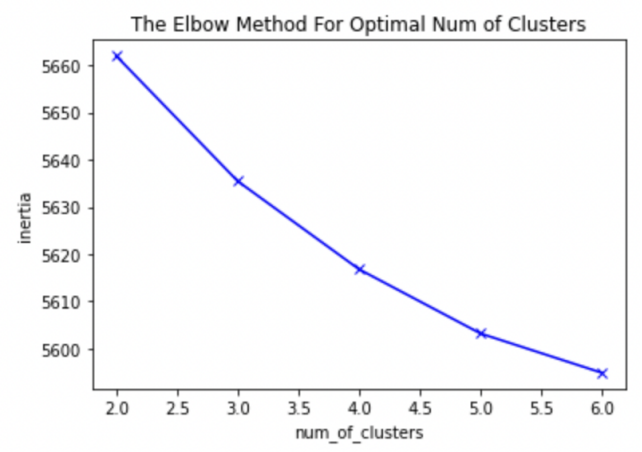
我們將利用K-means Clustering將上述模型所區分出來的正負面評論，分別作進一步的分群。首先，把正負面評論都先轉成TFIDF vectors，如圖八所示：

```
from sklearn.feature_extraction.text import TfidfVectorizer
TFIDF_vectorizer = TfidfVectorizer(min_df = 2, stop_words = 'english')
TFIDF_vectors = TFIDF_vectorizer.fit_transform(positive_review)
print(TFIDF_vectors[0])
```

```
  (0, 5968)     0.22625685199696616
  (0, 1580)     0.30899165383179017
  (0, 6038)     0.29981305742012004
  (0, 2964)     0.24463969399370206
  (0, 15203)    0.3151314061577685
  (0, 11080)    0.22115541331277538
  (0, 9678)     0.12580501051080203
  (0, 15631)    0.1661298570648135
  (0, 4426)     0.13178597823270366
  (0, 2369)     0.19113262420953117
  (0, 17874)    0.275029460358818
  (0, 15601)    0.34607136438266295
  (0, 2545)     0.23468859720181903
  (0, 2374)     0.23642147670066668
  (0, 5970)     0.21911090174317593
  (0, 7427)     0.18573726191033862
  (0, 20239)    0.20315452999578737
  (0, 21624)    0.16701091268539608
```

圖八：建立正面評論之TFIDF vectors

接著，利用The Elbow Method 選定適當之分群數目，如圖九所示。將選定數值帶入後即可得到分群結果如圖十與圖十一。



圖九：The Elbow Method決定適當分群數

Cluster 0 :	Cluster 1 :	Cluster 2 :	Cluster 3 :	Cluster 4 :
nice	place	great	great	br
place	stay	place	location	great
stay	clean	stay	place	place
clean	great	location	stay	stay
great	chicago	host	clean	chicago
location	comfortable	clean	host	location
really	perfect	recommend	easy	clean
host	apartment	definitely	space	apartment
neighborhood	host	communication	communication	host
good	location	hosts	definitely	nice
apartment	recommend	chicago	recommend	comfortable
comfortable	definitely	space	apartment	easy
quiet	home	value	good	room
close	easy	comfortable	comfortable	really
easy	good	neighborhood	perfect	home

圖十：正面分群結果（每群前15筆資料）

Cluster 0 :	Cluster 1 :	Cluster 2 :	Cluster 3 :	Cluster 4 :
room	br	place	place	parking
clean	place	clean	host	street
bathroom	apartment	good	stay	place
place	stay	stay	apartment	clean
bed	host	location	unit	free
stay	room	great	airbnb	great
house	just	easy	like	easy
good	airbnb	apartment	dirty	spot
location	clean	train	check	location
like	night	close	did	stay
night	bathroom	need	door	apartment
living	check	walk	clean	good
just	unit	little	bathroom	park
rooms	like	chicago	br	car
people	good	line	bed	garage

圖十一：負面分群結果（每群前15筆資料）

四、結果分析

初步觀察分群結果，雖然並沒有非常鮮明的主題性，與一開始所預想的結果有點落差，某些單字如great，place，etc.重複出現在數群資料當中，顯示出這些關鍵字應該在大多數評論都會被提及，較不具有代表性，但可以發現某些特定單字如neighborhood，communication，etc.只在特定集群中出現，仍然能看出每群有些微故事性。

接著，為了確認群跟群之間的關係以及正負面分類的準確度，挑選部分單字回到正負面評論list中尋找相對應的評論。

就正面評論而言，可以發現消費者較在意的點有：風景、整潔、房主接待、交通、地理位置等要素，因此若房子本身具備上述優點，我們建議房主可以多加行銷。

而就負面評論來說，模型其實無法非常精準地將負面評論區分出來，負面評論中參雜到部分正面評論，通常這種情況發生在具有轉折語氣的評論，但是仍然可以看出消費者在意的缺點可分成：有無衛浴用品、設備隔音程度、有無停車位或停車場距離等，因此我們建議房主能夠多加注意或改善這些部分。

五、參考項目

data form review.csv in Chicago, Illinois, United States

URL:http://insideairbnb.com/get-the-data