作業概述：

針對假新聞做檢測，分辨一篇文章是否是可以信任的。其中分別利用 GBDT、LightGBM、xgboost 等三種演算法對 train.csv 進行建模，並將 test.csv 帶入訓練好的模型，跑出之預測結果與 sample_submission 裡已有的真正的 label 進行比較計算 Accuracy、Precision、Recall、F-measure 等等。

結果與討論：

文字探勘前處理的辦法：

首先一開始發現其中一筆資料的 label 有問題，將那筆資料剔除，然後將 train.csv 裡所有文本都放入一個 list 裡面，接著透過 sklearn.feature_extraction.text 裡的 CountVectorizer 套件，先將內建預設的 stopwords 從文本中剔除掉，接著將 token 出現在至少 2 篇文本且出現在不超過一半的文本當成 term 建立 term frequency 然後透過 TfidfTransformer 套件，利用 tf-idf 調整各個 term 在不同文本的權重，最後由於跑出來的 term 仍然有 30000 多個，利用 pca 進行降維，降維之後的結果搭配 label 值進行建模。

GBDT、LightGBM、xgboost 模型之結果比較：(在測試集上)

	LightGBM	GBDT	xgboost
Accuracy	0.4916	0.4836	0.4916
Precision	0.4816	0.4708	0.48
Recall	0.3598	0.3533	0.3306
F-measure	0.4119	0.4037	0.3916

其中有觀察到在建模後，將訓練集帶入訓練好的模型評估，準確率都達到了極高的水準，但在測試集上的表現卻不盡理想，甚至出現準確率沒有超過直接將全部之值猜測較多的一類這種最原始的方法。分析其背後原因，原因可能有以下幾點：

1.有可能是因為訓練集與測試集本身的文本相似度就不高，導致即使在怎麼樣的訓練，也不可能測試集上有著很好的表現。

2.也有可能是由於模型的參數過多，即使是透過 pca 降維以及透過模型本身中的隨機選擇 k 個參數進行建模，維度過多導致模型過於複雜加上訓練的樣本不夠多，導致過擬合(增加樣本可以緩解此一問題)。

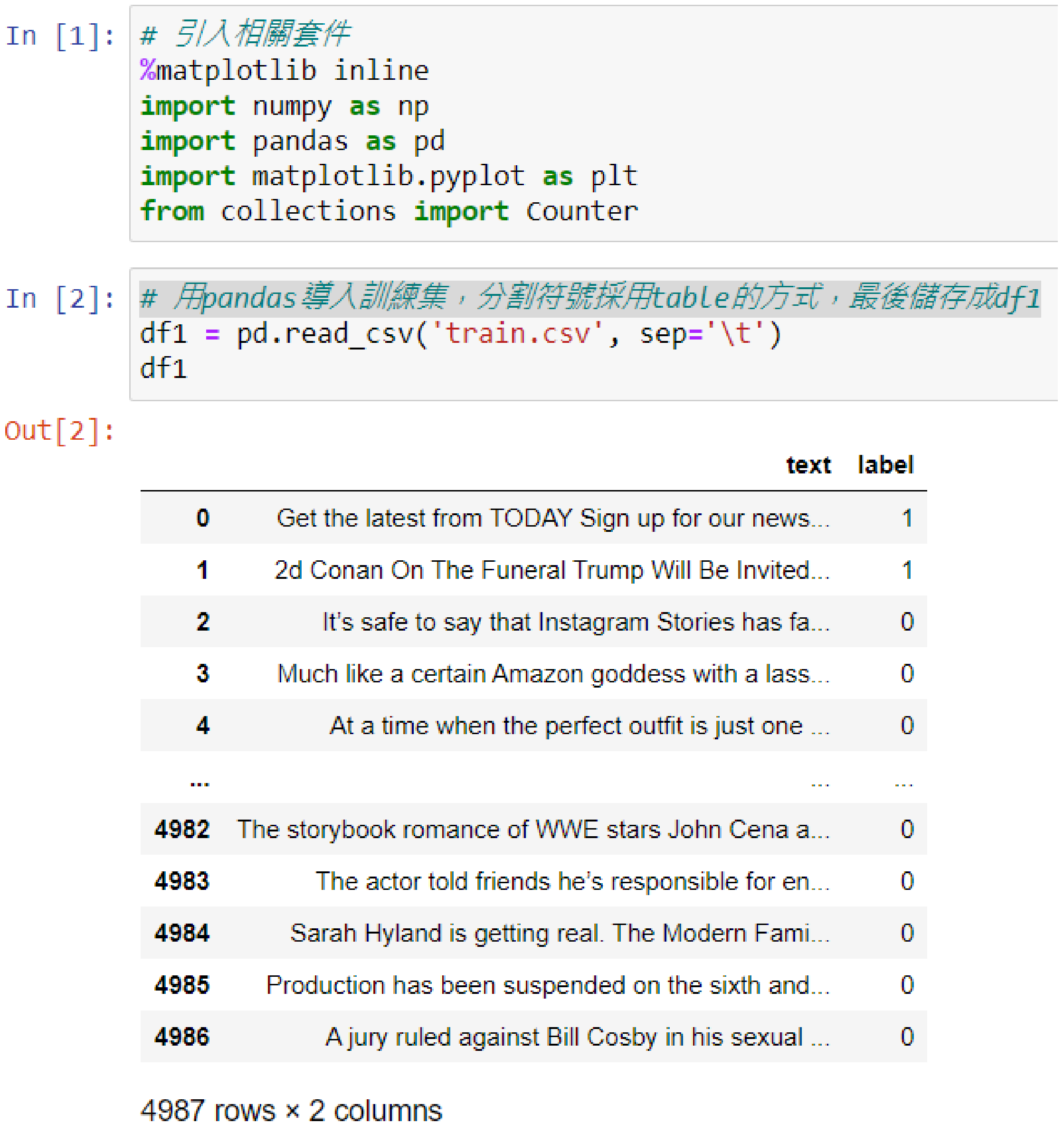
3.雜訊太多，觀察原始文本發現資料集不算非常的乾淨，有著許多表情符號、數字以及由英文字母組成的奇怪單字，導致建模被這些數據大幅影響其表現。

結論：

1.這次的三個演算法都很重要，到現在仍不是完全熟悉其背後的原理，因此會加緊時間多爬些資料熟悉它們。

2.這次的作業與下一次的作業都是針對假新聞作分析，或許可以嘗試透過自創 stopwords 的方式以更好的貼近現實，這也是我從這次作業中得到的心得。

程式碼：



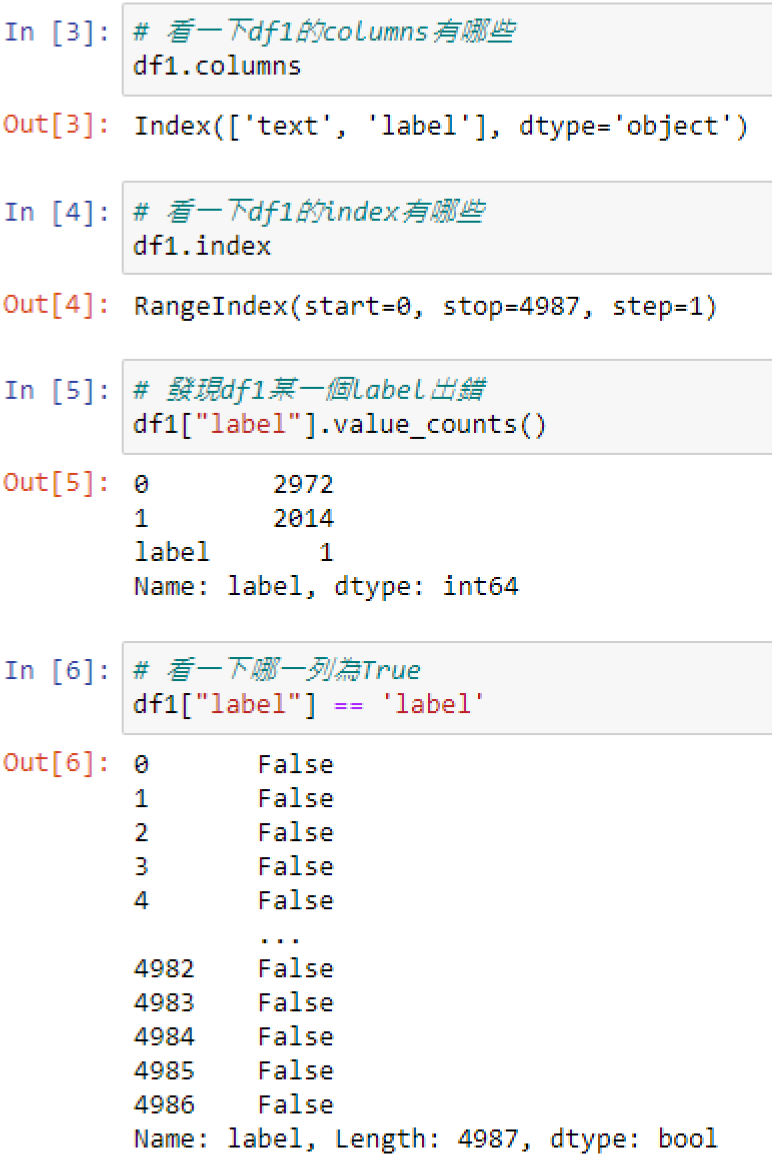
```
In [1]: # 引入相關套件
        %matplotlib inline
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        from collections import Counter
```

```
In [2]: # 用pandas導入訓練集，分割符號採用table的方式，最後儲存成df1
        df1 = pd.read_csv('train.csv', sep='\t')
        df1
```

Out[2]:

	text	label
0	Get the latest from TODAY Sign up for our news...	1
1	2d Conan On The Funeral Trump Will Be Invited...	1
2	It's safe to say that Instagram Stories has fa...	0
3	Much like a certain Amazon goddess with a lass...	0
4	At a time when the perfect outfit is just one ...	0
...	...	...
4982	The storybook romance of WWE stars John Cena a...	0
4983	The actor told friends he's responsible for en...	0
4984	Sarah Hyland is getting real. The Modern Fami...	0
4985	Production has been suspended on the sixth and...	0
4986	A jury ruled against Bill Cosby in his sexual ...	0

4987 rows × 2 columns

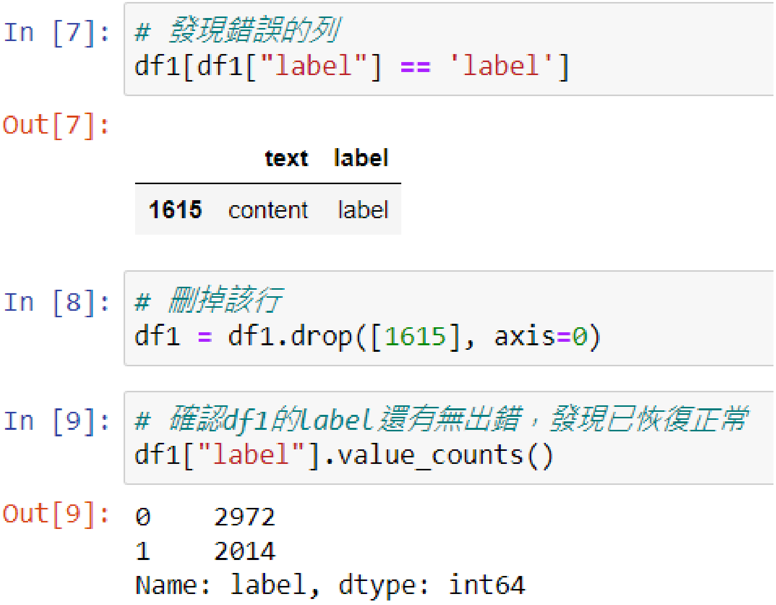


```
In [3]: # 看一下df1的columns有哪些
        df1.columns
Out[3]: Index(['text', 'label'], dtype='object')
```

```
In [4]: # 看一下df1的index有哪些
        df1.index
Out[4]: RangeIndex(start=0, stop=4987, step=1)
```

```
In [5]: # 發現df1某一個label出錯
        df1["label"].value_counts()
Out[5]: 0        2972
        1        2014
        label       1
        Name: label, dtype: int64
```

```
In [6]: # 看一下哪一列為True
        df1["label"] == 'label'
Out[6]: 0       False
        1       False
        2       False
        3       False
        4       False
                ...
        4982    False
        4983    False
        4984    False
        4985    False
        4986    False
        Name: label, Length: 4987, dtype: bool
```

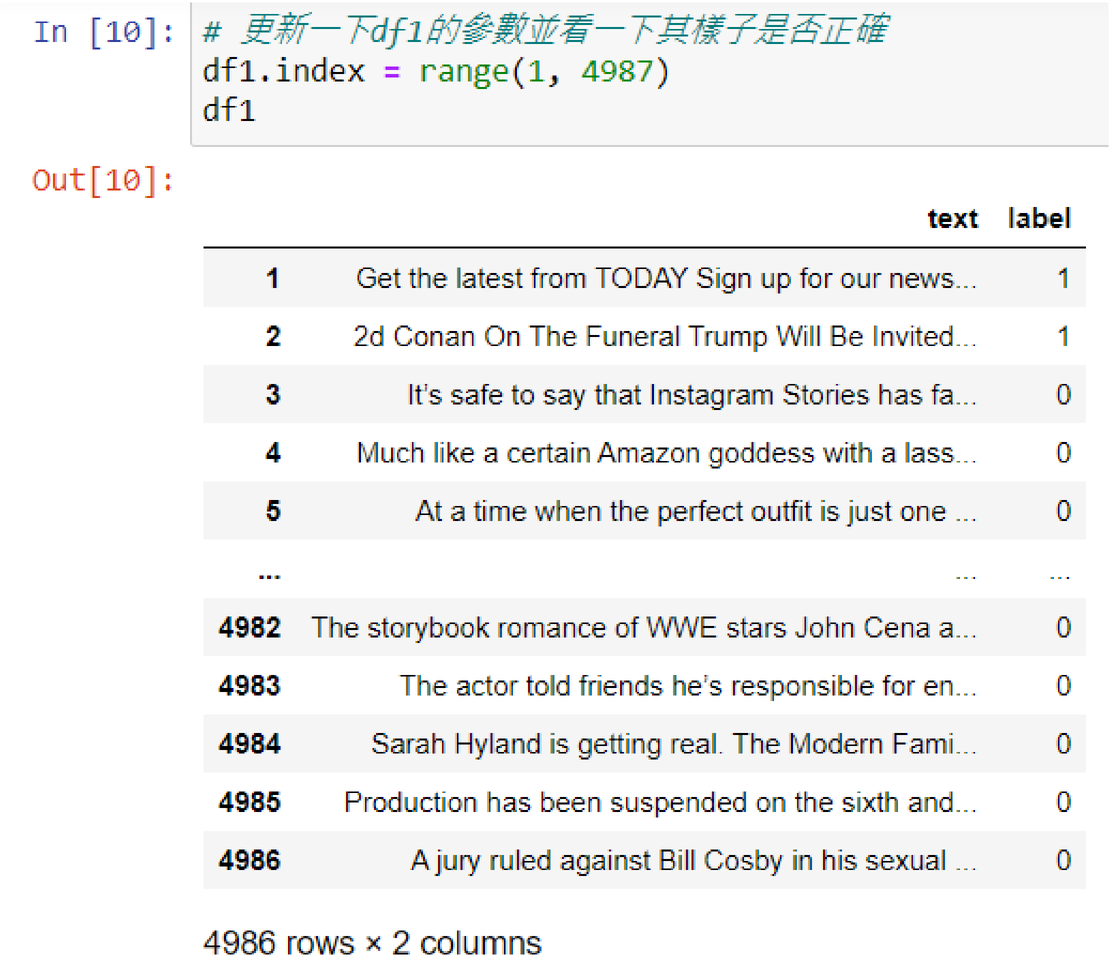


```
In [7]: # 發現錯誤的列
        df1[df1["label"] == 'label']
Out[7]:
```

	text	label
1615	content	label

```
In [8]: # 刪掉該行
        df1 = df1.drop([1615], axis=0)
```

```
In [9]: # 確認df1的label還有無出錯，發現已恢復正常
        df1["label"].value_counts()
Out[9]: 0    2972
        1    2014
        Name: label, dtype: int64
```



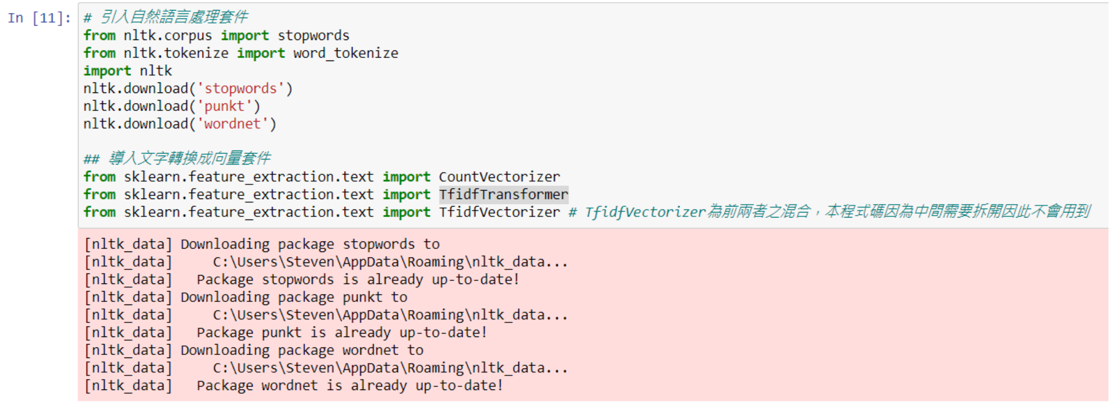
In [10]:

```python
# 更新一下df1的參數並看一下其樣子是否正確
df1.index = range(1, 4987)
df1
```

Out[10]:

	text	label
1	Get the latest from TODAY Sign up for our news...	1
2	2d Conan On The Funeral Trump Will Be Invited...	1
3	It's safe to say that Instagram Stories has fa...	0
4	Much like a certain Amazon goddess with a lass...	0
5	At a time when the perfect outfit is just one ...	0
...	...	...
4982	The storybook romance of WWE stars John Cena a...	0
4983	The actor told friends he's responsible for en...	0
4984	Sarah Hyland is getting real. The Modern Fami...	0
4985	Production has been suspended on the sixth and...	0
4986	A jury ruled against Bill Cosby in his sexual ...	0

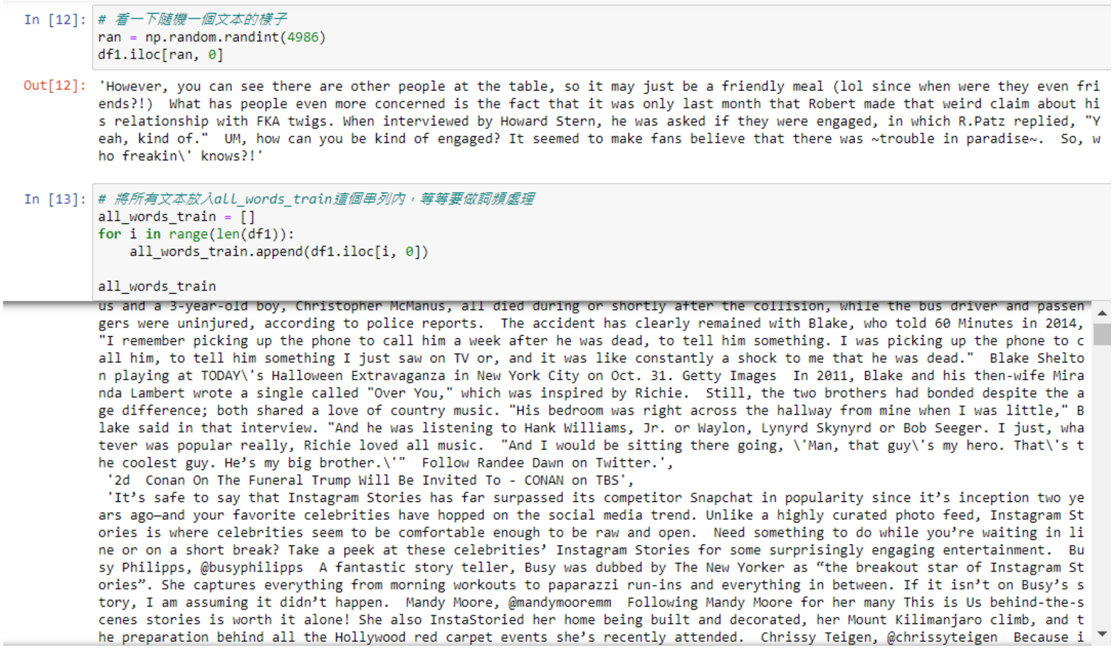
4986 rows × 2 columns



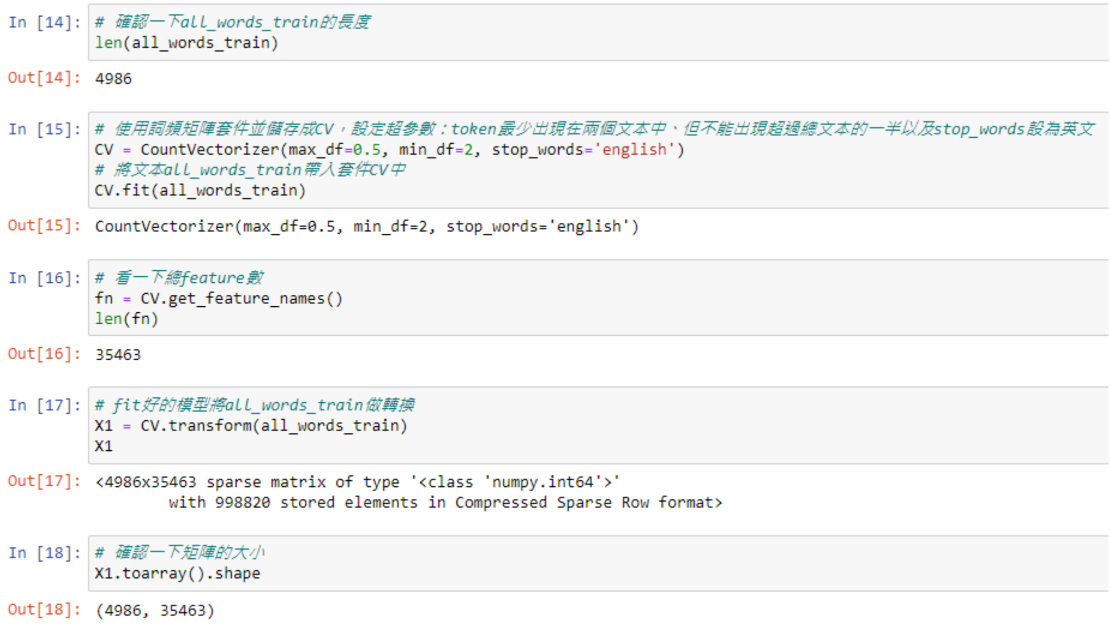
In [11]:

```python
# 引入自然語言處理套件
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import nltk
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')

## 導入文字轉換成向量套件
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import TfidfVectorizer # TfidfVectorizer為前兩者之混合，本程式碼因為中間需要拆開因此不會用到
```

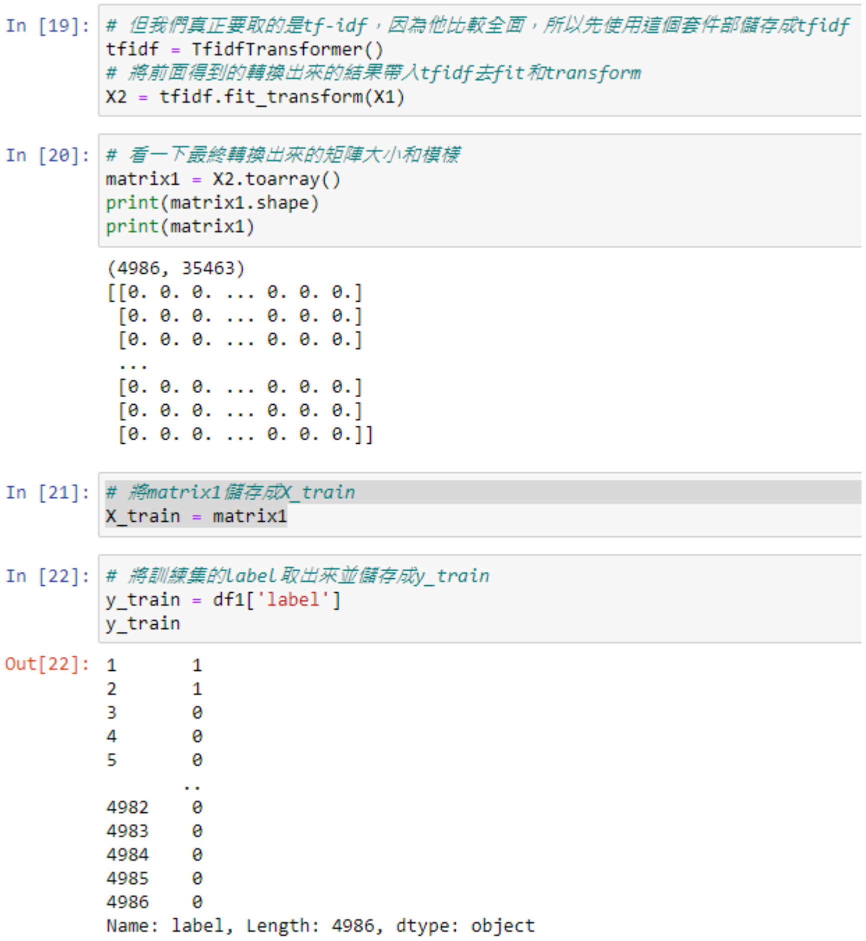
```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Steven\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Steven\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\Steven\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```



```
In [12]: # 看一下隨機一個文本的樣子
         ran = np.random.randint(4986)
         df1.iloc[ran, 0]

Out[12]: 'However, you can see there are other people at the table, so it may just be a friendly meal (lol since when were they even fri
         ends?!)  What has people even more concerned is the fact that it was only last month that Robert made that weird claim about hi
         s relationship with FKA twigs. When interviewed by Howard Stern, he was asked if they were engaged, in which R.Patz replied, "Y
         eah, kind of."  UM, how can you be kind of engaged? It seemed to make fans believe that there was ~trouble in paradise~.  So, w
         ho freakin\' knows?!'

In [13]: # 將所有文本放入all_words_train這個串列內，等等要做詞頻處理
         all_words_train = []
         for i in range(len(df1)):
             all_words_train.append(df1.iloc[i, 0])

         all_words_train
         us and a 3-year-old boy, Christopher McManus, all died during or shortly after the collision, while the bus driver and passen
         gers were uninjured, according to police reports.  The accident has clearly remained with Blake, who told 60 Minutes in 2014,
         "I remember picking up the phone to call him a week after he was dead, to tell him something. I was picking up the phone to c
         all him, to tell him something I just saw on TV or, and it was like constantly a shock to me that he was dead."  Blake Shelto
         n playing at TODAY\'s Halloween Extravaganza in New York City on Oct. 31. Getty Images  In 2011, Blake and his then-wife Mira
         nda Lambert wrote a single called "Over You," which was inspired by Richie.  Still, the two brothers had bonded despite the a
         ge difference; both shared a love of country music. "His bedroom was right across the hallway from mine when I was little," B
         lake said in that interview. "And he was listening to Hank Williams, Jr. or Waylon, Lynyrd Skynyrd or Bob Seeger. I just, wha
         tever was popular really, Richie loved all music.  "And I would be sitting there going, \'Man, that guy\'s my hero. That\'s t
         he coolest guy. He's my big brother.\'"  Follow Randee Dawn on Twitter.',
          '2d  Conan On The Funeral Trump Will Be Invited To - CONAN on TBS',
          'It's safe to say that Instagram Stories has far surpassed its competitor Snapchat in popularity since it's inception two ye
         ars ago—and your favorite celebrities have hopped on the social media trend. Unlike a highly curated photo feed, Instagram St
         ories is where celebrities seem to be comfortable enough to be raw and open.  Need something to do while you're waiting in li
         ne or on a short break? Take a peek at these celebrities' Instagram Stories for some surprisingly engaging entertainment.  Bu
         sy Philipps, @busyphilipps  A fantastic story teller, Busy was dubbed by The New Yorker as "the breakout star of Instagram St
         ories". She captures everything from morning workouts to paparazzi run-ins and everything in between. If it isn't on Busy's s
         tory, I am assuming it didn't happen.  Mandy Moore, @mandymooremm  Following Mandy Moore for her many This is Us behind-the-s
         cenes stories is worth it alone! She also InstaStoried her home being built and decorated, her Mount Kilimanjaro climb, and t
         he preparation behind all the Hollywood red carpet events she's recently attended.  Chrissy Teigen, @chrissyteigen  Because i
```

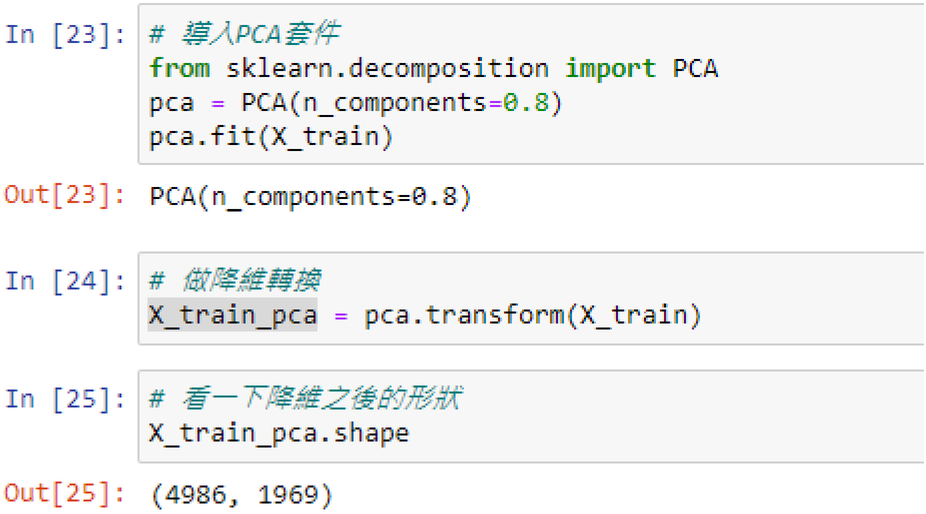


```
In [14]: # 確認一下all_words_train的長度
         len(all_words_train)

Out[14]: 4986

In [15]: # 使用詞頻矩陣套件並儲存成CV，設定超參數：token最少出現在兩個文本中、但不能出現超過總文本的一半以及stop_words設為英文
         CV = CountVectorizer(max_df=0.5, min_df=2, stop_words='english')
         # 將文本all_words_train帶入套件CV中
         CV.fit(all_words_train)

Out[15]: CountVectorizer(max_df=0.5, min_df=2, stop_words='english')

In [16]: # 看一下總feature數
         fn = CV.get_feature_names()
         len(fn)

Out[16]: 35463

In [17]: # fit好的模型將all_words_train做轉換
         X1 = CV.transform(all_words_train)
         X1

Out[17]: <4986x35463 sparse matrix of type '<class 'numpy.int64'>'
                 with 998820 stored elements in Compressed Sparse Row format>

In [18]: # 確認一下矩陣的大小
         X1.toarray().shape

Out[18]: (4986, 35463)
```



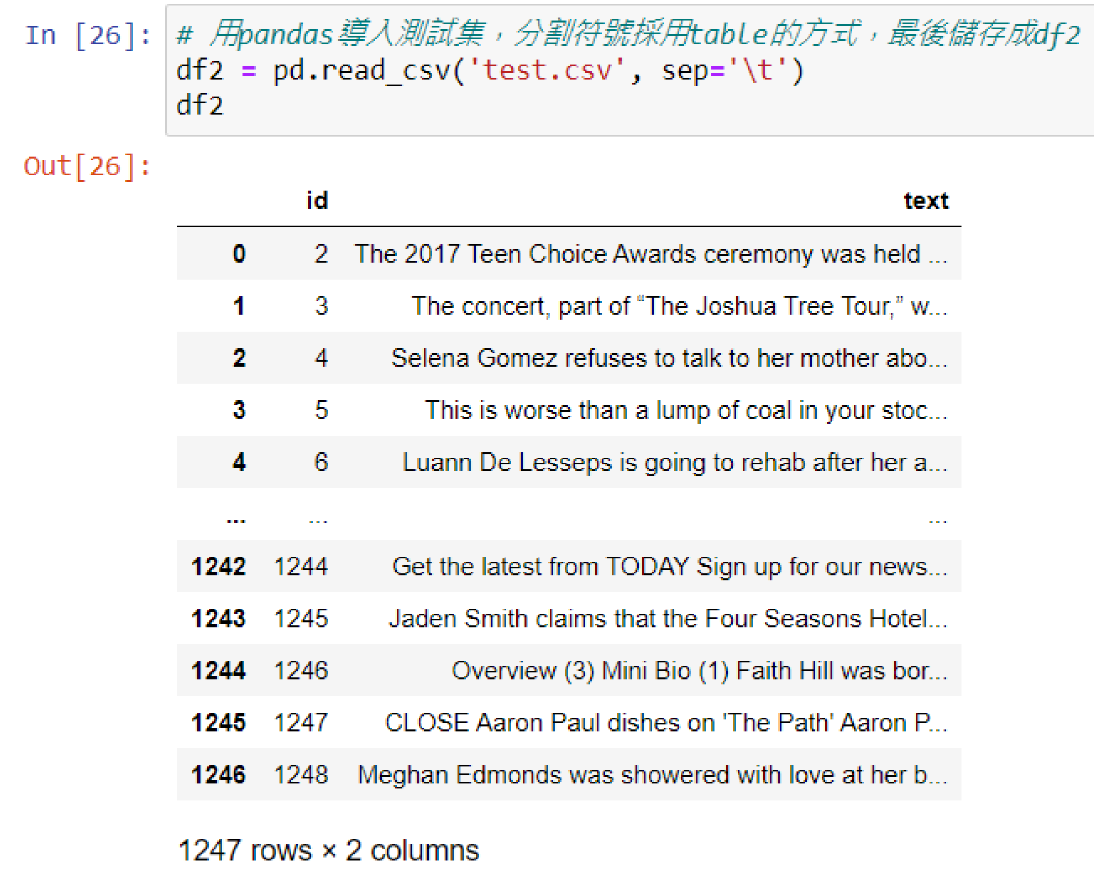
```
In [19]: # 但我們真正要取的是tf-idf，因為他比較全面，所以先使用這個套件部儲存成tfidf
         tfidf = TfidfTransformer()
         # 將前面得到的轉換出來的結果帶入tfidf去fit和transform
         X2 = tfidf.fit_transform(X1)

In [20]: # 看一下最終轉換出來的矩陣大小和模樣
         matrix1 = X2.toarray()
         print(matrix1.shape)
         print(matrix1)

         (4986, 35463)
         [[0. 0. 0. ... 0. 0. 0.]
          [0. 0. 0. ... 0. 0. 0.]
          [0. 0. 0. ... 0. 0. 0.]
          ...
          [0. 0. 0. ... 0. 0. 0.]
          [0. 0. 0. ... 0. 0. 0.]
          [0. 0. 0. ... 0. 0. 0.]]

In [21]: # 將matrix1儲存成X_train
         X_train = matrix1

In [22]: # 將訓練集的label取出來並儲存成y_train
         y_train = df1['label']
         y_train

Out[22]: 1       1
         2       1
         3       0
         4       0
         5       0
                ..
         4982    0
         4983    0
         4984    0
         4985    0
         4986    0
         Name: label, Length: 4986, dtype: object
```

因為維度過高，因此先用PCA降維看看



```
In [23]: # 導入PCA套件
         from sklearn.decomposition import PCA
         pca = PCA(n_components=0.8)
         pca.fit(X_train)

Out[23]: PCA(n_components=0.8)

In [24]: # 做降維轉換
         X_train_pca = pca.transform(X_train)

In [25]: # 看一下降維之後的形狀
         X_train_pca.shape

Out[25]: (4986, 1969)
```

再來是對測試集進行處理

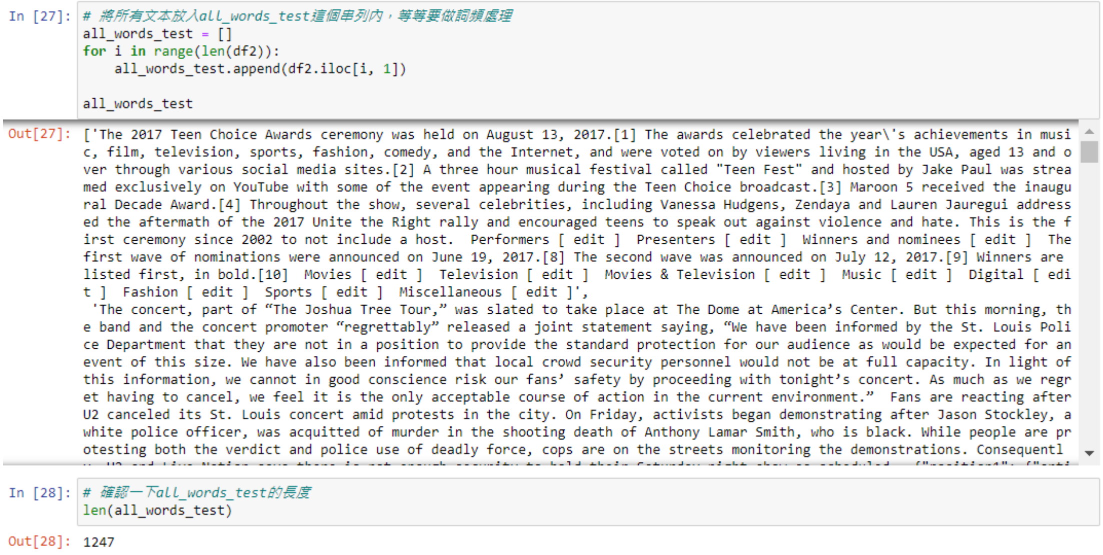


```
In [26]: # 用pandas導入測試集，分割符號採用table的方式，最後儲存成df2
         df2 = pd.read_csv('test.csv', sep='\t')
         df2
```

Out[26]:

	id	text
0	2	The 2017 Teen Choice Awards ceremony was held ...
1	3	The concert, part of “The Joshua Tree Tour,” w...
2	4	Selena Gomez refuses to talk to her mother abo...
3	5	This is worse than a lump of coal in your stoc...
4	6	Luann De Lesseps is going to rehab after her a...
...	...	...
1242	1244	Get the latest from TODAY Sign up for our news...
1243	1245	Jaden Smith claims that the Four Seasons Hotel...
1244	1246	Overview (3) Mini Bio (1) Faith Hill was bor...
1245	1247	CLOSE Aaron Paul dishes on 'The Path' Aaron P...
1246	1248	Meghan Edmonds was showered with love at her b...

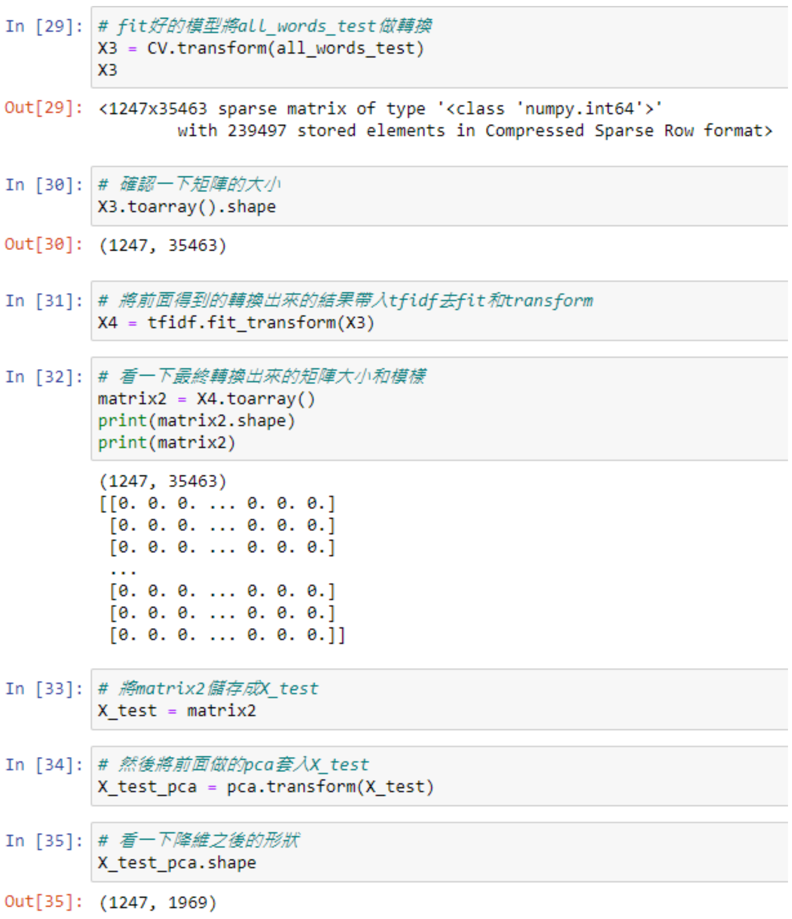
1247 rows × 2 columns



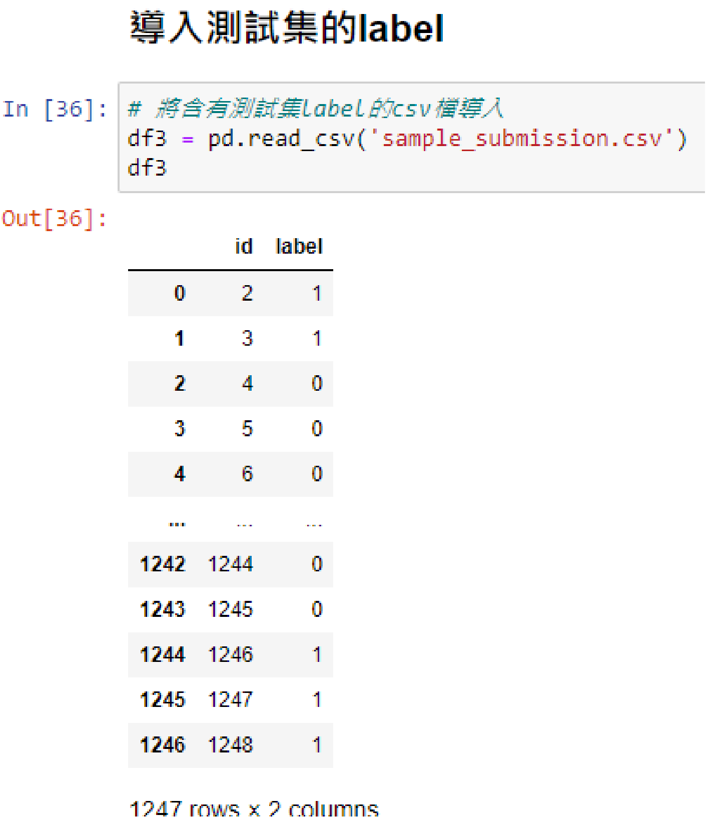
```
In [27]: # 將所有文本放入all_words_test這個串列內，等等要做詞頻處理
         all_words_test = []
         for i in range(len(df2)):
             all_words_test.append(df2.iloc[i, 1])

         all_words_test
```

```
Out[27]: ['The 2017 Teen Choice Awards ceremony was held on August 13, 2017.[1] The awards celebrated the year\'s achievements in musi
         c, film, television, sports, fashion, comedy, and the Internet, and were voted on by viewers living in the USA, aged 13 and o
         ver through various social media sites.[2] A three hour musical festival called "Teen Fest" and hosted by Jake Paul was strea
         med exclusively on YouTube with some of the event appearing during the Teen Choice broadcast.[3] Maroon 5 received the inaugu
         ral Decade Award.[4] Throughout the show, several celebrities, including Vanessa Hudgens, Zendaya and Lauren Jauregui address
         ed the aftermath of the 2017 Unite the Right rally and encouraged teens to speak out against violence and hate. This is the f
         irst ceremony since 2002 to not include a host.  Performers [ edit ]  Presenters [ edit ]  Winners and nominees [ edit ]  The
         first wave of nominations were announced on June 19, 2017.[8] The second wave was announced on July 12, 2017.[9] Winners are
         listed first, in bold.[10]  Movies [ edit ]  Television [ edit ]  Movies & Television [ edit ]  Music [ edit ]  Digital [ edi
         t ]  Fashion [ edit ]  Sports [ edit ]  Miscellaneous [ edit ]',
          'The concert, part of “The Joshua Tree Tour,” was slated to take place at The Dome at America’s Center. But this morning, th
         e band and the concert promoter “regrettably” released a joint statement saying, “We have been informed by the St. Louis Poli
         ce Department that they are not in a position to provide the standard protection for our audience as would be expected for an
         event of this size. We have also been informed that local crowd security personnel would not be at full capacity. In light of
         this information, we cannot in good conscience risk our fans’ safety by proceeding with tonight’s concert. As much as we regr
         et having to cancel, we feel it is the only acceptable course of action in the current environment.”  Fans are reacting after
         U2 canceled its St. Louis concert amid protests in the city. On Friday, activists began demonstrating after Jason Stockley, a
         white police officer, was acquitted of murder in the shooting death of Anthony Lamar Smith, who is black. While people are pr
         otesting both the verdict and police use of deadly force, cops are on the streets monitoring the demonstrations. Consequentl
```

```
In [28]: # 確認一下all_words_test的長度
         len(all_words_test)

Out[28]: 1247
```



```
In [29]: # fit好的模型將all_words_test做轉換
         X3 = CV.transform(all_words_test)
         X3

Out[29]: <1247x35463 sparse matrix of type '<class 'numpy.int64'>'
                 with 239497 stored elements in Compressed Sparse Row format>

In [30]: # 確認一下矩陣的大小
         X3.toarray().shape

Out[30]: (1247, 35463)

In [31]: # 將前面得到的轉換出來的結果帶入tfidf去fit和transform
         X4 = tfidf.fit_transform(X3)

In [32]: # 看一下最終轉換出來的矩陣大小和模樣
         matrix2 = X4.toarray()
         print(matrix2.shape)
         print(matrix2)

         (1247, 35463)
         [[0. 0. 0. ... 0. 0. 0.]
          [0. 0. 0. ... 0. 0. 0.]
          [0. 0. 0. ... 0. 0. 0.]
          ...
          [0. 0. 0. ... 0. 0. 0.]
          [0. 0. 0. ... 0. 0. 0.]
          [0. 0. 0. ... 0. 0. 0.]]

In [33]: # 將matrix2儲存成X_test
         X_test = matrix2

In [34]: # 然後將前面做的pca套入X_test
         X_test_pca = pca.transform(X_test)

In [35]: # 看一下降維之後的形狀
         X_test_pca.shape

Out[35]: (1247, 1969)
```

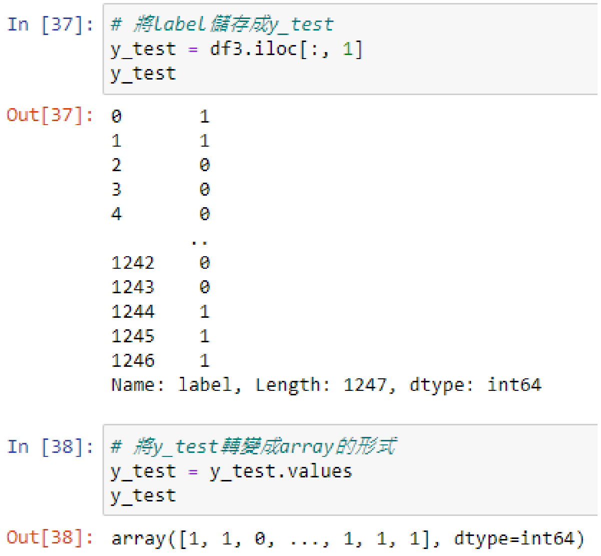


導入測試集的label

```
In [36]: # 將含有測試集label的csv檔導入
         df3 = pd.read_csv('sample_submission.csv')
         df3

Out[36]:
```

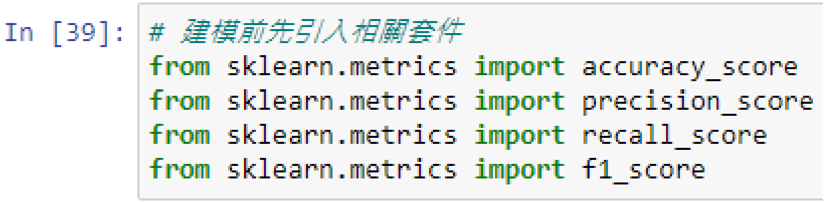
	id	label
0	2	1
1	3	1
2	4	0
3	5	0
4	6	0
...	...	...
1242	1244	0
1243	1245	0
1244	1246	1
1245	1247	1
1246	1248	1

1247 rows x 2 columns



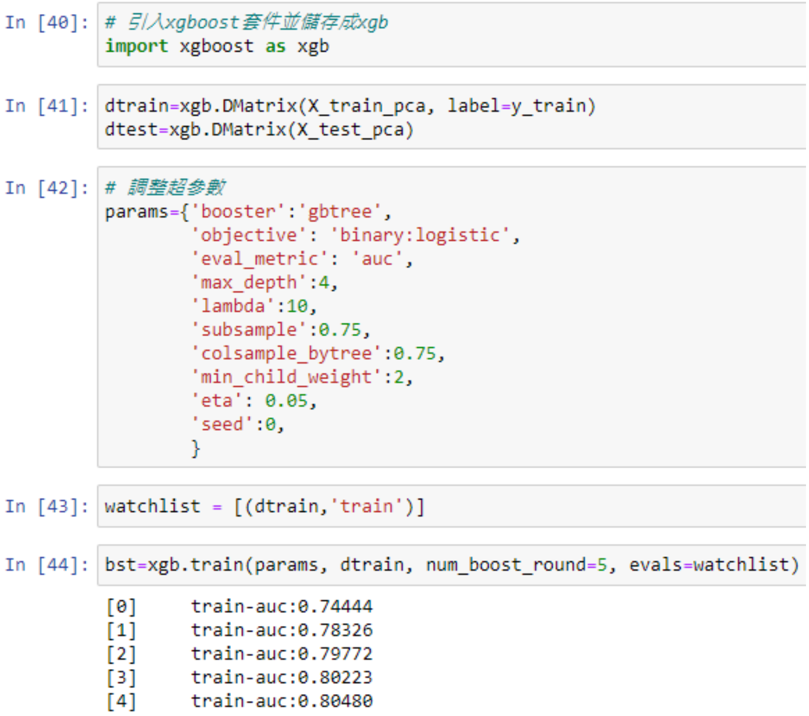
```
In [37]: # 將label儲存成y_test
         y_test = df3.iloc[:, 1]
         y_test

Out[37]: 0       1
         1       1
         2       0
         3       0
         4       0
                ..
         1242    0
         1243    0
         1244    1
         1245    1
         1246    1
         Name: label, Length: 1247, dtype: int64

In [38]: # 將y_test轉變成array的形式
         y_test = y_test.values
         y_test

Out[38]: array([1, 1, 0, ..., 1, 1, 1], dtype=int64)
```

建模

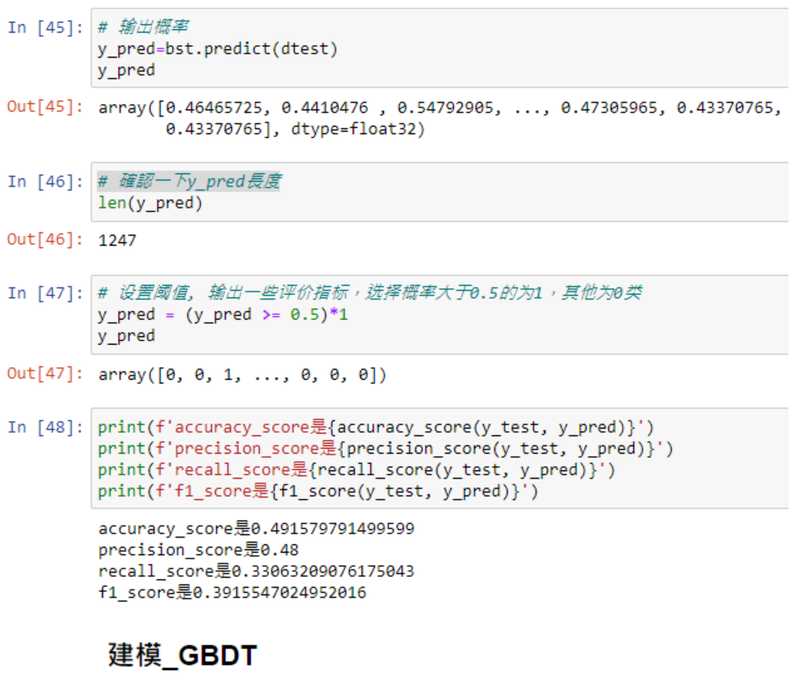


```
In [39]: # 建模前先引入相關套件
         from sklearn.metrics import accuracy_score
         from sklearn.metrics import precision_score
         from sklearn.metrics import recall_score
         from sklearn.metrics import f1_score
```

建模_xgboost

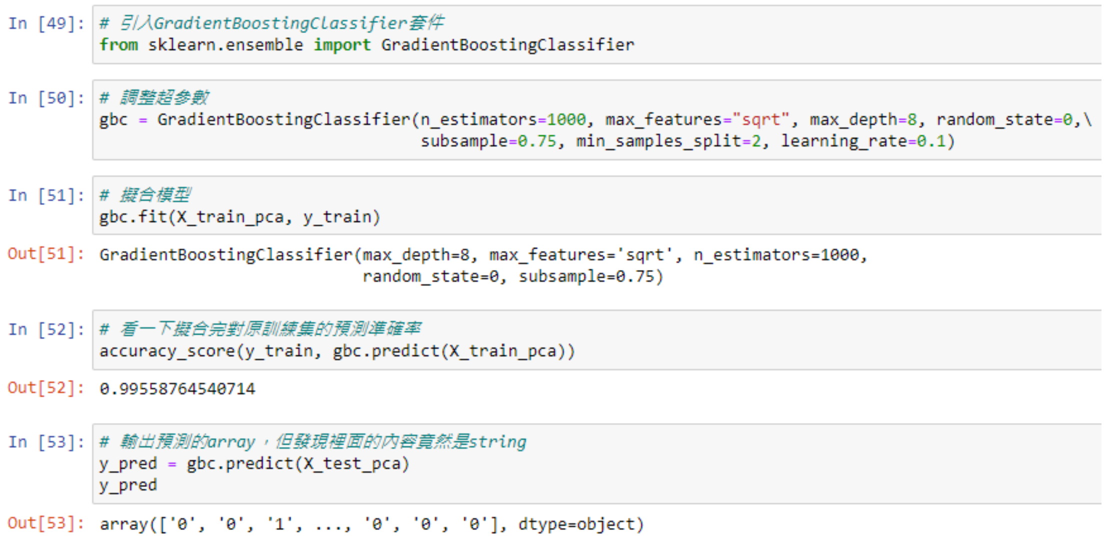


```
In [40]: # 引入xgboost套件並儲存成xgb
         import xgboost as xgb

In [41]: dtrain=xgb.DMatrix(X_train_pca, label=y_train)
         dtest=xgb.DMatrix(X_test_pca)

In [42]: # 調整超參數
         params={'booster':'gbtree',
                 'objective': 'binary:logistic',
                 'eval_metric': 'auc',
                 'max_depth':4,
                 'lambda':10,
                 'subsample':0.75,
                 'colsample_bytree':0.75,
                 'min_child_weight':2,
                 'eta': 0.05,
                 'seed':0,
                 }

In [43]: watchlist = [(dtrain,'train')]

In [44]: bst=xgb.train(params, dtrain, num_boost_round=5, evals=watchlist)

         [0]     train-auc:0.74444
         [1]     train-auc:0.78326
         [2]     train-auc:0.79772
         [3]     train-auc:0.80223
         [4]     train-auc:0.80480
```

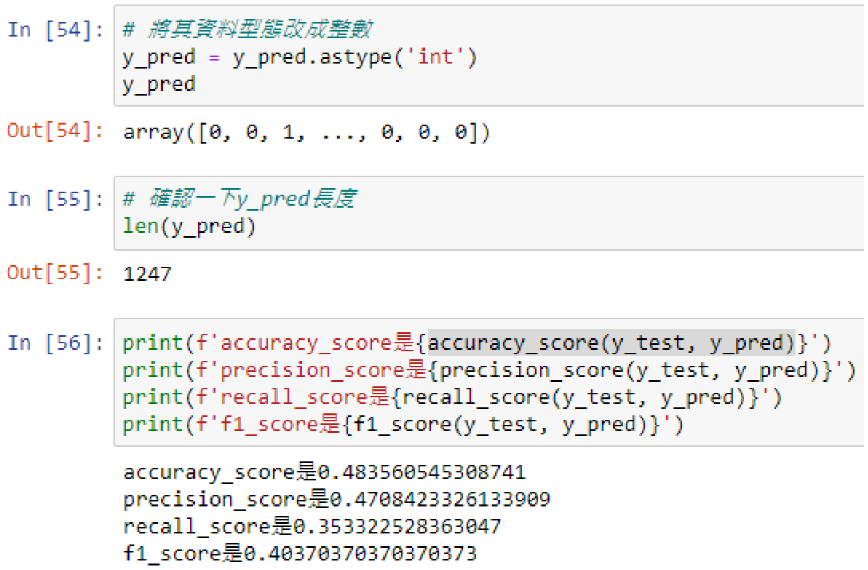


```
In [45]: # 输出概率
         y_pred=bst.predict(dtest)
         y_pred

Out[45]: array([0.46465725, 0.4410476 , 0.54792905, ..., 0.47305965, 0.43370765,
                0.43370765], dtype=float32)

In [46]: # 確認一下y_pred長度
         len(y_pred)

Out[46]: 1247

In [47]: # 设置阈值, 输出一些评价指标，选择概率大于0.5的为1，其他为0类
         y_pred = (y_pred >= 0.5)*1
         y_pred

Out[47]: array([0, 0, 1, ..., 0, 0, 0])

In [48]: print(f'accuracy_score是{accuracy_score(y_test, y_pred)}')
         print(f'precision_score是{precision_score(y_test, y_pred)}')
         print(f'recall_score是{recall_score(y_test, y_pred)}')
         print(f'f1_score是{f1_score(y_test, y_pred)}')

         accuracy_score是0.491579791499599
         precision_score是0.48
         recall_score是0.33063209076175043
         f1_score是0.3915547024952016
```

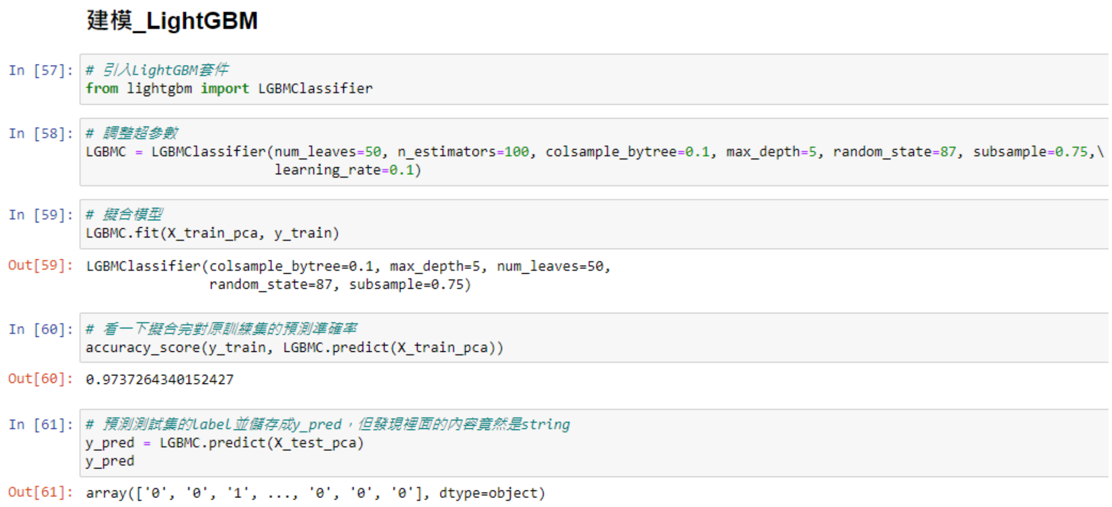
建模_GBDT



```
In [49]: # 引入GradientBoostingClassifier套件
         from sklearn.ensemble import GradientBoostingClassifier

In [50]: # 調整超參數
         gbc = GradientBoostingClassifier(n_estimators=1000, max_features="sqrt", max_depth=8, random_state=0,\
                                          subsample=0.75, min_samples_split=2, learning_rate=0.1)

In [51]: # 擬合模型
         gbc.fit(X_train_pca, y_train)

Out[51]: GradientBoostingClassifier(max_depth=8, max_features='sqrt', n_estimators=1000,
                                    random_state=0, subsample=0.75)

In [52]: # 看一下擬合完對原訓練集的預測準確率
         accuracy_score(y_train, gbc.predict(X_train_pca))

Out[52]: 0.99558764540714

In [53]: # 輸出預測的array，但發現裡面的內容竟然是string
         y_pred = gbc.predict(X_test_pca)
         y_pred

Out[53]: array(['0', '0', '1', ..., '0', '0', '0'], dtype=object)
```

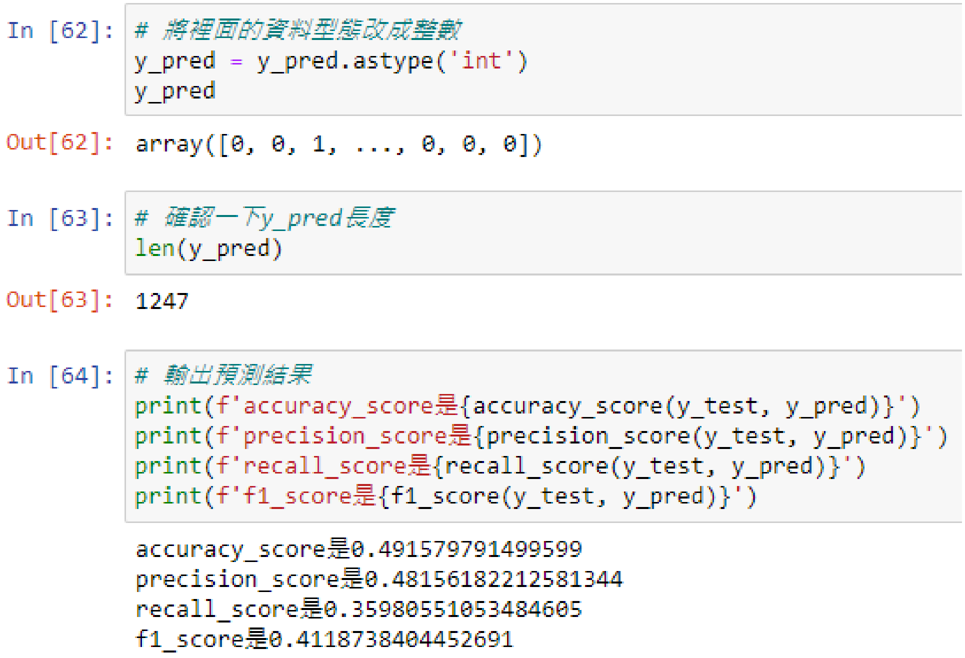


```
In [54]: # 將其資料型態改成整數
         y_pred = y_pred.astype('int')
         y_pred

Out[54]: array([0, 0, 1, ..., 0, 0, 0])

In [55]: # 確認一下y_pred長度
         len(y_pred)

Out[55]: 1247

In [56]: print(f'accuracy_score是{accuracy_score(y_test, y_pred)}')
         print(f'precision_score是{precision_score(y_test, y_pred)}')
         print(f'recall_score是{recall_score(y_test, y_pred)}')
         print(f'f1_score是{f1_score(y_test, y_pred)}')

         accuracy_score是0.483560545308741
         precision_score是0.4708423326133909
         recall_score是0.353322528363047
         f1_score是0.40370370370370373
```



建模_LightGBM

```
In [57]: # 引入LightGBM套件
         from lightgbm import LGBMClassifier

In [58]: # 調整超參數
         LGBMC = LGBMClassifier(num_leaves=50, n_estimators=100, colsample_bytree=0.1, max_depth=5, random_state=87, subsample=0.75,\
                               learning_rate=0.1)

In [59]: # 擬合模型
         LGBMC.fit(X_train_pca, y_train)

Out[59]: LGBMClassifier(colsample_bytree=0.1, max_depth=5, num_leaves=50,
                        random_state=87, subsample=0.75)

In [60]: # 看一下擬合完對原訓練集的預測準確率
         accuracy_score(y_train, LGBMC.predict(X_train_pca))

Out[60]: 0.9737264340152427

In [61]: # 預測測試集的label並儲存成y_pred，但發現裡面的內容竟然是string
         y_pred = LGBMC.predict(X_test_pca)
         y_pred

Out[61]: array(['0', '0', '1', ..., '0', '0', '0'], dtype=object)
```



```
In [62]: # 將裡面的資料型態改成整數
         y_pred = y_pred.astype('int')
         y_pred

Out[62]: array([0, 0, 1, ..., 0, 0, 0])

In [63]: # 確認一下y_pred長度
         len(y_pred)

Out[63]: 1247

In [64]: # 輸出預測結果
         print(f'accuracy_score是{accuracy_score(y_test, y_pred)}')
         print(f'precision_score是{precision_score(y_test, y_pred)}')
         print(f'recall_score是{recall_score(y_test, y_pred)}')
         print(f'f1_score是{f1_score(y_test, y_pred)}')

         accuracy_score是0.491579791499599
         precision_score是0.48156182212581344
         recall_score是0.35980551053484605
         f1_score是0.4118738404452691
```