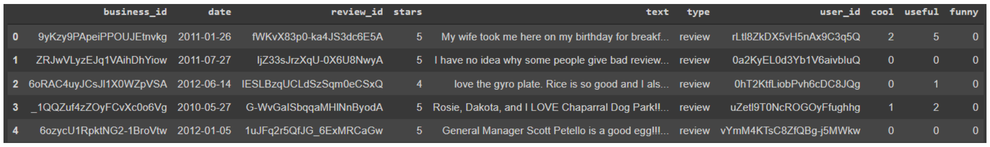
Sentiment analysis
資料探勘 HW2

Department : IMM, 洪翊誠, ID : 310653005

October 2021

Load data

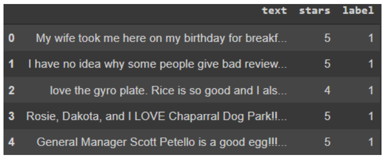
- 讀資料



	business_id	date	review_id	stars	text	type	user_id	cool	useful	funny
0	9yKzy9PApeiPPOUJEtnvkg	2011-01-26	fWKvX83p0-ka4JS3dc6E5A	5	My wife took me here on my birthday for breakf...	review	rLtl8ZkDX5vH5nAx9C3q5Q	2	5	0
1	ZRJwVLyzEJq1VAihDhYiow	2011-07-27	IjZ33sJrzXqU-0X6U8NwyA	5	I have no idea why some people give bad review...	review	0a2KyEL0d3Yb1V6aivbIuQ	0	0	0
2	6oRAC4uyJCsJl1X0WZpVSA	2012-06-14	IESLBzqUCLdSzSqm0eCSxQ	4	love the gyro plate. Rice is so good and I als...	review	0hT2KtfLiobPvh6cDC8JQg	0	1	0
3	_1QQZuf4zZOyFCvXc0o6Vg	2010-05-27	G-WvGaISbqqaMHlNnByodA	5	Rosie, Dakota, and I LOVE Chaparral Dog Park!!...	review	uZetl9T0NcROGOyFfughhg	1	2	0
4	6ozycU1RpktNG2-1BroVtw	2012-01-05	1uJFq2r5QfJG_6ExMRCaGw	5	General Manager Scott Petello is a good egg!!!...	review	vYmM4KTsC8ZfQBg-j5MWkw	0	0	0

Figure 1: DataFrame

- 取出我們所需保留資料



	text	stars	label
0	My wife took me here on my birthday for breakf...	5	1
1	I have no idea why some people give bad review...	5	1
2	love the gyro plate. Rice is so good and I als...	4	1
3	Rosie, Dakota, and I LOVE Chaparral Dog Park!!...	5	1
4	General Manager Scott Petello is a good egg!!!...	5	1

Figure 2: DataFrame

- 瀏覽完資料內容後，隨後步驟開始分割句子段落，將具整理成文字陣列，將其轉成數字向量再進行分類，為了讓過程進行得更整潔流暢，把一些結構包裝成函數的形式。

Function

- K-folder

```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
def K_fold_CV(k, data, label):
  subset_size = len(data)//k
  Acc_list = []
  for i in range(k):
    print(f'第{i+1}次切割 : ')
    test_start, test_end = i*subset_size,(i+1)*subset_size
    test_data, test_label = data[test_start:test_end], label[test_start:test_end]
    train_data = np.vstack([data[:test_start], data[test_end:]])
    train_label = np.hstack([label[:test_start], label[test_end:]])
    rfc = RandomForestClassifier()
    rfc.fit(train_data, train_label)
    pred_rfc = rfc.predict(test_data)
    acc = accuracy_score(test_label , pred_rfc)
    print(f'accuracy : {acc:.4f}')
    Acc_list.append(acc)
  return sum(Acc_list)/k
```

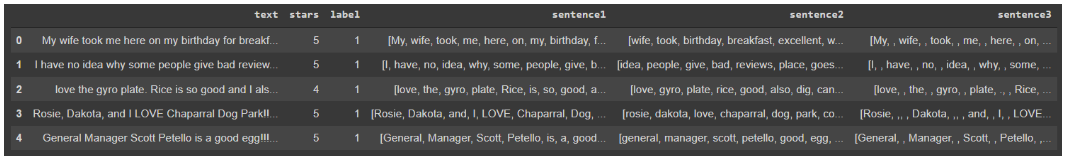
- Stop Words

```python
import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
def review_to_wordlist(review, remove_stopwords=True):
    """
    Convert a review to a list of words. Removal of stop words is optional.
    """
    # remove non-letters
    review_text = re.sub("[^a-zA-Z]"," ", review)
    # convert to lower case and split at whitespace
    words = review_text.lower().split()
    # remove stop words (false by default)
    if remove_stopwords:
        stops = set(stopwords.words("english"))
        words = [w for w in words if not w in stops]
    return words
```

Splitting Method

- 使用三種不同分割方式，見出三種不同 sentence 欄位，未來進行比較時可以觀測。
  - 自行切割
  - 下載停頓詞資訊庫使用 (nltk)
  - 下載停頓詞套件輔助 (jieba)

```python
import re
import jieba
df['sentence1'] = df['text'].map(lambda x:
    re.sub(r'[0-9\.\-\!\"\(\)\,]', '', x).split())
df['sentence2'] = df['text'].apply(review_to_wordlist)
df['sentence3'] = df['text'].apply(jieba.lcut) # Take time
```



	text	stars	label	sentence1	sentence2	sentence3
0	My wife took me here on my birthday for breakf...	5	1	[My, wife, took, me, here, on, my, birthday, f...	[wife, took, birthday, breakfast, excellent, w...	[My, , wife, , took, , me, , here, , on, ...
1	I have no idea why some people give bad review...	5	1	[I, have, no, idea, why, some, people, give, b...	[idea, people, give, bad, reviews, place, goes...	[I, , have, , no, , idea, , why, , some, ...
2	love the gyro plate. Rice is so good and I als...	4	1	[love, the, gyro, plate, Rice, is, so, good, a...	[love, gyro, plate, rice, good, also, dig, can...	[love, , the, , gyro, , plate, ., , Rice, ...
3	Rosie, Dakota, and I LOVE Chaparral Dog Park!!...	5	1	[Rosie, Dakota, and, I, LOVE, Chaparral, Dog, ...	[rosie, dakota, love, chaparral, dog, park, co...	[Rosie, ,, , Dakota, ,, , and, , I, , LOVE...
4	General Manager Scott Petello is a good egg!!!...	5	1	[General, Manager, Scott, Petello, is, a, good...	[general, manager, scott, petello, good, egg, ...	[General, , Manager, , Scott, , Petello, ,...

Figure 3: Sentences split comparison

呈現資料表格其實不難發現整理得最乾淨的是 sentence2，進行大小寫處理，不必要詞彙例如 My 也移除，針對動作物品進行擷取最到位

Apply in 3 different Ways

1. Vectorization 向量化文字

```python
CV = CountVectorizer(stop_words='english')
data_text = CV.fit_transform(df['text']).toarray()
data_label = df['label'].tolist()
```

- 開始進行 k folder 設定 $k = 4$

切割第 i 份	1	2	3	4	平均
accuracy	0.7888	0.7988	0.7960	0.7972	0.7952

2. Word2Vector
   因為模型計算時間久，建議儲存下來之後直接 load 回來。

```
model_name = 'Word2Vec_model1'
if not os.path.exists(model_name):
  model = Word2Vec(df['sentence1'], size=250, iter=10, sg=1,
      min_count=1)
  model.save(model_name)
else:
  model = Word2Vec.load(model_name)

model_name = 'Word2Vec_model2'
if not os.path.exists(model_name):
  model = Word2Vec(df['sentence2'], size=250, iter=10, sg=1,
      min_count=1)
  model.save(model_name)
else:
  model = Word2Vec.load(model_name)

model_name = 'Word2Vec_model3'
if not os.path.exists(model_name):
  model = Word2Vec(df['sentence3'], size=250, iter=10, sg=1,
      min_count=1)
  model.save(model_name)
else:
  model = Word2Vec.load(model_name)
```

- Evaluate the accuracy

```
for num in range(1,4):

  sentence_num = f'sentence{num}'
  model_name = f'Word2Vec_model{num}'
  model = Word2Vec.load(model_name)
  data_text = np.zeros((len(df[sentence_num]), 250),
      dtype='float64')
  for i in range(len(df[sentence_num])):
    for j in range(min(len(df[sentence_num][i]), 100)):
      data_text[i, j] =
          model.wv.vocab[df[sentence_num][i][j]].index + 1
  print(f'Case {num}: analysis with the {sentence_num}')
  acc_avg = K_fold_CV(k=4, data=data_text, label=data_label)
```

- 開始進行 k folder 設定 $k = 4$

切割第 i 份 Sentence	1	2	3	4	平均
sentence1	0.6856	0.6904	0.6768	0.6872	0.6850
sentence2	0.6832	0.6892	0.6732	0.6896	0.6838
sentence3	0.6820	0.6884	0.6756	0.6888	0.6837

3. TfidfVectorizer

```python
#Import TfIdfVectorizer from scikit-learn
from sklearn.feature_extraction.text import TfidfVectorizer
#Define a TF-IDF Vectorizer Object. Remove all english stop words such as 'the', 'a'
tfidf = TfidfVectorizer(stop_words='english')
#Replace NaN with an empty string
df['text'] = df['text'].fillna('')
#Construct the required TF-IDF matrix by fitting and transforming the data
tfidf_data = tfidf.fit_transform(df['text']).toarray()
#Output the shape of tfidf_matrix
tfidf_data.shape, tfidf_data.sum()
```

- Evaluate the accuracy

```python
print('Case in TfidfVectorizer :')
acc_avg = K_fold_CV(k=4, data=tfidf_data, label=data_label)
```

- 開始進行 k folder 設定 $k = 4$

切割第 i 份	1	2	3	4	平均
accuracy	0.7820	0.7908	0.7744	0.7896	0.7842

Conclusion

就結果比較而言，看似 TfidfVectorizer 結果最好，但是必須考量到在使用 Word2Vector 時，有設定 size 大小，一個超高維度的資料限縮在使用 250 維度來表示，可能會限制其所代表的資料量，以至於在最後的準確度上有所差異；時間上的差異，word2vector 體趕來說真的是裡面相對季間比較久的，而且在這部分也透過了不同的停頓詞分解去觀測解析上的差異，或許可能要再調整 size 才能得到更好的對比分析。另外有一個問題，k-folder 致力於對不同的 validation 做訓練，將所有屬於 training 的資料都使用完善，但在做這些動作之前，是否應該對資料集先切成 training, testing，再從 training 做 k-folder 切成 training 和 validation，最後再對 testing 進行 inference 分析，這麼做才能

說明到我的 model 經過 k-folder 一連串之訓練後能達到對完全沒觀察過的資料有合理且正確的預測結果。