Data Mining Research and Practice HW2

Department: IAM Student ID: 309652008 Name: 廖家緯

October 28, 2021

1 使用套件

```
import re
import numpy as np
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
from sklearn.ensemble import RandomForestClassifier
from gensim.models import Word2Vec
```

- re: 字串處理
- pandas: 讀取資料
- numpy: 矩陣計算
- sklearn: CountVectorizer, TfidfTransformer, RandomForestClassifier
- gensim: Word2Vec

2 資料前處理

2.1 讀取資料

```
df = pd.read_csv('yelp.csv')
df.head()
```

	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. Data Frame

2.2 取出 text、stars 欄位

```
data = df[['text', 'stars']]
```

2.2.1 定義 Label

$$\text{label} = \begin{cases} 1, & \text{if stars} \geq 4 \\ 0, & \text{if stars} < 4 \end{cases}$$

```
data['label'] = data['stars'].map(lambda x: int(x>=4))
```

2.2.2 拆解句子

去除數字與標點符號，並將單字整理成陣列

```
data['sentence'] = data['text'].map(lambda x: re.sub(r'[0-9\.\-\!\"\(\)\,]', '',
    x).split())
```

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

Figure 2. Data Frame

2.3 文字轉向量

2.3.1 TF-IDF

先去除 stop words，再使用 TfidfTransformer

$$\text{TFIDF}(w, D) = \frac{n(w, D)}{\max_{w'} n(w', D)} \times \left(1 + \log \frac{|D|}{n(w, D)}\right)$$

其中 $n(w, D)$ 表示單字 w 出現在文件 D 的次數，$|D|$ 表示文件 D 的總字數

```
count_vector = CountVectorizer(stop_words='english')
count_vector_matrix = count_vector.fit_transform(data['text']).toarray()
tfidf_transformer = TfidfTransformer()
X = tfidf_transformer.fit_transform(count_vector_matrix).toarray()
Y = data['label'].tolist()
```

2.3.2 Word to Vector

```
def word_vector(w2v, tokens, size):
    vec = np.zeros(size).reshape((1, size))
    count = 0

    for word in tokens:
        try:
          vec += w2v[word].reshape((1, size))
          count += 1

        except KeyError:
          continue

    if count != 0:
        vec /= count

    return vec
```

```
w2v = Word2Vec(data['sentence'], min_count=1, size=250, iter=30, sg=1)
w2v_matrix = np.zeros((len(data['sentence']), 250))

for i in range(len(data['sentence'])):
    w2v_matrix[i,:] = word_vector(w2v, data['sentence'][i], 250)

X = w2v_matrix
Y = data['label']
```

- size: encode 的空間維度
- sg: skip gram 方法
- 與 days 相似的單字

word	cosine similarity
weeks	0.4538
spring	0.4197
hour	0.4166
later	0.4121
noon	0.4119

其中 $\cos(w_1, w_2) = \dfrac{w_1 \cdot w_2}{|w_1||w_2|}$

3 隨機森林建模

利用 k-fold cross-validation

- split_K_fold: 將資料集切分成 k 份
- get_train_test: k 份資料集輪流當 test set，其餘當 training set
- train_test_step: 使用隨機森林建模
- fit: 執行以上步驟

```
class K_fold_CV(object):
    def __init__(self, X, Y, k):
        self.X = X
        self.Y = Y
        self.k = k


    def split_K_fold(self):
        self.split_data = [self.X[0+i*len(self.X)//self.k :
            (i+1)*len(self.X)//self.k] for i in range(self.k)]
        self.split_label = [self.Y[0+i*len(self.Y)//self.k :
            (i+1)*len(self.Y)//self.k] for i in range(self.k)]

        return None


    def get_train_test(self, iter):
        train_data = np.vstack([self.split_data[j] for j in range(self.k) if
            j!=iter])
        train_label = np.hstack([self.split_label[j] for j in range(self.k) if
            j!=iter])
        test_data = self.split_data[iter]
        test_label = self.split_label[iter]

        return train_data, train_label, test_data, test_label


    def train_test_step(self, train_data, train_label, test_data, test_label):
        rf = RandomForestClassifier(n_estimators=100, n_jobs=-1)
        rf.fit(train_data, train_label)
        acc = rf.score(test_data, test_label)

        return acc


    def fit(self):
        sum_acc = 0
        self.split_K_fold()
        for iter in range(self.k):
```

```
            train_data, train_label, test_data, test_label =
                self.get_train_test(iter)
            acc = self.train_test_step(train_data, train_label, test_data,
                test_label)
            sum_acc += acc
            print(f"k={iter+1}/{self.k}, acc={acc}")

        print(f"average acc={sum_acc / self.k}")

        return None
```

4 執行結果

```
K_fold_CV(X, Y, k=4).fit()
```

fold	1	2	3	4	Average
tf-idf	0.7832	0.7920	0.7696	0.7856	0.7826
word2vec	0.7704	0.7840	0.7748	0.7740	0.7758

tf-idf 是根據文件出現頻率訂定的指標，理論依據較不足。而 word to vector 利用單字訓練出 latent space，擁有空間結構，我們可以用幾何方法對不同單字做相似度分析，可解釋性較強。以上兩個方法可以做到差不多水準，但無法做公平性的比較。