Amazon Reviews Analysis

Python

實作報告

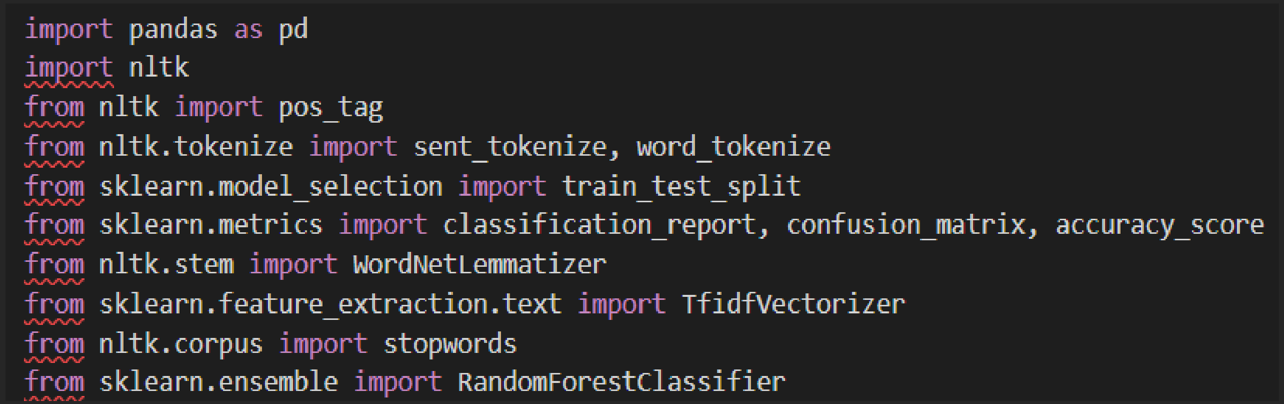
Importing Libraries & Data Set

Mainly use:

Data processing: pandas

Text processing: nltk

ML: sklearn (Random Forest)



```
import pandas as pd
import nltk
from nltk import pos_tag
from nltk.tokenize import sent_tokenize, word_tokenize
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
from nltk.corpus import stopwords
from sklearn.ensemble import RandomForestClassifier
```

Data Set:

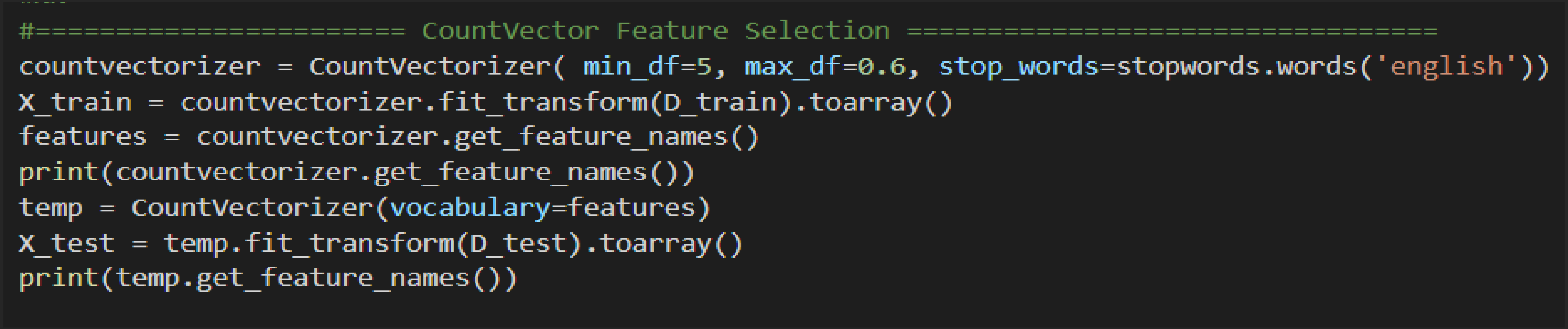
Amazon Review Full Score   Dataset

(constructed by Xiang Zhang)

The dataset train.csv and test.csv contain all the training samples as comma-sparated values. There are 3 columns in them, corresponding to class index (1 to 5), review title and review text.

(example: 4, Surprisingly delightful, This is a fast read filled with unexpected humour and profound insights into the art of politics and policy. In brief, it is sly, wry, and wise.)

Feature Selection



```
#====================== CountVector Feature Selection ================================
countvectorizer = CountVectorizer( min_df=5, max_df=0.6, stop_words=stopwords.words('english'))
X_train = countvectorizer.fit_transform(D_train).toarray()
features = countvectorizer.get_feature_names()
print(countvectorizer.get_feature_names())
temp = CountVectorizer(vocabulary=features)
X_test = temp.fit_transform(D_test).toarray()
print(temp.get_feature_names())
```

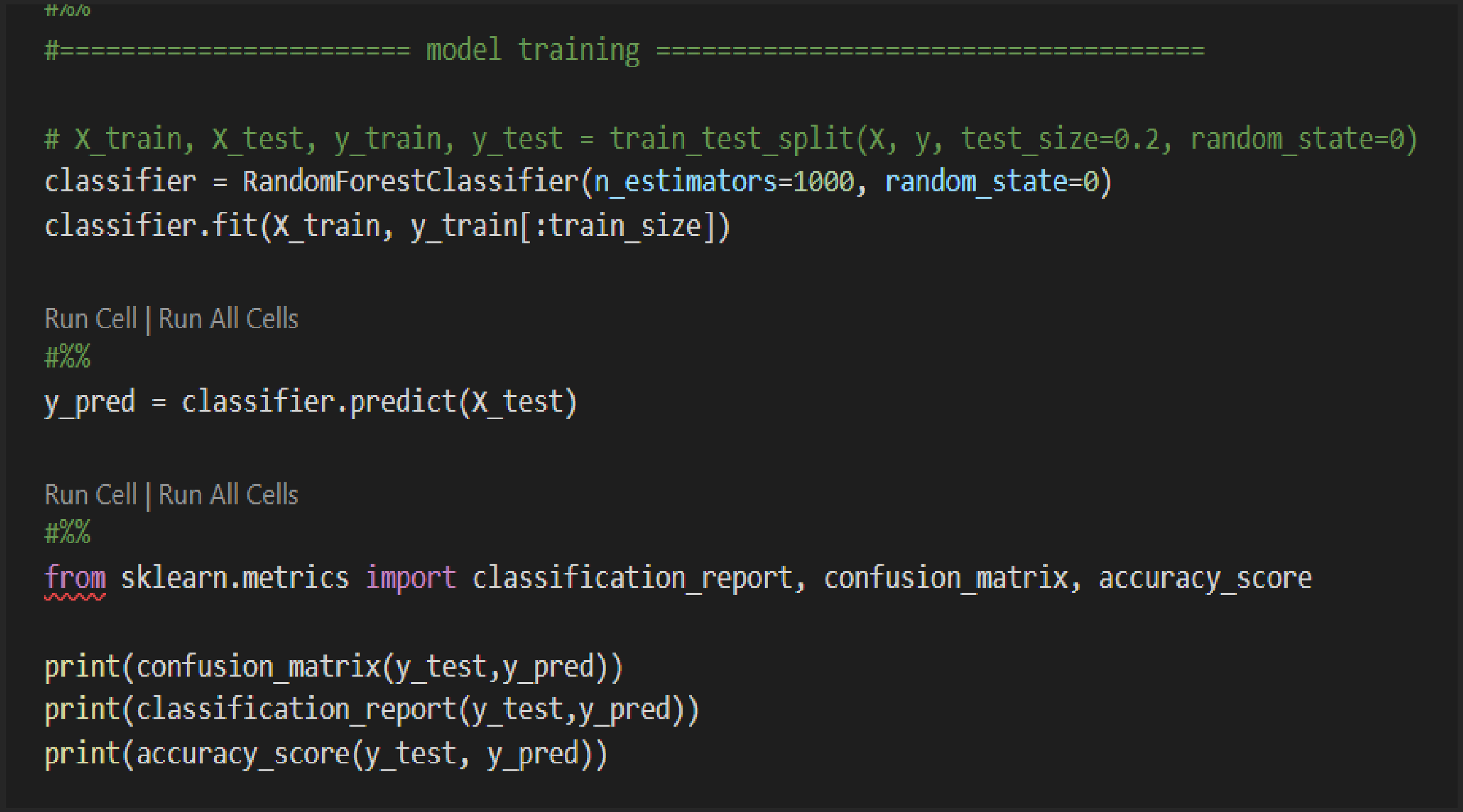
挑選Features:

我們選擇使用 CountVector 當作選擇Feature時的指標，濾掉詞頻出現過多(超過60%文章出
現過)或詞頻過少的單字(80000篇中低於5篇)，挑選出1500個最具代表性的單字作為
Feature

Model Training & Predicting Category

Random Forest:

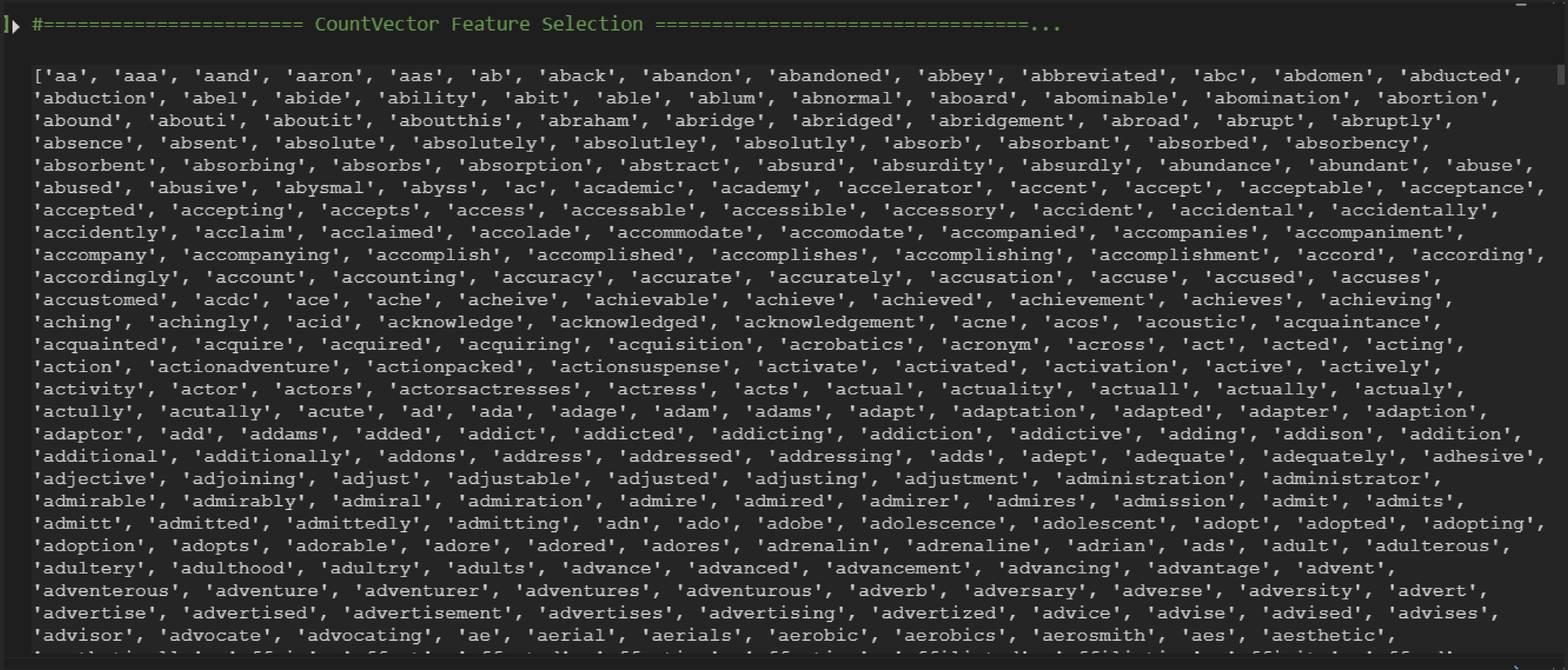
用1000個estimators 建立模型去預測test data的分類結果



```
#%%
#======================= model training ===================================

# X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
classifier = RandomForestClassifier(n_estimators=1000, random_state=0)
classifier.fit(X_train, y_train[:train_size])

Run Cell | Run All Cells
#%%
y_pred = classifier.predict(X_test)

Run Cell | Run All Cells
#%%
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(accuracy_score(y_test, y_pred))
```

Output (1500 features)

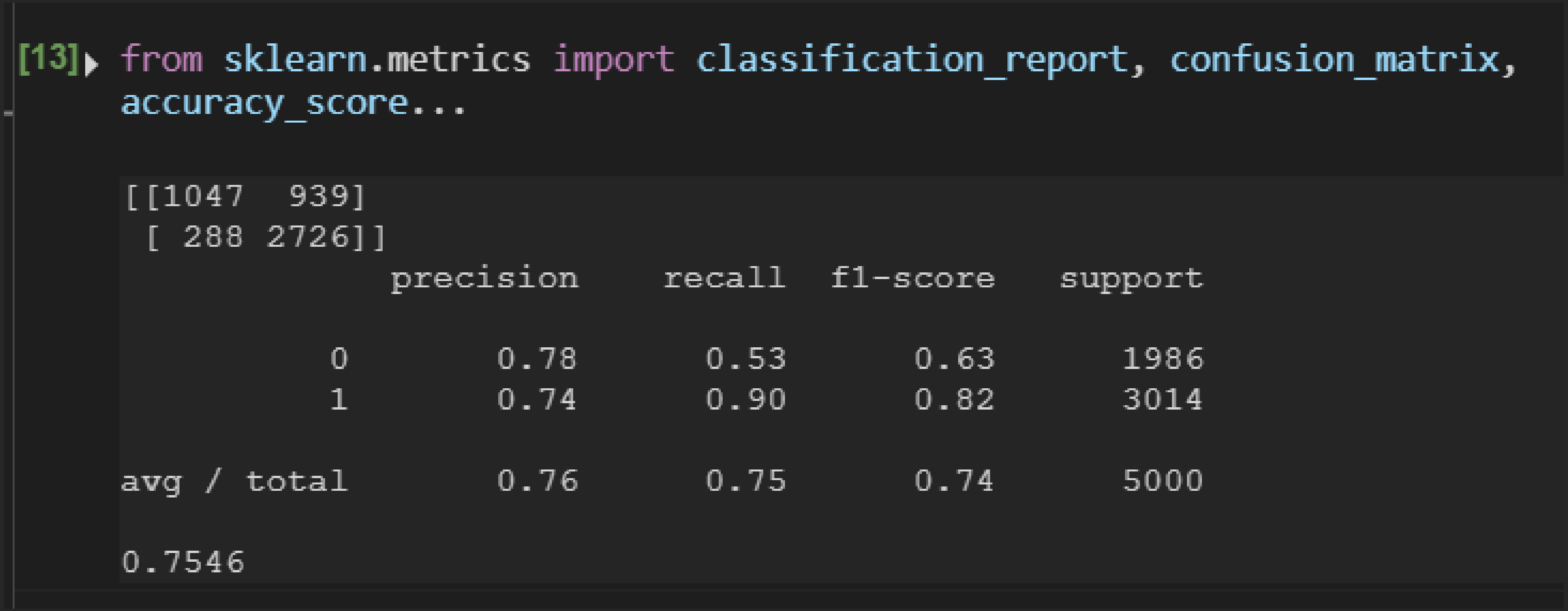
Selected features:



```
#===================== CountVector Feature Selection ==================================...

['aa', 'aaa', 'aand', 'aaron', 'aas', 'ab', 'aback', 'abandon', 'abandoned', 'abbey', 'abbreviated', 'abc', 'abdomen', 'abducted',
'abduction', 'abel', 'abide', 'ability', 'abit', 'able', 'ablum', 'abnormal', 'aboard', 'abominable', 'abomination', 'abortion',
'abound', 'abouti', 'aboutit', 'aboutthis', 'abraham', 'abridge', 'abridged', 'abridgement', 'abroad', 'abrupt', 'abruptly',
'absence', 'absent', 'absolute', 'absolutely', 'absolutley', 'absolutly', 'absorb', 'absorbant', 'absorbed', 'absorbency',
'absorbent', 'absorbing', 'absorbs', 'absorption', 'abstract', 'absurd', 'absurdity', 'absurdly', 'abundance', 'abundant', 'abuse',
'abused', 'abusive', 'abysmal', 'abyss', 'ac', 'academic', 'academy', 'accelerator', 'accent', 'accept', 'acceptable', 'acceptance',
'accepted', 'accepting', 'accepts', 'access', 'accessable', 'accessible', 'accessory', 'accident', 'accidental', 'accidentally',
'accidently', 'acclaim', 'acclaimed', 'accolade', 'accommodate', 'accomodate', 'accompanied', 'accompanies', 'accompaniment',
'accompany', 'accompanying', 'accomplish', 'accomplished', 'accomplishes', 'accomplishing', 'accomplishment', 'accord', 'according',
'accordingly', 'account', 'accounting', 'accuracy', 'accurate', 'accurately', 'accusation', 'accuse', 'accused', 'accuses',
'accustomed', 'acdc', 'ace', 'ache', 'acheive', 'achievable', 'achieve', 'achieved', 'achievement', 'achieves', 'achieving',
'aching', 'achingly', 'acid', 'acknowledge', 'acknowledged', 'acknowledgement', 'acne', 'acos', 'acoustic', 'acquaintance',
'acquainted', 'acquire', 'acquired', 'acquiring', 'acquisition', 'acrobatics', 'acronym', 'across', 'act', 'acted', 'acting',
'action', 'actionadventure', 'actionpacked', 'actionsuspense', 'activate', 'activated', 'activation', 'active', 'actively',
'activity', 'actor', 'actors', 'actorsactresses', 'actress', 'acts', 'actual', 'actuality', 'actuall', 'actually', 'actualy',
'actully', 'acutally', 'acute', 'ad', 'ada', 'adage', 'adam', 'adams', 'adapt', 'adaptation', 'adapted', 'adapter', 'adaption',
'adaptor', 'add', 'addams', 'added', 'addict', 'addicted', 'addicting', 'addiction', 'addictive', 'adding', 'addison', 'addition',
'additional', 'additionally', 'addons', 'address', 'addressed', 'addressing', 'adds', 'adept', 'adequate', 'adequately', 'adhesive',
'adjective', 'adjoining', 'adjust', 'adjustable', 'adjusted', 'adjusting', 'adjustment', 'administration', 'administrator',
'admirable', 'admirably', 'admiral', 'admiration', 'admire', 'admired', 'admirer', 'admires', 'admission', 'admit', 'admits',
'admitt', 'admitted', 'admittedly', 'admitting', 'adn', 'ado', 'adobe', 'adolescence', 'adolescent', 'adopt', 'adopted', 'adopting',
'adoption', 'adopts', 'adorable', 'adore', 'adored', 'adores', 'adrenalin', 'adrenaline', 'adrian', 'ads', 'adult', 'adulterous',
'adultery', 'adulthood', 'adultry', 'adults', 'advance', 'advanced', 'advancement', 'advancing', 'advantage', 'advent',
'adventerous', 'adventure', 'adventurer', 'adventures', 'adventurous', 'adverb', 'adversary', 'adverse', 'adversity', 'advert',
'advertise', 'advertised', 'advertisement', 'advertises', 'advertising', 'advertized', 'advice', 'advise', 'advised', 'advises',
'advisor', 'advocate', 'advocating', 'ae', 'aerial', 'aerials', 'aerobic', 'aerobics', 'aerosmith', 'aes', 'aesthetic',
```

Output (3000 features)

Total - accuracy: 0.7546



```
[13]  from sklearn.metrics import classification_report, confusion_matrix,
      accuracy_score...

      [[1047  939]
       [ 288 2726]]
                   precision    recall  f1-score   support

                0       0.78      0.53      0.63      1986
                1       0.74      0.90      0.82      3014

      avg / total       0.76      0.75      0.74      5000

      0.7546
```