Introduction

這個作業主要是利用關鍵字和MultinomialNB訓練一分析文章的model,並用此model分析其它文章的內容並預測其分類

Objective

主要目標

1. 取199筆文章並做關鍵字分析和training
2. 取50筆文章測試training model並給出F-score, Precision score 和 Recall score

Actions

我們選用 MultinomialNB訓練我們從文章所得之關鍵字,關鍵字之截取因程式碼過長所以不列出

```
vectorizer=CountVectorizer()
transformer=TfidfTransformer()
tfidf=transformer.fit_transform(vectorizer.fit_transform(finalkey)
)
y = np.array([1,2,3,4,5,6,7])#1:htc 2:samsung 3:apple 4:xiaomi
5:sony 6:lg 7:asus
clf = MultinomialNB().fit(tfidf, y)
```

預測test data之內容並印出score

```
y_pred=np.array([])
y_true=np.array([])
brand_dict={"htc":1,"sam":2,"apple":3,"xm":4,"sony":5,"lg":6,"asus
":7}

with open('test_data.json',encoding = 'utf8')  as content_file:
      for line in content_file:
      item=json.loads(line)
      testcontent=""

testcontent=testcontent+''.join(item['title'])+''.join(item['conte
nt'])
      testkeyword=convert_doc_to_wordlist(testcontent,False)
      testkey=[]
      testkey.append(' '.join(testkeyword))
```

```
testset=transformer.fit_transform(vectorizer.transform(testkey))
      y_pred=np.append(y_pred,clf.predict(testset))
      y_true=np.append(y_true,np.array(brand_dict[item['brand']]))

print("y_prediction:",y_pred,"y_true:",y_true)
print("F1 score:",f1_score(y_true, y_pred, average="macro"))
print("Precision score:",precision_score(y_true, y_pred,
average="macro"))
print("Recall score:",recall_score(y_true, y_pred,
average="macro"))
```

最後結果

```
y_prediction: [ 2.  7.  1.  2.  5.  6.  3.  2.  1.  1.  2.  5.  3.
5.  2.  7.  3.  4.
  3.  1.  1.  5.  6.  7.  3.  5.  2.  7.  7.  3.  2.  7.  3.  4.
7.  1.
  1.  7.  5.  1.  1.  4.  3.  4.  2.  5.  1.  1.  5.  5.] y_true:
[ 3.  7.  1.  2.  5.  6.  6.  2.  7.  1.  2.  5.  3.  1.  2.  7.
3.  4.
  3.  1.  7.  5.  6.  2.  3.  5.  2.  1.  2.  3.  2.  2.  3.  4.
7.  1.
  1.  4.  5.  1.  2.  4.  3.  2.  2.  2.  1.  1.  2.  5.]
F1 score: 0.726400979972
Precision score: 0.752705627706
Recall score: 0.741666666667
```