Programming Assignment 2

姓名：黃奕瑋 學號：R14725055

1. 執行環境

- Jupyter Notebook (.ipynb 檔案)

2. 程式語言

- Python == 3.10.7
- nltk == 3.9.1
- pandas == 2.2.2

3. 執行方式

- 所需套件：pip install nltk==3.9.1 pandas==2.2.2
  1. 用支援 Jupyter 格式的 ide 開啟 pa2.ipynb (我是使用 vscode)
  2. 點擊 Run All 執行所有 cell
  3. 程式會在 output 資料夾產出 dictionary.txt 以及 *.txt (* 為 1～1095)

4. 處理邏輯說明

Q1 建立 dictionary.txt：

1. 用 os.walk() 讀取資料夾內所有文件，並將每份文件 append 進 list 中。
2. 將每份文件依序執行 tokenization，並存成 list (list 格式：[doc1[term1, term2, ...], doc2[term1, term2, ...], ...])，tokenize 流程如下
   - I. 文字轉小寫
   - II. 去除特殊符號，只保留 a~z 以及空格
   - III. 用空格斷詞
   - IV. 移除停用字，停用字存放在 stopwords.txt，是從 nltk.corpus 的 stopwords 下載，並加上一些我自己認為的停用字。
   - V. Stemming，用 nltk 的 PorterStemmer()
3. 計算 df 值
   - I. 統計全部文本中出現過的 term
   - II. loop 每個 term，每個 term 比對全部 document
   - III. if term in doc: df++
   - IV. 得到每個 term 的 df value

4. 將 term 以及其 df 存成 pandas dataframe，並依字母順序 sort
5. 將他們依序編號，產出 t_index 值
6. 存成 dictionary.csv

Q2 計算 tfidf：

1. 依序為每個 doc 計算裡面每個 term 的 tfidf
2. 計算 文件詞頻 以及 文件長度
3. loop 該 doc 內每個 term
4. 計算 tf = (term 在該文件出現次數) / (文件長度)
5. 計算 idf = math.log10 (文件總數 / df)
6. tfidf 正規化，tfidf_i = tfidf_i / sqrt(sum(tfidf_i ** 2))
7. 存成 txt 檔

Q3 計算 cosine similarity：

1. 讀取兩個 txt 檔
2. cosine similarity = (X · Y) / (|X| * |Y|)
3. 計算 (X · Y)
    I. 用 pandas.merge() 找出交集的 term
    II. 將兩份文件的相同 term 的 tfidf 相乘後加總
4. 計算 (|X| * |Y|)
    I. |X| -> sqrt(sum(tfidf_X ** 2))
    II. |Y| -> sqrt(sum(tfidf_Y ** 2))
5. 最後帶入 cosine similarity 公式即可完成