1. What kind of pre-processing did you apply to the document data or question text? Additionally, please discuss how different preprocessing methods affected the performance of the models?

   我希望僅留下有意義的字來做接下來的比對，所以對文本做了以下處理。

   - 把所有的字元都轉成小寫。
     以免出現僅僅是大小寫不一樣就沒有準確比對的情況，也避免影響詞彙庫的建立（如：apple 和 Apple 被當作不同字）。
   - 把不是英文字的符號（如標點符號）去掉。
     以免影響詞彙庫的建立（如：apple 和 apple,被當作不同字）及比對的瑕疵。
   - 去除一些比較沒有象徵性的字。
     利用了 nltk 套件中的 stopwords，它提供的 stopwords 包含許多代名詞，如：”I”、”me”、”you”等等。
     另外，由於資料是 HTML，Document 中有許多與內容無關的標頭，如 h1、h2、td 等等，這些也加入 stopwords 一起去除。
   - 建立 tokens
     把一個一個關鍵字存起來，原本的 Document 會變成關鍵字的字串陣列。

   一開始只轉小寫和去除標點符號跑出來的結果就不錯了，但是把其他預處理加上去對於模型預測結果的提昇是期許他能夠更加精準的。

2. Please provide details on how you implemented the vector model and BM25.

   ## Vector model：

   A. 讀取資料集、訓練集、測試集（使用了 pandas）。

   B. 提取 tokens（做預處理並使用 nltk.tokenize 的 word_tokenize）。
      對資料集的 Document_HTML 和訓練集及測試集的 Question 進行相同的預處理，統一比對標準，預處理描述於第 1 個問題。

C. 建立詞彙庫。
把資料集、訓練集的 tokens 中所有不同的字提取出來。
實作上用了 dictionary，如{“word”:index}，方便之後建立向量等，找到該字的位置。

D. 計算 idf_vector。
一起計算各個 document 的 idf，並且把它組成一個 vector 方便之後使用。

E. 把 document 和 test 的 tokens 轉成向量（計算 tfidf_vector）。
計算 tf_vector：算各個詞彙庫中的字在 document 中有幾個。
用 tf_vector 和 idf_vector 相乘，即為 tfidf_vector。

F. 用 Vector Model 取前三高相似的 Document ID。
用 cosine_similarity 計算 document 和 test 的 tfidf_vector 來比較相似度，之後 sort 分數，提取分數前 3 高的 document 其 Document ID。

G. 輸出結果為 csv。

BM25：

A 到 D 步驟與 Vector model 相同。

E. 用 BM25 公式計算，取前三高相似的 Document ID。
BM25 公式計算：

$$\text{score}(D,Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i,D) \cdot (k_1+1)}{f(q_i,D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

計算 tf_vector（上圖公式中的 f(q,D)即為 term frequency）：算各個詞彙庫中的字在 document 中有幾個。
計算 query_vector：算各個詞彙庫中的字在 query 中有幾個。
算 avgdl：所有 document 的平均長度。
自訂 k1、b：通常 1.2 < k1 < 2、0 < b < 1。我取 k1 = 1.5、b = 0.75。

以 BM25 公式算出分數後，進行 sort，提取分數前 3 高的 document

其 Document ID。

F. 輸出結果為 csv。

嘗試調整 IDF 的算法，把 log(N / (ni + 1))，+1 的目的是為了避免分母為 0，之後改成 BM25 算 IDF 的方式 log(N – ni + 0.5 / (ni + 0.5) +1)，效果從 Kaggle 上面來看是有提升的。

3. Compare the strengths and weaknesses of the vector model and BM25. What factors might account for the differences in their performance?

使用 vector model，運算速度上面比 BM25 快上許多，但精確度來說，比 BM25 還低上一點，不過還是有不錯的表現。

使用 BM25，由於需要不停地計算，非常吃效能，時間是一個大考驗，但只要自訂參數選擇是適當的，成果出來基本上是讓人滿意的，。

如果以少量資料運行來看我覺得 BM25 會是相對好的選擇，反之用 vector model 可能會較為適合。

相較 BM25，vector model 非常依賴詞出現的頻率，這可能導致一些相似度的誤判，所以當 query 較短時，使用 BM25 的精確度比較容易看的出來跟 vector model 的差別。