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。

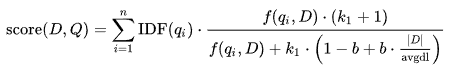
1. Please provide details on how you implemented the vector model and BM25.

**Vector model：**

1. 讀取資料集、訓練集、測試集（使用了pandas）。
2. 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。
3. 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。
4. 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。
5. 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。
6. 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。
7. 輸出結果為csv。

**BM25：**

A 到 D步驟與 Vector model相同。

1. 用BM25公式計算，取前三高相似的Document ID。  
   BM25公式計算：  
   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。
2. 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。

1. 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。