# HW3 Report

R07725021 資管碩1 洪靖雯

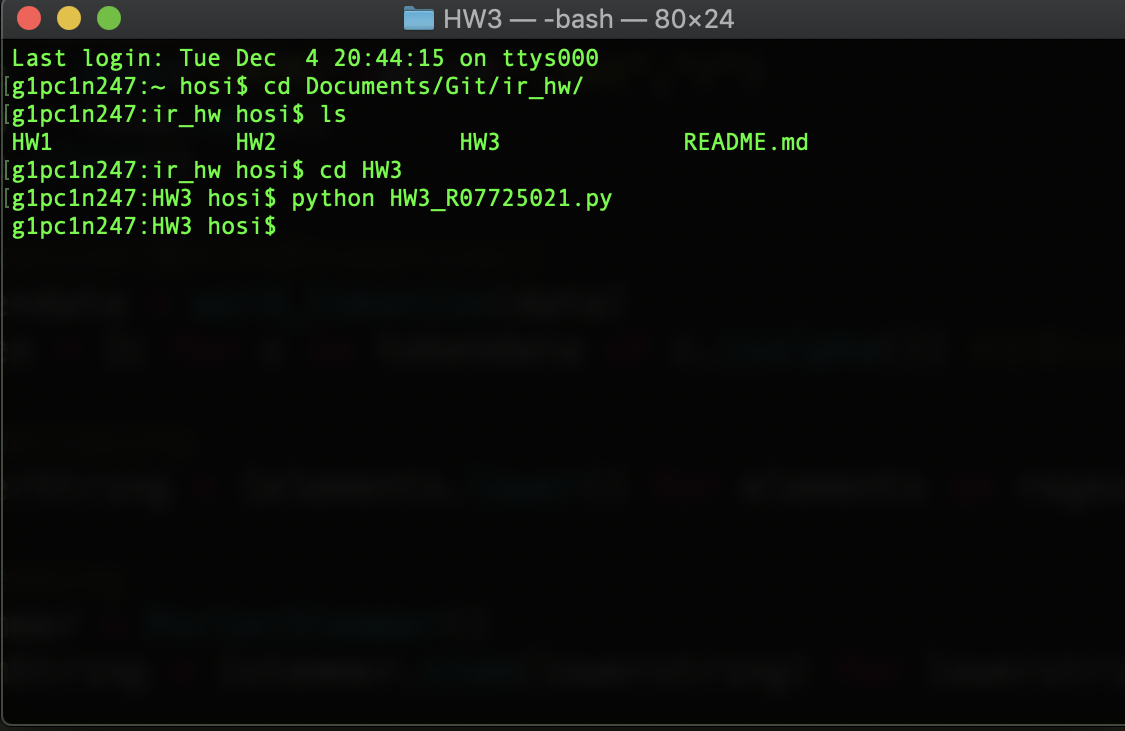
## 1.執行環境

* Sublime
* Cmder

## 2.程式語言

* Python 3.6.6

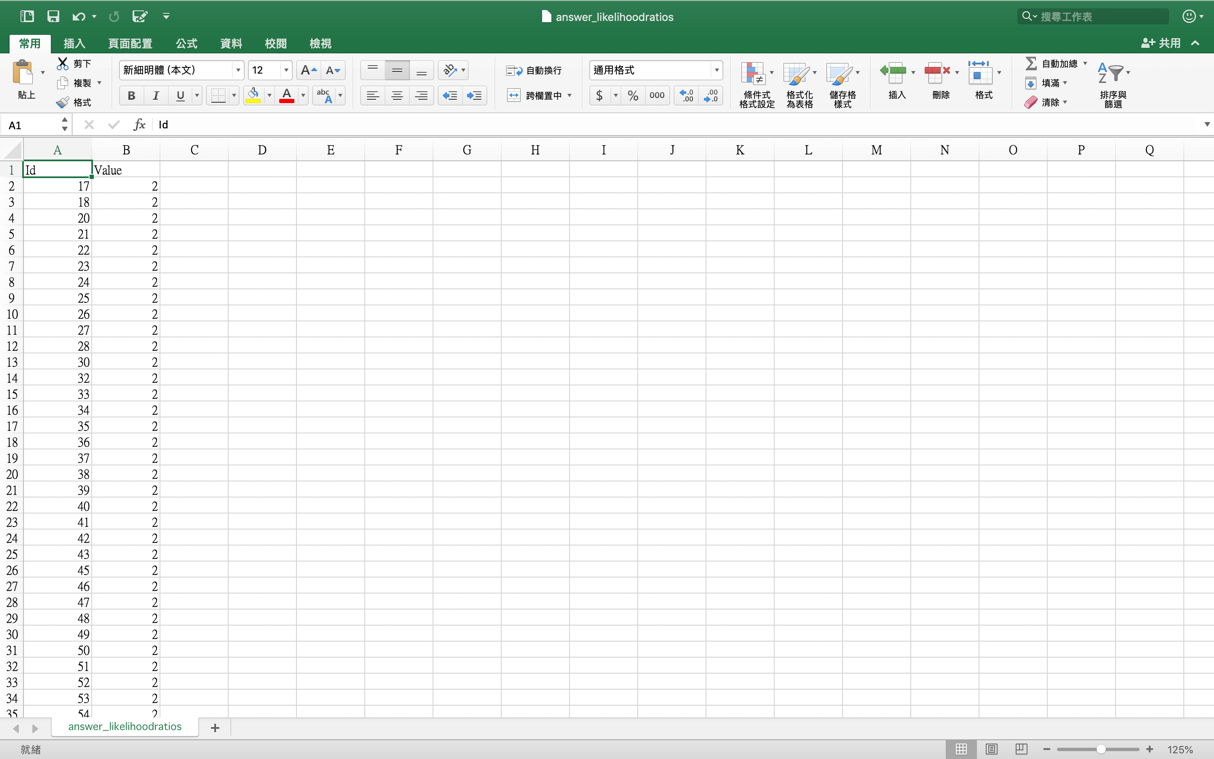
## 3.執行方式

* 套件: nltk
  + 有另外用conda 4.5.8架虛擬環境安裝，直接用在system裡應該也可行
* 編譯方式: 使用python HW3\_R07725021.py即可  
  
* 輸出結果

1. answer\_xxx.csv (總共有三個檔案，分別是用chi-square, likelihood ratios和expected mutual information(EMI) feature selection做出的結果，預測出來效果略有差異)



* answer\_likelihood.csv（目前效果最好），共900個testing data



## 4.作業處理邏輯說明

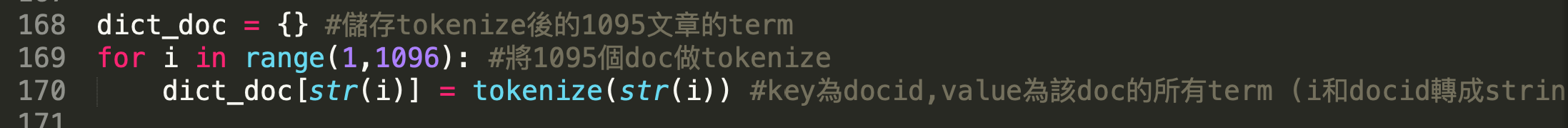
1. **HW3\_R07725021\_1.py**

* Import必要的套件

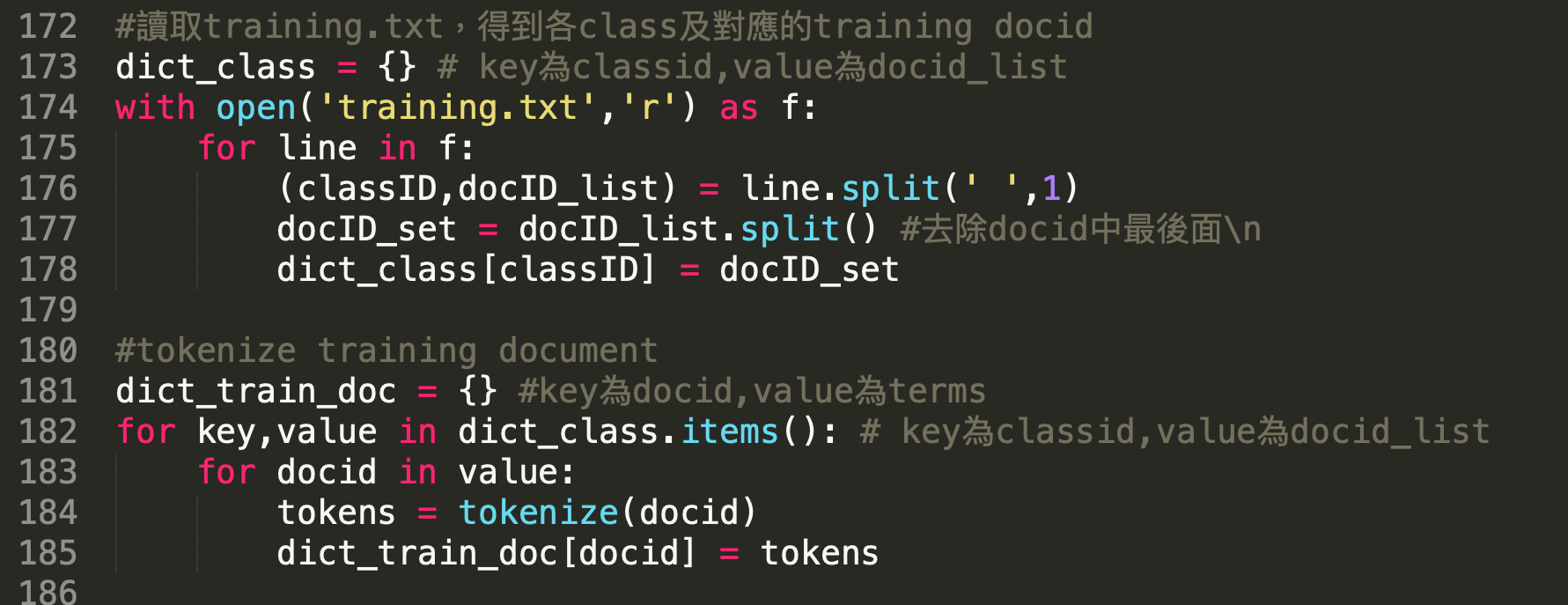


* 先將1095個documents做tokenize(運用之前HW1的方法)，運用dictionary的結構來儲存，key為docid, value為該文章的terms。

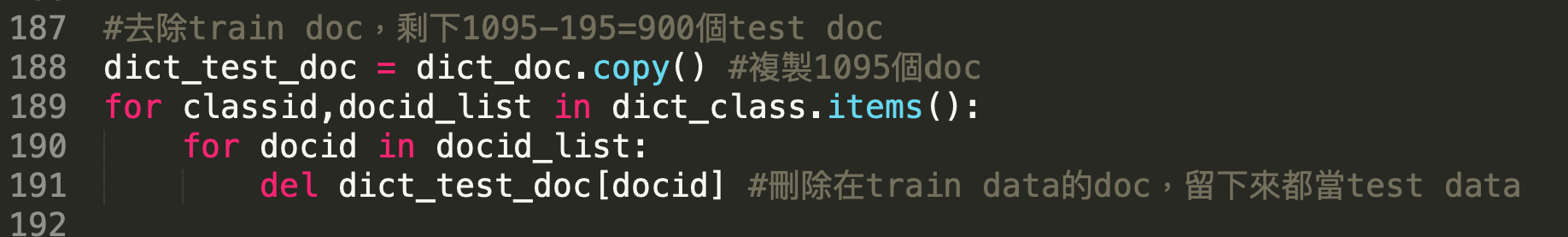




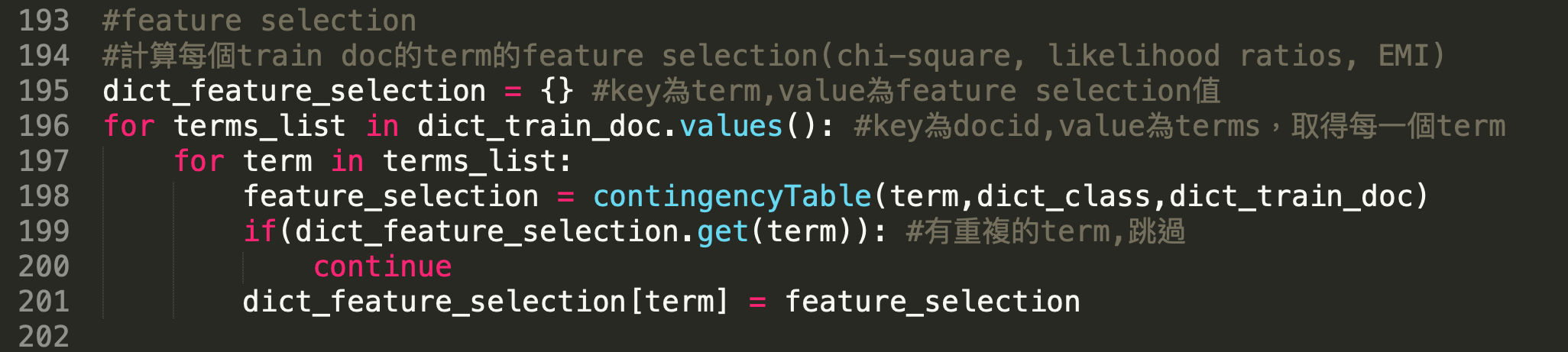
* 讀取training.txt得到各class及對應的training doc，並將training data做tokenize！一樣用dictionary結構儲存，dict\_class儲存traing.txt，key為classid, value為15個docid; dict\_train\_doc儲存tokenize後的training data，key為docid, value為terms。



* 除去training data,剩下的為testing data,也做tokenize用dict儲存，key為docid, value為terms。

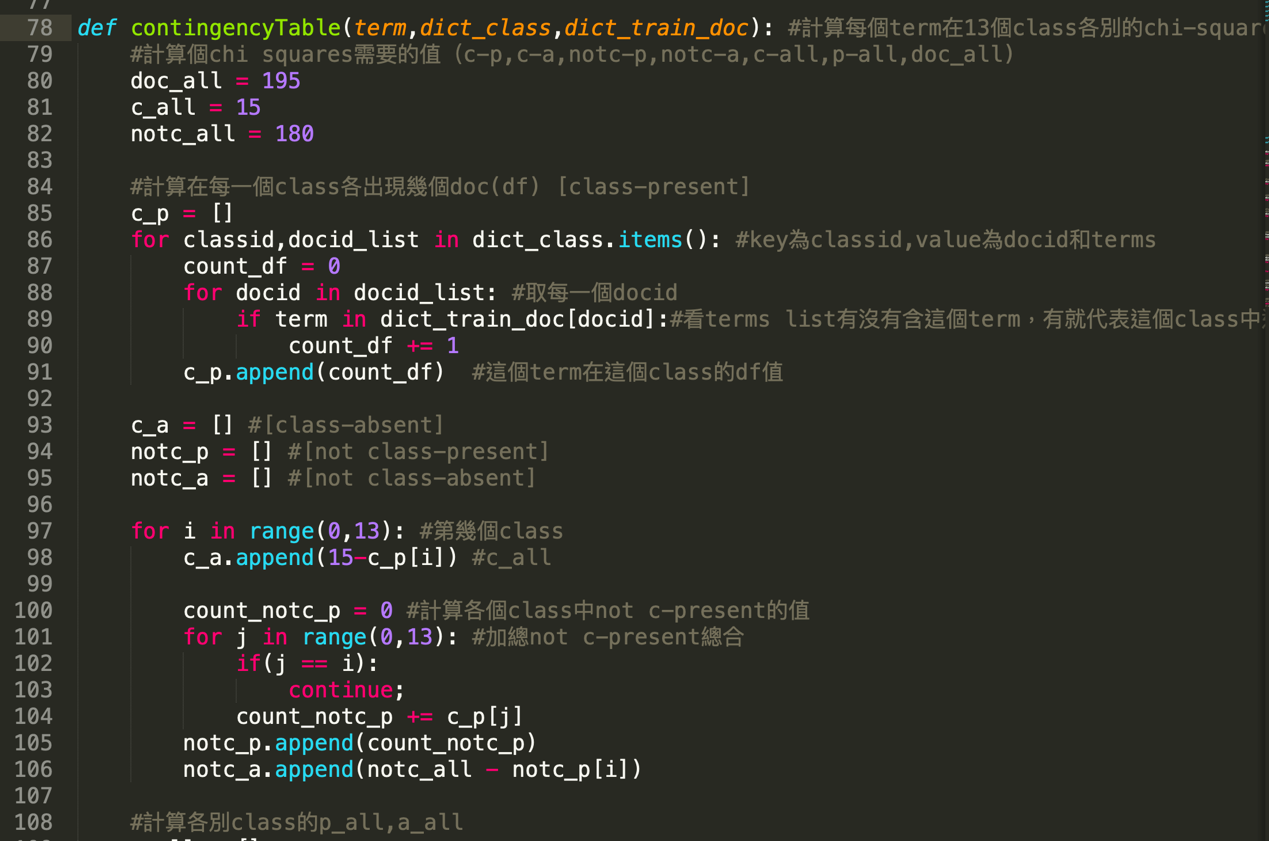


* 將training data做feature selection，計算每個term的數值，用dict儲存，key為term, value為feature selection值



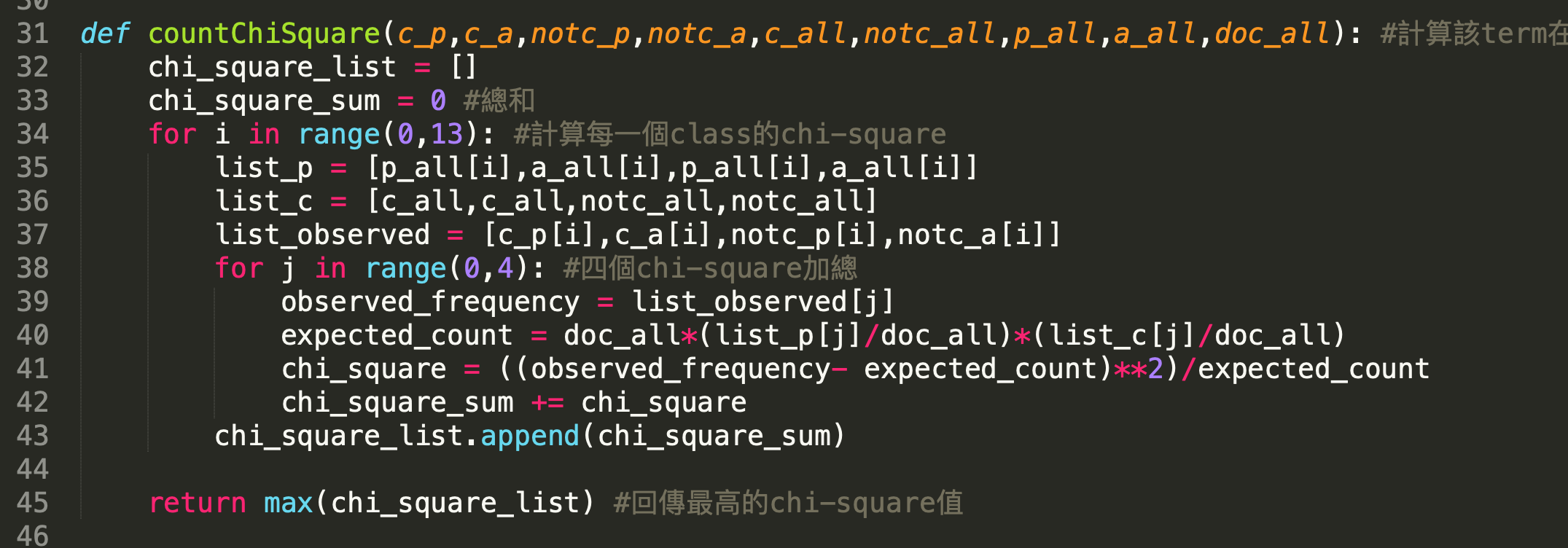
* + 先算出contingency table

有些值像是全部文章數為1095篇, 13個class每個class全部文章為15篇, 非該class的文章為195 – 15 = 180篇; 其他皆以list來儲存個別class的每個term的present和absent儲存，以及 on class和not on class也儲存。後半則以選擇feature selection方法來做呼叫。

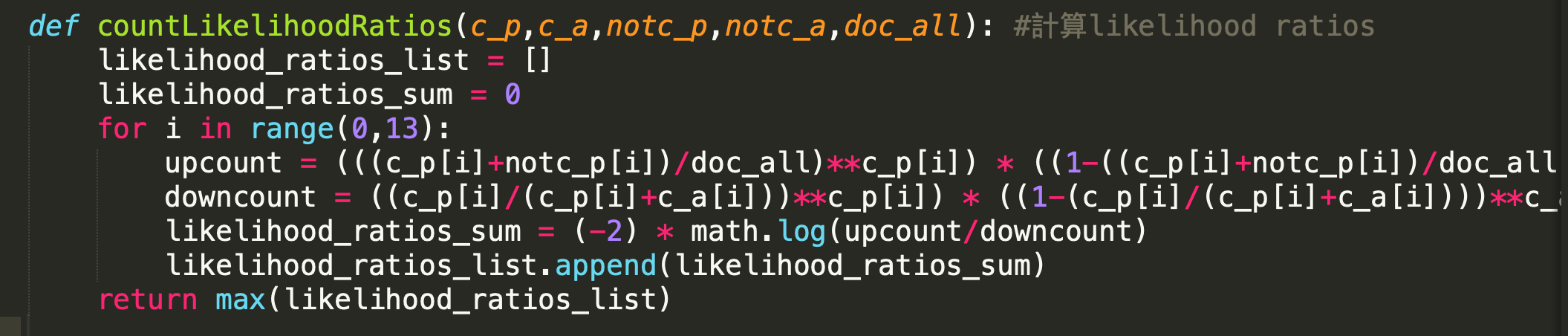




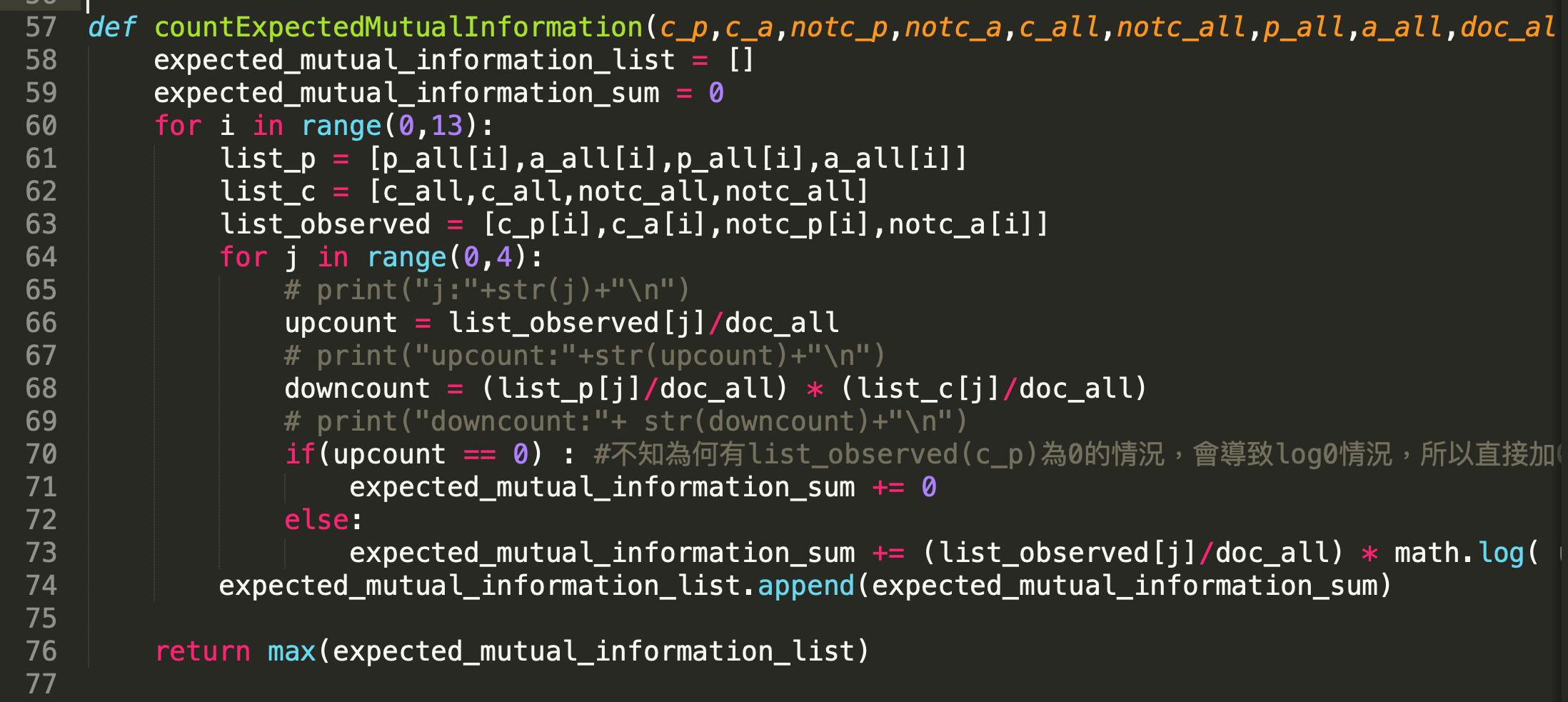
* + Chi-square feature selection



* + Likelihood ratios feature selection

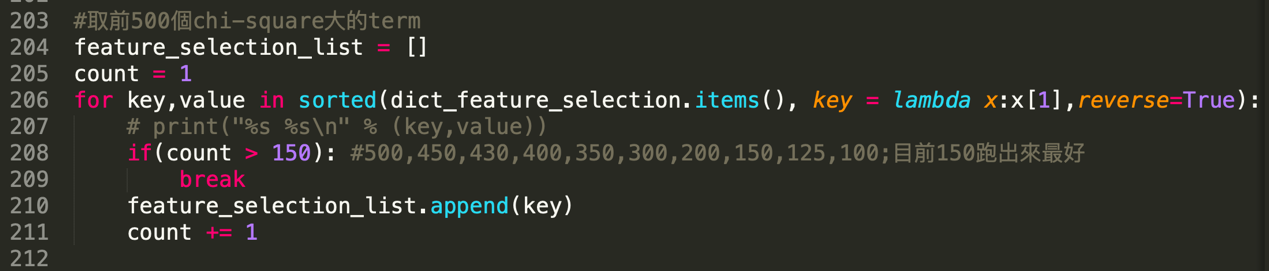


* + Expected mutual information

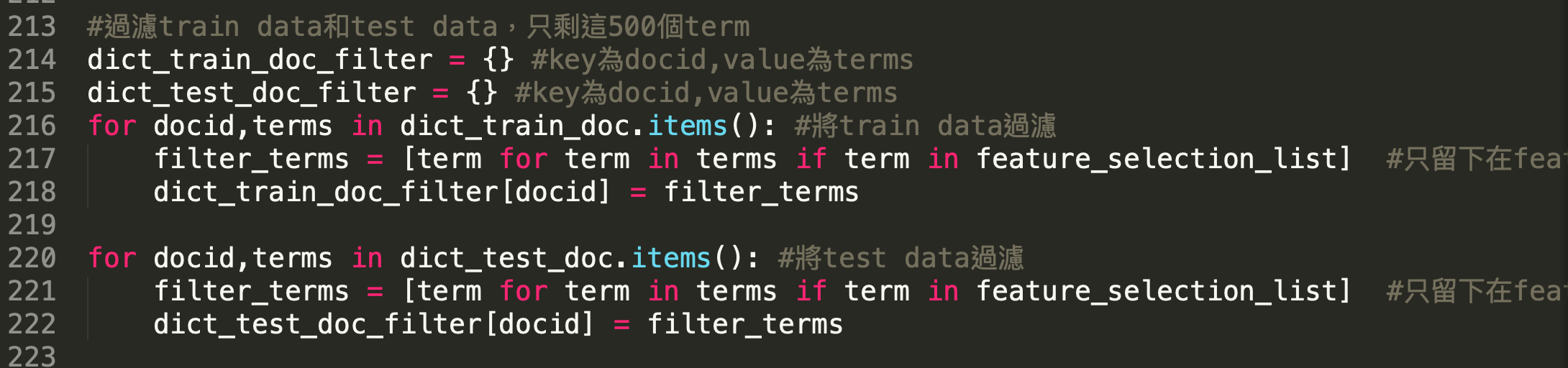


以上述任一種feature selection方法，計算出每個terms在各13個class的數值，選擇最大的數值回傳，以做後續的比較。

* 計算出term的數值後，排序dict，取前500個數值高terms數來做vocabulary。（目前發現terms取150個預測準確度最高）



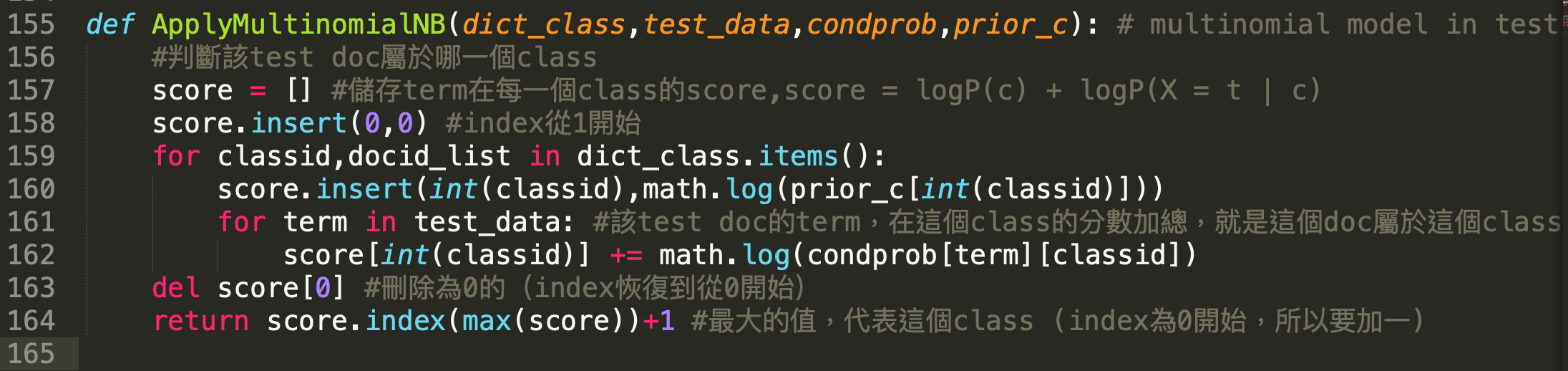
* 用這些feature selection後的term，來將training data和testing data做過濾，只剩下有出現在500個feature selection terms裡的，排除沒看過的字。



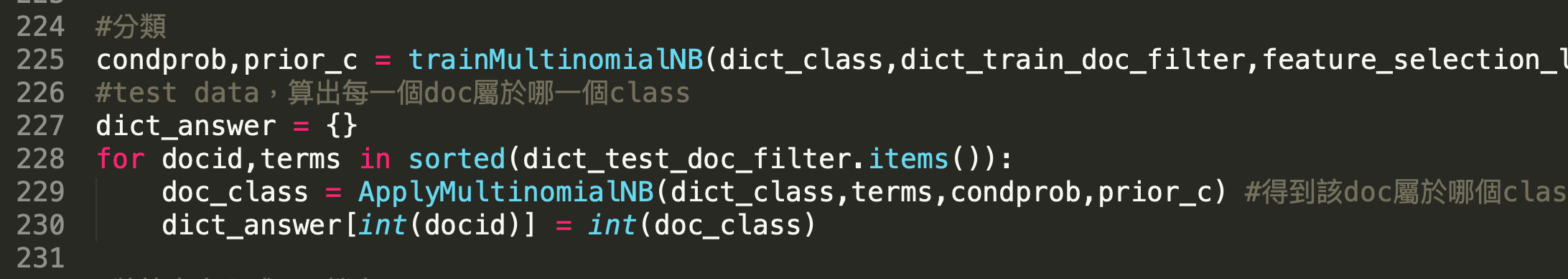
* 接著以multinomial Naïve Bayes model來做分類
  + Training phase, 算出condprob, prior\_c



* + Testing phase,丟入testing data，為每一個test doc做分類，得出它在13個class中數值最高，便把它分配在這個class。



* + 最後回傳test docid和對應的classid，以dict儲存，key為docid, value為classid。



* 最後將答案輸出成csv檔

