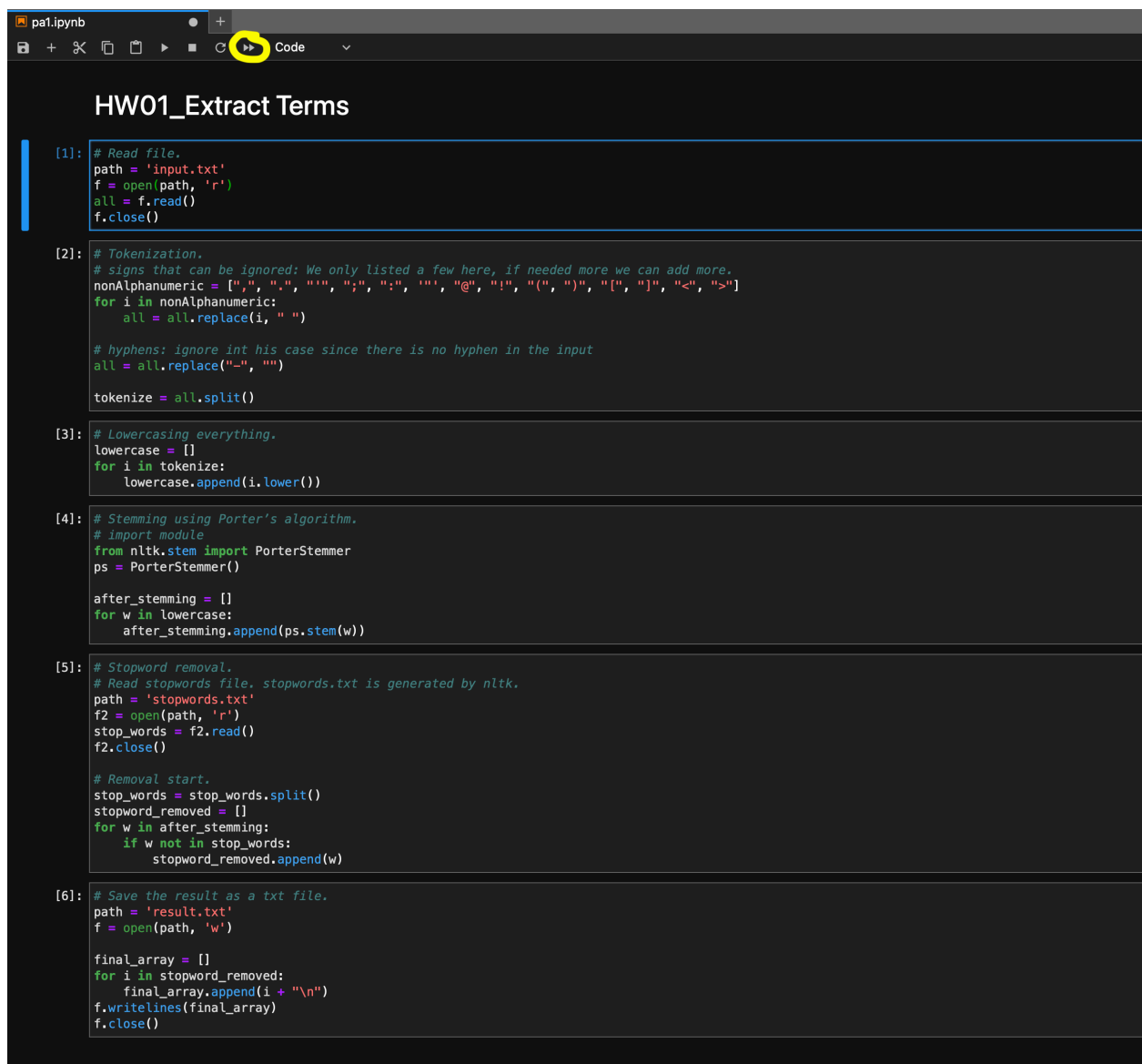


1. 執行環境：Jupyter Lab
2. 程式語言：Python 3.10.5
3. 執行方式：

使用 pa3.ipynb 檔

- (1) 使用 Jupyter Lab 或 notebook 開啟 pa3.ipynb 檔
- (2) pip3 install nltk 用於 Porter's Algorithm
- (3) 將整個檔案全部執行即可



```
pa1.ipynb
HW01_Extract Terms

[1]: # Read file.
path = 'input.txt'
f = open(path, 'r')
all = f.read()
f.close()

[2]: # Tokenization.
# signs that can be ignored: We only listed a few here, if needed more we can add more.
nonAlphanumeric = [" ", ".", ":", ";", ",", "!", "(", ")", "[", "]", "<", ">"]
for i in nonAlphanumeric:
    all = all.replace(i, " ")

# hyphens: ignore int his case since there is no hyphen in the input
all = all.replace("-", " ")

tokenize = all.split()

[3]: # Lowercasing everything.
lowercase = []
for i in tokenize:
    lowercase.append(i.lower())

[4]: # Stemming using Porter's algorithm.
# import module
from nltk.stem import PorterStemmer
ps = PorterStemmer()

after_stemming = []
for w in lowercase:
    after_stemming.append(ps.stem(w))

[5]: # Stopword removal.
# Read stopwords file. stopwords.txt is generated by nltk.
path = 'stopwords.txt'
f2 = open(path, 'r')
stop_words = f2.read()
f2.close()

# Removal start.
stop_words = stop_words.split()
stopword_removed = []
for w in after_stemming:
    if w not in stop_words:
        stopword_removed.append(w)

[6]: # Save the result as a txt file.
path = 'result.txt'
f = open(path, 'w')

final_array = []
for i in stopword_removed:
    final_array.append(i + "\n")
f.writelines(final_array)
f.close()
```

4. 作業處理邏輯說明：

A. Split Training and Testing data by the given training.txt.

B. Tokenize all doc:

(1) Read file: Use `open()` read only to read file and `close()` to finish.

(2) Tokenization: Craft a list with signs we want to split as `nonAlphanumeric`. Then, replace the signs with blank and eliminate the hyphens and dots among words. Finally, use `“.split()”` to split string in to a list of words whenever the next element is a `“\n”` or a blank.

(3) Remove Digits: Remove all digits in the document.

(4) Lowercasing everything: Use `“.lower()”` to lower case everything.

(5) Stop word removal: Read file `“stopwords.txt”`, which is generated by `“nltk”` (approved by the professor). Whenever the word before stemming is not a stop word, we keep it to the next section.

(6) Stemming using Porter's algorithm: Use the api `“from nltk.stem import PorterStemmer”` allowed by the professor for stemming words.

(7) Save the results in `all_doc` array.

C. Feature Selection: By using X^2 test.

(1) Extract all vocabularies from the training data.

(2) Count number of training documents.

(3) Compute the table 13 x 2 table for each term. Each columns are occurrence and non-occurrence of the term. Each row corresponds to one of the classes.

(4) Calculate the margin number of the row and column.

(5) Finally, count chi-square statistic by using the formula in the lecture note. And after testing different values, we recommend to select the 200 terms with the largest chi-square statistic.

D. Multinomial NB Classifier — Training:

(1) Calculate the prior probability of each class.

(2) Calculate the frequency of terms selected in the previous section in documents of each class.

(3) Calculate the conditional probability with add-one smoothing by the result of (2).

E. Multinomial NB Classifier — Testing:

(1) Calculate each classes log score of the final probability by adding each class's log of prior probability.

(2) Adding log of conditional probability of every term. Terms not in the selected vocabulary by training section are ignored.

(3) Pick the highest log probability score's class as the class of the testing document.

F. Output output.csv: Output the results of testing data to output.csv in the same structure as the sample given on Kaggle.

Result:

Kaggle score: 0.98888, results can be generated by pa3.ipynb