Machine Learning HW#4　R04942131_蔡銘穎

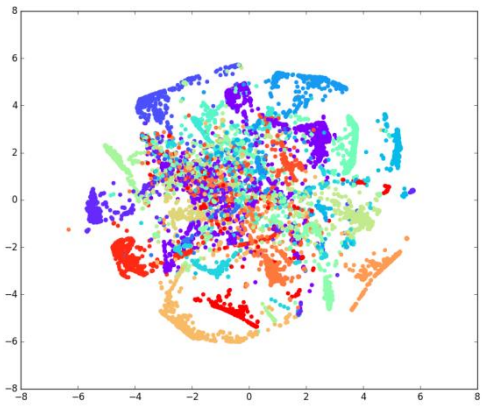
1. Analyze the most common words in the clusters. Use TF-IDF to remove irrelevant words such as “the”.

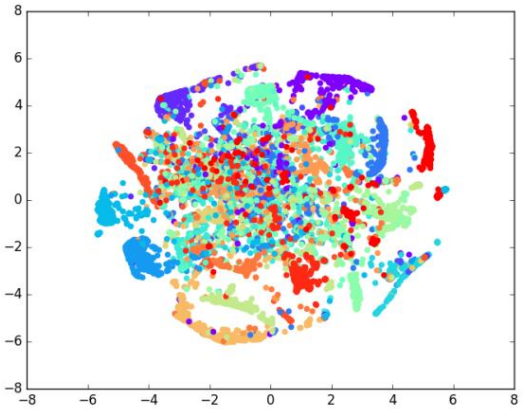
由於我們並不知道文件的分類依據為何，所以我在一開始試了每 N 行視為一份文件，但 performance 和預期的一樣很差，所以接著移除一些 Stopwords，這裡除了 sklearn 內建的 stopword 之外，我也另外建立了一份，並且觀察 docs.txt 後，也試著移除標點符號、數字，以及以”空格”、”Tab” 開頭的文件，理由是這些可能是網址、程式碼等等，我認為是不必要的資訊。

因分類數目為 20，故以下我列了分數排名前 20 高的單字

1.	hkey	6.linq	11.root	16.vista
2.	asmx	7.dataset	12.expose	17.setup
3.	datatable	8.environment	13.studio	18.machine
4.	hateful	9.windows	14.visual	19.cross
5.	registry	10.hive	15.typed	20.entries

2. Visualize the data by projecting onto 2-D space. Plot the results and color the data points using your cluster predictions. Comment on your plot. Now plot the results and color the data points using the true labels. Comment on this plot.





圖左為我的分類結果，其實可以看到外圍的部分分的非常乾淨，代表我們的分類能力是相當不錯的，而在中心的部分之所以會亂成一團的原因是，因為我們將一個高維的向量投影到低維所導致，因為 20 個 cluster 並不能僅由兩個維度去充分表達，所以才會有這種現象。

圖右為使用 True label 的結果，讓我比較訝異的是他的外圍並沒有像我一樣是單一顏色，有可能是我的 PCA 的結果並不是非常理想的原因所導致。

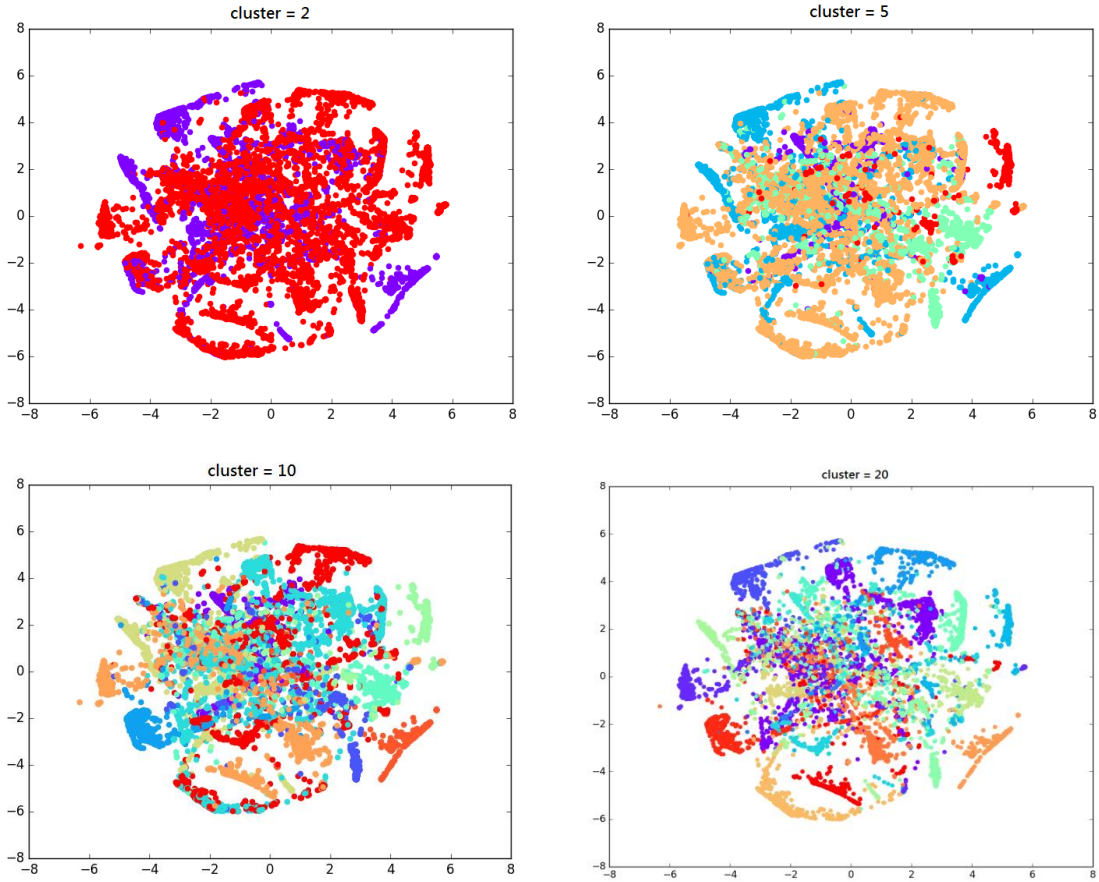
3. Compare different feature extraction methods.

我試了非常多種 feature extraction，一開始是每 10 行取出一份文件，在用 bag of word 的方式取得向量集，但其實並不曉得分類的依據，所以這種方法非常容易取到其他 title 的單字，所以 performance 並不佳。

再來也是每 N 行取一份文件，這裡的 N 是用試的試出來，最後結果為 N=20，但與第一個方法不同的是不做 bag of word，直接拿去做 TF-IDF，但其實我也不太曉得這種方式為什麼 perfornmance 會比第一種好非常多，可能的原因是第一種 bag of word 取到的 word 太差，或是第二種方式試了相當多不同的參數，碰巧得到了一個理想的答案。

4. Try different cluster numbers and compare them. You can compare the scores and also visualize the data.

以下我試了幾種不同 cluster 數目的結果：



做完結果我發現，做了 PCA 降為兩維之後，真的不太容易看出好的分類結果，可能是因為 document classification，這個議題確實要用比較高維的角度去分析，而無法像我們這樣用低維的角度去看出一些端倪。