Machine Learning Report

HW4 Unsupervised Clustering & Dimensionality Reduction

姓名：許義宏 學號：B02901053

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

首先，我針對全部的data去做common word 的idf的排序，找出全部的text裡最common 的word：

當有先使用stop words 過濾corpus時:

The most common 20 words in the cluster 是：[u'using', u'hibernate', u'wordpress', u'magento', u'excel', u'drupal', u'linq', u'spring', u'matlab', u'scala', u'oracle', u'file', u'sharepoint', u'ajax', u'haskell', u'visual', u'studio', u'bash', u'apache', u'qt']

而若沒有使用stop words時：

top 20 common word 則是 ：[u'to', u'in', u'how', u'the', u'with', u'of', u'and', u'for', u'is', u'on', u'from', u'do', u'using', u'an', u'can', u'hibernate', u'what', u'wordpress', u'magento', u'excel']

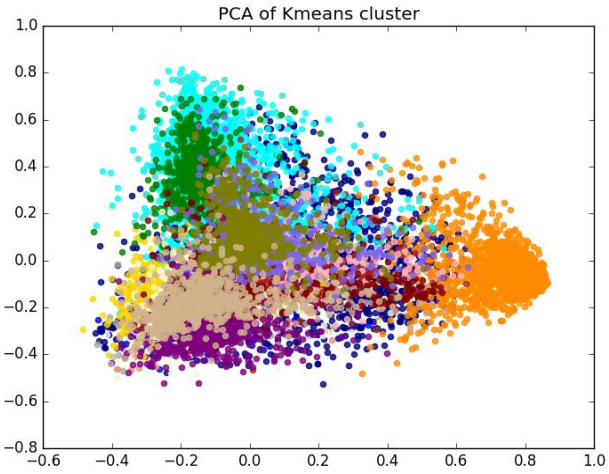
可以看出：1）在有使用stopwords的情況下，common words跟tags（紅字標出）的一致性是相當高的。2）再沒有使用的情況下，common word 的前幾名會被我們生活中常用的介系詞.....等詞彙所取代，到後面的排名才會出現tags 。我認爲這是符合預期的，因爲stop words 能成功的刪去一些不太重要的或着說是其實在語義上不太顯著的詞彙，而讓idf的功能能發揮更大的功效，去代表性的點出在各個title中哪樣的類別詞彙是在跨篇中的commom word。

若是針對已分群好的data去做tf-idf的話則各個group 前三common的word則爲：**group 0** [u'ajax', u'jquery', u'net'] **group 1** [u'magento', u'product', u'products'] **group 2** [u'qt', u'mac', u'os'] **group 3** [u'scala', u'java', u'class'] **group 4** [u'spring', u'using', u'bean'] **group 5** [u'hibernate', u'mapping', u'query'] **group 6** [u'excel', u'data', u'vba'] **group 7** [u'wordpress', u'page', u'post'] **group 8** [u'sharepoint', u'list', u'web'] **group 9** [u'svn', u'subversion', u'repository'] **group 10** [u'haskell', u'function', u'type'] **group 11** [u'matlab', u'array', u'matrix'] **group 12** [u'bash', u'script', u'command'] **group 13** [u'apache', u'rewrite', u'mod'] **group 14** [u'drupal', u'view', u'node'] **group 15** [u'oracle', u'sql', u'table'] **group 16** [u'file', u'files', u'bash']（應該要是cocoa) **group 17** [u'using', u'use', u'qt'] **group 18** [u'linq', u'query', u'sql'] **group 19** [u'visual', u'studio', u'2008']

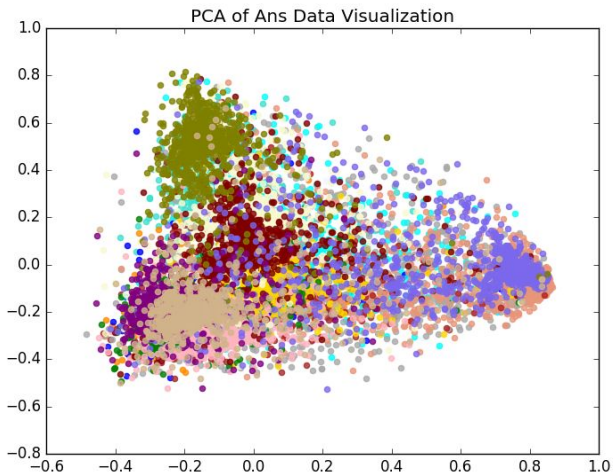
2. Visualize the data by projecting onto 2-D space. Plot the results and color the data points using your cluster predictions. Comment on your plot. Now plot the results and color the data points using the true labels. Comment on this plot.

圖一爲我將Kmeans Cluster過後的data使用PCA進行Dimension Reduction至2維後的圖，可以發現在角落的部分能夠清楚的看出群聚的現象，例如橘色的資料點；而中間的資料點則因爲投影的關係稍微混雜，但大體上能夠看的出群聚的分點

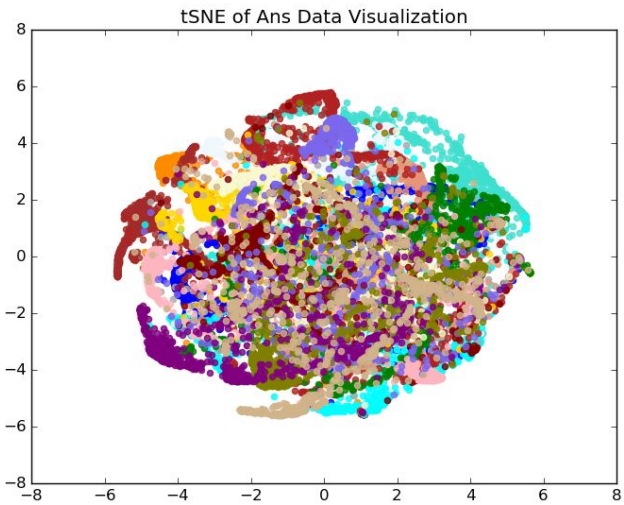
圖二爲使用正確的label爲資料點進行標記，相較於圖一看得出顏色較爲混雜。且可以注意到原先於圖一橘色鮮明而清楚的群集資料點在分類上的錯誤，因爲在圖二中實際上他的label大致是兩類，而我的Kmeans結果則有點把他們混在一起，不過在中間的資料點其實分類的正確性看起來不太差，也能夠相信這個在kaggle上的評分正確度不算差的準確性。



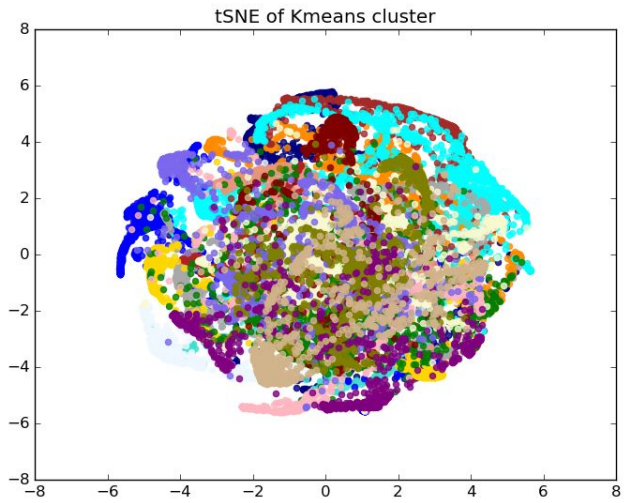
（圖一）



（圖二）



(圖三 用PCA先降至4維後使用tSNE降至2維的ans)



(圖四 用PCA先降至4維後使用tSNE降至2維的Kmeans)

3. Compare different feature extraction methods.

using k-means cluster to 20 groups

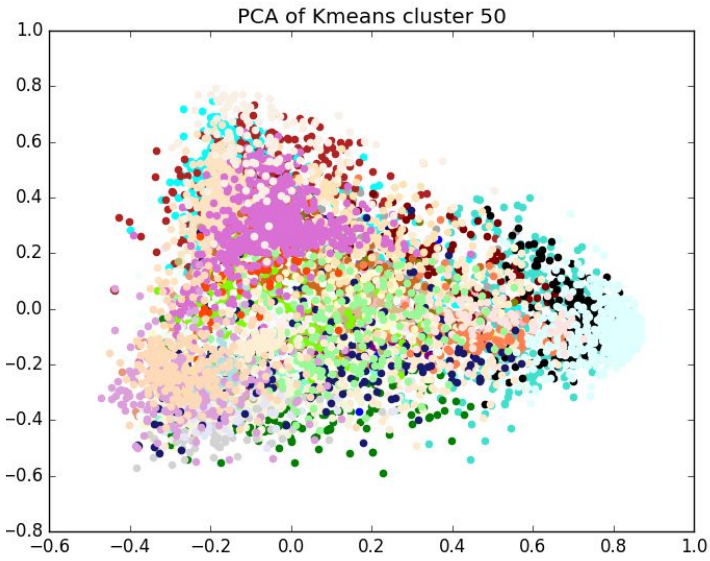
idf	Stopwords	LSA(降成20維)	Score
O	O	O	0.62779
X	O	O	0.54459

O	X	O	0.33306
X	O	X	0.29100
O	X	X	0.24070
O	O	X	0.23315
X	X	O	0.17303
X	X	X	0.13766

總結： 由於K-means會受到random的影響， 所以不全然以上的排序就等同於重要性，然而從把一個功能拿掉的三個去做比較（橘色標記者）可以明顯得看出來沒有idf的影響是最少的，其實因爲他的功能跟stopword有點重疊，而且效果還略比stopword差，這樣的推論也可從紅色標記的兩個分數差不多得到驗證。相對起來LSA的功能相對重要，因爲沒有他先做降維的動作只用K-means直接降至20維效果相當不好。

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

Cluster Numbers	Score
20	0.62779
30	0.75059
50	0.79176
70	0.80561
90	0.80421
100	0.80787



分析：

當cluster 夠多，在F-measure上得到的分數能夠得到提升，主要原因是因爲較不容易將兩者分到同一群裡，一定要資料點足夠相像才會cluster到相同的群。這樣的效果能有效的降低把0誤判成1 （也就是false positive)的值，因此能得到分數上的提升

從圖來看（以分成50群爲例），其實無法明顯的感受出分群的優化，只能看出集中的群集更加集中，但是整體來說點的顏色複雜度更高。