Machine Learning HW4

姓名：顏修溫
學號：r05922034
系級：資工所 碩一

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

下面列出每個 cluster 中出現頻率前十高的單字與出現次數。從下面結果可以發現同一個 cluster 中的單字確實具有滿高的相關性，舉例來說 Cluster0 裡頭出現的單字'ajax'、'jquery'、'javascript'這些都是和 ajax tag 滿相關的單字。

Cluster0	('ajax', 751), ('jquery', 113), ('using', 70), ('request', 56), ('net', 52), ('asp', 51), ('javascript', 49), ('php', 44), ('page', 42), ('form', 33)
Cluster1	('wordpress', 871), ('page', 124), ('post', 111), ('posts', 89), ('plugin', 78), ('category', 68), ('blog', 64), ('custom', 50), ('php', 46), ('theme', 42)
Cluster2	('scala', 804), ('java', 90), ('class', 70), ('type', 55), ('method', 52), ('using', 41), ('list', 37), ('way', 36), ('map', 34), ('actors', 33)
Cluster3	('sharepoint', 738), ('list', 155), ('web', 132), ('site', 85), ('2007', 66), ('custom', 55), ('document', 47), ('using', 39), ('services', 38), ('lists', 34)
Cluster4	('svn', 610), ('subversion', 250), ('files', 186), ('repository', 122), ('directory', 71), ('file', 69), ('working', 58), ('commit', 54), ('copy', 53), ('server', 52)
Cluster5	('hibernate', 857), ('mapping', 76), ('query', 69), ('using', 60), ('criteria', 53), ('key', 46), ('table', 44), ('object', 43), ('cache', 39), ('join', 39)
Cluster6	('magento', 876), ('product', 139), ('products', 85), ('add', 69), ('custom', 68), ('page', 57), ('admin', 51), ('order', 45), ('attribute', 43), ('category', 42)
Cluster7	('qt', 606), ('window', 70), ('application', 63), ('windows', 46), ('widget', 45), ('using', 36), ('creator', 27), ('gui', 23), ('library', 22), ('use', 20)
Cluster8	('drupal', 843), ('node', 97), ('content', 86), ('view', 83), ('form', 76), ('module', 75), ('views', 72), ('page', 65), ('menu', 60), ('custom', 58)
Cluster9	('spring', 829), ('bean', 78), ('using', 75), ('hibernate', 67), ('mvc', 67), ('security', 66), ('framework', 43), ('web', 41), ('use', 39), ('application', 38)
Cluster10	('matlab', 828), ('array', 81), ('matrix', 71), ('function', 65), ('image', 60), ('using', 59), ('plot', 48), ('file', 38), ('data', 29), ('vector', 27)
Cluster11	('using', 297), ('use', 163), ('way', 131), ('cocoa', 129), ('code', 100), ('best', 84), ('create', 79), ('file', 79), ('net', 71), ('multiple', 70)
Cluster12	('mac', 484), ('os', 343), ('cocoa', 186), ('osx', 118), ('app', 83), ('application', 78), ('10', 57), ('development', 46), ('leopard', 42), ('terminal', 39)
Cluster13	('visual', 719), ('studio', 692), ('2008', 142), ('project', 96), ('2005', 68), ('solution', 44), ('files', 41), ('projects', 41), ('build', 37), ('add', 35)

Cluster14	('haskell', 729), ('type', 167), ('function', 122), ('list', 62), ('error', 39), ('data', 32), ('types', 30), ('using', 28), ('functions', 25), ('string', 24)
Cluster15	('apache', 619), ('error', 92), ('rewrite', 92), ('mod_rewrite', 91), ('problem', 89), ('server', 89), ('php', 81), ('htaccess', 77), ('url', 67), ('redirect', 64)
Cluster16	('oracle', 761), ('sql', 198), ('table', 98), ('database', 80), ('query', 57), ('pl', 56), ('using', 48), ('stored', 43), ('procedure', 41), ('select', 38)
Cluster17	('excel', 875), ('data', 154), ('vba', 139), ('cell', 86), ('file', 77), ('using', 61), ('range', 51), ('macro', 48), ('sheet', 47), ('text', 47)
Cluster18	('linq', 853), ('sql', 160), ('query', 159), ('using', 97), ('xml', 43), ('list', 42), ('select', 42), ('group', 41), ('join', 38), ('data', 36)
Cluster19	('bash', 673), ('script', 252), ('file', 242), ('command', 119), ('line', 94), ('shell', 78), ('variable', 64), ('string', 53), ('output', 51), ('files', 47)

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.**

我是使用 PCA 對每個 title 的 feature vector 進行降維，接著用 matplotlib 畫圖。下方左圖 Fig.1 就是使用我自己 predict 的 label 去畫圖，下方右圖 Fig.2 是我用 true label 進行畫圖。可以發現若使用我自己的 prediction label 去畫的話，Fig.1 每個 cluster 感覺比較密集、群與群之間的界線會比較明顯。但是由於我的 prediction 不一定會是正確的，所以當我使用 true label 去畫圖時，會發現 Fig.2 圖中右半部分的 cluster 就失去群的結構，那就代表這半邊的預測是較不準的；而 Fig.2 的左半部分和 Fig.1 的左半部的分布滿相近的，代表幾個 cluster 的預測結果和真實情況頗接近。(兩張圖使用的顏色順序不一樣)

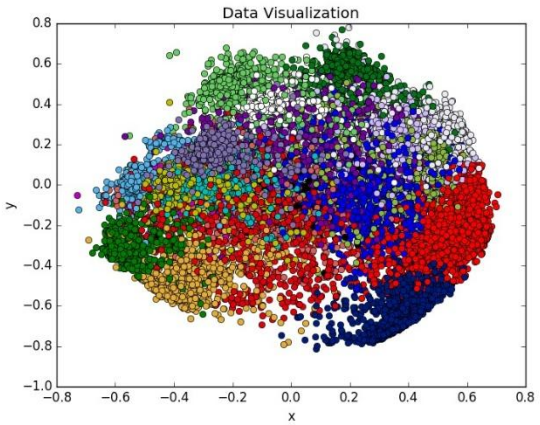

Fig.1 prediction label

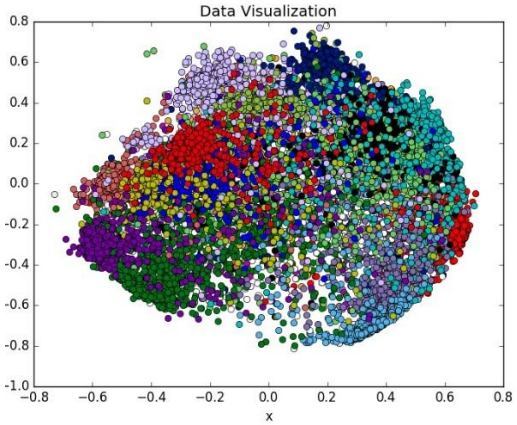

Fig.2 true label

3. **Compare different feature extraction methods.**

關於 Feature extraction 的部分，我有使用過以下幾種方法：Bag-of-Words、

TF-IDF、LSA、Word Vectors。在使用這些方法以前，我都會先用 nltk 把 stop words 給捨棄掉、把標點符號移除掉，並且把所有字元轉成小寫。以下比較五種方法的做法與效能。

- Bag-of-Words：使用 sklearn 的 CountVectorizer 計算每個 title 中單字的出現次數，接著對兩萬個 title 的 feature vector 進行 k-means cluster，在 kaggle 上得到的分數為 0.227(如果沒有濾掉 stopwords 的話，分數只有 0.141，所以先移除掉 stopwords 是滿重要的)。
- TF-IDF：使用 sklearn 的 TfidfVectorizer 去取得每個 title 的 TF-IDF vector，每個 title 的 vector 當中都含有 5236 維 feature。接著對兩萬個 title 的 feature vector 進行 k-means cluster，在 kaggle 上得到的分數為 0.29，確實有比 BoW 的效果還要好一些。
- TF-IDF + LSA：得到每個 title 的 TF-IDF vector 後，使用 LSA 將 vector 原本的 5236 維 feature 降到 20 維，降維後才對 vector 進行 k-means cluster。使用 LSA 降維後效果極為顯著，此方法可以得到 Kaggle best 分數 0.79217。
- Word Vectors：使用 gensim 的 word2vec。先用 docs.txt 這個檔案 train 出 wordvector，接著把 title 中每個單字的 word vector 相加取平均後就會得到此 title 的 feature vector，接著對兩萬個 title 的 feature vector 進行 k-means cluster，在 kaggle 上得到的分數為 0.487。
- Word Vectors + LSA：如上述利用 wordvector 得到每個 title 的 feature vector 後，使用 LSA 將 feature vector 降到 20 維，降維後才對 vector 進行 k-means cluster。此方法可以得到 Kaggle 分數 0.632。

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

我嘗試將 cluster number 設為 15、20、30、40 去比較，很神奇的是分數竟然會越來越高，甚至我後來去嘗試去分成 cluster number 為 50、60，分數都還是居高不下，是個十分有趣的現象。

Cluster number	15	20	30	40
Kaggle score	0.453	0.643	0.792	0.819
Visualize