1. **Analyze the most common words in the clusters**

[u'add', u'ajax', u'all', u'an', u'and', u'apache', u'application', u'are', u'as', u'bash', u'be', u'best', u'by', u'can', u'change', u'cocoa', u'code', u'create', u'custom', u'data', u'database', u'do', u'does', u'drupal', u'error', u'excel', u'file', u'files', u'for', u'form', u'from', u'function', u'get', u'haskell', u'have', u'hibernate', u'how', u'if', u'in', u'into', u'is', u'it', u'linq', u'list', u'mac', u'magento', u'make', u'matlab', u'multiple', u'my', u'net', u'not', u'object', u'of', u'on', u'one', u'or', u'oracle', u'os', u'page', u'php', u'problem', u'qt', u'query', u'scala', u'script', u'server', u'set', u'sharepoint', u'spring', u'sql', u'string', u'studio', u'subversion', u'svn', u'table', u'text', u'that', u'the', u'there', u'this', u'to', u'type', u'use', u'user', u'using', u'value', u'view', u'visual', u'vs', u'way', u'web', u'what', u'when', u'why', u'with', u'without', u'wordpress', u'xml', u'you']

Use TF-IDF to remove irrelevant words such as “the”.

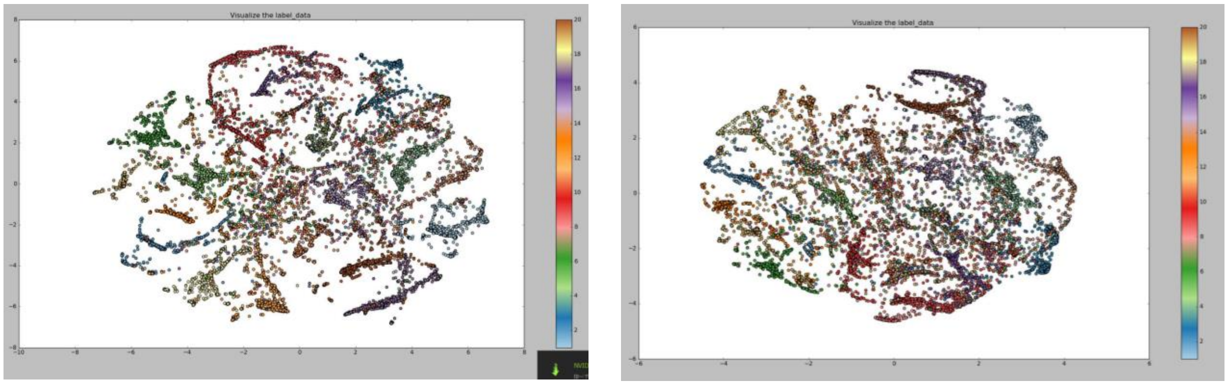
my_stop_words = frozenset(["get","file","an","in","with","and","the","can","you","for","of","from","is","what","on","to","not","how","do" ,"english","using"])

vector = CountVectorizer(analyzer = "word", lowercase = True,stop_words=my_stop_words ,max_features = 100,max_df = 0.4, min_df = 2,)

transformer = TfidfTransformer(norm = 'l2', use_idf =True, smooth_idf=True, sublinear_tf = False)

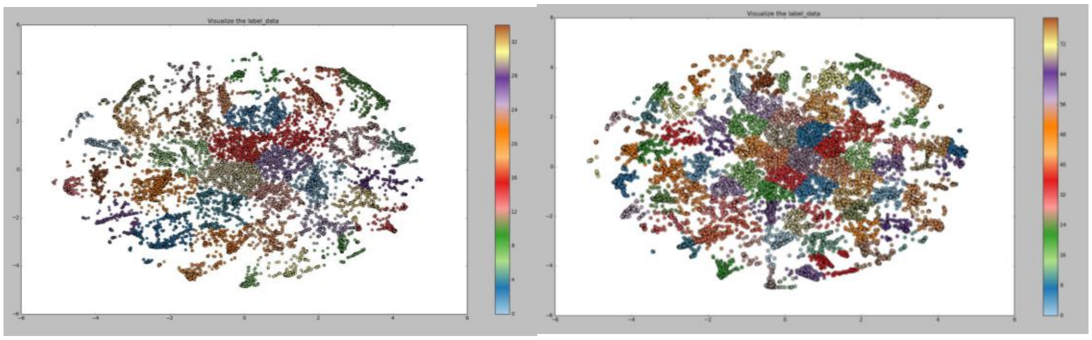
2 Visualize the data by projecting onto 2-D space. Plot the results and color the data points using your cluster predictions. Comment on your plot. Now plot the results and color the data points using the true labels. Comment on this plot. (1%)

Label_data



用 TSNE 去 visualize Label data 的 cluster，會發現可以看得出大致的分類，但是仍會有些地方沒有分乾淨，推估是因為我的 predict data，跟真實的 label 有些許不一樣，所以導致這樣的情形。

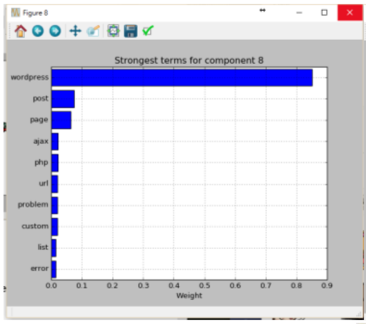
My predict data(dim = 35)

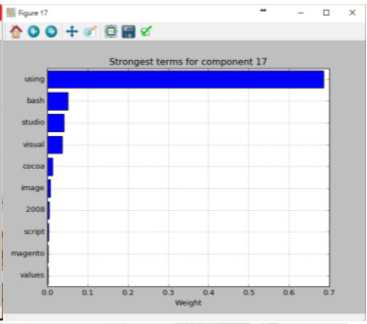


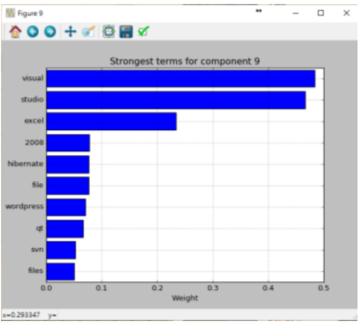
比 label data 的 visualize 乾淨許多，中間緊密，外面雖有些空隙兼具但仍可看清楚分群。而分群的形狀會隨 random_state、k_cluster 的數量以及 init 的參數而有些許不一樣。

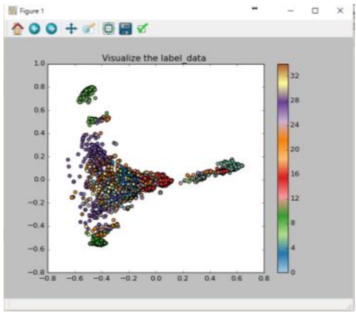
3 **Compare different feature extraction methods.** (2%)

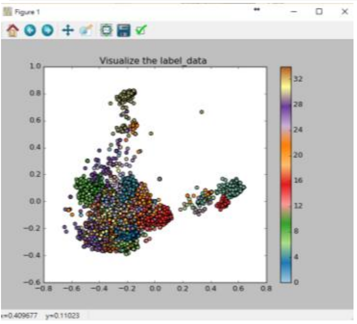
CountVectorizer + TfidfTransformer + default stop words + SVD + LSA







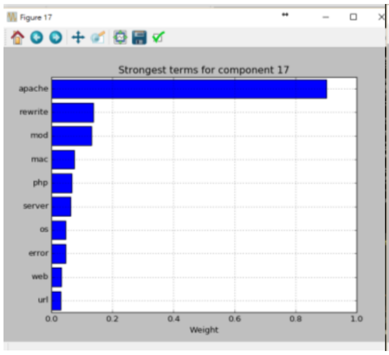


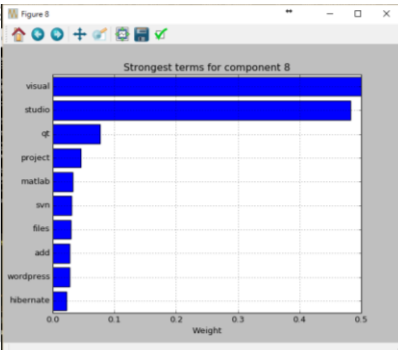


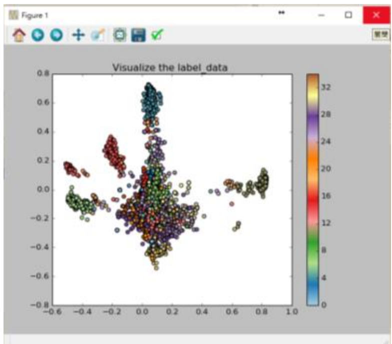
用 LSA 看起來有成功分群把我們要的分在同一類，但看起來會有重疊的情形是因為他的維度在 22 維，被壓在 2 維中，本來不會重疊的因此看起來重疊了。

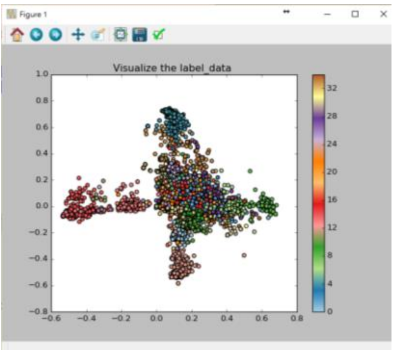
在預設的 stopwords 裡，會發現有一個最強的 component 是'using'，但它不是我要的分群之一，所以用自訂的 mystopwords 去把它去除掉。

CountVectorizer + TfidfTransformer + my stop words + SVD + LSA



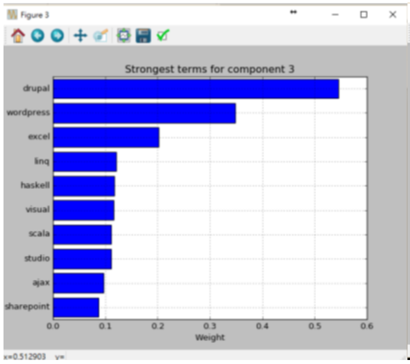


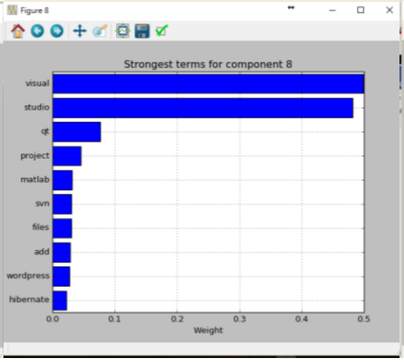


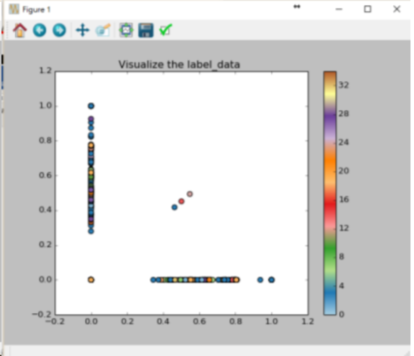


發現在裡面沒有'using'這個分群的存在了。

CountVectorizer + TfidfTransformer + my stop words

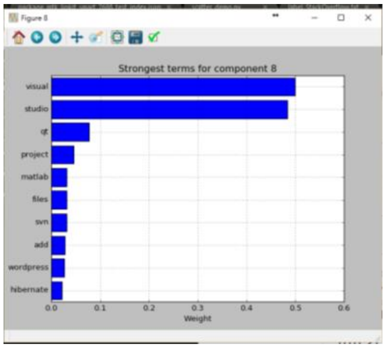


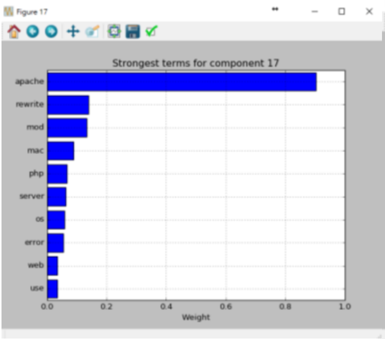


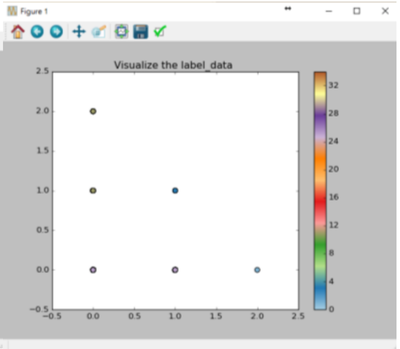


看起來很奇怪是因為還沒有用 LSA 去把相似的分群，所以會變成只有兩維，在各自 weight=0 時看另外一個 weight 的變化

CountVectorizer

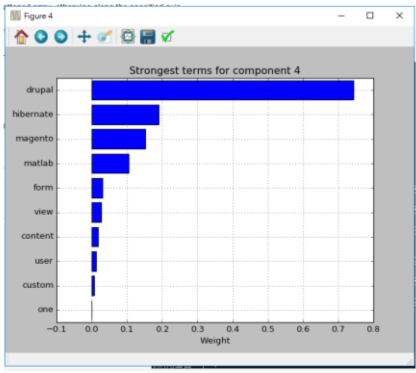


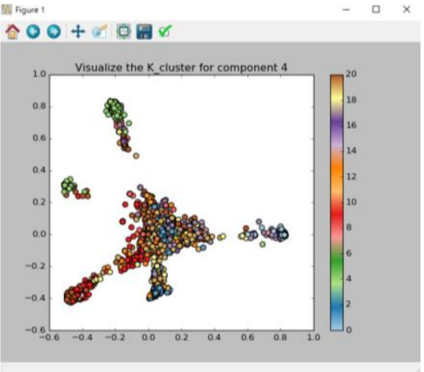




還沒有用 tfidf 去去除 stopwords 所以沒有 weight 存在，看不出字的重要性

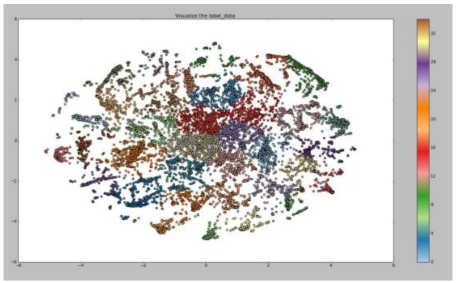
Autoencoder



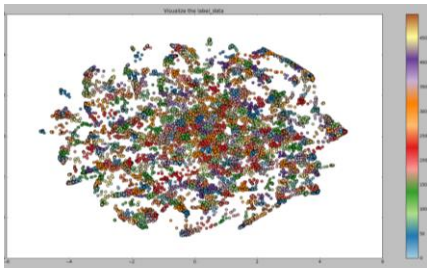


與 LSA 相近，因為它也是利用高維去把字做分群

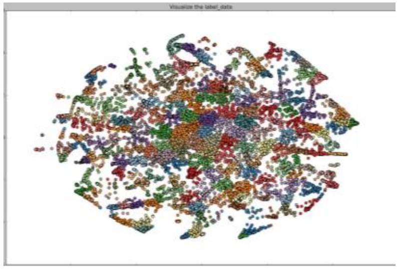
4 Try different cluster numbers and compare them. You can compare the scores and also visualize the data. (1%)



35clusters



500clusters



80clusters

不同的 kcluster 會有不同的分類，我們需要 20 個分群但不能剛好 20 個，有些要用來裝垃圾，但也不能太多個，會讓原本同一群的被分為不同群。所以太多群反而壞看起來凌亂。