R語言與資料科學導論

Final Project

Group 10

知乎

B02611013 生機四 黃薇甄 / B02704033 國企四 張景淵 / B02705002 資管四 周晅瑢 B02705006 資管四 高偉立 / B02705027 資管四 陳信豪 / B02705028 資管四 冷俊瑩

一、介紹 (Introduction)

　　知乎是大陸的一個社會化問答網站，產品形態模仿了美國類似網站Quora，也有點類似於台灣的 Yahoo 奇摩知識+，目前的註冊使用者已達5000萬，並擁有相當豐富的內容可以探勘。知乎網站架構下，有分為許多種主題，讓人們去發表文章、提問以及回答問題，而使用者們可以給這些內容物予以贊同、收藏和追蹤，全站目前累計產生了1000萬個問題，3400萬個回答及3500萬贊同。

　　我們此次 Final Project 想要建立機器學習模型幫知乎回答給分數。我們將選擇一些特定主題，搜集其下的問題文本、回答文本、評論數、收藏數 …… 等，並以贊同數當作我們的評分標籤。我們希望能在不依靠贊同數的情況下，透過這個模型為知乎的回答提供一個客觀的參考依據，讓發問者可以迅速知道哪些是好的回答以及讓回答者知道自己的回答水平如何。而我們的研究結果呈現在如下的網址，本報告將以解釋該頁面之功能為主。

知乎網站: <https://www.zhihu.com/>

Final Project 網站: <https://oomegappandda.shinyapps.io/shiny_dsr_zhihu/>

二、資料搜集 (Data Collecting)

我們使用網路上的非官方 Python API ，自己撰寫爬蟲程式來取得知乎平台上的資料，此為該 API 的來源：<https://github.com/7sDream/zhihu-oauth>。爬取的資料欄位如下表。



我們最後蒐集到了 21 份資料集 (即 21 種不同話題的資料)，總計 1,300,000 筆資料。我們將運用以上的資料來進行我們後續的資料分析。

三、數據資料檢視 (Numeric Data Observing)

在左方的側欄中選擇想要檢視的資料集來檢視，若想要同時檢視多筆資料進行比較可以直接選取多筆資料急，點選更新頁面後即可檢視以下初步分析資料 ：

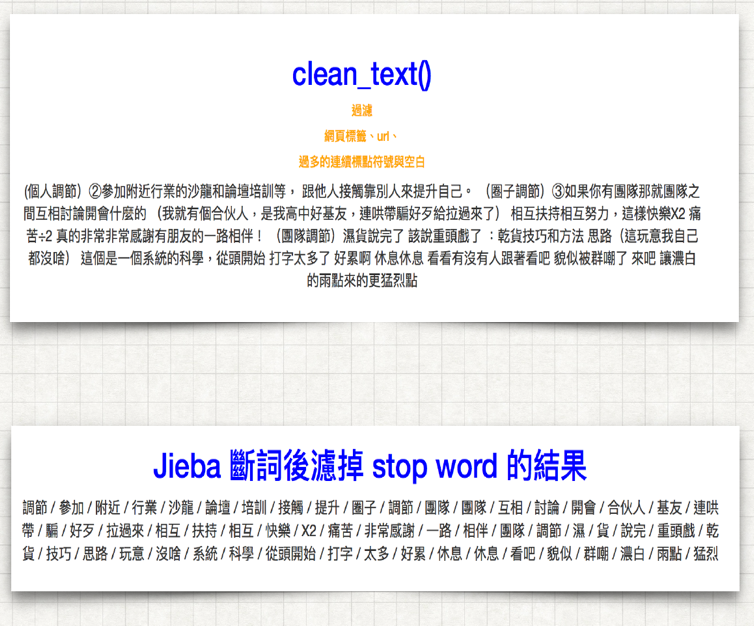
1. 各資料集的資料筆數
2. 回答贊同數與時間的交互關係
3. 回答贊同數與回答者總讚數的交互關係
4. 回答贊同數與回答者被追蹤的人數的交互關係
5. 回答贊同數與回答者追蹤的人數的交互關係
6. 回答贊同數與回答者獲得總感謝數的交互關係
7. 回答贊同數與回答者發問數的交互關係
8. 回答贊同數與回答者回答數的交互關係
9. 回答贊同數與回答者文章發表數的交互關係
10. 回答贊同數與回答蒐藏數的交互關係
11. 回答贊同數與回答評論數的交互關係

四、文本前處理 (Text Preprocessing)

這個部分是針對剛利用爬蟲程式抓下來的資料進行初步整理及過濾，主要包括以下三個步驟：

1. 過濾網頁元素標籤和過多的冗贅無意義字元。
2. 斷詞：使用 JiebaR 來進行斷詞。
3. 過濾 stop words：取出所有文本中 df 值排名在 2.5‰ 以上的字詞進行過濾，因為這些字詞通常不具重要意義，例如：的、了、是、我、都......。另外我們也使用了 tmcn 套件裡的 stopwordsCN() 作為我們停止詞的補充。

為了讓大家能更明白我們前處理的效果，在我們的網頁中有展示前處理的區塊，在 text area 中輸入利用爬蟲程式抓下來的內容，點擊送出文本後將立即開始進行上述的處理，並且可以看到我們文本清理和斷詞的結果，以下是一個簡單的範例。

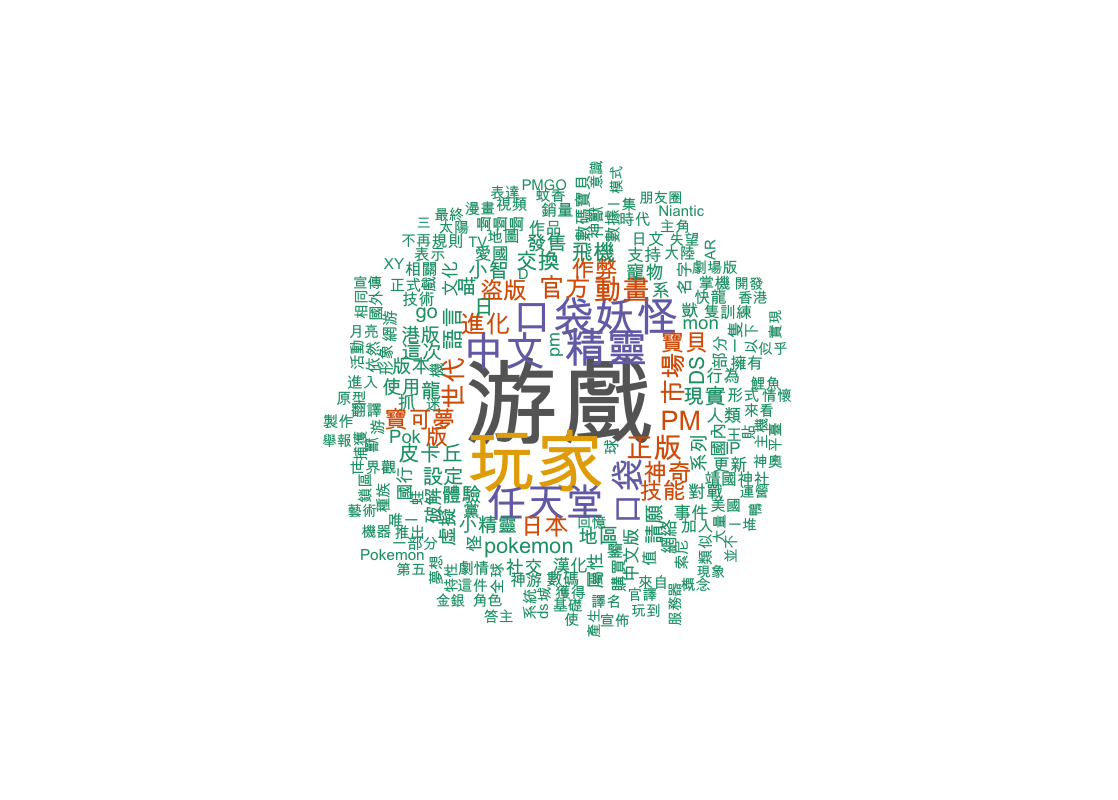
 

五、文本主題 (Text Topic)

探勘文本主題有幾種方法，在這裡我們主要是運用了文字雲、tf-idf 關鍵詞 和 LDA 這三種方法來進行觀察，以下我們以pokemon資料集為例。

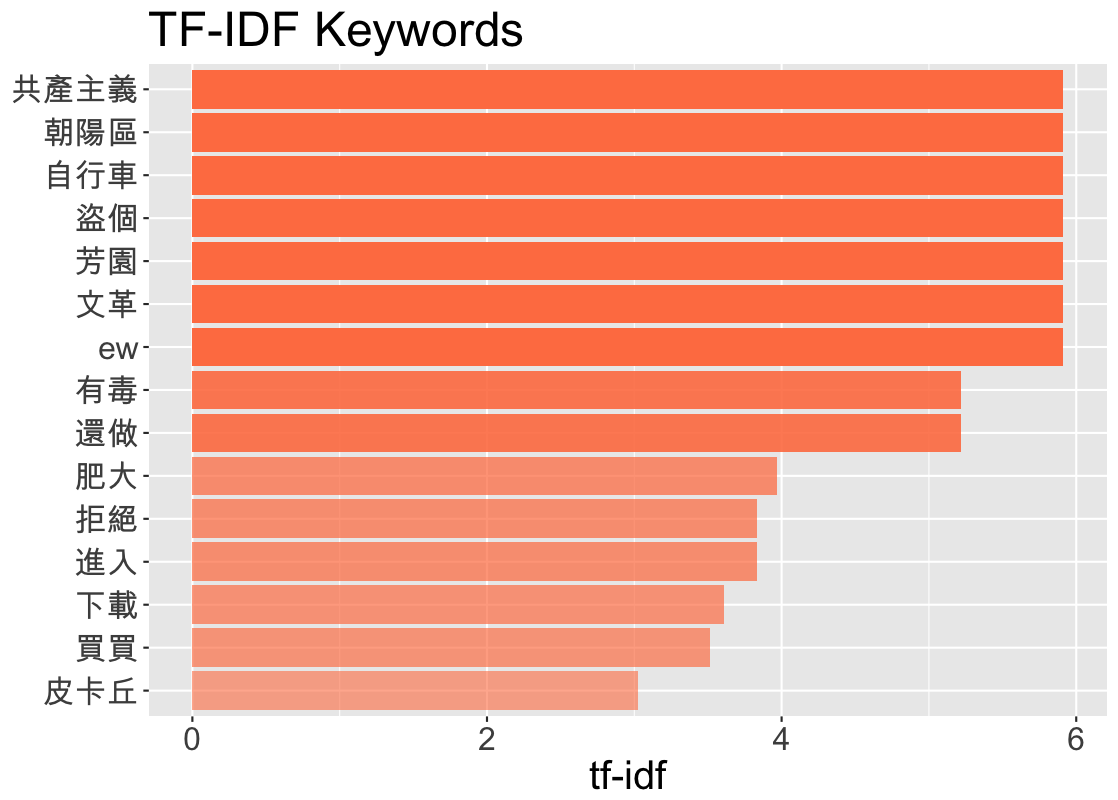
1. 主題文字雲：

根據詞頻產生文字雲。



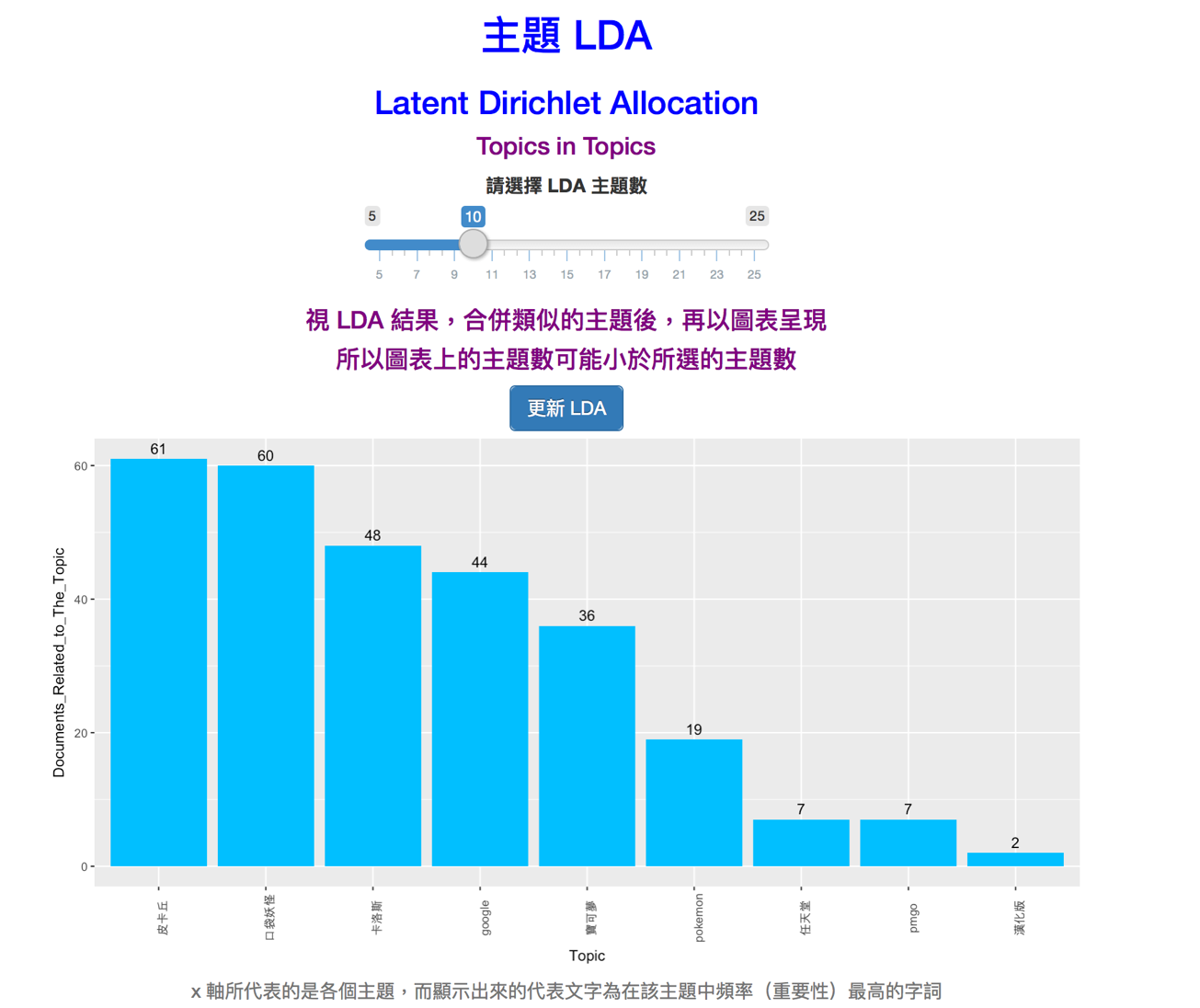
1. tf-idf 關鍵詞：

我們將主題中 tf-idf 值進行排序，前幾名高的詞彙對該主題的代表性就越強。



1. 主題LDA (Latent Dirichlet Allocation) 分析

LDA 是一種非監督式的文本主題探勘模型，我們將資料集中的所有文章透過LDA 分析後就可以發現其中隱藏的子主題。(Topics in Topics)



1. 除了上述三種方法外，我們的網站也有將熱門的問題和答案做成 table，大家可以透過操作看看有沒有什麼有趣的問題和答案。

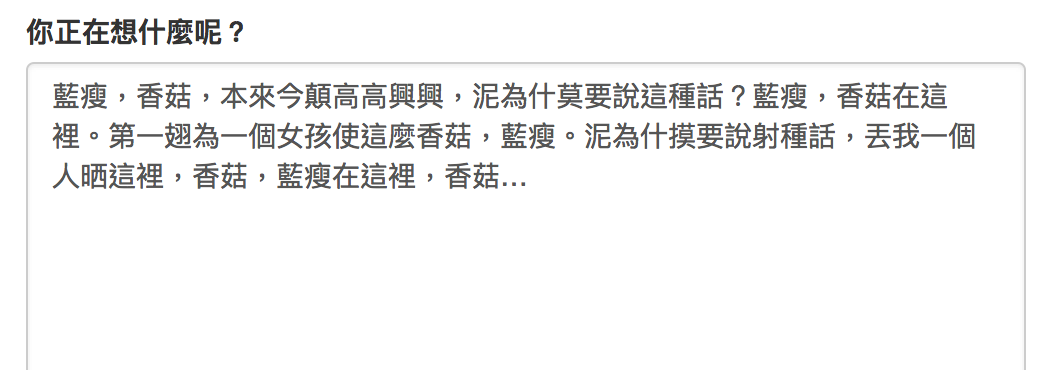




六、情感分析 (Sentiment)

情感分析的部分，我們在網頁上製作了一個情感分析機，使用者可以輸入一段文字並得到一個情感指標。

1. 請輸入一段文字

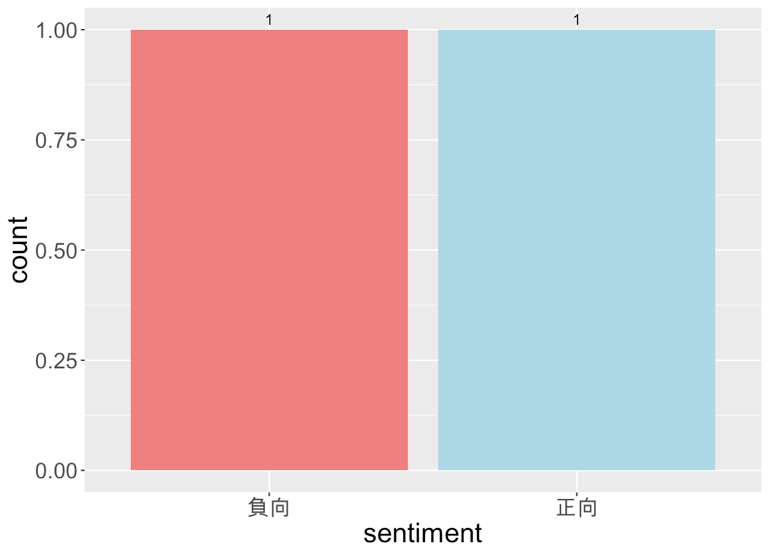


1. 依據文字內容作情感分析並顯示心情指數

* 正負面詞彙文字雲



* 詞頻統計



* 心情指數和一隻代表你心情的皮卡丘



1. 如果覺得不準，也可以自行新增正、負面詞彙



1. 情感字典來源

* NTUSD:

<https://docs.google.com/forms/d/e/1FAIpQLSe20EyOE3bp9cKT0gF6R4DodTHOmriIGegkGYa03oHYejhi9g/viewform?c=0&w=1>

* der3318:

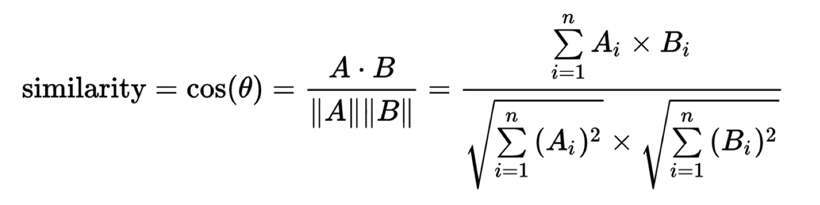
<https://github.com/der3318/SentimentAnalyzer/tree/master/docs>

* Tidytext bling:

<https://github.com/juliasilge/tidytext>

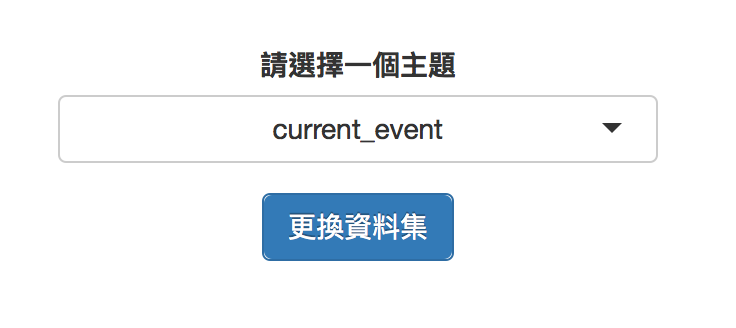
七、文本相似度 (Essay Similarity)

要計算兩個文本的相似性，我們利用了餘弦相似性，透過 cosine 我們可以算出 A 和 B 兩個 vector 的相似度。



在這邊我們令 A 代表問題文本詞頻的 vector、B 代表回答文本 tf-idf 值的 vector，並通過 cosine 計算出來兩者之間的相似度。

1. 首先選擇一主題



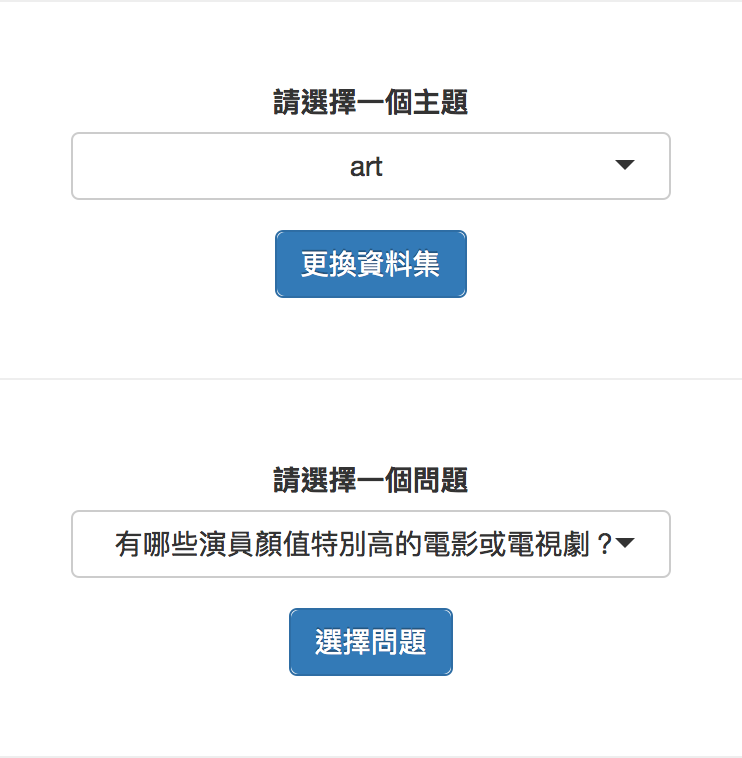
1. 選擇與此主題相關的問題以及其下的某一回答，就能顯示出回答與問題之間的相關程度。



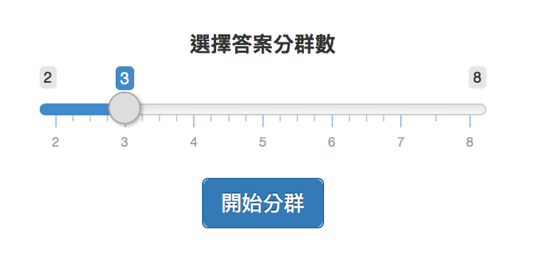
八、文本分群 (Clustering)

我們分群的方法是將 Term Document Matrix用 PCA 降階後再餵給 kmeans。

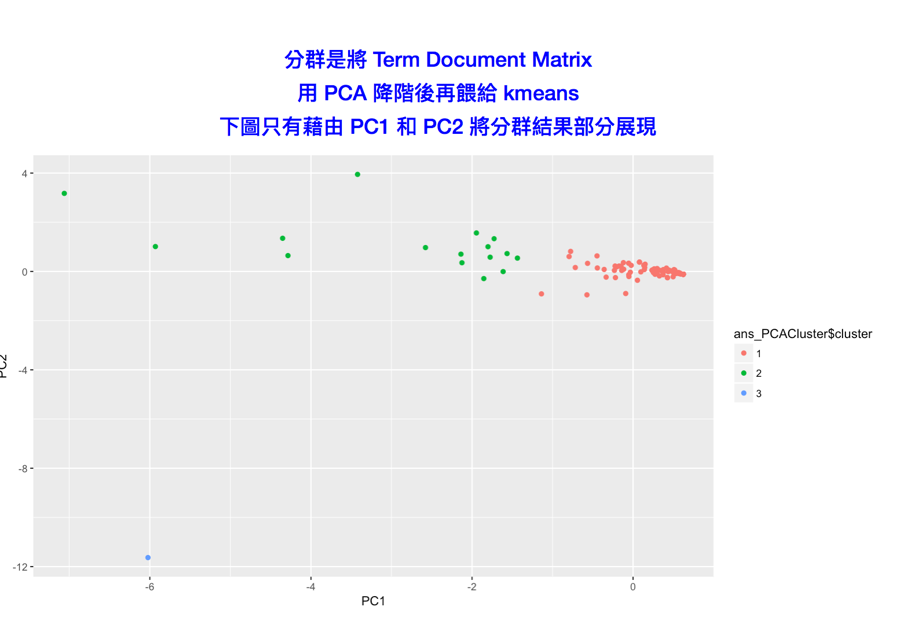
* + - 1. 先選擇一個主題以及問題



* + - 1. 選擇答案分群數



1. 展示分群效果 (以下只畫出 PC1 和 PC2 的關聯)

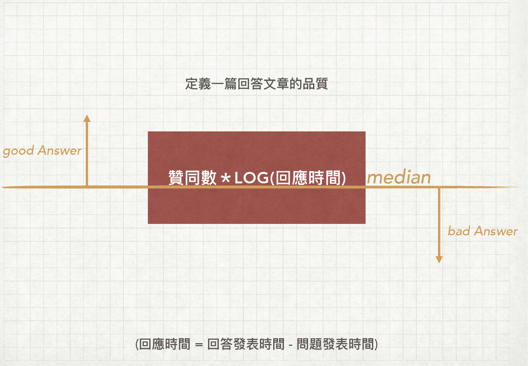


我們認為分群後可以將文本與各分群群簇的接近度作為文本 features 來看待，舉例來說假設今天我們的文本集是有關政治的文本集，那分群後的結果有可能各個群簇都各代表了某一政黨的言論。那以一篇文章來說，我們可以觀察其在分群後的與各群簇的接近度，並以此判斷他比較屬於哪一個或哪一些黨派的言論。

九、支援向量機 (SVM)

這部分是我們這份專題最關鍵的部分，希望根據我們之前分析的結果進行回答的評分。在這邊我們使用了 SVM 的機器學習模型，使用 e1071 的套件來完成這項任務。

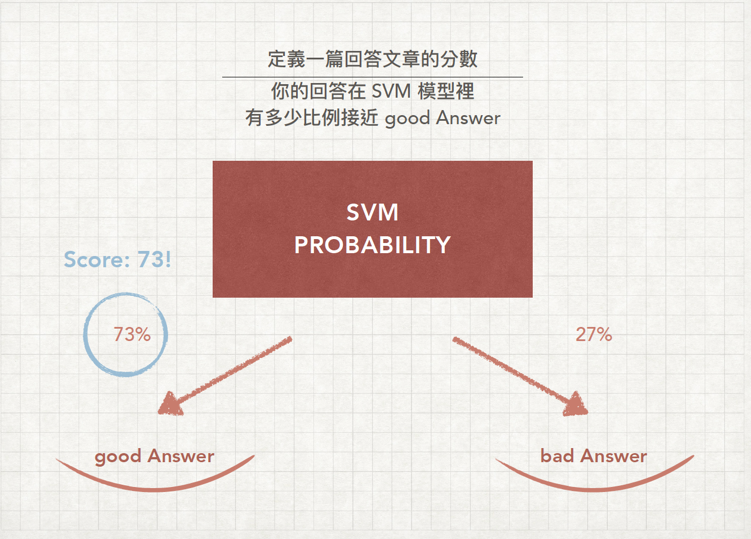
在幫回答評分之前我們必須先定義和區分 training data 中好的答案和壞的答案，在這邊我們認為答案之好壞可以從其贊同數來觀察，另外因為回應時間也是一個影響贊同數的因素，所以我們將 scale(贊同數 \* log(回應時間)) 作為我們評判好壞的標準值，在標準值的中位數以下即代表壞答案，標準值的中位數以上即代表好答案。



在 SVM 這個監督式的機器學習模型裡，我們將答案的好壞作為我們的目標 label，然後將以下項目作為我們的 features 來進行 training。

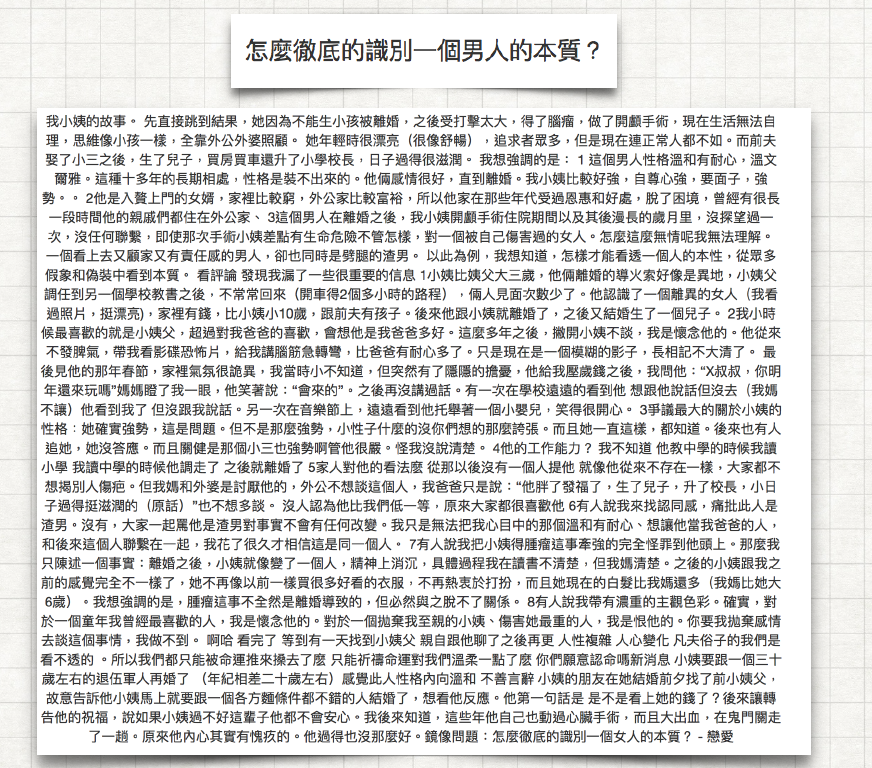
* + - 1. 回答文本的情感分數
      2. 問題文本與回答文本的相似度
      3. 文本分群後抽出來的接近度 features
      4. 回答文本的字數
      5. 回答文本的詞數
      6. 回答文本的 stop word 數
      7. 回答文本的 stop word 比例
      8. 回答者的被追蹤數
      9. 回答者的追蹤數
      10. 回答者獲得的總讚數
      11. 回答者的總感謝數
      12. 回答者的發問數
      13. 回答者的回答數
      14. 回答者的文章發表數

最後，SVM.probability 產出來的模型可以 predict 出答案有多少比例是接近好答案，有多少比例是接近壞答案，那我們就依其接近好答案的比例作為我們對該答案的評分。舉例來說假設今天有一個答案經過 SVM 的 prediction 後，得知他有 73% 會是好答案，27% 會是壞答案，那就將此答案的分數定為 73 分。



本頁有提供一個範例題目，使用者可以在欄位中任意輸入相關的答案，再填入個人的基本資料後點擊「看看我的回答水準」即可查看該回答之評分，評分越高者及代表你的回答品質越高。

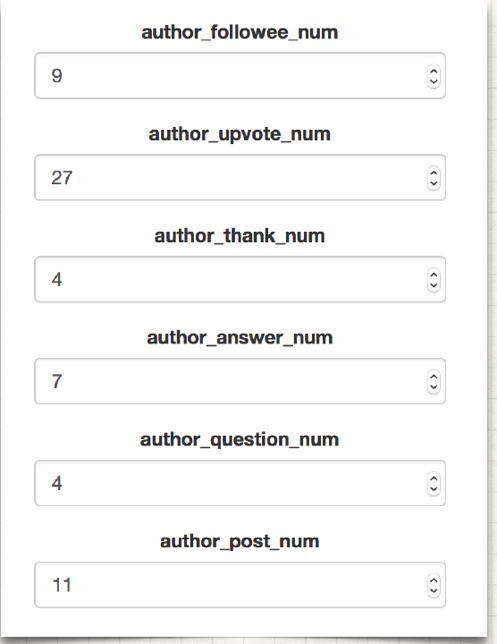
1. 瀏覽問題



1. 輸入回答



1. 輸入個人基本資訊

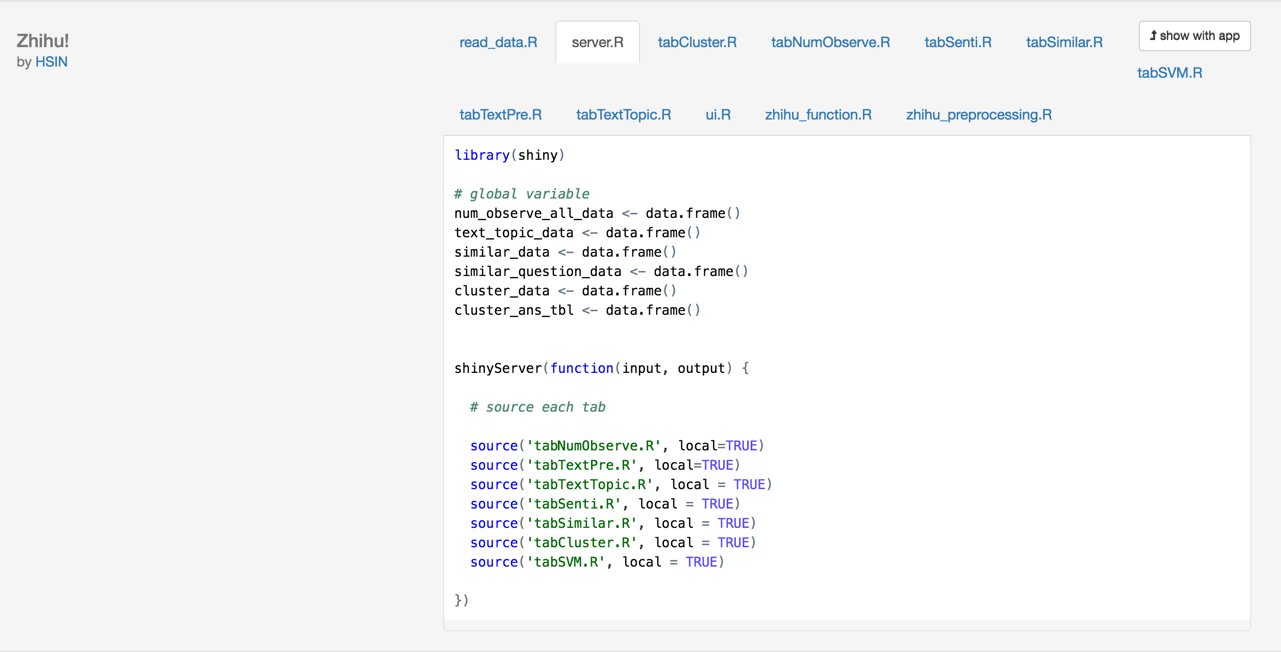


1. 看分數



十、程式展現

本網頁由 R語言撰寫，並經由 Shiny app 展現，在網頁的最下方有我們的 display code，可以完整的看到我們程式是如何寫的。



十一、分工細項

1. 爬資料：陳信豪、張景淵、冷俊瑩
2. 清理資料（斷詞、濾 stop words、標記資料等）：陳信豪、周晅瑢、高偉立
3. 基本分析（數據統計部分）：周晅瑢
4. Model（機器學習部分）：陳信豪、高偉立
5. 資料呈現彙整（Shiny / Notebook html)：陳信豪
6. 書面：冷俊瑩、黃薇甄

十二、附錄

本專案之 Coding 和相關檔案：<https://github.com/OOmegaPPanDDa/shiny_dsr_zhihu>