The Simpsons: Whose Line Is It?

104403011范哲豪、104403044吳謦廷

Abstract

Text Classification是當今資料科學中很重要的一個課題，特別是現在的社群網路媒體十分發達，要從龐大的文章中截取重要的訊息時，Text Classification便是一項重要的工具。此篇研究專注於將美國動畫影集「辛普森家族」所有集數的臺詞透過Text Classification的方式，從各角色的臺詞中，推斷及預測出屬於該角色的臺詞。從調整前處理的方式及分類器的選擇中，尋找最適合的方法以達到Precision Rate & Recall Rate表現最佳化。

1. Overview of the Study
   1. What is Text Classification?

Text Classification是一個會將文字分類的過程。分類模型會對文字進行分析並且針對特定的文字内容做分類。而這些可以自動將文章中的文字進行分類的機器學習模型就就稱作文字分類器。

* 1. Implications of the study

在這個研究中，我們從最一開始的資料導入及資料前處理直至最後的分類器選擇有通盤的瞭解。

進行資料的前處理，對於一個raw text來説，是非常重要的一環，且如何進行前處理會影響後續的資料表現。並不是做愈多的preprocessing工作，分類器就會有好的表現，例如進行sentiment analysis時，並不適合做過多的資料處理，因為文字前後和標點符號都會是影響情感分析的關鍵 。

進行分類及預測資料時，對於一個完成前處理的資料來説，分類器的選擇會影響到最後資料預測的表現。在此篇研究中，我們嘗試使用Naïve Bayes及kNN來檢驗預測結果。而相同的資料集在不同分類器中為何會有不同的表現，是在做資料探勘中值得分析的重點。

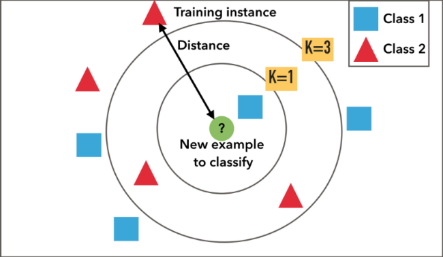
* 1. Objectives of the study

我們在這個研究中想要從輸入的臺詞中預測是哪一個角色說的話。所以在整個研究中，會從兩個大方向著手：

(1)調整前處理的方式，例如：標點符號及單一字母（I am…，I’m🡪I m）的去留，重複子母的調整（Ohhhh🡪Oh）。

(2)\*選擇不同的分類器，例如Linear Regression、Naïve Bayes、KNN、Random Forest。

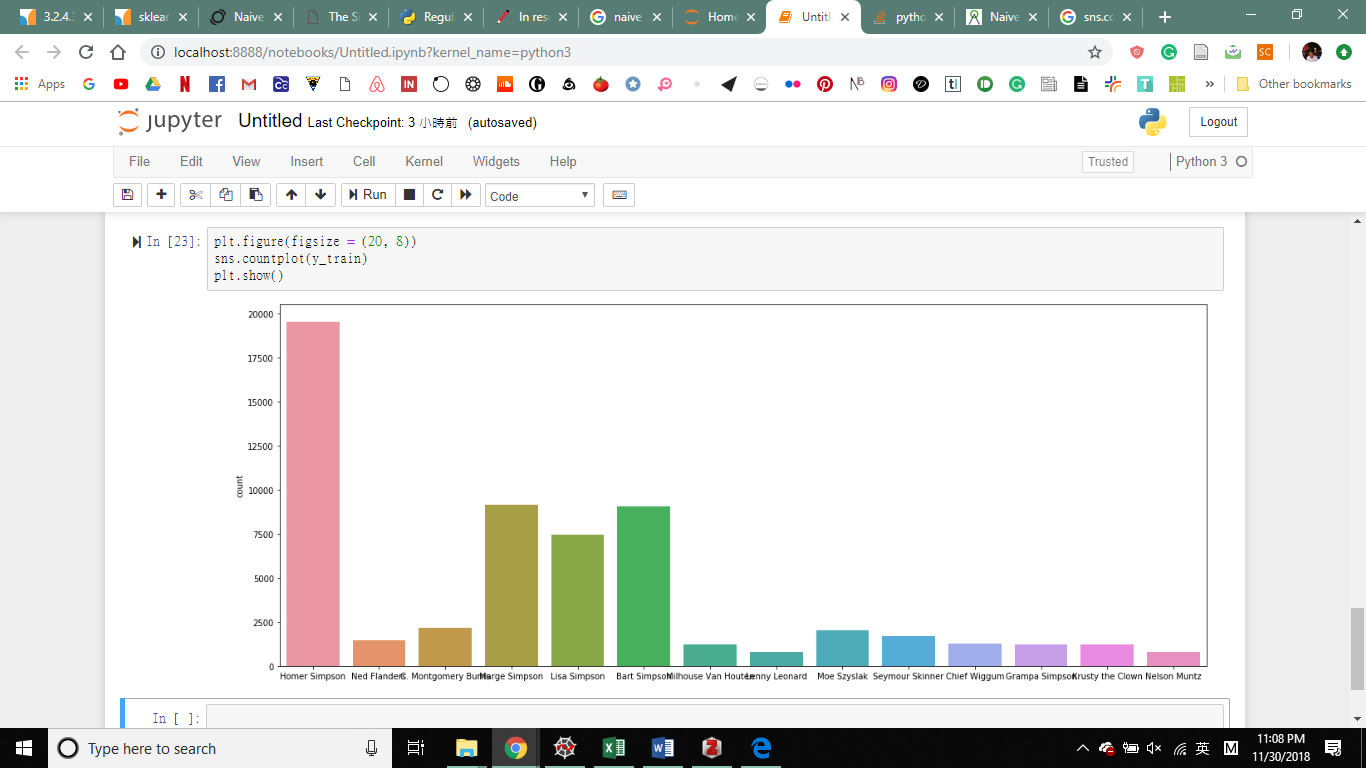
藉由調整其中的參數，選擇在Precision Rate 及Recall Rate之中表現最好的選項。

1. Brief literature reviews on text classification techniques

我們在這裏嘗試使用KNN作爲分類器。KNN是一個基於特徵相似程度的演算法，也就是一個新測試的點會被分類成最靠近附近k個樣本點中出現頻率最高的一類。在這邊我們更動的參數為n\_neighbor也就是鄰近參考點的數量。[[1]](#footnote-1)

這邊我們也嘗試使用Naïve Bayes作爲分類器。Naïve Bayes 是一個簡單的分類器，其中在預測新的資料時，是獨立且不受到原先資料的影響，也就是說分類器會從特徵中計算該新資料於各類別中出現的機率並視概率最大值之類別為此新資料所分的類別。

1. Methodologies
   1. Description/Stats of Dataset

我們使用的是名爲Simpsons\_script\_line的csv檔案。其中的資料集包含大約六百集辛普森家庭電視劇的角色、演出情節、及所有角色的臺詞。

我們在截取的14個角色臺詞中發現Homer Simpson所說的臺詞佔整體資料及的35%，如果說以一個平衡的資料集角度看，理當一個人説的臺詞不超過7%，但Homer卻是其5倍，加上其他家庭成員的臺詞Marge 14165、Lisa 11510、Bart 13792，四人所說的臺詞即佔整體資料集的81%，可見此資料集確實為一Imbalanced Dataset。我們可以使用Oversampling的方式解決此問題。即是透過增加少樣本數類別(Simpson家庭四人以外的十人) 的樣本數量，以實現樣本均衡，最簡單的方法就是複 製少樣本數類別的樣本，這樣的缺點就是會有過度擬和(Overfitting)的現象， 也就是在調適模型的過程中，可能使用過多的參數了。SMOTE (Synthetic Minority Oversampling Technique) 演算法就是 Over-sampling 的改良方案， 主要是對少樣本數類別進行分析，再將樣本人工合成新樣本新增到類別底下。所以我們可以嘗試使用此演算法改善此資料集不平衡的狀況。

* 1. Experimental Setups: How you tried to achieve good classification performance

我們訓練分類器的步驟可以分成三大部分 : 第一部分是重整原始資料，第二部分是資料集的前處理，第三部分是訓練多種分類器。

第一部分：重整原始資料成我們想要的資料集





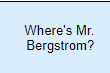
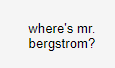
經過上課討論過後，我們統一出來只採用樣本數較高的14位角色，以及將column [ speaking\_line ]為false者拿掉後，一共剩下85403筆樣本，這85403筆樣本也就是我們要做文字探勘的資料集。

第二部分：資料集的前處理

1. 將字母轉成小寫

利用map( )對字串轉換成小寫



經過小寫轉換以後的資料

1. 清理標點符號

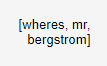




1. 向量化每個句子

利用nltk的word\_tokenize將每個句子向量化。

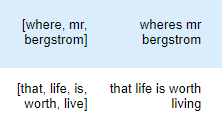




1. Stemming

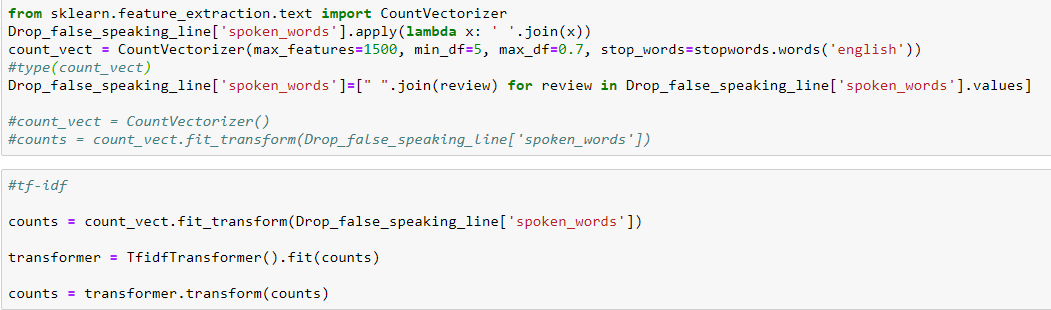
利用Porter Stemmer處理Stemming文字

可以看到像是”wheres”，經過stemming後只會留下 “where”， ”living”經過stemming後只會留下 “live”。

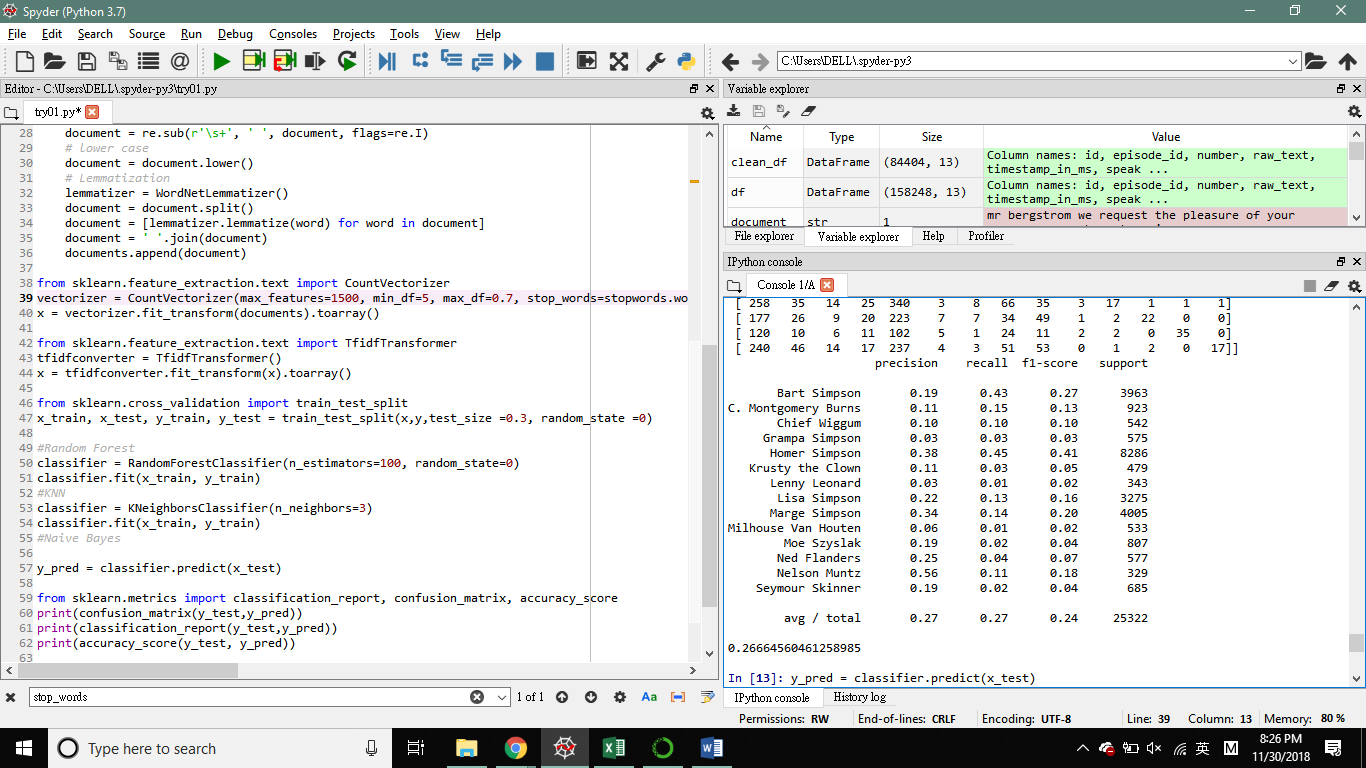


第三部分：訓練多種分類器（KNN和Naïve Bayes）

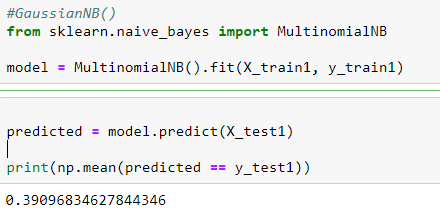
在訓練分類器之前，我們先使用tf-idf評估一個單字在資料集中的重要性，再利用train\_test\_split將資料集以training : testing = 7 : 3分。

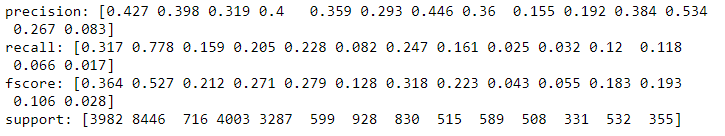


接著我們第一次進行分類器的嘗試為KNN，參數更改n\_neighbor=3,4,5，但accuracy, Precision Rate & Recall Rate 皆不變，結論於Results and Analysis中再詳述。我們所進行KNN分類器的嘗試，在不更動其他參數的狀況下，發現n\_neighbors=3,4,5的正確率及Precision Rate & Recall Rate皆相同，發生的原因尚待考證。

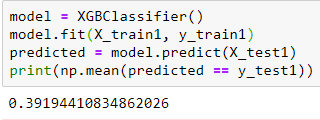


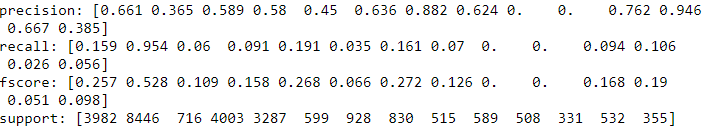
而第二個採用的分類器則是Naïve Bayes，Naïve Bayes所得的Precision Rate較高。





為了能夠尋找更好的分類器，所以我們採用了2015年的Kaggle競賽神器—XGboost (eXtreme Gradient Boosting)，但是XGboost的表現與Naïve Bayes差不多。





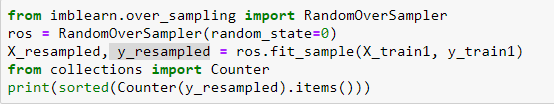
* 1. Explanation of study limitations

此資料集最大的困難與限制來自於我們前面提及的Data Imbalance現象，在Simpsons家族四人佔有大多數的資料的情況下，如果在訓練分類器之前，調整training資料，其實無論用哪一種分類器，其Precision幾乎不可能超過50%。

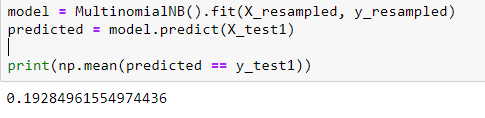
1. Results and Analysis

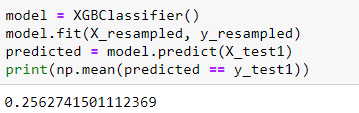
此資料集必須經過調整，再放入分類器，才有可能提高其Precision Rate，因此我們面對的問題是，要如處理這筆資料?

我們回頭複習了第一次的作業，思考該如何處理Simpson這個非常不平衡的資料集，決定採用Oversampling，也就是將樣本比較少的characters增加樣本數，我採用的是RandomOverSampler，首先我會先將前面train\_test\_split完的X\_training和y\_training做resampling，變成新的X\_resampled和y\_resampled，再將它們再次放入NaiveBayes和XGboost。



我們來看看經過Oversampling完後，NaiveBayes和XGboost的表現 :





可以看到，我們多做了OverSampling後，兩個分類器的表現卻更糟糕了，糟糕到我們可以直接猜Homer Simpson的Precision Rate還更高。這裡要提出我們實驗後的心得：

由於我們對Oversampling實際操作資料的經驗還不夠，我們目前只是使用Random Sampling，而很明顯的，在Random Sampling完後，提高了小樣本類別的樣本數，以致於 overfitting了。未來會再嘗試使用SMOTE看看。

1. Study Summary

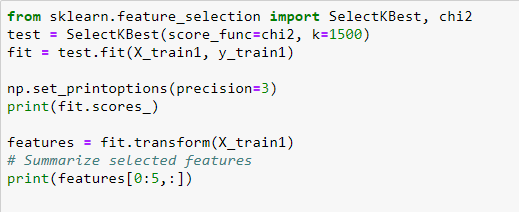
本篇報告著重在我們實作Simpsons資料集的過程記錄與問題探討，在methodologies中，我們描述了我們是如何做這筆資料的前處理；在Result and Analysis中，我們嘗試分析出此資料集訓練結果不佳的問題，並從Imbalanced Dataset著手，利用Oversampling的方式，期望可以提高Accuracy Rate，但是結果卻取得更低的Accuracy Rate。

Simpsons這筆資料集，是我們第一次完整的從資料分割、資料前處理再到分類器的訓練完成的資料集，雖然結果不盡人意，但是我們從中學習到了非常多，期望我們能不僅僅只是套用現有的分類器，而也能夠更深入的了解我們自己的Data，這樣我們才有可能觸碰到處理這筆資料的關鍵。

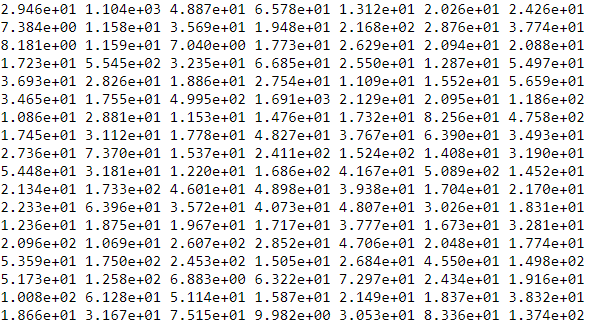
1. Revise on Previous Study

經過課程上的討論過後，我們決定針對資料集本身做更進一步的處理，再經由分類器訓練，期待能夠提高accuracy。

Feature selection是針對資料集，解少feature數量，達到降為的目的，最主要的是期望能夠增強labels和features之間的理解。老師上課時介紹了很多種feature selection的方法，我們決定採用chi-square，原因是我們在大二修過統計，對於chi-square的運算以及原理比較清楚，卡方分配將經過tf-idf vectorized過後的資料，計算每個feature對於character\_id的卡方值，卡方值愈大，代表此feature對於character\_id的關聯性愈強，反之則關聯性愈小。這裡的features指的是spoken\_words中的單字，而為了能夠提高accuracy，我們希望只留下那些對character\_id關聯性比較強的features，再放進分類器裡面訓練。



使用Sklearn.feature\_selection的SelectKBest和chi2就可以進行chi-square feature selection。



上圖為feature算出來的chi-square值，而SelectKBest會將排名比較前面的feature留下來。參數k可以選擇要留下多少的feature，我們希望留下來的都是在分類器訓練過程中有用的feature，從這個角度來說，我們會想要有少一點的feature，但是太少feature一定會影響分類器的品質，因為我們很有可能利用feature selection把關鍵的單字淘汰掉了，如此accuracy也不會提高，因此，SelectKBest的參數k是feature selection中非常關鍵的參數。

經過chi-square feature selection的資料集，並沒有讓我們所使用的三個分類器accuracy提高更多，我們經過討論過後分析出兩個原因:

1. 我們無法選定SelectKBest中最適合的參數K值，而使得我們將決定性的feature淘汰掉，以致於訓練出來的結果更不好。
2. 此資料集的feature可能不適合使用chi-square分析features，可能要嘗試使用其他feature selection的方法。
3. References
4. <https://www.datacamp.com/community/tutorials/feature-selection-python?fbclid=IwAR2PElvL6WHsDRbncJzduM9TEGXU7Oa1xiXULPmj0nLvxedHCua9k6eN7y8>
5. <https://towardsdatascience.com/another-twitter-sentiment-analysis-with-python-part-8-dimensionality-reduction-chi2-pca-c6d06fb3fcf3?fbclid=IwAR0toQJvEuBhzg2ROJSEb2gOM3s1MhqXrt64cIdbi8v8oLXgfAtNP14JmGU>
6. <https://towardsdatascience.com/why-how-and-when-to-apply-feature-selection-e9c69adfabf2>
7. https://dl.acm.org/citation.cfm?id=657137

1. Bronshtein, “A Quick Introduction to K-Nearest Neighbors Algorithm.” [↑](#footnote-ref-1)