Data mining

-Kaggle report

109136504 林哲廣

目錄

1. Preprocessing
2. Feature Engineering
3. Explanation of Model
4. Preprocessing
5. tweets\_DM.json

因為json檔的巢狀格式，利用多個index將其內容取出，並存成DataFrame。

1. emotion.csv & data\_identification.csv

將這兩個檔案存成字典(加快搜尋速度)，並利用tweet\_id將對應的內容取出，加進DataFrame。

1. 了解每種emotion的分布

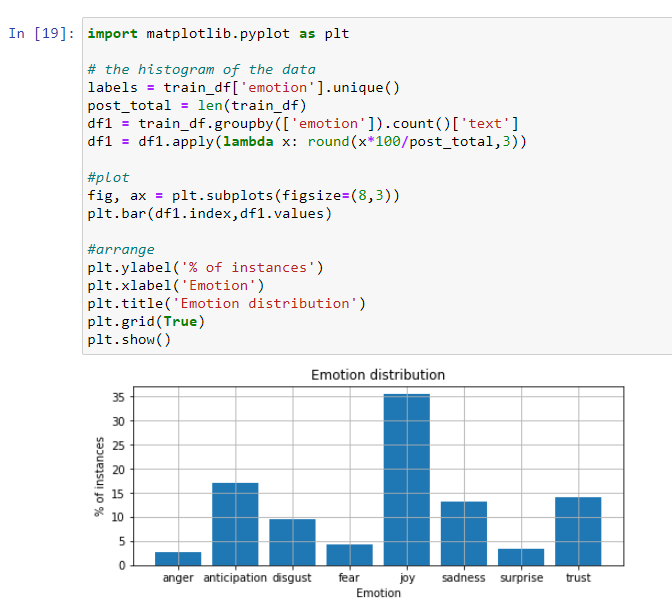


Figure 1 emotion distribution

從fig 1中看出，資料並不平衡，其中，anger, fear, surprise的資料量特別少，可能會造成後續訓練model時出問題。

1. Feature Engineering
2. Bag of word

我原本很貪心，想要將所有的資料轉換成baf of word，然而記憶體不足，因此改為取出現頻率較高的500個字做為feature，最後得到一個1455563\*500的array。

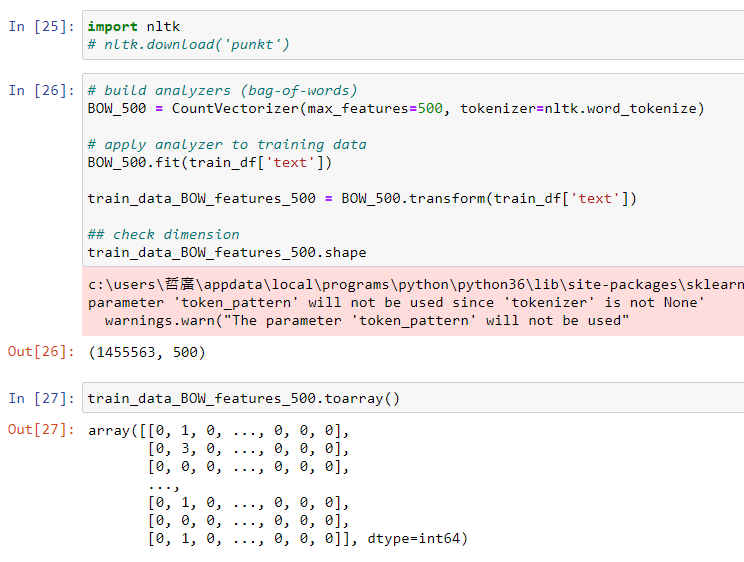


Figure 2 bag of word

1. Explanation of Model
2. Decision Tree

我想試試經典的model，因此選擇了decision tree，將原始訓練資料丟入bag of word轉換後，得到training data，再丟入Decision Tree(DT) 做訓練。

原本我很貪心，想讓DT將feature分到最細(雖然會overfitting)，因此沒設定DT的最大深度，然而訓練1天後，DT依然沒訓練好，因此放棄，開始設定DT的最大深度:max\_depth。

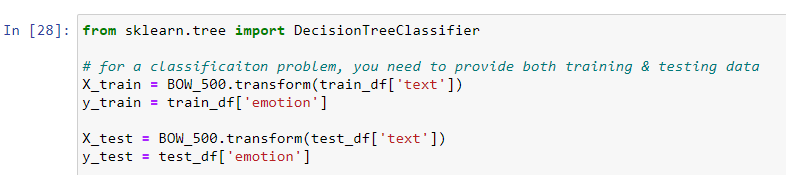


Figure 3 decision tree

1. max\_depth

下面3張圖片為不同的max\_depth對training accuracy造成的影響。可以看出隨著深度的增加，準確度也上升然而有其上限，深度=70~100時，準確度變化不大，因此最後折衷選擇90。

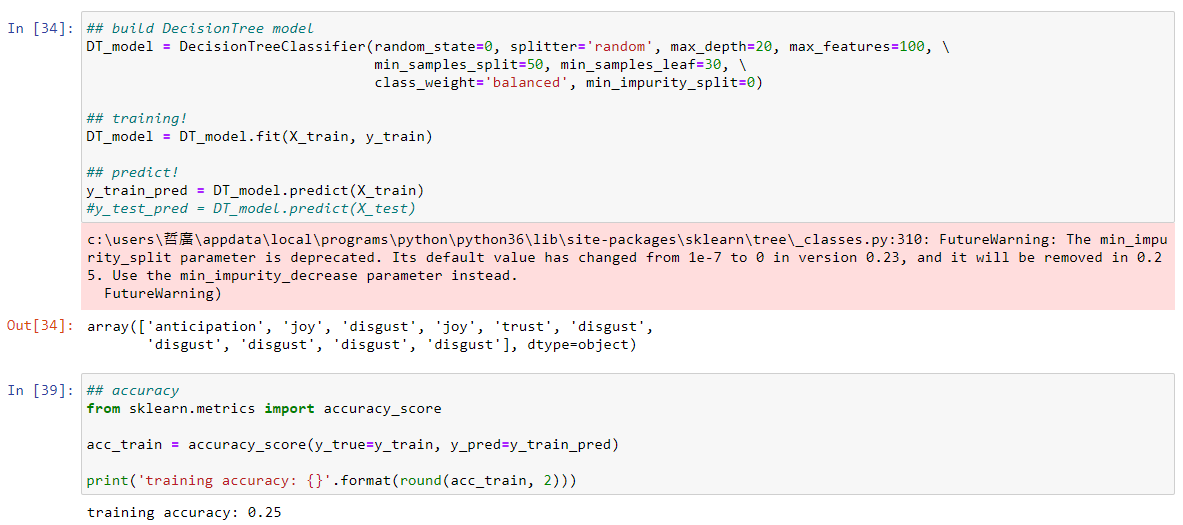


Figure 4 max\_depth = 20

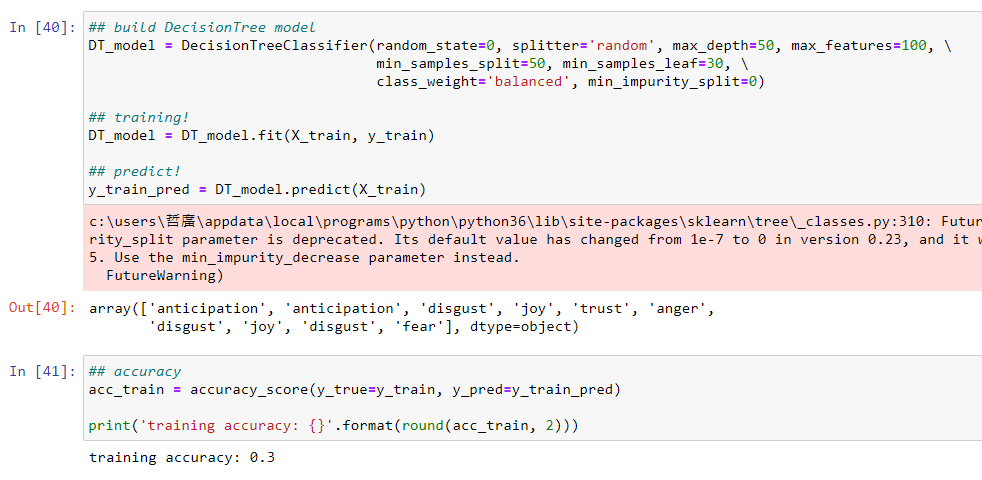


Figure 5 max\_depth = 50

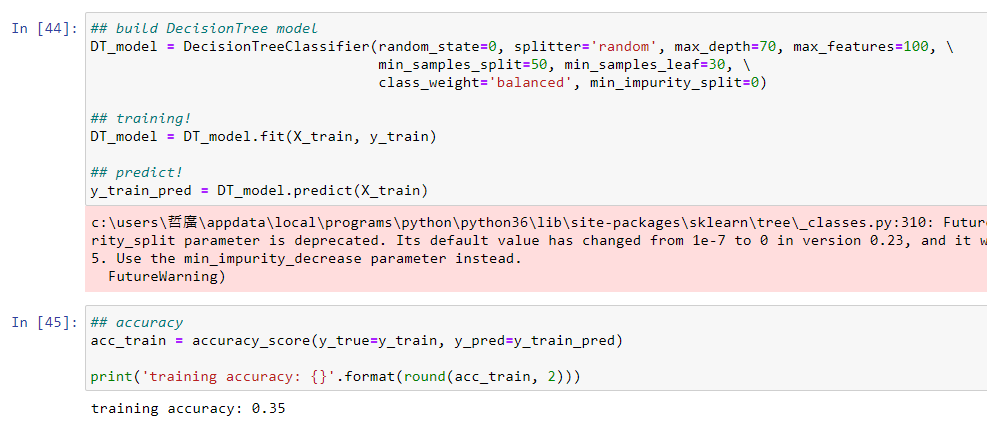


Figure 6 max\_depth = 70

1. 結論

Fig 7 是我經過不同的參數交叉測試後，得到的最佳DT，準確度為0.46，準確度不高的原因應該為Feature Engineering的部分沒做好。

Bag of word 500是取出出現頻率最高的500個word做為feature，然而出現頻率高不代表重要(例如the, a, that…皆為出現頻率高但不重要的字)，用TF-IDF可能會更好。

另為，bag of word無法表現出word之間的關聯性，這可能也是DT表現不好的原因之一，用word2vec可能會更好!

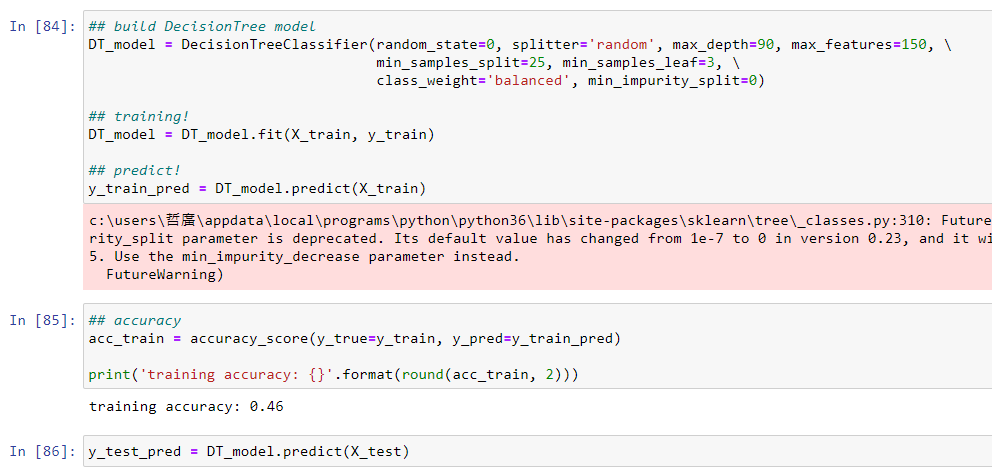


Figure 7 Final Model