資料分析與學習基石 hw3-2

統計 109 陳知遙 H24051053

一、資料集簡介與預測目標

我選擇的資料集是 Drug Review Dataset，這個資料集包含的資訊是患者在不同的症狀下服用藥物後對藥物的評論及評分，而預測的目標設定為根據使用者的評論來預測評分等級。以下是這個資料集的欄位說明：

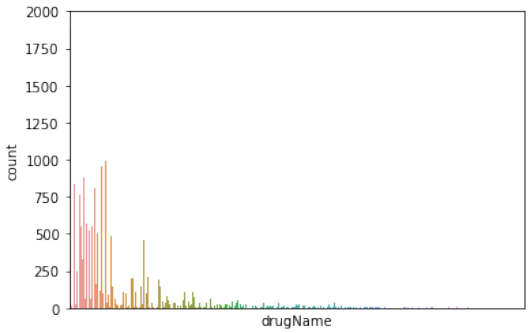
欄位名稱	說明
drugName (categorical)	藥物名稱
condition (categorical)	患者的症狀
review (text)	患者對藥物的文字評論
rating (numerical)	患者對藥物的評分(1-10)
date (date)	評分的日期
usefulCount (numerical)	這個患者的評論被標記有用的次數

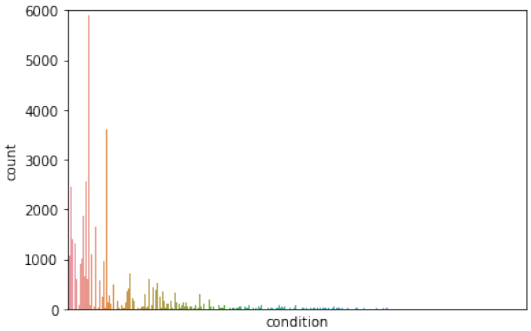
而這個資料集本身就已經幫我們切分好訓練及測試集，兩者的大小如下表：

訓練集樣本數	161,297
測試集樣本數	53,766

二、資料探索

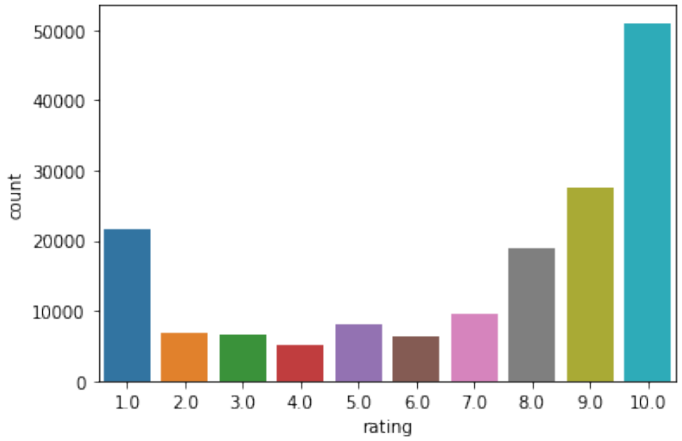
這個資料集的欄位數不多，且除了 usefulCount 之外都是文字或類別型資料，因此我從觀察兩個類別型變數 drugName 及 condition 開始，將他們在訓練集的出現次數分布圖畫出來如下兩張圖：





這兩個類別變數分別有 3436 及 885 種類別，可以明顯看到有分布不均的情況，其中又以 condition 更加明顯，最常出現的症狀就佔了近 6000 次。考慮到這樣的類別資料除了分布不均外，種類也太多太繁雜，無論是要進行 one-hot 還是其他編碼方式，對維度都會有不小的負擔，因此初步分析上先不考慮加入

這兩個變數。
接著觀察 rating，也就是我們預測目標在訓練集的分布如下圖，可以看到兩端，也就是 1 分及 9、10 分的出現次數較高外，中間值的分布較為平均。



三、模型建構過程
考慮到這個任務中，患者的文字評論應該是對於預測最直接相關的特徵，因此我從這個部分著手，先想辦法將文字轉換為向量，變成是模型能夠據以訓練的特徵，而這個部分我從最簡單的 CountVectorizer 開始。

CountVectorizer 轉換後的特徵高達 49,891 維，且象徵的意義只有這個文字在我們的資料集中出現了幾次，因此可想而見的應該不會太好，果不其然這個初步的模型只得到了如下的結果：

模型名稱	CountVectorizer
Accuracy	0.3613
F1 score	0.2255

在此因為是多類別的分類問題，且類別仍有資料不平均的現象，因此我加入考慮的 F1 score 作為衡量的指標。

接著我嘗試使用 tf-idf 將文字轉換為考慮其重要性的數值的方法，不過得到的結果相較於 CountVectorizer 沒有太大差異，整理如下：

模型名稱	TF-IDF
Accuracy	0.3535
F1 score	0.2252

以上兩種方法都沒有考慮到文字之間的詞意關聯，因此我想嘗試訓練一個 word2vec 的模型，讓每個字能夠轉換到低維空間裡，使得字與字之間的語意關係在這個空間中被保留下來，這個部份我借助了 gensim 這個方便的套件來完成。訓練完成後，我們可以得到每一個字的語意向量，不過一個樣本包含的是一個完整的段落，可能包含多個單字，這邊我採取相對簡單的做法，就是把這個句子中出現的每個字的向量平均起來，當成是表示這個句子的向量，最後餵給一個單層 MLP 來進行預測，得到不錯的提升。

模型名稱	1-Layer MLP
Accuracy	0.4492
F1 score	0.2312

不過這樣的表現還是不夠，況且這還只是在訓練集上的分數，因此我將 usefulCount 這個特徵加入，並且將 MLP 的層數增加為 2 層，發現結果明顯提升了，也到了一個我自己覺得還算可以接受的結果，整理成表格如下：

模型名稱	2-Layer MLP
Accuracy	0.6758
F1 score	0.6146

於是我將這個模型套用到測試集上，得到最後的結果：

模型名稱	2-Layer MLP
Accuracy	0.4811
F1 score	0.3623

可以看到模型還是有些微過擬合的現象，不過我認為這個任務本身的難度就偏高，因為通常做文字方面的預測，可能只需要分類是正向或負向，可能的類別就只有兩到三種，那麼無論是看準確率還是 F1 score 都較容易有好的分數，但今天這個任務可能的類別有 10 種，又沒有太多文字以外的特徵能夠採用，因此我認為模型這樣的表現並不算是真的太差。