Deep Learning

Competition 1

Team : 歐🆒比

107062208邱靖豪 107062132鍾皓崴 107061202蔣承軒 110062577景璞

一. Baseline60

在這次的competition中，起初我們使用助教提供的template，先將內文、作者名稱、時間、新聞種類，利用TF-IDF編碼，得到各自對應的feature，接著我們分別嘗試以SGD_Classifier, XGBoost, RandomForest當成分類器，我們發現到效果差異並不會很大，因此便以最熟悉的RandomForest做為classifier。然而，由於這樣連基礎的baseline60都無法達標，我們便有了將feature降維的想法。我們使用selectKBest這個方法降維，此方法可以將陣列降為指定的維度。而個別針對內文、作者名稱、時間、新聞種類做降維後，我們終於可以將分數提升到超過baseline60。

我們嘗試tune過的變數有以下幾種：

- Number of features in HashingVectorizer：2^15, 2^18, 2^20
- K in selectKBest：30, 50, 100
- learning rate in XGBoost：0.2, 0.3
- max depth in XGBoost：5, 7
- max depth in random forest：8, 10

二. Further improvement

基於前者所做出的嘗試，本組成員發現在此次競賽中若直接將文章以tfidf或hashing等方式，將整篇文章轉化成向量表示法，最有可能面臨的問題便是:所產生出來的高維度稀疏向量對於傳統機器學習所使用的方法來說過於複雜，不易學習出一個好的決策邊界; 此外，高維向量也容易造成不同特徵在結合時互相干擾。在前篇章節中，我們採用降維的方式，取得了一定程度的提升。然而，降維卻只能一定程度的保留原始特徵資訊。以本次任務來說，向量化本身即可能無法完全表示文章屬性，若再進行降維，在學習上存在一定侷限。

本組同學於此進行了另一種方向的嘗試，也就是將原文章的特徵，由文章中隱含的其他可量化指標表示。舉例來說，我們首先嘗試的方向是將文章時間轉換成共六維的數字(年月日時等)，單以這些時間資訊佐以最大深度8的random forest所得出的結果，便從原先bottleneck的0.53，大幅提升至0.55。

從本次實驗中可得知，若要有效提升模型分類效能，便應該在有限的維度中，盡可能以與任務有關之特徵進行分類。同時，有效降低特徵維度的方法便是，將原文章特徵量化為數字。以下我們會列舉數種最終分類模型所使用到之特徵:

- time

  原文章中存在新聞發布時間的標籤，將其量化後我們取前四維(年月日時)，以較少的維度來儲存時間資訊。此外，我們也回推文章發布星期，作為第五維加入。根據實驗，以同樣設定的random forest光是多加入此項資訊便能將AUC從0.55提升至約0.57。

- category

  在文檔中，每篇完張都有category的分類，我們將類別作為feature，並且利用target encoding的方法進行encoding。其中，由於類別過多，我們只挑選類別次數出現大於50次的類別，反之則當成noise捨棄，這樣可以大幅降低維度。另外，取前50常出現的category出現後feature的為度仍然非常大，我們就直接取做完target encoding後的前五個feature。

- data channel

  在文檔中有data channel這個tag，而已每個文檔只會擁有一個tag，我們同樣也利用target encoding的方法對data channel進行encoding。利用target encoding的原因是，若利用onehot encoding會導致維度太高，而且target encoding能夠保留不同channel對於popularity的影響。

  我們也嘗試了，圖片、影片數量，文章長短，作者編碼等。但在已有時間這一特徵的前提下，無法取得有效提升。

最後，我們也嘗試在更換新的特徵後將分類器從random forest改為XGBoost。若不調整參數，在此任務下XGBoost很容易overfitting，因此我們做了增加estimator的數量，調高參數gamma，等調參。在最終public board上獲得了目前本組submit 最高的結果。

Final XGBoost setting:

```
n_estimators=2000
max_depth=5
gamma=6
objective='binary:logistic'
eval_metric='auc'
```