社群平台觀看次數預測

臺大國企所 陳帝文 r10724050

摘要

本次作業是要預測社群平台上的觀看次數。在特徵選取方面主要是參考給予的資料特徵並且依照自身對社群平台觀看次數的認知進行一些假設後，計算出新的特徵，再用此些特徵利用 lightGBM 預測觀看次數。除此之外，利用遷移學習的想法訓練一個可以讀取照片後預測出觀看次數的神經網路。最後再結合兩者的分數進行優化後得到最終結果。

1. 數據觀察

我們有的特徵有 userID，Title，Tags，Category，Concept，Subcategory，Postdate 和照片。首先，我對這些特徵進行下列假設:

a. 照片為主的社群平台，像是 instagram，主要是照片影響觀看次數。人們不會因為 Title 的好壞決定是否觀看，因此這個特徵省略。

b. User 如果有比較多的發文，會累積一定的粉絲，因此觀看次數較高。所以我計算 userID 的出現次數給予新的特徵，uid_frequency。

c. Tags 的數量越多，越有機會被人看到。但也要計算 tags 品質的好壞，假如 tags 都是一些自創或冷門的文字，並不會吸引到其他人。依照此邏輯，我計算 numberoftags 和 tag_weighted_value。前者主要是代表 tags 的數量，後者使用 target encoded 的方法先去計算每個 tag 的 value，再依照該 tag 在所有資料中出現的次數給予權重進行平均。

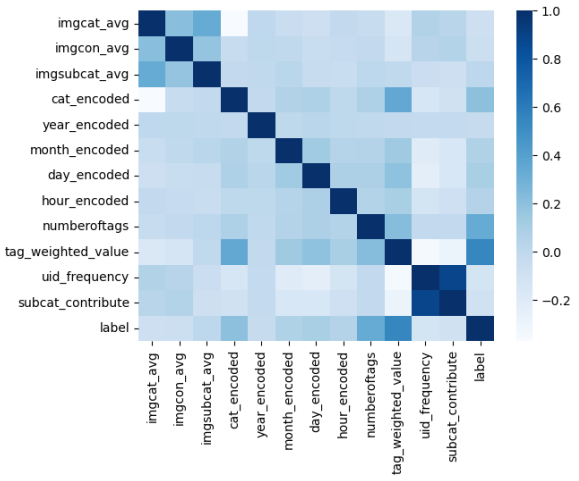
d. 為了提取 Subcategory 的訊息，我假設如果有某個用戶對某一 Subcategory 的貢獻太大，也就是大部分的人對此分類並無興趣，只有少部分的人使用此分類特徵，因此受到觀看的次數較少。依照此邏輯我計算了 subcat_contribution 的特徵。

e. 若照片的內容和分類相關性較低，則有可能是分錯類，影響觀看次數。例如照片的內容是狗，但分類卻是 Travel，mitt，Baseball，則會使喜愛狗的人沒辦法觀看到該照片。依照此邏輯我計算了 imgcat_avg，imgcon_avg，imgsubcat_avg，計算方法會在實驗方法中解釋。

除了上述的假設外，我也計算了 category 和 postdate 的 target encoded feature，已提取其中訊息。為了觀察依照人為概念計算出來的特徵是否有效，表(一)是將上述特徵和 labels 進行線性迴歸後的結果，可以看到 imgcat_avg，numberoftags，tag_weighted_valu 和 uid_frequency 十分顯著。雖然其他特徵的顯著性較低，但線性迴歸只能捕捉線性關係，使用機器學習的方法可以更好捕捉非線性關係，因此仍然可以嘗試將這些特徵用來預測。圖(一)則可以看到個特徵和觀看次數的相關性。

Feature	coef	std	t	P > |t|
imgcat_avg	1.583	0.476	3.325	0.001
imgcon_avg	-0.073	0.249	-0.292	0.771
imgsubcat_avg	0.243	0.277	0.879	0.380
cat_encoded	0.120	0.036	3.306	0.001
year_encoded	-1.085	0.156	-6.967	0.000
month_encoded	-0.043	0.086	-0.495	0.620
day_encoded	-0.015	0.058	-0.269	0.788
hour_encoded	-0.032	0.089	-0.363	0.716
numberoftages	0.033	0.001	32.370	0.000
tag_weighted_value	1.593	0.024	67.548	0.000
uid_frequency	0.003	0.001	4.365	0.000
subcat_contribute	-0.072	0.255	-0.280	0.779

表(一)特徵線性迴歸結果

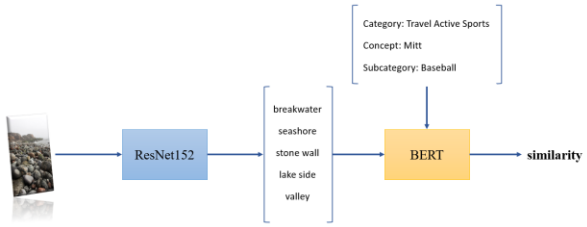


圖(一) 各特徵和觀看次數的相關性

2. 實驗方法

- *相關性模型*

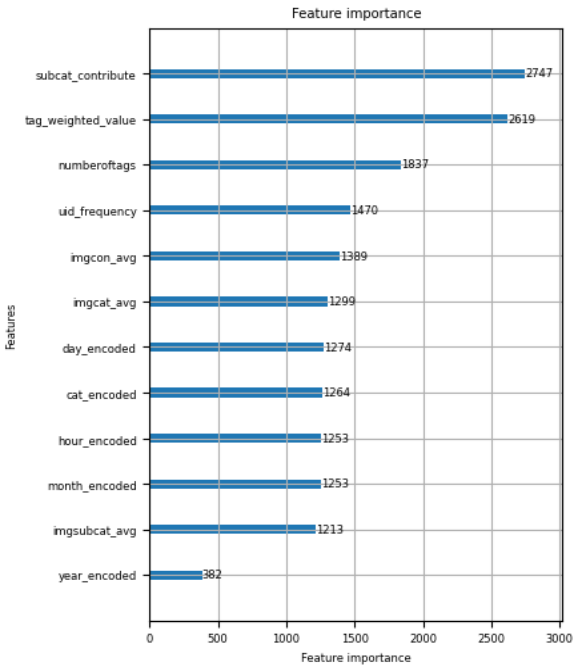
前面有提到我有計算圖片與 Category，Concept 和 Subcategory 的相關性。首先，我使用 pretrained ResNet152 模型和 imagenet1000 文本，讓模型分析圖片後預測出代表圖片的字，並且取前五大機率高者；接著使用 BERT 去生成代表這些字和 Category，Concept，Subcategory 的向量。最後去計算 Category，Concept 和 Subcategory 分別和這五個字的 cosine similarity，並取平均。圖(二)是整個模型的流程圖。



圖(二) 相關性模型流程圖

- *結構化特徵模型*

得到結構化特徵資料後，我使用 lightGBM 去預測觀看次數[1]，圖(三)是該模型對各參數的重要性排序圖。



圖(三) 特徵重要性比較

圖(三)中可以觀察到對於 lightGBM 模型而言，貢獻度特徵和 tags 相關的特徵都很重要，尤其是線性迴歸中無顯著相關的 sub_contribution，而這結果符合開頭假設的情況。

- *非結構化資料模型*

除了結構化特徵外，仍然有充滿資訊的圖片資料。為了處理此資料，我使用 vgg16 pretrained model 接 extractor 和 predictor。Extractor 的部分主要是使用 convolution layers 去將 pretrained model 的結果引導到這份資料集；predictor 則是將 extractor 萃取出的向量使用 linear layer 去預測觀看次數[2]。

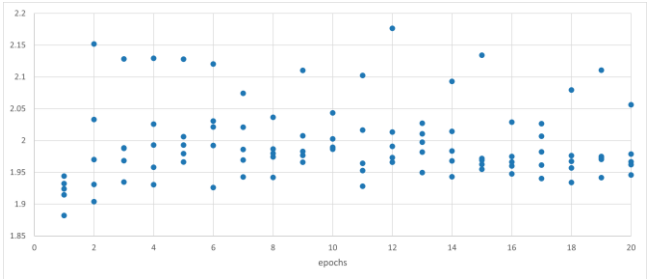
3. 實驗結果與分析

- *圖片模型預測*

在訓練時，我將資料分成 training data 和 validation data，然而不論我嘗試幾次都會發現 epoch 1 的 validation error 會比繼續訓練下去的還要小，例如 epoch 20 或 epoch 50 等。為了觀察是否是因為特定 validation data 造成的緣故，我使用 k-folds，使 $k = 5$ 去計算各 epoch 的 validation error 分布，圖(四)為此結果。由圖(四)可以看到在 epoch 1 的 validation error 會比訓練更多次的結果還要來的好，因此我選擇使用 epoch 1 來當作 testing data 的預測模型。

- *結構化特徵預測*

在結構化特徵預測中，我並未將資料分成 training data 和 validation data，而是直接使用 public testing data 當作我調整超參數的方法。在不加入圖片模型，只考慮結構化特徵的模型預測，其在 testing data 的最佳結果為 2.0165。



圖(四) 各 epoch 下的 5-fold validation errors

- *最佳化結果*

首先，得到分別訓練不同資料的模型 $f_1(x)$ 和 $f_2(x')$ 後，$f_1(x), f_2(x') \in \mathcal{R}$，$x \in \mathcal{R}^{12}$，$x' \in$

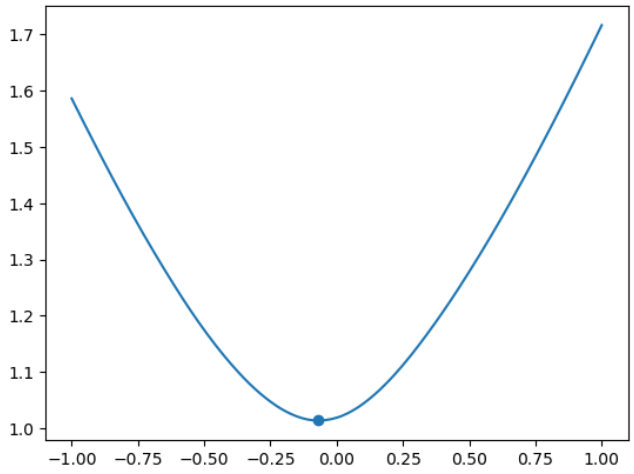
[1] 模型參數可以參考 PredictionModels.ipynb 檔。

[2] 模型架構和參數可以參考 ImageModel.ipynb 檔。

$\mathcal{R}^{224\times224}$，可以解下列方程式：

$$\min w_1 f_1(x) + w_2 f_2(x')$$
$$\text{s.t } w_1 + w_2 = 1$$

此處為了方便起見，我使用 brute force 法算出 $w_1 = -1, -0.99, \ldots, 0, 0.01, 0.02, \ldots, 0.99, 1$下，所有訓練集中的 MAE，得到圖(六)。

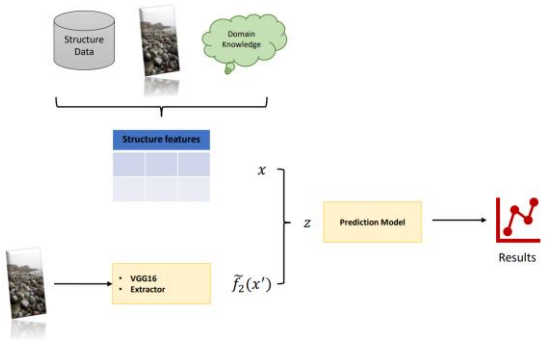


圖(五) 圖片模型的最佳權重

由圖(五)可以看到最佳的權重是圖片模型為-0.07，但這非常可能是 overfitting 的結果。將最佳化結果丟入 testing data 觀察可以得到 2.02101，比完全使用結構化特徵的結果來的差。從上方可以了解到我們沒辦法直接將不同模型預測的觀看人數進行權重最佳化。然而，為了使預測模型能夠加入圖片特徵進行預測，我將圖片模型給予的產出從觀看次數的預測改變成只擷取 extractor 輸出的向量$\tilde{f}_2(x')$，$\tilde{f}_2(x') \in \mathcal{R}^6$。取輸出向量為 6 的原因是，結構化特徵只有 12 個，若將圖像的特徵取太多會稀釋掉結構化特徵的比重。將該向量與結構化特徵合併後，丟入一個新的預測模型進行最後預測。表(二)顯示了結合結構特徵和圖像萃取特徵的新特徵 $z = x||\tilde{f}_2(x'), z \in \mathcal{R}^{18}$後，丟入各種模型在 testing data 的預測結果，圖(六)是最終流程圖。從結果中可以看到加入圖像萃取出的向量後丟入 lightGBM，相較於在結構化特徵預測中來的優秀。然而，若後方是接 Neural Network 則會明顯 overfitting，不論是使用 MAE 還是 MSE 當 loss function。

Final Model	Testing MAE
Linear Regression	2.36639
Lasso	2.35732
SVM	2.50468
lightGBM	2.01453
CatBoost	1.98176
NN L1Loss[3]	3.08171
NN MSELoss	3.26420

表(二) 各最佳化模型在 testing data 的 MAE



圖(六) 最終流程圖

4. 總結與討論

此次作業中我建立了一個預測社交平台觀測次數的模型。在觀察原始特徵後，我使用自身對社群平台觀看次數多寡原因的了解計算出較富含意義的特徵，例如發文者的發文次數，文章 tags 的數目和品質等。值得注意的是，為了驗證我自行設計的特徵是否有效，我將這些新特徵對觀看次數線性迴歸後發現在線性模型中也非常顯著。

除了結構化特徵外，圖像的品質對於觀看次數的多寡也同樣有影響，為了加入此資訊，我用了兩個方法。第一是直接訓練出一個讀取圖像並且對觀看次數做預測的模型，接著再將結構化特徵模型和圖像模型的結果進行權重最佳化。然而，圖(六)中顯示此方法並不成功。接著我嘗試訓練一個吐出代表圖像的向量v的模型，$v \in \mathcal{R}^6$。將此特徵和結構化特徵結合後，再訓練一預測模型進行最後觀看次數的預測。表(二)顯示了各模型下在 testing data 的結果。

[3] 模型架構可以參考 PredictionModels.ipynb 檔。

附錄

由於本次作業的模型較多且為了保留模型建立順序和方便觀看資料的特性，我選擇使用 ipynb 檔建立模型和資料分析，並且將不同模型分別在不同 ipynb 檔中建立，將每階段的資料輸出成 csv 檔供其他階段使用。StructuredFeatures.ipynb 是在處理結構化特徵的檔案，其中包括了利用 ResNet152 和 BERT 去訓練圖片與 Category，Concept 和 Subcategory 的模型；ImageModel.ipynb 則是訓練萃取圖片資訊的模型，其中包括了圖(五)中 k-fold 的程式；PredictionModels.ipynb 則是預測觀看次數的模型檔案，包含表(一)的線性迴歸分析、結構化特徵預測的 lightGBM、最佳化權重和表(二)各預測模型。