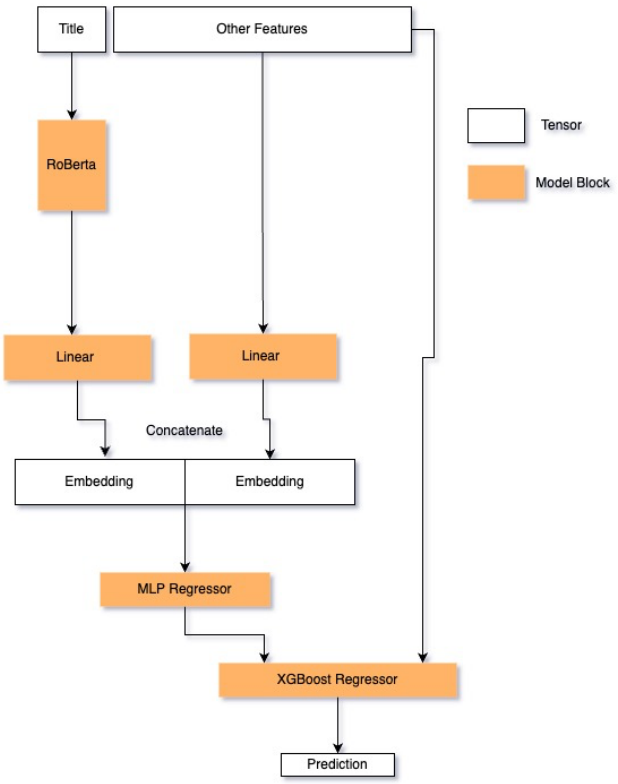
Dcard Intern Homework

1. 最後模型架構



2. 最後結果

訓練設備是 RTX 4090 + Intel i7 7700

Models/Metrics	MSE	MAE	MAPE	Training Time
XGBoost + RoBerta + MLP (Multi Layer Perceptron) **[Final]**	12437	19.3	0.301	27hr 13min
CatBoost+ RoBerta + MLP (Multi Layer Perceptron)	13455	22.3	0.412	27hr 1min
RoBerta + MLP	12155	23.3	0.406	26hr 47min
Linear Regression	14353	26.5	0.457	1min
Bert + SVM	19569	31.4	0.718	20hr 3min
LightGBM + MLP	14051	25.1	0.551	17hr 3min
MLP	13631	24.9	0.414	16hr 3min
XGBoost	12651	24.1	0.535	6min

若以上的模型有以下的參數，則都一樣（像所有 Tree model 的 tree number 都是 1526），沒有寫出來就代表用 default

Parameters	Tree number	Tree learning rate	Tree Max depth	MLP Learning rate	Transformer lr (Bert, etc)	Batch size	epochs
value	1526	1e-2	10	1e-4	3e-5	64	256

3. 想法

首先觀察了一下此 dataset，前 6h 的 likes count 跟 24h 的 likes count 的 correlation 有 0.7，所以先拿了 linear regression 試試水溫，由於 title 是中文，沒想到很好的方法加入 features 就先拿掉；而時間的部分是只擷取小時、分鐘和秒（日期不會重複，故不採用），得到上面表格的結果。

接下來，由於這是一個 regression 的 task，用 Multi Layer Perceptron 一直都是一個還行的解法，一樣沒有加入 title 後實驗得到和 Linear Regression 差不多的結果。

到了這個時候，我就在想因為 forum_id 和 user_id 都是 categorial，沒有數值含義且 category 太多，one-hot 會爆炸所以轉向 Tree-base model，直接拿 XGBoost 硬做但效果差不多。

Title 是中文的問題可以用 Word2Vec 等老方法來解決，但因為 Bert 的效果好很多，我就拿中文 Bert 和 RoBerta（效果比較好的 Bert）來擷取詞向量，接下游的 MLP，直接跟其他 feature concat 在一起輸入 MLP，如此的效果有好些；後來覺得 Bert 的 output dim 和其他 feature 的 dim 差很多，會有 unbalanced 的問題，所以把他兩個都通過一個 Linear 層獲得兩個一樣 dimension 的 embedding 後再 concat 起來，MAPE 有降低 0.5 左右。

經過觀察發現>1000 likes 的 item 十分稀少，無法學習到高 likes 導致 MSE 會相當高，但對 MAPE 的影響較少；經過實驗使用 SMOTE 或 GAN 來生成 data 並不會讓效果提高，反而因為 validation set 很多少 likes 的 item 而預測不準確。

最後經過一些排列組合（上面表格）和測試，我決定把模型設計成下列 procedure：

1. 把 data 拆成 title (經過 Bert Tokenizer) 和其他 features 兩個 tensor.
2. title 通過 RoBerta 之後會形成一個高維度的 tensor，把他通過一層 linear 後獲得比較少 dimension 的 embedding.
3. 將 features tensor 通過另一個 linear 後獲得與 title embedding 一樣 dimension 的 embedding. (for alignment)
4. Concatenate title embedding 和 features embedding, 通過一個 Multi Layer Perceptron 後獲得一個預測值（預測 24h likes count, loss 是 MAPE）
5. 因為某些 features 是 categorical，使用 XGBoost 來增強理解, 將一開始的 features tensor 和 MLP 的預測值輸入 XGBoost, 獲得最後的結果。