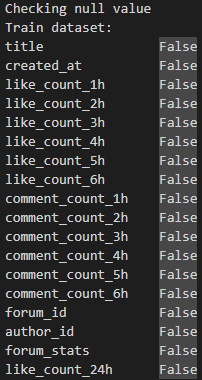
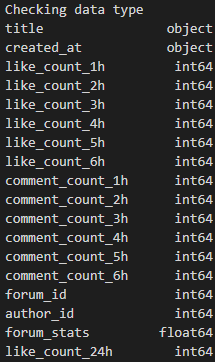
首先拿到資料集的第一件事，就是要先對資料有一定的了解及處理才可以開始做模型訓練，所以我先對資料集的每一個feature去檢查是否有遺失值的存在，得到的結果如下圖。



確認資料集皆沒有遺失值後，再來了解每一個feature的data type為何，以幫助我在選擇模型及資料預處理上有大致方向，結果如下。



對資料集有一定了解後，我選擇將“created\_at”、”forum\_id”、”author\_id”和”forum\_stats”這四項與預測結果相關度不高的features移除，並且因為title的型別為string，要可以與其他features併入一起做訓練的話，要將string轉換成數字的型別，如int或float。此時，我想到我學過的transformer的觀念，我決定將title去做word embedding將其轉換成數字vector再與其他features串接做成訓練集及驗證集。

根據網上查找到的資料，我選擇利用pretrained好的Bert model去做tokenization，並利用model的output取得embedding的結果，每一個title固定皆為768維度。

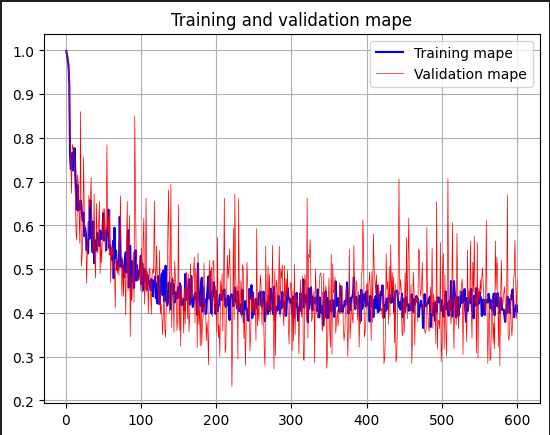


但若是直接將768維的word embedding title和剩下12維的feature結合在一起放入neural network去做訓練，得到的結果貌似模型會逐漸偏重那768維的feature，使得原本給定的累積愛心數和累積評論數派不上用場，在訓練後期的mape不降反升，調整後的結果約在70%~80%。

在調整model的期間我有嘗試使用Random forest和Polynomial regression去做結果評估，所得出的結果比先前的NN再上升一些，約在85%~90%。

考慮到word embedding features影響可能有一點大，於是我利用一個自定義的BertNetwork先將768維經過兩層運算降維成64維，並利用另一個FeatureNetwork將累計愛心數和累積評論數從12維經過一層運算升維到64，再將兩者串接起來的128維放入最終訓練的四層Network，所得到的結果有明顯的下降一些，約在55%~65%。

但在經過Hyperparameters的調整及運用Learning rate scheduling逐步降低lr，所得到的結果也只有使mape降低幾個百分點。此時我注意到訓練過程的loss率震蕩不已，從幾百到幾萬都有，我便嘗試使用SmothL1Loss去取代MSE loss做模型訓練，得到的結果讓訓練過程沒有那麼震蕩，mape更是下降了不少，約在40%左右。



最後再對模型架構和其他Hyperparameters做些許調整，便形成了最終的模型，

後記：

在繳交前一個晚上突然有想到利用LSTM之類的具有時間序列特性的模型對累積愛心數和累積評論數做處理應該會是好選擇，但礙於近期為期中考周，時間所剩無幾所以只能先選擇以目前模型繳交。

期中考周結束後會嘗試將以上的想法付諸實現，實驗看看結果是否會變好，後續會再推上Github上。