Data Mining HW3

1. Describe how you solve this problem. Details include preprocessing, embeddings, model selection, hyperparameters should be provided.
   - **Baseline：**

     讀取提供的實體向量 `train/test_entity_embedding.vec`（約 100 維），新聞向量 = 實體均值，用戶向量 = 歷史新聞均值，分數 = 點積，無訓練，直接生成提交。
   - **GRU:**
     - **歷史序列處理：**將用戶 clicked_news 轉為新聞向量序列，截斷最近 max_history_len（預設 50），用零向量右填充到固定長度，便於 Mask。
     - **模型：**歷史序列 (max_history_len, dim) + 候選新聞向量 (dim)。歷史序列經 Masking(0) -> GRU(hidden=128 默認) 得到用戶向量。候選經 Dense(hidden, ReLU) 投影後與用戶向量相加，再 ReLU，最後 Dense(1) + Sigmoid 輸出點擊概率。
     - **訓練：**對每條行為的每個帶標籤候選生成 (歷史序列, 候選向量, 標籤) 樣本；shuffle 後 batch（預設 128），使用 BCE 損失，Adam(1e-3 默認)，訓練 epochs=2 默認。
   - **TF-IDF + Sentence-BERT + MLP:**
     - **文本稀疏特徵：**HashingVectorizer（n_features=2**18，英語停用詞，lowercase=True，norm='l2'，`alternate_sign=False）生成 TF-IDF；用戶向量取歷史平均。
     - **句向量：** SentenceTransformer("all-MiniLM-L12-v2") ， batch=512（encode batch=64），歸一化；用戶句向量為歷史平均，無歷史回退零向量。
     - **特徵：**每個 user-candidate 組合構造[tfidf_cos, dense_cos, user_dense, cand_dense, user_dense * cand_dense]，其中前兩項為餘弦相似度，最後一項為逐元素乘積交互。
     - **模型：**兩層 MLP（256-ReLU + Dropout0.2，128-ReLU，Sigmoid 輸出），BCE 損失，Adam(1e-3)。訓練使用前 30k 行行為（max_train_rows=30000），batch_size=2048，epochs=4，驗證 5%。
     - **推理：**按同樣特徵生成概率，不足 15 候選補零，輸出分數。

2. Choose a **variable (e.g. different model, different approach) <u>excluding hyperparameters</u>** and compare their performance. Explain what causes the difference of performance or why.

Method	Score
Baseline	0.5436
GRU	0.6140
TF-IDF + Sentence-BERT + MLP	0.6963

Baseline 只用實體均值點積，實體覆蓋有限且缺詞級語義，很多新聞退化成近零向量，導致相關性弱，得分最低。

GRU 仍只吃實體均值，雖然加了序列建模能利用最近歷史，但輸入信息不足，提升有限。

TF-IDF + Sentence-BERT + MLP 同時用詞級匹配（TF-IDF 餘弦）和句級語義（SBERT），再用逐元素乘積顯式建 user–candidate 交互，信息量和交互力最強；對無實體、短標題或噪聲實體的樣本也更穩健，因此得分最高

3. Do some error analysis or case study. Is there anything worth mentioning while checking the mispredicted data? Share with us.
    - 相似候選的相對順序：同一 impression 內多條高度相似的新聞，模型給出接近分數，微小噪聲導致排序錯位；可加入去重或候選重排策略緩解。
    - 時間漂移：歷史多為舊內容時，對新事件打分偏低，錯殺近期熱門；時間權重或近期窗口能減少此類誤差。