IR ASSIGNMENT 4 REPORT

Libraries Used:

1) pandas and numpy for data handling and numerical processing.
2) re and nltk for text preprocessing; re allows me to use regular expressions and find patterns that I want to replace and nltk provides library implementations for lemmatization.
3) sklearn for splitting the data into train and test sets.
4) pytorch and transformers to load the pretrained “gpt-2” model and tokenizer, finetune the model for the task of review text summarization.
5) tqdm to track my run.
6) rouge_score library to calculate the rouge scores and evaluate the model.

Code Flow:

1) Randomly sample 3200 data samples.
2) Preproces the review text and summary.
3) Split the data into train and test sets.
4) Make a reviews column as follows:

   concatenate the review text and summary with TL;DR
   TL;DR here is the task designator which helps the model understand the task it has to perform in case the text is followed with a TL;DR.

5) The mean length of the reviews column created came out to be 84. So I went ahead with the assumption that most of the text in the review should fall within the range (2*mean_length).
6) Prepare a dataset class and data loader to prepare input sequences for model training and batch the data for quicker training.

7) Calculate loss at the beginning of each epoch.

```
Epochs:   0%|          | 0/5 [00:00<?, ?it/s]
Loss: 10.938871383666992
Epochs:  20%|██        | 1/5 [02:23<09:33, 143.32s/it]
Loss: 3.130673408508301
Epochs:  40%|████      | 2/5 [04:48<07:13, 144.42s/it]
Loss: 3.147874116897583
Epochs:  60%|██████    | 3/5 [07:13<04:49, 144.84s/it]
Loss: 3.1942152976989746
Epochs:  80%|████████  | 4/5 [09:39<02:25, 145.05s/it]
Loss: 3.140655040740967
Epochs: 100%|██████████| 5/5 [12:04<00:00, 144.92s/it]
```

8) For summary generation during model evaluation, I use the model.generate to generate output sequences for a given encoded sequence (review text + " TL;DR ") and then decode the output sequence using tokenizer.decode()

9) Calculate the rouge scores for the test set.

```
Average Precision Rouge1: 0.036961541505995184
Average Recall Rouge1: 0.5394280308507282
Average F1-Score Rouge1: 0.06804923278510763



Average Precision Rouge2: 0.007060701557238087
Average Recall Rouge2: 0.1308815073815074
Average F1-Score Rouge2: 0.01316072319721092



Average Precision RougeL: 0.029185710586058233
Average Recall RougeL: 0.44607799455661296
Average F1-Score RougeL: 0.05392174730772774
```

Arguments Used in “model.generate()” :

1) max_length: Maximum length of the generated output
2) num_beams: The number of candidate tokens generated while the model predicts the next token in order to generate the output. The token with the highest probability is picked as the next token.

note: gpt-2 is a causal language model, it predicts the next token conditioning on the previous tokens.

3) repetition penalty: penalize repeated tokens in the output sequence. (>=1 → penalize repeated tokens, <1 encourage repeated tokens).
4) length penalty: penalize longer output sequences. (>=1 → penalize longer output sequences, <1 discourage longer output sequences)
5) early_stopping: Stop inference when eos token is generated.