**REPORT NLP TASK 2**

**1.** **Data Preprocessing**

The initial step involves reading a JSON file containing user posts and preprocessing the data. The `preprocess\_json` function is utilized to extract relevant information, including usernames and post texts. The preprocessed data is then converted into a Pandas DataFrame and saved as ‘preprocessed\_data.csv'. Sample data is taken for the demonstration process

**2.** **Topic Extraction**

To identify topics for each user, the code utilizes the OpenAI GPT-3.5 Turbo model. The `topics\_per\_user` dictionary is populated by making API calls for each user's posts. A personalized prompt is constructed to generate topics related to the user's posts. The identified topics are then saved in 'user\_profiling.csv'. The file has username and interested topics of the user. For Wikipedia article Wikipedia api be used for getting article names.

**3.** **Post-Level Topic Analysis**

For each post in the dataset, the GPT-3.5 Turbo model is employed again to extract topics. The same prompt is used to summarize the post and give the relevant topics. The `post\_topics\_list` is generated by making API calls for each post. The results are stored in a DataFrame, and the associated post topics are saved in 'post\_topics.csv'. This file has the post and the topics summarized from the post.

**4. User Recommendations**

User recommendations are provided for each post based on the similarity between post topics and user interests. The Jaccard similarity and NDCG scores are calculated to measure the relevance of recommendations. The top users are determined and stored in 'post\_recommendations.csv'.

The similarity between a post and a user's interests is calculated using the Jaccard Similarity coefficient. The Jaccard Similarity is a measure of similarity between two sets and is defined as the size of the intersection divided by the size of the union of the sets.

Intersection:

- It represents the number of topics that are common between the post and the user's interests.

Union:

- It represents the total number of distinct topics between the post and the user's interests.

Jaccard Similarity:

- It is calculated as the ratio of the intersection size to the union size.

This similarity score reflects the proportion of common topics relative to the total unique topics in both the post and the user's interests. The resulting Jaccard Similarity ranges from 0 to 1, where 0 indicates no common topics, and 1 indicates a perfect match. The higher the Jaccard Similarity, the more similar the post and the user's interests are considered to be.

**5. Evaluation of Model Performance**

The code evaluates the recommendation model's performance by calculating the average Jaccard similarity and NDCG scores across all posts. These metrics offer insights into the effectiveness of the model in suggesting relevant users for each post. The score is calculated by taking actual users and recommended users . Actual users are the users who posted the text and recommended users are users who are recommended to the post .

Jaccard similarity is employed to evaluate the overlap between the users who actually engaged with a post (represented by actual\_user) and the users recommended by the model (contained in recommended\_users).

* The intersection of these two sets is computed by finding the common elements using the & operator, and the union is determined by combining all unique elements from both sets using the | operator.
* The Jaccard similarity is then calculated as the ratio of the intersection size to the union size, providing a measure of how well the model's recommendations align with the actual user engagement. This approach ensures that the Jaccard similarity is meaningful and avoids division by zero by checking if the union size is greater than zero before performing the division.
* Average Jaccard similarity score : 0.5

NDCG Scores

Relevance Scores:

Assign 1 to the actual user, 0 to others.

Discounted Cumulative Gain (DCG):

Sum relevance scores with logarithmic discount.

Ideal DCG (IDCG):

Max possible DCG, using sorted relevance scores.

Normalized DCG (NDCG):

Ratio of DCG to IDCG, normalizing relevance.

NDCG ranges from 0 to 1, indicating the quality of recommendations—higher values mean more accurate suggestions.

Average NDCG score : 0.69

**6. Execution Time Measurement**

Execution time is measured using the `time` module. The code captures the start and end times, calculates the elapsed time, and prints the result. A loop is included for demonstration purposes.