Post Recommendation System

Assessment Report

Sahil Awasarkar

1 Introduction

The ability to recommend the right content to the right user is a crucial part of most digital platforms today. This assessment focuses on building a simple yet structured pipeline for a post recommendation system. The core objective was to recommend the top three posts for each user by leveraging their profile interests, past engagement patterns, and the attributes of available posts.

The work was carried out in a Jupyter Notebook and is structured to mimic a real-world data science workflow. The notebook begins with loading and checking data quality, followed by exploratory data analysis (EDA) to understand the characteristics of the users, posts, and engagement data. From there, we move toward constructing a baseline recommendation approach and discussing possible ways to extend and improve the model in the future.

2 Approach

The entire approach can be thought of as a pipeline with multiple stages:

2.1 Data Loading and Validation

The first step involved importing the three provided datasets: `Users.csv`, `Posts.csv`, and `Engagements.csv`. Basic validation checks such as verifying data types, checking for missing values, and ensuring consistent IDs across datasets were performed. These checks ensured that subsequent steps were reliable and did not run into issues caused by data inconsistencies.

2.2 Exploratory Data Analysis (EDA)

EDA was a central part of this task. We looked at user demographics such as age distribution and gender proportions, which helped in understanding the composition of the platform's user base. Post-level statistics, including the distribution of content types, provided insights into the variety of available content. Finally, the engagement data revealed how users interacted with different types of posts, showing which content categories attracted higher activity.

Several patterns were discovered during this stage. For instance, younger users were observed to engage more actively, while certain content formats consistently outperformed others. These findings guided the recommendation logic by highlighting what attributes should be prioritized.

2.3 Feature Engineering

In order to make meaningful recommendations, it was necessary to combine information from different datasets. Features such as a user's preferred content type, their historical engagement frequency, and the popularity of posts in the wider community were engineered. This step essentially transformed raw data into useful signals that a recommendation model could leverage.

2.4 Recommendation Logic

For the assessment, a simple ranking-based strategy was applied. Each user was matched with posts by considering two main signals: their past engagement behavior (interests inferred from history) and the popularity of posts in the dataset. By blending personal preferences with global popularity, the system was able to suggest the top three posts for each user in a balanced manner. While this is a basic method, it serves as a solid foundation on which more advanced approaches can be layered.

3 Metrics

Evaluating a recommendation system requires well-defined metrics. Although the assessment did not involve exhaustive model evaluation, we identified the following metrics as most appropriate:

- **Precision@K:** Measures how many of the recommended posts are actually relevant to the user. A high precision means the recommendations are accurate.

- **Recall@K:** Measures how many of the relevant posts the system was able to retrieve out of all possible relevant posts. High recall indicates broader coverage.
- **Hit Rate:** A simpler metric that checks whether at least one of the top recommendations was relevant. This is often used in industry for quick evaluations.
- **Engagement Lift:** Beyond offline metrics, real-world systems often measure whether recommendations lead to higher likes, shares, or comments compared to non-recommended content.

4 Possible Extensions

The assessment mainly aimed to set up a pipeline and demonstrate baseline logic. However, several extensions can greatly improve the system:

- **Content-Based Filtering:** By analyzing post text, hashtags, or images using natural language processing or embeddings, we can recommend posts that are semantically similar to what a user has liked before.
- **Collaborative Filtering:** Leveraging similarities between users (for example, matrix factorization or neural collaborative filtering) can help capture hidden patterns that are not visible from raw attributes.
- **Hybrid Models:** Combining popularity signals, collaborative filtering, and content features usually produces the best results, as it balances personalization with trending content.
- **Context-Aware Recommendations:** Factoring in time of day, device type, or session activity can make the system more dynamic and relevant.
- **Real-Time Personalization:** Updating recommendations in near real-time as users interact with new posts would improve responsiveness and engagement.
- **Explainability:** Providing short explanations such as "recommended because you liked similar posts" can increase user trust and adoption of the system.

5 Conclusion

This assessment showcased a step-by-step approach to building a simple recommendation pipeline. Starting from basic validation and exploratory analysis, the notebook gradually progressed toward a recommendation method that combined user preferences with post popularity.

The exercise highlights the importance of structured data analysis, careful feature engineering, and the value of explainable recommendations. With additional enhancements such as collaborative filtering and deep learning approaches, the system could evolve into a production-ready recommendation engine capable of serving millions of users.

Prepared as part of assessment exercise.