Building a News Recommender for 'JhakaasNewsVala'
Strategy and Implementation Plan

Research :

Building a recommender system.The system helps users discover personalized items of their own interests. Some platforms, such as Spotify and Netflix, rely on the accuracy of the recommendation system to Maximize user stickiness and satisfaction. The successful application of the recommendation system is a powerful means to improve the profitability of commercial platforms. Here on the similar lines we are preparing a recommender system for **'JhakaasNewsVala'.** Typically, recommender systems are based on content-based and collaborative filtering.

1. Content-Based Filtering:

- This technique recommends news articles to users based on the similarity between the content of the articles and the user's preferences or past interactions.
- Content-based methods often utilize natural language processing (NLP) techniques to analyze the textual content of articles and user profiles. Techniques like TF-IDF, word embeddings, or deep learning models can be employed to compute similarity between articles and user preferences.

2. Collaborative Filtering:

- Collaborative filtering recommends news articles to users based on the preferences of similar users or groups of users.
- Collaborative filtering methods can include user-based or item-based approaches. User-based methods recommend articles based on the behavior of similar users, while item-based methods recommend articles similar to those previously interacted with by the user. Matrix factorization techniques such as Singular Value Decomposition (SVD) or matrix factorization with deep learning models are commonly used in collaborative filtering.

Exploratory Data Analysis:

- News Article Corpus from Kaggle

- Contains 210,294 records ,2012 - 2022.
- **Category:** Category article belongs to
- **Headline:** Headline of the article
- **Authors:** Person authored the article
- **Link:** Link to the post
- **Short_description:** Short description of the article
- **Date:** Date the article was published

-User Data Generated (faker library)

- **User ID:** Unique identifier for the user.
- **Name:** Name of the user.

- **Email:** Email address of the user.
- **Interested Categories:** User's preferences include Science, Health, and Business.

-Clickstream Data for User (faker library)

- **User ID:** Unique identifier for each user who interacted with the articles.
- **Session ID**: Identifier for each session during which the user interacted with the articles..
- **Article ID:** Unique identifier for each article that the user interacted with.
- **Article Rank:** The rank or position of the article within the context of the user's interaction session.
- **Time Spent:** The duration of time that the user spent interacting with the article.

Build Strategy :

-Phase1:

- Data Generation and EDA: Dataset selection and user profile , clickstream data generation .
- Implementing content based recommendation for selecting articles to serve certain users.
- Developing user interface
    - Backend - Flask
    - Frontend - Bootstrap , Jinja
    - Database - FAISS

-Phase2:

- Experimenting with multiple Collaborative Filtering (Matrix Factorization, KNN) approaches.
- Dockerization: Creation of Dockerfile for creating docker images as and when required.
- Developing Flask API.

Quality of the Proposed Solution

- **Balanced Strategy:** Combining content-based filtering and collaborative filtering ensures relevance for both new and returning users.
- **Flask API:** Ensures smooth communication between frontend and backend, facilitating easy integration.
- **Dockerization:** Provides consistent, reproducible environments, enhancing scalability and maintainability.
- **Robust Implementation:** Using Flask, Bootstrap, Jinja, and Docker ensures a solid, maintainable infrastructure.

Role of team members:

1) Harjaspreet Singh: Data generation, EDA, Build Strategy
2) Divjot Singh: Quality Assurance , Testing, Build Strategy
3) Manish: Research, Application, Build Strategy
4) Harsehraab Singh: Build Strategy, Dockerization, Flask