Proposal for News Recommender System

Group Name: SASS

Dhruva Sambrani, Gowri A, Rupali Sharma, Shiv Shankar Singh

April 5, 2022

Contents

Data collection pipeline

The first point of any machine learning algorithm is to collect a corpus of data to cater to a wide variety of user interests. This data needs to come from the user if we want to make it personalised. However, when we onboard a user, we do not have any data which is personal to them. This leads to what is known as the cold start problem. We will start by making two vector databases: one for user info and one for article items.

User Clickstream dataset (User Profiler Bot)

Once the user starts consuming news, they leave behind a clickstream which can then be used to uncover user interests and provide personalised recommendations. The data we collect will be in the following manner -

- User ID - given by the app
- Session ID - each new visit to the app
- Article ID - unique id assigned to each article
- Click - A boolean data that tells us whether the article was clicked or not.
- Time Spent - The amount of time spent by the user. This acts as a proxy for how much the user liked the article.
- Like/Dislike - Rating of an article from the user. Correlated with the time spent.
- Agree/Disagree/Uncared - Whether the user liked, disliked or uncaring.

While the time spent by the user may not always positively correlate with the user rating, the long time spent on the article may still imply that the user cares about the topic. Agree/disagree may also be highly correlated with like/dislike, but this data allows us to prevent the users from becoming part of echo chambers, with news reinforcing user bias.

News article dataset

To supply data for the cold start problem, we will scrape news articles from multiple news sources. We will select the top 5 most viewed news sources within India and the top 5 globally. We collect and make a database of the following attributes.

- Article title/headline
- Newspaper name/id
- Date of publication
- Author name
- Source provided tags if any

We then analyse these articles to try and apply sentiment analysis to get some further information about the articles.

- Article ID - A hashed version of the title
- The model defined sentiment tags.

New articles are collected with the same information as before and shown to the user according to the recommender algorithm.

Implementation (Recommender Bot)

First, we will tokenise and clean the raw data we get from scrapping by standardising, stemming, lemmatizing, and removing stop words. We will cluster the articles into different categories and recommend one from each category to the new user. A Corpus of recent (fresh) news articles is created and tagged using sentiment analysis.

To cluster these, we will use different vectorisation models such as those given below:

- TF-IDF - assigning a weight amplifies the importance of a word in a document
- LDA - can be used to understand how much an article is devoted to a particular topic. It also helps to reduce the dimensions of the data.
- Word2Vec
- Hashing

Cold-start problem

We plan to recommend one article from each category for new users for new users, providing them with a diverse set of options. It will be done till we have 10 data points in the user clickstream dataset for the user.

Content-Based Systems

These systems recommend articles with similar content as the ones the user liked in the past. The user profile is used here, hence mostly useful for old users. A combination of minhashLSHforest, which performs Locality Sensitive Hashing, and random forest, and Hashing Vectorizer can be used.

Otherwise, a TFIDF based approach generates a tfidf_matrix with the words and corresponding scores. We then calculated the cosine similarity between the whole corpus and the user interacted article. This way, new items are compared to the user profile, and 10 of the most similar ones are recommended.

We can also use a topic modelling based approach that uses LDA and cosine similarity to recommend similar items.

Collaborative Recommendation system:

To diversify news recommendations and minimise the cold-start problem, this approach will be combined with a content-based recommendation system. It will recommend articles based on content consumed by similar users clustered using clustering algorithms. From this system also, 10 article recommendations will be obtained.

Collaborative filtering will be done based on :

- user similarity score- by creating user v/s article rating matrix and using it to find the cosine similarity between different users
- Matrix decomposition- creating user v/s article rating matrix and reducing its dimension using the Singular Value Decomposition method. The user v/s article matrix will be re-calculated to recommend articles based on the estimated ratings.

Hybridized Recommendations:

After getting 10 recommendations from both the filtering methods, we plan to score the articles again using the scoring method from content-based filtering. Then choose the top 10 recommendations, which are finally provided to the user.

Reducing Bias

By collecting more news articles from various sources and making a database with diverse user read database of article we can reduce the bias.

Also by assigning higher weightage to the scores assigned by the active user and lower weightage to the new/flimsy user we can reduce the error.

For stories that get served often after first few serving the weightage of the score assigned to the article for further serving will be lowered to reduce bias.

Evaluation

The ultimate objective is to increase click-through. The standard metrics to evaluate is to set a threshold chosen as the median value of time spent or the rating obtained from the user. We can apply the mean average precision and recall metrics to see how good/bad our recommendations are. We can also set the threshold using the Receiver operator characteristic curve.

If we're using the k-means clustering algorithm, we can use the silhouette score to decide how many clusters we can form from the data. This score refers to how similar an object is to its cluster.

Further Application

We can use this strategy to make a recommender for scientific articles on arXiv and music/ YouTube video recommendation.