Transparent Recommendation System Minimizing Bias

Team #36 - Elaine Bradley, José Zavala, Kelvin Chou, Caeden Meade, and Mingyuan Zhang

Summary
Current recommendation systems, such as those used in social media, tend to polarize users toward content aligned with their pre established beliefs. This polarization can have a negative impact on society, driving people apart rather than encouraging unity and collaboration between people. There are consequences of severe polarization, particularly in the realm of news when biases and misinformation become a problem, such as the case with COVID-19 misinformation.

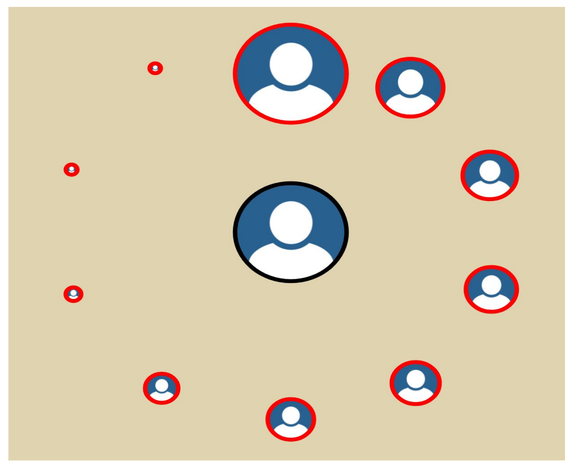
What are current recommendation systems like?
The disadvantages or flaws of existing recommendation systems are that they are geared towards keeping users on their platform, ignoring biases since polarization can maintain engagement, and lacking transparency behind the recommendations to the users. This lack of transparency, and tendency to recommend polarizing and biased content leaves room to improve existing systems.

Why does transparency matter?
When users are in control of the information that is presented to them, they are allowed to explore their recommendations according to their questions. Users build trust through learning why they were recommended certain content. In addition to providing users with more autonomy in their decisions, transparency would allow users to understand why they are being recommended specific content and it helps visualize trends or biases in their recommendations.

Algorithm approach
Utilizing a combination of user-based similarity and content-based similarity, we generated a list of recommended articles to present to the user. In order to determine similarity between users and articles or articles and articles, we implemented a cosine similarity calculation across the data. This cosine similarity function compares the user and article vectors for each data entry and two different articles as well.

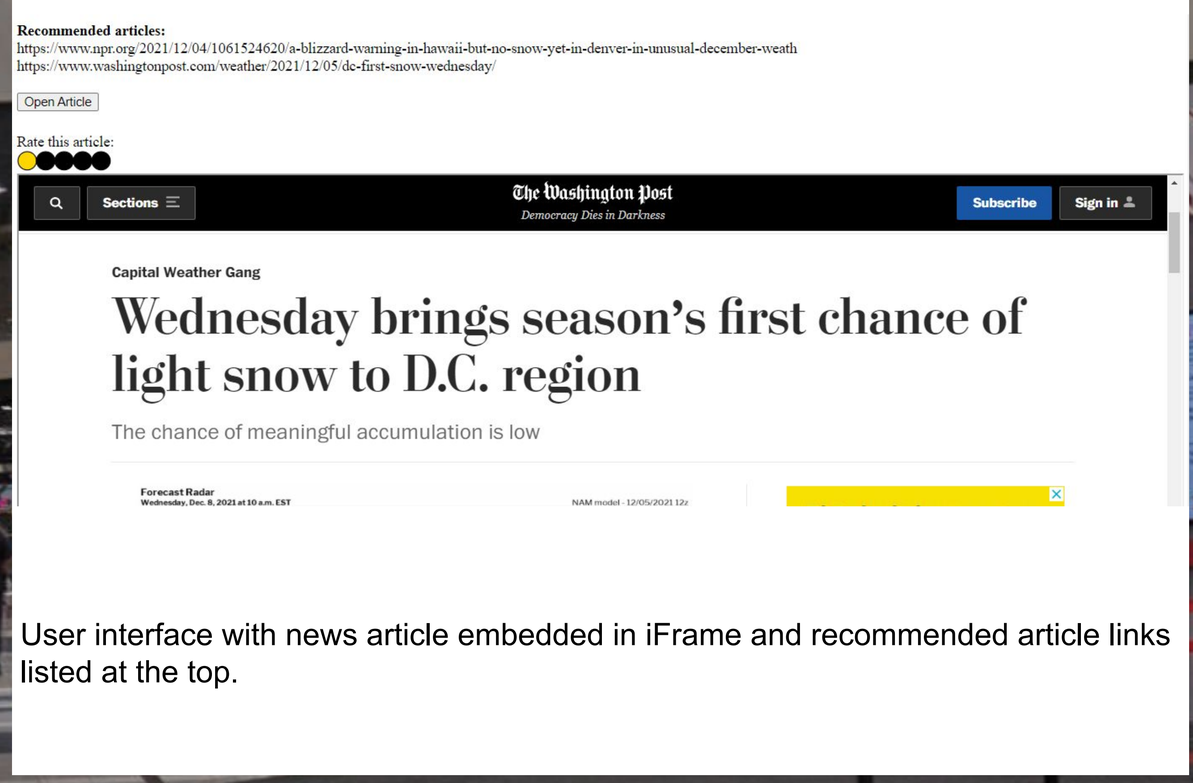
Dataset
In order to minimize bias, we tried to limit our news data source to news articles with minimal bias. For our experiment, we worked with a static dataset of roughly 200,000 articles. This could be expanded via API or data scraping with future implementation of our project. In regard to our user dataset, we generated a user base of 30,000 users and associated various ratings for a multitude of user-article relationships. Our final user-article relationship dataset ended up being just over 600,000 relationships.

Visualization
In order to provide transparency about recommendations, we visualize the relationship from the user to the top 10 similar users from the user database. To visualize the correlation of the relationship between each user, the size of the similar users' nodes adjust based on how similar the relationship is. The larger the node, the more similar the two users are to each other. The recommended articles then branch out from each similar user node.



Visualization of current user similarity strength to top 10 most similar users.

User Interface
The web framework for our user interface was structured with Django. Within our framework, we have various HTML files for the visualization and recommendation page. Users must log in before they can receive recommendations so that we are able to build upon further recommendations. Once a user is logged in, the can browse recommended article based on their user history. In order to view visualizations of their recommendations, they can simply click a button to toggle the visualization on or off.



User interface with news article embedded in iFrame and recommended article links listed at the top.