**OneDrive link for all the files:** <https://iiitbac-my.sharepoint.com/:f:/g/personal/chaithanya_challapalli_iiitb_ac_in/El-um7SSOnNPqzPvgYDj7MYBilqCRq2TLhMg61D-IsZlAw?e=vRqNrd>

**File descriptions:**

* **Preprocessing.ipynb**: In this file we remove unwanted columns, repeated rows from the dataset. We created an extra column for the news dataset which is a combination of category, subcategory, title and abstract. Then we removed the stopwords from this column.
* **Embeddings.ipynb** : In this file we used the BERT model to get the embeddings for the news articles. We applied BERT on the newly created column in the news dataset. Then we used the user's dataset to find the embeddings for the user which is the average of articles the user read.
* **Different\_user\_embeddings.ipynb**: We created different representations for the user other than the average of articles the user read. We didn’t use these different embeddings to train any model.
* **Without\_kmeans.py**: This file has the basic model for recommendation of articles. We are recommending the articles which are closer to the articles the user read. This has the user interaction.
* **Without\_kmeans.ipynb**: This file has a basic model for recommendation of articles. There are multiple models in this file. Firstly, we directly compared the articles the user read with all other articles and we recommended which are closer to the user read articles. Secondly, we are recommending articles which are closer to the user embedding.
* **Kmeans.ipynb**: In this file K-Means model is used for the recommendation of articles.
* **BPR.ipynb** : There are two models in this file. First one is normal BPR which was discussed in the class and the second one is our novel idea.
* **MAB.py:** We used Thompson’s sampling for the article recommendations. This file has the user interaction.
* **MAB-analysis.ipynb:** We analyzed our Thompson's sampling model using some graphs.
* **Trending.ipynb**: This file has user interaction. This model shows trending news articles based on the time period selected by the user and the number of times the article is clicked.

**Instructions for running the files:**

* **MAB.py**

News articles will be displayed, if you like the displayed article then please enter 1, else enter 0. To exit the user interaction enter -1.

.

* **without\_kmeans.py**

Multiple news articles along with the index will be displayed, please enter the article index which you would like to read. The user has to choose articles from “Main page” twice in the beginning. After that, user either has choice to explore news articles from ‘Main Page’ or they can choose to get recommendations. The user’s state will be updated as and when he chooses new news articles.

Top k recommendations will be shown to the user based on the L2 norm of the difference between user embedding and every other news embedding.

* **Trending.ipynb**

**To get the Trending recommendations:**

Input date and time must be given in the format ‘MM/DD/YYYY hour:minutes:sec AM/PM’ as shown in the below code. date\_trending indicates the exact date and time when the user opens the news application’s Trending section. number\_of\_days & number\_of\_trending articles to be recommended can be adjusted as required.

**date\_trending = '11/11/2019 9:05:58 AM'**

**input\_date = datetime.strptime(date\_trending,'%m/%d/%Y %H:%M:%S %p')**

**number\_of\_days = 1**

**begin\_date = input\_date - timedelta(days=number\_of\_days)**

**number\_of\_trending\_articles = 25**

**print(input\_date)get the Trending recommendations:**