News Article Recommendations System Report

Nikhil Gupta
Student, M. Tech Computer Science Engineering
MT2023187

Nabaneet Dutta Kanungoe
Student, M. Tech Computer Science Engineering
MT2023194

Chittaranjan Chandwani
Student, M. Tech Computer Science Engineering
MT2023193

Nikita Mishra
Student, M. Tech Computer Science Engineering
MT2023052

***Abstract* - Recommendation Systems are machine learning models which help in suggesting items to users by filtering from a large number of choices and using various criteria like user's previous likes and dislikes, view history, product engagement among other factors. In modern times recommendation engines are the need of the hour, due to the increasing amount of data and content available to the users and them being spoiled for choice, increasing the relevance of such systems. News is a very important mode of information broadcasting in modern times. Everyday millions of news pieces are consumed by the users. Also with increasing competition among various media houses it is important to keep one's users close and engaged in your content. This is where news recommendation systems come into play. In this paper various methods for providing recommendations have been discussed, like Content-based recommendation, K-means clustering recommendation, followed by some Bandit algorithms like Upper Confidence Bound (UCB) and Thompson Sampling. Lastly a hybrid UCB-Thompson sampling method has also been tried.**

Keywords: Machine Learning, Recommendation Systems, News, Bandit Algorithms.

I. INTRODUCTION

In the modern age, where a vast amount of information is available at one's fingertips, navigating through content has become increasingly complex. News consumption, in particular, is a vital aspect of society, shaping opinions, influencing decisions, and providing deep insights into global events. However, with the rapidly increasing sources of news and the amount of articles published daily, users often find themselves thrown in the water with choices, struggling to discover content that aligns with their interests and preferences. Hence, recommendation systems have emerged as indispensable tools, leveraging machine learning algorithms to sift through heaps of datasets and deliver personalised suggestions to users. These systems analyse various user attributes, such as past behaviour, explicit preferences, and demographic information, to generate tailored recommendations that enhance user engagement and satisfaction.

In such a competitive landscape, media organisations strive to retain their audience by delivering relevant and compelling news articles that capture their attention. This is where news recommendation systems play a crucial role, by intelligently curating content based on individual preferences and consumption patterns.

In this project an attempt has been made to create recommendation systems for the news article recommendation problem. The Microsoft News Dataset (MIND) has been used which is a very rich repository of news articles from a variety of genres and domains. The dataset also provides us with user behaviour in the form of the user's past click history and impressions of the news articles in the dataset. Many techniques and approaches exist in the recommendation system domain, some of them have been used in this project.

Content-based recommendation is a simplistic but quite a popular approach that has been implemented here. In this approach one simply uses the 'content' which is news articles in this case to determine what the user prefers. This approach however does not factor in the past history of the user. Hence, to incorporate the user preferences other techniques need to be used. K-Means clustering based approach which allows us to capture user similarities through preferences is another great way of providing recommendations. The methods mentioned until now focus more exploitation of user preferences, however in recommender systems it is necessary to balance exploration and exploitation in order to recommend articles to the user based on their preferences but at the same time also push popular content that the user may not have viewed but can possibly like. Bandit algorithms are an amazing set of algorithms that help us beautifully encourage exploration while not compromising the user preferences and offer a much more personalised user experience with a variety of recommendations. Hence, the popular bandit algorithms like Upper Confidence Bound (UCB) and Thompson Sampling have been implemented. Lastly, an attempt has been made to create a hybrid model using the two bandit algorithms. The performance of all the models and the findings have been discussed towards the end.

II. DATASET AND EXPLORATORY ANALYSIS

For the purpose of this project the Microsoft News Dataset (MIND) has been used. This dataset contains two files, ‘news.tsv’ and ‘behaviours.tsv’ .

- news.tsv- This file contains the news article details like News ID, Category, Sub-category, URL, Title, Abstract, Title Entities and Abstract Entities.

- behaviours.tsv- This file contains user behaviours. It stores the user’s ID, past click history, and impression details.

A basic exploratory data analysis was undertaken to understand the data in a better way, learning about the underlying distribution and subtle details of the data. Discussed below are the results of the analysis.

- There are over 17 categories and 264 sub-categories of news in the dataset.
- There are about 50,434 unique News articles and 50,000 unique Users in the dataset.
- The distribution of news articles is such that the ‘news’, ‘sports’ and ‘finance’ categories are the significant contributors to the dataset.

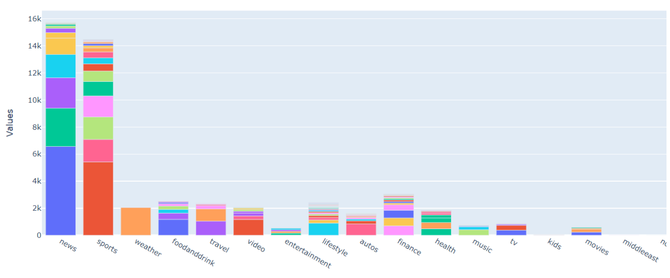

Figure 1: Category-wise distribution of news articles.

Figure 1 depicts the category-wise distribution of news articles along with sub-categories within each news category. From the figure it is clear that ‘news’ and ‘sports’ categories dominate the dataset.

Word clouds were also created to see what kind of news and popular words in popular categories. Figure 2 and 3 display the word clouds for the ‘news’ and ‘sports’ category.

Figure 2: Wordcloud- ‘News’ Category.

Figure 3: Wordcloud: ‘Sports’ Category

Additionally the API provided by NewsApi.com has also been used for creating a real time system in one of the models. The NewsApi.com API provides news from 7 categories- Business, Entertainment, General, Health, Science, Sports, Technology.

IV. DATA PRE-PROCESSING

A. Data Cleaning

- The data cleaning process was started by first removing the duplicate news articles in the ‘news.tsv’ file.
- Thereafter unwanted columns like URL were removed.
- Now the ‘title’, ‘abstract’, ‘title-entities’ and ‘abstract-entities’ columns were concatenated to create a ‘final-string’ column.
- This new column ‘final-string’ will further be used to create the news article embeddings. This also helped in handling the NaN values in cases where abstracts or titles were missing.

B. News and User Embeddings

The recommendation system models require to create embeddings for both the news articles and the users in order to be able to mathematically compare the similarities which can be further used to provide adequate recommendations.

We have created the news article embeddings using two methods:

- First for the content based recommendation model we have created the news article embeddings using the pre-trained Word2vec model.
- The pre-processed 'final-string' column was used to make the news article embeddings using the pre-trained BERT transformer model. The BERT embeddings are quite rich and capture the context as well. These embeddings were used in the other models.

User Embeddings:

- The user embeddings are created using the news article embeddings and the users 'click_history' in order to best capture the users preferences.
- The embeddings are made by averaging over the embeddings of the news articles that have been read by the user in the past i.e. the 'click_history'
- Let the embeddings of each of the news articles read by the user by denoted by N_i, and we have 'n' such news article embeddings. The user embedding for each user U_i is created as follows:

$$U_i = \frac{\sum_{i=1}^{n} N_i}{n}$$

Creating the embeddings in this manner allows to give equal importance to each news article read by the user.

V. RECOMMENDATION MODELS

A. Content-based Recommender

This recommendation model is a very basic and simplistic model. Here the user is first shown 5 articles at random (Figure-4). The user is given an option to select the one which they prefer the most. Based on the user's selection similar articles are then recommended to them.

In this method the Word2vec embeddings were used in order to find similar news articles. Cosine-similarity S_c, between two news articles N_i and M_i is calculated as follows,

$$S_c = \frac{\sum_{i=1}^{k} N_i M_i}{\sqrt{\sum_{i=1}^{k} N_i^2}\sqrt{\sum_{i=1}^{k} M_i^2}}$$

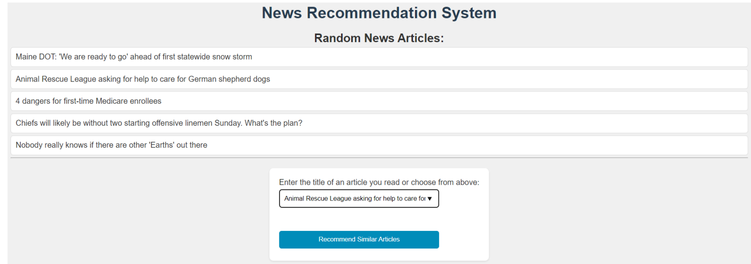

Figure 4: Random articles displayed to the new user

As seen in Figure 4 from the displayed articles the user selects the article related to 'Animal rescue'. Accordingly, the top 5 articles with the most similarity with the article selected by the user are recommended (Figure-5). Content-based recommendation provides quite rudimentary results and can be very useful in handling new users or in situations where we have fairly less knowledge about the likes, dislikes of the user. This model does not factor in the users past click history and thus the users preferences do not have any bearing on the recommendations.

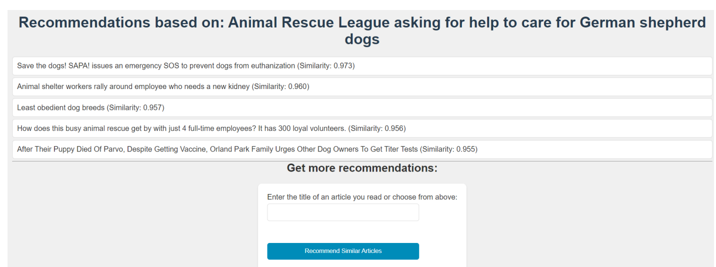

Figure 5: Similar articles being recommended to the user

It can be clearly seen that the recommended articles are similar to the user's choice. Articles about 'Animal Shelter', 'Save Dogs' etc. are being recommended. The high similarity index verifies this claim.

B. K-Means Clustering based Recommender

In this model the K-Means clustering algorithm has been used to cluster the similar users and use these clusters to recommend articles to existing or a new user.

- The elbow method was first used to find the optimal number of clusters in which the users must be clustered in order to provide the best possible recommendations.

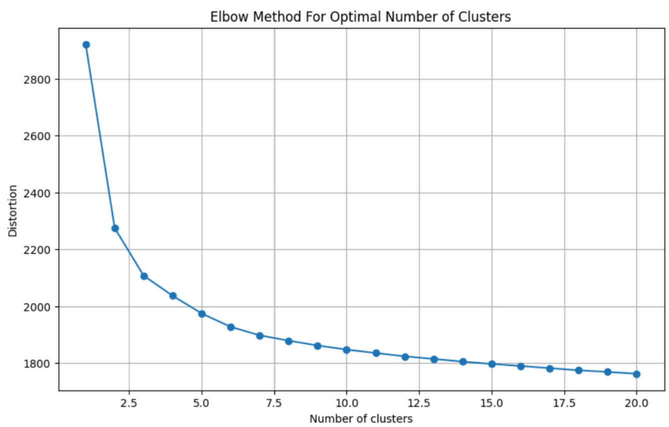

Figure 6: Elbow method results for K-Means clustering

- Figure 6, shows the results of the elbow method. From the results it is clear that the elbow forms around 4-6, 6 were chosen as the number of clusters for the model so as to not loose a lot of subtle details in the embeddings but at the same time also retaining the clusterings.
- Now having clustered the users, the articles read by the users belonging to a particular cluster were aggregated. This led to creating a pool of relevant articles for each cluster.
- When an existing user enters the system, the top 5 most viewed articles from the pool of articles of the cluster the user belongs to and those that they haven't read are recommended to this user.
- For a new user entering the system, the user provides with some articles that they like, using those articles the embeddings of this user are created.
- Now using the K-Means model a cluster is assigned to this new user and in a similar fashion from the pool of articles the top articles are recommended for this new user.
- In order to ensure that repetitions of article recommendations do not take place, every time an article is recommended to the user it is added to the users click history and the database is updated.

The creation of the pool of articles in each cluster helps in recommending a more diverse set of articles to the user as no attention is given to the user's preferred categories during inference. Although the K-Means clustering model focuses more on exploitation of user's choices, this way by creating a pool of articles we introduce a factor of exploration as well, as the top articles recommended may belong to a category that this user may not have ever read.

C. *Upper Confidence Bound (UCB) Bandit Algorithm*

Bandit algorithms are reward based probabilistic models that are quite popular in the reinforcement learning domain. The multi-armed bandits are a model in which a decision maker iteratively picks arms over time by not affecting the properties of the other at the same time. These algorithms exemplify the exploitation-exploration tradeoff in an excellent manner.

Upper Confidence Bound (UCB) is a bandit algorithm which models uncertainty in rewards (action-value) to balance exploration and exploitation. The aim is to maximise the expected reward by selecting the action that has the highest action-value.

- For the UCB model the news categories have been selected as the arms. The UCB score for each arm is calculated as follows,

$$UCB_i = Q_i + \sqrt{\frac{\alpha . ln(T)}{N_T(i)}}$$

  here UCB_i is the UCB score for a category i, Q_i is the exploitation term which signifies the estimated action-value for arm i, while the term under the root is the exploration term. $N_t(i)$ is the number of times the arm i has been picked until time T.
- The model keeps the track of action-values, UCB scores and the counts to facilitate recommendations.
- Categories are selected by calculating their UCB scores and picking those with highest value.
- When a user interacts with the article, the model updates the values and scores based on the feedback.

Additionally, we also simulated the UCB algorithm for a particular user over 2000 iterations to depict the algorithm's exploration and eventual convergence to oracle probabilities. Figure 7 displays the results of the same.

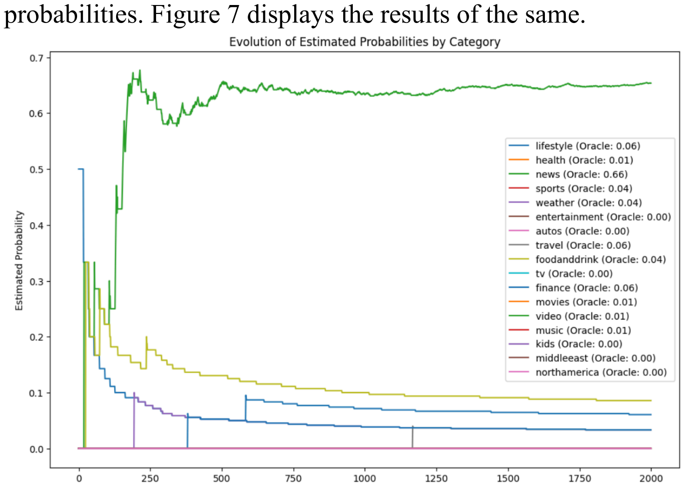

Figure 7: UCB simulation

D. Thompson Sampling

Thompson Sampling, also known as Posterior Sampling is another Bandit algorithm popularly used in reinforcement learning and recommendation systems. It is a probabilistic model which addresses the exploration-exploitation problem in the multi-arm bandit algorithmic domain. Thompson sampling beautifully balances the exploitation of available information along with encouraging exploration.

- Each Category (17 in total) in the dataset is considered an arm of the bandit. Due to the binary reward nature we use the Bernoulli distribution and because of which beta distributions are assigned as priors to each category (arms).
- The beta distribution parameters are initialised based on the users click history . For any category the user has seen the alpha parameter value for those arms are increased.
- Having initialised the parameters the user on entering the system is provided with recommendations based on the parameters (Figure-8) , by sampling from the distributions of each category..
- Now the user is given the option to like or not like the recommended articles (Figure -8). Based on the user's choice the beta distribution parameters are updated and further recommendations can be made (Figure -9).

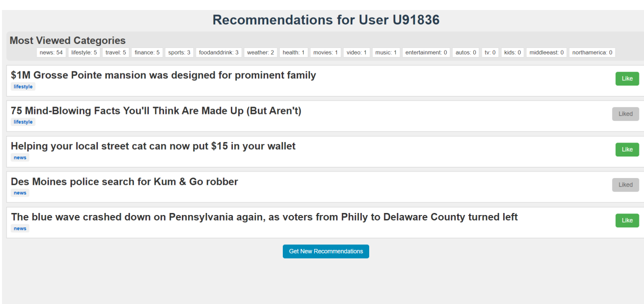

Figure 8: Recommendations based on initialised parameters

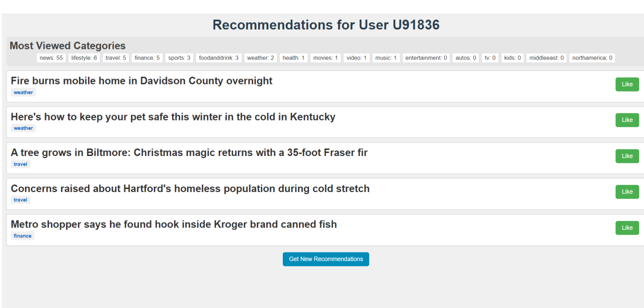

Figure 9: Recommendations from Thompson Sampling

In Figure 8, the articles recommended to the user are mainly from the news and lifestyle category which belong to the most viewed categories by the user. Now based on the user's likes the parameters are updated. In Figure 9 the final recommended articles are shown. Here we can see that apart from articles from the top 5 categories, articles from other categories are also being recommended, indicating the fact that indeed our Thompson sampling model inhibits exploration along with exploitation.

In order to check whether, given some oracle probabilities does the Thompson Sampling model converge towards them over a number of iterations, a simulation of Thompson Sampling was also carried out.

- The beta distribution parameters of the user was initialised to (1,1)
- The oracle probabilities were generated using the users past click history according to the categories viewed by the user being allotted higher probabilities.
- Using these probabilities the simulation of thompson sampling was performed on about 2000 iterations and it was seen that the model successfully converged to the near almost the given oracle probabilities.

Figure 10 depicts the simulation for a particular user. where the oracle probabilities were as, [(0.65), (0.09), (0.05), (0.02, …….(0.01), (0,0)]. We can see that the model successfully converges.

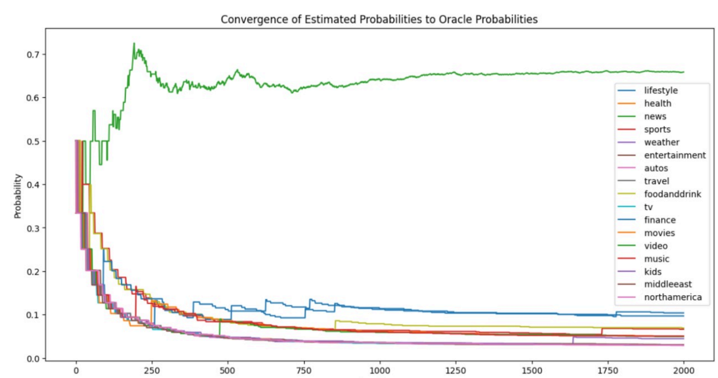

Figure 10: Thompson Sampling Simulation

Additionally, the Thompson Sampling model was also run on the fetched news articles from NewsApi.com.

- A user is initially shown the 7 categories (Figure-11) from which NewsApi.com provides news from and is asked to select two.
- Now for the categories selected by the user the beta distribution parameter of alpha is increased by a factor of 3 in order to increase the chances of them being sampled.

- Then with the updated parameters news categories are sampled resulting in 3 articles each from the selected categories being recommended to the user (Figure-12).
- The user is given the option of liking or disliking the recommended news. Based on the user's choice, when an article is liked the alpha parameter is updated and when disliked the beta parameter of that category is updated. Thereby capturing the user's preferences and balancing exploration and exploitation (Figure - 13).

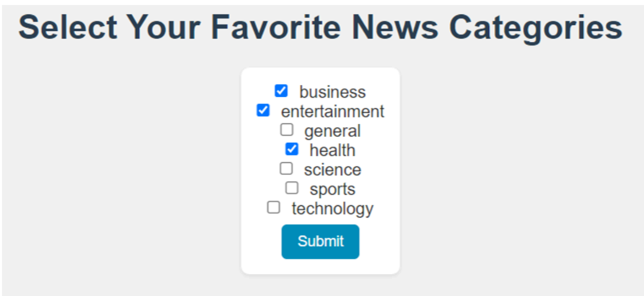

Figure 11: Selection of Categories by the user

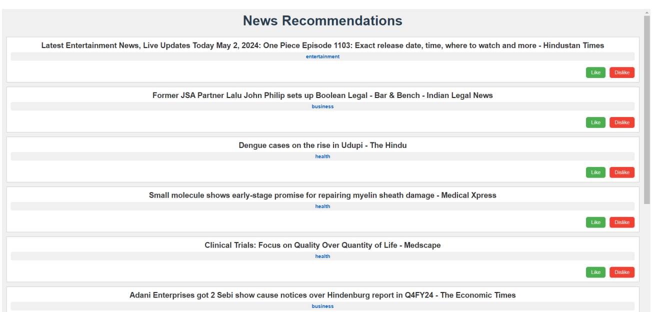

Figure 12: Recommendations based on Selection

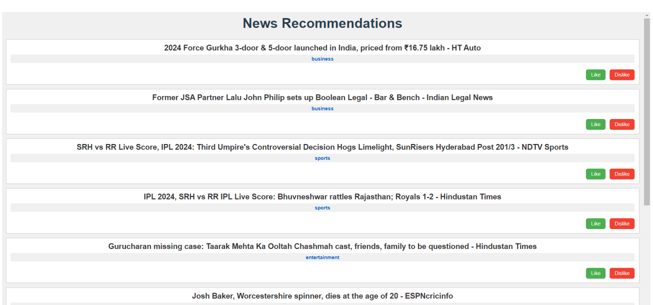

Figure 13: Final recommendations based on like/dislike

In Figure 11, the user selects business, health and entertainment as its preferred categories. Now based on that further recommendations are made as seen in Figure 12. The user now having disliked all health articles asks for further recommendations. Figure 13 clearly does not recommend articles from health but from other categories.

Thompson Sampling successfully modelled the user's preferences while also having the factor of exploitation which enables media organisations to push trending and popular content to its users.

E. Hybrid Model

Thompson sampling does balance exploration and exploitation but over time it can also start making recommendations focusing only on exploitation. The model discussed above made improved recommendations by increasing the exploration as compared to K-Means based model, but the level of exploration was not up to the mark. In news recommendation systems in order to promote popular, trending and latest news articles a significant amount of time exploration is required. The UCB algorithm encourages exploration a lot. Hence an attempt was made to combine the properties of both of these models in order to increase the exploration factor and the hybrid model was created.

- In this model similar to Thompson sampling the user's beta distribution parameters are initialised using the users click history. For any category the user has seen the alpha parameter value for those arms are increased.
- Now the sampling is performed to get a preference probability for each of the categories.
- In this probability term another UCB like exploration term called 'adjusted preferences' is added to create a final score for every category as follows,

$$S_i = p(i) + \sqrt{\frac{2.log(R_T+1)}{R_T(i)+1}}$$

here S_i, is the adjusted preferences, $p(i)$ is the sampled preference probability from Thompson sampling. R_T represents the total number of recommendations made and $R_T(i)$ represents the number of recommendations for category i.
- From these adjusted scores, the top categories are considered for recommending news articles.

This hybrid model helps in encouraging more exploration in the recommendations. The adjusted preferences term is responsible for doing that. Another advantage of adding this term is that for categories that may have been disliked by the user, they will not be displayed for a while but after several iterations when the exploration term starts dominating again in those categories they will be recommended again.

VI. PERFORMANCE EVALUATION

Evaluating the performance of the models discussed is important to determine which model should be preferred and why. To measure the performance of the models implemented, two metrics, namely: Match Rate and Coverage have been used.

Match Rate measures the proportion of recommended items that are relevant (belonging to the top 5 categories of the user) or match the user's interests based on their past behaviour or preferences to the total number of recommendations. It is a direct measure of relevance. Greater the value of match rate, greater is the amount of exploitation performed by the model

$$Match\ Rate = \frac{Number\ of\ recommendations\ from\ top\ 5\ categories}{Total\ number\ of\ recommendations}$$

Coverage measures the ratio of the number of top categories of that user are covered in the recommendations to the number of top categories. If more number of top categories are being recommended then it tells that exploitation is being preferred, whereas lower coverage value indicates that non top categories are also being recommended to the user, signifying exploration.

$$Coverage = \frac{Number\ of\ top\ categories\ in\ recommendations}{Total\ number\ of\ top\ categories}$$

The models were evaluated by calculating the average Match Rate and average Coverage values over 20 iterations for each model. The results of the various models on the above two metrics can be seen in Table 1 and are discussed further.

Table I - Metric Results on recommendation models

Metric Name	**K-Means Clustering**	**UCB**	**Thompson Sampling**	**Hybrid Model**
Match Rate	0.73	0.19	0.64	0.59
Coverage	0.43	0.19	0.36	0.32

From the results it is clearly evident that the UCB model is significantly encouraging exploration and focuses very less on exploitation, hence it does not incorporate the user's preferences. This is not good considering that user needs are supposed to be catered to. It is important to factor in the user's preferences.

The K-means clustering and Thompson sampling models are comparable as they display significant exploitation along with some exploration as well. The K-Means algorithm, although designed for exploitation, is able to inhibit exploration qualities due to the fact that we created a pool of articles for each cluster allowing us to recommend more diverse and popular articles.

Thompson Sampling due to its probabilistic nature encourages a lot more exploration. However, as we initialised the beta distribution parameters based on the user's click history, it allows Thompson sampling to exploit the user's preferences as well. Still our model is more biased towards exploitation only.

The hybrid model that has been attempted shows a significant drop in match rate and coverage signalling towards more exploration taking place. This model is the closest to the 50-50 balance between exploration and exploitation, however Thompson Sampling also is a great option as it also performs similarly to it.

VII. CONCLUSION & FUTURE SCOPE

The recommendation problem has a diverse variety of solutions possible, each with its own benefits and pitfalls. The correct solution greatly depends on the specific scenario and the organisation's needs and requirements. It also highly depends on the type and amount of data available or not.

For the News article recommendation system, it is evident that user preferences, signifying what genre and what kind of news they like is important but in order to push the latest, trending and popular news articles it is equally important to promote exploration. Hence, Thompson Sampling and the Hybrid model implemented in this project would be the suitable choices in this scenario.

In future attempts can be made to improve the K- Means Clustering model by implementing it in tandem with Thompson Sampling/Hybrid model. In this model once the user clusters are created, in order to sample articles from those clusters we can deploy Thompson Sampling/Hybrid model to increase exploration in that model and make it even more suitable for our problem at hand.

REFERENCES

[1] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu and Ming Zhou. MIND: A Large-scale Dataset for News Recommendation. ACL 2020.

[2] A Tutorial on Thompson Sampling Daniel J. Russo , Benjamin Van Roy , Abbas Kazerouni , Ian Osband and Zheng Wen