# Machine Learning (CSE 343/ECE 363) Final-Project Report: Machine Learning-Driven News Recommendation

Baljyot Singh Modi
baljyot22133@iiitd.ac.in

Debjit Banerji
debjit22146@iiitd.ac.in

Jaleel Ahmed Radhu Khwaja
jaleel2022225@iiitd.ac.in

Vijval Ekbote
vijval22569@iiitd.ac.in

## 1. Abstract

The project aims to develop a News Recommendation System to aid users by suggesting relevant articles according to their interests. In this project, we first researched some of the existing work and research papers on this topic. We then identified relevant datasets and performed preprocessing as required. We extracted the necessary features from the datasets and utilized these features for our two approaches. Our first approach is based on content-based recommendation where we have developed Clustering as well as Multi-Class Classification Models. Our second approach is based on combining content-based and collaborative-based recommendation systems. We have then presented our model performances after hyper-parameter tuning, and have given our analysis of the project results. Finally, we have concluded by stating the purpose of this project and how we were able to achieve it.
GitHub Repository Link

## 2. Introduction

In today's data driven world, there is a plethora of information available online in various forms, which, while beneficial, also poses a dilemma for the user, who would like to prioritise what information they consume in order to save time and maximise the utility they obtain. Thus, with the vast amount of news available online, users often struggle to find content that matches their interests. Our goal is to develop a machine learning-based news recommender system that continuously learns and adapts to changing user preferences and news feeds. This will ensure that the recommended content remains relevant and personalized, thereby enhancing the user experience and improving engagement.

## 3. Literature Review

1. **A Proposal for News Recommendation Based on Clustering Techniques**
   Link: https://link.springer.com/chapter/10.1007/978-3-642-13033-5_49
   This paper explores clustering algorithms for the task of news recommendation. In particular, K-Means and Aspect Model are used for clustering, with Aspect Model seen to perform better. The paper used the Ahora dataset with 2, 3, 5, and 10 classes, and the metrics they used to compare their predicted category-wise likelihoods for a particular user were Spearman's Rank Correlation Coefficient and Kullback Leibler Divergence.

2. **Content-Based Collaborative Filtering for News Topic Recommendation**
   Link: https://cdn.aaai.org/ojs/9183/9183-13-12711-1-2-20201228.pdf
   This paper explores the benefits of combining both content-based and collaborative filtering approaches, thus leveraging the advantages of both these techniques. The main tool in collaborative filtering approaches is probabilistic matrix factorization, used to derive latent factors for the user and the items. On the other hand, content-based approaches make use of a weighted sum of neighbours to calculate predictions. This papers adds this neighbourhood model to its latent factor model, and uses Stochastic Gradient Descent for training. Calculating the neighbourhood of an item i involves making use of a similarity measure, in this case, a Fischer Kernel.

## 4. Dataset

### 4.1. Discovering Relevant Datasets

After extensive research, we have found two useful datasets for our project.

(i) **The News Article Dataset:**
This dataset contains around 210k news headlines from 2012 to 2022 from HuffPost. This is one of the

biggest news datasets and can serve as a benchmark for a variety of computational linguistic tasks. Due to changes in the website, there are about 200k headlines between 2012 and May 2018 and 10k headlines between May 2018 and 2022.

The dataset can be accessed using the following link:
News Article Dataset

(ii) **The MIND Dataset:**
The MIND dataset for news recommendation was collected from anonymized behavior logs of the Microsoft News website. The data randomly sampled 1 million users who had at least 5 news clicks during 6 weeks from October 12 to November 22, 2019. To protect user privacy, each user is de-linked from the production system when securely hashed into an anonymized ID. Also collected were the news click behaviors of these users in this period, which are formatted into impression logs. The dataset can be accessed using the following link:
MIND Dataset

### 4.2. Text Pre-processing

We used standard text-cleaning techniques, such as removing stopwords, numbers, punctuations, HTML tags, expanding contractions, lemmatization, etc. to preprocess the text.

### 4.3. Feature Extraction

1. **TF-IDF:** Term Frequency-Inverse Document Frequency (TF-IDF) is a numerical statistic that reflects how important a word is to a document relative to a collection of documents. It is used to transform the textual data into numerical form based on word frequencies, emphasizing more unique terms for each article. The formula for the TF-IDF Feature Extraction Technique is as follows:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t)$$

where,

$$\text{TF}(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

$$\text{IDF}(t) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

Here, $t$ means the word/term, $d$ means the document or the article, $f_{t,d}$ means the frequency of term t in article d, $N$ indicates the total number of articles in the corpus and $|\{d \in D : t \in d\}|$ indicates the number of articles that contain the term $t$.

2. **LDA:** Latent Dirichlet Allocation (LDA) is a probabilistic topic modeling technique that identifies hidden topics in a set of documents by assigning words to topics and documents to those topics. It helps in understanding the thematic structure of the articles. The formula for the LDA Feature Extraction Technique is as follows:

$$P(z|d, w) \propto P(w|z) \cdot P(z|d)$$

where,

$P(w|z)$ is the probability of word $w$ given topic $z$,

$P(z|d)$ is the probability of topic $z$ given document $d$

## 5. Methodology

**Approach 1:**
To obtain the feature matrix, we first extracted relevant features from the text corpus using `TF-IDF` and `LDA` vectors. After extracting the `TF-IDF` features, we applied SVD to tackle the sparse matrix and an AutoEncoder to reduce the dimensionality of the feature matrix whilst retaining the most important linear and non-linear `TF-IDF` features.

#### Clustering

Our first approach is a clustering-based approach. We first create clusters based on the feature matrix extracted as described above.

We will use K-means clustering for the clustering-based approach.

For K-Means clustering, we try to minimize the following objective function, which measures the distance of each point from the centroid of its cluster.

$$J = \sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - \mu_i\|^2$$

Now, after training the model, we calculate features for these new articles, by adding the features obtained from the autoencoder and SVD to form vectors, and then find the nearest cluster centroids to these articles using cosine similarity. The articles belonging to these clusters and closest to the given article will be our final recommendation for the user. For recommending similar articles to the user, we have used the following similarity measure:

$$\text{Cosine Similarity}(P, Q) = \frac{P \cdot Q}{\|P\|\|Q\|} = \frac{\sum_{i=1}^{n} p_i q_i}{\sqrt{\sum_{i=1}^{n} p_i^2} \sqrt{\sum_{i=1}^{n} q_i^2}}$$

**Multi-Class Clssification**

We also aimed to accomplish the category prediction for each article from the obtained features. For this task, we trained various supervised learning algorithms such as Random Forest, Support Vector Machine, XGBoost, AdaBoost, LightGBM, etc.

**Approach 2 (Based on paper 2):**
We proceeded more or less along the lines of paper 2. We also try the item-based approach but with a different similarity measure, such as cosine similarity between tf-idf vectors. We also convert this into a binary classification task, where we try to predict whether a particular user is likely to click on a particular item or not.

We first create a user-item interaction matrix, then perform SVD to obtain $\mathbf{U}$, $\Sigma$, and $\mathbf{V}^\top$. The user and item latent vector matrices are generated by multiplying $\mathbf{U}$ and $\mathbf{V}^\top$ with $\Sigma$. We pre-compute the neighborhood sum for each item and use this, along with the user and item latent vectors, for prediction.

The equation for what we predict in this approach is:

$$\hat{r}_{ui} = \sigma\left(\mathbf{p}_u\left(\mathbf{q}_i + |N(\theta,i)|^{-1}\sum_{j\in N(\theta,i)}\theta_{ji}\mathbf{y}_j\right)^\top\right)$$

Also, given a user-item matrix $M$ of shape $m$ x $n$, in probabilistic matrix factorization we estimate the following:

$$\hat{M} = p^\top q$$

Where $\mathbf{p}$ and $\mathbf{q}$ are latent vectors

We make use of the binary cross entropy loss rather than a regularalized squared error loss (as used in the paper), since our task is a bianry classification task.

All in all, some small differences (compared to paper 2) we have introduced from our side in this method are:

1. Turning it into a binary classification task, and thus using a different loss function
2. Using different similarity measures for calculating neighbourhoods.

# 6. Results and Analysis

## 6.1. Feature Extraction Outcomes

- **TF-IDF Features**: For each article, TF-IDF scores were computed to represent the frequency and importance of terms. This method helps down-weight common words that appear across all articles and emphasizes terms that are more unique to individual articles or smaller groups of articles.
- **LDA Topic Distributions**: Using Latent Dirichlet Allocation (LDA), we identified key topics across the news articles. Each article was represented as a mixture of these topics, with distributions indicating the prominence of each topic within the article. For example, prominent topics identified include politics, technology, and sports, among others.

Figure 1: Correlation Heatmap between categories and topics obtained from LDA

LDA gave us a good picture of the various topics, enabling us to correlate these with our categories. For instance, for the category 'Sports', we can see that topic 0 is highly expressed or maximally correlated. Looking at the representative words for this topic (obtained using LDA), we can see various words related to sports, such as Olympics, athlete, player, team, etc. Similarly, the same thing can be observed for the category 'Business' as the representative words for this topic are company, work, money, industry, etc.

## 6.2. Clustering Results

We applied the K Means algorithms and arrived at the following results. In figure 2, we visualize the clusters using t-SNE, and in figure 3, we use PCA for visualization. We can clearly see the different clusters obtained by applying the algorithms to the extracted features.

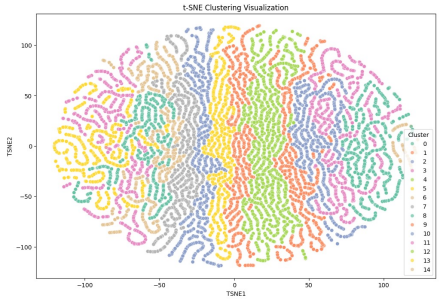


Figure 2: t-SNE plot, topic wise, after K-Means Clustering (2D)

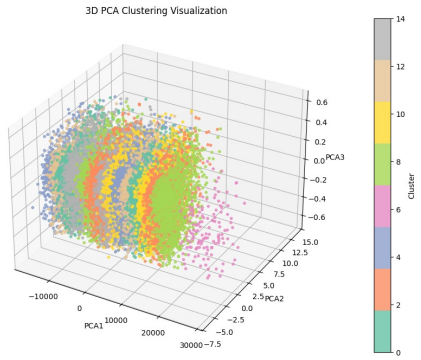


Figure 3: PCA plot, topic wise, after K-Means Clustering (2D)

### 6.3. Category Classification Results

After applying various models on the data, we obtained the following results:

| Model | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| SVC | 0.732798 | 0.733100 | 0.724081 | 0.724203 |
| Logistic Regression | **0.753460** | **0.750616** | **0.747449** | **0.747967** |
| Decision Tree | 0.536610 | 0.532343 | 0.529309 | 0.529549 |
| Random Forest | 0.694483 | 0.692247 | 0.685535 | 0.685590 |
| XGBoost | 0.735005 | 0.732072 | 0.727474 | 0.727751 |
| KNN | 0.541023 | 0.552661 | 0.534071 | 0.536060 |
| ANN | 0.728385 | 0.743227 | 0.719741 | 0.718100 |
| Naive Bayes | 0.273821 | 0.357697 | 0.266731 | 0.237802 |
| Extra Trees | 0.677834 | 0.674261 | 0.668164 | 0.667406 |
| LightGBM | 0.731595 | 0.729765 | 0.724272 | 0.725245 |
| CatBoost | 0.735005 | 0.732319 | 0.726651 | 0.726840 |
| AdaBoost | 0.515547 | 0.499606 | 0.501248 | 0.489733 |

Table 1: Performance of Machine Learning Models

The metrics of the table clearly indicates that the Logistic Regression Model performs the best among all the models, followed by CatBoost and XGBoost models.

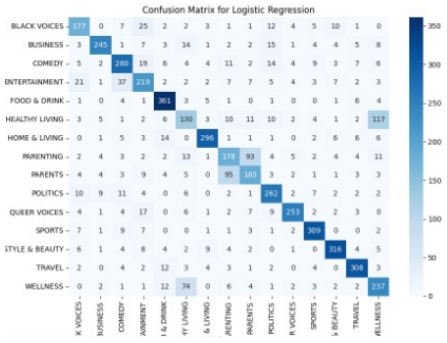


Figure 4: Confusion matrix for Logistic Regression

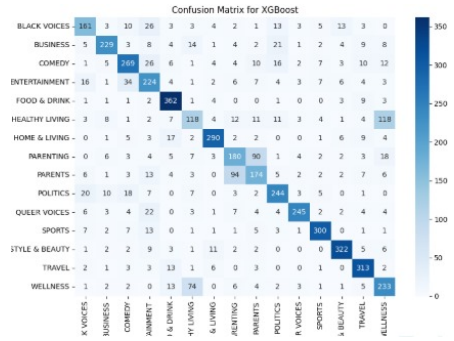


Figure 5: Confusion matrix for XGBoost

Some observations we made: XGBoost shows confusion between the categories 'Parenting' and 'Parents' since these are very similar. Also, there is also some confusion between 'Wellness' and 'Healthy Living'.

### 6.4. Results from Appproach 2:

We obtain the following results:

| Accuracy | F1 Score | Precision | Recall |
|---|---|---|---|
| 0.64 | 0.75 | 0.92 | 0.64 |

The confusion matrix is as follows:

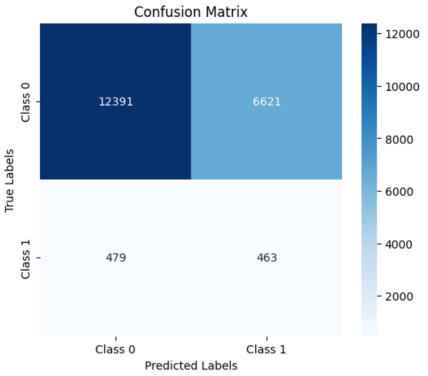


Figure 6: Confusion Matrix for Approach 2

## 7. Conclusion

This project involved using content based, and a mixture of content-based and collaborative filtering for recommendation. XGBoost and Logistic regression performed well for our tasks. For approach 2, one of the limitations in our work is that we did not have access to a dataset that provided both interaction data and detailed article texts. In future works, we can also apply some more advanced NLP techniques for feature extraction, which take into account the relationship among words in a better way, which may be the reason we were unable to get homogenous clusters.

## 8. References


(1) Misra, Rishabh. "News Category Dataset." arXiv preprint arXiv:2209.11429 (2022).

(2) Misra, Rishabh and Jigyasa Grover. "Sculpting Data for ML: The first act of Machine Learning." ISBN 9798585463570 (2021).

(3) Cleger-Tamayo, S., Fernández-Luna, J.M., Huete, J.F., Pérez-Vázquez, R., Rodríguez Cano, J.C. (2010). A Proposal for News Recommendation Based on Clustering Techniques. In García-Pedrajas, N., Herrera, F., Fyfe, C., Benítez, J.M., Ali, M. (Eds.), *Trends in Applied Intelligent Systems. IEA/AIE 2010*. Lecture Notes in Computer Science (Vol. 6098). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-13033-5_49

(4) Lu, Z., Dou, Z., Lian, J., Xie, X., Yang, Q. (2015). Content-Based Collaborative Filtering for News Topic Recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence, 29*(1). https://doi.org/10.1609/aaai.v29i1.9183

(5) Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '08). Association for Computing Machinery, New York, NY, USA, 426–434. https://doi.org/10.1145/1401890.1401944

(6) https://matteucci.faculty.polimi.it/Clustering/tutorial_html/cmeans.html

(7) https://en.wikipedia.org/wiki/K-means_clustering#:~:text=The%20objective%20function%20in%20k,necessarily%20to%20the%20global%20optimum.

(8) Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. 2020. MIND: A Large-scale Dataset for News Recommendation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 3597–3606, Online. Association for Computational Linguistics.