Microsoft ie SCHOOL OF HUMAN SCIENCES & TECHNOLOGY

News Recommendation System

Boosting User Engagement, Driving Revenue!

Group 20

SUMIAH ALDUHAIM, LAKSHMI PRIYANKA JAKKA, JOHNNY NAIME,M WALID TARRAB,HANG CHI KU(ADRIAN)

Ethics Statement
I hereby certify that this report and the accompanying presentation is my own original work in its entirety, unless where indicated and referenced.

Signed by:

JOHNNY NAIME, SUMIAH ALDUHAIM, LAKSHMI PRIYANKA JAKKA, HANG CHI KU, M WALID TARRAB

Priyak J

Contents:

Executive summary

The News Recommendation system uses the MIND News dataset to create a powerful recommender system. This dataset contains behavioural information from roughly 1 million users over a six-week period and 160,000 news items. The objective is to anticipate user preferences and suggest articles that are more likely to be clicked, increasing user engagement and satisfaction and resulting in greater traffic and more money from advertising.

Potential dataset limitations include uneven distribution of impressions by class, inconsistent title lengths, and a lack of user responses. The authors offer particular remedies to these problems, such as limiting titles to those longer than four words, producing synthetic batches of impressions that downsample the negative impressions, and suggesting that future implementations include additional data on user behaviour.

Different approaches are also discussed. User profiling is one such strategy that entails making thorough user profiles based on interests, preferences, and consumption patterns. This strategy may be useful for user segmentation, personalised content distribution, and targeted marketing. Transformation on the dataset are also discussed, such as data extraction, splitting impressions, assessing imbalances, and extracting news IDs.

The value proposed is discussed in terms of the potential ROI that this new recommender system proposes, along with an explained timeline of implementing this system.

For Data engineering and preparation, we used robust technologies, such as Python, specifically Jupyter notebooks, and Microsoft Azure Blob Storage. This setup helps facilitate the structure, cleanliness, and transfer of data for this recommender system. Transformations undergone are those of data extraction, splitting impressions, imbalance assessment, and news ID extraction.

As for the data science section, we mention the EDA undergone, the model built and comparison to other preexisting models, model performance and the relation to the ROI. The recommendation system can give better recommendations to the user in the given dataset. However, the model is just a proof of concept and has some limitations due to limited regional diversity, lack of historical data, and constraint on computational power. To further improve the performance, extensive experiments on real-world dataset would be recommended.

As for the final part production level considerations, we have included a technical architecture, CI/CD, legal and IP aspects related to the dataset, and alternatives to the dataset with a methodology for building an in-house dataset.

1. Initial data exploration and hypotheses

The MIND News dataset comprises an extensive collection of behavioral data from approximately 1 million users, coupled with 160,000 news articles spanning a six-week timeframe. This project aims to utilize this rich dataset to develop a sophisticated recommender system capable of predicting user preferences and recommending articles with a higher probability of being clicked. By leveraging this system, our objective is to enhance user engagement and satisfaction, leading to increased traffic and thus higher advertising revenue. This paper outlines the key components and methodologies involved in constructing and implementing a recommender system trained based on the MIND dataset, highlighting its potential to yield a high return on investment (ROI) by improving user satisfaction[1].

1.1 Potential limitations of the provided dataset

The dataset has some potential caveats that might limit the scope of our recommender system. For example, class imbalance in the impressions, inconsistency in the titles' length, and lack of user responses. The positive impression rate generated by users makes up 4 percent of the total impressions. This imbalance if not addressed can raise many issues such as bias and poor generalizability. Another limitation is that some of the titles in the dataset might be too long/short to extract enough information to deduce the content of the article. The dataset also lacks additional useful information like user preferences to better predict the user's likings.

- for the title, we decided to only use the titles that are longer than 4 words
- for the class imbalance, we would synthetically create batches of impressions that downsample the negative impressions. (16.5% positive)
- lack of user responses data, for future implementations it is recommended to include more information pertaining to user behavior.

1.2 Approaches

The MIND News dataset offers a range of potential avenues for extracting and modeling meaningful insights. One such approach is user profiling, whereby the dataset's behavioral data can be utilized to create detailed user profiles based on interests, preferences, and consumption patterns. This profiling approach holds significant value for targeted marketing, personalized content delivery, and user segmentation, enabling more effective engagement strategies[2]. There are several features that can be considered for user profiling, these features help capture user preferences, behaviors, and interests related to news consumption shown in figure 1.2.1.

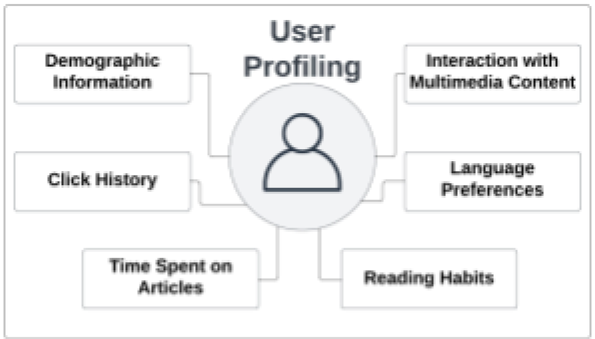


Figure 1.2.1: User Profiling

Another valuable approach involves news clustering and topic modeling, wherein the dataset can be analyzed to identify common themes and cluster news articles accordingly. This process facilitates better organization and categorization of news content, leading to improved content discovery for users as shown in figure 1.2.2..

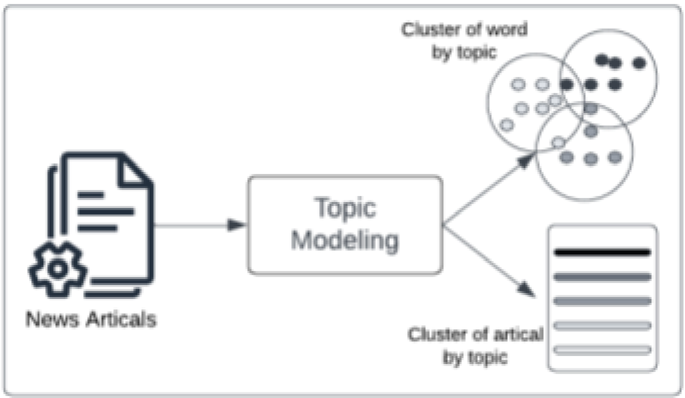


Figure 1.2.2: Topic Modeling

Moreover, Identify trending topics or news events by analyzing temporal patterns in user interactions and article publication dates. Determine the lifespan of different news topics and understand how user interest evolves over time. Additionally, leveraging the dataset for news sentiment analysis allows for the classification of news articles into positive, negative, or neutral sentiment categories. Such analysis provides valuable insights into trending sentiments and public opinion surrounding news topics.

The dataset can be employed for news event detection, enabling real-time monitoring and trend analysis. By tracking time-related data and analyzing news content, significant events and emerging trends can be identified, empowering proactive decision-making. Lastly, the most impactful approach aligning with business objectives is the development of a news recommender system. This approach utilizes user behavior and data to generate and enhance personalized news recommendations tailored to individual user profiles and preferences. By leveraging the extensive dataset, this recommender system has the potential to drive increased user engagement, satisfaction, and ultimately support business growth objectives as shown in figure 3.

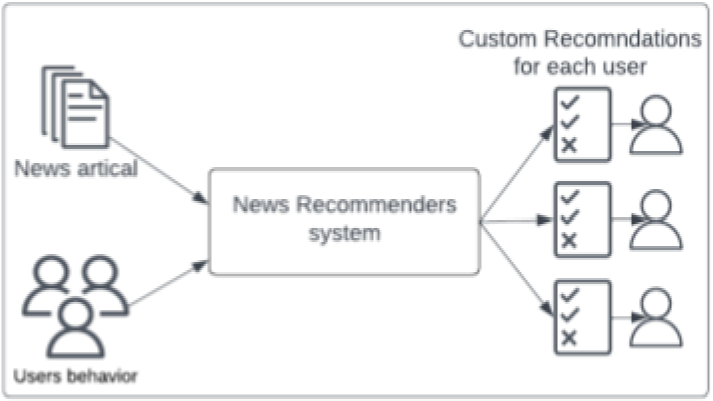


Figure 2.1.3: News recommender system

1.3 Value proposed

News agencies employ a diverse range of revenue models to sustain their operations and deliver valuable journalistic content to their audience. These revenue streams are but not limited to the following points[3]:

- Advertising: News agencies sell ad space on their websites, mobile apps, or print publications. They may offer different ad formats, such as display ads, sponsored content, or video ads. The revenue generated depends on factors like the agency's audience size, engagement, and ad rates.
- Subscription/Membership: Some news agencies offer premium content or exclusive features to subscribers or members. This model involves charging a recurring fee for access to in-depth articles, specialized newsletters, or ad-free browsing. The revenue comes from subscription fees and the retention of loyal subscribers.

The project focuses on generating higher user engagement and traffic (subscribers/conversion rate). First, the news recommender system will work in analyzing the patterns and preferences of the user. Based on the behavioral data, users will encounter content that is appealing to them, matching with their preferences. This system will encourage users to be more engaged with the website, increasing user satisfaction. Higher user satisfaction will generate more traffic, increased engagement, improved reputation, and enhanced loyalty. satisfied users are more likely to engage with news content, translating to more page view, session time, and lower bounce rates. They are also more likely to become loyal subscribers of the agency as they bookmark the website, subscribe to newsletters, or follow the social media account. Together, they collectively contribute to the growth and success of news agencies. Some of the biggest benefits of having a recommendation system include higher web traffic, better news category targeting, personalized news pages and boards, and thus generate overall higher revenue.

To begin with, higher web traffic is essential for increasing viewership of news on the agency's website and a bigger reputation in the industry of news reporting. Higher viewership will increase the ad revenue and potential new subscribers. The current revenue model is generating 5 euros per 1000 impressions. The Assumption has been made that after implementing the recommender system, web traffic will increase by 10% which translates to at least 10% increase in ad revenue.

Current business structure

Total number of users	50,000
Conversion Rate without the recommendation system	2%
Average revenue per conversion (subscription fee)	$75
Website traffic per year	13M
CPM (Cost per mille/cost per thousand impression)	$5

Improved Business Structure

Cost of implementation (news recommender system)	$20,000
% of new conversion rate	2.4%
Expected website traffic per year	14.3M

ROI Calculation:

Step 1- Calculate the net profit:

Additional Revenue = Increased conversions + Increased ad revenue

Increased conversions: Assuming a 20% improvement in conversion rate with the recommendation system

Increased conversions per year = number of users, 50,000 * change in conversion rate, 0.004 = 200

Increased revenue per year = 200 additional conversions * $75 per conversion = $15,000

Increased ad revenue: Assuming a 10% increase in traffic with the recommendation system

yearly ad revenue = (cost per 1000 impressions) * traffic per year

(5 /1000) * 13M = $65,000 yearly ad revenue – before recommendation system

(5 /1000) * 14.3M = $71,500 yearly ad revenue – after recommendation system

Increased ad revenue = $6,500

Increase in revenue after implementing recommendation system: $15,000 + $6,500= $21,500

Step 2: ROI = ((Revenue - Cost of Implementation – Maintenance) / (Cost of Implementation + Maintenance) * 100

First year profit= $21,500 - $20,000= 1500$

second year profit = $21,500

Cost of Implementation = $20,000

Maintenance of system = $2,000

ROI = (($43,000 - $20,000 - $2,000) / $20,000 + $2,000) * 100

ROI = 95.45%

The ROI of implementing the news recommendation system is 95.45%. This indicates that for every dollar spent on the implementation, the company would receive a return of $0.9545. However, please note that these numbers are purely hypothetical and should not be considered as precise projections. Actual ROI calculations require accurate and detailed financial data from your specific news organization.

1.4 Timeline

The timeline for this process would go for 2 years to realise the ROI for the recommended changes. During the implementation phase, it is expected that the company will start by allocating the resources and take the necessary steps to execute the strategies.

After the implementation phase, the company should start seeing the results of the new strategy. Here the company can see how well the strategy is doing and optimizing it to generate the best outcome. Monitoring this process is crucial to properly evaluate how the model is doing and if it has the potential to generate more revenue for the company

Over the course of the subsequent months, the company can expect to see improvements in areas such as customer satisfaction, operational efficiency, or an improved market position. The overall effect of the ROI should be felt by the 2nd year, since after the cost of implementation only the maintenance cost will stand

2. Data engineering and preparation

2.1 Internal data tools

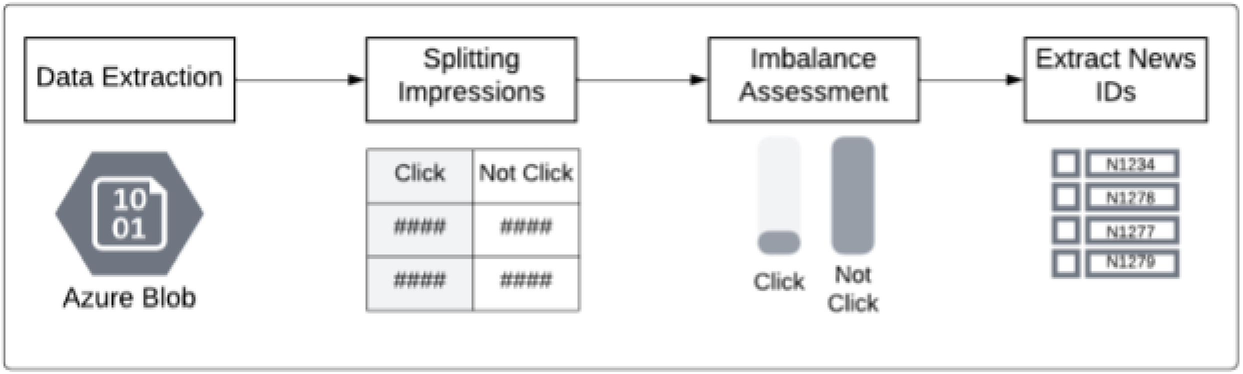
Recommender system plays an important role in analysing user behavior and historical data to deliver personalized news recommendations to millions of users. Behind the scenes, a well-designed technology stack enables efficient data processing. Data engineering and preparation help to ensure the data is organized, cleaned and transferred for downstream data driven applications. It is considered a foundation to build a robust and scalable recommendation system. Before choosing technology stack tools, it is important to consider some factors based on recommender system needs. such as data volume and complexity, performance and latency, integration with existing systems and skills and expertise of the existing team.

In this context, open source python utilities and Jupyter notebooks are common tools and best practices to develop a powerful recommender system. dataset utility, reads the data stored in Microsoft Azure Blob Storage, it’s designed to manage large amounts of data. it offers a reliable, scalable, and highly available storage solution. Also, it can be integrated with Azure services as well as reading different sizes of data for testing purposes (Demo, Small, Large).

Using Jupyter notebooks to utilize the powerful pandas library for data manipulation and analysis which is a high performance tool including DataFrame. Moreover, leveraging GPU capabilities for accelerated computations and speed up the data engineering and preparation processes. By harnessing the power of this technology stack, you can streamline the process of data engineering and preparation, paving the way for the development of robust recommender systems and other data-driven applications.

2.2 Main data transformations

Dataset stored in Microsoft Azure Blob Storage splitted into training and validation sets, each with a large and small version, the format of the files are the same. the dataset goes into pipeline processes to do the main transformations as following[4]:



Click	Not Click
####	####
####	####

Figure 2.2: Main data transformation

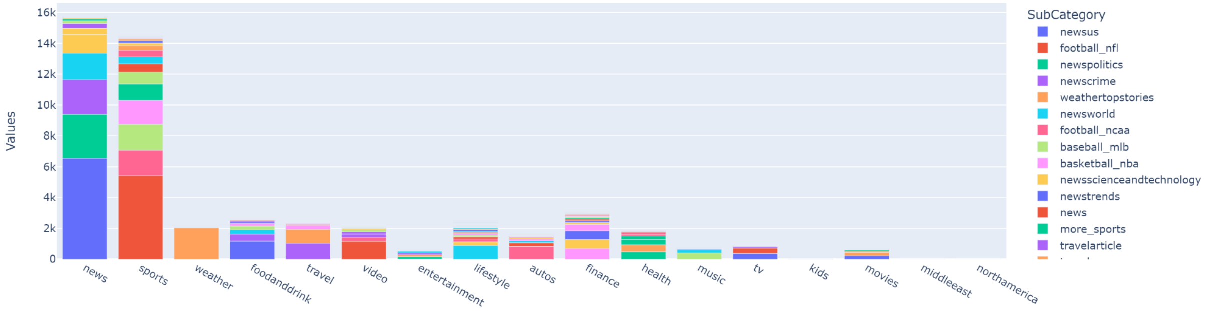
- **Data extraction:** retrieve behaviors and news data files to serve as raw for further analysis. The behaviors file contains the impression logs and users' news click histories, while the news file contains information about the articles.
- **Splitting impressions:** begin with analyzing behaviors file, the impressions logs have been splitted into two columns, click and not click. Click indicates the user interacted with this news article by clicking on it, while not click indicates the user didn't interact with the news article. This information can be valuable in optimizing content recommendations or designing effective strategies to increase user engagement.
- **Imbalance assessment:** After splitting the impressions, there is an imbalance between click and not click category. Having a highly unbalanced dataset can skew the results and lead to biased conclusions. That is why it is necessary to address this issue, it will be handled under the data science section to have a representative dataset.
- **Extract News IDs:** The history in the behaviors file can be processed by splitting it into a list of news IDs. News ID is a unique identifier for each news article. Doing this serves as a foundation for different types of analysis for example the average number of articles clicked by user or most frequently clicked articles.

The aforementioned transformations empower the dataset, enabling the extraction of valuable insights and the development of effective solutions. By applying these transformations, the dataset is prepared to uncover meaningful patterns and trends, leading to informed decision-making and the implementation of impactful strategies.

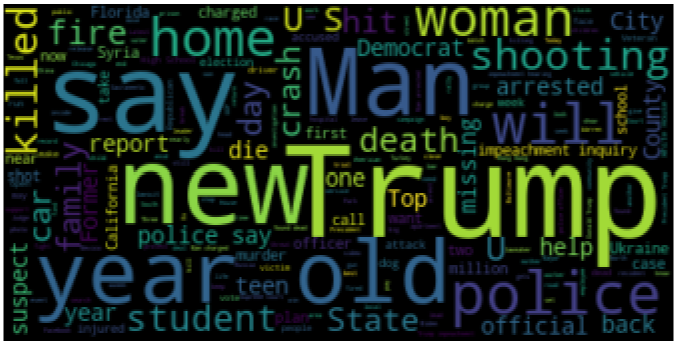
3. Data science

3.1 Exploratory data analysis (EDA)

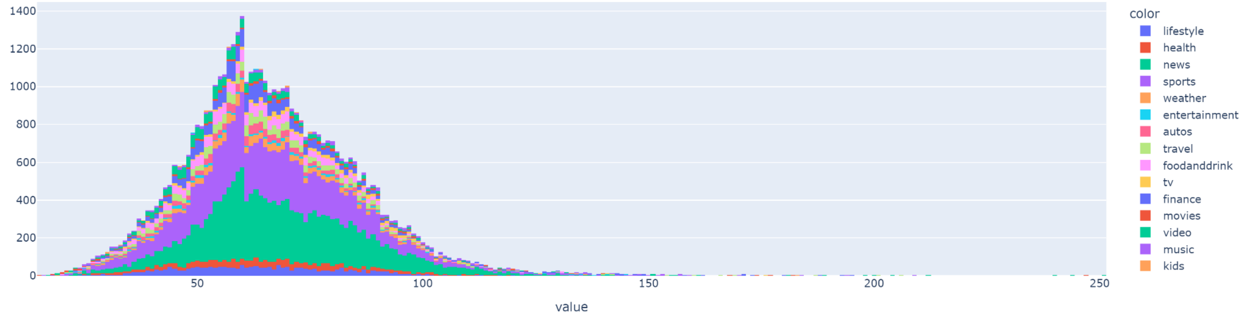
After the data was imported, we looked at the distribution of the categories and subcategories of the news. Most of the news originated in the USA, which means that the model should be more representative of the US market.



Taking a further dive into the news in the US news, we looked at the most frequent words in the abstract. The word "Trump" appeared the most, which aligns with the time period 2019 when Trump was still in office. Some of the notable events that occurred during this period that were widely reported.



On October 12, 2019, Trump announced that he would not be hosting the G7 summit at his golf resort in Doral, Florida, following criticism from both Democrats and Republicans. On October 27, 2019, Trump announced that the leader of ISIS, Abu Bakr al-Baghdadi, had been killed during a U.S. military operation in Syria. On November 13, 2019, Trump's former advisor Roger Stone was found guilty on seven counts, including lying to Congress and witness tampering. On November 20, 2019, Trump's longtime friend and advisor Roger Stone was sentenced to 40 months in prison for obstruction of justice, witness tampering, and lying to Congress.



Most news titles' lengths are at the 60 words mark, with most of them falling between 50-100. Some of the news have very short titles making the process of extracting any useful information out of them impractical. These articles will be removed in the preprocessing step.

3.2 Base model

Building a recommender system from scratch is not feasible due to time and technical constraints. It is a laborious and costly process, requiring extensive data to train a complex model. Furthermore, evaluating a recommendation system in production can be challenging because obtaining the ground truth data can be difficult at a large scale. If not put into production, it is impossible to measure the system's performance over time to see if it continues to provide relevant and accurate recommendations. Therefore, we will leverage pre-existing models provided by Microsoft to develop the recommender system as a proof of concept.

However, considering that most of these models including the dataset are open source only for research purposes, for the baseline model, we will create our own collaborative model using the DistillBERT model to tokenize the news' titles that analyzes the relationship between the users and the news titles using that detect similar users and make predictions based on these estimated proximities. This will be a poc that demonstrate the general process and outcome of building such a recommendation system. The system takes in a user's click history and generates recommendations based on the user's interests. After generating the embeddings of the new's titles, we also apply PCA to reduce the dimensionality of the embeddings and calculate cosine similarity to find the most similar articles.

The preprocessing step involves dropping articles with nas, duplicates, and titles with less than four words. The Unbert model and its tokenizer are applied to create a dataframe that stores the titles and corresponding embeddings. Although we are using a distilled version of BERT, the embeddings were still huge and may affect the prediction time. In order to reduce the size of the embeddings, PCA was applied to reduce the size of the embeddings while keeping the important variations as possible. For the embeddings, we found that the sweet spot is about to be 2500 components explaining about 0.82 of the total variables while reducing the size to about ⅗. Then, a function was used to calculate the cosine similarity between 2 embedding vectors of news titles.

However, many users in the test set have no prior history or the history is based on newer articles that have never appeared in the training set. Therefore, a collaborative filtering model was used by creating the embeddings for the current user's click history and then finding the embedding of the in the training set users closest to the in the test set.

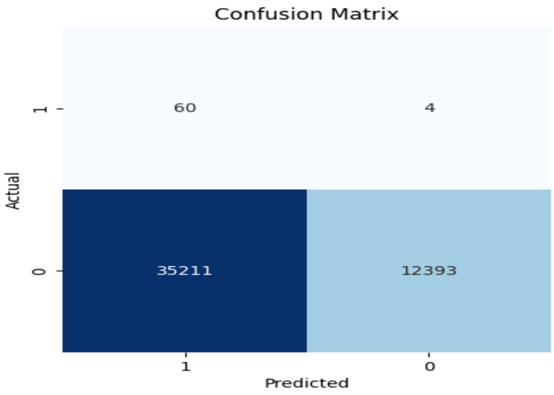
Then a click predictor was created to give binary click decisions of the user. The threshold can be further optimized by iterating it on all the users, however for our computational power, it is very hard to do so. Therefore, for now, we will be only presenting the performance based on 100 random users instead of the entire test set. We have managed to achieve 0.59 mean auc on the small test set which is comparable to models like LibFM (Rendle, 2012), which extracts TF-IDF features from users' browsed news and candidate news, and concatenates them as the input. If the model is run on the larger dataset, it is expected that the performance would also increase.

3.3 Evaluation Metrics

Besides the evaluation metrics used by the other recommendation models including: AUC-ROC, ranking the most relevant items for the user, MRR, ranking the most relevant items at the top of the recommendations, and nDCG@K with K = 5 and 10, which looks at the top 5 and 10 items in terms of relevancy, we also looked at the recall of the model, which looked at the proportion of relevant items that were recommended out of all the relevant items available. So out of all the relevant news in the news articles available in the test set, 81% of them were correctly recommended to the user. The performance is the average of these metrics on all impression logs. Since test set labels of MIND-large are not provided, the test performance is obtained through testing on the MIND-small dataset.

3.4 Relationship between metrics and ROI

Assuming the implementation of a recommendation system, it is expected that the status quo without the system will be 0.5 for all metrics. This is because the system will be displaying recommendations at random. However, after the implementation, the AUC is expected to increase to 0.59, which translates to a 18% performance increase in recommending the most relevant news to the user. This improvement in customer experience will likely result in an increase in customer satisfaction and engagement, translating to higher traffic and subscription.



Our optimization strategy for the system was focused on minimizing negatives rather than false positives. This is because false negatives can lead to poor user experience, where the user may miss out on valuable items. On the other hand, false positives are usually less harmful, as the user can simply ignore irrelevant recommendations. By reducing the risk of users missing out on valuable content, the system can improve customer loyalty and retention, leading to increased revenue in the long run. However, the threshold for optimization should be adjusted based on the performance in production. The system's performance may be affected by various factors such as changes in user behavior, content availability, or external events in production. Therefore, it is important to continuously monitor and adjust the system's performance in production to ensure that it continues to provide relevant and accurate recommendations. We have included some CI/CD integration later on in the production level considerations.

4. Production-level considerations

4.1 Envisioned End-to-End Solution and Technical Architecture

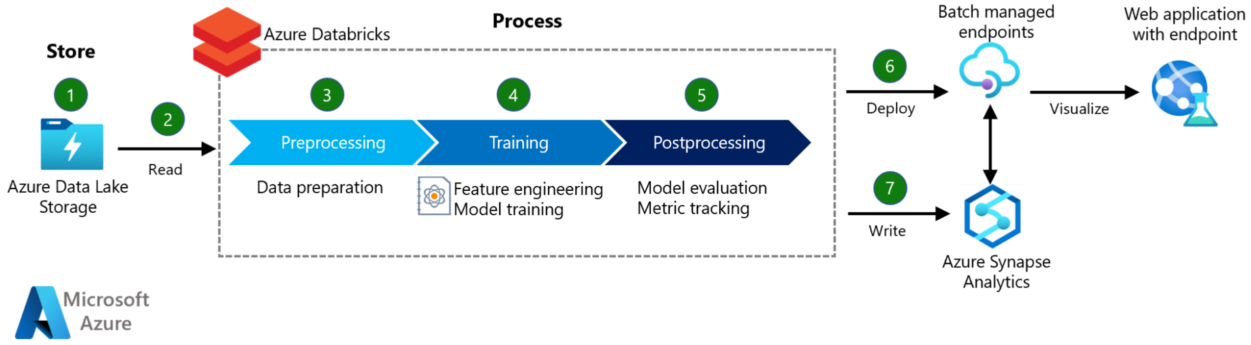


Figure 4.1: Technical Architecture using Azure ML

Data Storage & Ingestion: For this, we use Azure Data Lake Storage which serves as a scalable and robust repository for storing large volumes of user and consumer behavior data. It provides reliable data storage capabilities to support this recommendation system's needs. Azure Databricks seamlessly connects to Azure Data Lake Storage, enabling efficient data ingestion. This integration facilitates preprocessing and training steps required for model registration, ensuring the readiness of data for subsequent analysis.

Data Preprocessing: The preprocessing stage plays a pivotal role in refining the data for optimal utilization within the recommendation system. Through comprehensive cleansing, transformation, and preparation processes, the data is shaped into required format for enhancing the accuracy and effectiveness of subsequent modeling efforts.

Model Training: Azure Databricks conducts two key steps during model training: feature engineering and model training. Leveraging the preprocessed dataset, feature engineering uncovers relevant patterns and insights to explain user behavior. Model training employs advanced algorithms and techniques to develop the best-performing recommendation model. LSTUR model combines session-based and user-based modeling techniques, leveraging both short-term interactions and long-term preferences to provide personalized and diverse recommendations that align with users' evolving interests.[4]

Model Evaluation and Selection: Post Processing involves a rigorous evaluation and selection process, aiming to identify the most effective recommendation model. Several Models are trained and based on the best grouped AUC score the Model to implement the recommendation system is selected. Grouped AUC score is considered as it provides group level evaluation and supports group based personalizations and also it is robust to imbalance data.

Model Deployment: Azure Databricks can be used for model management, ensuring its availability for deployment. Through endpoints, the recommendation model is exposed to front-end display systems. These endpoints facilitate seamless integration, enabling access to new data via new endpoints. The architecture supports both batch and near-real-time recommendations, catering to diverse user needs.

Continuous integration(CI)/Continuous Development(CD): Once after the deployment, To assess the impact of the recommendation system implementation, periodic model retraining is necessary. After the first year, an analysis is required to compare the performance of the recommendation system against the status quo. to check for the need of improvement(5).

- **Updated Dataset:** During this period as the user interactions and subscriptions grow we continue collecting user interactions and relevant data to capture changes in user preferences, content trends, and news agency objectives. Also if there is any additional data that helps to improve in providing proper recommendations we also collect and integrate this data into the training pipeline for subsequent model retraining. Azure ML's data ingestion and preprocessing capabilities with Azure Data Lake facilitate this process.

- **Model Retraining, Evaluation & Iterative Improvements:** Based on the evaluated results and user engagement rate we can implement necessary changes to the recommendations system such as modifying the algorithm, doing feature engineering or we incorporate the updated dataset, retrain the model, and evaluate its performance using appropriate evaluation metrics Grouped AUC score.

4.2 Legal and IP aspects related to the dataset

"MIND: A Large-scale Dataset for News Recommendation," published at ACL 2020 (ACL 2020 refers to the Annual Meeting of the Association for Computational Linguistics, which is a premier conference in the field of natural language processing (NLP) and computational linguistics. ACL is one of the most influential conferences where researchers and practitioners present their latest findings, advancements, and innovations in various areas of NLP, including text mining, machine learning, language understanding, and generation.), in all relevant works and the dataset is free to download for research purposes under Microsoft Research License Terms.

4.3 Alternative or Complementary Datasets

Given the potential limitations of the provided dataset, it is important to explore alternative or complementary datasets and consider building an "in-house" dataset to overcome these constraints.The limitations include class imbalance in impressions, inconsistency in title length, and the lack of user responses, which may impact the performance and generalizability of the recommender system. Gathering additional data, such as user feedback or explicit preference indications, can augment the dataset and improve the performance and personalization capabilities of the recommender system.

In addition to addressing these limitations, it is worth exploring alternative data sources, such as external datasets or public APIs, to enrich the existing dataset. Additionally, building an "in-house" dataset through potential methods mentioned below can provide valuable insights and enhance the system's recommendation capabilities.

Potential Methodologies for Building an "In-House" Dataset:

- **Contextual Information Integration:** Gathering and integrating demographic data, user profiles, social network connections, and location data alongside user behavior.
- **Explicit Preference Indications:** Allowing users to express their preferences through explicit ratings, feedback, and customization options enables the collection of precise and detailed user preferences.
- **Surveys and Research Studies:** Conducting targeted surveys and research studies within the organization's user base helps gather specific information about user preferences, interests, and needs.
- **Data Augmentation Techniques:** Leveraging external data sources, public APIs, or integrating data from related domains can augment the existing dataset, offering a broader perspective of user preferences and behavior.

By considering alternative or complementary datasets and employing a methodology for building an "in-house" dataset, it is possible to mitigate the limitations of the provided dataset and develop a more robust and accurate recommender system that aligns with the business goals of the organization.

5. References

1. Das, T. K. (2015). A customer classification prediction model based on machine learning techniques. In 2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT) (pp. 211-215).

2. Graham, S., Min, J. K., & Wu, T. (2019). Microsoft recommenders: tools to accelerate developing recommender systems. In RecSys '19: Proceedings of the 13th ACM Conference on Recommender Systems (pp. 707-708).

3. Humble, J., & Farley, D. (2010). Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation.

4. Ihlström, C., & Palmer, J. (2002). Revenues for Online Newspapers: Owner and User Perceptions. Electronic Markets, 12(4), 228-236.

5. Mitra, K., Dey, S., & Roy, S. (2019). A Hybrid Approach to Recommender Systems using Autoencoders and Convolutional Neural Networks. arXiv preprint arXiv:1910.14025v2.

6. Rendle, S. (2012). Factorization machines with libFM. ACM Transactions on Intelligent Systems and Technology, 3(3), 1-22.

7. Wang, H., Zhang, F., Wang, J., & Huang, K. (2019). Neural News Recommendation with Long- and Short-term User Representations. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 425-434).

8. Wu, F., Qiao, Y., Chen, J.-H., Wu, C., Qi, T., Lian, J., ... & Zhou, M. (2020). MIND: A Large-scale Dataset for News Recommendation. ACL 2020.

9. Azure Databricks: Azure Databricks - Unified Analytics.