News Article Recommendation System

INFO 442 Final Project

Kevin Shi, Richardson Chhin, Benjamin Leung, Kathryn Swatek

1. Problem Statement
2. Dataset Overview
3. Data Cleaning
4. Exploratory Data Analysis (EDA)
5. Modeling & Evaluation
6. Lessons Learned
7. References
8. Q&A

Agenda

Problem Statement

- MSN.com (owned by Microsoft) received 168M visitors in June 2025
- Minimal research into developing recommenders for news articles compared to other domains
- Convenient access to trustworthy information is vital to our society, especially during key elections

- **Project Goal**: Develop a news recommender system that delivers personalized content to users efficiently

- Available courtesy of Microsoft News Dataset (MIND)
- Anonymized behavior logs from Microsoft News website from 50,000 randomly sampled users
- Split into training and validation sets
  - Behaviors.tsv: Users' behavior logs tracking impressions and clicks
  - News.tsv: News article information (title, abstract, category, etc.)
- Data collected from October 12 to November 22, 2019

Dataset Overview

Behaviors

- Convert time column to datetime
- Fill NaNs for users with no history
- Convert history from string to list
- Parse impressions into pairs & explode into 2 columns

News

- Remove all articles missing abstracts
- Remove references to dropped articles from behavior logs

Embeddings

- Load into dictionaries

Item Vector

- One-hot-encode category & sub-cat to get binary vector
- TFIDF vector of Title & Abstract
- Word2Vec (Gensim) vector of Title & Abstract
- BERT vector of Title & Abstract using Hugging Face model

Data Cleaning

User Overlap	
Measure	**Count**
Unique users in Train	50,000
Unique users in Val	50,000
Users in both Train & Val	5,675

EDA – Users

News Article Overlap	
Measure	**Count**
Unique news IDs in Train	51,282
Unique news IDs in Val	40,393
News IDs in both Train & Val	27,186

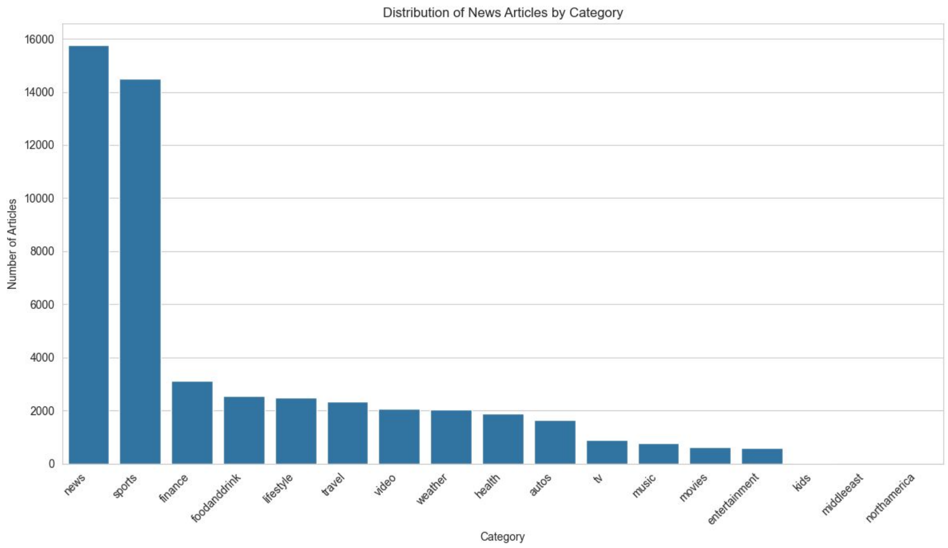
*Percentage of news_id in validation also in training: 67.30%
*CTR on seen news articles: 0.0932
*CTR on unseen news articles: 0.3308

EDA – Items

What is the content of the news articles we're recommending?

17 total article categories:

category	
news	15774
sports	14510
finance	3107
foodanddrink	2551
lifestyle	2479
travel	2350
video	2068
weather	2048
health	1885
autos	1639
tv	889
music	769
movies	606
entertainment	587
kids	17
middleeast	2
northamerica	1



264 total article subcategories

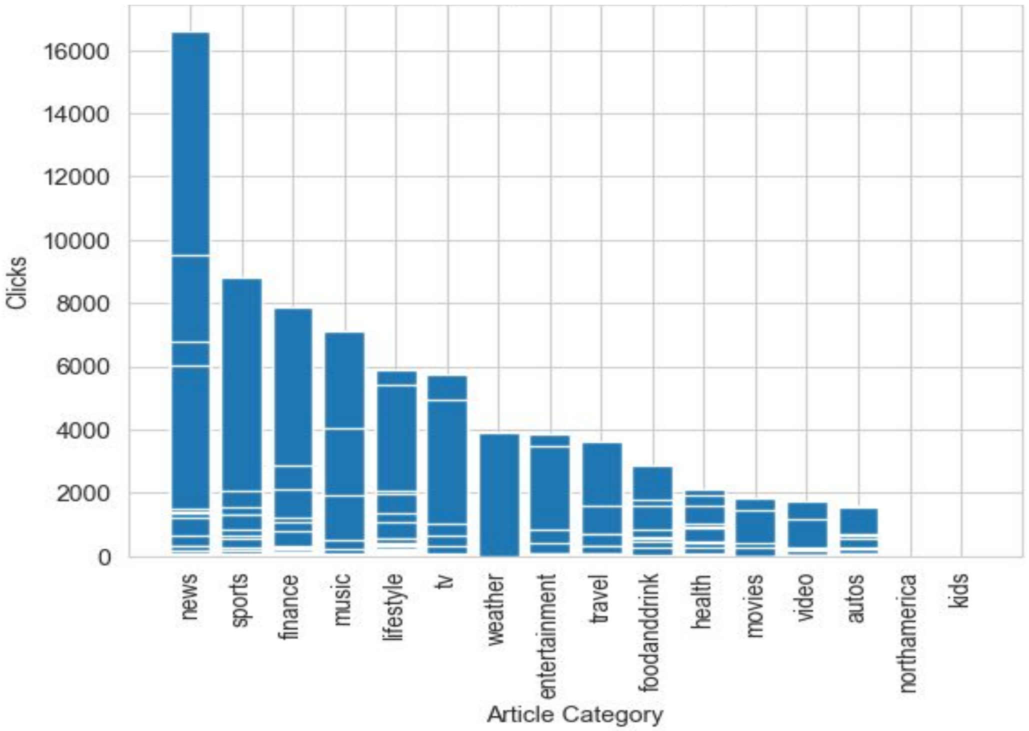
Top 20 subcategories:

subcategory	
newsus	6564
football_nfl	5420
newspolitics	2826
newscrime	2254
weathertopstories	2047
newsworld	1720
football_ncaa	1665
baseball_mlb	1661
basketball_nba	1555
newsscienceandtechnology	1210
news	1185
newstrends	1176
more_sports	1065
travelarticle	1042
travelnews	902
lifestylebuzz	894
autosnews	837
basketball_ncaa	774
financenews	697
finance-real-estate	584

EDA

Who are our users and how do they interact with the news?

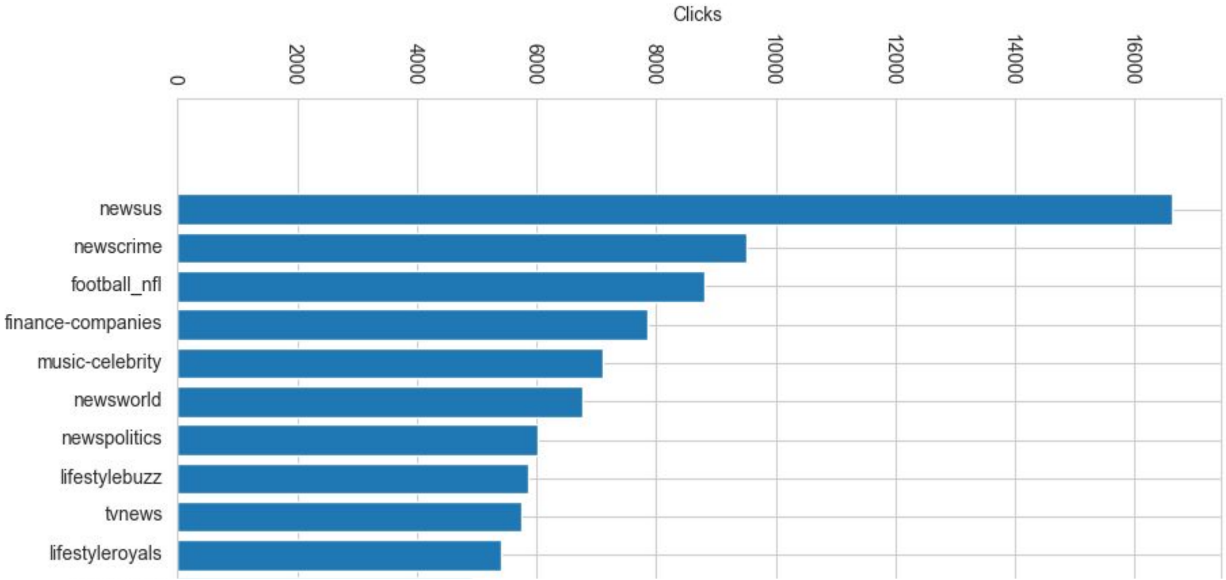
User Clicks by Category:



EDA

Who are our users and how do they interact with the news?

User Clicks by Subcategory (Top 10):



EDA

EDA

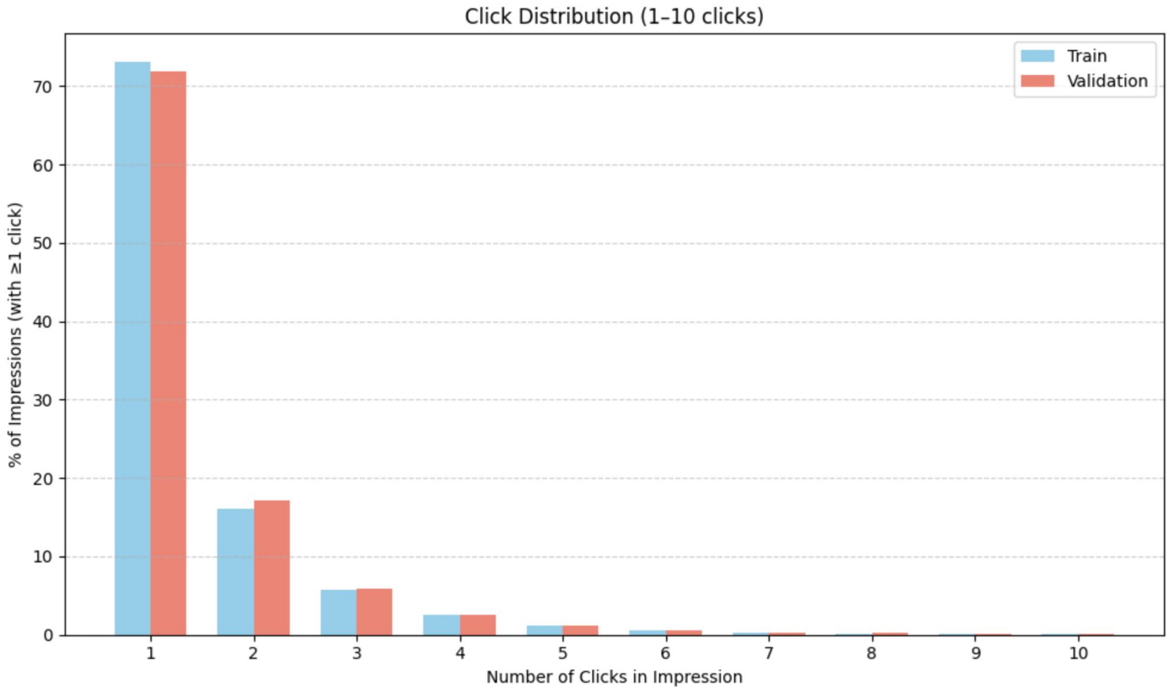
Total # of Impressions: Train – 156963, Val – 73152

User History Length (Impressions with ≥1 Click)			
Dataset	Average	Median	# Impressions
Train	31.67	18.0	151718
Val	31.57	18.0	69372

Candidate News Articles (Impressions with ≥1 Click)			
Dataset	Average	Median	# Impressions
Train	36.04	24.0	151718
Val	37.27	23.0	69372

Daily Average CTR		
Date	Average	
2019-11-09	0.111314	Train
2019-11-10	0.128754	Train
2019-11-11	0.131336	Train
2019-11-12	0.113323	Train
2019-11-13	0.109054	Train
2019-11-14	0.094920	Train
2019-11-15	0.102925	Val

EDA



Train and validation sets have similar click distributions

Goal: Build a content-based system that compares a user profile with item profiles to suggest items similar to those the user has previously interacted with

Process:

1. **Create Item Profiles**: Convert each news article into a vector that represents its content. We experimented with 3 different NLP techniques.
2. **Create User Profiles**: Generate a profile for each user by averaging all the articles they have previously clicked. This creates a single vector representing their unique interests.
3. **Rank & Recommend**: Use cosine similarity to measure the angle between a user's profile and the profile of a candidate article. Articles with the highest similarity score are then recommended to the user.

Modeling Approach

Goal: Convert text of each news article into an item profile vector

1. **TF-IDF**: Created a 5,000-dimension vector for each article representing the TF-IDF scores for the top 5,000 words in our dataset
2. **Word2Vec**: Trained a model on our news articles and created a 100-dimension profile for each by averaging the vectors of all its words
3. **BERT**: Used a pre-trained BERT model to generate a single 768-dimension vector representing each article's overall meaning

Creating Item Profile

- ➢ **ROC AUC:** measures a model's ability to distinguish between classes
- ➢ **Mean Reciprocal Rank (MRR):** position of first relevant recommendation
- ➢ **nDCG@K:** evaluates ranking quality by giving more weight to relevant items that appear higher up on the list
- ➢ **Recall@K:** fraction of relevant items retrieved at K
- ➢ **Precision@K:** fraction of retrieved items that are relevant at K
- ➢ **MAP@K:** overall ranking precision across positions at K

Evaluation Metrics

Model	Set	ROC AUC	MRR	nDCG@5	nDCG@10
TFIDF	Train	0.5986	0.3094	0.2846	0.3455
TFIDF	Val	0.5506	0.2880	0.2649	0.3209
Word2Vec	Train	0.5978	0.2989	0.2777	0.3393
Word2Vec	Val	0.5488	0.2825	0.2593	0.3154
BERT	Train	0.5952	0.3024	0.2797	0.3407
BERT	Val	0.5612	0.2892	0.2689	0.3244

Evaluation

Top Results

Rank	Team	AUC	MRR	nDCG@5	nDCG@10
1 OCT. 05, 2021	UniUM-Fastformer-Pretrain	0.7304	0.3770	0.4180	0.4718
2 SEPT. 02, 2021	MINER	0.7275	0.3724	0.4102	0.4661
3 AUG. 08, 2021	UniUM-Fastformer	0.7268	0.3745	0.4151	0.4684
4 SEPT. 14, 2022	pengwj	0.7256	0.3720	0.4101	0.4660

*Where Our Results Fall

Rank	Team	AUC	MRR	nDCG@5	nDCG@10
273 JUN. 07, 2021	chang861224	0.5703	0.2769	0.2945	0.3495
274 JUN. 15, 2022	group_5	0.5680	0.2575	0.2698	0.3268
275 FEB. 12, 2021	lwj	0.5519	0.2469	0.2573	0.3136
276 JAN. 05, 2022	shoemaker	0.5397	0.2475	0.2574	0.3135

Bottom Results

Rank	Team	AUC	MRR	nDCG@5	nDCG@10
309 MAR. 19, 2022	leemeng	0.4800	0.2150	0.2197	0.2743
310 APR. 14, 2022	pevnak	0.4798	0.2136	0.2198	0.2758

Average Over All Models (Val)

ROC AUC	MRR	nDCG@5	nDCG@10
0.5535	0.2866	0.2644	0.3202

Evaluation

Screenshots from "Leaderboard" section on MSNews (https://msnews.github.io/)

Model	Set	Recall@5	Recall@10	Precision@1	Precision@5	Precision@10	MAP @10
TFIDF	Train	0.4090	0.5848	0.1564	0.1044	0.0773	0.2557
TFIDF	Val	0.3747	0.5357	0.1503	0.0973	0.0717	0.2394
Word2Vec	Train	0.4080	0.5854	0.1412	0.1037	0.0772	0.2478
Word2Vec	Val	0.3697	0.5316	0.1440	0.0956	0.0709	0.2337
BERT	Train	0.4074	0.5832	0.1471	0.1035	0.0768	0.2503
BERT	Val	0.3853	0.5445	0.1464	0.0999	0.0728	0.2410

Evaluation

- ➢ Best Performance: BERT
  - Val AUC: 0.5612
  - Val MRR: 0.2892
  - Val nDCG@5: 0.2689
  - Val nDCG@10: 0.3244
- ➢ TFIDF achieves surprisingly strong performance – strong keyword-topic alignment!
- ➢ Performance gap between train & validation is consistent across models (~3–5% drop)
  - All models generalize well on validation set
  - Possibility of slight overfitting?
- ➢ MRR suggests first relevant hit is in top 3-4
- ➢ nDCG@10 performs better than nDCG@5
- ➢ Recall@10 performs better than Recall@5

Key Findings Summary

- Content-based recommender systems
- Text representation techniques: TFIDF, Word2Vec, and BERT
- Offline evaluation metrics (meanings and calculations)
- Complexities of news recommendation
  - Future work:
    - Integrate temporal considerations to account for news freshness and user interest shifts over time
    - Address the cold start problem
    - Model hyperparameter tuning
    - Offline evaluation
    - Advanced deep learning techniques (multi-view neural networks, transformer-based architectures, attention mechanisms, etc.)

Lessons Learned

References

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.

Q&A