Breaking the News Content Bubble

A Hybrid Recommendation System Approach

Kaushika Uppu, Miranda Billawala
CMPE 256 Spring 2025

Introduction

Motivation

Social media and news outlets, with the rise of highly personalizable recommender systems, tend to push only content a user is likely to agree with

Leads to the creation of a news content bubble or echochamber that solely reflects a user's views

Objective

Develop a news recommender system where users recommended articles they are likely to click on

Incorporate recommendations of articles within categories user prefers, but fall outside their typical readings

Dataset

Microsoft News Dataset (MIND)

Behavior logs of 50,000 users for ~65,000 articles

Each log has a user and an article they were shown, and whether they clicked on it or not

Algorithms

Neural Collaborative Filtering

Uses both matrix factorization and multi-layer perceptron to predict interactions between users and articles

Chose this to capture non-linear relationships between users and articles

Extended base model by incorporating article metadata from dataset

XGBoost

Uses decision trees and boosting w/parallel processing

Performs well with complicated datasets, minimal overfitting

Incorporated metadata with high dimensionality, but remained a fast algorithm

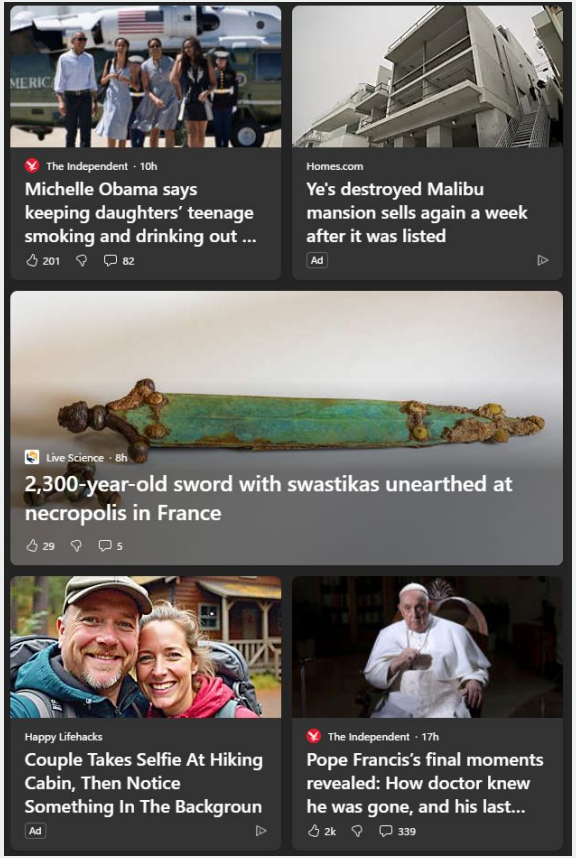
Data Preprocessing

User impression log: list of all articles shown + response (0 for not clicked, 1 for clicked)

- Exploded impression logs to have each article+impression on separate row
- Separated data into clicked/not clicked CSVs to help build user profiles later on

News Data (containing article metadata): combined training and validation sets and dropped duplicates

Data Cleaning: filled in all NA values, only present in some abstracts



Feature Engineering

Categories & Subcategories

Converted to categorical variables using OneHotEncoder (18 for category, 270 for subcategory)

PCA on combined category + subcategory columns and reduced to 18 components

Article Title & Abstract

Dataset had embeddings for title + abstract entities but ~25% of them did not have any ↛ worsened performance

Used SentenceTransformer (computes dense text embeddings that captures semantic meaning)

Encoded title and abstract together, used PCA to reduce (20 for NCF, 50 for XGB)

Sentiment Analysis

Used TextBlob (text processing Python module)

Takes in text and returns a polarity score that ranges from -1 to 1 (highly negative to highly positive)

Got sentiment scores for all titles and abstracts in the data as additional metadata inputs for models

Evaluation

Train-Test Split:

Provided by the training set

Training set - fifth week of impressions

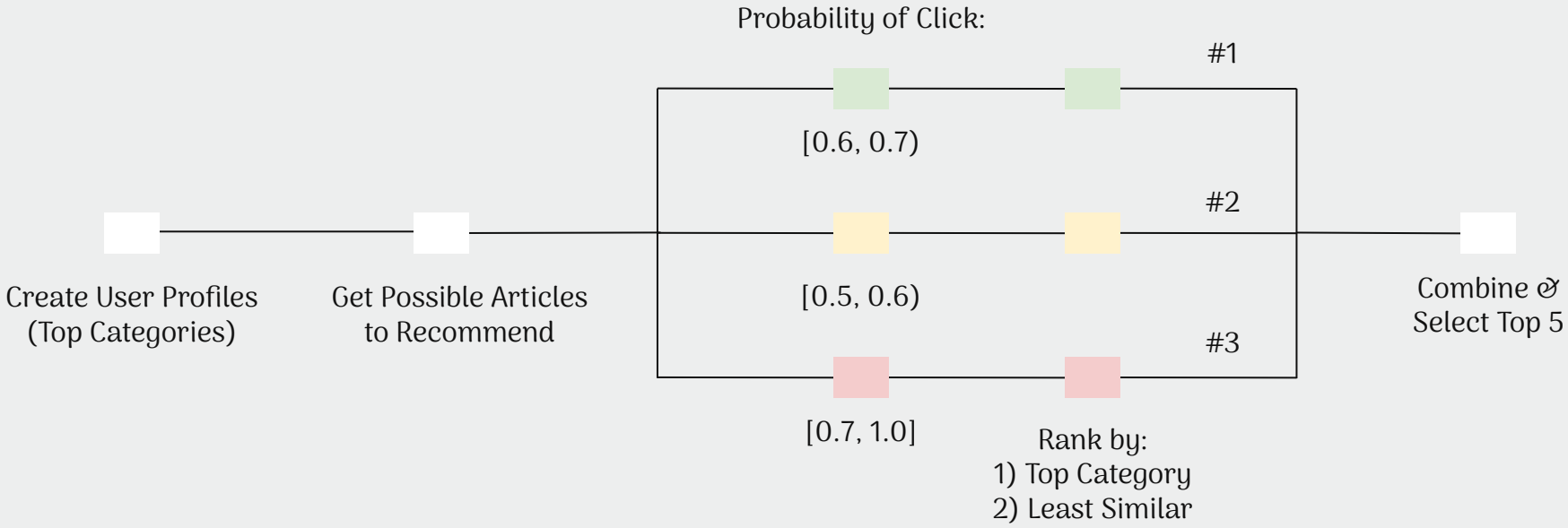
Test set - impressions on last day of fifth week

Both sets were extremely imbalance due to seeing far more articles than clicking

Set additional metrics to handle imbalances and prevent model from biasing towards the majority class (not-clicked)

Metric	XGBoost	NCF
ROC-AUC	0.644	0.547
Accuracy	0.771	0.587
Recall	0.414	0.458

User Recommendations



Things that Worked / Didn't Work

Calculated Title/Abstract Embeddings + Categories/Subcategories	Two features that resulted in significant increases in performance
PCA	Reduced feature dimensionality, especially of embeddings, to decrease number of columns without sacrificing score
Given Title/Abstract Embeddings	Thought they would significantly help predict click probability, but they did not improve performance, likely due to a number of abstracts missing one or both
Imbalanced Dataset	Model biased heavily towards majority class, tried weighting classes & saw minor improvements
Size of Dataset & Features	Large number of rows and high dimensional embeddings meant very long runtimes and on many occasions, the kernel crashing
Feature Engineering	Additional features saw very little improvements (sometimes even made the model worse) while increasing computational costs

Conclusion

Significance

Building a model which can recommend articles that extend a user's knowledge will help solve the current widespread issue of user content bubbles

Takeaways

Dataset ended up being ill-fitted for the task, resulting in low scores and inability to test our framework for recommendation

Future Work

Better fitting feature engineering

Better aligned dataset

Once we can get a more consistent and reliable click predictor, implement testing using methods such as A/B testing

Implement sentiment analysis on articles to recommend which may better decide what are alternate perspectives

Thank You!

Questions?