Building a Recommendation System for Pratilipi

Soham Gupta

February 2025

1 Introduction

Pratilipi is a platform for stories, and the objective of this assignment is to build a predictive model to recommend at least five stories (pratilipis) to users based on their historical reading behavior. The model should predict future reading preferences using a dataset containing user interactions and metadata about stories.

2 Dataset Description

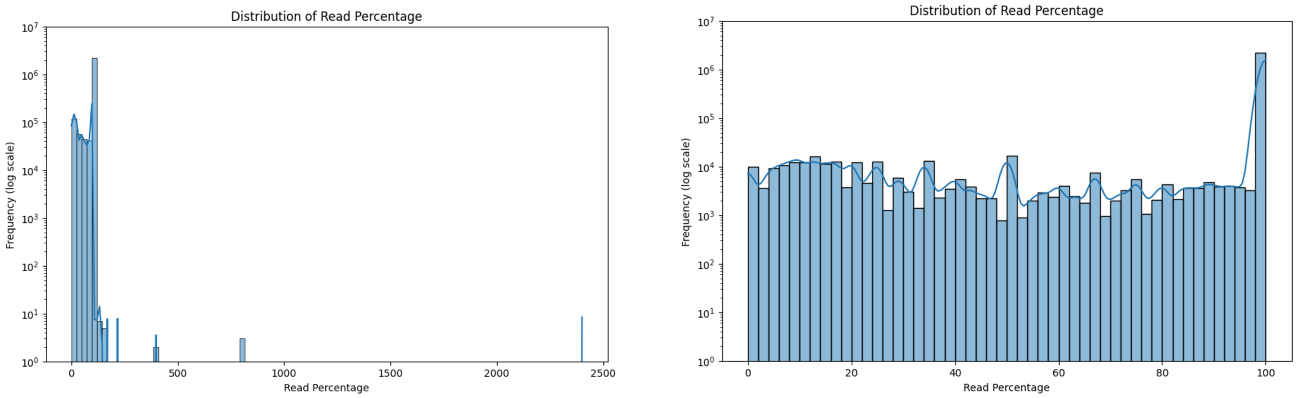
Two datasets are provided:

- **User_interaction.csv**: Contains user interactions with pratilipis, including user ID, pratilipi ID, read percentage, and timestamp.
- **Meta_data.csv**: Contains metadata about pratilipis, including author ID, pratilipi ID, category, reading time, and timestamps.

3 Exploratory Data Analysis (EDA)

The dataset was analyzed to understand user behavior and pratilipi metadata. Key findings include:

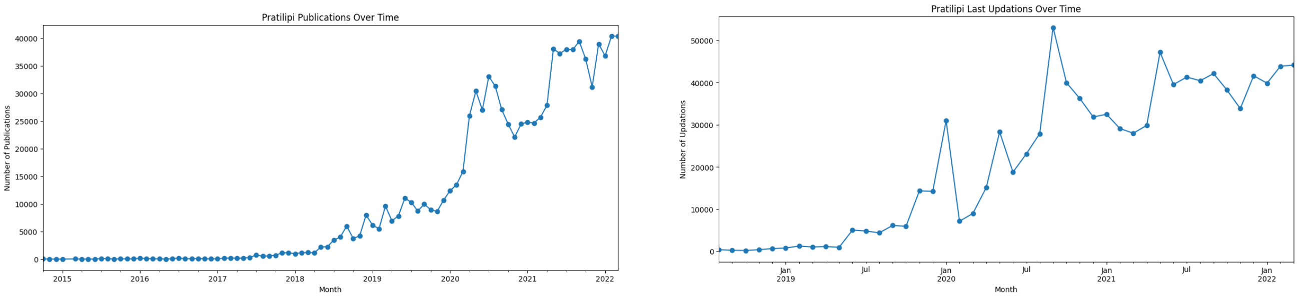
- Some read percentages exceeded 100, which were treated as outliers and removed.
- Distribution of read percentages shows a peak at 100, indicating complete readings.
- Categories like Romance and Suspense are the most published and read.
- Reading time varies widely, with some stories taking up to 80,000 seconds.
- The number of pratilipis published over time has increased consistently.



(a) Before Removing Outliers

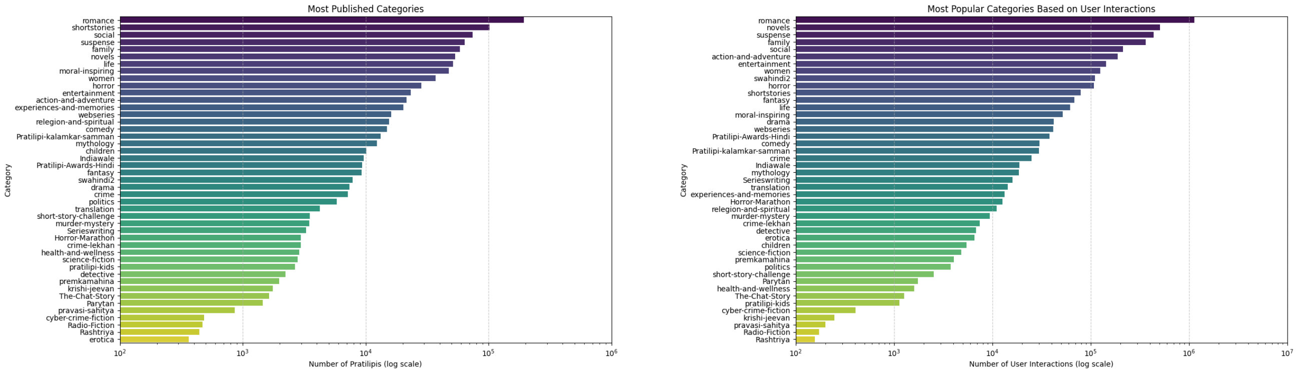
(b) After Removing Outliers

Figure 1: Distribution of Read Percentage Before and After Removing Outliers



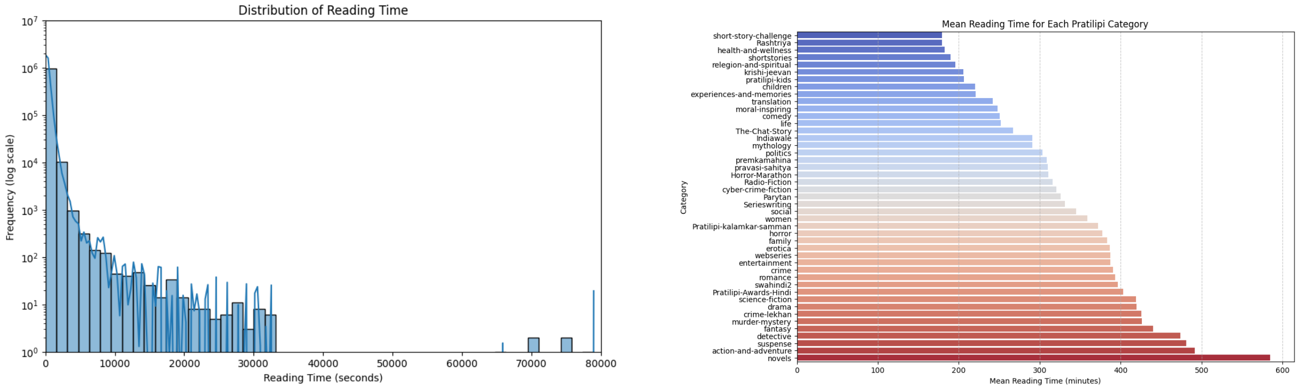
(a) No. of Publications over time (b) No. of updations over time

Figure 2: User interactions over time



(a) Most published Pratilipi categories (b) Most popular Pratilipi categories

Figure 3: Pratilipi Categories



(a) Distribution of pratilipi reading time (b) Avg reading time of each category

4 Methodology

4.1 Popularity-Based Model

- Recommends pratilipis based on how frequently they have been read by users.
- Ranks pratilipis by the total number of unique users who interacted with them.
- Uses author popularity (number of works published) as a secondary ranking factor.
- Useful for new users with no prior interactions (cold-start problem).
- Lacks personalization since it does not consider individual user preferences.

4.2 Collaborative Filtering (SVD)

- Uses Singular Value Decomposition (SVD) to factorize the user-item interaction matrix.
- Learns latent factors representing user preferences and pratilipi characteristics.
- Estimates missing values to predict future engagements.
- Struggles with data sparsity, as many users have limited interaction history.
- Poor generalization due to lack of sufficient user-item interactions.

4.3 Content-Based Filtering

- Recommends pratilipis based on metadata such as category, author, and reading time.
- Uses feature encoding and cosine similarity to find similar pratilipis.
- Applies K-Nearest Neighbors (KNN) to identify the most relevant recommendations.
- Does not rely on other users' behavior, making it independent of user interactions.
- Performs poorly due to weak feature representation and lack of explicit user preferences.

4.4 Hybrid Model (SVD + XGBoost)

- Combines collaborative filtering (SVD) with XGBoost ranking.
- SVD generates initial recommendations based on user interactions.
- XGBoost refines recommendations using metadata features like category encoding and reading time.
- Intended to improve ranking quality by incorporating additional features.
- Fails to significantly enhance results due to weak initial SVD predictions and insufficient training data for ranking (on an average 10 pratilipis per user in a dataset of over 9 lakh pratilipis).

5 Results

The models were evaluated using:

- Precision@5
- Recall@5
- NDCG@5 (Normalized Discounted Cumulative Gain)
- MRR (Mean Reciprocal Rank)

Comparison of different models and their performance is shown in Table 1.

Model	Precision@5	Recall@5	NDCG@5
Popularity-Based	0.00%	0.00%	0.00%
Collaborative Filtering (SVD)	0.02%	0.03%	0.01%
Content-Based Filtering	0.01%	0.04%	0.03%
Hybrid Model (SVD + XGBoost)	0.02%	0.05%	0.02%

Table 1: Comparison of Recommendation Models

6 Conclusion

- All models were able to generate pratilipi recommendations for users.
- However, the recommended pratilipis did not align with the ones that users actually read in the dataset.
- The popularity-based model suggested widely read pratilipis, but these did not match individual user preferences.
- Collaborative filtering (SVD) struggled due to data sparsity, leading to inaccurate predictions.
- Content-based filtering relied on metadata but lacked user preference signals, reducing relevance.
- The hybrid approach (SVD + XGBoost) attempted to refine predictions, but weak initial SVD recommendations limited improvements.
- The poor alignment between predictions and actual user behavior suggests the need for better feature engineering, richer user interaction data, and potentially deep learning-based approaches.
- Future improvements could include explicit feedback mechanisms, embeddings for user-item relationships, and neural network-based recommendation systems.