Team 27 Books Reviews Score predictions

In an era of millions of books online, discovering titles that match unique preferences is challenging. Our project develops a machine learning-powered book recommendation system using a rich Amazon dataset with over 3 million reviews across 212,404 books from 1996 to 2014. By decoding reading habits and review patterns, we deliver personalized suggestions to help readers find books they will enjoy and boost engagement on book platforms.

Business Objectives and Data Overview

Business Goals

Enhance user experience with personalized book recommendations, increase platform engagement, and drive higher retention and purchases benefiting publishers and retailers.

Dataset Components

Two main files: Books_rating.csv with 3M+ user reviews, and books_data.csv with metadata for 212,404 books, enabling collaborative and content-based recommendations.

Books_data.csv Dataset Details

Content

Metadata for 212,404 books including title, description, authors, cover image URL, preview links, publisher, publication date, categories, and ratings count.

Uses

Enables content-based recommendations and hybrid approaches by combining descriptive fields with collaborative data.

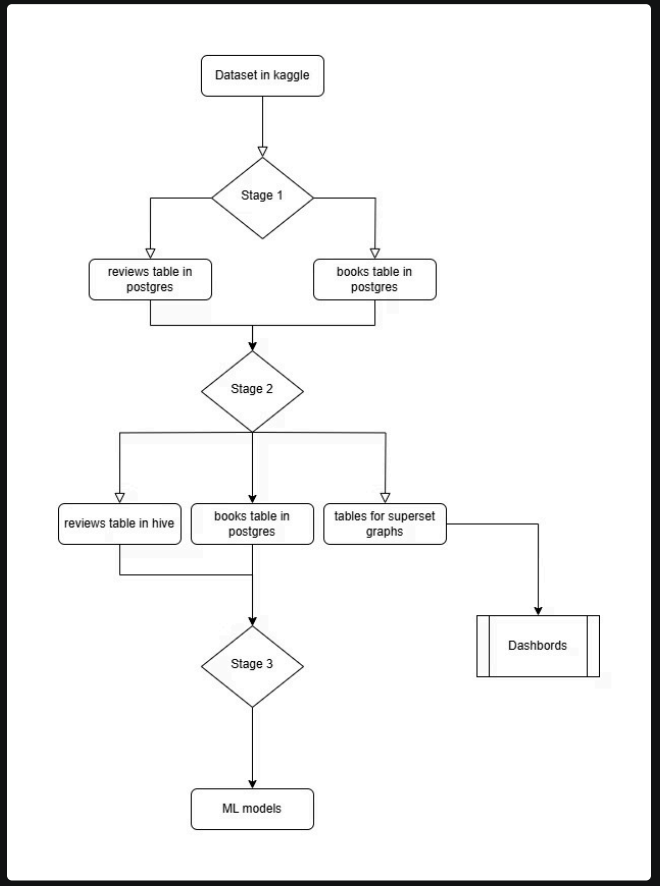
Data Pipeline Architecture and Preparation

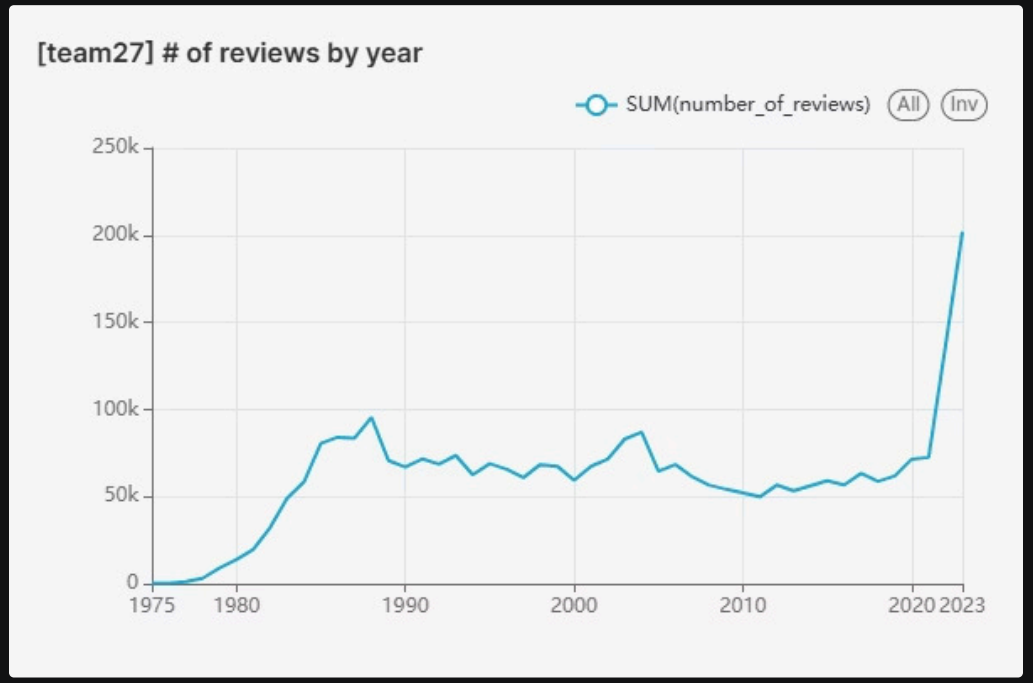
Pipeline Stages

1. Download, clean, and upload data to Postgres
2. Analyze data in Postgres, build dashboards, upload to Hive
3. Preprocess data and train machine learning models

Data Cleaning

- Remove rows with NULL critical columns
- Fix author columns to proper arrays
- Delete broken symbols in titles and descriptions





Data description

Dashboard overview

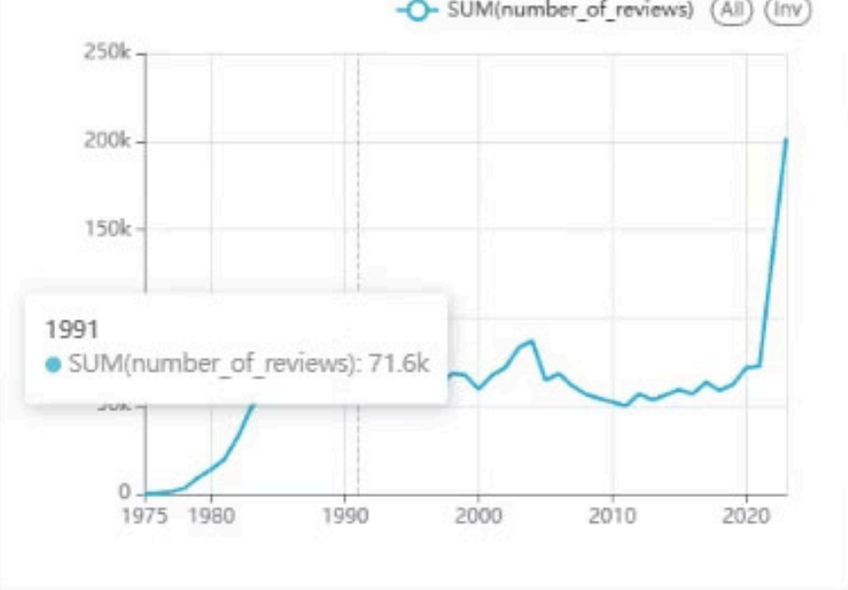
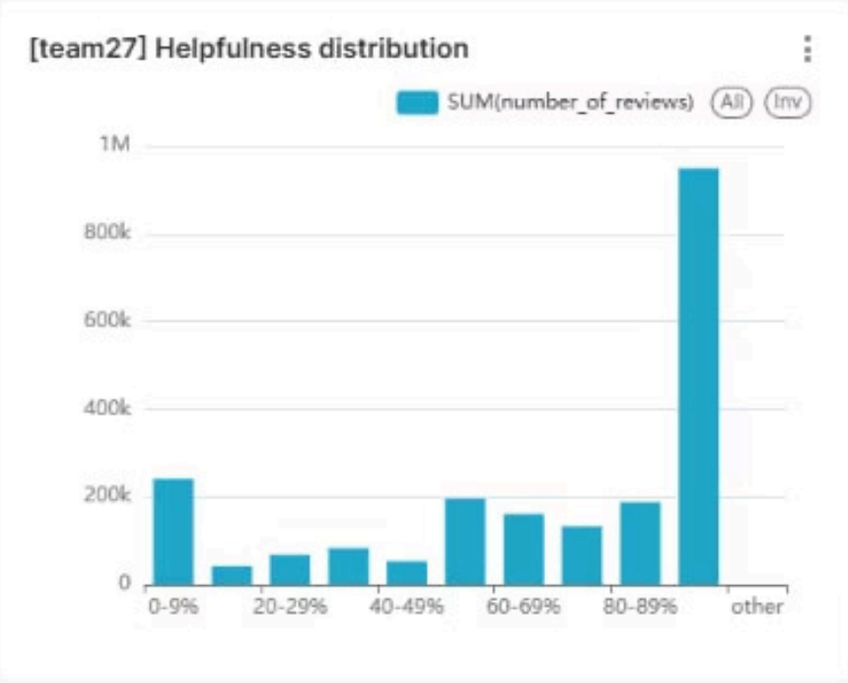


Chart is showing number of reviews per book's year. Slowly rising till 1990, then slightly falling and rapidly rising till today



Exploratory Data Analysis Insights

Rating Distribution	Positive reviews dominate; 2-star ratings are rare
Top Reviewed Books	Bestsellers dominate review volume
Review Trends Over Time	Slow growth until 1990, rapid rise post-2000
Publisher Performance	Indie publishers often outperform large ones
Most Active Users	Small group contributes disproportionate reviews
Helpfulness Analysis	Most reviews are helpful; mid-range ratings get more engagement

Key Insights and Recommendations

Rating Bias

Address skew by weighting low-rated reviews more in recommendations.

Temporal Signals

Leverage seasonal spikes for targeted marketing campaigns.

Publisher Trends

Partner with high-performing indie publishers for curated selections.

User Segmentation

Identify and incentivize super-reviewers to maintain engagement.

Feature Engineering and Text Processing

Key Steps

- Helpfulness score calculation with Wilson score
- Temporal features encoded cyclically
- Categorical encoding of authors, categories, and IDs
- Text processing with tokenization, stopword removal, and TF-IDF

Output

Train/test split saved as JSON with encoded features and metadata for model training.

Machine Learning Modeling Overview

We tried to train 4 models

1 FMRegressor

- factorSize: [4, 6, 8]
- initStd: [0.01, 0.1, 1.0]
- regParam: [0.01, 0.1, 1.0]

2 GBTRegressor

- maxDepth: [4, 6, 8]
- featureSubsetStrategy: ["log2", "sqrt", "onethird"]
- subsamplingRate: [0.6, 0.8, 1.0]

3 RandomForestClassifier

- numTrees: [10, 17, 25]
- maxDepth: [5, 7, 10]

4 LogisticRegression (multinomial)

- regParam: [0.05, 0.17, 0.25]
- elasticNetParam: [0.1, 0.25, 0.5]

Evaluation

model	RMSE	R2
RandomForestRegressor	1.4162094562456848	-0.438005245048173 95
LogisticRegression	1.422036996664622	-0.4498640489738863
GBTRegressor	1.428302134567891	-0.443209812345678
FMRegressor	1.449876542309876	-0.451763920123456

Challenges, Reflections, and Recommendations

1

Challenges

Interdependent stages required careful coordination; cluster performance issues caused delays; learning new technologies was demanding.

2

Recommendations

Improve cluster resource management and enhance team communication for knowledge sharing and efficiency.