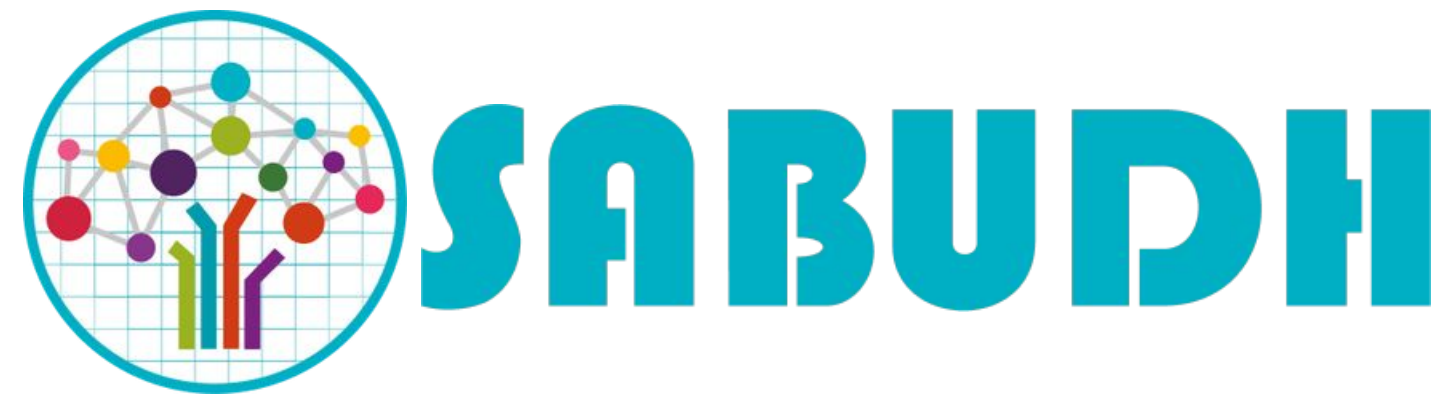


# *AI for Book Analysis*



## *Final Project Presentation*

20-12-2024

Project Mentor(s): Apoorav Mittal  
Aashish Rana  
Jaybrata Chakraborty

Presented by: Rohan Singh  
Ankita Saha  
Sanghamitra Goswami  
Shreya Chatterjee  
Jaykishan Padia

# ***Problem Statement***

- In today's world, the sheer volume of books available across different genres, platforms, and formats has made it increasingly difficult for readers to choose the right book that suits their preferences.
- The diversity of choices often leads to decision fatigue, missed opportunities to discover books of interest, and frustration for readers who wish to explore new content.
- This project aims to create a AI-driven book recommendation system.

# ***Project objectives***

01

To review existing research and methodologies, the techniques used.

---

02

To introduce the concept of recommendation systems(Collaborative and Content Based Filtering).

---

03

To develop a system and implement an user interface using Gradio for seamless user interaction.

---

# METHODOLOGY

# DATA FETCHING

- We fetched datasets from github repository.
- It was a goodreads book datasets.

## Datasets

---

### Meta-Data of Books

- Detailed book graph (~2gb, about 2.3m books): [goodreads\\_books.json.gz](#)
- Detailed information of authors: [goodreads\\_book\\_authors.json.gz](#)
- Detailed information of works (i.e., the abstract version of a book regardless any particular editions): [goodreads\\_book\\_works.json.gz](#)
- Detailed information of book series (Note: Unfortunately, the series id included here cannot be used for URL hack): [goodreads\\_book\\_series.json.gz](#)
- Extracted fuzzy book genres (genre tags are extracted from users' popular shelves by a simple keyword matching process): [goodreads\\_book\\_genres\\_initial.json.gz](#)

### Book Shelves

- Complete user-book interactions in 'csv' format (~4.1gb): [goodreads\\_interactions.csv](#)  
User Ids and Book Ids in this file can be reconstructed by joining on the following two files: [book\\_id\\_map.csv](#), [user\\_id\\_map.csv](#).
- Detailed information of the complete user-book interactions (~11gb, ~229m records): [goodreads\\_interactions\\_dedup.json.gz](#)
- User-[Book Club](#) mapping information: [book\\_clubs.json](#)

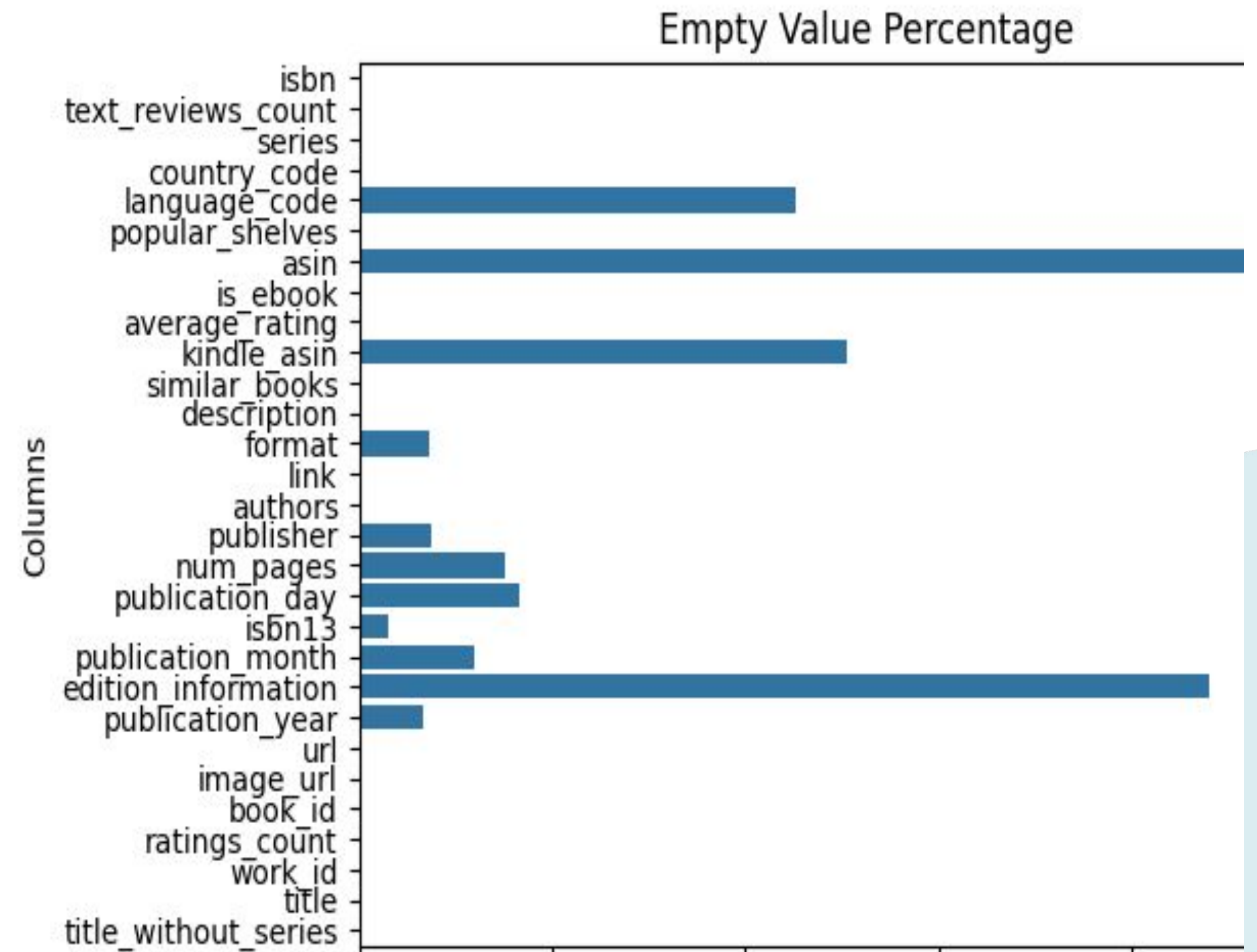


# Data Cleaning

Columns with more than 50% missing values and those that were irrelevant were removed to ensure data quality and relevance.

## Columns Retained:

- isbn
- title
- author
- description
- link
- image\_url
- average\_rating



# Data Processing

**Objective:** Ensure all book descriptions are in English for consistency in analysis.

**Method:**

- **Language Detection:** Used langdetect to identify the language of each description. If the detected language was not English, the text was flagged for translation.
- **Translation:** Leveraged googletrans to translate flagged descriptions into English, standardizing all text data.

**Tools/Libraries:**

- **langdetect:** To detect language codes of descriptions.
- **googletrans:** To translate non-English descriptions into English.

**Error Handling:**

- Managed LangDetectException errors from langdetect for cases with insufficient text, ensuring the process ran smoothly.

# Data Processing

- To test functionality, created a synthetic dataset where:
  1. Initialize number of users.
  2. Take a subset of the books available.
  3. Select books and distribute them as per gaussian distribution to different users.
  4. Edit the image links as per the new system goodreads has implemented.
  5. Generate user ratings by creating a gaussian distribution about the average rating.

## Data Storing and Retrieval:

- Data has been stored on MongoDB for storage and retrieval purposes.



## Data Processing Workflow



Fetch Data



Perform Preprocessing



Perform Operations



Generate Synthetic Dataset

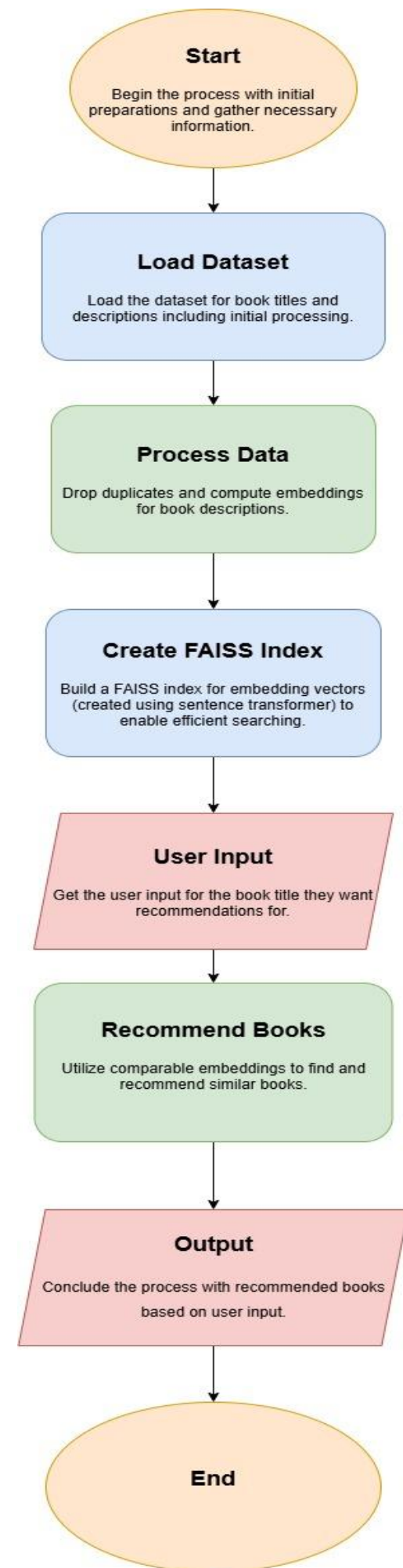


Upload to MongoDB



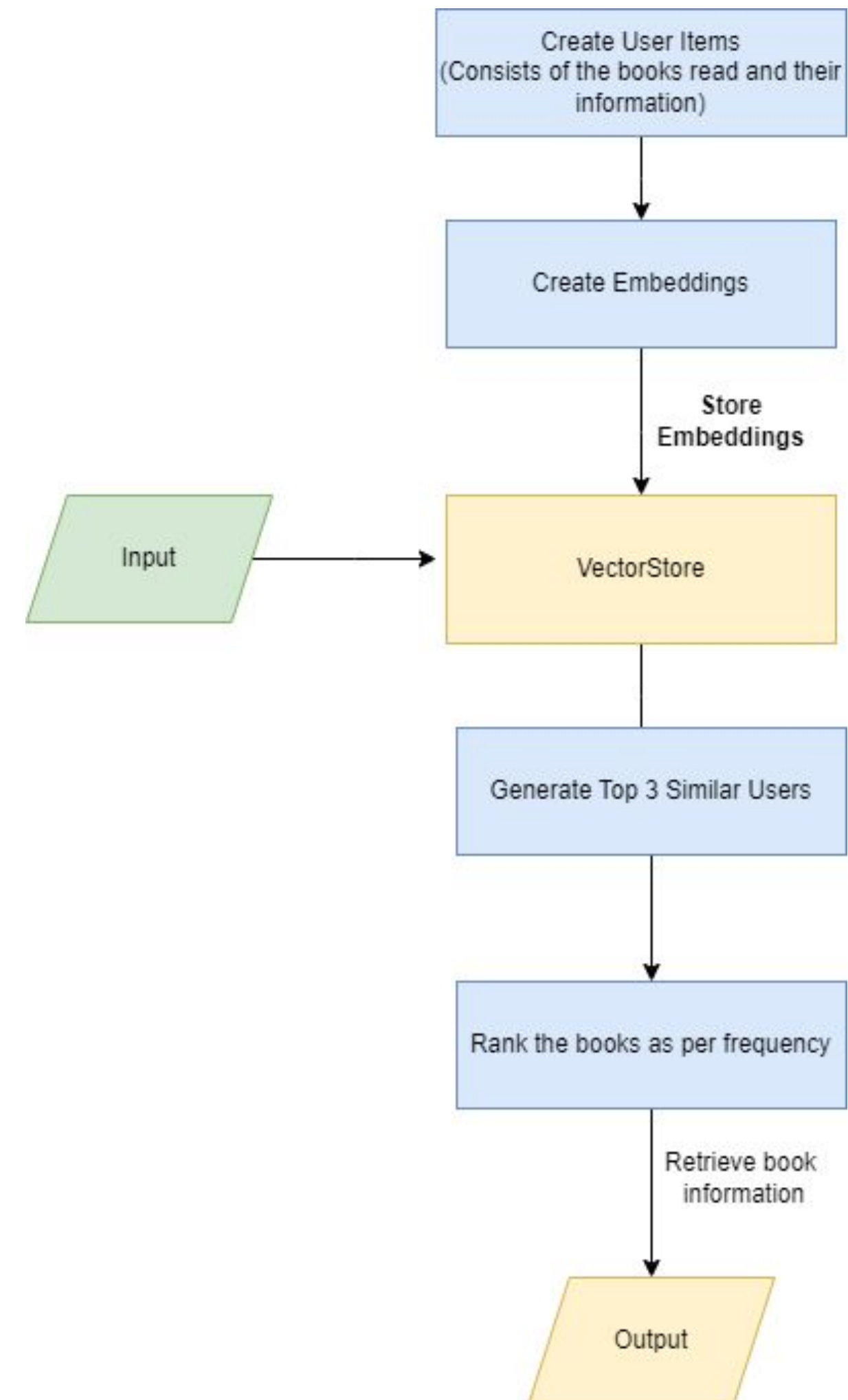
# Content Based Recommendation System

- Initially used word2vec to generate embeddings.
- For better contextual understanding and to deal with memory issues, used sentence transformers to generate embeddings.
- Computed semantic embeddings for book descriptions using the "all-MiniLM-L6-v2" SBERT model. Embedding generation with Sentence Transformer to capture the semantic meaning of book descriptions
- Indexed embeddings using the FAISS vectorsore for efficient similarity searches, for quick retrieval of similar books based on cosine similarity.
- Developed a function to recommend books based on a given title by searching for semantically similar descriptions.
- Demonstrated functionality with example input on Gradio interface, outputting relevant book recommendations.




# Collaborative Filtering

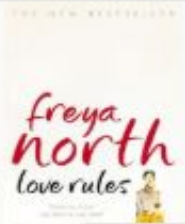
- Initially created embeddings using word2vec where recommendation was done based on the isbn number of the book read. Move on to using sentence transformers, so that greater details could be used and for greater contextual understanding.
- This method identifies users who are similar to the target user (based on their past behaviors) and recommends items that those similar users have liked.
- For each user we will create a string which will be a combination of all the books he has read and their information.
- Embedding model used: all-MiniLM-L6-v2
- Vectorstore: FAISS vector store
- Similarity Measure: Cosine Similarity




# Screenshots




The Bridge To ...




Love Rules



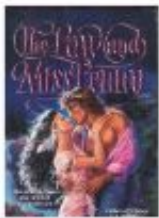
I Heart Christmas




Every Time a B...



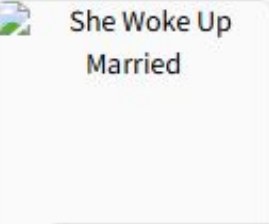
Heat (Blood B...



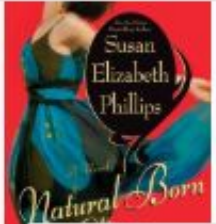
The Law and M...



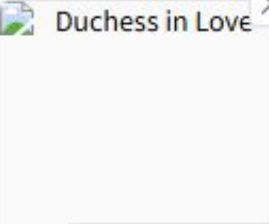
The Enchante...



She Woke Up ...



Natural Born C...



Duchess in Love

Generate Book Recommendations

User ID

1

1

500

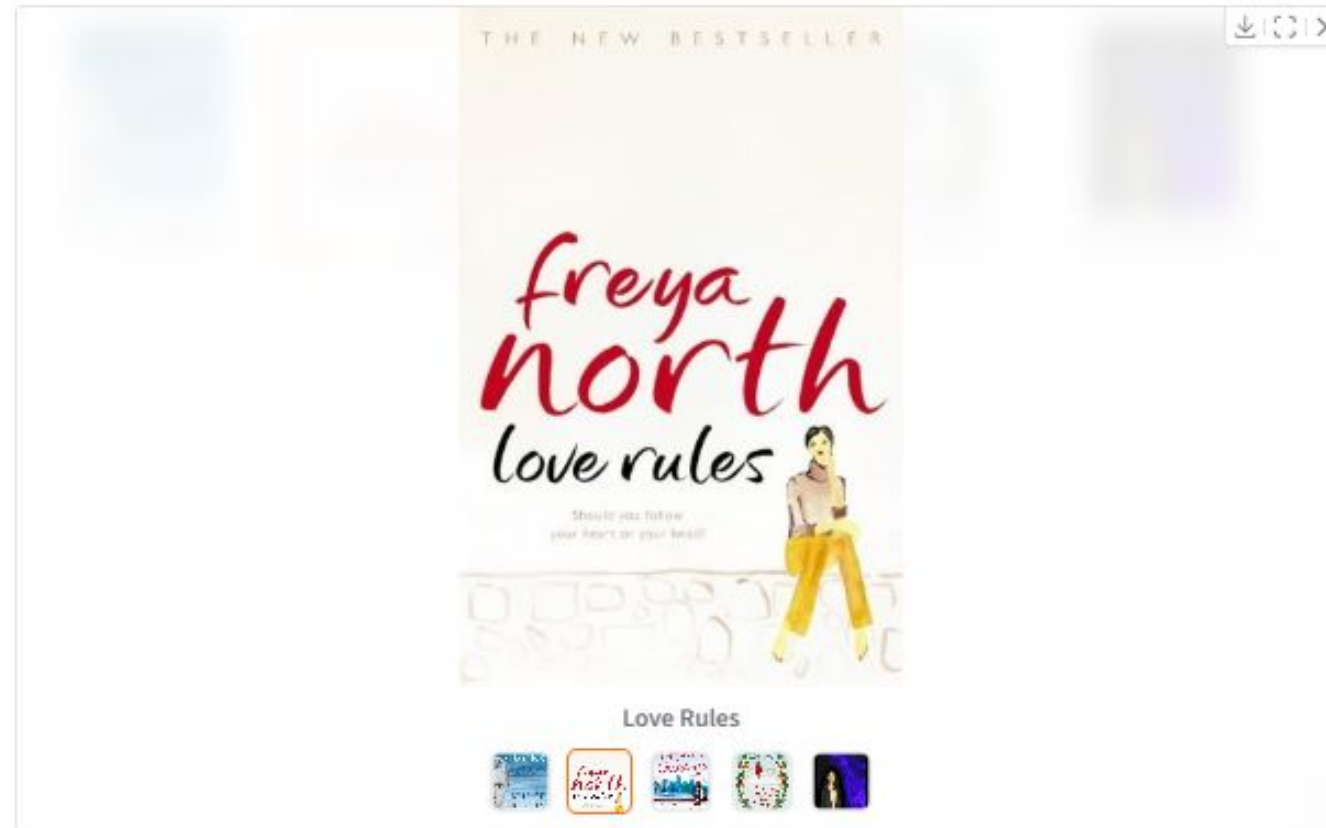
Generate Book Recommendations

Book Title

Love Rules



# Screenshots



```
1  ▼ {  
2    "Title": "Love Rules",  
3    "Author": "[{'author_id': '249163', 'role': ''}]",  
4    "Description":  
    "Is love ever enough? Thea - sensible and cautious - has always believed  
    in old-fashioned romance. Her best friend Alice is more of a 'fun first,  
    think later' kind of girl. But just recently they've both been behaving  
    out of character. When Thea Luckmore falls head over heels for a man she  
    meets on Primrose Hill and Alice Heggarty marries her best friend Mark,  
    both women are blissfully happy - for a while. But it's not long before  
    Alice admits to herself she likes the thrill of the chase more than the  
    easy pace of married life. And then Thea makes a shocking discovery - one  
    which forces her to rewrite her rules for everlasting love !"  
5  }
```

Generate Book Recommendations

User ID

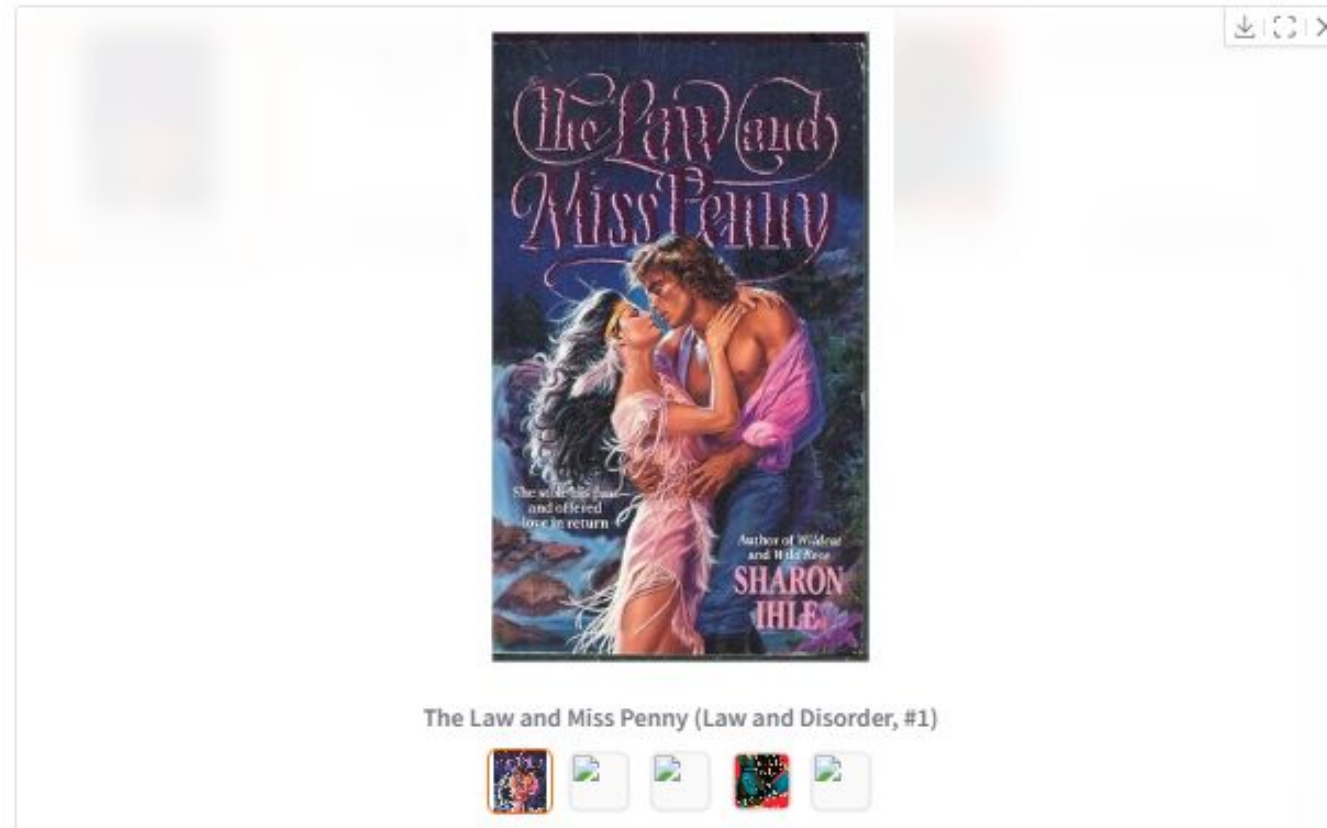
1



1



500



```
1  ▼ {  
2    "Title": "The Law and Miss Penny (Law and Disorder, #1)",  
3    "Author": "[{'author_id': '394530', 'role': ''}]",  
4    "Description":  
    "THROUGH HER EYES, HE SAW A WHOLE NEW WORLD When he awoke by the side of  
    the muddy road, with a throbbing head and a lovely woman bending over him,  
    he couldn't remember even his own name. Morgan Slater, U.S. Marshal, had  
    to accept pretty Mariah Penny's word that he was her cousin, Cain Law.  
    Mariah had good reasons for misleading Morgan, but there was an attraction  
    between the raven-haired beauty and the lawman that made her long to turn  
    cousinly conversation into more than a friendly encounter. Cain soon found  
    himself running afoul of the law he had sworn to protect, all in the name  
    of a forbidden passion."  
5  }
```

Generate Book Recommendations

Book Title

Love Rules

# Conclusion

- Successfully created a book recommendation system that provides personalized suggestions.
- Used different representations(word2vec and miniLM) to recommend books.
- Utilized collaborative filtering and content based filtering to provide recommendations.



# Future Scope

- Integrate additional features such as genres, authors, user demographic and user reviews for better accuracy.
- Incorporate advanced techniques to get better results.
- Create User interface and deploy it on AWS.



***Thank you***