Building a Book Recommender System

Abichal Ghosh (1225427294) Syed Asad Husain (1225380117)
Nikhil Kumar (1225658070) Meesum Ali Khan(1225453632)
Soham Nag (1225463863)

September 23, 2022

Abstract

A recommender system provides suggestions pertinent to a particular user. The goal of this project is to develop a hybrid recommendation system for recommending books to new and/or existing users, aiding the process of book search. The three main stages in this project are selecting a proper dataset, choosing a metric to evaluate the performance of the model and developing a baseline model for the recommender system. The next step would be to gauge our model against existing recommender systems to determine ways in which our model can be improved.

1 Group

Technical Background: Good familiarity with programming in Python, some experience with Machine Learning and relevant libraries in Python.

1. Soham Nag, sohamnag@asu.edu
2. Meesum Ali Khan, makhan35@asu.edu
3. Nikhil Kumar, nkumar82@asu.edu
4. Syed Asad Husain, shusain6@asu.edu
5. Abichal Ghosh, aghosh55@asu.edu

2 Introduction

Recommender systems are employed in some way or the other in practically every form of retail and services business. The multitude of services that the general public uses in their everyday lives constantly serves recommendations, be it in the form of video content, shopping items or even places to visit. Thus, recommender systems have become a very impactful part of our daily lives. The inspiration behind this project is to learn how recommender systems work, with a focus on book recommendation. The global book publishing industry amounts to about $100bn. With over 300,000 books published annually in the US alone, selecting a book to read can be challenging for a reader. Promoting a book to a consumer who has shown interest in similar books makes it convenient to boost sales of that book whilst also catering to the consumer's needs. This is the advantage that a recommender system is able to offer.

Recommender systems are based on the concepts of Clustering and Reinforcement Learning. The objective is to explore both Content-Based Filtering, which is an approach that analyzes and finds similarities between features that a user likes/dislikes, and also Collaborative Filtering, which tries to match users based on their tastes, and come up with recommendations. This concept of comparing features of books and users and finding some similarities is based on Clustering, which is a subpart of Unsupervised Learning. Furthermore, an attempt would be undertaken to test a hybrid approach, which will combine aforementioned approaches and give a better result. Through the process of building a book recommender

system, the intent is to understand the pipeline, challenges and intricacies of such a task.

3 Background

With a plethora of widespread information, it becomes overwhelming to make desired decisions for the most likable and appropriate content. Recommender Systems are developed and designed to recommend the most likely choice to each and every individual user/customer. Recommender systems can be understood as a branch of filtering algorithms which outputs favorable choices based on preferences. This results in saving a lot of time any user would otherwise spend searching for desired results.

These systems could be broadly classified into two categories:

1. Context Based Filtering
2. Collaborative Filtering

3.1 Context Based Filtering

In context-based filtering, product features are used to filter similar products. Explicit feedback and the user's previous actions are also considered while making the decision. This type of model is highly scalable as it is not dependent on any other user. Furthermore, it can also capture the specific interests of any user and make very personal recommendations. On the other hand, since the features are hand-engineered the performance of the model can only perform as good as the selected features.

3.2 Collaborative Filtering

To overcome the disadvantages of context-based filtering, the concept of collaborative filtering was introduced where, on top of the similarity between items, the similarity between users is also used to make recommendations. Additionally, the features could be identified automatically without any manual intervention. In contrast to the earlier method, here we don't need any domain knowledge. The machine learning model can recommend something very novel to the user based on other similar users' likings and hence the user can develop new interests. But there is a major problem of cold start, where predictions are made by doing dot product (customer, product) and its embedding.

However, collaborative filtering has its own restrictions. Hybrid filtering models are able to deal with these limitations and are thus used as a competitive approach for filtering.

Recommender systems are generally perceived as a black box and a transparent intuition needs to be developed. Further, the model performance is gauged based on the dataset used and steps involved in generating those recommendations. Moreover with multiple sources of data being fed in any real time recommendation engine, there is a need to learn a good trade off between architectural complexity and robustness of recommendations.

4 Measures of Success

The success of a recommender system can be measured in multiple ways. This work only considers evaluation metrics and not business-specific ones (such as Click-Through-Rate, Conversion Percentage, etc).

4.1 Baseline

The baseline of the project is an implementation of a basic application which can suggest books to a user based on user inputs which would include (but are not limited to) the books a user has read in past, the genre of books the user typically likes to read, etc.

Some examples of evaluation metrics for recommender systems based on past research include the following-

1. Similarity Metrics
   (a) Cosine Similarity
   (b) Pearson Correlation Coefficient
2. Predictive Metrics

   (a) Root Mean Squared Error

3. Decision Support Metrics
   (a) Precision@k
   (b) Recall@k

4. Ranking Metrics
   (a) Mean Average Precision
   (b) Discounted Cumulative Gain

Individually, each metric provides a different insight into the model's performance. However, to get a broader overview, a combination of multiple metrics can be used.

4.2 Stretch

The developed application can be compared with existing recommender systems to understand the difference in performance. The next step would be fine-tuning and optimising the application, which can be done by following research papers and attempting to implement the proposed methodologies.

5 Preliminary Plan

5.1 Key Elements

The project consists of the following key components:

1. Dataset Confirmation
   (a) Finding datasets (Amazon Book Dataset, GoodReads Dataset, etc.). **[Week 1]** [1]
   (b) Comparing Datasets based on features, size, etc. to select Dataset/s. **[Week 1]**
   (c) Preprocessing the Dataset. **[Week 1]**

2. Choosing metric for comparison
   (a) Check previous papers/ articles/ models to find similar metrics in addition to already mentioned metrics,(4.1). **[Week 1]**
   (b) Understand and analyze the metrics. **[Week 1]**
   (c) Based on the dataset chosen decide on a metric or combination of metrics. **[Week 3]**

3. Baseline Model Creation
   (a) Create a basic structure, i.e. decide on an architecture which takes Dataset as input and passes it through data transformation pipelines. **[Week 2]**
   (b) Build the preliminary machine learning model based on transformed data. **[Week 3]**
   (c) Validation of model accuracy and hyperparameter tuning. **[Week 4]**

4. Stretch
   (a) Check existing recommender system models and their research papers. **[Week 4]**
   (b) Compare the baseline model with other models. **[Week 5]**
   (c) Try to improve and modify the model by taking inspiration from the state-of-the-art models. **[Week 5 and 6]**
   (d) Analyze the added components and try to improve them to achieve better results. **[Week 7]** [2]

5.2 Division of Work

1. Abichal - 1a, 1b, 1c, 3a, 3b, 4a, 4c, 4d
2. Asad - 1a, 1b, 1c, 3a, 3b, 4a, 4c, 4d
3. Nikhil - 1a, 1b, 1c, 3a, 3b, 4a, 4c, 4d
4. Meesum - 2a, 2b, 2c, 3a, 3c, 4b, 4c, 4d
5. Soham - 2a, 2b, 2c, 3a, 3c, 4b, 4c, 4d

[1] Week 1 starts from September 26

[2] Presentation on November 14, end of Week 7