Project Report

For my project, I wanted to create a content-based filtering recommendation system by looking at a dataset of book ratings. The dataset I used was found on Kaggle at: (https://www.kaggle.com/datasets/ruchi798/bookcrossing-dataset). To begin with, I looked at the structure of the dataset, looking at the columns present and the values that existed within specific columns. I also looked at the value counts of the columns to better understand the distribution present in the data.

I used the content-based filtering code from the notebook Week 2.1 as a starting point for creating my recommender system and was able to run through to predict 10 books based on a suggested book. The system was made using Pytorch. The results of this prediction function were unsophisticated due to the simplistic nature of the way I was handling the data, so the next step was to look at different ways of predicting content and then also see how I could aggregate features to produce more advanced results.

The recommendation system I was using made a vocab for the specific feature I wanted to extract and then pulled out a one-hot encoding layer that could be used to find the cosine similarity of each book. This could then be used to create a prediction function that would take in a book and output 10 new books.

To begin with, I looked at the category column, which contains the genres for the books. The dataset was partially unlabelled with books in category ‘9’ making up 406102 rows of the 1031175, almost 40% of the data. Because my laptop was struggling to move through a dataset this large, I decided to remove the values that contained ‘9’ in the category column as I did not think they would add much to my interpretations gained from the exploration of the dataset.

For the sake of processing, I reduced the dataset down to 10,000 rows so that I could process the data more efficiently whilst not losing too many examples. I made sure to do this in a way that maintained the distribution of the feature I was looking at.

The category features, even without the category ‘9’ were still unnuanced, with the most common being, ‘Fiction’, ‘Juvenile Fiction’, and ‘Autobiography and biography’. Not only did ‘Fiction’ make up the majority of the dataset but it is a very vague category that does not lend itself to further investigation of the differences between books within the category fiction.

A way to have worked around this might have been to use natural language processing on the book descriptions to be able to find books that contained similar descriptions but for the sake of investigating the data at a larger scale, I settled on focusing my energy on the categories as they existed already. As previously mentioned, the results were very simplistic and did not take into account the ratings.

Next I looked at how I could use the ratings to produce recommendations that were most similar. I did this by getting the average rating of every book along with the number of times it was rated to produce a more nuanced rating. This allowed me to weigh the importance of the reviews given to each book, which in turn gives more trustworthy recommendations. Yet again whilst looking at the data I was presented with a place to remove useless data. The majority of the books were rated 0, an artifact of the data collection process, I think, rather than users actually going out of their way to rate a book 0. So I decided to remove the rating of 0. I was not too worried about how this might affect the distribution of the data because of the suspected data collection issues.

With the new system, the ratings that came back from the prediction function were much higher than the original recommendations, which reflects the difference in how the ratings were recorded.

From there I decided to look at how the country of the user might affect the books they are recommended. The location column was fairly specific in the dataset so first I decided to look at the country. This was to ensure that I had enough examples from my reduced dataset for the recommendation system to pull from. I was able to make successful predictions based on the country whilst retaining the weighted ratings, which resulted in books that were predicted to be from the same country. I realized the error in assuming that the country of the book represented the most common country the book was read in so I decided to make a column in the dataset that represented the most common country. This doesn’t necessarily mean it’s the most popular book depending on the region but I was able to train a recommendation system on this ‘most common country’ data which allowed me to get more tailored results.

After exploring these avenues, I decided to pull back and see if I could work with a different method to produce potentially more nuanced results. I decided to look at making a trained model that could then produce embeddings. I could then work out the cosine similarity of the embeddings and produce recommendations from there. Instead of looking at the category I decided to explore using the ISBN (the international standard book number), which is used by publishers and denotes one title from a specific publisher. It is used to handle metadata around a book more easily. Essentially the ISBN number can be used as a place holder the name of the book but allows me to be more specific in the event that the title has been input incorrectly.

Using the embeddings, I was able to recommend books based on one book but also recommend books based on a user. This presented more flexibility in producing results. There were issues that I faced with this system, primarily the predictions returning multiples of the same books, and the simplicity of the data meant that recommendations were still fairly unnuanced but allowed me to see how I could get different results depending on the system.

A lot of the issues that I experienced whilst exploring the data were mainly the limitations of the categories, by not having more nuance in the categories column I was unable to produce better more usable results for a recommendation system.

To further explore this data, I think that I would, like previously mentioned, use NLP to process the descriptions that would allow me to have more data to work with about each book rather than some of the superficial features like ‘category’ or ‘country’ which gave very surface level results. Another would be to look at how I could make recommendations that find books that are similar in every way but one feature and then recommend that as a way of creating more novelty in the system. My focus with that idea would be on the country of origin, where I could then recommend translated books.

Translations are often less popular due to a multitude of factors but by looking at these features I could produce recommendations that are interesting and valuable to a user whilst also introducing them to new books. Overall, this project allowed me to better sift through this dataset and understand how changing the features that are input into the prediction function could produce different results.