**Library Book Recommendation System**

Project Report submitted for

**Artificial Intelligence (UC411)**



By

Rishabh Goyal Roll Number: 102103585

Hitesh Aggarwal Roll Number: 102103596

Aaryan Sood Roll Number: 102103574

Ishaan Sharma Roll Number: 102103583

From batch 2CO21

**Introduction**

Most online products such as video or music streaming services, e-commerce websites that rely on engagement to earn revenue utilise recommendation systems to predict what their customers like and retain their user base.

We as customers, or as developers, are constantly exposed to recommendation systems, and they have become an integral part of our daily lives. As such we were interested in how they worked and have come up with an intelligent agent that recommends a library member more books to read from the library’s collection based on the book that they are currently checking-out for issuing.

The agent will expect books to be issued as input, and give recommendations for future orders after checkout. The criteria for the recommendations the present book’s content, i.e., comparisons will be based on book summaries.

Link to GitHub repository: <https://github.com/Hitesh-Aggarwal/Librain>

**Literature Survey**

**What is a recommendation system?**

Loosely, a recommendation system works by comparing user input against the entire dataset, giving each item a score, and returning the items with the highest score. The comparison can work using any number or kinds of features.

**There are two approaches to building a recommendation system:**

•**Exploitation:** The system chooses documents similar to those for which the user has already expressed a preference.

•**Exploration:** The system chooses documents where the user profile does not provide evidence to predict the user’s reaction.

These two approaches can also be used simultaneously, allowing users to experience something they never would have in their comfort zone.

**Exploitative recommendation systems largely belong to two categories:**

**Collaborative filtering:** Collaborative filtering is the process of predicting the interests of a user by identifying preferences and information from many users. This is done by filtering data for information or patterns using techniques involving collaboration among multiple agents, data sources, etc.

•**Content-based filtering:** Content based systems generate recommendations based on the users preferences and profile. They try to match users to items which they’ve liked previously. The level of similarity between items is generally established based on attributes of items liked by the user.

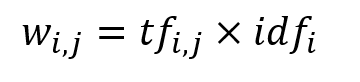
•We choose the content-based filtering approach since collaborative filtering requires extensive data from multiple users to give relevant suggestions.

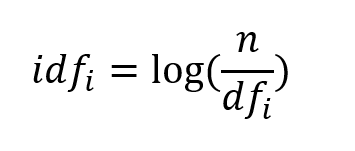
**In order to make our model work, we must clean our data and encode textual entries:**

It is important to note that when working with textual data the first thing to do is to convert this text into **numbers**. This is essential because most machine learning algorithms perform mathematical operations on data to provide a result. We found two of such approaches to make this conversion:

•Character encoding through the Bidirectional Transformers for Language Understanding algorithm that involves natural language processing

•Giving a string a TF-IDF score on the basis of the words it comprises of. Its mathematical formula is as follows:



Here the term w represents TF-IDF score for each word in a particular entry, that is a product of term frequency and inverse document frequency. The inverse document frequency is defined as: 

**Methodology**

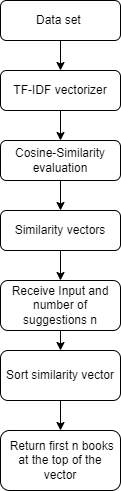
**Dataset**

Our training data is contained in the data.csv. It contains a list of the books available at the library along with brief summaries encapsulating the book’s content. Since most machine learning algorithms involve mathematical computations to produce results, we first clean our data and encode it so that the model is easy to understand and work with. The encoding is done by evaluating each book’s summary text into matrices that tally words against their TF-IDF scores for each entry in the dataset. Such a dataset gives weight to words that are rarely used in documents/summaries and thus are more likely to distinctly represent the content of a novel and largely ignores words like ‘are’, ‘and’ etc that frequently occur across documents but do not provide any value to the algorithm.

**Algorithm**

After the TF-IDF vectorizer processes the textual data, we compare each book with the rest and evaluate a similarity index in each case. These indices are stored in similarity vectors that each correspond to distinct books. The similarity vector is filled using cosine-similarity. The features used to plot books in space are the words for which the TF-IDF scores have been compiled. A mapping is created between the title of a book and its index in the array that contains the similarity vectors so that they can be referred to easily. When the model receives an input (the name of the book that the user checks-out, and the number of suggestions they desire, n), its similarity vector is fetched and sorted in descending order so that the most similar books remain in the beginning of the vector. The model returns n books at the top of the vector.

**Flowchart**



**Future Scope**

This model can be further refined by using aspects of collaborative filtering recommendation-systems. For example, using book ratings from a platform like Goodreads that has a very active user base, or recommending books that friends (exploration recommendations).The model may also consider features like authors, publishing date, publishers, genres and so. A recommendation system that takes the above features into consideration can provide relevant recommendations and introduce book lovers to titles that they would absolutely adore but would otherwise not have the opportunity to read. It can provide a smooth experience for library members or boost sales at bookstores.

**Result**

Thus our proof-of-concept content based book recommendation system provides suggestions to users based on a book they have previously liked, read, or checked out.