Anjuman-I-Islam's

M.H. Saboo Siddik College of Engineering

M.H. Saboo Siddik Polytechnic Road, Byculla, Mumbai-400 008

Department of Electronics Engineering

A mini project on

Book Recommendation Engine Based On Collaborative Filtering Technique.

Prepared by

TAJ MOHAMMED SHAH (4119006)

ANSARI MOHAMMED HAMZA SHAKEEL AHMED(4119007)

ADVAIT GURUNATH CHAVAN (4119008)

AAKASH NANHKURAM GUPTA (4119009)

Under the guidance of

Prof. Jillani Sayyed

Under the academic year **2021-22**

Book Recommendation Engine Based On Collaborative Filtering Technique.

\

ABSTRACT

One of the effective personalization technologies powering the adaptive web is collaborative filtering. Collaborative filtering (CF) is the manner of filtering or comparing objects via the critiques of different people. CF technology brings collectively the reviews of massive interconnected communities at the web, assisting filtering of widespread portions of statistics. in this project we introduce the middle concepts of collaborative filtering, its primary makes use of for users of the adaptive internet, the theory and practice of CF algorithms, and layout decisions regarding rating systems and acquisition of rankings. We additionally talk the way to evaluate CF structures, and the evolution of rich interaction interfaces. Thus building a recommendation systems by the technique of Collaborative Filtering is a good way to show to the user contents in a personalized manner.

Keywords - Collaborative Filtering, Recommendation Systems, Personalization.

INTRODUCTION

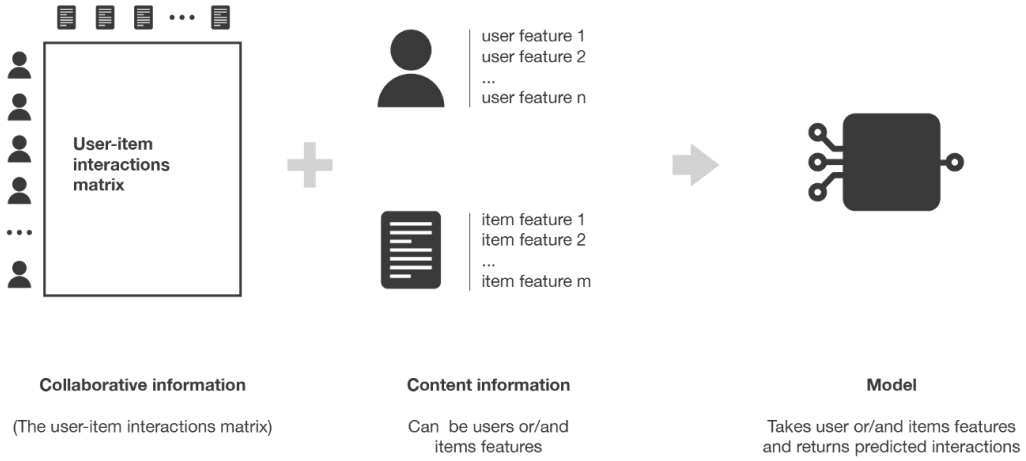
A recommendation system is a subclass of facts filtering systems that seeks to expect the rating or the choice a user would possibly deliver to an item. In simple phrases, it's miles an set of rules that shows applicable objects to customers. Eg: within the case of Netflix which movie to observe, in the case of e-trade which product to buy, or within the case of kindle which e book to examine, and so forth.. Recommender systems are really critical in some industries as they can generate a huge amount of income when they are efficient or also be a way to stand out significantly from competitors. As a proof of the importance of recommender systems, we can mention that, a few years ago, Netflix organised a challenges (the “Netflix prize”) where the goal was to produce a recommender system that performs better than its own algorithm with a prize of 1 million dollars to win.

Recommender systems can be roughly divided into two types based on the technoque

Content based Recommender systems:

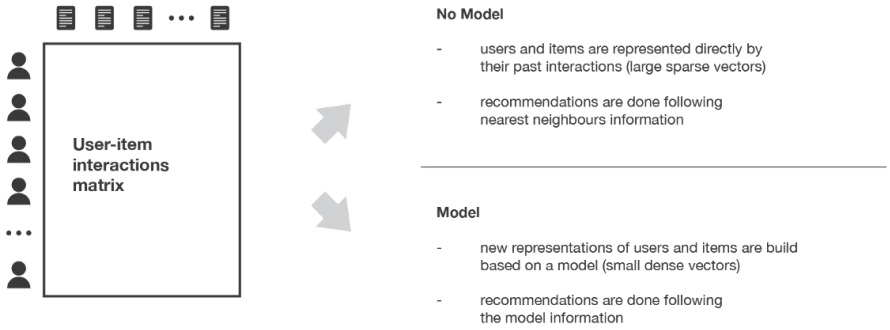
Then, the idea of content based methods is to try to build a model, based on the available “features”, that explain the observed user-item interactions. Still considering users and movies, we will try, for example, to model the fact that young women tend to rate better some movies, that young men tend to rate better some other movies and so on. If we manage to get such model, then, making new predictions for a user is pretty easy: we just need to look at the profile (age, sex, …) of this user and, based on this information, to determine relevant movies to

suggest



Collaborative filtering Recommender Systems:

Collaborative methods for recommender systems are methods that are based solely on the past interactions recorded between users and items in order to produce new recommendations. These interactions are stored in the so-called "user-item interactions matrix".

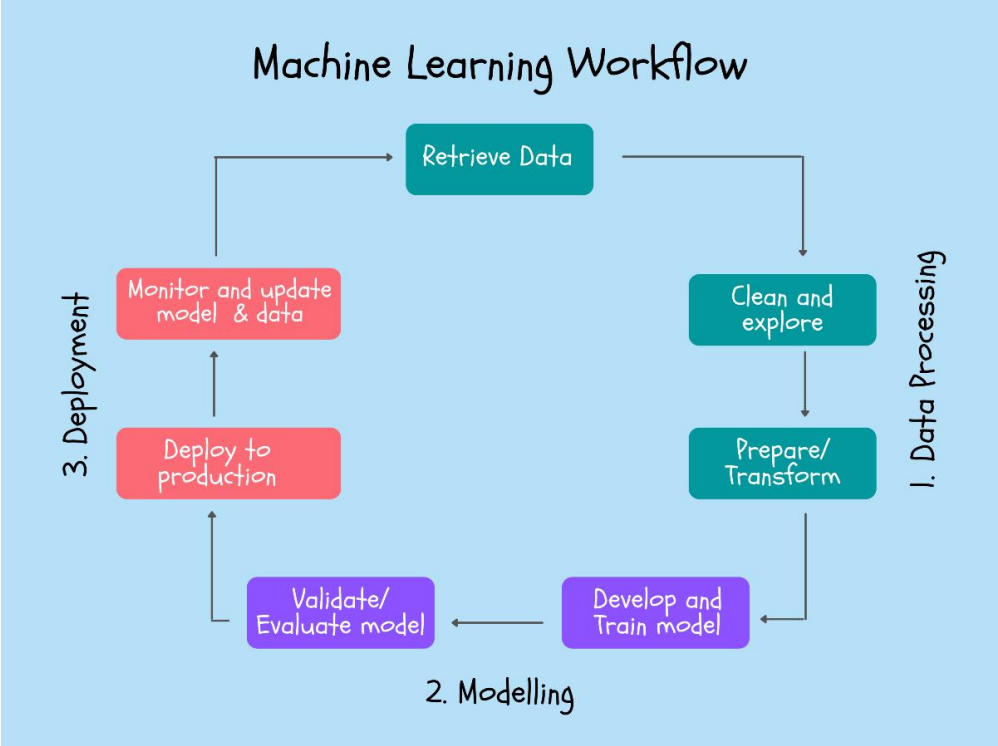


For our project we have opted for the collaborative filtering approach for building a Machine Learning model.

PROPOSED WORK :

Just like any other machine learning project recommendation system also follows the machine learning workflow.Machine learning workflows define which phases are implemented during a machine learning project. The typical phases include data collection, data pre-processing, building datasets, model training and refinement, evaluation, and deployment to production. You can automate some aspects of the machine learning operations workflow, such as model and feature selection phases, but not all.

The work folow can be given with the figure shown below.



Step 1 : Gathering the data.
For the purpose of building a book recommender system the data is gathered from the following website in the form of a csv file.

Dataset link - http://www2.informatik.uni-freiburg.de/~cziegler/BX/

Dataset name - Book Crossing Dataset.

BX-Users
Contains the users. Note that user IDs (`*User-ID*`) have been anonymized and map to integers. Demographic data is provided (`*Location*`, `*Age*`) if available. Otherwise, these fields contain *NULL*-value

BX-Books
Books are identified by their respective ISBN. Invalid ISBNs have already been removed from the dataset. Moreover, some content-based information is given (`*Book-Title*`, `*Book-Author*`, `*Year-Of-Publication*`, `*Publisher*`), obtained from Amazon Web Services. Note that in case of several authors, only the first is provided. URLs linking to cover images are also given, appearing in three different flavours (`*Image-URL-S*`, `*Image-URL-M*`, `*Image-URL-L*`), i.e., small, medium, large. These URLs point to the Amazon web sit

BX-Book-Ratings
Contains the book rating information. Ratings (`*Book-Rating*`) are either explicit, expressed on a scale from 1-10 (higher values denoting higher appreciation), or implicit, expressed by 0.

Step 2 : Data Preprocessing.

Data Preprocessing includes the steps we need to follow to transform or encode data so that it may be easily parsed by the machine. For this project we were required to do just the following things.

Selecting only those columns which were required.

- Deleting the duplicate values.
- Merging all the dataset to one.
- Creating a pivot table of the merged dataset.
- Calculating sparse matrix for the pivot table.

Step 3 : Model Building.

For our model we'll use NearestNeighbors algorithm for calculating the distance of each book with every other book when the user enters a new book and for implementing the model we'll use sklearn library which has great implementation of different Machine Learning algorithm.

1. Import the NearestNeighbors class form the sklearn library.

```
from sklearn.neighbors import NearestNeighbors
model = NearestNeighbors(algorithm='brute')
```

2. Fit the model with the sparse matrix which we have already calculated. This will calculate the distance of each book from the pivot table with every other book.

```
model.fit(sparse_matrix)
```

3. The class NearestNeighbors in sklearn returns distances as well as the neighbors of the specified book.

```
#claculating distance of each book with every other books
distances,suggestions=model.kneighbors(rating_pivot_table.iloc[0,:].values.reshape(1,-1),n_neighbors=6)
```

Thus, the suggestions is the array of the books which will be recommended to the user once the user has entered the the book name.

Here , our model is complete. Now to make it more interesting we'll convert this model to a webpage where if a user enters a book name then they will be recommended books based on that

Step 4 : Pickling the model.

To save the state of a model in a varible so that we could use it in later stages of the project without having to run the model all from the beginning, that is when pickling comes in play. To pickle the model simply import the pickle module from python and do the following.

```
import pickle

pickle.dump(rating_pivot_table,open("rating_table.pkl","wb"))

pickle.dump(books_image_data,open("books_image_data.pkl","wb"))
```

Step 5 : Converting the model into Webpage.
For building a webpage we will python library called streamlit which makes it easy for building the webpage.

1. Open vscode and create a virtual enviroment for our project. Type the following into the vscode terminal - virtualenv myenv.

2. Import the required libraries.

```
import pandas as pd
import numpy as np
import pickle
from scipy.sparse import csr_matrix
from sklearn.neighbors import NearestNeighbors
```

3. Create a functon for recommending books.

```
#Function for recommending books
def recommend(book_name):
  recommended_books = []
  image_url = []
  book_index = np.where(rating_table.index==book_name)[0][0]
  distances , suggestions = model.kneighbors(rating_table.iloc[book_index,:].values.reshape(1,-1),n_neighbors=5)
  suggestions = np.ravel(suggestions, order='C') #2d to 1d array
  for i in suggestions:
    recommended_books.append(rating_table.index[i])

  for i in recommended_books:
    image_url.append(books_image_data[books_image_data["title"] == i ].image.to_string(index=False))

  return recommended_books,image_url
```

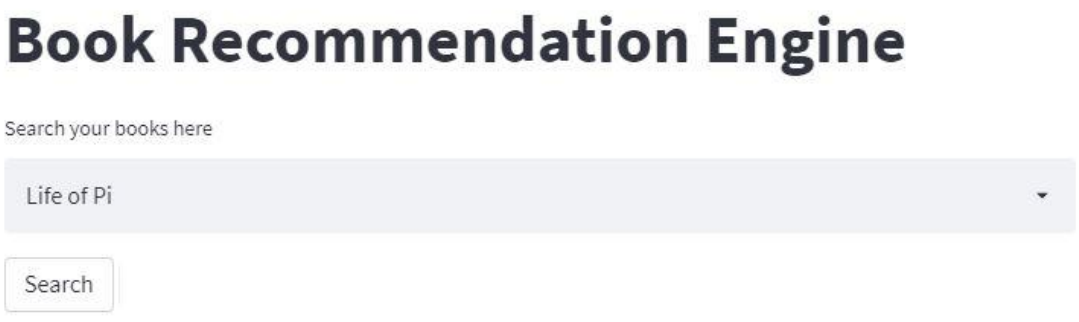
4. Create a frontend using streamlit[follow the documentation on the streamlit website].

```
import streamlit as st

st.title("Book Recommendation Engine")

selected_book = st.selectbox(
    'Search your books here',
    books_name)

if st.button('Search'):
    books,images = recommend(selected_book)

    container1 =st.container()
    container1.write("You have searched for : ")
    container1.header(books[0])
    container1.image(images[0])

    col1, col2, col3,col4 = st.columns(4)

    with col1:
        st.subheader(books[1])
        st.image(images[1])
    with col2:
        st.subheader(books[2])
        st.image(images[2])
    with col3:
        st.subheader(books[3])
        st.image(images[3])
    with col4:
        st.subheader(books[4])
        st.image(images[4])
```

5. Run the app using the command - streamlit run main..py.

WORKING OF WEBSITE :

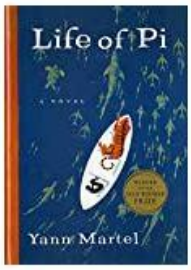
- Type name name of book you want to search.



- The website will show you result of what you searched and also those books which was liked by other users.

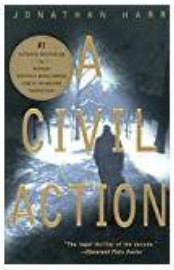
YOU HAVE SEARCHED FOR -

Life of Pi



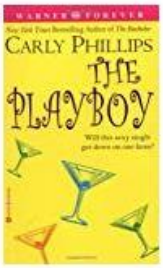
PEOPLE ALSO LIKED -

A Civil Action

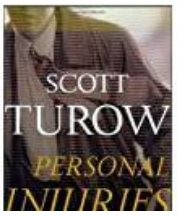


The Little Friend

The Playboy



Personal Injuries



GITHUB LINK FOR PROJECT :
https://github.com/Vidrow/Book-Recommendation-Engine

REFERENCES :

https://www.researchgate.net/publication/200121027_Collaborative_Filtering_Recommender_Systems

https://www.hindawi.com/journals/misy/2019/3560968/

https://iopscience.iop.org/article/10.1088/1742-6596/1437/1/012095/pdf

https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-recommendation-engine-python/

https://medium.com/fnplus/everything-you-need-to-know-about-recommendation-systems-348cdfccbaf0