Minor Project Final Progress Report

(2022, group 22)

BOOK RECOMMENDATION ENGINE

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

In

Computer Science and Engineering

Submitted By ;-

Konark Lohat (1905801)

Gautam Jain (1905785)

Jaskamal Singh (1906006)

Guru Nanak Dev Engineering College Ludhiana - 141006

1. INTRODUCTION

A recommendation engine based on dynamic content, further classified or categorized into neighborhood-based collaborative learning (milestone 1) using KNN (cosine similarity) algorithm and content-based learning (milestone 2) to be used in the recommendation system.

A Book Recommendation Engine Based on Collaborative Filtering Concept using K Nearest Neighbors algorithm (Cosine Similarity) to recommend books based on the books liked by other readers and their corresponding ratings (ranking system).

Based on the reading provided by all the readers, books are recommended to the users based on the users they follow and their liked books.

The Collaborative Filtering System is established using the cosine similarity relation between ratings and the recommendations with maximum match are returned.

1.1 Objectives

- Implementing neighborhood based collaborative filtering (KNN Cosine similarity).
- Creating a framework for recommendation engine/system.
- Implementing Content-based filtering.

1.2 Technology used

1. **Python** – Python is an interpreted general-purpose high-level language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

2. **ML/AI –** Machine learning is a subfield of artificial intelligence (AI). The goal of machine learning generally is to understand the structure of data and fit that data into models that can be understood and utilized by people.

3. **Git** – Git is an open-source version control system. It was designed and developed by Linus Torvalds (creator of the Linux kernel) and is the most popular version control system to date.

2. SYSTEM REQUIREMENTS

2.1 Hardware Requirements

- Windows 10+ / Linux / Mac Operating System
- i5 8 Gen+
- 4+ GB RAM
- Dual Core processor.

2.2 Software Requirements

- Python 3.6+ installed
- Python Packages (like NumPy, pandas, matplotlib etc.)
- Seaborn

3. SOFTWARE REQUIREMENT ANALYSIS

Most of the recommendation systems used nowadays in big companies/products are not open source. Pushing these technologies like a recommendation engine to be an open-source product creates more horizons for beginners in ML/AI and also to make optimized and accurate recommendations.

This recommendation engine can be used by system architectures to make fast and reliable integration of AI in their application/system. To create a high open-source impact, the reliable scalability design of the recommendation engine has been given proper consideration.

Recommender systems help the users to get personalized recommendations, make correct decisions in their online transactions, increase sales and redefine the user's web browsing experience, retain the customers, enhance their shopping experience, etc. The information overload problem is solved by search engines, but they do not provide the personalization of data.

Recommendation engines provide personalization. There are different types of recommender systems such as content-based, collaborative filtering, hybrid recommender system, demographic, and keyword-based recommender systems. A variety of algorithms are used by various researchers in each type of recommendation system.

\

4. LITERATURE

1. Google Developers Community [1]: Recommendation System by Google Developers Platforms

- Proposed Learnings:
  - Describe the purpose of recommendation systems.
  - Understand the components of a recommendation system including candidate generation, scoring, and re-ranking.
  - Develop a deeper technical understanding of common techniques used in candidate generation.
  - Use TensorFlow to develop models used for recommendation.

2. Joeran Beel et al. [2]: Research-paper recommender systems: a literature survey

- This paper has introduced recommender systems to new research. This paper has also identified key problems which need research in recommender systems.
- This paper can help Ph.D. and Master's students in choosing their area of research. The research gap is already presented in this paper to form different problems of recommender systems.
- The recommendation system finds its utility in major areas of web Applications. As these problems get solved more and more useful recommendation systems will become.

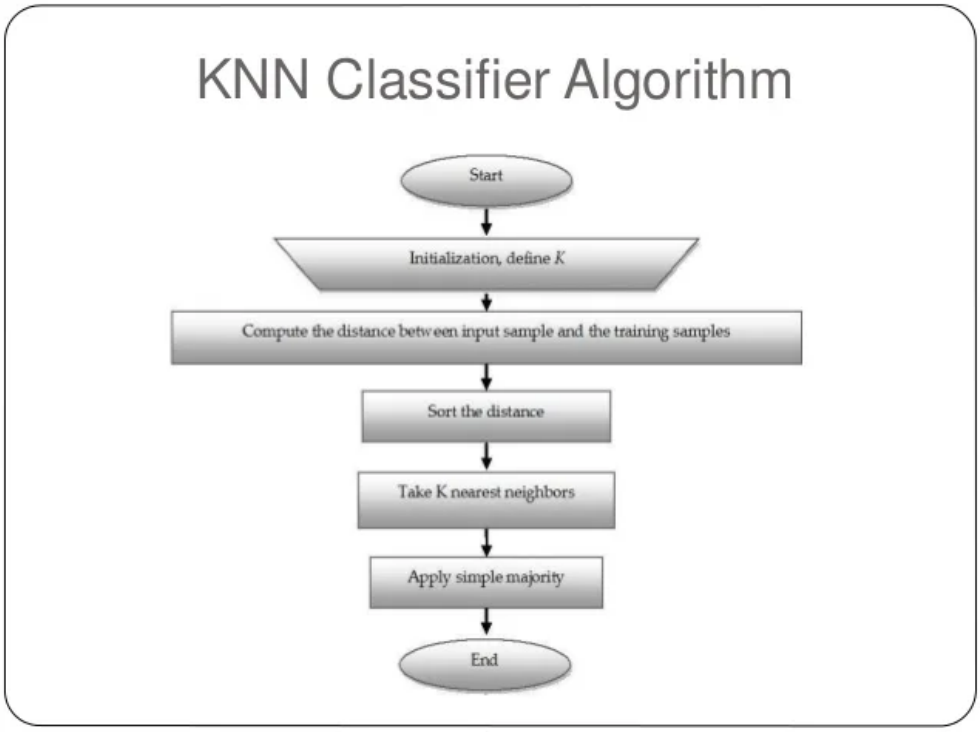
3. Nitin Mishra et al. [3]: Research Problems in Recommender systems

- It is a system that helps users to choose items that they may need. A Good Recommender System saves user's time and keeps the user engaged in the system resulting in higher revenue.
  - Collaborative Filtering
  - Content-based
  - Hybrid Recommendation System

5. SOFTWARE DESIGN

5.1 KNN Model Features

- KNN is a Supervised Learning algorithm that uses labeled input data set to predict the output of the data points.
- It is one of the most simple Machine learning algorithms and it can be easily implemented for a varied set of problems.
- It is mainly based on feature similarity. KNN checks how similar a data point is to its neighbor and classifies the data point into the class it is most similar to.



5.2 EDA and data preparation

- First we create, and choose appropriate features and data from the the dataset in which we only include the users from USA and Canada for better performance and quick tests.
- We merge the data from the three datasets (users, books, ratings) into one matrix where book title, usersID, rating of each and every user for the particular book (book title).
- Then we create a sparse matrix with the help of scipy package in order to fit the model, in which we have the y-axis as the book title and x-axis as the userID.
- From the rating dataset we merge the userID, and book title with ISBN as the primary key and then to rotate the matrix in anti-clockwise direction, we pivot the matrix around the index as ‘bookTitle’, column = ‘userID’ and the values being ‘bookRating’.

BookTitle	**UID-1**	**UID-2**	**UID-3**	**UID-4**
Harry potter	0	9	4	0
Jenkins	10	0	5	9
Ghost	4	6	0	2

5.3 KNN Model Building with Cosine Similarity metric.

- Then we apply KNN (cosine similarity as the metric) because of it’s flexibility in recommending the users with similarly aligned interest irrespective of their magnitude.
- The selected book will have it’s own vector formed from the pivoted sparse matrix, and the cosine angle will be calculated with each and

every entry in the sparse matrix from which we devise their respective cosine distance.

- Top 5 results where the cosine distance is found to be the minimum are selected for the recommendations.

5.4 Pearson Correlation

5.5 EDA for TF IDF

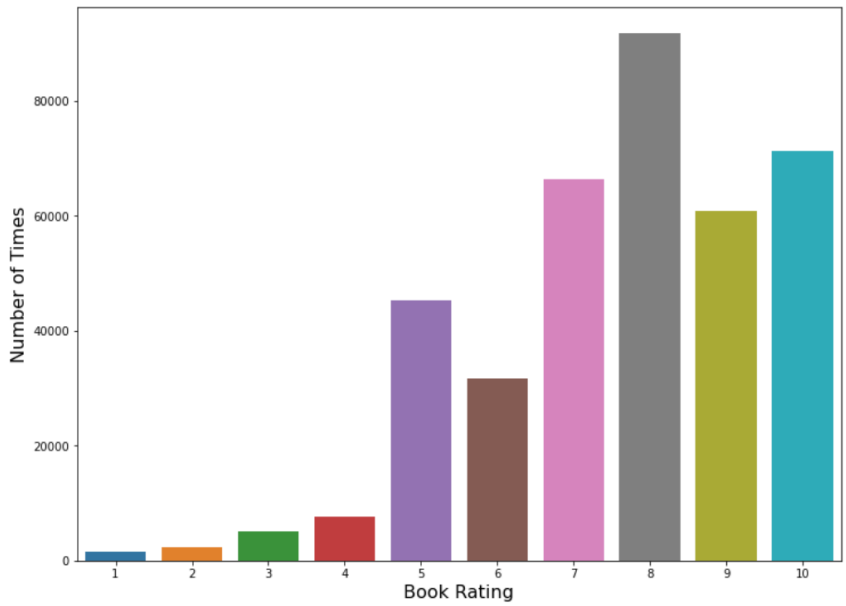
5.6 TF IDF

6. CODING / CORE MODULE

6.1 Exploratory Data Analysis

The following observations and conclusions have been made through various analysis and visualizations performed in this project:

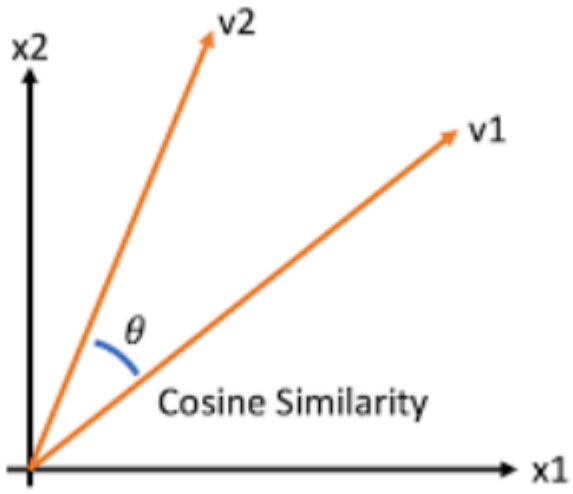
- The top 3 most ranked authors are Stephen King, Nora Roberts and John Grisham.
- The most popular novels/books are The Secret Life of Bees, The Da Vinci Code and The Lovely Bones.
- The Year in which the highest number of books have been published is 2002 and the Publisher with the highest number of publications is "Ballantine Books".
- The top 5 countries where most of the users who are indulge in reading are USA, Canada, germany, united kingdom and spain.



A countplot of bookRating indicates that higher ratings are more common amongst users and rating 8 has been rated highest number of times.

6.2 KNN - Cosine Similarity

Cosine similarity is a metric used to determine how similar the documents are irrespective of their size. Cosine similarity among two objects measures the angle of cosine between the two objects. It compares two documents on a normalized scale. It can be done by finding the dot product between the two identities.



(Fig. 1) Graph representation for cosine similarity between two vectors, v1 and v2

As the above diagram shows, the angle between v1 and v2 is. The Less the angle between the two vectors the more is the similarity. It means if the angle between two vectors is small, they are almost alike each other and if the angle between the two vectors is large then the vectors are very different from each other.

$$\cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \cdot \|\vec{b}\|}$$

(Fig. 2) Cosine similarity mathematical formula

6.3 Collaborative Filtering

It depends upon the users who have similar interests and gives the result based on all the users.

User-based: In user-based collaborative filtering, it is considered that a user will like the items that are liked by other users with whom they have comparable taste.

Item-based: Item-based collaborative-filtering is different, it expects the users to like items that are related to items that he has liked earlier.

6.4. Why cosine similarity metric?

The cosine similarity is beneficial because even if the two similar data objects are far apart by the Euclidean distance because of the size, they could still have a smaller angle between them. Smaller the angle, higher the similarity.

When plotted on a multi-dimensional space, the cosine similarity captures the orientation (the angle) of the data objects and not the magnitude.

6. PERFORMANCE / OUTPUT

- After EDA, to improve the performance of the model on local systems, users from USA and Canada only were taken for the recommendations.

In [78]:
```
combined = rating_popular_book.merge(users, left_on = 'userID', right_on = 'userID', how = 'left')

us_canada_user_rating = combined[combined['Location'].str.contains('usa|canada')]
us_canada_user_rating = us_canada_user_rating.drop('Age', axis = 1)
us_canada_user_rating.head()
```

Out[78]:

	userID	ISBN	bookRating	bookTitle	totalRatingCount	Location
0	276725	034545104X	0	Flesh Tones: A Novel	60	tyler, texas, usa
1	2313	034545104X	5	Flesh Tones: A Novel	60	cincinnati, ohio, usa
2	6543	034545104X	0	Flesh Tones: A Novel	60	strafford, missouri, usa
3	8680	034545104X	5	Flesh Tones: A Novel	60	st. charles county, missouri, usa
4	10314	034545104X	9	Flesh Tones: A Novel	60	beaverton, oregon, usa

- Randomly selecting a book as a query for which similar books are recommended.

In [59]:
```
us_canada_user_rating_pivot.index[query_index]
```

Out[59]:
```
'Witness in Death (Eve Dallas Mysteries (Paperback))'
```

- With respect to the above query cosine distance is calculated and instances (books) with the least distance is concluded and decided to be recommended to the users who have shown interest in the original queried book.

In [80]:

```
for i in range(0, len(distances.flatten())):
    if i == 0:
        print("Recommendations for {0}:\n".format(us_canada_user_rating_pivot.index[query_index]))
    else:
        print("{0}: {1}, with distance of {2}".format(i, us_canada_user_rating_pivot.index[indices.flatten()[i]], distances.flatten()[i]))
```

```
Recommendations for Witness in Death (Eve Dallas Mysteries (Paperback)):

1: Seduction in Death, with distance of 0.4620604265166146
2: Judgment in Death, with distance of 0.5190443536196638
3: Reunion in Death, with distance of 0.5239857027197188
4: Purity in Death, with distance of 0.5505452214328186
5: Castles, with distance of 0.6118648730329626
```

6.5 Pearson Correlation

6.6 TF IDF

7. REFERENCES

[1] Google Developers Community (2020). Recommendation System by Google developers platforms. Link - https://developers.google.com/machine-learning/recommendation

[2] Joeran Beel, Bela Gipp, Stefan Langar, Corinna, Breitinger et al., 2016. Research-paper recommender systems: a literature survey. Journal on Digital Libraries. Link - https://d-nb.info/1147681678/3442-6596/1717/1/012002/pdf

[3] Nitin Mishra, Saumya Chaturvedi, Aanchal Vij, Sunita Tripathi et al., 2020. Research Problems in Recommender systems - Journal of Physics: Conference Series. Link - https://iopscience.iop.org/article/10.1088/1742-6596/1717/1/012002/pdf