Recommendation System with Deep Learning

Developing a Personalized Book Recommendation System

Affiliation: Hamm-Lippstadt University of Applied Sciences
Author: Sayed Galib Hossen Rizvi
Email: sayed-galib-hossen-rizvi@stud.hshl.de
Matriculation number: 2200588

Abstract

The presentation addresses the problem of information overload and the increasing relevance of recommendation systems.

Explains the use of Neural Collaborative Filtering (NCF) to integrate collaborative and content-based filtering.

Focuses on leveraging deep learning techniques to develop a personalized book recommendation system.

https://medium.com/@xaradxarma/book-recommendation-system-8cdb77585b65

Introduction

Importance of Recommendation Systems:

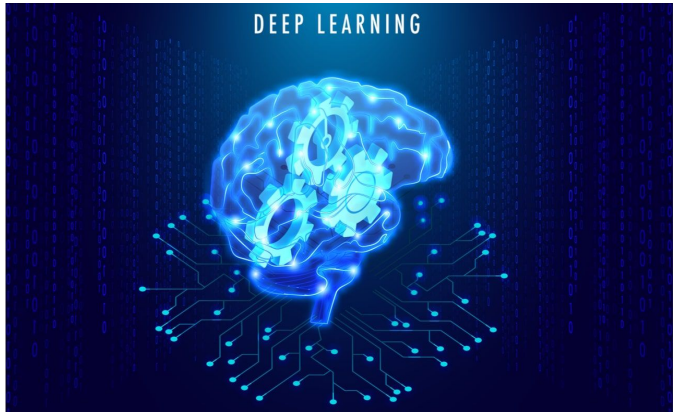
- Widely used in e-commerce, social media, and entertainment platforms.
- Improve user experience by providing personalized suggestions.

Challenges of Traditional Methods:

- Cold-start problem (new users/items lack historical data).
- Data sparsity in large-scale datasets.

Why Deep Learning?:

- Extracts latent features directly from data.
- Handles large-scale, complex interactions.
- Offers scalable solutions with improved accuracy.



https://online.york.ac.uk/the-next-step-in-machine-learning-deep-learning/

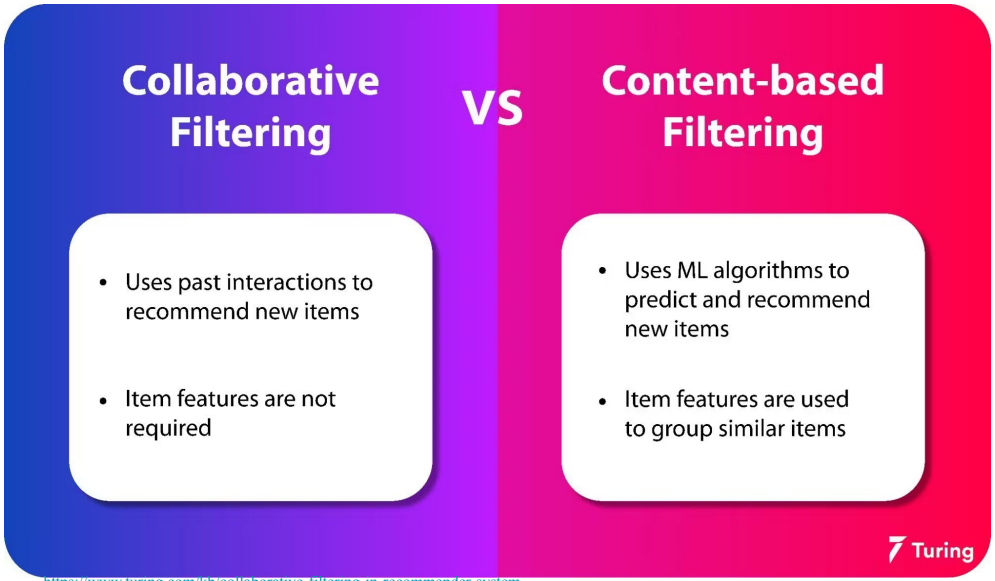
Objective

Develop a deep learning-based book recommendation system that combines

Collaborative Filtering

Content-Based Metadata

Overcome traditional limitations to deliver accurate and diverse recommendations.



https://www.turing.com/kb/collaborative-filtering-in-recommender-system

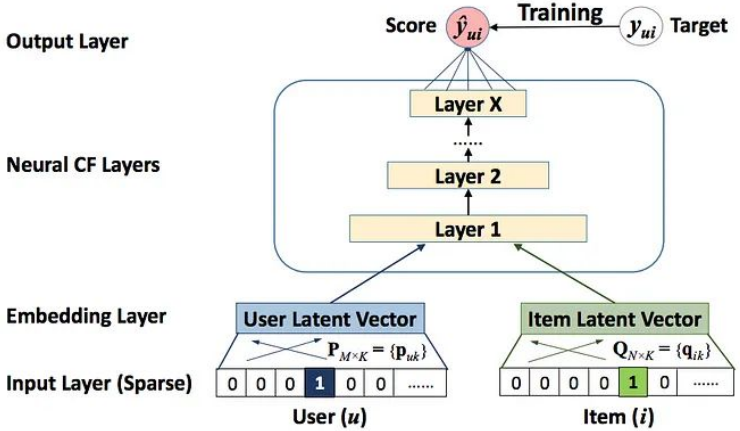
Methodology

Neural Collaborative Filtering (NCF):

- Embedding layers for user and book features
- Dense layers to model non-linear relationships
- Predict user preferences by learning interactions between embeddings.

Advantages

- Scalable for large datasets
- Learns complex patterns for personalized recommendations.



https://towardsdatascience.com/neural-collaborative-filtering-96cef1009401

Dataset and Preprocessing

Dataset Features

Book details: Titles, authors, ratings, pages, and metadata.

User interactions with books.

Preprocessing Steps

Data Cleaning: Remove missing values.

Encoding: Convert categorical data into numerical values.

Normalization: Scale numeric fields for uniform impact.

Splitting: Divide data into training, validation, and test sets.

books.csv (1.56 MB)

Detail | Compact | Column | 10 of 12 columns

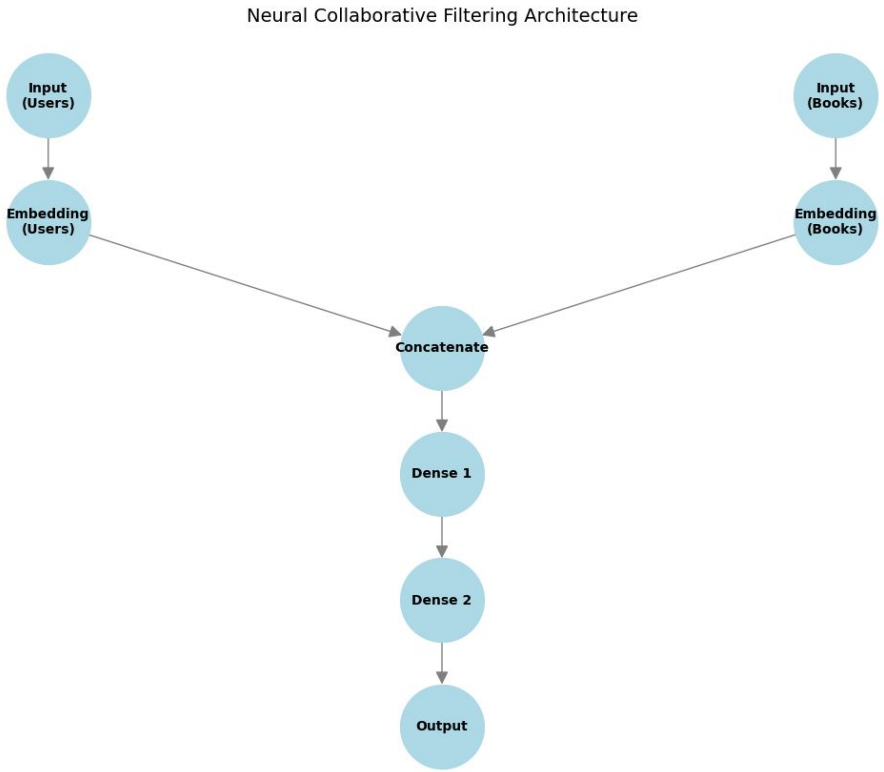
About this file

This file contains the detailed information about the books, primarily. Detailed description for each column can be found alongside.

bookID	title	authors	average_rating	isbn	isbn13
A unique Identification number for each book.	The name under which the book was published.	Names of the authors of the book. Multiple authors are delimited with -.	The average rating of the book received in total.	Another unique number to identify the book, the International Standard Book Number.	A 13-digit identify th of the sta ISBN.
1 45.6k	10352 unique values	6643 unique values	0 5	11126 unique values	675m
1	Harry Potter and the Half-Blood Prince (Harry Potter #6)	J.K. Rowling/Mary GrandPré	4.57	0439785960	97804397
2	Harry Potter and the Order of the Phoenix (Harry Potter #5)	J.K. Rowling/Mary GrandPré	4.49	0439358078	97804393
4	Harry Potter and the Chamber of Secrets (Harry Potter #2)	J.K. Rowling	4.42	0439554896	97804395

https://www.kaggle.com/datasets/jealousleopard/goodreadsbooks

NCF Architecture



NCF Architecture Flow

Input Layers: User IDs and book IDs

Embedding Layers: Dense vector representations of users and books.

Concatenation Layer: Combines user and book embeddings

Dense Layers: Models complex user-book interactions

Output Layer: Predicts the strength of a user's preference for a book.

Implementation

Frameworks Used:

TensorFlow, Keras for deep learning.

Pandas and scikit-learn for data handling and preprocessing.

Model Design:

Embedding size: 50 dimensions.

Loss function: Mean Squared Error (MSE).

Optimizer: Adam.

```
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Embedding, Flatten, Input, Dense, Concatenate, Dropout
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import seaborn as sns
import networkx as nx
```

Training and Validation

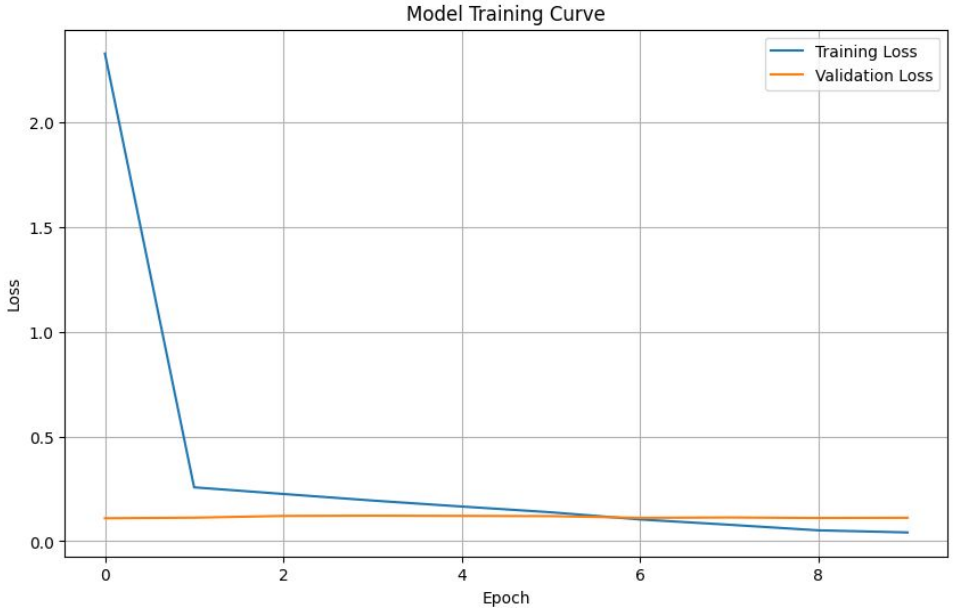
Performance

Trained over 10 epochs with batch size = 32.

Validation loss decreases consistently.

Evaluation Metrics:

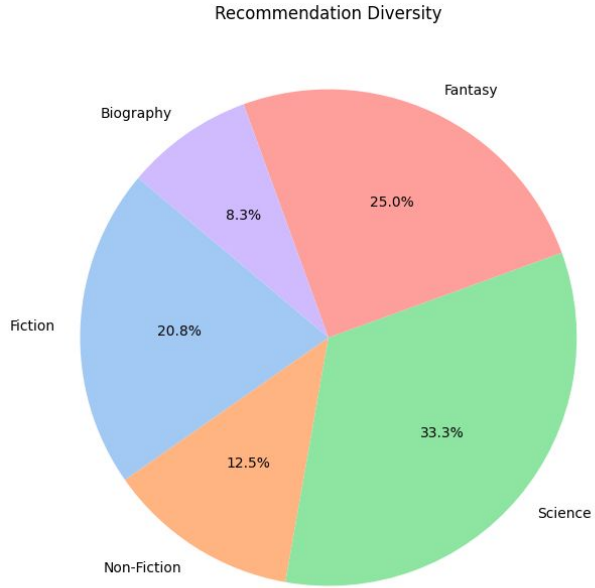
Mean Absolute Error (MAE): 0.254.



Results: Personalized Book Recommendations

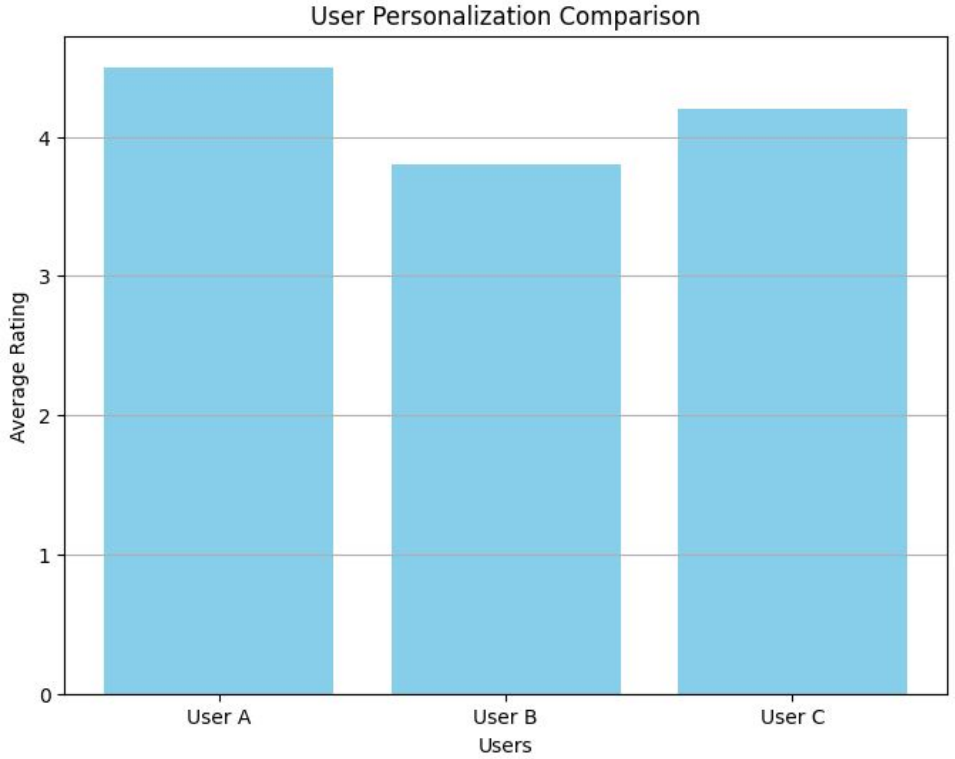
Table of Top Recommendations:

Rank	Recommended Book Title	Predicted Rating
1	Middlesex Borough (Images of America: New Jersey)	4.9
2	Bulgakov's The Master and Margarita: The Text as a Cipher	4.8
3	The Goon Show Volume 11: He's Fallen in the Water!	4.7
4	Winchester Shotguns	4.6
5	Delwau Duon: Peintiadau Nicholas Evans = Symphonies in Black	4.5



Average Ratings

This chart compares the average ratings given to the recommendations by three users: User A, User B, and User C.



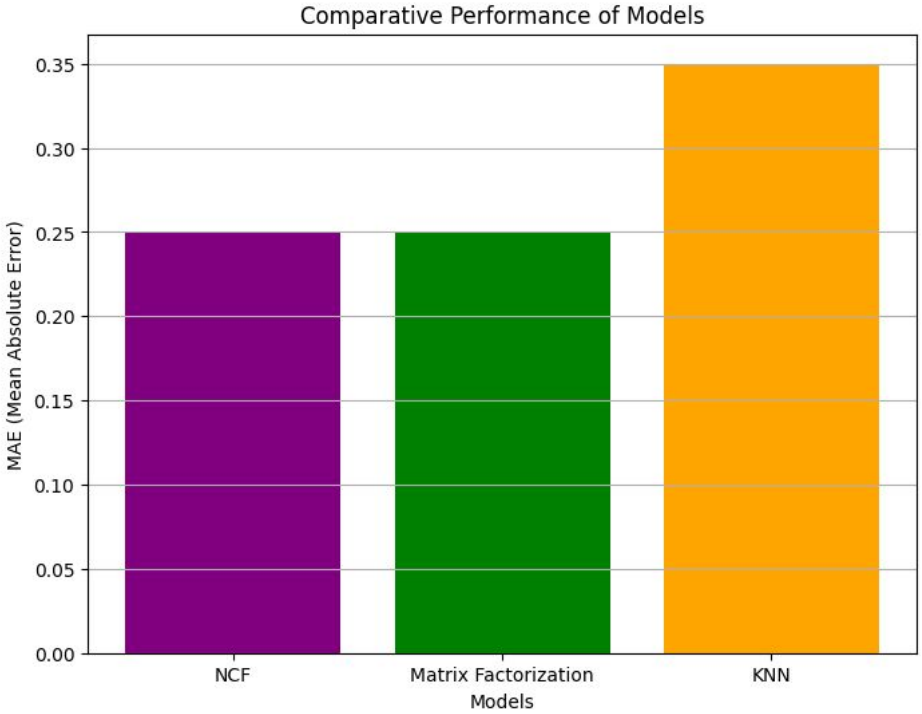
Comparative Analysis

Compared models:

NCF: Lowest MAE, superior accuracy.

Matrix Factorization and KNN: Inferior in handling complex interactions.

Visualization: Bar chart comparing MAE scores.



Challenges and Future Directions

Key Issues:

Cold-start problem: Insufficient data for new users/items.

Data sparsity: Low-density interactions in large datasets.

Scalability: Computational challenges with growing datasets.

Proposed Enhancements:

Multimodal Data: Incorporate images, text reviews, and audio features.

Real-Time Personalization: Adapt recommendations dynamically.

Explainable AI: Provide transparency for why items are recommended.

Conclusion

Summary:

Neural Collaborative Filtering effectively models user preferences.

Deep learning-based systems overcome traditional limitations.

Promising applications in education, media, and e-commerce.

???