📚 Documentation for Book Recommendation System

🔍 Introduction

The **Book Recommendation System** is designed to provide personalized book recommendations based on user preferences, ratings, and similarities among books. It leverages **collaborative filtering** (both user-based and item-based) as well as supervised and unsupervised machine learning models to enhance recommendation accuracy.

Key Features: -

- ➢ User and item-based collaborative filtering.
- ➢ Genre, author, and rating-based book suggestions.
- ➢ Secure login authentication using Google OAuth.
- ➢ Responsive front-end with HTML, CSS, Bootstrap, and JavaScript.

This system bridges the gap between data insights and user experience through thorough **Data Analytics** and an interactive interface.

🪟 System Overview

The core functionalities of the Book Recommendation System include:

Collaborative Filtering

- ➢ **User-Based Filtering**: Recommends books by finding similar users based on rating patterns.
- ➢ **Item-Based Filtering**: Recommends books similar to the ones rated highly by the user.

Genre-Based Recommendations

Due to the lack of genre data, **Wikipedia scraping** was implemented to fetch book genres, author details, and summaries.

Author and Rating-Based Suggestions

- Books by the **same author** as those already liked are suggested.
- High-rated books with similar trends are prioritized.

Data Analytics

Extensive **cleaning**, **EDA**, and **visualizations** were applied to draw insights about books and user behavior.

💻 Technology Stack

The following tools and technologies were used:

- **Frontend**: HTML, CSS, Bootstrap, JavaScript.
- **Backend**: Flask (Python-based web framework).
- **Database**: SQLite for lightweight data management, SQLAlchemy for ORM.
- **Data Analysis**: Pandas, NumPy for data manipulation and numerical computations.
- **Machine Learning Models**: Scikit-learn (RandomForestClassifier, PCA, Linear Regression, etc.).
- **Web Scraping**: BeautifulSoup, Requests for fetching and scraping book details.
- **Authentication**: Google OAuth for secure login using **Authlib**.
- **Visualizations**: Matplotlib, Seaborn for data visualizations.
- **Libraries**: bcrypt (for secure password encryption), SMOTE (for balancing imbalanced datasets).

🧹🔧📊 Data Preparation

Datasets

Books Dataset: Includes ISBN, Book Title, Author, Year of Publication, and Publisher.

Users Dataset: Includes User Id, Location and Age of user.

Ratings Dataset: Contains user ratings for books along with User Id and ISBN.

Genre Data: Fetched from Wikipedia through scraping.

Data Cleaning

Steps Taken:

1. **Handling Missing Values**: Null entries were replaced with handling logic.
2. **Duplicate Removal**: Identified and removed duplicates.
3. **Row and Column Cleaning:** Removed nan values, handled non-ASCII values, changed adequate data-types for each column, etc.

4. **Normalization**: Standardized text and numerical fields.
5. **Outlier Detection**: Managed using visualization tools.
6. **Scaling**: StandardScaler was applied for machine learning readiness.

Exploratory Data Analysis (EDA)

Visualizations included: -

★ Total Books Published each year
★ Top-rated books and authors
★ User rating trends over time
★ Density of books published
★ Correlation heatmaps for ratings and features.

🧠 Machine Learning Models

Supervised Learning

1. **Linear Regression**: Predicts user ratings for specific books.

2. **Ridge and Lasso Regression**: Improves feature selection and reduces overfitting.

3. **RandomForestClassifier**: After careful consideration, Random Forest was selected for user preference classification tasks due to its ability to handle non-linear relationships and its robustness against overfitting.

Unsupervised Learning

1. **Principal Component Analysis (PCA)**: Reduces dimensionality for faster collaborative filtering.

2. **SMOTE**: Balances class imbalance within the ratings dataset.

Evaluation Metrics

• Mean Squared Error (MSE) for regression models.

• Accuracy Score and Classification Reports for classifiers.

🤖 Recommendation Techniques

Memory-Based Collaborative Filtering

- **User-Based Filtering**: Finds similar users based on rating history.
- **Item-Based Filtering**: Finds books similar to previously liked ones.

Genre-Based Recommendations

- **Web Scraping**: Book genres, summaries, and publication dates were fetched from Wikipedia using **BeautifulSoup**.
- **Recommendations**: Suggestions are based on matching genres of previously liked books.

Author and Rating-Based Suggestions

- **Author Matching**: Books by authors similar to those rated highly.
- **Rating Trends**: Suggest books with similar rating patterns.

🛠️ Challenges Faced

Key Challenges:

1. **Lack of Genre Data**: Overcome by implementing **web scraping** using BeautifulSoup.
2. **Data Imbalance**: Resolved using **SMOTE** to balance datasets.
3. **Google OAuth Integration**: Managing sessions and secure redirection was tricky.
4. **Scalability**: Optimized collaborative filtering for large datasets.
5. **Outlier Management**: Addressed using statistical analysis and visual tools.

🔑 Login Authentication

Custom Login/Signup page: Created a tailored login and signup system using SQLAlchemy and SQLite for managing user data.

Google OAuth Integration: Implemented using **authlib** for secure Google login. Stored user credentials in **SQLite** with **bcrypt** encryption.

Session Management: Ensures secure user-specific content and recommendations.

📖 Libraries and Modules Explanation

Library	Purpose
Pandas	Data manipulation and analysis
NumPy	Numerical computations and data manipulation
Matplotlib/Seaborn	Data visualizations for exploratory data analysis (EDA)
Scikit-learn	Machine learning models and evaluation
SMOTE	Balancing imbalanced datasets in classification tasks
BeautifulSoup	Web scraping for extracting book genres, summaries, etc.
Requests	Fetching web pages and data for scraping
Flask	Web framework for building the backend of the application
SQLite3	Lightweight database management
OAuth (authlib)	Google login authentication integration
bcrypt	Secure password encryption for user authentication
PCA (Principal Component Analysis)	Dimensionality reduction for data preprocessing
RandomForest	User preference classification model

🎉 Conclusion

The **Book Recommendation System** combines advanced **machine learning techniques**, robust **data analytics**, and an interactive **web interface** to deliver a seamless user experience. The system successfully integrates multiple recommendation strategies—**collaborative filtering**, **genre-based suggestions**, and **author-based recommendations**.

Key Highlights: -

Improved user experience through web scraping and OAuth.

Balanced datasets and enhanced accuracy using supervised models.

Robust back-end processing with Flask and SQLite.