

Book Recommendation System

Problem Statement: -

Leverage neural networks to analyze and predict customer behavior, recommend books, and forecast demand in the publishing industry, improving both customer experience and sales.

This project is a Streamlit-based web application designed to recommend books and provide book-related details. It uses a machine learning model, a database, and Lottie animations for a user-friendly experience.

Features: -

1. Interactive User Interface: Navigate through pages with ease and search for books based on various criteria like Title, Author, Publisher Name, or Published Year.
2. Machine Learning: Predict book genres using a pre-trained neural network.
3. Database Integration: Fetch book details and recommendations from a local SQL database.
4. Lottie Animations: Enhance the user experience with animated visual elements.
5. Recommendations: Get related book suggestions based on predicted genres.

Prerequisites: -

1. Python 3.8+
2. Required Python libraries:
3. Streamlit ($\geq 1.2.0$)
4. Pandas ($\geq 1.3.0$)
5. SQLAlchemy ($\geq 1.4.0$)
6. TensorFlow ($\geq 2.6.0$)
7. Streamlit-Lottie ($\geq 0.0.2$)

Database: -

A Microsoft SQL Server database with a table named book_data.

Pre-trained files:-

1. book_genre_model.keras
2. vectorizer.pkl
3. label_binarizer.pkl
4. Lottie animation JSON file: Animation.json

Deep Learning Model Building: -

1. Data Preparation:
 - Preprocess the book data (e.g., clean text, tokenize, and vectorize).
 - Encode the genres using a label binarizer.
2. Model Architecture:
3. Training: (train_test_split)
4. Book Details and Recommendations
 -
5. Create a def function to display the predicted Genres related Book details to fetch. * Fetches book details and related books based on genres.

Setup: -

Instructions: -

1. Clone the Repository: git clone cd
2. Install Dependencies: pip install -r requirements.txt
3. Prepare the Database:
4. Ensure your SQL Server database is set up and populated with book data in a table named book_data.
5. Update the connection string in the get_db_connection() function with your database credentials.

Place the Required Files:-

1. Save the pre-trained files (book_genre_model.keras, vectorizer.pkl, label_binarizer.pkl) in the project directory.
2. Place the Lottie animation JSON file at the specified path (D:\Tools\VS Code\MDT35\Final_Project\Book_Recommendation_Project\Animation.json).

- 3. Run the Application:

```
streamlit run app.py
```

- 6. Access the Application:

Open your browser and navigate to <http://localhost:8501>.

File Descriptions: -

1. app.py: Main application script.
2. requirements.txt: List of required Python libraries.
3. Animation.json: JSON file for Lottie animations.

Code Overview: -

Connects to the local SQL Server database.

-

Genre Prediction: -

Prepares input, transforms it using a vectorizer, and predicts the genre using a trained model.

Navigation and Pages: -

-

1. Home Page: Displays a welcome message and an animation.
2. Details Page: Allows users to search for books and view recommendations.

AWS Deployment

1. Create an EC2 Instance:

1. Launch an Amazon EC2 instance with appropriate specifications (e.g., Ubuntu 20.04).
2. Configure the security group to allow HTTP, HTTPS, and SSH traffic.

2. Install Dependencies:

1. Install Python and required libraries on the EC2 instance.
2. Transfer the project files using SCP or a similar tool.

3. Setup Streamlit:

1. Run the Streamlit application on a specific port (e.g., 8501).
2. Use screen or tmux to keep the application running.

4. Access the Application:

1. Use the public IP of the EC2 instance to access the application via the browser

Future Enhancements: -

1. Add user authentication for personalized recommendations.
2. Enable book uploads to update the database dynamically.
3. Incorporate additional recommendation algorithms.
4. Deploy the application as a scalable service using AWS Elastic Beanstalk or Kubernetes.

Author: -

Developed by [Sudhagar Chandar]. For queries, contact [sudhakarchandar0210@gmail.com].

•