

Project Report:

Audible Insights – Intelligent Book Recommendations

1. Project Overview

Objective:

To build a smart book recommendation system using NLP, clustering, and content-based filtering. The system allows users to explore personalized book suggestions based on title, description, and genre similarities, all accessible via an interactive Streamlit interface.

2. Business Use Cases

- **Personalized Reading** – Tailored suggestions based on reader's interests
- **Enhanced Library Systems** – Support for better inventory & user engagement
- **Publisher Insights** – Identify reader preferences and emerging genres
- **Reader Engagement** – Recommend trending & high-rated books

3. Approach & Methodology

STEP 1: Data Preparation

- Loaded datasets: Audible_Catlog.csv, Audible_Catlog_Advanced_Features.csv
- Merged on: "Book Name" and "Author"

STEP 2: Data Cleaning

- Dropped missing or critical null values (e.g., Description)
- Filled missing numeric fields (Rating, Price, Reviews)
- Removed duplicates and cleaned whitespace

STEP 3: Exploratory Data Analysis (EDA)

Visual Insights:

- **Ratings Distribution:** Most books rated between 4.0 and 4.6
- **Top Reviewed Books:** e.g., *Atomic Habits*, *The Subtle Art of Not Giving a F*ck*
- **Top Authors by Rating:** Jay Shetty, James Clear, etc.
- **Genre Frequency:** Self-help and Fiction dominate
- **Publication Year Trend:** Optional (based on data availability)

4. Feature Extraction & Clustering

NLP Technique:

- TF-IDF vectorization of Description (top 5000 features)

Clustering:

- Algorithm: KMeans, with n_clusters = 5
- Purpose: Group similar books by textual description

Visualization:

PCA applied to TF-IDF vectors showed clearly separated clusters in 2D space

5. Recommendation System

A. Content-Based Filtering

- Finds similar books based on cosine similarity of TF-IDF features

B. Clustering-Based Recommendation

- Suggests other books from the same cluster

C. Hybrid Approach

- Combines clustering + similarity for refined results

6. Evaluation: Precision

To simulate recommendation quality, computed Precision using 100 random books.

Model	Precision
Content-Based	0.77
Cluster-Based	0.65
Hybrid	0.84

Hybrid performed best, balancing cluster context and description similarity.

7. Streamlit App

Features:

- Select a book
- Choose recommendation type
- Displays 5 recommendations
- Backend powered by: pandas, scikit-learn, TF-IDF, KMeans

8. Deployment Plan

- **Platform:** AWS EC2
- **Tech Stack:** Python + Streamlit + scikit-learn
- **Access:** Public IP over port 8501 (Streamlit default)
- **Optional:** Elastic Beanstalk or Docker setup for scalable deployment

9. Conclusion & Results

- Built a scalable, accurate book recommender system
- Achieved high Precision with hybrid model
- Streamlit UI provides real-time, accessible recommendations
- Suitable for libraries, book retailers etc.