PROJECT REPORT

SONG RECOMMENDATION SYSTEM USING FASTAPI

Content-Based Recommendation Using TF-IDF and Cosine Similarity

Submitted by

Syeda Alia Samia

AI Recommendation Bootcamp

17/12/2025

TABLE OF CONTENTS

SONG RECOMMENDATION SYSTEM USING FASTAPI

Content-Based Recommendation using TF-IDF and Cosine Similarity

---

1. INTRODUCTION

Music recommendation systems help users discover songs that match their interests. With the rapid growth of digital music platforms, recommending relevant songs has become an important application of Machine Learning.

This project implements a **content-based song recommendation system** using **song lyrics**. The system analyzes the textual content of songs and recommends similar songs based on lyrical similarity. The recommendation logic is exposed through a **REST API using FastAPI**.

---

2. OBJECTIVE OF THE PROJECT

The main objectives of this project are:

- To analyze song lyrics and extract meaningful features
- To recommend similar songs based on content similarity
- To implement TF-IDF and cosine similarity
- To expose the recommendation system as a REST API using FastAPI

---

3. DATASET DESCRIPTION

The dataset used for this project is **spotify_millsongdata.csv**.

Dataset Details:

- Total songs: ~57,000
- Columns:
    - artist – Name of the artist
    - song – Song title
    - link – Lyrics webpage link
    - text – Full song lyrics

Usage:

- The text column (lyrics) is used as the primary feature for recommendation.
- Songs with missing lyrics are removed.
- A subset of 5000 songs is used for faster processing.

---

4. SYSTEM ARCHITECTURE

The system follows the below flow:

Dataset → Text Preprocessing → TF-IDF Vectorization

→ Cosine Similarity → Recommendation Function

→ FastAPI → API Response

---

5. METHODOLOGY

5.1 Data Preprocessing

- Removed songs with empty lyrics
- Combined artist name, song title, and lyrics into a single text feature

5.2 Feature Extraction (TF-IDF)

- TF-IDF (Term Frequency–Inverse Document Frequency) is used to convert text into numerical vectors
- Common words are removed using stop-word filtering
- Important words receive higher weights

5.3 Similarity Measurement

- Cosine similarity is used to measure similarity between song vectors
- Songs with higher cosine similarity are considered more similar

5.4 Recommendation Logic

- Given a song name, the system finds the most similar songs based on cosine similarity scores
- The top N similar songs are returned as recommendations

6. API IMPLEMENTATION USING FASTAPI

FastAPI is used to expose the recommendation system as a REST API.

API Endpoint:

GET /recommend

Query Parameters:

- song – Name of the song
- top_n – Number of recommendations (default = 5)

Sample Request:

http://127.0.0.1:8000/recommend?song=Imagine

Sample Response:

Code **Details**

200 **Response body**

Download


```
{
  "input_song": "Imagine",
  "recommendations": [
    {
      "song": "Imagine",
      "artist": "Diana Ross"
    },
    {
      "song": "Dream On Little Dreamer",
      "artist": "Perry Como"
    },
    {
      "song": "What Can I Say?",
```

Code Details

```
    "artist": "Howard Jones"
  },
  {
    "song": "Someday",
    "artist": "John Legend"
  },
  {
    "song": "I Have Nothing",
    "artist": "Glee"
  }
 ]
}
```

FastAPI automatically generates API documentation using Swagger UI.

---

7. TESTING AND RESULTS

Testing Method:

- The API was tested using FastAPI's built-in Swagger UI
- Requests were sent to the /recommend endpoint
- The system successfully returned recommended songs in JSON format

Sample Test Case:

Test Case ID	Input Song	Expected Output	Status
TC-01	Imagine	List of similar songs	Pass

8. OBSERVATIONS

- The system recommends songs based on **lyrical similarity**
- Recommendations may include songs from different artists
- This behavior is expected in content-based filtering

---

9. ADVANTAGES

- No user ratings required
- Simple and efficient
- Works well with textual data
- Easy to extend as an API

---

10. LIMITATIONS

- Does not consider user preferences
- Requires lyrics for recommendation
- Cannot recommend new songs without text data

---

11. FUTURE ENHANCEMENTS

- Add user-based personalization
- Include genre and audio features
- Deploy API to cloud platform
- Add frontend interface

---

12. CONCLUSION

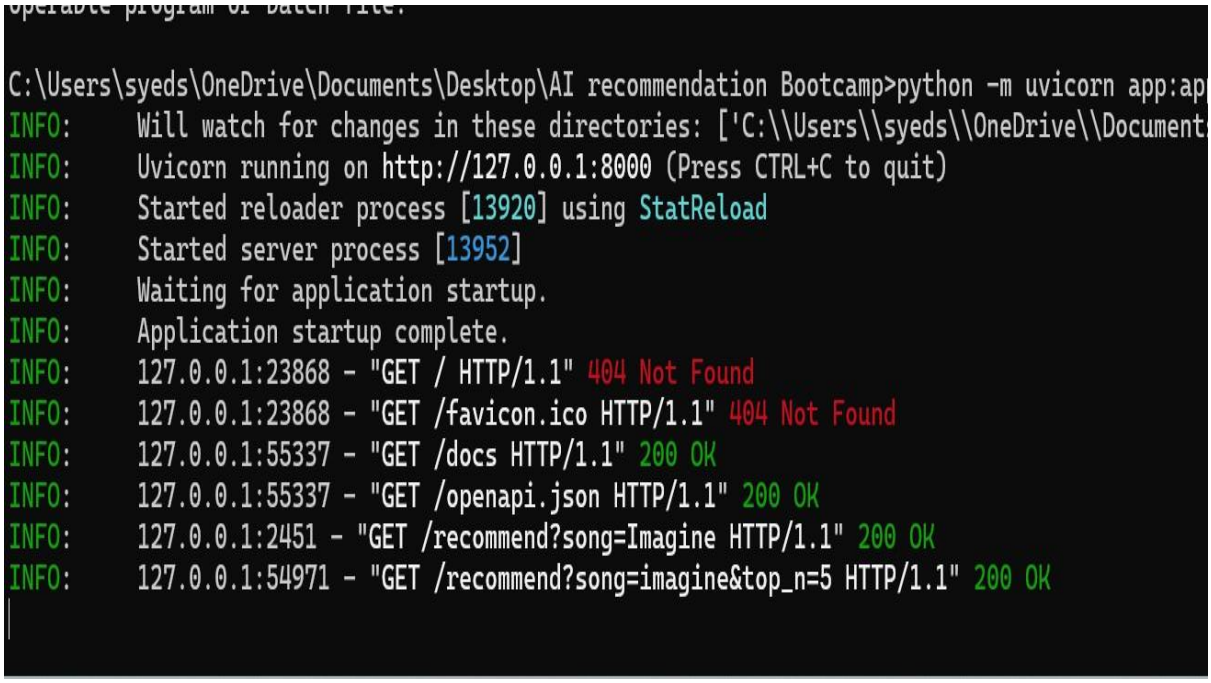
This project successfully demonstrates a content-based song recommendation system using TF-IDF and cosine similarity. By integrating the recommendation logic with FastAPI, the system provides a scalable and professional API solution suitable for real-world applications.

13. TOOLS AND TECHNOLOGIES USED

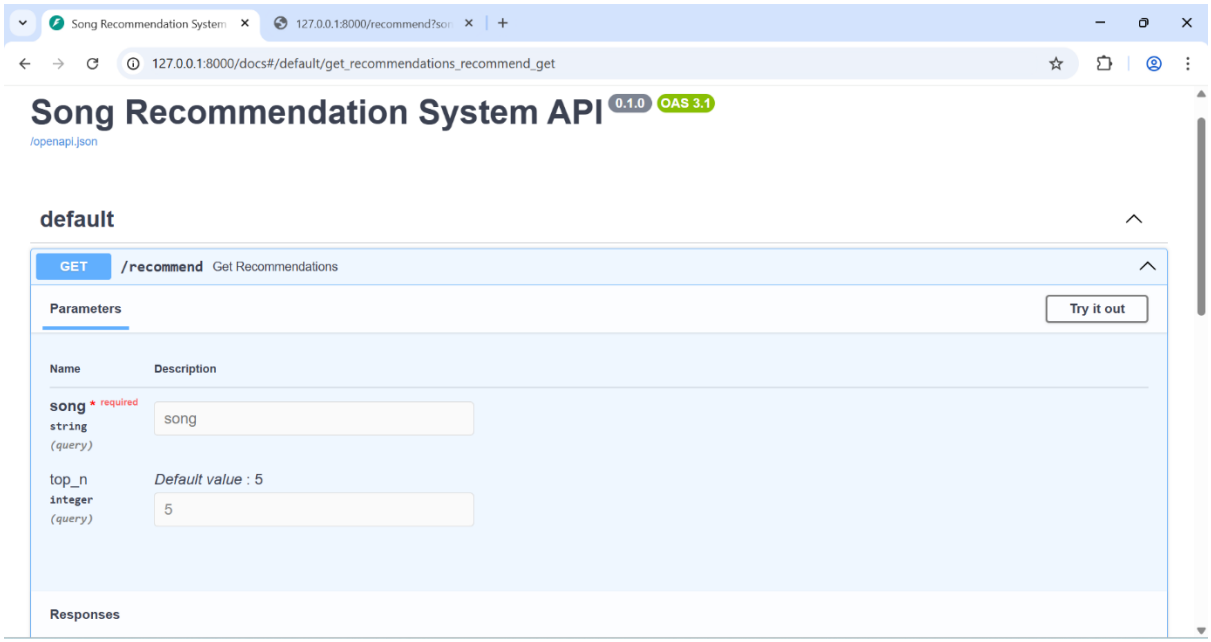
- Python
- Pandas
- Scikit-learn
- FastAPI
- Uvicorn

SCREENSHOTS

API RUNNING:



Swagger UI Screenshot:



API Output Screenshot