

Project Title: Music Recommendation System Using Python

Short Description: Develop a system that suggests songs to users based on their listening history and preferences, utilizing Python and the Spotify API.

Difficulty Level: Intermediate

AI-ML Concepts Used: Content-Based Filtering, Collaborative Filtering

Learning Outcome: Recommend songs based on preferences

Programming Skills Mastered: API Integration, Data Analysis

Introduction

A Music Recommendation System assists users in discovering new and relevant musical content based on their preferences and listening behavior. Platforms like Spotify and Apple Music utilize such systems to enhance user experience by providing personalized music suggestions. In this project, we will build a Music Recommendation System using Python and the Spotify API. This project is suitable for Class 9 and 10 CBSE/ICSE students, offering an introduction to recommendation systems and API integration.

Implementation Steps

1. Set Up Spotify Developer Account

To access Spotify's music data, we need to create a Spotify Developer Account and register an application to obtain the necessary credentials (Client ID and Client Secret).

- **Create a Spotify Account:** Sign up at [Spotify](#) if you don't have an account.
- **Access Spotify Developer Dashboard:** Visit the [Spotify Developer Dashboard](#) and log in.
- **Create an App:** In the dashboard, click "Create an App," provide necessary details, and retrieve your Client ID and Client Secret.

2. Install Required Libraries

We'll use the `requests` library to interact with the Spotify API and `pandas` for data handling.

```
bash
CopyEdit
pip install requests pandas
```

3. Obtain Access Token

Use the Client ID and Client Secret to obtain an access token, which allows authorized requests to the Spotify API.

```
python
CopyEdit
import requests
import base64

CLIENT_ID = 'your_client_id'
CLIENT_SECRET = 'your_client_secret'

auth_url = 'https://accounts.spotify.com/api/token'
auth_header =
base64.b64encode(f"{CLIENT_ID}:{CLIENT_SECRET}".encode()).decode()
auth_data = {'grant_type': 'client_credentials'}

response = requests.post(auth_url, headers={'Authorization': f'Basic
{auth_header}'}, data=auth_data)
access_token = response.json().get('access_token')
```

4. Fetch User's Recently Played Tracks

Retrieve the user's recently played tracks to analyze their listening habits.

```
python
CopyEdit
recently_played_url = 'https://api.spotify.com/v1/me/player/recently-
played'
headers = {'Authorization': f'Bearer {access_token}'}

response = requests.get(recently_played_url, headers=headers)
recently_played = response.json()
```

5. Extract Track Features

For each track, extract relevant features such as genre, artist, and tempo to understand user preferences.

```
python
CopyEdit
track_ids = [item['track']['id'] for item in
recently_played['items']]
features_url = f"https://api.spotify.com/v1/audio-
features?ids={'.'.join(track_ids)}"

response = requests.get(features_url, headers=headers)
features = response.json()
```

6. Recommend Similar Tracks

Based on the extracted features, recommend tracks that align with the user's preferences.

```
python
CopyEdit
recommendations_url =
f"https://api.spotify.com/v1/recommendations?seed_tracks={track_ids[0
]}&limit=10"

response = requests.get(recommendations_url, headers=headers)
recommendations = response.json()
```

7. Display Recommendations

Present the recommended tracks to the user in a readable format.

```
python
CopyEdit
for track in recommendations['tracks']:
    print(f"Track: {track['name']}, Artist:
{track['artists'][0]['name']}")
```

Conclusion

By following these steps, students will learn how to integrate with external APIs, process and analyze data, and implement basic recommendation algorithms. This project provides a practical introduction to building personalized recommendation systems using real-world data.

Disclaimer

The projects provided on this website are for educational purposes only. They are designed to help students understand fundamental concepts in AI, machine learning, and programming. While we strive for accuracy and quality, we do not guarantee the correctness or reliability of the projects for real-world applications.

Users are encouraged to modify and test the projects as per their requirements. We are not responsible for any issues, data loss, or unintended consequences resulting from the use of these projects.

For any queries or further assistance, please get in touch via email at contact@aimltechacademy.org.

A large, light blue watermark logo for AI-ML TECH ACADEMY is centered on the page. The logo consists of a circular border containing a stylized gear or circuit-like symbol at the top. Below the symbol, the text "AI-ML" is written in a large, bold, sans-serif font, and "TECH ACADEMY" is written in a smaller, all-caps, sans-serif font below it. At the bottom of the circle is a decorative horizontal line with a central V-shaped element.

AI-ML
TECH ACADEMY