

```

import pandas as pd

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.metrics.pairwise import cosine_similarity


def load_data(file_path):
    """Loads music dataset from a CSV file."""
    return pd.read_csv(file_path)


def preprocess_data(df):
    """Combines relevant features into a single string for similarity computation."""
    df['combined_features'] = df['artist'] + ' ' + df['genre'] + ' ' + df['lyrics']
    return df


def compute_similarity(df):
    """Computes cosine similarity between songs."""
    vectorizer = TfidfVectorizer(stop_words='english')
    feature_matrix = vectorizer.fit_transform(df['combined_features'])
    return cosine_similarity(feature_matrix)


def recommend_songs(song_title, df, similarity_matrix, top_n=5):
    """Recommends top_n similar songs based on cosine similarity."""
    if song_title not in df['title'].values:
        return "Song not found in the dataset."

    song_idx = df[df['title'] == song_title].index[0]
    similarity_scores = list(enumerate(similarity_matrix[song_idx]))
    sorted_songs = sorted(similarity_scores, key=lambda x: x[1], reverse=True)[1:top_n+1]

```

```
return df.iloc[[i[0] for i in sorted_songs]][['title', 'artist', 'genre']]
```

```
# Example usage
```

```
if __name__ == "__main__":
```

```
    dataset_path = "music_data.csv" # Ensure this dataset exists with 'title', 'artist', 'genre',  
    'lyrics' columns
```

```
    df = load_data(dataset_path)
```

```
    df = preprocess_data(df)
```

```
    similarity_matrix = compute_similarity(df)
```

```
    song_name = "Shape of You"
```

```
    recommendations = recommend_songs(song_name, df, similarity_matrix)
```

```
    print(recommendations)
```