Song Similarity Script

This script compares the similarity between two songs using two types of audio embeddings: **EffNet** and **Maest**. It calculates the **cosine similarity** and **Euclidean distance** between their vector representations.

🧠 What It Does

- Loads **EffNet embeddings** from JSON files and **Maest embeddings** from pickle files.
- Embeddings come from two populations: songs released **before 2012** and **after 2018**.
- Computes similarity between two songs using:
    - **Cosine Similarity** (1 = identical direction, 0 = orthogonal)
    - **Euclidean Distance** (0 = identical location in space)

📁 Folder Structure

```
project_root/
├── code/
│   └── similarities.py  ← this script
├── song_embeddings/
│   ├── embeddings_effnet/
│   │   ├── before_2012_effnet_embeddings.json
│   │   └── after_2018_effnet_embeddings.json
│   └── Archivo_maest_pkl/
│       ├── before_2012_maest_embeddings.pkl
│       └── after_2018_maest_embeddings.pkl
```

📦 Dependencies

- `numpy`
- `scipy`

- `json`
- `pickle`
- `os`

Install with:

```
pip install numpy scipy
```

🚀 How to Use

Run the script directly:

```
python similarities.py
```

The script compares **two predefined songs**:

- `"Els_Catarres::Caramelles"`
- `"31_fam::al_cantu"`

It prints the cosine similarity and Euclidean distance for both EffNet and Maest embeddings (if available).

🧾 Output Example

```
Cosine Similarity: 0.8732
Euclidean Distance: 2.6147

[Maest Embeddings]
Cosine Similarity: 0.7910
Euclidean Distance: 5.4921
```

🔍 Functions Overview

- `load_effnet_embeddings()`: Loads and averages frame-wise EffNet embeddings from JSON files.
- `load_maest_embeddings()`: Loads and flattens Maest embeddings from pickled files.
- `compute_similarity(e1, e2)`: Returns cosine similarity and Euclidean distance between two embeddings.

📝 Notes

- Embedding keys are in the format: `"artist::song"`.
- The Maest embeddings are truncated to a fixed size of 1000 elements for comparison.

📬 Contact

For any questions, reach out to the developers of the MTG-102 project or the Music Technology Lab team.