Hiring Assignment - Senior AI/ML Research Engineer

Assessment: Research Innovation Lab

We have designed this assessment to evaluate your ability to approach complex problems in artificial intelligence research and implementation. The assessment consists of **two distinct challenges**, each targeting a specific area of expertise: **Agentic Audio Intelligence for Cover Song Similarity** and **Title Normalization and Agentic Enrichment for Music Asset Titles**.

The candidate will perform a **full implementation of task A**, while **B** will be a **detailed description of the approach** that could be followed to address the problem.

A. Implemenation: Agentic Audio Intelligence for Cover Song Similarity

Objective

This task evaluates your ability to design and implement an **AI agent-based system** that **quantifies and explains the similarity** between original songs and their cover versions using **multimodal data**. You are expected to leverage the **Da-TACOS dataset** and implement a **multi-tool agent** capable of reasoning over audio features, metadata, and (optionally) lyrics to produce a **similarity score** with **interpretable outputs**.

The goal is to demonstrate how **agentic systems** can enhance music similarity understanding in a modular and explainable way—critical for use cases in **digital rights management**, **copyright verification**, and **music recommendation**.

Scenario

Imagine an AI Agent tasked with validating whether two musical recordings represent original/cover relationships. To perform this, it can activate a set of specialized tools that operate over different modalities—**audio**, **metadata**, and (optionally) **lyrics**—and then combine the outputs into a **final similarity score** with justification.

Dataset

Use the Da-TACOS Dataset, which includes:

- Cover cliques with known relationships
- Audio-derived features (Chroma, HPCP)
- Metadata (e.g., title, artist, release year)

You may optionally supplement with:

- Lyrics (via APIs or scraped metadata)
- Pretrained music/audio embeddings (e.g., CLAP, MERT, MuLan)

Agent Architecture Requirements

You are required to build an **AI Agent** with at least the following **three tools**:

Tool 1: Audio Embedding Similarity Tool

- Uses pre-extracted audio features (e.g., chroma/HPCP from Da-TACOS) or pretrained audio embedding models (e.g., MERT, CLAP).
- Computes similarity between two tracks based on tonal, harmonic, or timbral alignment.
- Outputs: `audio_similarity_score ∈ [0,1]`

Tool 2: Metadata Comparison Tool

- Parses metadata fields (e.g., title, artist name, year) and computes overlap/similarity.
- Can apply fuzzy string matching or embedding-based approaches (e.g., sentence transformers).
- Outputs: `metadata_similarity_score ∈ [0,1]` and justification (e.g., “same label”, “name overlap”)

Tool 3: Multimodal Aggregator + Explanation Tool (Optional)

- Combines scores from audio and metadata tools using a weighted scheme or a learned model (e.g., XGBoost, shallow MLP).
- Produces final similarity score along with a **natural language explanation** (e.g., “High tonal overlap and same release label detected”).
- Outputs:
  - `final_similarity_score ∈ [0,1]`
  - `explanation: str`

You may optionally implement:

- A **Lyrics Embedding Tool** that extracts and compares semantic similarity using models like BERT or MPNet.
- A **Visual Explanation Tool** that uses SHAP/LIME to highlight input contributions.

What to Build or Propose

- A functioning AI Agent (script or framework) that takes two tracks (or IDs from the Da-TACOS dataset) and:
  - Invokes the tools
  - Aggregates results
  - Outputs a similarity score and human-readable explanation

---

Evaluation Criteria

Aspect	Criteria
Multimodal Reasoning	Effective use of audio features + metadata (optional lyrics)
Tool Design	Modularity and clarity in how each tool works
Explainability	Quality and interpretability of the generated justifications

Scalability	Ability to extend to new tools or larger datasets
Code Quality	Maintainability, modularity, documentation
Research Thinking	Creativity, understanding of limitations, alignment with current trends

---

Deliverables

- A Python-based prototype implementing the agent and tools (script or notebook)
- CLI or function to run: `compare_tracks(song_id_1, song_id_2)`
- Sample outputs and evaluations (e.g., comparisons between known original/cover and unrelated songs)
- A short report (~2–4 pages) describing:
  - Agent architecture
  - Tools and models used
  - Design decisions and trade-offs
  - Limitations and ideas for future improvements

Submission

- Upload your implementation to a public or private GitHub repository
- Include:
  - `README.md` with setup and usage instructions
  - Annotated code for tools and agent logic
  - PDF report or `REPORT.md`

Encouraged Libraries & Tools

- **Audio**: Librosa, Essentia, TorchAudio, CLAP, MERT
- **Embeddings**: Sentence Transformers, BERT, MPNet
- **Similarity**: Scikit-learn, FAISS, cosine similarity
- **Explainability**: SHAP, LIME, attention visualization
- **Agents**: LangChain, OpenAgents, CrewAI (or simple Python logic mimicking agent behavior)

B. Mini Survey: Title Normalization and Agentic Enrichment for Music Asset Titles

Objective

This task evaluates your ability to design a **machine learning or LLM-based system** to **clean and normalize music asset titles**, while integrating **agentic reasoning**, **external knowledge enrichment**, and **explanation of the decision process**.

Assuming you have a dataset of annotated music asset titles, where each token has been labeled with categories such as clean title, artist name, version info, album title, or miscellaneous. Your job is to **design a comprehensive technical report** that showcases your ability to reason over noisy metadata and propose generalizable solutions with explainability.

Dataset

Asset Title Annotations — A CSV dataset with annotated asset titles.

Each row contains:

- Full `raw_title` string
- Annotations indicating:
  - `title` (main title of the asset)
  - `artist` (e.g., performer or featured names)
  - `version` (e.g., Acoustic, Radio Edit)
  - `album_title` (if mentioned)
  - `misc` (e.g., country names, references, dates)

The dataset exhibits:

- Redundancies and false matches in version/artist segments
- Sparse frequency for some tags like `album_title`, `misc`

Goal

Create or propose a system that:

1. **Cleans and normalizes the title** by isolating the true `title` field.
2. Enriches ambiguous/missing parts using **external knowledge** (e.g., from Wikipedia, Spotify, or MusicBrainz).
3. Produces a **transparent explanation** (via rules, XAI, or LLM reasoning) for how each segment of the input was classified or discarded.

Mini Research Survey and Technical Proposal

Deliver a detailed technical report that:

- Explores and compares possible modeling options
- Proposes a modular agentic architecture for this task
- Compares tools for enrichment (Spotify API, Wikipedia scraping, etc.)
- Discusses evaluation metrics and error analysis strategies
- Includes examples and architecture diagrams

Evaluation Criteria

Category	Description

Title Normalization	Ability to identify and clean titles accurately using NLP/ML tools
External Knowledge Use	Ability to propose or use enrichment for disambiguation and validation
Explanation & Reasoning	Use of interpretable outputs or explainable models
Modularity	Clear separation of components (classifier, enrichment, resolver)
Research Skills	Quality of the mini-survey or report (clarity, citations, comparison)
Technical Depth	Quality and creativity of chosen models, frameworks, or tools

---

Deliverables

Mini Survey:

- PDF or Markdown technical report
- Architecture diagram (if proposing Agentic system)
- Real examples with theoretical enrichment/explanation
- Comparative analysis of existing methods (optional: XAI, LLMs)

Submission

- PDF report
- Bonus: mention limitations and potential for automation