Round 2

AI/ML Take-Home Task - 48h

Task Overview:

Goal: show clear thinking, sensible trade-offs and clean, testable endpoitns.
(Note: Heavy modeling is not required.)

Using the provided sample data, build a mini-pipeline that:

Pre-Transcription *(implementation not required - outline only)***:**
Briefly outline what audio pre-processing (if any) you would consider and why.

1. **Transcription:**
   Use Whisper or similar. Keep timestamps if convenient.

2. **Topic Extraction:**
   Extract the top 5 discussion topics from the transcript.
   Optionally, you may also consider other conversational/social cues *(explain your approach, only topics must be implemented).*

3. **Vectorization:**
   Choose approach for extracted topics (one-hot, TF-IDF, or embeddings).

4. **Load User Profiles:**
   Use provided JSON psychometric profiles. Each user represents a speaker in the audio file: User 1 = Joe Rogan; User 2 = Elon Musk.

5. **Fusion: topics x personality (*open-ended)***
   Propose and justify how you weight topics relative to each user's personality before combining. Construct a combined vector per user.

6. **Compatibility Score + Interpretation:**
   Compute a compatibility score between the two users (e.g. cosine similarity over their combined vectors) and returns an interpretation of that score.

7. **Fast API Endpoints:**
   Wrap your pipeline into a FastAPI application that exposes:
   POST/transcribe (returns a transcript), POST/summarise (returns topic list) and POST/match (returns compatibility score).

8. **Live, Testable Demo:**
   Make the app reachable so it can be tested via browser (Swagger/OpenAPI UI).

Sample Data (in sample_data/):

Audio: 'Is Mars the Future of Humanity?' - Joe Rogan Asks Elon Musk

JSON:

Traits = [openness, conscientiousness, extraversion, agreeableness, neuroticism]

```
[
  { "id": "user_1", "psychometrics": [0.8, 0.4, 0.7, 0.2, 0.9] },
  { "id": "user_2", "psychometrics": [0.3, 0.9, 0.1, 0.6, 0.4] }
]
```

Sugested Tech Stack:

Use any combination of:
Transcription: OpenAI Whisper, Hugging Face whisper, AssemblyAI
Topic Extraction: spaCy, Gensim, Transformers (BERT/DistilBERT), keyBERT
Vectorization: scikit-learn TF-IDF, SentenceTransformers, et
Matching Logic: cosine similarity on combined vectors (numpy / scikit-learn)
LLM Fine Tuning: GPT-2, LLaMA via Hugging Face transformers + accelerate or OpenAI API
Orchestration: PyTorch/TensorFlow; LangChain (optional)

Deliverables:

1. GitHub repo with clean, dated, well-documented code.

2. README covering:
   - How to load and use the sample data
   - Architecture & design decisions
   - Pre-Processing Outline (if any)
   - Topic vectorization method and any considerations of social cues.
   - Fusion/Weighting Rationale
   - Matching logic and edge-case handling (thresholds or re-sampling)
   - Live URL

3. Short write-up (≤ 300 words) on: reasoning, trade-offs and next steps.