Research Report: LLM-Based Smart Recommendation Engine for Vedaz

LLM Stack Recommendation

I recommend using OpenAI's Flagship chat models like GPT-4.1 or GPT-4o. We can also use o4-mini if we want a faster, more affordable reasoning model. Alternatively, we can use open-source models like LLaMA/Mistral + Sentence Transformers which are free to use but we require GPU Infrastructure to run these models.

Why use OpenAI API?

1. High Accuracy
2. Managed API - Doesn't require building GPU Infrastructure, no need of custom hosting.
3. Flexible Pricing - gpt-4.1(Input: $2.00, Output: $8.00 per 1 Million Tokens), gpt-4.1-mini($0.40, $1.60), gpt-4o($2.50, $10.0), gpt-4o-mini($0.15, $0.60).

Hosting & Scaling

- **Cloud:** AWS or GCP
- **LLM Inference:** OpenAI API
- **Backend API:** AWS Lambda or Google Cloud Functions
- **Vector DB:** Pinecone Standard Plan, ~$50/month with pay-as-you-go scaling

Data Flow:

1. Send user chat or profile to OpenAI embedding
2. Use Pinecone to store and retrieve vectors
3. Use similarities to rank astrologers
4. Apply LLM for reasoning in suggestions (optional)

Monthly Cost Estimate

Assumptions:

- 50,000 users, ~2 sessions/month, total interactions = ~100,000
- Each interaction: embedding (~500 tokens) + inference (~1,000 tokens)
- Using gpt-4o-mini, text-embedding-3-small, Pinecone and AWS Lambda

Component	Assumptions	Unit Cost	Cost Estimation (/month)
OpenAI embedding	100k inputs × 500 tokens = 50M tokens	$0.02/1M tokens	$1.00
GPT-4o-mini inference (optional)	100k interactions @ 500 input + 500 outputs = 100M tokens	$0.60/1M tokens	$60.00
Pinecone	Standard plan	$50/month + pay as you go	$50.00 (baseline)
Backend Hosting (AWS Lambda)	Under the free tier limit, 1M requests /month, 3.2M seconds of compute time /month	Free, $0.0000166667 for every GB-second, $0.20 per 1M requests	Free
Total			~$110, ~$51 if no inference from gpt-4o

Privacy & Safety Considerations

- **User Anonymity:** We make sure personal information like names, contact details, or birth dates are not directly used in model prompts. This helps protect user identity while still allowing useful recommendations.
- **Pseudonymization:** Instead of using real user data, we replace it with fake IDs or coded values. This keeps the actual data hidden, even if something goes wrong.
- **Secure Data Storage:** Any user information or chat history used for improving recommendations is stored securely, with access only given to authorized systems or people.
- **Model Boundaries:** The AI doesn't make health or life decisions. It only gives soft recommendations like "you might like this astrologer," not predictions or serious advice.
- **No Unnecessary Tracking:** We don't collect extra user data beyond what's needed for the recommendation engine to work properly. This avoids overreach and builds trust.