Raj Kumar Nelluri

(201) 554-8009 | rajkumarn2002@gmail.com | LinkedIn | GitHub | Portfolio

Summary

AI/ML Engineer building production LLM and agentic systems. Contributed upstream fix (PR #20, CLA signed) to Meta FAIR's TRIBE v2 — added int8 fallback enabling non-CUDA devices to run the full pipeline. Shipped three live AI systems on AWS EC2: **CineNeuro** (multimodal neural audience intelligence using TRIBE v2 + Llama 3.2 + V-JEPA2), a **LangGraph financial research agent** with deterministic decision logic and zero LLM hallucination risk, and an **enterprise RAG chatbot** with MMR retrieval and hallucination guardrails. MS Computer Science, Pace University.

Technical Skills

Languages: Python, SQL
LLM & Agentic Systems: LangChain, LangGraph, RAG Pipelines, OpenAI API (GPT-4o), Llama 3.2, Prompt Engineering, Embeddings, ChromaDB, FAISS, Vector Databases, Hugging Face, MMR Retrieval, Hallucination Guardrails, Deterministic Decision Systems
Machine Learning: PyTorch, TensorFlow, Scikit-learn, XGBoost, YOLOv8, ResNet-50, LSTM, Prophet, SHAP
Multimodal / Foundation Models: Meta FAIR TRIBE v2, V-JEPA2, Wav2Vec-BERT, Whisper
Backend & APIs: FastAPI, REST APIs, Redis, Streamlit, Docker, Nginx
Cloud & MLOps: AWS (EC2, SageMaker, Lambda, S3, Kinesis, RDS, CloudWatch, EventBridge, Comprehend), MLflow, Weights & Biases, GitHub Actions
Data & Analytics: Pandas, NumPy, Matplotlib, Tableau, Power BI

Experience

CineNeuro — Neural Audience Intelligence Platform
Meta TRIBE v2, V-JEPA2, Wav2Vec-BERT, Llama 3.2, FastAPI, React, Docker, AWS EC2
Live: cineneuro.rajkumarai.dev | Code: github.com/Rajkumar2002-Rk/CineNeuro

- **Engineered** a multimodal brain encoding pipeline predicting fMRI activations across **20,484 cortical vertices per second** of video, using Meta FAIR's TRIBE v2 fed by three feature extractors (V-JEPA2 for vision, Wav2Vec-BERT for audio, Llama 3.2 for transcripts)
- **Submitted PR #20 to Meta FAIR's facebookresearch/tribev2** (CLA signed, checks passing) — added int8 `compute_type` fallback for WhisperX on non-CUDA devices (Apple Silicon, CPU-only Linux), enabling the full video → audio → TRIBE v2 brain prediction pipeline to run on non-NVIDIA hardware with <1% WER degradation
- **Mapped** activations to 5 emotion channels across 7 brain regions with neuroscience-based weighting, producing second-by-second audience emotion maps for trailer edit intelligence
- **Designed** two-tier cost-optimized deployment: g4dn.xlarge GPU (Tesla T4, 15GB VRAM) on-demand at $0.53/hr for inference, t3.micro serves React dashboard 24/7 with pre-computed JSON at <100ms latency — monthly hosting cost near $0
- **Solved** VRAM constraint (12.89GB across three extractors barely fitting 15GB T4) with careful sequential model loading; multi-stage Docker build keeps production image at ∼500MB with no ML weights
- **Benchmarked** against 5 reference trailers (Oppenheimer, Avengers Endgame, Inception, Interstellar, The Dark Knight); full 2-minute trailer analysis costs ∼$0.50 in GPU time vs thousands for a traditional test screening

FinTel — AI Financial Research Agent
LangGraph, GPT-4o, FastAPI, Alpha Vantage, Redis, Docker, AWS EC2, Nginx
Live: fintel.rajkumarai.dev | Code: github.com/Rajkumar2002-Rk/financial-research-agent

- **Architected** a **7-step async LangGraph pipeline** producing deterministic BUY/HOLD/SELL recommendations in **under 30 seconds** by integrating live market data, company fundamentals, and news sentiment
- **Constrained GPT-4o to explanation-only role** with zero influence over the decision — all recommendations produced by a normalized multi-factor scoring engine (Technical 0–25, Fundamental 0–40, Sentiment 0–15), ensuring reproducible and auditable outputs with zero LLM hallucination risk
- **Implemented** conflict-aware decision logic (signal variance >0.15 overrides to HOLD in 35–55% ambiguity band) plus a 5-factor confidence model blocking recommendations below 20% confidence — solving the false-conviction failure mode of generic LLM finance tools
- **Cut** repeated query latency from ∼15s to <100ms (**10x API cost reduction**) with Redis TTL 900s caching; missing-data handling excludes factors from denominator rather than penalizing, eliminating false SELL signals
- **Shipped** portfolio ranking mode supporting 2–8 tickers with proportional capital allocation; deployed multi-service architecture with Nginx domain-based routing and Docker Compose on AWS EC2

Enterprise RAG Chatbot — AI Document Intelligence
LangChain, ChromaDB, OpenAI API, FastAPI, Streamlit, Redis, Docker, AWS EC2, Nginx
Live: chatbot.rajkumarai.dev | Code: github.com/Rajkumar2002-Rk/rag-chatbot

- **Built and deployed** a production RAG system end-to-end — PDF ingestion, chunking (1000-char / 200-char overlap, RecursiveCharacterTextSplitter), embedding (OpenAI text-embedding-3-small), persistent ChromaDB vector store, and streaming GPT-4 generation behind Nginx with HTTPS via Let's Encrypt

- **Used MMR (Maximal Marginal Relevance) retrieval** over cosine similarity to fetch a larger candidate pool and re-rank for relevance plus diversity, preventing near-duplicate chunks from reaching the LLM; achieved **87% retrieval accuracy** on benchmark query set
- **Added hallucination guardrails** gating the LLM call on retrieval confidence — low scores, empty retrievals, or insufficient context return a fallback message instead of guessing, producing zero hallucination on grounded queries
- **Engineered** a rule-based query classifier (no extra LLM call) detecting factual / complex / ambiguous / keyword intent and dynamically adjusting top_k and fetch_k retrieval parameters
- **Enforced** three cost guards firing *before* any LLM call: rate limiting via slowapi (20 req/min/IP), 500-char input cap, and tiktoken-based token budget; added Redis caching with query normalization for semantically-identical queries
- **Structured** prompts with explicit `[SOURCE N: filename --- Page: X]` citation format forcing the LLM to cite every fact from retrieved context; shipped dual interfaces (Streamlit chat UI + FastAPI Swagger at /api/docs)

Education

Pace University — New York, NY
Master of Science in Computer Science — *Sep 2023 – Apr 2025*
Amrita Vishwa Vidyapeetham — Bengaluru, India
Bachelor of Technology in Artificial Intelligence — *Jul 2019 – Apr 2023*

Additional Projects

Customer Churn Prediction — MLOps Pipeline
XGBoost, AWS SageMaker, Lambda, CloudWatch, EventBridge, Comprehend, MLflow, SHAP, Docker
Code: github.com/Rajkumar2002-Rk/Customer_Chrun_Prediction

- **Built** end-to-end MLOps pipeline achieving **91% accuracy, 89% F1**: S3 ingestion → Lambda ETL → SageMaker training → real-time inference endpoint with 21 engineered features
- **Integrated** AWS Comprehend NLP for sentiment signals from 7k+ customer support tickets as additional churn predictors; CloudWatch + EventBridge triggers automated retraining on data drift; MLflow tracks experiments; SHAP waterfall plots make every prediction auditable

Certifications

AWS Certified Cloud Practitioner — Amazon Web Services — 2025