Krishna Chaitanya Bodepudi

St. Louis, MO (Open to Relocate) | 6468211711 | kcbodepudi021@gmail.com | LinkedIn | Portfolio

APPLIED AI / ML ENGINEER

Applied AI / ML Engineer with experience building and deploying ML inference services, LLM-powered backends, and embedding-based retrieval systems for production workflows. Strong background in FastAPI-based APIs, NLP pipelines, and data pipelines, with emphasis on reliability, failure handling, and real-world performance. Contributed to applied ML systems that reduced inference latency by up to 25% and lowered malformed or partial responses by 30% through robust validation and batching strategies.

TECHNICAL SKILLS

Languages: Python, SQL
Machine Learning: PyTorch, Scikit-learn, Feature Engineering, Model Training & Evaluation
LLM & NLP Systems: Hugging Face Transformers, RAG Architectures, Embeddings, Vector Search
Inference & Deployment: FastAPI, REST APIs, Async Processing, Latency Optimization
MLOps: Docker, CI/CD (GitHub Actions)
Data & Storage: PostgreSQL, SQLite, MongoDB, Pandas, NumPy
Cloud & Systems: AWS, Linux, Git
Supporting / Familiar: FAISS, SQLAlchemy, Streamlit, Experiment Tracking, Model Monitoring

PROFESSIONAL EXPERIENCE

ML Engineer — Melotech — Remote, Saint Louis, MO
January 2024 – Present

- Implemented ML inference APIs using FastAPI that standardized model access across internal experimentation workflows, reducing setup overhead and enabling faster iteration cycles.
- Built embedding-based retrieval pipelines that improved the relevance and consistency of contextual data supplied to downstream ML and content workflows, reducing noisy retrieval during internal evaluations.
- Added input validation, response normalization, and fallback handling, reducing malformed or partial inference responses during testing by approximately 30%.
- Improved inference responsiveness by applying batching and response-size controls, resulting in 20–25% lower average latency during parallel internal evaluations.
- Analyzed model outputs across real media samples to identify semantic drift, low-confidence retrieval, and recurring failure patterns, contributing to targeted adjustments during iteration cycles.

Software Engineer — New Mek Solutions — Hyderabad, India
January 2022 – December 2023

- Developed and deployed ML-backed inference services using FastAPI and Docker, enabling consistent and reusable model access across internal NLP and analytics workflows.
- Built and maintained REST APIs to serve ML inference results, supporting concurrent internal requests and reducing ad-hoc model execution and manual testing overhead.
- Developed Python and SQL data pipelines for model training and evaluation, improving data preparation consistency and enabling repeatable experimentation across datasets ranging from tens of thousands to low millions of records.
- Implemented NLP pipelines using Hugging Face Transformers for document classification, summarization, and information extraction, enabling automated processing of unstructured data for downstream analytics use cases.
- Improved average API response times by 25–35% through async request handling and database query optimization.

PROJECTS

Persistent Memory Layer for LLM Applications — GitHub
Python | FastAPI | FAISS | SQLite | SQLAlchemy | LM Studio

- Built a persistent, task-scoped memory service for LLM applications to support long-term recall without cross-task or cross-user leakage.
- Implemented semantic retrieval using FAISS, indexing thousands of memory entries per user and returning only top-k relevant context.
- Validated namespace-based isolation through parallel request and multi-session testing.
- Developed an async FastAPI backend with durable SQLite persistence, ensuring consistent behavior across restarts and crash scenarios.
- Integrated local LLM inference via OpenAI-compatible APIs using LM Studio, separating chat and embedding workloads.
- Reduced average prompt size by 30–40% by decoupling conversational history from long-term memory retrieval.

Clinical Communication Memory System — GitHub
FastAPI | SQLite | Vector Embeddings | Semantic Search

- Built a visit-scoped semantic memory system for multilingual doctor–patient communication scenarios, focusing on strict data isolation requirements.
- Implemented UUID-based visit scoping across request handling and retrieval logic to prevent cross-patient data access.
- Added fail-safe retrieval behavior to ensure embedding or vector search failures returned empty results rather than incorrect data.
- Logged and audited semantic retrieval operations to support debugging, validation, and compliance-oriented review.
- Evaluated isolation guarantees under concurrent and adversarial request patterns to validate correct behavior under parallel access.

EDUCATION

Saint Louis University — St. Louis, MO
Master of Science in Information Systems — GPA: 3.90