ALEXIS C. MANDARIO

IT Professional | Machine Learning | Artificial Intelligence

Phone: +63 997 203 3371

Email: mandarioalexis@gmail.com

LinkedIn: https://www.linkedin.com/in/alyx-mandario-b546881a8/

GitHub: https://github.com/Alyxx-The-Sniper

Medium: https://medium.com/@kaikuh

Personal Statement

I am a detail-oriented IT professional with a passion for solving complex problems through scalable and innovative technology. With a strong background in machine learning, cloud infrastructure, and network administration, I thrive in fast-paced environments where learning never stops. My mission is to use technology to simplify life and empower organizations to reach new heights.

Education

- BS in Information Technology, ICCT Colleges Foundation
- Antipolo National High School
- Bagong Nayon I Elementary School

Certifications

- Cisco CCNP (ENCOR & ENARSI)
- CCNA (200-301)
- Linux System Administration
- Customer Service Excellence

Technical Skills

- Languages: Python 3
- Data Science: Exploratory Data Analysis, Hypothesis Testing, Regression, Classification, Decision Trees, Random Forest, Gradient Boost
- Machine Learning & AI: NLP, Computer Vision
- DevOps: CI/CD
- Cloud: AWS Lightsail, Web Deployment, S3, IAM
- Linux: Ubuntu, Kali, Red Hat
- Networking: CCNP, BGP, OSPF, VLANs
- Tools: Git, SAP, Jupyter Notebook

- AI Frameworks: Model Fine-tuning, LangChain, Unsloth, Ollama, LLMs

Selected Projects

Voice_Time_Capsule

Send messages to your future self and receive responses from your 'future you.'

Semantic_Games_Search

A semantic search engine for game titles using NLP and vector search (FAISS).

Productivity_Prediction

A machine learning model to predict employee productivity in the garment industry.

Publications:

- Outlier and Anomaly Detection
- Feature Engineering in ML

Achievements

- Published multiple ML articles on Medium
- Deployed models on Hugging Face

My Understanding of Creating an AI Chatbot Using RAG

First, planning the project is key-define what the AI agent will do (for example, troubleshoot a machine). Gather relevant datasets (PDFs, scraped Q&A from forums, etc.). Clean and structure this data into a standard format (ShareGPT, Alpaca, ChatML, etc.) using custom scripts, LangChain parsers, or tools like FireCrawl. Start small (10-50 samples) for prototyping and scale up once results are promising. Curate your dataset for quality: bad data = bad AI.

Embed your dataset and store it in a vector database (FAISS for simple cases, ChromaDB or Pinecone for scalability). For the backend (where RAG happens), write retrieval code to search for similar questions/answers using your vector DB (I use ChromaDB). Return the top k results to your chosen LLM, and set thresholds for fallback strategies (web search, human escalation).

Due to hardware limitations, I quantize models to 4-bit and use Ollama for loading. The main app is built with FastAPI, handles POST requests, and uses Streamlit for the frontend. When satisfied, wrap it all in Docker Compose. If not, improve data, model, or fine-tuning as needed.

For fine-tuning, I use Google Colab for free GPU, quantize to 4-bit, apply LoRA adapters, and use Unsloth for speed.

Here are some important fine-tuning parameters to consider:

- Sample size:
  Always start with a small sample size to ensure your setup works; scale up as results improve.

- Batch size per device:
  Depends on your hardware-typical values are 8, 16, or 32. Larger batches speed up training but require more memory.

- Gradient accumulation steps:
  This is the number of batches to process before updating the optimizer. Useful if your hardware can't handle large batch sizes.

- Max steps:
  The total number of optimizer updates (i.e., training steps). Controls how long your model trains.

- Warmup steps:
  The number of initial steps where the learning rate gradually increases from zero to the target value (commonly set to around 5 percent of max steps). Helps stabilize early training.

- Learning rate:
  Usually provided as a parameter (for example, 1e-4). Controls how quickly the model adapts during training.

- Weight decay:
  A regularization parameter that helps prevent overfitting by penalizing large weights during training. Typical values range from 0 (no regularization) to 0.1.

Hobbies & Interests

- Reading articles and blogging about AI and tech trends
- Creating cool AI projects
- Volunteering for local coding workshops
- Cosplay: I cosplay Yone from League of Legends; my first cosplay was Inosuke from Demon Slayer
- Favorite anime: Dragon Ball

- Favorite movie: Interstellar
- Favorite song: 'Who Am I' by Casting Crowns
- I love coffee-especially matcha (it-s healthy, don-t @ me!)
- Traveling: My favorite place in the Philippines is Atok, Benguet; I love cold places and stargazing
- I-ve been to Thailand-EDM music festivals like Songkran were amazing!
- I love Thai food (Pad Krapao, Pad Thai, steamed chicken with yellow rice, Teh Tarik)
- Jollibee spaghetti is my comfort food
- Next travel goal: Sapa, Vietnam
- I can do TikTok dance-actually, I-m pretty good at it!