Zeyad El-Sayed

AI / ML Engineer

zeyadusf | zeyadusf | huggingface.co/zeyadusf | ziayd.usf@gmail.com | +20 10 2350 4597

Military Service: Completed

Technical Summary

AI/ML Engineer with hands-on experience designing and deploying production-grade NLP pipelines, Retrieval-Augmented Generation (RAG) systems, and computer vision solutions. Skilled in fine-tuning large language models using PEFT techniques (LoRA, QLoRA), building end-to-end deep learning architectures (Transformers, CNNs, GANs), and delivering scalable backend APIs with FastAPI and Docker.

Skills

Programming Languages: Python, C++, SQL
Frameworks & Libraries: PyTorch, Keras, HuggingFace Transformers, Scikit-learn, Pandas, NumPy, OpenCV
LLM Engineering: Retrieval-Augmented Generation (RAG), LangChain, Prompt Engineering, Fine-Tuning (PEFT)
Deep Learning: CNNs, RNNs, LSTMs, Transformers, Autoencoders, GANs
Computer Vision: Image Classification, Image Denoising, GAN-based Synthesis
Backend & APIs: FastAPI, RESTful APIs, Django (Basic)
DevOps & MLOps: Docker, GitHub Actions (CI/CD), Model Deployment, Railway, Cloudflare
Data Engineering & BI: SQL, Tableau, Power BI, Excel
Developer Tools: Git, Jupyter Notebook, VS Code, Linux

Experience

Information Technology Institute (ITI) Jul 2023 – Sep 2023

Business Intelligence Trainee *Egypt*

- Mastered SQL fundamentals, relational data modeling, and data warehouse design patterns.
- Designed and delivered interactive dashboards using **Tableau** and **Power BI**, translating raw datasets into actionable business insights.
- Applied BI tooling to real-world analytics projects covering KPI tracking, trend analysis, and reporting.
- Gained foundational experience in ETL pipelines and dimensional modeling (star/snowflake schemas).

Projects

TesseractRAG | *FastAPI, FAISS, BM25, BGE Embeddings, HuggingFace, Docker, Backblaze B2* GitHub | Live Demo

- Architected a fully deployed production RAG system — FastAPI backend (Railway) + frontend (Cloudflare Workers) with multi-session document Q&A and persistent storage via Backblaze B2.
- Implemented hybrid retrieval (BM25 + FAISS + RRF fusion) with cross-encoder reranking (ms-marco-MiniLM-L-6-v2) for high-precision context selection.
- Integrated **Llama-3.1-8B-Instruct** via HuggingFace Inference API with rule-based query routing and 3000-char context budget management.
- Engineered memory-efficient deployment under a 512 MB RAM constraint using lazy-loaded reranker, serialized FAISS indices (BytesIO), and on-demand BM25 rebuilding.

Catch the AI | *Python, Django, React, PostgreSQL, ViT, DeBERTa, Wav2Vec 2.0* GitHub | Website

- Built ViT and EfficientNet models to detect AI-generated images with **96%** accuracy for media verification.
- Built an NLP-based detector (DeBERTa, RoBERTa) with **95%** accuracy in spotting AI-generated text.
- Implemented Wav2Vec 2.0 for speech detection, achieving **90%** accuracy in identifying AI-generated audio.

Fine-Tuning LLMs | *PyTorch, PEFT techniques, HuggingFace, NLP* GitHub | HF Spaces

- Curated a fine-tuning repository and deployed models on Hugging Face Spaces for text classification, generation, and summarization.
- Applied LoRA and QLoRA PEFT techniques to reduce trainable parameters by up to 99% while maintaining performance.
- Evaluated models using ROUGE-L, BLEU, and additional metrics, achieving 95%+ performance across tasks.

LLMs from Scratch | *PyTorch, NLP, Transformers* GitHub

- Implemented a Large Language Model from scratch — tokenization, multi-head self-attention, positional encodings, and full transformer blocks in pure PyTorch.
- Built end-to-end training and evaluation pipelines, deepening understanding of modern NLP architecture internals.

Imbalanced Data Treatment | *Python, imbalanced-learn, Scikit-learn, Matplotlib* GitHub

- Benchmarked KNN, RF, SGD, XGBoost, and DT classifiers on imbalanced datasets before/after applying SMOTE, TomekLinks, and hybrid sampling strategies.
- Achieved measurable improvements in recall and F1-score, demonstrating the impact of proper data balancing on production model reliability.

Credit Score Classification | *Python, PyCaret, Tableau, Scikit-learn* GitHub

- Cleaned and analyzed data with heavy missing/incorrect values using statistical methods and Tableau for visualization.
- Used PyCaret to train and compare models (Extra Trees, RF, XGBoost, KNN, DT), building a solid preprocessing pipeline.

Education

Benha University Oct 2020 – Jun 2024

B.S. in Computers & Artificial Intelligence *GPA: 3.4 / 4.0*

Languages

Arabic: Native | **English:** Professional working proficiency (reading, writing, technical communication)