# Aryan Pagaria

Chennai, India | aryanpagaria7@gmail.com | GitHub | LinkedIn

## Summary

Artificial Intelligence undergraduate with project experience in PyTorch model training, parameter-efficient LLM fine-tuning, and retrieval-augmented generation. Built modular training, evaluation, and inference pipelines, with an emphasis on reproducible experimentation. Research interests include memory-efficient language models, reliable generative AI, and research software engineering.

## Education

**SRM Arts and Science College** Chennai, India
Bachelor of Science in Artificial Intelligence 2025 – 2028

## Technical Skills

**Programming:** Python, SQL, Bash

**Machine Learning:** PyTorch, Scikit-learn, deep learning, regression, model training and evaluation, uncertainty estimation

**LLMs and Retrieval:** Hugging Face Transformers, PEFT, LoRA, QLoRA, 4-bit quantization, RAG, LangChain, FAISS

**Experimentation:** Dataset validation, tokenization, gradient accumulation, checkpointing, learning-rate scheduling, evaluation pipelines

**Software and Infrastructure:** FastAPI, REST APIs, Docker, Git, GitHub, GitHub Actions, Linux, AWS

## Experience

**SpazorLabs** Chennai, India
*Artificial Intelligence Intern* September 2026 – Present

- Contribute to production-oriented LLM pipelines and AI agent systems, including multi-step workflows and API/tool integration.
- Work on agent orchestration, context and state management, prompt engineering, and structured outputs.
- Support evaluation and observability while experimenting with models and inference strategies to assess reliability and performance.

## Selected Projects

**Parameter-Efficient LLM Fine-Tuning for Financial QA** 2026 | GitHub
*Python, PyTorch, Hugging Face Transformers, PEFT, Qwen2.5-3B*

- Fine-tuned Qwen2.5-3B-Instruct on FinQA using QLoRA, combining 4-bit quantization with LoRA adapters for parameter-efficient adaptation to financial question answering.
- Built a reproducible pipeline for dataset validation, instruction formatting, tokenization, gradient accumulation, learning-rate scheduling, and checkpointing.
- Implemented training, inference, and evaluation modules with Exact Match, Normalized Exact Match, and Numerical Accuracy, including checkpoint recovery for interrupted experiments.

**Document Question Answering with RAG** 2026 | GitHub
*Python, Hugging Face, FAISS, FastAPI, Docker*

- Built a retrieval-augmented generation pipeline for answering questions over uploaded documents, combining document chunking, embedding generation, and FAISS similarity search.
- Developed FastAPI endpoints for document ingestion and question answering, separating preprocessing, retrieval, and inference into modular components.
- Containerized the application with Docker to provide a reproducible local execution environment.

**Jet Mass Regression with Uncertainty Estimation** 2025 | GitHub
*Python, PyTorch, CMS Open Data*

- Trained deep learning regression models to predict jet mass from CMS Open Data; extended the models with negative log-likelihood loss to estimate predictive uncertainty.
- Evaluated prediction quality and uncertainty calibration using RMSE, pull distributions, and prediction coverage.
- Organized data processing, training, evaluation, and visualization into modular pipelines for reproducible experimentation.

## Independent Research

**Modus_X: Memory-Efficient Language Modeling** June 2026 – Present

- Exploring memory-efficient recurrent architectures for language modeling as alternatives to attention-based context storage.
- Designing controlled experiments comparing recurrent and Transformer baselines on memory retention, associative recall, language-model quality, and inference-memory usage.

## Achievements

**Codex Hackathon Finalist, SRM:** Built and shipped a working prototype within 24 hours and advanced past the qualifying round.