Samarth Pratap Singh

Pratapgarh, U.P.
+91-9452026413 | samarthsin2006@gmail.com | Linkedin | Github | LeetCode | Hugging Face

EDUCATION

VIT Bhopal University, Bhopal **2023 – 2027**
B.Tech - Computer Science and Engineering - ***CGPA - 8.45*** *Bhopal, Madhya Pradesh*

COURSEWORK

- DSA
- OOP Concepts
- Cloud Computing
- DBMS
- Operating Systems
- Computer Networks
- Software Engineering

PROJECTS

DoCopilot (RAG Document Q&A) | Next.js, FastAPI, Qdrant (Hybrid), Reranking, Groq **Dec 2025**

- Built a full-stack RAG app to upload PDFs/TXT, index content using Qdrant hybrid search (BM25 + dense vectors), and answer questions with citations.
- Improved retrieval quality via RRF fusion + cross-encoder reranking; selected top-k contexts before generation to reduce noise.
- Added guardrails (prompt-injection detection, PII redaction, source-grounding checks) and an evaluation + ablation study with LLM-as-Judge; reported 89.2% avg correctness, 90.5% relevance, 100% source rate, and 2.86s avg latency on 40/40 queries.

FLAN-T5 Dialogue Summarizer (GenAI) | PEFT, LoRA, Transformers, Gradio **Oct 2025**

- Fine-tuned FLAN-T5-base with LoRA on SAMSum (14.7K dialogues), achieving 49.01 ROUGE-1, 72.25 BERTScore F1, and 42.51 METEOR.
- Implemented parameter-efficient training (LoRA r=16, α=32), updating only 2% of parameters with FP16 mixed precision for faster, memory-efficient training.
- Deployed an interactive Gradio app on Hugging Face Spaces with configurable decoding (beam search, max length) and published the model with reproducible evaluation (ROUGE, BERTScore, METEOR, BLEU).

RoBERTa for Banking Intent Classification(Banking77) | PyTorch, Transformers,CUDA **Sep 2025**

- Fine-tuned RoBERTa-base on Banking77 (77 intents, 13k queries) to 93.7% accuracy and 93.6% macro-F1 on a held-out split, with per-epoch tracking and best-checkpoint selection for robust evaluation.
- Implemented standard transformer fine-tuning with AdamW and weight decay (2e-5 LR; batch 16/32; 5 epochs), using FP16 on a GPU-accelerated Linux environment for efficient training.
- Added experiment hygiene: fixed seeds, consistent tokenization/padding, and epoch-level metrics; verified inference with a minimal loading script to streamline reviewer replication.

Project Loom | Next.js, TypeScript, Sanity.io, NextAuth.js **Jan 2025**

- Engineered a full-stack project-sharing platform using Next.js, leveraging Server-Side Rendering (SSR) and Incremental Static Regeneration (ISR) to decrease initial page load times by 50%.
- Architected a scalable backend with the Sanity.io headless CMS, designing content models to efficiently manage and serve over 1,000 project entries and user profiles.
- Implemented secure user authentication with NextAuth.js and a PostgreSQL database, enabling users to manage profiles, post projects, and interact with content.

TECHNICAL SKILLS

Languages: Python, C++, JavaScript/TS, SQL
ML/DL: PyTorch, TensorFlow, Scikit-learn, CNN/LSTM, Transformers, PEFT/LoRA
LLMOps/MLOps: FastAPI, LangChain, FAISS, Qdrant, LangSmith, MLflow, Weights & Biases
Production: Gradio/Streamlit, Hugging Face, Docker, AWS
Full-Stack: Next.js, Node.js, PostgreSQL, NextAuth.js, Sanity.io
Tools: Linux/CUDA, Git/GitHub

CERTIFICATIONS

- Applied Machine Learning in Python - Coursera