Saurabh Vishwakarma

Chandigarh, India
+91-9695877472 sv9327545@gmail.com LinkedIn GitHub Portfolio

Summary

ML Engineer & Python Developer — building and deploying end-to-end AI systems using PyTorch, FastAPI, and Docker. IIT Madras BS Data Science.

Project Experience

LeafLens — AI Image Super-Resolution Platform *Deployed · 2026*
Python, PyTorch, FastAPI, Docker Demo

- Architected and deployed an SRGAN-based super-resolution system upscaling **32×32 leaf images to 128×128** (4× factor) using a 16-block residual generator with PReLU activations, Batch Normalization, and PixelShuffle upsampling; live at `plant-super-resolution.onrender.com`
- Implemented **8-variant Test-Time Augmentation (TTA)** — 4 rotations + horizontal flip with reverse-transform averaging — reducing prediction artifacts and improving output consistency at inference time
- Built a production **FastAPI inference backend** exposing a `/predict` endpoint with `/health` monitoring; containerized end-to-end with Docker for reproducible cloud deployment on Render
- Designed a **zero-dependency frontend** (plain HTML/CSS/JS) with real-time upload, TTA toggle, and one-click download of enhanced output — full-stack ownership without framework overhead

ATS Intelligence Engine *2026*
Python, FastAPI, Sentence Transformers, PyMuPDF, Docker Source Code

- Built an AI-powered resume screening system outputting a **blended ATS score** by combining cosine similarity from Sentence Transformer embeddings with rule-based skill overlap; served via FastAPI behind an **Nginx reverse proxy**
- Designed a **dual-input REST API** supporting raw text and PDF uploads via two endpoints (`/predict/ats` and `/predict/ats/upload`), with PyMuPDF handling unstructured PDF extraction across varied resume formats
- Structured codebase with clean separation of concerns (`services/scorer.py`, `utilities/pdf_parser.py`, `schemas.py`) and deployed with **Docker Compose** orchestrating FastAPI backend and Nginx frontend in separate containers
- Integrated **HuggingFace Hub** for model serving — download, load, embed, and score at inference time with no retraining required

Language Modeling using GRU *2025*
Python, PyTorch Source Code

- Implemented a **complete GRU architecture from first principles** using raw PyTorch tensors — manually coding update gate, reset gate, candidate hidden state, and output layer — without using `nn.GRU`, to deeply understand sequential learning mechanics
- Trained on real **Amazon review text data**: full preprocessing pipeline (tokenization → vocabulary construction → sliding window input-target pairs) and a trainable **embedding matrix** learned via backpropagation; used CrossEntropyLoss + manual SGD with BPTT to generate coherent, sentiment-aware text

House Price Prediction Platform *Deployed · 2025*
Python, Pandas, NumPy, Scikit-learn, Flask Demo

- Delivered a **full regression ML pipeline** from raw data to deployed web app — EDA, feature engineering, model selection, RMSE evaluation with cross-validation, and Flask-based real-time inference
- Applied **multicollinearity detection, correlation analysis, and residual diagnostics** during EDA to validate linear model assumptions and guide feature selection before training

Education

Chandigarh University *2023 – 2026*
BCA — CGPA: 8.03

IIT Madras *2023 – 2027*
BS in Data Science — CGPA: 7.2

Technical Skills

Languages & Tools	Python, C++, SQL, Git/GitHub, Linux, Docker
ML/DL Frameworks	PyTorch, TensorFlow, Scikit-learn, Sentence Transformers, Hugging Face Transformers, TorchVision
Deep Learning & AI	CNNs, GANs, Computer Vision, NLP, LLMs, Transfer Learning, Image Super-Resolution, Sequence Modeling
MLOps & Deployment	FastAPI, Flask, Docker Compose, Nginx, REST APIs, Model Serving, Render, Uvicorn
Data Science	Pandas, NumPy, Matplotlib, Seaborn, OpenCV
Databases & Cloud	PostgreSQL, MySQL, AWS (Basics)