Sanchit Lamba

8826908248 | mail@sanchit.cc | linkedin.com/in/sanchit-lamba | github.com/sanchit-lamba

EDUCATION

Manipal University Jaipur — Jaipur, RJ
Bachelors of Technology in Computer Science and Engineering, Hons. in AI and ML — *Aug. 2022 – May 2026*

EXPERIENCE

Python Developer — September 2024 – Present
Carbon Crunch — *Hybrid*
- Leading the development of a platform to integrate data from multiple financial and environmental reports released by listed companies.

Technical Instructor — January 2023 – April 2023
Camp K12 — *Remote*
- Taught basics of python OOP to students ranging from clsses 8th to 12th on an hourly rate.

Freelance — June 2023 – Present
Multiple clients — *Remote*
- Freelance work for a few clients in field of web development and animation.

PROJECTS

State load prediction[Smart India Hackathon finale] | *M,Python,matplotlib,numpy* — December 2024
sanchit.cc/sih — github.com/sih
- Gathered and merged data from State Load Dispatch Center and Grid India reports.
- Utilized Power Query M language for data retrieval and performed extensive data cleaning and preprocessing.
- Tested various models, including ARIMA and SARIMA, with SARIMA achieving the best results with specific parameter values.

DocAssist-LLM | *Python, Qdrant, LangChain, Tensorflow, Numpy* — October 2023 – November 2023
sanchit.cc/dallm — github.com/dallm
- The project implemented a Retrieval Argumented Generation(RAG) based approach to use internal/non public documentation alongside LLMs trained on regular data
- In the scope of the project we used Qdrant to form a vector database of the two javascript books and entirety of it's documentation which was scraped from their site.
- Each query asked by the user was ran through the Qdrant database and the returned results were ran through an LLM with the initial query and supporting results from the documentation.
- The LLM would then spew out a result relavent to the query and all this is possible even if the documentation isin't a part of the initial training data

Autonomous Vehicle Guidance System | *Python, CARLA, Keras, Tensorflow* — October 2023 – November 2023
sanchit.cc/avgs — github.com/avgs
- Ingestion of data from the simulated lidars and cameras was done using a simple numpy array and opencv respectively.
- The RGBA data from OpenCV was then passed onto a CNN that simulated the camera data gathered from the car and the surrounding enviroment.
- A reinforcement learning model was fed all of the pre-processed input data, which included LiDAR point clouds and visual features from the CNN. By processing the combined sensor inputs, this model made used Proximal Policy Optimization (PPO), which allowed the system to continuously interact with the simulated environment and learn the best driving behaviors.
- The reinforcement learning model was designed with a reward system that penalized crashes and dangerous driving behaviors while periodically rewarding safe navigation and adherence to traffic rules.

TECHNICAL SKILLS

Languages: Python
Developer Tools: Git, Poetry, Docker, Azure, Bash, GNU/Linux, Nixos
Libraries: pandas, NumPy, Matplotlib, Tensorflow, OpenCV, Matplotlib, Keras, PyGame, Arelle