Saransh Surana

+1 (631) 816-8344 | ssurana818@gmail.com | linkedin.com/in/saransh-surana | Portfolio | Open to Relocation

EDUCATION

SUNY - Stony Brook University **Aug 2023 – May 2025**
Master of Science in Data Science *Stony Brook, NY, USA*
Relevant Courses: Machine Learning, Data Analysis, Data Management, DSA, LLMs, Statistics, and Cloud Computing.

SKILLS

- **Languages**: Python, R, Java, SQL, Bash, C/C++, SAS, NoSQL, Ocaml
- **Frameworks & Tools**: TensorFlow, Keras, PyTorch, Docker, Flask, Kafka, PySpark, MLflow, Jenkins, Kubernetes, Agile
- **Libraries**: NumPy, Pandas, Scikit-learn, SciPy, NLTK, OpenCV, FAISS, CUDA, Matplotlib
- **Databases**: MySQL, MongoDB, Oracle, Hadoop, Hive, BigQuery
- **Statistics & ML**: Regression, Classification, A/B Testing, Time Series, Bayesian Inference, ANOVA, Chi-square, SHAP, Statistical Modeling, Hypothesis Testing, SPSS
- **Software**: Power BI, Tableau, Weights&Biases, Jupyter, Grafana, Excel, PowerPoint, MapReduce
- **Cloud**: Google Cloud Platform (GCP), Microsoft Azure

WORK EXPERIENCE

Schizophrenia & Psychosis Action Alliance **Jul 2025 – Present**
AI Software Research Volunteer Remote, USA
- Scraped behavioral health housing data from multiple public sources using Playwright, generating semi-structured JSON, then processed it with **Gemini Pro** and **regex** to create fully structured datasets.
- Designed **prompt flows**, handled **async API orchestration**, and applied logic to ensure **county-level relevance**.

Stony Brook University **Jan 2025 – May 2025**
Research Assistant Stony Brook, NY, USA
- Built **Python** pipelines to clean and preprocess unstructured data from **web pages**, **PDFs**, and other raw formats, version-controlled with **Git** for reproducibility and collaboration.

Ford Motor Company **May 2024 – Aug 2024**
Data Science Intern Dearborn, MI, USA
- Used **BigQuery** and **SQL** on **GCP** to extract and transform 45GB+ manufacturing sensor data, enabling large-scale anomaly detection for predictive maintenance workflows.
- Trained **unsupervised models** (**Isolation Forest**, **One-Class SVM**) to detect anomalies in manufacturing sensor data, achieving **78% recall** and **73% precision**, supporting early fault detection.
- Explained model results to **technical and non-technical teams** and engaged with **data science experts** to learn more about the field, supporting fault resolution and alignment.

Napuor **Aug 2022 – Jan 2023**
Data Science Intern Bengaluru, KA, India
- Developed real-time **demand forecasting** and **inventory optimization** by deploying **XGBoost** models on **GCP** using **FAST API** and **Docker**, reducing forecast error by **18%** across **30+ SKUs**.
- Designed end-to-end **ML data pipelines** on unstructured data and **SQL-based ETL workflows** using **Spark**, **Hive**, and **Kafka**, accelerating deployment time by **40%** and supporting analysis of **1K+ events daily**.
- Drove 30% **marketing ROI** uplift by applying **clustering** on 10K+ customer profiles, enabling business teams to target high-value segments effectively.
- Conducted **A/B testing** on promotional strategies and new product placements across multiple regions, identifying winning variants that increased sales conversion by **7%**.

PROJECTS

AI-Powered Mental Health Chatbot for Personalized Support | *Prompt Engineering, Gemini Pro, FAISS, RAG*
- Developed a mental health chatbot using RAG with FAISS search and Gemini Pro for context-aware responses.
- Used prompt engineering and few-shot learning to improve response relevance, while filtering unsafe completions.

CNN vs Vision Transformer: CIFAR-10 Benchmarking | *Python, TensorFlow, PyTorch, NumPy, Pandas*
- Benchmarked ResNet50 and ViT on CIFAR-10 using transfer learning; ViT achieved 91% accuracy with faster convergence and stronger generalization

Reinforcement Learning Book Recommender | *Python, PyTorch, Stable-Baselines3, Gymnasium, Pandas, Matplotlib*
- Designed a PPO-based recommender in a custom Gymnasium environment with baselines and full evaluation metrics.
- Achieved CTR@1-per-step of 0.025 vs 0.007 for Random, with reproducible preprocess → train → eval pipeline.