

# INDRANEEL SARODE

San Jose, CA

☎ +1-669-360-7809 ✉ [indraneelsarode@gmail.com](mailto:indraneelsarode@gmail.com) 🔗 [linkedin.com/in/indraneel-sarode-587178206](https://www.linkedin.com/in/indraneel-sarode-587178206) 🌐 [github.com/neels22](https://github.com/neels22)

🌐 [indraneelsarode.com](https://indraneelsarode.com)

## Summary

MS Software Engineering @ SJSU (Dec 2026). Data Analytics intern candidate focused on SQL + Python, data modeling (facts/dimensions, star schema), and Power BI (basics). Built analytics views and ETL scripts in projects; currently leveling up on Snowflake concepts (staging → bronze/silver/gold) and Azure data tooling (ADF, ADLS).

## Education

### San Jose State University

Master of Science in *Software Engineering* GPA: 3.56/4.0

Jan 2025 – Dec 2026

San Jose, CA, USA

### University of Pune

Bachelor of Engineering in *Computer Engineering* GPA: 3.87/4.0

Aug 2020 – Jun 2024

Pune, India

## Technical Skills

**Analytics & BI:** Power BI, Excel, Matplotlib

**Warehousing & SQL:** **Snowflake** (star schema, fact/dim), PostgreSQL/MySQL; SQL (CTEs, window functions, aggregates), performance basics

**Languages & Libraries:** Python (**pandas**, NumPy), JavaScript/TypeScript; FastAPI, Node.js/Express, Next.js/React

**APIs & Integration:** REST, Webhooks, OAuth2, JWT/RBAC; Power BI, OpenAPI/Swagger

**Cloud, Ops & Observability:** Docker, CI/CD (GitHub Actions), Azure (ADF, ADLS, Functions, Logic Apps), AWS (EC2, S3, Lambda, ECS), Grafana, Prometheus, OpenTelemetry (basic)

## Internships

### Software Engineer Intern

Coditas Solutions LLP

Jan 2024 – Jun 2024

Pune, India

- Shipped a **LangChain**-based **RAG** service for **PDF/YouTube/web** with source-linked citations; on a seeded corpus, median answers consistently included **1 relevant citation**.
- Evaluated retriever settings (chunk size/overlap, vector DB) on a seeded Q&A set; calculated simple **precision/recall** and recommended a **hybrid retrieval** default in staging.
- Added guardrails to the **NL→SQL** path (read-only checks, linting/retries) and tracked failure types; built a small report (counts by error class, top schemas) that reduced malformed queries during testing.
- Built a lightweight **Next.js** dashboard to visualize request counts, latency bands, and citation-rate trends so stakeholders could self-serve instead of filing ad-hoc report tickets.
- Versioned the **FastAPI** service (v1→1.1) and added basic request/DB-connection metrics; compared before/after latency with SQL/pandas to verify improvements under like load.

## Projects

### Fake News Detection | *Python, pandas, NumPy, scikit-learn, Hugging Face Transformers, NLTK/spaCy*

2025

- Built an **NLP pipeline** to classify fake vs. real news from Kaggle/WelFake; cleaned text and created balanced train/val/test splits.
- Engineered TF-IDF features and fine-tuned **BERT**; benchmarked against Logistic Regression, Naïve Bayes, and Random Forest.
- Measured precision, recall, F1, and ROC-AUC with stratified k-fold CV and a held-out test; improved **minority-class recall** using class weights/SMOTE.
- Ran error analysis (satire, clickbait, domain shift) and built a concise dashboard (confusion matrices, PR/ROC curves) with deployment guidance.

### A/B Testing on E-commerce Platform | *Python, SQL, Spark, Excel/Power BI*

2025

- Designed and ran controlled **A/B tests** on product page layout; defined hypotheses and success metrics (CTR, add-to-cart, conversion).
- Built clean experiment datasets (exposure, assignment, outcomes) with **SQL/Spark**; added pre-checks for SRM and metric sanity.
- Analyzed results with precision/recall where relevant, lift, p-values, and **95% CIs**; segmented by device/source to uncover heterogeneous effects.
- Delivered a concise results dashboard (Excel/Power BI) and recommendations; **implemented the winning variant**, improving engagement and conversion rates.

### Perplexity-2.0 Link | Demo | *Next.js, TypeScript, FastAPI, LangGraph, Google Gemini, SSE*

2025

- Architected a **AI chatbot** with Server-Sent Events for progressive, word-by-word **streaming**; added **stage indicators** (search → read → write) to reduce perceived latency.
- Built an agentic **LangGraph** workflow with conditional tool-calling; integrated **Tavily web search** + **Gemini** to auto-generate queries and synthesize source-grounded answers.
- Implemented stateful conversation with MemorySaver checkpointing across sessions; exposed a **FastAPI REST API** with streaming + CORS and containerized it for deployment (**Docker→Render**).