

Projects - LinkedIn Copy/Paste Descriptions

Srijan Gupta | Generated February 12, 2026

Each entry includes impact bullets and a skills/tools list.

1. ClearClause - Production Legal RAG System (Hybrid Search + Reranking)

- Built a production-grade legal retrieval + grounded Q&A system for contracts using hybrid search (semantic + lexical/BM25) with reranking for high-precision passage selection.
- Achieved 0.87 precision@10 and reduced p95 latency 93% (4.3s -> 300ms) using Redis semantic caching + response streaming.
- Deployed on Kubernetes with HPA (3-15 replicas) to support 200+ QPS, with observability instrumentation and production reliability guardrails (PII redaction, structured validation).

Skills / Tools: Python, FastAPI, pgvector, Redis, BM25, embeddings, reranking, response streaming, Docker, Kubernetes (HPA), OpenTelemetry/Prometheus/Grafana, PII redaction

2. BuckyConnect - Real-Time Collaboration Platform (WebSockets + GraphQL)

- Built a real-time collaboration platform using WebSockets for live updates and GraphQL for structured reads, with Redis Pub/Sub for event fanout across instances.
- Supported 280+ concurrent users, sustained 1,000+ events/sec, and maintained <500ms latency targets under real-time load.
- Deployed cloud-native infrastructure on AWS ECS + CloudFront + DynamoDB with autoscaling; improved UX performance via code splitting, lazy loading, and Web Workers (40% bundle reduction).

Skills / Tools: React, Node.js/TypeScript, WebSockets, GraphQL, Redis Pub/Sub, AWS ECS, CloudFront, DynamoDB, autoscaling, frontend performance optimization

3. Spotify Music Popularity Pipeline - ML Training + Serverless Inference

- Built a reproducible ML pipeline to predict track popularity using tuned ensemble models (XGBoost/LightGBM + stacking) with Optuna hyperparameter optimization (100+ trials).
- Improved accuracy by 45% and achieved MAE ~10.0 (0-100); processed 1M+ records end-to-end.
- Deployed inference on AWS Lambda with MLflow model versioning and a Redis-backed feature store; reduced latency ~30% via parallelization/provisioned concurrency.

Skills / Tools: Python, XGBoost, LightGBM, scikit-learn, Optuna, MLflow, AWS Lambda, Redis, feature engineering, pipeline automation, Docker

4. Real-Time Fraud Detection API - Low-Latency ML Inference + Drift Monitoring

- Built a production fraud scoring API using FastAPI + Redis caching, deployed on Kubernetes with autoscaling and circuit-breaker reliability patterns.
- Trained models on 284K transactions, handling class imbalance with SMOTE + ensembles; achieved 96% precision / 89% recall.
- Sustained 500+ QPS with ~80-100ms p95 latency, backed by Prometheus/Grafana monitoring, Jaeger tracing, and drift tracking + retraining triggers.

Skills / Tools: Python, FastAPI, scikit-learn/XGBoost, SMOTE, Redis, Kubernetes, Prometheus, Grafana, Jaeger, CI/CD (GitHub Actions), Trivy, MLOps monitoring

5. E-Commerce Backend System - REST API Performance + Load Testing

- Built FastAPI REST services backed by PostgreSQL and improved read performance using indexing + connection pooling.
- Implemented Redis cache-aside and rate limiting for stable performance under concurrency.
- Achieved ~35% latency reduction and ~60% DB load reduction, validated via Locust load tests simulating 1,000+ concurrent users.

Skills / Tools: Python, FastAPI, PostgreSQL, Redis, indexing, connection pooling, rate limiting, Docker, pytest, Locust, performance benchmarking

6. Algorithmic Trading Strategy Backtest - Momentum + Walk-Forward Validation

- Built a research-grade backtesting engine in C++ with Python analysis tooling and PostgreSQL storage for systematic momentum strategies.
- Applied walk-forward validation and execution realism (slippage + volatility-adaptive sizing) to reduce overfitting risk.
- Achieved 18.2% annualized return, 1.47 Sharpe, outperformed SPY by 820 bps; accelerated simulation throughput 3x (15s -> 5s) for 100K+ sims using OpenMP + SIMD.

Skills / Tools: C++, Python (NumPy/pandas), PostgreSQL, walk-forward validation, backtesting, OpenMP, SIMD, time-series evaluation, quantitative research

7. Options Pricing & Greeks Engine - Black-Scholes + Monte Carlo + Calibration

- Implemented an options pricing and risk engine supporting Black-Scholes and Monte Carlo (10K paths), with Greeks via finite differences.
- Calibrated volatility surfaces using Newton-Raphson for real-time risk analysis workflows.
- Achieved 99.8% accuracy vs Bloomberg, Greeks <0.1% error, <2s runtimes; delivered 85% performance optimization using SIMD + variance reduction.

Skills / Tools: C++, Eigen, Boost, Python (NumPy/SciPy), Monte Carlo simulation, finite differences, Newton-Raphson, volatility surface calibration, SIMD, variance reduction

8. Statistical Arbitrage Pairs Trading - Cointegration + Kalman Filters

- Built a pairs trading pipeline identifying 200+ equity pairs via Johansen cointegration and trading mean reversion with z-score signals.
- Used Kalman filtering to estimate dynamic hedge ratios under changing market regimes.
- Achieved 65% win rate and 2.1 profit factor; engineered a low-latency system processing 50K+ ticks/sec with <100ms signal-to-order latency and degradation alerts.

Skills / Tools: Python (statsmodels), R, PostgreSQL/TimescaleDB, Redis, Johansen cointegration, Kalman filters, low-latency pipelines, monitoring/alerting

9. Healthcare Spending vs Life Expectancy - ETL + Fixed Effects + Tableau Storybook

- Built a reproducible pipeline ingesting 150+ state-level indicators (2020-2023), cutting data prep 2 days -> 2 hours via automated validation and standardized joins.
- Fitted multivariate models with region fixed effects and robustness checks; quantified interpretable relationships between per-capita spending and longevity outcomes.
- Published an interactive Tableau storybook with coefficient plots and drill-downs by region, income quintile, and mortality-linked categories for stakeholder-ready analysis.

Skills / Tools: Python (pandas/statsmodels), R (fixest/plm/did variants), ETL automation, regression modeling, fixed effects, robustness checks, Tableau, data storytelling

10. Customer Churn Prediction & Intervention - Explainable ML + Playbook

- Built churn models across 500K+ customers using XGBoost/scikit-learn, achieving 87% precision / 82% recall.
- Added SHAP explainability to surface drivers and prioritize interventions; modeled time-to-churn using Cox proportional hazards in a consulting variant.
- Reduced churn 12% -> 7% and improved operational speed with workflows that cut time-to-intervention 14 days -> 2 days.

Skills / Tools: SQL (PostgreSQL), Python, XGBoost, scikit-learn, SHAP, Tableau, survival analysis (Cox), intervention design, KPI monitoring

11. Retail Demand Forecasting System - SKU Forecasts + External Regressors

- Built demand forecasts for 50+ SKUs using Prophet/ARIMA and ensemble methods, incorporating external drivers (holidays, promotions, weather).
- Achieved 8.2% MAPE (~40% better than baseline) through feature + model iteration and evaluation discipline.
- Delivered dashboards with forecast uncertainty bands to support safety-stock and scenario planning for decision-makers.

Skills / Tools: Python (Prophet, ARIMA, XGBoost), R, time-series modeling, feature engineering, Tableau, uncertainty bands, scenario analysis

12. Retail Demand Elasticity Analysis - ARIMA/VAR + Structural Breaks

- Modeled demand dynamics using ARIMA + VAR and validated long-run relationships with cointegration diagnostics.
- Tested stability using Chow structural break tests to avoid fragile elasticity estimates under regime changes.
- Used impulse response functions to run counterfactual simulations and communicate how shocks propagate through demand over time.

Skills / Tools: Python (statsmodels), R (forecast/vars), ARIMA, VAR, cointegration, Chow tests, impulse response functions, time-series diagnostics

13. E-Commerce Conversion Optimization - A/B Testing + Power + Segmentation

- Designed and analyzed A/B tests over 50K+ users, including power analysis and covariate balance checks for statistical rigor.
- Measured 2.2% conversion lift ($p < 0.05$, 95% CI) and quantified heterogeneity (e.g., 3.5% lift on mobile).
- Built automated reporting that reduced repetitive manual analysis by 95% and packaged insights in stakeholder-ready dashboards.

Skills / Tools: Python (SciPy/stats), SQL, Tableau, A/B testing, power analysis, covariate balance, segmentation, confidence intervals, reporting automation

14. Product Metrics & Analytics Framework - Funnels + Health Metrics

- Built a product analytics framework tracking 15+ health metrics across 10K+ users, including funnel views and onboarding breakdowns.
- Reduced time-to-insight 3 days -> 10 minutes by standardizing metric definitions and building dashboards for repeat analysis.
- Identified onboarding drop-offs and guided roadmap decisions that drove +15% engagement after iteration.

Skills / Tools: SQL, Python, Tableau, funnel analysis, KPI design, dashboarding, metric definition, product analytics

15. Randomized Controlled Trial Evaluation - LATE via 2SLS + Heterogeneity

- Analyzed a stratified RCT with 50K+ participants, including power calculations (MDE=1.5%) and compliance-aware estimation.
- Estimated causal impacts using LATE via 2SLS with clustered standard errors for robustness.
- Quantified heterogeneity using causal forests (e.g., 3.5% mobile vs 1.1% desktop) and delivered reproducible, policy-ready results.

Skills / Tools: Python (statsmodels), R (estimatr/randomizr), causal inference, 2SLS, LATE, clustered SE, power analysis, causal forests, experimental design

16. Minimum Wage Employment Effects - Modern DID + Event Study

- Evaluated policy impact using Callaway-Sant'Anna DID to address pitfalls of TWFE under staggered adoption.
- Validated assumptions using event study pre-trend checks and communicated time dynamics clearly.
- Reported heterogeneous effects (e.g., -2.1% food service) and produced a structured 15-page policy brief with defensible methodology.

Skills / Tools: R (did, fixest), Stata, difference-in-differences, Callaway-Sant'Anna, event study, BLS microdata, policy writing, causal diagnostics

17. AI Trend Radar - Trend Detection + Evidence-Backed Summarization

- Built a multi-source trend intelligence pipeline that surfaces emerging topics using time-window velocity/momentum scoring and novelty/persistence signals.
- Reduced noise via deduplication and semantic clustering so repeated reposts don't inflate trend strength; grouped scattered mentions into coherent trend clusters.
- Added an LLM layer to generate concise, evidence-backed trend summaries (key entities, sentiment direction, why it's trending) with drill-down examples for verification.

Skills / Tools: Python, embeddings, semantic clustering, deduplication, trend scoring, LLM summarization, entity extraction, sentiment signals, PostgreSQL, Redis, Docker, dashboarding (React/Next.js)