Brejesh Balakrishnan

Machine Learning Engineer · GenAI / LLM Engineer
Bengaluru, Karnataka, India · +91 8870013219 · brejesh.bala@gmail.com
linkedin.com/in/brejesh-balakrishnan · github.com/brej-29 · huggingface.co/BrejBala

SUMMARY

Machine learning and GenAI engineer combining 1.5+ years of production software engineering at Accenture with end-to-end ML systems shipped to production. Build LLM / RAG applications with retrieval and faithfulness evaluation, calibrated predictive models with explainability, and full MLOps (MLflow registries, drift monitoring, CI quality gates, Docker) — deployed live. Strong Python and an engineering-first habit of measured decisions over assumed best-practices.

TECHNICAL SKILLS

Languages: Python, SQL, Java
ML & Deep Learning: scikit-learn, XGBoost, LightGBM, CatBoost, PyTorch, transfer learning (ResNet), Optuna (HPO), SHAP, calibration, feature engineering, model evaluation (PR-AUC / ROC-AUC)
GenAI / LLM: RAG, LangGraph (agentic / CRAG), LangChain, embeddings, vector databases (Pinecone, FAISS), sentence-transformers, LLM APIs (Groq, Gemini), retrieval evaluation (recall@k / MRR / nDCG), faithfulness / grounding eval, prompt-injection hardening
MLOps & Cloud: MLflow (champion / challenger registry), DVC, DagsHub, Pandera (data contracts), Evidently (drift), FastAPI, Docker (multi-stage), CI/CD (GitHub Actions, model-quality gates), Prometheus, AWS (SageMaker, Lambda, S3)
Engineering: REST APIs, SSE streaming, automated testing (pytest), Git / GitHub, Agile / Scrum, Linux

EXPERIENCE

Accenture Jul 2024 – Mar 2026
Advanced Application Engineering Analyst Bengaluru, India

- Optimized AWS-hosted REST APIs (Java / Spring Boot), cutting high-latency responses (>5s) from ~80% to <20% via profiling, SQL query tuning, and caching — improving responsiveness and scalability on a financial-services platform.
- Delivered 200+ story points across Agile / Scrum sprints, translating ambiguous requirements into production features and collaborating across UI, QA, and DevOps teams with regular stakeholder demos.
- Maintained >90% automated test coverage (unit, contract, BDD) tracked via SonarQube; strengthened CI/CD reliability through code reviews, linting / test gates, and root-cause-driven defect closure.
- Built and modified REST endpoints with business-rule logic; debugged DevOps pipeline issues and optimized release workflows on AWS (Jenkins CI/CD).

UnisLink Jul 2023 – Aug 2023
Data Engineering Intern Chennai, India

- Researched and benchmarked natural-language-to-SQL (NLQ / NL-to-SQL) approaches, documenting accuracy–complexity tradeoffs to guide early product direction.
- Built and validated SQL Server replication / synchronization checks and a replication test plan to improve data consistency across large tables.

Yaltech Global Consulting Jul 2022 – Sep 2022
Machine Learning Intern Tiruchirappalli, India

- Trained and cross-validated classification / regression models (Logistic Regression, Random Forest) in Python / scikit-learn on structured datasets, evaluating with cross-validation and standard metrics.
- Performed EDA, data cleaning, and feature engineering (Pandas / NumPy); communicated findings to non-technical stakeholders with clear visualizations.

PROJECTS

RAG Agent Workbench — Agentic Retrieval Backend *FastAPI · LangGraph · Pinecone · Groq*

- Built a production RAG backend as an 8-node LangGraph pipeline with conditional routing: query normalization → multi-turn contextualization → dense retrieval (Pinecone, llama-text-embed-v2) →

bounded corrective retrieval (CRAG) → web-fallback routing (Tavily) → grounded generation (Groq Llama 3.1) → faithfulness verification.
- Built a deterministic retrieval-evaluation harness (recall@k, MRR, nDCG@k, precision@k as pure functions over a hand-labeled golden set, designed to avoid circular validation) and used it to A/B-test a hosted reranker — measuring it net-negative (nDCG@3 −0.057, +435 ms) and disabling it on the data rather than shipping on assumption.
- Engineered structural guarantees: a two-threshold retrieval gate with a deterministic abstention path (the LLM is never asked to answer from empty context), a bounded CRAG loop with a provably non-runaway iteration cap, embedder-independent two-layer faithfulness checking, and indirect-prompt-injection hardening via structural delimiting of untrusted content.
- Instrumented per-request token / cost accounting and Prometheus histogram latency percentiles (replacing an unreliable 20-sample estimate) with true SSE token streaming; 343 tests and CI installing from a hash-pinned lockfile; deployed on Docker / Hugging Face Spaces (backend) + Streamlit Cloud (frontend).

Customer Churn Prediction — End-to-End MLOps + GenAI *XGBoost · MLflow · SHAP · FastAPI*

- Built a leakage-safe sklearn pipeline on the IBM Telco dataset (7,043 records, 26.5% churn); benchmarked 6 models under 5-fold stratified CV on PR-AUC and tuned XGBoost with Optuna (PR-AUC 0.62→0.67), with isotonic calibration and a cost-based decision threshold (0.174) from an explicit 5:1 false-negative:false-positive cost — test PR-AUC 0.66 / ROC-AUC 0.85 / recall 0.87, optimism gap −0.01.
- Layered a SHAP-grounded LLM explanation system: per-prediction SHAP drivers feed a Pydantic-validated explanation with RAG-retrieved retention tactics (sentence-transformers + FAISS) and a provider-agnostic Gemini→Groq wrapper with a deterministic fallback; a two-tier faithfulness eval caught a real grounding bug and raised faithfulness 0.72→0.90 (n=50, seeded).
- Engineered a reproducible MLOps stack: a single-source Pandera data contract enforced across train and serve (a test fails if the API schema diverges), Evidently drift monitoring with a 30%-drift retrain rule, and an MLflow champion / challenger registry on DagsHub with gated promotion and DVC data versioning.
- Built GitHub Actions CI with a model-quality gate that blocks merges if the champion drops below a PR-AUC floor; cut the Docker API image 57% (11.4→4.9 GB) via CPU-only torch and a split slim UI image; deployed live to Hugging Face Spaces (FastAPI + Streamlit) pulling the champion from the registry at startup (~399 tests).

Inventory Bin Classification — CV on AWS SageMaker *PyTorch · ResNet-50 · Lambda*

- Built an end-to-end CV pipeline classifying warehouse bin item counts from 10,000+ images using ResNet-50 transfer learning, with deterministic stratified splits stored as S3 ImageFolder channels.
- Ran SageMaker Bayesian hyperparameter tuning (learning rate, weight decay, batch size, backbone-freeze) with 2-instance distributed training and managed spot instances; ~22% macro-F1 gain over baseline.
- Deployed a real-time SageMaker endpoint with a custom PyTorch inference handler and AWS Lambda integration (image URL / S3 URI → JSON with label, confidence, probabilities), with full resource cleanup.

Disaster Tweet Classification — NLP Benchmarking *TensorFlow · TF Hub · scikit-learn*

- Benchmarked 7 architectures (TF-IDF + Naive Bayes, Dense, LSTM, GRU, BiLSTM, Conv1D, and Universal Sentence Encoder transfer learning) on ~7,600 tweets with a consistent 90/10 split.
- USE transfer learning achieved the best results (81.5% accuracy, 0.81 F1), demonstrating that pre-trained encoders outperform from-scratch models on small datasets.

EDUCATION & CERTIFICATIONS

B.E., Electronics & Communication Engineering 2020 – 2024
College of Engineering, Guindy (Anna University), Chennai · CGPA 8.54 / 10

- **AWS Machine Learning Engineer Nanodegree** — Udacity (Dec 2025)
- **Oracle Cloud Infrastructure 2025 Data Science Professional** — Oracle (Oct 2025 – Oct 2027)
- **Advanced Data Science & AI Program** — Logicmojo Academy (Jan 2026)
- **IBM Data Science** (Tools, Python, Databases & SQL) · **Version Control with Git** — Atlassian