YI-LEI LI

Newark, DE | ray1998514@gmail.com | LinkedIn | GitHub

Summary

Data analyst with an **M.S. in Computer Science** (University of Delaware, May 2026) and a banking-focused portfolio: credit-risk analytics on 30,000 customers (**SQL** window functions, **Python**, **XGBoost + SHAP**, 0.78 AUC, 3.1x targeting lift over random outreach), a **Spring Boot** transaction service whose ledger is checked by **SQL reconciliation** queries in CI, and a production-style **ETL pipeline** processing 22,000+ records per run. Prior backend engineering internship at Cathay Financial Holdings, in a regulated banking environment.

Technical Skills

SQL & Data:	PostgreSQL, complex SQL (window functions, CTEs), DuckDB, ETL pipelines, data cleaning
Python:	pandas, scikit-learn, XGBoost, SHAP, matplotlib
ML & AI:	logistic regression, gradient boosting, model explainability (SHAP), lift/segmentation analysis, LLM integration (Anthropic Claude API), AI-assisted development
Visualization & Reporting:	Tableau, Microsoft Excel, PowerPoint, matplotlib, KPI dashboard design
Engineering:	Java (Spring Boot 3, JPA, JUnit), TypeScript/React, REST APIs, Git, CI/CD (GitHub Actions)

Projects

Credit Card Default Risk Analysis — SQL, Python, XGBoost/SHAP, Tableau [GitHub] [Dashboard]

- Analyzed **30,000 credit-card customers** with five PostgreSQL-compatible SQL analyses (**NTILE** deciles, **LAG** over an unpivoted payment ledger, multi-CTE pipelines): the 4% of the book in serious delinquency defaults at **62.9%**, and month-over-month repayment deterioration flags 56.5%-risk customers 1–2 statements before serious delinquency.
- Built a logistic-regression baseline vs **XGBoost** (held-out AUC 0.744 / **0.777**) with **SHAP** explainability; the top-scored decile defaults at 68.7% and captures **31% of all defaults — 3.1x lift** over random outreach.
- Added an **LLM reporting layer** (Claude API) that drafts the weekly portfolio-risk memo from aggregated results only — grounded prompting, no customer-level data exposure.

Banking Transactions API — Java 21, Spring Boot 3, PostgreSQL [GitHub]

- RESTful core-banking service with **atomic transfers**: ordered pessimistic row locks (`SELECT ... FOR UPDATE`) prevent double-spend and deadlock; an immutable audit ledger records post-transaction balances.
- Wrote **SQL reconciliation checks** over the ledger: a **window-function** running total must match every recorded balance, transfer legs must pair up, and money must be conserved; CI proves each check catches a deliberately corrupted row.
- **32 tests** in **GitHub Actions CI**, including concurrency and reconciliation tests on real **PostgreSQL** (Testcontainers).

Job-Opening Monitor — Node.js, TypeScript, Supabase [GitHub]

- Scheduled **ETL pipeline** pulling **22,000+ job postings per run** from **116 sources** (Workday, Greenhouse, Lever, . . .) behind one normalization layer; **idempotent** re-runs via a Postgres state table; GitHub Actions cron with Telegram/email alerts.
- **Data-quality reporting** each run: per-source health with p95 latency and failure type (timeout, rate limit, HTTP, parse); **343 Vitest tests** over matching, sponsorship-classification, and normalization logic.

Job Application Tracker + OPT Counter — React, TypeScript, Supabase [GitHub]

- Full-stack app for tracking my own applications: storage abstraction runs on `localStorage` or Supabase with Postgres **Row-Level Security**; **43 Vitest tests** including merge-overlapping-intervals date logic for the OPT unemployment counter.

Education

University of Delaware Newark, DE
Master of Science in Computer Science — GPA 3.66/4.0 *May 2026*

National Kaohsiung University of Science and Technology Kaohsiung, Taiwan
Bachelor of Science in Electrical Engineering *2020*

Experience

Cathay Financial Holdings Co., Ltd. *Jul 2023 – Nov 2023*
Back-End Software Engineering Intern Taipei, Taiwan

- Delivered a CRUD service (**Java + SQL**) as part of an **Agile** development team, tracing the firm's transaction-processing flow end to end in a regulated banking environment.
- Studied how the banking system protects correctness — sufficient-balance checks before transfers and concurrency control serializing transactions per account (**ACID** integrity and isolation).

Graduate Research Assistant — Chang Gung University *2021 – 2022*
Researched **Neural Radiance Fields (NeRF)** for 3D scene reconstruction from multi-view images; earlier built a full-stack IoT sensor-monitoring dashboard (REST + WebSocket) as an undergraduate research assistant.