AI Lead Prediction Model - Nutto Hybrid Engine v2

Project Overview

Full-stack app analyzing lead data with Random Forest ML predictions and RAG-powered chat. Tech: React/Vite/Tailwind frontend; FastAPI/Pandas/Scikit-learn/FAISS/SQLite backend. Docker-ready for enterprise deployment.

Core Features

- **Lead Scoring**: CSV upload → feature engineering (EngagementScore, InteractionCount) → probability (0-1), Priority (High>0.3).
- **RAG Chat**: SentenceTransformers embeddings + FAISS for queries like "high priority Google leads."
- **Dashboard**: Recharts visuals (F1-score, precision/recall), fuzzy search, prediction history.

Tech Stack & Setup

Backend: `cd backend; pip install -r requirements.txt; uvicorn main:app --reload`

Frontend: `cd frontend; npm install; npm run dev`

Endpoints: /train (historical CSV), /predict (new leads), /chat (RAG queries).

Enhancements Roadmap

- XGBoost upgrade, Celery/Redis async processing, disk-persisted FAISS.
- UI feedback loops, smart imputation, auto-retraining on conversion data.

Enterprise AWS Mumbai Costs (₹/month, Moderate Usage)

Service	Instance/Type	On-Demand	Savings Plans	Notes
EC2 Backend/ML	t3.medium	5,500	3,300	FastAPI/Random Forest
RDS Database	db.t3.micro	2,200	1,300	HA SQLite replacement
S3 Storage	100GB Standard	800	800	CSVs/FAISS indexes
EBS Volumes	gp3 100GB	400	400	Upload persistence
Embeddings Inference	CPU/HF 100q/day	3,500	2,100	SentenceTransformers
Load Balancer/Transfer	ALB in-region	1,800	1,200	Minimal egress
Total		**14,200**	**9,100**	Spot/Graviton: 70% off

Chart Reference: Monthly breakdown bar chart shows ~₹16,000 base scaling to ₹8,000 optimized.

Monthly cost breakdown for enterprise-scale deployment handling 1000s of leads and queries daily

Usage Instructions

1. Train: Upload CSV with 'Converted' target to /train.
2. Predict: New leads → scored priorities.
3. Chat: "Show high priority leads" → contextual RAG response.

Optimizations: Savings Plans (40-60% off), Enterprise Discount Program (20-30%), Graviton instances (20% CPU savings).