

AI Stock Oracle — MVP Build Spec (Local + Cloud Setup)

Core MVP Objective

Build a browser-based dashboard that identifies under-the-radar (UTR) suppliers whose activity synchronizes with AI, power, or water sector expansion — using real-time public data feeds, NLP extraction, and correlation scoring.

Key MVP Features

1. Input Sources (Phase 1 MVP)
 - SAM.gov RFP Feed (daily via API)
 - NewsAPI.org (filtered for AI infra terms)
 - Reddit (scraped): r/dataengineering, r/utilityworkers, r/municipal
 - Local job posts: RSS or LinkedIn scraper (optional)
 - Manual CSV import for small test datasets
2. NLP Engine (Local)
 - spaCy: NER + phrase patterns
 - Basic OpenIE-style pattern extraction for signals
 - Output: (Entity, Signal, Location, Date, Type)
3. Signal Correlation Logic
 - Local Rolling Growth Index builder
 - Manual benchmark curves
 - Calculate Pearson correlation (r) locally via NumPy
4. Scoring & Tiering
 - Oracle Score (basic version)
 - Tier tags: Low / Medium / High
 - Highlight UTR matches ($r > 0.8$)
5. Dashboard (Browser)
 - FastAPI backend + Next.js frontend
 - Features: Entity cards, Filters, CSV Export, Manual tagging

Stack Mapping (Your Setup)

Layer: Tools

Ingestion: Python scripts (run via Conda) using requests, schedule, aiohttp

Storage: Raw JSONs on 1TB external / SQLite or PostgreSQL

Processing: spaCy NLP, custom scoring, NumPy correlation

Frontend: Next.js hosted locally

Backend API: FastAPI running on M4 Mini or staging server

Logging: Local logs + optional cloud stream

Versioning: GitHub + VS Code forks

Orchestration: Manual cron/daemon for now

First 3 Subscriber-Facing Outputs

1. Watchlist Feed: UI stream of live UTR matches
2. Weekly Digest PDF: Top 10 Oracle Picks
3. Exportable JSON/API: Developer-ready filtered output

Local Deployment Notes

- All can run headless on your M4 Mac Mini
- Use 1TB external for workspace
- Conda env: oracle-dev
- VS Code (Cursor/Windsail) for dev
- Deploy to public IP when ready

MVP = Done when:

- Dashboard loads
- One UTR supplier shown with score > 0.8
- List can be exported
- Tested with one real and one fake data source