POI Data Validation and Correction Pipeline

Guadalajara Hackathon 2025 - HERE Technologies

Ricardo Gutierrez

May 18, 2025

Hackathon Challenge

Objective:
Detect, validate, and correct POI (Point of Interest) errors in large geospatial datasets.

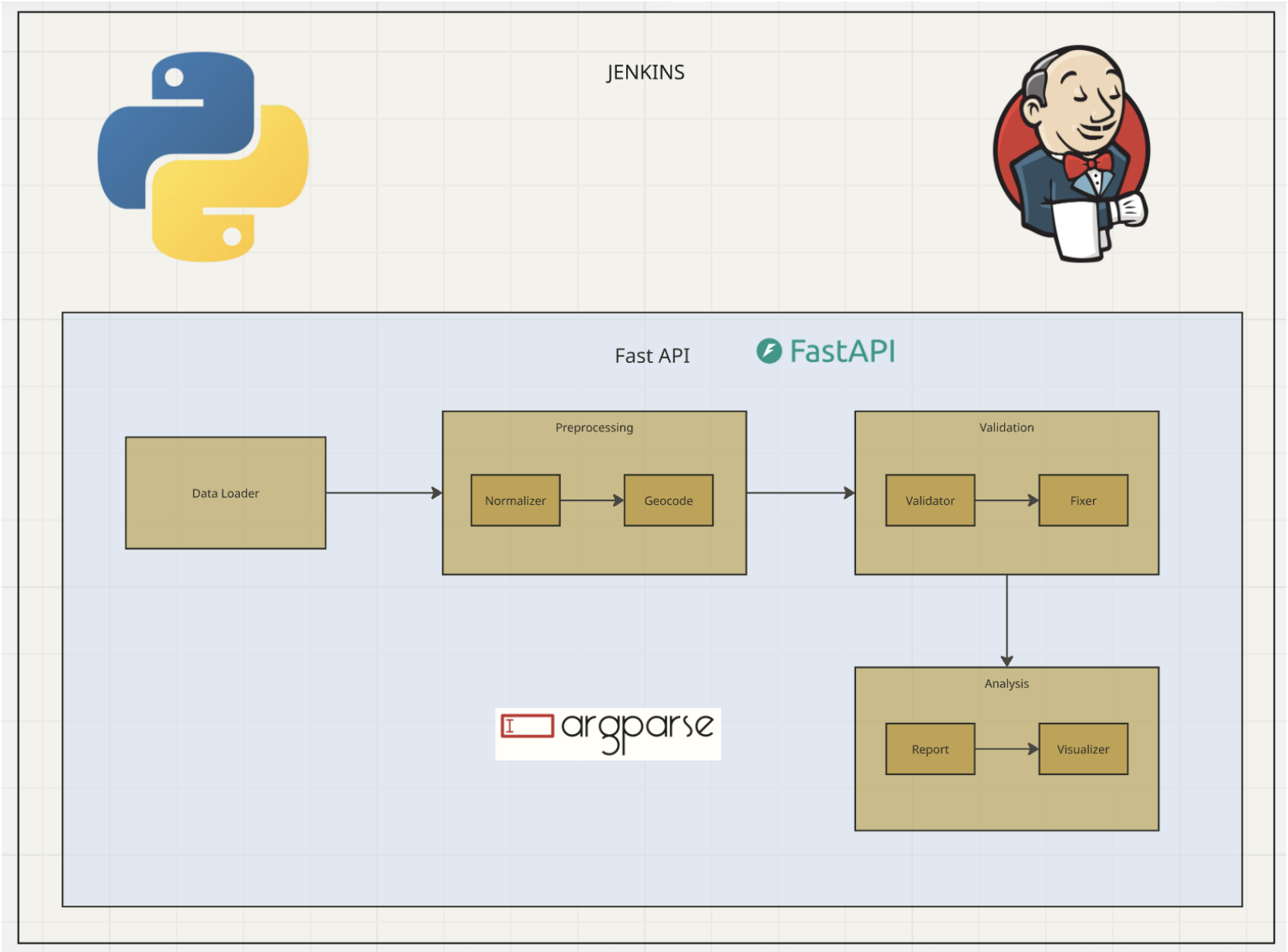
- **Scenario 1:** Delete POIs in wrong locations
- **Scenario 2:** Relocate misplaced POIs to correct geometry
- **Scenario 3:** Fix incorrect MULTIDIGIT attributes
- **Scenario 4:** Identify legitimate exceptions with justification

Solution Overview

How our pipeline addresses the challenge:

- **Modular Python pipeline**: each stage handles a scenario (load, validate, fix, report).
- **Automatic detection & correction**: rules for each scenario are implemented and extensible.
- **Human-readable reports**: detailed HTML for analysts and auditing.
- **Web dashboard**: run, monitor, and visualize results in real time.
- **CI/CD ready**: can be deployed and automated using Jenkins.

Architecture Diagram



Pipeline Stages

1. **Load Data:** Ingest POI and street datasets (CSV/GeoJSON).
2. **Normalization:** Standardize attribute formats for processing.
3. **Geocoding:** Assign geometry to POIs based on closest valid street segment.
4. **Validation:** Detect errors for each scenario using business rules.
5. **Auto-fixing:** Suggest and apply corrections (deletion, relocation, attribute fixes).
6. **Reporting:** Generate detailed HTML reports for results and exceptions.
7. **Logging/Monitoring:** Store logs for each execution; show progress in dashboard.

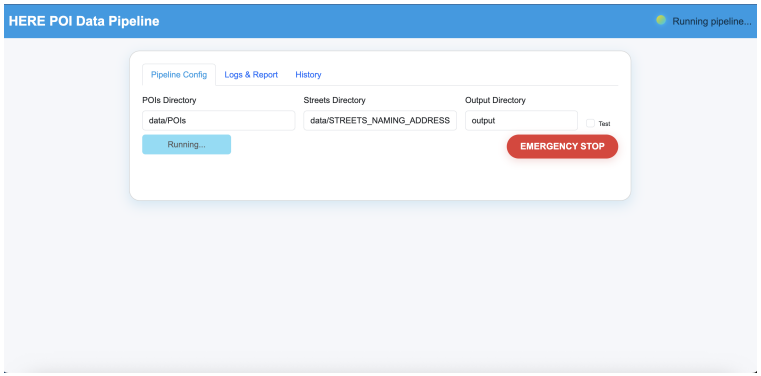
Stage Example: Validation

- Detects misplaced POIs using spatial analysis.
- Flags attributes with incorrect MULTIDIGIT values.
- Identifies records for deletion or correction.

```
st/test_scenarios.py
   poi_id  violation_code                                violation_detail
0       1          DELETE            POI missing or invalid (empty name)
1       2     UPDATE_SIDE  Street segment not found — possibly wrong side
2       3  FIX_MULTIDIGIT                Multiply Digitised should be N
3       4  FIX_MULTIDIGIT                Multiply Digitised should be N
4       5  FIX_MULTIDIGIT                Multiply Digitised should be N
   poi_id    poi_name  link_id poi_st_sd  percfrref
1       2   Valid POI   999999         R         80
2       3   Multi POI     2002         R         50
3       4  OutOfRange     2002         L         50
4       5     AllGood     2002         L         25
rguti@MacBook-Air-35 Here Hackathon %
```

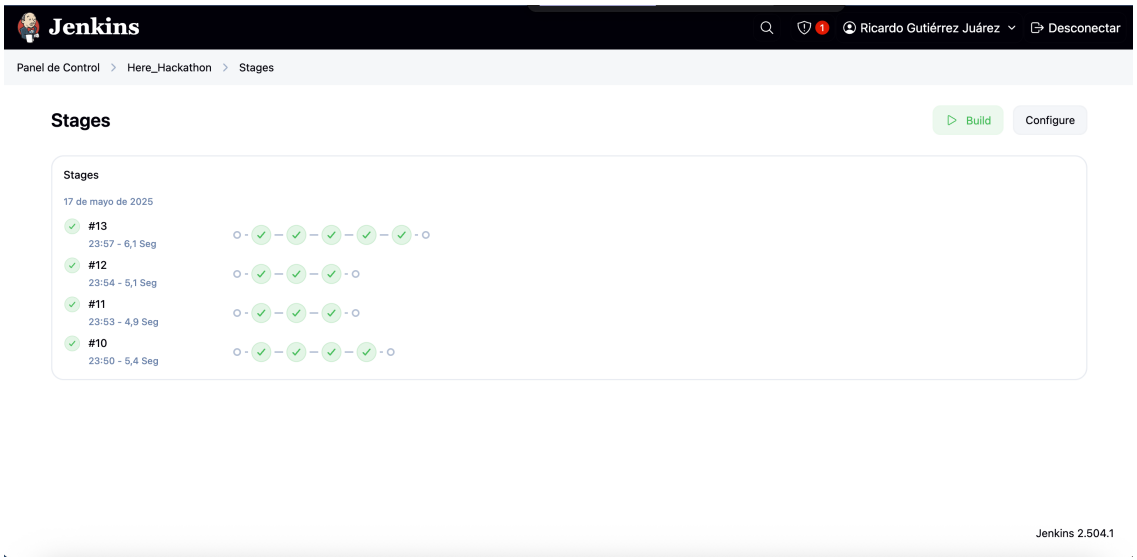
Web Dashboard

- Launch and monitor pipeline from browser
- Real-time logs and emergency stop
- View and download reports
- See full history of previous runs



Jenkins Automation & CI/CD

- Pipeline can be triggered and monitored via Jenkins
- Supports automated testing, deployment, and notifications
- Ensures reproducibility and continuous integration for large teams



Scalability and Extensibility

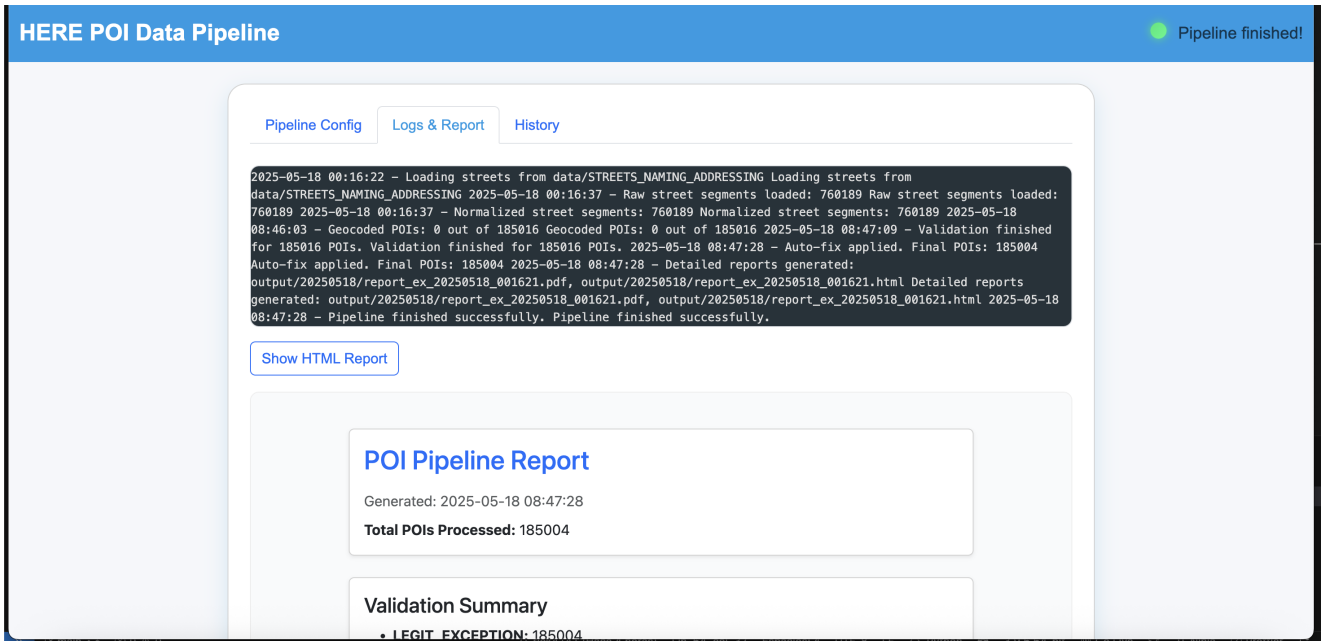
- **Modular stages:** Add new validation rules or fixers as Python modules
- **Country-specific configs:** Plug in rules or data per country/region
- **Cloud-ready:** Can run locally, in Docker, or on cloud servers
- **User-friendly:** Non-technical users can operate via web interface

Requirements

- Python 3.9+ (FastAPI, Pandas, Geopandas, etc.)
- Node.js (for optional frontend features)
- Docker (for deployment)
- Jenkins (for CI/CD)
- All dependencies listed in `requirements.txt`

Demo and Results

- Live demo: running pipeline and showing corrections
- Before/After: visual examples of corrections applied
- Downloadable reports and log files for auditing



Conclusion

- Flexible, extensible, and robust pipeline for POI data quality
- End-to-end workflow: from raw data to report and monitoring
- Ready for global scale, multi-team, and automated deployment
- All code and docs open source for future improvements

Thank you!

Q&A

Questions?