

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.

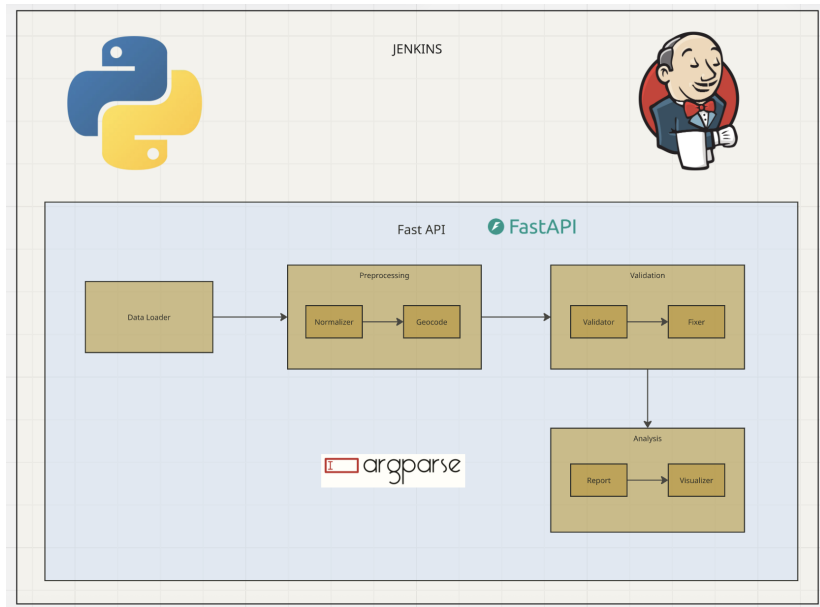
- **Scenario 1:** Delete POIs in wrong locations
- **Scenario 2:** Relocate misplaced POIs to correct geometry
- **Scenario 3:** Fix incorrect MULTIDIGIT attributes
- **Scenario 4:** Identify legitimate exceptions with justification

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

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

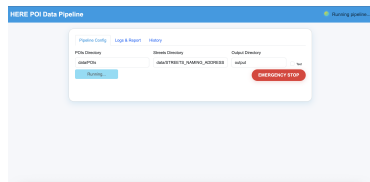
Stage Example: Validation

- 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
rauti@MacBook-Air-35 Here Hackathon %
```

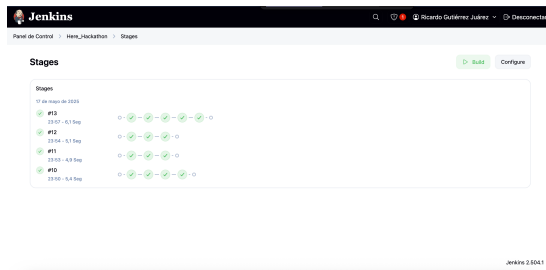
Web Dashboard

- 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

The screenshot displays the 'HERE POI Data Pipeline' web interface. At the top, a blue header bar contains the title 'HERE POI Data Pipeline' on the left and a green status indicator 'Pipeline finished!' on the right. Below the header, there are three tabs: 'Pipeline Config', 'Logs & Report' (which is active), and 'History'. The 'Logs & Report' tab shows a log window with a black background and white text detailing the pipeline execution. The log text includes timestamps, file paths, and counts of processed street segments and POIs. Below the log window is a button labeled 'Show HTML Report'. Underneath this button is a section titled 'POI Pipeline Report' which states 'Generated: 2025-05-18 08:47:28' and 'Total POIs Processed: 185004'. Below the report is a 'Validation Summary' section showing '+ 1 LEGIT EXCEPTION: 185004'. At the bottom of the interface, there is a navigation bar with various icons for navigation and search.

HERE POI Data Pipeline Pipeline finished!

Pipeline Config **Logs & Report** History

```
2025-05-18 08:16:22 - Loading streets from data/STREETS_NAMING_ADDRESSING Loading streets from
data/STREETS_NAMING_ADDRESSING 2025-05-18 08:16:37 - Row street segments loaded: 768189 Row street segments loaded:
768189 2025-05-18 08:16:37 - Normalized street segments: 768189 Normalized street segments: 768189 2025-05-18
08:46:83 - Geocoded POIs: 0 out of 185816 Geocoded POIs: 0 out of 185816 2025-05-18 08:47:09 - Validation finished
for 185816 POIs. Validation finished for 185816 POIs. 2025-05-18 08:47:28 - Auto-fix applied. Final POIs: 185804
Auto-fix applied. Final POIs: 185804 2025-05-18 08:47:28 - Detailed reports generated:
output/20250518/report_ex_20250518_081621.pdf, output/20250518/report_ex_20250518_081621.html Detailed reports
generated: output/20250518/report_ex_20250518_081621.pdf, output/20250518/report_ex_20250518_081621.html 2025-05-18
08:47:28 - Pipeline finished successfully. Pipeline finished successfully.
```

Show HTML Report

POI Pipeline Report

Generated: 2025-05-18 08:47:28

Total POIs Processed: 185004

Validation Summary

+ 1 LEGIT EXCEPTION: 185004

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!

Questions?