

Ask Syracuse Data: A City-Scale Natural Language Analytics System for Civic Insight

Prathyusha Murala

Project Title and Summary

Ask Syracuse Data is a city-scale natural language analytics system that enables residents, journalists, and city staff to ask plain-English questions about housing conditions, public safety, and neighborhood-level trends in Syracuse, NY. Building on a previously developed personal NL-to-SQL analytics project focused on small, well-structured business datasets, this project scales that approach to heterogeneous, geospatial municipal data. The system translates user questions into validated analytical queries executed over curated Syracuse Open Data tables, returning transparent, reproducible answers supported by charts, maps, and clearly stated limitations. By lowering technical barriers while preserving analytical rigor, this project helps the Syracuse community better understand and responsibly use public data for decision-making.

Problem Statement

The City of Syracuse maintains an extensive Open Data portal containing over one hundred datasets covering housing, crime, infrastructure, education, and city services. While this data is publicly available, it remains largely inaccessible to non-technical users due to fragmented schemas, inconsistent update frequencies, and the need for advanced analytical and geospatial skills. As a result, residents, journalists, and community organizations often rely on anecdotal evidence rather than data-driven insights when discussing neighborhood conditions and policy outcomes.

This project addresses the question: *How can non-technical users ask meaningful, data-backed questions about Syracuse neighborhoods while ensuring accuracy, reproducibility, and ethical interpretation?* The answer matters because misinterpretation of civic data can reinforce harmful narratives, obscure structural issues, or erode trust in public institutions. Ask Syracuse Data aims to provide a governed, transparent analytics interface that allows stakeholders to explore city data responsibly, without requiring them to upload datasets into general-purpose language models or manually write queries.

Data Sources

The initial phase of the project will focus on a curated subset of Syracuse Open Data datasets chosen for policy relevance, data quality, and analytical depth:

- **Code Violations:** Housing enforcement data sourced from Syracuse Open Data with historical records available at the time of download. While the underlying system may update frequently, this project relies on static snapshots as published on the Open Data portal. Limitations include reporting bias and variation in enforcement intensity.
- **Syracuse Rental Registry:** Records of registered rental properties as published on Syracuse Open Data. The analysis will use a fixed snapshot of the data available during the project period. Coverage depends on landlord participation and enforcement.
- **Vacant and Unfit Properties:** Indicators of housing instability and neighborhood disinvestment. Definitions may change across years.
- **Crime Data (Part 1 offenses):** Incident-level data. Requires normalization and careful framing to avoid stigmatization.

No external datasets will be incorporated in Phase 1 to maintain clarity of provenance and reproducibility.

Technical Approach

The system will be implemented as a modular analytics pipeline separating data ingestion, transformation, analysis, and presentation. All analyses will be performed on static data snapshots downloaded from the Syracuse Open Data portal, rather than live or continuously updating data streams. Raw datasets will be stored and processed using Python, Pandas, DuckDB, and GeoPandas to support scalable tabular and spatial analysis. A schema registry will define allowed tables, fields, joins, and aggregation types to prevent invalid or unsafe queries.

Large Language Models (LLMs) will be used strictly as an interface layer to translate natural language questions into structured query plans and to draft narrative explanations. All factual results will be produced by deterministic SQL or Python computations, not by the LLM itself. Validation strategies will include schema enforcement, aggregation sanity checks, comparison of LLM-suggested results against ground-truth calculations, and rejection of unsupported or ambiguous queries. This approach directly applies techniques from earlier tasks involving LLM output validation, prompt iteration, and bias detection, particularly around framing, normalization, and uncertainty communication.

Deliverable Description

The final deliverable will be a deployed interactive web application (planned to be built with Streamlit) that allows users to ask questions about Syracuse housing and safety data in plain English using curated, static datasets sourced from Syracuse Open Data. The application will return reproducible answers accompanied by charts and geospatial maps, along with data source citations and clearly stated caveats. A supporting GitHub repository will include full documentation, data dictionaries, prompt logs, validation examples, and deployment instructions.

Success Criteria

- Users can successfully ask and receive correct answers to predefined categories of civic questions (housing, crime, neighborhood comparisons).
- All outputs are reproducible and traceable to executed queries over verified datasets.
- The system prevents unsupported joins, hallucinated results, and stigmatizing narratives.
- Documentation is sufficient for city staff or other developers to understand, run, and extend the project.
- The project demonstrates clear advancement from prior small-dataset NL-to-SQL work to city-scale analytics.

Timeline

- Weeks 1–2 Dataset selection, stakeholder framing, proposal refinement
- Weeks 3–4 Data acquisition, cleaning, schema definition, exploratory analysis
- Weeks 5–6 Core analytics pipeline and geospatial joins implementation
- Weeks 7–8 LLM integration, prompt design, validation logic
- Weeks 9–10 Application development and working prototype
- Weeks 11–12 Testing, documentation, bias review, and polish

Risks and Mitigations

- **Data quality issues:** Mitigated through explicit documentation of limitations and exclusion of unreliable fields.
- **LLM hallucinations:** Mitigated by strict schema enforcement and grounding all outputs in executed queries.

- **Scope creep:** Mitigated by limiting Phase 1 to housing and crime datasets only.
- **Harmful interpretations:** Mitigated through normalization, careful language framing, and explicit caveats.
- **Technical complexity:** Mitigated by modular design and incremental development milestones.