Improving Urban Livability: An Analysis of 311 Complaints and Their Relationship to Short-Term Rentals

Shreya Akotiya, Ashmita Paruchuri Balaji, Nikitha Seelam Balaji, Shashira Guntuka
DATA 226 Data Warehouse
San Jose State University

***Abstract*—The aim of this project is to build a system that looks at 311 complaints across the cities of the United States and explores how short-term rentals like Airbnb, along with tourism in general, may be linked to these complaints. The idea is to understand if there are patterns between rental activity and issues that affect day-to-day life such as noise, sanitation problems or illegal occupancy. To achieve the project will combine two main datasets: 311 service requests and Airbnb listings/booking data. By combining these dataset we aim to determine whether more 311 complaints are also reported in areas with higher short-term rental activity.The data is stored and processed using tools like MongoDB Atlas, AWS S3, AWS Glue, and Amazon Redshift, with workflows managed through AWS Airflow (MWAA). We use Amazon SageMaker for data analysis and machine learning, and Amazon QuickSight to visualize the results in an interactive and easy to interpret way. The results give us useful insights for ci, local community groups, and planners who are trying to make the most of tourism while still keeping neighborhoods comfortable and livable for residents.**

***Index Terms*—NYC 311, Airbnb, data warehouse, ETL, AWS Glue, Redshift, QuickSight, Airflow, urban analytics.**

I. INTRODUCTION

Modern urban areas are trying to maintain a balance between the quality of life of their citizens and development. While tourism and short-term rentals such as Airbnb contribute to the economy and offer new possibilities of making life difficult for local residents. Noise, sanitation problems, illegal rentals, and housing shortages are some of the issues that are frequently reported via 311 complaint systems. These records provide a glance at the way city residents live their lives on a daily basis, but alone they don't clarify what could be the reason for the complaints.

In this research, we concentrate on New York City which is not only the place with one of the most active 311 systems in the country but also has one of the biggest Airbnb markets. Our objective is to find out if there is any correlation between short-term rental activities and city livability which is measured by the kinds and the number of 311 complaints. By combining NYC Open Data (311 Service Requests) and InsideAirbnb data, we intend to unravel the scenarios that demonstrate how the quality of life in a neighborhood may be influenced by the concentration of short-term rentals.

II. PROBLEM STATEMENT

Although New York City makes both 311 complaint data and Airbnb listings available to the public, there is very little joint analysis of the two datasets. Consequently, the effects of the growth of short-term rentals on residents' daily life are still quite obscure to policymakers and urban planners. Existing analyses have been mostly silent on the issue of data combining, focusing instead on individual datasets or using mere snapshots of time that fail to capture longer-term trends. The core issue is the fact that these two datasets are in different formats and have different update cycles, which hinders efficient combining and analysis. In the absence of a consolidated platform, it becomes a nearly impossible task to monitor how noise complaints due to Airbnb activities correlate with complaints about sanitation or housing issues in certain neighborhoods.

Our project sets out to address this issue by building a comprehensive data warehouse that integrates 311 and Airbnb data into one unified, structured space. After the integration, one can employ SQL queries and statistical methods to spot the trends - for example, whether a higher number of Airbnb listings in an area corresponds with a greater number of 311 complaints. Such a method provides a more accurate and convincing picture based on data how tourism and short-term rentals affect the quality of life in the city.

III. MOTIVATION

Across many cities in the United States the rise of short term rentals such as Airbnb has created both opportunities and challenges. These platforms increase the flexibility of tourism, improve the economy and provide new ways of income to the property owners. Also the short-term rentals can make life difficult in neighborhoods where residents are already dealing with everyday concerns. Problems like noise from house parties or construction activities, sanitation issues, parking shortages, and even housing affordability are becoming more common and they are often reported through 311 complaint systems. New York City is one of the appropriate cases where these challenges can be seen. Tens of thousands of Airbnb postings have shown a noticeable overlap between neighbourhoods that have significant rental activity and those with higher resident complaint volumes. But this isn't just a New York problem, similar patterns are showing up in cities like San Francisco, Los Angeles, and Chicago, where residents feel the strain of tourism on local services. The motivation behind this project is to understand these connections in a

systematic way by combining complaint data with short term rental data. By doing this we can highlight how short term rental activity is affecting urban livability across U.S. cities and give the policymakers and communities better evidence to design solutions that balance economic growth with residents' quality of life.

IV. GOAL

The goal of this project is to design, implement, and analyze a scalable data warehouse that integrates NYC 311 complaint data with Airbnb listings to uncover how short-term rental activity influences neighborhood conditions. Specifically, the project aims to:

- Build an automated ETL pipeline using AWS services (Glue, Lambda, S3, Redshift, Airflow).
- Create dimension and fact tables that enable efficient multi-dimensional queries across time, borough, and complaint type.
- Conduct analytical SQL queries to identify correlations between complaint volume, Airbnb density, and property characteristics.
- Develop dashboards and visualization layers to help policymakers, analysts, and researchers interpret the data for actionable insights.

V. LITERATURE SURVEY

Several studies have explored the link between short-term rentals like Airbnb and urban issues such as noise complaints, housing prices, and neighborhood well-being. Together, these works provide valuable background for understanding how short-term rental activity can shape local living conditions and community dynamics.

Lee and Kim (2023) in "Four Shades of Airbnb and Its Impact on Locals: A Spatiotemporal Analysis of Airbnb, Rent, Housing Prices, and Gentrification" [1], examine how Airbnb listings influence housing affordability and neighborhood change. Their study finds that entire-home rentals and professional operators tend to raise rents and home prices the most, while smaller, casual hosts have only minor effects. This research highlights that short-term rentals do not impact all cities or communities equally, suggesting that regulations should focus on high-impact operators rather than individual hosts.

The "Detailed Data Analysis: The Rise of NYC 311 Noise Complaints" study [2], published on the NYC Data Science blog, analyzes trends in New York City's noise-related 311 calls. It identifies when and where noise complaints are most frequent and shows that such complaints have grown in certain neighborhoods, particularly in areas with higher population density or lower income levels. This analysis demonstrates how open city data can help identify quality-of-life concerns and shows a connection between neighborhood characteristics and resident well-being.

Kontokosta, Hong, and Korsberg (2017) in "Equity in 311 Reporting: Understanding Socio-Spatial Differentials in the Propensity to Complain" [3] provide another important perspective. They argue that 311 complaint data, while useful, can be biased because not all residents report issues at the same rate. Their findings show that under-reporting is more common in lower-income, minority, or limited-English-speaking neighborhoods, while over-reporting is more frequent in wealthier, more educated areas. This means that complaint data reflects both the actual problems and the social factors that influence who feels empowered to report them. For projects using 311 data—like ours—it's important to account for these differences to avoid misleading conclusions.

The article "The Perceived Impacts of Short-Term Rental Platforms: Comparing the Social Exchange Perspective" (2024) [4] examines how residents perceive the growth of short-term rental platforms in their neighborhoods. Using the social exchange framework, the authors compare perceived benefits such as tourism and extra income against negative impacts like noise, loss of community, and rising housing costs. Their results show that people's opinions about Airbnb vary widely, depending on their social and economic background. This study emphasizes that Airbnb's influence is not just about economics—it also affects how people experience everyday community life.

Finally, "The Relationship of Airbnb to Neighborhood Calls for Service in Three Cities" (2021) [5] directly connects Airbnb activity to urban service calls, including 311 and police data. By studying Nashville, New Orleans, and Portland, the authors found that neighborhoods with more Airbnb listings reported higher numbers of disturbance and disorder-related complaints. This suggests that a higher density of short-term rentals can increase pressure on local services and affect neighborhood livability.

Together, these studies provide important insights into how Airbnb and short-term rentals intersect with urban complaints and community well-being. They also highlight the value—and limitations—of using 311 data to study neighborhood conditions. Our project builds on this foundation by integrating Airbnb and 311 datasets to explore whether areas with more short-term rental activity also experience higher complaint levels, particularly those related to noise and community disturbances.

References

[1] S. Lee and H. Kim, "Four Shades of Airbnb and Its Impact on Locals: A Spatiotemporal Analysis of Airbnb, Rent, Housing Prices, and Gentrification," *Tourism Management Perspectives*, vol. 49, p. 101192, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S2211973623001204

[2] "Detailed Data Analysis: The Rise of NYC 311 Noise Complaints," *NYC Data Science Blog*, 2022. [Online]. Available: https://nycdatascience.com/blog/student-works/detailed-data-analysis-the-rise-of-nyc-311-noise-complaints/

[3] C. E. Kontokosta, B. Hong, and M. Korsberg, "Equity in 311 Reporting: Understanding Socio-Spatial Differentials in the Propensity to Complain," *arXiv preprint arXiv:1710.02452*, 2017. [Online]. Available: https://arxiv.org/pdf/1710.02452

[4] "The Perceived Impacts of Short-Term Rental Platforms: Comparing the Social Exchange Perspective," *International Journal of Sociology and Social Policy*, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0160791X24001349

[5] "The Relationship of Airbnb to Neighborhood Calls for Service in Three Cities," *Cities*, vol. 113, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0264275121001414

VI. SYSTEM ARCHITECTURE

Fig. 1 shows the end-to-end flow. We stage raw files in Amazon S3, transform them with AWS Glue (PySpark), orchestrate workflows through Apache Airflow, and load curated tables into Amazon Redshift. Amazon Sagemaker provides dashboards.

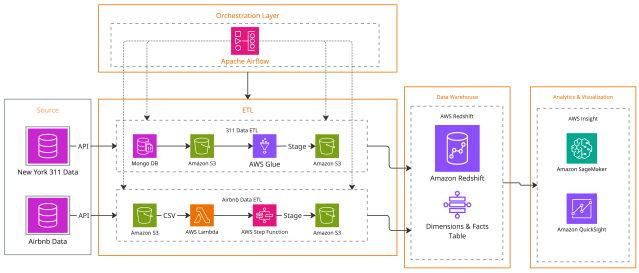


Fig. 1: High-level architecture: Ingestion (NYC 311 API, Inside Airbnb) → S3 (raw/staging/processed) → Glue / Step transforms → Redshift star schema → Sagemaker.

VII. METHODOLOGY: ETL PIPELINE

Data Collection, Ingestion, and Preparation

Data Collection

The data was collected from two primary sources:

- **NYC 311 Complaints Dataset:** Obtained from the NYC Open Data Portal. The dataset was provided in the form of large JSON files containing detailed records of public service complaints across New York City.
- **Airbnb Listings Dataset:** Sourced from Inside Airbnb, which provides publicly available Airbnb data. For New York City, the data was available as five large CSV files containing details about hosts, room types, prices, and reviews.

Data Ingestion

- **Data Extraction:** Raw complaint data was retrieved from the NYC Socrata Open Data API (SODA) for the years 2024 and 2025. Due to API limitations, data was downloaded in batches using the `offset` method, and each year's data was stored as separate JSON files.
- **Ingestion Pipeline:** A Python-based ingestion pipeline was developed using the `requests` and `pandas` libraries. The pipeline included retry logic and timeout handling to ensure reliable data extraction and storage.
- **Data Storage:** The ingested data was initially stored in **Amazon S3** for persistence and further processing.

Data Cleaning and Preparation

Data Cleaning: Data cleaning constituted a major part of the project due to the heterogeneity of the two data sources — NYC 311 complaints and Airbnb listings. The key cleaning steps included:

- **311 Complaints Dataset:**
  - Removed records missing essential location details such as latitude, longitude, or address.
  - Eliminated duplicate complaints sharing the same time, location, and description to prevent redundancy.
- **Airbnb Listings Dataset:**
  - Removed listings lacking coordinates or host information.
  - Retained only listings within NYC's geographic boundary (latitude: 40.4–41.0, longitude: –74.3– –73.7).
  - Converted the `price` column to numeric format by removing special characters such as dollar signs and commas.
  - Excluded listings with zero or negative prices.
  - Standardized text fields (e.g., neighbourhood, host name, room type) for consistency.
  - Removed duplicate listing IDs to ensure unique property representation.

Feature Engineering: Several new features were engineered to enhance analytical insights:

- **Geohash Generation:** Created an alphanumeric **geohash** based on latitude and longitude for precise spatial grouping and matching between Airbnb and 311 complaint datasets.
- **Time-Based Features:**
  - Computed complaint duration in the 311 dataset as the difference between closed date and creation date.
  - Preserved Airbnb attributes such as availability and review count to analyze the relationship between listing activity and local complaint patterns.

Data Preparation: After cleaning and feature engineering, both datasets were prepared for analysis:

- Converted the processed data into **Parquet** format for efficient storage and querying.
- Uploaded cleaned data to **Amazon S3** for long-term storage and accessibility.
- Loaded datasets into **Amazon Redshift** to enable analytical joins using shared attributes such as date, borough, and geohash.

VIII. DATA PIPELINE SETUP

Different data formats and sizes were the reasons for which we have developed two main ETL pipelines for our project: one for NYC 311 complaints and another for Airbnb listings. We automated and orchestrated both pipelines with Apache Airflow that we executed in a Docker container for simple configuration, isolation, and reusability. This configuration has made the pipeline fully automated and stable in various environments.

Using Python scripts and Airflow tasks, we made a call to the API to get the data dynamically from the city's open data portal. In this way, we could always get the latest records without manual downloads or saving files locally. The data that was grabbed in **MongoDB** where it is kept as document store. This was then traversed and save to S3 finally.

We organized data in Amazon S3 by making folders with a defined structure:

- **raw/311-complaints/** - holding 311 API data in JSON format.
- **raw/airbnb/** - holding Airbnb listing data in CSV format.

The scheduling and retry features of Airflow were instrumental in the proper consumption of data. A failed upload would be automatically retried, and the logging of job runs would take place.

Once data is loaded into S3 another Airflow dag was created to run end-to-end ETL pipeline for both data sources. **AWS glue** is used for 311 complaint data and **AWS Step function** is used for Airbnb data. ETL'd data was then loaded into **AWS Redshift** in a start schema format. Figure 2 illustrates the complete ETL pipeline.

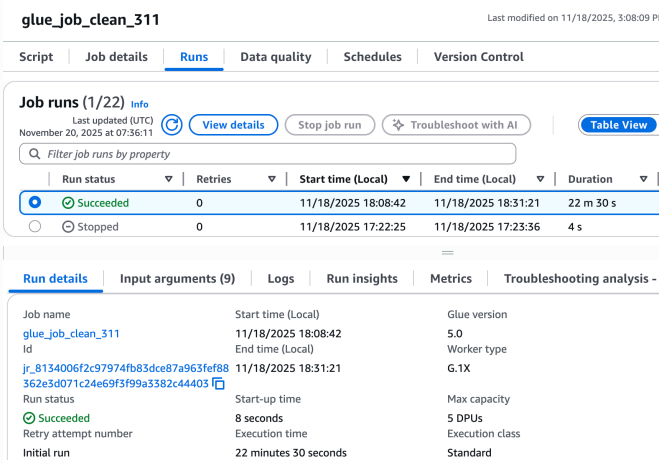


Fig. 3: ETL for 311 Complaints Data (AWS Glue)

ETL for 311 Complaints Data (AWS Glue)

The 311 ETL pipeline was designed using AWS Glue, as it works well with huge volumes of semi-structured data. Glue automated the process of extracting and transforming the complaint records downloaded directly from the NYC Open Data API and then stored them in Amazon S3.

Data cleaning, standardization, and enrichment based on the rules defined in the pre-processing phase were done within Glue; this included things like timestamp transformations, validating coordinates, normalization of categorical fields, and preparing derived metrics for analysis.

This cleaned dataset was saved, after processing, back to Amazon S3 in Parquet format for optimized storage and fast querying for downstream loading into Amazon Redshift. The Glue job was completely automated and orchestrated through Apache Airflow to ensure reliable and repeatable data updates.

ETL for Airbnb Data (AWS Lambda + Step Functions)

The Airbnb dataset was a bit smaller, but it still required automation and cleaning. To do all the ETL jobs in a serverless environment, we implemented AWS Lambda functions that were orchestrated by AWS Step Functions.

The Step Function workflow, which is represented in our pipeline diagram, comprised three Lambda steps:

- UploadToS3 – took the raw Airbnb CSV file from the local drive and uploaded it to S3.
- uncheckedCleanAirbnb – fixed invalid coordinates, normalized prices, and filtered out duplicates or records with missing data.
- uncheckedTransformToDimFact – Took the cleaned data and converted into dimension and fact tables with a clear structure as shown in next section.

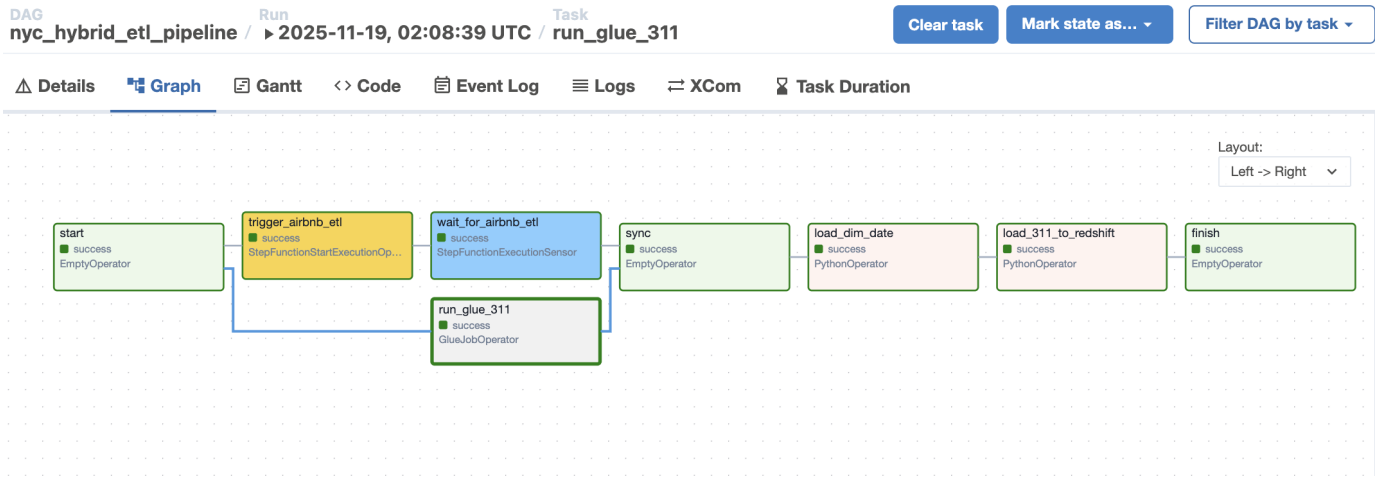


Fig. 2: Airflow DAG representing the orchestration of Airbnb and 311 ETL pipelines. The workflow automates data ingestion, transformation, and loading into Redshift through AWS Glue and Step Functions.

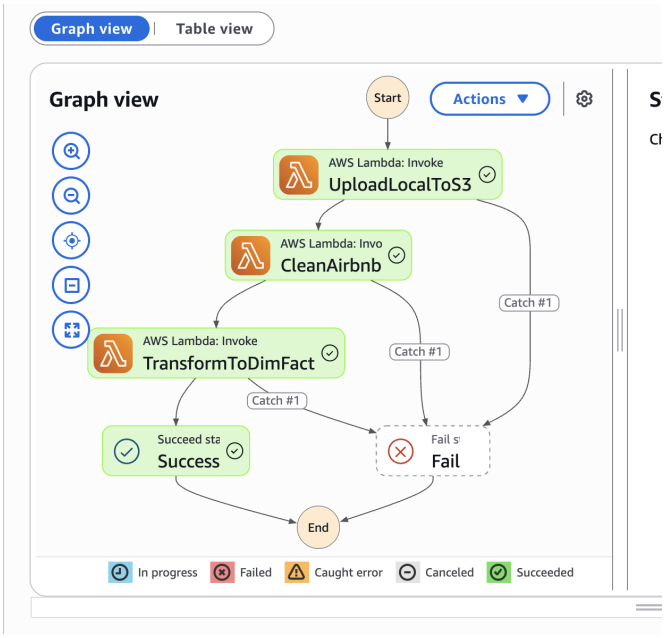


Fig. 4: ETL for Airbnb Data (AWS Lambda + Step Functions)

S3 to Amazon Redshift

After the cleaning of the data, the two datasets were uploaded from S3 into Amazon Redshift which we used as our data warehouse. With the help of the COPY command, data was imported from Parquet files to the structured tables in Redshift.

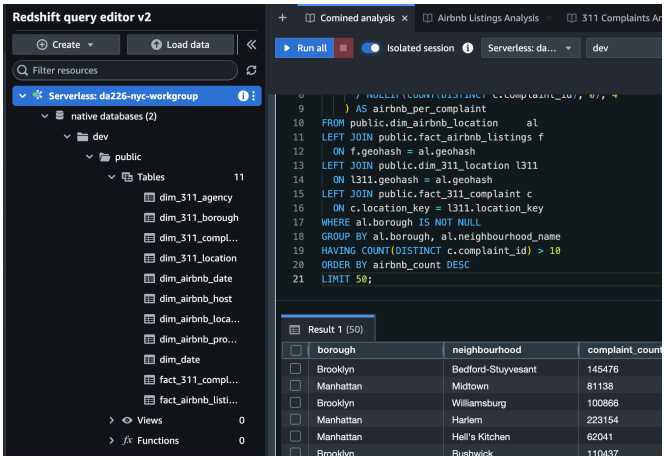


Fig. 5: Start Schema in AWS RedShift

We created a star schema (displayed below) with two main fact tables and a few dimension tables for the structured analysis:

- Fact Tables:
  - fact_311_complaint (311 service requests)
  - fact_airbnb_listings (Airbnb metrics)
- Dimension Tables:
  - dim_date – was used as a shared dimension across both datasets for time-based analysis
  - dim_location – contains geohash, borough, and coordinates
  - dim_host, dim_room_type, and dim_property – the Airbnb-specific dimensions
  - dim_agency, dim_complaint, and dim_borough – 311-specific dimensions

IX. DATA MODELING

In the data modeling phase, we focused on designing a star schema within Amazon Redshift to store, query, and analyze data from multiple sources—NYC 311 complaints and Airbnb listings. The goal was to create a structured and analytics-friendly warehouse model that supports fast joins and visualizations.

A. Schema Design

The model follows a star schema approach, consisting of:

- **Fact Tables:** Store measurable aggregates and transactional data.
- **Dimension Tables:** Store descriptive attributes for filtering, grouping, and reporting.

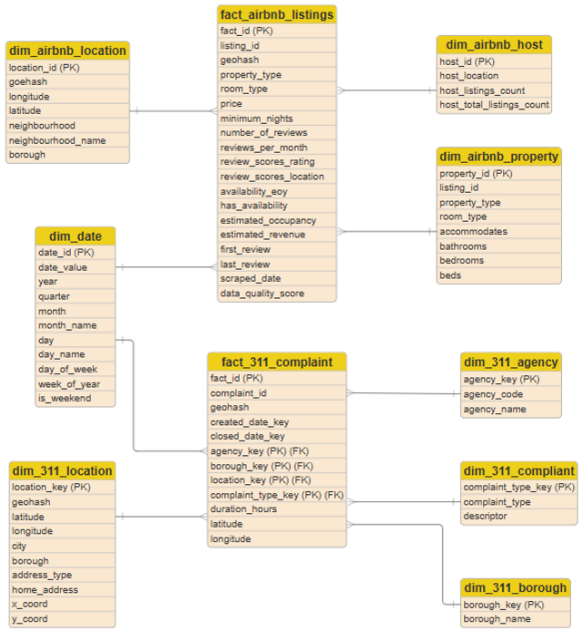


Fig. 6: Star schema model showing fact and dimension tables for NYC 311 and Airbnb datasets.

The schema diagram (Fig. 6) illustrates how the fact tables **fact_311_complaint** and **fact_airbnb_listings**—connect to their respective dimensions. Shared dimensions like **dim_date** and **dim_location** enable cross-domain analysis.

B. Fact Tables

fact_311_complaint: Contains detailed records of 311 complaints with references to dimensions for date, location, agency, borough, and complaint type.
Key Fields: complaint_id, created_date_key, location_key, agency_key, borough_key, complaint_type_key, duration_hours.
Measures: Complaint count, resolution duration.

fact_airbnb_listings: Stores Airbnb listing details and key aggregates such as price, review scores, and availability.
Key Fields: listing_id, geohash, property_type, room_type, price, availability_eoy, estimated_revenue_l365d.
Measures: Listing count, average price, estimated revenue.

C. Dimension Tables

- **dim_date:** Date, month, quarter, and weekday (shared dimension).
- **dim_311_borough:** Borough information and geohash.
- **dim_311_location:** Latitude, longitude, city, and geohash for spatial analysis.
- **dim_311_agency:** Agency details.
- **dim_311_complaint:** Complaint types and descriptions.
- **dim_airbnb_location:** Borough, neighbourhood, latitude, and longitude.
- **dim_airbnb_property:** Room type, property type, accommodation capacity.
- **dim_airbnb_host:** Host information and total listings.

D. Relationships

Each fact table is linked to its respective dimension tables via foreign keys:

- fact_311_complaint.location_key → dim_311_location. location_key
- fact_airbnb_listings.geohash → dim_airbnb_location. geohash
- fact_311_complaint.complaint_type_key → dim_311_complaint. complaint_type_key

This setup enables many-to-one relationships (many facts per dimension) and allows efficient roll-ups.

X. DATA ANALYSIS AND VISUALIZATION

After loading data into Redshift, we linked Amazon SageMaker for a deeper look at the relationships between Airbnb listings and complaint density. Jupyter notebooks were used to explore how complaint categories (e.g., noise or sanitation) changed with rental activity across neighborhoods. Finally, we created interactive dashboards in Amazon QuickSight to visualize key metrics and spatial trends.

A. 311 Complaints by Borough

The bar chart shows complaints across boroughs. Brooklyn tops the list with nearly two million complaints, followed by Queens, Bronx, and Manhattan with similarly high volumes. Staten Island and "Unspecified" record far fewer cases. Brooklyn's consistently high complaint numbers may reflect both its large population and high density of short-term rentals.

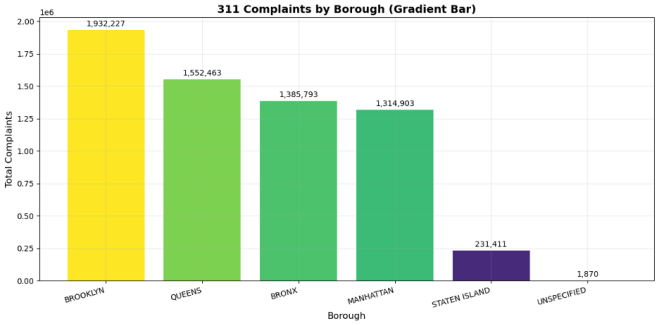


Fig. 7: 311 complaints by borough.

B. Top 311 Complaint Types

Illegal Parking and Noise–Residential are the two most frequent issues, followed by Heat/Hot Water problems. Noise-related complaints are consistently among the top, highlighting common livability concerns in densely populated and tourist-heavy areas.

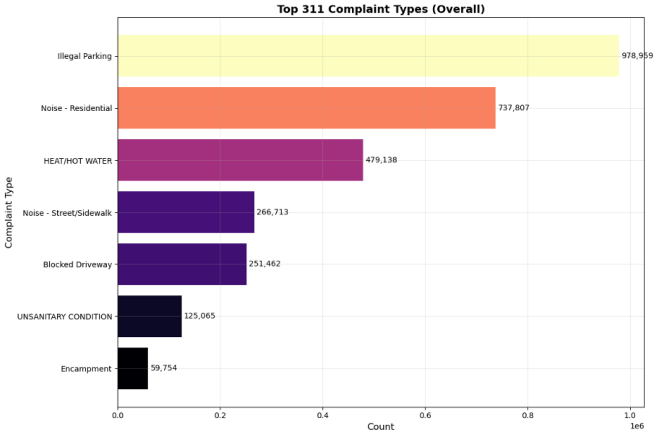


Fig. 8: Top 311 complaint types across NYC.

C. 311 Complaints vs. Airbnb Listings by Borough

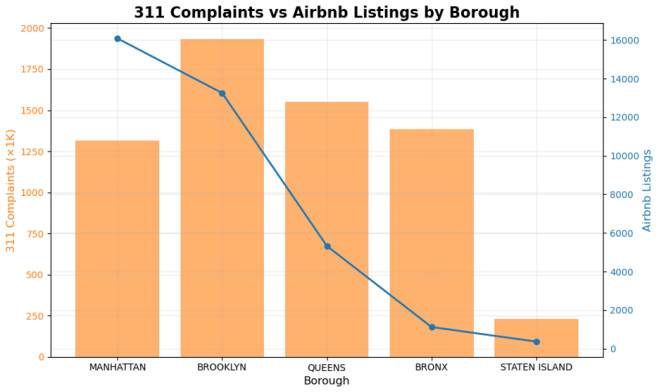
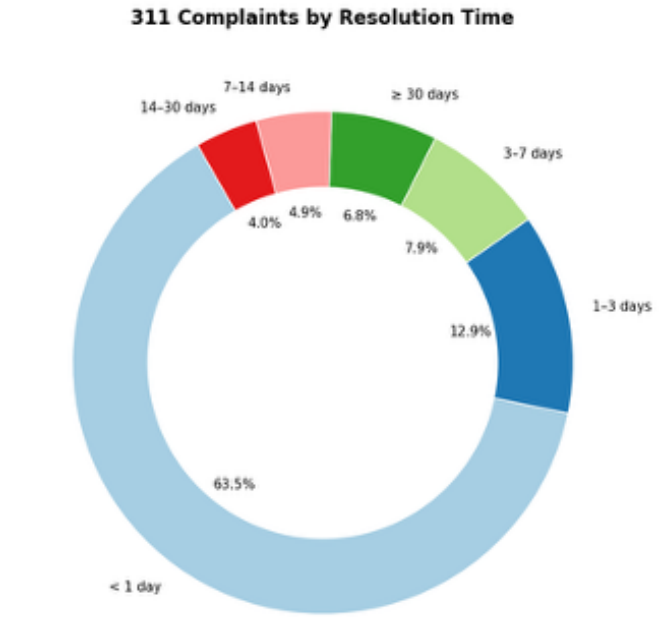


Fig. 9: 311 complaints vs. Airbnb listings by borough.

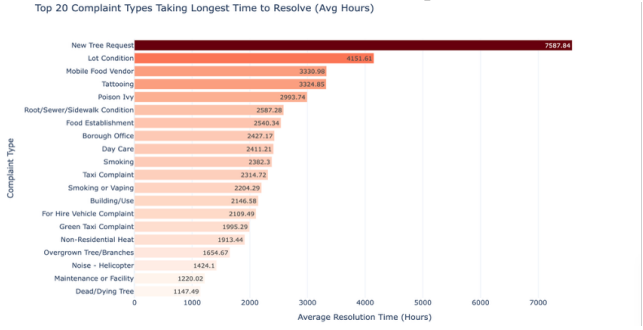
This combined bar and line chart compares complaint volumes (bars) with Airbnb listing counts (line). Manhattan and Brooklyn—the boroughs with the most Airbnb listings—also report

the highest number of 311 complaints, indicating a strong spatial relationship.

D. Resolution Time Distribution



(a) Resolution time distribution (pie chart).



(b) Average resolution time by complaint type.

Fig. 10: Distribution and averages of 311 complaint resolution times.

Approximately 75% of 311 complaints were resolved within three days, showing that city agencies respond quickly to most issues. Complaints resolved over longer timelines—two weeks to a month—were mostly related to categories such as *New Tree Request* and *Lot Condition*. The bar chart shows these longer-term categories clearly, while the pie chart highlights that most cases are closed in under three days.

E. Airbnb Count vs. Complaint Count and Ratio by Neighborhood

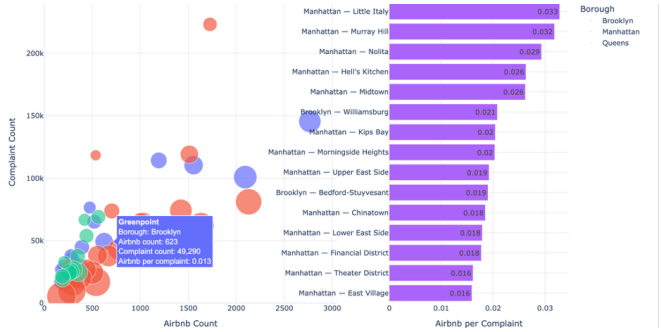


Fig. 11: Airbnb listings vs. 311 complaints by neighborhood.

We found a clear relationship between Airbnb listing density and 311 complaints. The bubble size represents the ratio of Airbnb listings to complaints. Manhattan again stands out as the most active borough, with higher Airbnb concentrations corresponding to higher complaint frequencies.

F. Spatial Distribution Map

(a) 311 complaints (red dots).

(b) Airbnb listings (blue dots).

Fig. 12: Spatial distribution of 311 complaints and Airbnb listings across NYC neighborhoods.

The two scatter maps illustrate the geographic overlap between 311 complaints and Airbnb listings. Areas dense with blue dots (Airbnb activity) tend to show high concentrations of red dots (complaints). Manhattan and parts of Brooklyn stand out as major hotspots, confirming the spatial relationship between short-term rentals and resident-reported disturbances.

XI. OPTIMIZATION TECHNIQUES

- **Incremental Loading:** Only new or modified entries are loaded into the fact and dimension tables. This approach avoids loading overhead, minimizes time, and ensures Redshift does not capture duplicate data.
- **Staging and Merging Pattern:** Data is first loaded into staging tables (`staging_311` and `staging_airbnb`).

From the staging tables, fact and dimension tables are populated. This design supports pre-validation of loads, clear traceability of data lineage, protects production tables from corruption, and enables easier rollback of errors.

- **Parquet File Optimization:** Data is stored in compressed columnar Parquet format and loaded via the Redshift `COPY` command from S3. This results in faster I/O, reduced S3 transfer costs, and better compression for analytical workloads.
- **Set-Based SQL Operations:** Inserts and updates use bulk SQL statements (`INSERT SELECT DISTINCT`, `LEFT JOIN + WHERE key IS NULL`) instead of row-by-row operations. Redshift's parallel processing optimizes these set-based operations, leading to substantial performance gains.
- **Airflow DAG Optimization:** The DAG design uses modular and idempotent tasks that allow isolation and reusability. Retry logic prevents duplication while ensuring fault tolerance, improving resource efficiency and orchestration.
- **Shared Dimension Reuse:** The `dim_date` table is shared across both Airbnb and 311 datasets, preventing duplicate calculations and ensuring temporal consistency across datasets.
- **Dimension Lookup Optimization:** Joins in the fact load use pre-cast dimension tables (`dim_agency`, `dim_311_location`). Keys are cached during fact inserts, avoiding multiple subqueries and reducing CPU overhead in Redshift.
- **Serverless and Distributed Architecture:** AWS Glue handles large-scale transformations, while AWS Step Functions orchestrate the Airbnb pipeline. Redshift Serverless allows elastic scaling—compute resources are allocated only when needed, optimizing cost and performance.
- **Data Cleaning Before Load:** Functions such as `NULLIF`, `TRIM`, and `UPPER` ensure consistency before ingestion into Redshift. This prevents key mismatches (e.g., "Manhattan" vs. "MANHATTAN") and enhances join efficiency.

XII. KEY INSIGHTS

1) **Airbnb Listings Correlate with 311 Complaints:** The more Airbnb listings in an area, the higher the number of complaints reported to 311—especially regarding noise and quality of life. This trend is strongest in Manhattan, where listings and complaints are most concentrated.
2) **Manhattan is the Hub:** Among all boroughs, Manhattan has the most properties and the highest complaint rate per capita. Short-term rental activity has a greater impact on livability here than in other boroughs.
3) **Majority of Complaints Are Resolved Quickly:** Approximately 75% of complaints are resolved within three days, indicating that city agencies respond efficiently to most issues. Common complaints such as illegal parking and residential noise are often resolved within one day.
4) **Longer Resolution Times Are Category-Specific:** Complaints taking two weeks to a month to resolve typically fall into categories like *New Tree Request* and *Lot Condition*, which require multi-agency coordination and fieldwork.
5) **Spatial Visualizations Confirm Patterns:** Scatter plots and density maps confirm that Airbnb hotspots overlap with complaint hotspots. Higher Airbnb density areas consistently show more 311 complaints.
6) **Data Pipeline and Visualization Worked as Intended:** The end-to-end data pipeline—built using AWS Glue, Redshift, MWAA (Airflow), and QuickSight—successfully processed, analyzed, and visualized large-scale public data. Interactive dashboards and analytics performed efficiently without errors.

XIII. TECHNICAL DIFFICULTY

With multiple public datasets, cloud ETL components, and layered analytics, this project presented several technical challenges across all stages of data collection, transformation, modeling, and visualization. The key challenges and solutions are summarized below:

- **Multiple Sources of Data:** Merging large public datasets such as NYC 311 complaints and InsideAirbnb required resolving schema discrepancies and structural inconsistencies. The 311 data was semi-structured JSON with inconsistent field names and timestamp formats, while InsideAirbnb listings were CSV/Parquet files containing nested text fields. Normalizing these datasets using geohash, borough, and neighborhood keys required extensive preprocessing and deduplication before standardization.
- **ETL Pipeline Orchestration:** Building an orchestrated ETL pipeline using Apache Airflow, AWS Step Functions, and AWS Glue required careful dependency management. Handling retries, transient network failures during S3 ingestion, and ensuring idempotent job execution added orchestration complexity. Fault tolerance and checkpointing at the DAG level ensured pipeline robustness.
- **Schema Design and Incremental Loads:** Implementing a star schema in Amazon Redshift with incremental loading was challenging. Each dimension and fact table required deduplication, surrogate key generation, and join optimization. Incremental inserts using `LEFT JOIN ... IS NULL` filters and handling slowly changing dimensions (SCD) had to avoid performance degradation.
- **Spatial and Temporal Mapping:** Mapping Airbnb listing coordinates to NYC boroughs and neighborhoods, and aggregating 311 complaints spatially and temporally, required complex spatial joins and time-based merges. Issues included missing or malformed coordinates and inconsistent timestamps across time zones.
- **Redshift Performance Tuning:** Loading data via `COPY` from S3 Parquet files required careful tuning of `DISTKEY` and `SORTKEY` for large joins. Performance optimization involved periodic `VACUUM` operations, reassessing distribution strategies, and optimizing queries for incremental updates.
- **Data Integration and Quality Assessment:** Both datasets had mixed-case fields, missing borough values,

and inconsistent timestamps. Applying transformations such as `TRIM`, `UPPER`, and `NULLIF` improved data consistency. Continuous validation maintained referential integrity across the star schema.

- **Analytical Correlation and Visualization:** Computing Pearson correlations and building Plotly/Mapbox dashboards for millions of records required efficient aggregation and caching. Ensuring fast query performance for Amazon QuickSight visualizations required materialized views and optimized SQL query design in Redshift.
- **Summary:** The major technical challenge was designing a scalable, fault-tolerant ETL system capable of integrating structured and semi-structured data into a unified warehouse. The solution supported spatial and temporal analytics for deriving meaningful urban livability insights from New York City data.

XIV. SIGNIFICANCE TO THE REAL WORLD

This project demonstrates how open data can be transformative when effectively utilized. By integrating NYC 311 complaint data with Airbnb listings, we revealed how tourism and short-term rentals influence urban livability. Complaints related to noise, illegal parking, or sanitation become more interpretable when analyzed alongside rental density.

Since both data sources are publicly accessible, insights from this project have practical value for residents, policymakers, and city planners. They can help target enforcement against illegal listings, allocate resources more efficiently, and shape regulations for balancing tourism with community well-being.

XV. LESSONS LEARNED

Working on this project extended beyond technical implementation—it emphasized the realities of working with complex, imperfect data. We learned to handle API limits, incomplete records, schema drift, and the need for robust automation. Setting up and orchestrating tools such as MongoDB Atlas, AWS Glue, and Amazon Redshift gave us hands-on experience with professional-grade data pipelines.

These lessons reinforced principles of reproducibility, data validation, and pipeline resilience. We also gained an appreciation for cross-functional collaboration and the patience required to handle unpredictable real-world datasets.

XVI. INNOVATION

What distinguishes this project is the integrated use of multiple technologies to answer a meaningful civic question. The automated pipeline collects open data from NYC APIs, stages it in MongoDB, transforms it with AWS Glue, loads it into Amazon Redshift, and visualizes the outcomes in Amazon QuickSight.

For exploratory analysis, Amazon SageMaker was used to compute correlations and develop advanced metrics. This multi-platform workflow exemplifies how cloud ecosystems can enable near–real-time urban analytics—bridging open data and policy insights.

XVII. NEW TOOL COVERED IN HE PROJECT

*a) **Luigi**:* In this project, a Luigi-based data pipeline was used to tokenize complaint records from the NYC 311 dataset. The pipeline first extracts complaint text from the raw JSON files, then uses the tiktoken library (compatible with ChatGPT) to tokenize each complaint. Both the tokenized results and token counts are saved to output files.

The pipeline is designed to run automatically, generate all necessary output files, and provide warnings if no valid complaints are found in the input data. This approach enables scalable and reproducible preprocessing and tokenization of large complaint datasets in a batch-wise manner.

*b) **Lambda + Step Functions as ETL Tool**:* In this project, we used AWS Lambda and AWS Step Functions to build a serverless ETL pipeline. Lambda handled individual tasks such as data extraction, cleaning, and loading, while Step Functions managed the workflow and task sequence.

This approach was cost-effective, since it only ran when needed, and scalable, as AWS automatically adjusted resources based on the workload. It also integrated easily with other AWS services such as Amazon S3, AWS Glue, and Amazon Redshift,enabling a seamless data processing and analytics workflow.

XVIII. VERSION CONTROL

Our code management was completely handled using Git and GitHub to maintain an organized, transparent, and collaborative workflow. Each component of the pipeline was developed in its own branch, merged after testing and peer review, and tracked through descriptive commit messages. This branching strategy prevented accidental overwrites and allowed multiple contributors to work simultaneously without conflicts. Every commit served as a record of progress, helping to keep the project clean, reproducible, and auditable.

- https://github.com/Sbnikitha/ADI-226-Datawarehouse-project

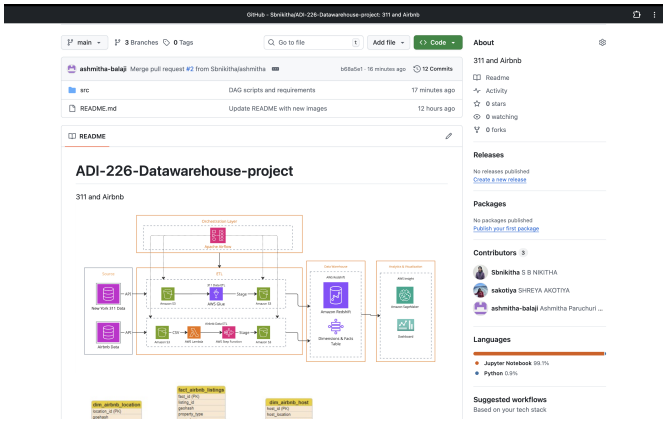


Fig. 13: GitHub repository and branching workflow illustrating collaborative development and version control.

XIX. TEAMWORK

Collaboration was at the core of the project's success. Each team member contributed to distinct aspects like API extrac-

tion, cloud setup, ETL pipeline engineering, analysis, and visualization.

We used GitHub for code sharing and version management, held regular check-ins to track progress, and collectively debugged issues as they arose. This collaborative, open approach fostered knowledge sharing, efficiency, and accountability. Individual contributions combined seamlessly into a unified, functional data platform.

XX. ANALYTICS SUPPORT BUSINESS DECISIONS

By analyzing patterns between Airbnb activity and 311 complaints, we were able to identify how short-term rentals might be affecting neighborhood well-being and city operations.

For example, through data visualization and trend analysis, we found that areas with a higher density of Airbnb listings often showed more noise, sanitation, and illegal parking complaints. This kind of insight can help city planners and policymakers decide where to tighten short-term rental regulations or where to allocate more public resources, such as sanitation or noise enforcement teams.

From a business perspective, the analytics also revealed how Airbnb hosts and tourism-driven activities influence local economies and housing availability.

XXI. PRACTICED AGILE/SCRUM

As part of the project workflow, we followed the Agile methodology using a Scrum-based approach. The work was divided into three main sprints, each focusing on a specific phase of the project: setup and data collection, ETL and warehouse build, and analytics with final submission. Regular sprint planning and review meetings were conducted over Google Meet to monitor progress and ensure timely delivery. (https://trello.com/b/sLe3mkUC/nyc-311-airbnb-data-warehouse-project)

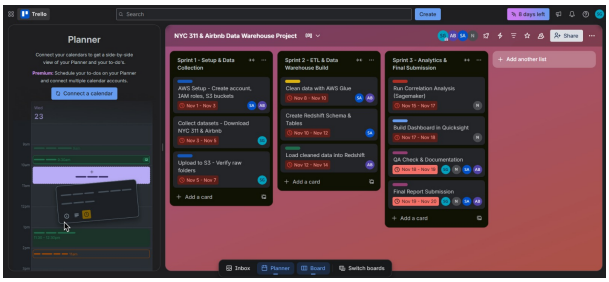


Fig. 14: Kanban board showing sprint progress and task tracking.

Shreya acted as the Scrum Master, while the other members contributed as team members. Retrospective discussions were held at the end of each sprint to evaluate progress, identify challenges, and plan improvements for the next phase. Here is the attached screenshot of our Kanban board:

XXII. Practiced Pair Programming

As a part of the collaborative workflow, our team held weekly meetings via google meet to track the progress, discuss issues and difficulty faced at each stage. In these meetings one member would share their screen and present the ongoing work while others reviewed and provided feedback. We have added a meet screenshot where one of our team members presenting the AWS Glue job script used to merge the 311 Airbnb dataset. During this call, team members were actively involved to discuss the data integration logic. This pair programming style helped ensure that everyone understood the implementation and contributed to improving the overall code quality.

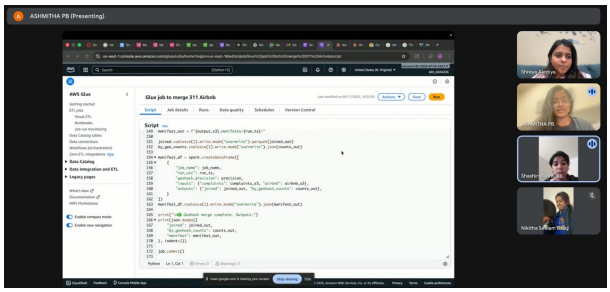


Fig. 15: Code review in Zoom.

XXIII. CREDIT TAXONOMY

TABLE I: Contributor Roles and Responsibilities (CRediT Taxonomy)

Contributor Role	Team Member(s)
Literature Survey	Nikitha Seelam Balaji
Project Proposal	Shreya Akotiya, Shashira Guntuka, Nikitha Seelam Balaji, Ashmitha Paruchuri Balaji
Logical Planning	Ashmitha Paruchuri Balaji, Shreya Akotiya
Methodology	Shreya Akotiya
Data Import	Ashmitha Paruchuri Balaji
Data Processing and Cleaning	Nikitha Seelam Balaji, Shreya Akotiya
ETL Pipeline	Ashmitha Paruchuri Balaji, Shreya Akotiya
GitHub Maintenance & Scrum Planning	Nikitha Seelam Balaji, Shashira Guntuka
Dimension Modelling	Ashmitha Paruchuri Balaji
Visualization	Nikitha Seelam Balaji
Validation	Shreya Akotiya
Writing – Original Draft	Shreya Akotiya
Writing – Review & Editing	Shreya Akotiya, Ashmitha Paruchuri Balaji, Nikitha Seelam Balaji, Shashira Guntuka

References

[1] NYC Open Data, "311 Service Requests," https://data.cityofnewyork.us/.

[2] Inside Airbnb, "Get the Data," http://insideairbnb.com/get-the-data.html.

[3] Amazon Web Services, "AWS Glue Documentation," https://docs.aws.amazon.com/glue/.

[4] Amazon Web Services, "Amazon Redshift Documentation," https://docs.aws.amazon.com/redshift/.

[5] Amazon Web Services, "Amazon QuickSight Documentation," https://docs.aws.amazon.com/quicksight/.

[6] Apache Software Foundation, "Apache Airflow Documentation," https://airflow.apache.org/docs/.