Predicting Chicago Food Inspection Failures

Jingzhi Yang Lingyue Ji Nate Assefa

1. Data Description

1.1. Objective

Our research question is clear and focused: *Can we predict, at the moment an inspection occurs, whether a Chicago food facility will fail?* This question specifies both the prediction target, which is inspection failure, and the timing constraint, that only pre-inspection information may be used.

The City of Chicago Food Inspections dataset is well suited to this goal. It provides detailed facility characteristics, inspection metadata, location, and free-text violation notes. These fields capture operational risk factors and historical outcomes that are essential for modeling future failures.

Our research strategy is a supervised machine learning approach. We first clean and normalize the raw data, engineer features such as an ordinal risk score and calendar variables, and then train classification models using the cleaned training set while holding out a test set for final evaluation. This strategy matches the research question because it builds a predictive model while respecting the constraint that only information available before the inspection is included. The public documentation and data dictionary of the dataset serve as the primary codebook and reference for variable definitions.

1.2. Team Collaboration

Lingyue found the raw data, exported cleaned training and testing CSV files, and created the repository. Nate cleaned the data, checked schema, and performed visualization. Jingzhi managed the files, created the data dictionary, and wrote the report.

1.3. Variables

The dataset contains both original inspection fields and engineered features for modeling. Key original variables include:

- **inspection_id**: Unique identifier for each inspection event
- **inspection_date**: Calendar date of the inspection; used for time-based analysis
- **facility_type**: Establishment category (e.g., restaurant, grocery store, school) capturing operational risk
- **risk**: City-defined risk level with three categories: Risk 1 (High), Risk 2 (Medium), Risk 3 (Low)
- **inspection_type**: Reason for inspection (e.g., canvass, complaint, license, re-inspection)
- **zip**, **city**, **latitude**, **longitude**: Facility location variables for geographic analysis
- **results**: Official outcome string (pass, pass w/ conditions, fail, or out of business); defines the target label.
- **violations**: Free-text list of cited health-code violations, often semi-structured with numbered items

We also create engineered features to support modeling:

- **viol_count**: Count of numbered violations parsed from the free-text field
- **risk_ord**: Ordinal encoding of risk (High=3, Medium=2, Low=1)
- **ins_year**, **ins_month**, **ins_wday**: Year, month, and weekday extracted from the inspection date
- **target_fail**: Binary target variable; equals 1 if the inspection result is "fail," 0 otherwise

These variables together describe the facility, its risk context, the inspection circumstances, and outcomes. They form the basis for predicting inspection failure while avoiding data leakage by excluding violation text from the predictive model.

1.4. Data Cleaning

Each data split contains 1,000 inspections. After cleaning, the tables hold 18 columns, and about 19% of training records show a failed inspection. Facility type has about 41 categories, with restaurants most common. Inspection type has about 19 categories, and risk has three. Coordinates cluster tightly around Chicago (about 41.88° N, −87.67° W). Years range from 2010 to 2025 with even coverage. Failure rates rise with higher risk, though risk 2 shows a slightly higher rate than risk 3.

Cleaning follows a single reproducible pipeline. We drop duplicate inspection IDs and standardize text. City names convert to title case, and a new field flags Chicago locations. ZIP codes convert from floats to five-digit strings with leading zeros. Dates parse to `datetime` objects, and we derive year, month, and weekday. Result strings normalize to a compact set ("pass," "pass w/ conditions," "fail," and similar). We encode the risk field to an ordinal integer. Violation notes become a numeric count using a layered parsing strategy. Latitude and longitude outside conservative Chicago bounds become missing.

1.5. Challenges

The first dataset we considered was from Kaggle and was eight years ago. After discussion, we switched to the current dataset, which is more up-to-date.

Text fields contained inconsistent casing and spelling, so we standardized them. Some cells contained the string "nan," which we converted to missing. ZIP codes required conversion from float to string. Violation text is semi-structured, so we used a conservative counting method. The test set includes facility types not present in training, so we map unseen categories to an "Other" bucket and use encoders that tolerate unknown levels.

References

City of Chicago. *Food Inspections | City Of Chicago | Data Portal*, September 2025. Retrieved September 26, 2025, from https://data.cityofchicago.org/Health-Human-Services/Food-Inspections/4ijn-s7e5/data_preview.