Predicting New York City Restaurant Food Safety Violations

Jonathan A Toy, Julie C Helmers, Seda Bilaloglu

New York University Center for Data Science

Advisor: Dr. Bonnie Ray

Center for Data Science

Introduction

Goal:

- To classify New York City restaurants according to whether they will have at least two critical food safety violations on their next inspection.

Motivation:

- Chicago has been successful in tackling this prediction problem to more intelligently dispatch food safety inspectors.

Data Sets Used

Primary data set:

- DOHMH inspection results from NYC OpenData
  - Each row = a single violation found on an inspection of a restaurant
  - Includes restaurant name/information, restaurant location, inspection date, violation type/reason, and inspection grade and letter rating if available

Additional data sets:

- 311 calls from NYC OpenData
  - Only looked at complaints with type of food establishments, food poisoning, rodents, dirty conditions, missed trash collection, electricity, safety, police matter, or general
- Temperature and humidity data from the Weather Underground archives
- Liquor and sidewalk cafe license information from NYC OpenData
- Google business ratings of restaurants from Google Places API

Data Preprocessing and Feature Generation

- Inspection results were condensed by restaurant to so that each row represented a restaurant and its inspection history, rather than a single violation. We also excluded restaurants with only one inspection.
- We extracted aggregate features like total number of inspections, time since last inspection, average critical and non-critical violations per inspection, and fraction of inspections yielding 2+ critical violations.
- The latitude and longitude of the restaurant's location were calculated from the address by using OpenStreetMap search engine, Nominatim. Nominatum was not able to convert 10% percent of the addresses, so for those the Google API (2500 request limit) was used.
- For the 311 complaints, we calculated the number of complaints placed within 0.3 km (~3 streets) radius of a restaurant within the last year.
- Three-day average temperature/humidity values in the restaurant's zipcode were retrieved using Python's BeautifulSoup package to crawl www.wunderground.com. Central Park info was used for missing values.
- Restaurant ratings were retrieved using the Google Places API. Zipcode-level and city-level means were used for missing ratings.
- Numerical features were normalized to fall in the interval [0,1].

Modeling and Performance Evaluation

Evaluating performance and measuring success:

- As 66% of the restaurants in our dataset had 0 or 1 critical violations on their most recent inspection, our baseline algorithm was to predict the majority class (less than 2 critical violations) for all restaurants.
- As our goals included both identifying problematic restaurants and conserving limited DOHMH resources, we used **accuracy, recall, precision, F_1 score, and ROC curves** to evaluate model performance.

Improving performance:

- We engineered features from our secondary data sets.
- We tested several models, including logistic regression, naïve Bayes, and random forest, with various parameters.
- The default 50% cutoff threshold for predictions led to significant skew towards precision at the expense of recall, so we examined ROC curves to manually set cutoff thresholds.

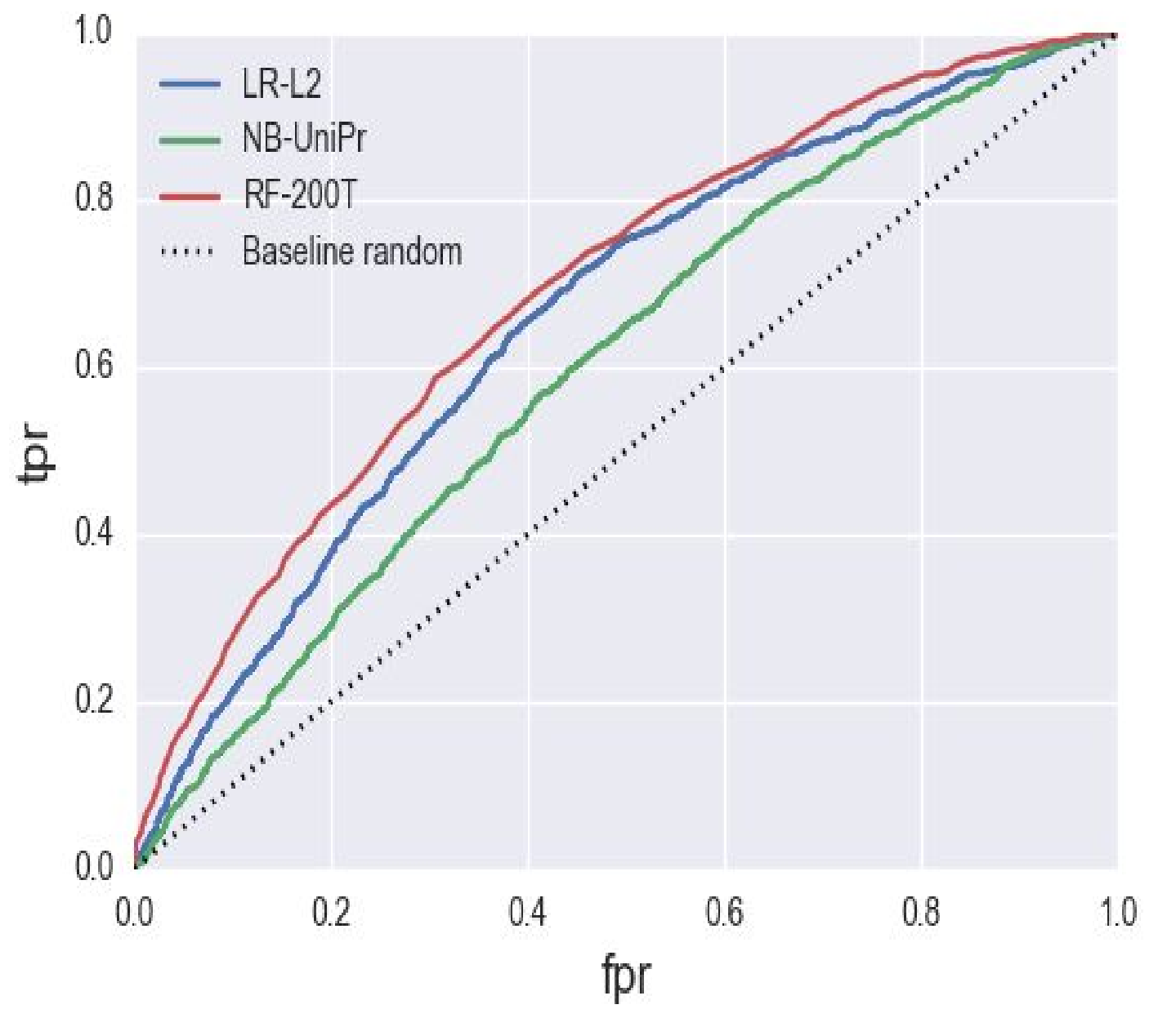
Challenges encountered:

- New features were computationally costly to obtain.
- Many of the new features had data quality issues (missing values).
- Some features were less important than we expected (see below).

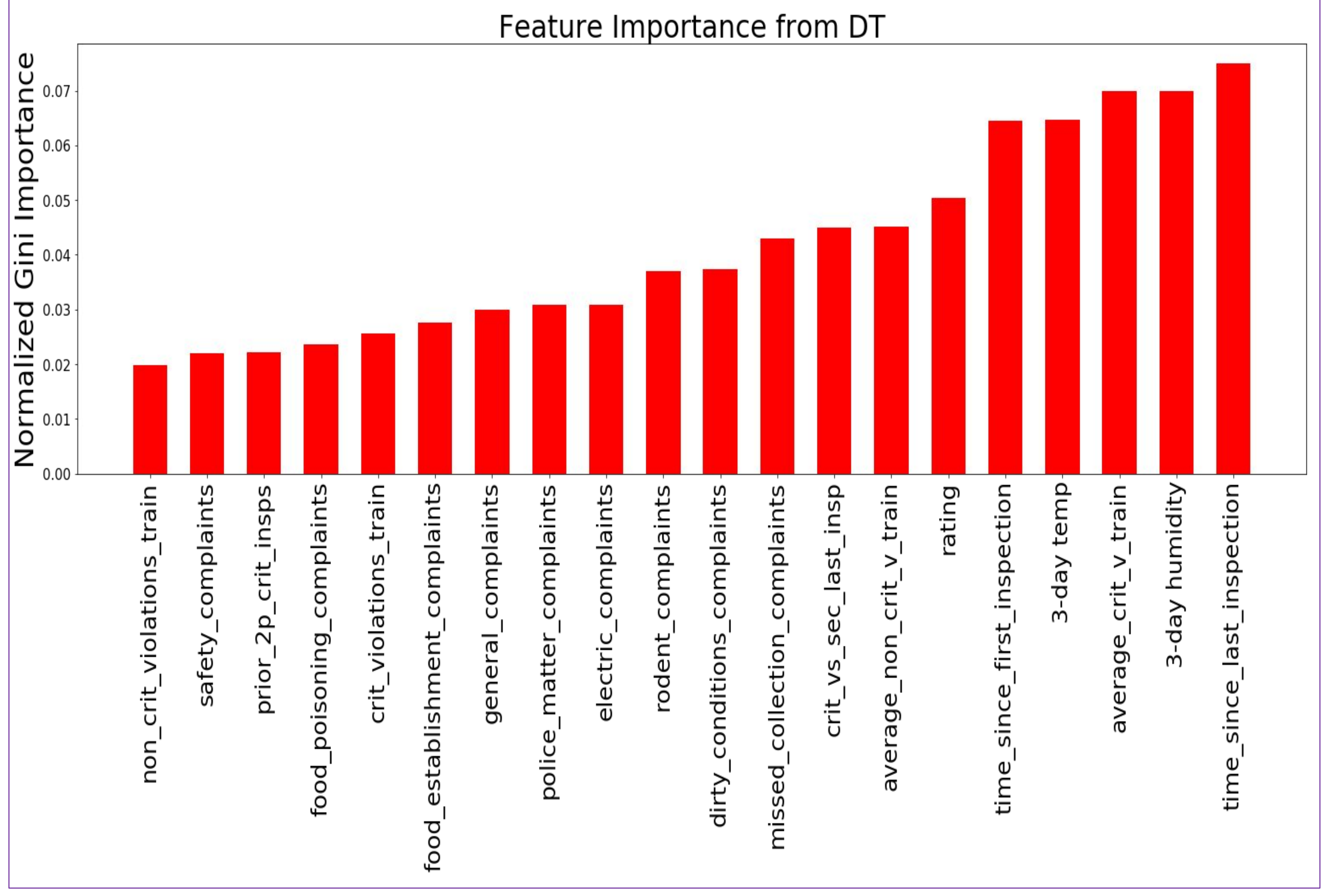
Winning Model

RF with 200 trees

The random forest model had recall and precision that were globally optimal to the NB and LR models. The 200 tree cutoff was chosen to take into account computational resources.

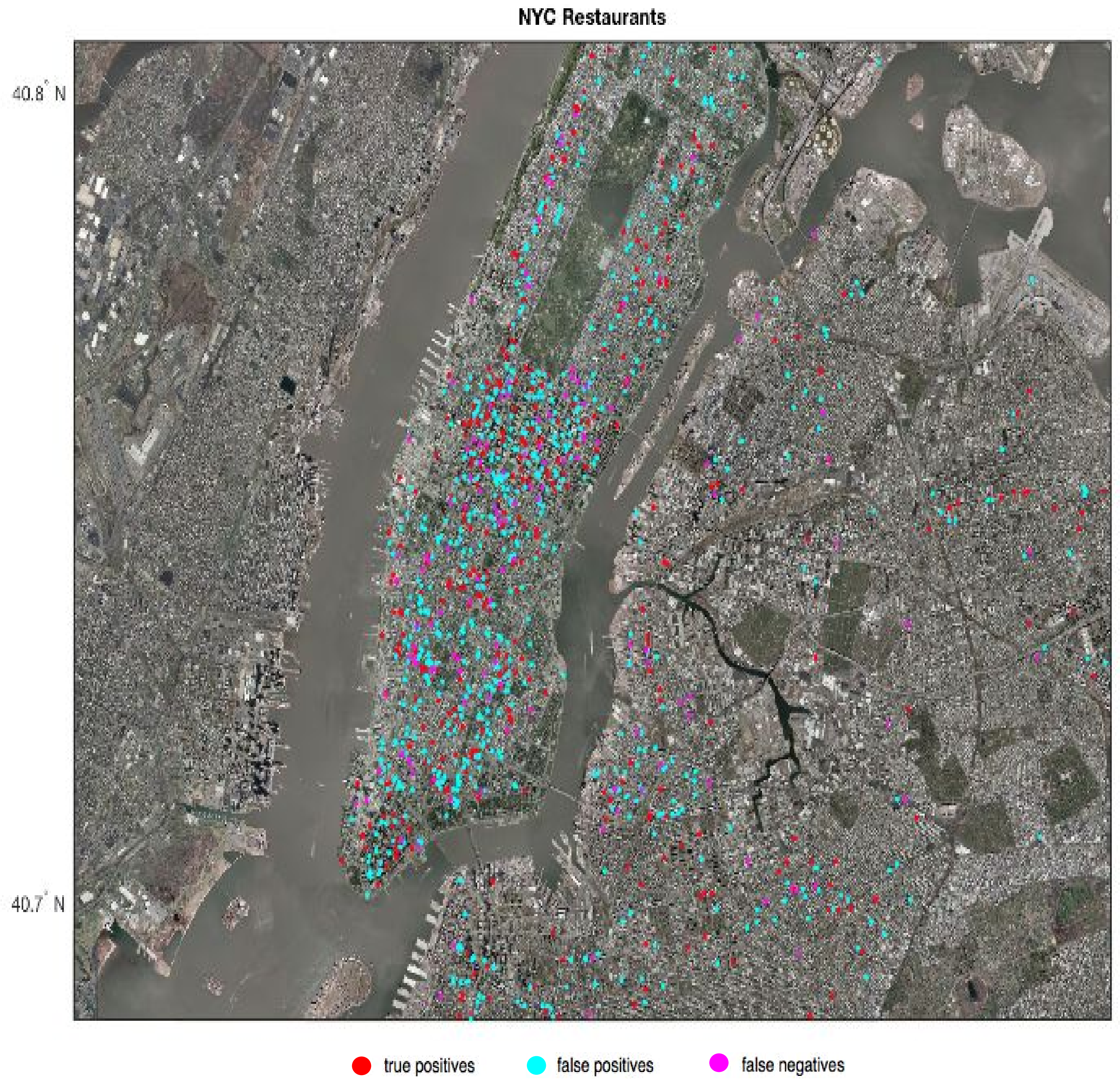


Feature Comparison



Results

- If we set the classification threshold to be 30.5% rather than 50% likely to have 2+ critical violations, we are able to to correctly flag 75% of the offending restaurants while only visiting approximately 59% of all restaurants that are due for an upcoming inspection.
- If inspectors prioritize visiting places in the order of the rankings from our model, they will have 41% of their time left to find the remaining 25% offending restaurants the model didn't identify.



Future Work and Deployment

- Using restaurant data from TripAdvisor and Yelp, such as price range and noise level, could help improve performance.
- We'd like to add information about restaurant size/staffing. (We weren't able to find publicly available datasets for number of employees, etc.)
- Open question: How much are restaurant inspection results impacted by inspector who conducts them? Chicago incorporated inspector ID.

Bibliography

Data Sources

- NYC OpenData https://opendata.cityofnewyork.us/
- Weather Underground Historical Weather https://www.wunderground.com/history/
- OpenStreetMap http://nominatim.openstreetmap.org/
- Google Places API

Project Inspiration

- Chicago Food Inspections Evaluation https://chicago.github.io/food-inspections-evaluation/
- Open Data Nation FIVAR http://www.opendatanation.com/