# Inspection Problems

# Project typology

- **Early Warning**: Can I detect which children are going to get lead poisoning early?
- **Resource Prioritization**: Can I prioritize homes to inspect with potential health and safety violations?
- **Scheduling and Assignment**: Can I determine which medics/ambulances/fire trucks to schedule/assign to respond to an incident?
- **Routing**: Can I determine which office/department to route an incoming service request/311 call to?
- **Policy Recommendations**: Can I recommend which policies to modify to improve maternal mortality?
- **Impact Evaluation**: Can I evaluate the impact of a new policy?
- **Data Creation**: Can I turn this text/images/video/audio I have collected in to “data” i can analyze?

# Three Examples

- CDPH
- NYC PEU
- DT

# Problem 0: Cost of Inspection

- Inter-inspector reliability issues
- Labels are costly to collect
  - Are there less costly options?

# Problem 1: Metrics

- Time constraints
- Type constraints
  - E.g. mining, manufacturing
- Space constraints

# Problem 2: Missing Labels

- Train set vs test set
  - Test set imputation requires strong assumptions about what the labels would have been. Our test set should reflect what the partner expects to see, e.g. if we assign zeroes to missing labels, model assessment will be wildly incorrect
- Look at other metrics

# Problem 2: Missing Labels

- Train Set (supervised / semi-supervised / un-supervised)
  - 0 label imputation: assumes none of the missed inspections would have found violations
  - 1 label imputation: assumes all of the missed inspections would have found violations
  - Smarter imputation

# Problem 3: Prediction Schedule

- When are the lists made?

# Problem 4: Prediction Window

- Lead vs. NYC/DT

# Problem 5: Confidence in Predictions

- Ideally, we would model something like Pr(violation)
- We cannot model Pr(violation) directly because we only have labels for inspections. We can model Pr(violation | inspection).
- If Pr(violation | inspection) ~ Pr(violation), as would happen under random sampling, we get what we want. The bigger the difference, the less confidence we can have in how our model would perform on non-inspected entities. Can get this multiple ways:
    - Most common: model Pr(inspection) with a logit, probit, etc.
    - Less common (but perhaps worth trying): clustering. If an entity looks like a bunch of inspected entities on the feature space, we can have more confidence in whether an inspection would find a violation

# Problem 6: Implementation

- All exploration: best baseline, no learning about the model
- All exploitation: select/assess models on cases with labels, no idea about cases missing labels
- Can use a combination of exploration and exploitation to get benefits of both

- If interventions don’t work, exploration results don’t change
- If the world is stable and the interventions work well, exploration % declines. Exploitation % first increases and then decreases.