Using Machine Learning to Target Assistance: Identifying Tenants at Risk of Landlord Harassment

Rebecca Johnson [1,2], Teng Ye [3], Samantha Fu [4], Jerica Copeny [5], Bridgit Donnelly [6], Alex Freeman [7], Mirian Lima [8], Joe Walsh [8], and Rayid Ghani [8]

[1] Visiting Data Scientist, The Lab at DC, [2]Assistant Professor (Quantitative Social Science), Dartmouth College (July 2020)

[3]University of Michigan, [4]Evansville Public Library, [5]London School of Economics,

[6]Mayor's Public Engagement Unit (former), [7]Mayor's Public Engagement Unit,

[8]Center for Data Science and Public Policy

DS at DC: 10.16.19

Overview

- **Substance:** Background on why NYC agency we partnered with– the Public Engagement Unit (PEU) and their Tenant Support Unit (TSU) within NYC's Mayor De Blasio's – 1) wanted to use machine learning to target outreach efforts to tenants; and 2) viewed these outreach efforts as important in combating housing instability before it reached the point of eviction/homelessness: https://dl.acm.org/citation.cfm?id=3332484

Overview

- **Substance:** Background on why NYC agency we partnered with– the Public Engagement Unit (PEU) and their Tenant Support Unit (TSU) within NYC's Mayor De Blasio's – 1) wanted to use machine learning to target outreach efforts to tenants; and 2) viewed these outreach efforts as important in combating housing instability before it reached the point of eviction/homelessness: https://dl.acm.org/citation.cfm?id=3332484
- **Code:** Repository is private but I made public-facing version with 1) all our `utils` (Python helper functions we wrote to help make various ETL, preprocessing, and model estimation tasks more efficient), and 2) example code from some of the pipeline steps: https://github.com/rebeccajohnson88/sharing_ml_landlord

Overview

- **Substance:** Background on why NYC agency we partnered with– the Public Engagement Unit (PEU) and their Tenant Support Unit (TSU) within NYC's Mayor De Blasio's – 1) wanted to use machine learning to target outreach efforts to tenants; and 2) viewed these outreach efforts as important in combating housing instability before it reached the point of eviction/homelessness: https://dl.acm.org/citation.cfm?id=3332484
- **Code:** Repository is private but I made public-facing version with 1) all our `utils` (Python helper functions we wrote to help make various ETL, preprocessing, and model estimation tasks more efficient), and 2) example code from some of the pipeline steps: https://github.com/rebeccajohnson88/sharing_ml_landlord
- **Discussion to brainstorm similar applications in DC**: agencies have limited time/resources and want to direct those resources to the people they can help the most or the places that have the highest risk of problems

Overview

- **Substance:** Background on why NYC agency we partnered with– the Public Engagement Unit (PEU) and their Tenant Support Unit (TSU) within NYC's Mayor De Blasio's – 1) wanted to use machine learning to target outreach efforts to tenants; and 2) viewed these outreach efforts as important in combating housing instability before it reached the point of eviction/homelessness: https://dl.acm.org/citation.cfm?id=3332484
- **Code:** Repository is private but I made public-facing version with 1) all our `utils` (Python helper functions we wrote to help make various ETL, preprocessing, and model estimation tasks more efficient), and 2) example code from some of the pipeline steps: https://github.com/rebeccajohnson88/sharing_ml_landlord
- **Discussion to brainstorm similar applications in DC**: agencies have limited time/resources and want to direct those resources to the people they can help the most or the places that have the highest risk of problems
  - **Older approaches:** judgment or simpler checklists (e.g., VI-SPDAT)

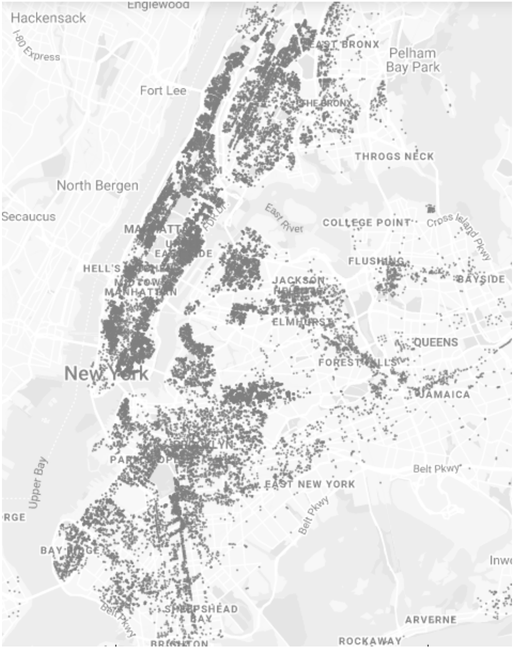
Overview

- **Substance:** Background on why NYC agency we partnered with– the Public Engagement Unit (PEU) and their Tenant Support Unit (TSU) within NYC's Mayor De Blasio's – 1) wanted to use machine learning to target outreach efforts to tenants; and 2) viewed these outreach efforts as important in combating housing instability before it reached the point of eviction/homelessness: https://dl.acm.org/citation.cfm?id=3332484
- **Code:** Repository is private but I made public-facing version with 1) all our `utils` (Python helper functions we wrote to help make various ETL, preprocessing, and model estimation tasks more efficient), and 2) example code from some of the pipeline steps: https://github.com/rebeccajohnson88/sharing_ml_landlord
- **Discussion to brainstorm similar applications in DC**: agencies have limited time/resources and want to direct those resources to the people they can help the most or the places that have the highest risk of problems
  - **Older approaches:** judgment or simpler checklists (e.g., VI-SPDAT)
  - **Newer approaches:** predictive models

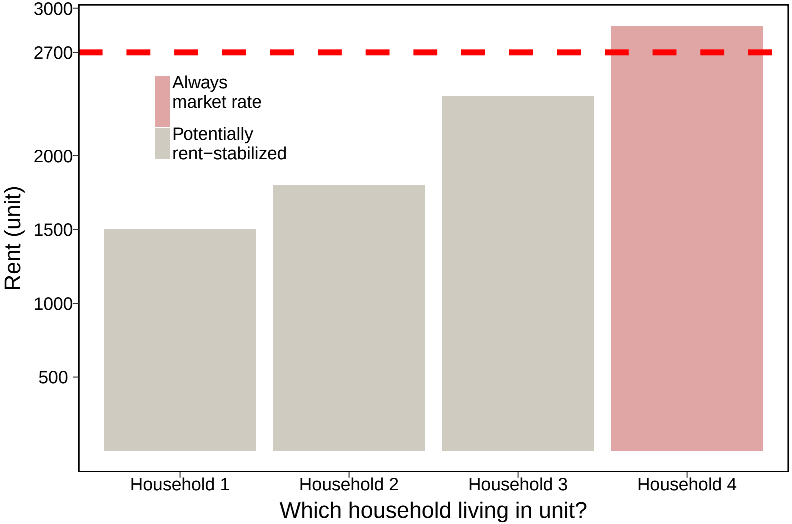
Overview

- **Substance:** Background on why NYC agency we partnered with– the Public Engagement Unit (PEU) and their Tenant Support Unit (TSU) within NYC's Mayor De Blasio's – 1) wanted to use machine learning to target outreach efforts to tenants; and 2) viewed these outreach efforts as important in combating housing instability before it reached the point of eviction/homelessness: https://dl.acm.org/citation.cfm?id=3332484
- **Code:** Repository is private but I made public-facing version with 1) all our `utils` (Python helper functions we wrote to help make various ETL, preprocessing, and model estimation tasks more efficient), and 2) example code from some of the pipeline steps: https://github.com/rebeccajohnson88/sharing_ml_landlord
- **Discussion to brainstorm similar applications in DC**: agencies have limited time/resources and want to direct those resources to the people they can help the most or the places that have the highest risk of problems
    - **Older approaches:** judgment or simpler checklists (e.g., VI-SPDAT)
    - **Newer approaches:** predictive models
        - Lab at DC: past projects on housing code violations; rate inspections; future on student absenteeism

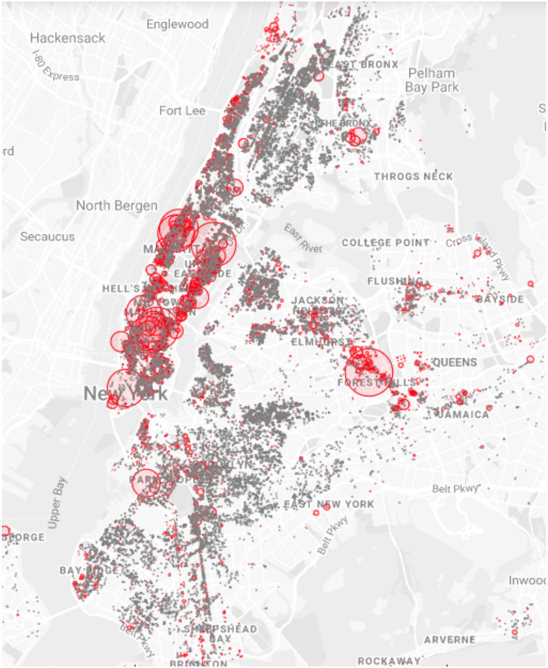
Rent stabilization is an important policy lever against housing instability...



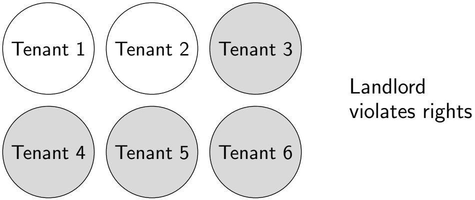
...But some landlords exploit legal loopholes to convert rent-stabilized apartments to market-rate ones



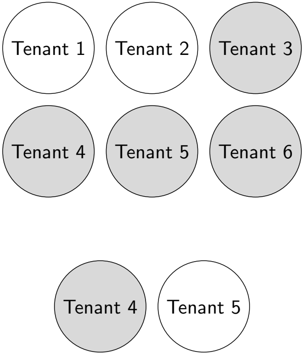
Contributing to conversion of over 38,000 rent-stabilized apartments to market-rate ones (2007-2015)



Reactive rights enforcement can lead to biases in which tenants receive help to combat landlord harassment



Reactive rights enforcement can lead to biases in which tenants receive help to combat landlord harassment

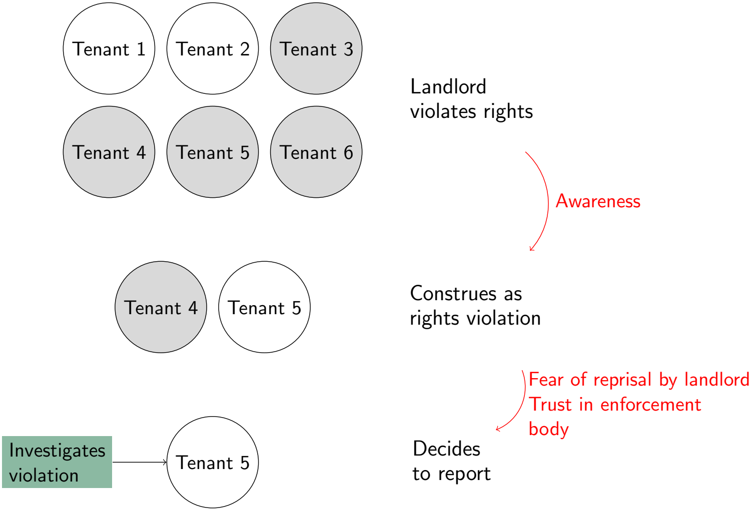


Landlord violates rights

Awareness

Construes as rights violation

Reactive rights enforcement can lead to biases in which tenants receive help to combat landlord harassment



Potential way to ameliorate biases: *proactive* rights enforcement

When it comes to protecting tenants and affordable housing, we don't wait for a 311 call to come in.

Potential way to ameliorate biases: *proactive* rights enforcement

When it comes to protecting tenants and affordable housing, we don't wait for a 311 call to come in.

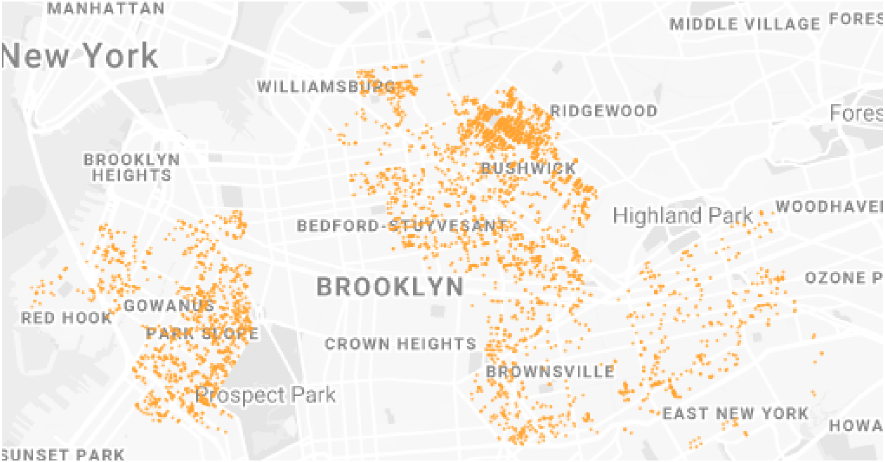
Potential way to ameliorate biases: *proactive* rights enforcement

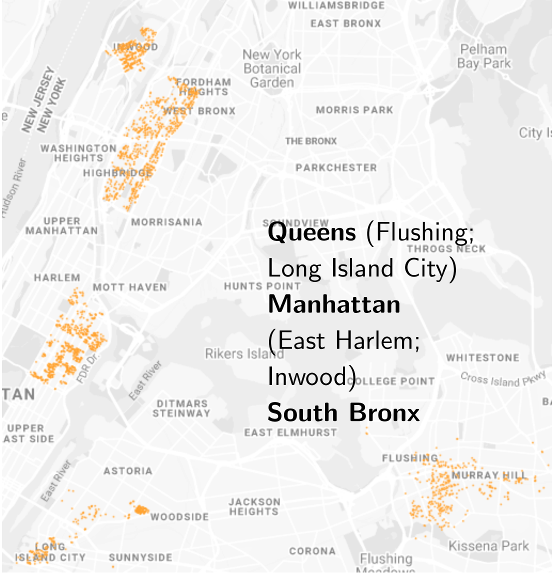
When it comes to protecting tenants and affordable housing, we don't wait for a 311 call to come in.

We have teams [Tenant Support Unit (TSU)] knocking on doors in fast-changing neighborhoods to solve problems then and there. (Mayor Bill de Blasio, 2016)

How should the Tenant Support Unit prioritize visits among ~ 6500 buildings containing $\sim 142{,}000$ residential units?

Brooklyn (Bushwick; Gowanus; East New York)





Queens (Flushing; Long Island City)

Manhattan (East Harlem; Inwood)

South Bronx

Have thus far gone block by block generating monthly lists for proactive engagement

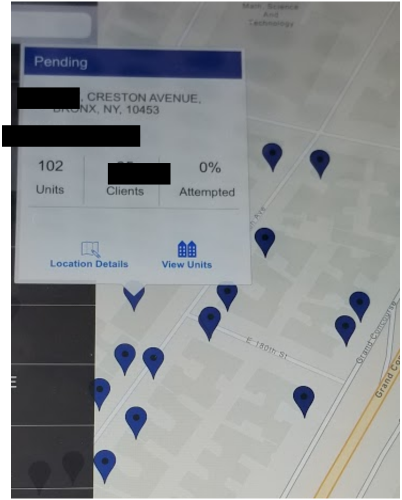
Outreach list: Bushwick Sub-Team
June 2016

Address	*Borough*	Sub-*team*
a5243	Brooklyn	Bushwick
a2110	Brooklyn	Bushwick
⋮		
a0052	Brooklyn	Bushwick

Outreach list: Flushing Sub-Team
June 2016

Address	*Borough*	Sub-*team*
a0031	Queens	Flushing
a1947	Queens	Flushing
⋮		
a6042	Queens	Flushing

Using large-scale data to improve prioritization



k: knocks; o: door opens; c: harassment cases

ID	Date	k	o	c
a1	06-01-2016	18	5	1
a1	06-02-2016	0	NA	NA
a2	06-01-2016	20	7	0
a2	06-02-2016	30	10	2
⋮	⋮	⋮	⋮	⋮
a_n	06-01-2016	10	0	0

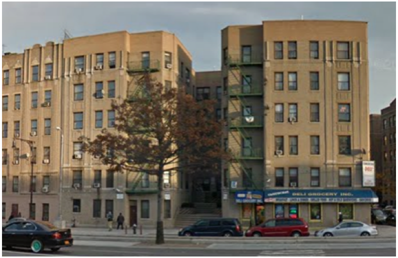
Using large-scale data to improve prioritization

Building A in Bronx:

Total knocks ($\sum_{m=1}^{32} k_{bm}$): 75

Total opens ($\sum_{m=1}^{32} o_{bm}$): 52

Total cases ($\sum_{m=1}^{32} c_{bm}$): 21



Building B in Queens:

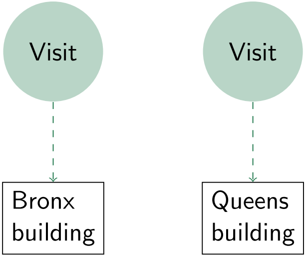
Total knocks: 523

Total opens: 115

Total cases: 0

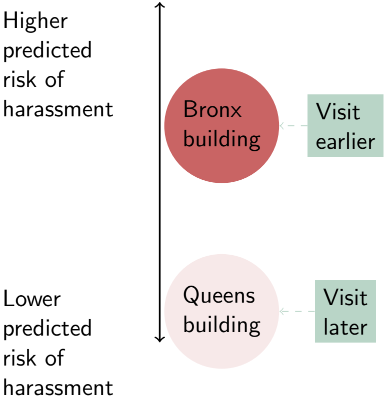
Applying machine learning to that large-scale data to improve prioritization

Current outreach:

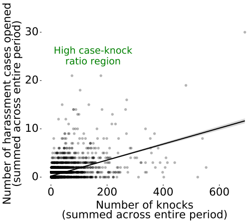


vs.

Machine learning (ML)-guided prioritization:



Steps in ML-guided prioritization: (1) define label



1. Labels to model (for building b in month m):

- Any new case in next month
- New cases/units $>$ threshold

Steps in ML-guided prioritization: (1) define label; (2) choose features

1. Define label

2. Choose features:
Internal (e.g., "which specialists visit?")
Building (e.g., "who is landlord?"(use fuzzy string matching to match
BAINBRIDGE CLASTER AS;
BAINBRIDGE CLUSTER AS;
BAINRIDGE CLUSTER ASS))
Violations (e.g., "how many violations found by code enforcement agency?")
Neighborhoods (ACS tract) (e.g., "what's the demographic composition? When are people home?")

Details on features and pre-processing

Source	Unit of analysis	Example features
Tenant Support Unit	Building	Total cases up to month m; which specialist visits; which zip code
Primary Land Use and Tax Lot (PLUTO)	Building	Landlord (use fuzzy string matching to match BAINBRIDGE CLASTER AS; BAINBRIDGE CLUSTER AS; BAINRIDGE CLUSTER ASS); Building value
HPD, Housing Court, Subsidized Housing (NYC Open data)	Building	Code violations; litigation against landlord
ACS 5-year estimates	Tract	Racial/socioeconomic composition; rent burden; hours work outside home

Total: ~ 400; using 120 for current model; pre-processed using imputation, normalization of continuous features with minimum-maximum scaling, and converting categorical to dummy indicators for levels with $\geq$ buildings

Interfacing with NYC's open data

Goal: interface directly rather than point and click download of a flat file (given daily updates for things like housing code violations): script

```python

## for now, didnt use credentials but later change
client_data = Socrata("data.cityofnewyork.us",
                    creds['nyc_api_getdata']['apikey'],
                    username = creds['nyc_api_getdata']['username'],
                    password = creds['nyc_api_getdata']['password'])
client_metadata = Socrata("data.cityofnewyork.us",
                    creds['nyc_api_getmetadata']['apikey'],
                    username = creds['nyc_api_getmetadata']['username'],
                    password = creds['nyc_api_getmetadata']['password'])


### cleaned version of json ID
jsonid_forget = clean_json(data = opendata_links_valid,
                           dataname = dataname)

### limit parameter - either feeds it
### approximate size if data has unique ID
### or uses the approximate size to get row limit
if int(opendata_links_valid[opendata_links_valid.dataname == dataname]['unique_id']) == 1:
    limit_parameter = generate_limit_parameter(data = opendata_links_valid,
            dataname = dataname, client_metadata = client_metadata)
else:
    limit_parameter = int(opendata_links_valid[opendata_links_valid.dataname == dataname]['size_approx']) + 10000


logger.info("generated lim parameter: " + str(limit_parameter) + " for: " + dataname)


## generate chunks of data to pool and pull from API in parallel
offset_range = list(range(0, limit_parameter, 10000))
print(len(offset_range))
pool = Pool() # Create a multiprocessing Pool; could test with 1 to test sequential version and put time buffer
print('started pool')
df_list = pool.map(getdf_fromapi_onechunk, offset_range)

```

```python
@retry(wait_exponential_multiplier=1000, wait_exponential_max=10000)
def getdf_fromapi_onechunk(offset_start):
    one_get = client_data.get(jsonid_forget, limit = 10000,
                              offset = offset_start) # call
    one_df = pd.DataFrame.from_records(one_get)
    return(one_df)
```

Using SQL for efficiency gains relative to Python

```sql
-- and distkey and sortkey
DROP TABLE IF EXISTS dssg_clean.housinglitig_tomerge;
CREATE TABLE dssg_clean.housinglitig_tomerge
AS

SELECT
            bin_clean
            ,week_start
            ,month_start
            ,count(*) AS housinglitig_count
            ,sum(housinglitig_tenantaction) AS housinglitig_tenantaction_count
            ,sum(housinglitig_heatwater) AS housinglitig_heatwater_count
            ,CASE WHEN count(*) > 0 THEN 1 ELSE 0 END AS housinglitig_any
            ,CASE WHEN sum(housinglitig_tenantaction) > 0 THEN 1 ELSE 0 END AS housinglitig_tenantaction_any
            ,CASE WHEN sum(housinglitig_heatwater) > 0 THEN 1 ELSE 0 END AS housinglitig_heatwater_any
            ,SUM(COUNT(*)) OVER(PARTITION BY bin_clean ORDER BY bin_clean, week_start ASC ROWS UNBOUNDED PRECEDING) AS housingl
FROM
        (SELECT *,
                CASE WHEN casetype LIKE 'Tenant Action%' THEN 1 ELSE 0 END AS housinglitig_tenantaction,
                CASE WHEN casetype LIKE 'Heat%' THEN 1 ELSE 0 END AS housinglitig_heatwater,
                date_trunc('week', date_clean) as week_start,
                date_TRUNC('month', date_clean) AS month_start
         FROM
                -- convert from varchar to date-time format for date-- use the date when DOB query is pushed to open data
                -- convert from integer to varchar for bin
                    (SELECT *,
                        to_date(caseopendate, 'MM--DD-YYYY') as date_clean,
                        bin::varchar(5204) AS bin_clean
                    FROM raw.housingcourt_litigation
                    )  as cleandate
         ) cleandate2

-- aggregate by bin and start of week
GROUP BY bin_clean, week_start, month_start
ORDER BY bin_clean, week_start ASC;
```

Hack-y way to give intuitive variable names for many ACS tract-level features

poverty	B06012_001	poverty status	Total Population In The United States For Whom Poverty Status Is Determined
poverty	B06012_002		Below 100 percent of the poverty level
poverty	B06012_003		100 to 149 percent of the poverty level
poverty	B06012_004		At or above 150 percent of the poverty level
employment	B08007_001	place of work	Total Workers 16 Years And Over

```
## create prefix and suffix columns
df_acs_long['variable_prefix'], df_acs_long['variable_suffix'] = df_acs_long['variable'].str.split('_', 1).str
```

Steps in ML-guided prioritization: (1) define label; (2) choose features; (3) split data temporally and estimate model in training set

1. Define label

2. Choose features:

3. Estimate model in training set:
(e.g., all features/labels from March 2016-October 2016)

Estimating (and then evaluating) $N \sim 800$ models

DT: Decision Tree; RF: Random Forest; GB: Gradient Boosting; LR: Penalized Logistic Regression (Ridge and Lasso)

```
large_grid = {
    'RF':{'n_estimators': [1,10,100,1000,10000], 'max_depth': [1,5,10,20,50,100],
    'max_features': ['sqrt','log2'],'min_samples_split': [2,5,10], 'n_jobs': [-1]},

    'LR': { 'penalty': ['l1','l2'], 'C': [0.00001,0.0001,0.001,0.01,0.1,1,10]},

    'GB': {'n_estimators': [1,10,100,1000,10000],
    'learning_rate' : [0.001,0.01,0.05,0.1,0.5],
    'subsample' : [0.1,0.5,1.0], 'max_depth': [1,3,5,10,20,50,100]},

    'DT': {'criterion': ['gini', 'entropy'], 'max_depth': [1,5,10,20,50,100],
    'min_samples_split': [2,5,10]},
        }
```

Ways we made more automatic

Directory with some snippets

- Used naming conventions for different types of features (e.g., ACS versus HPD violations) so that we could easily create lists that combine different features

```
## iterate over patterns and choose the cols
pattern_list = ['^internal.*', '^internal.*static',
                 '^internal.*count_next.*',
                 '^internal.*any_next.*',
                 '.*cases.*this.*|.*knocks.*this.*|.*opens.*this.*',
                '^hpd.*', '^housinglitig.*',
                '^pluto.*', '^acs_2015.*',
                '^acs_2016.*',
                 '^acs_2015_tract_percent.*',
                 '^acs_2016_tract_percent.*',
                 '^acs_2015_tract_count.*',
                 '^acs_2016_tract_count.*',
                '^subs|^rent.*']

names_list = ['internal_all', 'internal_static',
              'internal_continuous_labels',
              'internal_binary_labels',
              'internal_lagged_labels',
               'hpd', 'housing_litigation',
               'pluto', 'acs_2015_all',
               'acs_2016_all',
              'acs_2015_percent',
              'acs_2016_percent',
              'acs_2015_count',
              'acs_2016_count',
               'misc']
```

Ways we made more automatic

- Created a schema – `results` –that creates a unique identifier for each model run and parses key information

```
def pull_from_eval(alchemy_connection, id_lower = 0, id_upper=900000000, subset_list = False, id_list = '(0)',
                label_topull = 'internal_cases_opened_any_next_month',
                boroughs_fit_topull = 'Bronx_Queens_Staten Island_Brooklyn_Manhattan',
                maxgaptraintest_topull = 40):

    if subset_list == True:

        print('pulling ID list')

        pull_results_info = """
                select * from dssg_results.eval_meta
                where model_id in {id_list}
                and label = '{label_topull}'
                and borough_fit_on = '{boroughs_fit_topull}'
                order by model_id desc;
                """.format(id_list = id_list,
                          label_topull = label_topull,
                          boroughs_fit_topull = boroughs_fit_topull)

                ## pull from database
        df_eval = readquery_todf_postgres(sqlalchemy.text(pull_results_info),
                                         alchemy_connection)

        return(df_eval)

    else:

        print('pulling ids in range')

        pull_results_info = """
                select * from dssg_results.eval_meta
                where model_id >= {id_lower}
                and model_id <= {id_upper}
```

Ways we made more automatic

- Used config file that we drew upon and beginning of each model run (or "experiment") that drew parameters for that run

```yaml

## features
param_name_feature_lists: 'features_dictionary.yaml' # no need to change
param_features_to_pull: 'for_deep_dive' # choose sets of features to use

############ split dates to use
param_split_start_date: '2016-07-01' # pick first split date
param_split_end_date: '2017-12-01' # NOTE THIS IS A DUMMY FOR CONFIG DICTIONARY IN THIS RUN pick end date for te
param_split_increment: '1' # number of months to increment by

############# data to load
param_which_timeunit: 'month' # model isn't set up to run weekly, but we could make the split function dynamic (
param_label_type: 'binary' # choose whether to use binary, continuous, or ratio. Ratio means the building is in
param_primary_label: 'internal_cases_opened_any_next_month' # choose primary label to subset on for missingness
param_label_quantile_threshold: 0.9 # if primary label is a ratio, quantile threshold over which ratio is coded
param_only_observed_labels_train: True # choose whether to include obs with missing labels
param_test_targetzip: True # choose whether test data should include only addresses from TSU target zips

############ storing config file
param_update_config: True # whether to update master config file with parameters; set to false for test runs; Tr
param_return_all_config: False # whether to return the master config file; set to true if want to view for some

############# imputation methods
param_binary_features_imputation_methods: 'impute_median' # method to use to impute binary features: can be mean
param_continuous_features_imputation_methods: 'impute_mean' # method to use to impute continuous features: can b
param_label_imputation_method: 'None'

############# categorical encoding
param_threshold_for_dummy: 10 # how many distinct addresses a level of a categorical var needs to have in order

############# model params
param_unit_id: 'address_id'
param_time_id: 'month_start'
param_model_comment_toadd: 'None'
param_fileid_table_comment_toadd: 'None'
param_borough_fit_list: ['Bronx', 'Queens', 'Staten Island', 'Brooklyn', 'Manhattan'] # can be all 5 or subsets
param_borough_predict_list: ['Bronx', 'Queens', 'Staten Island', 'Brooklyn', 'Manhattan']
param_model_list: 'models_800ish'
```

Ways we made more automatic

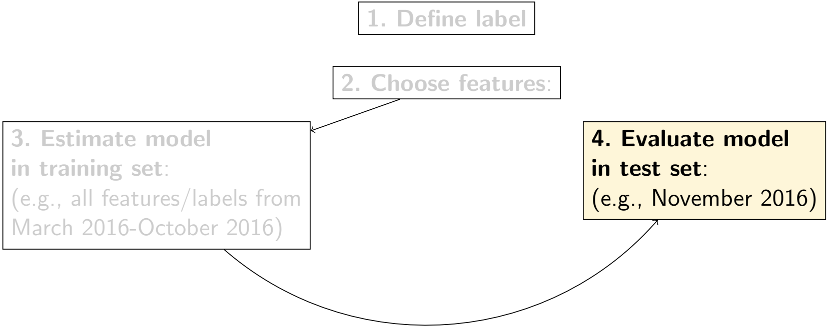
```
random.seed(20190307)
with open('/home/yet/nyc_peu_inspections/pipeline/models/experiments/parameters_0307.yaml','r') as stream:
        parameters = yaml.load(stream)

locals().update(parameters)
parameter_names = [key[0] for key in parameters.items()]
update_master_config_new(creds = creds,
                     parameter_names = parameter_names,
                     parameters = parameters,
                     update_config = param_update_config,
                     return_all_config = param_return_all_config,
                    simpler_loading = True)


log.info('updated master config')


models_dictionary = download_backups_s3('models_dict.yaml', creds,
                                  filetype_string = 'yaml')
model_list = models_dictionary[param_model_list]
## initiate cursor and alchemy connection
```

Steps in ML-guided prioritization: (1) define label; (2) choose features; (3) split data temporally and estimate model in training set, (4) evaluate performance in test set

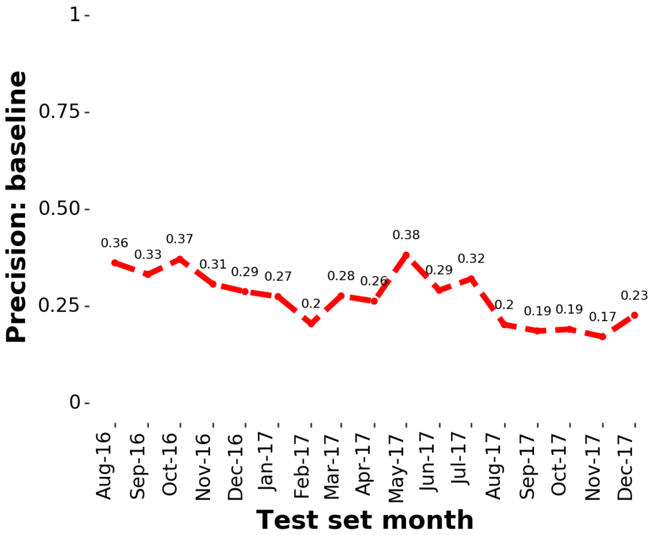


Steps in ML-guided prioritization: (1) define label; (2) choose features, (3) split data temporally and estimate model in training set, (4) evaluate performance in test set

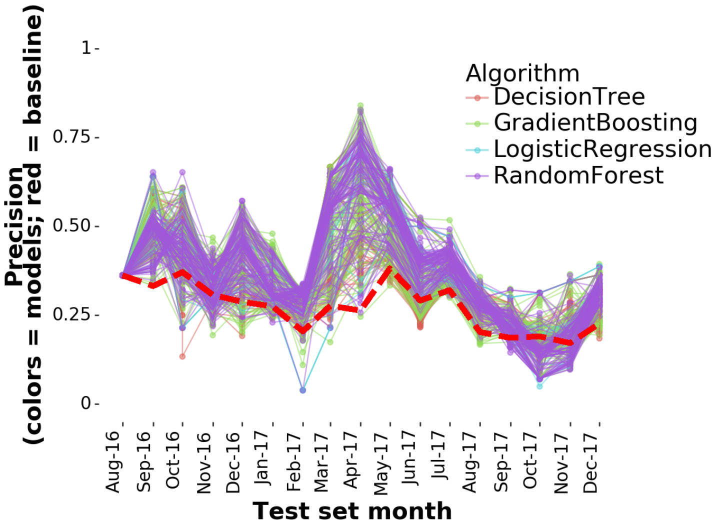
ID	Y **True label**	Score	$\hat{Y}$ **Pred. label**	# of units	# of cases
a5	1	0.8	**1**	153	34
a7	NA	0.79	**1**	23	NA
a8	0	0.65	**1**	77	0
Total units				*253*	
a4	NA	0.46	**0**	100	NA
a1	0	0.45	**0**	10	0

$$\frac{\#\text{ true positive labels below capacity threshold}}{\#\text{ of labels below capacity threshold}} = \frac{1}{2}$$

What we want to outperform: TSU's existing prioritization

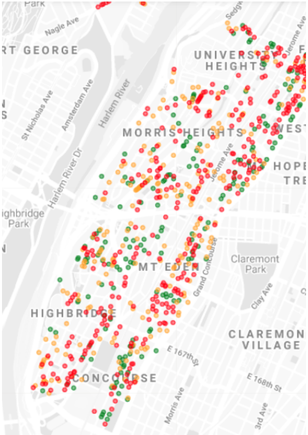


Nearly all models outperform that existing prioritization



Focusing on best-performing model (gradient boosting with 10,000 estimators)

Risk tertiles in South Bronx (any case):

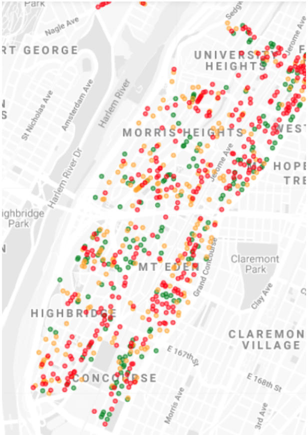


Staying at their same capacity to visit ∼ 60,000 residential units each year, 40-80% increase in performance:

- *Before ML-guided prioritization:* TSU finds ∼ **1800 cases** of landlord harassment

Focusing on best-performing model (gradient boosting with 10,000 estimators)

Risk tertiles in South Bronx (any case):



Staying at their same capacity to visit ∼ 60,000 residential units each year, 40-80% increase in performance:

- *Before ML-guided prioritization:* TSU finds ∼ **1800 cases** of landlord harassment
- *Using ML-guided prioritization:* TSU finds ∼ **2500-3300 cases** of landlord harassment

Does this increase in efficiency come at the expense of equity?

- Critics of social service organizations using machine learning to prioritize argues that increased efficiency may come at the expense of fairness (e.g., O'Neil, 2017; Eubanks, 2018; Bakalar and Zevenbergen, 2017; Bakalar and Zevenbergen, 2017)

Does this increase in efficiency come at the expense of equity?

- Critics of social service organizations using machine learning to prioritize argues that increased efficiency may come at the expense of fairness (e.g., O'Neil, 2017; Eubanks, 2018; Bakalar and Zevenbergen, 2017; Bakalar and Zevenbergen, 2017)
- Suggests that, in addition to examining *aggregate* improvements, should ensure that buildings flagged as highest-risk align with substantive notions of need and vulnerability

Evaluating fairness: ability to capture different forms of vulnerability to harassment

- Mayor De Blasio: "We have teams knocking on doors in fast-changing neighborhoods"

Evaluating fairness: ability to capture different forms of vulnerability to harassment

- Mayor De Blasio: "We have teams knocking on doors in fast-changing neighborhoods"
- Theory (e.g., Sharkey, 2013; Hwang and Sampson, 2014; Desmond, 2016) highlights different potential pathways into high harassment risk:

Buildings in 20 target zip codes with at least 1 rent-stabilized unit

Type of neighborhood: "gentrifying" (poverty + large increase in median income (2000-2010))

Reason for harassment: strong financial incentives to convert units to market rate

Evaluating fairness: ability to capture different forms of vulnerability to harassment

- Mayor De Blasio: "We have teams knocking on doors in fast-changing neighborhoods"
- Theory (e.g., Sharkey, 2013; Hwang and Sampson, 2014; Desmond, 2016) highlights different potential pathways into high harassment risk:

Buildings in 20 target zip codes with at least 1 rent-stabilized unit

Type of neighborhood: "gentrifying" (poverty + large increase in median income (2000-2010))

Reason for harassment: strong financial incentives to convert units to market rate

Evaluating fairness: ability to capture different forms of vulnerability to harassment

- Mayor De Blasio: "We have teams knocking on doors in fast-changing neighborhoods"
- Theory (e.g., Sharkey, 2013; Hwang and Sampson, 2014; Desmond, 2016) highlights different potential pathways into high harassment risk:

Buildings in 20 target zip codes with at least 1 rent-stabilized unit

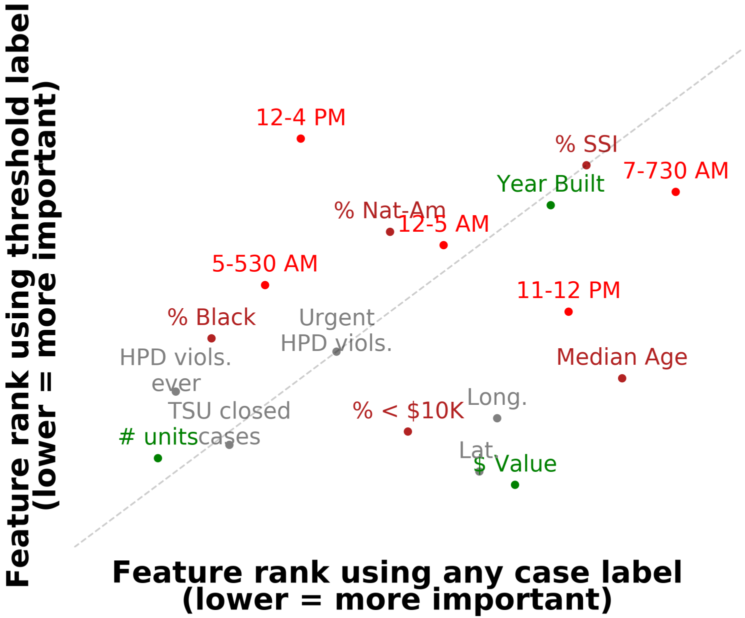
Type of neighborhood: "gentrifying" (poverty + large increase in median income (2000-2010))

Reason for harassment: strong financial incentives to convert units to market rate

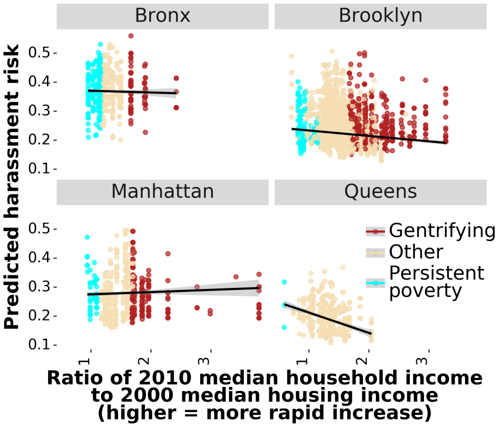
Type of neighborhood: "persistent poverty" (poverty + little increase in median income (2000-2010)

Reason for harassment: persistent landlord-tenant power asymmetries


Feature rank using threshold label
(lower = more important)
Feature rank using any case label
(lower = more important)


12-4 PM
% SSI
7-730 AM
Year Built
% Nat-Am
12-5 AM
5-530 AM
11-12 PM
% Black
Urgent
HPD viols.
HPD viols.
ever
Median Age
Long.
TSU closed
cases
% < $10K
# units
Lat.
$ Value
Feature rank using threshold label
(lower = more important)
Feature rank using any case label
(lower = more important)

High predicted harassment risk in both gentrifying and persistent poverty neighborhoods



Use LTDB (Logan et al., 2012); similar to Ellen and Torrats-Espinosa (2018), use large change in median household income as a measure of gentrification

Discussion and next steps

1. **Policy:** before deploying, field trial to generate exogenous variation in knocks (non-random missingness in building's harassment label, which is only observed in month *m* for buildings they visited and where at least one tenant opened the door)
   - Selective labels problem: Lakkaraju et al. (2017); Casey et al. (2018); Knox, Lowe, Mummolo (2019)
2. **Theory:**
   - More direct comparison to predicted risk if used reactive rights enforcement (e.g., go to high 311 call-volume areas)
   - Landlords
   - Zip code discontinuities and tenant outcomes

Thanks!

http://scholar.princeton.edu/rebeccajohnson/

raj2@princeton.edu

Appendix

Background: policy levers to increase housing affordability

$$\frac{\text{Monthly rent}}{\text{Monthly income}}$$

Background: policy levers to increase housing affordability

$$\frac{\text{Monthly rent}}{\text{Monthly income}}$$

1. Housing vouchers:
Acts on: entire ratio;
Attaches to: individuals

Background: policy levers to increase housing affordability

$$\frac{\text{Monthly rent}}{\text{Monthly income}}$$

1. Housing vouchers:
Acts on: entire ratio;
Attaches to: individuals

2. Rent control:
Acts on: numerator;
Attaches to: individuals + a housing unit

Background: policy levers to increase housing affordability

$$\frac{\text{Monthly rent}}{\text{Monthly income}}$$

1. Housing vouchers:
Acts on: entire ratio;
Attaches to: individuals

2. Rent control:
Acts on: numerator;
Attaches to: individuals + a housing unit

3. Rent stabilization:
Acts on: numerator;
Attaches to: a housing unit

TSU's current method for prioritizing which buildings to visit: expert judgment

Target universe: buildings in
20 target zip codes with
at least 1 rent-stabilized unit

TSU's current method for prioritizing which buildings to visit: expert judgment

Target universe: buildings in 20 target zip codes with at least 1 rent-stabilized unit

↓

TSU team lead puts building b on knock list in month m

Labels: details

Throughout: since TSU ranks buildings at the beginning of each month, risk of a *new case* of landlord harassment in the upcoming month

Any case label - any case in the next month:

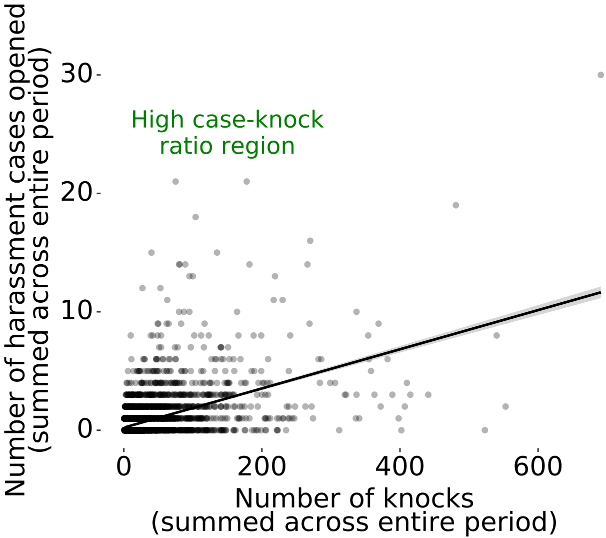
$$y_{bm} = \begin{cases} 0 & \textit{if } k_{bm} \geq 1, o_{bm} \geq 1, \\ & c_{bm} = 0 \\ 1 & \textit{if } k_{bm} \geq 1, o_{bm} \geq 1, \\ & c_{bm} \geq 1 \\ NA & \text{otherwise} \end{cases}$$

Threshold label- case/units $>$ ratio in next month:

τ = percentile threshold;
i_b = # of units at building b

$$y_{bm} = \begin{cases} 0 & \textit{if } k_{bm} \geq 1, o_{bm} \geq 1, \\ & \frac{c_{bm}}{i_b} < \tau \\ 1 & \textit{if } k_{bm} \geq 1, o_{bm} \geq 1, \\ & \frac{c_{bm}}{i_b} \geq \tau \\ NA & \text{otherwise} \end{cases}$$

Labels: details



Features: details

Source	Unit of analysis	Example features
Tenant Support Unit	Building	Total cases up to month *m*; which specialist visits; which zip code
Primary Land Use and Tax Lot (PLUTO)	Building	Landlord (use fuzzy string matching to match BAINBRIDGE CLASTER AS; BAINBRIDGE CLUSTER AS; BAINRIDGE CLUSTER ASS); Building value
HPD, Housing Court, Subsidized Housing (NYC Open data)	Building	Code violations; litigation against landlord
ACS 5-year estimates	Tract	Racial/socioeconomic composition; rent burden; hours work outside home

Total: $\sim$ 400; using 120 for current model; pre-processed using imputation, normalization of continuous features with minimum-maximum scaling, and converting categorical to dummy indicators for levels with $\geq$ buildings

Details on temporal split

ID	Month	Y	HPD viols. (ever)	Tract % Black	...
a1	06-2016	1	0	50	
a5	06-2016	0	20	70	
a8	06-2016	0	5	5	
⋮					
a8	10-2016	0	8	5	

Details on temporal split

ID	Month	Y	HPD viols. (ever)	Tract % Black	...
a1	06-2016	1	0	50	
a5	06-2016	0	20	70	
a8	06-2016	0	5	5	
⋮					
a8	10-2016	0	8	5	
a1	11-2016	NA	5	50	
a2	11-2016	NA	54	70	
a3	11-2016	1	2	15	
⋮					

Step three for learning harassment risk: use machine learning to learn risk as a flexible function of those features

1. For each split, train model j on data from month $= 1 \ldots$ m

Step three for learning harassment risk: use machine learning to learn risk as a flexible function of those features

1. For each split, train model j on data from month $= 1 \ldots$ m
2. Generate predictions for model j on data from month $m + 1$

Step three for learning harassment risk: use machine learning to learn risk as a flexible function of those features

1. For each split, train model j on data from month $= 1 \ldots$ m
2. Generate predictions for model j on data from month $m + 1$
3. Evaluate test set predictions:

Step three for learning harassment risk: use machine learning to learn risk as a flexible function of those features

1. For each split, train model j on data from month $= 1 \ldots$ m
2. Generate predictions for model j on data from month $m + 1$
3. Evaluate test set predictions:
   - Rank all test set buildings by $\hat{y}$

Step three for learning harassment risk: use machine learning to learn risk as a flexible function of those features

1. For each split, train model j on data from month $= 1 \ldots$ m
2. Generate predictions for model j on data from month $m + 1$
3. Evaluate test set predictions:
   - Rank all test set buildings by $\hat{y}$
   - Draw capacity threshold τ at half of TSU's observed outreach capacity

Address ID	Score	# of Units	Pred. Label	True Label	# of Cases
a5	0.81	153	1	1	34
a7	0.68	23	1		
a8	0.62	77	1	0	12
Total units				*253*	
a4	0.48	300	0		
a1	0.46	100	0	1	23

Step three for learning harassment risk: use machine learning to learn risk as a flexible function of those features

1. For each split, train model j on data from month $= 1 \ldots$ m
2. Generate predictions for model j on data from month $m + 1$
3. Evaluate test set predictions:
   - Rank all test set buildings by $\hat{y}$
   - Draw capacity threshold τ at half of TSU's observed outreach capacity

Address ID	Score	# of Units	Pred. Label	True Label	# of Cases
a5	0.81	153	1	1	34
a7	0.68	23	1		
a8	0.62	77	1	0	12
Total units				*253*	
a4	0.48	300	0		
a1	0.46	100	0	1	23

   - Using buildings with observed labels, calculate metric (main: precision at τ):

$$\frac{\#\text{ true positive labels below } \tau}{\#\text{ of labels below } \tau} = \frac{1}{2}$$

Step three for learning harassment risk: use machine learning to learn risk as a flexible function of those features

1. For each split, train model j on data from month $= 1 \ldots$ m
2. Generate predictions for model j on data from month $m+1$
3. Evaluate test set predictions:
   - Rank all test set buildings by $\hat{y}$
   - Draw capacity threshold τ at half of TSU's observed outreach capacity

Address ID	Score	# of Units	Pred. Label	True Label	# of Cases
a5	0.81	153	1	1	34
a7	0.68	23	1		
a8	0.62	77	1	0	12
Total units				*253*	
a4	0.48	300	0		
a1	0.46	100	0	1	23

   - Using buildings with observed labels, calculate metric (main: precision at τ):

$$\frac{\#\text{ true positive labels below }\tau}{\#\text{ of labels below }\tau} = \frac{1}{2}$$

4. Repeat for model $j+1$

Step four in learning harassment risk: evaluate performance in the held-out test set

- Rank all test set buildings by $\hat{y}$

Step four in learning harassment risk: evaluate performance in the held-out test set

- Rank all test set buildings by $\hat{y}$
- Draw capacity threshold at half of TSU's observed outreach capacity

Address ID	Score	# of Units	Pred. Label	True Label	# of Cases
a5	0.81	153	1	1	34
a7	0.68	23	1		
a8	0.62	77	1	0	12
Total units		*253*			
a4	0.48	300	0		
a1	0.46	100	0	1	23

Step four in learning harassment risk: evaluate performance in the held-out test set

- Rank all test set buildings by $\hat{y}$
- Draw capacity threshold at half of TSU's observed outreach capacity

Address ID	Score	# of Units	Pred. Label	True Label	# of Cases
a5	0.81	153	1	1	34
a7	0.68	23	1		
a8	0.62	77	1	0	12
Total units		*253*			
a4	0.48	300	0		
a1	0.46	100	0	1	23

- Using buildings with observed labels, calculate metric (main: precision at capacity threshold):

$$\frac{\#\text{ true positive labels below capacity threshold}}{\#\text{ of labels below capacity threshold}} = \frac{1}{2}$$

Compare $N \sim 800$ models to that expert judgment

DT: Decision Tree; RF: Random Forest; GB: Gradient Boosting; LR: Penalized Logistic Regression (Ridge and Lasso)

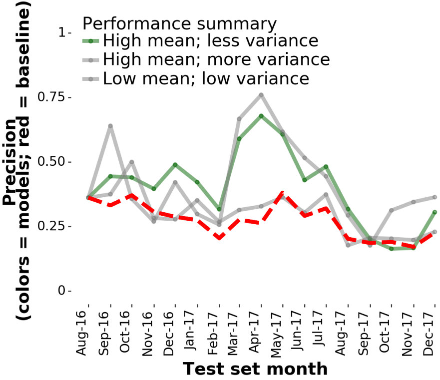
```
large_grid = {
    'RF':{'n_estimators': [1,10,100,1000,10000], 'max_depth': [1,5,10,20,50,100],
    'max_features': ['sqrt','log2'],'min_samples_split': [2,5,10], 'n_jobs': [-1]},

    'LR': { 'penalty': ['l1','l2'], 'C': [0.00001,0.0001,0.001,0.01,0.1,1,10]},

    'GB': {'n_estimators': [1,10,100,1000,10000],
    'learning_rate' : [0.001,0.01,0.05,0.1,0.5],
    'subsample' : [0.1,0.5,1.0], 'max_depth': [1,3,5,10,20,50,100]},

    'DT': {'criterion': ['gini', 'entropy'], 'max_depth': [1,5,10,20,50,100],
    'min_samples_split': [2,5,10]},
        }
```

Weighting performance across different test set months



Result: gradient boosting with 10,000 estimators; learning rate of 0.001; split criterion is Friedman mean squared error; average performance ratio of 1.54 means TSU can visit the same number of buildings and find 54 more buildings with any case

Machine learning (ML) and demography/social science

Athey (2017), Molina and Garip (2019), and others discuss what social science contributes to ML and what ML contributes to social science

1. **Predictions**: fairness and variation in predicted risk under different label definitions

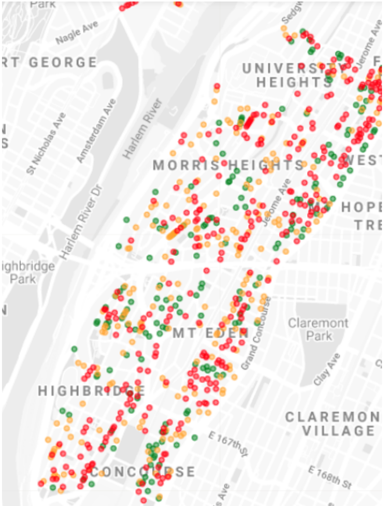
Machine learning (ML) and demography/social science

Athey (2017), Molina and Garip (2019), and others discuss what social science contributes to ML and what ML contributes to social science

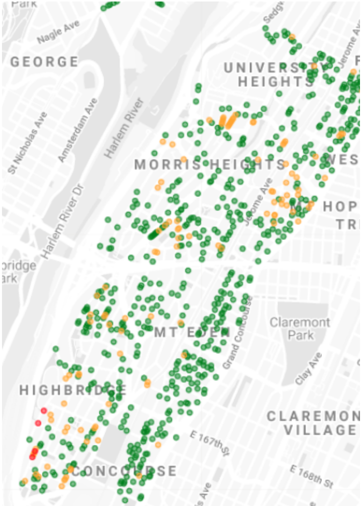
1. **Predictions**: fairness and variation in predicted risk under different label definitions

Risk tertile: area of Bronx under different label definitions (same gradient boosting model + same hyperparameters)

Any case:

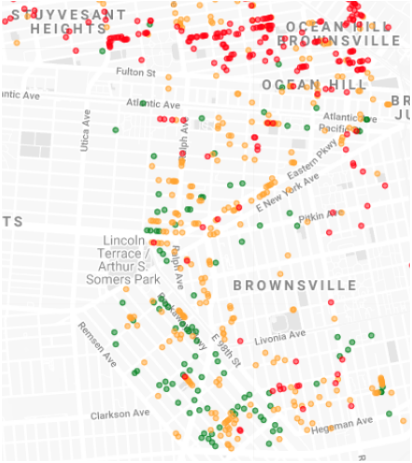


Case per units > threshold:

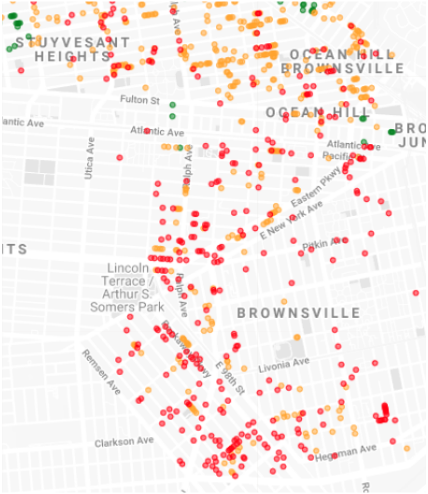


Risk tertile: area of Brooklyn under each label

Any case:



Case per unit $>$ threshold:



No single correct label definition; social science research on landlord-tenant dynamics may lend insight

- *Reasons to use any case label*:

No single correct label definition; social science research on landlord-tenant dynamics may lend insight

- *Reasons to use any case label*:
  - Landlord typically use tactics against all tenants in rent-stabilized units in a building

No single correct label definition; social science research on landlord-tenant dynamics may lend insight

- *Reasons to use any case label*:
  - Landlord typically use tactics against all tenants in rent-stabilized units in a building
  - Fear of reprisal means only one speaks out

No single correct label definition; social science research on landlord-tenant dynamics may lend insight

- *Reasons to use any case label*:
  - Landlord typically use tactics against all tenants in rent-stabilized units in a building
  - Fear of reprisal means only one speaks out
- *Reasons to use case /units label:*

No single correct label definition; social science research on landlord-tenant dynamics may lend insight

- *Reasons to use any case label*:
  - Landlord typically use tactics against all tenants in rent-stabilized units in a building
  - Fear of reprisal means only one speaks out
- *Reasons to use case /units label:*
  - Landlords typically use tactics against some proportion of more vulnerable tenants in rent-stabilized units in a building

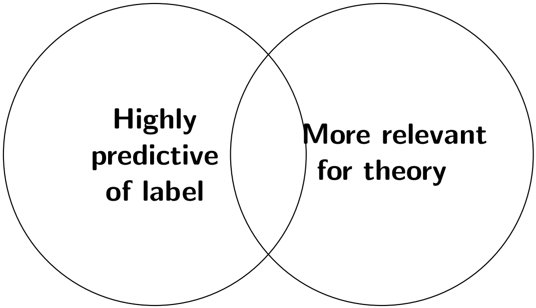
No single correct label definition; social science research on landlord-tenant dynamics may lend insight

- *Reasons to use any case label*:
  - Landlord typically use tactics against all tenants in rent-stabilized units in a building
  - Fear of reprisal means only one speaks out
- *Reasons to use case /units label:*
  - Landlords typically use tactics against some proportion of more vulnerable tenants in rent-stabilized units in a building
  - Variation in that proportion is relevant

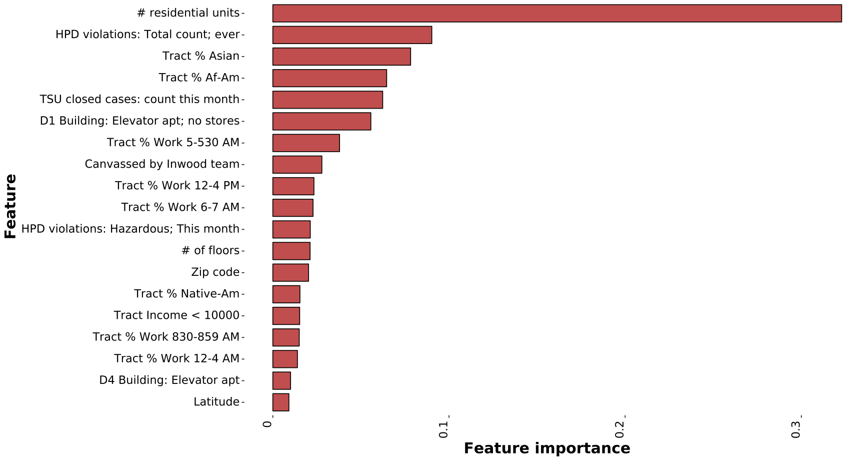
Machine learning (ML) and demography/social science

Athey (2017), Molina and Garip (2019), and others discuss what social science contributes to ML and what ML contributes to social science

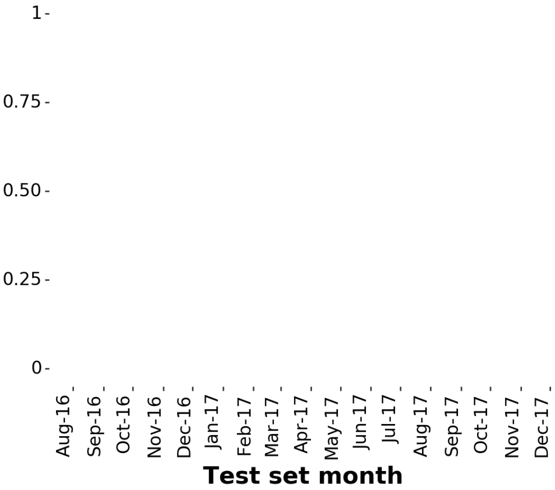
1. **Predictions**: fairness and variation in predicted risk under different label definitions
2. **Feature interpretation:** Goldilocks region:



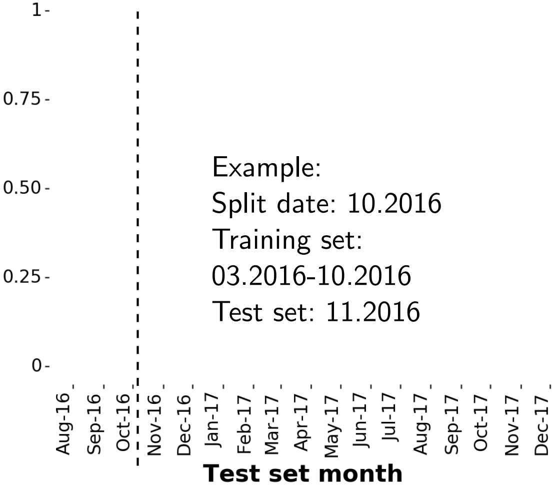
Largest feature importances: any case label



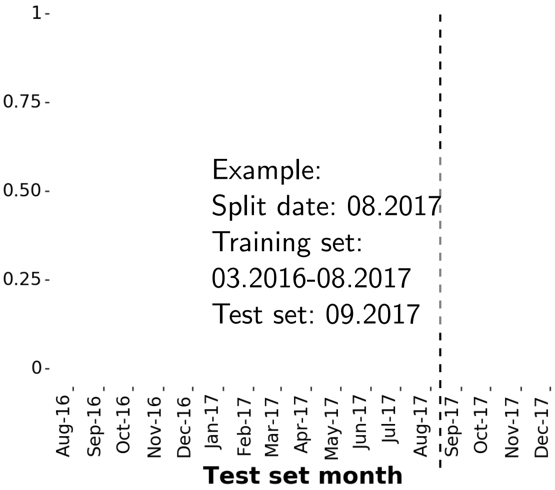
Step three in learning harassment risk: use training set to estimate risk as flexible function of features



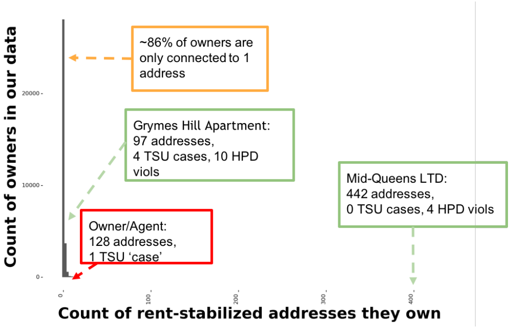
Step three in learning harassment risk: use training set to estimate risk as flexible function of features



Step three in learning harassment risk: use training set to estimate risk as flexible function of features



Features: relevant for theory but not highly predictive of label



Despite fuzzy string matching to map multiple spellings to same owner, e.g.:
BAINBRIDGE CLASTER AS
BAINBRIDGE CLUSTER AS
BAINRIDGE CLUSTER ASS

Features: relevant for theory and highly predictive of label

- Mayor De Blasio: "We have teams knocking on doors in **fast-changing neighborhoods**"

Features: relevant for theory and highly predictive of label

- Mayor De Blasio: "We have teams knocking on doors in **fast-changing neighborhoods**"
- **Theory:** narratives about gentrification mask differences between multiple types of disadvantaged neighborhoods (Hwang and Sampson, 2014): ones that remain racially hypersegregated (Sharkey, 2013) and ones that are disadvantaged but more racially diverse

Features: relevant for theory and highly predictive of label

- Mayor De Blasio: "We have teams knocking on doors in **fast-changing neighborhoods**"
- **Theory:** narratives about gentrification mask differences between multiple types of disadvantaged neighborhoods (Hwang and Sampson, 2014): ones that remain racially hypersegregated (Sharkey, 2013) and ones that are disadvantaged but more racially diverse
- In turn, we might see same outcome–landlord harassment–in the two types of neighborhoods but for different reasons:

Features: relevant for theory and highly predictive of label

- Mayor De Blasio: "We have teams knocking on doors in **fast-changing neighborhoods**"
- **Theory:** narratives about gentrification mask differences between multiple types of disadvantaged neighborhoods (Hwang and Sampson, 2014): ones that remain racially hypersegregated (Sharkey, 2013) and ones that are disadvantaged but more racially diverse
- In turn, we might see same outcome–landlord harassment–in the two types of neighborhoods but for different reasons:
  - "Gentrifying" neighborhoods: stronger incentives to convert units to market-rate ones

Features: relevant for theory and highly predictive of label

- Mayor De Blasio: "We have teams knocking on doors in **fast-changing neighborhoods**"
- **Theory:** narratives about gentrification mask differences between multiple types of disadvantaged neighborhoods (Hwang and Sampson, 2014): ones that remain racially hypersegregated (Sharkey, 2013) and ones that are disadvantaged but more racially diverse
- In turn, we might see same outcome–landlord harassment–in the two types of neighborhoods but for different reasons:
  - "Gentrifying" neighborhoods: stronger incentives to convert units to market-rate ones
  - "Stuck in place" neighborhoods: landlords take actions like neglect serious repairs or cut off heat less to try to get tenants to move out and more due to power asymmetries/as a way to extract unpaid rent (Desmond, 2016)

How should the Tenant Support Unit prioritize visits among $\sim$ 6500 buildings containing $\sim$ 142,000 residential units?

How should the Tenant Support Unit prioritize visits among $\sim$ 6500 buildings containing $\sim$ 142,000 residential units?

$E[\text{Y} = \text{new case of illegal landlord harassment at building b in month m}|?]$

How should the Tenant Support Unit prioritize visits among $\sim$ 6500 buildings containing $\sim$ 142,000 residential units?

$E[\mathrm{Y} = \text{new case of illegal landlord harassment at building b in month m}|?]$