1. (10 points) Did they talk about the exploratory analysis they did? If not indicated explicitly, what exploratory analysis do you think will be great to see?

The researchers gathered data from multiple sources, so as allow them to get a better description of the buildings, knock attempts & tenants, and landlords history.

"To explore variables that can help us predict which buildings maybe at risk of harassment, we combined data from multiple sources."

Although they did not specifically mention any EDA technique, gathering the proper data is extremely important. One suggestion would be to check out the correlations between different features, their distribution, the demographics/income profiles of the zip codes, and perhaps the effect of time on zip codes (is there any change for bad, worse, or same).

2. (10 points) What preprocessing did the authors perform?

Feature Extraction/Generation, the authors combined information from the additional datasets (building info, landlord info,…) into the current dataset and ensured a congruent format with the records that they generated and the recorder that they already had (92 total features).

- Building level
- Tract level

Feature pre-preprocessing, one-hot encoding re-format of previously created features.

Operations,

- Clean data (i.e., preprocessed such as removing duplicate records)
- Generated the features and matched data from different sources by location indicators.
- Extrapolation to impute missing data in the features (not the label)
    - such as imputing missing records in 2018.2 with data from 2018.1.
- Normalize continuous features (Min-Max scaler)

3. (10 points) Did they do this before or after dividing that data into training/test sets?

Although not explicitly mentioned, it seems (and makes sense) as if the data has been pre-processed & assembled BEFORE any train/test split.

4. (15 points) How did they mitigate overfitting/underfitting? If it is not indicated explicitly, what do you think they can do (such as what hyperparameter can be tuned) to mitigate overfitting/underfitting?

At the moment, there seem to be two pressing issues in terms of bias:

1. There is a strong correlation between a building being larger and a building being identified as higher risk: buildings with larger numbers of units were more likely to be predicted as buildings of high rental harassment risk.
    a. solution: standardize a building's count of cases by the number of tenants who might have a case
2. Selection bias in the labels. Since there are only labels from buildings canvassed by TSU, and there is some bias in how they select buildings to canvass, the model is trained only on that data and will most likely be only confident on predictions made on similar buildings.
    a. Solution: record new cases based on alternative methods (probably not feasible).

Model tuning/hyper parameterization to reduce possible under/over fitting

1. Random Forest (RF)
    a. n_esetimators, this represents the number of trees in our forest. The higher the number of trees the better the performance of our model.Hence, **increase** n_estimators parameter (from 100 to say [200,300]) if the given processor can handle it
    b. max_features, this represents the size of the random subsets of features to consider when splitting a node. Ideally, since only a subset of the features truly matters, we want to set max_features smaller than the numbers of predictors in the model. Also, in a case where we are overfitting, we might want to **decrease** the number of features
    c. max_depth, controls maximum depth (number of split points), and it is a common way to reduce tree complexity and overfitting. In our case, if we are overfitting (finding relationships when none exist), we want to prune the tree (i.e. be less specific)
2. Logistic Regression (LR)
    a. Solver, this algorithm is used to optimize the problem.
    b. Penalty, this is used to specify the norm used in the penalization.
    c. C parameter, Inverse of regularization strength. Smaller values specify stronger regularization. It is a penalty term, meant to disincentivize and regulate against overfitting.
3. Decision Trees(DT)
    a. Criterion, this measures the quality of a split. Supported criteria are "gini" for the Gini impurity and "entropy" for the information gain.
    b. Splitter , this is used to choose the split at each node. Supported strategies are "best" to choose the best split and "random" to choose the best random split.
    c. max_depth, (as in RF)
    d. class_weight is used to provide a weight or bias for each output class
    e. *min_samples_leaf,* this is the minimum number of samples required to be at a leaf node.
4. Gradient Boosting (GB)
    a. learning_rate, this determines the contribution of each tree on the final outcome and controls how quickly the algorithm proceeds down the gradient descent (learns
    b. ccp_alpha, this is the complexity parameter used for pruning.
    c. n_estimator, The number of boosting stages to perform. Gradient boosting is fairly robust to over-fitting so a large number usually results in better performance.

d. min_samples_split, controls the complexity of each tree. Since we tend to use shorter trees this rarely has a large impact on performance.

5. (15 points) Did they use cross validation? How? If not, why do you think they did not? And how did they do validation?

As mentioned in section 5.2 (Splitting Data into Training and Testing Sets), the authors DO PERFORM *temporal cross-validation.* More specifically, the authors say that in order to evaluate models with temporal cross-validation, they followed the rule of time-dependent knowledge restriction to temporally split the data into training and testing sets.

Their procedure can be explained through an example. Suppose in one data split, if we wanted to use data until end of March 2017 (i.e., testing features) to predict the risk of harassment during April 2017 (i.e., testing label), the training set should contain features only until end of February 2017. The training label would then be generated using cases from records during March 2017.

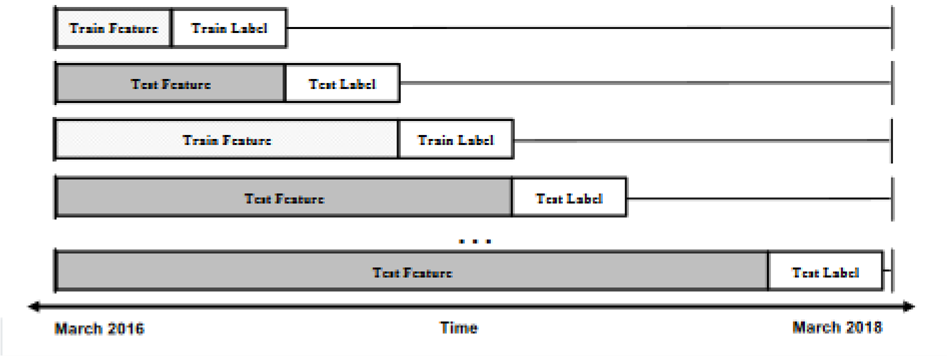

Figure 2: An example illustrating training and testing splits.

The authors, split the training data into 17 folds as illustrated above to conduct temporal validation (i.e. validation).

6. (15 points) What did they do to prevent data leakage? Was there anything more that they could have done?

In the cross-validation scheme, the authors had to consider time given the time-dependent knowledge restriction. As a result, they needed to ensure that the knowledge in the future (i.e., the testing set) did not inform predictions in the past (i.e., the training set). By doing so, they have avoided contaminating training/testing set & the stemming results. Also, the final models were trained on data from July 2016 to December 2017 and were tested on out-reach records from January 2018.

At a first glance, it appears as if the authors have done a good job at not commit any faulty behavior that would cause serious problems such as data leakage. Perhaps, it mentions that they divided the training data into 17 folds; one additional step could be to perform a second train/test on each of these 17 folds. This will ensure that the parameters that are picked are indeed stable throughout.

7. (15 points) What evaluation Metrics did they report? What do you think went into this decision?

Precision = (# of true positive labels in top k)/ (# of total labels in top)

Precision (at the top k) is the proportion of buildings that are labeled as positive (i.e., resulted in true cases) in the top k building list. It measures the efficiency of the model.

Recall = (# of true positive labels in top k)/ (# of true positive labels in test set)

Recall (at the top k) represents the proportion of buildings with true positive labels (i.e., with cases identified) that the model captures in the top k list. It measures model coverage.

K-choice: this was pre-determined by TSU (half of possible canvassing's capacity).

Top-k: Rankings for residential buildings by predicted risk score.

Model-performance: top-k list of buildings based ONLY on labeled buildings data which were ranked by the predicted risk of harassment.

Why these metrics?

The reason for picking such metrics resides in their *flexibility* in a scenario where there are missing labels in that data (e.g. building has not been canvassed, so there is no report). The goal of the project is indeed to push inspectors into places (new perhaps) that there is a need. Hence, being able to use metrics that can overcome the difficulty of missing data is crucial.

8. (10 points) How will they use or verify the results in the real world?

As the authors mention, their results can be measured in the real world in two possible ways

1. They recommend conducting a field trial with proactive canvassing on the previously not canvassed buildings to further validate the model on both labeled and unlabeled data.

2. Prior to the results being used to inform TSU's process, the agency should conduct a field trial to validate the predictions of the developed model as well as run a thorough bias and fairness analysis. This field trial can better inform whether buildings that the model flags as high risk are more likely to yield cases than buildings that the model flags as low risk.