New York University

DS-GA 1001

Introduction to Data Science Term Project

Predicting NYC Housing Violations

Authors: Kristen Kwan, Rafael Garcia Cano Da Costa, Evelina Bakhturina, Ben Jakubowski

Instructor: Professor Brian D'Alessandro

December 21, 2015

Index:

I. Business Understanding

The New York City Department of Housing Preservation and Development (HPD) ensures that building owners comply with the City's Housing Maintenance Code (HMC) and the New York State Multiple Dwelling Law. These regulations provide the minimum housing standards for residential buildings in New York City.

If a tenant believes their building or apartment does not meet these standards, they may file a housing complaint with HPD through NYC's 311 Citizen Service Center. When filing a complaint, the tenant can cite one or more specific problems in their apartment or building. After the housing complaint is filed, a housing inspector attempts to contact the building's managing agent to advise them a complaint has been filed and violations may be issued if the problems are not corrected. The housing inspector then attempts to call the complaining tenant to see if the problems were corrected. If the problems were not corrected or HPD cannot reach the tenant, an inspector is sent to inspect the reported problems and determine whether any violations should be issued (NYC HPD 2015). For a complete outline of the housing enforcement process, see Appendix A: Housing Complaint and Inspection Flowchart.

Housing inspections are labor intensive. During the 2015 fiscal year, each inspection team completed an average of 12.2 inspections per day, for a total of 664,960 inspections during the fiscal year. Importantly, inspections do not always produce violations, and the number of complaints are increasing each year while the number of inspectors is stagnant. In 2015, HPD issued 408,874 violations in response to 553,135 complaints (NYC Mayor's Office of Operations, 2015). In fact, with most complaints citing multiple housing problems, the majority of tenant-identified problems do not result in a violation being issued.

A tool that can predict whether a housing complaint problem will result in a violation would aid HPD in more effectively allocating human resources and improve more tenants' quality of life and living conditions. Specifically, we envision the following use scenario: A tenant calls the NYC's 311 Citizen Service Center and registers a complaint problem. A deployed binary classification model then predicts whether the complaint problem will result in a violation. This class estimate is then used to calculate the expected number of violations in a given inspector's caseload. Finally, HPD management uses this value to rebalance caseloads and monitor inspector performance.

This model deployment is expected to add value to HPD's housing maintenance code inspection services since it will ensure limited human resources are used at maximal efficiency. We anticipate that an inspector's performance declines when overloaded with too many violation-producing cases; similarly, an inspector's productivity is (by definition) low when his/her caseload results in few violations. We propose inspector productivity and performance will be maximized when inspectors are given balanced caseloads, and our model can assist HPD management in achieving this management objective.

II. Data Understanding

Our model is built on five datasets. Our primary dataset is NYC HPD's Housing Complaint Problems dataset, which includes all information collected by HPD about tenant complaint problems. Each instance in the dataset consists of a unique problem ID and its associated complaint ID; if several instances have the same complaint ID, one tenant filed a complaint consisting of multiple problems. In addition, each instance in this dataset contains informative attributes about the problem, including the type of problem, the location of the problem in the building or apartment, and the severity of the problem. Finally, each complaint problem is associated with a text feature, 'Status Description', that describes HPD's response to the complaint. This description is parsed to construct our target variable.

The other four datasets in this analysis are used to expand the feature space. The types of features obtained from each supporting dataset are summarized below, and feature engineering details of each dataset are discussed in detail in the subsequent section.

Dataset	Features
Housing Maintenance Code Complaints	This dataset links ComplaintIDs to BBLs (a unique tax lot identifier), which is used as a key in many subsequent merges.
House Maintenance Code Violations (HMCV)	This dataset provides data on the housing maintenance code violations previously issued to a building.
NYC Pluto	This dataset provides data on the age, size, and assessed tax value of residential buildings.
American Community Survey (ACS)	Census-tract level median annual income was obtained from the American Community Survey's five year summary dataset to capture any potential systematic differences in HPD response to housing complaints based on socioeconomic characteristics.

III. Data Preparation

1. **Housing Complaint Problems**:

    a. **Inclusion Criteria**: Only closed complaint problems were included in our dataset (since open complaint problems are considered unlabeled). Additionally, only records with feature values defined as valid by the HPD published data dictionary (HPD 2015) were included in our analytic dataset.

    b. **Feature Engineering:**

        i. **Features**: Categorical ID variables were converted to binary dummy variables. In addition, to avoid overfitting, we dropped classes with fewer than 500 observations (corresponding to 0.08% of the 609,122 total records). Note this

threshold was chosen heuristically. Future improvement to the model may be obtained by learning an optimal threshold.

ii. **Target Variable**: Crucially, the binary target variable was obtained by parsing the 'Status Description' reported in this dataset. Specifically, exploratory analysis revealed a set of substrings that partition status descriptions (see Appendix B: Parsing Status Descriptions). Two of these substrings indicated a violation was issued. The other substrings indicated either (a) no violation was issued, (b) the record was effectively unlabeled, or (c) the record was effectively a duplicate.

2. **Housing Maintenance Code Complaints**:

   a. **Inclusion Criteria**: Records were excluded if they had missing data or misentered Borough code. In addition, only housing maintenance code complaints reported between November 1st, 2014, and October 31st, 2015 were included (see Figure 1).

   b. **Feature Engineering:** BoroID, Block, and Lot numbers were concatenated to form BBL numbers, which are unique tax lot identifiers used as a proxy for a unique building identifier in our analysis (note it is a proxy since condominiums have unique BBLs; as such, it is possible for a single physical building to contain multiple BBLs).

3. **Housing Maintenance Code Violations (HMCV):**

   a. **Inclusion Criteria**: Records were excluded if they had missing data, or if the recorded BoroID or Class values were misentered. Additionally, only records with violation approval dates between November 1st, 2009, and October 31st, 2014 were included (see Figure 1).

b. **Feature Engineering:** BoroID, Block, and Lot numbers were concatenated to form BBL numbers. Then data were grouped by BBL and aggregated by counting the total number of Class A, Class B, and Class C violations issued to a BBL during the five-year period.

4. **NYC Pluto**:

   a. **Inclusion Criteria**: Records with missing or invalid values (i.e. a Year Built recorded as calendar year "0") were dropped.

   b. **Feature Engineering**: First, features describing building dates (year built, year modified 1, year modified 2) were converted from calendar years to relative ages (i.e. 15 years old, instead of 2000). Then average assessed value per unit was calculated by dividing total assessed tax value by total number of residential units in a BBL.

5. **American Community Survey**: This dataset included only one feature - median income - and it did not require any additional feature engineering.

11/1/2009 - 10/31/2010	11/1/2010 - 10/31/2011	11/1/2011 - 10/31/2012	11/1/2012 - 10/31/2013	11/1/2013 - 10/31/2014	11/1/2014 - 10/31/2015
Period used to determine total number of class A, B, and C violations issued to BBL					Complaints included in analytic set

Figure 1: Note the date range of complaint problems included in our analytic dataset was disjoint from the time period used to calculate total numbers of class A, B, and C violations issued to a BBL. This is important since it avoids the leakage that would occur if class A, B, and C violation totals included violations resulting from our training and test set complaint problems.

Following data processing, the individual datasets were merged as shown in Figure 2.

Merging Cleaned Datasets to Create the Analytic Dataset

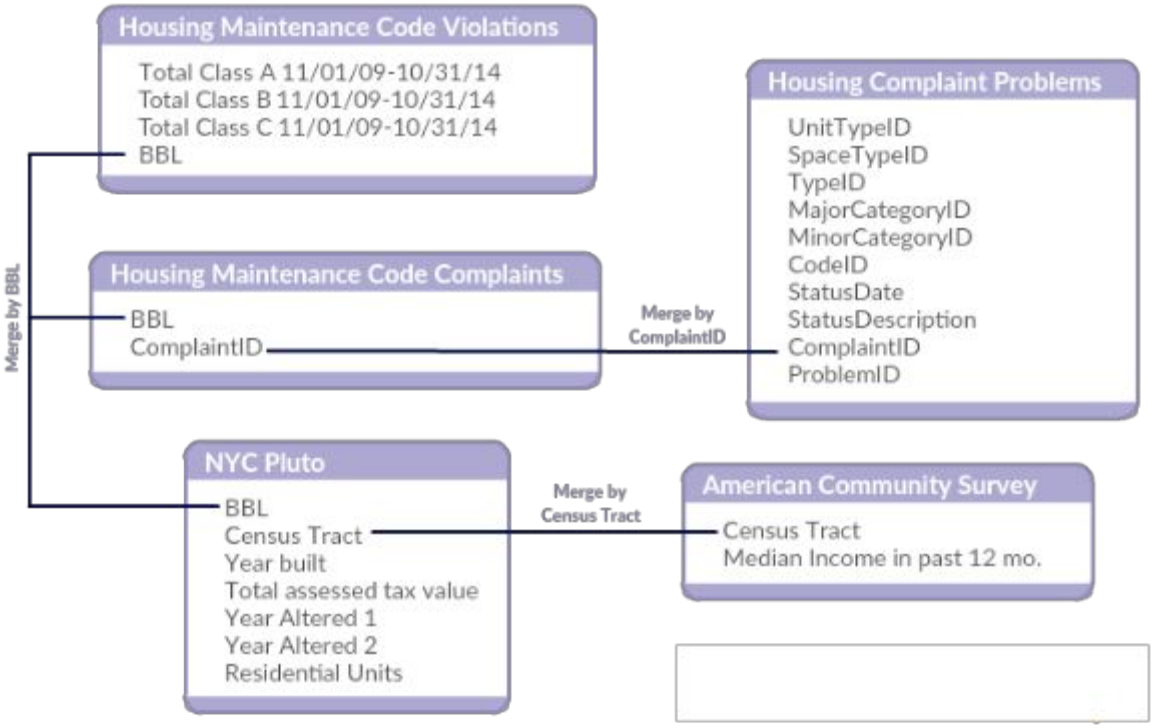


Figure 2: Five processed datasets were merged to construct our analytic dataset. Note the features used as merge keys are identified by connecting lines.

IV. Modeling

Once the data was processed and merged to produce the final analytic dataset, 50% training, 25% validation, and 25% testing splits were generated randomly. Next, the baseline model was constructed. This had two purposes. First, the baseline model established a hurdle rate for model selection. Second, the baseline model served as a check that our dataset had signal we could exploit. For our baseline model, we trained an out-of-box decision tree on the training split and tested it on the validation set. Using ROC-AUC as our metric, we obtained a training set AUC of 1.00 and a validation set AUC = 0.636.

Given the difference between training and validation set AUC values, this unpruned decision tree grossly overfit; however, an AUC > 0.5 indicates there is structure to be exploited in the data. This is further supported by feature importance values (Figure 3), which reveals informative features including MajorCategoryID_65 (which corresponds to a water leak), followed by average assessed tax value per unit, median income, and the building's total assessed value.

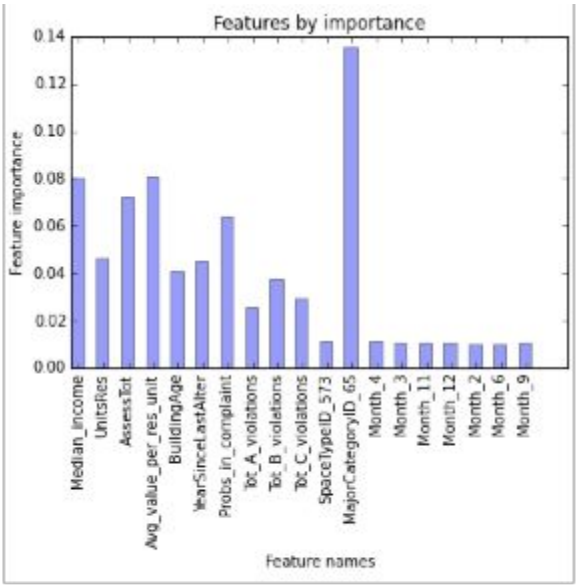


Figure 3: Feature importance in the out-of-box decision tree. Note Major Category 65 is a water leak

After establishing a baseline model (the out-of-box decision tree), several additional algorithms were explored, as described below:

1. **Support Vector Machines (SVM)**

   a. **Pros**: (1) Powerful method with geometric interpretation (rankings based on distance from discriminant). (2) Supports non-linear modeling (when kernelized).

   b. **Cons:** Support Vector Machines are computationally expensive. The libsvm-based implementation in scikit-learn "scales between $O(n_{features} \text{ x } n^2_{samples})$ and $O(n_{features} \text{ x } n^3_{samples})$". In fact, the runtime to train SVM using the Amazon Elastic Compute Cloud web service on a compute-optimized server exceeded our budget (both temporally and financially), and this method was not fully implemented.

2. **Decision Trees**

    a. **Pros:** (1) Highly interpretable. For example, in the out-of-box baseline decision tree baseline model, it was simple to compare feature importances and identify potential conditions that are highly informative (such as a water leak). (2) If allowed to grow to an arbitrary depth, decision trees can approximate any function of predictors. This is beneficial in our case, since we anticipate interactions may be significant (for example, interactions between the category and space variable).

    b. **Cons:** Generally lacks predictive power when compared to more complex models, such as random forests.

3. **Logistic Regression:**

    a. **Pros:** Highly interpretable, and supports statistical hypothesis testing.

    b. **Cons:** Won't capture feature interactions (unless explicitly included in feature space) or higher-order terms (unless kernel logistic regression used).

4. **Random Forests:**

    a. **Pros:** Highly powerful model that is anticipated to perform well (based on literature-reported performance on other classification tasks).

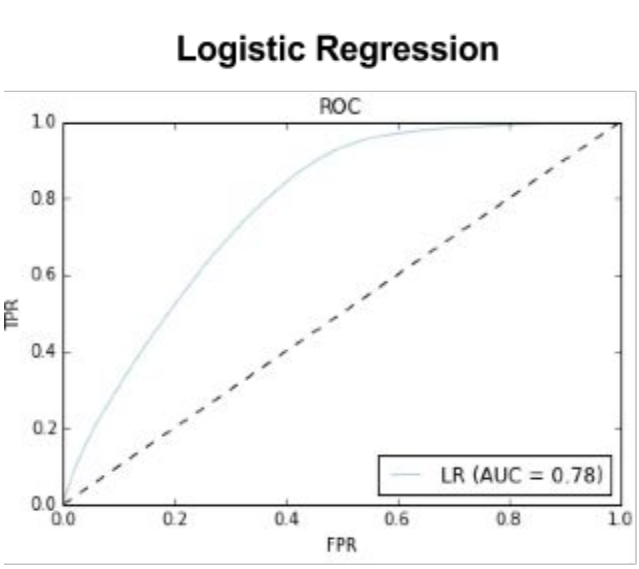
    b. **Cons:** Somewhat poor interpretability. The feature importance can be determined (by averaging across the trees in the forest), but this is still a black box model.

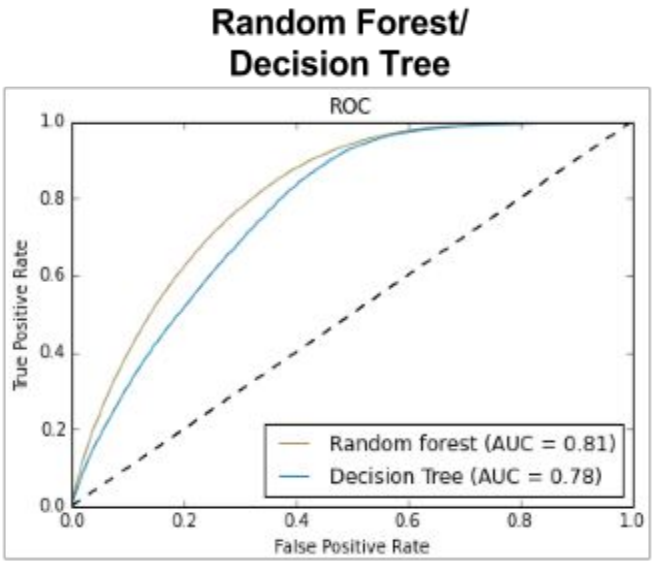
To compare possible models, we used grid-search to empirically optimize hyperparameters for each model family. Specifically, for each combination of hyperparameters in the grid, we trained the

model on the test set and validated it on the validation set using ROC-AUC as a metric. Grid search parameters for each model are specified below (again noting the computational demands of SVM exceeded our constrained resources):

Algorithm	Hyperparameters
Decision Trees (DT)	● Minimum sample split = {2, 4, 10, 25, 59, 138, 322, 752, 1755, 4096} ● Minimum sample leaf = {4096, 1755, 752, 322, 138, 59, 25, 10, 4, 2}
Logistic Regression	● Regularization = {L_1, L_2} ● Regularization parameter C = numpy.logspace(-30, 20, 50)
Random Forests (RF)	● Number of trees = {10, 100, 150} ● Max features = 15, following heuristic (number of features)$^{1/2}$ ● Minimum sample split and Minimum sample leaf same grid as DT

We then compared the best models from each family using the validation set ROC-AUC (see subsequent section for justification) to determine the optimal model. First, ROC curves for each model on the validation set are shown below (note they are shown on two separate axes due to the fact that models were trained and evaluated in parallel by two different authors).





LR Hyperparameters:
- C = 0.39069
- Regularization = L2

DT Hyperparameters:
- min_samples_leaf=322,
- min_samples_split=752,

RF Hyperparameters:
- min_samples_leaf=2,
- min_samples_split=4,
- n_estimators=150,
- max_features = 15

V. Evaluation

Various metrics were considered for evaluation, including lift, ROC-AUC, and accuracy. Since the data has a strong class imbalance (with approximately 83% of cases being negative), accuracy and lift were deemed inappropriate since they are sensitive to class imbalances, and ROC-AUC was chosen as the evaluation metric. Precision, Recall, and F-score are other possible metrics for binary classification, but these metrics focus on positive cases and may be inappropriate in this context, where negative cases still require HPD response.

Thus after training, each model was evaluated using ROC-AUC on the validation set (recall ROC-AUC ranges from 0 to 1, with 0.5 being random and 1 being perfect prediction):

Algorithm	Logistic Regression	Decision Tree	Random Forest
Validation set ROC-AUC	0.78	0.78	0.81

By way of interpretation, note that our best model (Random Forest) achieved an AUC of 0.81, so it will rank a randomly chosen violation-earning problem above a randomly-chosen non-violation-earning problem 81% of the time. Additionally, note an AUC of 0.81 represents a significant gain over our baseline model (out-of-box decision tree), which achieved an ROC-AUC of 0.636.

Finally, while our proposed Random Forest model is optimal (in the sense of AUC on the validation set), it may not be the optimal solution given constraints imposed by HPD. For instance, if HPD required a more interpretable model, we may suggest deploying the simpler logistic regression model even though it achieved 0.03 lower AUC performance.

VI. Deployment

To deploy the model, HPD should use our Random Forest classifier to predict whether each new complaint problem will produce a violation. Then, HPD management will use these predictions to estimate the number of expected violation cases within a given inspector's caseload. This will allow HPD management to balance inspectors' caseloads, under the objective of optimizing inspector performance and productivity (i.e. ensuring inspector performance is not degraded due to an overload of violation-producing cases). This could also be used by HPD management to benchmark individual inspectors against the expected number of violations in his/her caseload; if an inspector is producing many more or many fewer violations than our model predicts, management could use our model to normalize inspector practice. Note that it is imperative the model be used as a management tool, and that inspectors not have access to the model's predictions, as the predictions could bias inspection results.

When deploying our model, HPD should be aware that data mining is a process that results in a non-static model. Thus, they will need to work with our team as we continually modify the model based on our evolving business and data understanding. We anticipate that this model will need to be regularly retrained and evaluated following HPD management feedback.

Since we trained the model on data from a full calendar year (11/01/14-10/31/15), we don't anticipate our model to have poor cross-season generalizability. However, with new housing policies, new housing developments, and changes in city leadership as likely drivers of non-stationarity, this model should be regularly retrained and evaluated following key housing-related political and policy transitions.

VII. Reflecting on the Ethical Implications of Modeling for Public Service Delivery

Finally, we close by reflecting on an essential ethical consideration that drove our modeling objective. The New York City's Department of Housing Preservation and Development is charged with enforcing the Housing Maintenance Code (HMC). Given the need to ensure all tenants receive equal protection under the law, it would be unethical to deploy a model that ultimately denied certain tenants due protection. Instead, any model deployed by HPD must improve operations and not reduce the responsiveness of HPD to "low-probability" tenant complaints. Thus, we are proposing our model be deployed to determined expected numbers of violations in inspectors' caseloads; we are not proposing that our model be used to prioritize predicted-positives or to deny or reduce the inspection services offered to predicted-negative cases.

It is also important to note that this model (like most models) is strictly descriptive; the model being presented can predict whether tenant problems will result in violations (with some fidelity, as demonstrated in an AUC of 0.81). However, it in no way indicates whether a tenant problem should result in a violation, which is primarily a legal and ethical judgement formalized in the HMC. Our model is completely non-normative, and as such if there are problematic systematic biases in HPD's interpretation or enforcement of the HMC, our model would reinforce these biases.

Bibliography:

1. *The ABC'S of Housing*. New York: Department of Housing Preservation and Development, 2015. *The ABC'S of Housing*. NYC.gov, 2015. Web. 1 Dec. 2015. <http://www1.nyc.gov/assets/hpd/downloads/pdf/ABCs-housing-singlepg.pdf>.

2. *Enforcement Services*. New York: Department of Housing Preservation and Development, 2014. *Enforcement Services*. NYC.gov, Dec. 2014. Web. 1 Dec. 2015. <http://www1.nyc.gov/assets/hpd/downloads/pdf/code-enforcement-guide.pdf>.

3. *The Mayor's Management Report*. New York: Mayor, 2015. *The Mayor's Management Report*. NYC.gov, Sept. 2015. Web. 1 Dec. 2015. <http://www1.nyc.gov/assets/operations/downloads/pdf/mmr2015/2015_mmr.pdf>.

4. *HPD Open Data*. New York: Department of Housing Preservation and Development, 2015. *NYC*. NYC.gov, 2014. Web. 1 Dec. 2015. <http://www1.nyc.gov/assets/hpd/downloads/pdf/BuildingsOpenDataDoc.zip>.

5. *The Council of the City of New York*. New York: Department of Housing Preservation and Development, 2015. *NYC*. NYC.gov, 21 May 2014. Web. 1 Dec. 2015. <http://council.nyc.gov/downloads/pdf/budget/2015/15/eb/hpd.pdf>.

6. "Scikit-learn." *: Machine Learning in Python — 0.17 Documentation*. N.p., n.d. Web. 18 Nov. 2015.

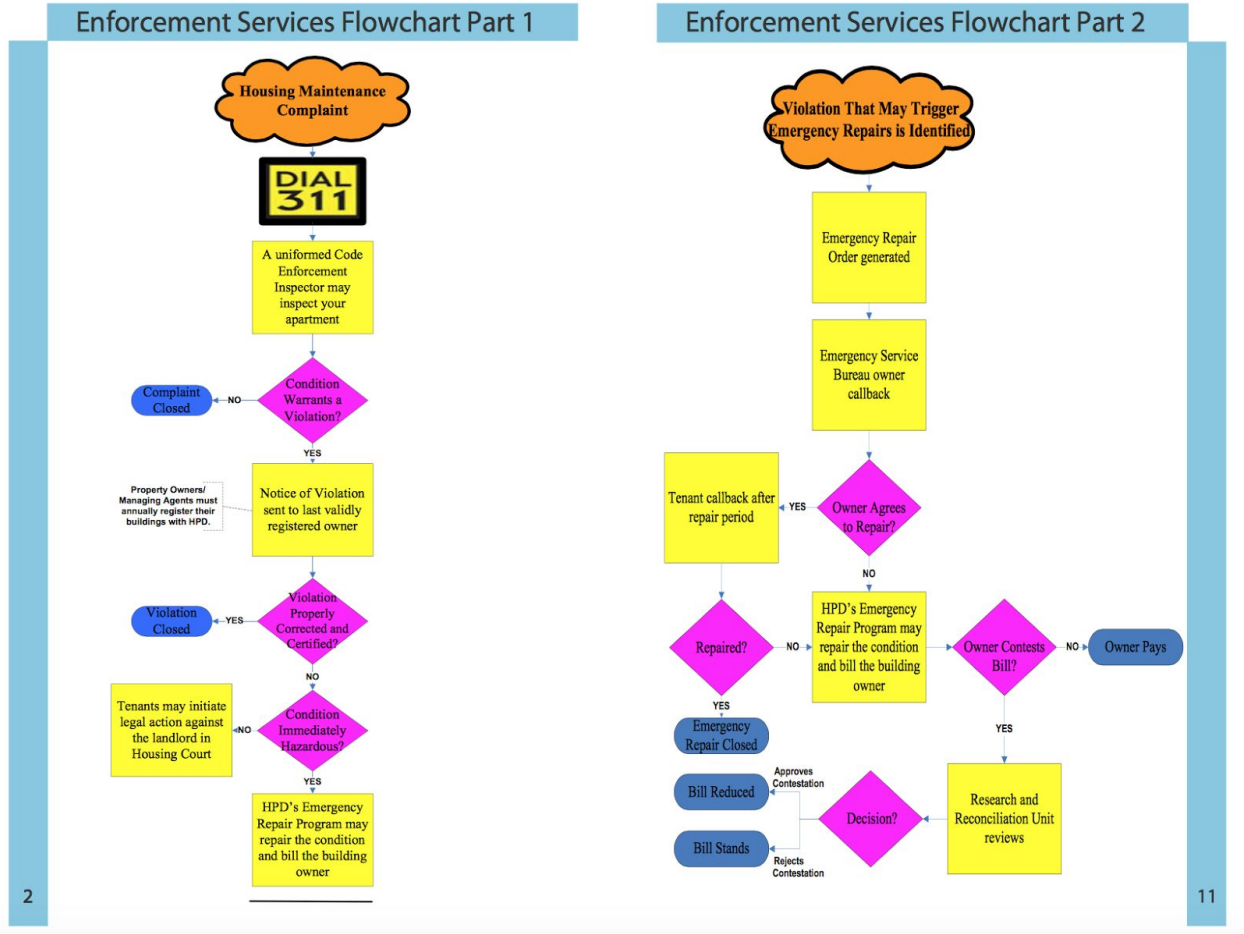
7. D'Alessandro, Brian. "Introduction to Data Science Data Mining for Business Analytics." NYU - Intro to Data Science. New York University, New York, NY. 2015. Reading.

8. Brian D'Alessandro. "Briandalessandro/DataScienceCourse." *GitHub*. N.p., n.d. Web. 18 Nov. 2015.

Appendices:

Appendix A. Housing Complaint and Inspection Flowchart.

The following flowchart describes the enforcement process, beginning with a housing maintenance complaint. The essential goal of the analysis is to build a model that can predict the answer to the first question in Flowchart Part 1 -- "Condition Warrants a Violation?".



Flow chart describing the enforcement process, beginning with housing maintenance complaints. Source: NYC HPD (2014)

Appendix B. Parsing Status Descriptions.

To construct our binary target variable, we parsed human-readable status descriptions from the Housing Complaint Problems dataset. Following exploratory analysis, the following strings were found to partition the complaint problems from 11/01/14 to 10/31/15:

No Violations Issued (Target = 0)	Violations Issued (Target = 1)	Duplicated or Unlabeled Records (dropped)
● "not able to gain access" ● "unable to access" ● "inspected the following conditions. No violations were issued." ● "Heat was not required at the time of the inspection. No violations were issued" ● "advised by a tenant"' [*note this string is used in cases where HPD contacted the tenant a specific concern was addressed, i.e. 'advised by a tenant that heat was restored'.*] ● "conditions were corrected"	● "Violations were issued" ● "A Section 8 Failure was issued."	● "Violations were previously issued" ● "conditions are still open" ● "inspection to test the paint for lead" (i.e. follow-up inspection to be conducted)

Appendix C. Team Member Contributions.

All team members contributed collaboratively to the term project.