HBURS: a housing bureaucrat prediction AI

Alessandro G. Magnasco - amagnasco@gradcenter.cuny.edu
For CUNY GC - DATA 71200 - Summer 2022

Abstract

When analyzing governmental actors' selective enforcement of protections and remediations against human rights violations, application of supervised and unsupervised machine learning models can derive new insights on biases and unseen trend development. Using open data on housing code violations by commercial residential properties in New York City, our classification attempts to track building selection for the stricter Alternative Enforcement Program. We find that even poorly implemented binary classifiers can replicate 4-dimensional enforcement decisions with 78% accuracy and 74% positive precision (n=32986). Dimensionality reduction and cluster analysis using unsupervised methods are functional on this low-complexity sparse dataset, but should return large efficiencies with future additions of dozens of additional dimensions. The author considers this proof-of-concept sufficient for additional work.

Introduction

In New York City (NYC)[1], the human right to adequate housing is not fully recognized or protected by the Human Rights Law or Commission on Human Rights[2]. The condition and management of housing in the City has deteriorated to a point requiring a public health emergency to be declared by the State of New York[3]. This project is part of a series of analysis on key trends in local housing to determine focal points for grassroots human rights organizing.

As required by law, the City of New York's Department of Housing Preservation & Development (HPD) keeps several regularly-updated datasets on the NYC Open Data Portal[4]. These include the list of code violations found by its inspectors, and emergency repairs done by the City on buildings where the landlord did not correct such violations. HPD places certain serious offenders in

[1] Within Lenapehoking, the unceded homeland of the Lenni-Lenape, many of which were violently displaced by genocidal colonialism, and many whom remain.

[2] Establishment of protected classes and reasonable accommodations are not sufficient. See: United Nations Office of the High Commissioner for Human Rights, Fact Sheet No. 21 v1: "The human right to adequate housing" (https://www.ohchr.org/en/special-procedures/sr-housing/human-right-adequate-housing). Accessed July 2022.

[3] New York State, "Housing Stability and Tenant Protection Act of 2019", 2019.

[4] NYC Open Data platform, filtered for HPD datasets. (https://data.cityofnewyork.us/browse?Dataset-Information_Agency=Department+of+Housing+Preservation+and+Development+%28HPD%29) Accessed July 2022.

programs for stricter monitoring, enforcement, and/or legal restrictions. These are unfortunately restricted in capacity due to political considerations, only classifying a certain number of buildings into the programs even if others would have been theoretically eligible.

The Alternative Enforcement Program (AEP) is a good starting point for machine learning analysis, as it has a clear list of few selection parameters, all of which are available on open datasets, as well as the list of targeted buildings. "Selection criteria for AEP include the number of class "B" hazardous and class "C" immediately hazardous housing maintenance code violations and the dollar value of emergency repair charges incurred as a result of the work performed by HPD. Failure to correct the qualifying conditions may result in emergency repair charges, liens, and significant fees. Only multiple dwellings are selected."[5]

HBURS is a set of machine learning algorithms that uses official open data records on NYC housing to analyze trends in housing violations. The system is built in Python Jupyter Notebooks, mostly using the widely-used open source Pandas, Numpy, and Scikit-Learn toolkits. A supervised learning algorithm uses all the listed criteria of the NYC Housing Code to classify buildings as likely eligible for AEP or not, using the official selection list as training & test data. Then, an unsupervised learning algorithm finds trends through individual analysis of the supervised findings of false positives, false negatives, et cetera. For the proof of concept, all analysis is done on a cleaned dataset of official AEP selection criteria from 2015, for the 2016 batch of entrants into AEP; the unsupervised algorithm does not yet filter for subsets. Later versions will establish time-series analysis, review of subsets as well as the full dataset, other enforcement programs[6], and add in related parameters from other datasets[7] for a more complete analysis.

Merging and cleaning of the source data into the proof-of-concept dataset:

All buildings in NYC are assigned a unique geographic identifier using a parcel numbering system called Borough, Block, and Lot (BBL). This has proven to be the best way of correlating different building datasets, as identifiers such as BID are non-unique, and others are not universal. Each of the three datasets used was downloaded as CSV, imported as Pandas dataframes, filtered down by the relevant criteria, dropped rows with critical null values, ensured there were no duplicates, and aggregated by BBL number. Then, a full outer join was performed to combine the 3 datasets into a new “bg” dataframe using the BBL identifier as key. These initial steps proved to be some

[5] NYC HPD, “Alternative Enforcement Program” (https://www1.nyc.gov/site/hpd/services-and-information/alternative-enforcement-program-aep.page). Accessed July 2022.

[6] Both the Certification of No Harassment, and Underlying Conditions Program are upcoming candidates.

[7] The author looks forward to continuing collaboration with the Housing Data Coalition in this regard.

of the hardest parts of the project; future work will add more datasets using time-series comparison to start analysis from a firmer footing.

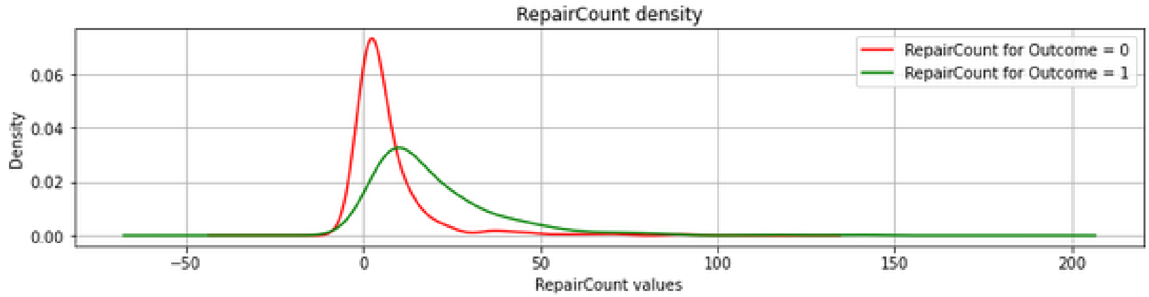
All datasets were accessed in June 2022.

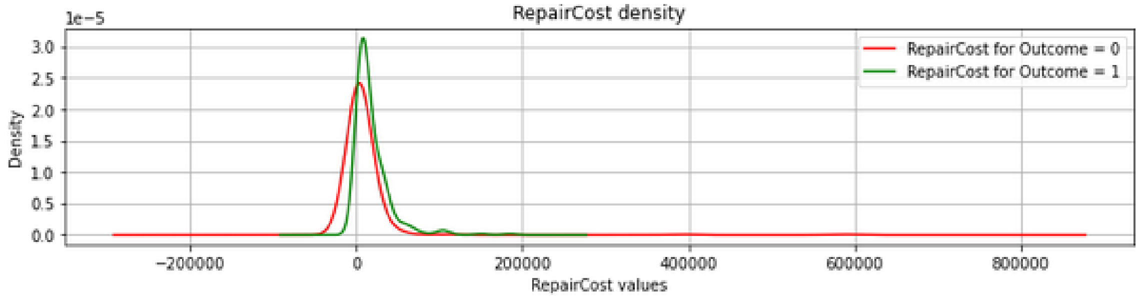
- Housing maintenance code violations
  - https://data.cityofnewyork.us/Housing-Development/Housing-Maintenance-Code-Violations/wvxf-dwi5
  - Shape: 7,383,759 rows, 8 columns; ~450MB
- Open market orders for emergency repair
  - https://data.cityofnewyork.us/Housing-Development/Open-Market-Order-OMO-Charges/mdbu-nrqn
  - Shape: 403,125 rows, 7 columns; ~21MB
- Buildings selected for alternative enforcement
  - https://data.cityofnewyork.us/Housing-Development/Buildings-Selected-for-the-Alternative-Enforcement/hcir-3275
  - Shape: 3,137 rows, 4 columns; ~98KB
- Final combined dataset “bg”
  - All steps from here on out use the final bg.csv compiled dataset available on Github.
  - Shape: 126,378 rows, 10 columns; ~11MB
  - Columns:
    - BBL: BBL unique identifier
    - OfficialViols: If in the AEP program, count of open violations (not used)
    - AEP: dummy target variable where 1 represents being in AEP
    - RepairCount: number of emergency repairs done by the City
    - RepairCost: sum of dollar cost of emergency repairs done by the City
    - ClassB: number of Class B building code violations
    - ClassC: number of Class C building code violations
    - (the other 4 columns were for testing and are not used)

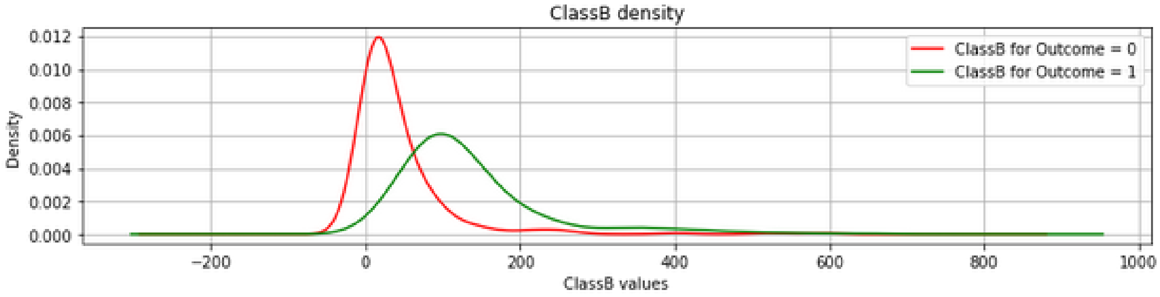
Supervised Learning

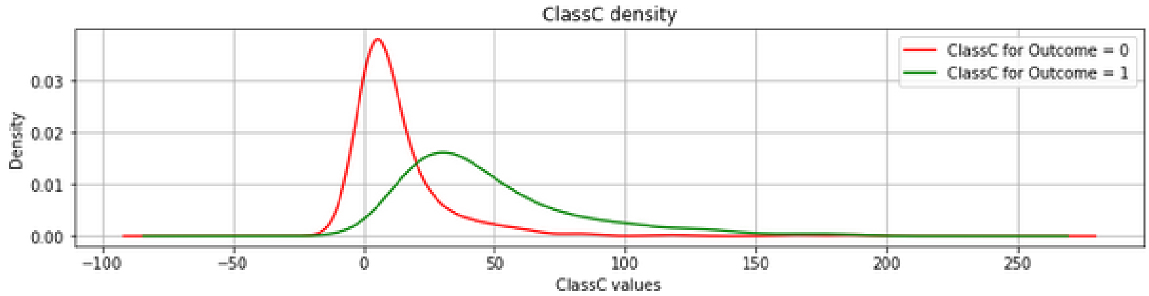
To determine the best binary classifier for this purpose, hyperparameter-optimized K-nearest neighbors and a gradient-boosted decision tree were tested on a downsampled and cleaned dataset, as follows: The bg.csv file generated previously was loaded as a Pandas dataframe, and any row with a null value for a critical parameter (RepairCost, RepairCount, ClassB, or ClassC) was dropped. Any remaining null values were filled in as 0. The resulting dataframe had n=32986, of which 248 were positive for AEP. This discrepancy in positive/negative ratio proved to be insufficient to train the tested algorithms correctly, so negative values were downsampled to 250 for an even split. Histograms and brief exploratory analysis showed this subset to be sufficiently balanced.

Feature density plots for the downsampled dataset (n=498) were clearly distinct:



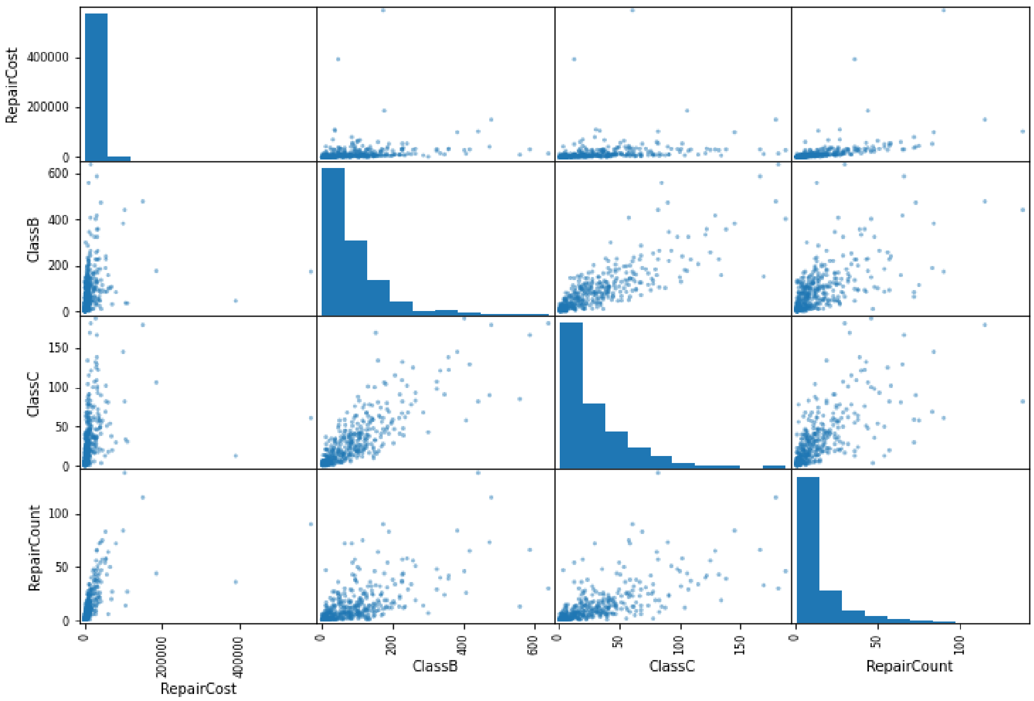


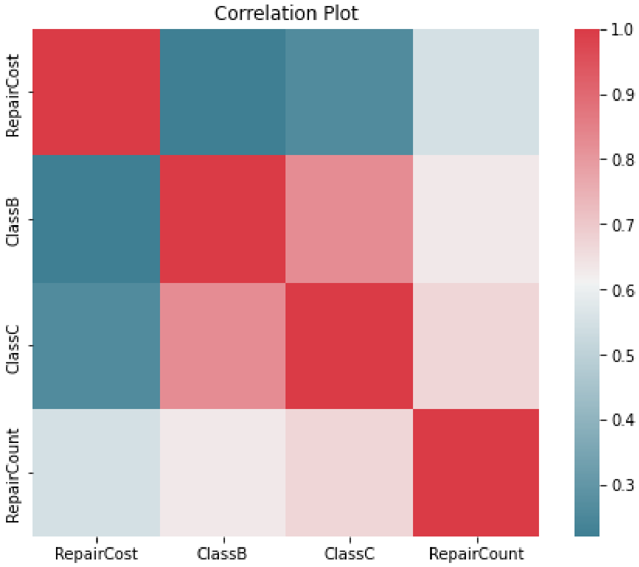




Training and testing subsets were generated using Scikit-Learn's StratifiedShuffleSplit function, with five splits on 20% testing and 80% training.

Perfunctory review and Pearson correlation on the downsampled training set were promising:





These charts had a reductive effect on the stress levels of the author, as the unsampled dataset had proved too unbalanced for analysis. Having fixed all major issues to the point where the binary classifiers could run correctly, the models were fit, run, and compared for relative efficacy.

k-Nearest Neighbors (KNN)

kNN is a simple non-generalizing learning model that classifies any given point to whatever the majority of its k closest neighbors are classed as. It performs well on clearly clustered, low-dimensional datasets.

This model was chosen as buildings in disrepair should form clusters. The model could not find any positive values initially, until the data was downsampled. A grid search algorithm determined the best-performing value of k to be 5, and was both fit and trained on the downsampled dataset.

Gradient-boosted Decision Tree (GBDT)

GBDT is a pre-pruning optimization on large and deep random forests of decision trees. By having a large number of small decisions, which each model applies to the error rate of the previous model, the resulting low variance allows the final model to perform better on new data.

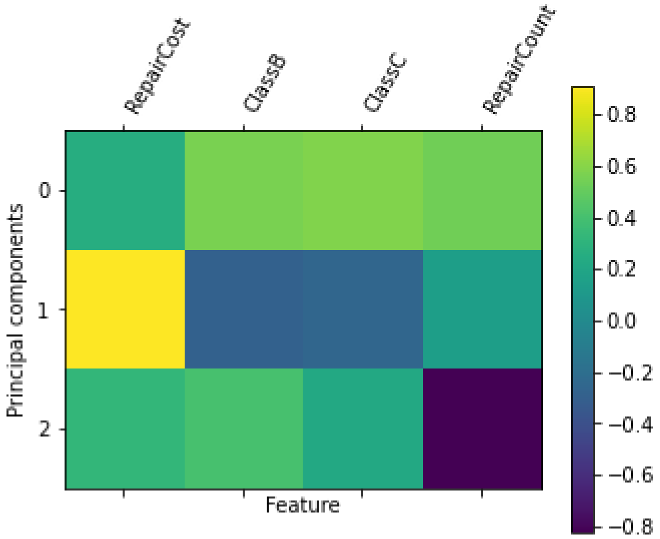
Decision trees were initially chosen, as AEP classification decisions should follow a logical hierarchy. Having low-dimensional sparse mixed data, performance was in question. However, precision on positive identifications was about 6% accurate, so the algorithm was shifted to be random forests. Performance was still not optimal, so gradient pre-pruning was implemented. A grid search algorithm determined the best-performing hyperparameters on the downsampled dataset to be 400 estimators at depth 1, using Gini criteria, and square root for the maximum number of features. Since the optimal depth was determined to be 1, the gradient boost was not implemented, and therefore validation errors would need to be corrected with the staged_predict method; it could prove more efficient to use a different decision tree ensemble such as AdaBoost. Precision on positive identifications improved to 74% on the downsampled dataset, which was deemed sufficient for this proof-of-concept, but will be corrected in future.

Future work would also apply a MinMax scaler and variance threshold to the data to find support hyperplanes of building disrepair with support vector machines, as that algorithm would appear to be a good fit for this model. Additionally, we would test different downsampling ratios to determine the optimal amount of data that should be fed to the binary classifiers.

Unsupervised learning

Ideally, unsupervised learning would be carried out on specific subsets generated by the binary classifier: the false positives and false negatives being especially interesting. As a proof of concept, however, we used the same bg.csv dataset, and ran three algorithms: k-means, hierarchical, and DBSCAN. Although we started from the same dataset as the supervised ones, the preparation of the data was different here, standardizing the inputs to what the algorithms expect to ensure results would be accurate. Any row with null values for any parameters was dropped, and all other NA values were filled with 0. Instead of downsampling the data, a standard scaler was run. Both the scaled dataset and an unsampled copy dataset had a stratified shuffle split applied, with one split, into 20% testing and 80% training sets.

Since we still have too many parameters for proper visualization, a principal component analysis was applied to the scaled dataset for dimensionality reduction. Of the four remaining parameters, 94.94% of the total variance could be explained by only three dimensions:



To enable proper comparison, a random cluster was generated. To determine the optimal number of clusters for each algorithm, elbow/knee visualizations were rendered for both the scaled and unscaled datasets. The unscaled set appears optimal at 3 and the scaled at 4 clusters.

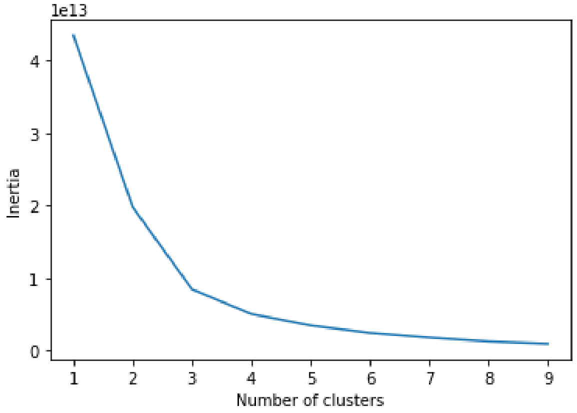


Figure 1: Unscaled optimal clusters

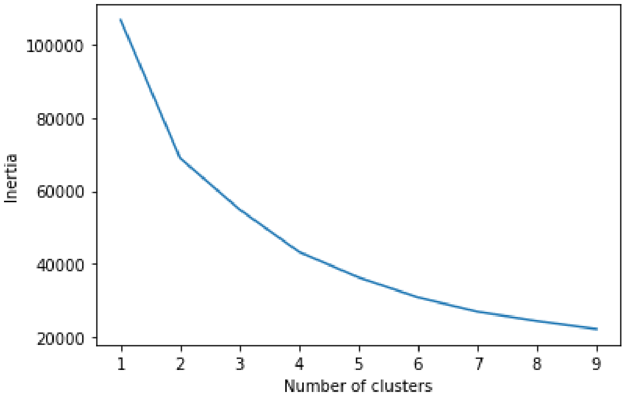


Figure 2: Scaled optimal clusters

Future work would use an automatic elbow locator to determine optimality.

k-means

This algorithm attempts to cluster data into k groups of equal variance, assuming the clusters are of roughly the same size and isotropic. As this is not the case with this dataset, it performed poorly in metrics, even after scaling and dimensionality reduction. PCA did make a visible difference as to how the clusters were defined in the scatter plot renderings.

hierarchical

This family of algorithms finds nested hierarchies of clusters by distance; visualized here is a dendogram of the Ward algorithm classifying the scaled dataset. The unscaled visualization was similar to that of k-means, but the scaled visualization applied different cluster boundaries. It performed slightly better than k-means, but still not ideally.

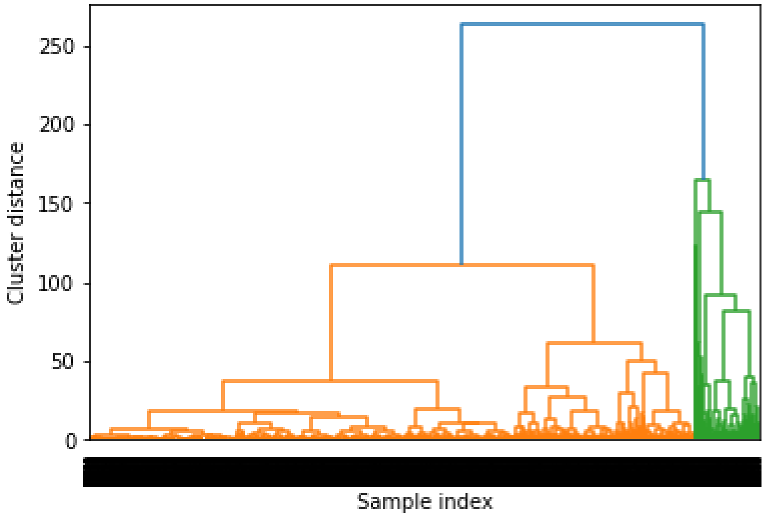


Figure 3: Dendogram for hierarchical clustering (ward)

DBSCAN

This algorithm determines clusters of any shape by finding areas of high density surrounded by other areas of low density. While the shape restrictions were more in line with our expectations for the dataset, optimal hyperparameters for DBSCAN were initially unclear. On further research, the Manhattan distance (“city block”) metric was established, along with epsilon consistent with the optimal cluster sizes (3 for unscaled and 4 for scaled datasets). A sample size of 8 was derived from the heuristic of having the number of samples be roughly twice the number of dimensions used. Both unscaled and scaled visualizations were visibly inconsistent with other models and appeared to make less sense visually; metrics were likewise negligible in performance. Future work will determine better hyperparameter values.

RESULTS & CONCLUSIONS

Supervised performance	Training score	N=498 test score	N=32986 test score
K-nearest neighbors	84.67%	71.00%	71.07%
Gradient-boosted decision tree	99.70%	78.00%	81.20%

Unsupervised performance (n=32986)	Adjusted Rand Score		Silhouette Coefficient	
	Unscaled	Scaled w/PCA	Unscaled	Scaled w/PCA
k-means	-0.0004	0.049	0.997	0.629
Hierarchical	-0.0004	0.056	0.877	0.689
DBSCAN	-0.013	-0.002	-0.521	0.936

We find that even having implemented these algorithms poorly, 4-dimensional enforcement decisions on AEP classification can be reliably replicated with machine learning to at least 78% accuracy and 74% positive precision on a small selected dataset of 32986 buildings with active violations in 2015, even when the actual entrants to AEP enforcement were only 250 of that group. While the proof-of-concept was established, significant future work will need to be done for these algorithms to be production-stage. Dimensionality reduction and cluster analysis using unsupervised methods were somewhat functional on this low-complexity sparse dataset, but should return large efficiencies with future additions of dozens of additional dimensions. The author looks forward to deepening their understanding of these algorithms and applying them more effectively in the coming months.