Early detection of properties at risk of blight using spatiotemporal data

Eduardo Blancas Reyes[*], Jennifer Helsby[*], Katharina Rasch, Paul van der Boor[*], Rayid Ghani[*], Lauren Haynes[*] and Edward P. Cunningham[**]

[*]Center for Data Science and Public Policy, University of Chicago

[**]The Department of Buildings & Inspections, City of Cincinnati

edu.blancas@gmail.com, jen@redshiftzero.com, kat@krasch.io, pvboor@gmail.com, rayid@uchicago.edu, lhaynes@uchicago.edu, edward.Cunningham@cincinnati-oh.gov

Abstract

Urban blight is a domino effect phenomenon: properties first fall into disrepair, then land values decline, and finally home abandonment and vacancy follows. This effect spreads from one home to another in the neighborhood, depressing values of nearby properties [8]. In partnership with the City of Cincinnati Office of Performance and Data Analytics and their Department of Buildings & Inspections, we used geographical data from the city and historical data on home inspections to train a Machine Learning model to provide proactive suggestions for property inspections targeted at catching blight early. Our best model reaches a precision of 70% for the top 6,000 predictions. This is a significant improvement over the discovery rate of the current approach, where 60% (in 2015) of citizen complaints result in the discovery of code violations. While our model can have a huge impact in tackling the blight problem, without field validation, the model can potentially have unintended consequences and ethical issues, such risks are being taken into account for the development of the project.

Keywords— Social Good; Supervised Learning; Urban Blight

1 Introduction

Blight impacts fundamental quality of life for affected neighborhood residents. Individuals experience serious financial losses due to the neighborhood blight problem: residents save to buy a home, and are then unable to sell their property when its value declines due to factors beyond their control. Blight is associated with additional negative outcomes in the city, including an uptick in crime, a decrease in availability of services, as well as a decrease in public health [10]. This effect is like a disease: if left untreated, it spreads from one home to another in the neighborhood, depressing values of properties nearby [8].

In partnership with the City of Cincinnati Office of Performance and Data Analytics and their Department of Buildings & Inspections, we use predictive analytics to provide proactive suggestions for property inspections in Cincinnati. The building inspections team is charged with keeping the properties of the city in a safe condition for residents. To assist these building inspectors in their task, we built a classification model that determines whether or not a home will have a building code violation.

2 Current Approach and Our Solution

In the City of Cincinnati, the current process addresses blight via reactive building inspections and code enforcement. Inspectors respond to citizen complaints and then work with property owners to bring their buildings into compliance with regulations. Under the current process, for homes that will one day become vacated, inspectors only receive a complaint in approximately 25% of cases. Thus, a large fraction of at-risk homes are unknown to the building inspectors who are trying to reduce neighborhood blight.

Around 6,000 inspections take place in Cincinnati every year - which represents roughly 4% of the total number of properties. 60% of homes inspected are found to have some type of building code violation. Enabling inspectors to identify properties at risk without relying on citizen complaints will allow building inspectors to start working with property owners earlier. This proactive approach will reduce the eco-

nomic and time costs of revitalizing neighborhoods in the city. Previous work done with the City Of Memphis [1] focused on how to efficiently identify homes in need of rehabilitation and to predict the impacts of potential investments on neighborhoods.

In partnership with the Office of Performance and Data Analytics and their Department of Buildings & Inspections, we use geographical data from the city and historical data on home inspections to train a Machine Learning model to provide proactive suggestions for property inspections targeted at catching blight early. This model generates a ranked list of properties that is used to determine which properties should be inspected, allowing Cincinnati to intervene and reverse problems while they are in the early stages.

Our goal is to find properties that are at risk of code violations as efficiently as possible, i.e. for the number of inspections performed in one year, we want to maximize the fraction that result in violations. This metric is called precision in the Machine Learning literature - we use precision at the top 6,000 scored parcels as the first evaluation metric.

To target our models at finding homes not already in the process of blight, we designed a metric, neighborhood blight score, that measures the level of blight in the community by assigning a score to each property.

3 Data sources

In this section we describe each dataset we used to train our predictive model, our datasets span different intervals, but we have the most data for 2012 - 2015 [1] so we decided to concentrate our analysis on this time frame[2].

Inspections. We used data from The Department of Buildings & Inspections which contains information about each inspection performed such as date, parcel identifier and the result of the inspection. This data provides the class labels we need for training the classifier.

Cincinnati Area Geographic Information System. Hamilton county provided us with spatial data that includes parcel-level information in Cincinnati in addition to spatial boundaries. We refer to this dataset as CAGIS in later sections [3].

Property Taxes. The auditor of Hamilton County audits homes in Cincinnati every three years. Home values are estimated and taxes are determined for every property. In addition, home values might be recorded if a sale occurs.

Census. By combining data from the U.S. Census Bureau with CAGIS, we matched properties in Cincinnati with their corresponding sociodemographic data in the U.S. Census from 2010.

311 Service Requests. Contains information about each complaint received e.g. property damage, trash on the streets, etc [2].

Building Permits. Contains information on every building permit granted, including a permit classification (e.g. wrecking, new building, the address and relevant dates in the permit process [4].

Crime. We use a database of crime incidents as recorded by the Cincinnati Police Department. The most important columns in this dataset include: date, crime description (e.g. burglary, vandalism) and address[5].

Fire Department. This dataset includes any incident that the Fire Department was called out on, including but not limited to fire alarms [6].

Property Sales. This dataset includes information on property sales (identified by an address), the date of sale, the previous owner and the new owner as well as general information about the building.

4 Data preparation

In this section we describe in detail the steps we followed to prepare our data for training. All source code for this project is available in a public repository [7].

4.1 Labeling the inspections data

The inspections data contains the entire process that every parcel undergoes when being inspected, for example, the data for an example parcel is shown in Table 1. We generate binary labels for the data, categorizing the outcome as 1 if any type of violation was found and 0 for no violation. Since we are interested in ranking our predictions to produce prioritized inspections, we will use the violation risk score that our model will give to each parcel to rank all properties in the city of Cincinnati.

[1] Exceptions are the 2010 Census dataset and the Inspections data (April, 2016), we subset inspections to match the rest of our datasets

[2] The links provided on this section are public ones, which may differ from the ones we use since public versions are anonymized

Parcel ID	*Date*	*Event*
0211	January 3, 2010	Reported
0211	January 6, 2010	Initial inspection
0211	January 10, 2010	Orders issued (Violation)
0211	April 20, 2010	Final notice

Table 1: An example inspection process for a single parcel

Dataset names	*Type of location data*
Inspections & Tax	Parcel ID
Census	Shape file
311 & Permits	Latitude, Longitude
Crime, Fire & Sales	Address

Table 2: Summary of location data

4.2 Taking into account the spatial and temporal dimensions

Every inspection has a spatial (which parcel) and temporal (when it happened) component. The location and size of a home is typically static in time, whereas most other features - such as the home value, structure, how many times it has been inspected, who lives there, whether it has been found in violation - change over time. Therefore the majority of the features generated must be associated with both a location and a point in time. To generate these features, we need to match the rest of our data to certain inspections taking into account both dimensions.

To match in time we need to compare the events' timestamps, whereas matching in space requires several intermediate steps. We use a PostgresSQl database with the PostGIS extension. The raw data contains location information in different formats, Table 2 describes which type of location information was available in our datasets.

For datasets with Parcel ID, we found their location using the CAGIS data. The census data comes in a Shapefile format, so we can use PostGIS functions to match census data with parcels. 311 and Permits observations include Latitude and Longitude for most of the data points. For the remaining datasets we only had an address, in order to locate them, we performed geocoding using the Census Bureau Geocoder Batch API [9]. We were not able to geocode all addresses (for example, for the Sales dataset we only geocoded 57% of the addresses). Geocoding addresses is an error-prone process, especially if data was human-entered: one reason for failure is that the Census API may fail to recognize typos or missing details in the address.

4.3 Model features

While most of our data contains information in space and time, some only contains information for one dimension. The following list presents a summary of the different features generated for the model:

Parcel-level features. Characteristics of each parcel (e.g. year built, parcel area, type of family).

Aggregated features. These features are aggregations in time **or** space (e.g. mean building value in the last 3 years, population density in the census block group).

Spatiotemporal features. For each inspection we took the location and found events within 50 m, 400 m, 700 m and 1000 m (we called this parameter `max_dist`), then for each distance we filter events that happened in the past 3 months, 6 months and 9 months (we refer to this parameter as `n_months`) at most from inspection date. With those restrictions, we computed frequencies for each event in the following datasets: 311 calls, Permits, Crime, Fire, Sales)

4.4 Constructing Training and Test Data

When creating our training and tests sets, there are three important parameters to take into account. The first one is `train_start_date` which defines the earliest date in our training set. The second parameter is `train_end_date` which defines the latest date used in our training set. To validate our model, we used a last parameter that defines the period of time starting in `train_end_date`, this third parameter is called `validation_window` and is used to define our test set, we use a value of 6 months for our current set of experiments.

For training the models we selected some feature sets, one including all the features we have and the rest including a subset of spatiotemporal features (`max_dist`=50m and `n_months`=3 months, 50m and 9 months, 50m and all `max_dist` values, 400m and all `max_dist` values, features up to 400m and all `n_months` values), the reason is that we want to see if spatiotemporal models will be able to better identify local effects, since they contain data at a very granular level.

5 Model evaluation

We used Python's scikit-learn package to train our models with the following classifiers: AdaBoost, Random Forest, Extra Trees, Gradient Boosting, Logistic Regression and

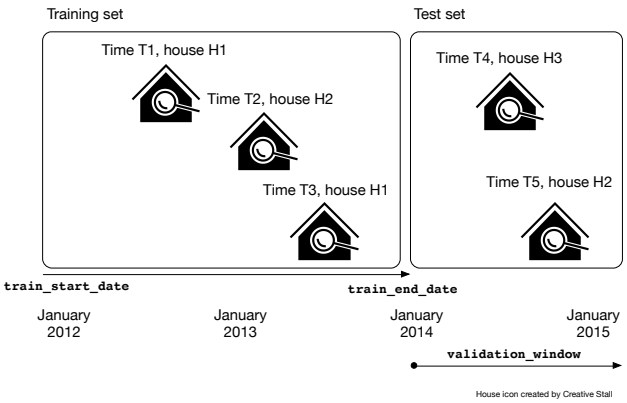

Figure 1: Building the train and test sets for temporal cross-validation

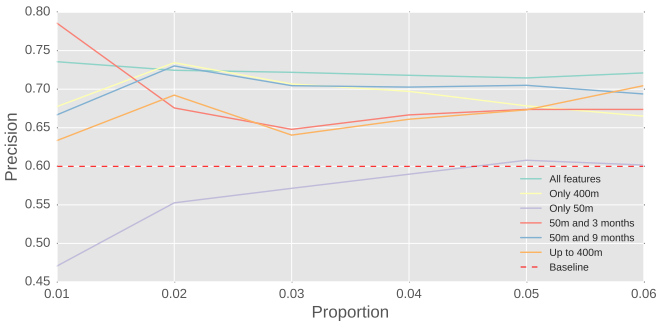

Figure 2: Precision curve (up to 6%) for selected models. Training set from Jan 2013 to May 2015

Support Vector Machines. For each of the model types selected, we trained on different combinations of hyperparameters and then selected the *best* one from the entire pool.

The specific goal of the City of Cincinnati Inspections Team is to maximize the number of inspections that produce a violation (true positives) while minimizing the number of inspections that find no issues (false positives). Since the city has limited resources to inspect properties and they can only inspect a fixed number of properties, we want to maximize the precision while inspecting a small number of properties that our models predict to be most at-risk.

We are interested in knowing the precision for the top 6,000 (4%) parcels in Cincinnati (since that's roughly the number of inspections done per year) but we have limited labeled data (not all parcels have ever been inspected and we build the test set using 6 months of inspections). Depending on the year this represents between 3200 and 4000 inspections, leaving most of our data unlabeled. Strictly speaking, we cannot compute the precision at 6,000 since we don't have enough labeled data, so we rank every parcel, take the top 6,000 and compute the precision using the labeled data and ignoring the unlabeled data points, we did the same for other precision values. In addition to optimizing for precision, we also need to consider that we want to flag properties at-risk in areas where blight hasn't taken hold, we discuss both problems in the following sections.

5.1 Selecting features to optimize for precision

Figure 2 shows the precision curves for selected models with different feature sets. As we can see, *all features* models have the best performance in terms of precision. After evaluating several experiments, we confirmed that models with all features generally outperform *spatiotemporal models*.

`train_end_date`	*Feature set*	*Prop. below median*
Dec 31, 2013	50m, 3 months	4.75%
	All features	0.37%
Jul 1, 2014	50m, 3 months	18.50%
	All features	8.69%
Dec 31, 2014	50m, 3 months	9.13%
	All features	6.61%
May 1, 2015	50m, 3 months	2.57%
	All features	2.12%

Table 3: Percentage of top 4% flagged properties with `neighborhood_score` below the median. `train_start_date` is January 1, 2013 for every model

5.2 Identifying models flagging properties in non-blighted neighborhoods

Since our goal is not only to find properties at risk of blight, but to do so in early stages, we incorporate a new metric called `neighborhood_score`. This metric assigns a score for every parcel counting the number of unique violations over the number of unique inspections in the past 6 months and within 500m. Since only a small fraction of properties and some areas in the city have not been inspected, we cannot rely in the neighborhood score in every parcel, for that reason we calculate the `inspections ratio` (unique inspections/number of houses) and ignore `neighborhood_score` for properties with *inspections ratio* below the city's median. Once we filter out parcels with low `inspections_ratio`, we calculate the city's median `neighborhood_score`, take the top 4% predictions for each model and compute the percentage of properties with a score below the city's median, this gives us a sense of how blighted the surroundings of each property are. Table 3

shows a comparison between our *All features* model vs. the model with the most granular spatiotemporal features. We see that the spatiotemporal model has a higher proportion of top predictions in areas with low `neighborhood_score` while still maintaining a good precision value. This trade-off is critical when selecting which model to use for generating a list of properties for The Department of Buildings & Inspections.

5.3 Selecting a model

Selecting a model to flag properties at-risk of blight is not trivial and even if we could select the best one today it won't be as good in future years. For that reason we need to continuously train and evaluate models as new data comes into the system. For the purpose of this paper, we are using the discovery rate for 2015 (60%) as a reference, so we take the models that included 2015 data for training (until May, 2015) and selected one by balancing both precision and `neighborhood_score`. Our best model is a spatiotemporal one with `max_dist`=400m and all values for `n_months`, such model achieves a precision at 4% of 0.70 and has 4.51% of properties with `neighborhood_score` below the city's median.

6 Ethical Considerations

While our model can have a huge impact in tackling the blight problem, it is important to understand the limitations of the data it is trained and evaluated on. Since the labels we are using come from a biased inspection process (only a 27% of all parcels in the city have ever been inspected), acting on the model without further field validation can potentially have unintended consequences and ethical issues. More concretely, because the legal process that follows a building code violation can lead to homeowners facing lawsuits (when the code violation is not fixed) and in some cases, jail, it is critical to be aware of the validity of the model and the consequences of taking action based on the risk scores. We have discussed these issues with the City of Cincinnati team and are actively working with them to mitigate this risk.

In general, the ethics behind data-driven actions that affect people's lives in a significant way need to be an important consideration. Organizations such as the City of Cincinnati looking to use data-driven approaches to improve processes within the organization should always understand the assumptions put into the models as well as biases that were present in the data that was used to train the models Furthermore, organizations should be transparent about the use of these tools and inform citizens why they have been flagged as at-risk (which also imposes a challenge in the realm of model interpretability). It is critical to remember the ultimate objective of this work as well as the City of Cincinnati is to improve the quality of life for residents by improving neighborhoods, and making them safer and healthier places to live, not maximize the number of violations (or fines) found per year.

7 Future work

This project is an ongoing effort and we see several avenues for improving upon our work so far.

7.1 Field Testing

The output of our project is a ranked list of parcels to inspect[3]. We are working closely with The Department of Buildings & Inspections to evaluate our model in a field test. To this end, we first create features for all properties in the city for the desired date of inspection, then we use our best model according to our performance metric and predict on all properties.

From the ranked list of properties, we select a subset that will inform our model the most. For example, if our model is predicting blight in a seemingly non-blighted home just for being in a blighted neighborhood, we would suggest an inspection there. Similarly, we suggest an inspection if our model predicts no violations in a blighted home just for being in a non-blighted neighborhood. We expect to start our field test in the following months.

7.2 Feature generation and selection

One important potential improvement is geocoding more addresses, especially for the Sales datasets where we lost a considerable amount of data, which could be a source of bias in our current model.

The spatiotemporal features are at a very granular level, but are basic. A potential way for further improving the model is to create more complex features. Furthermore, our spatiotemporal parameters `n_months` and `max_dist` were used to set a limit in which data we used to create features, but we could take another approach and use the distance/time as a weight for features.

[3]Even though we approached the problem as a binary classification, we are not using the predicted class, but the raw score predicted by the model.

Furthermore, we could improve our feature selection process. We are currently selecting features based only on their spatiotemporal parameters, a potentially better approach would be to use a feature selection algorithm to better prune non-informative features.

8 Conclusions

In this paper we presented a predictive approach for prioritizing city inspections as tool to identify and prevent urban blight in the city of Cincinnati. Our model is built upon a number of parcel-level features and spatiotemporal features and predicts whether a home is at risk of having a building code violation in the near future. Using this model, the city can increase the precision of their building inspections from 60% to 70%. This model can also be of use by other city agencies. For example, several community development corporations are active in Cincinnati, purchasing and renovating blighted properties to increase the attractiveness of their neighborhoods. We also identified ethical concerns that need to be considered before deploying such a model, to ensure that this work helps improving neighborhoods and making them safer and healthier places to live. In addition, it is also important to note that this type of work is only a component of a larger urban planning strategy aimed at tackling blight and urban decay and needs to be used in conjunction with other tools.

Acknowledgments

The work described in this paper started as part of the The Eric & Wendy Schmidt Data Science for Social Good Fellowship in 2015 and later on, the project continued at the Center for Data Science and Public Policy at The University of Chicago. We thank Chad Kenney, Cincinnati's Chief Performance Officer, and his team for their commitment and support to this project.

References

[1] B. Green, A. Caro, M. Conway, R. Manduca, T. Plagge, and A. Miller. Mining administrative data to spur urban revitalization. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 1829–1838, New York, NY, USA, 2015. ACM.

[2] City of Cincinnati. Cincinnati 311 (Non-Emergency) Service Requests. `https://data.cincinnati-oh.gov/Thriving-Healthy-Neighborhoods/Cincinnati-311-Non-Emergency-Service-Requests/4cjh-bm8b`, 2012. [Online; accessed 25-May-2016].

[3] City of Cincinnati. Cincinnati Area Geographic Information System (CAGIS). `http://cagismaps.hamilton-co.org/cagisportal`, 2012. [Online; accessed 25-May-2016.

[4] City of Cincinnati. Cincinnati Building Permits. `https://data.cincinnati-oh.gov/Thriving-Healthy-Neighborhoods/Cincinnati-Building-Permits/uhjb-xac9`, 2012. [Online; accessed 25-May-2016].

[5] City of Cincinnati. Cincinnati Crime Data. `https://data.cincinnati-oh.gov/Safer-Streets/Police-Crime-Incident-Data/w7vh-beui`, 2016. [Online; accessed 25-May-2016].

[6] City of Cincinnati. Cincinnati Fire Department Data. `(2015)https://data.cincinnati-oh.gov/Safer-Streets/2015-Cincinnati-Fire-Department-Incident-Data/96sp-aysv`, 2016. [Online; accessed 25-May-2016].

[7] Data Science for Social Good Github. Cincinnati Project Repository. `https://github.com/dssg/cincinnati`, 2015. [Online; accessed 8-February-2016].

[8] The National Vacant Properties Campaign. Vacant properties: The true costs to communities. `http://www.smartgrowthamerica.org/documents/true-costs.pdf`, 2005. [Online; accessed 7-February-2016].

[9] United States Census Bureau. Geocoder. `http://geocoding.geo.census.gov`, 2010. [Online; accessed 15-January-2016].

[10] R. J. Sampson. *Great American City: Chicago and the Enduring Neighborhood Effect*. The University of Chicago Press, 2012.