Blight Final Report

Brad Allen

July 16, 2016

Submit a brief report describing your features, your method, and the accuracy of the model. If your method is effective, provide sufficient detail to convince the reviewer of this. Be detailed and specific!

Executive Summary

For the capstone project to the "Data Science at Scale" course offered by the University of Washington, we were asked to do an analysis of blighted properties within the City of Detroit.

We were asked to, based on commonly accepted predictors of blight, develop a model to determine which individual homes would become blighted / up for demolition. The datasets provided can be found at the Detroit Open Data Portal and the course Github repo.

After transforming and preparing the data for analysis, my model (random forest) was able to accurately predict the blighted properties 75.7% of the time with a Kappa statistic of 0.5149, which Landis and Koch would describe as providing "moderate agreement."

All of my work can be found and used for reproduction in my Github repo at this link.

```
# Confusion Matrix and Statistics
#
# Reference
# Prediction False True
#     False   197    5
#     True     93  109
#
#               Accuracy : 0.7574
#                 95% CI : (0.7126, 0.7984)
#    No Information Rate : 0.7178
#    P-Value [Acc > NIR] : 0.04177
#
#                  Kappa : 0.5149
# Mcnemar's Test P-Value : < 2e-16
```

Approach

Some brief domain research provided by the course gave some preliminary clues for successful modeling—for example, Morckel (2013) found that three factors predict housing abandonment at the neighborhood level: market conditions, gentrification, and physical neglect.

Particularly counterintuitive to me was the "gentrification factor": such as, the percentage of properties built prior to 1945, the percentage of residents over 65 years of age, the percentage of residents 25 years and older without a bachelor's degree or higher, and the percentage of residents who are in poverty.

This was not a suitable avenue for what we were provided, however: 311 call data, crime reports, and blight violations. This led us more towards viewing physical neglect. The literature also emphasized the challenges in defining housing abandonment and creating predictors: for example, homes that are close to being fully blighted and up for demolition may actually have their larceny rates *drop* due to the fact that there is less that is available to steal. As a result, analyses strictly based on the rate of crimes may reach a point in a building's life at which they may begin to misclassify.

I began my approach by restructuring the call, crime, and blight record data to reflect the status of Detroit homes on a building-by-building basis.

Data Processing

Data processing was by far the most time-intensive component of this exercise. All of the feature data we were provided was in a "per incident" format and needed to be repurposed to a "per building" format. A second issue was that all of our data were for buildings that presumably had reason to be blighted—therefore, if we based our "building-by-building" database using this information exclusively, we would likely have a bias in our set that wasn't representative of Detroit housing more broadly. To get around this, I added the Parcel Points Ownership dataset from the Detroit Open Data website linked earlier.

To get to my final dataset (which can be found in my repo linked in the Exec Sum), I alternated between using Python notebooks for fast `FOR loops` and R for cleaning, joining, and statistical analysis.

My first exercise involved cleaned all of the files so that each would have a Latitude field labeled LAT and a Longitude field labeled LON. I used R for cleaning and then I used Python to create my master data set: I wanted to develop a FOR loop to match files that have proximate LAT and LON fields - I noticed that R did not work well for these loops - it got stuck with information in memory and has performance issues.

I used a few tools to visualize the data (based on the great feedback here): CartoDB and the FME Workbench for doing ETL activities. I didn't notice any major discrepancies - I did not try to match a text address to the LAT / LON provided. I also did "programmatic visualization" - some high level EDA to get a sense of the shape of the data and what was available to me.

Using the LAT and LON files from the Parcel Points dataset (~384K records), I matched all demolition permit records that had a LAT and LON difference of less than .0002. This left me with 810 records matching. I then randomly sampled 810 records of the negative set to have a balanced dataset of 1620 records.

With this 1620 records, I looped the crime, blight, and 311 data and stored all matches in a dictionary. I created extra fields / features based on the frequency with wich each "type of" crime, blight, or 311 call occurred.

Model Features

With the created features, I ran a random forest model to try and get a sense of the difference in importance to the final output. It was unsurprising that the blight violations calls drove much of the accuracy in the model; however, I was surprised at the *extent to which* it dwarfed other features of crime and 311 calls. This ties back to the original research of "physical neglect."

```
#varImp() output of randomForest() model:
#
#                             Overall
# CALLS_COUNT               2.02973918
# CALLS_DUMPING             0.30152068
# CALLS_POTHOLES            0.49670267
# CALLS_WATER               0.37422692
# CALLS_ABANDONED           0.45088188
# CALLS_TREE                0.52644108
# CALLS_CLOGGED             0.92316501
# CALLS_TRASH               0.31293938
# CALLS_DPW                 0.07254137
# CALLS_TRAFFICSIGN         0.60961322
# CALLS_WATERMAIN           0.58805763
# CALLS_TRAFFICSIGNAL       0.10332144
# CALLS_STREETLIGHT         0.53195711
# CALLS_MANHOLE             0.02093165
# CALLS_HYDRANT             0.00000000
# BLIGHT_COUNT             89.58513785
# BLIGHT_COMPLIANCE        35.55500730
# BLIGHT_WASTE             12.09910740
# BLIGHT_REGISTRATION      18.55543623
# BLIGHT_WASTEACCUMULULATE  9.90612603
# BLIGHT_WEEDS             17.03895183
# CRIME_COUNT               4.37055292
# CRIME_MOTORCYCLE          0.00000000
# CRIME_PROPERTY            0.00000000
# CRIME_ASSAULT             0.00000000
# CRIME_STOLENVEHICLE       0.00000000
# CRIME_LARCENY             0.00000000
# CRIME_BURGLARY            0.00000000
# CRIME_AGGASSAULT          0.00000000
# CRIME_FRAUD               0.00000000
# CRIME_DRUGS               0.00000000
```

A strict decision tree was also helpful in generating new information; it was able to determine that 95% of properties with greater than 28 records were up for demolition. These types of heuristics (which are not present in random forest models) can be great guidance for an "on the ground" team doing preventative work.

With additional time, I would have preferred to include temporal aspects, such as how crime rates change as buildings get more blighted. Specific kinds of blight (e.g., "waste" might show up for truly blighted properties, where as "failure to comply" is just negligence.) or patterns of blight early in the "violations process" may also be useful for distinguishing between outcomes.

Model Accuracy

As mentioned in the Executive Summary, my model (random forest) was able to accurately predict the blighted properties 75.7% of the time with a Kappa statistic of 0.5149, which Landis and Koch would describe as providing "moderate agreement."

Reviewing the decision tree, it seems like the analytical tools did not drill too much farther beyond the strict counting of total violations, and so I would treat this model as a surface-level accuracy. With more time, I would have converted my dataset to a temporal view and looked for patterns as to how violations accumulate. Any advice or suggestions for how to improve the analytical process would be much appreciated as well. Thank you for reading!

```
# Confusion Matrix and Statistics
#
# Reference
# Prediction False True
#     False   197    5
#     True     93  109
#
#               Accuracy : 0.7574
#                 95% CI : (0.7126, 0.7984)
#    No Information Rate : 0.7178
#    P-Value [Acc > NIR] : 0.04177
#
#                  Kappa : 0.5149
# Mcnemar's Test P-Value : < 2e-16
```