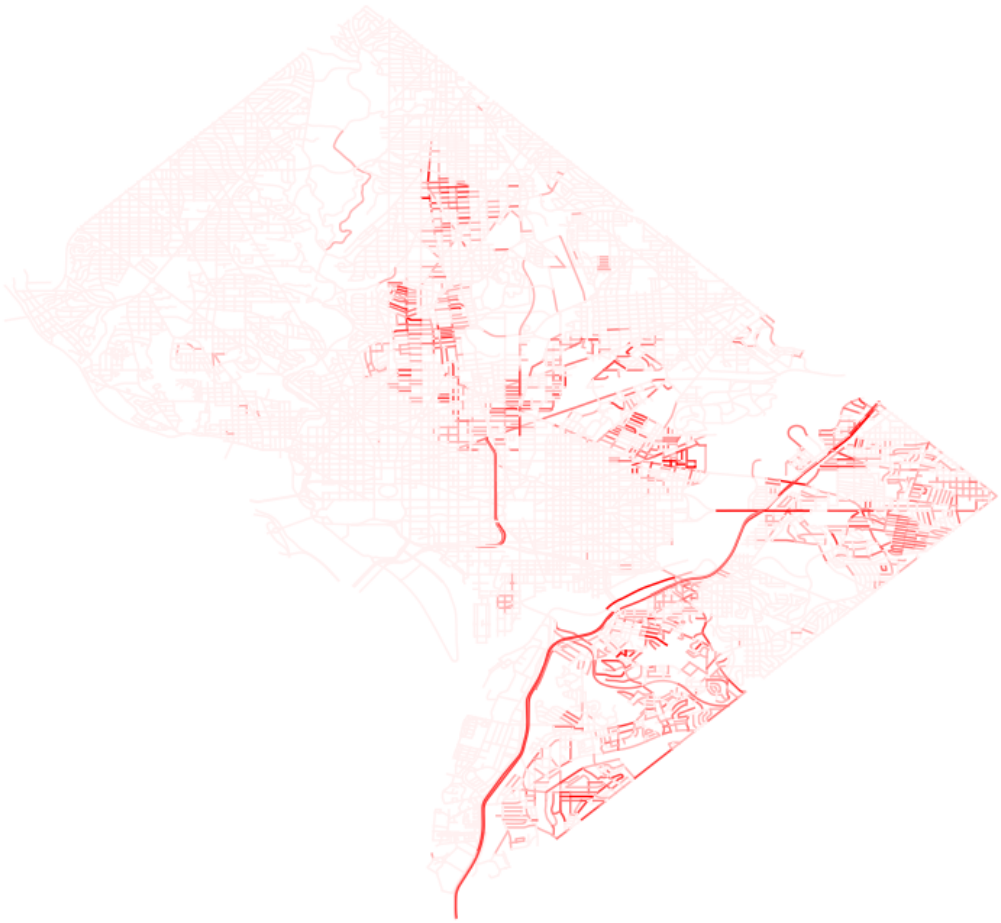
Udacity Machine Learning

Engineer Capstone Project

Predicting DC Homicides

Brian Friederich

Thursday, June 7, 2018



Predicting DC Homicides

1. Definition

Project Overview

"Aside from the murders, DC has one of the lowest crime rates in the country."

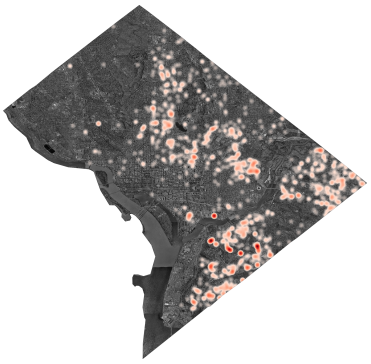
- Former DC Mayor Marion Barry

While rates of serious urban crime nationwide have dropped since a peak in the early 1990's and are now near historic lows, several major cities such as Chicago and Las Vegas are bucking the trend and have experienced an uptick, especially in homicides and other violent crimes.[1] While the police shooting of Michael Brown in Ferguson, Missouri and ensuing tensions surrounding related incidents have led some pundits to blame community-police tensions or weakened law enforcement, experts disagree on the underlying causes of this upswing in some but not most major urban areas. Evidence suggests diverging crime trends between increasingly safe cities such as New York and increasingly troubled cities such as Chicago have much older origins and are likely a result of complex and multifaceted conditions. Solutions, therefore, might be similarly multifaceted, and should move beyond the scope of merely increasing the capacity of law enforcement toward a variety of targeted public programs. With limited resources, however, cities could benefit from knowing not only which areas of their cities have experienced violent crimes, but what areas are most at risk and could likely benefit

[1] https://www.reviewjournal.com/crime/homicides/fbi-director-expresses-concern-over-murder-increase-in-us-cities-including-las-vegas/

most from targeted city programs addressing the factors that make violent crimes more likely.

In fact, several US cities, including Los Angeles and Santa Cruz in California are attempting just this with tools such as Predpol.[2][3] These approaches typically rely on neural networks and incorporate data such as tweets and news articles for more accurate prediction, but produce an algorithm that's hard to interpret and potentially open to racial, class, and other biases in the data fed into the algorithm. In contrast, two scholars at Rutgers devised a boosted tree algorithm called "Risk Terrain Modeling (RTM)"[4] which provides much more interpretable results leading to clearer policy solutions and more control over explicitly limiting or at least remaining aware of correlations with systemic inequalities. Scholars in an RTM project examining shootings in Newark found that 11 geographic features, mostly boiling down to specific locations people can hang out without becoming suspicious or attracting police attention, accounted for most of the variation in homicide rates geographically throughout the city.[5] In Newark, these locations were near open businesses in otherwise derelict or un-operating areas, with the open businesses providing cover for hanging out. The policy implication was that business revitalization in troubled areas and creating spaces that discouraged loitering could reduce violent crime without increasing policing. Similar experiments have been conducted in Kansas City, Missouri, Glendale, Arizona, and Chicago, Illinois resulting in actionable findings that not only show areas of higher predicted crime, but clear insights into what about these locations puts them at risk.



Heat Map of Homicides January 2008 - October 2017

Problem Statement

Although DC's homicide rates show a strong downward trend from a peak in the 1990's, its relatively high rate and wealth of publicly available data make it an ideal test case to further explore machine learning as a geographic predictor and analytic tool to help cities efficiently allocate resources to tackle homicides preventatively, and not just through increased policing. In this project, I plan to predict the likelihood each street segment in DC (portions of a street between two intersections or end of a street) experienced a homicide between 1 January 2007 and 31

---

[2] https://www.theguardian.com/cities/2014/jun/25/predicting-crime-lapd-los-angeles-police-data-analysis-algorithm-minority-report

[3] https://www.technologyreview.com/s/428354/la-cops-embrace-crime-predicting-algorithm/

[4] http://www.rutgerscps.org/rtm.html

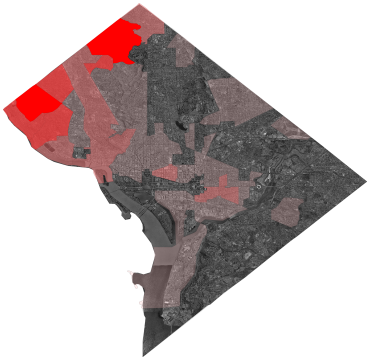
[5] Andrew Guthrie Ferguson. "The Rise of Big Data Policing." iBooks. https://itunes.apple.com/us/book/the-rise-of-big-data-policing/id1276191246?mt=11

Predicting DC Homicides

October 2017, as defined below in the Metrics section, through a boosted tree classification model using XGBoost based on neighborhood characteristics such as zoning and median property values. I will further simplify the model by turning homicide incidence into a Boolean feature instead of a continuous feature and running an XGBoost classification, although an XGBoosted tree regression could easily be used on raw homicide counts per street for a more granular prediction of how many homicides a given street is predicted to have experienced. Finally, I will visualize the results on map using QGIS. If the XGBoost model doesn't perform well, I may try a LASSO-regularized linear regression, which will automatically extract the most explanatory of the features for analysis.

Metrics

To evaluate my model, I will compare the results of my model to the mean homicide rates in DC, randomly distributed. More specifically, I will create an F1 score as if I had randomly distributed the correct number of street segments that experienced homicide among total street segments in DC.



Census Tracts by 2015 Median FAGI

If the F1 of my model performs better than the F1 for baseline model, then the results suggest my model might help better allocate resources than blanket police district-wide programs to tackle crime, and I will consider this a workable base model off of which cities can improve. I will aim for at least a 500% improvement in the F1 score, in other words a five-times-greater score. I will keep in mind that crime prediction abilities tend to be modest at best: A 21-month single-blind randomized control trial in three Los Angeles police divisions recently predicted crime with twice the accuracy of existing best practices, though correct predictions citywide still fell below 10%.[6] Those predictions were time bound and based off of time-stamped data such as weather and trending hashtags, however, which I ignore as it would add an extra dimension of complexity to my model.

Data Preprocessing

For my label, I downloaded the "street segments" shapefile from the DC government's open data website,[7] dividing streets in DC into 13,664 segments. I originally intended to use a publicly-available shapefile roughly dividing DC into city blocks, but found homicide data clustered along streets rather than within blocks. This makes further sense on the logic that crime rates on opposite sides of one street or alleyway are

[6] http://newsroom.ucla.edu/releases/predictive-policing-substantially-reduces-crime-in-los-angeles-during-months-long-test

[7] http://opendata.dc.gov/datasets

Predicting DC Homicides

Heat Map of City Service Requests for Graffiti in 2016

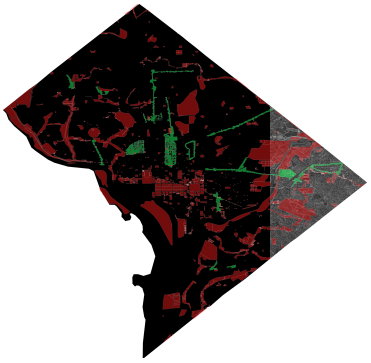
likely to be more similar to each other than crime rates between one side of a block and another. I indexed the street segments to match datasets later into one final dataset, and included street length, as longer streets may be at higher risk statistically of having experienced a homicide. I then extrapolated homicides from Kaggle's DC metro crime dataset[8] from 1 January 2008 to 31 October 2017. The original dataset had 342,867 rows and 32 columns, and was therefore too large to upload to GitHub via traditional means. I pared down the dataset to only include homicides, resulting in only 1,234 entries, and included the attributes of latitude and longitude of each reported incident to 13 decimal places. Actual homicide locations could be slightly different from total actual homicides due to unreported incidents and unsolved missing persons cases, moved cadavers, or clerical error in documenting crime locations. When mapped onto street segments with QGIS' 'Count Points in Polygon' tool to parse the street segments into segments where homicides had occurred and segments where homicides had not occurred within the dataset's timeframe. We should note that when a murder occurred at an intersection, QGIS arbitrarily assigned the point to one street on the intersection, which functionally labels the other streets as false negatives and negatively biases the factors on the other streets potentially contributing to homicide risk. The parsing resulted in 987 streets that had experienced at least one homicide. I opted to one-hot encode whether a street had experienced a homicide for ease of analysis later, though this information could be added back in for a regression-type analysis rather than a classification. I truncated other data when possible to only include the years 2010-2017 since the predictive features in my model were a snapshot of the present and did not necessarily represent the city's data in the medium to distant past, especially in rapidly gentrifying conditions.

My independent or predictive features were downloaded from opendata.dc.gov unless otherwise specified, and mapped onto the street segments with QGIS. I downloaded police districts as a shapefile rather than going by the police district number included in the crime dataset to sidestep potential clerical error (count of 7 districts). DC's 2010 census tracts shapefile had 179 census tracts, including information on 2010 and 2015 federal adjusted gross income (FAGI), population, total housing, vacant housing, and area in square miles, among other features. From these, I calculated the new features of median FAGI change between 2010 and 2015 (a proxy for gentrification), population density, housing density, and percentage of total housing vacant. Although I could account for street segments straddling police districts with one-hot encoding, the quantitative dimension of the census tracts proved a more difficult challenge. In the end, I visualized each street on a map and hand picked which census tract it should belong to from the plurality of the street's total area. FAGI was not available for 2015 for DC's 62.02 census tract, almost entirely made up of the mall, white house, and several other federal buildings, so I left these values as "nan." I added coordinates from

[8] https://www.kaggle.com/vinchinzu/dc-metro-crime-data/data

Predicting DC Homicides

a service requests csv dataset including 302,985 incidents, but was able to collapse this to reported graffiti, map a 100 foot radius around each incident, and then record whether a street segment intersected or touched at least one of these radii in one-hot encoding via GIS. I did the same for gas stations, metro entrances, liquor stores, and public schools (500 foot radii), further sub-identifying the metro line colors (blue and silver entrances were identical for all stops in the district, so I grouped them) and grade level of schools (elementary, middle, and high school, with other schools left out) one-hot encoded. I also one-hot encoded whether a street segment bordered or intersected federally-owned property or a designated main street commercial corridor based on shapefiles for each. The data was joined by street segment id and saved as a csv file in excel. For ethical as well as legal reasons, demographic information such as race, income, or national origin have not been included directly in my model, but could be inadvertently proxied through seemingly innocuous included features such as neighborhood, property value, etc., and their potential to bias my results must be kept in mind. While unfortunately many of these problematic correlations are systemic in US cities, and especially DC, we must be aware of their potential reflection in the results to avoid unjust or unfairly targeted solutions or resources.



Federally-Owned Properties (Red) and Designated Main Street Commercial Corridors (Green)

2. Analysis

Exploratory Data Analysis and Visualization

"There's blood on the sidewalk from someone I don't know

I step around it so I won't get it on my shoe

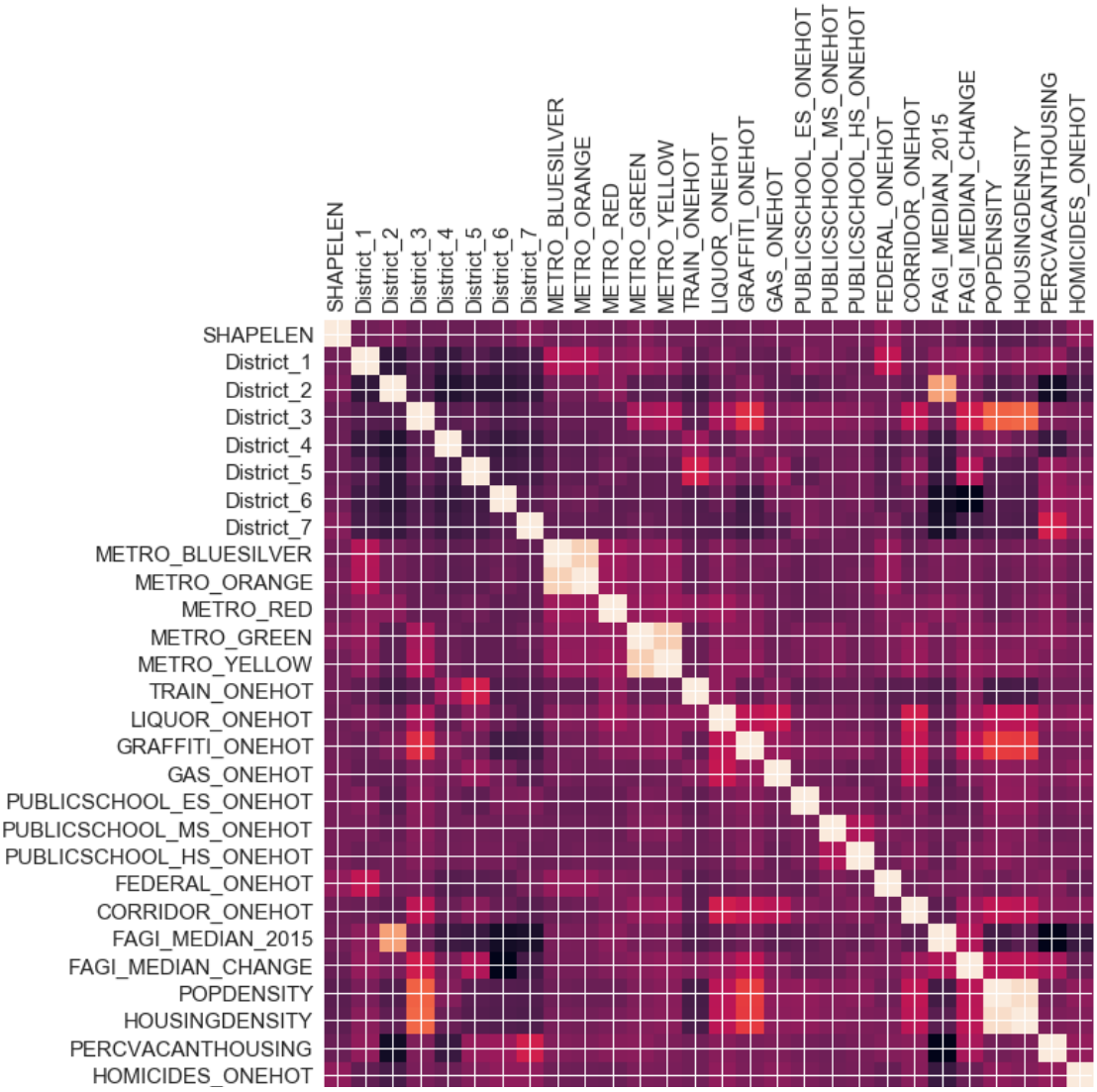
On my way to work in Washington D.C."

- SONIA, "Disappear Fear"

Cursory calculations on the data confirmed that my dataset had 13664 observations, or street segments, 987 of which had experienced a homicide within my timeframe, which came to 7.22% of streets. This was a small number, which indicated a sparse dataset on which a tree classifier like XGBoost would perform well.

For classification-type trees, regularization is typically not as much of a problem as in regression-type problems, so I declined to regularize my data. I did, however, run a correlation heat map matrix to make sure my features were sufficiently independent, which can be important for clean branches in trees that capture the feature in question and do not "steal" some of the predictive power of correlated features.

Predicting DC Homicides



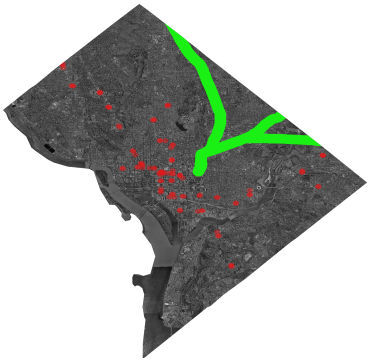
Correlation Matrix

The matrix showed high positive correlation among some metro lines, which makes sense since many lines share tracks with other lines for a majority of stops within DC. It also showed high positive correlation between 2015 FAGI and police district 2, historically the most privileged and one of the most expensive areas of the city composed of neighborhoods West of rock creek park. The small and more central District 3 had somewhat high positive correlations with population and housing density as well as reported graffiti, and strong negative correlation with vacant housing. Traditionally underserved Districts 6 and 7, comprising the area of Anacostia, had strong negative correlations

with median 2015 FAGI as well as change in FAGI, indicating gentrification's inability, so far, to extend across the Anacostia River. Lastly, percent vacant housing had a strong negative correlation with median FAGI.

After the heat map matrix, I ran a numerical correlation matrix to check correlations with streets that experienced a homicide, our model's label (see matrix in the appendix). Street proximity to a liquor store, percentage vacant housing in a street's census tract, and a street's location in District 7 all correlated positively with homicide incidence, while a street's presence in District 2 and the FAGI of the street's census tract correlated negatively with homicide incidence, all at the level of (-)0.1 or more.

Algorithms and Techniques



Areas within 500 Feet of a Metro Entrance (Red) or 1000 Feet of a Maryland Transit Train (Green)

I chose XGBoost's classifier algorithm for ease and simplicity compared with the deep learning solutions employed by more real-world crime prediction tools, as well as high accuracy compared with other algorithms that can easily fit on a local machine. XGBoost regularly wins in machine learning competitions, and is well suited to data like mine, with limited observations and features, some null values, and fairly weak correlation between most features. Essentially, the algorithm ran many "weak" classifiers, or classifiers with lower accuracy, and combined their average weights into one tree, which it turns out mathematically has much better accuracy with less risk of overfitting. I trained this XGBoost model with 5-fold cross validation on 70% of the data (9564 samples through Scikitlearn's train_test_split) to compile a final optimized algorithm. Finally, I tested this final algorithm on the final testing set of the reserved 30% (4100 samples) of street segments to assess its fit. For assessment, I used an F1 score, which balanced the weight of false positives and false negatives with true positives. This was a more appropriate scoring measure for this problem than using accuracy, given the clear potential legal, societal, and civic problems with neglecting a crime-prone area or falsely labeling an area as at elevated risk of homicide.

Benchmark

As discussed earlier, I calculated a baseline F1 score as if I'd calculated the correct number of streets in DC that experienced a homicide and then randomly assigned that number of streets as my prediction for streets at risk. To compute this, I took the 7.22% figure for streets that had experienced homicide and multiplied it by the number of streets that had experienced homicide to calculate the predicted number of true positives in my random predictor, . I also multiplied the number of streets that experienced homicide by 1 minus 7.22% to get a figure for probable false negatives, that is, streets that experienced homicide which I would

wrongly label as not at risk, or . I finally subtracted the number of streets that experienced homicide from the total number of streets to get total streets that did not experience a homicide during this period and multiplied that by 7.22% to get a number of probable false positives, or . I then ran those figures through a formula as follows to get my f1 score for my random predictor:

$$f1 = \frac{2\times(p(a)\times a)}{2\times(p(a)\times a) + (p(1-a)\times a) + (p(a)\times(1-a))}$$

The f1 score for my baseline was 0.014893127088544485, so any f1 score higher than that would be a better predictor than a random prediction.

3. Methodology

Implementation and Refinement

"The nation's capital didn't tally it's first homicide until six days after the new year, but the numbers quickly added up to eight and followed trends for the past two years."

- D.C. Witness,
  1 February 2017

Following a blog post by Jesse Steinweg-Woods, Ph.D. for advice on best practices with XGBoost,[9] I decided to run a grid search on parameters, two parameters at a time with three potential values each, when implementing my model. The grid search automatically gave me insight into optimal levels for each feature to get the best performance, and levels could have been further adjusted in performance could have been further optimized (this ended up not being the case; after the grid search in the notebook, I ran a more granular grid search with smaller intervals surrounding the previously optimized parameter and performance did not improve).

First, I ran the grid search on an XGB classifier with f1 as scoring, a 5-fold cross validation, and n_jobs as -1. Using -1 for n_jobs ensures the computation will be dispatched on all the CPUs of the computer. I kept learning rate, number of estimators, the seed, the subsample rate, the column sample by tree, and the objective (binary logistic so I could get probabilities rather than a simple 1 or 0) constant while running the grid search on max depth (3, 5, and 7) and minimum child weight (1, 3, and 5). After fitting the classifier to my training data, which returned

[9] Jesse Steinweg-Woods, Ph.D., "A Guide to Gradient Boosted Trees with XGBoost in Python", https://jessesw.com/XG-Boost/

deprecation warnings likely due to the “nan” values in my dataset, I printed the optimized grid f1 scores, finding that a maximum depth of 7 and a minimum child weight of 1 provided the highest score. These did not improve significantly with maximum depths of 6, 8, or 9, nor with minimum child weights of 2 or 4, not included in my final notebook to save space and run time.

```
[mean: 0.15845, std: 0.02048, params: {'max_depth': 3, 'min_child_weight': 1},
 mean: 0.14625, std: 0.01409, params: {'max_depth': 3, 'min_child_weight': 3},
 mean: 0.14076, std: 0.01873, params: {'max_depth': 3, 'min_child_weight': 5},
 mean: 0.18754, std: 0.04089, params: {'max_depth': 5, 'min_child_weight': 1},
 mean: 0.18814, std: 0.03414, params: {'max_depth': 5, 'min_child_weight': 3},
 mean: 0.18047, std: 0.03194, params: {'max_depth': 5, 'min_child_weight': 5},
 mean: 0.19498, std: 0.04387, params: {'max_depth': 7, 'min_child_weight': 1},
 mean: 0.19114, std: 0.05150, params: {'max_depth': 7, 'min_child_weight': 3},
 mean: 0.18948, std: 0.04519, params: {'max_depth': 7, 'min_child_weight': 5}]
```

I then ran another grid search, adjusted the fixed parameters to the optimized maximum depth and minimum child weight found above and giving the learning rate and subsample parameters ranges of (0.1, 0.5, and 0.01) and (0.6, 0.7, 08) respectively. After running these, I got the highest mean f1 scores on a learning rate of 0.5 and a subsample of 0.7. Once again, these did not improve with more granular adjustment and I left the more granular cross validation out of my notebook to save time and space.

```
[mean: 0.20303, std: 0.03567, params: {'learning_rate': 0.1, 'subsample': 0.6},
 mean: 0.19360, std: 0.05449, params: {'learning_rate': 0.1, 'subsample': 0.7},
 mean: 0.19498, std: 0.04387, params: {'learning_rate': 0.1, 'subsample': 0.8},
 mean: 0.19399, std: 0.02493, params: {'learning_rate': 0.5, 'subsample': 0.6},
 mean: 0.20495, std: 0.03549, params: {'learning_rate': 0.5, 'subsample': 0.7},
 mean: 0.19578, std: 0.04182, params: {'learning_rate': 0.5, 'subsample': 0.8},
 mean: 0.09975, std: 0.00909, params: {'learning_rate': 0.01, 'subsample': 0.6},
 mean: 0.09996, std: 0.01155, params: {'learning_rate': 0.01, 'subsample': 0.7},
 mean: 0.09019, std: 0.01529, params: {'learning_rate': 0.01, 'subsample': 0.8}]
```

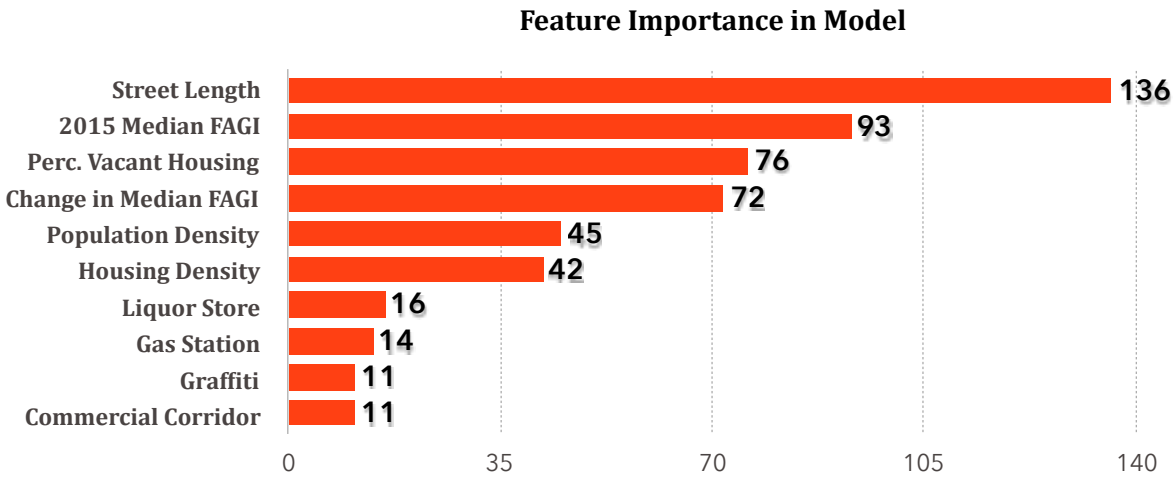
4. Results

Model Evaluation and Validation

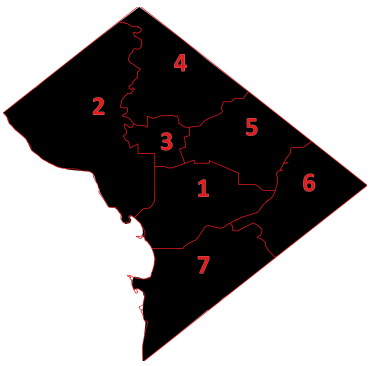
"I'm staring at the asphalt wondering what's buried underneath."

-The Postal Service "The District Sleeps Alone Tonight"

Next, I transformed the data frame into a DMatrix, the format most optimized to XGBoost, to see if I could squeeze any more optimization out of the training data, and trained the model on this DMatrix. Through the Seaborn module, I was able to plot the most relevant features in my model:



I also plotted the first three decision trees from my model to assess how different features were affecting a prediction score. Street length, unsurprisingly, was the best indicator of a street having experienced a homicide, and appeared to positively correlate from the sample trees below, which makes sense since more area would equal more risk. 2015 median FAGI was the next best indicator, probably negatively correlated judging from the sample trees, a reflection that street segments in poorer areas are at much higher risk of homicide. Interestingly, median FAGI change was also a significant indicator, with the trees suggesting that gentrification or rises in median FAGI may actually reduce incidence of crime. Population and housing density were also strong indicators, likely negatively correlated according to the sample trees, which makes sense as more crowded conditions and more people could discourage violent crime. Locational data like liquor stores, schools, and metro entrances appeared more mixed in their effects. Please see the appendix for the decision trees.



DC Police Districts

I ran my final model, an XGBoost classifier with a learning rate of 0.5, a seed of 0, a subsample rate of 0.7, a column sample by tree of 0.8, a maximum depth of 7, and a minimum child weight of 1 as well as a binary logistic objective, against my final test data of 30% reserved streets, giving me an array of probabilities for each street as to whether it likely experienced a homicide or not. I transformed this probability, classifying anything at or above 50% probability into "at risk" and anything below as "not at risk" in one-hot encoding, and ran a final F1 test between the data set's actual values and its predicted values. My F1 was 0.123, a decent score. As XGBoost as a model automatically averages many decision trees in compiling its final model, the model is inherently very robust and should not change significantly with minor adjustments, for example, if we received 2015 median FAGI numbers for census tract 62.02.

Justification

My final F1 score on my reserved test data yielded an 824% improvement on my random predictor. This is an excellent improvement and a strong case for machine learning as a viable tool for predicting crime incidence where patterns and contributing factors to crime are not easily discerned, and for drawing out those contributing factors for policy targeting in an effort to reduce crime incidence. However, as my model's F1 score was still only 0.123 on the final testing data, it best serves merely as a guide and should not completely replace other methods of predicting crime incidence.

5. Conclusion

Free-Form Visualization

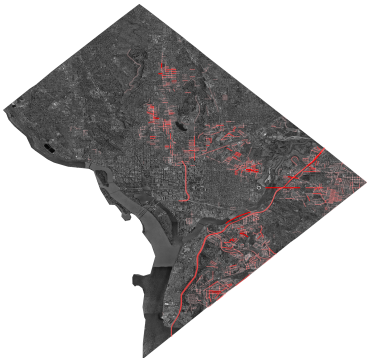
"Our story is about local Washington, a city where bad things happen in good neighborhoods and terrifying things can happen in poor neighborhoods."

-Harry Jaffe and Tom Sherwood, "Dream City: Race, Power, and the Decline/(Revival?) of Washington D.C."

For my visualization (please see the index), I transformed my original featured data frame into a DMatrix, and ran my trained model on this data frame to get an array of predictions. After converting this to a list, I appended the list as a "Predictions" column onto my original data frame and ran some diagnostic tests to check that the predictions were mapped to the correct rows. Once this was checked, I exported my data frame as a csv and joined it with my street segments shapefile in GIS for two maps, one comparing homicide locations and one comparing a homicide heat map to the streets my model predicted as high risk. While the point map shows locations of each homicide, the heat map better represents areas that experienced multiple homicides in the same location.

We can see from my visualizations that my predictions line up fairly nicely with the location of actual homicides, especially on the heat map, which is very interesting considering the heat map indicates the number of homicides in a location, an attribute masked from my trained model by one-hot encoding homicide locations. One notable failure was an area in Navy Yard along the Anacostia River. It's dark green indicating multiple murders in one location, but none of the streets in the area seem to pass the 0.2 predictor mark. Another shortcoming is that long streets are coded entirely in one color, even if only a small portion intersects an area where a homicide occurred.

Predicting DC Homicides



Homicide Prediction Scores for Street Segments (Higher Scores are Darker Red)

Reflection

Whew! What a long way we've come! From downloading shapefiles from dc.gov and a crime dataset from Kaggle, to learning QGIS in order to merge the data, onward to spending hours trying to figure out why my computer wouldn't install XGBoost (hint: it had to do with Macs using clang instead of another gcc compiler), and finally teaching myself XGBoost, it's been an adventure, and I'm happy with the result.

Overall, I believe my XGBoost classifier model did well on very limited data and without considering the time-bounded elements of change in a rapidly changing city. The model showed that lower-income areas with more vacant housing and lower population density and which have experienced the least change in median FAGI are at the highest risk of homicides. While gentrification can be problematic for its own reasons, its economic benefits appear to correlate with reduced risk of homicide, and these indicators need not only occur through gentrification. DC might do well to promote economic revitalization in poorer, at-risk neighborhoods while increasing programs and protections that keep existing residents in their homes. By bringing economic opportunity and raising incomes in at-risk neighborhoods while preventing evictions and other forces leading to vacant housing and low population densities, DC might combat its relatively high homicide rate in a more sustainable and arguably ethical way.

Improvement

This project used limited data drawn from opendata.dc.gov.[10] Incorporating more features and tracking these features' changes over time, as well as including each homicide's timestamp, could give us a model that not only predicts where homicides might occur, but when. Alternatively, total homicide counts per street could be imported into the model and run through an XGBoost regression to improve the precision and recall of the model's scoring, to predict how many homicides are likely to have occurred in each location. These improvements likely would require more powerful processors, but with enough resources, the model can increase its usefulness and the insights into alternative policy solutions to combat homicides.

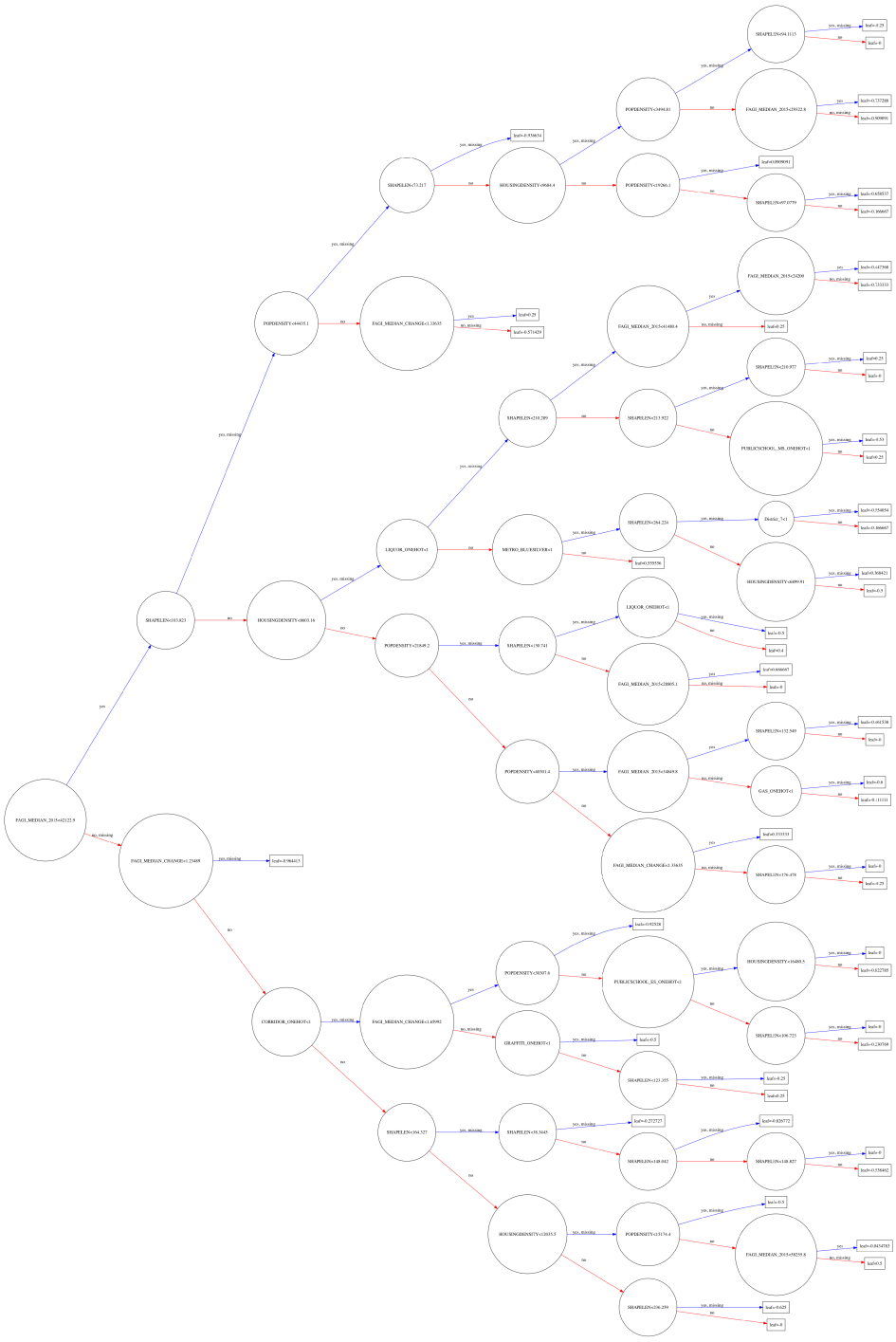
[10] http://opendata.dc.gov/datasets

Appendix

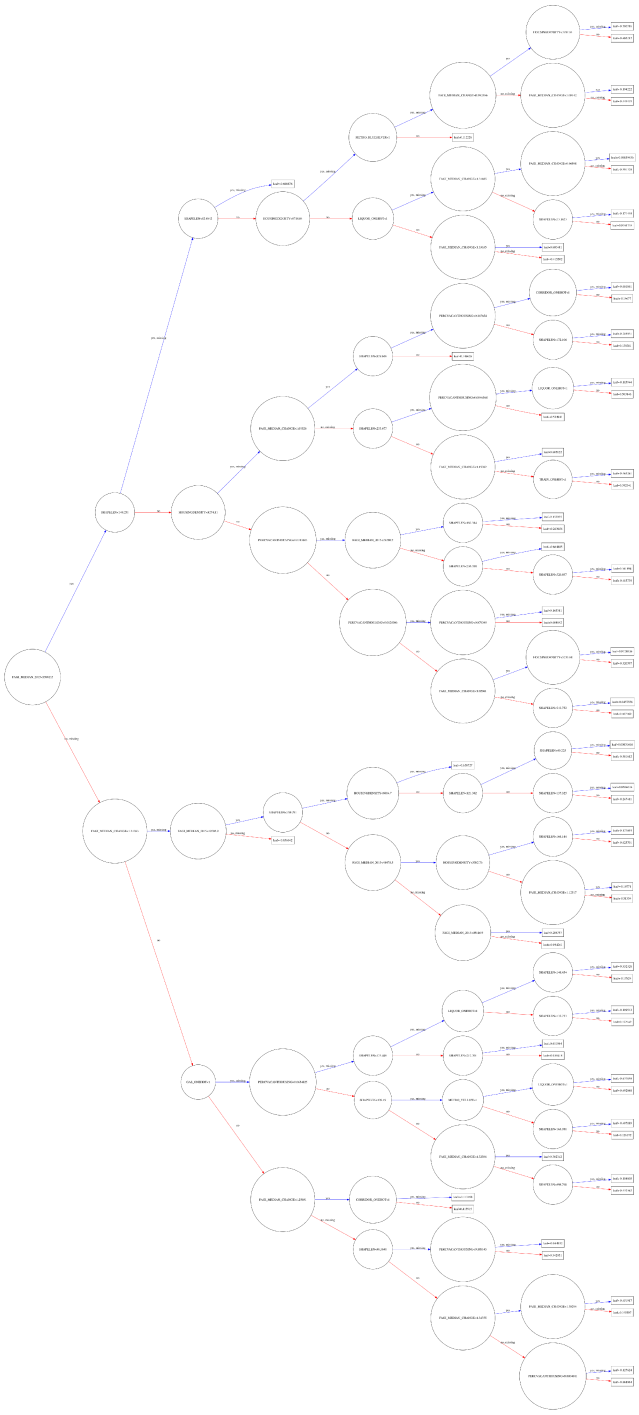
Predicting DC Homicides

Correlation Table	SHAPELEN	District_1	District_2	District_3	District_4	District_5	District_6	District_7	METRO_ BLUESILVER	METRO_ ORANGE	...	PUBLIC SCHOOL_ MS_ONEHOT	PUBLIC SCHOOL_ HS_ONEHOT	FEDERAL_ ONEHOT	CORRIDOR_ ONEHOT	FAGI_MEDIAN_ 2015	FAGI_MEDIAN_ CHANGE	POPDENSITY	HOUSING DENSITY	PERCVACANT HOUSING	HOMICIDES_ ONEHOT
SHAPELEN	1.000000	0.002208	0.031514	-0.024746	-0.020627	-0.019289	-0.003570	0.060920	0.003477	0.023830	...	0.015753	0.015639	0.105853	-0.009684	-0.000795	-0.034265	-0.077189	-0.067644	-0.024327	0.098937
FEDERAL_ONEHOT	0.105853	0.259621	-0.004556	0.011090	-0.095253	-0.084560	-0.067994	0.006598	0.115298	0.113749	...	0.039679	0.001633	1.000000	-0.010897	0.016118	0.030165	-0.005266	0.034406	0.055059	-0.036288
HOMICIDES_ONEHOT	0.098937	-0.055526	-0.141288	0.030325	-0.021895	0.053155	0.098285	0.100296	-0.010541	-0.013819	...	0.000244	0.014589	-0.036288	0.070341	-0.189608	-0.023814	0.095226	0.057608	0.111659	1.000000
District_7	0.060920	-0.141325	-0.200019	-0.098128	-0.169223	-0.139846	-0.076734	1.000000	-0.049163	-0.050100	...	-0.005521	-0.022282	0.006598	-0.088675	-0.307374	-0.154546	-0.104802	-0.122215	0.317320	0.100296
METRO_GREEN	0.055714	0.111240	-0.078111	0.165250	-0.025705	-0.054613	-0.055283	-0.016171	0.099594	0.097092	...	0.047616	-0.000544	0.044632	0.110558	0.014568	0.076084	0.045250	0.059078	0.084472	0.035737
District_2	0.031514	-0.225552	1.000000	-0.110365	-0.272642	-0.228889	-0.231698	-0.200019	0.002022	-0.000996	...	-0.035423	-0.006016	-0.004556	-0.059890	0.732923	0.027404	-0.112199	-0.057603	-0.343319	-0.141288
METRO_ORANGE	0.023830	0.207347	-0.000996	-0.038226	-0.069375	-0.057332	-0.011598	-0.050100	0.901924	1.000000	...	-0.018333	-0.001981	0.113749	-0.003991	0.024757	0.015359	-0.020054	0.007091	0.042324	-0.013819
GAS_ONEHOT	0.016464	-0.040785	-0.050667	0.056623	0.018327	0.115368	0.013011	-0.080040	0.009245	-0.008447	...	-0.004795	-0.030211	-0.018959	0.229318	-0.069596	0.049738	0.097735	0.086936	0.046928	0.083668
PUBLICSCHOOL_ES_ONEHOT	0.016360	0.047515	-0.081796	0.077485	-0.002526	-0.058032	0.035998	0.040618	-0.033283	-0.040010	...	0.022958	0.023484	-0.029743	0.056967	-0.047370	-0.022955	0.105470	0.099941	0.010919	0.035795
PUBLICSCHOOL_MS_ONEHOT	0.015753	0.016658	-0.035423	0.073592	-0.007065	0.001096	-0.003332	-0.005521	-0.017990	-0.018333	...	1.000000	0.200717	0.039679	0.030793	-0.012013	-0.010032	0.064851	0.059520	-0.004412	0.000244
PUBLICSCHOOL_HS_ONEHOT	0.015639	-0.019984	-0.006016	0.054767	-0.000460	0.010622	0.013380	-0.022282	-0.015785	-0.001981	...	0.200717	1.000000	0.001633	0.057200	-0.034256	0.043890	0.080646	0.044001	0.023191	0.014589
METRO_RED	0.007632	0.088659	0.083541	-0.038837	-0.039566	-0.006419	-0.061495	-0.053087	0.146680	0.143251	...	-0.019426	-0.003701	0.054330	0.050810	0.081308	0.051952	-0.021107	0.027740	0.073959	-0.023327
METRO_BLUESILVER	0.003477	0.213330	0.002022	-0.037436	-0.068077	-0.056259	-0.024915	-0.049163	1.000000	0.901924	...	-0.017990	-0.015785	0.115298	-0.018835	0.030709	0.017124	-0.013054	0.014602	0.053648	-0.010541
District_1	0.002208	1.000000	-0.225552	-0.090862	-0.199390	-0.099691	-0.165604	-0.141325	0.213330	0.207347	...	0.016658	-0.019984	0.259621	-0.019315	0.086844	0.109955	0.026415	0.077456	0.105144	-0.055526
GRAFFITI_ONEHOT	0.000509	-0.026979	0.040262	0.377590	0.055901	-0.003125	-0.163519	-0.153392	0.003362	0.006557	...	0.051099	0.056087	-0.012131	0.262335	-0.001416	0.245289	0.427312	0.417913	0.030756	0.035273
METRO_YELLOW	-0.000002	0.081314	-0.069735	0.192964	-0.013936	-0.048756	-0.049355	-0.042607	0.115428	0.112717	...	0.057035	0.002813	0.037804	0.126663	0.024667	0.105374	0.071791	0.079770	0.038135	0.027197
FAGI_MEDIAN_2015	-0.000795	0.086844	0.732923	-0.017347	-0.180269	-0.189393	-0.337269	-0.307374	0.030709	0.024757	...	-0.012013	-0.034256	0.016118	-0.064206	1.000000	0.203317	-0.160619	-0.083631	-0.406613	-0.189608
TRAIN_ONEHOT	-0.000880	-0.052843	-0.149112	-0.073154	0.139960	0.319260	-0.104671	-0.091104	-0.036650	-0.037349	...	-0.028572	-0.023840	-0.078986	-0.068180	-0.108801	0.060716	-0.146202	-0.136944	-0.001288	-0.002052
District_6	-0.003570	-0.165604	-0.231698	-0.113670	-0.196025	-0.160775	1.000000	-0.076734	-0.024915	-0.011598	...	-0.003332	0.013380	-0.067994	-0.030449	-0.337269	-0.409520	-0.099419	-0.102801	0.131685	0.098285
LIQUOR_ONEHOT	-0.008325	0.023259	-0.037184	0.177402	-0.041537	0.083917	-0.032885	-0.098960	0.059324	0.050455	...	0.015745	-0.000162	-0.039670	0.316277	-0.055917	0.130246	0.235181	0.244463	0.074670	0.101415
CORRIDOR_ONEHOT	-0.009684	-0.019315	-0.059890	0.250963	-0.022599	0.074377	-0.030449	-0.088675	-0.018835	-0.003991	...	0.030793	0.057200	-0.010897	1.000000	-0.064206	0.133552	0.251524	0.238002	0.077261	0.070341
District_5	-0.019289	-0.099691	-0.228889	-0.058461	-0.144945	1.000000	-0.160775	-0.139846	-0.056259	-0.057332	...	0.001096	0.010622	-0.084560	0.074377	-0.189393	0.202377	-0.080863	-0.093299	0.119193	0.053155
District_4	-0.020627	-0.199390	-0.272642	-0.080657	1.000000	-0.144945	-0.196025	-0.169223	-0.068077	-0.069375	...	-0.007065	-0.000460	-0.095253	-0.022599	-0.180269	0.022048	0.056852	-0.026546	-0.190897	-0.021895
PERCVACANTHOUSING	-0.024327	0.105144	-0.343319	0.045846	-0.190897	0.119193	0.131685	0.317320	0.053648	0.042324	...	-0.004412	0.023191	0.055059	0.077261	-0.406613	0.176864	0.005673	0.040365	1.000000	0.111659
District_3	-0.024746	-0.090862	-0.110365	1.000000	-0.080657	-0.058461	-0.113670	-0.098128	-0.037436	-0.038226	...	0.073592	0.054767	0.011090	0.250963	-0.017347	0.311576	0.532573	0.548981	0.045846	0.030325
FAGI_MEDIAN_CHANGE	-0.034265	0.109955	0.027404	0.311576	0.022048	0.202377	-0.409520	-0.154546	0.017124	0.015359	...	-0.010032	0.043890	0.030165	0.133552	0.203317	1.000000	0.230022	0.235533	0.176864	-0.023814
HOUSINGDENSITY	-0.067644	0.077456	-0.057603	0.548981	-0.026546	-0.093299	-0.102801	-0.122215	0.014602	0.007091	...	0.059520	0.044001	0.034406	0.238002	-0.083631	0.235533	0.946292	1.000000	0.040365	0.057608
POPDENSITY	-0.077189	0.026415	-0.112199	0.532573	0.056852	-0.080863	-0.099419	-0.104802	-0.013054	-0.020054	...	0.064851	0.080646	-0.005266	0.251524	-0.160619	0.230022	1.000000	0.946292	0.005673	0.095226

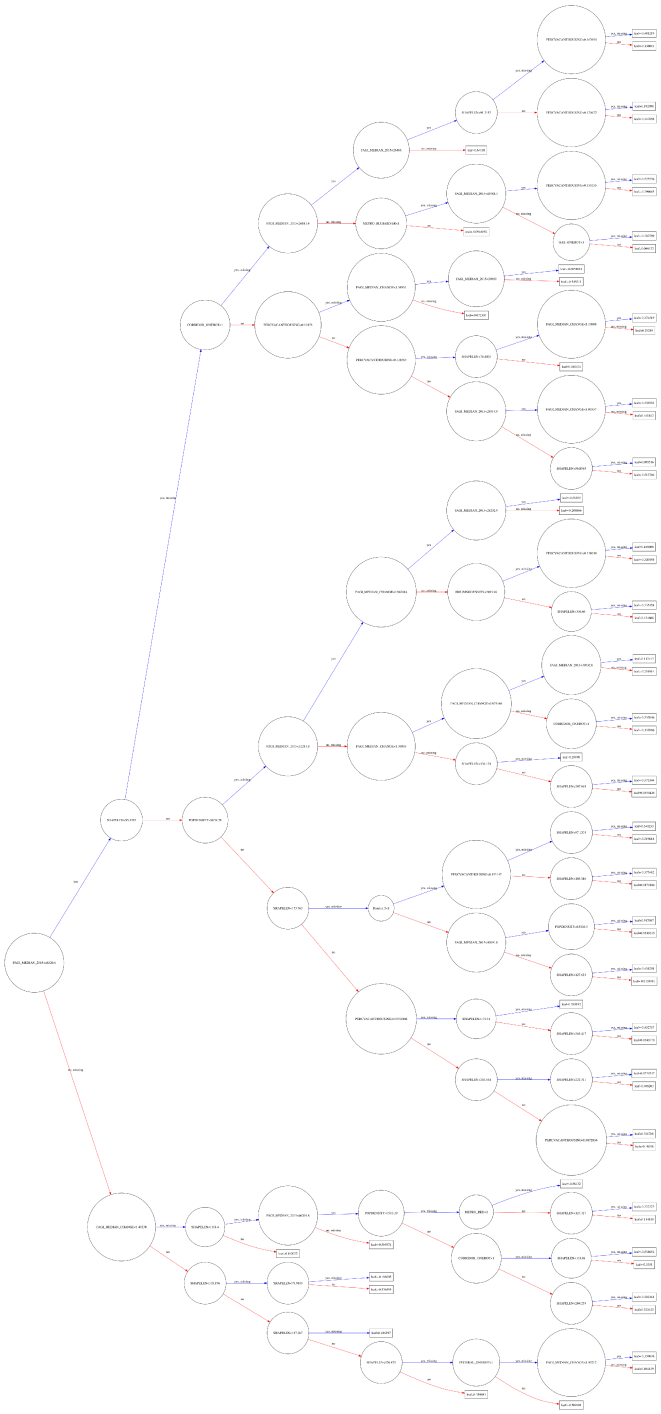
Predicting DC Homicides



Predicting DC Homicides

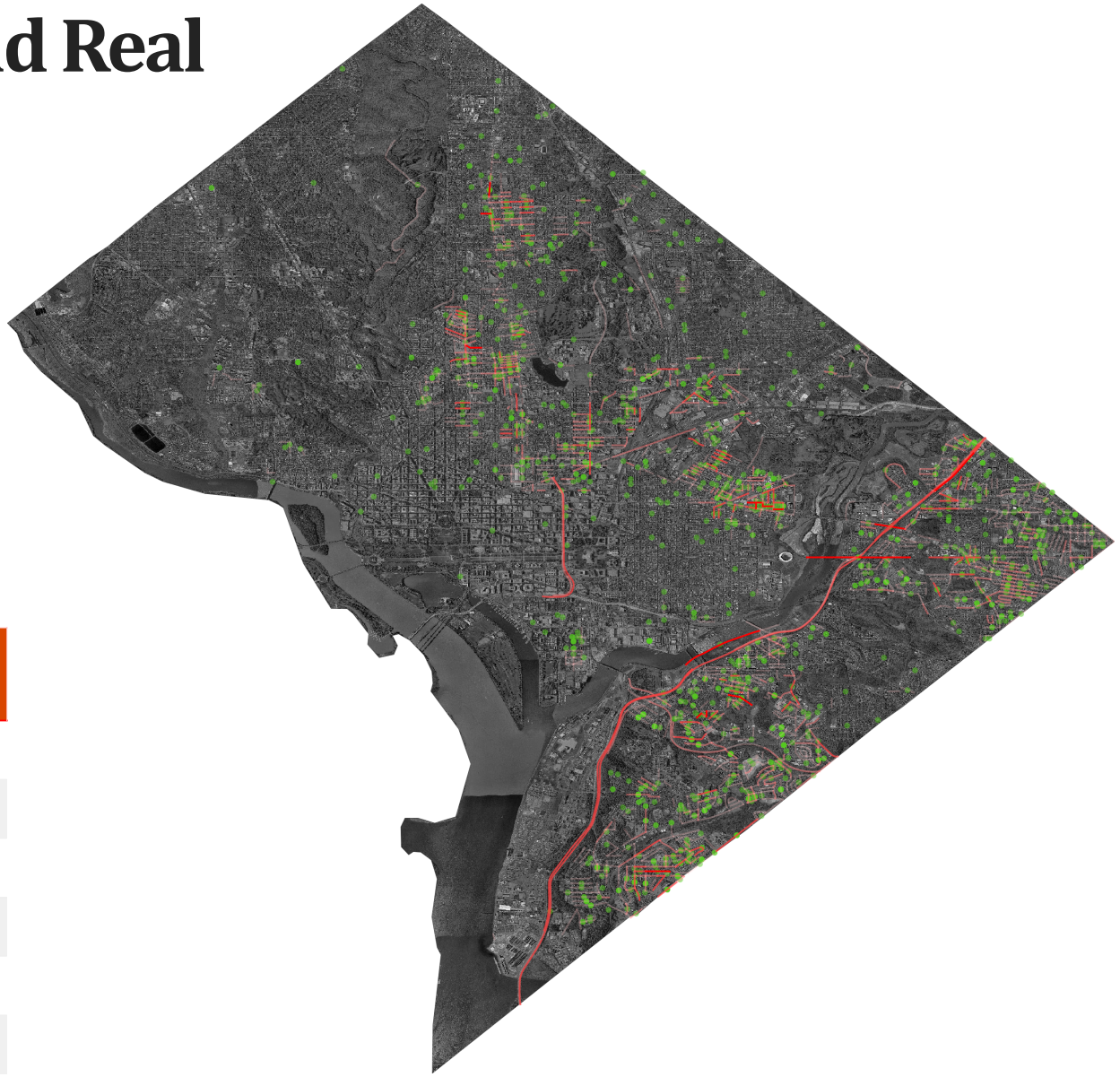


Predicting DC Homicides



Predicted and Real Homicide Incidence

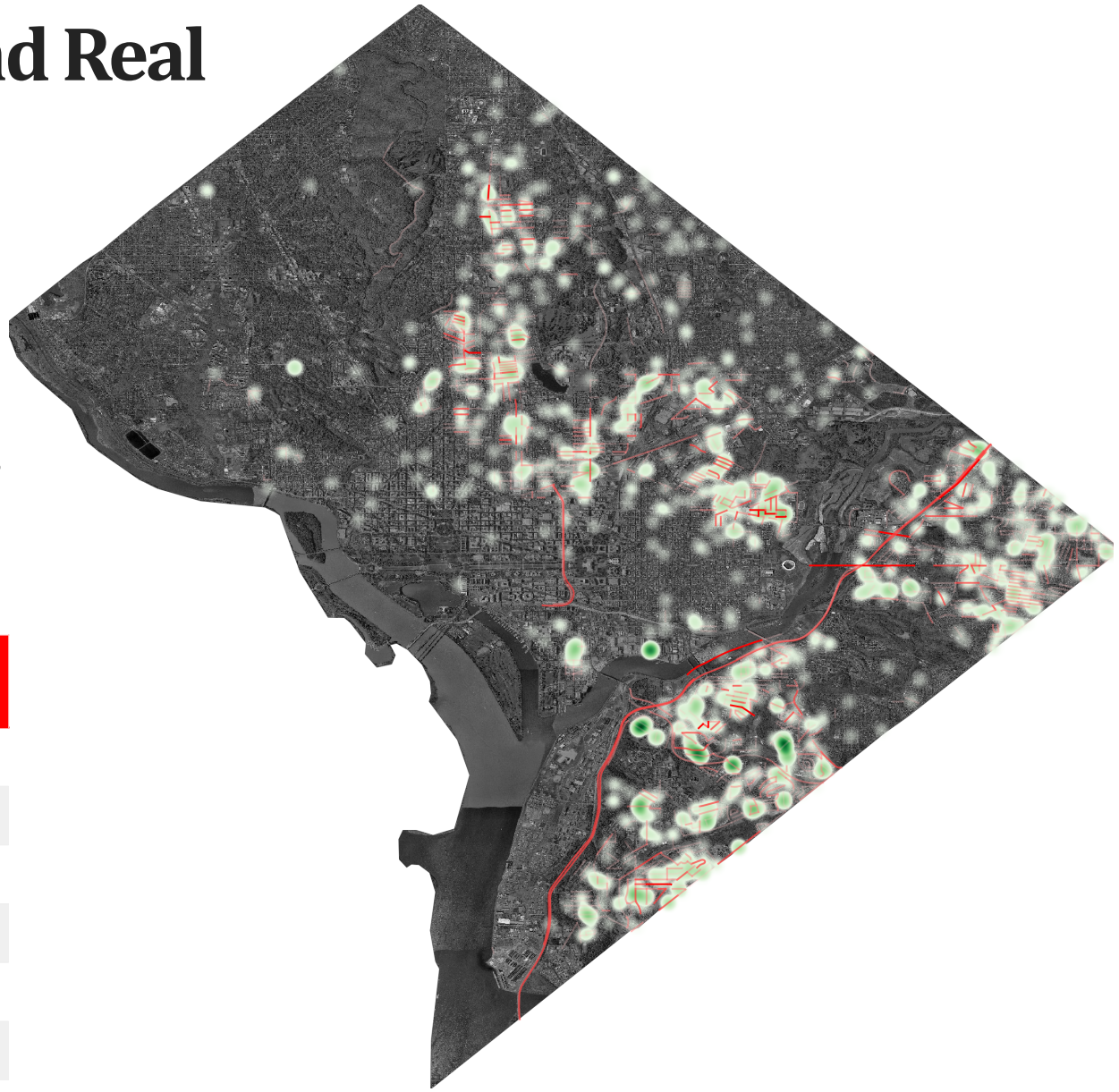
Points and Street Predictions



Legend
Street Score 0.0 - 0.2
Street Score 0.2 - 0.4
Street Score 0.4 - 0.6
Street Score 0.6 - 0.8
Street Score 0.8 - 1.0
Recorded Homicide

Predicted and Real Homicide Incidence

Heat Map and
Street Predictions



Legend
Street Score 0.0 - 0.2
Street Score 0.2 - 0.4
Street Score 0.4 - 0.6
Street Score 0.6 - 0.8
Street Score 0.8 - 1.0
Homicide Incidence