Predicting Crime in DC
Debbie Yu
General Assembly – DAT7
August 2015

Problem Statement and Hypothesis

While crime rates in DC have steadily decreased over the past 20 years, DC still struggles with relatively high crime rates like most major US cities. I am interested in predicting the likelihood/probability of violent crimes occurring in DC, with a focus on location and time of year in DC. So, the question for my project is:

Given a particular observation of crime, can I predict whether or not the crime was of violent or nonviolent nature?

Studies have shown correlations between violent crime and socioeconomics in neighborhoods, weather, and the time of year, so I am interested in building a predictive model about crime in DC that can potentially inform DC Government policy or safety program implications.

Description of the Data

I used three primary datasets for this project:

Crime Data
Like many other major US cities, DC Government has an online open data portal (http://opendata.dc.gov). This portal currently hosts 634 different datasets across a wide range of topics and date ranges (e.g. demographic information from 1990, planning land use and zoning metrics from 2006, and DC Government purchase orders from 2004). The District currently has several datasets of all crime incidents in the district from 2012-2014, as well as a crime report that will have crime incidents that have occurred since the last 30 days of the pull date of the report. For the purposes of my project, I chose to use the DC Crime Incident report from 2014.

This report contained the following 21 columns

1. ccn
2. reportdatetime
3. policeshift
4. offense
5. method
6. lastmodifieddate
7. blocksiteaddress
8. blockxcoord
9. blockycoord
10. ward
11. anc
12. district
13. psa
14. neighborhoodcluster
15. businessimprovementdistrict
16. block_group
17. census_tract
18. voting_precinct
19. start_date
20. end_date
21. esri_oid

The dataset had 38,388 rows, with each row representing a single crime incident in 2014.

Crime Data – Data Cleaning Process
While the crime dataset was relatively complete, there was some basic cleaning required with regard to the types of data in each data field. Date and time are a particular point of interest in my analysis, so I wanted to make sure I could utilize the pandas datetime object type. I fortunately had the date and time that a crime incident was logged, in a string format that had the month, date, year, hours, and minutes. Using this field, called 'reportdatetime', I created a separate field with the date only, and another field that had the hour. I found that slicing the reportdatetime string for the date and converting that string to a datetime object worked well. Then I converted the reportdatetime string to a datetime object to create an extra field that only contained the hour. I think created a few subgroupings of the days and hours – one binary classification for 'weekend' and a binary classification for daytime/nighttime.

I also created a categorization of non-violent and violent crimes. To simplify my model, I though it would be best to focus on these two broader categories than creating predictions for each of the 9 crime types listed. I categorized the crimes as such in an additional column called 'violent':

Violent crimes (mapped to a 1): sexual abuse, homicide, assault with a dangerous weapon, robbert
Non-violent crimes (mapped to a 0): arson, burglary, theft/other, theft from an automobile, motor vehicle theft

Neighborhood Data
I also wanted to gather information about specific neighborhoods in DC, so that my analysis could focus on a geographic granularity that would be relatable to the average citizen. I discovered that DC Government collects data by 'neighborhood cluster,' which are groupings of 3-5 neighborhoods throughout the city. (Single neighborhoods are not used as a geographic unit by because there are no official neighborhood boundaries determined by DC government; individual neighborhood boundaries shift rapidly and the exact locations are often under debate.) This information is collected for public consumption at http://neighborhoodinfodc.org. The clusters and associated neighborhoods are shown below:

Cluster 1: Kalorama Heights, Adams Morgan, Lanier Heights
Cluster 2: Columbia Heights, Mt. Pleasant, Pleasant Plains, Park View
Cluster 3: Howard University, Le Droit Park, Cardozo/Shaw
Cluster 4: Georgetown, Burleith/Hillandale
Cluster 5: West End, Foggy Bottom, GWU
Cluster 6: Dupont Circle, Connecticut Avenue/K Street
Cluster 7: Shaw, Logan Circle
Cluster 8: Downtown, Chinatown, Penn Quarters, Mount Vernon Square, North Capitol Street
Cluster 9: Southwest Employment Area, Southwest/Waterfront, Fort McNair, Buzzard Point
Cluster 10: Hawthorne, Barnaby Woods, Chevy Chase
Cluster 11: Friendship Heights, American University Park, Tenleytown
Cluster 12: North Cleveland Park, Forest Hills, Van Ness
Cluster 13: Spring Valley, Palisades, Wesley Heights, Foxhall Crescent, Foxhall Village, Georgetown Reservoir
Cluster 14: Cathedral Heights, McLean Gardens, Glover Park
Cluster 15: Cleveland Park, Woodley Park, Massachusetts Avenue Heights, Woodland-Normanstone Terrace
Cluster 16: Colonial Village, Shepherd Park, North Portal Estates
Cluster 17: Takoma, Brightwood, Manor Park
Cluster 18: Brightwood Park, Crestwood, Petworth

Cluster 19: Lamond Riggs, Queens Chapel, Fort Totten, Pleasant Hill
Cluster 20: North Michigan Park, Michigan Park, University Heights
Cluster 21: Edgewood, Bloomingdale, Truxton Circle, Eckington
Cluster 22: Brookland, Brentwood, Langdon
Cluster 23: Ivy City, Arboretum, Trinidad, Carver Langston
Cluster 24: Woodridge, Fort Lincoln, Gateway
Cluster 25: NoMa, Union Station, Stanton Park, Kingman Park
Cluster 26: Capitol Hill, Lincoln Park
Cluster 27: Near Southeast, Navy Yard
Cluster 28: Historic Anacostia
Cluster 29: Eastland Gardens, Kenilworth
Cluster 30: Mayfair, Hillbrook, Mahaning Heights
Cluster 31: Deanwood, Burrville, Grant Park, Lincoln Heights, Fairmont Heights
Cluster 32: River Terrace, Benning, Greenway, Fort Dupont
Cluster 33: Capitol View, Marshall Heights, Benning Heights
Cluster 34: Twining, Fairlawn, Randle Highlands, Penn Branch, Fort Davis Park, Dupont Park
Cluster 35: Fairfax Village, Naylor Gardens, Hillcrest, Summit Park
Cluster 36: Woodland/Fort Stanton, Garfield Heights, Knox Hill
Cluster 37: Sheridan, Barry Farm, Buena Vista
Cluster 38: Douglass, Shipley Terrace
Cluster 39: Congress Heights, Bellevue, Washington Highlands

This website contains a variety of socioeconomic data points for each neighborhood cluster, collected from the US Census, American Community Survey, and DC government data sources. From this website, I decided to collect the following datapoints for each neighborhood cluster using a webscraper:

1. Population, 2010
2. %black, non-hispanic 2010
3. %hispanic, 2010
4. %white, non-hispanic 2010
5. %asian, PI non-Hispanic 2010
6. Poverty rate from 2008-2012
7. Unemployment rate from 2008-2012
8. % population 16 years old and older that is employed, from 2008-2012
9. % population of persons without a high school diploma, from 2008-2012
10. Average family income, 2008-2012
11. Persons receiving food stamps, 2014
12. Persons receiving TANF, 2014
13. Median sales price of single-family homes, 2013

Neighborhood Data – Data Cleaning Process
With regard to data cleaning on my own scraped data, I needed to do some column formatting and data type conversions. Certain numbers were represented as strings with commas after I scraped them, so I converted them to integers and floats by removing the commas from the strings an converting them to integers.

I had mostly complete data for all of the data fields I scraped with the exception of median home prices for 4 clusters. The data was not available for the median home prices in 2013. I used a human learning technique here by either using the median home price of the average or neighboring cluster. I realize that this technique is not scalable for a large dataset because I hard-coded these numbers, but thought it would be ok for a small dataset. In future analyses with much larger datasets, I would likely not take this approach.

Weather Data

Finally, I obtained weather data from the National Oceanic and Atmospheric Administration. I filled out an online request form and requested weather data for the District in 2014. I received a

csv file with over 91 fields, many of which provided highly detailed information (e.g. number of days included in the multiday precipitation total) that I did not need.

Weather Data – Data Cleaning Process
With so much weather data, I needed to significantly pare down this dataset to the small number of features I wanted to use. These features were:

1. Date
2. Temperature Min
3. Temperature Max
4. Snow
5. Precipitation

The temperature fields were provided in Celcius in tenths of degrees, so I needed to translate these to Fahrenheit and convert the data field to floats. I also used the max and min temps to create a new field, which was average temp in Fahrenheit.

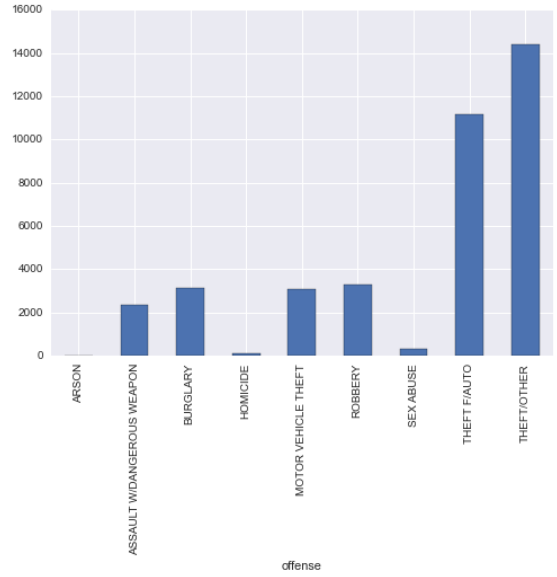
I encountered an interesting phenomenon when I tried to convert the date to a pandas datetime object. Since the date was represented as a string and did not include hour/minute/second info, converting the date without formatting to a datetime object mis-coded the values, where the year was 1970 for all entries, and the year, month, and day were represented as a string in the middle of the field. I had to reformat this field by converting it back to a string, slicing the date and time out, and then converting it back to a datetime object.

I also created a new data series using the pandas datetimeindex object 'dayofweek', which indicates what particular day of the week a date falls on. With dayofweek, Mondays are represented with 0 and Sundays are represented with 6. For example, dayofweek called on 1/1/2014 returns a value of "2", which is a Wednesday.

Taking these 3 datasets, I merged them into one large dataset that would have the crime incidents, weather on the day the crime was reported, and socioeconomic data with regard to the neighborhood that the crime was reported. This was saved as one large .csv file I could read into my analysis and visualization file

Descriptive and Exploratory Analyses and Visuals

I first took a look at the distribution of types of crimes in DC by total counts:



Looking at the distribution of types of offenses, it was interesting to know that a huge proportion of crimes committed in DC were of the non-violent type. Using this information I downsampled my data for the models so that there were approximately the same # of observations for both violent and nonviolent crimes.

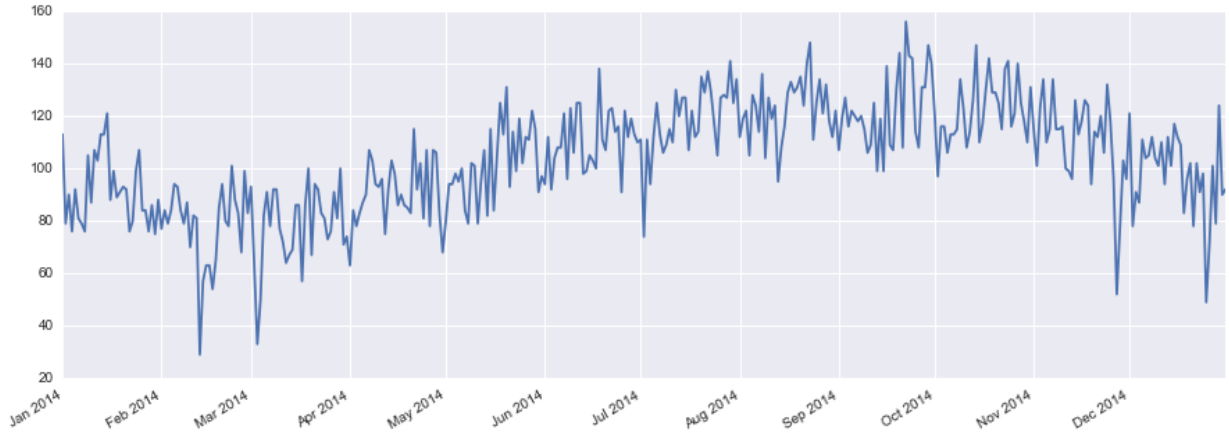
Violent vs Non-violent Crime Counts

	Count
Violent (assault w/dangerous weapon, sex abuse, homicide, robbery)	6063
Non-violent (arson, burglary, theft/other, theft f/auto, motor vehicle theft)	31,806
TOTAL	37,869

Time elements

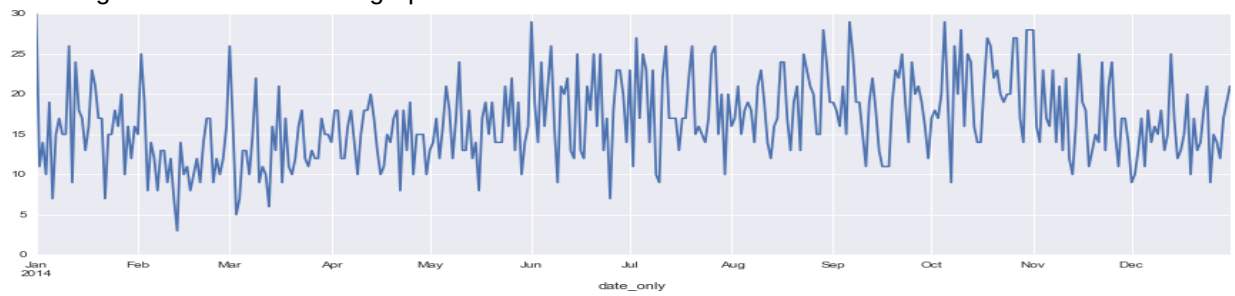
I then wanted to see if certain time elements would be compelling features. The first visualization shows the total count of crimes for each day of 2014:

OBJ



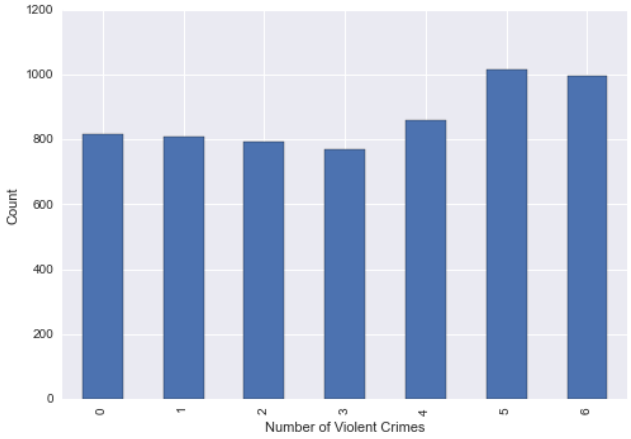
Also, no surprise here that crime is generally lower in the colder months (Dec-March), and climbs as the weather gets warmer. It appears that overall crime peaks in September/October, before slowing down again as Fall turns to winter.

Here is a specific count of violent crimes for each day in 2014. The trend for violent crimes isn't as strong as it is for the overall graph above.



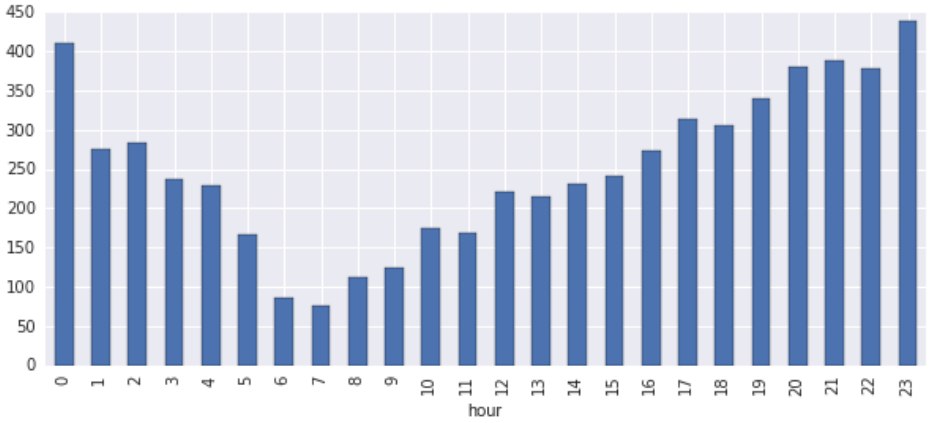
Violent crimes by day of the week:

Monday = 0
Tuesday = 1
Wednesday = 2
Thursday = 3
Friday = 4
Saturday = 5
Sunday = 6

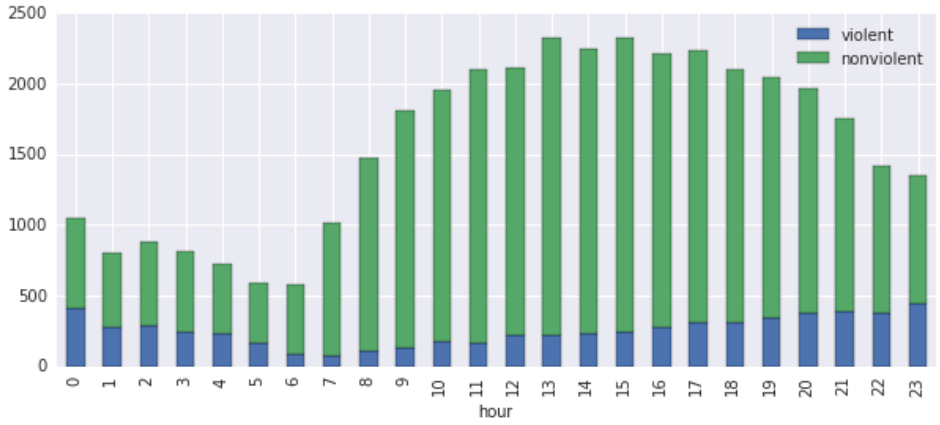


I was actually a bit surprised that there was little variation of crimes committed between the different weekdays, and the difference between the weekdays and weekends is not as large as I would have expected.

Violent crimes by hour

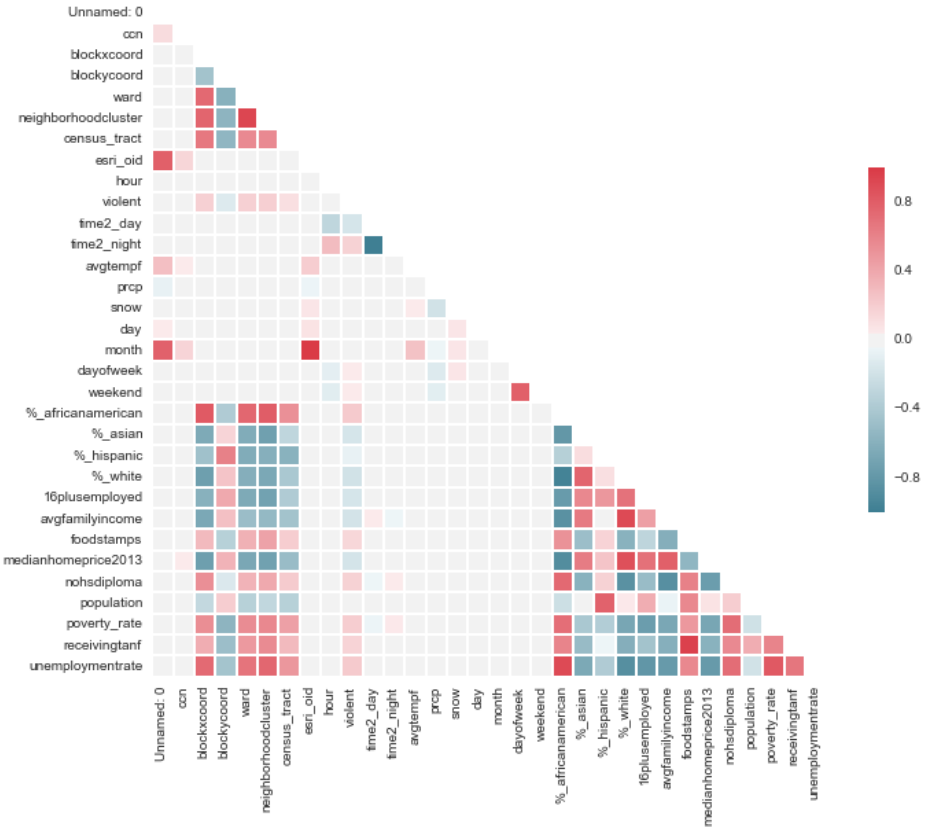


Violent & Nonviolent crimes by time of day (hr):



I think I was most surprised by the variation in the hour that crimes are committed. It makes sense intuitively that violent crimes aren't committed very often in the early hours of the morning like 6 or 7AM. Conversely while it makes sense that the most crimes are committed at night, there seems to be such a large drop between midnight and 1AM, when this doesn't seem like something that would matter as much. This may speak to the fact that I am using the date and time of which a crime is reported, but this may not be an accurate representation of when the crimes were actually committed.

Modeling and Model Evaluation

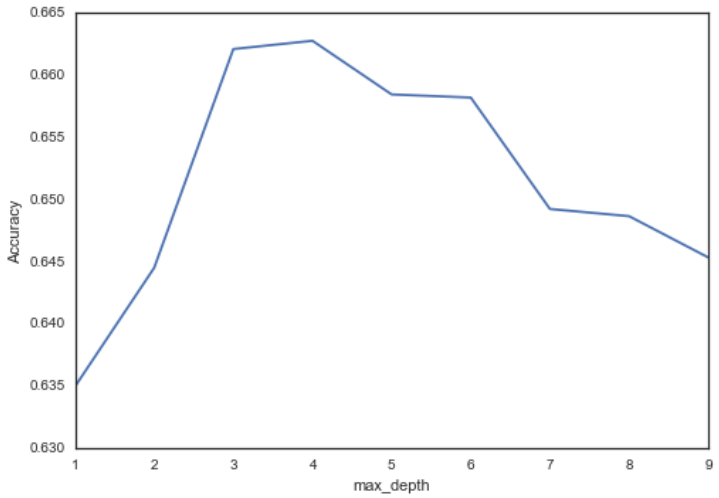


I have a lot of data fields that I didn't actually plan on using in my model but thought they would be interesting to visualize. A lot of my features are (not surprisingly), correlated, particularly the

ones related to socioeconomic measures (e.g. 'poverty rate' and 'receiving tanf'). Thus, I will avoid using these variables but try to stick to one variable that might represent some general measure (e.g. socioeconomics of a neighborhood).

First model type: Decision Trees

I thought I would first use Decision Trees to determine what some of the most useful features were. I was really interested in seeing how the model would handle interactions between variables that were correlated, (e.g. poverty rate and avg family income, for example). Using the Decision Tree Classifier, I looped through max depth ranges 1-9, with a max depth of 5. I found that the max depth with the highest accuracy score was a depth of 4.



I then ran a Decision Tree with a max depth of 4 to look at the feature importance:

Feature	Importance
Avgfamilyincome	0.592477
Time2_night (day vs night)	0.342999
Weekend (binary classification)	0.019776
Census_tract	0.018494
Unemploymentrate	0.011607
Medianhomeprice2013	0.005980
Prcp	0.0000
Month	0.0000
Poverty rate	0.0000
% of population over 16 employed	0.0000
% of population with no high school diploma	0.0000

This decision tree had nodes with gini indexes that ranged from 0.000 (one node with 7 samples), and 0.4999. Since most of my nodes did not have very pure 'leaves', this is consistent with my highest possible accuracy score with this tree being just above 0.660.

I saw a lot of variability as I used different features within different decision trees, which is indicate of the model type – which is that it has a tendency to have high variance, depending on the data.

I came to the realization that my feature for neighborhood cluster should have been a dummy variable – as the neighborhood clusters were represented by numbers, from 1-39, and yet there

was no ordinal importance of those numbers. I then saw that census tract was coming up as an important feature, and while census tract was also represented as a range of numbers (from 1 to 111) or so, I saw that the numbers corresponded loosely to geographic areas. This mean that tracts from 70 and upwards were tracts that are actually located in the Anacostia area, and tracts with smaller numbers are located in the wealthier parts of Washington DC in the northwest. While this is convenient, it isn't the best use of a feature that should really be turned into dummy variables.

Second model type: Random Forests

Using the same features that I used for the above Decision Tree model, I then used the random forest classifier. Using 150 estimaters and the highest possible # of features being 5, by out of bag score was 0.617. I was surprised that this model did not perform any better than by Decision Tree model. A comparison of the feature importance is below:

Feature	Decision Tree Importance	Random Forest Importance
Avgfamilyincome	0.592477	0.065141
Time2_night (day vs night)	0.342999	0.064250
Weekend (binary classification)	0.019776	0.045719
Census_tract	0.018494	0.249930
Unemploymentrate	0.011607	0.028633
Medianhomeprice2013	0.005980	0.044259
Prcp	0.0000	0.172188
Month	0.0000	0.279835
Poverty rate	0.0000	0.017257
% of population over 16 employed	0.0000	0.012498
% of population with no high school diploma	0.0000	0.020291

Some of the features overlapped, but I was surprised to find that random forests chose a couple of other features that were otherwise ignored by Decision Trees.

Third model type: Logistic Regression

Using the features that were deemed important by both the Decision Tree and Random Forest models, I plugged those features into a couple logistic regression models

	Train, test, split accuracy score	Cross validation score (mean)	AUC
DT Features model	0.6353	0.6258	0.6831
Random Forest features model	0.6171	0.615	0.6527

The two models had virtually the same ROC curves, since their AUC scores were only different be 0.03.

Key Takeaways/Next Steps

While my models were better than the null value of 0.5, they didn't perform as well as I had hoped. I think this has to do with the data that I had for my features, and allotting more time for feature engineering and thinking more critically about the usefulness of the features. I gathered a lot of features that are highly correlated; I probably could have spent a bit more time plugging in different combinations of those features. I also did not scrape data by census tract and chose instead to scrape data at the neighborhood level, which is a geographically larger region. Perhaps if I scraped the census data I would have been able to have a more precise model. Additionally, I had features that actually had the geographic location (city block of where each crime was reported, but I did not have the time to explore pandas capabilities to use this data.