DC Travel Recommendations

Introduction

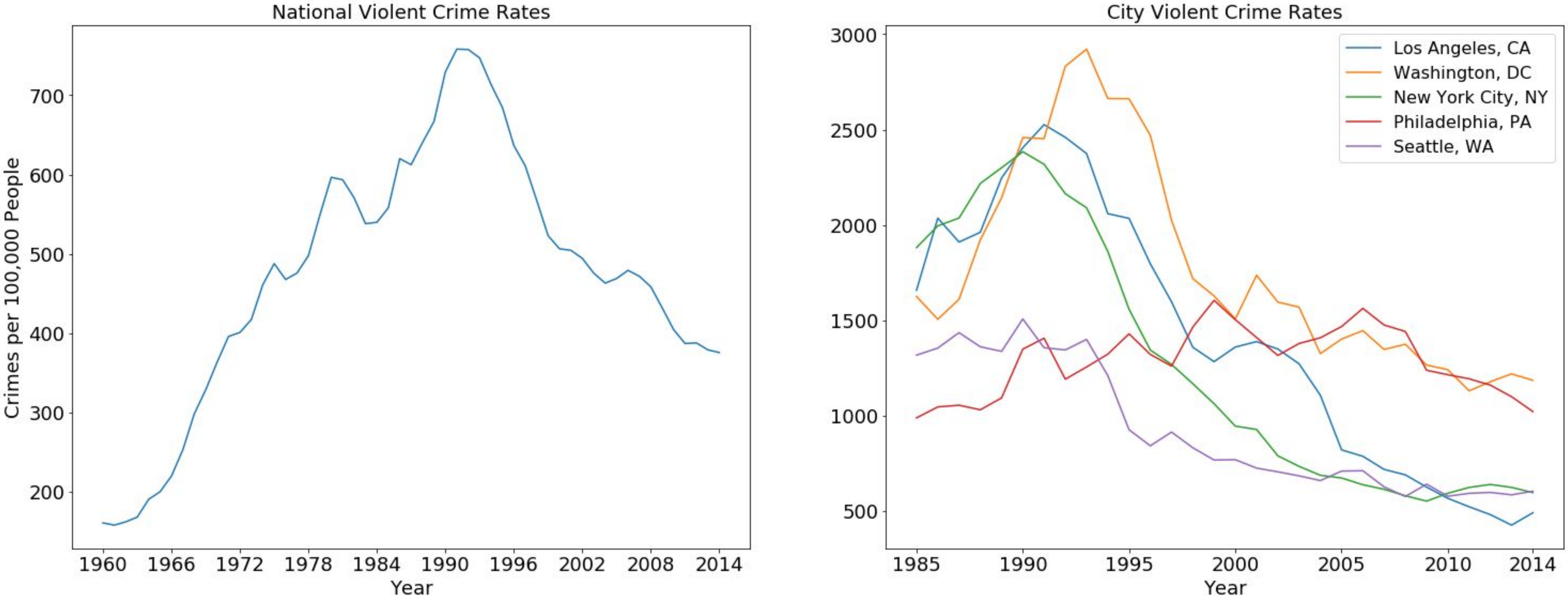
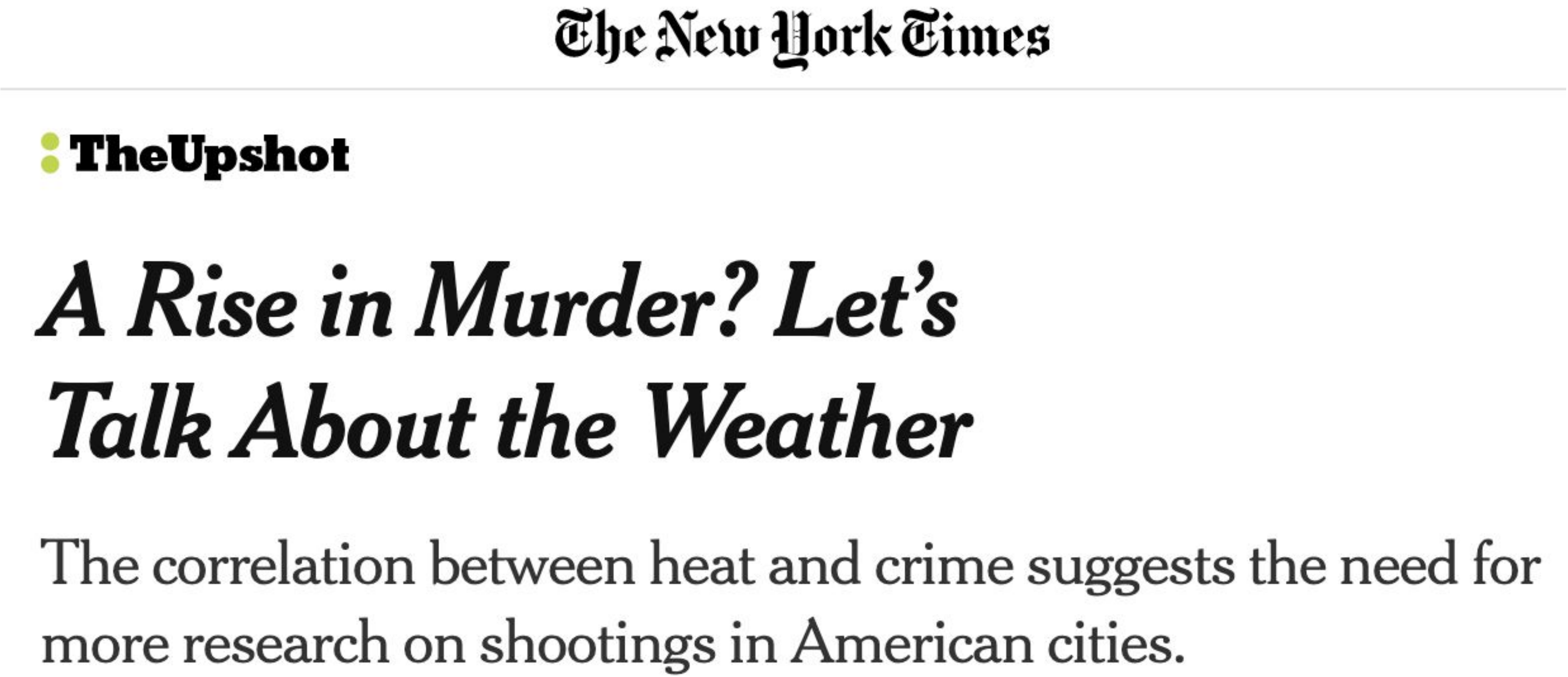


Figure 1: Violent crime rates, Federal Bureau of Investigation, Universal Crime Reporting Statistics Database, 1960 - 2014 (left), 1984 - 2014 (right). Extracted 5/25/2019.



The New York Times

TheUpshot

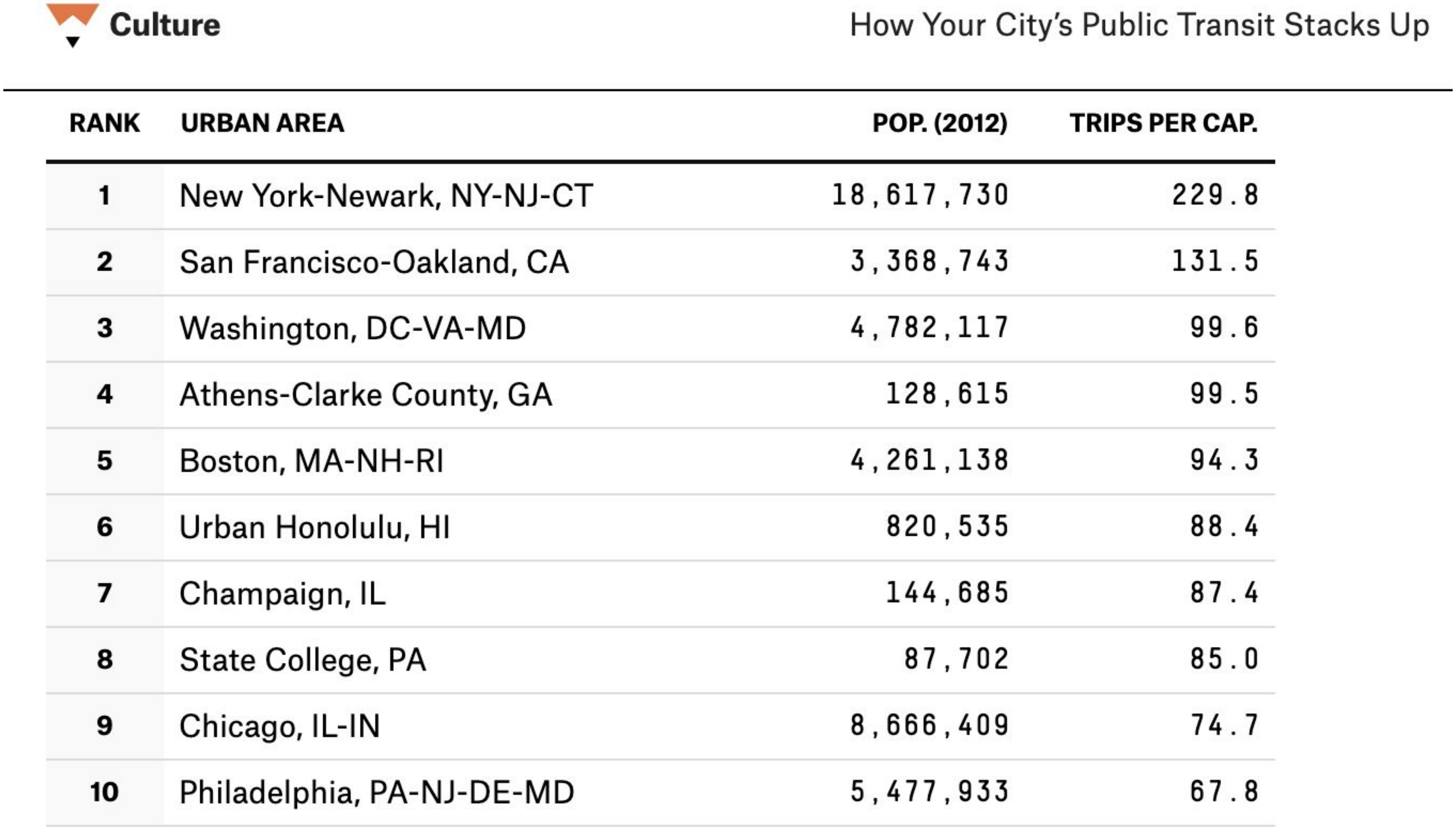
A Rise in Murder? Let's Talk About the Weather

The correlation between heat and crime suggests the need for more research on shootings in American cities.

Mayor Bowser Kicks Off 2019 Safer, Stronger DC Summer Crime Initiative

Wednesday, May 1, 2019

Mayor and MPD to Focus Resources to Reduce Violent Crime in Identified DC Neighborhoods



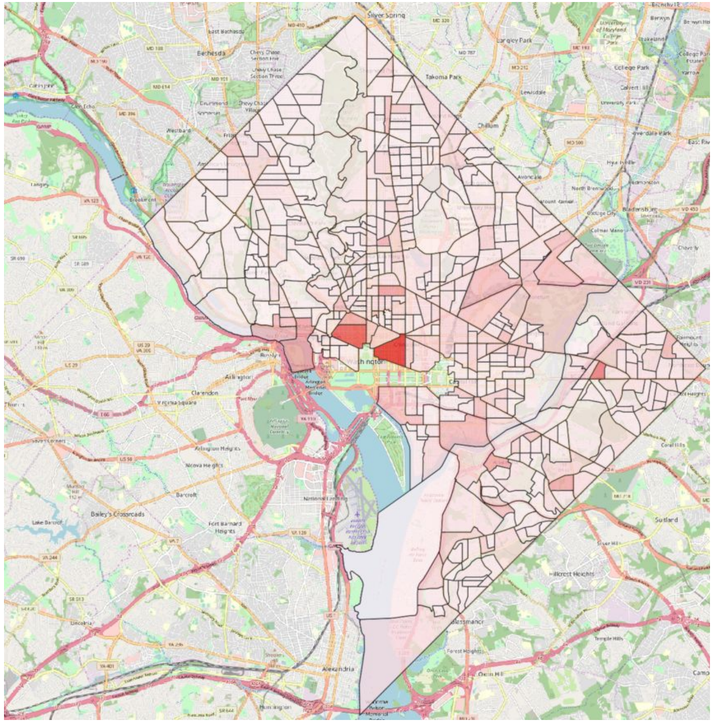
Culture

How Your City's Public Transit Stacks Up

RANK	URBAN AREA	POP. (2012)	TRIPS PER CAP.
1	New York-Newark, NY-NJ-CT	18,617,730	229.8
2	San Francisco-Oakland, CA	3,368,743	131.5
3	Washington, DC-VA-MD	4,782,117	99.6
4	Athens-Clarke County, GA	128,615	99.5
5	Boston, MA-NH-RI	4,261,138	94.3
6	Urban Honolulu, HI	820,535	88.4
7	Champaign, IL	144,685	87.4
8	State College, PA	87,702	85.0
9	Chicago, IL-IN	8,666,409	74.7
10	Philadelphia, PA-NJ-DE-MD	5,477,933	67.8

Project Goal

Use machine learning to predict crime rates by day, time, and location to offer users transit recommendations.



Project Overview

- A user looking for a safe transit recommendation inputs a Washington, DC address and date and time.

- Extract features based on geography, date, and time, and feed into model.

- If crime rate is predicted to be “high”, recommend user takes a lyft or rideshare. Otherwise, recommend metro, bus, and capitol bike share options close to the user’s input address.

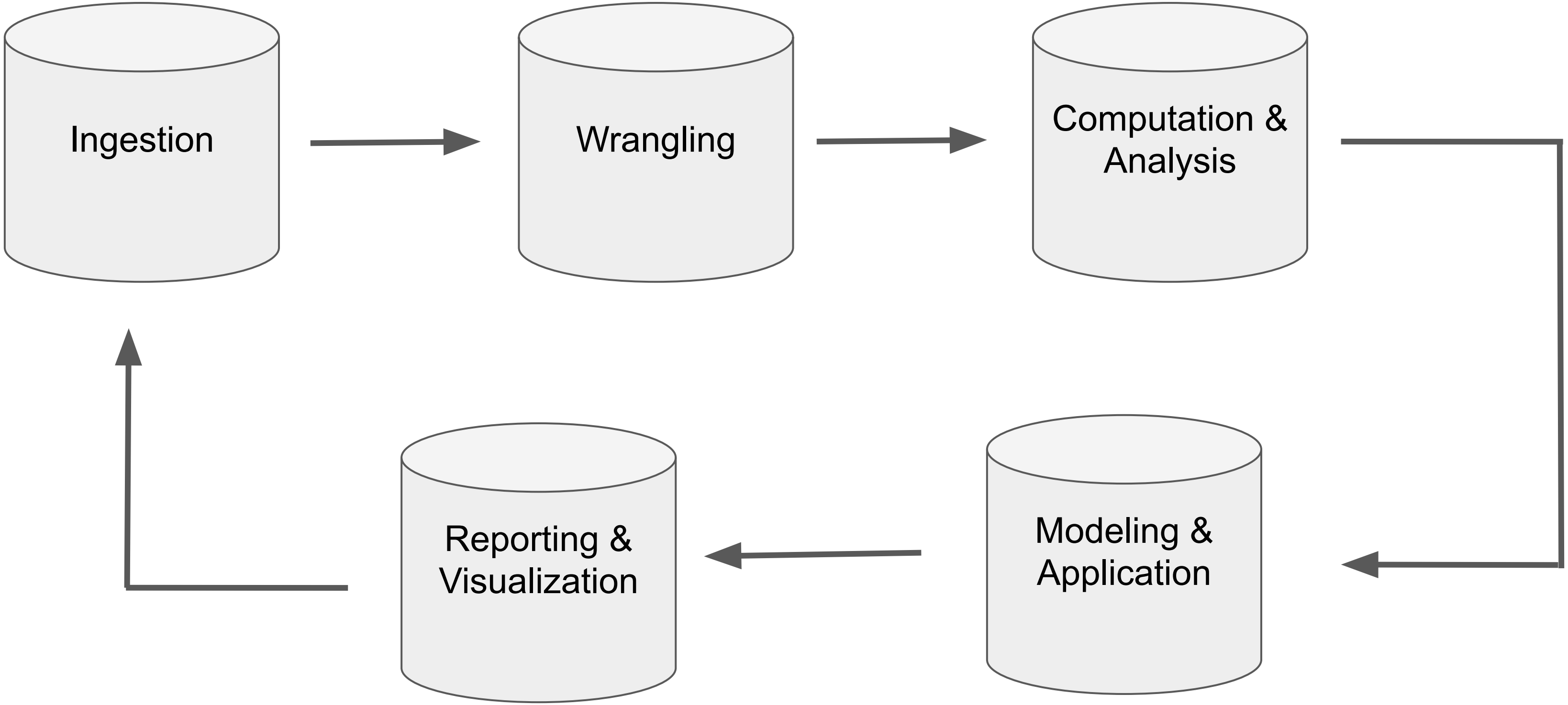
Hypothesis

- Initial hypothesis was to predict crime types in an area, but as the underlying assumption was unappealing and had been done before.

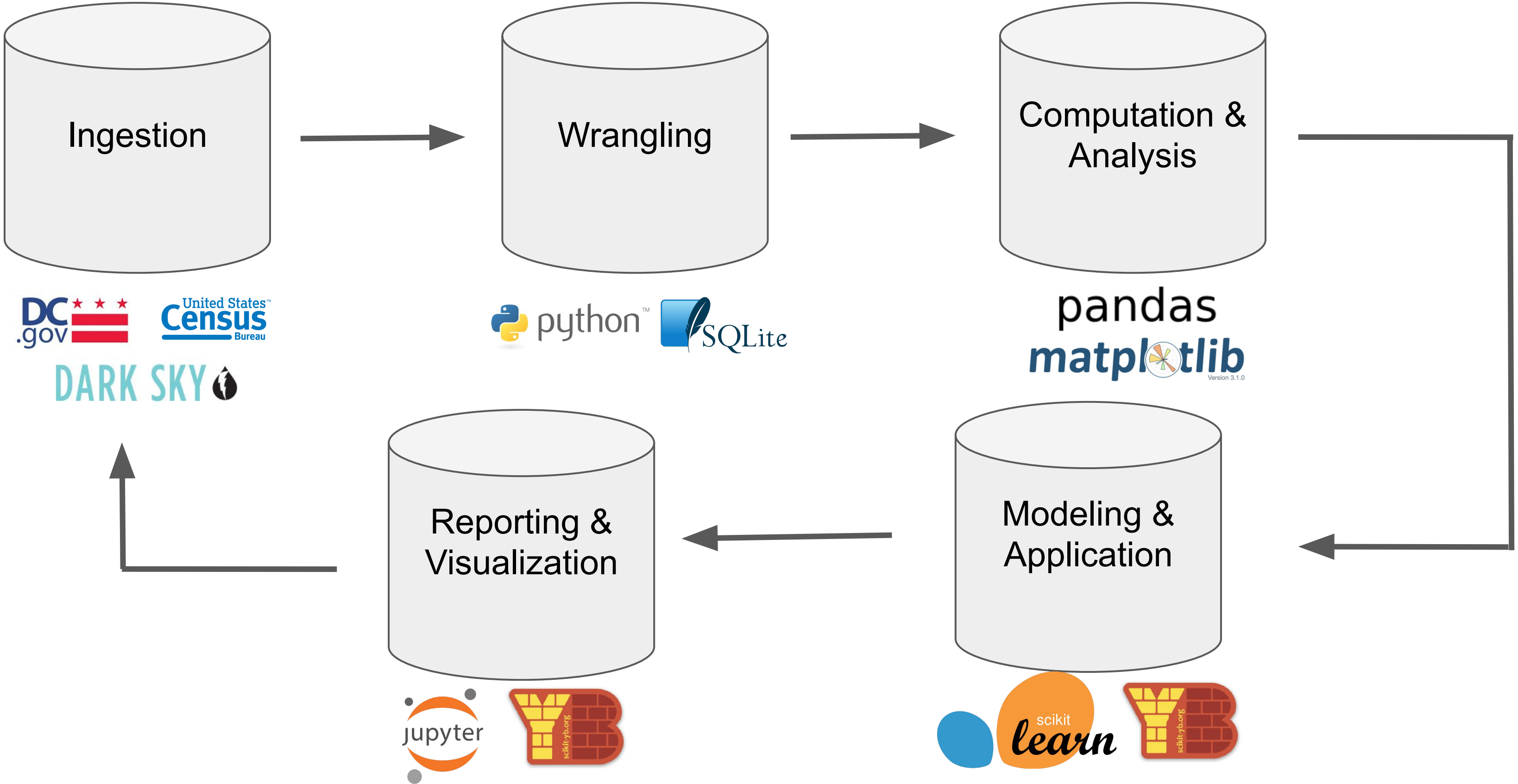
- The team redefined the hypothesis to predict crime rates in an area for a particular time-of-day.

- Final Hypothesis was to turn the regression problem into a classification problem, predicting labeled crime rates, low, medium, and high.

Data Science Pipeline: Architecture



Data Science Pipeline: Architecture



Ingestion

Wrangling

Computation & Analysis

Modeling & Application

Reporting & Visualization

Ingestion

Crime Data: DC.gov

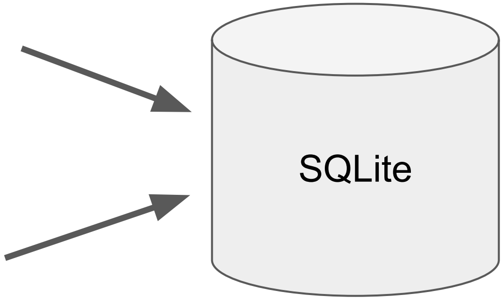
- ~380,000 reported crimes, 28 features, 1 target

Weather data: DARK SKY

- ~380,000 weather instances, 22 features

Census data: United States Census Bureau

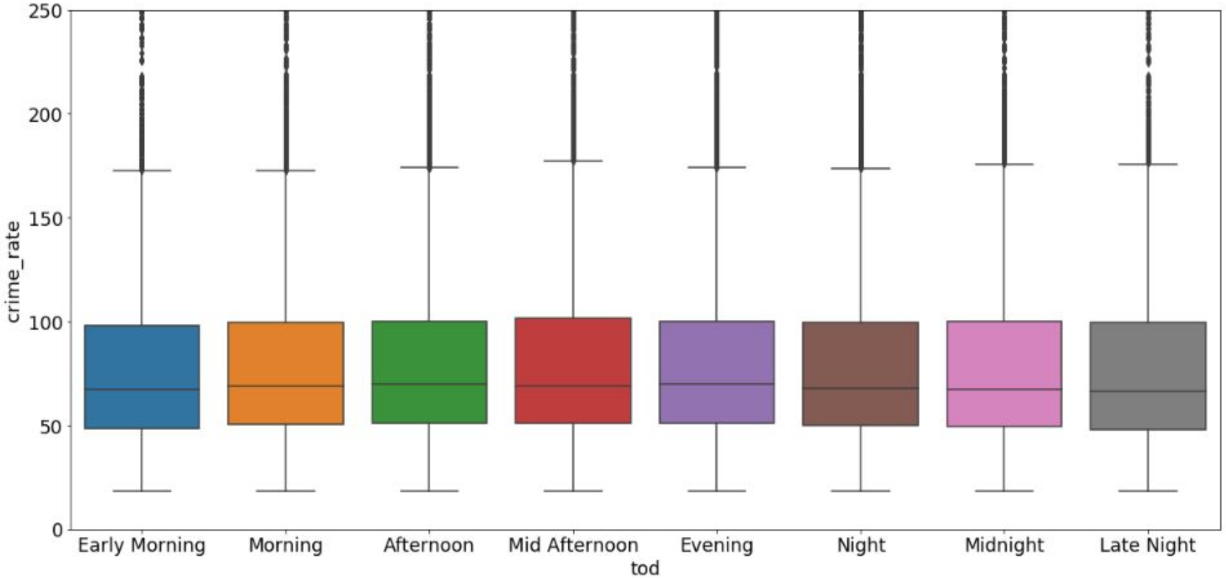
- 450 Census Blocks, 7 features, 1 target
- Missing 2010-2012



SQLite

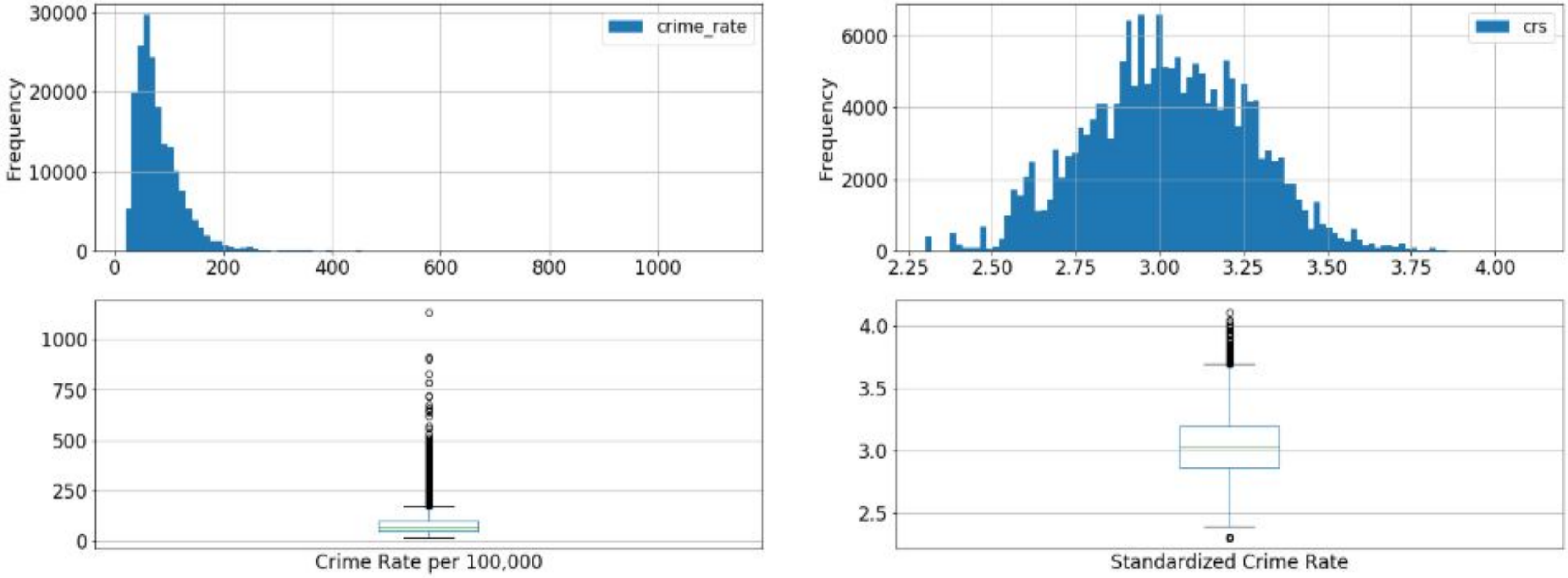
Wrangling

- Feature Creation:
  - Weekday
  - Time of day
- Target:
  - Count crime reports
  - Calculate crime rates
  - Rescale Crime rates



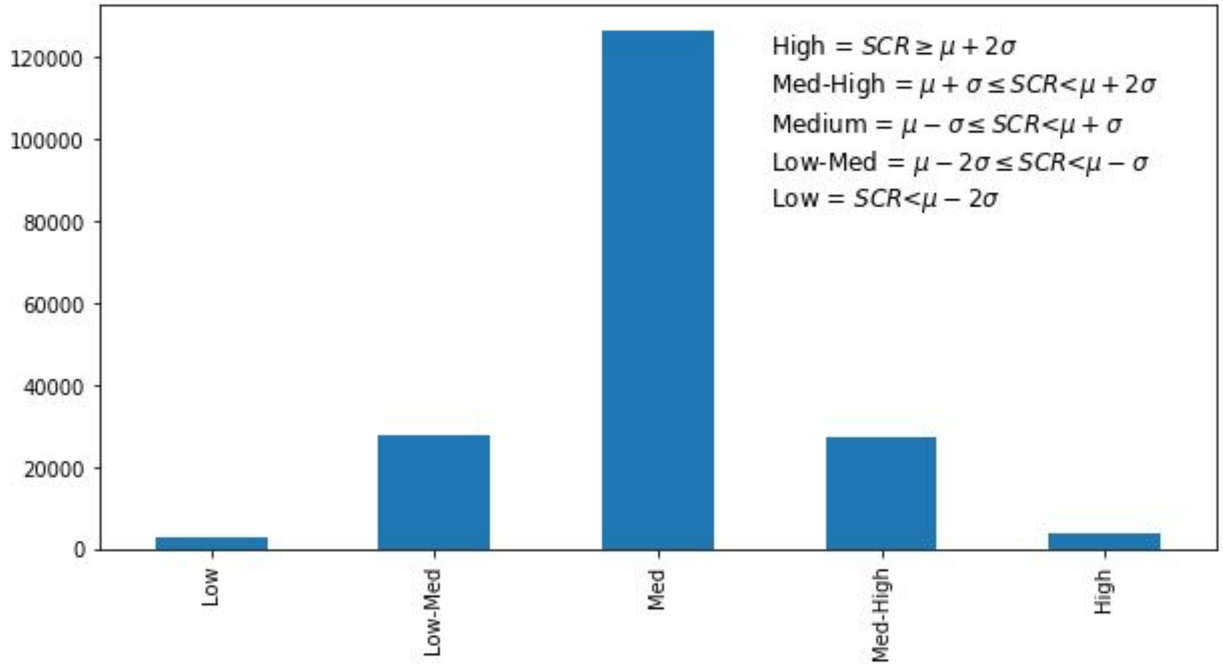
Creating Target

- Classifying crime rates into buckets.

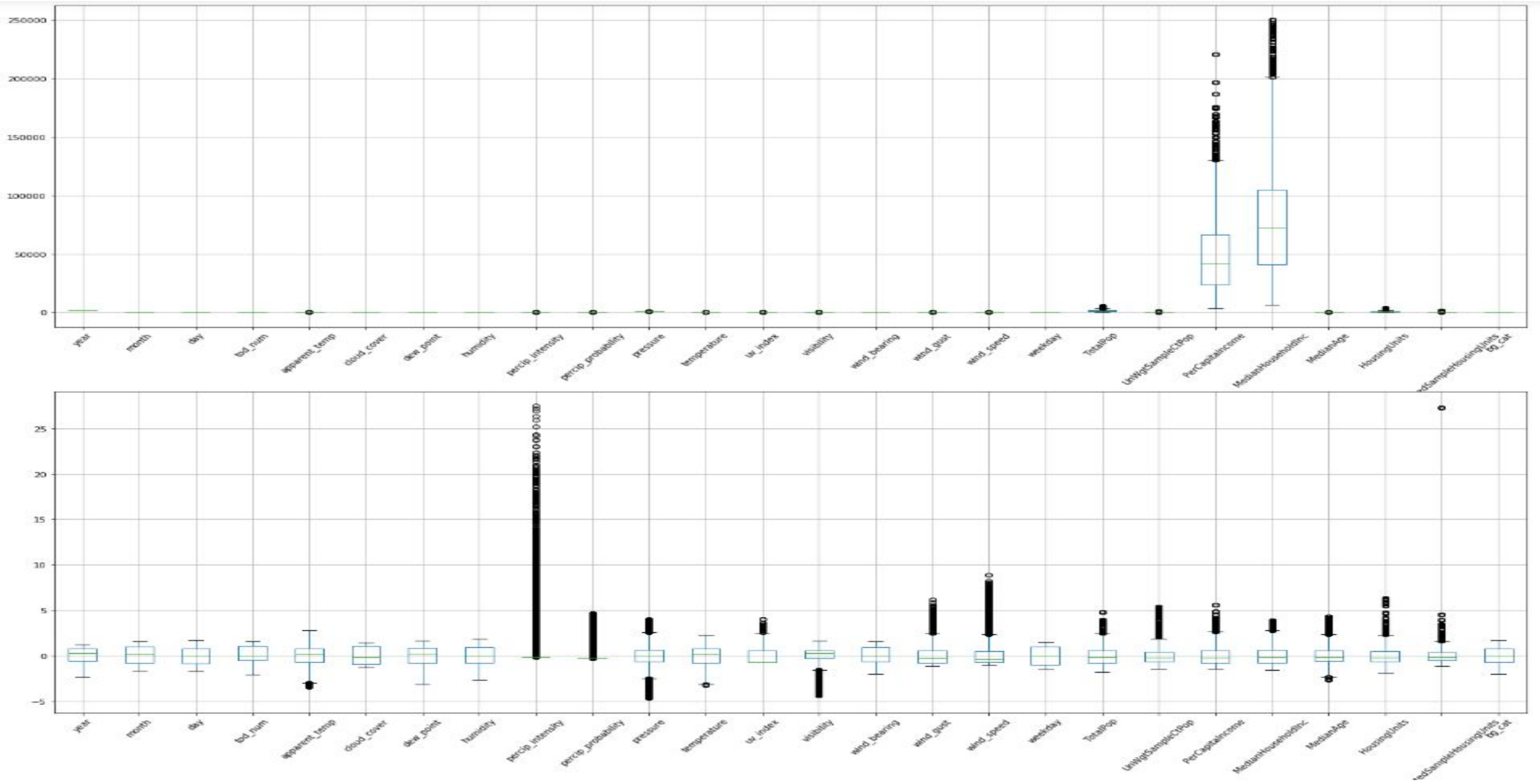


Creating Target

- Final class balance.



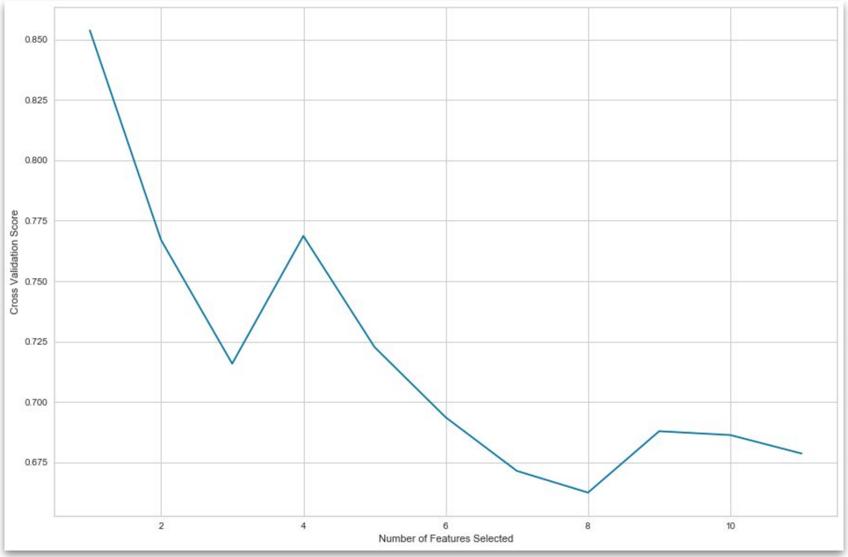
Standardization



Feature Selection

- Various methods used to identify feature set used for model evaluation
  - Cross validation scores
  - Yellowbrick Feature Importance
  - Common Sense

- Complexity of model increases as feature count increases

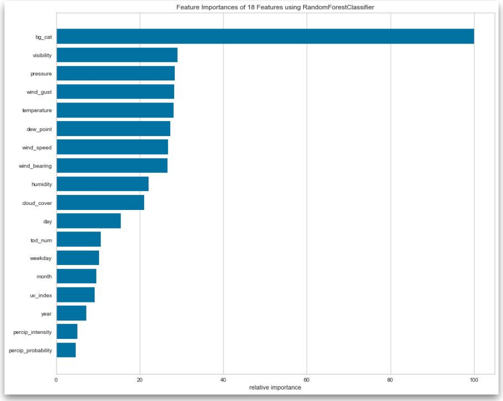


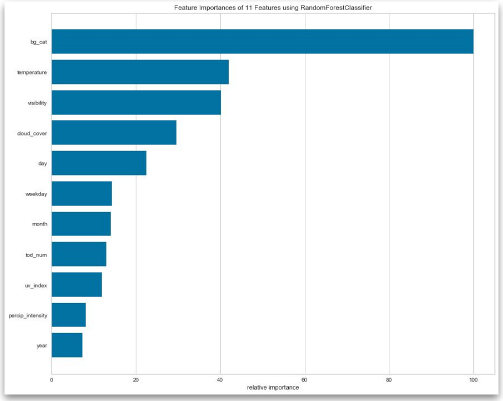
Fewer features improved model performance

Feature Selection (Continued)

- Preliminary model investigation produced unusually high accuracy and performance
  - Team concluded census features were leaking information to the model
  - Total population used in calculation of crime rate
    - Remaining census features temporally correlated with target, and subsequently removed from feature set

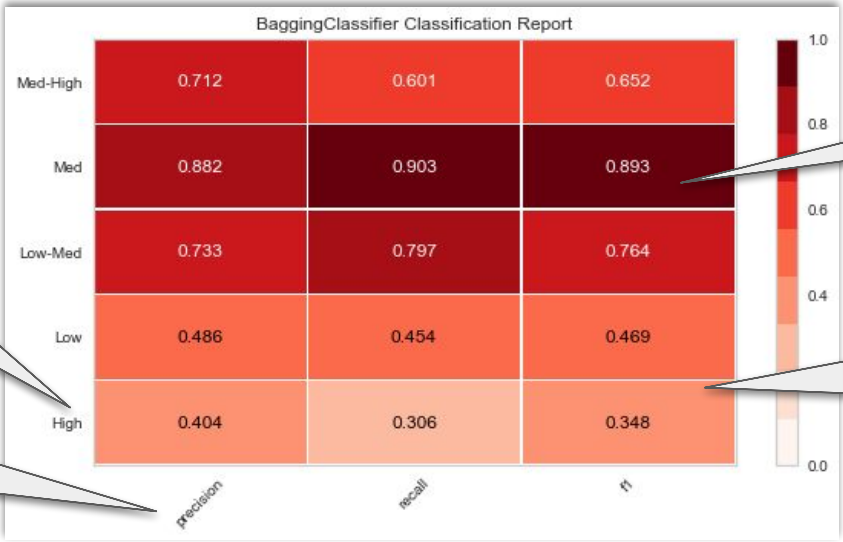
- Yellowbrick feature importance used to explore relative ranking amongst feature set
  - Helped eliminate features that continuously produced low relative feature importance (e.g. year)





Modeling

- Final model features included
  - Block Group Category, Time of Day, Day of the month, Weekday, temperature, and UV Index
- Bagging classifier using the decision tree estimator chosen as final model
  - produced the highest accuracy, precision, recall, and f1 scores



BaggingClassifier Classification Report

	precision	recall	f1
Med-High	0.712	0.601	0.652
Med	0.882	0.903	0.893
Low-Med	0.733	0.797	0.764
Low	0.486	0.454	0.469
High	0.404	0.306	0.348

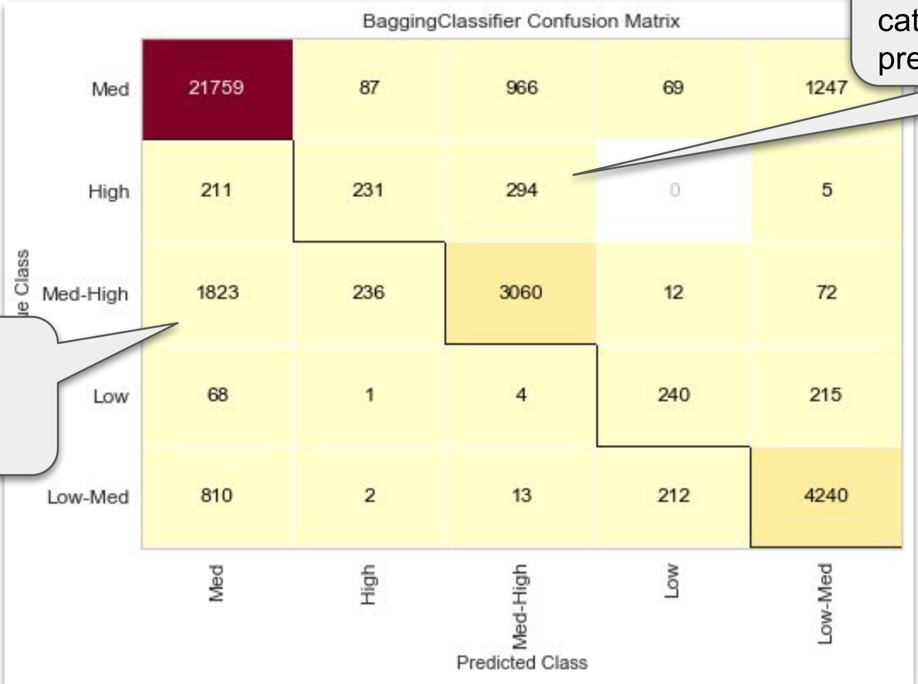
High category produced undesirable results

Comparable results when comparing precision and recall

Medium category produced highest f1 score

Stratified sampling, or undersampling could potentially improve categorical performance

Results



BaggingClassifier Confusion Matrix

	Med	High	Med-High	Low	Low-Med
Med	21759	87	966	69	1247
High	211	231	294	0	5
Med-High	1823	236	3060	12	72
Low	68	1	4	240	215
Low-Med	810	2	13	212	4240

Incorrect model predictions for high were categorized in adjacent categories (e.g. high incorrectly predicted into med/high category)

Visual inspection shows pattern of incorrect predictions categorized in nearby categories

Model rarely predicted higher rate categories into lower rate categories

Potential Model Improvements

- Undersample or stratify data
  - Addressing imbalanced target sample distribution could potentially improve model scores
- Hyper Parameter tuning
- Refine geographic level used in modeling
  - Initially constrained by census block group as the lowest geographic level
    - Potential to define custom geographic areas based on location

Limitations and Next Steps

- Data limitations
    - Limited to only geography, crimes reported, and weather
    - Potential other features to explore: big events, high foot traffic locations
- Limited crimes to 2009 to 2017 (excluding 2010-2012)
    - Census ACS data unavailable at block group level for 2010-2012
        - Could impute this data and re-train model
    - Rewrite program to constantly ingest reported crimes and associated weather data and regularly re-train model
- Program predicts crime rate category for only the census block group for the address the user inputs
    - Future development could also predict crime rate category in neighboring block groups, and return transit options from the “safest” block group at the time a user wants to travel
- Only considering starting location, not ending location

Demo

Launch Jupyter Notebook