



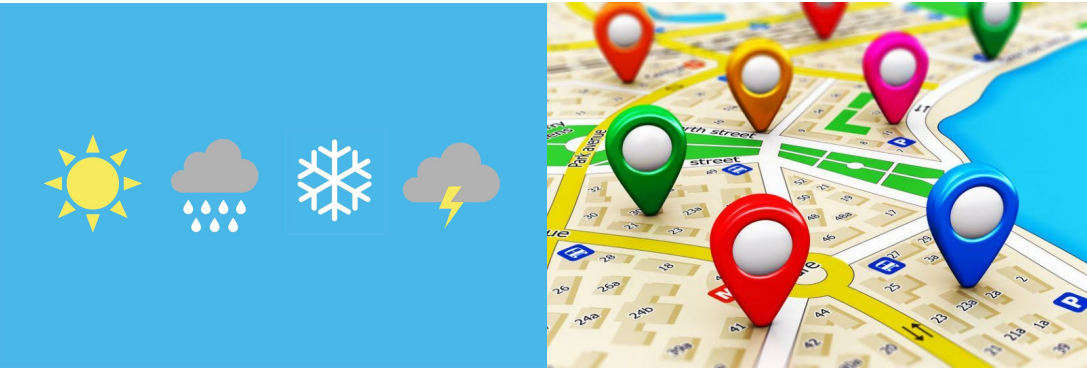
New York City Crime Analysis

Clarence Li
Michael Gokhler
William Sit
Brian McCullough



Crime Prediction

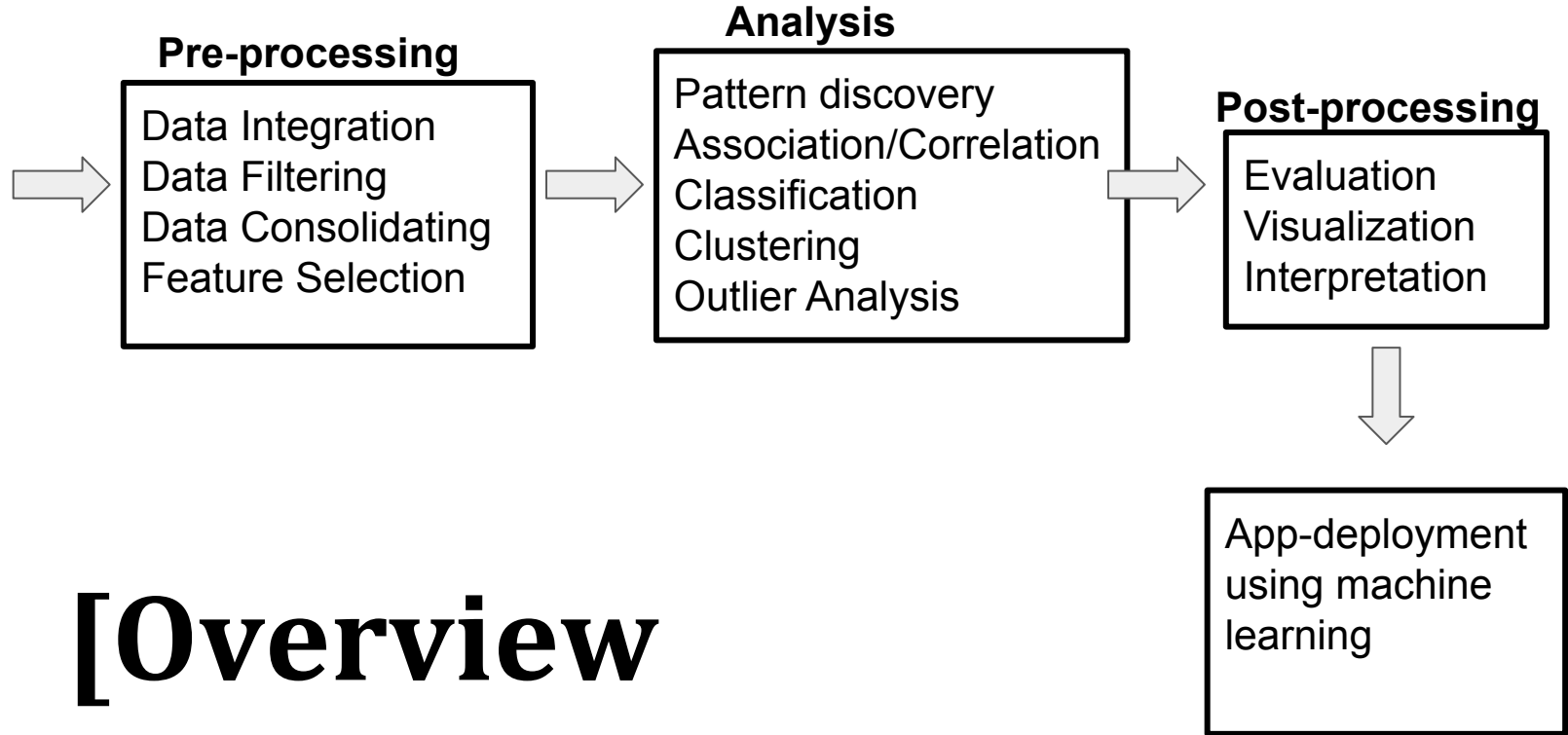
- How can we predict crime, based on (via Machine Learning):
 - Weather (temperature)
 - Location (borough/ ZIP code)
 - Time of day (holidays)
 - Others....



Possible uses of the tool:

- law enforcement workforce resources planning
- Government policy planning
- Travel plans

**Raw
Data**



**[Overview
]**

More in depth view of tools

- Python
 - Pandas
 - Datetime
 - Numpy
 - Flask
 - Joblib
 - Machine Learning
 - Linear Regression
 - HTML
 - Excel
 - Heroku
- API
 - Google Map API
 - MapQuest API
 - Tableau



Data Cleaning

- Filtering:
 - NYPD Jurisdiction
 - Dates from 2006-2017
 - Proper age groups
- Dropping:
 - Removing unnecessary columns
- Renaming:
 - Short-handed nomenclature into legal column titles
- Combining:
 - Consolidating related data



Filtering Jurisdiction to NYPD

```
# Check all of the NYC policing jurisdictions  
nycrime['JURIS_DESC'].value_counts()
```

N.Y. POLICE DEPT	5363022
N.Y. HOUSING POLICE	465254
N.Y. TRANSIT POLICE	132465
PORT AUTHORITY	29354
OTHER	16899
POLICE DEPT NYC	8955
DEPT OF CORRECTIONS	7813
TRI-BORO BRDG TUNNL	5072
HEALTH & HOSP CORP	3045
N.Y. STATE POLICE	1488
METRO NORTH	656
LONG ISLAND RAILRD	491
FIRE DEPT (FIRE MARSHAL)	477
STATN IS RAPID TRANS	385
N.Y. STATE PARKS	374
NYC PARKS	327
U.S. PARK POLICE	222
NEW YORK CITY SHERIFF OFFICE	192
AMTRACK	170
NYS DEPT TAX AND FINANCE	84
SEA GATE POLICE DEPT	30
CONRAIL	14
NYC DEPT ENVIRONMENTAL PROTECTION	14



Filtering Dates to 2006 to 2017

```
# Check starting date values. Verify datatype of starting date as string.  
nycrime_renameCOL['Complaint_StartDate'].head()
```

```
0    11/10/2006  
1    11/10/2006  
2    11/10/2006  
3    11/10/2006  
4    11/10/2006
```

```
Name: Complaint_StartDate, dtype: object
```

```
# Convert date string into datetime
```

```
nycrime_renameCOL["dateTime_start"]=pd.to_datetime(nycrime_renameCOL['Complaint_StartDate'], errors='coerce', format='%m/%d/%Y')
```

```
# Verify datatype of starting date as datetime
```

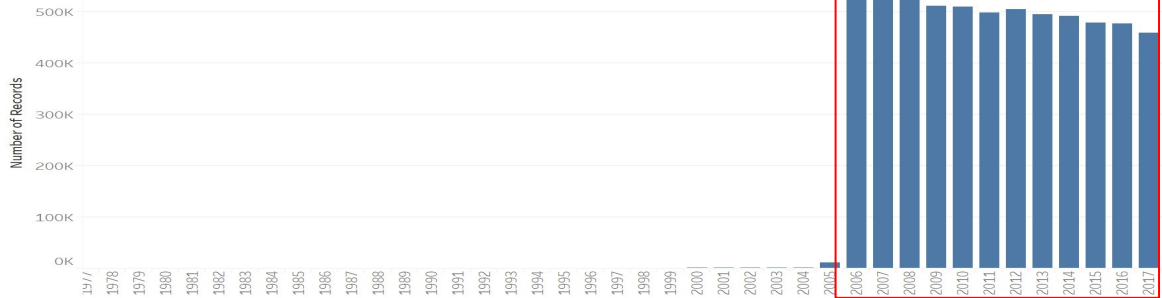
```
nycrime_renameCOL["dateTime_start"].head()
```

```
0    2006-11-10  
1    2006-11-10  
2    2006-11-10  
3    2006-11-10  
4    2006-11-10
```

```
Name: dateTime_start, dtype: datetime64[ns]
```

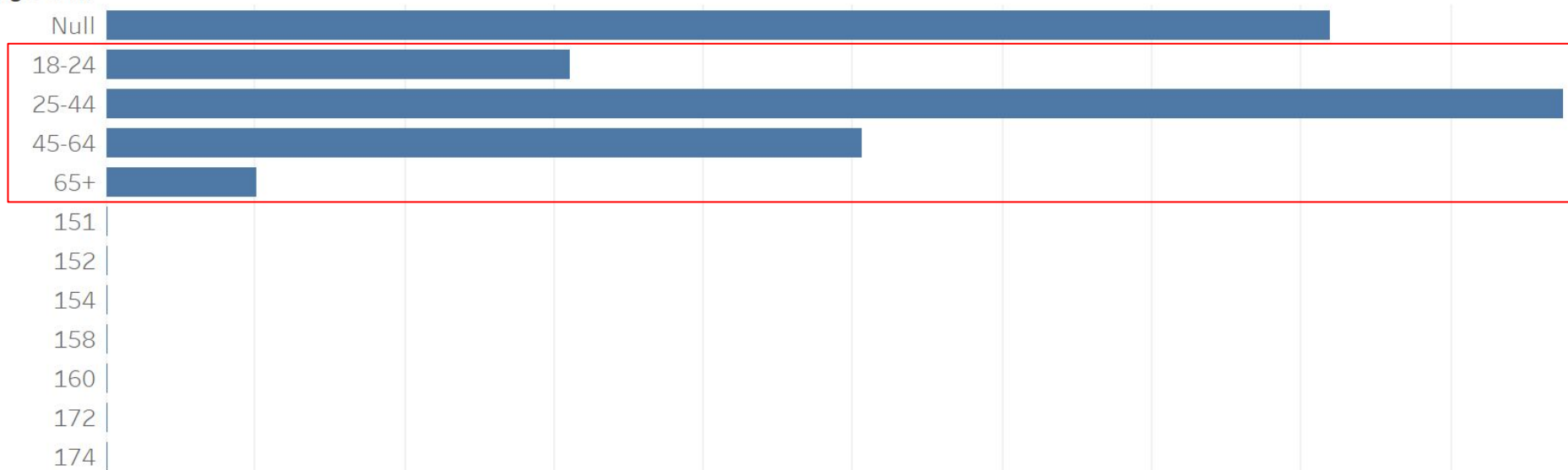
```
# Filter date to start at 2006
```

```
nycrime_filterDate = nycrime_renameCOL[(nycrime_renameCOL['dateTime_start'] >= '2006-01-01')]
```



Filtering Age Groups

Vic Age Gro..



```
# Filter age values to sensible ranges
```

```
filterAge_list = ['UNKNOWN', '<18', '18-24', '25-44', '45-64', '65+']
```

```
nycrime_filterAge = nycrime_filterDate[nycrime_filterDate.Suspect_Age.isin(filterAge_list) & nycrime_filterDate.Victim_Age.
```

Consolidating Similar Crime Types

Source: NYC law (New York State Law):

<http://ypdcrime.com/penal.law/>

```
# Check List of article offenses
```

```
nycrime_TitleOffense['ArticleOffense'].value_counts()
```

HARASSMENT,SUBD 3,4,5	188814
ASSAULT 3	167239
HARASSMENT,SUBD 1,CIVILIAN	138147
AGGRAVATED HARASSMENT 2	118331
ASSAULT 2,1,UNCLASSIFIED	53026
LARCENY,PETIT FROM STORE-SHOPL	51189
CRIMINAL CONTEMPT 1	37127
CRIMINAL MISCHIEF,UNCLASSIFIED 4	33701
LARCENY,PETIT FROM BUILDING,UN	29882
MENACING,UNCLASSIFIED	25839

```
# Combining similar article offenses together. Nomenclature can be referenced here: http://ypdcrime.com/penal.law/
```

```
nycrime_ArticleOffense = nycrime_TitleOffense
```

```
nycrime_ArticleOffense['ArticleOffense'] = nycrime_ArticleOffense['ArticleOffense'].replace(  
    {
```

```
# PENAL LAW=====
```

```
# TITLE G ANTICIPATORY OFFENSES
```

```
    # ARTICLE 100 CRIMINAL SOLICITATION
```

```
        'SOLICITATION 3,2,1, CRIMINAL':'CRIMINAL SOLICITATION',
```

```
        'SOLICITATION 4, CRIMINAL':'CRIMINAL SOLICITATION',
```

```
        'SOLICITATION 5,CRIMINAL':'CRIMINAL SOLICITATION',
```

```
    # ARTICLE 105 CONSPIRACY
```

```
        'CONSPIRACY 2, 1':'CONSPIRACY',
```

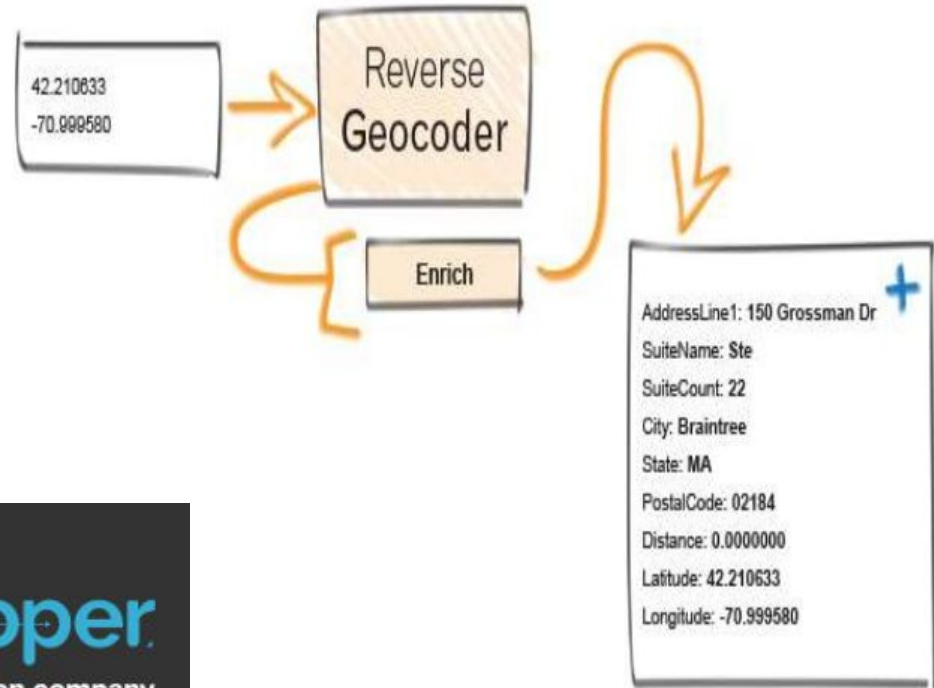
```
        'CONSPIRACY 4, 3':'CONSPIRACY',
```

```
        'CONSPIRACY 6, 5':'CONSPIRACY',
```

```
    # ARTICLE 110 ATTEMPT
```

Reverse Geocoding

- Process of back-coding from a point location (latitude/longitude) into an address (or in this case, ZIP code)
 - Google Maps API
 - Python Libraries:
 - GeoPy
 - Pygeocoder
 - Etc...
 - MapQuest API





Latitude	Longitude
40.663613	-73.917945



coordinate
40.663612562,-73.917945322

```
gkey= ' ' ,
```

```
zipcode=[]  
coordinate=[]
```

```
for i in range(len(nycrime_dropCOL)):  
    latlon = f'{nycrime_dropCOL["Latitude"][i]},{nycrime_dropCOL["Longitude"][i]}'  
    # print(latlon)  
    coordinate.append(latlon)
```

```
nycrime_dropCOL['coordinate'] = coordinate
```

```
# https://maps.googleapis.com/maps/api/geocode/json?latlng={40.714224, -73.961452}&key=YOUR_API_KEY
for i in range(len(nycrime_dropCOL["coordinate"])):
    target_url=f'https://maps.googleapis.com/maps/api/geocode/json?latlng={nycrime_dropCOL["coordinate"][i]}&key={gkey}'

    geo_data = requests.get(target_url).json()
    components = geo_data["results"][0]['address_components']
    for c in components:
        if "postal_code" in c['types']:
            zipcode.append(c['short_name'])

    if (i+1) % 100 == 0:
        row=i+1
        print(f'{row} rows has been successfully requested.')

# wait secs/request to avoid request-blocking
time.sleep(1)
```



Not enough budget for collecting millions of requests...

```
100 rows has been successfully requested.
200 rows has been successfully requested.
300 rows has been successfully requested.
400 rows has been successfully requested.
500 rows has been successfully requested.
600 rows has been successfully requested.
```

```
-----
OSError
~\Anaconda3\lib\site-packages\urllib3\conn
rt_same_host, timeout, pool_timeout, relea
599
--> 600
601
```

```
ProtocolError: ('Connection aborted.', OSError(0, 'Error'))
```

During handling of the above exception, another exception occurred:

```
ConnectionError
```

Traceback (most recent call last)

```
<ipython-input-33-291dd98a95f0> in <module>()
3
```

```
target_url=f'https://maps.googleapis.com/maps/api/geocode/json?latlng={nycrime_dropCOL["coordinate"][i]}&key={gkey}'
```




Source: <http://more.stevemorse.org/latlonbatch2.html?direction=reverse>

Via MapQuest Developer API key use

Monthly limit: 15,000

Latitude,Longitude pairs

40.66361256	-73.91794532
40.72404577	-73.93807074
40.69271257	-73.97905746
40.61052609	-73.96928323
40.73282833	-73.98606286

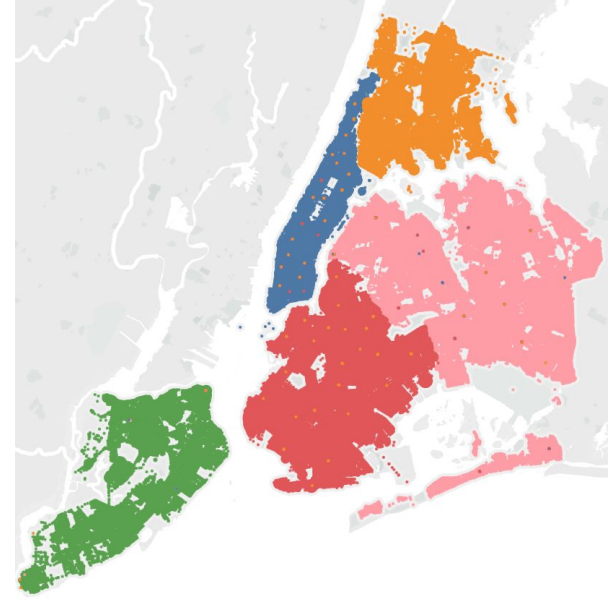
Addresses

187, Grafton Street, Brooklyn Communi
763, Meeker Avenue, Greenpoint, Kings
Kingsview Homes, Ashland Place, Fort
1648, East 5th Street, Gravesend, Bro
227, East 14th Street, Manhattan Comm

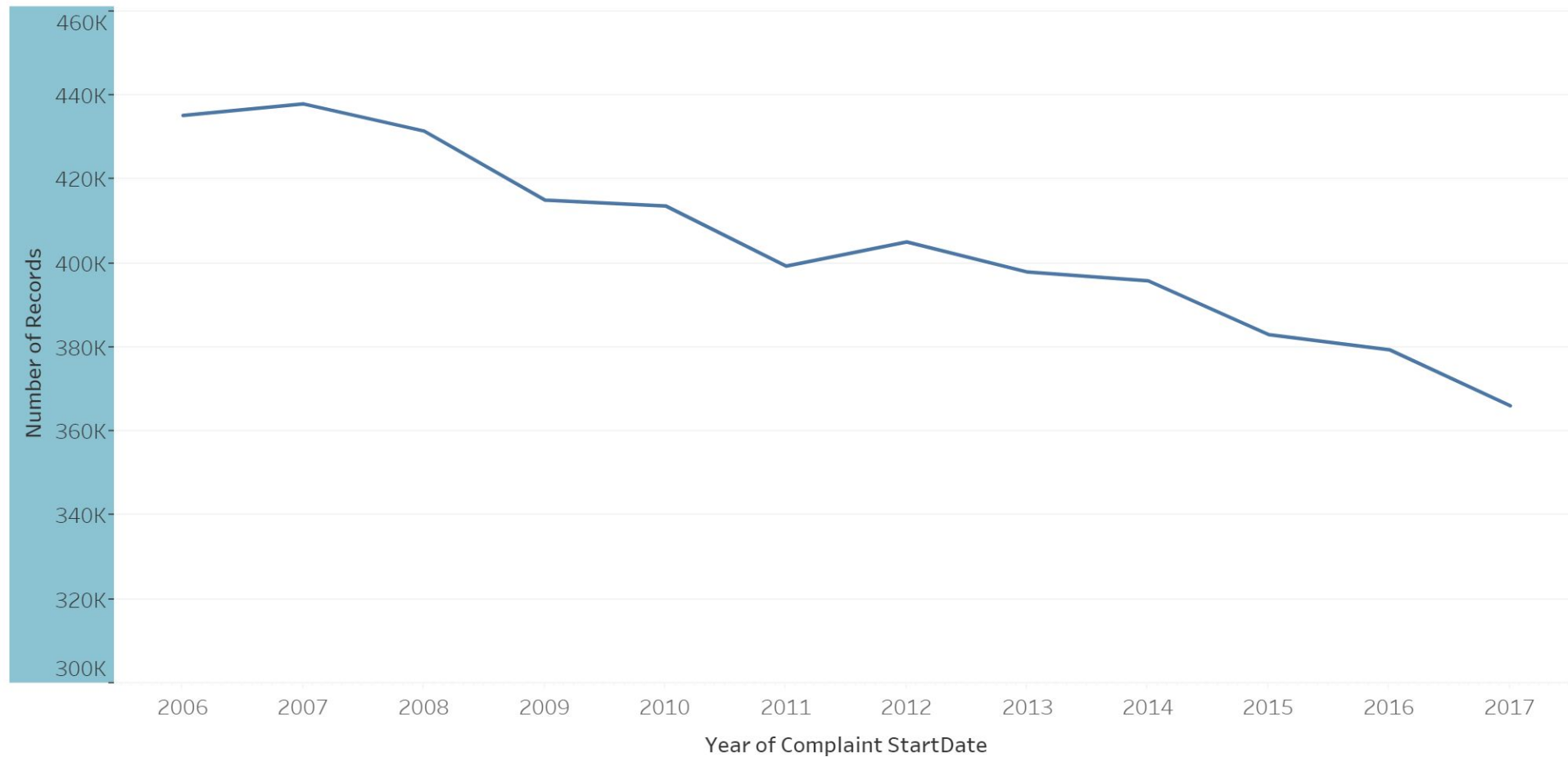
Latitude	Longitude	Address	Zipcode
40.66361	-73.9179	187, Grafton Street, Brooklyn Community Board 17 Neighborhoods, Brooklyn, Kings County, New York City, New York, 11212, United States of America	11212
40.72405	-73.9381	763, Meeker Avenue, Greenpoint, Kings County, New York City, New York, 11222, United States of America	11222
40.69271	-73.9791	Kingsview Homes, Ashland Place, Fort Greene, Brooklyn, Kings County, New York City, New York, 11217, United States of America	11217
40.61053	-73.9693	1648, East 5th Street, Gravesend, Brooklyn, Kings County, New York City, New York, 11230, United States of America	11230
40.73283	-73.9861	227, East 14th Street, Manhattan Community Board 6, New York County, New York City, New York, 10003, United States of America	10003

Data Visualizations

Using Tableau

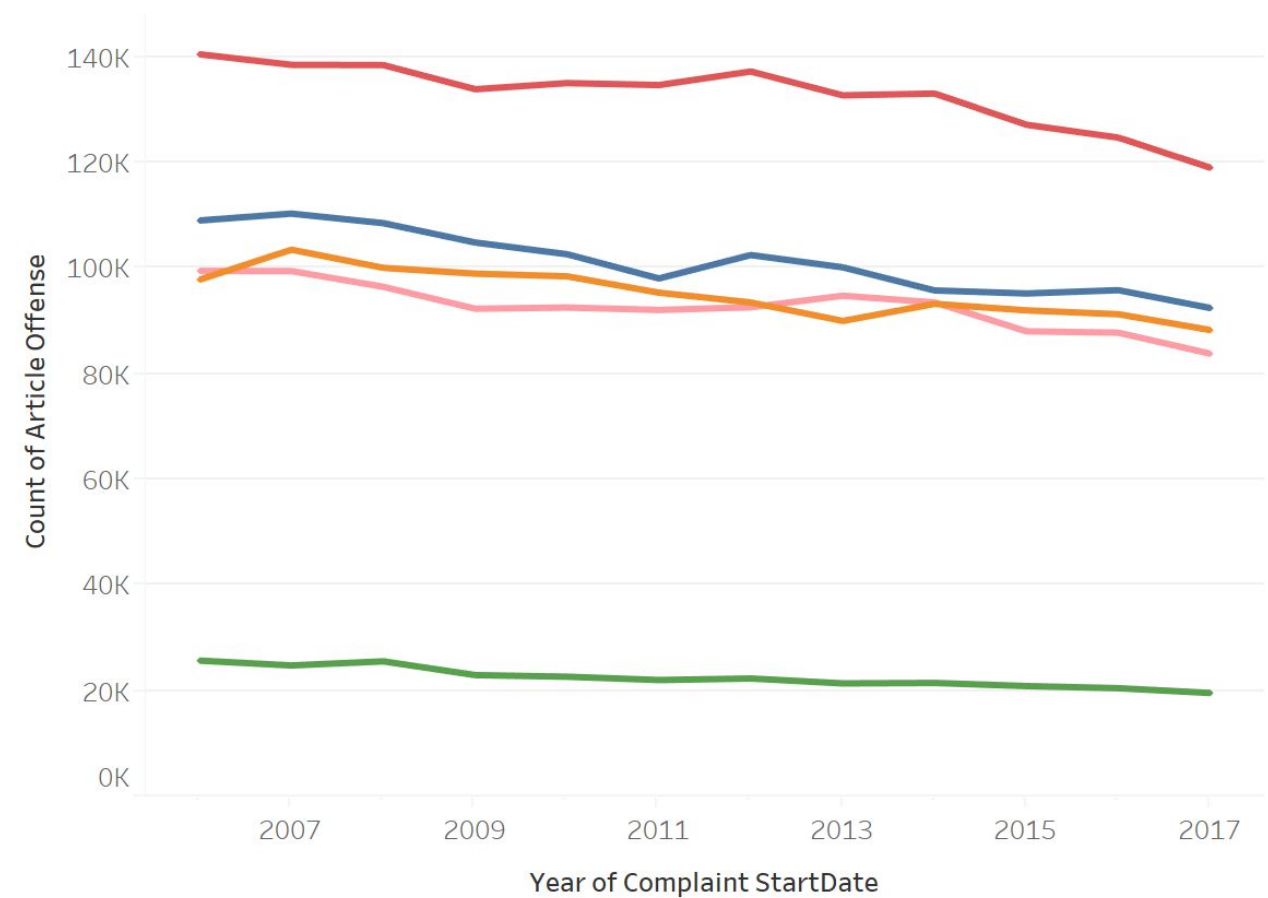


TOTAL over time



The trend of sum of Number of Records for Complaint StartDate Year. The data is filtered on Article Offense and top 10. The Article Offense filter keeps 55 of 55 members. The top 10 filter keeps 11 members.

Crime Type Trends per Year by Borough



Borough

BRONX

BROOKLYN

MANHATTAN

QUEENS

STATEN ISLAND

Article Offense

☐ (All)

☐ Null

☒ ADMINISTRATIVE LAW OFFENSES

☒ AGRICULTURE AND MARKETS LAW OFFEN...

☒ ALCOHOLIC BEVERAGE CONTROL LAW OF...

☒ ARSON

☒ ASSAULT AND RELATED OFFENSES

☒ BRIBERY INVOLVING PUBLIC SERVANTS A...

☒ BRIBERY NOT INVOLVING PUBLIC SERVA...

☒ BURGLARY AND RELATED OFFENSES

☒ CONSPIRACY

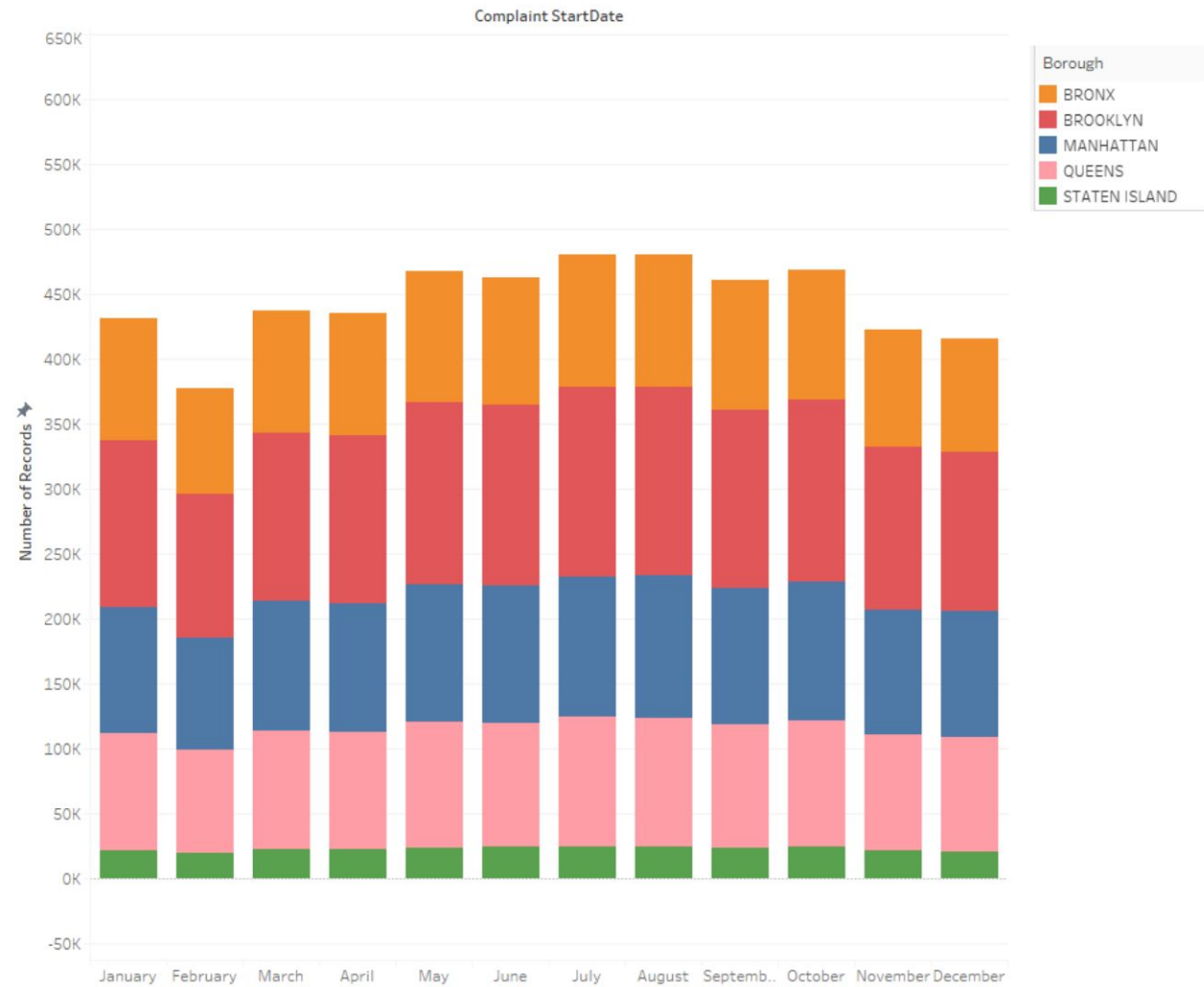
☒ CONTROLLED SUBSTANCES OFFENSES

☒ CRIMINAL FACILITATION

☒ CRIMINAL MISCHIEF AND RELATED OFFE...

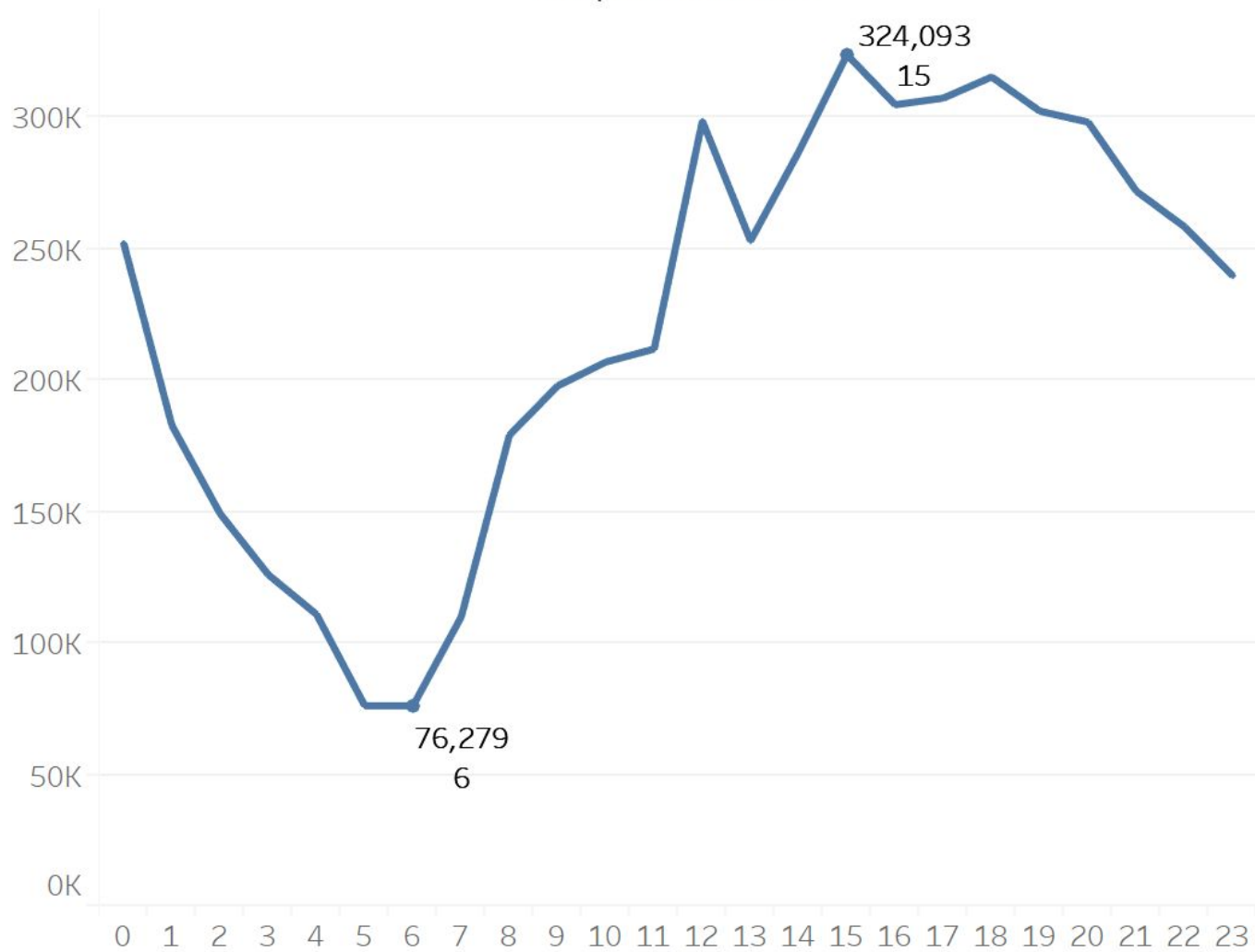
☒ CRIMINAL SOLICITATION

Total Number of Crimes From 2006-2017

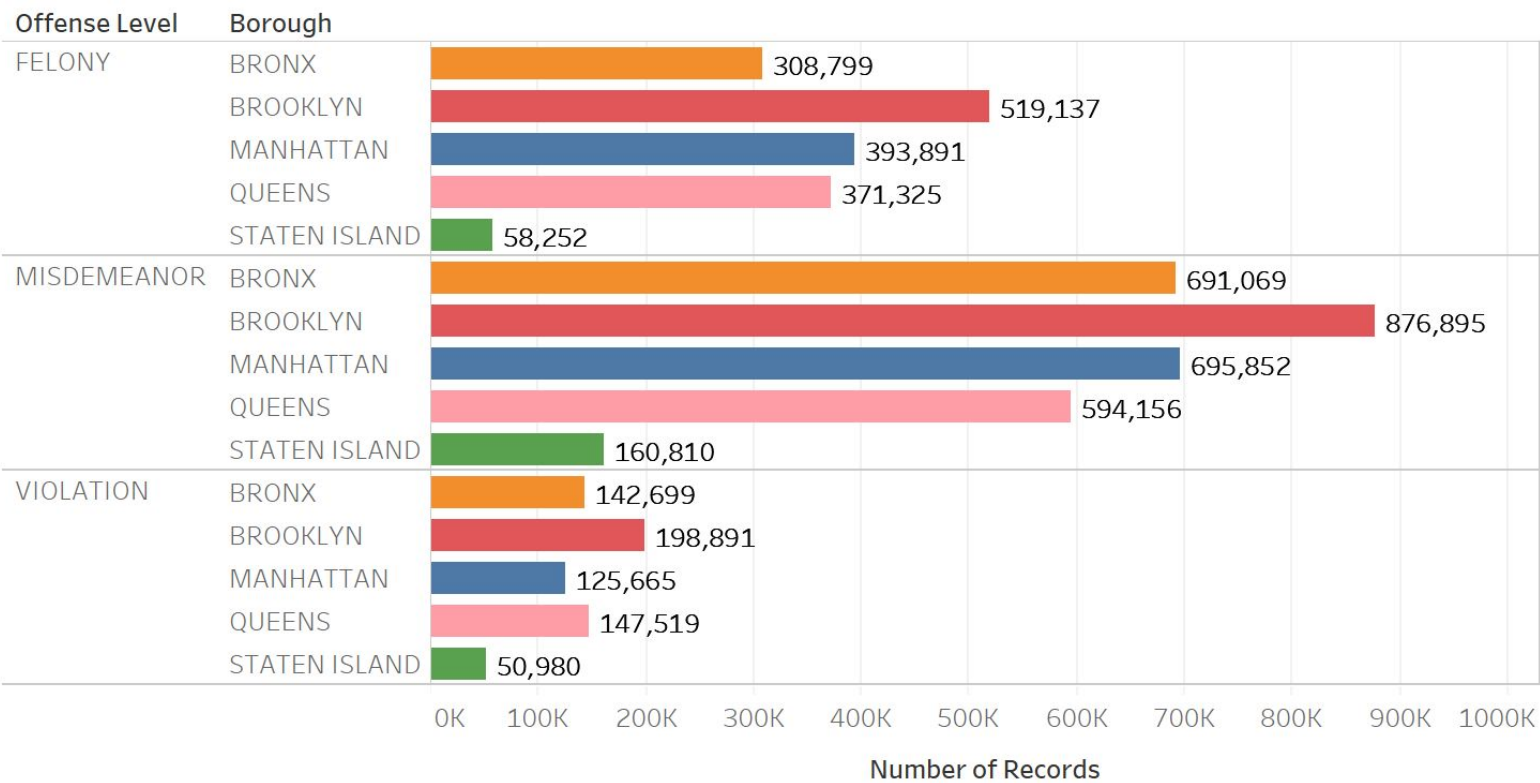


Complaint StartTime

Number of Records



BoroCrimeLevel

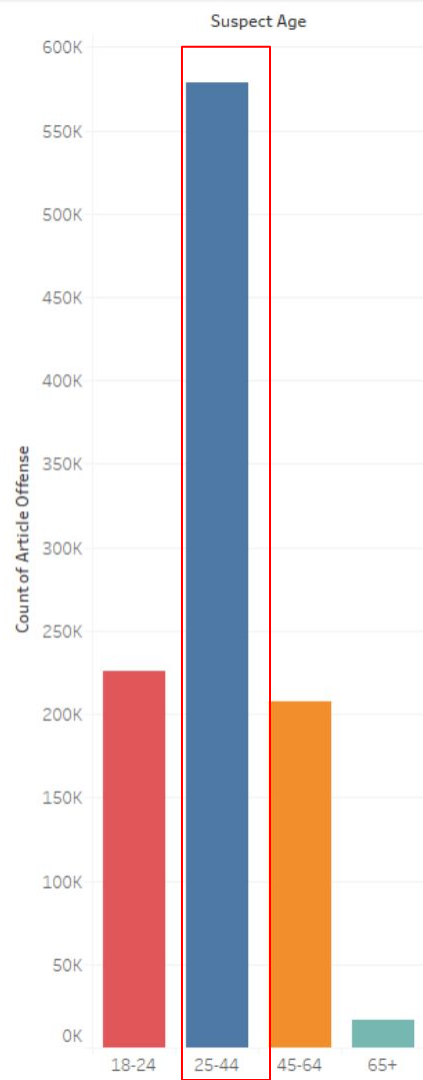
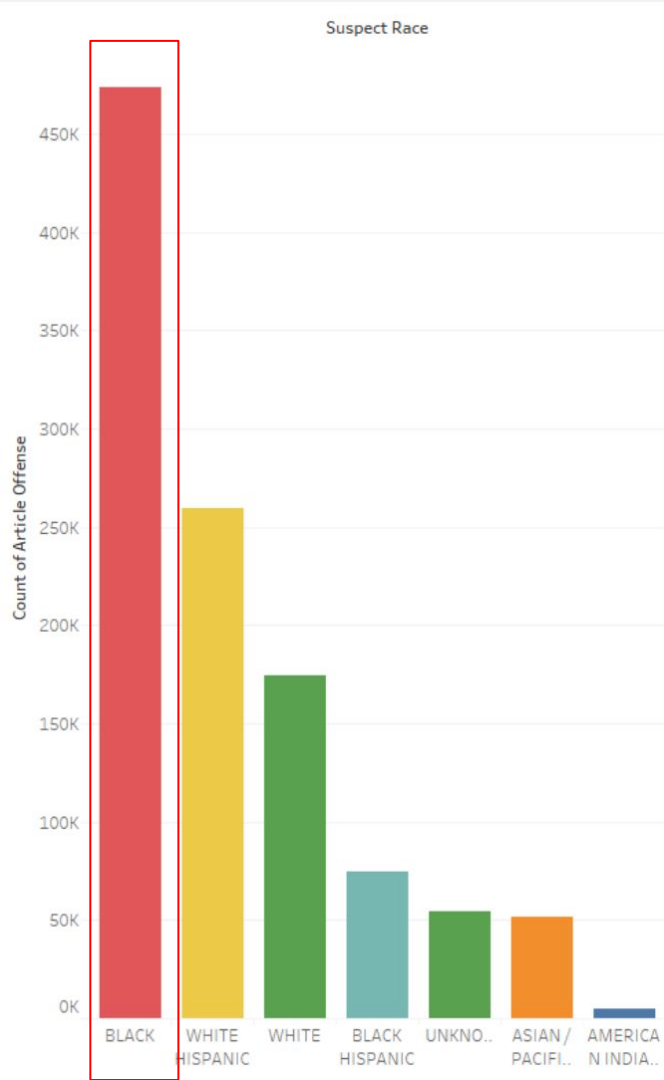
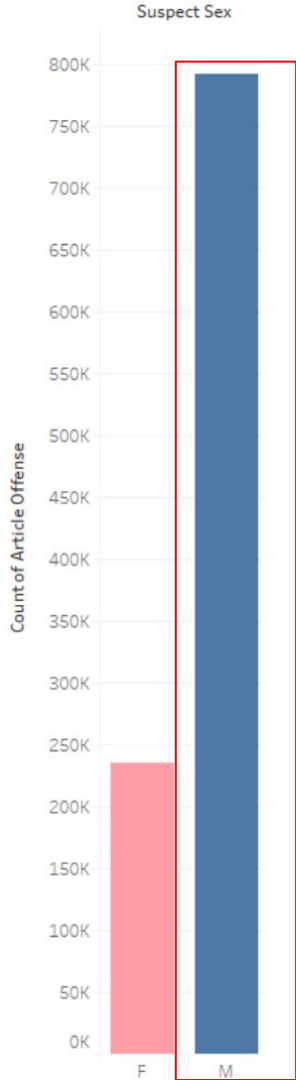


Borough

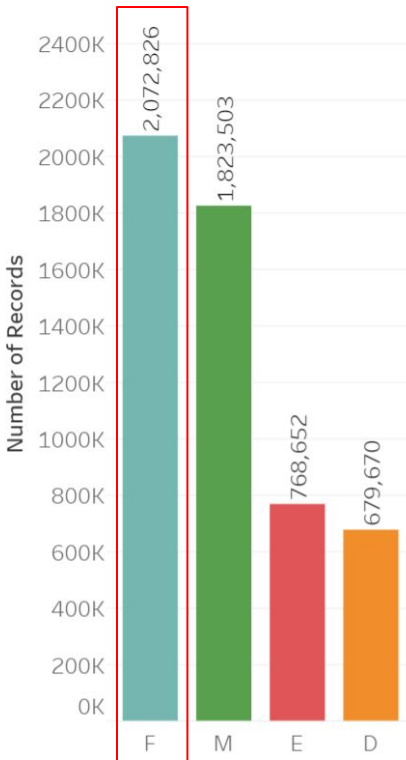
- BRONX
- BROOKLYN
- MANHATTAN
- QUEENS
- STATEN ISLAND

YEAR(Complaint Start...

- ☒ (All)
- ☒ 2006
- ☒ 2007
- ☒ 2008
- ☒ 2009
- ☒ 2010
- ☒ 2011
- ☒ 2012
- ☒ 2013
- ☒ 2014
- ☒ 2015
- ☒ 2016
- ☒ 2017

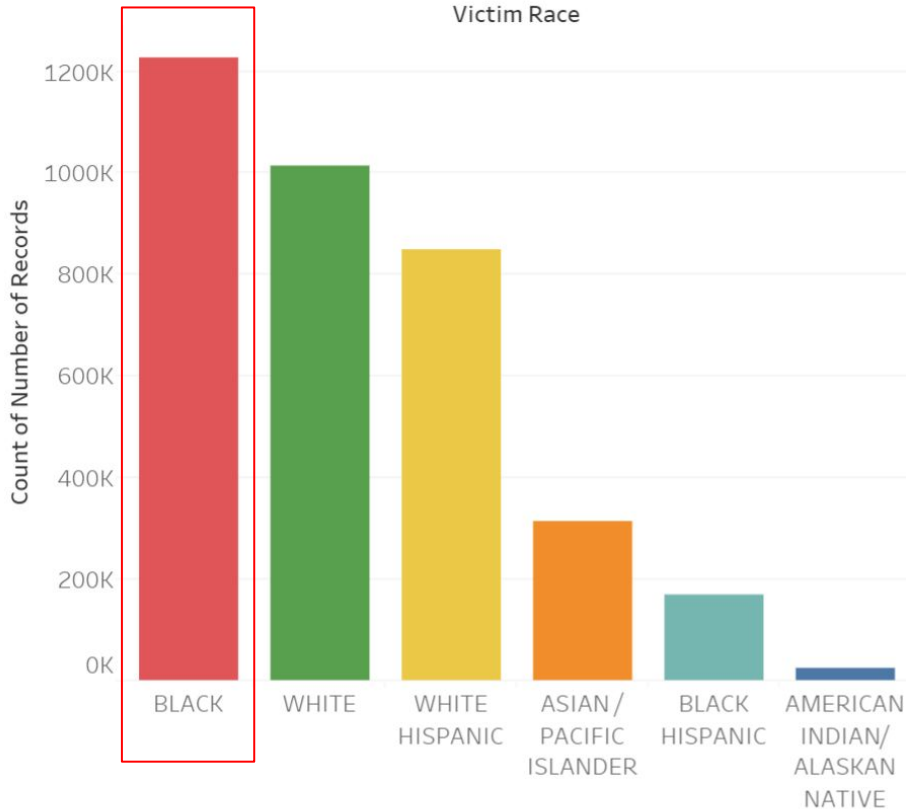


Victim Sex



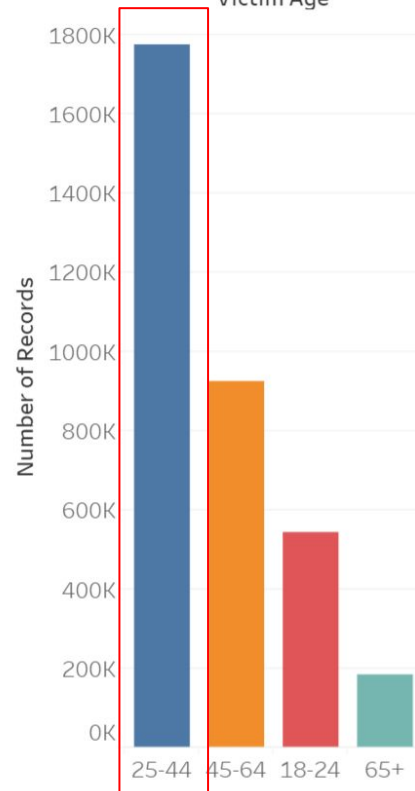
Sum of Number of Records for each Victim Sex. Color shows details about Victim Sex. The view is filtered on Victim Sex, which keeps D, E, F and M.

Victim Race



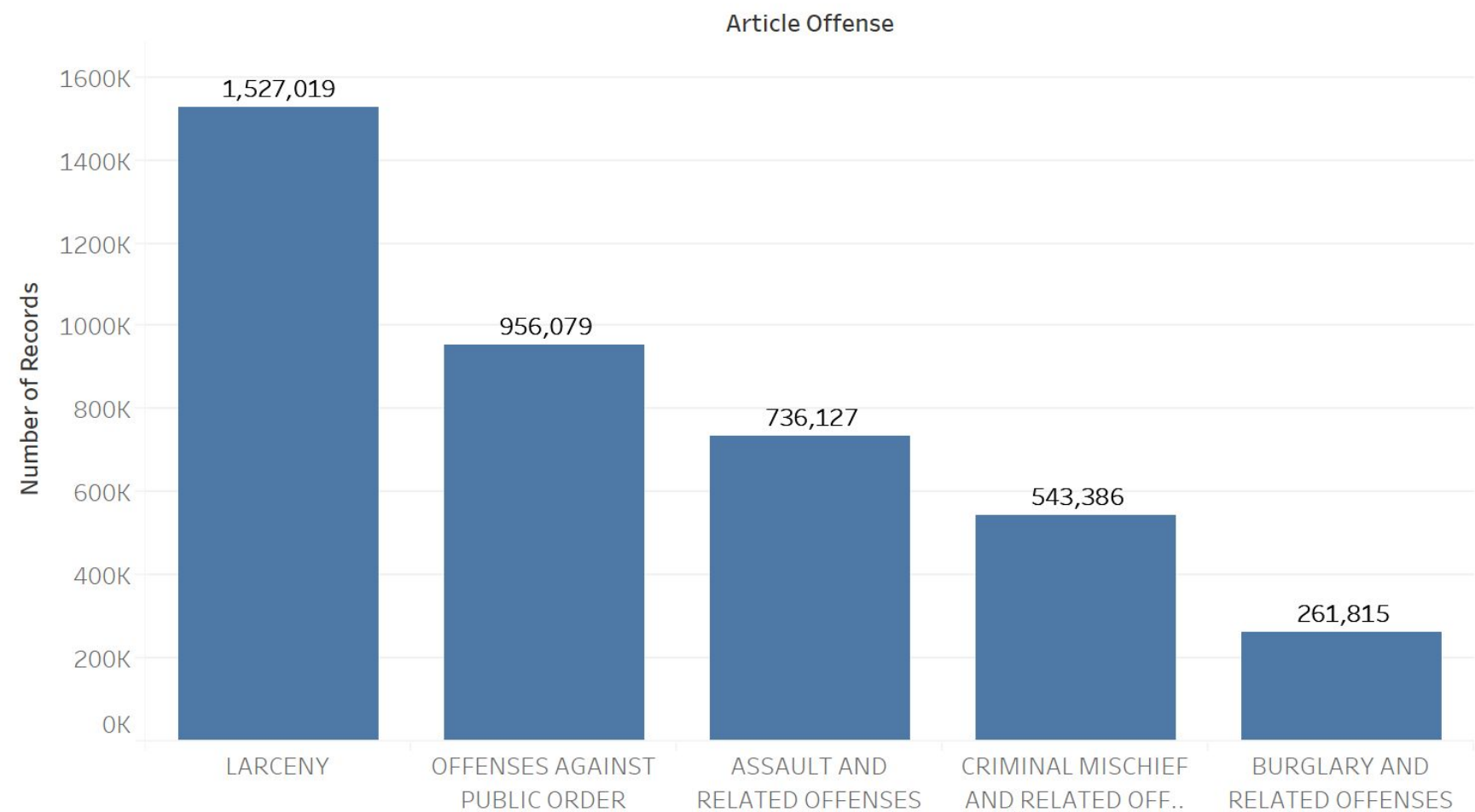
Count of Number of Records for each Victim Race. Color shows details about Victim Race. The view is filtered on Victim Race, which excludes Null, OTHER and UNKNOWN.

Victim Age



Sum of Number of Records for each Victim Age. Color shows details about Victim Age. The view is filtered on Victim Age, which keeps 18-24, 25-44, 45-64 and 65+.

TOP5_ArticleOffenses



CrimeMap_All



Borough

BRONX

BROOKLYN

MANHATTAN

QUEENS

STATEN ISLAND

Year

(All)

2006

2007

2008

2009

☒ 2010

2011

2012

2013

2014

2015

2016

2017

CNT(Article Offense)

20

40

60

80

100

123

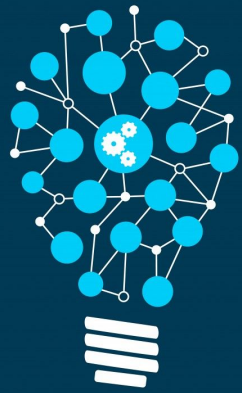
Machine Learning

Model target:

predict total number of crimes in specific location

Model used : multivariable linear regression

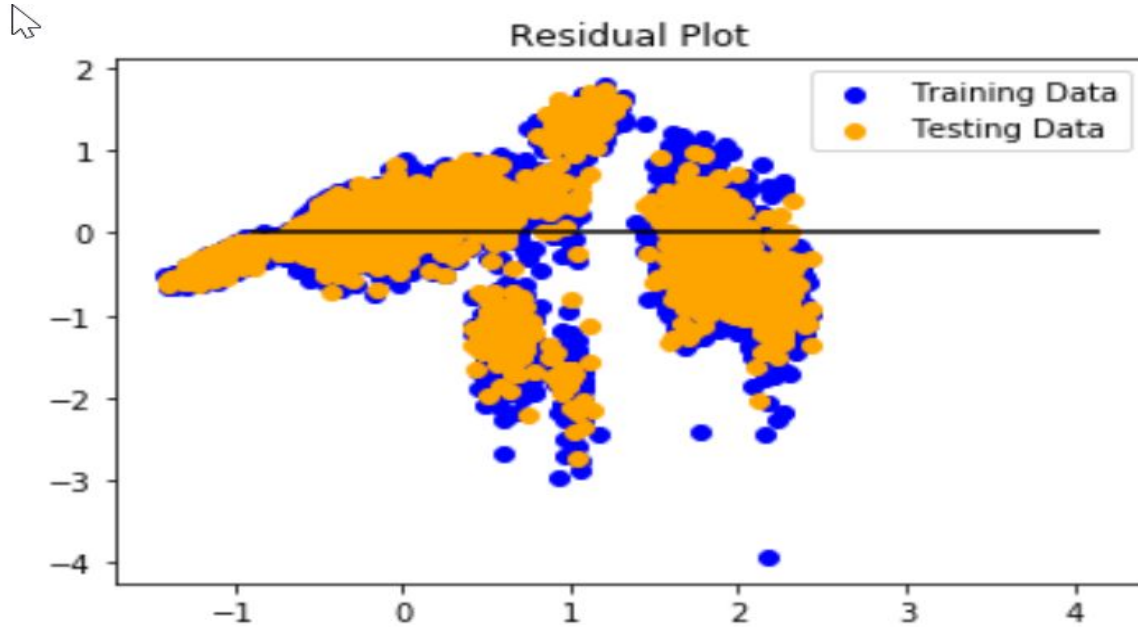
MACHINE
LEARNING



Parameters used for prediction

- Seasonal effect : month of the year
- Decreasing trend : year
- Weekday
- Holidays
- Location - borough
- Weather

Model analysis



MSE: 0.19444144488396295, R2: 0.7984829980163528

App Deployment

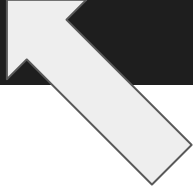
App goal: Take user's input, and use the input to predict total crime

Deployment method: **Flask & Flask-WTF, Heroku**

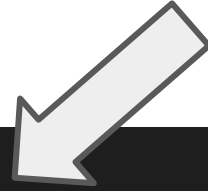


Load the model

```
def load_model():  
    global model  
    model = joblib.load(os.path.join('Model', 'finalized_model.sav'))  
  
load_model()
```



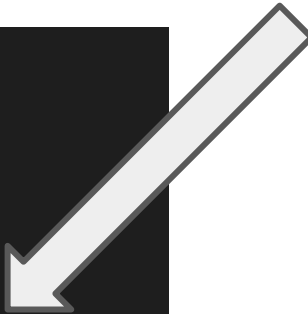
Creating Form Parameters



```
class InputForm(FlaskForm):
    temperature_high = IntegerField("Temperature High (F)", [validators.DataRequired()])
    temperature_low = IntegerField("Temperature Low (F)", [validators.DataRequired()])
    zipcode = IntegerField("Zipcode", [validators.DataRequired()])
    day = SelectField(u'Day of Month', choices=[('1', '1'), ('2', '2'), ('3', '3'), ('4', '4'), ('5', '5'), ('6', '6'), ('7', '7'), ('8', '8'), ('9', '9'), ('10', '10'), ('11', '11'), ('12', '12')])
    weekday = SelectField(u'Day of Week', choices=[('Sunday', 'Sunday'), ('Monday', 'Monday'), ('Tuesday', 'Tuesday'), ('Wednesday', 'Wednesday'), ('Thursday', 'Thursday'), ('Friday', 'Friday'), ('Saturday', 'Saturday')])
    month = SelectField(u'Month', choices=[('1', 'January'), ('2', 'February'), ('3', 'March'), ('4', 'April'), ('5', 'May'), ('6', 'June'), ('7', 'July'), ('8', 'August'), ('9', 'September'), ('10', 'October'), ('11', 'November'), ('12', 'December')])
    holiday = SelectField(u'Holiday', choices=[('Yes', 'Yes'), ('No', 'No')])
    submit = SubmitField("Submit")
```

Submit.html Template

```
<form action="" method="post" novalidate>
  <p>
    {{ csrf_token() }}
    {{ form.hidden_tag() }}
  </p>
  <p>
    {{ form.temperature_high.label }}<br>
    {{ form.temperature_high(size=32) }}
    {% for error in form.temperature_high.errors %}
    <span style="color: red;">[{{ error }}]</span>
    {% endfor %}
  </p>
  <p>
    {{ form.temperature_low.label }}<br>
    {{ form.temperature_low(size=32) }}
    {% for error in form.temperature_low.errors %}
    <span style="color: red;">[{{ error }}]</span>
    {% endfor %}
  </p>
</form>
```



Temperature High (F)

[This field is required.]

Temperature Low (F)

[This field is required.]

Zipcode

[This field is required.]

Required

Day of Month

Day of Week

Month

Holiday

Submit

Default

If else, return template

```
@app.route('/input', methods=['GET', 'POST'])
def input():
    form = InputForm()
    if form.validate_on_submit():
        xxxxxxxxx
        return render_template('results.html', form=form, results=results, predict=predicted_results)

    if not form.validate_on_submit():
        flash(form.errors)

    return render_template('submit.html', form=form)
```



NYC Crime: [Input](#)

• {}

Input Info

ImFhZWM5OWNhNmEwNGNmYzA3MGVINmVjMjVmNjhhNmIyOWNhNDIxMDYi.XNHRmQ.IyvSdYMLQY2_tlVA4jnwbtO60Xw

Temperature High (F)

Temperature Low (F)

Zipcode

Day of Month

Day of Week

Month

Holiday

Submit

Temperature High (F)

40

Temperature Low (F)

23

Zipcode

10474

Day of Month

16 ▼

Day of Week

Tuesday ▼

Month

March ▼

Holiday

No ▼

Submit

ImFhZWM5OWNhNmEwNGNmYzA3MGVINmVjMjVmNjhNmlIyOWNhNDIxMDYi.XNRUfw.-ZFyuVA9nVzvGhy-wkVoDRo8pyE

Results:

Temperature High:40

Temperature Low:23

Zipcode:10474

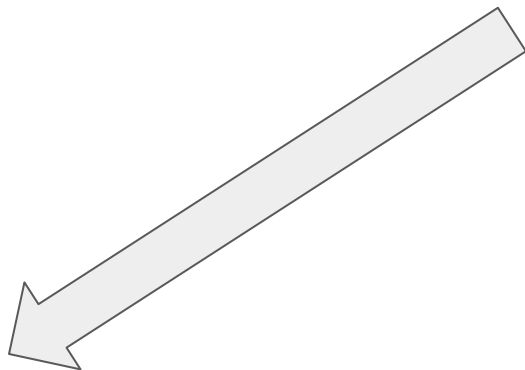
Day of Month:16

Day of Week:Tuesday

Month:3

Holiday:No

Predicted:[[44.07654169]]



Heroku Deployment Issues

```
from flask_wtf.csrf import CSRFProtect  
  
csrf = CSRFProtect(app)
```

```
csrf = CSRFProtect()  
  
def create_app():  
    app = Flask(__name__)  
    csrf.init_app(app)
```

The change you wanted was rejected.

Maybe you tried to change something you didn't have access to.

If you are the application owner check the logs for more information.

CSRF = Cross-site Request Forgery

Sources

- NYC crime data (NYC Open Data):
<https://data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Historic/qgea-i56i>
- NYC weather data (National Oceanic and Atmospheric Administration):
<https://www.ncdc.noaa.gov/cdo-web/search;jsessionid=7AB6BB0B7A386F5775DAF0896B5D355E>
- Real-world machine learning (Wired):
<https://www.wired.com/insights/2013/08/predictive-policing-using-machine-learning-to-detect-patterns-of-crime/>
- NYC law (New York State Law): <http://ypdcrime.com/penal.law/>