The Crime Detectives

- **Natasha Arokium**
- **Jiffar Abakoyas**
- **Bethold Owusu**
- **Thierno Diallo**

NYPD arrest data Analysis 2019

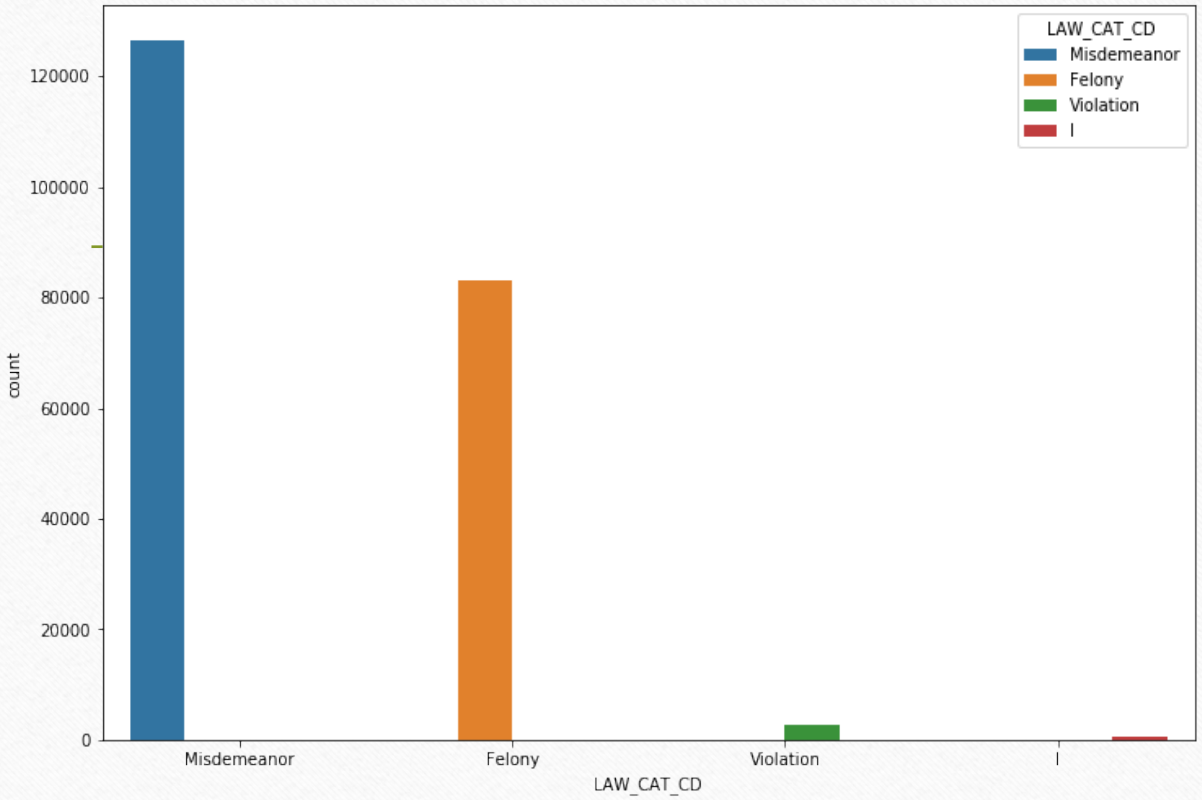
- NYC has been ranked #25 with the most violent crime rates in the United States.
- In this project we attempt to classify NYC arrests by predicting their level of severity (Felony or Misdemeanor) using other known features from the arrest.

NYPD made 213,089 arrests in 2019

Principal features for analysis

- Level of Offense (target feature)
- Borough of arrest
- Precinct where the arrest occurred
- Perpetrator's sex description
- Perpetrators' race description
- Date of the arrest
- Most frequently occurring arrests/locations on the map/month

Arrests by Level of Crime

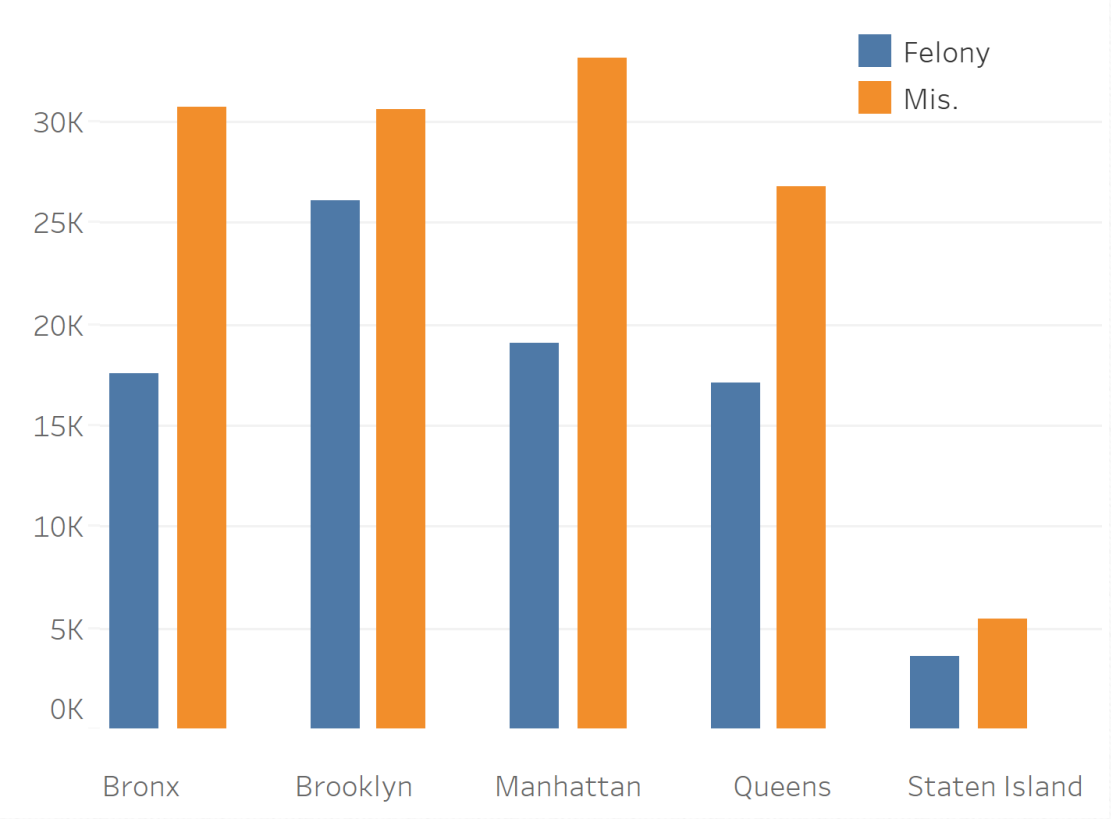


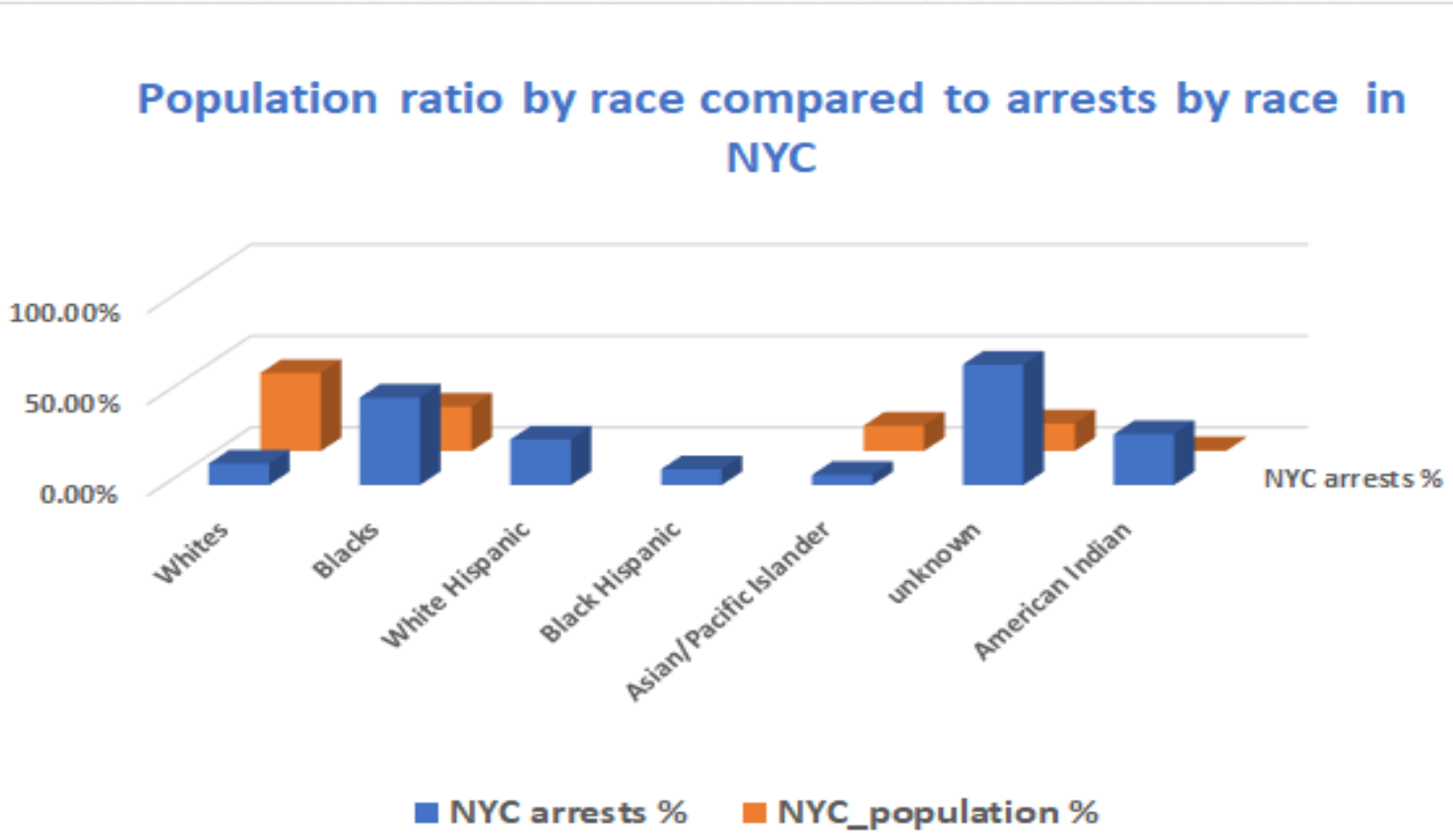
Level Of Arrest	Count	Percent
Misdemeanor	126590	59.41%
Felony	83244	39.07%
Violation	2822	1.32%
I	433	0.2%

Most frequently occurring arrests by description

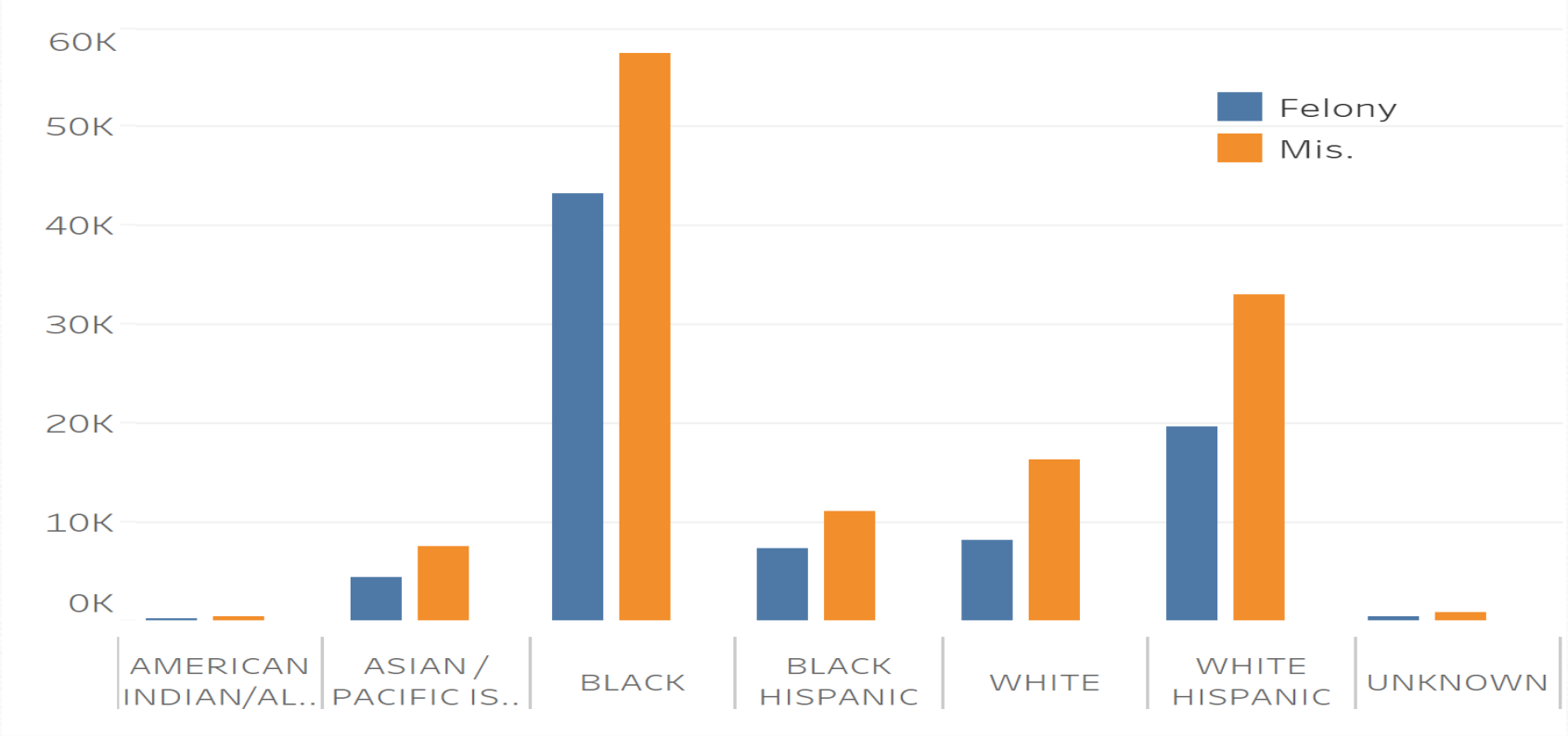
ARREST DESCRIPTION	COUNTS	PERCENT
ASSAULT 3 & RELATED OFFENSES	31988	15.01%
PETIT LARCENY	21627	10.15%
DANGEROUS DRUGS	21071	9.89%
FELONY ASSAULT	15208	7.14%
VEHICLE AND TRAFFIC LAWS	14243	6.68%
...	...	...
HOMICIDE-NEGLIGENT-VEHICLE	5	0.0%
PARKING OFFENSES	3	0.0%
NYS LAWS-UNCLASSIFIED VIOLATION	2	0.0%
LOITERING FOR DRUG PURPOSES	2	0.0%
UNLAWFUL POSS. WEAP. ON SCHOOL	1	0.0%

Arrests by Borough

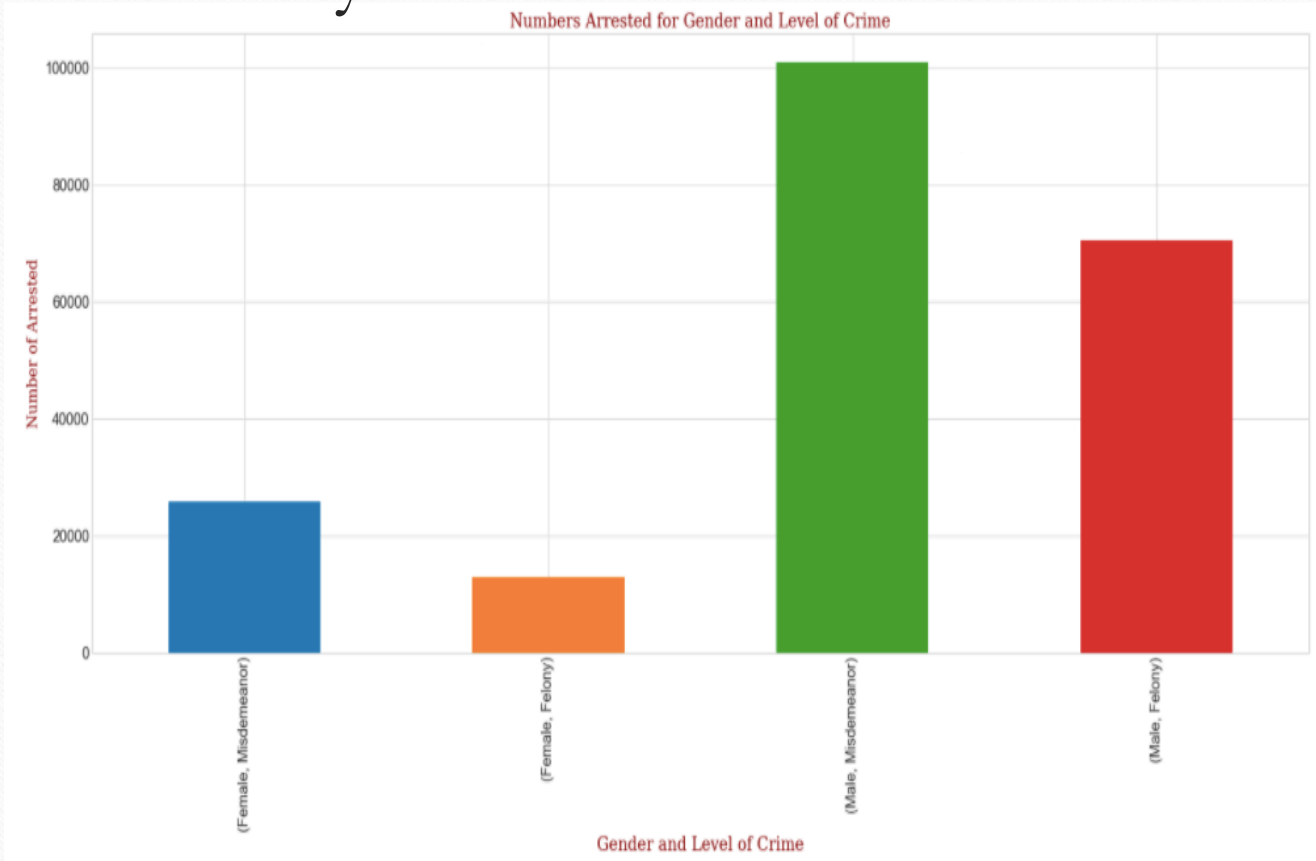



Population ratio by race compared to arrests by race in NYC
100.00%
50.00%
0.00%
Whites
Blacks
White Hispanic
Black Hispanic
Asian/Pacific Islander
unknown
American Indian
NYC arrests %
NYC arrests %
NYC_population %

Arrests by Race & Level of Crime

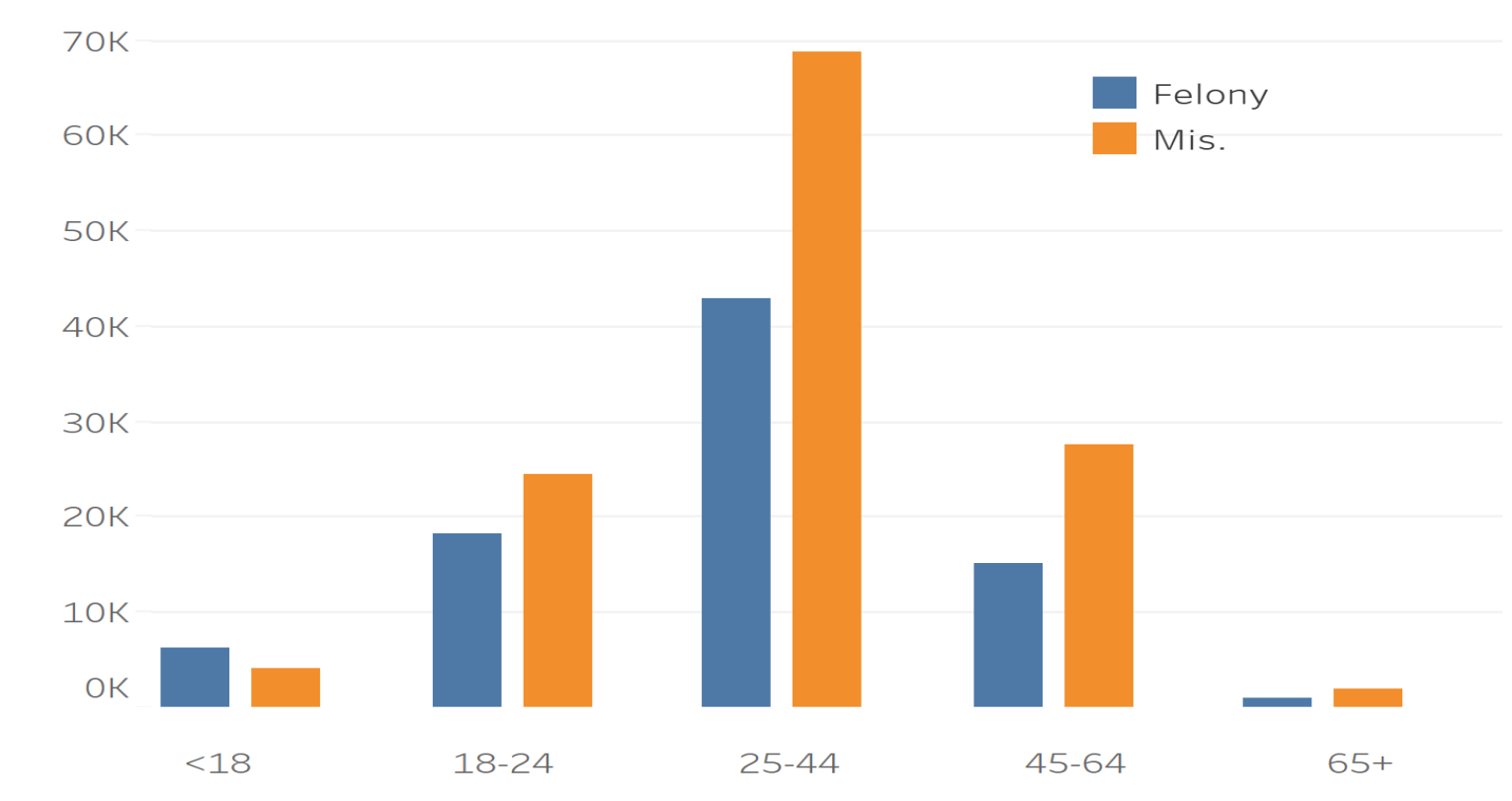


Arrests by Gender

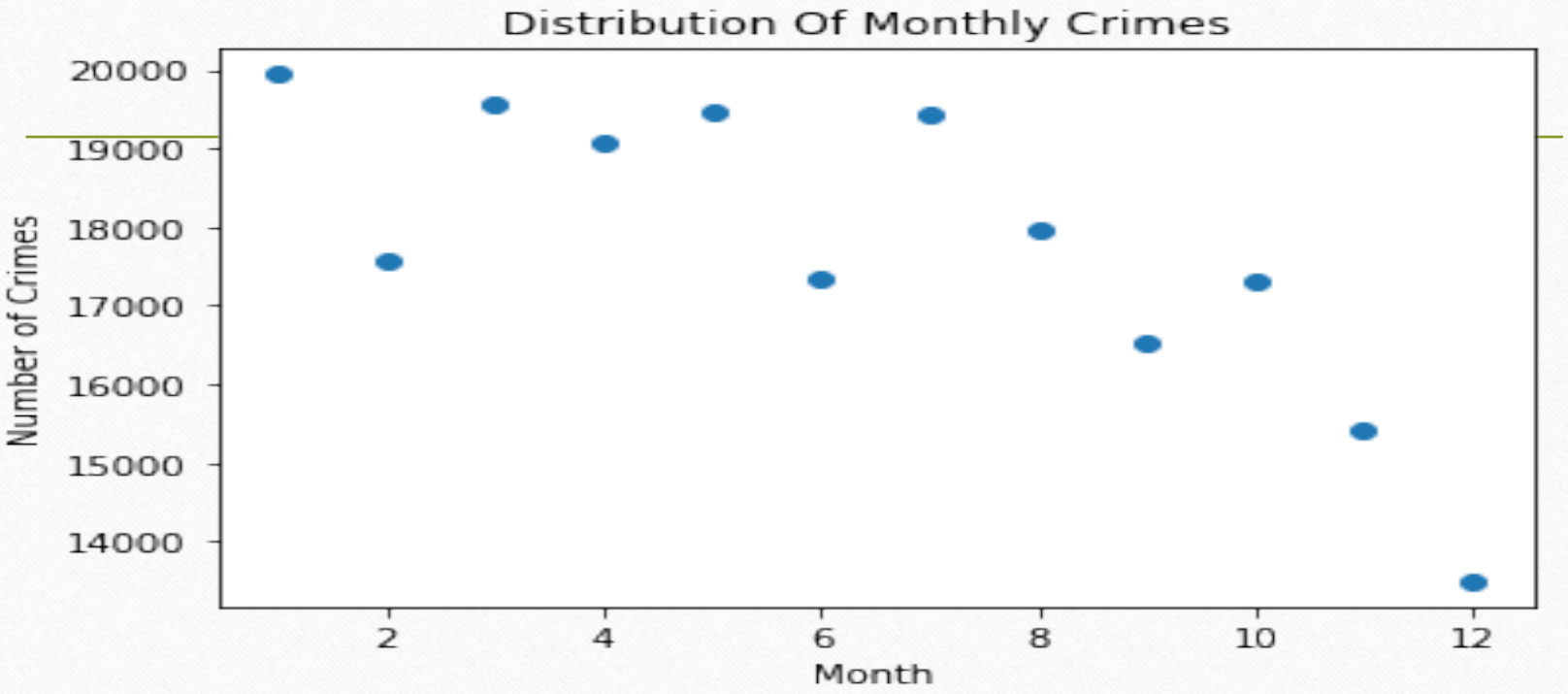


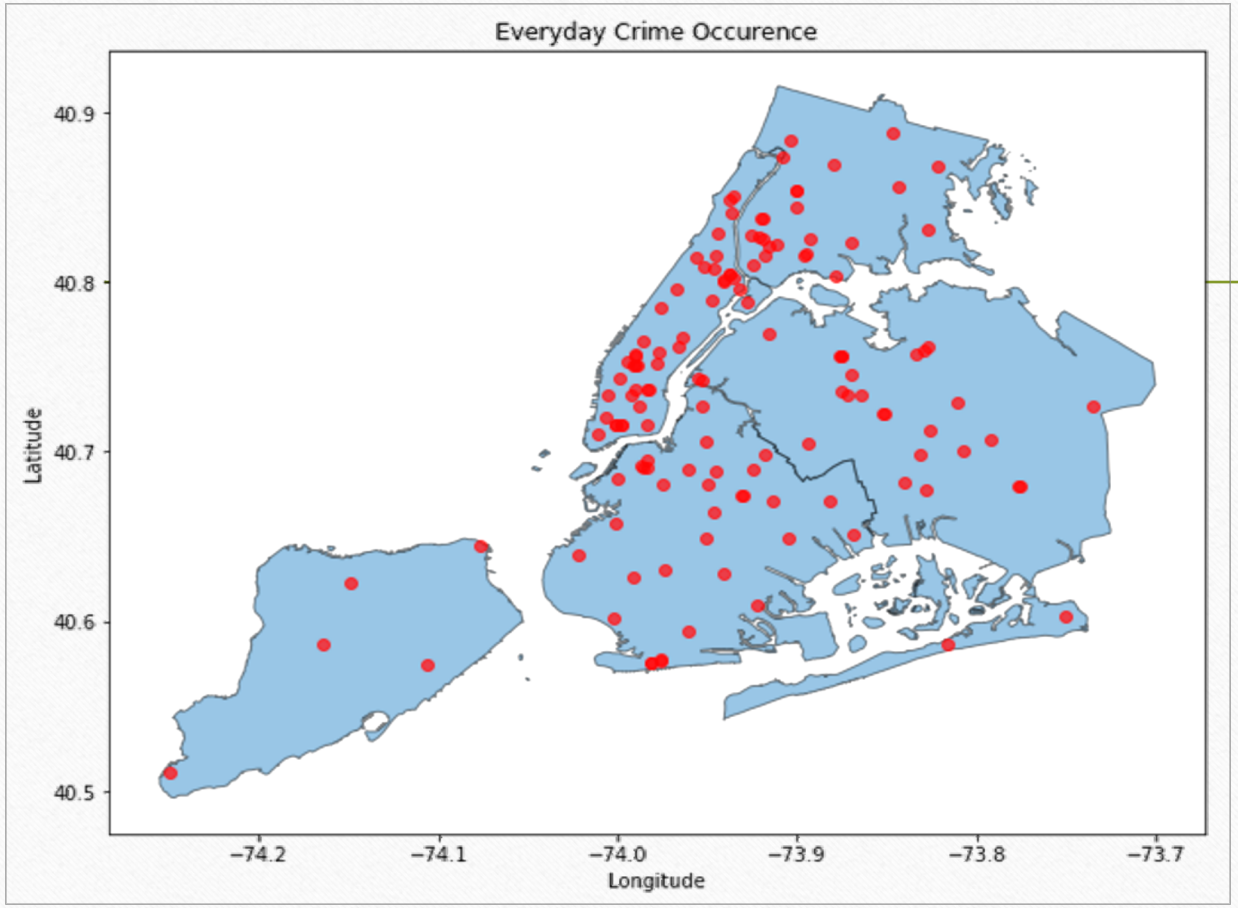
SEX	COUNT	PERCENT
M	173964	81.64%
F	39125	18.36%

Arrests by Age & Type of Crime



Arrests by month





Algorithm to see where crimes occur daily

- ALBEE SQUARE (BK)
- QUEENS BOULEVARD EXPRESSWAY
- ROCKEFELLER CENTER
- FULTON MALL (BK)
- GATEWAY DR
- NOSTRAND SUBWAY STATION
- JFK AIRPORT
- TRANS MANHATTAN EXP
- BARTOW AVE (COOP CITY MALL)

One Hot Encoding:

Because the numeric columns contain internal NYPD codes and are not measurements, for our model every column is treated as nominal categorical for our classification (i.e: there is no ordering to the values)

```
#Create one hot encoded data
#Creating a one-hot encoding that assumes all numeric columns are numeric (only encodes string col
umns)
target_removed = data.loc[:, data.columns != 'LEVEL_OF_OFFENSE']
X = pd.get_dummies(target_removed,drop_first=True)
target = data.loc[:,'LEVEL_OF_OFFENSE']
X.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 209834 entries, 0 to 214616
Columns: 126 entries, ARREST_BORO_Brooklyn to ARREST_MONTH_12
dtypes: uint8(126)
memory usage: 26.8 MB
```

Dummy Classifier

Dummy classifier Validation accuracy: 0.5235896824924048

	precision	recall	f1-score	support
Felony	0.40	0.40	0.40	13447
Misdemeanor	0.60	0.61	0.60	20127
avg / total	0.52	0.52	0.52	33574

Logistic Classifier

Logistic Validation accuracy: 0.6207481980103652

	precision	recall	f1-score	support
Felony	0.57	0.21	0.31	13447
Misdemeanor	0.63	0.89	0.74	20127
avg / total	0.61	0.62	0.57	33574

Logistic with PCA (45 components)

Logistic Validation accuracy: 0.6174718532197534

	precision	recall	f1-score	support
Felony	0.57	0.18	0.27	13447
Misdemeanor	0.62	0.91	0.74	20127
avg / total	0.60	0.62	0.55	33574

Grid Search Best Parameters

- 5fold Cross-val mean: 0.62686, std: 0.00273,
- params: {'class_weight': None, 'multi_class': 'multinomial', 'solver': 'newton-cg'}

- The above were grid search best parameters for logistic regression with different parameters. The performance did not increase significantly from initial results.

Support Vector Machine

```
#Print validation classification report
print(classification_report(y_val, svc_y_pred_val))
```

```
             precision    recall  f1-score   support

     Felony       0.57      0.21      0.31     13447
Misdemeanor       0.63      0.89      0.74     20127

avg / total       0.61      0.62      0.57     33574
```

SVM performance was almost identical to that of Logistic Regression for all measures: precision, recall and f1-score.

Decision Tree

```
#Print validation classification report
print(classification_report(y_val, decision_tree_y_pred_val))
```

	precision	recall	f1-score	support
Felony	0.50	0.44	0.46	13447
Misdemeanor	0.65	0.70	0.68	20127
avg / total	0.59	0.60	0.59	33574

Random Forest

```
#Print validation classification report
print(classification_report(y_val, rforest_y_pred_val))
```

	precision	recall	f1-score	support
Felony	0.57	0.21	0.31	13447
Misdemeanor	0.63	0.89	0.74	20127
avg / total	0.61	0.62	0.57	33574

Both Decision tree and Random Forest models performed similarly to Logistic Regression and/or SVM. Although Random Forest did show significant increase in misdemeanor recall compared to Decision Tree, however it performed worse in recall for felony class.

Random Forest Feature Importance

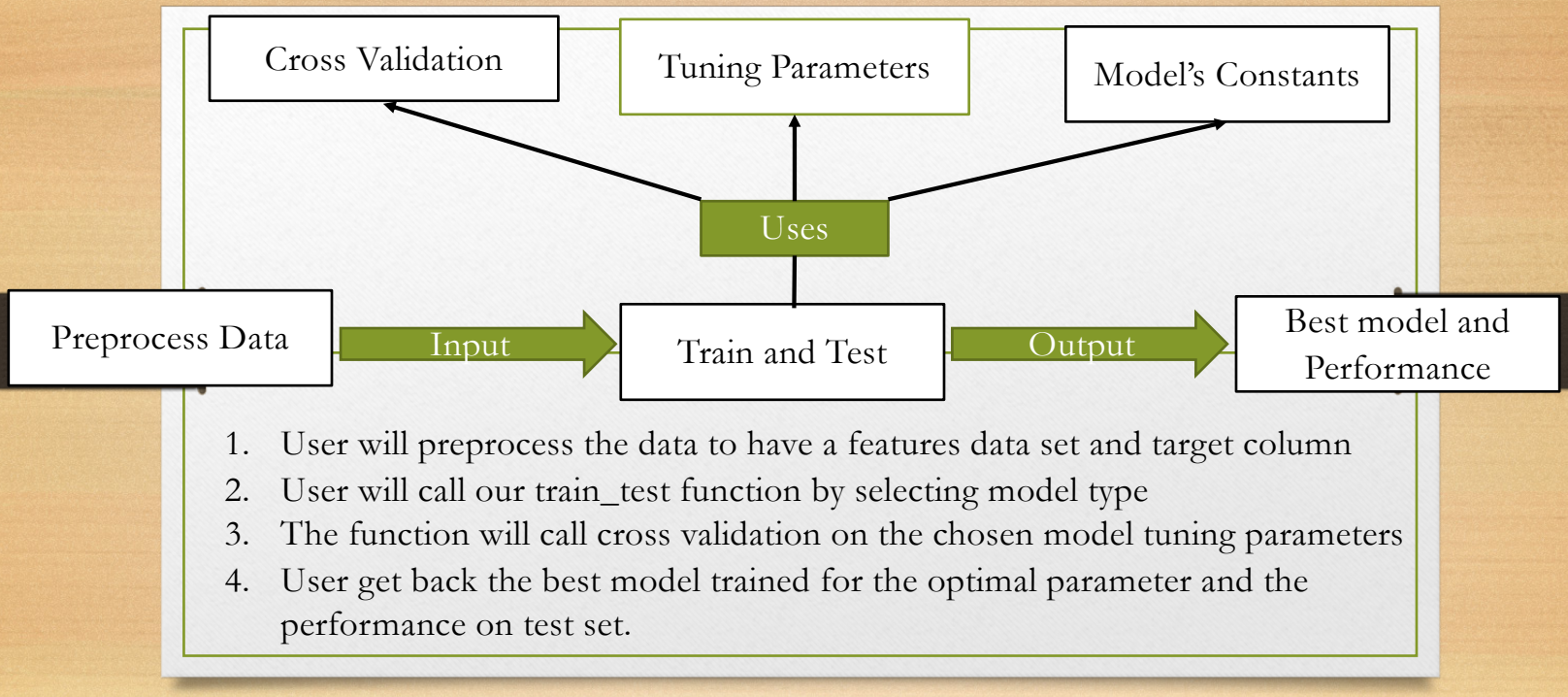
Using RandomForest for Feature Importance Selection

```
]: #Create feature labels list
feat_labels = X_train.columns
#Take the feature importance list from the random forest object that was just used
importances = rforest.feature_importances_
# Store the indices to use for ranking
indices = np.argsort(importances)[::-1]
#Select from the model best features
sfm = SelectFromModel(rforest, threshold=0.01, prefit=True)
X_selected = sfm.transform(X_train)
print('Number of features that meet this threshold','criterion:', X_selected.shape[1])
for f in range(X_selected.shape[1]):
    print("%2d) %-*s %f" % (f + 1, 30, feat_labels[indices[f]],importances[indices[f]]))
```

```
Number of features that meet this threshold criterion: 12
 1) PERP_SEX_Male                  0.067346
 2) AGE_GROUP_25-44                0.061832
 3) AGE_GROUP_45-64                0.041888
 4) AGE_GROUP_<18                  0.018202
 5) JURISDICTION_CODE_2            0.013454
 6) ARREST_BORO_Brooklyn           0.013445
 7) JURISDICTION_CODE_1            0.013123
 8) ARREST_PRECINCT_115            0.012728
 9) AGE_GROUP_65+                  0.011207
10) JURISDICTION_CODE_97           0.010741
11) ARREST_PRECINCT_72             0.010505
12) ARREST_BORO_Queens             0.010377
```

Using Random Forest and the feature importance based on information gain, were able to confirm that features of importance included being male and being in specific age groups. The arrest borough was also a feature of importance, as well as the arresting precinct.

Our ML Classifier Workflow



1. User will preprocess the data to have a features data set and target column
2. User will call our train_test function by selecting model type
3. The function will call cross validation on the chosen model tuning parameters
4. User get back the best model trained for the optimal parameter and the performance on test set.

Sample Use of the Classifier

main.py | libs.py | main_notebook.ipynb | model_constants.py | Initial_models.ipynb | NYPD_Arrest_Analysis-checkpoint.ipynb

```
1  import training_testing, data_preprocess
2
3  df, y_column = data_preprocess.data_preprocess1()
4
5  model, model_perf = training_testing.train_and_test(df, y_column, 'Logistic Regression', 0.3)
6
7  print(model_perf)
8
```

Call the preprocess function

Call the train test function

Machine Learning Takeaway

1. All the features (Race, Age Group, gender, Borough) we looked at were not good predictor of the target variable (type of crime)
2. All the model we tried gave us precisions below 0.70
3. Our Dataset was not a good fit for machine learning

Project Stats

April 12, 2020 – May 12, 2020

Period: 1 month

Overview

35 Active Pull Requests

0 Active Issues

34	1	0	0
Merged Pull Requests	Proposed Pull Request	Closed Issues	New Issues

Excluding merges, **4 authors** have pushed **44 commits** to master and **56 commits** to all branches. On master, **0 files** have changed and there have been **0 additions** and **0 deletions**.

34 Pull requests merged by **4** people

Contribution Statistics

Apr 19, 2020 – May 12, 2020

Contributions to master, excluding merge commits