CSS444 Final Analysis: Modeling NYPD STOPS

Jericho Timbol

03/09/2023

Introduction

Bias and fairness are two essential factors to consider when using AI and ML in any context. Predictions or assumptions made by these systems can heavily range in the severity of harm it can cause. The model and systems in programming can be designed in such a way that does not promote existing biases, more importantly, the accuracy and determinations made are only as accurate as the dataset provided. In this analysis, me and my team worked together to analyze the New York Police Department 2010 Stops dataset in order to explore the question: Are there any biases in NYPD stops based on the ethnicity of the person in question?

When it comes to fairness and bias, this topic resides in a context of life changing severity. Whether you get arrested, searched, or frisked, situations involving police are often high pressure and high risk. Observing patterns of action taken by the NYPD as a whole serve as an insight that could lead to positive changes in stop procedures no matter the suspect. Based on this dataset, we decided to implement a Binary Classification Decision Tree model. In this binary classification problem, we used a decision tree model in order to predict outcomes of stops when looking at numerous features (most notably race). Microsoft's "AI Fairness Checklist" states that "AI systems can behave unfairly because of societal biases reflected in the datasets used to trained them." (Madaio et al). As we will find later, the existing racial biases in our dataset, are reflected in the predictions our model made. While we tested and optimized for fairness metrics such as Equal Opportunity, and Relaxation of Separation, we can still see the existing biases produced by the result set between classes.

Methodology

As our team approached this problem, we, we first familiarized ourself with the NYPD Stops dataset we were working with. With over 114 columns of data, our first task was to preprocess and clean the data whilst creating DataFrames that would serve as possible features of comparison and statistical analysis. First and foremost, I looked to detail the count and distribution of stops between the different races. The snippet below summarizes my processing and creation of a suitable column to observe:

```
racedf = rawdata
racedf['race'] = rawdata['race'].replace(['B', 'P','Q', 'W', 'Z','A', 'U', 'I'],
           ['Black', 'Black', 'White-Hispanic', 'White', 'Other','Asian', 'Unknown',
'American Indian/Alaskan Native'])
racedf.drop(racedf[(racedf['race'] == 'Other') | (racedf['race'] == 'Unknown') |
(racedf['race'] == 'American Indian/Alaskan Native')].index,inplace= True  )
```

Figure 1

Cleaning 'race' column in the dataset

Looking at the unique values of the column, I renamed them from their keyword terms in the dataset to understandable names. Not only that, but in our entire analysis, we grouped the 'Black-Hispanic' and 'Black' race values into the same value of 'Black'. Due to sample size issues, and ambiguity, we determined it was best to omit the race values of 'Other', 'Unknown' and 'American Indian/Alaskan Native'. We made this decision in order to minimize underfitting and overfitting issues in the decision tree. Too few samples will result in underfitting. Additionally, overfitting could occur for these classes as there are too few representatives that would occur in the training set.

Additionally, we took a look at income data related to these stops. Wealth is an interesting comparison in analyzing how different financial classes are treated at a stop. Daniel took the time to find data for per capita income and median income DataFrame in relation to precinct data. By creating a separate csv and merging it on the precinct value, we were able to make observations for these income-based stops in our original dataset. The snippet below displays the creation of such DataFrame in which we proceeded to statistically analyze income and race with stop outcomes:

```
merged_df = pd.merge(rawdata, incdata[['pct', 'perCapInc','medHouseInc']], on='pct')
rawdata['perCapInc'] = merged_df['perCapInc']
rawdata['medHouseInc'] = merged_df['medHouseInc']
```

Figure 2

Creation and merging of 'perCapInc' and 'medHouseInc' to original dataset (rawdata)

Another key formatting issue was the columns involving the use of force. With 9 different types of force used columns with T/F values, it was hard to get a picture of using force as an entirety. That being said, we decided to combine them into a succinct column in the original data labeled 'forceUsed' representing any type of force used. These columns involve force by: hand, weapon, baton, draw weapon, handcuffs, pepper spray, ground physicality, wall physicality, and others. The addition of this column is shown below.

```
rawdata['usedForce'] = 'N'
rawdata.loc[(rawdata['pf_hands'] == 'Y') | (rawdata['pf_wall'] == 'Y') |
(rawdata['pf_grnd'] == 'Y') |
           (rawdata['pf_drwep'] == 'Y') |(rawdata['pf_baton'] == 'Y') |
(rawdata['pf_ptwep'] == 'Y') |
           (rawdata['pf_hcuff'] == 'Y') |(rawdata['pf_pepsp'] == 'Y')
|(rawdata['pf_other'] == 'Y'), 'usedForce'] = 'Y'
```

Figure 3

Creation of usedForce column based on any force used.

The last set of analysis and observation took into consideration the ‘usedForce’ column in relation to race and city. By looking at the distribution of force used in cities, we can begin to exam possible location-based stop patterns. This is taken a step further when compared to the value counts of race involvement within these cities. Cleaning the city column involved removing the unique ‘ ‘ blank value residing within the dataset. Python scripts are shown in figure 4.

```
city_ten_df = racedf[racedf.city != ' ' ]

print(pd.concat(
[

    city_ten_df.groupby('city')["race"].value_counts(),
    city_ten_df.groupby('city')["race"].value_counts(normalize= True),

],
keys=['Force used counts', 'Force used normalized'],
axis=1,
))
```

Figure 4

DataFrame analysis of force used by city and race

After all of the analysis on data it was time to create our model based on our decided features. For our target feature, we chose to analyze ‘arstmade’ (arrest made) as our predicted value. For our features we chose:

precinct, frisked, searched, pistol, usedForce, reason for stop: fits description, reason for stop: violent crime, reason for stop: suspicious bulge, sex, race, age and medHouseInc.

With our target set and features chosen. It was time to use our 2012.csv rather than the 2010.csv we analyzed. Using this sample as the training data helps to prevent overfitting and show the system new

values. We chose to test with the first 100,000 data points. In figure 5 the snippet of our sklearn decision tree implementation is provided.

```
features =
['pct','frisked','searched','pistol','usedForce','cs_descr','sex','race','age','cs_vcr
im','cs_bulge','medHouseInc']
```

```
X = pd.get_dummies(datadf)
X_test = pd.get_dummies(testdf)
Y = rawdata['arstmade']
yTestTrue = rawtest['arstmade']
dTree = tree.DecisionTreeClassifier(max_depth=10)
dTree.fit(X,Y)
print("done making tree")
y_pred = dTree.predict(X_test)
output = pd.DataFrame({'serNum': rawtest.ser_num,'arrestMade': y_pred})
print(output)

treeAcc = round(dTree.score(X,Y) * 100,2)
print(treeAcc)
```

Figure 5

Creation of decision tree using arrest made and selected features and printing predication set accuracy.

After the creation of the decision tree, we tested our model against several metrics. The first metric was the classification report involving precision, recall, f1-score, support, ROC curve and AUC score. In order to test fairness, we chose to look at Equal Opportunity: FN/ (FN + TP). This gives us the False Negative Rate among the races. For our case it means that a person was not arrested but the model predicted them to be arrested. Looking at Relaxation of Separation as our other fairness benchmark, we calculated the False Positive Rate of each class. This means the person was arrested but the model predicted them to not be arrested.

Results

The results of our methodology can be regarded in three main categories. Dataset Analysis, Decision Tree Modeling, and Fairness. We will look at some of the outputs in the queries and scripts presented earlier

Data Analysis Results

The first round of analysis considered race. In figure 6, the first graph shows value counts of stops by race. To go one step further we plotted the value counts of race stops where force was used in figure 7.

	Stop Count	Stop Count Normalized
Black	353772	0.611057
White-Hispanic	150637	0.260190
White	54810	0.094671
Asian	19732	0.034082

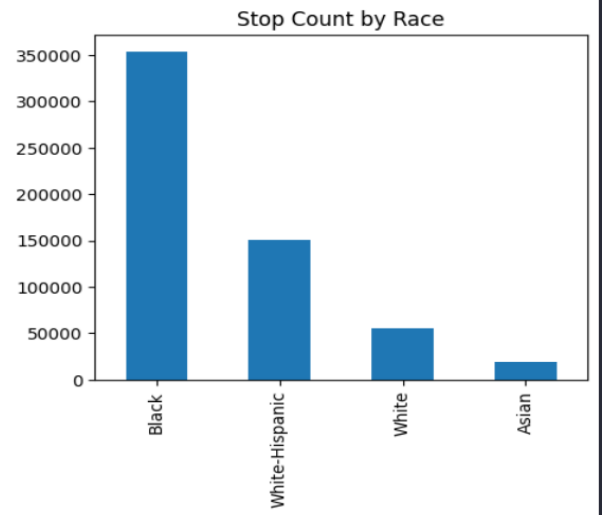

Figure 6

Output of value_counts() displayed in DF and Plot

	Force Count	Force Count Normalized
Black	83125	0.615964
White-Hispanic	39272	0.291009
White	9191	0.068106
Asian	3363	0.024920

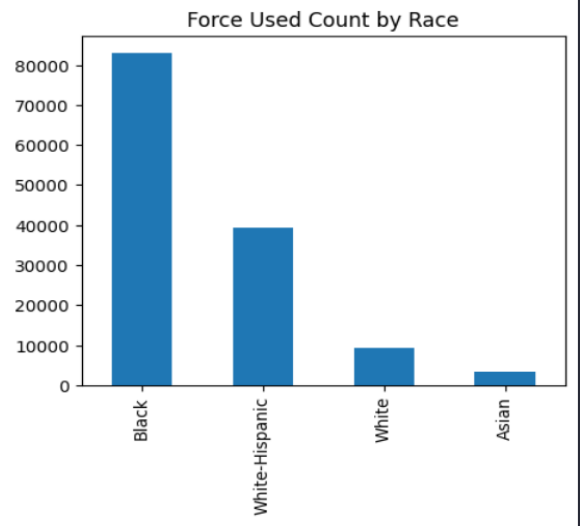

Figure 7

Output of value_counts() 'usedForce' is True

Next, we will look at how per capita income affects racial stops. Figure 8 will contain high income 'perCapInc' which is greater than or equal to $150,000. In figure 9, low income is set to less than or equal to $25,000

```
Total Number of Black/African-American Stops:  17565
Percentage Searched or Frisked: 0.5902077996014802
Percentage Arrested or Summoned to court: 0.12183319100483916
Percentage of Black/African-American in Dataset:  0.6005128205128205

Total Number of White Stops:  2540
Percentage Searched or Frisked: 0.4519685039370079
Percentage Arrested or Summoned to court: 0.1283464566929134
Percentage of White in Dataset:  0.08683760683760684

Total Number of Asian Stops:  984
Percentage Searched or Frisked: 0.48272357723577236
Percentage Arrested or Summoned to court: 0.13516260162601626
Percentage of Asian in Dataset:  0.03364102564102564

Total Number of Hispanic Stops:  7031
Percentage Searched or Frisked: 0.6040392547290571
Percentage Arrested or Summoned to court: 0.13796046081638458
Percentage of Hispanic in Dataset:  0.24037606837606837
```

Figure 8

High-income group statistics

```
Total Number of Black/African-American Stops:  23910
Percentage Searched or Frisked: 0.5980761187787537
Percentage Arrested or Summoned to court: 0.13304056879966542
Percentage of Black/African-American in Dataset:  0.5922568180129301

Total Number of White Stops:  3707
Percentage Searched or Frisked: 0.44807121661721067
Percentage Arrested or Summoned to court: 0.12489884003237119
Percentage of White in Dataset:  0.09182333853508706

Total Number of Asian Stops:  1298
Percentage Searched or Frisked: 0.4953775038520801
Percentage Arrested or Summoned to court: 0.13559322033898305
Percentage of Asian in Dataset:  0.03215179212801268

Total Number of Hispanic Stops:  10040
Percentage Searched or Frisked: 0.6098605577689243
Percentage Arrested or Summoned to court: 0.12559760956175298
Percentage of Hispanic in Dataset:  0.2486933690025018
```

Figure 9

Low-income group statistics

Decision Tree Modeling Results

After running the script for our decision tree model, we were able to return the predictions shown in figure 10. Calculating the accuracy of the model yielded 94.37% in the output as well. For figure 11, we used the classification report shown in class to generate values for the metrics displayed in the output.

```
done making tree
       serNum arrestMade
0          17          N
1         691          N
2        3714          N
3         633          N
4          36          N
...       ...        ...
99995    5074          N
99996    5075          N
99997     524          N
99998     653          N
99999    1202          N

[100000 rows x 2 columns]
94.37
```

Figure 10

Decision tree predictions series output. Accuracy = 94.37%

	precision	recall	f1-score	support
N	0.96	0.99	0.97	94383
Y	0.61	0.28	0.39	5617
accuracy			0.95	100000
macro avg	0.78	0.64	0.68	100000
weighted avg	0.94	0.95	0.94	100000

Figure 11

Running sklearn classification_report module to get metrics on our model provided this output.

In figure 12, our ROC curve (receiver operating characteristic curve) shows an AUC score of .64. We created this output to look at our performance among all classification thresholds.

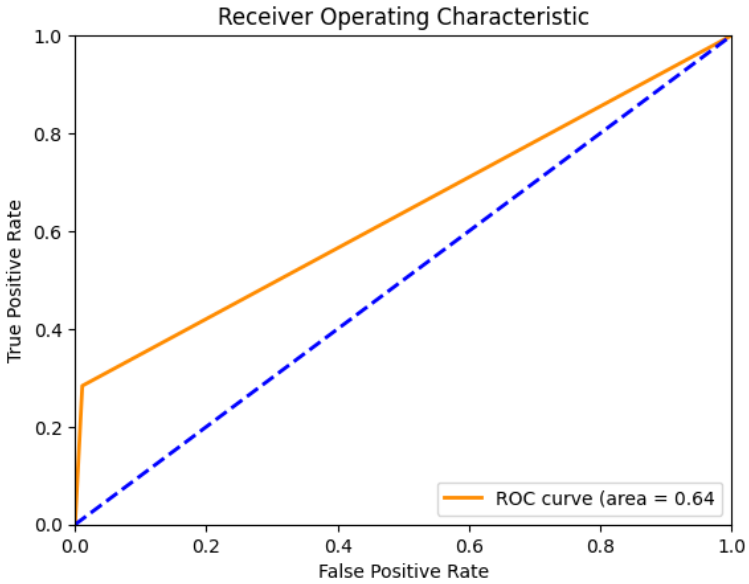

Figure 12

Plotting ROC curve with matplotlib parameters TPR and FPR

Fairness Results

We chose to use two fairness metrics in analyzing FNR and FPR. First in figure 13 we look at equal opportunity model in which our goal is to have the FNR close between all classes.

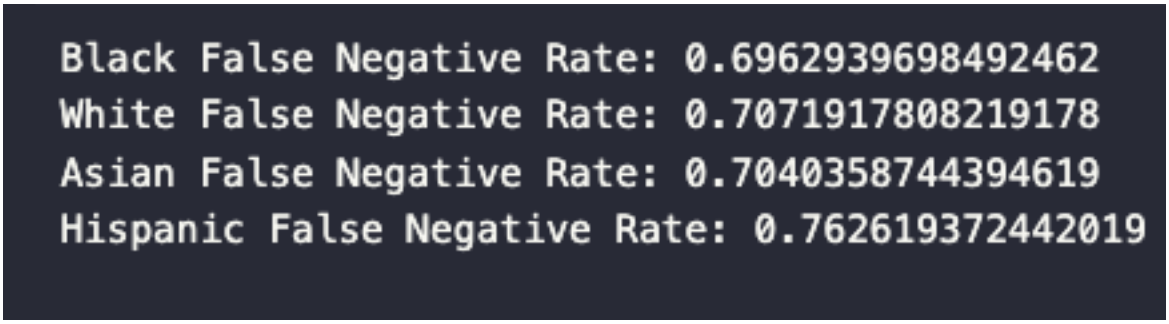

Figure 13

Calculating 'Equal Opportunity Fairness' between all the classes. FN/(FN + TP)

In figure 14, we see that this time we are looking for FPR in our relaxation of separation model. With the results all being 10% from eachother except Asian. The calculation at the bottom denotes the Asian FPR divided by black FPR to show the 60% difference between the two classes.

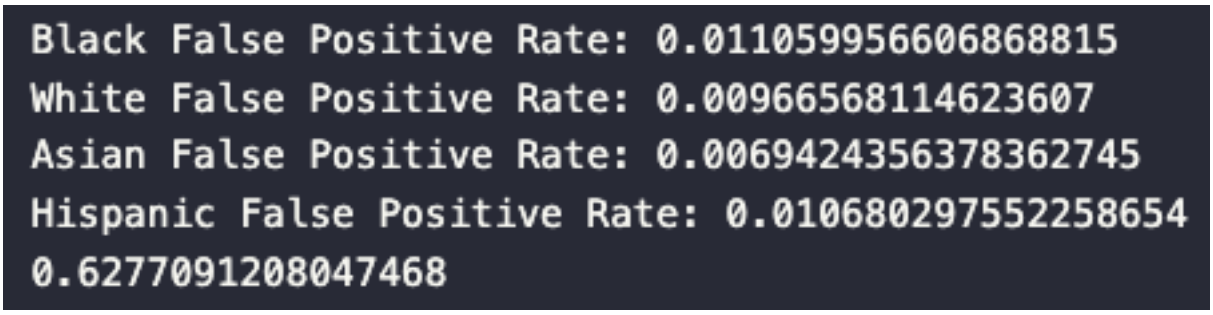

Figure 14

Calculating 'Equal Opportunity Fairness' between all the classes. FN/(FN + TP)

Discussion

Data Analysis

Our data analysis results showed a high quantity of records of those in the black class and Hispanic. Traditionally this definitely skews our observations when looking at all demographics. There were some we even had to leave out due to lack of data. This leads me to believe that overall, this isn't a great representation of biases in NYPD's policing procedures. While there definitely are more records of certain groups, we can't be certain based on our analysis among different cities, incomes and stop types.

Decision Tree

The decision tree portion of our project was accurate in predicting individuals belonging in the not arrested class. With .61 precision and .28 recall in the arrested class, our model had a harder time predicting true positives returning a high number of false negatives due to the lower recall. Looking at the ROC curve and AUC score of .64 this isn't great but rather poor in considering the performance of our model in all classifications. However, it is still better than guessing 50/50.

Fairness

We are looking at FNR in the Equal Opportunity Fairness model and how close values are between classes. False Negative in our model is if the person was not arrested but the model predicted them to be arrested. There is equal opportunity in the decision tree model as the FNR across the different races are within 10% of one another. Black, White, and Asian are extremely close at only 1-2% difference. Hispanic was the only one that stood out at an 8% difference (bFNR/hFNR).

We are looking at False Positive Rate (FPR) in the Relaxation of Separation model. False Positive in our model is if the person was arrested but the model predicted them to not be arrested. There is relaxation of separation in the decision tree model as the FPR across the all races but Asian is within 10% of one another. Black, White, and Hispanic are extremely close at only 1-2% difference. Asian was the only one that stood out at a 60% difference (aFPR/bFPR). There is inequality for the Asian race.