Client Requirements

Summary

For the past few years local police departments have faced challenges with receiving bulk of complaints regarding its stop and search policy. A decision about searching of a stopped car has been often based on a pure police officer intuition and in most of the cases is biased against people of certain backgrounds. The police department provided the consultancy *Awkward Problem Solutions*™ with data collected for the last 5 years (2013-2018) including the following information:

- If the stopped car was searched;
- A record whether contraband was found;
- Date and time of the intervention;
- Location of the intervention;
- Police department name and identification ID of an officer;
- A driver's town/state residency, age, gender, *race, ethnicity*.*

The main goals for researchers are to determine whether criticism is constructive, and subsequently to create a service which would *fairly* decide whether or not to search a car.

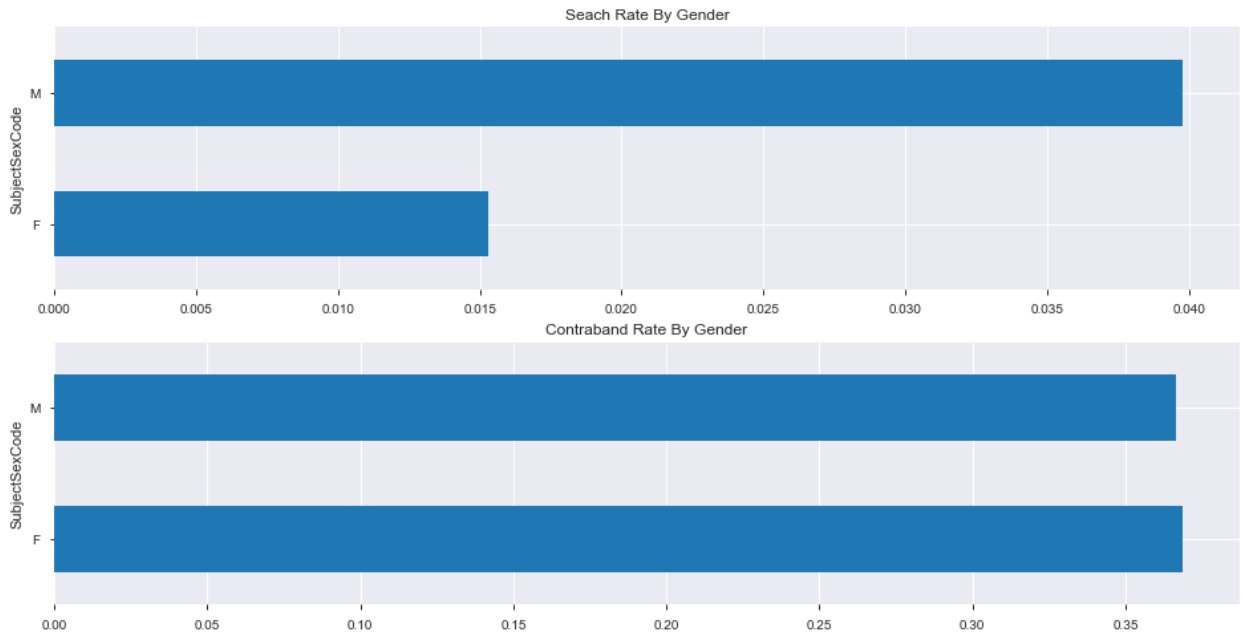
Requirements clarifications

For proper data analysis, is crucial to understand how data was collected. The following information will help with understanding if the collected data is fully representative.
The first thing we need to clarify is the way how our dataset was collected and if it is a sample (and randomly sampled) or the complete set. That will explain if our dataset neatly represents the population and the protected classes distribution.
According to the supplied information, the data about race and ethnicity have been added based on officers intuition and cannot be confirmed by relevant drivers documents. Therefore, these data cannot be considered as representative enough (e.g. black haired person can be marked as Hispanic or Middle Eastern), but our model is nonetheless bound by this information.
The first given requirement, i.e. 50% success rate for all searches, actually means the improvement of the found contraband by 13%, since the contraband rate in the provided dataset equals 37%.
The second requirement such as no less than 5% discrepancy between protected classes we can ensure if the provided dataset doesn't have this gap between noted classes, otherwise, it's very unlikely that our model will be able to overcome that.

Dataset Analysis

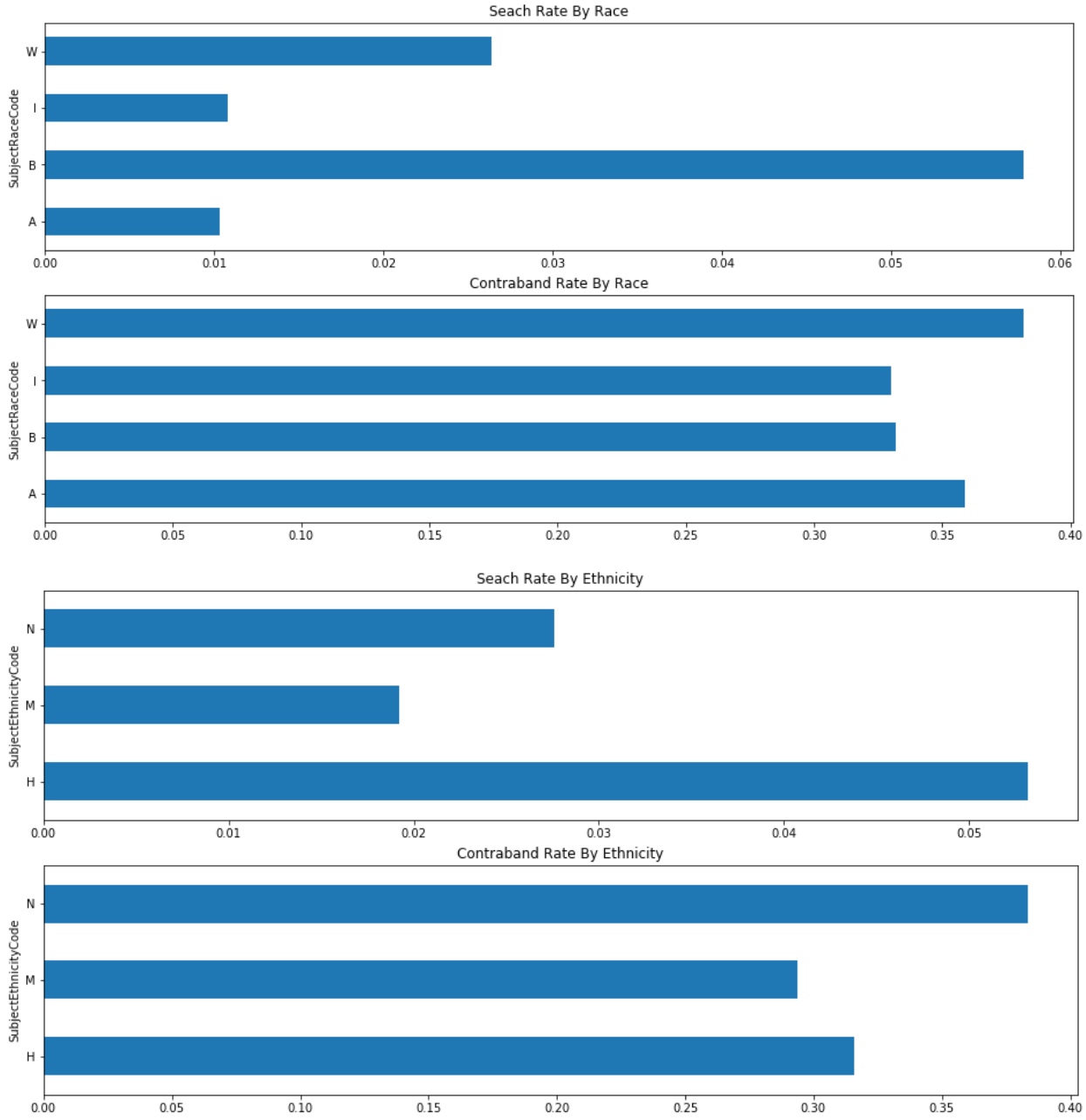
General Analysis

The given dataset consists of information about both searched and non searched cars. In 97% of all stops, the vehicle was not searched at all, while only 0.001% of non searched vehicles was found with contraband. When the car was searched, contraband was found in 33% of the searches.
As noted in the requirements, the decision to observe a car is often at the discretion of the police officer. Within that condition, let's see if there are any distinguishing patterns in driver demographics and stop outcome.
We can see that 63% of the stops are of men drivers, and almost 37% are of women. For that amount, 4% of men and 1.5% of women have been searched.

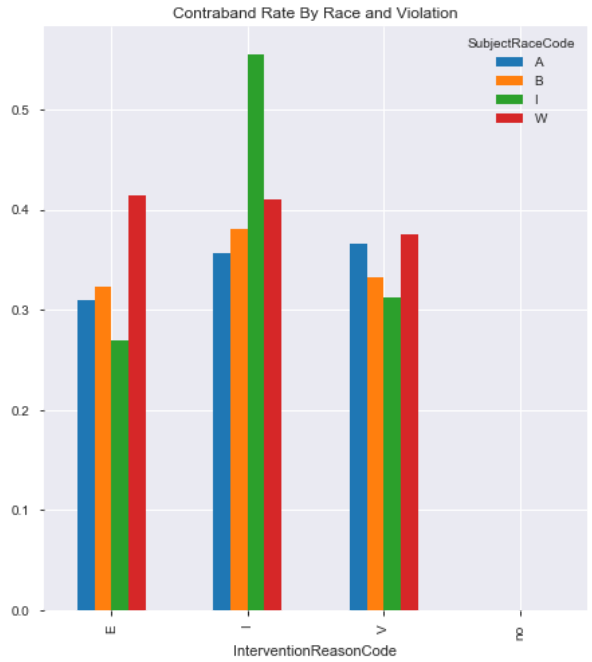


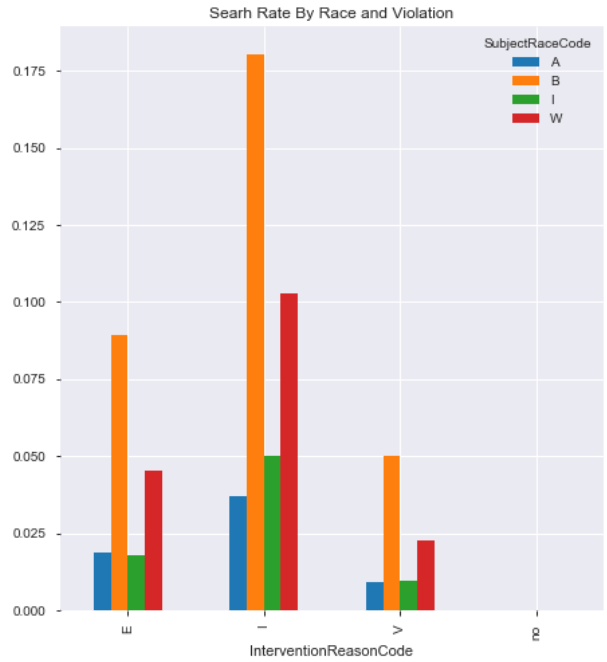
Nevertheless, the proportion of found contraband is about 37% for each class. According to that, thereafter we need to check if this result demonstrates a good officers intuition or actually the lack of data regarding searches of women.
The amount of drivers who've been stopped nearly corresponds with Connecticut's population race and ethnic percentages overall, i.e. we see that most of stopped vehicles were ridden by whites - 82%, Blacks - 15%, Asians - 2% and 1% of them are Indians.
Taking a closer look at the search rate, we see that black people's is 6%, then whites with the rate almost twice less - 2.6%, Asians and Indians are sharing almost the same search rate, which is 1% for each group. Along with that, checking the Ethnicity column, we can see that

Hispanic drivers search rate is 5%, for Middle Eastern 2% and 3% for others. And their contraband rate is 3%, where's other ethnicities contraband rate is almost 4.

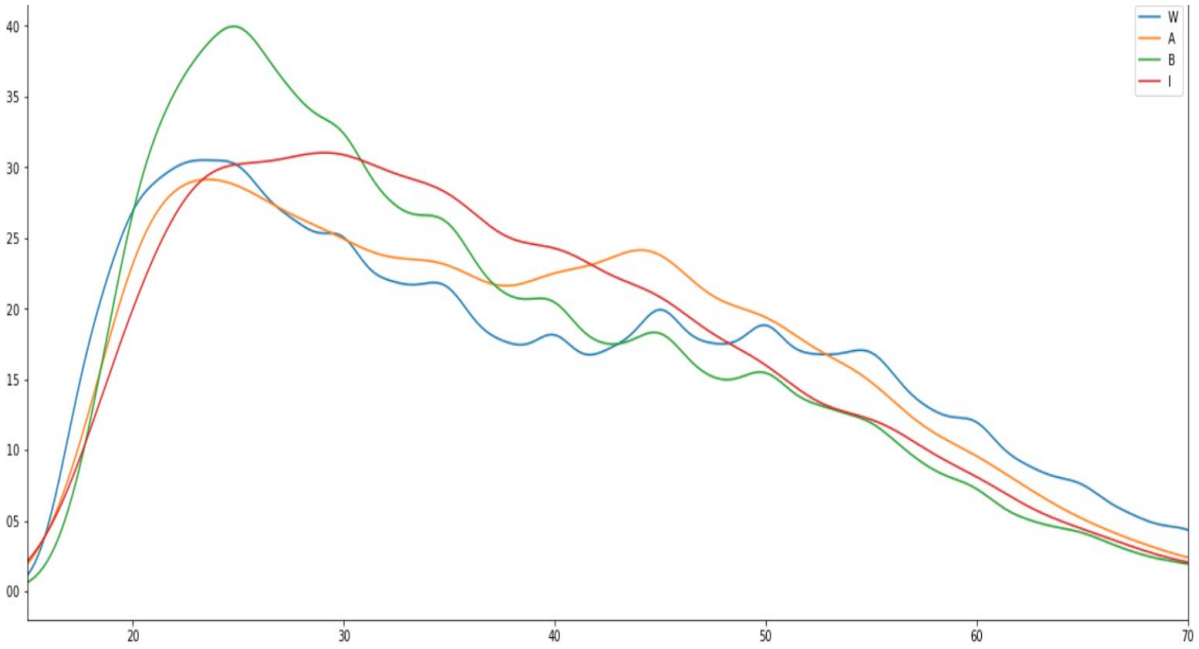


But if we consider the contraband rate, whites have it the highest: 38% of the searches, then Asians 36%, for Black is 33% and Indians contraband rate is 32% of the occured searches. Let's try to identify the cause of discrepancy taking a closer look at the search reason.





Here we can see that Black people have the highest chance to get investigated; the search rate for investigation reasons is 18%. While for white people search rate due to investigation reasons is almost twice less - 10% of searches; however, for the investigation reason Whites contraband rate is 3% higher than for Black race.



We can see that black people between 20 and 30 y.o. are the most likely to be pulled over, as well as other races. After 30 y.o. for Black drivers and after 35 y.o. for Indian drivers the contraband rate drops steadily. White and Asian drivers have a slight drop between 30 and 40,

and then after 40 y.o. we are witnessing a small increase for both races, which evenly drops for Asians after reaching 45 y.o. and unevenly for white people after 55 y.o.
And regarding gender, we see that both men and women are more likely to have a contraband between 20 and 30 y.o.

Business Questions Analysis

By now, we found some regularities in searches which might have negatively affected the contraband rate. Most notably, our research has shown that the police officers are much more often searching the vehicles driven by Black race than whites regarding their proportion in amount of stops. Along with that, men are searched more often than women. Under these circumstances, to meet the requirements, we need to strike a balance between the accuracy of our prediction, i.e. probability of contraband presence, and a fairness towards protected classes.

Conclusions and Recommendations

From the provided dataset, we concluded that Black people are more often investigated than other ethnicities. Whenever possible, the information below could help to investigate the police behavior and might help to get rid off some biases and, as a consequence, there's a chance that our prediction could be much more accurate and fair. However, it is significant to note that the information below will initially help with the context of the condition how data is inputted.

- **Car category and model.** What if trucks are more likely to have contraband because of higher capacity? Are old cars models more likely to smuggle contraband?
- **Driver's address.** The 82% of searched vehicles were driven by the state residents, but just 34% of drivers were belonging to a town where they've been stopped. To find out whether we're able to capture group behavior of certain neighborhoods, we need the particular address of stopped drivers.
- **Weathe**r. Analyzing the effect of weather on policing would help with understanding the police and drivers behaviour, including the stop duration.
- **Duration of the stop.** This information would help to understand how careful and detailed the search was, thereby, if it's correlated with the found contraband.
- **Police officer gender, age, ethnicity.** To see if officer physical traits differences have any impact on his decision whether to search a vehicle.

Modeling

Model expected outcomes overview

To start with, our model aimed to tell the officer whether to make a search of the stopped vehicle. The success rate of that search should be no less than 50%. We've been provided with data where 37% of actual searches resulted in contraband found. Appropriately, for us to provide with 50% success rate, we need to get more data to see if we are able to meet all the requirements.
Ideally, after receiving the suggestion from our model, the officer would input the results of that search to the app and the app will send the results to our agency. Thereafter, upon collecting sufficient volume of data, these results would be processed on the next stage of training our model.
The provided dataset has disproportionate ratio of observations in each class, and the major problem is that it's related to protected classes, e.g. the higher search rate for Black race or Hispanic Ethnicity and men gender. Therefore, to understand if black people are more likely to have contraband and if there's any reliance on gender, we will train without sensitive features.

Based on that, our app will suggest the officer whether to search the stopped car and after recollecting the data, we'll be able to see whether the training our model without sensitive features helped with improving the hit rate and at the same time, with keeping a discrepancy no bigger than 5% in the search success rate between protected classes.

Model specifications

Considering the given target (Contraband Indicator), the result should be a binomial response, since it turns out to be a classification problem. Our target variable is False for 99% of all stops, however, since we have the need to filter the data by including only the searched vehicles, on these terms the contraband indicator rate is 33%. We have just one numerical feature (driver age), the remaining ones are treated as categorical variables.
The original dataset has data quality problems, namely duplicated data (76% of duplicates are from Willimantic police department only) and inaccurate data: frequent typo errors in intervention location name, department name and subject age, also subject residential information (5% of town residents are shown not to belong to its respective state), a gap of data in 2015 and corresponding disappearance of the value 'Middle Eastern' during this time. The feature 'Reporting Officer ID' has the largest number of unique identifiers, and this variable almost means nothing when it comes to prediction. Another skipped feature is 'Intervention Date time' since we didn't find strong differences in the date time data, except of a noted gap in 2015. Also 'Intervention Location Name' has the highest amount of unique names. But this feature can be the most important for building a model without sensitive classes.

We built our model using a simple pipeline. Since most of the data types are categoricals, we've been using categorical encoder to encode them. After putting our model, e.g. Logistic Regression, in the pipeline, we fitted the pipeline to our training set.

Analysis of expected outcomes based on training set

There's a common idea that simply removing sensitive attributes from the training data might be able to solve the problem of biasness. Basically, it's a tradeoff against accuracy of our model and fairness towards protected classes.
Regarding that, in this chapter we'll consider the outcomes of our model for dataset with and without sensitive features.
For this project, the Logistic regression shown the highest result within a prediction score.
Building a model using all the features for searched vehicles except of date time column and officer ID, we've received:
F1-score: 0.67
Precision: 0.69
Recall: 0.66
Accuracy 0.73
In contrast, we considered the results for Logistic regression taking out the sensitive features.
We received the similar accuracy as with training our model on sensitive features, namely:
F1-score: 0.662
Precision: 0.68
Recall: 0.66
Accuracy: 0.723

As we can see, the drop of the sensitive features actually doesn't affect so much our accuracy, so we can build our model without them.

Alternatives considered

As an alternative, we took into consideration a Random Forest Classifier. The accuracy is a bit worse than for Logistic Regression. With using all the features it is 0.696, without - 0.68.
Regarding other metrics for dataset with sensitive features:
F1-score: 0.63
Precision: 0.64
Recall: 0.62
Without:
F1-score: 0.63
Precision: 0.63
Recall: 0.62
Regarding fairness, training a classifier without using sensitive features, gave us predictions with more or less the same score for protected classes.

Known issues and risks

Our model is trained on data which is not fully representative of protected classes. It has been built on a historical data and subsequently it's facing the sample bias. As a result, it's not able to produce the same score for all demographic groups. And yes, again, it sustains the common concern that data collected by humans can be highly-biased by design.
Besides of bias against protected classes, the record of contraband only comes from vehicles searched by police. The police department tends to dispatch more officers to the place that was found to have higher contraband rate initially and is thus more likely to record contraband in such regions. We cannot be sure if people in the locations with less searches having less possibility of contraband evidence as in locations searched more. The prediction system trained using data collected in the way based on the officer intuition tends to have his bias.
According to the received result, we'll put in production the model with the higher accuracy score, but if for the officer the fairness is more important than accuracy, we'll use Random Forest Classifier without sensitive features.

Model Deployment

Deployment specifications

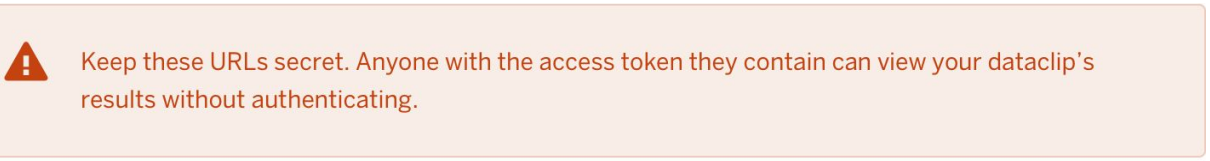
To deploy our model we created a Flask server, saved our trained model as a pickle file and loaded it to Heroku. Let's take a closer look at the entire process from scratch.
Firstly we need to preserve our model so that the value it generates can be used in a process or program apart from the one in which it was fitted. For this project we've been using a few Python packages: pickle from Python core and pipelines from scikit.
After the model got trained as part of the pipeline, we serialized it using pickle. To see how it essentially works, we made a random prediction in our notebook:
new_obs_str = '{"Department Name": "new Haven", "SearchAuthorizationCode": "C", "InterventionReasonCode": "V", "SubjectAge": 26, "InterventionLocationName": "New Haven","TownResidentIndicator":true }'
And we received a new observation as a json string afterwards. As is shows, Python works pretty well as a dictionary when it comes to json module. We received our prediction as False.
Next step after serialisation is implementation of a Flask server. In our python code we import the respective environment and made a single endpoint that served the predictions, thereby we created entire server that can be run by executing our application.
Creating endpoint is mandatory for deploying a model into the production environment, since the endpoint is defined as the interface to the model. This interface (endpoint) facilitates an ease of

communication between the model and the application. Specifically, this interface allows the application to send user data to the model and receives predictions back from the model based upon that user data.
For receiving new observations as a boolean response, we implemented get_json flask function.
After getting, producing and returning our predictions, we are using databases to keep a record of them in order to do some additional analysis of our model observations. For reducing the amount of code that needs to be written for working with databases, we'll use Object-Relational mapping (ORM) pewee. This will allow us to use a local database called sqlite (which is basically a file) when we are developing on our devices and use a more production-ready database called postgresql when deploying to heroku with very little change to our code.
After connecting to the database, defining the data model, creating the table and integrating it to web server, our app is finally ready to be deployed. We've been using heroku for this project since it is one of the oldest managed platforms out there and is quite robust, well-known, and documented.

Known Issues and risks

To sum up, a deployment to production is a method that integrates our model into an existing production environment so that the model can be used to make predictions based upon data input into the model. But the data can change over time, therefore the objective is to deploy models that would diagnose themselves and adapt to this changing data over time.
(The changes may be consequential, such as that the predictions made by a model trained on older historical data are no longer correct or as correct as they could be if the model was trained on more recent historical data.)

Another issue is related to privacy, our app's API is public which means that any citizen can use it to get predictions that are supposed to be disclosed.



Thereby, it's highly important to keep the access token just between our agency and a client and to keep an eye on how autonomously our model is working over time.

Annexes

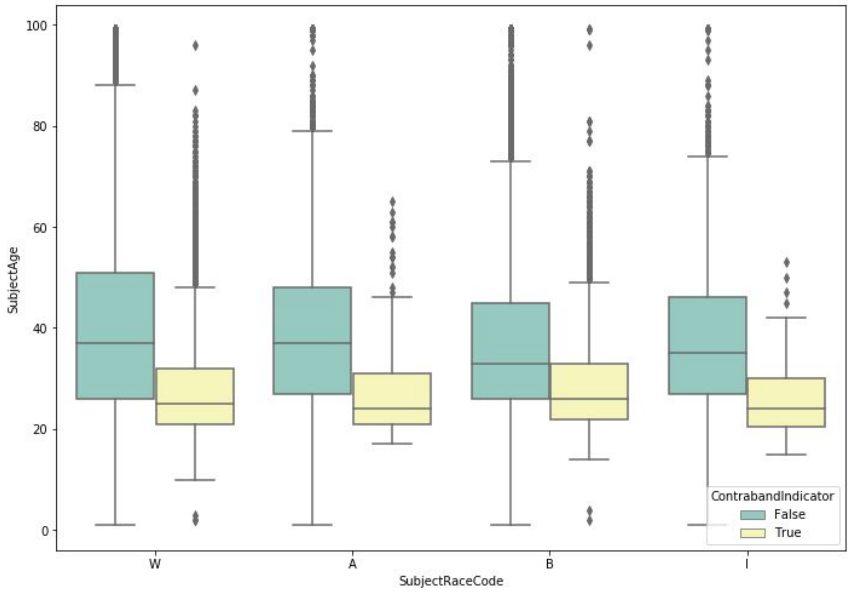
Dataset technical analysis

Let’s start off with a simple preview of our dataset.

```
RangeIndex: 2473643 entries, 0 to 2473642
Data columns (total 15 columns):
VehicleSearchedIndicator            bool
ContrabandIndicator                 bool
Department Name                     object
InterventionDateTime                object
InterventionLocationName            object
InterventionReasonCode              object
ReportingOfficerIdentificationID    object
ResidentIndicator                   bool
SearchAuthorizationCode             object
StatuteReason                       object
SubjectAge                          float64
SubjectEthnicityCode                object
SubjectRaceCode                     object
SubjectSexCode                      object
TownResidentIndicator               bool
dtypes: bool(4), float64(1), object(10)
memory usage: 217.0+ MB
```

count	2473643
mean	38
std	14
min	1
25%	26
50%	36
75%	50
max	99

We have just one numerical feature, which is Subject Age, let’s use a descriptive statistics to summarize the central tendency, dispersion and shape of Age distribution.
As we can see, minimum age is 1 and the maximum is 99 which obviously tells us about data quality problem. Let’s plot the age distribution for a raw data. Even though histogram is not the best for checking outliers and boxplot is already attached above in general analysis, we can see that outliers are insignificant.



And let's take a closer look at the proportion of search and contraband rate for age column overall excluding outliers.

SubjectAge	n_stops	n_searches	n_hits	search_rate	hit_rate
(15, 20]	199518.0	11289.0	5696.0	0.056581	0.504562
(20, 25]	385879.0	20918.0	8573.0	0.054209	0.409838
(25, 30]	345739.0	15355.0	5528.0	0.044412	0.360013
(30, 35]	283759.0	9564.0	2953.0	0.033705	0.308762
(35, 40]	235160.0	6430.0	1883.0	0.027343	0.292846
(40, 45]	223587.0	4252.0	1158.0	0.019017	0.272342
(45, 50]	220212.0	3406.0	959.0	0.015467	0.281562
(50, 55]	200695.0	2549.0	732.0	0.012701	0.287171
(55, 60]	155951.0	1604.0	453.0	0.010285	0.282419
(60, 65]	99869.0	744.0	224.0	0.007450	0.301075

Okay, the next question is: how actually complete is our dataset?

```
VehicleSearchedIndicator: 0.0%
ContrabandIndicator: 0.0%
Department Name: 0.0%
InterventionDateTime: 0.0%
InterventionLocationName: 0.0014553433943378248%
InterventionReasonCode: 8.085241079654582e-05%
ReportingOfficerIdentificationID: 8.085241079654582e-05%
ResidentIndicator: 0.0%
SearchAuthorizationCode: 0.0004042620539827291%
StatuteReason: 0.020496086136924368%
SubjectAge: 0.0%
SubjectEthnicityCode: 0.0%
SubjectRaceCode: 0.0%
SubjectSexCode: 0.0%
TownResidentIndicator: 0.0%
```

We see only five columns are incomplete: Intervention Reason, the Officer ID, Search Authorization code and Statute Reason. The amount of empty rows is not large, so we can drop the empty rows.

For the further analysis, we'll go into pandas profiling. As is well-known, the more values, the larger the memory consumption will be and since there are 2473643 values and 15 features, we've checked the EDA just for searched vehicles since it's less heavy for memory consumption and last but not least, our model must be trained just on searched vehicles, since the amount of found contraband for unsearched vehicles is statistically insignificant and the information about unsearched vehicles without contraband is not important for training our model. Attached is a panda profiling link with the basic statistical description of the searched vehicles.
We have Intervention date time feature as a string, after converting it to time series, we were able to observe if there are any specific patterns between period of day, day of the week, month and year in finding a contraband.

PeriodOfDay	n_hits	hit_rate
afternoon	6398.0	0.331279
evening	8675.0	0.352413
morning	3191.0	0.289933
night	7254.0	0.332630

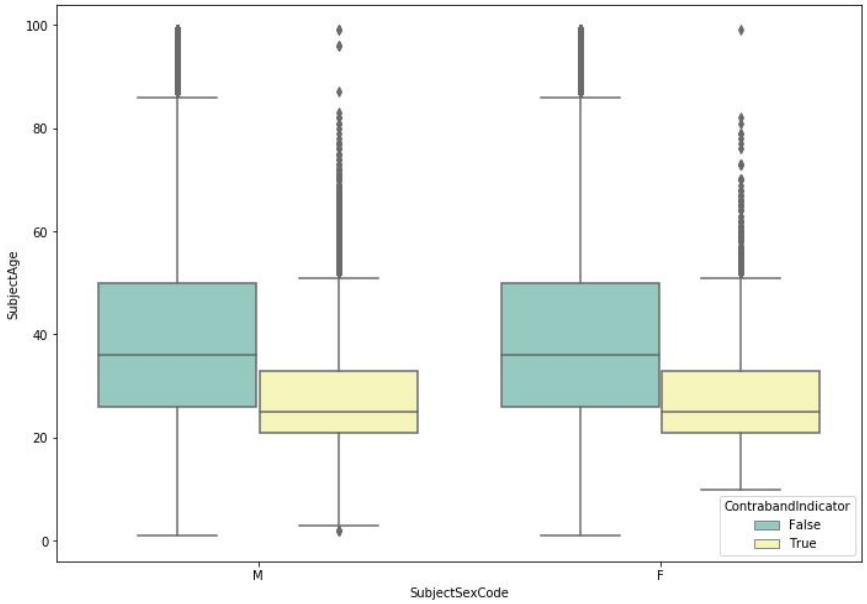
Weekday	n_hits	hit_rate
0	3166.0	0.324718
1	3482.0	0.332760
2	3512.0	0.324914
3	3861.0	0.336764
4	4298.0	0.341165
5	4185.0	0.341716
6	3014.0	0.320298

Year	n_hits	hit_rate
2013	1181.0	0.301045
2014	5461.0	0.305493
2015	2749.0	0.346527
2016	5582.0	0.339208
2017	5869.0	0.336005
2018	4676.0	0.357274

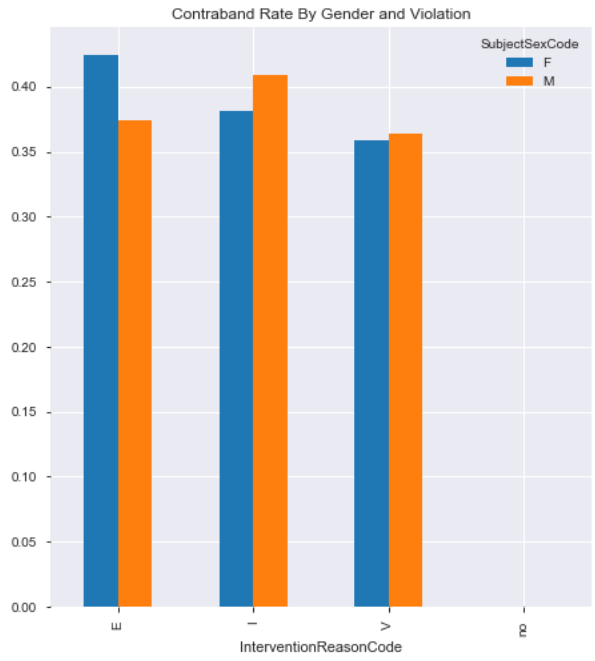
Comparing different features with time data, we found the strange loss of time for Middle Eastern ethnicity in 2015:

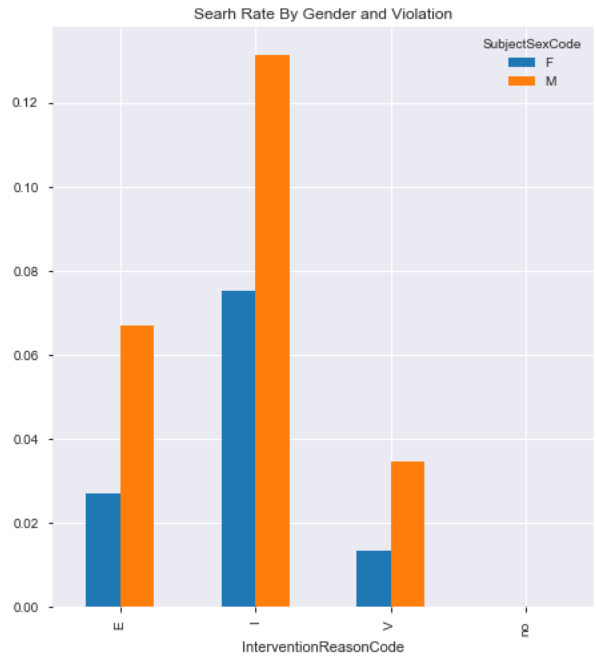
Year	SubjectEthnicityCode		
2013	H	209.0	0.250299
	M	55.0	0.276382
	N	917.0	0.317411
2014	H	882.0	0.238250
	M	170.0	0.282392
	N	4409.0	0.324860
2015	H	507.0	0.288560
	M	20.0	0.266667
	N	2222.0	0.364203
2016	H	1035.0	0.275266
	N	4547.0	0.358144
2017	H	1175.0	0.280430
	N	4694.0	0.353544
2018	H	1048.0	0.319512
	N	3628.0	0.369902

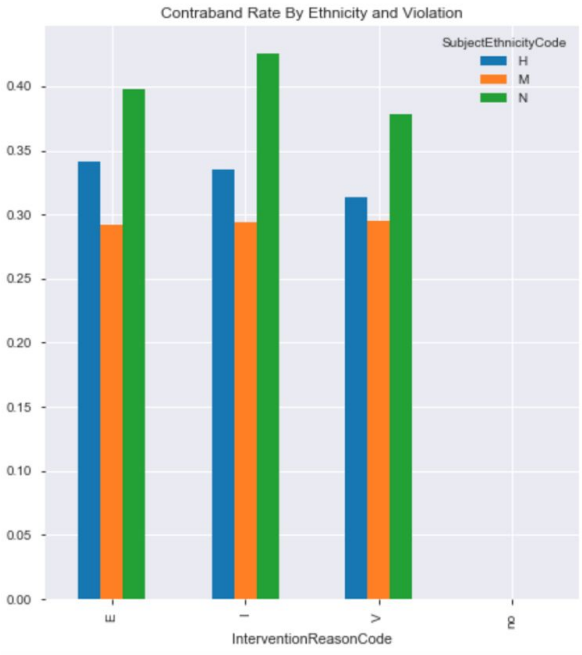
Again, we assume it happened because of data quality issues.
The further analysis will be related to the protected classes, e.g. Race, Genger, Ethnicity.

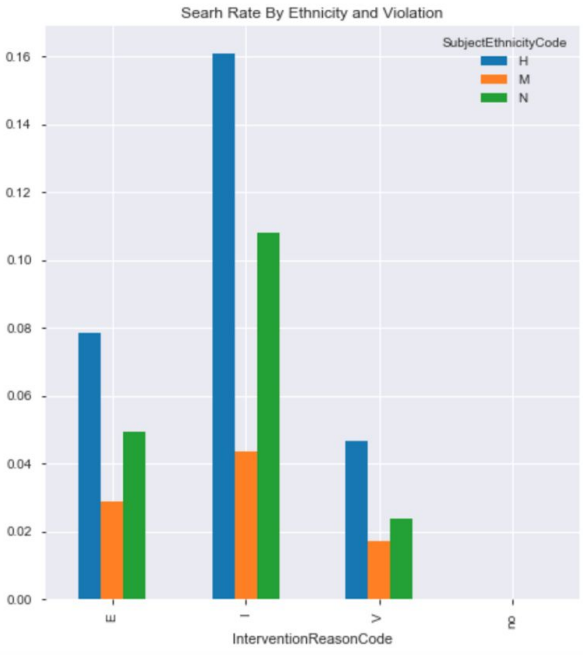


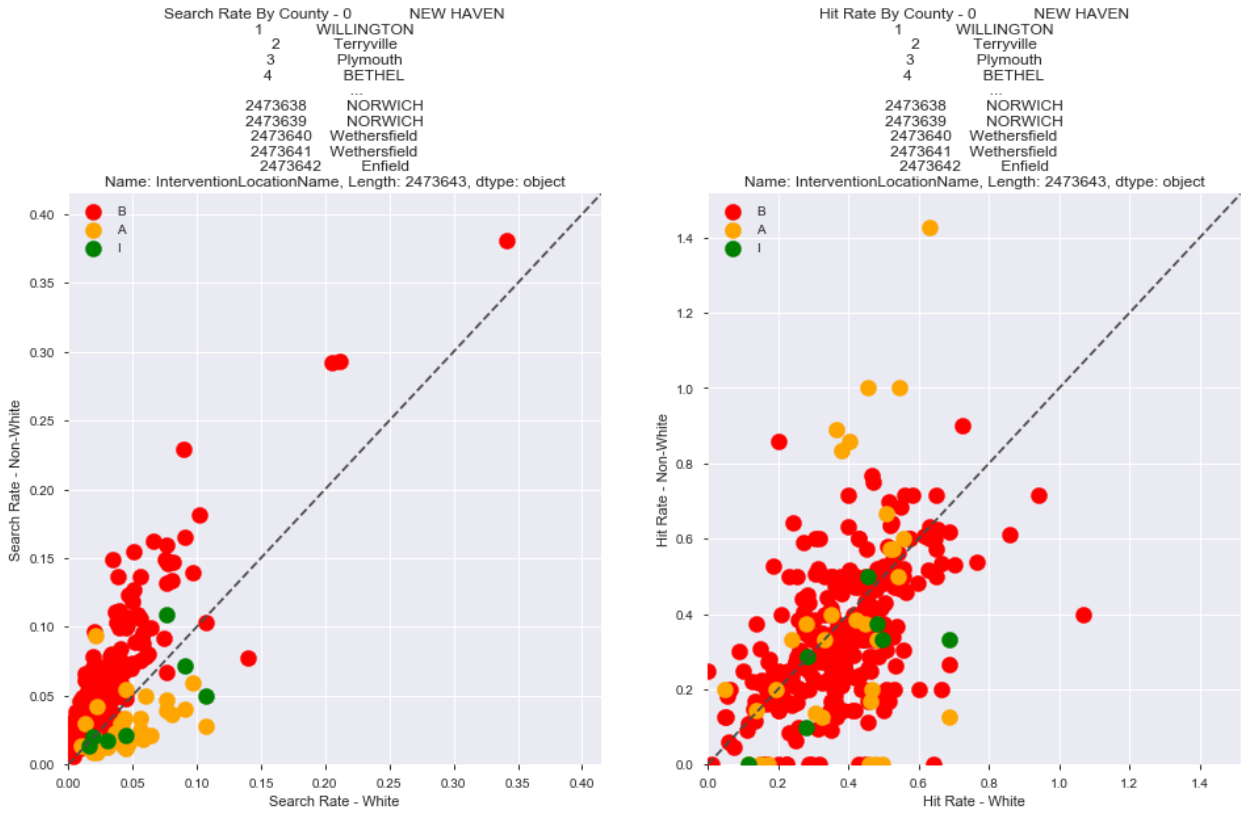
Contraband rate for genders depend on the age.


Contraband Rate By Gender and Violation
SubjectSexCode
F
M
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
E
I
V
no
InterventionReasonCode


Searh Rate By Gender and Violation
SubjectSexCode
F
M
0.12
0.10
0.08
0.06
0.04
0.02
0.00
E
I
V
no
InterventionReasonCode


Contraband Rate By Ethnicity and Violation
SubjectEthnicityCode
H
M
N
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
E
I
V
no
InterventionReasonCode


Searh Rate By Ethnicity and Violation
SubjectEthnicityCode
H
M
N
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
E
I
V
no
InterventionReasonCode



This scatter plot is demonstrating a few different things, first is a distribution of search_rate (left) and contraband_rate (right) for minority drivers compared with white drivers in each location name. If all of the dots (each of which represents the stats for a single county and race) followed the diagonal center line, the implication would be that white drivers and non-white drivers are searched at the exact same rate with the exact same standard of evidence. And as well, it's basically showing us data quality problems with Intervention Location name.
Such columns as Department name and Intervention Location name required a short preprocessing e.g. removing the whitespaces and normalizing the font size for location names.

Model technical analysis

As was noted before, the model selection for our project is based on a tradeoff between the accuracy and fairness.
Below are the results of Logistic Regression for a test set.
The prediction of our dataset with sensitive features using Logistic Regression:

Original test set (race):

SubjectRaceCode	n_hits	hit_rate
A	49.0	0.333333
B	1859.0	0.308445
I	18.0	0.305085
W	5085.0	0.339181

Predictions of our model based on the test set (race):

SubjectRaceCode	n_hits	hit_rate
A	23.0	0.156463
B	1321.0	0.219180
I	14.0	0.237288
W	3888.0	0.259338

Original test set (gender):

SubjectSexCode	n_hits	hit_rate
F	1328.0	0.326852
M	5780.0	0.336948

Predictions of our model based on test set(gender):

SubjectSexCode	n_hits	hit_rate
F	977.0	0.240463
M	4263.0	0.248513

Original test set (Ethnicity):

SubjectEthnicityCode	n_hits	hit_rate
H	1360.0	0.277608
M	82.0	0.281787
N	5666.0	0.353528

Predictions of our model (Ethnicity):

SubjectEthnicityCode	n_hits	hit_rate
H	874.0	0.178404
M	44.0	0.151203
N	4322.0	0.269670

The prediction of our dataset without sensitive features using Logistic Regression:

Original test set (race):

SubjectRaceCode	n_hits	hit_rate
A	23.0	0.287500
B	1114.0	0.308246
I	16.0	0.380952
W	3163.0	0.346706

Predictions of our model based on the test set (race):

SubjectRaceCode	n_hits	hit_rate
A	18.0	0.225000
B	901.0	0.249308
I	15.0	0.357143
W	2362.0	0.258906

Original test set (gender):

SubjectSexCode	n_hits	hit_rate
F	785.0	0.320277
M	3531.0	0.339258

Predictions of our model based on test set (gender):

SubjectSexCode	n_hits	hit_rate
F	676.0	0.275806
M	2620.0	0.251729

Original test set (Ethnicity):

SubjectEthnicityCode	n_hits	hit_rate
H	818.0	0.276632
M	58.0	0.306878
N	3440.0	0.354165

Predictions of our model based on test set (Ethnicity):

SubjectEthnicityCode	n_hits	hit_rate
H	648.0	0.219141
M	38.0	0.201058
N	2610.0	0.268712

Classification Report for Logistic regression excluding sensitive features:

	precision	recall	f1-score	support
False	0.76	0.85	0.80	14109
True	0.61	0.46	0.53	7108
accuracy			0.72	21217
macro avg	0.68	0.66	0.66	21217
weighted avg	0.71	0.72	0.71	21217

Classification Report for Logistic regression including sensitive features:

	precision	recall	f1-score	support
False	0.76	0.86	0.80	14109
True	0.61	0.45	0.52	7108
accuracy			0.72	21217
macro avg	0.68	0.65	0.66	21217
weighted avg	0.71	0.72	0.71	21217