Client requirements

Summary

Our company, *Awkward Problem Solutions™*, was awarded a contract by the police department of Connecticut. The police department has received lots of complaints about its stop and search policy. Every time a car is stopped, the police officers have to decide whether or not to search the car for contraband. According to critics, these searches have a bias against people of certain backgrounds.
This project aims at (1) determine whether these criticisms seem to be substantiated, and (2) create a service to fairly decide whether or not to search a car, based on objective data. This service will then be used by police officers to request authorization to search, and the service will return a Yes or No answer.
The project has the following requirements: 1) A minimum 50% success rate for searches (when a car is searched, it should be at least 50% likely that contraband is found); 2) No

police sub-department should have a discrepancy bigger than 5% between the search success rate between protected classes (race, ethnicity, gender); 3) The largest possible amount of contraband found, given the constraints above.

Requirements clarifications

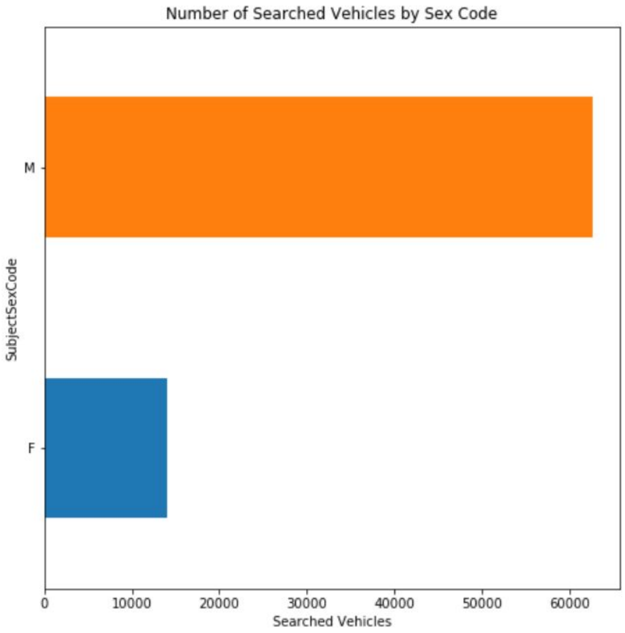
In technical terms, the success rate for car searches is equivalent to the precision of a machine learning model.
The precision is calculated as the number of true positives (cars that are searched and have contraband) divided by the number of true positives plus false positives (cars that are searched but do not have contraband).
As an example, if the false positives are 0 are the true positives are more than 0, the precision is 1 (meaning that 100% of the cars that are stopped have contraband). If the true positives are equal to the false positives (and both are more than 0), then the precision is 0.5 (meaning that 50% of the cars that are stopped have contraband).
Thus, the first requirement is that the model guarantees a minimum of 0.5 precision (at least half of the cars that are searched must have contraband).
The second requirement is that, between protected classes (race, ethnicity, gender) the discrepancy in precision should be less or equal than 5%, in each police sub-department. This is a requirement that aims to ensure fairness between the protected classes.
The third requirement aims at minimizing the number of false negatives (cars that were not stopped but did have contraband). In technical terms, this is the same as maximizing recall, which is calculated as the number of true positives divided by the number of true positives plus false negatives.
This requirement aims at catching the most contraband possible, constrained to the previous two requirements.
It is important to notice that this constrained optimization process means that we are willing to miss some cars with contraband in order not to annoy some protected classes with too many car searches.

Dataset analysis

General analysis

The dataset is a list of observations of stopped cars by the police of Connecticut. Please look at the data dictionary in the “Business questions technical support” Annex.

Beginning by looking at the sex of the drivers, there are about 1.5M male observations and 0.9M female observations in the dataset (roughly 1.5 times more males than females). However, looking at the distribution of the searched vehicles by sex code:



One can see that there are roughly 4 times (60000/15000) more cars stopped with a male driver than with a female driver. Thus, being a male increases the probability of the car being stopped (compared to being a female).

Regarding the race code, the volumes of the classes are very different (there are many more observations of the white race than the indian race) and thus it is difficult to visualize the results in a chart.

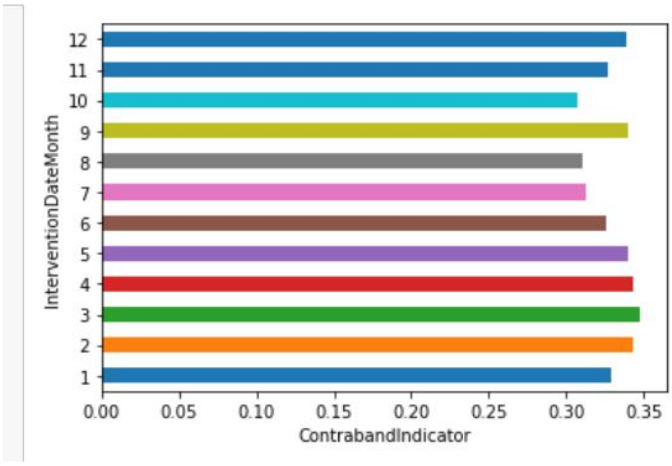
However, dividing the number of observations in the dataset (by each race) by the corresponding number of searched vehicles (again by each race), we get approximately the following results: 3% of all white race drivers, 6% of all black race drivers and 1% of both asian and indian race drivers are stopped. Thus, having a black race increases the probability of the car being stopped (compared to the other two races).

A similar procedure for the ethnicity code shows that 2% of all middle eastern drivers, 3% of all drivers without ethnicity and 5% of all hispanic drivers are stopped. It is then true that being hispanic increases the probability of the car being stopped (compared to the other two ethnicities).

Regarding the ReportingOfficerIdentificationID column, there are 8593 unique officers. The percentage of vehicles searched by an officer varies from 0 (an officer that never searches the cars that he stops) to 100% (an officer that always searches the car that he stops), as does the contraband finding percentage (there are officers that never find contraband in the cars that they stop and officers that always find contraband in the cars that they stop).

Regarding the Department Name column, there are 122 unique departments in the dataset. The percentage of vehicles searched by officers of a department varies from 0 (a police department in which the police officers never search the cars that they stop) to around 20% (a police department in which the police officers search around 20% of the cars that they stop). The contraband finding percentage varies from 0 (a police department in which the police officers never find contraband in the cars that they stop) to around 11% (a police department in which the police officers find contraband in around 11% of the cars that they stop).

Looking at the average contraband by month (where 1 is January and 12 is December):



One can see that there seems to be a cyclical trend of contraband, with a peak in month 9 (September). This is an important finding that will have an impact in the way that we will split the dataset in the following sections.

Business questions analysis

Some critics claim that Connecticut police searches have a bias against people of certain backgrounds (race, ethnicity, gender).
As we have seen in the last sub-section, it is indeed true that some races, ethnicities and genders are more likely to be searched than others.
However, that by itself does not prove that there is a bias: some races, ethnicities and genders may actually be a good predictor of contraband, in which case they're more likely to be searched because the police is doing a good job detecting contraband and not because they're biased.
A fair way to assess bias is then to compare the precision (defined earlier and equivalent to the search success rate) between the classes. Big differences in the precision between the classes indicate that one class is being searched more than the other and that difference is not being explained by the discovery of contraband.
We will then consider that there is a bias if there is a discrepancy greater than 5% between the search success rate between protected classes (race, ethnicity, gender). We choose this value (5%) since it was also used in the second client requirement. However, we will not calculate this difference for each sup-department. The explanation for this decision is presented in the section “Model expected outcomes overview”**.**
After calculating the discrepancies between each protected class (please check the Business questions technical support Annex for the in-depth explanation), we conclude that, globally:

1. there is no bias related to gender;
2. there is bias related to ethnicity: drivers without an identified ethnicity are being unfairly less searched than the other two ethnicities (hispanic and middle east);
3. there is a bias related to race: drivers with white race are being unfairly less searched than drivers with asian race.

It should also be noted that while the discrepancy between the search success rate of white and black drivers was less than 5% (4,97%) it is significant enough to say that with drivers with white race are also being unfairly less searched than drivers with black race.
It is also interesting to note that not all the discrepancies found correspond to classes that were probabilistically more likely to be searched, e.g., while 2% of white race drivers are searched compared to just 1% of asian race drivers, white race drivers are still being unfairly less searched than asian race drivers.
The claim that Connecticut police searches have a bias against people of certain backgrounds (race, ethnicity, gender) is then substantiated.
All these factors contribute to a global search success rate (precision) of only approximately 37%.

Conclusions and Recommendations

As can be seen in the Business questions technical support annex, police sub-departments CSP Troop I, Darien, ECSU, Enfield, Guilford, MTA Stamford, Meriden, Middletown, Orange, Redding, Rocky Hill and Winsted have a discrepancy greater than 90% between the search success rate of at least one of the protected classes (race, ethnicity or gender). We recommend that these departments are object of further investigation.
Similarly, officers 1000001949, 1000002608, 179, 199, 256, 30233 and 570 have a discrepancy greater than 90% between the search success rate of at least one of the protected classes (race, ethnicity, gender). In this analysis, only officers with more than 100 cars searches were considered. We took this decision because officers with very few cars searches can have big discrepancies in the search success rate of the protected classes without that necessarily meaning that they are being biased. A detailed example of why it may be misleading to analyze percentages with a low number of observations is shown in the next section.
We then recommend that these officers are object of further investigation.
In conclusion, the low global precision, the discrepancies in global precision of the protected classes and the police sub-departments and officers with very high precision discrepancies draw a picture in which the decision to search a car is based on intuition and preconceived opinions and not on objective data. This is the cause of bias.
We recommend that from now on, the decisions on whether or not to search a car should be based only in our model (with the obvious exception of cases where the contraband is visible).

Modeling

Model expected outcomes overview

Due to the high bias present in the data, we will not be able to fulfill all the requirements stated in the sub-section “Requirements clarifications”.
Namely, we will not be able to achieve a discrepancy in precision of less or equal than 5% in each police sub-department between protected classes (race, ethnicity, gender). This was not a reasonable objective to achieve.

To understand why, we can look, for example at the WCSU Police Department. This department has stopped 67 cars of drivers without an identified ethnicity, 28 cars of hispanic drivers and 1 car of a middle eastern driver. If the police decides to search the car of the middle eastern driver, only two outcomes are possible: they find contraband or they do not.

In the first outcome, precision for the middle eastern class will be 100% and imposes the restriction that precision for the other two ethnicity classes should be no less than 95%. That is an astonishing difficult precision to achieve in real world systems.

In the second outcome, precision for the middle eastern class will be 0% and imposes the restriction that precision for the other two ethnicity classes should be no more than 5%. This would imply that our model, for these two classes, would have to be much worse than the current one that has global search success rate (precision) of approximately 37%.

Instead, we will aim for a softer requirement of a global discrepancy in precision of less or equal than 10% between protected classes (race, ethnicity, gender). We believe that this is an attainable objective that will still guarantee that the police searches are fair and does not suffer from the problem described before.

We believe that the other two objectives: a minimum of 0.5 global precision and maximizing the recall taking into consideration the other two objectives will be obtained, however, since the minimum of precision is 0.5 we do not expect a high recall value (a looser restriction on Precision would allow for a higher Recall. This effect is known in technical literature as the precision-recall tradeoff).

Overall, we believe that the outcomes from the production run will achieve these reformulated project objectives.

Model specifications

We only trained the model with observations of vehicles that were searched, since the officers will only request the model’s assistance when they think that the car should be searched.

Due to the cyclical trend of the data observed in section “General analysis” we decided to use a time series cross validation to test our model.

The categorical columns: 'Department Name', 'InterventionLocationName', ‘InterventionReasonCode', 'SearchAuthorizationCode', 'StatuteReason', 'SubjectRaceCode' and 'ReportingOfficerIdentificationID' were encoded using a BinaryEncoder (which showed improved results compared to a OrdinalEncoder). The categorical features SubjectSexCode and SubjectEthnicityCode and the numerical feature ResidentIndicator were not used (excluded in a feature importances analysis). The feature SubjectRaceCode was later excluded because the discrepancy between the race class was lower when this feature was not included in the model (and it also improved roc_auc a little bit).

Regarding data cleaning, we replaced the values of SubjectAge that were under 12 by the median of the feature (for reference, the legal driving licence in Connecticut is 16).

For the features InterventionLocationName, InterventionReasonCode and ReportingOfficerIdentificationID the null values were filled with either ‘unknown’ or ‘U’. The null values of StatuteReason were filled with ‘Other/Error’ and SearchAuthorizationCode null values were filled with ‘N’, which stands for ‘Not Applicable’.

In the Department Name feature, Mohegan Tribal and Mashantucket Pequot departments were replaced by Mohegan Tribal Police and Mashantucket Pequot Police, respectively.

Regarding the InterventionDateTime feature, we extracted the year, month, day and hour and cyclically encoded them (except the year).
We used a RandomForestClassifier and hyperparameter tuned the following parameters: n_estimators, max_features, max_depth, min_samples_split, min_samples_leaf, bootstrap and class_weight. We decided to manually optimize these parameters instead of using a grid or random search because the model took a very long time to run. Our final model had the following features: n_estimators=150, max_features = ‘auto’ (default value), max_depth=9, min_samples_split=5, min_samples_leaf=1 (default value), bootstrap = True (default value) and class_weight = None (default value).
After analysing the precision recall curves, we choose to use a threshold of around 0.39. A more detailed explanation is done in the Model technical analysis annex.

Analysis of expected outcomes based on training set

The precision on the test set using a threshold of 0.39 was 0.59. This gives us a buffer of around 0.1 to “absorb” the expected drop in precision that the model will have in production. We expect this buffer to be enough and thus the precision of the production model to be compliant with the first client requirement.
Regarding the discrepancies of precision between classes, in the section “Model expected outcomes overview” we compromised ourselves with a global discrepancy in precision of less or equal than 10% between protected classes (race, ethnicity, gender). The 0.39 threshold has the following discrepancies between classes: 1% for the sex/gender, 3% for the ethnicity and 7% for the race class. Thus, in the worst case, we still have 3% to absorb an increase in the discrepancy of the race class. We expect this buffer to be enough and thus the discrepancy in precision between the protected classes of the production model to be compliant with the second modified client requirement.
Regarding the third requirement, the recall on the test set is around 0.71. Given the previous constraints, we believe this to be a very good value, meaning that more than 2 out of 3 cars with contraband get caught. Nevertheless, regardless of the performance in production, the model was designed to maximize recall under the constraints of global precision and precision discrepancies between protected classes, as per the client third requirement.

Alternatives considered

Regarding categorical encoding, since some features had a very high cardinality, we needed an encoding that ensured that the number of newly created features was kept within a reasonable value. For that reason we dismissed one-hot encoding. Since many of the categories of our categorical features had a low amount of observations, we decided not to use target encoding. We began with ordinal encoding (which was fine since we always used a random forest classifier), but ended up using a binary encoder due to a little improvement in performance.
Regarding time features, we ended up not using the minutes and seconds because they had a very high number of observations with value ‘0’. Knowing that it is likely that these observations are 0 just because the officer did not remember the specific time, we decided not to use this features.

Regarding the splitting of the dataset, as referred earlier, even though the splitting of the dataset into a 70% train set and 30% test set would be technically correct (if done deterministically and after sorting the whole dataset by date), we found out that it left the test set without observations of middle eastern drivers and thus we had no way to analyse the second client requirement. For this reason we ended up using a time series cross validation. Our precision, recall and threshold values for the protected classes were obtained for the last test folder of the cross validation, while the precision, recall and threshold for the middle east class was obtained for the first test folder of the cross validation (the only one with middle eastern drivers observations) .
The choice of a tree based model meant that we didn't have to worry about data scaling issues.

Known issues and risks

Regarding the performance in production several things can go wrong:
i) the production observations can have data that is underrepresented in the training set (e.g. some features like ReportingOfficerIdentificationID have many categories with a low amount of observations), which will result in a less than expected performance from the model;
ii) the production observations can have observations with classes that were not present at all in the training set (e.g. Unknown class in SubjectRaceCode feature) and while our encoder while have no technical problem with that, the model will not be able to use that class to predict;
iii)while cross validation was used to obtain the model with the best roc_auc, the discrepancy of precision between the classes was calculated using the last folder of the cross validation (except for the middle eastern class in which the precision, recall and threshold vectors were calculated using the first folder of the cross validation). So there is a possibility that this last folder was a particularly easy one to predict and has such we are being overly optimistic about the precision discrepancies between protected classes.

Model Deployment

Deployment specifications

Our model is deployed behind a HTTP server on heroku.
The HTTP server was setup with the Flask framework. It has just two HTTP endpoints: '/predict' that serves the predictions and '/update' which updates the target value of previously received observations, in the database .
When the server receives a predict HTTP POST with json data (the new observation) it uses the get_json() function from flask to parse the request, creates a python dataframe (using the previously deserialized columns and dtypes of the model) and executes the method predict_proba on the new observation, using the previously deserialized scikit pipeline (which contains the machine learning model). This probability is then compared to the defined threshold: if it his higher or equal then the threshold, the predicted_class is True,

otherwise it is False. This predicted_class is the value returned by the server in the predict endpoint.

The HTTP server is also integrated with a PostgreSQL database. The database has just five fields: observation_id, an unique integer field that acts as the index of the database, observation, a text field that stores the observation received with the predict HTTP POST request, proba, a float field that stores the result of the method predict_proba called upon the observation, predicted_class, an integer field that stores the predicted value of the ContrabandIndicator of the observation and true_class, an integer field that stores the target value of the observation, which is only received in update HTTP POST requests.

When the server receives an update HTTP post with json data, it uses the get_json() function from flask to parse the request and tries to save in the database the true_class of the observation (identified by its observation_id).

Currently the server is only catching integrity errors in the case of observations that have an observation_id that already exists in the database.

Known issues and risks

The model is deployed in a free tier of Heroku. In this free tier, the apps sleep automatically after 30 minutes of inactivity. This poses a risk because even though the app should wake automatically when a new web request is received, the longer time to serve that specific observation may cause a timeout error in the client side.

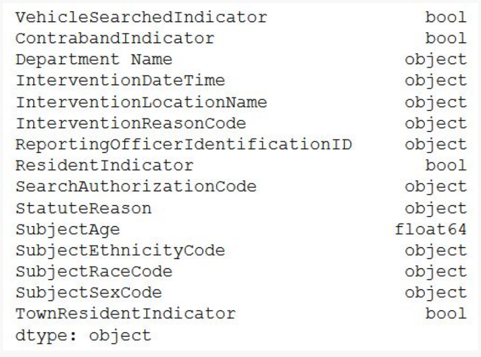
Another source of risk is the free PostgreSQL database. In the free tier, it has two limitations that could impact model performance: a limit of 10 thousand rows, which means that the model is only going to be able to serve 10 thousand predictions; a maximum of 4 hours of downtime per month, which means that during the production run week the model can be down for 4 hours (and will not serve predictions in that time).

Apart from this known issues, there are a myriad of issues that could come, for example, from incorrect data formats in the observation, e.g. if the ‘SubjectAge’ field comes with a string instead of a float value. Currently, in this case the observation dataframe creation would fail and no prediction would be given to the police officer.

Annexes

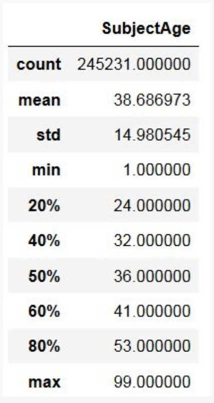
Dataset technical analysis

We began by looking at the times of data in our dataset:



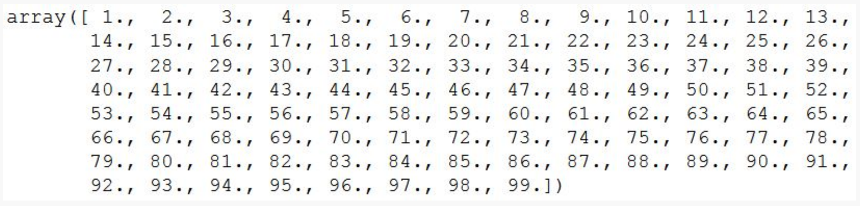
```
VehicleSearchedIndicator                bool
ContrabandIndicator                     bool
Department Name                       object
InterventionDateTime                  object
InterventionLocationName              object
InterventionReasonCode                object
ReportingOfficerIdentificationID      object
ResidentIndicator                       bool
SearchAuthorizationCode               object
StatuteReason                         object
SubjectAge                           float64
SubjectEthnicityCode                  object
SubjectRaceCode                       object
SubjectSexCode                        object
TownResidentIndicator                   bool
dtype: object
```

So we have 3 dtypes: boolean, float and object. We then begin to look at the statistics of each feature, beginning by the float feature SubjectAge:



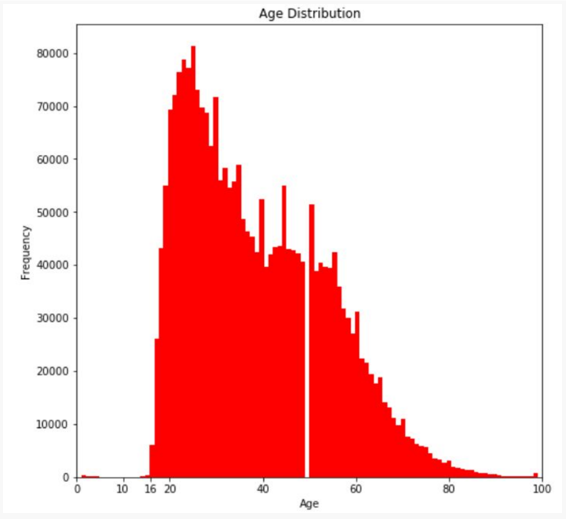
	SubjectAge
count	245231.000000
mean	38.686973
std	14.980545
min	1.000000
20%	24.000000
40%	32.000000
50%	36.000000
60%	41.000000
80%	53.000000
max	99.000000

So while there are some very old people driving (99 years), the interesting statistic here is that there is at least one observation with a wrong age (1). We investigate it further by listing the unique values of the feature:



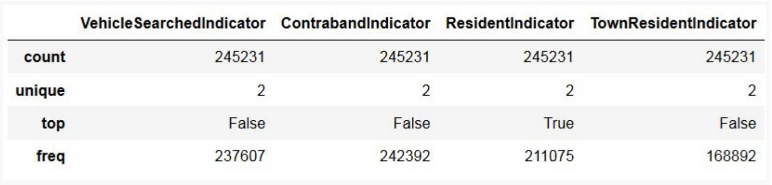
```
array([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10., 11., 12., 13.,
       14., 15., 16., 17., 18., 19., 20., 21., 22., 23., 24., 25., 26.,
       27., 28., 29., 30., 31., 32., 33., 34., 35., 36., 37., 38., 39.,
       40., 41., 42., 43., 44., 45., 46., 47., 48., 49., 50., 51., 52.,
       53., 54., 55., 56., 57., 58., 59., 60., 61., 62., 63., 64., 65.,
       66., 67., 68., 69., 70., 71., 72., 73., 74., 75., 76., 77., 78.,
       79., 80., 81., 82., 83., 84., 85., 86., 87., 88., 89., 90., 91.,
       92., 93., 94., 95., 96., 97., 98., 99.])
```

So as it turns out, there is a range of values that are clearly data imputation mistakes. Let us go even further and visualize the distribution of ages:



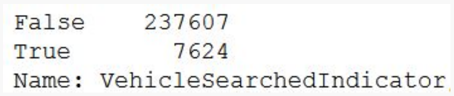
We can see that the ages that are lower than the driving age in Connecticut (16 years) are concentrated in the extremes of the interval [0, 16]. We can also see that around 50 years there is a gap of observations. We can also see a small bump in the 99 years which can mean that this is the maximum age that can be put in the system.

We follow up by analysing the boolean features:



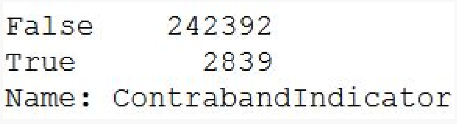
	VehicleSearchedIndicator	ContrabandIndicator	ResidentIndicator	TownResidentIndicator
count	245231	245231	245231	245231
unique	2	2	2	2
top	False	False	True	False
freq	237607	242392	211075	168892

Nothing really pops up. Let us dig up into VehicleSearchedIndicator and ContrabandIndicator since they are very important features both for modelling and for the assessment of bias/fairness:



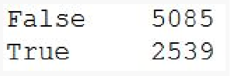
```
False    237607
True       7624
Name: VehicleSearchedIndicator
```

So only around 3% of cars that are stopped are actually searched.



```
False    242392
True       2839
Name: ContrabandIndicator
```

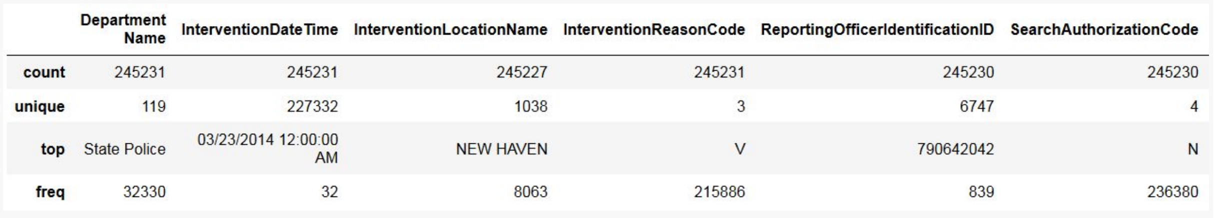
Similarly, around 1% of the cars that are stopped have contraband. But what about the search success rate? For that we look at the value counts of ContrabandIndicator when VehicleSearchedIndicator is True:



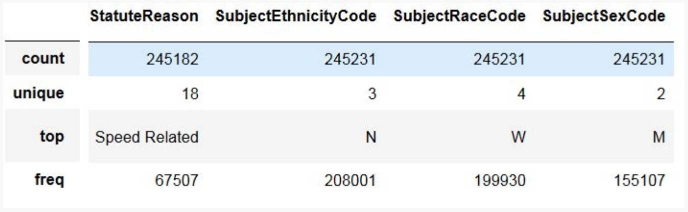
```
False    5085
True     2539
```

So when a car is searched, 33% of the times it has contraband.

We proceed to the evaluation of object features:

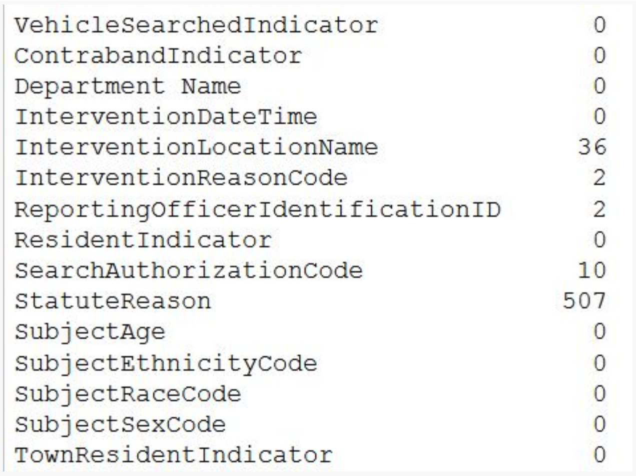


	Department Name	InterventionDateTime	InterventionLocationName	InterventionReasonCode	ReportingOfficerIdentificationID	SearchAuthorizationCode
count	245231	245231	245227	245231	245230	245230
unique	119	227332	1038	3	6747	4
top	State Police	03/23/2014 12:00:00 AM	NEW HAVEN	V	790642042	N
freq	32330	32	8063	215886	839	236380



	StatuteReason	SubjectEthnicityCode	SubjectRaceCode	SubjectSexCode
count	245182	245231	245231	245231
unique	18	3	4	2
top	Speed Related	N	W	M
freq	67507	208001	199930	155107

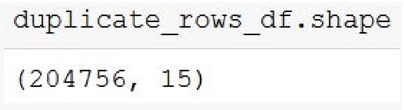
Nothing really stands out. Let’s check for null values:



```
VehicleSearchedIndicator              0
ContrabandIndicator                   0
Department Name                       0
InterventionDateTime                  0
InterventionLocationName             36
InterventionReasonCode                2
ReportingOfficerIdentificationID      2
ResidentIndicator                     0
SearchAuthorizationCode              10
StatuteReason                       507
SubjectAge                            0
SubjectEthnicityCode                  0
SubjectRaceCode                       0
SubjectSexCode                        0
TownResidentIndicator                 0
```

So it seems that 5 features have null values: InterventionLocationName, ReportingOfficerIdentificationID, InterventionReasonCode, SearchAuthorizationCode, and StatuteReason.

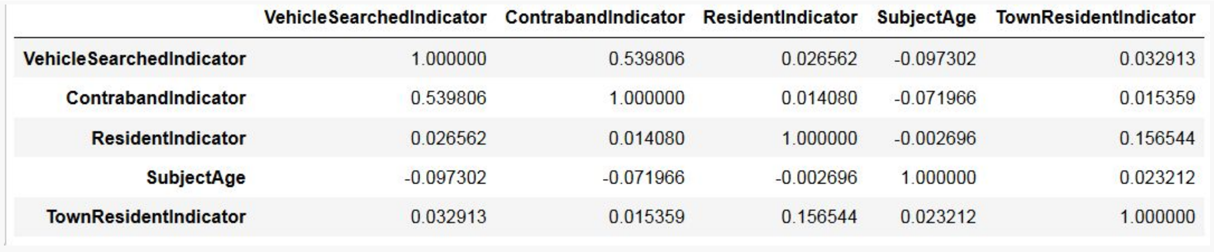
Let’s check if duplicates exist and if so how many observations are duplicated:



```
duplicate_rows_df.shape
```

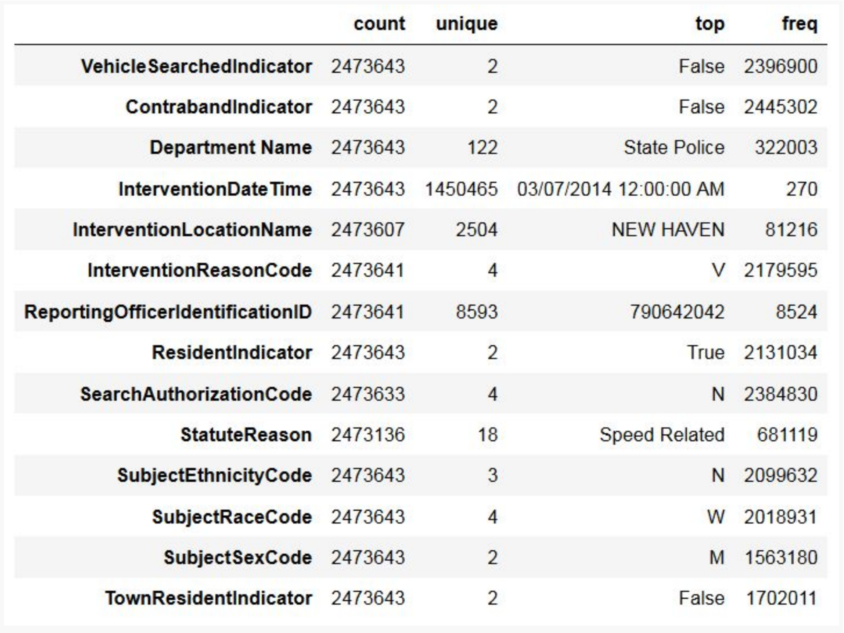
```
(204756, 15)
```

So around 8% of the dataset is duplicated!
After that we look at the correlation between non object features:



	VehicleSearchedIndicator	ContrabandIndicator	ResidentIndicator	SubjectAge	TownResidentIndicator
VehicleSearchedIndicator	1.000000	0.539806	0.026562	-0.097302	0.032913
ContrabandIndicator	0.539806	1.000000	0.014080	-0.071966	0.015359
ResidentIndicator	0.026562	0.014080	1.000000	-0.002696	0.156544
SubjectAge	-0.097302	-0.071966	-0.002696	1.000000	0.023212
TownResidentIndicator	0.032913	0.015359	0.156544	0.023212	1.000000

No suspicious values appear, other than the correlation between VehicleSearchedIndicator and ContrabandIndicator which is around 0.54 and that was expected since you have to search the car to find contraband (except for 2823 cases in the dataset where contraband was obvious).
We finalize the analysis by looking at the cardinality of our features:



	count	unique	top	freq
VehicleSearchedIndicator	2473643	2	False	2396900
ContrabandIndicator	2473643	2	False	2445302
Department Name	2473643	122	State Police	322003
InterventionDateTime	2473643	1450465	03/07/2014 12:00:00 AM	270
InterventionLocationName	2473607	2504	NEW HAVEN	81216
InterventionReasonCode	2473641	4	V	2179595
ReportingOfficerIdentificationID	2473641	8593	790642042	8524
ResidentIndicator	2473643	2	True	2131034
SearchAuthorizationCode	2473633	4	N	2384830
StatuteReason	2473136	18	Speed Related	681119
SubjectEthnicityCode	2473643	3	N	2099632
SubjectRaceCode	2473643	4	W	2018931
SubjectSexCode	2473643	2	M	1563180
TownResidentIndicator	2473643	2	False	1702011

InterventionDateTime has a very big number of uniques which is okay since it is basically an unprocessed timestamp. Other than that, the features with noteworthy cardinalities are ReportingOfficerIdentificationID and InterventionLocationName. However their values seem to be okay taking into account that the dataset has around 2M observations and comes from 122 different police departments.

Business questions technical support

The dataset is a list of observations of stopped cars by the police of Connecticut. It contains the following information: VehicleSearchedIndicator (Whether the vehicle was searched or not); ContrabandIndicator (Search Disposition: Contraband and/or evidence discovered);

Department Name (Name of the police department); InterventionDateTime (Date and time of the intervention); InterventionLocationName (Location of the intervention); InterventionReasonCode (Code for the reason given for stopping the vehicle with the following codes being used: "Investigation: I; Violation: V; Equipment: E"); ReportingOfficerIdentificationID (Reporting Officer Identification ID); Resident Indicator (Whether the subject was a resident of the state); search_authorization_code (Authority to search vehicle, with the following codes: "N-Not Applicable; C-Consent; I-Inventory; O-Other:Probable Cause, Reasonable Suspicion, Plain View Contraband, Incident to Arrest, Drug Dog Alert, Exigent Circumstances"); StatuteReason (Reason given for stopping the car); SubjectAge (Age of the main occupier of vehicle); SubjectEthnicityCode (Officer perception of the ethnicity of subject, with the following codes: "Hispanic: H; Middle Eastern: M; Not Applicable: N or N/A"); SubjectRaceCode (Officer perception of the race of subject with the following codes: "W - White; B - Black; I - Indian America/Alaskan Native; A-Asian/Pacific Islander; U - Unkown"); subject_sex_code (Subject Sex Code); TownResidentIndicator (Whether the subject was a resident of the Town.); InterventionLocationName (Intervention Location Name).

Business questions analysis:
We begin by analysing the precision for the protected classes:
The rounded precision for the female class is 0.37 while for the male class is also 0.37. There is aprroximately 0 difference between the precisions of the sex code classes and then we can then conclude that there is no bias in this class.
The rounded precision for the hispanic class is 0.32, for the middle east class is 0.29 and for the drivers without an identified ethnicity is 0.39. Thus, there is a difference of more than 5% points between the precision of the drivers without an identified ethnicity and the other two ethnicities. We can then conclude that there is bias in this class, drivers without an identified ethnicity are being unfairly less searched than the other two ethnicities.
The rounded precision for the white class is 0.38, for the black class is 0.33, for the asian class is 0.36 and for the indian class is 0.33. The non rounded difference between the white and black class is 4,97% and between the white and the asian class is 5,69%. Thus, there is a difference of more than 5% between the precision of white and asian classes. We can then conclude that there is bias in this class, drivers with white race are being unfairly less searched than drivers with asian race.
Out of the 122 police sub-departments, none simultaneously fulfill the requirement of having a difference of precision in the 3 protected classes of less or equal than 5%.
12 of the police sub-departments had a difference of higher than 90% in the precision of one of the 3 protected classes. They were: CSP Troop I, Darien, ECSU, Enfield, Guilford, MTA Stamford, Meriden, Middletown, Orange, Redding, Rocky Hill, Winsted.
Although it should be noted that precisions are calculated as a percentage and thus are very sensitive to low values (e.g. if there was 1 car with a male driver searched and found with contraband and 1 car with a female driver searched and without contraband, the difference

in precision would be 100%), we believe these police sub-departments to be worthy of further investigation.

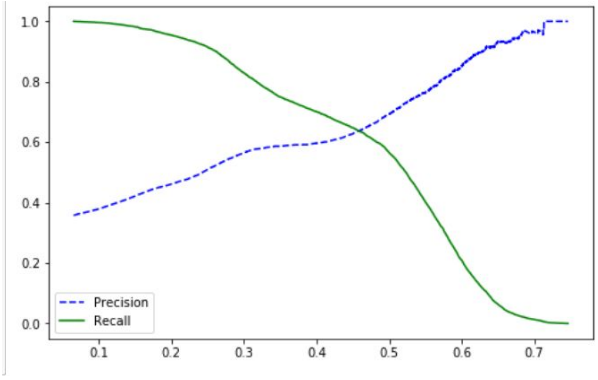
Regarding the police officers, the following ones have a difference of higher than 90% in the precision of at least one of the 3 protected classes (in this analysis, only officers with more than 100 searches were considered): 1000001949, 1000002608, 179, 199, 256, 30233 and 570.

Model technical analysis

In this section we will detail the choice of a threshold of around 0.39 (specifically, 0.3873617971247522).

We began by realizing that our test set (which corresponds to the final folder of the cross validation) does not contain observations of the middle east category of the ethnicity class. As such, the precision, recall and threshold vectors for this category were obtained from the first folder of the cross validation.

Drawing the precision recall curve of our model:



We then found the threshold for which the precision of our model fulfilled the first client requirement (0.5 precision). That threshold is approximately 0.24. Theoretically, with the interval of threshold [0.24, 1] we could guarantee the first requirement. (0.242740481467374 is the exact threshold for which the precision = 0.5).

However, the indian category of the race class has a recall of 0 and precision of 1 for the threshold of 0.69, and as such, we are confined to the interval [0.24, 0.69].

We then compute the maximum percentage discrepancy between the 3 classes for each threshold and choose the threshold that minimizes this percentage. This value is threshold = 0.39.

For this threshold, the global precision is 0.59 (which gives us a 0.1 “buffer” to absorb precision losses in production) and recall is approximately 0.71.

The descrepancy between the sex feature is 1%, between ethnicity is 3% and between race is 7%.

The individual precisions are:

Male: 59%

Female: 58%

Hispanic: 57%

Middle East: 59%

Not applicable ethnicity: 60%

White: 61%
Black: 55%
Indian: 55%
Asian: 61%