Client requirements

Summary

In this report, we present the results of a detailed statistical analysis of historical data, collected by the police department, concerning its traffic stop and search for contraband operations. The aim was to evaluate the existence of a bias against people of certain backgrounds in the search decision pattern used by police officers.
Additionally, we were asked to develop a model to objectively predict the risk of contraband based on the historical data provided. This model should be able to improve the current search success rate while respecting acceptable levels of bias between personal classes (race, ethnicity, and gender). A description of the proposed model and its expected results are presented in this report, the model itself has been deployed on an online platform[1].

Requirements clarifications

Awkward Problem Solutions™ has been asked to comply with 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 department should have a discrepancy bigger than 5 percentage points between the search success rate within protected classes (of race, ethnicity, gender)

3. The largest possible amount of contraband found, given the constraints above.

The reason for the third requirement is self-evident. If we can increase the percentage of contraband found we also increase the certainty of punishment, producing a stronger deterrent effect which prevents more people from committing contraband crimes. From a specification perspective, this is equivalent to ask for a maximization of the model's recall score: for a given number of stops the existing contraband is either searched and found (true positives) or not searched and not found (false negative), and we want to find as much contraband as possible (true positives) given its total amount (true positives + false negative). This requirement provides us with a metric, or scoring, for the evaluation of the classification algorithm, whereas the first two requirements give us constraints that should be complied with.

Regarding the first requirement, it possibly arises from the existence of a cost associated with the searching procedure and a benefit associated with finding contraband. In that case, the rationale for this requirement should be the fact that the benefit resulting from finding contraband (let's call it $2s$) is valued as twice the cost of searching for it (call it s), on average. In this way, since every time a search is made we incur in the cost s, searching is

[1] Available at https://heroku-app-model-deploy.herokuapp.com/

‘cost-effective’ only if the probability of obtaining the expected benefit *2s* by finding contraband is, at least, 50%. Once we have estimated the probability of finding contraband for a given observation, this requirement defines the classification threshold to be used by the classifier algorithm[2], i.e. we will predict contraband and clear the search authorization only if we expect contraband to be found with a probability higher than 50%. The definition of the loss function (the cost and benefit associated with each possible decision) is a client’s exclusive responsibility and it is a central part of any classification algorithm, therefore we will stick to this requirement.

The second requirement addresses a concern with the model’s fairness among protected classes, i.e., the model should perform similarly regardless of the individual classes a subject belongs to. From a statistical perspective, we interpret the search success rate (SR) as a precision score requirement[3]: among the predicted positives (search ‘clearance’), success is measured by the ratio number of findings (true positive) / number of searches performed (true positive + false positives). Another remark regarding this requirement: we believe that it is more appropriate to measure the discrepancy between protected classes in relative values (i.e., maximum difference of 5 percent), than in absolute values (5 percentage points), as has been required. We say so, because we think that a difference between a precision of 99% and 93% within a class (6% or 6 p.p.) would be more acceptable than, for example, a difference between a precision of 5% and 1% (80% or 4 p.p.), despite the fact that the last case ensures compliance with the requested requirement. We certainly approve this fairness concern, but we can anticipate that we have not been able to ensure its compliance, neither at police department level nor at a global level. We have assessed this metric, both for the police officers current performance and for our model’s expected performance, but we have not been able to comply with it. Even so, we have suggested improvements to the model that would allow us to deal with this requirement at a future time.

Dataset analysis

General analysis

To perform the requested analysis we were provided with a dataset concerning car stops, including whether the car was searched, and if any contraband was found. This dataset comprises records of 2,473,643 car stops between October 2013 and May 2018 and is described by 16 variables. A brief characterization of the dataset is presented below.

---

[2] Notice that the phrasing is not completely clear, as the sentence “a minimum 50% success rate for searches” may be interpreted as a request for an overall success rate of 50%. This interpretation would lead to constraining the precision score of the model to a minimum value of 50%. However, the clarification in parenthesis seems to indicate that the minimum 50% success rate should be attained ‘in every single search’, and not overall. We will follow this interpretation, which brings us to the option of defining a classification threshold at 50%. Notice the two concepts are closely related, since an increase of the threshold selected to classify each observation will also increase the overall success rate, and vice versa.

[3] Throughout the report, we use the terms “search success rate”, “success rate” and “finding rate” with the meaning of precision score.

- ➔ *VehicleSearchedIndicator*: 'True'/'False' variable indicating whether the vehicle was searched. We have 76,743 searches in a total of 2.47 million stops, i.e. only about 3,1% of stopped cars are searched.
- ➔ *ContrabandIndicator*: 'True'/'False' variable indicating whether contraband and/or evidence was discovered. True for 28,341 observations, which amounts to 1.15% of total and 33.25% among searched vehicles.
- ➔ *Department Name*: There are 122 police departments in the dataset. The number of traffic stops per department is very dispersed. While the 'State Department' accounts for 13% of observations, there are 12 departments with less than 1,000 observations and, between these, 6 departments with less than 100. The average value of traffic stops per department is 20,275 and the 50th percentile is 14,754 stops.
- ➔ *InterventionLocationName*: Location of the intervention, 1,500 places reported. Since each police department may perform traffic stops at different places, in this case, the dispersion in the number of traffic stops per local is even larger than above. We have more than 50% of the locations with a single stop reported; 76% locations with 5 or fewer stops; while for those 75 locations above the 95th percentile the average number of stops is 26,254.
- ➔ *InterventionDateTime*[4]: Date and time of the intervention, ranging from October 2013 and May 2018, with a break between April 2015 and September 2015.
- ➔ *InterventionReasonCode*: Code for the reason given for stopping the vehicle. Three distinct values: 'violation' (88% of the stops, i.e. 2,179,595 observations); 'equipment' (10%); 'investigation' (2%).
- ➔ *StatuteReason*: Reason given for stopping the car. Fifteen distinct classifications. Most frequent are: 'speed related' (in 27.5% of stops); 'defective lights' (9.2%); 'registration' (9.2%); 'cell phone' (9.0%); 'moving violation' (7.7%); and 'traffic control signal' (7.2%). 'other'/'other error'/missing amount for 8.9%.
- ➔ *SearchAuthorizationCode*: Authority to search the vehicle. Three distinct values: 'Consent' (35.2% of searches); 'Inventory' (20.5%); 'Other' (40.0%), which encompasses probable cause, reasonable suspicion, plain view contraband, incident to arrest, drug dog alert or exigent circumstances. Missing values or 'Not Applicable' in 4.3% of the searches.
- ➔ *ReportingOfficerIdentificationID*: There are 8,593 distinct officers in the dataset.
- ➔ *ResidentIndicator*: 'True'/'False' variable indicating whether the subject was a resident of the state. In the complete set, 86% of drivers stopped are state residents, whereas considering only the searched vehicles 91% are from state residents.
- ➔ *TownResidentIndicator*: 'True'/'False' variable indicating whether the subject was a resident of the town. In the complete set, 69% of drivers stopped are town residents, whereas considering only the searched vehicles 60% are from town residents.
- ➔ *SubjectAge*[5]: Age of the main occupier of the vehicle. The average age is 39 while the median is 36.
- ➔ *SubjectEthnicityCode*: Officer perception of the ethnicity of the subject. Two distinct values: 'Hispanic' (13.3% of all subjects, i.e., 328,450) and 'Middle Eastern' (1.8% or 45,561 subjects). 'Not Applicable' accounts for 85% of observations.

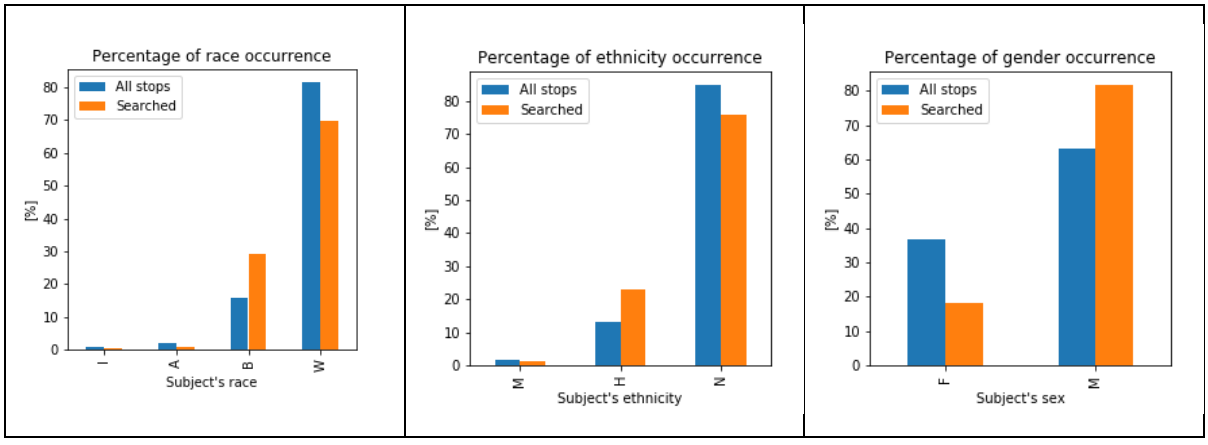
[4] Plots of the monthly evolution of traffic stops and number of searches available in the annexes.
[5] Histogram of subject's age available in the annexes.

- ➔ *SubjectRaceCode*: Officer perception of the race of the subject. Four distinct values: 'White' (81.6% of subjects), 'Black' (15.6%), 'Asian/Pacific Islander' (2.0%) and 'Indian America/Alaskan Native' (0.8%). No missing or unknown data, all observations belong to one of the four classifications.
- ➔ *SubjectSexCode*: Subject's gender. Two distinct values: 'Male' (63.2% of subjects) and 'Female' (36.8%). No missing or unspecified sex.

The result of the analysis of unexpected observations and missing values is presented in the The Annexes - Dataset technical analysis.

Business questions analysis

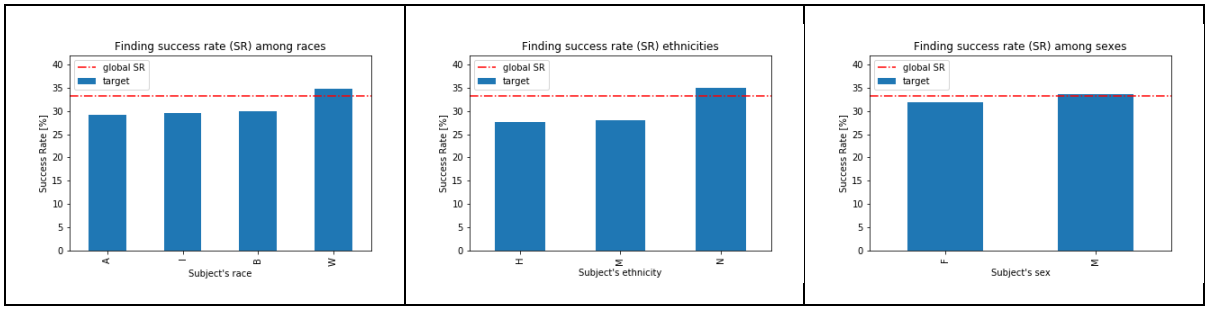
The first concern manifested in the briefing was the suspicion that the search decision criterion used by police officers might be biased against people of certain backgrounds. Considering being searched as a sample process from the complete population (in our case, drivers who are pulled over), bias happens if individuals are not equally likely to be selected, particularly if the difference in the probability of being searched arise from belonging to a certain class (race, ethnicity or gender). Under this scenario, certain population classes will be underrepresented or overrepresented in the sample (i.e. in the searches). The plots presented below show us that this effect is indeed happening. For example, 'Black' subjects are overrepresented in searches when compared to all traffic stops since the presence of this race class becomes higher when we consider only searched drivers instead of all drivers. The same occurs with the 'Hispanic' class and 'Male' class, and the opposite occurs with 'Whites', for example.



Having said that, we must remember that the purpose of a vehicle search is not to draw a random sample from the population, it is to find contraband or crime evidence, and therefore it may be reasonable to accept that the search criterion favors (in other words, biases) certain classes over others (as well as certain characteristics, such as driver's age or driving during specific hours of the day). In this case, the rationale is the following: if we know that the presence of certain characteristics makes contraband or crime evidence more likely to be found (i.e., if these characteristics are correlated with evidence finding), then we may use this information to increase the search success rate and contraband found, as intended. The search decision will be biased, but it may be considered fair. This behavior may even be

inevitable when we want to increase certain performance metrics, as requested in the briefing.

To assess the current decision-making process in terms of fairness, we have first of all to define the concept. Taking into account the concerns expressed in the briefing, we believe the Police intention is to have a process which doesn't "impose a relative disadvantage on persons based on their membership in some salient social group, e.g., race or gender"[6], i.e. a process which doesn't discriminate against any of the protected classes. Secondly, we must be able to find a way to measure it, using some precise metrics, so that we can assess both the current process and our proposed model. The adoption of the precision score as a criteria for fairness has been mentioned in the second requirement, then we should use it also for the assessment of the current method. It also happens to be an adequate metrics for the assessment of discrimination in te specific context we are dealing with[7], so we will stick with it. In the three plots below, we present the search success rate among the protected classes, where we can see that this metric has considerable differences between classes. Except for the case of gender, the overrepresented groups in the searched subset, 'Black' and 'Hispanic', actually have a lower search success rate when compared to other groups, as 'White' or ethnicity 'Not Applicable'. These results indicate that the current search criterion is unfair, at least according to the adopted metric. If, in the previous chapter, we argued that it could be acceptable to have a biased search decision criterion, what the plots below show us is that the existing bias was either not justifiable, or was taken too far, resulting in an unfair discriminatory system. The exact values are presented in Annexes - Business questions technical support.

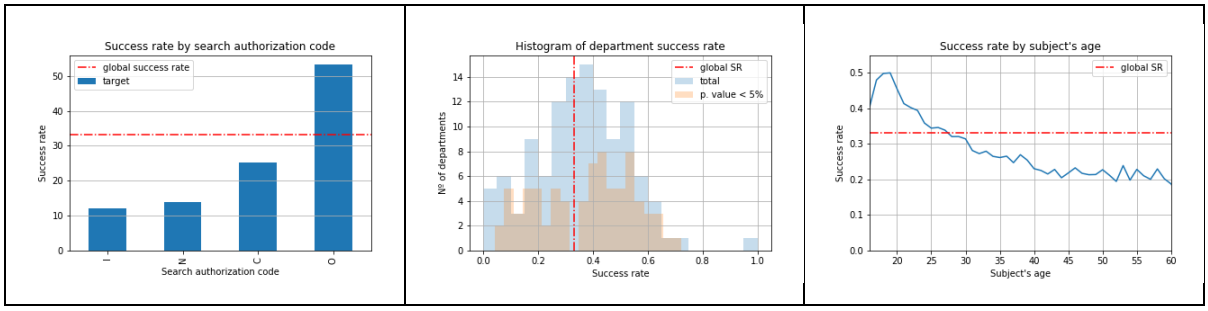


We will now focus on our proposed model. First of all, we investigated the predictive power of each variable for contraband detection, in other words, whether or not the probability of finding contraband changes when a variable changes its value. Based on this analysis we have selected the features *Department Name*, *SearchAuthorizationCode*, *SubjectAge,* *StatuteReason*, and *InterventionDateTime* to be included in our model. The plots below show us how the presence of contraband changes within the selected features (for the last two features the plots are presented in Annexes - Business questions technical support). If, when a feature takes a given value, the probability of finding contraband becomes significantly lower or higher than when we consider all searches (we named it global success

[6] Standard normative definition used for discrimination.
[7] See, for example, the fairness decision tree at http://www.datasciencepublicpolicy.org/projects/aequitas/

rate and represented it by a red dashed line), the model can use this knowledge to improve its performance. For the selected features we see that is indeed the case.

So far we have not mentioned the statistical significance of the presented data, but regarding the *Department Name* it becomes important to mention this concept. The problem with *Department Name* is that we have departments with records as low as 1 stop and 10% of them performed less than 40 stops. This means that some departments may have a very high (or very low) success rate just by chance and, in this case, the feature becomes useless since we can't get much information from it. It is similarly as when we toss a coin: getting 3 tails out of 10 tosses tell us nothing; but if we get 3 tails out of 100 tosses we will bet the coin is not a fair one. To detect this effect, we have performed for each department the binomial test[8] that its success rate was equal to the global success rate (i.e. considering all searches). We found that the test was false, with a significance level of 5%, for 80 out of 117 departments (represented by the red bars in the figure). What the binomial test tells us is that, although some departments may have a higher or lower rate just by change, that is not the case for most of them, reason why the feature is informative and we decided to use it.



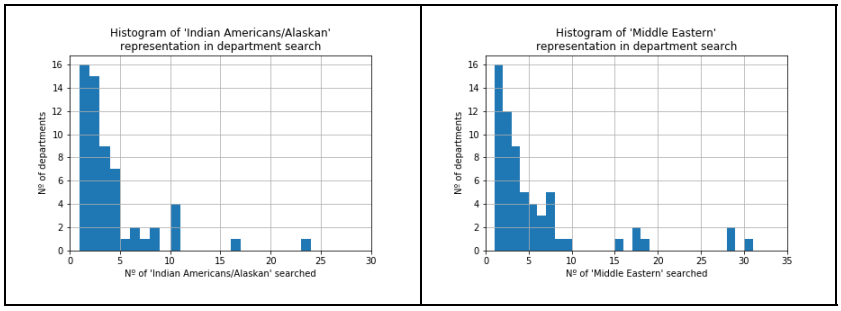
We will mention briefly all the other features that have not been used. Given our intention to develop a fair model, we decided not to use SubjectRaceCode, SubjectEthnicityCode, and SubjectSexCode, removing in this way the possibility of a direct discrimination. Although it is not a sufficient measure, as we will see, it is a first step in the direction of a fair model. Besides, when they were included in the final model, just to compare the results, the differences in the score metrics were completely negligible. For the reporting officer's ID, we notice that 20% of them were responsible for only one stop and 65% for less than 10 stops. What these numbers show us is that for the greater fraction of police officers their individual success rate would not be statistically significant, and for this reason it has not been included. For the *InterventionLocationName*, to some extent it reproduces the *Department Name* information. Among the 1,500 different locations present in the dataset, 86% of them are serviced by a single department. This means that using *InterventionLocationName* together with *Department Name* is somehow redundant. Even though we will lose some information, we have opted for the last feature because it has a lower number of distinct values (122 against 1500). *ResidentIndicator* and *TownResidentIndicator* appear to have some predictive power, namely a slight increase in the success rate when any of these features takes the value 'False', yet when included in the model its performance doesn't change, and therefore were rejected. A very similar case happened with *InterventionReasonCode*.

[8] https://en.wikipedia.org/wiki/Binomial_test

Conclusions and Recommendations

As we have demonstrated in this section, some of the features present in the dataset clearly enhance our predictive power to find contraband. In this way, we will be able to use this fact to improve search decision process, as defined by first and third requirement metrics.

Regarding the second requirement, we start by saying that it is not reasonable to request its compliance at a department level, and instead we recommend that it is attempted at a global level. When we look at the minority classes representation in the searches of each department individually considered, we find that the values are extremely low, as it is shown in the plots below. A great portion of the departments don't even have any search for these minorities (50 departments don't have, in the case of 'Middle Eastern', and 58 in the case of 'Indian Americans/Alaskan'), which would lead to a great lack of statistical significance if we tried to use these values.



Modeling

Model expected outcomes overview

In this section, we present the foreseen outcomes of our model based on its performance on previously unknown data. We expect the model to output a search 'clearance' for around 30% of all search requests sent by police officers. Notice that this result is not surprising since it is similar to the current success rate. If it was possible to build a perfect model, it would also clear around 30% of all search requests, but exactly those where contraband would be found. This is not the case of our model and wouldn't be the case of any real model. Among the cleared searches, we anticipate that contraband will be found for 60% of them, i.e., we expect to comply with the first requirement with a comfortable margin. These findings are expected to amount to 50% of the existing contraband (considering those cars for which a search request is submitted to the model because we have no information for the remaining ones), meaning this was the highest score, as defined in the third requirement, we were able to achieve. In the table below we present a comparison between four methods for search clearance, including the current method and our proposed model.

	Current method	Random search[9]	Our model	Perfect model
Percent. of searches	100%	Search Rate	29%	33%
Search success rate	33%	33%	60%	100%
Contraband found[10]	100%	Search Rate	51%	100%

The main highlight of the results is that our model was able to detect 51% of the existing contraband while performing only 29% of the searches. This performance will allow the police to reduce the resources with searches by more than ⅔, when compared to the current process, penalizing the number of findings in less than ½.

Regarding the fairness requirement, as we have said before, we haven't been able to address it in our model, although it would be feasible to attain compliance at a global level (considering all departments together), or at least to improve the current status. We expect our model to keep more or less the same level of discrimination (as defined in the second requirement) as the current search decision process. This result is certainly disappointing, and it is not a desirable result, but it was the best possible one, so far. In the Annexes - Business questions technical support, we present the comparision of the discrimination level between our model and the current process. It may seem that our model has worsen the discrimination metrics, since its absolute value as increased in most of the cases. However, when we consider the discrepancy in the search success rate using relative values, as suggested before, we see very slight improvements.

Model specifications

To develop the model, we have split the data into a training set with 60% of the observations and a test set with the remaining observations. Five features from the original dataset have been selected: Department Name, SearchAuthorizationCode, StatuteReason, SubjectAge, and InterventionDateTime. Below, we describe the preprocessing steps implemented in each of the features.

- ➔ *Department Name*, *SearchAuthorizationCode*, *StatuteReason*
  - Text cleaning: Non-letter characters are replaced by spaces; multiple spaces are replaced by a single space; leading and trailing spaces removed; string converted to lower-case (implemented with a user-developed class).
  - Ordinal encoding: since we are dealing with categorical text data and scikit-learn classifiers don't accept text, we had to convert it to numerical values.
- ➔ *SubjectAge*

[9] A random model which give officers a search clearance with a probability given by Search Rate.
[10] Percentage of existing contraband among searches vehicles that different models are able to find. Naturally, the current process finds all the existing contraband among searches vehicles, precisely because the vehicles have all been searched.

- Binarization: To reduce the number of existent values a binarization was used. Classification as high success rate ages (27 or below) and high finding rate hours (remaining hours) and low rate ages (above 27).

➔ *InterventionDateTime*

- Hour of the day: From date and time we have selected only the hour of the day.
- Binarization: Classification as low finding rate hours (from 2 a.m. to 11 a.m. together with 4 and 5 p.m.) and high finding rate hours (remaining hours).

At the end of the preprocessing phase missing values, if existent, are replaced by the feature most frequent value, a convenient method both for categorical variables and numerical ones. After these steps, the preprocessed training data was fed to a Random Forest Classifier.

In order to introduce slight improvements in the model performance, we have run a grid-search on a 6-fold cross-validation process. The trained model was then pickled, has a way to persist for future use in the deployment part without having to retrain it. When deployed it will receive observations sent by police officers and return a 'Clearance'/'No clearance' instruction.

Analysis of expected outcomes based on the training set

We have tried different alternatives for the partition of the data between the training set and the test set. Based on the comparison of the model score metrics[11] (precision, recall, and accuracy) for each of the sets, we ended up using a training set with 60% of the observations. The most relevant scores, given the client's first and third requirements, are precision of 0.60 and a recall of 0.51. Since the client requirements led us to define a specific threshold, the dynamic evaluation of the model is not so relevant in this case. Anyway, these results have been presented with more detail, including the dynamic evaluation, in Annexes - Model technical analysis.

Alternatives considered

We have tried two main approaches to the development of the classification model. The first and simpler approach, since we were dealing mainly with categorical features (even *SubjectAge* and *InterventionDateTime* can be considered categorical after preprocessing), some with a high cardinality like *Department Name*, was to leave the features more or less in their natural state and use a classifier that could deal with them, like the Random Forest Classifier. This ended up being our most successful model and the option we have implemented. It was presented in the previous sections.

The approach described above is not adequate to be implemented with some algorithms, like the Logistic Regression, that is more suited for numerical variables (i.e., in which the numbers have a quantitative meaning and are not exclusively a label attributed to the observation). In order to use those kinds of algorithms with categorical variables, the most

[11] See Model scores on training and test sets, in Annexes - Model technical analysis.

common solution is to apply a one-hot encoding to them. In our case, since we have used Department Name (117 categories), SearchAuthorizationCode (3 categories) and StatuteReason (15 categories), this means we have replaced 3 features by 132. To use these algorithms another advisable preprocessing step is to perform some kind of feature standardization, which we did. As expected, the prolific number of features was detrimental to the performance of the algorithm.
In order to correct this problem, we have tried to approaches to reduce the number of features (i.e. select the more informative ones).

1) Still use all variables, but impose a Lasso penalty with increasing strength.
2) Select the K best features using the corresponding class from scikit-learn[12].

This approach was simultaneously more complex and less performative than the previous one, which is the reason why we have not implemented it.

A third approach, which we have not attempted, would be to find a meaningful way of converting the categorical variables into numerical ones. We have in mind, for example, mapping each category to one or more relevant values related to that category (e.g. replace the name of each police department by its average success rate in the training set), and then try to different algorithms suited for numerical variables. In this case we would have to do regularization and standardization of variables as well.

For the limitation of success rate discrepancy between classes, two different alternatives have been imagined.

1) To introduce a dynamic threshold instead of a fixed one. This approach would have to be implemented in the web application, since it requires an evaluation of the previously classified observations. The logic would be the following: if the discrepancy between two classes gets close to the maximum admissible level, we would increase the classification threshold for the class with lower precision and/or reduce the threshold of the other class.
2) To introduce a penalization term in the optimization objective of the classifier. In a similar way that Ridge and Lasso regularization push the algorithm to drop the less relevant features, we would need to define a penalization term that would push the algorithm to reduce the discrepancy in the success rate between protected classes. In this case, the solution would be implemented at when training the classifier, without any change in the web application.

We regret that we didn't have enough time to try any of these approaches. We are very confident that it would be possible to attain the desired discrepancy levels, but of course to a certain level at the expense of model score results.

[12] https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html

Known issues and risks

Subjects may adapt to the model. For example, if for older people the chance of being searched is low, because the success rate among older people was low on the training set and the model learned that, then people might realize this characteristic of the model and start using it to their advantage. A simple way to mitigate this risk would be to include a random element in the 'Clearance'/'No clearance' decision (i.e., to give the officer a 'Clearance' instruction if the classifier's predicted probability is greater than the threshold, or if a random variable decides so), such that even a person who is very knowledgeable about the classifier behaviour would never be sure of its output.

We have trained our model on a sub-sample of stopped cars, those which have been searched by police officers, and this sub-sample we had access to might not be an accurate representation of our population (stopped cars). If this happens to be the case, we are facing a problem of selection bias. This effect could undermine the ability of our results to be generalized to the complete population. However, since this is not the purpose of our model because the whole process is going to be maintained - a first selection of subjects by the police officers, and only then a search decision provided by the classifier - we don't expect the production results to differ significantly from the results on the test set.

Even if we don't expect a direct impact on the results, this problem of selection bias might be blinding us to a significant part of the population. When the training set is a random sample from the population, then the classifier predicted probabilities are expected to be unbiased. However, if we admit the possibility that there was some bias in the drawing for the training set, i.e. if the police search decision was itself biased, then the predicted probabilities will reflect this effect. A possible solution would be, in the same way as before, to introduce some random mechanism on the selection process and then use the true classes of these random selected observations to retrain the classifier.

Model Deployment

Deployment specifications

We have integrated our model in a web application developed with Flask and deployed it to heroku, in order to make it online available[13]. The web application has two basic functionalities:

- ➔ Predict[14]: Returns a search 'clearance'/'no clearance' instruction for the specific observation received. It is the user's responsibility to provide the model with the observation's data. A classifier, which has previously been trained and uploaded to the model, is fed with the observation and outputs a predicted probability of finding contraband. 'Clearance' is provided to the user if the predicted probability exceeds

[13] At https://heroku-app-model-deploy.herokuapp.com/
[14] More details provided in Annexes - Model deployment additional specifications.

the 50% threshold (see Requirements clarifications), 'no clearance' is returned otherwise.

- ➔ Update[15]: The deployed model accepts user updates about the true class of previously sent observations. This functionality allows us to perform a later assessment to the model performance, and eventually retrain the classifier, using the received updates.

In order to keep a record of the model's operation we have connected it to a database table having the following columns:

- ➔ Observation ID: Integer value which uniquely identifies the observation. It is the responsibility of the user requesting the predict service to provide this unique ID (together with the observation itself).
- ➔ Observation: Observations data, in json format, provided by the user in the predict request.
- ➔ Predicted probability: The probability of finding contraband, as predicted by the classifier running in the model.
- ➔ True class: The true class of the observation (*ContrabandIndicator* 'True'/'False'). Available only if the update service has been provided by the user for the observation (not required), otherwise is 'NULL'.

In order to make the predict and updated services operational, as well as to connect to the database, when we launch the model the following steps are performed one time:

- ➔ Create an instance of a PostgreSQL database (if running locally then an instance of a Sqlite database in the model's folder is created instead). This instance is used to manage the connections to the database. We have used the peewee ORM.
- ➔ Define a class with the structure of the database as described in the previous paragraph.
- ➔ Load the trained classifier (a pipeline with the feature preprocessing and the classifier itself - as described in subsection Model specifications - has been previously fitted to the training set and serialized, using scikit-learn joblib's dump function).
- ➔ Start running the web application, making the predict and update services available.

Known issues and risks

- ➔ We have implemented no security measures whatsoever. Any person with knowledge of the application can access the predict and update services, populating the database with trash observations or replacing the true values provided by trustworthy updates.
- ➔ We currently have a limit of 10,000 observations in our database. It is reasonable to expect that this limit can easily be exceeded.
- ➔ The very simple interface to our web application, created with Flask, is really not user-friendly and will probably hinder the officers job.
- ➔ Using Flask micro-framework, while providing us with a quick and simple way to set up a HTTP server focusing on including only what we really need, lacks the structure

[15] More details provided in Annexes - Model deployment additional specifications.

and scalability provided by a standardized framework, like Django, with a large online community and extensive documentation.

Annexes

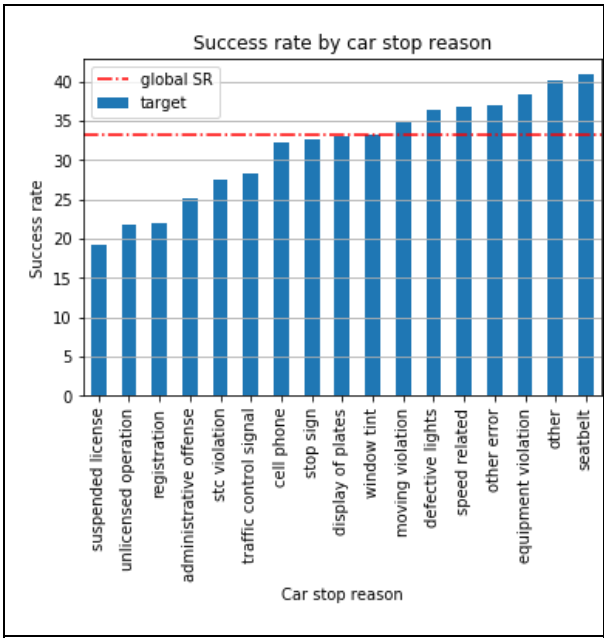
Dataset technical analysis

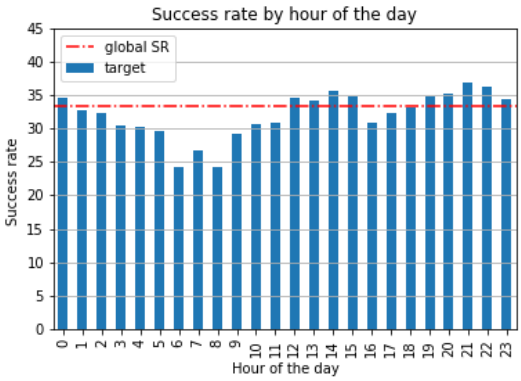
Unexpected observations or missing values in the dataset:

- ➔ *ContrabandIndicator*: Expected to be ‘False’ when no search was performed (i.e, *VehicleSearchedIndicator* equal to ‘False’), but there were 2,823 observations with contraband finding without search indication. For training the model these were ignored since we considered only those observations with VehicleSearchedIndicator equal to ‘True’.
- ➔ *SearchAuthorizationCode*: Expected to be missing or ‘Not Applicable’ when no search was performed (i.e, *VehicleSearchedIndicator* equal to ‘False’). However, 15,360 of stops without search indication have a valid *SearchAuthorizationCode* classification.
- ➔ *TownResidentIndicator*: All state residents were expected to be town residents as well. Nevertheless, 44,904 cases exist (1.82% of the dataset) where town residents are not state residents.
- ➔ *SubjectAge*: 1,117 unexpected observations with the subject’s age below 16 years old (4.8%). Also no subject older than 99 years old. It is likely that all subjects with more than 100 or more years of age have been classified as being 99 because the number of subjects with this age is 661, while the number of subjects with the age of 98 is 132 and with the age of 97 is 216. Another possible justification for the unexpected ages is that ages were computed from the date of birth and the year of birth was recorded with only two digits.

Business questions technical support

Features used in the model for which the plot was not included in the report.





Comparision of the discrimination level between our model and the current process.

	Feature *SubjectRaceCode*	
	Current method	Our model
	Precision	Precision
'White' [%]	34.68	61.99
'Black' [%]	29.97	54.82
'Asian/Pacific' [%]	29.11	54.76
'Indian American' [%]	29.54	65.71
Max. difference [p.p.]	5.58	10.95
Max. difference [%]	16.05	16.67
Std. Deviation [p.p.]	2.59	5.45

	Feature *SubjectEthnicityCode*	
	Current method	Our model
	Precision	Precision
'Not applicable' [%]	34.99	60.64
'Hispanic' [%]	27.71	57.64

‘Middle Eastern’ [%]	27.97	50.00
Max. difference [p.p.]	7.28	10.64
Max. difference [%]	20.82	17.55
Std. Deviation [p.p.]	4.13	5.48

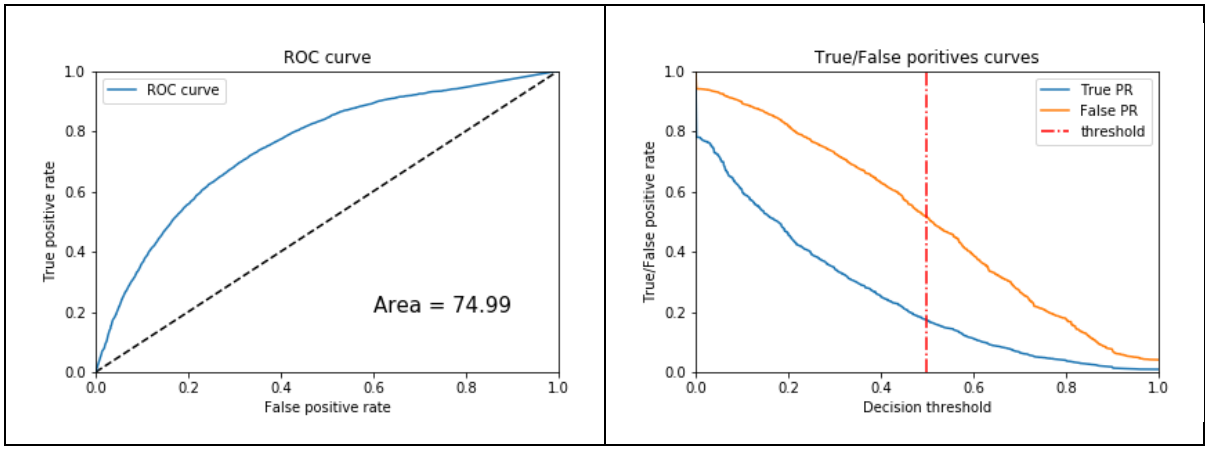
	Feature *SubjectSexCode*	
	Current method	Our model
	Precision	Precision
‘Male’ [%]	33.54	59.73
‘Female’ [%]	31.96	61.13
Max. difference [p.p.]	1.57	1.40
Max. difference [%]	4.70	2.29
Std. Deviation [p.p.]	1.11	0.99

Model technical analysis

Model scores on training and test sets

	Training set (60%)	Test set (40%)
Precision	0.7153	0.5998
Recall	0.6130	0.5144
Accuracy	0.7916	0.7231

Model dynamic evaluation plots.



Decision tree example.

Model deployment additional specifications

Although the features listed below are expected, they are not required, since the model will select only *Department Name*, *SearchAuthorizationCode*, *StatuteReason*, *SubjectAge* and *InterventionDateTime*.

['Department Name', 'InterventionDateTime', 'InterventionLocationName', 'InterventionReasonCode', 'ReportingOfficerIdentificationID', 'ResidentIndicator', 'SearchAuthorizationCode', 'StatuteReason', 'SubjectAge', 'SubjectEthnicityCode', 'SubjectRaceCode', 'SubjectSexCode', 'TownResidentIndicator']

Glossary of terms and formatting

Definition of the terms and text formatting options that have been used in the report.

Italic: Feature names (e.g., *SubjectRaceCode*)
Quotation marks: Different classes in a given feature (e.g., ‘Black’ or ‘White’ in *SubjectRaceCode*)

We use the terms “search success rate”, “success rate” and “finding rate” with the meaning of precision score

Complete with most relevant terms.