Summary

The model developed for this project was deployed and received requests for authorization of searches during the year of 2020. For a subset of these the true outcome of the search was later provided by the client. A detailed analysis of the population of those requests, the performance of the model and the fulfilment of the client requirements is presented in the following sections. In this section we summarize the most important conclusions.

The API of the model received requests from only 4 of the 41 police stations analyzed in the previous report : "City of London", "Nottinghamshire", "Durham" and "Cambridgeshire". Some differences were also found for the features of the model between the received requests and the population used to train the model, which are made explicit in section "Population Analysis". This prevented a fair and direct comparison between the model expectations and the actual results, but nevertheless some conclusions could be established.

The model rejected a very small fraction of search requests (0.5%). It basically replicated the policy for stop and searches on which it was trained. It satisfied the following requirements:

- The probability that a criminal offence is detected in an authorized search is > 10%.
- The probability that a non-authorized search corresponds to a criminal offence is very low (1%).
- The same stop and search policy is used for all stations.

It didn't satisfy the requirements related to the absence of discrimination between ethnic groups and to the uniformity of the search success rate between stations and objects of search:

- The difference in the model precision between different ethnic groups for a given station is higher than 5%.
- The difference in the average precision between stations is higher than 10%.
- The difference in the average precision between objects of search is higher than 10%.

This was expected since these requirements were also not satisfied for the validation dataset of the model, despite the model not using protected characteristics (gender, age, ethnicity) to authorize searches. It could not overcome the imbalance in the distribution of the features used to assess if search should be authorized or not. In section "Next steps" we discuss possible solutions to this problem and improvements to the model.

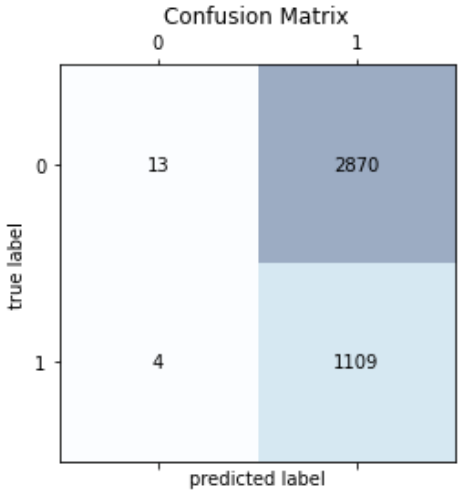
No problems were detected in the deployment of the model. For the requests that were monitored the API did not receive invalid inputs and was able to handle missing values in the features. The response time was lower than 50 ms.

Result Analysis

Model performance

The API received 7902 requests for the **should_search** endpoint. Of these, 3996 have received a request for the API endpoint **search_result** containing the true value of the outcome. The confusion matrix for this subset of requests is shown below.

Figure 1 : Confusion matrix of search outcome for production data



The model authorized the vast majority of searches. Only 0.5% of the searches were not authorized. There were a lot of type I errors : 72% of the authorized searches were false positives, that is, the true value of the outcome was unsuccessful. The rate of type II errors was lower: 24% of the non authorized searches would correspond to a true outcome. Therefore the model prefers to authorize a search which would be unsuccessful than to reject a search than would be successful. The performance metrics of the model are presented in the table below.

Table 1 : Performance metrics of the model : expected (validation dataset) and actual (production dataset)

	Expected (validation dataset)	Actual (production dataset)
Precision	23%	28%
Recall	93%	99.6%
ROC score	0.66	0.60

Therefore the model achieved a low precision but a high recall (since it rejected very few searches). The ROC score is only 10% higher than a random classifier, so the model is not very good at predicting if a search will be successful or not. The performance metrics evaluated on the production data are similar to the ones observed in the validation dataset: The precision increased 5%, the recall increased 6% and the ROC score decreased 6%.

The performance of the model satisfies the requirements and agrees with expectations based on validation data. However, it would be worth considering an alternative model with better ROC score and which rejects more searches, thereby increasing the precision.

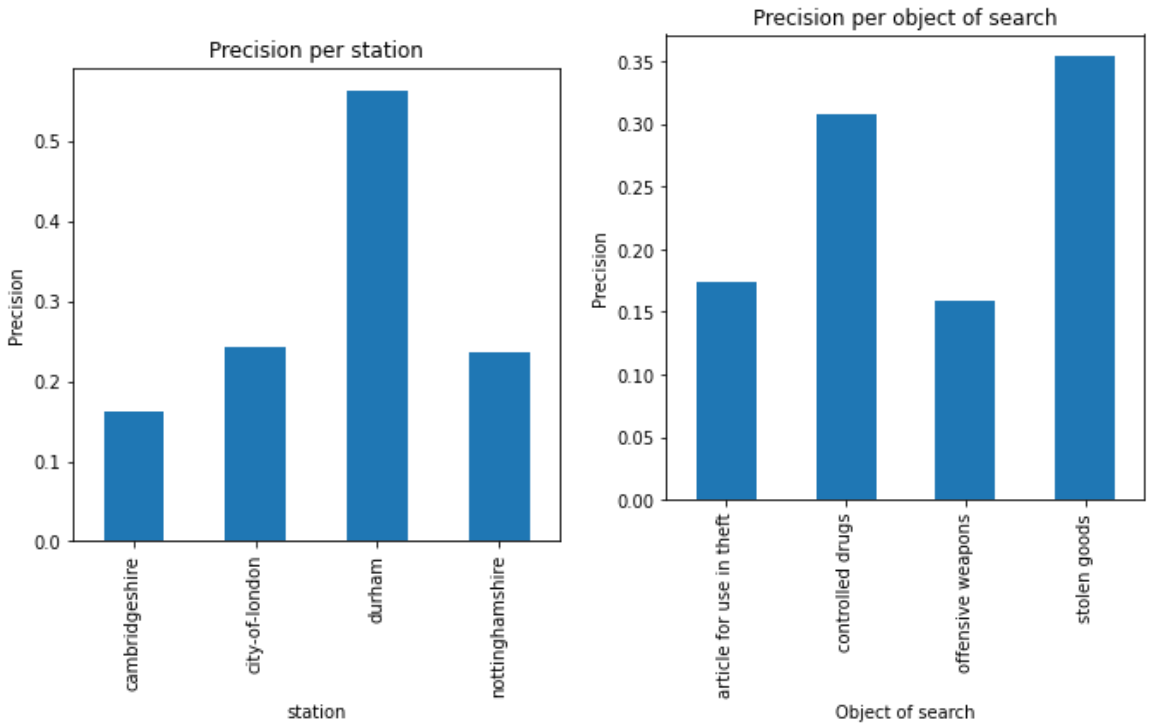
The client sent us their measurement of precision for the model. It was 24%, which is very close to the one expected from the validation data. The difference in precision between the client measurement and our measurement might be due to a difference in the subsets for which the true outcome is known. It is possible that the API missed some requests for endpoint **search_result.**

Success on requirements

The client required that the model provided a precision above 10% for every station. This requirement was satisfied. However the precision is not uniform among the stations. The average precision is plotted below for each station.

Figure 1. Search precision per station

Figure 2. Search precision per object of search



The client required that the difference in the average precision between stations should not be greater than 10%. This requirement was violated because the precision is much greater for station “Durham” than for the other stations. “Cambridgeshire” station also has a lower precision than stations “City of London” and “Nottinghamshire”.

The precision is also greater than 10% for every object of search, but imbalanced between objects. The precision for “Article for use in theft” and “Offensive weapons” is lower than for “Controlled Drugs and Stolen goods”. Since the disparity is greater than 10% the requirement on uniformity of precision was violated. These disparities were also present for the validation dataset of the model, so it was expected that these requirements would not be fulfilled.

The discrimination metric for ethnicity was evaluated on the production dataset for which the true outcome was known and used the same definition of the previous report. For 2 of the 4 stations (Cambridheshire and Durham) the number of data samples per ethnicity group and station was lower than 30, so discrimination was not computed.

Table 1: Discrimination metric of the model on production data for each station and ethnicity

Discrimination metric	Ethnicity			
Station	black	asian	other	mixed
city-of-london	0%	-5.6%	-2.4%	
nottinghamshire	-3.8%	-4.6%		-5.6%

In the table above blank entries correspond to groups with insufficient statistics. For the station “City-of-london” there is discrimination of asians w.r.t to whites. This was not expected from validation data, which might be an indication that feature distributions have changed. For “Nottinghamshire” there is discrimination of mixed ethnic groups w.r.t to whites. The discrimination for a specific station and group is never greather than 6%, better than in validation data. The median of the total discrimination is greather in production that in validation. The spread of total discrimination is much lower for the production data. So there is a nominal improvement in the spread over the model expectations. However we could only make comparisons for 2 out of the 41 stations, which is quite limited. The requirement for non-discrimination between ethnic groups was violated as expected from the validation of the model.

Table 2 : Discrimination metrics of the model for etchnicity in production data.

# Stations with discrimination	# Stations with positive discrimination	# Stations with negative discrimination	Median of total discrimination	Spread of total discrimination
17	11	9	4.7%	48.0%
2	0	2	-11%	-5.9%

The discrimination metric of the model for each station and gender is presented in the table 3. For all the stations it is lower than 5%, which means that is there no significant discrimination between gender groups. Therefore the requirement of no discrimination is fullfiled for gender groups. This is contrary to the expectation from the validation dataset. However, the comparison is made for different sets of stations.

Table 3: Discrimination metric of the model on production data for each station and gender

Discrimination metric	Gender
Station	female
city-of-london	0.1%
nottinghamshire	-4.5%
durham	4.1%

Table 4: Discrimination metric of the model for gender

	# Stations with discrimination	# Stations with positive discrimination	# Stations with negative discrimination	Median of total discrimination	Spread of total discrimination
Validation dataset	16	0	16	-3.4%	14.2%
Production dataset	0	0	0	0.1%	8.7%

Population Analysis

Several differences were found between the dataset used to train the model and the one received during production. For production **should_search** requests were only received for 4 stations : “City of London”, “Nottinghamshire”, “Durham” and “Cambridgeshire” . The station with most search requests in production was “Nottinghamshire”.

The client sent us the true outcome of a search for 51% of the **should_search** search requests. If all the searches for which the outcome was known were approved, the search sucess_rate would be 27.9%. For the training dataset it was 20.1%. Therefore there is a ~ 8% shift in the search success rate.

There are several imbalances in the set of **should_search** requests. Females represent a minority (9%) of the population and most of the searches are targeted at finding controlled drugs. These imbalances were already found in the training dataset. The proportion of requests for minorities belonging to ‘asian’ and ‘black’ minorities has increased w.r.t to the training

dataset. On the other hand, the proportion of the white population decreased from 79% to 65%, stations. In the production dataset there is also a higher proportion of requests for searches in age ranges “25-34” and “over 34”.

Other two features present different proportions in the production dataset : 28.4% of the searches were requested as part of a policing operations (5.3% in the training data) and almost 57% of the requests do not include the latitude and longitude of the the search (they were 25% in the training data).

To summarize, there are differences in the features distributions between the populations used to train the model and the population on which the model was deployed. Although for gender and age range the difference is not significant, it is relevant for ethnicity, whether the search is part of policing operations and the coordinates of the search. It must be stressed that this comparison was made between a training set composed of 41 stations, and a production dataset containing only data for 4 stations. It would be more fair to compare the production data with the subset of training data corresponding to those 4 stations. The datasets also have different numbers of samples, which is another source of statistical bias.

On 9th February 2021 heroku was out of service for a few hours. This was registered in the incident https://status.heroku.com/incidents/2173. The impact on the distribution of features is unknown. This topic is addressed in section “Model deployment”.

Figure 3: Number of search requests per station during production

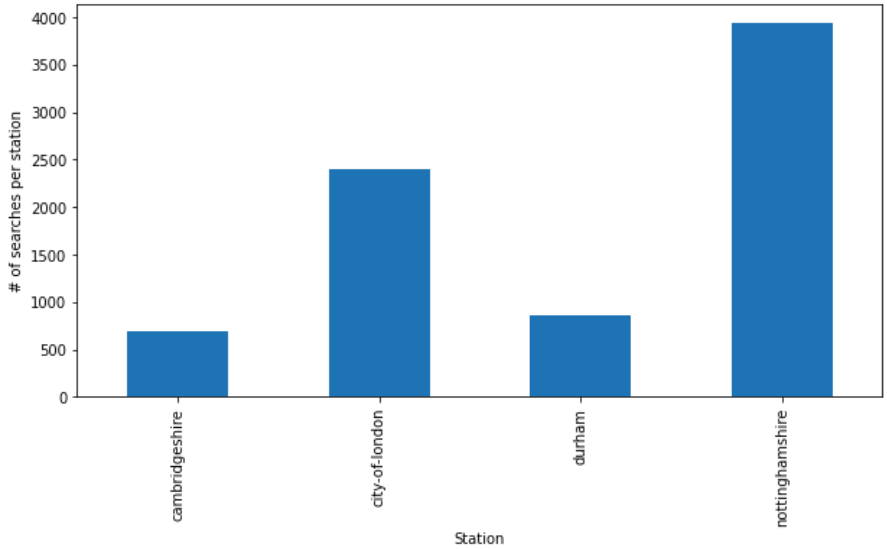


Figure 4 : Search success rate per gender

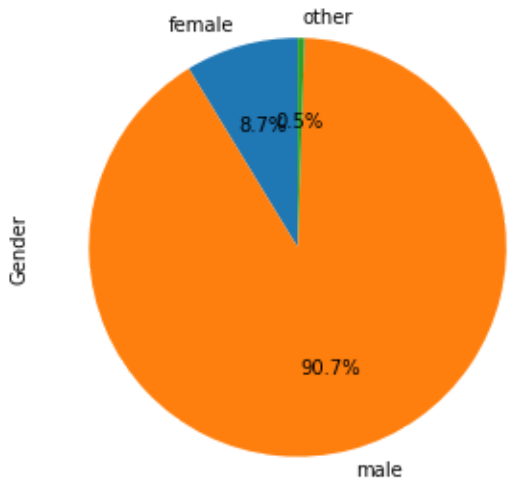


Figure 5 : Search success rate per age range

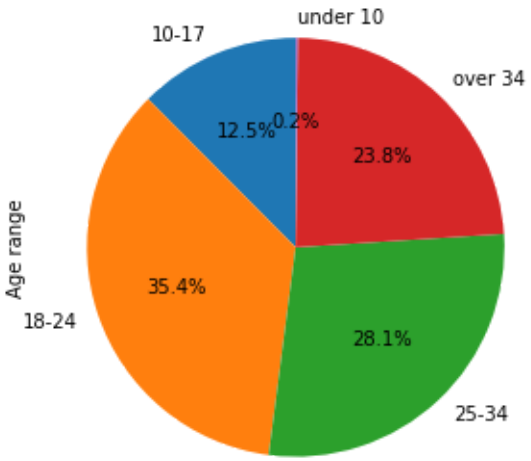


Figure 6 : Search success per ethnicity

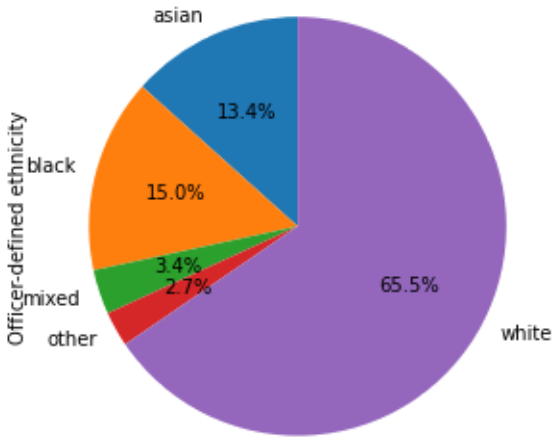
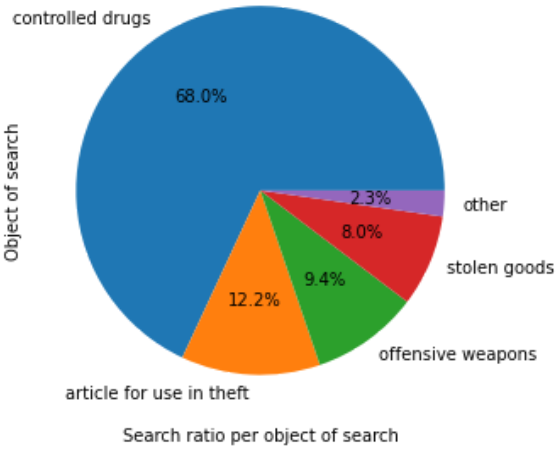


Figure 7 : Search success rate per object of search



Next steps

The main requirement that this project did not fullfil was the absence of the discrimination in the decisions of the model. This problem is a subject of active research and is commonly designated as “Discrimination-aware classification”. A good overview of the problem is presented in papers by Kamiran et al. [1,2,3]. This project attempted to develop a non discrimination model by removing the sensitive features from the list of features used to train the model and make decisions. This approach was unsuccessful. As discussed in [2,3] this approach does not take into account that the other features of the model might not be not independent from the sensitive feature. From the analysis present in the previous report, the disparity in search success rate between white and non-white groups depends on the station and the object being searched. Supposing that some minorities would commit more offences related to robbery than to controlled drugs or that they reside mostly in areas of some police stations for which searches are more frequent, then there will be discrimination. This effect is known in the literature as redlining. An historical example was the practice of denying inhabitants of certain racially determined areas from services such as loans. The model did not use the station directly to make classification, but used location coordinates and object of search. Removing these features would not be a good solution, because the predictions will be less precise, since they carry information about the outcome of the search.

Strategies must be devised to reverse the discrimination alrealy present in the dataset used to train the model. The first strategy would be to partition the data into subgroups with a fixed station, object of search and ethnicity. Then for each subgroup train a different classifier. The acceptance threshold of each classifier should yield a precision and a recall that is constant between subgroups. A disadvantage of this strategy is that data available to train each classifier would be reduced and that each station would have a different threshold for approving searches.

A second strategy would be to preprocess the training dataset in order to correct for the imbalance in the distribution of the features over the sensitive feature and the outcome of search. This can be done by massaging the data, i.e. change the outcome of the search for some minorities which are discriminated against white population. For example, start by partitioning the data in subgroups with a fixed station, object of search and ethnicity. If the search success rate is higher for this partition compared to the average precision then change the outcome of a subset of samples from true to false. If the success rate is lower do the opposite. It is expected that the search success rate will be more balanced between subgroups. An alternative to data massaging is to apply preferential resampling : remove samples with outcome true from partitions having higher search success rates and duplicate samples with outcome true for partitions having lower search success rates. A python toolbox to be considered is ‘imbalanced-learn’.

(https://machinelearningmastery.com/imbalanced-classification-with-python/)

A third strategy is to adapt the classifier itself: either include in the cost function the metric for discrimination and then minimize it or create a tree with number of leaves and branching criteria subject to non-discrimination constraints.

The next step would be to implement the first and second strategy to reverse discrimination and compare their performance with the current model. The third strategy is more complex and would require more effort. If these strategies are successful, the next steps would be to find a classifier with an optimal score for precision and recall and optimize its hyperparameters (e.g. maximum number of leaves for a decision tree) using cross validation.

Another improvement would be to include in the metric for discrimination a measure of statistical uncertainty. Currently the metrics for discrimination computes the difference in precision between groups wherever the number of samples is higher than 30. It should be improved to take into account a measure of the statistical uncertainty of the precision and give higher importance to subgroups where more data is available.[2] Furthermore before comparing the performance of the model between production and training the data sets should be restricted to the stations and objects of search which are present in both sets.

1) F. Kamiran, Discrimination-aware classification, Technische Universiteit Eindhoven (2011). https://doi.org/10.6100/IR717576
2) Kamiran, F., Calders, T. Data preprocessing techniques for classification without discrimination. *Knowl Inf Syst* **33,** 1–33 (2012). https://doi.org/10.1007/s10115-011-0463-8
3) Kamiran, F., Žliobaitė, I. & Calders, T. Quantifying explainable discrimination and removing illegal discrimination in automated decision making. *Knowl Inf Syst* **35,** 613–644 (2013). https://doi.org/10.1007/s10115-012-0584-8

Deployment issues

Re-deployment

The model was not re-deployed because for the subset of requests which were monitored it responded successfully to the requests with a latency lower than 50ms. The actual precision and recall corresponded to expected values. Since the problem related to the discrimination of ethnic groups by the classifier was not ready in time for re-depoylemnt, there was no justification to update the model.

Unexpected problems

We assumed that the logs of heroku router would be stored and be available for offline analysis for some weeks. This was incorrect, since in the free tier the logs are only kept for one week. Therefore the monitoring of the response codes and latency relied on random inspection of the results of the console and missed the last days of production. As a consequence it is not possible to ensure that all **should_search** requests were answered with a valid response and all **search_results** requests were stored in the application database. We received the true outcome of the searches for only half of the search authorization requests. It is unknown if this was the intent of the client or if it was a problem of the API.

On 9th February 2021 heroku was out of service for a few hours.This was registered in the incident https://status.heroku.com/incidents/2173. When service was resumed the API restarted to receive requests. The impact on the distribution of features is unknown. It could happen that during that period a significant fraction of requests for a given station or corresponding to a given etchnic group were missed, thereby adding more bias to the dataset.

What would you do differently next time

Ensure that the logs of application are stored in a persistent location to enable offline analysis of respondes codes and latencies. Increase the resilience of the application to problems in the platform by using a cloud service with high availability and a satisfactory Service Level Agreement. This would imply that the application is deployed into different servers and possibly in different cloud regions. This will also require that the cloud service provides routers which are able to detect problems in the servers and direct the external requests to the ones which are available. The database of the model must be also replicated.

We should have asked the client to provide statistics on the number of requests sent daily to each API endpoint and the fraction of requests answered successfully. This will allow us to check the fraction of requests which were not received or answered by the API.

Furthermore we should have asked for clarifications on big disparity between training and production data, e.g. why there were only requests for a subset of stations.