LISBON DATA SCIENCE ACADEMY

Reducing Bias on Stop-and-search Operations

Model Performance and Future Improvements Report

Prepared for:
United Kingdom Department of Police

Prepared by:
João Sá
Data Science Specialist
DS Team @ Awkward Problem Solutions

11th of June 2023

Table Of Contents

1. Summary

The United Kingdom Department of Police (UKDP) has been dealing with accusations of discrimination in its stop-and-search operations. In the first report, we established that there was evidence of discrimination in some departments. Specifically, we found that Black-Female, Other-Male, and White-Female individuals are oversearched for a large number of stations. We also found that the search objective with the highest success difference between race-gender tuples was Controlled drugs, due to a low success rate for Black-Other individuals that are clearly oversearched for this objective. Finally, we found a reason for concern that some stations overly requested the removal of more than outer clothing for white women over 25.

As part of this project, we were tasked with developing a system that approves stop-and-search operations and levels the success across race-gender tuples. The system should have at least 80% recall so that the ability to detect infractions wasn't severely compromised. We also expected that precision should not vary by more than 5 percentage points across all protected classes in each of the stations so that we could level the discovery success for each race-gender tuple and reduce discrimination. Finally, stop-and-search should only be suggested when there is more than a 10% of the likelihood of success.

To build this system we developed a Machine Learning model based on an implementation of the Gaussian Naive Bayes algorithm. To suggest whether a stop-and-search should take place the model used the following features - Date, Type, Object of search, Part of a policing operation, Latitude, and Longitude. We established a preprocessing pipeline that applied separate transformations for date columns, categorical columns, boolean columns, and numerical columns and finally trained the model. The transformations considered included filling in missing values, feature engineering for the date column, encoding categorical data, and, scaling the data. Since the provided data was unbalanced we also used resampled the training data by randomly removing records from the majority class (search-and-stop unsuccessful) until a balance between target classes was reached. With this model, we were able to obtain a recall of 0.821 on the test dataset as well as an average difference between the success rate for each race-gender tuple across stations of 0.113. We considered that the system would suggest stop-and-search when there was more than 50% of likelihood of success. As such the only requirement we weren't able to deliver was keeping the average difference below 0.05, although we were able to increase the number of stations where such a difference existed from 2 to 8.

The final model was deployed as a resting Application Programming Interface (API) on railway.app under the flask framework with two modules - /should_search/, which handles requests for prediction, and /search_result/ that handles updates to a record with the true outcome result. The requests are stored on a PostgreSQL database.

2. Results Analysis

2.1 Model Performance

During the first round of requests we successfully received and provided predictions for 4000 requests. We expected a recall of 0.821, but we observed a recall of 0.73 during the first round. This was due to 236 missed cases (27% of all search-and-stop successes). In this respect, the model did slightly worse than what we expected - missing 17.91% of all possible cases. This might be due to population differences compared to the training data, as we will see in section 2.3, or due to a bad choice of decision threshold, which was kept at 0.5.

Out of the 4000 requests, the model suggested a search for 2579, which means a reduction of 35.5%. During deployment, we expected a reduction of 31.3%, which means the model behaved close to what was expected.

According to the true outcome of the operations, the success rate of search operations was 22%, whereas when considering searches suggested by the model the success rate would have been 25%. During development, we established that the model showed a success rate of 22%. This means that the model not only did better than the agents at identifying infractions without oversearching, but it even did better than expected during development. This, of course, came at the expense of missed cases, as evidenced by a poorer recall value.

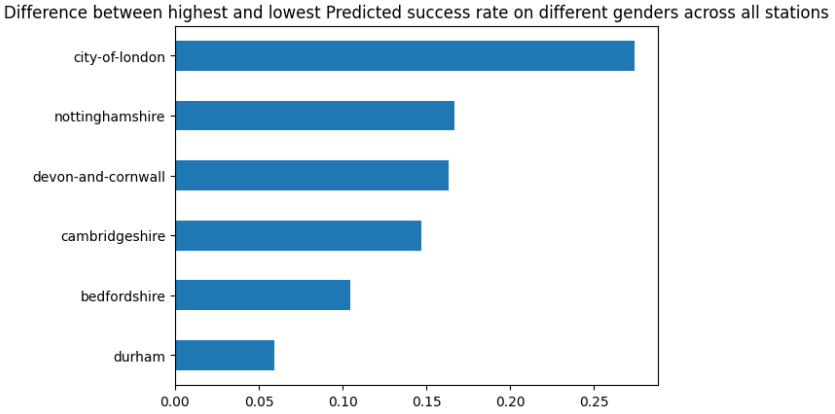

Image 2.1 - Predicted Succes Rate difference for different genders

In regards to discrimination, we expected an average success rate difference between race-gender pairs for all stations of 0.09. During the first round of requests, we actually got an average difference of 0.15, which was worse than expected. Image 2.1 above shows the differences for each station for the predicted outcomes, and Table 1 of section 5 provides the full list of differences for each race-gender tuple and station. We can see that the station where the model showed the highest difference was city-of-London, due to a very high success rate for Black Males and a low success on White Females. It's interesting to note that when comparing with the True Outcomes, city-of-London was actually the department with the lowest difference, as can be seen in Image 1 of section 5.

2.2 Success on requirements

The model requirements that we established in report #1 were:

- the system should approve stop-and-search whit at least 80% recall. This means that the model will be able to capture most offenses.
- the model's precision should not vary by more than 5 percentage points across all protected classes, for each of the stations. This means that case discovery should be level across departments.
- Finally, the client has requested that searches are performed only when there is more than a 10% of likelihood of success. This means that we should keep the decision threshold above 0.1.

The developed model was able to clear the first requirement successfully, having shown a recall of 0.821 in the test set. Unfortunately, this was not true when dealing with the new data, where the recall dropped to 0.73 and missed the requirement. As mentioned in section 2.1, this might have been due to data differences between the new data and the training data used to develop the model. This analysis will be expanded upon in section 2.3.

In regard to the second requirement, we were not able to achieve a model that provided such functionality. In fact, we saw that the best we could achieve was an average precision difference of 0.113. During the first round of requests, the average precision difference between race-gender tuples across stations was actually 0.173. As can be seen in image 2.2, we weren't even able to clear this for a single station.

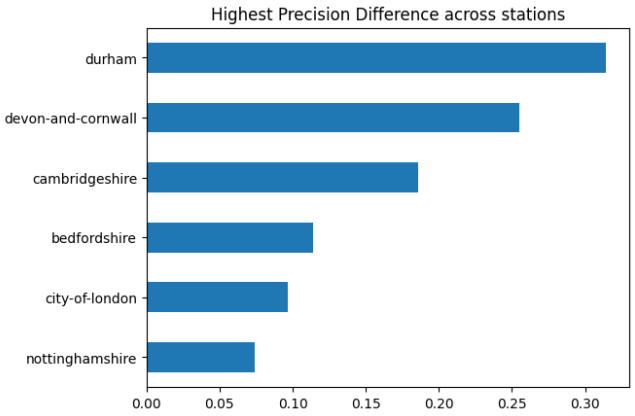

Image 2.2 - Precision Difference across stations

As described in section 2.1, this translated to higher success rates between protected classes. For the predicted outcome, the station that did worse was city-of-London, whereas the station that did the best was Durham, followed by Devon-and-Cornwall. This was significantly different from the True Outcome, seen in image 1 of section 5, where the best station was actually city-of-London and Durham was the second worst station, following Devon-and-Cornwall. This indicates that there might be some issue with the way the model is suggesting searches, specifically some unwarranted bias against specific classes.

In fact, when analyzing the success rate of the predicted outcome for each race-gender tuple, we see that White Females and Males have the lowest success rate on more stations (image 5 of section 5), whereas for the True Outcome, it was actually Black Males and White Females (image 3 of section 5). This might indicate that the model is

inadvertently “targetting” White Males more than agents. Looking at the tuples with the highest success rate for the predicted outcome we can see White Females and Asian Males for more stations (image 4 of section 5) whereas for the True Outcome, it was actually Asian Males and Black Males (image 2 of section 5). This might indicate that the model was unsuccessful in predicting the outcomes for Black-Males, leading to a poorer success rate when compared with the True Outcomes.

Image 2.3 shows that the tuple with the lowest precision was White-Females, whit a significant drop when compared to most other tuples. This is proof that the model seems to be underperforming for this class and over-suggesting searches, which led to poorer overall results. If we look back to Report #1, we might recall that White-Females were the class with the highest success rate for 14 stations. The following tuple was Other-Male which had the highest success in only 6 stations. This suggests that White-Females were clearly under-searched in the original data used. It is possible that the model picked this bias during training and that is the reason it is now underperforming when White-Females aren’t as “protected” in the new data.

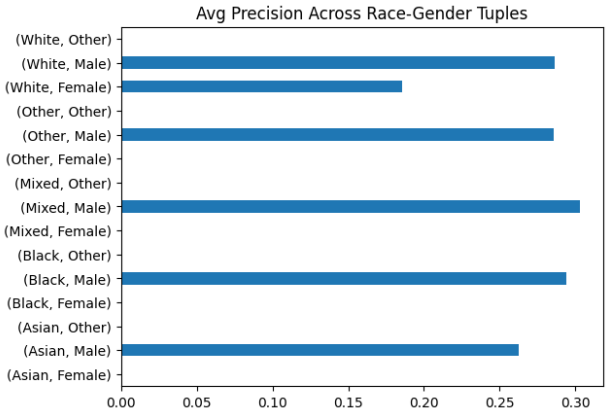

Image 2.3 - Average precision across Race-Gender Tuples

Finally, in regard to the last requirement, we kept the decision threshold at 50%. Although this did clear the client’s request, as was exposed in report #1, we did not perform decision threshold analysis and did not try different values for the decision threshold. As mentioned in section 2.1, this might have been the cause of missing a large number of cases, and as such we did try to optimize this value during redeployment, as will be shown in section 3.1.

2.3 Population Analysis

As has been mentioned in previous sections we did find significant differences between the original data and the new data received during the first round of requests. The first evidence of this can be seen in image 2.4, where it’s clear that we have a higher rate of success in the new data when compared to the original data.

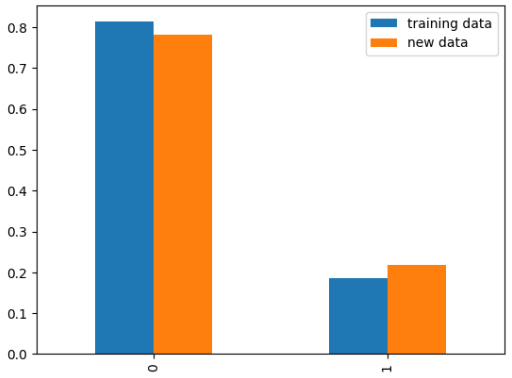

Image 2.4 - Distribution of target class between Training and New data

Furthermore, it was noticed that we only received requests for a subset of the stations present in the original data, as can be seen in image 6 of section 5. In the original data we used to have records for much of the UK, whereas now the observations are much more dispersed geographically.

Another topic of interest is that, as can be seen in image 8 of section 5, the most common tuple in the new data is now White-Males-Over 34, unlike in the original data where it was actually White-Males in the 18 to 24 range. Something similar happened for White-Females, where now the 18 to 24 range has the highest representation, whereas in the original data, it was actually the 18-24 range. Finally, it should be noted that Black males have a much higher ratio of the records in the new data than they had in the original data, except for the 10-17 range, where the ratio is lower in the new data. All of this might have a significant impact on the model when trying to deal with discrimination.

We also noticed that now there were no records of the type “Vehicle and Person search” (image 7 of section 5), but this should not have much impact on results, seen as the number of records for this type in the original data was almost negligible. As in the original data, Person search is still the type for the majority of records.

When comparing the success rate difference between race-gender tuples across stations, we noticed that in the new data, the average value in the original data was 0.1, and this is still the case for the new data (image 1 of section 5). Despite that, the variance of this difference is much lower now, with the highest value being 0.175 and the lowest 0.066, whereas in the original data, it varied from 0.03 to 0.23.

Feature	p-value
Type	0,000
Date	0,000
Part of a policing operation	0,000
Latitude	0,000
Longitude	0,000
Gender	1,000
Age range	0,000
Officer-defined ethnicity	0,000
Legislation	0,000
Object of search	0,017
station	0,000

Finally, we performed the Kmogorov-Smirnov test to check whether the new data differed from the original significantly. The table to the right provides the p-values obtained for each feature. If we establish that a p-value below 0.005 is the threshold to consider data drift for a specific feature, we can clearly see that the only features that don’t show data drift are Gender and Object of search. As such, it’s safe to say that redeployment is necessary to guarantee the correct performance of the model before the second round of requests.

3. Deployment Issues

3.1 Re-deployment

As established in section 2.3 we found evidence that there was significant data drift between the training data and the new data. Also, we found some issues after deployment that we wanted to address during this redeployment window. The first had to do with analyzing different values for the decision threshold. Finally, we did not remove all station that was missing Outcome linked to the object of search for more than 90% of records.

Just as we did for the first deployment, here we established an iterative process where we addressed each of the issues above by comparing different hypotheses and choosing the resulting model that showed the highest recall and the lowest average precision difference across stations for the test set. The complete list of performance metrics for models trained at each step can be found in Table 2 of section 5. We will be referring back to this table in the following paragraphs.

Starting with the model established during the first deployment, we tried to address data drift for the first step of redeployment. For this, we tested three hypotheses - include all data for training (2020, 2021, and 2022), include the most recent data (2021 and 2022), and include just the new data (2022). At this point, we also removed all stations missing Outcome linked to the object of search for more than 90% of records from the original dataset - Humberside, Lancashire, Metropolitan, West-midlands, and Leicestershire. Furthermore, we decided to remove the year feature from the data. This meant that from the date feature, we only extracted quarters, hours, and the day of the week on the feature engineering step. As can be seen in the table, including just data from 2021 to 2022 proved to deliver the highest recall - 0.836 in the test set compared to 0.821 in the previous model. The average precision difference also dropped from 0.113 to 0.1.

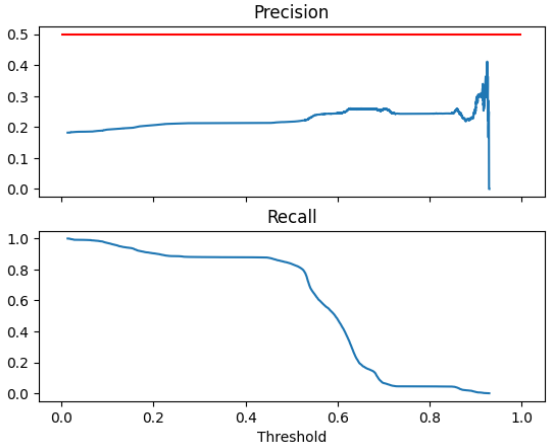

Image 3.1 - Precisio-Recall curve

For the next step, we tried to optimize the decision threshold. Has can be seen in image 3.1, most threshold values have little impact on precision. In fact, we would only see some increase in precision for thresholds that would reduce recall to almost 0. We also noticed that recall would only show a significant drop around 0.55, but we could increase it slightly if we chose 0.4 instead. As such, the two hypotheses tested at this step were a threshold of 0.4 and a threshold of 0.6. Looking back to table 2, we decided to choose the

latter since it once again improved the recall to 0.879. Interestingly, even though the global precision dropped from 0.294 to 0.213, the average precision difference between classes reduced from 0.1 to 0.097.

Finally, we tested what the results would be if we removed stations that had a low success rate. This meant that we would be dropping Lincolnshire and Dyfed-Powys, since they had very low success rates (0 and 0.056 respectively). South-yorkshire also had a success rate of 0.06, but we chose to keep it since it still had a significant number of successes (638). This however proved to have little to negative impact in the results, with recall on the test set dropping to 0.876 and average precision difference increasing to 0.1

As such we decided to redeploy a new model that improved on the previous one in the following ways:

- Considered only data from 2021 to 2022, and transformed the date feature into 3 features - quarters, hours and the day of the week.
- Removed Humberside, Lancashire, Metropolitan, West-midlands, and Leicestershire from the original training data
- Considered a decision threshold of 0.4
  All other aspects of the model were unchanged.

Unfortunately, we were still unable to develop a model that leveled discovery across all protected classes below 0.05, but we did improve model bias slightly with this new development. Also, we expect this new model to miss less cases, since we were able to improve recall by 0.05.

3.2 Unexpected problems

We did not find any major problems during deployment. Thankfully the API did not crash at any point in the first round of requests. As mentioned in section 2.1 we received 4000 new records and were able to store them in the database and return predictions for all of them successfully. This is proof that the validation rules developed for the API were well-designed.

At one point we did receive some weird requests that can be seen in image 3.2. This was a known issue that was mentioned in Report #1. Since the API was publicly accessible we were prone to receive such requests. Despite that, the API was unaffected by these spam requests and was able to still produce replies for good requests that came at a similar time.

Image 3.2 - Unknown API requests

One final thing to mention here was that we did run out of credits some days after the end of the first round of requests. We could have removed the deployment and saved some credits, but unfortunately did not do so. As such we had to change to the paid tier to be able to deal with the second round of requests.

3.3 Learnings and Future Improvements

Regarding the API we found that it behaved quite well during the first round of requests. As such we see no reason to make significant changes to the current implementation. As explained in the previous section, the API was able to deal successfully with spam requests and with real requests. We would still suggest deploying the API to a better platform for production. Potentially one with better performance and not publicly accessible - potentially a paid service.

Concerning the Machine Learning Model that was developed, we found that it was quite hard to completely remove existing bias from the dataset.

We were able to deliver a model with a recall higher than 0.80 and a decision threshold higher than 0.1 as requested by the client, but we were not able to level the discovery rate across race-gender tuples below 0.05. We believe that this is due to the bias that is present in the dataset that we were not able to remove completely. Ultimately we decided on keeping as much data as possible to try and capture any underlying subtleties. This might not have been a good approach since the model seemed to pick some bias from the data and was unable to completely level the discovery rate.

4. Learnings and Future Improvements

As mentioned in section 3.3, the hardest task in this project was to level discovery for each race-gender tuple across all stations. We can think of two reasons why this was so:

- The provided data was biased - this was proven in Report #1. Only a very small number of stations showed a difference in discovery between protected classes below the 0.05 threshold that was established. This was also true for the new data, where no station showed a difference below the threshold of 0.05. Training a model on biased data will lead to a biased model most of the time.
- We were unable to successfully remove said bias from the training data

Regarding the first point, we have already suggested in report #1 some further analysis that could be conducted by the UKDP to understand why there is bias and how best to understand it. They should check whether the bias is introduced not only by some stations, but also by specific agents, and if so, conduct debriefing sessions to understand why this is. They should also analyze whether bias is higher due to agent over-zealousness in communities that could be deemed at risk by analyzing the success rate together with the area's average income rate.

With respect to removing bias from the training data, we have some suggestions that weren't tried during development but should be explored in the future:

- Instead of using random undersampling, we could try to remove specific records so that we get a more homogenous dataset that shows an average success rate for each station and race-gender tuple below 0.05.
- We established that some stations showed higher bias against some protected classes. We suggest removing these and checking whether the average precision difference improved.
- We saw in report #1 that some search objectives showed higher bias against specific classes (namely Controlled drugs and Stolen goods). We suggest removing these and checking for improvements in the results.

Finally, we are also somewhat concerned that with the new model, we might miss important cases. A recall of 0.80 seems reasonable from a technical standpoint, simply because of the degree of unbalance of the target feature. However, missing 20% of possible crimes is not acceptable in the real world. As such, we believe that model suggestions should not discourage agents from acting on their instincts. Our suggestion is that the model should actually be used as a sanity check before initiating a stop-and-search so that the agent fully acknowledges any unconscious bias against a specific race-gender tuple.

In fact, an interesting reframing of the problem would be to provide not only a go/no-go response but also the past success rate that this agent or station had for the race-gender in question. As an example, if the officer is pondering searching a White-Female and the model suggests not conducting a stop-and-search and displays that the success rate in the past for this class has been below 5%, then the agent can ponder whether their instinct to search might be due to some unconscious bias against this class before initiating the operation.

In conclusion, for this project, we did a comprehensive analysis of discrimination in the UKDP. We found that there are some stations that conduct stop-and-search on some classes more than others, with no significant improvement in crime discovery. We have created a resting API to level the discovery rate across protected classes (race-gender

tuples). Unfortunately, we were not able to completely remove bias from the model underlying the API. Nevertheless, we believe that this model and API are a good first step toward reducing discrimination in the UKDP. For the next steps, we suggest an improvement to the API so that it not only shows a go/no-go prediction but also presents the success rate that the agent has for the targeted race-gender class. We believe that this will act as a deterrent against any unconscious bias that the agents might have. We also suggest conducting debriefing sessions with the agents that show the highest success rate difference between race-gender tuples to better understand what could be causing this difference and if there are any learnings that can be taken from these agents to improve department practices and reduce discrimination in the UKDP.

5. Annexes

station	Asian	Asian	Asian	White	White	White	Other	Other	Other	Black	Black	Black	Mixed	Mixed	Mixed	diff	worst tuple	best tuple
	Male	Female	Other	Male	Female	Other	Male	Female	Other	Male	Female	Other	Male	Female	Other			
cambridgeshire	0,67			0,57	0,55					0,70						0,15	('Black', 'Male')	('White', 'Female')
city-of-london	0,68			0,62			0,40			0,55						0,27	('Asian', 'Male')	('Other', 'Male')
devon-and-cornwall				0,73	0,76					0,60						0,16	('White', 'Female')	('Black', 'Male')
durham				0,21	0,27											0,06	('White', 'Female')	('White', 'Male')
nottinghamshire	0,77			0,71	0,60					0,71			0,66			0,17	('Asian', 'Male')	('White', 'Female')
bedfordshire	0,79			0,71	0,81					0,81						0,10	('White', 'Female')	('White', 'Male')

Table 1 - Success Rate of predictions for each Race-Gender Tuple across all stations of the First Round of Deployment, with the largest difference for each station and the “worst” and “best” tuple

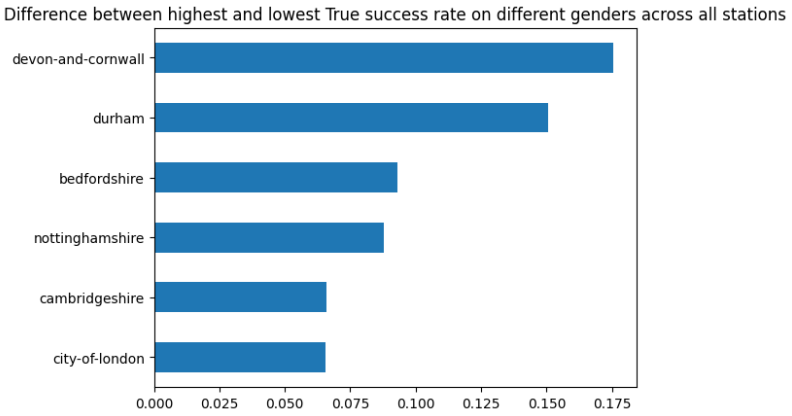

Image 1 - Highest Success rate difference of True Outcomes between Race-gender tuples for each station

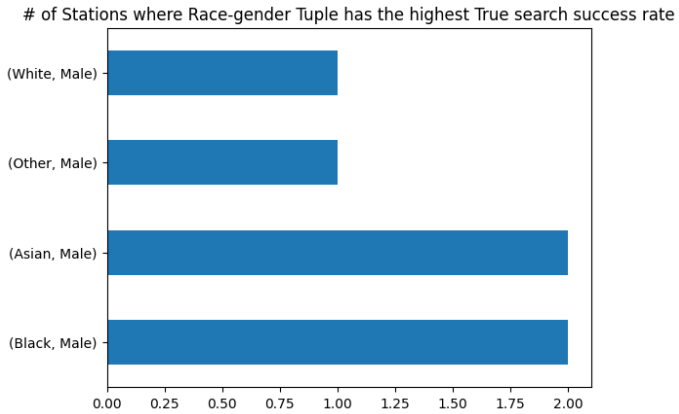

Image 2 - Number of stations where the race-gender tuple has the highest success rate for the True Outcome

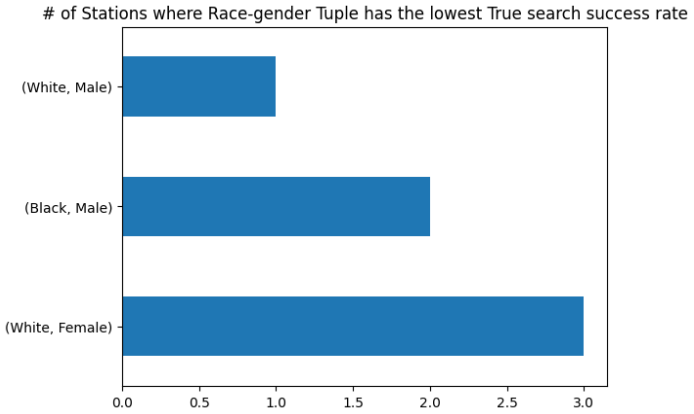

Image 3 - Number of stations where the race-gender tuple has the lowest success rate for the True Outcome

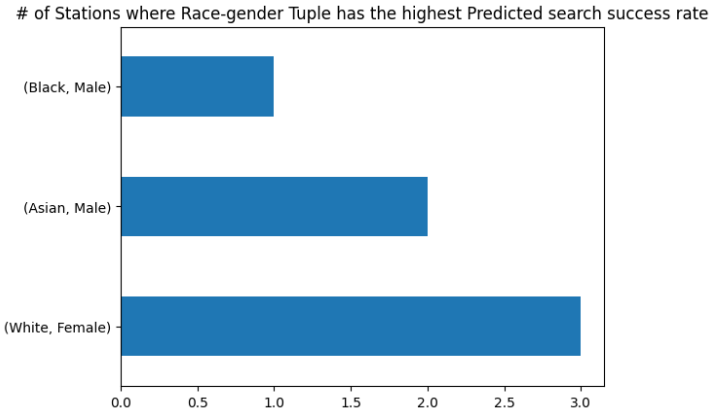

Image 4 - Number of stations where the race-gender tuple has the highest success rate for the Predicted Outcome

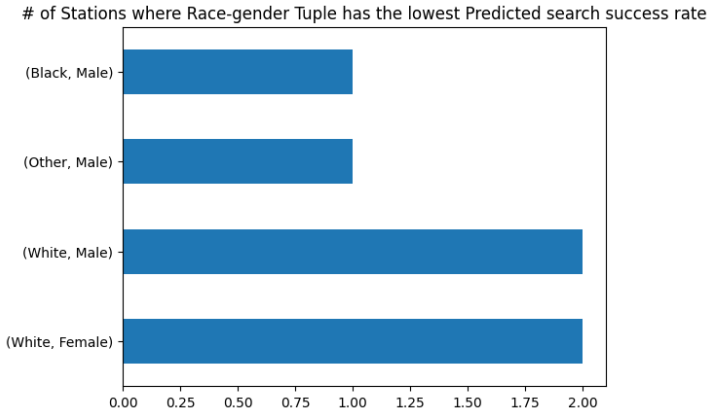

Image 5 - Number of stations where the race-gender tuple has the lowest success rate for the Predicted Outcome

Image 6 - latitude and longitude distribution of search-and-stop action for the new data, colored by station

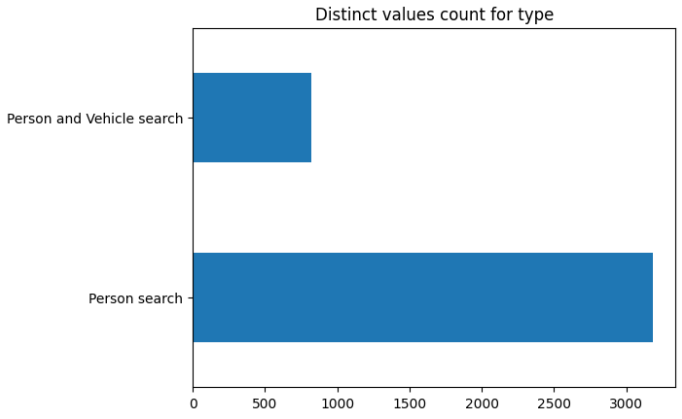

Image 7 - Number of Records for the Type Feature

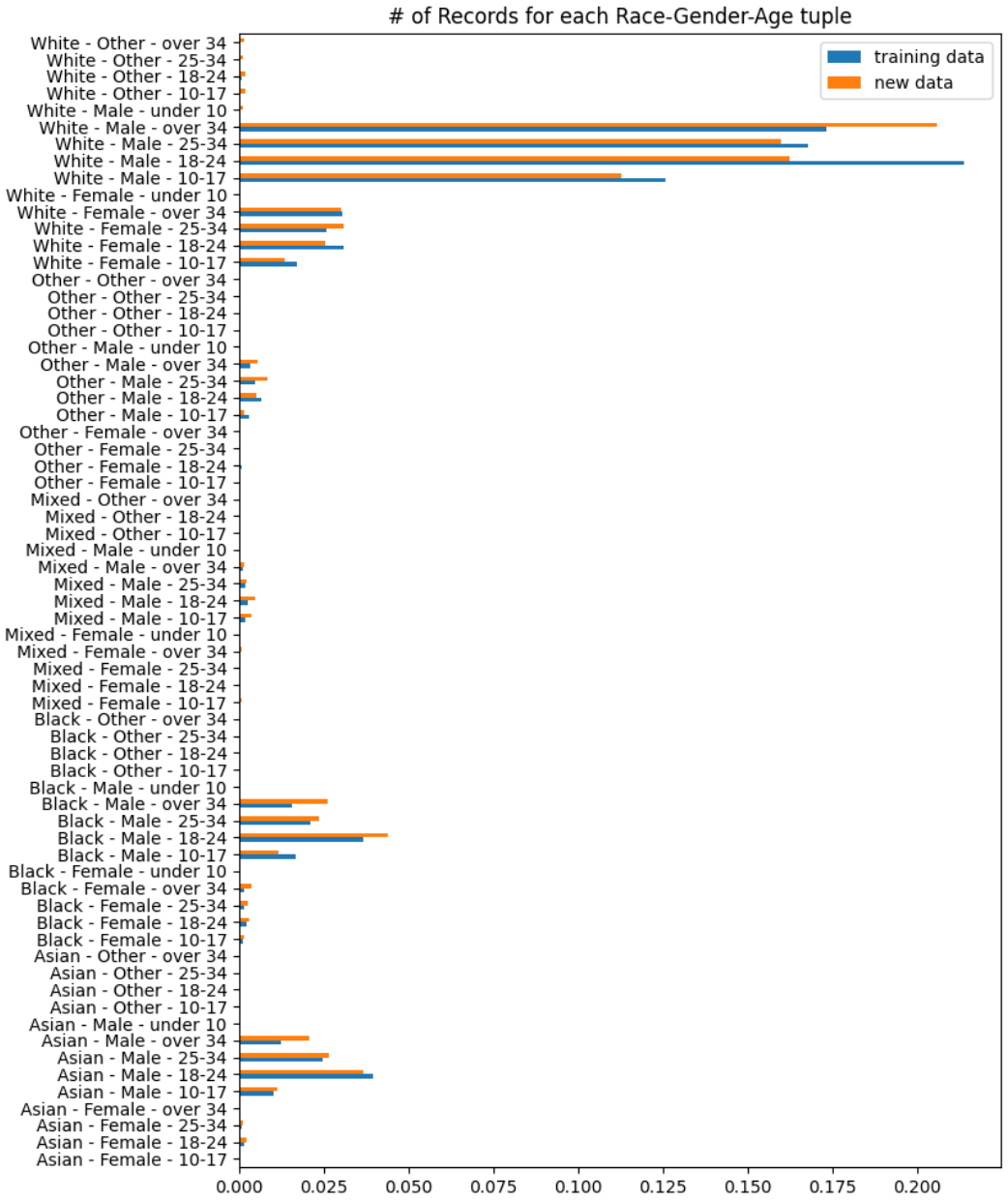

Image 8 - Ratio of records for each Race-Gender-Age tuple in the original data versus the new data

Step	Model	Train Set				Test Set				AUROC	Discrimination		
		Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score		Number of Good Departments	Number of Problematic Departments	Average Precision Difference
Deployment #1	Remove Sensitive Features	0,582	0,555	0,821	0,663	0,429	0,217	0,821	0,343	0,61	3	34	0,113
Deal with Data Drift	Include all data (2020, 2021, 2022)	0,58	0,554	0,819	0,661	0,432	0,225	0,827	0,354	0,61	2	34	0,114
	Include only new data (2022)	0,591	0,594	0,571	0,583	0,58	0,294	0,61	0,397	0,62	4	2	0,053
	Include only recent data (2021, 2022)	0,589	0,553	0,832	0,664	0,422	0,217	0,836	0,345	0,62	8	27	0,1
	Include only recent data and just stations in new data	0,589	0,572	0,713	0,635	0,518	0,276	0,723	0,4	0,62	0	6	0,157
Change Decision Threshold	Reduce Threshold to 0.4	0,575	0,547	0,873	0,673	0,387	0,213	0,879	0,343	0,62	8	27	0,097
	Increase Threshold to 0.6	0,572	0,59	0,471	0,524	0,636	0,245	0,479	0,324	0,62	8	27	0,15
Remove Stations with Low Representation	Remove Stations with Success Rate < 0.1	0,576	0,548	0,871	0,673	0,396	0,221	0,876	0,353	0,62	5	28	0,1

Table 2 - Performance Metrics for Models trained during each step of the redeployment process