Client Requirements

Summary

Each police department has a stop and search policy in place that is supposed to ensure that officers only stop people and/or cars when there is probable cause. So far, there have been several accusations in the press that the police are behaving in a biased manner towards certain minority groups, who are being stopped and searched more often than others.

Another sensitive area is related to the question of asking suspects to remove articles of clothing for search, in which the accusations are that women of certain age groups and ethnicities are being the most targeted.

Awkward Problem Solutions^{TM} has looked into this with great detail, developing a model to help decide when a person should be stopped without being biased towards their age, gender or ethnicity. Furthermore, a thorough analysis has been made to verify these allegations and conclude if, in fact, these accusations are valid or not.

This report firstly describes the analysis made on the historical data provided by the United Kingdom Department of Police about past searches. This data has all the information available about each intervention, the gender, ethnicity and age of the person as well as the objective of the search. Secondly, this report also describes the employed solution and how to use the API.

Requirements Clarifications

An analysis of the historical data should be done in order to find out from all the people that were stopped, if, in relative terms, people from sensitive groups were being more targeted when compared to others.

One caveat to take into consideration when determining discrimination is that there are classes that are stopped more often simply because that class exists in more volume in the population when compared to others. To tackle this, we have defined discrimation as **the rate of people stopped by a station, divided by all the people that were stopped from that class in all stations, which has to be higher than a specific threshold when compared to other stations.** The chosen threshold was **15%** when validating stopping criteria and **10%** when validating clothes removal, being the latter stricter since it can be considered as more offending. This is explained in more detail later in the report.

Regarding the model, it was requested that the success rate of the searches should not vary significantly (**5%**) between population sub-groups which were defined as (station, ethnicity, gender) and the average per station should not be larger than **10%**. Success rate was translated as the precision of the model[1], maximizing the true positives.

[1] See Annexes: Business questions technical support for more detail

Dataset Analysis

General Analysis

The provided dataset has a span of 2 years, between December 2017 and December 2019 with 660611 rows with 15 different columns, being the most important summarized below:

Station

The station that conducted the stop and search. In the dataset there are 42 different stations and their quantity of stops is quite different

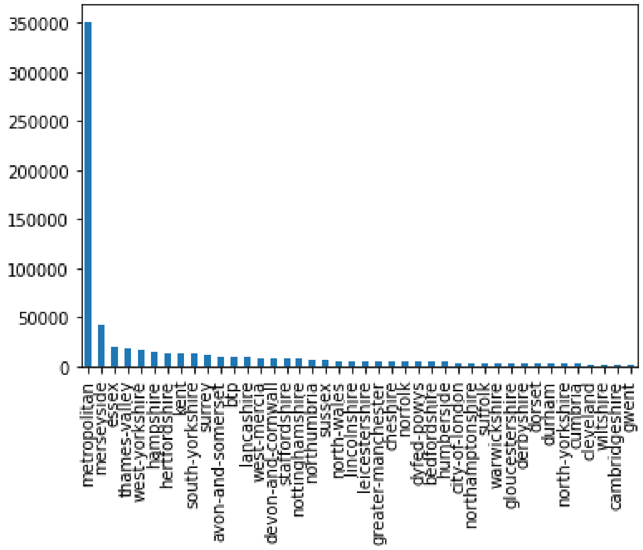


Figure 1. Number of stops per station

Outcome, Object of search and Outcome linked to object of search

The columns that describe the target of the problem. Of the people that were stopped and searched, what was the outcome. **A search is considered successful if the "*Outcome*" is both positive and is related to the search**. The majority are negative cases, turning this problem into an imbalanced classification problem[1].

Having this in consideration, **the metropolitan station never has a successful search since the "*Outcome linked to object of search*" column is not populated.** Due to this, the station was not used to train the model.

[1] See Annexes: Business questions technical support for more detail

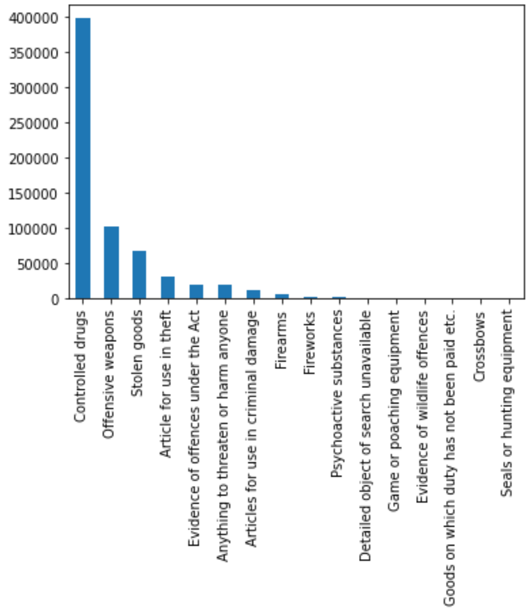


Figure 2. Count of Object of Search

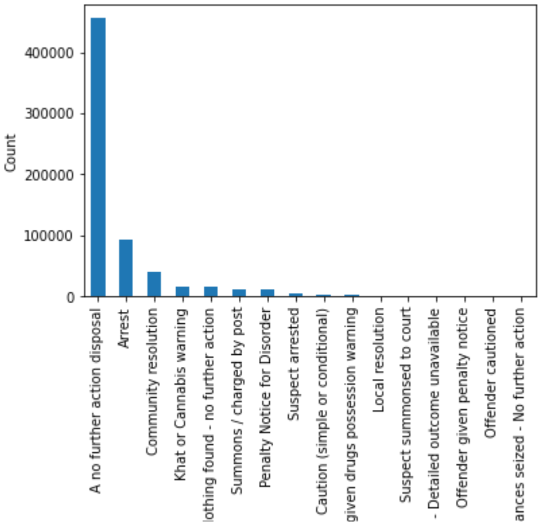
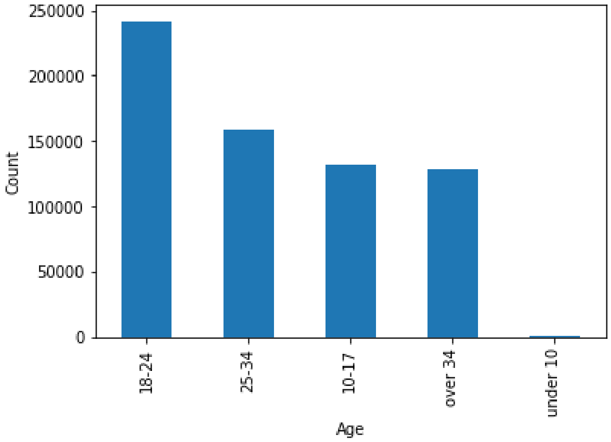


Figure 3. Count of Outcome

Age Range and Gender

The columns that identify the sensitive classes which are prone to discrimination. There are 5 different age ranges as visible in the plot below, being *“under 10”* one of them. However the latter almost has no cases.

The Gender column is defined by Male, Female and Other, being very unbalanced for the Male gender, meaning that males are in general more often stopped than women.



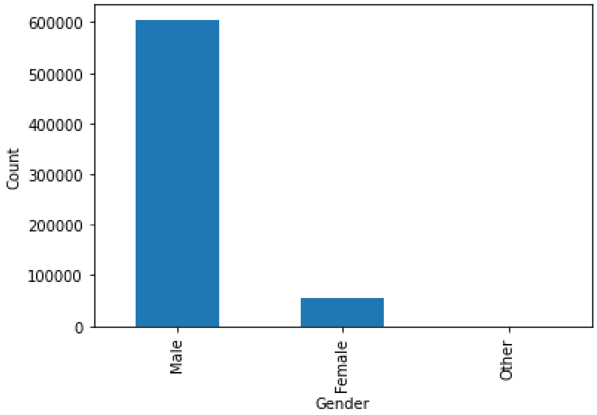


Figure 4. Count of Age range

Figure 5. Count of Gender

[1] See Annexes: Business questions technical support for more detail

Ethnicity

There are two columns that may describe ethnicity, *“Self-defined ethnicity”*, which is defined by the person that is being stopped (it also has some missing entries). Moreover, it has many more options when compared to *“Officer-self defined ethnicity”*, which is selected by the officer conducting the search and can only have 5 different values. It can, however, introduce officer bias as it is open to the officer’s interpretation and choice.

Removal of more than just outer clothing

This column provides information about whether the stopped person was asked to remove more than just their outer clothing or not. It is the column that needs to be analysed in order to understand if the accusations are true. There are several values that are missing, 426549 rows to be exact (64,6% of the dataset), which we were told to fill in as False only when the search type is not “*vehicle search*” which should be left missing. The majority of cases, around 95%, are not asked to remove more than just their outer clothing.

Business Questions Analysis

In order to identify discrimination or bias against stopping people from certain backgrounds and characteristics, each station has to be analysed and compared between each other for every sensitive column: **gender**, **ethinicty** and **age**.

To find discrimination between classes an assumption was made that the quantity of the station itself may vary, as an example Males are much more targeted in general than Females. This might not mean discrimination, it might just mean that the class exists in more quantity therefore being stopped more frequently.

Having this in consideration, discrimination was understood as a high variation in stopping rate or/and a difference in rates higher than a threshold when compared between classes. So if somehow a station stops and over searches a class when compared to other stations, this might mean a discrimination from that station towards that class in particular as the population may be well distributed across stations. For the stopping rate, the chosen threshold was 15 percentage points.

In a similar way, when considering clothes removal, we want to analyse from all the people that were stopped, what is the rate that were asked to remove more than just their outer clothing. Of all the people that were asked to, how many of those are in each sensitive class. Ideally that rate should be stable within the station and/or similar between each station, indicating a fair and equal rate between stations. For this, the chosen threshold was 10 percentage points being more conservative as it might determine a more serious problem.

Finally, time was also taken into consideration. At a first instance an analysis was done monthly to the stopping rates for each class. However, monthly data may introduce several fluctuations and be too detailed. Due to this, the analysis was switched to quarterly as it is believed to

[1] See Annexes: Business questions technical support for more detail

introduce the amount of detail needed as well as taking training into consideration, which are supposedly often done quarterly.

The evolution of rates over time of some deemed important stations that stood out from this analysis were plotted, will be referenced in the conclusions below and can be seen in the Annexes[1].
When taking the category Gender in consideration, the protected class that is said to be targeted is **Female**.

Regarding the Age analysis, the value *under 10* was not used as it was not present in all stations and represented too few of the population for a conclusion to be trustworthy.

Before analysing Ethnicity in terms of stopping and removal of clothes, we first conducted an analysis to conclude if there were any differences between the Officer and self-defined ethnicities. There are 124000 rows that do not match, however if we remove the “Other” and “Mixed” values that may introduce unclear interpretations, then we only have 8500 mismatches, representing 1.3% of the dataset. Of all the mismatches, most people claim to be White whereas officers claim that they stopped mostly Black people. As this is present in only a very small subset of the dataset, this can be ignored and interpreted as a user input mistake, however we thought it important to include in the report.

For this category, Other and Mixed value rows were also not used as all are not present in every station and represent a very low percentage of the data as well as not being meaningful or providing much information about the person being stopped.

Conclusions and Recommendations

Regarding Gender discrimination, when only taking stops into account, the rates seems well distributed and equal rates when compared between stations, only **Merseyside** demonstrating a low rate of search on Females. However, when looking at the clothes removal analysis, we can conclude that there are some stations, namely **Cumbria**, **Warwickshire** and **Durham**, that show very high variations, ending on high rates for Females which should be followed closely.

When taking Age into consideration, it is possible to see that **Cleveland**, **North Yorkshire** and **Dorset**, show high stop rates for people over 34 years old, to the detriment of searching between the age ranges of 18-24. Furthermore, **City of London** and **Bedfordshire** stations seem to search more over the range of ages between 25-24 and 18-24. In the clothes removal analysis, **Derbyshire** and **Dorset** seem to have very fluctuating rates but end on an average term. On the other hand, **Cumbria**, shows a very high search rate in the later year of 2019 of people between 25 and 34 years old, which if joined with the Gender analysis we can conclude that most of these are women which can be seen as a red flag. Moreover, **Warwickshire**, presents a very high ending peak in the last quarter of 2019, with an incredible high rate search for people over 34 which we also know are Females with a high clothes removal rate.

[1] See Annexes: Business questions technical support for more detail

In the Ethnicity analysis, we can conclude that **Metropolitan** almost only stops Black and Asian people when compared with other stations, this should be followed up upon since it seems a very urgent and alarming difference. On the other hand, **Cumbria**, **Dyfed Powys** and **Cleveland** almost only stop White people. When taking clothes removal into consideration, there are stations that almost never ask White people to remove their clothes whereas for other ethnic groups, 1 in every 3 people, and sometimes even in every 2, are asked to remove their clothing. These stations are **Suffolk**, **Derbyshire**, **Hampshire**, **Northamptonshire** and **Durham**.

Modelling

Model expected outcomes overview

When a police officer is conducting a stop and search, our API should be sent all the available information of the person that can be collected. In turn, the API will provide a decision if either that person should be stopped or not. As previously explained, the model should not be biased towards certain protected classes and subgroups, namely (station, ethnicity, gender). Moreover, the model's performance should not vary between stations

Having this in consideration, columns with any outcome value and clothes removal cannot be used for model training as it introduces leakage in the model, meaning that we don't know this information prior to making a decision.

Concluding, unfortunately we are not expecting to achieve all the requirements for the model since in order to reduce bias we will be reducing the final score of the model or vice-versa.

Model Specifications

The type of supervised model that we want to train is a binary classification as our target, or label, only has two categorical values, Search or Don't Search, which in turn can be encoded as 1 and 0, respectively. This label, or target, was created by using both Outcome and Outcome related to the search columns. As previously explained, a search is considered successful, encoded as 1, if the Outcome is both positive and related to the search.

The data was split beforehand into training and testing sets in a stratified manner. Meaning that both sets should have the same percentage of the chosen value. Initially we planned on stratifying considering station, age, ethnicity, outcome and gender to try and have all class values present in equal percentage both in the training and test sets. This, in turn, would allow a more diverse dataset and to tackle the bias. However, this was not possible since it requires enough quantity of each class to exist so it can be split. Therefore, the dataset was only stratified in having the previously deduced target in consideration.

[1] See Annexes: Business questions technical support for more detail

Afterwards, we made use of pipeline objects to prepare the model for feature selection, missing values and the encoding of categories. The column with the Date information is properly parsed and converted to a timestamp and other features are extracted from it, namely, the hour, day, month and day of the week. Later on, only hour and day of the week were used.

Numeric missing values, as the hour, were imputed using the median value of the training set whereas categorical missing values were simply imputed using a “missing” label. Furthermore, numeric values were scaled between 0 and 1, to avoid any improper weighting, using a Standard Scaler. Even though we are using a tree based classifier as a model, we thought it important to scale the features as it prepares for future classifiers.

On the other hand, categories such as Gender, Ethnicity, amongst others, were encoded using a OneHotEncoder. This technique simply consists of converting every value to a column, attributing it 1 if the value is present and 0 in all other columns, meaning not present.

At this time, we have all features cleaned and encoded. We suggest using a tree based model, namely a Random Forest. One particularity of this dataset is that the target is unbalanced, meaning that almost every sample has a positive outcome which may lead to a bias of the model. To tackle this, the training data that was present in least quantity was over sampled, meaning that it was replicated until the quantity of both targets is equal. Moreover, the chosen model was initialized with a balanced class weight, with a maximum depth of 3 to avoid any overfitting.

Analysis of expected outcomes based on training set

The described model was fitted using the training set and tested with the test set. From the results obtained with our test set we have obtained the following results through the classification report:

-	precision	recall	**f1-score**	count
Search not successful - 0	0.87	0.50	0.64	49404
Search successful - 1	0.43	0.75	0.55	12460
accuracy			0.60	61964
Macro avg	0.60	0.59	**0.59**	61864
Weighted avg	0.75	0.59	0.66	61864

Table 1. Evaluation of the model performance

1 See Annexes: Business questions technical support for more detail

An AUROC of 0.59 was also obtained. Below, we can also see what are the most important features for the model, meaning what columns convey the best information for the model to make a decision on whether the stop will be successful or not.

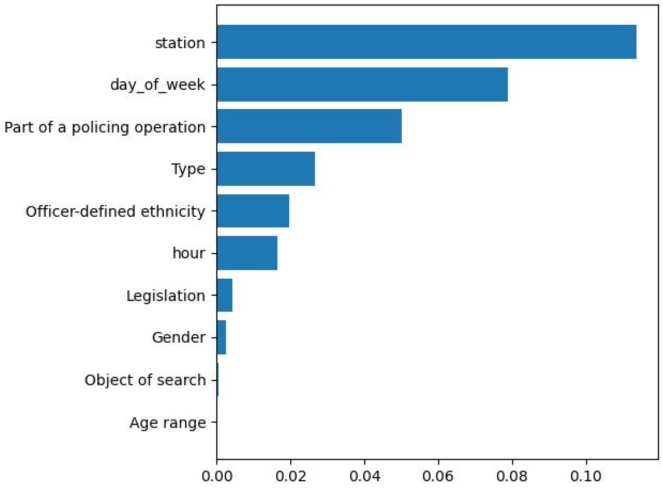


Figure 6. Feature importance

In terms of bias, the model had two requests to fulfil: that the success rate of the searches should not vary significantly (**5%**) between population sub-groups which were defined as (station, ethnicity, gender) and the average per station should not be larger than **10%**.

Note that this does not represent all stations since if the number of samples that were in the test set from the sensitive tuple was less than 30, the samples were removed from the analysis. With this in consideration, the model achieves a quite good performance in 23 stations for the problematic classes, failing in 15. Regarding the average performance between stations we have 17 stations that do not comply. This can also be seen in the Figure above as the model does not use sensitive columns to make a decision, proving once more that it is not biased in its decision.

Known issues and Risks

As already mentioned, not all requirements were achieved, having failed in some stations. Nevertheless, we believe we found a good median point where we fulfilled the number of requirements for many stations.

However, fulfilling almost all requirements came with a cost in performance, where the delivered model shows a not so good score. We hope we can work on this in the next iteration and update it accordingly. We plan on using more features in next iterations, using Legislation and Latitude and Longitude which might help by giving away the location of the stop. This, however, may introduce bias to the model.

[1] See Annexes: Business questions technical support for more detail

We can also receive new and unseen values for the variables that were not present in the training set. As of now, these variables are encoded as “missing” and do not add anything valuable for the model, not helping in the decision making.

Model Deployment

Deployment Specifications

The model pipeline was saved in a pickle file, together with a column descriptor and the corresponding types of each column. Moreover, these files were uploaded to our API which is hosted by *Awkward Problem Solutions*^{TM} and can receive requests and answers in real time, within less of a second.

Our rest API has 2 endpoints: */should_search/* and */search_result/*. Both endpoints are of type POST where the first one should be requested by the police officers in order to have a prediction. The following information should be sent in a JSON object: “observation_id”, "Type", "Date", "Part of a policing operation", "Latitude", "Longitude","Gender", "Age range", "Officer-defined ethnicity", "Legislation", "Object of search" and "station". The API will then answer, with a boolean, True or False, as a value of an object with key “outcome”. This outcome will define if the stop will be successful or not. All the data that is received by this endpoint will be saved in a database. If the observation already exists in the database, an error is sent saying that observation was already predicted.

Secondly, */search_result/*, allows you to provide any information about requests that were already made and saved. This endpoint allows to retrieve and update the outcome of older requests. It receives “observation_id” and the real “outcome” in order to update in the database the already existing observation_id with its true outcome. If the id does not exist, an error is provided. On the other hand, if the id exists, then the API returns the object_id, the outcome that was predicted by the model and the true outcome of the observation.

The api was deployed using heroku and its url is the following:
https://batch4-capstone-dbsousa01.herokuapp.com/

Known issues and Risks

Unit tests were not developed for the API. It is prepared to deal with any missing columns and values but not all variations were tested and some might pass through. Because of this, there is a risk where a police officer may happen to crash our API due to bad data imputation and never receive an answer back, not knowing the reason why. We plan on improving this in the next iteration.

[1] See Annexes: Business questions technical support for more detail

The server where we deployed the API has a limited database size dedicated to the application. If the maximum number of requests is achieved it may stop saving requests, which in turn, will not allow them to be updated with their true outcome. If this happens please contact us so we can increase the allocated memory.

Annexes

Dataset Technical Analysis

For a better visualisation you can download and read the html file here

660611 rows × 15 columns

```
df.columns
Index(['Type', 'Date', 'Part of a policing operation', 'Latitude', 'Longitude',
       'Gender', 'Age range', 'Self-defined ethnicity',
       'Officer-defined ethnicity', 'Legislation', 'Object of search',
       'Outcome', 'Outcome linked to object of search',
       'Removal of more than just outer clothing', 'station'],
      dtype='object')
```

```
df.isna().sum()
Type                                             0
Date                                             0
Part of a policing operation                153564
Latitude                                    112316
Longitude                                   112316
Gender                                           0
Age range                                        0
Self-defined ethnicity                        5574
Officer-defined ethnicity                        0
Legislation                                  27940
Object of search                                 0
Outcome                                          0
Outcome linked to object of search          473100
Removal of more than just outer clothing    426549
station                                          0
dtype: int64
```

Several columns with NaNs, most seems in unimportant columns which might not be used or can be replaced by False in case of binary

```
df.station.nunique(), df.station.unique()

 (42,
```

[1] See Annexes: Business questions technical support for more detail

```
 array(['devon-and-cornwall', 'dyfed-powys', 'derbyshire', 'bedfordshire',
        'avon-and-somerset', 'cheshire', 'sussex', 'north-yorkshire',
        'cleveland', 'merseyside', 'north-wales', 'wiltshire', 'norfolk',
        'suffolk', 'thames-valley', 'durham', 'warwickshire',
        'leicestershire', 'hertfordshire', 'cumbria', 'metropolitan',
        'essex', 'south-yorkshire', 'surrey', 'staffordshire',
        'northamptonshire', 'northumbria', 'city-of-london',
        'nottinghamshire', 'gloucestershire', 'cambridgeshire',
        'lincolnshire', 'btp', 'west-yorkshire', 'dorset', 'west-mercia',
        'kent', 'hampshire', 'humberside', 'lancashire',
        'greater-manchester', 'gwent'], dtype=object))
```

```
# how many different outcomes we have?
df.Outcome.nunique(), df.Outcome.unique()
```

```
(16,
 array(['A no further action disposal', 'Arrest', 'Community resolution',
        'Summons / charged by post', 'Khat or Cannabis warning',
        'Caution (simple or conditional)', 'Penalty Notice for Disorder',
        'Nothing found - no further action',
        'Offender given drugs possession warning', 'Local resolution',
        'Suspect arrested', 'Article found - Detailed outcome unavailable',
        'Offender cautioned', 'Suspect summonsed to court',
        'Offender given penalty notice',
        'Suspected psychoactive substances seized - No further action'],
       dtype=object))
```

```
df.Outcome.value_counts(normalize = True)
```

```
A no further action disposal                                    0.689274
Arrest                                                          0.142471
Community resolution                                            0.062535
Khat or Cannabis warning                                        0.025996
Nothing found - no further action                               0.024505
Summons / charged by post                                       0.018516
Penalty Notice for Disorder                                     0.016975
Suspect arrested                                                0.007037
Caution (simple or conditional)                                 0.004812
Offender given drugs possession warning                         0.003182
Local resolution                                                0.002018
Suspect summonsed to court                                      0.000963
Article found - Detailed outcome unavailable                    0.000793
Offender given penalty notice                                   0.000627
Offender cautioned                                              0.000286
Suspected psychoactive substances seized - No further action    0.000011
```

[1] See Annexes: Business questions technical support for more detail

```
Name: Outcome, dtype: float64

# how many age ranges we have?
df["Age range"].nunique(), df["Age range"].unique()

(5, array(['18-24', '25-34', 'over 34', '10-17', 'under 10'], dtype=object))
```

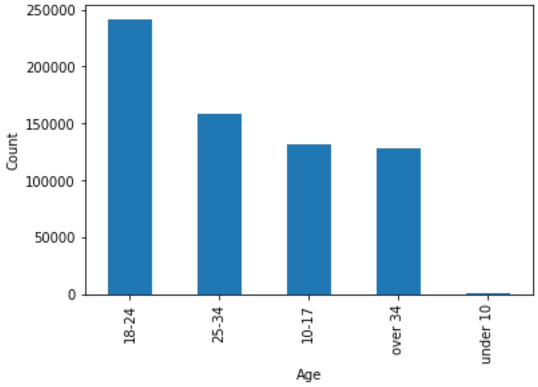


Figure 7.

```
df["Age range"].value_counts(normalize=True)

18-24       0.366184
25-34       0.239447
10-17       0.199456
over 34     0.194332
under 10    0.000581
Name: Age range, dtype: float64
```

There seems to be a weird age which is under 10, most likely to be dropped

```
# Let's check on gender distribution
```

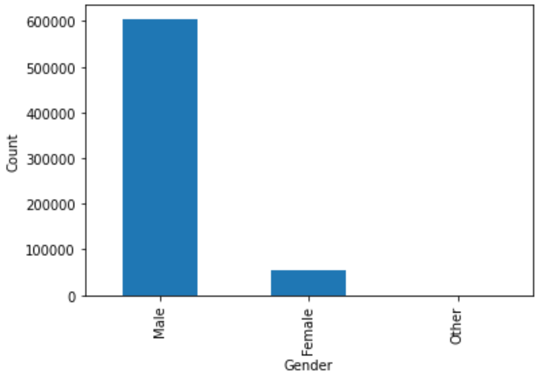


Figure 8.

```
df["Gender"].value_counts(normalize=True)

Male      0.916458
Female    0.082972
```

[1] See Annexes: Business questions technical support for more detail

```
Other      0.000571
Name: Gender, dtype: float64
```

the class looks very unbalanced when comparing Male to Female - Other might be dropped

```
# Let's check on clothing removal info
df["Removal of more than just outer clothing"].value_counts().plot(kind="bar")
```

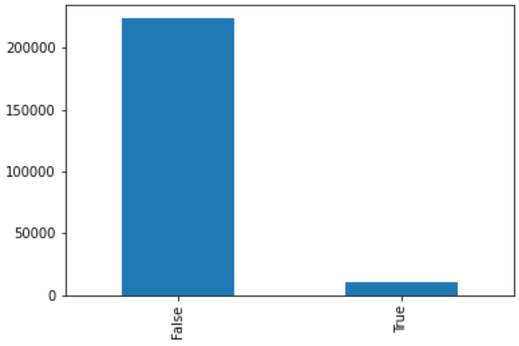


Figure 9.

```
# Let's check on clothing removal info
df["Removal of more than just outer clothing"].value_counts(normalize=True)

False     0.954435
True      0.045565
Name: Removal of more than just outer clothing, dtype: float64
```

Again, a very high unbalance in both classes - even more if we convert the NaNs to False

There are a lot of NaN values around this class...hmm what to do ? (426549/660611) -> 64,6%

```
# and what gender does the NaNs most belong to?
df[df["Removal of more than just outer
clothing"].isna()].Gender.value_counts().plot(kind="bar")
```

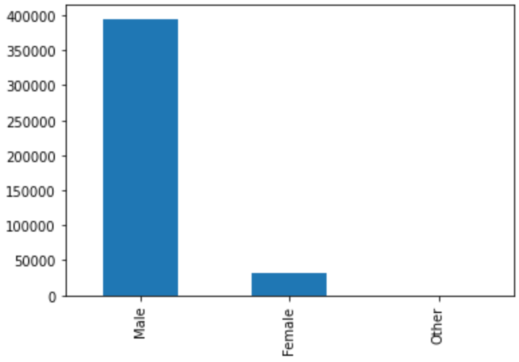


Figure 10.

```
df[df["Removal of more than just outer
clothing"].isna()].Gender.value_counts(normalize=True)
```

[1] See Annexes: Business questions technical support for more detail

```
Male      0.925722
Female    0.073863
Other     0.000415
Name: Gender, dtype: float64
```

It still presents the same distribution has the Gender in the dataset - need to ask about this to the client since this class that we need to track is very much filled with NaNs

```
# Let's check ethnicity now
df["Self-defined ethnicity"].value_counts().plot(kind="bar")
```

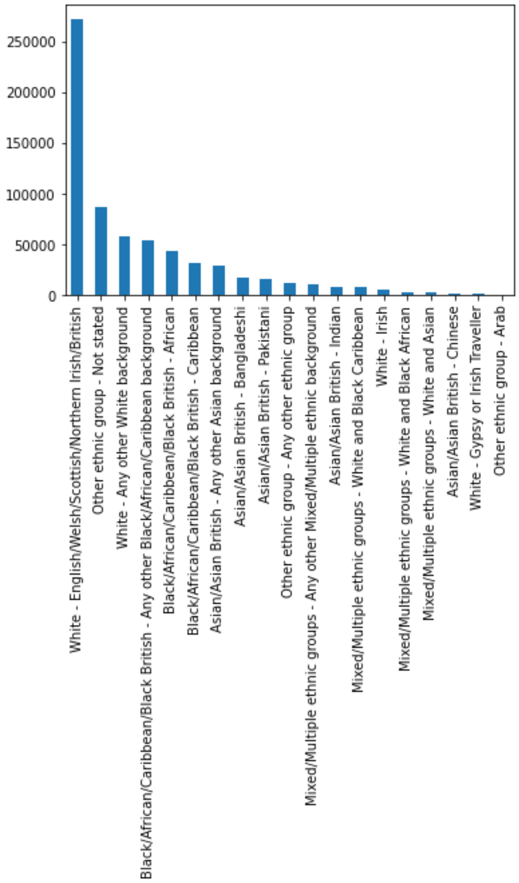


Figure 11.

```
df["Officer-defined ethnicity"].value_counts().plot(kind="bar")
```

[1] See Annexes: Business questions technical support for more detail

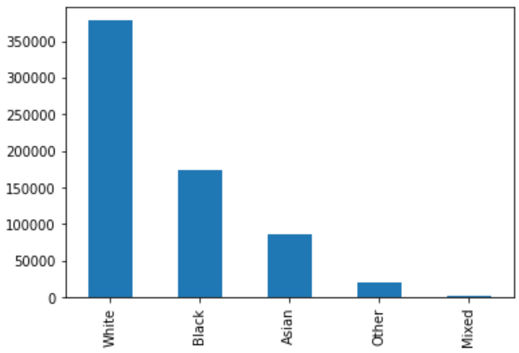


Figure 12.

the latter seems a much better defined class but it might introduce the officer's bias

```
# Misc types of distribution
df["Type"].value_counts().plot(kind="bar") # might be a good class for one hot
encoding?
```

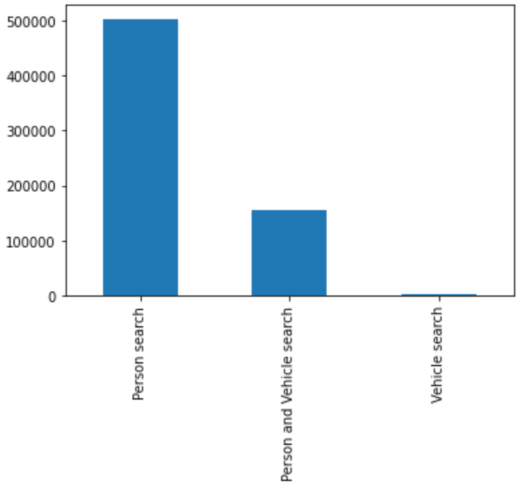


Figure 13.

```
df["Part of a policing operation"].value_counts().plot(kind="bar")
```

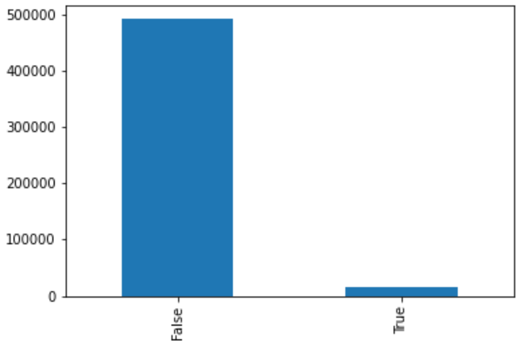


Figure 14.

```
df["Legislation"].value_counts().plot(kind="bar")
```

[1] See Annexes: Business questions technical support for more detail

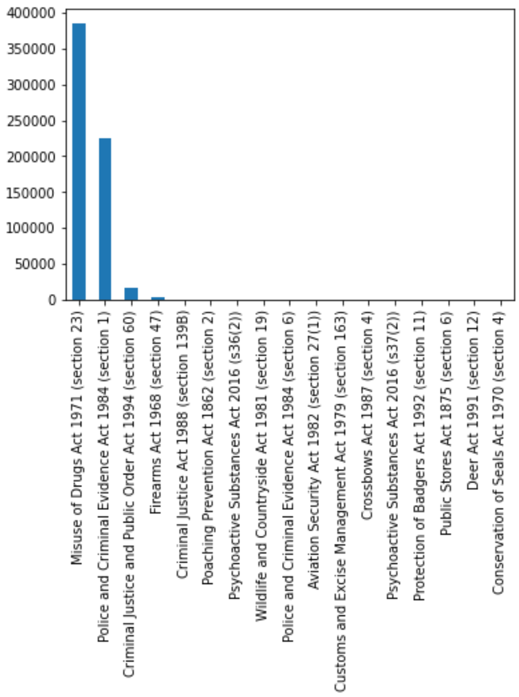


Figure 15.

```
df["Object of search"].value_counts().plot(kind="bar")
```

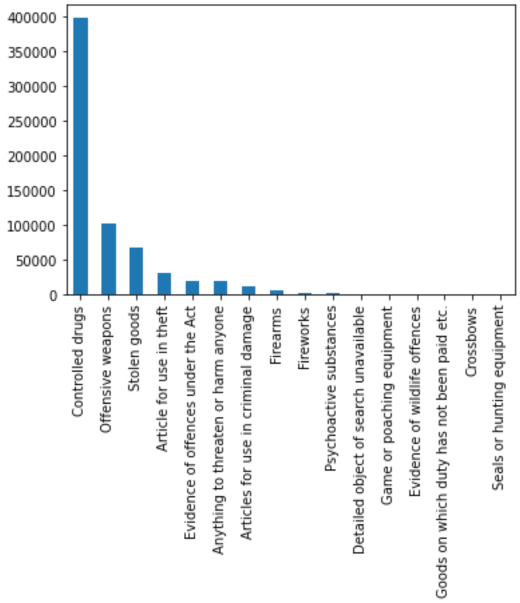


Figure 15.

1 See Annexes: Business questions technical support for more detail

Business Questions Technical Support

Precision

$$Precision = \frac{True\ Positives}{True\ Positives + True\ Negatives}$$

Precision, as defined above, is the division between the number of True Positives (cases that are predicted being positive and are, in fact, positive - TP) and the sum of True Positives and True Negatives (cases that are predicted being negative and are, in fact, negative). Precision can be defined as the fraction of relevant instances among the retrieved instances and is a measure of correctness of the model.

There are also False Positives, which are cases that are predicted being positive but are in fact negative and False Negatives, which are the other way around, being predicted as negatives but are in fact positives.

Imbalanced Classification Problem

Term that is used to identify a labelling problem, in this case, a successful search or not, that has a relatively high difference in quantity of the same labels, one occurring much more often than the other.

Gender analysis

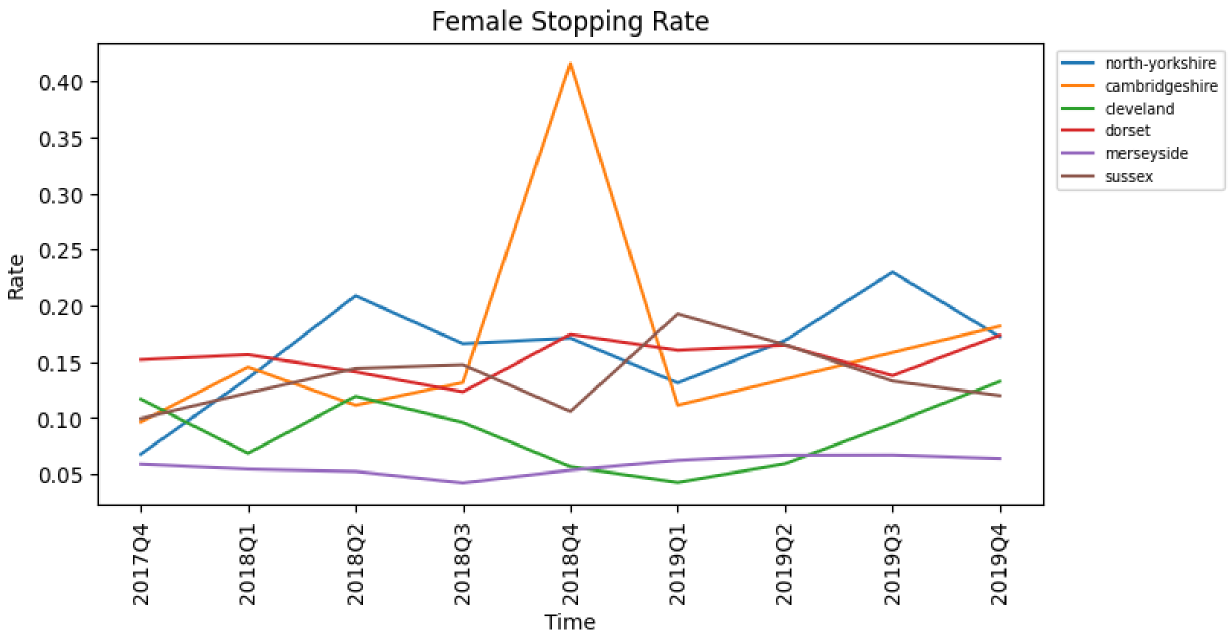


Figure 16. Female stopping rate over time

1 See Annexes: Business questions technical support for more detail

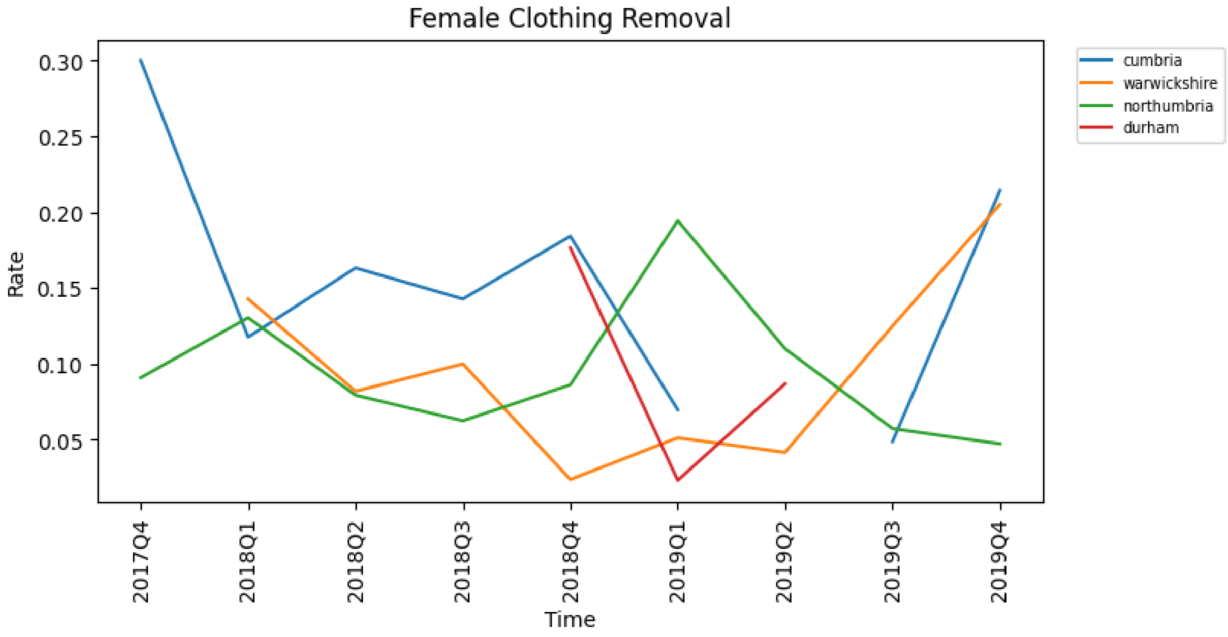


Figure 17. Female clothes removal rate over time

Age

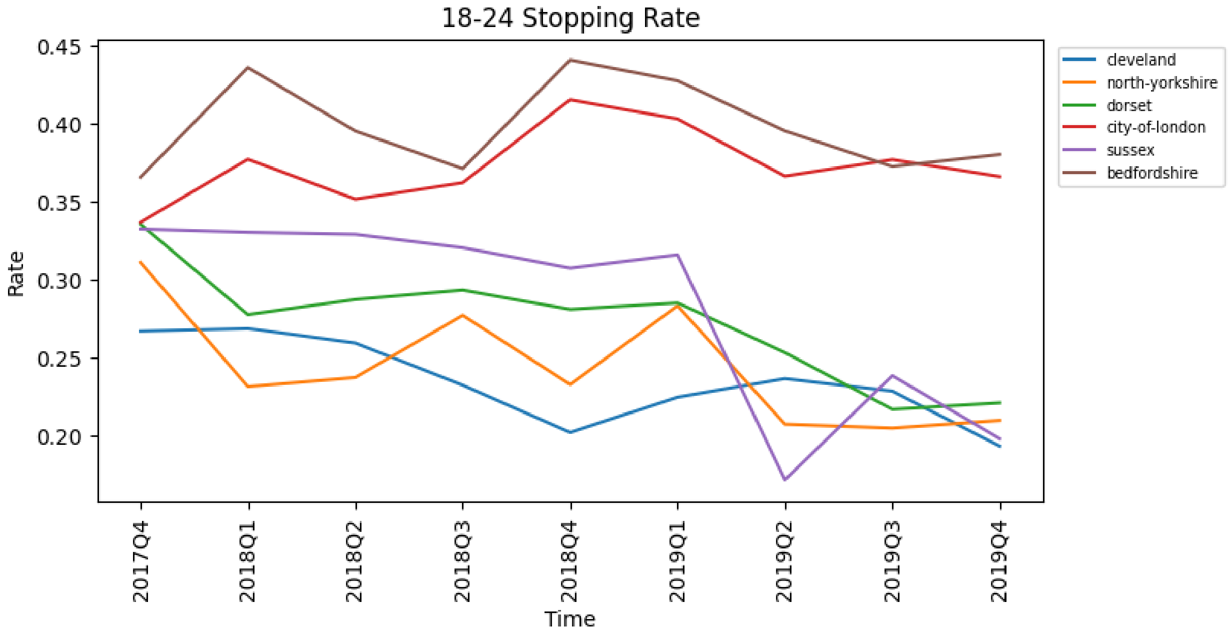


Figure 18. Age range stopping rate over time

1 See Annexes: Business questions technical support for more detail

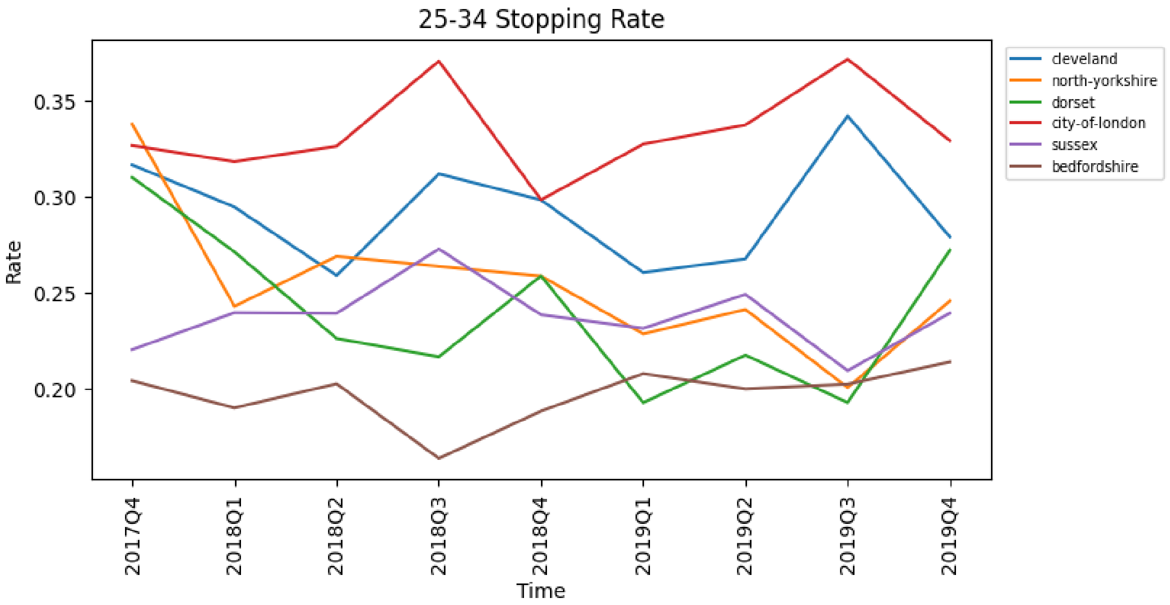


Figure 19. Age range stopping rate over time

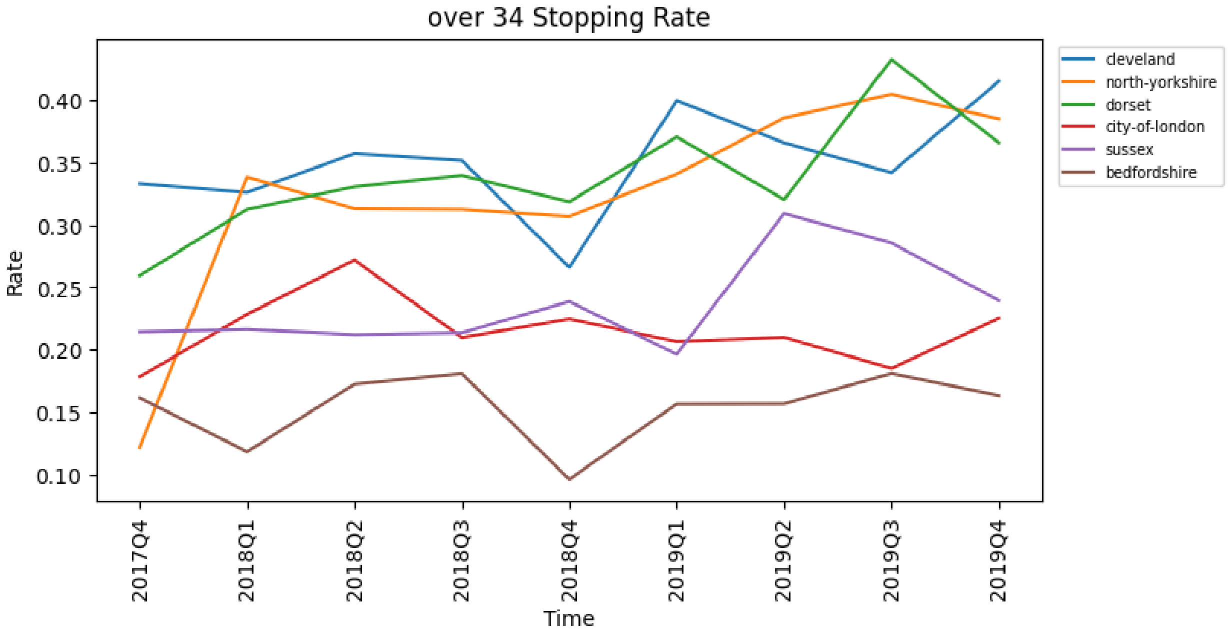


Figure 20. Over 34 Age range stopping rate over time

[1] See Annexes: Business questions technical support for more detail

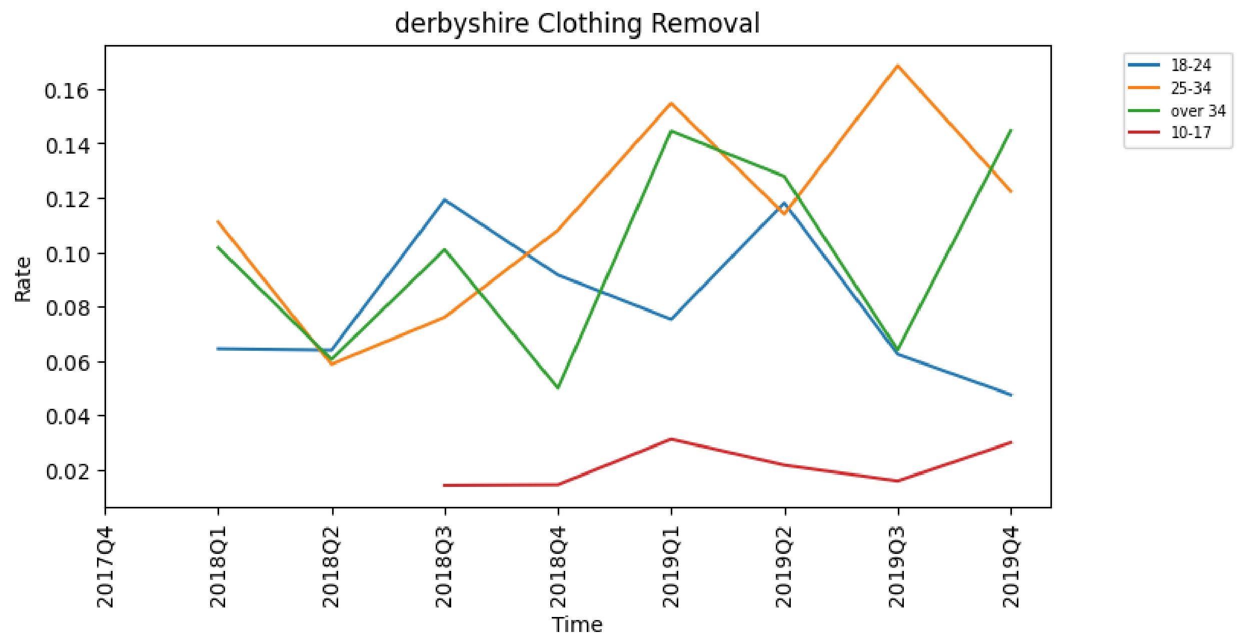


Figure 21. Derbyshire clothing removal rate over time.

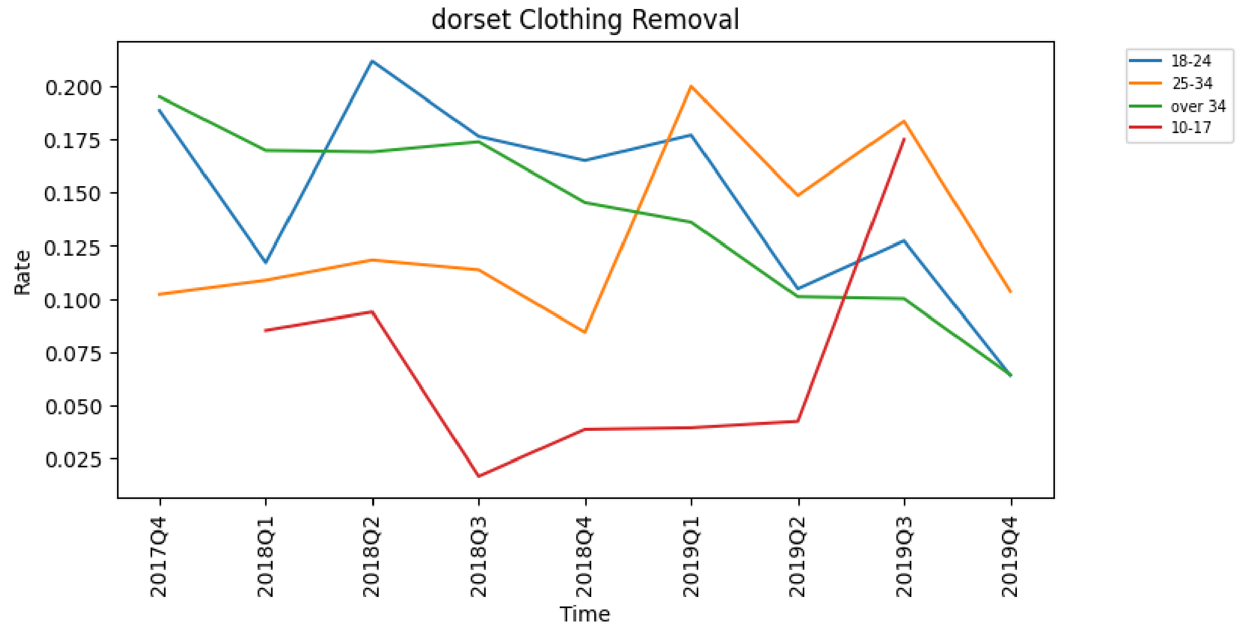


Figure 22. Dorset clothing removal rate over time.

[1] See Annexes: Business questions technical support for more detail

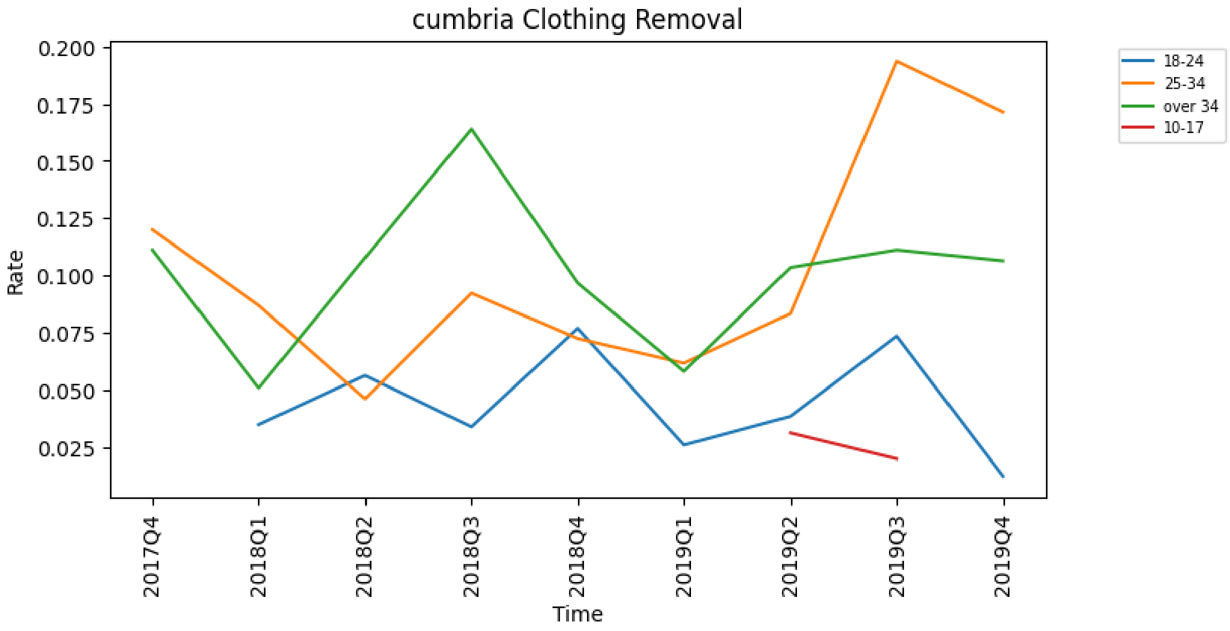


Figure 23. Cumbria clothing removal rate over time.

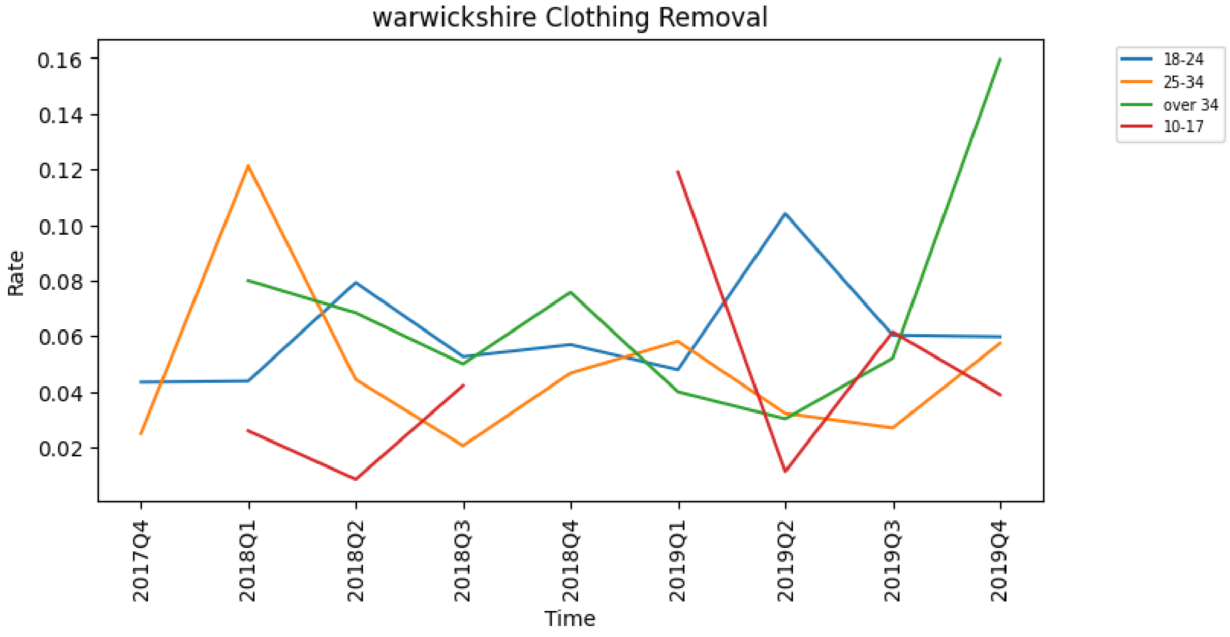


Figure 24. Warwickshire clothing removal rate over time.

[1] See Annexes: Business questions technical support for more detail

Ethnicity

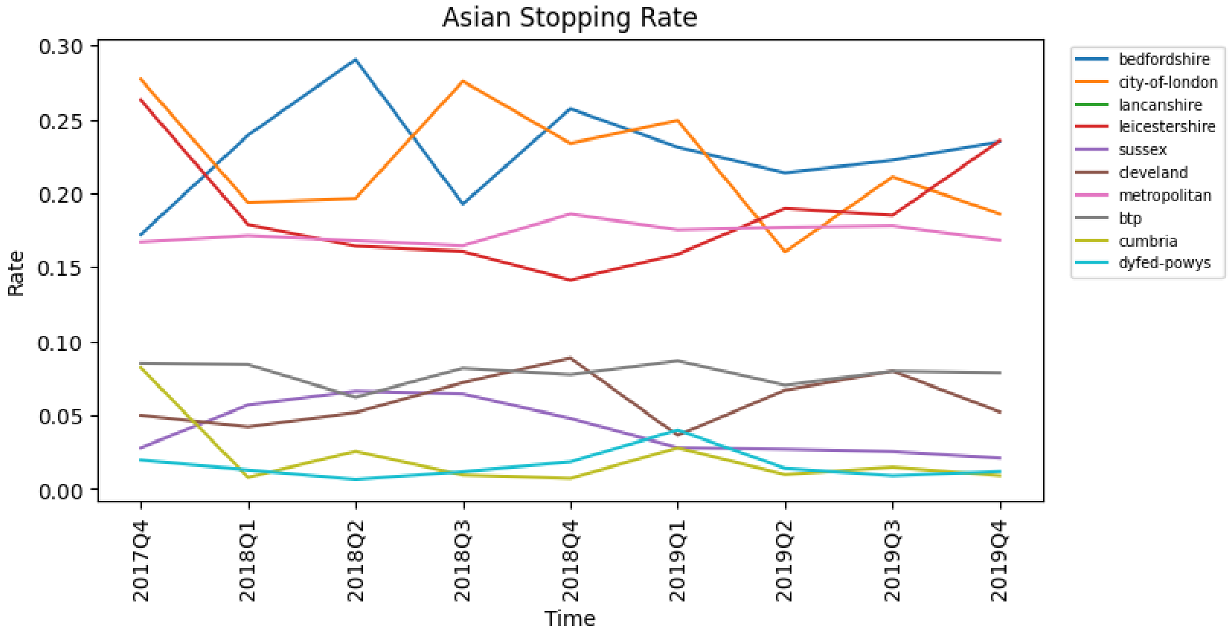


Figure 25. Asian stopping rate over time

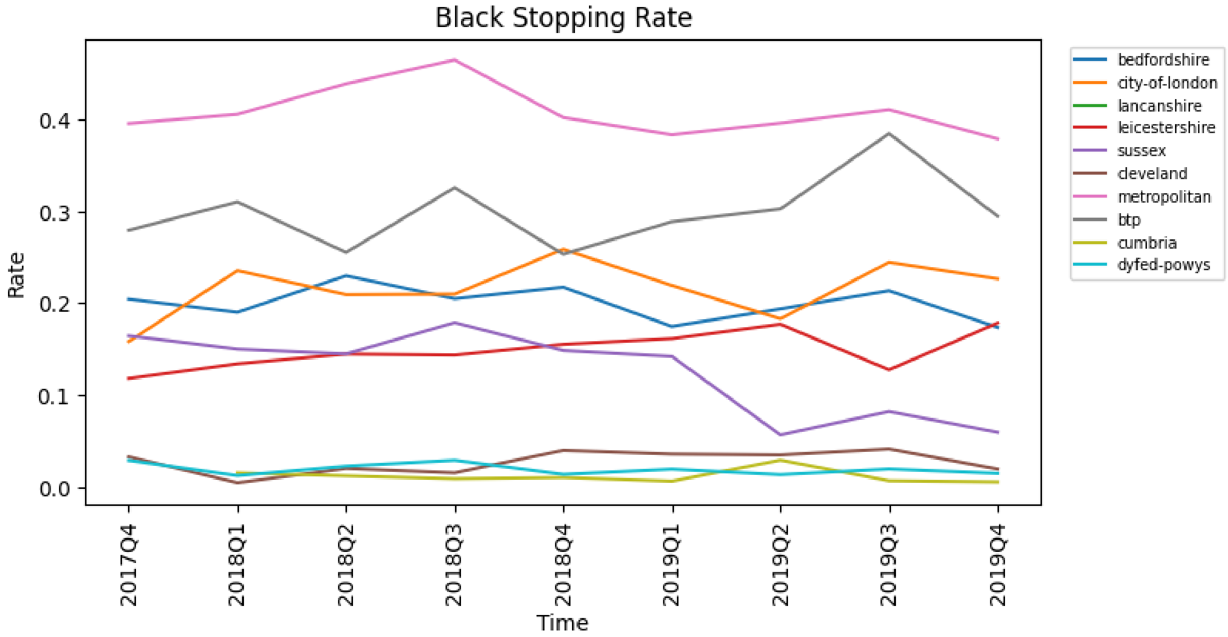


Figure 26. Black stopping rate over time

[1] See Annexes: Business questions technical support for more detail

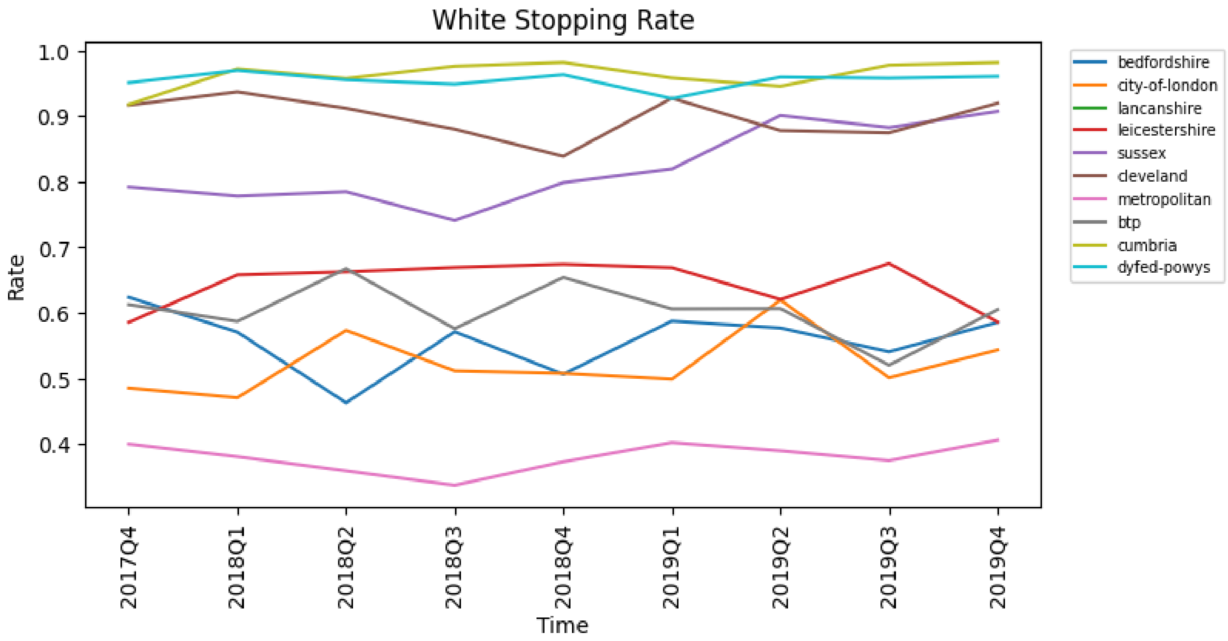


Figure 27. White stopping rate over time

1 See Annexes: Business questions technical support for more detail