PROJECT REPORT: OSAS Iteration of Analysis of gender bias in policing, to address the problem of gender inequality.

By Ahmed Suhail
asuh702
for INFOSYS722

Contents

1. Situation Understanding

1.1. Objectives of the situation: Analysis of gender bias in policing, to address the problem of gender inequality.

How often does gender come into play when traffic cops stop motorists on the road? For several years now, especially since the Black Lives Matter movement started, people have advocated for better policing. There is an immediate need for analysis of how policing is done, spot discrimination (if any!). There is also an immediate need to empower women and address the problem of Gender Equality, as is stated in the UN's Sustainable Development Goals (https://bit.ly/1Qk5cqI).

1.2. An interdisciplinary team of statisticians, computer scientists, and journalists at Stanford University came together to study possible bias in policing. They "filed public record requests with all 50 states to obtain details of each stop carried out by state patrol officers over the last 10 years".(Pierson et al., 2017) They "found that black and Hispanic drivers were often searched based on lesser evidence than whites. This double standard was widespread, not confined to any one state or geographic region." Data was then made to the available to the public, to facilitate further discussion of policing practices. The team hoped, "this resource helps researchers, policymakers, and journalists assess and improve police contact with the public."(Sharad, Goel & Philips, 2017)

A problem with crime analysis is data is often private, and not all parts are safe for release to the public. Similar was the case with this project, wherein to protect the privacy of individuals, number plate details of the drivers and the location of the stop was removed.

Also, police officers might have been biased in their stops and might have falsely added some attributes, such as listing suspicious behaviour as reason for search when in actuality the person was just fine.

1.3. Data Mining Objectives

i. Analysis of gender discrimination in police stops in parts of USA (Florida for this report)

ii. Evaluate the effects of date, race, and other features available on traffic stops by the police.

iii. Analyse for discrimination in stops by gender using a test that effectively categorizes discriminatory stops from non-discriminatory ones, building on the ISAS Iteration.

1.4. Data Mining Success Criteria

i. Successful examination and study of the rate at which male and female drivers are stopped and arrested, of the hypotheses that arrests are discriminatory and being able to state with confidence if they are/are not.

ii. Expand on the ISAS Iteration by coming up with a test that better classifies discriminatory stops from non-discriminatory ones. Perform other types of modelling and gather successful insights.

2. Data Understanding

2.1. Data Collection: An issue when collecting data for crime analysis is that law enforcement agencies do not make data publicly available because of vested interests and privacy concerns. Police data is often inaccessible, making rigorous analysis difficult. A database of over 100 million records from 31 states was collected by a research team at Stanford University. All the data collected was cleaned to some extent and made available for public use, in hope that data analysts over the world can dive deeper into the data to more insights and build upon their work.

Link to the project: https://openpolicing.stanford.edu/

2.2. Data Description:

i. The data was in files of the format .csv (Comma Separated Values) and .rds for RStudio. The data was loaded into a Pandas dataframe, which is a data manipulation library for Python. An interesting feature of dataframes is that they can store different types of data, making modelling and manipulation easier.

ii. Data was publicly available for download around June 2017, for analysis and help in getting better insights. Studies of location-specific data and evaluation of effects of different features on stops, better analyses were be made.

iii. For the OSAS iteration, focus was on data from Rhode Island, expanding on the findings from ISAS Iteration. The State Patrol from Rhode Island had provided data of all vehicular stops from January 2005 to December 2015, which were analysed in the ISAS Iteration. This iteration builds on that, proposing a better test for discrimination and using other modelling techniques for insights.

iv. The available data consisted of the following features:

- raw_row_number: A record ID for reference to the raw data
- Date: Stop date in the format of YYYY-MM-DD
- Stop Time: Time of the Stop, in a 24 hour, HHMM format.
- Stop Location: Textual, non-standardised location.
- Driver Race: Race of the stopped subject.
- Driver Age: Age of the stopped subject.
- Driver Sex: Gender of the stopped subject.
- Search Conducted: Boolean value that indicates whether a search was conducted.
- Contraband Found: Boolean value that indicates if contraband was found
- Citation Issued: Boolean value indicating outcome, if a citation was issued.
- Warning Issued: If a stop resulted in the issue of warning, this Boolean was selected.
- Frisk Performed: If a quick Frisk was performed, this Boolean value was set True
- Arrest Made: If the person were arrested, this Boolean value was set True.
- reason_for_stop: The visible reason for stopping the driver.

2.3. Data Exploration: Initial exploration of data reveals the need to prepare data and fix errors, and missing values before plots for better understanding can be made. Data was explored using functions from the pandas library – standard library for data manipulation and analysis. The effects of group and zone on male and female arrests were also evaluated, which suggest males are more likely to be arrested than females.

In [1]:
```
import pandas as pd
ri = pd.read_csv(r'C:\Users\iahme\Documents\722\Rhode Island Statewise Dataset\share\data\opp-for-archive\ri_statewide_2019_02_2!
ri.head()
```

```
C:\Users\iahme\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3057: DtypeWarning: Columns (6,17,18) have mixed ty
pes. Specify dtype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

Out[1]:

	raw_row_number	date	time	zone	subject_race	subject_sex	department_id	type	arrest_made	citation_issued	...	contraband_weapons	contrab
0	1	2005-11-22	11:15:00	X3	white	male	200	vehicular	False	True	...	NaN	
1	2	2005-10-01	12:20:00	X3	white	male	200	vehicular	False	True	...	NaN	
2	3	2005-10-01	12:30:00	X3	white	female	200	vehicular	False	True	...	NaN	
3	4	2005-10-01	12:50:00	X3	white	male	200	vehicular	False	True	...	NaN	
4	5	2005-10-01	13:10:00	X3	white	female	200	vehicular	False	True	...	NaN	

5 rows × 24 columns

Figure 1: Importing data into a dataframe and taking a glimpse.

```
In [8]: print(ri.groupby(['reason_for_stop', 'subject_sex']).search_conducted.mean())
```

```
reason_for_stop                   subject_sex
APB                               female         0.165138
                                  male           0.255319
Call for Service                  female         0.042230
                                  male           0.092419
Equipment/Inspection Violation    female         0.040245
                                  male           0.070916
Motorist Assist/Courtesy          female         0.033133
                                  male           0.089802
Other Traffic Violation           female         0.038021
                                  male           0.059156
Registration Violation            female         0.054700
                                  male           0.103589
Seatbelt Violation                female         0.017746
                                  male           0.031705
Special Detail/Directed Patrol    female         0.018045
                                  male           0.010249
Speeding                          female         0.007738
                                  male           0.026630
Suspicious Person                 female         0.216216
                                  male           0.305970
Violation of City/Town Ordinance  female         0.060185
                                  male           0.073171
Warrant                           female         0.148148
                                  male           0.311111
Name: search_conducted, dtype: float64
```

Figure 2: Males are more likely to be arrested for various reasons than females.

```
In [9]: print(ri.groupby(['zone', 'subject_sex']).search_conducted.mean())
```

```
zone  subject_sex
K1    female         0.016528
      male           0.038785
K2    female         0.013018
      male           0.031662
K3    female         0.023989
      male           0.052726
X1    female         0.005181
      male           0.013020
X3    female         0.013057
      male           0.030472
X4    female         0.026174
      male           0.059040
Name: search_conducted, dtype: float64
```

Figure 3: Male and female arrests controlled for zones

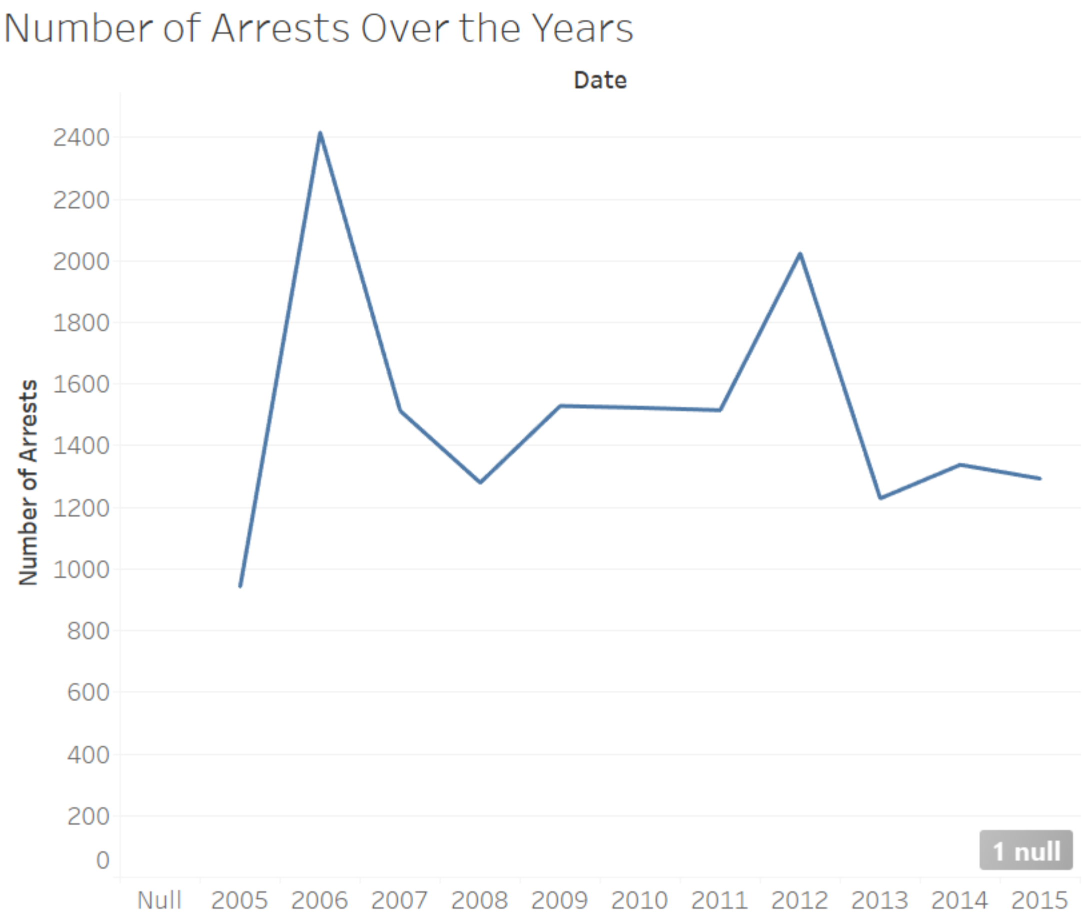


Figure 4: Number of arrests over the years using Tableau.

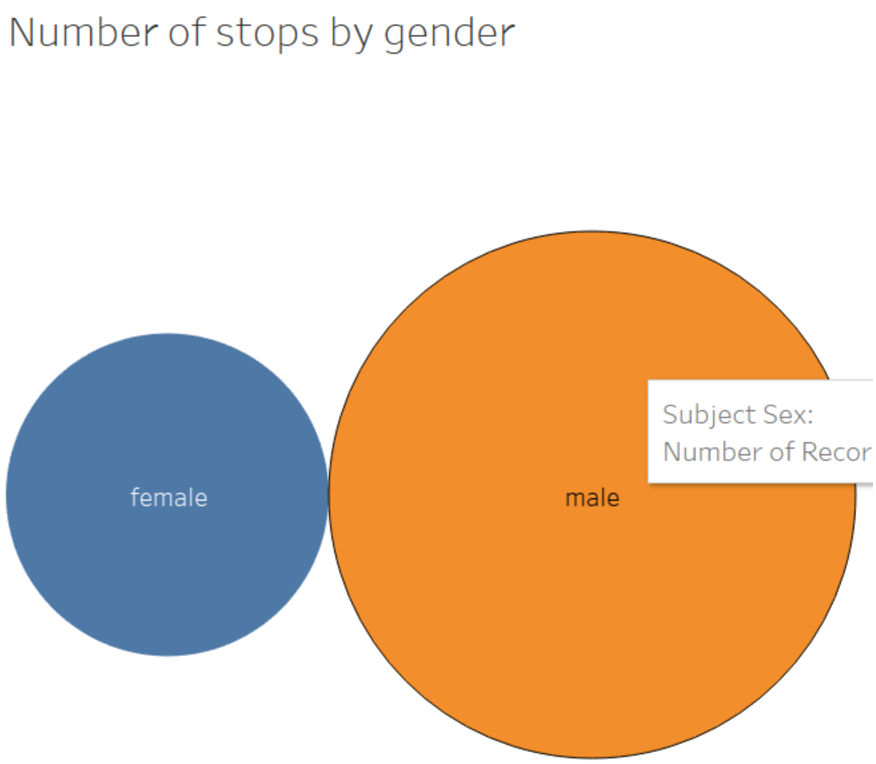


Figure 5: Number of stops in Rhode Island vs sex using Tableau

2.4. Data Quality: The ri.isnull().sum() function indicated good quality data, with not much time being spent on data cleaning. Since data was already cleaned for ISAS, I already had an intuition of how the data needed to be processed. with thoughts that very less time would be spent on Data Preparation.

```
In [3]: ri.isnull().sum()

Out[3]: raw_row_number              0
        date                       10
        time                       10
        zone                       10
        subject_race            29073
        subject_sex             29097
        department_id              10
        type                        0
        arrest_made             29073
        citation_issued         29073
        warning_issued          29073
        outcome                 35841
        contraband_found       491919
        contraband_drugs       493693
        contraband_weapons     497886
        contraband_alcohol     508464
        contraband_other            0
        frisk_performed            10
        search_conducted           10
        search_basis           491919
        reason_for_search      491919
        reason_for_stop         29073
        vehicle_make           191564
        vehicle_model          279593
        dtype: int64
```

Figure 6: Missing values in data.

```
In [4]: ri.dropna(subset=['subject_sex'], inplace=True)

In [5]: ri.isnull().sum()

Out[5]: raw_row_number              0
        date                        0
        time                        0
        zone                        0
        subject_race                0
        subject_sex                 0
        department_id               0
        type                        0
        arrest_made                 0
        citation_issued             0
        warning_issued              0
        outcome                  6763
        contraband_found       462822
        contraband_drugs       464596
        contraband_weapons     468789
        contraband_alcohol     479367
        contraband_other            0
        frisk_performed             0
        search_conducted            0
        search_basis           462822
        reason_for_search      462822
        reason_for_stop             0
        vehicle_make           162525
        vehicle_model          250553
        dtype: int64
```

Figure 7: Cleaning missing records

3. Data Preparation:

3.1. Data Types of different features in the dataset were corrected for better exploratory analysis and plots. The dataset was *cleaned* by ensuring that each of the columns has the proper data type.

```
In [7]: ri['subject_sex'] = ri.subject_sex.astype('category')
        ri['subject_race'] = ri.subject_race.astype('category')
        ri['arrest_made'] = ri.arrest_made.astype('bool')
        ri['citation_issued'] = ri.citation_issued.astype('bool')
        ri['warning_issued'] = ri.warning_issued.astype('bool')
        ri['frisk_performed'] = ri.frisk_performed.astype('bool')
        ri['reason_for_stop'] = ri.reason_for_stop.astype('category')
        ri['search_conducted'] = ri.search_conducted.astype('bool')
        ri['contraband_found'] = ri.search_conducted.astype('bool')
```

Figure 8: Corrected data types

3.2. During the instantiation, the types of data were set, with time and date using the timestamps and other values correctly being set. Missing values were filtered out from the dataset, using the drop function, because the dataset was large enough to let go of some of the missing values. The drop function was run separately for records and features in the Jupyter Notebook, wherever was felt necessary separately for exploration and modelling. Cleaning the records that are not useful for my analysis was important for better performance.

```
In [4]: ri.dropna(subset=['subject_sex'], inplace=True)

In [5]: ri.isnull().sum()

Out[5]: raw_row_number          0
        date                    0
        time                    0
        zone                    0
        subject_race            0
        subject_sex             0
        department_id           0
        type                    0
        arrest_made             0
        citation_issued         0
        warning_issued          0
        outcome              6763
        contraband_found   462822
        contraband_drugs   464596
        contraband_weapons 468789
        contraband_alcohol 479367
        contraband_other        0
        frisk_performed         0
        search_conducted        0
        search_basis       462822
        reason_for_search  462822
        reason_for_stop         0
        vehicle_make       162525
        vehicle_model      250553
        dtype: int64
```

Figure 9: Missing Values Select Node Table

3.3. Wherever necessary, data was appropriately constructed, like forming a year column by trimming the date string to contain values of year. Constructing such a column could be good for modelling, as I've done and indicated in previous steps.

3.4. The dataset was split into two, and integrated using read_csv() method. Refer to the Jupyter notebook.

3.5. For analysis of discrimination in searches, only records where a contraband was found were selected for modelling. The idea behind this test was that a police stop is not discriminatory, if the stop resulted in contraband being found from search. If the rate at which females were searched and the rate at which males were searched resulted in same rates in contraband being found for both the categories, then the stops were not discriminatory.

```
In [7]: ri.contraband_found.value_counts()

Out[7]: False    11183
        True      6579
        Name: contraband_found, dtype: int64
```

Figure 10: Checking for values where contraband is found.

4. Data Transformation

4.1. For modelling, the data required to be transformed into only numerical values, as scikit learn, the primary Machine Learning Library for Python used for modelling only works with numerical values. The labelEncoder module was used to create a new dataframe with label encoded values, which would be then fit to the model. Only features that a police officer would know before stops and conducting searches, which would be relevant to the model were selected. The model was built with various features for input in the SASS Itearation, so I already knew which of the features yield best results.

```
In [16]: #Create a dataframe where frisk was performed
         ri2 = pd.DataFrame(ri)

In [17]: #Label Encoding
         from sklearn import preprocessing

         #create the LabelEncoder object
         le = preprocessing.LabelEncoder()

         #convert the categorical columns into numeric
         ri2['date'] = le.fit_transform(ri2['date'])
         ri2['zone'] = le.fit_transform(ri2['zone'])
         ri2['subject_race'] = le.fit_transform(ri2['subject_race'])
         ri2['subject_sex'] = le.fit_transform(ri2['subject_sex'])
         ri2['reason_for_stop'] = le.fit_transform(ri2['reason_for_stop'])


         #display the initial records
         ri2.head()
```

Figure 11: Code for Label Encoding.

Out[17]:

	date	zone	subject_race	subject_sex	arrest_made	contraband_found	frisk_performed	search_conducted	reason_for_stop
0	112	4	4	1	False	False	False	False	8
1	60	4	4	1	False	False	False	False	8
2	60	4	4	0	False	False	False	False	8
3	60	4	4	1	False	False	False	False	8
4	60	4	4	0	False	False	False	False	8

Figure 12: head() method revealing encoded dataset

4.2. The data was split into train and test datasets, for validation of results from the predictive Linear regression modelling that was to be done. Splitting the dataset into train and test ensures that your model is effective on data it hasn't seen before, and confirms it's validity. A 70/30 split ensures there are enough values for training and testing.

```
In [20]: from sklearn.model_selection import train_test_split

         y = ri2['contraband_found'].values
         X = ri2.drop('contraband_found', axis=1).values #Drop values linearly related also

         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=100)
```

Figure 13: Train Test Split

Some of the concerns that remain are that there is no possible effective comparison with the base population, as there is no effective dataset that might contain the records of all the drivers in Rhode Island. Therefore, a base comparison between male and female drivers, though helpful, is not possible.

5. Data Mining Method Selection:

5.1. Both Supervised and Unsupervised techniques were hence used, for building on findings from the previous iterations and for looking for patterns using KMeans Clustering. KMeans Clustering was used to find patterns in the data that miss the eye, but initial runs of the model didn't yield any helpful results. I didn't want to waste much time there, and instead build on the SPSS Modeler Iteration, so I focused mainly on Logistic Regression.

5.2. After positive results of the Linear Regression performed in ISAS Iteration, Logistic Regression was run again, with a better test format. The dataset was filtered out to contain only stops where searches were conducted. Then, the model was fit in order to predict if contraband was found or not. This follows from the strategy adopted by the researchers of the Stanford Open Policing Project, where they argue that **a stop is not discriminatory if it results in contraband being found in males at the same rate as it results in contraband being found in females.** Unsupervised learning techniques, which would help find similar rows in data by clustering them together, were also run on the data. The primary machine learning library in Python, scikit learn has many clustering algorithms, and a consistent API for accessing modelling techniques.

```
In [29]: #Unsupervised Learning using KMeans

         from sklearn.cluster import KMeans

         model = KMeans(n_clusters = 2)
         model.fit(ri)

         labels = model.predict(ri) #Predict on the same data
```

```
In [30]: #KMeans Evaluation
         df = pd.DataFrame({'labels': labels, 'Arrested or not': ri['arrest_made']})

         print(df)

         #Cross Tabulation to check meaningful clusters
         ct = pd.crosstab(df['labels'], df['Arrested or not'])
         print(ct)
```

Figure 14: Clustering using KMeans from scikit learn

```
[480584 rows x 2 columns]
Arrested or not   False  True
labels
0                225430  8846
1                238551  7757
```

Figure 15: Cross Tabulation results from KMeans

6. Data Mining Algorithm Selection

6.1. Logistic regression is the most suitable algorithm for this kind of Supervised Analysis on this data (dependent variable binary, meaningful variables, independent variables independent to one another, large size dataset, not much computational resources are required, highly interpretable, not much data transformation was needed) which led me to fit a Logistic Regression model on the data. Logistic over linear regression was chosen because of no linear relationship between dependent and independent variables. Some people might also have advocated for discriminant analysis, but my research found that for not a categorization problem, Logistic Regression would be better

6.2. . Moreover, a similar model being run for a **different test** would establish and help better analyse if the discrimination between men and women in policing is as much as the LR Model in ISAS showed it was. **Logistic Regression was run with filtered values where searches were conducted looking for values where contraband was found.** Also, Unsupervised Learning using KMeans failed to cluster the data into logical clusters, as was verified using cross tabulation.

6.3. A Logistic Regression model was successfully fit into the training partition of the data, with 6 features which were healthy features with data that the officer knew prior to conducting a search, that would help him determine whether the suspect would have contraband or not. Also, features linearly independent and not parallel were chosen for best results with logistic regression. Date, Zone, Race, Sex, and Reason for stopping were some of the features that were fit for modelling. The idea was to interpret odds ratio as how much more likely is it that contraband will be found when search is conducted in males vs females. (Ed, n.d.)

```
In [21]: #Logistic regression
         from sklearn.linear_model import LogisticRegression

         # instantiate the model (using the default parameters)
         logreg = LogisticRegression()

         #fit the model with data
         logreg.fit(X_train,y_train)

         #Predictions
         y_pred=logreg.predict(X_test)

C:\Users\iahme\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning: Default solver will be changed
to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
```

Figure 16: Logistic Regression model fitting using scikit learn.

7. Data Mining:

7.1. After running multiple iterations with different parameters, a Logistic Regression Model was built. Train and test were 70/30 (universally accepted ratios of train and test sizes so that enough data is available for training and you also have enough data for validating your results). A model evaluation

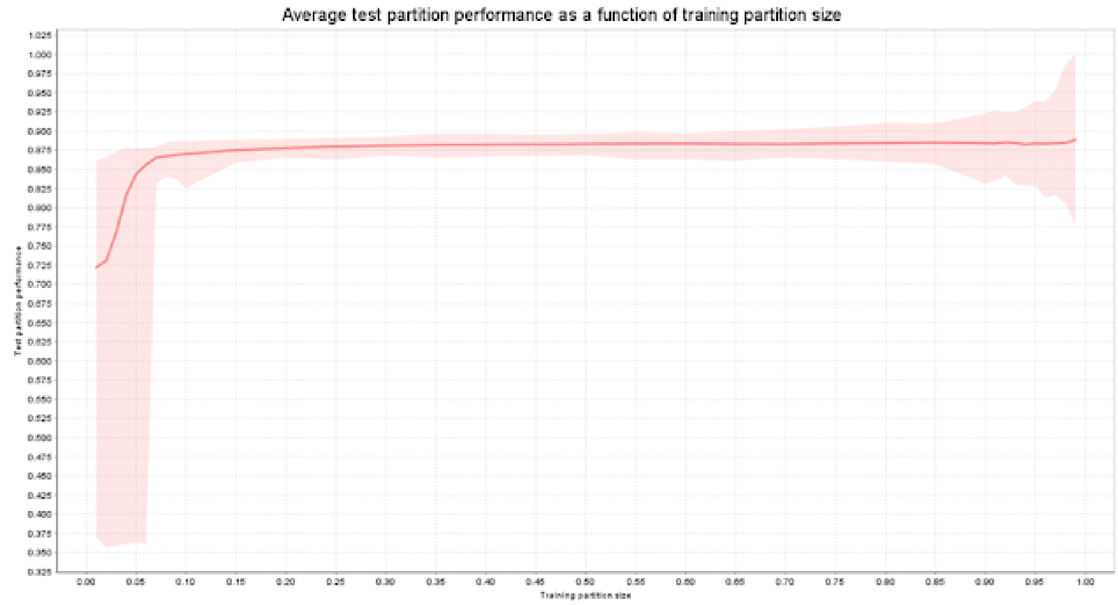


Figure 17: An evaluation of test train splits, showing saturation after a point in accuracy even if test data is increased. Link: http://information-gain.blogspot.com/2012/07/why-split-data-in-ratio-7030.html

7.2. The model was run using the .fit method

7.3. The model was successfully fit into the training data and when validated, gave the following metrics of performance.

```
from sklearn import metrics

print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
print("Precision:", metrics.precision_score(y_test, y_pred))
print("Recall:", metrics.recall_score(y_test, y_pred))

Accuracy: 0.9633989013428033
Precision: 0.45314505776636715
Recall: 0.06783243658724059
```

Figure 18: Validation of Logistic Regression

Overall, the model indicated good accuracy, but a limitation that I found with scikit learn is pertinent to mention here. Scikit learn doesn't provide methods to view significance of individual features, which is relevant to the interpretation being drawn. Significance of the features will be discussed is the resubmission of ISAS Iteration.

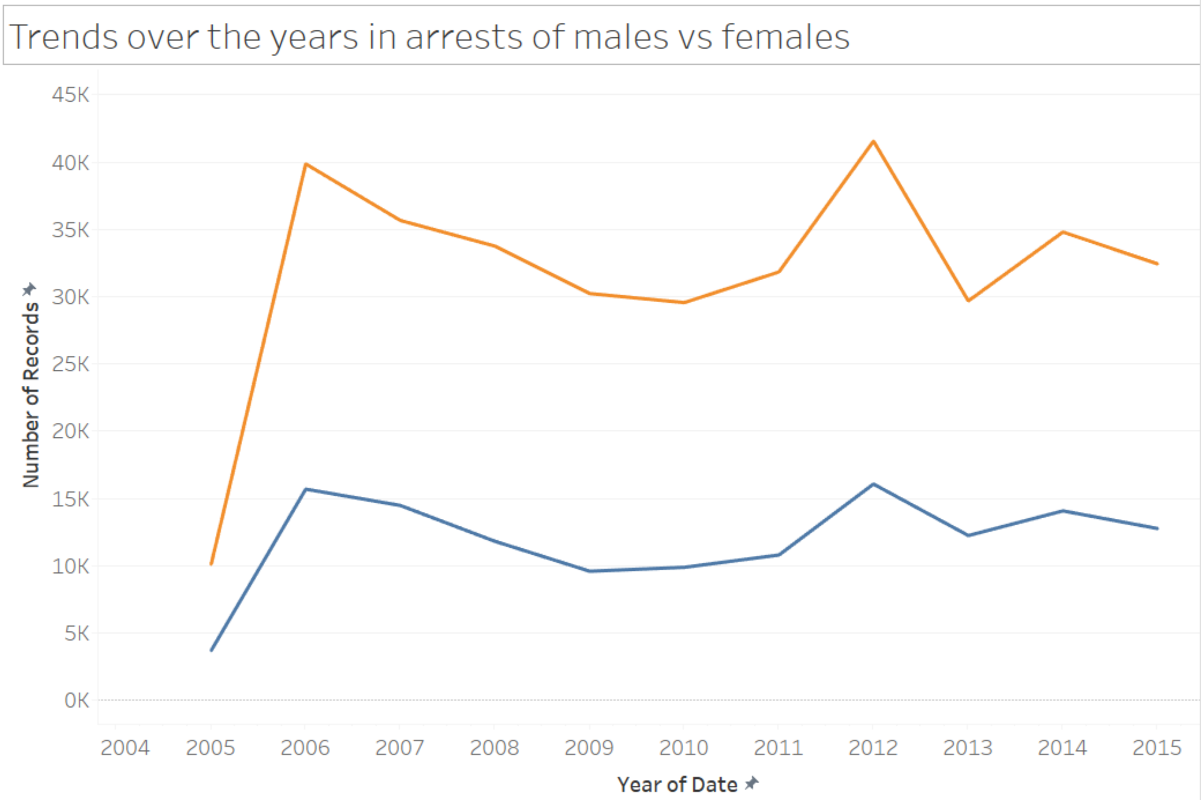
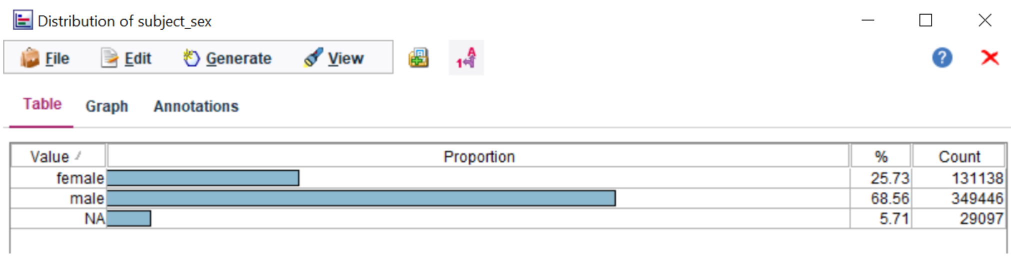


Figure 19: Arrest Trends over the years, with orange as female and blue as male, generated using Tableau. (see 8.1)

8. Interpretation:

8.1. In general, the plots made of the data show that there are possibilities that cops in Rhode Island are discriminatory when it comes to the gender of the person they are arresting. Visual data exploratory analysis suggested that the problem is huge, and that the cops are very much biased when it comes to males being searched.


Distribution of subject_sex

File Edit Generate View

Table Graph Annotations

Value	Proportion	%	Count
female		25.73	131138
male		68.56	349446
NA		5.71	29097

Figure 20: Distribution of subject_sex in the stops data.

8.2. The following visualisations from pandas can help better understand the situation. Visual data exploration done here helps in better judging the situation, and even though many of the visuals are not modelling related, they are critical to understanding the situation. Also, visualization through tables is as important as visualisation through graphs, and both have been done using

Python and Tableau.

```
In [11]: #Search rate for female and male drivers
         print('female = ' + str(ri[ri.subject_sex == 'female'].search_conducted.mean()))
         print('male = ' + str(ri[ri.subject_sex == 'male'].search_conducted.mean()))

female = 0.018751239152648355
male = 0.04379217389811301
```

Figure 21: Males are searched more often than females.

```
In [13]: #COMPARING SPEEDING OUTCOMES BY SEX

         #Create a DataFrame of female drivers stopped for speeding
         female_and_speeding = ri[(ri.subject_sex == 'female') & (ri.reason_for_stop == 'Speeding')]

         #Create a DataFrame of male drivers stopped for speeding
         male_and_speeding = ri[(ri.subject_sex == 'male') & (ri.reason_for_stop == 'Speeding')]

         #Compute the stop outcomes for female drivers (as proportions)
         print(female_and_speeding.outcome.value_counts(normalize=True))

         #Compute the stop outcomes for male drivers (as proportions)
         print(male_and_speeding.outcome.value_counts(normalize=True))

citation    0.954609
warning     0.039059
arrest      0.006332
Name: outcome, dtype: float64
citation    0.946763
warning     0.036167
arrest      0.017070
Name: outcome, dtype: float64
```

Figure 22: Females are more likely to be let off by a warning while males are more likely to be arrested for speeding.

```
In [12]: #Create a DataFrame of stops in which a search was conducted
         searched = ri[ri.search_conducted == True]

         #Calculate the overall frisk rate by taking the mean of 'frisk'
         print('mean of frisk performed = ' + str(searched.frisk_performed.mean()))

         #Calculate the frisk rate for each gender
         print(searched.groupby('subject_sex').frisk_performed.mean())

mean of frisk performed = 0.5248282851030289
subject_sex
female    0.437983
male      0.538783
Name: frisk_performed, dtype: float64
```

Figure 23: Males were frisked more often than females.

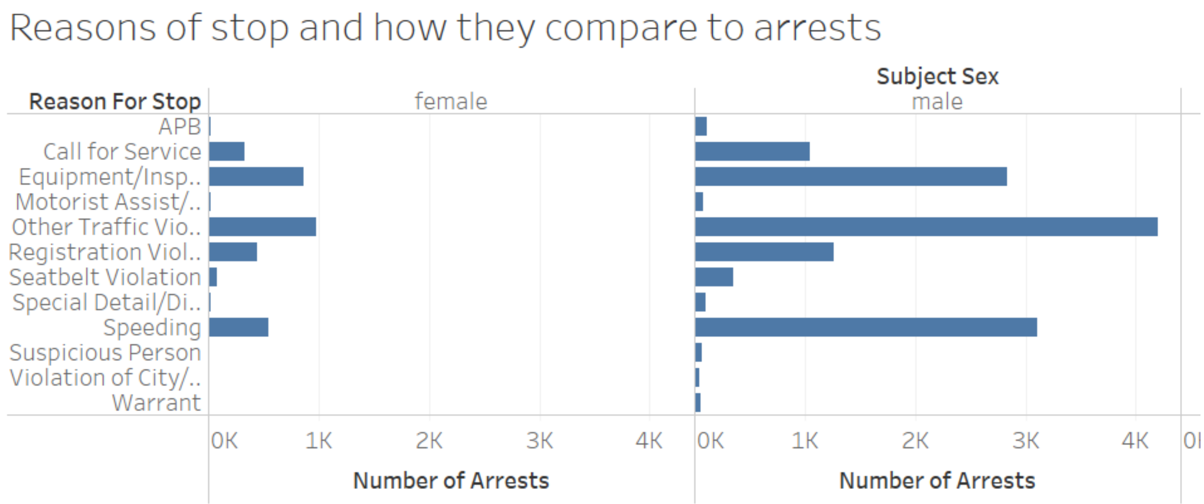


Figure 24: Gender comparion of the reasons for which people are stopped, using Tableau.

8.3. Evaluating on the results from ISAS, the model run with contraband_found set as the target variable yielded different results. A calculation of the odds ratio using numpy showed that the odds ratio for subject_sex was 1.89747451. Using the head() function on the dataset before fitting into the label encoder revealed that male was encoded to 1 while as female was encoded to 0. A positive relationship (>1) of the odds ratio revealed that as gender "increases" from 0 to 1 (based on our encoding, is male) the odds of contraband being found are 1.8 times than for females.

```
In [23]: #Check confidence values in equation of LR
         logreg.coef_

Out[23]: array([[-2.23173814e-04, -1.10745421e-02, -1.74829490e-01,
                  6.40523796e-01,  2.78508645e+00, -1.56715692e-01]])
```

```
In [24]: #Calculate the Odds Ratio
         import numpy as np
         np.exp(logreg.coef_)

Out[24]: array([[ 0.99977685,  0.98898655,  0.83960017,  1.89747451, 16.20121837,
                  0.85494709]])
```

Figure 25: Calculation of Odds Ratio

8.4. An analysis of Odds ratio of subject_sex = male 1.89747451. A value greater than 1 of the odds ratio indicated positive relationship, which meant that one was more likely to be arrested if one is male. The insight from this iteration is in stark opposition to what was found in the ISAS Iteration. **Whilst males are more likely to be arrested for the same features, searches in males more often reveal contraband than in females, justifying male searches.**

An analysis of significance values couldn't be done in this iteration and will be done in the resubmission of Iteration 2, so we can't comment on the confidence we have in the odds ratio being rightly interpret.

8.5. Multiple Iterations were done, as seen from ISAS Iteration for fine tuning the model results. Many of the limitations of the ISAS Iteration were hence tackled by doing a better test for discrimination, which revealed that the situation with the police for men is not as bad as it seems from initial exploration. Some limitations continue to exist, though. One, there is no comparison with a base population of Rhode Island. Even when compared with a base population, finding a dataset that includes information of male vs female drivers in Rhode Island would be difficult. The patterns found in the analysis of gender bias in policing, but just like all other discriminatory analysis, there is a limitation to what can be concluded from statistics alone!

9. Future Work

Data from a different city in the United States could be analysed to backup the insights found in these iterations.

10. Acknowledgement

I acknowledge that the submitted work is my own original work in accordance with the University of Auckland guidelines and policies on academic integrity and copyright.
(See: https://www.auckland.ac.nz/en/students/forms-policies-and-guidelines/student-policies-and-guidelines/academic-integrity-copyright.html).

I also acknowledge that I have appropriate permission to use the data that I have utilised in this project. (For example, if the data belongs to an organisation and the data has not been published in the public domain then the data must be approved by the rights holder.) This includes permission to upload the data file to Canvas. The University of Auckland bears no responsibility for the student's misuse of data.

11. References

Ed. (n.d.). There's Nothing Odd about the Odds Ratio: Interpreting Binary Logistic Regression. Retrieved from https://www.statisticssolutions.com/theres-nothing-odd-about-the-odds-ratio-interpreting-binary-logistic-regression/

Pierson, E., Simoiu, C., Overgoor, J., Corbett-Davies, S., Ramachandran, V., Phillips, C., & Goel, S. (2017). *A large-scale analysis of racial disparities in police stops across the United States*. 1–10. Retrieved from http://arxiv.org/abs/1706.05678

Sharad, Goel, C., & Philips. (2017). Police Data Suggests Black and Hispanic Drivers Are Searched More Often Than Whites. Retrieved from https://slate.com/technology/2017/06/statistical-analysis-of-data-from-20-states-suggests-evidence-of-racially-biased-policing.html

Corbett-Davies, S., Pierson, E., And, A. F., & Goel, S. (2016). A computer program used for bail and sentencing decisions was labeled biased against blacks. It's actually not that clear. Retrieved from https://www.washingtonpost.com/news/monkey-cage/wp/2016/10/17/can-an-algorithm-be-racist-our-analysis-is-more-cautious-than-propublicas/

Julia Angwin, Jeff Larson, S. M. and L. K. (2016). Machine Bias. Retrieved from https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing