Identification of Racial Bias in the stop-and-frisk program

Abhilash Nagaraja Ronald Pukadyil Ipseeta Deka

Department of Information & Decision Sciences

University of Illinois at Chicago

ABSTRACT

With the increased number of incidents reported across the United States revealing brutality shown by the police against minority groups, there lies a scope in conducting a thorough statistical study of the various factors leading to an arrest. With that intent, this project attempts to verify the existence of racial bias in the stop-and-frisk initiative from NYPD by harnessing the capabilities of various statistical techniques and Machine Learning algorithms to model the relationship between various key features leading to the arrest.

1 INTRODUCTION

Since its inception, the NYPD's stop-and-frisk program has been subjected to a great deal of criticism. Although the New York City Police Department claims the program to be an absolute necessity in today's world with a high crime rate, critics maintain that the practice often portrays brutality and violates civil rights.

The stop-and-frisk program is a practice advocated by the NYPD where civilians and suspects on the streets are temporarily detained, questioned, and at times frisked for possibility of finding weapons and other contraband. The police department has often credited the practice for the sharp decline in crime rates in New York city. However, the practice has often drawn attention for various controversies of racial profiling and has raised eyebrows of several civil rights activists. It has also prompted angry reactions from minority groups over the years. Often, has been debated the cost of detaining individuals on the street at the cost of preventing a possible crime.

The project attempts to study the NYPD stop-and-frisk data from the year 2015-16 with the intent of identifying whether there exists a significant relationship between race and the decision to arrest a stopped individual or not. Throughout the course of this paper we present the several statistical techniques conducted to explore the relationship between various factors recorded during the event of stop-and-frisk program and the target variable of interest – arrest made or not.

1.1 Data

The data is sourced from the open-source NYPD government website for the year of 2015-16. Data consists of 34,967 instances of stops resulting from the stop-and-frisk practice conducted by on-duty police officers in and around the limits of New York city. Information regarding every stop made is recorded across 100 different features. This feature set houses a wide variety of feature subsets capturing data regarding various elements. Locality based information like *time, street name, area code*, etc., crime-related features include *weapons carried, contraband found, summons issued, suspect frisked*, etc. The data also contains data elements describing the physical appearance of the suspect like *height, weight, build, hair color, age*, and *race*. The target variable of interest is *arrest made* which follows a binary distribution of Yes or No values.

Although the goal of identification of racial bias involves verifying the relationship between the variable *race* and the target variable, we intend to model the relationship of all the factors using various Machine Learning models.

An initial descriptive analysis of the data reveals a few key insights about the nature of the practice. **Figure 1** shows some initial insights derived from the data including the bar chart of the number of individuals across different races stopped because of the practice.

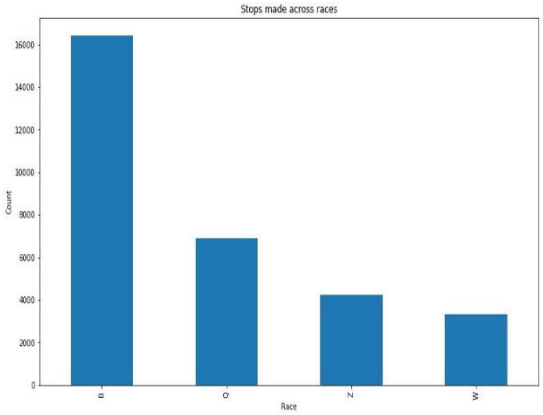


Figure 1: Number of individuals stopped across race

2 MODEL

We set off with a goal to design a classifier that takes into consideration the key features from the data to accurately predict the outcome of the stop-and-frisk event. In this regard, we attempt to design a mathematical model to map the underlying relationship between the features and the target variable. Some of the classification techniques we explore include probabilistic models like Naïve-Bayes classifier and instance-based learner like k-nearest neighbor algorithms.

We further aim to use some of the statistical techniques like t-tests, chi-square tests to verify the statistical significance of the relationship between features and the target variable. We finally try to utilize the interpretability of classification algorithms like the Logistic Regression model to derive the key driving factors of an arrest. Once the key factors are determined, we verify with the p-values of each of the features to determine their significance.

2.1 K-Nearest Neighbors algorithm

K-Nearest Neighbors is one of the most basic yet essential instant based classification algorithms in Machine Learning. We have chosen K-Nearest Neighbors as the baseline model for this project as it is non-parametric, which means it does not make any underlying assumptions about the distribution of data. The mathematical representation for KNN can be given as:

Predict $h(x') = \arg\max (\sum_i \in kNN(x')\ 1\ [y_{(i)} = y])$, where the training data is $D = \{(x^{(1),}\ y^{(1)}),\ldots, (x^{(m)}, y^{(m)})$. Here, the feature vectors are $x^{(i)} \in X$ and output scalar $y^{(i)} \in Y$. K-Nearest Neighbors uses a similarity function K:X x X -> R, where k is the number of nearest neighbors.

KNN works well with classification based on similarity. We have done one-hot encoding as most of the variables are categorical. With this scaled data modeling was done on the optimal value of K using grid search and minimum error.

Appendix A shows the pseudocode for the KNN model.

2.2 Bernoulli Naïve Bayes

A probabilistic model which functions on the underlying assumption of conditional independence among its features.

For classification we can use the Bayes' rule,

$$P(c \mid d) = \frac{P(c)P(d \mid c)}{P(d)},$$

where P(d) plays no role in selecting $c*$ and in estimating the term P(d|c).

With majority of the features in the data following a Bernoulli distribution (result of One-hot encoding of the features), we implemented a Bernoulli-Naïve Bayes model by calculating prior probabilities along with a 5-fold cross validation.

Appendix B shows the pseudo-code for our Bernoulli Naïve-Bayes model.

2.3 Logistic Regression

Logistic regression is one another function which is suitable for classification problems. It is a part of generalized linear model family and relies upon sigmoid link function which gives output in range of [0,1]

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}},$$

$$g(z) = \frac{1}{1 + e^{-z}}$$

g(z) is called **Sigmoid Function.** We can see that the continuous function is bound between 0 and 1. We can keep $x_0 = 1$ and overall notation will be as follows:

$$\theta^T x = \theta_0 + \sum_{j=1}^{n} \theta_j x_j$$

We implemented a logistic regression with L2 regularization along with a stochastic gradient descent training algorithm on the training data. Threshold values for predictions were manipulated to minimize the false positives in predictions on the validation set.

Appendix C shows the pseudo-code for our Logistic Regression model.

3 EXPERIMENTAL RESULTS

This section of the paper describes our initial attempts to understand the data and implementation of feature selection techniques. We then proceed to discuss the outcomes of the various machine learning models we implemented.

3.1 Data Description

The target feature of interest was a binary class with Y or N values. The initial data consisted of 34,967 instances with more than 6,000 arrests indicating a slight class imbalance. The features were mostly categorical with a few exceptions like height and weight. Some of the redundant features were eliminated based on intuition. To support our hypothesis, we conducted statistical tests to establish the statistical significance of the relationship of features with target.

To enhance our understanding of the data prior to engaging in building classifiers, we conducted a thorough descriptive analysis revealing some key insights about the data. For example, **Figure 2 & Figure 3** reveal that the number of arrests made and the number of frisks made was evenly distributed across individuals of all races. Yet from **Figure 1** we know that most of the individuals stopped were belonging to the Black community. To understand this further we proceed to model the features and design a classification algorithm.

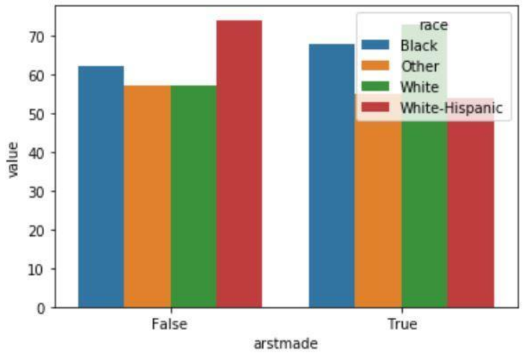


Figure 2: Arrest made across all races

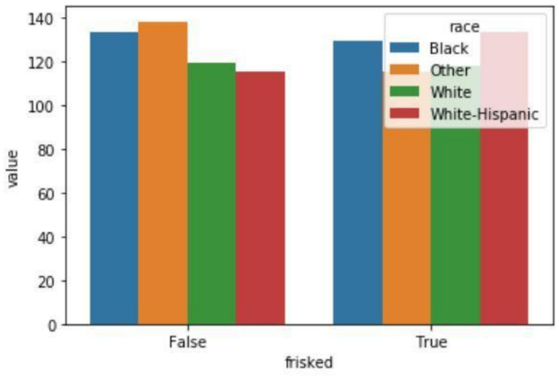


Figure 3: Frisks made across all the races

3.2 Feature Selection

Presence of redundancy among the features can be a curse for classification algorithms often leading to overfitting problems. It was crucial to eliminate any inherent correlation between the features in the data. The feature sets were divided into numerical and categorical features. Parametric t-tests were conducted on a set of numerical features on two independent samples of features (arrested 1/0) and non-parametric chi-square tests were conducted on all the categorical features. From the p-values of the respective sets close to 30% of the features indicating no significant statistical significance in their relationship with the target class were eliminated.

3.3 Model Design

All the classification techniques discussed in the previous section were designed and trained with 5-fold cross validation on the training data and predictions were made on a portion of data priorly reserved purely for validation. To compare the performance of each of these models, three performance evaluation metrics were used: Accuracy, Precision and AUC-PR curve. We further tuned the model parameters to minimize the number of False positives in our prediction. The arrest made classification problem is analogous to the spam classification setting. The cost of an innocent getting arrested is treated greater than the cost of a criminal not getting arrested.

Cost (FP) >> Cost (FN)

As we intend to reduce the false positives in prediction, the model evaluation parameters of our interest were Precision and Accuracy.

Using cross validation, the optimal k-value was chosen**. Figure 4** indicates the error values across different k-values resulted in the process of optimal selection of k-value for the k-nearest neighbor algorithm. The optimal value of k=20 was chosen with the least error rate on validation data.

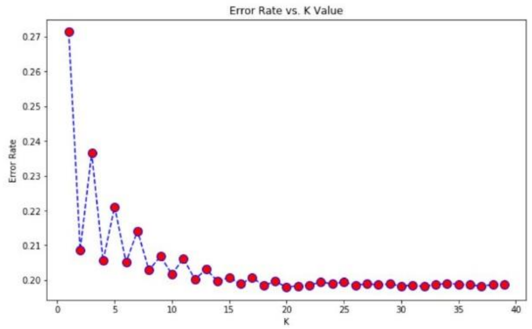


Figure 4: Optimizing the k-value by minimizing error rate

The baseline model (K- Nearest Neighbor) was found to have a test accuracy of 80.20%, precision of 51.76%, and an AUC of 0.6 in the ROC-curve (**Figure 5)**.

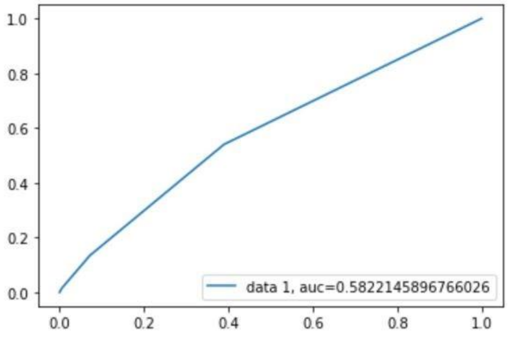


Figure 5: ROC curve for k-nn classifier

A Bernoulli Naïve Bayes designed on the Bernoulli distribution of all features with prior probabilities calculated. The model yielded an Accuracy of 86.50%, Precision of 64.30% on the test data with an AUC of 0.73 in the PR-curve. (**Figure 6**).

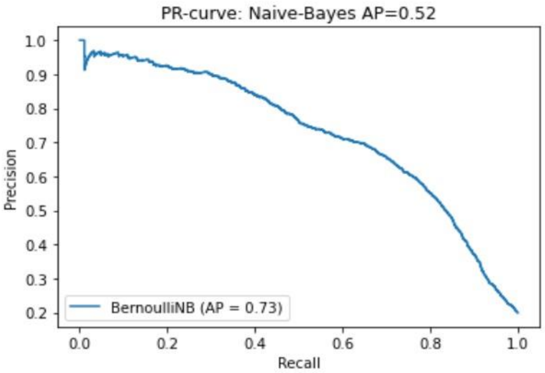


Figure 6: PR-curve for Naive Bayes

Our final classifier, Logistic Regression model performed better compared to the other two models. An Accuracy of about 88.4%, Precision: 74% and a 0.73 AUC value in the PR curve (**Figure 7**) was observed on test data.

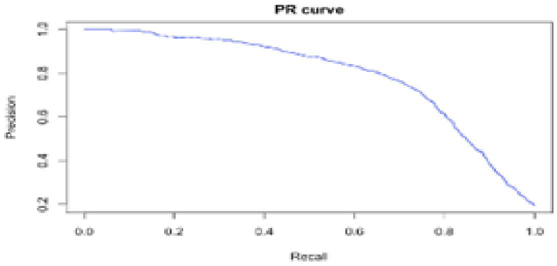


Figure 7: PR-curve for Logistic Regression

Later we performed hyper parameter tuning by using stochastic gradient descent as well as Ridge regularization with a learning rate of 0.01 which was obtained by using grid search. The model yielded an Accuracy of 89%, a Precision of 77% and an AUC of 0.78 in the PR curve. However, our focus was to increase the Precision, and the AUC scores with a decent increase in the accuracy. We further modified the prediction thresholds of the model to minimize the False positives in our predictions. On setting the threshold at 0.5 initially, we got an Accuracy of 89.96%, Precision of 78% and AUC of 0.79 in the PR-curve. With a further tuning of got the optimal threshold at 0.6 which improved our Precision to 85.10, with an Accuracy of 89.55 and an AUC-PR of 0.79. **Figure 8** shows performance and evaluation of different models.

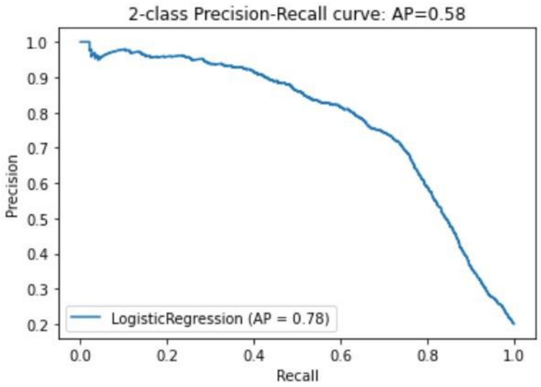


Figure 8: PR-curve L2-reg Logistic Regression with stochastic GD

3. 4 Model Evaluation:

Throughout our analysis we have focused on minimizing the False positives in our predictions and in that regard, our goal is to pick the model with the highest value of Precision on the test set.

The confusion matrix for each of the classifiers implemented indicating the predictions on the test set are given in *Table 1*, *Table 2*, and *Table 3*, respectively.

Actual Class	**Predicted Class**		**Performance measures**	
	N	**Y**	**Precision**	0.5176
N	7422	1830	**Accuracy**	0.8020
Y	9	8	**PR-AUC**	0.58

Table 1: Confusion Matrix for k-NN classifier

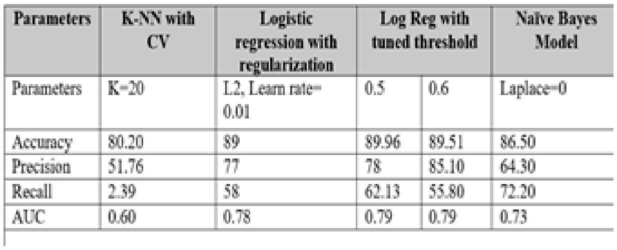
Actual Class	**Predicted Class**		**Performance measures**	
	N	**Y**	**Precision**	0.6430
N	6687	744	**Accuracy**	0.8650
Y	518	1320	**PR-AUC**	0.73

Table 2: Confusion Matrix for Bernoulli Naïve-Bayes

Actual Class	**Predicted Class**		**Performance measures**	
	N	**Y**	**Precision**	0.8510
N	7292	176	**Accuracy**	0.8955
Y	796	1005	**PR-AUC**	0.79

Table 3: Confusion Matrix for L2-regularised Logistic Regression with 0.6 as threshold

We put together the evaluation metrics of all the classifiers in the Table 4. From the table, we can see that the L2-regularized Logistic Regression, trained with a learning rate of 0.01 with a prediction threshold value of 0.6 gives the highest precision value. We decide to choose this variant of Logistic Regression to be our best model.



Parameters	K-NN with CV	Logistic regression with regularization	Log Reg with tuned threshold		Naïve Bayes Model
Parameters	K=20	L2, Learn rate= 0.01	0.5	0.6	Laplace=0
Accuracy	80.20	89	89.96	89.51	86.50
Precision	51.76	77	78	85.10	64.30
Recall	2.39	58	62.13	55.80	72.20
AUC	0.60	0.78	0.79	0.79	0.73

Table 4: Feature Importance given by Logistic Regression model

3.5 Feature Importance

One of the advantages of our chosen model of L2-regularized Logistic Regression is its ability to describe the statistical significance of the relationships between all the features and the target variable. We leverage this interpretability of the classifier to determine key driving features leading to an arrest in the event of stop-and-frisk.

Table 5 lists the top 6 features with their corresponding co-efficient values determined from the log-odds of the probabilities resulting from the chosen variant of the Logistic Regression model. The table also includes p-values indicating the significance of the relationship each of the features carry with the target variable.

Features	Co-efficient value	p-value
weaponsY	2.220e+00	<2e-16
searchedY	1.895e+00	<2e-16
contrabandY	3.073e+00	<2e-16
friskedY	-1.820e-01	0.002469
sb_hdbjY	-7.24e-01	4.54e-11
sumissueY	-3.623e+00	<2e-16

Table 5: Top 6 features leading to arrest as given by our best version of Logistic Regression model

Table 5 suggests that the key factors that help in deciding whether to make an arrest or not are: whether the suspect was carrying a weapon, contraband, or any other hidden object, whether a summons was issued in his/her arrest, and whether the person was frisked or searched.

To back our results, the significant factors learned by our model are the same as the ones suggested by statistical tests (previously mentioned in section 3.2). Above argument rules out the significance of any factor that describes the physical aspects of a suspect like race, age, build, etc.

4 CONCLUSIONS

Determining a list of key predictors of arrest with the aid of our best classifier was key in establishing the insignificance of Race as a key factor in decision to arrest. However, from the initial exploration of the data (Figure 1) we found that the people from Black community were stopped more often than individuals of other race raising the question about racial profiling. Our analysis revealed that there is no reason to stop and frisk individuals based on their race as the determining factors of arrest were related to presence of suspicious weapons, contrabands, etc.

Although, we have conducted thorough analysis to determine that race is not a key factor in arrests, our analysis is highly limited to the NYPD stop-and-frisk data. In the future scope of the project, we intend to provide more conclusive evidence to our results by analyzing the data of similar programs across various cities in the United States or by extending our search beyond the years of 2015-16. We also intend to implement other machine learning techniques like Random Forrest and Ada Boost models to harness the power of mathematical interpretability of Information theory underlying the tree-based models.

Appendix A.

This appendix shows the pseudo-code used for the design of our K-Nearest Neighbor model:

```
Classify **(X, Y,** x) // **X**: training data, **Y**: class labels of X, x: test sample

for i=1 to n do
        Calculate distance $d(X_i)$
End for
        Calculate set i containing indices for the k-smallest distances $d (X_i, x)$
**return** majority label for $\{Y_i$ where $i \epsilon I\}$
```

Appendix B.

This appendix shows the pseudo-code used for the design of our Naïve-Bayes model:

Given X, the set of n features following a Bernoulli Distribution **X =** $\{X_1, X_2, X_3, ..., X_n\}$

and **Y** is our target variable '*arstmade'* (with 1-Yes and 0-No values)

- We calculate the prior probability of the classes $P(Y=1)$ and $P(Y=0)$

  $P(Y=1)$ = sum [i to m $1(y=1)$] / m

  $P(Y=0) = 1 - P(Y=1)$

- We compute the conditional probabilities of each of the features w.r.t the target

- **For (x_i in X):**

  Compute prior probability
    - P (x_i | y_i=1) = Sum i to m 1{x_i=1, y_i=1} / sum i to m 1{y_i=1}
    - P (x_i | y_i=0) = Sum i to m 1{x_i, y=0} / sum i to m 1{y_i=0}
- Predict P (y=1 | x_i) = P (x_i | y_i=1)

P(y_i=1) / P(x_i)

- If **probability predicted > 0.5** Y=1

  Else Y=0

Appendix C.

This appendix shows the pseudo-code used for the design of our Logistic Regression model:

Step 1:

Set the following:
Acceptable threshold for Cost Function: €
Maximum number of Epochs, max_iter
Number of Epochs, m
Initialization of Augmented Weight Matrix, $\theta = 1$ Initialization of Cost Function, $J(\theta) = 0$

Step 2:

Select mapping functions for the dependent features

Step3:

Updating the augmented weights matrix θ with $\theta j(n) = \theta j(n-1_{)} + \alpha.\ Vj$

Step4:

Find the Cost Function or Average Cost $J(\theta)$ using $J(\theta) = J(\theta) + (-1\ /m\ )\ (\sum^{m}_{i=1}\ y_{(i)} \log (h_{\theta}(x_{(i)})) + (1 - y_{(i)}) \log(1 - h_{\theta}(x_{(i)})))$

Step5:

If $|J_{(\theta)}|$ < or = € (or) m = max_iter Goto Step 6
Else Goto Step 3 and update the Augmented Weight Matrix θ

Step6:

We get the optimum weight for matrix θ

Testing:

Predict the output on the test set

REFERENCES:

1. Www1.nyc.gov. 2020. *Publications, Reports - NYPD.* [online] Available at: <https://www1.nyc.gov/site/nypd/stats/reports-analysis/stopfrisk.page> [Accessed 8 December 2020].
2. JohnM. (2020, November 02). Police Violence & Racial Equity - Part 2 of 3. Retrieved December 08, 2020, from https://www.kaggle.com/jpmiller/police-violence-racial-equity
3. Ammirati, S. (2016, December 04). NYPD Stop and Frisk -- Full Code. Retrieved December 08, 2020, from https://statsworks.info/category/projects/nypd_stop_frisk_full
4. Brownlee, J. (2019, December 11). How To Implement Logistic Regression From Scratch in Python. Retrieved December 08, 2020, from https://machinelearningmastery.com/implement-logistic-regression-stochastic-gradient-descent-scratch-python
5. Kutner, D. (2020, March 17). Stop and Frisk - data visualization and analysis. Retrieved December 08, 2020, from https://medium.com/@daniellekutner/stop-and-frisk-data-visualization-and-analysis-504c9a41ab6c
6. Spitzer, E. (1999), "The New York City Police Department's "Stop and Frisk" Practices," Office of the New York State Attorney General; available at www.oag.state.ny.us/press/reports/stop_frisk/stop_frisk.html
7. Derek A. Epp, Macey Erhardt. (2020) The use and effectiveness of investigative police stops. Politics, Groups and Identities 0:0, pages 1-14
8. Akinola, S. and Oyabugbe, O. (2015) Accuracies and Training Times of Data Mining Classification Algorithms: An Empirical Comparative Study. Journal of Software Engineering and Applications, 8, 470-477
9. Epp, Charles R., Steven Maynard-Moody and Donald P. Haider-Markel. 2014. Pulled Over: How Police Stops Define Race and Citizenship. Chicago: University Chicago Press