Machine Learning Analysis on the Fairness of the Boston Police Department Stop and Frisk Practices

Dheekshita Kumar, Emily Tang, Mary Zhong
{dhkumar, emitang, mzhong}@mit.edu
December 6, 2018

Table of Contents

1 Executive Summary

Fairness in the decisions made by police officers, specifically in stop and frisk encounters, has been a contentious topic in the recent history of the United States. Public pressure pushing for transparency behind the actions of police departments has caused them to release data, such as the New York Police Department. Many organizations have begun to analyze this data, such as the New York Civil Liberties Union[1]. In this paper, we use machine learning to analyze the fairness of the Boston Police stop and frisk practices. We used the Boston Police Department Field Interrogation and Observation (FIO) Data, which provides over 150,000 records of stop and frisk encounters from 2011 to 2015.

In order to make conclusions about fairness in stop and frisk related data, we need to define what fairness is. Deciding what it means to be fair has always been a difficult task, and it is particularly difficult in situations related to the police and stop and frisk. In this paper, we discuss 5 different definitions of fairness: Group Unaware, Group Aware, Demographic Parity, Equal opportunity, and Equal Access. To determine which definitions of fairness to use in our machine learning analysis, we examine past legal cases regarding stop and frisk encounters to determine which definitions of fairness were used to evaluate those situations. We concluded that the definitions of fairness that we should use were Group Unaware, Demographic Parity, and Equal Opportunity because these were the definitions that aligned best with the definitions of fairness set by legal precedence.

We use machine learning based analysis. Specifically, we train machine learning models to model the decision making process of Boston police officers. We analyze these models based on different features given in the data, such as race and age. Oftentimes, statistical analysis is used to claim bias in a decision making process, especially stop and frisk practices. While a statistical analysis can show interesting tendencies in the outcomes of a process, it is not a tool powerful enough to provide meaningful insights into a *decision making* process. This is because statistical analysis rarely considers multiple features and how they contribute to an overall result, and does not produce a model of the decision making process.

For each of the definition of fairness we developed a separate method of analysis.

For a dataset to be Group Unaware fair with respect to a particular feature, it means that the feature should not matter in the final decision. For example, if our data set is group

[1] NYCLU. “Stop-and-frisk Data.” https://www.nyclu.org/en/stop-and-frisk-data.

unaware fair with respect to race, then that would mean that race is not a feature the police officer takes into account when deciding frisk. To analyze the data with respect to this definition, we trained a model that did not account for a particular feature. We then used that model to classify the data, and compared how that classification matched the real life results.

For a dataset to be fair with respect to Demographic parity, it means that if 20% of the original population had a particular feature, then 20% of the frisked population must also have that feature. To analyze the data with regards to this definition, we statistically compared how the demographic breakdown of the stopped population matched the demographic breakdown of the frisked population for various features (race, sex, district, etc.)

For a dataset to be fair with respect to Equal Opportunity, it means that across variations in one feature, all else being the same, people have the same chance of being frisked. In other words, a low risk Asian man and a low risk white man should have the same chance of being frisked if all other qualities (aside from their race) are the same.

Using this analysis, we were able to show that officer ID, sex, age, and race play significant roles in a Boston police officers' decision to search and frisk someone. In particular, we found that certain races (Asians and Blacks) tend to have more chance of being frisked (according to the equal opportunity definition of fairness), while Hispanics are frisked more than would be expected given the percentage of Hispanics in the original population (i.e. Demographic parity view of fairness). We also found that with regards to age and sex, men and younger people were generally more likely to be frisked. The younger a person was, the more chance they had of being frisked from an equal opportunity point of view, and the more they were frisked than expected according to the demographic parity fairness point of view. There were similar results if the person was a male.

We hope that our results provide a foundation for future machine learning based analysis on stop and frisk data. In legal settings, we also hope that our analysis and results are an example of how to provide a way to show quantitatively why a certain stop and frisk action by an officer may or may not be fair.

2 What is “Stop and Frisk”?

In order to understand the large amount of BPD FIO data that was released, and in order to decide what type of machine learning analysis to perform, we needed a knowledge of what stop and frisk is, its history, and why it’s important.

Stop and frisk was first established in 1968 by a case called Terry v. Ohio[2], in which the Supreme Court ruled that the Fourth Amendment allows law enforcement officers to stop, detain, and frisk people on the sidewalk using the legal standard of reasonable suspicion. Many argue that certain minorities are unfairly targeted by stop and frisk, and are therefore at higher risk of being accidentally or purposely killed during such a stop. On the other hand, many others argue that the process is fair by some definition. By examining these incidents through a computational lens, we can introduce a degree of impartiality in the analysis.

In order to analyze data related to stop and frisk, we explore what qualifies a situation as a stop and frisk situation to gain background understanding. To do this, we go through a brief history in the development of stop and frisk in the eyes of the nation and the law to provide background and context.

In the original Terry v. Ohio case, a police officer stopped and patted down three men that he suspected of examining a storefront and strategizing a future robbery. Upon patting down the suspects, the police officer discovered two guns and arrested two of the men on the grounds of illegal possession of firearms. One of the suspects then sued, on the grounds that the police officer had violated the Fourth Amendment with his search, and therefore the grounds for arrest was inadmissible. The Supreme Court’s ruling set the grounds for police departments, such as New York City’s, to implement more comprehensive policing strategies based in Stop and Frisk on the grounds of reasonable suspicion, and the stops became known as *Terry Stops*.

Policing then became more strict and focused in the 1990s due to changes in personnel and new advances in technology. A number of key figures in New York, such as George L. Kelling, the Chief of the New York Transit Police, William Bratton, the commissioner of the NYPD, and Mayor Rudy Giuliani all began implementing tougher enforcement policies in

[2] Terry v. Ohio, 392 U.S. 1 (1976).

the 1990s. This time period gave rise to the idea of "broken windows policing"[3], which is the idea that visible disorder and low level crime gives way to larger scale crime. During this same time period, CompStat[45], or computer comparison statistics, also saw implementation in police forces beginning in New York City. CompStat is a computer management system that is used to increase efficiency in the department and allow for better policing.

These changes laid the groundwork for the friction and divisiveness that have come to dominate the national dialogue in the last few years. Broken windows policing and CompStat were regarded as policies that eroded trust with the neighborhoods officers were supposed to be policing. Furthermore, with CompStat's quantitatively driven performance metrics and high degree of personal accountability, many police officers felt mounting pressure to deliver more results and better statistics, which can come at the cost of responsible policing[6]. The case of Alabama v. White[7] in 1990 also opened the door for more liberal use of Terry stops. In the case, the police responded to an anonymous tip about possession of cocaine, and made an arrest on that basis after tailing, stopping, and searching the suspect. When the case made it to the Supreme Court, the court ruled that while the tip was not sufficient grounds for a warrant, it constituted enough reasonable suspicion for a Terry stop, further increasing the range of situations a Terry stop could be made in. These changes therefore led to Terry stops being used more frequently in order to prevent crime before its onset and to boost police officer's performance metrics as documented by CompStat.

A lot of this issue revolves around two key words: reasonable suspicion, for which we are seeking to find a computational metric. What counts as reasonable suspicion and grounds for a Terry stop in the eyes of the law? As mentioned previously, the Alabama v. White case determined that an anonymous tip was sufficient. When the range of situations allowing a Terry stop is spread so wide, what could end up being the deciding factor is each police officer's individual intuition. This can at times be problematic, as demonstrated in 2010, when The Village Voice published recordings made by a former police officer,

---

[3] Center for Evidence-Based Crime Policy. "Broken Windows Policing." https://cebcp.org/evidence-based-policing/what-works-in-policing/research-evidence-review/broken-windows-policing/.

[4] SFPD. "Compstat." http://sanfranciscopolice.org/compstat.

[5] National Police Foundation. "Compstat and Organizational Change: A National Assessment." https://www.policefoundation.org/projects/compstat-and-organizational-change-a-national-assessment/.

[6] Willis, James and Stephen Mastrofski. "Compstat and Community Policing: Are They Compatible?" https://cops.usdoj.gov/pdf/workshops/thursday/WillisMastrofski.pdf.

[7] Alabama v. White, 496 U.S. 325 (1990).

Adrian Schoolcraft[8], that revealed that police officers had been given orders that were discriminatory against black residents. In 2012, a lawsuit was filed by several organizations including the New York Civil Liberties Union[9] against stop and frisk, and a video released by The Nation[10] revealed the extent of racially motivated stops by the police.

A major change to the nature of Terry stops came in 2013, when the U.S. Second Circuit Court of Appeals ruled in Floyd v. City of New York[11] that the New York City Police Department was responsible for a pattern of racial profiling and unwarranted stops. In Floyd v. New York, the prosecution found that 85% of those stopped by the police were Black or Latino, even though those racial groups only made up 52% of the city population. The court ruled that this was in violation of the Fourteenth Amendment, which guarantees equal protection under the Constitution. In a mayoral race later that year that revolved largely around Stop and Frisk, Bill de Blasio won, and has since taken major steps to reform stop and frisk. Studies by ProPublica[12] have found that with the decrease in Terry stops has come a further decrease in crime, which may indicate that Terry stops have not been the most accurate policing policy in terms of locating and effectively handling crime. Furthermore, while conditions have improved, some neighborhoods have still seen little change and others will still require more time to rebuild the trust eroded by stop and frisk[13].

More recently, due to public pressure, police departments around the country have been pushed to release data on stop and frisk situations. Our paper deals with one such instance, the BPD Field Interrogation and Observation data. While much purely statistical analysis has been performed on this set of data, and other similar data sets, we perform machine learning based analysis to give quantitative results about the actual decision making process of Boston police officers.

---

[8] Rayman, Graham. "NYPD Tapes 4: The Whistleblower, Adrian Schoolcraft." *The Village Voice*. https://www.villagevoice.com/2010/06/15/nypd-tapes-4-the-whistleblower-adrian-schoolcraft/.
[9] NYCLU. "Class Action Lawsuit Challenges NYPD Patrols of Private Apartment Buildings." https://www.nyclu.org/en/press-releases/class-action-lawsuit-challenges-nypd-patrols-private-apartment-buildings.
[10] Tuttle, Ross and Erin Schneider. "Stopped-and-Frisked." https://www.thenation.com/article/stopped-and-frisked-being-fking-mutt-video/.
[11] Floyd v. City of New York, 959 F. Supp. 2d 540 (2013).
[12] Sexton, Joe. "In New York, Crime Falls Along With Police Stops." *ProPublica*. https://www.propublica.org/article/in-new-york-crime-falls-along-with-police-stops.
[13] Wofford, Taylor. "Did Bill de Blasio keep his promise to reform stop-and-frisk?" *Newsweek*. https://www.newsweek.com/did-bill-de-blasio-keep-his-promise-reform-stop-and-frisk-266310.

3 Fairness in Legal Cases

3.1 What is fairness and how do we define it?

Defining fairness is a major concern in the AI and machine learning community[14]. There is much research dedicated to the topic because differing definitions of fairness can lead to very different results. In fact, it is even possible for definitions of fairness to contradict each other. Thus it is very important before we use machine learning to assess fairness, that we clearly define "what is fair" (i.e. our fairness criteria).

We are planning on using 5 definitions of fairness[15] to evaluate our results against: Group Unaware, Group Thresholds, Demographic Parity, Equal opportunity, and Equal Accuracy.

1. In the **Group Unaware** view of fairness, a stop and frisk decision making process is fair with regards to race if it does not take race into account at all.

2. In the **Group Thresholds** view of fairness, historical biases reflected in the data are accounted for in making a decision making model. For example, in the case of race, certain racial groups of people maybe more likely to come into contact with drugs, weapons, and the police themselves. This, however, should be accounted for in any decision making process.

3. In the **Demographic Parity** view of fairness, the demographics of people who are ultimately searched or frisked after being stopped should be proportional the demographics of the people who were stopped. In other words, if 60% of the people stopped were men, then 60% of the people who were ultimately frisked or searched should also be men.

4. In the **Equal Opportunity** view of fairness, with respect to age, the same percentage of people across different age groups, who are likely to be innocent, are not frisked or searched. In other words, a decision making process would not be fair if 90% of middle aged persons who are low risk at being offenders are left alone, but only 40% of teenagers who are low risk at being offenders are left alone.

[14] See the ACM Conference on Fairness, Accountability, and Transparency https://fatconference.org/
[15] https://pair-code.github.io/what-if-tool/ai-fairness.html

5. In the **Equal Accuracy** view of fairness, which is similar to the Equal Opportunity view, we look at how often the decision made was incorrect. A decision making process is fair if the number of times the officer is wrong about a certain decision and the number of times a officer is right about a certain decision is uniform across the different races/ages/genders/etc..

Note here that both Demographic Parity and Equal Accuracy definitions of fairness are based on how fair a past decision was, whereas the Group Unaware definition, Group Threshold definition, and Equal Opportunity definition are based on "potential" decisions. Existing statistical analysis techniques can be used, therefore, in the case of analyzing the fairness of past decisions.

Thus, we will see later, in section 4, that when dealing with definition of fairness such as Demographic Parity, we will use statistical analysis to directly analyze the fairness of the situation. When dealing with "potential" decision fairness definitions, such as Equal Opportunity and Group Unaware, we will use machine learning to create a model, then assess the fairness of the model, and analogize the fairness of the model to fairness in reality.

3.2 Which definition(s) of fairness does the law follow?

Before running a machine learning analysis, however, it's important to see which of these definitions the law and the courts have aligned with in their rulings.

The best example comes from the Supreme Court's ruling in Floyd v. City of New York, where the Supreme Court ruled that the New York Police Department was in fact unfairly targeting minorities and employing discriminatory practices. Following the court's ruling, a court monitor was appointed to file annual reports by federal mandate to make sure progress was being made. Some of the results of the monitor analysis were[16]:

- Hispanic citizens were more likely to be searched and arrested following stops
- Black citizens were less likely to be found with guns than white citizens

These statistics fall under the third and fourth definitions of fairness, Demographic Parity and Equal Opportunity. The fact that hispanic citizens were more likely to face further action after being stopped breaks with the Demographic Parity view that an equal proportion of those stopped should face further action across demographics.

[16] Arnold & Porter Kaye Scholer, LLP. "Analysis of NYPD Stops, 2013-2015". https://www.nytimes.com/interactive/2017/05/30/nyregion/nypd-stop-and-frisk-report.html

The fact that black citizens were less likely to be found with guns than white citizens breaks with the Equal Opportunity view that the same percentage of people who are likely to be innocent across demographics are searched (i.e. more black citizens are being searched but turning out to not have guns). In addition, we could argue that the statistics fall under the Group Unaware definition of fairness, since it was clear from the analysis that the stop and frisk practices of the NYPD definitely took race into account.

A study conducted by Columbia University in New York[17] reached similar conclusions. The study found that minority groups were more likely to be stopped than white citizens, both compared to the groups' overall percentage of the population and the actual rate of crime committed. This ties in once again to the third and fourth definitions.

In the city of Boston, whose data we examined for this paper, while there was not a formal court case, the American Civil Liberties Union[18] has run data analysis and made the following observation: even after controlling for crime rates, black neighborhoods are more frequently policed, and the residents there are stopped more frequently. While hitting the third and fourth definitions, this also dips into the first or second definition of fairness, the Group Unaware and the Group Thresholds view. Without knowing the motivations of the officers, it is impossible to tell which definition is more apt—all the report accounts for is that even after controlling for other mitigating factors, black residents of Boston are stopped more often, with fewer stops leading to actual arrests or discovery of contraband.

As definition two of fairness have to do with subtler things such as different histories across different places, it seems reasonable to focus on definitions one, three, and four: Group Unaware, Demographic Parity, and Equal Opportunity. Statistically, the Demographic Parity and Equal Opportunity views are valuable for analyzing where unfair practices may have occurred before, and what numbers and data should be targeted for change for the future. Our methods of analysis will match these these chose definitions of fairness. In this way, we can provide analytical results that can support arguments made regarding these three types of fairness, especially in legal cases.

---

[17] Gelman, Andrew, Jeffrey Fagan, and Alex Kiss. "An Analysis of the New York City Police Department's "Stop-and-Frisk" Policy in the Context of Claims of Racial Bias." Journal of the American Statistical Association, September 2007, 813-23.
http://www.stat.columbia.edu/~gelman/research/published/frisk9.pdf.

[18] ACLU of Massachusetts. "Ending Racist Stop and Frisk".
https://www.aclum.org/en/ending-racist-stop-and-frisk

4 Methods: Building a Machine Learning Model

Machine learning (ML) is an excellent tool for talking about fairness and discrimination because ML is used to model decision making processes. In our case, the decision making process we wish to model, and then assess, is the decision making process a police officer undergoes before deciding to search or frisk someone who has been stopped. There are also several frameworks for determining fairness in these decision-making algorithms, and these same frameworks can be used to talk about fairness in real world decision-making. If one has a model that quantitatively captures a decision making process, and then that model is determined to be unfair, it suggests the real-world decision making process is also unfair.

4.1 What is Machine Learning? How Does it Work?

At a high level, machine learning is the science of getting computers to learn the same way humans do. The idea is to feed data into a model, and have the model use this information to iteratively improve their learning over time.

For the purposes of our analysis, we will be using a type of machine learning algorithm model known as a *classifier*, and specifically, a *binary classifier*. The binary aspect of the model we are using is the decision of whether an individual will be frisked or searched.

Finally, when we talk about the data that a classifier uses, we distinguish between a *training* dataset and a *test* dataset. The *training* dataset is the set of data that is initially fed into the machine learning model. The model takes this data and determines how much each data feature affects the binary outcome (whether or not someone is frisked or searched). After this training period is complete, we now have a model that we can use on the *test* data. When test data is fed into the model, the model outputs 0 or 1, and, in our specific case, classifies the dataset into two groups: frisked or searched vs. not frisked and not searched.

For one of our methods of analysis, we chose a random sample of 120,000 stop and frisk records as the training dataset for building the model. We then used the model with the remaining 330,320 records as the test data for gaining insights into the fairness produced

by the model—and by assumption, the farness of the stop and frisk behavior being modeled.[19]

4.2 The Boston Police Stop and Frisk Data

4.2.1 Understanding and Modifying the Data

In January 2016, the Boston Police Department released data from the Field Interrogation and Observation (FIO) program from 2011 to 2015[20], in response to an ACLU public records request from over a year ago. This FIO data contains 150,320 records of stop and frisk encounters by the BPD from 2011-2015. The data released include information about four types of FIO actions: observe (O), interrogate (I), frisk (F), and search (S). Along with the type of FIO action taken, each data entry in the table includes identifiers for the individual affected such as race, sex, age, location, officer ID of the officer, FIO reasons, and so on.

In order to use the BPD FIO dataset to train a machine learning model, we needed to modify the data. In addition, we did not use every column of information in the original dataset. Table 1 has the columns that used in the training data set and the values that resulted from our modifications.

Each record in the modified data set has the following features (Table 1):

Table 1: Summary of Features in Boston Police FIO Records

Column Name	Possible Values
SEX	FEMALE, MALE, BLANK
FIO_DATE	<mm>/<dd>/<yyyy> 12:00:00 AM[21]
PRIORS	YES, NO, BLANK

[19] This assumption is not necessarily valid, because our model might not be a good model of the behavior. One extension to this work would be to repeat the analysis with a variety of machine learning models to test the validity of their conclusions.
[20] Gaffin, Adam. "BPD releases data on people officers talk to long enough to warrant a report." *Universal Hub*.
https://www.universalhub.com/2016/bpd-releases-data-people-officers-talk-long-enough.
[21] All rows have a time of 12:00:00 AM.

COMPLEXION	Brown, Clear, Dark, Fair, Light, Med, NO DATA ENTERED, OTHER, Ruddy, White
FIOFS_REASONS	Reasons including, INVESTIGATE, ALCOHOL, TRESPASSING, DISTURBANCE, DRUGS, etc.
AGE_AT_FIO_CORRECTED	TEENS (less than 19), TWENTIES (20 to 29), THIRTIES (30 to 39), MIDDLE (40 to 59), SENIOR (60 or older)
DESCRIPTION	A(Asian or Pacific Islander), B(Black), H(Hispanic), I(American Indian or Alaskan Native), M(Middle Eastern or East Indian), NO DATA ENTERED, UNKNOWN, W(White)
DIST	Boston area district codes (such as B2, D4, A7, etc.)[22]
OFFICER_ID	officer <ID number>
FIOFS_TYPE	***A combination of I, O, F, and S***

The data in bold is the data column that we ultimately used as our label column, the column our machine learning model is trying to predict.

To perform this data modification, we wrote a script that takes in the original dataset file and parses through it, taking only the columns we wanted to use and changing values from the data in those columns as needed. The script used can be found in Appendix B. Examples of some rows of modified data can be found in Appendix D.

4.2.2 Preparing the Data for Use in Model

In order to create a model, we had to obtain the list of features for each record in the data. For each feature, we also had to find the label. This results, therefore, in 4 components: Feature

[22] "Districts." *bpdnews*. http://bpdnews.com/districts/.

1. Training Data
    a. Training Data: Features
    b. Training Data: Labels
2. Test Data
    a. Test Data: Features
    b. Test Data: Labels

For the "labels", we used the value from the FIOFS_TYPE column. If the column had a value containing S or F, we classified it as 1. If the column did not contain those values, we classified it as 0. In other words, our labels told us whether or not the person was searched or frisked (See Table 2).

Table 2: Summary of the Binary Classification Condition

Condition: FIOFS_Type	Classification
Either Searched (S) or Frisked (F), or both	1
Neither Searched or Frisked (a combination of I or O)	0

The "feature" components were all the other values: SEX, FIO_DATE, PRIORS, COMPLEXION, FIOFS_REASONS, AGE_AT_FIO_CORRECTED, DESCRIPTION, DIST, and OFFICER_ID.

To split our modified BPD data into training and test data, we took a random subset of 120,000 rows (each row is one stop and frisk encounter with an individual) for the training data and used everything else as training data (30,320).

4.2.3 TensorFlow and TensorBoard

TensorFlow[23] is an open source tool developed by Google that allows users to do a variety of data analysis and machine learning tasks. In our analysis, we use TensorFlow to help us create a machine learning classifier model for our data. TensorBoard[24] is a visualization tool that allows the user to better understand, debug, and improve TensorFlow models.

[23] "TensorFlow." https://www.tensorflow.org/.
[24] "TensorBoard: Visualizing Learning." https://www.tensorflow.org/guide/summaries_and_tensorboard.

TensorBoard also allows other tool integrations, such as the What-If Tool (explained in the section 4.2.4 below).

To use TensorFlow, we simply inputted our training data and which feature we wanted to use as a label (FIOFS_TYPE). TensorFlow splits the data into features and labels and outputs a classifier as a tensorflow model. We could then use this Tensorflow Model with Google's What-if tool.

4.2.4 Google's What-If Tool

In September of 2018, Google released the open source What-If Tool, a new dashboard that integrates into the TensorBoard web application. The main purpose of this tool is to allow users to analyze machine learning models without writing their own code. The tool allows users to visualize how their model performs on data sets, see how the results change with changes in the data, analyze each feature of the data, and much more.

For the analysis presented in the analysis section, we primary used the following two capabilities of the What-If Tool:

1. *Performance and Fairness*
2. *Show Nearest Counterfactual*

4.2.4.1 Performance and Fairness

The first What-If Tool analysis feature that we used is the confusion matrix, found in the *Performance and Fairness* tab, which shows percentages for true positive, false positive, true negative, and false negative. For our model, these four values mean the following:

- True positive: percentage of data points that the model *correctly* classifies as 1
- False positive: percentage of data points that the model *incorrectly* classified as 1 (individuals that were not frisked (F) or searched (S), but the model says that they were)
- True negative: percentage of data points that the model *correctly* classifies as 0
- False negative: percentage of data points that the model *incorrectly* classifies as 0

Using these four values from the confusion matrix, we can evaluate the accuracy of a model on classifying individuals. In addition, we can compare these values over different

models and different test data and make conclusions based on the differences and similarities we see.

To see more details about the tools in the *Performance and Fairness* tab, see Appendix G.

4.2.4.2 Show nearest counterfactual

The *Show Nearest Counterfactual* tool within the What-If Tool, we can find the nearest neighbor of a point from a different classification. For example, if individual A was classified as 1, then *Show Nearest Counterfactual* will find an individual B with classification 0. The values for the features of individual B will be as close as possible to the values for individual A.

For our project, this was helpful in doing manual investigation into why similar points were classified differently, providing insight into which features make a difference in classifications. See Appendix G for more detail on how this tool works.

4.3 Overall Model Building Process - Step By Step

To perform the process of generating training and test data, training a model, and evaluating that model using the What-If Tool, we performed the following steps.

1. Use *import.py* script in Appendix B to output our training data set, which is a modified version of the Boston Police FIO Data, but with data encoded to work better with our classifier (See Table 1 in section 4.2.1 for specific feature encoding)
2. Generate a random subset of data to be our testing data set, also using a variation of *import.py*, with the same data modifications.
3. Use the *WIT from scratch - From CSV to trained model to WIT.ipynb* (Appendix C) to feed our training data into TensorFlow and generate a linear classifier model.
4. The script from step 3 also sets up a TensorBoard dashboard with the What-If Tool in the browser, and outputs a tensorflow model that runs locally.
5. Use the in-built analysis capabilities of the What-If tool to draw further conclusions.

With our model established, we then moved to assess the fairness of the model, discussed below in section 5.

5 Methods - Analyzing Our Model

As previously mentioned, our definitions of fairness are of two kinds: one kind deals with the fairness of past decisions (like Demographic Parity and Equal Accuracy) and the other kind deals with the fairness of potential, future decisions (Group Unaware, Group Threshold, Equal Opportunity).

When dealing with definition of fairness such as Demographic Parity, we will use standard methods of statistical analysis to directly analyze the fairness of the situation. When dealing with "potential" decision fairness definitions, however, such as Equal Opportunity and Group Unaware, we will use machine learning to create a model, then assess the fairness of the model, and analogize the fairness of the model to fairness in reality. This means, we need to define how we plan to analyze fairness of a model and how to analogize the fairness of a model to fairness in reality.

5.1 Assessing the Fairness of a Model

To assess the fairness of our model, we need a way to compare the model predictions to to the real results. To this end we introduce the concepts of a True Positive, False Positive, True Negative, False Negative, all of which can be found in a *confusion matrix* (section 4.2.4.1):

1. *True Positive:* the predicted and actual classification are both positive[25]
2. *False Positive:* the predicted classification is positive, but actual classification is negative
3. *True Negative:* When the predicted and actual classification are both negative
4. *False Negative:* the predicted classification is negative, but actual classification is positive

In other words the "False" metrics tell us when our model is incorrect in its prediction, and "True" metrics tell us when our model is correct. "Positives" tell us when the model, which we assume mimics police behavior, wants to frisk someone, and "Negatives tell us when the model wants to let someone go.

[25] Note that positive here means an output of 1, which means that the individual was frisked or searched (see Table 2 in section 4.3.2).

Our ultimate motivation is to provide a method of analysis that can be used to gain insights into stop and frisk police data, which can then be used to inform policy makers, police departments, and the general public who are interested in stop and frisk. Thus, we chose to use definitions of fairness that would map easily to legal and societal definitions of fairness without oversimplifying or overcomplicating. In particular, we chose to analyze the Group Unaware, Equal Opportunity, and Demographic Parity definitions of fairness.

These three definitions of fairness, we determined, are the definitions that provide the best analogy to the real world legal definitions of fairness discussed in Section 3.2.

5.2 Defining Fairness for Our Specific Analysis

From the original five definition of fairness presented in section 3.1, we chose three definitions of fairness that we felt matched the legal definitions of fairness the best. We looked at **Group Unaware**, **Equal Opportunity**, and **Demographic Parity**[26].

To illustrate the three metrics of fairness in specific relation to this data, suppose we are trying to determine whether the stop and frisk process is fair on the basis of race (DESCRIPTION column in our data).

1. In the **Demographic parity** view of fairness, the demographics of people who are ultimately searched or frisked after being stopped should be proportional the demographics of the people who were stopped. In other words, if 80% of people stopped were Hispanic, then 80% of the people who were frisked or searched should also be 80%. Note that this is a direct evaluation of whether or not the past decisions made are “fair.”
2. In the **Group Unaware** view of fairness, we say that it is fair when police choose to search or frisk someone, they do not take race into account at all. In other words, from a decision making point of view, a white person and an Asian person with identical features other than race would have the same chance of being searched or frisked. This view of fairness is common. Note, that here we are evaluating whether a potential decision (resulting from the current decision making process), is fair, and thus we assess the fairness of our ML model then analogize this to the real world.
3. In the **Equal Opportunity** view of fairness, the same percentage of people across different races, who are likely to be innocent, have the same chance of frisked or searched. In other words, what would be unfair is 90% of white people who are low

[26] https://pair-code.github.io/what-if-tool/ai-fairness.html

risk at being offenders to be left alone, but only 40% of black people who are low risk at being offenders to be left alone. Note, that here we are again evaluating whether a potential decision (resulting from the current decision making process), is fair and assess the fairness of our ML model then analogize this to the real world.

It must be noted also, that none of these definitions are perfect. The group unaware view of fairness can easily lead to disparate impact, where one racial group could inadvertently end up being the target of stop-and-frisk procedures. This could be because the other features are, in the absence of race, weighted more heavily, and are societally more likely to be found among a particular racial group through no fault of their own. The demographic parity and equal opportunity views of fairness contrast the group unaware view because race is accounted for and analyzed.

Thus, it must be made clear that in evaluating the fairness of the Boston Police Stop and Frisk practice, we are evaluating with respect to these specific fairness criteria, which may contradict other metrics of fairness. In other words, the procedures could be fair from one view of fairness and unfair from another. As a result, iit is essential to understand the definition of fairness used in a particular method of analysis.

In the next section we discuss the specific analysis method used for each of the three definitions of fairness we choose.

5.2.1 Methods of Analysis for Demographic Parity

To analyze the data with regards to the demographic parity view of fairness, we simply took our model's predictions and compared it to the actual classifications in the test data. This analysis was purely statistical and did not involve any machine learning.

If our test data was 80% male, then we would expect that out of the people who the model classified as searched or frisked, 80% would be male. If this is not the case, then it implies that the process by which the boston police decide to frisk or search is not fair from a demographic parity definition.

5.2.2 Methods of Analysis for Group Unaware Fairness

To test for group unawareness, we used a method of analysis we are calling *feature control*. Essentially, we used one original training dataset to generate multiple training data sets that trivialize different features.

We trivialize a given feature by replacing all values of the particular feature with the same, single word. For example, with DESCRIPTION (i.e. race), we replaced all values (B(Black), H(Hispanic), W(White), etc.) with the word "RACE." This way of trivializing a feature is valid because the model will only place value on features that are different from each other. By making the feature uniform, we essentially tell the model that this feature is not an important differentiator in the data. We then fed these different datasets into our machine learning algorithm to generate multiple hypotheses.

Thus, for each of our 9 features, we built 9 different models, where each disregarded one unique feature. We also built an additional all-features model that did not trivialize any feature and treated this model as our baseline. We then ran all of our models with the same test data set, and compared how each one-feature-less model compared with the all-features model.

We expect a one-feature-less model to look the same as our all-features model if there is group unaware fairness for that feature, as it suggests that the one particular feature was not very important in the ultimate model. To do this, we examined the values in the confusion matrices for these models.

For example, we could have a model that was generated without considering race, finding correlations in the training data based on non-race features. We then evaluate this model on some test data and determine how well it classifies this unseen data. If our model classifies the data very differently from the all-features model classification, this would imply that race is a crucial component in the decision making process of whether or not someone is frisked or searched. Similarly, if the model classifies the test data very similarly to the all-features model, then this implies that race is not a crucial component in the police's decision making process of whether or not someone is frisked or searched, and that the police are group-unaware fair with regards to race.

5.2.3 Methods of Analysis for Equal Opportunity.

The method we created to test for the equal opportunity view of fairness we called *test control*. With this method, we used all the features to come up with a good model that

characterizes the data (the same model as the all-features model from 5.2.2 above). We then created a new, almost-identical set of data where we would change one set of features to be a particular value. For example, in the case of race ,we would change our test data set such that every value for DESCRIPTION in the data set would be “W(White)”. If we were doing test control analysis on the SEX feature, we might change every value for the SEX column to be “MALE.”

For example, suppose there is data point for a Hispanic man who was not frisked nor searched (output is 0). We could perform test control analysis and make his DESCRIPTION value to be “W(White”) (instead of the original “H(Hispanic)”). Then, suppose the model predicts that he is frisked or searched (output is 1). The two data points are identical except for race, but the model classifies them differently. This means that the white man is more likely to be frisked or searched than a hispanic man. In other words, race altered this individual’s chance of being frisked or searched. This situation would, therefore, fail the equal opportunity fairness test.

This method of analysis is very similar to using the *show nearest counterfactual* functionality of the What-If Tool that was discussed before in section 4.3.3.2. However, since there are 50,000 records in the test data, we chose to view the results of our test control analysis using numerical values from confusion matrices generated from the model. For example, if the false positive rate went up by changing all DESCRIPTION values to “W(White),” this means that being white increases an individual’s chance of being frisked or searched, and this would again fail the equal opportunity fairness test.

6 Results of Analysis[27]

6.1 Demographic Parity Fairness

For the analysis done for the demographic view of fairness, we did not use any machine learning. Rather, we wrote a python script that performed statistical analysis (Appendix E).

Below are the breakdowns of our test data for each feature. Our test data was 50,000 records. The "# in Subgroup" category is the number of people in our test data that fit a particular value for a given feature. The "% of total" column expresses the percent composition each value makes with regards to the total 50,000 records. Out of these 50,000 records, 1173 were frisked or searched. The column "F or S" expresses the number of individuals belonging to a particular feature value who were frisked or searched. The "% of F or S" column states the percent composition each feature value makes with regards to the 1173 frisked or searched individuals.

For example, in Table 3 below, we see that out of the 50,000 FIO encounters in our test data, 29,715 of them were with black individuals, which is 58.4% of the test data. 678 of those black individuals were frisked or searched, and this accounted for 57.8% ($\frac{678}{1173}$) of all frisks and/or searches in the test data.

Table 3

Description (Race)	# in Subgroup	% of Total	F or S	% of F or S
Black	29175	58.4%	678	57.80%
Hispanic	6550	13.1%	194	16.54%
White	11426	22.9%	257	21.91%
Middle Eastern/East Indian	153	0.3%	2	0.17%
Asian or Pacific Islander	430	0.9%	5	0.43%
American Indian or Alaskan Native	28	0.1%	0	0.00%
UNKNOWN	189	0.4%	2	0.17%
NO DATA ENTERED	2049	4.1%	35	2.98%
TOTAL	**50000**	**100.00%**	**1173**	**100.00%**

[27] See Appendix A for a list of relevant files and scripts to our analysis

The demographics of people who are ultimately searched or arrested after being stopped should be proportional the demographics of the people who were stopped. Since 13.1% of the people stopped were Hispanic, then around 12-14% of the people who were ultimately arrested and searched should also be Hispanic. Instead, however, we see a larger deviation. 16% of the people who were frisked are Hispanic. Looking at the other races, however, we see that all other races, are generally proportional. Thus the Boston Police are generally fair across races from a demographic parity point of view, except for Hispanics, where they "unfairly" frisk a larger percentage of Hispanics than the total group Hispanic percentage warrants.

Table 4

SEX	Total	% of Total	F or S	% of F or S
Male	44050	88.1%	1091	93.01%
Female	5861	11.7%	82	6.99%
Unknown	89	0.2%	0	0.00%
TOTAL	**50000**	**0.00%**	**1173**	**100.00%**

Since 88.1% of the people stopped were male, then around 87-89% of the people who were ultimately arrested and searched should also be male. Instead, however, we see that 93.01% of those who were frisked/searched were male. This implies that according to the demographic parity definition of fairness, the police have a bias against males in stop and frisk encounters.

Table 5

PRIORS	Total	% of Total	F or S	% of F or S
Yes	37163	74.3%	868	74.00%
No	5392	10.8%	150	12.79%
Unknown	541	1.1%	15	1.28%
Blank	6904	13.8%	140	11.94%
TOTAL	**50000**	**0.00%**	**1173**	**100.00%**

Interestingly it seems as though, when it comes to prior history, overall, the treatment is fair. Approximately 74% of those stopped had prior history with the police, and approximately 74% of those frisked/searched had prior history with the police.

Table 6

AGE	Total	% of Total	F or S	% of F or S
Teens	9082	18.2%	254	21.65%
Twenties	23071	46.1%	541	46.12%
Thirties	9283	18.6%	219	18.67%
Middle	7977	16.0%	149	12.70%
Senior	587	1.2%	10	0.85%
TOTAL	**50000**	**100%**	**1173**	**100.00%**

Since 18.2% of the people stopped were teenagers, but 21.65% of the people who were ultimately arrested and searched were teenagers, we can say that demographic parity-wise, the police are biased against teenagers, and are more likely to search/frisk them than the percentage out of the total warrants. We see this increase of approximately 3% reflected in a decrease of approximately 3% among people of "middle" age. Middle age is defined as those above 40 and below 60. Thus, with this view of fairness, the police seem to be more suspicious of teenagers and less suspicious of those who look to be middle-age.

Table 7

Complexion	Total	% of Total	F or S	% of F or S
Light	10727	21.5%	294	25.06%
Dark	9103	18.2%	247	21.06%
Med	19922	39.8%	472	40.24%
Brown	1083	2.2%	11	0.94%
White	170	0.3%	1	0.09%
Ruddy	34	0.1%	1	0.09%
Fair	585	1.2%	16	1.36%
Clear	10	0.0%	1	0.09%
Other	817	1.6%	13	1.11%
No Data Entered	7549	15.1%	117	9.97%
TOTAL	**50000**		**1173**	**100.00%**

Being "light" or "dark" results in an approximately 3-4% increase deviation between the "fair" percentage (% total) and the actual percent frisked (% of F or S). Thus, it would seem

the police are slightly biased against those who are “light” skinned and those who are “dark” skinned, but because complexion is so difficult to classify (i.e. what’s the difference between white/ light/ fair or dark/ brown?). Since these definitions are not very clear, we are skeptical of drawing any major conclusions with these numbers, especially since they are so small (<5%). Being “brown” resulted in the largest decrease from the expected percentage, but this is almost 1%, so we believe we can consider this negligible. Also, we noticed that a large percentage of the data had “no data entered.” If we split this equally between dark and light, however, we noticed that the results are generally fair.

If we map Fair, Clear and White to “Light” and “Ruddy”/”Brown” to Dark, and split the “No Data Entered equally among both we get the following: (Table 8)

Table 8: Modified version of Table 7

Complexion	**Total**	**% of Total**	**F or S**	**% of F or S**
Light	15266.5	30.53%	370.5	31.59%
Dark	13994.5	27.98%	317.5	27.07%
Med	19922	39.84%	472	40.24%
OTHER	817	1.63%	13	1.11%
Total	50000	100%	1173	100.00%

In this we can see that the results are generally fair overall.

Table 9: Summary of Results of Demographic Parity

Feature	**Fairness Assessment**
DESCRIPTION (Race)	**Mostly Fair;** except for with Hispanics, where they are frisked/searched more than the expected amount (3% more)
SEX	**Unfair**; Men are frisked/searched more than expected (5% more)
PRIORS	**Fair** Treatment
AGE_AT_FIO_CORRECTED	**Unfair**; Teens are frisked/searched more than expected (3% more), Middle aged people are fisked less than expected (3% less)
COMPLEXION	**Difficult to Conclude, but Fair with modifications**.

6.2 Group Unaware Fairness: Feature Control Analysis

Screenshots from the What-If Tool for our feature control analysis can be found in Appendix H.

Table 10: Summary of Confusion Matrix Findings using Feature Control Analysis

Feature	True Positive	False Positive	True Negative	False Negative
Baseline	17.50%	10.70%	54%	17.70%
SEX	17.30%	11.10%	53.60%	17.90%
PRIORS	17.30%	10.70%	54.10%	17.90%
COMPLEXION	18.00%	11.40%	53.40%	17.20%
FIOFS_REASONS	17.50%	12.70%	52.00%	17.70%
OFFICER_ID	12.90%	10.30%	54.40%	22.40%
FIO_DATE	17.10%	10.60%	54.10%	18.10%
DESCRIPTION (Race)	17.70%	11.10%	54.70%	17.50%
DIST	17.90%	11.40%	53.40%	17.30%
AGE_AT_FIO_CORRETED	17.50%	10.90%	53.90%	17.70%

To test for the *Group Unaware* view of fairness, we test how a one-feature-less model compares against the all-feature model. If the model not using the DESCRIPTION feature classifies with less accuracy than the all-feature model, this would imply that race is a critical feature considered in the decision to frisk or search someone, and that therefore the police are not racially group unaware. What does classifying with "less accuracy" mean with regards to the above table?

In the above chart, we look to see how much the false positives and false negatives increase or decrease. If the absolute value of the difference between the baseline and false positive rate is greater than that of the baseline and false negative, it means without that particular feature, the police are more likely to frisk or search. In other words:

$$\text{If } |\text{Baseline FP} - \text{Feature X FP}| > |\text{Baseline FN} - \text{Feature X FN}|,$$
then without Feature X, the police are more likely to frisk or search

For example, with FIOFS_REASONS, the absolute difference in false positive from baseline false positive is 2% ($|12.70 - 10.70| = 2$), whereas the absolute difference from the baseline of the false negative is 0% ($17.70\text{-}17.70 = 0$). Thus, in this case, we would say that without accounting for FIOFS_REASONS, the police are more likely to frisk or search. Conversely, it also means that when accounting for FIOFS_REASONS, the police are less likely to frisk or search. Knowing the reason for the FIO action can help avoid being frisked and searched, since without knowing the FIOFS_REASONS data, the model seems to want to frisk or search more people.

With that in mind, looking at Table 10, we see that a large outlier is the OFFICER_ID feature. When we created a model with all of the same values for the OFFICER_ID column and used that to classify the data, there was a large 5% increase in false negatives from the baseline, but not a large increase in false positives. The true positive rate decreased by 5%.

This means that 5% of the data which was a true positive were all incorrectly classified as negative. So without accounting for the OFFICER_ID, the model is more likely to classify someone as not being searched or frisked. In other words, the identity of the officer who makes the arrest matters, increasing the number of searches and frisks!

This was in line with what we saw when we were looking at counterfactuals. Some of the *show nearest counterfactual* data point pairs only differed in which officer was handling the situation, as shown below in Figure 3.

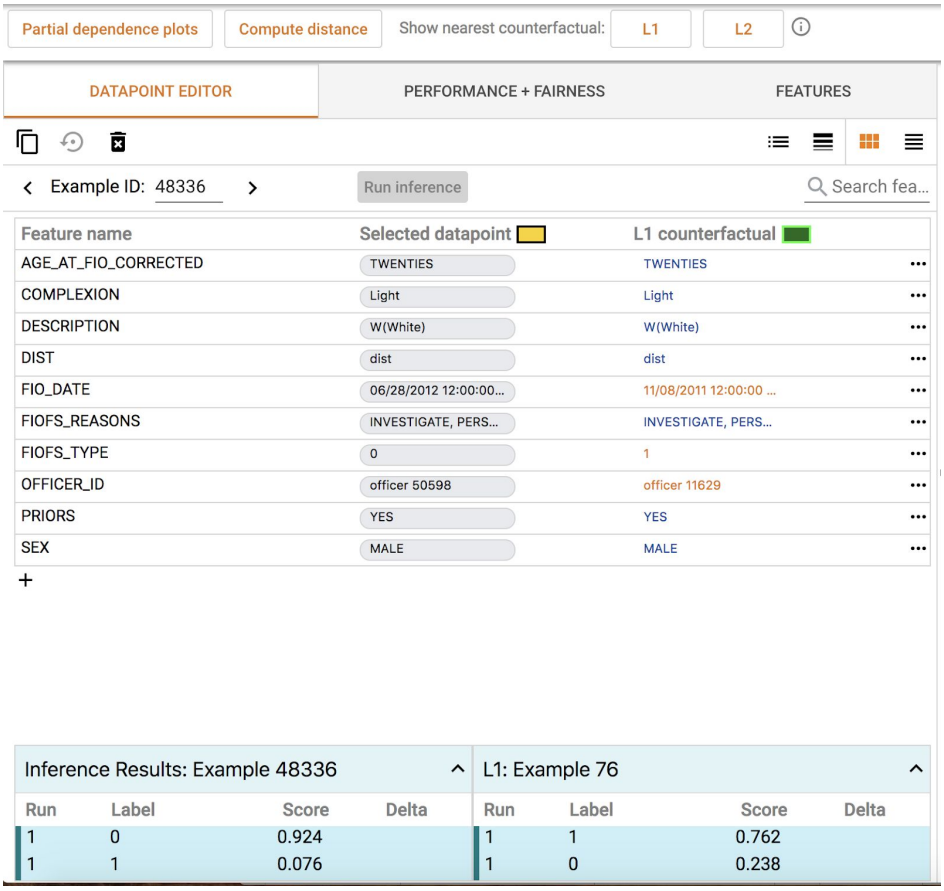


Figure 3: show nearest counterfactual result, where features values are identical except for a different OFFICER_ID and FIO_DATE

We can see in Figure 3 that every single value for the features are the same except the FIO_DATE and the OFFICER_ID. We weight this change in outcome as a result of different OFFICER_ID values and not different dates, given that FIO_DATE values made minimal different in the feature control analysis results (none of the confusion matrix percentages differed by more than 0.5% from the baseline).

This means that having a different OFFICER_ID resulted in one person being frisked or searched and another being let off.

From a group unaware fairness perspective, it seems like the boston police as a whole are very fair and group unaware in regards to most features, except with OFFICER_ID. The decision making process does depend on the specific police officer who is making the decision, meaning that this decision making is, perhaps, more subjective than ideal.

Table 11: Summary of Results of Group Unaware

Feature	Fairness Assessment
Baseline	Fair
SEX	Fair
PRIORS	Fair
COMPLEXION	Fair
FIOFS_REASONS	Fair
OFFICER_ID	Unfair: officer making the decision matters!
FIO_DATE	Fair
DESCRIPTION (Race)	Fair
DIST	Fair
AGE_AT_FIO_CORRECTED	Fair

6.3 Equal Opportunity Fairness: Test Control Analysis

Screenshots from the What-If Tool for our test control analysis can be found in Appendix H.

Table 12

DESCRIPTION	TP	FP	TN	FN
Baseline	18.10%	9.30%	55.30%	17.20%
A(Asian or Pacific Islander)	19.20%	10.60%	54%	16.20%
B(Black)	18.50%	9.80%	54.80%	16.80%
H(Hispanic)	17.90%	9.20%	55.50%	17.40%
I(American Indian or	11.50%	4%	60.60%	23.90%

Alaskan Native)				
M(Middle Eastern/ East Indian)	14%	5.70%	58.90%	21.30%
W(White)	16.50%	7.70%	57.00%	18.90%

We test here whether changing one's race changes the chance of being frisked or searched.

When everyone's race was changed to be A(Asian or Pacific Islander), the algorithm assigned more people to be positive (frisked or searched). In the baseline, 27.4% of individuals were assigned the label "frisked or searched" (true positive + false positive). When everyone was changed to being Asian, 29.8% of individuals were assigned the label "frisked or searched," with everything else being the same as before. So, we can conclude that being perceived as Asian increases your chance of being frisked or searched.

Being Black also increases your chance of being frisked or searched, though by slightly less than being Asian. When everyone was changed to being Black, 28.3% of individuals were assigned the label frisked or searched.

Being Hispanic slightly decreases your chance of being frisked or searched. Changing everyone to be hispanic resulted in 27.1% of individuals being classified as searched or frisked.

At first glance, being American Indian/Alaskan Native or Middle Eastern may seem like it drastically decreases your chance of being frisked or searched due to the drastic change in TP, FP, TN, and FN, but this is likely due to the fact that there are no Native Alaskans/Americans Indians who were frisked or searched and there were very few Middle Eastern people who were frisked or searched. Thus, the model is highly biased in always outputting a negative for these two races. Hence, we see the heavy increase in true negative and false negative. Thus, we claim that while being American Indian/Alaskan Native may decrease your chance of being frisked or searched in some small way, it is not easy to determine the exact extent of the change in chance.

In the Asian test, however, you would expect similar results since the number of Asian people who were searched and frisked was also very small, but it instead had the opposite effect, making the Asian results above more interesting and significant.

Being White decreases your chance of being frisked or searched. When everyone was changed to being White, 24.2% of individuals were assigned the label frisked or searched, 3.2% higher than the baseline.

Table 13

SEX	**TP**	**FP**	**TN**	**FN**
Baseline	18.10%	9.30%	55.30%	17.20%
MALE	18.70%	10.70%	53.90%	16.60%
FEMALE	4.80%	1.00%	63.60%	30.60%

Being Male also increases your chance of being frisked or searched. When everyone was changed to being Male, 29.4% of individuals were assigned the label frisked or searched, compared to only 27.4% in the baseline test data.

Table 14

PRIORS	**TP**	**FP**	**TN**	**FN**
Baseline	18.10%	9.30%	55.30%	17.20%
YES	19.00%	10.40%	54.30%	16.30%
NO	17.20%	8.60%	56.10%	18.10%

The model is more likely to predict that a person with prior experience will be frisked or searched. When everyone had prior experience, we see that 29.4% of individuals were assigned the label of frisk/searched, which is the same amount as when all of them were made male, and 2% higher than the baseline.

Table 15

AGE	**TP**	**FP**	**TN**	**FN**
Baseline	18.10%	9.30%	55.30%	17.20%
TEENS	20.80%	12.70%	52.00%	14.60%

TWENTIES	18.20%	9.70%	54.90%	17.10%
THIRTIES	17.00%	8.50%	56.20%	18.40%
MIDDLE	14.40%	6.10%	58.60%	20.90%
SENIOR	8.50%	2.50%	62.20%	26.80%

In Table 15, we clearly see that as age increases, there are less frisks and searches. Compared to the baseline of 27.4% positive outcomes, teens were predicted to be searched or frisked a whopping 33.5% of the time. For people in their twenties, that percentage was 27.9%, which isn't too far from the baseline. As we increase age, however, that percentage continues to drop. People in their thirties are 25.5% likely to be classified as searched or frisked. For middle aged people, it was 20.5%. For seniors, it was 11%. The biggest jump is between being middle aged and being a senior, which implies that being above 60 years old significantly reduces your chance of being searched or frisked, compared to all other age groups.

Table 16

COMPLEXION	**TP**	**FP**	**TN**	**FN**
Baseline	**18.10%**	**9.30%**	**55.30%**	**17.20%**
Brown	17.40%	8.80%	55.90%	17.90%
Clear	16.30%	7.60%	57.00%	19.00%
Dark	18.70%	10.10%	54.50%	16.60%
Fair	14.70%	6.30%	58.40%	20.60%
Light	18.90%	10.30%	54.40%	16.50%
Medium	18.60%	9.90%	54.70%	16.80%
Ruddy (only 34 records)	25.80%	20.10%	44.60%	9.60%
White	12.10%	4.40%	60.20%	23.20%

Similar to the situation with the DESCRIPTION (race) feature, some of these categories only describe very small subsets of the total group, and are therefore, difficult to

confidently analyze. In this case, “White”, “Clear”, or “Ruddy” feature values are difficult to analyze since they only contained approximately 100 or less people each.

Regarding the other complexion categories, however, we see that being “Brown” or “Fair” decreases your chance of being searched or frisked. When everyone was made “Brown”, 26.2% were classified as being searched or frisked and similarly, with “Fair”, 21% were classified as being searched or frisked.

Thus, it is advantageous to be “Fair” or “Brown”. Being “Fair” in particular, results in a large decrease of likelihood of being frisked (27.4% to 21% drop).

Using similar analysis, we determined that being “Light” or “Medium” or “Dark” results in a slight increase of being frisked. These results, however, are difficult to analyze also because complexion is not very objective. For example, we do not know the difference between white, fair, and light or the difference between brown, dark, and medium.

Table 17: Summary of Results of Equal Opportunity

Feature	**Fairness Assessment**
DESCRIPTION (Race)	**Asians:** disadvantaged -- more likely to be S or F **White:** advantaged -- less likely to be S or F **Black:** slightly disadvantaged **Hispanic:** slightly advantaged **Native Alaskan/American Indian:** difficult to say **Middle East:** difficult to say
SEX	**Male**: disadvantaged, **Female:** advantaged
PRIORS	**Yes:** disadvantaged slightly
AGE_AT_FIO_CORRECTED	**Teens:** significantly disadvantaged **Twenties:** slightly disadvantaged **Thirties:** advantaged **Middle:** significantly advantaged **Senior:** significantly advantaged
COMPLEXION	Difficult to Conclude, but being “Fair” seems to have an advantage

6.4 Results Summary for 3 Definitions of Fairness

Here we provide the summary tables of the analysis results from each of the definitions of fairness in a consolidated section.

Demographic Parity

Feature	**Fairness Assessment**
DESCRIPTION (Race)	**Mostly Fair;** except for with Hispanics, where they are frisked/searched more than the expected amount (3% more)
SEX	**Unfair**; Men are frisked/searched more than expected (5% more)
PRIORS	**Fair** Treatment
AGE_AT_FIO_CORRECTED	**Unfair**; Teens are frisked/searched more than expected (3% more), Middle aged people are fisked less than expected (3% less)
COMPLEXION	**Difficult to Conclude, but Fair with modifications**.

Group Unaware

Feature	**Fairness Assessment**
Baseline	Fair
SEX	Fair
PRIORS	Fair
COMPLEXION	Fair
FIOFS_REASONS	Fair
OFFICER_ID	Unfair: officer making the decision matters!
FIO_DATE	Fair
DESCRIPTION (Race)	Fair
DIST	Fair
AGE_AT_FIO_CORRECTED	Fair

Equal Opportunity

Feature	Fairness Assessment
DESCRIPTION (Race)	**Asians:** disadvantaged -- more likely to be S or F **White:** advantaged -- less likely to be S or F **Black:** slightly disadvantaged **Hispanic:** slightly advantaged **Native Alaskan/American Indian:** difficult to say **Middle East:** difficult to say
SEX	**Male**: disadvantaged, **Female:** advantaged
PRIORS	**Yes:** disadvantaged slightly
AGE_AT_FIO_CORRECTED	**Teens:** significantly disadvantaged **Twenties:** slightly disadvantaged **Thirties:** advantaged **Middle:** significantly advantaged **Senior:** significantly advantaged
COMPLEXION	Difficult to Conclude, but being “Fair” seems to have an advantage

7 Conclusion

7.1 Significant Results

Through our analysis, we found many significant results regarding the decision making of the BPD in stop and frisk situation from 2011 to 2015.

First, through our feature control analysis (for Group Unaware definition of fairness), we found that the specific officer in the situation makes a difference in the outcome. In this analysis, taking away the OFFICER_ID values in the training data set for our model caused an increase in classifications of 0, suggesting that OFFICER_ID contributed significantly to individuals being classified as 1 (frisked or searched).

Second, through our test control analysis (for Equal Opportunity definition of fairness), we found that sex and age have a significant impact on decisions made by Boston police officers. Specifically, males were much more likely to be searched or frisked, and the younger an individual is, the more likely they are to be searched or frisked.

Third, we saw some important outcomes in our analysis regarding race, the DESCRIPTION column in the data. Our analysis of race using our three methods reveals exactly why fairness is sometimes hard to determine—our three methods of analysis yielded fairly different results in terms of the race column. Our demographic parity analysis concluded that outcomes for race were mostly fair, with the exception of being Hispanic. Our equal opportunity analysis concluded that White people had the most advantage and Hispanic had slight advantage, while being Asian gave you significant disadvantage and being Black gave you slight disadvantage. Finally, our feature control analysis showed no significant differences caused by race. Our three methods of analysis model three different interpretations of fairness, and we see here that the three sets of results differ as well.

Given our results, it is clear that specific officers, sex, age, and race all still play a big part in the decision making process of Boston police officers. These features all play an important role in at least one view of fairness we discussed, and these should be factors considered when thinking of stop and frisk. In legal cases, machine learning analysis can be used as a way to quantitatively support the reasoning behind why certain officers were biased in a stop and frisk situation. For example, in *Floyd v. City of New York*, machine learning analysis similar to ours could have been used to provide stronger support as to why the stop and frisk practices of NYPD officers were biased against people of a certain

race. Overall, we hope that our analysis provides a foundation and a framework for future machine learning analysis on stop and frisk data, as well as provides a way to provide quantitative justification for decisions about stop and frisk in legal situations.

7.2 Future Work

For future work on this subject, we hope to be able to perform more analysis using different machine learning models, beyond the simple linear classifier we used for the analysis presented in this paper. Our current model is only X% accurate. With more time, the next step would be to build a model that is more accurate, so that variations in the classification (i.e. changes in the confusion matrix) can be fully attributed to the feature we are testing, and not due to the general inaccuracy of the model. In the analysis above, we mitigated the effect of the general inaccuracy of the model by comparing all of our results to the baseline model (i.e. with no modifications). With a better baseline model, however, our conclusions would be asserted with more confidence.

In this way, we hope to verify our results with different types of models. In addition, we hope to dive deeper into feature control analysis (group unaware fairness) based on OFFICER_ID. Specifically, it would be interesting to train models based on the decisions of a specific police officer, and see what patterns or biases emerge from that trained model. Given that our feature control analysis showed us that police officer ID has a significant impact on the decision to search or frisk, we would expect to refine and add to this conclusion with further analysis.

We would also want to do feature control and test control analysis using the FIOFS_REASONS column present in the original data, that lists reasons why the officer stopped an individual, such as “INVESTIGATIVE, PERSONS.” This would be an interesting column to investigate because it ties in to the idea of “reasonable suspicion,” and analysis on this column could shed further light on what Boston police officers think reasonable suspicion is.

8 References

ACLU of Massachusetts. “Ending Racist Stop and Frisk”. https://www.aclum.org/en/ending-racist-stop-and-frisk

Arnold & Porter Kaye Scholer, LLP. “Analysis of NYPD Stops, 2013-2015”. https://www.nytimes.com/interactive/2017/05/30/nyregion/nypd-stop-and-frisk-report.html

Barrett, John. Q.. "’STOP AND FRISK’ IN 1968:DECIDING THE STOP AND FRISK CASES:A LOOK INSIDE THE SUPREME COURT'S CONFERENCE," St. John's Law Review, 72, 749 (Summer / Fall, 1998). https://advance.lexis.com/api/document?collection=analytical-materials&id=urn:contentItem:3VWV-S730-00CW-F01P-00000-00&context=1516831.

“BPD FIELD INTERROGATION AND OBSERVATION (FIO).” Analyze Boston. https://data.boston.gov/dataset/boston-police-department-fio.

Brownlee, Jason. “Gentle Introduction to Vector Norms in Machine Learning.” *Machine Learning Mastery*. https://machinelearningmastery.com/vector-norms-machine-learning/.

Center for Evidence-Based Policing. “Broken Window Policing”. https://cebcp.org/evidence-based-policing/what-works-in-policing/research-evidence-review/broken-windows-policing/.

“Data Key (Old RMS).” Analyze Boston. https://data.boston.gov/dataset/boston-police-department-fio/resource/1e5f1bc5-a0b4-4dce-ae1c-7c01ab3364f6.

“Districts.” bpdnews. http://bpdnews.com/districts/.

“Facets”, https://pair-code.github.io/facets/.

Gaffin, Adam. “BPD releases data on people officers talk to long enough to warrant a report.” Universal Hub. January 9, 2016.

https://www.universalhub.com/2016/bpd-releases-data-people-officers-talk-long-enough.

Gelman, Andrew, Jeffrey Fagan, and Alex Kiss. "An Analysis of the New York City Police Department's "Stop-and-Frisk" Policy in the Context of Claims of Racial Bias." Journal of the American Statistical Association, September 2007, 813-23. http://www.stat.columbia.edu/~gelman/research/published/frisk9.pdf.

National Police Foundation. "Compstat and Organizational Change: A National Assessment." https://www.policefoundation.org/projects/compstat-and-organizational-change-a-national-assessment/.

New York Civil Liberties Union. "Class Action Lawsuit Challenges NYPD Patrols of Private Apartment Buildings." March 28, 2012. https://www.nyclu.org/en/press-releases/class-action-lawsuit-challenges-nypd-patrols-private-apartment-buildings.

New York Civil Liberties Union. "Stop-and-frisk Data." https://www.nyclu.org/en/stop-and-frisk-data

"Playing with AI Fairness.". What-If Tool. https://pair-code.github.io/what-if-tool/ai-fairness.html.

Rayman, Graham. "NYPD Tapes 4: The Whistleblower, Adrian Schoolcraft." The Village Voice. June 15, 2010. https://www.villagevoice.com/2010/06/15/nypd-tapes-4-the-whistleblower-adrian-schoolcraft/.

San Francisco Police Department. "Compstat". http://sanfranciscopolice.org/compstat.

Sexton, Joe. "In New York, Crime Falls Along With Police Stops." ProPublica. January 16, 2018. https://www.propublica.org/article/in-new-york-crime-falls-along-with-police-stops.

"TensorFlow." https://www.tensorflow.org/.

"TensorBoard: Visualizing Learning." TensorFlow. https://www.tensorflow.org/guide/summaries_and_tensorboard.

"tf.estimate.LinearClassifier." TensorFlow. https://www.tensorflow.org/api_docs/python/tf/estimator/LinearClassifier.

The People of the State of New York, Respondent, v. John Francis Peters, Appellant, 18 N.Y.2d 238, 219 N.E.2d 595, 273 N.Y.S.2d 217, 1966 N.Y. LEXIS 1188 (Court of Appeals of New York July 7, 1966, Decided). https://advance.lexis.com/api/document?collection=cases&id=urn:contentItem:3RRS-WDK0-003C-C324-00000-00&context=1516831.

Tuttle, Ross and Erin Schneider. "Stopped-and-Frisked." The Nation. October 8, 2012. https://www.thenation.com/article/stopped-and-frisked-being-fking-mutt-video/.

Verma, Sahil and Julia Rubin. "Fairness Definitions Explained." *2018 ACM/IEEE International Workshop on Software Fairness*. http://fairware.cs.umass.edu/papers/Verma.pdf.

"What-If Tool." https://pair-code.github.io/what-if-tool/.

Willis, James and Stephen Mastrofski. "Compstat and Community Policing: Are They Compatible?" https://cops.usdoj.gov/pdf/workshops/thursday/WillisMastrofski.pdf.

Wofford, Taylor. "Did Bill de Blasio keep his promise to reform stop-and-frisk?" Newsweek. August 25, 2014. https://www.newsweek.com/did-bill-de-blasio-keep-his-promise-reform-stop-and-frisk-266310.

Appendix A: Relevant Files and Tools

In addition to TensorFlow, TensorBoard, and the What-If Tool, the following are Python scripts and CSV files that were relevant to our analysis.

- boston-police-department-fio.csv[28]
    - The original dataset from the Boston Police Department. Contains 152,230 rows of data.
- bpd_training_data.csv (Appendix D)
    - The dataset used to train our TensorFlow model.
- import.py (Appendix B)
    - The Python script we wrote to create bpd_training_data.csv from boston-police-department-fio.csv. In addition to making modifications to the data and outputting it as a new file, this script also randomizes the order of the rows. The modifications to data values are shown in section 4.2.1.
    - We also used variations of this script to divide the data into a training data set and a test data set for our feature control analysis (section 6.2), and to modify values in our test data for our test control analysis (section 6.3).
- WIT from scratch - From CSV to trained model to WIT.ipynb (Appendix C)
    - Jupyter notebook Python file provided by James Wexler, Google employee who worked on creating the What-If Tool. It takes in a dataset, trains a TensorFlow linear classifier, and sets up a TensorBoard dashboard with the What-If Tool in the browser.
    - We modified the parts of the code that took care of reading an input csv file to work with bpd_training_data.csv.
    - This file also changes the data values in the FIOFS_TYPE column to 0 or 1—0 if the data value doesn't contain "F" or "S", and 1 if it contains either.
- data_breakdown.py (Appendix E)
    - Python script we wrote to show how many individuals fell under each value for each column, and for each of those values, the split between frisked or searched and not frisked nor searched.
    - This script was used for demographic parity analysis (section 6.1).
    - The output of this script is data_breakdown.csv (Appendix F).

[28]https://data.boston.gov/dataset/boston-police-department-fio/resource/c696738d-2625-4337-8c50-123c2a85fbad

Appendix B: Data Modification Script (import.py)

```
import csv
import random
import sys
import pandas

"""
To run, navigate to a directory that contains the
boston-police-department-fio.csv file and this python script.

Then, run the following in command line:

python3 import.py

This script outputs bpd_training_data.csv, which is a modification of the
origin input csv file.

In addition, it randomizes the rows.

Feature Encoding designed by Dheekshita Kumar (dhkumar@mit.edu) and Mary Zhong
(mzhong@mit.edu)

Written by Mary Zhong (mzhong@mit.edu) for 6.805 Group Project
"""

with open("./boston-police-department-fio.csv") as file,
open('bpd_training_data.csv', mode='w') as output:
    file_copy = open("./boston-police-department-fio.csv")
    total = sum(1 for line in file_copy) - 1
    print("Total rows in file: {0}".format(total))
    reader = list(csv.DictReader(file, delimiter=",", quotechar='"',
quoting=csv.QUOTE_MINIMAL))

    writer = csv.writer(output, delimiter=',', quotechar='"',
quoting=csv.QUOTE_MINIMAL)

    # headers:
    writer.writerow(["SEX", "FIO_DATE", "PRIORS", "COMPLEXION", "FIOFS_TYPE",
"FIOFS_REASONS", "AGE_AT_FIO_CORRECTED", "DESCRIPTION", "DIST", "OFFICER_ID"])

    index = 0
    reader_index = 0
    random_order = list(range(total))
```

```
    random.shuffle(random_order)
    for i in random_order:
        row = reader[i]

        sex = row["SEX"] if row["SEX"] != "" else "BLANK"
        priors = row["PRIORS"] if row["PRIORS"] != "" else "BLANK"
        complexion = row["COMPLEXION"] if row["COMPLEXION"] != "" else "BLANK"
        fiofs_type = row["FIOFS_TYPE"] if row["FIOFS_TYPE"] != "" else "BLANK"
        fiofs_reasons = row["FIOFS_REASONS"] if row["FIOFS_REASONS"] != "" else
"BLANK"
        officer_id = "officer " + row["OFFICER_ID"] if row["OFFICER_ID"] != ""
else "BLANK"
        fio_date = row["FIO_DATE"] if row["FIO_DATE"] != "" else "BLANK"
        description = row["DESCRIPTION"] if row["DESCRIPTION"] != "" else
"BLANK"
        dist = row["DIST"] if row["DIST"] != "" else "BLANK"

        age = row["AGE_AT_FIO_CORRECTED"]
        if age == "":
            age = "BLANK"
        elif int(age) < 20:
            age = "TEENS"
        elif int(age) < 30:
            age = "TWENTIES"
        elif int(age) < 40:
            age = "THIRTIES"
        elif int(age) < 60:
            age = "MIDDLE"
        else:
            age = "SENIOR"


        data = [sex,fio_date, priors, complexion, fiofs_type, fiofs_reasons,
age, description, dist, officer_id]

        writer.writerow(data)
```

Appendix C: WIT from scratch - From CSV to trained model to WIT

```
# This notebook shows the process of loading up a dataset from CSV, training a
simple classifier to
# predict one of the columns, then using the What-If Tool to analyze the
dataset and the model trained on it.

# It is shown with both the UCI census binary classification task and the UCI
iris multiclass classification task.

### Setup (install Jupyter, TF, and TF Serving in a virtualenv):

# virtualenv tf
# source tf/bin/activate
# pip install --upgrade pip
# pip install jupyter
# pip install tensorflow (or tensorflow-gpu)
# docker pull tensorflow/serving

### Make sure there is a folder named "data" in the same directory as this
file.



## Define helper functions
import pandas as pd
import numpy as np
import tensorflow as tf
from tensorflow import data

# Writes a pandas dataframe to disk as a tfrecord file of tf.Example protos,
# using only the dataframe columns specified. Non-numeric columns are treated
# as strings.
def write_df_as_tfrecord(df, filename, columns):
    writer = tf.python_io.TFRecordWriter(filename)
    for index, row in df.iterrows():
        example = tf.train.Example()
        for col in columns:
            if df[col].dtype is np.dtype(np.int64):
                example.features.feature[col].int64_list.value.append(row[col])
            elif df[col].dtype is np.dtype(np.float64):
                example.features.feature[col].float_list.value.append(row[col])
            else:
```

```
example.features.feature[col].bytes_list.value.append(row[col].encode('utf-8'))
        writer.write(example.SerializeToString())
    writer.close()

# Creates a tf feature spec from the dataframe and columns specified.
def create_feature_spec(df, columns):
    feature_spec = {}
    for f in columns:
        if df[f].dtype is np.dtype(np.int64):
            feature_spec[f] = tf.FixedLenFeature(shape=(), dtype=tf.int64)
        elif df[f].dtype is np.dtype(np.float64):
            feature_spec[f] = tf.FixedLenFeature(shape=(), dtype=tf.float32)
        else:
            feature_spec[f] = tf.FixedLenFeature(shape=(), dtype=tf.string)
    return feature_spec

# Parses a serialized tf.Example into input features and target feature from
# the provided label feature name and feature spec.
def parse_tf_example(example_proto, label, feature_spec):
    parsed_features = tf.parse_example(serialized=example_proto,
features=feature_spec)
    target = parsed_features.pop(label)
    return parsed_features, target

# An input function for providing input to a model from tf.Examples from tf
record files.
def tfrecords_input_fn(files_name_pattern, feature_spec, label,
mode=tf.estimator.ModeKeys.EVAL,
                       num_epochs=None,
                       batch_size=64):
    shuffle = True if mode == tf.estimator.ModeKeys.TRAIN else False
    file_names = tf.matching_files(files_name_pattern)
    dataset = data.TFRecordDataset(filenames=file_names)

    if shuffle:
        dataset = dataset.shuffle(buffer_size=2 * batch_size + 1)

    dataset = dataset.batch(batch_size)
    dataset = dataset.map(lambda tf_example: parse_tf_example(tf_example,
label, feature_spec))
    dataset = dataset.repeat(num_epochs)
    iterator = dataset.make_one_shot_iterator()

    features, target = iterator.get_next()
    return features, target
```

```
# Creates simple numeric and categorical feature columns from a feature spec
and a
# list of columns from that spec to use.
#
# NOTE: Models might perform better with some feature engineering such as
bucketed
# numeric columns and hash-bucket/embedding columns for categorical features.
def create_feature_columns(columns, feature_spec):
    ret = []
    for col in columns:
        if feature_spec[col].dtype is tf.int64 or feature_spec[col].dtype is
tf.float32:
            ret.append(tf.feature_column.numeric_column(col))
        else:

ret.append(tf.feature_column.categorical_column_with_vocabulary_list(col,
list(df[col].unique())))
    return ret


## BPD FIO Training data

tfrecord_path = './data/bpd_training.tfrecord'
label_col = 'FIOFS_TYPE'
model_path = './bpd_model'
n_classes = 2

# Read data from CSV to dataframe
df = pd.read_csv(
    "./feature_control/bpd_training_data.csv",
    skipinitialspace=True)

# Make the label column numeric (0 and 1), for use in our model
df[label_col] = np.where(df[label_col].str.contains("F|S", regex=True), 1, 0)

# Get list of all columns from the dataset we will use for model input or
output.
# We will ignore the fnlwgt column in the dataset for training this model.
features_and_labels = [df.columns.values.tolist()]


## BPD TEST DATA (you have to modify the training data)

test_data_path = './data/bpd_test.tfrecord'
test_label_col = 'FIOFS_TYPE'
test_model_path = './bpd_model'
test_n_classes = 2
```

```
# Read data from CSV to dataframe
df_test = pd.read_csv(
    "./feature_control/bpd_test_data.csv",
    skipinitialspace=True)

# Make the label column numeric (0 and 1), for use in our model
df_test[test_label_col] = np.where(df_test[test_label_col].str.contains("F|S",
regex=True), 1, 0)

# Get list of all columns from the dataset we will use for model input or
output.
test_features_and_labels = df_test.columns.values.tolist()

write_df_as_tfrecord(df_test, test_data_path, test_features_and_labels)


## Create and train the classifier

import functools

# Write the records to disk as tf.Example protos in tf record file, for use in
model training
# and later for use by WIT.
write_df_as_tfrecord(df, tfrecord_path, features_and_labels)

# Create a feature spec for the classifier
feature_spec = create_feature_spec(df, features_and_labels)

# print feature_spec

# Create a list of just the input features the classifier will use (removing
the label feature)
features = [f for f in features_and_labels if f != label_col]

# Define and train the classifier
train_inpf = functools.partial(tfrecords_input_fn, tfrecord_path, feature_spec,
label_col)
classifier =
tf.estimator.LinearClassifier(feature_columns=create_feature_columns(features,
feature_spec),
                                            n_classes=n_classes)
classifier.train(train_inpf, steps=10000)

# Save the classifier to disk for serving
# Uses a parsing serving input receiver function so that it can classify from
serialized tf.Examples
# using the TensorFlow Serving Classify API.
```

```
serving_input_fn =
tf.estimator.export.build_parsing_serving_input_receiver_fn(feature_spec)
classifier.export_savedmodel(model_path, serving_input_fn)


## What-If Tool usage instructions (serve model, launch TensorBoard, configure
What-If Tool)

# sudo docker run -p 8500:8500 --mount
type=bind,source=/Users/mzhong/6.805-project/bpd_model,target=/models/my_model/
-e MODEL_NAME=my_model -t tensorflow/serving
# tensorboard --logdir .
# Navigate to
http://localhost:6006/#whatif&inferenceAddress=localhost%3A8500&modelName=my_mo
del
# Set examples path to ./data/bpd_test.tfrecord and click accept button
```

Appendix D: bpd_training_data.csv

Below is an example of some rows from *bpd_trainig_data.csv*, the modified data that we trainined our TensorFlow model on. In addition, the format of the test data was similar.

SEX	FIO_DATE	PRIORS	COMPLEXION	FIOFS_TYPE	FIOFS_REASONS	AGE_AT_FIO_CORRECTED	DESCRIPTION	DIST	OFFICER_ID
MALE	05/29/2012 12:00:00 AM	YES	Light	IOF	INVESTIGATE, PERSON	MIDDLE	W(White)	E5	officer 80200
MALE	12/04/2012 12:00:00 AM	YES	Med	OF	TRESPASSING	MIDDLE	B(Black)	B2	officer 95177
MALE	06/09/2014 12:00:00 AM	YES	Med	IO	INVESTIGATE, PERSON	TEENS	B(Black)	B3	officer 62601
MALE	05/17/2012 12:00:00 AM	YES	Med	IO	INVESTIGATE, PERSON	MIDDLE	H(Hispanic)	B2	officer 98663
MALE	05/24/2012 12:00:00 AM	YES	Light	IO	INVESTIGATE, PERSON	TWENTIES	W(White)	D4	officer 11106
MALE	08/05/2014 12:00:00 AM	YES	NO DATA ENTERED	OF	DISTURBANCE	THIRTIES	W(White)	D4	officer 102680
MALE	01/09/2012 12:00:00 AM	YES	Light	IOF	VAL	TWENTIES	H(Hispanic)	B2	officer 11804

FEMALE	10/02/2012 12:00:00 AM	YES	Dark	OFS	PROSTITUTION, COMMON NIGHT WALKER	THIRTIES	B(Black)	C11	officer 108890
MALE	10/12/2011 12:00:00 AM	YES	Light	OF	INVESTIGATE, PERSON	TWENTIES	W(White)	D4	officer 91893
MALE	07/10/2012 12:00:00 AM	YES	OTHER	IO	INVESTIGATE, PERSON	MIDDLE	B(Black)	D4	officer 12011

Appendix E: Data Breakdown Script for Demographic Parity Analysis

```
import csv
import random
import sys
import pandas
import re
import functools

"""
    This script takes in a csv file with modified BPD FIO data.
    It outputs a csv file that lists the counts for each category for each
feature (column).
    In addition, for each count, it lists how many were searched or frisked,
and how many were not.

    Used for the 6.805 Project (Dheekshita Kumar, Emily Tang, Mary Zhong).
    Written by Mary Zhong.
"""

def check_frisked_counts(current, outcome, frisked_counts, not_frisked_counts):
    """
    Given an individual with value current for a feature column,
    with outcome = True if they were frisked or searched and 0 if not,
    this function correctly adds counts to frisked_counts or
not_frisked_counts.
    """
    if re.match(r"[SF]", outcome):
        if current not in frisked_counts:
            frisked_counts[current] = 1
        else:
            frisked_counts[current] = frisked_counts[current] + 1
    else:
        if current not in not_frisked_counts:
            not_frisked_counts[current] = 1
        else:
            not_frisked_counts[current] = not_frisked_counts[current] + 1


with open("./test_control/bpd_test_data_original.csv") as file,
open('data_breakdown.csv', mode='w') as output:
    file_copy = open("./test_control/bpd_test_data_original.csv")
    total = sum(1 for line in file_copy) - 1
```

```
    print("Total rows in file: {0}".format(total))
    reader = csv.DictReader(file, delimiter=",", quotechar='"',
quoting=csv.QUOTE_MINIMAL)
    writer = csv.writer(output, delimiter=',', quotechar='"',
quoting=csv.QUOTE_MINIMAL)

    # headers:
    columns = ["SEX", "FIO_DATE", "PRIORS", "COMPLEXION", "FIOFS_REASONS",
"AGE_AT_FIO_CORRECTED", "DESCRIPTION", "DIST", "OFFICER_ID"]

    data = {}
    frisked = {}
    not_frisked = {}
    for col in columns:
        data[col] = {}
        frisked[col] = {}
        not_frisked[col] = {}

    for row in reader:
        outcome = row["FIOFS_TYPE"]
        for col in columns:
            counts = data[col]
            frisked_counts = frisked[col]
            not_frisked_counts = not_frisked[col]

            current = row[col]

            if current not in counts:
                counts[current] = 1
            else:
                counts[current] = counts[current] + 1

            check_frisked_counts(current, outcome, frisked_counts,
not_frisked_counts)

            data[col] = counts
            frisked[col] = frisked_counts
            not_frisked[col] = not_frisked_counts

    for col in columns:
        # 50,000 rows
        data_total = functools.reduce(lambda x,y: x + y, data[col].values())
        writer.writerow([col, data_total])

        counts = []

        for key in data[col]:
            counts.append(["    " + key, data[col][key]])
```

```
            if key in frisked[col]:
                counts.append(["        frisked or searched",
frisked[col][key]])
            if key in not_frisked[col]:
                counts.append(["        not frisked nor searched",
not_frisked[col][key]])

        for row in counts:
            writer.writerow(row)

        writer.writerow([])
```

Appendix F: data_breakdown.csv

SEX	50000
FEMALE	5861
frisked or searched	82
not frisked nor searched	5779
MALE	44050
frisked or searched	1091
not frisked nor searched	42959
UNKNOWN	89
not frisked nor searched	89

AGE_AT_FIO_CORRECTED	50000
TEENS	9082
frisked or searched	254
not frisked nor searched	8828
THIRTIES	9283
frisked or searched	219
not frisked nor searched	9064
TWENTIES	23071
frisked or searched	541
not frisked nor searched	22530
MIDDLE	7977
frisked or searched	149
not frisked nor searched	7828
SENIOR	587
frisked or searched	10
not frisked nor searched	577

DESCRIPTION	50000
B(Black)	29175
frisked or searched	678
not frisked nor searched	28497
H(Hispanic)	6550
frisked or searched	194
not frisked nor searched	6356
W(White)	11426
frisked or searched	257
not frisked nor searched	11169
M(Middle Eastern or East Indian)	153
frisked or searched	2
not frisked nor searched	151
NO DATA ENTERED	2049
frisked or searched	35
not frisked nor searched	2014
A(Asian or Pacific Islander)	430
frisked or searched	5
not frisked nor searched	425
UNKNOWN	189
frisked or searched	2
not frisked nor searched	187
I(American Indian or Alaskan Native)	28
not frisked nor searched	28

PRIORS	50000

NO	5392
frisked or searched	150
not frisked nor searched	5242
YES	37163
frisked or searched	868
not frisked nor searched	36295
BLANK	6904
frisked or searched	140
not frisked nor searched	6764
UNKNOWN	541
frisked or searched	15
not frisked nor searched	526

COMPLEXION	50000
Light	10727
frisked or searched	294
not frisked nor searched	10433
Dark	9103
frisked or searched	247
not frisked nor searched	8856
Med	19922
frisked or searched	472
not frisked nor searched	19450
NO DATA ENTERED	7549
frisked or searched	117
not frisked nor searched	7432
Brown	1083
frisked or searched	11

not frisked nor searched	1072
White	170
frisked or searched	1
not frisked nor searched	169
Ruddy	34
frisked or searched	1
not frisked nor searched	33
OTHER	817
frisked or searched	13
not frisked nor searched	804
Fair	585
frisked or searched	16
not frisked nor searched	569
Clear	10
frisked or searched	1
not frisked nor searched	9

Appendix G: What-If Tool Details

1. Performance and Fairness

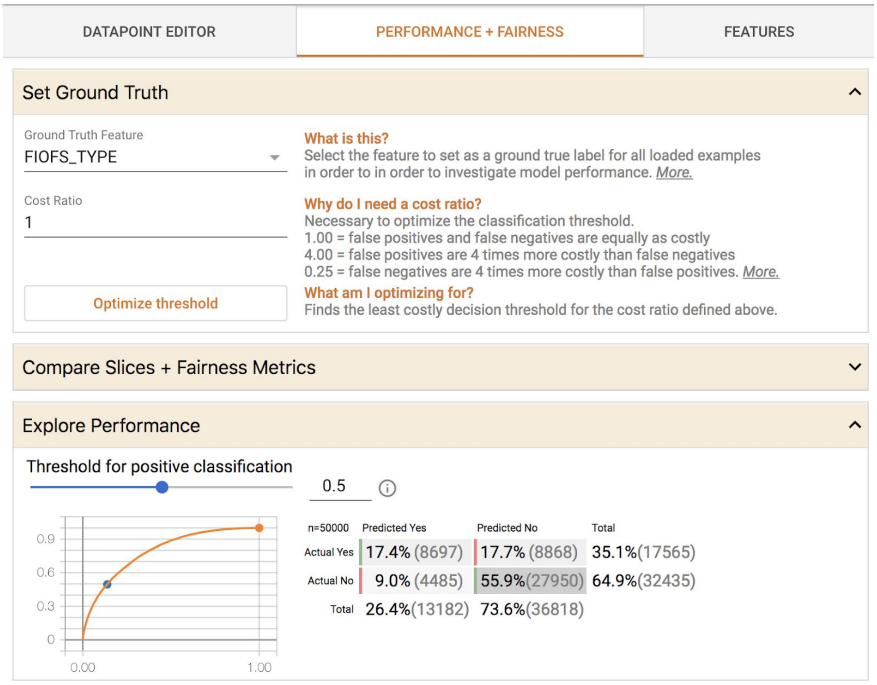


Figure 1: Performance and Fairness Tab of the Google What-If Tool

These tools can be found in the *Performance + Fairness* tab, shown above in figure 1. First, the top section asks for a *Ground Truth* feature. This is the feature that the results of the model will be compared against.

The results of the comparison between values in the Ground Truth column and the results from the model are shown in the the bottom section, *Explore Performance*. Here, the user can see a confusion matrix (on the bottom right in figure 1), that shows the true positive, true negative, false positive, and false negative percentages. For example, the following would be the interpretation of the confusion matrix from the above photo:

True positive: 17.4%

- The model accurately predicted that 8,697 individuals would be frisked or searched, which is 17.4% of the total dataset.

True negative: 55.9%

- The model accurately predicted that 27,950 individuals were not frisked or searched.

False positive: 9.0%

- The model inaccurately predicted that 4,485 individuals would be frisked or stopped, when their FIOSFS_TYPE did not include F or S. In other words, in the BPD FIO data, these individuals were not frisked or searched, but our model predicted that they would be .

False negative: 17.7%

- The model inaccurately predicted that 8,868 individuals were not frisked or searched, when the BPD FIO data showed that they actually were.

In addition to the confusion matrix, the Explore Performance section of this tab in the What-If Tool provides a ROC curve, which plots the true positive rate against the false positive rate.

2. Show Nearest Counterfactual

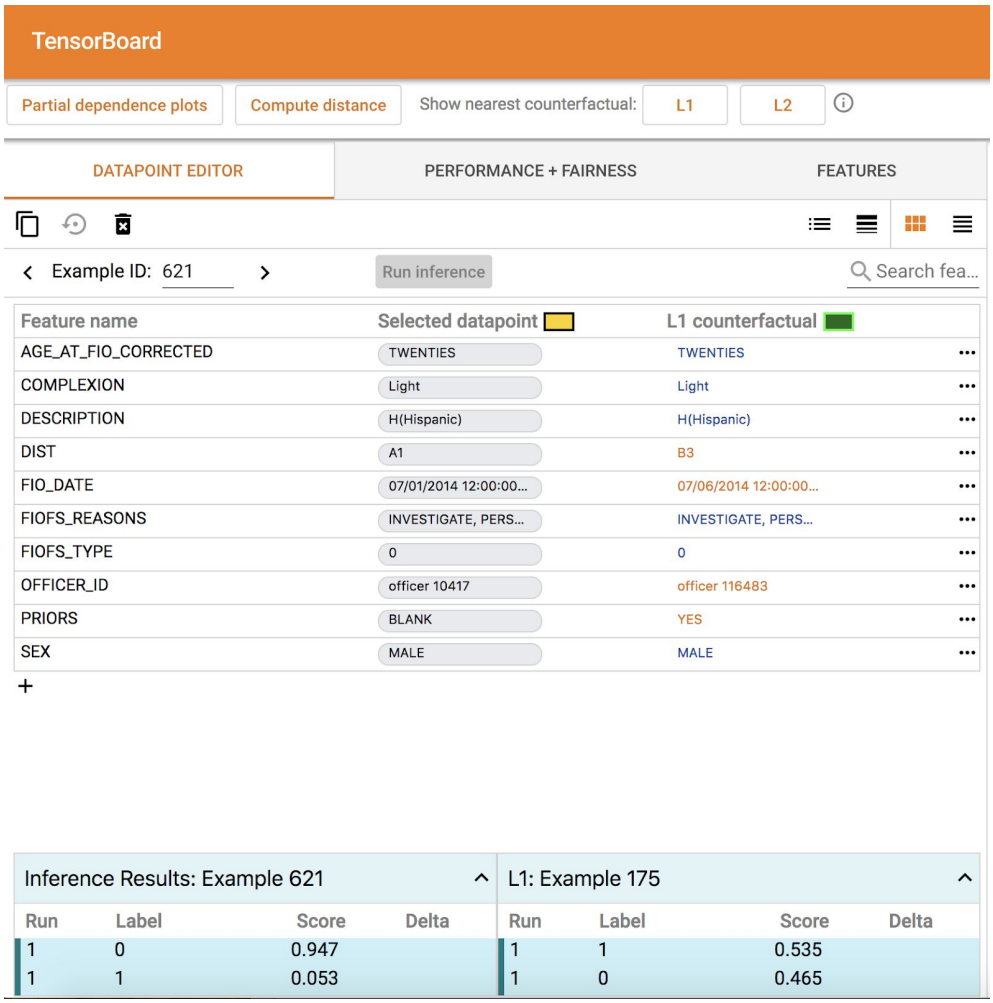


Figure 2: Data Point Editor Tab of the Google What-If Tool, Finding CounterFactual

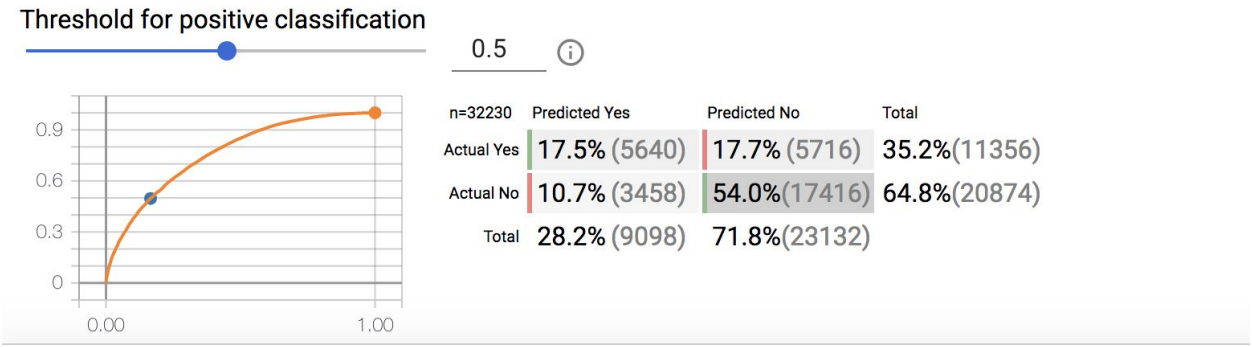
At the top of the image, there is a label *Show nearest counterfactual*, with the two options of L1 or L2. Clicking on a data point and then selecting L1 compares the selected example with its nearest neighbor from a different classification using L1 or L2 norm[29]. In the image above, we see that the selected data point (Example 621) was classified with a 0, and the L1 counterfactual data point (Example 175) was classified with a 1. We can also compare the values of the features of these two data points, and indeed see that they share many of the same values.

[29]Brownlee, Jason. "Gentle Introduction to Vector Norms in Machine Learning." *Machine Learning Mastery*. https://machinelearningmastery.com/vector-norms-machine-learning/.

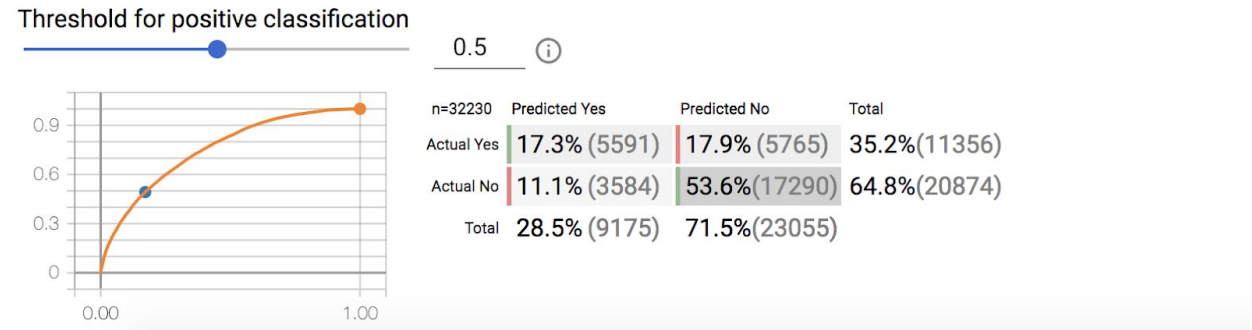
Appendix H: Results of Analysis - Screenshots from the What-If Tool

Feature Control (120,000 training records, rest of 32,230 are test records)

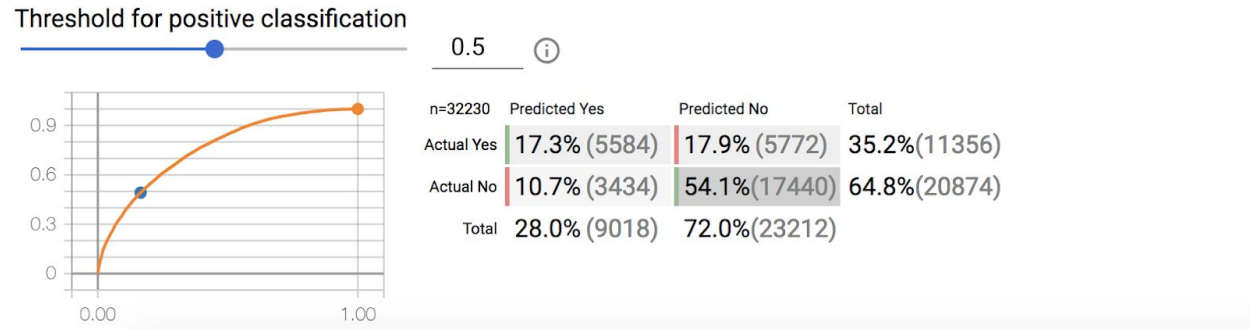
Baseline



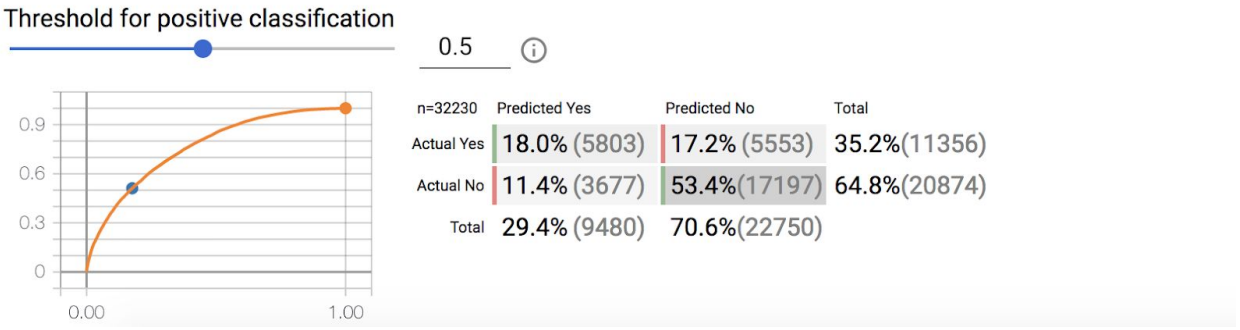
SEX = "SEX"



PRIORS = "PRIORS"

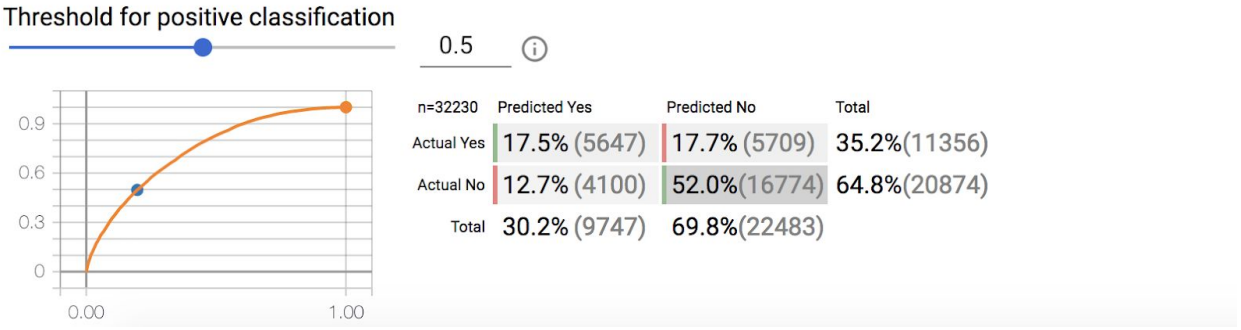


COMPLEXION = “COMPLEXION”



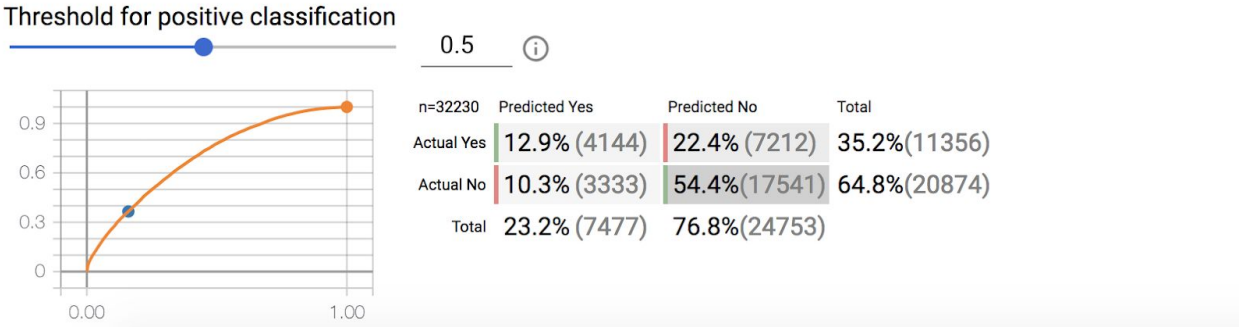
n=32230	Predicted Yes	Predicted No	Total
Actual Yes	18.0% (5803)	17.2% (5553)	35.2%(11356)
Actual No	11.4% (3677)	53.4%(17197)	64.8%(20874)
Total	29.4% (9480)	70.6%(22750)	

FIOFS_REASONS = “FIOFS_REASONS”



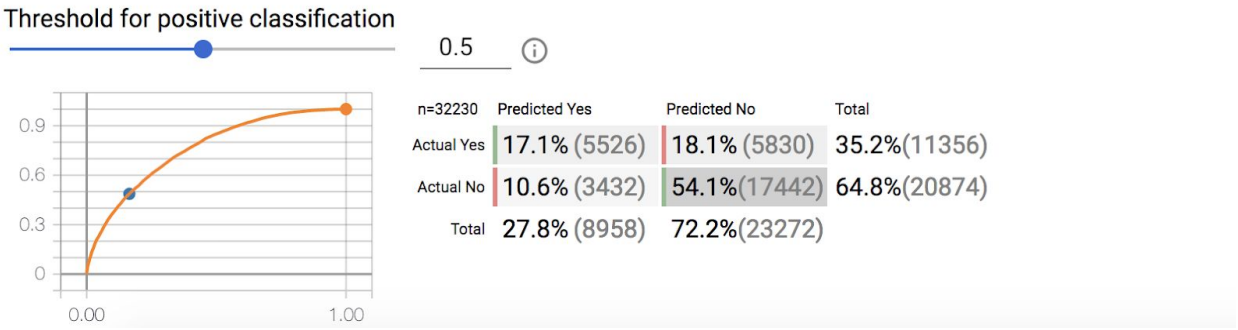
n=32230	Predicted Yes	Predicted No	Total
Actual Yes	17.5% (5647)	17.7% (5709)	35.2%(11356)
Actual No	12.7% (4100)	52.0%(16774)	64.8%(20874)
Total	30.2% (9747)	69.8%(22483)	

OFFICER_ID = “OFFICER_ID”

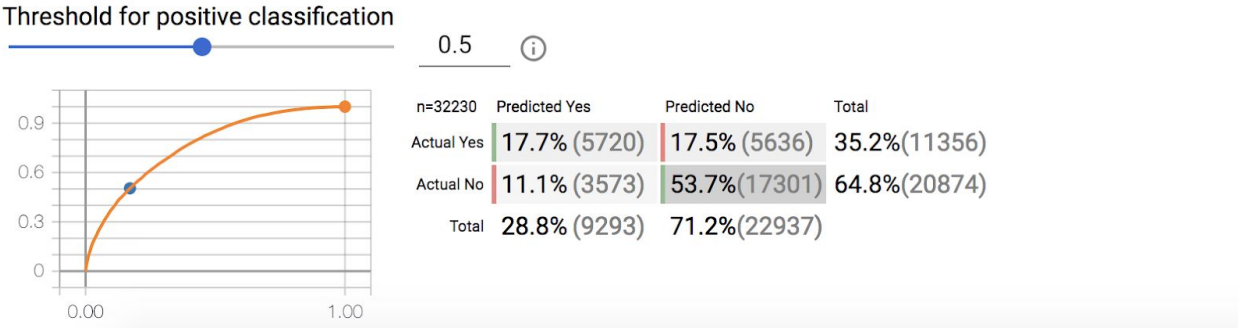


n=32230	Predicted Yes	Predicted No	Total
Actual Yes	12.9% (4144)	22.4% (7212)	35.2%(11356)
Actual No	10.3% (3333)	54.4%(17541)	64.8%(20874)
Total	23.2% (7477)	76.8%(24753)	

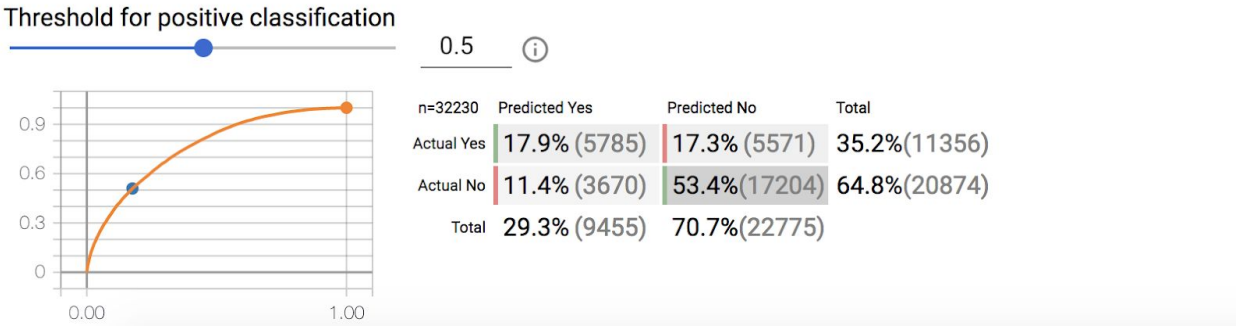
FIO_DATE = “FIO_DATE”



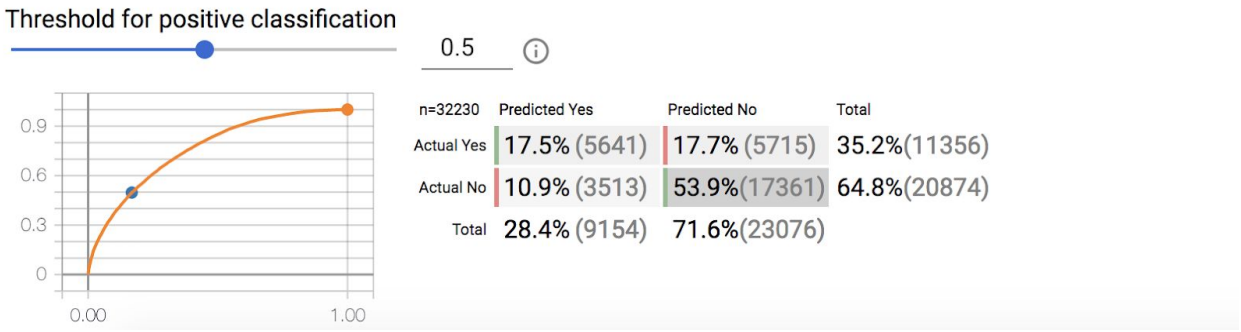
DESCRIPTION = "DESCRIPTION"



DIST = "DIST"

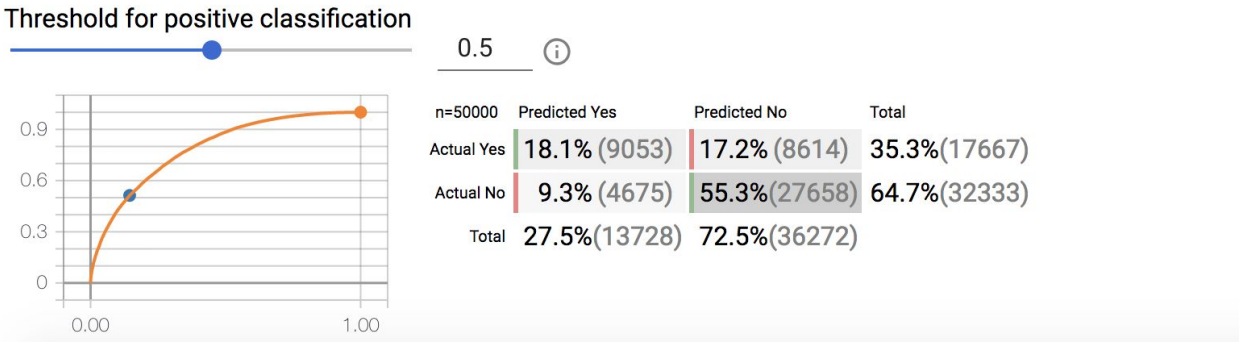


AGE_AT_FIO_CORRECTED = "AGE"



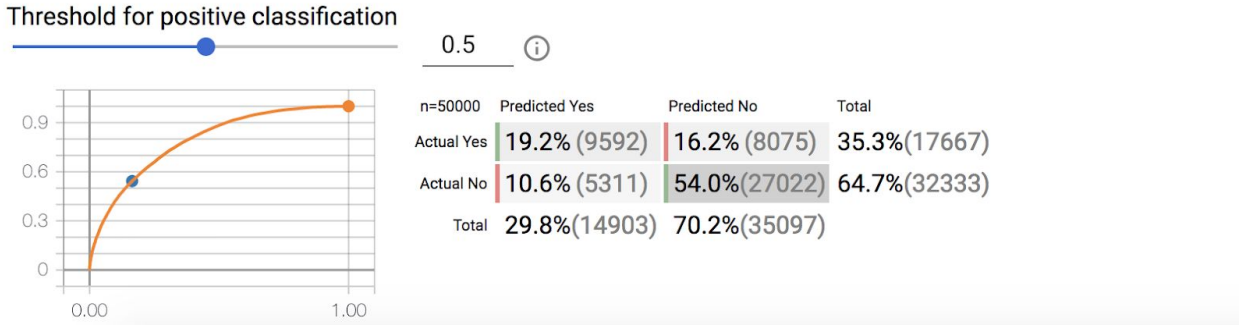
Test Control (all 152,230 are training data, the same randomly picked and modified 50,000 are test data)

Baseline



DESCRIPTION

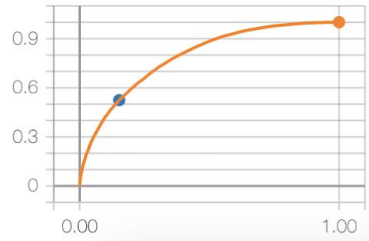
- A(Asian or Pacific Islander)



- B(Black)

Threshold for positive classification

0.5

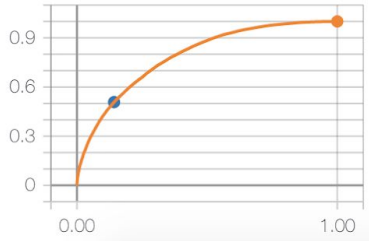


n=50000	Predicted Yes	Predicted No	Total
Actual Yes	18.5% (9265)	16.8% (8402)	35.3%(17667)
Actual No	9.8% (4924)	54.8%(27409)	64.7%(32333)
Total	28.4%(14189)	71.6%(35811)	

- H(Hispanic)

Threshold for positive classification

0.5

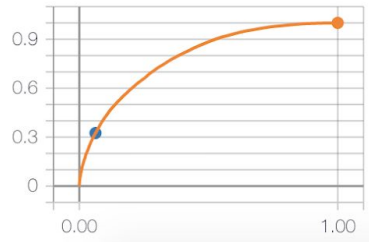


n=50000	Predicted Yes	Predicted No	Total
Actual Yes	17.9% (8967)	17.4% (8700)	35.3%(17667)
Actual No	9.2% (4607)	55.5%(27726)	64.7%(32333)
Total	27.1%(13574)	72.9%(36426)	

- I(American Indian or Alaskan Native)

Threshold for positive classification

0.5

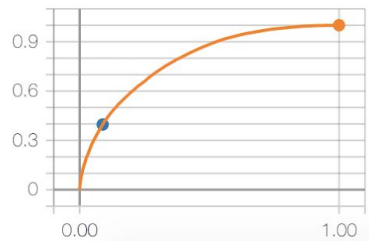


n=50000	Predicted Yes	Predicted No	Total
Actual Yes	11.5% (5734)	23.9%(11933)	35.3%(17667)
Actual No	4.0% (2018)	60.6%(30315)	64.7%(32333)
Total	15.5% (7752)	84.5%(42248)	

- M(Middle Eastern or East Indian)

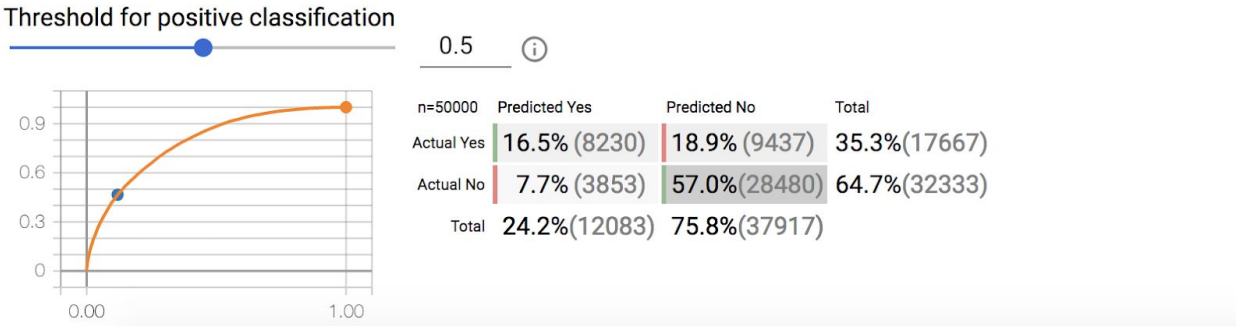
Threshold for positive classification

0.5



n=50000	Predicted Yes	Predicted No	Total
Actual Yes	14.0% (7003)	21.3%(10664)	35.3%(17667)
Actual No	5.7% (2869)	58.9%(29464)	64.7%(32333)
Total	19.7% (9872)	80.3%(40128)	

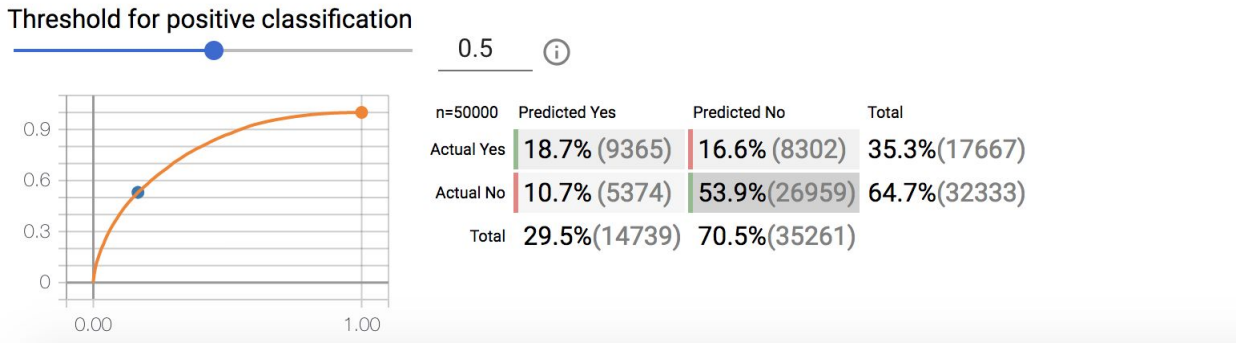
- W(White)



n=50000	Predicted Yes	Predicted No	Total
Actual Yes	16.5% (8230)	18.9% (9437)	35.3%(17667)
Actual No	7.7% (3853)	57.0%(28480)	64.7%(32333)
Total	24.2%(12083)	75.8%(37917)	

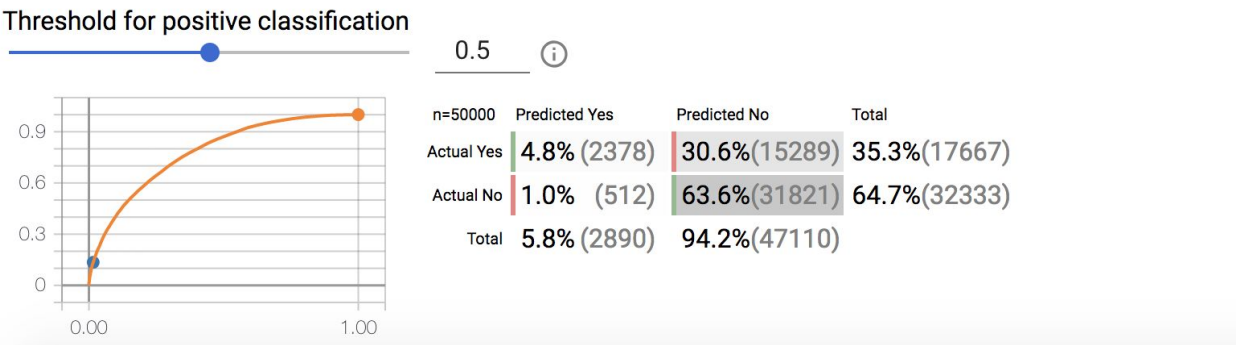
SEX

- MALE



n=50000	Predicted Yes	Predicted No	Total
Actual Yes	18.7% (9365)	16.6% (8302)	35.3%(17667)
Actual No	10.7% (5374)	53.9%(26959)	64.7%(32333)
Total	29.5%(14739)	70.5%(35261)	

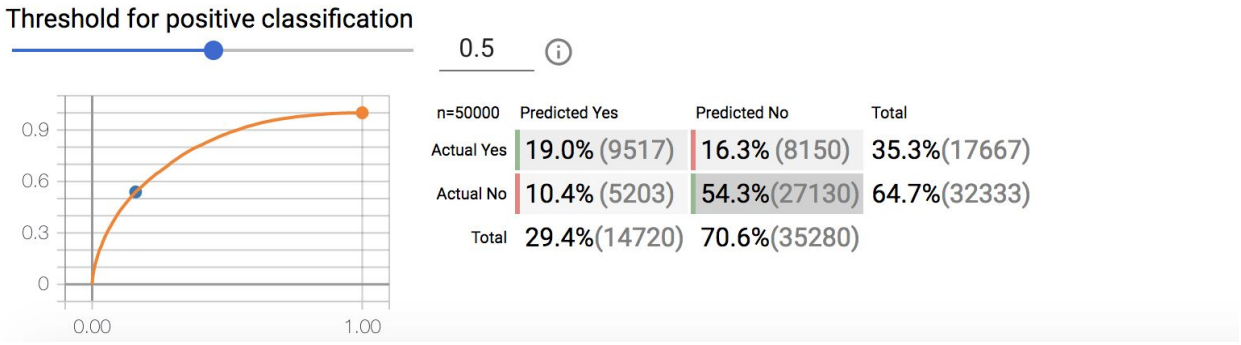
- FEMALE



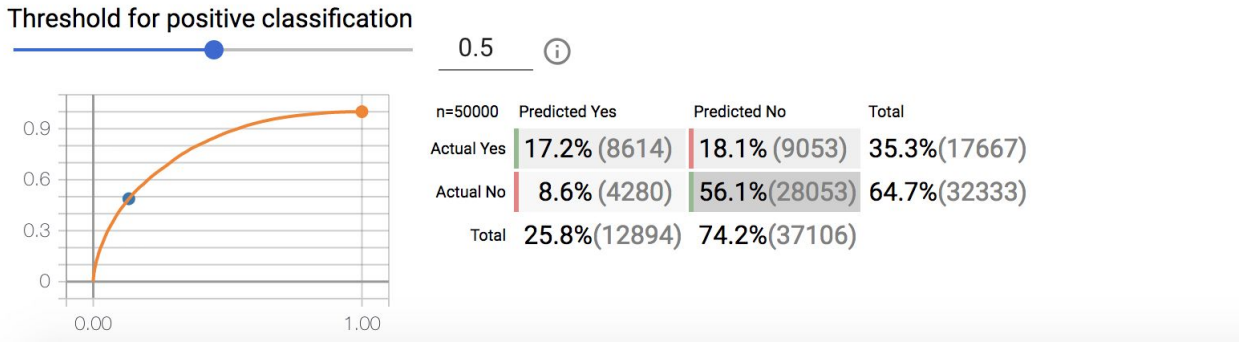
n=50000	Predicted Yes	Predicted No	Total
Actual Yes	4.8% (2378)	30.6%(15289)	35.3%(17667)
Actual No	1.0% (512)	63.6%(31821)	64.7%(32333)
Total	5.8% (2890)	94.2%(47110)	

PRIORS

- YES

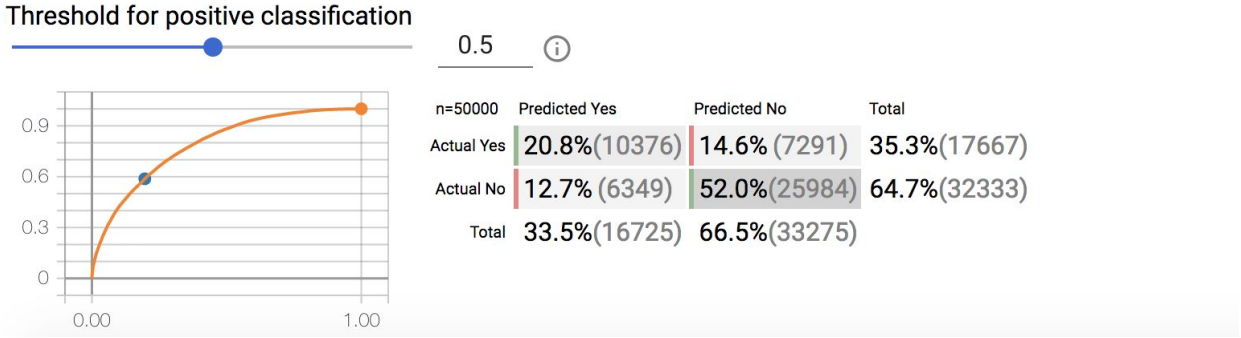


- NO



AGE_AT_FIO_CORRECTED

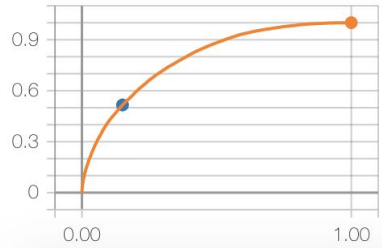
- TEENS



- TWENTIES

Threshold for positive classification

0.5

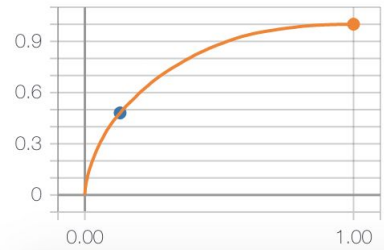


n=50000	Predicted Yes	Predicted No	Total
Actual Yes	18.2% (9111)	17.1% (8556)	35.3%(17667)
Actual No	9.7% (4871)	54.9%(27462)	64.7%(32333)
Total	28.0%(13982)	72.0%(36018)	

- THIRTIES

Threshold for positive classification

0.5

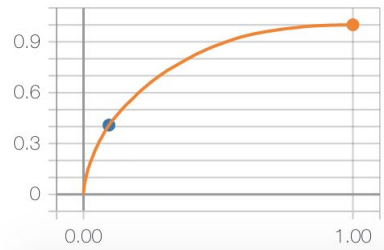


n=50000	Predicted Yes	Predicted No	Total
Actual Yes	17.0% (8485)	18.4% (9182)	35.3%(17667)
Actual No	8.5% (4239)	56.2%(28094)	64.7%(32333)
Total	25.4%(12724)	74.6%(37276)	

- MIDDLE

Threshold for positive classification

0.5

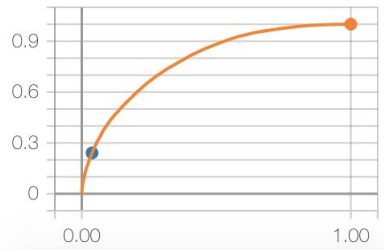


n=50000	Predicted Yes	Predicted No	Total
Actual Yes	14.4% (7214)	20.9%(10453)	35.3%(17667)
Actual No	6.1% (3051)	58.6%(29282)	64.7%(32333)
Total	20.5%(10265)	79.5%(39735)	

- SENIOR

Threshold for positive classification

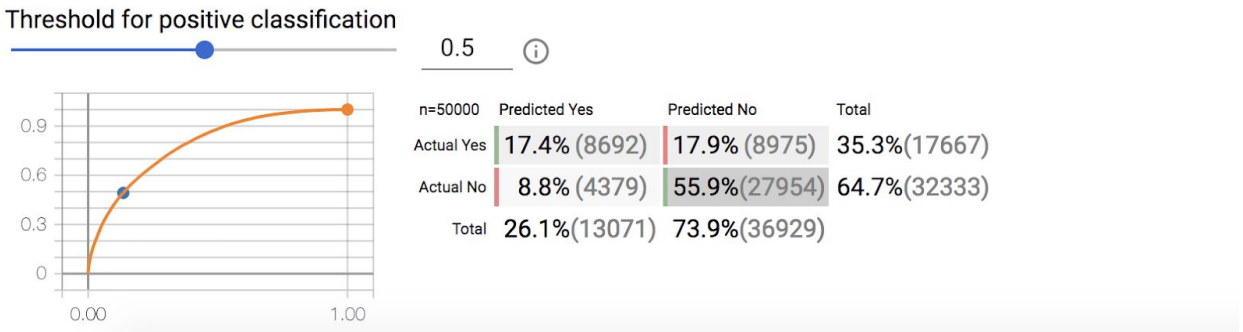
0.5



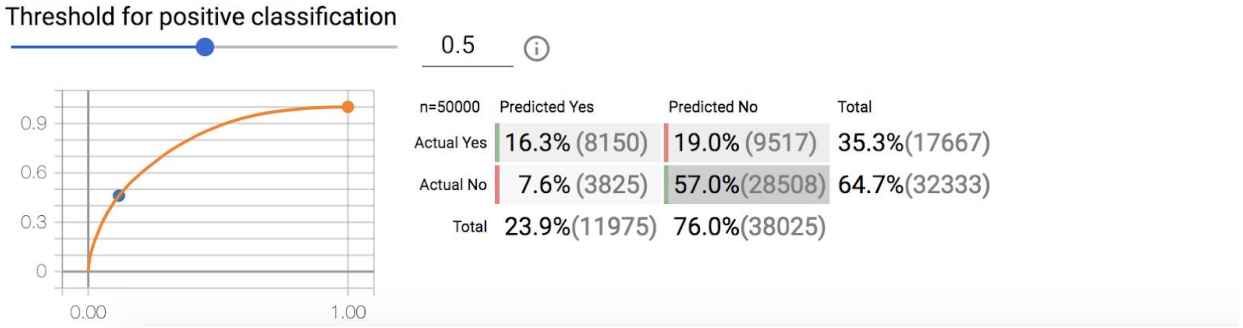
n=50000	Predicted Yes	Predicted No	Total
Actual Yes	8.5% (4252)	26.8%(13415)	35.3%(17667)
Actual No	2.5% (1232)	62.2%(31101)	64.7%(32333)
Total	11.0% (5484)	89.0%(44516)	

COMPLEXION

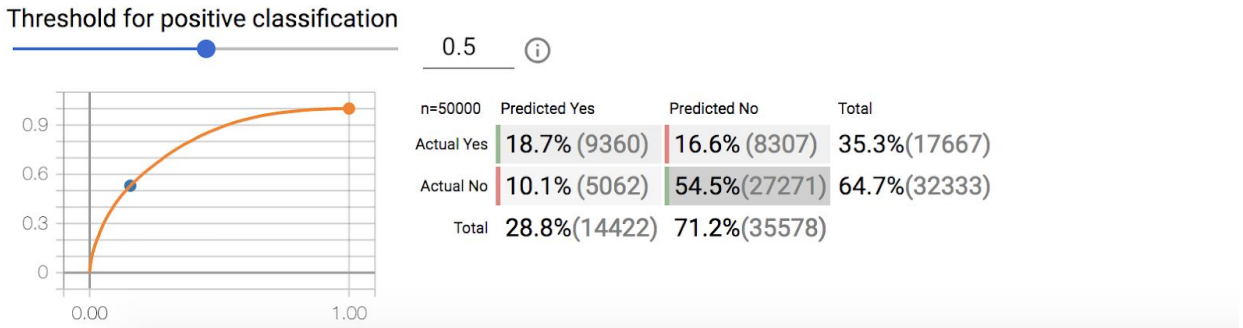
- Brown



- Clear



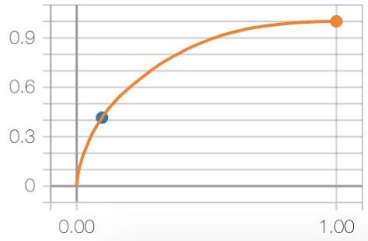
- Dark



- Fair

Threshold for positive classification

0.5

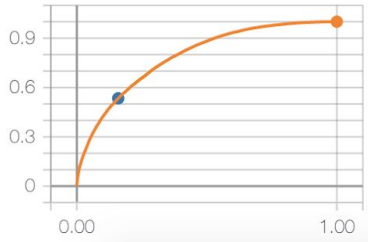


n=50000	Predicted Yes	Predicted No	Total
Actual Yes	14.7% (7342)	20.6%(10325)	35.3%(17667)
Actual No	6.3% (3145)	58.4%(29188)	64.7%(32333)
Total	21.0%(10487)	79.0%(39513)	

- Light

Threshold for positive classification

0.5

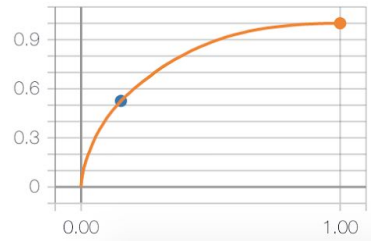


n=50000	Predicted Yes	Predicted No	Total
Actual Yes	18.9% (9429)	16.5% (8238)	35.3%(17667)
Actual No	10.3% (5142)	54.4%(27191)	64.7%(32333)
Total	29.1%(14571)	70.9%(35429)	

- Med

Threshold for positive classification

0.5

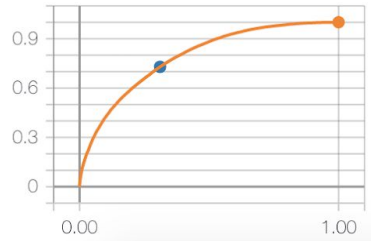


n=50000	Predicted Yes	Predicted No	Total
Actual Yes	18.6% (9288)	16.8% (8379)	35.3%(17667)
Actual No	9.9% (4968)	54.7%(27365)	64.7%(32333)
Total	28.5%(14256)	71.5%(35744)	

- Ruddy (only 34 original records)

Threshold for positive classification

0.5

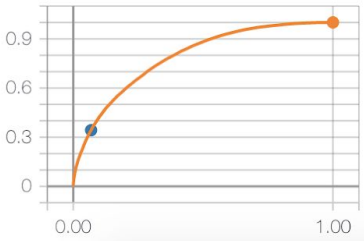


n=50000	Predicted Yes	Predicted No	Total
Actual Yes	25.8%(12884)	9.6% (4783)	35.3%(17667)
Actual No	20.1%(10047)	44.6%(22286)	64.7%(32333)
Total	45.9%(22931)	54.1%(27069)	

- White

Threshold for positive classification

0.5



n=50000	Predicted Yes	Predicted No	Total
Actual Yes	12.1% (6050)	23.2%(11617)	35.3%(17667)
Actual No	4.4% (2215)	60.2%(30118)	64.7%(32333)
Total	16.5% (8265)	83.5%(41735)	