Assessing and Mitigating Algorithmic Bias in Criminal Risk Scores

Lucius Bynum, Preethi Seshadri

19 November 2017

Introduction

Problem Statement

Data Description

ProPublica collected data on **TODO**

In the data set we look at, they considered only people who either "recidivated within two years of their crime or"recidivated in tow years, or had at least tow years outside of a correctional facility."

Data Dictionary

ProPublica did not provide a data dictionary explaining their variables. Through some manual exploration, we came up with the following descriptions for our best guess at what each variable measures.

Variable	Description
id	unque identifier for each individual
name	first and last name
first	first name
last	last name
compas_screening_date	date on which decile_score was given
sex	sex (male or female)
dob	date of birth
age	age in years
age_cat	age category (less than 25, 25-45, greater than 45)
race	race (African-American, Asian, Caucasian, Hispanic, Native American, Other)
juv_fel_count	juvenile felony count
decile_score	COMPAS Risk of Recidivism score from 1 to 10
juv_misd_count	juvenile misdemeanor count
juv_other_count	juvenile other offenses count
priors_count	prior offenses count
days_b_screening_arrest	number of days between COMPAS screening and arrest **TODO** negative values?
c_jail_in	jail entry date for original crime
c_jail_out	jail exit date for original crime
c_case_number	case number for original crime
c_offense_date	offense date of original crime
c_arrest_date	arrest date for original crime
c_days_from_compas	days between COMPAS screening and original crime offense date
c_charge_degree	charge degree of original crime
c_charge_desc	description of charge for original crime

Variable	Description
is_recid	binary indicator of recidivation (1=individual recidivated, 0=individual did not recidivate)
r_case_number	case number of follow-up crime
r_charge_degree	charge degree of follow-up crime
r_days_from_arrest	number of days between follow-up crime and arrest date **TODO** why negative value here?
r_offense_date	date of follow-up crime
r_charge_desc	description of charge for follow-up crime
r_jail_in	jail entry date for follow-up crime
r_jail_out	jail exit date for follow-up crime
violent_recid	values are all `NA`. This column is ignored.
is_voilent_recid	binary indicator of violent follow-up crime (1=follow-up crime was violent, 0=follow-up crime was non-violent)
vr_case_number	case number for violent follow-up crime
vr_charge_degree	charge degree for violent follow-up crime
vr_offense_date	date of offense for violent follow-up crime
vr_charge_desc	description of charge for violent follow-up crime
type_of_assessment	the type of COMPAS score given for decile_score (here all values are Risk of Recidivism)
decile_score.1	repeat column of decile_score
score_text	ProPublica-defined category of decile_score (High=8-10, Medium=5-7, Low=1-4)
screening_date	repeat column of compas_screening_date
v_type_of_assessment	the type of COMPAS score given for v_decile_score (here all values are Risk_of_Violence)
v_decile_score	COMPAS Risk of Violence score from 1 to 10
v_score_text	ProPublica-defined category of v_decile_score (High=8-10, Medium=5-7, Low=1-4)
v_screening_date	date on which v_decile_score was given
in_custody	date on which individual was brought into custody
out_custody	date on which individual was released from custody
priors_count.1	repeat column of priors_count
start	**TODO**
end	**TODO**
event	**TODO**

ProPublica obtained this data with the goal of analyzing Northpointe Inc.'s commercial recidivism modeling tool – COMPAS. Aggregating data from public records, they collected data on 18,610 individuals who received COMPAS scores from 2013 to 2014, including demographic information, public criminal records, and incarceration records.

How are COMPAS scores used?

ProPublica describes that at least in Broward County, they "primarily [use] the score to determine whether to release or detain a defendant before his or her trial." 11,757 of the individuals in the database had their COMPAS scores used to assess whether or not they should be released before their trial.

What are COMPAS scores?

There are three types of COMPAS score. Each measures a type of 'risk' associated with a criminal re-offending in some way on a scale of 1 (low) to 10 (high). As ProPublica describes, these include

- *Risk of Recidivism*: ProPublica defines this as the person in question committing a "criminal offense that [results] in a jail booking and [takes] place after the crime for which the person was COMPAS scored." Northpointe hopes to use this score to predict "a new misdemeanor or felony offense within two years of the COMPAS administration date."

- *Risk of Violence*: They use the FBI definition of violent crime:

  In the FBI's Uniform Crime Reporting (UCR) Program, violent crime is composed of four offenses: murder and nonnegligent manslaughter, forcible rape, robbery, and aggravated assault. Violent crimes are defined in the UCR Program as those offenses which involve force or threat of force. - ucr.fbi.gov

- *Risk of Failure to Appear*: As evidenced by the name, this describes a failure to appear at the court hearing.

Exploratory Data Analysis

We first load in the data provided by ProPublica, and take a quick look at summaries of each variable to get a sense of distribution and potential outliers. We'll look at half of the columns at a time to keep things manageable.

```
compas_data <- read.csv('compas-scores-two-years.csv')
summary(compas_data[1:(ncol(compas_data)/2)])
```

```
##        id                        name                first
##  Min.   :    1   anthony smith    :   3   michael    : 149
##  1st Qu.: 2735   angel santiago   :   2   christopher: 109
##  Median : 5510   anthony gonzalez :   2   james      :  84
##  Mean   : 5501   anthony louis    :   2   anthony    :  83
##  3rd Qu.: 8246   brandon whitfield:   2   robert     :  76
##  Max.   :11001   carlos vasquez   :   2   john       :  74
##                  (Other)          :7201   (Other)    :6639
##        last       compas_screening_date     sex              dob
##  williams:  83   2013-02-20:  32       Female:1395   1987-02-04:   5
##  johnson :  76   2013-03-20:  32       Male  :5819   1987-12-21:   5
##  brown   :  68   2013-02-07:  31                     1989-04-27:   5
##  smith   :  65   2013-04-20:  30                     1989-08-31:   5
##  jones   :  57   2013-01-03:  29                     1990-02-22:   5
##  davis   :  46   2013-04-25:  28                     1990-05-02:   5
##  (Other) :6819   (Other)   :7032                     (Other)   :7184
##       age                    age_cat                      race
##  Min.   :18.0   25 - 45        :4109   African-American:3696
##  1st Qu.:25.0   Greater than 45:1576   Asian           :  32
##  Median :31.0   Less than 25   :1529   Caucasian       :2454
##  Mean   :34.8                          Hispanic        : 637
##  3rd Qu.:42.0                          Native American :  18
##  Max.   :96.0                          Other           : 377
##
##  juv_fel_count     decile_score   juv_misd_count   juv_other_count
##  Min.   : 0.000   Min.   : 1.00   Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 0.000   1st Qu.: 2.00   1st Qu.: 0.000   1st Qu.: 0.000
##  Median : 0.000   Median : 4.00   Median : 0.000   Median : 0.000
##  Mean   : 0.067   Mean   : 4.51   Mean   : 0.091   Mean   : 0.109
##  3rd Qu.: 0.000   3rd Qu.: 7.00   3rd Qu.: 0.000   3rd Qu.: 0.000
##  Max.   :20.000   Max.   :10.00   Max.   :13.000   Max.   :17.000
##
##   priors_count   days_b_screening_arrest                c_jail_in
##  Min.   : 0.00   Min.   :-414.0                                : 307
```

```
##  1st Qu.: 0.00   1st Qu.:  -1.0          2013-01-01 01:31:55:   1
##  Median : 2.00   Median :  -1.0          2013-01-01 03:16:15:   1
##  Mean   : 3.47   Mean   :   3.3          2013-01-01 03:28:03:   1
##  3rd Qu.: 5.00   3rd Qu.:   0.0          2013-01-01 04:17:22:   1
##  Max.   :38.00   Max.   :1057.0          2013-01-01 04:29:04:   1
##                  NA's   :307             (Other)            :6902
##               c_jail_out         c_case_number     c_offense_date
##                    : 307                :  22             :1159
##  2013-09-12 10:31:00:   3   00004068CF10A:   1   2013-01-14:  26
##  2013-09-14 05:58:00:   3   00022077MM10A:   1   2013-02-22:  26
##  2013-09-28 02:10:00:   3   01004839CF10A:   1   2013-03-01:  24
##  2013-02-06 10:01:51:   2   01006487CF10D:   1   2013-01-11:  23
##  2013-06-13 10:32:00:   2   01007205MM10A:   1   2013-02-16:  23
##  (Other)            :6894   (Other)      :7187   (Other)   :5933
##     c_arrest_date  c_days_from_compas c_charge_degree
##            :6077   Min.   :   0       F:4666
##  2013-02-06:   9   1st Qu.:   1       M:2548
##  2013-03-22:   8   Median :   1
##  2013-05-15:   8   Mean   :  58
##  2013-01-10:   7   3rd Qu.:   2
##  2013-01-11:   7   Max.   :9485
##  (Other)   :1098   NA's   :22
##                        c_charge_desc     is_recid           r_case_number
##  Battery                       :1156   Min.   :0.000                :3743
##  arrest case no charge         :1137   1st Qu.:0.000   13000349MM10A:   1
##  Possession of Cocaine         : 474   Median :0.000   13000445MM20A:   1
##  Grand Theft in the 3rd Degree: 425   Mean   :0.481   13000677MM20A:   1
##  Driving While License Revoked: 200   3rd Qu.:1.000   13000758MM30A:   1
##  Driving Under The Influence  : 135   Max.   :1.000   13000785MM30A:   1
##  (Other)                       :3687                   (Other)      :3466
```

Here we notice there may be large outliers in many of the crime count variables, such as `juv_fel_count`, `juv_misd_count`, `juv_other_count`, and `priors_count`. We expect these are simply accurate observations corresponding to individuals with high numbers of prior offenses. Thus we will not remove these individuals from the data but will be aware of them as potential influential points when we later fit any models. We also note that the values for the 'days_from' variables are quite variable which may be relevant if we use those variables in later analysis. Looking at the second half of the columns, we have:

```r
summary(compas_data[(ncol(compas_data)/2 + 1):ncol(compas_data)])
```

```
##  r_charge_degree r_days_from_arrest    r_offense_date
##         :3743    Min.   : -1                 :3743
##  (M1)   :1201    1st Qu.:  0        2014-12-08:  12
##  (M2)   :1107    Median :  0        2015-01-28:  11
##  (F3)   : 892    Mean   : 20        2014-09-15:  10
##  (F2)   : 168    3rd Qu.:  1        2014-10-17:  10
##  (F1)   :  51    Max.   :993        2015-02-10:  10
##  (Other):  52    NA's   :4898       (Other)   :3418
##                            r_charge_desc       r_jail_in
##                                   :3801             :4898
##  Driving License Suspended        : 258   2014-05-27:   9
##  Possess Cannabis/20 Grams Or Less: 253   2013-11-22:   8
##  Resist/Obstruct W/O Violence     : 201   2014-06-05:   8
##  Battery                          : 192   2014-07-10:   8
##  Operating W/O Valid License      : 172   2014-10-17:   8
```

```
## (Other)                          :2337   (Other)   :2275
##       r_jail_out   violent_recid  is_violent_recid       vr_case_number
##           :4898   Mode:logical   Min.   :0.000                 :6395
## 2014-02-18:   9   NA's:7214      1st Qu.:0.000    13001383CF10A:   1
## 2014-12-09:   9                  Median :0.000    13001876CF10A:   1
## 2015-05-15:   9                  Mean   :0.114    13002119CF10A:   1
## 2013-11-13:   8                  3rd Qu.:0.000    13002546CF10A:   1
## 2014-07-11:   8                  Max.   :1.000    13003421CF10A:   1
## (Other)   :2273                                   (Other)      : 814
## vr_charge_degree   vr_offense_date                        vr_charge_desc
##        :6395              :6395                                  :6395
## (M1)   : 344     2015-08-15:   6   Battery                         : 329
## (F3)   : 228     2013-11-14:   4   Battery on Law Enforc Officer :  38
## (F2)   : 162     2014-02-18:   4   Felony Battery (Dom Strang)   :  38
## (F1)   :  38     2014-10-29:   4   Aggravated Assault W/Dead Weap:  37
## (M2)   :  19     2014-12-26:   4   Aggrav Battery w/Deadly Weapon:  34
## (Other):  28     (Other)   : 797   (Other)                         : 343
##          type_of_assessment decile_score.1   score_text
## Risk of Recidivism:7214     Min.   : 1.00   High  :1403
##                             1st Qu.: 2.00   Low   :3897
##                             Median : 4.00   Medium:1914
##                             Mean   : 4.51
##                             3rd Qu.: 7.00
##                             Max.   :10.00
##
##    screening_date       v_type_of_assessment v_decile_score
## 2013-02-20:  32   Risk of Violence:7214      Min.   : 1.00
## 2013-03-20:  32                              1st Qu.: 1.00
## 2013-02-07:  31                              Median : 3.00
## 2013-04-20:  30                              Mean   : 3.69
## 2013-01-03:  29                              3rd Qu.: 5.00
## 2013-04-25:  28                              Max.   :10.00
## (Other)   :7032
## v_score_text    v_screening_date      in_custody       out_custody
## High  : 714   2013-02-20:  32             : 236             : 236
## Low   :4761   2013-03-20:  32   2013-02-22:  20   2020-01-01:  61
## Medium:1739   2013-02-07:  31   2013-12-12:  20   2013-05-14:  25
##               2013-04-20:  30   2014-01-04:  20   2014-02-04:  24
##               2013-01-03:  29   2014-01-22:  20   2013-11-26:  23
##               2013-04-25:  28   2013-01-27:  19   2013-02-15:  21
##               (Other)   :7032   (Other)   :6879   (Other)   :6824
## priors_count.1      start           end           event
## Min.   : 0.00   Min.   :  0.0   Min.   :   0   Min.   :0.000
## 1st Qu.: 0.00   1st Qu.:  0.0   1st Qu.: 148   1st Qu.:0.000
## Median : 2.00   Median :  0.0   Median : 530   Median :0.000
## Mean   : 3.47   Mean   : 11.5   Mean   : 553   Mean   :0.383
## 3rd Qu.: 5.00   3rd Qu.:  1.0   3rd Qu.: 914   3rd Qu.:1.000
## Max.   :38.00   Max.   :937.0   Max.   :1186   Max.   :1.000
##
```

With the second half we have similar characteristics as before. We remove the `violent_recid` column given that all values are `NA` (as mentioned in the data dictionary). Apart from that column, we make no other changes.

ProPublica's Bias-Assessment Model

This section is meant to recreate ProPublica's logistic regression model for assessing bias in COMPAS scores. The code and explanations in this section are adapted from their published methodlogy and code.

Using the same data set, they first filter rows based on the following criteria.

1. consider only individuals with a COMPAS score
2. assure the COMPAS score corresponds to the correct crime i.e. the score was given within 30 days of the arrest
3. do not include ordinary traffic offenses

Next we use their code to perform this filtering:

```
# code from https://github.com/propublica/compas-analysis
df <- compas_data %>%
  select(age, c_charge_degree, race, age_cat, score_text, sex, priors_count, days_b_screening_arrest, d
  filter(days_b_screening_arrest <= 30) %>%
  filter(days_b_screening_arrest >= -30) %>%
  filter(is_recid != -1) %>%
  filter(c_charge_degree != "O") %>%
  filter(score_text != 'N/A')
nrow(df)
```

```
## [1] 6172
```

In order to use this data to assess racial bias in scoring, the ProPublica analysts first create several factor variables from the existing columns.

```
# code from https://github.com/propublica/compas-analysis
df <- mutate(df, crime_factor = factor(c_charge_degree)) %>%
      mutate(age_factor = as.factor(age_cat)) %>%
      within(age_factor <- relevel(age_factor, ref = "25 - 45")) %>%
      mutate(race_factor = factor(race)) %>%
      within(race_factor <- relevel(race_factor, ref = "Caucasian")) %>%
      mutate(gender_factor = factor(sex, labels= c("Female","Male"))) %>%
      within(gender_factor <- relevel(gender_factor, ref = "Male")) %>%
      mutate(score_factor = factor(score_text != "Low", labels = c("LowScore","HighScore")))
```

For the `age_factor` they make "25 - 45" the reference level, for `race_factor` "Caucasian" is the reference level, for `gender_factor` "Male" is the reference level.

Next they fit a logistic regression model to predict `score_factor` from the other variables.

```
# code from https://github.com/propublica/compas-analysis
pp_model <- glm(
  score_factor ~ gender_factor + age_factor + race_factor + priors_count + crime_factor + two_year_reci
  family="binomial",
  data=df
)
summary(pp_model)
```

```
##
## Call:
## glm(formula = score_factor ~ gender_factor + age_factor + race_factor +
##     priors_count + crime_factor + two_year_recid, family = "binomial",
##     data = df)
##
## Deviance Residuals:
```

```
##    Min      1Q  Median      3Q     Max
## -2.997  -0.792  -0.330   0.812   2.602
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -1.5255     0.0785  -19.43  < 2e-16 ***
## gender_factorFemale          0.2213     0.0795    2.78  0.00539 **
## age_factorGreater than 45   -1.3556     0.0991  -13.68  < 2e-16 ***
## age_factorLess than 25       1.3084     0.0759   17.23  < 2e-16 ***
## race_factorAfrican-American  0.4772     0.0693    6.88  5.9e-12 ***
## race_factorAsian            -0.2544     0.4782   -0.53  0.59472
## race_factorHispanic         -0.4284     0.1281   -3.34  0.00083 ***
## race_factorNative American   1.3942     0.7661    1.82  0.06878 .
## race_factorOther            -0.8263     0.1621   -5.10  3.4e-07 ***
## priors_count                 0.2689     0.0111   24.22  < 2e-16 ***
## crime_factorM               -0.3112     0.0665   -4.68  2.9e-06 ***
## two_year_recid               0.6859     0.0640   10.71  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8483.3  on 6171  degrees of freedom
## Residual deviance: 6168.4  on 6160  degrees of freedom
## AIC: 6192
##
## Number of Fisher Scoring iterations: 5
```

Calculating Relative Risk by Demographic

Next they compute how much more likely different demographic groups are to receive a higher score than others. The logistic regression model allows us to measure this difference after correcting for the other variables included in the model. The quantity ProPublica uses to compare black defendants to white defendants (or men to women, old defendants to young defendants, etc.) is called **relative risk**. ProPublica does not explain where this quantity comes from in their analysis, so we'll provide some quick background on logistic regression to justify the calculation. The following explanation is inspired by USC professor Sandy Eckel's slides here.

Logistic regression models a linear relationship between the log odds ratio for the probability of interest and the given predictor variables. An odds ratio measures the odds of success

$$\texttt{odds ratio} = \frac{\texttt{probability of success}}{\texttt{probability of failure}} = \frac{P}{1-P}$$

where P is the probability of success. The log odds ratio is simply the log of this quantity. Thus the logistic regression model for observation x_i is

$$\log(\frac{P_{x_i}}{1-P_{x_i}}) = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}$$

where the probability of success for x_i is P_{x_i}, and we have p predictors with corresponding coeffecients β_j and observed values x_{ij} for $j = 1, \ldots, p$. Given that our model here uses categorical predictors (factors), the coeffecients we estimate give us the **change in log odds** for the corresponding variable. Thus if we let P_{x_i}

be the probability that individual x_i gets a high COMPAS score, then with coefficent β_1 for `gender_factor`, we would have

$$\beta_0 : \text{the log odds of getting a high COMPAS score for men}$$

$$\beta_1 : \text{the difference in log odds of getting a high COMPAS score for women compared to men}$$

The important observation here is that because men are the reference level for the `gender_factor` categorical variable, β_1 measures a **difference** relative to men. Thus if we want to answer the question, "How much more likely are women to get a high COMPAS score than men?" we'll want to use

$$\beta_0 + \beta_1 : \text{the log odds of getting a high COMPAS score for women}$$

to get the comparison. One other observation will also be helpful to calculate relative risk. We solve for P_{x_i} as follows.

$$\log(\frac{P}{1-P}) = x \rightarrow P = \frac{e^x}{1+e^x} = \texttt{sigmoid}(x)$$

Thus we calculate relative risk for the categorical variable corresponding to β_1 as:

$$\texttt{relative risk} = \frac{P_1}{P_2} = \frac{\texttt{sigmoid}(\beta_0 + \beta_1)}{\texttt{sigmoid}(\beta_0)}$$

ProPublica computes relative risk to compare blacks to whites, men to women, and people under 25 to middle-aged people in terms of COMPAS scores. They get the following results.

```
# code adapted from https://github.com/propublica/compas-analysis
model_intercept <- coef(pp_model)['(Intercept)']
black_coef <- coef(pp_model)['race_factorAfrican-American']
(relative_risk_black_v_white <- sigmoid(model_intercept + black_coef) / sigmoid(model_intercept))
```

```
## (Intercept)
##       1.453
```

As ProPublica states, this shows us that "Black defendants are 45% more likely than white defendants to receive a higher [COMPAS] score correcting for the seriousness of their crime, previous arrests, and future criminal behavior." Similarly, women are 19.4% more likely than men and people under 25 are 2.5 times as likely as middle aged people to get a higher score:

```
# code adapted from https://github.com/propublica/compas-analysis
woman_coef <- coef(pp_model)['gender_factorFemale']
(relative_risk_woman_v_man <- sigmoid(model_intercept + woman_coef) / sigmoid(model_intercept))
```

```
## (Intercept)
##       1.195
```

```
# code adapted from https://github.com/propublica/compas-analysis
age_coef <- coef(pp_model)['age_factorLess than 25']
(relative_risk_young_v_middleage <- sigmoid(model_intercept + age_coef) / sigmoid(model_intercept))
```

```
## (Intercept)
##       2.496
```

Our Logistic Regression Model

Variable Selection

TODO: talk about excluding race but including sex

```
vars <- compas_data %>%
  select(sex, age, ends_with('count'), c_charge_degree, is_recid)

# ggplot(vars, aes(x = race, y = decile_score)) + geom_boxplot()
```

Training the Model

```
regmod <- glm(is_recid ~., family = binomial(link='logit'), data = vars)
summary(regmod)
```

```
##
## Call:
## glm(formula = is_recid ~ ., family = binomial(link = "logit"),
##     data = vars)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##  -3.14   -1.04   -0.56    1.09    2.46
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)         0.73332    0.09947    7.37  1.7e-13 ***
## sexMale             0.32061    0.06522    4.92  8.8e-07 ***
## age                -0.04534    0.00246  -18.40  < 2e-16 ***
## juv_fel_count       0.20658    0.09108    2.27   0.0233 *
## juv_misd_count      0.05195    0.07813    0.66   0.5062
## juv_other_count     0.19885    0.06559    3.03   0.0024 **
## priors_count        0.15587    0.00730   21.34  < 2e-16 ***
## c_charge_degreeM -0.14752    0.05407   -2.73   0.0064 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9990.5  on 7213  degrees of freedom
## Residual deviance: 8844.8  on 7206  degrees of freedom
## AIC: 8861
##
## Number of Fisher Scoring iterations: 4
```

```
# fitted_values <- fitted(regmod)
```

Parameter Tuning

Using ProPublica's Bias-Assessment Model on Our Model

Big TODO: we should nest this bias calculation inside cross-validation to make sure we're assessing our bias on future data (also interesting to look at the bias on the training data, however)

We can see if our model suffers from the same bias using ProPublica's bias-assessment methodology. We'll set up the same model they did, but this time instead of predicting `score_factor` for the COMPAS score, we'll create a `score_factor` variable *from our model's predictions.* To get valid predictions for the training data, we use 10-fold cross-validation.

```
set.seed(123)
all_predictions <- data.frame(id = compas_data$id, predicted_score = NA)

# split the data into 10 folds
all_indices <- 1:nrow(compas_data)
folds <- createFolds(all_indices, k = 10)

for (i in 1:10) {
  # make train/test split
  test_indices <- folds[[i]]
  training_indices <- all_indices[-test_indices]
  test_data <- compas_data[test_indices,]
  training_data <- compas_data[training_indices,]
  # build model on training data
  vars <- training_data %>%
    select(sex, age, ends_with('count'), c_charge_degree, is_recid)
  regmod <- glm(is_recid ~., family = binomial(link='logit'), data = vars)
  # store predictions on test data
  predicted_scores <- predict(regmod, newdata = test_data, type = 'response')
  all_predictions[names(predicted_scores),]$predicted_score <- predicted_scores
}
```

Next we create a new data frame with `new_score` as our risk of recidivism score. We also create a `new_score_text` variable that defines score categories in the same way, but this time based on a 0-1 scale instead of a 0-10 scale. Recall ProPublica defined this category as: High=8-10, Medium=5-7, Low=1-4. Thus we'll do: High=0.75-1.0, Medium=0.45-0.75, Low=0-0.45.

```
predicted_scores <- all_predictions$predicted_score
our_scores <- compas_data %>%
  mutate(new_score = predicted_scores) %>%
  mutate(
    new_score_text = as.factor(
      ifelse(new_score >= 0.75, 'High',
             ifelse(new_score >= 0.45, 'Medium', 'Low')))
  )
```

Now we train the same bias-assessment model, this time using `new_score` instead of `decile_score` and `new_score_text` instead of `score_text`.

```
new_df <- our_scores %>%
  select(age, c_charge_degree, race, age_cat, new_score_text, sex, priors_count, days_b_screening_arrest
  filter(days_b_screening_arrest <= 30) %>%
```

```r
  filter(days_b_screening_arrest >= -30) %>%
  filter(is_recid != -1) %>%
  filter(c_charge_degree != "O") %>%
  filter(new_score_text != 'N/A') %>%
  mutate(crime_factor = factor(c_charge_degree)) %>%
  mutate(age_factor = as.factor(age_cat)) %>%
  within(age_factor <- relevel(age_factor, ref = "25 - 45")) %>%
  mutate(race_factor = factor(race)) %>%
  within(race_factor <- relevel(race_factor, ref = "Caucasian")) %>%
  mutate(gender_factor = factor(sex, labels= c("Female","Male"))) %>%
  within(gender_factor <- relevel(gender_factor, ref = "Male")) %>%
  mutate(score_factor = factor(new_score_text != "Low", labels = c("LowScore","HighScore")))
```

```r
new_pp_model <- glm(
  score_factor ~ gender_factor + age_factor + race_factor + priors_count + crime_factor + two_year_recid
  family="binomial",
  data=new_df
)
```

Bias against black vs. white people

```r
model_intercept <- coef(new_pp_model)['(Intercept)']
black_coef <- coef(new_pp_model)['race_factorAfrican-American']
(relative_risk_black_v_white <- sigmoid(model_intercept + black_coef) / sigmoid(model_intercept))
```

```
## (Intercept)
##       1.209
```

Bias against women vs. men

```r
woman_coef <- coef(new_pp_model)['gender_factorFemale']
(relative_risk_woman_v_man <- sigmoid(model_intercept + woman_coef) / sigmoid(model_intercept))
```

```
## (Intercept)
##     0.03223
```

Bias against under 25 vs. middle-aged

```r
age_coef <- coef(new_pp_model)['age_factorLess than 25']
(relative_risk_young_v_middleage <- sigmoid(model_intercept + age_coef) / sigmoid(model_intercept))
```

```
## (Intercept)
##        5.75
```

Mitigating Bias

Much work has been done exploring techniques that allow for fair modeling and prediction in the presence of biased data or algorithms, and this area of study is rich and actively researched. Researchers Faisal Kamiran and Toon Calders, in their 2011 paper "Data preprocessing techniques for classification without discrimination," describe several techniques that *alter the given dataset* to (ideally) eliminate the source of the bias. Several other techniques deal with modifying classification/regression algorithms themselves to make fairer predictions. Here, the general idea is to instead clean the bias out of the data, after which normal classification methods can be used. Before we discuss the techniques we employ for dealing with bias, we first introduce a few key concepts.

Algorithmic Bias

This Wikipedia article provides a nice overview of different types of Algorithmic bias and several examples. Bias can manifest in an algorithm in various ways and can have multiple causes. The Wikipedia article linked above uses to term "pre-existing bias" to loosely describe a type of bias resulting from baking social or institutional biases into algorithms. Here the source of bias is not necessarily the algorithm but instead the values encoded into it by its creators or by the data it sees. For our brief foray into algorithmic bias here, we focus on this type of bias under the assumption that some underlying bias does exist in the United States criminal justice system and thus in our data set.

Protected/Sensitive Attributes

The term *protected attribute* or *sensitive attribute* typically refers to a discriptor of an individual upon which it is illegal to discriminate under the Fair Work Act. These include characteristics such as sexual orientation, gender identity, and race.

Combatting Algorithmic Bias

Broadly speaking, research in algorithmic bias works to (1) identify where and in what way bias is present (ProPublica's analysis is one such example), (2) come up with ways to constrain models to enforce fair predictions, or (3) alter the underlying data to minimize bias in the data itself. For our recidivism predictions, we'll employ some techniques from category 3 – altering the underlying data – based on Kamiran and Calders' work.

Mitigating Bias Through Data Preprocessing

Kamiran and Calders outline four types of techniques for preprocessing to mitigate bias, described at length in their paper:

1. **Suppression:** Given a sensitive attribute S, find and remove S and the other features most correlated with S.
2. **Massaging the dataset:** Given a sensitive attribute S, swap the labels of some observations with differing values for S to decrease discrimination while maintaining the same overall class distribution.
3. **Reweighing:** Up- or down-weight observations in the training data based on whether they have an under- or over-represented combinations of sensitive attribute S and the response variable.
4. **Sampling:** Calculate sample sizes for combinations of sensitive attribute S and the response that would, as Kamiran and Calders put it, "make the dataset descrimination-free." Then under- or over-sample observations accordingly to create a data set with those proportions.

Here we focus on Sampling.

Applying Sampling to the COMPAS Data to Combat Racial Bias

To narrow the scope of this report, we focus on *racial bias* against black people as compared to white people, and we implement sampling to combat this type of bias only. If our example words for racial bias, we could theoretically extend it to each of the protected attributes in the data.

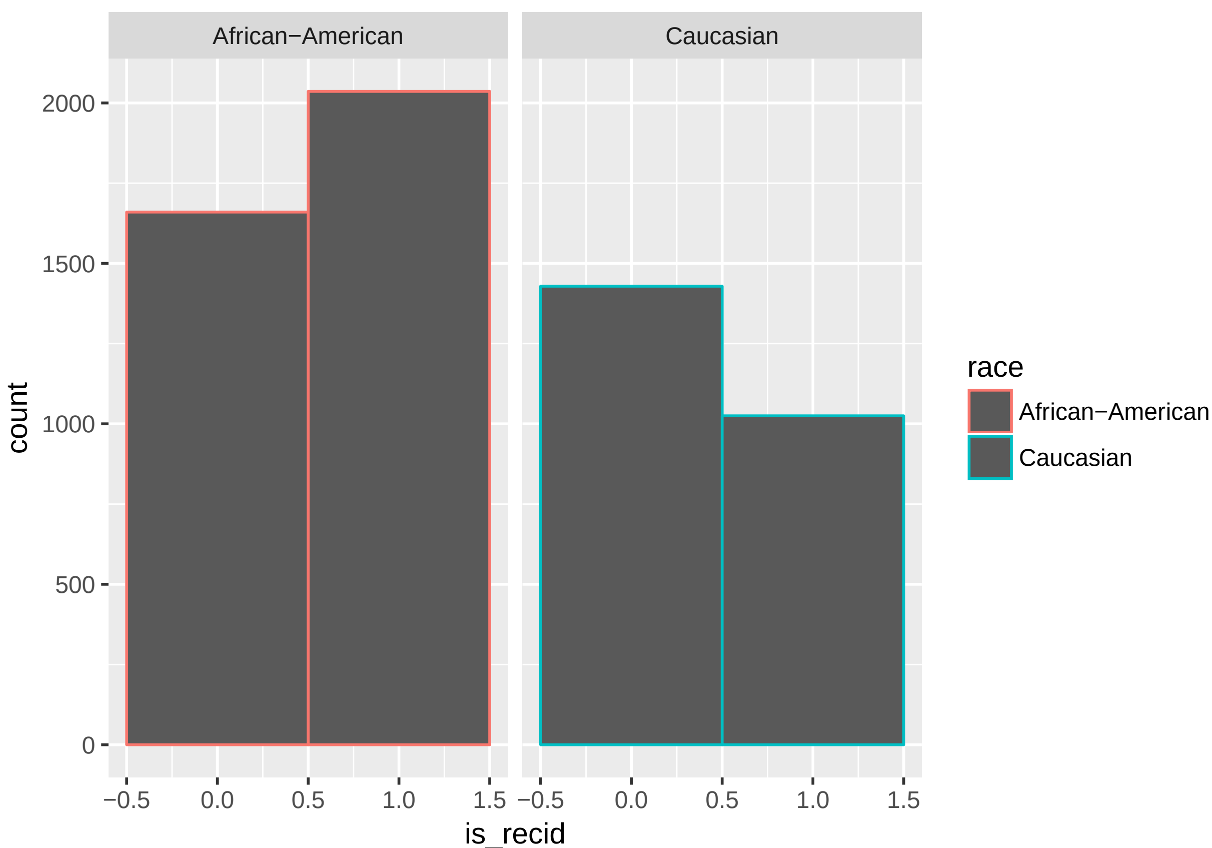
First we need to establish what the class imbalance is between blacks and whites in our data. We look for an imbalance in terms of recidivism rates, since recidivism is what our model predicts.

```
(recid_rates <- compas_data %>%
  select(race, is_recid) %>%
  group_by(race) %>%
  summarize(yes_recid = sum(is_recid == 1), no_recid = sum(is_recid == 0)))
```

```
## # A tibble: 6 x 3
##               race yes_recid no_recid
##             <fctr>     <int>    <int>
## 1 African-American      2036     1660
## 2            Asian        11       21
## 3        Caucasian      1025     1429
## 4         Hispanic       245      392
## 5  Native American        11        7
## 6            Other       143      234
```

Focusing on African-Americans and Caucasians and plotting this table as a bar chart, we have:

```
compas_data %>%
  filter(race %in% c('African-American', 'Caucasian')) %>%
  ggplot(aes(x = is_recid, color = race)) +
    geom_histogram(bins = 2) +
    facet_wrap(~ race)
```



Kamiran and Calders define four types of observation with respect to a protected attribute and a class label. Their definitions are as follows:

- Deprived community with positive class labels (DP)

- Deprived community with negative class labels (DN)
- Favored community with positive class labels (FP)
- Favored community with negative class labels (FN)

In our case these correspond to:

- DP: African-Americans with `is_recid` = 1
- DN: African-Americans with `is_recid` = 0
- FP: Caucasians with `is_recid` = 1
- FN: Caucasians with `is_recid` = 0

```
DP_data <- compas_data %>%
  filter(race == 'African-American' & is_recid == 1)
DN_data <- compas_data %>%
  filter(race == 'African-American' & is_recid == 0)
FP_data <- compas_data %>%
  filter(race == 'Caucasian' & is_recid == 1)
FN_data <- compas_data %>%
  filter(race == 'Caucasian' & is_recid == 0)
```

The process for sampling to get a non-discrimatory sample is as follows.

1. Compute the expected size for each group if the data were non-discriminatory
2. Under- and over-sample accordingly to eliminate descrimination
3. Train the model on the sampled data

In order to get an estimation of how biased our model is after making this adjustment to the data, we nest this process in cross-validation. Thus our overall process will be the following (using k-fold cross-validation).

1. Split the data into k folds
2. Hold out one fold as test data, use the remaining k-1 folds as training data
3. Perform sampling on the training data
    - Compute the expected size for each group if the data were non-discriminatory
    - Under- and over-sample accordingly to eliminate descrimination
4. Train our model on the training data
5. Predict recidivism scores on the test data and store predictions
6. Repeat (2) - (5) for each fold, getting one prediction for each data point
7. Compute relative risk with new predictions

First we walk through an example of the sampling process on the whole data set before embedding the enitre process in cross-validation.

Demonstration of the sampling process

1. Compute the expected size for each group if the data were non-discriminatory

Here we have the following counts for each group:

```
recid_rates %>% filter(race %in% c('African-American', 'Caucasian'))
```

```
## # A tibble: 2 x 3
##               race yes_recid no_recid
##             <fctr>     <int>    <int>
## 1 African-American      2036     1660
## 2        Caucasian      1025     1429
```

Thus we'll say the expected number of observations for each group is the mean of the four counts:

```r
(expected_num_observations <- mean(c(nrow(DP_data), nrow(DN_data), nrow(FP_data), nrow(FN_data))))
```

```
## [1] 1538
```

2. Under- and over-sample to eliminate descrimination

We want equal representation for each group (DP, DN, FP, and FN). This means getting 1537.5 observations of each. We under- or over-sample accordingly. Kamiran and Calders describe two methods for performing this sampling: *Preferential Sampling*, which is more likely to sample boundary observations (with a high probability of being in either class); and *Uniform Sampling*, which samples all points with equal probability. Here we perform Uniform Sampling with replacement.

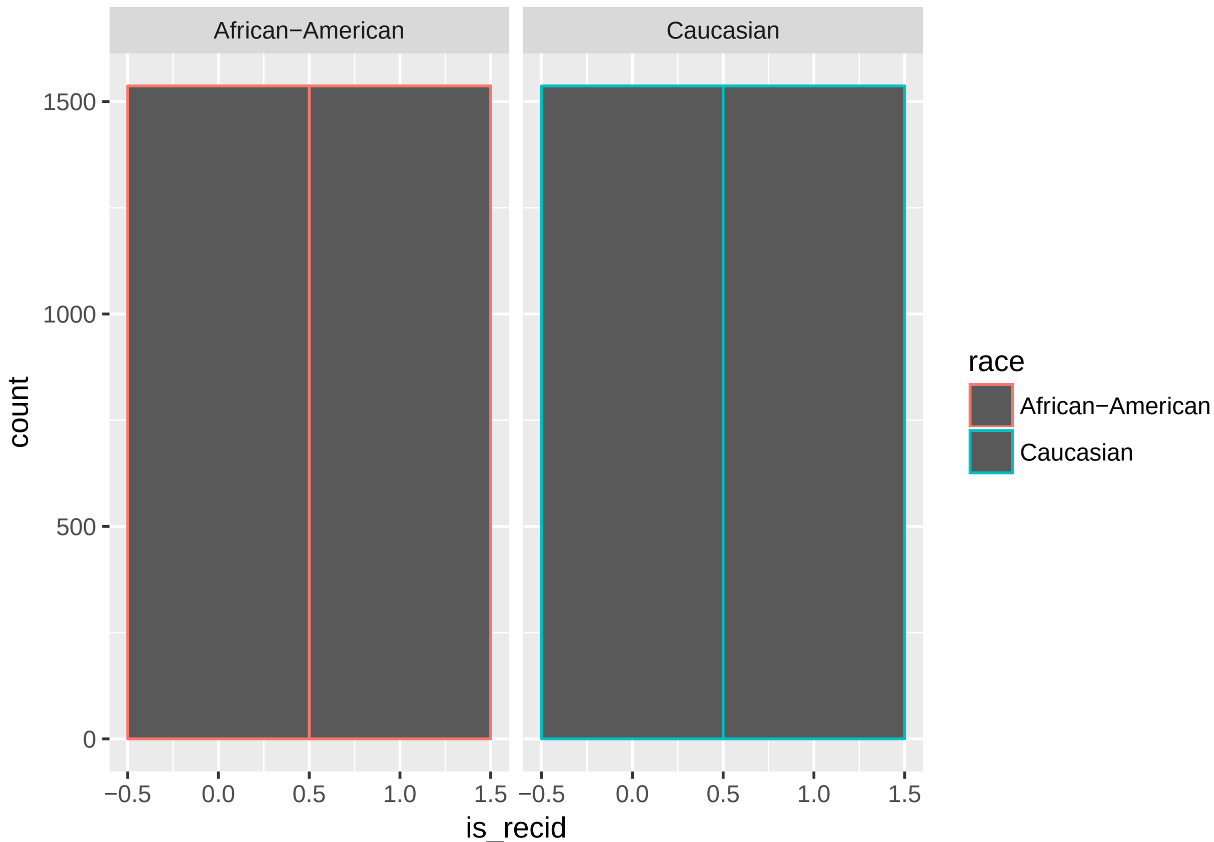
```r
set.seed(123)
# sample indices
DP_sampled_indices <- sample(1:nrow(DP_data), size = expected_num_observations, replace = TRUE)
DN_sampled_indices <- sample(1:nrow(DN_data), size = expected_num_observations, replace = TRUE)
FP_sampled_indices <- sample(1:nrow(FP_data), size = expected_num_observations, replace = TRUE)
FN_sampled_indices <- sample(1:nrow(FN_data), size = expected_num_observations, replace = TRUE)
```

And we build a new data set from the sampled observations.

```r
# build a new data set
sampled_compas_data <- rbind(
  DP_data[DP_sampled_indices,],
  DN_data[DN_sampled_indices,],
  FP_data[FP_sampled_indices,],
  FN_data[FN_sampled_indices,]
)
```

We double check visually that our sampling successfully leveled the classes for each partition.

```r
sampled_compas_data %>%
  filter(race %in% c('African-American', 'Caucasian')) %>%
  ggplot(aes(x = is_recid, color = race)) +
    geom_histogram(bins = 2) +
    facet_wrap(~ race)
```



With Uniform Sampling applied to our data for the *black/white* protected race attributes, we're ready to train our model again and see (1) if its level of bias has decreased and also see (2) how its predictive accuracy has changed.

3. Train the model on the sampled data

Here we'll train the model following the same process from above and see how our bias (measured by ProPublica's methodology) compares to the bias before sampling.

We train our logistic regression model.

```
vars <- sampled_compas_data %>%
  select(sex, age, ends_with('count'), c_charge_degree, is_recid)
post_sample_regmod <- glm(is_recid ~., family = binomial(link='logit'), data = vars)
```

Next we would predict scores **for observations we haven't seen**. Given that we didn't hold out a test set, we don't have any such observations. However, we have now seen how sampling works and are thus ready to do so within cross-validation.

Sampling with Cross-validation

Given that we have seen most of the following process step by step in previous sections, we implement this section without explanation of those parts we have seen already.

```
set.seed(123)
all_predictions <- data.frame(id = compas_data$id, predicted_score = NA)

#### 1. Split the data into 10 folds
```

```r
all_indices <- 1:nrow(compas_data)
folds <- createFolds(all_indices, k = 10)

#### 2. Hold out one fold as test data, use the remaining k-1 folds as training data
for (i in 1:10) {
  test_indices <- folds[[i]]
  training_indices <- all_indices[-test_indices]
  test_data <- compas_data[test_indices,]
  training_data <- compas_data[training_indices,]

  #### 3. Perform sampling on the training data
  # create partitions
  DP_data <- training_data %>%
    filter(race == 'African-American' & is_recid == 1)
  DN_data <- training_data %>%
    filter(race == 'African-American' & is_recid == 0)
  FP_data <- training_data %>%
    filter(race == 'Caucasian' & is_recid == 1)
  FN_data <- training_data %>%
    filter(race == 'Caucasian' & is_recid == 0)
  # get expected number of observations
  expected_num_observations <- mean(c(nrow(DP_data), nrow(DN_data), nrow(FP_data), nrow(FN_data)))
  # under- and over-sample to eliminate descrimination
  DP_sampled_indices <- sample(1:nrow(DP_data), size = expected_num_observations, replace = TRUE)
  DN_sampled_indices <- sample(1:nrow(DN_data), size = expected_num_observations, replace = TRUE)
  FP_sampled_indices <- sample(1:nrow(FP_data), size = expected_num_observations, replace = TRUE)
  FN_sampled_indices <- sample(1:nrow(FN_data), size = expected_num_observations, replace = TRUE)
  # build new data set
  sampled_training_data <- rbind(
    DP_data[DP_sampled_indices,],
    DN_data[DN_sampled_indices,],
    FP_data[FP_sampled_indices,],
    FN_data[FN_sampled_indices,]
  )

  #### 4. Train our model on the training data
  vars <- sampled_training_data %>%
    select(sex, age, ends_with('count'), c_charge_degree, is_recid)
  post_sample_regmod <- glm(is_recid ~., family = binomial(link='logit'), data = vars)

  #### 5. Predict recidivism scores on the test data and store predictions
  predicted_scores <- predict(post_sample_regmod, newdata = test_data, type = 'response')
  all_predictions[names(predicted_scores),]$predicted_score <- predicted_scores
}
#### 7. Compute relative risk with new predictions
# turn predictions from log odds ratios into probabilities of recidivating
recidivism_scores <- all_predictions$predicted_score
# create new scores and score categories
post_sample_scores <- compas_data %>%
  mutate(new_score = recidivism_scores) %>%
  mutate(
    new_score_text = as.factor(
      ifelse(new_score >= 0.75, 'High',
```

```
            ifelse(new_score >= 0.45, 'Medium', 'Low'))))
# build ProPublica's data frame with our scores
post_sample_df <- post_sample_scores %>%
  select(age, c_charge_degree, race, age_cat, new_score_text, sex, priors_count, days_b_screening_arres
  filter(days_b_screening_arrest <= 30) %>%
  filter(days_b_screening_arrest >= -30) %>%
  filter(is_recid != -1) %>%
  filter(c_charge_degree != "O") %>%
  filter(new_score_text != 'N/A') %>%
  mutate(crime_factor = factor(c_charge_degree)) %>%
  mutate(age_factor = as.factor(age_cat)) %>%
  within(age_factor <- relevel(age_factor, ref = "25 - 45")) %>%
  mutate(race_factor = factor(race)) %>%
  within(race_factor <- relevel(race_factor, ref = "Caucasian")) %>%
  mutate(gender_factor = factor(sex, labels= c("Female","Male"))) %>%
  within(gender_factor <- relevel(gender_factor, ref = "Male")) %>%
  mutate(score_factor = factor(new_score_text != "Low", labels = c("LowScore","HighScore")))
# train ProPublica's model
post_sample_pp_model <- glm(
  score_factor ~ gender_factor + age_factor + race_factor + priors_count + crime_factor + two_year_reci
  family="binomial",
  data=post_sample_df
)
```

Now we can measure the amount of bias in our model after applying sampling and compare to the bias before.

Bias Before and After Sampling

We can re-compute the relative risk for blacks vs. whites and compare it to the relative risk for our model before sampling to see if sampling improved the level of bias in our mode.

```
model_intercept <- coef(post_sample_pp_model)['(Intercept)']
black_coef <- coef(post_sample_pp_model)['race_factorAfrican-American']
(post_sample_relative_risk_black_v_white <- sigmoid(model_intercept + black_coef) / sigmoid(model_inter
```

```
## (Intercept)
##        1.26
```

Taking the ratio and subtracting from 1, we were able to decrease the bias in relative risk by:

```
pre_sample_relative_risk_black_v_white <- relative_risk_black_v_white
1 - post_sample_relative_risk_black_v_white / pre_sample_relative_risk_black_v_white
```

```
## (Intercept)
##    -0.04212
```

References