COMPAS fairness analysis

Isha Doshi

2022-12-04

Analyzing if COMPAS is fair

Loaded the COMPAS data, and performed basic sanity checks.

```
compas=read.delim("compas-score-data.csv.bz2")
head(compas)
```

```
##   age c_charge_degree             race         age_cat  sex priors_count
## 1  69               F            Other Greater than 45 Male            0
## 2  34               F African-American         25 - 45 Male            0
## 3  24               F African-American    Less than 25 Male            4
## 4  44               M            Other         25 - 45 Male            0
## 5  41               F        Caucasian         25 - 45 Male           14
## 6  43               F            Other         25 - 45 Male            3
##   decile_score two_year_recid
## 1            1              0
## 2            3              1
## 3            4              1
## 4            1              0
## 5            6              1
## 6            4              0
```

```
any(is.na(compas))
```

```
## [1] FALSE
```

```
#summary(compas)
```

Filtering the data to keep only Caucasians and African-Americans

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
compas=compas %>%
  filter(race %in% c("African-American", "Caucasian"))
head(compas)
```

```
##   age c_charge_degree             race      age_cat    sex priors_count
## 1  34               F African-American      25 - 45   Male            0
## 2  24               F African-American Less than 25   Male            4
## 3  41               F        Caucasian      25 - 45   Male           14
## 4  39               M        Caucasian      25 - 45 Female            0
## 5  27               F        Caucasian      25 - 45   Male            0
## 6  23               M African-American Less than 25   Male            3
##   decile_score two_year_recid
## 1            3              1
## 2            4              1
## 3            6              1
## 4            1              0
## 5            4              0
## 6            6              1
```

Created a new dummy variable based off of COMPAS risk score (decile_score), which indicates if an individual was classified as low risk (score 1-4) or high risk (score 5-10).

```
compas=compas%>%mutate(risk_score=case_when(decile_score <= 4 ~ "low risk",
  decile_score > 4 ~ "high risk"))
head(compas)
```

```
##   age c_charge_degree             race      age_cat    sex priors_count
## 1  34               F African-American      25 - 45   Male            0
## 2  24               F African-American Less than 25   Male            4
## 3  41               F        Caucasian      25 - 45   Male           14
## 4  39               M        Caucasian      25 - 45 Female            0
## 5  27               F        Caucasian      25 - 45   Male            0
## 6  23               M African-American Less than 25   Male            3
##   decile_score two_year_recid risk_score
## 1            3              1   low risk
## 2            4              1   low risk
## 3            6              1  high risk
## 4            1              0   low risk
## 5            4              0   low risk
## 6            6              1  high risk
```

Now analyzing the offenders across this new risk category.

Checking the recidivism rate for low-risk and high-risk individuals

```
#low risk rate = (no. of two_year_recid =1 where risk rate is low)/total no. of low risk rate
low_risk_recidivism = count(compas%>%filter(risk_score=="low risk")%>%filter(two_year_recid==1)) / coun
paste("recidivism rate for low-risk is ",low_risk_recidivism)
```

```
## [1] "recidivism rate for low-risk is  0.320014529604068"
```

```
#low risk rate = (no. of two_year_recid =1 where risk rate is high)/total no. of high risk rate
high_risk_recividism= count(compas%>%filter(risk_score=="high risk")%>%filter(two_year_recid==1)) / cou
paste("recidivism rate for high-risk is", high_risk_recividism)
```

```
## [1] "recidivism rate for high-risk is 0.634455445544554"
```

Checking the recidivism rates for African-Americans and Caucasians

```
#For African American = no. of two_year_recid where race equals African American/ No. of African Americ
african_american_recidivism= count(compas%>%filter(race=="African-American")%>%filter(two_year_recid==1
paste("recidivism rate for African Americans is",african_american_recidivism)
```

```
## [1] "recidivism rate for African Americans is 0.523149606299213"
```

```
#For Caucasian = no. of two_year_recid where race equals Caucasians/ No. of Caucasians
caucasian_american_recidivism= count(compas%>%filter(race=="Caucasian")%>%filter(two_year_recid==1)) / 
paste("recidivism rate for Caucasians is",caucasian_american_recidivism)
```

```
## [1] "recidivism rate for Caucasians is 0.390870185449358"
```

Creating a confusion matrix comparing COMPAS predictions for recidivism (is/is not low risk) and the actual two-year recidivism. To keep things coherent, let's call recidivists "positive". The precision here is 0.634, When COMPAS predicts high recidivism, 63.4% times it is correctly predicted. The ratio of correct positive predictions to the total positives recidivism was 64%. 881 people were predicated as low risk when they were recidivists whereas 923 people were mis-classified as high risk when they were low risk.

```
#creating confusion matrix
#accuracy is the number of true positives and true negatives divided by total
table(compas$risk_score,compas$two_year_recid)
```

```
##
##                0    1
##   high risk  923 1602
##   low risk  1872  881
```

```
accuracy <- (1872+1602)/(1872+923+1602+881)
paste("accuracy is", accuracy)
```

```
## [1] "accuracy is 0.658203865100417"
```

```
#Precision is the number of true positives divided by the number of true positives plus the number of f
precision<- 1602/(1602+923)
paste("precision is",precision)
```

```
## [1] "precision is 0.634455445544554"
```

```
#recall is the number of true positives divided by the number of true positives plus the number of fals
recall<-1602/(1602+881)
paste("recall is",recall)
```

```
## [1] "recall is 0.645187273459525"
```

COMPAS is only 65% accurate. I would not feel comfortable having a judge use COMPAS to inform my sentencing guideline. If they model had a higher precision (about 85-90 percent), I would have been more comfortable. Or if the false positives were very low and false negatives were high, then I would be more comfortable. Yes, human judges are also not perfect, but if we are developing an algorithm, we should make an effort to make it better than humans.

```
#Misclassification is 1-accuracy or (all incorrect / all) = FP + FN / TP + TN + FP + FN
misclassification=1-accuracy
misclassification
```

```
## [1] 0.3417961
```

Repeating the confusion matrix calculation and analysis but this time I will do it separately for African-Americans and Caucasians:

This classification is slightly more accurate for Caucasians than for African Americans.

```
compas_AA=compas %>%
  filter(race =="African-American")
table(compas_AA$risk_score,compas_AA$two_year_recid)
```

```
##
##                0    1
##   high risk  641 1188
##   low risk   873  473
```

```
accuracy <- (1188+873)/(1188+641+873+473)
paste("accuracy for African-Americans is", accuracy)
```

```
## [1] "accuracy for African-Americans is 0.649133858267717"
```

```
compas_C=compas %>%
  filter(race =="Caucasian")
table(compas_C$risk_score,compas_C$two_year_recid)
```

```
##
##               0   1
##   high risk 282 414
##   low risk  999 408
```

```
accuracy <- (414+999)/(282+414+999+408)
paste("accuracy for Caucasians is", accuracy)
```

```
## [1] "accuracy for Caucasians is 0.671897289586305"
```

There is a higher false positive rate for African Americans.

```
#FPR=FP/(FP+TN)
fpr_AA=641/(641+873)
paste("False postive rate for African Americans is", fpr_AA)
```

```
## [1] "False postive rate for African Americans is 0.42338177014531"
```

```
fpr_C=282/(282+999)
paste("False postive rate for Caucasian is", fpr_C)
```

```
## [1] "False postive rate for Caucasian is 0.220140515222482"
```

There is a higher rate of false negatives for Caucasians.

```
#FN/(FN+TP)
fnr_AA=473/(473+1188)
paste("False negative rate for African Americans is", fnr_AA)
```

```
## [1] "False negative rate for African Americans is 0.28476821192053"
```

```
fnr_C=408/(408+414)
paste("False negative rate for Caucasian is", fnr_C)
```

```
## [1] "False negative rate for Caucasian is 0.496350364963504"
```

COMPAS's true negative and true positive percentages are similar for African-American and Caucasian individuals, but that false positive rates and false negative rates are different. I don't think COMPAS algorithm is fair. I believe similar groups of people, here defined by race, should be treated similarly. So whites who do not re-offend should have the same misclassification rate as blacks who do not re-offend. This is clearly violated with COMPAS score.

Attempting to make my own COMPAS!

Before we start: F1 score is a weighted average of precision and recall. As we know in precision and in recall there is false positive and false negative so it also consider both of them. F1 score is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall. We will use all these performance measures for this task.

Split my data into training and validation set. Developed a model using logistic regression, and changing the dependent variables. We get a better model when we consider priors_count+age. The AIC score for this model is lower - 5344.

```
#removing variables
compas4Model=read.delim("compas-score-data.csv.bz2")
compas4Model=compas4Model%>%select(age, c_charge_degree, race, age_cat,sex,priors_count,two_year_recid)
head(compas4Model)
```

```
##   age c_charge_degree             race         age_cat  sex priors_count
## 1  69               F            Other Greater than 45 Male            0
```

```
## 2  34               F African-American         25 - 45 Male            0
## 3  24               F African-American    Less than 25 Male            4
## 4  44               M            Other         25 - 45 Male            0
## 5  41               F        Caucasian         25 - 45 Male           14
## 6  43               F            Other         25 - 45 Male            3
##   two_year_recid
## 1              0
## 2              1
## 3              1
## 4              0
## 5              1
## 6              0
```

```
#splitting data into training and validation set
library(dplyr)

#make this example reproducible
set.seed(1)

#create ID column
compas4Model$id <- 1:nrow(compas4Model)

#use 70% of dataset as training set and 30% as test set
train <- compas4Model %>% dplyr::sample_frac(0.70)
test  <- dplyr::anti_join(compas4Model, train, by = 'id')
head(train)
```

```
##   age c_charge_degree             race         age_cat    sex priors_count
## 1  44               F        Caucasian         25 - 45   Male            0
## 2  31               M        Caucasian         25 - 45 Female            0
## 3  50               M        Caucasian Greater than 45   Male            2
## 4  22               M        Caucasian    Less than 25   Male            3
## 5  29               F African-American         25 - 45   Male           13
## 6  27               F African-American         25 - 45   Male            5
##   two_year_recid   id
## 1              1 1017
## 2              0 4775
## 3              1 2177
## 4              1 5026
## 5              1 1533
## 6              1 4567
```

```
head(test)
```

```
##   age c_charge_degree             race         age_cat    sex priors_count
## 1  34               F African-American         25 - 45   Male            0
## 2  41               F        Caucasian         25 - 45   Male           14
## 3  27               F        Caucasian         25 - 45   Male            0
## 4  37               M        Caucasian         25 - 45 Female            0
## 5  47               F        Caucasian Greater than 45 Female            1
## 6  31               F African-American         25 - 45   Male            7
##   two_year_recid id
## 1              1  2
```

```
## 2              1  5
## 3              0  8
## 4              0 10
## 5              1 12
## 6              1 13
```

```
#logistic regression
model1<-glm(two_year_recid~priors_count, data=train, family=binomial())
summary(model1)
```

```
##
## Call:
## glm(formula = two_year_recid ~ priors_count, family = binomial(),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6172  -0.9889  -0.9319   1.2481   1.4446
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.60930    0.03980  -15.31   <2e-16 ***
## priors_count  0.14819    0.00874   16.95   <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: 5963.3  on 4319  degrees of freedom
## Residual deviance: 5578.2  on 4318  degrees of freedom
## AIC: 5582.2
##
## Number of Fisher Scoring iterations: 4
```

```
model2<-glm(two_year_recid~priors_count+age, data=train, family=binomial())
summary(model2)
```

```
##
## Call:
## glm(formula = two_year_recid ~ priors_count + age, family = binomial(),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5586  -1.0437  -0.6143   1.1383   2.2948
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.879451   0.106260   8.276   <2e-16 ***
## priors_count  0.169426   0.009154  18.507   <2e-16 ***
## age          -0.046248   0.003163 -14.623   <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: 5963.3  on 4319  degrees of freedom
## Residual deviance: 5338.3  on 4317  degrees of freedom
## AIC: 5344.3
##
## Number of Fisher Scoring iterations: 4
```

Added sex to the model. It improves the performance (the AIC score is 5326, which is lower), but just a tiny bit. We are adding another variables, which lowers the AIC score as well.

```
model3<-glm(two_year_recid~priors_count+age+sex, data=train, family=binomial())
summary(model3)
```

```
##
## Call:
## glm(formula = two_year_recid ~ priors_count + age + sex, family = binomial(),
##     data = train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.551  -1.022  -0.609   1.119   2.258
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.569145   0.126909   4.485 7.30e-06 ***
## priors_count  0.165118   0.009161  18.024  < 2e-16 ***
## age          -0.045956   0.003161 -14.537  < 2e-16 ***
## sexMale       0.383194   0.086148   4.448 8.66e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5963.3  on 4319  degrees of freedom
## Residual deviance: 5318.2  on 4316  degrees of freedom
## AIC: 5326.2
##
## Number of Fisher Scoring iterations: 4
```

Added race. The model improves very little, the AIC score decreases to 5323.

```
model4<-glm(two_year_recid~priors_count+age+sex+race, data=train, family=binomial())
summary(model4)
```

```
##
## Call:
## glm(formula = two_year_recid ~ priors_count + age + sex + race,
##     family = binomial(), data = train)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -2.5470  -1.0272  -0.6048   1.1093   2.2381
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           0.607294   0.128479   4.727 2.28e-06 ***
## priors_count          0.161958   0.009344  17.333  < 2e-16 ***
## age                  -0.045578   0.003234 -14.093  < 2e-16 ***
## sexMale               0.395813   0.086353   4.584 4.57e-06 ***
## raceAsian            -1.143839   0.576614  -1.984   0.0473 *
## raceCaucasian        -0.029251   0.075059  -0.390   0.6968
## raceHispanic         -0.235448   0.124707  -1.888   0.0590 .
## raceNative American -1.023769   0.805607  -1.271   0.2038
## raceOther            -0.294446   0.150076  -1.962   0.0498 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5963.3  on 4319  degrees of freedom
## Residual deviance: 5305.5  on 4311  degrees of freedom
## AIC: 5323.5
##
## Number of Fisher Scoring iterations: 4
```

The accuracy of my model is also 65%, which is almost the same as COMPAS. Our model is more precise. Recall on other hand is lower for our model. I considered gender and race as a part of the model, since it did improve the AIC score a little. I feel judges should consider the predictions made by these models only if they are highly accurate and precise and have a good recall. I feel judges should still consider the case and make their decision as well, not compeletly relying on the model to make the decision.

```
test <- test %>% mutate(predicted_two_year_recid = predict(model4, test))
test=test%>%mutate(predicted_two_year_recid=case_when(predicted_two_year_recid>=0.5~0,predicted_two_yea
head(test)
```

```
##   age c_charge_degree             race         age_cat    sex priors_count
## 1  34               F African-American         25 - 45   Male            0
## 2  41               F        Caucasian         25 - 45   Male           14
## 3  27               F        Caucasian         25 - 45   Male            0
## 4  37               M        Caucasian         25 - 45 Female            0
## 5  47               F        Caucasian Greater than 45 Female            1
## 6  31               F African-American         25 - 45   Male            7
##   two_year_recid id predicted_two_year_recid
## 1              1  2                        1
## 2              1  5                        0
## 3              0  8                        1
## 4              0 10                        1
## 5              1 12                        1
## 6              1 13                        0
```

```
table(test$predicted_two_year_recid, test$two_year_recid)
```

```
##
##       0   1
##   0  69 236
##   1 968 579
```

```
accuracyM=(968+236)/(968+236+69+579)
paste("accuracy of our model is", accuracyM)
```

```
## [1] "accuracy of our model is 0.650107991360691"
```

```
precisionM<- 236/(236+69)
paste("precision of our model is",precisionM)
```

```
## [1] "precision of our model is 0.773770491803279"
```

```
recallM<-236/(236+579)
paste("recall of our model is",recallM)
```

```
## [1] "recall of our model is 0.289570552147239"
```