

# Post-processing measures

## Imports

```
library(fairmodels)
library(tidymodels)
```

```
## -- Attaching packages ----- tidymodels 0.2.0 --
```

```
## v broom      0.8.0    v recipes      0.2.0
## v dials      0.1.1    v rsample      0.1.1
## v dplyr      1.0.9    v tibble      3.1.7
## v ggplot2    3.3.6    v tidyr       1.2.0
## v infer      1.0.0    v tune        0.2.0
## v modeldata  0.1.1    v workflows   0.2.6
## v parsnip    0.2.1    v workflowsets 0.2.1
## v purrr      0.3.4    v yardstick   0.0.9
```

```
## -- Conflicts ----- tidymodels_conflicts() --
```

```
## x purrr::discard() masks scales::discard()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x recipes::step()  masks stats::step()
## * Search for functions across packages at https://www.tidymodels.org/find/
```

```
library(rpart)
```

```
##
```

```
## Attaching package: 'rpart'
```

```
## The following object is masked from 'package:dials':
```

```
##
```

```
##      prune
```

```
library(discrim)
```

```
##
```

```
## Attaching package: 'discrim'
```

```
## The following object is masked from 'package:dials':
```

```
##
```

```
##      smoothness
```

```
source("../scripts/metrics_on_dataset.R")

## -- Attaching packages ----- tidyverse 1.3.1 --

## v readr 2.1.2      v forcats 0.5.1
## v stringr 1.4.0

## -- Conflicts ----- tidyverse_conflicts() --
## x readr::col_factor() masks scales::col_factor()
## x purrr::discard()    masks scales::discard()
## x dplyr::filter()     masks stats::filter()
## x stringr::fixed()    masks recipes::fixed()
## x dplyr::lag()         masks stats::lag()
## x readr::spec()       masks yardstick::spec()
```

## Data

```
df <- read_rds("../data/selection.rds") %>%
  select(-gender, -rating) %>%
  mutate(accepted = as.factor(accepted))
df_test <- read_rds("../data/selection_test.rds") %>%
  select(-gender, -rating) %>%
  mutate(accepted = as.factor(accepted))
```

## Naive Bayes ensemble

### Functions

```
adjust_fit <- function(cutoffs, direction){
  if (direction == "up"){
    cutoffs["Non_Dutch"] <- cutoffs["Non_Dutch"] - 0.01
  } else if (direction == "down"){
    cutoffs["Dutch"] <- cutoffs["Dutch"] + 0.01
  }
  cutoffs
}

df_disc <- function(df){
  summary_true <- group_fairness(df, nationality, predicted)[[1]] %>%
    filter(predicted == "TRUE")
  max_val <- max(select(summary_true, perc))
  min_val <- min(select(summary_true, perc))

  list(disc = max_val - min_val, n = sum(select(summary_true, total)))
}

predictions <- function(df, fitted_models, cutoffs) {
```

```

df_dutch <- filter(df, nationality == "Dutch")
df_non_dutch <- filter(df, nationality != "Dutch")

predictions_dutch <- predict(fitted_models[["Dutch"]], df_dutch, type="prob")[".pred_TRUE"]
predictions_non_dutch <- predict(fitted_models[["Non_Dutch"]], df_non_dutch, type="prob")[".pred_TRUE"]

df_dutch["predicted"] <- as.factor(predictions_dutch >= cutoffs["Dutch"])
df_non_dutch["predicted"] <- as.factor(predictions_non_dutch >= cutoffs["Non_Dutch"])

joined_df = bind_rows(df_dutch, df_non_dutch)
}

causal_discrimination_joined_model <- function(df, fitted_models, cutoffs) {
  pop_size <- nrow(df)

  df_dutch <- filter(df, nationality == "Dutch")
  df_non_dutch <- filter(df, nationality != "Dutch")

  predictions_dutch <- predict(fitted_models[["Dutch"]], df_dutch, type="prob")[".pred_TRUE"]
  inv_predictions_dutch <- predict(fitted_models[["Non_Dutch"]], df_dutch, type="prob")[".pred_TRUE"]

  predictions_non_dutch <- predict(fitted_models[["Non_Dutch"]], df_non_dutch, type="prob")[".pred_TRUE"]
  inv_predictions_non_dutch <- predict(fitted_models[["Dutch"]], df_non_dutch, type="prob")[".pred_TRUE"]

  df_dutch["predicted"] <- as.factor(predictions_dutch >= cutoffs["Dutch"])
  df_dutch["inv_predicted"] <- as.factor(inv_predictions_dutch >= cutoffs["Non_Dutch"])
  df_non_dutch["predicted"] <- as.factor(predictions_non_dutch >= cutoffs["Non_Dutch"])
  df_non_dutch["inv_predicted"] <- as.factor(inv_predictions_non_dutch >= cutoffs["Dutch"])

  joined_df <- bind_rows(df_dutch, df_non_dutch) %>%
    mutate(different = predicted != inv_predicted)

  list(sum(joined_df$different)/pop_size, joined_df)
}

```

## Setup

```

original_n <- sum(df$accepted == "TRUE")

df_dutch <- filter(df, nationality == "Dutch")
df_non_dutch <- filter(df, nationality != "Dutch")

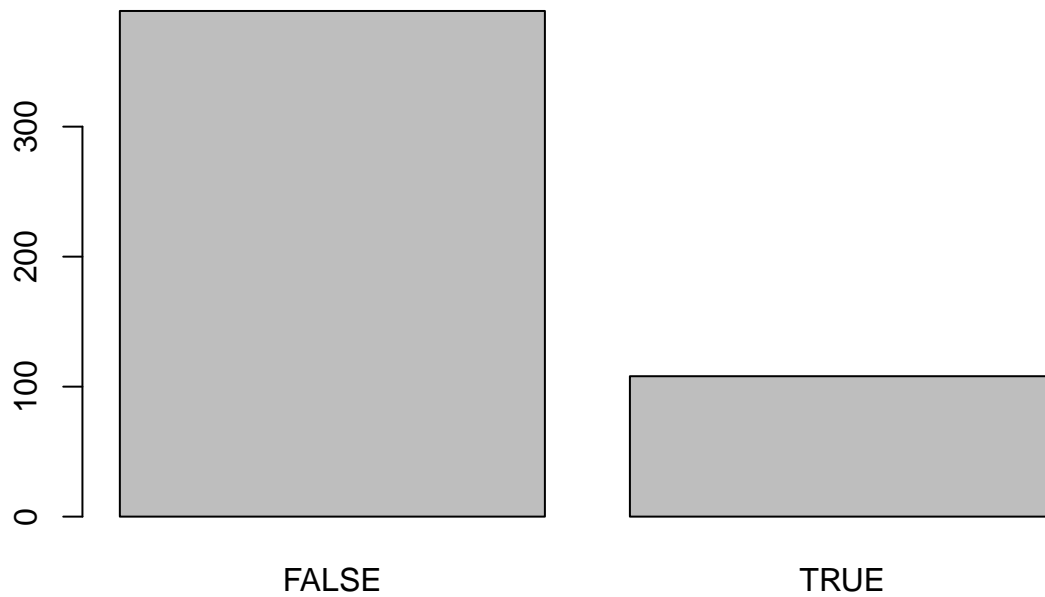
model <- naive_Bayes()

cutoffs <- c(Dutch = 0.5, Non_Dutch = 0.5)

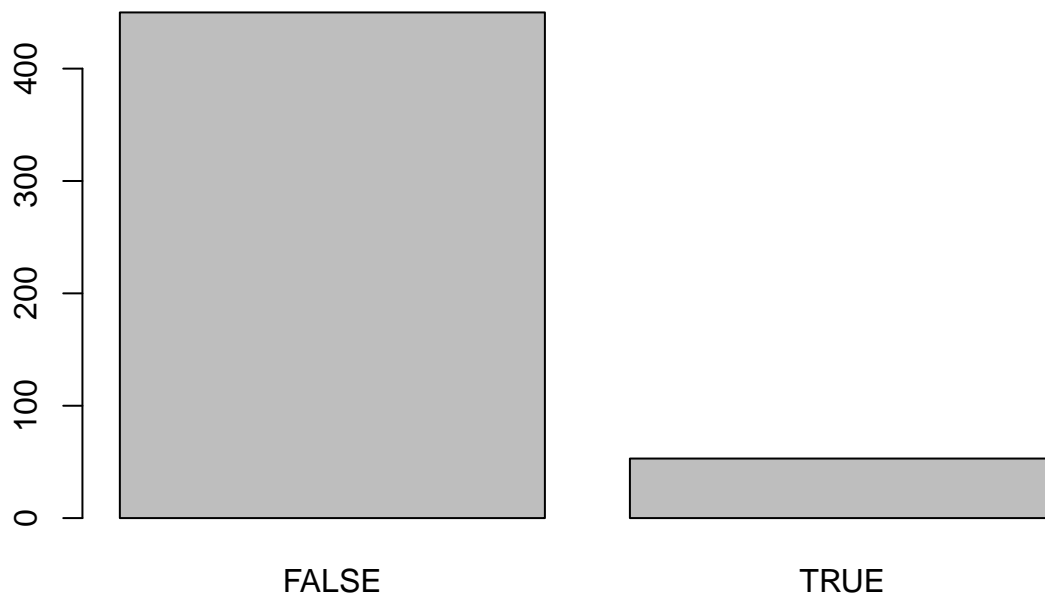
models <- list(Dutch = model %>%
  fit(accepted ~ test_score + english_cert + extracurricular, df_dutch),
  Non_Dutch = model %>%
  fit(accepted ~ test_score + english_cert + extracurricular, df_non_dutch))

```

```
predictions_df = predictions(df, models, cutoffs)
plot(filter(predictions_df, nationality == "Dutch") %>% select(predicted))
```



```
plot(filter(predictions_df, nationality == "Non_Dutch") %>% select(predicted))
```



```
result <- df_disc(predictions_df)

i <- 0
while (result$disc > 2) {

  if (result$n < original_n) {
    cutoffs <- adjust_fit(cutoffs, "up")
  } else {
    cutoffs <- adjust_fit(cutoffs, "down")
  }

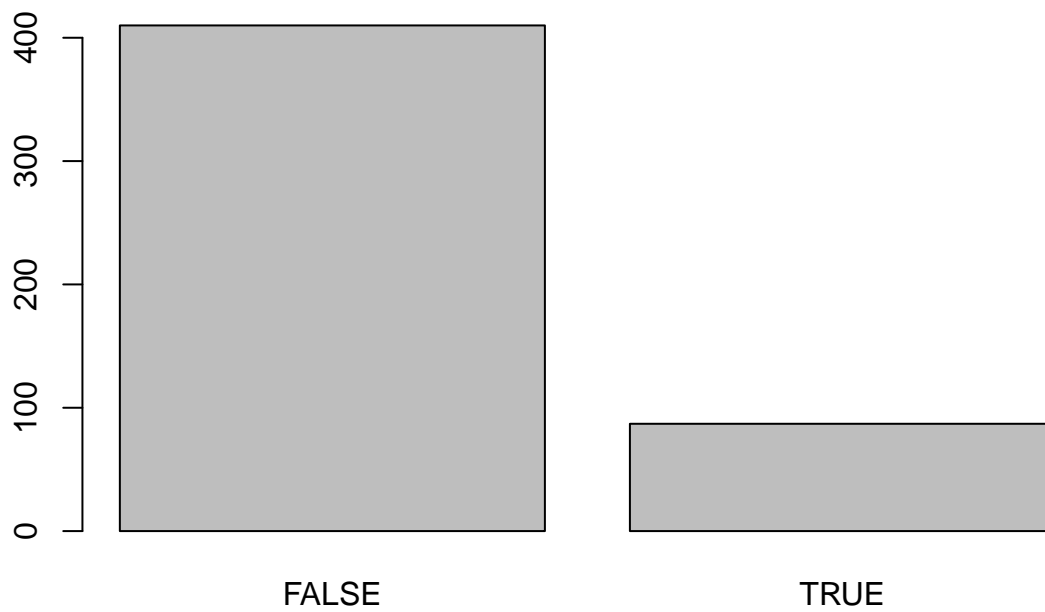
  predictions_df = predictions(df, models, cutoffs)

  result <- df_disc(predictions_df)
  # print(result)
}

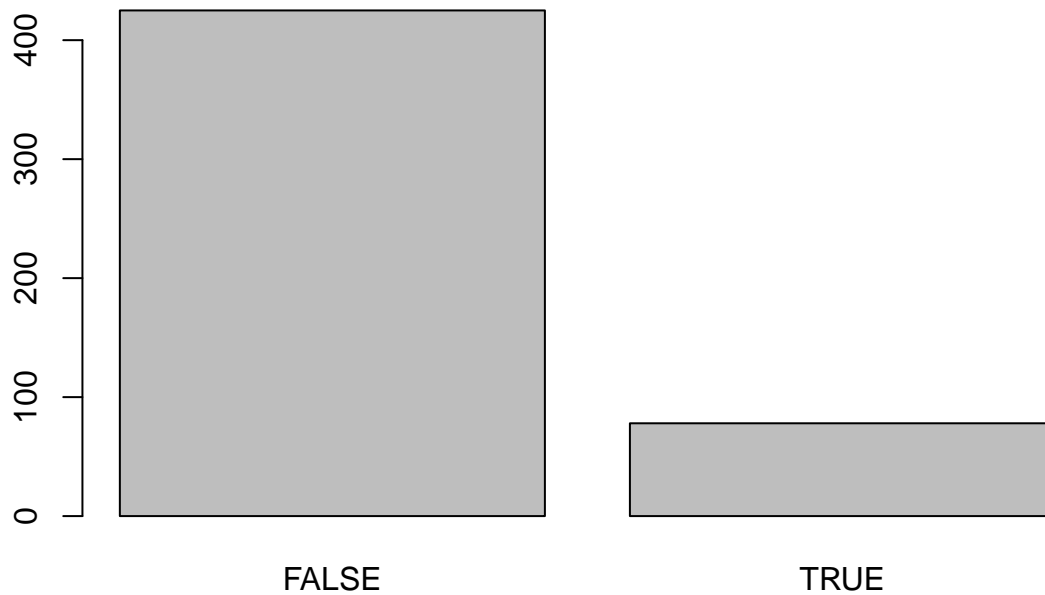
print(cutoffs)
```

```
##      Dutch Non_Dutch
##      0.70      0.02
```

```
plot(filter(predictions_df, nationality == "Dutch") %>% select(predicted))
```



```
plot(filter(predictions_df, nationality == "Non_Dutch") %>% select(predicted))
```



## Metrics

```
predicted_test <- predictions(df_test, models, cutoffs)
print(group_fairness(predicted_test, nationality, predicted)[[1]])
```

```
## # A tibble: 4 x 4
##   nationality predicted total  perc
##   <fct>         <fct>    <int> <dbl>
## 1 Dutch        FALSE      404  80.2
## 2 Dutch        TRUE       100  19.8
## 3 Non_Dutch    FALSE      433  87.3
## 4 Non_Dutch    TRUE        63  12.7
```

```
print(causal_discrimination_joined_model(df_test, models, cutoffs)[[1]])
```

```
## [1] 0.071
```