Model Evaluation

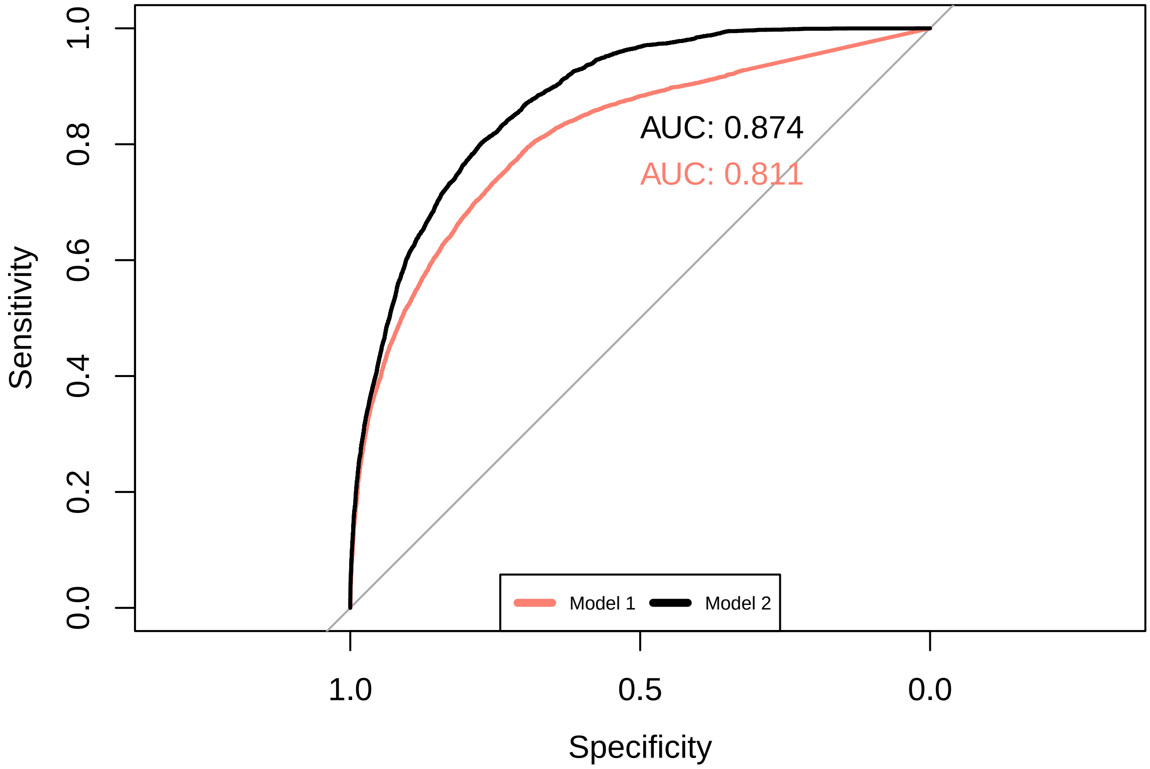
2023-06-28

Task 2: evaluation of predictive models

Evaluation metrics for a binary classification problem:

(a) AUC-ROC comparison

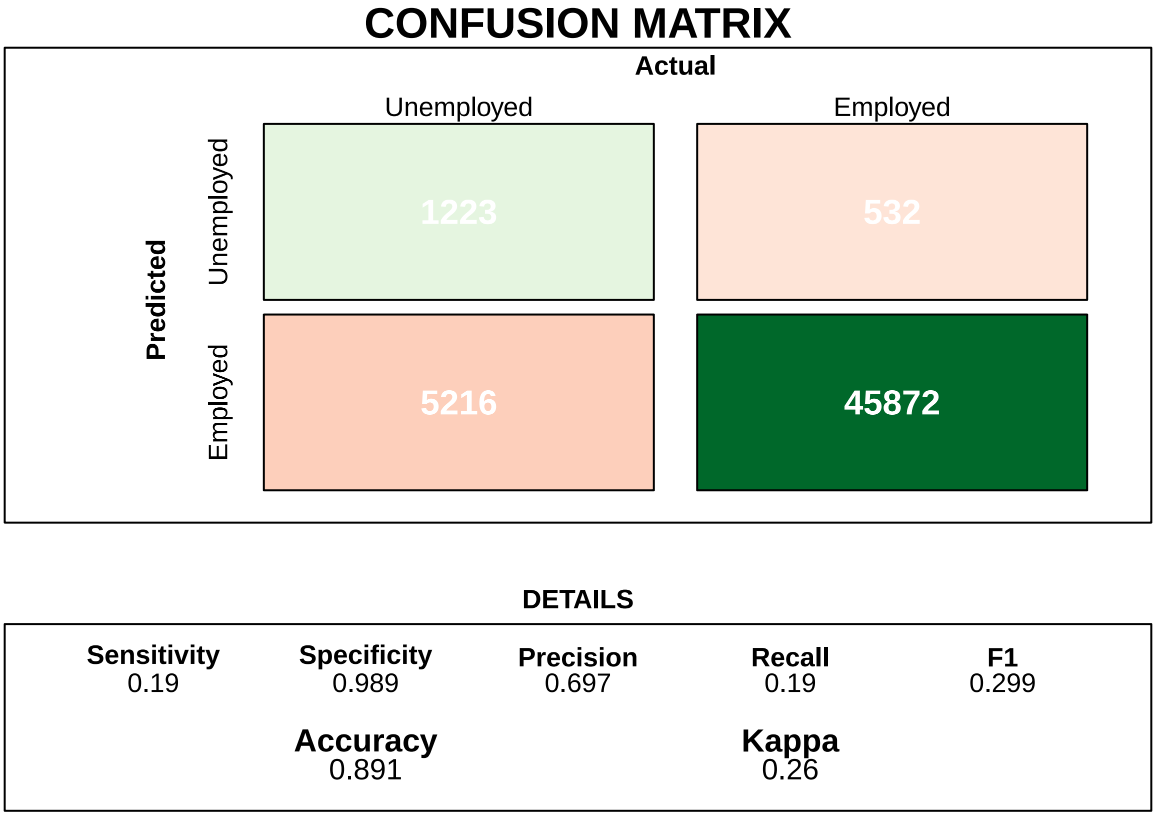
```
setwd('/Users/asya/Dropbox/_ jobs/CPL DS Assessment')
library(tidyverse)
library(pROC)
library(caret)

# load df and transform dates into R Date format
DF_edd <- read.csv('EDD.csv') %>%
  mutate(ui_start_date = as.Date(ui_start_date))

# ROC-AUC
roc_m1 <- roc(DF_edd$unemployed, DF_edd$m1_pred_prob)
roc_m2 <- roc(DF_edd$unemployed, DF_edd$m2_pred_prob)

plot(roc_m1, print.auc=TRUE, percent=TRUE, col="salmon", print.auc.y = .77)
plot(roc_m2, print.auc=TRUE, percent=TRUE, add=TRUE, print.auc.y = .85)
legend("bottom",
       legend=c("Model 1", "Model 2"),
       col=c("salmon", "black"),
       lwd=4, cex = 0.6, xpd = TRUE, horiz = TRUE)
```



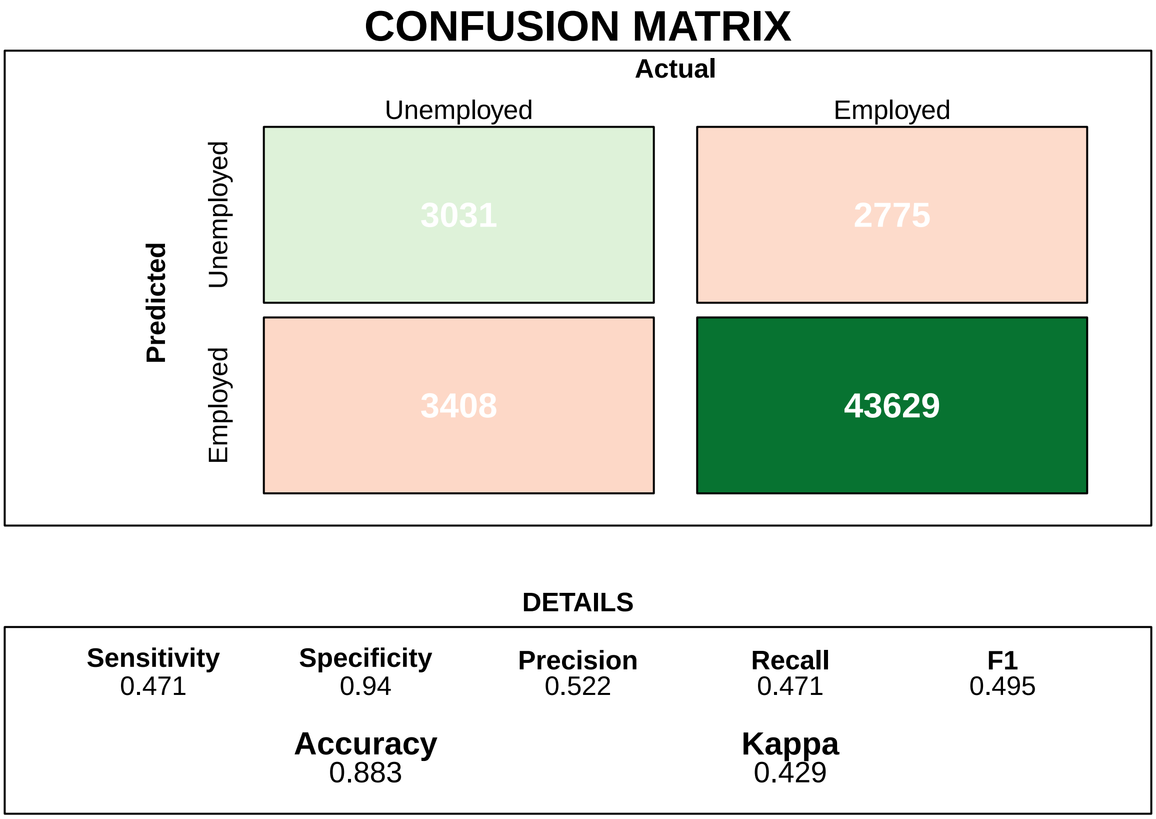
(b) Confusion matrices

```r
lvs <- c("normal", "abnormal")
truth <- factor(rep(lvs, times = c(86, 258)),
                levels = rev(lvs))
pred <- factor(
               c(
                 rep(lvs, times = c(54, 32)),
                 rep(lvs, times = c(27, 231))),
               levels = rev(lvs))

xtab <- table(pred, truth)

# confusion matrix
# transform probas into boolean vectors with predictions

### pred_prob == unemployed outcome
### first lvl == 'neg' result (ie unemployed)
DF_edd <- DF_edd %>%
  mutate(m1_pred = ifelse(m1_pred_prob > 0.5, 1, 0),
         m2_pred = ifelse(m2_pred_prob > 0.5, 1, 0))

truth <- factor(DF_edd$unemployed,
                  levels = c(1, 0),
                  labels = c("Unemployed", "Employed"))
```

```
pred_1 <- factor(DF_edd$m1_pred,
                 levels = c(1, 0),
                  labels = c("Unemployed", "Employed"))
pred_2 <- factor(DF_edd$m2_pred,
                 levels = c(1, 0),
                  labels = c("Unemployed", "Employed"))

cm_1 <- confusionMatrix(table(pred_1, truth))
cm_2 <- confusionMatrix(table(pred_2, truth))


# a function from Stakeoverflow I found a while ago
draw_confusion_matrix <- function(cm) {

  total <- sum(cm$table)
  res <- as.numeric(cm$table)

  # Generate color gradients. Palettes come from RColorBrewer.
  greenPalette <- c("#F7FCF5","#E5F5E0","#C7E9C0","#A1D99B","#74C476","#41AB5D","#238B45","#006D2C","#00
  redPalette <- c("#FFF5F0","#FEE0D2","#FCBBA1","#FC9272","#FB6A4A","#EF3B2C","#CB181D","#A50F15","#670
  getColor <- function (greenOrRed = "green", amount = 0) {
    if (amount == 0)
      return("#FFFFFF")
    palette <- greenPalette
    if (greenOrRed == "red")
      palette <- redPalette
    colorRampPalette(palette)(100)[10 + ceiling(90 * amount / total)]
  }

  # set the basic layout
  layout(matrix(c(1,1,2)))
  par(mar=c(2,2,2,2))
  plot(c(100, 345), c(300, 450), type = "n", xlab="", ylab="", xaxt='n', yaxt='n')
  title('CONFUSION MATRIX', cex.main=2)

  # create the matrix
  classes = colnames(cm$table)
  rect(150, 430, 240, 370, col=getColor("green", res[1]))
  text(195, 435, classes[1], cex=1.2)
  rect(250, 430, 340, 370, col=getColor("red", res[3]))
  text(295, 435, classes[2], cex=1.2)
  text(125, 370, 'Predicted', cex=1.3, srt=90, font=2)
  text(245, 450, 'Actual', cex=1.3, font=2)
  rect(150, 305, 240, 365, col=getColor("red", res[2]))
  rect(250, 305, 340, 365, col=getColor("green", res[4]))
  text(140, 400, classes[1], cex=1.2, srt=90)
  text(140, 335, classes[2], cex=1.2, srt=90)

  # add in the cm results
  text(195, 400, res[1], cex=1.6, font=2, col='white')
  text(195, 335, res[2], cex=1.6, font=2, col='white')
  text(295, 400, res[3], cex=1.6, font=2, col='white')
  text(295, 335, res[4], cex=1.6, font=2, col='white')
  # add in the specifics
  plot(c(100, 0), c(100, 0), type = "n", xlab="", ylab="", main = "DETAILS", xaxt='n', yaxt='n')
  text(10, 85, names(cm$byClass[1]), cex=1.2, font=2)
  text(10, 70, round(as.numeric(cm$byClass[1]), 3), cex=1.2)
  text(30, 85, names(cm$byClass[2]), cex=1.2, font=2)
  text(30, 70, round(as.numeric(cm$byClass[2]), 3), cex=1.2)
  text(50, 85, names(cm$byClass[5]), cex=1.2, font=2)
  text(50, 70, round(as.numeric(cm$byClass[5]), 3), cex=1.2)
  text(70, 85, names(cm$byClass[6]), cex=1.2, font=2)
  text(70, 70, round(as.numeric(cm$byClass[6]), 3), cex=1.2)
  text(90, 85, names(cm$byClass[7]), cex=1.2, font=2)
  text(90, 70, round(as.numeric(cm$byClass[7]), 3), cex=1.2)

  # add in the accuracy information
  text(30, 35, names(cm$overall[1]), cex=1.5, font=2)
  text(30, 20, round(as.numeric(cm$overall[1]), 3), cex=1.4)
  text(70, 35, names(cm$overall[2]), cex=1.5, font=2)
  text(70, 20, round(as.numeric(cm$overall[2]), 3), cex=1.4)
}


# maybe inferior to Python's Seaborn but still nice
par(mfrow=c(1,2))
#png(filename="/Users/asya/Dropbox/_ jobs/CPL DS Assessment/cm_1.png")
draw_confusion_matrix(cm_1)
```



```
#dev.off()


draw_confusion_matrix(cm_2)
```



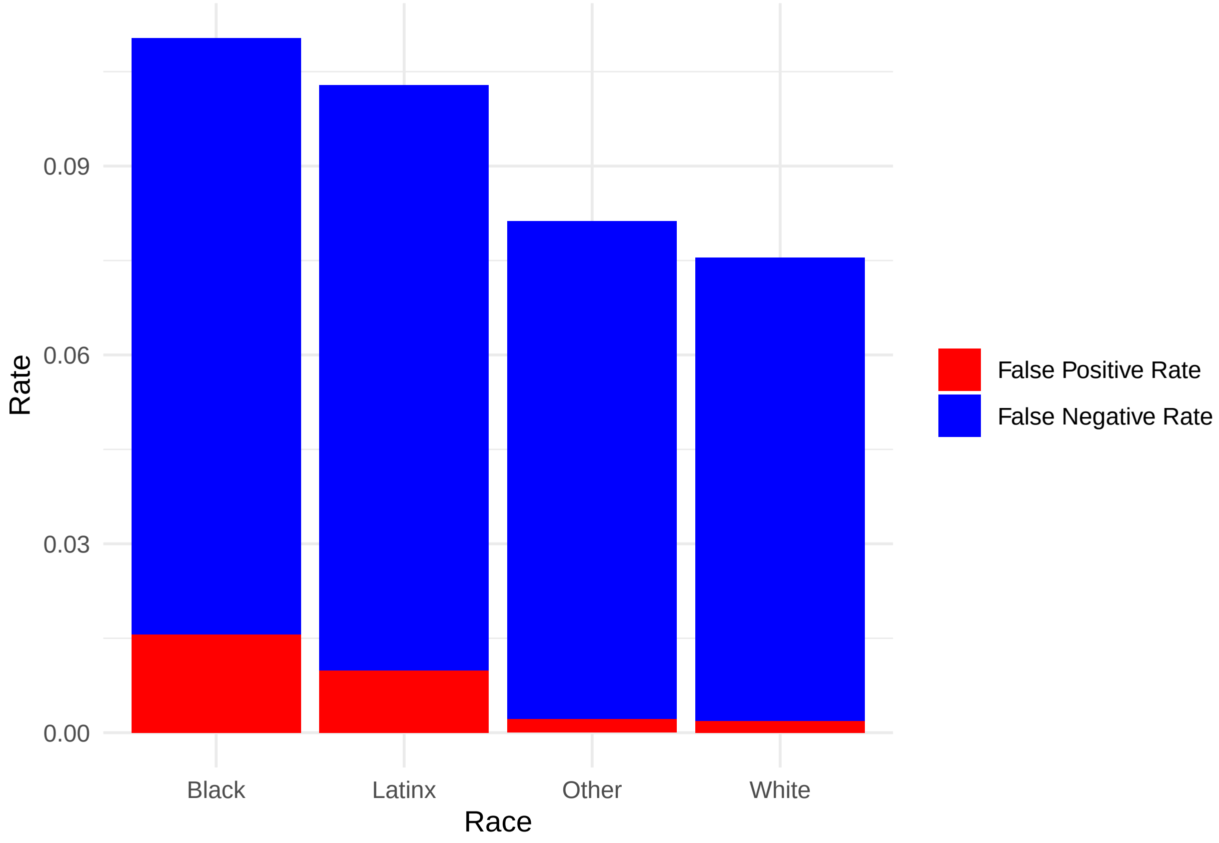
```
# RACE
### m1
fpr_fnr_1 <- DF_edd %>%
  group_by(race) %>%
  summarise(FNR = mean(ifelse(m1_pred_prob > 0.5 & unemployed == 0, 1, 0)),
            FPR = mean(ifelse(m1_pred_prob <= 0.5 & unemployed == 1, 1, 0)))
fpr_fnr_1
```

```
# A tibble: 4 x 3
  race        FNR    FPR
  <chr>     <dbl>  <dbl>
1 Black  0.0156   0.110
2 Latinx 0.00985  0.103
3 Other  0.00213  0.0812
4 White  0.00187  0.0754
```

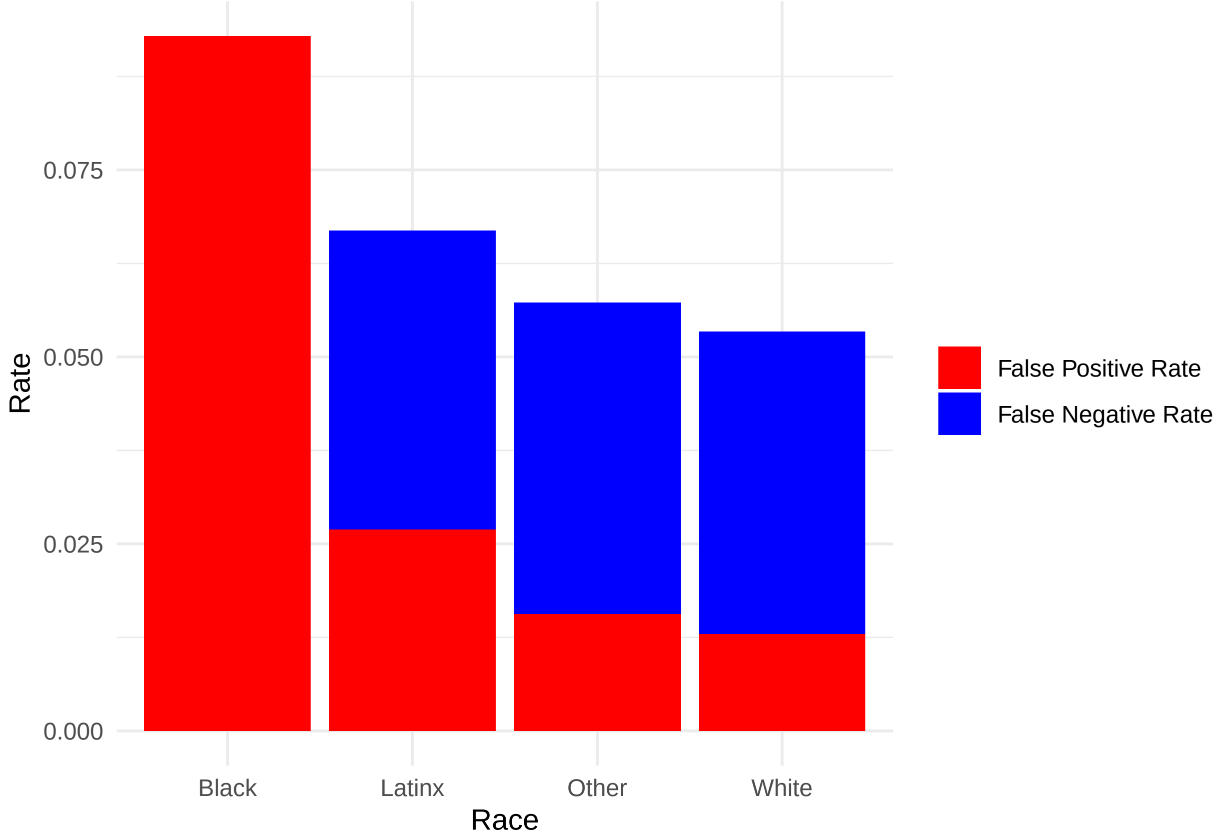
```
## m2
fpr_fnr_2 <- DF_edd %>%
  group_by(race) %>%
  summarise(FNR = mean(ifelse(m2_pred_prob > 0.5 & unemployed == 0, 1, 0)),
            FPR = mean(ifelse(m2_pred_prob <= 0.5 & unemployed == 1, 1, 0)))
fpr_fnr_2
```

```
# A tibble: 4 x 3
  race       FNR    FPR
  <chr>    <dbl>  <dbl>
1 Black  0.0929 0.0695
2 Latinx 0.0269 0.0669
3 Other  0.0156 0.0573
4 White  0.0130 0.0534
```

```
# m1: more bias against black and latin clients

ggplot(fpr_fnr_1, aes(x = race)) +
  geom_bar(aes(y = FPR, fill = "FPR"), stat = "identity", position = "dodge") +
  geom_bar(aes(y = FNR, fill = "FNR"), stat = "identity", position = "dodge") +
  labs(x = "Race", y = "Rate", fill = NULL) +
  scale_fill_manual(values = c("FPR" = "blue", "FNR" = "red"),
                    labels = c("False Positive Rate", "False Negative Rate")) +
  theme_minimal()
```



```
ggplot(fpr_fnr_2, aes(x = race)) +
  geom_bar(aes(y = FPR, fill = "FPR"), stat = "identity", position = "dodge") +
  geom_bar(aes(y = FNR, fill = "FNR"), stat = "identity", position = "dodge") +
  labs(x = "Race", y = "Rate", fill = NULL) +
  scale_fill_manual(values = c("FPR" = "blue", "FNR" = "red"),
                    labels = c("False Positive Rate", "False Negative Rate")) +
  theme_minimal()
```



```
# ----------------

# AGE
### no visible discrimination,
### models make more errors in the age group of 30s
### but it looks like it's due to the models sensitivity to unbalanced data
### (we have more observations for people in their 30s)
DF_edd %>%
  mutate(age_cat = case_when(
    age < 40 ~ 30,
    age >= 40 & age < 50 ~ 40,
    age >= 50 ~ 50
  )) %>%
  group_by(age_cat) %>%
  summarise(FNR = mean(ifelse(m1_pred_prob > 0.5 & unemployed == 0, 1, 0)),
            FPR = mean(ifelse(m1_pred_prob <= 0.5 & unemployed == 1, 1, 0)))
```

```
# A tibble: 3 x 3
  age_cat     FNR    FPR
```

```
    <dbl>   <dbl>  <dbl>
1      30 0.0113  0.106
2      40 0.0102  0.0925
3      50 0.00741 0.0894
```

```r
DF_edd %>%
  mutate(age_cat = case_when(
    age < 40 ~ 30,
    age >= 40 & age < 50 ~ 40,
    age >= 50 ~ 50
  )) %>%
  group_by(age_cat) %>%
  summarise(FNR = mean(ifelse(m2_pred_prob > 0.5 & unemployed == 0, 1, 0)),
            FPR = mean(ifelse(m2_pred_prob <= 0.5 & unemployed == 1, 1, 0)))
```

```
# A tibble: 3 x 3
  age_cat     FNR     FPR
    <dbl>  <dbl>  <dbl>
1      30 0.0554 0.0680
2      40 0.0483 0.0621
3      50 0.0501 0.0592
```

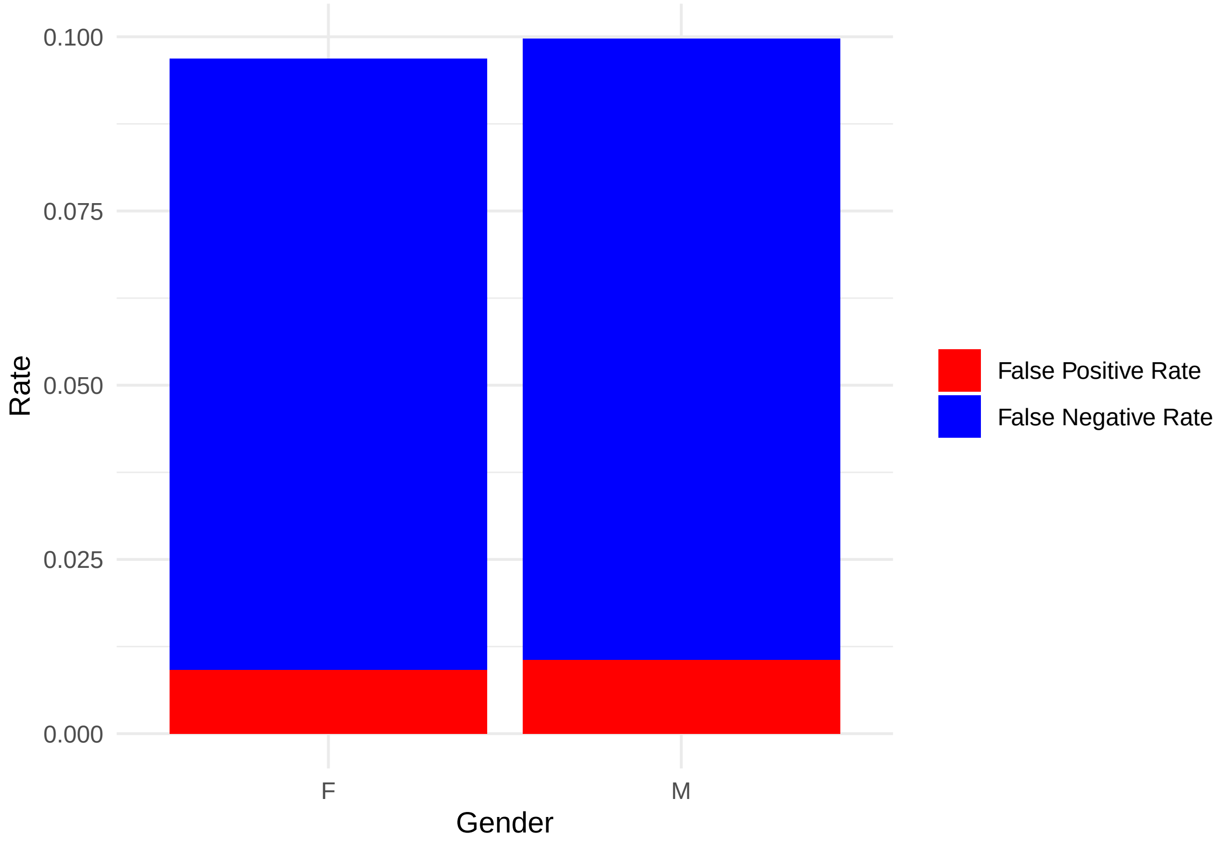
```r
DF_edd %>%
  mutate(age_cat = case_when(
    age < 40 ~ 30,
    age >= 40 & age < 50 ~ 40,
    age >= 50 ~ 50
  )) %>%
  group_by(age_cat) %>%
  summarise(len = length(age_cat))
```

```
# A tibble: 3 x 2
  age_cat   len
    <dbl> <int>
1      30 27907
2      40 11843
3      50 13093
```

```r
# ----------------
# SEX
fpr_fnr_sex1 <- DF_edd %>%
  group_by(sex) %>%
  summarise(FNR = mean(ifelse(m1_pred_prob > 0.5 & unemployed == 0, 1, 0)),
            FPR = mean(ifelse(m1_pred_prob <= 0.5 & unemployed == 1, 1, 0)))

fpr_fnr_sex1
```

```
# A tibble: 2 x 3
  sex        FNR    FPR
  <chr>    <dbl>  <dbl>
1 F      0.00915 0.0969
2 M      0.0106  0.0998
```

```
ggplot(fpr_fnr_sex1, aes(x = sex)) +
  geom_bar(aes(y = FPR, fill = "FPR"), stat = "identity", position = "dodge") +
  geom_bar(aes(y = FNR, fill = "FNR"), stat = "identity", position = "dodge") +
  labs(x = "Gender", y = "Rate", fill = NULL) +
  scale_fill_manual(values = c("FPR" = "blue", "FNR" = "red"),
                    labels = c("False Positive Rate", "False Negative Rate")) +
  theme_minimal()
```



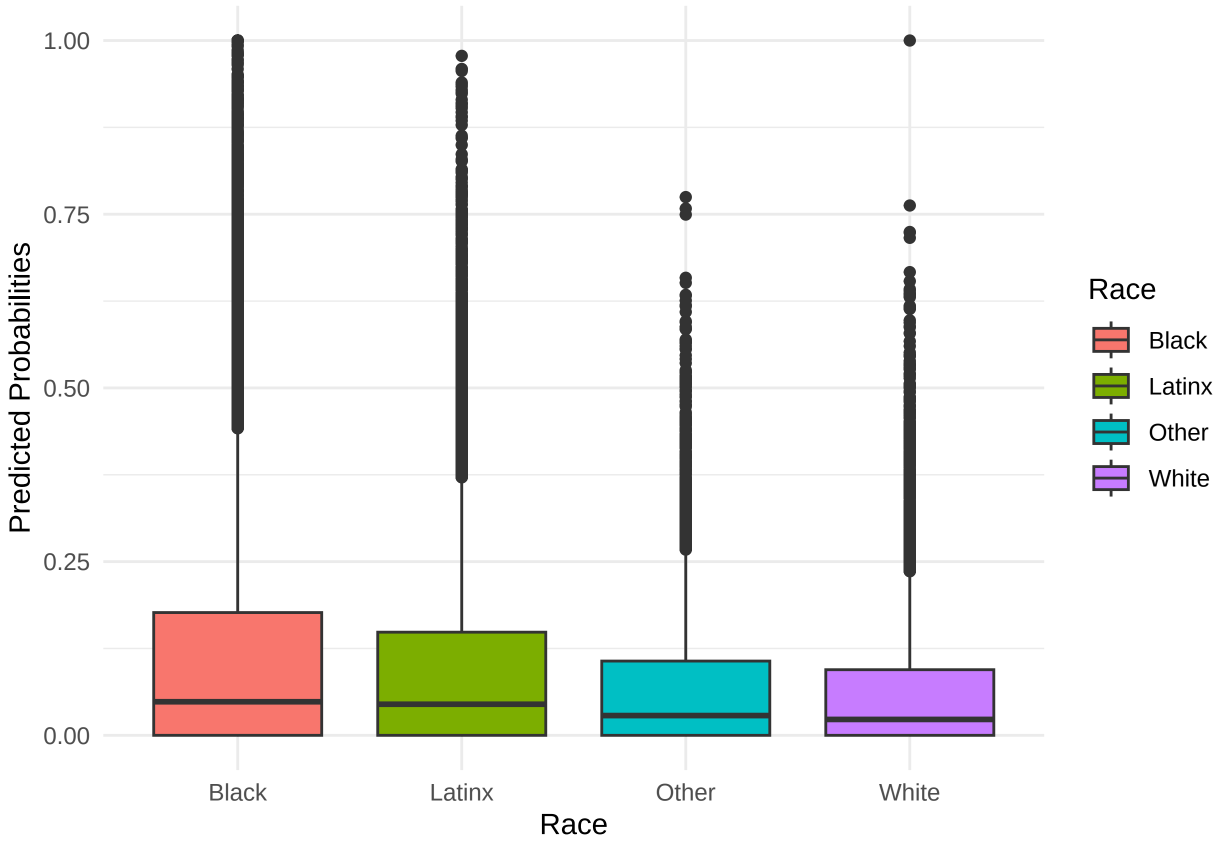
```
## m2
DF_edd %>%
  group_by(sex) %>%
  summarise(FNR = mean(ifelse(m1_pred_prob > 0.5 & unemployed == 0, 1, 0)),
            FPR = mean(ifelse(m1_pred_prob <= 0.5 & unemployed == 1, 1, 0)))
```

```
# A tibble: 2 x 3
  sex        FNR    FPR
  <chr>    <dbl>  <dbl>
1 F      0.00915 0.0969
2 M      0.0106  0.0998
```

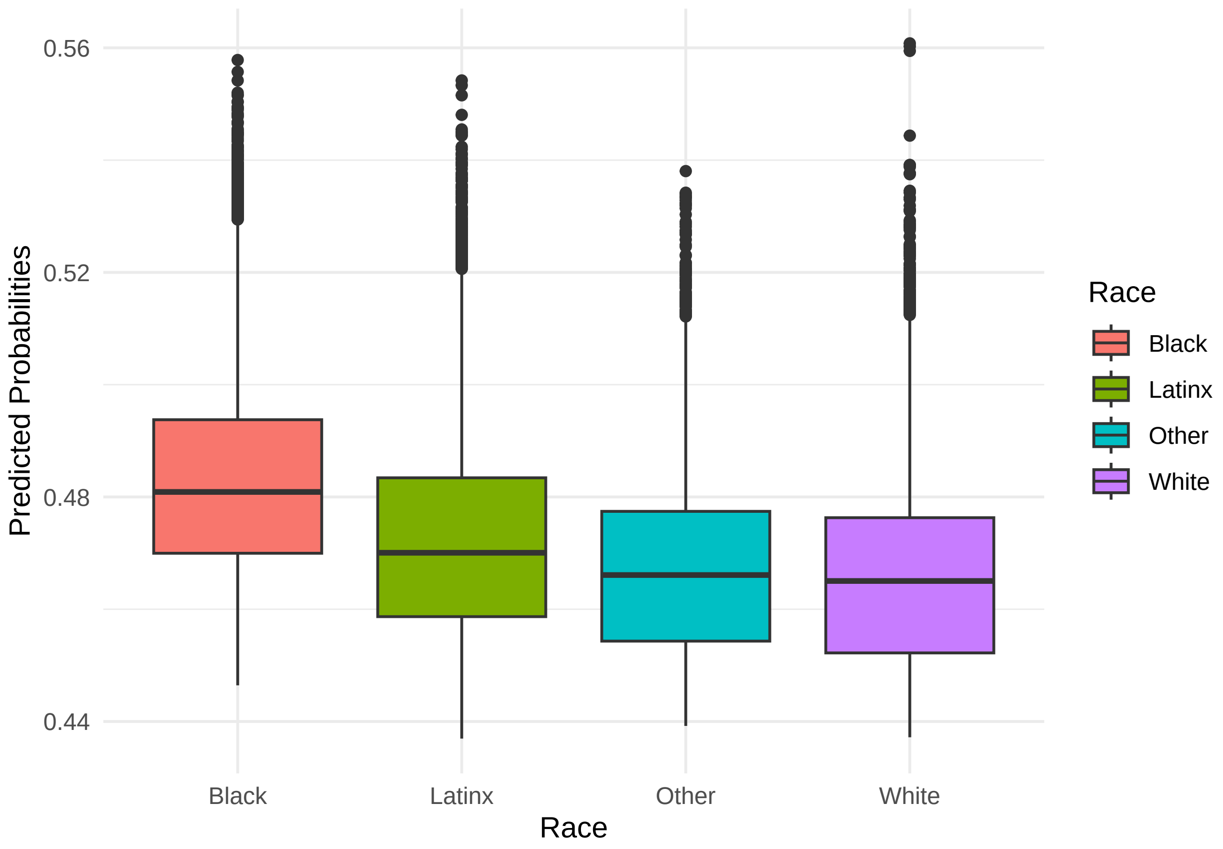
```
# balanced more or less
```

```
library(ggplot2)
```

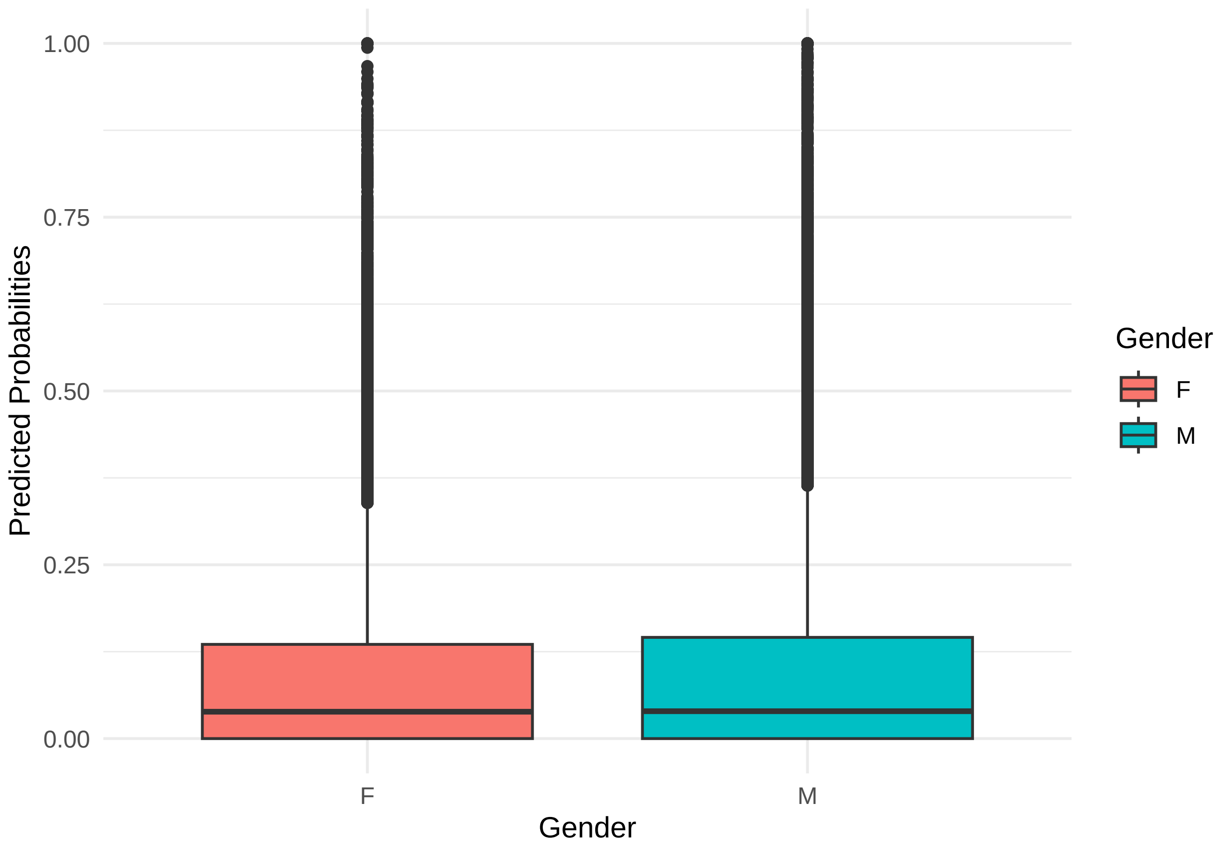
```
# RACE
ggplot(DF_edd, aes(x = race,
                   y = m1_pred_prob, fill = race, colour = unemployed)) +
  geom_boxplot() +
  labs(x = "Race", y = "Predicted Probabilities", fill = "Race") +
  theme_minimal()
```



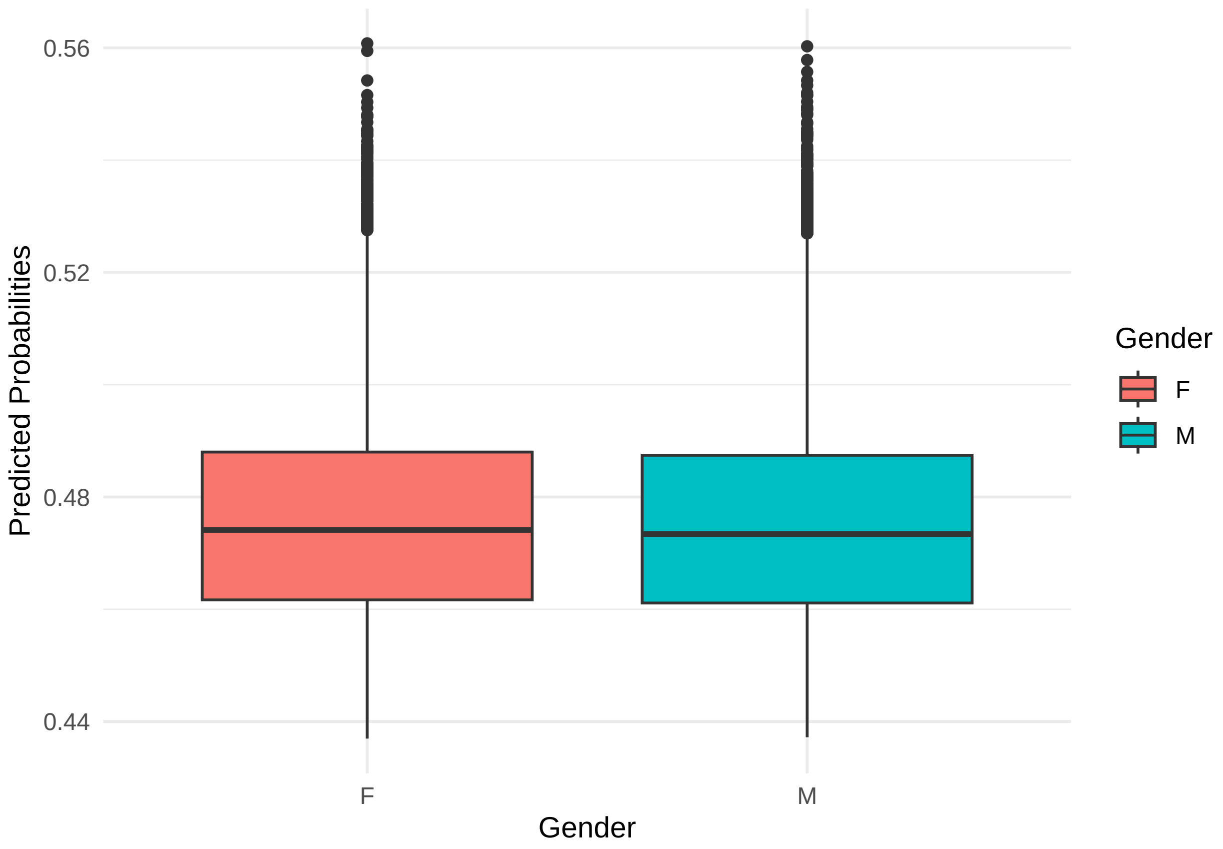
```
ggsave("m1_by_race.png")

ggplot(DF_edd, aes(x = race,
                   y = m2_pred_prob, fill = race)) +
  geom_boxplot() +
  labs(x = "Race", y = "Predicted Probabilities", fill = "Race") +
  theme_minimal()
```



```r
ggsave("m2_by_race.png")


# GENDER
ggplot(DF_edd, aes(x = sex,
                   y = m1_pred_prob, fill = sex)) +
  geom_boxplot() +
  labs(x = "Gender", y = "Predicted Probabilities", fill = "Gender") +
  theme_minimal()
```



```
ggsave("m1_by_gender.png")

ggplot(DF_edd, aes(x = sex,
                   y = m2_pred_prob, fill = sex)) +
  geom_boxplot() +
  labs(x = "Gender", y = "Predicted Probabilities", fill = "Gender") +
  theme_minimal()
```



```
ggsave("m2_by_gender.png")
```