judgment day

```r
library(reticulate)
library(data.table)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::between()   masks data.table::between()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::transpose() masks data.table::transpose()
```

```r
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
use_virtualenv("../venv/", required = TRUE)
source_python("util.py")
source_python("models.py")
```

Functions

```r
calculate_yhat_lr <- function(model, x) {
  return(model$predict_proba(x)[,2])
}

calculate_yhat_NN <- function(model, x) {
  return(model$predict(x))
}

calculate_TPR <- function(y, yhat) {
  P <- sum(y == 1)
  TP <- sum(((yhat > 0.5) == 1) & (y == 1))
  TPR <- TP / P
```

```
  return(TPR)
}

calculate_AUC <- function(y, yhat) {
  # https://stackoverflow.com/questions/50866797/how-do-i-calculate-auc-from-two-continuous-variables-i
  df <- data.frame(a=y, p=yhat)
  df <- df[order(df$a),]
  roc_obj <- roc(df$a, df$p)
  AUC <- auc(roc_obj)
  TPR10 <- coords(roc_obj, x=0.9, input="specificity")$sensitivity
  return(c(AUC=AUC, TPR10=TPR10))
}

# This function assumes that the data variables are already defined in the environment.
# E.g. Black_x, AsianPI_y, etc.
calculate_results <- function(model, model_yhat_fn) {
  # (1) Prepare results matrix
  race_eth_all <- c('White', 'Black', 'AsianPI', 'AmeriIndian')
  results <- matrix(nrow = length(race_eth_all), ncol = 3)
  rownames(results) <- race_eth_all
  colnames(results) <- c('TPR', 'AUC', 'TPR10')

  # (2) Calculate TPR, AUC, and TPR10 for each race/ethnicity
  for (i in 1:length(race_eth_all)) {
    race_eth <- race_eth_all[i]
    x <- get(paste0(race_eth, '_x'))  # get() gets a variable in the environment by name
    y <- get(paste0(race_eth, '_y'))
    yhat <- model_yhat_fn(model, x)  # Note this depends on function arguments

    results[i, 1] <- calculate_TPR(y, yhat)
    results[i, c(2, 3)] <- calculate_AUC(y, yhat)
  }

  return(as_tibble(results))
}
```

Bias?? hm

Late Stillbirth

```
x <- fread('../data/final/stillbirth_test.csv')
outcome <- 'late stillbirth'
x$outcome <- as.numeric(x$outcome == outcome)

AmeriIndian_x <- x %>% filter(race_AmeriIndian == 1)
AmeriIndian_y <- AmeriIndian_x$outcome
AmeriIndian_x <- AmeriIndian_x %>% select(-outcome)

AsianPI_x <- x %>% filter(race_AsianPI == 1)
AsianPI_y <- AsianPI_x$outcome
AsianPI_x <- AsianPI_x %>% select(-outcome)
```

```r
Black_x <- x %>% filter(race_Black == 1)
Black_y <- Black_x$outcome
Black_x <- Black_x %>% select(-outcome)

White_x <- x %>% filter(race_White == 1)
White_y <- White_x$outcome
White_x <- White_x %>% select(-outcome)
```

SELU Network

```r
selu_late <- load_NN("../models/selu_late")
calculate_results(selu_late, calculate_yhat_NN)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## # A tibble: 4 x 3
##     TPR   AUC TPR10
##   <dbl> <dbl> <dbl>
## 1 0.956 0.554 0.113
## 2 0.905 0.588 0.122
## 3 0.908 0.539 0.109
## 4 1     0.553 0.112
```

```r
mean((selu_late$predict(AmeriIndian_x) > 0.5) == AmeriIndian_y)
```

```
## [1] 0.07696349
```

```r
mean((selu_late$predict(AsianPI_x) > 0.5) == AsianPI_y)
```

```
## [1] 0.1490291
```

```r
mean((selu_late$predict(Black_x) > 0.5) == Black_y)
```

```
## [1] 0.2801312
```

```r
mean((selu_late$predict(White_x) > 0.5) == White_y)
```

```
## [1] 0.1307677
```

Early Stillbirth

```r
x <- fread('../data/final/stillbirth_test.csv')
outcome <- 'early stillbirth'
x$outcome <- as.numeric(x$outcome == outcome)
```

```r
AmeriIndian_x <- x %>% filter(race_AmeriIndian == 1)
AmeriIndian_y <- AmeriIndian_x$outcome
AmeriIndian_x <- AmeriIndian_x %>% select(-outcome)

AsianPI_x <- x %>% filter(race_AsianPI == 1)
AsianPI_y <- AsianPI_x$outcome
AsianPI_x <- AsianPI_x %>% select(-outcome)

Black_x <- x %>% filter(race_Black == 1)
Black_y <- Black_x$outcome
Black_x <- Black_x %>% select(-outcome)

White_x <- x %>% filter(race_White == 1)
White_y <- White_x$outcome
White_x <- White_x %>% select(-outcome)
```

Logistic Regression

```r
lr_early <- load_pickle("../models/lr_early")
calculate_results(model = lr_early, model_yhat_fn = calculate_yhat_lr)
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## # A tibble: 4 x 3
##     TPR   AUC TPR10
##   <dbl> <dbl> <dbl>
## 1 0.632 0.774 0.476
## 2 0.870 0.812 0.525
## 3 0.615 0.805 0.522
## 4 0.567 0.674 0.367
```

LightGBM

```r
gb_early <- load_pickle("../models/gb_early")
calculate_results(model = gb_early, model_yhat_fn = calculate_yhat_NN)
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## # A tibble: 4 x 3
##     TPR   AUC TPR10
##   <dbl> <dbl> <dbl>
## 1 0.290 0.645 0.361
## 2 0.389 0.694 0.450
## 3 0.396 0.698 0.456
## 4 0.3   0.650 0.370
```

Notes

Further research would also include inspecting the reproducibility of the results beyond the population of the United States with different ML models. The best machine learning models (SELU network, LGBM and averaged ensemble) were able to produce repeatable performance over two data sets. Using these machine learning models, especially for early stillbirth, could provide earlier identification of at-risk pregnancies with high accuracy and provide tools for better utilization of healthcare resources targeted to those needing it most.