Pertussis Challenge 1.1

Runqi Zhang

2024-11-21

```
# Load necessary libraries
suppressPackageStartupMessages({
  library(dplyr)         # Data manipulation
  library(tidyr)         # Data reshaping
  library(readr)         # Reading TSV files
  library(ggplot2)       # Visualization
  library(glmnet)        # Regularized regression (LASSO, Ridge)
  library(agua)          # H2O AutoML integration
  library(knitr)         # Knitting reports
  library(tibble)
})
```

```
## Warning: package 'tidyr' was built under R version 4.3.3
```

```
## Warning: package 'readr' was built under R version 4.3.3
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
## Warning: package 'glmnet' was built under R version 4.3.3
```

```
## Warning: package 'agua' was built under R version 4.3.3
```

```
## Warning: package 'parsnip' was built under R version 4.3.3
```

```
## Warning: package 'knitr' was built under R version 4.3.3
```

```
# Set working directory and initialize H2O
workDir <- "C:/Users/zhang/Desktop/cmi-pb-3rd-final/Runqi/CMI-PB"
agua::h2o_start()
```

```
## Warning: JAVA not found, H2O may take minutes trying to connect.
```

```
## Warning in h2o.clusterInfo():
## Your H2O cluster version is (11 months) old. There may be a newer version available.
## Please download and install the latest version from: https://h2o-release.s3.amazonaws.com/h2o/latest_
```

```r
# Read input data for 2020-2023
options(readr.show_col_types = FALSE)

read_data <- function(year) {
  list(
    pts = read_tsv(file = file.path(workDir, paste0("data/", year, "LD_subject.tsv"))),
    sample = read_tsv(file = file.path(workDir, paste0("data/", year, "LD_specimen.tsv"))),
    ab = read_tsv(file = file.path(workDir, paste0("data/", year, "LD_plasma_ab_titer.tsv")))
  )
}

data2020 <- read_data("2020")
data2021 <- read_data("2021")
data2022 <- read_data("2022")
data2023 <- list(
  pts = read_tsv(file = file.path(workDir, "data/2023BD_subject.tsv")),
  sample = read_tsv(file = file.path(workDir, "data/2023BD_specimen.tsv")),
  ab = read_tsv(file = file.path(workDir, "data/2023BD_plasma_ab_titer.tsv"))
)
```

1 Challenge1.1

```r
# Prepare aligned target variables for training
yDF <- data2020$ab %>%
  filter(isotype == "IgG", antigen == "PT") %>%
  inner_join(data2020$sample, by = "specimen_id") %>%
  inner_join(data2020$pts, by = "subject_id") %>%
  filter(planned_day_relative_to_boost == 14) %>%
  dplyr::select(subject_id, MFI_normalised)

yDF2 <- data2021$ab %>%
  filter(isotype == "IgG", antigen == "PT") %>%
  inner_join(data2021$sample, by = "specimen_id") %>%
  inner_join(data2021$pts, by = "subject_id") %>%
  filter(planned_day_relative_to_boost == 14) %>%
  dplyr::select(subject_id, MFI_normalised)


yDF3 <- data2022$ab %>%
  filter(isotype == "IgG", antigen == "PT") %>%
  inner_join(data2022$sample, by = "specimen_id") %>%
  inner_join(data2022$pts, by = "subject_id") %>%
  filter(planned_day_relative_to_boost == 14) %>%
  dplyr::select(subject_id, MFI_normalised) #%>%
```

2 pts info

```r
# Function to prepare predictors for Day 0 baseline IgG levels
prepare_predictors <- function(ab_data, sample_data, pts_data) {
  ab_data %>%
    inner_join(sample_data, by = "specimen_id") %>%
    inner_join(pts_data, by = "subject_id") %>%
    filter(planned_day_relative_to_boost == 0, grepl("IgG", isotype)) %>%
    mutate(cname = paste0(isotype, "_", antigen)) %>%
    dplyr::select(subject_id, cname, MFI_normalised) %>%
    pivot_wider(names_from = cname, values_from = MFI_normalised) %>%
    column_to_rownames(var = "subject_id")
}

# Prepare predictors for all years
xDF <- prepare_predictors(data2020$ab, data2020$sample, data2020$pts)
x2DF <- prepare_predictors(data2021$ab, data2021$sample, data2021$pts)
x3DF <- prepare_predictors(data2022$ab, data2022$sample, data2022$pts)
x4DF <- prepare_predictors(data2023$ab, data2023$sample, data2023$pts)
```

```r
# Function to align and scale datasets based on the intersection of column names
align_and_scale <- function(datasets) {
  # Compute the intersection of column names across all datasets
  common_cols <- Reduce(intersect, lapply(datasets, colnames))
  # Align all datasets to the common columns
  aligned <- lapply(datasets, function(x) x[, common_cols, drop = FALSE])
  # Scale each dataset independently
  scaled <- lapply(aligned, scale)
  return(scaled)
}

# Apply the function to align and scale the datasets
datasets <- align_and_scale(list(xDF, x2DF, x3DF, x4DF))

# Extract the processed datasets
xDF <- datasets[[1]]
x2DF <- datasets[[2]]
x3DF <- datasets[[3]]
x4DF <- datasets[[4]]
```

```r
# Match yDF to xDF by subject_id
yDF <- yDF %>% slice(match(rownames(xDF), subject_id))
xDF <- xDF[rownames(xDF) %in% yDF$subject_id, , drop = FALSE]
yDF2 <- yDF2 %>% slice(match(rownames(x2DF), subject_id))
x2DF <- x2DF[rownames(x2DF) %in% yDF2$subject_id, , drop = FALSE]
yDF3 <- yDF3 %>% slice(match(rownames(x3DF), subject_id))
x3DF <- x3DF[rownames(x3DF) %in% yDF3$subject_id, , drop = FALSE]

trainDF <- rbind(xDF, x2DF, x3DF) %>%
  as.data.frame() %>%
  mutate(MFI_normalised = c(
    yDF$MFI_normalised,
    yDF2$MFI_normalised,
    yDF3$MFI_normalised
  ))
```

```r
# Train regression model using H2O AutoML
set.seed(3)

auto_fit <- auto_ml() %>%
  set_engine("h2o", max_runtime_secs = 5) %>%
  set_mode("regression") %>%
  fit(MFI_normalised ~ ., data = trainDF)
```

```r
# Predict on training data
train_predictions <- predict(auto_fit, new_data = trainDF)$.pred
```

```
## Warning in doTryCatch(return(expr), name, parentenv, handler): Test/Validation
## dataset is missing column ’‘IgG1_FIM2/3‘’: substituting in a column of NaN

## Warning in doTryCatch(return(expr), name, parentenv, handler): Test/Validation
## dataset is missing column ’‘IgG2_FIM2/3‘’: substituting in a column of NaN

## Warning in doTryCatch(return(expr), name, parentenv, handler): Test/Validation
## dataset is missing column ’‘IgG3_FIM2/3‘’: substituting in a column of NaN

## Warning in doTryCatch(return(expr), name, parentenv, handler): Test/Validation
## dataset is missing column ’‘IgG4_FIM2/3‘’: substituting in a column of NaN
```

```r
# Calculate correlations
pearson_cor <- cor(train_predictions, trainDF$MFI_normalised, method = "pearson")
spearman_cor <- cor(train_predictions, trainDF$MFI_normalised, method = "spearman")

# Display correlation results
cat("Pearson Correlation: ", pearson_cor, "\n")
```

```
## Pearson Correlation:  0.8856824
```

```r
cat("Spearman Correlation: ", spearman_cor, "\n")
```

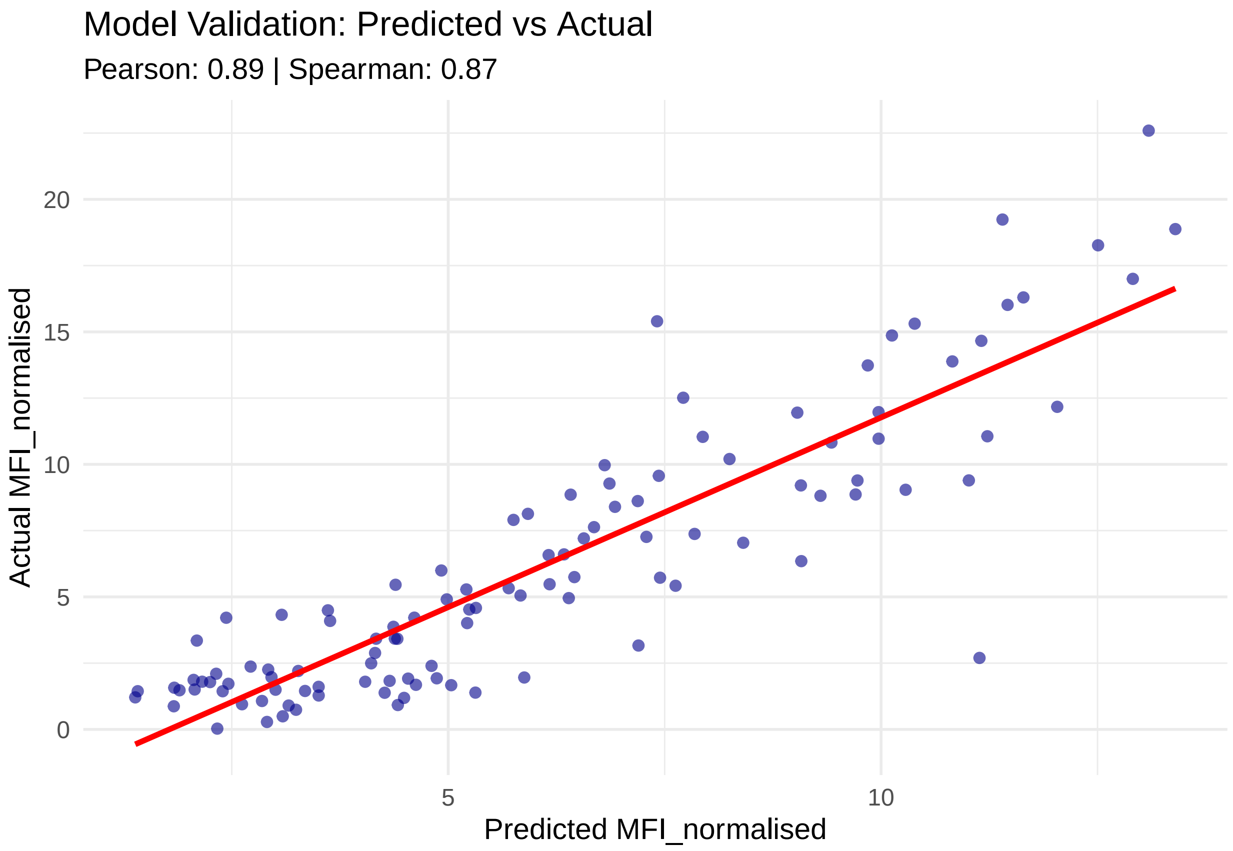
```
## Spearman Correlation:  0.8732011
```

```r
# Create a correlation plot
library(ggplot2)

correlation_plot <- ggplot(data = data.frame(
  Predicted = train_predictions,
  Actual = trainDF$MFI_normalised
), aes(x = Predicted, y = Actual)) +
  geom_point(alpha = 0.6, color = "darkblue") +  # Scatter plot
  geom_smooth(method = "lm", color = "red", se = FALSE) +  # Regression line
  labs(
    title = "Model Validation: Predicted vs Actual",
    subtitle = paste0("Pearson: ", round(pearson_cor, 2),
                      " | Spearman: ", round(spearman_cor, 2)),
    x = "Predicted MFI_normalised",
    y = "Actual MFI_normalised"
  ) +
  theme_minimal()

# Display the plot
print(correlation_plot)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
# Predict and rank for 2023
yhat <- predict(auto_fit, new_data = x4DF)$.pred
```

```
## Warning in doTryCatch(return(expr), name, parentenv, handler): Test/Validation
## dataset is missing column '`IgG1_FIM2/3`': substituting in a column of NaN

## Warning in doTryCatch(return(expr), name, parentenv, handler): Test/Validation
## dataset is missing column '`IgG2_FIM2/3`': substituting in a column of NaN

## Warning in doTryCatch(return(expr), name, parentenv, handler): Test/Validation
## dataset is missing column '`IgG3_FIM2/3`': substituting in a column of NaN

## Warning in doTryCatch(return(expr), name, parentenv, handler): Test/Validation
## dataset is missing column '`IgG4_FIM2/3`': substituting in a column of NaN
```

```r
rhat <- rank(-1 * yhat, ties.method = "first")  # Break ties deterministically
print(cbind(rownames(x4DF), rhat))
```

```
##               rhat
##  [1,] "142" "44"
##  [2,] "146" "21"
##  [3,] "163" "48"
##  [4,] "124" "46"
##  [5,] "134" "42"
##  [6,] "170" "47"
##  [7,] "132" "15"
##  [8,] "140" "9"
##  [9,] "155" "36"
## [10,] "172" "50"
## [11,] "148" "2"
## [12,] "135" "18"
## [13,] "123" "10"
## [14,] "128" "28"
## [15,] "164" "8"
## [16,] "159" "24"
## [17,] "167" "32"
## [18,] "158" "11"
## [19,] "151" "41"
## [20,] "150" "12"
## [21,] "133" "1"
## [22,] "126" "4"
## [23,] "125" "40"
## [24,] "130" "25"
## [25,] "145" "53"
## [26,] "122" "7"
## [27,] "138" "37"
## [28,] "157" "26"
## [29,] "119" "35"
## [30,] "136" "30"
## [31,] "149" "45"
## [32,] "165" "20"
## [33,] "131" "14"
## [34,] "169" "23"
## [35,] "160" "17"
## [36,] "168" "27"
## [37,] "141" "38"
## [38,] "154" "51"
## [39,] "153" "52"
## [40,] "147" "33"
## [41,] "156" "6"
## [42,] "121" "19"
## [43,] "120" "43"
## [44,] "144" "5"
## [45,] "143" "54"
## [46,] "171" "34"
## [47,] "127" "31"
## [48,] "129" "16"
## [49,] "162" "49"
## [50,] "166" "39"
## [51,] "152" "3"
## [52,] "161" "29"
## [53,] "137" "13"
## [54,] "139" "22"
```

```
# Load and update submission template
submission_file <- file.path(workDir, "3rdChallengeSubmissionTemplate_revised.tsv")
data <- read_tsv(submission_file)

# Update rankings for Challenge 1.1
ranking_df <- data.frame(
  SubjectID = as.numeric(rownames(x4DF)),
  `1.1) IgG-PT-D14-titer-Rank` = rhat,
  check.names = FALSE # Prevent automatic renaming of column names
)
data <- data %>%
  mutate(
    `1.1) IgG-PT-D14-titer-Rank` = ifelse(
      SubjectID %in% ranking_df$SubjectID,
      ranking_df$`1.1) IgG-PT-D14-titer-Rank`[match(SubjectID, ranking_df$SubjectID)],
      `1.1) IgG-PT-D14-titer-Rank`
    )
  )

# Save updated submission file
write_tsv(data, submission_file)
```

```
# End H2O session and display session info
agua::h2o_end()
sessionInfo()
```

```
## R version 4.3.1 (2023-06-16 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 11 x64 (build 22631)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## time zone: America/Los_Angeles
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
```

```
##  [1] tibble_3.2.1   knitr_1.49     agua_0.1.4     parsnip_1.2.1  glmnet_4.1-8
##  [6] Matrix_1.5-4.1 ggplot2_3.5.1  readr_2.1.5    tidyr_1.3.1    dplyr_1.1.3
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.2.1    timeDate_4041.110   farver_2.1.2
##  [4] bitops_1.0-9        fastmap_1.2.0       RCurl_1.98-1.16
##  [7] digest_0.6.37       rpart_4.1.19        timechange_0.3.0
## [10] lifecycle_1.0.4     yardstick_1.3.1     survival_3.5-5
## [13] magrittr_2.0.3      compiler_4.3.1      rlang_1.1.1
## [16] tools_4.3.1         utf8_1.2.3          yaml_2.3.10
## [19] data.table_1.16.2   labeling_0.4.3      bit_4.5.0
## [22] curl_6.0.1          DiceDesign_1.10     withr_3.0.2
## [25] purrr_1.0.2         workflows_1.1.4     h2o_3.44.0.3
## [28] nnet_7.3-19         grid_4.3.1          tune_1.2.1
## [31] fansi_1.0.4         colorspace_2.1-0    future_1.34.0
## [34] globals_0.16.3      scales_1.3.0        iterators_1.0.14
## [37] MASS_7.3-60         cli_3.6.1           crayon_1.5.3
## [40] rmarkdown_2.29      generics_0.1.3      rstudioapi_0.17.1
## [43] future.apply_1.11.3 tzdb_0.4.0          splines_4.3.1
## [46] dials_1.3.0         parallel_4.3.1      vctrs_0.6.3
## [49] hardhat_1.4.0       jsonlite_1.8.9      hms_1.1.3
## [52] bit64_4.5.2         listenv_0.9.1       foreach_1.5.2
## [55] gower_1.0.1         recipes_1.1.0       glue_1.6.2
## [58] parallelly_1.39.0   codetools_0.2-19    rsample_1.2.1
## [61] lubridate_1.9.3     shape_1.4.6.1       gtable_0.3.6
## [64] munsell_0.5.1       GPfit_1.0-8         pillar_1.9.0
## [67] furrr_0.3.1         htmltools_0.5.8.1   ipred_0.9-15
## [70] lava_1.8.0          R6_2.5.1            lhs_1.2.0
## [73] vroom_1.6.5         evaluate_1.0.1      lattice_0.21-8
## [76] class_7.3-22        Rcpp_1.0.11         nlme_3.1-162
## [79] prodlim_2024.06.25  mgcv_1.8-42         xfun_0.48
## [82] pkgconfig_2.0.3
```