```
# ==============================================================================
# TITLE: Automated Multi-omics IntegratedLearner Pipeline (Final Hardened
Version)
#        Classification, Continuous Prediction, and Interactive Visualization
#
# OUTPUT PATH: D:\DOWNLOADS
# ==============================================================================
#
# --- DATA SOURCES ---
# 1. PRISM Dataset (IBD Classification):
#    Ref: Franzosa et al. (2019) [cite_start][cite: 32-34]
#
# 2. Pregnancy Dataset (Continuous Prediction):
#    Ref: Ghaemi et al. (2019) [cite_start][cite: 254-255]
# ==============================================================================

# ------------------------------------------------------------------------------
# 1. SETUP & DEPENDENCIES
# ------------------------------------------------------------------------------

# [CRITICAL] Set Java memory to 5GB (Must be first executable line)
options(java.parameters = "-Xmx5g")

# Define Output Directory
output_dir <- "D:/DOWNLOADS"
if (!dir.exists(output_dir)) dir.create(output_dir, recursive = TRUE)

# 1.1 Install Libraries if missing
pkgs <- c("plotly", "DT", "htmlwidgets", "pROC", "IntegratedLearner",
          "bartMachine", "tidyverse", "SuperLearner", "caret", "cowplot",
"bayesplot")

for (pkg in pkgs) {
  if (!require(pkg, character.only = TRUE, quietly = TRUE)) {
     if (pkg == "IntegratedLearner") {
        if(!require("devtools")) install.packages("devtools")
        devtools::install_github("himelmallick/IntegratedLearner", upgrade =
"never")
     } else {
        install.packages(pkg)
     }
  }
}

# 1.2 Load Libraries
library(IntegratedLearner)
library(tidyverse)
library(SuperLearner)
library(caret)
library(cowplot)
library(bayesplot)
library(bartMachine)
library(plotly)
library(DT)
library(htmlwidgets)

# 1.3 Java Connection
target_java <- "C:/Program Files/Eclipse Adoptium/jdk-17.0.10.7-hotspot"
if (dir.exists(target_java)) Sys.setenv(JAVA_HOME = target_java)

message(">>> Environment Ready. Results will be saved to: ", output_dir)

# ------------------------------------------------------------------------------
# [cite_start]2. EXAMPLE 1: IBD CLASSIFICATION (Binary) [cite: 32-253]
```

```
# ------------------------------------------------------------------------
message("\n>>> [STEP 2] Processing IBD Classification Dataset...")

# Load Data
load(url("https://github.com/himelmallick/IntegratedLearner/blob/master/data/
PRISM.RData?raw=true"))
ft_train <- pcl$feature_table; sm_train <- pcl$sample_metadata; fm_train <-
pcl$feature_metadata; rm(pcl)

load(url("https://github.com/himelmallick/IntegratedLearner/blob/master/data/
NLIBD.RData?raw=true"))
ft_valid <- pcl$feature_table; sm_valid <- pcl$sample_metadata; rm(pcl)

# Train Model (Random Forest + NNLS)
fit_binary <- IntegratedLearner(
  feature_table = ft_train, sample_metadata = sm_train, feature_metadata =
fm_train,
  feature_table_valid = ft_valid, sample_metadata_valid = sm_valid,
  folds = 5, base_learner = 'SL.randomForest', meta_learner = 'SL.nnls.auc',
  verbose = TRUE, family = binomial()
)

# --- 2.1 ROC Curve (PNG + HTML) ---
message("Saving ROC Curve...")

tryCatch({
  # Capture built-in plot
  IntegratedLearner:::plot.learner(fit_binary)
  p_roc_captured <- ggplot2::last_plot()

  # Save Files
  ggsave(filename = file.path(output_dir, "1_IBD_ROC_Curve.png"), plot =
p_roc_captured, width = 8, height = 6, dpi = 300)
  saveWidget(ggplotly(p_roc_captured), file.path(output_dir,
"1_Interactive_IBD_ROC.html"))
}, error = function(e) { message("Skipping ROC Plot due to error: ",
e$message) })

# --- 2.2 Weights (CSV + HTML) ---
if (!is.null(fit_binary$weights)) {
  weights_df <- data.frame(Layer = names(fit_binary$weights), Weight =
round(fit_binary$weights, 4))
  write.csv(weights_df, file.path(output_dir, "1_IBD_Weights.csv"), row.names =
FALSE)
  saveWidget(datatable(weights_df, caption = "IBD Layer Weights"),
file.path(output_dir, "1_Interactive_IBD_Weights.html"))
}

# ------------------------------------------------------------------------
# [cite_start]3. EXAMPLE 2: PREGNANCY PREDICTION (Continuous) [cite: 254-329]
# ------------------------------------------------------------------------
message("\n>>> [STEP 3] Processing Pregnancy Prediction Dataset...")

# Load Data
load(url("https://github.com/himelmallick/IntegratedLearner/blob/master/data/
pregnancy.RData?raw=true"))
ft_preg <- pcl$feature_table; sm_preg <- pcl$sample_metadata; fm_preg <-
pcl$feature_metadata; rm(pcl)

# Subset (50 features)
set.seed(123)
subsetIDs <- sample(nrow(ft_preg), 50)
ft_preg <- ft_preg[subsetIDs, ]; fm_preg <- fm_preg[subsetIDs, ]
```

```
# Train Model (FIX: Switched to Random Forest to prevent crash)
message("Training Pregnancy Model (Random Forest + NNLS)...")
fit_cont <- IntegratedLearner(
  feature_table = ft_preg, sample_metadata = sm_preg, feature_metadata =
fm_preg,
  folds = 17,
  base_learner = 'SL.randomForest', # STABLE FIX
  meta_learner = 'SL.nnls'          # Standard NNLS for continuous
)

# --- 3.1 Performance Plot (PNG + HTML) ---
tryCatch({
  IntegratedLearner:::plot.learner(fit_cont)
  p_cont_captured <- ggplot2::last_plot()

  ggsave(filename = file.path(output_dir, "2_Pregnancy_Performance.png"), plot =
p_cont_captured, width = 8, height = 6, dpi = 300)
  saveWidget(ggplotly(p_cont_captured), file.path(output_dir,
"2_Interactive_Performance.html"))
}, error = function(e) { message("Skipping Performance Plot: ", e$message) })

# ------------------------------------------------------------------------------
# 4. PREGNANCY VISUALIZATION (Updated for Random Forest)
# ------------------------------------------------------------------------------
message("\n>>> [STEP 4] Generating Visualization...")

# --- Process Predictions ---
tryCatch({
  preds <- fit_cont$SL_pred
  obs <- fit_cont$y_train

  plot_df <- data.frame(
    ID = rownames(fit_cont$sample_metadata),
    Predicted = as.numeric(preds),
    Observed = as.numeric(obs)
  )

  # Safety check for ID mismatches
  if(nrow(plot_df) != length(obs)) {
    plot_df <- data.frame(ID=1:length(obs), Predicted=as.numeric(preds),
Observed=as.numeric(obs))
  }

  # Sort for plotting
  plot_df <- plot_df[order(plot_df$Predicted, decreasing = TRUE), ]
  plot_df$ID <- factor(plot_df$ID, levels = plot_df$ID)

  # Save Predictions CSV
  write.csv(plot_df, file.path(output_dir, "2_Pregnancy_Predictions.csv"),
row.names = FALSE)

  # --- 4.1 Actual vs Predicted Scatter Plot ---
  p_scatter <- ggplot(plot_df, aes(x = Observed, y = Predicted, text =
paste("ID:", ID))) +
    geom_point(color = "darkcyan", alpha = 0.7, size = 3) +
    geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
    theme_bw() +
    labs(title = "Pregnancy: Actual vs Predicted Gestational Age", x = "Observed
Age", y = "Predicted Age")

  ggsave(filename = file.path(output_dir, "3_Pregnancy_Scatter.png"), plot =
p_scatter, width = 8, height = 6, dpi = 300)
  saveWidget(ggplotly(p_scatter, tooltip = "text"), file.path(output_dir,
"3_Interactive_Scatter.html"))
```

```
}, error = function(e) { message("Skipping Scatter Plot: ", e$message) })

# --- 4.2 Layer Weights (Feature Source Importance) ---
if (!is.null(fit_cont$weights)) {
  weights_cont_df <- data.frame(Layer = names(fit_cont$weights), Weight =
fit_cont$weights)

  # Plot
  p_weights <- ggplot(weights_cont_df, aes(x = reorder(Layer, Weight), y =
Weight, fill = Layer)) +
    geom_bar(stat = "identity") + coord_flip() + theme_bw() +
theme(legend.position = "none") +
    labs(title = "Importance of Omics Layers (Pregnancy Model)", x = "", y =
"Weight")

  ggsave(filename = file.path(output_dir, "4_Pregnancy_Layer_Weights.png"), plot
= p_weights, width = 8, height = 6, dpi = 300)
  saveWidget(ggplotly(p_weights), file.path(output_dir,
"4_Interactive_Layer_Weights.html"))

  # Table
  saveWidget(datatable(weights_cont_df, caption = "Layer Weights"),
file.path(output_dir, "4_Interactive_Layer_Weights_Table.html"))
}

message("\n============================================================")
message(">>> ANALYSIS COMPLETE! ALL FILES SAVED TO: ", output_dir)
message("============================================================")
```