Project 8 Template

```r
# Install only if needed
if (!requireNamespace("pacman", quietly = TRUE)) install.packages("pacman")
if (!requireNamespace("randomForest", quietly = TRUE)) install.packages("randomForest")

pacman::p_load(
  tidyverse,
  ggthemes,
  ltmle,
  tmle,
  SuperLearner,
  tidymodels,
  caret,
  dagitty,
  ggdag,
  here,
  randomForest,
  pROC,
  doParallel
)
```

```r
heart_disease <- read_csv(here::here("heart_disease_tmle.csv"))
```

```
## Rows: 10000 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (14): age, sex_at_birth, simplified_race, college_educ, income_thousands...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```r
head(heart_disease)
```

```
## # A tibble: 6 x 14
##     age sex_at_birth simplified_race college_educ income_thousands   bmi
##   <dbl>        <dbl>           <dbl>        <dbl>            <dbl> <dbl>
## 1  32.9            0               1            2             91.3  27.1
## 2  53.9            1               1            2             38.8  27.6
## 3  65.3            1               3            2             35.5  27.5
## 4  16.8            1               1            2             93.8  24.9
## 5  56.1            1               1            2             85.7  22.8
## 6  57.2            1               1            2             70.8  24.0
## # i 8 more variables: blood_pressure <dbl>, chol <dbl>,
## #   blood_pressure_medication <dbl>, bmi_2 <dbl>, blood_pressure_2 <dbl>,
## #   chol_2 <dbl>, blood_pressure_medication_2 <dbl>, mortality <dbl>
```

Introduction

Heart disease is the leading cause of death in the United States…

Data

This dataset was simulated using R…

- **blood__pressure__medication...** ...

SuperLearner

```r
sl_library <- c("SL.glm", "SL.randomForest", "SL.gam", "SL.mean")
```

```r
heart_disease_t1 <- heart_disease %>%
  select(-ends_with("_2")) %>%
  mutate(
    blood_pressure_medication = as.factor(blood_pressure_medication),
    mortality = as.factor(mortality),
    sex_at_birth = as.factor(sex_at_birth),
    simplified_race = as.factor(simplified_race),
    college_educ = as.factor(college_educ)
  )
```

```r
set.seed(123)
train_index <- createDataPartition(heart_disease_t1$mortality, p = 0.7, list = FALSE)
train_data <- heart_disease_t1[train_index, ]
test_data <- heart_disease_t1[-train_index, ]
```

```r
X_train <- train_data %>% select(-mortality)
Y_train <- train_data$mortality %>% as.numeric() - 1
X_test <- test_data %>% select(-mortality)
Y_test <- test_data$mortality %>% as.numeric() - 1
```

```r
# Ensure all predictors are numeric
X_train <- X_train %>% mutate(across(where(is.factor), ~ as.numeric(as.character(.))))
X_test <- X_test %>% mutate(across(where(is.factor), ~ as.numeric(as.character(.))))

# Train SuperLearner
set.seed(123)
sl_model <- tryCatch({
  SuperLearner(
    Y = Y_train,
    X = X_train,
    family = binomial(),
    SL.library = sl_library,
    method = "method.NNLS",
    verbose = TRUE
  )
}, error = function(e) {
  message("SuperLearner training failed: ", e$message)
  return(NULL)
})
```

```
## Number of covariates in All is: 9

## CV SL.glm_All

## CV SL.randomForest_All

## CV SL.gam_All

## CV SL.mean_All

## Number of covariates in All is: 9

## CV SL.glm_All

## CV SL.randomForest_All

## CV SL.gam_All

## CV SL.mean_All

## Number of covariates in All is: 9

## CV SL.glm_All

## CV SL.randomForest_All

## CV SL.gam_All

## CV SL.mean_All

## Number of covariates in All is: 9

## CV SL.glm_All

## CV SL.randomForest_All

## CV SL.gam_All

## CV SL.mean_All

## Number of covariates in All is: 9
## CV SL.glm_All

## CV SL.randomForest_All

## CV SL.gam_All

## CV SL.mean_All

## Number of covariates in All is: 9

## CV SL.glm_All

## CV SL.randomForest_All

## CV SL.gam_All

## CV SL.mean_All

## Number of covariates in All is: 9

## CV SL.glm_All

## CV SL.randomForest_All

## CV SL.gam_All

## CV SL.mean_All

## Number of covariates in All is: 9

## CV SL.glm_All

## CV SL.randomForest_All

## CV SL.gam_All

## CV SL.mean_All

## Number of covariates in All is: 9

## CV SL.glm_All

## CV SL.randomForest_All

## CV SL.gam_All

## CV SL.mean_All
## Number of covariates in All is: 9

## CV SL.glm_All

## CV SL.randomForest_All

## CV SL.gam_All

## CV SL.mean_All

## Non-Negative least squares convergence: TRUE

## full SL.glm_All

## full SL.randomForest_All

## full SL.gam_All

## full SL.mean_All
```

```
if (is.null(sl_model)) stop("SuperLearner model is NULL. Check your input data and SL.library.")
```

Risk and Coefficients

```
print(sl_model$coef)
```

```
##          SL.glm_All SL.randomForest_All          SL.gam_All         SL.mean_All
##         0.384261034         0.606169209         0.000000000         0.009569757
```

```
print(cbind(Risk = sl_model$cvRisk, Algorithm = sl_model$libraryNames))
```

```
##                     Risk                Algorithm
## SL.glm_All          "0.235008014423171" "SL.glm_All"
## SL.randomForest_All "0.231676439794315" "SL.randomForest_All"
## SL.gam_All          "0.235064575237546" "SL.gam_All"
## SL.mean_All         "0.249866215441215" "SL.mean_All"
```

Test Performance

```
if (!is.null(sl_model)) {
  sl_preds <- predict(sl_model, X_test, onlySL = TRUE)
  all_preds <- predict(sl_model, X_test, onlySL = FALSE)

  discrete_index <- which.min(sl_model$cvRisk)
  discrete_preds <- all_preds$library.predict[, discrete_index]
  discrete_name <- sl_model$libraryNames[discrete_index]
```

```r
  sl_binary_preds <- ifelse(sl_preds$pred >= 0.5, 1, 0)
  discrete_binary_preds <- ifelse(discrete_preds >= 0.5, 1, 0)

  cat("SuperLearner accuracy:", mean(sl_binary_preds == Y_test), "\n")
  cat("Discrete winner accuracy:", mean(discrete_binary_preds == Y_test), "\n")
  cat("Discrete winner algorithm:", discrete_name, "\n")
} else {
  cat("Skipping predictions due to failed SuperLearner model.\n")
}
```

```
## SuperLearner accuracy: 0.5741914
## Discrete winner accuracy: 0.5638546
## Discrete winner algorithm: SL.randomForest_All
```

```r
sl_roc <- roc(Y_test, sl_preds$pred)
```

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

```
## Warning in roc.default(Y_test, sl_preds$pred): Deprecated use a matrix as
## predictor. Unexpected results may be produced, please pass a numeric vector.
```

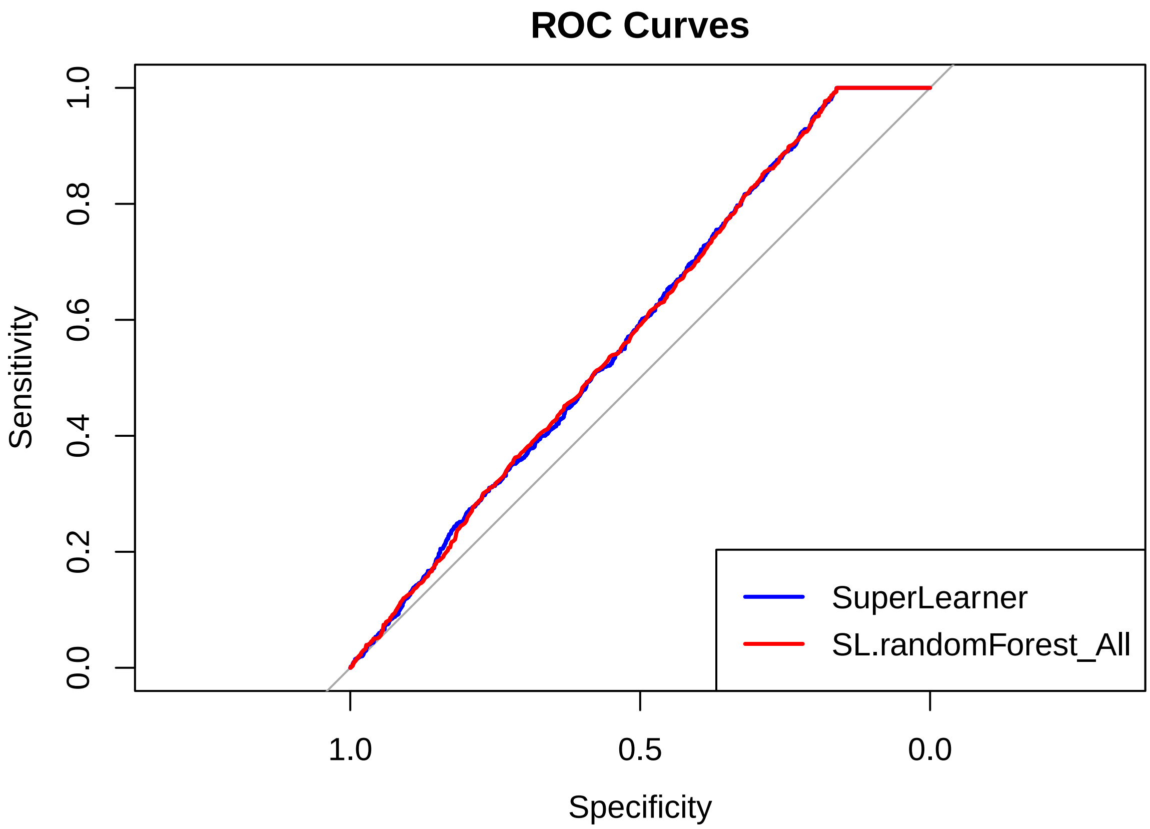
```
## Setting direction: controls < cases
```

```r
discrete_roc <- roc(Y_test, discrete_preds)
```

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

```
## Setting direction: controls < cases
```

```r
plot(sl_roc, col = "blue", main = "ROC Curves")
lines(discrete_roc, col = "red")
legend("bottomright", legend = c("SuperLearner", discrete_name), col = c("blue", "red"), lwd = 2)
```



Confusion Matrix

```
sl_conf_matrix <- table(Predicted = sl_binary_preds, Actual = Y_test)
print(sl_conf_matrix)
```

```
##          Actual
## Predicted    0    1
##         0  453  281
##         1  996 1269
```

```
precision <- sl_conf_matrix[2, 2] / sum(sl_conf_matrix[2, ])
recall <- sl_conf_matrix[2, 2] / sum(sl_conf_matrix[, 2])
f1 <- 2 * precision * recall / (precision + recall)

cat("Precision:", precision, "\n")
```

```
## Precision: 0.5602649
```

```
cat("Recall:", recall, "\n")
```

```
## Recall: 0.8187097
```

```
cat("F1 Score:", f1, "\n")
```

```
## F1 Score: 0.6652687
```

TMLE

```
heart_disease_tmle <- heart_disease_t1 %>%
  mutate(
    mortality = as.numeric(as.character(mortality)),
    blood_pressure_medication = as.numeric(as.character(blood_pressure_medication))
  )

Y <- heart_disease_tmle$mortality
A <- heart_disease_tmle$blood_pressure_medication
W <- heart_disease_tmle %>% select(-mortality, -blood_pressure_medication)

complete_cases <- complete.cases(cbind(Y, A, W))
Y <- Y[complete_cases]
A <- A[complete_cases]
W <- W[complete_cases, ]

W <- W %>% mutate(across(everything(), ~ as.numeric(as.character(.))))
W <- scale(W)

subset_idx <- sample(seq_len(nrow(W)), size = 100)
Y <- Y[subset_idx]
A <- A[subset_idx]
W <- W[subset_idx, , drop = FALSE]

cl <- makeCluster(detectCores() - 1)
registerDoParallel(cl)

tmle_result <- tmle(
  Y = Y,
  A = A,
  W = W,
  Q.SL.library = sl_library,
  g.SL.library = sl_library,
  family = "binomial",
  verbose = TRUE
)
```

```
##  Estimating initial regression of Y on A and W
##   using SuperLearner
##  Estimating treatment mechanism
##  Estimating missingness mechanism
##  Estimating treatment mechanism - ATT
```

```r
stopCluster(cl)

print(tmle_result)
```

```
##  Marginal mean under treatment (EY1)
##    Parameter Estimate:  0.17988
##    Estimated Variance:  0.0083737
##               p-value:  0.049333
##     95% Conf Interval:  (0.00052495, 0.35923)
##
##  Marginal mean under comparator (EY0)
##    Parameter Estimate:  0.56999
##    Estimated Variance:  0.0027884
##               p-value:  <2e-16
##     95% Conf Interval:  (0.46649, 0.67348)
##
##  Additive Effect
##    Parameter Estimate:  -0.39011
##    Estimated Variance:  0.011226
##               p-value:  0.00023147
##     95% Conf Interval:  (-0.59777, -0.18245)
##
##  Additive Effect among the Treated
##    Parameter Estimate:  -0.41644
##    Estimated Variance:  0.0088968
##               p-value:  1.0098e-05
##     95% Conf Interval:  (-0.60131, -0.23157)
##
##  Additive Effect among the Controls
##    Parameter Estimate:  -0.45875
##    Estimated Variance:  0.011012
##               p-value:  1.2327e-05
##     95% Conf Interval:  (-0.66442, -0.25308)
##
##  Relative Risk
##    Parameter  Estimate: 0.31558
##    Variance(log scale): 0.268
##                p-value: 0.025891
##      95% Conf Interval: (0.11441, 0.87051)
##
##  Odds Ratio
##    Parameter  Estimate: 0.16547
##    Variance(log scale): 0.43295
##                p-value: 0.0062563
##      95% Conf Interval: (0.045565, 0.6009)
```

```r
ci_low <- tmle_result$estimates$ATE$psi - 1.96 * sqrt(tmle_result$estimates$ATE$var.psi)
ci_high <- tmle_result$estimates$ATE$psi + 1.96 * sqrt(tmle_result$estimates$ATE$var.psi)

cat("ATE:", round(tmle_result$estimates$ATE$psi, 4), "\n")
```

```
## ATE: -0.3901
```

```r
cat("95% CI:", round(ci_low, 4), "-", round(ci_high, 4), "\n")
```

```
## 95% CI: -0.5978 - -0.1824
```

```r
cat("p-value:", tmle_result$estimates$ATE$pvalue, "\n")
```

```
## p-value: 0.0002314667
```