04_predictive_analysis.R

laurenkay

2025-08-05

```r
# ============================================================================
# 04: Comprehensive Predictive Analysis - Previous Year Bleaching vs Thermal Stress
# ============================================================================
# Purpose: Compare the predictive power of different variables for coral response,
#          specifically examining whether previous year bleaching extent is a better
#          predictor than current thermal stress (DHW) for coral recovery patterns.
#          Use data-driven quartile classifications and multiple model validation approaches.
# Author: Coral Bleaching Analysis Pipeline
# Date: Analysis of 2023-2025 bleaching events
# Dependencies: Requires outputs from scripts 01, 02, and 03
# ============================================================================

# Load required libraries with explicit purpose for each
library(dplyr)        # Data manipulation and summarization
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)      # Advanced plotting and visualization
library(readr)        # Efficient CSV reading
library(tidyr)        # Data reshaping for analysis
library(viridis)      # Perceptually uniform color scales
```

```
## Loading required package: viridisLite
```

```r
library(scales)       # Scale functions for plots
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:viridis':
##
##     viridis_pal

## The following object is masked from 'package:readr':
##
##     col_factor
```

```r
library(gridExtra)    # Multiple plot arrangement
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
library(corrplot)     # Correlation analysis visualization
```

```
## corrplot 0.92 loaded
```

```r
library(broom)        # Tidy model outputs
library(modelr)       # Model validation utilities
```

```
##
## Attaching package: 'modelr'

## The following object is masked from 'package:broom':
##
##     bootstrap
```

```r
library(car)          # Regression diagnostics
```

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode
```

```r
library(stringr)      # String manipulation

cat("============================================================================\n")
```

```
## ============================================================================
```

```r
cat("COMPREHENSIVE PREDICTIVE ANALYSIS: PREVIOUS BLEACHING vs THERMAL STRESS\n")
```

```
## COMPREHENSIVE PREDICTIVE ANALYSIS: PREVIOUS BLEACHING vs THERMAL STRESS
```

```r
cat("============================================================================\n\n")
```

```
## ============================================================================
```

```r
# ============================================================================
# LOAD PROCESSED DATA FROM PREVIOUS ANALYSES
# ============================================================================

cat("STEP 1: Loading processed datasets from previous analyses\n")
```

```
## STEP 1: Loading processed datasets from previous analyses
```

```r
cat("---------------------------------------------------------\n")
```

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

```r
# Load extent data for response variables
# Rationale: Response analysis provides the dependent variables for prediction
if(file.exists("03_combined_response_metrics.csv")) {
```

```r
  response_data <- read_csv("03_combined_response_metrics.csv")
  cat("Loaded response metrics from Step 03\n")
} else {
  stop("Required file 03_combined_response_metrics.csv not found. Run script 03 first.")
}
```

```
## Rows: 66 Columns: 12

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): site, period, recovery_category, response_direction, recovery_effic...
## dbl (7): initial_bleaching, final_bleaching, response_magnitude, recovery_ac...
##
## 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.

## Loaded response metrics from Step 03
```

```r
# Load thermal stress data for predictor variables
# Rationale: Thermal stress metrics provide key environmental predictor variables
if(file.exists("02_comprehensive_thermal_data.csv")) {
  thermal_data <- read_csv("02_comprehensive_thermal_data.csv")
  cat("Loaded thermal stress data from Step 02\n")
} else {
  cat("Warning: Thermal data not found. Analysis will proceed with limited predictors.\n")
  thermal_data <- NULL
}
```

```
## Rows: 66 Columns: 28
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): site, dhw_stress_category, temp_var_category
## dbl (25): year, max_dhw, mean_dhw, total_dhw_accumulation, max_weekly_temp, ...
##
## 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.

## Loaded thermal stress data from Step 02
```

```r
# Load site response patterns for cross-validation
if(file.exists("03_site_response_patterns.csv")) {
  site_patterns <- read_csv("03_site_response_patterns.csv")
  cat("Loaded site response patterns from Step 03\n")
} else {
  site_patterns <- NULL
}
```

```
## Rows: 33 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): site, period_p1, period_p2, pattern_classification, recovery_traje...
## dbl (16): initial_bleaching_p1, final_bleaching_p1, response_magnitude_p1, r...
##
## 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.

## Loaded site response patterns from Step 03
```

```r
# Load response category thresholds for consistent classification
if(file.exists("03_response_category_thresholds.csv")) {
  response_thresholds <- read_csv("03_response_category_thresholds.csv")
  cat("Loaded response category thresholds from Step 03\n")
} else {
  response_thresholds <- NULL
}
```

```
## Rows: 7 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): metric, percentile, description
## dbl (1): threshold
##
## 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.
```

```
## Loaded response category thresholds from Step 03
```

```r
cat(sprintf("Primary response dataset dimensions: %d rows × %d columns\n", nrow(response_data), ncol(re
```

```
## Primary response dataset dimensions: 66 rows × 12 columns
```

```r
# ============================================================================
# PREDICTIVE DATASET CONSTRUCTION
# ============================================================================

cat("\nSTEP 2: Constructing comprehensive predictive dataset\n")
```

```
##
## STEP 2: Constructing comprehensive predictive dataset
```

```r
cat("----------------------------------------------------\n")
```

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

```r
# Create the main predictive dataset focusing on Period 2 (2024 Annual → 2025 PBL)
# Rationale: Period 2 is our primary response of interest, as it represents the
# most recent coral responses and allows us to use Period 1 and thermal data as predictors

# Extract Period 2 responses (our dependent variables)
period2_responses <- response_data %>%
  filter(period == "2024_to_2025_PBL") %>%
  select(site, response_magnitude, recovery_achieved, recovery_proportion,
         initial_bleaching, final_bleaching, worsening_achieved) %>%
  rename(
    response_2024_to_2025 = response_magnitude,
    recovery_2024_to_2025 = recovery_achieved,
    recovery_prop_2024_to_2025 = recovery_proportion,
    baseline_2024_annual = initial_bleaching,
    outcome_2025_pbl = final_bleaching,
    worsening_2024_to_2025 = worsening_achieved
  )

cat(sprintf("Period 2 response data: %d sites\n", nrow(period2_responses)))
```

```
## Period 2 response data: 33 sites
```

```r
# Extract Period 1 responses (potential predictors)
period1_responses <- response_data %>%
  filter(period == "2023_to_2024_PBL") %>%
  select(site, response_magnitude, recovery_achieved, recovery_proportion,
         initial_bleaching, final_bleaching) %>%
  rename(
    response_2023_to_2024 = response_magnitude,
    recovery_2023_to_2024 = recovery_achieved,
    recovery_prop_2023_to_2024 = recovery_proportion,
    baseline_2023_annual = initial_bleaching,
    outcome_2024_pbl = final_bleaching
  )

cat(sprintf("Period 1 predictor data: %d sites\n", nrow(period1_responses)))
```

```
## Period 1 predictor data: 33 sites
```

```r
# Merge response data
predictive_base <- period2_responses %>%
  left_join(period1_responses, by = "site") %>%
  filter(!is.na(baseline_2023_annual))  # Ensure complete temporal coverage

cat(sprintf("Merged response dataset: %d sites\n", nrow(predictive_base)))
```

```
## Merged response dataset: 33 sites
```

```r
# Add thermal stress predictors if available
if(!is.null(thermal_data)) {
  # Prepare thermal data for merging
  thermal_predictors <- thermal_data %>%
    select(site, year, max_dhw, temp_sd, total_dhw_accumulation,
           max_weekly_temp, temp_range, weeks_with_dhw) %>%
    pivot_wider(names_from = year,
                values_from = c(max_dhw, temp_sd, total_dhw_accumulation,
                                max_weekly_temp, temp_range, weeks_with_dhw),
                names_sep = "_") %>%
    mutate(
      # Calculate additional thermal stress indices
      cumulative_dhw = total_dhw_accumulation_2023 + total_dhw_accumulation_2024,
      max_annual_dhw = pmax(max_dhw_2023, max_dhw_2024, na.rm = TRUE),
      thermal_variability = temp_sd_2023 + temp_sd_2024,
      thermal_persistence = weeks_with_dhw_2023 + weeks_with_dhw_2024,
      dhw_escalation = max_dhw_2024 - max_dhw_2023
    )

  # Merge thermal predictors
  predictive_dataset <- predictive_base %>%
    left_join(thermal_predictors, by = "site") %>%
    filter(!is.na(max_dhw_2023) & !is.na(max_dhw_2024))

  cat(sprintf("Final predictive dataset with thermal data: %d sites\n", nrow(predictive_dataset)))
} else {
  predictive_dataset <- predictive_base
  cat(sprintf("Final predictive dataset without thermal data: %d sites\n", nrow(predictive_dataset)))
}
```

```
## Final predictive dataset with thermal data: 32 sites
```

```r
# =============================================================================
# DATA-DRIVEN PREDICTOR CATEGORIZATION
# =============================================================================

cat("\nSTEP 3: Establishing data-driven predictor categories using observed quartiles\n")
```

```
##
## STEP 3: Establishing data-driven predictor categories using observed quartiles
```

```r
cat("------------------------------------------------------------------------------\n")
```

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

```r
# Calculate quartiles for key predictor variables
# Rationale: Data-driven categories ensure balanced groups and meaningful
# ecological interpretation based on observed distributions

# Previous year bleaching quartiles
prev_bleaching_quartiles <- quantile(predictive_dataset$baseline_2023_annual,
                                     probs = c(0, 0.25, 0.5, 0.75, 1.0), na.rm = TRUE)
cat("2023 Annual bleaching extent quartiles:\n")
```

```
## 2023 Annual bleaching extent quartiles:
```

```r
print(prev_bleaching_quartiles)
```

```
##       0%      25%      50%      75%     100%
##  0.00000 15.00000 31.38889 38.60417 80.46469
```

```r
if(!is.null(thermal_data)) {
  # DHW quartiles for 2024 (concurrent stress)
  dhw_2024_quartiles <- quantile(predictive_dataset$max_dhw_2024,
                                 probs = c(0, 0.25, 0.5, 0.75, 1.0), na.rm = TRUE)
  cat("\n2024 Maximum DHW quartiles:\n")
  print(dhw_2024_quartiles)

  # DHW quartiles for 2023 (preceding stress)
  dhw_2023_quartiles <- quantile(predictive_dataset$max_dhw_2023,
                                 probs = c(0, 0.25, 0.5, 0.75, 1.0), na.rm = TRUE)
  cat("\n2023 Maximum DHW quartiles:\n")
  print(dhw_2023_quartiles)

  # Temperature variability quartiles
  temp_var_quartiles <- quantile(predictive_dataset$thermal_variability,
                                 probs = c(0, 0.25, 0.5, 0.75, 1.0), na.rm = TRUE)
  cat("\nThermal variability quartiles:\n")
  print(temp_var_quartiles)
}
```

```
##
## 2024 Maximum DHW quartiles:
##        0%       25%       50%       75%      100%
##  3.301951 20.503948 21.587552 23.040335 26.410173
##
## 2023 Maximum DHW quartiles:
##        0%       25%       50%       75%      100%
```

```
##  2.101927 17.049057 18.643864 19.255842 22.613353
##
## Thermal variability quartiles:
##       0%      25%      50%      75%     100%
## 1.755946 2.595363 2.662090 2.707967 2.988871
```

```r
# Apply data-driven categorizations to predictive dataset
predictive_dataset <- predictive_dataset %>%
  mutate(
    # Previous bleaching categories (key predictor of interest)
    prev_bleaching_category = case_when(
      baseline_2023_annual <= prev_bleaching_quartiles[2] ~ "Low Previous Impact",
      baseline_2023_annual <= prev_bleaching_quartiles[3] ~ "Moderate Previous Impact",
      baseline_2023_annual <= prev_bleaching_quartiles[4] ~ "High Previous Impact",
      baseline_2023_annual > prev_bleaching_quartiles[4] ~ "Severe Previous Impact",
      TRUE ~ "Unknown"
    ),
    prev_bleaching_category = factor(prev_bleaching_category,
                                     levels = c("Low Previous Impact", "Moderate Previous Impact",
                                                "High Previous Impact", "Severe Previous Impact"))
  )

if(!is.null(thermal_data)) {
  predictive_dataset <- predictive_dataset %>%
    mutate(
      # Current DHW categories (competing predictor)
      dhw_2024_category = case_when(
        max_dhw_2024 <= dhw_2024_quartiles[2] ~ "Low Current Stress",
        max_dhw_2024 <= dhw_2024_quartiles[3] ~ "Moderate Current Stress",
        max_dhw_2024 <= dhw_2024_quartiles[4] ~ "High Current Stress",
        max_dhw_2024 > dhw_2024_quartiles[4] ~ "Extreme Current Stress",
        TRUE ~ "Unknown"
      ),
      dhw_2024_category = factor(dhw_2024_category,
                                 levels = c("Low Current Stress", "Moderate Current Stress",
                                            "High Current Stress", "Extreme Current Stress")),

      # Combined thermal stress categories
      thermal_variability_category = case_when(
        thermal_variability <= temp_var_quartiles[2] ~ "Low Variability",
        thermal_variability <= temp_var_quartiles[3] ~ "Moderate Variability",
        thermal_variability <= temp_var_quartiles[4] ~ "High Variability",
        thermal_variability > temp_var_quartiles[4] ~ "Extreme Variability",
        TRUE ~ "Unknown"
      ),
      thermal_variability_category = factor(thermal_variability_category,
                                            levels = c("Low Variability", "Moderate Variability",
                                                       "High Variability", "Extreme Variability"))
    )
}

# Print category thresholds for documentation
cat("\nData-driven predictor category thresholds:\n")
```

```
##
```

```
## Data-driven predictor category thresholds:
```

```r
cat(sprintf("Previous bleaching impact categories (2023 Annual %%):\n"))
```

```
## Previous bleaching impact categories (2023 Annual %):
```

```r
cat(sprintf("  Low: 0 - %.1f%% (0-25th percentile)\n", prev_bleaching_quartiles[2]))
```

```
##   Low: 0 - 15.0% (0-25th percentile)
```

```r
cat(sprintf("  Moderate: %.1f - %.1f%% (25th-50th percentile)\n", prev_bleaching_quartiles[2], prev_blea
```

```
##   Moderate: 15.0 - 31.4% (25th-50th percentile)
```

```r
cat(sprintf("  High: %.1f - %.1f%% (50th-75th percentile)\n", prev_bleaching_quartiles[3], prev_bleachi
```

```
##   High: 31.4 - 38.6% (50th-75th percentile)
```

```r
cat(sprintf("  Severe: >%.1f%% (75th-100th percentile)\n", prev_bleaching_quartiles[4]))
```

```
##   Severe: >38.6% (75th-100th percentile)
```

```r
# ============================================================================
# CORRELATION ANALYSIS
# ============================================================================

cat("\nSTEP 4: Comprehensive correlation analysis between predictors and responses\n")
```

```
##
## STEP 4: Comprehensive correlation analysis between predictors and responses
```

```r
cat("----------------------------------------------------------------------------\n")
```

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

```r
# Define predictor variables for correlation analysis
# Rationale: Systematic correlation analysis reveals which variables have
# the strongest linear relationships with coral responses

predictor_vars <- list(
  "2023_Bleaching" = "baseline_2023_annual",
  "2023_Recovery" = "recovery_2023_to_2024",
  "2024_Baseline" = "baseline_2024_annual"
)

if(!is.null(thermal_data)) {
  thermal_predictors <- list(
    "2024_DHW" = "max_dhw_2024",
    "2023_DHW" = "max_dhw_2023",
    "Cumulative_DHW" = "cumulative_dhw",
    "Max_Annual_DHW" = "max_annual_dhw",
    "Thermal_Variability" = "thermal_variability",
    "Thermal_Persistence" = "thermal_persistence",
    "DHW_Escalation" = "dhw_escalation"
  )
  predictor_vars <- c(predictor_vars, thermal_predictors)
}

# Define response variables
response_vars <- list(
```

```r
  "2025_Response" = "response_2024_to_2025",
  "Recovery_Achieved" = "recovery_2024_to_2025",
  "Recovery_Proportion" = "recovery_prop_2024_to_2025"
)

# Calculate correlation matrix
correlation_results <- data.frame()

for(pred_name in names(predictor_vars)) {
  for(resp_name in names(response_vars)) {
    pred_var <- predictor_vars[[pred_name]]
    resp_var <- response_vars[[resp_name]]

    if(pred_var %in% names(predictive_dataset) && resp_var %in% names(predictive_dataset)) {
      corr_value <- cor(predictive_dataset[[pred_var]], predictive_dataset[[resp_var]],
                        use = "complete.obs")

      # Calculate p-value for correlation
      corr_test <- cor.test(predictive_dataset[[pred_var]], predictive_dataset[[resp_var]])

      correlation_results <- rbind(correlation_results,
                                   data.frame(
                                     Predictor = pred_name,
                                     Response = resp_name,
                                     Correlation = corr_value,
                                     P_Value = corr_test$p.value,
                                     N_Obs = sum(!is.na(predictive_dataset[[pred_var]]) &
                                                 !is.na(predictive_dataset[[resp_var]])),
                                     Abs_Correlation = abs(corr_value)
                                   ))
    }
  }
}

# Sort by absolute correlation strength
correlation_results <- correlation_results %>%
  arrange(desc(Abs_Correlation))

cat("Top predictors by correlation strength:\n")
```

```
## Top predictors by correlation strength:
```

```r
print(head(correlation_results, 15))
```

```
##              Predictor            Response Correlation      P_Value N_Obs
## 1        2024_Baseline   Recovery_Achieved   0.9086745 6.582434e-13    32
## 2        2024_Baseline       2025_Response  -0.9011324 2.055323e-12    32
## 3        2024_Baseline Recovery_Proportion   0.5544268 9.921673e-04    32
## 4   Thermal_Variability       2025_Response  -0.3748435 3.452953e-02    32
## 5   Thermal_Variability   Recovery_Achieved   0.3701783 3.702702e-02    32
## 6        2023_Recovery   Recovery_Achieved   0.2639608 1.443409e-01    32
## 7        2023_Recovery       2025_Response  -0.2625150 1.466336e-01    32
## 8       2023_Bleaching   Recovery_Achieved   0.2325174 2.003261e-01    32
## 9       2023_Bleaching       2025_Response  -0.2237294 2.183500e-01    32
```

```
## 10           2023_DHW Recovery_Proportion  -0.1926201 2.908754e-01    32
## 11 Thermal_Persistence Recovery_Proportion  -0.1914498 2.938726e-01    32
## 12      DHW_Escalation   Recovery_Achieved  -0.1767600 3.331580e-01    32
## 13      DHW_Escalation       2025_Response   0.1613747 3.775837e-01    32
## 14            2024_DHW   Recovery_Achieved  -0.1447803 4.291802e-01    32
## 15      DHW_Escalation Recovery_Proportion   0.1299415 4.784305e-01    32
##    Abs_Correlation
## 1        0.9086745
## 2        0.9011324
## 3        0.5544268
## 4        0.3748435
## 5        0.3701783
## 6        0.2639608
## 7        0.2625150
## 8        0.2325174
## 9        0.2237294
## 10       0.1926201
## 11       0.1914498
## 12       0.1767600
## 13       0.1613747
## 14       0.1447803
## 15       0.1299415
```

```r
# ============================================================================
# VISUALIZATION 1: Correlation Matrix Heatmap
# ============================================================================

# Create comprehensive correlation matrix visualization
# Justification: Visual correlation matrix reveals patterns and relationships
# that may not be apparent in tabular form

# Prepare data for correlation matrix
correlation_vars <- predictive_dataset %>%
  select(all_of(unlist(predictor_vars)), all_of(unlist(response_vars))) %>%
  select_if(~ !all(is.na(.)))  # Remove columns with all NA values

# Calculate correlation matrix
cor_matrix <- cor(correlation_vars, use = "complete.obs")

# Create heatmap
p1 <- ggplot(expand.grid(Var1 = rownames(cor_matrix), Var2 = colnames(cor_matrix)),
             aes(x = Var1, y = Var2, fill = as.vector(cor_matrix))) +
  geom_tile(color = "white") +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1, 1), space = "Lab",
                       name = "Correlation\nCoefficient") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 10),
        axis.text.y = element_text(size = 10)) +
  coord_fixed() +
  labs(
    title = "Correlation Matrix: Predictors vs Coral Response Variables",
    subtitle = "Red = positive correlation, Blue = negative correlation",
    x = "Variables",
```

```r
    y = "Variables",
    caption = "This correlation matrix reveals the strength of linear relationships\nbetween predictor v
  ) +
  theme(plot.caption = element_text(hjust = 0, size = 9, color = "gray30"))

ggsave("04_plot_correlation_matrix.png", p1, width = 12, height = 10, dpi = 300)
cat("\nSaved: 04_plot_correlation_matrix.png\n")
```

```
##
## Saved: 04_plot_correlation_matrix.png
```

```r
# ============================================================================
# LINEAR REGRESSION MODEL COMPARISON
# ============================================================================

cat("\nSTEP 5: Comprehensive linear regression model comparison\n")
```

```
##
## STEP 5: Comprehensive linear regression model comparison
```

```r
cat("-----------------------------------------------------------\n")
```

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

```r
# Build and compare multiple predictive models
# Rationale: Systematic model comparison reveals which variables provide
# the best predictive power for coral responses

# Define response variable for main analysis
response_var <- "recovery_2024_to_2025"
cat(sprintf("Primary response variable: %s\n", response_var))
```

```
## Primary response variable: recovery_2024_to_2025
```

```r
# Model 1: Previous year bleaching only
model_prev_bleaching <- lm(recovery_2024_to_2025 ~ baseline_2023_annual,
                           data = predictive_dataset)

# Model 2: Current year baseline only
model_current_baseline <- lm(recovery_2024_to_2025 ~ baseline_2024_annual,
                             data = predictive_dataset)

# Model 3: Previous recovery performance
model_prev_recovery <- lm(recovery_2024_to_2025 ~ recovery_2023_to_2024,
                          data = predictive_dataset)

model_list <- list(
  "Previous_Bleaching_Only" = model_prev_bleaching,
  "Current_Baseline_Only" = model_current_baseline,
  "Previous_Recovery_Only" = model_prev_recovery
)

if(!is.null(thermal_data)) {
  # Model 4: Current year DHW only
  model_current_dhw <- lm(recovery_2024_to_2025 ~ max_dhw_2024,
                          data = predictive_dataset)
```

```r
  # Model 5: Previous year DHW only
  model_prev_dhw <- lm(recovery_2024_to_2025 ~ max_dhw_2023,
                       data = predictive_dataset)

  # Model 6: Combined previous bleaching + current DHW
  model_combined <- lm(recovery_2024_to_2025 ~ baseline_2023_annual + max_dhw_2024,
                       data = predictive_dataset)

  # Model 7: Comprehensive model
  model_comprehensive <- lm(recovery_2024_to_2025 ~ baseline_2023_annual + max_dhw_2024 +
                            thermal_variability + recovery_2023_to_2024,
                            data = predictive_dataset)

  # Model 8: Thermal-only comprehensive
  model_thermal_comprehensive <- lm(recovery_2024_to_2025 ~ max_dhw_2024 + max_dhw_2023 +
                                    thermal_variability + cumulative_dhw,
                                    data = predictive_dataset)

  thermal_models <- list(
    "Current_DHW_Only" = model_current_dhw,
    "Previous_DHW_Only" = model_prev_dhw,
    "Combined_Bleaching_DHW" = model_combined,
    "Comprehensive_Model" = model_comprehensive,
    "Thermal_Comprehensive" = model_thermal_comprehensive
  )

  model_list <- c(model_list, thermal_models)
}

# Extract model performance metrics
model_comparison <- data.frame()

for(model_name in names(model_list)) {
  model <- model_list[[model_name]]
  model_summary <- summary(model)

  # Calculate additional metrics
  predictions <- predict(model)
  residuals <- residuals(model)

  model_comparison <- rbind(model_comparison,
                            data.frame(
                              Model = model_name,
                              R_Squared = model_summary$r.squared,
                              Adj_R_Squared = model_summary$adj.r.squared,
                              RMSE = sqrt(mean(residuals^2, na.rm = TRUE)),
                              MAE = mean(abs(residuals), na.rm = TRUE),
                              AIC = AIC(model),
                              BIC = BIC(model),
                              N_Predictors = length(model$coefficients) - 1,
                              N_Observations = nobs(model)
                            ))
}
```

```r
# Sort by adjusted R-squared
model_comparison <- model_comparison %>%
  arrange(desc(Adj_R_Squared)) %>%
  mutate(
    R_Squared = round(R_Squared, 4),
    Adj_R_Squared = round(Adj_R_Squared, 4),
    RMSE = round(RMSE, 2),
    MAE = round(MAE, 2),
    AIC = round(AIC, 1),
    BIC = round(BIC, 1)
  )

cat("Model performance comparison (ranked by Adjusted R²):\n")
```

```
## Model performance comparison (ranked by Adjusted R²):
```

```r
print(model_comparison)
```

```
##                       Model R_Squared Adj_R_Squared  RMSE   MAE   AIC   BIC
## 1   Current_Baseline_Only    0.8257        0.8199  9.21  6.82 238.9 243.3
## 2   Thermal_Comprehensive    0.2283        0.1140 19.37 15.06 292.5 301.3
## 3     Comprehensive_Model    0.1839        0.0630 19.92 16.34 294.3 303.1
## 4  Previous_Recovery_Only    0.0697        0.0387 21.27 17.26 292.5 296.9
## 5 Previous_Bleaching_Only    0.0541        0.0225 21.45 17.37 293.0 297.4
## 6  Combined_Bleaching_DHW    0.0697        0.0055 21.27 16.97 294.5 300.3
## 7         Current_DHW_Only    0.0210       -0.0117 21.82 18.50 294.1 298.5
## 8        Previous_DHW_Only    0.0089       -0.0242 21.95 18.75 294.5 298.9
##   N_Predictors N_Observations
## 1            1             32
## 2            4             32
## 3            4             32
## 4            1             32
## 5            1             32
## 6            2             32
## 7            1             32
## 8            1             32
```

```r
# ============================================================================
# VISUALIZATION 2: Model Performance Comparison
# ============================================================================

# Create comprehensive model performance visualization
# Justification: Visual comparison makes model performance differences clear
# and helps identify the best predictive approaches

model_performance_long <- model_comparison %>%
  select(Model, R_Squared, Adj_R_Squared, RMSE) %>%
  pivot_longer(cols = c(R_Squared, Adj_R_Squared, RMSE),
               names_to = "Metric", values_to = "Value") %>%
  mutate(
    Model = str_replace_all(Model, "_", " "),
    Metric = case_when(
      Metric == "R_Squared" ~ "R²",
      Metric == "Adj_R_Squared" ~ "Adjusted R²",
```

```r
      Metric == "RMSE" ~ "RMSE"
    )
  )

p2 <- ggplot(model_performance_long %>% filter(Metric != "RMSE"),
             aes(x = reorder(Model, Value), y = Value, fill = Metric)) +
  geom_col(position = "dodge", alpha = 0.8) +
  geom_text(aes(label = sprintf("%.3f", Value)),
            position = position_dodge(width = 0.9),
            hjust = -0.1, size = 3) +
  scale_fill_viridis_d(name = "Metric") +
  coord_flip() +
  labs(
    title = "Predictive Model Performance Comparison",
    subtitle = "Models ranked by explanatory power (R² metrics)",
    x = "Model",
    y = "Performance Value",
    caption = "This plot compares the explanatory power of different predictive models.\nHigher R² valu
  ) +
  theme_minimal() +
  theme(plot.caption = element_text(hjust = 0, size = 9, color = "gray30"))

ggsave("04_plot_model_performance.png", p2, width = 12, height = 8, dpi = 300)
cat("Saved: 04_plot_model_performance.png\n")
```

```
## Saved: 04_plot_model_performance.png
```

```r
# ============================================================================
# DETAILED PREDICTOR ANALYSIS
# ============================================================================

cat("\nSTEP 6: Detailed analysis of key predictor relationships\n")
```

```
##
## STEP 6: Detailed analysis of key predictor relationships
```

```r
cat("--------------------------------------------------------\n")
```

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

```r
# Extract key insights about predictor performance
# Rationale: Detailed analysis provides context for model performance differences

best_model_name <- model_comparison$Model[1]
best_model <- model_list[[best_model_name]]

cat(sprintf("Best performing model: %s\n", best_model_name))
```

```
## Best performing model: Current_Baseline_Only
```

```r
cat(sprintf("Adjusted R²: %.4f\n", model_comparison$Adj_R_Squared[1]))
```

```
## Adjusted R²: 0.8199
```

```r
cat(sprintf("RMSE: %.2f\n", model_comparison$RMSE[1]))
```

```
## RMSE: 9.21
```

```r
# Compare key competing predictors
prev_bleaching_performance <- model_comparison %>%
  filter(Model == "Previous_Bleaching_Only") %>%
  select(Adj_R_Squared, RMSE)

if(!is.null(thermal_data)) {
  current_dhw_performance <- model_comparison %>%
    filter(Model == "Current_DHW_Only") %>%
    select(Adj_R_Squared, RMSE)

  cat("\nKey predictor comparison:\n")
  cat(sprintf("Previous bleaching (2023): Adj R² = %.4f, RMSE = %.2f\n",
              prev_bleaching_performance$Adj_R_Squared, prev_bleaching_performance$RMSE))
  cat(sprintf("Current DHW (2024): Adj R² = %.4f, RMSE = %.2f\n",
              current_dhw_performance$Adj_R_Squared, current_dhw_performance$RMSE))

  # Determine which is stronger
  if(prev_bleaching_performance$Adj_R_Squared > current_dhw_performance$Adj_R_Squared) {
    stronger_predictor <- "Previous bleaching"
    advantage <- prev_bleaching_performance$Adj_R_Squared - current_dhw_performance$Adj_R_Squared
  } else {
    stronger_predictor <- "Current DHW"
    advantage <- current_dhw_performance$Adj_R_Squared - prev_bleaching_performance$Adj_R_Squared
  }

  cat(sprintf("\n%s is the stronger predictor by %.4f Adj R² units\n", stronger_predictor, advantage))
}
```

```
##
## Key predictor comparison:
## Previous bleaching (2023): Adj R² = 0.0225, RMSE = 21.45
## Current DHW (2024): Adj R² = -0.0117, RMSE = 21.82
##
## Previous bleaching is the stronger predictor by 0.0342 Adj R² units
```

```r
# ============================================================================
# VISUALIZATION 3: Key Predictor Relationships
# ============================================================================

# Create detailed visualization of key predictor relationships
# Justification: Scatter plots reveal the nature of relationships and identify outliers

if(!is.null(thermal_data)) {
  p3 <- ggplot(predictive_dataset, aes(x = baseline_2023_annual, y = recovery_2024_to_2025)) +
    geom_point(aes(color = max_dhw_2024, size = thermal_variability), alpha = 0.7) +
    geom_smooth(method = "lm", se = TRUE, color = "blue", linetype = "solid") +
    scale_color_viridis_c(name = "2024 DHW") +
    scale_size_continuous(name = "Thermal\nVariability", range = c(2, 8)) +
    labs(
      title = "Previous Bleaching vs Recovery Response",
      subtitle = "Point color = 2024 DHW, Point size = thermal variability",
      x = "2023 Annual Bleaching Extent (%)",
      y = "2024→2025 Recovery Achieved (%)",
      caption = "This plot examines the relationship between previous bleaching and recovery.\nThe blue
```

```r
    ) +
    theme_minimal() +
    theme(plot.caption = element_text(hjust = 0, size = 9, color = "gray30"))

  ggsave("04_plot_previous_bleaching_relationship.png", p3, width = 12, height = 8, dpi = 300)
  cat("Saved: 04_plot_previous_bleaching_relationship.png\n")

  # Second key relationship plot
  p4 <- ggplot(predictive_dataset, aes(x = max_dhw_2024, y = recovery_2024_to_2025)) +
    geom_point(aes(color = baseline_2023_annual, size = thermal_variability), alpha = 0.7) +
    geom_smooth(method = "lm", se = TRUE, color = "red", linetype = "solid") +
    scale_color_viridis_c(name = "2023 Bleaching\nExtent (%)") +
    scale_size_continuous(name = "Thermal\nVariability", range = c(2, 8)) +
    labs(
      title = "Current DHW vs Recovery Response",
      subtitle = "Point color = 2023 bleaching extent, Point size = thermal variability",
      x = "2024 Maximum DHW",
      y = "2024→2025 Recovery Achieved (%)",
      caption = "This plot examines the relationship between current thermal stress and recovery.\nThe i
    ) +
    theme_minimal() +
    theme(plot.caption = element_text(hjust = 0, size = 9, color = "gray30"))

  ggsave("04_plot_current_dhw_relationship.png", p4, width = 12, height = 8, dpi = 300)
  cat("Saved: 04_plot_current_dhw_relationship.png\n")

} else {
  p3 <- ggplot(predictive_dataset, aes(x = baseline_2023_annual, y = recovery_2024_to_2025)) +
    geom_point(aes(color = recovery_2023_to_2024), size = 3, alpha = 0.7) +
    geom_smooth(method = "lm", se = TRUE, color = "blue", linetype = "solid") +
    scale_color_viridis_c(name = "2023→2024\nRecovery (%)") +
    labs(
      title = "Previous Bleaching vs Current Recovery Response",
      subtitle = "Point color shows previous period recovery performance",
      x = "2023 Annual Bleaching Extent (%)",
      y = "2024→2025 Recovery Achieved (%)",
      caption = "This plot examines the relationship between previous bleaching and current recovery.\n'
    ) +
    theme_minimal() +
    theme(plot.caption = element_text(hjust = 0, size = 9, color = "gray30"))

  ggsave("04_plot_previous_bleaching_relationship.png", p3, width = 12, height = 8, dpi = 300)
  cat("Saved: 04_plot_previous_bleaching_relationship.png\n")
}
```

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

## Saved: 04_plot_previous_bleaching_relationship.png

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

## Saved: 04_plot_current_dhw_relationship.png
```

```r
# ============================================================================
# PREDICTOR CATEGORY ANALYSIS
# ============================================================================
```

```r
cat("\nSTEP 7: Analyzing responses by predictor categories\n")
```

```
##
## STEP 7: Analyzing responses by predictor categories
```

```r
cat("--------------------------------------------------\n")
```

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

```r
# Analyze responses within data-driven predictor categories
# Rationale: Category-based analysis reveals non-linear relationships
# and validates the utility of quartile-based classifications

# Previous bleaching category analysis
prev_category_analysis <- predictive_dataset %>%
  group_by(prev_bleaching_category) %>%
  summarise(
    n_sites = n(),
    mean_recovery = mean(recovery_2024_to_2025, na.rm = TRUE),
    median_recovery = median(recovery_2024_to_2025, na.rm = TRUE),
    sd_recovery = sd(recovery_2024_to_2025, na.rm = TRUE),
    q25_recovery = quantile(recovery_2024_to_2025, 0.25, na.rm = TRUE),
    q75_recovery = quantile(recovery_2024_to_2025, 0.75, na.rm = TRUE),
    prop_positive_recovery = mean(recovery_2024_to_2025 > 0, na.rm = TRUE),
    prop_strong_recovery = mean(recovery_2024_to_2025 > 15, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  arrange(prev_bleaching_category)

cat("Recovery by previous bleaching impact category:\n")
```

```
## Recovery by previous bleaching impact category:
```

```r
print(prev_category_analysis)
```

```
## # A tibble: 4 x 9
##   prev_bleaching_category  n_sites mean_recovery median_recovery sd_recovery
##   <fct>                      <int>         <dbl>           <dbl>       <dbl>
## 1 Low Previous Impact            9          17.4             9          26.4
## 2 Moderate Previous Impact       7          20.2            17.2        19.7
## 3 High Previous Impact           8          24.7            16.8        27.6
## 4 Severe Previous Impact         8          28.1            30          15.9
## # i 4 more variables: q25_recovery <dbl>, q75_recovery <dbl>,
## #   prop_positive_recovery <dbl>, prop_strong_recovery <dbl>
```

```r
if(!is.null(thermal_data)) {
  # Current DHW category analysis
  dhw_category_analysis <- predictive_dataset %>%
    group_by(dhw_2024_category) %>%
    summarise(
      n_sites = n(),
      mean_recovery = mean(recovery_2024_to_2025, na.rm = TRUE),
      median_recovery = median(recovery_2024_to_2025, na.rm = TRUE),
      sd_recovery = sd(recovery_2024_to_2025, na.rm = TRUE),
      q25_recovery = quantile(recovery_2024_to_2025, 0.25, na.rm = TRUE),
      q75_recovery = quantile(recovery_2024_to_2025, 0.75, na.rm = TRUE),
```

```r
      prop_positive_recovery = mean(recovery_2024_to_2025 > 0, na.rm = TRUE),
      prop_strong_recovery = mean(recovery_2024_to_2025 > 15, na.rm = TRUE),
      .groups = "drop"
    ) %>%
    arrange(dhw_2024_category)

  cat("\nRecovery by current DHW stress category:\n")
  print(dhw_category_analysis)
}
```

```
##
## Recovery by current DHW stress category:
## # A tibble: 4 x 9
##   dhw_2024_category       n_sites mean_recovery median_recovery sd_recovery
##   <fct>                     <int>         <dbl>           <dbl>       <dbl>
## 1 Low Current Stress            8          10.1           0.111        16.8
## 2 Moderate Current Stress       8          34.0          37.4          22.3
## 3 High Current Stress           8          27.9          33.0          18.4
## 4 Extreme Current Stress        8          18.1          12.4          26.9
## # i 4 more variables: q25_recovery <dbl>, q75_recovery <dbl>,
## #   prop_positive_recovery <dbl>, prop_strong_recovery <dbl>
```

```r
# ============================================================================
# VISUALIZATION 4: Recovery by Predictor Categories
# ============================================================================

# Create comprehensive category-based analysis visualization
# Justification: Box plots reveal distributions within categories and
# validate the effectiveness of quartile-based classifications

p5 <- ggplot(predictive_dataset, aes(x = prev_bleaching_category, y = recovery_2024_to_2025,
                                     fill = prev_bleaching_category)) +
  geom_boxplot(alpha = 0.7, outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.6, size = 2) +
  stat_summary(fun = mean, geom = "point", color = "red", size = 3, shape = 18) +
  scale_fill_viridis_d(name = "Previous\nBleaching\nCategory") +
  labs(
    title = "Recovery Response by Previous Bleaching Impact Category",
    subtitle = "Red diamonds show category means; data-driven quartile categories",
    x = "Previous Bleaching Impact Category (2023 Annual)",
    y = "Recovery Achieved 2024→2025 (%)",
    caption = "This plot examines recovery patterns within data-driven categories of\nprevious bleaching
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "none",
    plot.caption = element_text(hjust = 0, size = 9, color = "gray30")
  )

ggsave("04_plot_recovery_by_previous_category.png", p5, width = 12, height = 8, dpi = 300)
cat("Saved: 04_plot_recovery_by_previous_category.png\n")
```

```
## Saved: 04_plot_recovery_by_previous_category.png
```

```
if(!is.null(thermal_data)) {
  p6 <- ggplot(predictive_dataset, aes(x = dhw_2024_category, y = recovery_2024_to_2025,
                                        fill = dhw_2024_category)) +
    geom_boxplot(alpha = 0.7, outlier.shape = NA) +
    geom_jitter(width = 0.2, alpha = 0.6, size = 2) +
    stat_summary(fun = mean, geom = "point", color = "red", size = 3, shape = 18) +
    scale_fill_viridis_d(name = "Current DHW\nCategory", option = "plasma") +
    labs(
      title = "Recovery Response by Current DHW Stress Category",
      subtitle = "Red diamonds show category means; data-driven quartile categories",
      x = "Current DHW Stress Category (2024)",
      y = "Recovery Achieved 2024→2025 (%)",
      caption = "This plot examines recovery patterns within data-driven categories of\ncurrent thermal
    ) +
    theme_minimal() +
    theme(
      axis.text.x = element_text(angle = 45, hjust = 1),
      legend.position = "none",
      plot.caption = element_text(hjust = 0, size = 9, color = "gray30")
    )

  ggsave("04_plot_recovery_by_dhw_category.png", p6, width = 12, height = 8, dpi = 300)
  cat("Saved: 04_plot_recovery_by_dhw_category.png\n")
}
```

```
## Saved: 04_plot_recovery_by_dhw_category.png
```

```
# ============================================================================
# EXTREME RESPONDER PREDICTION ANALYSIS
# ============================================================================

cat("\nSTEP 8: Analyzing prediction of extreme responders\n")
```

```
##
## STEP 8: Analyzing prediction of extreme responders
```

```
cat("--------------------------------------------------\n")
```

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

```
# Examine how well different predictors identify extreme responders
# Rationale: Extreme responders are often of greatest interest for
# understanding resilience mechanisms and vulnerability factors

# Define extreme responders using data-driven thresholds
response_quartiles <- quantile(predictive_dataset$recovery_2024_to_2025,
                               probs = c(0, 0.25, 0.5, 0.75, 1.0), na.rm = TRUE)

extreme_responders <- predictive_dataset %>%
  mutate(
    response_extreme_category = case_when(
      recovery_2024_to_2025 <= response_quartiles[1] + 0.1 ~ "No Recovery",
      recovery_2024_to_2025 >= response_quartiles[5] - 5 ~ "Exceptional Recovery",
      recovery_2024_to_2025 >= response_quartiles[4] ~ "Strong Recovery",
      recovery_2024_to_2025 <= response_quartiles[2] ~ "Poor Recovery",
      TRUE ~ "Moderate Recovery"
```

```r
    )
  ) %>%
  filter(response_extreme_category %in% c("No Recovery", "Poor Recovery", "Strong Recovery", "Exceptiona

cat(sprintf("Extreme responder analysis: %d sites\n", nrow(extreme_responders)))
```

```
## Extreme responder analysis: 16 sites
```

```r
# Analyze predictor patterns in extreme responders
extreme_analysis <- extreme_responders %>%
  group_by(response_extreme_category) %>%
  summarise(
    n_sites = n(),
    mean_prev_bleaching = mean(baseline_2023_annual, na.rm = TRUE),
    mean_prev_recovery = mean(recovery_2023_to_2024, na.rm = TRUE),
    .groups = "drop"
  )

if(!is.null(thermal_data)) {
  extreme_analysis <- extreme_responders %>%
    group_by(response_extreme_category) %>%
    summarise(
      n_sites = n(),
      mean_prev_bleaching = mean(baseline_2023_annual, na.rm = TRUE),
      mean_prev_recovery = mean(recovery_2023_to_2024, na.rm = TRUE),
      mean_dhw_2024 = mean(max_dhw_2024, na.rm = TRUE),
      mean_thermal_var = mean(thermal_variability, na.rm = TRUE),
      .groups = "drop"
    )
}

cat("Extreme responder predictor patterns:\n")
```

```
## Extreme responder predictor patterns:
```

```r
print(extreme_analysis)
```

```
## # A tibble: 3 x 6
##   response_extreme_category n_sites mean_prev_bleaching mean_prev_recovery
##   <chr>                       <int>               <dbl>              <dbl>
## 1 Exceptional Recovery            1                14.1               14.1
## 2 No Recovery                     8                25.5               21.0
## 3 Strong Recovery                 7                37.2               34.5
## # i 2 more variables: mean_dhw_2024 <dbl>, mean_thermal_var <dbl>
```

```r
# ============================================================================
# SAVE COMPREHENSIVE PREDICTIVE ANALYSIS RESULTS
# ============================================================================

cat("\nSTEP 9: Saving comprehensive predictive analysis results\n")
```

```
##
## STEP 9: Saving comprehensive predictive analysis results
```

```r
cat("--------------------------------------------------------\n")
```

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

```r
# Save all predictive analysis datasets
write_csv(predictive_dataset, "04_predictive_dataset_complete.csv")
write_csv(correlation_results, "04_correlation_analysis_detailed.csv")
write_csv(model_comparison, "04_model_performance_comparison.csv")
write_csv(prev_category_analysis, "04_previous_bleaching_category_analysis.csv")

if(!is.null(thermal_data)) {
  write_csv(dhw_category_analysis, "04_dhw_category_analysis.csv")
}

write_csv(extreme_responders, "04_extreme_responders_analysis.csv")

# Save model objects for future use
saveRDS(model_list, "04_predictive_models.rds")

# Save predictor thresholds for documentation
predictor_thresholds_data <- data.frame(
  predictor = c("prev_bleaching_low", "prev_bleaching_moderate", "prev_bleaching_high", "prev_bleaching_
  threshold = c(prev_bleaching_quartiles[2], prev_bleaching_quartiles[3],
                prev_bleaching_quartiles[4], prev_bleaching_quartiles[5]),
  percentile = c("25th", "50th", "75th", "100th"),
  description = c("Low previous impact threshold", "Moderate previous impact threshold",
                  "High previous impact threshold", "Severe previous impact threshold")
)

if(!is.null(thermal_data)) {
  thermal_thresholds <- data.frame(
    predictor = c("dhw_2024_low", "dhw_2024_moderate", "dhw_2024_high", "dhw_2024_extreme"),
    threshold = c(dhw_2024_quartiles[2], dhw_2024_quartiles[3],
                  dhw_2024_quartiles[4], dhw_2024_quartiles[5]),
    percentile = c("25th", "50th", "75th", "100th"),
    description = c("Low current stress threshold", "Moderate current stress threshold",
                    "High current stress threshold", "Extreme current stress threshold")
  )
  predictor_thresholds_data <- rbind(predictor_thresholds_data, thermal_thresholds)
}

write_csv(predictor_thresholds_data, "04_predictor_category_thresholds.csv")

# ============================================================================
# FINAL SUMMARY STATISTICS
# ============================================================================

cat("\nFINAL PREDICTIVE ANALYSIS SUMMARY\n")
```

```
##
## FINAL PREDICTIVE ANALYSIS SUMMARY
```

```r
cat("=================================\n")
```

```
## =================================
```

```r
cat(sprintf("Sites in predictive analysis: %d\n", nrow(predictive_dataset)))
```

```
## Sites in predictive analysis: 32
```

```r
cat(sprintf("Predictor variables tested: %d\n", length(predictor_vars)))
```

```
## Predictor variables tested: 10
```

```r
cat(sprintf("Models compared: %d\n", nrow(model_comparison)))
```

```
## Models compared: 8
```

```r
cat("\nTop 3 performing models:\n")
```

```
##
## Top 3 performing models:
```

```r
top_models <- head(model_comparison, 3)
for(i in 1:nrow(top_models)) {
  cat(sprintf("%d. %s: Adj R² = %.4f, RMSE = %.2f\n",
              i, top_models$Model[i], top_models$Adj_R_Squared[i], top_models$RMSE[i]))
}
```

```
## 1. Current_Baseline_Only: Adj R² = 0.8199, RMSE = 9.21
## 2. Thermal_Comprehensive: Adj R² = 0.1140, RMSE = 19.37
## 3. Comprehensive_Model: Adj R² = 0.0630, RMSE = 19.92
```

```r
cat("\nKey findings:\n")
```

```
##
## Key findings:
```

```r
if(!is.null(thermal_data)) {
  prev_r2 <- model_comparison$Adj_R_Squared[model_comparison$Model == "Previous_Bleaching_Only"]
  dhw_r2 <- model_comparison$Adj_R_Squared[model_comparison$Model == "Current_DHW_Only"]

  if(prev_r2 > dhw_r2) {
    cat(sprintf("- Previous bleaching is a stronger predictor (Adj R² = %.4f) than current DHW (Adj R² =
                prev_r2, dhw_r2))
  } else {
    cat(sprintf("- Current DHW is a stronger predictor (Adj R² = %.4f) than previous bleaching (Adj R² =
                dhw_r2, prev_r2))
  }
}
```

```
## - Previous bleaching is a stronger predictor (Adj R² = 0.0225) than current DHW (Adj R² = -0.0117)
```

```r
cat(sprintf("- Best single predictor: %s\n", correlation_results$Predictor[1]))
```

```
## - Best single predictor: 2024_Baseline
```

```r
cat(sprintf("- Strongest correlation: %.3f\n", correlation_results$Correlation[1]))
```

```
## - Strongest correlation: 0.909
```

```r
cat("\nFiles saved:\n")
```

```
##
## Files saved:
```

```r
cat("  - 04_predictive_dataset_complete.csv\n")
```

```
##   - 04_predictive_dataset_complete.csv
```

```r
cat("  - 04_correlation_analysis_detailed.csv\n")
```

```
##   - 04_correlation_analysis_detailed.csv
```

```r
cat("  - 04_model_performance_comparison.csv\n")
```

```
##   - 04_model_performance_comparison.csv
```

```r
cat("  - 04_previous_bleaching_category_analysis.csv\n")
```

```
##   - 04_previous_bleaching_category_analysis.csv
```

```r
if(!is.null(thermal_data)) {
  cat("  - 04_dhw_category_analysis.csv\n")
}
```

```
##   - 04_dhw_category_analysis.csv
```

```r
cat("  - 04_extreme_responders_analysis.csv\n")
```

```
##   - 04_extreme_responders_analysis.csv
```

```r
cat("  - 04_predictive_models.rds\n")
```

```
##   - 04_predictive_models.rds
```

```r
cat("  - 04_predictor_category_thresholds.csv\n")
```

```
##   - 04_predictor_category_thresholds.csv
```

```r
cat("\nVisualizations saved:\n")
```

```
##
## Visualizations saved:
```

```r
cat("  - 04_plot_correlation_matrix.png\n")
```

```
##   - 04_plot_correlation_matrix.png
```

```r
cat("  - 04_plot_model_performance.png\n")
```

```
##   - 04_plot_model_performance.png
```

```r
cat("  - 04_plot_previous_bleaching_relationship.png\n")
```

```
##   - 04_plot_previous_bleaching_relationship.png
```

```r
if(!is.null(thermal_data)) {
  cat("  - 04_plot_current_dhw_relationship.png\n")
  cat("  - 04_plot_recovery_by_dhw_category.png\n")
}
```

```
##   - 04_plot_current_dhw_relationship.png
##   - 04_plot_recovery_by_dhw_category.png
```

```r
cat("  - 04_plot_recovery_by_previous_category.png\n")
```

```
##   - 04_plot_recovery_by_previous_category.png
```

```r
cat("\n============================================================================\n")
```

```
##
## ============================================================================
```

```r
cat("STEP 4 COMPLETE: Comprehensive predictive analysis using data-driven approaches\n")
```

```
## STEP 4 COMPLETE: Comprehensive predictive analysis using data-driven approaches
```

```r
cat("Next: Advanced visualization and synthesis\n")
```

```
## Next: Advanced visualization and synthesis
```

```r
cat("================================================================================\n")
```

```
## ================================================================================
```