Palmer Penguins Data Analysis Series (Part 5): Random Forest vs Linear Models - The Final Comparison

Settling the interpretability vs performance debate in ecological modeling

Your Name

2025-01-05

Table of contents

Figure 1: Two penguins at a crossroads - one holding a linear regression equation, the other holding a decision tree, representing the classic interpretability vs performance trade-off!

Photo: African penguins at Boulders Beach, South Africa. Licensed under CC BY 2.0 via Wikimedia Commons

Palmer Penguins Data Analysis Series - FINALE

This is **Part 5** (Final) of our 5-part series exploring penguin morphometrics:

1. Part 1: EDA and Simple Regression
2. Part 2: Multiple Regression and Species Effects
3. Part 3: Advanced Models and Cross-Validation
4. Part 4: Model Diagnostics and Interpretation
5. **Part 5: Random Forest vs Linear Models** (This post - FINALE!)

1 Introduction

Welcome to the grand finale of our Palmer penguins data analysis journey! We've traveled far together - from basic exploratory analysis to sophisticated diagnostics, building and validating models that explain 86% of the variance in penguin body mass. But one crucial question has been lurking beneath the surface throughout our journey:

Should we sacrifice the beautiful interpretability of linear models for potentially better predictive performance from machine learning algorithms?

This question sits at the heart of modern data science, especially in scientific research where understanding *why* is often as important as predicting *what*. Today, we'll conduct a comprehensive head-to-head comparison between our carefully crafted linear models and their machine learning challenger: random forests.

In this final post, we'll explore:

- Comprehensive performance comparison using multiple metrics
- Feature importance analysis and partial dependence plots
- The interpretability-performance tradeoff in action
- Practical guidance for model selection in ecological research
- When to choose linear models vs. random forests
- Best practices for communicating model choices to stakeholders

By the end of this series finale, you'll have a complete framework for making informed decisions about model complexity in your own research.

2 Setup and Model Arsenal

Let's assemble our complete modeling toolkit and establish fair comparison conditions:

```
library(palmerpenguins)
library(tidyverse)
library(randomForest)
library(caret)
library(broom)
library(knitr)
library(patchwork)

# Conditional loading of optional packages
optional_packages <- c("pdp", "vip", "DALEX")
for (pkg in optional_packages) {
  if (requireNamespace(pkg, quietly = TRUE)) {
```

```
    library(pkg, character.only = TRUE)
    cat("[OK] Loaded", pkg, "\n")
  } else {
    cat("[INFO] Package '", pkg, "' not available. Install with: install.packages('", pkg, "
  }
}
```

```
[INFO] Package ' pdp ' not available. Install with: install.packages(' pdp ')
[INFO] Package ' vip ' not available. Install with: install.packages(' vip ')
[INFO] Package ' DALEX ' not available. Install with: install.packages(' DALEX ')
```

```
# Set theme and colors
theme_set(theme_minimal(base_size = 12))
penguin_colors <- c("Adelie" = "#FF6B6B", "Chinstrap" = "#4ECDC4", "Gentoo" = "#45B7D1")

# Load and prepare data
data(penguins)
penguins_clean <- penguins %>% drop_na()

# Set seed for reproducibility
set.seed(42)

cat("=== FINALE: The Ultimate Model Comparison ===\n")
```

```
=== FINALE: The Ultimate Model Comparison ===
```

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

```
============================================
```

```
cat(sprintf("Dataset: %d penguins across %d variables\n", nrow(penguins_clean), ncol(penguins
```

```
Dataset: 333 penguins across 8 variables
```

```
cat("Comparison: Linear Models vs Random Forests\n")
```

```
Comparison: Linear Models vs Random Forests
```

```r
cat("Evaluation: Cross-validation with multiple metrics\n")
```

```
Evaluation: Cross-validation with multiple metrics
```

```r
cat("Goal: Determine optimal approach for ecological modeling\n")
```

```
Goal: Determine optimal approach for ecological modeling
```

3 Comprehensive Model Training

Let's train our complete arsenal of models using identical cross-validation procedures:

3.1 Cross-Validation Setup

```r
# Consistent cross-validation setup
train_control <- trainControl(
  method = "cv",
  number = 10,
  savePredictions = "final",
  verboseIter = FALSE
)

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

```
--- Experimental Design ---
```

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

```
========================
```

```r
cat("Cross-validation: 10-fold\n")
```

```
Cross-validation: 10-fold
```

```r
cat("Seed: 42 (consistent across all models)\n")
```

```
Seed: 42 (consistent across all models)
```

```r
cat("Metrics: RMSE, R-squared, MAE\n")
```

```
Metrics: RMSE, R-squared, MAE
```

```r
cat("Evaluation: Performance + interpretability\n")
```

```
Evaluation: Performance + interpretability
```

3.2 Linear Model Variants

```r
# Simple linear model
linear_simple <- train(
  body_mass_g ~ flipper_length_mm,
  data = penguins_clean,
  method = "lm",
  trControl = train_control
)

# Multiple regression
linear_multiple <- train(
  body_mass_g ~ bill_length_mm + bill_depth_mm + flipper_length_mm,
  data = penguins_clean,
  method = "lm",
  trControl = train_control
)

# Species-aware model (our champion from previous parts)
linear_species <- train(
  body_mass_g ~ bill_length_mm + bill_depth_mm + flipper_length_mm + species,
  data = penguins_clean,
  method = "lm",
  trControl = train_control
)
```

```r
# Polynomial model
linear_poly <- train(
  body_mass_g ~ poly(flipper_length_mm, 2) + poly(bill_length_mm, 2) +
                poly(bill_depth_mm, 2) + species,
  data = penguins_clean,
  method = "lm",
  trControl = train_control
)

cat("[OK] Linear models trained:\n")
```

```
[OK] Linear models trained:
```

```r
cat("• Simple (flipper only)\n")
```

```
• Simple (flipper only)
```

```r
cat("• Multiple (all morphometrics)\n")
```

```
• Multiple (all morphometrics)
```

```r
cat("• Species-aware (morphometrics + species)\n")
```

```
• Species-aware (morphometrics + species)
```

```r
cat("• Polynomial (quadratic features + species)\n")
```

```
• Polynomial (quadratic features + species)
```

3.3 Random Forest Variants

```r
# Basic random forest (morphometrics only)
rf_basic <- train(
  body_mass_g ~ bill_length_mm + bill_depth_mm + flipper_length_mm,
  data = penguins_clean,
  method = "rf",
```

```
  trControl = train_control,
  ntree = 500,
  importance = TRUE
)
```

note: only 2 unique complexity parameters in default grid. Truncating the grid to 2 .

```
# Full random forest (all available predictors)
rf_full <- train(
  body_mass_g ~ bill_length_mm + bill_depth_mm + flipper_length_mm +
                species + sex + island + year,
  data = penguins_clean,
  method = "rf",
  trControl = train_control,
  ntree = 500,
  importance = TRUE
)

# Tuned random forest (optimized hyperparameters)
rf_tuned <- train(
  body_mass_g ~ bill_length_mm + bill_depth_mm + flipper_length_mm +
                species + sex + island,
  data = penguins_clean,
  method = "rf",
  trControl = train_control,
  tuneGrid = expand.grid(mtry = c(2, 3, 4, 5)),
  ntree = 500,
  importance = TRUE
)

cat("\n[OK] Random forest models trained:\n")
```

[OK] Random forest models trained:

```
cat("• Basic (morphometrics only)\n")
```

• Basic (morphometrics only)

```
cat("• Full (all predictors)\n")
```

• Full (all predictors)

```
cat("• Tuned (optimized hyperparameters)\n")
```

• Tuned (optimized hyperparameters)

4 Performance Comparison

4.1 Comprehensive Performance Table

```
# Compile all models
all_models <- list(
  "Linear: Simple" = linear_simple,
  "Linear: Multiple" = linear_multiple,
  "Linear: Species" = linear_species,
  "Linear: Polynomial" = linear_poly,
  "RF: Basic" = rf_basic,
  "RF: Full" = rf_full,
  "RF: Tuned" = rf_tuned
)

# Extract performance metrics
performance_comparison <- map_dfr(all_models, function(model) {
  results <- model$results
  best_idx <- which.min(results$RMSE)

  data.frame(
    RMSE_mean = results$RMSE[best_idx],
    RMSE_sd = sd(model$resample$RMSE),
    Rsquared_mean = results$Rsquared[best_idx],
    Rsquared_sd = sd(model$resample$Rsquared),
    MAE_mean = results$MAE[best_idx],
    MAE_sd = sd(model$resample$MAE)
  )
}, .id = "Model") %>%
  arrange(RMSE_mean) %>%
```

```
  mutate(
    Rank = row_number(),
    Model_Type = ifelse(str_detect(Model, "Linear"), "Linear", "Random Forest")
  )

# Format for display
performance_display <- performance_comparison %>%
  mutate(
    RMSE = sprintf("%.1f ± %.1f", RMSE_mean, RMSE_sd),
    R_squared = sprintf("%.3f ± %.3f", Rsquared_mean, Rsquared_sd),
    MAE = sprintf("%.1f ± %.1f", MAE_mean, MAE_sd)
  ) %>%
  select(Rank, Model, Model_Type, RMSE, R_squared, MAE)

kable(performance_display,
      caption = "Complete Model Performance Comparison (Cross-Validated)",
      col.names = c("Rank", "Model", "Type", "RMSE (g)", "R²", "MAE (g)"))
```

Table 1: Complete Model Performance Comparison (Cross-Validated)

Rank	Model	Type	RMSE (g)	R^2	MAE (g)
1	RF: Tuned	Random Forest	294.9 ± 45.6	0.866 ± 0.050	235.9 ± 38.5
2	RF: Full	Random Forest	296.0 ± 29.9	0.870 ± 0.024	236.7 ± 25.2
3	Linear: Polynomial	Linear	310.3 ± 42.9	0.858 ± 0.035	249.4 ± 34.1
4	Linear: Species	Linear	315.7 ± 32.2	0.856 ± 0.022	251.6 ± 24.8
5	RF: Basic	Random Forest	341.4 ± 36.2	0.827 ± 0.042	269.0 ± 38.7
6	Linear: Simple	Linear	390.9 ± 54.0	0.775 ± 0.038	314.1 ± 42.9
7	Linear: Multiple	Linear	392.6 ± 41.7	0.769 ± 0.049	312.9 ± 27.3

```
# Identify top performers
top_linear <- performance_comparison %>% filter(Model_Type == "Linear") %>% slice_min(RMSE_me
top_rf <- performance_comparison %>% filter(Model_Type == "Random Forest") %>% slice_min(RMSE

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

```
=== Top Performers ===
```

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

```
==================
```

```
cat(sprintf("Best Linear Model: %s (RMSE: %.1f)\n", top_linear$Model, top_linear$RMSE_mean))
```

```
Best Linear Model: Linear: Polynomial (RMSE: 310.3)
```

```
cat(sprintf("Best Random Forest: %s (RMSE: %.1f)\n", top_rf$Model, top_rf$RMSE_mean))
```

```
Best Random Forest: RF: Tuned (RMSE: 294.9)
```

```
cat(sprintf("Performance Gap: %.1f grams RMSE\n", top_linear$RMSE_mean - top_rf$RMSE_mean))
```

```
Performance Gap: 15.4 grams RMSE
```

4.2 Statistical Significance Testing

```
# Compare top models statistically
top_models <- list(
  "Best Linear" = all_models[[top_linear$Model]],
  "Best RF" = all_models[[top_rf$Model]]
)

model_resamples <- resamples(top_models)
summary(model_resamples)
```

```
Call:
summary.resamples(object = model_resamples)

Models: Best Linear, Best RF
Number of resamples: 10

MAE
                Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
Best Linear 200.8293 224.4082 246.0062 249.3694 262.8090 313.4748    0
```

```
Best RF     192.8103 213.8498 226.3612 235.9357 248.0156 321.0243    0

RMSE
               Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
Best Linear 249.1883 278.4448 300.9560 310.2954 335.3805 376.1442    0
Best RF     240.0675 261.9729 278.6417 294.9201 319.6756 381.0386    0

Rsquared
                 Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
Best Linear 0.7875702 0.8395449 0.8599082 0.8579618 0.8831732 0.9028425    0
Best RF     0.7337503 0.8651822 0.8712287 0.8663053 0.8872005 0.9195034    0
```

```
# Statistical test
model_diff <- diff(model_resamples)
summary(model_diff)
```

```
Call:
summary.diff.resamples(object = model_diff)

p-value adjustment: bonferroni
Upper diagonal: estimates of the difference
Lower diagonal: p-value for H0: difference = 0

MAE
            Best Linear Best RF
Best Linear             13.43
Best RF     0.2413

RMSE
            Best Linear Best RF
Best Linear             15.38
Best RF     0.4359

Rsquared
            Best Linear Best RF
Best Linear             -0.008343
Best RF     0.5137
```

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

```
--- Statistical Comparison ---
```

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

```
==========================
```

```
cat("Testing: Best Linear vs Best Random Forest\n")
```

```
Testing: Best Linear vs Best Random Forest
```

```
cat("Metric: RMSE difference\n")
```

```
Metric: RMSE difference
```

4.3 Performance Visualization

```
# Create comprehensive comparison plots
all_cv_results <- map_dfr(all_models, function(model) {
  data.frame(
    RMSE = model$resample$RMSE,
    Rsquared = model$resample$Rsquared,
    MAE = model$resample$MAE
  )
}, .id = "Model") %>%
  mutate(
    Model_Type = ifelse(str_detect(Model, "Linear"), "Linear", "Random Forest"),
    Model = factor(Model, levels = performance_comparison$Model)
  )

# RMSE comparison
p1 <- ggplot(all_cv_results, aes(x = Model, y = RMSE, fill = Model_Type)) +
  geom_boxplot(alpha = 0.7) +
  stat_summary(fun = mean, geom = "point", shape = 23, size = 3, fill = "white") +
  scale_fill_manual(values = c("Linear" = "#E74C3C", "Random Forest" = "#27AE60")) +
  labs(title = "RMSE Comparison Across All Models",
       subtitle = "Lower is better; white diamonds show means",
       y = "RMSE (grams)", fill = "Model Type") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```