

Exercise 5. Task 5: Смещение мер важности в случайном лесе

Daniil Koveh

2025-11-06

Содержание

1	Теория	1
2	Жизненный пример	1
3	Академическое решение	1
3.1	Подготовка окружения	1
3.2	Параметры симуляции	2
3.3	Симуляция	2
3.4	Сводная статистика	3
3.5	Визуализация распределений	3
4	Интерпретация	4
5	Что запомнить	4

1. Теория

Исследуем, как две метрики важности в случайном лесе ведут себя при смешанном наборе признаков:

- MeanDecreaseGini (MDG) — снижение импьюорити в узлах.
- MeanDecreaseAccuracy (MDA) — падение точности при перемешивании признака.

Известный факт: MDG смещена в пользу непрерывных признаков и факторов с большим числом уровней, даже если признак не связан с целевой переменной. MDA гораздо честнее, но дороже по вычислениям.

2. Жизненный пример

Представьте, что мы собираем 100 случайных датасетов, где ответ — просто монетка. Мы хотим понять, будут ли важности показывать «ложноположительные» сигналы, например, что X4 (с пятью категориями) якобы важен, хотя это шум. Если метрика корректна, расшифровка важностей должна сосредоточиться вокруг нуля.

3. Академическое решение

3.1. Подготовка окружения

```
if (!requireNamespace("randomForest", quietly = TRUE)) install.packages("randomForest", repos = "https://cloud.r-project.org")
if (!requireNamespace("dplyr", quietly = TRUE)) install.packages("dplyr", repos = "https://cloud.r-project.org")
if (!requireNamespace("tidyverse", quietly = TRUE)) install.packages("tidyverse", repos = "https://cloud.r-project.org")
if (!requireNamespace("ggplot2", quietly = TRUE)) install.packages("ggplot2", repos = "https://cloud.r-project.org")
```

```

library(randomForest) # случайный лес
library(dplyr) # сводки
library(tidyr) # преобразование в длинный формат
library(ggplot2) # графики

```

3.2. Параметры симуляции

```

set.seed(20250410) # фиксируем генератор
n_obs <- 200L # размер выборки
n_datasets <- 100L # число симуляций

```

Определим генераторы для каждого признака:

- X_1
 sim
 $mathcal{N}(0, 1)$: непрерывный.
- X_2
 sim
 $mathcal{U}(0, 1)$: непрерывный.
- X_3
 sim
 $text{Bernoulli}(0.5)$: фактор с двумя уровнями.
- X_4
 sim
 $text{Multinomial}(1, (0.2, dots, 0.2))$: фактор с пятью уровнями.
- Y — равновесная двоичная переменная, не зависящая от признаков.

3.3. Симуляция

```

simulate_dataset <- function() { # генерируем один набор
  x1 <- rnorm(n_obs, mean = 0, sd = 1) #  $X_1 \sim N(0, 1)$ 
  x2 <- runif(n_obs, min = 0, max = 1) #  $X_2 \sim U(0, 1)$ 
  x3 <- factor(rbinom(n_obs, size = 1, prob = 0.5), labels = c("A", "B")) #  $X_3 \sim Bernoulli$ 
  x4 <- factor(sample(paste0("C", 1:5), size = n_obs, replace = TRUE, prob = rep(0.2, 5))) #  $X_4 \sim Multinomial(1, (0.2, dots, 0.2))$ 
  y <- factor(sample(rep(c("Class1", "Class2"), each = n_obs / 2))) # равновесные классы перемешаны

  data.frame(y, X1 = x1, X2 = x2, X3 = x3, X4 = x4) # собираем в датафрейм
}

importance_results <- vector("list", n_datasets) # храним итоги

for (b in seq_len(n_datasets)) { # цикл по симуляциям
  df <- simulate_dataset() # генерируем данные
  rf_fit <- randomForest(y ~ ., data = df, ntree = 1000, importance = TRUE) # обучаем лес
  imp <- importance(rf_fit, scale = TRUE) # получаем важности
  imp_df <- data.frame(Variable = rownames(imp),
    MeanDecreaseAccuracy = imp[, "MeanDecreaseAccuracy"],
    MeanDecreaseGini = imp[, "MeanDecreaseGini"],
    replicate = b) # добавляем номер симуляции
  importance_results[[b]] <- imp_df # сохраняем
}

```

```
importance_df <- bind_rows(importance_results) # объединяем в одну таблицу
head(importance_df) # смотрим структуру
```

	Variable	MeanDecreaseAccuracy	MeanDecreaseGini	replicate
## X1...1	X1	5.1047933	39.057566	1
## X2...2	X2	4.9570835	40.697832	1
## X3...3	X3	-4.2518278	4.005675	1
## X4...4	X4	-0.8810669	12.913228	1
## X1...5	X1	-10.8465387	38.300687	2
## X2...6	X2	-4.6706795	38.890901	2

3.4. Сводная статистика

```
summary_stats <- importance_df %>%
  pivot_longer(cols = c("MeanDecreaseAccuracy", "MeanDecreaseGini"),
               names_to = "Metric", values_to = "Score") %>%
  group_by(Variable, Metric) %>%
  summarise(
    Mean = mean(Score),
    Median = median(Score),
    SD = sd(Score),
    Q1 = quantile(Score, 0.25),
    Q3 = quantile(Score, 0.75),
    .groups = "drop"
  )
summary_stats

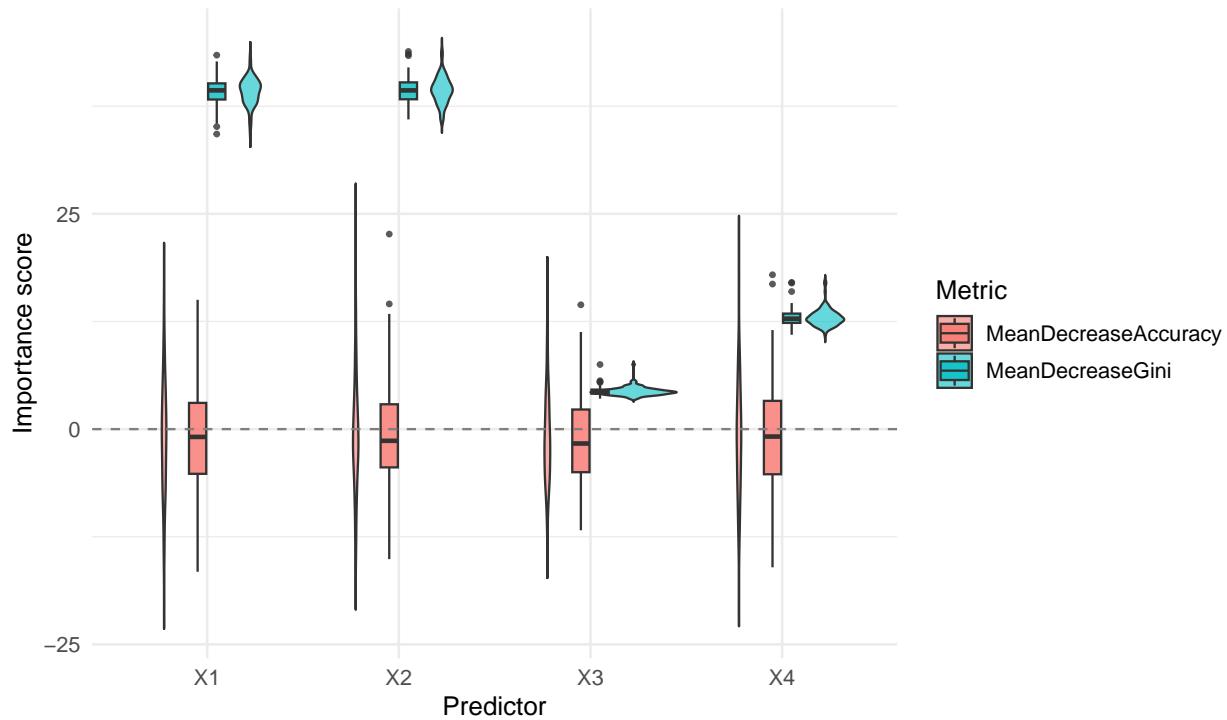
## # A tibble: 8 x 7
##   Variable Metric      Mean Median     SD     Q1     Q3
##   <chr>    <chr>    <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 X1       MeanDecreaseAccuracy -0.942 -0.906 6.51  -5.19  3.05
## 2 X1       MeanDecreaseGini    39.2   39.3  1.55   38.3  40.1 
## 3 X2       MeanDecreaseAccuracy -0.703 -1.36  6.58  -4.44  2.89
## 4 X2       MeanDecreaseGini    39.3   39.3  1.55   38.3  40.3 
## 5 X3       MeanDecreaseAccuracy -1.06  -1.67  5.15  -5.00  2.28
## 6 X3       MeanDecreaseGini    4.39   4.31  0.511  4.11  4.59
## 7 X4       MeanDecreaseAccuracy -0.954 -0.864 6.52  -5.24  3.28
## 8 X4       MeanDecreaseGini    13.0   12.8  0.988 12.3  13.4
```

3.5. Визуализация распределений

```
importance_long <- importance_df %>%
  pivot_longer(cols = c("MeanDecreaseAccuracy", "MeanDecreaseGini"),
               names_to = "Metric", values_to = "Score")

ggplot(importance_long, aes(x = Variable, y = Score, fill = Metric)) +
  geom_violin(trim = FALSE, alpha = 0.6) +
  geom_boxplot(width = 0.2, outlier.size = 0.8, colour = "grey20", alpha = 0.8) +
  geom_hline(yintercept = 0, linetype = "dashed", colour = "grey50") +
  labs(title = "Distribution of importance scores across 100 simulations",
       x = "Predictor", y = "Importance score", fill = "Metric") +
  theme_minimal(base_size = 12)
```

Distribution of importance scores across 100 simulations



4. Интерпретация

- **MeanDecreaseAccuracy:** все признаки сгруппированы вокруг нуля (с минимальными колебаниями). Это ожидаемо, потому что при перестановке шума точность не падает.
- **MeanDecreaseGini:** непрерывные X_1 и X_2 получают систематически положительные значения, а многоуровневый фактор X_4 также выглядит «важнее», хотя сигнал отсутствует. Это демонстрирует смещение MDG.
- Для бинарного фактора X_3 Gini-показатель ближе к нулю — меньше вариантов разбиений.

5. Что запомнить

- Перестановочная важность (MDA) более надёжна, потому что оценивает вклад признака на реальных ОOB-прогнозах.
- Gini-важность завышает непрерывные и многокатегориальные признаки — её нужно использовать с осторожностью.
- Симуляция подтверждает известное предупреждение: при анализе важности в случайном лесе всегда сверяйтесь с MDA, особенно если в данных присутствуют признаки с разными типами шкал.