

Fast Interpretable Greedy-Tree Sums with **figsr**: CHEATSHEET



What is FIGS?

FIGS grows a small sum of compact, easy-to-read trees instead of one deep tree: $\hat{y} = \sum_{k=1}^K f_k(x)$. At every step it looks at one pool of candidates (a brand-new root, or a split at any leaf of any tree grown so far) and accepts whichever candidate cuts the residual sum of squares the most. It stops once no candidate's gain is above $1e-6$, so the final sum stays small and each tree stays simple to read.

Install

```
install.packages("figsr")
# development version
pak::pak("bonijoao/figsr")
```

Fit a model: `figs()`

```
fit <- figs(y ~ x1 + x2 + log(x3),
  data = train, max_splits = 10,
  max_trees = NULL, min_n = 5,
  mode = "regression", na_method = "omit")
```

<code>max_splits</code>	total splits (stop rule)
<code>max_trees</code>	cap on trees (NULL=unlim.)
<code>min_n</code>	min. obs. per node
<code>mode</code>	"regression"/"classif."
<code>na_method</code>	"omit" OR "mia"

Variable importance

```
figsr_importance(fit)
figsr_importance(fit,
  relative = FALSE)
```

Sums each split's SSE-reduction (gain) by feature. `relative = TRUE` (default) rescales to % of total. Under "mia", a split on missingness alone is its own row: `missing(<predictor>)`.

Predict

```
predict(fit, new_data = test)
predict(fit, new_data = test,
  type = "prob")
predict_figs(fit, new_data = test,
  type = "class")
```

type: "numeric", "class" OR "prob" (2-class only). Returns a tidy tibble (.pred, .pred_class, .pred_level).

Inspect: print & summary

```
print(fit) # tree sum, glance
summary(fit) # IF-THEN rules
```

```
Tree 1:
|-- x1 < 0.42
|  |-- value: -1.03
|-- x1 >= 0.42
   |-- value: 0.87
```

Tuning: dials

```
max_splits(range = c(2L, 40L))
max_trees(range = c(1L, 5L))
grid <- dials::grid_regular(
  max_splits(), max_trees(),
  levels = 4
)
tune::tune_grid(spec, resamples,
  grid = grid)
```

Plot: `plot()`

```
plot(fit) # one panel/tree
plot(fit, tree_idx = 1)
plot(fit, style = "modern")
```

Base graphics only, see plot styles below.

Ensembles: `bagging_figs()`

```
bag <- bagging_figs(y ~ ., data = train,
  n_estimators = 10, max_splits = 6)
predict(bag, new_data = test)
```

Bootstrap ensemble of independent FIGS fits, averaged at predict time.

tidymodels: `figs_tree()`

```
library(parsnip)
spec <- figs_tree(max_splits = 8,
  max_trees = 3) |>
  set_engine("figsr") |>
  set_mode("regression")
spec |> fit(y ~ ., data = train)
```

Parsnip engine "figsr". Works with workflows, recipes, tune.

Plot styles



Editorial, pastel rounded, and high-contrast minimal, in that order.

Gotchas

- Classification: **two classes only**; leaf values already estimate $P(y=class_2)$ (no sigmoid).
- $\log(x)$ ok; $\text{poly}(x)$ rejected at fit time.
- A missing value on an untrained branch, or an unseen factor level, errors at predict time.