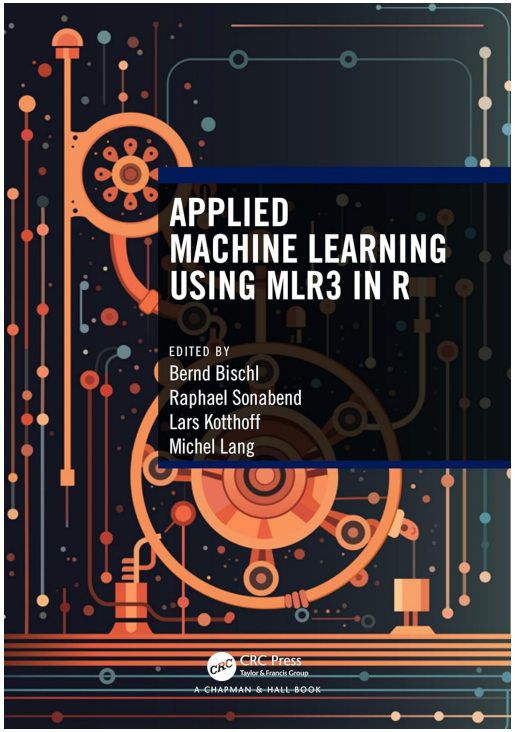
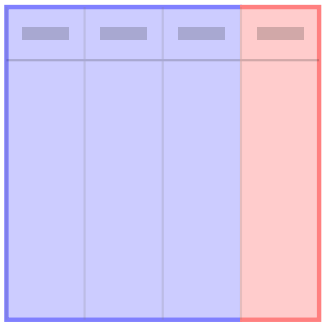
Machine Learning in R Using mlr3

- **Website:** https://mlr-org.com/
- **Github:** https://github.com/mlr-org
- **Book:** https://mlr3book.mlr-org.com/



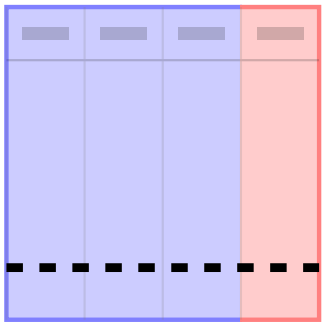
April 16, 2024

RESAMPLING



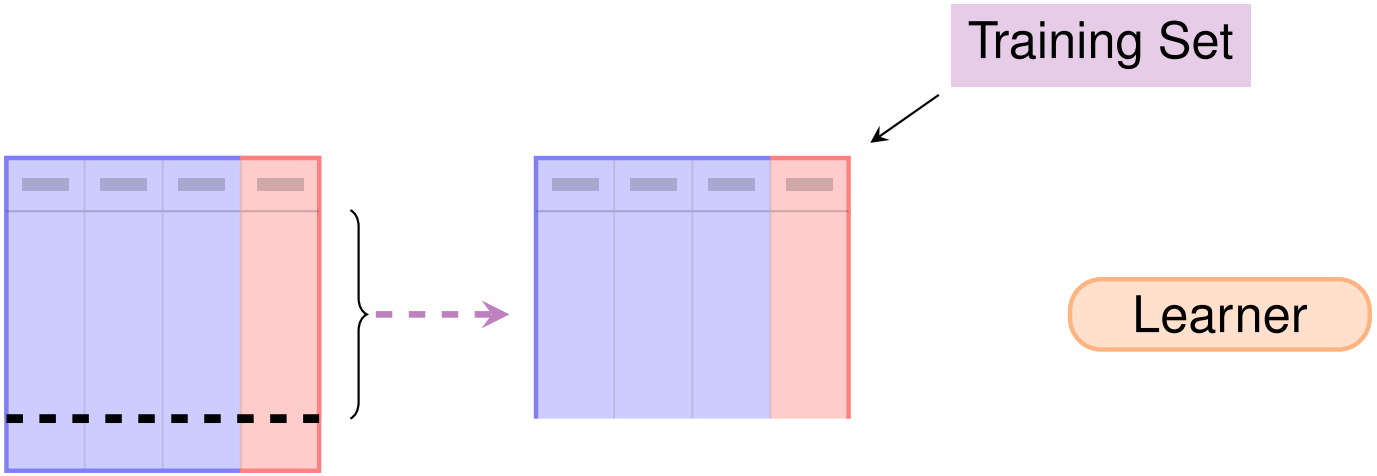
Learner

RESAMPLING

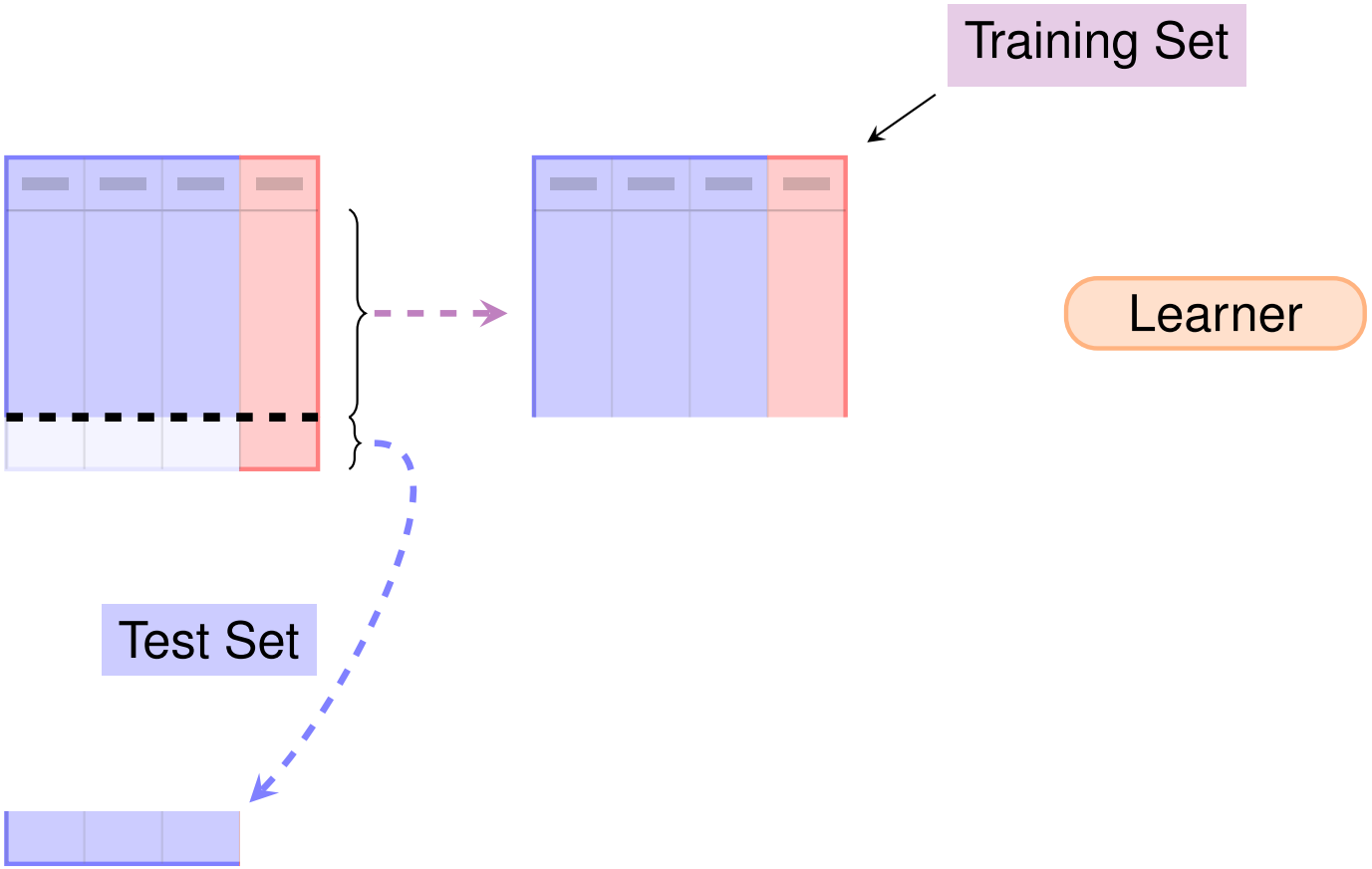


Learner

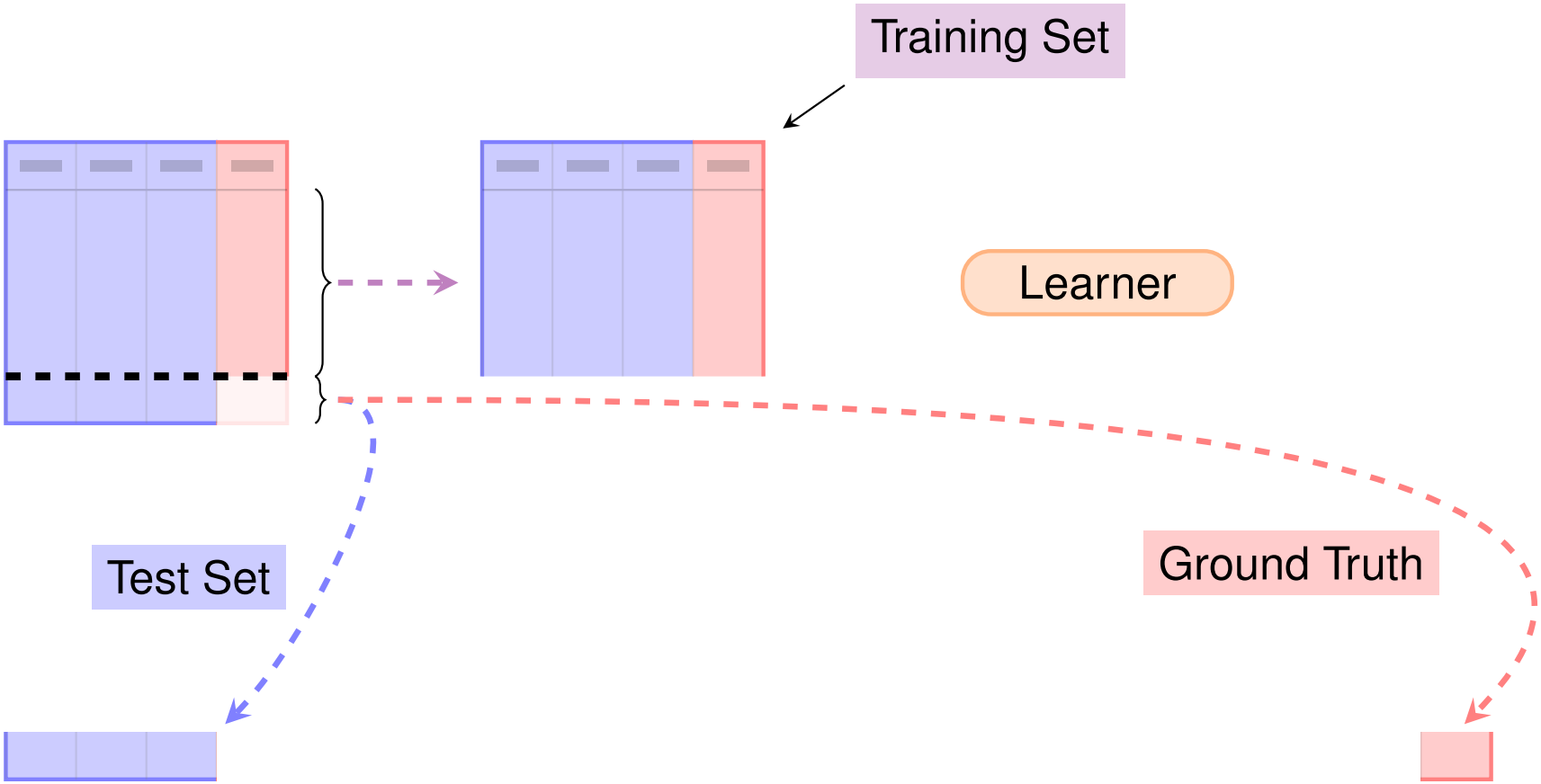
RESAMPLING



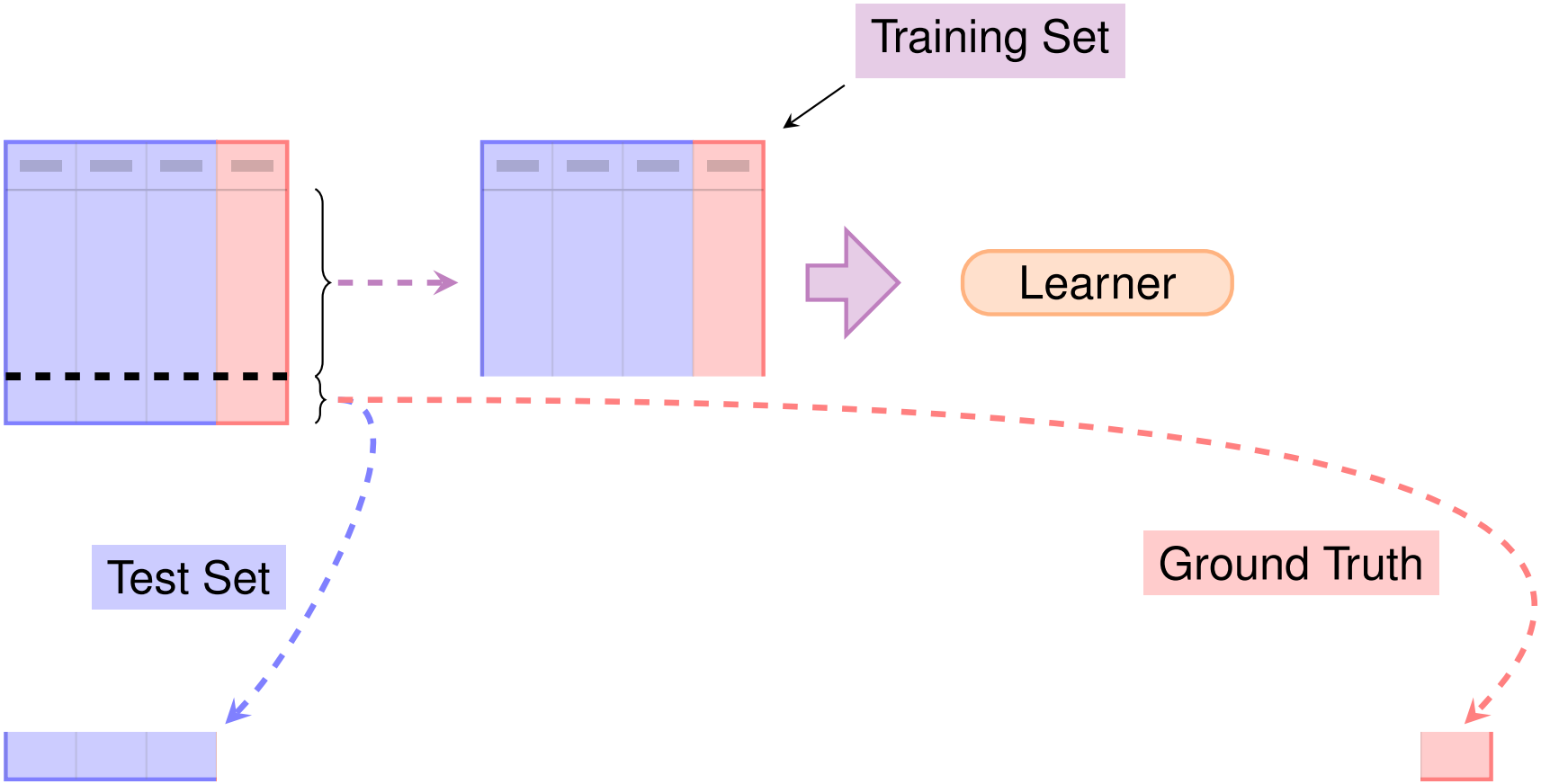
RESAMPLING



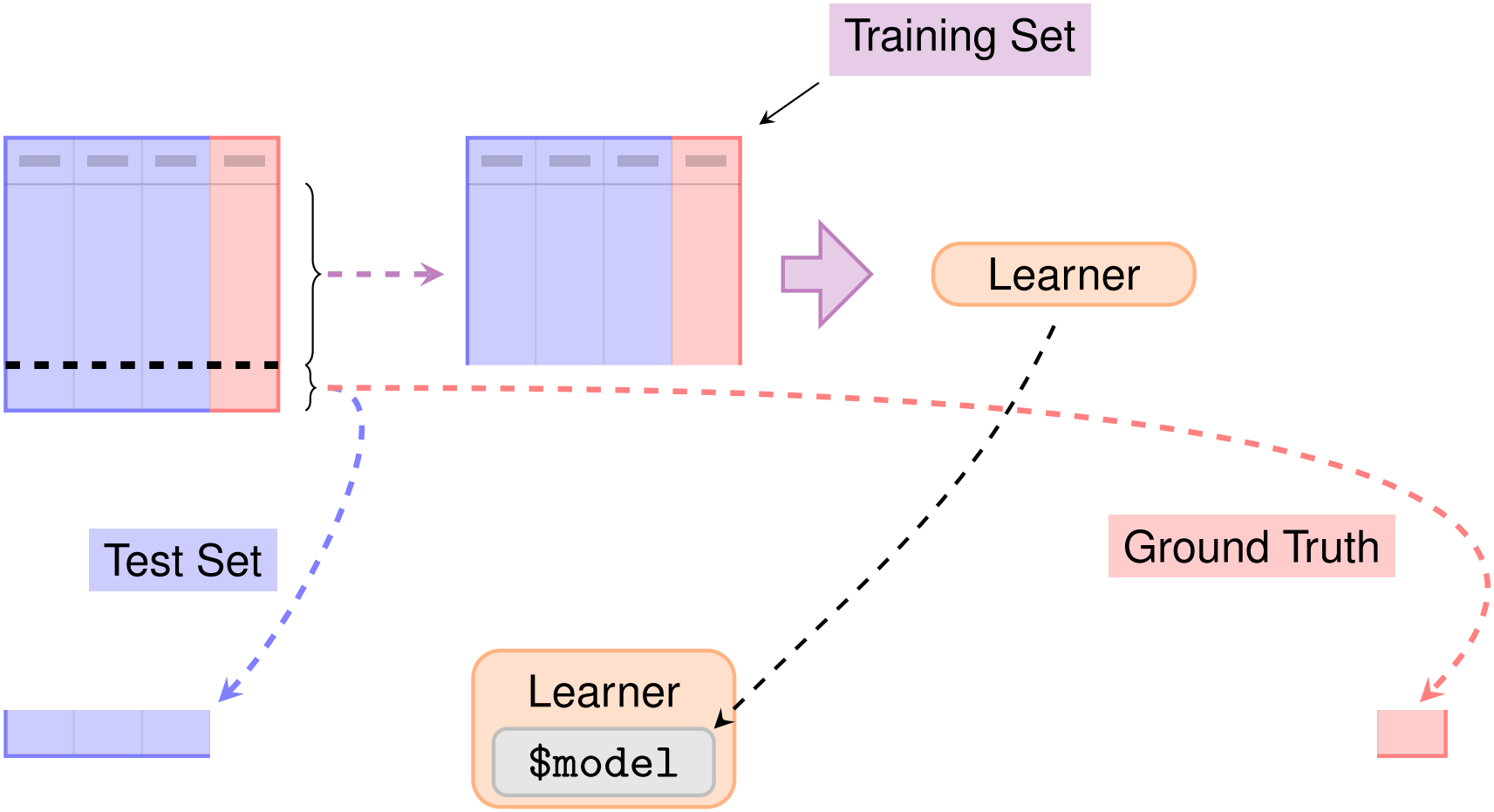
RESAMPLING



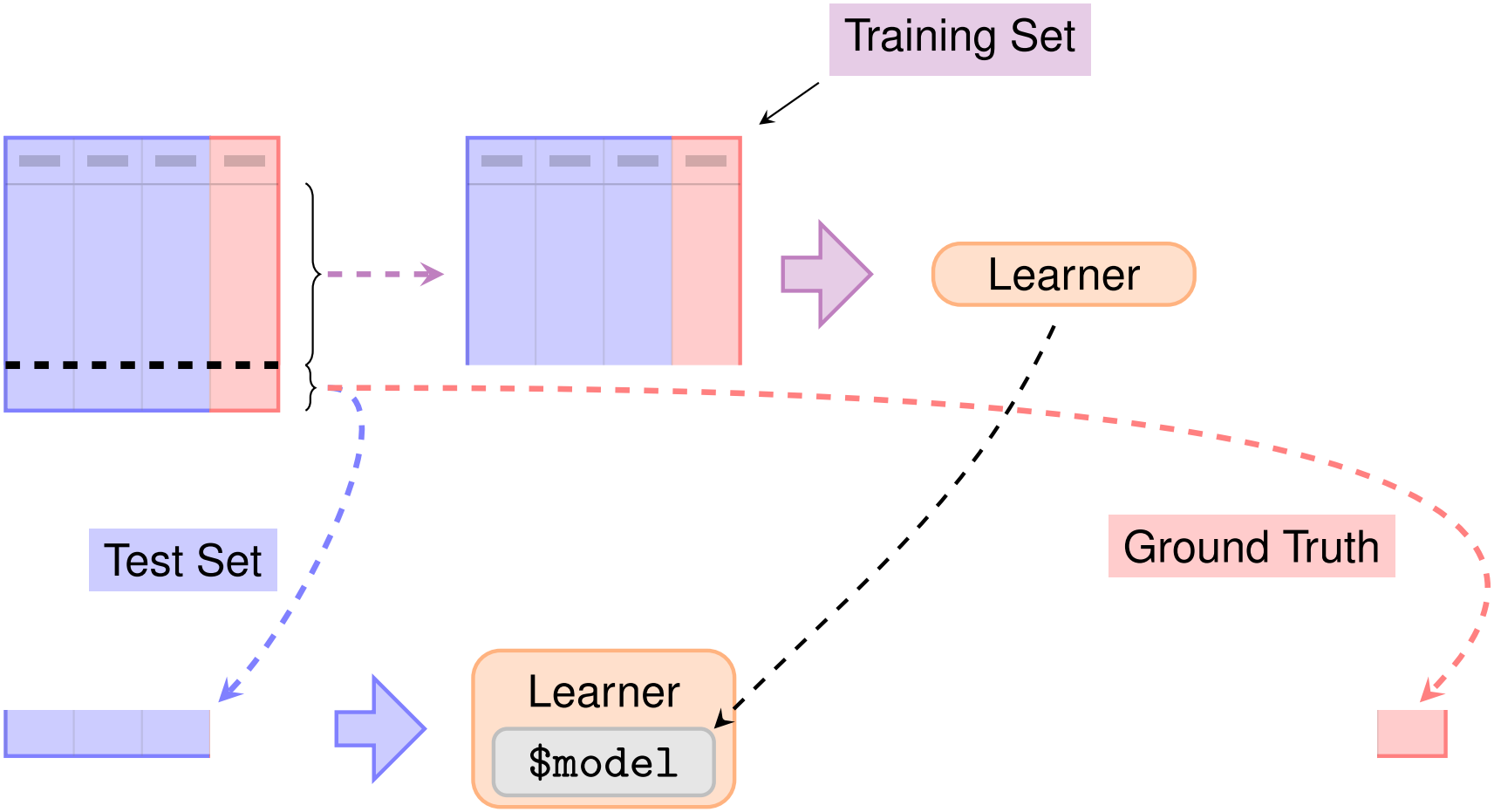
RESAMPLING



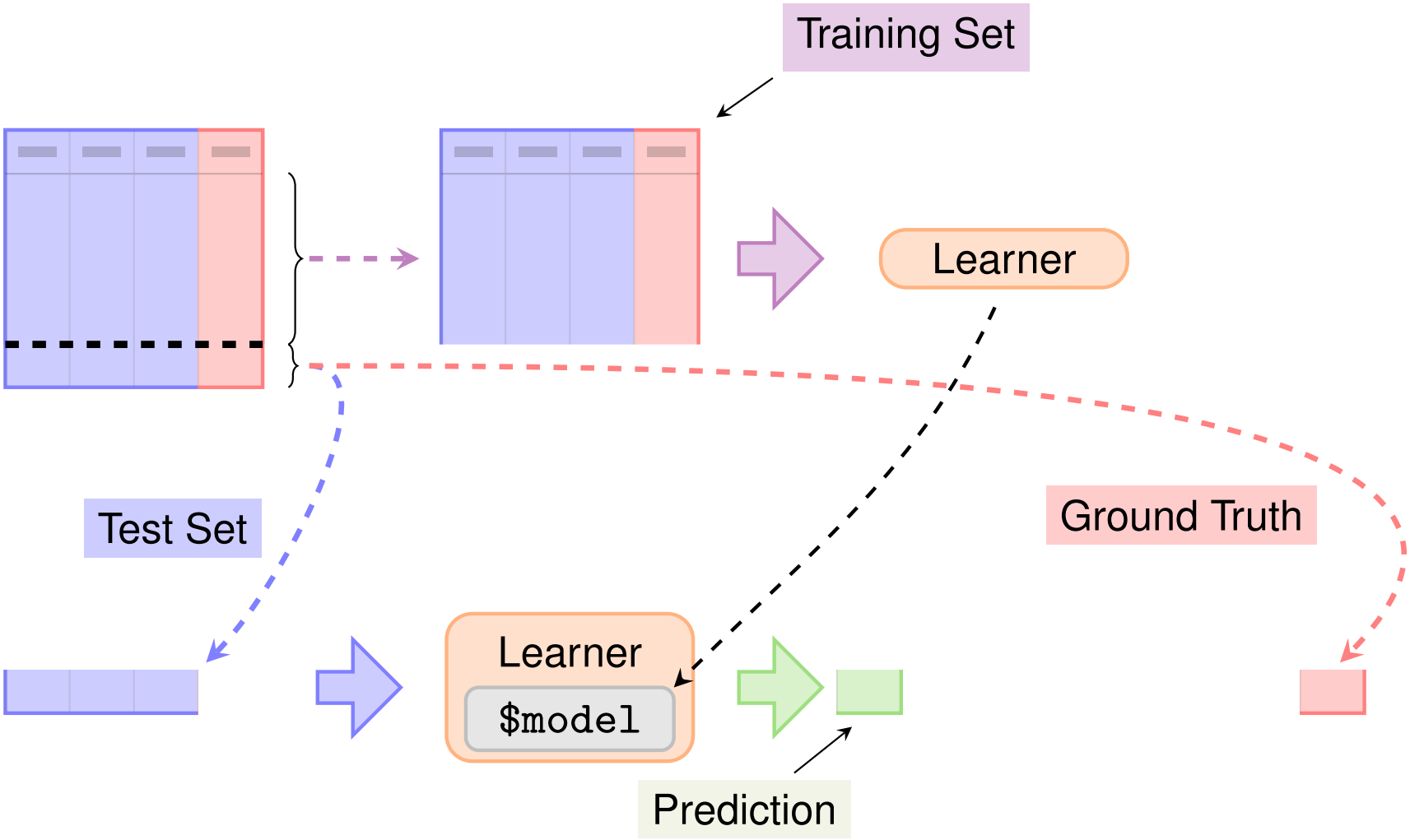
RESAMPLING



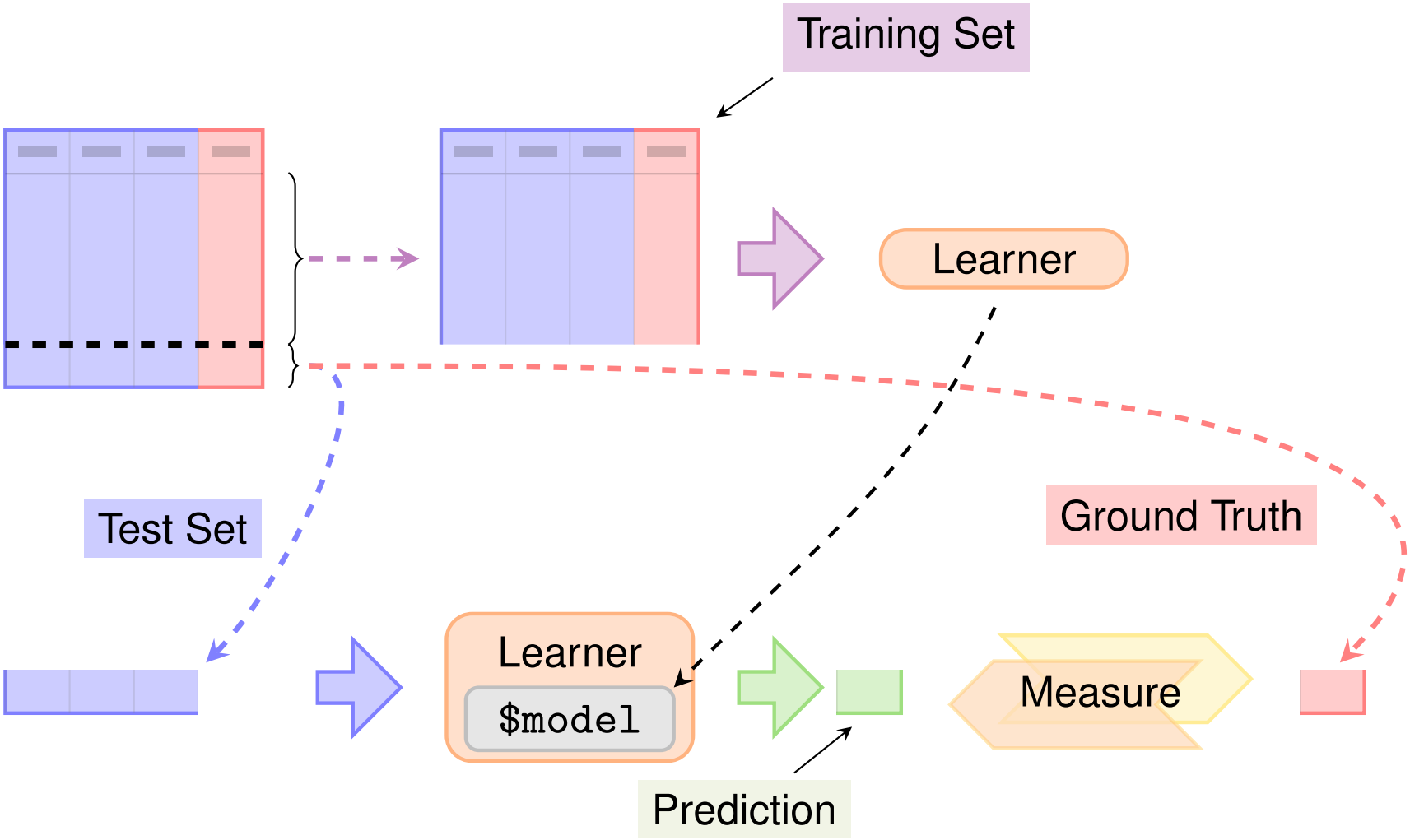
RESAMPLING



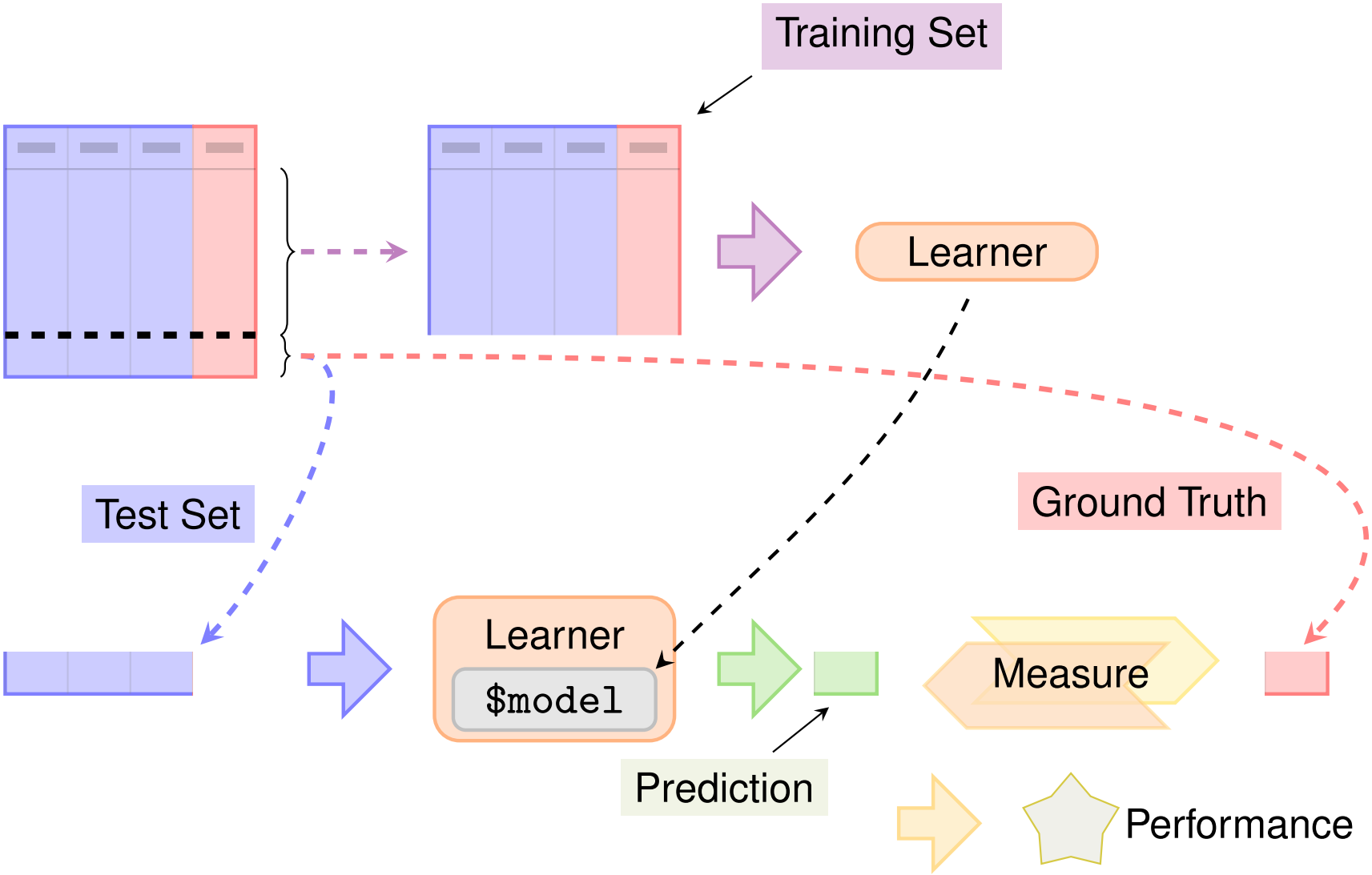
RESAMPLING


Training Set
Learner
Test Set
Ground Truth
Learner
$model
Prediction

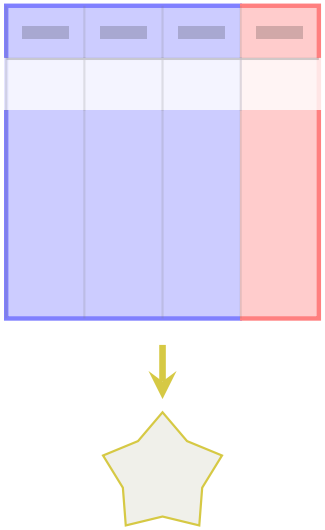
RESAMPLING


Training Set
Learner
Test Set
Ground Truth
Learner
$model
Measure
Prediction

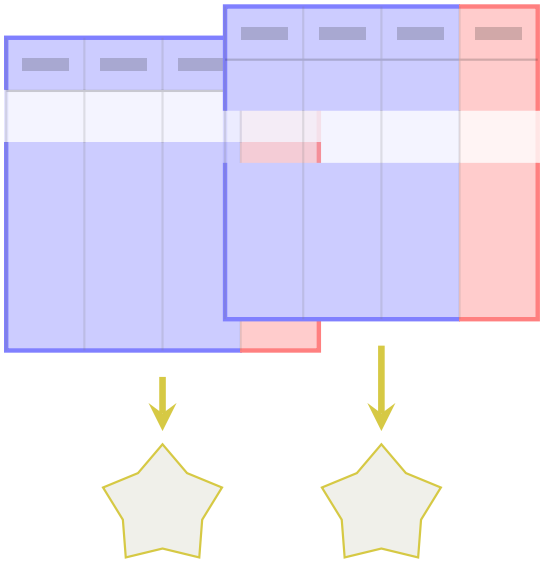
RESAMPLING


Training Set
Learner
Test Set
Ground Truth
Learner
$model
Measure
Prediction
Performance

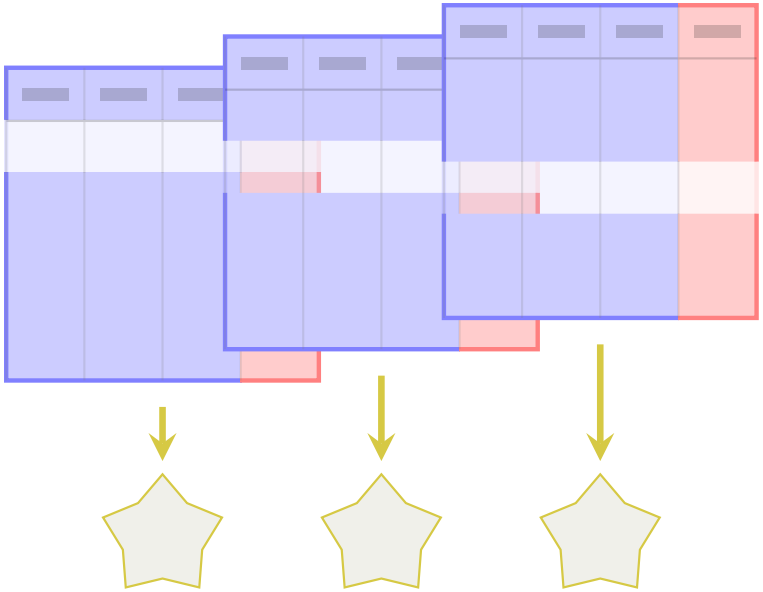
RESAMPLING



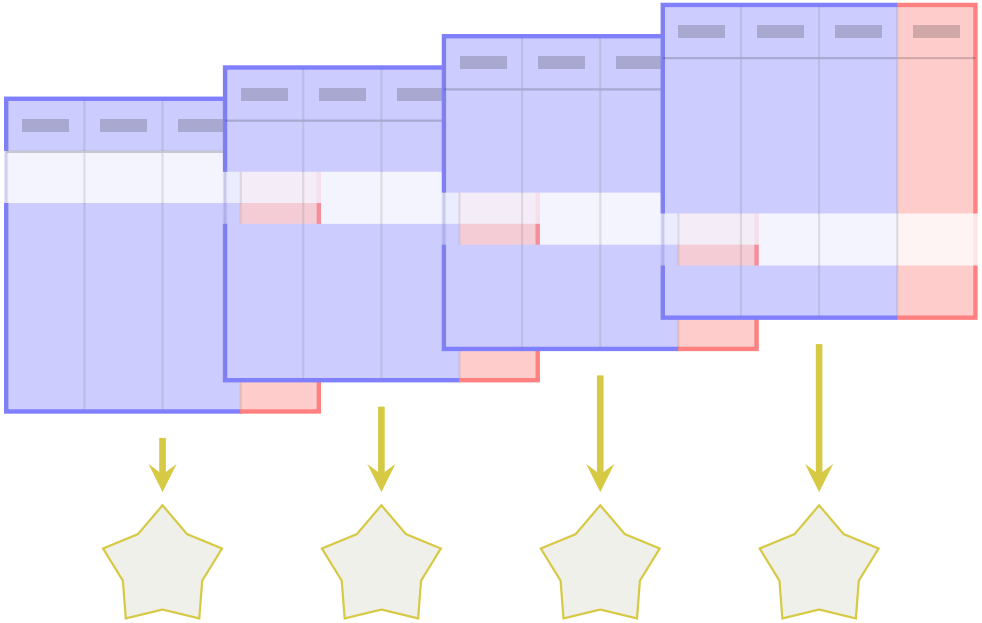
RESAMPLING



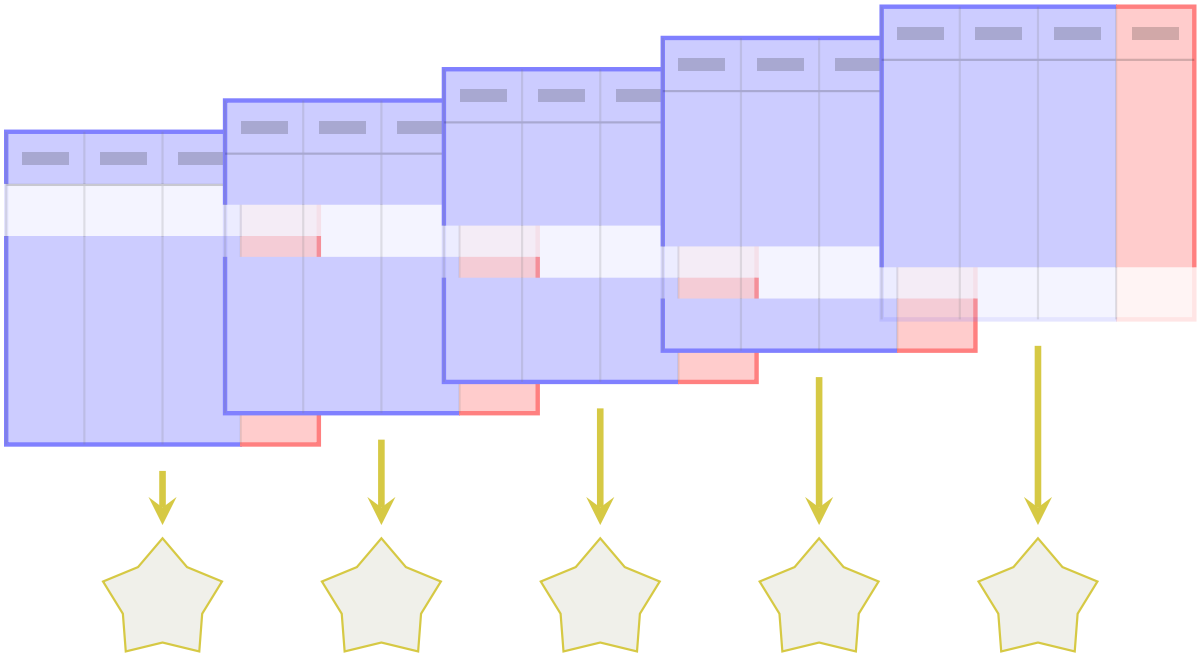
RESAMPLING



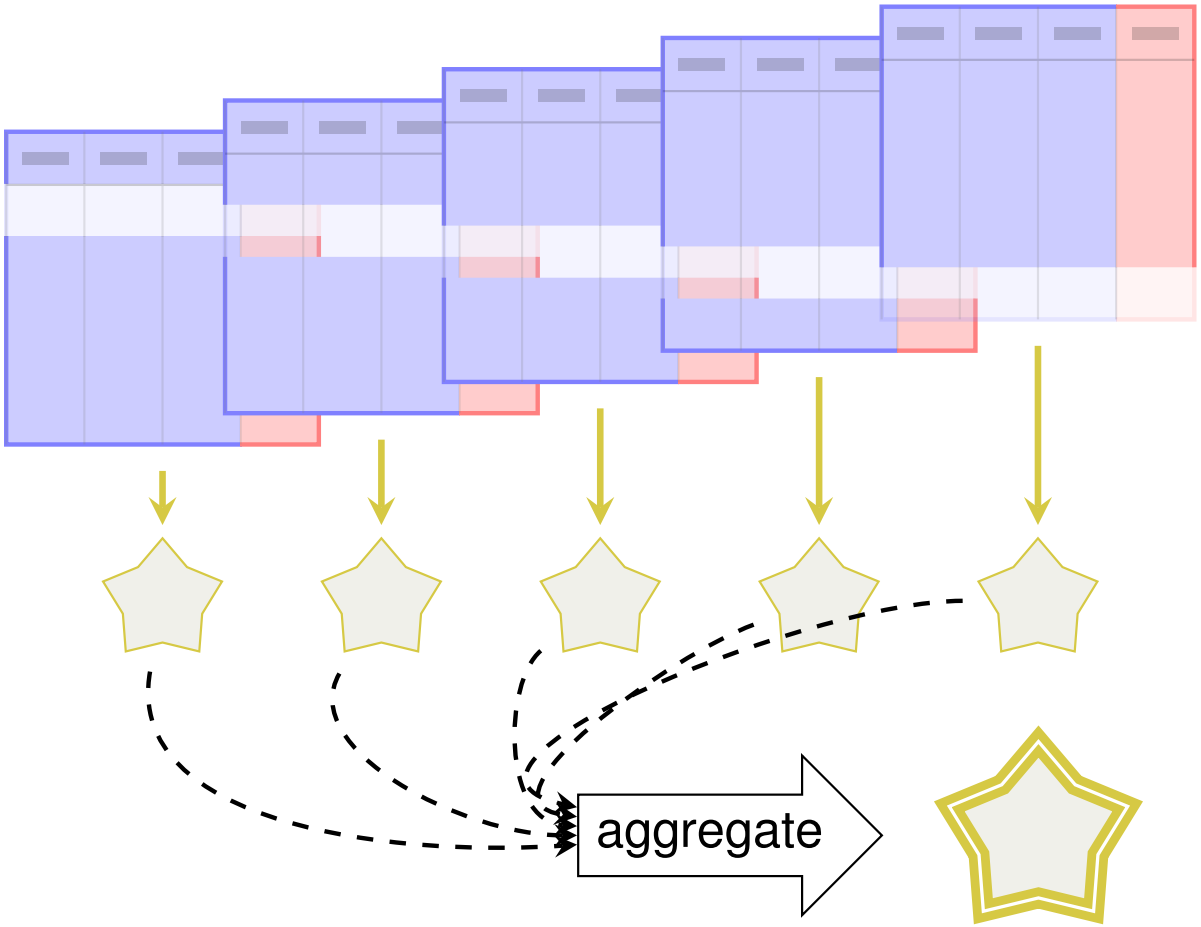
RESAMPLING



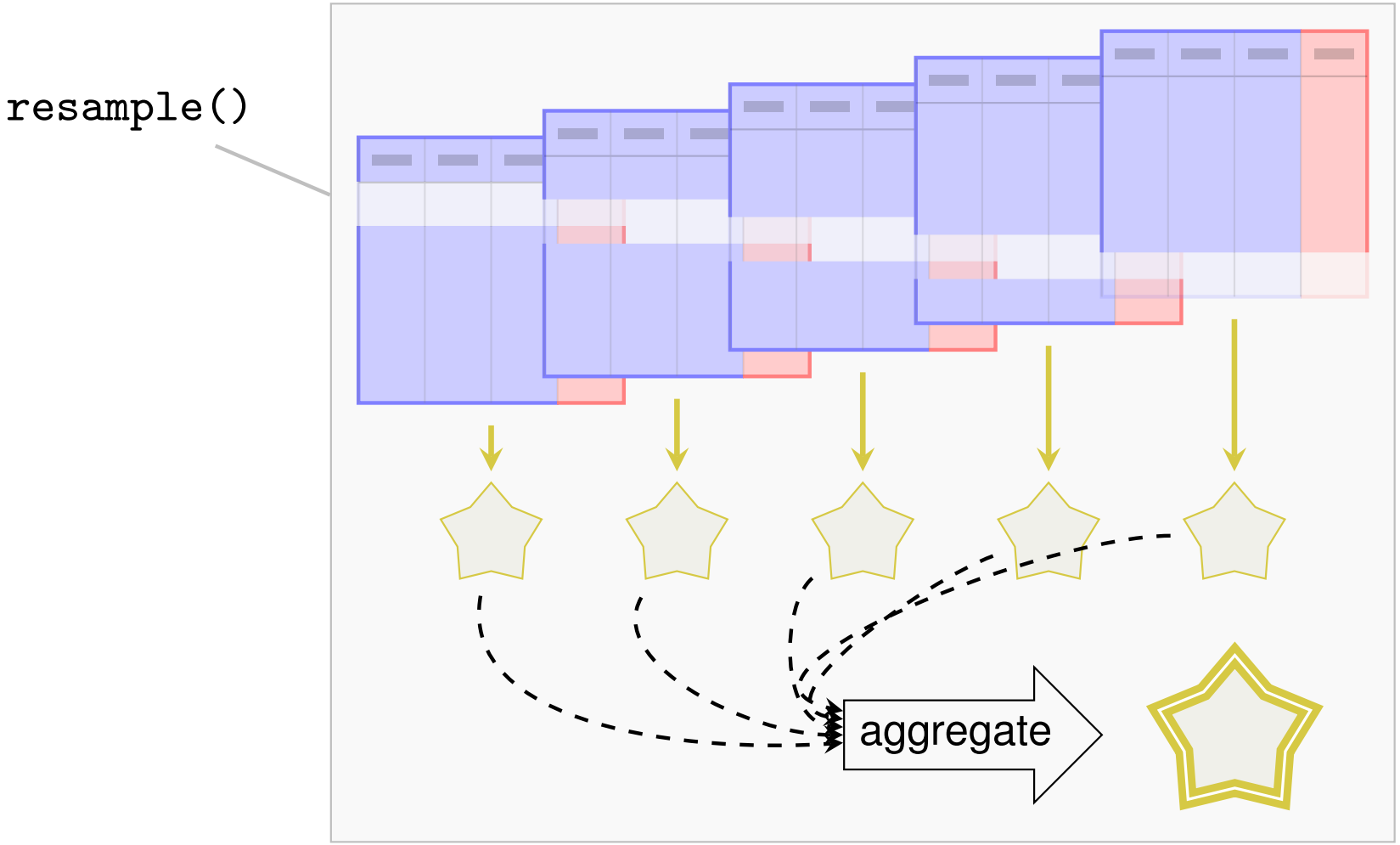
RESAMPLING



RESAMPLING



RESAMPLING



RESAMPLING

- Resample description: How to split the data

```
cv5 = rsmp("cv", folds = 5)
```

RESAMPLING

- Resample description: How to split the data

```
cv5 = rsmp("cv", folds = 5)
```

- Use the `resample()` function for resampling:

```
task = as_task_classif(x = iris, target = "Species", id = "iris")
learner = lrn("classif.rpart")
rr = resample(task, learner, cv5)
```

RESAMPLING

- Resample description: How to split the data

```
cv5 = rsmp("cv", folds = 5)
```

- Use the `resample()` function for resampling:

```
task = as_task_classif(x = iris, target = "Species", id = "iris")
learner = lrn("classif.rpart")
rr = resample(task, learner, cv5)
```

- We get a `ResamplingResult` object:

```
print(rr)
#> <ResampleResult> with 5 resampling iterations
#>  task_id    learner_id resampling_id iteration warnings errors
#>     iris classif.rpart            cv         1        0      0
#>     iris classif.rpart            cv         2        0      0
#>     iris classif.rpart            cv         3        0      0
#>     iris classif.rpart            cv         4        0      0
#>     iris classif.rpart            cv         5        0      0
```

RESAMPLING RESULTS

What exactly is a `ResamplingResult` object?

RESAMPLING RESULTS

What exactly is a `ResamplingResult` object?
Remember `Prediction`:

RESAMPLING RESULTS

What exactly is a `ResamplingResult` object?
Remember `Prediction`:

- Get a table representation using `as.data.table()`

```
rr_table = as.data.table(rr)

print(rr_table)
#                   task                                learner
#                 <list>                                 <list>
# 1: <TaskClassif:iris> <LearnerClassifRpart:classif.rpart>
# 2: <TaskClassif:iris> <LearnerClassifRpart:classif.rpart>
# 3: <TaskClassif:iris> <LearnerClassifRpart:classif.rpart>
# 4: <TaskClassif:iris> <LearnerClassifRpart:classif.rpart>
# 5: <TaskClassif:iris> <LearnerClassifRpart:classif.rpart>
#        resampling iteration          prediction
#            <list>     <int>              <list>
# 1: <ResamplingCV>         1 <PredictionClassif>
# 2: <ResamplingCV>         2 <PredictionClassif>
# 3: <ResamplingCV>         3 <PredictionClassif>
# 4: <ResamplingCV>         4 <PredictionClassif>
# 5: <ResamplingCV>         5 <PredictionClassif>
```

RESAMPLING RESULTS

What exactly is a `ResamplingResult` object?
Remember `Prediction`:

- Get a table representation using `as.data.table()`

```
rr_table = as.data.table(rr)

print(rr_table)
#                   task                                learner
#                 <list>                                 <list>
# 1: <TaskClassif:iris> <LearnerClassifRpart:classif.rpart>
# 2: <TaskClassif:iris> <LearnerClassifRpart:classif.rpart>
# 3: <TaskClassif:iris> <LearnerClassifRpart:classif.rpart>
# 4: <TaskClassif:iris> <LearnerClassifRpart:classif.rpart>
# 5: <TaskClassif:iris> <LearnerClassifRpart:classif.rpart>
#        resampling iteration          prediction
#            <list>     <int>              <list>
# 1: <ResamplingCV>         1 <PredictionClassif>
# 2: <ResamplingCV>         2 <PredictionClassif>
# 3: <ResamplingCV>         3 <PredictionClassif>
# 4: <ResamplingCV>         4 <PredictionClassif>
# 5: <ResamplingCV>         5 <PredictionClassif>
```

- Active bindings and functions that make information easily accessible

RESAMPLING RESULTS

- Calculate performance:

```
rr$aggregate(msr("classif.ce"))
#> classif.ce
#>       0.06
```

RESAMPLING RESULTS

- Calculate performance:

```
rr$aggregate(msr("classif.ce"))
#> classif.ce
#>       0.06
```

- Get predictions

```
rr$prediction()
#> <PredictionClassif> for 150 observations:
#>     row_ids     truth  response
#>           2    setosa    setosa
#>           7    setosa    setosa
#>           9    setosa    setosa
#> ---
#>         141 virginica virginica
#>         148 virginica virginica
#>         149 virginica virginica
```

RESAMPLING

- Predictions of individual folds

```
predictions = rr$predictions()
predictions[[1]]
#> <PredictionClassif> for 30 observations:
#>     row_ids     truth   response
#>           2    setosa     setosa
#>           7    setosa     setosa
#>           9    setosa     setosa
#> ---
#>         135 virginica versicolor
#>         136 virginica  virginica
#>         142 virginica  virginica
```

RESAMPLING

- Predictions of individual folds

```
predictions = rr$predictions()
predictions[[1]]
#> <PredictionClassif> for 30 observations:
#>     row_ids     truth   response
#>           2    setosa     setosa
#>           7    setosa     setosa
#>           9    setosa     setosa
#> ---
#>         135 virginica versicolor
#>         136 virginica  virginica
#>         142 virginica  virginica
```

- Score of individual folds

```
scores = rr$score()
scores[1:3, c("iteration", "classif.ce")]
#>    iteration classif.ce
#>        <int>      <num>
#> 1:         1      0.067
#> 2:         2      0.100
#> 3:         3      0.067
```

Resampling

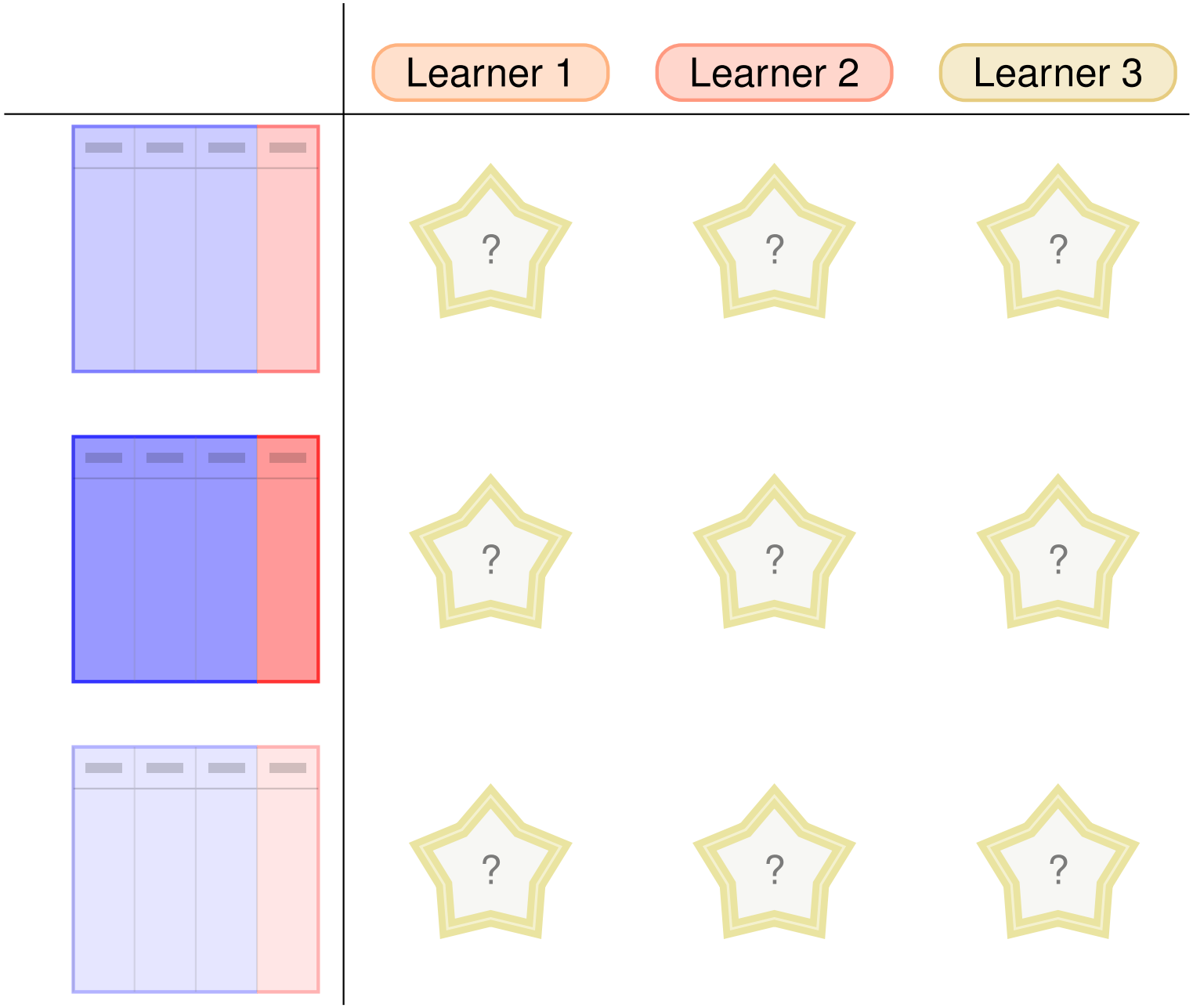
Benchmark

Control of Execution

How to get Help

Outro

PERFORMANCE COMPARISON



PERFORMANCE COMPARISON

- Multiple Learners, multiple Tasks:

```
library("mlr3learners")
learners = list(lrn("classif.rpart"), lrn("classif.kknn"))
tasks = list(tsk("iris"), tsk("sonar"), tsk("wine"))
```

PERFORMANCE COMPARISON

- Multiple Learners, multiple Tasks:

```
library("mlr3learners")
learners = list(lrn("classif.rpart"), lrn("classif.kknn"))
tasks = list(tsk("iris"), tsk("sonar"), tsk("wine"))
```

- Set up the *design* and execute benchmark:

```
design = benchmark_grid(tasks, learners, cv5)
bmr = benchmark(design)
```

PERFORMANCE COMPARISON

- Multiple Learners, multiple Tasks:

```
library("mlr3learners")
learners = list(lrn("classif.rpart"), lrn("classif.kknn"))
tasks = list(tsk("iris"), tsk("sonar"), tsk("wine"))
```

- Set up the *design* and execute benchmark:

```
design = benchmark_grid(tasks, learners, cv5)
bmr = benchmark(design)
```

- We get a `BenchmarkResult` object which shows that `kknn` outperforms `rpart`:

```
bmr_ag = bmr$aggregate()
bmr_ag[, c("task_id", "learner_id", "classif.ce")]
#>    task_id    learner_id classif.ce
#>     <char>        <char>      <num>
#> 1:    iris classif.rpart      0.053
#> 2:    iris  classif.kknn      0.040
#> 3:   sonar classif.rpart      0.274
#> 4:   sonar  classif.kknn      0.130
#> 5:    wine classif.rpart      0.157
#> 6:    wine  classif.kknn      0.039
```

BENCHMARK RESULT

What exactly is a `BenchmarkResult` object?

BENCHMARK RESULT

What exactly is a `BenchmarkResult` object?
Just like `Prediction` and `ResamplingResult`!

BENCHMARK RESULT

What exactly is a `BenchmarkResult` object?
Just like `Prediction` and `ResamplingResult`!

- Table representation using `as.data.table()`

BENCHMARK RESULT

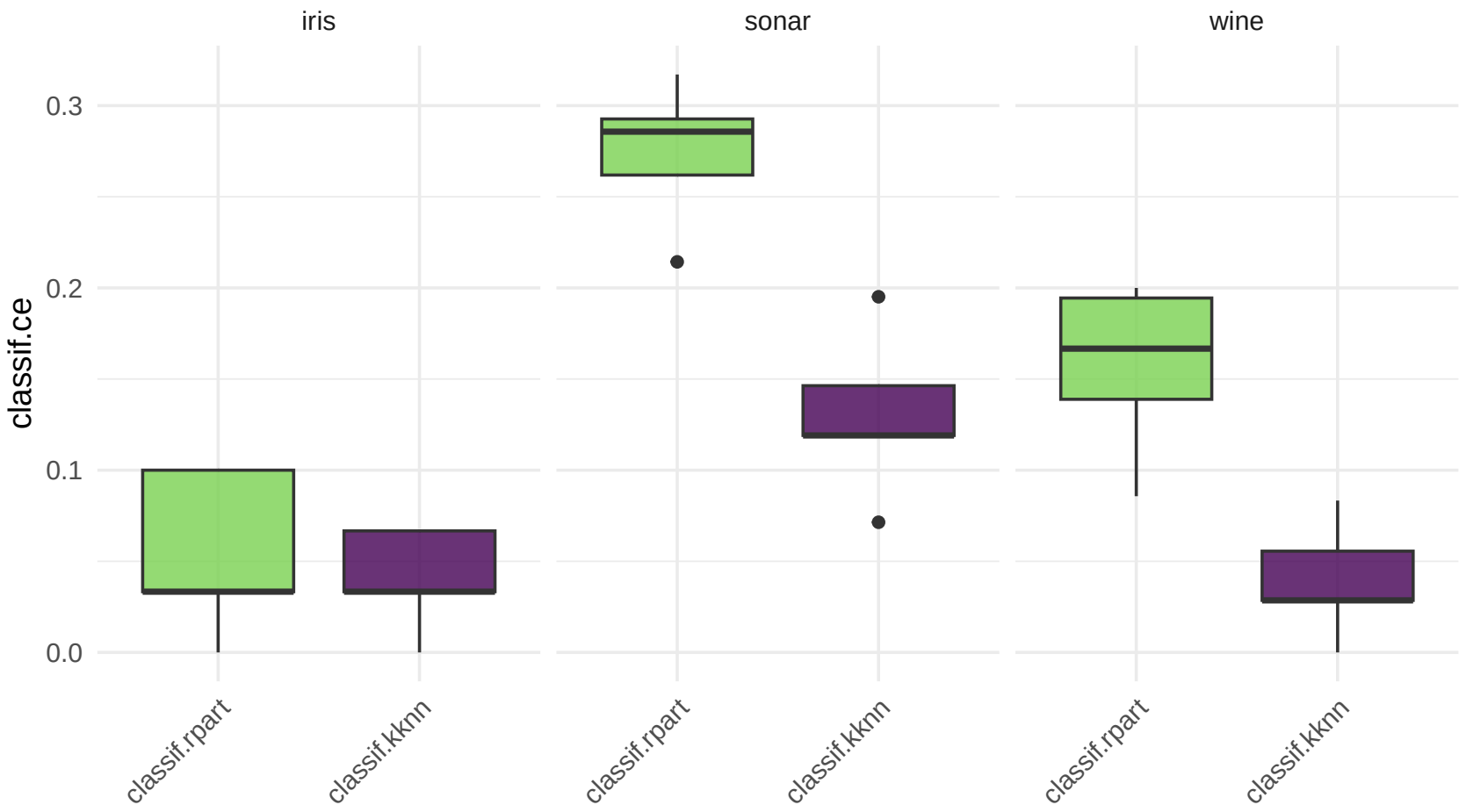
What exactly is a `BenchmarkResult` object?
Just like `Prediction` and `ResamplingResult`!

- Table representation using `as.data.table()`
- Active bindings and functions that make information easily accessible

BENCHMARK RESULT

The `mlr3viz` package contains `autoplot()` functions for many mlr3 objects

```
library(mlr3viz)
autoplot(bmr)
```



Resampling

Benchmark

Control of Execution

How to get Help

Outro

CONTROL OF EXECUTION

Parallelization

```
future::plan("multicore")
```

- runs each resampling iteration as a job
- also allows nested resampling (although not needed here)

Encapsulation

```
learner$encapsulate = c(train = "callr", predict = "callr")
```

- Spawns a separate R process to train the learner
- Learner may segfault without tearing down the session
- Logs are captured
- Possibilty to have a fallback to create predictions

Resampling

Benchmark

Control of Execution

How to get Help

Outro

HOW TO GET HELP

- Where to start?
  - Check these slides
  - **Check the mlr3book https://mlr3book.mlr-org.com**

HOW TO GET HELP

- Where to start?
  - Check these slides
  - **Check the mlr3book https://mlr3book.mlr-org.com**
- Get help for R6 objects?
  1. Find out what kind of R6 object you have:

     ```
     class(bmr)
     #> [1] "BenchmarkResult" "R6"
     ```

  2. Go to the corresponding help page:

     ```
     ?BenchmarkResult
     ```

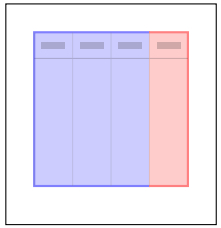
  New: open the corresponding man page with

  ```
  learner$help()
  ```

OVERVIEW

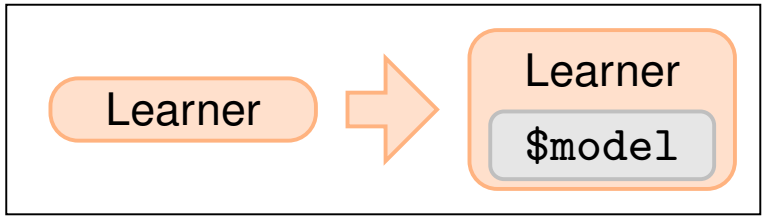
Ingredients:

Data



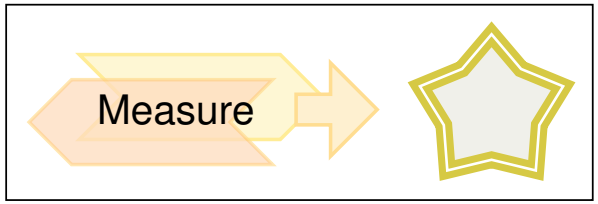
`TaskClassif`,
`TaskRegr`,
`tsk()`

Learning Algorithms



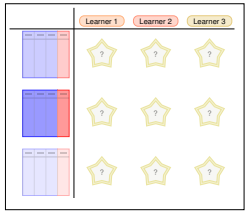
`lrn()` ⇒ `Learner`,
↪`Learner$train()`,
↪`Learner$predict()` ⇒ `Prediction`

Performance Evaluation



`rsmp()` ⇒ `Resampling`,
`msr()` ⇒ `Measure`,
`resample()` ⇒ `ResamplingResult`,
↪ `ResamplingResult$score()`,
↪ `ResamplingResult$aggregate()`

Performance Comparison



`benchmark_grid()`,
`benchmark()` ⇒ `BenchmarkResult`