



Applied Machine Learning

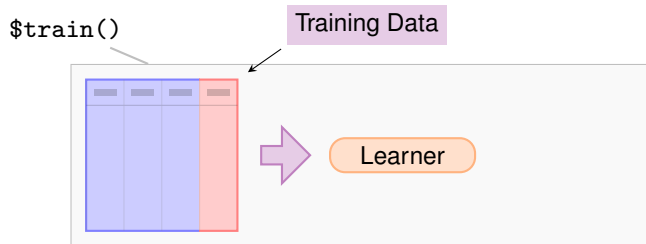
Machine Learning in R: MLR3 Learning & Evaluation



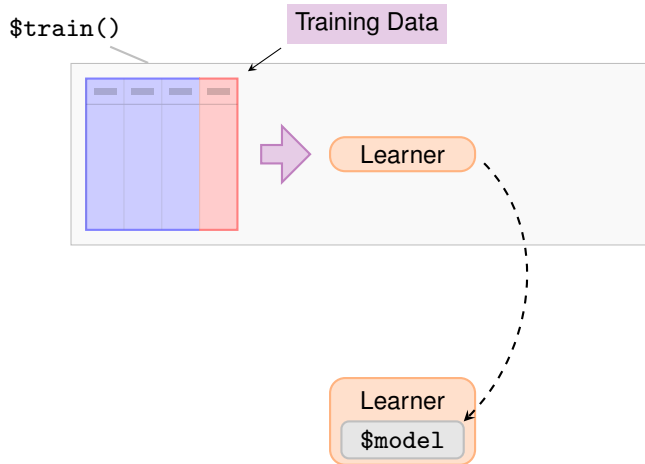
Learning goals

- Working with Learners in MLR3
- Train, Predict, Evaluate workflow
- Understanding model performance evaluation

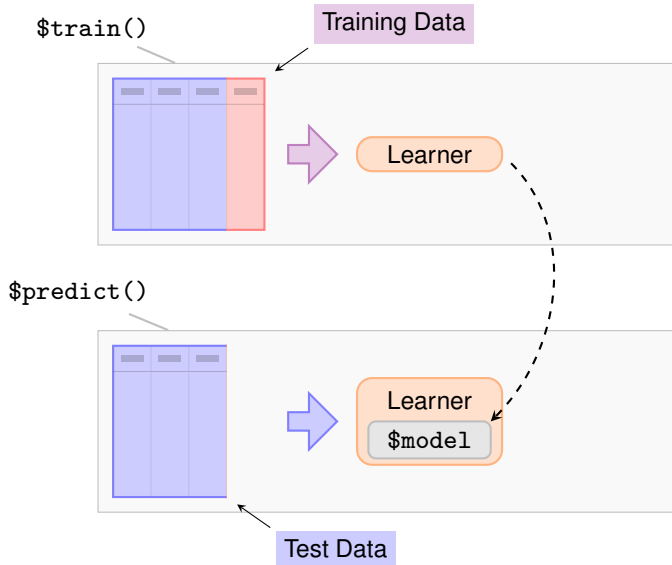
LEARNING ALGORITHMS



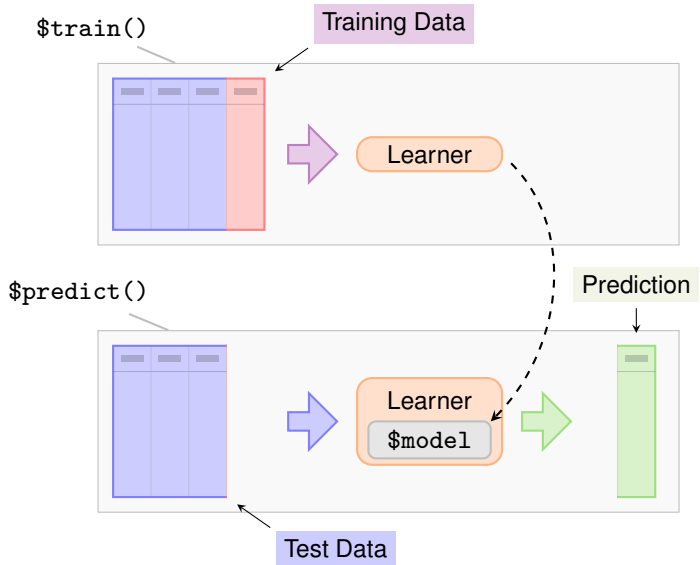
LEARNING ALGORITHMS



LEARNING ALGORITHMS



LEARNING ALGORITHMS



LEARNING ALGORITHMS

- Get a Learner provided by mlr

```
learner = lrn("classif.rpart")
```



LEARNING ALGORITHMS

- Get a Learner provided by mlr

```
learner = lrn("classif.rpart")
```

- Train the Learner

```
learner$train(task)
```



LEARNING ALGORITHMS

- Get a Learner provided by mlr

```
learner = lrn("classif.rpart")
```

- Train the Learner

```
learner$train(task)
```

- The \$model is the rpart model: a decision tree

```
print(learner$model)

#> n= 150
#>
#> node), split, n, loss, yval, (yprob)
#>      * denotes terminal node
#>
#> 1) root 150 100 setosa (0.333 0.333 0.333)
#>   2) Petal.Length< 2.4 50   0 setosa (1.000 0.000 0.000) *
#>   3) Petal.Length>=2.4 100  50 versicolor (0.000 0.500 0.500)
#>     6) Petal.Width< 1.8 54   5 versicolor (0.000 0.907 0.093) *
#>     7) Petal.Width>=1.8 46   1 virginica (0.000 0.022 0.978) *
```



HYPERPARAMETERS



- Learners have *hyperparameters* that control their behavior

```
as.data.table(learner$param_set)[, 1:6]
```

```
#>           id      class lower upper      levels nlevels
#>      <char>   <char> <num> <num>    <list>    <num>
#> 1:         cp ParamDbl      0     1              Inf
#> 2:   keep_model ParamLgl    NA    NA TRUE,FALSE      2
#> 3:   maxcompete ParamInt      0   Inf              Inf
#> 4:    maxdepth ParamInt      1   30              30
#> 5: maxsurrogate ParamInt      0   Inf              Inf
#> 6:    minbucket ParamInt      1   Inf              Inf
#> 7:    minsplit ParamInt      1   Inf              Inf
#> 8: surrogatestyle ParamInt      0     1              2
#> 9:   usesurrogate ParamInt      0     2              3
#> 10:          xval ParamInt      0   Inf              Inf
```

HYPERPARAMETERS



- Changing hyperparameters after the creation of a Learner object:

```
learner$param_set$values = list(maxdepth = 1, xval = 0)
```

- Changing hyperparameters when the Learner object is created:

```
learner = lrn("classif.rpart", maxdepth = 1, xval = 0)
```

HYPERPARAMETERS



- Changing hyperparameters after the creation of a Learner object:

```
learner$param_set$values = list(maxdepth = 1, xval = 0)
```

- Changing hyperparameters when the Learner object is created:

```
learner = lrn("classif.rpart", maxdepth = 1, xval = 0)
```

- The Learner behavior changes and gives a smaller decision tree:

```
learner$train(task)
print(learner$model)
#> n= 150
#>
#> node), split, n, loss, yval, (yprob)
#>      * denotes terminal node
#>
#> 1) root 150 100 setosa (0.33 0.33 0.33)
#>  2) Petal.Length< 2.4 50  0 setosa (1.00 0.00 0.00) *
#>  3) Petal.Length>=2.4 100 50 versicolor (0.00 0.50 0.50) *
```

PREDICTION

- Let's make a prediction for some new data, e.g.:

```
new_data
```

```
# Sepal.Length Sepal.Width Petal.Length Petal.Width  
# 1           4           3           2           1  
# 2           2           2           3           2
```



PREDICTION



- Let's make a prediction for some new data, e.g.:

```
new_data
#   Sepal.Length Sepal.Width Petal.Length Petal.Width
# 1           4           3           2           1
# 2           2           2           3           2
```

- To do so, we call the `$predict_newdata()` method using the new data:

```
prediction = learner$predict_newdata(new_data)
```

PREDICTION



- Let's make a prediction for some new data, e.g.:

```
new_data
#   Sepal.Length Sepal.Width Petal.Length Petal.Width
# 1             4           3           2           1
# 2             2           2           3           2
```

- To do so, we call the `$predict_newdata()` method using the new data:

```
prediction = learner$predict_newdata(new_data)
```

- We get a Prediction object:

```
prediction
#> <PredictionClassif> for 2 observations:
#>   row_ids truth  response
#>       1  <NA>   setosa
#>       2  <NA> versicolor
```

PREDICTION



- Let's make a prediction for some new data, e.g.:

```
new_data
# Sepal.Length Sepal.Width Petal.Length Petal.Width
# 1           4           3           2           1
# 2           2           2           3           2
```

- To do so, we call the `$predict_newdata()` method using the new data:

```
prediction = learner$predict_newdata(new_data)
```

- We get a Prediction object:

```
prediction
#> <PredictionClassif> for 2 observations:
#> row_ids truth response
#> { 1 <NA> setosa
#>   2 <NA> versicolor
```

PREDICTION



- We can make the Learner predict *probabilities* when we set `predict_type`:

```
learner$predict_type = "prob"
learner$predict_newdata(new_data)
# <PredictionClassif> for 2 observations:
#   row_ids truth  response prob.setosa prob.versicolor
#         1  <NA>   setosa          1             0.0
#         2  <NA> virginica          0             0.5
#   prob.virginica
#             0.0
#             0.5
```


PREDICTION

What exactly is a Prediction object?

- Contains predictions and offers useful access fields / methods



PREDICTION



What exactly is a Prediction object?

- Contains predictions and offers useful access fields / methods

⇒ Use `as.data.table()` to extract data

```
as.data.table(prediction)
#>   row_ids truth  response
#>   <int> <fctr>   <fctr>
#> 1:     1  <NA>    setosa
#> 2:     2  <NA> versicolor
```

PREDICTION



What exactly is a Prediction object?

- Contains predictions and offers useful access fields / methods

⇒ Use `as.data.table()` to extract data

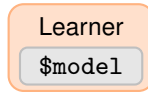
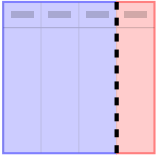
```
as.data.table(prediction)
#>   row_ids  truth  response
#>   <int> <fctr>   <fctr>
#> 1:     1   <NA>    setosa
#> 2:     2   <NA> versicolor
```

⇒ Active bindings and functions that give further information:

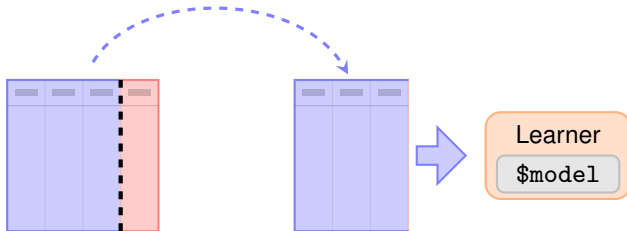
`$response`, `$truth`, ...

```
prediction$response
#> [1] setosa    versicolor
#> Levels: setosa versicolor virginica
```

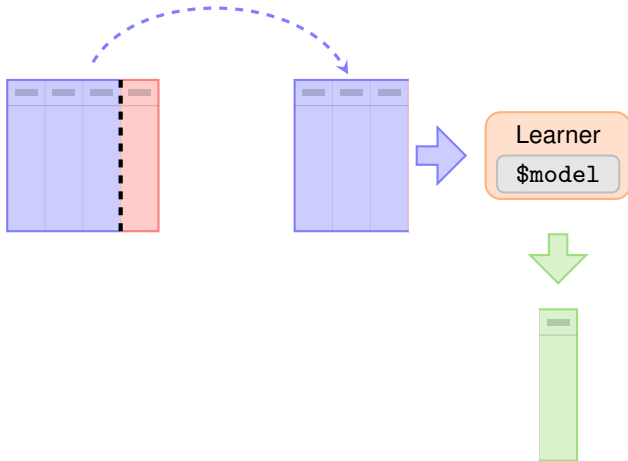
PERFORMANCE EVALUATION



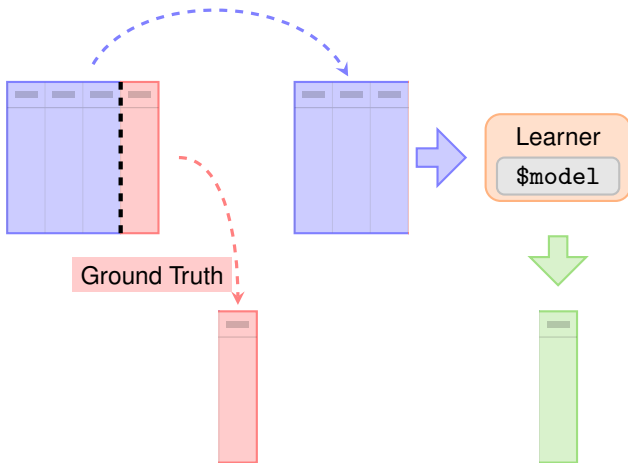
PERFORMANCE EVALUATION



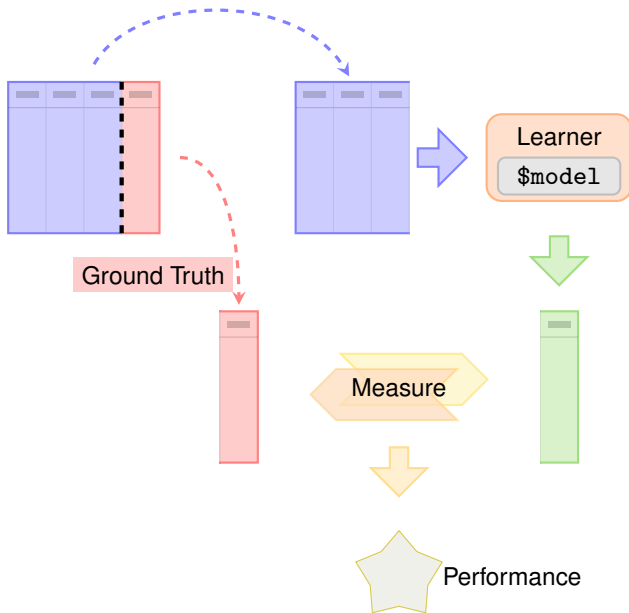
PERFORMANCE EVALUATION



PERFORMANCE EVALUATION



PERFORMANCE EVALUATION



PERFORMANCE EVALUATION

- Prediction 'Task' with known data

```
known_truth_task$data()
```

```
#   Species Petal.Length Petal.Width Sepal.Length Sepal.Width
#   <fctr>      <num>      <num>      <num>      <num>
# 1:  setosa         2         1         4         3
# 2:  setosa         3         2         2         2
```



PERFORMANCE EVALUATION

- Prediction 'Task' with known data

```
known_truth_task$data()
#   Species Petal.Length Petal.Width Sepal.Length Sepal.Width
#   <fctr>      <num>      <num>      <num>      <num>
# 1:  setosa         2         1         4         3
# 2:  setosa         3         2         2         2
```

- Predict again

```
pred = learner$predict(known_truth_task)
pred
#> <PredictionClassif> for 2 observations:
#>   row_ids truth  response
#>       1 setosa   setosa
#>       2 setosa virginica
```



PERFORMANCE EVALUATION



- Prediction 'Task' with known data

```
known_truth_task$data()
#   Species Petal.Length Petal.Width Sepal.Length Sepal.Width
#   <fctr>      <num>      <num>      <num>      <num>
# 1:  setosa         2         1         4         3
# 2:  setosa         3         2         2         2
```

- Predict again

```
pred = learner$predict(known_truth_task)
pred
#> <PredictionClassif> for 2 observations:
#>   row_ids truth  response
#>       1 setosa   setosa
#>       2 setosa  virginica
```

- Score the prediction

```
pred$score(msr("classif.ce"))
#> classif.ce
#>       0.5
```

PERFORMANCE EVALUATION



- Prediction 'Task' with known data

```
known_truth_task$data()
#   Species Petal.Length Petal.Width Sepal.Length Sepal.Width
#   <fctr>      <num>      <num>      <num>      <num>
# 1:  setosa         2         1         4         3
# 2:  setosa         3         2         2         2
```

- Predict again

```
pred = learner$predict(known_truth_task)
pred
#> <PredictionClassif> for 2 observations:
#>   row_ids truth  response
#>       1 setosa   setosa
#>       2 setosa  virginica
```

- Score the prediction

```
pred$score(msr("classif.ce"))
#> classif.ce
#>       0.5
```