Package 'mlr3learners'

September 17, 2019

Title Recommended Learners for 'mlr3'

Version 0.1.3

Description Recommended Learners for 'mlr3'. Extends 'mlr3' with interfaces to essential machine learning packages on CRAN. This includes, but is not limited to: (penalized) linear and logistic regression, linear and quadratic discriminant analysis, k-nearest neighbors, naive Bayes, support vector machines, and gradient boosting.

License LGPL-3

URL https://mlr3learners.mlr-org.com, https://github.com/mlr-org/mlr3learners

BugReports https://github.com/mlr-org/mlr3learners/issues

Depends R (>= 3.1.0)

Imports data.table, mlr3 (>= 0.1.2), mlr3misc, paradox, R6

Suggests checkmate, DiceKriging, e1071, glmnet, kknn, lgr, MASS, ranger, testthat, withr, xgboost

Encoding UTF-8

NeedsCompilation no

RoxygenNote 6.1.1

Author Michel Lang [cre, aut] (<https://orcid.org/0000-0001-9754-0393>), Quay Au [aut] (<https://orcid.org/0000-0002-5252-8902>), Stefan Coors [aut] (<https://orcid.org/0000-0002-7465-2146>), Patrick Schratz [aut] (<https://orcid.org/0000-0003-0748-6624>)

Maintainer Michel Lang <michellang@gmail.com>

Repository CRAN

Date/Publication 2019-09-17 16:00:02 UTC

R topics documented:

`mlr3learners-package` *mlr3learners: Recommended Learners for 'mlr3'*

Description

Recommended Learners for 'mlr3'. Extends 'mlr3' with interfaces to essential machine learning packages on CRAN. This includes, but is not limited to: (penalized) linear and logistic regression, linear and quadratic discriminant analysis, k-nearest neighbors, naive Bayes, support vector machines, and gradient boosting.

Author(s)

Maintainer: Michel Lang <michellang@gmail.com> (0000-0001-9754-0393)

Authors:

- Quay Au <quayau@gmail.com> (0000-0002-5252-8902)
- Stefan Coors <mail@stefancoors.de> (0000-0002-7465-2146)
- Patrick Schratz <patrick.schratz@gmail.com> (0000-0003-0748-6624)

See Also

Useful links:

- https://mlr3learners.mlr-org.com
- https://github.com/mlr-org/mlr3learners
- Report bugs at https://github.com/mlr-org/mlr3learners/issues

LearnerClassifGlmnet *GLM with Elastic Net Regularization Classification Learner*

Description

Generalized linear models with elastic net regularization. Calls `glmnet::cv.glmnet()` from package **glmnet**.

Format

`R6::R6Class()` inheriting from mlr3::LearnerClassif.

Construction

```
LearnerClassifGlmnet$new()
mlr3::mlr_learners$get("classif.glmnet")
mlr3::lrn("classif.glmnet")
```

References

Jerome Friedman, Trevor Hastie, Robert Tibshirani (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software, 33(1), 1-22. doi: 10.18637/jss.v033.i01.

See Also

Dictionary of Learners: mlr3::mlr_learners

Examples

```
learner = mlr3::lrn("classif.glmnet")
print(learner)

# available parameters:
learner$param_set$ids()
```

LearnerClassifKKNN *k-Nearest-Neighbor Classification Learner*

Description

k-Nearest-Neighbor classification. Calls `kknn::kknn()` from package **kknn**.

Format

`R6::R6Class()` inheriting from mlr3::LearnerClassif.

Construction

```
LearnerClassifKKNN$new()
mlr3::mlr_learners$get("classif.kknn")
mlr3::lrn("classif.kknn")
```

References

Klaus Hechenbichler, Klaus Schliep (2004). Weighted k-nearest-neighbor techniques and ordinal classification. Discussion Paper 399, SFB 386, Ludwig-Maximilians University Munich doi: 10.5282/ubm/epub.1769

Thomas Cover and Peter Hart (1967). Nearest neighbor pattern classification. IEEE transactions on information theory 13.1: 21-27. doi: 10.1109/TIT.1967.1053964

See Also

Dictionary of Learners: mlr3::mlr_learners

Examples

```
learner = mlr3::lrn("classif.kknn")
print(learner)

# available parameters:
learner$param_set$ids()
```

LearnerClassifLDA *Linear Discriminant Analysis Classification Learner*

Description

Linear discriminant analysis. Calls `MASS::lda()` from package **MASS**.

Format

`R6::R6Class()` inheriting from mlr3::LearnerClassif.

Construction

```
LearnerClassifLDA$new()
mlr3::mlr_learners$get("classif.lda")
mlr3::lrn("classif.lda")
```

References

William N. Venables, Brian D. Ripley (2002). Modern Applied Statistics with S. Fourth Edition. Springer, New York. ISBN 0-387-95457-0. doi: 10.1007/9780387217062.

See Also

Dictionary of Learners: mlr3::mlr_learners

Examples

```
learner = mlr3::lrn("classif.lda")
print(learner)

# available parameters:
learner$param_set$ids()
```

`LearnerClassifLogReg` *Logistic Regression Classification Learner*

Description

Classification via logistic regression. Calls `stats::glm()`.

Format

`R6::R6Class()` inheriting from mlr3::LearnerClassif.

Construction

```
LearnerClassifLogReg$new()
mlr3::mlr_learners$get("classif.log_reg")
mlr3::lrn("classif.log_reg")
```

See Also

Dictionary of Learners: mlr3::mlr_learners

Examples

```
learner = mlr3::lrn("classif.log_reg")
print(learner)

# available parameters:
learner$param_set$ids()
```

LearnerClassifNaiveBayes *Naive Bayes Classification Learner*

Description

Naive Bayes classification. Calls `e1071::naiveBayes()` from package **e1071**.

Format

`R6::R6Class()` inheriting from mlr3::LearnerClassif.

Construction

```
LearnerClassifNaiveBayes$new()
mlr3::mlr_learners$get("classif.naive_bayes")
mlr3::lrn("classif.naive_bayes")
```

See Also

Dictionary of Learners: mlr3::mlr_learners

Examples

```
learner = mlr3::lrn("classif.naive_bayes")
print(learner)

# available parameters:
learner$param_set$ids()
```

LearnerClassifQDA *Quadratic Discriminant Analysis Classification Learner*

Description

Quadratic discriminant analysis. Calls `MASS::qda()` from package **MASS**.

Format

`R6::R6Class()` inheriting from mlr3::LearnerClassif.

Construction

```
LearnerClassifQDA$new()
mlr3::mlr_learners$get("classif.qda")
mlr3::lrn("classif.qda")
```

References

William N. Venables, Brian D. Ripley (2002). Modern Applied Statistics with S. Fourth Edition. Springer, New York. ISBN 0-387-95457-0. doi: 10.1007/9780387217062.

See Also

Dictionary of Learners: mlr3::mlr_learners

Examples

```
learner = mlr3::lrn("classif.qda")
print(learner)

# available parameters:
learner$param_set$ids()
```

`LearnerClassifRanger` *Ranger Classification Learner*

Description

Random classification forest. Calls `ranger::ranger()` from package **ranger**.

Format

`R6::R6Class()` inheriting from mlr3::LearnerClassif.

Construction

```
LearnerClassifRanger$new()
mlr3::mlr_learners$get("classif.ranger")
mlr3::lrn("classif.ranger")
```

References

Marvin N. Wright, Andreas Ziegler (2017). ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R. Journal of Statistical Software, 77(1), 1-17. doi: 10.18637/jss.v077.i01.

Breiman, L. (2001). Random Forests. Machine Learning 45(1). doi: 10.1023/A:1010933404324.

See Also

Dictionary of Learners: mlr3::mlr_learners

Examples

```
learner = mlr3::lrn("classif.ranger")
print(learner)

# available parameters:
learner$param_set$ids()
```

`LearnerClassifSVM` *Support Vector Machine*

Description

A learner for a classification support vector machine implemented in `e1071::svm()`.

Format

`R6::R6Class()` inheriting from mlr3::LearnerClassif.

Construction

```
LearnerClassifSVM$new()
mlr3::mlr_learners$get("classif.svm")
mlr3::lrn("classif.svm")
```

References

Corinna Cortes, Vladimir Vapnik (1995). Machine Learning 20: 273. doi: 10.1007/BF00994018.

See Also

Dictionary of Learners: mlr3::mlr_learners

Examples

```
learner = mlr3::lrn("classif.svm")
print(learner)

# available parameters:
learner$param_set$ids()
```

LearnerClassifXgboost *Extreme Gradiant Boosting Classification Learner*

Description

eXtreme Gradient Boosting classification. Calls `xgboost::xgb.train()` from package **xgboost**.

Format

`R6::R6Class()` inheriting from mlr3::LearnerClassif.

Construction

```
LearnerClassifXgboost$new()
mlr3::mlr_learners$get("classif.xgboost")
mlr3::lrn("classif.xgboost")
```

References

Tianqi Chen, Carlos Guestrin (2016). XGBoost: A Scalable Tree Boosting System. 22nd SIGKDD Conference on Knowledge Discovery and Data Mining. doi: 10.1145/2939672.2939785.

See Also

Dictionary of Learners: mlr3::mlr_learners

Examples

```
learner = mlr3::lrn("classif.xgboost")
print(learner)

# available parameters:
learner$param_set$ids()
```

LearnerRegrGlmnet *GLM with Elastic Net Regularization Regression Learner*

Description

Generalized linear models with elastic net regularization. Calls `glmnet::cv.glmnet()` from package **glmnet**. Hyperparameter `family` is set to `"gaussian"`.

Format

`R6::R6Class()` inheriting from mlr3::LearnerRegr.

Construction

```
LearnerRegrGlmnet$new()
mlr3::mlr_learners$get("regr.glmnet")
mlr3::lrn("regr.glmnet")
```

References

Jerome Friedman, Trevor Hastie, Robert Tibshirani (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software, 33(1), 1-22. doi: 10.18637/jss.v033.i01.

See Also

Dictionary of Learners: mlr3::mlr_learners

Examples

```
learner = mlr3::lrn("regr.glmnet")
print(learner)

# available parameters:
learner$param_set$ids()
```

LearnerRegrKKNN *k-Nearest-Neighbor Regression Learner*

Description

k-Nearest-Neighbor regression. Calls `kknn::kknn()` from package **kknn**.

Format

`R6::R6Class()` inheriting from mlr3::LearnerRegr.

Construction

```
LearnerRegrKKNN$new()
mlr3::mlr_learners$get("regr.kknn")
mlr3::lrn("regr.kknn")
```

References

Klaus Hechenbichler, Klaus Schliep (2004). Weighted k-nearest-neighbor techniques and ordinal classification. Discussion Paper 399, SFB 386, Ludwig-Maximilians University Munich doi: 10.5282/ubm/epub.1769

Thomas Cover and Peter Hart (1967). Nearest neighbor pattern classification. IEEE transactions on information theory 13.1: 21-27. doi: 10.1109/TIT.1967.1053964

See Also

Dictionary of Learners: mlr3::mlr_learners

Examples

```
learner = mlr3::lrn("regr.kknn")
print(learner)

# available parameters:
learner$param_set$ids()
```

`LearnerRegrKM` *Kriging Regression Learner*

Description

Kriging regression. Calls `DiceKriging::km()` from package **DiceKriging**.

- The predict type hyperparameter "type" defaults to "SK" (simple Kriging).
- The additional hyperparameter `nugget.stability` is used to overwrite the hyperparameter `nugget` with `nugget.stability * var(y)` before training to improve the numerical stability. We recommend a value of `1e-8`.
- The additional hyperparameter `jitter` can be set to add `N(0,[jitter])`-distributed noise to the data before prediction to avoid perfect interpolation. We recommend a value of `1e-12`.

Format

`R6::R6Class()` inheriting from mlr3::LearnerRegr.

Construction

```
LearnerRegrKM$new()
mlr3::mlr_learners$get("regr.km")
mlr3::lrn("regr.km")
```

References

Olivier Roustant, David Ginsbourger, Yves Deville (2012). DiceKriging, DiceOptim: Two R Packages for the Analysis of Computer Experiments by Kriging-Based Metamodeling and Optimization. Journal of Statistical Software, 51(1), 1-55. doi: 10.18637/jss.v051.i01.

See Also

Dictionary of Learners: mlr3::mlr_learners

Examples

```
learner = mlr3::lrn("regr.km")
print(learner)

# available parameters:
learner$param_set$ids()
```

LearnerRegrLM *Linear Model Regression Learner*

Description

Ordinary linear regression. Calls `stats::lm()`.

Format

`R6::R6Class()` inheriting from mlr3::LearnerRegr.

Construction

```
LearnerRegrLM$new()
mlr3::mlr_learners$get("regr.lm")
mlr3::lrn("regr.lm")
```

See Also

Dictionary of Learners: mlr3::mlr_learners

Examples

```
learner = mlr3::lrn("regr.lm")
print(learner)

# available parameters:
learner$param_set$ids()
```

LearnerRegrRanger *Ranger Regression Learner*

Description

Random regression forest. Calls `ranger::ranger()` from package **ranger**.

Format

`R6::R6Class()` inheriting from mlr3::LearnerClassif.

Construction

```
LearnerRegrRanger$new()
mlr3::mlr_learners$get("regr.ranger")
mlr3::lrn("regr.ranger")
```

References

Marvin N. Wright, Andreas Ziegler (2017). ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R. Journal of Statistical Software, 77(1), 1-17. doi: 10.18637/jss.v077.i01.

Breiman, L. (2001). Random Forests. Machine Learning 45(1). doi: 10.1023/A:1010933404324.

See Also

Dictionary of Learners: mlr3::mlr_learners

Examples

```
learner = mlr3::lrn("regr.ranger")
print(learner)

# available parameters:
learner$param_set$ids()
```

LearnerRegrSVM *Support Vector Machine*

Description

A learner for a regression support vector machine implemented in `e1071::svm()`.

Format

`R6::R6Class()` inheriting from mlr3::LearnerRegr.

Construction

```
LearnerRegrSVM$new()
mlr3::mlr_learners$get("regr.svm")
mlr3::lrn("regr.svm")
```

References

Corinna Cortes, Vladimir Vapnik (1995). Machine Learning 20: 273. doi: 10.1007/BF00994018.

See Also

Dictionary of Learners: mlr3::mlr_learners

Examples

```
learner = mlr3::lrn("regr.svm")
print(learner)

# available parameters:
learner$param_set$ids()
```

`LearnerRegrXgboost` *Extreme Gradient Boosting Regression Learner*

Description

eXtreme Gradient Boosting regression. Calls `xgboost::xgb.train()` from package **xgboost**.

Format

`R6::R6Class()` inheriting from mlr3::LearnerRegr.

Construction

```
LearnerRegrXgboost$new()
mlr3::mlr_learners$get("regr.xgboost")
mlr3::lrn("regr.xgboost")
```

References

Tianqi Chen, Carlos Guestrin (2016). XGBoost: A Scalable Tree Boosting System. 22nd SIGKDD Conference on Knowledge Discovery and Data Mining. doi: 10.1145/2939672.2939785.

See Also

Dictionary of Learners: mlr3::mlr_learners

Examples

```
learner = mlr3::lrn("regr.xgboost")
print(learner)

# available parameters:
learner$param_set$ids()
```

Index