# Package 'orf'

May 9, 2026

**Type** Package

**Title** Ordered Random Forests

**Version** 0.1.4

**Date** 2022-07-21

**Author** Gabriel Okasa [aut, cre], Michael Lechner [ctb]

**Maintainer** Gabriel Okasa <okasa.gabriel@gmail.com>

**Description** An implementation of the Ordered Forest estimator as developed in Lechner & Okasa (2019) <doi:10.48550/arXiv.1907.02436>. The Ordered Forest flexibly estimates the conditional probabilities of models with ordered categorical outcomes (so-called ordered choice models). Additionally to common machine learning algorithms the 'orf' package provides functions for estimating marginal effects as well as statistical inference thereof and thus provides similar output as in standard econometric models for ordered choice. The core forest algorithm relies on the fast C++ forest implementation from the 'ranger' package (Wright & Ziegler, 2017) <doi:10.48550/arXiv.1508.04409>.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 2.10)

**Imports** ggplot2, ranger, Rcpp, stats, utils, xtable

**RoxygenNote** 7.2.1

**Suggests** knitr, rmarkdown, testthat

**VignetteBuilder** knitr

**URL** https://github.com/okasag/orf

**BugReports** https://github.com/okasag/orf/issues

**LinkingTo** Rcpp

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2022-07-23 22:40:02 UTC

# Contents



---

`orf-package` *orf: Ordered Random Forests*

---

## Description

An implementation of the Ordered Forest estimator as developed in Lechner & Okasa (2019). The Ordered Forest flexibly estimates the conditional probabilities of models with ordered categorical outcomes (so-called ordered choice models). Additionally to common machine learning algorithms the `orf` package provides functions for estimating marginal effects as well as statistical inference thereof and thus provides similar output as in standard econometric models for ordered choice. The core forest algorithm relies on the fast C++ forest implementation from the `ranger` package (Wright & Ziegler, 2017).

## Author(s)

Gabriel Okasa, Michael Lechner

## References

- Lechner, M., & Okasa, G. (2019). Random Forest Estimation of the Ordered Choice Model. arXiv preprint arXiv:1907.02436. https://arxiv.org/abs/1907.02436
- Goller, D., Knaus, M. C., Lechner, M., & Okasa, G. (2021). Predicting Match Outcomes in Football by an Ordered Forest Estimator. A Modern Guide to Sports Economics. Edward Elgar Publishing, 335-355. doi:10.4337/9781789906530.00026
- Wright, M. N. & Ziegler, A. (2017). ranger: A fast implementation of random forests for high dimensional data in C++ and R. J Stat Softw 77:1-17. doi:10.18637/jss.v077.i01.

## Examples

```
## Ordered Forest
require(orf)

# load example data
data(odata)

# specify response and covariates
Y <- as.numeric(odata[, 1])
X <- as.matrix(odata[, -1])

# estimate Ordered Forest with default settings
orf_fit <- orf(X, Y)

# print output of the orf estimation
print(orf_fit)

# show summary of the orf estimation
summary(orf_fit)

# plot the estimated probability distributions
plot(orf_fit)

# predict with the estimated orf
predict(orf_fit)

# estimate marginal effects of the orf
margins(orf_fit)
```

---

`margins` *Marginal Effects*

---

## Description

S3 generic method for estimation of marginal effects of an Ordered Forest objects of class `orf`.

## Usage

```
margins(forest, eval = NULL, inference = NULL, window = NULL, newdata = NULL)
```

## Arguments

| | |
|---|---|
| `forest` | estimated Ordered Forest object of class `orf` |
| `eval` | string, defining evaluation point for marginal effects. These can be one of "mean", "atmean", or "atmedian". (Default is "mean") |
| `inference` | logical, if TRUE inference on marginal effects will be conducted (default is inherited from the `orf` object) |

| | |
|---|---|
| window | numeric, share of standard deviation of X to be used for evaluation of the marginal effect (default is 0.1) |
| newdata | numeric matrix X containing the new observations for which the marginal effects should be estimated |

## Author(s)

Gabriel Okasa

## See Also

margins.orf, summary.margins.orf and print.margins.orf

## Examples

```
## Ordered Forest
require(orf)

# load example data
data(odata)

# specify response and covariates
Y <- as.numeric(odata[, 1])
X <- as.matrix(odata[, -1])

# estimate Ordered Forest
orf_fit <- orf(X, Y)

# estimate default marginal effects of the orf
orf_margins <- margins(orf_fit)
```

---

# margins.orf *Marginal Effects for the Ordered Forest*

---

## Description

S3 method for estimation of marginal effects of an Ordered Forest objects of class orf.

## Usage

```
## S3 method for class 'orf'
margins(forest, eval = NULL, inference = NULL, window = NULL, newdata = NULL)
```

## Arguments

| | |
|---|---|
| `forest` | estimated Ordered Forest object of class `orf` |
| `eval` | string, defining evaluation point for marginal effects. These can be one of "mean", "atmean", or "atmedian". (Default is "mean") |
| `inference` | logical, if TRUE inference on marginal effects will be conducted (default is inherited from the `orf` object) |
| `window` | numeric, share of standard deviation of X to be used for evaluation of the marginal effect (default is 0.1) |
| `newdata` | numeric matrix X containing the new observations for which the marginal effects should be estimated |

## Details

`margins.orf` estimates marginal effects at the mean, at the median, or the mean marginal effects, depending on the `eval` argument. It is advised to increase the number of subsampling replications in the supplied `orf` object as the estimation of the marginal effects is a more demanding exercise than a simple Ordered Forest estimation/prediction. Additionally to the estimation of the marginal effects, the weight-based inference for the effects is supported as well. Note, that the inference procedure is much more computationally exhausting exercise due to the computation of the forest weights. Additionally, the evaluation window for the marginal effects can be regulated through the `window` argument. Furthermore, new data for which marginal effects should be computed can be supplied as well as long as it lies within the support of `X`.

## Value

object of type `margins.orf` with following elements

| | |
|---|---|
| `info` | info containing forest inputs and data used |
| `effects` | marginal effects |
| `variances` | variances of marginal effects |
| `errors` | standard errors of marginal effects |
| `tvalues` | t-values of marginal effects |
| `pvalues` | p-values of marginal effects |

## Author(s)

Gabriel Okasa

## See Also

`summary.margins.orf`, `print.margins.orf`

## Examples

```
## Ordered Forest
require(orf)

# load example data
data(odata)

# specify response and covariates
Y <- as.numeric(odata[, 1])
X <- as.matrix(odata[, -1])

# estimate Ordered Forest
orf_fit <- orf(X, Y)

# estimate marginal effects of the orf (default)
orf_margins <- margins(orf_fit)

# estimate marginal effects evaluated at the mean
orf_margins <- margins(orf_fit, eval = "atmean")

# estimate marginal effects with inference
# (orf object has to be estimated with honesty and subsampling)
orf_margins <- margins(orf_fit, inference = TRUE)

# estimate marginal effects with custom window size
orf_margins <- margins(orf_fit, window = 0.5)

# estimate marginal effects for some new data (within support of X)
orf_margins <- margins(orf_fit, newdata = X[1:10, ])

# estimate marginal effects with all custom settings
orf_margins <- margins(orf_fit, eval = "atmedian", inference = TRUE,
                               window = 0.5, newdata = X[1:10, ])
```

---

# odata *Simulated Example Dataset*

---

## Description

A simulated example dataset with ordered categorical outcome variable containing different types of covariates for illustration purposes.

## Usage

```
odata
```

## Format

A data frame with 1000 rows and 5 variables

## Details

For the exact data generating process, see the example below.

## Value

| | |
|---|---|
| `Y` | ordered outcome, classes 1, 2, and 3 |
| `X1` | continuous covariate, N(0,1) |
| `X2` | categorical covariate, values 1, 2, and 3 |
| `X3` | binary covariate, values 0 and 1 |
| `X4` | continuous covariate, N(0,10) |

## Examples

```
# generate example data

# set seed for replicability
set.seed(123)

# number of observations
n  <- 1000

# various covariates
X1 <- rnorm(n, 0, 1)    # continuous
X2 <- rbinom(n, 2, 0.5) # categorical
X3 <- rbinom(n, 1, 0.5) # dummy
X4 <- rnorm(n, 0, 10)   # noise

# bind into matrix
X <- as.matrix(cbind(X1, X2, X3, X4))

# deterministic component
deterministic <- X1 + X2 + X3
# generate continuous outcome with logistic error
Y <- deterministic + rlogis(n, 0, 1)
# thresholds for continuous outcome
cuts <- quantile(Y, c(0, 1/3, 2/3, 1))
# discretize outcome into ordered classes 1, 2, 3
Y <- as.numeric(cut(Y, breaks = cuts, include.lowest = TRUE))

# save data as a dataframe
odata <- as.data.frame(cbind(Y, X))

# end of data generating
```

orf *Ordered Forest Estimator*

## Description

An implementation of the Ordered Forest estimator as developed in Lechner & Okasa (2019). The Ordered Forest flexibly estimates the conditional probabilities of models with ordered categorical outcomes (so-called ordered choice models). Additionally to common machine learning algorithms the `orf` package provides functions for estimating marginal effects as well as statistical inference thereof and thus provides similar output as in standard econometric models for ordered choice. The core forest algorithm relies on the fast C++ forest implementation from the `ranger` package (Wright & Ziegler, 2017).

## Usage

```
orf(
  X,
  Y,
  num.trees = 1000,
  mtry = NULL,
  min.node.size = NULL,
  replace = FALSE,
  sample.fraction = NULL,
  honesty = TRUE,
  honesty.fraction = NULL,
  inference = FALSE,
  importance = FALSE
)
```

## Arguments

| | |
|---|---|
| `X` | numeric matrix of features |
| `Y` | numeric vector of outcomes |
| `num.trees` | scalar, number of trees in a forest, i.e. bootstrap replications (default is 1000 trees) |
| `mtry` | scalar, number of randomly selected features (default is the squared root of number of features, rounded up to the nearest integer) |
| `min.node.size` | scalar, minimum node size, i.e. leaf size of a tree (default is 5 observations) |
| `replace` | logical, if TRUE sampling with replacement, i.e. bootstrap is used to grow the trees, otherwise subsampling without replacement is used (default is set to FALSE) |
| `sample.fraction` | scalar, subsampling rate (default is 1 for bootstrap and 0.5 for subsampling) |
| `honesty` | logical, if TRUE honest forest is built using sample splitting (default is set to TRUE) |

honesty.fraction
: scalar, share of observations belonging to honest sample not used for growing the forest (default is 0.5)

inference
: logical, if TRUE the weight based inference is conducted (default is set to FALSE)

importance
: logical, if TRUE variable importance measure based on permutation is conducted (default is set to FALSE)

## Details

The Ordered Forest function, orf, estimates the conditional ordered choice probabilities, i.e. P[Y=m|X=x]. Additionally, weight-based inference for the probability predictions can be conducted as well. If inference is desired, the Ordered Forest must be estimated with honesty and subsampling. If prediction only is desired, estimation without honesty and with bootstrapping is recommended for optimal prediction performance.

In order to estimate the Ordered Forest user must supply the data in form of matrix of covariates X and a vector of outcomes 'codeY to the orf function. These data inputs are also the only inputs that must be specified by the user without any defaults. Further optional arguments include the classical forest hyperparameters such as number of trees, num.trees, number of randomly selected features, mtry, and the minimum leaf size, min.node.size. The forest building scheme is regulated by the replace argument, meaning bootstrapping if replace = TRUE or subsampling if replace = FALSE. For the case of subsampling, sample.fraction argument regulates the subsampling rate. Further, honest forest is estimated if the honesty argument is set to TRUE, which is also the default. Similarly, the fraction of the sample used for the honest estimation is regulated by the honesty.fraction argument. The default setting conducts a 50:50 sample split, which is also generally advised to follow for optimal performance. Inference procedure of the Ordered Forest is based on the forest weights and is controlled by the inference argument. Note, that such weight-based inference is computationally demanding exercise due to the estimation of the forest weights and as such longer computation time is to be expected. Lastly, the importance argument turns on and off the permutation based variable importance.

orf is compatible with standard R commands such as predict, margins, plot, summary and print. For further details, see examples below.

## Value

object of type orf with following elements

forests
: saved forests trained for orf estimations (inherited from ranger)

info
: info containing forest inputs and data used

predictions
: predicted values for class probabilities

variances
: variances of predicted values

importance
: weighted measure of permutation based variable importance

accuracy
: oob measures for mean squared error and ranked probability score

## Author(s)

Gabriel Okasa

## References


- Lechner, M., & Okasa, G. (2019). Random Forest Estimation of the Ordered Choice Model. arXiv preprint arXiv:1907.02436. https://arxiv.org/abs/1907.02436
- Goller, D., Knaus, M. C., Lechner, M., & Okasa, G. (2021). Predicting Match Outcomes in Football by an Ordered Forest Estimator. A Modern Guide to Sports Economics. Edward Elgar Publishing, 335-355. doi:10.4337/9781789906530.00026
- Wright, M. N. & Ziegler, A. (2017). ranger: A fast implementation of random forests for high dimensional data in C++ and R. J Stat Softw 77:1-17. doi:10.18637/jss.v077.i01.


## See Also

summary.orf, plot.orf predict.orf, margins.orf

## Examples

```
## Ordered Forest
require(orf)

# load example data
data(odata)

# specify response and covariates
Y <- as.numeric(odata[, 1])
X <- as.matrix(odata[, -1])

# estimate Ordered Forest with default parameters
orf_fit <- orf(X, Y)

# estimate Ordered Forest with own tuning parameters
orf_fit <- orf(X, Y, num.trees = 2000, mtry = 3, min.node.size = 10)

# estimate Ordered Forest with bootstrapping and without honesty
orf_fit <- orf(X, Y, replace = TRUE, honesty = FALSE)

# estimate Ordered Forest with subsampling and with honesty
orf_fit <- orf(X, Y, replace = FALSE, honesty = TRUE)

# estimate Ordered Forest with subsampling and with honesty
# with own tuning for subsample fraction and honesty fraction
orf_fit <- orf(X, Y, replace = FALSE, sample.fraction = 0.5,
                     honesty = TRUE, honesty.fraction = 0.5)

# estimate Ordered Forest with subsampling and with honesty and with inference
# (for inference, subsampling and honesty are required)
orf_fit <- orf(X, Y, replace = FALSE, honesty = TRUE, inference = TRUE)

# estimate Ordered Forest with simple variable importance measure
orf_fit <- orf(X, Y, importance = TRUE)

# estimate Ordered Forest with all custom settings
orf_fit <- orf(X, Y, num.trees = 2000, mtry = 3, min.node.size = 10,
```

```
                    replace = TRUE, sample.fraction = 1,
                    honesty = FALSE, honesty.fraction = 0,
                    inference = FALSE, importance = FALSE)
```

---

plot.orf *Plot of the Ordered Forest*

---

## Description

plot the probability distributions estimated by the Ordered Forest object of class orf

## Usage

```
## S3 method for class 'orf'
plot(x, ...)
```

## Arguments

| | |
|---|---|
| x | estimated Ordered Forest object of class orf |
| ... | further arguments (currently ignored) |

## Details

plot.orf generates probability distributions, i.e. density plots of estimated ordered probabilities by the Ordered Forest for each outcome class considered. The plots effectively visualize the estimated probability density in contrast to a real observed ordered outcome class and as such provide a visual inspection of the overall in-sample estimation accuracy. The dashed lines locate the means of the respective probability distributions.

## Author(s)

Gabriel Okasa

## Examples

```
# Ordered Forest
require(orf)

# load example data
data(odata)

# specify response and covariates
Y <- as.numeric(odata[, 1])
X <- as.matrix(odata[, -1])

# estimate Ordered Forest
orf_fit <- orf(X, Y)
```

```
# plot the estimated probability distributions
plot(orf_fit)
```

---

`predict.orf` *Prediction of the Ordered Forest*

---

## Description

Prediction for new observations based on estimated Ordered Forest of class `orf`

## Usage

```
## S3 method for class 'orf'
predict(object, newdata = NULL, type = NULL, inference = NULL, ...)
```

## Arguments

| | |
|---|---|
| `object` | estimated Ordered Forest object of class `orf` |
| `newdata` | numeric matrix X containing the observations for which the outcomes should be predicted |
| `type` | string, specifying the type of the prediction, These can be either "probs" or "p" for probabilities and "class" or "c" for classes. (Default is "probs"). |
| `inference` | logical, if TRUE variances for the predictions will be estimated (only feasible for probability predictions). |
| `...` | further arguments (currently ignored) |

## Details

`predict.orf` estimates the conditional ordered choice probabilities, i.e. P[Y=m|X=x] for new data points (matrix X containing new observations of covariates) based on the estimated Ordered Forest object of class `orf`. Furthermore, weight-based inference for the probability predictions can be conducted as well. If inference is desired, the supplied Ordered Forest must be estimated with honesty and subsampling. If prediction only is desired, estimation without honesty and with bootstrapping is recommended for optimal prediction performance. Additionally to the probability predictions, class predictions can be estimated as well using the `type` argument. In this case, the predicted classes are obtained as classes with the highest predicted probability.

## Value

object of class `orf.prediction` with following elements

| | |
|---|---|
| `info` | info containing forest inputs and data used |
| `predictions` | predicted values |
| `variances` | variances of predicted values |

## Author(s)

Gabriel Okasa

## See Also

summary.orf.prediction, print.orf.prediction

## Examples

```
# Ordered Forest
require(orf)

# load example data
data(odata)

# specify response and covariates for train and test
idx <- sample(seq(1, nrow(odata), 1), 0.8*nrow(odata))

# train set
Y_train <- as.numeric(odata[idx, 1])
X_train <- as.matrix(odata[idx, -1])

# test set
Y_test <- as.numeric(odata[-idx, 1])
X_test <- as.matrix(odata[-idx, -1])

# estimate Ordered Forest
orf_fit <- orf(X_train, Y_train)

# predict the probabilities with the estimated orf
orf_pred <- predict(orf_fit, newdata = X_test)

# predict the probabilities with estimated orf together with variances
orf_pred <- predict(orf_fit, newdata = X_test, inference = TRUE)

# predict the classes with estimated orf
orf_pred <- predict(orf_fit, newdata = X_test, type = "class")
```

---

# print.margins.orf *Print of the Ordered Forest Marginal Effects*

---

## Description

print of estimated marginal effects of the Ordered Forest of class margins.orf

## Usage

```
## S3 method for class 'margins.orf'
print(x, ...)
```

## Arguments

| | |
|---|---|
| x | estimated Ordered Forest Marginal Effect object of type margins.orf |
| ... | further arguments (currently ignored) |

## Details

print.margins.orf provides a first glimpse of the Ordered Forest marginal effects, printed directly to the R console. The printed information contains the results for the marginal effects for each covariate and each outcome class.

## Author(s)

Gabriel Okasa

## Examples

```
## Ordered Forest
require(orf)

# load example data
data(odata)

# specify response and covariates
Y <- as.numeric(odata[, 1])
X <- as.matrix(odata[, -1])

# estimate Ordered Forest
orf_fit <- orf(X, Y)

# estimate marginal effects of the orf
orf_margins <- margins(orf_fit)

# print marginal effects
print(orf_margins)
```

---

# print.orf *Print of the Ordered Forest*

---

## Description

print of an estimated Ordered Forest object of class orf

### Usage

```
## S3 method for class 'orf'
print(x, ...)
```

### Arguments

| | |
|---|---|
| x | estimated Ordered Forest object of class orf |
| ... | further arguments (currently ignored) |

### Details

print.orf provides a first glimpse of the Ordered Forest estimation, printed directly to the R console. The printed information contains the main inputs of the orf function.

### Author(s)

Gabriel Okasa

### Examples

```
# Ordered Forest
require(orf)

# load example data
data(odata)

# specify response and covariates
Y <- as.numeric(odata[, 1])
X <- as.matrix(odata[, -1])

# estimate Ordered Forest
orf_fit <- orf(X, Y)

# print output of the orf estimation
print(orf_fit)
```

---

## print.orf.prediction *Print of the Ordered Forest Prediction*

---

### Description

print of Ordered Forest predictions of class orf.prediction

### Usage

```
## S3 method for class 'orf.prediction'
print(x, ...)
```

### Arguments

| | |
|---|---|
| x | predicted Ordered Forest object of class orf.prediction |
| ... | further arguments (currently ignored) |

### Details

print.orf.prediction provides a first glimpse of the Ordered Forest prediction, printed directly to the R console. The printed information contains the main inputs of the predict.orf function.

### Author(s)

Gabriel Okasa

### Examples

```
# Ordered Forest
require(orf)

# load example data
data(odata)

# specify response and covariates for train and test
idx <- sample(seq(1, nrow(odata), 1), 0.8*nrow(odata))

# train set
Y_train <- as.numeric(odata[idx, 1])
X_train <- as.matrix(odata[idx, -1])

# test set
Y_test <- as.numeric(odata[-idx, 1])
X_test <- as.matrix(odata[-idx, -1])

# estimate Ordered Forest
orf_fit <- orf(X_train, Y_train)

# predict the probabilities with the estimated orf
orf_pred <- predict(orf_fit, newdata = X_test)

# print the prediction object
print(orf_pred)
```

---

## summary.margins.orf *Summary of the Ordered Forest Marginal Effects*

---

### Description

summary of estimated marginal effects of the Ordered Forest of class margins.orf

## Usage

```
## S3 method for class 'margins.orf'
summary(object, latex = FALSE, ...)
```

## Arguments

| | |
|---|---|
| object | estimated Ordered Forest Marginal Effect object of type margins.orf |
| latex | logical, if TRUE latex coded summary will be generated (default is FALSE) |
| ... | further arguments (currently ignored) |

## Details

summary.margins.orf provides estimation results of the Ordered Forest marginal effects. The summary contains the results for the marginal effects for each covariate and each outcome class, optionally with inference as well. Furthermore, summary output as a LaTeX table is supported in order to directly extract the results for the documentation.

## Author(s)

Gabriel Okasa

## Examples

```
## Ordered Forest
require(orf)

# load example data
data(odata)

# specify response and covariates
Y <- as.numeric(odata[, 1])
X <- as.matrix(odata[, -1])

# estimate Ordered Forest
orf_fit <- orf(X, Y)

# estimate marginal effects of the orf
orf_margins <- margins(orf_fit)

# summary of marginal effects
summary(orf_margins)

# summary of marginal effects coded in LaTeX
summary(orf_margins, latex = TRUE)
```

summary.orf *Summary of the Ordered Forest*

## Description

summary of an estimated Ordered Forest object of class orf

## Usage

```
## S3 method for class 'orf'
summary(object, latex = FALSE, ...)
```

## Arguments

| | |
|---|---|
| object | estimated Ordered Forest object of class orf |
| latex | logical, if TRUE latex coded summary will be generated (default is FALSE) |
| ... | further arguments (currently ignored) |

## Details

summary.orf provides a short summary of the Ordered Forest estimation, including the input information regarding the values of hyperparameters as well as the output information regarding the prediction accuracy.

## Author(s)

Gabriel Okasa

## Examples

```
# Ordered Forest
require(orf)

# load example data
data(odata)

# specify response and covariates
Y <- as.numeric(odata[, 1])
X <- as.matrix(odata[, -1])

# estimate Ordered Forest
orf_fit <- orf(X, Y)

# show summary of the orf estimation
summary(orf_fit)

# show summary of the orf estimation coded in LaTeX
summary(orf_fit, latex = TRUE)
```

summary.orf.prediction

*Summary of the Ordered Forest Prediction*

## Description

summary of Ordered Forest predictions of class `orf.prediction`

## Usage

```
## S3 method for class 'orf.prediction'
summary(object, latex = FALSE, ...)
```

## Arguments

| | |
|---|---|
| `object` | predicted Ordered Forest object of class `orf.prediction` |
| `latex` | logical, if TRUE latex coded summary will be generated (default is FALSE) |
| `...` | further arguments (currently ignored) |

## Details

`summary.orf.prediction` provides a main summary of the Ordered Forest prediction, including the input information regarding the values of hyperparameters as well as the inputs of the `predict.orf` function.

## Author(s)

Gabriel Okasa

## Examples

```
# Ordered Forest
require(orf)

# load example data
data(odata)

# specify response and covariates for train and test
idx <- sample(seq(1, nrow(odata), 1), 0.8*nrow(odata))

# train set
Y_train <- as.numeric(odata[idx, 1])
X_train <- as.matrix(odata[idx, -1])

# test set
Y_test <- as.numeric(odata[-idx, 1])
X_test <- as.matrix(odata[-idx, -1])
```

```
# estimate Ordered Forest
orf_fit <- orf(X_train, Y_train)

# predict the probabilities with the estimated orf
orf_pred <- predict(orf_fit, newdata = X_test)

# summary of the prediction object
summary(orf_pred)

# show summary of the orf prediction coded in LaTeX
summary(orf_pred, latex = TRUE)
```

# Index