

# Package ‘FeatureTerminatorR’

July 21, 2025

**Type** Package

**Title** Feature Selection Engine to Remove Features with Minimal Predictive Power

**Version** 1.0.0

**Maintainer** Gary Hutson <hutsons-hacks@outlook.com>

**Description** The aim is to take in data.frame inputs and utilises methods, such as recursive feature engineering, to enable the features to be removed.

What this does differently from the other packages, is that it gives you the choice to remove the variables manually, or it automated this process.

Feature selection is a concept in machine learning, and statistical pipelines, whereby unimportant, or less predictive variables are eliminated from the analysis, see Boughaci (2018) <[doi:10.1007/s40595-018-0107-y](https://doi.org/10.1007/s40595-018-0107-y)>.

**License** GPL-3

**Encoding** UTF-8

**Imports** ggplot2, caret, tibble, dplyr, stats, utils, lattice, e1071, randomForest

**Suggests** knitr, markdown, rmarkdown

**LazyData** FALSE

**VignetteBuilder** knitr

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Author** Gary Hutson [aut, cre] (ORCID: <<https://orcid.org/0000-0003-3534-6143>>)

**Repository** CRAN

**Date/Publication** 2021-07-14 09:00:04 UTC

## Contents

mutlicol_terminator . . . . .	2
rfeTerminator . . . . .	3

<b>Index</b>	<b>5</b>
--------------	----------

---

mutlicol_terminator	<i>Multicollinearity TerminatorR - Feature Selection to remove highly correlated values</i>
---------------------	---

---

## Description

This function looks at highly correlated features and allows for a correlation cutoff to be set. Outputs from this function allow for correlations and covariance matrices to be created, alongside visuals and the ability to remove highly correlated features from your statistic pipeline.

## Usage

```
mutlicol_terminator(df, x_cols, y_cols, alter_df = TRUE, cor_sig = 0.9)
```

## Arguments

df	The data frame to pass with the x and y variables
x_cols	The independent variables we want to analyse for multicollinearity
y_cols	The dependent variables(s) in your predictive model
alter_df	<b>Default=TRUE</b> - Determines whether the underlying features are removed from the data frame, with TRUE being the default.
cor_sig	<b>Default=0.9</b> - A correlation significance for the cut-off in inter-feature correlation

## Value

A list containing the outputs highlighted hereunder:

det

- **"rfe\_model\_fit\_results"** a list of the model fit results. Including the optimal features
- **"rfe\_reduced\_features"** a data.frame object with the reduced variables and data
- **"rfe\_original\_data"** a data.frame object with the original data passed for manual exclusion based on fit outputs
- **"rfe\_reduced\_data"** output of setting the alter\_df=TRUE will remove the features / IVs from the data.frame

## Examples

```
library(caret)
library(FeatureTerminatorR)
library(tibble)
library(dplyr)
df <- iris
mc_fit <- mutlicol_terminator(df, 1:4,5, cor_sig = 0.90, alter_df = TRUE)
#View the correlation matrix
mc_fit$corr_matrix
#View the reduced data
head(mc_fit$feature_removed_df,10)
```

## Description

This function removes the redundant features in a model and automatically selects the best combination of features to remove. This utilises, by default, the random forest mean decrease in accuracy methods, from the caret package, reference Kuhn (2021). This function is a wrapper for the **rfe()** function

## Usage

```
rfeTerminator(
  df,
  x_cols,
  y_cols,
  method = "cv",
  kfolds = 10,
  sizes = c(1:100),
  alter_df = TRUE,
  eval_funcs = rfFuncs,
  ...
)
```

## Arguments

df	data frame to fit the recursive feature engineering algorithm to
x_cols	the independent variables to be used for the recursive feature engineering algorithm
y_cols	the dependent variables to be used in the prediction
method	<b>Default = "cv"</b> - cross validation method for resampling, other options "repeatedcv"
kfolds	<b>Default = 10</b> - the number of k folds - train / test splits to compute when resampling
sizes	the sizes of the search boundary for the search
alter_df	<b>Default = TRUE</b> - will remove the redundant features, due to having a lesser affect on the mean decrease in accuracy, or other measures.
eval_funcs	<b>Default = rfFuncs</b> (Random Forest Mean Decrease Accuracy method). Other options: rfe, lmFuncs, rfFuncs, treebagFuncs, nbFuncs, pickSizeBest, pickSizeTolerance.
...	Function forwarding to main 'caret::rfe()' function' to pass in additional parameters native to caret

## Details

With the `df_alter` set to `TRUE` the recursive feature algorithm chosen will automatically remove the features from the returned tibble embedded in the list.

## Value

A list containing the outputs highlighted hereunder:

- **"rfe\_model\_fit\_results"** a list of the model fit results. Including the optimal features
- **"rfe\_reduced\_features"** a data.frame object with the reduced variables and data
- **"rfe\_original\_data"** a data.frame object with the original data passed for manual exclusion based on fit outputs
- **"rfe\_reduced\_data"** output of setting the `alter_df=TRUE` will remove the features / IVs from the data.frame

## References

Kuhn (2021) Recursive Feature Elimination. <https://topepo.github.io/caret/recursive-feature-elimination.html>

## Examples

```
library(caret)
library(tibble)
library(FeatureTerminatorR)
library(dplyr)
df <- iris
# Passing in the indexes as slices x values located in index 1:4 and y value in location 5
rfe_fit <- rfeTerminator(df, x_cols= 1:4, y_cols=5, alter_df = TRUE, eval_funcs = rfFuncs)
#Explore the optimal model results
print(rfe_fit$rfe_model_fit_results)
# Explore the optimal variables selected
print(rfe_fit$rfe_model_fit_results$optVariables)
# Explore the original data passed to the frame
print(head(rfe_fit$rfe_original_data))
# Explore the data adapted with the less important features removed
print(head(rfe_fit$rfe_reduced_data))
```

# Index

`mutlicol_terminator`, [2](#)

`rfeTerminator`, [3](#)