

# Package: nonet (via r-universe)

October 22, 2024

**Title** Weighted Average Ensemble without Training Labels

**Version** 0.4.0

**Description** It provides ensemble capabilities to supervised and unsupervised learning models predictions without using training labels. It decides the relative weights of the different models predictions by using best models predictions as response variable and rest of the mo. User can decide the best model, therefore, It provides freedom to user to ensemble models based on their design solutions.

**Depends** R (>= 3.5.0)

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** TRUE

**Imports** caret (>= 6.0.78), dplyr, randomForest, ggplot2, rlist (>= 0.4.6.1), glmnet, tidyverse, e1071, purrr, pROC (>= 1.13.0), rlang (>= 0.2.1),

**RoxygenNote** 6.1.1

**Suggests** testthat, knitr, rmarkdown, ClusterR

**URL** <https://open.gslab.com/nonet/>

**BugReports** <https://github.com/GSLabDev/nonet/issues>

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Aviral Vijay [aut, cre], Sameer Mahajan [aut]

**Maintainer** Aviral Vijay <aviral.vijay@gslab.com>

**Date/Publication** 2019-01-15 10:50:03 UTC

**Repository** <https://aviralvijay-gslab.r-universe.dev>

**RemoteUrl** <https://github.com/cran/nonet>

**RemoteRef** HEAD

**RemoteSha** 304487e49552d1e00b9f3e0f20227c960b9767b6

## Contents

banknote_authentication . . . . .	2
nonet_ensemble . . . . .	3
nonet_plot . . . . .	4
<b>Index</b>	<b>7</b>

---

banknote\_authentication

*Bank Note Authentication Data Set*

---

### Description

Bank Note Authentication Data Set is used to show the functionality of nonet package. This Data Set has below attribute information as variance of Wavelet Transformed image (continuous), skewness of Wavelet Transformed image (continuous), curtosis of Wavelet Transformed image (continuous), entropy of image (continuous), class (integer).

### Usage

```
data("banknote_authentication")
```

### Format

A data frame with 1372 observations on the following 5 variables.

variance a numeric vector

skewness a numeric vector

curtosis a numeric vector

entropy a numeric vector

class a numeric vector

### Details

Data were extracted from images that were taken from genuine and forged banknote-like specimens. For digitization, an industrial camera usually used for print inspection was used. The final images have 400x 400 pixels. Due to the object lens and distance to the investigated object gray-scale pictures with a resolution of about 660 dpi were gained. Wavelet Transform tool were used to extract features from images.

### Source

This DataSet is fetched from UCI Website. URL is <https://archive.ics.uci.edu/ml/datasets/banknote+authentication>

## References

Owner of database: Volker Lohweg (University of Applied Sciences, Ostwestfalen-Lippe, volker.lohweg '@' hs-owl.de) Donor of database: Helene Dörksen (University of Applied Sciences, Ostwestfalen-Lippe, helene.doerksen '@' hs-owl.de) Date received: August, 2012

## Examples

```
data(banknote_authentication)
dataframe <- data.frame(banknote_authentication)
head(dataframe)
str(banknote_authentication)
```

---

nonet_ensemble	<i>Ensemble Prediction without using training labels</i>
----------------	--

---

## Description

Ensemble Prediction without using training labels

## Usage

```
nonet_ensemble(object, best_modelname)
```

## Arguments

**object** prediction\_list object, as from 'tune\_models'  
**best\_modelname** Best model name is one which performance better while evaluating using any evaluation matrix like confusion matrix.

## Value

A list of ensembled predictions. You can evaluate the performance of ensembled prediction using the evaluation matrix as Confusion matrix or AUROC.

## Examples

```
# nonet_ensemble functionality can be explained via below example
# Setup
library(caret)
library(nonet)
library(rlist)

# Load Data
dataframe <- data.frame(banknote_authentication[600:900, ])
dataframe$class <- as.factor(ifelse(dataframe$class >= 1, 'Yes', 'No'))

# First Model
# Splitting into train and test
```

```

index <- createDataPartition(dataframe$class, p=0.75, list=FALSE)
trainSet <- dataframe[ index,]
testSet <- dataframe[-index,]

#Feature selection
control <- rfeControl(functions = rfFuncs,
  method = "repeatedcv",
  repeats = 1,
  verbose = FALSE)

outcomeName <- 'class'
predictors <- c("variance", "skewness")

banknote_rf <- train(trainSet[,predictors],trainSet[,outcomeName],method='rf')
preds_rf_first <- predict.train(object=banknote_rf,testSet[,predictors],type="prob")
preds_rf_first_raw <- predict.train(object=banknote_rf,testSet[,predictors],type="raw")

# Second Model
# Splitting into train and test
index <- createDataPartition(dataframe$class, p=0.75, list=FALSE)
trainSet <- dataframe[ index,]
testSet <- dataframe[-index,]

#Feature selection
control <- rfeControl(functions = rfFuncs,
  method = "repeatedcv",
  repeats = 2,
  verbose = FALSE)

outcomeName <- 'class'
predictors <- c("curtosis", "entropy")

banknote_rf <- train(trainSet[,predictors],trainSet[,outcomeName],method='rf')
preds_rf_second <- predict.train(object=banknote_rf,testSet[,predictors],type="prob")
preds_rf_second_raw <- predict.train(object=banknote_rf,testSet[,predictors],type="raw")

Stack_object <- list(preds_rf_first$Yes, preds_rf_second$Yes)
names(Stack_object) <- c("model_rf_first", "model_rf_second")

# Prediction using nonet_ensemble function
prediction_nonet <- nonet_ensemble(Stack_object, "model_rf_second")

```

---

nonet\_plot

---

*Plot the predictions or results of nonet\_ensemble*


---

## Description

Plot the predictions or results of nonet\_ensemble

**Usage**

```
nonet_plot(x, y, dataframe, plot_type = NULL, nonet_size = 20,  
           nonet_alpha = 0.3, nonet_bins = 25)
```

**Arguments**

x	x axis variable name or histogram entity name
y	y axis variable name
dataframe	dataframe which is used for plotting purpose.
plot_type	type of plot, if not provided it takes "NULL"
nonet_size	size of plot need to feed in ggplot
nonet_alpha	value of alpha for ggplot
nonet_bins	number of bins for histogram

**Value**

plotted for the plot results provided as input.

**Examples**

```
# nonet_plot functionality can be explained via below example  
# Setup  
library(caret)  
library(nonet)  
library(ggplot2)  
  
# Load Data  
dataframe <- data.frame(banknote_authentication[600:900, ])  
dataframe$class <- as.factor(ifelse(dataframe$class >= 1, 'Yes', 'No'))  
  
# Splitting into train and test  
index <- createDataPartition(dataframe$class, p=0.75, list=FALSE)  
trainSet <- dataframe[ index,]  
testSet <- dataframe[-index,]  
  
# Feature selection  
control <- rfeControl(functions = rfFuncs,  
                      method = "repeatedcv",  
                      repeats = 2,  
                      verbose = FALSE)  
  
outcomeName <- 'class'  
predictors <- c("curtosis", "entropy")  
  
# Model Training & predictions  
banknote_rf <- train(trainSet[,predictors], trainSet[,outcomeName], method='rf')  
predictions_rf_raw <- predict.train(object=banknote_rf, testSet[,predictors], type="raw")  
  
# Results
```

```
nonet_eval_rf <- confusionMatrix(predictions_rf_raw, testSet[, outcomeName])
eval_rf_df <- data.frame(nonet_eval_rf$table)
nonet_plot(eval_rf_df$Prediction, eval_rf_df$Reference, eval_rf_df, plot_type = "point")
```

# Index

\* **datasets**

banknote\_authentication, 2

banknote\_authentication, 2

nonet\_ensemble, 3

nonet\_plot, 4