Fitting prediction rule ensembles using R package pre

A short tutorial

First, we have to install and load **pre**, which is available from CRAN as well as GitHub. The latest state-of-the-art version is always available from Github and so we install that version:

```
library(devtools)
install_github("marjoleinF/pre")
```

```
library(pre)
```

To illustrate some of the package functionality, we use an example dataset from a paper by Carillo et al. (2001). The data was used to predict depression (bdi) based on personality scale scores. Sex (sexo) and age (edad) were also included:

```
library(foreign)
car_data <- read.spss("https://github.com/marjoleinF/misc/raw/master/data Carillo et al.sav",
                      to.data.frame = TRUE)
names(car_data)
```

```
##  [1] "n1"      "n2"      "n3"      "n4"      "n5"      "n6"      "ntot"
##  [8] "e1"      "e2"      "e3"      "e4"      "e5"      "e6"      "etot"
## [15] "open1"   "open2"   "open3"   "open4"   "open6"   "opentot" "altot"
## [22] "contot"  "bdi"     "sexo"    "edad"    "open5"
```

To fit the prediction rule ensemble, we regress depression (bdi) on all other variables in the dataset:

```
set.seed(2896904)
car_pre <- pre(formula = bdi ~ ., data = car_data)
```

Note we set the random seed to be able to reproduce our results later.

We can check out the resulting prediction rule ensemble:

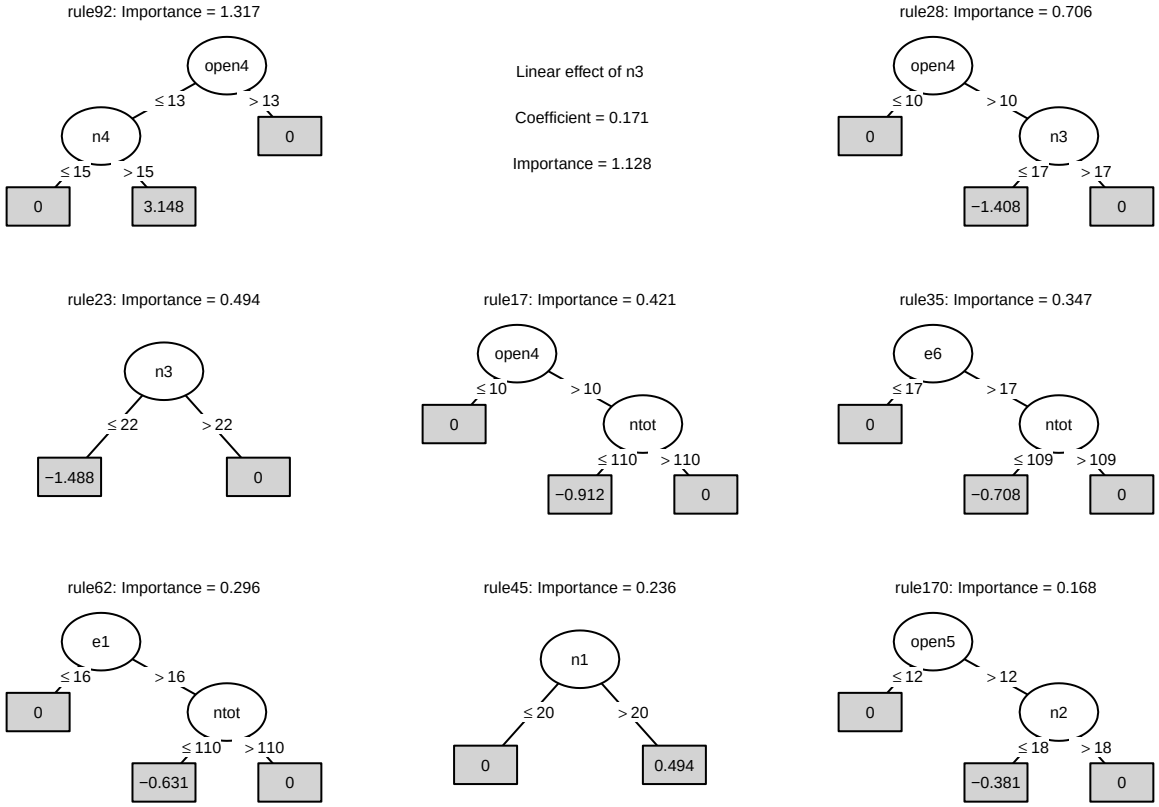
```
print(car_pre)
```

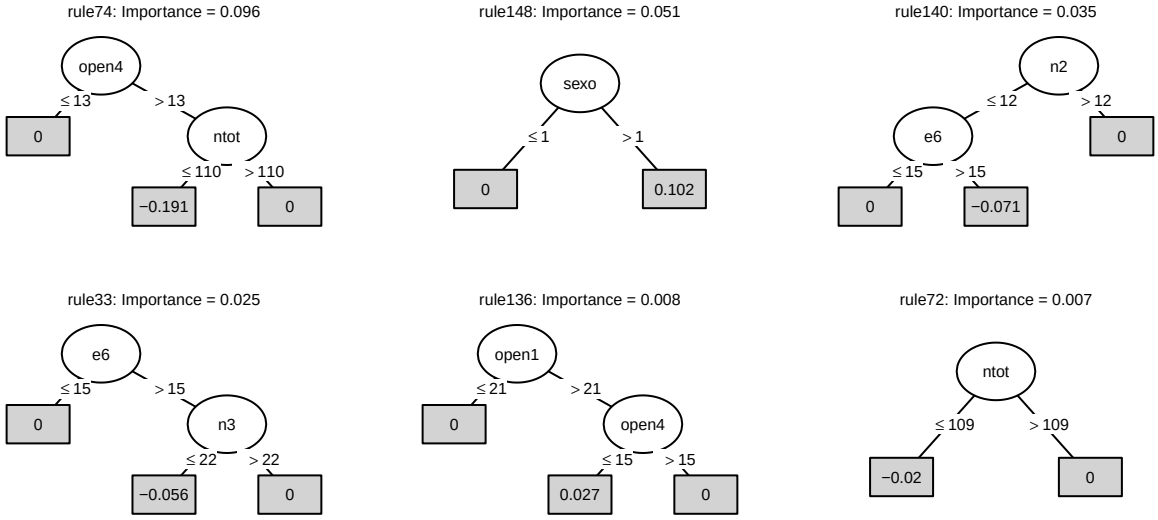
```
##
## Final ensemble with cv error within 1se of minimum:
##   lambda =  0.6932192
##   number of terms = 15
##   mean cv error (se) = 39.56747 (8.596313)
##
##   cv error type : Mean-Squared Error
##
##          rule  coefficient               description
##   (Intercept)   8.50656056                      <NA>
##        rule92   3.14839596    n4 > 15 & open4 <= 13
##        rule23  -1.48780821                  n3 <= 22
##        rule28  -1.40833139    n3 <= 17 & open4 > 10
##        rule17  -0.91188999 ntot <= 110 & open4 > 10
##        rule35  -0.70790325    ntot <= 109 & e6 > 17
##        rule62  -0.63108360    ntot <= 110 & e1 > 16
##        rule45   0.49403398                   n1 > 20
##       rule170  -0.38112203    n2 <= 18 & open5 > 12
##        rule74  -0.19097083 ntot <= 110 & open4 > 13
##            n3   0.17087688        2 <= n3 <= 30.225
##       rule148   0.10169458                 sexo > 1
```

```
##       rule140  -0.07111969        e6 > 15 & n2 <= 12
##        rule33  -0.05557589        n3 <= 22 & e6 > 15
##       rule136   0.02684615  open4 <= 15 & open1 > 21
##        rule72  -0.02030925               ntot <= 109
```

We can plot the ensemble:

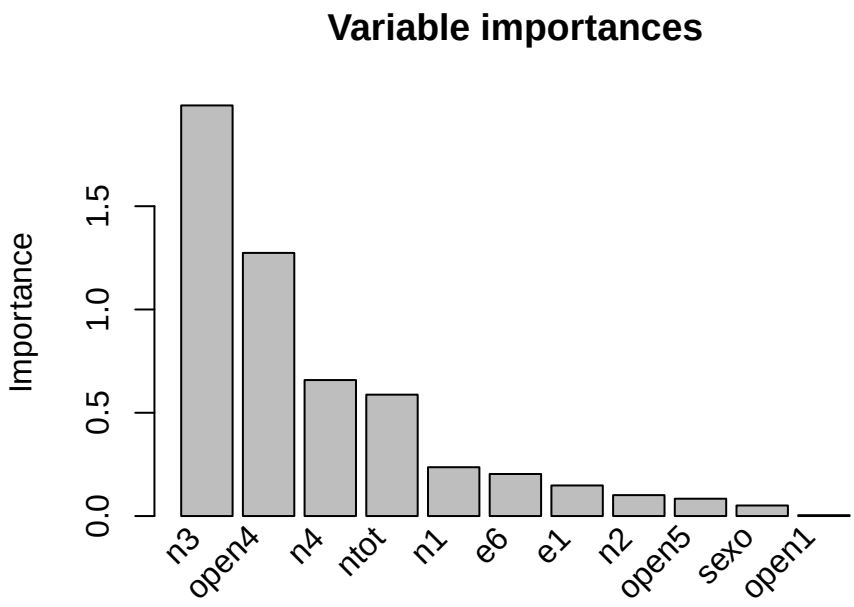
```
plot(car_pre, max.terms.plot = 9, cex = .5)
```





We can get an estimate of the importance of variables and base learners:

```
imps <-importance(car_pre, round = 4)
```



We can generate predictions for new observations (though note that these observations are not really new, they were already used for training the ensemble):

```
predict(car_pre, newdata = car_data[1:10,])
```

```
##         1         2         3         4         5         6         7
##  5.374477 13.101640  5.505058  4.522029  5.465690  8.541132 17.574715
##         8         9        10
##  7.226256  2.640446  5.016063
```

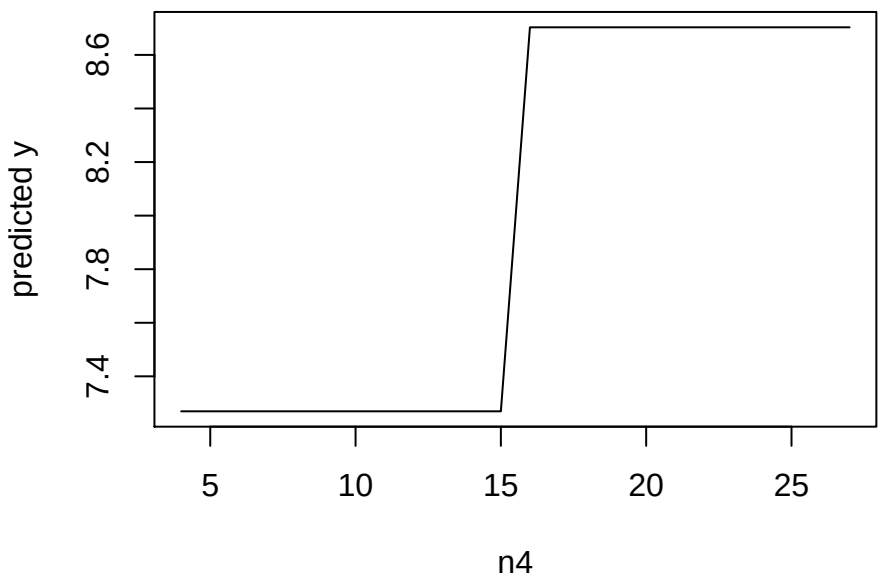
We can obtain a realistic estimate of future prediction error through full k-fold cross validation (k is set to 10, by default):

```
set.seed(321447)
cv_car <- cvpre(car_pre)
cv_car$accuracy
```

```
## $MSE
##       MSE        se
## 41.316454  6.852863
##
## $MAE
##       MAE        se
## 4.8539940 0.3999459
```

We can assess the effect of a single variable on the predictions of the ensemble:

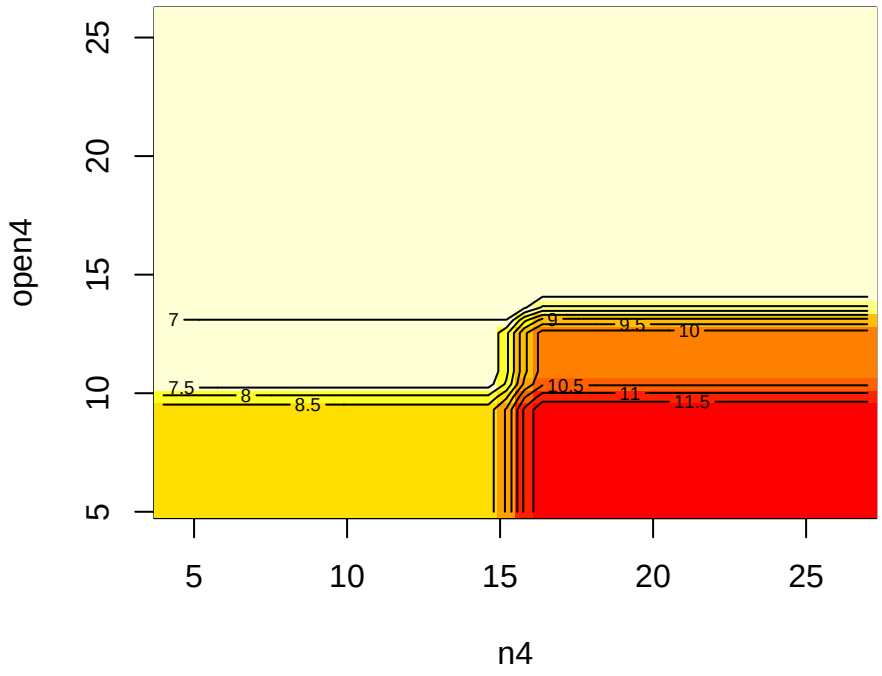
```
singleplot(car_pre, "n4")
```



We can assess the effect of pairs of variables on the predictions of the ensemble:

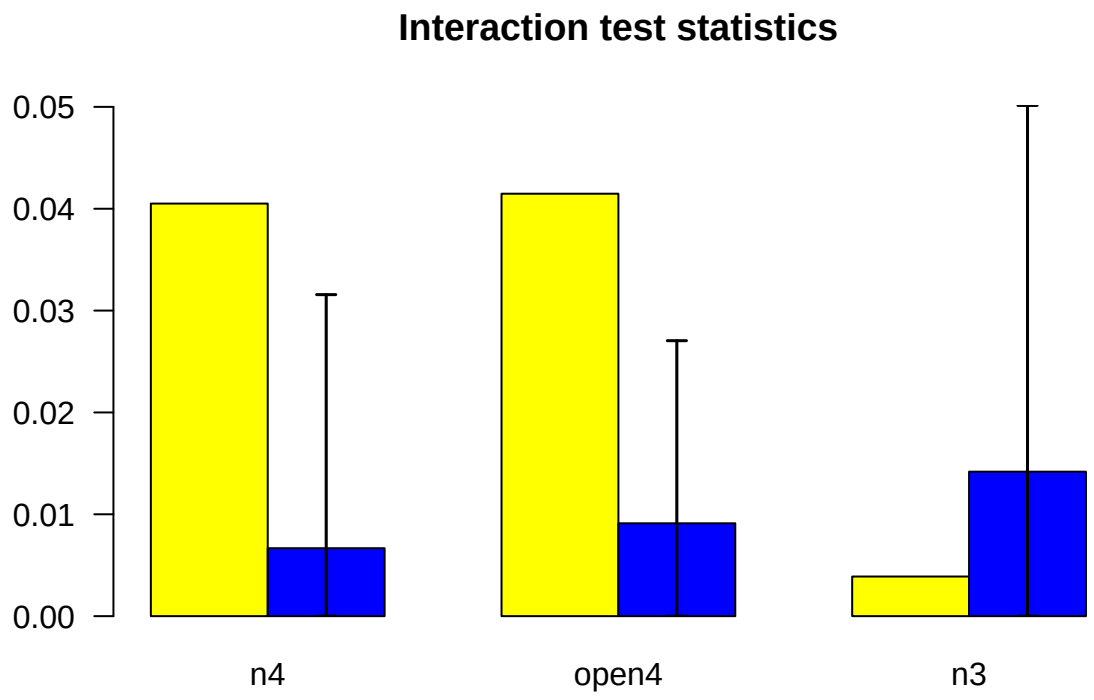
```
pairplot(car_pre, c("n4", "open4"))
```

```
## NOTE: function pairplot uses package 'akima', which has an ACM license. See also https://www.acm.org/
```



There could be an interaction: the effect of open4 may depend on n4. For higher levels of openness, there seems to be no association between neuroticism and depression, whereas for lower levels of openness, there seems to be a positive association between neuroticism and depression. On the other hand, this pattern may occur because of intercorrelatedness and two main effects of open4 and n4. We can statistically test whether predictor variables are involved in interactions in the data:

```
set.seed(74276387)
null_mods <- bsnullinteract(car_pre)
ints <- interact(car_pre, c("n4", "open4", "n3"), nullmods = null_mods)
```



The yellow bars represent the interaction test statistic of the predictor variable in the fitted ensemble. The blue bars represent the median (and the error bars represent the .05 and .95 quantiles) of the null interaction models generated by bootstrap sampling. The plot suggests that n4 and open4 may indeed be involved in interactions, and n3 not. Note that 10 null interaction models are generated by default, which is not adequate for statistical testing, but has been chosen as the default for computational speed.

References

Carrillo, J. M., Rojo, N., Sanchez-Bernardos, M. L., & Avia, M. D. (2001). Openness to experience and depression. European Journal of Psychological Assessment, 17(2), 130.

Fokkema, M. & Christofferson, B. (2017). pre: Prediction Rule Ensembles. R package version 0.2.2.

Fokkema, M., Smits, N., Kelderman, H., & Penninx, B. W. (2015). Connecting clinical and actuarial prediction with rule-based methods. Psychological assessment, 27(2), 636.

Friedman, J. H., & Popescu, B. E. (2008). Predictive learning via rule ensembles. The Annals of Applied Statistics, 2(3), 916-954.