Empirical Industrial Organisation 2b-Part 2: Comparing Out-Of-Sample Prediction Accuracy of a Multinomial Logit Model and a Random Forest

Previously in the lecture, we analysed a data set about heating choices using a multinomial logit model, that allows us to estimate utility functions for heating systems for different households.

In a similar spirit as in your exercise classes about machine learning, we now want to analyse, how well our multinomial logit model predicts out-of-sample compared to a random forest. Random forest is a very popular machine learning method that often yields very good out-of-sample prediction accuracy.

a) Load the dataset Heating from the package mlogit. Split it into a training data set (2/3 of rows) and a test data set (1/3 of rows)

```r
# Load data
library("mlogit")
data("Heating", package = "mlogit")
dat = Heating

# Set a random seed for reproducibility
set.seed(123456789)

# Choose 600 rows for training
train.ind = sample.int(900,600,replace = FALSE)

# Create training and test data frames
train = dat[train.ind,]
test = dat[-train.ind,]
```

b) Estimate a multinomial logit model, with alternative specific constants and investment and operation cost as explanatory variables on the training data set.

```r
# First we transform the training data set
# into the long format used by the
# mlogit function
train.long = mlogit.data(train, shape="wide", choice="depvar", varying=c(3:12))

# Estimate the mlogit model on the
# training data
ml <- mlogit(depvar~ic+oc, train.long, reflevel = "hp")
# Show a summary of the results
summary(ml)
```

```
##
## Call:
## mlogit(formula = depvar ~ ic + oc, data = train.long, reflevel = "hp",
##     method = "nr", print.level = 0)
##
## Frequencies of alternatives:
##       hp       ec       er       gc       gr
## 0.058333 0.071667 0.096667 0.641667 0.131667
##
## nr method
## 6 iterations, 0h:0m:0s
## g'(-H)^-1g = 4.66E-06
## successive function values within tolerance limits
##
## Coefficients :
##                   Estimate  Std. Error z-value  Pr(>|z|)
## ec:(intercept)  1.80227716  0.54706680  3.2944 0.0009862 ***
## er:(intercept)  1.97239765  0.43920786  4.4908 7.095e-06 ***
## gc:(intercept)  1.66686598  0.27534956  6.0536 1.416e-09 ***
## gr:(intercept)  0.14882292  0.25118008  0.5925 0.5535192
## ic             -0.00142809  0.00077326 -1.8468 0.0647701 .
## oc             -0.00765296  0.00190821 -4.0105 6.058e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-Likelihood: -668.82
## McFadden R^2:  0.015419
## Likelihood ratio test : chisq = 20.948 (p.value = 2.8258e-05)
```

c) Using the test data set predict choice probabilities for each heating system based on our model estimated with the training data. We will analyse and compare the predictions with the random forest later.

```r
# Transform test data set into required
# format
test.long = mlogit.data(test, shape="wide", choice="depvar", varying=c(3:12))

# Predict on test data
ml.pred = predict(ml,test.long)
# Show first 3 rows
ml.pred[1:3,]
```

```
##           hp         ec         er        gc         gr
## 1 0.05802481 0.07249368 0.12104249 0.5751534 0.17328566
## 2 0.04081689 0.04719849 0.09811959 0.7239022 0.08996279
## 3 0.04472541 0.08940306 0.15610797 0.5533020 0.15646156
```

d) Now use the package `ranger` to train a random forest that predicts choice probabilities for the heating system on our training data set. We don't tune the parameters of the random forest, but take the default parameters. You can use all explanatory variables that could be relevant for a households choice. Afterwards compute predicted choice probabilities for the test data set.

```r
library(ranger)


# Use all columns except for idcase to
# predict heating choice
rf = ranger(depvar ~ . - idcase, train, probability = TRUE)
rf
```

```
## Ranger result
##
## Call:
##  ranger(depvar ~ . - idcase, train, probability = TRUE)
##
## Type:                             Probability estimation
## Number of trees:                  500
## Sample size:                      600
## Number of independent variables:  14
## Mtry:                              3
## Target node size:                  10
## Variable importance mode:          none
## Splitrule:                         gini
## OOB prediction error (Brier s.):  0.3852378
```

```
# Compute predicted probabilities
# for test data set
rf.pred = predict(rf,test)$prediction
rf.pred[1:3,]
```

```
##             gc        gr         ec        er         hp
## [1,] 0.5918714 0.1480071 0.04759127 0.1218690 0.09066111
## [2,] 0.6588532 0.1608968 0.04561825 0.0977373 0.03689444
## [3,] 0.5947698 0.1224127 0.05580000 0.1225159 0.10450159
```

```
rf.pred = rf.pred[, colnames(ml.pred)]
rf.pred[1:3,]
```

```
##              hp         ec        er        gc        gr
## [1,] 0.09066111 0.04759127 0.1218690 0.5918714 0.1480071
## [2,] 0.03689444 0.04561825 0.0977373 0.6588532 0.1608968
## [3,] 0.10450159 0.05580000 0.1225159 0.5947698 0.1224127
```

e) The following code creates data frame of prediction probailities in a long format that will be suited for later analysis with dplyr and ggplot2. Try to understand step by step, what the following code does.

```
library(dplyr)
library(tidyr)

pred.wide = rbind(
  cbind(data.frame(model="mlogit", choice=test$depvar), ml.pred),
  cbind(data.frame(model="rf",choice=test$depvar), rf.pred)
)
pred.wide[1:3,]
```

```
##    model choice         hp         ec         er        gc         gr
## 1 mlogit     er 0.05802481 0.07249368 0.12104249 0.5751534 0.17328566
## 2 mlogit     gc 0.04081689 0.04719849 0.09811959 0.7239022 0.08996279
## 3 mlogit     gc 0.04472541 0.08940306 0.15610797 0.5533020 0.15646156
```

```
pred = pred.wide %>%
  gather(key="option",value="prob", ec, er, gc, gr, hp) %>%
  arrange(model,  option)
pred[1:3,]
```

```
##    model choice option       prob
## 1 mlogit     er     ec 0.07249368
## 2 mlogit     gc     ec 0.04719849
## 3 mlogit     gc     ec 0.08940306
```

```
cpred = filter(pred, option==choice)
cpred[1:3,]
```

```
##    model choice option       prob
## 1 mlogit     ec     ec 0.10374839
## 2 mlogit     ec     ec 0.08167612
## 3 mlogit      ec      ec 0.03986662
```

f) Use the dplyr functions `group_by` and `summarize` to compute the mean predicted probability of the actual chosen heating system in the test data set for the mlogit and the (not tuned) random forest. Which model has better out-of-sample accuracy according to this measure? Also show this measure separately for each heating system.

```r
# Average predicted probability for the actually
# chosen heating system
cpred %>% group_by(model) %>%
  summarize(mean.prob = mean(prob))
```

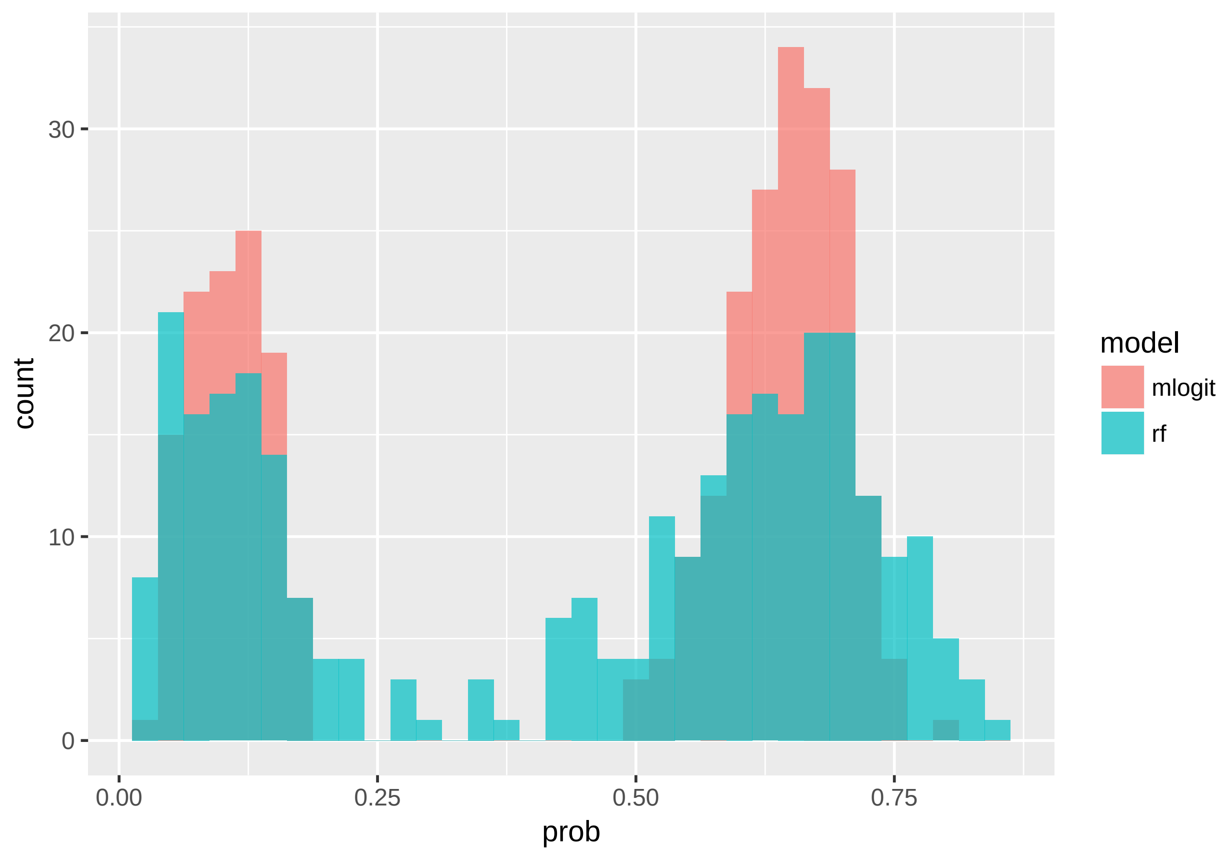
```
## # A tibble: 2 x 2
##   model  mean.prob
##   <fct>      <dbl>
## 1 mlogit     0.445
## 2 rf         0.435
```

```r
# Separately for each heating system
cpred %>% group_by(model,choice) %>%
  summarize(mean.prob = mean(prob)) %>%
  spread(model, mean.prob)
```

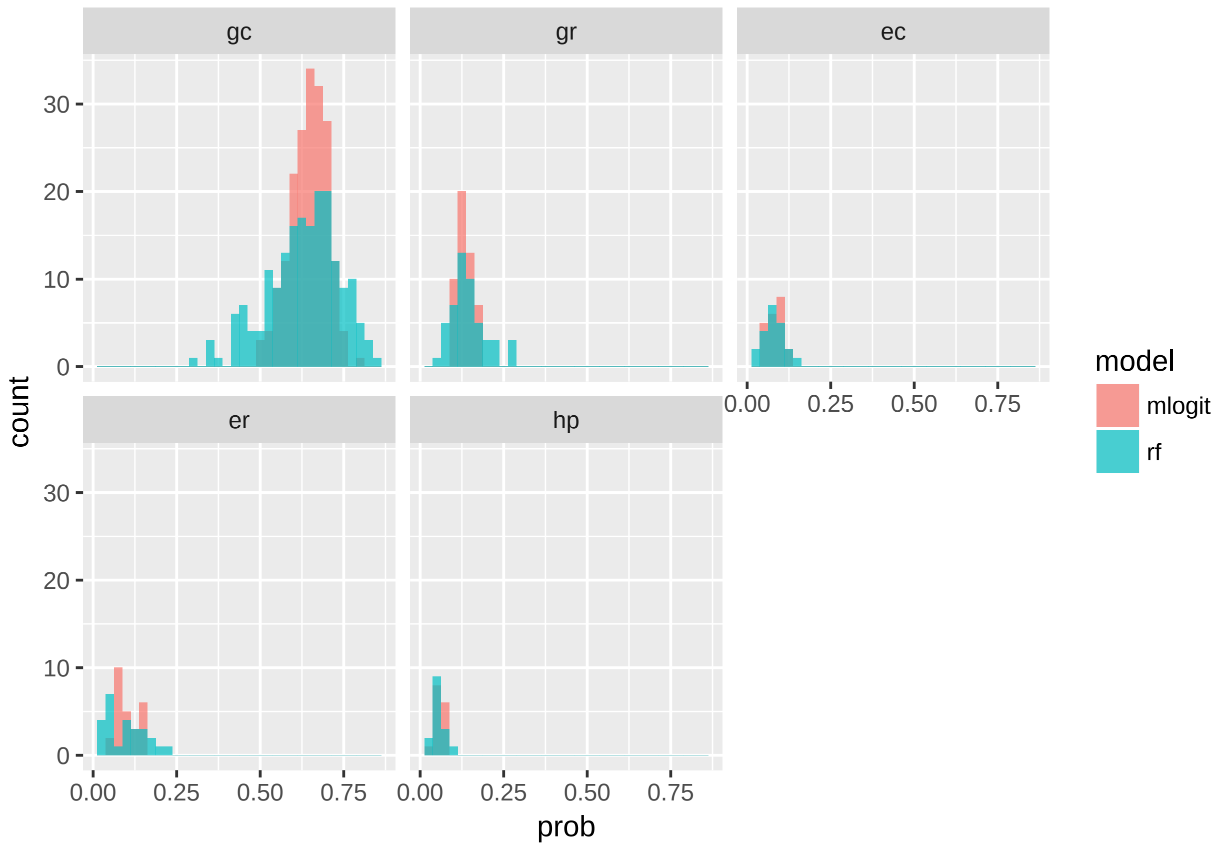
```
## # A tibble: 5 x 3
##   choice mlogit     rf
##   <fct>   <dbl>  <dbl>
## 1 gc     0.646  0.628
## 2 gr     0.135  0.145
## 3 ec     0.0832 0.0794
## 4 er     0.102  0.0966
## 5 hp     0.0597 0.0586
```

g) Compare graphically the histograms of predicted probabilities of the chosen alternative for both models. Use the ggplot2 package.

```r
library(ggplot2)

ggplot(cpred, aes(x=prob, group=model, fill=model)) +
  geom_histogram(alpha=0.7,binwidth = 0.025, position = "identity")
```



```r
# Separately for each heating system
ggplot(cpred, aes(x=prob, group=model, fill=model)) +
  geom_histogram(alpha=0.7,binwidth = 0.025, position = "identity") +
  facet_wrap(~choice)
```



h) (Optional) In the exercise classes on machine learning, you learned how one can use cross-validation for parameter tuning. Another application of cross validation is to repeat the whole procedure to estimate models on the training data set and predict it on a test data set, for different folds for test and training data.

This allows for better estimates of the out-of-sample prediction accuracy, but it takes longer. Take a look at the code below that implements this procedure.

```r
library(restorepoint)
est.and.pred = function(train, test, fold=0) {
  restore.point("est.and.pred")

  # Estimate mlogit on train data
  train.long = mlogit.data(train, shape="wide", choice="depvar", varying=c(3:12))
  ml <- mlogit(depvar~ic+oc | income, train.long)

  # Predict mlogit on test data
  test.long = mlogit.data(test, shape="wide", choice="depvar", varying=c(3:12))
  ml.pred = predict(ml,test.long)

  # Estimate random forest on train data
  rf = ranger(depvar ~ . - idcase, train, probability = TRUE)

  # Predicted probabilities
  # for random forest on test data
  rf.pred = predict(rf,test)$prediction
  rf.pred = rf.pred[, colnames(ml.pred)]
  # Create prediction data set
  # in a nice long format
  pred.wide = rbind(
    cbind(data.frame(model="mlogit", choice=test$depvar), ml.pred),
    cbind(data.frame(model="rf",choice=test$depvar), rf.pred)
  )
  pred = pred.wide %>%
    gather(key="option",value="prob", ec, er, gc, gr, hp)

  # Only return probabilities for
  # actually chosen options
  cpred = filter(pred, option==choice) %>%
    mutate(fold=fold)
  cpred
}

# Create k=3 folds
k = 3
# Assign a random fold to each row
folds = sample.int(900) %% 3 +1
head(folds)
```

```
## [1] 1 3 2 1 2 3
```

```r
table(folds)
```

```
## folds
##   1   2   3
## 300 300 300
```

```r
# Let each of the k-folds be the test data set and repeat or procedure above
li = lapply(1:k, function(fold) {
  cat("\nfold ", fold)

  train = dat[folds != fold,]
  test = dat[folds == fold,]
  cpred = est.and.pred(train, test, fold)
  cpred
})
```

```
##
## fold  1
## fold  2
## fold  3
```

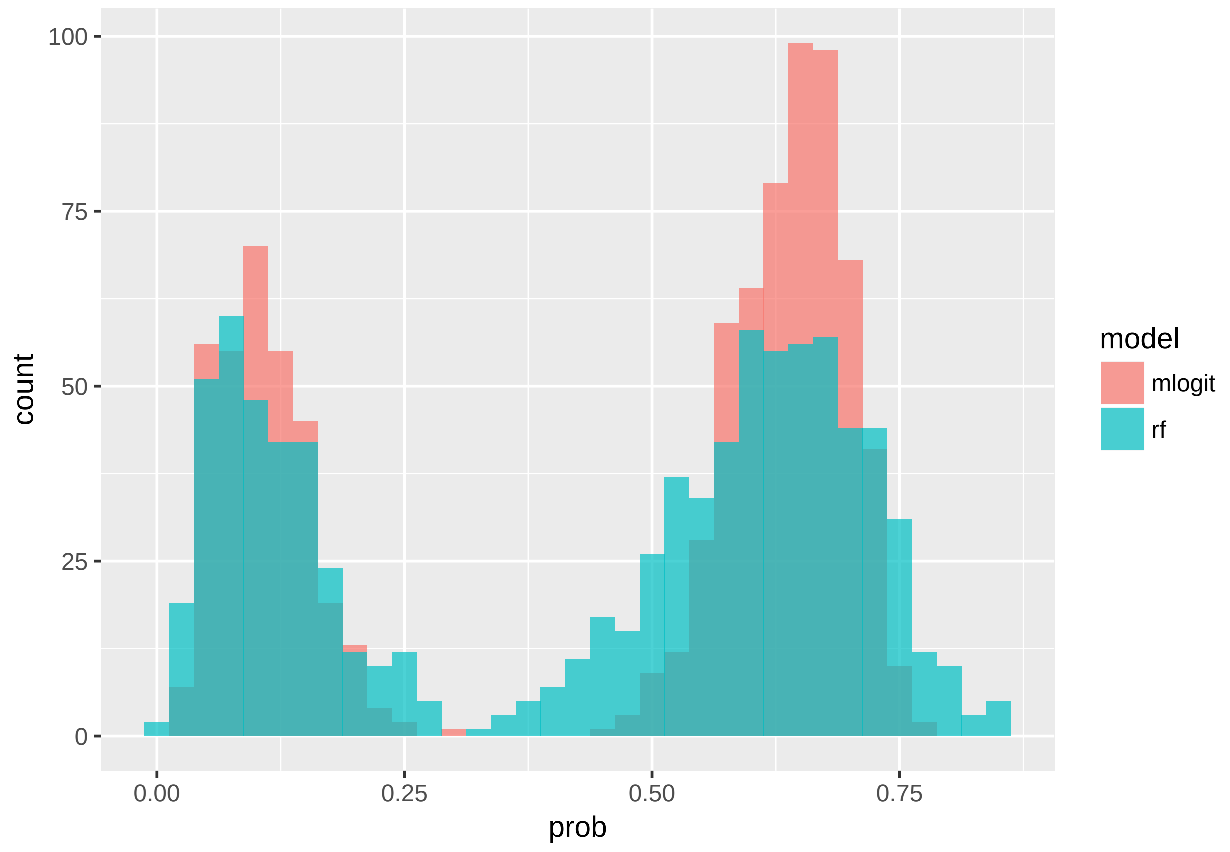
```r
# Combine the returned predictions of all
# folds
cpred = bind_rows(li)

# Average predicted probability for the actually
# chosen heating system
cpred %>% group_by(model) %>%
  summarize(mean.prob = mean(prob))
```

```
## # A tibble: 2 x 2
##   model  mean.prob
##   <fct>       <dbl>
## 1 mlogit      0.446
## 2 rf          0.435
```

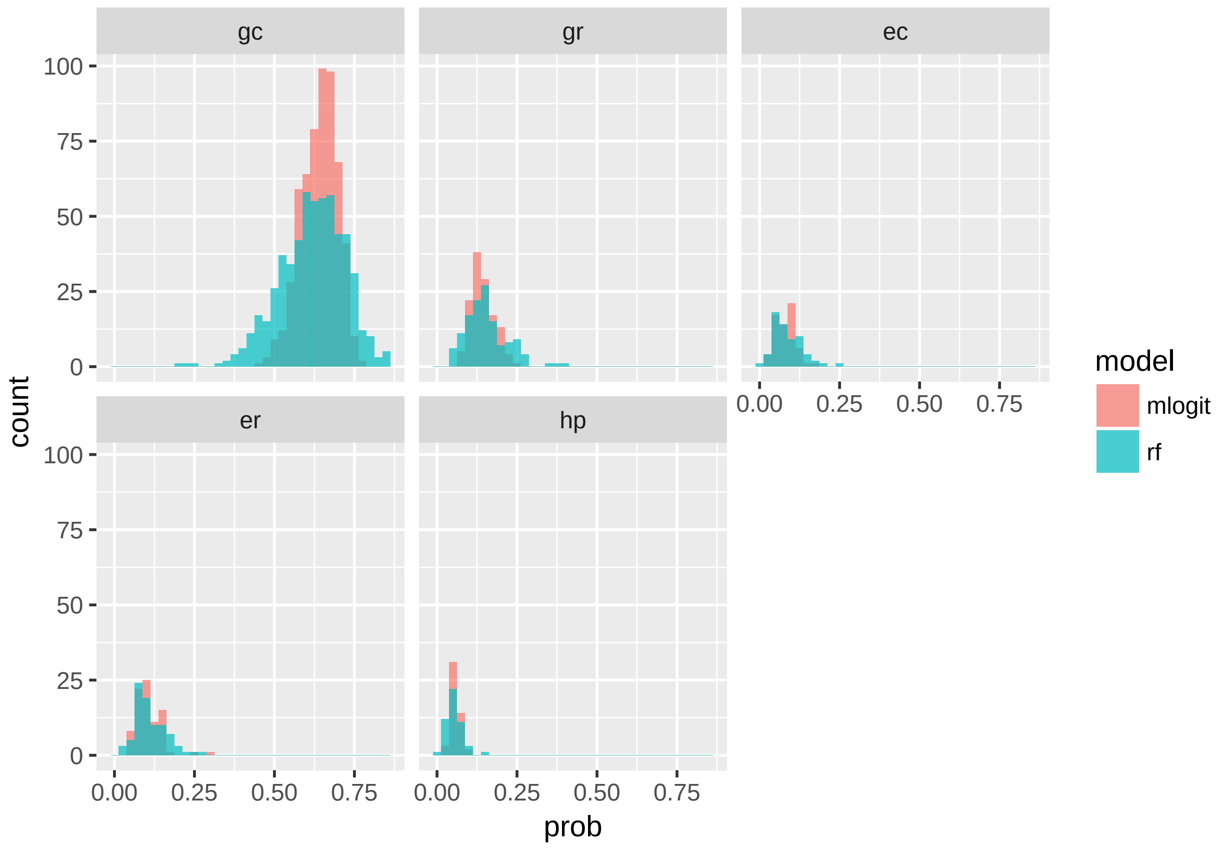
```
# Separately for each heating system
cpred %>% group_by(model,choice) %>%
  summarize(mean.prob = mean(prob)) %>%
  spread(model, mean.prob)
```

```
## # A tibble: 5 x 3
##   choice mlogit     rf
##   <fct>   <dbl>  <dbl>
## 1 gc     0.639  0.618
## 2 gr     0.143  0.154
## 3 ec     0.0803 0.0876
## 4 er     0.107  0.110
## 5 hp     0.0575 0.0551
```

```
# Graphical analysis
ggplot(cpred, aes(x=prob, group=model, fill=model)) +
  geom_histogram(alpha=0.7,binwidth = 0.025, position = "identity")
```



```
# Separately for each heating system
ggplot(cpred, aes(x=prob, group=model, fill=model)) +
  geom_histogram(alpha=0.7,binwidth = 0.025, position = "identity") +
  facet_wrap(~choice)
```



i) (Optional) Let us take a look at the variable importance of the random forest model.

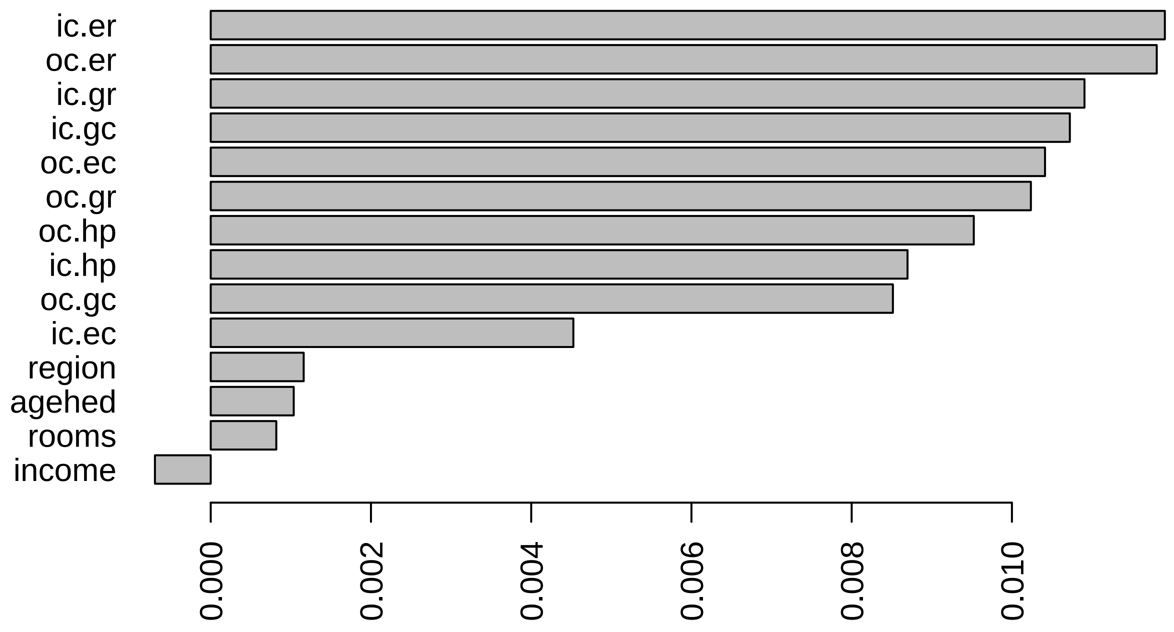
```
rf = ranger(depvar ~ . - idcase, train, importance = "permutation",probability = TRUE)
rf
```

```
## Ranger result
##
## Call:
##  ranger(depvar ~ . - idcase, train, importance = "permutation",      probability = TRUE)
##
## Type:                             Probability estimation
## Number of trees:                  500
## Sample size:                      600
## Number of independent variables:  14
## Mtry:                             3
## Target node size:                 10
## Variable importance mode:         permutation
## Splitrule:                        gini
## OOB prediction error (Brier s.):  0.3875945
```

```
# Show variable importance
imp.rf =  sort(importance(rf),decreasing = TRUE)
imp.rf
```

```
##        ic.er        oc.er        ic.gr        ic.gc        oc.ec
## 0.0119087770 0.0118073354 0.0109056233 0.0107227359 0.0104130176
##        oc.gr        oc.hp        ic.hp        oc.gc        ic.ec
## 0.0102366470 0.0095238529 0.0086969223 0.0085144663 0.0045262271
##        region        agehed         rooms        income
##  0.0011601413  0.0010359868  0.0008185064 -0.0006961809
```

```
# Show a plot
par(las=2) # make label text perpendicular to axis
barplot(rev(imp.rf), horiz=TRUE)
```


ic.er
oc.er
ic.gr
ic.gc
oc.ec
oc.gr
oc.hp
ic.hp
oc.gc
ic.ec
region
agehed
rooms
income