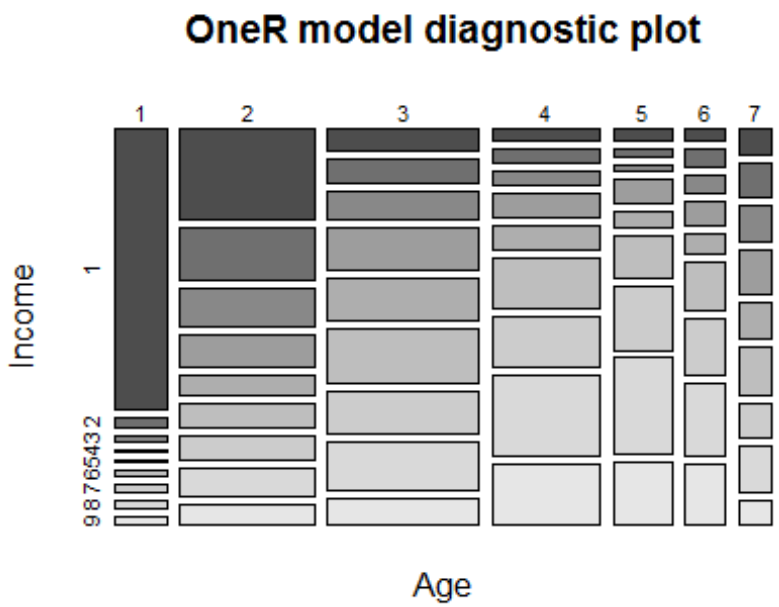
OneR_-_income_bracket_prediction_with_80_20.R

Holger

Mon Apr 17 12:33:13 2017

```
# http://sci2s.ugr.es/keel/dataset.php?cod=163
data <- read.csv("marketing1.dat")
data_names <- names(data)
data <- cbind(data[-ncol(data)], factor(data$Income))
names(data) <- data_names
set.seed(12) # for reproducibility
random <- sample(1:nrow(data), 0.8 * nrow(data))
data_train <- data[random, ]
data_test <- data[-random, ]

library(OneR)

## Warning: package 'OneR' was built under R version 3.3.2

data <- optbin(data_train)
model <- OneR(data, verbose = TRUE)

##
##     Attribute        Accuracy
## 1 * Age              28.2%
## 2   MaritalStatus    28.11%
## 3   Occupation       28.07%
## 4   HouseholdStatus  27.56%
## 5   DualIncome       27.04%
## 6   Education        25.98%
## 7   HouseholdMembers 22.51%
## 8   Under18          20.69%
## 9   TypeOfHome       19.36%
## 10  EthnicClass      19.29%
## 11  Sex              18.07%
## 12  Language         17.82%
## 13  YearsInSf        17.75%
## ---
## Chosen attribute due to accuracy
## and ties method (if applicable): '*'

summary(model)

##
## Call:
## OneR(data = data, verbose = TRUE)
##
## Rules:
```

```
## If Age = 1 then Income = 1
## If Age = 2 then Income = 1
## If Age = 3 then Income = 6
## If Age = 4 then Income = 8
## If Age = 5 then Income = 8
## If Age = 6 then Income = 8
## If Age = 7 then Income = 6
##
## Accuracy:
## 1551 of 5500 instances classified correctly (28.2%)
##
## Contingency table:
##        Age
## Income     1     2     3     4     5     6     7  Sum
##    1   * 421 * 352    99    43    21    15    25  976
##    2      16   204   107    39    13    22    33  434
##    3       9   147   122    49    12    21    35  395
##    4       5   121   188    71    39    29    42  495
##    5       3    77   179    81    29    23    34  426
##    6      10    93 * 234   156    70    56 * 47  666
##    7      12    92   185   155   107    66    33  650
##    8      12   111   211 * 251 * 160 * 86    44  875
##    9      11    76   114   187   104    69    22  583
##    Sum   499  1273  1439  1032   555  387  315 5500
## ---
## Maximum in each column: '*'
##
## Pearson's Chi-squared test:
## X-squared = 2671.2, df = 48, p-value < 2.2e-16

plot(model)
```



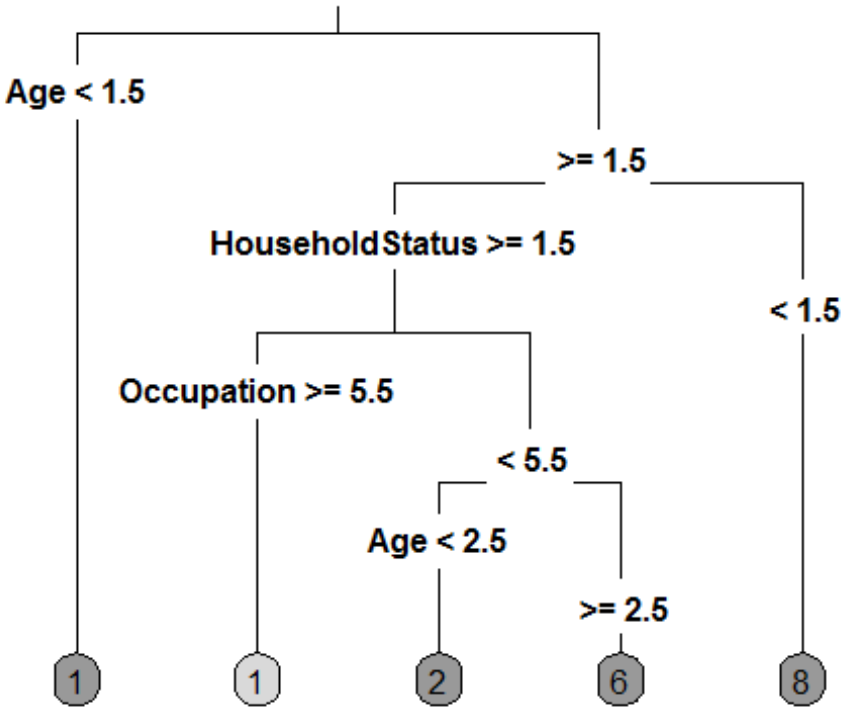

```
prediction <- predict(model, data_test)
eval_model(prediction, data_test)

##
## Confusion matrix (absolute):
##            Actual
## Prediction    1    2    3    4    5    6    7    8    9  Sum
##          1  232   45   46   32   33   27   19   27   24  485
##          2    0    0    0    0    0    0    0    0    0    0
##          3    0    0    0    0    0    0    0    0    0    0
##          4    0    0    0    0    0    0    0    0    0    0
##          5    0    0    0    0    0    0    0    0    0    0
##          6   31   30   44   44   41   66   44   57   50  407
##          7    0    0    0    0    0    0    0    0    0    0
##          8   16   20   20   47   27   87   71  110   86  484
##          9    0    0    0    0    0    0    0    0    0    0
##          Sum 279   95  110  123  101  180  134  194  160 1376
##
## Confusion matrix (relative):
##            Actual
## Prediction    1    2    3    4    5    6    7    8    9  Sum
##          1  0.17 0.03 0.03 0.02 0.02 0.02 0.01 0.02 0.02 0.35
##          2  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##          3  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##          4  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##          5  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##          6  0.02 0.02 0.03 0.03 0.03 0.05 0.03 0.04 0.04 0.30
```

```
##         7   0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##         8   0.01 0.01 0.01 0.03 0.02 0.06 0.05 0.08 0.06 0.35
##         9   0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##         Sum 0.20 0.07 0.08 0.09 0.07 0.13 0.10 0.14 0.12 1.00
##
## Accuracy:
## 0.2965 (408/1376)
##
## Error rate:
## 0.7035 (968/1376)
##
## Error rate reduction (vs. base rate):
## 0.1176 (p-value < 2.2e-16)
```

```
library(rpart)
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 3.3.2
```

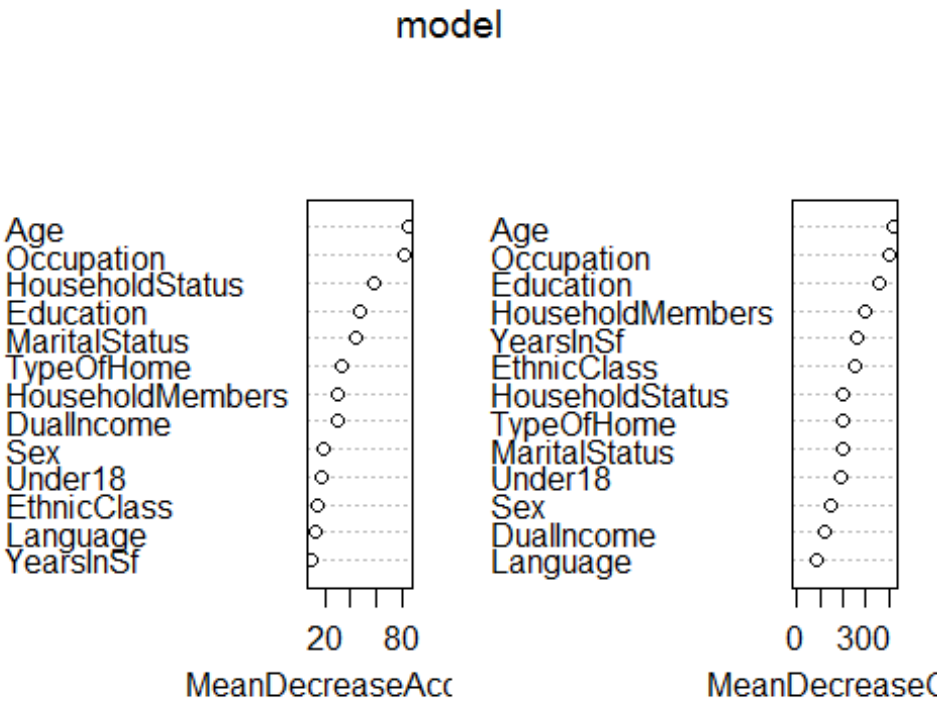
```
model <- rpart(Income ~., data = data_train)
rpart.plot(model, type = 3, extra = 0, box.palette = "Grays")
```




```
prediction <- predict(model, data_test, type = "class")
eval_model(prediction, data_test)
```

```
##
## Confusion matrix (absolute):
##           Actual
```

```
## Prediction    1    2    3    4    5    6    7    8    9  Sum
##          1  201   36   22   13   16   12    8   15   12  335
##          2   43   25   32   22   17   12   10   14    6  181
##          3    0    0    0    0    0    0    0    0    0    0
##          4    0    0    0    0    0    0    0    0    0    0
##          5    0    0    0    0    0    0    0    0    0    0
##          6   18   24   40   50   42   68   32   33   22  329
##          7    0    0    0    0    0    0    0    0    0    0
##          8   17   10   16   38   26   88   84  132  120  531
##          9    0    0    0    0    0    0    0    0    0    0
##          Sum  279   95  110  123  101  180  134  194  160 1376
##
## Confusion matrix (relative):
##            Actual
## Prediction    1    2    3    4    5    6    7    8    9  Sum
##          1  0.15 0.03 0.02 0.01 0.01 0.01 0.01 0.01 0.01 0.24
##          2  0.03 0.02 0.02 0.02 0.01 0.01 0.01 0.01 0.00 0.13
##          3  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##          4  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##          5  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##          6  0.01 0.02 0.03 0.04 0.03 0.05 0.02 0.02 0.02 0.24
##          7  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##          8  0.01 0.01 0.01 0.03 0.02 0.06 0.06 0.10 0.09 0.39
##          9  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##          Sum 0.20 0.07 0.08 0.09 0.07 0.13 0.10 0.14 0.12 1.00
##
## Accuracy:
## 0.3096 (426/1376)
##
## Error rate:
## 0.6904 (950/1376)
##
## Error rate reduction (vs. base rate):
## 0.134 (p-value < 2.2e-16)

library(randomForest)

## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

set.seed(4543)
model <- randomForest(Income ~., data = data_train, importance = TRUE)
varImpPlot(model)
```




```
prediction <- predict(model, data_test)
eval_model(prediction, data_test)
```

```
##
## Confusion matrix (absolute):
##            Actual
## Prediction    1    2    3    4    5    6    7    8    9  Sum
##        1    222   33   24   16   17   11    9   18   18  368
##        2     21   16   15   18   10   11    2    2    2   97
##        3     10   10   10   14    8    5    2    3    2   64
##        4      6   14   25   22   12   21    6    8    4  118
##        5      1    4    9   10    4    9    5    3    1   46
##        6      5    5   15   12   19   38   21   19   14  148
##        7      7    4    6   16   13   30   25   23    5  129
##        8      4    7    5   11   13   45   50   87   65  287
##        9      3    2    1    4    5   10   14   31   49  119
##        Sum  279   95  110  123  101  180  134  194  160 1376
##
## Confusion matrix (relative):
##            Actual
## Prediction    1    2    3    4    5    6    7    8    9  Sum
##        1   0.16 0.02 0.02 0.01 0.01 0.01 0.01 0.01 0.01 0.27
##        2   0.02 0.01 0.01 0.01 0.01 0.01 0.00 0.00 0.00 0.07
##        3   0.01 0.01 0.01 0.01 0.01 0.00 0.00 0.00 0.00 0.05
##        4   0.00 0.01 0.02 0.02 0.01 0.02 0.00 0.01 0.00 0.09
##        5   0.00 0.00 0.01 0.01 0.00 0.01 0.00 0.00 0.00 0.03
##        6   0.00 0.00 0.01 0.01 0.01 0.03 0.02 0.01 0.01 0.11
```

```
##        7   0.01 0.00 0.00 0.01 0.01 0.02 0.02 0.02 0.00 0.09
##        8   0.00 0.01 0.00 0.01 0.01 0.03 0.04 0.06 0.05 0.21
##        9   0.00 0.00 0.00 0.00 0.00 0.01 0.01 0.02 0.04 0.09
##        Sum 0.20 0.07 0.08 0.09 0.07 0.13 0.10 0.14 0.12 1.00
##
## Accuracy:
## 0.3438 (473/1376)
##
## Error rate:
## 0.6562 (903/1376)
##
## Error rate reduction (vs. base rate):
## 0.1768 (p-value < 2.2e-16)
```