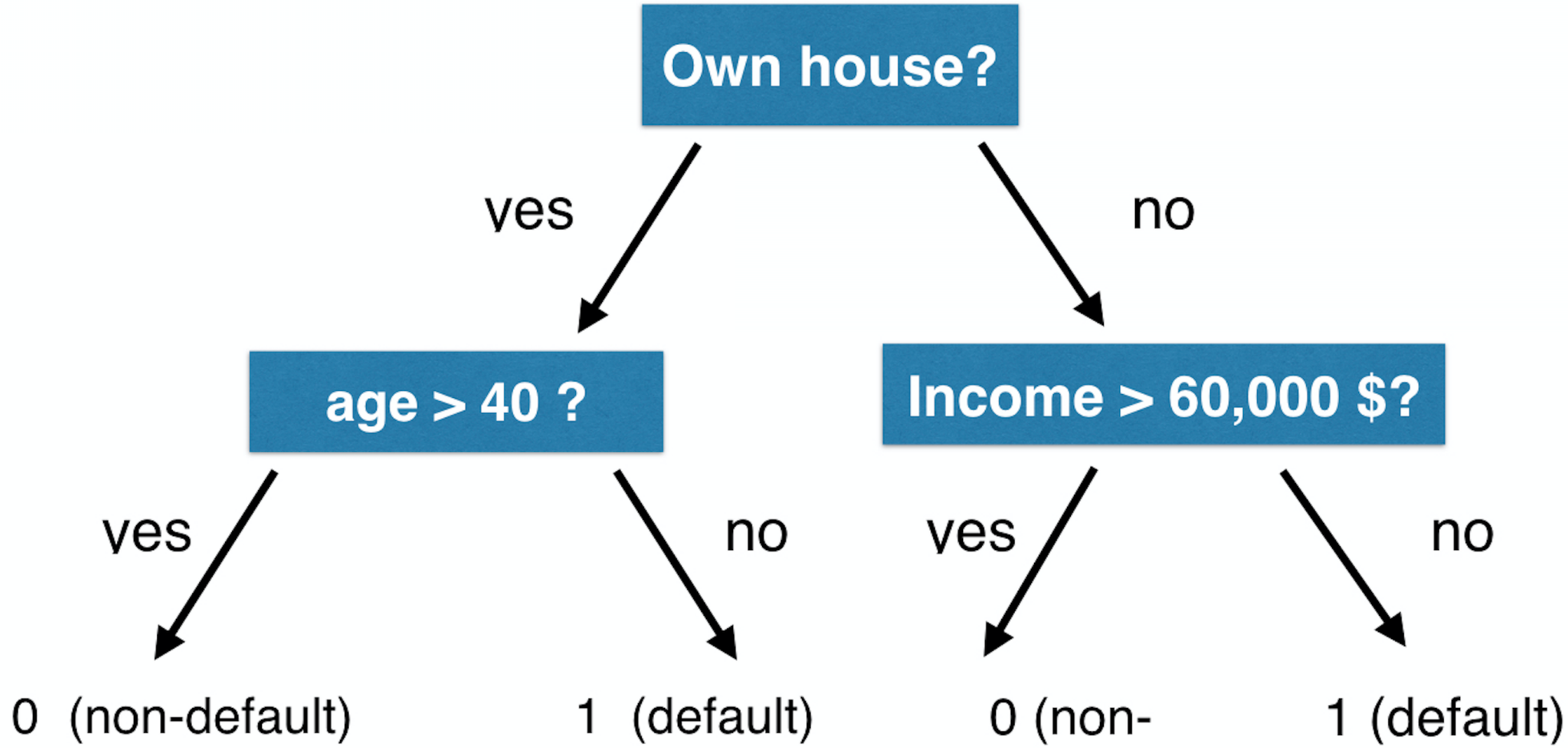
What is a decision tree?

CREDIT RISK MODELING IN R

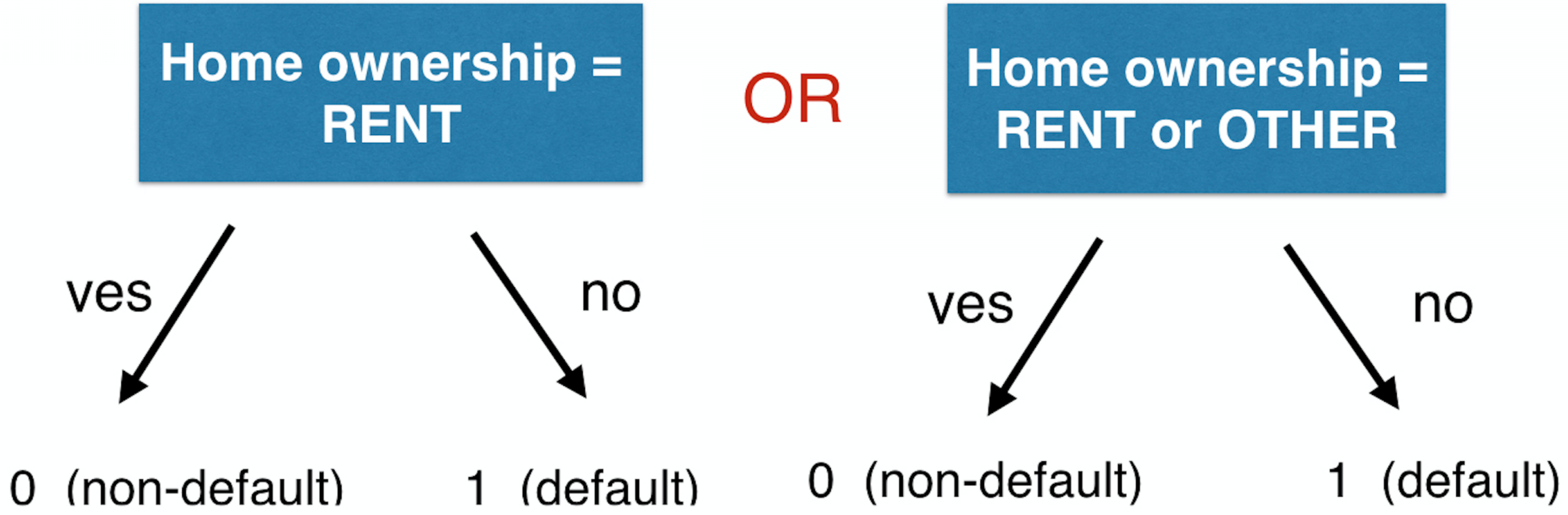
Lore Dirick

Manager of Data Science Curriculum at Flatiron School

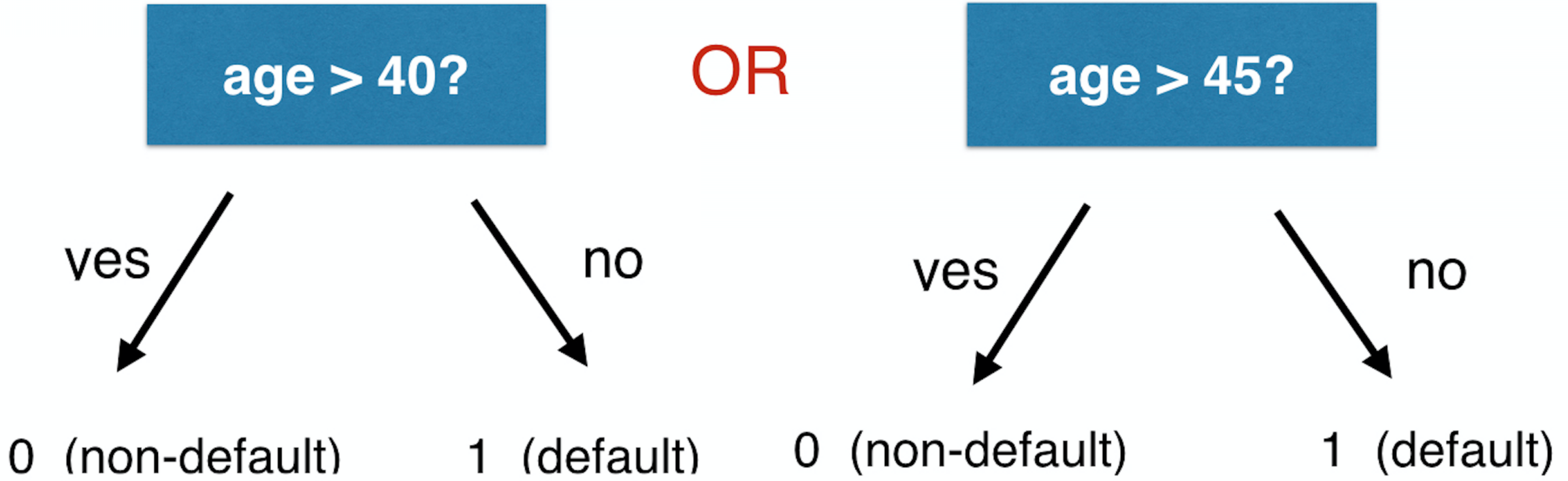
Decision tree example



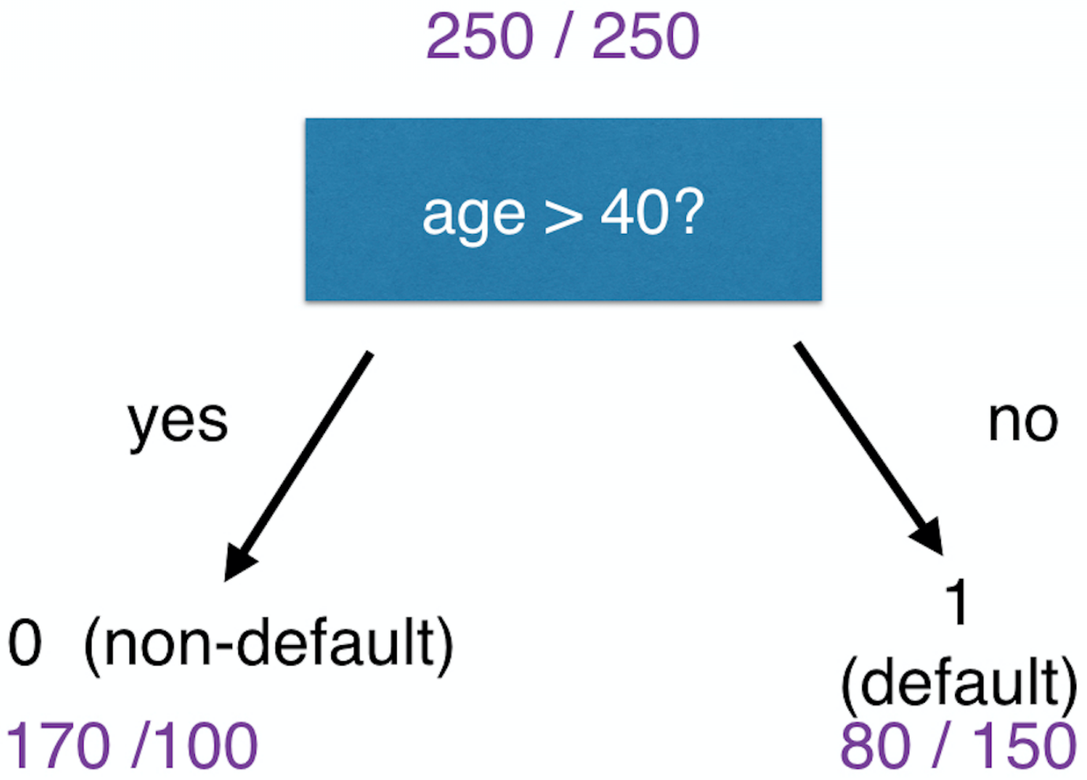
How to make splitting decision?



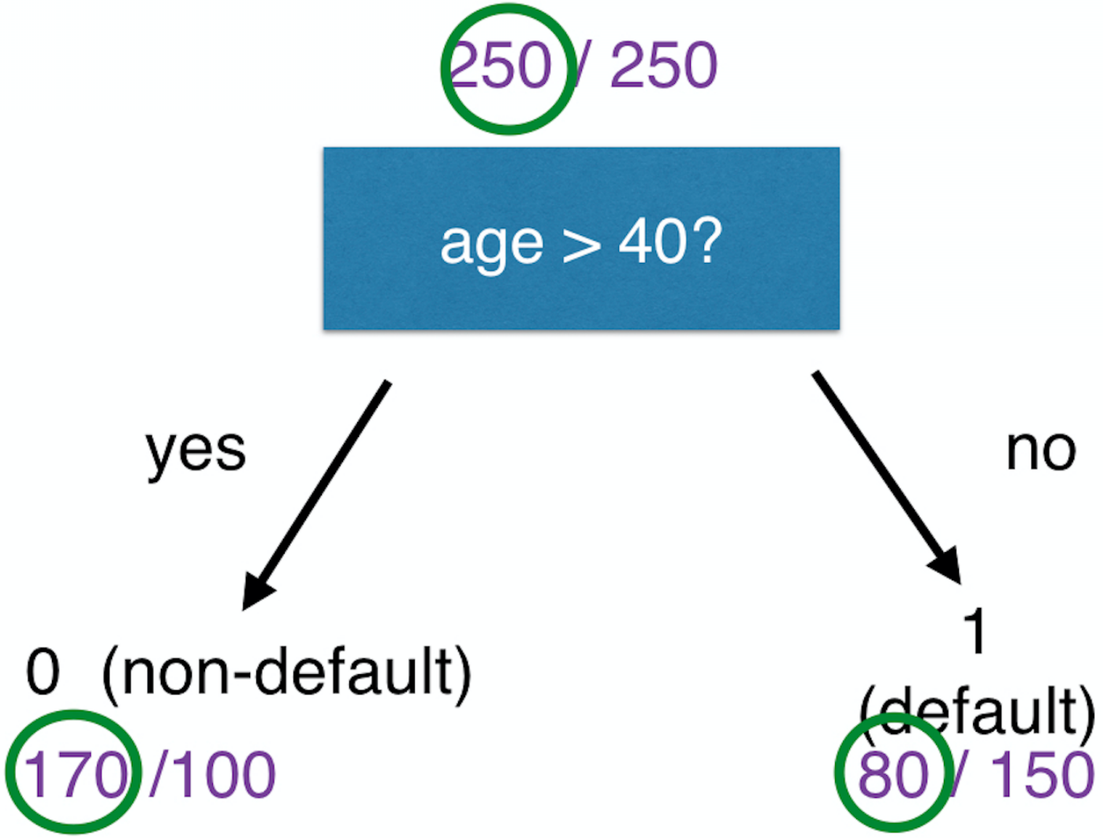
How to make splitting decision?



Example

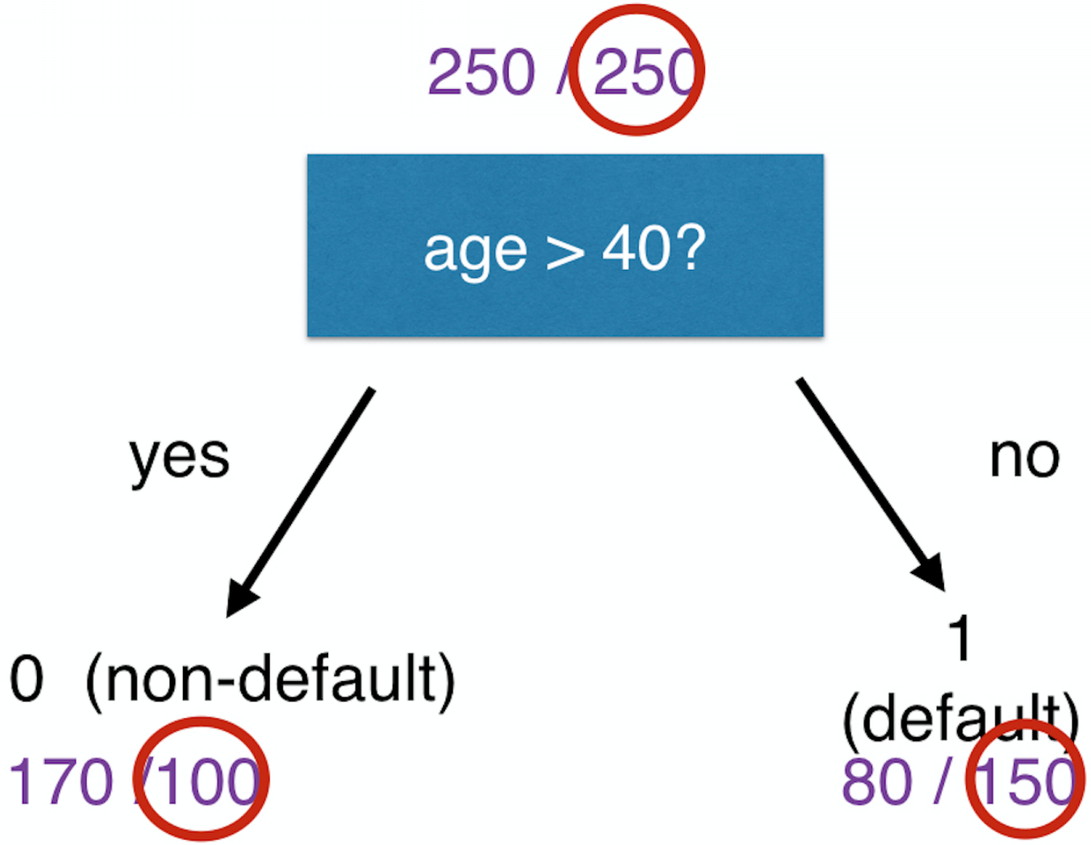


Example



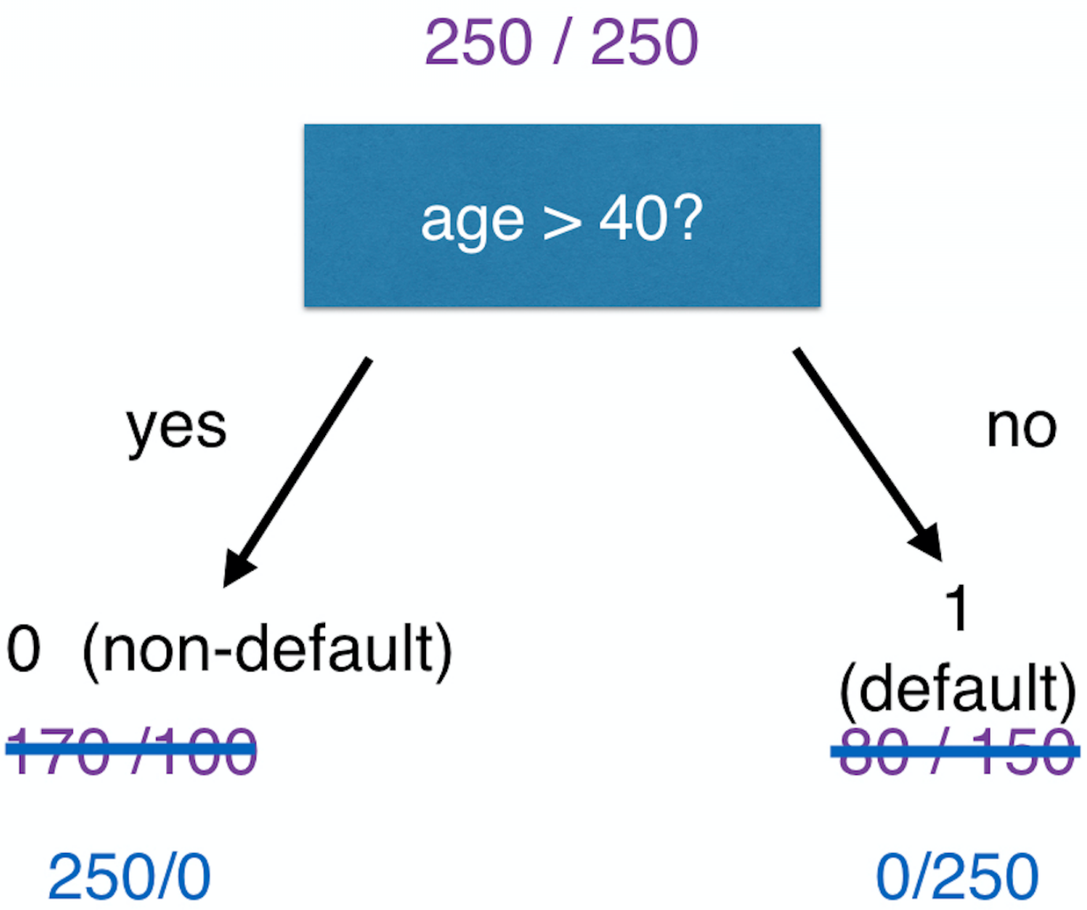
- Actual non-defaults in this node using this split

Example



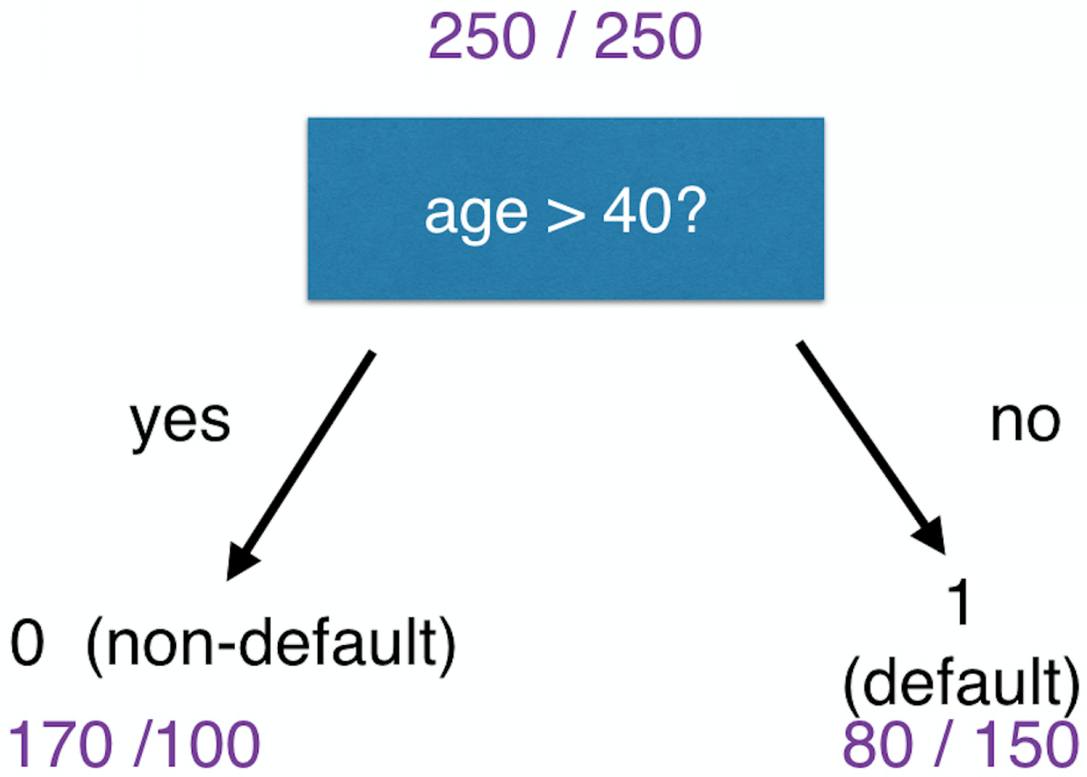
- Actual defaults in this node using this split

Example



Ideal scenario

Example



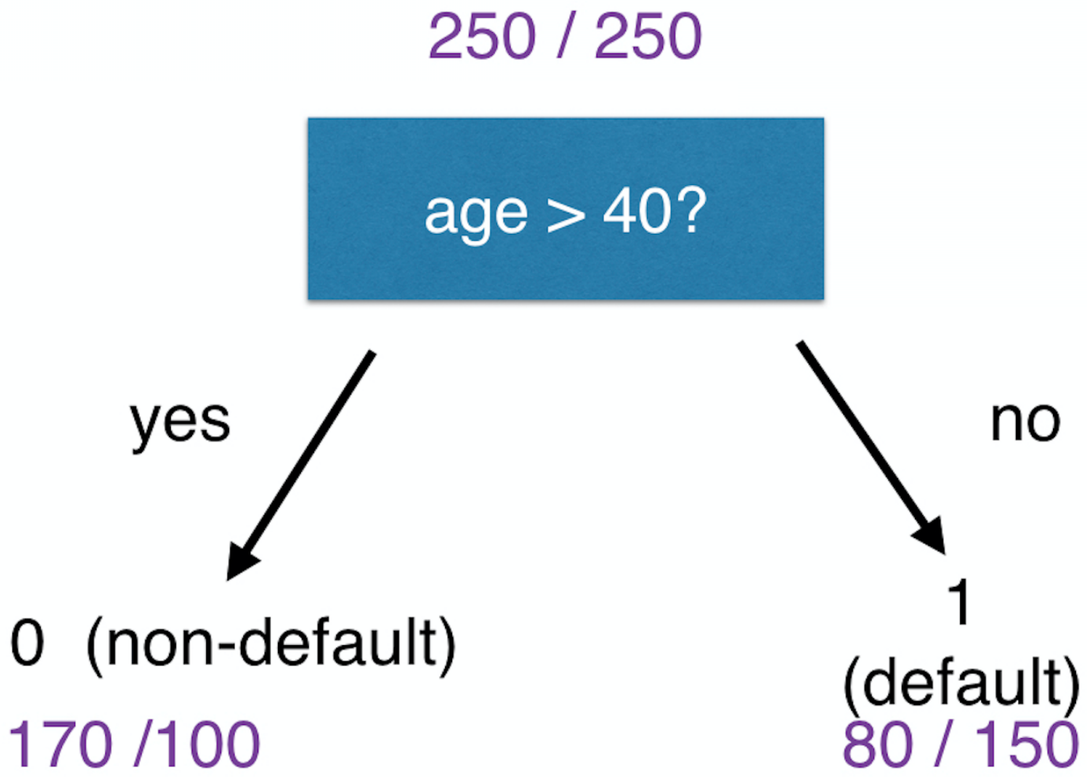
Gini = 2*prop(default)*prop(non-default)

Gini_R = 2*(250/500)*(250/500) = 0.5

Gini_N2 = 2*(80/230)*(150/230) = 0.4536

Gini_N1 = 2*(170/270)*(100/270) = 0.4664

Example



Gain

= Gini_R-prop(cases in N1)*Gini_N1 - prop(cases in N2) * Gini_N1

= 0.5 - 270/500 * 0.4664

= 0.039488

- **Maximum gain**

Let's practice!

CREDIT RISK MODELING IN R

Building decision trees using the rpart()-package

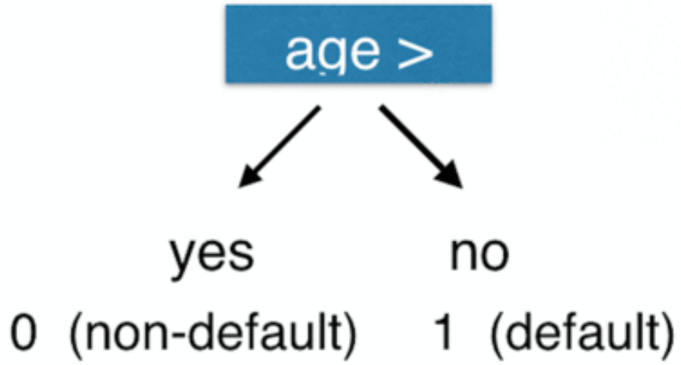
CREDIT RISK MODELING IN R

Lore Dirick

Manager of Data Science Curriculum at Flatiron School

datacamp

Imagine...



rpart() package! But...

- Hard building nice decision tree for credit risk data
- Main reason: unbalanced data

```
fit_default <- rpart(loan_status ~ ., method = "class",
                     data = training_set)
plot(fit_default)
```

```
Error in plot.rpart(fit_default) : fit is not a tree, just a root
```

Three techniques to overcome unbalance

- Undersampling or oversampling
  - Accuracy issue will disappear
  - Only training set
- Changing the prior probabilities
- Including a loss matrix

Validate model to see what is best!

Let's practice!

CREDIT RISK MODELING IN R

Pruning the decision tree

CREDIT RISK MODELING IN R

Lore Dirick

Manager of Data Science Curriculum at Flatiron School

datacamp

Problems with large decision trees

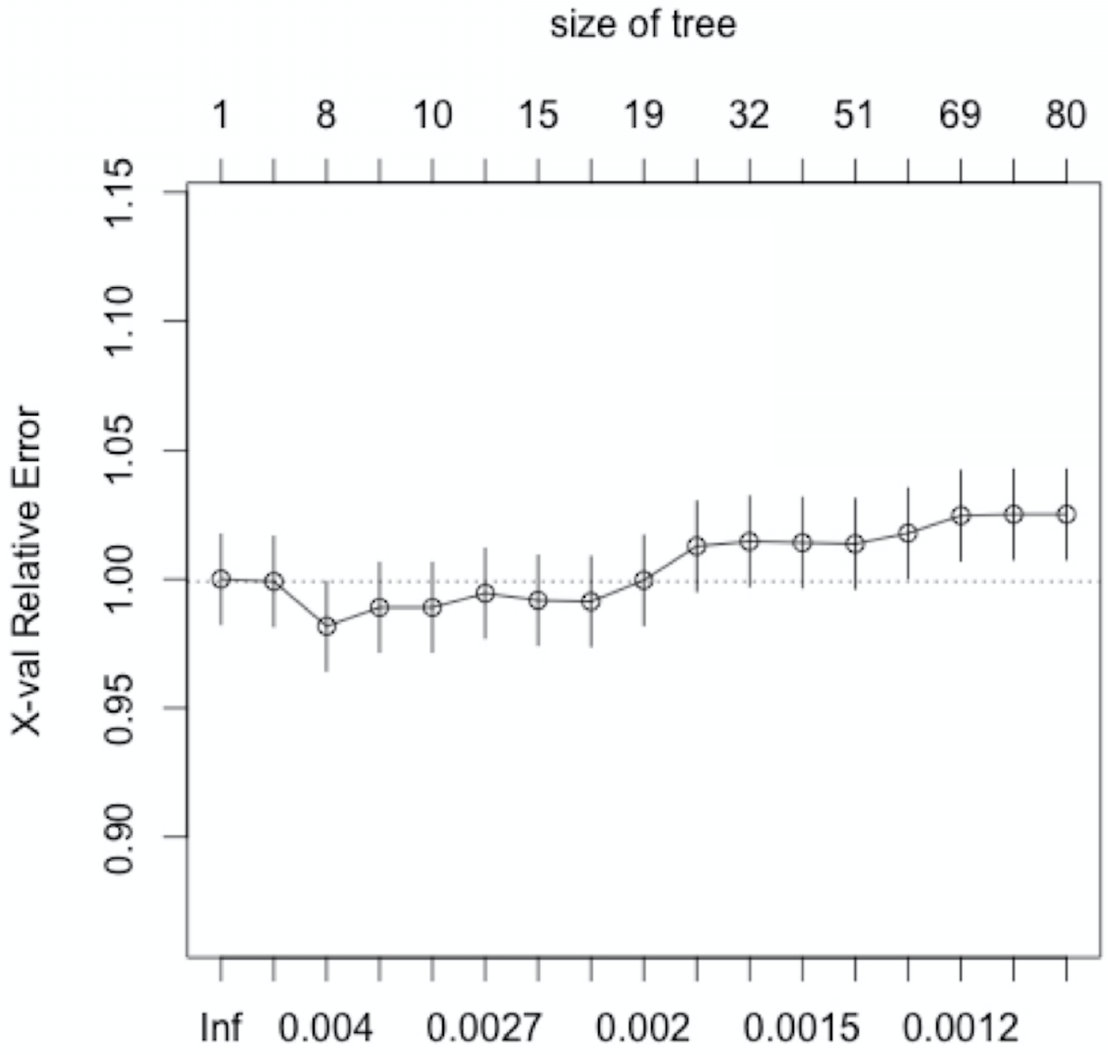
- Too complex: not clear anymore
- Overfitting when applying to test set
- Solution: use `printcp()` , `plotcp()` for pruning purposes

Printcp and tree_undersample

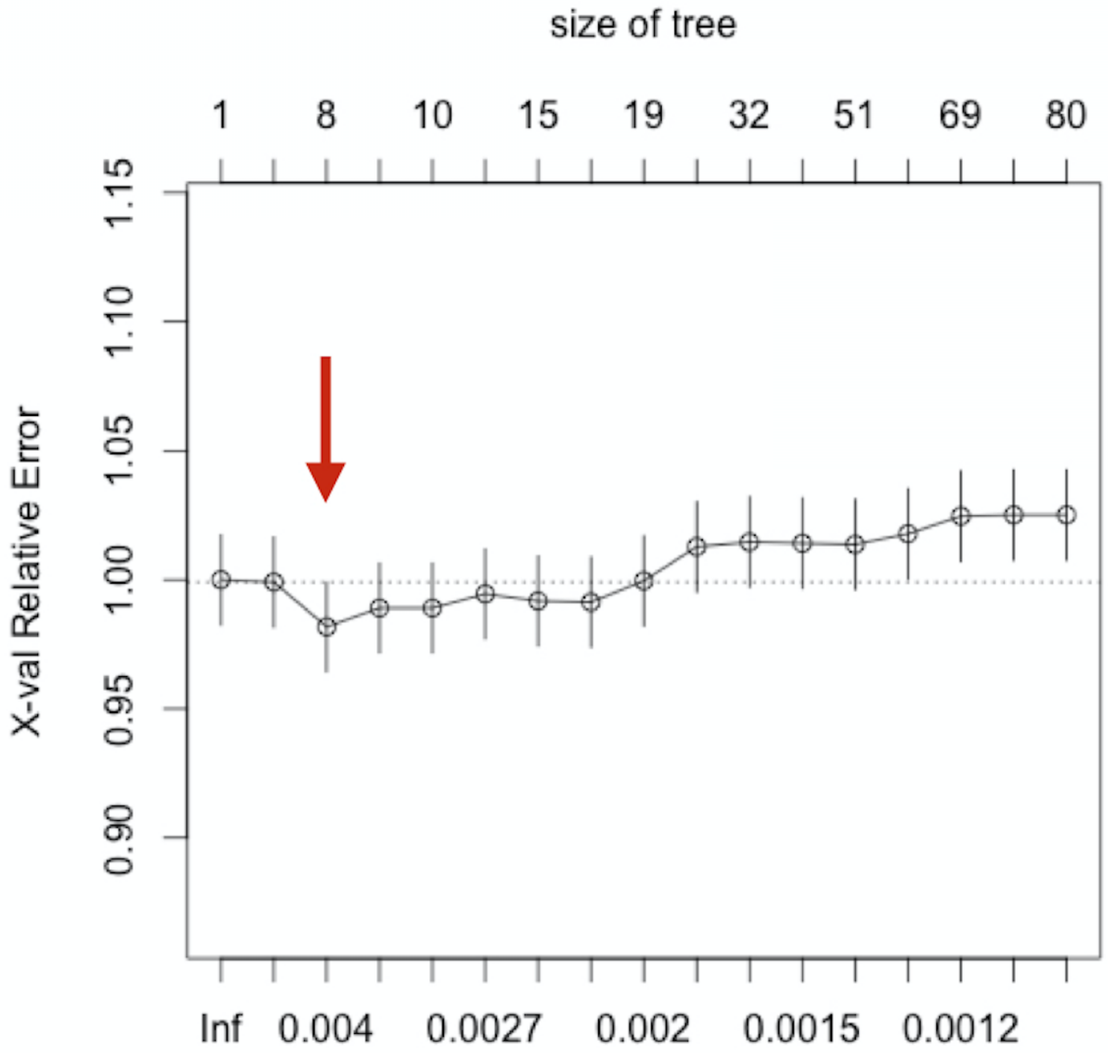
```
printcp(tree_undersample)
```

```
Classification tree:
rpart(formula = loan_status ~ ., data = undersampled_training_set, method = "class",
 control = rpart.control(cp = 0.001))
Variables actually used in tree construction:
age    annual_inc     emp_cat     grade    home_ownership   ir_cat     loan_amnt
Root node error: 2190/6570 = 0.33333
n= 6570
        CP    nsplit  rel error   xerror      xstd
1  0.0059361      0    1.00000   1.00000   0.017447
2  0.0044140      4    0.97443   0.99909   0.017443
3  0.0036530      7    0.96119   0.98174   0.017366
4  0.0031963      8    0.95753   0.98904   0.017399
               ...
16 0.0010654     76    0.84247   1.02511   0.017554
17 0.0010000     79    0.83927   1.02511   0.017554
```

Plotcp and tree_undersample

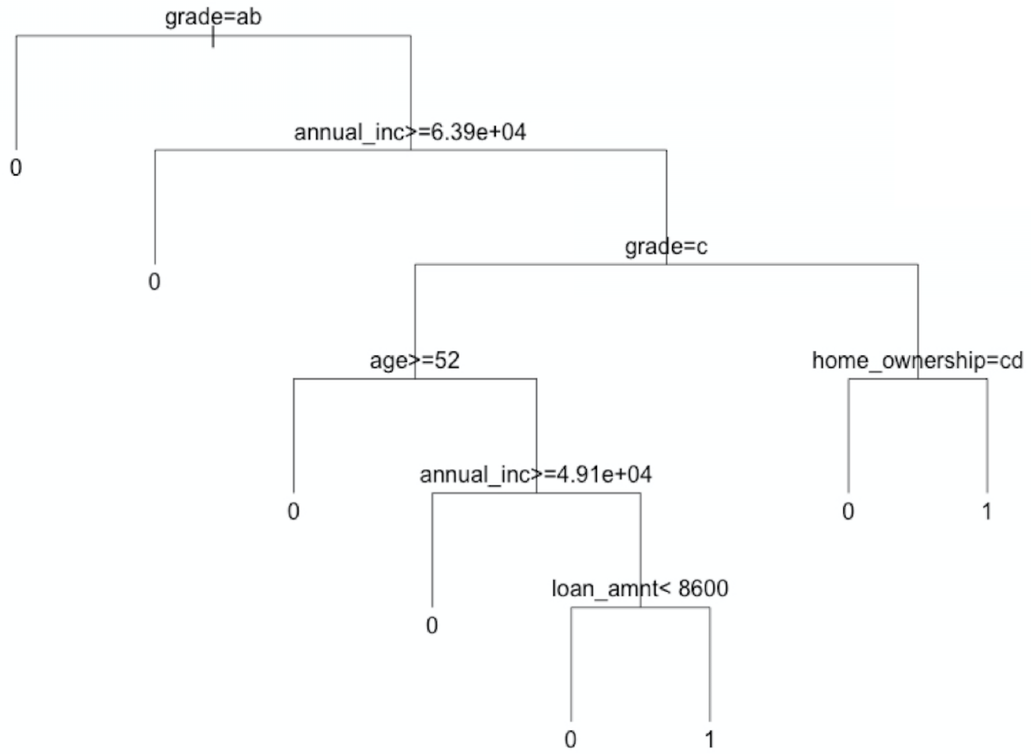


Plotcp and tree_undersample



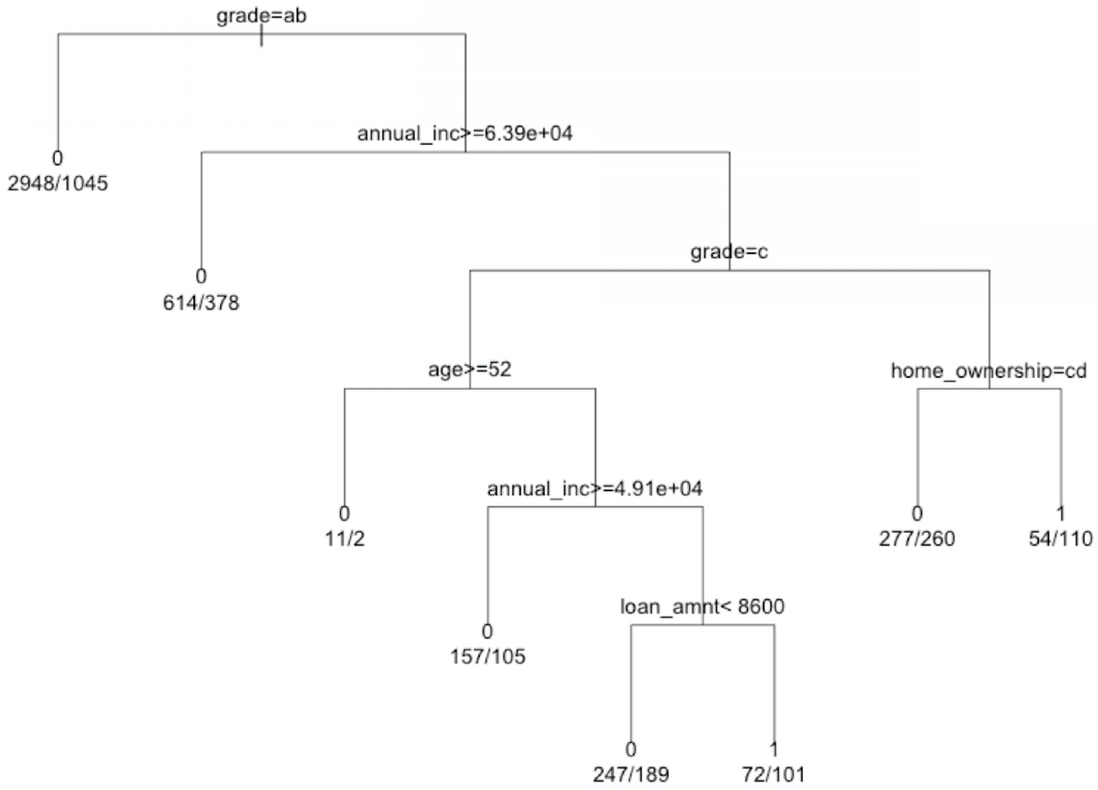
$$CP = 0.003653$$

Plot the pruned tree



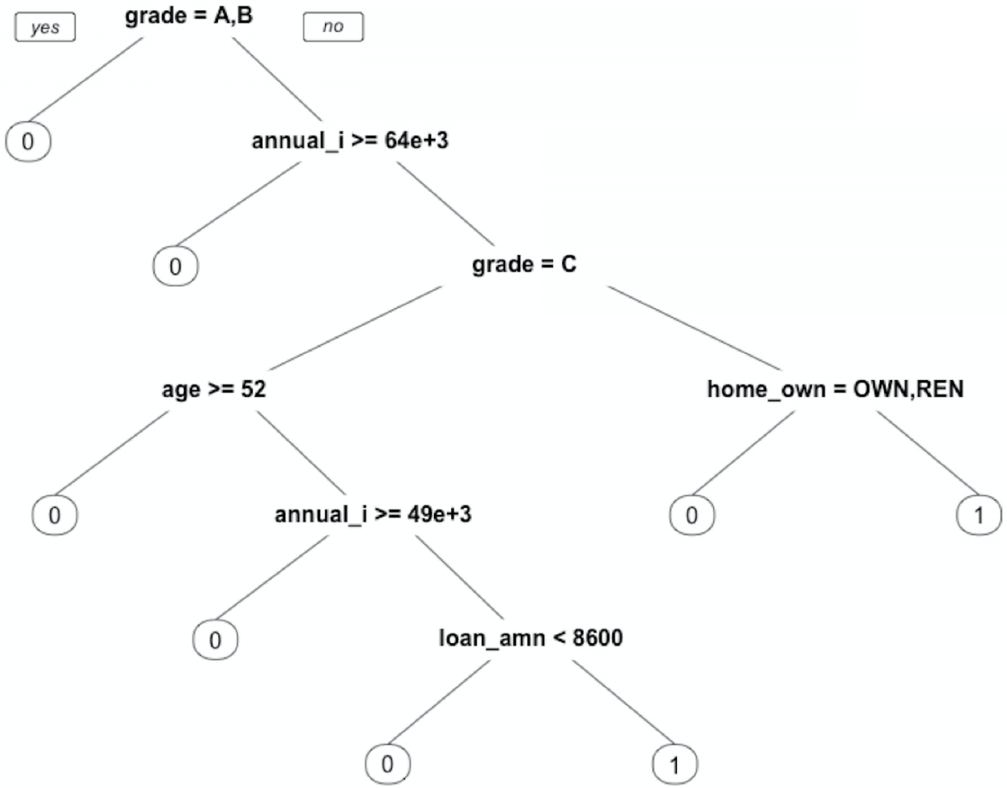

```
ptree_undersample=prune(tree_undersample,
                        cp = 0.003653)

plot(ptree_undersample,
     uniform=TRUE)

text(ptree_undersample)
```

Plot the pruned tree



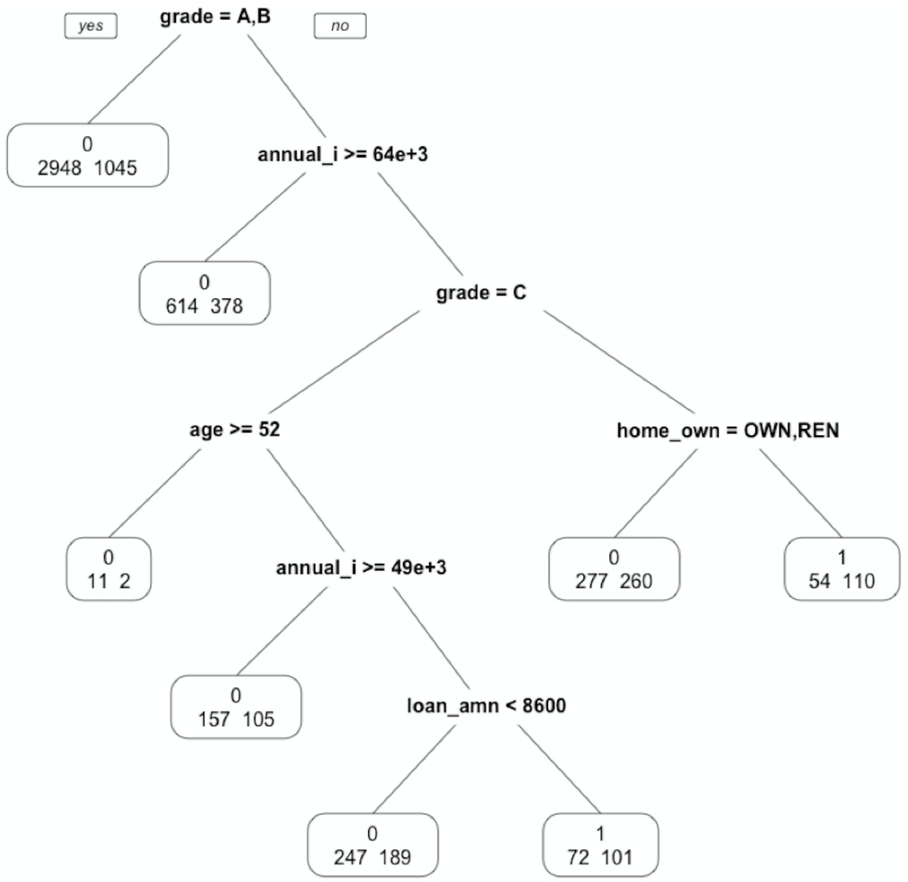

```
ptree_undersample=prune(tree_undersample,
                        cp = 0.003653)

plot(ptree_undersample,
     uniform=TRUE)

text(ptree_undersample,
     use.n=TRUE)
```

prp() in the rpart.plot-package




```
library(rpart.plot)
prp(ptree_undersample)
```

prp() in the part.plot-package




```
library(rpart.plot)
prp(ptree_undersample, extra = 1)
```

Let's practice!

CREDIT RISK MODELING IN R

Other tree options and the construction of confusion matrices

CREDIT RISK MODELING IN R

Lore Dirick

Manager of Data Science Curriculum at Flatiron School

Other interesting rpart() - arguments

- In `rpart()`
  - `weights` : include case weights
- In the control argument of `rpart()` (`rpart.control`)
  - `minsplit` : minimum number of observations for split attempt
  - `minbucket` : minimum number of observations in leaf node

```
pred_undersample_class = predict(ptree_undersample, newdata = test_set, type ="class")
```

```
1     2     3    ...   29073 29079 29084 29090 29091
0     0     0    ...     1     0     0     0     0
```

OR

```
pred_undersample = predict(ptree_undersample, newdata = test_set)
```

```
          0         1
1     0.7382920 0.2617080
2     0.5665138 0.4334862
3     0.5992366 0.4007634
          ...         ...
29084 0.7382920 0.2617080
29090 0.7382920 0.2617080
29091 0.7382920 0.2617080
```

Constructing a confusion matrix

```
table(test_set$loan_status, pred_undersample_class)
```

```
pred_undersample_class
       0    1
  0 8314  346
  1  964   73
```

Let's practice!

CREDIT RISK MODELING IN R