What is a decision tree?

CREDIT RISK MODELING IN R

Lore Dirick
Manager of Data Science Curriculum at Flatiron School
Decision tree example

Own house?
- yes
  - age > 40 ?
    - yes
      - 0 (non-default)
    - no
      - 1 (default)
- no
  - Income > 60,000 $?
    - yes
      - 0 (non-default)
    - no
      - 1 (default)
How to make splitting decision?

Home ownership = RENT

- yes
  - 0 (non-default)
- no
  - 1 (default)

OR

Home ownership = RENT or OTHER

- yes
  - 0 (non-default)
- no
  - 1 (default)
How to make splitting decision?

- **age > 40?**
  - **yes**
    - 0 (non-default)
  - **no**
    - 1 (default)

- **OR**

- **age > 45?**
  - **yes**
    - 0 (non-default)
  - **no**
    - 1 (default)
Example

250 / 250

age > 40?

Yes
0 (non-default)
170 / 100

No
1 (default)
80 / 150
Example

- Actual non-defaults in this node using this split

```
250 / 250
age > 40?

yes
0 (non-default)
170 / 100

no
1 (default)
80 / 150
```
Example

- Actual defaults in this node using this split

```plaintext
age > 40?

yes

0 (non-default)
170 / 100

no

1 (default)
80 / 150
```
Example

Ideal scenario

```
age > 40?

yes
0 (non-default)
170 / 100
250 / 0

no
1 (default)
80 / 150
0 / 250
```
Example

- **Age > 40?**
  - Yes: 0 (non-default), 170/100
  - No: 1 (default), 80/150

Gini $= 2 \times \text{prop}(\text{default}) \times \text{prop}(\text{non-default})$

- Gini$_R = 2 \times (250/500) \times (250/500) = 0.5$
- Gini$_N2 = 2 \times (80/230) \times (150/230) = 0.4536$
- Gini$_N1 = 2 \times (170/270) \times (100/270) = 0.4664$
Example

Gain

\[
\text{Gain} = \text{Gini}_R \cdot \text{prop(cases in N1)} \cdot \text{Gini}_N - \\
\text{prop(cases in N2)} \cdot \text{Gini}_N
\]

\[
= 0.5 - \frac{270}{500} \cdot 0.4664
\]

\[
= 0.039488
\]

- Maximum gain
Let’s practice!

CREDIT RISK MODELING IN R
Building decision trees using the \texttt{rpart()}-package

\textsc{Credit Risk Modeling in R}

Lore Dirick
Manager of Data Science Curriculum at Flatiron School
Imagine...

- age >
  - yes
  - no

0 (non-default) 1 (default)
rpart() package! But...

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

```r
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
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

<table>
<thead>
<tr>
<th>CP</th>
<th>nsplit</th>
<th>rel error</th>
<th>xerror</th>
<th>xstd</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0059361</td>
<td>0.0044140</td>
<td>0.0036530</td>
<td>0.0031963</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1.00000</td>
<td>1.00000</td>
<td>0.017447</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>0.97443</td>
<td>0.99909</td>
<td>0.017443</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>0.96119</td>
<td>0.98174</td>
<td>0.017366</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0.95753</td>
<td>0.98904</td>
<td>0.017399</td>
</tr>
<tr>
<td>16</td>
<td>76</td>
<td>0.84247</td>
<td>1.02511</td>
<td>0.017554</td>
</tr>
<tr>
<td>17</td>
<td>79</td>
<td>0.83927</td>
<td>1.02511</td>
<td>0.017554</td>
</tr>
</tbody>
</table>
Plotcp and tree undersample
Plotcp and tree_undersample

CP = 0.003653
Plot the pruned tree

\[
ptree_{\text{undersample}} \leftarrow \text{prune}(\text{tree}_{\text{undersample}}, \quad \text{cp} = 0.003653)
\]

plot(\text{ptree}_{\text{undersample}}, \quad \text{uniform}=\text{TRUE})

text(\text{ptree}_{\text{undersample}})
Plot the pruned tree

```r
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()`
  - **weigths**: 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")

OR

pred_undersample = predict(ptree_undersample, newdata = test_set)
Constructing a confusion matrix

table(test_set$loan_status, pred_undersample_class)

<table>
<thead>
<tr>
<th>pred_undersample_class</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>8314</td>
<td>346</td>
</tr>
<tr>
<td>1</td>
<td>964</td>
<td>73</td>
</tr>
</tbody>
</table>
Let's practice!

CREDIT RISK MODELING IN R