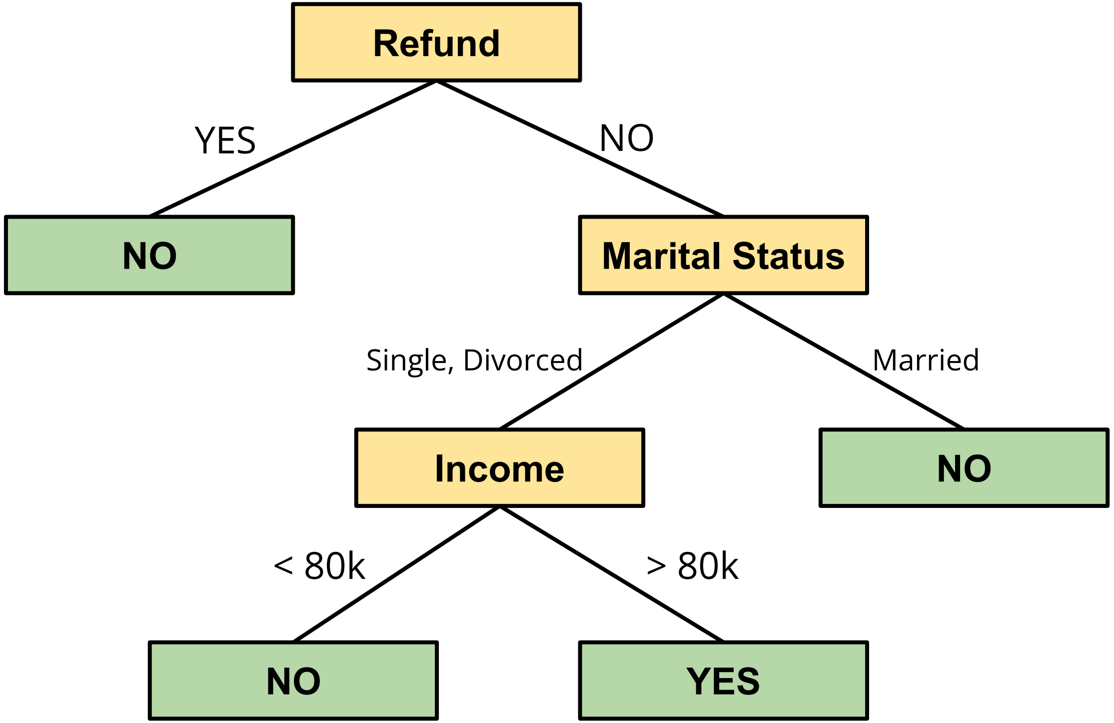
Decision Trees

Boston University CS 506 - Lance Galletti

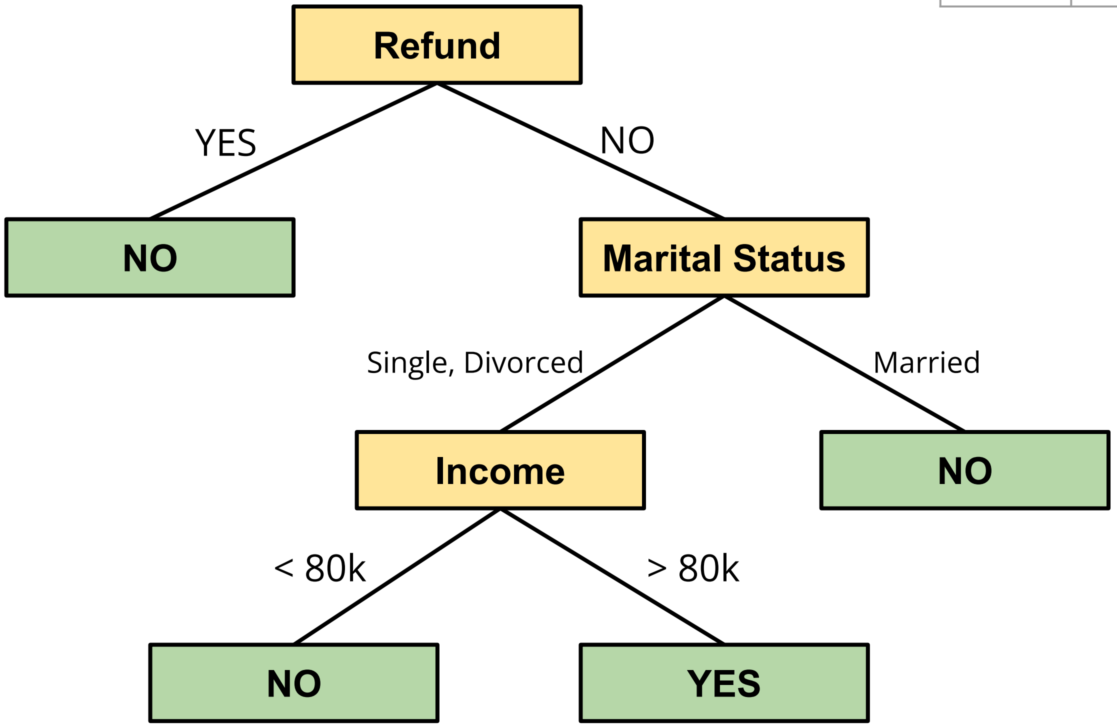
Refund	Marital Status	Income	Class
Yes	Single	125k	No
No	Married	100k	No
No	Single	70k	No
Yes	Married	120k	No
No	Divorced	90k	Yes
No	Married	60k	No
Yes	Divorced	220k	No
No	Single	85k	Yes
No	Married	75k	No
No	Single	90k	Yes

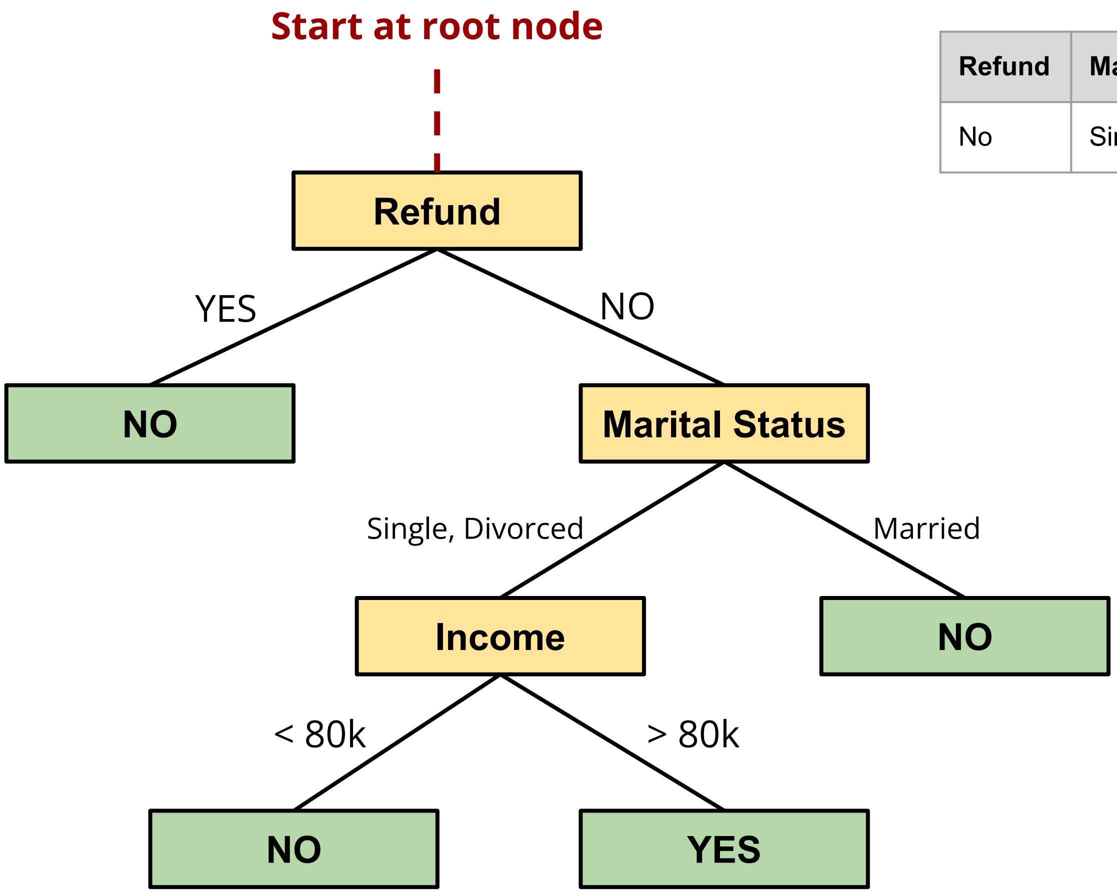
What a Decision Tree looks like



How it works

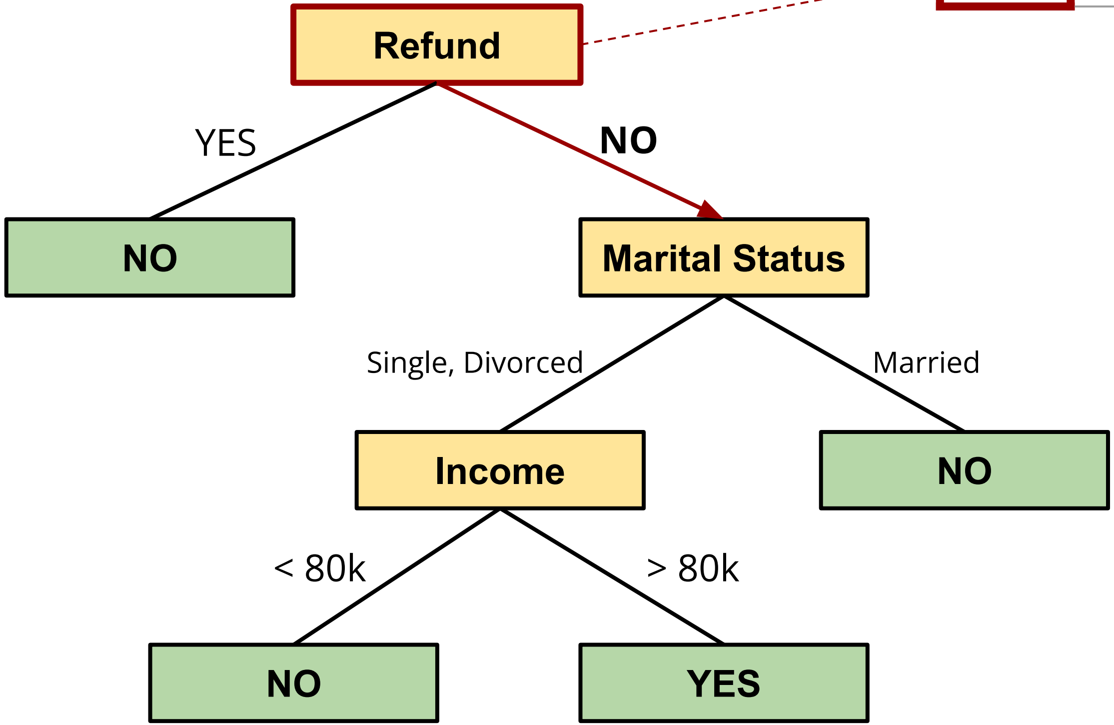
Refund	Marital Status	Income	Class
No	Single	70k	?



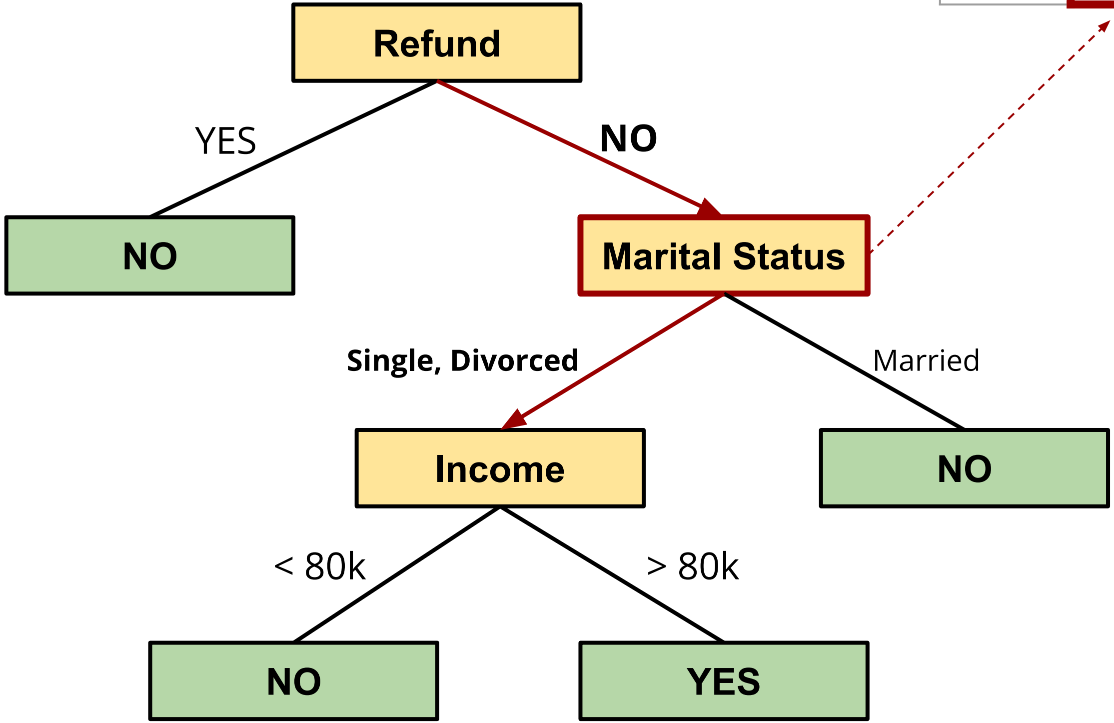


Refund	Marital Status	Income	Class
No	Single	70k	?

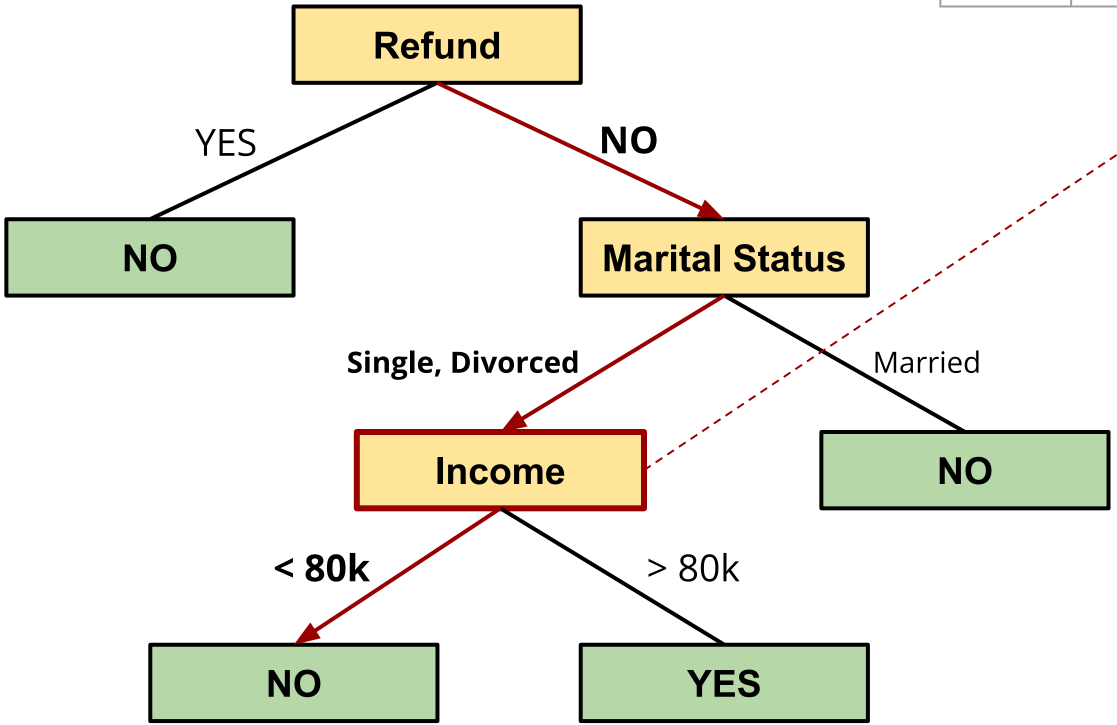
Refund	Marital Status	Income	Class
No	Single	70k	?



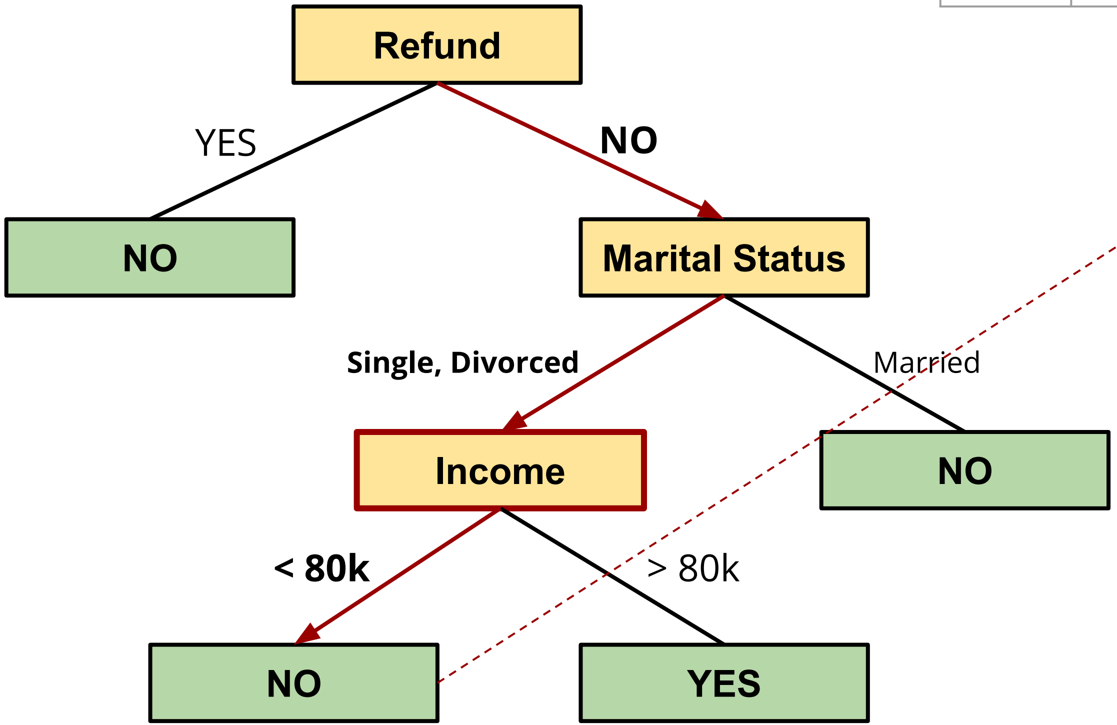
Refund	Marital Status	Income	Class
No	Single	70k	?



Refund	Marital Status	Income	Class
No	Single	70k	?



Refund	Marital Status	Income	Class
No	Single	70k	NO



worksheet

How do we learn it?

Refund	Marital Status	Income	Class
Yes	Single	125k	No
No	Married	100k	No
No	Single	70k	No
Yes	Married	120k	No
No	Divorced	90k	Yes
No	Married	60k	No
Yes	Divorced	220k	No
No	Single	85k	Yes
No	Married	75k	No
No	Single	90k	Yes

IF marital status == Married

Refund	Marital Status	Income	Class
Yes	Single	125k	No
No	Married	100k	No
No	Single	70k	No
Yes	Married	120k	No
No	Divorced	90k	Yes
No	Married	60k	No
Yes	Divorced	220k	No
No	Single	85k	Yes
No	Married	75k	No
No	Single	90k	Yes

IF marital status == Married

Refund	Marital Status	Income	Class
No	Married	100k	No
Yes	Married	120k	No
No	Married	60k	No
No	Married	75k	No

THEN class = NO

Refund	Marital Status	Income	Class
Yes	Single	125k	No
No	Married	100k	No
No	Single	70k	No
Yes	Married	120k	No
No	Divorced	90k	Yes
No	Married	60k	No
Yes	Divorced	220k	No
No	Single	85k	Yes
No	Married	75k	No
No	Single	90k	Yes

IF income < 60k

Refund	Marital Status	Income	Class
Yes	Single	125k	No
No	Married	100k	No
No	Single	70k	No
Yes	Married	120k	No
No	Divorced	90k	Yes
No	Married	60k	No
Yes	Divorced	220k	No
No	Single	85k	Yes
No	Married	75k	No
No	Single	90k	Yes

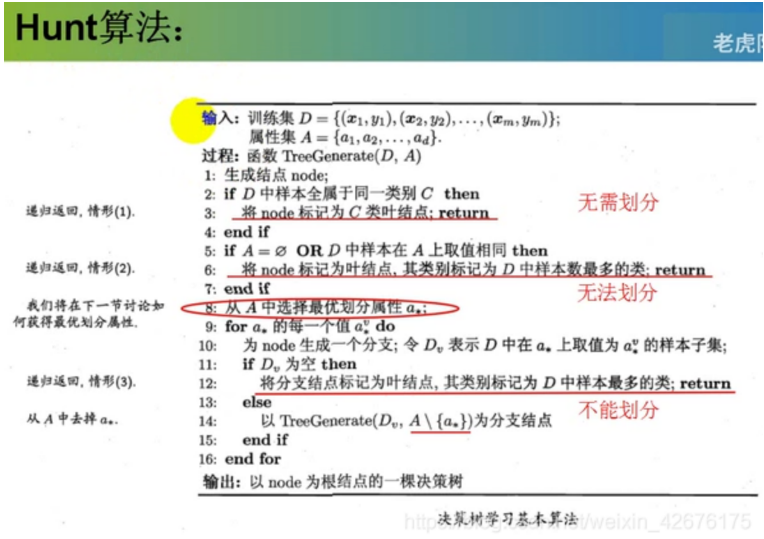
IF income < 60k

Refund	Marital Status	Income	Class

THEN ?

Hunt's Algorithm

- Recursive Algorithm
  - Repeatedly split the dataset based on attributes
- Base cases:
  - IF Split and all data points in the same class
    - Great! Predict that class
  - IF Split and no data points
    - No problem! Predict a reasonable default



Hunt算法：

输入：训练集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
属性集 $A = \{a_1, a_2, \ldots, a_d\}$.
过程：函数 TreeGenerate(D, A)
生成结点 node;
if D 中样本全属于同一类别 C then
将 node 标记为 C 类叶结点; return
end if
if $A = \varnothing$ OR D 中样本在 A 上取值相同 then
将 node 标记为叶结点，其类别标记为 D 中样本数最多的类; return
end if
从 A 中选择最优划分属性 a_*;
for a_* 的每一个值 a_*^v do
为 node 生成一个分支; 令 D_v 表示 D 中在 a_* 上取值为 a_*^v 的样本子集;
if D_v 为空 then
将分支结点标记为叶结点，其类别标记为 D 中样本最多的类; return
else
以 TreeGenerate(D_v, $A \setminus \{a_*\}$)为分支结点
end if
end for
输出：以 node 为根结点的一棵决策树

递归返回，情形(1).
递归返回，情形(2).
我们将在下一节讨论如何获得最优划分属性.
递归返回，情形(3).
从 A 中去掉 a_*.

无需划分
无法划分
不能划分

决策树学习基本算法

Hunt's Algorithm

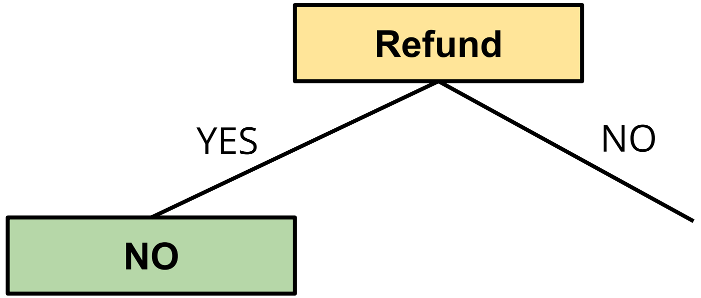
The recursion (IF split and data points belong to more than one class)

- Find the attribute (and best way to split that attribute) that best splits the data

Example

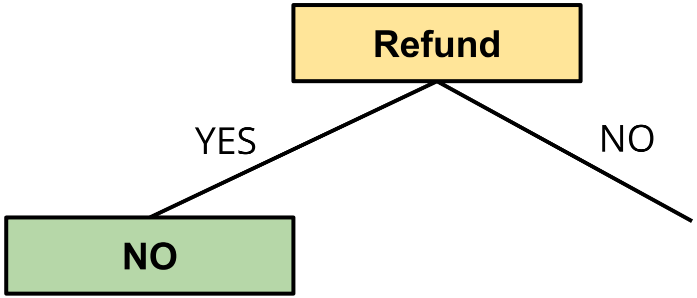
Refund	Marital Status	Income	Class
Yes	Single	125k	No
No	Married	100k	No
No	Single	70k	No
Yes	Married	120k	No
No	Divorced	90k	Yes
No	Married	60k	No
Yes	Divorced	220k	No
No	Single	85k	Yes
No	Married	75k	No
No	Single	90k	Yes

Refund	Marital Status	Income	Class
Yes	Single	125k	No
No	Married	100k	No
No	Single	70k	No
Yes	Married	120k	No
No	Divorced	90k	Yes
No	Married	60k	No
Yes	Divorced	220k	No
No	Single	85k	Yes
No	Married	75k	No
No	Single	90k	Yes



Refund	Marital Status	Income	Class
Yes	Single	125k	No
Yes	Married	120k	No
Yes	Divorced	220k	No

Refund	Marital Status	Income	Class
No	Married	100k	No
No	Single	70k	No
No	Divorced	90k	Yes
No	Married	60k	No
No	Single	85k	Yes
No	Married	75k	No
No	Single	90k	Yes

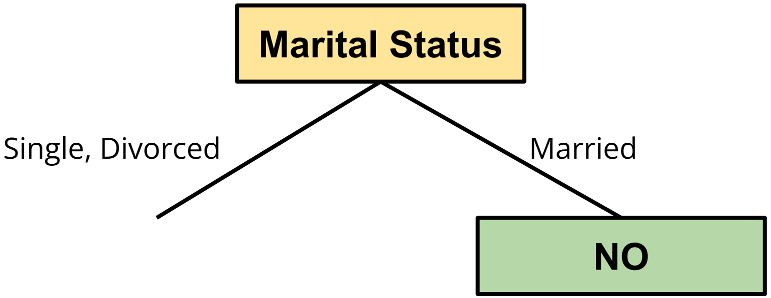


Refund	Marital Status	Income	Class
Yes	Single	125k	No
Yes	Married	120k	No
Yes	Divorced	220k	No

Refund	Marital Status	Income	Class
No	Married	100k	No
No	Single	70k	No
No	Divorced	90k	Yes
No	Married	60k	No
No	Single	85k	Yes
No	Married	75k	No
No	Single	90k	Yes

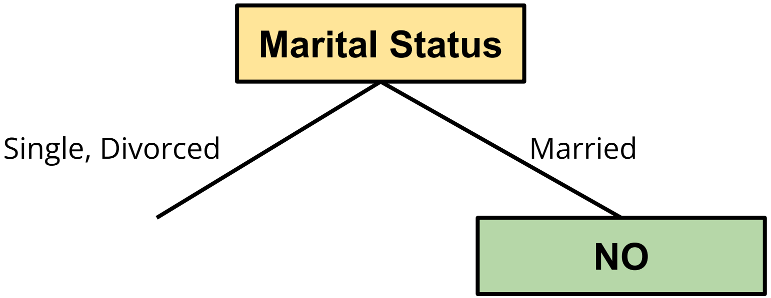
Refund	Marital Status	Income	Class
No	Married	100k	No
No	Single	70k	No
No	Divorced	90k	Yes
No	Married	60k	No
No	Single	85k	Yes
No	Married	75k	No
No	Single	90k	Yes

Refund	Marital Status	Income	Class
No	Married	100k	No
No	Single	70k	No
No	Divorced	90k	Yes
No	Married	60k	No
No	Single	85k	Yes
No	Married	75k	No
No	Single	90k	Yes



Refund	Marital Status	Income	Class
No	Single	70k	No
No	Divorced	90k	Yes
No	Single	85k	Yes
No	Single	90k	Yes

Refund	Marital Status	Income	Class
No	Married	100k	No
No	Married	60k	No
No	Married	75k	No

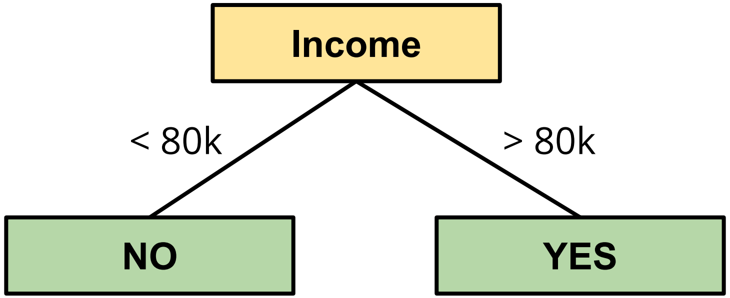


Refund	Marital Status	Income	Class
No	Single	70k	No
No	Divorced	90k	Yes
No	Single	85k	Yes
No	Single	90k	Yes

Refund	Marital Status	Income	Class
No	Married	100k	No
No	Married	60k	No
No	Married	75k	No

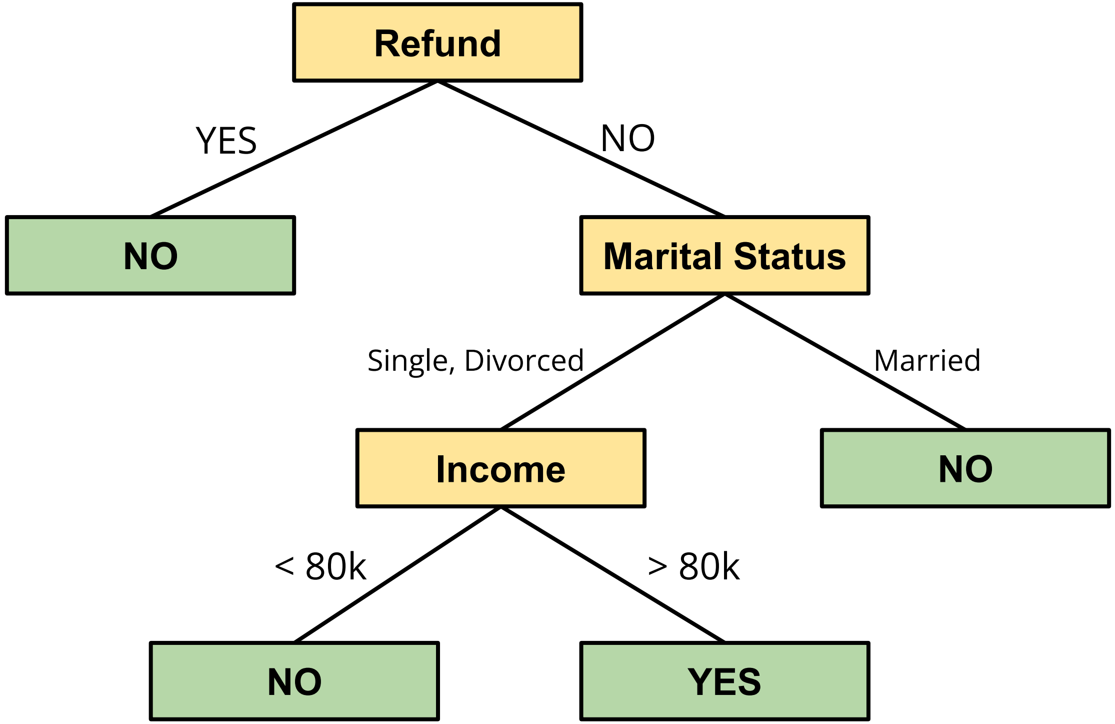
Refund	**Marital Status**	**Income**	**Class**
No	Single	70k	No
No	Divorced	90k	Yes
No	Single	85k	Yes
No	Single	90k	Yes

Refund	Marital Status	Income	Class
No	Single	70k	No
No	Divorced	90k	Yes
No	Single	85k	Yes
No	Single	90k	Yes



Refund	Marital Status	Income	Class
No	Single	70k	No

Refund	Marital Status	Income	Class
No	Divorced	90k	Yes
No	Single	85k	Yes
No	Single	90k	Yes

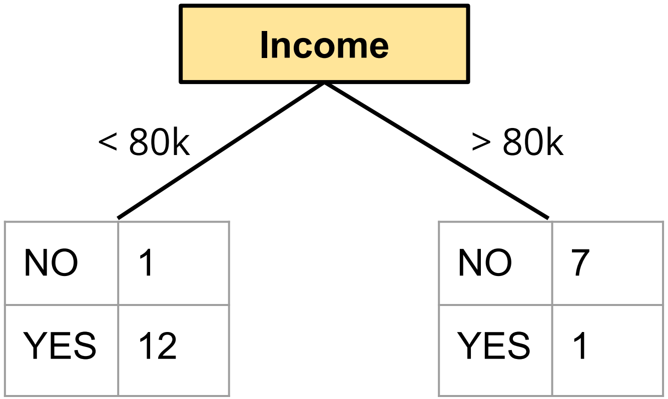

Refund
YES
NO
NO
Marital Status
Single, Divorced
Married
Income
NO
< 80k
> 80k
NO
YES

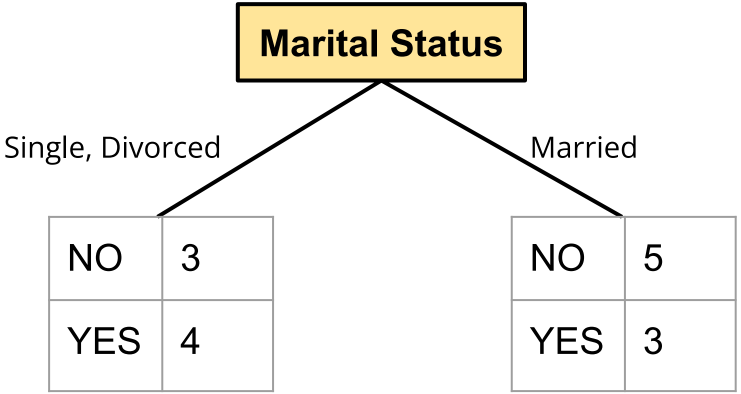
What do we mean by best split?

Many ways to split a given attribute

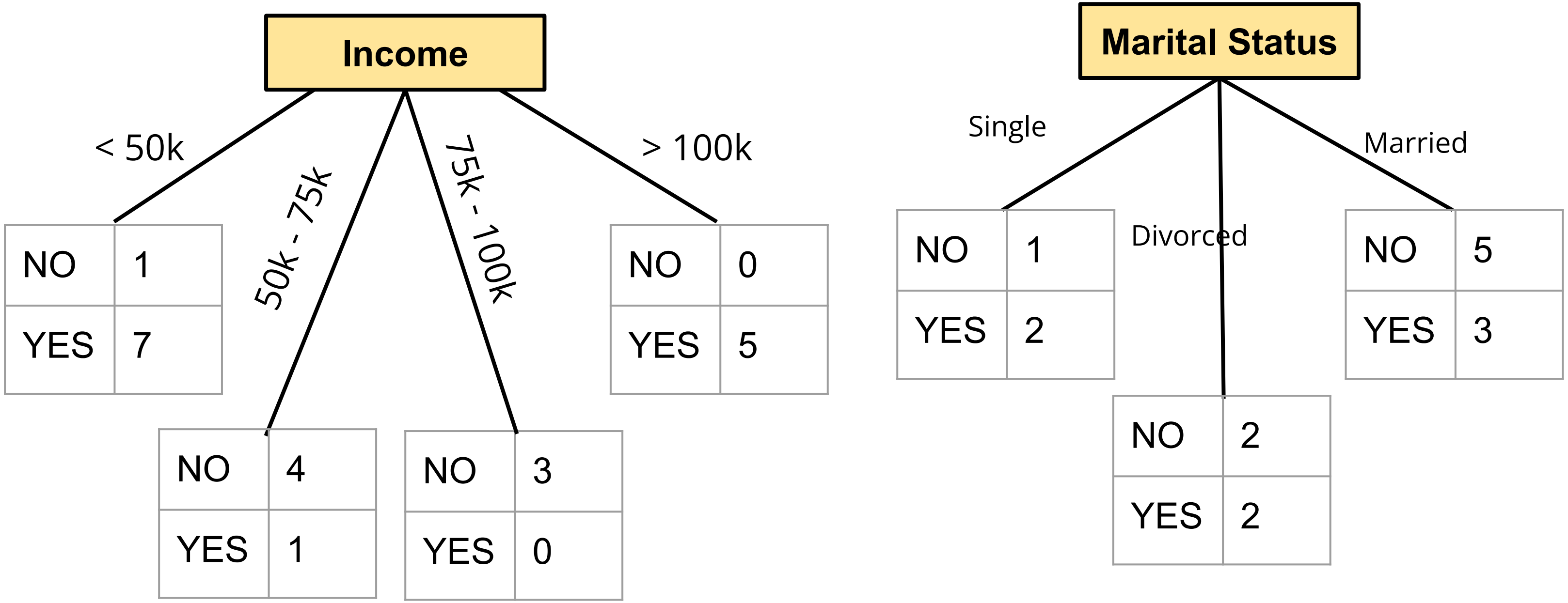
- Binary Split
- Multi-Way Split

Binary Split





Multi-Way Split



Continuous Variables

- Use binning before running the decision tree
  - Can use clustering for that for example
- Compute a threshold while building the tree
  - $A > t$ vs $A < t$

Need a metric

That favors nodes like this:

NO	1
YES	7

Over nodes like this:

NO	4
YES	4

GINI index

Denote $p(j \mid t)$ as the relative frequency of class j at node t.

NO	1
YES	7

$p(\text{NO} \mid t) = 1/8$
$p(\text{YES} \mid t) = 7/8$

NO	4
YES	3

$p(\text{NO} \mid t) = 4/7$
$p(\text{YES} \mid t) = 3/7$

GINI index

$$GINI(t) = 1 - \sum_j p(j|t)^2$$

NO	1
YES	7

p(NO | t) = ⅛
p(YES | t) = ⅞

GINI(t) = 1 - 1/64 - 49/64 = 14/64

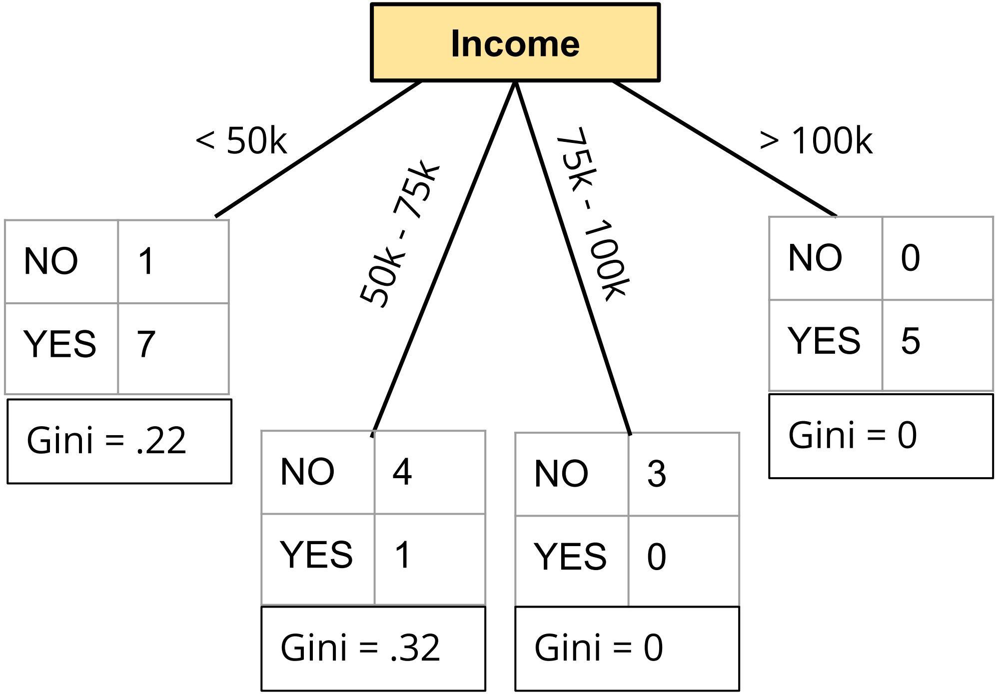
NO	4
YES	3

p(NO | t) = 4/7
p(YES | t) = 3/7

GINI(t) = 1 - 16/49 - 9/49 = 24/49

worksheet

GINI of the Split



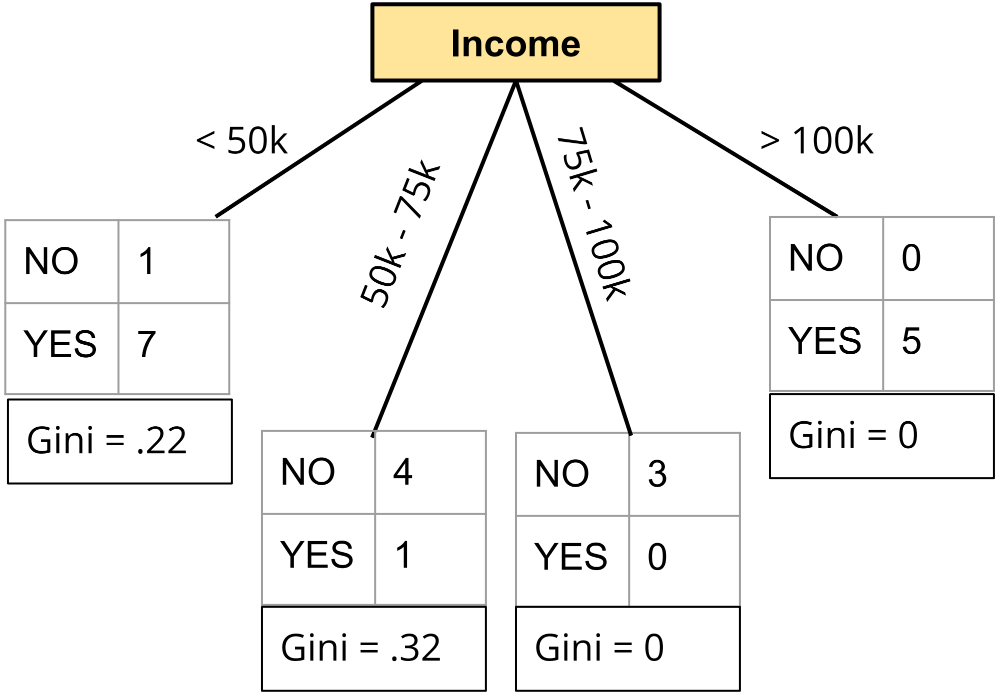
GINI of the split

$$GINI_{split} = \sum_{t=1}^{k} \frac{n_t}{n} GINI(t)$$

where:

n_t = number of data points at node t
n = number of data points before the split (parent node)

GINI of the split



$$GINI_{split} = \sum_{t=1}^{k} \frac{n_t}{n} GINI(t)$$

n = 21

$$\begin{aligned} GINI_{split} &= .22 * 8/21 \\ &+ .32 * 5/21 \\ &+ 0 * 3/21 \\ &+ 0 * 5/21 \\ &= .16 \end{aligned}$$

worksheet

Putting it all together

Before splitting

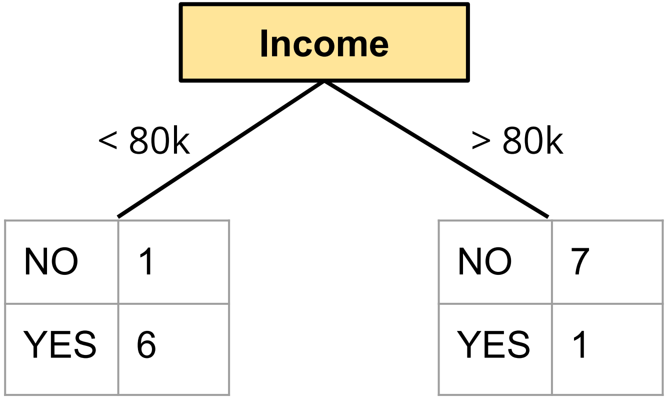
NO	8
YES	7

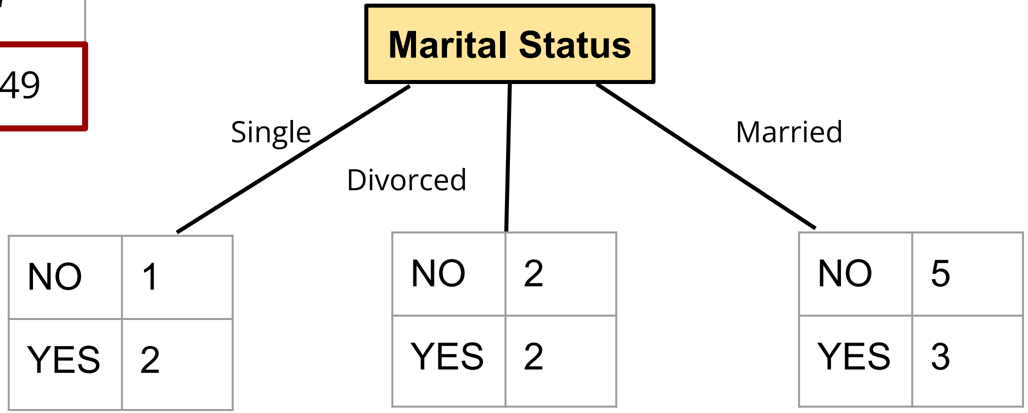
Gini = .49

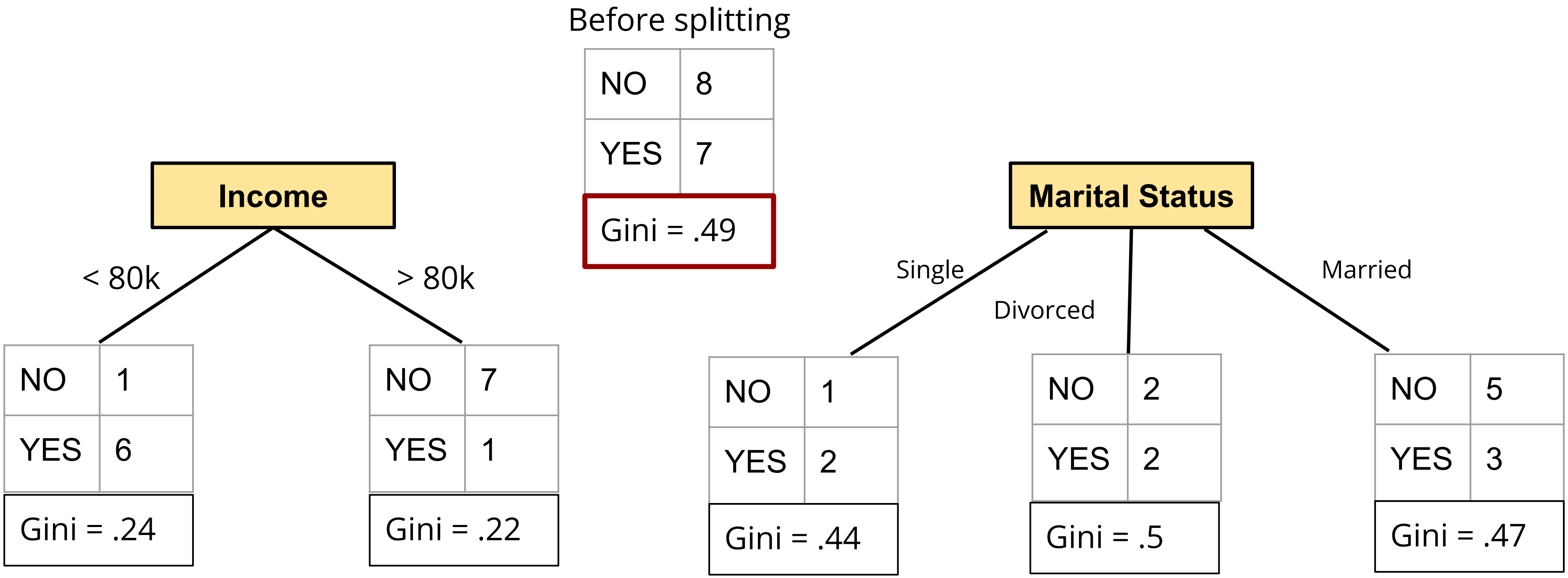
Before splitting

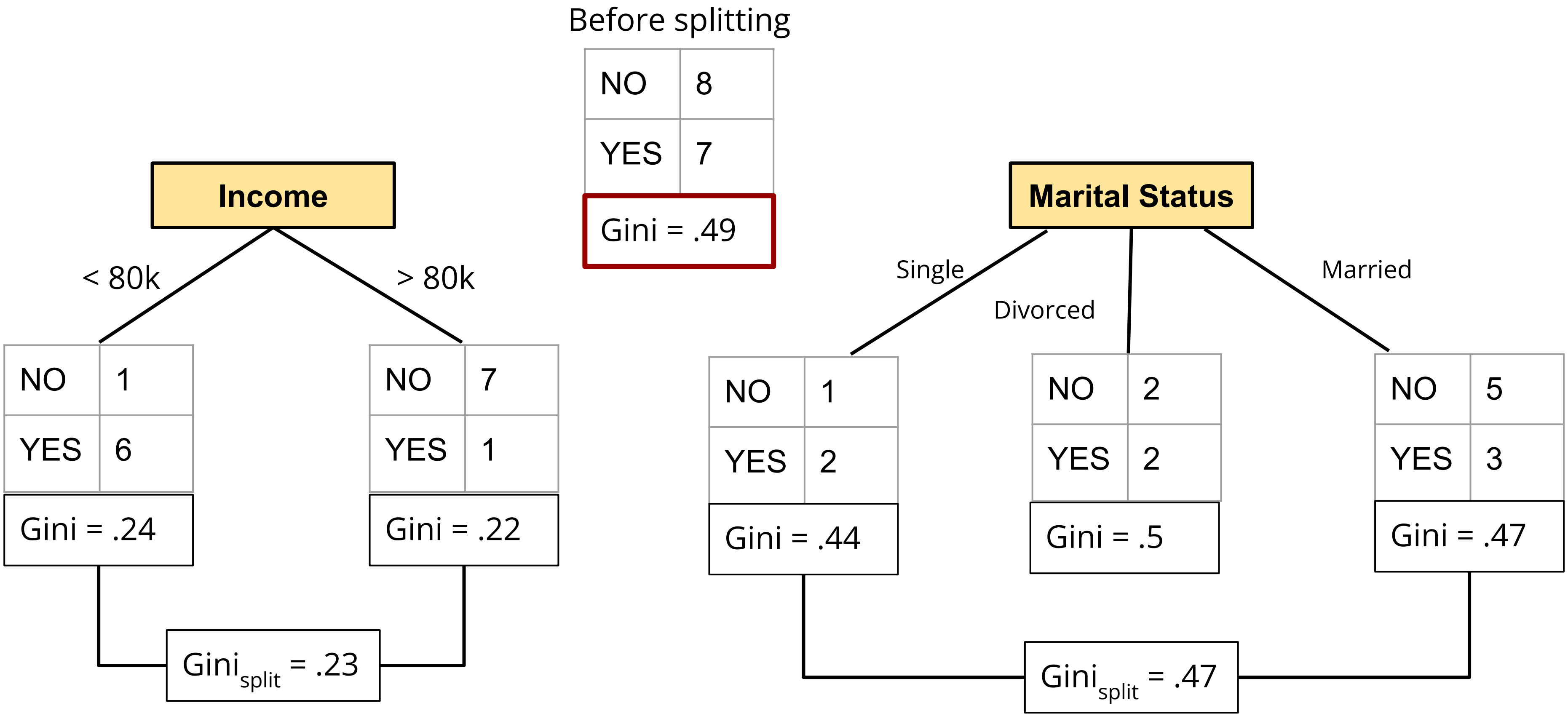
NO	8
YES	7

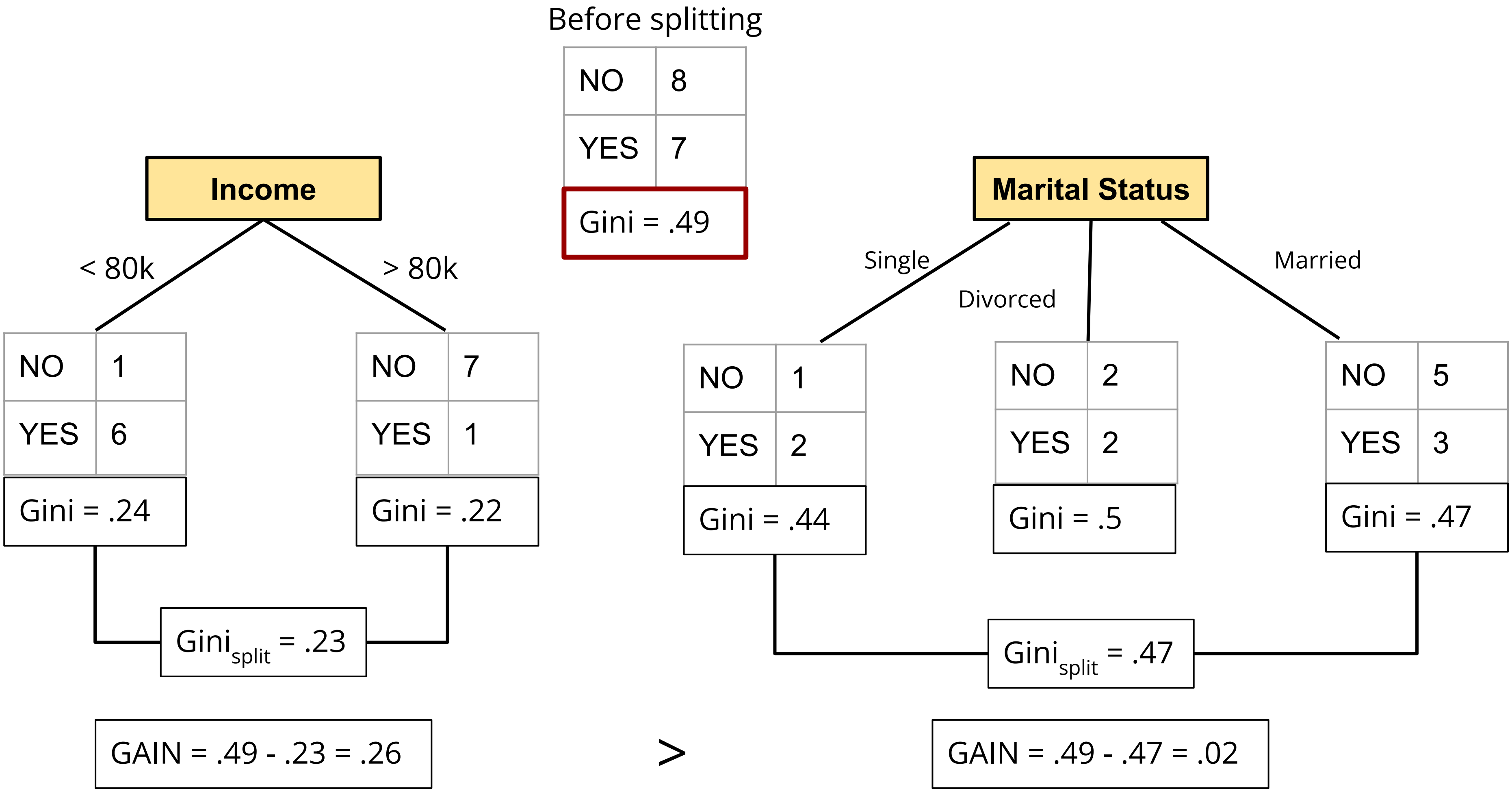
Gini = .49

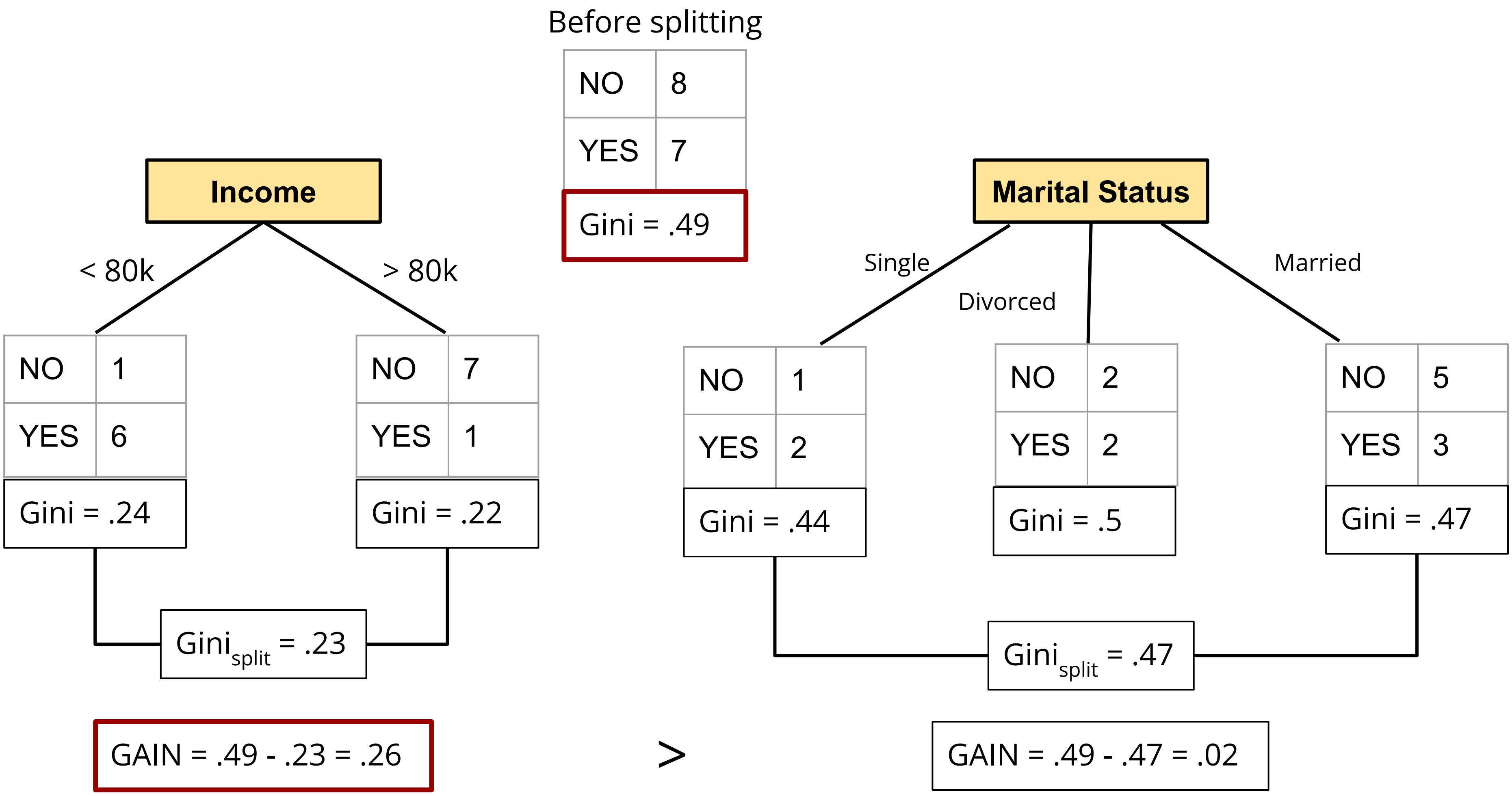





Before splitting
NO 8
YES 7
Gini = .49
Income
< 80k
> 80k
NO 1
YES 6
Gini = .24
NO 7
YES 1
Gini = .22
Marital Status
Single
Divorced
Married
NO 1
YES 2
Gini = .44
NO 2
YES 2
Gini = .5
NO 5
YES 3
Gini = .47

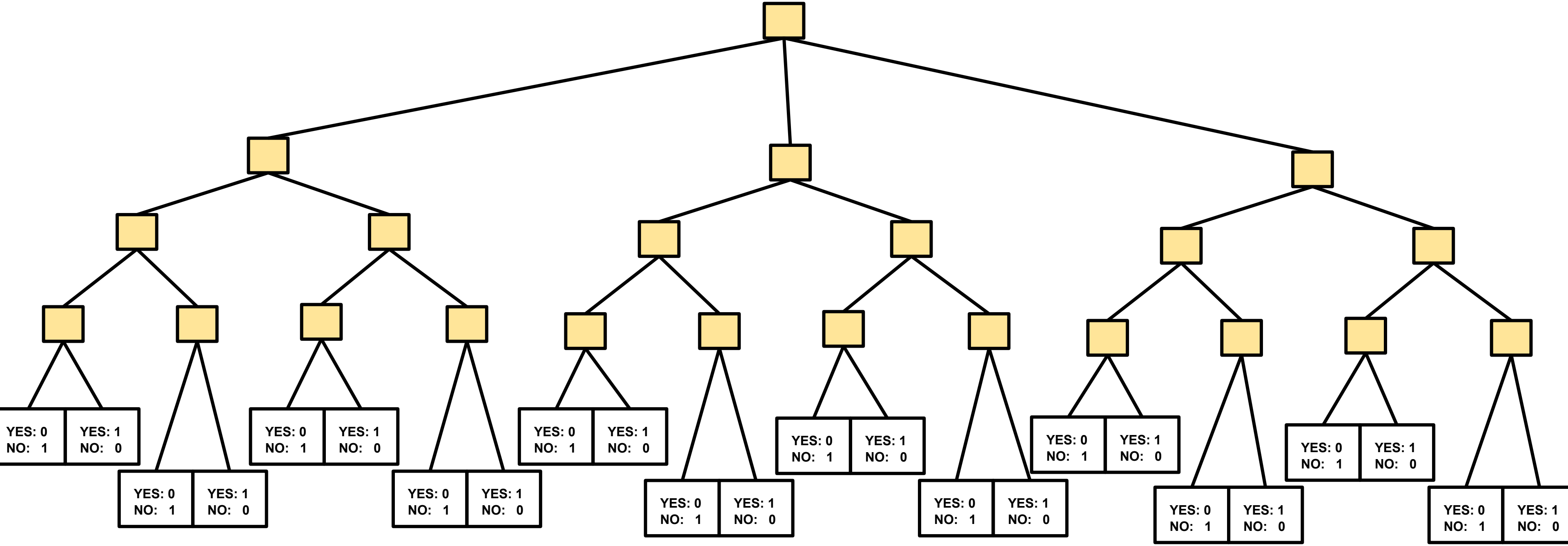

Before splitting
NO 8
YES 7
Gini = .49
Income
< 80k
> 80k
NO 1
YES 6
Gini = .24
NO 7
YES 1
Gini = .22
$Gini_{split}$ = .23
Marital Status
Single
Divorced
Married
NO 1
YES 2
Gini = .44
NO 2
YES 2
Gini = .5
NO 5
YES 3
Gini = .47
$Gini_{split}$ = .47


Before splitting
NO 8
YES 7
Gini = .49
Income
< 80k
> 80k
NO 1
YES 6
Gini = .24
NO 7
YES 1
Gini = .22
Gini_split = .23
GAIN = .49 - .23 = .26
>
Marital Status
Single
Divorced
Married
NO 1
YES 2
Gini = .44
NO 2
YES 2
Gini = .5
NO 5
YES 3
Gini = .47
Gini_split = .47
GAIN = .49 - .47 = .02


Before splitting
NO 8
YES 7
Gini = .49
Income
< 80k
> 80k
NO 1
YES 6
Gini = .24
NO 7
YES 1
Gini = .22
Gini_split = .23
GAIN = .49 - .23 = .26
>
Marital Status
Single
Divorced
Married
NO 1
YES 2
Gini = .44
NO 2
YES 2
Gini = .5
NO 5
YES 3
Gini = .47
Gini_split = .47
GAIN = .49 - .47 = .02

worksheet

Limitations


YES: 0 NO: 1
YES: 1 NO: 0
YES: 0 NO: 1
YES: 1 NO: 0
YES: 0 NO: 1
YES: 1 NO: 0
YES: 0 NO: 1
YES: 1 NO: 0
YES: 0 NO: 1
YES: 1 NO: 0
YES: 0 NO: 1
YES: 1 NO: 0
YES: 0 NO: 1
YES: 1 NO: 0
YES: 0 NO: 1
YES: 1 NO: 0
YES: 0 NO: 1
YES: 1 NO: 0
YES: 0 NO: 1
YES: 1 NO: 0
YES: 0 NO: 1
YES: 1 NO: 0
YES: 0 NO: 1
YES: 1 NO: 0

Easy to construct a tree that is too complex and overfits the data.

Solutions:

- Early termination (stop before tree is fully grown - use majority vote at leaf node)
  - Stop at some specified depth
  - Stop if size of node is below some threshold
  - Stop if gini does not improve
- Pruning (create fully grown tree then trim)

Extensions

Other measures of node purity

- Entropy

$$\mathrm{Entropy}(t) = -\sum_j p(j|t)\log(p(j|t))$$

- Misclassification Error

$$\mathrm{Error}(t) = 1 - \max_j(p(j|t))$$

worksheet