Decision Trees

Predicting loan defaults with decision trees

What makes a loan risky?

"I want to buy a new house"

Loan Application

- Credit
- Income
- Term
- Personal Info

Did I pay previous loans on time?

Example: excellent, good, or fair

→ Credit history

What's my income?

Example: $80k per year

→ Income

How soon do I need to pay the loan?

Example: 3 years, 5 years

→ Term

Age, reason for the loan, marital status, ...

Example: home loan for a married couple

→ Personal info

Intelligent application

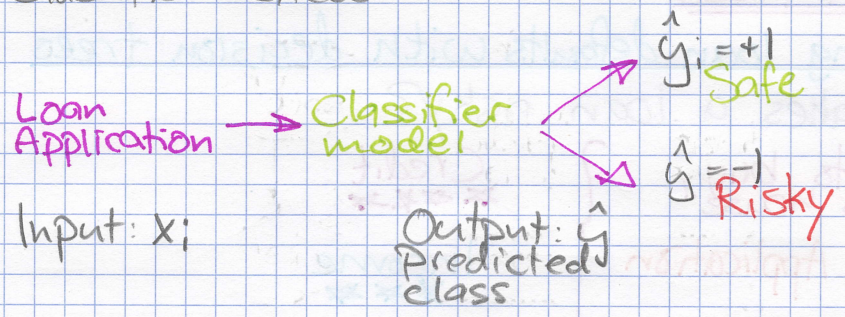
Loan Applications

#1
#2
#3

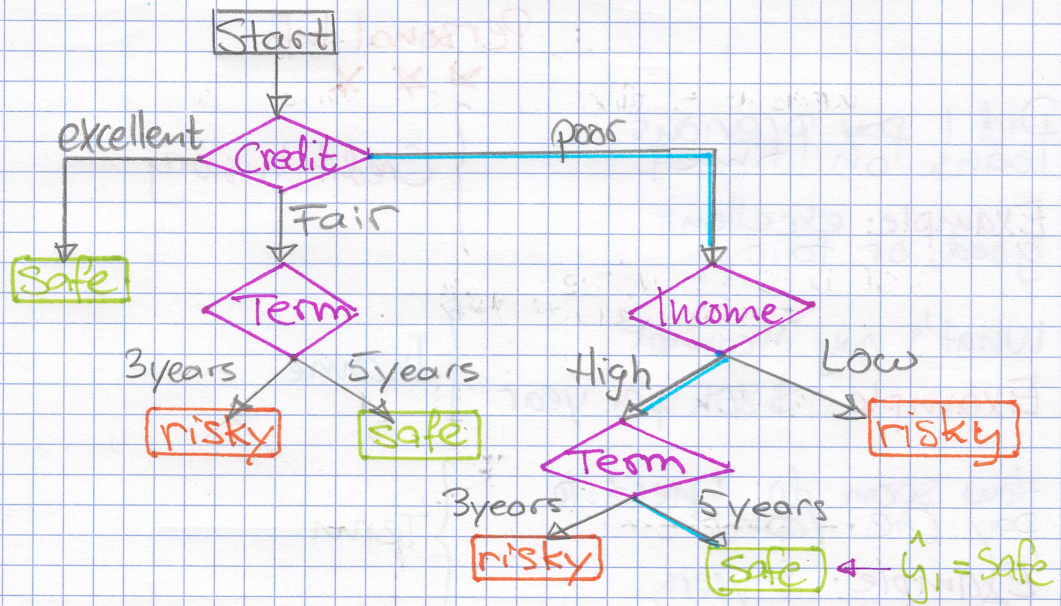
→ Intelligent loan application review system

- → Safe
- → Risky
- → Risky

Classifier review



This module... decision trees



Scoring a loan application

$x_i = (Credit = poor, Income = high, Term = 5 years)$

Decision tree learning task

Learn decision tree from data?

Credit	Term	Income	y
excellent	3 yrs	high	safe
fair	5 yrs	low	risky
fair	3 yrs	high	safe
poor	5 yrs	high	risky
excellent	3 yrs	low	risky
fair	5 yrs	low	safe
poor	3 yrs	high	risky
poor	5 yrs	low	safe
fair	3 yrs	high	safe
$h_1(x)$	$h_2(x)$	$h_3(x)$	loan status

Training data
N observations
(x_i, y_i)

Optimize quality metric on training data

$T(x)$

Quality metric: Classification error

- Error measures fraction of mistakes

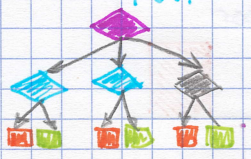
$$\text{Error} = \frac{\#\text{incorrect predictions}}{\#\text{examples}}$$

- Best possible value: 0.0
- Worst possible value: 1.0

How do we find the best tree?

Exponentially large number of possible trees makes decision tree learning hard!
(NP-hard problem)

$T_1(x)$ $T_2(x)$ $T_3(x)$

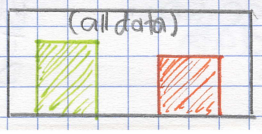


Simple (greedy) algorithm finds "good" trees

Greedy decision tree algorithm:

Algorithm outline

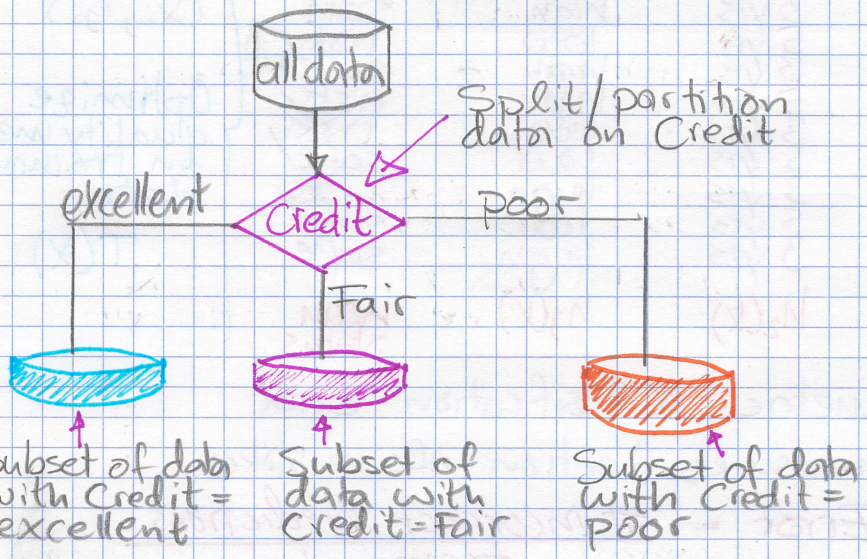
Step 1: Start with an empty tree



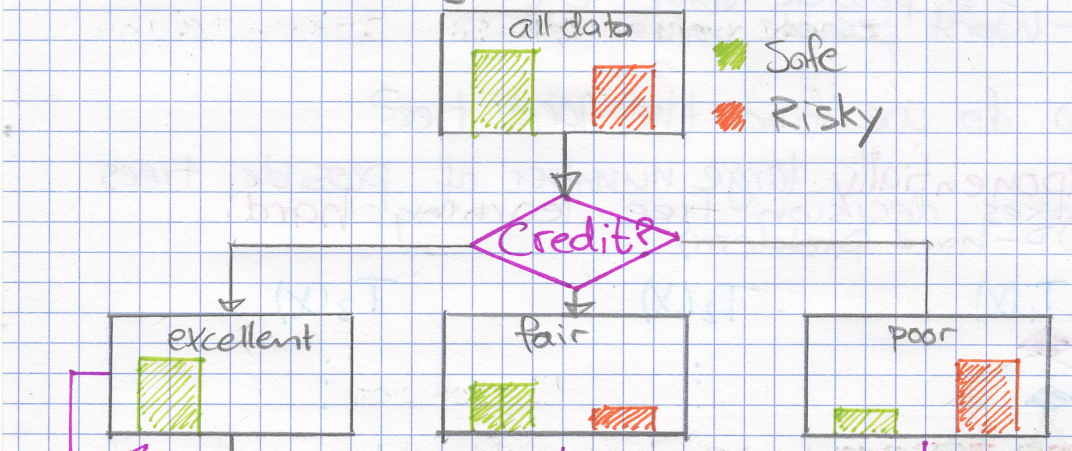
Safe

Risky

Step2: Split on a feature



Step3: Making predictions



Step4: Recursion

Nothing more to do here

Build tree from these data points

Build tree from these data points

Greedy decision tree learning

- Step1: Start with an empty tree
- Step2: Select a feature to split data
- For each split of the tree:
  - Step3: If nothing more to, make prediction
  - Step4: Otherwise, go to Step2 & continue (recurse) on this split

Recursion

Problem1: Feature split selection

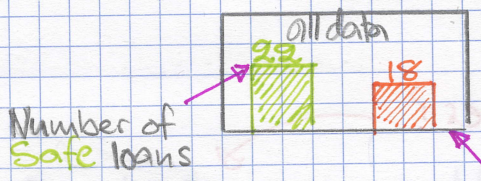
Problem2: Stopping condition

Learning a decision stump

Feature split learning = decision stump learning

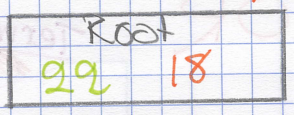
Start with all the data

Loan status Safe Risky

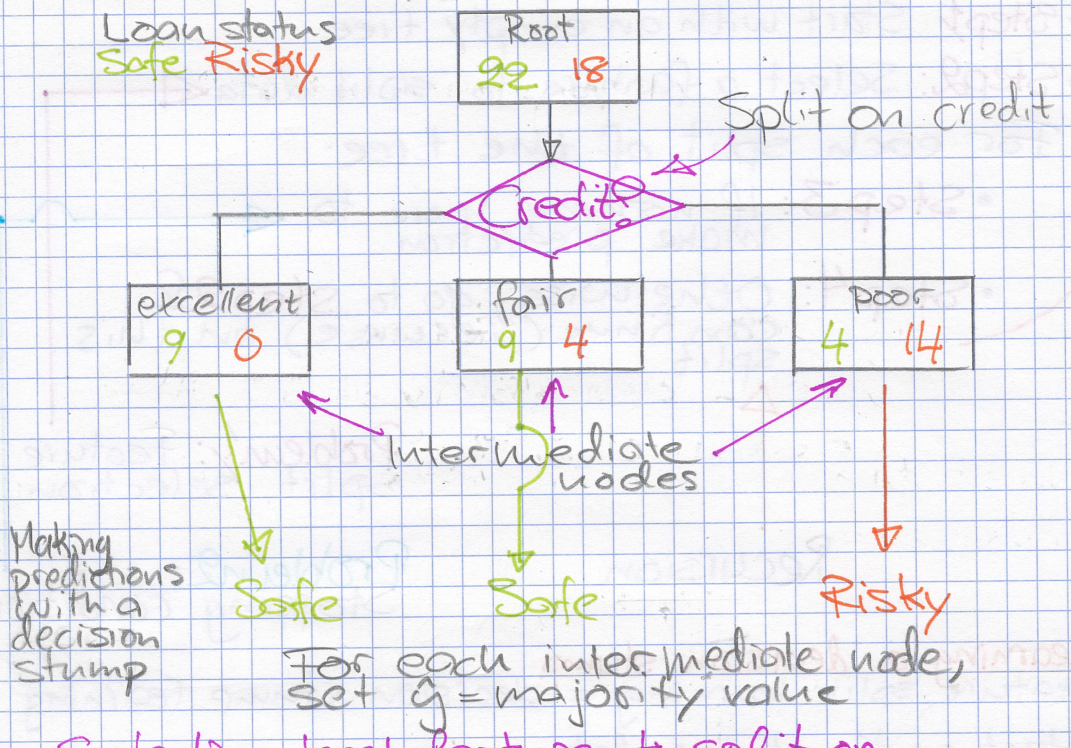


Compact visual notation: Root node

Loan status Safe Risky

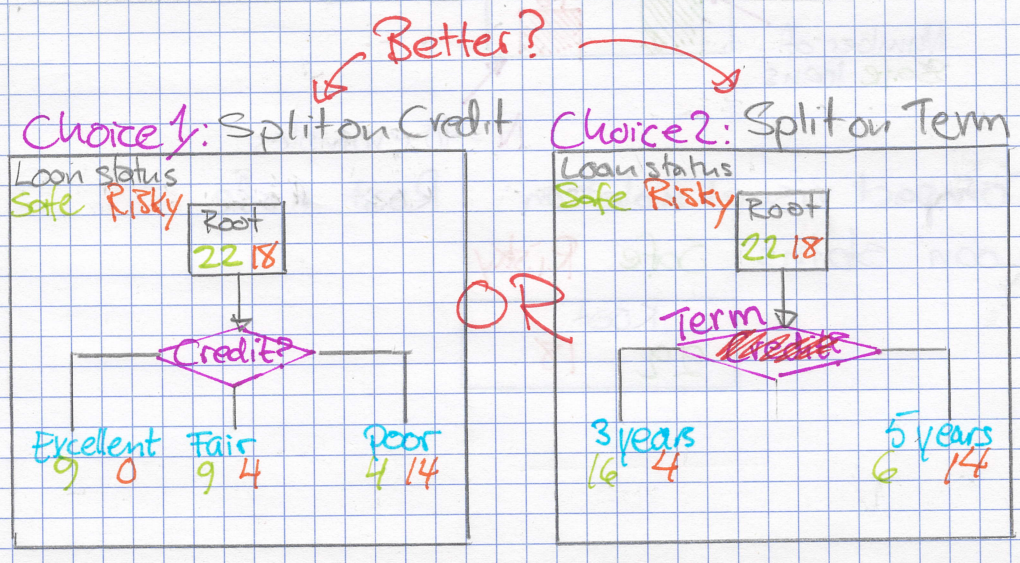


Decision stump: Single level tree



Selecting best feature to split on

How do we select the best feature



How do we measure effectiveness of a split?

Loan status: Safe, Risky — Root: 22, 18

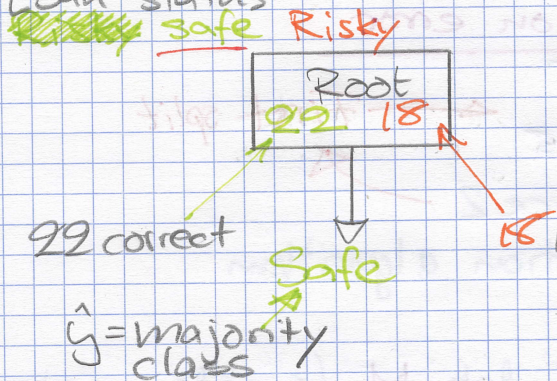
Credit: excellent (9, 0), fair (9, 4), poor (4, 14)

Idea: Calculate classification error of this decision stump

$$\text{Error} = \frac{\#\text{mistakes}}{\#\text{data points}}$$

Calculating classification Error

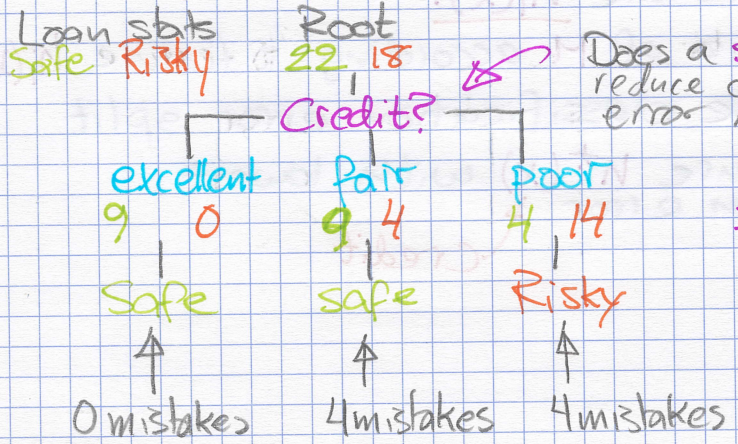
- **Step 1:** $\hat{y}$ = class of majority of data in node
- **Step 2:** Calculate classification error of predicting $\hat{y}$ for this data



$$\text{Error} = \frac{18}{22+18} = 0.45$$

Tree	Classification error
(root)	0.45

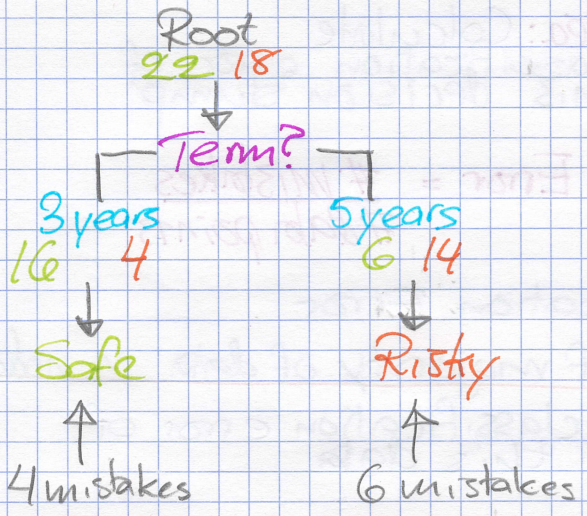
Choice 1: Split on credit history



Does a split on Credit reduce classification error below 0.45?

$$\text{Error} = \frac{4+4}{40} = 0.20$$

Choice 2: Split on Term



$$\text{Error} = \frac{4+6}{40} = 0.25$$

Tree Classification error

(root)	0.45
Split on Credit	0.2
Split on Term	0.25

← First split

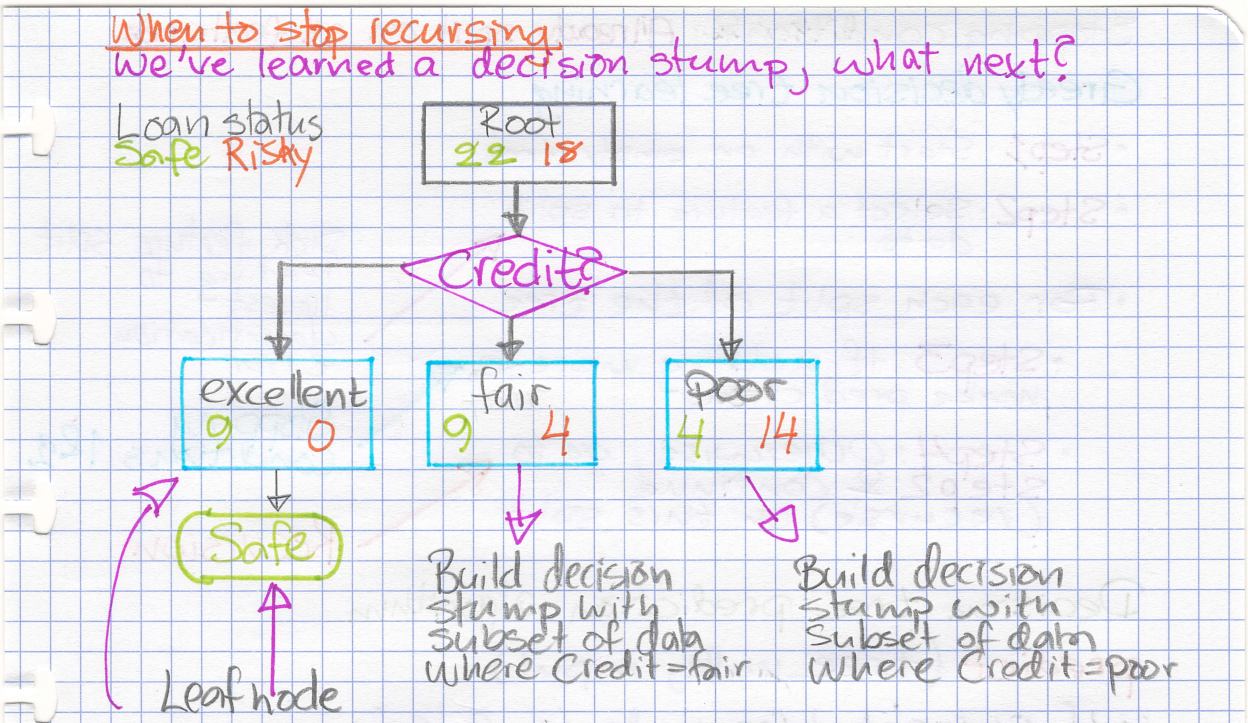
Choice 1 vs Choice 2

Feature split selection algorithm

- Given a subset of data M (a node in a tree)
- For each feature $h_i(x)$: ← credit, term, income
  1: Split data of M according to feature $h_i(x)$
  2: Compute classification error split
- Chose feature $h^*(x)$ with lowest classification error ← Credit

When to stop recursing

We've learned a decision stump, what next?



All data points are Safe → nothing else to do with this subset of data

Simple greedy decision tree learning

Pick best feature to split on

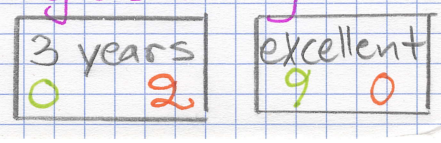
Learn decision stump with this split

For each leaf of decision stump, recurse

When do we stop?

Stopping condition 1: All data agrees on y

All data in these nodes have same y value → nothing to do



Stopping condition2: Already split on all features

Greedy decision tree learning

- Step1: Start with an empty tree
- Step2: Select a feature to split data
- For each split of the tree
  - Step3: If nothing more to, make predictions
  - Step4: Otherwise, go to Step2 & continue (recurse) on this split

Pick feature split leading to lowest classification error

Stopping conditions 1&2

Recursion

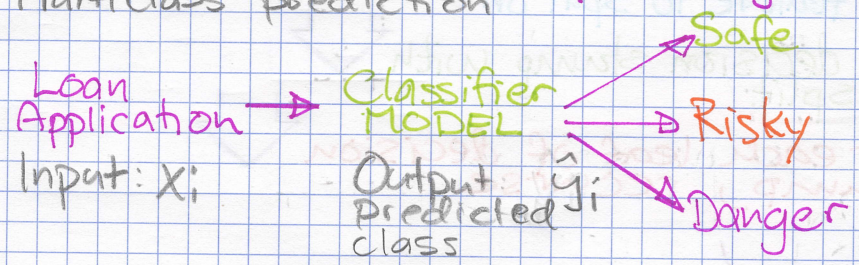
Decision tree prediction algorithm

predict(tree_node, input)

- If current tree_node is a leaf:
  - return majority class of data points in leaf
- else:
  - next_node = child of tree_node whose feature value agrees with input
  - return predict(next_node, input)

Multiclass classification & predicting probabilities

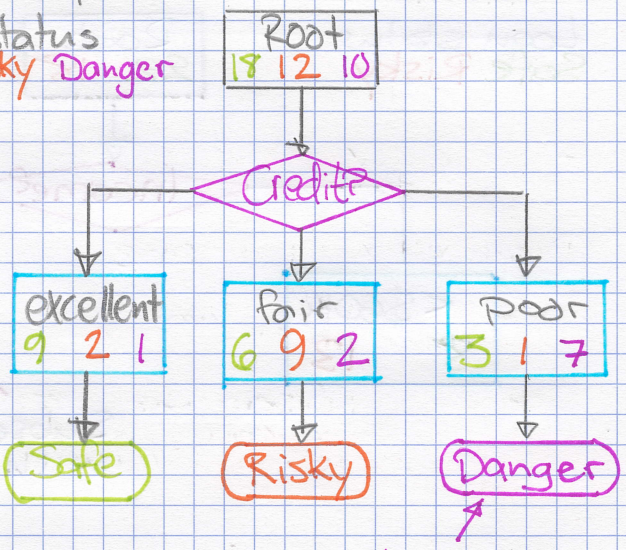
Multiclass prediction



Multiclass decision stump

N = 40
1 feature
3 classes

Loan status
Safe Risky Danger



$$P(y = \text{danger} \mid x) = \frac{7}{3+1+7} = 0.64$$

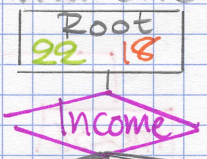
Decision tree learning: Real valued features

How do we use real values inputs?

Income	Credit	Term	y
$ 105K	excellent	3 yrs	Safe
112	good	5 yrs	Risky
73	fair	3 yrs	Safe
69	excellent	5 yrs	Safe
217	excellent	3 yrs	Risky
120	good	5 yrs	Safe
64	fair	3 yrs	Risky
340	excellent	5 yrs	Safe
60	good	3 yrs	Risky

Split on each numeric value?

Danger: May only contain one data point per node

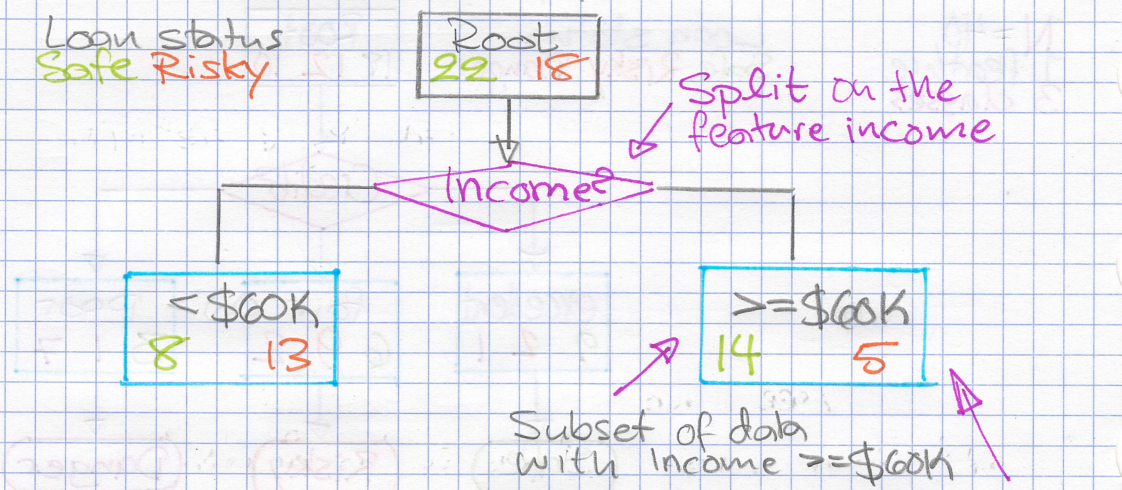


Loan status
Safe Risky

$30K 31.4K 39.5K ------ $61.1K $91.3K
0 1 1 0 0 1 0 1 0 1

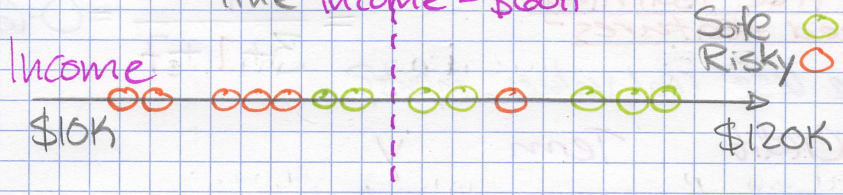
Can't trust prediction (overfitting)

Alternative: Threshold split



Threshold splits in 1-D

Threshold split is the line Income = $60K



Each split partitions the 2-D space