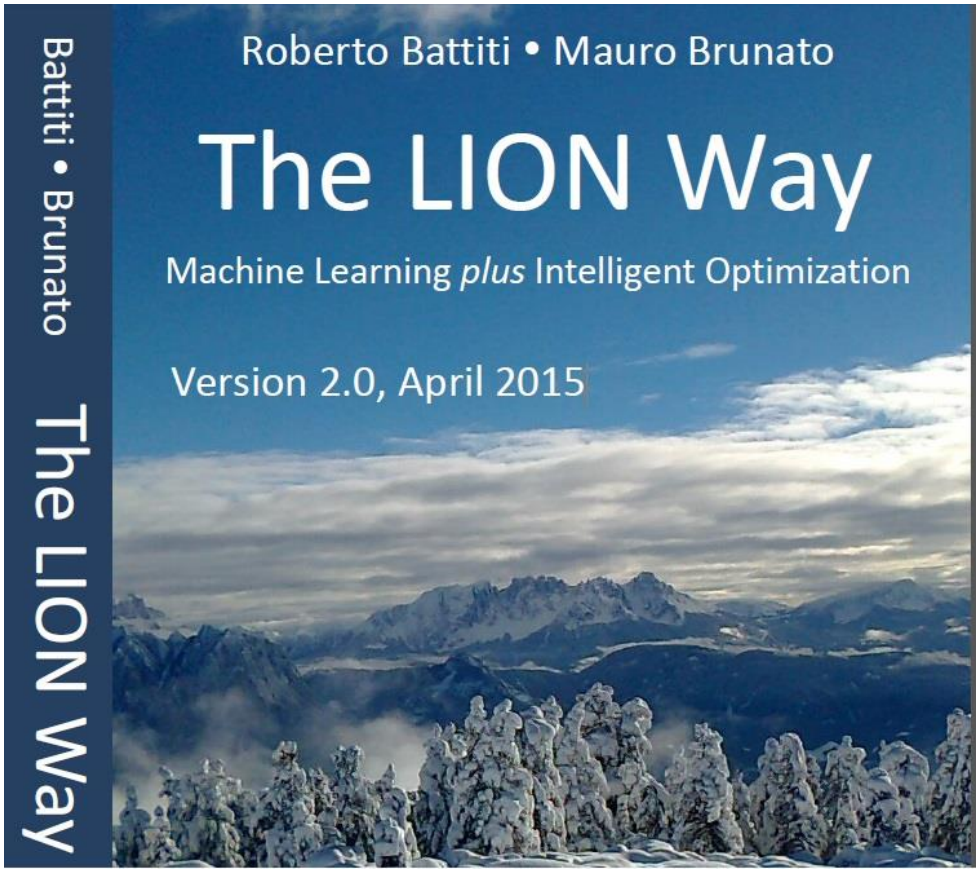


ROBERTO BATTITI, MAURO BRUNATO.
The LION Way: Machine Learning plus *Intelligent Optimization*.
LIONlab, University of Trento, Italy,
Apr 2015

http://intelligent-optimization.org/LIONbook

Chap.6 Rules, decision trees, and forests

If a tree falls in the forest and there's no one there to hear it, does it make a sound?

Rules

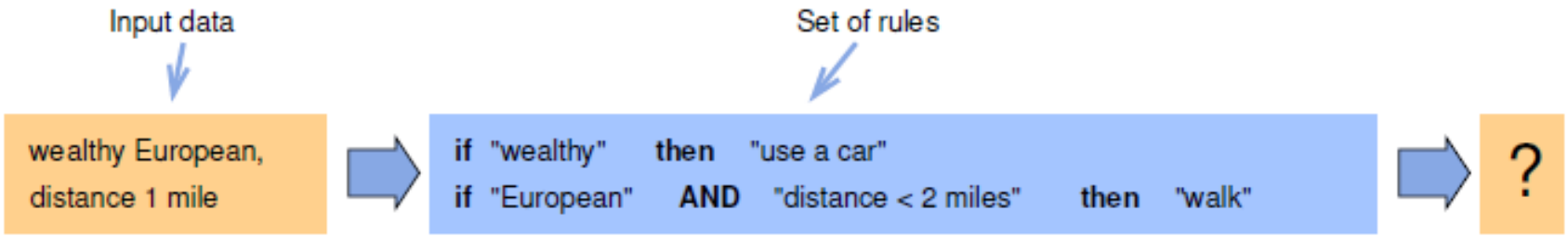
- Rules are a way to condense nuggets of knowledge in a way amenable to human understanding

"customer is wealthy" → "he will buy my product."

"body temperature > 37 degrees Celsius" → "patient is sick."

Non-contradictory sets of rules

- If the set of rules gets large, complexities can appear, like rules leading to contradictory classifications



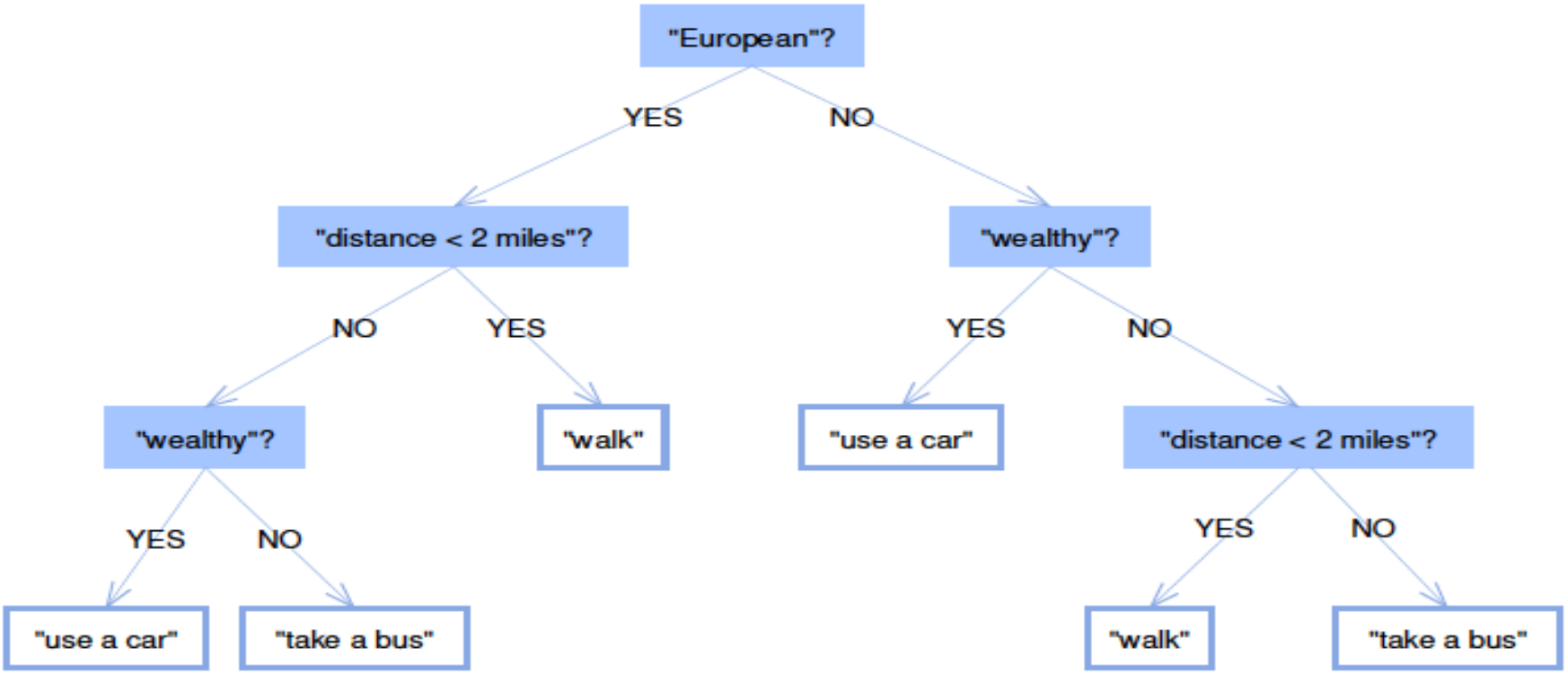
- Automated ways to extract non-contradictory rules from data are precious

Automated generation of rules

- breaking rules into a chain of simple questions is key
- the most informative questions are better placed at the beginning of the sequence, leading to a hierarchy of questions
- decision trees are organized hierarchical arrangement of decision rules, without contradictions

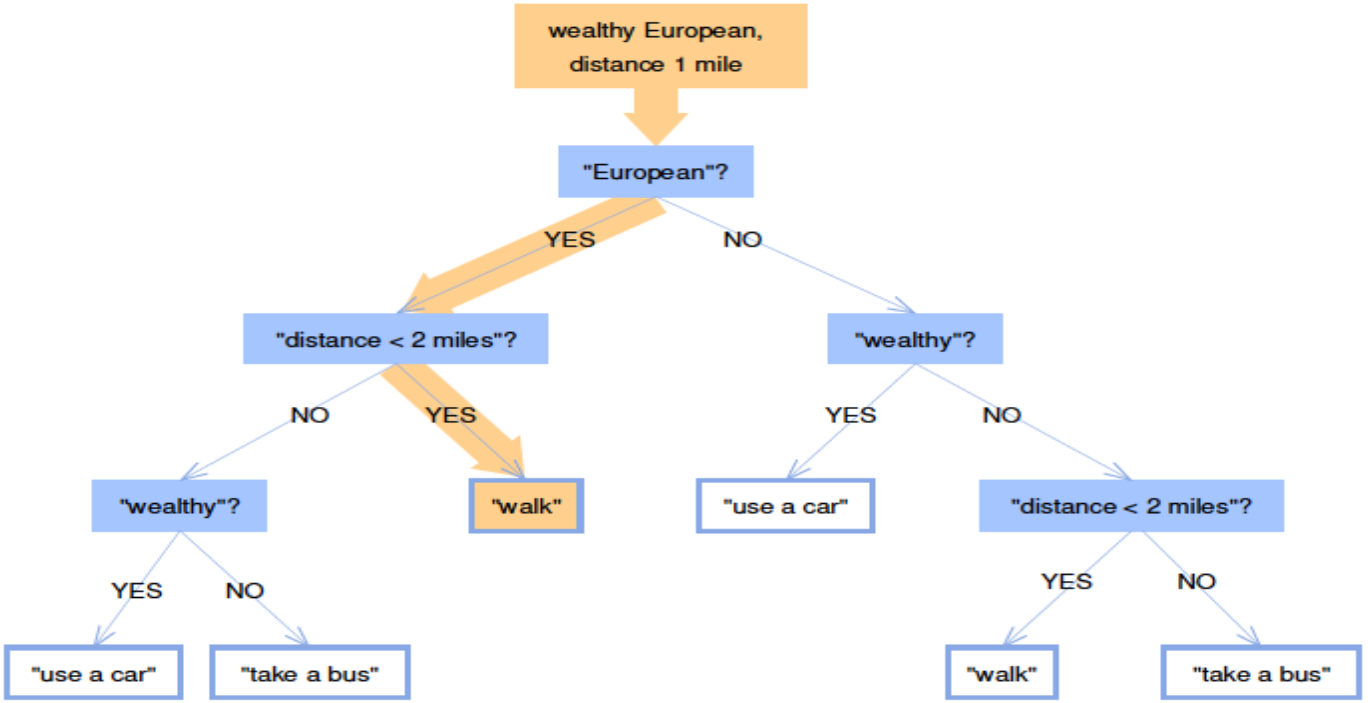
Decision trees

- A decision tree is a set of questions organized in a hierarchical manner and represented graphically as a tree



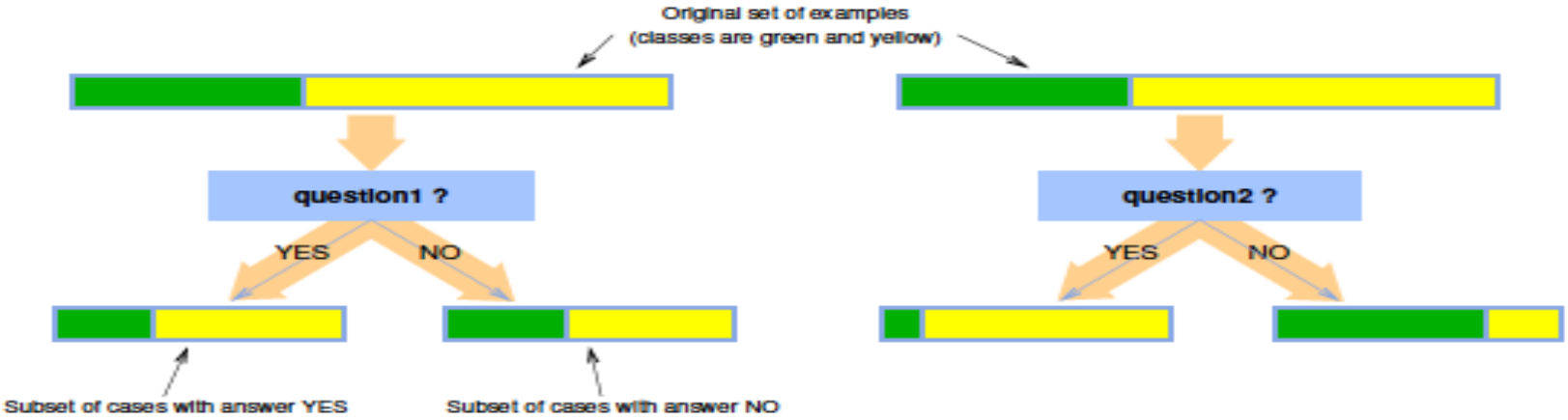
Decision trees

- For an input object, a decision tree estimates an unknown property by asking successive questions about its known properties



Building decision trees

- Basic idea: ask the most informative questions first
- Recursively construct a hierarchy of questions

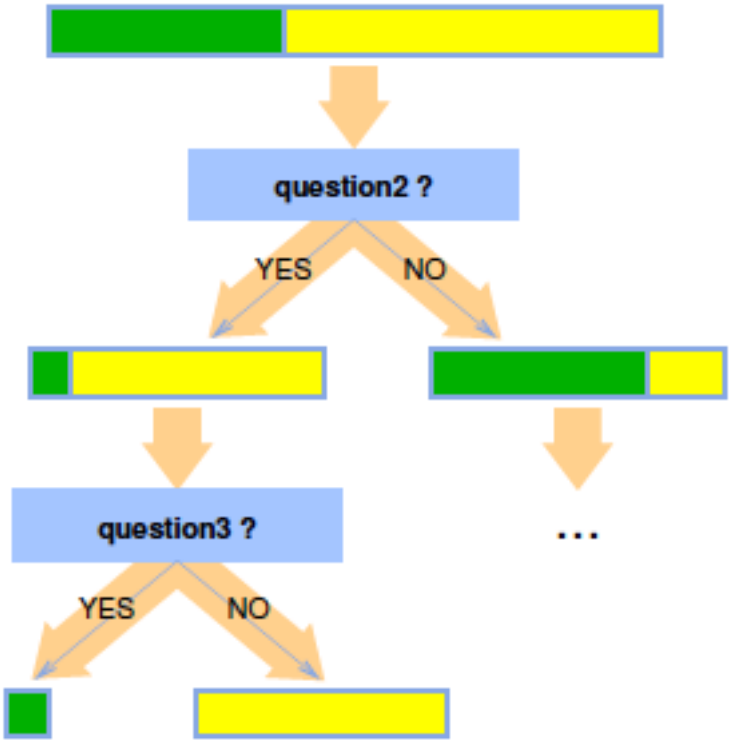


- At each step, choose the question that leads to the purest subsets

Building decision trees

Recursive step in tree building:

after the initial purification by question2 ,the same method is applied on the *left* and *right* example subsets.



Question3 is sufficient to completely purify the subsets.

No additional recursive call is executed on the pure subsets

How do we measure purity?

- We need a quantitative measure of purity

The two widely used measures of purity of a subset are the

- Information gain
- Gini impurity

Shannon Entropy

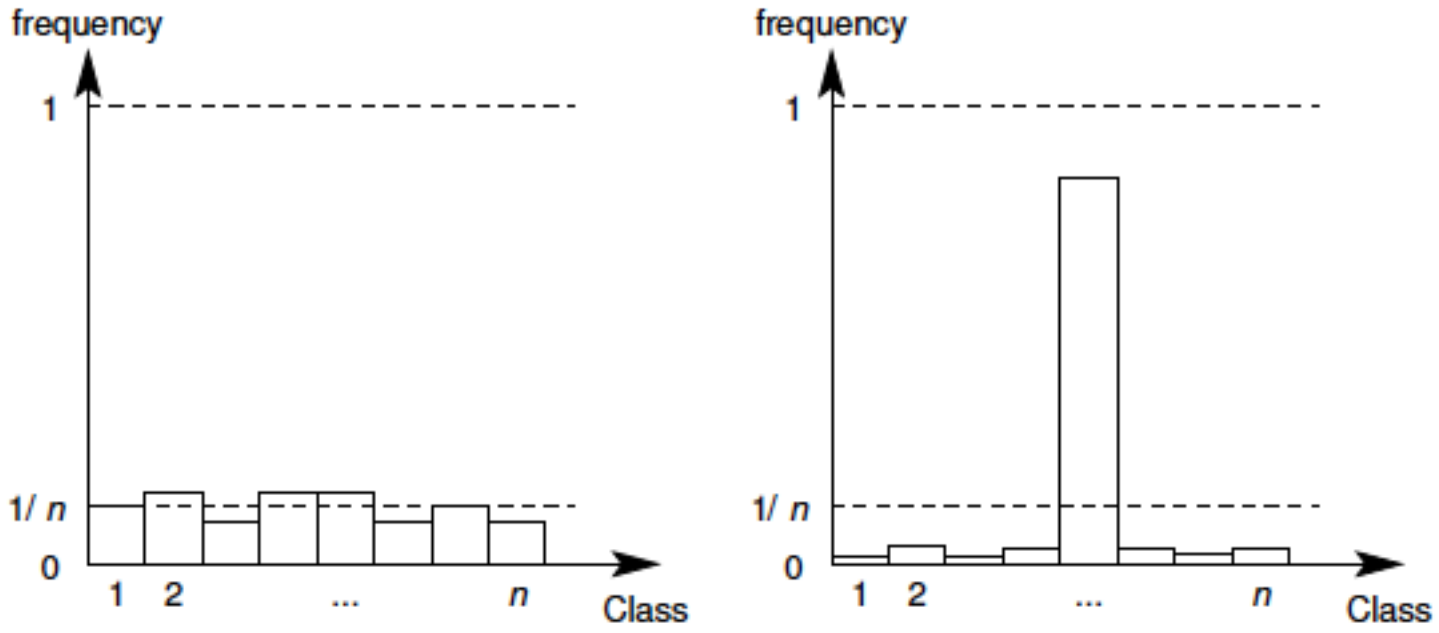
Suppose we sample from a set associated to an internal node or to a leaf of the tree

We are going to get examples of a class y with a probability Pr(y) proportional to the fraction of cases of the class present in the set

The statistical uncertainty in the obtained class is measured by Shannon's entropy of the label probability distribution:

$$H(Y) = -\sum_{y \in Y} \Pr(y) \log \Pr(y).$$

Shannon entropy



High entropy (left): events have similar probabilities, the uncertainty high.

Low entropy (right): events have very different probabilities, the uncertainty is because one event collects most probability.

Information Gain

The average reduction in entropy after knowing the answer is the information gain (or mutual information)

$$\mathrm{IG} = H(\mathcal{S}) - \frac{|\mathcal{S}_{\mathrm{YES}}|}{|\mathcal{S}|} H(\mathcal{S}_{\mathrm{YES}}) - \frac{|\mathcal{S}_{\mathrm{NO}}|}{|\mathcal{S}|} H(\mathcal{S}_{\mathrm{NO}}).$$

An optimal question maximizes IG, that is, reduces the average entropy as much as possible.

S is the current set of examples, S=S_yes U S_no is the splitting obtained after answering a question

Gini impurity

Suppose that there are m classes, and let f_i be the fraction of items labeled with value i in the set.

Then the Gini Impurity is defined as

$$\mathrm{GI}(f) = \sum_{i=1}^{m} f_i(1 - f_i) = \sum_{i=1}^{m} (f_i - {f_i}^2) = \sum_{i=1}^{m} f_i - \sum_{i=1}^{m} {f_i}^2 = 1 - \sum_{i=1}^{m} {f_i}^2.$$

The Gini impurity measures how often a randomly chosen element from the set would be incorrectly labeled if it were randomly labeled according to the distribution of labels in the subset

This method produces zero errors if the set is pure, and a small error rate if a single class gets the largest share of the set

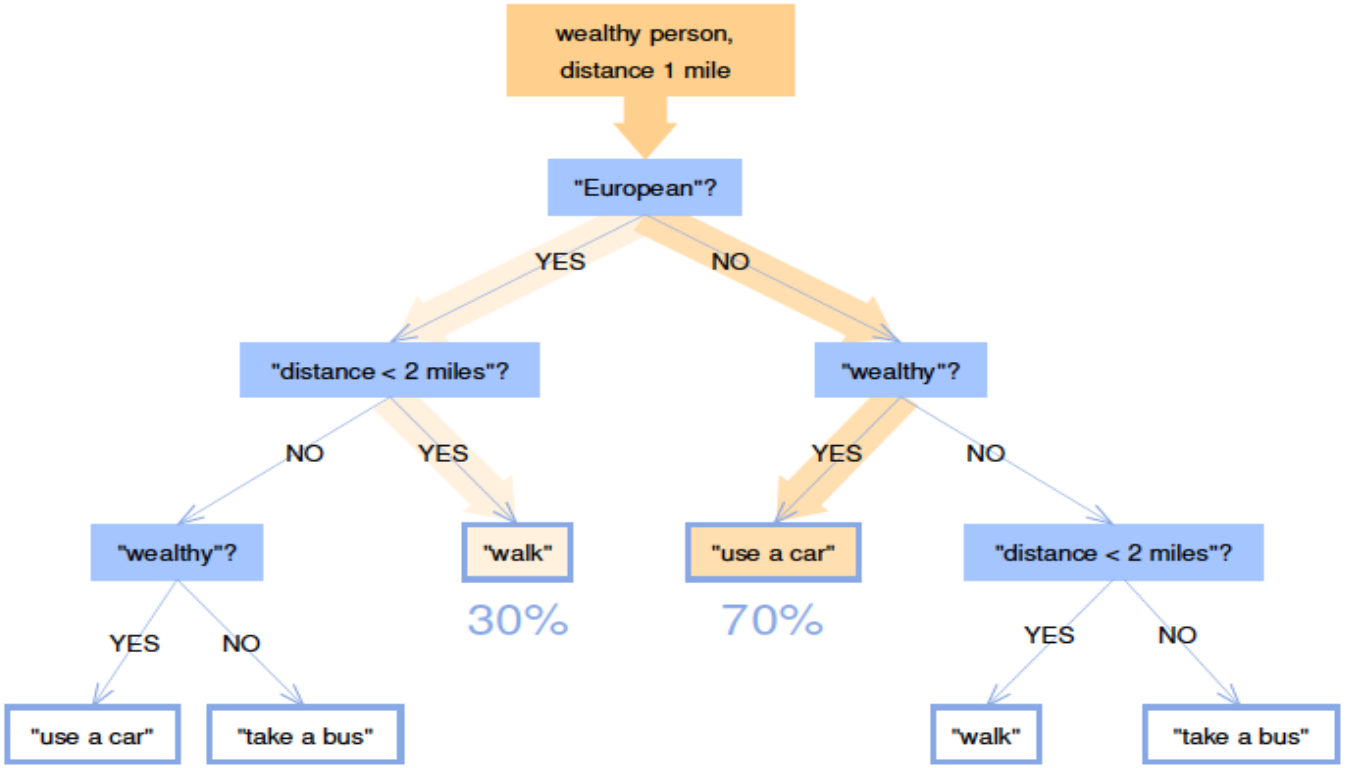
How to pose questions?

- In general, question should have a binary output.
- For a categorical variable, the test can be based on the variable having (or not having) a subset of the possible values
- For real-valued variables, the tests can be on a single variable or on simple (linear) combination of a subset of variables

Dealing with missing values

- In real-world data, missing values are abundant
- If an instance reaches a node and the question cannot be answered because data is lacking, ideally "split the instance into pieces", and send part of it down each branch in proportion to the number of training instances going down that branch.
- When the different pieces of the instance eventually reach the leaves, the corresponding leaf output values can be averaged.

Dealing with missing values



Missing nationality information. The data point arriving at the top node is sent both to its left and right child nodes with different weights depending on the frequency of "YES" and "NO" answers in the training set.

Decision forests

- Basic idea: use an ensemble of trees to make decisions in a democratic way

HOW?

- Train different trees on different sets of examples
- Allow for randomized choices during training

- an isolated tree will be rather weak, however, the majority (or weighted average) rule will provide reasonable answers.

Gist

- Simple **“if-then” rules** condense nuggets of information in a way which can be understood by human people.
- To avoid contradictory rules we proceed with a hierarchy of questions (the most informative first) leading to an organized structure of simple successive questions called a **decision tree**

Gist2

- Trees can be learned in a **greedy** and **recursive** manner, starting from the complete set of examples, picking a test to split it into two subsets which are as pure as possible

- The recursive process terminates when the remaining subsets are sufficiently pure

Gist3

- The abundance of memory and computing power permits training very large numbers of different trees. They can be fruitfully used as **decision forests** by collecting all outputs and averaging (regression) or voting (classification).
- They are fast and efficient thanks to their parallelism and reduced set of tests per data point.