Introduction to Machine Learning
NPFL 054

http://ufal.mff.cuni.cz/course/npfl054

Barbora Hladká
hladka@ufal.mff.cuni.cz

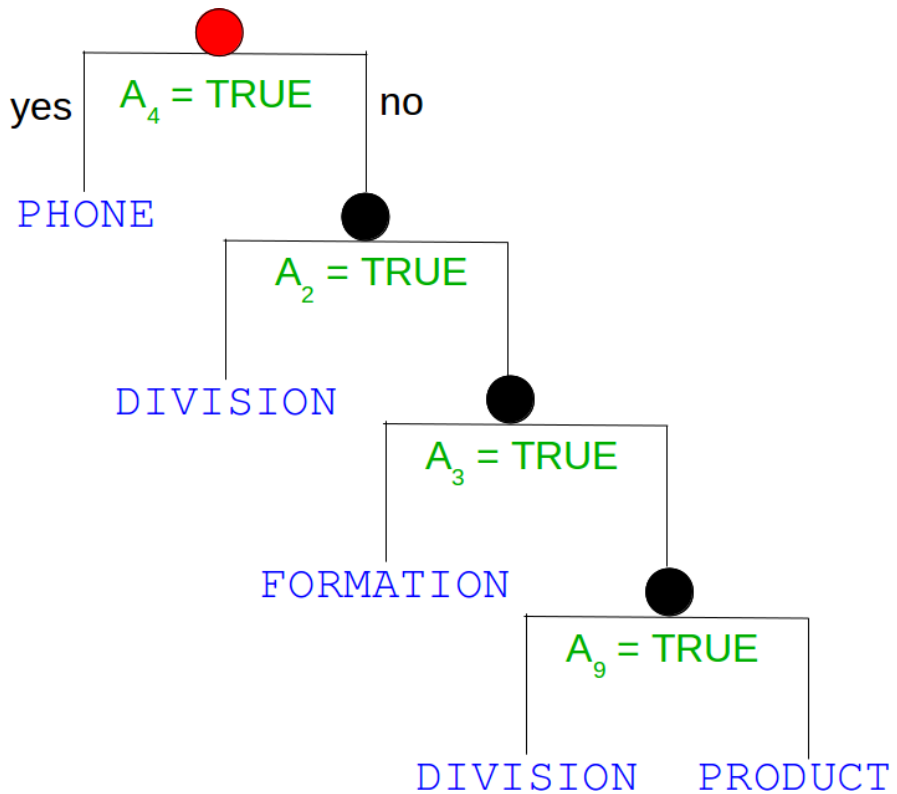
Martin Holub
holub@ufal.mff.cuni.cz

Charles University,
Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics

Decision tree structure

A **decision tree** $T = (V, E)$ is a rooted tree where V is composed of internal **decision nodes** and terminal **leaf nodes**.

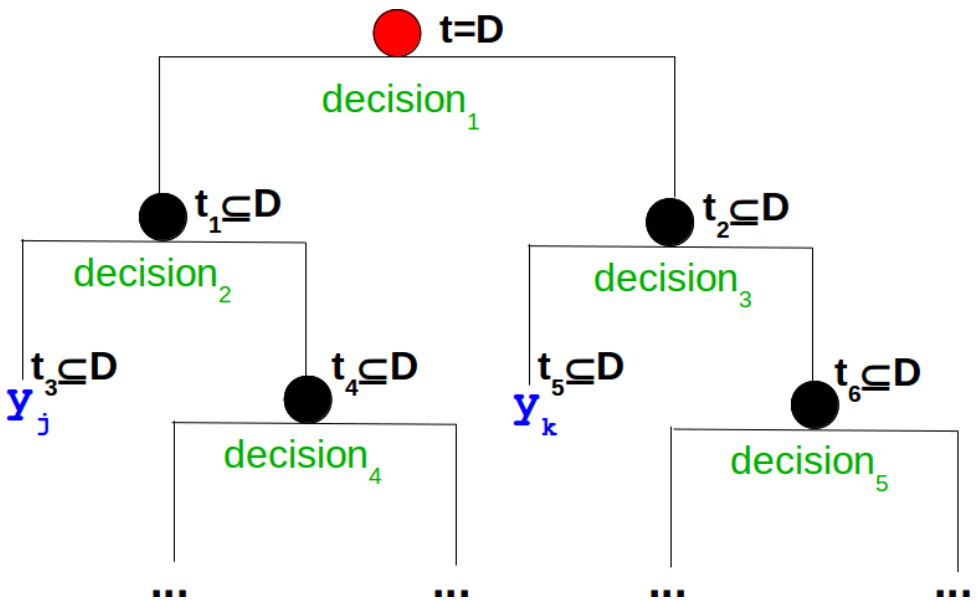
- Nodes
  - Root node
  - Internal nodes with conditions on selected features
  - Leaf nodes with `TARGET OUTPUT VALUES`
- Decisions



Decision trees — learning from training data

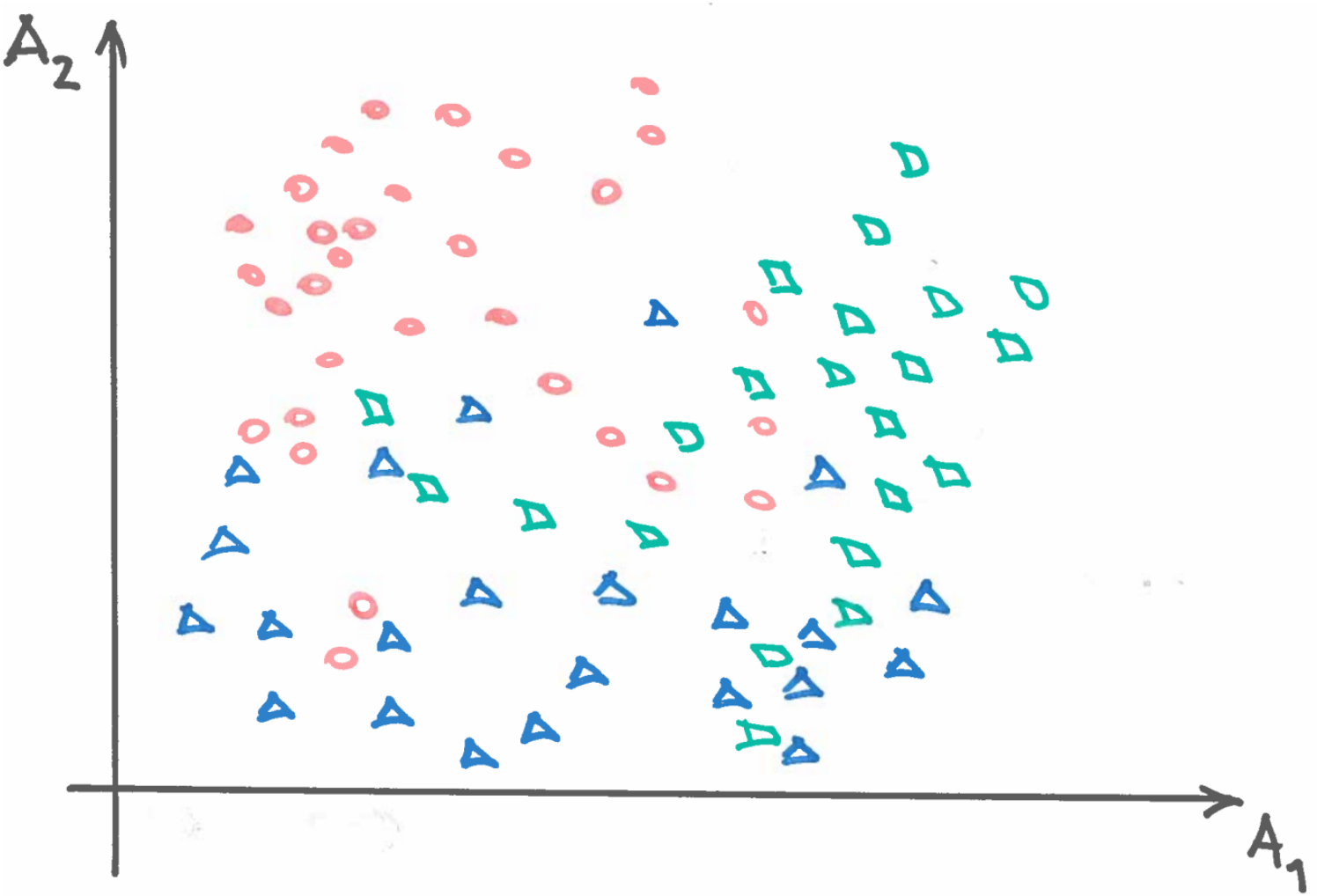
Decision tree learning

- Building a decision tree $T_D = (V, E)$ is based on a training data set $D = \{\langle \mathbf{x}, y \rangle : \mathbf{x} \in X, y \in Y\}$.
- Each node is associated with a set t, $t \subseteq D$. The root node is associated with $t = D$.
- Each leaf node is asociated with a fixed output value.



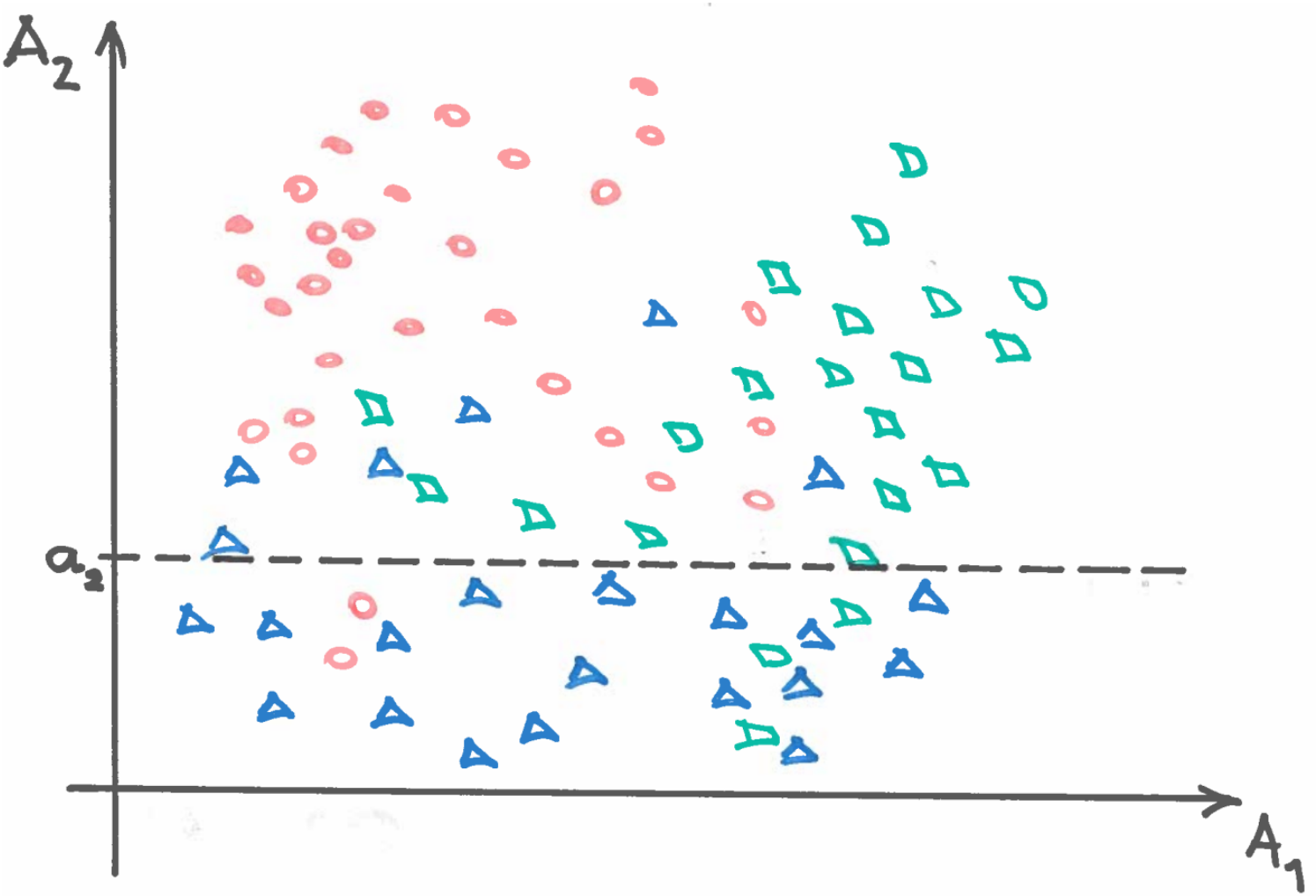
Learning decision tree – example training data

Two continuous features A_1 and A_2, and three target classes



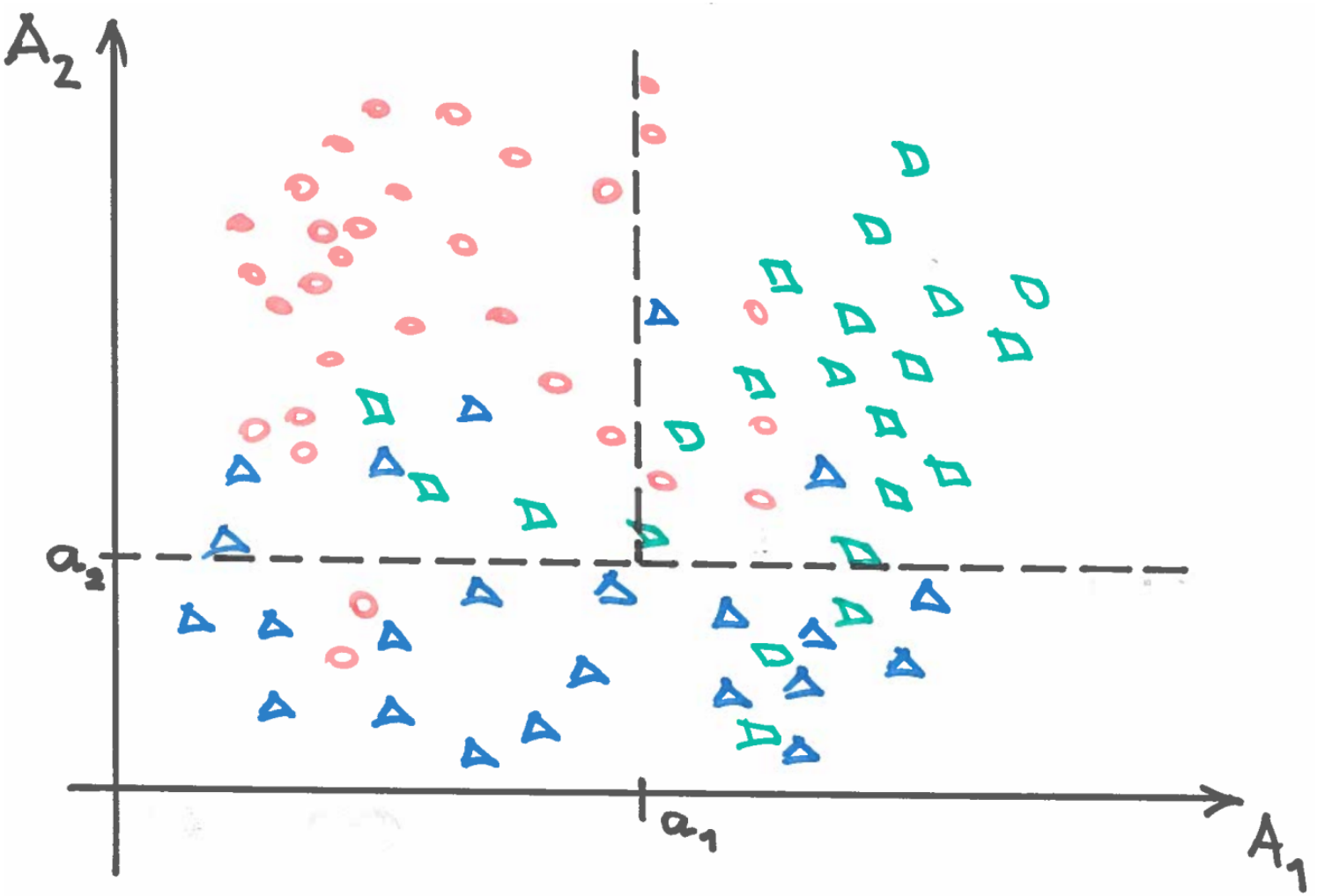
Learning decision tree – example first split

First split divides the training data set into two partitions by condition $A_2 \geq a_2$



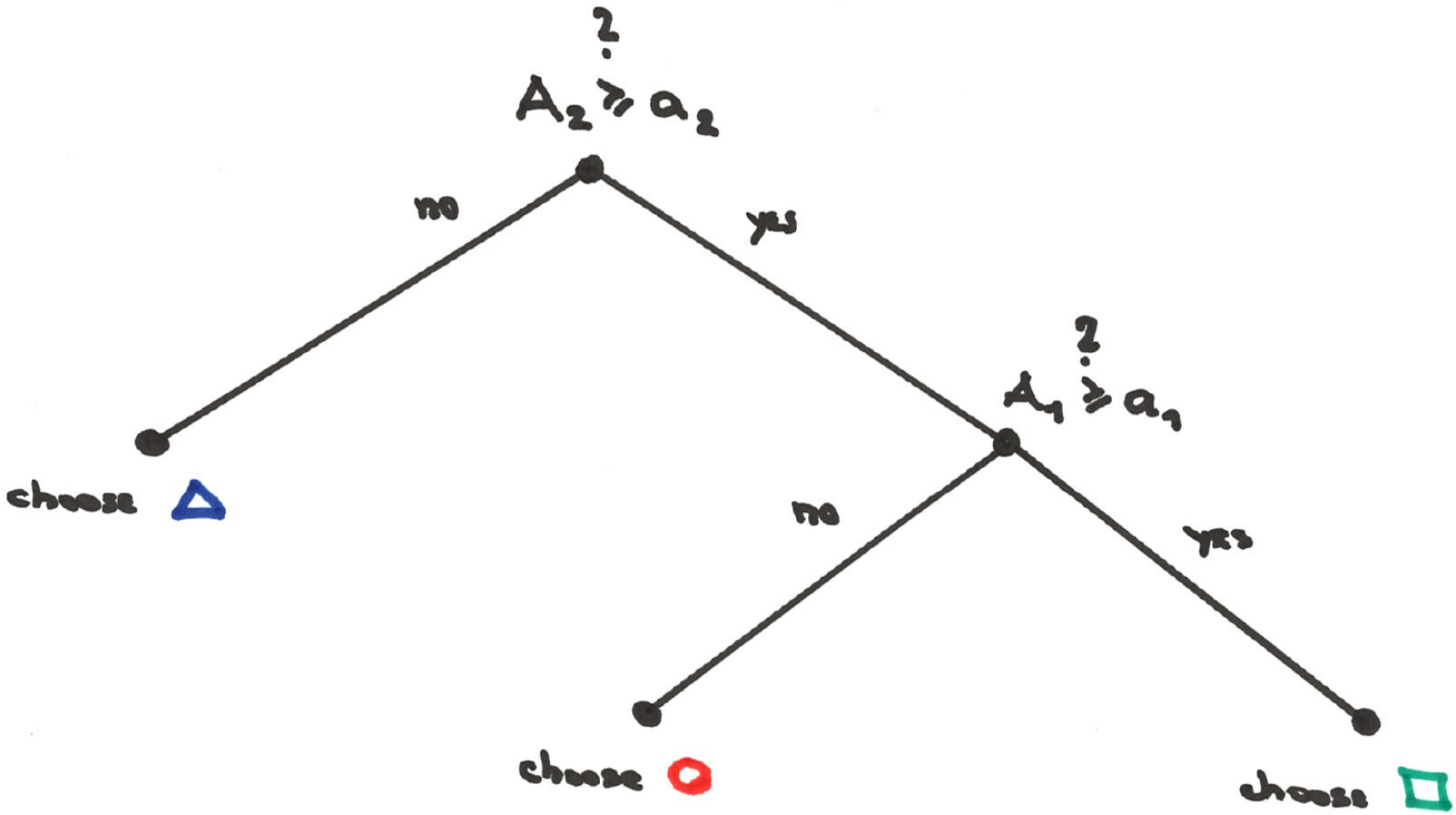
Learning decision tree – example second split

Second split is defined by $A_1 \geq a_1$ and applies only if $A_2 \geq a_2$



Learning decision tree – example resulting tree

Two splits in the example produce a tree with two inner nodes and three leaves



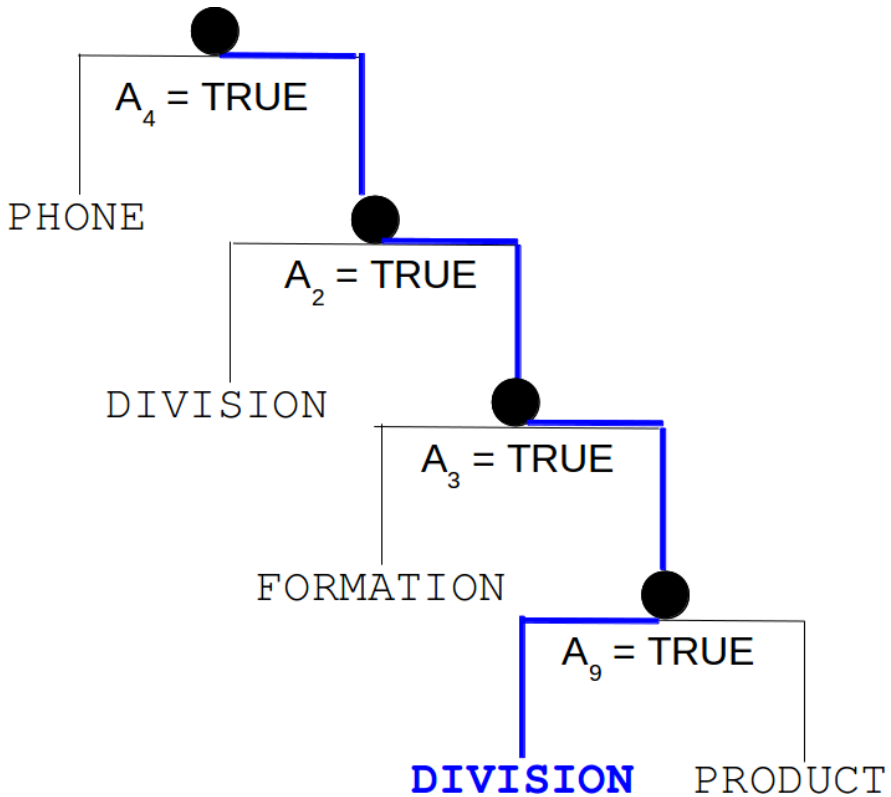
Prediction on test data

Once a decision tree predictor is built, an unseen instance is predicted by starting at the root node and moving down the tree branch corresponding to the feature values asked in decisions.

Prediction on test data – example

Decision tree predictor for the WSD-*line* task

According to existing feature values in a given test instance you can use the decision tree as a predictor to get the classification of the instance.



Decision trees for classification and for regression

Decision trees can be used both for classification and regression tasks

Classification trees

- Categorical output value

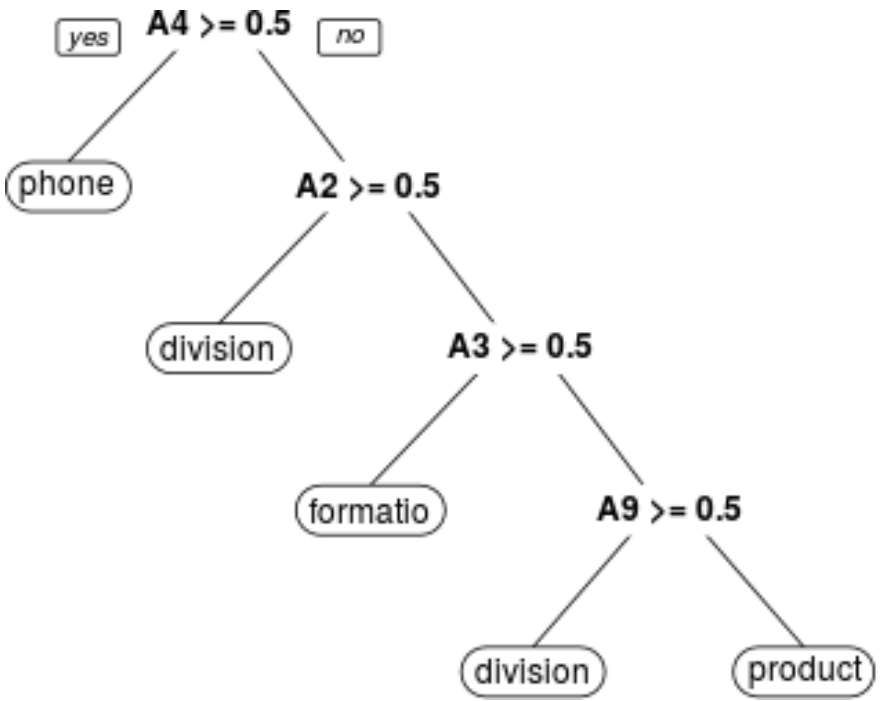


Figure: Tree for predicting the sense of *line* based on binary features.

Regression trees

- Numerical output value

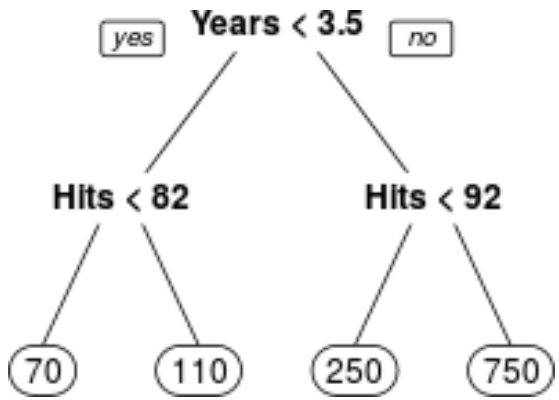


Figure: Tree for predicting the salary of a baseball player based on the number of years that he has played in the major leagues (`Year`) and the number of hits that he made in the previous year (`Hits`). See the `ISLR` `Hitters` data set.

Classification and regression trees

Each terminal node in the decision tree is associated with one of the regions in the feature space. Then

Classification trees

- **output value**: the most common class in the data associated with the terminal node

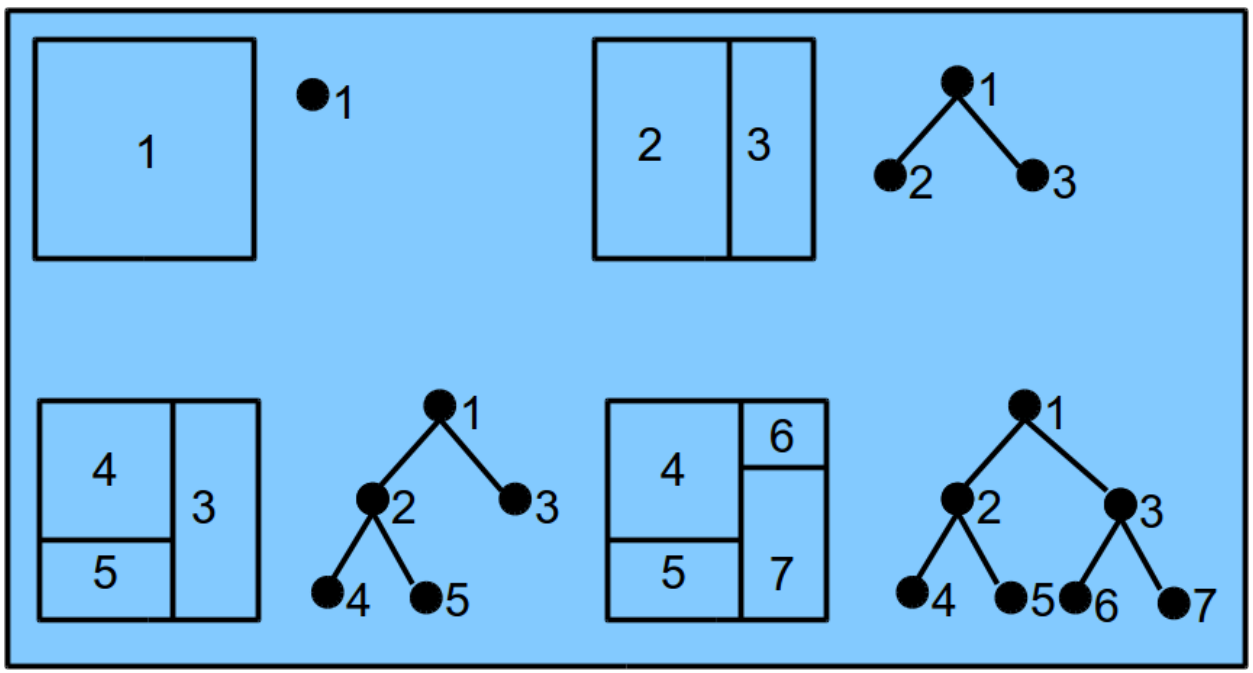
Regression trees

- **output value**: the mean output value of the training instances associated with the terminal node

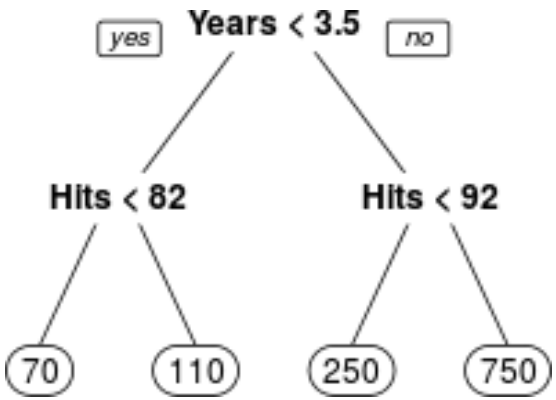
Building a tree = recursive data partitioning

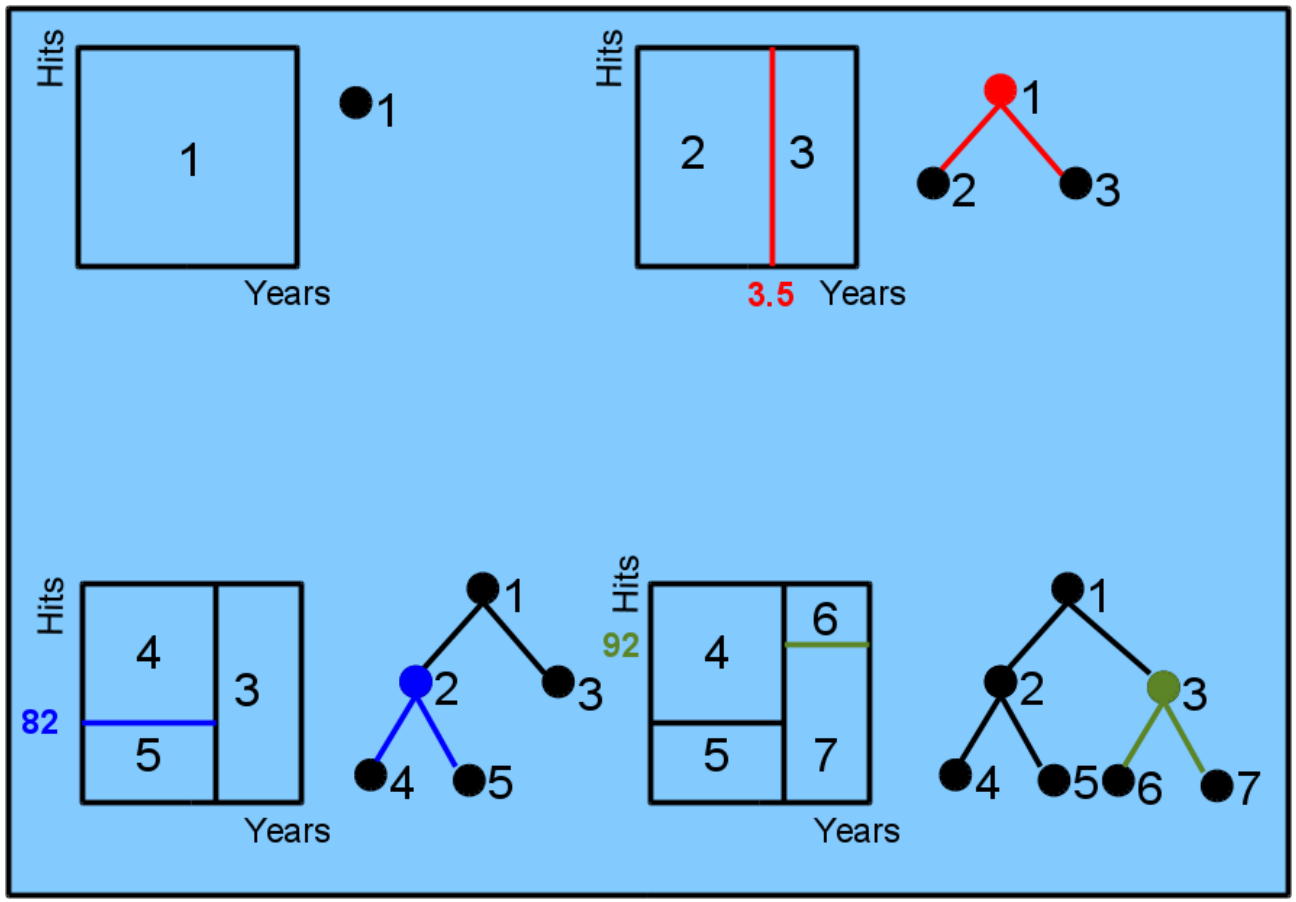
Building a decision tree is in fact a recursive partitioning process

Tree growing
The growing process is based on subdividing the feature space recursively into non-overlapping regions.

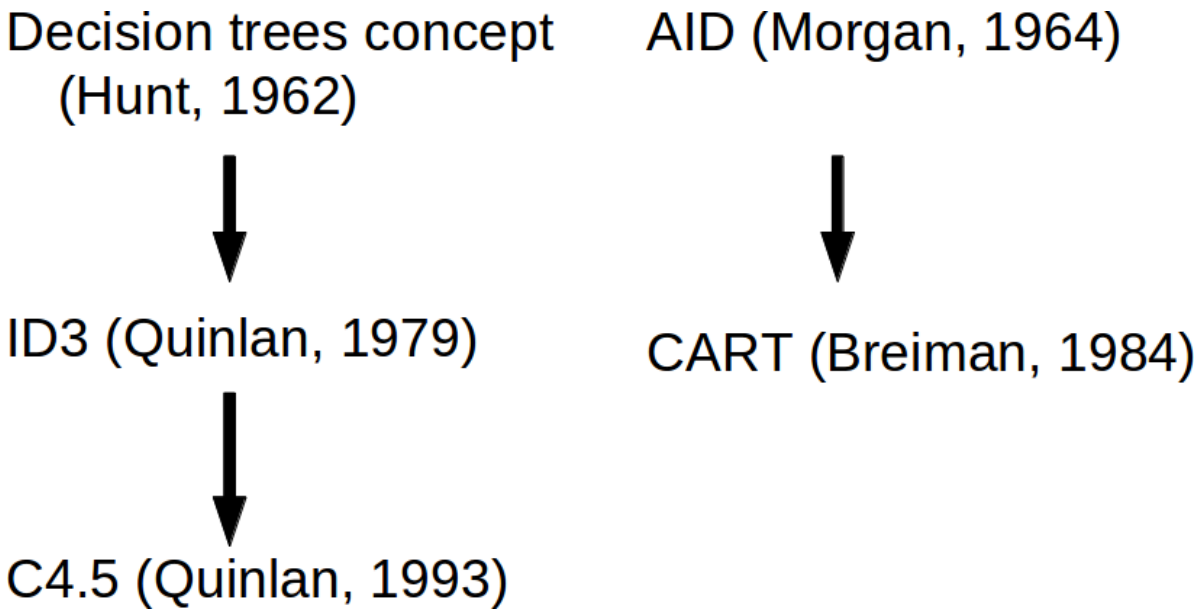


Recursive data partitioning – regression case





Historical excursion



- ID3 $\sim$ Iterative Dichotomiser
- AID $\sim$ Automatic Interaction Detection
- CART $\sim$ Classification and Regression Trees

Probably most well-known is the "C 5.0" algorithm developed by Quinlan for commercial use, which has also become the industry standard. C 5.0 is an improved extension of C 4.5. Single-threaded version is distributed under the terms of the GNU General Public License.

Learning a decision tree – key problems

Building a decision tree means to make a hierarchical sequence of splits. Each practical algorithm must be able to efficiently decide the following key questions:

(1) How to choose a suitable splitting condition?

(2) When to stop the splitting process?

A practical answer to problem (1) is to employ entropy or another similar measure. Each node is defined by an associated subset of examples with a specific distribution of target values. After a split, the entropy in child nodes should decrease in comparison with entropy in the parent node.

The splitting process should be duly stopped just to not produce model that overfits the training data. To avoid overfitting, practical implementations usually use pruning after building a relatively deep tree.