DLI Accelerated Data Science Teaching Kit

Lecture 14.6 - Decision Tree

The Accelerated Data Science Teaching Kit is licensed by NVIDIA, Georgia Institute of Technology, and Prairie View A&M University under the Creative Commons Attribution-NonCommercial 4.0 International License.

Decision Tree

Decision tree builds classification or regression models in the form of a tree structure.

A decision tree allows you to predict the value of a target variable by following the decisions in the tree from the root (beginning) down to a leaf node.

A tree consists of branching conditions where the value of a predictor is compared to a trained weight.

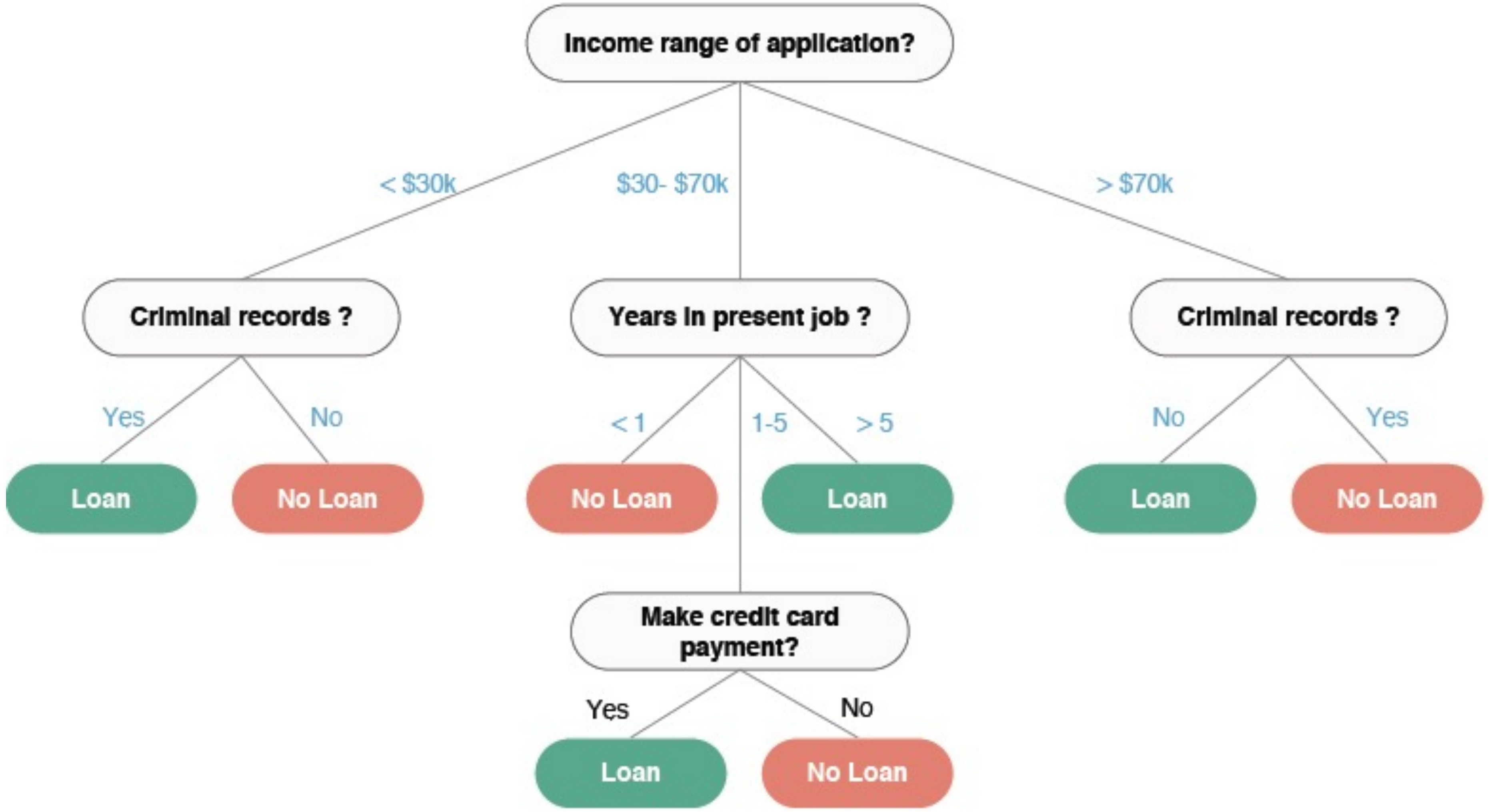
- The number of branches and the values of weights are determined in the training process.

Decision trees are prone to overfitting, additional modification, or pruning, may be used to simplify the model.

Application

Decide whether or not to offer someone a loan based on features below

- Income
- Criminal records
- Years in present job
- Payment for credit card

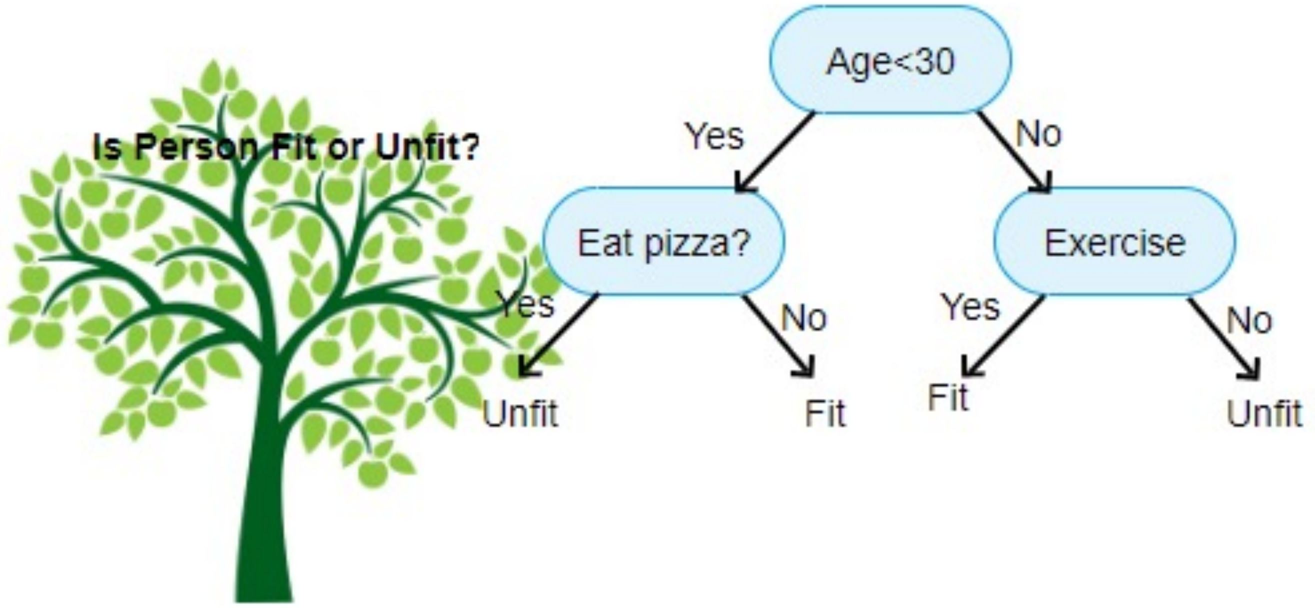


NVIDIA. DEEP LEARNING INSTITUTE | Georgia Tech | PRAIRIE VIEW A&M UNIVERSITY

Logistic Regression

Decision Tree generates the output as a tree-like structure

- Built top-down from a root node
- Break down a dataset into smaller and smaller subsets
- An associated decision tree is incrementally developed
- A tree with decision nodes and leaf nodes.
- A decision node has two or more branches.
- Leaf node represents a classification or decision.
- The topmost decision node in a tree which corresponds to the best predictor called root node.



Source: https://www.aitimejournal.com/@akshay.chavan/a-comprehensive-guide-to-decision-tree-learning

Building a Decision Tree

Strategy: top- down Recursive divide-and-conquer fashion

1. First: select attribute for root node
   - Create branch for each possible attribute value

2. Then: split the data set into subsets
   - One for each branch extending from the node

3. Finally: repeat recursively for each branch

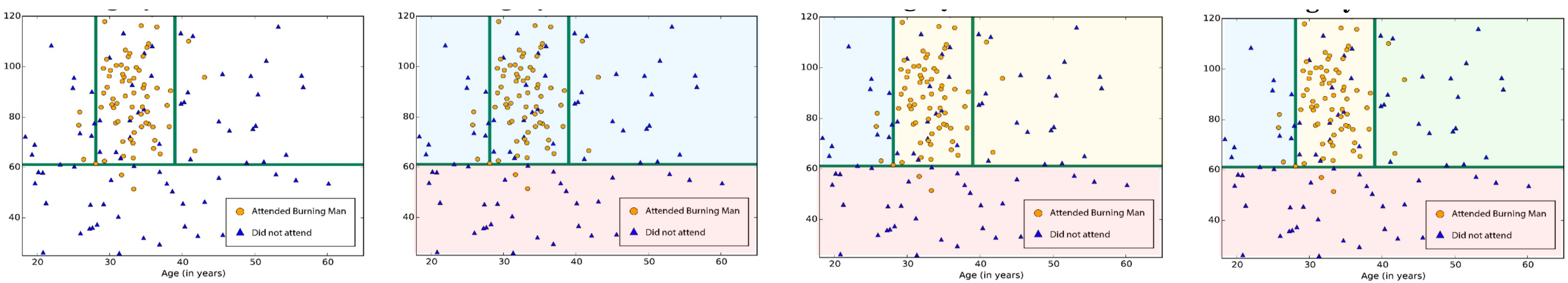
Recursively partitions the training set until each partition consists entirely or dominantly of examples from one class.

Each non-leaf node of the tree contains a split point that is a test on one or more attributes and determines how the data is partitioned.

The tree is built by recursively partitioning the data.

Splitting Data

A decision tree subdivides a feature space into regions of roughly uniform values



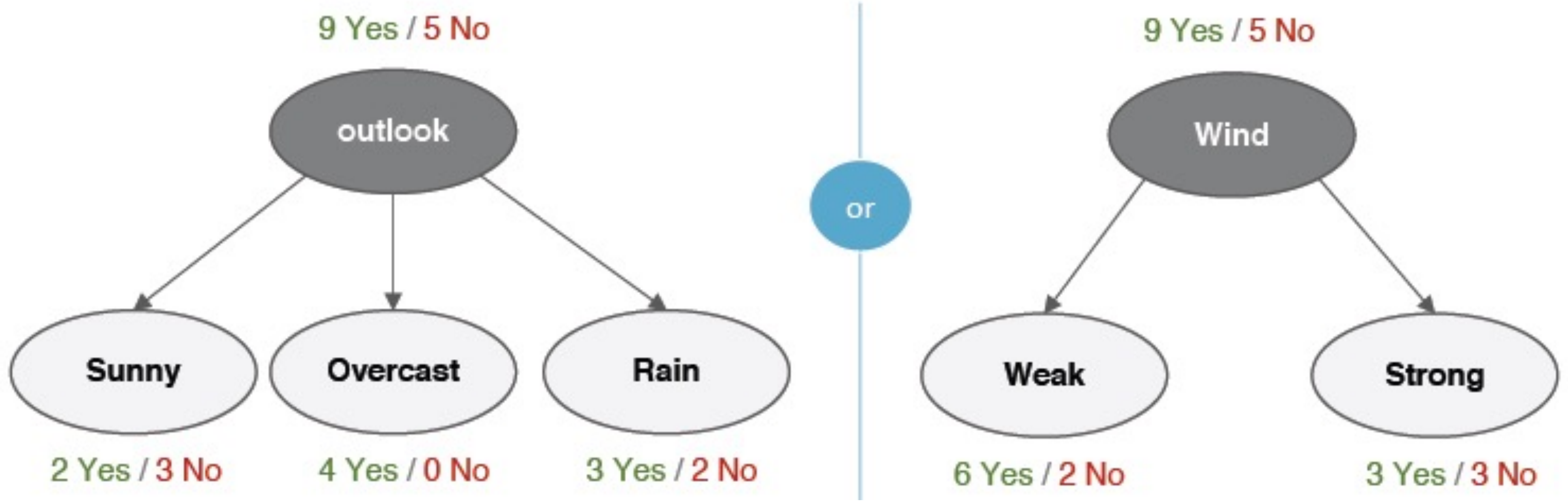
Selecting Feature

Growing a tree involves deciding on which features to choose and what conditions to use for splitting, along with knowing when to stop.

In this procedure all the features are considered, and different split points are tried and tested using a cost function.

- The split with the best cost (or lowest cost) is selected.

Compare the different ways to split data in a node



Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No

Entropy and Information Gain

Generally, entropy is a measure of disorder or uncertainty

Entropy is a concept used in Physics, mathematics, computer science (information theory) and other fields of science.

Generally, information entropy is the average amount of information conveyed by an event.

The measure of information entropy associated with each possible data value is the negative logarithm of the probability mass function for the value.

$$Entropy = -\sum_{i=1}^{n} p_i log p_i$$

Entropy and Information Gain

The measure of information entropy associated with each possible data value is the negative logarithm of the probability mass function for the value.

$$Entropy = -\sum_{i=1}^{n} p_i log p_i$$

where p_i is the probability of getting the i^{th} value when randomly selecting one from the set.

In other words, there are n classes, and p_i is the probability an object from the i^{th} class appearing.

DEEP LEARNING INSTITUTE

Entropy and Information Gain

Information gain increases with the average purity of the subsets.

- Strategy: choose attribute that gives greatest information gain.

A reduction of entropy is often called an information gain.

- Uses entropy to calculate the homogeneity of a sample.

Constructing a decision tree is all about finding attribute that returns the highest information gain (i.e., the most homogeneous branches)

- A decision tree is built to up-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogenous).
- The information gain is based on the decrease in entropy after a dataset is split on an attribute..

Entropy and Information Gain

The goal is to decrease in entropy (uncertainty) after splitting.

$$Information\ Gain(S, A) = \overset{\substack{\textbf{Entropy before splitting} \\ \text{(parent)}}}{H(S)} - \underbrace{\sum_{v \in Value\,(A)} \frac{|s_v|}{|s|} H(s_v)}_{\substack{\textbf{Entropy after splitting} \\ \text{(Weighted) Average Entropy of Children}}}$$

Where

v : *possible value of* A

S: *set of example* $\{x\}$

s_v: *subset where* $x_A = v$

Example

Data

Example	crust size	shape	filling size	Class
$e1$	big	circle	small	**pos**
$e2$	small	circle	small	**pos**
$e3$	big	square	small	**neg**
$e4$	big	triangle	small	**neg**
$e5$	big	square	big	**pos**
$e6$	small	square	small	**neg**
$e7$	small	square	big	**pos**
$e8$	big	circle	big	**pos**

$$\begin{aligned} H(T) &= -p_{\text{pos}} \log_2 p_{\text{pos}} - p_{\text{neg}} \log_2 p_{\text{neg}} \\ &= -(5/8)\log(5/8) - (3/8)\log(3/8) = 0.954 \end{aligned}$$

$$\begin{aligned} H(\texttt{shape=square}) &= -(2/4)\cdot\log(2/4) - (2/4)\cdot\log(2/4) = 1 \\ H(\texttt{shape=circle}) &= -(3/3)\cdot\log(3/3) - (0/3)\cdot\log(0/3) = 0 \\ H(\texttt{shape=triangle}) &= -(0/1)\cdot\log(0/1) - (1/1)\cdot\log(1/1) = 0 \end{aligned}$$

$$H(T, \texttt{shape}) = (4/8)\cdot 1 + (3/8)\cdot 0 + (1/8)\cdot 0 = 0.5$$

Example

Data

Example	crust size	shape	filling size	Class
$e1$	big	circle	small	**pos**
$e2$	small	circle	small	**pos**
$e3$	big	square	small	**neg**
$e4$	big	triangle	small	**neg**
$e5$	big	square	big	**pos**
$e6$	small	square	small	**neg**
$e7$	small	square	big	**pos**
$e8$	big	circle	big	**pos**

$$H(T, \texttt{crust}-\texttt{size}) = 0.951$$
$$H(T, \texttt{filling}-\texttt{size}) = 0.607$$

$$\begin{aligned}
I(T, \texttt{shape}) &= H(T) - H(T, \texttt{shape}) &&= 0.954 - 0.5 &&= 0.454 \\
I(T, \texttt{crust}-\texttt{size}) &= H(T) - H(T, \texttt{crust}-\texttt{size}) &&= 0.954 - 0.951 &&= 0.003 \\
I(T, \texttt{filling}-\texttt{size}) &= H(T) - H(T, \texttt{filling}-\texttt{size}) &&= 0.954 - 0.607 &&= 0.347
\end{aligned}$$

DLI Accelerated Data Science Teaching Kit

Thank You