Technische Hochschule Rosenheim

Chapter 03 – Decision Tree Classification

Lecture A2I2

Prof. Dr. Kai Höfig

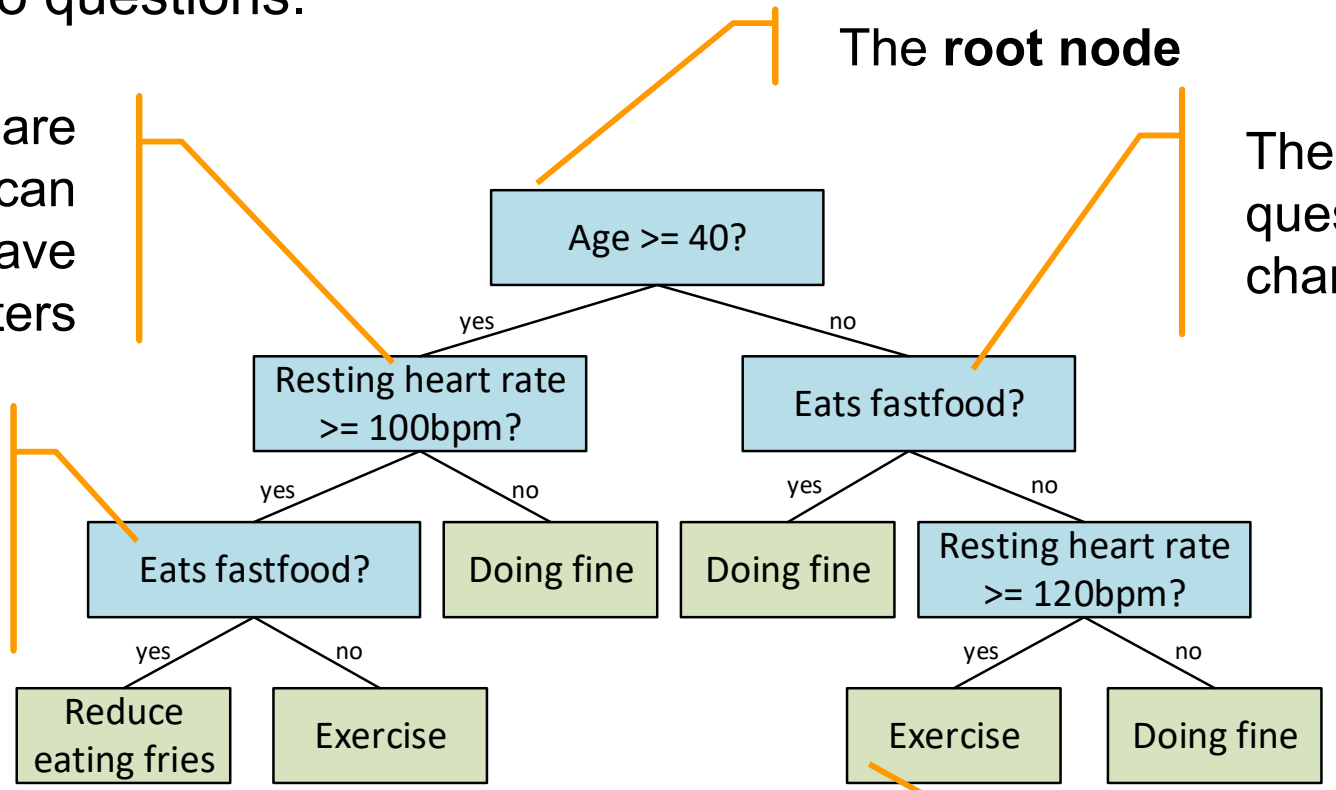
Goals

After this chapter you should

- Be able to read a decision tree
- Understand how decision tree architectures can be learned from data
- Transform a dataset using pandas in a way that a scikit learn decision tree classifier can be trained
- Train a scikit learn decision tree classifier and select appropriate parameters
- Test and visualize the generated tree
- Classify samples

Decision Tree, a simple example

- A (Binary) Decision Tree is a way of classifying data into categories using simple yes/no questions.

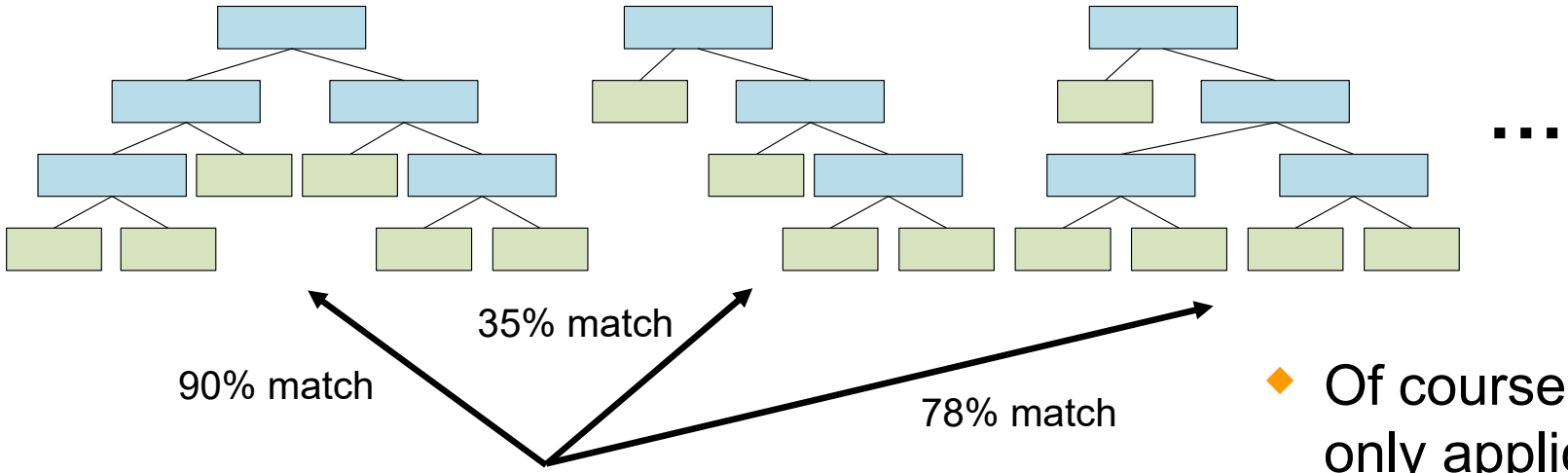


A **sample**, either to be categorized or for training

Joshua
Age 22
Loves fries
Heart rate 130bpm
Classification: He is doing fine!

Decision Tree Learning

- Decision Trees can be learned from training data.
- A simple and optimal algorithm would be to generate **all possible trees** from the parameters and put the categories accordingly to training data.



Sample	Age	Bpm	Fries	Class
Joshua	22	130	Yes	fine
Brittany	44	110	No	exercise
..	..	..	..	..

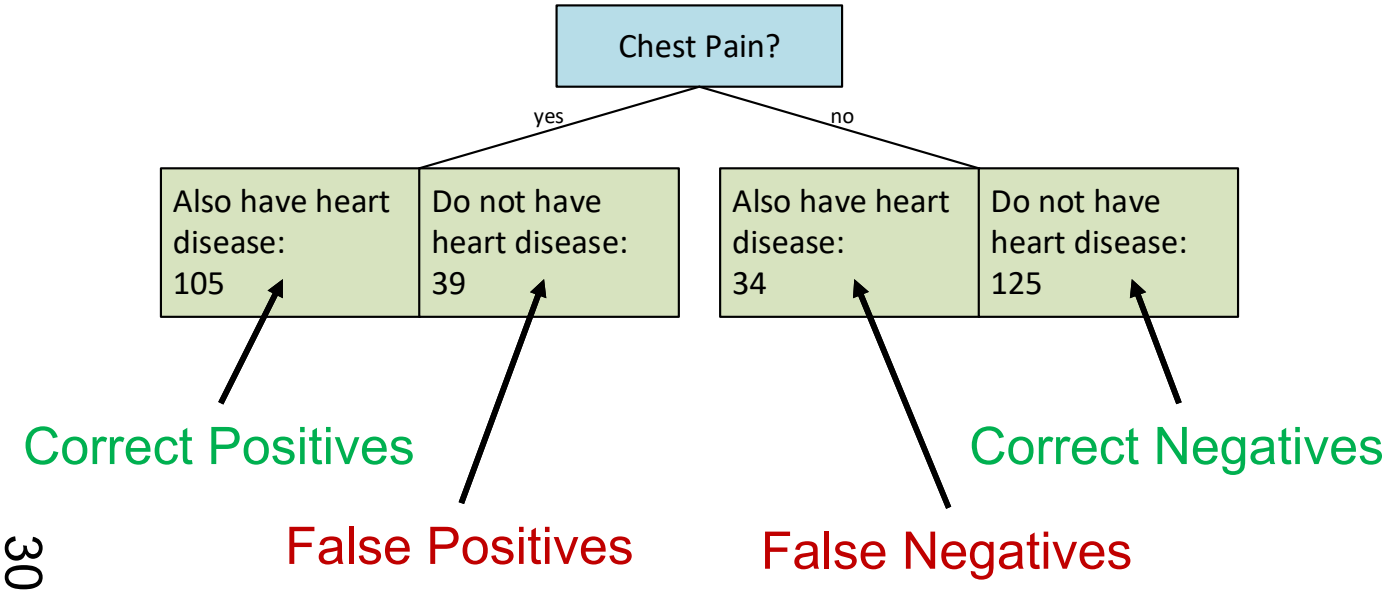
- Of course, this algorithm is only applicable in theory and is most probably in NP
- We now use **Gini** and **Entropy**-values in a different approach

1st Problem – Finding the Root Node

- If we take the following data and we want to classify a person according to these parameters to decide weather a person has a heard disease or not, the first question to answer if we build a decision tree is which parameter is measured at the root node of our decision tree?

Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	No
..	..	..	..

303 Samples

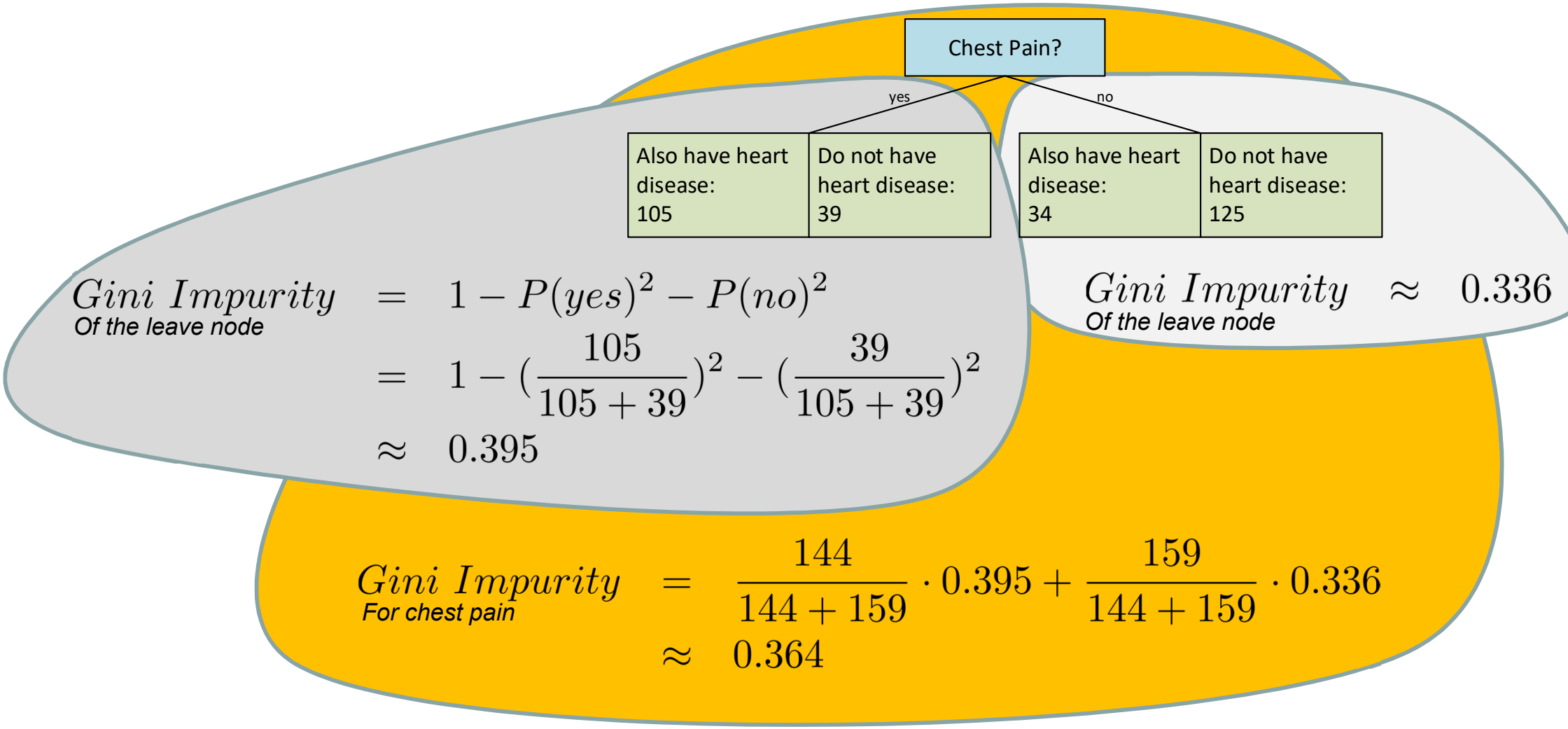


"if that would be our test"

How good is that? Its bad, because its not 100%

Gini Impurity

- Chest pain is an **impure** node. It does not divide the samples 100% correctly into the classes *heard disease* and *no heart disease.*



$$Gini\ Impurity = 1 - P(yes)^2 - P(no)^2$$
$$= 1 - (\frac{105}{105+39})^2 - (\frac{39}{105+39})^2$$
$$\approx 0.395$$

Of the leave node

$$Gini\ Impurity \approx 0.336$$

Of the leave node

$$Gini\ Impurity = \frac{144}{144+159} \cdot 0.395 + \frac{159}{144+159} \cdot 0.336$$
$$\approx 0.364$$

For chest pain

Finding the root node

- As a first step, we identify the root node of the decision tree we are about to build using the lowest Gini Impurity value.

$$
\begin{aligned}
G(Chest\ Pain) &= 0.364 \\
G(Good\ Blood\ Circulation) &= 0.360 \longrightarrow \\
G(Blocked\ Arteries) &= 0.381
\end{aligned}
$$

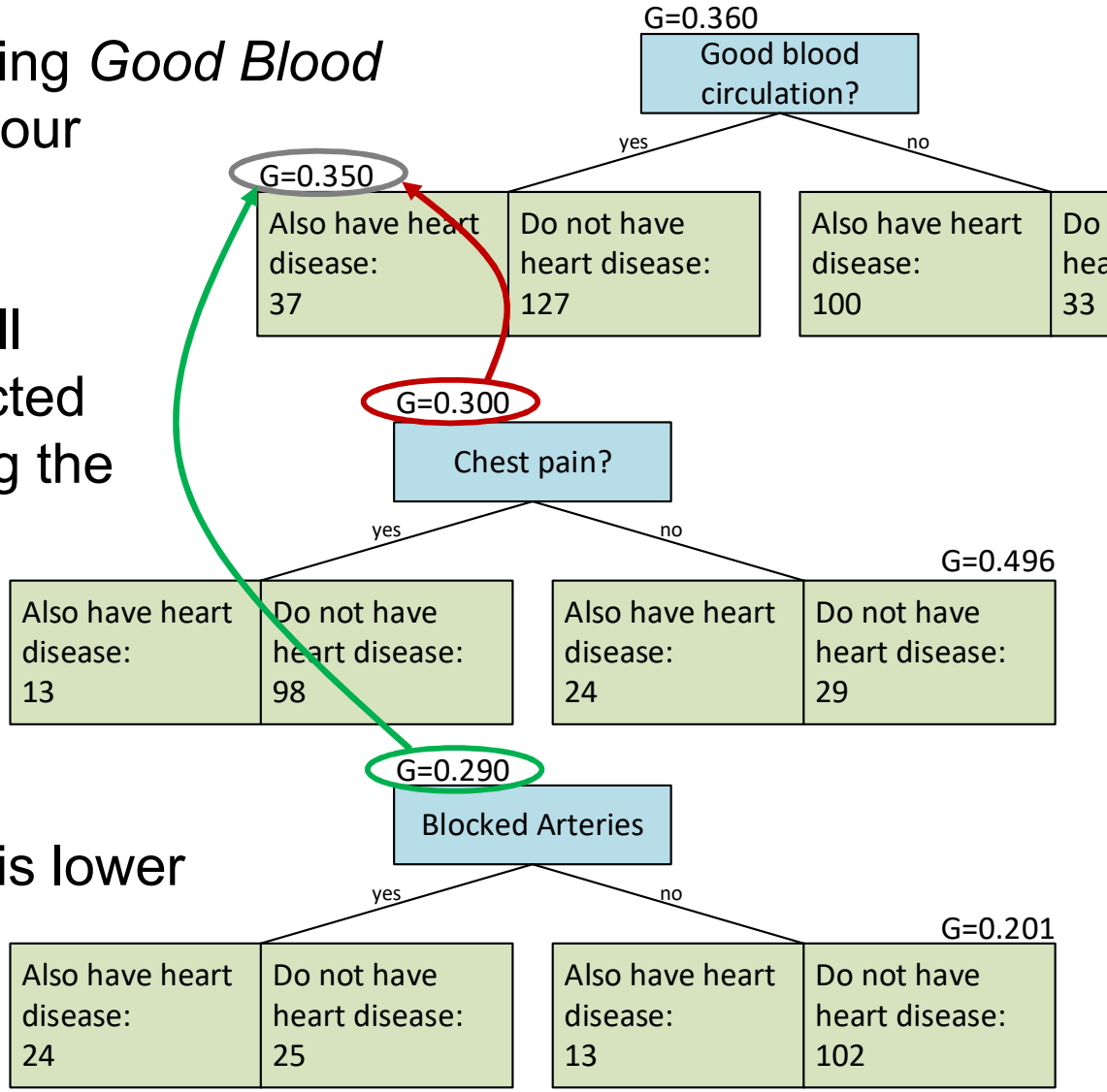
Lowest Gini Impurity, it separates patients with and without heart disease the best.

- In a way, Gini impurity gives us the probability, that a sample is misclassified. So we want the lowest misclassification probability uppermost in the tree, because from there, it can only get better.

- In a next step, we repeat this calculation for the new leave nodes

2nd Problem – Finding the next node

- From the last step, we know that taking *Good Blood Circulation* is the best root node for our decision tree.
- Now calculate the Gini impurity for all parameters that have not been selected so far on the way up to the root using the data valid for this node.
- Select the parameter with the lowest impurity.
- If the impurity of the new parameter is lower than the impurity of the current node, use the parameter, otherwise make the current node a leaf.



How to deal with **numeric** data?

- Numeric data can be sorted from low to high and than calculate the Gini value for each possible cut. The best cut is then used. All other cuts stay in the algorithm and can be used later on for further nodes on the same path up to the root node.

Weight	Heart disease
155 lbs	No
180 lbs	Yes
190 lbs	No
220 lbs	yes

~~Weight <= 167.5?~~ → G=0.392

~~Weight <= 185.0?~~ → G=0.281

~~Weight <= 205.0?~~ → G=0.121

How to deal with **ordinal** data?

- Ranked data can be treated in the same way as numeric data is processed. The only difference is, that mean values are not calculated, since they do not exist in ordinal scales.

Dress size	Heart disease
S	No
L	Yes
XL	No
XXL	yes

Dress size <=S? → G=0.392

Dress size <=L? → G=0.281

Dress size <=XL? → G=0.311

How to deal with **categoric** data?

- For categoric data, we can generate nodes that represent real subsets of the categories.

Favorite color	Heart disease
red	No
green	Yes
blue	No
..	..

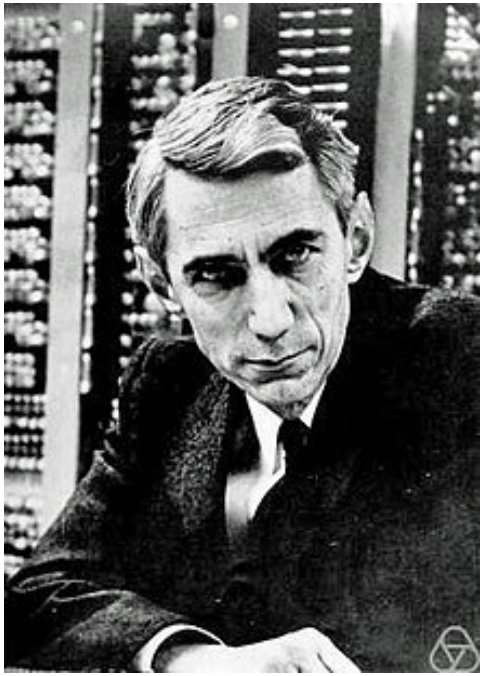
Favorite color = red?	G=0.392
Favorite color = green?	G=0.441
Favorite color = blue?	G=0.311
Favorite color = red or green?	G=0.392
Favorite color = green or blue?	G=0.301
Favorite color = blue or red?	G=0.311

Decision Tree Learning Algorithms

- CART (Classification and regression tree), see L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone: *CART: Classification and Regression Trees*. Wadsworth: Belmont, CA, **1984**. *based on Gini*
- ID 3 Algorithm (Iterative Dichotomiser 3), see Quinlan, J. R. 1986. Induction of Decision Trees. Mach. Learn. 1, 1 (Mar. **1986**), 81–106 *based on Entropy*
- C4.5 Algorithm, see Quinlan, J. R. C4.5: *Programs for Machine Learning*. Morgan Kaufmann Publishers, **1993**. *based on Entropy*
- C5.0 Algorithm, https://rulequest.com/download.html, latest version 2010, *based on Entropy*

Claude Shannon

- A genius Rockstar in math born 1916 in Petoskey, Michigan that found a measure for *how much the fu** is going on*!
  - How much does it surprise you that I found these pictures?
  - How much does it surprise you that the house with the two ‘H’ letters is still there?



Claude Shannon at the age of 47

Mitchell Street in Petoskey 1919

Mitchell Street in Petoskey 2020

Entropy

- *Entropy* is a concept from thermodynamics and is expressed in Joule per Kelvin. It sometimes is described as a measure of chaos.

Ice, low entropy

Liquid water, medium entropy

Water steam, high entropy

Entropy in Information Theory

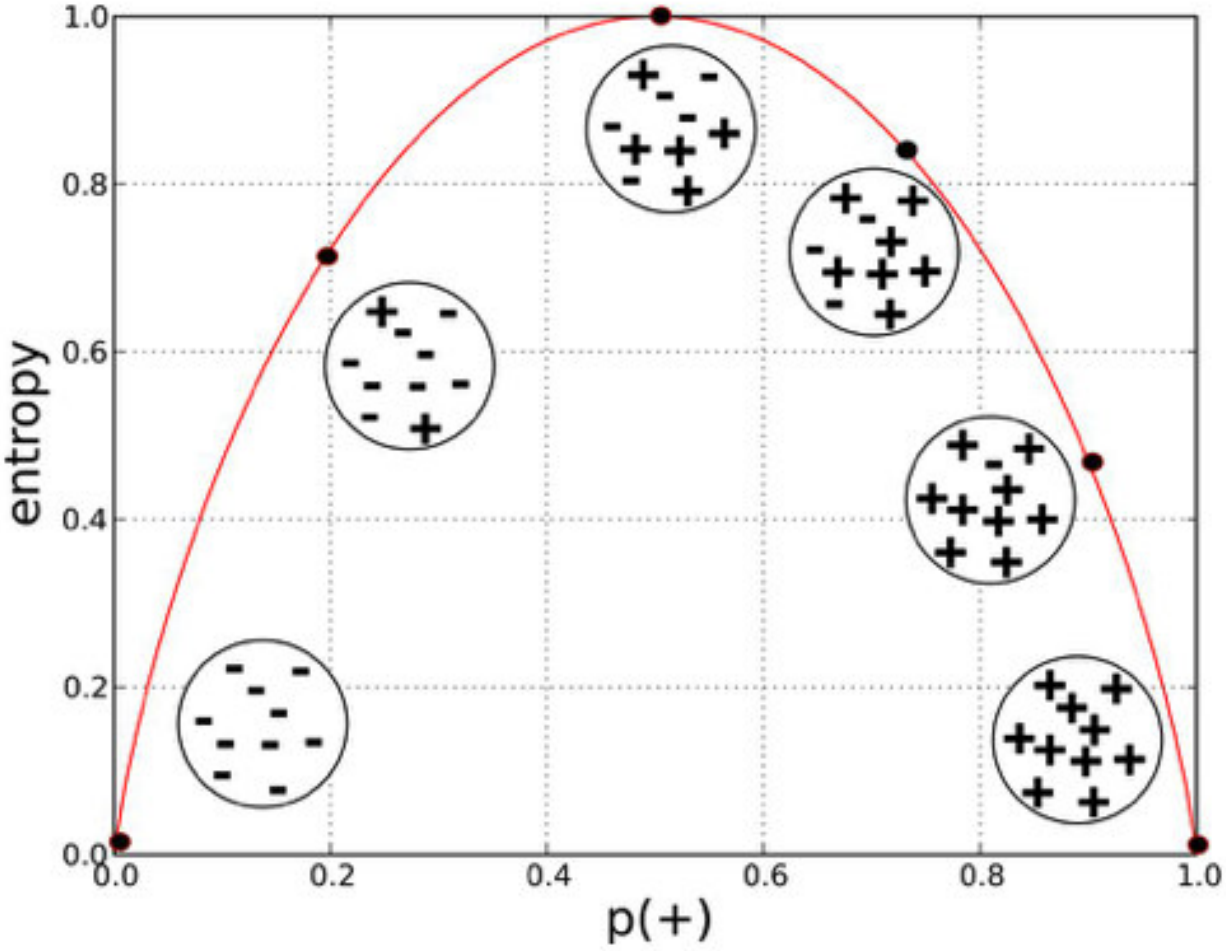
- Entropy in information theory is a measure of *how surprising* an event is.

If I pull from the nuts, I will not be very surprised: low entropy.

If I pull from the jelly beans I will be surprised a little more, medium entropy.

If I pull from the toys, I will be outrageously surprised, high entropy.

Distribution of Entropy



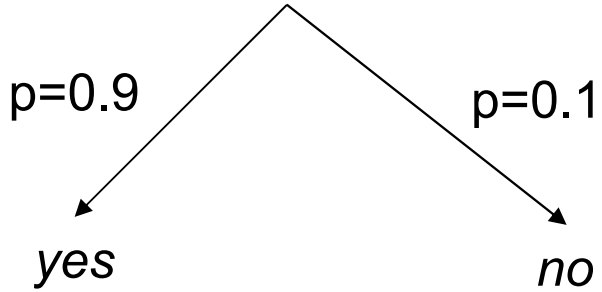
Provost, Foster; Fawcett, Tom. Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking

A stupid little entropy game

- I am pulling one ball from a bucket with only blanks.
- I don’t show you what I got.
- How many questions do you have to ask on average to know for sure what I pulled?
  - *Answer 0, well, that was easy.*
  - *Entropy is 0.*

Still a stupid little entropy game

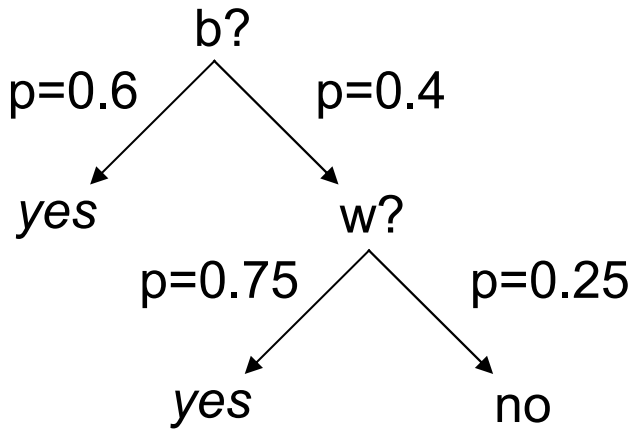
- I am pulling one ball from a bucket with 9 blanks and one winning
- I don't show you what I got.
- How many questions do you have to ask on average to know for sure what I pulled?
  - *Answer 1 question: did you pull a blank?*
  - *Entropy is 1.*



$$H = 0.9 \cdot 1 + 0.1 \cdot 1 = 1$$

A not so stupid little entropy game

- I am pulling one ball from a bucket with six blanks, three winnings and a jackpot



$$
\begin{aligned}
\Omega &= (b, w, j) \\
p_b &= 0.6 \\
p_w &= 0.3 \\
p_j &= 0.1
\end{aligned}
$$

$$
\begin{aligned}
H &= 0.6 \cdot 1 + \\
&\quad 0.4 \cdot 0.75 \cdot 2 + \\
&\quad 0.4 \cdot 0.25 \cdot 2 \\
&= 1.4
\end{aligned}
$$

$$H = -((0.6 \cdot log_2(0.6) + 0.3 \cdot log_2(0.3) + 0.1 \cdot log_2(0.1)) \approx 1.3$$

The formula for entropy

- For a discrete finite probability space

$$\Omega = (\omega_1, \ldots, \omega_n)$$

- And a random variable representing a certain event

$$P(X = \omega_i) = p_i$$

- The entropy is defined as

$$H(p_1, \ldots, p_n) = -\sum_{i=1}^{n} p_i \cdot log_2(p_i)$$

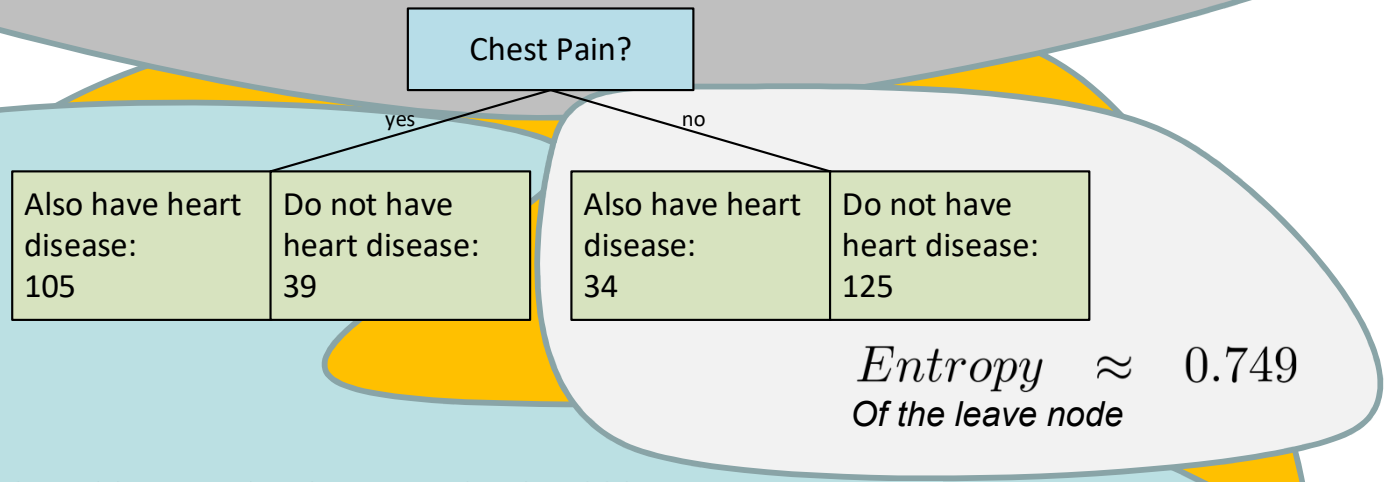
The entropy is a measure of chaos. It correlates with the number of questions to be asked to know the result of an experiment.

Using Entropy to Learn a Tree

- Basically the same algorithm as for Gini, but with a different parameter calculation

$$\begin{aligned} Entropy &= -(P(yes) \cdot log_2(P(yes)) + P(no) \cdot log_2(P(no))) \\ &= -((\frac{105+34}{105+39+34+125}) \cdot log_2(\frac{139}{303}) + (\frac{39+125}{303}) \cdot log_2(\frac{164}{303})) \\ &\approx 0.995 \end{aligned}$$

Of the root node



$$Entropy \approx 0.749$$

Of the leave node

$$\begin{aligned} Entropy &= -(P(yes) \cdot log_2(P(yes)) + P(no) \cdot log_2(P(no))) \\ &= -((\frac{105}{105+39}) \cdot log_2(\frac{105}{105+39}) + (\frac{39}{105+39}) \cdot log_2(\frac{39}{105+39})) \\ &\approx 0.842 \end{aligned}$$

Of the leave node

$$\begin{aligned} Information\ gain &= 0.995 - (\frac{144}{144+159} \cdot 0.842 + \frac{159}{144+159} \cdot 0.749) \\ &\approx 0.202 \end{aligned}$$

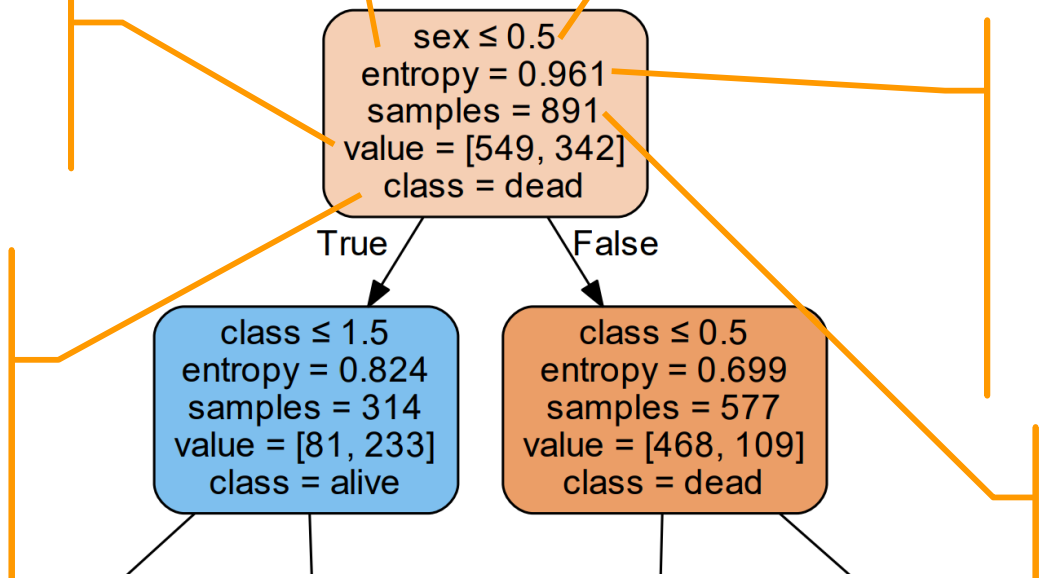
For chest pain

What do we see in the decision tree graph of scikit learn?

These numbers belong to the node and therefore have nothing to do with the question asked above them!

549 of the samples are dead and 342 survived

If classification would take place at this node, a sample would be classified as dead (majority of the values above). Since this is the root node, any person would be classified as dead.



Question asked at the current node

Entropy or Gini value for the current node. This is not the information gain nor the Normalized Gini value

Total numbers of samples left at this node. Here we have a total of 891 passengers, since this is the root node

For this subtree, any women survives and any men dies.

Task – Train your own Decision Tree

- Take a look at the *Decision Tree – Iris* Jupyter notebook and take the *Decision Tree – Titanic* notebook as a template

1. Create a decision tree, that can classify a given size of a flower into the iris categories *setosa*, *versicolor* and *virginica*
2. Print the tree
3. How good is the prediction? What can we expect when we look at the gini parameters?
4. Partition the given data set into *train* and *test* so that we can see how good the method actually applies to the classification problem.

5. *Additional: calculate the fraction of correct predictions per category*

Summary

- The solution to a classification problem using decision trees is not only available as a program, but can also be
  - Visualized
  - Can easily be reviewed
  - Is good to comprehend
- Decision trees are easy to calculate
- Differences in the decision tree learning algorithms result in shorter and more effective trees but do not influence accuracy coarsely.

https://rulequest.com/see5-comparison.html