DIGITAL CULTURE
ITMO UNIVERSITY

Entropy. Decision Trees

High School of Digital Culture
ITMO University
dc@itmo.ru

Contents

1 Introduction

Hello everyone! This module will cover decision trees being one of the most popular and effective techniques used to solve classification problems. For ease of understanding, you can think of decision trees as detailed instructions on what to do and when. Look at the example in fig. **??**. What you see is a simplified algorithm for the decision-making of a human who is trying to figure out whether to accept or decline a job offer. The example shows the definite advantage of decision

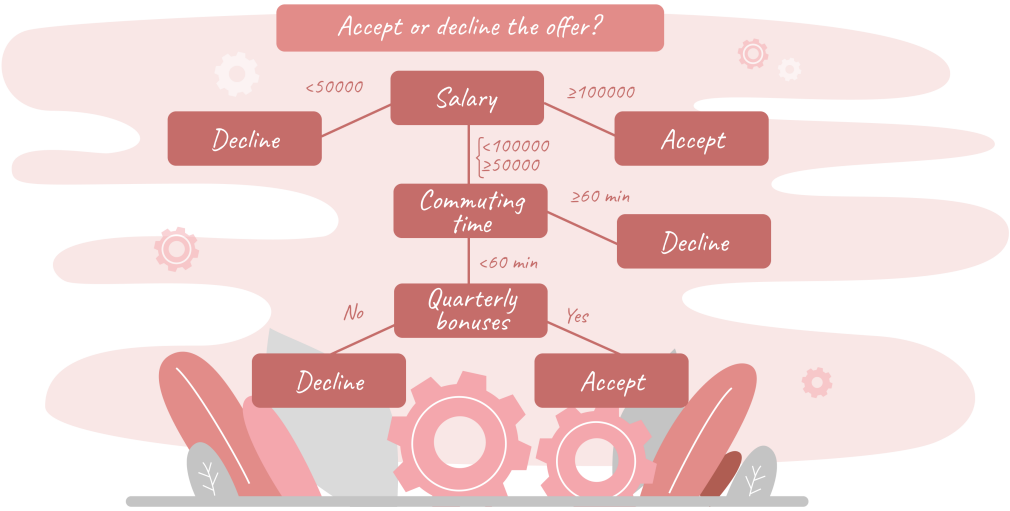

Figure 1: An example of a decision tree.

trees that gained them popularity not only in machine learning. Decision trees are intuitive, and there's no need to interpret them. According to the described algorithm, the person estimates the offered salary and then chooses between three options. If the offered salary is less than 50 thousand, the job offer is declined, but if more than or equal to 100 thousand, the job offer is accepted right away. If the offered salary is between 50 and 100 thousand, the choice is not so obvious. To decide, it's necessary to find out how much time it takes to commute. However, if it takes more than an hour to get to work, the offer is not good enough and declined. If you look lower, you'll see that the decision depends on whether quarterly bonuses are paid. If yes, the offer is accepted, if no, declined.

Despite the simplicity, the described algorithm raises questions. For example, what to do when there's no data on quarterly bonuses? The described algorithm doesn't offer the option 'data unavailable'. There are only two possible responses: yes or no. But what if we are talking about a job in delivery? Sometimes it will take less than an hour to get home when, for example, the delivery point is nearby, and sometimes it will take more than an hour when the delivery point is far away. Not to mention that some jobs require occasional business trips. The algorithm cannot handle these questions, because each of them should be considered on a

case-by-case basis. This module will cover so-called binary decision trees that help to solve the described problems.

As you may have understood from the example, it is important to prioritize the features when creating a tree. In the given example, the most important feature is a salary. Commuting time takes second place, and the last one is quarterly bonuses. At the same time, features and their priorities heavily depend on a particular problem. To decide on a loan approval for a customer, banks first consider the age of the customer, then the income, education, marital status, and then everything else. It's logical and reasonable but, the more features there are, the more difficult it is to make the right series. We need an optimal algorithm that works fast rather than a life-long series of questions.

Let's clarify this using a simple Yes or No question game or a 20-question game. The main idea of these two games is the same. Players ask the answerer who answers with a simple Yes or No. Players who know the theme of the game should guess the film, TV series, or anything else. When the aim is to guess the actor, we eliminate many possibilities by asking about the gender, but when we ask, "Is this Nicolas Cage?", we exclude only one. This intuitively corresponds to the concept of the information gain based on entropy, which measures the uncertainty of an event or experiment. The first question provided us with more information because we excluded many incorrect options, and the experiment uncertainty has significantly decreased. Uncertainty barely changed after the second question because only one option was crossed-off.

Thus, before creating a decision tree, it's important to consider a series of features very carefully. For this purpose, we can separate informative features from those less informative. But what does informative mean? How can we measure it? Let's discuss these questions first.

2 Entropy

2.1 About Degree of Uncertainty

We often think about different events in everyday life. Sometimes we are more certain about outcomes than we are at other times. For example, we know that most people commute during rush hour, so it's better not to travel at that time, meanwhile, three hours later, there are no traffic jams but who wants to go home late at night. When a child asks about a black bird that snatches food from pigeons, we will likely say that that it's a crow. What are we trying to say? Well, not all these cases are truly questionable, and there is not much uncertainty.

At the same time, there are opposite examples too. For example, will the person who you will see first tomorrow morning after you leave home be male or female? Or, for example, when you're looking at people around you in the metro

car, can you tell who will go to the furthest station? Probably, not. In the cases described, uncertainty is high.

Thus, it's important to construct a mathematical model of the uncertainty. In particular, we want to be able to obtain a numerical value of the uncertainty. So let's approach a mathematical formulation of the problem (not in a general form but only to the extent we need to) and get our hands on the uncertainty in numbers.

Assume that an experiment has only n possible smallest disjoint (those that cannot happen at the same time) outcomes $\omega_1, \omega_2, \ldots, \omega_n$. Moreover, each outcome ω_i has defined probability $\mathsf{P}_i \geq 0$, $i \in \{1, \ldots, n\}$, so that the sum of these probabilities equals one because, apart from $\omega_1, \omega_2, \ldots, \omega_n$, nothing else can happen:

$$\sum_{i=1}^{n} \mathsf{P}_i = 1.$$

For brevity and convenience, the outcomes and their probabilities are written in the following table:

ω_1	ω_2	$\ldots$	ω_n
P_1	P_2	$\ldots$	P_n

Let's consider a simple example. The experiment is tossing a fair coin. What does fair mean? A coin is fair if heads and tails have equal chances of coming up on each toss. In the notations, the experiment has two possible outcomes: $\omega_1 =$ heads and $\omega_2 =$ tails. The probabilities of these outcomes are the same and equal to $\frac{1}{2}$, respectively. Let's plot the data in the table:

Heads	Tails
$\frac{1}{2}$	$\frac{1}{2}$

.

What can we say about the result of the experiment? There's nothing specific actually. What outcome is preferable? We don't know because the probabilities of these outcomes are the same. From an informational standpoint, the situation is serious.

An unfair coin case is much easier. Take a look at the table:

Heads	Tails
$\frac{99}{100}$	$\frac{1}{100}$

Perhaps, everyone will confidently bet on heads. The uncertainty has decreased substantially. However, don't be overconfident. There's still a small chance of getting tails, which means there's the uncertainty.

What if the coin has heads on both sides? In this case, the experiment is described by the simple table:

Heads
1

There's no uncertainty whatsoever. We know for sure that we will get heads. From an informational standpoint, the worst case is when the outcomes are equiprobable. The best case is when the probability of one of the outcomes is one. Remember that :)

Here's an important point. Assume that we are tossing a fair die. There are 6 outcomes (the outcome ω_i means that we get the value of i), their probabilities are the same and equal $\frac{1}{6}$, and the experiment is described by the table:

1	2	3	4	5	6
$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$

Compare it to that when the fair coin was tossed:

Heads	Tails
$\frac{1}{2}$	$\frac{1}{2}$

.

Which one is more informative? Perhaps, the die case is not a whit better off. Apart from the fact that all the outcomes are equipossible, they have grown in number, and the uncertainty has increased.

We can give more examples, but you probably get the idea. Let's figure out how to bring these intuitive conceptions together in a particular number.

2.2 Shannon's Entropy and Expected Properties

Let's revisit the things we discussed and introduce the unified notations and definitions for future use. Well, let the experiment Ω have only n outcomes $\omega_1, \omega_2, \ldots, \omega_n$. As has been noted, outcomes are the smallest inseparable disjoint (those that cannot happen at the same time) results of the considered experiment. Let's match each outcome ω_i to its probability $\mathsf{P}_i \geq 0$ so that the sum of all probabilities equals 1:

$$\sum_{i=1}^{n} \mathsf{P}_i = \mathsf{P}_1 + \mathsf{P}_2 + \ldots + \mathsf{P}_n = 1.$$

This leads us to the following definition.

Definition 2.2.1 *The experiment Ω is an arbitrary set of outcomes $\omega_1, \omega_2, \ldots, \omega_n$, each corresponding to a value $\mathsf{P}_i \geq 0$, $i \in \{1, 2, \ldots, n\}$, called the probability of the outcome ω_i given that*

$$\sum_{i=1}^{n} \mathsf{P}_i = \mathsf{P}_1 + \mathsf{P}_2 + \ldots + \mathsf{P}_n = 1.$$

The experiment can be described by (and even considered as) the following table:

$$\begin{array}{c|c|c|c|c} \Omega & \omega_1 & \omega_2 & \dots & \omega_n \\ \hline \mathsf{P} & \mathsf{P}_1 & \mathsf{P}_2 & \dots & \mathsf{P}_n \end{array}.$$

The first row contains the possible outcomes of the experiment Ω. The second row is filled out with the probabilities of these outcomes. A convenient measure of uncertainty of the experiment Ω described by the table

$$\begin{array}{c|c|c|c|c} \Omega & \omega_1 & \omega_2 & \dots & \omega_n \\ \hline \mathsf{P} & \mathsf{P}_1 & \mathsf{P}_2 & \dots & \mathsf{P}_n \end{array},$$

is a value

$$\mathsf{H}(\Omega) = -\sum_{i=1}^{n} \mathsf{P}_i \log \mathsf{P}_i,$$

where, if $\mathsf{P}_i = 0$, the value of the expression $\mathsf{P}_i \log \mathsf{P}_i$ is considered zero because

$$\lim_{x \to 0+} x \log x = 0.$$

Remark 2.2.1 *The justification is purely mathematical. We inherently understand that the addition of the outcome ω_i that corresponds to the number P_i (the probability of this outcome) and equals zero has no impact on the experiment uncertainty (thus, no terms are added to the sum). There's no need to add this outcome to the table.*

The function log in the formulas is a logarithm to an arbitrary base greater than one. The base is irrelevant because a change in the logarithm base corresponds to a change in units of measurement of the value H (later on, we will use two as a base unit). For convenience, let's combine everything into one definition.

Definition 2.2.2 *Let the experiment Ω be described by the table:*

$$\begin{array}{c|c|c|c|c} \Omega & \omega_1 & \omega_2 & \dots & \omega_n \\ \hline \mathsf{P} & \mathsf{P}_1 & \mathsf{P}_2 & \dots & \mathsf{P}_n \end{array}.$$

Entropy (uncertainty measure) $\mathsf{H}(\Omega)$ of the experiment Ω is a value

$$\mathsf{H}(\Omega) = -\sum_{i=1}^{n} \mathsf{P}_i \log \mathsf{P}_i,$$

where log *is a logarithm to an arbitrary base greater than one, and the expressions of the form $0 \log 0$ are considered zero.*

Let's see what do we get in the discussed examples. Well, in the fair coin example, the experiment $\Omega = \{\text{Heads, Tails}\}$ is described by the table

Ω	Heads	Tails
P	$\frac{1}{2}$	$\frac{1}{2}$

.

The entropy of the experiment is calculated as follows:

$$\mathsf{H}(\Omega) = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} = -\log\frac{1}{2}.$$

The unfair coin experiment is described by the following table

Ω	Heads	Tails
P	$\frac{99}{100}$	$\frac{1}{100}$

,

thus, the entropy of this experiment is calculated as:

$$\mathsf{H}(\Omega) = -\frac{99}{100}\log\frac{99}{100} - \frac{1}{100}\log\frac{1}{100}.$$

As earlier, we will use a logarithm to the base 2. We will explain the choice of a logarithm a little bit later. After selecting a logarithm base, we can approximately calculate the expressions. The fair coin entropy will be equal

$$-\log_2\frac{1}{2} = 1,$$

and the unfair coin entropy will be

$$-\frac{99}{100}\log_2\frac{99}{100} - \frac{1}{100}\log_2\frac{1}{100} \approx 0.081.$$

As you can see, the results match the expectations. The greater the intuitive uncertainty, the greater the entropy, and vice versa.

Let's see whether the expectations will be met not only in a specific but also general case. First, entropy is a non-negative value.

Theorem 2.2.1

$$\mathsf{H}(\Omega) \geq 0.$$

Proof. Since $\mathsf{P}_i \in [0, 1]$, then $\mathsf{P}_i \log \mathsf{P}_i \leq 0$ because the logarithm base is greater than one (according to the definition of entropy). Thus,

$$\sum_{i=1}^{n} \mathsf{P}_i \log \mathsf{P}_i \leq 0,$$

as a sum of non-positive terms. Hence,

$$\mathsf{H}(\Omega) = -\sum_{i=1}^{n} \mathsf{P}_i \log \mathsf{P}_i \geq 0.$$

□

Therefore, a measure of uncertainty is non-negative. Perhaps, it should be zero when, and only when, there's no uncertainty, in other words, when an outcome occurs with the probability 1. This is also satisfied.

Theorem 2.2.2 *Entropy is zero if and only if a value of* P_i *equals one, that is:*

$$\mathsf{H}(\Omega) = 0 \Leftrightarrow \exists\ i \in \{1, 2, \ldots, n\} :\ \mathsf{P}_i = 1.$$

Proof. The proof towards the one side follows from the fact that, if $\mathsf{P}_i = 1$, the rest P_k given that $k \neq i$ are equal to zero. Thus, the terms corresponding to them are also zero (since $0 \cdot \log 0 = 0$ as agreed). The term under the number i is zero because $1 \cdot \log 1 = 0$.

The proof towards another side is also quite easy. The function $x \log x$ on the interval $[0, 1]$ is zero only at its ends (zero as agreed and one since $\log 1 = 0$). The function is negative at all points of the interval $(0, 1)$. Thus, for entropy to be zero, P_i should take values 0 or 1. However, since $\sum\limits_{i=1}^{n} \mathsf{P}_i = 1$, there's only one value i so that $\mathsf{P}_i = 1$. □

What are the properties that entropy should have? Based on the considered examples, entropy should be the highest when all outcomes are equiprobable (as in the experiments with a fair coin and fair die). The more equiprobable outcomes, the higher the entropy, which is inherently understood from the examples.

The introduced function also satisfies this requirement.

Theorem 2.2.3 *Entropy* $\mathsf{H}(\Omega)$ *is the highest when all the outcomes of the experiment are equipossible, that is, when the experiment is described by the following table:*

Ω	ω_1	ω_2	$\ldots$	ω_n
P	$\frac{1}{n}$	$\frac{1}{n}$	$\ldots$	$\frac{1}{n}$

In this case, entropy equals

$$\mathsf{H}\left(\Omega\right) = \log n.$$

Proof. For the experiment described by the table

Ω	ω_1	ω_2	$\ldots$	ω_n
P	$\frac{1}{n}$	$\frac{1}{n}$	$\ldots$	$\frac{1}{n}$

,

the equality is satisfied, since

$$\mathsf{H}(\Omega) = -\sum_{i=1}^{n} \frac{1}{n} \log \frac{1}{n} = -\log \frac{1}{n} = \log n.$$

All we've got left is to show that the value is maximal. To do so, we will use Jensen's inequality for downward-convex functions. According to it, for values $p_1, p_2, \ldots, p_n > 0$ so that $p_1 + p_2 + \ldots + p_n = 1$ and any $x_1, x_2, \ldots, x_n$ of the convex interval, the following is true

$$f(p_1 x_1 + p_2 x_2 + \ldots + p_n x_n) \leq p_1 f(x_1) + p_2 f(x_2) + \ldots + p_n f(x_n),$$

or

$$f\left(\sum_{i=1}^{n} p_i x_i\right) \leq \sum_{i=1}^{n} p_i f(x_i).$$

Let's consider the function $f(x) = x \log x$. Since the logarithm base is greater than one (as agreed), the function is convex downward. Assume that, in Jensen's inequality,

$$x_i = \mathsf{P}_i, \quad p_i = \frac{1}{n},$$

then,

$$f\left(\sum_{i=1}^{n} p_i x_i\right) = f\left(\frac{1}{n} \sum_{i=1}^{n} \mathsf{P}_i\right) = f\left(\frac{1}{n}\right) = -\frac{1}{n} \log n.$$

Moreover,

$$\sum_{i=1}^{n} p_i f(x_i) = \frac{1}{n} \sum_{i=1}^{n} \mathsf{P}_i \log \mathsf{P}_i.$$

Thus, according to Jensen's inequality,

$$-\frac{1}{n} \log n \leq \frac{1}{n} \sum_{i=1}^{n} \mathsf{P}_i \log \mathsf{P}_i \Leftrightarrow -\sum_{i=1}^{n} \mathsf{P}_i \log \mathsf{P}_i \leq \log n \Leftrightarrow \mathsf{H}(\Omega) \leq \log n,$$

which completes the proof. □

Actually, if we make additional assumptions, we can prove that Shannon's representation is unique up to the positive factor. To put it differently, if the function satisfies the three described properties (and something else that we will not discuss), the function equals

$$-\alpha \sum_{i=1}^{n} \mathsf{P}_i \log \mathsf{P}_i$$

for some $\alpha > 0$. It means that the logarithm base in the expression for the function can be any but greater than one.

3 Conditional Entropy

3.1 Heuristic Arguments

As you may have noticed, probabilities of experiment outcomes are usually unknown in practice. However, when we conduct an experiment and collect the results, we obtain the statistics that match each outcome with some numeric value. For example, you carry out a survey to find out whether your friends are going to play football. The survey data is given in the table:

Yes	No
9	5

The table shows that 14 friends in total were asked. 9 of them are willing to play, and 5 don't want to. Considering this, we can estimate the probability of each experiment outcome using frequency. In the given example, outcome probability estimations for Yes (to play football) and No (not to play football) will be equal to

$$\mathsf{P}(\text{Yes}) = \frac{9}{14}, \quad \mathsf{P}(\text{No}) = \frac{5}{14},$$

and the entropy of the experiment described by the table

Ω	Yes	No
P	$\frac{9}{14}$	$\frac{5}{14}$

that is filled out with the calculated frequencies equals

$$\mathsf{H}(\Omega) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} \approx 0.94.$$

Entropy is close to one, which is maximum in the experiment when there are only two outcomes, and the logarithm base is two ($\log_2 2 = 1$). Thus, we can say nothing specific about the survey.

In the experiment, we considered only answers of persons chosen at random. However, what will happen if we consider additional factors? Say, when making a frequency table, we won't simply ask friends about playing football. We will also take into account the current weather conditions. The table becomes more complicated. It can look as follows:

Weather \ Play football	Yes	No
Sunny	6	0
Cloudy	2	2
Rainy	1	3

What is the entropy of such a system? Did the entropy decrease with new data? Besides, how to calculate the entropy in this case? Perhaps, the developed apparatus is not enough for one experiment. Let's fix this.

3.2 Definition of Conditional Entropy

We are going to consider two experiments Ω and Θ. The first one consists of the outcomes ω_i, $i \in \{1, 2, \ldots, m\}$, and the second one consists of the outcomes θ_j, $j \in \{1, 2, \ldots, n\}$. When considering the pair of experiments (Ω, Θ), it's logical to match the pair of outcomes (ω_i, θ_j), $i \in \{1, 2, \ldots, m\}$, $j \in \{1, 2, \ldots, n\}$ with the probability $\mathsf{P}_{ij} \geq 0$ so that

$$\sum_{i=1}^{m} \sum_{j=1}^{n} \mathsf{P}_{ij} = 1,$$

because no other options are possible (only some outcome of the first experiment can occur and some of the second one). It's reasonable to introduce the following definition.

Definition 3.2.1 *The experiment* (Ω, Θ) *is an arbitrary set of outcome pairs* (ω_i, θ_j), $i \in \{1, 2, \ldots, m\}$, $j \in \{1, 2, \ldots, n\}$, *each corresponding to a value* $\mathsf{P}_{ij} \geq 0$ *called the probability of the outcome* (ω_i, θ_j) *so that*

$$\sum_{i=1}^{m} \sum_{j=1}^{n} \mathsf{P}_{ij} = 1.$$

The experiment (Ω, Θ) can be described by (and even considered as) the following table:

(Ω, Θ)	θ_1	θ_2	$\ldots$	θ_n
ω_1	P_{11}	P_{12}	$\ldots$	P_{1n}
ω_2	P_{21}	P_{22}	$\ldots$	P_{2n}
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
ω_m	P_{m1}	P_{m2}	$\ldots$	P_{mn}

Since we are dealing with an experiment (having $m \cdot n$ outcomes), its entropy is written as

$$\mathsf{H}\left((\Omega, \Theta)\right) = -\sum_{i=1}^{m} \sum_{j=1}^{n} \mathsf{P}_{ij} \log \mathsf{P}_{ij}.$$

The table shows how to reconstruct the experiments Ω and Θ separately. To find the probability of the outcome ω_i, we can sum up all the probabilities in the ith row of the table:

$$\mathsf{P}(\omega_i) = \sum_{j=1}^{n} \mathsf{P}_{ij}, \ i \in \{1, 2, \ldots, m\},$$

and, to find the probability of the outcome θ_j, we can sum up all the probabilities in the jth column of the table:

$$\mathsf{P}(\theta_j) = \sum_{i=1}^{m} \mathsf{P}_{ij},\ j \in \{1, 2, \ldots, n\}.$$

This leads us to the following theorem.

Theorem 3.2.1 *Let the experiment* (Ω, Θ) *be given by the table:*

(Ω, Θ)	θ_1	θ_2	$\ldots$	θ_n
ω_1	P_{11}	P_{12}	$\ldots$	P_{1n}
ω_2	P_{21}	P_{22}	$\ldots$	P_{2n}
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
ω_m	P_{m1}	P_{m2}	$\ldots$	P_{mn}

Then the experiments Ω *and* Θ *can be reconstructed using the following relations:*

$$\mathsf{P}(\omega_i) = \sum_{j=1}^{n} \mathsf{P}_{ij},\ i \in \{1, 2, \ldots, m\},$$

$$\mathsf{P}(\theta_j) = \sum_{i=1}^{m} \mathsf{P}_{ij},\ j \in \{1, 2, \ldots, n\}.$$

Let's jump right into the example. Let the experiment (Ω, Θ) be the toss of a coin. Two boys, Pete and Stu, are tossing it. The experiment (Ω, Θ) is described by the table:

(Ω, Θ)	Pete	Stu
Heads	0.4	0.275
Tails	0.1	0.225

The entropy of this experiment equals

$$\mathsf{H}((\Omega, \Theta)) = -0.4 \log_2 0.4 - 0.275 \log_2 0.275 - 0.1 \log_2 0.1 - 0.225 \log_2 0.225 \approx 1.86.$$

Since the highest entropy of the experiment can be equal to $\log_2 4 = 2$, we can conclude that the experiment (Ω, Θ) has high uncertainty.

The experiment Ω and the experiment Θ consist of two outcomes: $\Omega = \{\text{Heads, Tails}\}$, $\Theta = \{\text{Pete, Stu}\}$. Summing up the values in the rows leads us to the experiment Ω that can be written in the following table:

Ω	Heads	Tails
P	0.675	0.325

We can conclude that the coin is unfair because the chances of getting heads are higher than of getting tails. Summing up the values in the columns leads us to the experiment Θ described by the table:

Θ	Pete	Stu
P	0.5	0.5

.

The table allows us to conclude that Pete tosses a coin with the same (equal) probability as Stu, and vice versa.

We may find out that, in the experiment Θ, the event θ_j has occurred, then the probabilities of outcomes of the experiment Ω change in a predictable manner according to the conditional probability formula that we went over earlier while discussing a Bayes classifier. These probabilities are calculated as follows:

$$\mathsf{P}(\omega_i|\theta_j) = \frac{\mathsf{P}(\omega_i \cap \theta_j)}{\mathsf{P}(\theta_j)} = \frac{\mathsf{P}_{ij}}{\mathsf{P}(\theta_j)},\ i \in \{1, 2, ..., m\}, j \in \{1, 2, ..., n\}.$$

The case may be the opposite when we know that the outcome ω_i of the experiment Ω has occurred. Then the probabilities of outcomes of the experiment Θ are calculated as:

$$\mathsf{P}(\theta_j|\omega_i) = \frac{\mathsf{P}(\omega_i \cap \theta_j)}{\mathsf{P}(\omega_i)} = \frac{\mathsf{P}_{ij}}{\mathsf{P}(\omega_i)},\ i \in \{1, 2, ..., m\}, j \in \{1, 2, ..., n\}.$$

For the experiments with new probabilities, we can also calculate entropy. For example,

$$\mathsf{H}(\Omega|\theta_j) = -\sum_{i=1}^{m} \mathsf{P}(\omega_i|\theta_j) \log \mathsf{P}(\omega_i|\theta_j) = -\sum_{i=1}^{m} \frac{\mathsf{P}_{ij}}{\sum\limits_{i=1}^{m} \mathsf{P}_{ij}} \log \frac{\mathsf{P}_{ij}}{\sum\limits_{i=1}^{m} \mathsf{P}_{ij}}$$

and

$$\mathsf{H}(\Theta|\omega_i) = -\sum_{j=1}^{n} \mathsf{P}(\theta_j|\omega_i) \log \mathsf{P}(\theta_j|\omega_i) = -\sum_{j=1}^{n} \frac{\mathsf{P}_{ij}}{\sum\limits_{j=1}^{n} \mathsf{P}_{ij}} \log \frac{\mathsf{P}_{ij}}{\sum\limits_{j=1}^{n} \mathsf{P}_{ij}}.$$

These entropies are called conditional.

Definition 3.2.2 *Conditional entropy of the experiment Ω given that the experiment Θ turned out to be θ_j (that is, if the outcome θ_j has occurred), $j \in \{1, 2, \ldots, n\}$, is a value*

$$\mathsf{H}(\Omega|\theta_j) = -\sum_{i=1}^{m} \mathsf{P}(\omega_i|\theta_j) \log \mathsf{P}(\omega_i|\theta_j).$$

Remark 3.2.1 *Similarly (due to symmetry), we define conditional entropy of the experiment Θ given that the experiment Ω turned out to be ω_i.*

Let's return to the coin example and see how entropy changes. Take a look at the table that describes the experiment (Ω, Θ):

(Ω, Θ)	Pete	Stu
Heads	0.4	0.275
Tails	0.1	0.225

.

Assume that we know that Pete tossed the coin. Then,

$$\mathsf{P}(\text{Heads}|\text{Pete}) = \frac{\mathsf{P}(\text{Heads} \cap \text{Pete})}{\mathsf{P}(\text{Pete})} = \frac{0.4}{0.5} = 0.8,$$

and

$$\mathsf{P}(\text{Tails}|\text{Pete}) = \frac{\mathsf{P}(\text{Tails} \cap \text{Pete})}{\mathsf{P}(\text{Pete})} = \frac{0.1}{0.5} = 0.2.$$

Let's write the results in the table:

$(\Omega, \theta_1) = (\Omega|\text{Pete})$	(Heads|Pete)	(Tails|Pete)
P	0.8	0.2

.

The conditional entropy of the experiment Ω given that Pete tossed the coin is:

$$\mathsf{H}(\Omega|\theta_1) = \mathsf{H}(\Omega|\text{Pete}) = -0.8 \log_2 0.8 - 0.2 \log_2 0.2 \approx 0.72.$$

We perform similar calculations for the case when Stu tossed the coin and obtain the following table:

$(\Omega, \theta_2) = (\Omega|\text{Stu})$	(Heads|Stu)	(Tails|Stu)
P	0.55	0.45

.

In this case, the conditional entropy equals

$$\mathsf{H}(\Omega|\theta_2) = \mathsf{H}(\Omega|\text{Stu}) = -0.55 \log_2 0.55 - 0.45 \log_2 0.45 \approx 0.99.$$

The conditional entropy is less when Pete tosses the coin compared to the case when Stu flips the coin. The obtained values match the expectations. We can compare the tables:

$(\Omega, \theta_1) = (\Omega|\text{Pete})$	(Heads|Pete)	(Tails|Pete)
P	0.8	0.2

,

$(\Omega, \theta_2) = (\Omega|\text{Stu})$	(Heads|Stu)	(Tails|Stu)
P	0.55	0.45

.

When Pete is tossing the coin, the uncertainty is lower (the probabilities of outcomes differ significantly). Thus, the fact that Pete is tossing the coin gives us a little bit more information.

We are almost there. Since each outcome of the experiment Ω and Θ is accepted with some probability, conditional entropy takes its values with some probability. In other words, the value $\mathsf{H}(\Omega|\theta_1)$ is accepted with the probability $\mathsf{P}(\theta_1)$, the value $\mathsf{H}(\Omega|\theta_2)$ with the probability $\mathsf{P}(\theta_2)$, and so on. Since Θ is an experiment, the sum of probabilities of elementary outcomes equals one. Therefore, the conditional entropy $\mathsf{H}(\Omega|\theta_j)$ is a random variable with a distribution series

$\mathsf{H}(\Omega\|\theta_j)$	$\mathsf{H}(\Omega\|\theta_1)$	$\dots$	$\mathsf{H}(\Omega\|\theta_n)$
P	$\mathsf{P}(\theta_1)$	$\dots$	$\mathsf{P}(\theta_n)$

.

Definition 3.2.3 *This random variable $\mathsf{H}(\Omega|\theta_j)$ is called conditional entropy of the experiment Ω given that the experiment Θ has occurred.*

It turns out that the entire system is well characterized by so-called overall conditional entropy.

Definition 3.2.4 *Overall conditional entropy of the experiment Ω given that the experiment Θ has occurred is called a value*

$$\mathsf{H}(\Omega|\Theta) = \mathsf{E}\left(\mathsf{H}(\Omega|\theta_j)\right) = \sum_{j=1}^{n} \mathsf{P}(\theta_j)\mathsf{H}(\Omega|\theta_j).$$

Let's return to the coin example. We will write the calculated values of the conditional entropy of the event (Ω, θ) with corresponding probabilities as a distribution series:

$\mathsf{H}(\Omega\|\theta_j)$	$\mathsf{H}(\Omega\|\text{Pete})$	$\mathsf{H}(\Omega\|\text{Stu})$
P	$\mathsf{P}(\text{Pete})$	$\mathsf{P}(\text{Stu})$

.

Thus, we obtain the following table:

$\mathsf{H}(\Omega\|\theta_j)$	0.72	0.99
P	0.5	0.5

.

As a result, overall conditional entropy will be as follows:

$$\mathsf{H}(\Omega|\Theta) = 0.72 \cdot 0.5 + 0.99 \cdot 0.5 = 0.855.$$

Now let's see what happened to the entropy of the experiment Ω. In a general case, since the experiment Ω is described by the following table:

Ω	Heads	Tails
P	0.675	0.325

,

its entropy equals $\mathsf{H}(\Omega) = 0.910$. At the same time, the knowledge about what happened to the experiment Θ decreased the entropy since the overall conditional entropy became equal to $\mathsf{H}(\Omega|\Theta) = 0.855$.

3.3 An Example of Calculating the Conditional Entropy

Let's see what do we get in the discussed example. The friends were asked about playing football. The answers and weather conditions were recorded. The results are given in the table:

Weather \ Play football (Answer)	Yes	No
Sunny	6	0
Cloudy	2	2
Rainy	1	3

Later, we will find the overall conditional entropy of such a system, that is, $\mathsf{H}(\text{Answer}|\text{Weather})$. But before that, we are going to use the filled out table to create a probability table for the experiment $(Weather, Answer)$ using frequency as usual:

(Weather, Answer)	Yes	No
Sunny	$\frac{6}{14}$	0
Cloudy	$\frac{2}{14}$	$\frac{2}{14}$
Rainy	$\frac{1}{14}$	$\frac{3}{14}$

Each value in the table is obtained as the ratio of the number of occurrences of a particular outcome to the total number of outcomes. For example, the probability of the event that a friend will play football when it rains will be:

$$\mathsf{P}(\text{Answer} = \text{Yes} \cap \text{Weather} = \text{Rainy}) = \frac{1}{14}.$$

We simply divided the number of friends who wanted to play football when it rains by the total number of outcomes.

Let's move on to calculating the conditional entropy, but first, we would like to fix a state of the weather event. For example, let it rain. It's easy to find the probability that it will rain. It equals

$$\mathsf{P}(\text{Weather} = \text{Rainy}) = \frac{4}{14},$$

because there were 4 rainy-day cases out of 14. We can also further develop the theory:

$$\mathsf{P}(\text{Weather} = \text{Rainy}) = \mathsf{P}(\text{Weather} = \text{Rainy}|\text{Answer} = \text{Yes}) +$$

$$+\mathsf{P}(\text{Weather} = \text{Rainy}|\text{Answer} = \text{No}) = \frac{1}{14} + \frac{3}{14} = \frac{4}{14}.$$

Now we can calculate the conditional probability of the event that a friend will be willing to play football when it rains. The probability equals

$$\mathsf{P}(\text{Answer} = \text{Yes}|\text{Weather} = \text{Rainy}) =$$

$$= \frac{\mathsf{P}(\text{Answer} = \text{Yes} \cap \text{Weather} = \text{Rainy})}{\mathsf{P}(\text{Weather} = \text{Rainy})} = \frac{\frac{1}{14}}{\frac{4}{14}} = \frac{1}{4} = 0.25,$$

since there are two outcomes,

$$\mathsf{P}(\text{Answer} = \text{No}|\text{Weather} = \text{Rainy}) =$$

$$1 - \mathsf{P}(\text{Answer} = \text{Yes}|\text{Weather} = \text{Rainy}) = 0.75.$$

Based on the obtained conditional probabilities, we can find the conditional entropy of the experiment Answer given that the experiment Weather is being Rainy, that is:

$$\mathsf{H}(\text{Answer}|\text{Weather} = \text{Rainy}) = -0.25 \log_2 0.25 - 0.75 \log_2 0.75 \approx 0.81.$$

Similarly, we can find conditional entropies of the experiment Answer when the experiment Weather is being Sunny and Cloudy:

$$\mathsf{H}(\text{Answer}|\text{Weather} = \text{Sunny}) = -1 \log_2 1 - 0 \log_2 0 = 0,$$

$$\mathsf{H}(\text{Answer}|\text{Weather} = \text{Cloudy}) = -0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1.$$

We've found the conditional entropies. Therefore, we can write a distribution series for conditional entropy:

$\mathsf{H}(\text{Answer}\|\text{Weather})$	0	0.81	1
P	$\frac{6}{14}$	$\frac{4}{14}$	$\frac{4}{14}$

.

The expected value of the obtained random variable characterizes the overall conditional entropy of the experiment Answer relative to the experiment Weather. According to the definition, we obtain

$$\mathsf{E}\left(\mathsf{H}(\text{Answer}|\text{Weather})\right) = 0 \cdot \frac{6}{14} + 0.81 \cdot \frac{4}{14} + 1 \cdot \frac{4}{14} \approx 0.517,$$

which tells us about the average uncertainty of the system. We can make the same conclusion after looking at the original table:

Weather \ Play football (Answer)	Yes	No
Sunny	6	0
Cloudy	2	2
Rainy	1	3

.

If it's sunny, everyone is willing to play football, but if it rains, most of the friends will stay at home. When it's cloudy, the answers are fifty-fifty.

3.4 Entropy and Information Gain

Well, what did we get? We've learned that entropy shows the degree of uncertainty of the state of an experiment or system. The more we know about the system, the less uncertain its state is. That's why it's handy to measure a change in our knowledge about the system by tracking the change in entropy.

How to decrease the entropy of a system, or, what's the same, obtain the information gain? We can do this by employing additional knowledge about the system. We just need an event preceding the experiment to occur. In some sense, this event redefines the experiment.

We've already done the calculations but haven't written the final expression. Well, let's recall. First, we considered the experiment described by the table:

Yes	No
9	5

and we knew nothing about the system except the answers to the question. The entropy of the experiment was equal to

$$\mathsf{H}(\text{Answer}) \approx 0.94.$$

Next, after the second event (weather), we obtained new data, and the overall conditional entropy of the experiment Answer with respect to the experiment Weather became equal to

$$\mathsf{H}\left(\text{Answer}|\text{Weather}\right) \approx 0.52.$$

What information gain did we get? As you may guess, the information gain can be found as follows:

$$\mathsf{H}(\text{Answer}) - \mathsf{H}\left(\text{Answer}|\text{Weather}\right) = 0.42.$$

To sum up, we will introduce the definition.

Definition 3.4.1 *The information gain is a value*

$$\mathsf{IG}(\Omega|\Theta) = \mathsf{H}(\Omega) - \mathsf{H}(\Omega|\Theta).$$

Note that the information gain is always non-negative. It follows from the common-sense notion and something that is easy to prove

$$\mathsf{H}(\Omega) \geq \mathsf{H}(\Omega|\Theta)$$

for any experiment Θ.

3.5 An Example of Calculating the Information Gain

Now we can move to another example when there are more than two experiments (or events) and find the information gain with respect to different events. Imagine a cat show where cats are judged based on certain criteria, and those cats that are closest to a standard are awarded a title of Champion. The table contains information about the breed, coat color, and height from floor to shoulder. All of these criteria affect cuteness.

Breed	**Coat**	**Height**	**Cuteness**
British	White	Tall	No
British	Gray	Tall	Yes
British	White	Short	Yes
Maine coon	White	Tall	No
Maine coon	Brown	Tall	Yes
Maine coon	Brown	Short	No
Ragdoll	Gray	Tall	Yes
Maine coon	Gray	Short	Yes

What we want to find out is the most informative criterion. What was the most important for judges?

Imagine that you are attending the cat show and see the results listed in the Cuteness column. Based on this, you can tell cute cats from not-so-cute ones. To calculate the entropy H of the experiment (event) Ω Cuteness, we plot the data in the table:

Yes	No
5	3

.

Let's fill out the table of the experiment Ω. It has the following form:

Ω	Yes	No
P	$\frac{5}{8}$	$\frac{3}{8}$

.

The entropy of this experiment is easy to calculate. It equals

$$\mathsf{H}(\Omega) = -\frac{5}{8}\log_2\frac{5}{8} - \frac{3}{8}\log_2\frac{3}{8} \approx 0.954.$$

Since the highest entropy of the experiment can be equal to $\log_2 2 = 1$, we can conclude that there is only uncertainty. It turns out that the results of the final judging alone are not informative.

Assume there is one more breed (the experiment Θ). The table takes the following form:

Breed \ Cuteness	Yes	No
British	2	1
Maine coon	2	2
Ragdoll	1	0

To find out how informative the feature Breed is, we can calculate the overall conditional entropy and information gain. First, we calculate the conditional probabilities. They are equal to:

$$\mathsf{P}(\text{Cuteness} = \text{Yes}|\text{Breed} = \text{British}) = \frac{2}{3},$$

$$\mathsf{P}(\text{Cuteness} = \text{No}|\text{Breed} = \text{British}) = \frac{1}{3},$$

$$\mathsf{P}(\text{Cuteness} = \text{Yes}|\text{Breed} = \text{Maine coon}) = \frac{1}{2},$$

$$\mathsf{P}(\text{Cuteness} = \text{No}|\text{Breed} = \text{Maine coon}) = \frac{1}{2},$$

$$\mathsf{P}(\text{Cuteness} = \text{Yes}|\text{Breed} = \text{Ragdoll}) = 1,$$

$$\mathsf{P}(\text{Cuteness} = \text{No}|\text{Breed} = \text{Ragdoll}) = 0.$$

Now we can calculate conditional entropies.

$$\mathsf{H}(\text{Cuteness}|\text{Breed} = \text{British}) = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} \approx 0.918,$$

$$\mathsf{H}(\text{Cuteness}|\text{Breed} = \text{Maine coon}) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1,$$

$$\mathsf{H}(\text{Cuteness}|\text{Breed} = \text{Ragdoll}) = -\frac{1}{1}\log_2\frac{1}{1} = 0.$$

Thus, the overall conditional entropy equals:

$$\mathsf{H}(\text{Cuteness}|\text{Breed}) = \frac{3}{8}\cdot 0.918 + \frac{4}{8}\cdot 1 + \frac{1}{8}\cdot 0 = 0.844.$$

Hence, when the breed is known, the information gain equals

$$\mathsf{IG}(\text{Cuteness}|\text{Breed}) = 0.954 - 0.844 = 0.110.$$

To decide whether it is too small or too big, we need a comparison. We invite you to calculate the information gain for yourselves in a case when there is data on coat or height from floor to shoulder. The results should match the following:

$$\mathsf{IG}(\text{Cuteness}|\text{Coat}) \approx 0.355,$$

$$\mathsf{IG}(\text{Cuteness}|\text{Height}) \approx 0.003.$$

These values of the information gain show that the criterion Coat is the most informative. In other words, the results of the final judging and awarded titles are mostly linked to coat colors. The given example is somehow artificial, but it makes the idea clear.

Now you also know why we introduced the concepts of entropy and information gain. Well, let's move on to decision trees.

4 Decision Trees (DTs)

We know how to measure information gain using entropy. Thus, we are ready to construct to decision trees, but first, the definition should be clarified.

4.1 About Decision Trees

If you are familiar with graph theory, the definition of a decision tree can be given as follows.

Definition 4.1.1 *A decision tree is a tree (or equivalently a connected acyclic graph) having the following features:*

- *Nodes of the tree that are not leaves are attributes.*
- *Leaves contain responses that are classification results.*
- *Edges have a rule, that is, a value of the node (attribute) that originates an edge.*

If you know nothing about graph theory, we can illustrate the introduced concept using the decision tree that aims to split the job offers into two categories of to be accepted or to be declined. The figure shows that the upper tree node corresponding to the predictor (or attribute) Salary originates three edges. If the node value is less than 50 thousand, we get into the leaf of declining the offer. If the node value is greater than or equal to 100000, we get into the leaf with another response of accepting the offer. If the node value is between 50 thousand (inclusive) and 100 thousand (exclusive), the edge leads us to the next node that is responsible for commuting time. Depending on the value of this node, additional branches of the tree grow. You can see it for yourself. Note that every path through the constructed decision tree ends with a leaf, and the original (root) vertex leads to every leaf. It is a connection rule mentioned in the definition. An acyclic property, or absence of cycles, is due to the fact that every node (except the root) and every leaf is connected by exactly one incoming edge.

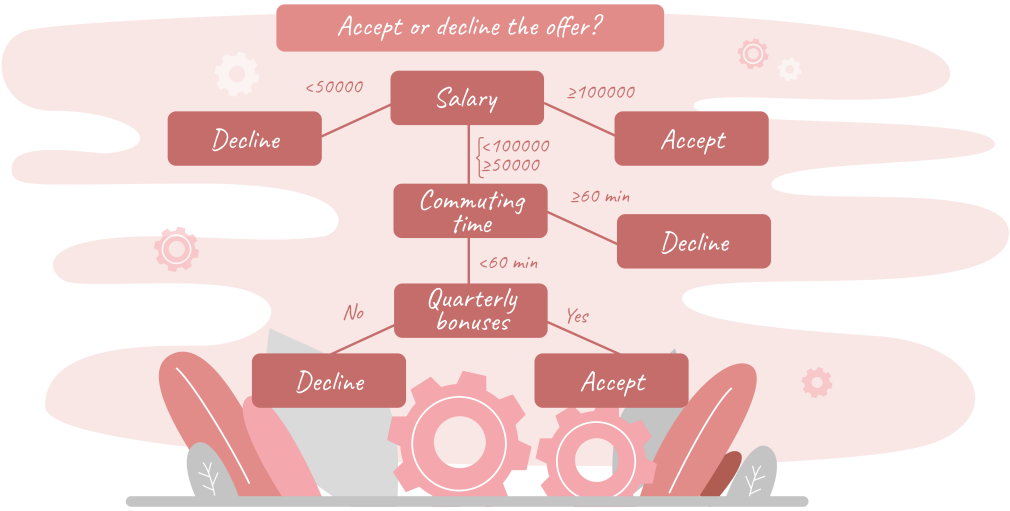

Figure 2: An example of a decision tree.

Definition 4.1.2 *The depth of a leaf (or a node) is the number of edges that connect it to the root. The depth of a tree is the largest depth of its leaves.*

There are 5 leaves in the example. The depth of each equals (from left to right): 1, 3, 3, 2, and 1, respectively. Therefore, the depth of the tree equals 3.

Now that you've figured out what a decision tree is and know how to find informative features, we can proceed to explain the algorithms for constructing decision trees.

4.2 An Algorithm for Constructing a Tree and the Cat Show Example

Let's develop an algorithm for constructing a decision tree using the training data $x_1, x_2, ..., x_n$ consisting of n elements with p predictors $X_1, X_2, ..., X_p$ each and the response Y.

1. Assume that the input is a set of objects X. Among p predictors, we need to choose one that maximizes the information gain. We solve the following problem:
$$\arg\max_{Q \in \{X_1, X_2, ..., X_p\}} \mathsf{IG}(Y|Q)$$

2. Let the predictor X_i be chosen. It takes exactly t unique values on the dataset X. We need to split the dataset X into two subsets $S_1, \ldots, S_t$ based on unique values of the predictor X_i.

3. For every set S_i, $i \in \{1, 2, ..., t\}$, if the response entropy is not zero, steps 1 and 2 are repeated.

In the terminology of the decision trees, the predictor X_i chosen in the first step of the algorithm is a node of the tree, and splitting the set of objects into t subsets based on a unique value of the predictor is creating t edges from the node X_i. Zero entropy for some set S_i means that the edge resulting in the set S_i points to the leaf that should be assigned the value of the responses of objects of S_i (due to zero entropy, all objects have the same response).

Let's apply the algorithm to the cat show example. The input data table in the given notations is as follows:

No.	**Breed** (X_1)	**Coat** (X_2)	**Height** (X_3)	**Cuteness** (Y)
x_1	British	White	Tall	No
x_2	British	Gray	Tall	Yes
x_3	British	White	Short	Yes
x_4	Maine coon	White	Tall	No
x_5	Maine coon	Brown	Tall	Yes
x_6	Maine coon	Brown	Short	No
x_7	Ragdoll	Gray	Tall	Yes
x_8	Maine coon	Gray	Short	Yes

To put it differently, each row of the table is a training object with three predictors X_1, X_2, X_3 (Breed, Coat, Height), respectively, and the response Y (Cuteness).

The first step has been already completed. According to the considered example, the criterion Coat (the predictor X_2) is the most informative. Since X_2 takes three unique values (White, Gray, and Brown), the second step is to split the training dataset into 3 groups (based on X_2). Let the first group S_{Brown} be a group of Brown coat cats. It includes x_5, x_6. The second group S_{White} of White coat cats consists of x_1, x_3, x_4. The last one S_{Gray}, a group of Gray cats, includes all the remaining cats, that is, x_2, x_7, x_8. Let's visualize it.

Look at fig. 2 (It somehow differs from the previously described trees in visualization and information conveyed). The upper white box (the root) contains the information about the input dataset $X = \{x_1, x_2, ..., x_n\}$. The second row of the box shows the entropy calculated based on the response Y in the input dataset. It approximately equals 0.954. The third row contains the number of elements of the studied dataset, while the fourth contains the relation between cute and less cute cats in the input dataset (3 less cute cats and 5 cuties).

Since the criterion Coat taking 3 different values is the most informative in the first step, the root that has three outgoing edges, and the second tree level emerges. It consists of three groups $S_{\text{Коричневый}}$, S_{White}, S_{Gray} corresponding to three different nodes. Since, among brown cats, 1 cat is cute and 1 is not so cute, the entropy for the response Cuteness on the dataset S_{Brown} equals 1. Similarly, since, among white cats, 2 cats are not so cute and 1 is cute, the entropy for the

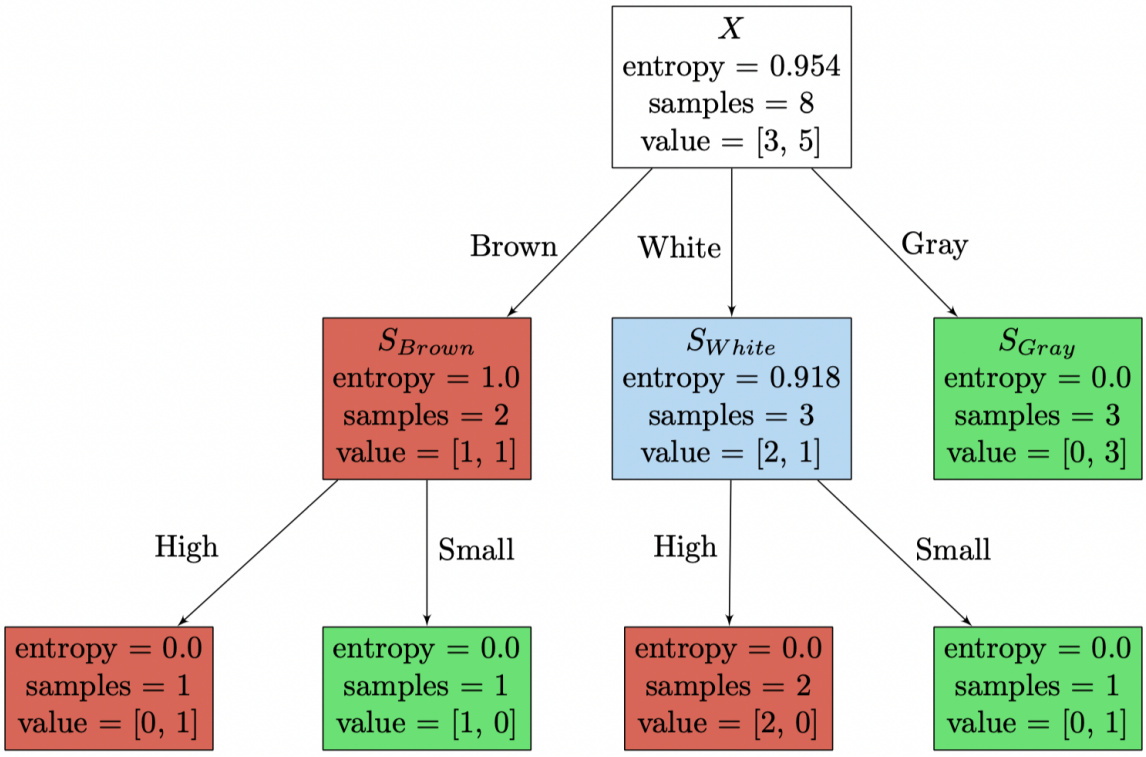

Figure 3: An example of a decision tree.

response Cuteness on the dataset S_{White} equals

$$-\frac{2}{3}\log_2\frac{2}{3}-\frac{1}{3}\log_2\frac{1}{3}\approx 0.918.$$

Among gray cats, all three are cute. Therefore, the entropy for the response Cuteness in the group S_{Gray} equals 0, and subsequent splitting into subsets of the group S_{Gray} is not needed unlike other two groups. The node corresponding to this box becomes a leaf with the response Yes.

There remains the uncertainty in the remaining subsets. Thus, according to the algorithm, we need to define new splitting criteria for each of them. If we carefully consider the subsets S_{Brown} and S_{White} (or simply count), we will see that the most informative criterion in each group is the height from floor to shoulder. This leads to the splitting at the third level of the tree. Each created subset (there are four of them) has zero entropy. Thus, the algorithm is completed, and all the nodes of the last level become the leaves with responses (from left to right): Yes, No, No, Yes, respectively.

The classification of new objects using the constructed tree is simple. First, a new object is classified in the upper level (by the coat color in our example). If it's gray, the decision tree says that judges will consider the cat cute. If a cat is brown or white, its height matters (the next level). According to the tree, a tall brown or short white cat will be considered cute, while a short brown or tall white cat will be less cute.

Remark 4.2.1 *The last point of the algorithm described in the beginning is one of the possible ways to stop splitting the training dataset into groups (thus, it's one of the ways to stop the construction of the DT). When entropy of each new group is zero, it means that all elements of each group have the same response, and the group is easy to characterize based on the response. On big data, splitting until the entropy is zero is a compelling difficulty that often leads to overfitting. That's why the following stopping criteria are used:*

- *A limitation on tree depth. Nodes having maximum established depth become leaves.*
- *A limitation on the minimum number of group elements. When the considered set is split into subsets based on unique values of the predictor, the created groups of training data contain fewer elements than established by a researcher, and splitting is stopped. In decision-tree terms, it means that the node becomes a leaf.*
- *Meeting a set criterion in a group, for example, an uncertainty value. In decision-tree terms, it also means that the node becomes a leaf.*

In each of these cases, there remains the question. What value to assign to a leaf? Usually, it's assigned based on the response of the majority of training data elements. We will discuss this matter a little bit later.

4.3 Binary Decision Tree

The considered algorithm for constructing a decision tree is useful when we are dealing with predictors, which possible values are contained in the training data. If a test object has a predictor value that is not contained in training data, the described classification algorithm is, obviously, not working. For example, we want to know whether judges conclude that the short black cat is cute. Fig. 2 makes it clear that the classification fails at the very first level because the choice of a branch is not obvious. The value of the feature Coat doesn't match any of the available options.

This problem is quite serious. Often, we don't know all the possible values that the attribute of the considered objects can take. Even more often, these values are infinite (especially, when the attribute values are numeric). Here are the examples of attributes that take an infinite number of values: a person's height, the weight of different goods on scales, current time or date (just to name a few). If an attribute is not numeric, it can rarely take many values, but still a lot of them. City names are such an example.

Moreover, the more attribute values we have, the longer it takes for a decision tree to grow. Additionally, it becomes very branchy, which impedes interpretation

of the model and leads to overfitting. Due to these reasons, it's necessary to develop a new approach to decision tree construction that can:

- Classify objects that have unknown predictor values (those that weren't included in the training data).
- Merge (group) many feature values.

Merging decreases the number of branches, makes interpretation easier, and increases model performance.

All of this leads us to so-called binary decision trees.

Definition 4.3.1 *A binary decision tree is a decision tree in which each node has two outgoing edges.*

Well, binary decision trees are still decision trees. Their distinctive feature is that splitting always results in only two subsets. Entropy is calculated for each subset after each split. The results allow us to define the most informative criterion for further splitting (if needed). Splitting criteria are as follows. If the attribute values meet a set criterion, the object is assigned to the first set, otherwise, to the second. Thus, objects with unknown feature values will be assigned to one of the possible features, and, as a result, they will be classified. But how to perform splitting? What are these set criteria?

4.3.1 Feature Types and Grouping

Let's begin with some important definitions. Note that we are working with a training dataset $x_1, x_2, ..., x_n$ of size n each object of which has p attributes

$$x_i = (x_{i1}, x_{i2}, ..., x_{ip}).$$

Definition 4.3.2 *An object feature is categorical if it takes a finite number of values.*

Categorical features can be numeric and non-numeric. In the cat show example, all features are categorical. The feature Coat takes some value from the finite set $\{\text{White}, \text{Gray}, \text{Brown}\}$, and the feature Height takes a value from the finite set $\{\text{Tall}, \text{Short}\}$. In addition to categorical features, it's handy to distinguish non-categorical features.

Definition 4.3.3 *If an object feature is not categorical, we will call it non-categorical.*

Non-categorical features can be height, the weight of different goods on scales, current time or date, and many more. Non-categorical features are usually numeric.

Now let's decide how to group features. In case of a non-categorical numeric feature X_i, a set of its values is split into two sets:

$$\text{the first set: } X_i \leq C, \text{ the second set: } X_i > C,$$

and a value C (or a so-called cut-off value) is chosen separately.

Remark 4.3.1 *Let's discuss in detail how to choose the value C. How to make this choice? Many options are possible, but we will consider only a few.*

1. *Here's the first option. To begin with, a set of values (an interval usually) $[a, b]$ is defined for a numeric feature X_i based on theoretical considerations or values of the feature on training data. In the latter case, the smallest value of the considered feature on the training data is taken as a, and the biggest is taken as b. The next step is to create an increment $h > 0$ to make incremental changes to C. With the increment h, we can consider the following values as values of C:*

   $$\{a, a + h, ..., a + (k - 1)h, a + kh\}, \ k \in \mathbb{Z},$$

   where $a + (k - 1)h < b$, but $a + kh \geq b$. For each of the chosen values, it's necessary to calculate the information gain corresponding to the two-class splitting: $X_i \leq C$ and $X_i > C$.

2. *The second option is like the first. The values of C are chosen so that each subsequent value of C adds exactly one new value of the predictor X_i to the set $X_i \leq C$. Let's describe it formally. Let $\{x_{1i}, x_{2i}, ..., x_{ti}\}$, $1 \leq t \leq n$ be a set of all possible values of the predictor X_i on training data of size n. We also assume that the values are sorted in ascending order, that is,*

   $$x_{1i} < x_{2i} < ... < x_{ti}.$$

   Thus, we can take the following values as the values of C:

   $$\{C_1, C_2, ..., C_t\},$$

   where $C_t \geq x_{ti}$, $x_{ji} \leq C_j < x_{(j+1)i}$, $j \in \{1, 2, ..., (t - 1)\}$.

Finally, splitting with the highest information gain is chosen.

If the feature X_i is categorical and takes the values from the set $M = \{x_{1i}, x_{2i}, ..., x_{ti}\}$, $t \in \{1, 2, ..., n\}$, its set of values can be separated into two halves, for example, like this:

$$\text{the first set: } X_i \in M_1, \text{ the second set: } X_i \in M_2,$$

where $M_1 \cap M_2 = \varnothing$, $M_1 \cup M_2 = M$, $M_j \neq \varnothing$, $j \in \{1, 2\}$. To put it differently, the entire set M of unique values of the feature X_i is divided into 2 nonempty disjoint parts.

Remark 4.3.2 *In different math packages, the set M_1 usually consists of exactly one value of the feature X_i, while M_2 includes all the remaining values. Hence, if $M = \{x_{1i}, x_{2i}, ..., x_{ti}\}$, $t \in \{1, 2, ..., n\}$, then*

$$M_1 = \{C\}, \; M_2 = M \setminus M_1,$$

where C takes, in turn, values of each element of the set M after each iteration. Finally, splitting with the highest information gain is chosen.

4.3.2 An Algorithm for Constructing a Binary Tree and the Cat Show Example

Let's develop an algorithm for constructing a binary decision tree for the training dataset $X = \{x_1, x_2, ..., x_n\}$ of size n with p predictors $X_1, X_2, \ldots, X_p$ and the response Y.

1. Assume that the input is a set of objects X. Among p predictors, choose one that maximizes the information gain for the fixed splitting of values into subsets. The following problem is solved:

$$\underset{Q \in \{X_1, X_2, ..., X_p\}}{\arg\max} \; \mathsf{IG}(Y|Q(C)),$$

   where C is defined based on a feature type according to the rules mentioned earlier.

2. It is assumed that the predictor X_i is chosen and C is found. The goal is to perform feature-based splitting into two subsets S_C and $\overline{S}_C$. The first subset includes those objects which values of the ith attribute meet the splitting criterion. The second subset includes all the remaining objects.

3. Steps 1 and 2 are repeated for the sets S_C and $\overline{S}_C$ according to the restriction on the number of levels (according to the note 4.2.1), or until zero entropy is reached in each group if no restriction is imposed.

Let's apply the algorithm to the cat show example. According to the previous calculations, the predictor Coat with the value Gray produced zero entropy (the highest information gain). This criterion will be the first when two subsets S_{Gray} and $\overline{S}_{\text{Gray}}$ are formed. The first set includes all the gray cats, the second all that remain. The entropy in the second set will be

$$-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} \approx 0.971.$$

There's no uncertainty in the first subset, but it remains in the second subset. According to the third step of the algorithm, the first two steps should be repeated for this subset. The subsequent splitting of the subset and criteria are shown in fig. 3. The color criterion is still the most informative at the second level, but the cats are separated into the groups of White and Not-White. At the last level, the cats are split into groups based on their size. All the subsets have zero entropy at the last level of the tree, which means that the construction is completed.

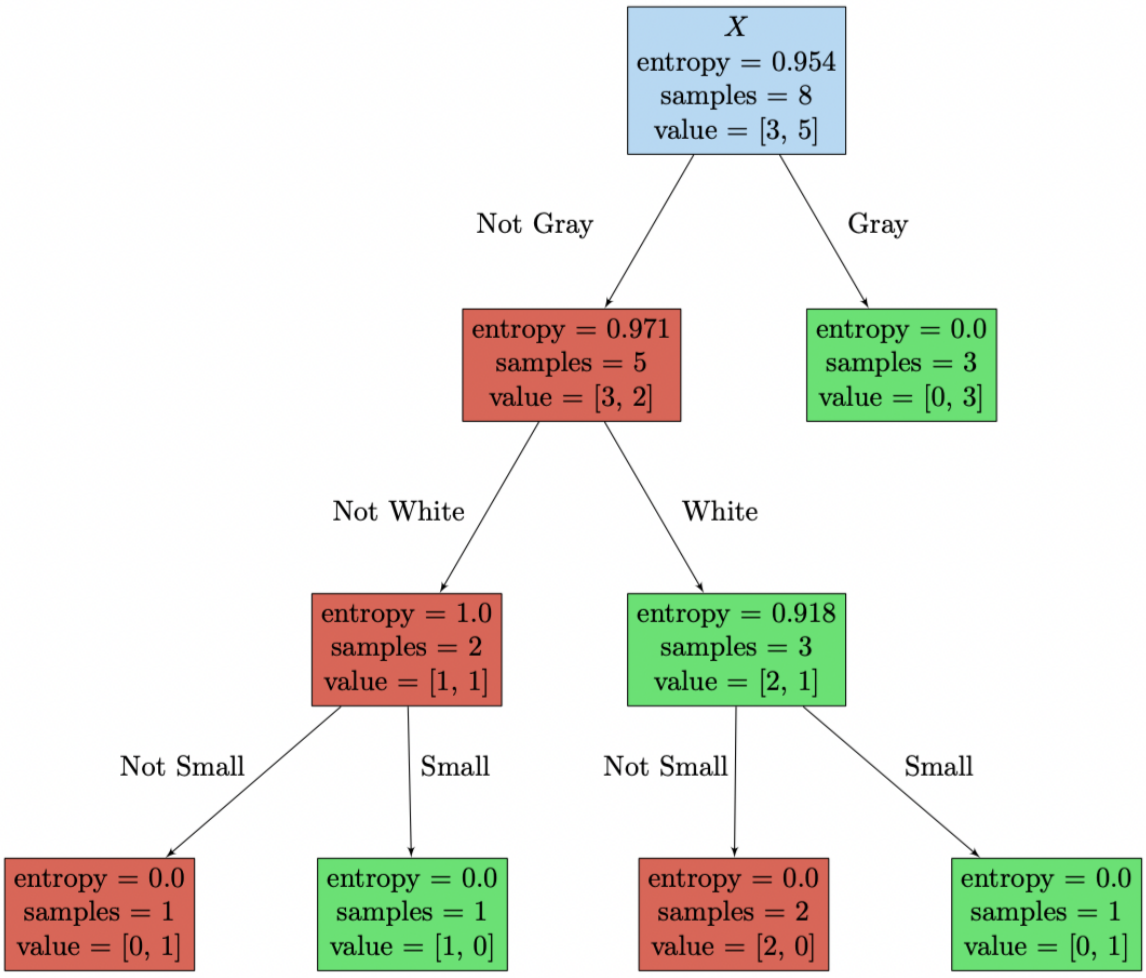

Figure 4: An example of a binary decision tree.

A binary tree can classify new objects even when the value of a predictor is unique (or new). The decision tree constructed earlier failed to classify a short

black cat. The binary tree that we just constructed can handle this. Since the cat is black, not gray, we are moving along the left branch. Similarly, since the cat is black, not white, we are also following the left branch. Finally, because the cat is short, we turn to the right branch and conclude that judges will think that the cat is not cute enough.

4.3.3 A Synthetic Example

Next, we are going to test the algorithm on synthetic data with two predictors X_1 and X_2. The data is intuitively split by a straight line into two subsets. The predictors corresponding to the data take only numeric values.

The goal is the same: to perform binary classification. The predictors of the data with the response 0 (green) are independent and have a distribution $\mathsf{N}_{0,1}$. There are 100 generated points. The predictors of the data with the response 1 (yellow) are also independent and have a distribution $\mathsf{N}_{2,1}$. A total of 200 values have been generated. They are shown in fig. 4.

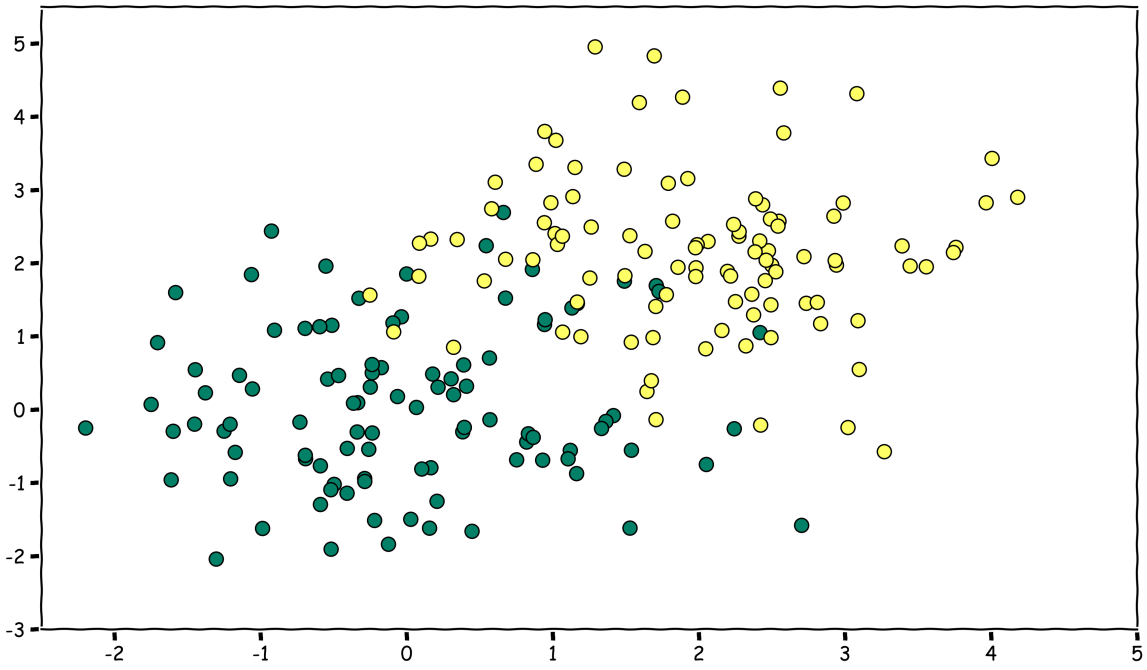

Figure 5: Objects are intuitively separable.

Let's construct a binary decision tree limited by two levels (the tree depth is 2). The initial entropy of the dataset of interest is the highest and equals 1 because there's an equal amount of zeros and ones in the response. The computations in a modeling system show that the input dataset splitting is optimal if based on the feature $X_2 \leq 0.77$. Thus, the steps are similar to those in the cat show example. However, the number of calculations increases.

Fig. 5 shows what has been discussed. The input dataset of 200 elements is split into two based on the feature $X_2 \leq 0.77$. The green box includes the data

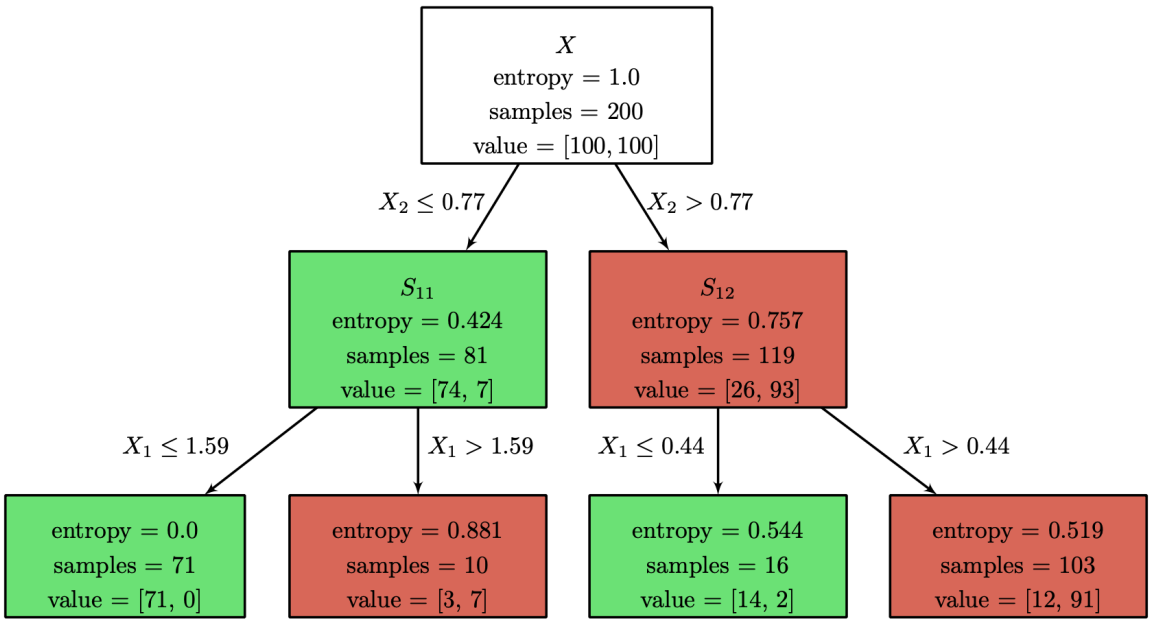

Figure 6: Decision tree.

that meets the splitting criterion, or, in other words, the data with the second feature of less than 0.77. The red box includes the remaining data. Since the entropy in each group is not zero, the algorithm continues to work. The next splitting in both groups is based on the feature X_1. The left group is split based on $X_1 \leq 1.59$, and the right based on $X_1 \leq 0.04$.

In the example, the depth is limited by two levels. However, even such a limited tree has a zero-entropy group which is, clearly, good. According to the algorithm, all the nodes at the second level of the tree become leaves with responses Green, Yellow, Green, Yellow, respectively. Why, for example, the second leaf corresponds to the response Yellow? Among the training data, this box includes 3 green and 7 yellow objects. The response is established based on the response of the majority of present objects. Since the objects with the response Yellow are in the majority, Yellow becomes the response of the entire leaf.

The DT classification (as well as splitting criteria) in the example can be visualized in the plane. First, it's sufficient to separate the plane by the straight line $X_2 = 0.77$. Fig. 6 shows the plane separated into two parts. The lower part corresponds to the left box of the tree ($X_2 \leq 0.77$), and the upper part corresponds to the right box ($X_2 > 0.77$).

The next criterion of splitting for the lower part of the plane (the left box of the tree) is the criterion $X_1 \leq 1.59$, and for the upper (the right box) the criterion $X_1 \leq 0.04$. The visualized split is shown in fig. **??**.

At this stage, the tree is considered trained. It makes errors on the training data, for example, when yellow points appear in the green region, or, conversely,

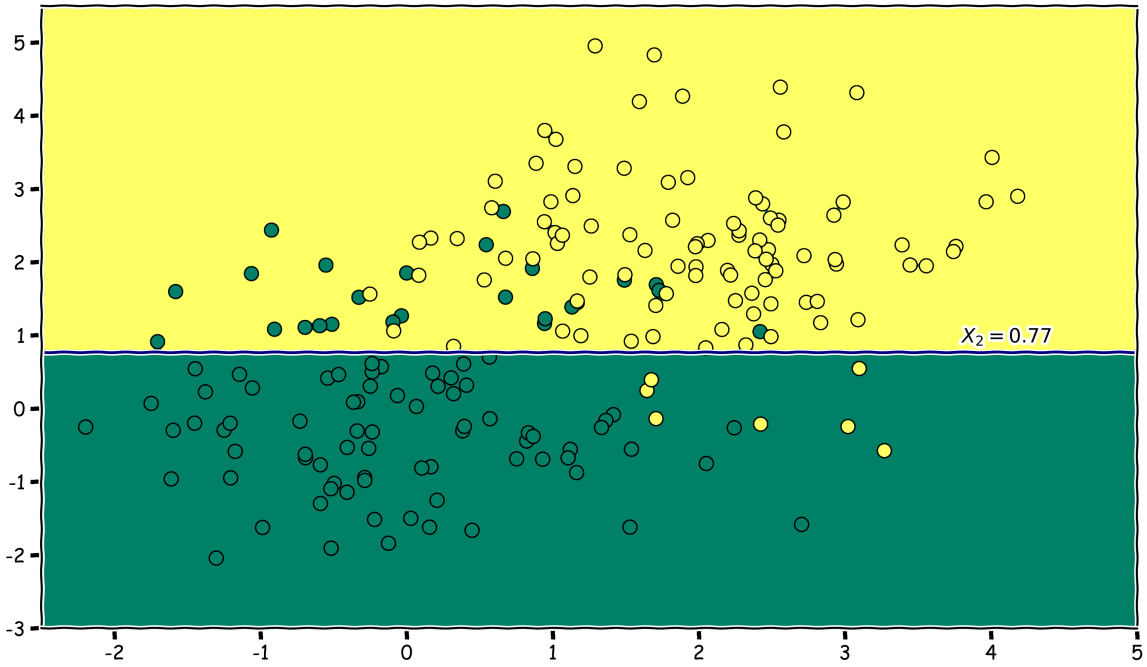

Figure 7: Decision tree classification (the first split).

when green points appear in the yellow region. There are 17 errors of this type. Let's see how the tree will work on a new sample from a normal distribution with the same parameters. The error rate is the same. There are 17 errors at the training stage and 21 during the testing. We inherently understand that the tree produces an error in 11% of the cases, which is not bad. Therefore, a two-level tree is enough. What happens when the number of levels increases up to, for example, six? Of course, the tree will grow bigger along with the number of groups with zero entropy. The figure shows the classification regions corresponding to the criteria of splitting into groups when constructing a tree based on the same input data and when constructing a two-level tree.

As you can tell by the tree visualization, there's overfitting. The number of regions and their bizarre forms seem totally unnatural. And that's how it is. After testing the tree on a test set, we observe empty regions that are not used at all (these regions do not contain the test data) and the regions that encompass many points of another class (classification errors).

5 Gini Impurity

5.1 Definition and Properties

Other information criteria can be used to construct decision trees. Among them, Gini impurity (that measures misclassification) deserves special attention.

What is misclassification? For ease of understanding, let's return to the

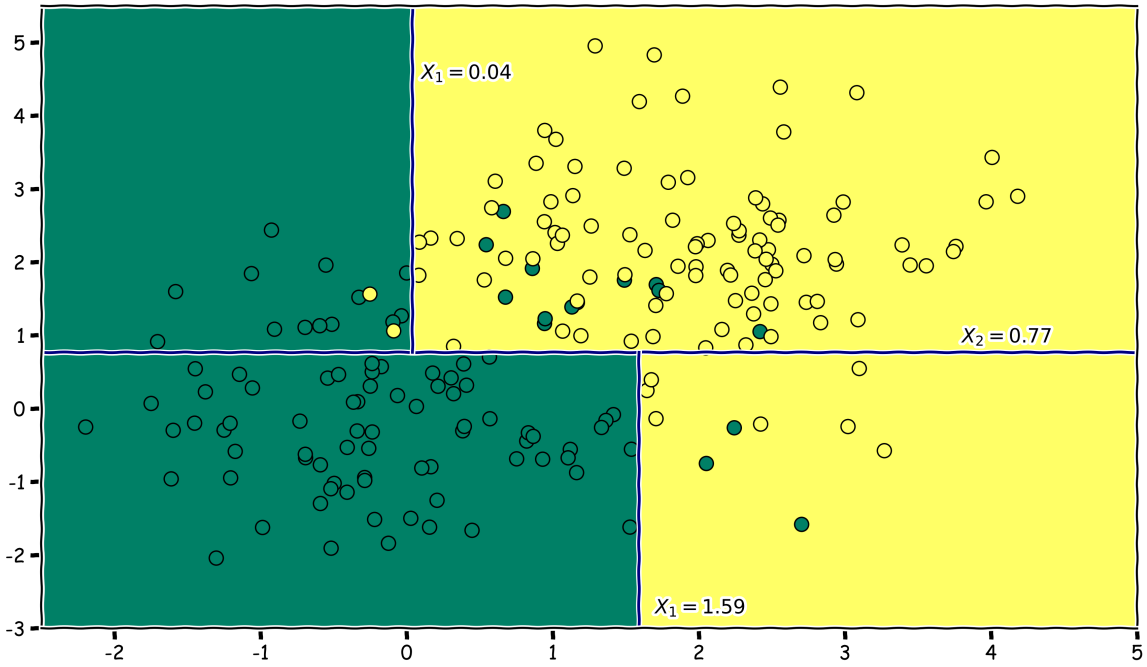

Figure 8: Decision tree classification (the final split).

example of the friend survey. Among 14 responders, 9 agreed to play football and 5 refused to go. Thus, the experiment is given by the following table:

Ω	Yes	No
P	$\frac{9}{14}$	$\frac{5}{14}$

.

Let's assume that we take a random friend. What is the probability of the event that this friend will be classified incorrectly? The event splits into two classes. The friend is either will be willing to play football but will be assigned to a group of those who refused to go, or the opposite case. The probability of such an event can be calculated as follows:

$$\mathsf{P}(\text{incorrectly classified friend}) = \frac{9}{14} \cdot \frac{5}{14} + \frac{5}{14} \cdot \frac{9}{14} = \frac{90}{196}.$$

This value is Gini impurity, or a misclassification measure. Let's give a definition.

Definition 5.1.1 *Let the experiment Ω be described by the table:*

Ω	ω_1	ω_2	$\dots$	ω_n
P	P_1	P_2	$\dots$	P_n

.

Gini impurity $\mathsf{G}(\Omega)$ of the experiment Ω is a value

$$\mathsf{G}(\Omega) = \sum_{i=1}^{n} \mathsf{P}_i(1 - \mathsf{P}_i).$$

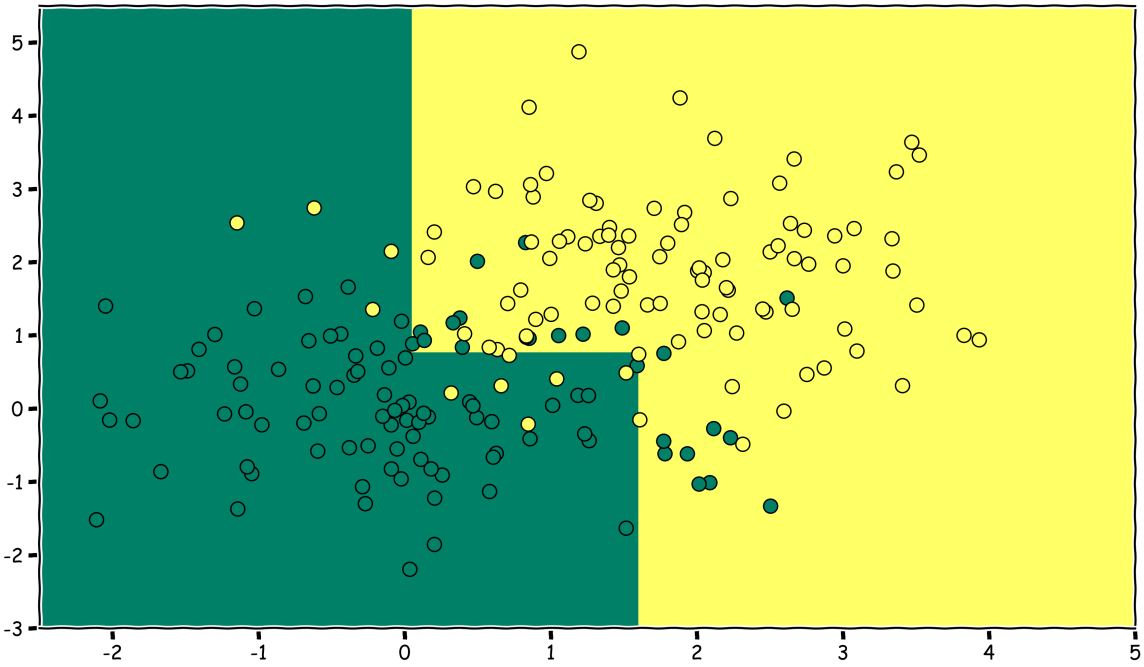

Figure 9: The classification of new objects using the trained model.

Remark 5.1.1 *Note that the Gini impurity expression can be rewritten as follows:*

$$\mathsf{G}(\Omega) = 1 - \sum_{i=1}^{n} \mathsf{P}_i^2.$$

It is obtained as follows:

$$\mathsf{G}(\Omega) = \sum_{i=1}^{n} \mathsf{P}_i(1 - \mathsf{P}_i) = \sum_{i=1}^{n} \left(\mathsf{P}_i - \mathsf{P}_i^2\right) = \sum_{i=1}^{n} \mathsf{P}_i - \sum_{i=1}^{n} \mathsf{P}_i^2 = 1 - \sum_{i=1}^{n} \mathsf{P}_i^2.$$

How are Gini impurity and entropy related? It turns out that the properties of Gini impurity are similar to those of entropy.

Theorem 5.1.1 *Assume that the experiment Ω is considered. Then,*

$$\mathsf{G}(\Omega) = 0 \Leftrightarrow \mathsf{H}(\Omega) = 0.$$

Since the entropy is zero if and only if exactly one outcome of the experiment has the probability 1 (which means there's no uncertainty), the same applies to Gini impurity.

Proof. If $\mathsf{H}(\Omega) = 0$, there's such an outcome of the experiment ω_i so that its probability $\mathsf{P}_i = 1$. Then, the probabilities of the remaining outcomes (if there any) are equal to 0, thus,

$$\mathsf{G}(\Omega) = 1 - \mathsf{P}_i^2 = 1 - 1 = 0.$$

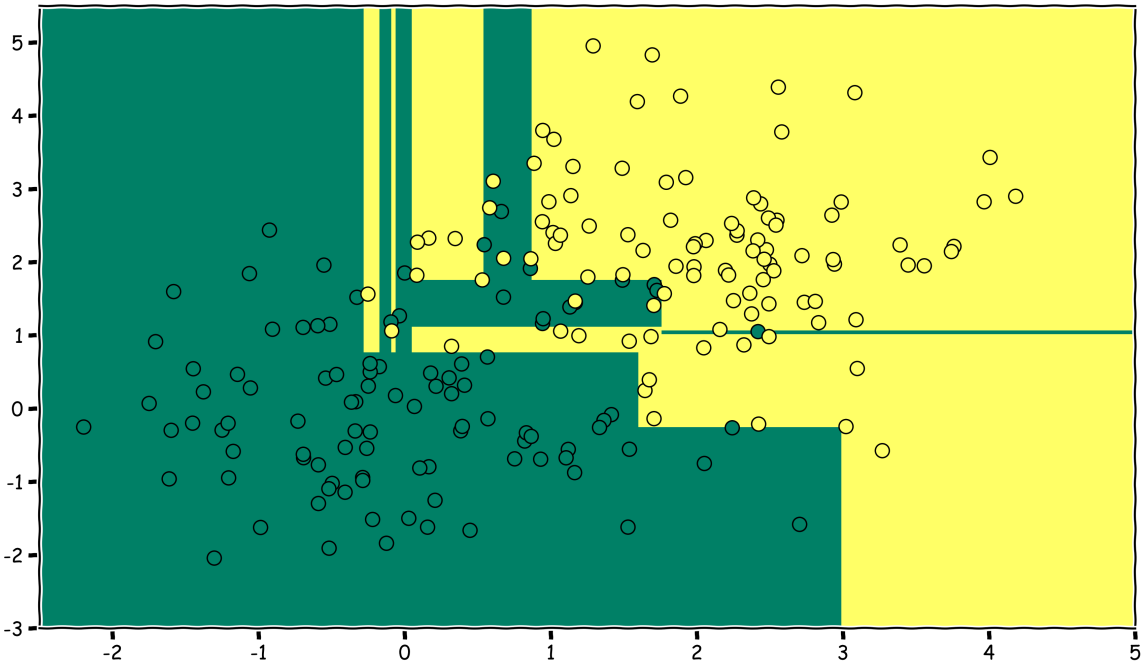

Figure 10: The decision tree classification (6 levels).

If $\mathsf{G}(\Omega) = 0$, then

$$\sum_{i=1}^{n} \mathsf{P}_i(1 - \mathsf{P}_i) = 0.$$

Since all the terms are non-negative, each of them equals 0. Thus, $\mathsf{P}_i(1-\mathsf{P}_i) = 0$ for all $i \in \{1, 2, ..., n\}$. So, for each P_i, one of two conditions is met: it's either 0, or 1. However, since the sum of all P_i equals 1, only one $\mathsf{P}_i = 1$, and the rest are equal to zero. Thus, the entropy of the experiment is zero. □

Similar to entropy, Gini impurity is maximized when all the outcomes of the experiment Ω are equipossible. The following observation is true.

Theorem 5.1.2 *Gini impurity* G *is maximized when all the outcomes of the experiment are equipossible, that is, when the experiment is described by the following table:*

Ω	ω_1	ω_2	...	ω_n
P	$\frac{1}{n}$	$\frac{1}{n}$	...	$\frac{1}{n}$

In this case, the Gini impurity equals:

$$\mathsf{G}(\Omega) = 1 - \frac{1}{n},$$

Proof. The experiment described by the table:

Ω	ω_1	ω_2	...	ω_n
P	$\frac{1}{n}$	$\frac{1}{n}$	...	$\frac{1}{n}$

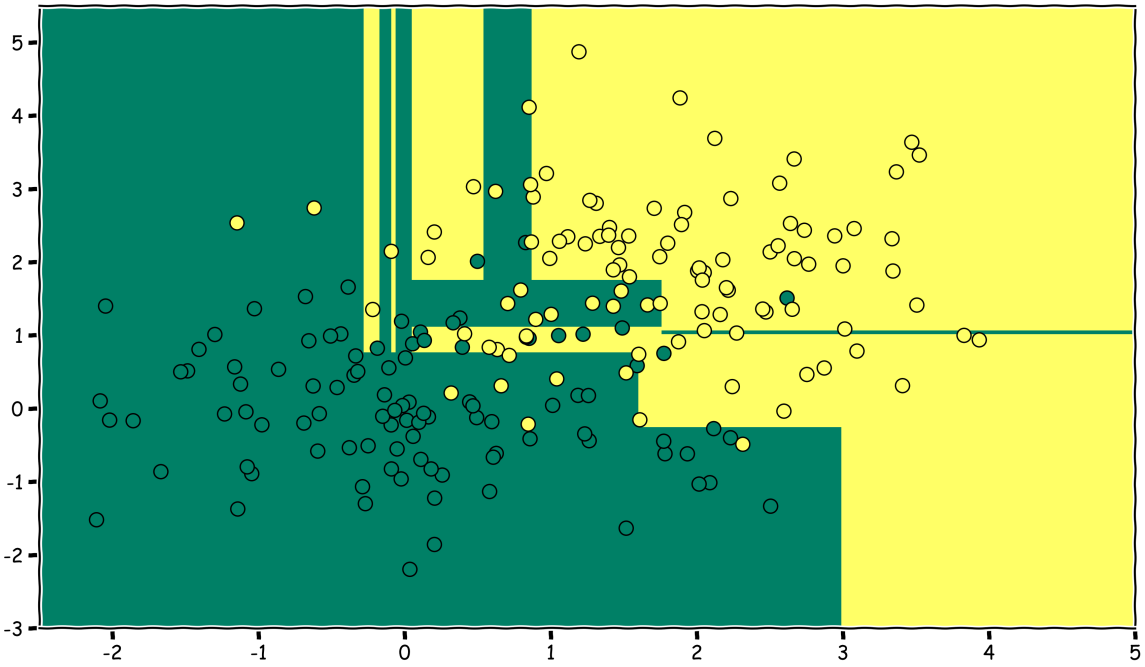

Figure 11: The classification of new objects using the trained model.

therefore,

$$\mathsf{G}(\Omega) = 1 - \sum_{i=1}^{n} \frac{1}{n^2} = 1 - \frac{1}{n}.$$

Let's prove that the value is maximal. Since the function $\frac{1}{x}$ is convex downward, Jensen's inequality can be used as follows: for any values $p_1, p_2, ..., p_n > 0$ so that $p_1 + p_2 + ... + p_n = 1$ and any $x_1, x_2, ..., x_n$ from the convex interval, the following is true:

$$f(p_1x_1 + p_2x_2 + ... + p_nx_n) \leq p_1f(x_1) + p_2f(x_2) + ... + p_nf(x_n)$$

or

$$f\left(\sum_{i=1}^{n} p_ix_i\right) \leq \sum_{i=1}^{n} p_if(x_i).$$

Let's assume that $p_i = \mathsf{P}_i$ and $x_i = \frac{1}{\mathsf{P}_i}$ to obtain

$$f\left(\mathsf{P}_1 \cdot \frac{1}{\mathsf{P}_1} + ... + \mathsf{P}_n \cdot \frac{1}{\mathsf{P}_n}\right) = f(n) = \frac{1}{n} \leq \sum_{i=1}^{n} \mathsf{P}_i \frac{1}{\frac{1}{\mathsf{P}_i}} = \sum_{i=1}^{n} \mathsf{P}_i^2,$$

hence,

$$1 - \sum_{i=1}^{n} \mathsf{P}_i^2 \leq 1 - \frac{1}{n}.$$

□

Similar to entropy, Gini impurity grows with the number of equipossible outcomes of the experiment Ω.

Corollary 5.1.3 *The latter theorem allows us to obtain an important corollary. Gini impurity* $\mathsf{G}(\Omega)$ *of any experiment* Ω *should be within the following range:*

$$0 \leq \mathsf{G}(\Omega) < 1.$$

It is reasonable to pose a question. How to define the Gini impurity for the experiment (Ω, Θ)? Recall that the experiment (Ω, Θ) is an arbitrary set of outcome pairs (ω_i, θ_j), $i \in \{1, 2, \ldots, m\}$, $j \in \{1, 2, \ldots, n\}$, each corresponding to a value $\mathsf{P}_{ij} \geq 0$ called the probability of the outcome (ω_i, θ_j) so that

$$\sum_{i=1}^{m} \sum_{j=1}^{n} \mathsf{P}_{ij} = 1.$$

The experiment (Ω, Θ) can be described by (and even considered as) the following table:

(Ω, Θ)	θ_1	θ_2	$\ldots$	θ_n
ω_1	P_{11}	P_{12}	$\ldots$	P_{1n}
ω_2	P_{21}	P_{22}	$\ldots$	P_{2n}
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
ω_m	P_{m1}	P_{m2}	$\ldots$	P_{mn}

Hence, for experiment Ω, given that the experiment Θ turned out to be θ_j, where $j \in \{1, 2, \ldots, n\}$, the following n conditional probability tables can be written:

(Ω, θ_j)	$(\omega_1 \mid \theta_j)$	$(\omega_2 \mid \theta_j)$	$\ldots$	$(\omega_m \mid \theta_j)$
P	$\mathsf{P}(\omega_1 \mid \theta_j)$	$\mathsf{P}(\omega_2 \mid \theta_j)$	$\ldots$	$\mathsf{P}(\omega_m \mid \theta_j)$

.

Each table describes the experiment for which Gini impurity can be calculated. Just as in the entropy case, let's introduce the following definition.

Definition 5.1.2 *A conditional Gini impurity of the experiment* Ω*, given that the experiment* Θ *turned out to be* θ_j*,* $j \in \{1, 2, ..., n\}$*, is a value*

$$\mathsf{G}(\Omega \mid \theta_j) = 1 - \sum_{i=1}^{m} \mathsf{P}^2(\omega_i \mid \theta_j).$$

Thus, for each state θ_j that occurs with some probability $\mathsf{P}(\theta_j)$, the Gini impurity takes values of $\mathsf{G}(\Omega \mid \theta_j)$. Thus, it's a random variable with a distribution series:

$\mathsf{G}(\Omega \mid \theta_j)$	$\mathsf{G}(\Omega \mid \theta_1)$	$\ldots$	$\mathsf{G}(\Omega \mid \theta_n)$
P	$\mathsf{P}(\theta_1)$	$\ldots$	$\mathsf{P}(\theta_n)$

.

The obtained system is well characterized by so-called weighted Gini impurity.

Definition 5.1.3 *Weighted Gini impurity of the experiment Ω given that the experiment Θ has occurred is called a value*

$$\mathsf{G}(\Omega|\Theta) = \mathsf{E}\left(\mathsf{G}(\Omega|\theta_j)\right) = \sum_{j=1}^{n} \mathsf{P}(\theta_j)\mathsf{G}(\Omega|\theta_j).$$

The information gain based on the concept of Gini impurity is introduced as follows.

Definition 5.1.4 *The Gini gain is a value*

$$\mathsf{GG}(\Omega|\Theta) = \mathsf{G}(\Omega) - \mathsf{G}(\Omega|\Theta).$$

The value of the Gini gain is used to construct decision trees in the same way as we used the information gain. Each time, the first step of the algorithm is to choose the largest Gini gain.

5.2 A Comparison of Gini Gain and Entropy

In practice, the considered criteria of information gain produce almost the same result. It's easy to prove by constructing, for example, graphs of entropy functions and double Gini impurity in the case of two outcomes:

$$I_{Entropy} = -p\log_2 p - (1-p)\log_2(1-p),$$

$$I_{Gini} = 2\cdot\left(p(1-p) + (1-p)(1-(1-p))\right).$$

Double Gini impurity allows normalizing the values of the function I_{Gini} to one, which equalizes the units of measurement. Fig. 10 shows that the impurity of the system with two outcomes for different values of probability is comparable.

Hence, there are no premises for using a particular method to measure information because both of them are coping well with the task and produce almost the same result. However, calculating entropy is more complicated (from a calculation standpoint) due to the logarithm mentioned in the definition. Due to this, Gini impurity is often used by default in different packages.

5.3 Gini Gain on Data

Let's return to the cat show example. The given table contains information about the breed, coat color, and height from floor to shoulder. All of these criteria affect the cuteness of a particular cat participating in the cat show.

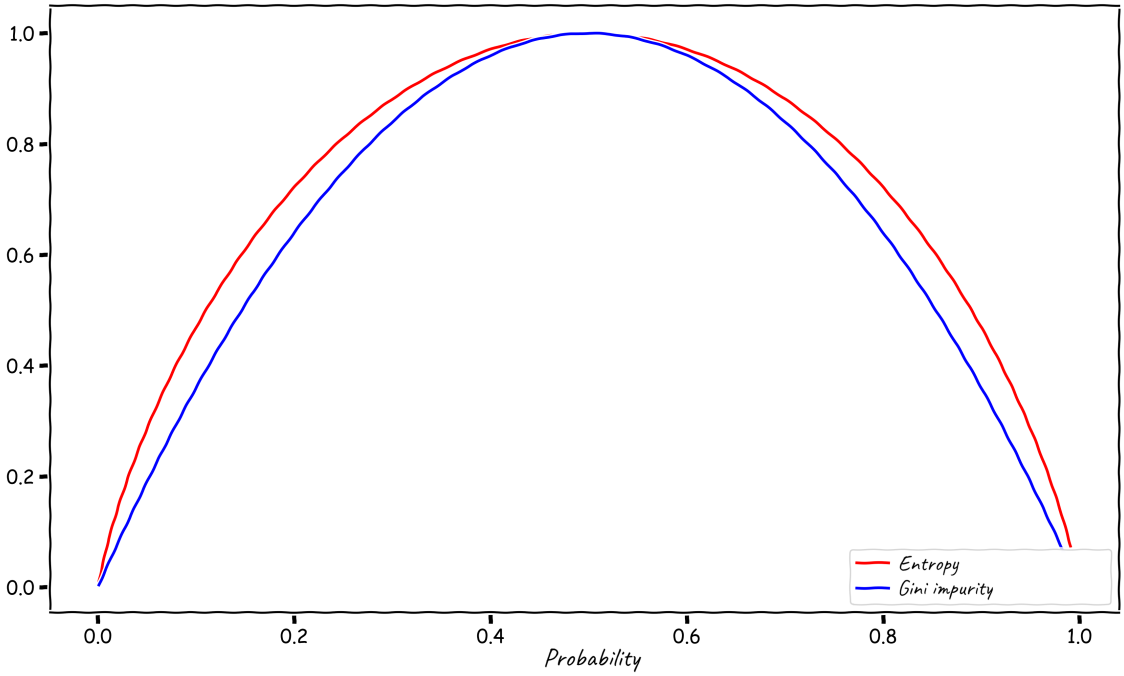

Figure 12: The comparison of entropy and Gini impurity.

Breed	**Coat**	**Height**	**Cuteness**
British	White	Tall	No
British	Gray	Tall	Yes
British	White	Short	Yes
Maine coon	White	Tall	No
Maine coon	Brown	Tall	Yes
Maine coon	Brown	Short	No
Ragdoll	Gray	Tall	Yes
Maine coon	Gray	Short	Yes

The conditional probabilities of assigning to the target class Cuteness based on a breed were found earlier. They are equal to:

$$\mathsf{P}(\text{Cuteness} = \text{Yes}|\text{Breed} = \text{British}) = \frac{2}{3},$$

$$\mathsf{P}(\text{Cuteness} = \text{No}|\text{Breed} = \text{British}) = \frac{1}{3},$$

$$\mathsf{P}(\text{Cuteness} = \text{Yes}|\text{Breed} = \text{Maine coon}) = \frac{1}{2},$$

$$\mathsf{P}(\text{Cuteness} = \text{No}|\text{Breed} = \text{Maine coon}) = \frac{1}{2},$$

$$\mathsf{P}(\text{Cuteness} = \text{Yes}|\text{Breed} = \text{Ragdoll}) = 1,$$

$$\mathsf{P}(\text{Cuteness} = \text{No}|\text{Breed} = \text{Ragdoll}) = 0.$$

Thus, Gini impurity for each breed will take the following values:

$$\mathsf{G}(\text{Cuteness}|\text{Breed} = \text{British}) = 1 - \left(\left(\frac{2}{3}\right)^2 + \left(\frac{1}{3}\right)^2\right) = \frac{4}{9},$$

$$\mathsf{G}(\text{Cuteness}|\text{Breed} = \text{Maine coon}) = 1 - \left(\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2\right) = \frac{1}{2},$$

$$\mathsf{G}(\text{Cuteness}|\text{Breed} = \text{Ragdoll}) = 1 - \left(1^2 + 0^2\right) = 0.$$

The states (the choice of a particular breed) are defined with some probability, that is: $\frac{3}{8}$, $\frac{4}{8}$ and $\frac{1}{8}$, respectively. Let's write the obtained values in the table:

$\mathsf{G}(\text{Cuteness}\|\text{Breed})$	$\frac{4}{9}$	$\frac{1}{2}$	0
P	$\frac{3}{8}$	$\frac{4}{8}$	$\frac{1}{8}$

.

The weighted Gini impurity will be:

$$\mathsf{G}(\text{Cuteness}|\text{Breed}) = \frac{3}{8} \cdot \frac{4}{9} + \frac{4}{8} \cdot \frac{1}{2} + \frac{1}{8} \cdot 0 = \frac{5}{12}.$$

The Gini impurity for the experiment Cuteness equals:

$$\mathsf{G}(\text{Cuteness}) = 1 - \left(\left(\frac{5}{8}\right)^2 + \left(\frac{3}{8}\right)^2\right) = \frac{15}{32},$$

thus, Gini gain:

$$\mathsf{GG}(\text{Cuteness}|\text{Breed}) = \frac{15}{32} - \frac{5}{12} = \frac{5}{96} \approx 0.052.$$

Similarly, Gini gain for the remaining attributes is found. The Coat attribute also best describes the information gain in our heuristics as in the entropy case.

6 A Real-World Example of Decision Tree Application

One of the examples of decision tree application is a scoring system used by banks to assess clients. For example, a bank has a client base and wants to find out who will accept a loan offer and who will decline it.

Let's consider the data (fig. **??**) about existing bank clients[1]. There are 12 predictors, including age, employment history, income, marital status, education, and so on, and the response $PersonalLoan$ taking the value 0 when the client

[1] https://www.kaggle.com/itsmesunil/bank-loan-modelling

declined the offer, and 1 when the client took the offer. The input dataset contains 5 thousand objects. To evaluate the model quality, the input dataset is split into training and test sets in the ratio of 70% to 30%. Hence, the model is trained on 3500 objects. 3158 of them belong to the class 0. These clients declined a loan offer. The rest of the clients are assigned to the class 1 of those who accepted the offer. Note that all the predictors take numeric values, which makes them

ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online	CreditCard
1	25	1	49	91107	4	1.6	1	0	0	1	0	0	0
2	45	19	34	90089	3	1.5	1	0	0	1	0	0	0
3	39	15	11	94720	1	1.0	1	0	0	0	0	0	0
4	35	9	100	94112	1	2.7	2	0	0	0	0	0	0
5	35	8	45	91330	4	1.0	2	0	0	0	0	0	1

Figure 13: The training data example.

non-categorical. Let's construct a binary decision tree. As a splitting criterion, we will use the criterion $X_i \leq C$, where C is a cut-off value from the range of values that the predictor X_i can take. At the first level, the most informative is the income-based splitting (the predictor $Income$) with the cut-off value of 92.5 thousand dollars a year. This splitting leads to the informative set S_{11} with near-zero entropy. At the same time, there is a high degree of uncertainty on the second set S_{12}.

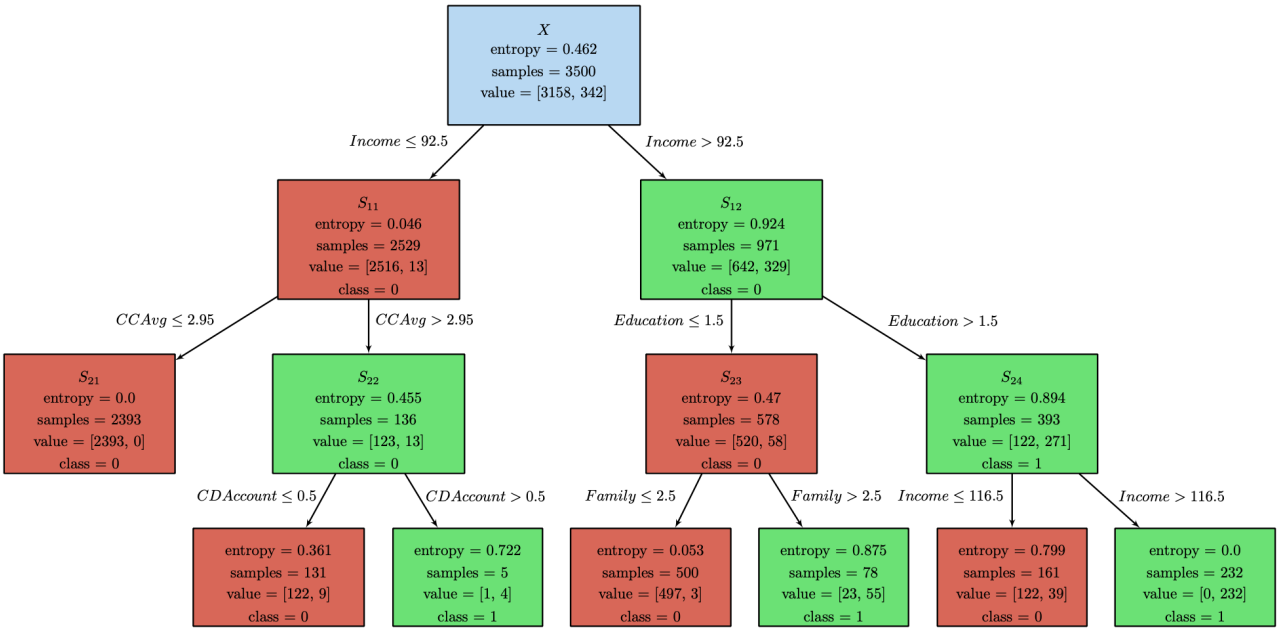

Figure 14: The decision trees (entropy).

Now let's construct the second level of the tree. For S_{11}, the most informative is the splitting based on the predictor $CCAvg$ (credit card spending per month).

For S_{12}, it will be *Education* (it takes 1 for students, 2 for graduates, and 3 for those with a degree). As a result, we obtain a subset of 2393 clients at the second level. They are assigned to the class 0 (the left box with zero entropy). These clients declined a loan offer. Their income was less than 92.5 thousand dollars a year, and their credit card spending per month didn't exceed 2.95 thousand dollars.

The third level of the tree was constructed out of the remaining sets with zero entropy (that is, of sets S_{22}, S_{23}, S_{24}). At this level, we also obtain the set with zero entropy (the right bottom box in fig. **??**). This time, the box corresponds to clients who accepted a loan offer. Who are these clients according to the constructed tree? They are the clients with high income (more than 116.5 thousand dollars a year) who graduated or graduated with a degree.

The remaining subsets include different clients, and we can either continue constructing the tree or end with three levels. In this case, the class is assigned based on the majority.

The test data classification and the constructed tree can be used to plot the confusion matrix: The obtained results indicate that the model has high precision

Confusion matrix		Initial class	
		+	−
Prediction	+	TP=108	FP=7
	−	FN=30	TN=1355

$$Precision = \frac{TP}{TP + FP} = \frac{108}{108 + 7} \approx 0.9391,$$

and recall

$$Recall = \frac{TP}{TP + FN} = \frac{108}{108 + 30} \approx 0.7826.$$

Training on similar data with Gini impurity produces the result shown in fig. **??**. Most of the criteria are similar to the tree constructed earlier. Besides, test data classification gives similar results. Precision and recall have slightly

Confusion matrix		Initial class	
		+	−
Prediction	+	TP=113	FP=4
	−	FN=25	TN=1358

increased:

$$Precision = \frac{TP}{TP + FP} = \frac{113}{113 + 4} \approx 0.9658,$$

$$Recall = \frac{TP}{TP + FN} = \frac{113}{113 + 25} \approx 0.8188.$$

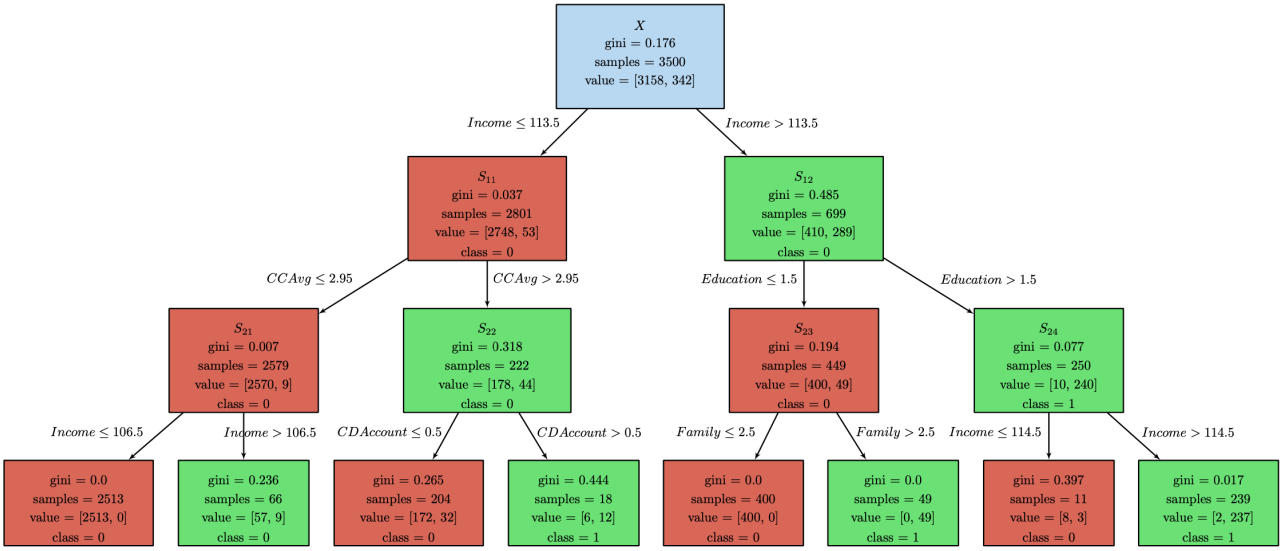

Figure 15: The decision tree (Gini).

7 Conclusion

This module covered one more classification technique called decision trees. They are mainly used for decision support in business and management. This technique doesn't offer such sophisticated features as hyperparameter tuning, but allows fast training. The choice of an uncertainty measure has no significant impact on classification results. Besides, limitations on tree depth or a number of elements in each node of the tree are heuristic approaches. Thus, they apply only to specific cases or don't guarantee the best results. Moreover, there are no proven recommendations on best approaches, and the final decision is up to a researcher.