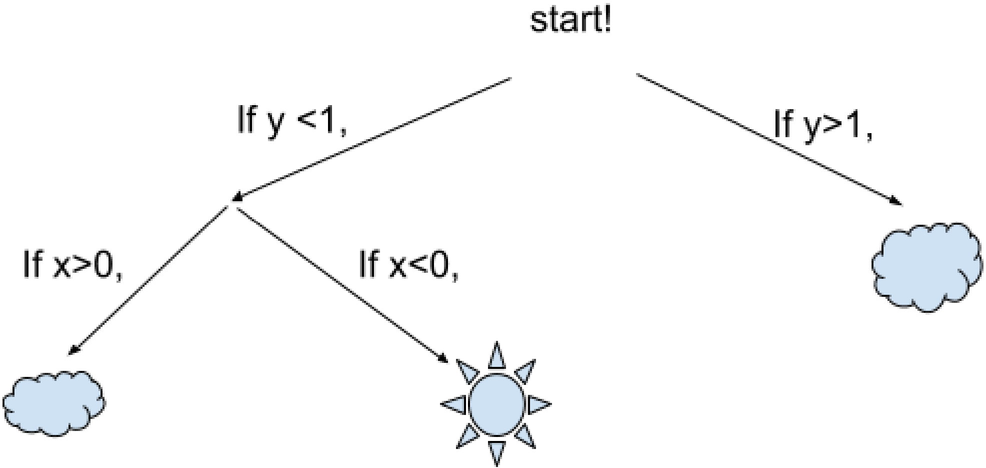
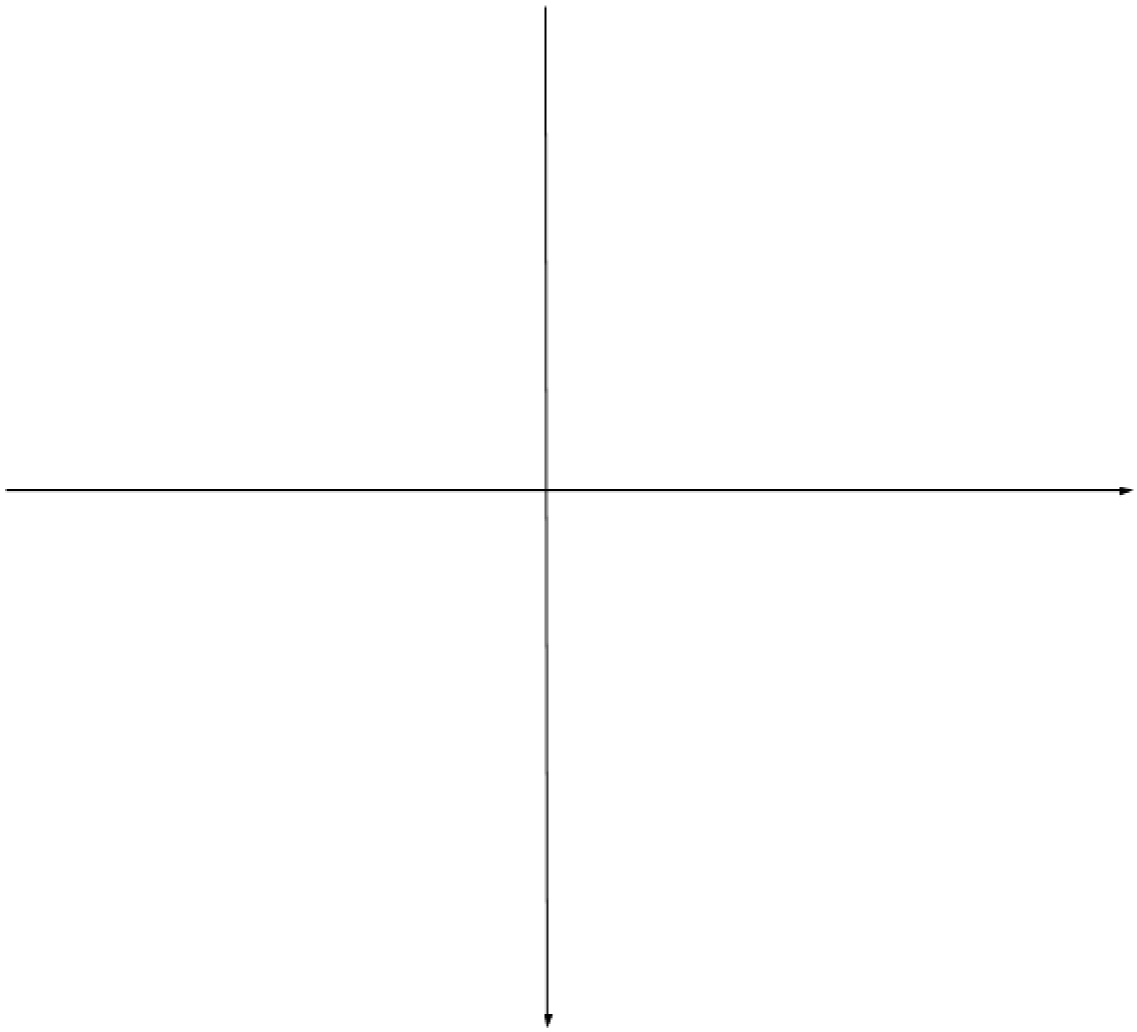


start!
If y <1,
If y>1,
If x>0,
If x<0,



The sun and cloud symbols above are fun but also point to a set of examples people have used in textbooks about machine learning and data mining (another set of words for learning from data). Here's a version of the problem:

You are trying to figure out if your little sister's soccer game will happen today. Last year, you know the following happened:

Temperature	Rain	Humidity	Play
90	no	high	no
73	no	low	yes
81	yes	high	no
67	no	high	yes
72	yes	high	yes
77	no	low	yes

96	no	low	no
81	yes	high	no
58	yes	high	yes
72	no	medium	yes

This weekend, you don't think it will rain but it'll be 90 degrees and 76 percent humidity. Do you think the game will happen? What about if it will rain but it will be 70 degrees with low humidity?

Here there are two outcomes (play/don't play) but three variables. Hard to draw a picture. Could you make a decision tree that works pretty well by trial and error? Use this as scratch paper and try making a decision tree:

Deciding on the decision tree

Back to the usual problem: what's the best decision tree? Depends on your definition of "best", of course! Two ways are really common: Gini index or Gini impurity, and information gain. We'll concentrate on information gain in this camp.

Information gain uses the concept of information entropy, how much information is contained in a set of data (!). Yes, this entropy is also the measure of disorder in the universe! Entropy is a number between zero and one. You can think of it as a measure of disorder or of uncertainty. For instance,

- if you have a two-headed coin, there is no uncertainty about the outcome! Its entropy is zero.
- If you have a fair coin, heads and tails are equally likely outcomes. There is maximum uncertainty. The entropy of the outcome is one.
- If you have an unfair coin that comes up heads ¾ of the time, there is less uncertainty! You'd want to bet on heads, right? The entropy of the outcome is about 0.81128. Can you check this?

Information gain is about the change in entropy as you make branches in the decision tree. We'll implement entropy calculations in Excel. They are kind of a pain to do. You will be very happy to move to Python….

Gini impurity measures how likely a mistake is if you stopped at a branch and just randomly picked some outcome from that division. Ask if you want to know more.

Let's look at entropy and information gain.

- You're doing a classification problem, so you want to classify outcomes as "yes" or "no", for instance. Let p_1 be the fraction of "yes" outcomes and p_2 the fraction of "no" outcomes.
- The entropy of the whole set, or information content, is $-p_1 \log_2 p_1 - p_2 \log_2 p_2$.
- If only one outcome happens (only "yes" or only "no" in our example) we say the entropy is zero.

Question 1: For the soccer game example, what is the entropy of the set, using p_1 for the fraction of games cancelled last year and p_2 for the fraction of games played as scheduled?

To decide what variable to split on, we need to find out how much the entropy changes in each case.

- If we split on the rain variable, we need to find the average entropy of each branch.
- If we split on the humidity variable, we need to find the average entropy of each branch.

- Temperature is a continuous variable instead of a categorical variable, so we'd have to try a bunch of temperature splits and test all of them! You can see why we want to use a computer here!

Question 2: Try computing the average entropy for splitting on the rain variable (so find the entropy for how many games played/cancelled for rain = yes, and the entropy for games played/cancelled for rain=no, and take their average).

Use Excel to do entropy calculations for you