



Describe the glass

“Is the glass half empty or half full?”

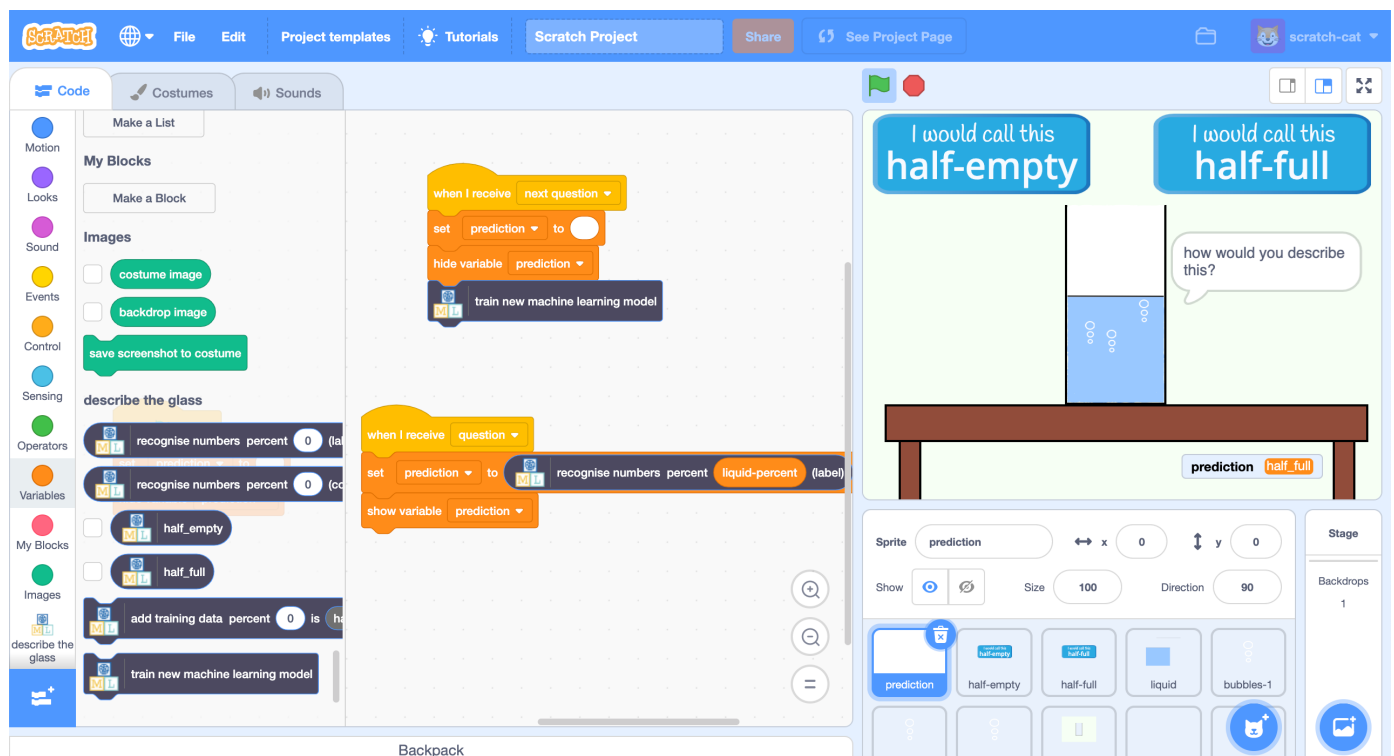
[Wikipedia describes this question](#) as a demonstration that a “situation may be seen in different ways depending on one’s point of view”, and as a test to “determine an individual’s worldview”.

A pessimistic person might describe a glass as half-empty.

An optimistic person might describe the same glass as half-full.

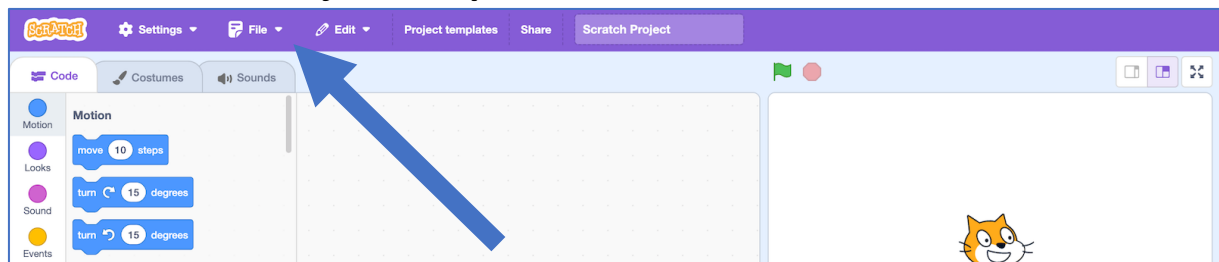
In this project, you’re going to teach a computer how you answer this question. And then you’ll see what it has learned from you.

The idea for this project came from Stanislav Gerasimovich. You can see his version of it at <https://sta-ger.bitbucket.io/apps/beer>

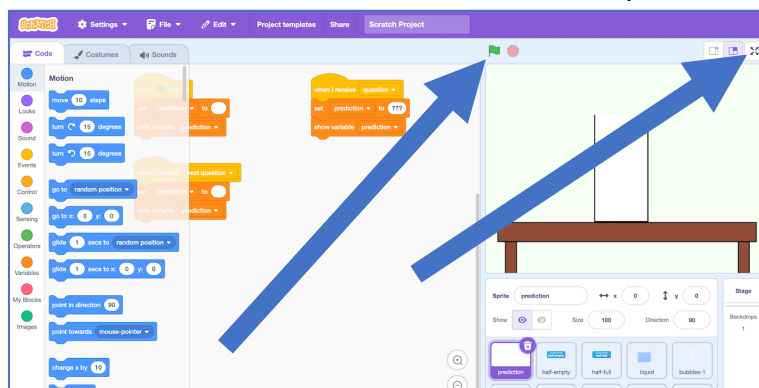


This project worksheet is licensed under a Creative Commons Attribution Non-Commercial Share-Alike License
<http://creativecommons.org/licenses/by-nc-sa/4.0/>

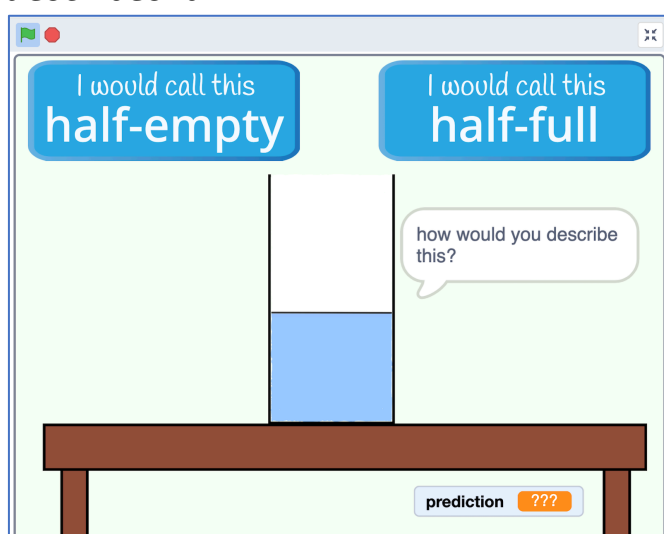
1. Go to <https://machinelearningforkids.co.uk/scratch>
2. Click on “Project templates”



3. Click on the “Describe the glass” template
4. Click on the **full-screen** button, then the **Green Flag**



5. Use the buttons at the top to answer a few questions
After explaining the idea, the Scratch project will show you a glass with a random amount of liquid and ask you to click the button you think describes it.



What have you done so far?

You based your description of the glass on how much liquid there was.

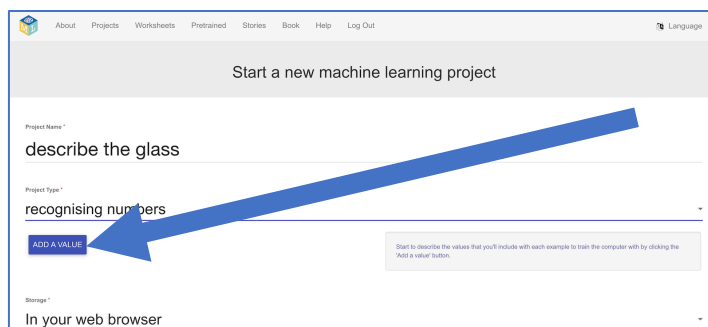
At the bottom of the screen, you can see the space where the computer is going to display a prediction of what you think. In this project, you are going to update the Scratch code so that it can generate this prediction.

You could do this by writing a rule that the computer could follow.

But, for this project, you're going to train the computer so that it learns for itself how you think about this question.

You'll collect examples of how you answer the question and use them to train a machine learning "model" that predicts what your answer will be.

6. Go to <https://machinelearningforkids.co.uk/>
7. Click on **"Get started"**
8. Click on **"Try it now"**
9. Click the **"+ Add a new project"** button.
10. Name your project **"describe the glass"** and set it to learn how to recognise **"numbers"**.
11. Choose where you want to store your project.
12. Click on **"Add a value"**



13. Create a “**number**” value called “percent”

14. Click on the “**Create**” button

Start a new machine learning project

Project Name *
describe the glass

Project Type *
recognising numbers

Value 1 *
percent

Type of value *
number

If percent can be described as numbers, choose "number".
If it can be described as choosing from a few options, choose "multiple-choice".

ADD ANOTHER VALUE

Storage *
In your web browser

CREATE CANCEL

15. “describe the glass” will be added to your list of projects. Click on it.

16. You need to prepare the types of prediction you want the computer to make. Click the “**Train**” button.

Language

"describe the glass"

Train
Collect examples of what you want the computer to recognise
Train

Learn & Test
Use the examples to train the computer to recognise numbers
Learn & Test

Make
Use the machine learning model you've trained to make a game or app, in Scratch or Python
Make

17. Click on “+ Add new label” and call it “half-empty”. Do that again and create a second bucket called “half-full”.

Language

Recognising **numbers** as **half_empty** or **half_full**

< Back to project

+ Add new label

half_empty

+ Add example Add file

half_full

+ Add example Add file

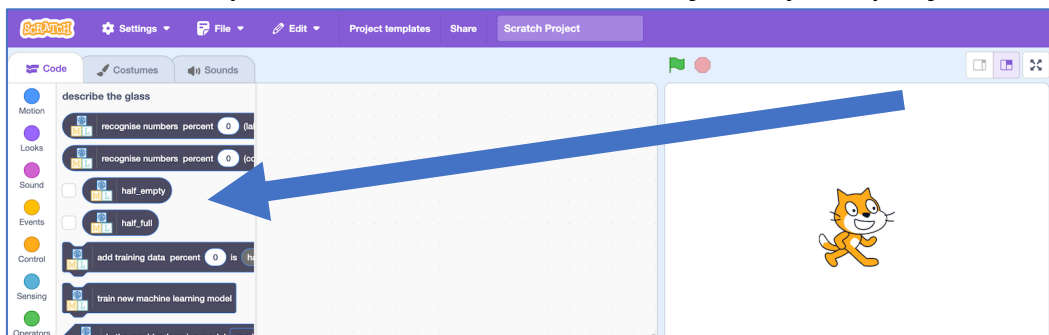
18. Click on the “< Back to project” link in the top-left

19. Click on the “Make” button

20. Click on the “Scratch 3” button

21. Click on the “straight into Scratch” button

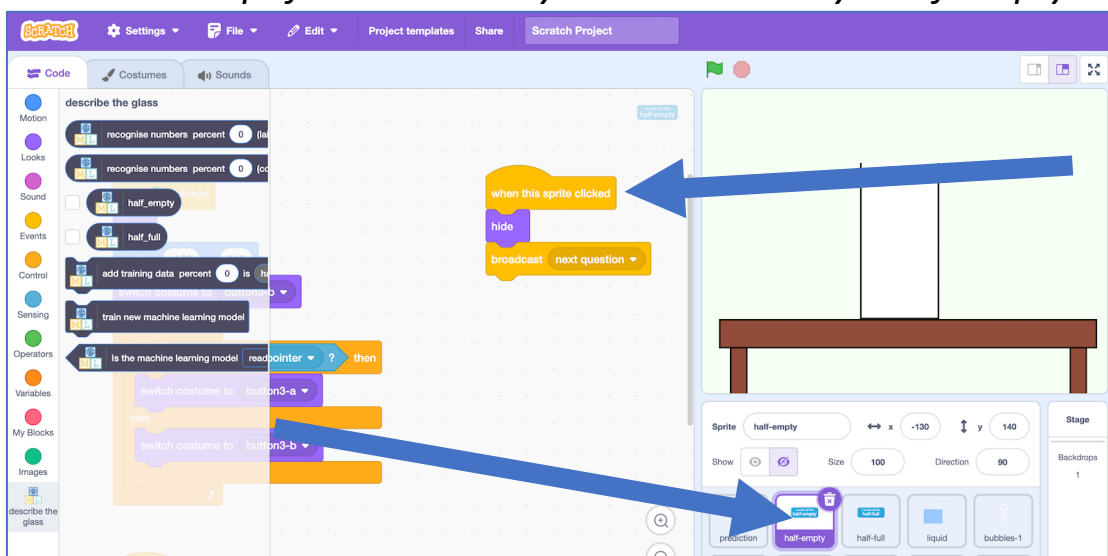
The page will warn you that you haven’t trained a model yet, but that’s okay as you’ll be using Scratch to collect training examples first. Scratch will open with additional blocks from your project.



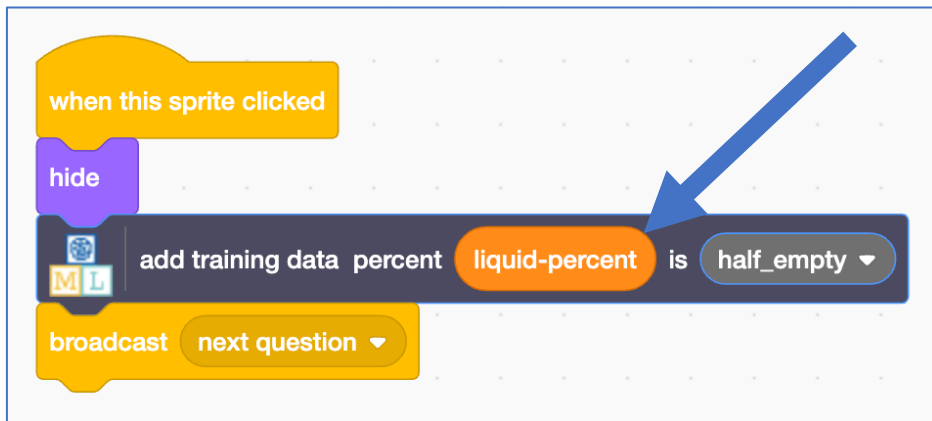
22. Click on the “Project templates” button.

23. Open the “Describe the glass” project template again.

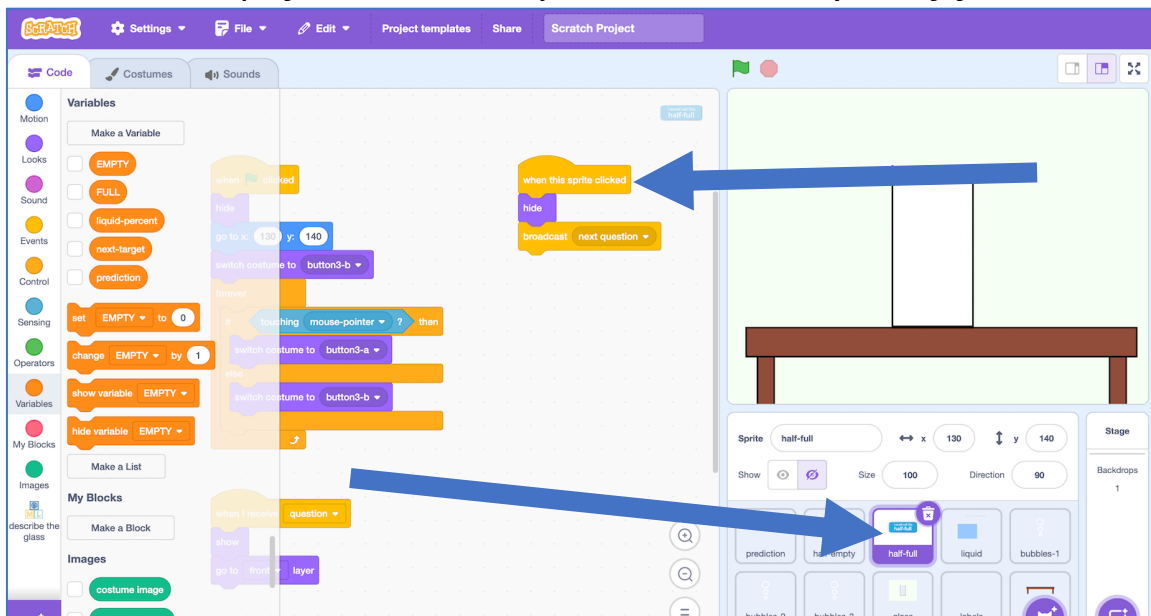
24. Click the “half-empty” sprite & find the “when this sprite clicked” script
This is the script for the button you click on to say “half-empty”



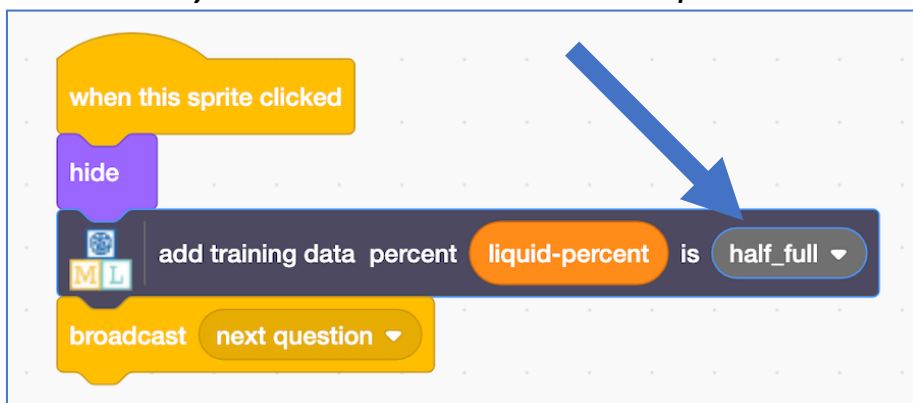
- 25.** Add an “add training data” block, with the “liquid-percent” value
This will add the example to your training bucket for “half-empty”



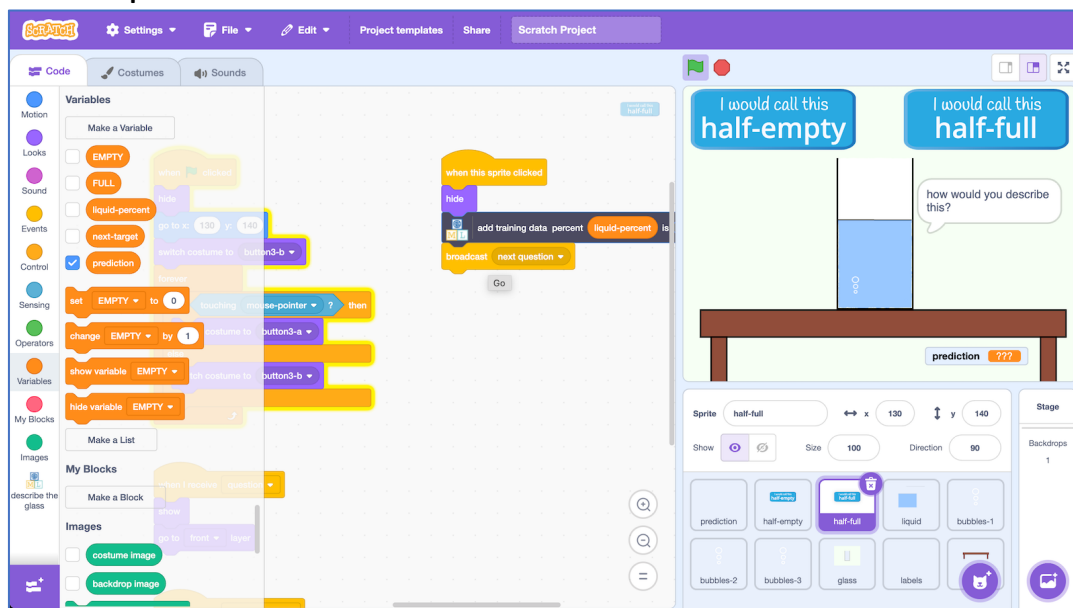
- 26.** Click the “half-full” sprite & find the “when this sprite clicked” script
This is the script for the button you click on to say “half-full”



- 27.** Add an “add training data” block, with the “liquid-percent” value
Make sure you choose to add the example to the “half-full” bucket



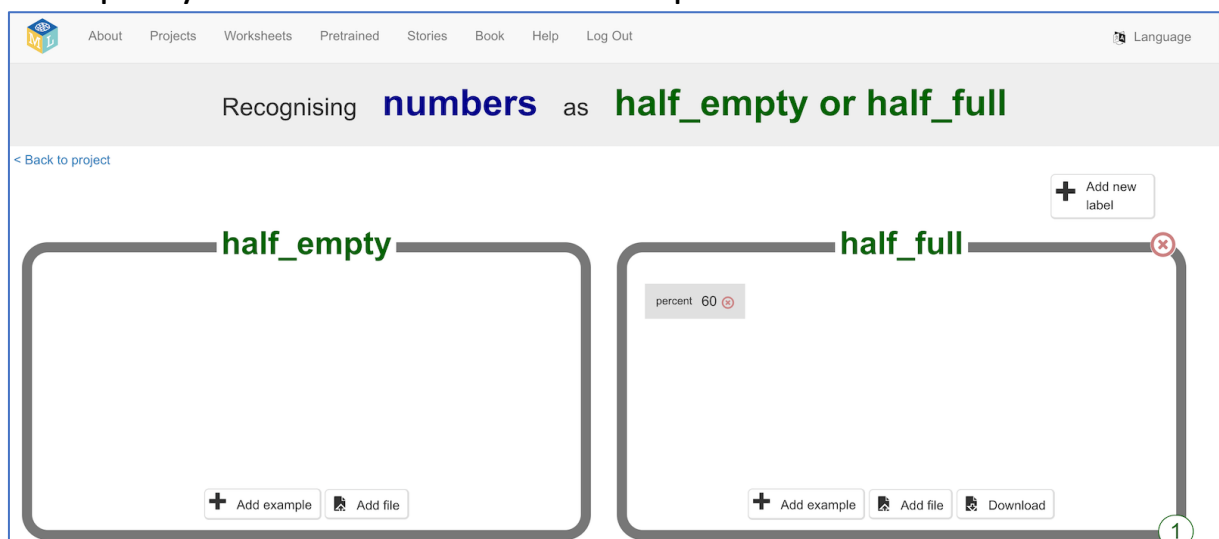
28. Click on the “**Green Flag**” and answer the “how would you describe this?” question once



29. In your other web browser window still on the machine learning tool, click on the “< **Back to project**” link in the top-left corner.

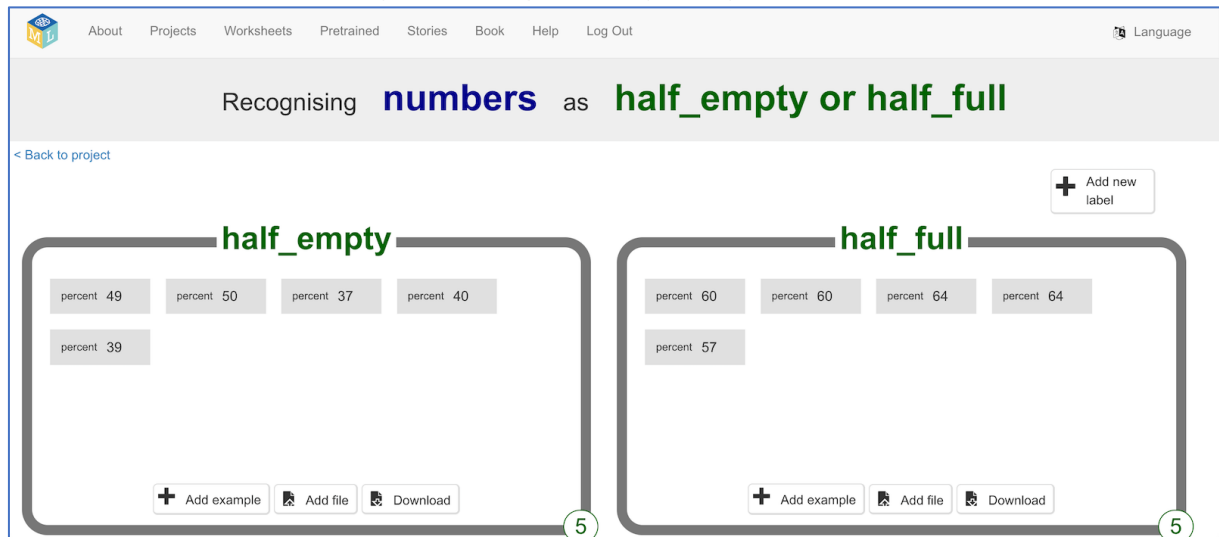
30. Click on the “**Train**” button.

31. Check that the answer you just gave has been added to the training examples you will use to train the computer.

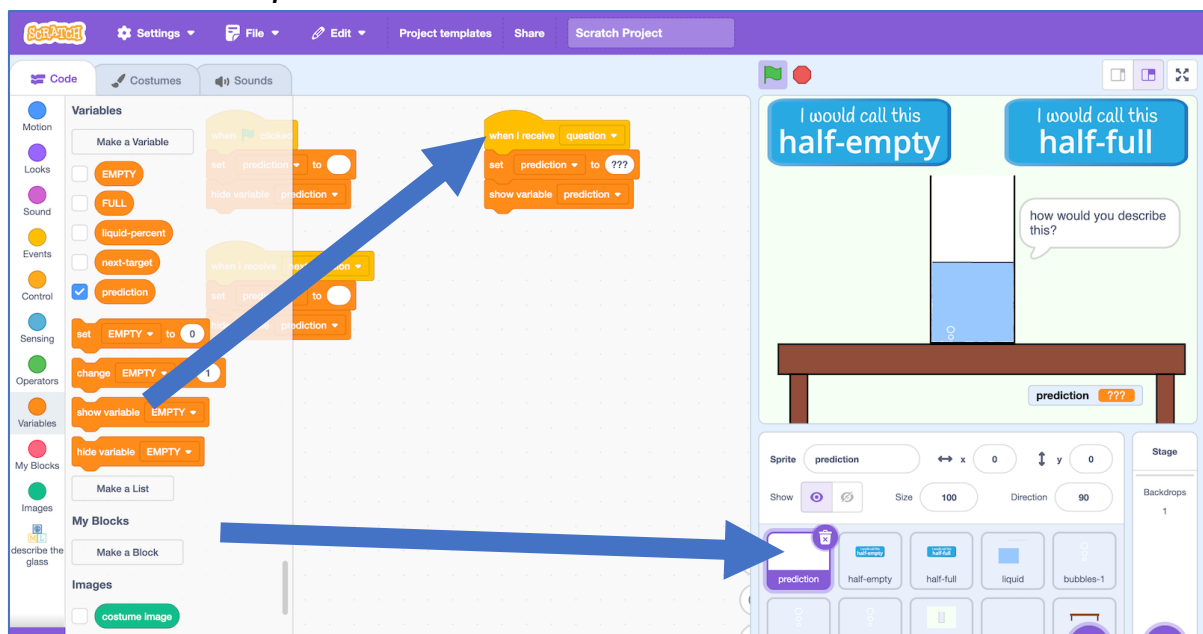


32. Go back to Scratch, and play the game again **nine** more times. *You might find it easier to play the game in full-screen mode.*

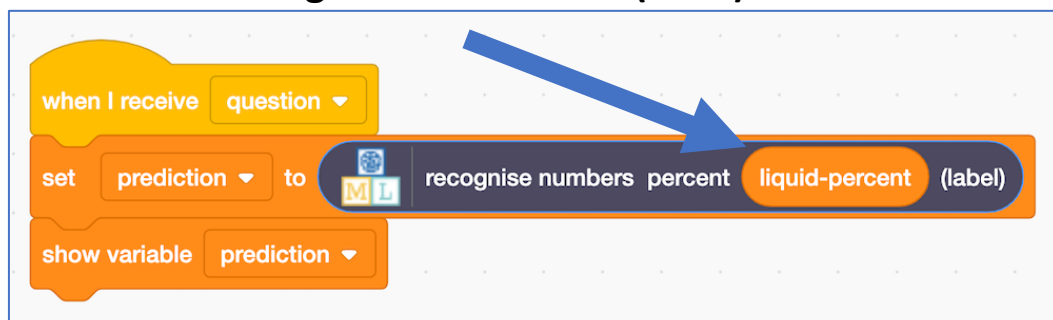
33. Check how many training examples you've collected



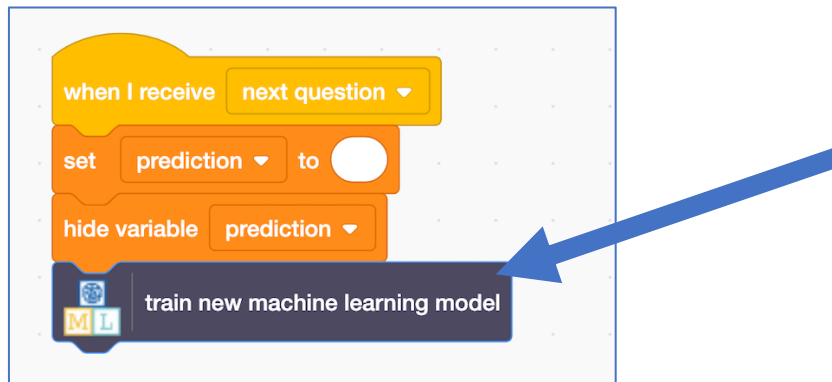
34. Click the “prediction” sprite & find the “when I receive question” script *You should have enough examples now to try using a machine learning model to make predictions.*



35. Add a “recognise numbers ... (label)” block to make a prediction



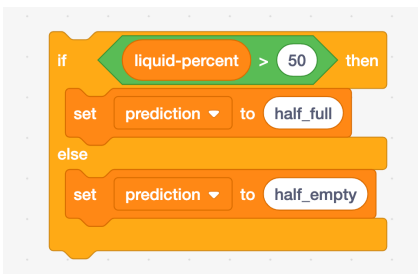
- 36.** Find the “when I receive next question” script
Add a “train new machine learning model” block



- 37.** Click on the full-screen button and then the **Green Flag** again
How good is your machine learning model at predicting your answers?

What have you done so far?

You’ve started to train a computer to predict your answers.



You could have written a rule to do this.

For example, if you knew that you always describe more-than-half as “half-full” and less-than-half as “half-empty”, you could have written code like this.

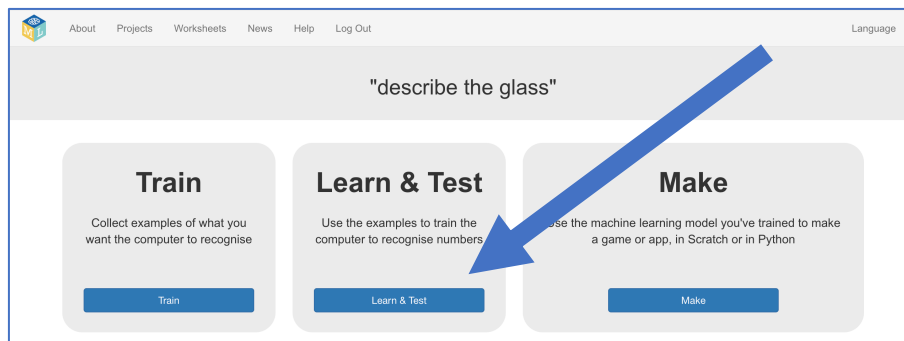
We use machine learning when we’re not sure how to write instructions the computer should follow, or if we think that will be too complicated.

Instead of writing instructions for the computer to follow, we let the computer learn for itself how it should do something by showing it examples.

The computer will learn from patterns in the examples. It will use these to make predictions.

38. On the machine learning tool page, click the “< Back to project” link

39. Click on “Learn & Test”



40. Click on “Describe your model”

*This page will show you a picture of your machine learning model.
Read the page to understand what it means.*

A screenshot of the "Understand your machine learning model" page. The page has a navigation bar at the top with links: About, Projects, Worksheets, News, Help, Log Out, and a Language dropdown. Below the navigation bar is a header area with the text "Understand your machine learning model". The main content area starts with a link "< Back to model". The text explains that the technique used is called a "Decision Tree Classifier". It states that this is not the only way to train a machine learning model, but it's very quick and easy to train, and it's one of the easiest techniques to understand. The page shows the decision tree created based on the training examples. A diagram of a decision tree is shown. The root node is a light blue box with the text "Is this test true?", "samples = how many training examples got here", and "class = prediction so far". It has two arrows pointing down: "Go this way if the test is true" to the left and "Go this way if the test is false" to the right. The left branch leads to an orange box with the text "percent <= 46.5", "samples = 53", "value = [22, 31]", and "class = half_full". The right branch leads to a blue box with the text "percent > 46.5", "samples = 31", "value = [0, 31]", and "class = half_full". The orange box has two arrows pointing down: "True" to the left and "False" to the right. The "True" branch leads to an orange box with the text "samples = 22", "value = [22, 0]", and "class = half_empty". The "False" branch leads to a blue box with the text "samples = 31", "value = [0, 31]", and "class = half_full". To the right of the decision tree diagram is a testing interface. It has a search bar, a "Test" button, and a "Reset" button. The text "Try out your machine learning model to see how it uses the decision tree to make predictions" is above the search bar. The search bar contains the text "percent".

What have you done?

The type of machine learning model you've trained is a “**decision tree classifier**”. The visualisation lets you see how your model makes predictions. It's a good way to see what patterns the computer found in the training data you collected.

The visualisation will show the patterns that the computer observed in your answers.

For example, in the screenshot on the last page, you can see that my machine learning model learned that I tend to describe anything over 46% as half-full.

What did your machine learning model learn about your answers?

Was that what you expected?

(If your answers weren't always consistent, your machine learning model might have a more complicated visualisation as the computer tried to show the different ways that you answer.)

Try comparing your machine learning model with a model trained by someone else. Did the computer learn that they were more or less optimistic than you?

Ideas and Extensions

Now that you've finished, why not give one of these ideas a try?

Or come up with one of your own?

Add additional predictions

Instead of just having two training buckets ("half-full", "half-empty") try adding more.

For example, try training a machine learning model to recognize "nearly empty", "half empty", "half full", "nearly full"

Add additional variables

What other factors might influence your answers apart from the amount of liquid?

For example, do you answer differently for different coloured liquids? Or different shaped glasses?

What about if the liquid has increased or decreased since the last question? Does that change how you answer?

Try making your own project to see how the computer learns a more complex set of patterns that affect your answers.

Try recognizing the picture of the glass

Instead of training the computer to recognize how full a glass is by describing it with a number, we could have used a picture of the glass. Try doing this as an images project instead of numbers.