



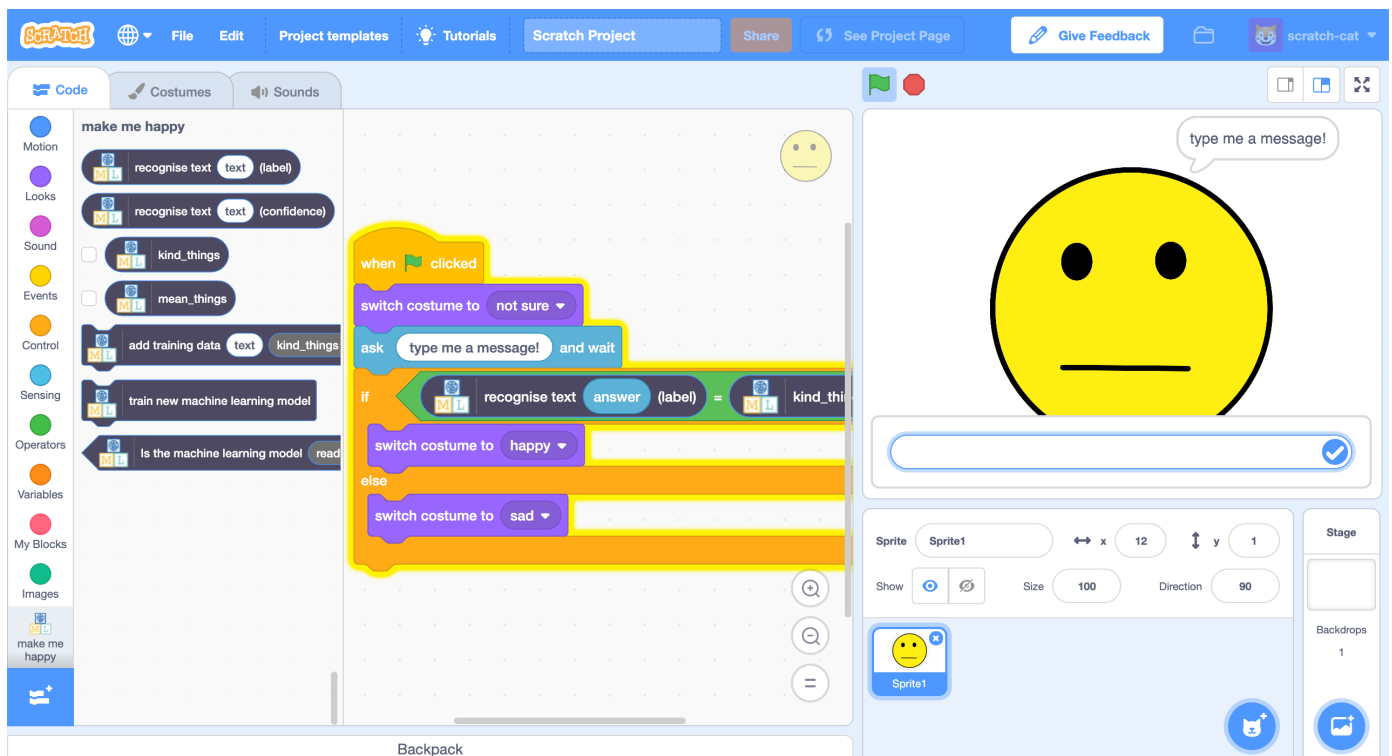
Make me happy

In this project you will make a character that reacts to what you say.

If you compliment it, it will look happy.

If you insult it, it will look sad.

You will teach the computer to recognise kind messages and mean messages by giving it examples of each.



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/> in a web browser
2. Click on **“Get started”**
3. Click on **“Try it now”**
4. Click the **“+ Add a new project”** button.
5. Name your project **“make me happy”** and set it to learn how to **recognise text**. Choose to store your project **in your web browser**. Click the **“Create”** button

Start a new machine learning project

Project Name *
make me happy

Project Type *
recognising text

Language
English

Storage *
In your web browser

Where do you want to store this project?
Storing in your web browser removes limits on how long your project can be.
Storing in the cloud will let you access the project from anywhere.
(See "What difference does it make where a project is stored" for more details.)

CREATE CANCEL

6. You should now see **“make me happy”** in the list of your projects. Click on it.
7. You need examples to train the computer. Click the **Train** button.

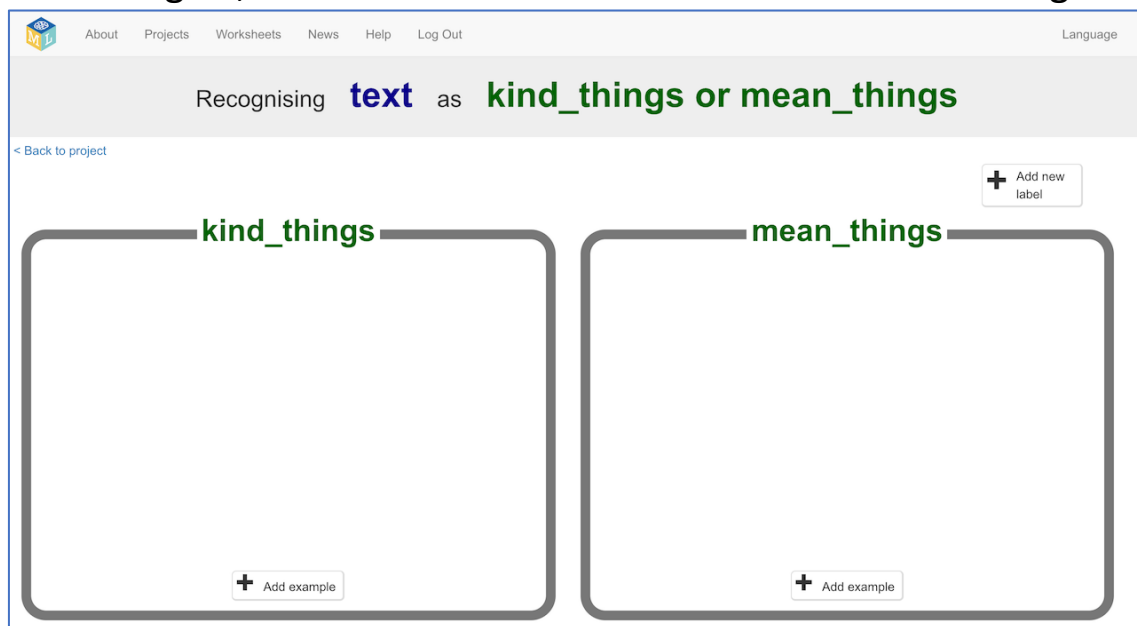
"make me happy"

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

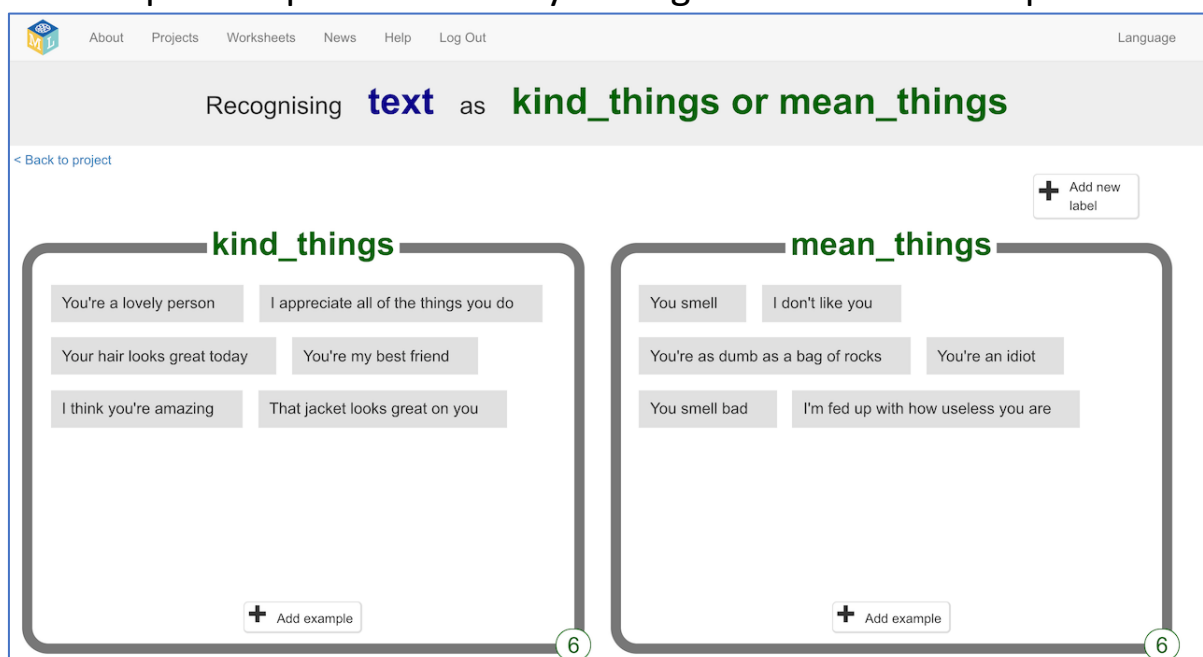
Learn & Test
Use examples to train the computer to recognise text
Learn & Test

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

8. Click on “+ Add new label” and call it “kind things”.
Do that again, and create a second bucket called “mean things”.



9. Click the “Add example” button in the “kind things” bucket, and type in the nicest, kindest compliment that you can think of.
10. Click on the “Add example” button in the “mean things” bucket, and type in the meanest, cruellest insult that you can think of.
11. Repeat steps 9 & 10 until you’ve got at least **six** examples of each.

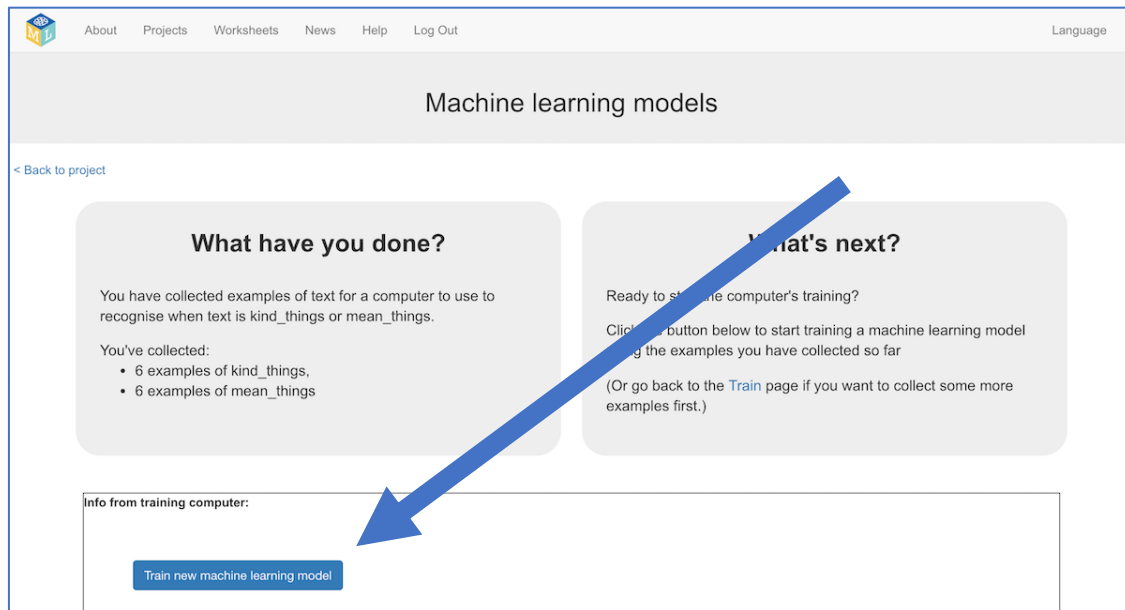


12. Click on the “< Back to project” link.

13. Click on the “Learn & Test” button.

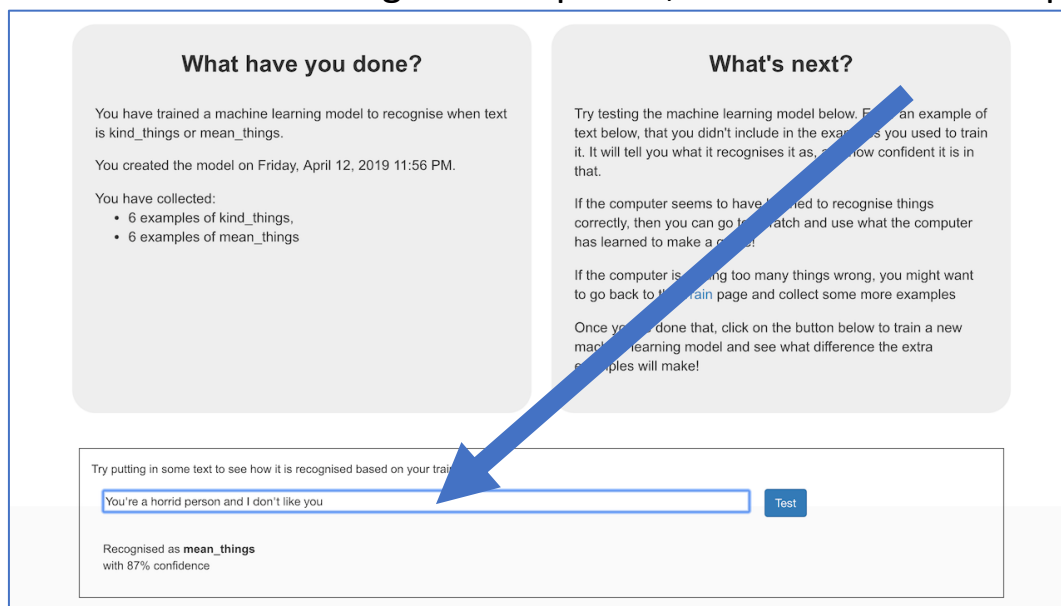
14. Click on the “Train new machine learning model” button.

As long as you’ve collected enough examples, the computer should start to learn how to recognise messages from the examples you’ve given to it.



15. Wait for the training to complete. This might take a few minutes.

16. Once the training has completed, a Test box will be displayed.



Try testing your machine learning model to see what the computer has learned.

Type something kind, and press enter. It should be recognised as kind.

Type something mean, and press enter. It should be recognised as mean.

Test it with examples that you haven't shown the computer before.

If you're not happy with how the computer recognises the messages, go back to step 11, and add some more examples.

Make sure you repeat step 13 to train with the new examples though!

17. Click the “< **Back to project**” link

18. Click the “**Make**” button

19. Click the “**Scratch 3**” button.

This page has instructions on how to use the new blocks in Scratch.

Keep the page open if you need to check back on how to use them.

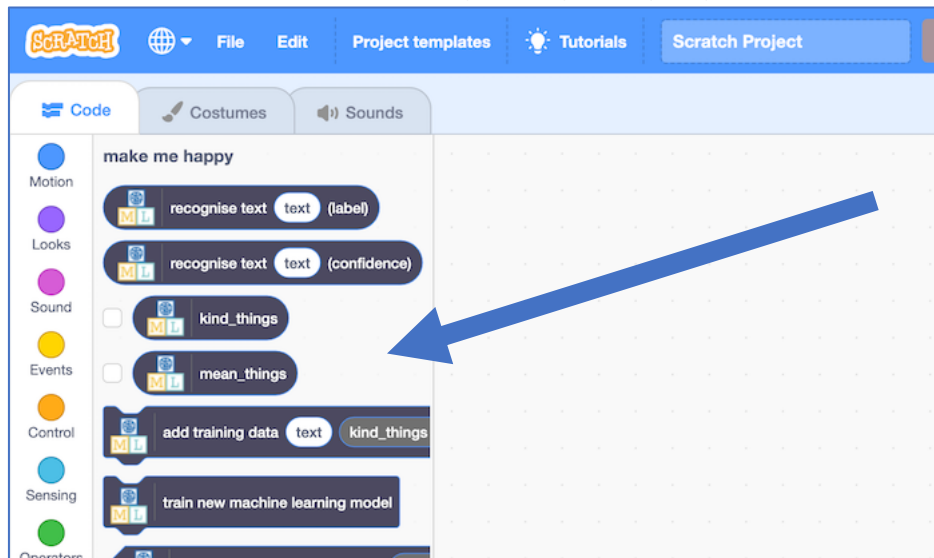
What have you done so far?

You've started to train a computer to recognise text as being kind or mean. Instead of trying to write rules to be able to do this, you are doing it by collecting examples. These examples are being used to train a machine learning “model”.

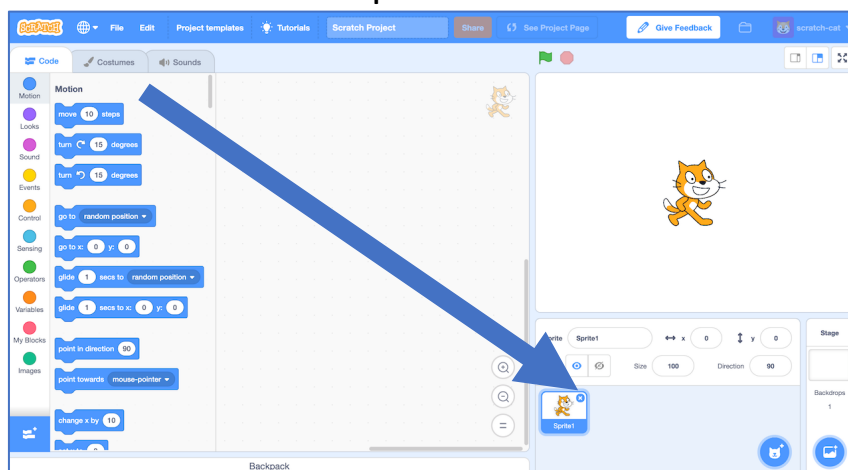
This is called “supervised learning” because of the way you are supervising the computer's training.

The computer will learn from patterns in the examples you've given it, such as the choice of words, and the way sentences are structured. These will be used to be able to recognise new messages.

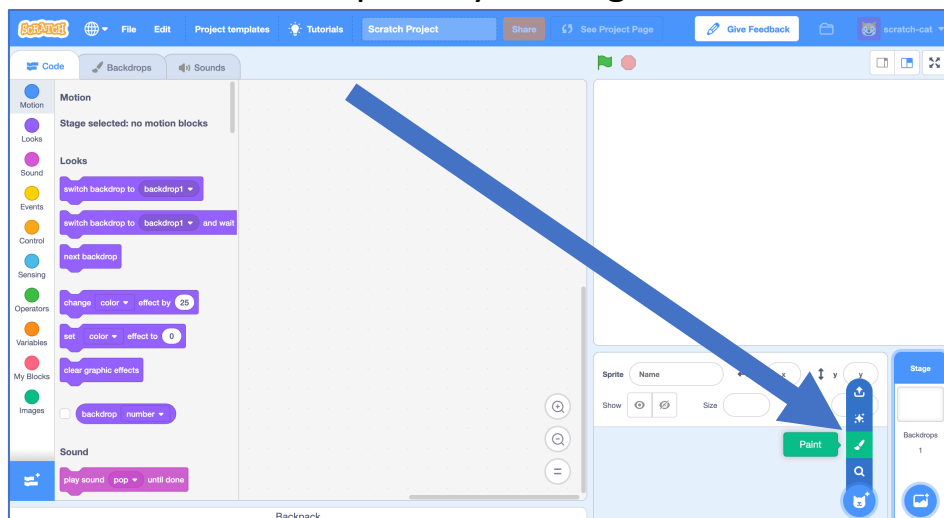
20. Click the **“Open in Scratch”** button
You should see new blocks from your project at the bottom of the list.



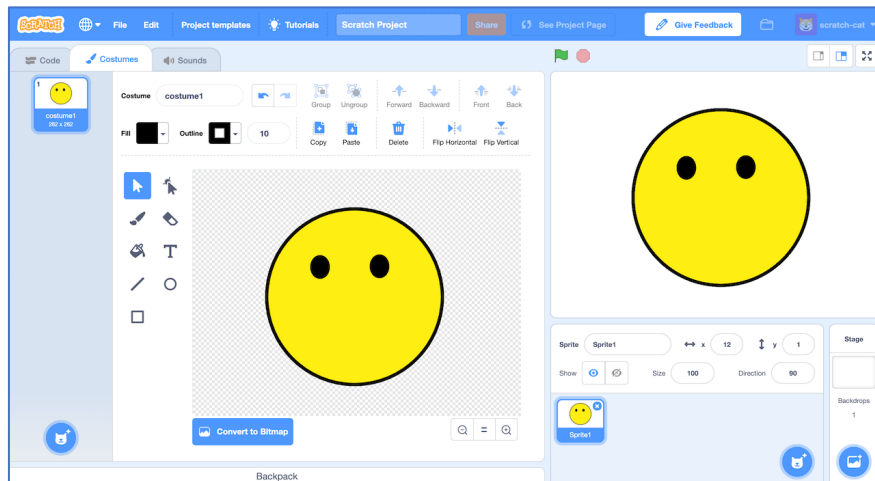
21. Delete the cat sprite



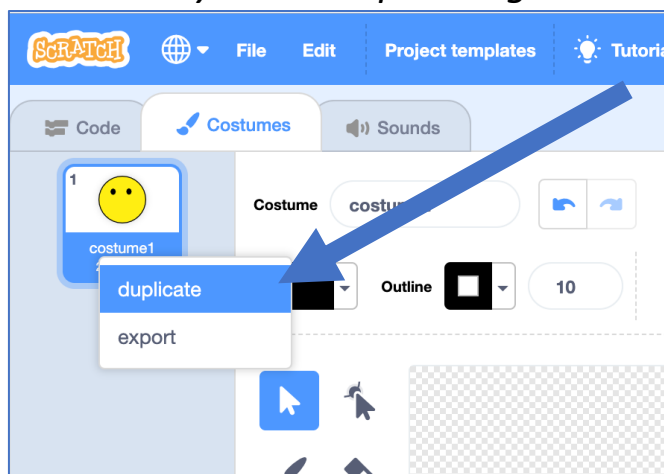
22. Create a new sprite by clicking the **Paint** icon



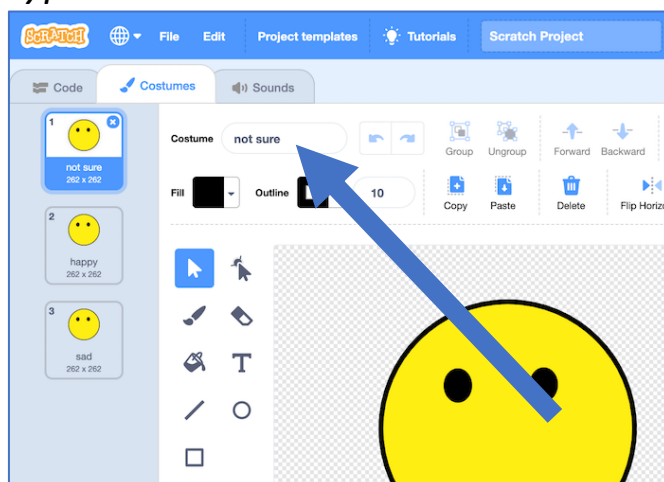
23. Draw a face without a mouth in the **Costumes** tab.



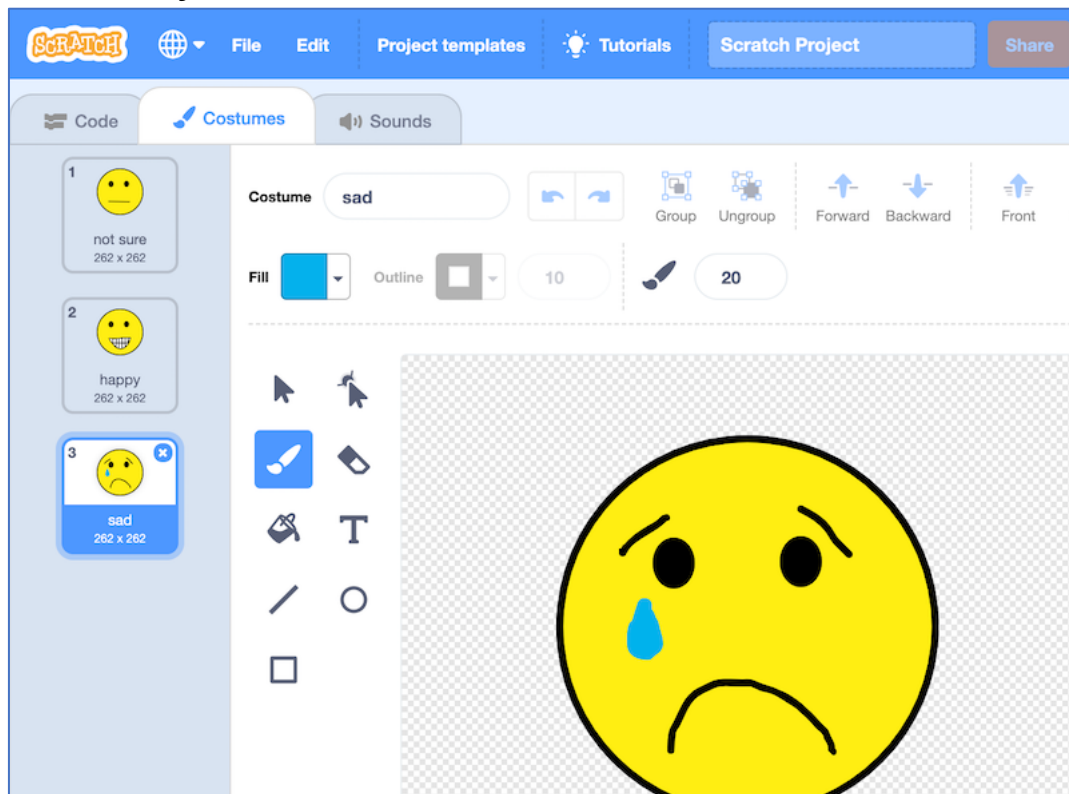
24. Right-click on the costume and click **“duplicate”**.
Do that again so you have **three** copies of the costume.
*Make sure you are duplicating the costume, **not** the sprite.*



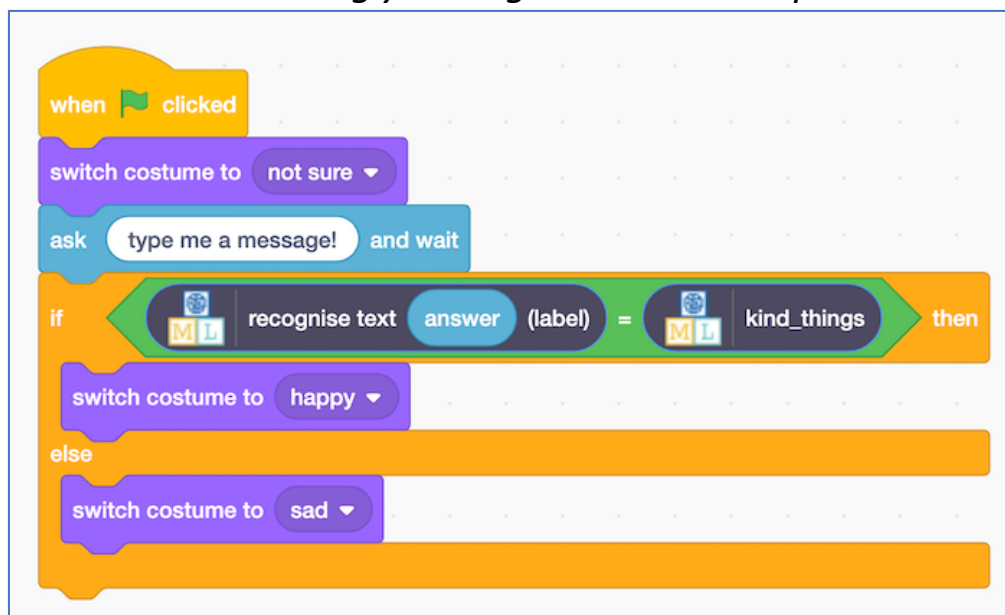
25. Name the three costumes “not sure”, “happy” and “sad”
Type the names into the white box shown by the arrow below.



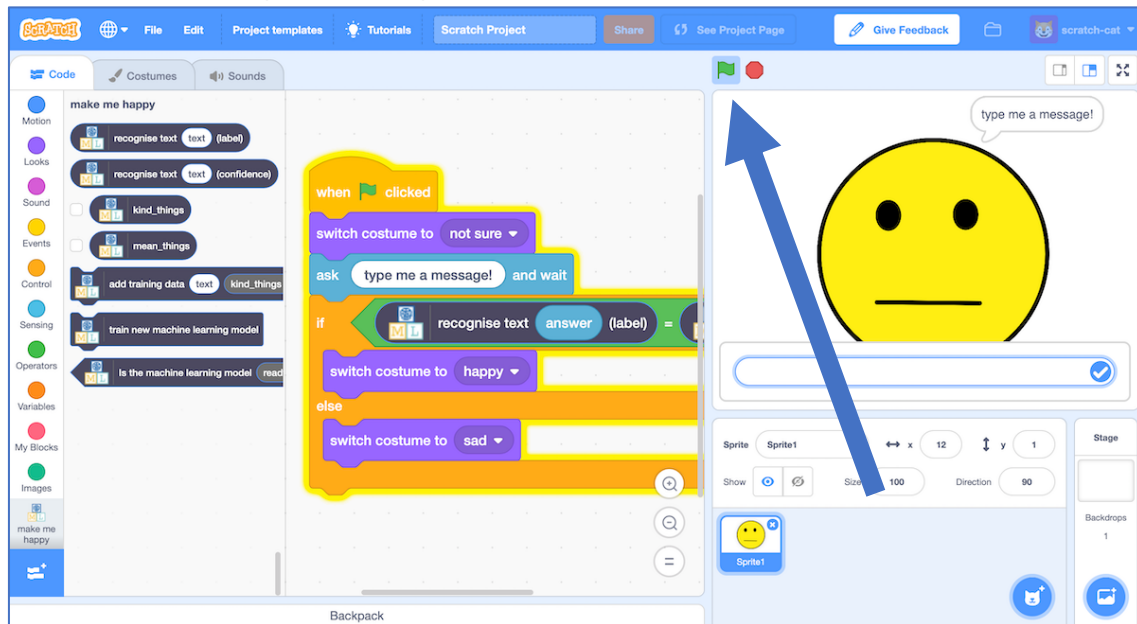
- 26.** Draw a mouth on each of the costumes.
The “not sure” face should be a straight line.
The “happy” face should have a smile.
The “sad” face should look sad.



- 27.** Click the “**Code**” tab, and enter the following script.
The “recognise text ... (label)” block is a new block added by your project.
If you give it text, it will recognise it as “kind things” or “mean things” based on the training you’ve given to the computer.



28. Click the **green flag** to test.



29. Type in a message and watch it react!

Type a kind message and press enter. The character should smile.

Click the green flag again. Type a mean and unkind message and press enter. The character should look sad.

This should work for messages that you didn't include in your training.

What have you done?

You've started to train a computer to recognise text as being kind or mean. Instead of trying to write rules to be able to do this, you are doing it by collecting examples. These examples are being used to train a machine learning “model”.

This is called “supervised learning” because of the way you are supervising the computer's training.

The computer will learn from patterns in the examples you've given it, such as the choice of words, and the way sentences are structured. These will be used to be able to recognise new messages.

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?

Write a reply

Instead of just changing the way they look, make your character reply, based on what it recognises in the message!

Try a different character

Instead of a person's face, why not try something different, like an animal?

For example, you could make a dog that wags their tail if you say something kind to it!

Use the confidence score

The other new Scratch block from your project gives the score for how sure your machine learning model is that it has recognised the message. How can you use this for messages that aren't compliments or insults?

Real world sentiment analysis

Can you think of examples where it's useful to be able to train a computer to recognise the emotion in writing?