

## Did you like it?

Create a machine learning model that will determine whether you liked a movie or book based on your comments.

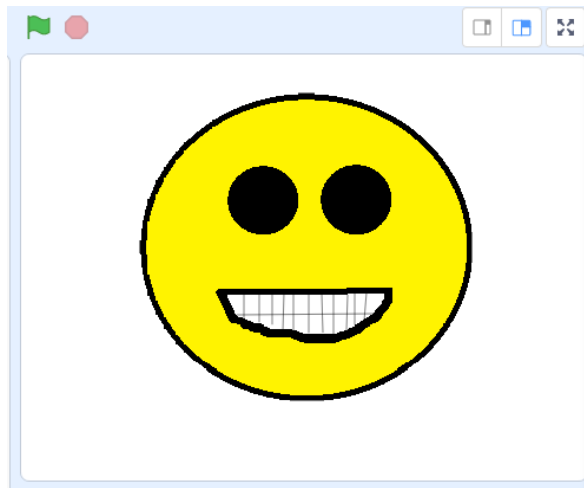


### Step 1 Introduction

In this project, you will use [machinelearningforkids.co.uk](https://machinelearningforkids.co.uk) ([machinelearningforkids.co.uk](https://machinelearningforkids.co.uk)) to make a character that will determine whether you liked or disliked a movie or book based on what you say. If you say something positive, it will smile. If you say something negative, it will cry.

First, you will program a list of rules for understanding messages, and learn why that approach isn't very good. Next, you will teach the computer to recognise messages as positive or negative by giving it examples of each.

#### What you will make



#### What you will learn

- How to train and test a machine learning model
- Why this approach is better than using a long list of rules
- How to use a trained model in a Scratch 3 program

#### What you will need

- A computer connected to the internet



## Additional information for educators

If you need to print this project, please use the **printer-friendly version** (<https://projects.raspberrypi.org/en/projects/did-you-like-it/print>).

## Licence

This project is dual-licensed under both a **Creative Commons Attribution Non-Commercial Share-Alike License** (<http://creativecommons.org/licenses/by-nc-sa/4.0/>) and an **Apache License Version 2.0** (<http://www.apache.org/licenses/LICENSE-2.0>).

We'd like to thank Dale from [machinelearningforkids.co.uk](http://machinelearningforkids.co.uk) for all his work on this project.

## Step 2 Create a project and sprite costumes

- Go to **machinelearningforkids.co.uk** (<https://machinelearningforkids.co.uk/>) in a web browser.
- Click on **Get Started**.
- Click on **Try it now**.



- Click the **+ Add a new project** button.
- Name your project **Did you like it** and set it to learn to recognise **text**. Click on **Create**.

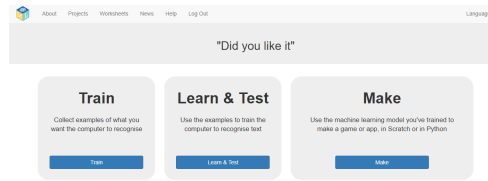
- You should now see **Did you like it** in the projects list. Click on this project.



- Now, get a project ready in Scratch.



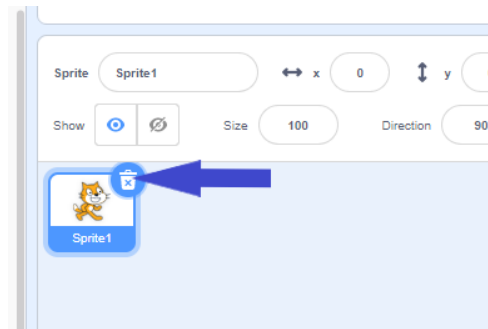
- Click on **Make**.



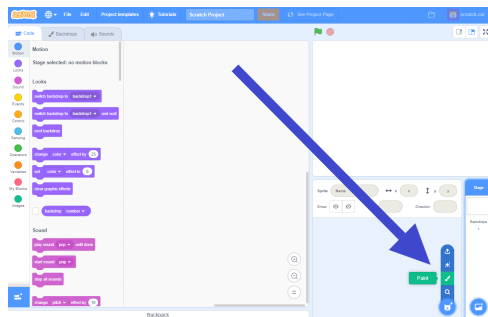
- Click on **Scratch 3**.

- The page then warns you that you haven't done any machine learning yet. Click on **Scratch by itself** to launch Scratch.

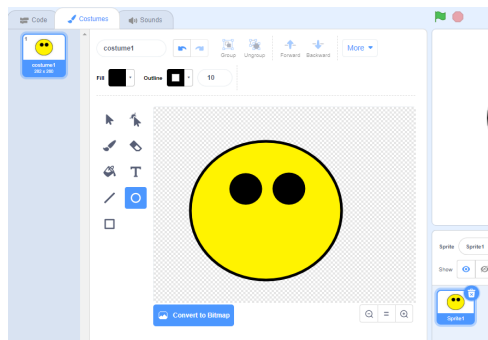
- Delete the cat sprite.



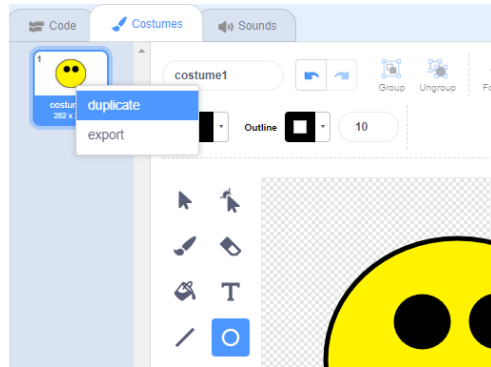
- Go to the new sprite menu and click on the **Paint** icon to create a new sprite.



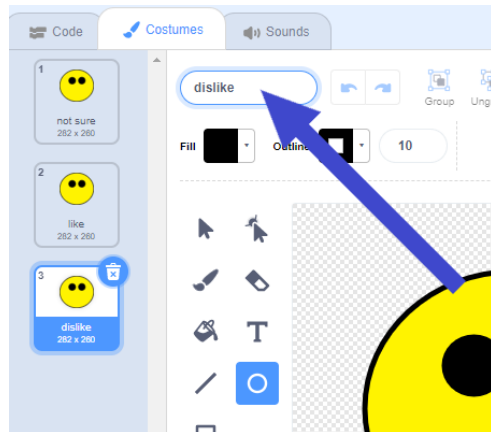
- Draw a face without a mouth.



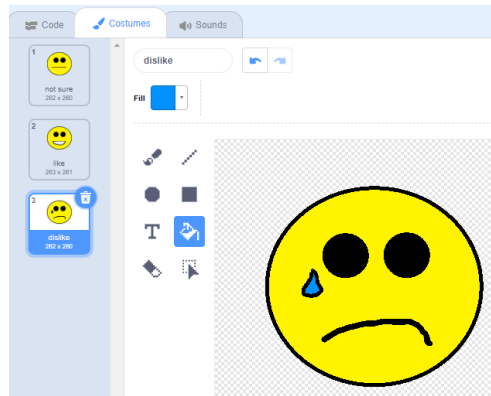
- Right-click on the costume and click on **duplicate**. Repeat one more time so that you have **three** copies of the costume.



- Name the three costumes **not sure**, **like** and **dislike**. Type the names into the white box shown by the arrow below.



- Draw a mouth on each of the costumes. The **not sure** face should have a straight line as its mouth. The **like** face should have a smile. The **dislike** face should look sad.



### Step 3 Add a list of rules

In this step, you will include a list of rules to change the costume to **like** or **dislike**.

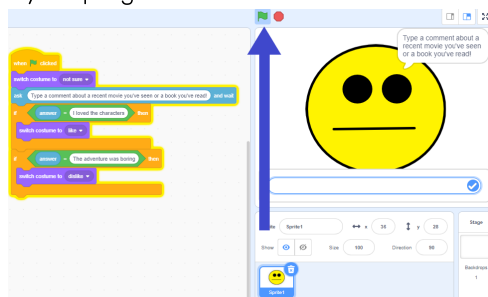
- Click on the **Code** tab and enter the following code.



```
when green flag clicked
  switch costume to not sure
  ask "Type a comment about a recent movie you've seen or book you've read!" and wait
  if answer = "I loved the characters" then
    switch costume to like
  if answer = "The adventure was boring" then
    switch costume to dislike
```

- Click on **File** and then on **Save to your computer** to save the program to a file.

- Click on the **green flag** to test your program.



- Type in a comment about a movie or book and watch it react! Type in **I loved the characters** and press **Enter**. The character will smile. Click on the green flag again, type in **The adventure was boring**, and press **Enter**. The character will cry. Type in anything else, press **Enter**, and the character's face won't change.

You have created a character that should react to what people type in, and programmed it using a simple rules-based approach.

If you want it to react to other messages, you would need to add more `if` blocks.

The problem with this is that you would need to predict exactly what messages the character will receive – it would take forever to make a list of every possible message!

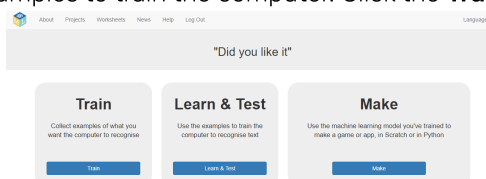
## Step 4 Collect examples for training

Next, try a better approach: teaching the computer to recognise messages for itself.

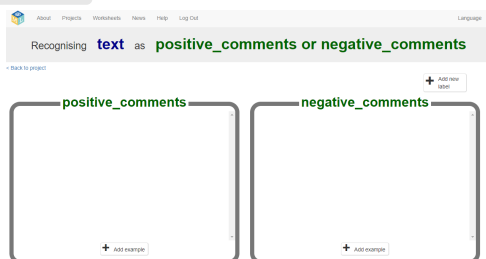
- Close the Scratch window and go back to the Machine Learning for Kids website.
- Click on **< Back to project**.



- You need to collect some examples to train the computer. Click the **Train** button.



- Click on **+ Add new label** and call it **positive comments**. Do that again, and create a second bucket called **negative comments**.



- Click on the **Add example** button in the **positive comments** bucket, and type in the nicest comment about a movie or book that you can think of.
- Click on the **Add example** button in the **negative comments** bucket, and type in the meanest comment about a movie or book that you can think of.



- Continue to **Add examples** until you have got at least **six** compliments and **six** insults.



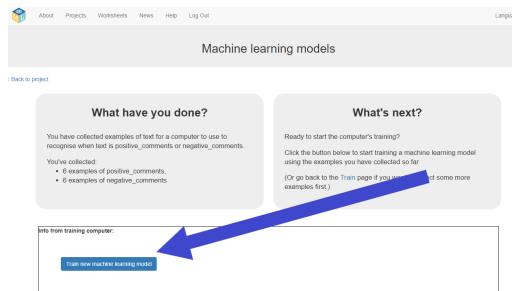
### Tips for creating good examples

- **More is good:** The more examples you give your program, the better the program should get at determining whether your comments are positive or negative.
- **Equal numbers:** Add roughly the same number of examples for each type of comment. If you have a lot of examples for one type and not the other, this can affect the way that the program learns to recognise them.
- **Make the examples really different from each other:** Try to think of lots of different types of example. For example, make sure that you include some long examples and some very short ones.

## Step 5 Train and test your machine learning model

In this step, you will train your machine to recognise whether your comment is positive or negative and automatically put it in one of the two buckets based on the examples that you have added.

- Click on **< Back to project**, then click on **Learn & Test**.
- Click on the **Train new machine learning model** button. If you have enough examples, the program should start to learn how to recognise comments as either positive or negative from the examples that you've given to it.



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

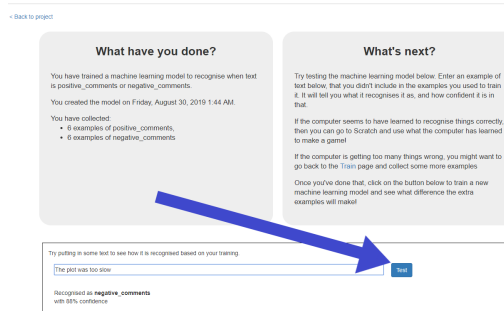
Once the training has completed, a test box will be displayed. Try testing your machine learning model to see what it has learned.



- Type something nice, and press **Enter**. It should be recognised as positive.
- Type something critical, and press **Enter**. It should be recognised as negative.
- Test it with examples that you haven't shown the computer before.

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

Make sure that you repeat these steps to train your computer with the new examples though!



You have started to train a computer to recognise text as being positive or negative. 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**'.

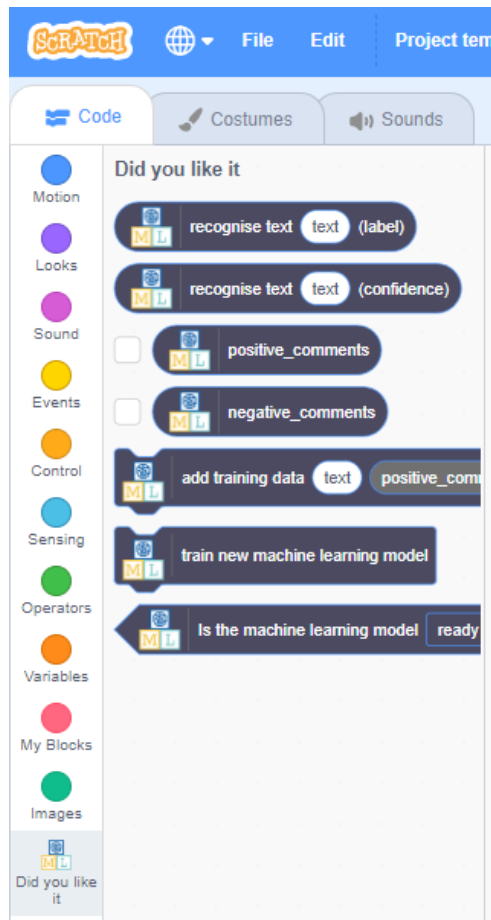
This is called **supervised learning** because of the way that you are supervising the computer's training.

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

## Step 6 Use your machine learning model in Scratch

Now, update your Scratch program to include your machine learning model instead of a rules-based approach.

- Click on < **Back to project**.
- Click on **Make**.
- Click on **Scratch 3**.
- Click the **Open in Scratch 3** button to launch the Scratch editor. You should see new blocks from your project at the bottom of the list.

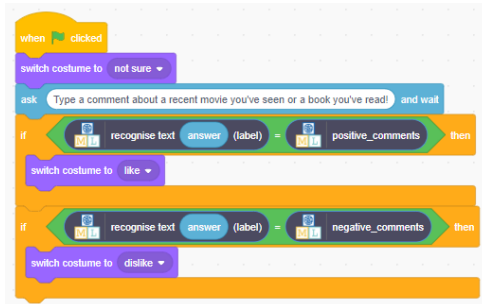


- Load the Scratch project that you saved earlier: click on **File**, then on **Load from your computer**, then select the Scratch project that you saved earlier.

- Click on the **Code** tab, then update your Scratch code to use your machine learning model **instead** of the rules that you added earlier. ✔

The **recognise text ... (label)** block is a new block added by your project. If you give it text, it will return either **positive comments** or **negative comments** based on the training that you've given to the computer.

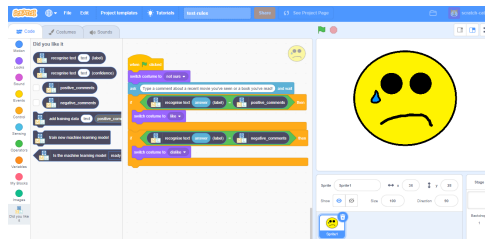
You can use this to choose the costume to switch to.



- Click on the **green flag** to test your project again. ✔



- Type in a nice message and press **Enter**. The character should smile. Click on the **green flag** again. Type in a negative message and press **Enter**. The character should look sad. Make sure that you test that this works **even for messages that you didn't include in your training**.



- Save your project: click on **File**, then on **Save to your computer**.

You have now modified your Scratch character to use machine learning instead of your earlier rules-based approach. Training the computer to be able to recognise messages for itself should be much quicker than trying to make a list of every possible message. The more examples you give it, the better it should get at recognising messages correctly.



## Challenge!

### Challenge: more characters and emotions

**Write a reply:** Instead of just changing the way that your character looks, make it 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? The character could react in different ways, instead of smiling. For example, you could make a dog that wags their tail if you say something nice about the movie or book!



**Different emotions:** Instead of positive and negative, could you train the character to recognise other types of messages?



**Real-world sentiment analysis:** Can you think of examples where it's useful to be able to train a computer to recognise emotion in writing?



## Step 7 What next?

---

If you enjoyed this project, why not try some of our other **machine learning projects** (<https://projects.raspberrypi.org/en/pathways/scratch-machine-learning>)?

---

Published by **Raspberry Pi Foundation** (<https://www.raspberrypi.org>) under a **Creative Commons** license (<https://creativecommons.org/licenses/by-sa/4.0/>).

**View project & license on GitHub** (<https://github.com/RaspberryPiLearning/did-you-like-it>)