



Shoebox

In this project you recreate the world's first speech-recognition system

It was called "Shoebox" and created in 1961. It was a voice-controlled calculator and could recognize and answer spoken mathematical sums.

You will learn a little about how it was made by recreating a simple "Shoebox" for yourself in Scratch.

The screenshot shows the Scratch workspace with the script for the 'shoebox' sprite. The script consists of the following blocks:

- When green flag clicked:
 - Start listening
 - Perform sum
 - Stop listening
- When I hear 0:
 - Add 0 to current value
- When I hear 1:
 - Add 1 to current value
- When I hear 2:
 - Add 2 to current value
- When I hear 3:
 - Add 3 to current value
- When I hear 4:
 - Add 4 to current value
- When I hear 5:
 - Add 5 to current value
- When I hear 6:
 - Add 6 to current value
- When I hear 7:
 - Add 7 to current value
- When I hear plus:
 - Add operation plus
- When I hear minus:
 - Add operation minus
- When I hear total:
 - Stop listening

The stage background features a black and white photograph of a man in a suit operating a complex electronic device with many knobs and switches. A digital display on the machine shows the number 68. To the right of the stage, there is a data table labeled 'input' with rows for numbers 1 through 6 and operators plus and minus, and a 'result' row showing 68. The 'length' of the input sequence is also indicated as 6.



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://ibm.biz/shoebox-demo> to watch a demo of what you will be recreating.

This two-minute video was recorded in 1961 and shows you the inventor (William Dersch) speaking math sums into a microphone. A computer recognises what he is saying and gives him the result of the sum.



2. If you have time, go to <https://ibm.biz/shoebox>

You can find a ten-minute video (also from the 1960's, but in colour!), explanations of the technology, lots of photos, and info about the history of speech-recognition.

A screenshot of a web page from the IBM Heritage section. The top navigation bar includes links for AI, Hybrid Cloud, Products, Consulting, Support, Think, IBM Heritage (which is underlined), IBMers, Inside IBM, and Advancing Humanity. Below the navigation, a breadcrumb trail shows Home / history / Speech recognition. The main content area has a title "Speech recognition" and a subtitle: "The world's first speech-recognition system was the size of a shoebox". To the right of the text is a black and white photograph of a person's hands working on a complex electronic device with many wires, knobs, and a small display screen. The device is housed in a wooden box.

- 3.** Go to <https://machinelearningforkids.co.uk/> in a web browser

- 4.** Click on “Get started”

- 5.** Click on “Log In” and type in your username and password

- 6.** Click on “Projects” on the top menu bar

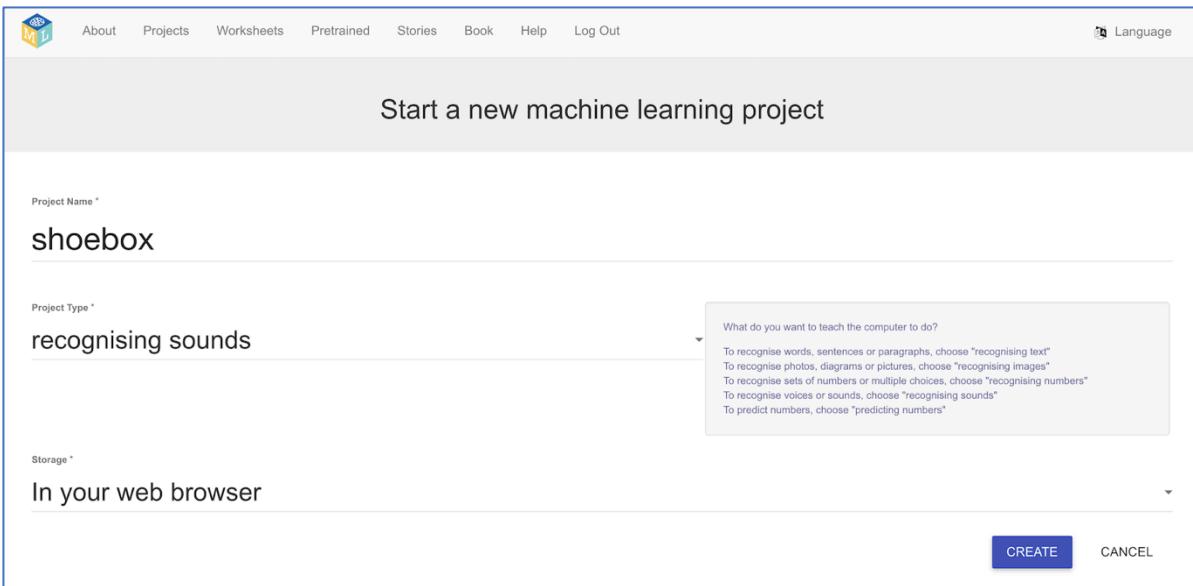
- 7.** Click the “+ Add a new project” button.

- 8.** Name your project “shoebox”

Set it to learn how to recognise “sounds”.

Choose to store the training examples “in your web browser”.

Click **Create**



Start a new machine learning project

Project Name *

shoebox

Project Type *

recognising sounds

What do you want to teach the computer to do?

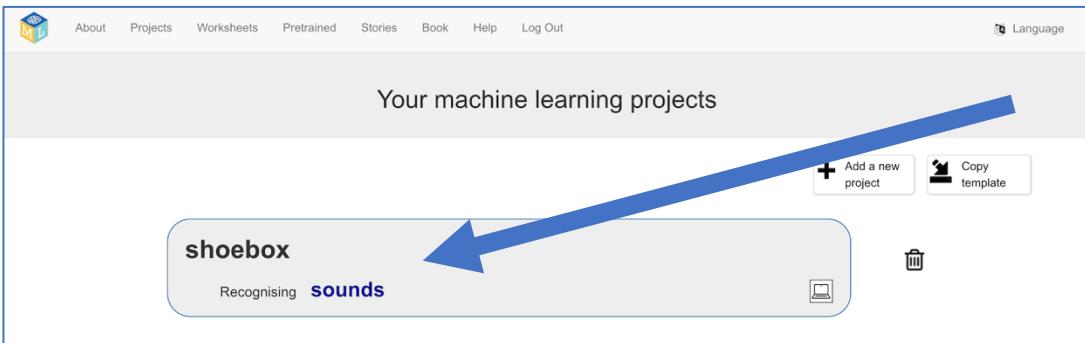
- To recognise words, sentences or paragraphs, choose "recognising text"
- To recognise photos, diagrams or pictures, choose "recognising images"
- To recognise sets of numbers or multiple choices, choose "recognising numbers"
- To recognise voices or sounds, choose "recognising sounds"
- To predict numbers, choose "predicting numbers"

Storage *

In your web browser

CREATE **CANCEL**

- 9.** You should see “shoebox” in your projects list. Click it.



Your machine learning projects

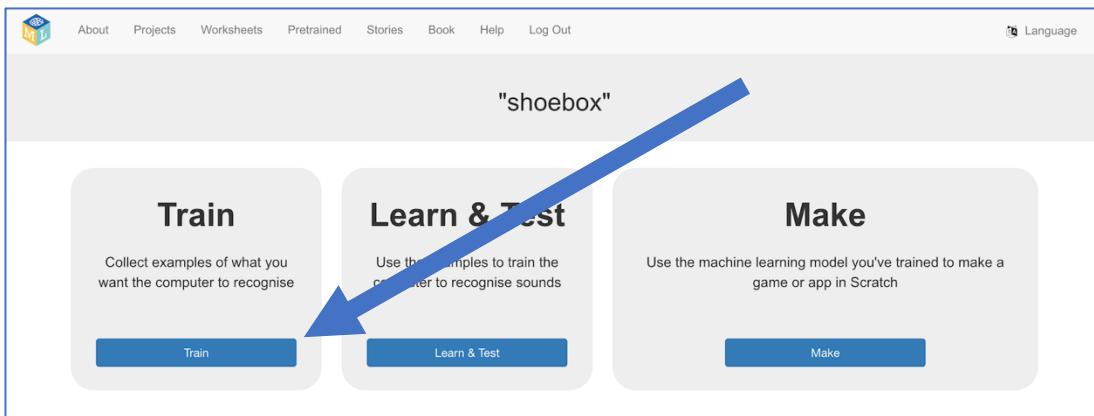
+ Add a new project Copy template

shoebox

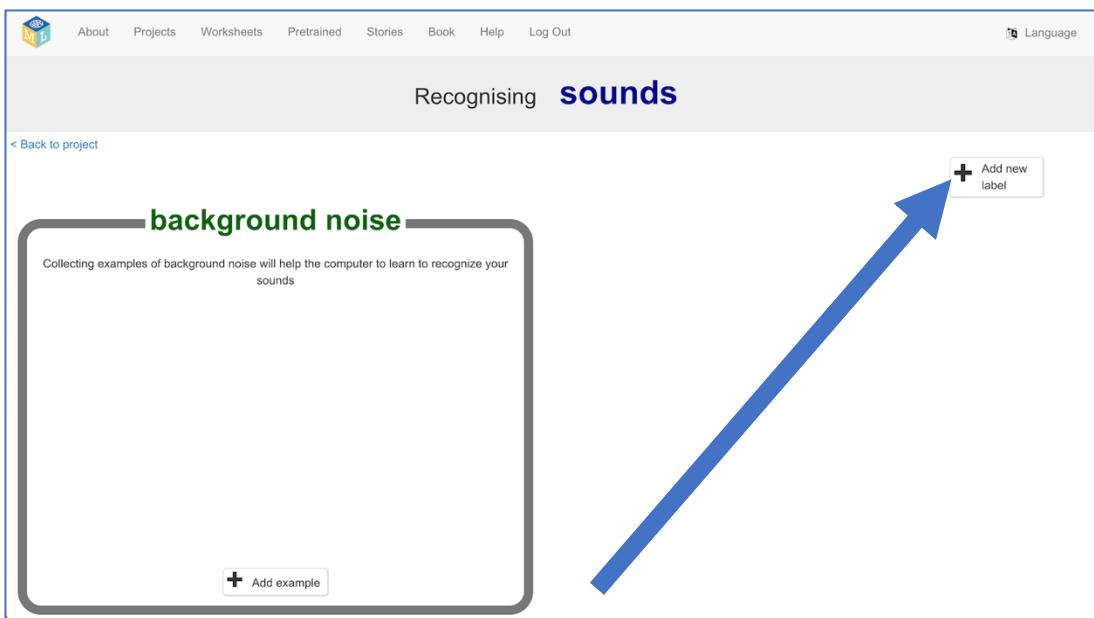
Recognising sounds

trash icon

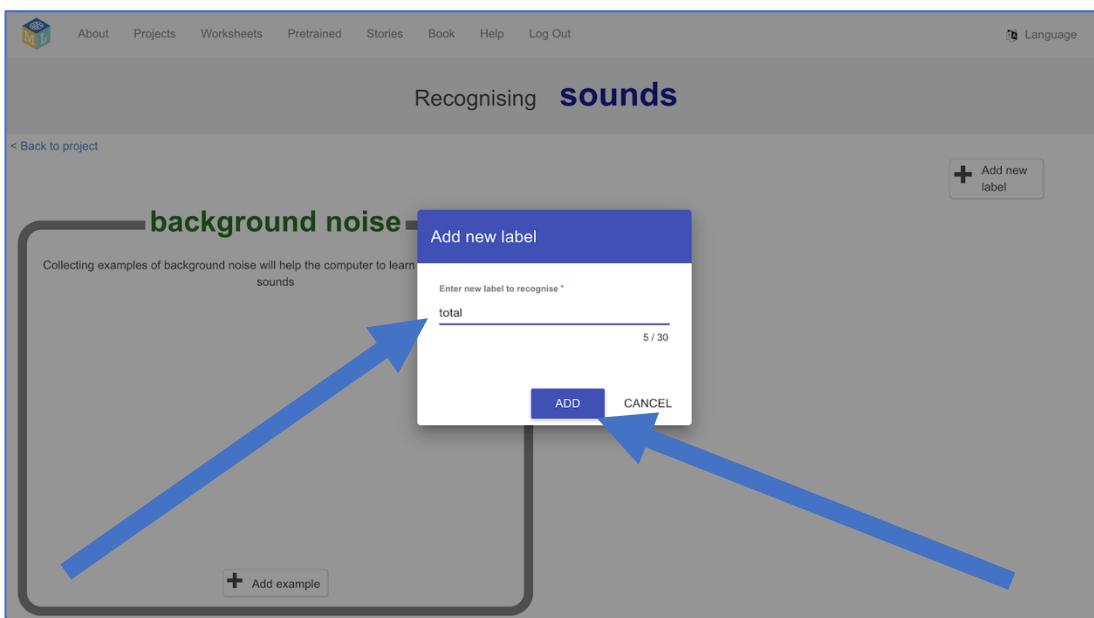
10. Click “Train”



11. Click “Add new label”



12. Create a label called “total” and click “ADD”



13. Repeat that to create labels for “plus” and “minus”

The screenshot shows a web-based application for audio recognition. At the top, there's a navigation bar with links for About, Projects, Worksheets, Pretrained, Stories, Book, Help, and Log Out. On the right side of the header is a Language selection icon. Below the header, the title "Recognising sounds as total, plus or minus" is displayed. Underneath the title, there's a link "[Back to project](#)". On the right side of the main area, there's a button "+ Add new label". The main content area contains four categories: "background noise" (green text), "total" (black text), "plus" (green text), and "minus" (green text). Each category has a descriptive text box below it and a button "+ Add example" at the bottom. The "background noise" category also includes a note: "Collecting examples of background noise will help the computer to learn to recognize your sounds".

14. Repeat that to create labels for numbers from 0 to 9

The screenshot shows a web-based application for audio recognition, similar to the previous one but with more labels. At the top, there's a "background noise" category (green text) with a note: "Collecting examples of background noise will help the computer to learn to recognize your sounds" and a "+ Add example" button. To the right are three empty boxes labeled "total", "plus", and "minus" (all green text). Below these are rows of numbered labels: "0", "1", "2", "3", "4", "5", "6", "7", "8", and "9" (all green text). Each number label has an associated empty box to its right and a "+ Add example" button at the bottom. A top-right corner button "+ Add new label" is also visible.

You need to record **eight** examples of yourself saying each of these words.

Recording examples for thirteen words (“total”, “plus”, “minus”, “0”, “1”, “2”, “3”, “4”, “5”, “6”, “7”, “8”, “9”) is a lot of work! If you don’t have time to do all of these it is okay to create labels for only a few numbers instead.

Your calculator will recognise the numbers you create labels for.

The screenshot shows a web-based application for sound recognition. At the top, there's a navigation bar with links like 'About', 'Projects', 'Worksheets', 'Pretrained', 'Stories', 'Book', 'Help', 'Log Out', and a 'Language' dropdown. Below the navigation, the title 'Recognising sounds as total, plus or 5 other classes' is displayed. The main area contains a grid of buckets for collecting sound examples. The first row has three buckets: 'background noise' (with a note: 'Collecting examples of background noise will help the computer to learn to recognize your sounds'), 'total', and 'plus'. Each bucket has a '+ Add example' button. The second row has three buckets: 'minus', '0', and '2'. The third row has two buckets: '5' and '7'. The fourth row has two buckets: '8' and '9'. There's also a small 'Add new label' button in the top right corner.

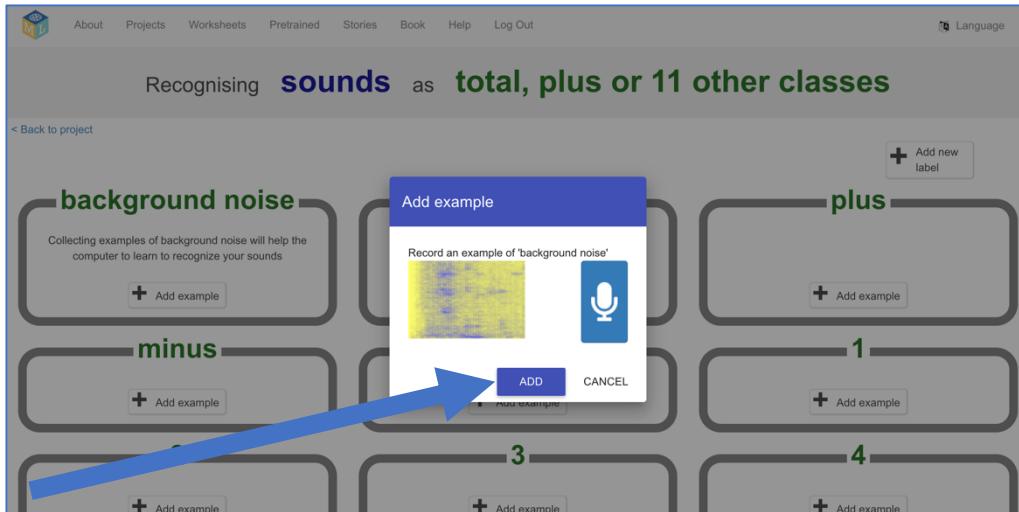
15. Click the “Add example” button in the “background noise” bucket

This screenshot is similar to the one above, but it includes a large blue arrow pointing from the bottom-left towards the '+ Add example' button inside the 'background noise' bucket. This indicates where the user should click to add a new example of background noise.

16. Click the microphone button to record an example of the background noise where you are. *Don't say anything while it records!*

This screenshot shows the 'Add example' dialog box open over the SoundKit interface. The dialog has a blue header 'Add example' and a note 'Record an example of 'background noise''. It features a large blue microphone icon in the center. At the bottom, there are 'ADD' and 'CANCEL' buttons. A blue arrow points from the bottom-left towards the microphone icon, guiding the user to click it to start recording.

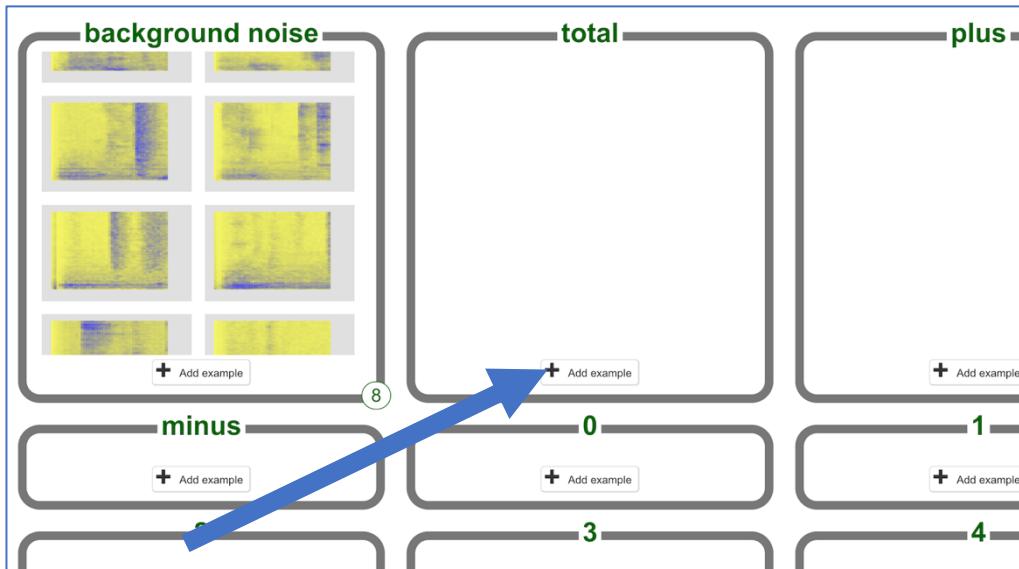
17. Click on “ADD” to save your recording



18. Record **eight** examples of the background noise where you are



19. Click “Add example” for the “total” training bucket



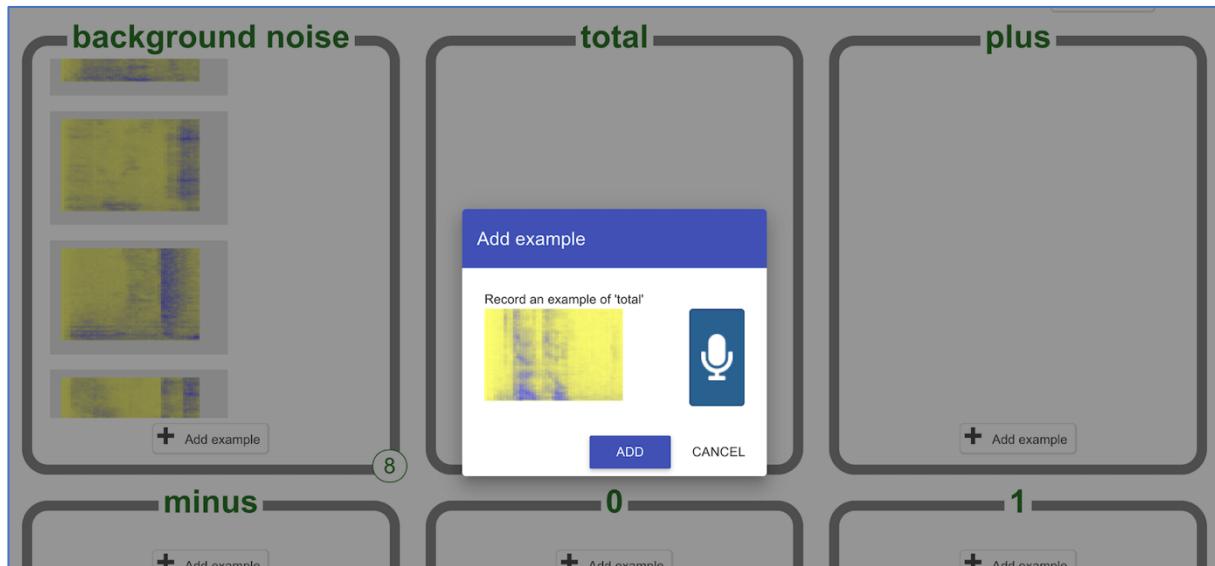
20. Record an example of yourself saying the word “total”
Click the microphone button.

Finish saying the word before the progress bar finishes.

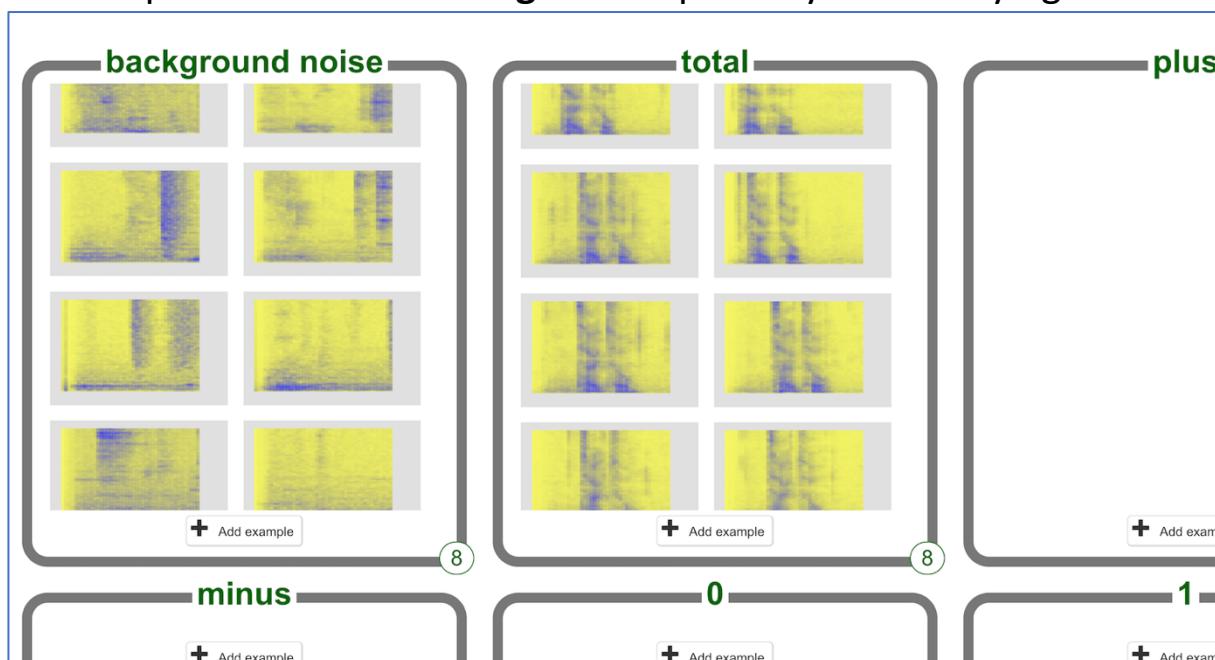
You will see a visual representation of the sound you make.

In my screenshot, you can see the peaks from the two “T” sounds I made as I said “total”.

Yours may look a bit different – that is okay.



21. Repeat that to record **eight** examples of yourself saying “total”

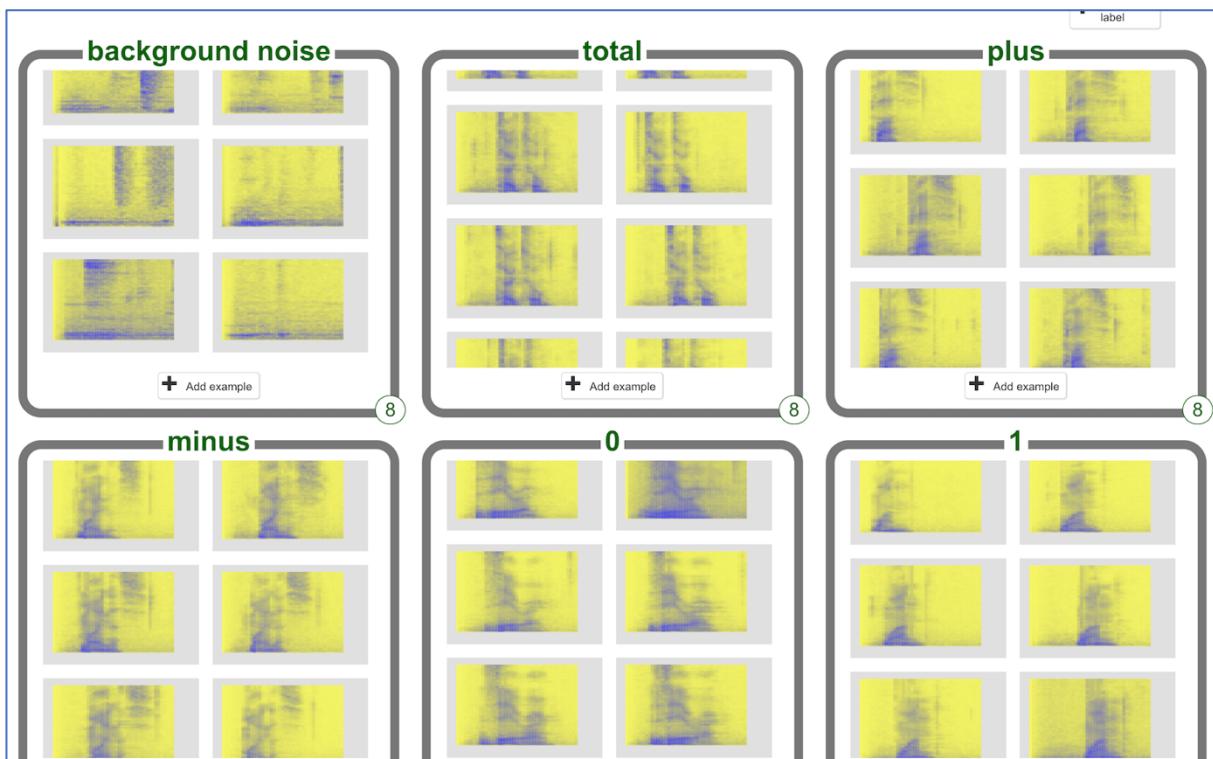


22. Record **eight** examples for all words your calculator needs to recognise.

These visualisations show what the computer will learn from to recognise your words.

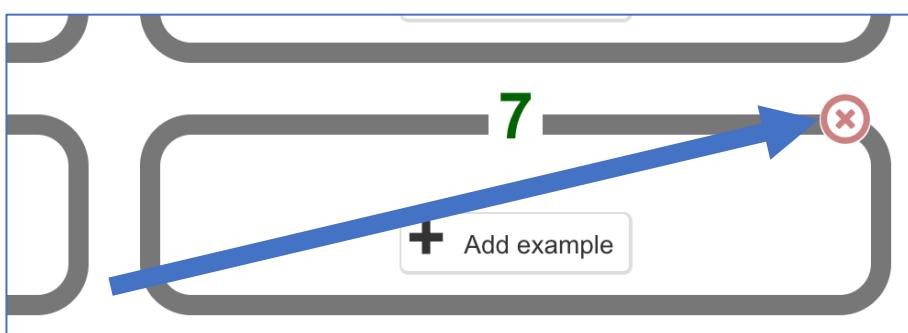
Do you see anything in common in the images in each training bucket?

Do you think you could recognise a command from a picture like this?



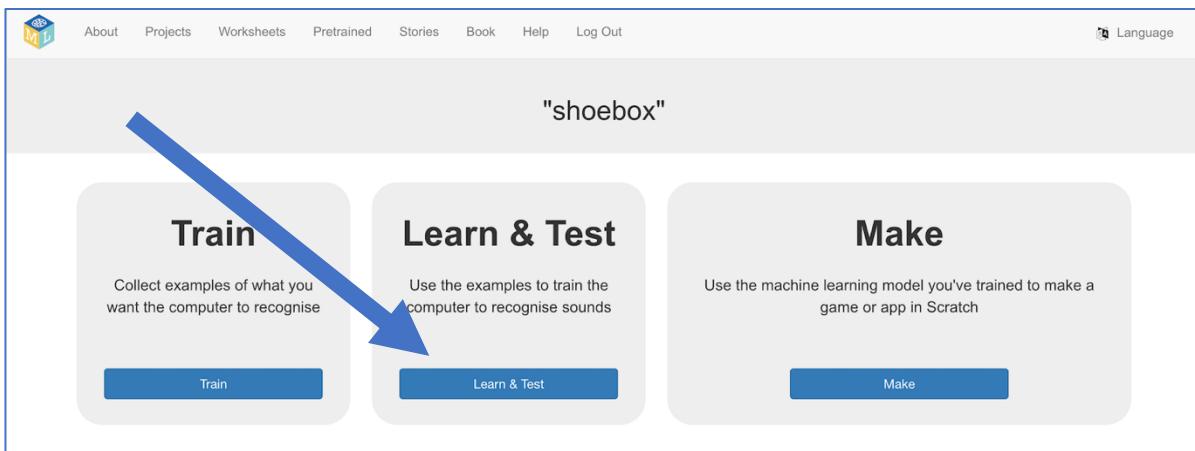
If this is taking too long, you can change your mind about how many words you want your calculator to learn to recognise.

Delete the training bucket for words you don't need the computer to recognise by clicking on the red cross in the corner.



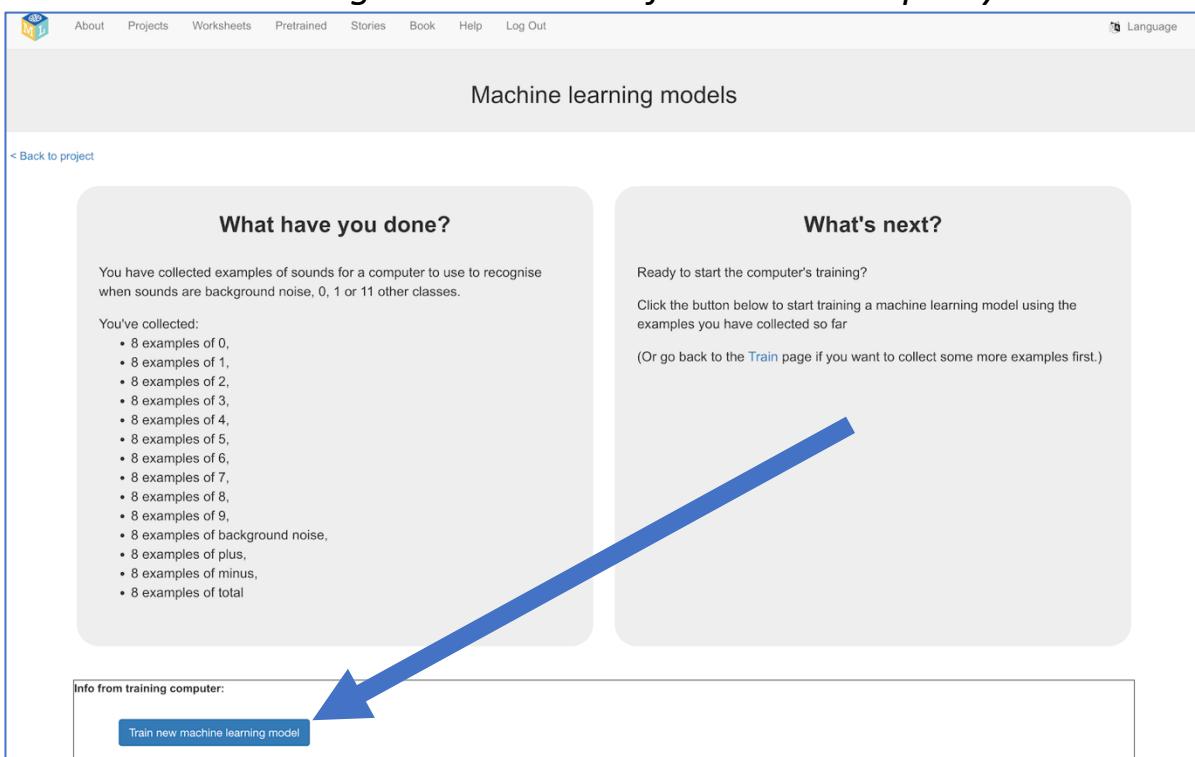
23. Once you have examples for every word, click “< Back to project”

24. Click “Learn & Test”



25. 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 commands from the examples you've recorded.



26. Wait for the training to complete.

While waiting, try to complete the machine-learning multi-choice quiz at the bottom of the page.

What have you done so far?

You've started to train a computer to recognise spoken words that will be used by your "Shoebox" calculator.

Instead of writing rules to do this, you are doing it by collecting examples. You've used these examples to train a machine learning "model".

The computer will learn from patterns in the recordings you've given it. These will be used to be able to recognise what it hears.

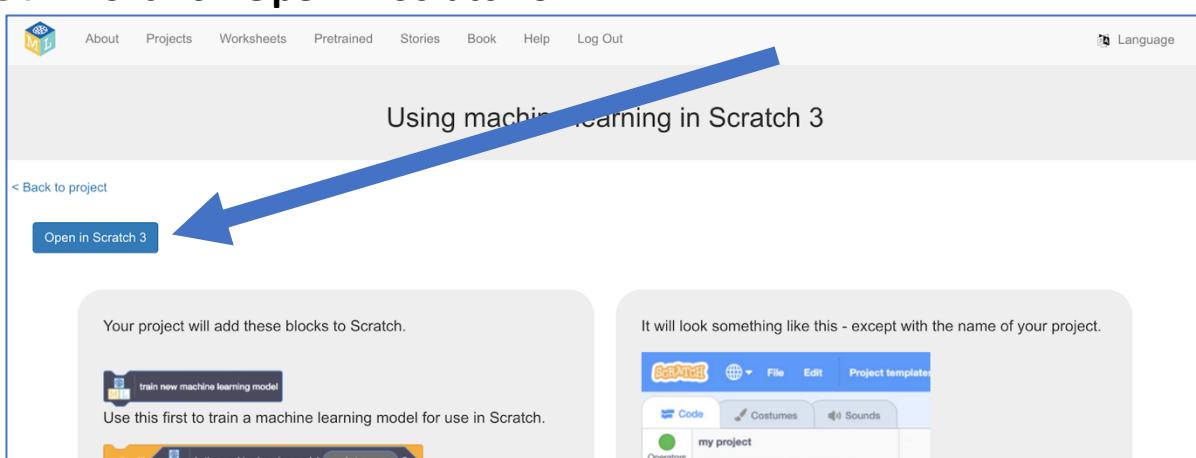
The coloured visualisations (called "spectrograms") give you an idea of the information that the computer is using to learn from.

27. Click on the "**< Back to project**" link

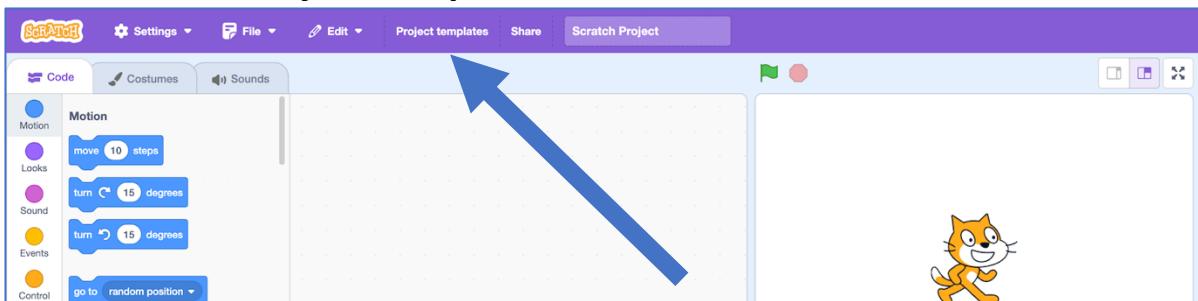
28. Click on **Make**

29. Click on **Scratch 3**

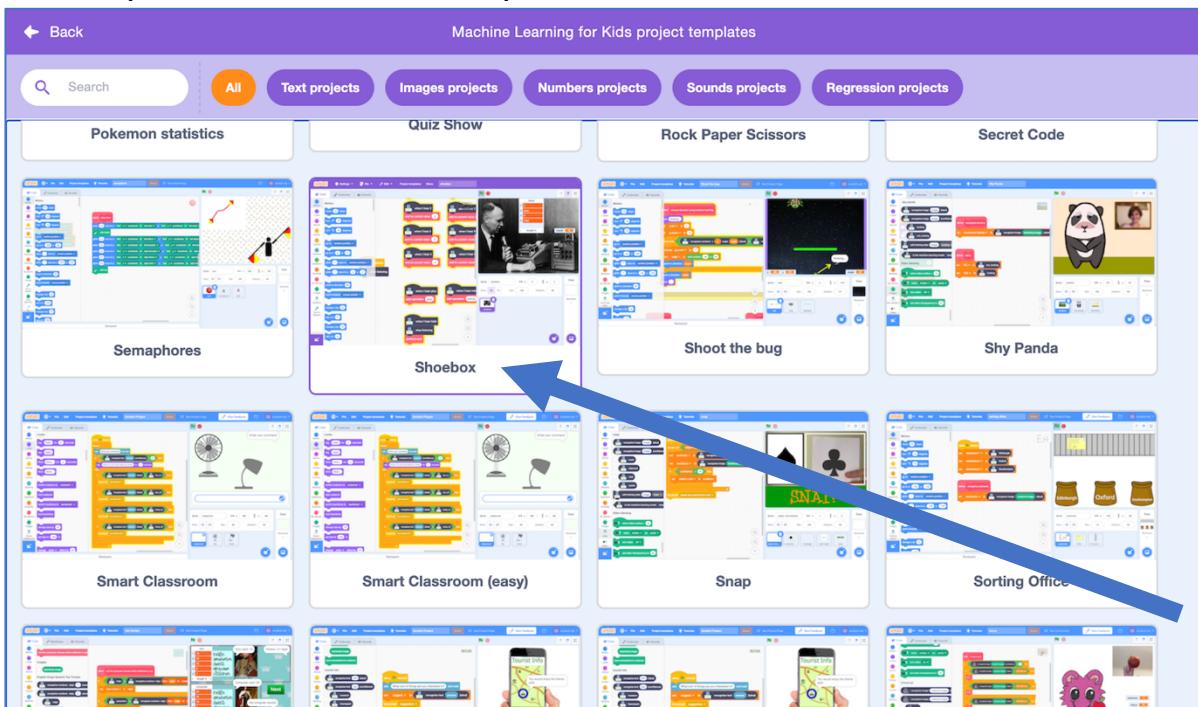
30. Click on **Open in Scratch 3**



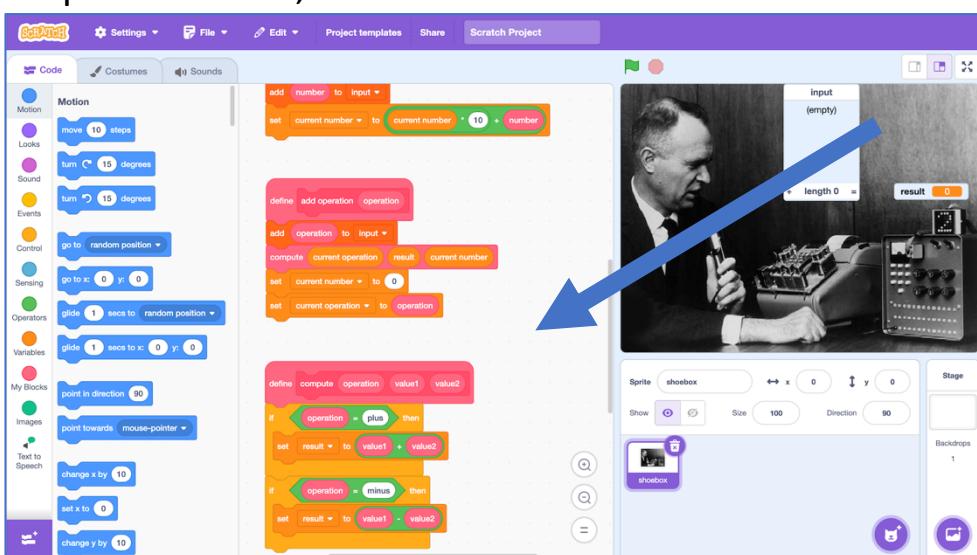
31. Click on Project templates



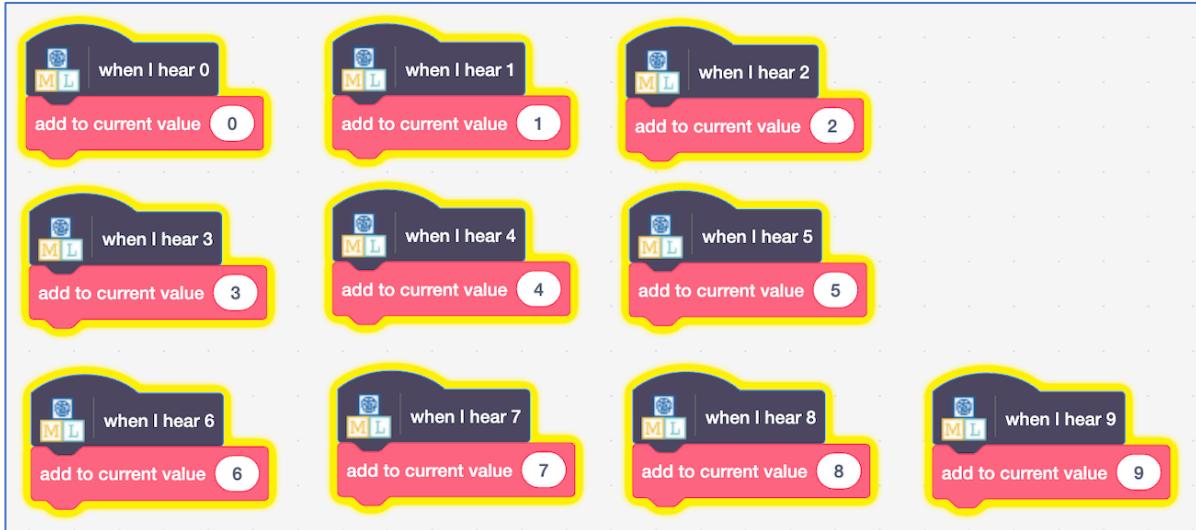
32. Open the Shoebox template



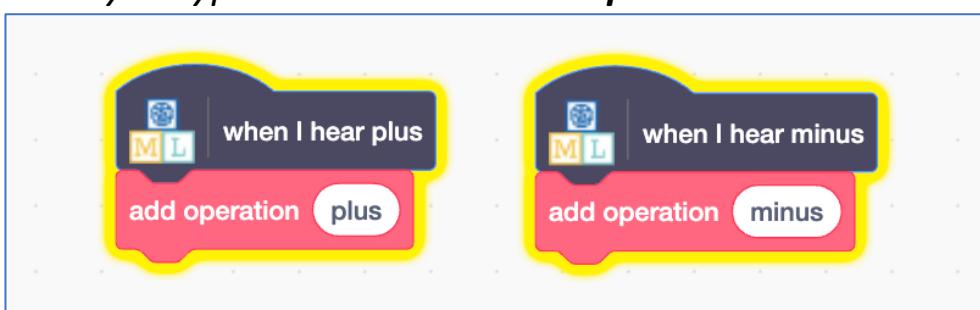
33. The template contains code blocks that implement the logic for a simple calculator, that can add and subtract numbers.



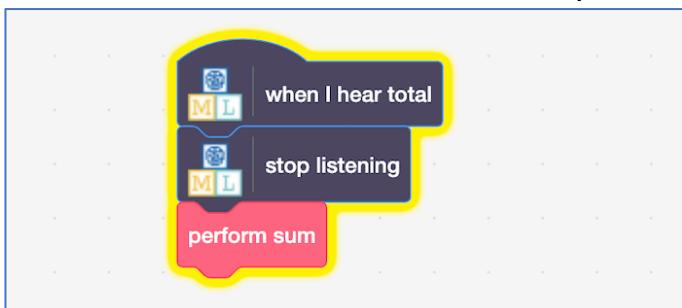
34. The first step is to pass numbers that your machine learning model recognises to the calculator. Create this code in the shoebox sprite.
If you only recorded examples of some numbers, you'll only have code blocks for those numbers. That is okay – but your calculator will only be able to use those numbers.



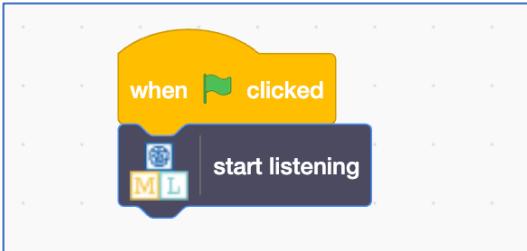
35. The next step is to recognise the “plus” and “minus” commands. Create this code in the shoebox sprite.
*Make sure that you spell “plus” and “minus” correctly (with no spaces) when you type them into the **add operation** blocks*



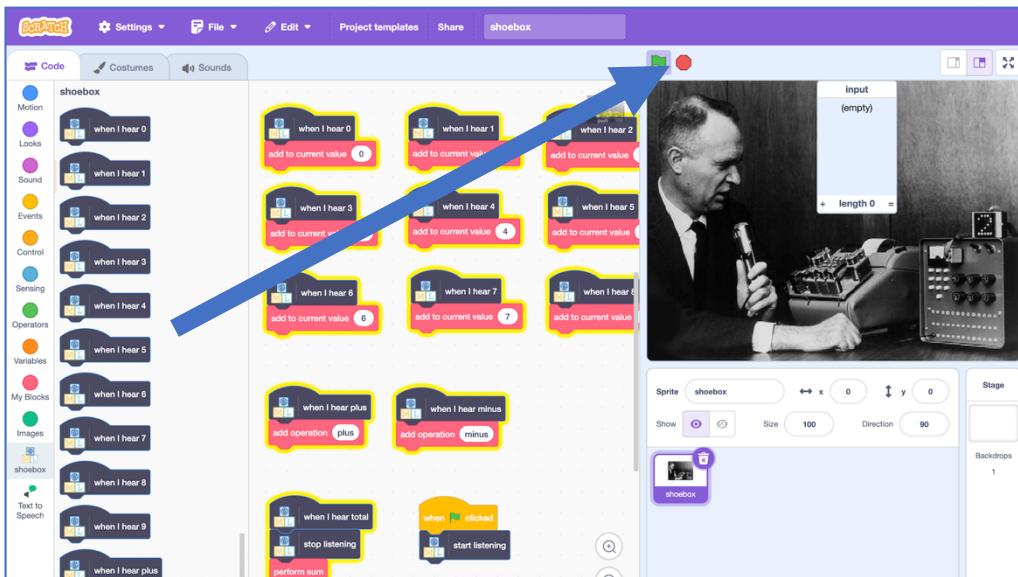
36. The next step is to recognise the “total” command that tells the calculator to give you the answer.
Create this code in the shoebox sprite.



37. The final step is to get the calculator to start listening to you.



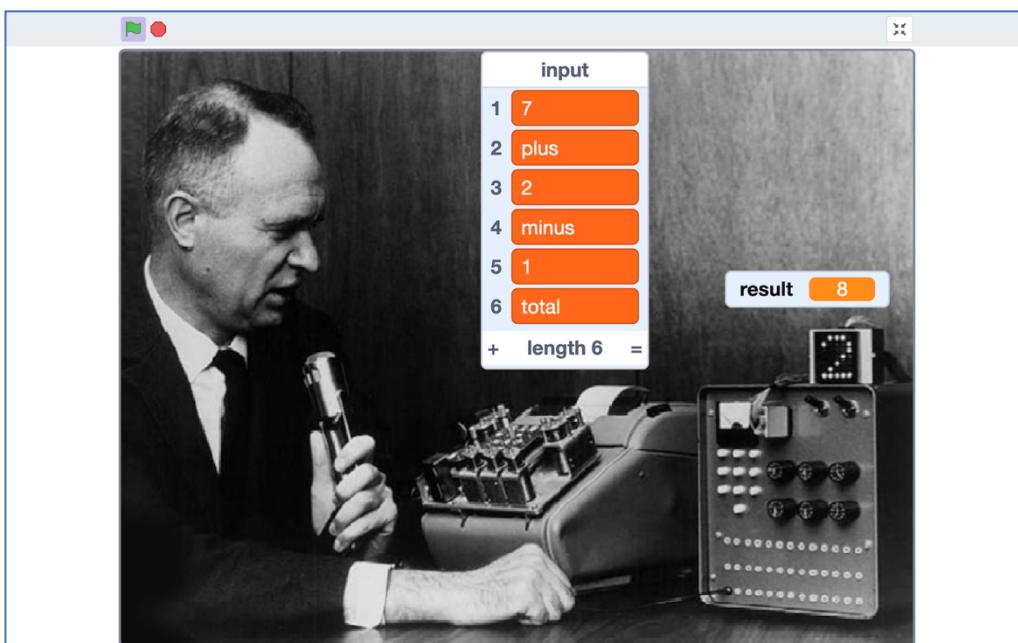
38. Time to try out your calculator. Click the Green Flag



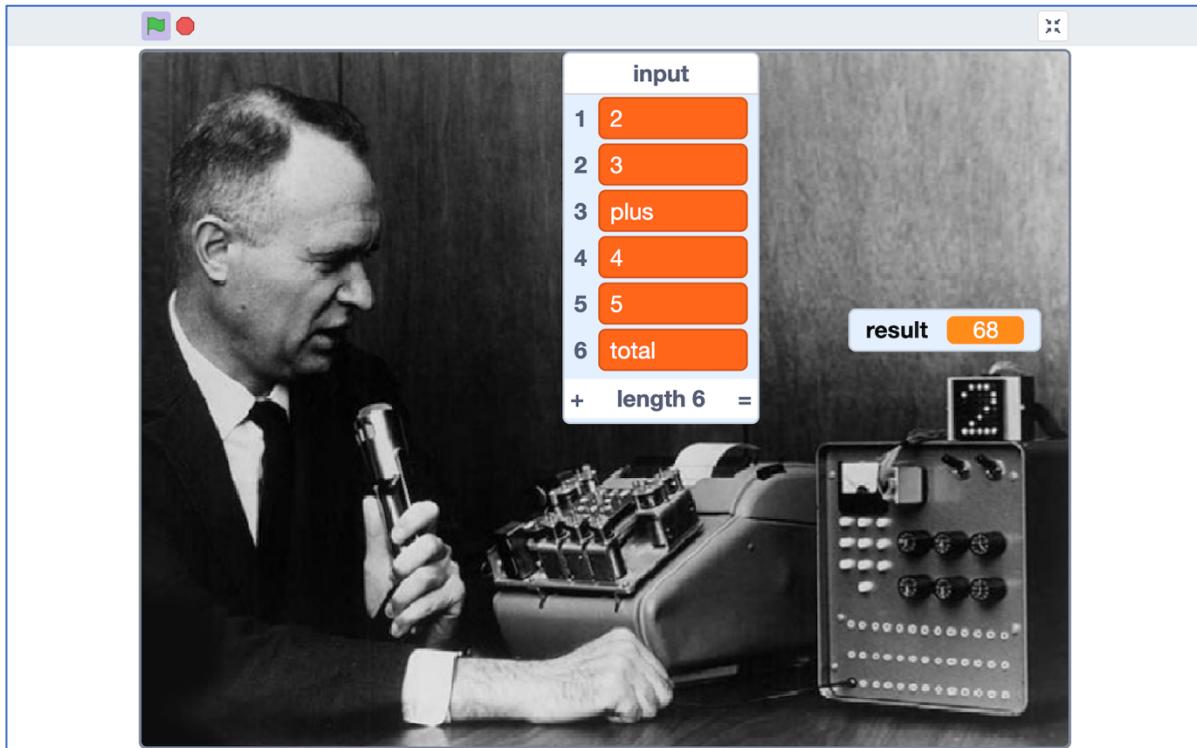
Say a sum.

Use the numbers you trained your project to recognise.

For example: "seven plus two minus one total"



You can do sums with larger numbers by saying them one digit at a time.
For example, to do 23 + 45 say “two three **plus** four five **total**”



What have you done?

You trained a voice-controlled assistant. Shoebox was an early predecessor of assistants we have today like Apple’s Siri and Amazon’s Alexa.

(It was called “Shoebox” because it was the size and shape of a shoebox.)

Artificial intelligence is a field of technology that has been developing for over eighty years. In this project, you explored a little early AI history.

If you didn’t have time to do it earlier, try <https://ibm.biz/shoebox> to learn a little more.

Tips

More examples!

Your machine learning model will make mistakes.

It could hear things when you don't say anything. It could hear you say numbers you do say more than once.

Be patient with it! You trained a very small machine learning model with a small number of examples. It won't be perfect.

The more examples you give it, the better the computer should get at recognising your words.

Background noise makes a difference

You recorded examples of the background noise in the room you're in. This was to help the computer to learn to tell what to listen out for.

The more the background noise changes while you work on this project, the worse your machine learning model will perform.

If you start playing background music while you test your project, it will affect how well your machine learning model recognises what you say.

If you trained it in a quiet room, but try to test it in a noisy room, it will likely struggle.

Ideas and Extensions

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

Fixing mistakes

Can you modify the Scratch project so you can get it to fix mistakes when it mishears what you say?

More types of sums

Your calculator can add and subtract numbers.

How could you modify it so that it could also multiply numbers?