



Catch the ball

In this project you will train a computer to predict where a ball will land.

You will teach the computer to make these predictions by giving it examples of many bouncing balls.

The image shows a Scratch project titled "Catch the ball". The stage view on the right displays a ball at coordinates (-230, 17) and a catcher sprite at (0, 0). The stage has a grid from -300 to 300 on both axes. The script area on the left contains the following code:

- A yellow "when green flag clicked" hat block with a blue "reset" sub-block.
- A blue "go to x: -240 y: -180" hat block.
- A yellow "when I receive [new simulation]" hat block with a blue "glide 0.3 secs to x: (X:0,Y:180)" sub-block. This block also includes parameters: starting location (x: -230, y: 17), starting speed (x: 11, y: 9), predict (bounce x: from start y: initial location y: speed: 50), and a comment: "once you have a model, you can use it now to predict what will happen".
- A yellow "when green flag clicked" hat block with a blue "forever" loop:
 - An orange "if key [left arrow v] pressed? then" control block with a blue "change x by -5" motion block.
 - An orange "if key [right arrow v] pressed? then" control block with a blue "change x by 5" motion block.



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 “Catch the ball” and set it to learn how to **predict numbers**. Store the project **in your web browser**. Click the “Create” button

Start a new machine learning project

Project Name *

Catch the ball

Project Type *

predicting numbers (beta)

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"

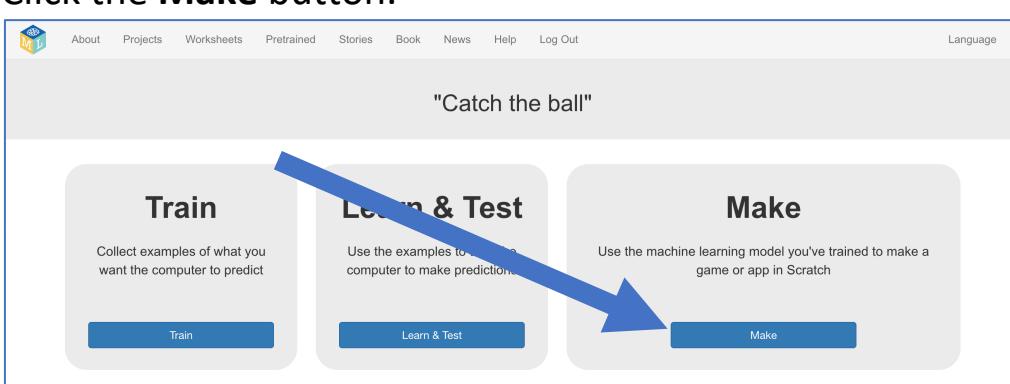
Storage *

In your web browser

CREATE CANCEL

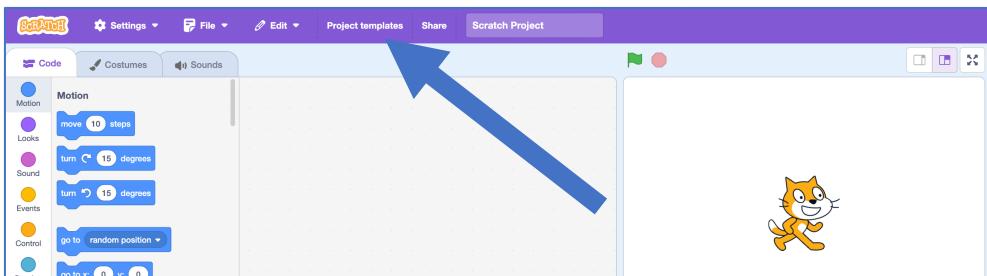
- 6.** You should see “Catch the ball” in your list of projects. Click on it.

- 7.** We’ll start by looking at what we’ll be training the computer to do. Click the **Make** button.



8. Click on the **Scratch 3** button, and then click on **Scratch by itself**

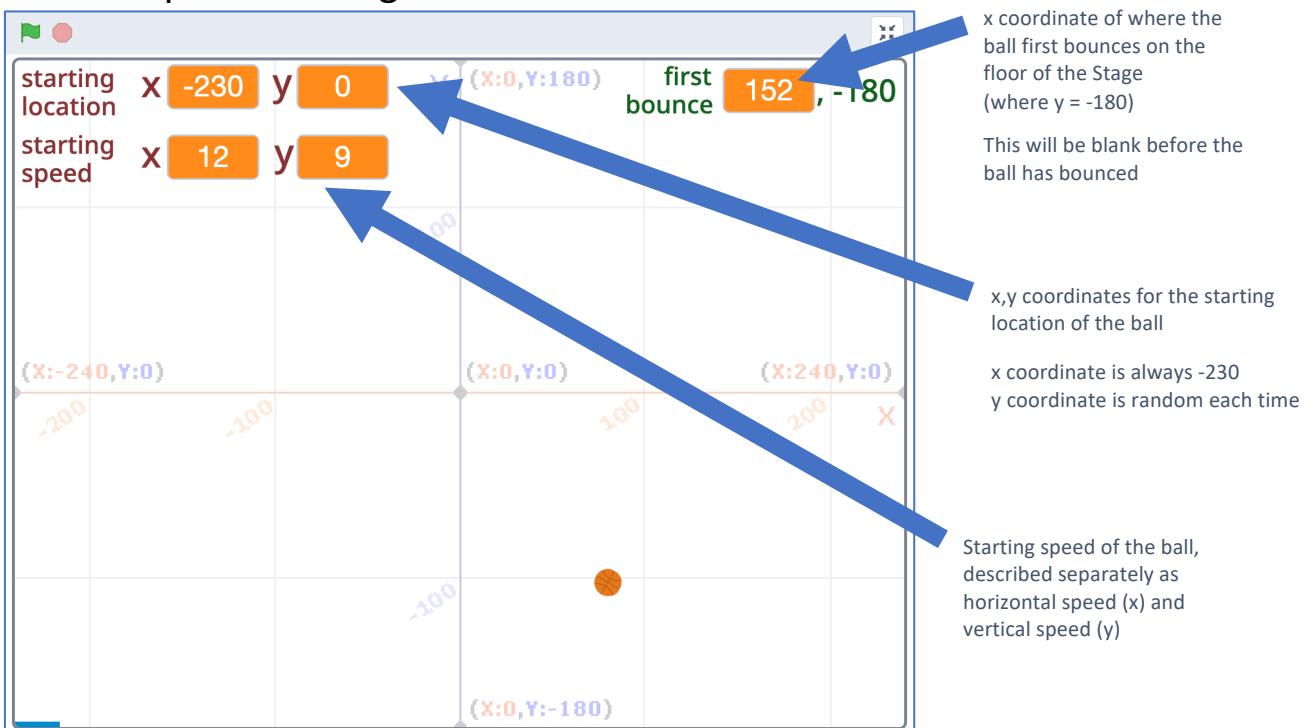
9. In the new Scratch window that opens, click on **Project templates**



10. Click on the “**Catch the ball**” template

11. Try clicking on the **Green flag**

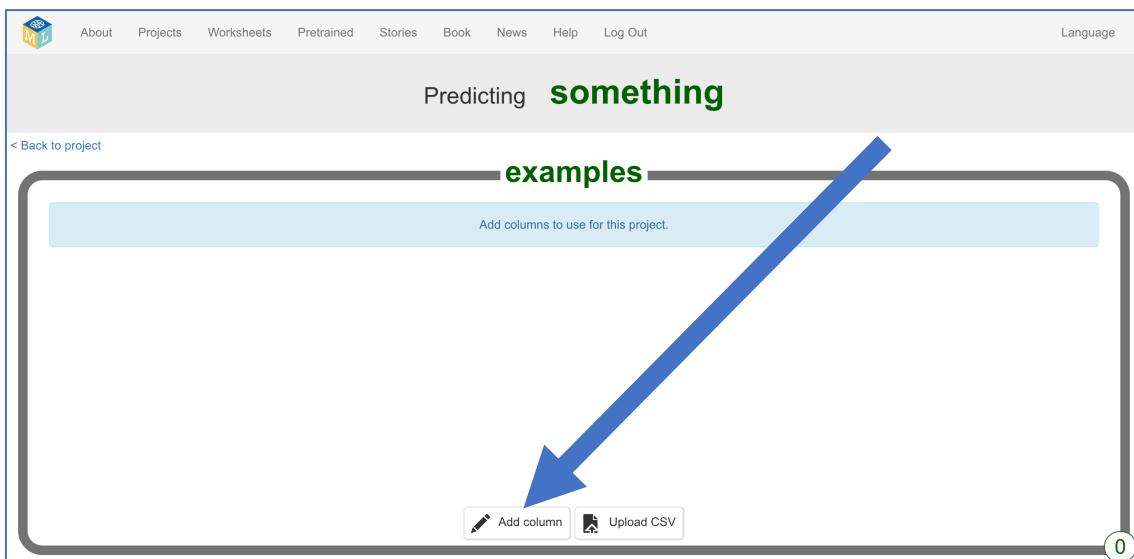
A ball will be fired from the left side of the Stage, towards the right, at a random speed and angle.



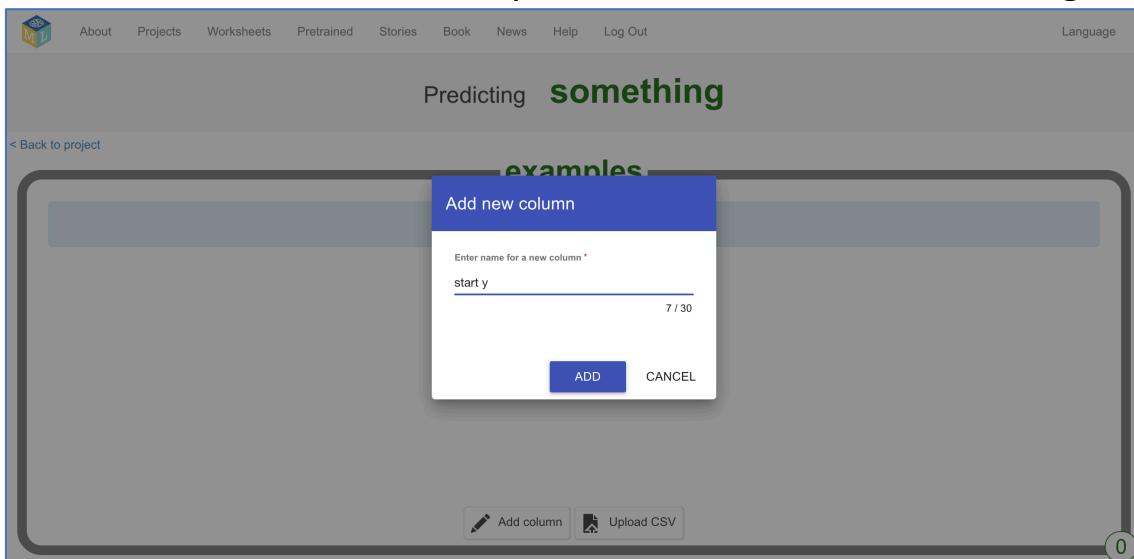
The goal of this project is to predict where the ball will bounce (the first time).

You will train the computer, using examples of observing the ball bouncing, so it can learn to predict where the ball will go (before it starts moving).

- 12.** Close the Scratch window and go back to the training tool
- 13.** Click on **Back to project**
- 14.** Next, we'll describe the data that we'll be using for the project.
Click the **Train** button
- 15.** Click the **Add column** button



- 16.** Enter “start y” for the y coordinate of the ball’s starting location
- 17.** Click on **Add**



18. Click the **Add column** button again, and create a column for the starting horizontal (x) speed of the ball : “speed x”

19. Click the **Add column** button again, and create a column for the starting vertical (y) speed of the ball : “speed y”

20. Click the **Add column** button again, and create a column for the x coordinate of the first bounce : “bounce x”

21. Click the checkbox underneath the “bounce x” column to show that this is the output value that the computer will learn to predict

The screenshot shows the 'examples' section of the ML Scratch interface. It displays four input columns: 'start y', 'speed x', 'speed y', and 'bounce x'. Each column has a checkbox below it. The 'bounce x' checkbox is checked, indicated by a blue arrow pointing to it. At the bottom of the section are three buttons: 'Change output column(s)', 'Add column', and 'Upload CSV'. To the right of the 'examples' section is a green circle with the number '0'.

22. You should have:

- * Three input columns – values you have at the start of a new bouncing ball simulation
- * One output column – the value that the computer will learn to predict

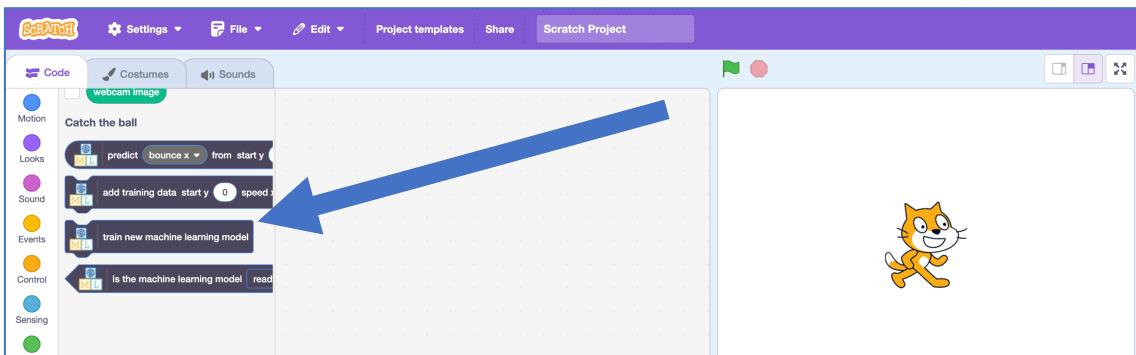
The screenshot shows the 'examples' section of the ML Scratch interface. It displays four input columns: 'start y', 'speed x', 'speed y', and 'bounce x'. The 'bounce x' column has a checked checkbox below it, indicating it is the output column. At the bottom of the section are three buttons: 'Change output column(s)', 'Add column', and 'Upload CSV'. To the right of the 'examples' section is a green circle with the number '0'.

23. Click on **Back to project**

24. It is time to collect training examples to train the computer.
Click the **Make** button

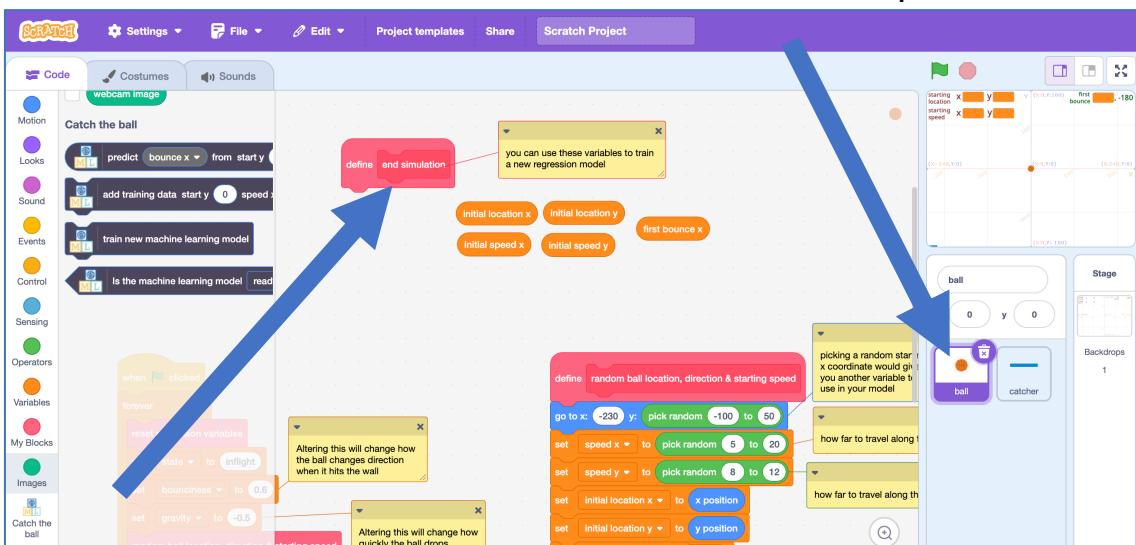
25. Click the **Scratch 3** button and then click on **straight into Scratch**

26. In the Scratch window that opens, scroll to the bottom of the toolbox to see the new blocks for your machine learning project



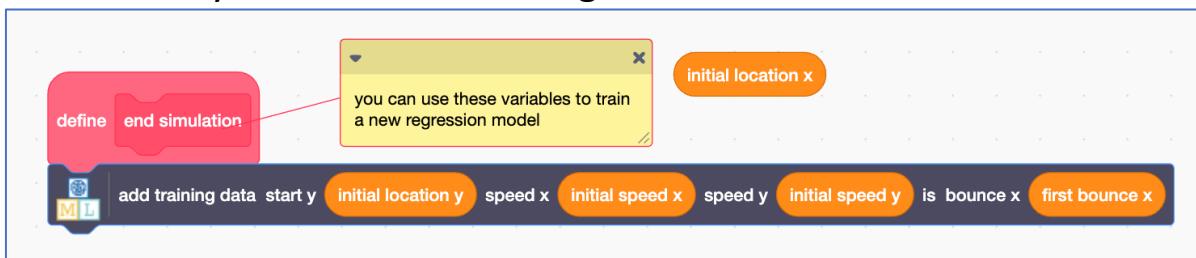
27. Click on **Project templates** & open the “**Catch the ball**” template

28. Find the “**end simulation**” block in the “**ball**” sprite



The code in this block will be run at the end of every simulation.
This is after the ball has finished bouncing, so the coordinates of the first bounce will be available.

- 29.** Update the code so that it stores the values for each simulation to use to train your machine learning model



Notice that we're not using the x coordinate for the initial ball location as it is always the same – the computer can't learn anything from it now.

- 30.** Click on the **Green Flag**

- 31.** Let the simulation play for **ten** balls

- 32.** In the training tool window, go back to the **Train** page

Keep the Scratch window open as you will return to it in a moment

- 33.** Review the training examples that you have collected

A screenshot of the Machine Learning training tool interface. The title bar says "Predicting **bounce x** from 3 input values". Below the title, there is a "examples" section containing a table with the following data:

start y	speed x	speed y	bounce x
19	12	8	166
-4	5	9	15
-42	5	8	-10
-62	9	8	148
-4	13	9	129
-54	16	11	-6
9	7	8	99
40	7	9	134
-93	10	8	170

At the bottom of the table are buttons for "Change output column(s)", "Upload CSV", "Download CSV", and "Delete all". A green circle with the number "10" is located in the bottom right corner of the table area.

- 34.** Click on **Back to project**

35. Click the Learn & Test button

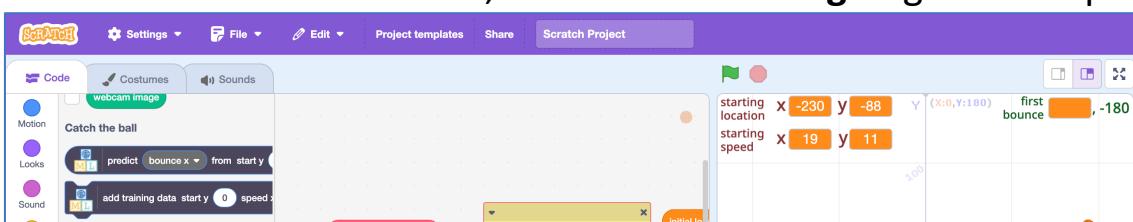
36. Click on Train new machine learning model

The screenshot shows the 'Machine learning models' page. On the left, under 'What have you done?', it says 'You have collected examples of numbers for a computer to use to predict the value of bounce x.' and 'You've collected: • 10 examples of data'. On the right, under 'What's next?', it says 'Ready to start the computer's training?' and 'Click the button below to start training a machine learning model using the examples you have collected so far'. A blue arrow points to the 'Train new machine learning model' button at the bottom of the right panel.

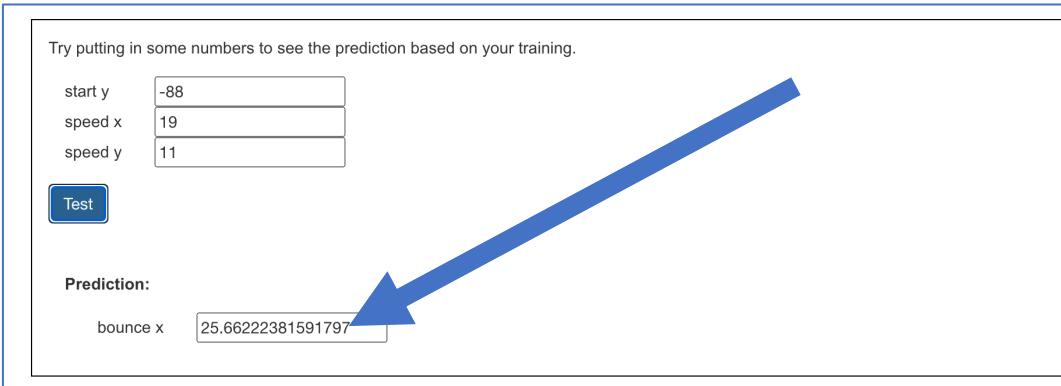
37. You have your first machine learning model. It can make a prediction if you provide input values here.

The screenshot shows the 'Machine learning models' page after training. Under 'What have you done?', it says 'You have trained a machine learning model to predict the values of bounce x.' and 'You created the model on Wednesday, February 21, 2024 10:19 PM.' Under 'What's next?', it says 'Try testing the machine learning model below. Enter an example of numbers below, that you didn't include in the examples you used to train it. It will tell you what it predicts the output should be.' A blue arrow points to the input fields for 'start y', 'speed x', and 'speed y' at the bottom left.

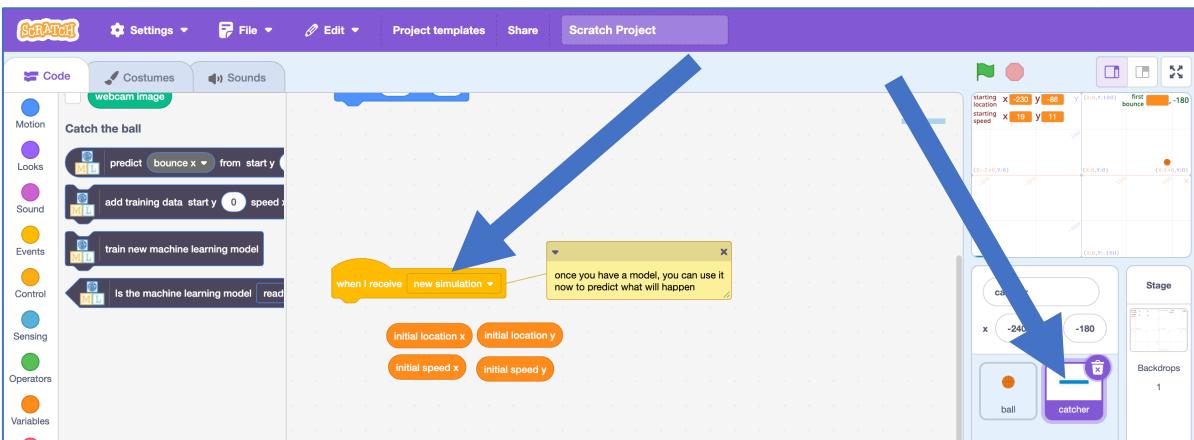
38. In the Scratch window, click the Green Flag to get new input values



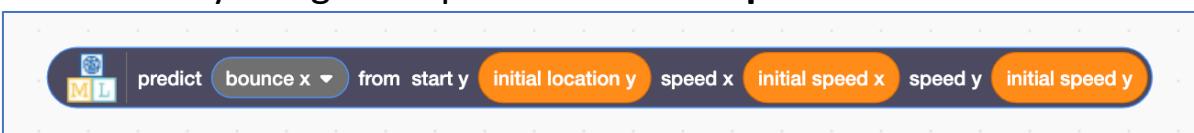
- 39.** Enter the input values and click on the **Test** button
The predicted x coordinate of the first bounce will be displayed



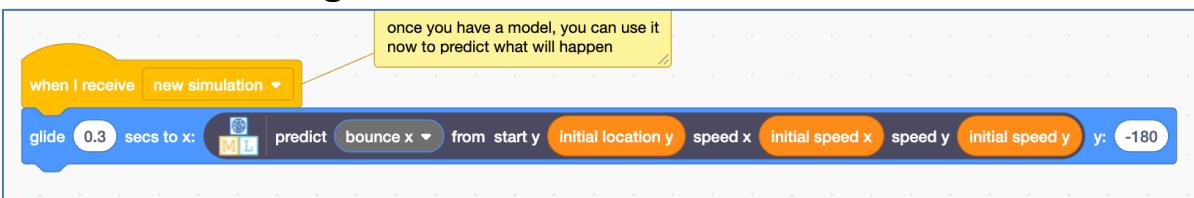
- 40.** Next, we'll use the prediction in Scratch. Return to the Scratch window, and find the “new simulation” code in the “catcher” sprite



- 41.** Start by using the input variables in a **predict** block



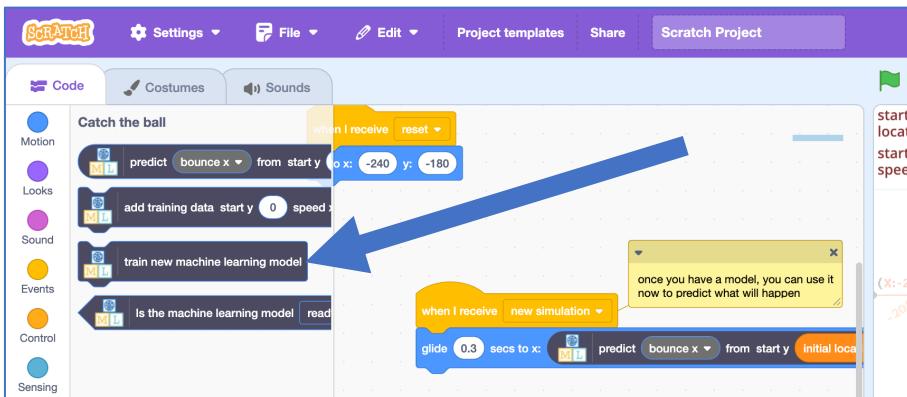
- 42.** Add this to a **glide** block and add it all to the **new simulation** hat



- 43.** Click on the **Green Flag** again.
The “**catcher**” sprite should move to where the ball will bounce.

44. If the catcher sprite doesn't move, this may be because your trained model was not detected by Scratch.

Click on the “train new machine learning model” block to train a new one



45. Observe a few ball simulations.

How many balls is the catcher sprite able to catch?

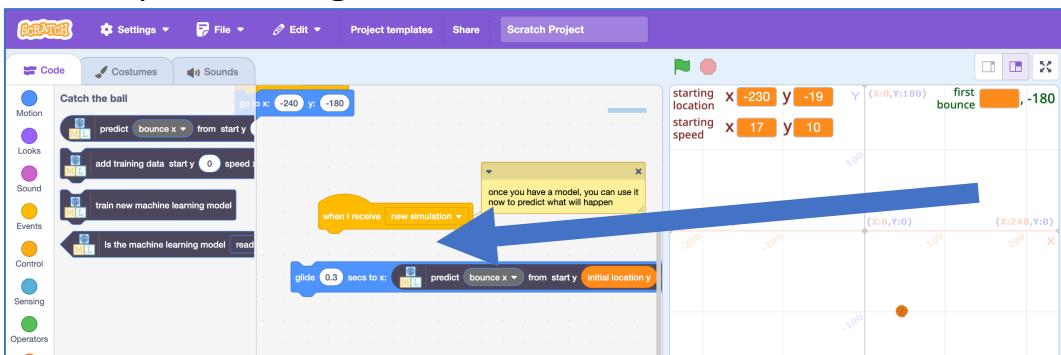
What have you done so far?

You've started to train a computer to make a prediction of the most likely output number, based on a few different input numbers. 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”.

The type of model you have trained is called a “regression” model.

With only ten training examples, your model is probably not making good predictions. More training examples should help it learn to do better.

46. Separate the glide block from the new simulation hat



47. Click the **Green Flag** again and leave the simulation running to collect lots more training examples.

48. You can review the training examples you have collected while your Scratch project runs. The number of examples is displayed in the corner.

The screenshot shows a Scratch project titled "Predicting bounce x from 3 input values". Below the title, there is a table with columns for start y, speed x, speed y, and bounce x. A large blue arrow points from the bottom right towards the number 101, which is enclosed in a green circle at the bottom right of the table. The table contains the following data:

start y	speed x	speed y	bounce x
19	12	8	166
-4	5	9	15
-42	5	8	-10
-62	9	8	148
-4	13	9	129
-54	16	11	-6
9	7	8	99
40	7	9	134
-93	10	8	170

At the bottom of the table are buttons for "Change output column(s)", "Upload CSV", "Download CSV", and "Delete all".

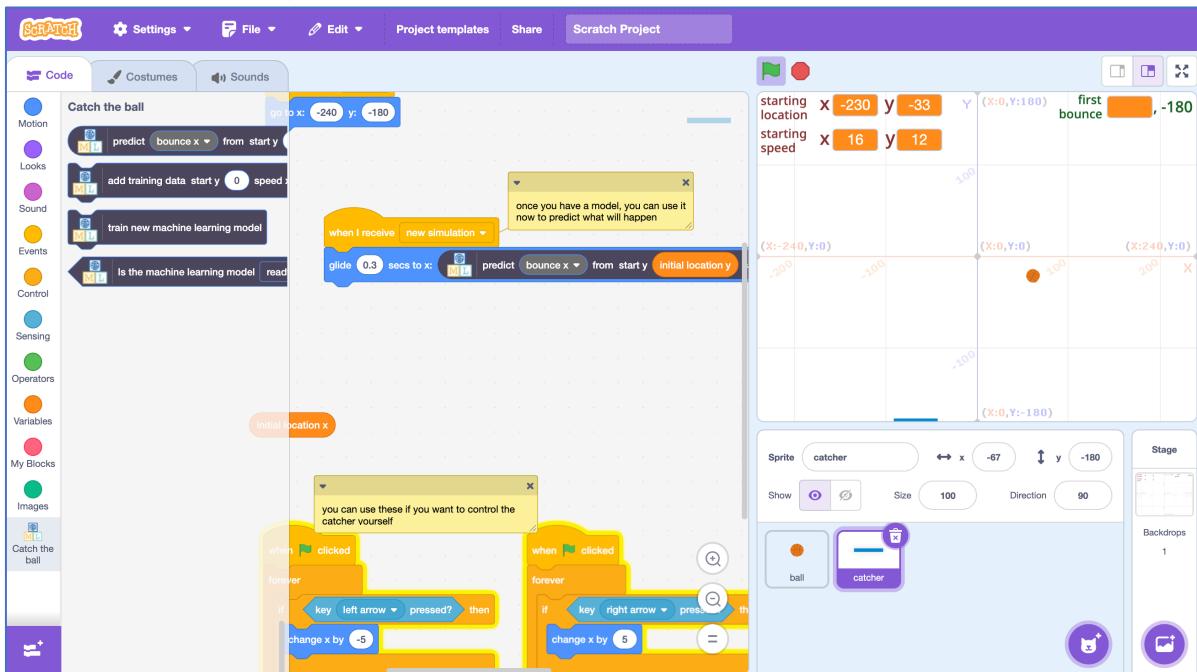
49. Click on the “train new machine learning model” block to train a new model using your new, larger set of training examples

The screenshot shows the Scratch project interface with the "train new machine learning model" block highlighted in yellow. A large blue arrow points from the bottom left towards this block. The script on the stage includes a "when I receive new simulation" hat and a "glide 0.3 secs to x: [predict bounce x from start y initial location y]" block.

50. Re-attach the glide block to the new simulation hat

The screenshot shows the Scratch project interface with the "glide 0.3 secs to x: [predict bounce x from start y initial location y]" block reattached to the "when I receive new simulation" hat. A large blue arrow points from the bottom left towards this reattachment point.

51. Click the Green Flag again to run the simulation with your improved machine learning model



What have you done?

You've improved the accuracy of your regression model by increasing the number of training examples that the model has had to learn from.

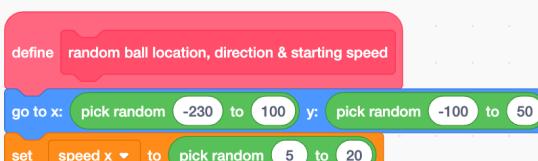
The computer learns from patterns in the examples you give it, and uses these to make predictions about the most likely output value.

Ideas and Extensions

Try the project again but make it more complicated!

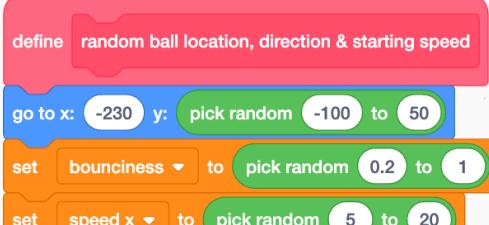
Change the starting x coordinate

Change the Scratch project code for the ball sprite to make the ball start at a random x coordinate.



Change the bounciness

Change the Scratch project code for the ball sprite to vary how bouncy the ball is.



Change gravity

Change the Scratch project code for the ball sprite to vary the effect of gravity on how the ball flies.



Use one (or more!) of these as an additional input variable.

Can you create a machine learning model that can predict the affect that it will have on where the ball bounces?