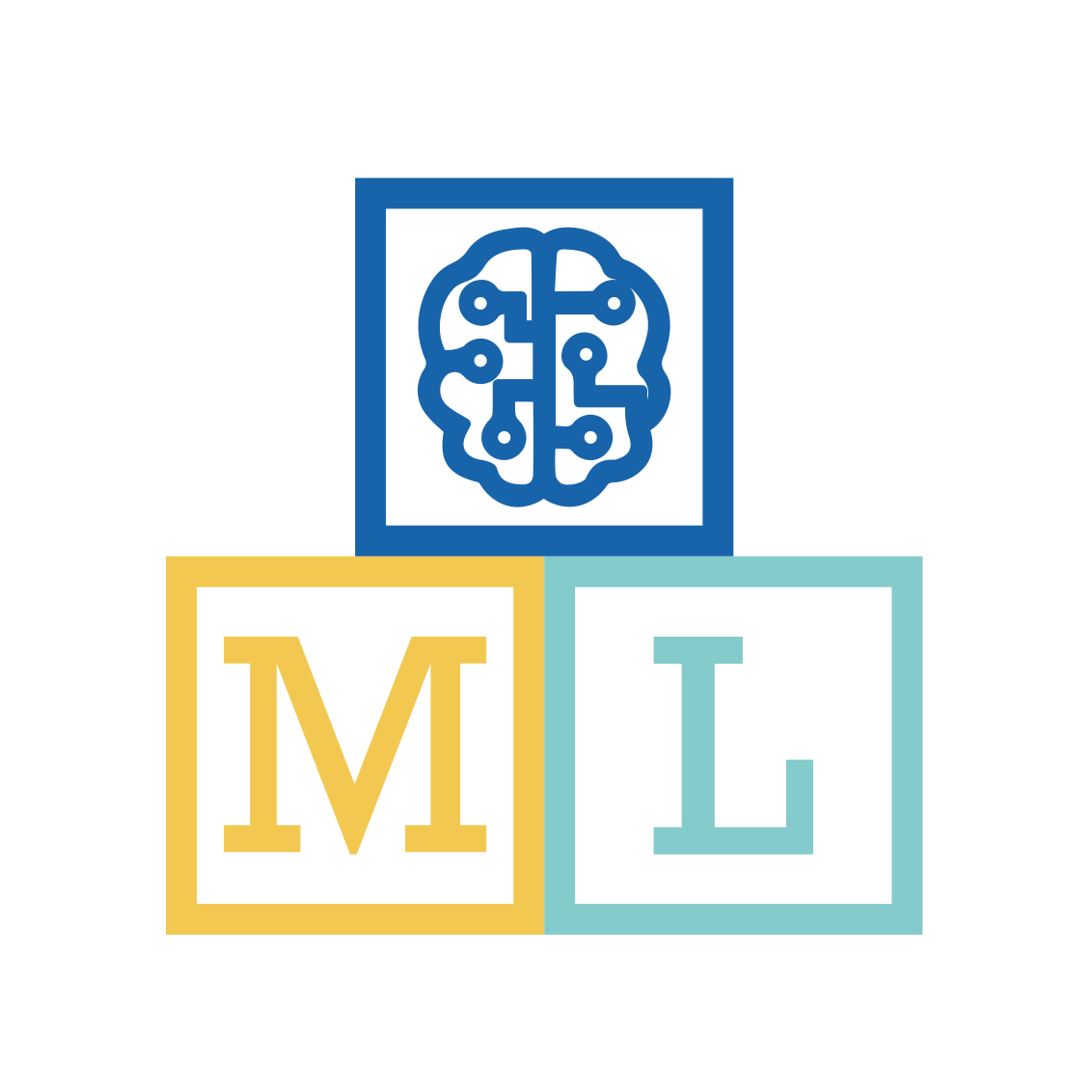
Catch the ball



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.

A screenshot of a chat

Description automatically generated

Creative Commons LicenceThis 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**”.   
   Click the “**Create**” button  
   A screenshot of a computer

   Description automatically generated
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.   
   A screenshot of a computer test

   Description automatically generated
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**A screenshot of a computer

   Description automatically generated
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.

x coordinate of where the ball first bounces on the floor of the Stage   
(where y = -180)

This will be blank before the ball has bounced

A screenshot of a graph

Description automatically generated

x,y coordinates for the starting location of the ball

x coordinate is always -230

y coordinate is random each time

Starting speed of the ball, described separately as horizontal speed (x) and vertical speed (y)

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).

1. Close the Scratch window and go back to the training tool
2. Click on **Back to project**
3. Next, we’ll describe the data that we’ll be using for the project.   
   Click the **Train** button
4. Click the **Add column** button  
   A screenshot of a computer

   Description automatically generated
5. Enter a name for the y coordinate of the ball’s starting location  
   A screenshot of a computer

   Description automatically generated
6. Click on **Add**
7. Click the **Add column** button again, and create a column for the starting horizontal (x) speed of the ball : “speed x”
8. Click the **Add column** button again, and create a column for the starting vertical (y) speed of the ball : “speed y”
9. Click the **Add column** button again, and create a column for the   
   x coordinate of the first bounce : “bounce x”
10. Click the checkbox underneath the “**bounce x**” column to indicate that this is the output value that the computer will learn to predict  
    A screenshot of a computer

    Description automatically generated
11. 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  
    A screenshot of a computer

    Description automatically generated
12. Click on **Back to project**
13. It is time to collect training examples to train the computer.   
    Click the **Make** button
14. Click the **Scratch 3** button and then click on **straight into Scratch**
15. In the Scratch window that opens, scroll to the bottom of the toolbox to see the new blocks for your machine learning project  
    A screenshot of a computer

    Description automatically generated
16. Click on **Project templates** & open the “**Catch the ball**” template
17. Find the “**end simulation**” block in the “**ball**” sprite  
    A screenshot of a computer

    Description automatically generated

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

1. Update the code so that it stores the values for each simulation to use to train your machine learning model  
   A screenshot of a computer

   Description automatically generated  
     
   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.
2. Click on the **Green Flag**
3. Let the simulation play for **ten** balls
4. 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
5. Review the training examples that you have collected  
   A screenshot of a computer

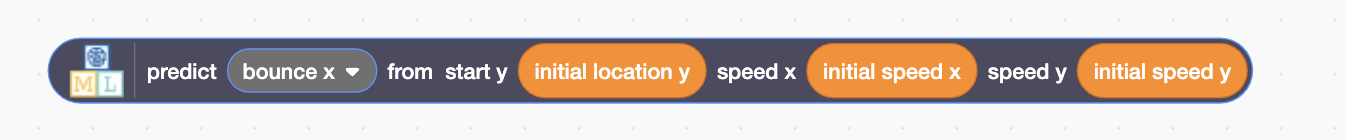
   Description automatically generated
6. Click on **Back to project**
7. Click the **Learn & Test** button
8. Click on **Train new machine learning model**   
   A screenshot of a computer program

   Description automatically generated
9. You have your first machine learning model.   
   It can make a prediction if you provide input values here.   
   A screenshot of a computer

   Description automatically generated
10. In the **Scratch** window, click the **Green Flag** to get new input values  
    A screenshot of a computer

    Description automatically generated
11. Enter the input values and click on the **Test** button  
    The predicted x coordinate of the first bounce will be displayed  
    A white rectangular object with black text

    Description automatically generated
12. Next, we’ll use the prediction in Scratch. Return to the Scratch window, and find the “**new simulation**” code in the “**catcher**” sprite   
    A screenshot of a chat

    Description automatically generated
13. Start by using the input variables in a **predict** block  
    
14. Add this to a **glide** block and add it all to the **new simulation** hat  
    A screenshot of a computer

    Description automatically generated
15. Click on the **Green Flag** again.  
    The “**catcher**” sprite should move to where the ball will bounce.
16. **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  
    *A screenshot of a chat

    Description automatically generated*
17. 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.

1. Separate the **glide** block from the **new simulation** hat   
   A screenshot of a chat

   Description automatically generated
2. Click the **Green Flag** again and leave the simulation running to collect lots more training examples.
3. You can review the training examples you have collected while your Scratch project runs. The number of examples is displayed in the corner.  
   A screenshot of a computer

   Description automatically generated
4. Click on the “**train new machine learning model**” block to train a new model using your new, larger set of training examplesA screenshot of a computer screen

   Description automatically generated
5. Re-attach the **glide** block to the **new simulation** hat   
   A screenshot of a chat

   Description automatically generated
6. Click the **Green Flag** again to run the simulation with your improved machine learning model  
   A screenshot of a chat

   Description automatically generated

**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.  
A screenshot of a phone

Description automatically generated

**Change the bounciness**

Change the Scratch project code for the ball sprite to vary how bouncy the ball is.  
A screenshot of a game

Description automatically generated

**Change gravity**

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

Description automatically generated

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?