

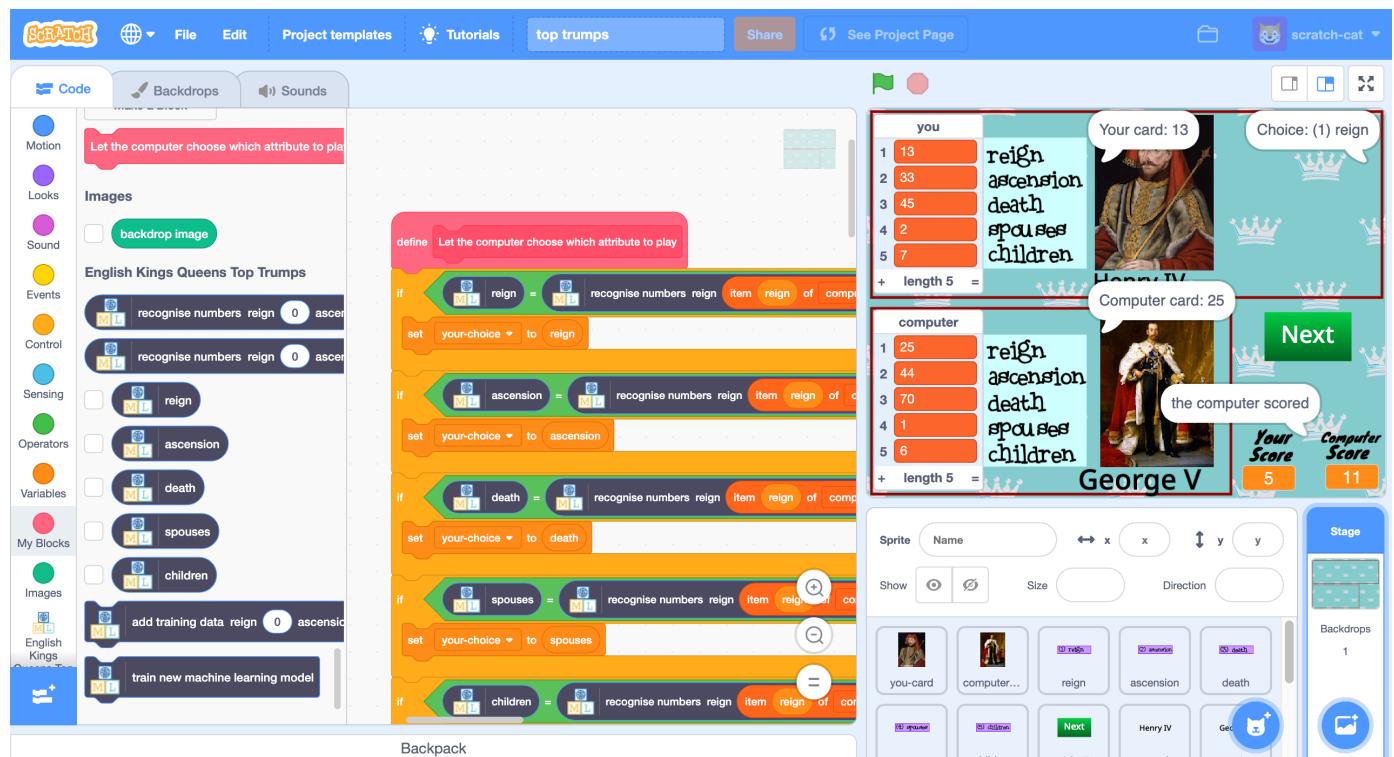
Top Trumps

In this project you will train a computer to play a card game.

For some values on the cards, you win by having the highest number. For others, you win by having the lowest. The range of numbers for different values will vary.

The aim will be for the computer to learn how to play the game well without you having to give it a list of all the cards or tell it the rules.

Instead, you'll try two different ways of training the computer to play the game by giving it examples of the game being played.



This project worksheet is licensed under a Creative Commons Attribution Non-Commercial Share-Alike License
<http://creativecommons.org/licenses/by-nc-sa/4.0/>

What will you be doing?

You'll be training a machine learning model to play a Top Trumps game based on Kings and Queens of England.

Each card is based on a different King or Queen.

Each card has five numbers on it:

Reign – how long they were king or queen for

Ascension – how old they were when they became king or queen

Death – how old they were when they died

Spouses – how many husbands or wives they had

Children – how many children they had

For example:



Charles I

1) He reigned for **23** years

If I choose this, and he was King **longer** than my opponent's card, I'll win

2) He ascended to the throne when he was **24**

If I choose this, and he became King **earlier** than my opponent's card, I'll win

3) He died when he was **48**

If I choose this, and he lived **longer** than my opponent's card, I'll win

4) He had **1** spouse

If I choose this, and he had **more** spouses than my opponent's card, I'll win

5) He had **9** children

If I choose this, and he had **more** children than my opponent's card, I'll win.

1. Go to <https://machinelearningforkids.co.uk/> in a web browser

2. Click on “**Get started**”

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

If you don't have a username, ask your teacher to create one for you.

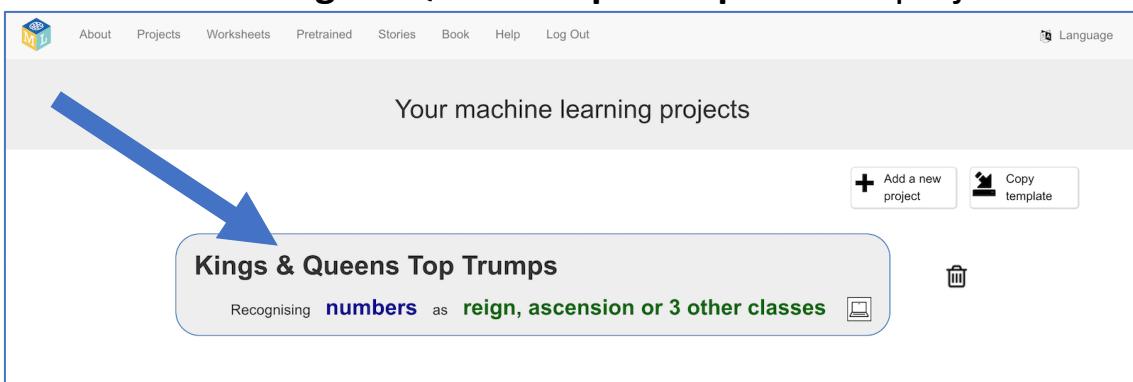
If you can't remember your username or password, ask your teacher to reset it for you.

4. Click on “**Projects**” on the top menu bar

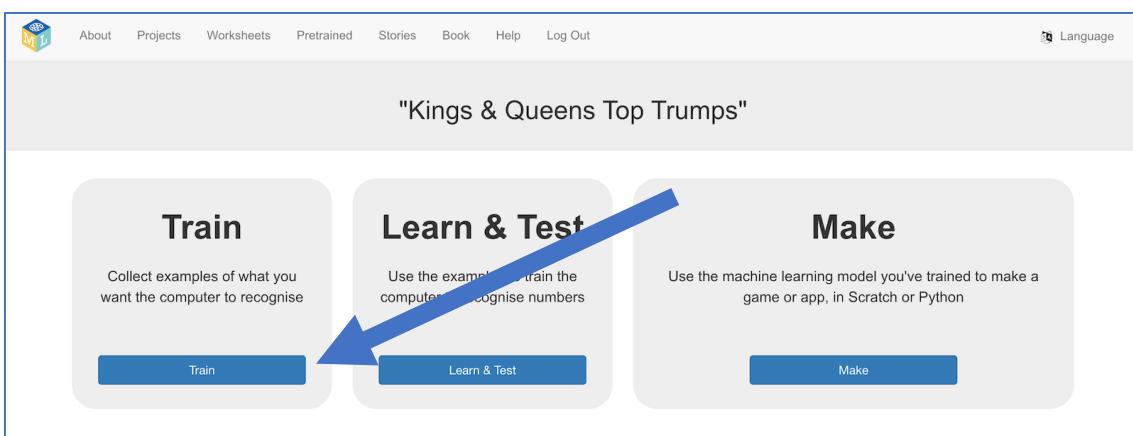
5. Click the “**Copy template**” button.

6. Import the “**Kings & Queens Top Trumps**” project template

7. Click on “**Kings & Queens Top Trumps**” in the projects list.



8. Click the “**Train**” button



- 9.** The project template gives you (empty) training buckets to store values from Top Trumps cards to train a machine learning model.

The screenshot shows a web-based machine learning project interface. At the top, there's a navigation bar with links for 'About', 'Projects', 'Worksheets', 'News', 'Help', 'Log Out', and 'Language'. Below the navigation, the title 'Recognising numbers as reign, ascension or 3 other classes' is displayed. Underneath the title, there are five rectangular boxes representing training buckets, each with a label above it: 'reign', 'ascension', 'death' in the top row, and 'spouses', 'children' in the bottom row. Each box has a small button labeled '+ Add example' at its bottom left. In the top right corner of the main area, there's a button labeled '+ Add new label' with a plus sign icon.

- 10.** Click the “< Back to project” link in the top-right corner

- 11.** Click the “Make” button

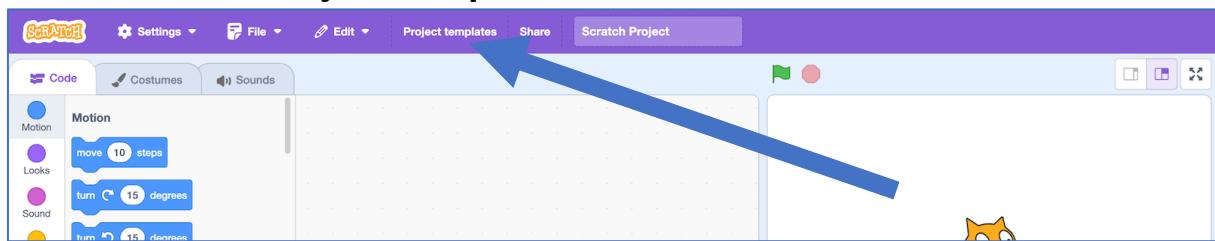
- 12.** Click the “Scratch 3” button

- 13.** Click on “straight into Scratch”

The page warns you that you haven't trained a machine learning model yet, but that's okay as you'll be using Scratch to collect training data.

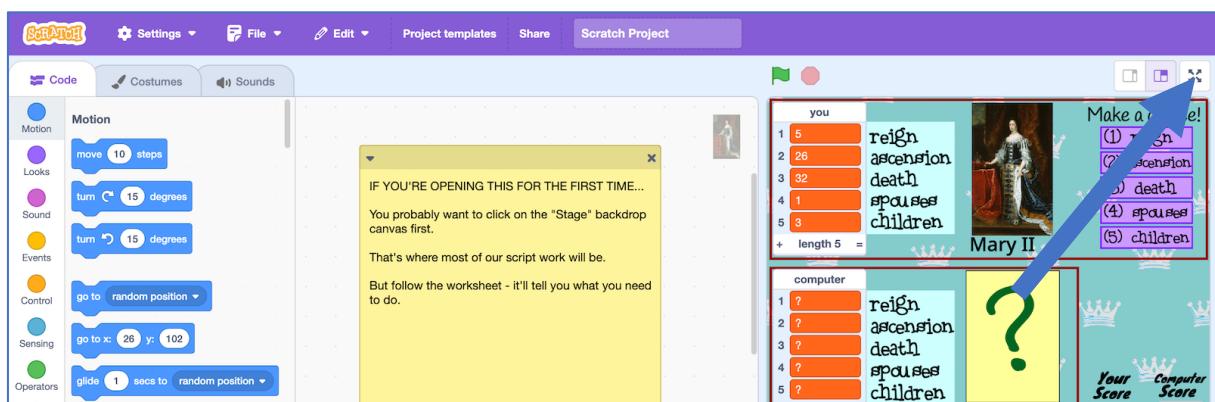
The screenshot shows the Scratch 3 interface. At the top, there's a navigation bar with links for 'About', 'Projects', 'Worksheets', 'News', 'Help', 'Log Out', and 'Language'. The title 'Using machine learning in Scratch 3' is displayed. A blue arrow points from the text 'The page warns you that you haven't trained a machine learning model yet.' to a button labeled 'straight into Scratch'. Below this, there are two sections: one explaining what your project will add to Scratch (blocks like 'recognise numbers distance 0 (label)') and another showing a screenshot of the Scratch 3 interface with a project titled 'my project'.

14. Click on “Project templates”



15. Click on “Top Trumps” to load the Top Trumps Scratch template

16. This is Top Trumps based on the Kings and Queens of England. Click the full-screen button.



17. Click the green flag to start

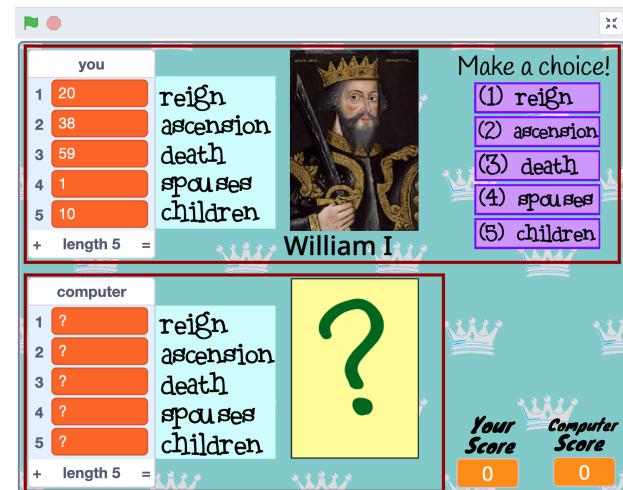
The top half of the screen is you.

The bottom half is the computer.

When you click the Green Flag to start, you can't see the computer's card yet.

It's all just question marks.

Choose a value from your king or queen by **clicking the purple button next to it**



When you choose, the computer card is revealed, and you see if you won or lost. The score in the bottom right corner is updated.

Click on the green Next button to move onto the next card and play again.

If you win or tie, it's your turn again.

If you lose, the computer will get to choose the next value instead.

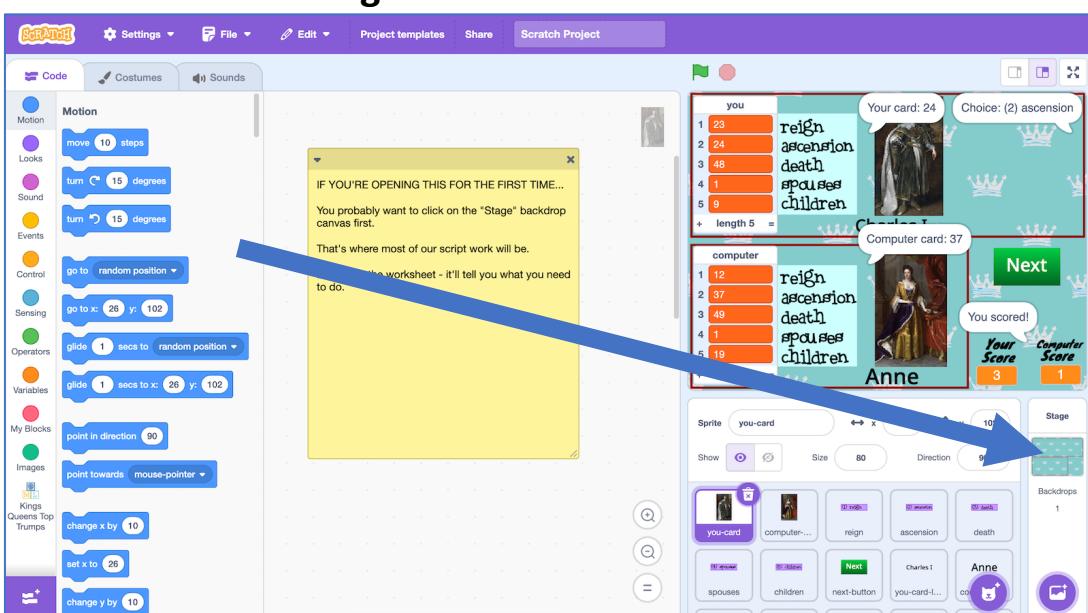
18. Play a few rounds of the game against the computer.

Try to work out how the computer is choosing values to play.

When you think you've worked out how the computer is playing, move onto the next step.

19. Click on the **full-screen** button again to go back to normal view.

Then click on the **Stage**



20. The code on the **Stage** shows how the computer has been coded.

The computer always picks “reign”.

Did you get it right?

```
define Let the computer choose which attribute to play
set your-choice to reign
```

21. Change the code so that the computer chooses a value at random when it's the computer's turn.

Choosing from 1 (reign) to 5 (children) at random.

```
define Let the computer choose which attribute to play
set your-choice to pick random 1 to 5
```

- 22.** Click the **green flag** to reset the scores to 0.
Go back to full-screen and play the game again.
Stop when either you or the computer reaches 10 points. Who won?

What have you done so far?

You've set up a bot to play Top Trumps and given it a simple strategy: choose values at random.

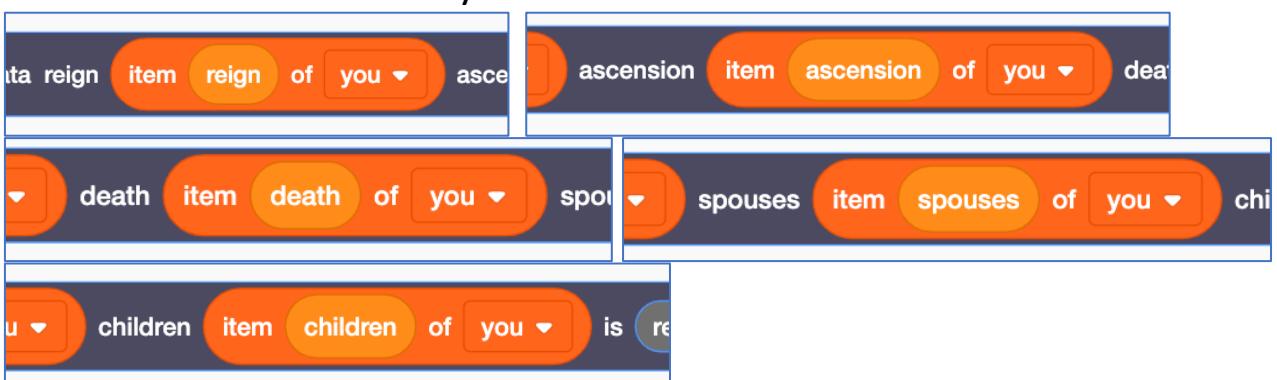
But people don't play like that. We learn how to choose which value would give us the best chance of winning. We do this based on the cards we've seen before, and on our understanding of the rules.

Next, you'll create Scratch code that collects examples using the moves that you make, that the computer can learn from.

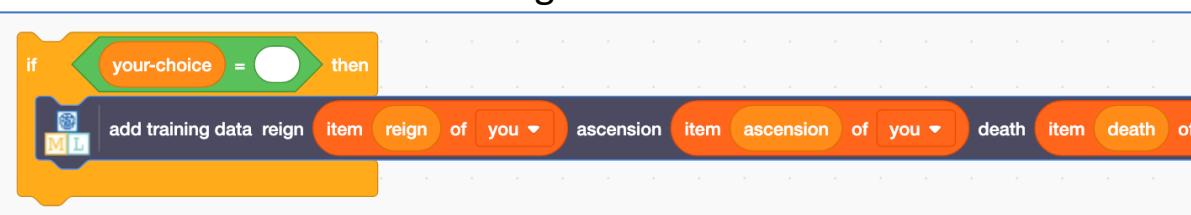
- 23.** Still on the **Stage**, drag the “add training data” block to the canvas



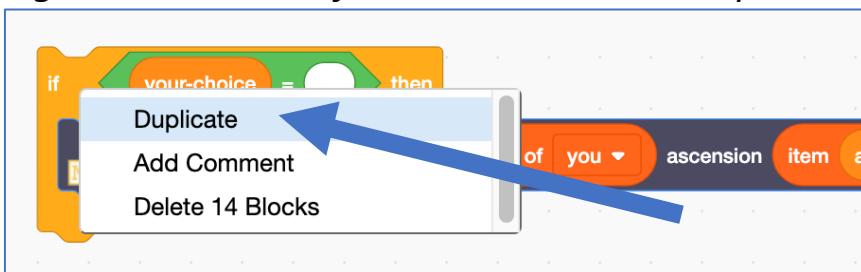
- 24.** Add the values from your card to the block



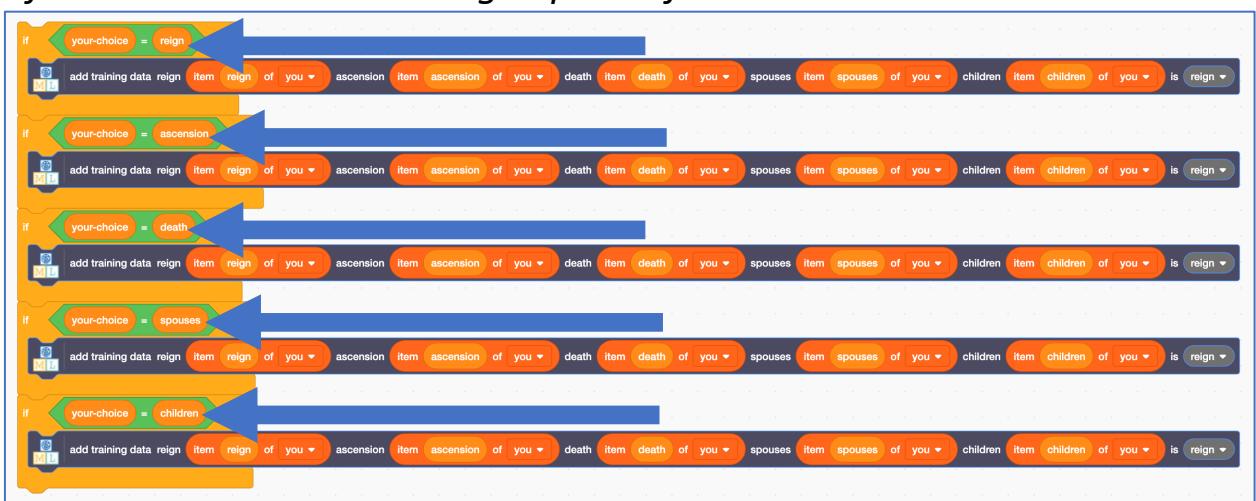
- 25.** Surround the “add training data” with an “if” block like this:



26. Duplicate it **five** times – once for each possible choice of value
Right-click on the “if” and then choose “Duplicate”



27. Add the choices to each if block : reign, ascension, death, spouses, children
“if” choices should use orange options from “Data”



28. Choose the matching choice for each “add training data” block



*Give it a quick check before you continue.
Have you matched everything (e.g. spouses with spouses) correctly?*

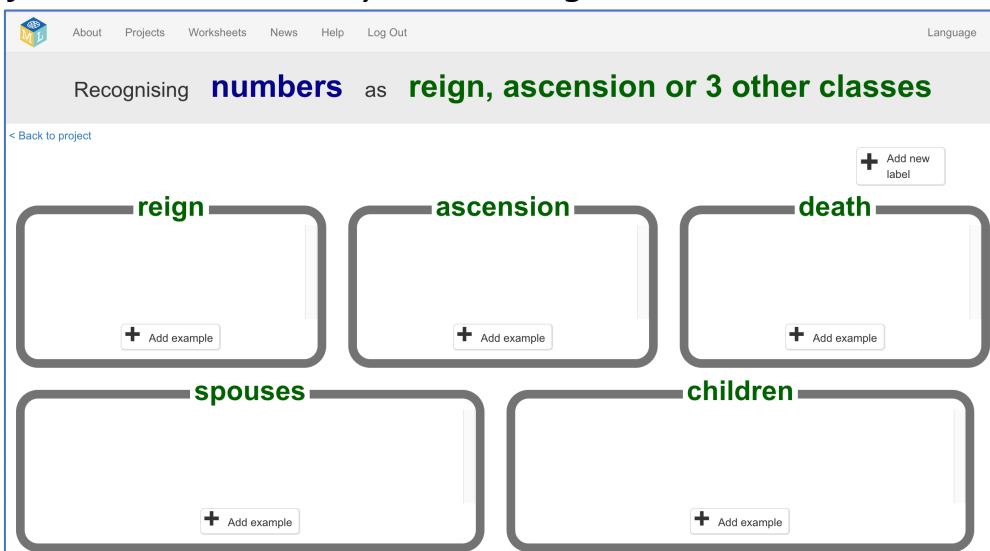
29. Finish the code so that this is called every time you win a hand
The values that were on your card, and the choice you made, will be added to the training examples every time you win.



30. Check your training data is empty

Accidentally clicking on the “add training data” block while creating your code can add empty values to your training buckets.

If there are values in your training, delete them now, so it looks like this.



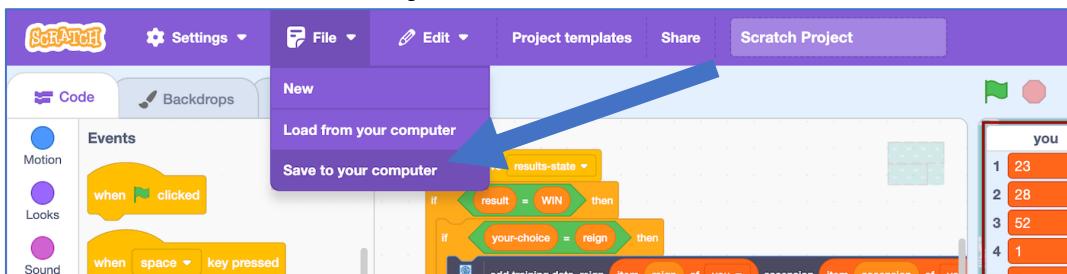
31. Time to create training data!

Click the **full-screen** button and **Green Flag** again.

Play games until your score reaches **10**.

32. Save your project

Click “**File**” -> “**Save Project**”



33. Leave the Scratch window open.

In the training window, click the “< Back to project” link.

34. Click the “Train” button

35. Check your training data

The ten winning moves that you made should have been added to your training examples.

Each example contains the numbers that were on your card.

The bucket (that the example is in) is the winning choice that you made.

A screenshot of a training data interface. At the top, it says "Recognising numbers as reign, ascension or 3 other classes". Below are five categories: "reign", "ascension", "death", "spouses", and "children". Each category has a list of examples with numerical values. Buttons for "+ Add example" are at the bottom of each category. Circled numbers 1 through 5 are placed next to the "ascension", "death", "spouses", "children", and "reign" categories respectively.

36. Click the “< Back to project” link. Then click “Learn & Test”

37. The training page won't let you train a model yet

The ten examples aren't enough yet to train a model.

The screenshot shows a web page titled "Machine learning models". At the top, there are navigation links: "About", "Projects", "Worksheets", "News", "Help", "Log Out", and "Language". Below the title, there is a link "[< Back to project](#)". The main content is divided into two sections: "What have you done?" and "What's next?".

What have you done?

You have collected examples of numbers for a computer to use to recognise when numbers are reign, ascension or 3 other classes.

You've collected:

- 5 examples of reign,
- 1 example of ascension,
- 4 examples of death,
- 0 examples of spouses,
- 3 examples of children

What's next?

Keep going!

Go back to the [Train](#) page and collect more examples for each of the labels.

The more you can get, the better it should learn, but you need at least five examples of each as an absolute minimum.

38. Leave the page open.

Go back to your game in Scratch and play more games.

39. After a while, go back to the “Machine learning models” page, and refresh the page.

Keep doing this until you’ve got enough examples for the “**Train new machine learning model**” button to appear.

The screenshot shows the same "Machine learning models" page after collecting more examples. The "What have you done?" section now lists 10 examples, and the "What's next?" section contains a button labeled "Train new machine learning model". A large blue arrow points to this button.

What have you done?

You have collected examples of numbers for a computer to use to recognise when numbers are reign, ascension or 3 other classes.

You've collected:

- 7 examples of reign,
- 5 examples of ascension,
- 7 examples of death,
- 4 examples of spouses,
- 7 examples of children

What's next?

Ready to start the computer's training?

Click the button below to start training a machine learning model using the examples you have collected so far

(Or go back to the [Train](#) page if you want to collect some more examples first.)

Train new machine learning model

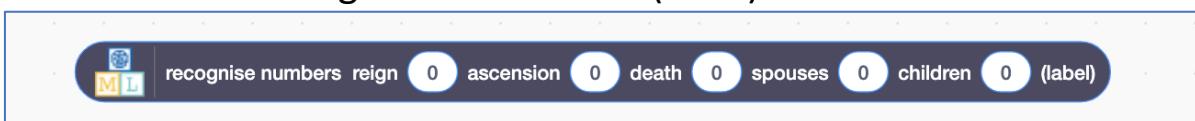
What have you done so far?

You've started to train a computer to learn about Top Trumps.

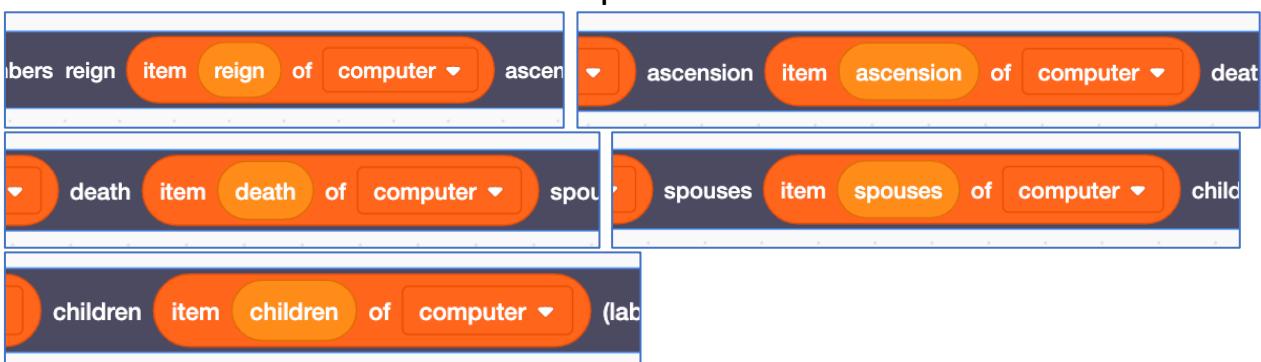
The examples help the computer learn what values to expect in cards: the range of numbers for each value, how often it should expect to see high values, how often it should expect to see low values.

The examples also help the computer to learn what numbers are likely to help it to win, without you needing to tell it what the rules are.

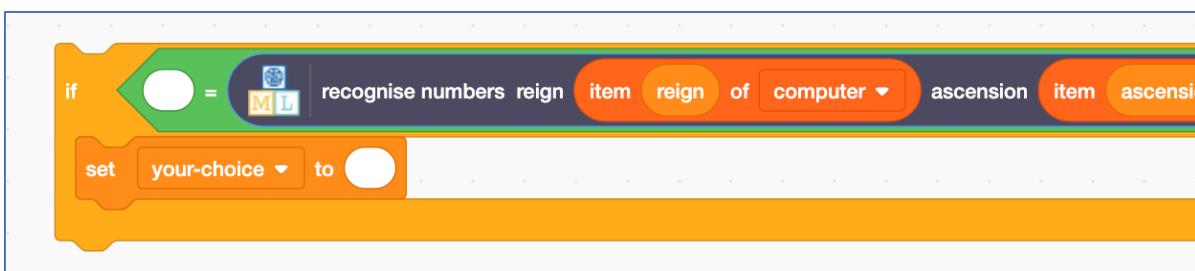
- 40.** Go back to the Scratch window
- 41.** Click on the “**Stage**” to get to where you added the code before
- 42.** Add the “recognise numbers ... (label)” block



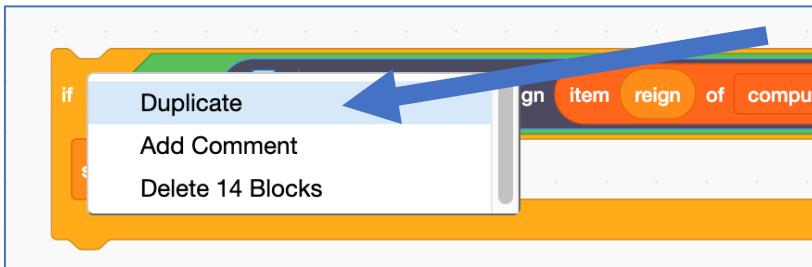
- 43.** Add the values from the computer's card to the block



- 44.** Add it to an “if” block to create code that looks like this:

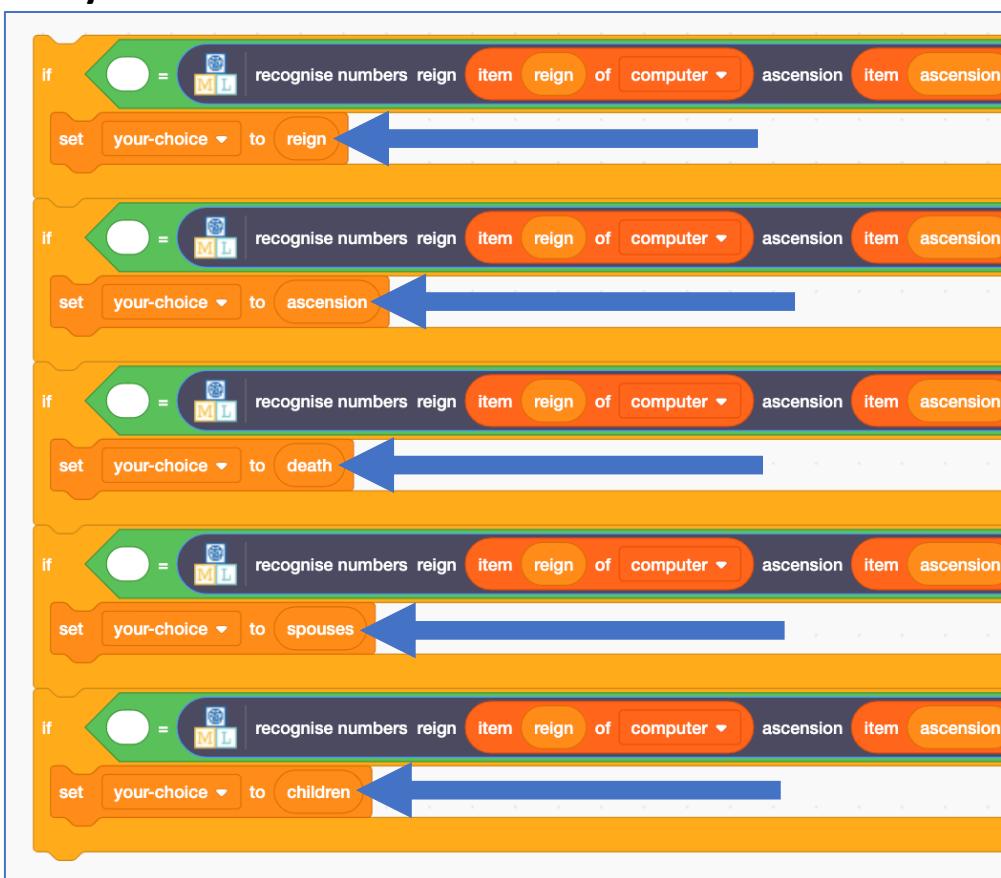


45. Right-click on the “if” block and click on “Duplicate”

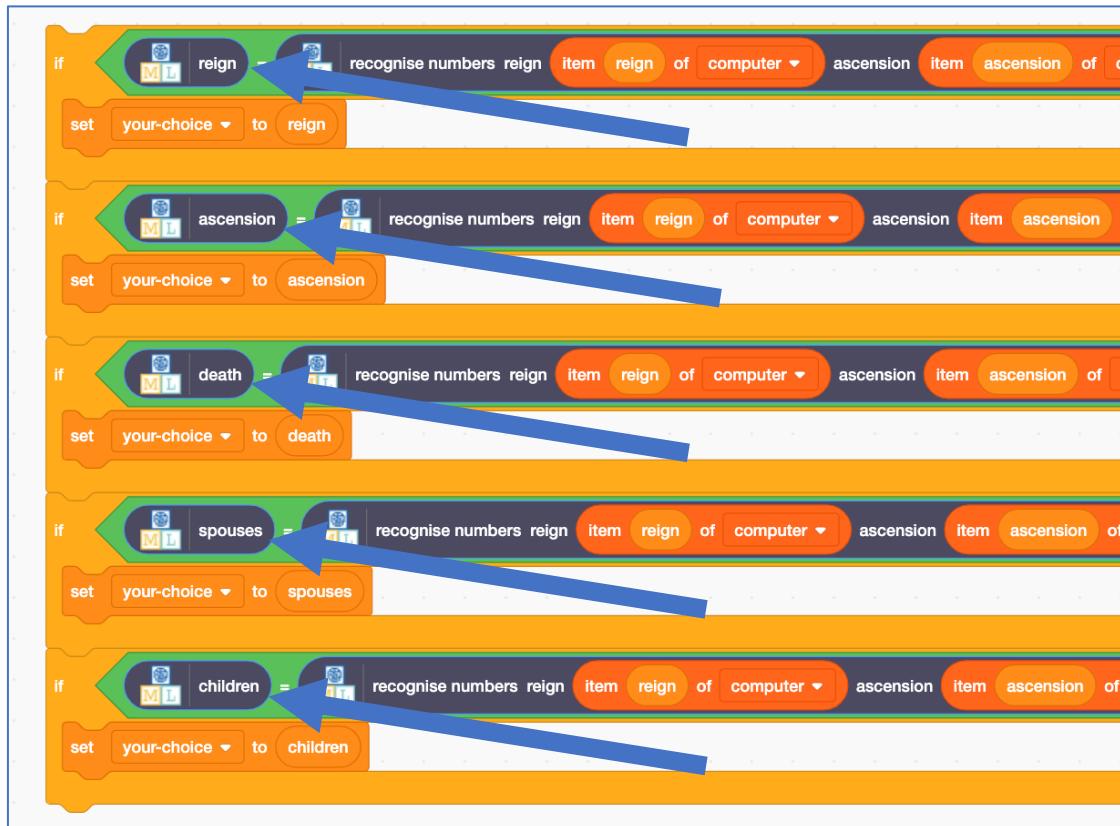


46. Repeat until you have 5 copies of the if block, joined together.

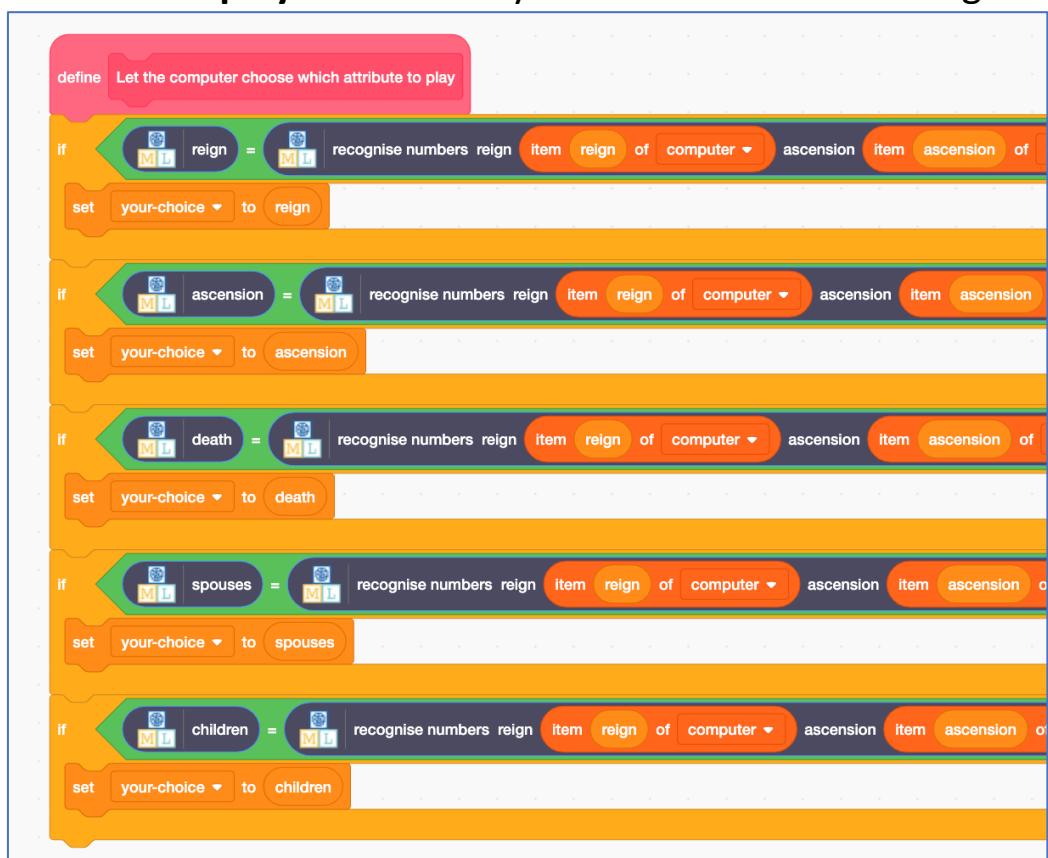
47. Copy the orange variable blocks for each choice into the “set your-choice” blocks



48. Copy the machine learning bucket blocks into the “if” blocks



49. Replace the contents of the “Let the computer choose which attribute to play” block with your new machine learning code



50. Save your project

Click “File” -> “Save to your computer”

51. Click the Green Flag and play against your machine learning model

Play until you reach the score of 20

What have you done so far?

You've modified your Scratch Top Trumps bot to use machine learning instead of your earlier random approach.

You haven't collected nearly enough examples to train a good model yet. The computer won't have seen enough examples of the game being played to have learned the types of values to expect, or the values that are more likely to win. Its predictions will often be wrong.

To get better, it needs more examples. Lots more examples.

52. Go back to the training tool window

53. Click the “< Back to project” link, then back to “Learn & Test”

54. Click the “Train new machine learning model” button

The screenshot shows the Scratch ML training interface. At the top left is a link to "Back to project". Below it are two sections: "What have you done?" and "What's next?". The "What have you done?" section shows statistics about the current model: 18 examples of reign, 7 ascension, 10 death, 4 spouses, and 11 children. The "What's next?" section provides instructions for testing the model with new numbers. At the bottom is the "Info from training computer" section, which displays the start time of the training (Sunday, March 15, 2020 1:49 AM) and the status (Available). A red "Delete this model" button is visible. A prominent blue arrow points from the text above to the "Train new machine learning model" button at the bottom of this section.

55. Switch back to the Scratch window.

If you accidentally closed it, you can get back to it by doing this:

- * Click the “**< Back to project**” link
- * Click the “**Make**” button
- * Click the “**Scratch 3**” button
- * Click the “**Open in Scratch 3**” button
- * Open the file you saved before, with “**File**”->“**Load from your computer**”

56. Play the game again.

Is it getting any better? Does the computer win more often now?

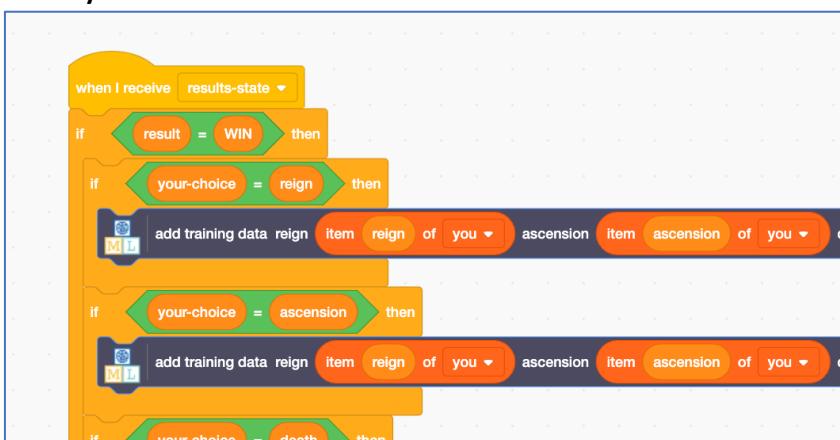
57. Repeat steps 51 – 55 to collect more examples, and then train a new machine learning model with them. Do this a few times.

What have you done so far?

The computer is only learning from the decisions that you make. To speed up the collection of training examples, next you’ll let the computer learn from its own moves as well.

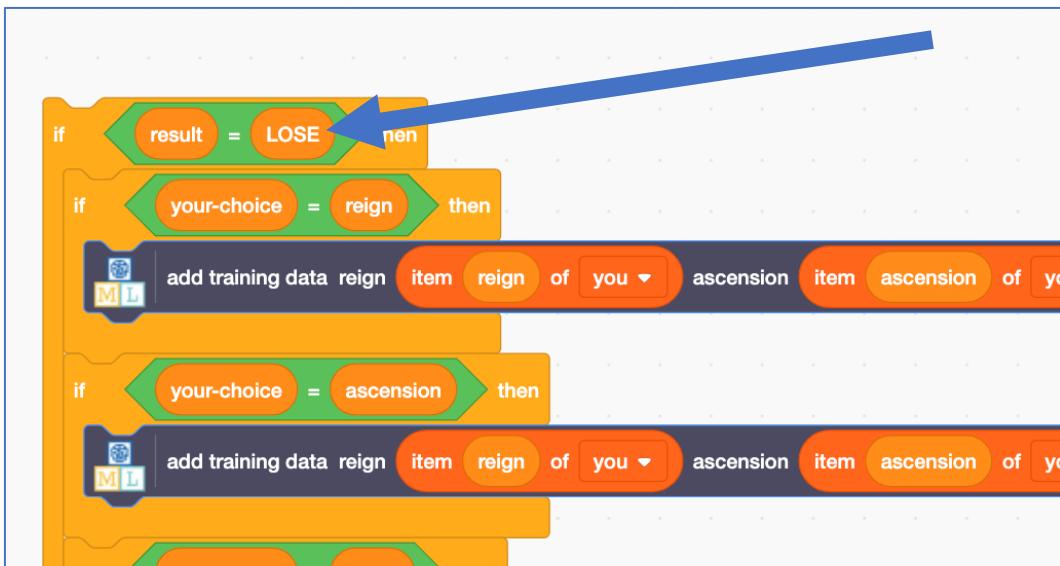
58. Open the game in Scratch

59. Find the “When I receive ‘results-state’” code in the Stage that you made earlier



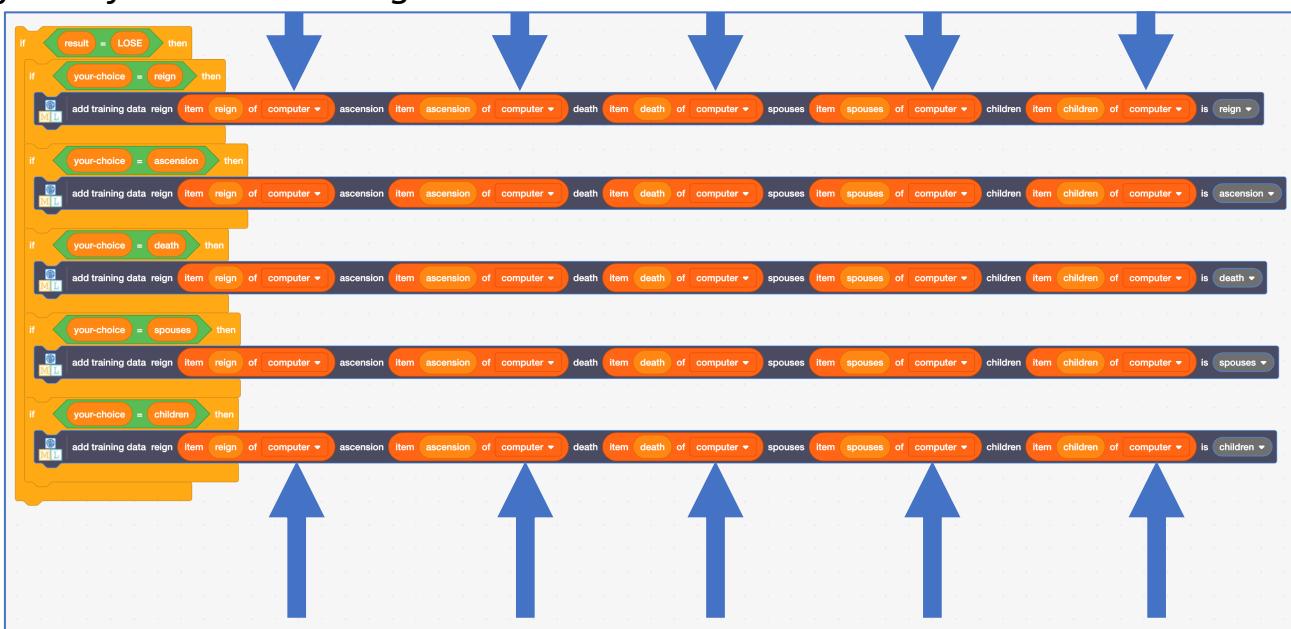
60. Duplicate the “**if result = WIN**” block

61. Modify the duplicate so that it says “**if result = LOSE**” at the top

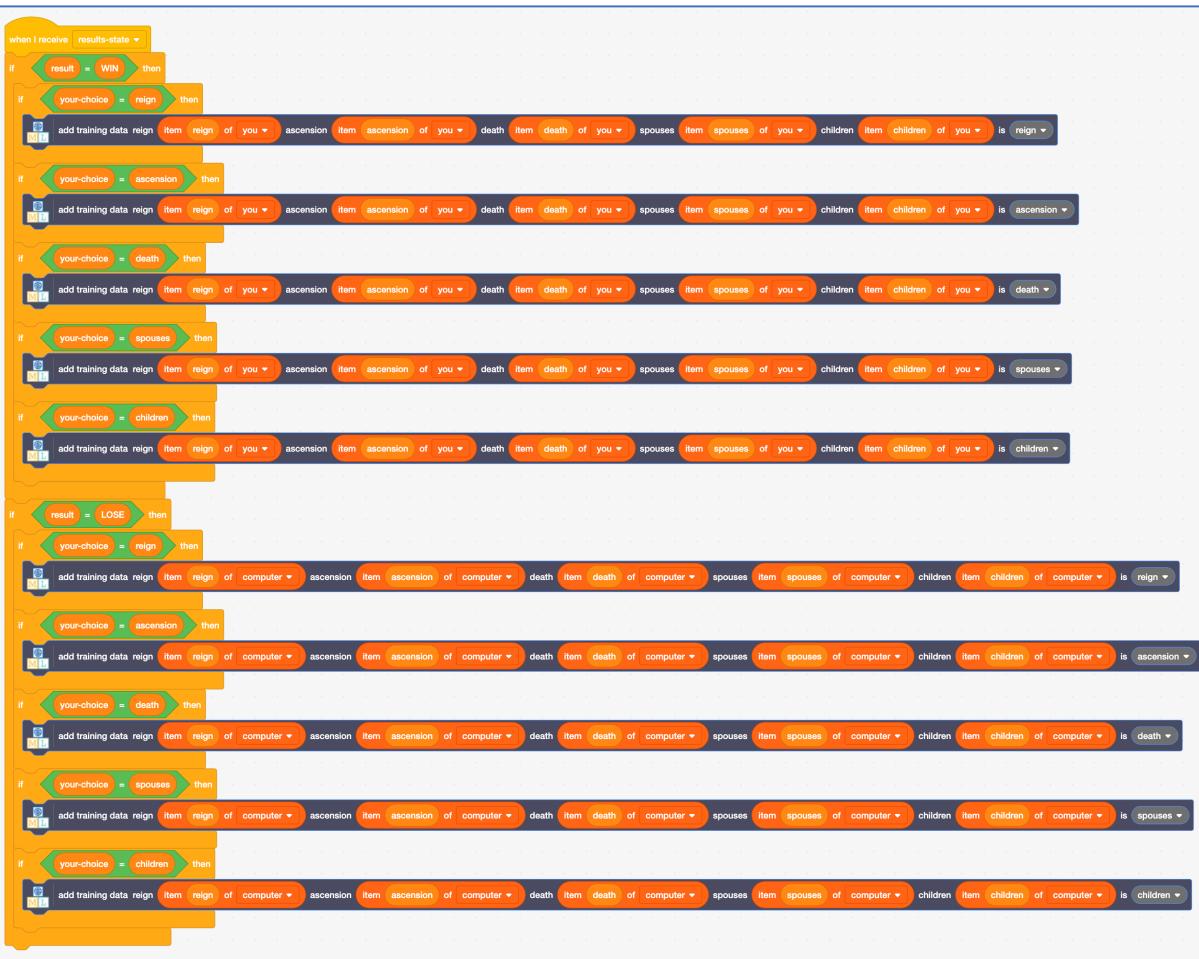


62. Modify the duplicate again, so that the computer’s moves can be added to the training examples when the computer scores a point.

Change the “you” to “computer” (to get values from the computer card) for all five “add training data” blocks



63. Join it all up



What have you done?

You've made a Top Trumps bot that can learn by playing the game. This means you don't need to wait for the computer to have learned before it can start playing.

It can start playing (even if it loses a lot at first), straight away. And by playing the game, it will learn from those experiences how to get better.

You haven't told the computer what to do, but allowed it to try out different choices and discover what choices are more likely to help it to win.

An example of training a Top Trumps bot

Your results will be different to this.

But these were the results I got from training my bot.

	Score	
	Human	Computer
No training – computer choosing at random	72	28
Trained with 100 examples	47	53
Trained with 200 examples	38	62
Trained with 300 examples	29	71
Trained with 400 examples	25	75
Trained with 500 examples	27	73
Trained with 600 examples	29	71
Trained with 700 examples	27	73

In general, more training is better.

*There were times where the computer did worse after more training.
Why do you think that was?*

*After a certain point, the computer's scores stopped improving, even
after I kept adding more and more training.*

Why do you think that was?

*Compare these results with the results from your bot. How has your
bot learned from the training you've given it?*