

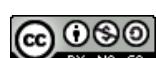
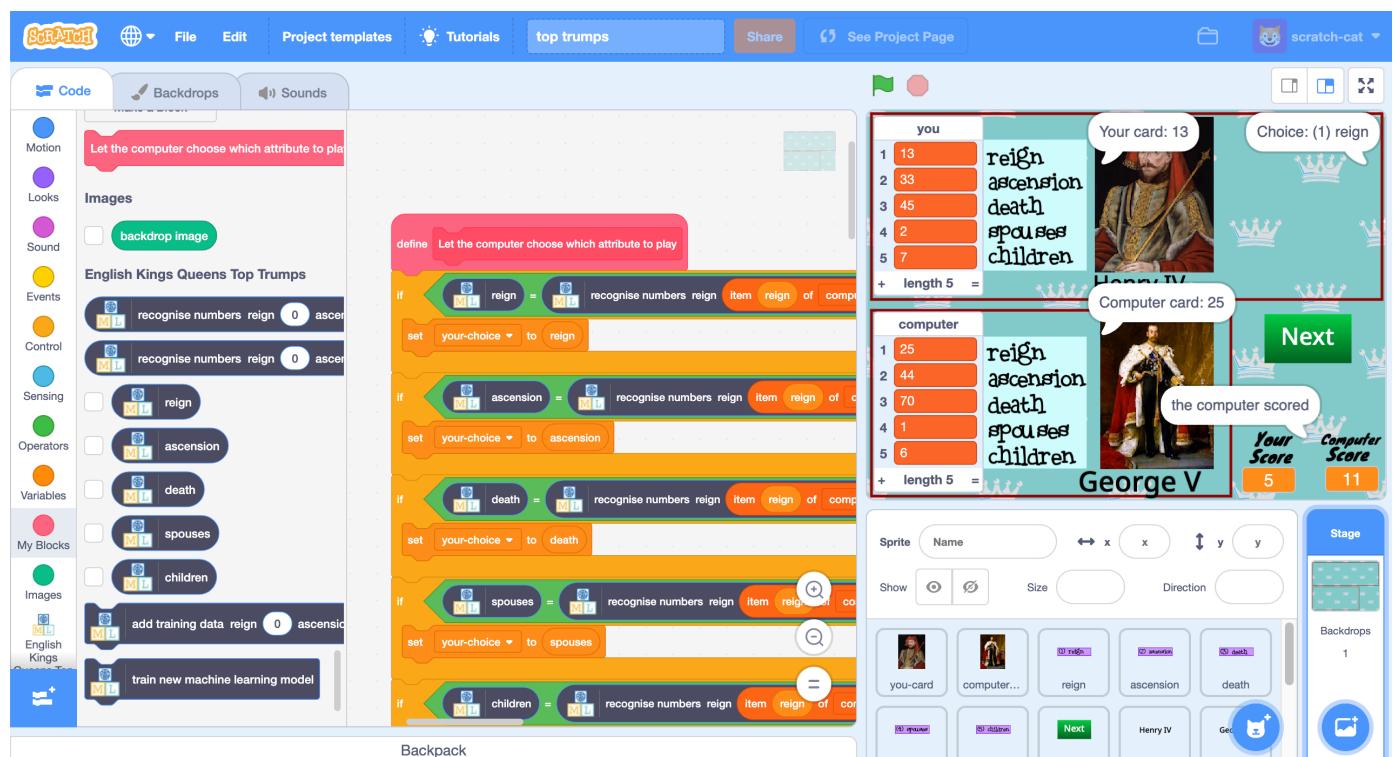
# Top Trumps

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

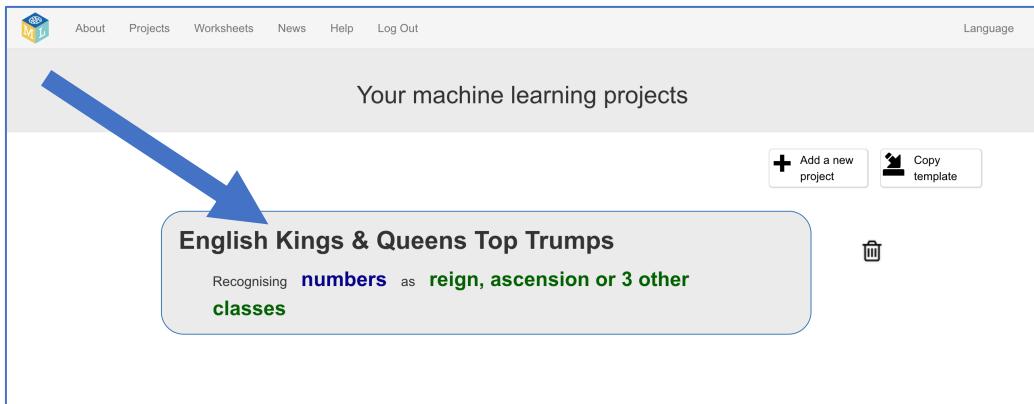
4) He had **1** spouse

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

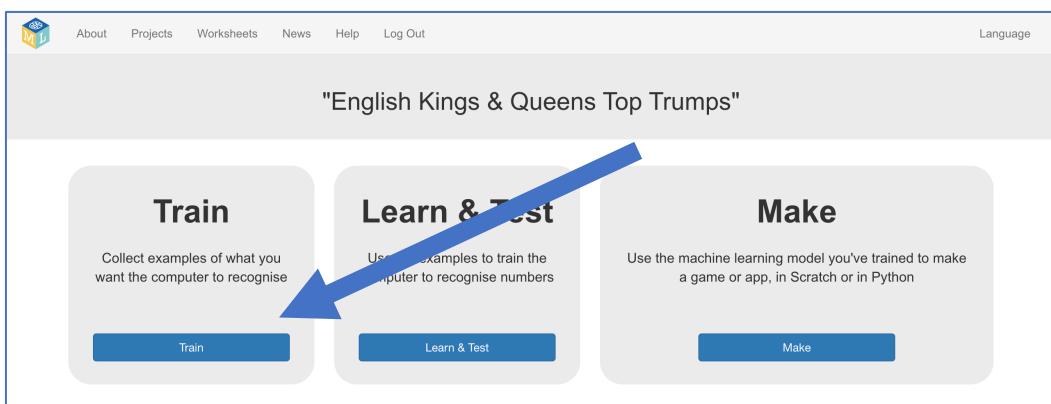
5) He had **9** children

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 “English Kings & Queens Top Trumps” in the projects list.



8. Click the “Train” button



- 9.** The project template prepares training buckets to store values from Top Trumps cards.

The screenshot shows a web-based machine learning project template. 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 box has a label at the top: 'reign', 'ascension', 'death' in the top row, and 'spouses', 'children' in the bottom row. Each box also has a small button labeled '+ Add example' at the 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

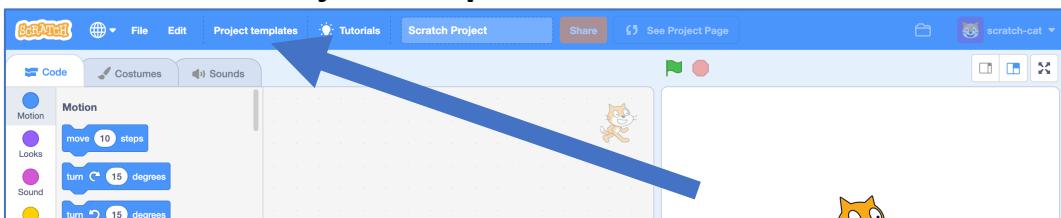
- 12.** Click the “Scratch 3” button

- 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.*

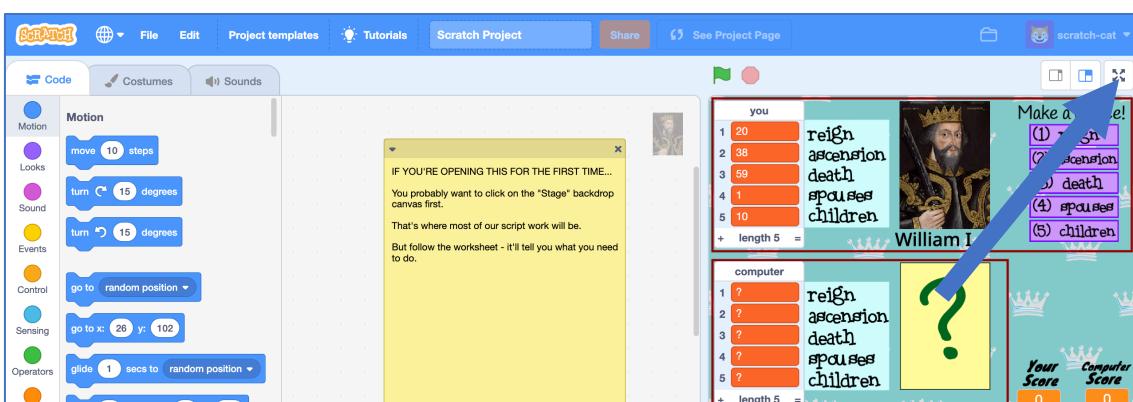
The screenshot shows a page titled 'Using machine learning in Scratch 3'. At the top, there's a navigation bar with links for 'About', 'Projects', 'Worksheets', 'News', 'Help', 'Log Out', and 'Language'. Below the navigation, a message says 'You haven't trained a machine learning model yet.' A blue arrow points to a button labeled 'straight into Scratch' which is part of a larger callout box. The callout box contains two sections: one about adding blocks to Scratch and another showing a screenshot of the Scratch 3 interface. The Scratch interface screenshot shows the 'Code' tab selected, with a script block visible: 'when green flag clicked [recognise numbers distance 0 (label)].' Below the interface, there's a note: 'Put numbers in the input for this, and it will return the label that your machine learning model recognises it as.'

## 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.**

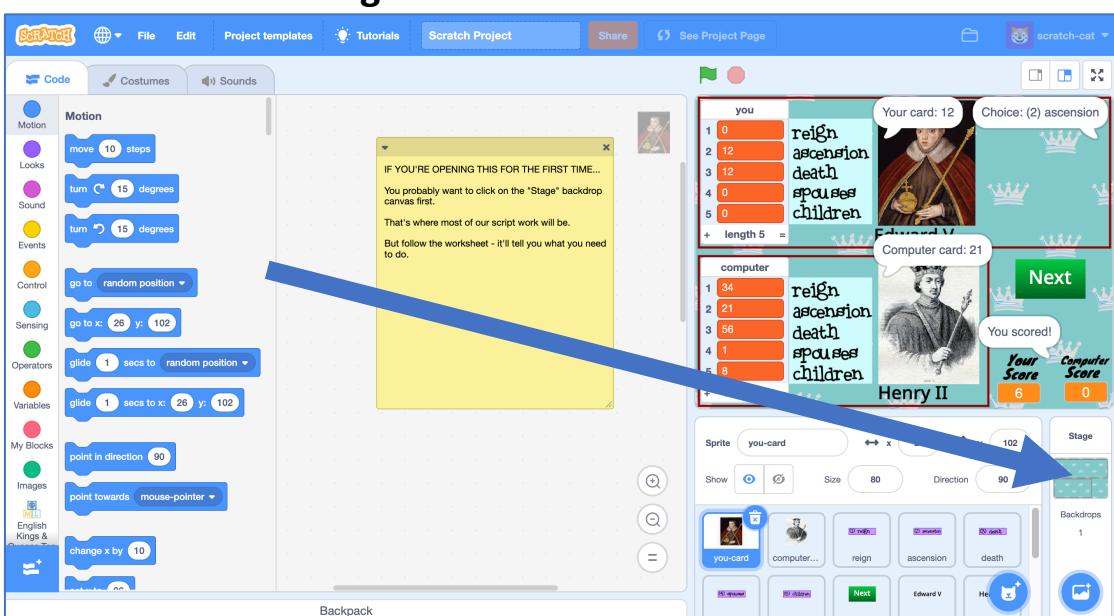
**18.** Play a few rounds of the game against the computer.

*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 script 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 script 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.

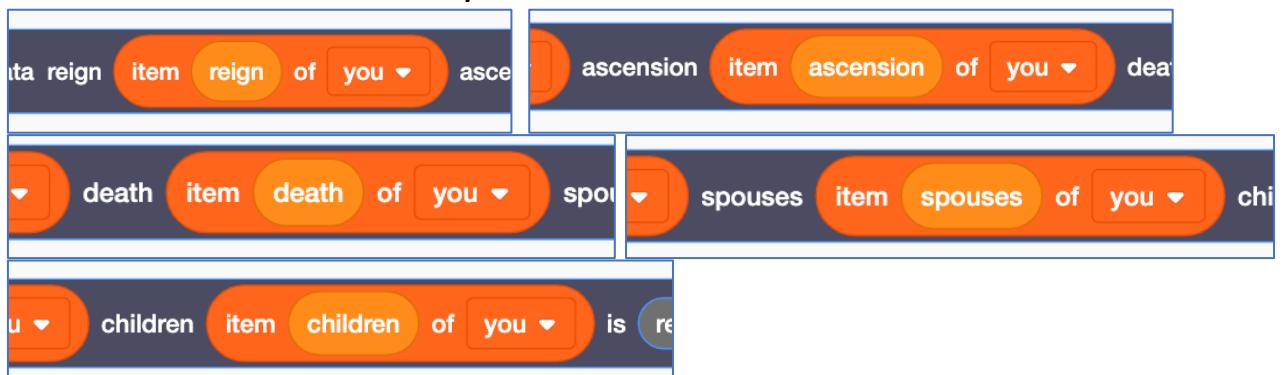
But people don't play like that. We learn how to choose which value would give us the best chance of winning. We do this based on the cards we've seen before, and on our understanding of the rules.

Next, you'll create a Scratch script that collects training examples using the moves that you make.

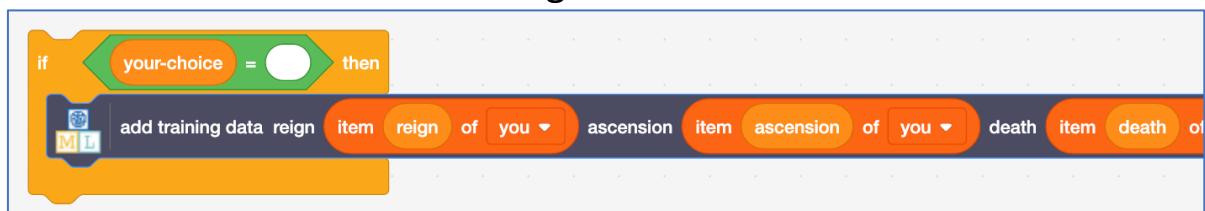
**23.** Still on the **Stage**, drag the “add training data” block to the canvas



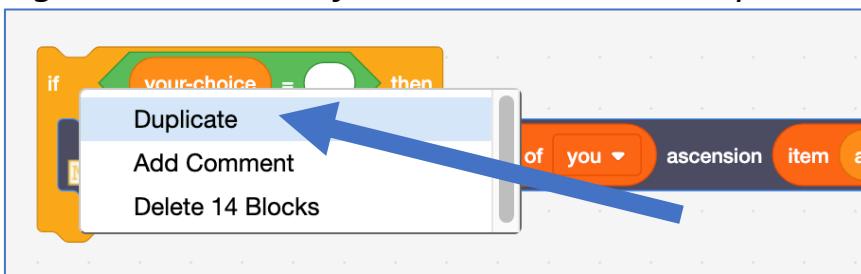
**24.** Add the values from your card to the block



**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  
*“if” choices should use orange options from “Data”*



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



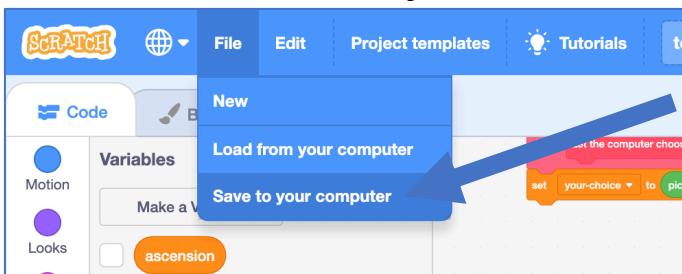
**29.** Finish the script so that this is called every time you win a hand  
*The values that were on your card, and the choice you made, will be added to the training examples every time you win.*



**30.** Click the full-screen button and green flag again.  
Play games until your score reaches **10**.

**31.** Save your project

*Click “File” -> “Save Project”*



**32.** Leave the Scratch window open.  
In the training window, click the “< Back to project” link.

### 33. Click the “Train” button

### 34. 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.*

Recognising **numbers** as **reign, ascension or 3 other classes**

Language

< Back to project

+ Add new label

**reign**

reign 18  
ascension 23  
death 43  
spouses 1  
children 4

reign 17  
ascension 31  
death 48  
spouses 2  
children 5

reign 0  
ascension 0  
death 0  
spouses 0  
children 0

reign 0  
ascension 12  
death 12  
spouses 12  
children 0

reign 0  
ascension 12  
death 12  
spouses 0  
children 0

+ Add example

**ascension**

reign 9  
ascension 31  
death 41  
spouses 1  
children 0

+ Add example

**death**

reign 12  
ascension 54  
death 67  
spouses 1  
children 2

reign 8  
ascension 31  
death 41  
spouses 1  
children 0

reign 24  
ascension 50  
death 54  
spouses 1  
children 3

reign 0  
ascension 41  
death 77  
spouses 1  
children 0

+ Add example

**spouses**

+ Add example

**children**

reign 20  
ascension 38  
death 59  
spouses 1  
children 10

reign 34  
ascension 21  
death 56  
spouses 1  
children 8

reign 21  
ascension 18  
death 40  
spouses 1  
children 10

+ Add example

3

5

1

4

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

### 36. The training page won’t let you train a model yet

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

Machine learning models

Language

< Back to project

**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.

### 37. Leave the page open.

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

**38.** 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 a web page titled "Machine learning models". At the top left is a link "[< Back to project](#)". Below it are two sections: "What have you done?" and "What's next?". The "What have you done?" section contains text about collected examples and a list of items. The "What's next?" section contains text about starting training and a "Train new machine learning model" button. A blue arrow points from the text "Ready to start the computer's training?" towards the "Train new machine learning model" button. At the bottom left of the main content area, there is a small box labeled "Info from training computer:" containing the text "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.

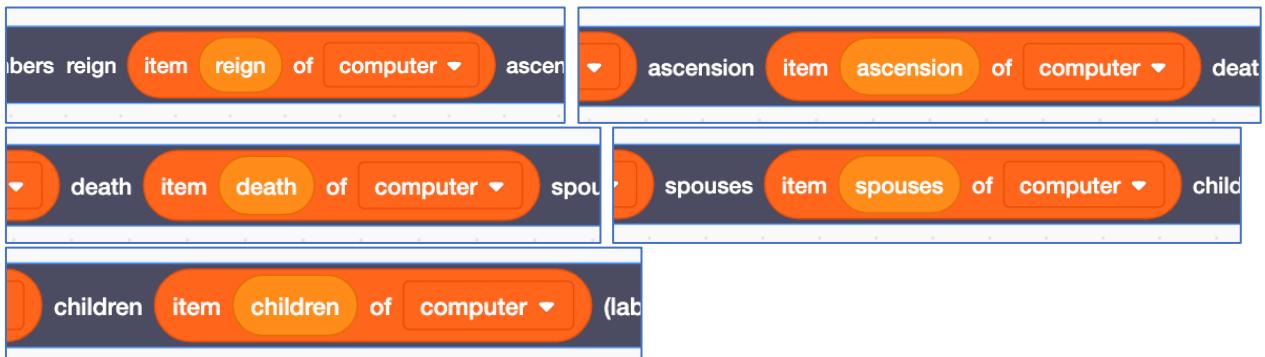
**39.** Go back to the Scratch window

**40.** Click on the “**Stage**” to get to where you added the scripts before

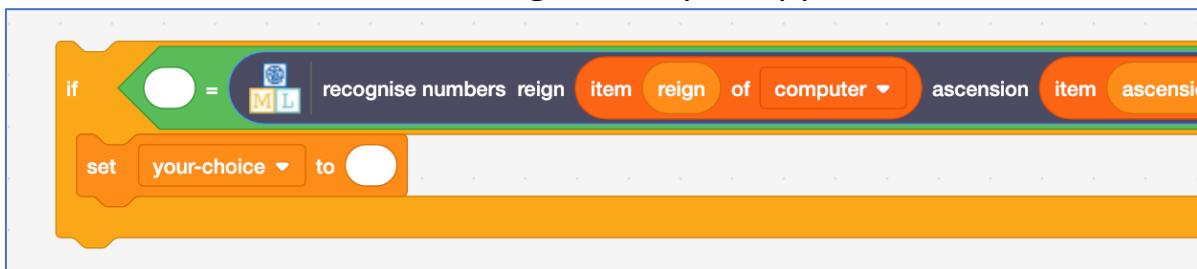
**41.** Add the “recognise numbers ... (label)” block



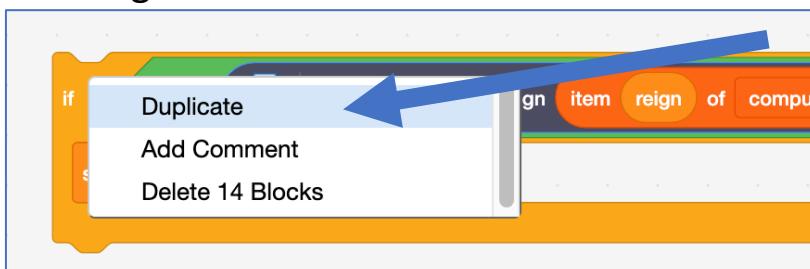
**42.** Add the values from the computer’s card to the block



**43.** Add it to an “if” block to get a script snippet that looks like this:

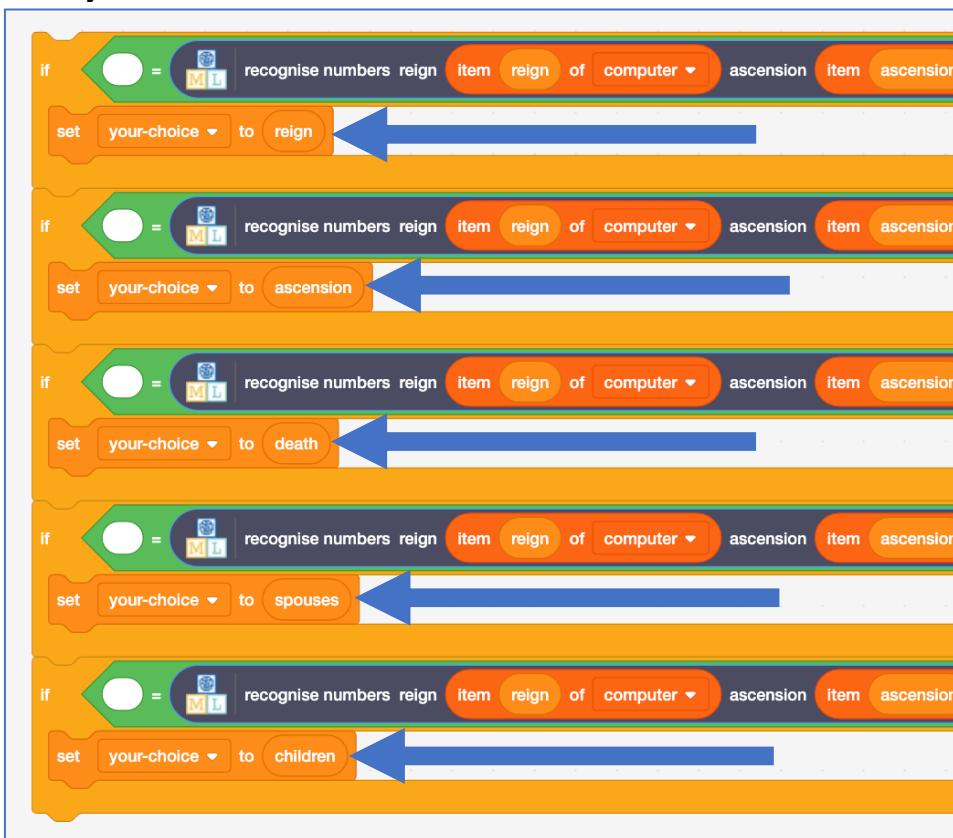


**44.** Right-click on the “if” block and click on “**Duplicate**”

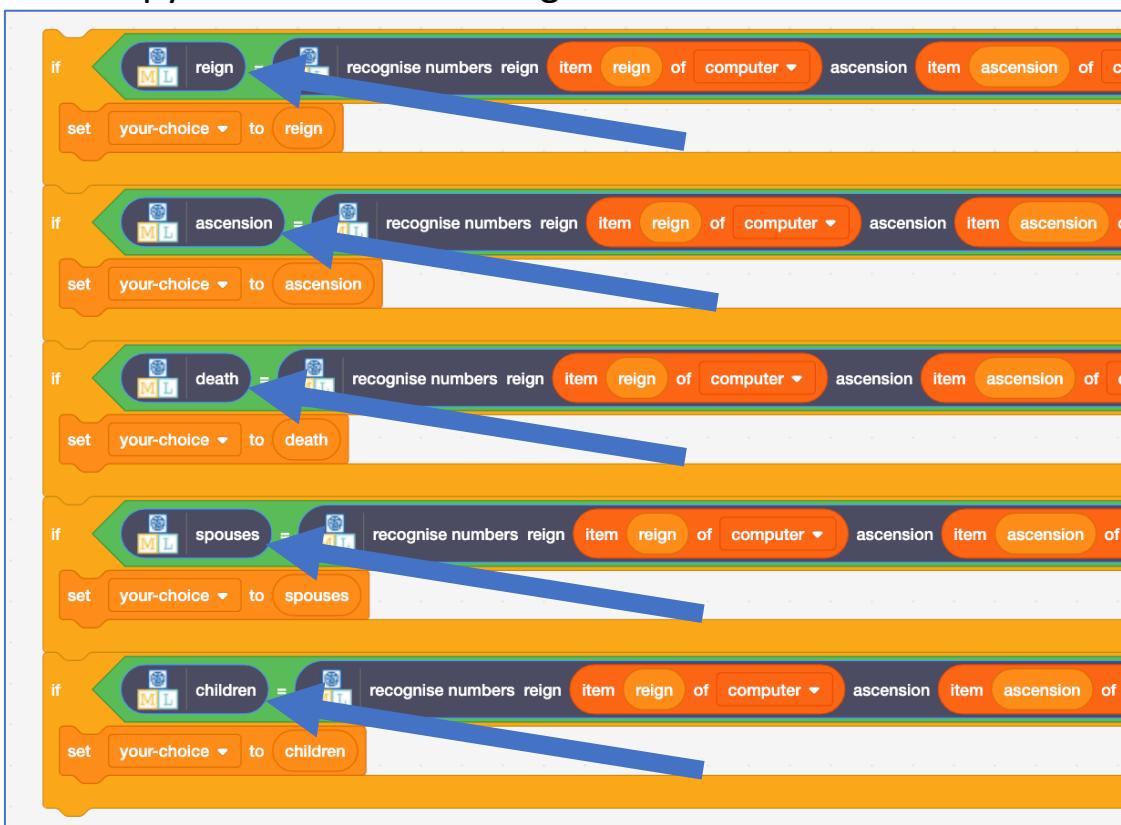


**45.** Repeat the duplicate until you have **5** copies of the if block, joined together.

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



**47.** Copy the machine learning bucket blocks into the “if” blocks



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



**49.** Save your project

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

**50.** 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.

**51.** Go back to the training tool window

**52.** Click the “[< Back to project](#)” link, then back to “[Learn & Test](#)”

**53.** Click the “[Train new machine learning model](#)” button

The screenshot shows the Scratch machine learning training tool interface. At the top left is a link to "Back to project". Below it are two main sections: "What have you done?" and "What's next?".

**What have you done?**

- You have trained a machine learning model to recognise when numbers are reign, ascension or 3 other classes.
- You created the model on Sunday, March 15, 2020 1:49 AM.
- You have collected:
  - 18 examples of reign,
  - 7 examples of ascension,
  - 10 examples of death,
  - 4 examples of spouses,
  - 11 examples of children

**What's next?**

- 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 recognises it as, and how confident it is in that.
- If the computer seems to have learned to recognise things correctly, then you can go to Scratch and use what the computer has learned to make a game!
- If the computer is getting too many things wrong, you might want to go back to the [Train](#) page and collect some more examples.
- Once you've done that, click on the button below to train a new machine learning model and see what difference the extra examples will make!

Below these sections is a test area where you can enter numbers to see the model's recognition. At the bottom, there is an "Info from training computer:" section showing the model started training at Sunday, March 15, 2020 1:49 AM and is currently available. It includes a "Delete this model" button and a "Train new machine learning model" button, which is highlighted with a large blue arrow pointing to it.

## 54. 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**”

## 55. Play the game again.

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

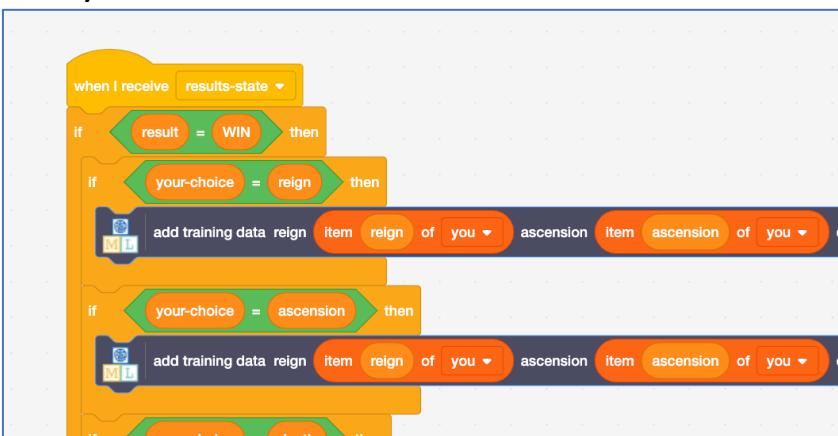
## 56. Repeat steps 51 – 55 to collect more examples, and then train a new machine learning model with them. Do this a few times.

### 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.

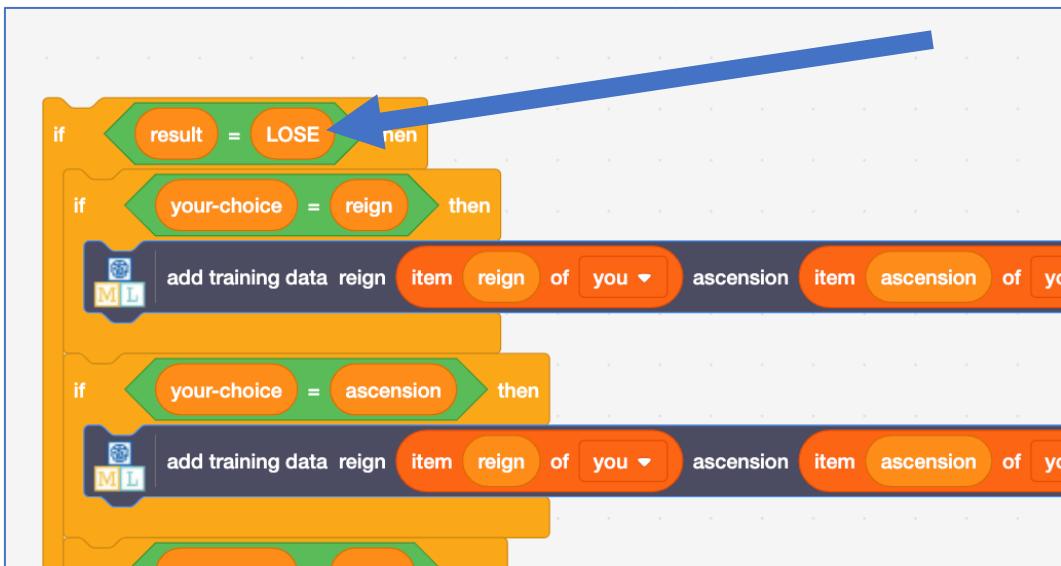
## 57. Open the game in Scratch

## 58. Find the “When I receive ‘results-state’” script in the Stage that you made earlier



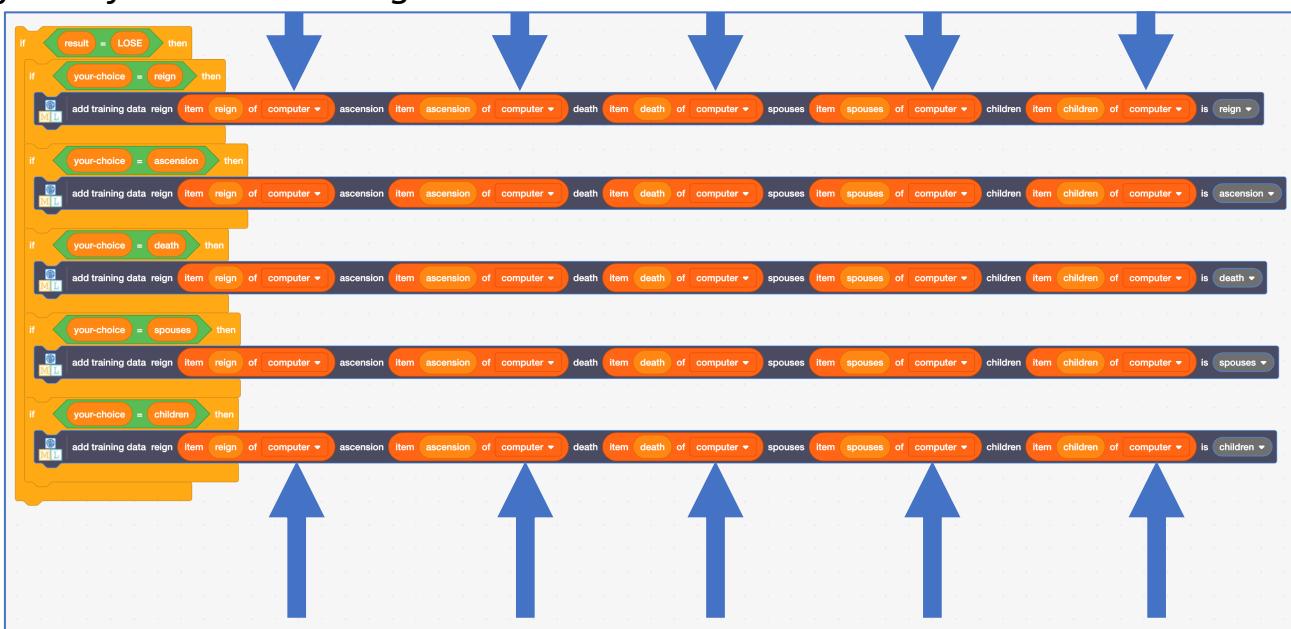
## 59. Duplicate the “if result = WIN” block

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

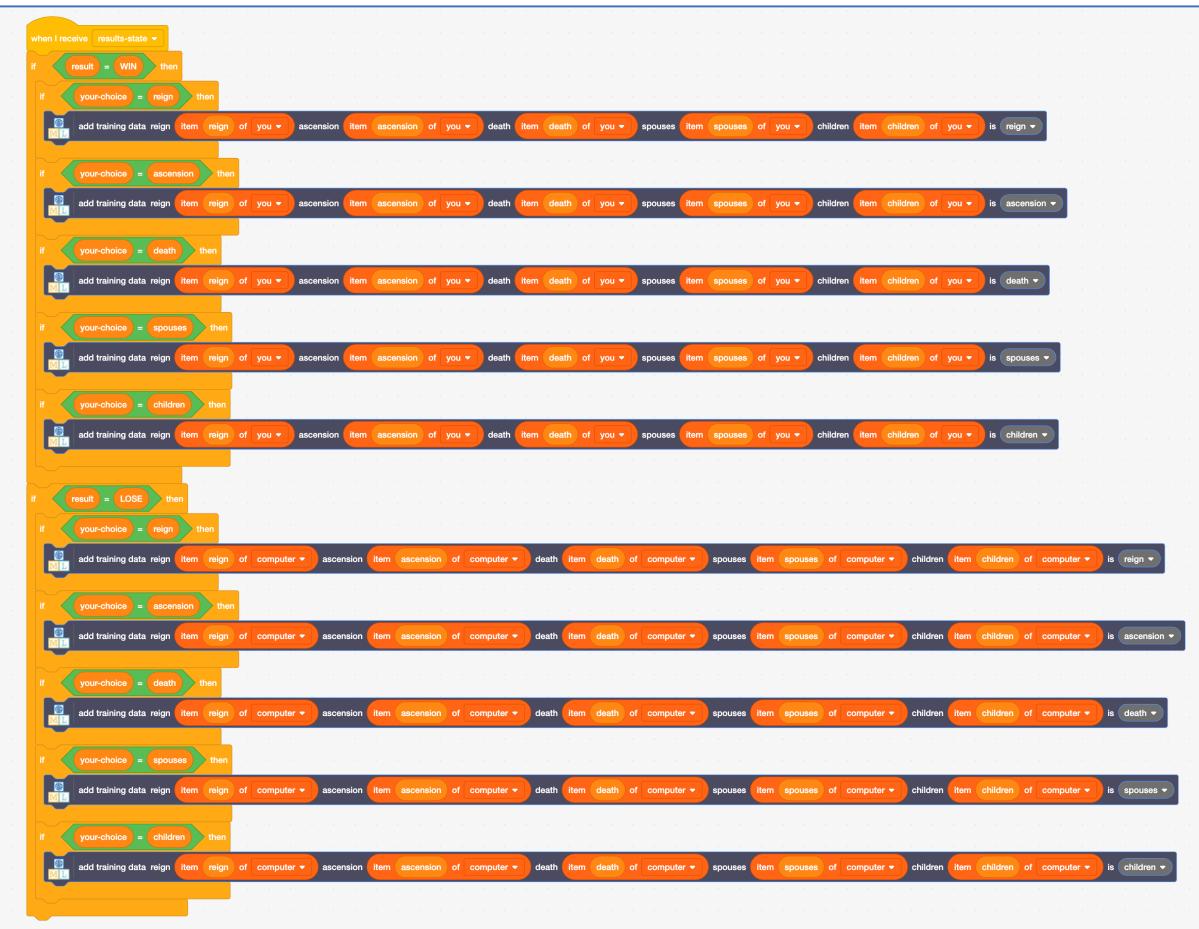


## 61. Modify the duplicate again, so that the computer's moves can be added to the training examples when the computer scores a point.

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



## 62. 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.

This is called "reinforcement learning". When it makes a good choice, this is reinforced by the computer being told that it has won.

## 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?*