Tourist Info

In this project you will make a mobile Tourist Information bot that makes recommendations to tourists about which attractions they should visit.

You'll train a machine learning model so the bot can learn to make recommendations based on what people say they're interested in.

You'll also learn about the affect of "bias" on machine learning projects, and see how this can happen.

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

2. Click on "**Get started**"

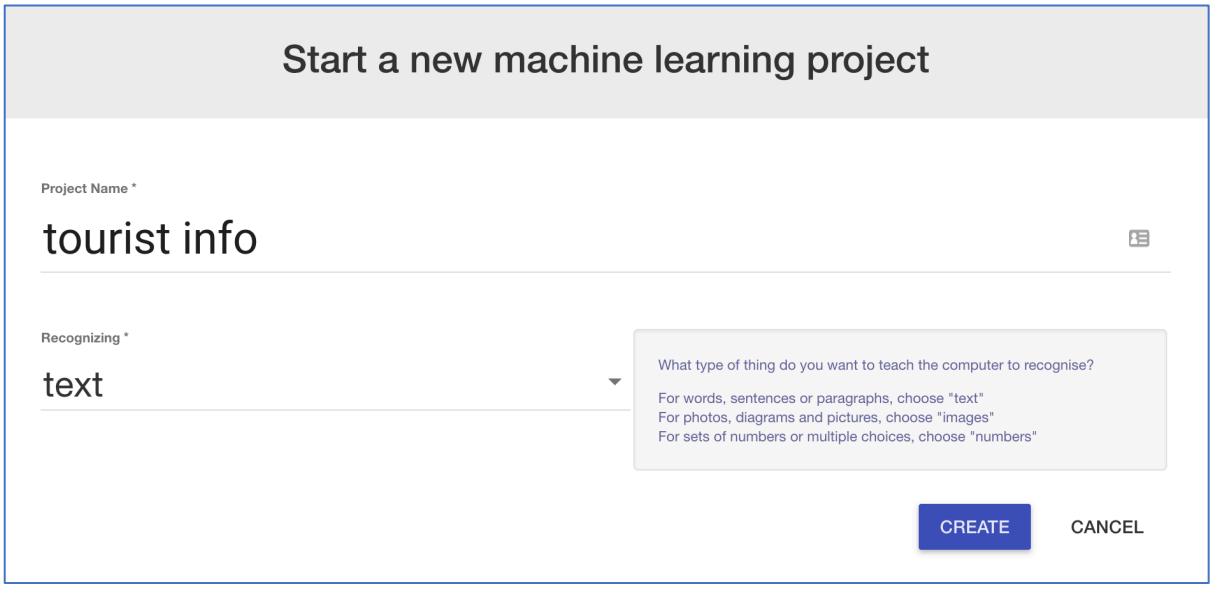
3. Click on "**Log In**" and type in your username and password

If you don't have a username, ask your teacher or group leader.

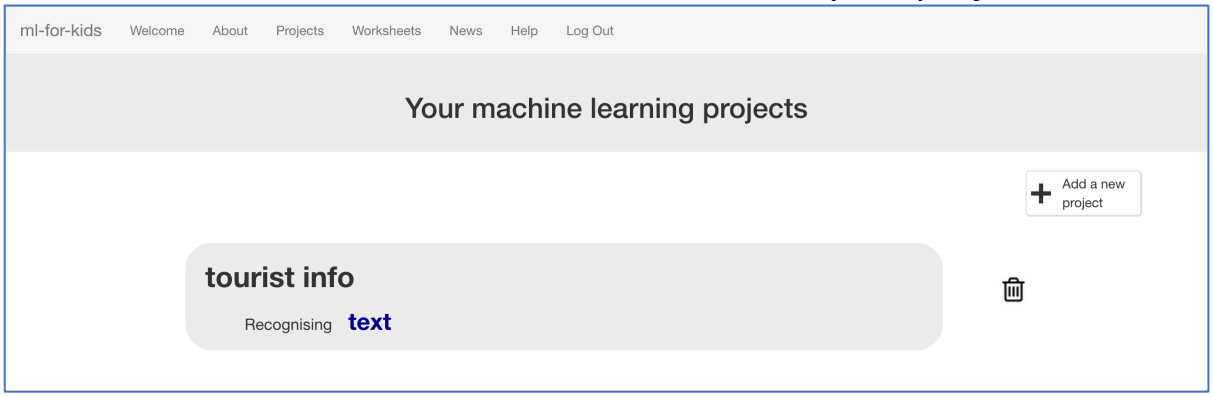
4. Click on "**Projects**" on the top menu bar

5. Click the **"+ Add a new project**" button.

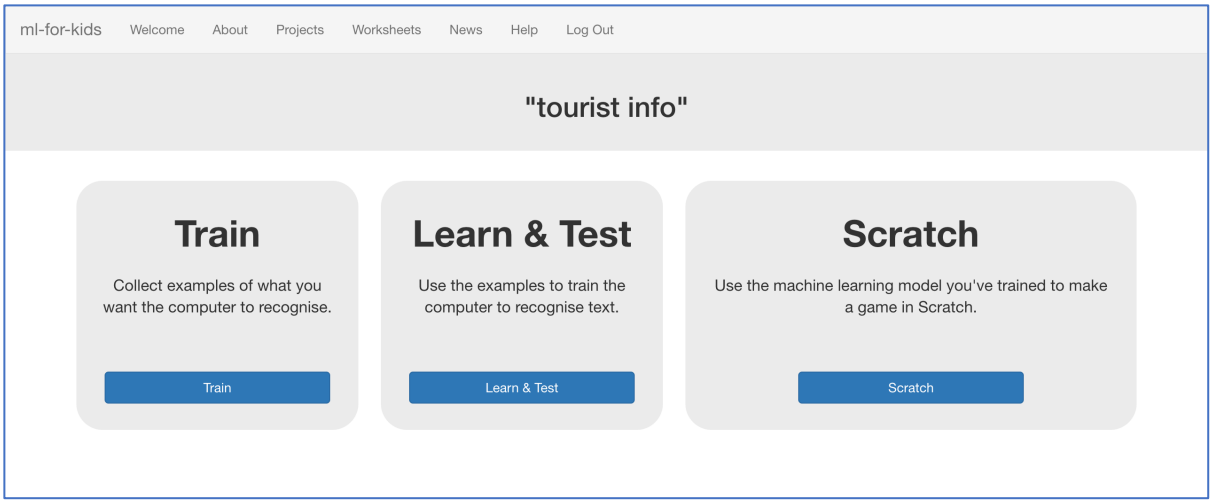
6. Name your project "tourist info". Set it to learn how to recognise "**text**". Click "**Create**"



7. You should see "**tourist info**" in the list of your projects. Click on it.



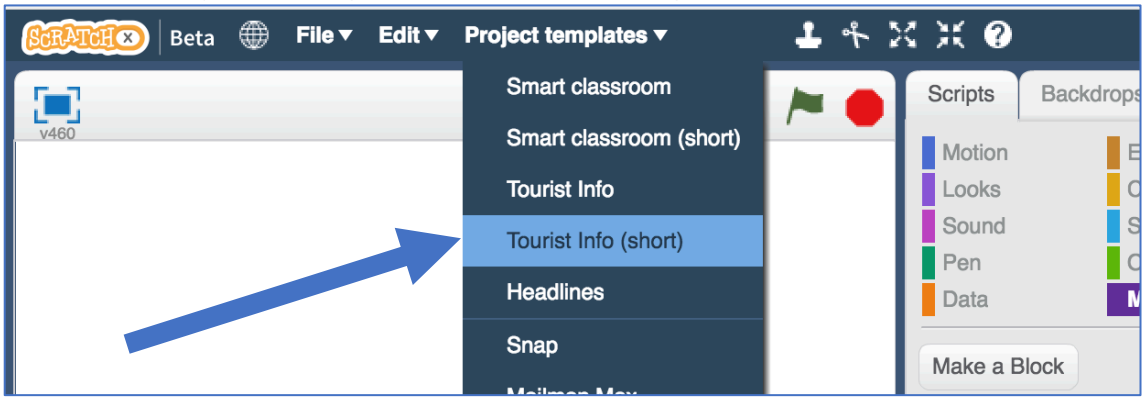
8. Start by getting a project ready in Scratch. Click the **Scratch** button.



9. The next page will warn you that you haven't done any machine learning yet. Click on **Scratch by itself** to launch Scratch anyway.

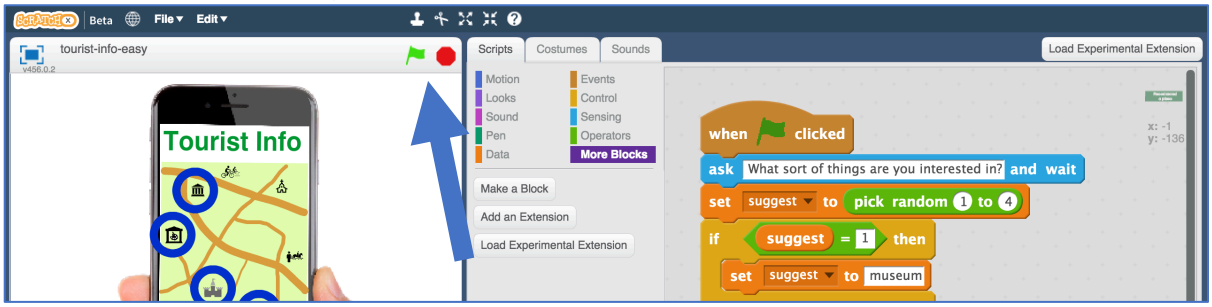
10. Open the **Tourist Info (short)** template project

*Click on **Project templates** -> **Tourist Info (short)***



11. Click the green flag to try it out

When it asks what you're interested in, type in something about what you like to do on holiday, and press Enter.



12. Try it a few times. Can you tell how the app is choosing what to recommend to you?
Have a look at the script on the "recommend" sprite. Can you see how it's choosing holiday destinations? Ask your group leader if you're not sure.

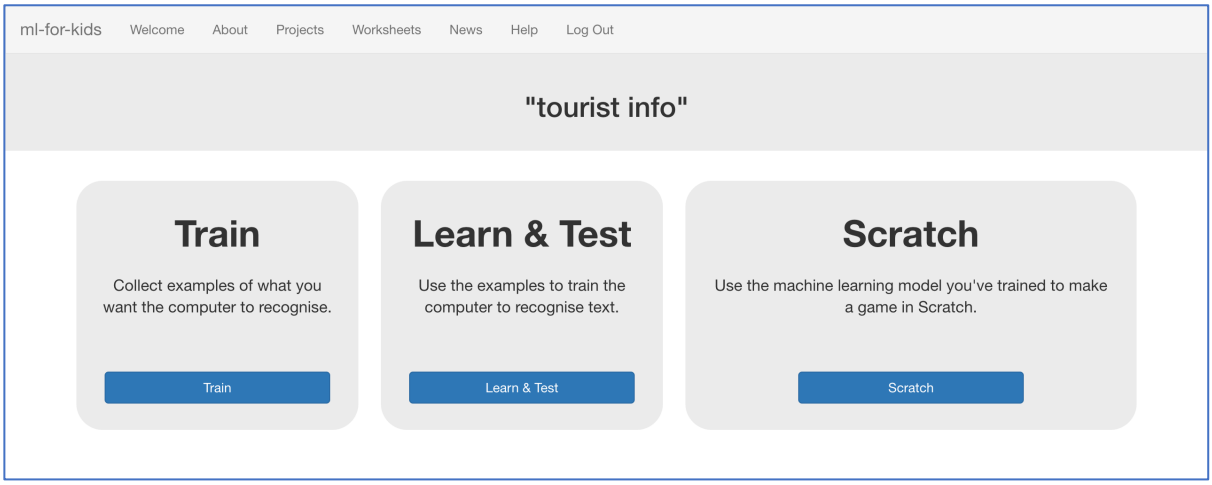
13. Close the Scratch window

What have you done so far?

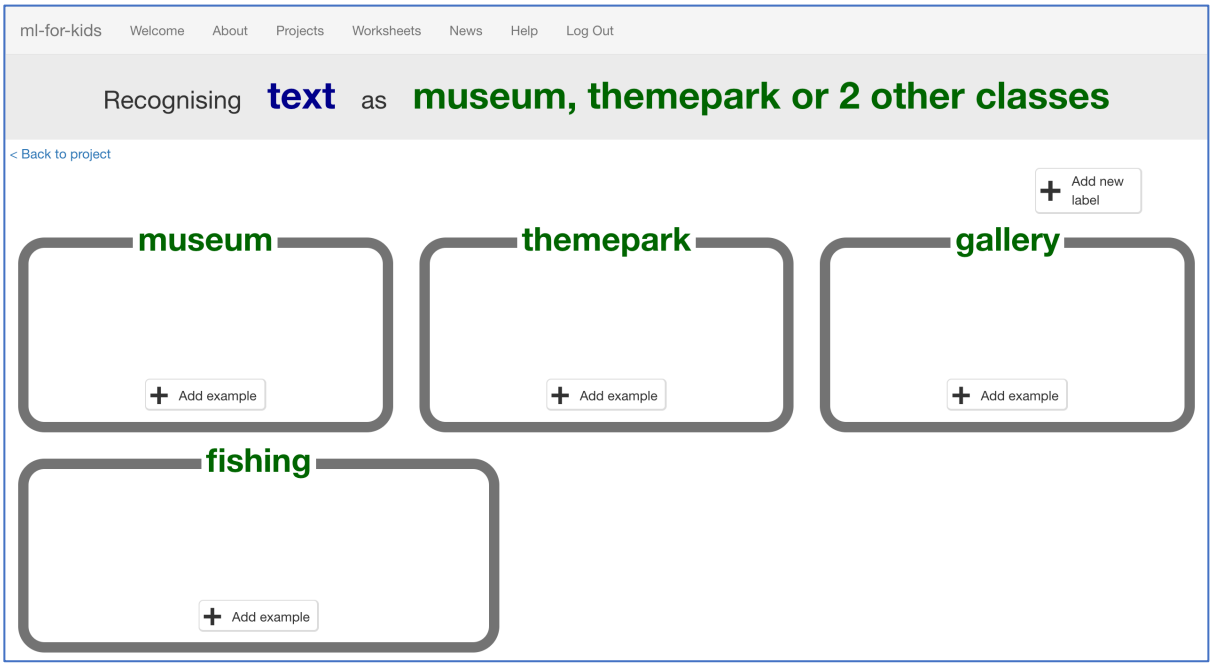
This is a mobile Tourist Information bot that will make recommendations to visitors to your town. It will ask them what they're interested in, so it can make the best recommendation. But for now, is choosing something at random to recommend.

You need to train it to be able to make recommendations so you can use machine learning in your bot.

14. We need examples to train the computer. Click the **"< Back to project**" link. Then click the **Train** button.



15. Click on **"+ Add new label**" and call it "museum".
Do that again, and create a second bucket called "themepark".
Create a third bucket called "gallery" and a fourth called "fishing".



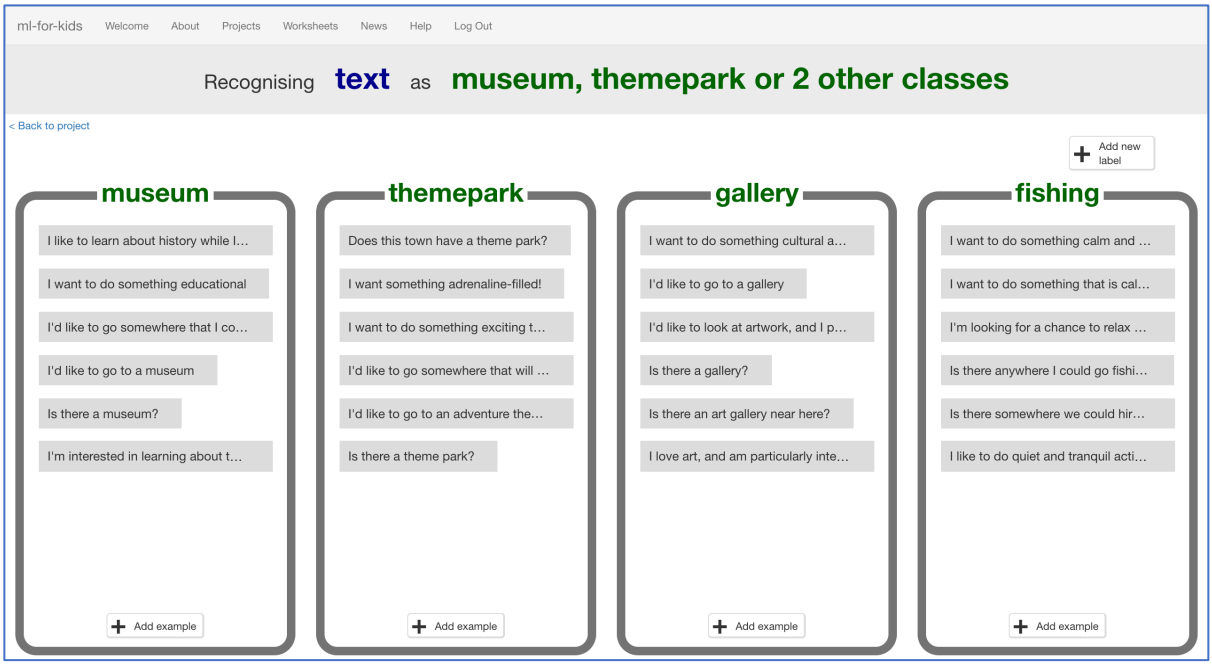
16. Click the "**Add example**" button in the "museum" bucket, and type in something a tourist who would like a museum might say.
For example: "I like to learn about history while I'm on holiday!"

17. Click the "**Add example**" button in the "themepark" bucket, and type in something a tourist who would like theme parks might say.
For example: "I want to do something exciting that gets my heart going"

18. Click the "**Add example**" button in the "gallery" bucket, and type in something a tourist who would like galleries might say.
For example: "I want to do something cultural and I enjoy art"

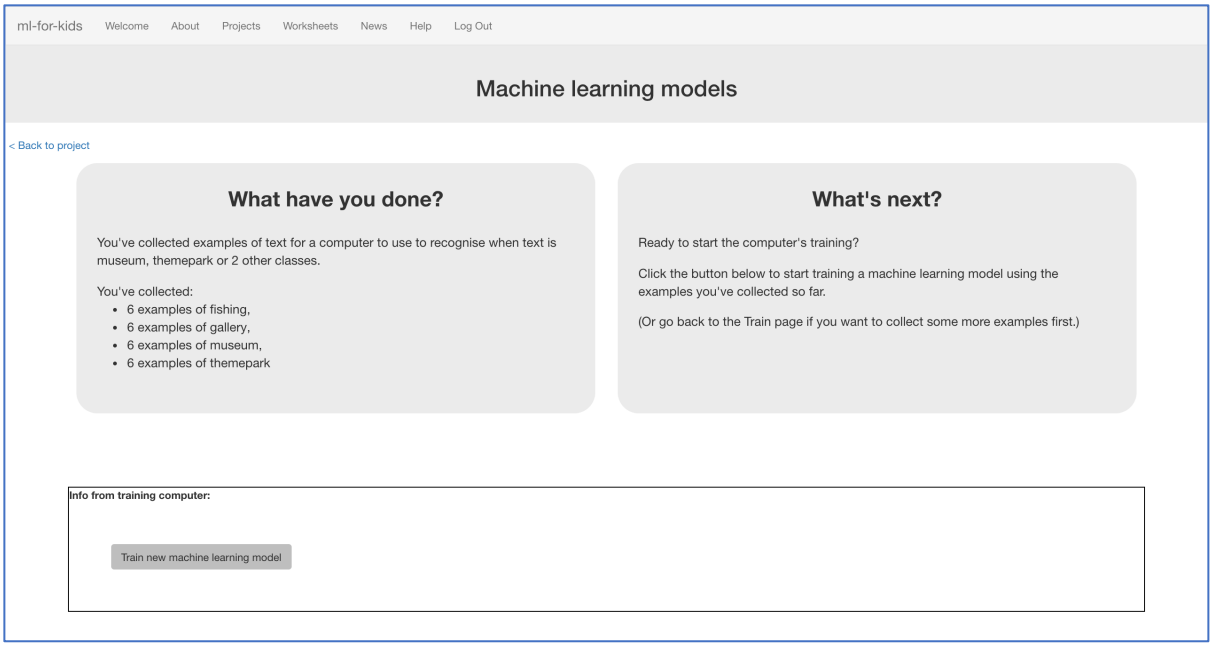
19. Click the "**Add example**" button in the "fishing" bucket, and type in something a tourist who would like fishing might say.
For example: "I'm looking for a chance to relax and I'd like to do something quiet"

20. Repeat steps 16 – 19 until you've written **five** examples of each.



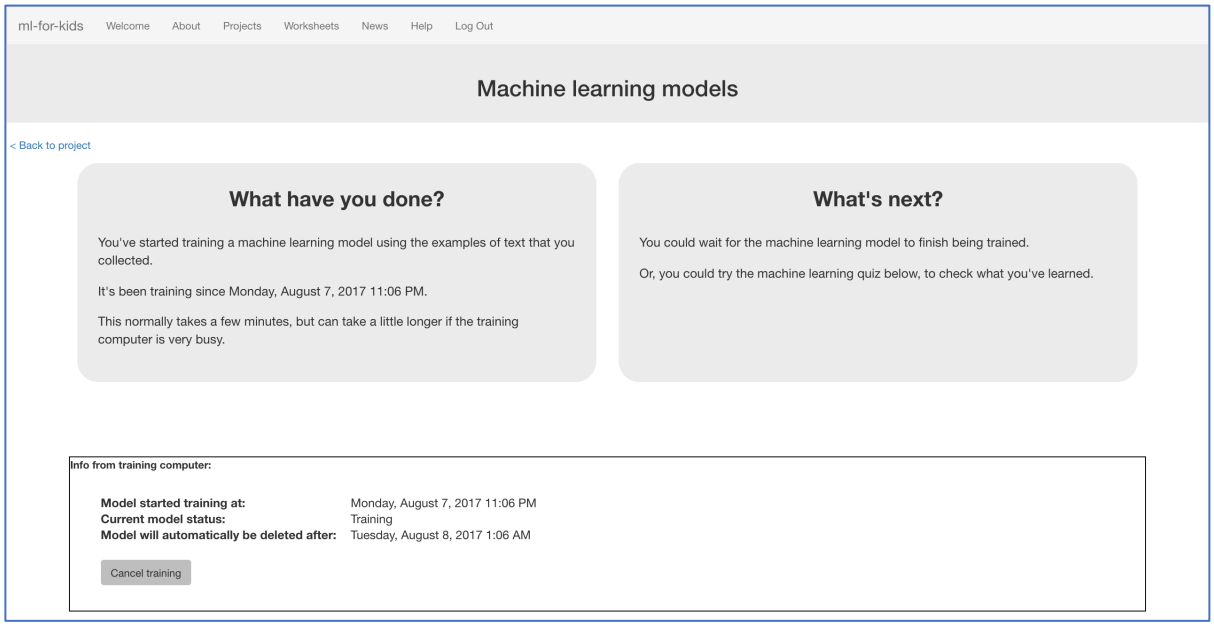
21. Click on the **"< Back to project**" link.
Then click on the "**Learn & Test**" button.

22. 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 messages from the examples you've given to it.



23. Wait for the training to complete. This might take a few minutes.

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



24. Once the training has completed, a Test box will be displayed.

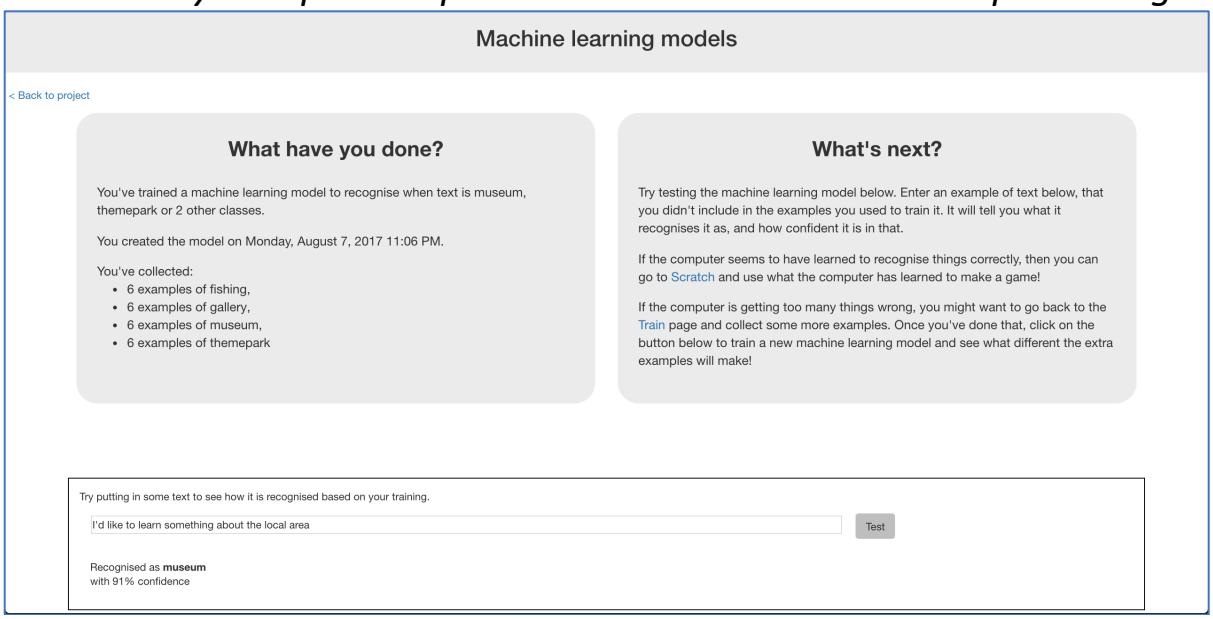
Test your machine learning model to see what the computer has learned.

Type a request from an imaginary tourist and see what it recommends.

Test it with examples that you haven't shown the computer before.

If you're not happy with how the computer makes recommendations, go back to step 20, and add some more examples.

Make sure you repeat step 22 to train with the new examples though!



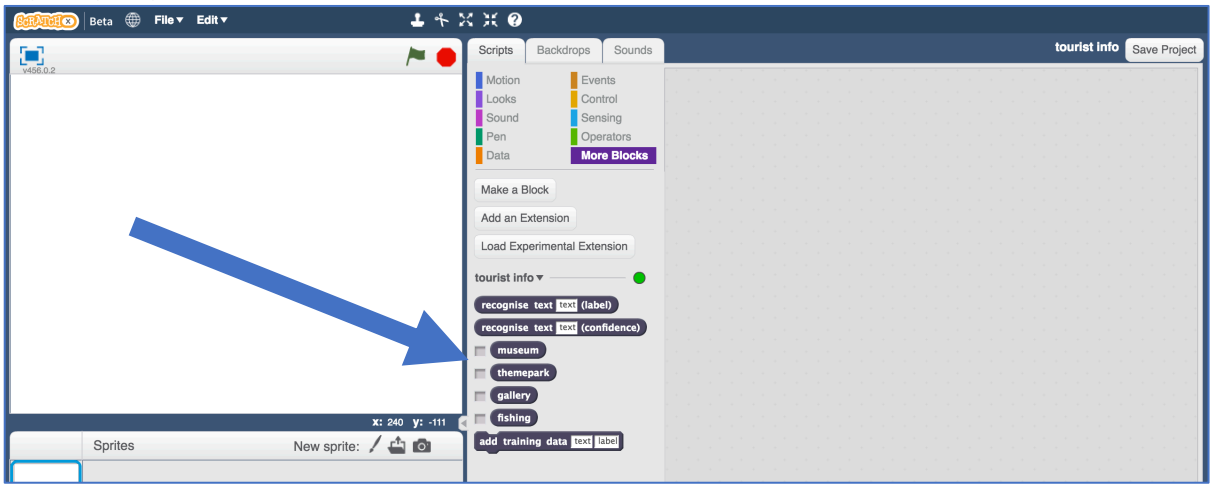
What have you done so far?

You've started to train a computer to recognise text so you can make personalised recommendations. 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 computer will learn from patterns in the examples you've given it, such as the choice of words, and the way sentences are structured. These will be used to be able to decide which place to recommend.

25. Click the **"< Back to project"** link, then click the "**Scratch**" button.
This page has instructions on how to use the new blocks in Scratch. Keep the page open if you need to check back on how to use them.

26. Click on the "**Open in Scratch**" button to launch the Scratch editor.
You should see new blocks in the "More blocks" section from your "tourist info" project.



27. Load the Scratch project you opened before.
*Click on **Project templates** -> **Tourist Info (short)***

Tips

More examples!

The more examples you give it, the better the computer should get at recognising patterns in what tourists who like different places would say.

Get examples from other people

Try asking the people sat near you to suggest questions from tourists. The more people you get examples from, the better chance you have of making them varied.

Other people will think of ways to phrase the examples that you might not have.

The more the better!

Mix things up with your examples

Try to come up with lots of different types of examples.

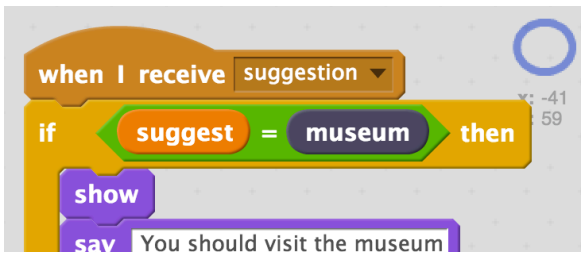
For example, make sure that you include some long examples and some very short ones.

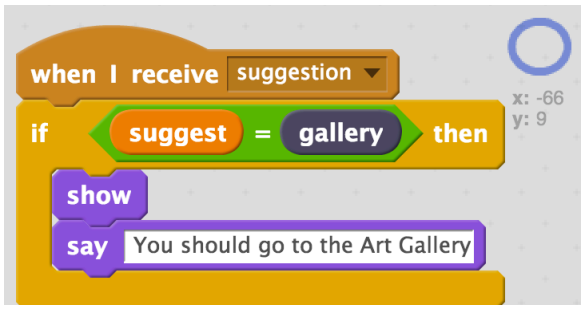
28. Click on the "**Scripts**" tab for the "recommend" button sprite, and update the script to use your machine learning model instead of the random choice.

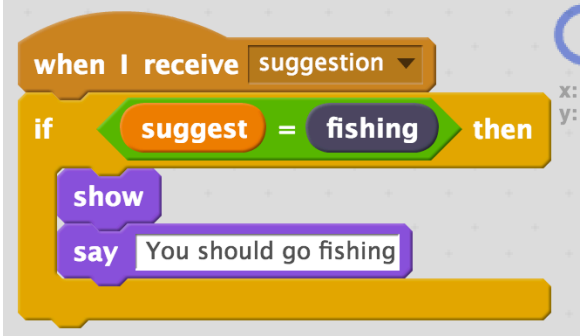
The "recognise text … (label)" block is a new block added by your project.

29. Click on the "Scripts" tab for each of the circle sprites, and update the **if** script block for each to use one of your new blocks

They should end up looking something like this

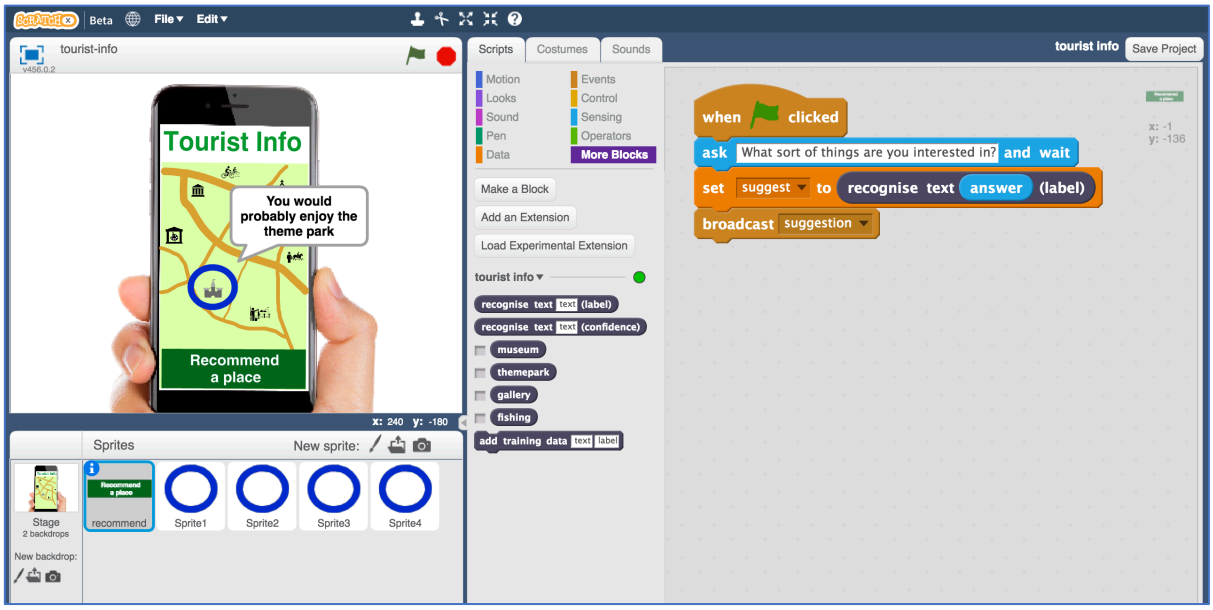






30. Test your project by clicking the **Green Flag**

Type a request from an imaginary tourist and press enter
It should recommend somewhere appropriate for them to visit
This should work for messages that you didn't include in your training.



31. Save your project.

*Click **File** -> **Save Project***

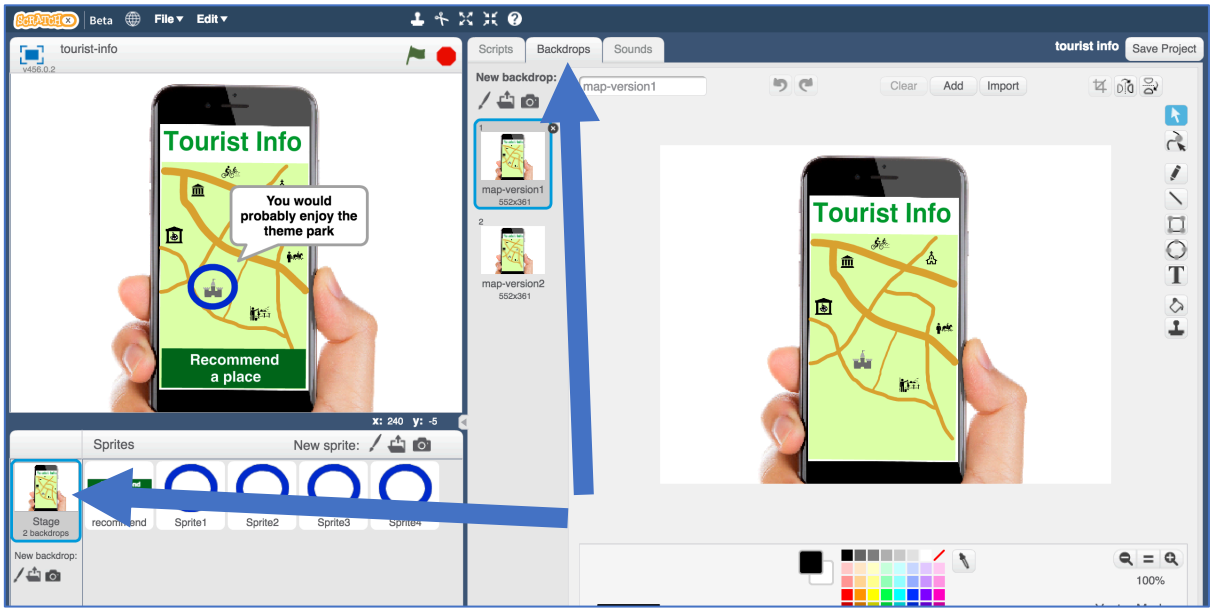
What have you done so far?

You've modified your Tourist Info bot to make recommendations using machine learning instead of your earlier random choices.

If you'd trained it with examples of requests from real tourists, instead of making them up, this is the sort of thing that would be advertised as:

"An artificial intelligence that helps answer tourists' questions and learns how to make recommendations based on their interests"

32. Click on the **Stage** and then click on **Backdrops**



33. Switch the backdrop to use **map-version2**

Can you see what's different?

A new fun-fair has arrived in town!

A **new funfair** has opened in town, and the owner of the funfair wants your Tourist Info bot to send tourists to their new attraction.

They're offering to **pay you a lot of money to train your bot** to make sure this happens.

---

34. Save your Scratch project

35. Close the Scratch window

36. Go back to the "**Train**" page

*Click the "**< Back to project**" link and then click the **Train** button*

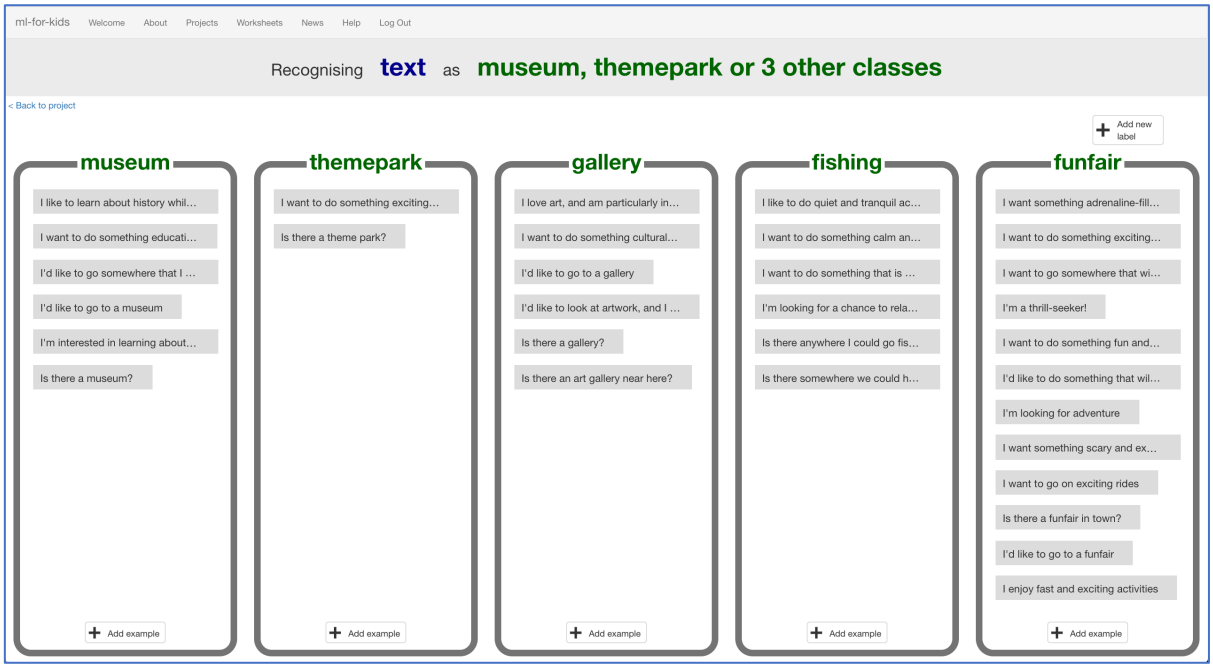
37. Add a new bucket for "funfair"

*Click the "**Add new label**" button. Call the new label "funfair"*

38. Add a lot of examples to the "funfair" bucket

Use the examples you had in the "themepark" bucket and then delete them from the themepark bucket afterwards. But leave 1 or 2 examples in the themepark bucket so it's not empty.

Then add a lot more new examples to the funfair bucket as well, so it has at least twice as many examples as any other attraction.



39. Train a new machine learning model with the new training

*Click the "**< Back to project**" link, then click the **Learn & Test** button.*

*Click the "**Train new machine learning model**" button*

It'll take a minute to re-train with the new examples.

40. Go back to Scratch

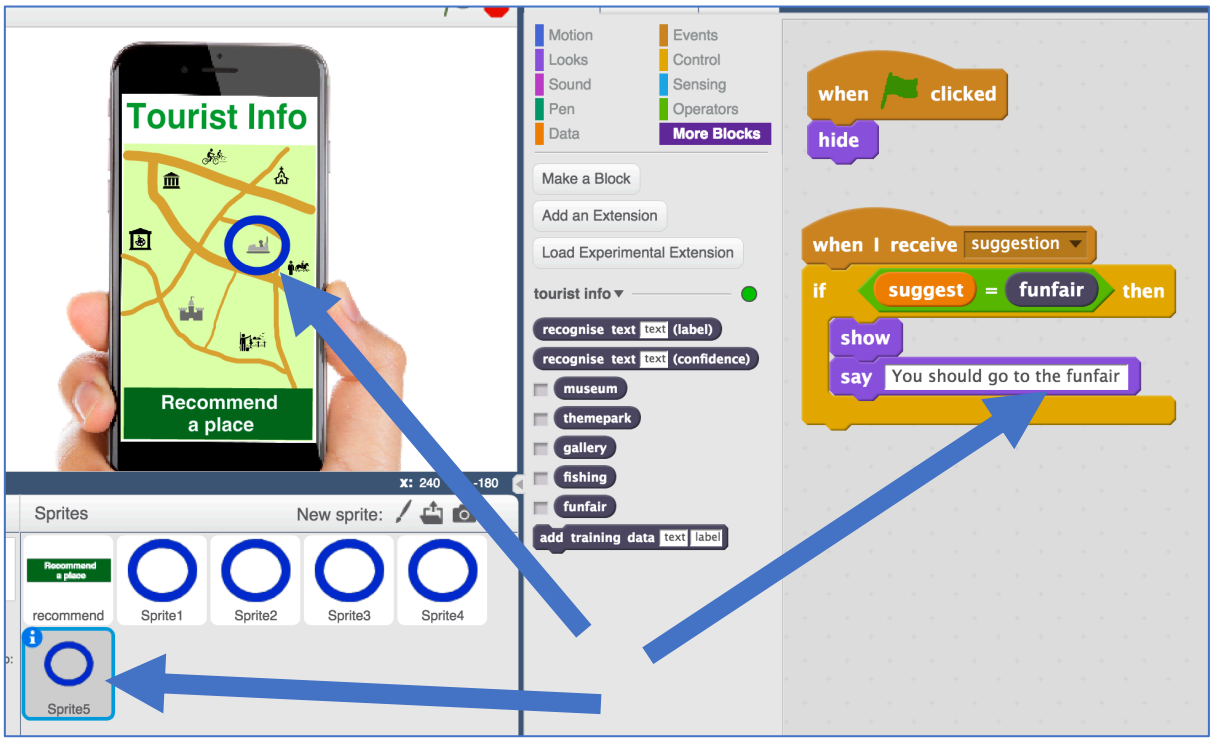
*Click the **"< Back to project**" link, then click the **Scratch** button.*
*Click the "**Open in Scratch**" button*
You should see the blocks added to the Scratch palette by your project now includes a new "funfair" block.

41. Open your project

*Click **File** -> **Load Project***

42. Duplicate one of the circle sprites to make a sprite for "funfair" recommendations

If you duplicate a hidden sprite, it's hard to know where to move it! Click on the blue i button, and tick "show" so you know where it is.
Make sure you put the circle in the right place.
Update the script to be a recommendation to go to the funfair.



43. Save your project

44. Test your project by clicking the Green Flag

Ask your Tourist bot for recommendations.
Try asking for something that would be good for a thrill-seeker who likes excitement and things that will get their heart pounding.
Does it recommend the Theme Park any more?

What have you done?

This is an example of "training bias". You've made your machine learning biased in favour of the funfair.

By giving it examples of thrillseekers with recommendations for funfair and not theme park, you're training the computer that it should make recommendations for the funfair and not the theme park.

By giving it more examples of funfair recommendations than anything else (in particular, more than the theme park), you're training the computer to learn the right answer is more often "funfair".

Is this fair?

Does the fact that the funfair owner paid for this bias make it more unfair?

Would it be okay if you'd done that accidentally and not intentionally – by collecting too many funfair examples without realising?

Would it make a difference if this bot was recommending medicines for doctors instead of holiday attractions to tourists?

What responsibilities do you think people training machine learning models should have about bias and being fair?