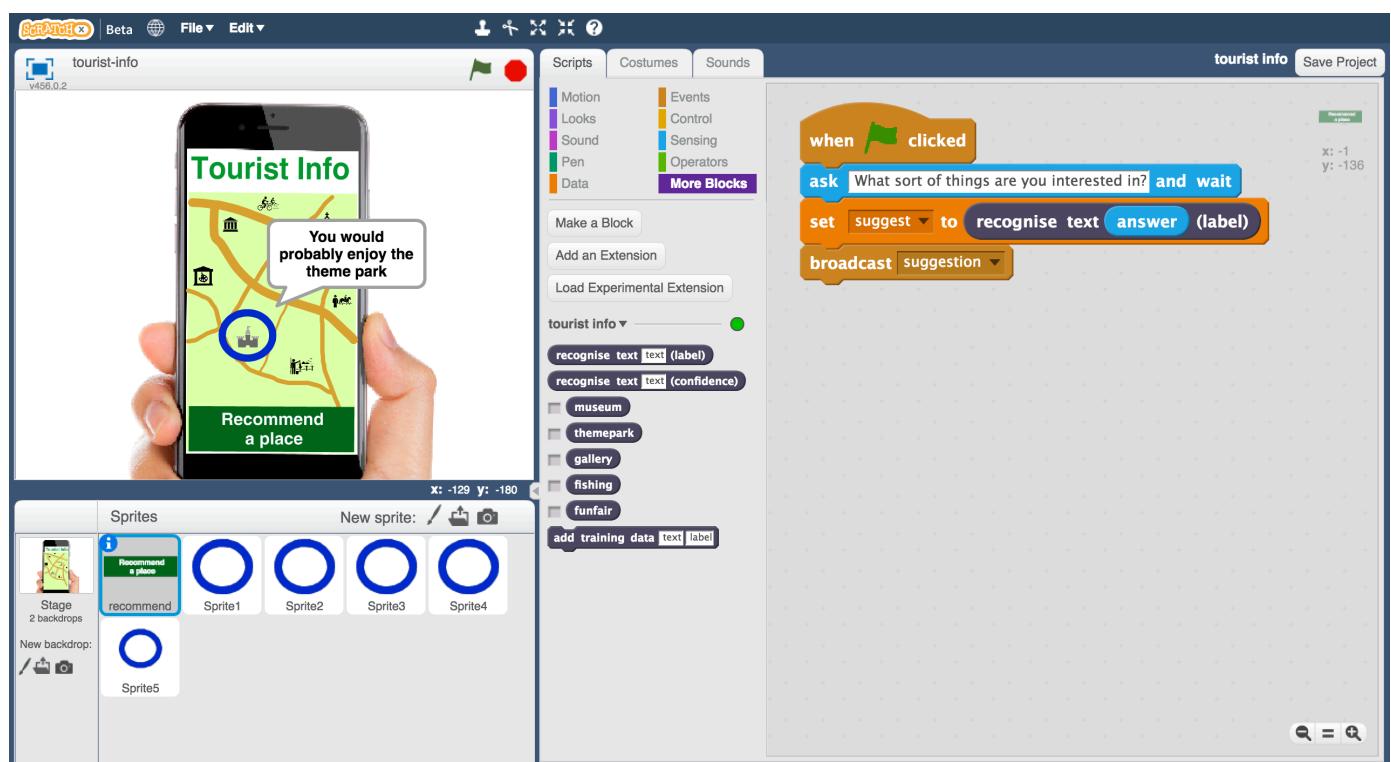


# Tourist Info

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 effect of "bias" on machine learning projects and see how this can happen.



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 “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”

Start a new machine learning project

Project Name \*

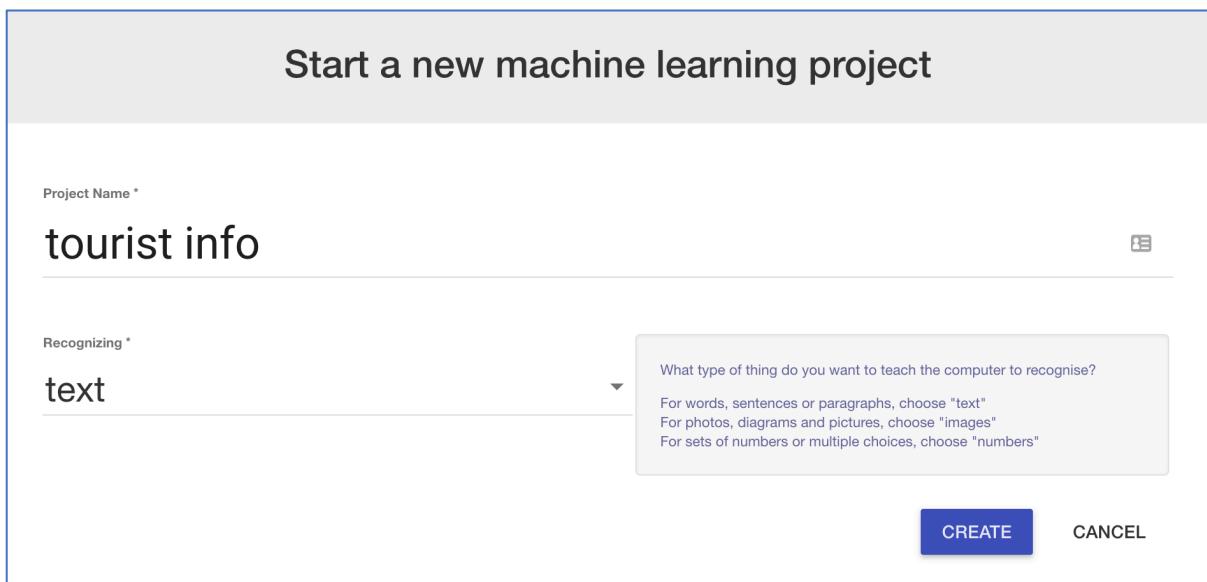
tourist info

Recognizing \*

text

What type of thing do you want to teach the computer to recognise?  
For words, sentences or paragraphs, choose "text"  
For photos, diagrams and pictures, choose "images"  
For sets of numbers or multiple choices, choose "numbers"

**CREATE**   **CANCEL**



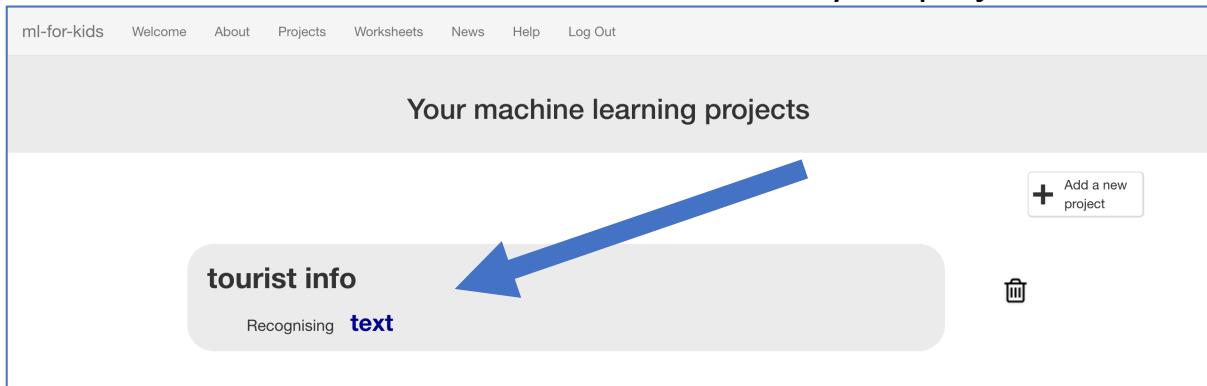
7. You should see “tourist info” in the list of your projects. Click on it.

ml-for-kids   Welcome   About   Projects   Worksheets   News   Help   Log Out

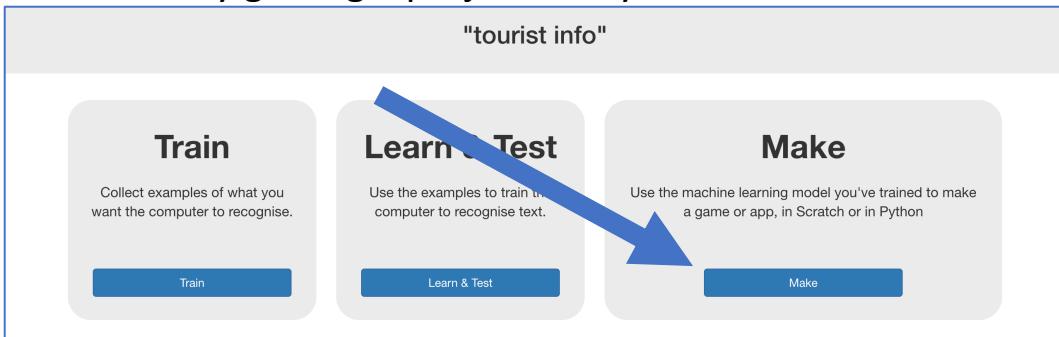
Your machine learning projects

**tourist info**      **Recognising text**

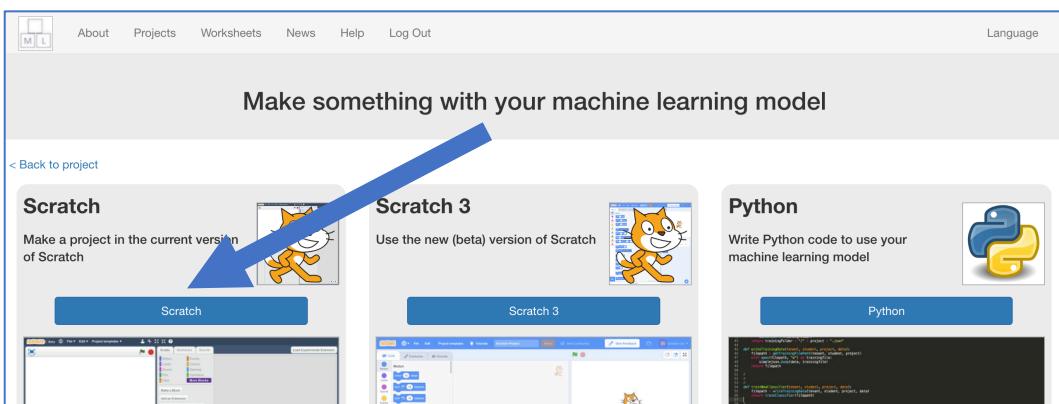
**+ Add a new project**   



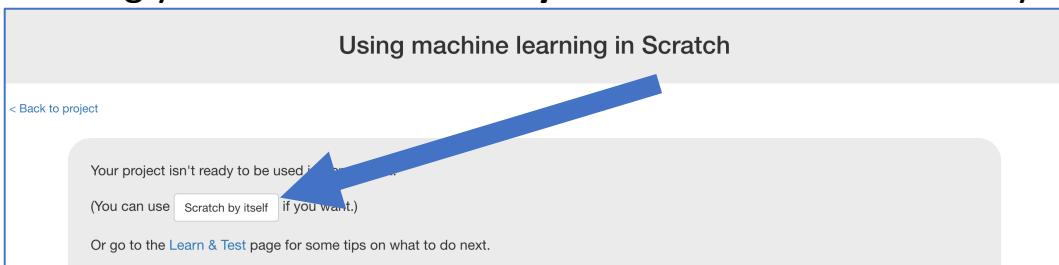
## 8. Start by getting a project ready in Scratch. Click the “Make” button.



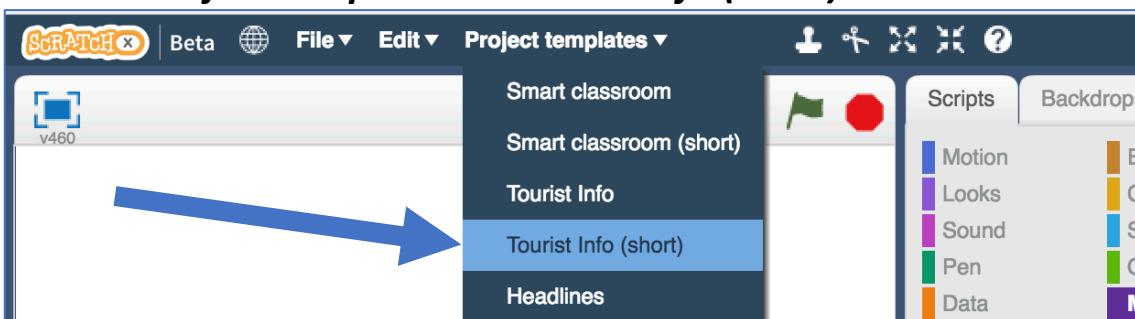
## 9. Click “Scratch”



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

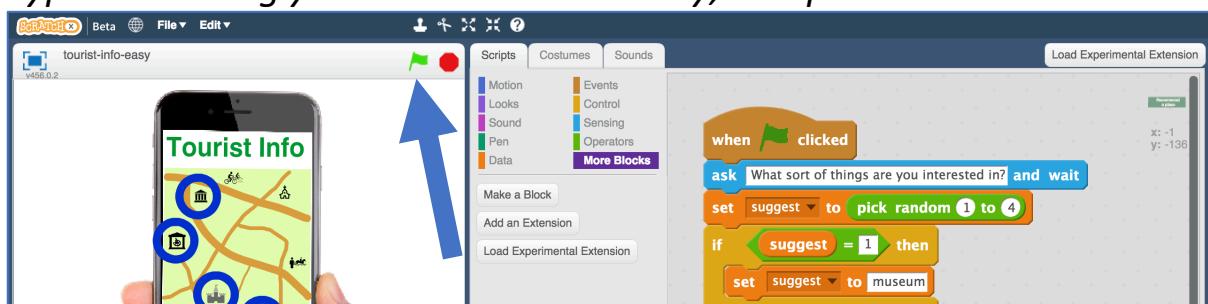


## 11. Open the **Tourist Info (short)** template project Click on **Project templates** -> **Tourist Info (short)**



## 12. Click the green flag to try it out

Type something you like to do on holiday, and press Enter.



## 13. 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.

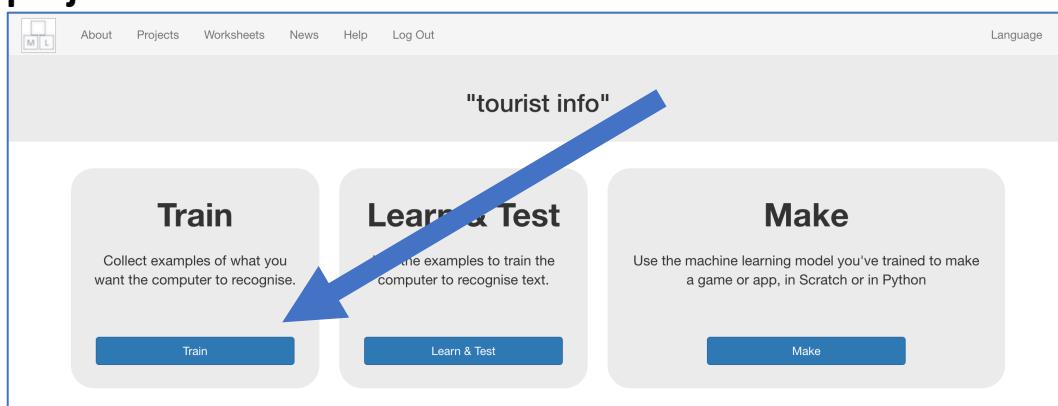
## 14. Close the Scratch window

### What have you done so far?

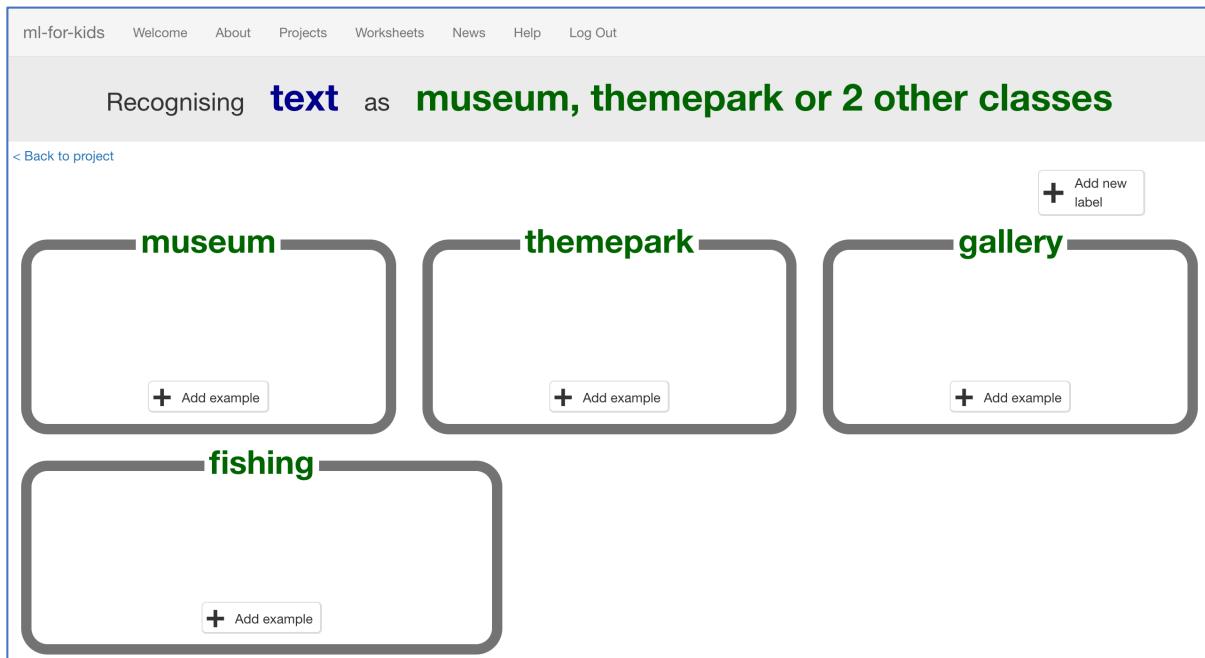
This is a mobile Tourist Information bot that will make recommendations to visitors. It will ask them what they’re interested in, so it can make the best recommendation. But for now, it’s choosing something at random.

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

## 15. We need examples to train the computer. Click the “< Back to project” link. Then click the Train button.



- 16.** 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”.



- 17.** 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!”*

- 18.** 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”*

- 19.** 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”*

- 20.** 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”*

## 21. Repeat steps 17 – 20 until you've written **five** examples of each.

The screenshot shows a web interface for collecting text examples. At the top, there's a navigation bar with links for About, Projects, Worksheets, News, Help, and Log Out. Below that, the title "Recognising text as museum, themepark or 2 other classes" is displayed. There are four main sections, each labeled with a category and a green header: "museum", "themepark", "gallery", and "fishing". Each section contains a list of example messages and a "+ Add example" button. A small green circle with the number "6" is positioned at the bottom of each section, indicating the count of examples collected. In the top right corner, there's a button labeled "+ Add new label".

## 22. Click on the “< Back to project” link.

Then click on the “Learn & Test” button.

## 23. 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.*

The screenshot shows a web interface for training a machine learning model. At the top, the title "Machine learning models" is displayed. Below that, there are two main sections: "What have you done?" and "What's next?". The "What have you done?" section contains a message about collecting examples and a list of what was collected: 6 examples of museum, 6 examples of themepark, 6 examples of gallery, and 6 examples of fishing. The "What's next?" section contains a message about being ready to start training and a button labeled "Click the button below to start training a machine learning model using the examples you have collected so far." A blue arrow points from the "What have you done?" section towards the "Train new machine learning model" button in the "Info from training computer:" box at the bottom. The "Info from training computer:" box also contains a "Train new machine learning model" button.

## 24. 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.*

**25.** 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 21, and add some more examples. Make sure you repeat step 23 to train with the new examples though!*

The screenshot shows a web-based machine learning interface. At the top, there's a navigation bar with links for About, Projects, Worksheets, News, Help, and Log Out. Below that is a header titled "Machine learning models". A blue arrow points from the text in step 25 to the "Test" input field in the screenshot. The "What have you done?" section contains text about training a model to recognize text like "museum", "themepark", "gallery", and "fishing". It also shows the date of creation: Wednesday, August 29, 2018 9:14 PM. The "What's next?" section provides instructions for testing the model with new text. The main area features a text input field with placeholder text "Try putting in some text to see how it is recognised based on your training." and a "Test" button. Below the input field, the text "I'd like to learn something about the local area" is entered, and the output "Recognised as museum with 91% confidence" is displayed.

## 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.

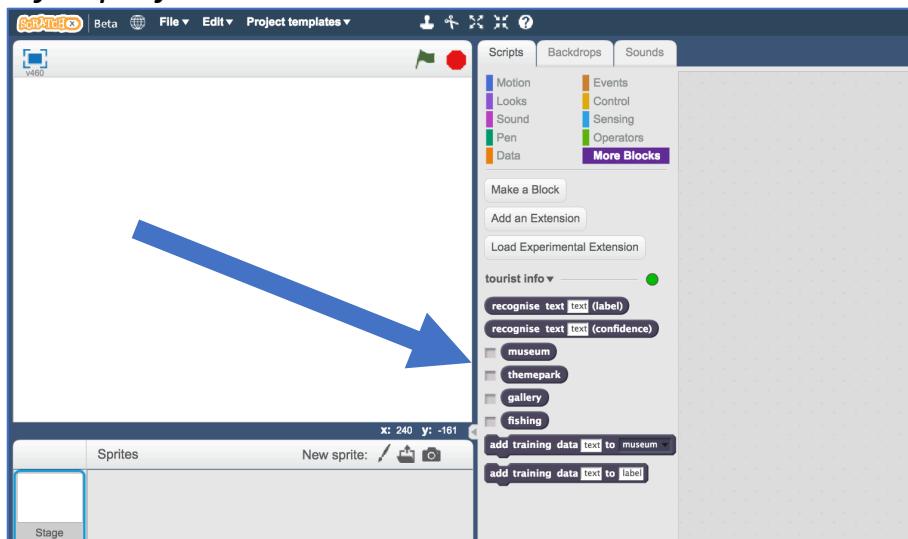
**26.** Click the “< Back to project” link

**27.** Click the “**Make**” button, 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.*

**28.** Click on the “**Open in Scratch**” button to launch the Scratch editor.

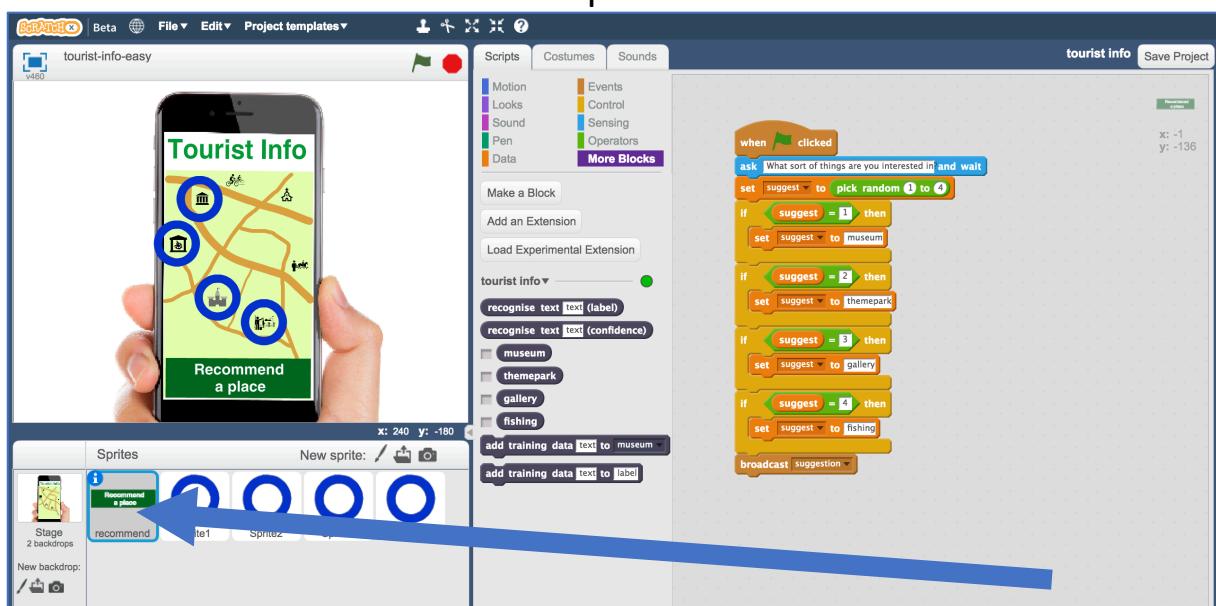
*You should see new blocks in the “More blocks” section from your “tourist info” project.*



**29.** Load the Scratch project you opened before.

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

**30.** Click on the “recommend” sprite



## 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.

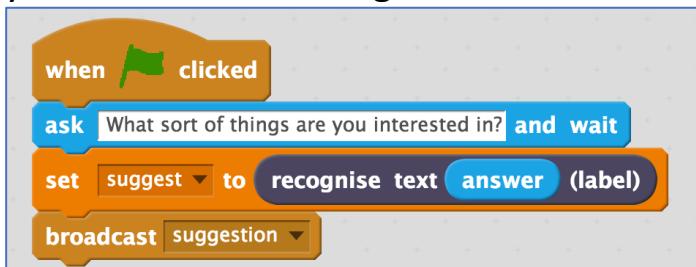
The more the better!

### Mix things up with your examples

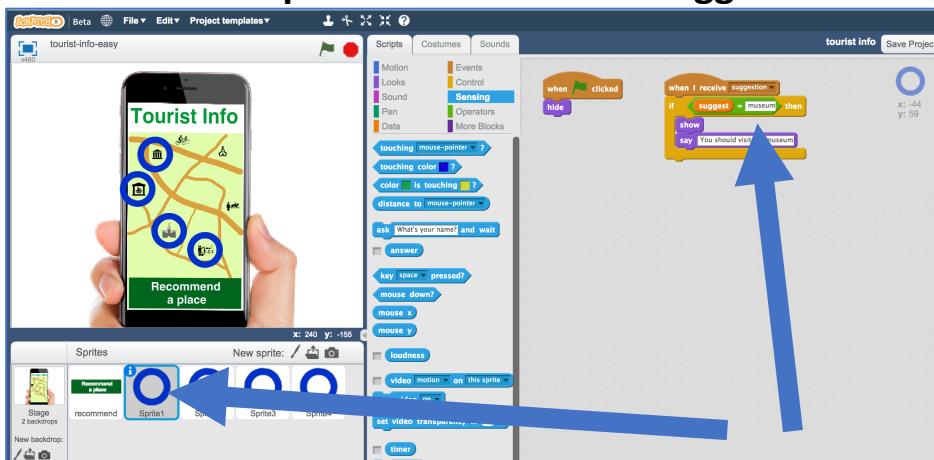
Try to come up with lots of different types of examples.

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

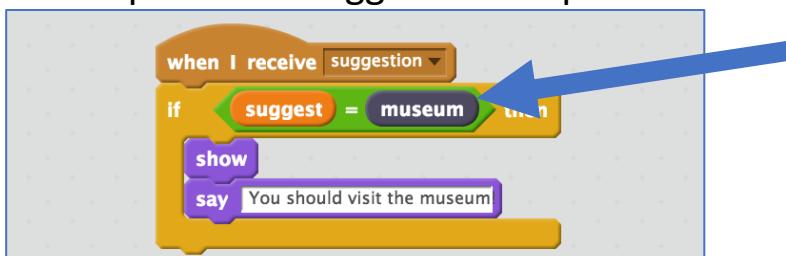
**31.** Replace the script that is there with this one below, that will use your machine learning model instead of the random choice.



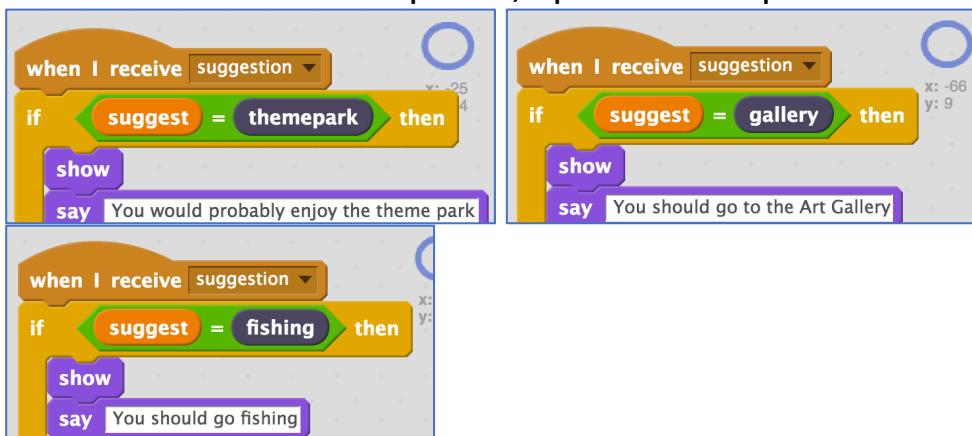
**32.** Click on “Sprite1” and find the “suggestion” script



**33.** Update the suggestion script to use one of your new blocks



**34.** Do the same for Sprite2, Sprite3 and Sprite4

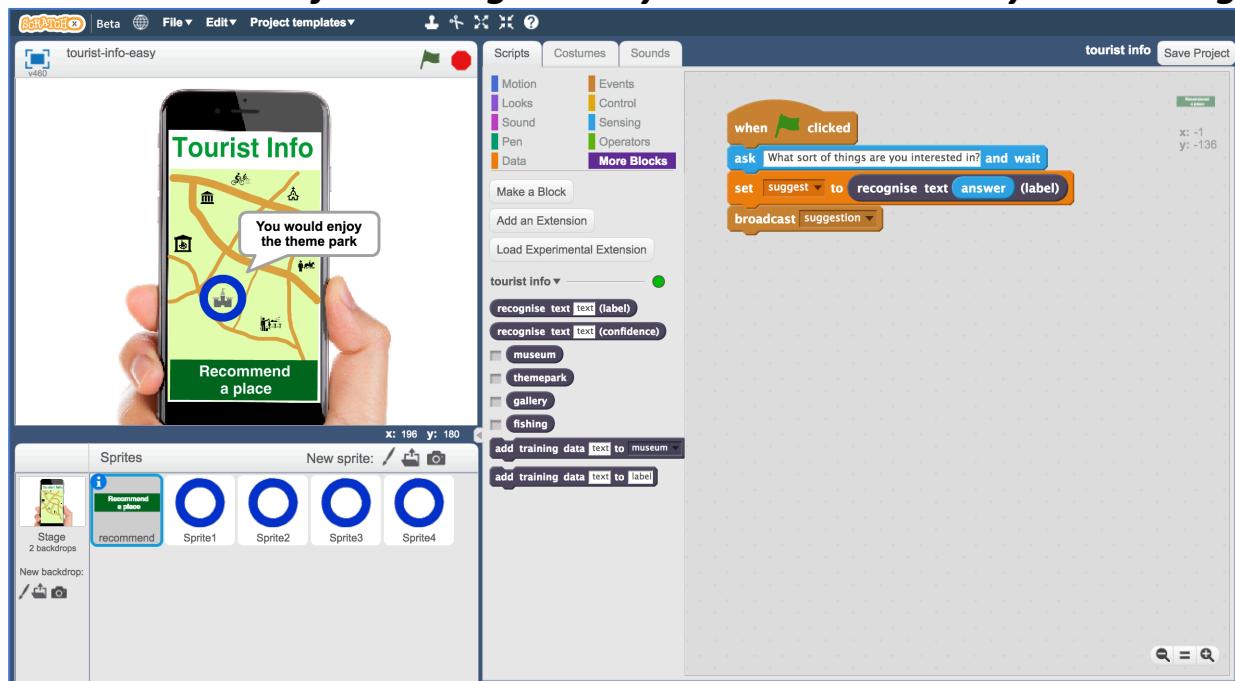


## 35. Test your project by clicking the Green Flag

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.**



## 36. Save your project.

**Click File -> Save Project**

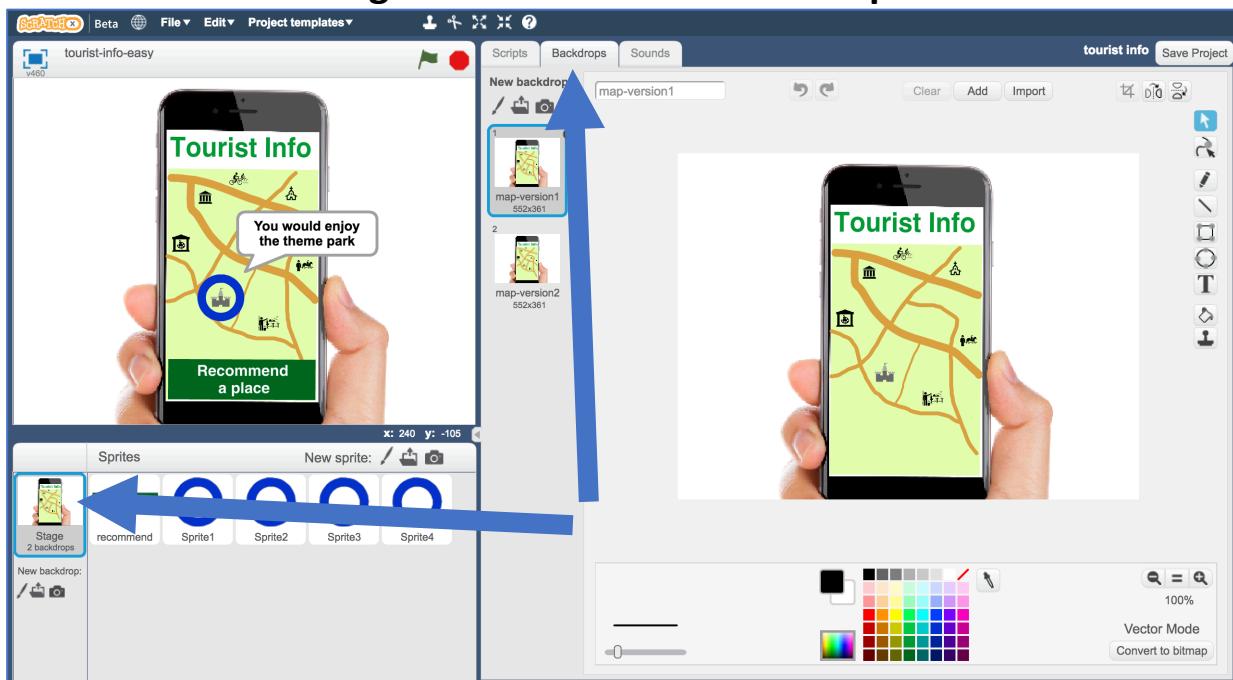
### What have you done so far?

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”

### 37. Click on the Stage and then click on Backdrops



### 38. 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.

---

### 39. Save your Scratch project

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

### 40. Close the Scratch window

### 41. Go back to the “Train” page

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

## 42. Add a new bucket for “funfair”

Click the “**Add new label**” button. Call the new label “funfair”

## 43. Add a lot of examples to the “funfair” bucket

Use examples from the “themepark” bucket, then delete them from themepark bucket. Leave 1 or 2 examples in the themepark bucket so it’s not empty. Add a lot more new examples to the funfair bucket as well, so it has lots more examples than any other attraction.

The screenshot shows a web-based machine learning interface for training a model to recognize text. At the top, it says "Recognising **text** as **museum, themepark or 3 other classes**". Below this are five labeled buckets:

- museum**: Contains 6 examples.
- themepark**: Contains 2 examples.
- gallery**: Contains 6 examples.
- fishing**: Contains 6 examples.
- funfair**: Contains 16 examples.

Each bucket has a "Add example" button at the bottom. A "Add new label" button is located in the top right corner of the main interface area.

## 44. 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.

## 45. Go back to Scratch

Click the “**< Back to project**” link.

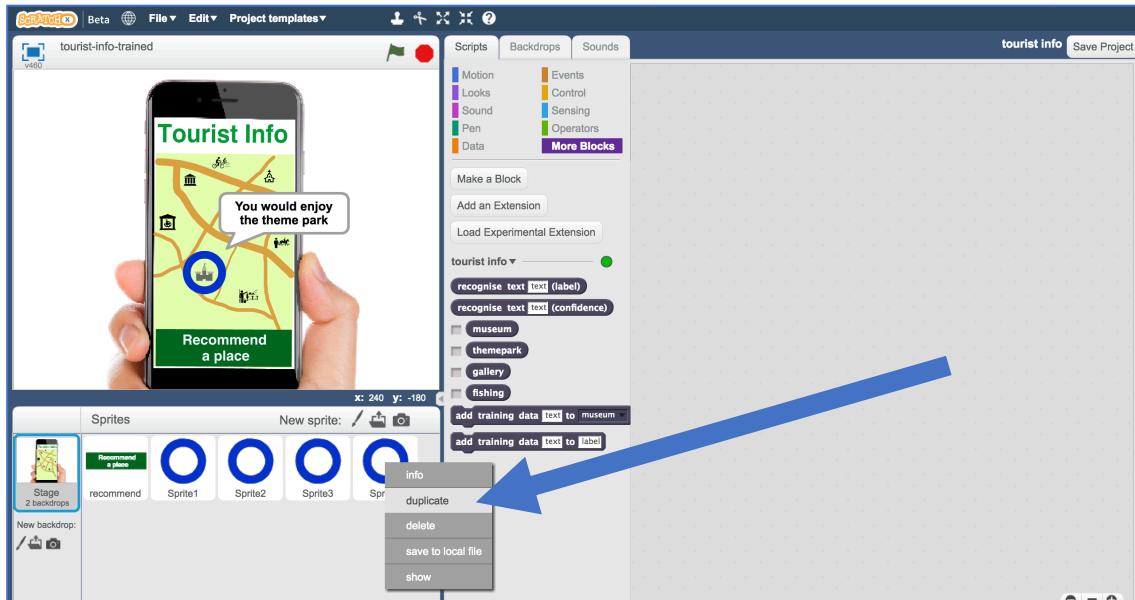
Click “**Make**” then click the **Scratch** button. Click “**Open in Scratch**”

You should see the blocks in the Scratch palette includes a “funfair” block.

## 46. Open your project

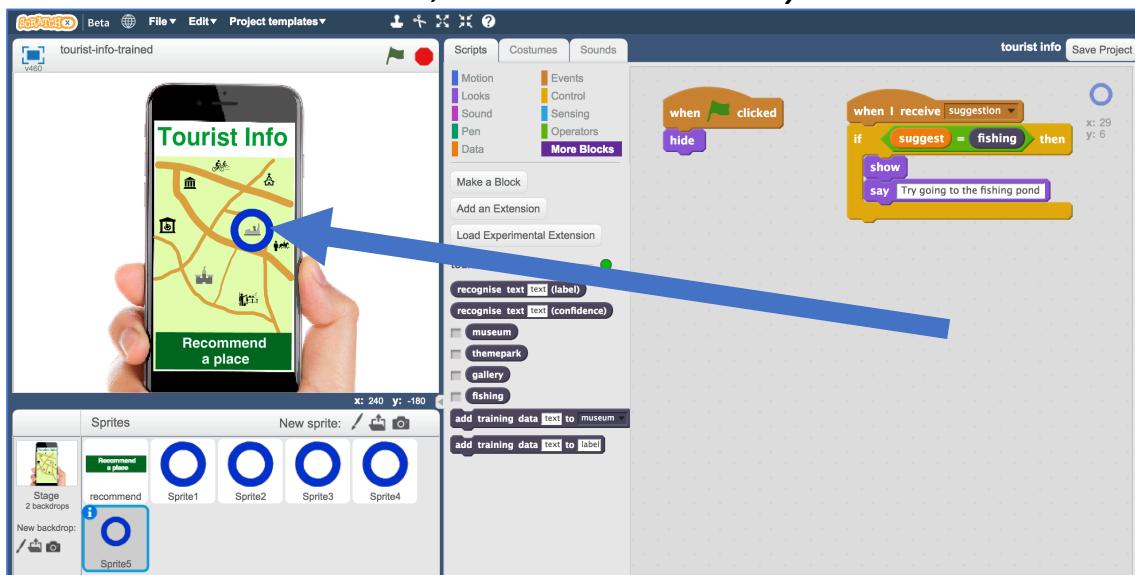
Click **File -> Load Project**

**47.** Right-click on one of the circle sprites, and click on “Duplicate”

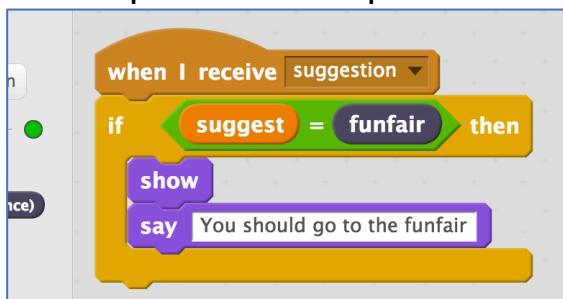


**48.** Move the new Sprite5 to the location of the funfair

*If you duplicated a hidden sprite, it's hard to know where it is!  
Click on the blue i button, and tick "show" so you know where it is.*



**49.** Update the script to be a recommendation to go to the funfair



## 50. Save your project

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

## 51. Test your project by clicking the Green Flag

*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 anymore?*

### 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 thrill-seekers 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 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?

