Chatbots

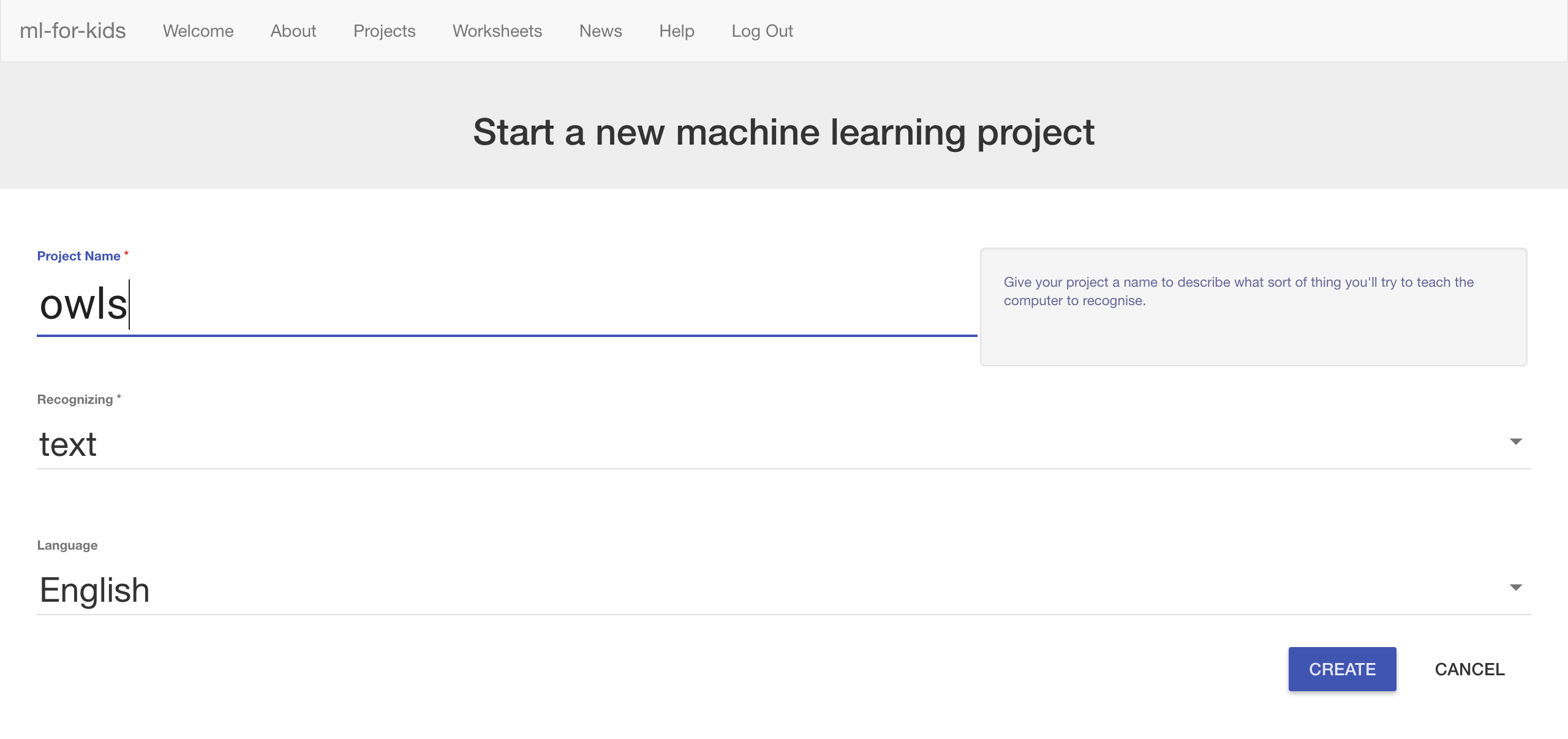
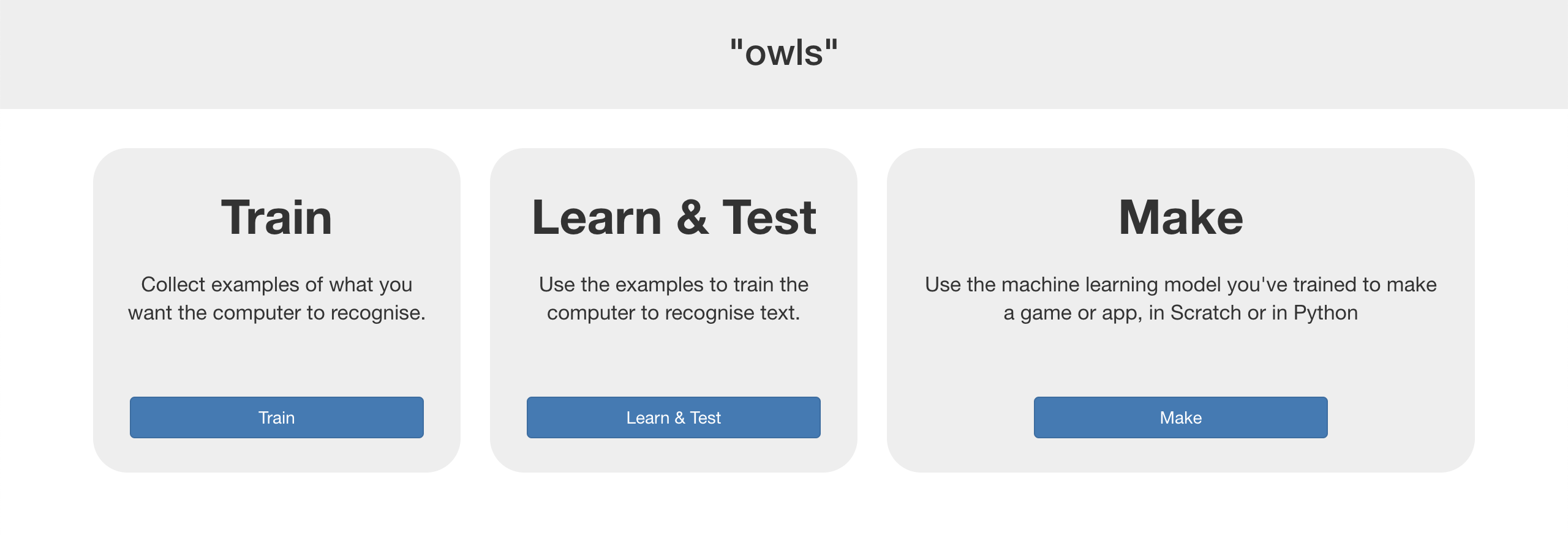
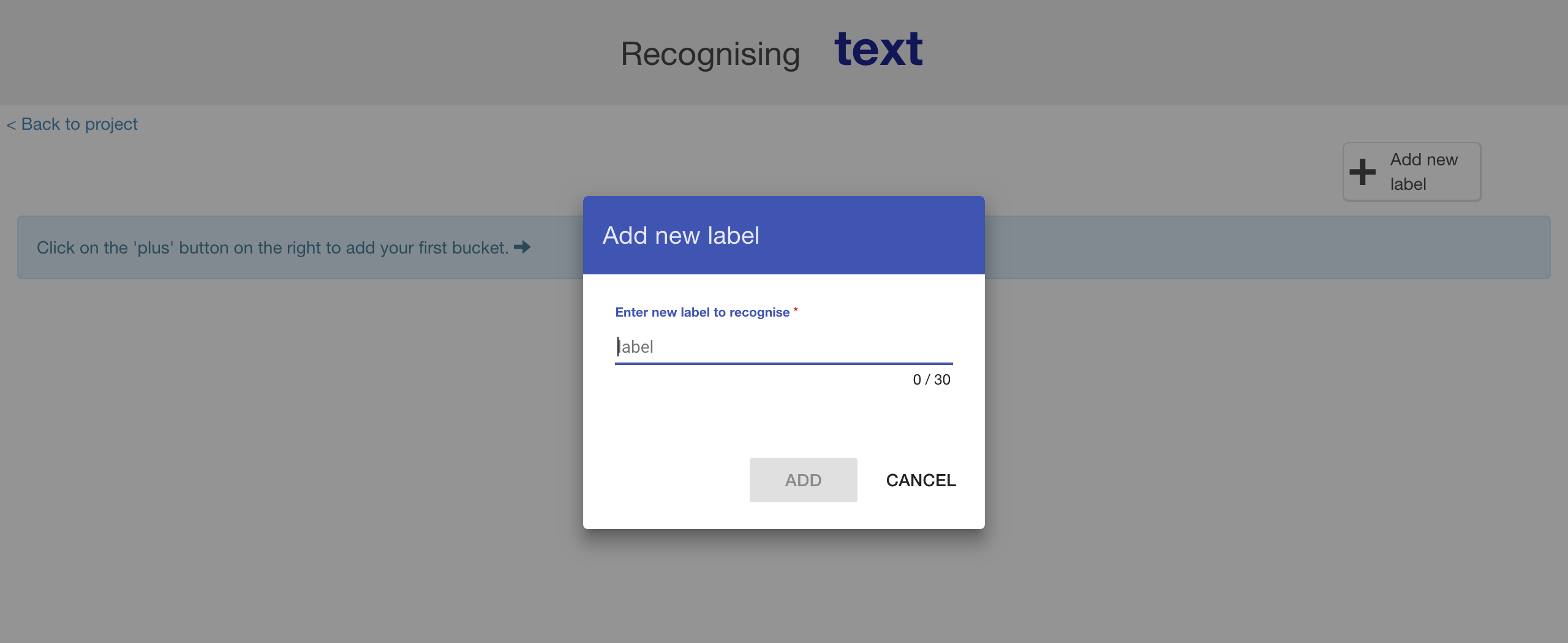
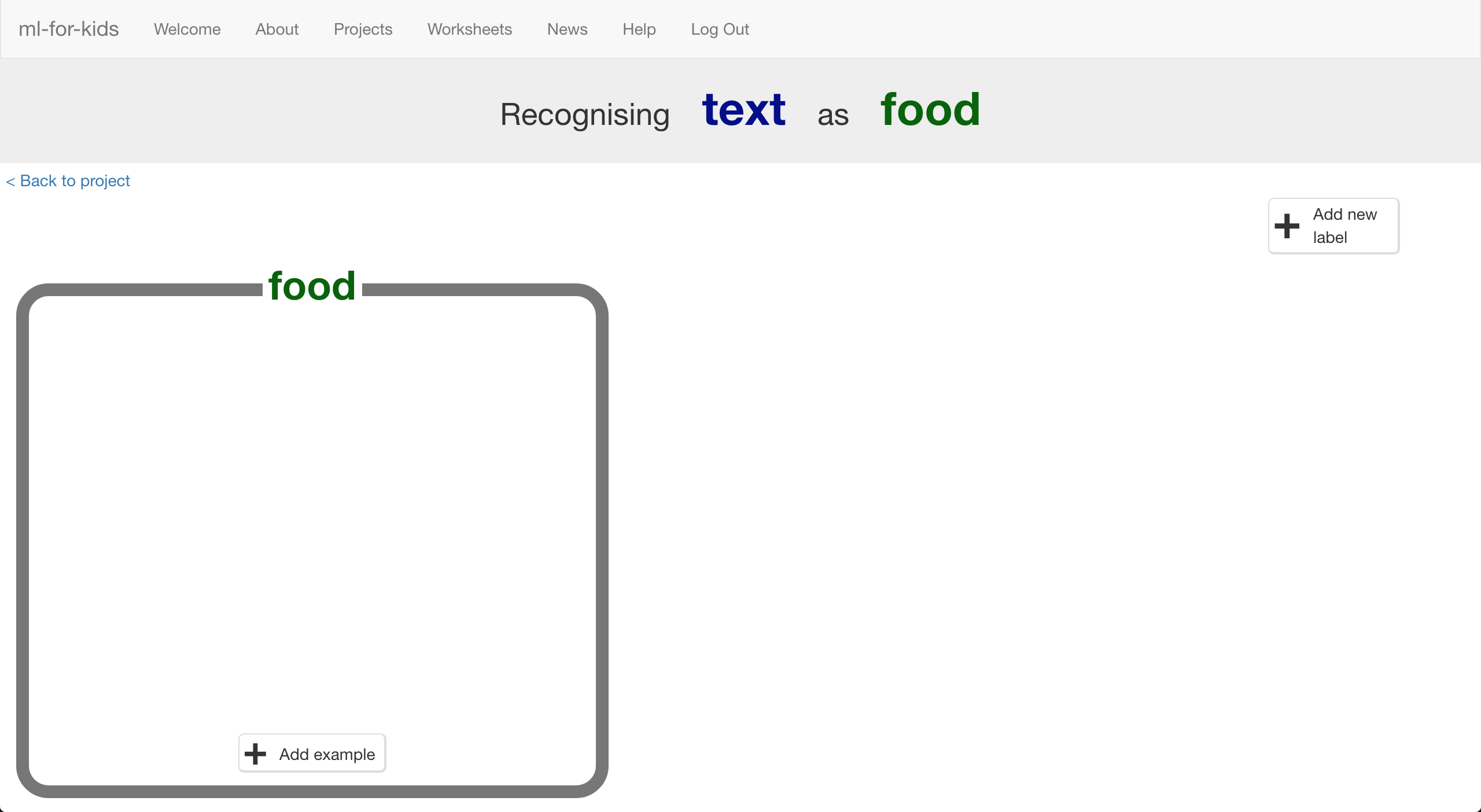
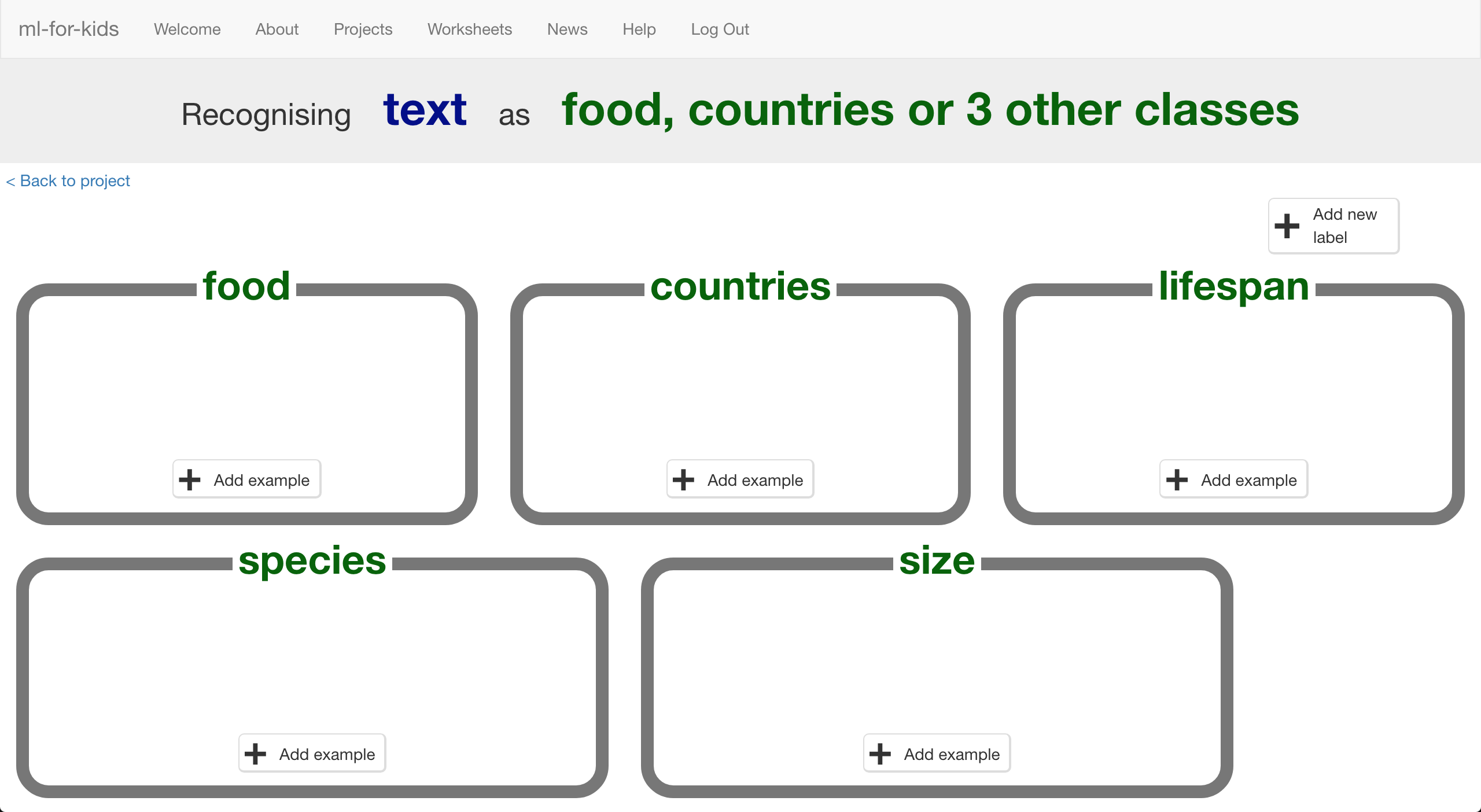
In this project you will make a chatbot in Python that can answer questions about a topic of your choice.

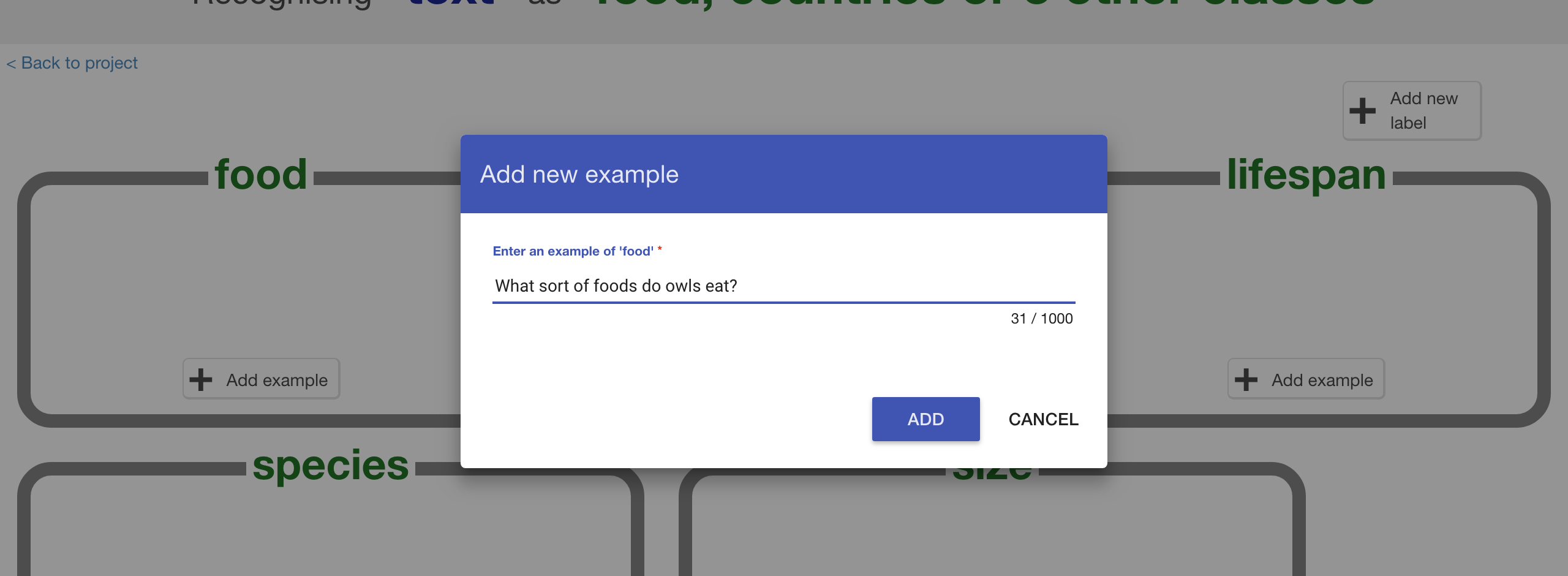
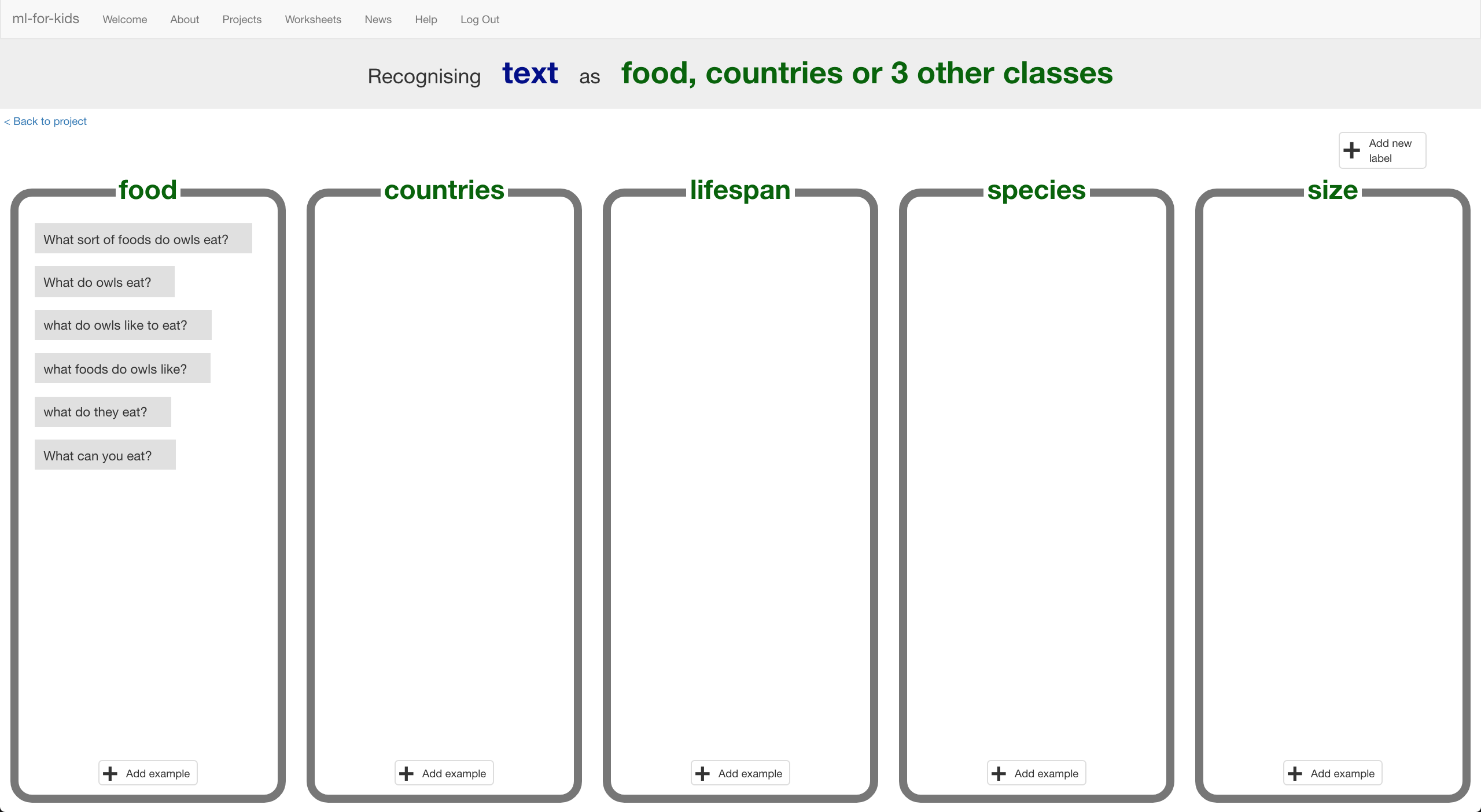
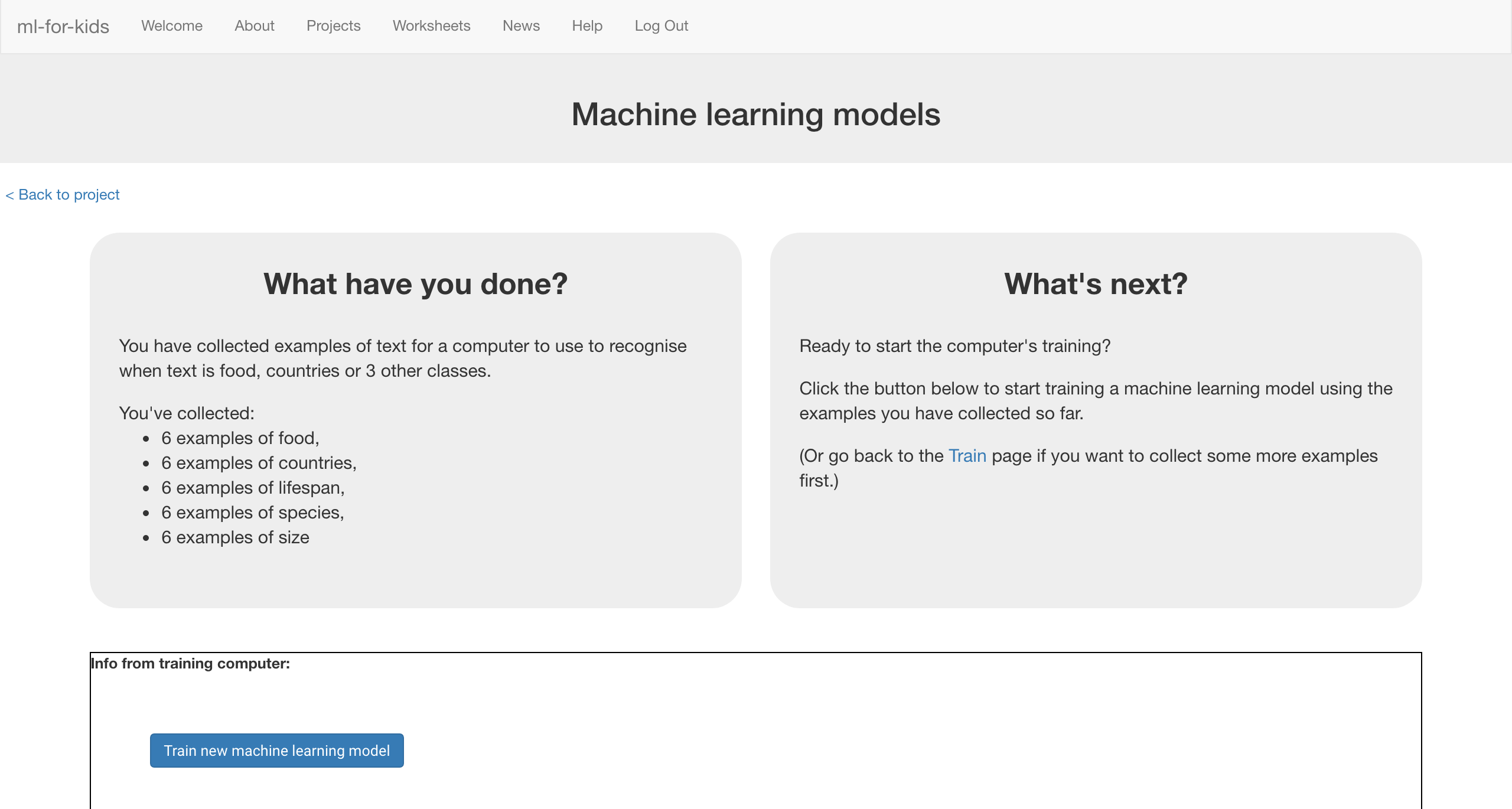
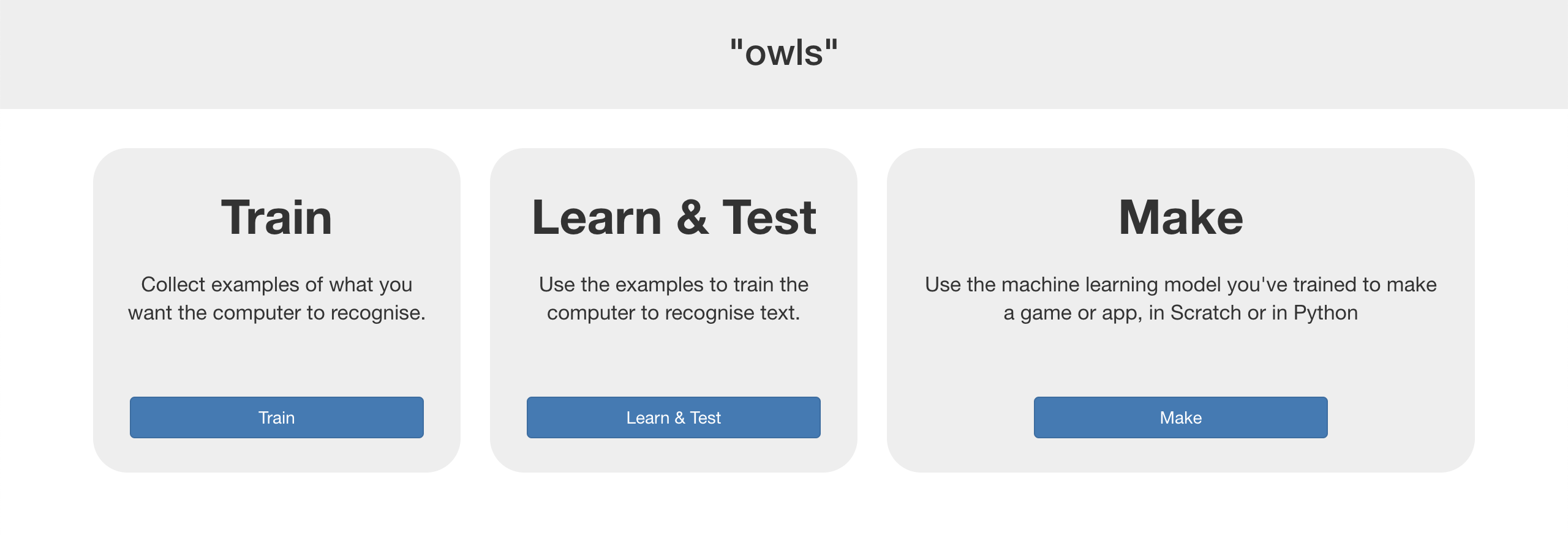
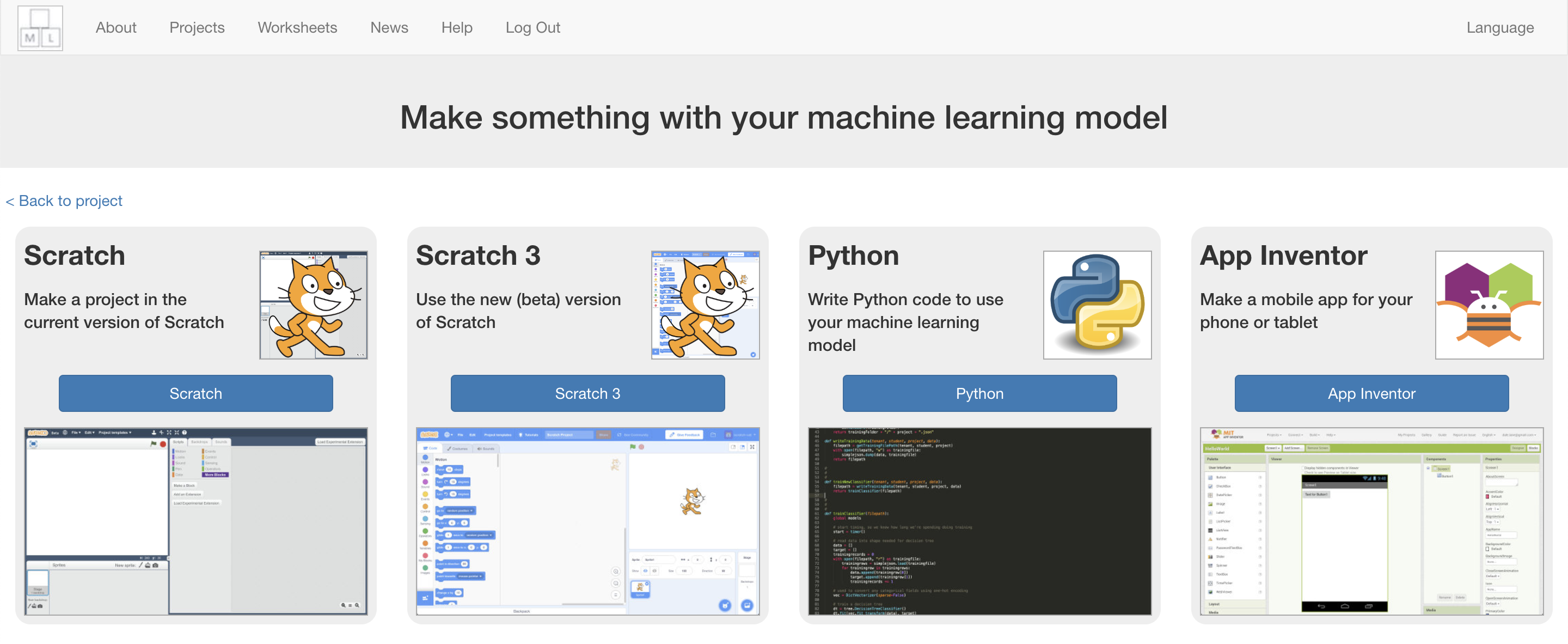
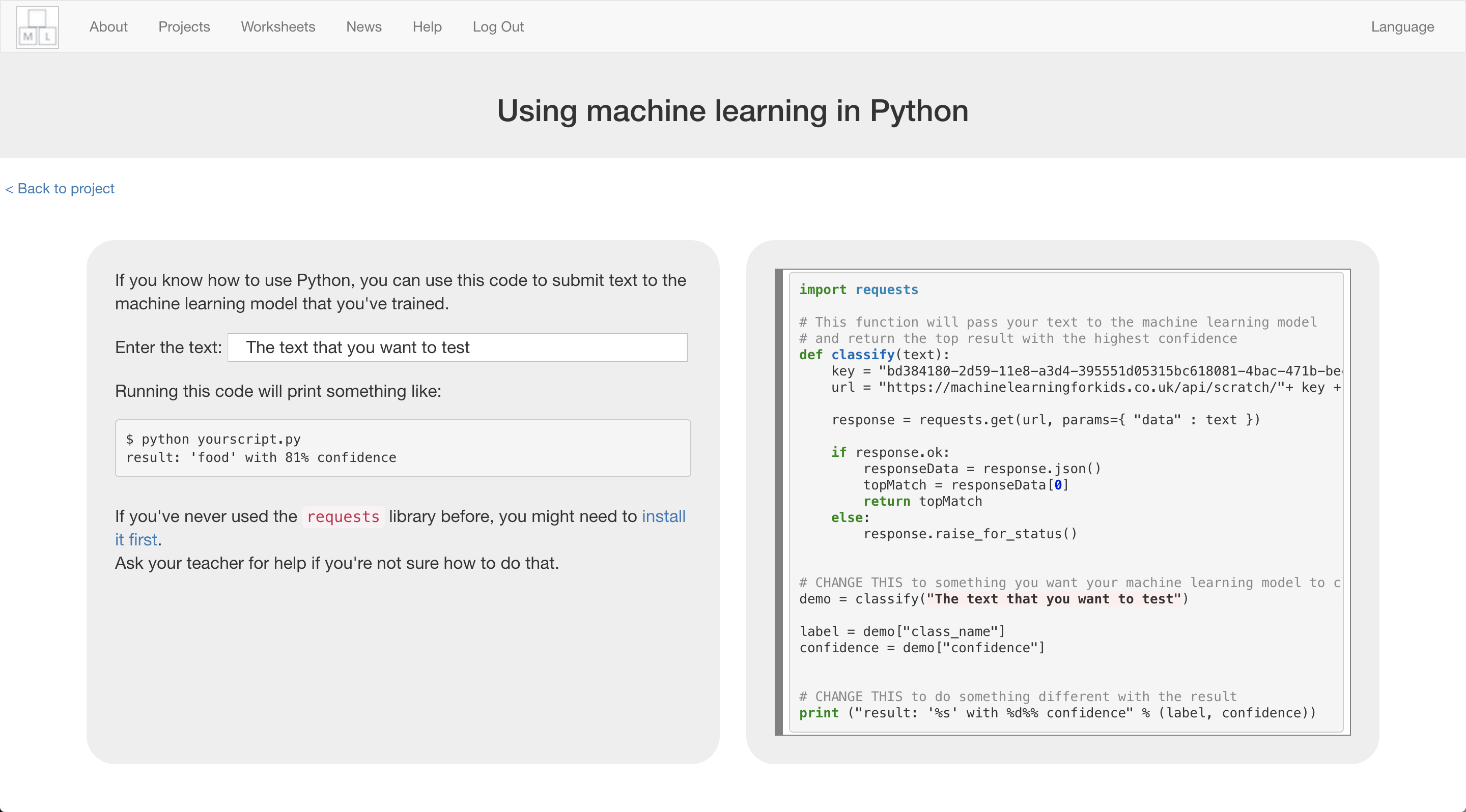
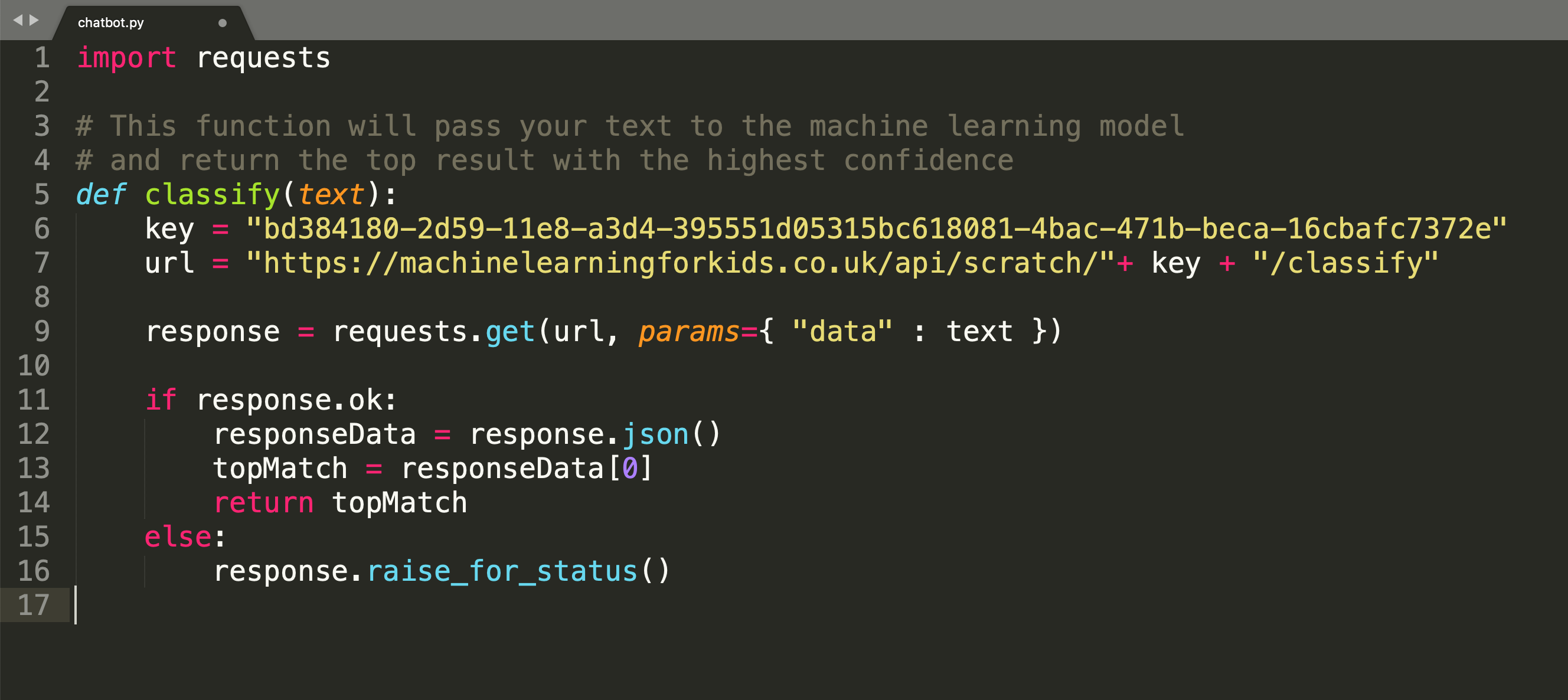
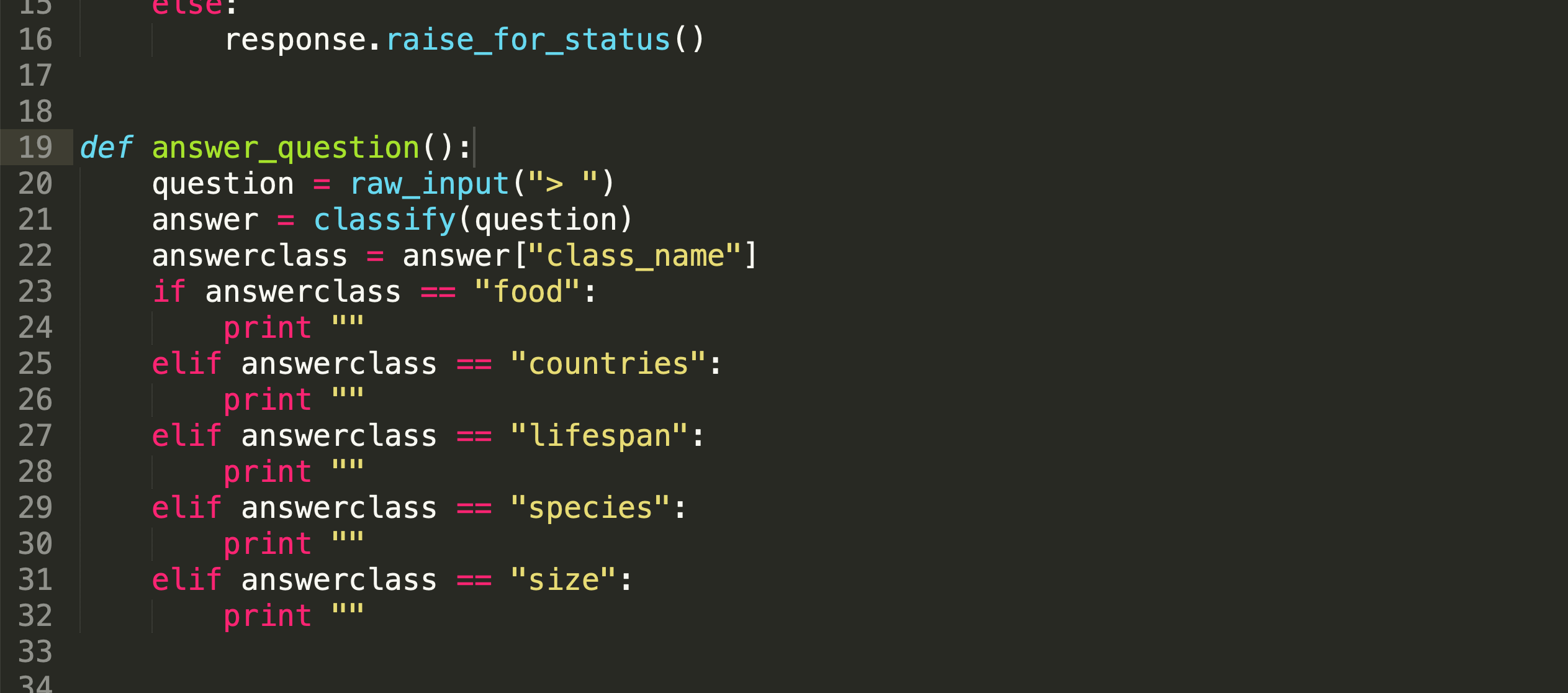
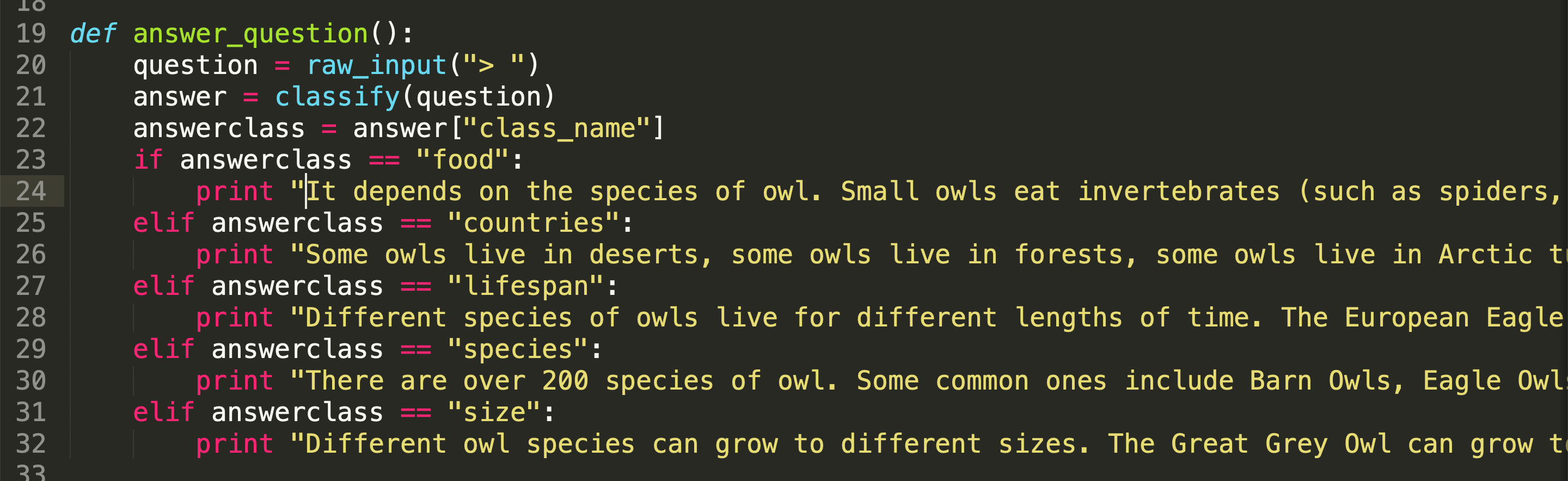
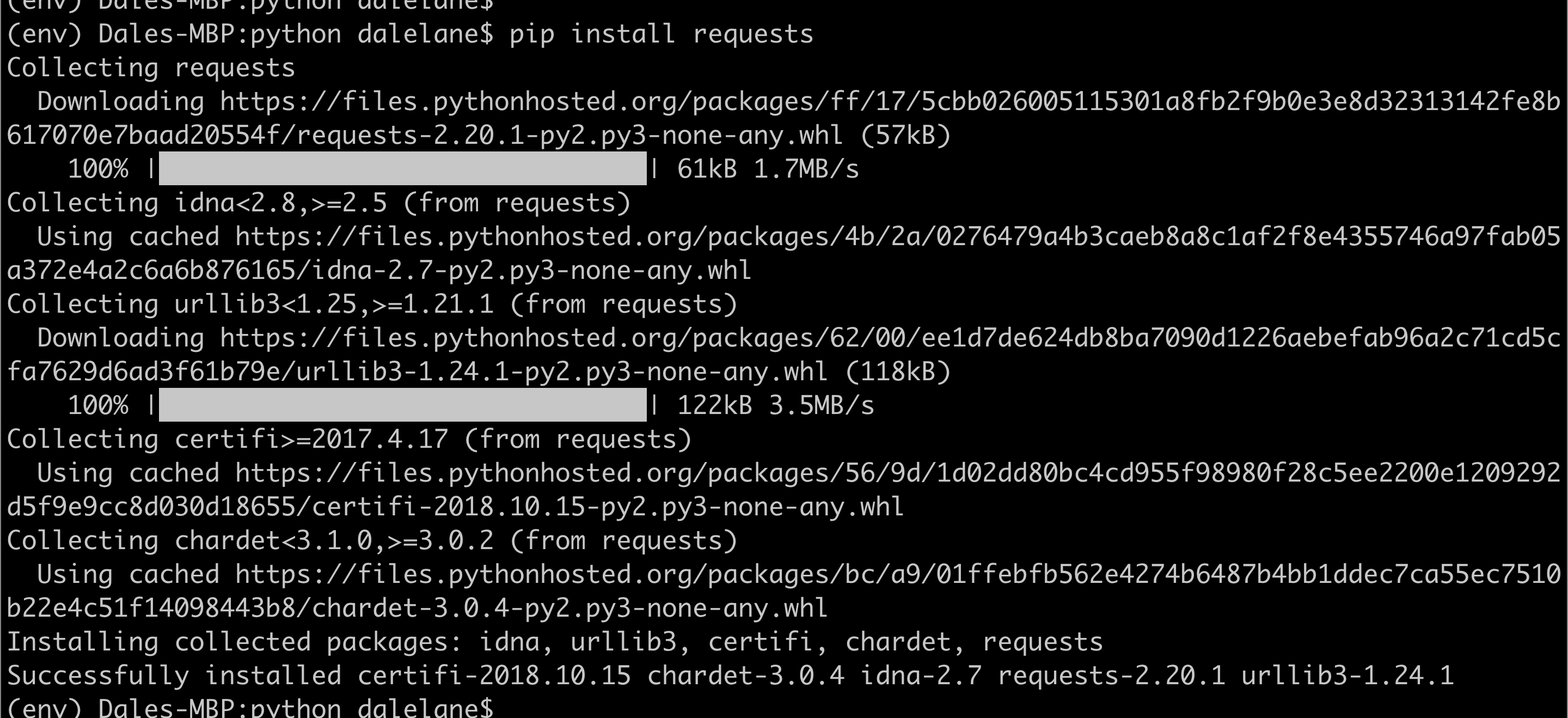
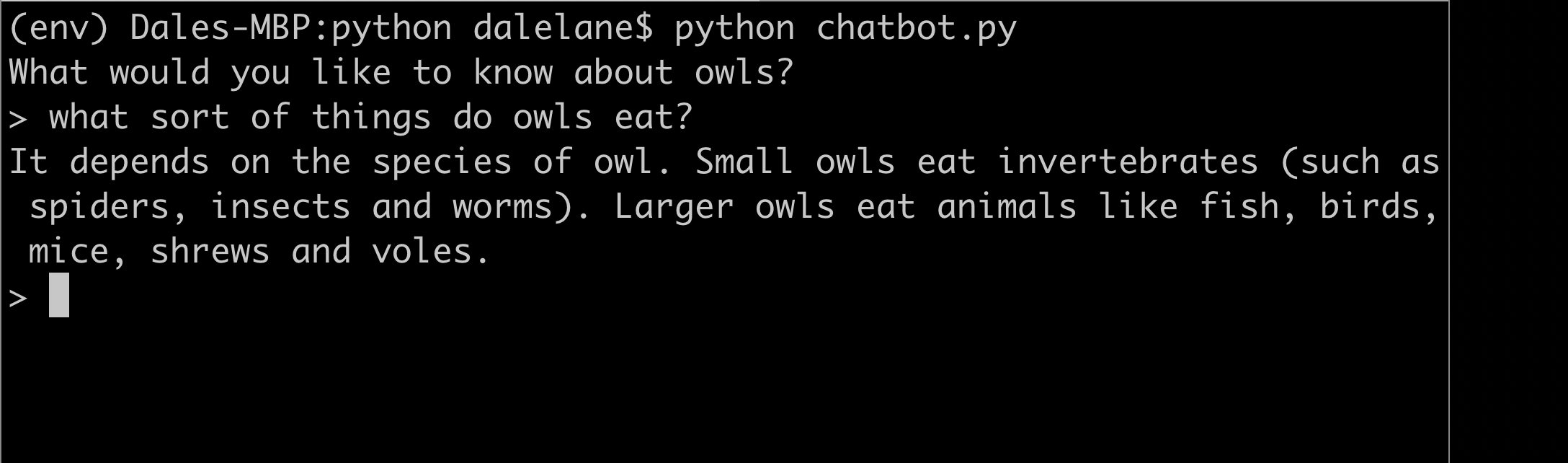


Creative Commons LicenceThis project worksheet is licensed under a Creative Commons Attribution Non-Commercial Share-Alike License

http://creativecommons.org/licenses/by-nc-sa/4.0/

1. Decide on **a topic** for your chatbot  
   Choose something that you know well enough to be able to answer questions about.   
   *It could be a place (e.g. The town where you live?)  
   It could be an animal (e.g. Tigers? Dinosaurs?)  
   It could be an organisation (e.g. Your school)  
   It could be something from history (e.g. Vikings? Romans?)  
   For the rest of this worksheet, I’ll be using* ***owls***
2. Think of **five things** someone might ask about your topic  
   *e.g. for* ***owls****, this could be:  
   \* What do owls eat?  
   \* Where in the world do owls live?  
   \* How long do owls live?  
   \* What types of owls are there?  
   \* How big do owls grow?*
3. Go to <https://machinelearningforkids.co.uk/> in a web browser
4. Click on “**Get started**”
5. Click on “**Log In**” and type in your username and password  
   *If you don’t have a username, ask your teacher or group leader to create one for you.  
   If you can’t remember your username or password, ask your teacher or group leader to reset it for you.*
6. Click on “**Projects**” on the top menu bar
7. Click the **“+ Add a new project**” button.

1. Name your project and set it to learn how to recognise “**text**”.   
   Click the “**Create**” button  
   
2. Click on your new project in the projects list
3. Click the **Train** button.  
   
4. Click the “**+ Add new label**” button  
   
5. Type in **one word** that sums up the first of your things from Step 2, then click **Add**.   
   *I used “food” to sum up questions like “What do owls eat?”*  
   **
6. Do that again for all of the things in your list from Step 2  
   *The words you choose don’t really matter, as long as* ***you*** *understand what they mean.*

1. Click the “**+ Add example**” button in one of the buckets
2. Type in an example of how someone might ask that question  
   
3. Click “**Add**”
4. Repeat until you’ve got **five examples** of how to ask that question.  
   
5. Repeat until you’ve got at least five examples in every bucket  
   
6. Click on the “**< Back to project**” link
7. Click the “**Learn & Test**” button
8. Click the “**Train new machine learning model**” button  
   *As long as you’ve collected enough examples, the computer should start to learn how to recognise questions from the examples you’ve given to it.*  
   
9. Wait for the training to complete.   
   *This might take a couple of minutes.  
   It’s finished once you see the “status” change to “Available”*
10. Click the “**< Back to project**” link
11. Click the “**Make**” button  
    
12. Click “**Python**”  
    
13. Copy the **import** line and the “**classify**” function   
    *You need the top half of the sample file as shown below*  
    
14. Paste that function into a new text file called **chatbot.py**  
    
15. Add a new function at the bottom, called “**answer\_question**”  
    If you’re using Python 3, you can replace “raw\_input” with “input”  
    *You need an* ***if*** *check for each of your answer types. Replace the “food”, “countries”, “lifespan”, etc. with the answer types that you used.*  
    
16. Put your answers to the questions into each of the “**print**” lines  
    
17. Add the following to the bottom of your script  
    *This will welcome you, and allow you to keep asking questions forever*  **
18. Install “**requests**”  
    *Line 1 in your Python script imports the “requests” library. If you’ve never used it before, you’ll probably need to install it now.  
    There are different ways to do this. One way is to type:***pip install requests** *Ask your teacher for advice on how to do this if you’re not sure.*  
    
19. Test your chatbot!  
    *Run your Python script and try asking a question  
    *

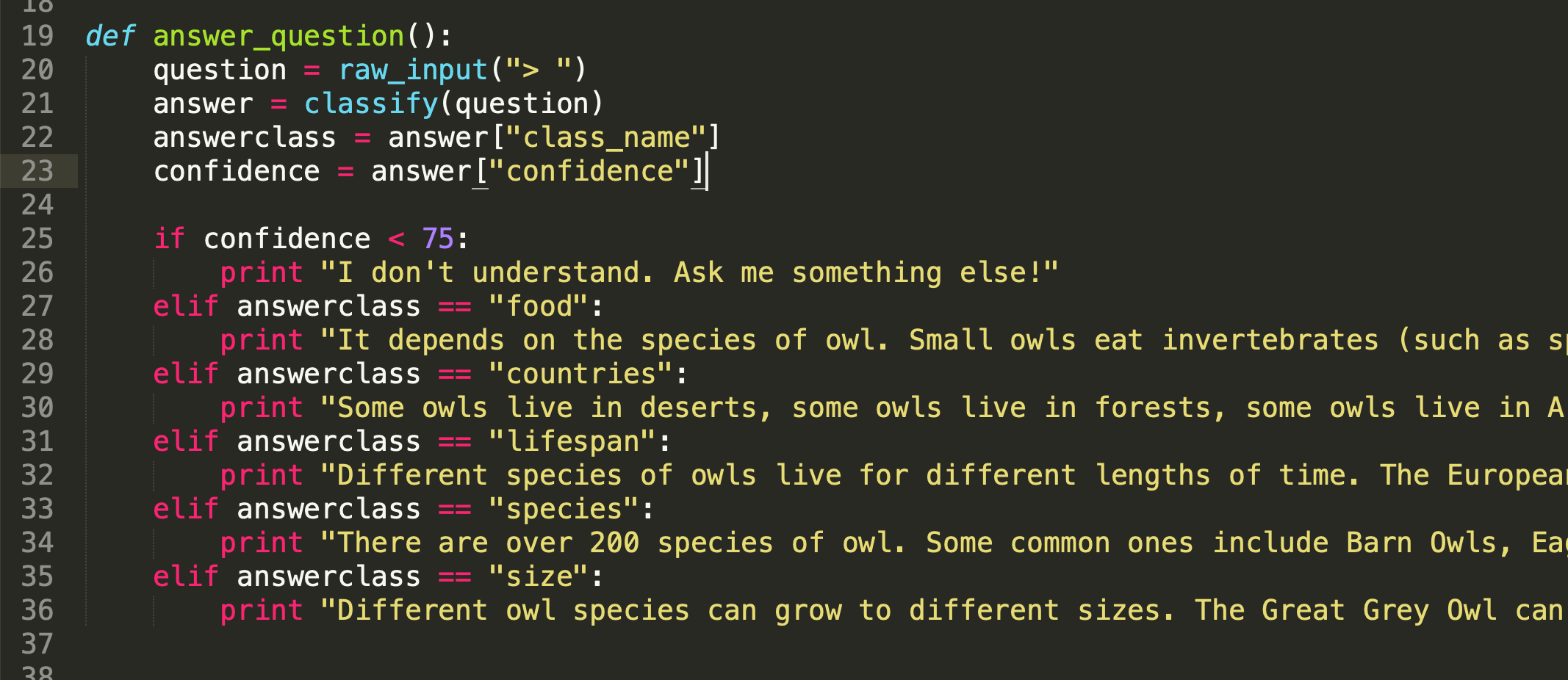
**What have you done so far?**

You’ve started to train a computer to recognise questions on a topic. You did this by collecting examples. These examples were used to train a machine learning “model”.

This is called “supervised learning” because of the way you are supervising the computer’s training.

The computer will learn from patterns in the examples you’ve given it, such as the choice of words, and the way questions are structured. These will be used to be able to recognise new questions.

The biggest problem with this is that if you ask it something unexpected, it will still give you one of the answers you’ve written

1. Change the **answer\_question** function to look like this  
   The bit that you need to change are shown below in lines 23 – 27  
   This will help when someone asks a question that wasn’t on your list.  
   *The confidence score is a percentage (from 0 to 100).   
   It will be lower if someone asks a question that isn’t similar to any of the examples you used to train the machine learning model.   
   Use this to return a “I don’t understand” message if the score is too low.   
   Experiment to work out the right confidence number to use.*  
   

**Ideas and Extensions**

Now that you’ve finished, why not give one of these ideas a try?

Or come up with one of your own?

**Try other chatbots**

<http://talktothetrex.com> is a good example of the sort of thing you’ve made. Give it a try and see if you can get any ideas of how to improve your bot.

**Add more topics**

Can you add more topics to your chatbot, so that there are more types of question that it can answer?

**Provide alternate answers**

If someone asks the same question more than once, they’ll get the exact same answer every time.

Can you update your Scratch script so that it varies the answers each time a little? Or just starts the answer with “You’ve asked me this before, but”

**Say goodbye**

Add a “goodbye” bucket to your training examples and fill it with examples of ways to say goodbye. Can you update your Python script so it exits when it recognises someone saying that they’ve had enough?