

# Make me happy

In this project you will make a Python program that can react to what messages that you send to it.

If you compliment it, it will act happy. If you insult it, it will act sad.

You'll use machine learning to train the computer to recognise the difference between kind and mean messages, by giving it examples of each.



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.** You need two pictures: a picture of a happy face, and a picture of a sad face. Draw them, or if you prefer, find pictures on the Internet instead *Save the two pictures somewhere on your computer and remember where you put them. You'll need them later.*
- 2. Go to <u>https://machinelearningforkids.co.uk/</u> in a web browser
- **3.** Click on "Get started"
- 4. 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.
- 5. Click on "Projects" on the top menu bar
- **6.** Click the **"+ Add a new project**" button.
- 7. Name your project "make me happy" and set it to learn how to recognise "text".

Click the "Create" button

Mbou	t Projects	Worksheets	News	Help	Log Out	Language
				Sta	t a new machine learning project	
Project Name* make	me ha	арру				
Recognising *					What type of thing do you want to teach the computer to recognise? For words, sentences or puragraphs, choose "teat" For photos, diagrams and pictures, choose "mange" For sets of nutures or mainley choose, choose "manges" For secies and sconds, choose "tourds"	
Language Englist	1					•
					CREATE	CANCEL

 You should now see "make me happy" in the list of your projects. Click on it.



10. Click on "+ Add new label" and call it "kind things".Do that again, and create a second bucket called "mean things".



**11.** Click the "Add example" button in the "kind things" bucket, and type in the nicest, kindest compliment that you can think of.

About Projects Worksheets News Help Log Ou	A.	Language
Recognising text	as kind_things or mean_things	
< Back to project kind_thing Add exam You're a low	nple 22 / 1000 ADD CANCEL	
+ Add example	+ Add example	

**12.** Click on the "Add example" button in the "mean things" bucket, and type in the meanest, cruellest message that you can think of.

**13.** Repeat steps 11 & 12 until you've got at least **six** examples of each.

About Projects Worksheets News Help Log Out	Language
Recognising <b>text</b> as <b>kind_</b>	things or mean_things
Sack to project       kind_things       You're a lovely person       I appreciate all of the things you do	You smell I don't like you
Your hair looks great today You're my best friend	You're as dumb as a bag of rocks You're an idiot You smell bad I'm fed up with how useless you are
+ Add example	+ Add example

- **14.** Click on the **"< Back to project**" link. Then click on the **"Learn & Test**" button.
- **15.** 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.



- **16.** Wait for the training to complete. This might take a few minutes.
- 17. Once the training has completed, a Test box will be displayed. Try testing your machine learning model to see what the computer has learned.

Type something kind, and press enter. It should be recognised as kind. Type something mean, and press enter. It should be recognised as mean.

Test it with examples that you haven't shown the computer before. If you're not happy with how the computer recognises the messages, go back and add some more examples.

What have you done?	What's next?
You have trained a machine learning model to recognise when text is kind_things or mean_things. You created the model on Friday, April 12, 2019 11:56 PM. You have collected: • 6 examples of kind_things, • 6 examples of mean_things	Try testing the machine learning model below. Enter an example of text below, that you didn't include in the examples you used to train it. It will tell you what it recognises it as, any ow confident it is in that. If the computer seems to have used to recognise things correctly, then you can get us or atch and use what the computer has learned to make use me! If the computer used to the use of the computer of the computer used to train a page and collect some more examples. Once ou've done that, click on the button below to train a new use ine learning model and see what difference the extra examples will make!
Try putting in some text to see how it is recognised based training. You're a horrid person and I don't like you Recognised as <b>mean_things</b> with 87% confidence	Test

### **18.** Click the "< Back to project" link



### 20. Click on the "Python" button



## **22.** Copy the skeleton code displayed into a text editor. Save it as a file called **makemehappy.py**



#### **23.** Delete the bottom half of the code Everything from # CHANGE THIS to the end of the file, highlighted below. Delete it, as you won't need it.



**24.** You will be using a Python library called **Pillow** to display your pictures. If you don't already have it installed, you need to do this now. Ask your teacher or group leader to help. Instructions can be found at <u>https://pillow.readthedocs.io/en/stable/installation.html</u>

**25.** Add "from PIL import Image" to the top of your code



**26**. Add this to the end of your code.

It will ask you to type in a message.

The message you type in will be stored in a variable called "input". That variable will be given to the machine learning model that you have trained so it can try to recognize it.

If the machine learning model recognizes it as a compliment, your program will display a "You're so nice" message and show your picture of a happy face.

You need to update happy.png and sad.png to match the names of your pictures of faces.

```
input = raw_input("What do you want to tell me? > ")
recognized = classify(input)
label = recognized["class_name"]
if label == "kind_things":
    print ("You're so nice!")
    img = Image.open("happy.png")
    img.show()
else:
    print ("You're so mean!")
    img = Image.open("sad.png")
    img.show()
```

### **27.** Run your program with the command: **python makemehappy**.py

```
🗬 makemehappy.py 🗙
      from PIL import Image
      import requests
    def classify(text):
          key = "f0c8f9db-9fca-4317-bc0e-ccfb976c1d637fc341b2-dbc3-400d-b0ce-c2ab6faa9ffa"
          url = "https://machinelearningforkids.co.uk/api/scratch/"+ key + "/classify"
          response = requests.get(url, params={ "data" : text })
          if response.ok:
              responseData = response.json()
              topMatch = responseData[0]
              return topMatch
             response.raise_for_status()
      input = raw_input("What do you want to tell me? > ")
      recognized = classify(input)
      label = recognized["class_name"]
 26 if label == "kind_things":
          print ("You're so nice!")
          img = Image.open("happy.png")
          img.show()
          print ("You're so mean!")
          img = Image.open("sad.png")
          img.show()
```

### What have you done?

You've made a Python program that asks for the users input. It will then use the machine learning model that you trained to recognize whether what the user inputted was a compliment or an insult. This is often described as "sentiment analysis".

The more examples you give it, the better it should get at recognising messages correctly.