

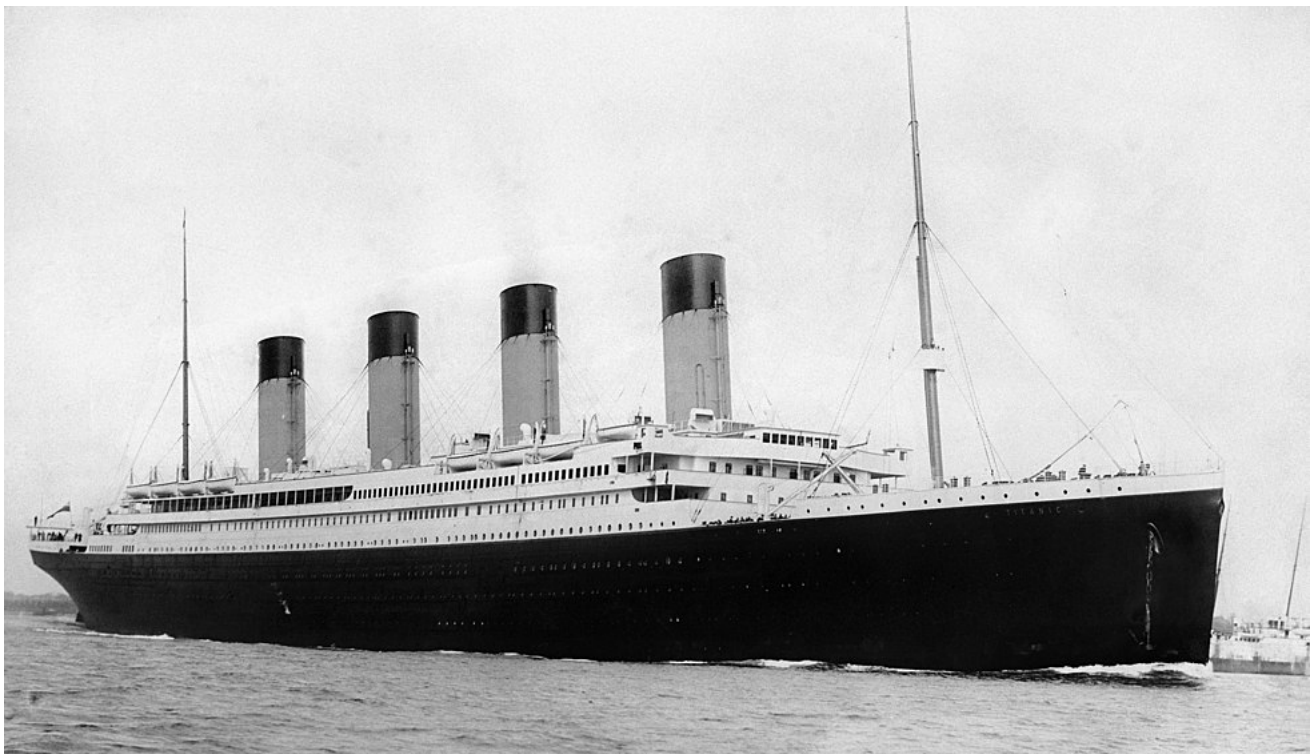


# Titanic

The *RMS Titanic* sank in the North Atlantic Ocean in 1912 during her maiden voyage from Southampton to New York. More than 1,500 of the 2,220 passengers and crew died when the ship sank.

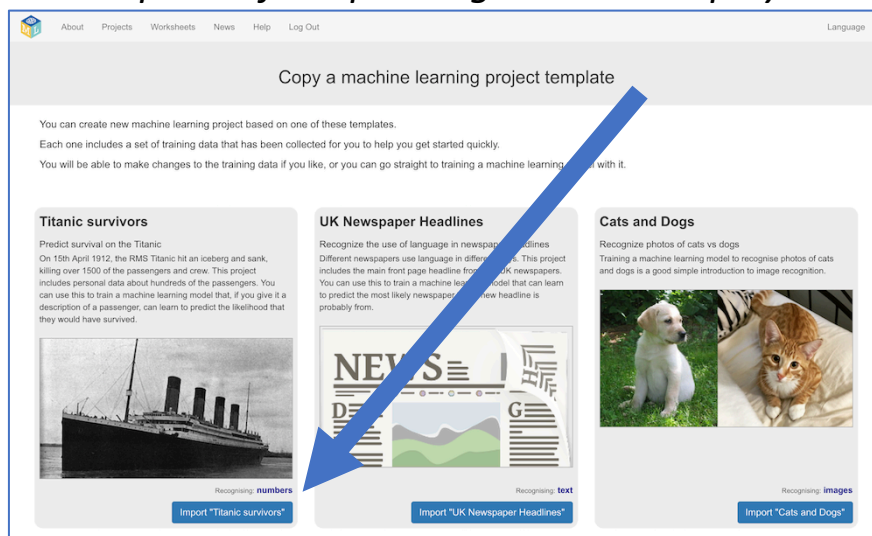
In this project you will make a Python program that can predict whether a specific passenger survived.

You'll use machine learning to train a predictive model using information about hundreds of examples of the passengers.

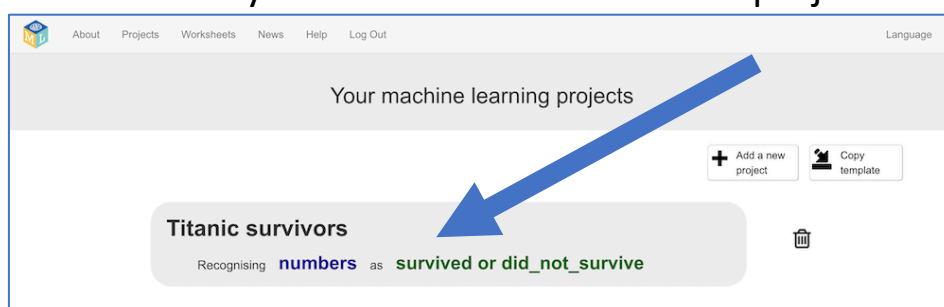


This project worksheet is licensed under a Creative Commons Attribution Non-Commercial Share-Alike License  
<http://creativecommons.org/licenses/by-nc-sa/4.0/>

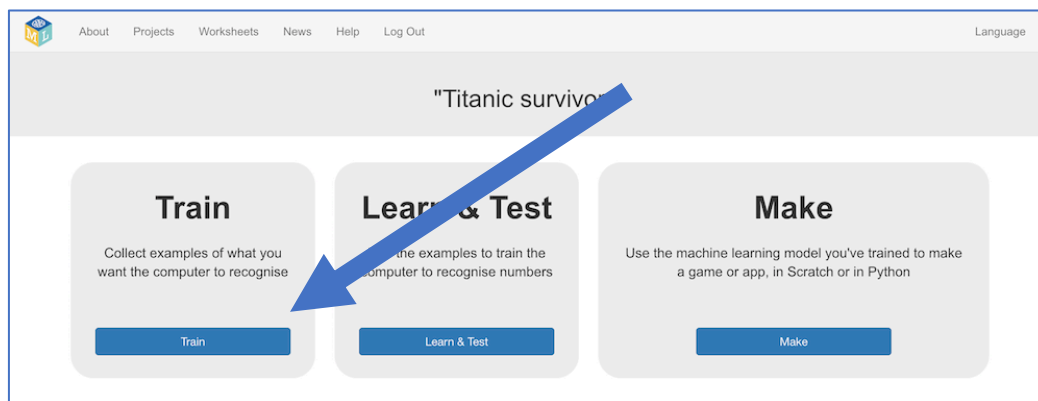
1. Go to <https://machinelearningforkids.co.uk/> in a web browser
2. Click on “**Get started**”
3. Click on “**Log In**” and type in your username and password  
*If you don't have a username, ask your teacher to create one for you.  
If you can't remember your username or password, ask your teacher or group leader to reset it for you.*
4. Click on “**Projects**” on the top menu bar
5. Click the “**Copy template**” button.
6. Find the **Titanic survivors** project template and click on the “**Import Titanic survivors**” button.  
*A description of the passenger data is displayed. Click “Import”.*



7. Click on your new “**Titanic survivors**” project



8. Start by displaying the training data that you've just copied. Click the **"Train"** button.



9. There are two training buckets.
- "survived"** contains examples of passengers/crew who survived.
- "did\_not\_survive"** bucket contains examples of people who did not.
- Each example gives you some information about the person:

ticket class	1, 2, 3	Did they have a first class, second class, or third class ticket?
gender	male, female	Were they male or female?
age	age in years	How old were they?
sibl. sp.	siblings or spouses	If they're a child, how many brothers and sisters were also on board? If they're an adult, 1 if they had a husband or wife with them, or 0 if not
par. ch.	parents or children	If they're a child, how many parents they had on board with them. If they're an adult, how many children they had on board with them.
ticket fare	cost in pounds	How much did their ticket cost?
embarked	Cherbourg, Southampton, Queenstown	Where did they get on board? Cherbourg (France), Southampton (England), or Queenstown (Ireland)?

The screenshot shows the ML4J web interface with the title 'Recognising numbers as survived or did\_not\_survive'. Below the title is a '< Back to project' link and an 'Add new label' button. The main content area is divided into two sections: 'survived' and 'did\_not\_survive'. Each section contains a grid of example cards. Each card displays the following information: ticket class, gender, age, sibl. sp., par. ch., ticket fare, and embarked. The 'survived' section has a counter of 288 examples, and the 'did\_not\_survive' section has a counter of 424 examples. Both sections have an 'Add example' button at the bottom.

- 10.** Review the training examples. Can you see any patterns in the data? Are there any common differences between passengers that survived, and passengers that didn't?

*For example, have you heard the phrase “Women and children first!” If that was really how the passengers and crew were got to the lifeboats, what sort of pattern would you expect to see in the training examples? Can you find that sort of pattern in the data in your project?*

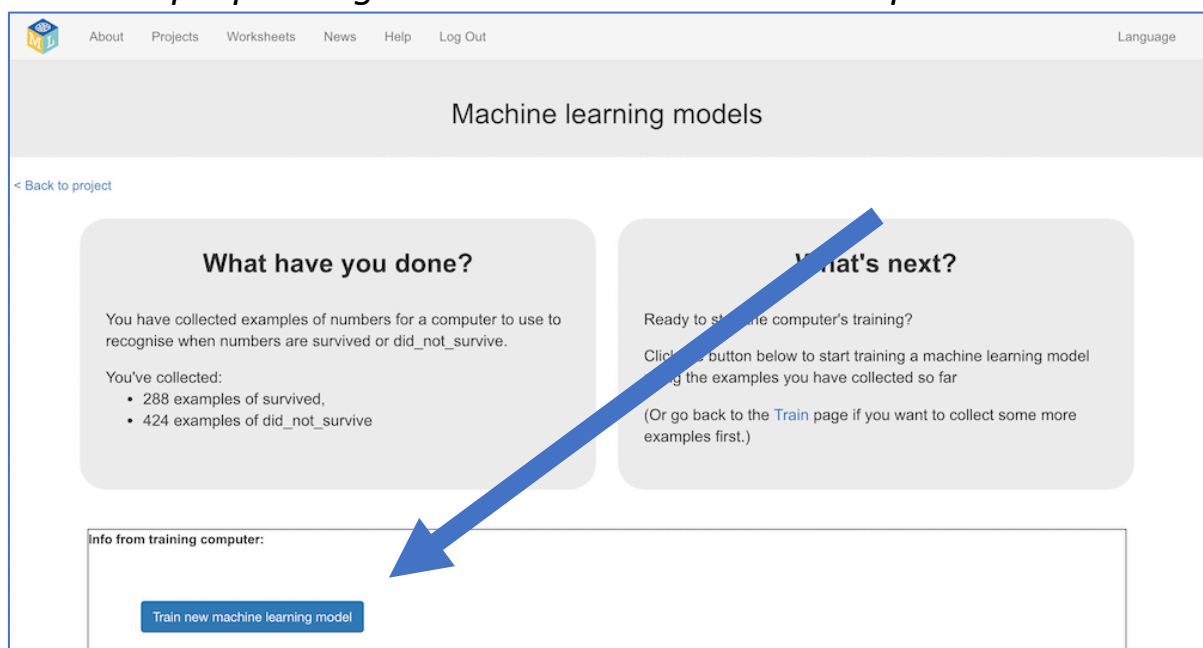
*What other patterns can you think of? For example, do you think passengers with more expensive tickets would have a better or worse chance of surviving the sinking?*

*Try to predict what sorts of patterns your machine learning model could learn to recognize. And then look to see if you can find that sort of pattern in your training data.*

- 11.** Click on the “< Back to project” link.

- 12.** Click on the “Learn & Test” button.

- 13.** Click on the “Train new machine learning model” button.  
*The example passenger data will be used to train a predictive model.*



We'll test your model using the characters from the 1997 movie, *Titanic*.

**Jack Dawson** (played by Leonardo DiCaprio)

Jack was a 20 year-old man.

He paid nothing for his third-class ticket, because he won it in a poker game.

He boarded RMS Titanic in Southampton, in England.

He had become an orphan when he was 15, and had no other family on board.

**Rose DeWitt-Bukater** (played by Kate Winslet)

Rose was a 17 year-old woman.

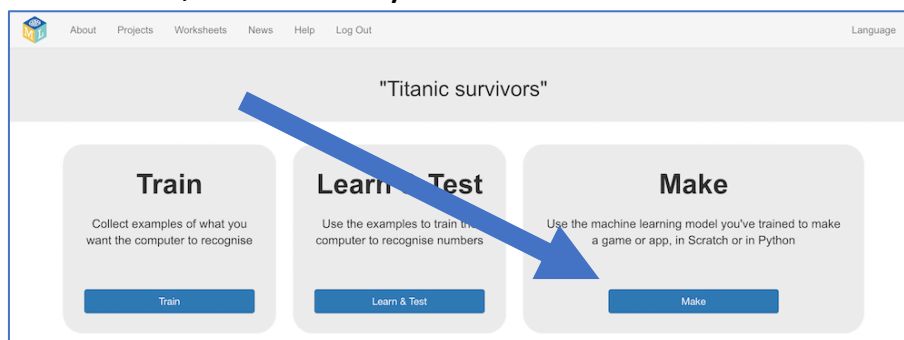
She had a first-class ticket. The film doesn't mention exactly how much it cost, but based on the room that she had, we can estimate her ticket cost £450.

She boarded the ship in Southampton, England.

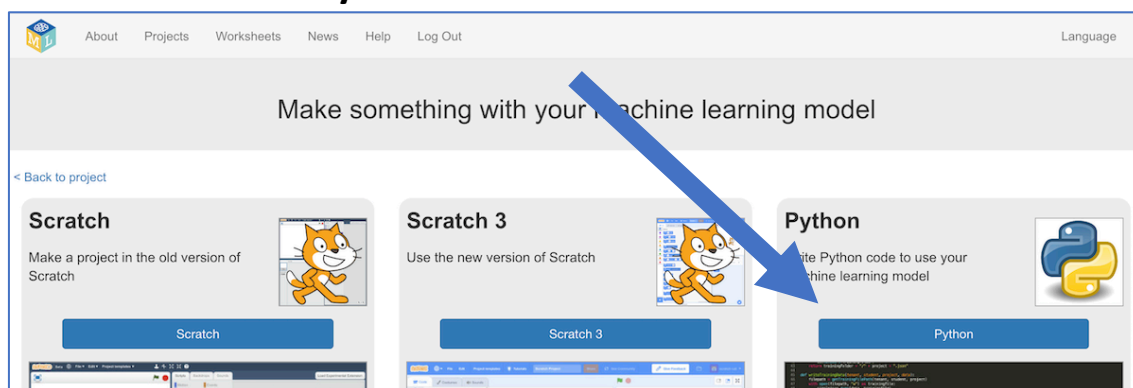
She came on board with her mother and her fiancé. She didn't have any brothers or sisters on board.

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

**15.** Next, we'll use Python. Click “Make”



**16.** Click on the **Python** button



- 17.** Fill in the information about the Jack Dawson character in the fields displayed on the left.  
*This will update the skeleton code on the right.*

Using machine learning in Python

< Back to project

You can use this code to submit numbers to the machine learning model that you've trained.

Enter the numbers below:

ticket class : 0

gender : male

age : 0

sibl. sp. : 0

par. ch. : 0

ticket fare : 0

embarked : Cherbourg

Running this code will print something like:

```
$ python yourscript.py
result: 'survived' with 81% confidence
```

If you've never used `requests` before, you might need to [install it first](#).  
Ask your teacher for help if you're not sure how to do that.

```
import requests

# This function will pass your numbers to the machine learning model
# and return the top result with the highest confidence
def classify(numbers):
    key = "46255e52-a356-4581-833f-3379fe13763a5102744e-af7f-4c50-bbce-dd7d7b1c4048"
    url = "https://machinelearningforkids.co.uk/api/scratch/" + key + "/classify"

    response = requests.get(url, params={ "data" : numbers })

    if response.ok:
        responseData = response.json()
        topMatch = responseData[0]
        return topMatch
    else:
        response.raise_for_status()

# CHANGE THIS to something you want your machine learning model to classify
data1 = 0
data2 = "male"
data3 = 0
data4 = 0
data5 = 0
data6 = 0
data7 = "Cherbourg"

demo = classify([ data1, data2, data3, data4, data5, data6, data7 ])

label = demo["class_name"]
confidence = demo["confidence"]

# CHANGE THIS to do something different with the result
print ("result: '%s' with %d%% confidence" % (label, confidence))
```

- 18.** Copy the updated skeleton code displayed into a text editor.  
Save it as a file called **jack.py**

- 19.** Find the last line of the skeleton code:
- ```
print ("result: '%s' with %d%% confidence" % (label, confidence))
```

Change the last line to this:

```
print ("%d%% sure that Jack %s" % (confidence, label))
```

- 20.** Run your program with the command: `python jack.py`  
*Your machine learning model will display its prediction for whether Jack survived the sinking of the Titanic.*  
*Does the prediction match what happened in the movie?*



## 21. Try doing the same for Rose's character in the movie.

Fill in the information about Rose's character in the webpage boxes.

Save the updated skeleton code to a file called **rose.py**

Update the last line of code to say:

```
print ("%d%% sure that Rose %s" % (confidence, label))
```

Run the program using the command: `python rose.py`

*Your machine learning model will display its prediction for whether Rose survived the sinking of the Titanic.*

*Does the prediction match what happened in the movie?*

### What have you done?

You've used a historical set of information about passengers and crew of the RMS Titanic to train a predictive model. That predictive model used the patterns in your historical training data to learn to recognize patterns in what sort of passengers survived or not.

When you looked for patterns in the training data, you hopefully thought about possible reasons to explain it. For example, younger passengers were more likely to survive because children would've been prioritized when boarding lifeboats. Computers won't have done that last bit. The computer will hopefully have seen the pattern between age and survival but won't have tried to explain it. A machine learning model will learn to recognize patterns in the data, but they won't care about why.

Finally, you made Python programs to test your predictive model, using information about fictional Titanic passengers from the movie, Titanic.

## 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 testing with other passengers**

Can you find any information about other Titanic passengers or crew that you can test your predictive model on?

### **Invent your own predictive model**

Can you think of another topic that you could collect numerical or multiple-choice values about?