



edunet  
foundation



## LAB MANUAL

# Unit VII – Natural Language Processing

## Lab Activity 3

### Sentiment Analysis using Machine Learning for kids

#### 1. Problem Statement

The objective of this project is to create an interactive character in Scratch that responds emotionally to different types of text input. The character will display a happy face in response to kind words and a sad face in response to mean words. Initially, this is achieved through rule-based programming, and then improved using machine learning to enable sentiment recognition.

#### 2. Objective

- To create a program that can detect sentiment (kind vs. mean) in user input and reflect the sentiment visually through the character's expressions.
- To explore the limitations of rule-based systems and the benefits of supervised machine learning in text recognition.
- To develop skills in Scratch and in using basic machine learning for text classification.

#### 3. Tools Used

- **Scratch 3:** A programming environment where the interactive character is created and programmed.
- **Machine Learning for Kids:** A platform used to train a model for recognizing kind and mean messages through examples.
- **Web Browser:** To access the Machine Learning for Kids website.

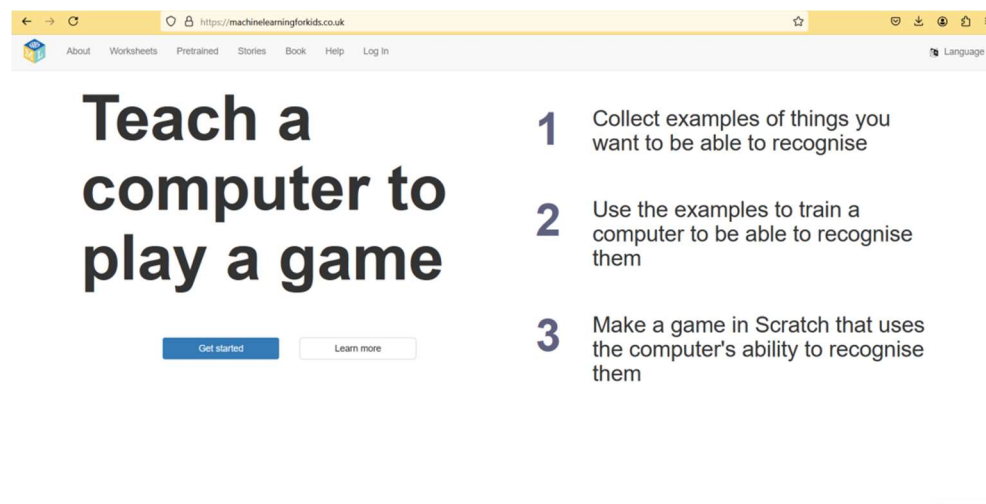
#### 4. Main Steps of Implementation

1. **Set Up the Project:**
  - Access [Machine Learning for Kids](#), log in, and create a new project named "Make Me Happy" with text recognition.
2. **Create the Character in Scratch:**
  - Open Scratch, delete the default sprite, and create a new sprite with three costumes labeled "not sure," "happy," and "sad."
  - Draw expressions on each costume: a straight mouth for "not sure," a smile for "happy," and a frown for "sad."
3. **Program Initial Reactions Using Rules:**
  - Write a basic Scratch script to recognize specific messages (e.g., compliments or insults) and switch the costume based on these messages.
  - Test by entering messages to see if the character responds correctly.
4. **Train the Machine Learning Model:**
  - Return to the Machine Learning for Kids project and add labels "kind things" and "mean things."
  - Provide examples of kind and mean messages (at least six of each) and train the model.

- Test the model with new examples to ensure it recognizes the messages correctly.
- 5. **Integrate Machine Learning into Scratch:**
  - Reopen Scratch, import the project, and update the script to use the machine learning model instead of rule-based responses.
  - Use the "recognize text" block to classify the input text as "kind" or "mean" and display the appropriate expression.
- 6. **Final Testing:**
  - Test the updated project by entering various messages to ensure the character correctly identifies new kind and mean messages and changes its expression accordingly.

## 5. Detailed Steps to Implement the "Sentiment Analysis" Project

1. **Access Machine Learning for Kids:**
  - Go to [Machine Learning for Kids](https://machinelearningforkids.co.uk) in a web browser.



- Click on "Get started," then log in using your username and password. If you don't have a username, you can sign up or opt for "Try without registering".
- 2. **Create a New Project:**
  - Click on "Projects" in the top menu, then select "+ Add a new project."
  - Name the project "Sentiment Analysis" and set it to recognize "text."
  - Click the "Create" button to set up the project.

Start a new machine learning project

Project Name \*

Sentiment Analysis

Project Type \*

recognising text

Language

English

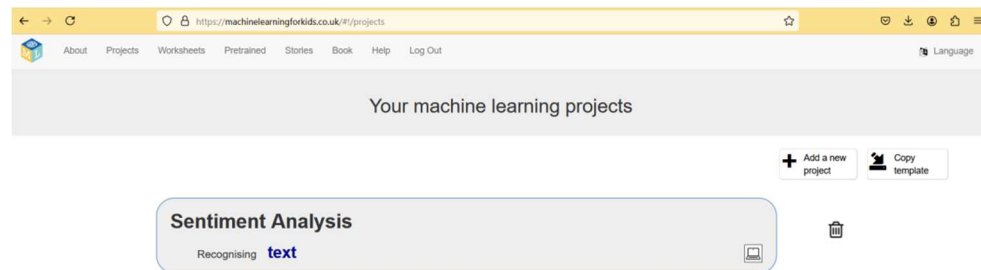
Storage \*

In your web browser

Where do you want to store this project?  
Storing in your web browser removes limits on how big your project can be.  
Storing in the cloud will let you access the project from any computer.  
(See "What difference does it make where a project is stored?")

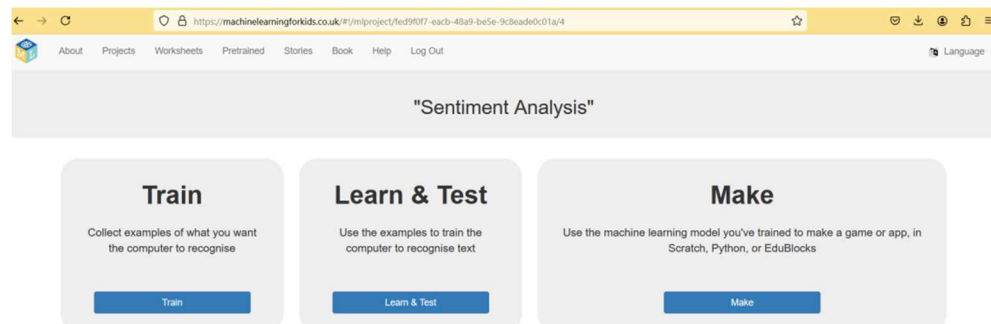
CREATE CANCEL

- Your Machine learning project is listed with its name and details. Click on the project.

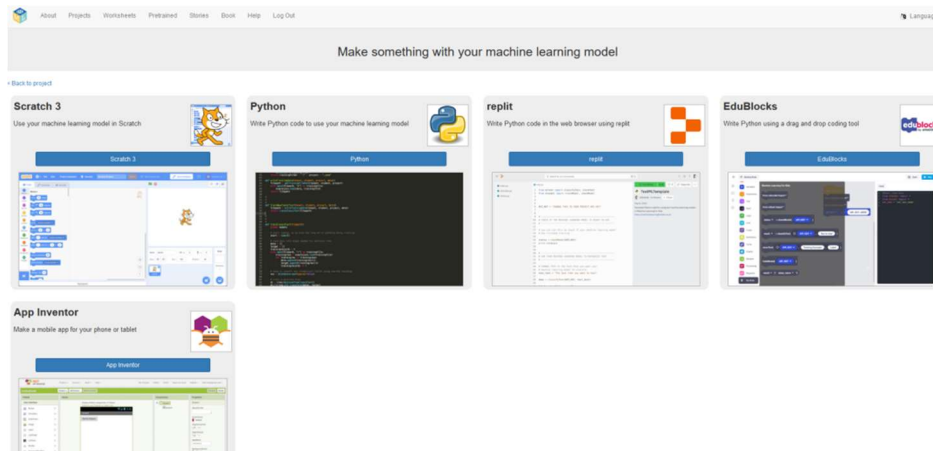


### 3. Open Scratch and Set Up Character:

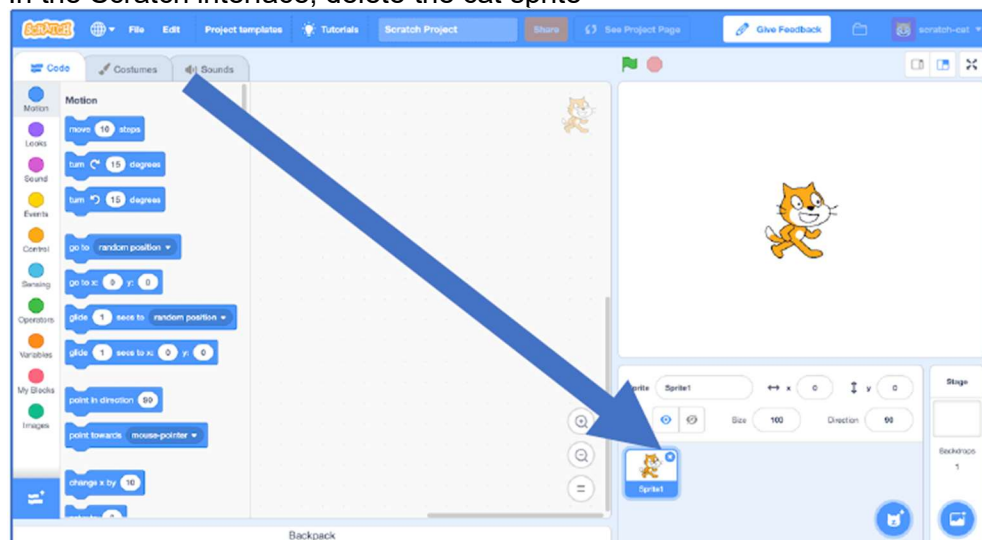
- In your Machine Learning for Kids project, click "Make"



- Select “Scratch 3” on your top left on the screen.

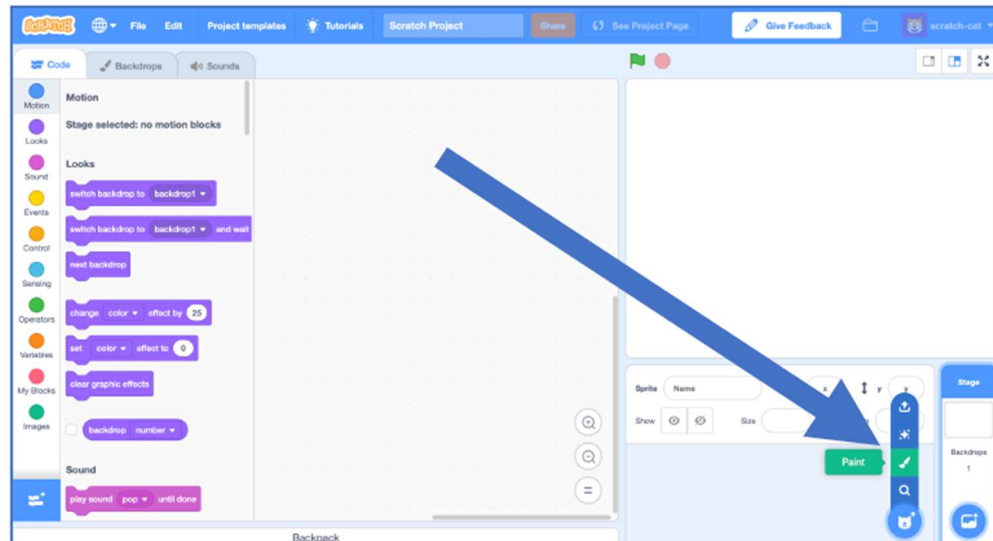


- Click “Scratch by itself”, the page will warn you that you haven’t done any machine learning yet, but clicking on Scratch by itself will launch Scratch anyway.
- In the Scratch interface, delete the cat sprite

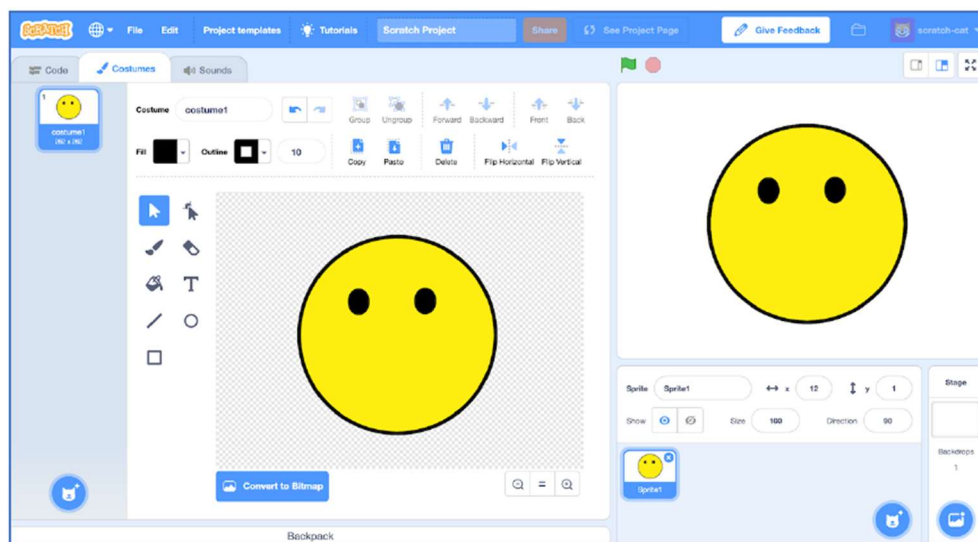


- Now, create a new sprite by clicking the Paint icon.

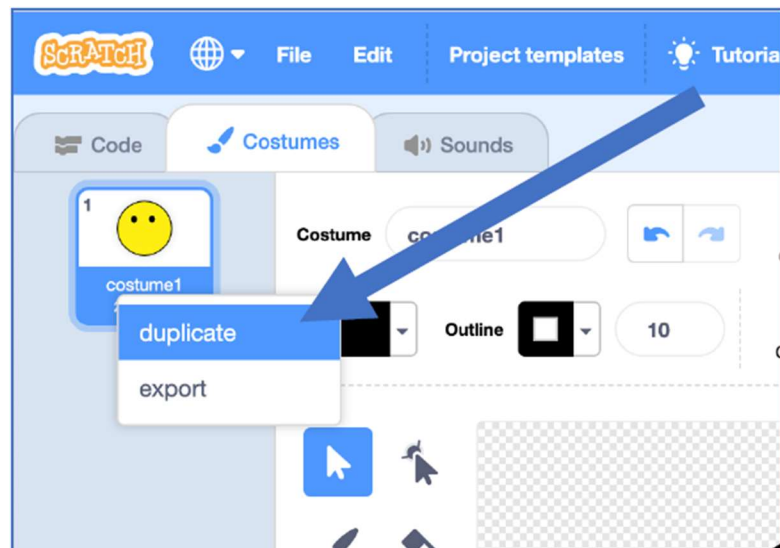




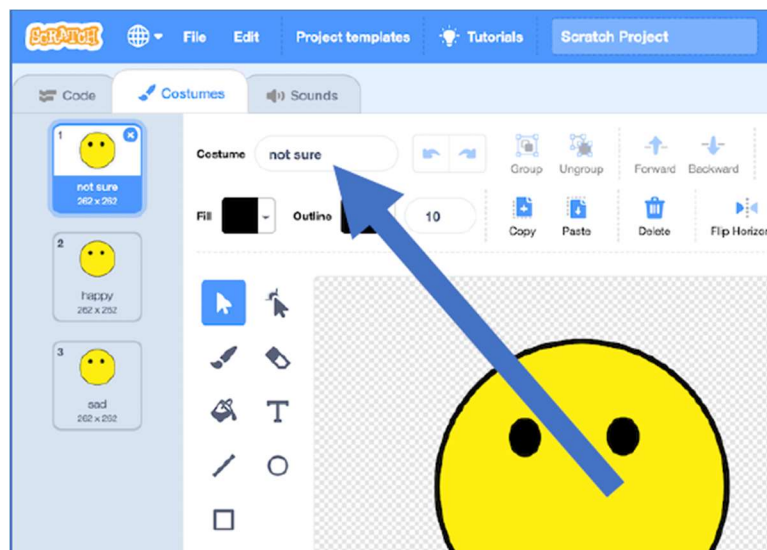
- In the Costumes tab, draw a face without a mouth.



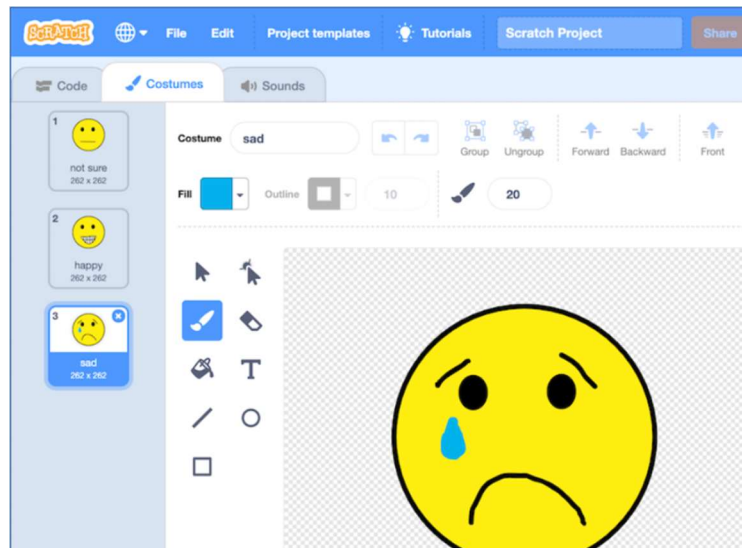
- Duplicate this face twice so you have three copies.



- Name these costumes “not sure,” “happy,” and “sad.”

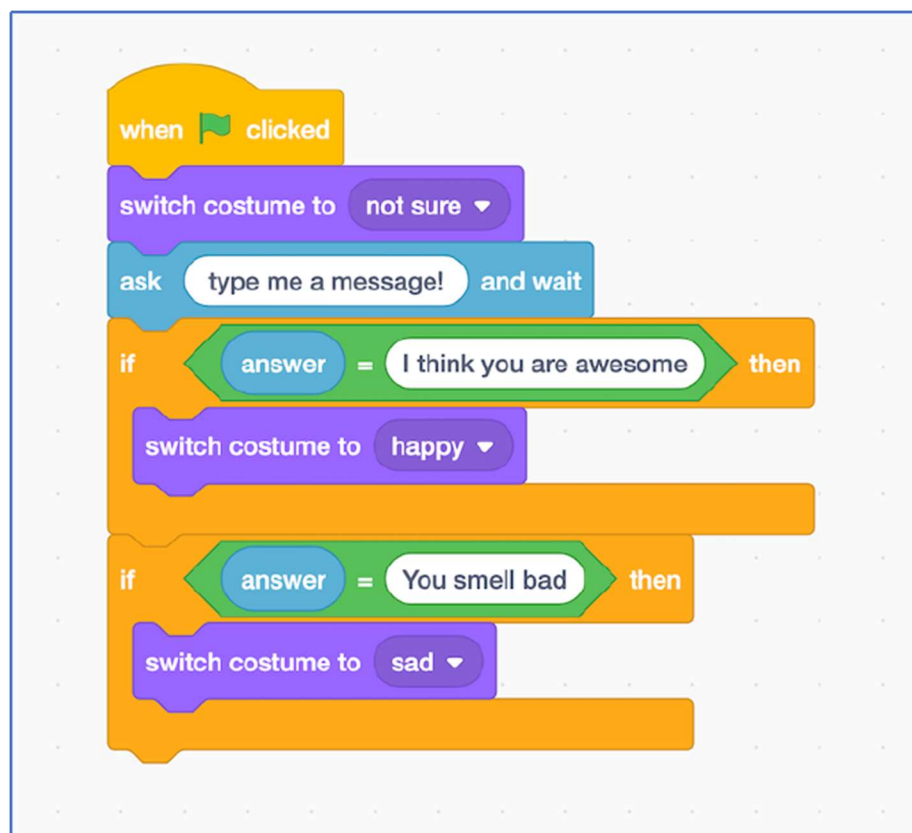


- Draw different expressions: a straight mouth for “not sure,” a smile for “happy,” and a frown for “sad.”



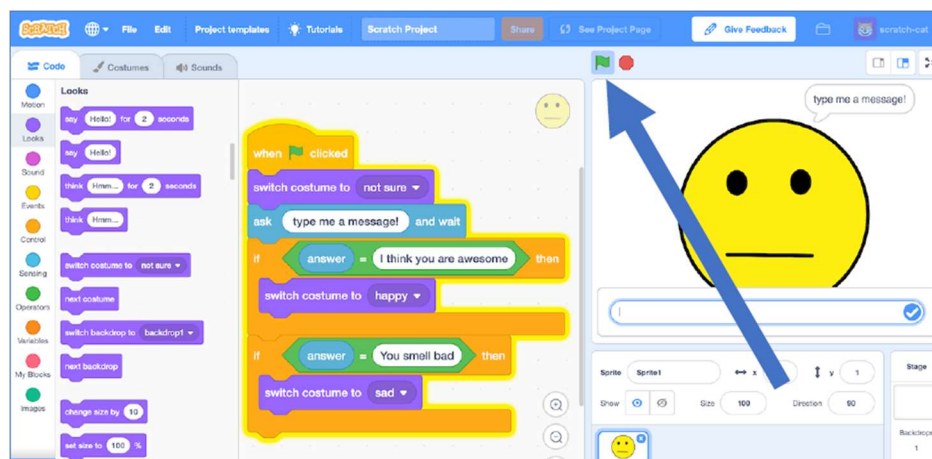
#### 4. Write the Initial Code in Scratch:

- Go to the “Code” tab and enter a script that allows the character to respond to specific messages (like compliments or insults).



- Save your project by clicking on File -> Save to your computer.
- #### 5. Test the Initial Rule-Based Approach:
- Click the green flag to test. Enter a kind message (e.g., “I think you are awesome”) to see the character smile.



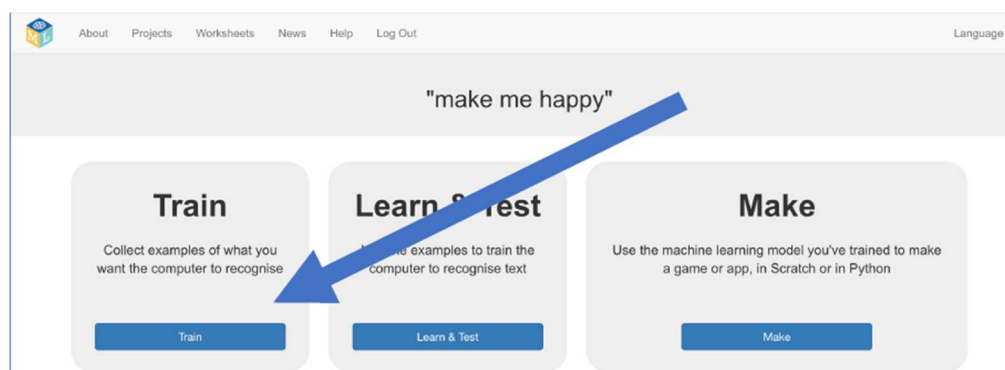


- Test a mean message (e.g., “You smell bad”) to see the character look sad.

This approach relies on hard-coded rules, so each new message requires additional code. In other words, so far you've created a character that reacts to what people type by following a set of rules. To make it respond to more messages, you would need to keep adding more "if" statements. The problem is that trying to list every possible message would take a very long time. Next, we'll try a better way—teaching the computer to understand messages on its own.

#### 6. Close Scratch and Start Training the Machine Learning Model:

- Close Scratch and return to your Machine Learning for Kids project.
- Click “Train” to begin training the model.



#### 7. Add Examples for Training:

- Create two labels: “kind things” and “mean things.”

Recognising **text** as **kind\_things** or **mean\_things**

< Back to project

+ Add new label

**kind\_things**

+ Add example

**mean\_things**

+ Add example

- Under each label, click “Add example” and enter several examples (at least six each) of kind and mean messages.

Recognising **text** as **kind\_things** or **mean\_things**

< Back to project

+ Add new label

**kind\_things**

You're a lovely person I appreciate all of the things you do

Your hair looks great today You're my best friend

I think you're amazing That jacket looks great on you

+ Add example

**mean\_things**

You smell I don't like you

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

- After adding the examples, click the “Learn & Test” button, then select “Train new machine learning model” to train the model.

Machine learning models

< Back to project

### What have you done?

You have collected examples of text for a computer to use to recognise when text is kind\_things or mean\_things.

You've collected:

- 6 examples of kind\_things,
- 6 examples of mean\_things

### What's next?

Ready to start the computer's training?

Click the button below to start training a machine learning model using the examples you have collected so far

(Or go back to the [Train](#) page if you want to collect some more examples first.)

Info from training computer:

[Train new machine learning model](#)

## 8. Test the Machine Learning Model:

- Once training is complete, use the Test box to see if the model can correctly identify kind and mean messages.

### What have you done?

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 1:05 PM.

You have collected:

- 6 examples of kind\_things,
- 6 examples of mean\_things

### What's next?

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, and how confident it is in that.

If the computer seems to have learned to recognise things correctly, then you can go to [Scratch](#) and use what the computer has learned to make a game!

If the computer is getting too many things wrong, you might want to go back to the [Train](#) page and collect some more examples

Once you've done that, click on the button below to train a new machine learning model and see what difference the extra examples will make!

Try putting in some text to see how it is recognised based on your training.

[Test](#)

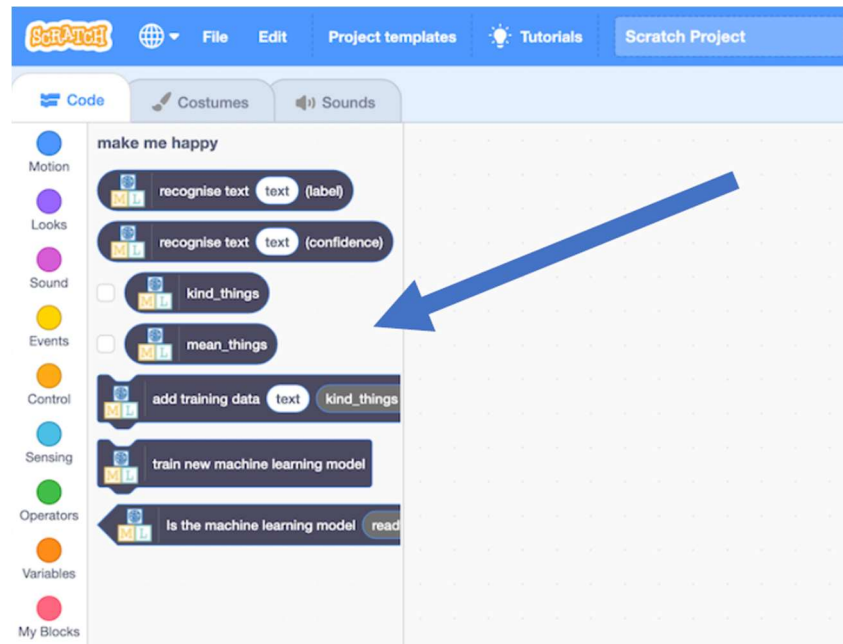
Recognised as mean\_things  
with 87% confidence

- If necessary, add more examples and retrain until the model performs as expected.

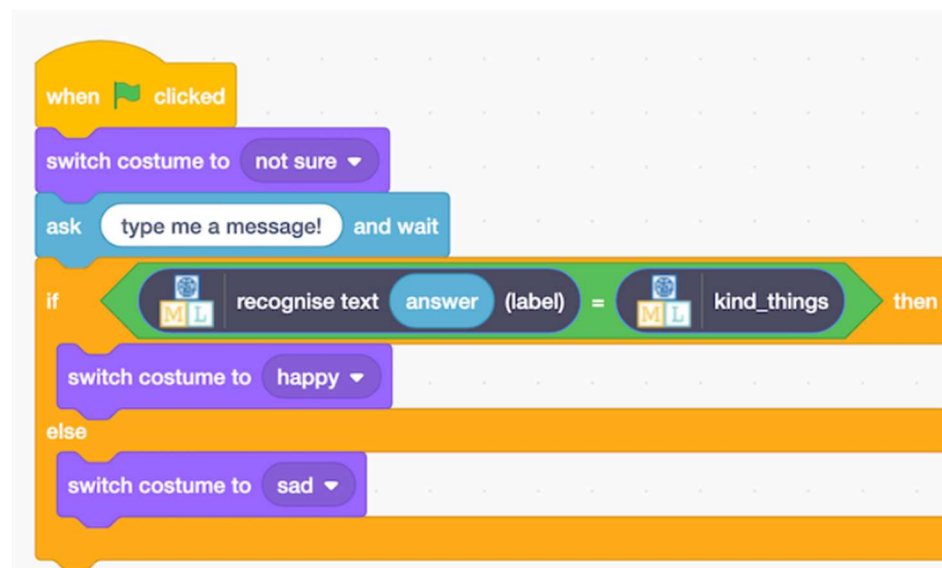
So far, you've started teaching the computer to recognize text as kind or mean. Rather than writing specific rules for each type of message, you're giving it examples to learn from. These examples are used to build a machine learning model. This process is known as "supervised learning" because you're guiding the computer's training. The computer picks up patterns in the examples, like word choice and sentence structure, and uses these patterns to identify new messages.

## 9. Integrate the Machine Learning Model into Scratch:

- Return to the project, click “Make,” then select “Scratch 3” and open the Scratch editor.



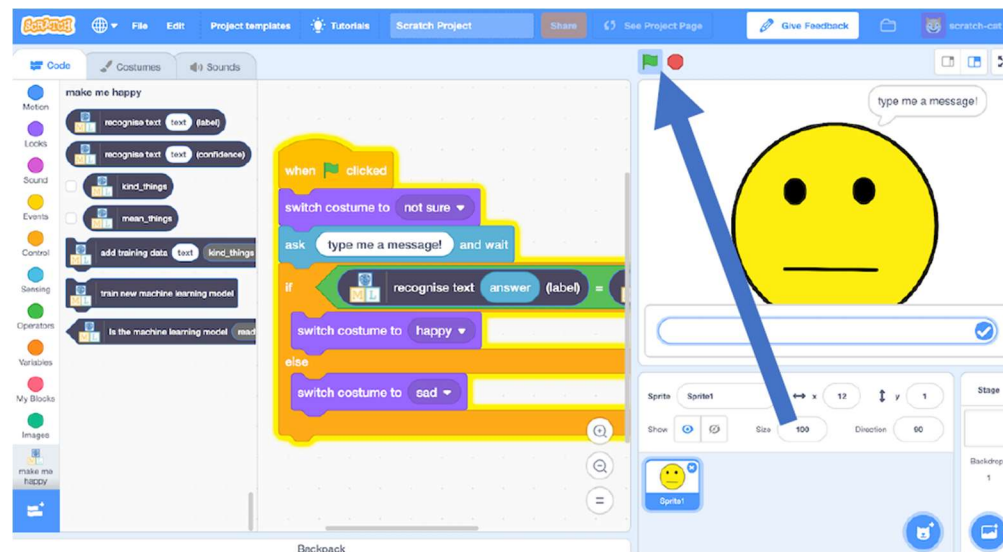
- Load your previously saved Scratch project, “Click on File -> Load from your computer.”
- Click on the “Code” tab and update the script to use the machine learning model’s “recognize text” block.



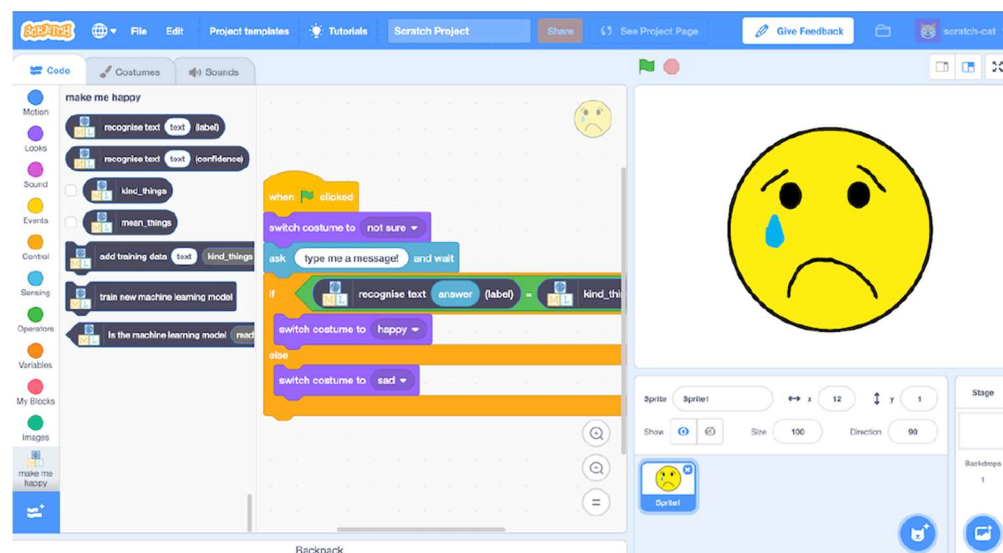
- This block will identify messages as “kind things” or “mean things” based on the training.

#### 10. Final Testing:

- Click the “green flag”



- Test the project with various messages to ensure the character displays the correct expressions. For an example, enter a kind message and press enter—the character should respond with a smile. Click the green flag again, then type a mean or unkind message and press enter—the character should appear sad. This should work even for messages that weren't part of your training examples.



- **The character should respond accurately to messages it hasn't been explicitly trained on.**

## 6. Conclusion

In conclusion, this project demonstrates how machine learning can enable a character to recognize and respond to various types of messages, even those it hasn't seen before. By training a model with examples of kind and mean statements, the character learns to identify patterns in language and react accordingly. This approach not only simplifies programming by reducing the need for numerous specific rules but also shows how supervised learning can help computers interpret text in a more human-like way. The more examples provided, the more accurately the



character can recognize and respond to new messages, making interactions more engaging and realistic.

## 7. Reference:

[1] <https://machinelearningforkids.co.uk/#!/worksheets>