

Student Performance Prediction

Objective	1
Dataset	1
Education, training, learning and testing.....	2
Implementation	14

Objective

- The goal of this project is to predict whether a student will pass or fail based on academic, behavioral, lifestyle, and socioeconomic factors that affect student performance.

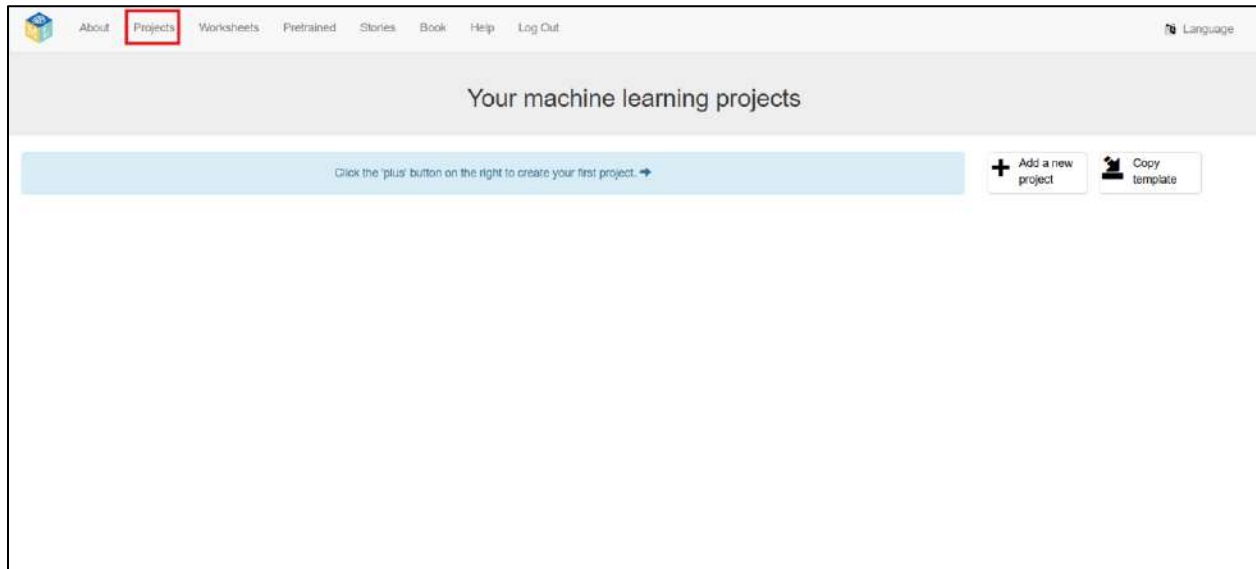
Dataset

- The original dataset is placed here: [student performance prediction dataset](#)
- We select 10 attributes/columns of the original dataset that are most relevant to our project
- The following table summarizes the selected columns:

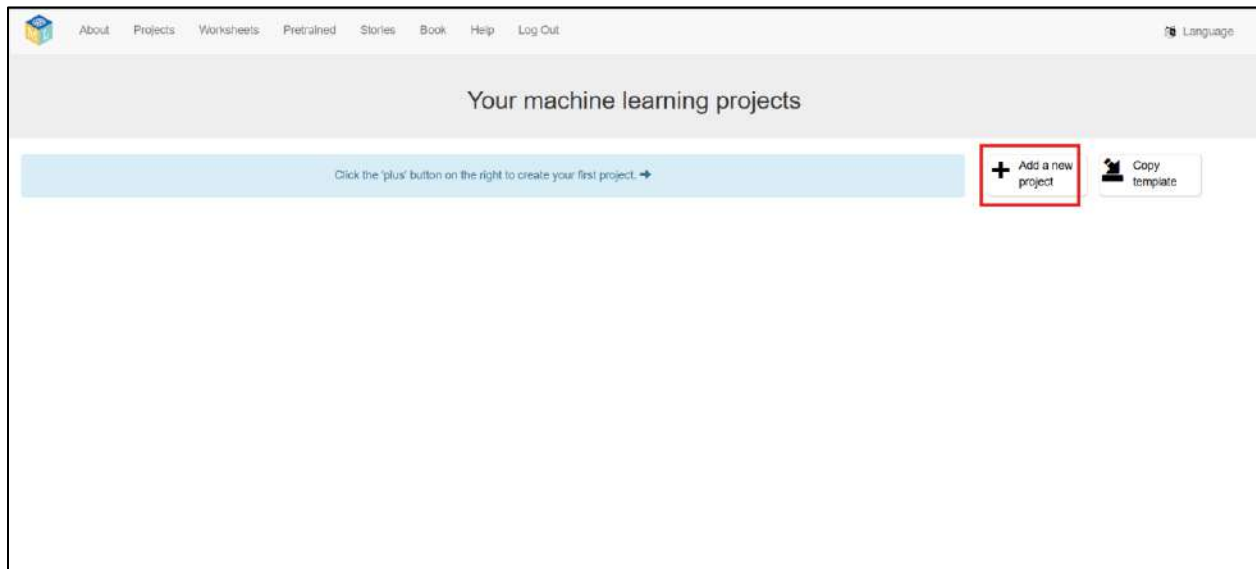
Attribute Name	Description	Possible values
study_hours	Number of hours studied by the student per day (number)	0 – 12 (float)
attendance	Percentage of courses attended (number)	0 – 100 (%)
prev_scores	Scores of previous exams or exams (number)	0 – 100
motivation	Level of student motivation (multiple-choice)	Low / Medium / High
sleep_hours	Average hours of sleep per night (number)	3 – 10
tutor_sess	Number of tutorials attended (number)	0 – 20
teacher_qual	Quality of teaching (multiple-choice)	Low / Medium / High
parent_inv	Level of parental involvement (multiple-choice)	Low / Medium / High
resources	Access to study materials/resources (multiple-choice)	Low / Medium / High
peer_infl	Influence of peers on studies (multiple-choice)	Negative / Neutral / Positive

Education, training, learning and testing

- Follow the link: <https://machinelearningforkids.co.uk/>
- Log in with your account or click "Try it now"
- Go to the "Projects" tab



- Click "Add a new project"



- Give your project a name
- As "Project Type" select "recognising numbers"
- Click "ADD A VALUE"

Start a new machine learning project

Project Name*
Students Performance

Project Type*
recognising numbers

ADD A VALUE

Start to describe the values that you'll include with each example to train the computer with by clicking the 'Add a value' button

Storage*

CREATE CANCEL

- We will create 10 values for each attribute/column in the table above
- In the "Value 1" field we type "study_hours" the name of the first column
- As "Type of value" we choose "number" because the possible values are numbers
- We will click on "ADD ANOTHER VALUE" to add a new value

Please choose where you would like to store your project

Project Name*
Students Performance

Project Type*
recognising numbers

Value 1*
study_hours

Type of value*
number

If study_hours can be described as numbers, choose "number".
If it can be described as choosing from a few options, choose "multiple choice".

ADD ANOTHER VALUE

Storage*

CREATE CANCEL

- If "Type of value" is not a number, then we select "multiple-choice".
- As "Value 4" we type "motivation". As "Type of value" we select "multiple-choice"
- A "Choices" field appears and we will type in the possible values, as follows:
 - **Low** , then click **enter**
 - **Medium** , and then click **enter**
 - **High** , and then click **enter**

Students Performance

Project Type*

recognising numbers

Value 1*	Type of value*	Value 2*	Type of value*	Value 3*	Type of value*
study_hours	number	attendance	number	prev_scores	number
Value 4*	Type of value*	Type in another choice to use in your multiple-choice list, then press Enter			
motivation	multiple-choice	Choices: <input type="radio"/> Low <input type="radio"/> Medium <input type="radio"/> High add a choice			

ADD ANOTHER VALUE

- When we add all the values, our screen should look like the one below:

recognising numbers

Value 1*	Type of value*	Value 2*	Type of value*	Value 3*	Type of value*
study_hours	number	attendance	number	prev_scores	number
Value 4*	Type of value*	Value 5*	Type of value*	Value 6*	Type of value*
motivation	multiple-choice	sleep_hours	number	tutor_sess	number
Value 7*	Type of value*	Value 8*	Type of value*	Value 9*	Type of value*
teacher_qual	multiple-choice	parent_inv	multiple-choice	resources	multiple-choice
Value 10*	Type of value*	Type in another choice to use in your multiple-choice list, then press Enter			
peer_infl	multiple-choice	Choices: <input type="radio"/> Positive <input type="radio"/> Neutral <input type="radio"/> Negative add a choice			

ADD ANOTHER VALUE

*You are not allowed more than 10 values in a project.

- Select "**Storage**" -> "**In your web browser**"
- Click "**CREATE**"

peer_init multiple-choice

Choices:

Positive Neutral Negative

add a choice

ADD ANOTHER VALUE

You are not allowed more than 10 values in a project.

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

- Click on the title of the project

About Projects Worksheets Pretrained Stories Book Help Log Out Language

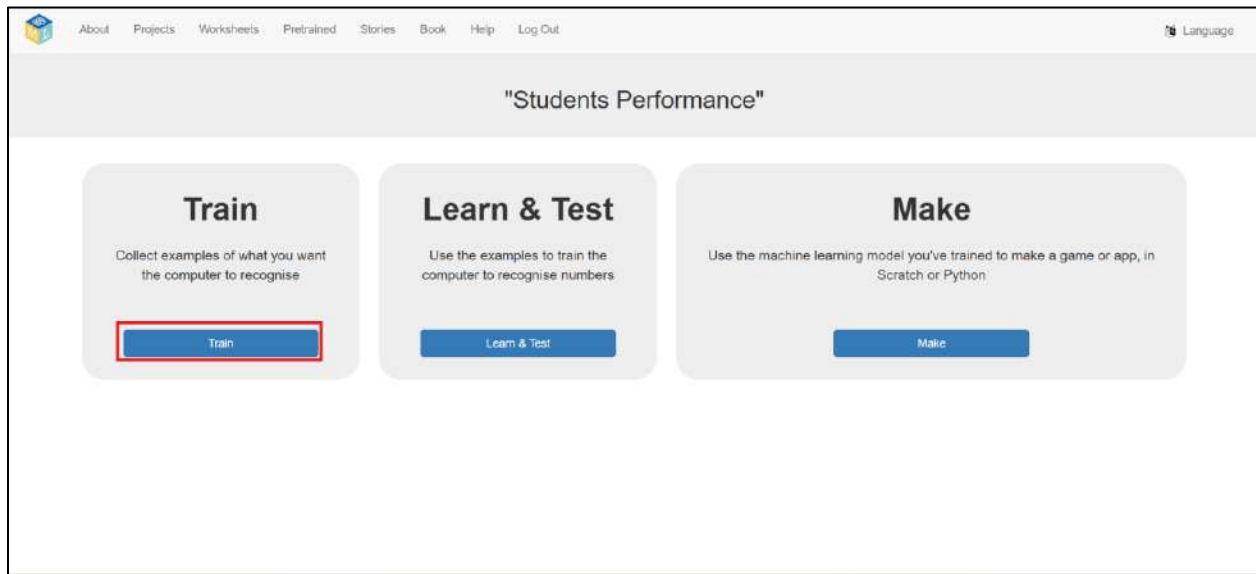
Your machine learning projects

+ Add a new project Copy template

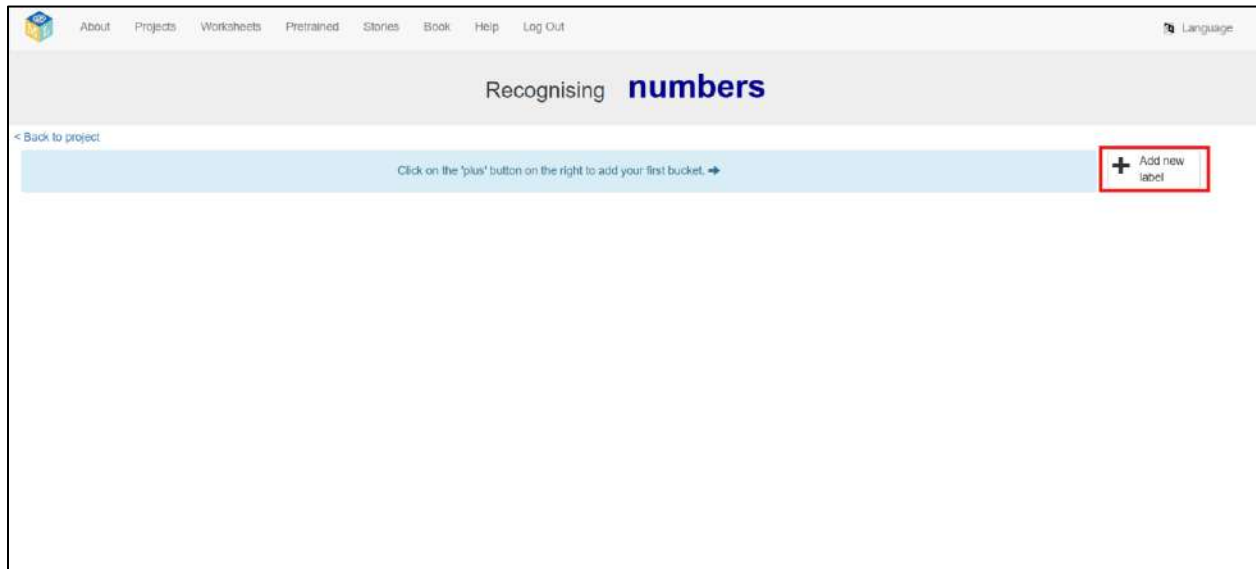
Students Performance

Recognising numbers

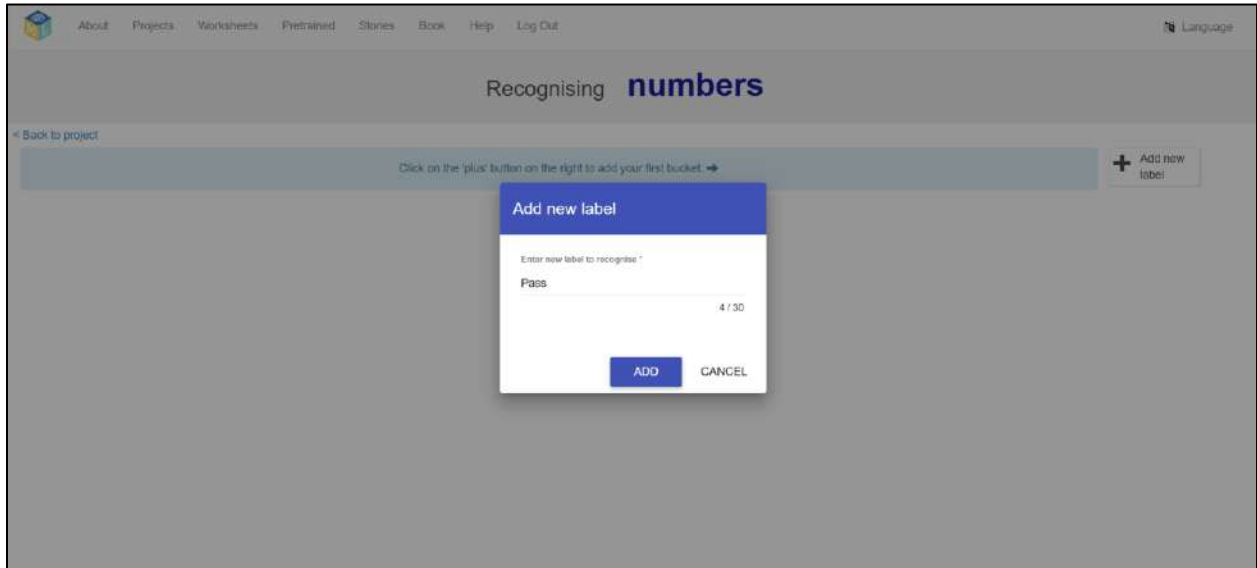
- Click "**Train**"



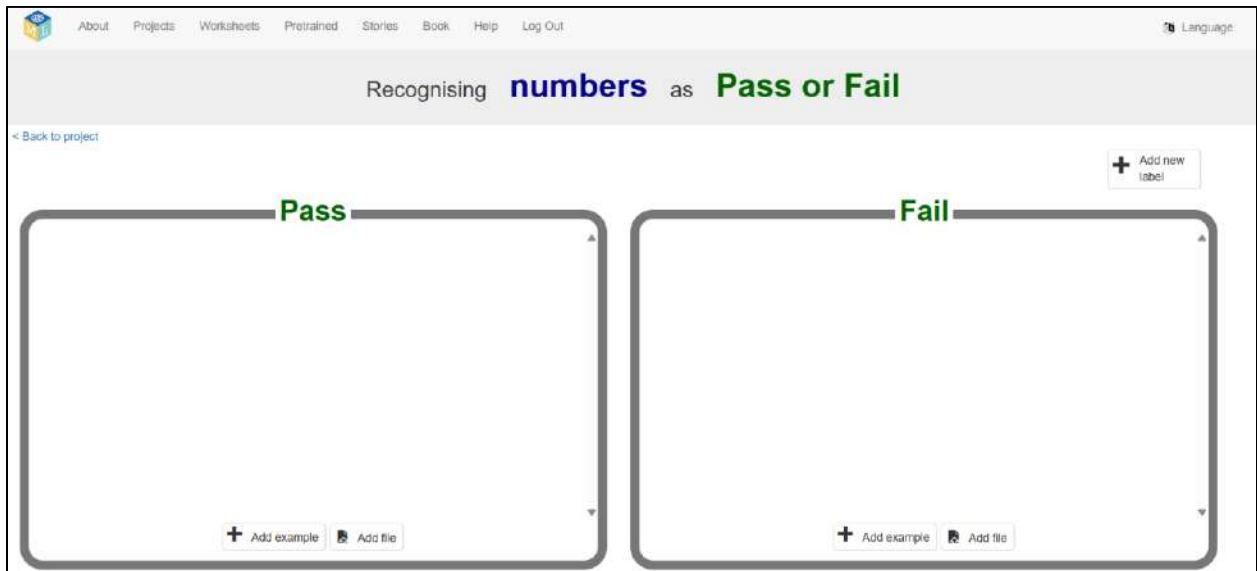
- Our model prediction for a student will be "**Pass**" or "**Fail**", in which case we will create two tags/categories
- Click on "**Add new label**"



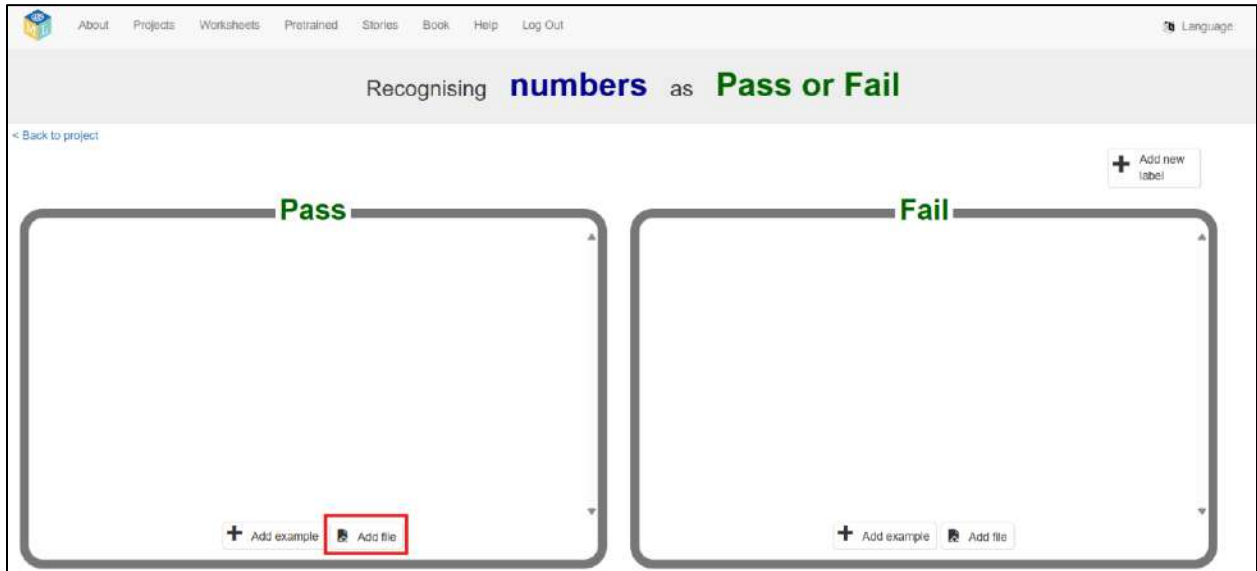
- In the "Enter new label to recognise" field, type "Pass" and click "ADD"



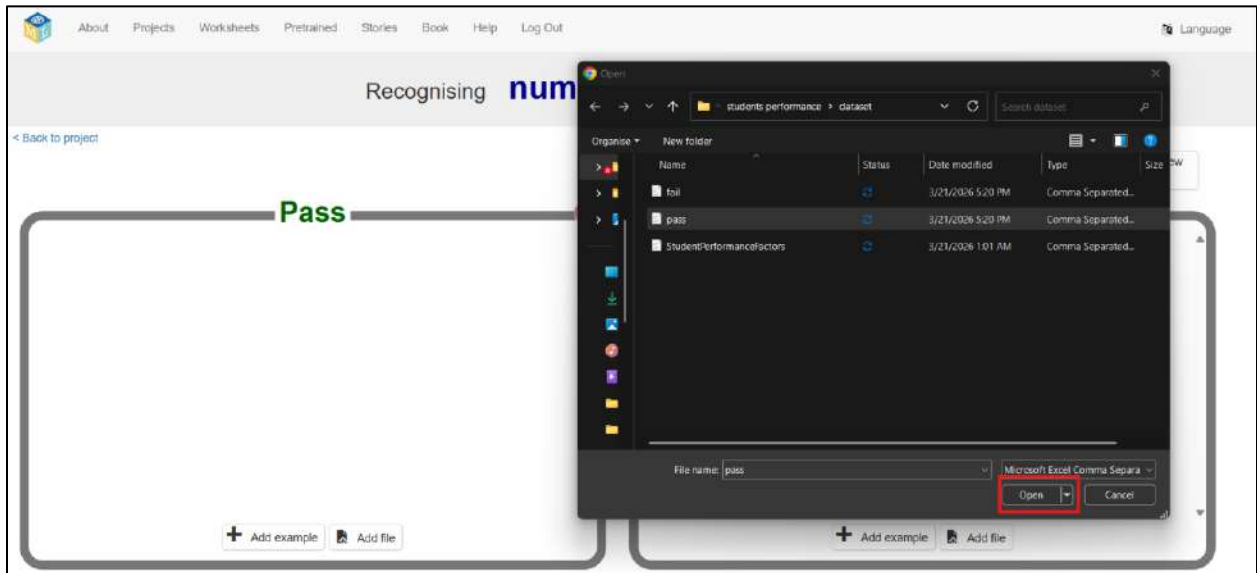
- Click "Add new label" again and create a "Fail" tag.
- After this step, your screen should look like this:



- In the [Training dataset](#) folder you will find two files:
 - **fail.csv**
 - **Pass.csv**
- Download these files to your computer
- In the "Pass" label, click on "Add file", select "pass.csv"



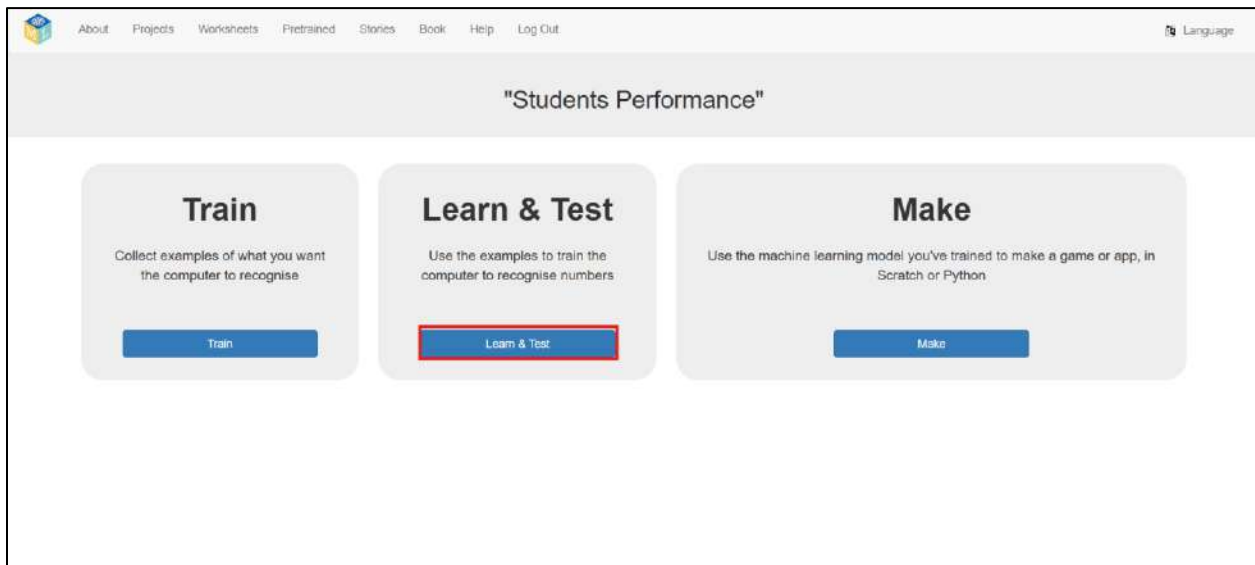
- Click "Open"



- Do the same for the "Fail" tag, but this time select "fail.csv"
- After these steps, your screen should look like the one below:



- Click on "Back to project"
- Click on "Learn & Test"



- Click on "Train new machine learning model"

Machine learning models

< Back to project

What have you done?

You have collected examples of numbers for a computer to use to recognise when numbers are Pass or Fail.

You've collected:

- 50 examples of Pass,
- 50 examples of Fail

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](#)

- After a few seconds your screen should look like the one below:

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 numbers to see how it is recognised based on your training.

study_hours	<input type="text"/>
attendance	<input type="text"/>
prev_scores	<input type="text"/>
motivation	Low ▾
sleep_hours	<input type="text"/>
tutor_sess	<input type="text"/>
teacher_qual	Low ▾
parent_inv	Low ▾
resources	Low ▾
peer_infl	Positive ▾

Test
Describe your model!

Info from training computer:

Model started training at: Saturday, March 21, 2020 4:31 PM

- Click on "Describe your model"
- This page explains the main ideas of a decision tree and shows the decision tree of your model

About Projects Worksheets Pretrained Stories Book Help Log Out Language

Understand your machine learning model

[← Back to model](#)

The technique used to create your machine learning model is called a **Decision Tree Classifier**.

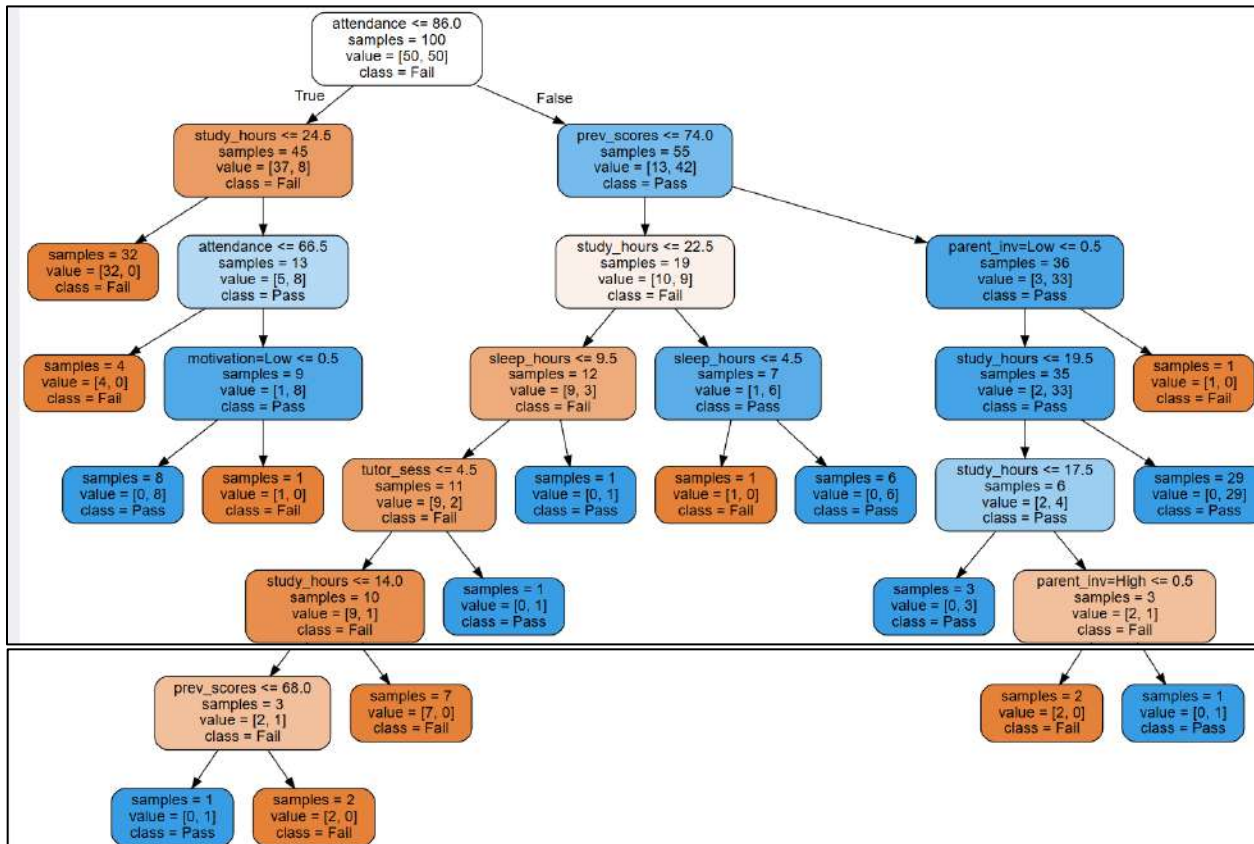
This is not the only way to train a machine learning model. We're using it because it's very quick and easy to train, and it is one of the easiest techniques to understand. This page shows you the decision tree that was created based on the training examples that you have collected.

When you test your model, the computer starts at the top of the tree, and follows a path until it reaches the bottom. The **class** at the bottom of the tree is the prediction that the machine learning model makes.

At each box in the tree, it reads the test described at the top of the box. If your test values pass the test described in the box, it follows the arrow to the left. If it doesn't pass the test, it follows the arrow to the right.

The **samples** shown in each box tells you how many examples in your training data matches that part of the decision tree.

The **value** shown in each box tells you how many examples in your training data passed the test shown at the top (following the left arrow) and how many examples didn't pass the test at the top (following the right arrow).



- You can test your model by filling in the fields on the right side

Try out your machine learning model to see how it uses the decision tree to make predictions

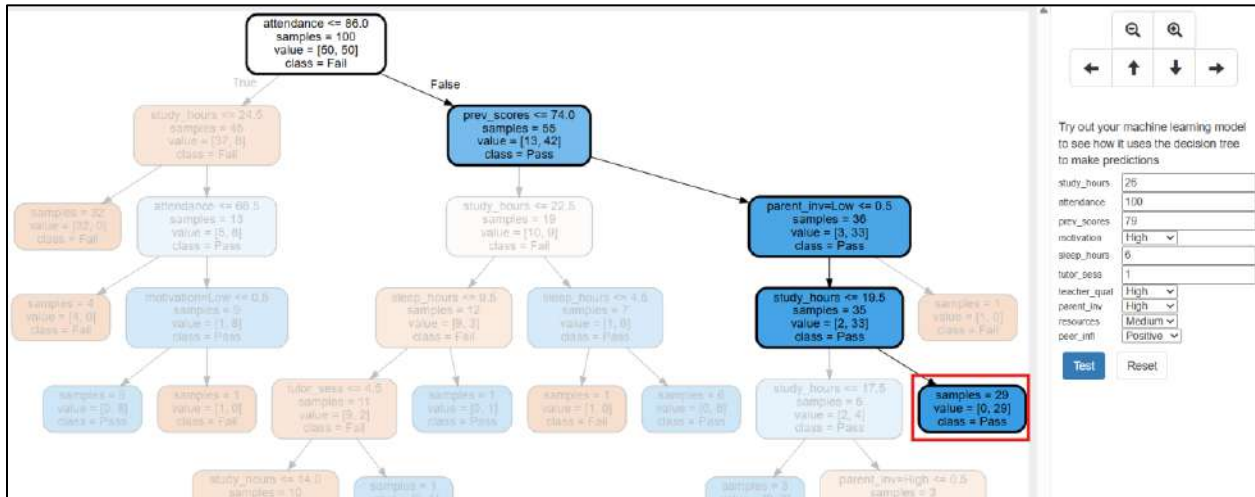
study_hours	<input type="text"/>
attendance	<input type="text"/>
prev_scores	<input type="text"/>
motivation	Low <input type="button" value="v"/>
sleep_hours	<input type="text"/>
tutor_sess	<input type="text"/>
teacher_qual	Low <input type="button" value="v"/>
parent_inv	Low <input type="button" value="v"/>
resources	Low <input type="button" value="v"/>
peer_infl	Positive <input type="button" value="v"/>

- In the [Test Dataset](#) folder you can find the "test.csv" file
 - This contains data that our model has never seen before, i.e. this data has not been used to train our model
 - The columns are the same as the training data "pass.csv" and "fail.csv", except for the last column "label" which contains the expected output of our model
- We will add the values of the first row of "test.csv" to the empty fields on the right side of the screen

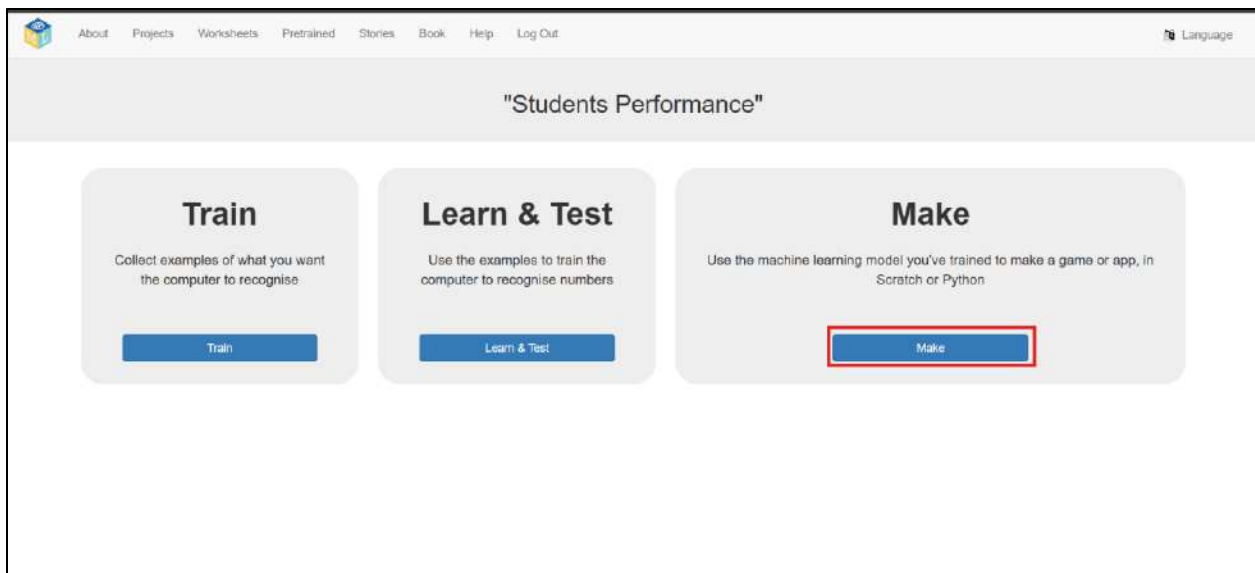
The screenshot shows a code editor window with a CSV file named 'test.csv' open. The first row of the CSV is highlighted, containing the following data: 26, 100, 79, High, 6, 1, High, High, Medium, Positive, Pass. A blue box highlights the first two columns (26, 100) and the last column (Pass). A blue box also highlights the condition 'prev_scores <= 74.0' in the code editor. An arrow points from the word 'False' to the 'Pass' label in the CSV row.

On the right side of the screenshot, the machine learning model testing interface is shown. The fields are filled with the values from the first row of the CSV: study_hours (26), attendance (100), prev_scores (79), motivation (High), sleep_hours (6), tutor_sess (1), teacher_qual (High), parent_inv (High), resources (Medium), and peer_infl (Positive). The 'Test' button is highlighted.

- Click on **"Test"**
- As we can see, our model predicts **"Pass"** which is correct since the column **"label"** has the value **"Pass"**

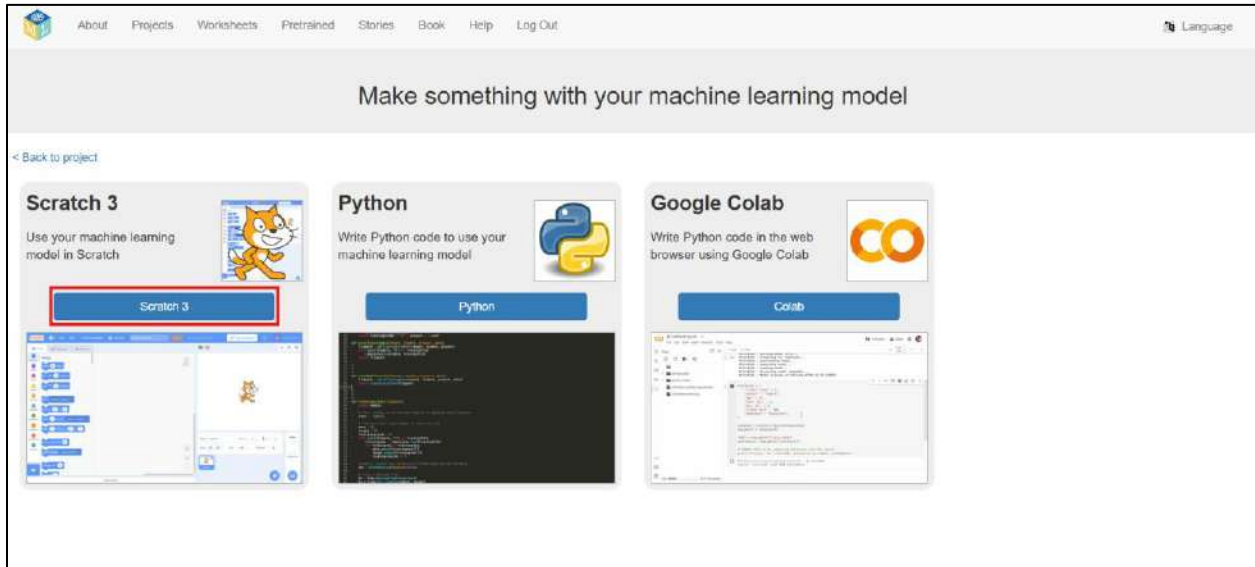


- Click **"Back to model"** (top left)
- Click on **"Back to project"** (top left)
- Click **"Make"**

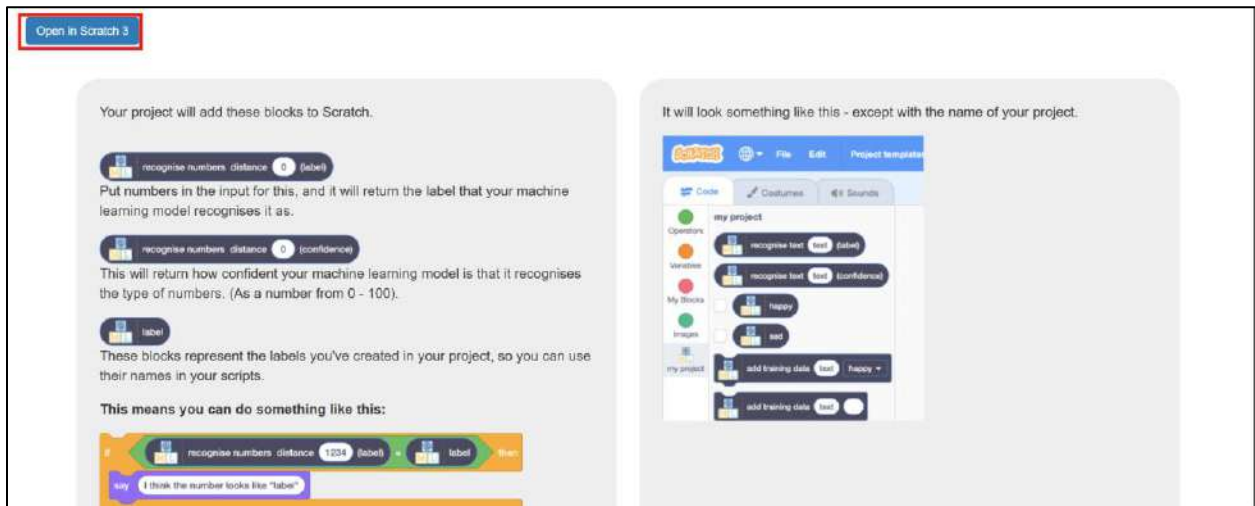


Implementation

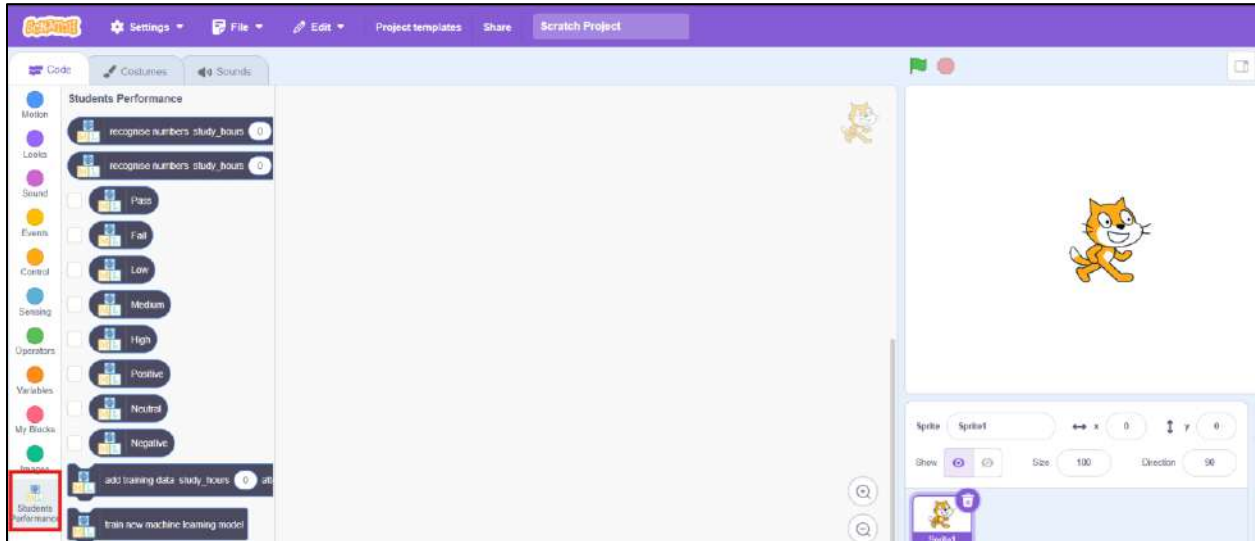
- Click on "Scratch 3"



- On this page you can find instructions on how to do a test of your model using scratch
- Click on "Open in Scratch 3"



- The "**Students Performance**" tab contains the commands related to the machine learning model that you have created



- We will create a block of code in scratch 3 to test the first line of "**test.csv**"
- The cat will tell the model's prediction
- Recreate the code block below, the colors will guide you to the correct choice of command
 - We will use the first command "**recognise numbers ... (label)**"
 - Be careful, there is a very similar command "recognise numbers ... (confidence)"

