Developer Student Clubs
University of Auckland

Machine Learning Workshop

The Colab Notebook

Insert notebook link here.

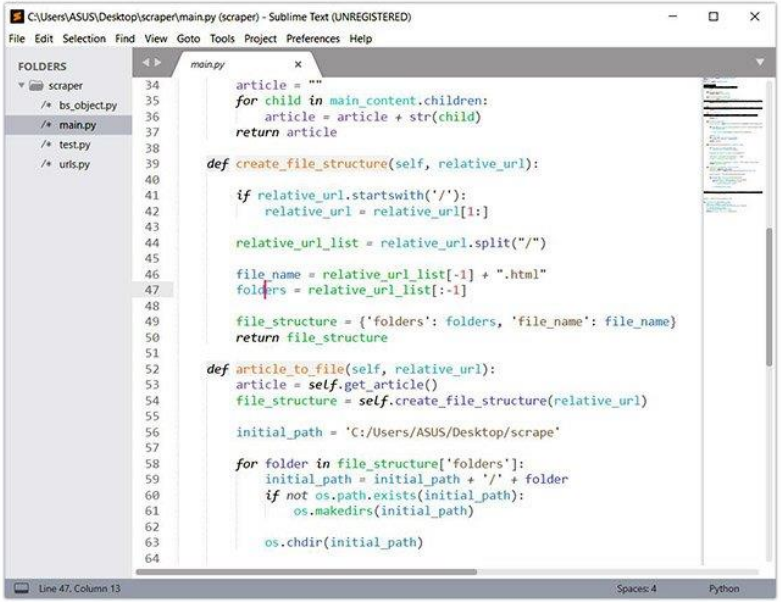
https://go.dscuoa.nz/mlcolab

https://colab.research.google.com/drive/1cBszWcDJP_9oe4K89GUhKjVMV5MxDXem?usp=sharing

Developer Student Clubs

What is Google Colab?

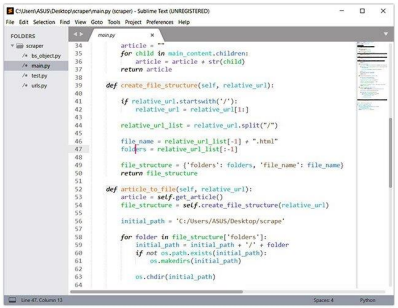
An introduction to the running theme of thievery



Developer Student Clubs

What is Google Colab?

An introduction to the running theme of thievery



Developer Student Clubs

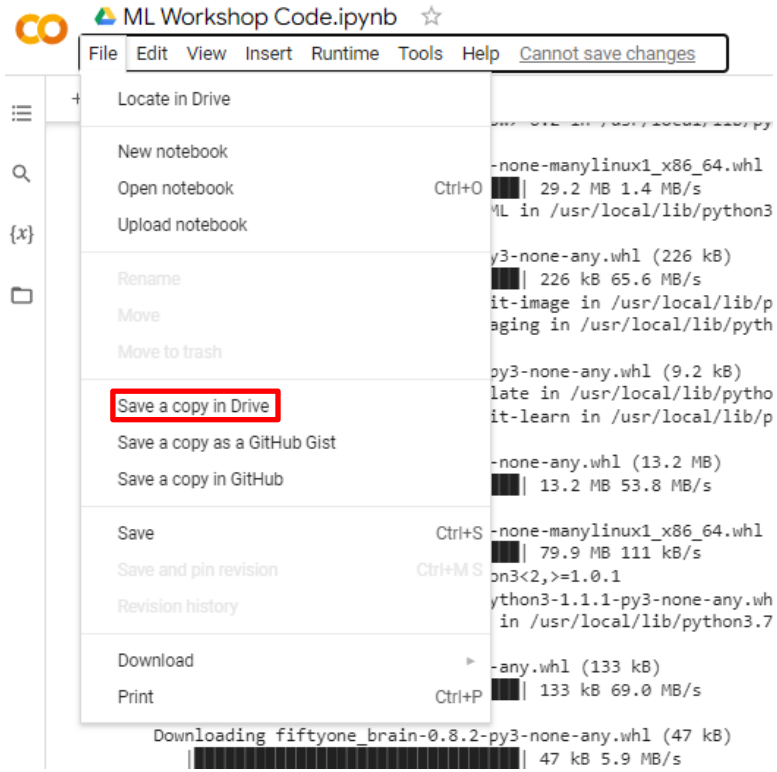
Disclaimer

The speaker may mention concepts and words such as “stealing” or “theft.”

Rest assured, what we are doing is 100% legal.

The Colab Notebook

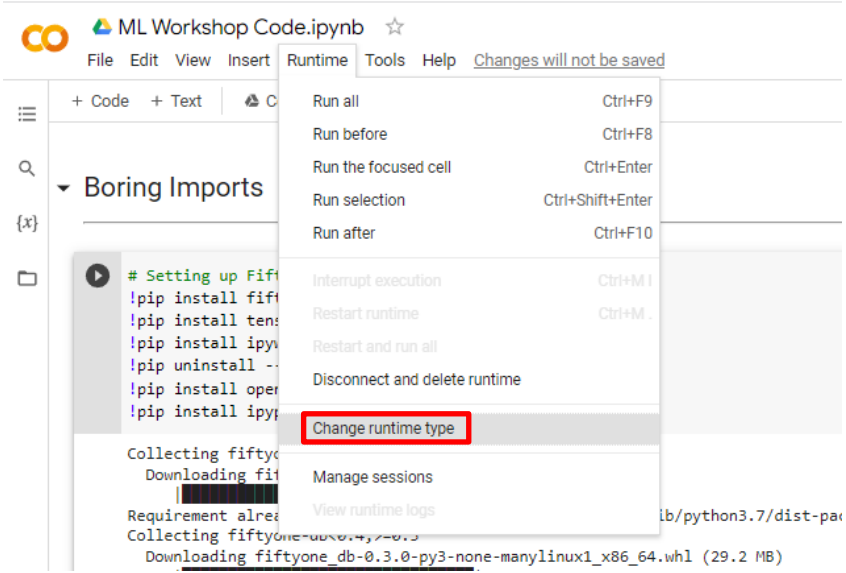
Step One: Stealing the notebook

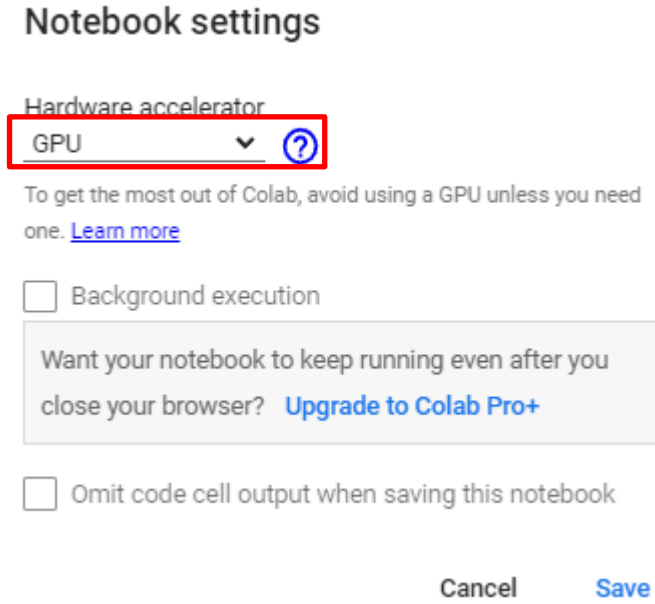


Developer Student Clubs

The Colab Notebook

Step Two: Stealing a GPU

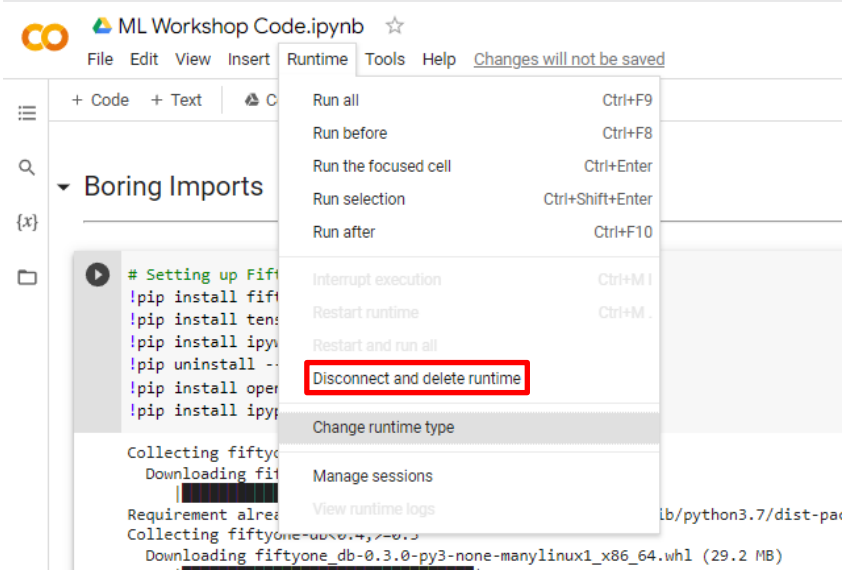




Developer Student Clubs

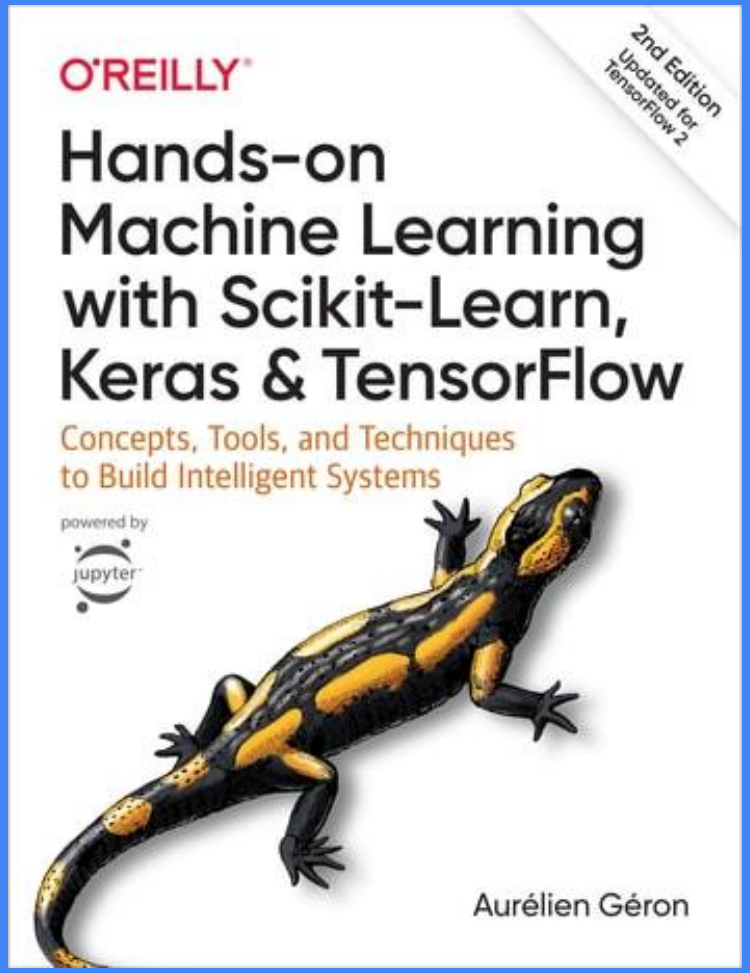
The Colab Notebook

Step Three: resetting the notebook



Developer Student Clubs

Generally Good Practical Resource



A Shameless Plug

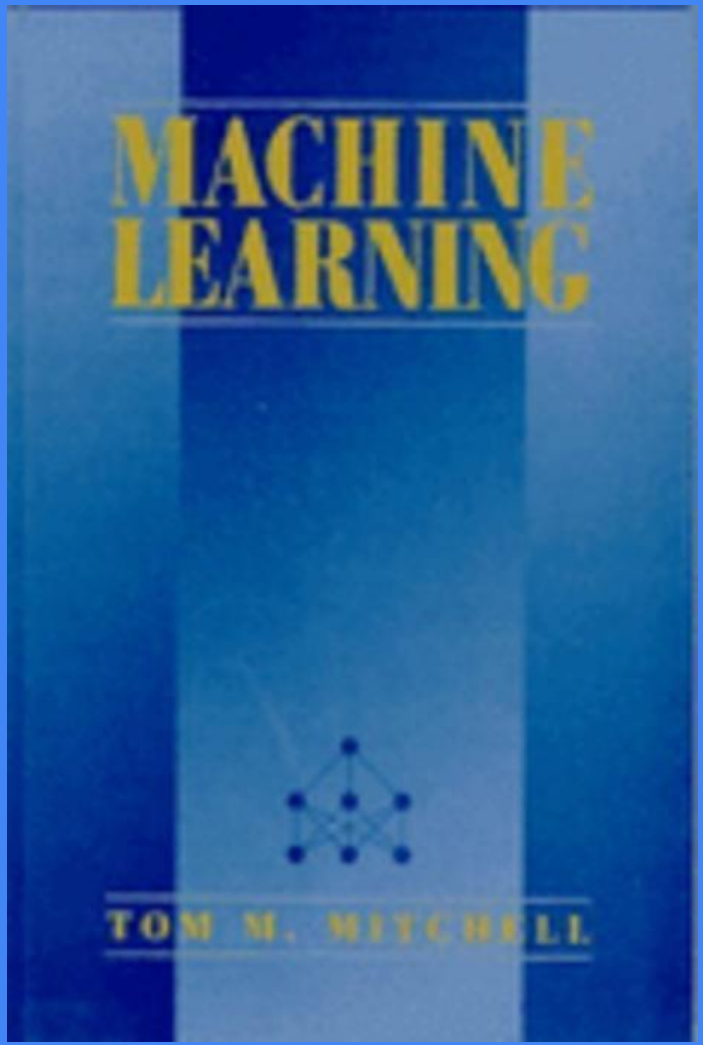
Aurélien Géron

Speaker Event

25th May

Developer Student Clubs

Generally Good Theoretical Resource



Developer Student Clubs

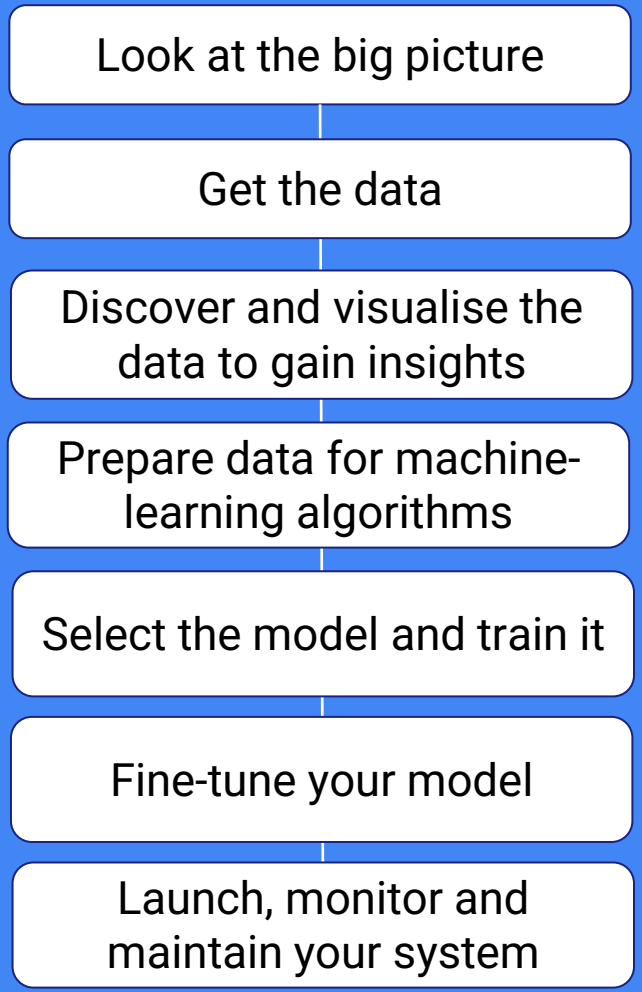
What this workshop will go through

Very practical focused

Impossible to teach the ins-and-outs of theory in a one hour workshop

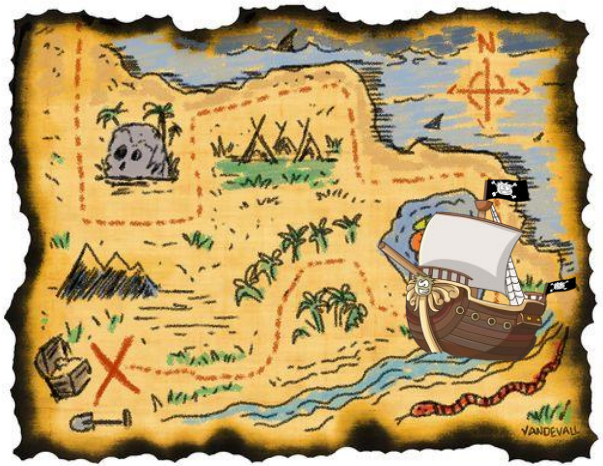
Further resources will be provided for your own discretion

Don’t worry about not understanding ALL the concepts or ALL the code!



What this workshop will cover:

The goal is to build something and have fun!



Developer Student Clubs

- Look at the big picture
- Get the data
- Discover and visualise the data to gain insights
- Prepare data for machine-learning algorithms
- Select the model and train it
- Fine-tune your model
- Launch, monitor and maintain your system

What this workshop will cover:

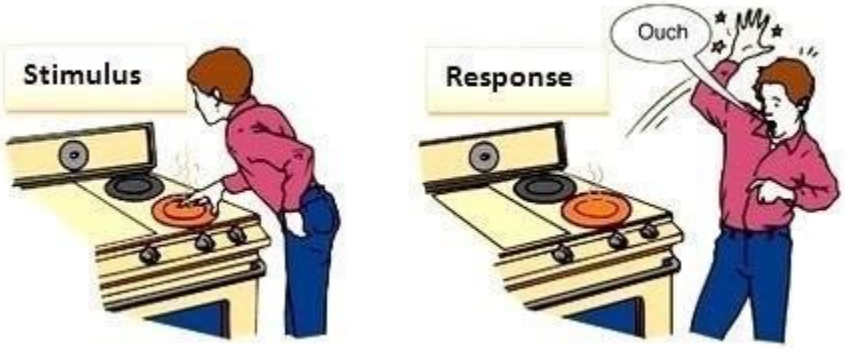
The goal is to build something and have fun!

What we intend to explore

Developer Student Clubs

- Look at the big picture
- Get the data
- Discover and visualise the data to gain insights
- Prepare data for machine-learning algorithms
- Select the model and train it
- Fine-tune your model
- Launch, monitor and maintain your system

What is machine learning?



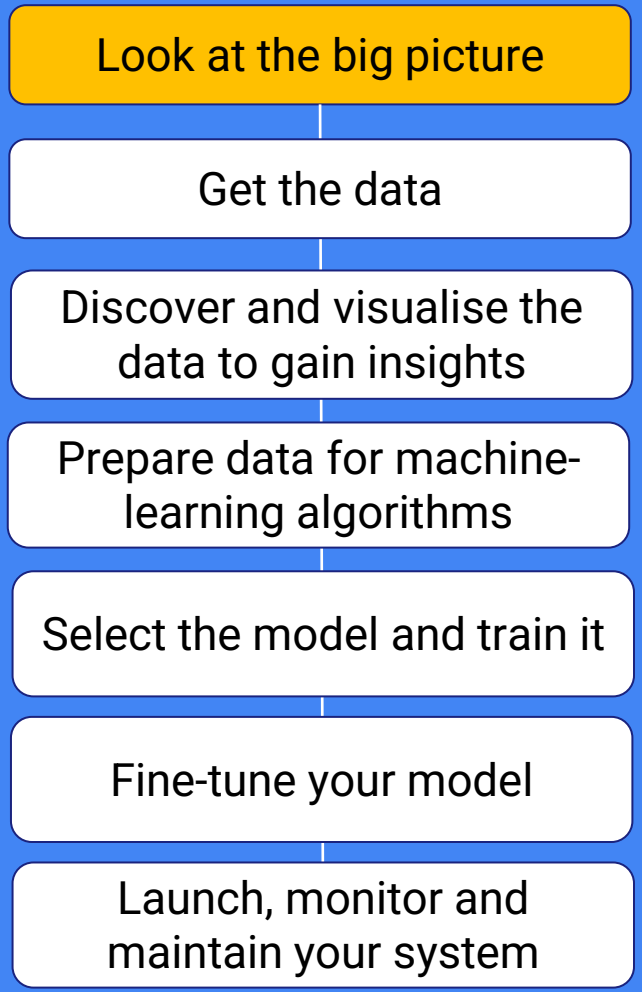
Developer Student Clubs

- Look at the big picture
- Get the data
- Discover and visualise the data to gain insights
- Prepare data for machine-learning algorithms
- Select the model and train it
- Fine-tune your model
- Launch, monitor and maintain your system

What is machine learning?

“A computer program is said to **learn** from experience *E* with respect to some class of tasks *T* and performance measure *P*, if its performance at tasks in *T*, as measured by *P*, improves with experience *E*.”

- Mitchell, T. M. (1997). Machine learning. New York: McGraw-Hill.

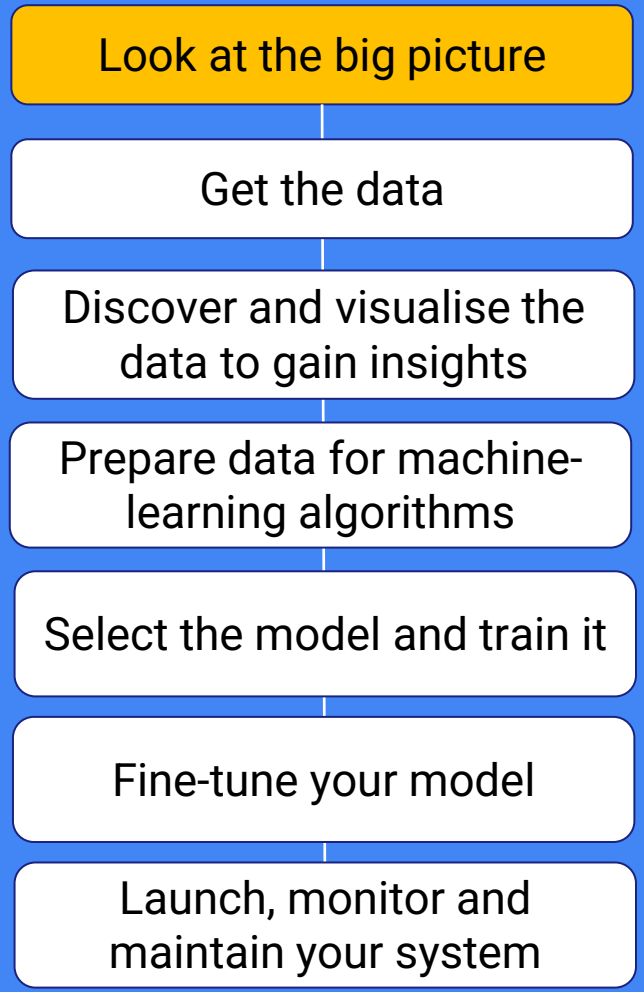


When to use machine learning?

Have a view of what you're trying to achieve.

Good places to start*:

- Can the rules be coded?
- Can the task be scaled?



Developer Student Clubs

*https://docs.aws.amazon.com/machine-learning/latest/dg/machinelearning-dg.pdf#when-to-use-machine-learning

When to use machine learning? An example:

Life insurance:
- The rules can be coded
- The problem can scale

Solution: NOT a machine-learning algorithm!

Developer Student Clubs

Look at the big picture

Get the data

Discover and visualise the data to gain insights

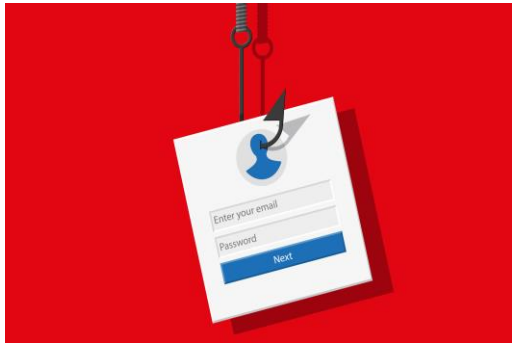
Prepare data for machine-learning algorithms

Select the model and train it

Fine-tune your model

Launch, monitor and maintain your system

When to use machine learning? An example:



Phishing emails
- The rules cannot be coded
- The problem can scale

Solution: A machine-learning algorithm!

Developer Student Clubs

Look at the big picture

Get the data

Discover and visualise the data to gain insights

Prepare data for machine-learning algorithms

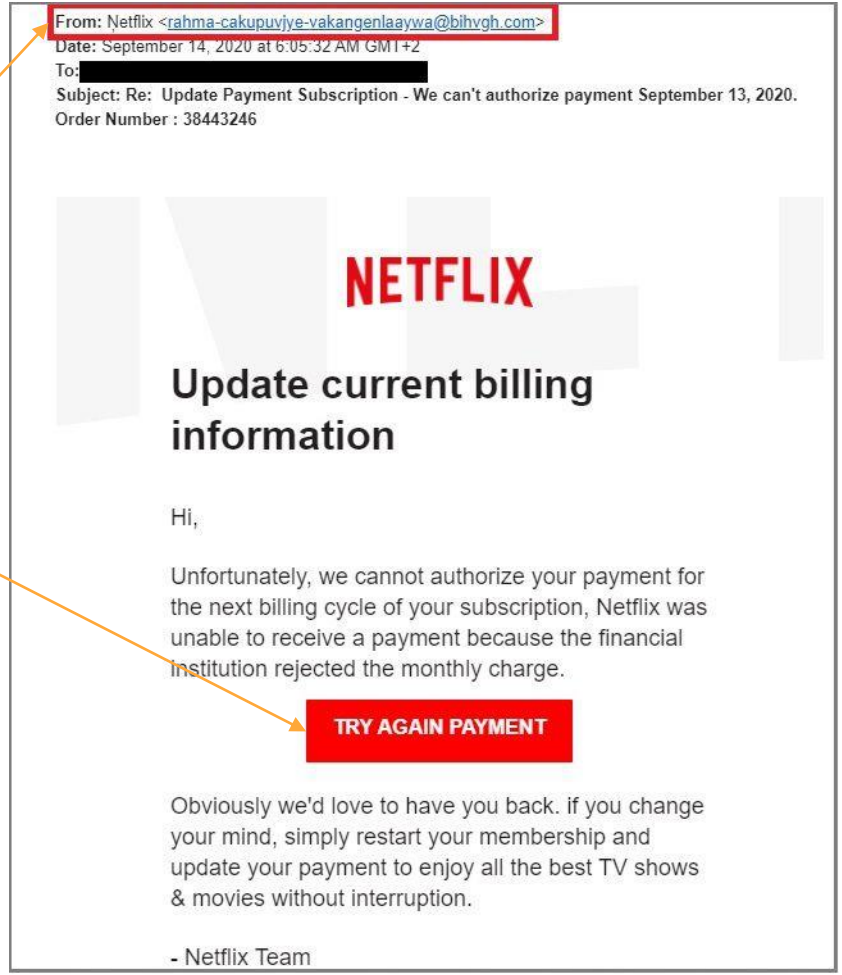
Select the model and train it

Fine-tune your model

Launch, monitor and maintain your system

The rules cannot be coded:

- Suspicious sender
- Poor English
- And many more!



Developer Student Clubs

Your task:

I want to find something from my room*

*Not my room

Quick and dirty checklist:

- The rules on searching for an item cannot be coded
- The problem can scale

Solution: A machine-learning algorithm!

Developer Student Clubs

Look at the big picture

Get the data

Discover and visualise the data to gain insights

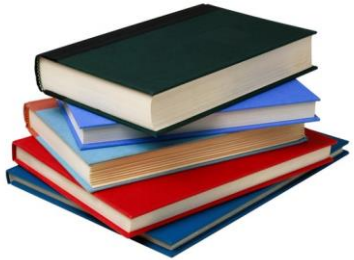
Prepare data for machine-learning algorithms

Select the model and train it

Fine-tune your model

Launch, monitor and maintain your system

What I’m trying to find



Get the data:

Highly dependent on the big picture identified in the previous step!

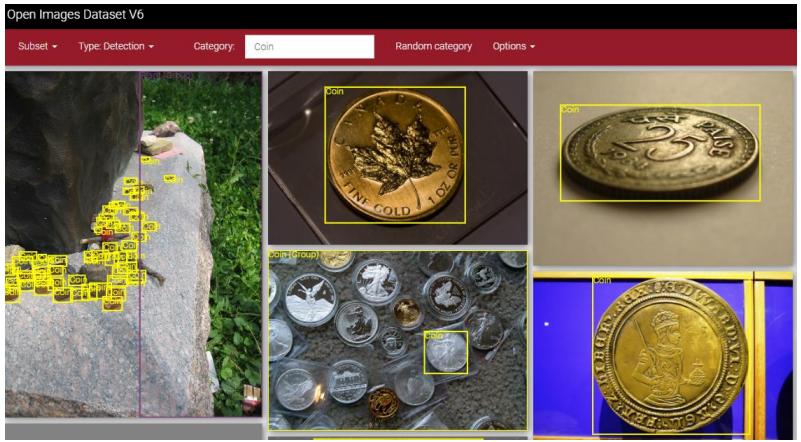
Let's say I want to find a coin

Developer Student Clubs

1. Look at the big picture
2. **Get the data**
3. Discover and visualise the data to gain insights
4. Prepare data for machine-learning algorithms
5. Select the model and train it
6. Fine-tune your model
7. Launch, monitor and maintain your system

Get the data for the task:

We will be using Google Open Images as our dataset



Look at the big picture

Get the data

Discover and visualise the data to gain insights

Prepare data for machine-learning algorithms

Select the model and train it

Fine-tune your model

Launch, monitor and maintain your system

FiftyOne

We will be using FiftyOne to handle importing the dataset

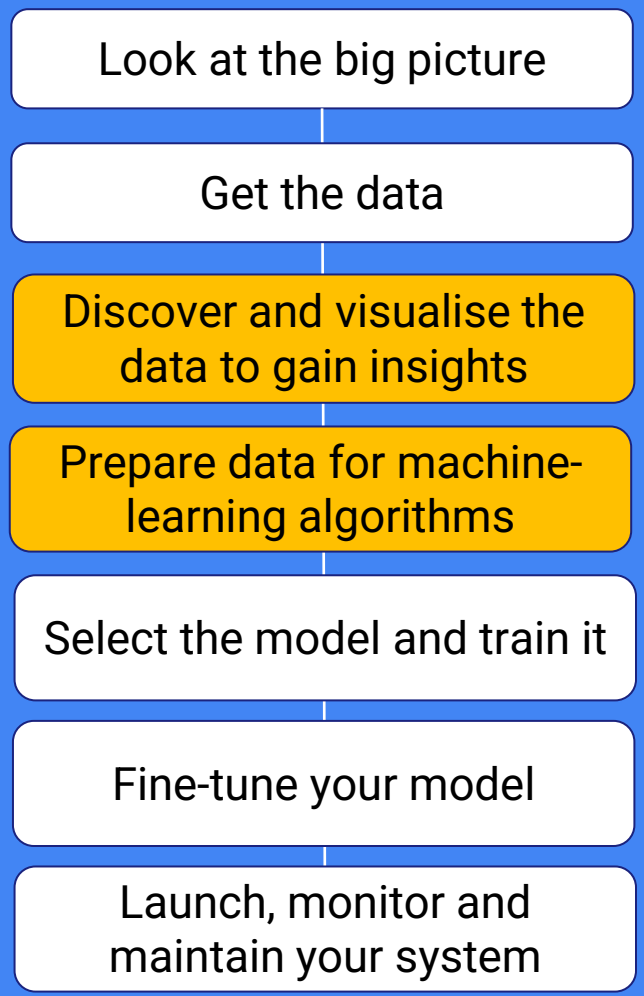
Developer Student Clubs

- Look at the big picture
- **Get the data**
- Discover and visualise the data to gain insights
- Prepare data for machine-learning algorithms
- Select the model and train it
- Fine-tune your model
- Launch, monitor and maintain your system

Import FiftyOne:

```
[ ]  import fiftyone as fo
     import fiftyone.zoo as foz
     import random

     desired_class = "Coin"

     dataset = foz.load_zoo_dataset(
         "open-images-v6",
         label_types=["detections"],
         classes=[desired_class.capitalize()],
         max_samples=250,
         shuffle=True,
         seed = 5,
         num_workers = 20,
     )
```

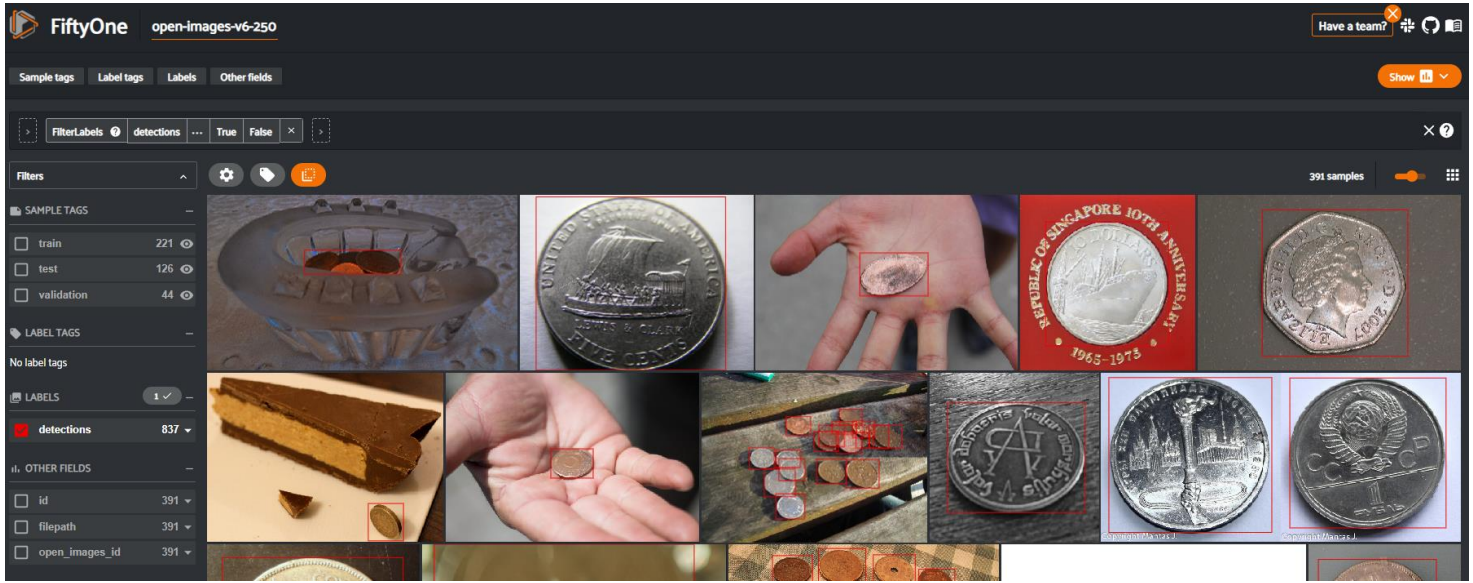
Developer Student Clubs



Viewing our desired_class

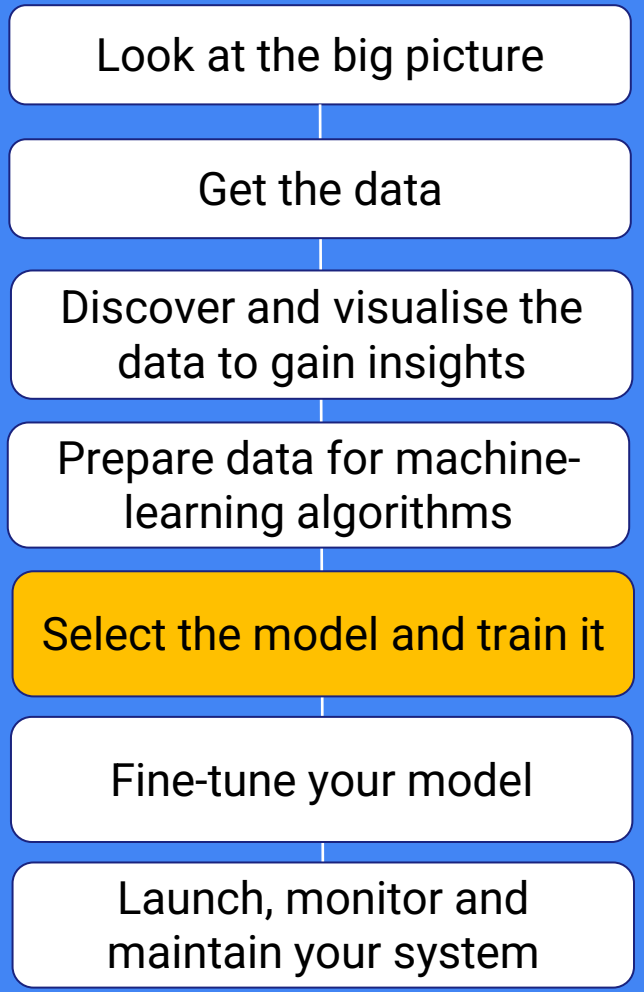
```
from fiftyone import ViewField as F

data_view = dataset.filter_labels("detections", (F("label") == desired_class) & (F("IsGroupOf") == False))

# Visualize the dataset in the FiftyOne App
session = fo.launch_app(view=data_view)
```

Viewing our desired_class

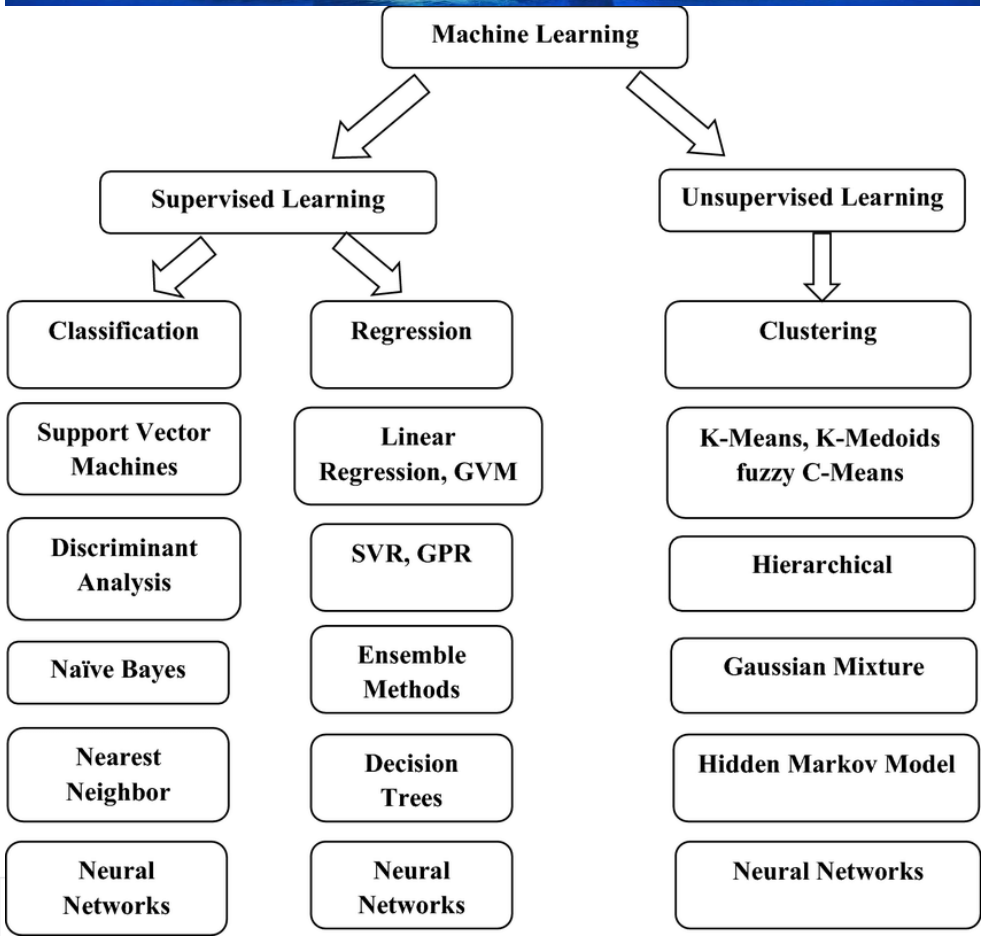


Developer Student Clubs

Determining our model:



Determining which algorithm to use:



Google Search

- Understand for what types of data the algorithm should be used for
- Understand when to use said algorithm
- Worry about the “what” rather than the “how”.

Developer Student Clubs

Supervised Learning

An overly simple analogy

COIN

COIN

COIN

NO COIN

X A very large number

Developer Student Clubs

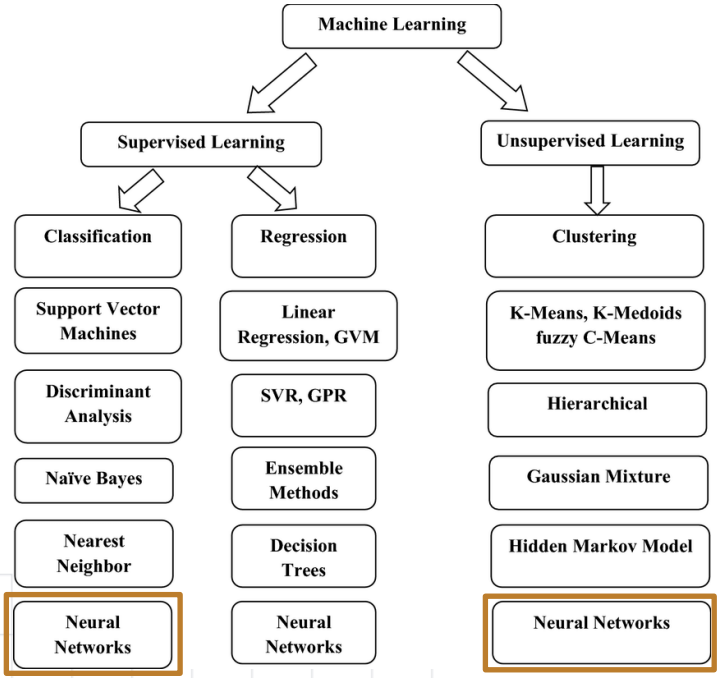
Unsupervised Learning

An overly simple analogy

X A very large number

Developer Student Clubs

Disclaimer



Developer Student Clubs

Look at the big picture

Get the data

Discover and visualise the data to gain insights

Prepare data for machine-learning algorithms

Select the model and train it

Fine-tune your model

Launch, monitor and maintain your system

Determining which algorithm to use:

A lot of resources to gather data, train and monitor an algorithm

The solution is thievery! (Stealing a pre-trained algorithm)

Developer Student Clubs

Look at the big picture

Get the data

Discover and visualise the data to gain insights

Prepare data for machine-learning algorithms

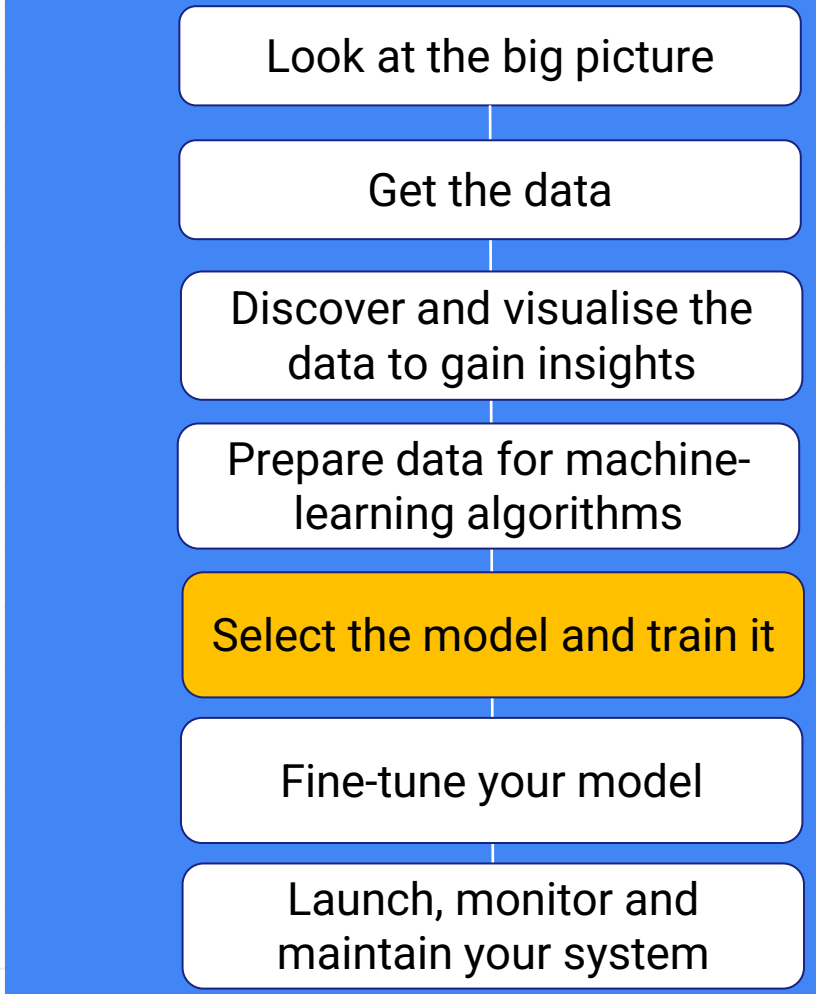
Select the model and train it

Fine-tune your model

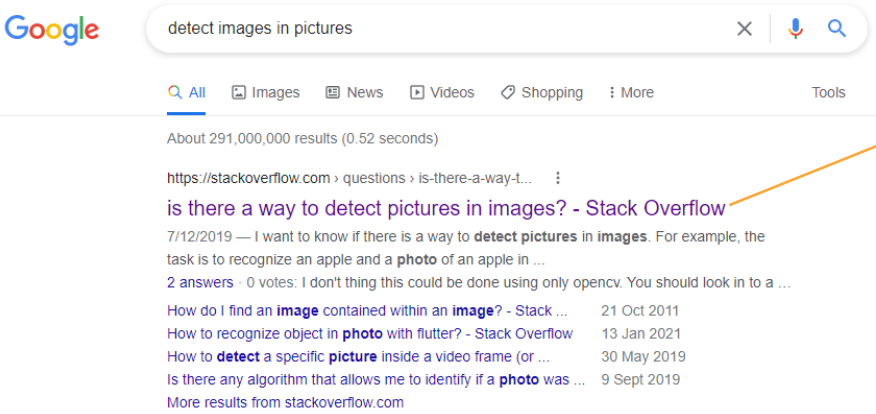
Launch, monitor and maintain your system

Disclaimer

Content from here on out will be focused on **NEURAL NETWORKS** and will be specific to **YOLOV5**



Determining which algorithm to use:



is there a way to detect pictures in images?

Asked 2 years, 4 months ago Modified 4 months ago Viewed 245 times

-2

I want to know if there is a way to detect pictures in images. For example, the task is to recognize an apple and a photo of an apple in given image. Is it possible with OpenCV or any other way?

opencv | image-processing | computer-vision | computer-science | object-recognition

Share Follow

asked Dec 7, 2019 at 4:04
Kundan
11

are you searching for some specific pictures or generally for any pictures? What kind of images does your application expect to see? – Micka Dec 8, 2019 at 22:30

Add a comment

2 Answers

Sorted by: Highest score (default)

0

I don't thing this could be done using only opencv. You should look in to a deep learning approach. There are many models trained for object detection. eg:- YOLO model

Share Follow

answered Dec 8, 2019 at 17:40
chamith mawela
36 ● 7

Developer Student Clubs

Our training algorithm

- YoloV5 is a supervised, neural network algorithm
- The algorithm is used to detect things in images
- The algorithm requires images and labels for inputs

Developer Student Clubs

Look at the big picture

Get the data

Discover and visualise the data to gain insights

Prepare data for machine-learning algorithms

Select the model and train it

Fine-tune your model

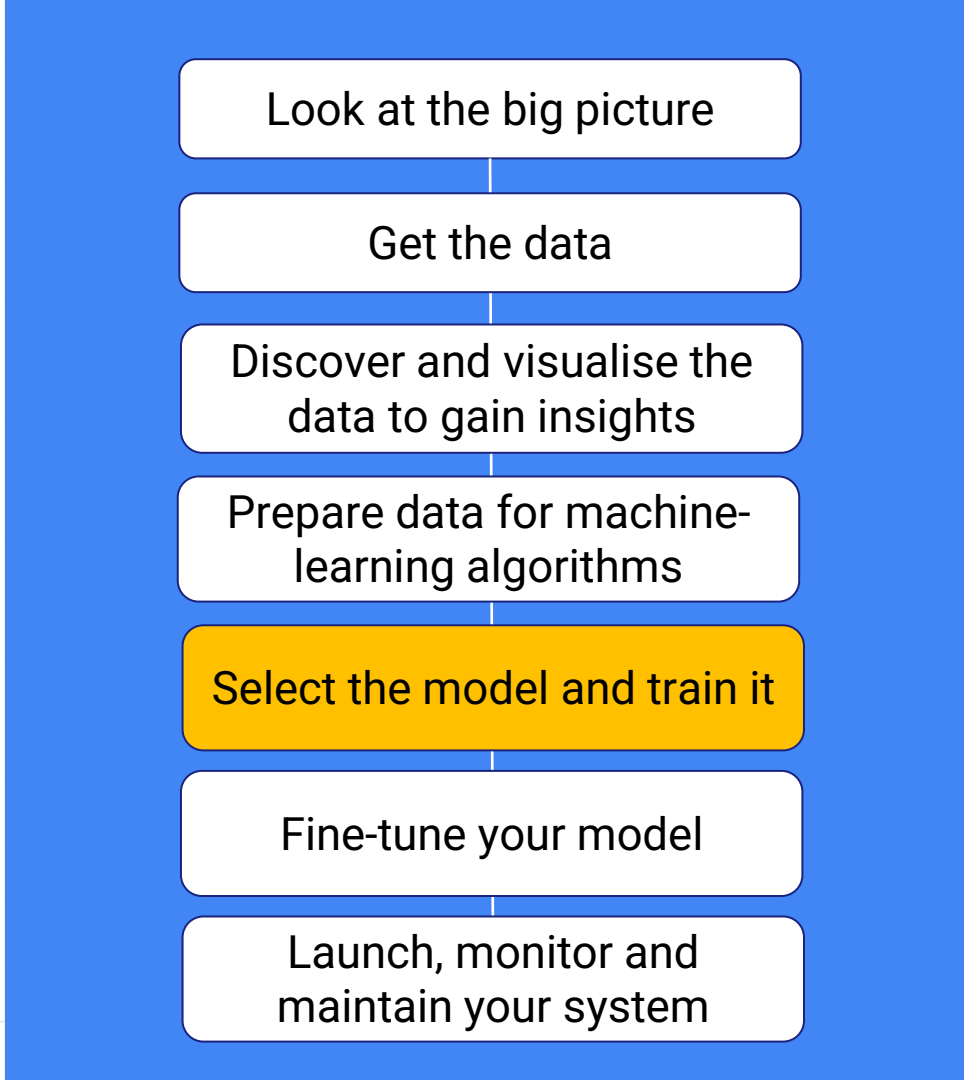
Launch, monitor and maintain your system

Installing YoloV5

Installing YoloV5

```
[4]  # Clone YOLO-V5, change to its directory and install dependencies
     !git clone https://github.com/ultralytics/yolov5
     %cd yolov5
     !pip install -r requirements.txt
```

Developer Student Clubs



Configuring the yaml file

```
[5]  # Constructing the yaml file
     import yaml

     yaml_file = dict(
         path = "../dataset/FiftyOne",
         train = "/content/dataset/train/images",
         val = "/content/dataset/validate/images",
         test = "/content/dataset/test/images",
         nc = 1,
         names = [desired_class],
                     )

     with open(r'/content/yolov5/data/FiftyOne.yaml', 'w') as file:
         documents = yaml.dump(yaml_file, file)
         print(yaml_file)
```

Developer Student Clubs

- Look at the big picture
- Get the data
- Discover and visualise the data to gain insights
- Prepare data for machine-learning algorithms
- **Select the model and train it**
- Fine-tune your model
- Launch, monitor and maintain your system

Creating the dataset directories

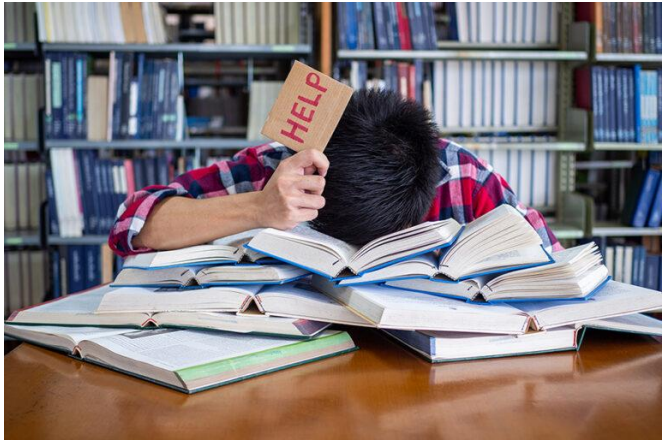
```
# Create dataset directories
!mkdir /content/dataset
!mkdir /content/dataset/train /content/dataset/train/images /content/dataset/train/labels
!mkdir /content/dataset/validate /content/dataset/validate/images /content/dataset/validate/labels
!mkdir /content/dataset/test /content/dataset/test/images /content/dataset/test/labels
```

Developer Student Clubs

- Look at the big picture
- Get the data
- Discover and visualise the data to gain insights
- Prepare data for machine-learning algorithms
- **Select the model and train it**
- Fine-tune your model
- Launch, monitor and maintain your system

The train, validate and test set

Train set:



Look at the big picture

Get the data

Discover and visualise the data to gain insights

Prepare data for machine-learning algorithms

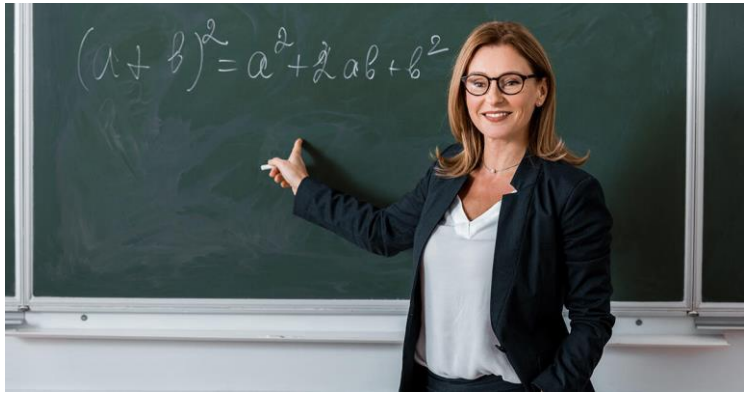
Select the model and train it

Fine-tune your model

Launch, monitor and maintain your system

The train, validate and test set

Validate set:



Developer Student Clubs

- Look at the big picture
- Get the data
- Discover and visualise the data to gain insights
- Prepare data for machine-learning algorithms
- **Select the model and train it**
- Fine-tune your model
- Launch, monitor and maintain your system

The train, validate and test set

Test set:

Developer Student Clubs

- Look at the big picture
- Get the data
- Discover and visualise the data to gain insights
- Prepare data for machine-learning algorithms
- **Select the model and train it**
- Fine-tune your model
- Launch, monitor and maintain your system

Export the dataset into directories

```
[7] # Export dataset into select directories
    import os
    import shutil

    export_dir = "/content"
    dataset_type = fo.types.YOLOv5Dataset
    label_field = "detections"

    data_view.export(
        export_dir=export_dir,
        dataset_type=dataset_type,
        label_field=label_field,
    )

    for filename in os.listdir("/content/images/val"):
      random_value = random.randrange(10)
      if random_value == 0:
        filename = filename[:filename.index(".")]
        shutil.move("/content/images/val/{}.jpg".format(filename), "/content/dataset/test/images/{}.jpg".format(filename))
        shutil.move("/content/labels/val/{}.txt".format(filename), "/content/dataset/test/labels/{}.txt".format(filename))
      elif random_value == 1:
        filename = filename[:filename.index(".")]
        shutil.move("/content/images/val/{}.jpg".format(filename), "/content/dataset/validate/images/{}.jpg".format(filename))
        shutil.move("/content/labels/val/{}.txt".format(filename), "/content/dataset/validate/labels/{}.txt".format(filename))
      else:
        filename = filename[:filename.index(".")]
        shutil.move("/content/images/val/{}.jpg".format(filename), "/content/dataset/train/images/{}.jpg".format(filename))
        shutil.move("/content/labels/val/{}.txt".format(filename), "/content/dataset/train/labels/{}.txt".format(filename))
```

Developer Student Clubs

- Look at the big picture
- Get the data
- Discover and visualise the data to gain insights
- Prepare data for machine-learning algorithms
- **Select the model and train it**
- Fine-tune your model
- Launch, monitor and maintain your system

Training the model

```
[8]  # Train the model
     !python train.py --img 640 --batch 32 --epochs 10 --data /content/yolov5/data/FiftyOne.yaml --weights yolov5n.pt --single-cls
```

A batch

Our dataset

Developer Student Clubs

Training the model

```
[8]  # Train the model
     !python train.py --img 640 --batch 32 --epochs 10 --data /content/yolov5/data/FiftyOne.yaml --weights yolov5n.pt --single-cls
```

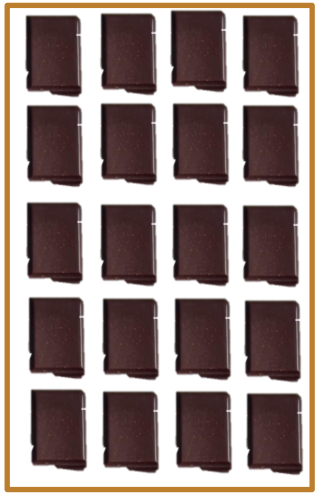
Developer Student Clubs

Training the model

```
[8]  # Train the model
     !python train.py --img 640 --batch 32 --epochs 10 --data /content/yolov5/data/FiftyOne.yaml --weights yolov5n.pt --single-cls
```

Training the model

```
[8]  # Train the model
     !python train.py --img 640 --batch 32 --epochs 10 --data /content/yolov5/data/FiftyOne.yaml --weights yolov5n.pt --single-cls
```



X 10

Developer Student Clubs

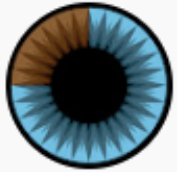
Utilise YoloV5’s algorithm with pre-trained weights

```
                 from  n    params  module                                  arguments
  0                -1  1      1760  models.common.Conv                      [3, 16, 6, 2, 2]
  1                -1  1      4672  models.common.Conv                      [16, 32, 3, 2]
  2                -1  1      4800  models.common.C3                        [32, 32, 1]
  3                -1  1     18560  models.common.Conv                      [32, 64, 3, 2]
  4                -1  2     29184  models.common.C3                        [64, 64, 2]
  5                -1  1     73984  models.common.Conv                      [64, 128, 3, 2]
  6                -1  3    156928  models.common.C3                        [128, 128, 3]
  7                -1  1    295424  models.common.Conv                      [128, 256, 3, 2]
  8                -1  1    296448  models.common.C3                        [256, 256, 1]
  9                -1  1    164608  models.common.SPPF                      [256, 256, 5]
 10                -1  1     33024  models.common.Conv                      [256, 128, 1, 1]
 11                -1  1         0  torch.nn.modules.upsampling.Upsample    [None, 2, 'nearest']
 12          [-1, 6]  1         0  models.common.Concat                    [1]
 13                -1  1     90880  models.common.C3                        [256, 128, 1, False]
 14                -1  1      8320  models.common.Conv                      [128, 64, 1, 1]
 15                -1  1         0  torch.nn.modules.upsampling.Upsample    [None, 2, 'nearest']
 16          [-1, 4]  1         0  models.common.Concat                    [1]
 17                -1  1     22912  models.common.C3                        [128, 64, 1, False]
 18                -1  1     36992  models.common.Conv                      [64, 64, 3, 2]
 19         [-1, 14]  1         0  models.common.Concat                    [1]
 20                -1  1     74496  models.common.C3                        [128, 128, 1, False]
 21                -1  1    147712  models.common.Conv                      [128, 128, 3, 2]
 22         [-1, 10]  1         0  models.common.Concat                    [1]
 23                -1  1    296448  models.common.C3                        [256, 256, 1, False]
 24      [17, 20, 23]  1      8118  models.yolo.Detect                      [1, [[10, 13, 16, 30, 33, 23], [30, 61, 62, 45, 59, 119], [116, 90, 156, 198, 373, 326]], [64, 128, 256]]
Model summary: 270 layers, 1765270 parameters, 1765270 gradients, 4.2 GFLOPs
```

Developer Student Clubs

Additional Resources

tds



3Blue1Brown

4.49M subscribers

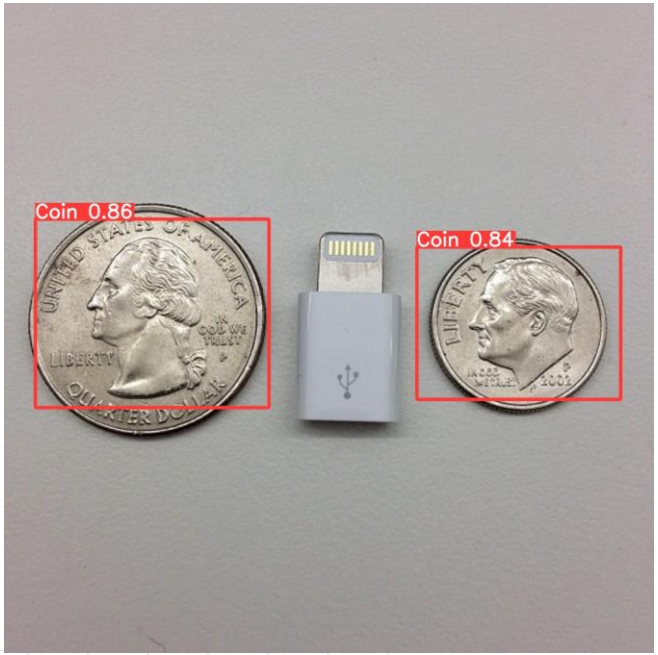
THE UNIVERSITY OF AUCKLAND

Te Whare Wānanga o Tāmaki Makaurau

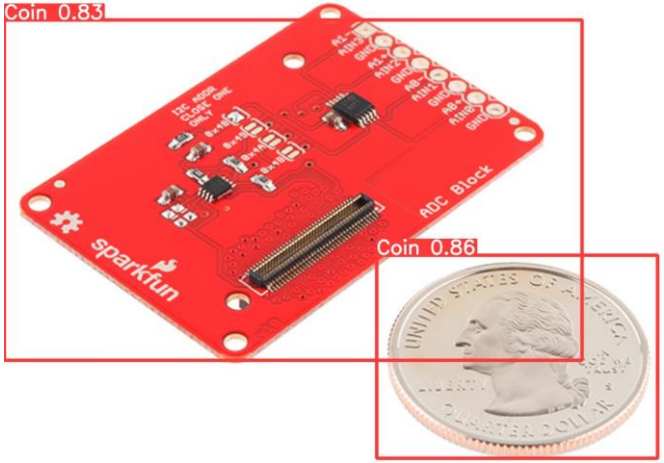
NEW ZEALAND

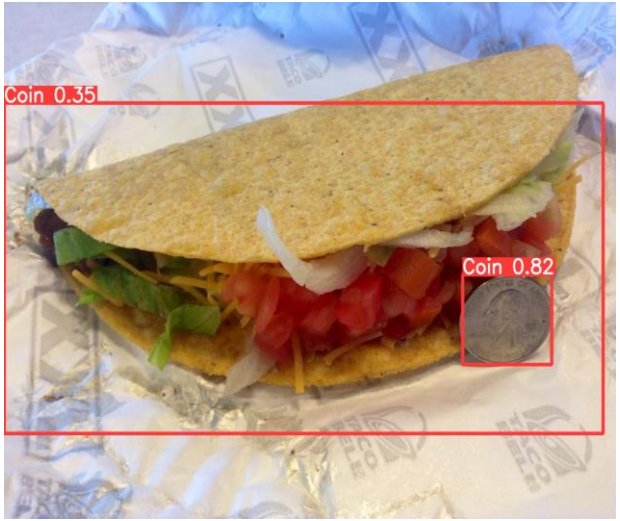
Developer Student Clubs

Detecting coins!



Whoops!

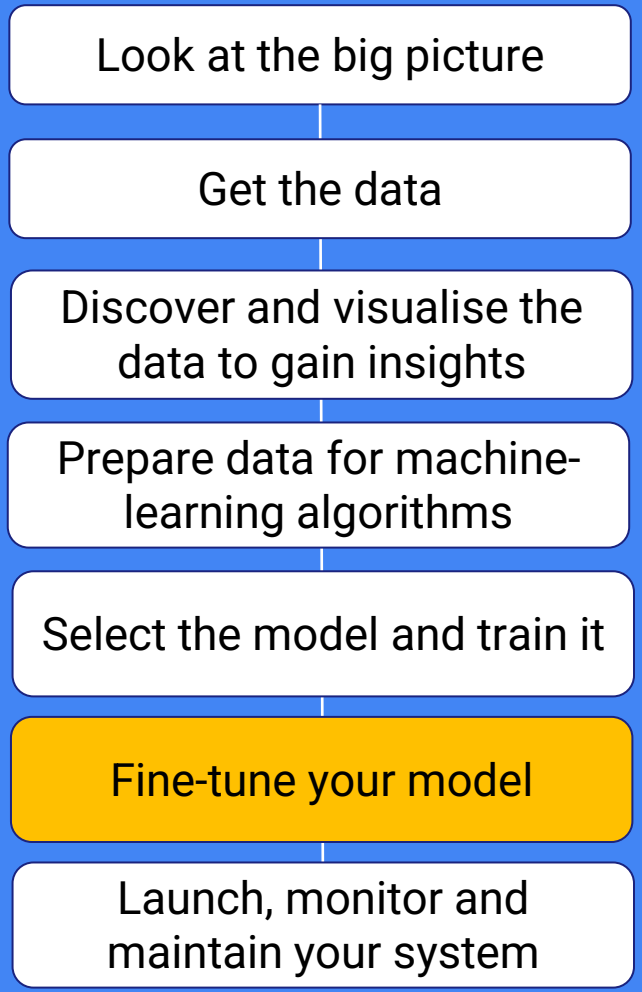




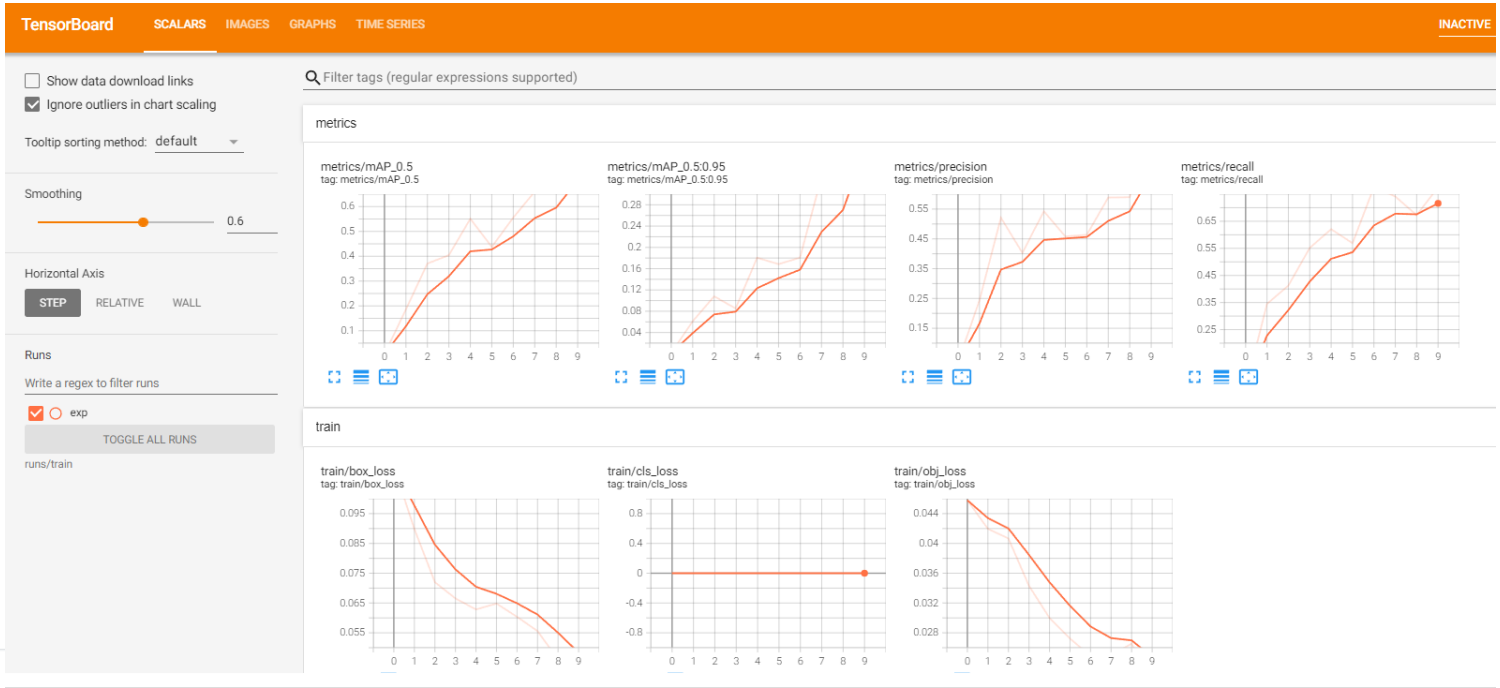
Developer Student Clubs

Fine-tune your model

```
[ ]  # View tensorboard
     %load_ext tensorboard
     %tensorboard --logdir runs/train
```

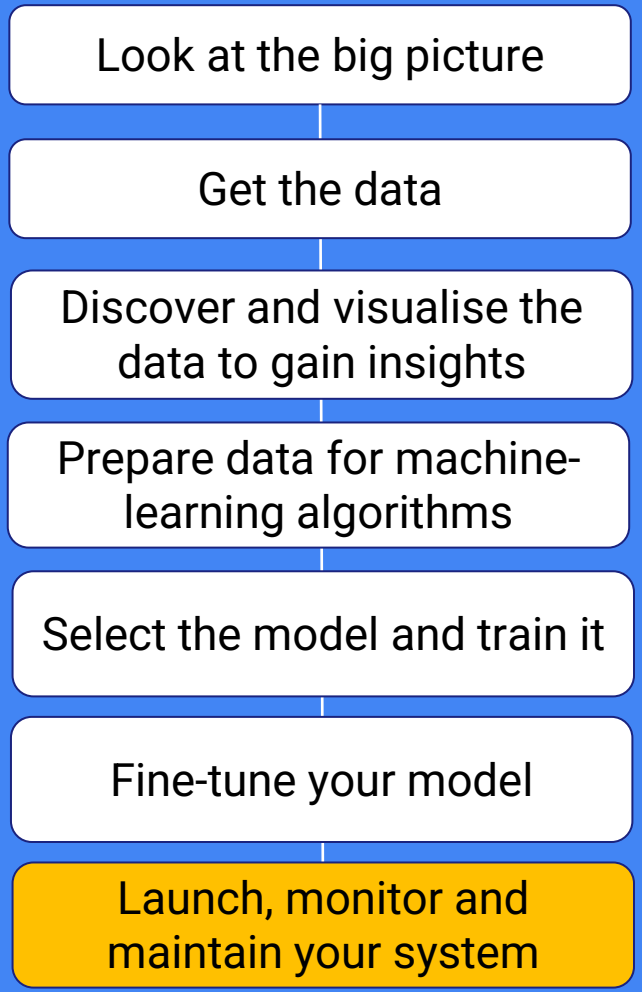


Fine-tune your model



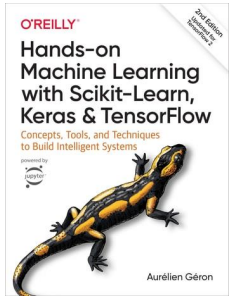
Launch, monitor and maintain your system

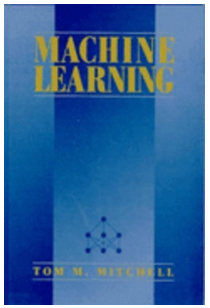
Shifting more towards software design



Where next?

THEORY





tds

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND

3Blue1Brown
4.49M subscribers

PRACTICE

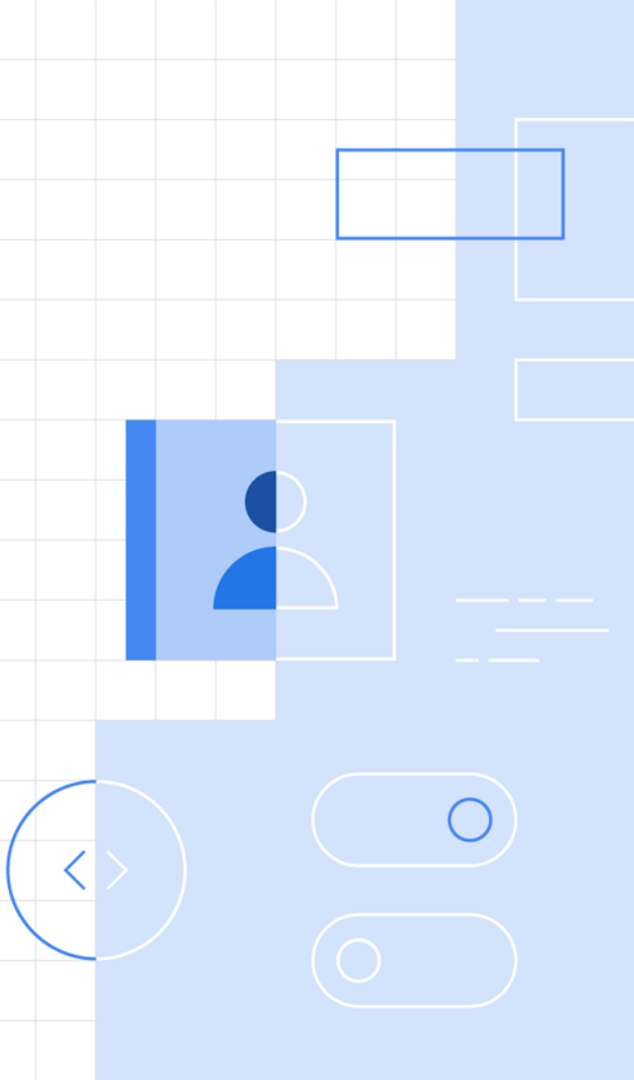
+ kaggle +

EXPLORATION

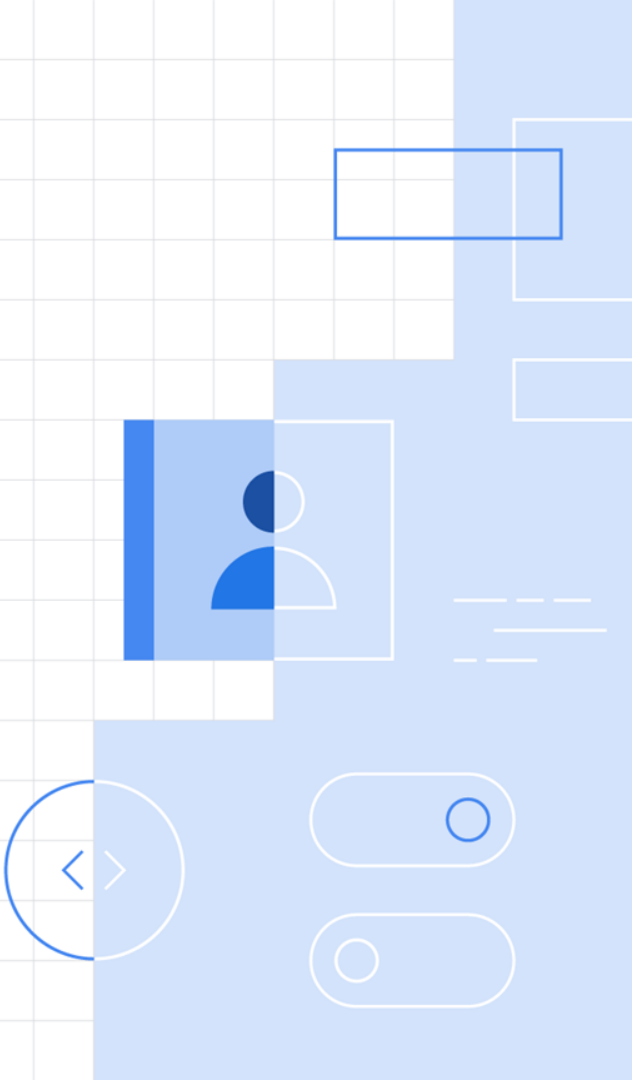
Google StackExchange

Developer Student Clubs

Breaktime!



Thank you!



Google Developers