Platforms GPU comparison fast.ai v3

2019/02/02 Marcello Morchio

marcello@datascienceseed.com

gradient kaggle colab Windows 10

Contents

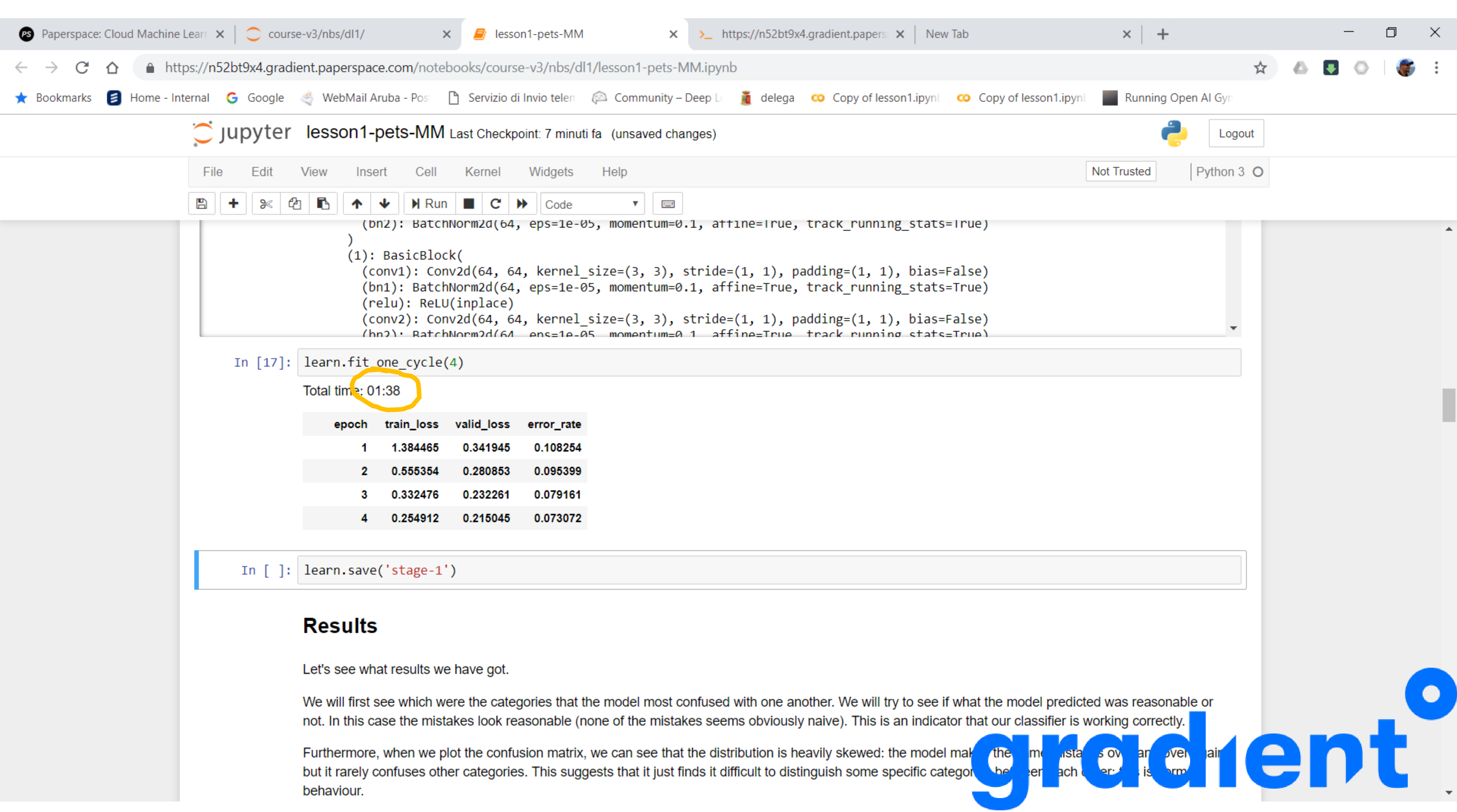
Notes taken while selecting theplatform for the Genova Fast.ai Learning Group

- Performance comparison
- Utilities
- Importing your dataset
- Issues

Performances summary (lesson1 code)

Platform	Video time 00:44:00
	Resnet34
	
	learn.fit_one_cycle(4)
		Video time 01:16:00
		Resnet34
		
		learn.unfreeze()
		learn.fit_one_cylce(1)
			Video time 01:25:00
			Resnet34
			
			learn.unfreeze()
			learn.fit_one_cycle(2, max_lr=slice(1e-6,1e-4))
				Video time 01:29:00
				Resten50
				
				learn.fit_one_cycle(8)

Fast.ai video	00:01:56	00:00:29	00:00:58	00:03:41 *(5 epochs, not 8)*
Gradient P5000	00:01:38	00:00:28	00:00:56	00:09:51
Kaggle	00:10:36	00:02:57	00:05:58	00:46:43
Colab	00:07:43	00:02:00	00:04:01	00:26:48
Intel CPU	01:11:26 **one epoch**			



```
          (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        )
        (1): BasicBlock(
          (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
          (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (relu): ReLU(inplace)
          (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
          (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
```

In [17]:

```
learn.fit_one_cycle(4)
```

```
Total time: 01:38
```

epoch	train_loss	valid_loss	error_rate
1	1.384465	0.341945	0.108254
2	0.555354	0.280853	0.095399
3	0.332476	0.232261	0.079161
4	0.254912	0.215045	0.073072

In [ ]:

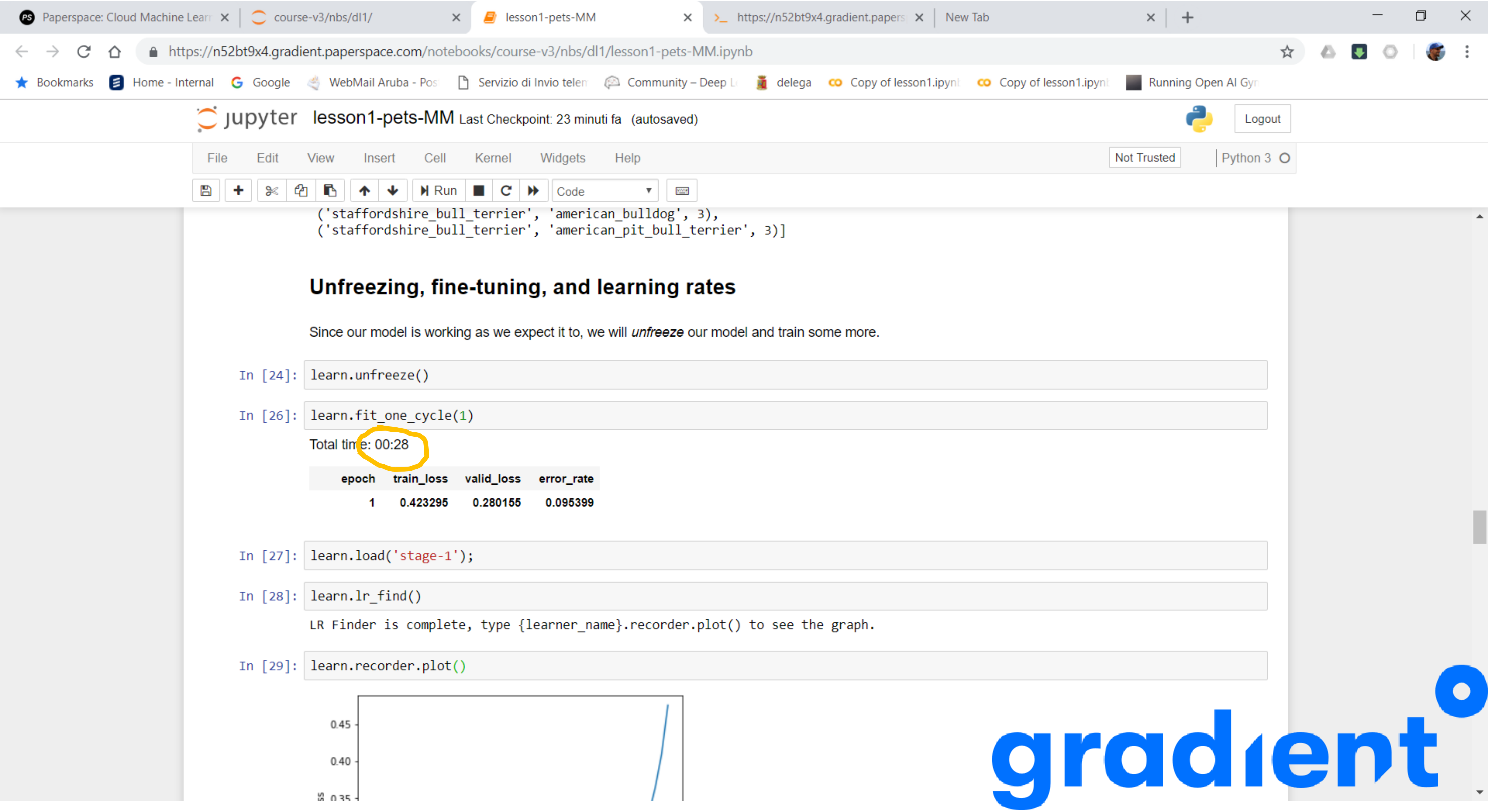
```
learn.save('stage-1')
```

Results

Let's see what results we have got.

We will first see which were the categories that the model most confused with one another. We will try to see if what the model predicted was reasonable or not. In this case the mistakes look reasonable (none of the mistakes seems obviously naive). This is an indicator that our classifier is working correctly.

Furthermore, when we plot the confusion matrix, we can see that the distribution is heavily skewed: the model mak[illegible] the [illegible] mistakes over and over again but it rarely confuses other categories. This suggests that it just finds it difficult to distinguish some specific categor[illegible] between each other: this is normal behaviour.



```
 ('staffordshire_bull_terrier', 'american_bulldog', 3),
 ('staffordshire_bull_terrier', 'american_pit_bull_terrier', 3)]
```

Unfreezing, fine-tuning, and learning rates

Since our model is working as we expect it to, we will *unfreeze* our model and train some more.

```
In [24]: learn.unfreeze()
```

```
In [26]: learn.fit_one_cycle(1)
```

Total time: 00:28

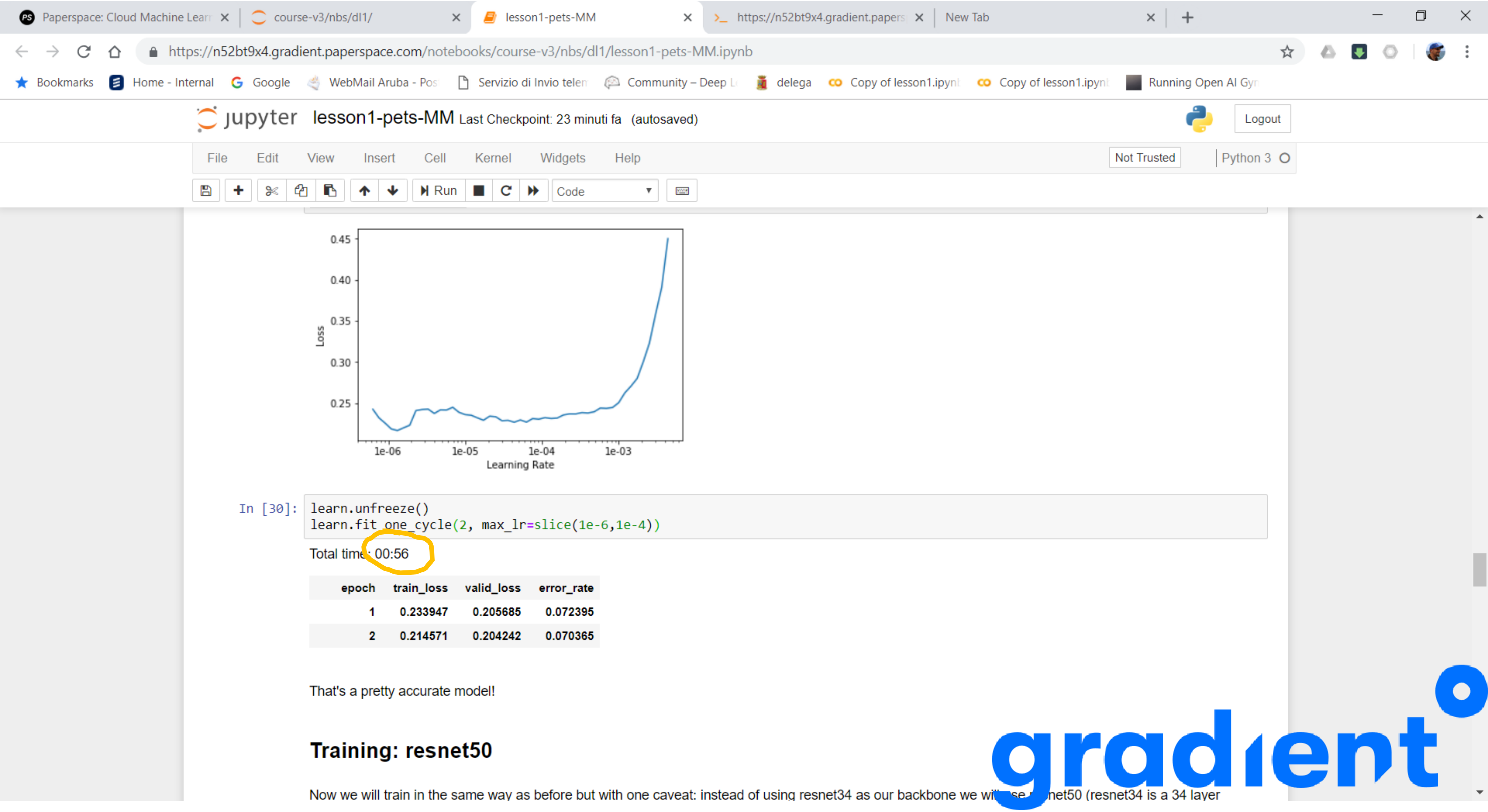
epoch	train_loss	valid_loss	error_rate
1	0.423295	0.280155	0.095399

```
In [27]: learn.load('stage-1');
```

```
In [28]: learn.lr_find()
```

```
LR Finder is complete, type {learner_name}.recorder.plot() to see the graph.
```

```
In [29]: learn.recorder.plot()
```



In [30]:

```
learn.unfreeze()
learn.fit_one_cycle(2, max_lr=slice(1e-6,1e-4))
```

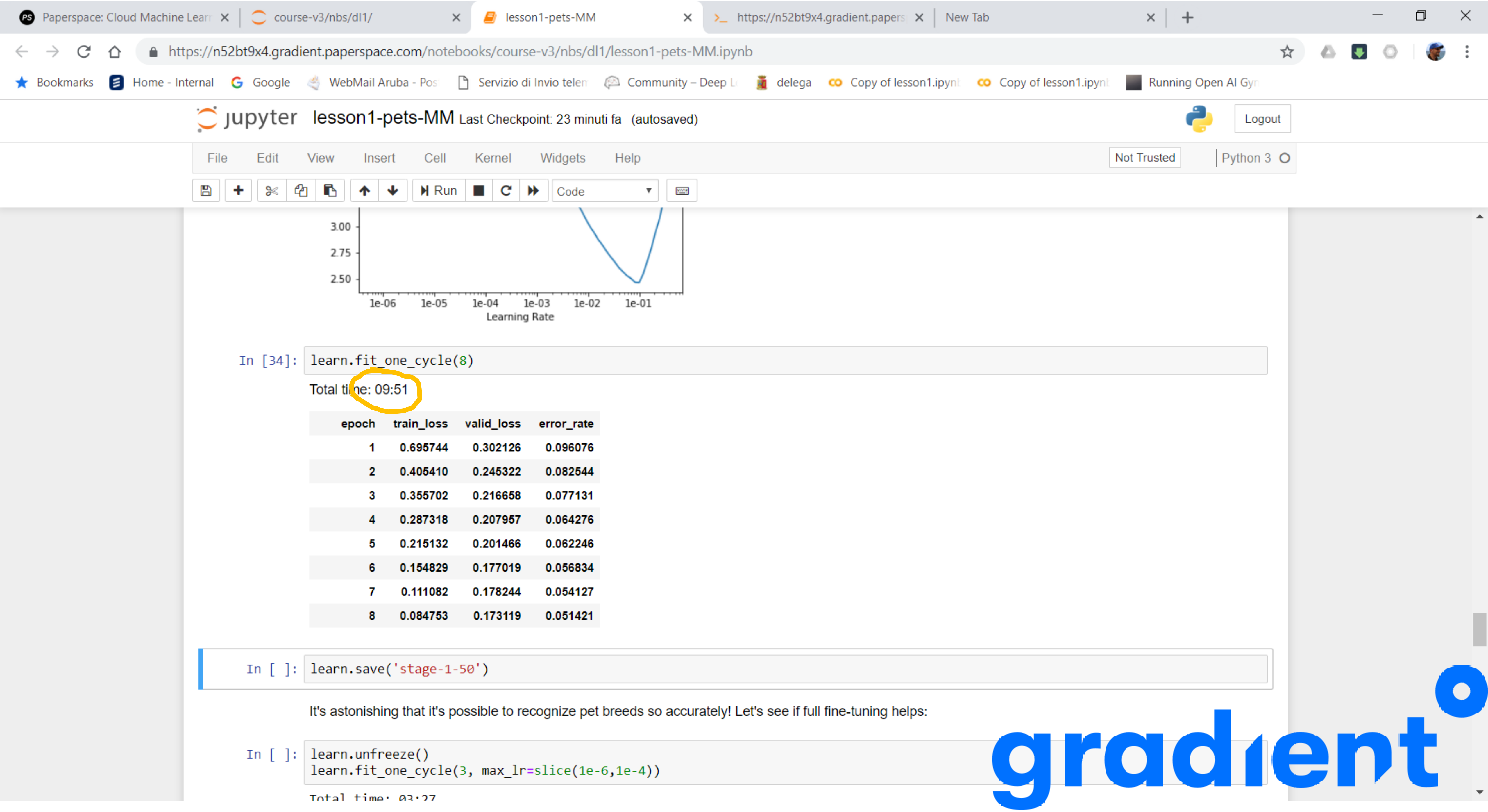
Total time: 00:56

epoch	train_loss	valid_loss	error_rate
1	0.233947	0.205685	0.072395
2	0.214571	0.204242	0.070365

That's a pretty accurate model!

Training: resnet50

Now we will train in the same way as before but with one caveat: instead of using resnet34 as our backbone we will use resnet50 (resnet34 is a 34 layer



In [34]:
```
learn.fit_one_cycle(8)
```

Total time: 09:51

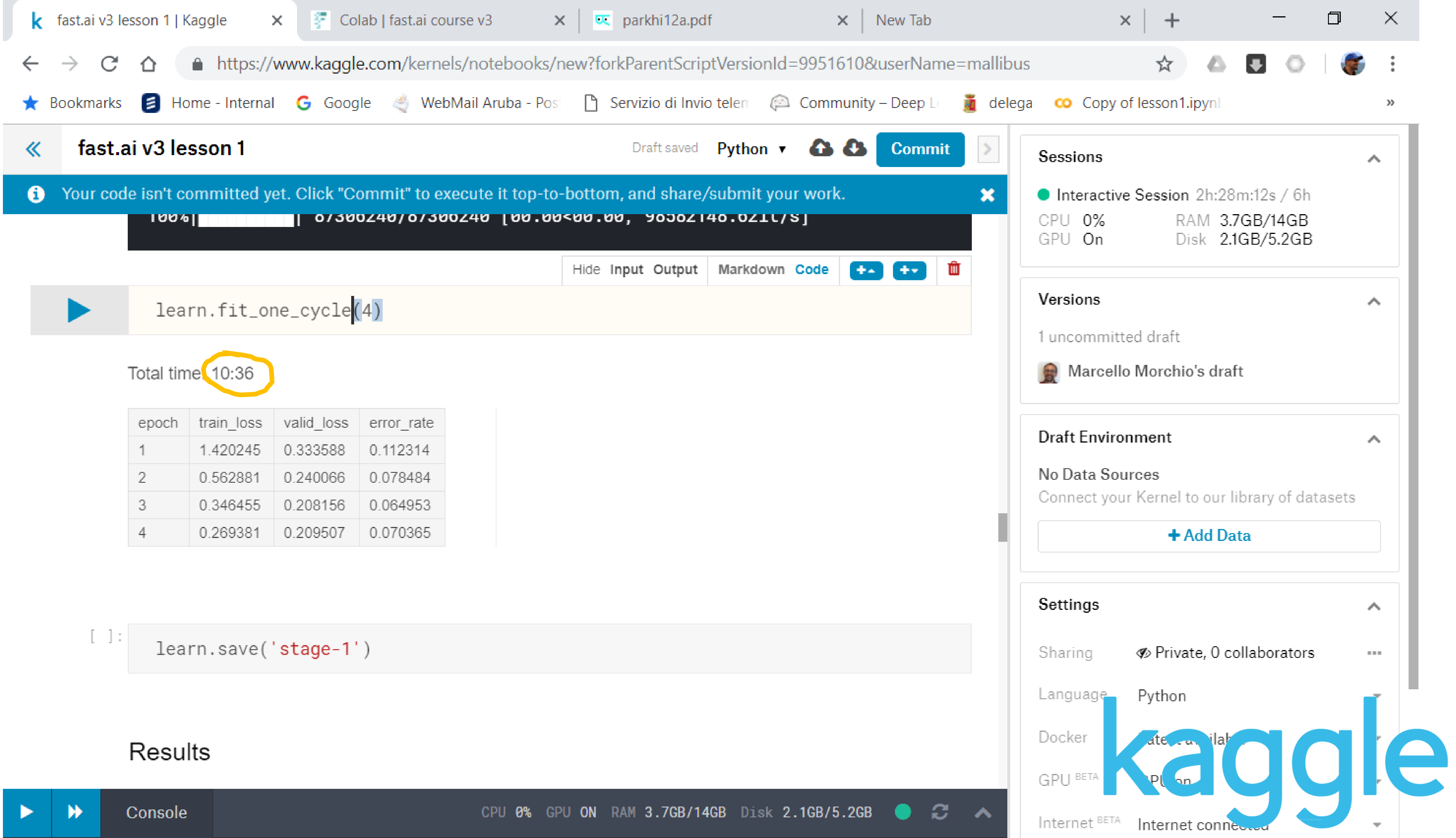
epoch	train_loss	valid_loss	error_rate
1	0.695744	0.302126	0.096076
2	0.405410	0.245322	0.082544
3	0.355702	0.216658	0.077131
4	0.287318	0.207957	0.064276
5	0.215132	0.201466	0.062246
6	0.154829	0.177019	0.056834
7	0.111082	0.178244	0.054127
8	0.084753	0.173119	0.051421

In [ ]:
```
learn.save('stage-1-50')
```

It's astonishing that it's possible to recognize pet breeds so accurately! Let's see if full fine-tuning helps:

In [ ]:
```
learn.unfreeze()
learn.fit_one_cycle(3, max_lr=slice(1e-6,1e-4))
```

Total time: 03:27



Your code isn't committed yet. Click "Commit" to execute it top-to-bottom, and share/submit your work.

```
100%|██████████| 87306240/87306240 [00:00<00:00, 98582148.62it/s]
```

```
learn.fit_one_cycle(4)
```

Total time: 10:36

epoch	train_loss	valid_loss	error_rate
1	1.420245	0.333588	0.112314
2	0.562881	0.240066	0.078484
3	0.346455	0.208156	0.064953
4	0.269381	0.209507	0.070365

```
[ ]: learn.save('stage-1')
```

Results

CPU 0% GPU ON RAM 3.7GB/14GB Disk 2.1GB/5.2GB

Sessions

Interactive Session 2h:28m:12s / 6h

CPU 0% RAM 3.7GB/14GB
GPU On Disk 2.1GB/5.2GB

Versions

1 uncommitted draft

Marcello Morchio's draft

Draft Environment

No Data Sources

Connect your Kernel to our library of datasets

+ Add Data

Settings

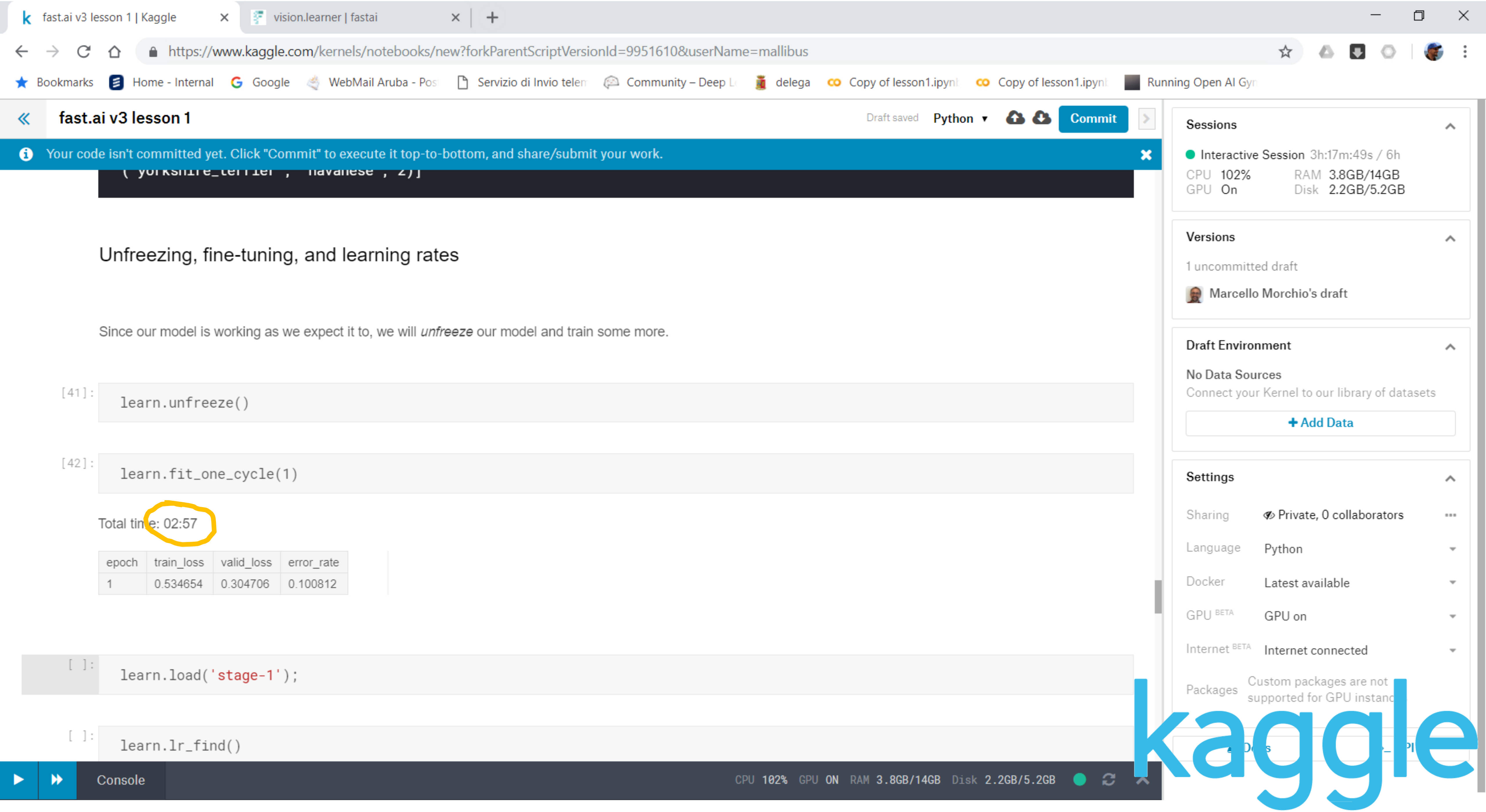
Sharing: Private, 0 collaborators

Language: Python

Docker: Latest available

GPU BETA: GPU on

Internet BETA: Internet connected



fast.ai v3 lesson 1

Your code isn't committed yet. Click "Commit" to execute it top-to-bottom, and share/submit your work.

```
 ('yorkshire_terrier', 'havanese', 2)]
```

Unfreezing, fine-tuning, and learning rates

Since our model is working as we expect it to, we will *unfreeze* our model and train some more.

```
[41]: learn.unfreeze()
```

```
[42]: learn.fit_one_cycle(1)
```

Total time: 02:57

epoch	train_loss	valid_loss	error_rate
1	0.534654	0.304706	0.100812

```
[ ]: learn.load('stage-1');
```

```
[ ]: learn.lr_find()
```

CPU 102% GPU ON RAM 3.8GB/14GB Disk 2.2GB/5.2GB

Sessions

Interactive Session 3h:17m:49s / 6h

CPU 102% RAM 3.8GB/14GB
GPU On Disk 2.2GB/5.2GB

Versions

1 uncommitted draft

Marcello Morchio's draft

Draft Environment

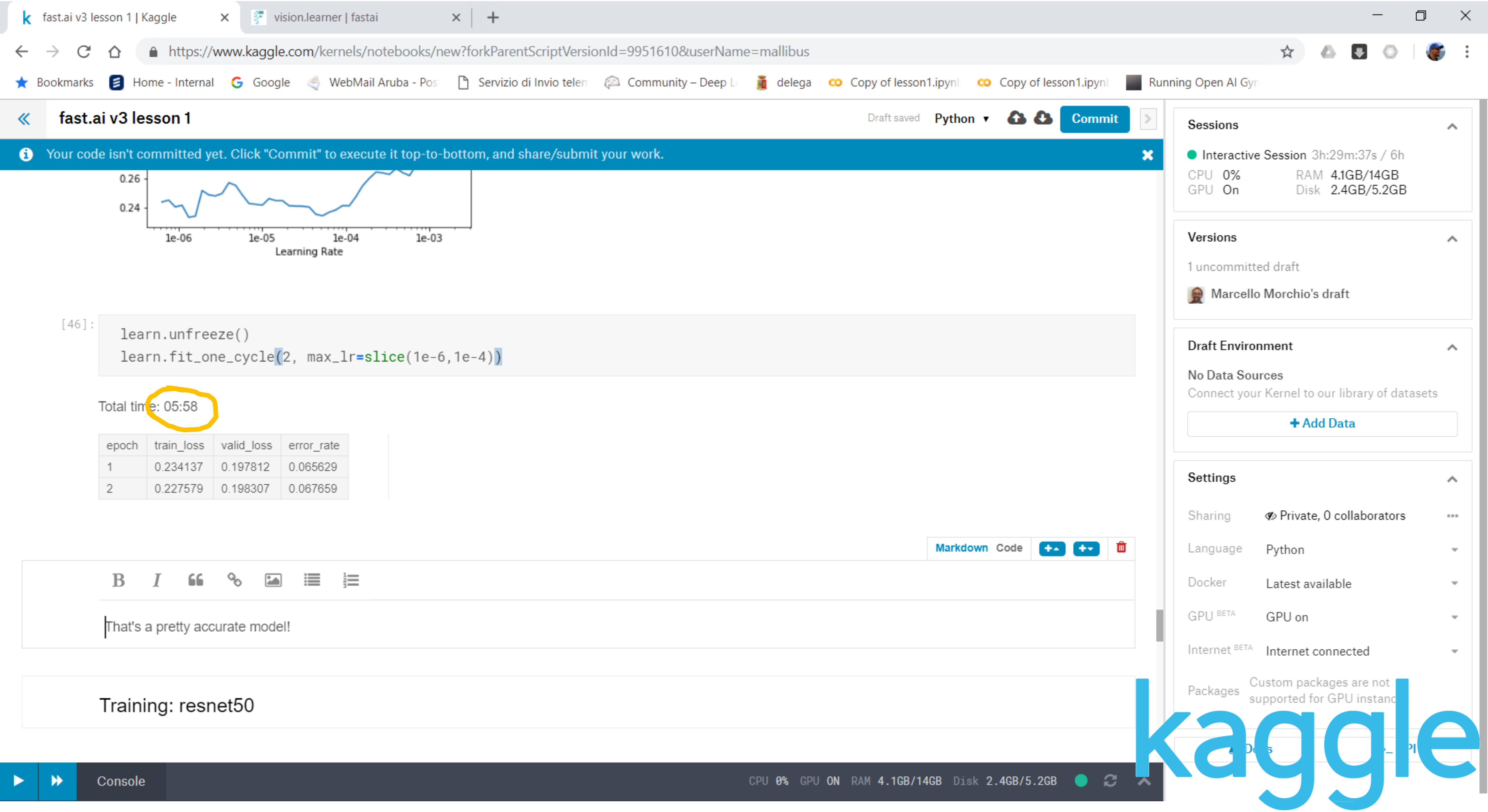
No Data Sources

Connect your Kernel to our library of datasets

+ Add Data

Settings

Sharing: Private, 0 collaborators
Language: Python
Docker: Latest available
GPU BETA: GPU on
Internet BETA: Internet connected
Packages: Custom packages are not supported for GPU instanc



```
[46]:
learn.unfreeze()
learn.fit_one_cycle(2, max_lr=slice(1e-6,1e-4))
```

Total time: 05:58

epoch	train_loss	valid_loss	error_rate
1	0.234137	0.197812	0.065629
2	0.227579	0.198307	0.067659

That's a pretty accurate model!

Training: resnet50

CPU 0% GPU ON RAM 4.1GB/14GB Disk 2.4GB/5.2GB

Sessions

Interactive Session 3h:29m:37s / 6h

CPU 0% RAM 4.1GB/14GB
GPU On Disk 2.4GB/5.2GB

Versions

1 uncommitted draft

Marcello Morchio's draft

Draft Environment

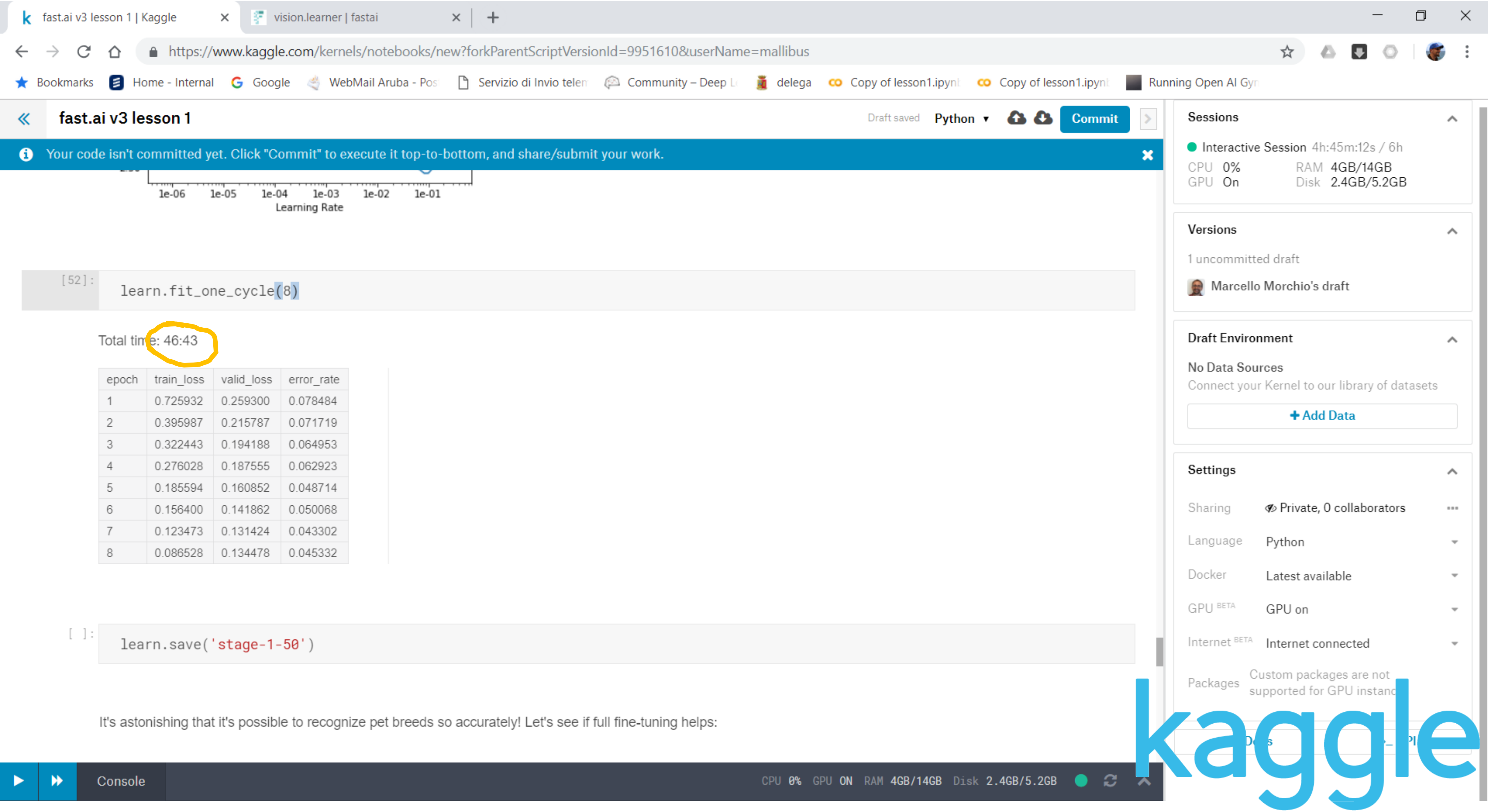
No Data Sources

Connect your Kernel to our library of datasets

+ Add Data

Settings

Sharing: Private, 0 collaborators
Language: Python
Docker: Latest available
GPU BETA: GPU on
Internet BETA: Internet connected
Packages: Custom packages are not supported for GPU instanc



Your code isn't committed yet. Click "Commit" to execute it top-to-bottom, and share/submit your work.

[52]:

```
learn.fit_one_cycle(8)
```

Total time: 46:43

epoch	train_loss	valid_loss	error_rate
1	0.725932	0.259300	0.078484
2	0.395987	0.215787	0.071719
3	0.322443	0.194188	0.064953
4	0.276028	0.187555	0.062923
5	0.185594	0.160852	0.048714
6	0.156400	0.141862	0.050068
7	0.123473	0.131424	0.043302
8	0.086528	0.134478	0.045332

[ ]:

```
learn.save('stage-1-50')
```

It's astonishing that it's possible to recognize pet breeds so accurately! Let's see if full fine-tuning helps:

Interactive Session 4h:45m:12s / 6h

CPU 0% RAM 4GB/14GB
GPU On Disk 2.4GB/5.2GB

1 uncommitted draft

Marcello Morchio's draft

No Data Sources

Connect your Kernel to our library of datasets

+ Add Data

Sharing: Private, 0 collaborators

Language: Python

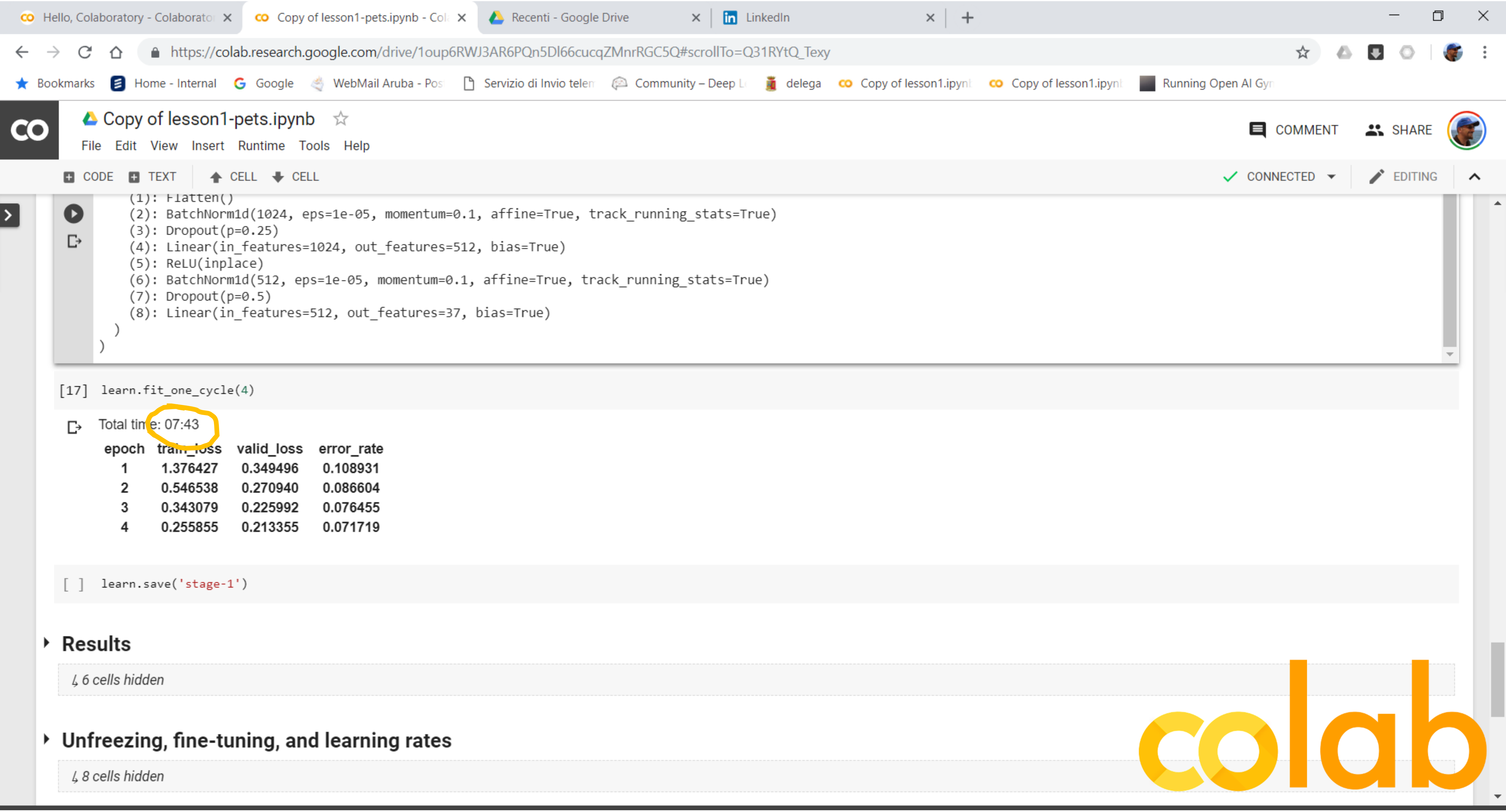
Docker: Latest available

GPU BETA: GPU on

Internet BETA: Internet connected

Packages: Custom packages are not supported for GPU instanc

kaggle



```
      (1): Flatten()
      (2): BatchNorm1d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (3): Dropout(p=0.25)
      (4): Linear(in_features=1024, out_features=512, bias=True)
      (5): ReLU(inplace)
      (6): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (7): Dropout(p=0.5)
      (8): Linear(in_features=512, out_features=37, bias=True)
    )
  )
```

```
[17] learn.fit_one_cycle(4)
```

```
Total time: 07:43
epoch  train_loss  valid_loss  error_rate
  1     1.376427    0.349496    0.108931
  2     0.546538    0.270940    0.086604
  3     0.343079    0.225992    0.076455
  4     0.255855    0.213355    0.071719
```

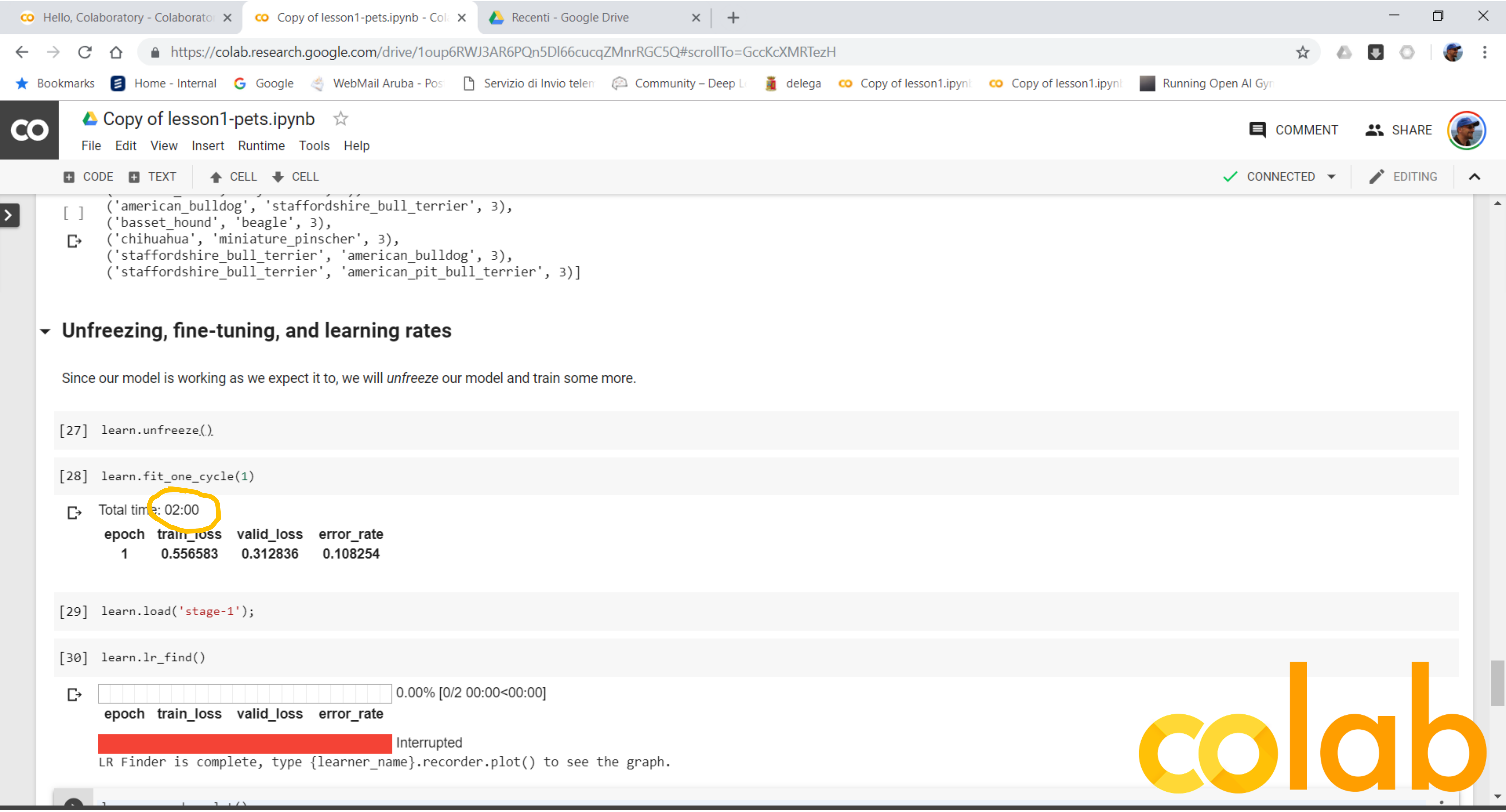
```
[ ] learn.save('stage-1')
```

Results

↳ 6 cells hidden

Unfreezing, fine-tuning, and learning rates

↳ 8 cells hidden



```
[ ]
 ('american_bulldog', 'staffordshire_bull_terrier', 3),
 ('basset_hound', 'beagle', 3),
 ('chihuahua', 'miniature_pinscher', 3),
 ('staffordshire_bull_terrier', 'american_bulldog', 3),
 ('staffordshire_bull_terrier', 'american_pit_bull_terrier', 3)]
```

Unfreezing, fine-tuning, and learning rates

Since our model is working as we expect it to, we will *unfreeze* our model and train some more.

```
[27] learn.unfreeze()
```

```
[28] learn.fit_one_cycle(1)
```

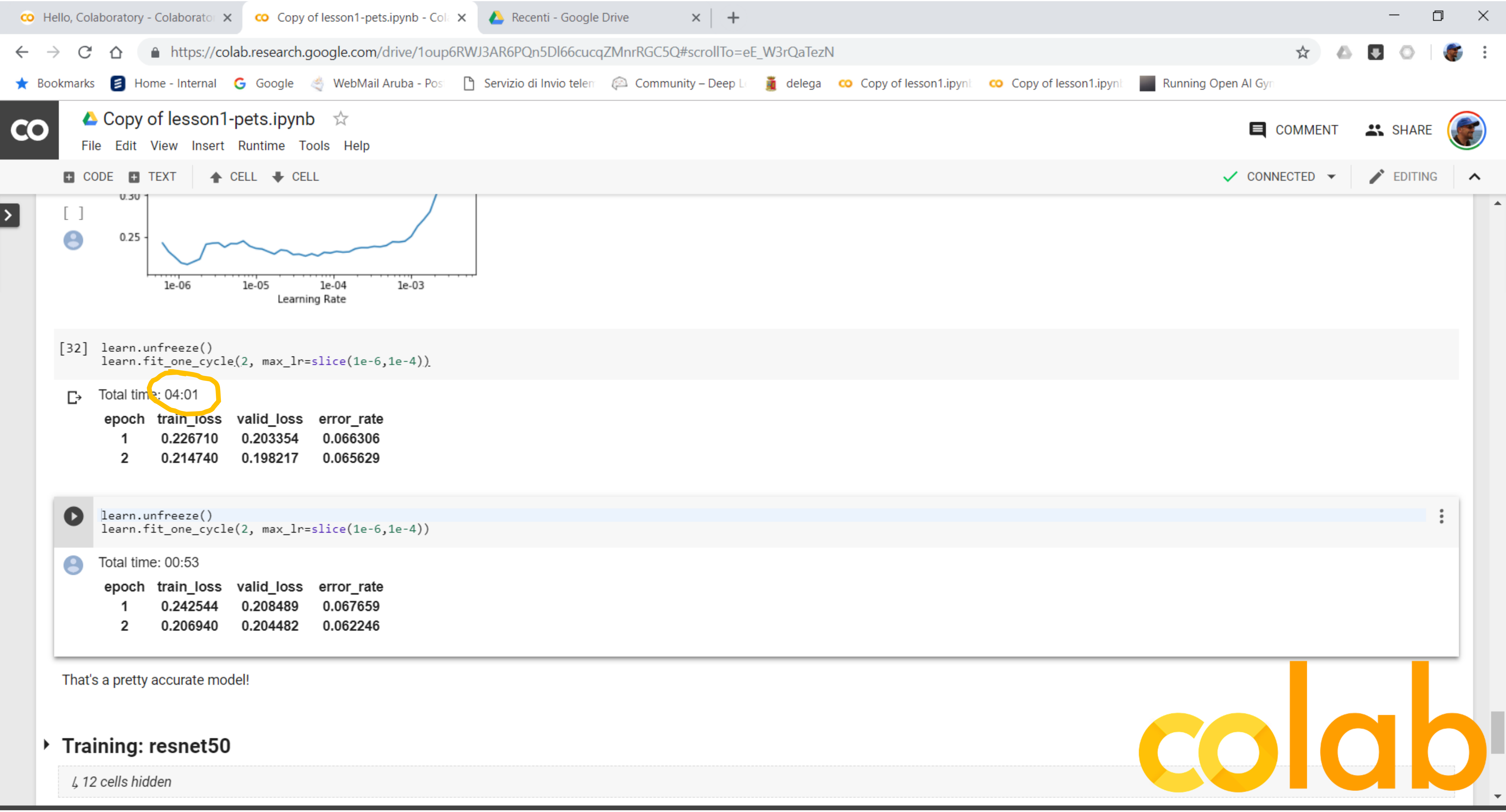
```
Total time: 02:00
epoch  train_loss  valid_loss  error_rate
  1     0.556583    0.312836    0.108254
```

```
[29] learn.load('stage-1');
```

```
[30] learn.lr_find()
```

```
0.00% [0/2 00:00<00:00]
epoch  train_loss  valid_loss  error_rate

Interrupted
LR Finder is complete, type {learner_name}.recorder.plot() to see the graph.
```



```
[32] learn.unfreeze()
     learn.fit_one_cycle(2, max_lr=slice(1e-6,1e-4))
```

```
Total time: 04:01
epoch  train_loss  valid_loss  error_rate
  1    0.226710    0.203354    0.066306
  2    0.214740    0.198217    0.065629
```

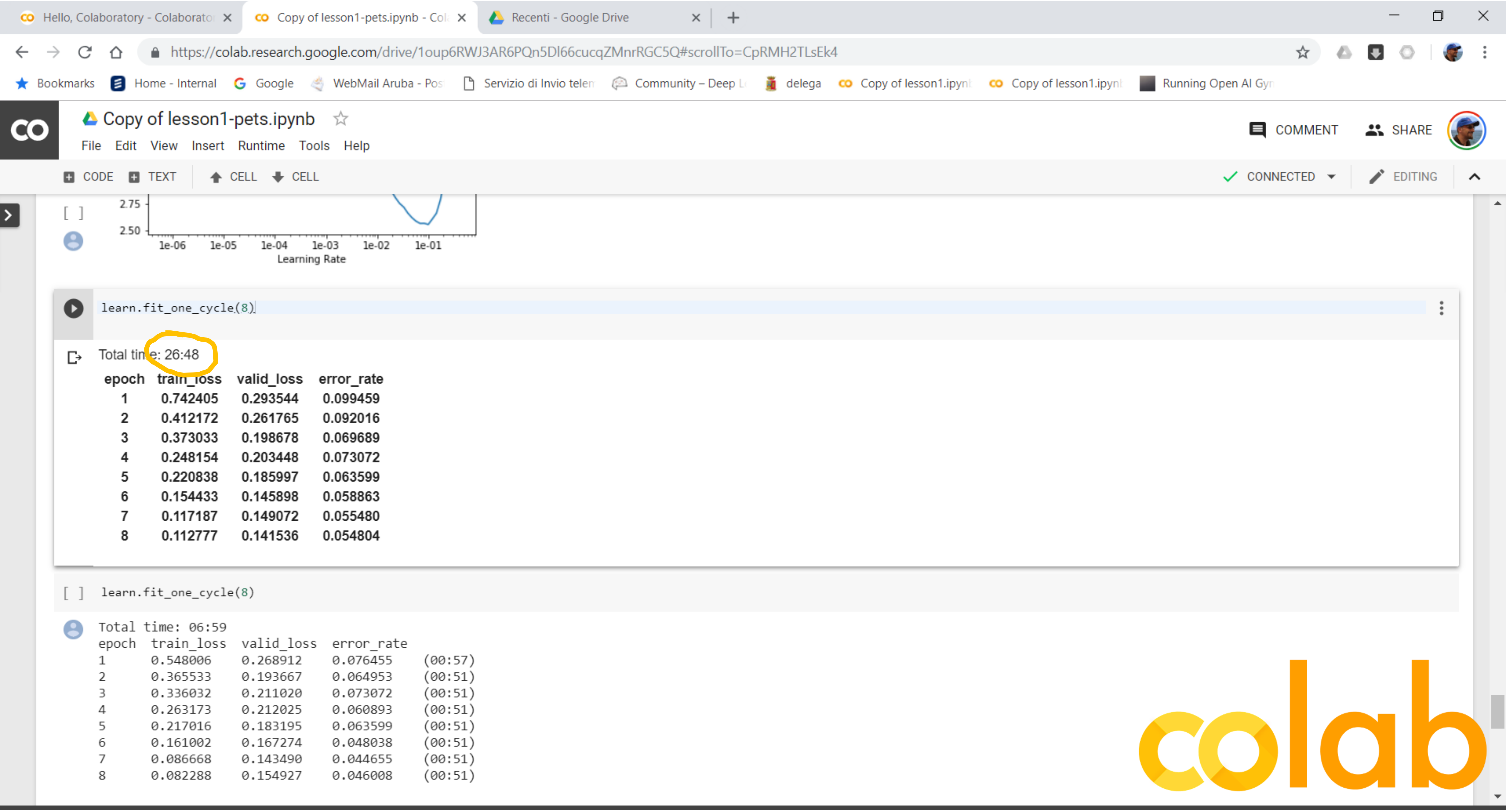
```
learn.unfreeze()
learn.fit_one_cycle(2, max_lr=slice(1e-6,1e-4))
```

```
Total time: 00:53
epoch  train_loss  valid_loss  error_rate
  1    0.242544    0.208489    0.067659
  2    0.206940    0.204482    0.062246
```

That's a pretty accurate model!

Training: resnet50

↳ 12 cells hidden



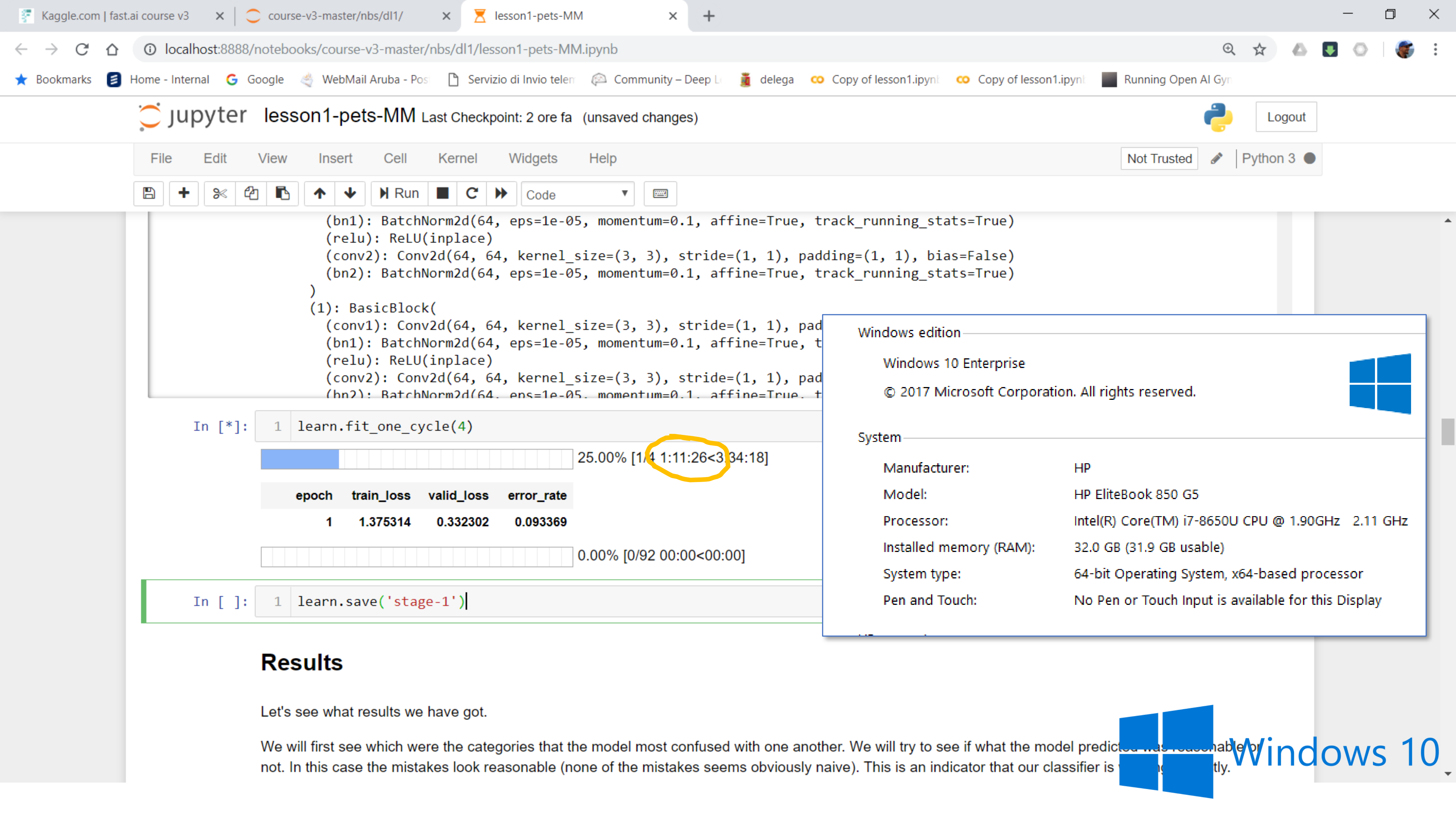
```
learn.fit_one_cycle(8)
```

Total time: 26:48

epoch	train_loss	valid_loss	error_rate
1	0.742405	0.293544	0.099459
2	0.412172	0.261765	0.092016
3	0.373033	0.198678	0.069689
4	0.248154	0.203448	0.073072
5	0.220838	0.185997	0.063599
6	0.154433	0.145898	0.058863
7	0.117187	0.149072	0.055480
8	0.112777	0.141536	0.054804

```
learn.fit_one_cycle(8)
```

```
Total time: 06:59
epoch  train_loss  valid_loss  error_rate
1      0.548006    0.268912    0.076455    (00:57)
2      0.365533    0.193667    0.064953    (00:51)
3      0.336032    0.211020    0.073072    (00:51)
4      0.263173    0.212025    0.060893    (00:51)
5      0.217016    0.183195    0.063599    (00:51)
6      0.161002    0.167274    0.048038    (00:51)
7      0.086668    0.143490    0.044655    (00:51)
8      0.082288    0.154927    0.046008    (00:51)
```



```
          (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (relu): ReLU(inplace)
          (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
          (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        )
        (1): BasicBlock(
          (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), pad
          (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, t
          (relu): ReLU(inplace)
          (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), pad
          (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, t
```

In [*]:

```
learn.fit_one_cycle(4)
```

```
25.00% [1/4 1:11:26<3:34:18]
```

epoch	train_loss	valid_loss	error_rate
1	1.375314	0.332302	0.093369

```
0.00% [0/92 00:00<00:00]
```

In [ ]:

```
learn.save('stage-1')
```

Windows edition

Windows 10 Enterprise

© 2017 Microsoft Corporation. All rights reserved.

System

Manufacturer:	HP
Model:	HP EliteBook 850 G5
Processor:	Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz 2.11 GHz
Installed memory (RAM):	32.0 GB (31.9 GB usable)
System type:	64-bit Operating System, x64-based processor
Pen and Touch:	No Pen or Touch Input is available for this Display

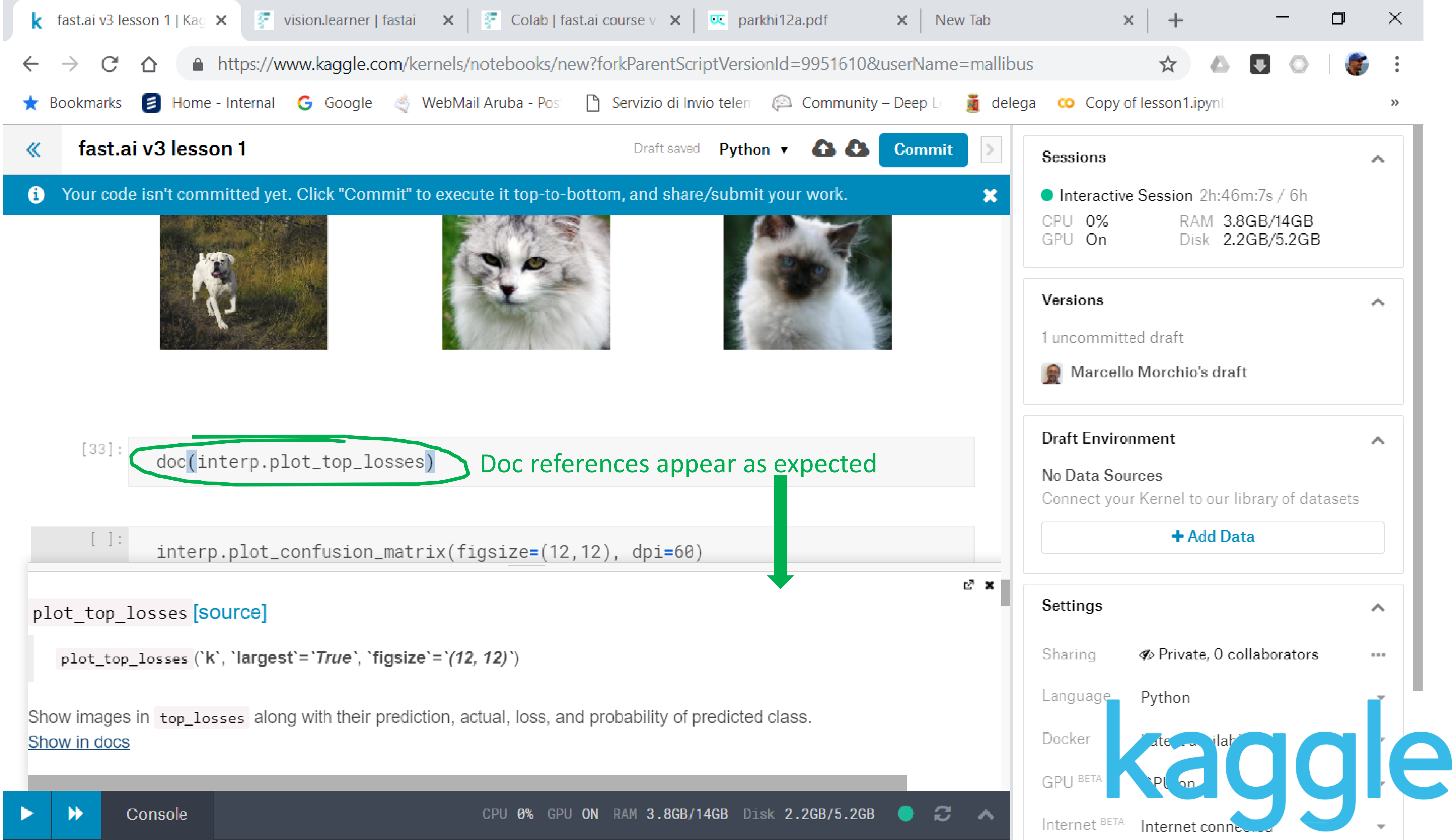
Results

Let's see what results we have got.

We will first see which were the categories that the model most confused with one another. We will try to see if what the model predic[…]able or not. In this case the mistakes look reasonable (none of the mistakes seems obviously naive). This is an indicator that our classifier is [...]tly.

Utilities

- Documentation visualization
  - Kaggle: OK - doc(function) pops up the text and the links
  - Colabs: Not OK - doc(function) does not pop up anything



fast.ai v3 lesson 1

Your code isn't committed yet. Click "Commit" to execute it top-to-bottom, and share/submit your work.

```
[33]: doc(interp.plot_top_losses)
```

Doc references appear as expected

```
[ ]: interp.plot_confusion_matrix(figsize=(12,12), dpi=60)
```

plot_top_losses [source]

plot_top_losses(`k`, `largest`=`True`, `figsize`=`(12, 12)`)

Show images in `top_losses` along with their prediction, actual, loss, and probability of predicted class.

Show in docs

```
CPU 0%  GPU ON  RAM 3.8GB/14GB  Disk 2.2GB/5.2GB
```

Sessions

Interactive Session 2h:46m:7s / 6h

CPU 0% RAM 3.8GB/14GB
GPU On Disk 2.2GB/5.2GB

Versions

1 uncommitted draft

Marcello Morchio's draft

Draft Environment

No Data Sources

Connect your Kernel to our library of datasets

+ Add Data

Settings

Sharing: Private, 0 collaborators

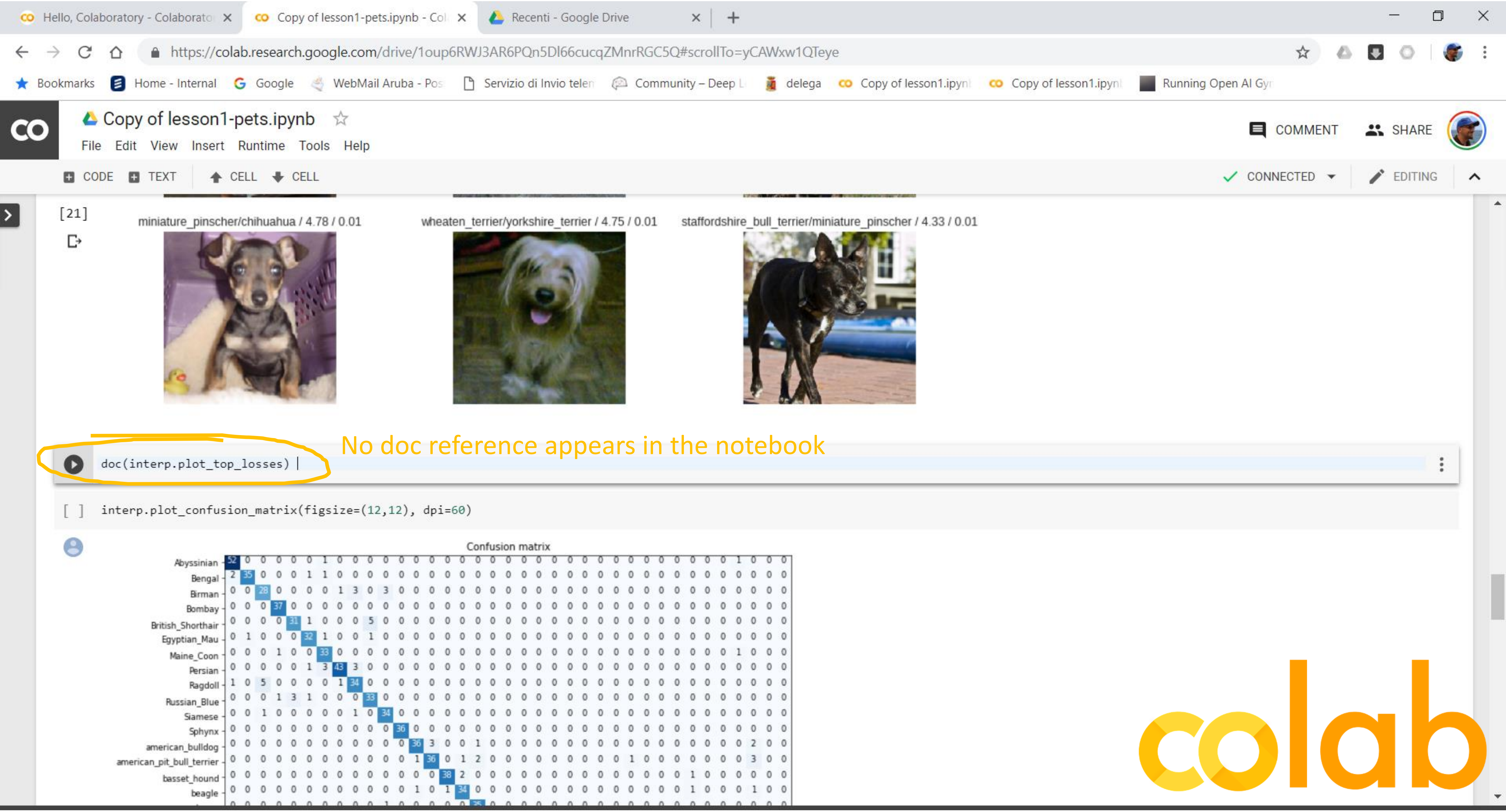
Language: Python

Docker

GPU BETA

Internet BETA: Internet conne

kaggle



miniature_pinscher/chihuahua / 4.78 / 0.01

wheaten_terrier/yorkshire_terrier / 4.75 / 0.01

staffordshire_bull_terrier/miniature_pinscher / 4.33 / 0.01

```
doc(interp.plot_top_losses)
```

No doc reference appears in the notebook

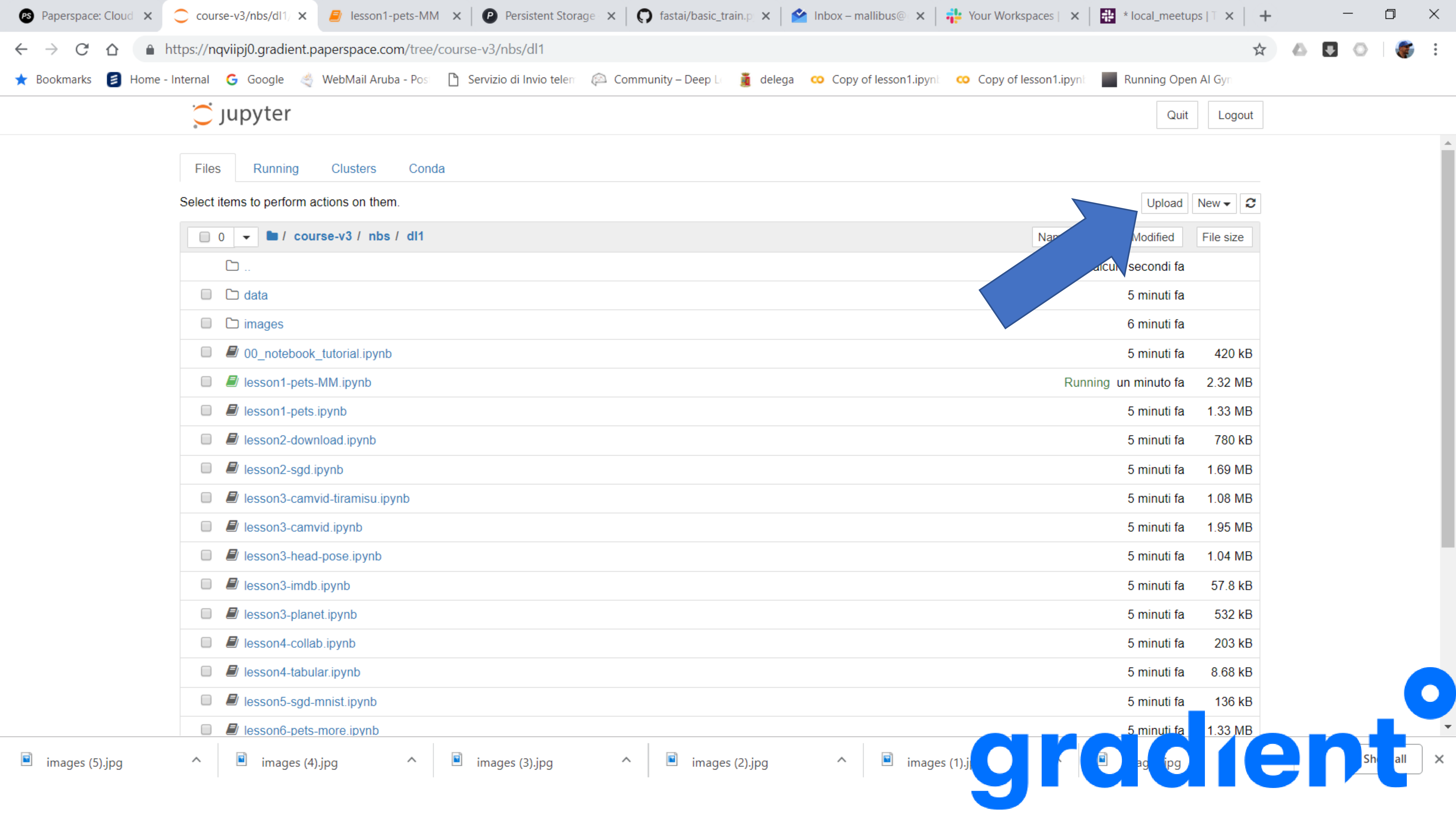
```
interp.plot_confusion_matrix(figsize=(12,12), dpi=60)
```

Confusion matrix

Abyssinian	52	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Bengal	2	35	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Birman	0	0	28	0	0	0	0	1	3	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bombay	0	0	0	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
British_Shorthair	0	0	0	0	31	1	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Egyptian_Mau	0	1	0	0	0	32	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maine_Coon	0	0	0	1	0	0	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Persian	0	0	0	0	0	1	3	43	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ragdoll	1	0	5	0	0	0	0	1	34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Russian_Blue	0	0	0	1	3	1	0	0	0	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Siamese	0	0	1	0	0	0	0	0	1	0	34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sphynx	0	0	0	0	0	0	0	0	0	0	0	36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
american_bulldog	0	0	0	0	0	0	0	0	0	0	0	0	36	3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
american_pit_bull_terrier	0	0	0	0	0	0	0	0	0	0	0	0	1	36	0	1	2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	3	0	0
basset_hound	0	0	0	0	0	0	0	0	0	0	0	0	0	0	38	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
beagle	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Importing your dataset

- Gradient
  - Upload data directly from you Jupyter notebook
- Kaggle
  - Add Data, zip the root folder and upload
  - Your dataset will be in /kaggle/input/
- Colab
  - Add the sippet to mount your Google Drive in the Colab instance file syste
  - Locate it in the file tab



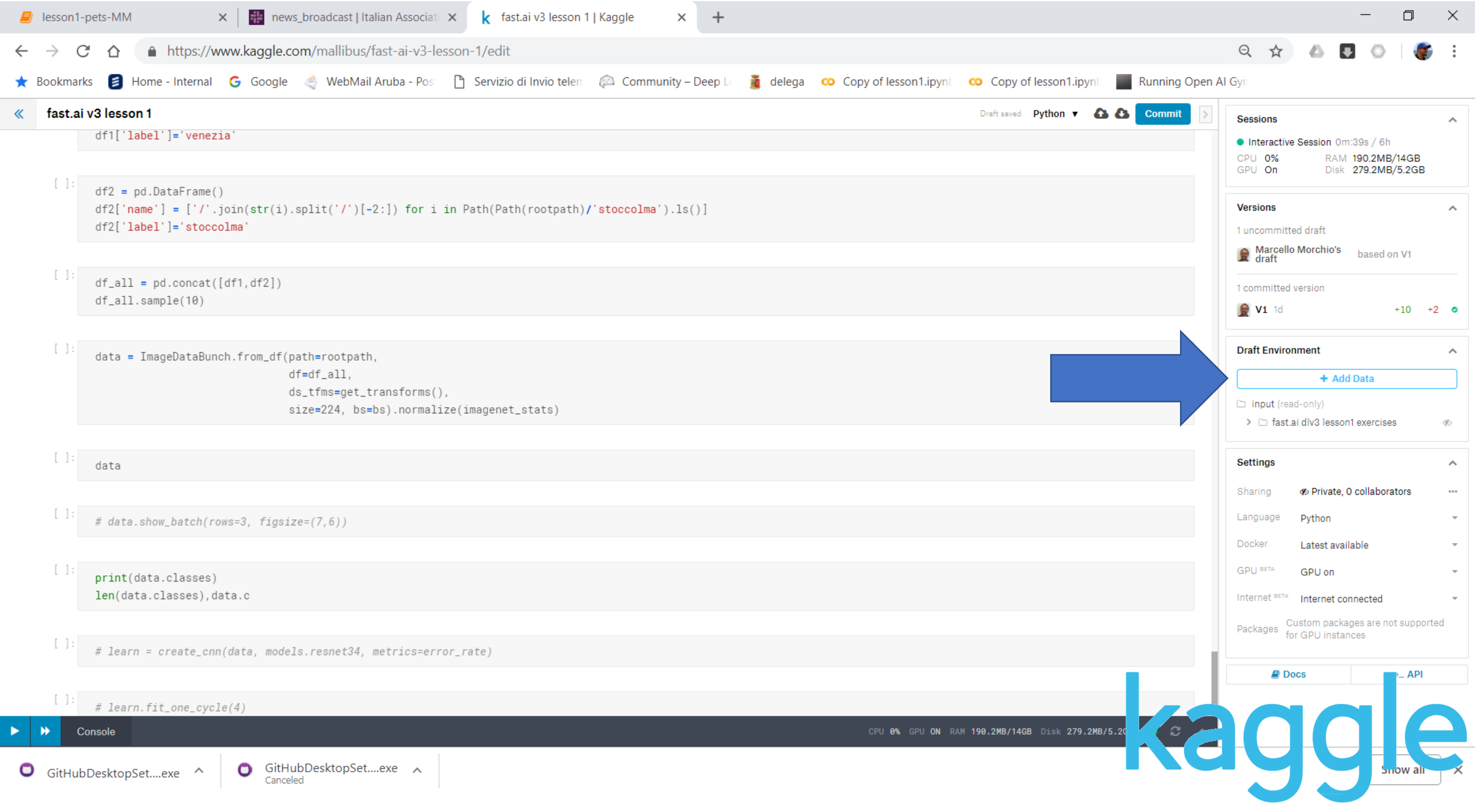
https://nqviipj0.gradient.paperspace.com/tree/course-v3/nbs/dl1

Select items to perform actions on them.

□ 0 / course-v3 / nbs / dl1

Na	Modified	File size
..	alcu secondi fa	
□ data	5 minuti fa	
□ images	6 minuti fa	
□ 00_notebook_tutorial.ipynb	5 minuti fa	420 kB
□ lesson1-pets-MM.ipynb	Running un minuto fa	2.32 MB
□ lesson1-pets.ipynb	5 minuti fa	1.33 MB
□ lesson2-download.ipynb	5 minuti fa	780 kB
□ lesson2-sgd.ipynb	5 minuti fa	1.69 MB
□ lesson3-camvid-tiramisu.ipynb	5 minuti fa	1.08 MB
□ lesson3-camvid.ipynb	5 minuti fa	1.95 MB
□ lesson3-head-pose.ipynb	5 minuti fa	1.04 MB
□ lesson3-imdb.ipynb	5 minuti fa	57.8 kB
□ lesson3-planet.ipynb	5 minuti fa	532 kB
□ lesson4-collab.ipynb	5 minuti fa	203 kB
□ lesson4-tabular.ipynb	5 minuti fa	8.68 kB
□ lesson5-sgd-mnist.ipynb	5 minuti fa	136 kB
□ lesson6-pets-more.ipynb	5 minuti fa	1.33 MB

images (5).jpg
images (4).jpg
images (3).jpg
images (2).jpg
images (1).j
ag jpg
Sh all



```
df1['label']='venezia'
```

```
df2 = pd.DataFrame()
df2['name'] = ['/'.join(str(i).split('/')[-2:]) for i in Path(Path(rootpath)/'stoccolma').ls()]
df2['label']='stoccolma'
```

```
df_all = pd.concat([df1,df2])
df_all.sample(10)
```

```
data = ImageDataBunch.from_df(path=rootpath,
                              df=df_all,
                              ds_tfms=get_transforms(),
                              size=224, bs=bs).normalize(imagenet_stats)
```

```
data
```

```
# data.show_batch(rows=3, figsize=(7,6))
```

```
print(data.classes)
len(data.classes),data.c
```

```
# learn = create_cnn(data, models.resnet34, metrics=error_rate)
```

```
# learn.fit_one_cycle(4)
```

Sessions

Interactive Session 0m:39s / 6h

CPU 0% RAM 190.2MB/14GB
GPU On Disk 279.2MB/5.2GB

Versions

1 uncommitted draft

Marcello Morchio's draft based on V1

1 committed version

V1 1d +10 +2

Draft Environment

+ Add Data

input (read-only)

fast.ai dlv3 lesson1 exercises

Settings

Sharing: Private, 0 collaborators

Language: Python

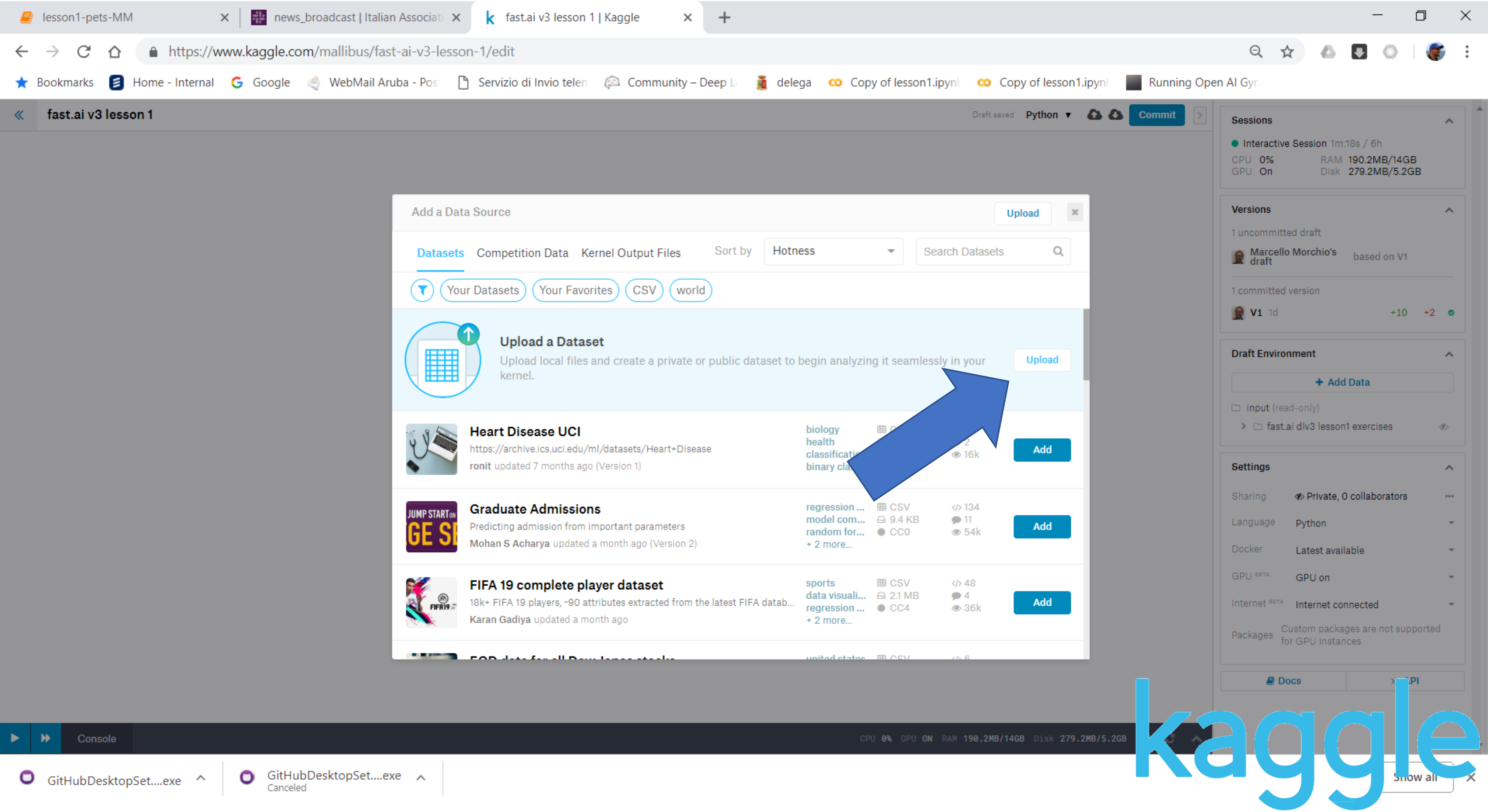
Docker: Latest available

GPU BETA: GPU on

Internet BETA: Internet connected

Packages: Custom packages are not supported for GPU instances

kaggle



Add a Data Source

Upload a Dataset

Upload local files and create a private or public dataset to begin analyzing it seamlessly in your kernel.

Heart Disease UCI
https://archive.ics.uci.edu/ml/datasets/Heart+Disease
ronit updated 7 months ago (Version 1)
biology
health
classificati
binary cla
16k

Graduate Admissions
Predicting admission from important parameters
Mohan S Acharya updated a month ago (Version 2)
regression ...
model com...
random for...
+ 2 more...
CSV
9.4 KB
CC0
134
11
54k

FIFA 19 complete player dataset
18k+ FIFA 19 players, ~90 attributes extracted from the latest FIFA datab...
Karan Gadiya updated a month ago
sports
data visuali...
regression ...
+ 2 more...
CSV
2.1 MB
CC4
48
4
36k

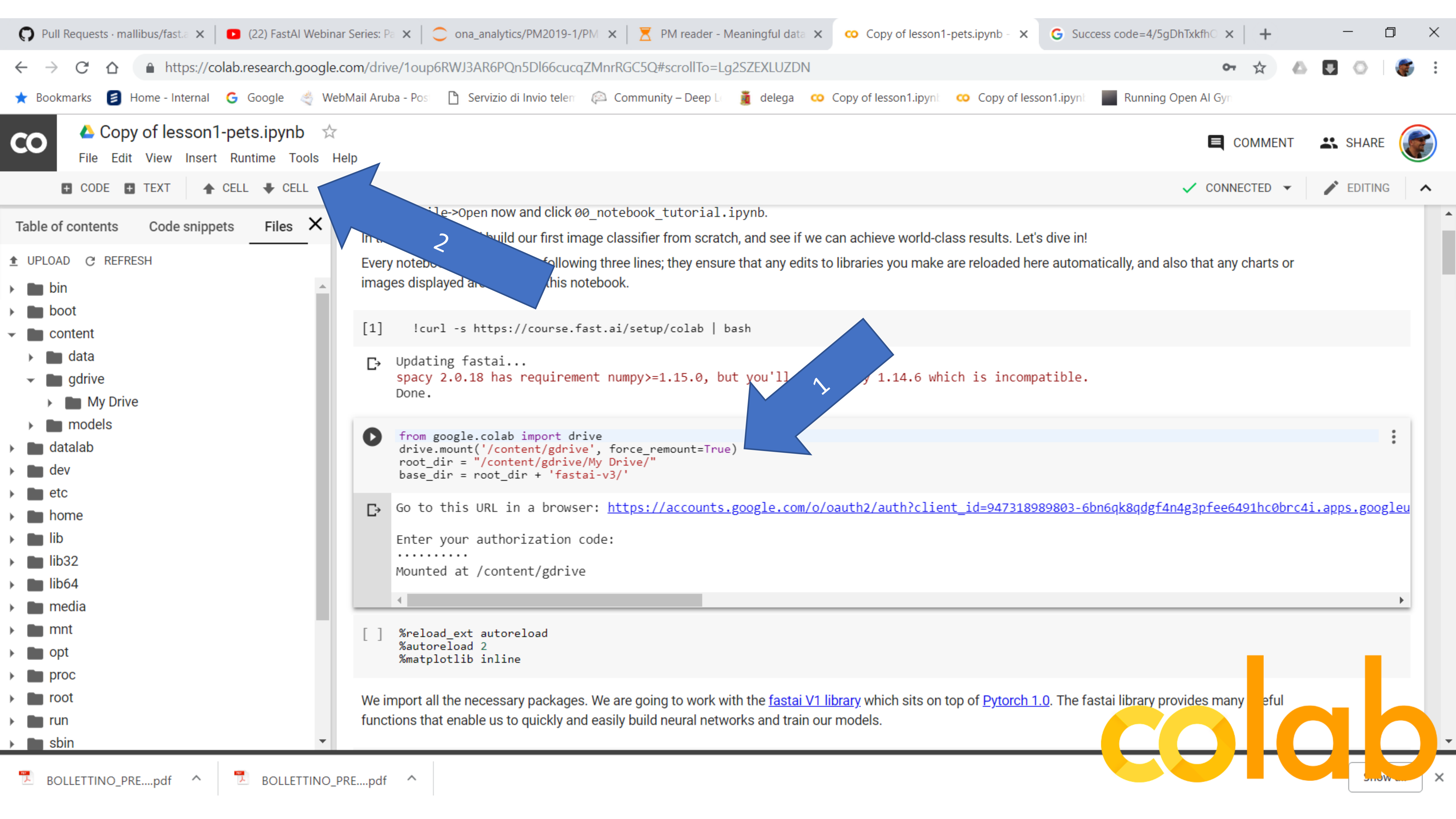
Sessions
Interactive Session 1m:18s / 6h
CPU 0% RAM 190.2MB/14GB
GPU On Disk 279.2MB/5.2GB

Versions
1 uncommitted draft
Marcello Morchio's draft based on V1
1 committed version
V1 1d +10 +2

Draft Environment
input (read-only)
fast.ai dlv3 lesson1 exercises

Settings
Sharing Private, 0 collaborators
Language Python
Docker Latest available
GPU BETA GPU on
Internet BETA Internet connected
Packages Custom packages are not supported for GPU instances

```
CPU 0%  GPU ON  RAM 190.2MB/14GB  Disk 279.2MB/5.2GB
```



…ile->Open now and click 00_notebook_tutorial.ipynb.

In t… build our first image classifier from scratch, and see if we can achieve world-class results. Let's dive in!

Every noteb… following three lines; they ensure that any edits to libraries you make are reloaded here automatically, and also that any charts or images displayed ar… this notebook.

```
[1]  !curl -s https://course.fast.ai/setup/colab | bash
```

```
Updating fastai...
spacy 2.0.18 has requirement numpy>=1.15.0, but you'll … 1.14.6 which is incompatible.
Done.
```

```
from google.colab import drive
drive.mount('/content/gdrive', force_remount=True)
root_dir = "/content/gdrive/My Drive/"
base_dir = root_dir + 'fastai-v3/'
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleu

Enter your authorization code:
··········
Mounted at /content/gdrive
```

```
[ ]  %reload_ext autoreload
     %autoreload 2
     %matplotlib inline
```

We import all the necessary packages. We are going to work with the fastai V1 library which sits on top of Pytorch 1.0. The fastai library provides many …eful functions that enable us to quickly and easily build neural networks and train our models.

Issues and solutions

(when solutions are available)

Regular Expressions in Windos file system

AttributeError: ‘NoneType’ object has no attribute ‘group’

Running on Windows the regular expressions referred to pathnames in the notebooks shall be changed

```
pat = re.compile(r'/([^/]+)_\d+.jpg$') # for linux
pat = re.compile(r'\\([^\\]+)_\d+.jpg$') #for windows
```

OR

```
pat = re.compile(r'[/\\]([^/\\]+)\d+.jpg$’) #For both
```

https://github.com/fastai/course-v3/issues/118

Windows 10

Small shared memory in Kaggle?

RuntimeError: DataLoader worker (pid 54) is killed by signal: Bus error.

Code of lession1 runs ok, but I tried to upload of images of mine and I got the error above when

```
data.show_batch(rows=3, figsize=(7,6))
```

Tried with parameter num_workers=0 as suggested around, no success