

# Step 1: login to Kaggle kernels

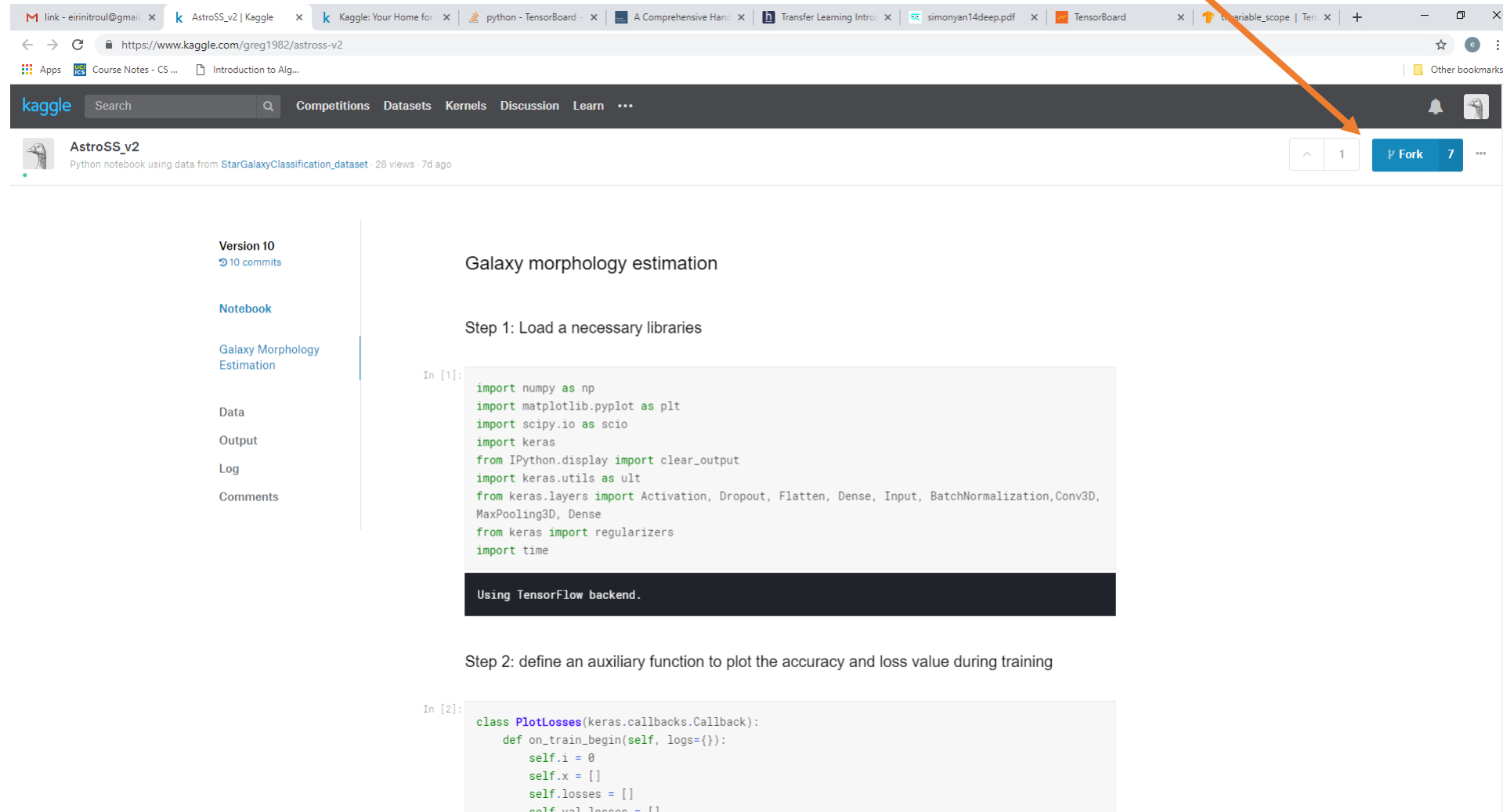
<https://www.kaggle.com/kernels>

The image shows a browser window at the URL <https://www.kaggle.com/kernels>. The page title is "Kernels | Kaggle". The navigation bar includes "kaggle", a search bar, and links for "Competitions", "Datasets", "Kernels", "Discussion", and "Learn". A "Sign In" button is in the top right. The main content area is titled "Kernels" and features a "Documentation" and "New Kernel" button. A modal window is open in the center, titled "Sign in or register with one click:". It offers three social login options: "Sign in with Facebook", "Sign in with Google", and "Sign in with Yahoo". Below these is an "or" separator and a section for "Use your Kaggle username or email:" with a "Register with email" link. The login form includes fields for "Username or Email" and "Password", a "Sign in" button, and a "Remember me" checkbox. A "Forgot Username / Password" link is also present. At the bottom of the modal, there is a disclaimer: "One account per individual. e.g., if you're joining as a company, please create one account for each participant." and instructions for users with university addresses or multiple email accounts. The background shows a list of kernels, including "Starter\_PyTorch", "My Titanic Try", "Interactive Suicid", "Object Oriented", "eda + factor analy", "EDA and Cluster A", "Titanic Survival P", "Aerial Cactus C", "overfit is a nightmare", and "Cardiovascular Disease Analysis".

## Step 2: Go to my kernel

<https://www.kaggle.com/greg1982/astross-v2>

## Step 3: Click on Fork



The screenshot shows a web browser window with the Kaggle website. The address bar displays the URL <https://www.kaggle.com/greg1982/astross-v2>. The page title is "AstroSS\_v2" and it is described as a "Python notebook using data from StarGalaxyClassification\_dataset". The notebook content is titled "Galaxy morphology estimation" and includes two code blocks. The first code block, labeled "Step 1: Load a necessary libraries", contains the following Python code:

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
import scipy.io as scio
import keras
from IPython.display import clear_output
import keras.utils as ult
from keras.layers import Activation, Dropout, Flatten, Dense, Input, BatchNormalization, Conv3D, MaxPooling3D, Dense
from keras import regularizers
import time
```

Below the code block, a black box displays the output: "Using TensorFlow backend."

The second code block, labeled "Step 2: define an auxiliary function to plot the accuracy and loss value during training", contains the following Python code:

```
In [2]: class PlotLosses(keras.callbacks.Callback):
def on_train_begin(self, logs={}):
self.i = 0
self.x = []
self.losses = []
self.val_losses = []
```

The interface also shows a sidebar on the left with navigation options: Version 10 (10 commits), Notebook, Galaxy Morphology Estimation, Data, Output, Log, and Comments. In the top right corner, there are navigation icons, a "Fork" button with a count of 7, and a "1" icon. An orange arrow points from the text "Step 3: Click on Fork" to the "Fork" button.

# You should see something like this

The screenshot shows a Kaggle notebook titled "AstroSS\_v2" in a "Draft" state. The browser tabs include "link - eirintrouli@gmail", "AstroSS\_v2 | Kaggle", "Kaggle: Your Home for...", "python - TensorBoard", "A Comprehensive Hand...", "Transfer Learning Intro...", "simonyan14deep.pdf", "TensorBoard", and "tf.variable\_scope | Ter...". The notebook's top bar shows "AstroSS\_v2" with a "Draft saved" indicator, a "File Edit Insert Run Help" menu, and buttons for "Add Dataset" and "Commit". A blue progress bar at the top indicates "data-v1: Downloaded 1/2 files: 321.03MB/605.24MB [53.04%] 3.50s (91.67MB/s)".

## Galaxy morphology estimation

### Step 1: Load a necessary libraries

```
In[]: import numpy as np
import matplotlib.pyplot as plt
import scipy.io as scio
import keras
from IPython.display import clear_output
import keras.utils as ult
from keras.layers import Activation, Dropout, Flatten, Dense, Input, BatchNormalization, Conv3D, MaxPooling3D, Dense
from keras import regularizers
import time
```

### Step 2: define an auxiliary function to plot the accuracy and loss value during training

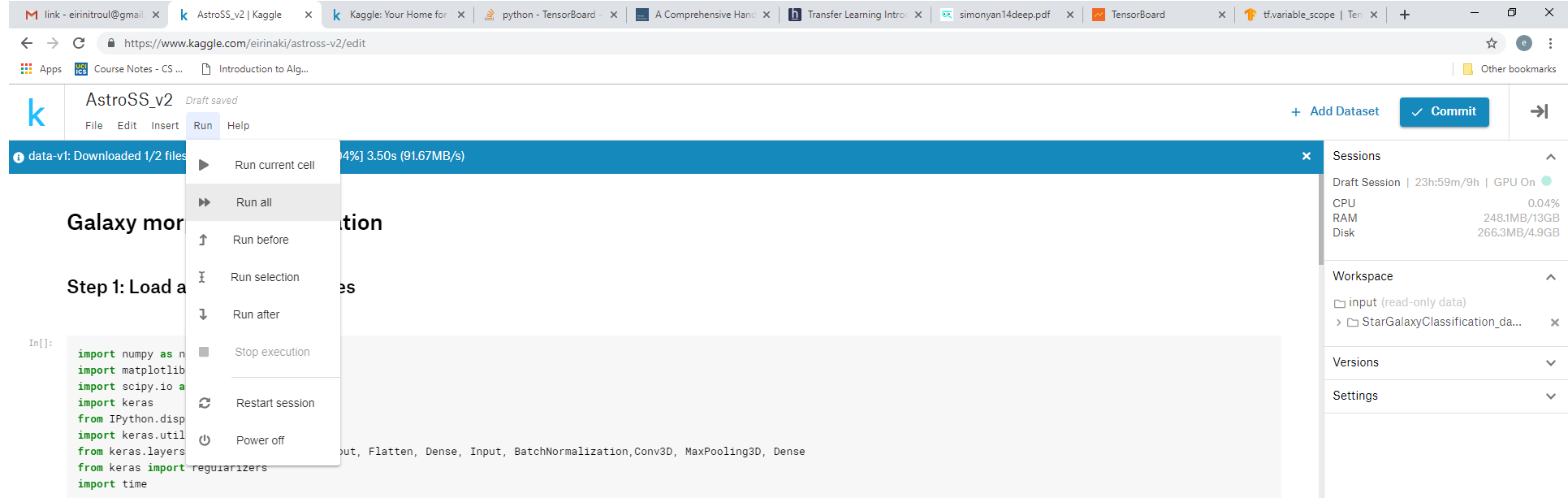
```
In[]: class PlotLosses(keras.callbacks.Callback):
    def on_train_begin(self, logs={}):
        self.i = 0
        self.x = []
        self.losses = []
        self.val_losses = []
        self.losses2 = []
        self.val_losses2 = []

        self.fig = plt.figure()
```

The right sidebar contains a "Sessions" panel showing "Draft Session | 23h:59m/9h | GPU On" with resource usage: CPU 4.72%, RAM 164.3MB/13GB, and Disk 266.1MB/4.9GB. Below it are "Workspace" (input (read-only data) and StarGalaxyClassification\_da...), "Versions", and "Settings" panels.

The bottom console bar shows "Console" and "Draft Session | CPU 5% | RAM 164.3MB/13GB | GPU On | Disk 266.1MB/4.9GB".

## Step 3: Click on Run -> Run all



The screenshot shows a Kaggle notebook titled 'AstroSS\_v2'. The 'Run' menu is open, and the 'Run all' option is highlighted. The notebook content includes a code cell with the following Python code:

```
In[]: import numpy as np
import matplotlib
import scipy.io as sio
import keras
from IPython.display import clear_output
import keras.utils
from keras.layers import Conv3D, MaxPooling3D, Dense
from keras import regularizers
import time
```

Below the code cell, the text 'Step 2: define an auxiliary function to plot the accuracy and loss value during training' is visible. The code cell below it contains the following Python code:

```
In[]: class PlotLosses(keras.callbacks.Callback):
    def on_train_begin(self, logs={}):
        self.i = 0
        self.x = []
        self.losses = []
        self.val_losses = []
        self.losses2 = []
        self.val_losses2 = []

        self.fig = plt.figure()
```

The right sidebar shows session information: Draft Session | 23h:59m/9h | GPU On | CPU 0.04% | RAM 248.1MB/13GB | Disk 266.3MB/4.9GB. The bottom status bar shows: Draft Session | CPU 0% | RAM 248.1MB/13GB | GPU On | Disk 266.3MB/4.9GB.

# The code should run and you should see things like that

The screenshot shows a Kaggle notebook titled "AstroSS\_v2" with the following code in a cell:

```
history=model_1.fit(Train_data,train_labels_cat, batch_size=100, epochs=10,validation_data=[Test_data,test_labels_cat],callbacks=[plot_losses],shuffle=True)
elapsed_time = time.time() - start_time
time.strftime("%H:%M:%S", time.gmtime(elapsed_time))
```

Below the code, two line plots are displayed. The left plot shows Accuracy vs. Epoch, with Training accuracy (solid blue line) and Testing accuracy (dashed orange line) both starting at approximately 0.65 and remaining stable around 0.7. The right plot shows Loss vs. Epoch, with Training loss (solid blue line) and Testing loss (dashed orange line) both starting at approximately 250 and decreasing to near 0 by epoch 10.

The output of the cell is a time string: '00:00:15'. Below this, a new code cell is shown with the following code:

```
ls,acc=model_1.evaluate(Test_data,test_labels_cat)
print("Loss value: %.2f" % (ls))
print("Accuracy: %.1f" % (acc*100))
```

The output of this cell is:

```
500/500 [=====] - 0s 111us/step
Loss value: 6.20
Accuracy: 68.0
```

A third code cell is shown with the following code:

```
preds=model_1.predict(Test_data)
print(preds[0,:])
print(test_labels[0])
```

The output of this cell is:

```
[3.6157888e-17 0.0000000e+00 1.0000000e+00]
1.0
```

The right sidebar shows session information: Draft Session | 1m/9h | GPU On | CPU 0.19% | RAM 2.5GB/13GB | Disk 266.4MB/4.9GB. The workspace contains a folder "input (read-only data)" and a file "StarGalaxyClassification\_da...".

The bottom console shows: Draft Session | CPU 0% | RAM 2.5GB/13GB | GPU On | Disk 266.4MB/4.9GB