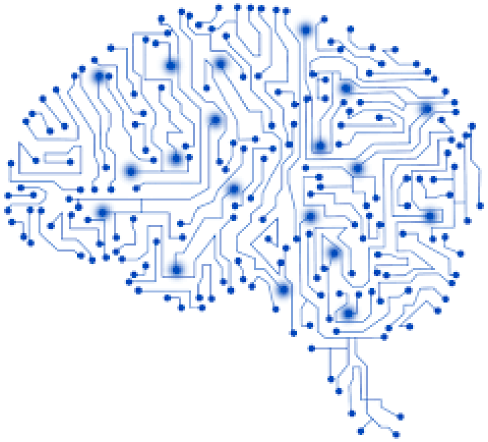
**University of Minho**
School of Engineering

ISLab



# Deep Learning with TensorFlow™

## Connective Systems and Classifiers

Perfil ML:FA @ MiEI/4º ano - 2º Semestre

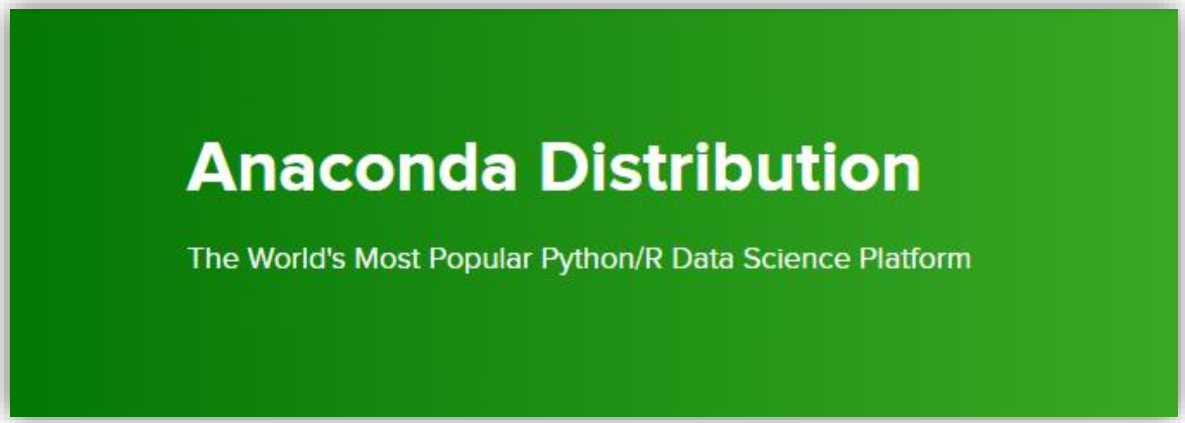
Bruno Fernandes, Victor Alves

13/02/2020

# Contents

- Anaconda
  - Welcome
  - How To
- TensorFlow
  - Introduction
  - 1.x vs 2.x
- ML Useful Libraries
- Using the Cloud
  - Google Colab
  - Kaggle Kernels
  - IBM Cognitive Class Labs
- Hands On

# ANACONDA®



## Anaconda Distribution

The World's Most Popular Python/R Data Science Platform

jupyter
spyder
NumPy
SciPy
Numba
pandas
DASK
Bokeh
HoloViews
Datashader
matplotlib
scikit learn
H2O.ai
TensorFlow
CONDA

# ANACONDA®

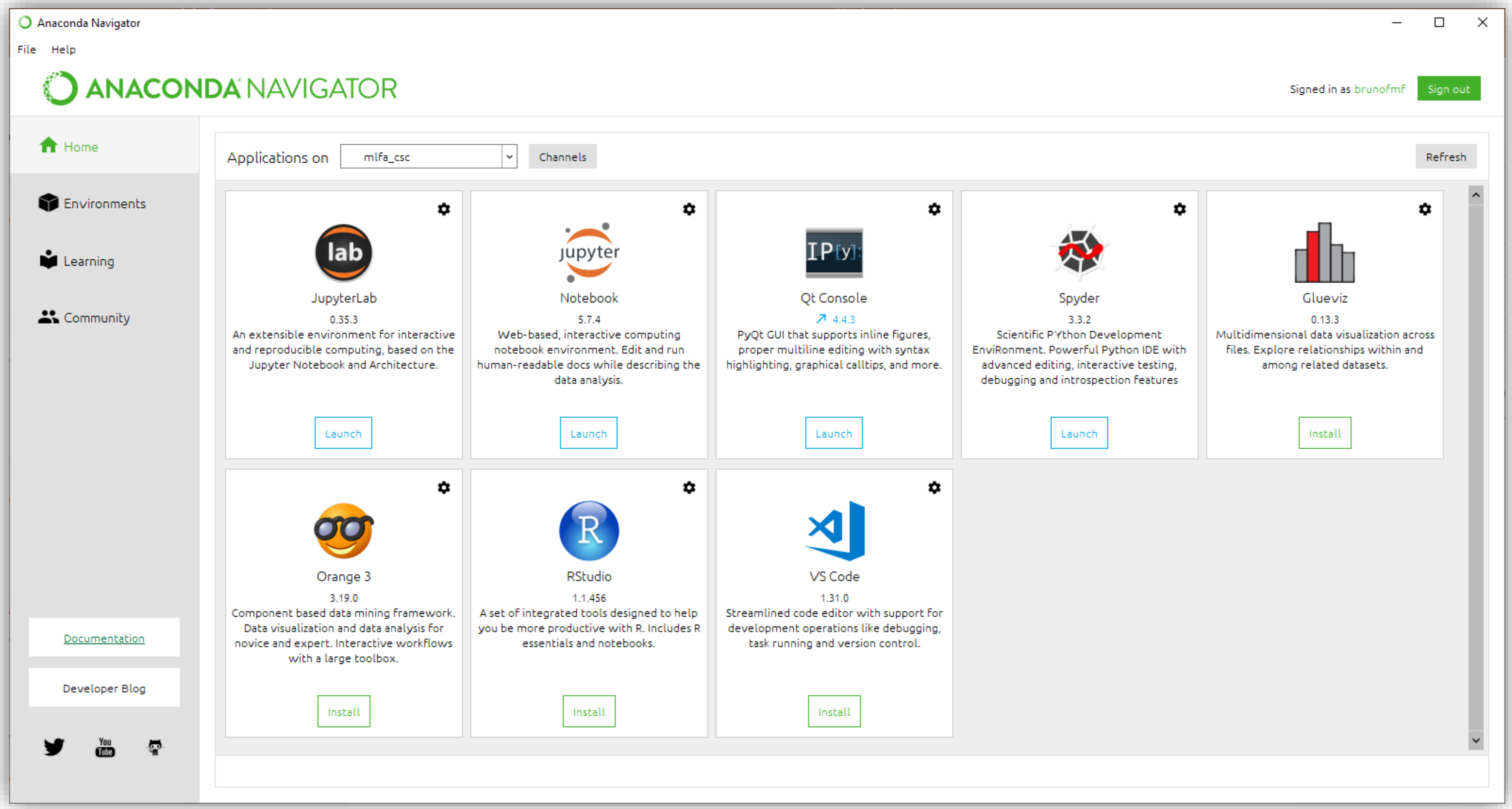
- FOSS
- Used for developing, testing and training ML models
- Share, collaborate on, and reproduce projects
- Highly supported by the community
- Conda, a package, dependency and environment manager
  - Easily create, save, load and switch between environments
    - ✓ An environment is a directory that contains a specific collection of packages that you have installed. For example, you may have one environment with TensorFlow 2.0 and another environment with TensorFlow 1.13 for legacy testing
  - Easily install, update and run any data science package (and its dependencies… automatically!)

Anaconda provides two user clients:

- Anaconda Navigator
- Anaconda Prompt (or the terminal on Linux and macOS)

Anaconda provides two user clients:

- Anaconda Navigator



JupyterLab
0.35.3
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook
5.7.4
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Qt Console
↗ 4.4.3
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Spyder
3.3.2
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Glueviz
0.13.3
Multidimensional data visualization across files. Explore relationships within and among related datasets.

Orange 3
3.19.0
Component based data mining Framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

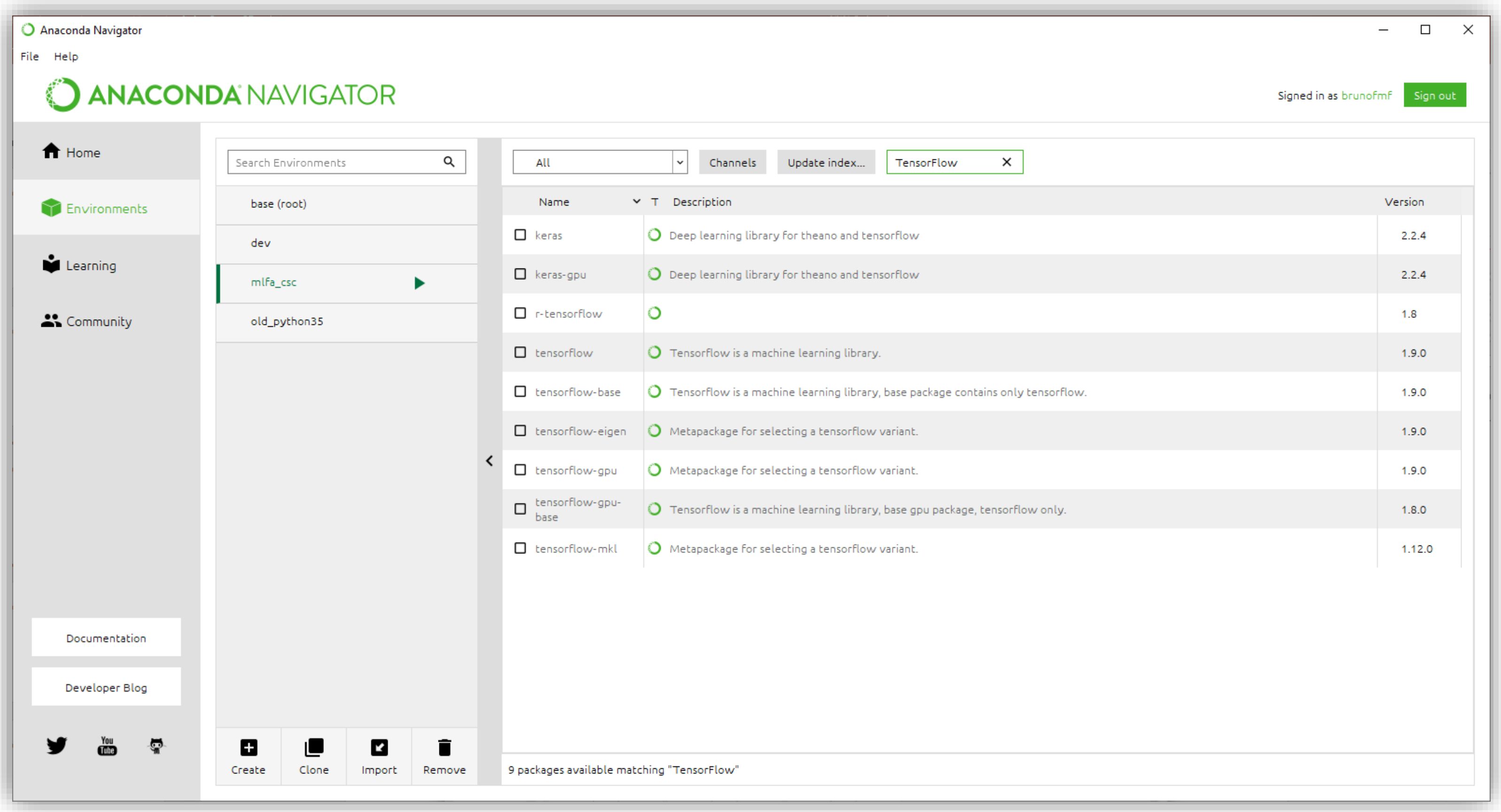
RStudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

VS Code
1.31.0
Streamlined code editor with support for development operations like debugging, task running and version control.

# ANACONDA

Anaconda provides two user clients:

- Anaconda Navigator

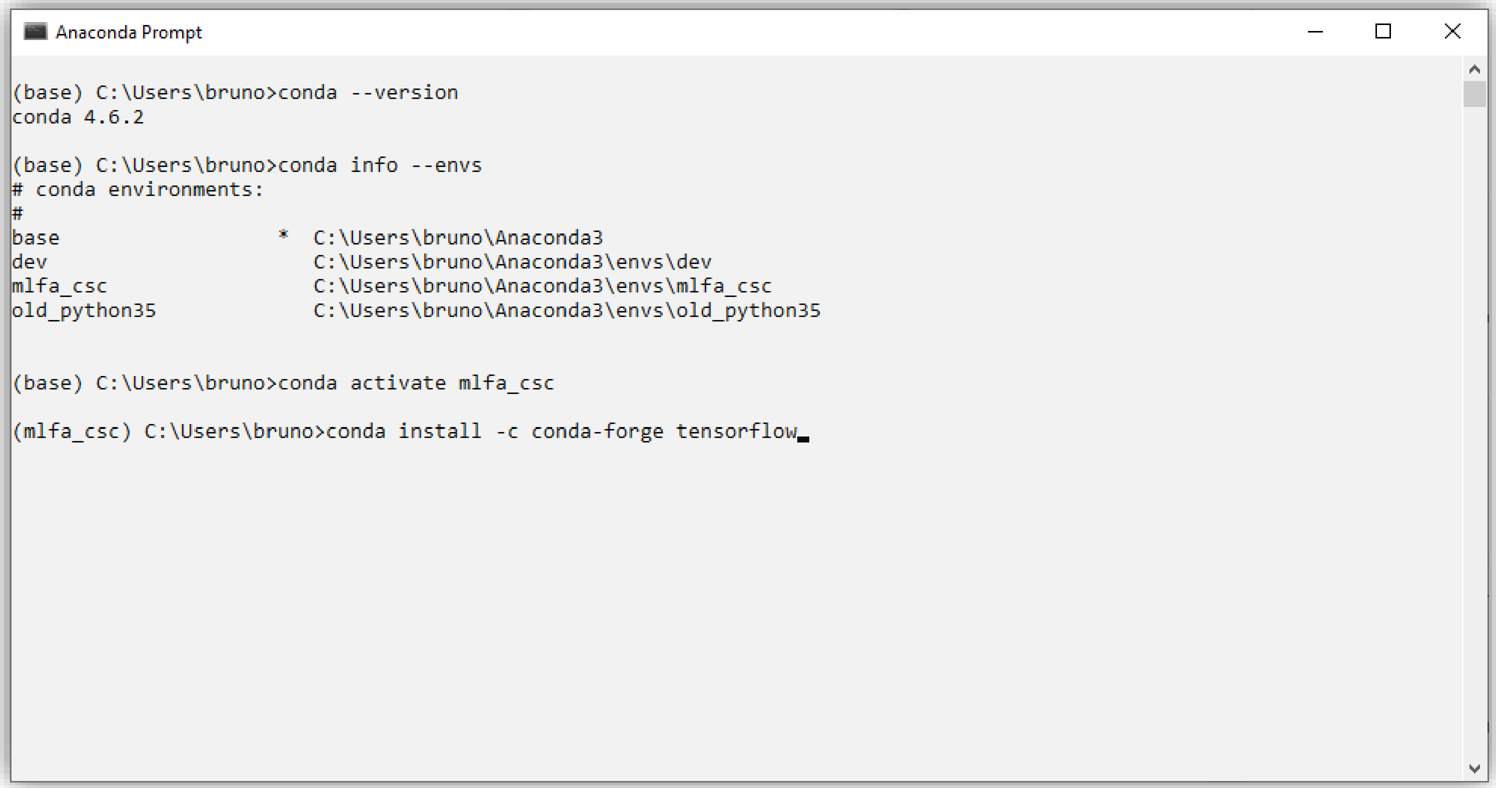


|  | Name | T | Description | Version |
|---|---|---|---|---|
| □ | keras | ○ | Deep learning library for theano and tensorflow | 2.2.4 |
| □ | keras-gpu | ○ | Deep learning library for theano and tensorflow | 2.2.4 |
| □ | r-tensorflow | ○ |  | 1.8 |
| □ | tensorflow | ○ | Tensorflow is a machine learning library. | 1.9.0 |
| □ | tensorflow-base | ○ | Tensorflow is a machine learning library, base package contains only tensorflow. | 1.9.0 |
| □ | tensorflow-eigen | ○ | Metapackage for selecting a tensorflow variant. | 1.9.0 |
| □ | tensorflow-gpu | ○ | Metapackage for selecting a tensorflow variant. | 1.9.0 |
| □ | tensorflow-gpu-base | ○ | Tensorflow is a machine learning library, base gpu package, tensorflow only. | 1.8.0 |
| □ | tensorflow-mkl | ○ | Metapackage for selecting a tensorflow variant. | 1.12.0 |

9 packages available matching "TensorFlow"

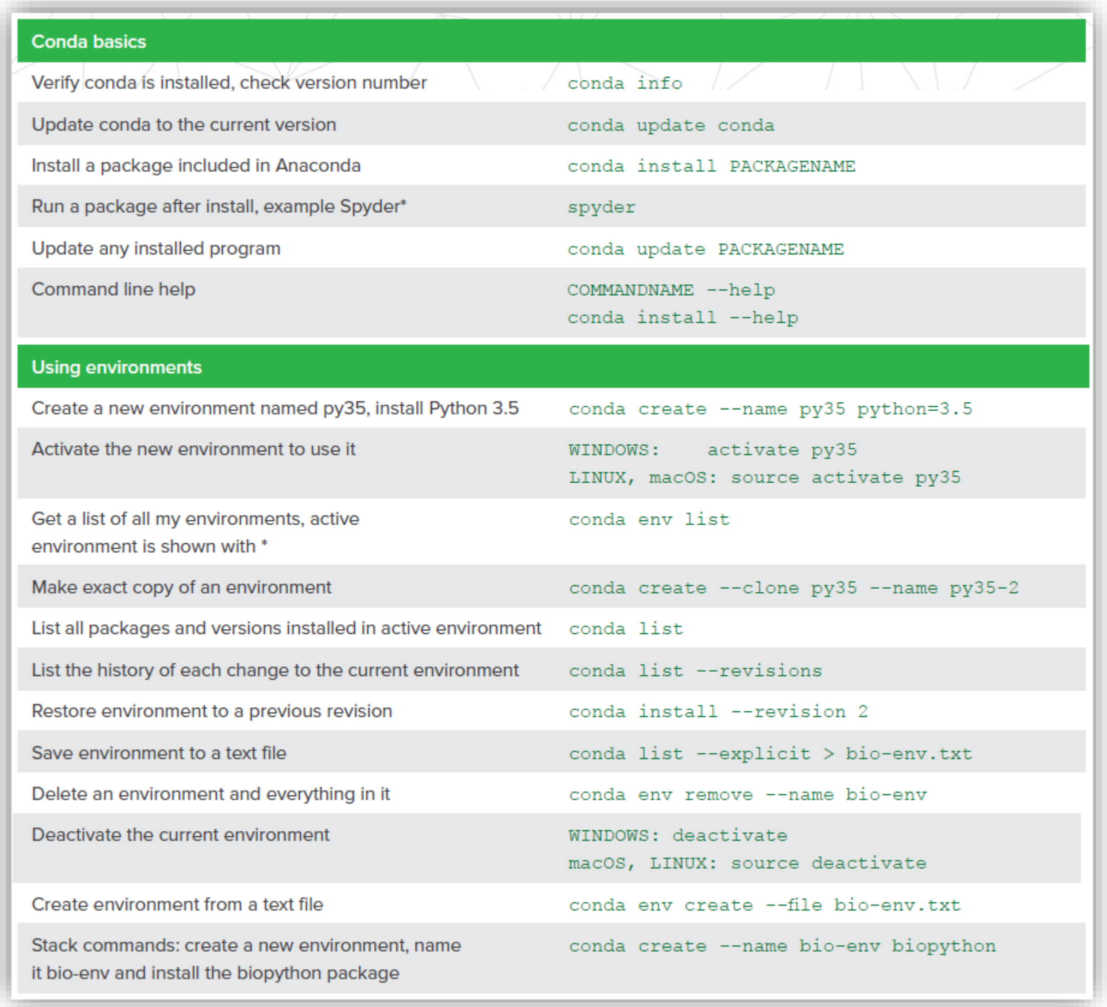
Anaconda provides two user clients:

- Anaconda Prompt (or Terminal on Linux and macOS)



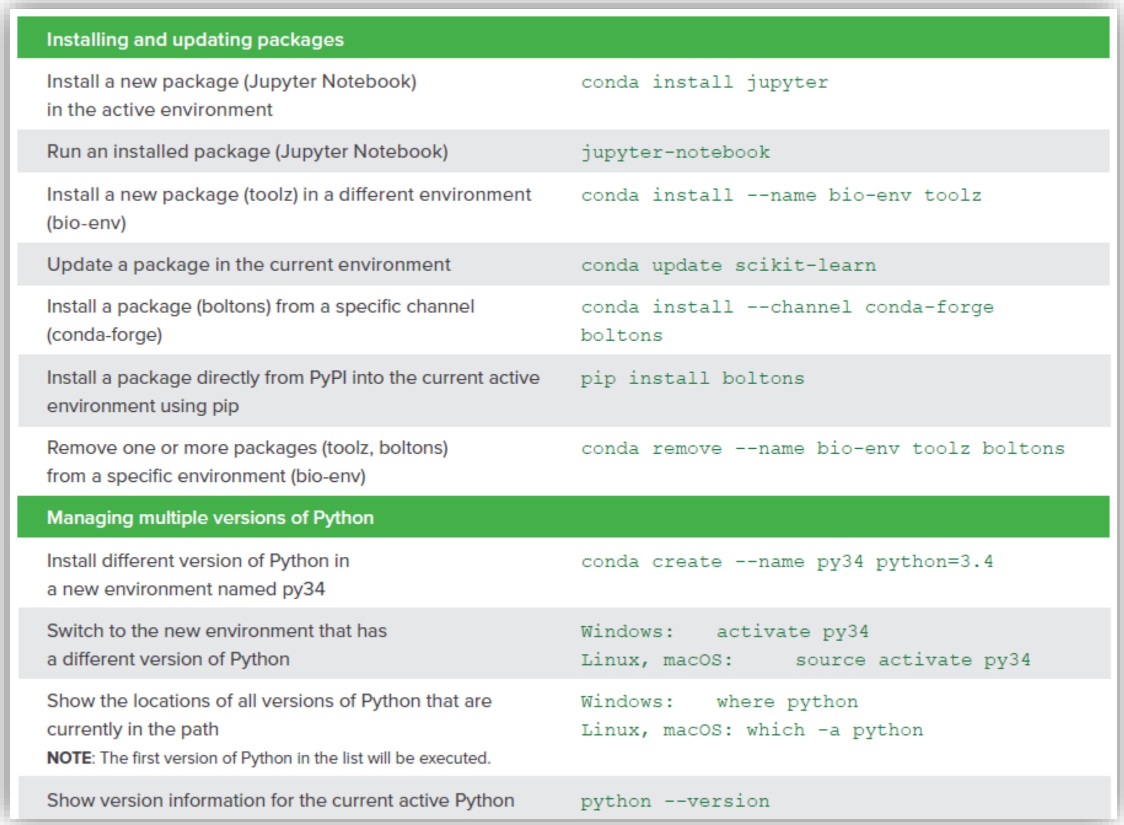
```
(base) C:\Users\bruno>conda --version
conda 4.6.2

(base) C:\Users\bruno>conda info --envs
# conda environments:
#
base                  *  C:\Users\bruno\Anaconda3
dev                      C:\Users\bruno\Anaconda3\envs\dev
mlfa_csc                 C:\Users\bruno\Anaconda3\envs\mlfa_csc
old_python35             C:\Users\bruno\Anaconda3\envs\old_python35


(base) C:\Users\bruno>conda activate mlfa_csc

(mlfa_csc) C:\Users\bruno>conda install -c conda-forge tensorflow
```

## Anaconda Prompt (or Terminal on Linux and macOS)



| Conda basics | |
|---|---|
| Verify conda is installed, check version number | `conda info` |
| Update conda to the current version | `conda update conda` |
| Install a package included in Anaconda | `conda install PACKAGENAME` |
| Run a package after install, example Spyder* | `spyder` |
| Update any installed program | `conda update PACKAGENAME` |
| Command line help | `COMMANDNAME --help`<br>`conda install --help` |
| **Using environments** | |
| Create a new environment named py35, install Python 3.5 | `conda create --name py35 python=3.5` |
| Activate the new environment to use it | `WINDOWS:     activate py35`<br>`LINUX, macOS: source activate py35` |
| Get a list of all my environments, active environment is shown with * | `conda env list` |
| Make exact copy of an environment | `conda create --clone py35 --name py35-2` |
| List all packages and versions installed in active environment | `conda list` |
| List the history of each change to the current environment | `conda list --revisions` |
| Restore environment to a previous revision | `conda install --revision 2` |
| Save environment to a text file | `conda list --explicit > bio-env.txt` |
| Delete an environment and everything in it | `conda env remove --name bio-env` |
| Deactivate the current environment | `WINDOWS: deactivate`<br>`macOS, LINUX: source deactivate` |
| Create environment from a text file | `conda env create --file bio-env.txt` |
| Stack commands: create a new environment, name it bio-env and install the biopython package | `conda create --name bio-env biopython` |

## Anaconda Prompt (or Terminal on Linux and macOS)



| | |
|---|---|
| **Installing and updating packages** | |
| Install a new package (Jupyter Notebook) in the active environment | `conda install jupyter` |
| Run an installed package (Jupyter Notebook) | `jupyter-notebook` |
| Install a new package (toolz) in a different environment (bio-env) | `conda install --name bio-env toolz` |
| Update a package in the current environment | `conda update scikit-learn` |
| Install a package (boltons) from a specific channel (conda-forge) | `conda install --channel conda-forge boltons` |
| Install a package directly from PyPI into the current active environment using pip | `pip install boltons` |
| Remove one or more packages (toolz, boltons) from a specific environment (bio-env) | `conda remove --name bio-env toolz boltons` |
| **Managing multiple versions of Python** | |
| Install different version of Python in a new environment named py34 | `conda create --name py34 python=3.4` |
| Switch to the new environment that has a different version of Python | Windows: `activate py34`<br>Linux, macOS: `source activate py34` |
| Show the locations of all versions of Python that are currently in the path<br>**NOTE**: The first version of Python in the list will be executed. | Windows: `where python`<br>Linux, macOS: `which -a python` |
| Show version information for the current active Python | `python --version` |

IDEs for all tastes:

- Spyder
  - Scientific PYthon Development EnviRonment
- Jupyter Notebooks
- PTVS
  - Python Tools for Visual Studio
- PyCharm
- PyDev
  - Python IDE for Eclipse
- …

# ANACONDA®

There are alternatives:

- Pip + Virtualenv
- + brew
- Miniconda (Anaconda may come with too much stuff…)
- …

# ANACONDA

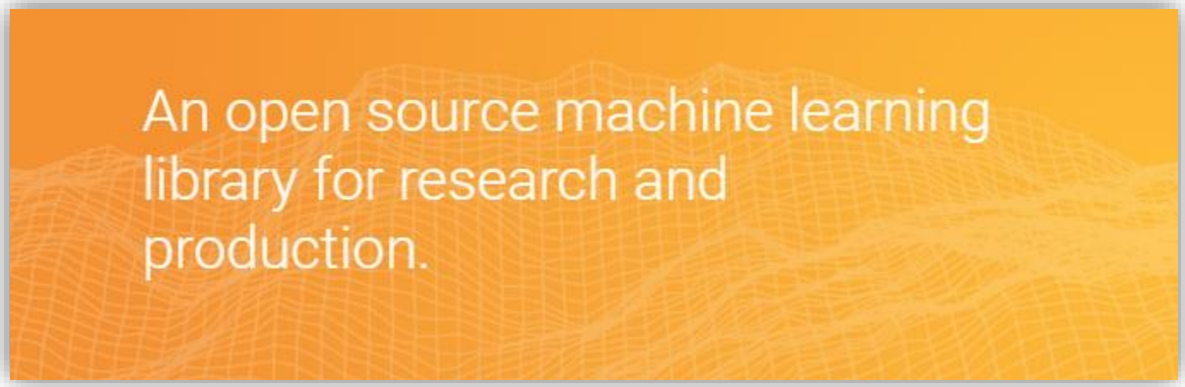
- Install Anaconda

- Try Conda and/or Anaconda Navigator
  - Create a new environment for this class using python v3.7
    ex.: conda create --name mlfa_csc_tf2 python=3.7
  - Install new packages such as tensorflow (v2.0), pandas, numpy or matplotlib
    ex.: conda activate mlfa_csc_tf2
    conda install tensorflow

- Install and try the IDEs
  - print("Hello World")

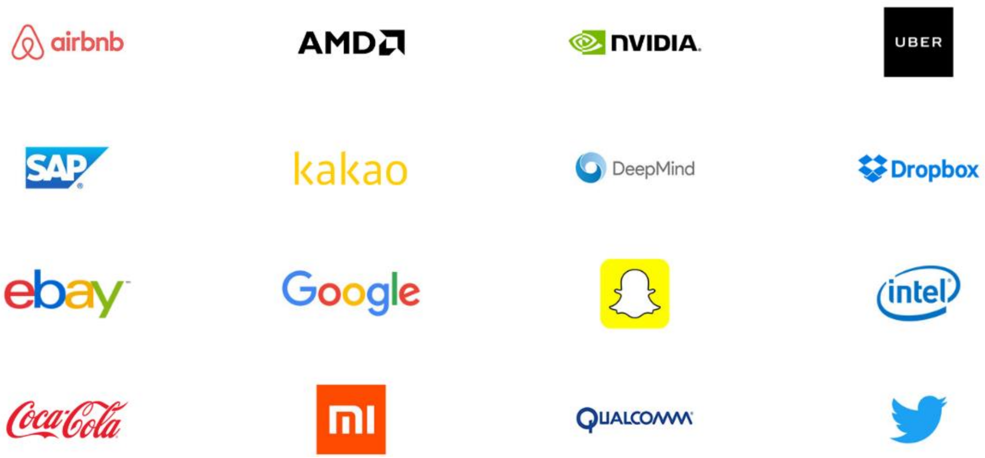
**HANDS ON**

# TensorFlow™



An open source machine learning library for research and production.

Companies using TensorFlow



# TensorFlow™

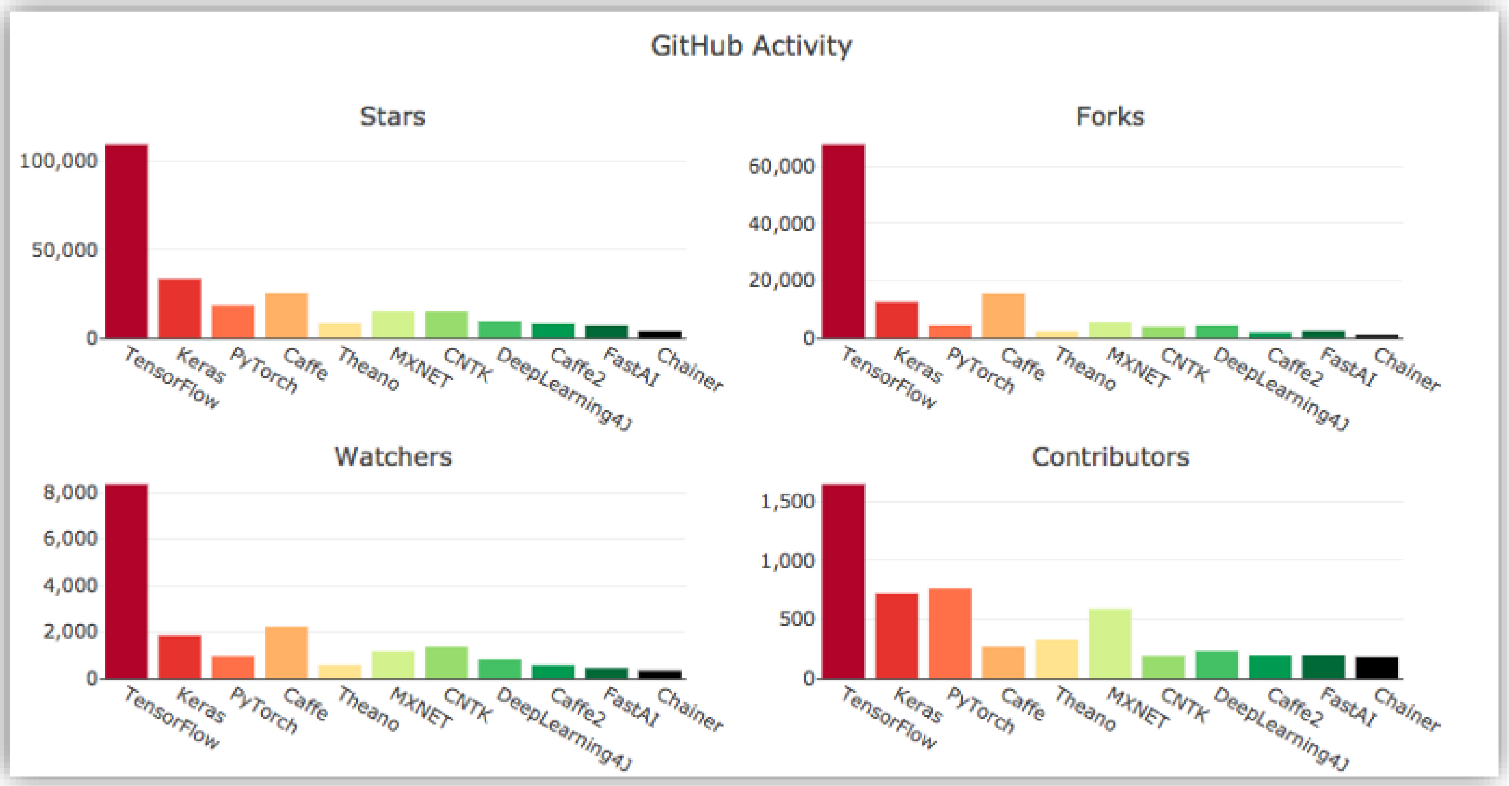
- Open source software library for high performance numerical computation
- Strong support for machine learning and deep learning
- It has seen tremendous growth and popularity in the machine learning community

# TensorFlow™

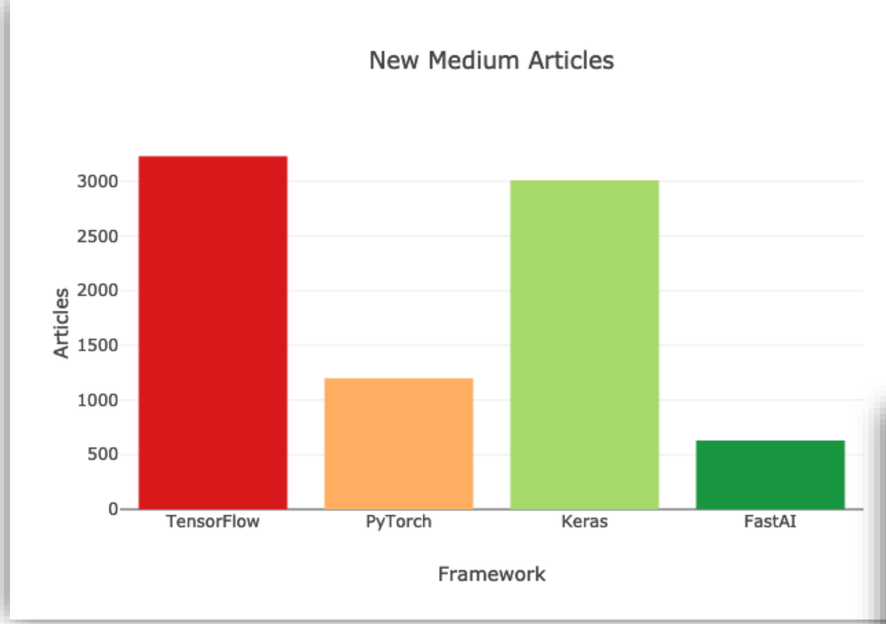
There are alternatives:

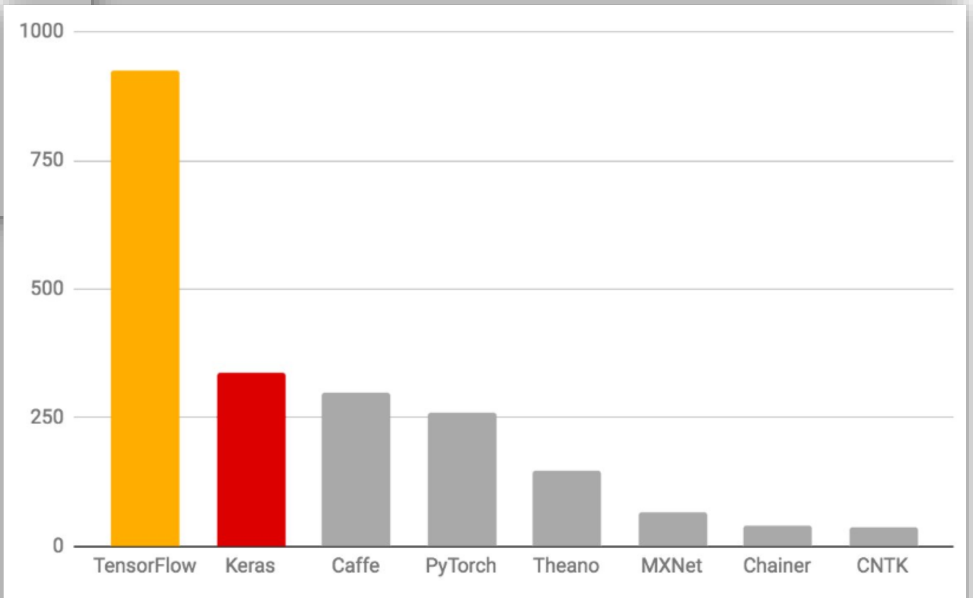
- PyTorch
- theano
- K Keras
- Caffe2
- …

TensorFlow™

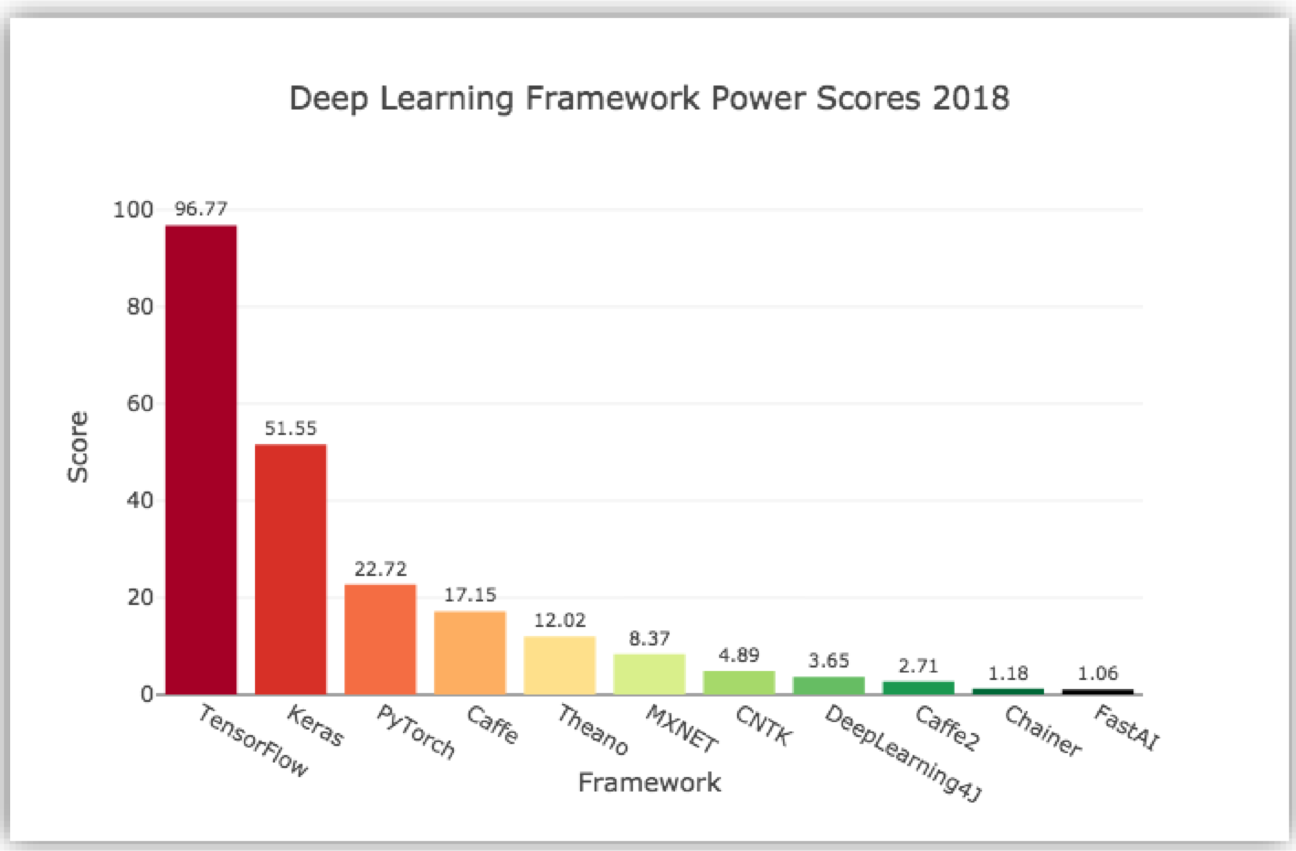

GitHub Activity
Stars
100,000
50,000
0
Forks
60,000
40,000
20,000
0
Watchers
8,000
6,000
4,000
2,000
0
Contributors
1,500
1,000
500
0
TensorFlow
Keras
PyTorch
Caffe
Theano
MXNET
CNTK
DeepLearning4J
Caffe2
FastAI
Chainer

# TensorFlow™

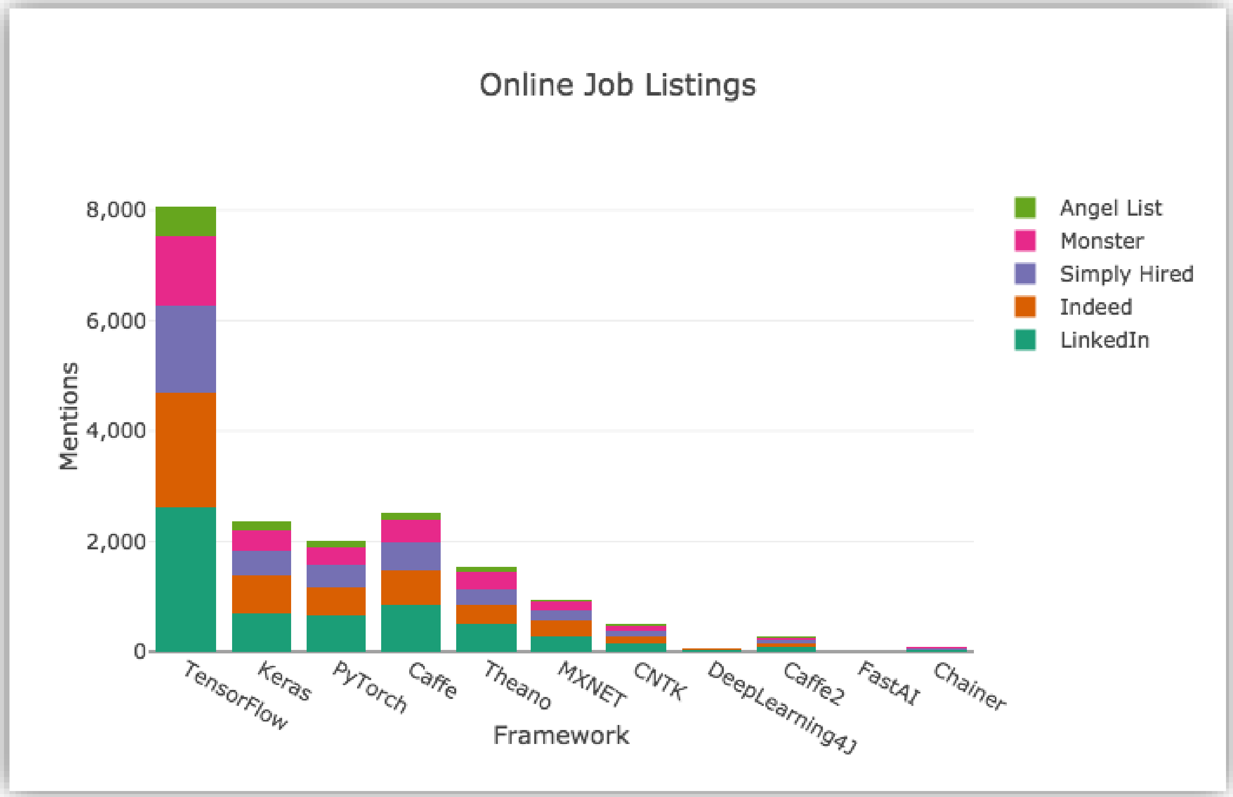




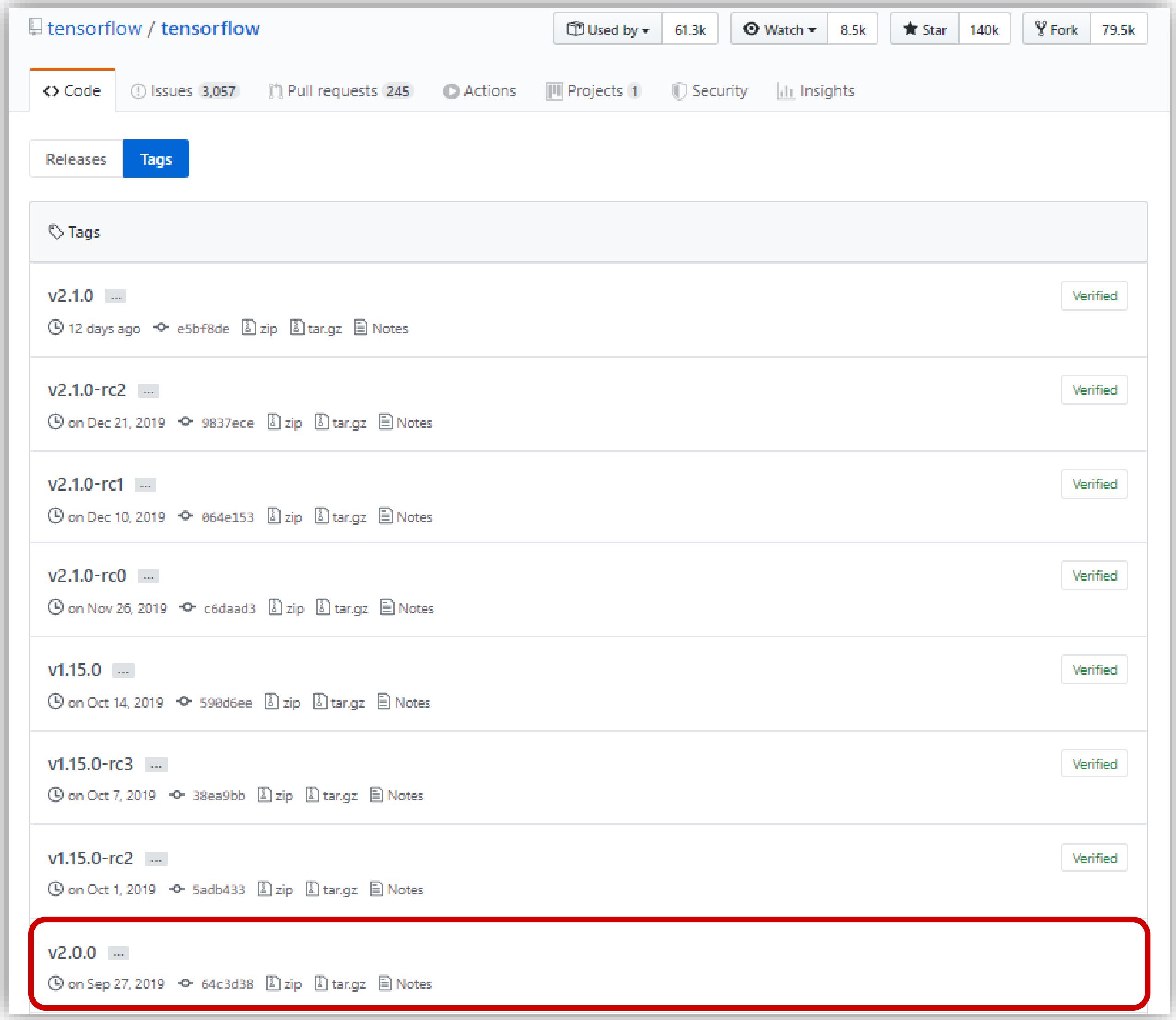
arXiv mentions as of 2018/03/07 (past 3 months)


Deep Learning Framework Power Scores 2018
Score
100
80
60
40
20
0
96.77
51.55
22.72
17.15
12.02
8.37
4.89
3.65
2.71
1.18
1.06
TensorFlow
Keras
PyTorch
Caffe
Theano
MXNET
CNTK
DeepLearning4J
Caffe2
Chainer
FastAI
Framework

# TensorFlow™


Online Job Listings
8,000
6,000
4,000
2,000
0
Mentions
TensorFlow
Keras
PyTorch
Caffe
Theano
MXNET
CNTK
DeepLearning4J
Caffe2
FastAI
Chainer
Framework
🟩 Angel List
🟥 Monster
🟪 Simply Hired
🟧 Indeed
🟩 LinkedIn

# TensorFlow ™



tensorflow / tensorflow

Used by 61.3k | Watch 8.5k | Star 140k | Fork 79.5k

<> Code | Issues 3,057 | Pull requests 245 | Actions | Projects 1 | Security | Insights

Releases | **Tags**

Tags

v2.1.0 … Verified
12 days ago · e5bf8de · zip · tar.gz · Notes

v2.1.0-rc2 … Verified
on Dec 21, 2019 · 9837ece · zip · tar.gz · Notes

v2.1.0-rc1 … Verified
on Dec 10, 2019 · 064e153 · zip · tar.gz · Notes

v2.1.0-rc0 … Verified
on Nov 26, 2019 · c6daad3 · zip · tar.gz · Notes

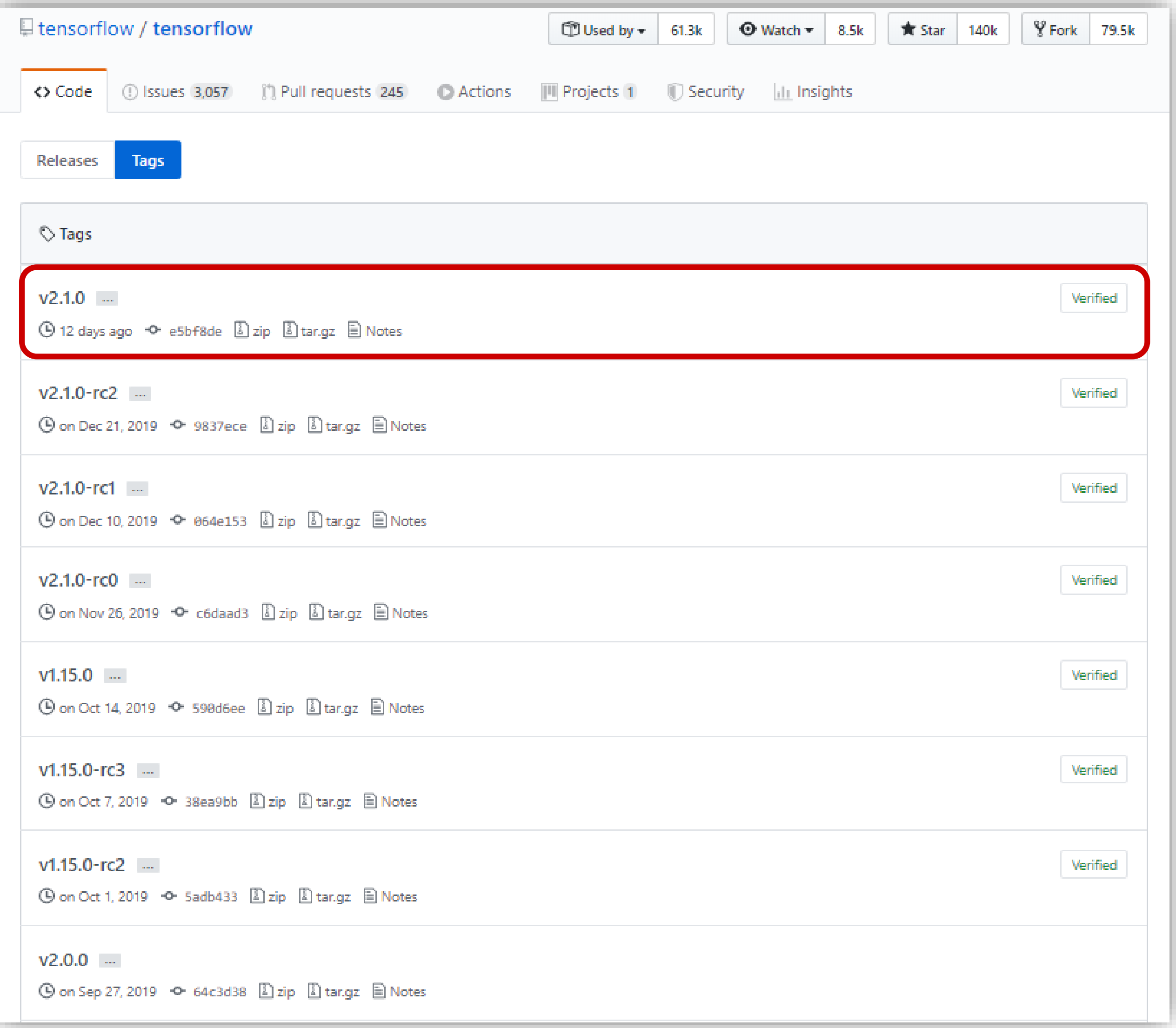
v1.15.0 … Verified
on Oct 14, 2019 · 590d6ee · zip · tar.gz · Notes

v1.15.0-rc3 … Verified
on Oct 7, 2019 · 38ea9bb · zip · tar.gz · Notes

v1.15.0-rc2 … Verified
on Oct 1, 2019 · 5adb433 · zip · tar.gz · Notes

v2.0.0 …
on Sep 27, 2019 · 64c3d38 · zip · tar.gz · Notes

TensorFlow ™



tensorflow / tensorflow

Used by 61.3k · Watch 8.5k · Star 140k · Fork 79.5k

<> Code · Issues 3,057 · Pull requests 245 · Actions · Projects 1 · Security · Insights

Releases · **Tags**

Tags

v2.1.0 … Verified
12 days ago · e5bf8de · zip · tar.gz · Notes

v2.1.0-rc2 … Verified
on Dec 21, 2019 · 9837ece · zip · tar.gz · Notes

v2.1.0-rc1 … Verified
on Dec 10, 2019 · 064e153 · zip · tar.gz · Notes

v2.1.0-rc0 … Verified
on Nov 26, 2019 · c6daad3 · zip · tar.gz · Notes

v1.15.0 … Verified
on Oct 14, 2019 · 590d6ee · zip · tar.gz · Notes

v1.15.0-rc3 … Verified
on Oct 7, 2019 · 38ea9bb · zip · tar.gz · Notes

v1.15.0-rc2 … Verified
on Oct 1, 2019 · 5adb433 · zip · tar.gz · Notes

v2.0.0 …
on Sep 27, 2019 · 64c3d38 · zip · tar.gz · Notes

There are multiple changes in TensorFlow 2.x:

1. API Cleanup
2. Redundant APIs removed
3. APIs more consistent (Unified RNNs, Unified Optimizers)
4. Functions, not sessions (tf.function decorator)
5. Easy model building with Keras and eager execution
6. …

More details here:

https://github.com/tensorflow/tensorflow/releases/tag/v2.0.0

And here:

https://www.tensorflow.org/guide/effective_tf2

# A few more libraries…

- NumPy
  - A universal data structure that enables data analysis in numerical computing by allowing the exchange of data between algorithms
  - Contains, among others, a powerful N-dimensional array object

- pandas
  - An open source library providing high-performance, easy-to-use data structures and data analysis tools
  - A game changer when it comes to analyzing data with Python

# A few more libraries…

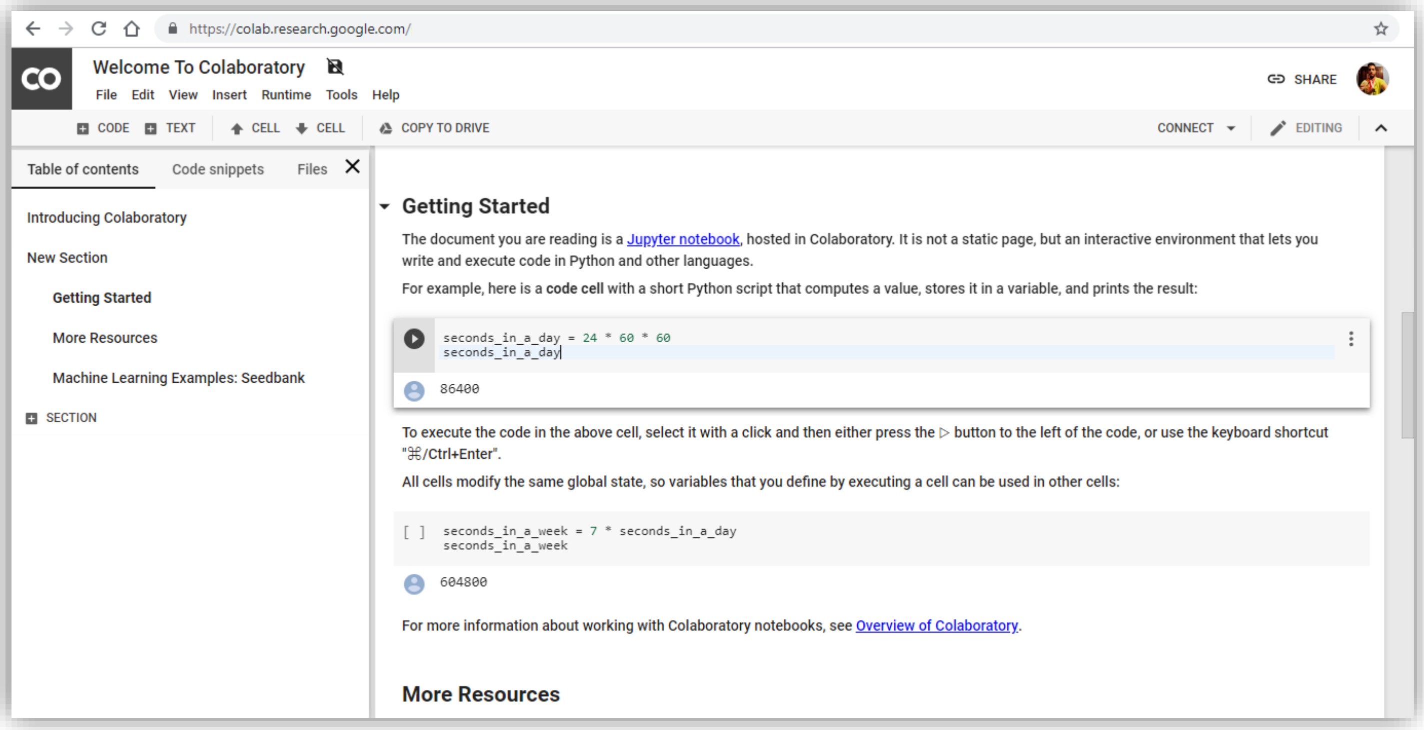
- Matplotlib
  - A 2D plotting library which produces publication-quality figures in a variety of hardcopy formats and interactive environments
  - A flexible and customizable tool for producing static and interactive data visualizations
- Scikit Learn
  - A free machine learning library for the Python programming language
  - Simple and efficient tools for data mining and data analysis
  - Features several classification, regression and clustering algorithms
  - Built on NumPy, SciPy, and matplotlib

# Using the Cloud

# Using the Cloud

Google’s platform for Machine Learning Research!

- Sync with GitHub and Google Drive
- Free GPUs and TPUs
- Tesla T4 GPUs (16GB) with a 12-hour limit for continuous assignment of a particular VM


https://colab.research.google.com/
CO
Welcome To Colaboratory
File Edit View Insert Runtime Tools Help
SHARE
CODE TEXT CELL CELL COPY TO DRIVE
CONNECT
EDITING
Table of contents Code snippets Files
Introducing Colaboratory
New Section
Getting Started
More Resources
Machine Learning Examples: Seedbank
SECTION

**Getting Started**

The document you are reading is a Jupyter notebook, hosted in Colaboratory. It is not a static page, but an interactive environment that lets you write and execute code in Python and other languages.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

```
seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
```

```
86400
```

To execute the code in the above cell, select it with a click and then either press the ▷ button to the left of the code, or use the keyboard shortcut "⌘/Ctrl+Enter".

All cells modify the same global state, so variables that you define by executing a cell can be used in other cells:

```
[ ]  seconds_in_a_week = 7 * seconds_in_a_day
     seconds_in_a_week
```

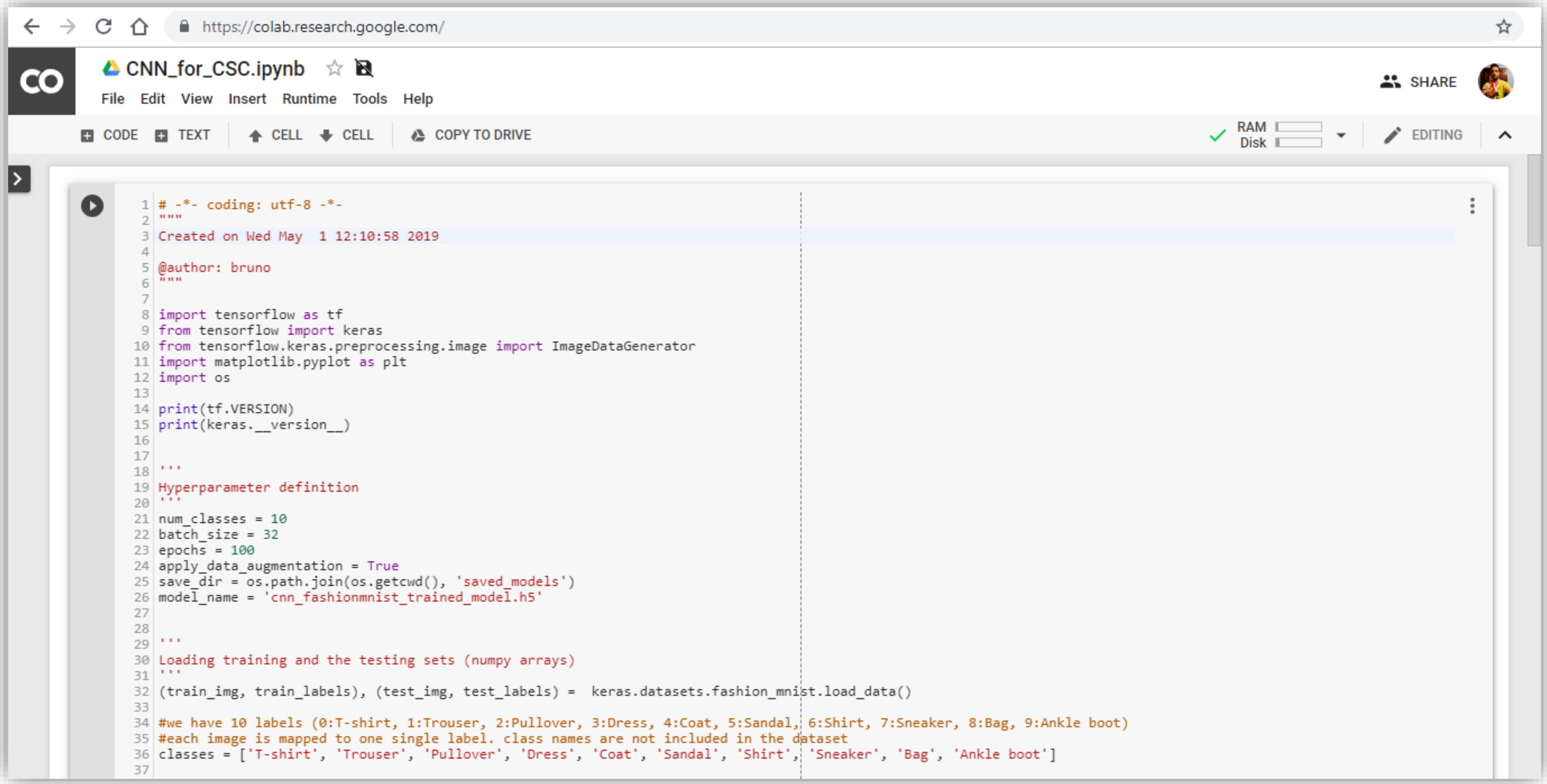
```
604800
```

For more information about working with Colaboratory notebooks, see Overview of Colaboratory.

**More Resources**

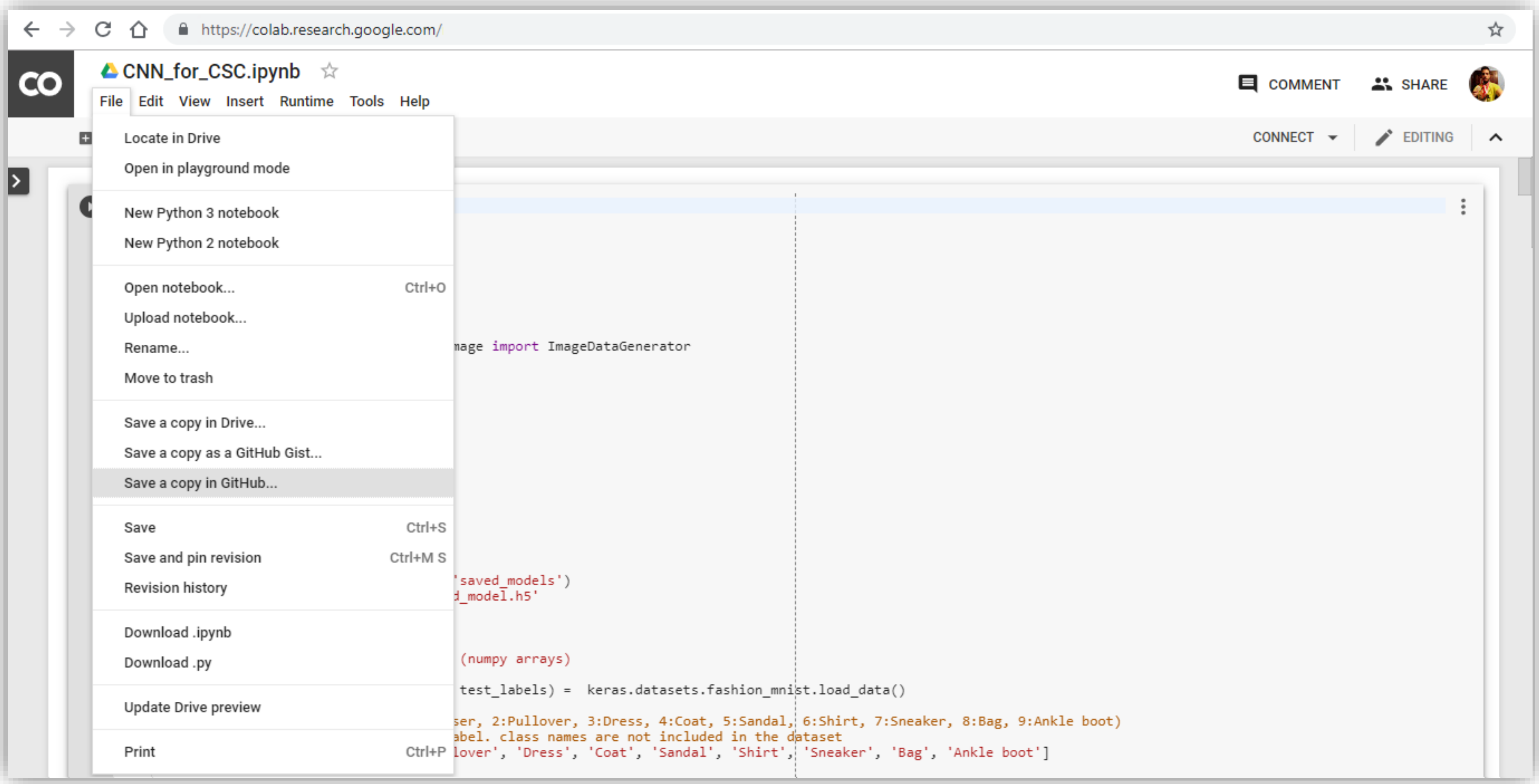
(https://colab.research.google.com/notebooks/welcome.ipynb)

# Using the Cloud



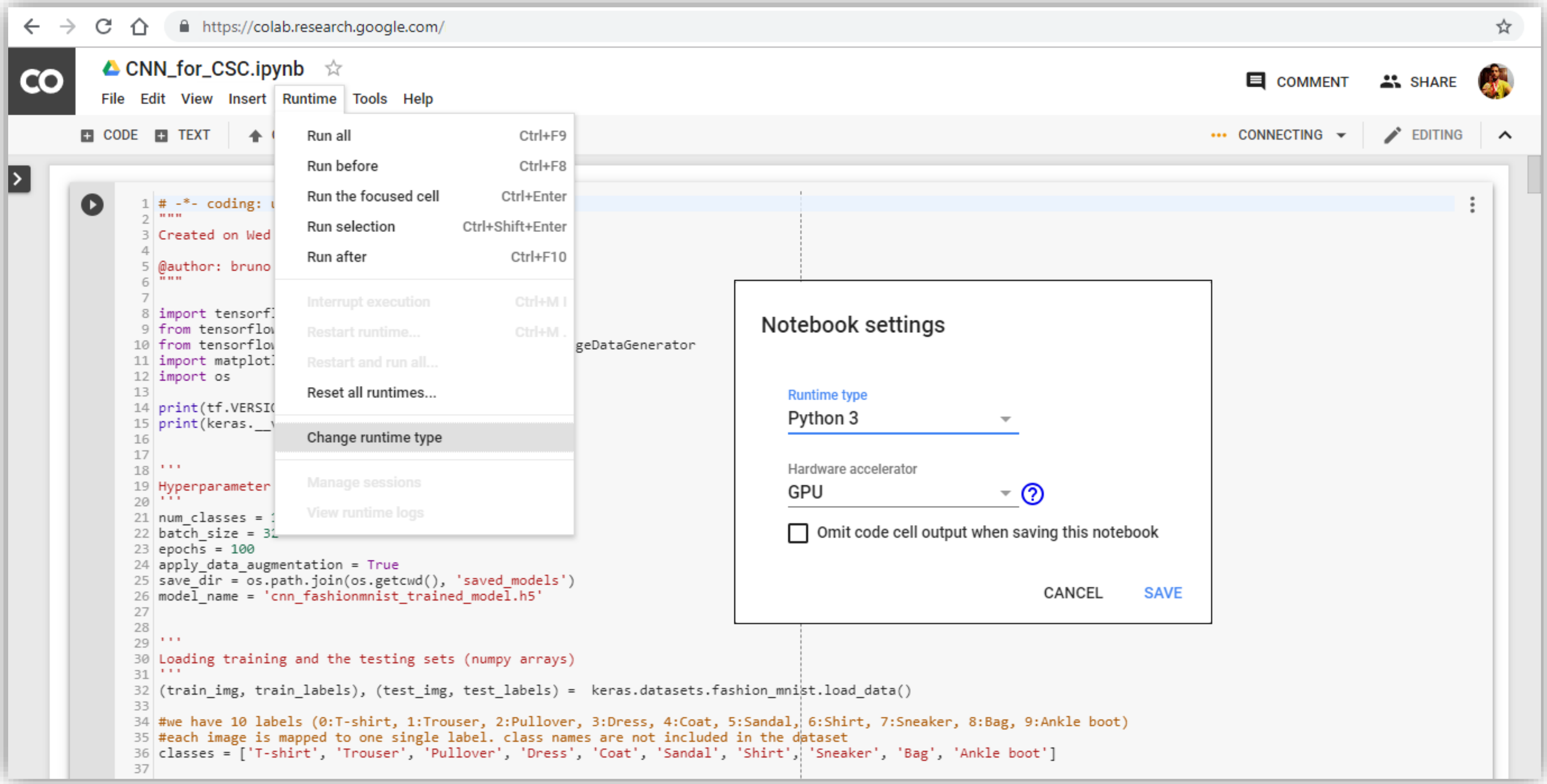
```
# -*- coding: utf-8 -*-
"""
Created on Wed May  1 12:10:58 2019

@author: bruno
"""

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt
import os

print(tf.VERSION)
print(keras.__version__)


'''
Hyperparameter definition
'''
num_classes = 10
batch_size = 32
epochs = 100
apply_data_augmentation = True
save_dir = os.path.join(os.getcwd(), 'saved_models')
model_name = 'cnn_fashionmnist_trained_model.h5'


'''
Loading training and the testing sets (numpy arrays)
'''
(train_img, train_labels), (test_img, test_labels) =  keras.datasets.fashion_mnist.load_data()

#we have 10 labels (0:T-shirt, 1:Trouser, 2:Pullover, 3:Dress, 4:Coat, 5:Sandal, 6:Shirt, 7:Sneaker, 8:Bag, 9:Ankle boot)
#each image is mapped to one single label. class names are not included in the dataset
classes = ['T-shirt', 'Trouser', 'Pullover', 'Dress', 'Coat', 'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle boot']
```

# Using the Cloud



```
mage import ImageDataGenerator
'saved_models')
d_model.h5'
(numpy arrays)
test_labels) =  keras.datasets.fashion_mnist.load_data()
ser, 2:Pullover, 3:Dress, 4:Coat, 5:Sandal, 6:Shirt, 7:Sneaker, 8:Bag, 9:Ankle boot)
abel. class names are not included in the dataset
lover', 'Dress', 'Coat', 'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle boot']
```

# Using the Cloud



```
# -*- coding: u
"""
Created on Wed

@author: bruno
"""

import tensorf
from tensorflo
from tensorflo                                      geDataGenerator
import matplot
import os

print(tf.VERSI
print(keras.__v


'''
Hyperparameter
'''
num_classes = 1
batch_size = 32
epochs = 100
apply_data_augmentation = True
save_dir = os.path.join(os.getcwd(), 'saved_models')
model_name = 'cnn_fashionmnist_trained_model.h5'


'''
Loading training and the testing sets (numpy arrays)
'''
(train_img, train_labels), (test_img, test_labels) =  keras.datasets.fashion_mnist.load_data()

#we have 10 labels (0:T-shirt, 1:Trouser, 2:Pullover, 3:Dress, 4:Coat, 5:Sandal, 6:Shirt, 7:Sneaker, 8:Bag, 9:Ankle boot)
#each image is mapped to one single label. class names are not included in the dataset
classes = ['T-shirt', 'Trouser', 'Pullover', 'Dress', 'Coat', 'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle boot']
```

Notebook settings

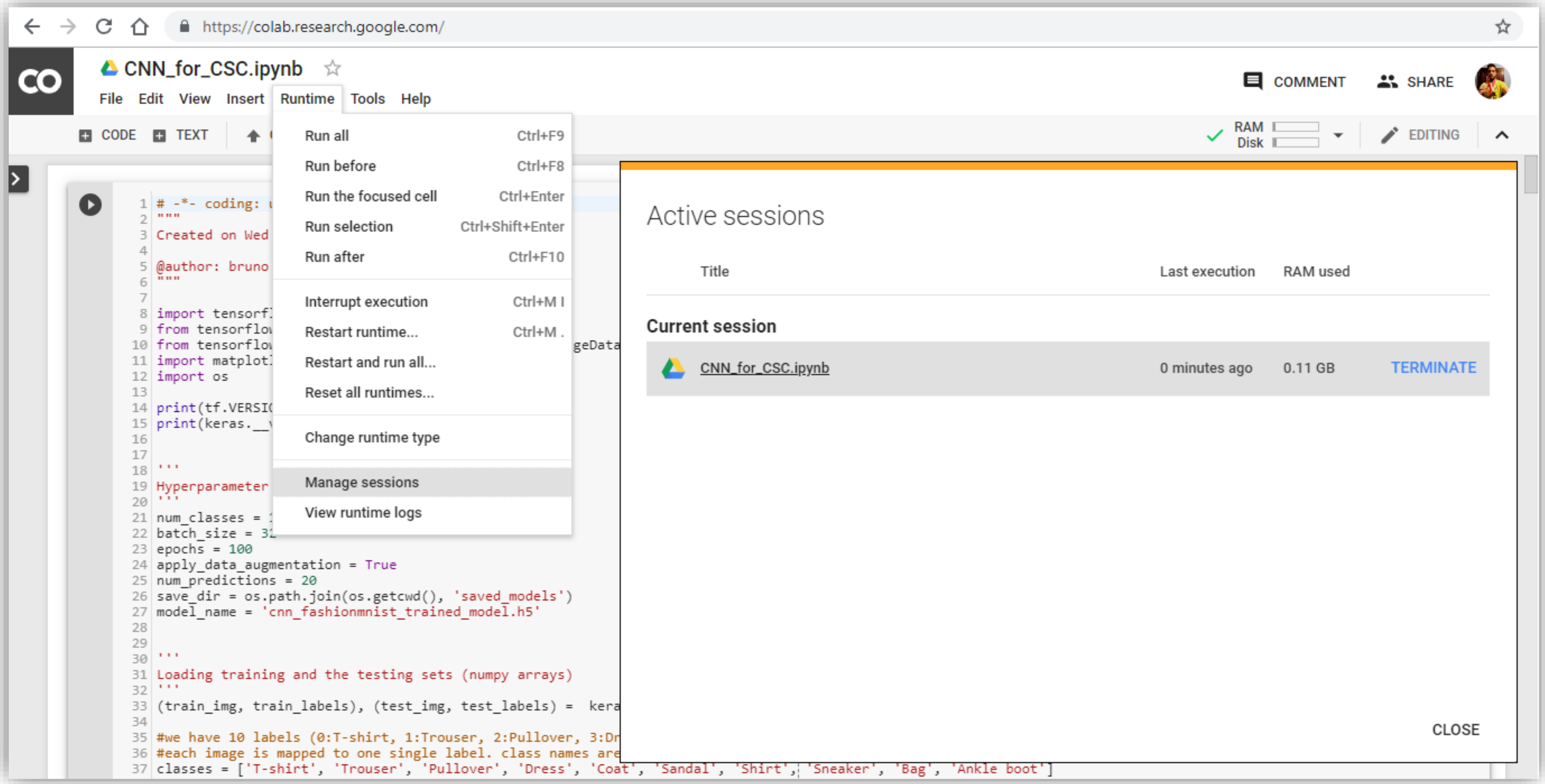
Runtime type: Python 3

Hardware accelerator: GPU

□ Omit code cell output when saving this notebook

CANCEL SAVE

# Using the Cloud



```
# -*- coding: u
"""
Created on Wed

@author: bruno
"""

import tensorfl
from tensorflo
from tensorflo                       geData
import matplot
import os

print(tf.VERSIO
print(keras.__v


'''
Hyperparameter
'''
num_classes = 1
batch_size = 32
epochs = 100
apply_data_augmentation = True
num_predictions = 20
save_dir = os.path.join(os.getcwd(), 'saved_models')
model_name = 'cnn_fashionmnist_trained_model.h5'


'''
Loading training and the testing sets (numpy arrays)
'''
(train_img, train_labels), (test_img, test_labels) =  kera

#we have 10 labels (0:T-shirt, 1:Trouser, 2:Pullover, 3:Dr
#each image is mapped to one single label. class names are
classes = ['T-shirt', 'Trouser', 'Pullover', 'Dress', 'Coat', 'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle boot']
```

Active sessions

Current session

| Title | Last execution | RAM used | |
|---|---|---|---|
| CNN_for_CSC.ipynb | 0 minutes ago | 0.11 GB | TERMINATE |

# Using the Cloud

Some last tips:

- Accessing the local file system and Google Drive
  - https://colab.research.google.com/notebooks/io.ipynb
- Using Google Colab with GitHub
  - https://colab.research.google.com/github/googlecolab/colabtools/blob/master/notebooks/colab-github-demo.ipynb
- Importing libraries and Upgrading TensorFlow
  - https://colab.research.google.com/notebooks/snippets/importing_libraries.ipynb
- Using GPUs and TPUs
  - https://colab.research.google.com/notebooks/gpu.ipynb
  - https://colab.research.google.com/notebooks/tpu.ipynb

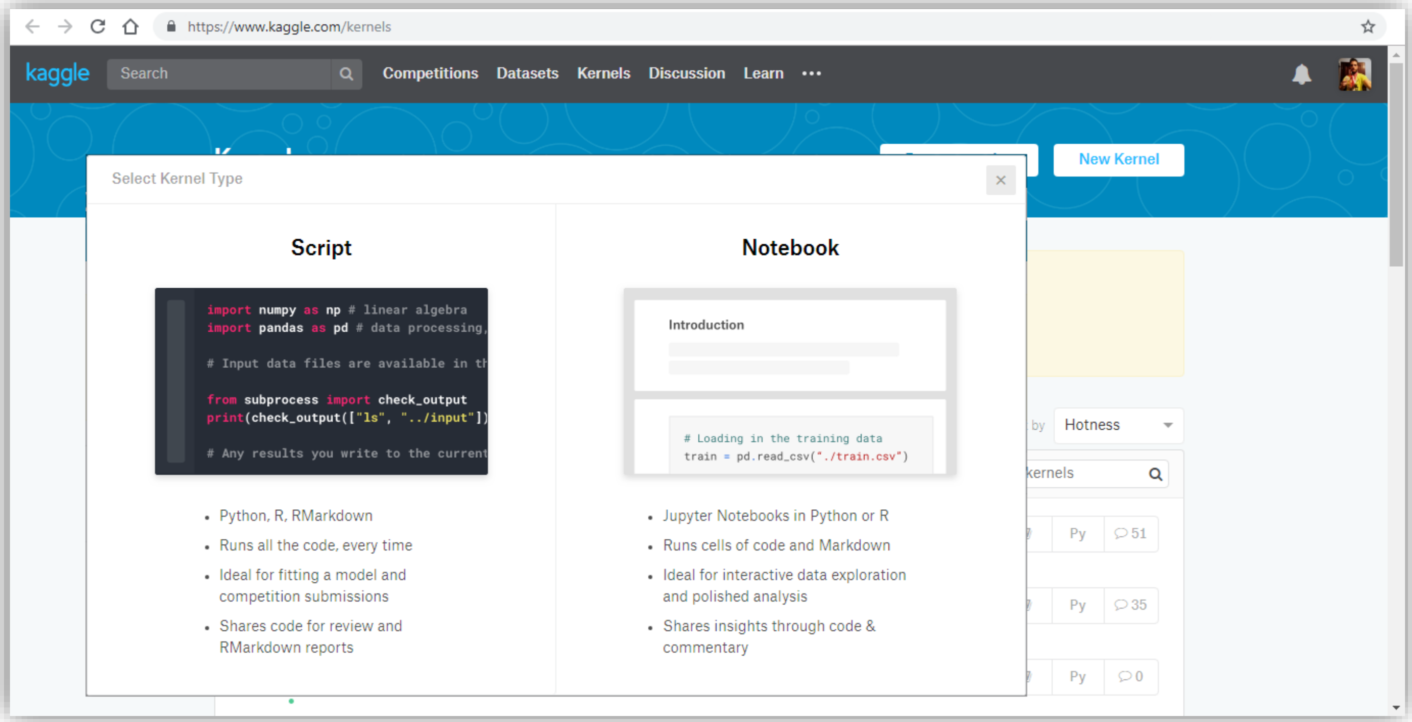
(https://research.google.com/colaboratory/faq.html)

# Using the Cloud

kaggle

Free - A platform for doing and sharing data science!

- Allows you to execute script-based code or Jupyter Notebooks - your choice!
- Allows for collaboration on Kernels!
- 9 hours of execution time, 4 core CPU with 17 GB of RAM and 2 core GPU with 14GB of RAM



Script

```
import numpy as np # linear algebra
import pandas as pd # data processing,

# Input data files are available in th

from subprocess import check_output
print(check_output(["ls", "../input"])

# Any results you write to the current
```

- Python, R, RMarkdown
- Runs all the code, every time
- Ideal for fitting a model and competition submissions
- Shares code for review and RMarkdown reports

Notebook

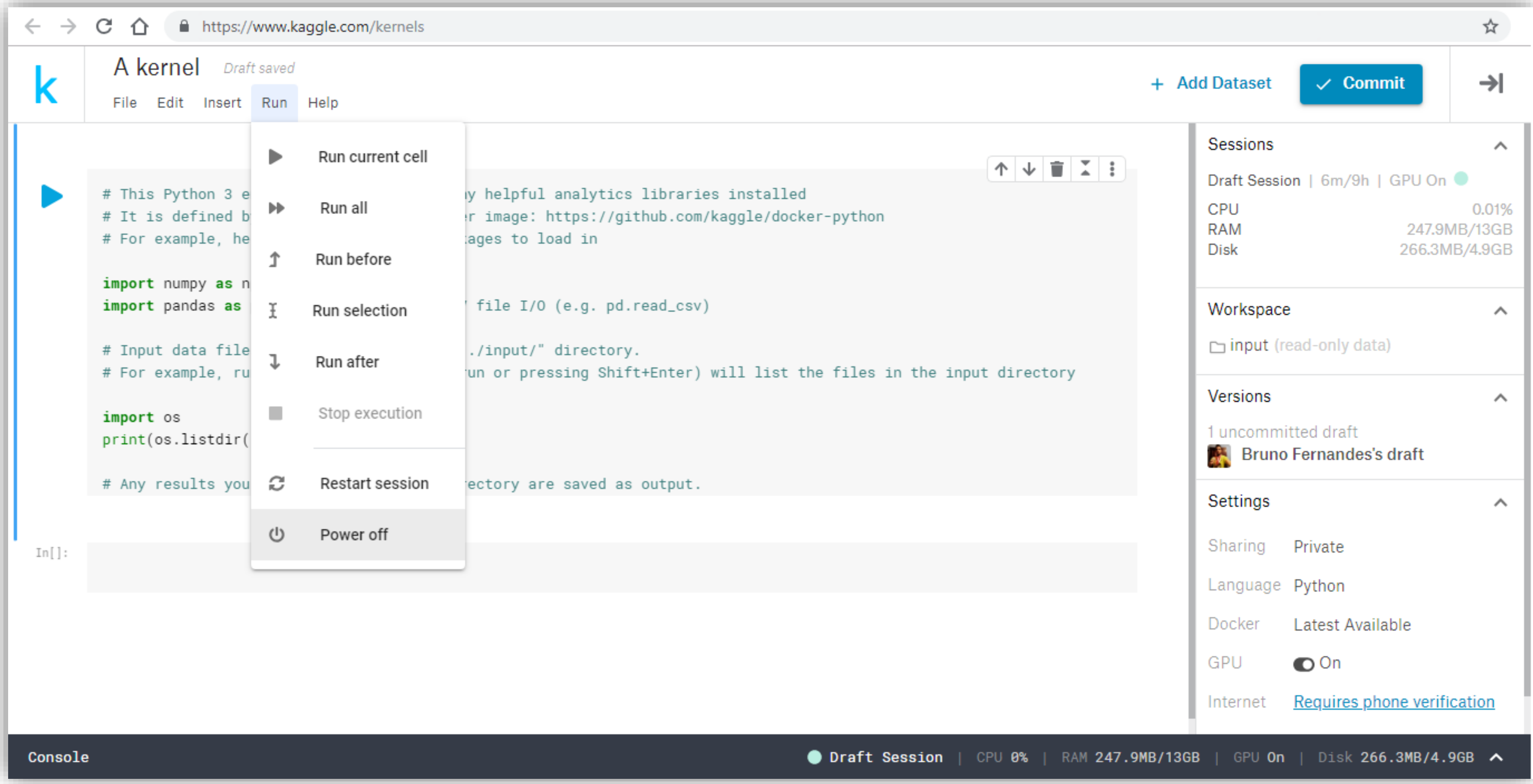
Introduction

```
# Loading in the training data
train = pd.read_csv("./train.csv")
```

- Jupyter Notebooks in Python or R
- Runs cells of code and Markdown
- Ideal for interactive data exploration and polished analysis
- Shares insights through code & commentary

(https://www.kaggle.com/kernels/welcome)

# Using the Cloud



https://www.kaggle.com/kernels

A kernel Draft saved

```
# This Python 3 e                       y helpful analytics libraries installed
# It is defined b                      r image: https://github.com/kaggle/docker-python
# For example, he                      ages to load in

import numpy as n
import pandas as                       file I/O (e.g. pd.read_csv)

# Input data file                      ./input/" directory.
# For example, ru                      un or pressing Shift+Enter) will list the files in the input directory

import os
print(os.listdir(

# Any results you                      ectory are saved as output.
```

Sessions

Draft Session | 6m/9h | GPU On

| | |
|---|---|
| CPU | 0.01% |
| RAM | 247.9MB/13GB |
| Disk | 266.3MB/4.9GB |

Workspace

input (read-only data)

Versions

1 uncommitted draft

Bruno Fernandes's draft

Settings

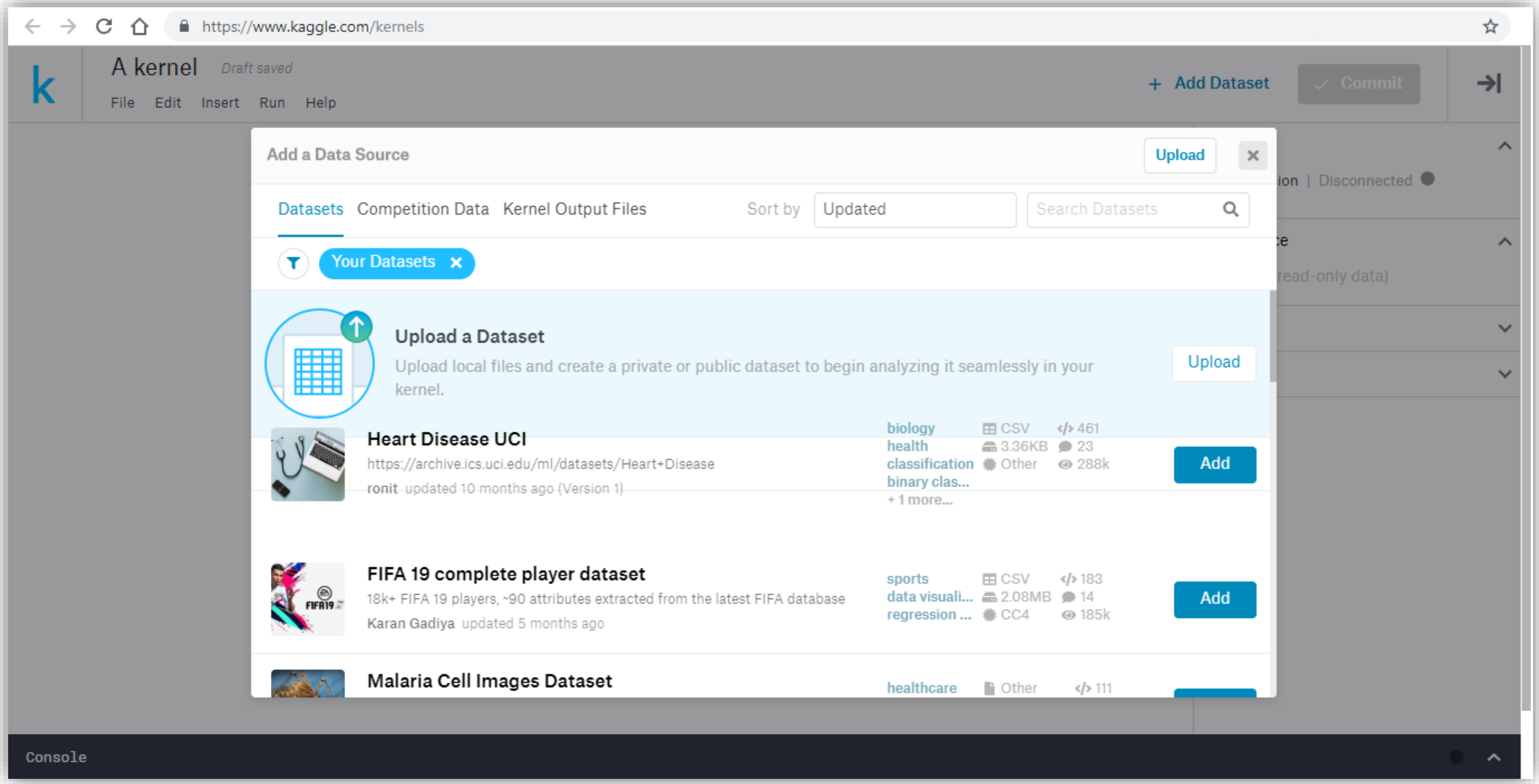
| | |
|---|---|
| Sharing | Private |
| Language | Python |
| Docker | Latest Available |
| GPU | On |
| Internet | Requires phone verification |

Console ● Draft Session | CPU 0% | RAM 247.9MB/13GB | GPU On | Disk 266.3MB/4.9GB

(https://www.kaggle.com/docs/kernels)

# Using the Cloud

kaggle



**Upload a Dataset**

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

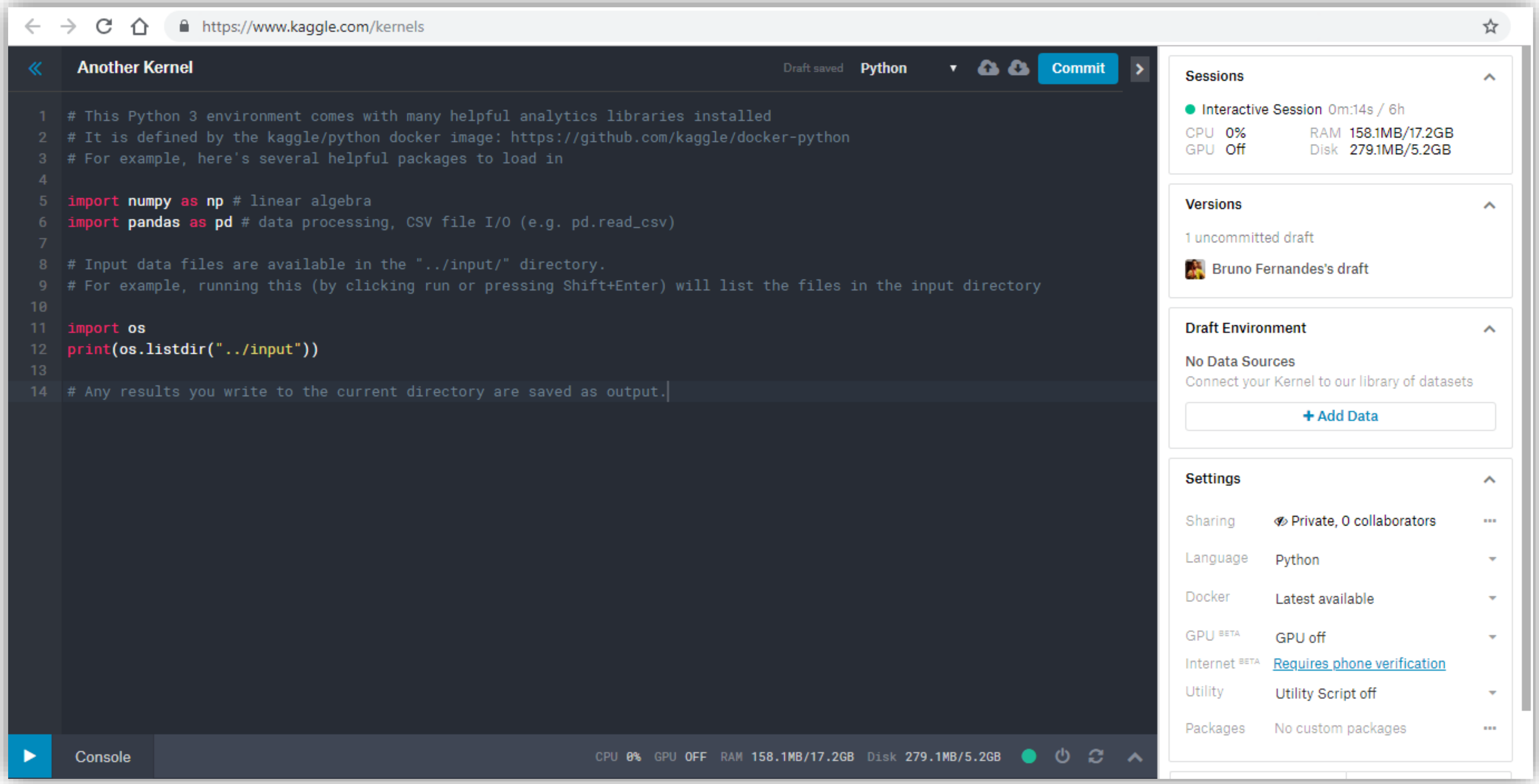
**Heart Disease UCI**
https://archive.ics.uci.edu/ml/datasets/Heart+Disease
ronit updated 10 months ago (Version 1)
biology, health, classification, binary clas..., + 1 more... | CSV, 3.36KB, Other | 461, 23, 288k

**FIFA 19 complete player dataset**
18k+ FIFA 19 players, ~90 attributes extracted from the latest FIFA database
Karan Gadiya updated 5 months ago
sports, data visuali..., regression ... | CSV, 2.08MB, CC4 | 183, 14, 185k

**Malaria Cell Images Dataset**
healthcare | Other | 111

(https://www.kaggle.com/docs/kernels)

# Using the Cloud



https://www.kaggle.com/kernels

**Another Kernel**

Draft saved Python Commit

```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load in

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list the files in the input directory

import os
print(os.listdir("../input"))

# Any results you write to the current directory are saved as output.
```

Console CPU 0% GPU OFF RAM 158.1MB/17.2GB Disk 279.1MB/5.2GB

**Sessions**

● Interactive Session 0m:14s / 6h

| | | | |
|---|---|---|---|
| CPU | 0% | RAM | 158.1MB/17.2GB |
| GPU | Off | Disk | 279.1MB/5.2GB |

**Versions**

1 uncommitted draft

Bruno Fernandes's draft

**Draft Environment**

No Data Sources

Connect your Kernel to our library of datasets

+ Add Data

**Settings**

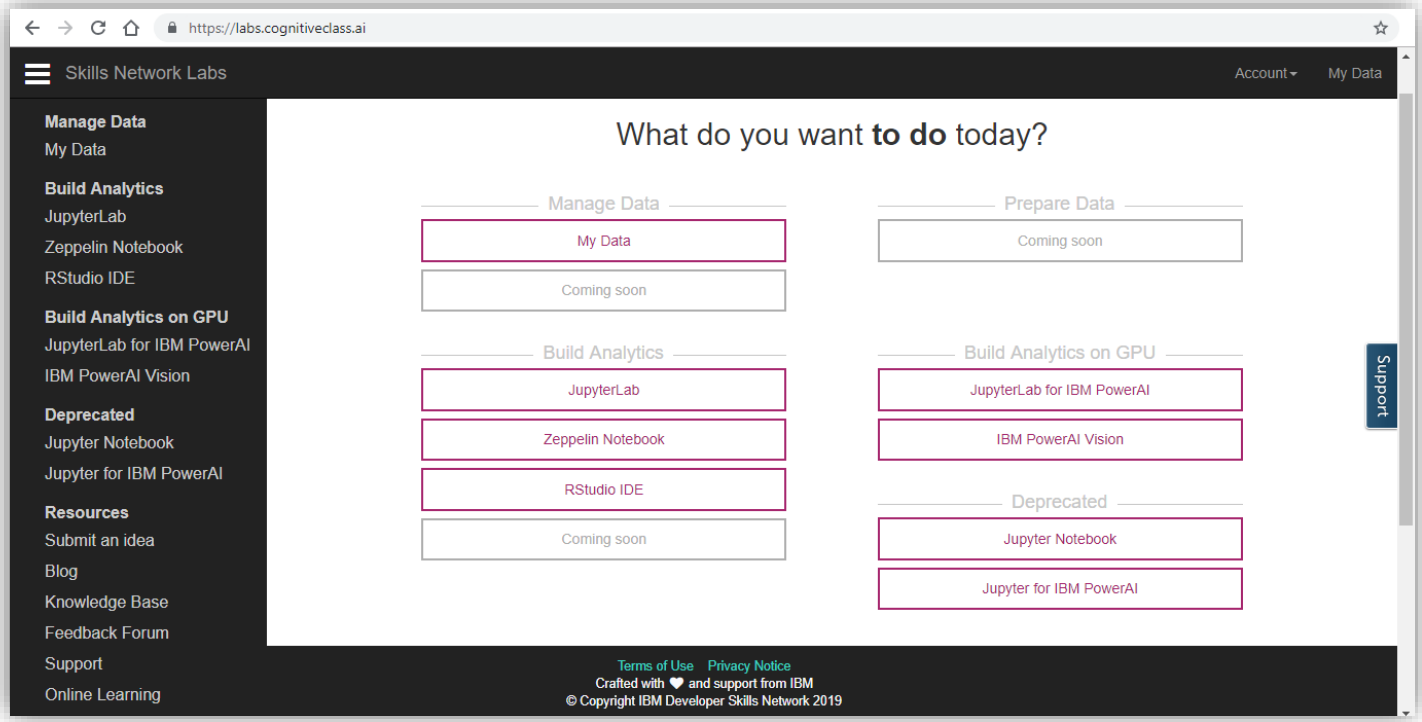
| | |
|---|---|
| Sharing | Private, 0 collaborators |
| Language | Python |
| Docker | Latest available |
| GPU BETA | GPU off |
| Internet BETA | Requires phone verification |
| Utility | Utility Script off |
| Packages | No custom packages |

(https://www.kaggle.com/docs/kernels)

# Using the Cloud

## Free - IBM’s All-in-One Tool for Data Scientists!

- All-in-one solution for programmers, data engineers and data scientists
- Former *Data Scientist Workbench*
- With blurred limits - told to be 100 GB of disk space with 16 GB of RAM



https://labs.cognitiveclass.ai

Skills Network Labs

Account ▾ My Data

**Manage Data**
My Data

**Build Analytics**
JupyterLab
Zeppelin Notebook
RStudio IDE

**Build Analytics on GPU**
JupyterLab for IBM PowerAI
IBM PowerAI Vision

**Deprecated**
Jupyter Notebook
Jupyter for IBM PowerAI

**Resources**
Submit an idea
Blog
Knowledge Base
Feedback Forum
Support
Online Learning

What do you want **to do** today?

Manage Data
My Data
Coming soon

Prepare Data
Coming soon

Build Analytics
JupyterLab
Zeppelin Notebook
RStudio IDE
Coming soon

Build Analytics on GPU
JupyterLab for IBM PowerAI
IBM PowerAI Vision

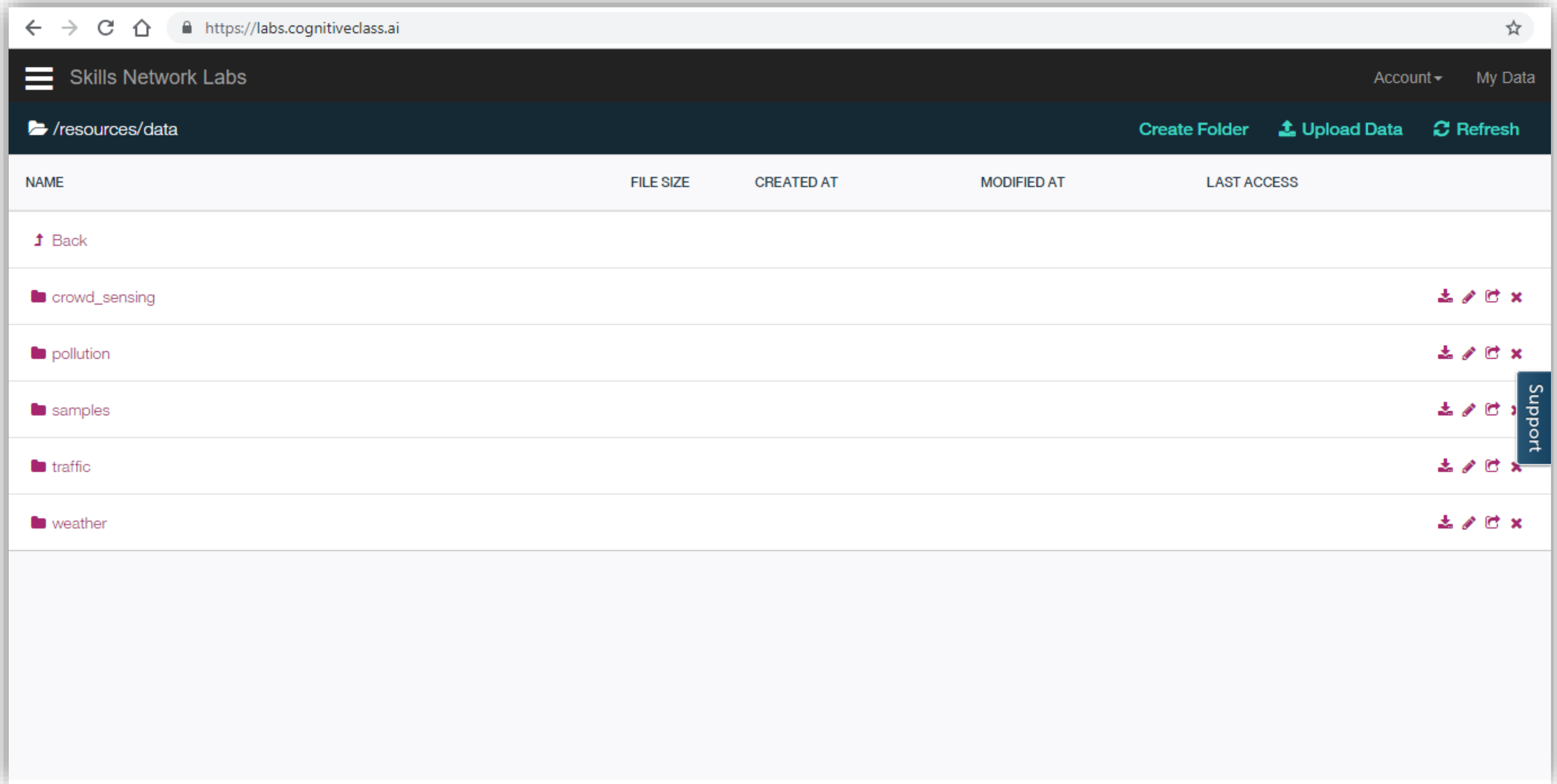
Deprecated
Jupyter Notebook
Jupyter for IBM PowerAI

Support

Terms of Use Privacy Notice
Crafted with ♥ and support from IBM
© Copyright IBM Developer Skills Network 2019

(https://labs.cognitiveclass.ai/)

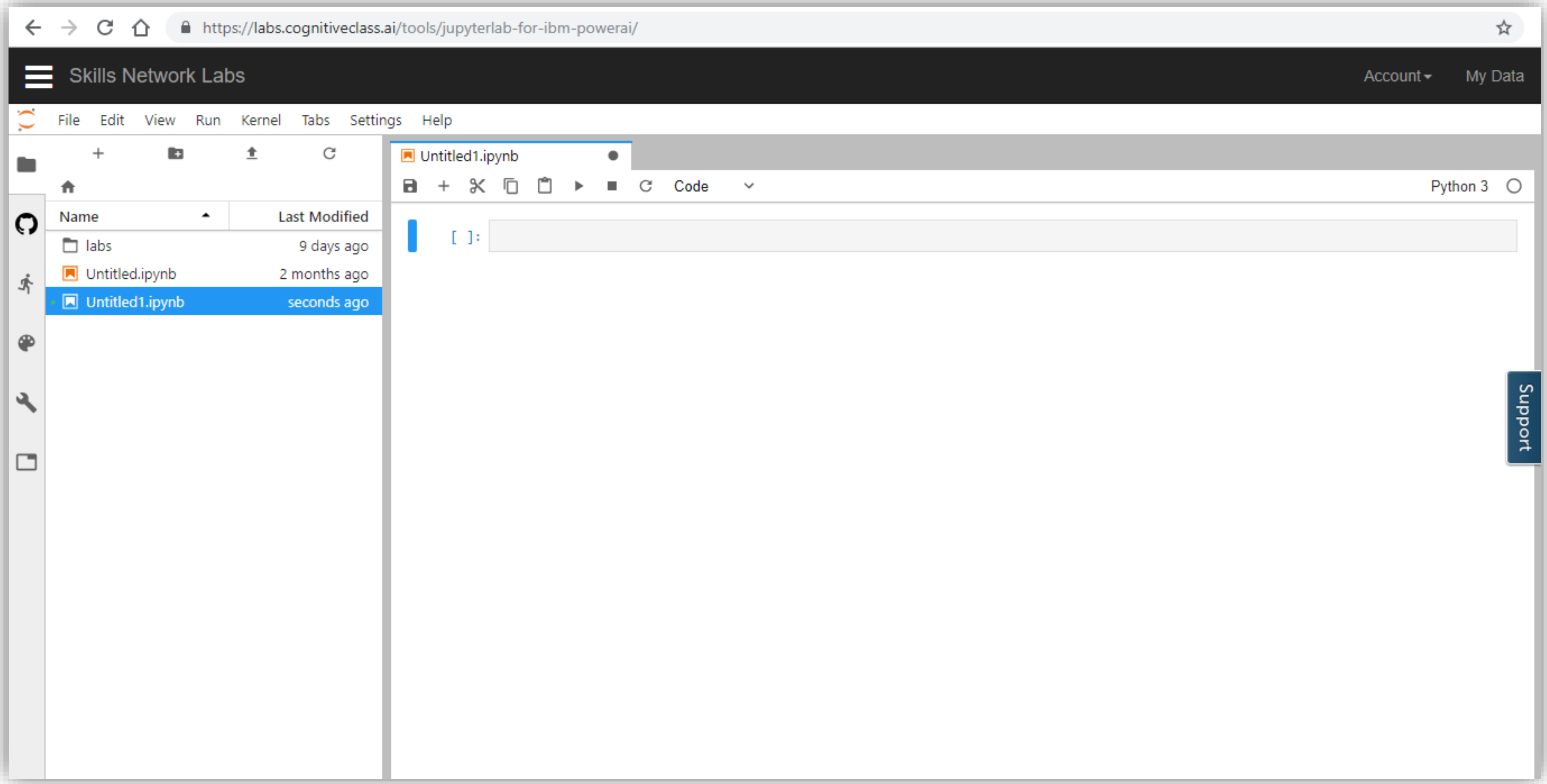
# Using the Cloud



| NAME | FILE SIZE | CREATED AT | MODIFIED AT | LAST ACCESS |
|---|---|---|---|---|
| Back | | | | |
| crowd_sensing | | | | |
| pollution | | | | |
| samples | | | | |
| traffic | | | | |
| weather | | | | |

(https://support.labs.cognitiveclass.ai/knowledgebase)

# Using the Cloud



https://labs.cognitiveclass.ai/tools/jupyterlab-for-ibm-powerai/

| Name | Last Modified |
|---|---|
| labs | 9 days ago |
| Untitled.ipynb | 2 months ago |
| Untitled1.ipynb | seconds ago |

```
[ ]:
```

(https://support.labs.cognitiveclass.ai/knowledgebase)

# Other (payed) options

# A Useful Tip

When you are done, shut down the VM!

# Glossary

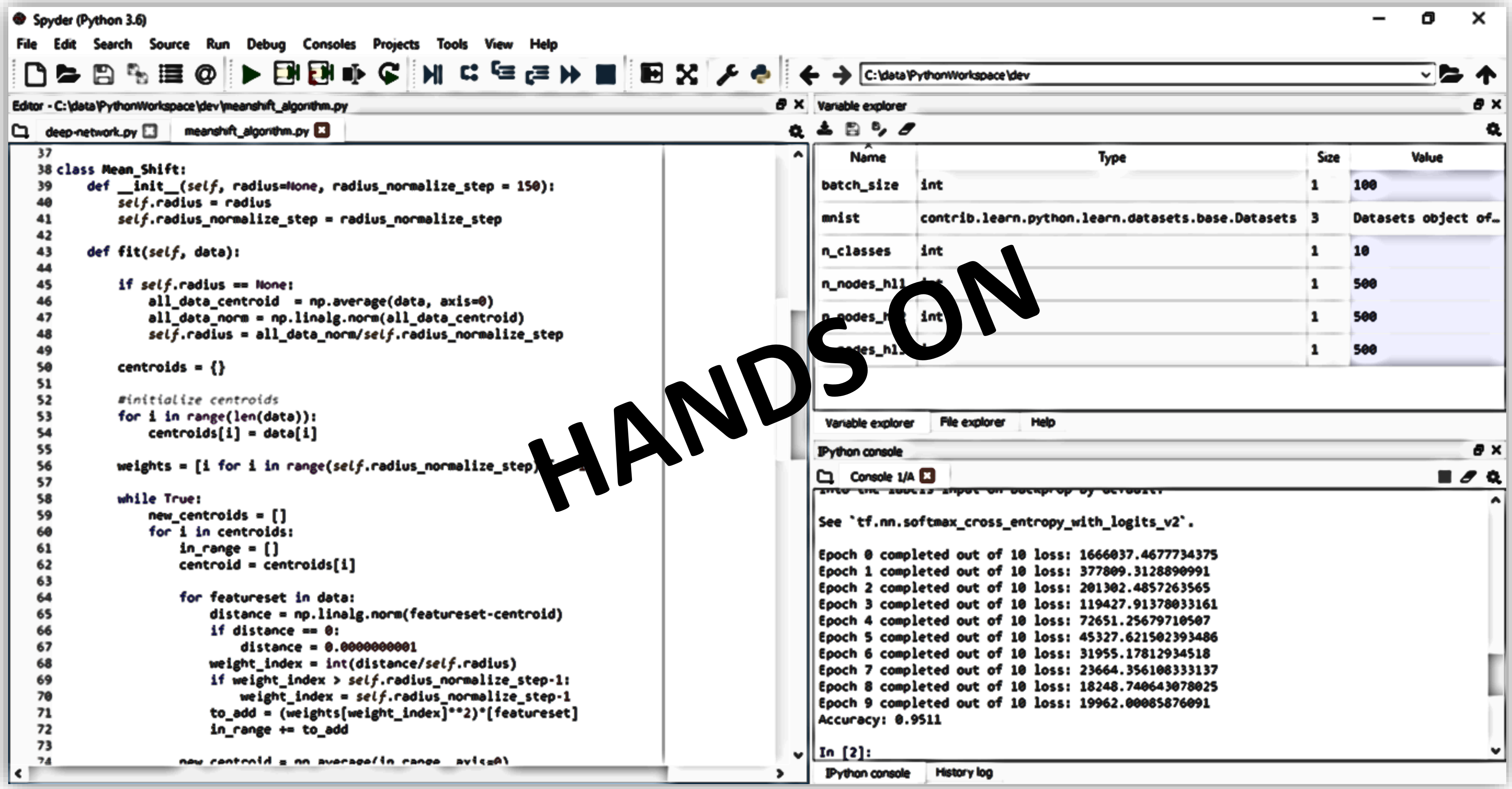
- Anaconda: a package manager, an environment manager, a Python/R data science distribution, and a collection of over 1,500+ open source packages for scientific computing.
- Conda: an open source, cross-platform, language-agnostic package manager and environment management system that installs, runs and updates packages and their dependencies (included in Anaconda).
- Google Colab: cloud-based platform for Machine Learning research. Doesn't require any main settings or installations. If the library that you want to use is not on Colab, just pip it as usual. Colaboratory is built on top of Jupyter Notebook. It shares the notion of magics from Jupyter.
- Python Environments: environments may have different versions of Python and/or packages installed in them. Switching or moving between environments is called activating the environment.
- TensorFlow: a large-scale, distributed, machine learning library. The term also refers to the base API layer in the TensorFlow stack, which supports general computation on dataflow graphs.

(From: https://developers.google.com/machine-learning/glossary)

# Resources

- Official Documentation
  - https://www.tensorflow.org/api_docs/
  - https://docs.conda.io/projects/conda/en/4.6.0/_downloads/52a95608c49671267e40c689e0bc00ca/conda-cheatsheet.pdf
  - https://colab.research.google.com/notebooks/intro.ipynb#
  - https://www.kaggle.com/kernels/welcome
  - …

# Hands On


Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\data\PythonWorkspace\dev
Editor - C:\data\PythonWorkspace\dev\meanshift_algorithm.py
deep-network.py
meanshift_algorithm.py
Variable explorer
Variable explorer File explorer Help
IPython console
Console 1/A
IPython console History log
HANDS ON

```python

class Mean_Shift:
    def __init__(self, radius=None, radius_normalize_step = 150):
        self.radius = radius
        self.radius_normalize_step = radius_normalize_step

    def fit(self, data):

        if self.radius == None:
            all_data_centroid  = np.average(data, axis=0)
            all_data_norm = np.linalg.norm(all_data_centroid)
            self.radius = all_data_norm/self.radius_normalize_step

        centroids = {}

        #initialize centroids
        for i in range(len(data)):
            centroids[i] = data[i]

        weights = [i for i in range(self.radius_normalize_step)[illegible]

        while True:
            new_centroids = []
            for i in centroids:
                in_range = []
                centroid = centroids[i]

                for featureset in data:
                    distance = np.linalg.norm(featureset-centroid)
                    if distance == 0:
                        distance = 0.0000000001
                    weight_index = int(distance/self.radius)
                    if weight_index > self.radius_normalize_step-1:
                        weight_index = self.radius_normalize_step-1
                    to_add = (weights[weight_index]**2)*[featureset]
                    in_range += to_add

                new_centroid = np.average(in_range, axis=0)
```

| Name | Type | Size | Value |
|---|---|---|---|
| batch_size | int | 1 | 100 |
| mnist | contrib.learn.python.learn.datasets.base.Datasets | 3 | Datasets object of… |
| n_classes | int | 1 | 10 |
| n_nodes_hl1 | [illegible] | 1 | 500 |
| n_nodes_hl2 | int | 1 | 500 |
| n_nodes_hl3 | [illegible] | 1 | 500 |

```
into the labels input on backprop by default.

See `tf.nn.softmax_cross_entropy_with_logits_v2`.

Epoch 0 completed out of 10 loss: 1666037.4677734375
Epoch 1 completed out of 10 loss: 377809.3128890991
Epoch 2 completed out of 10 loss: 201302.4857263565
Epoch 3 completed out of 10 loss: 119427.91378033161
Epoch 4 completed out of 10 loss: 72651.25679710507
Epoch 5 completed out of 10 loss: 45327.621502393486
Epoch 6 completed out of 10 loss: 31955.17812934518
Epoch 7 completed out of 10 loss: 23664.356108333137
Epoch 8 completed out of 10 loss: 18248.740643078025
Epoch 9 completed out of 10 loss: 19962.00085876091
Accuracy: 0.9511

In [2]:
```