AI Tech Labs 0⇒1

Jian Tao

jtao@tamu.edu

HPRC Short Course

10/30/2020

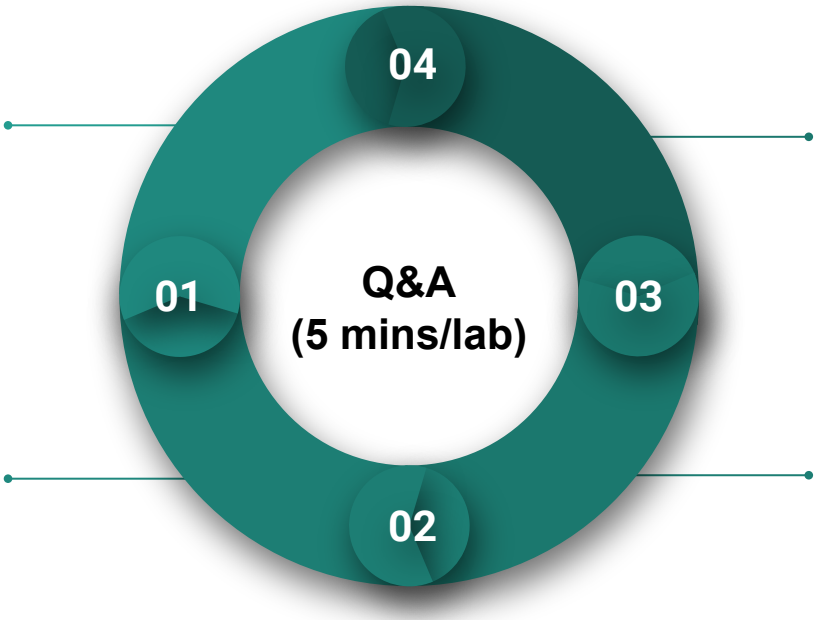
AI Tech Labs

Lab I. JupyterLab (15 mins)

We will set up a Python virtual environment and run JupyterLab on the HPRC Portal..

Lab II. Data Exploration (30 mins)

We will go through simple examples with two popular Python modules: Pandas and Matplotlib for simple data exploration.



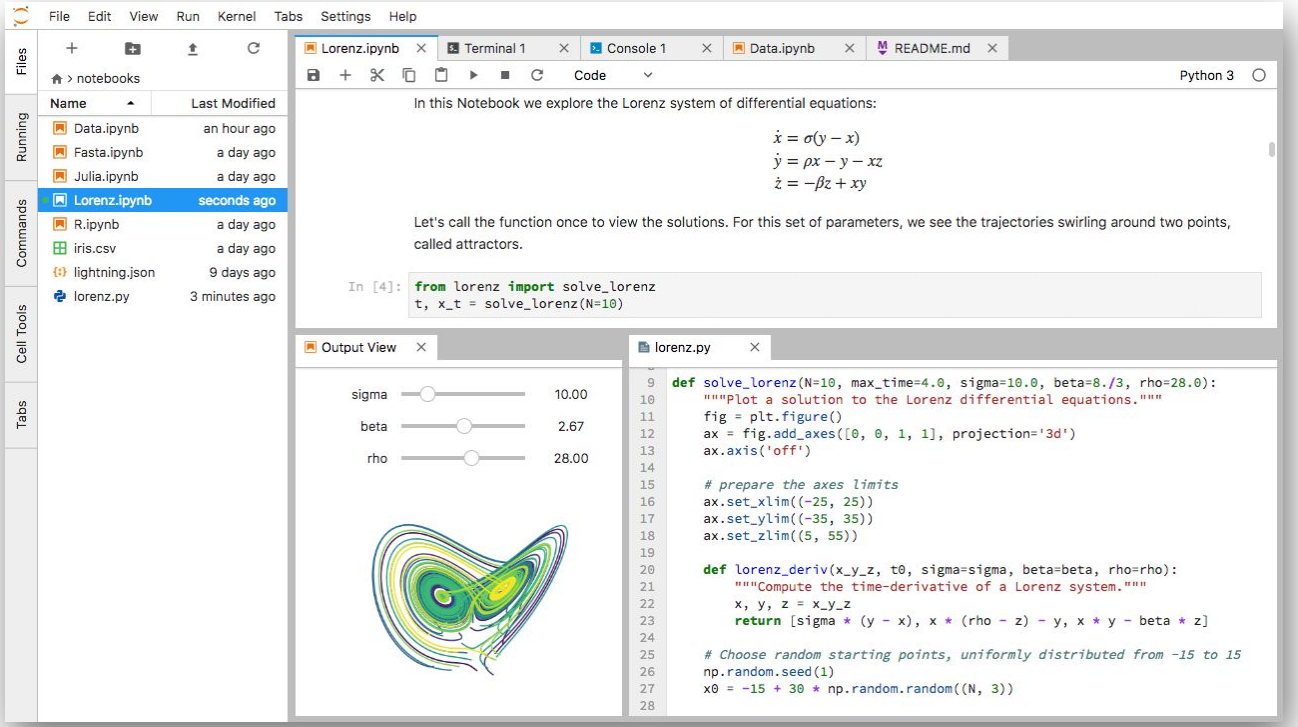
Lab IV. Deep Learning (30 minutes)

We will learn how to use Keras to create and train a simple image classification model with deep neural network (DNN).

Lab III Machine Learning (30 minutes)

We will learn to use scikit-learn for linear regression and classification applications.

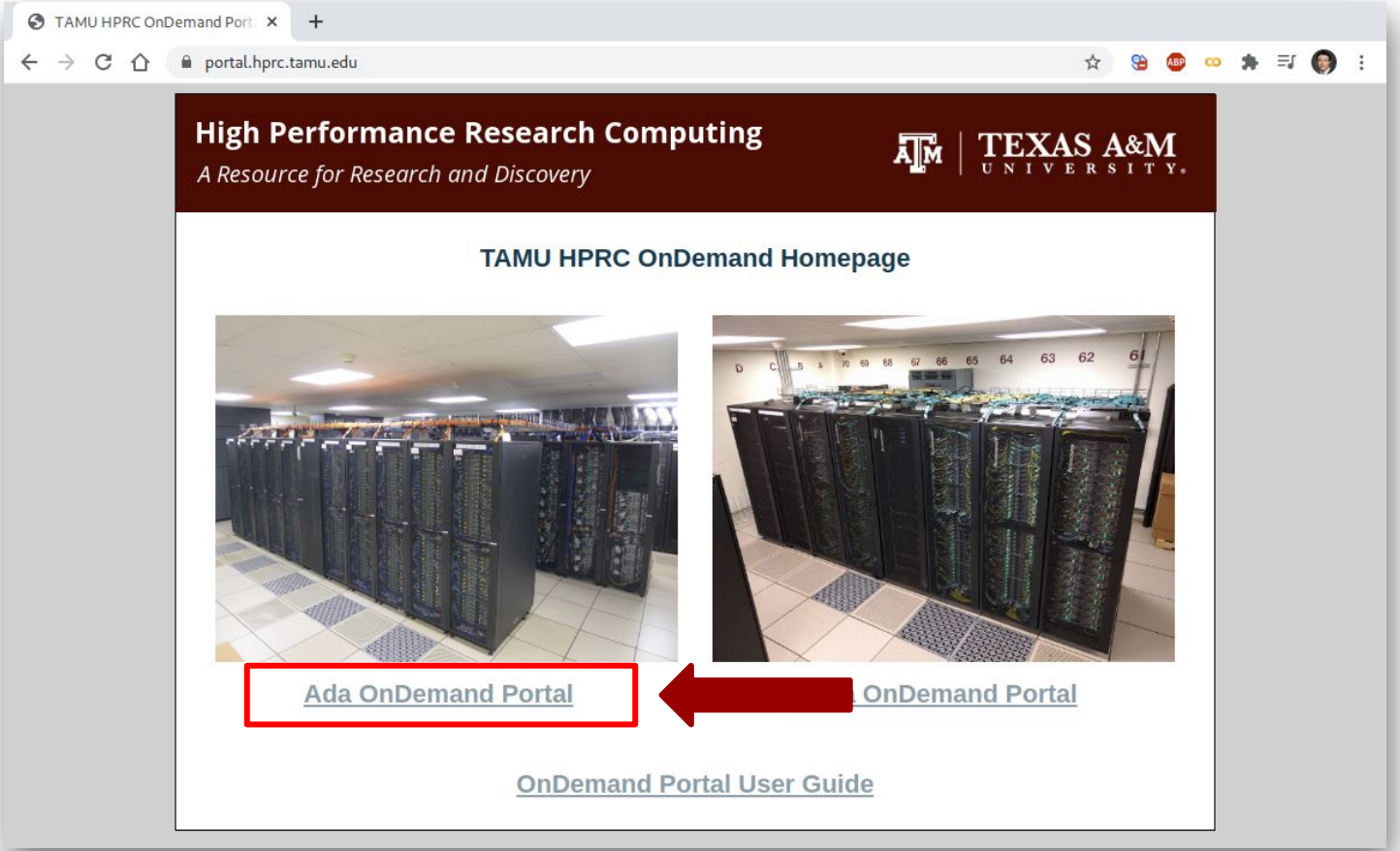
Lab I. JupyterLab



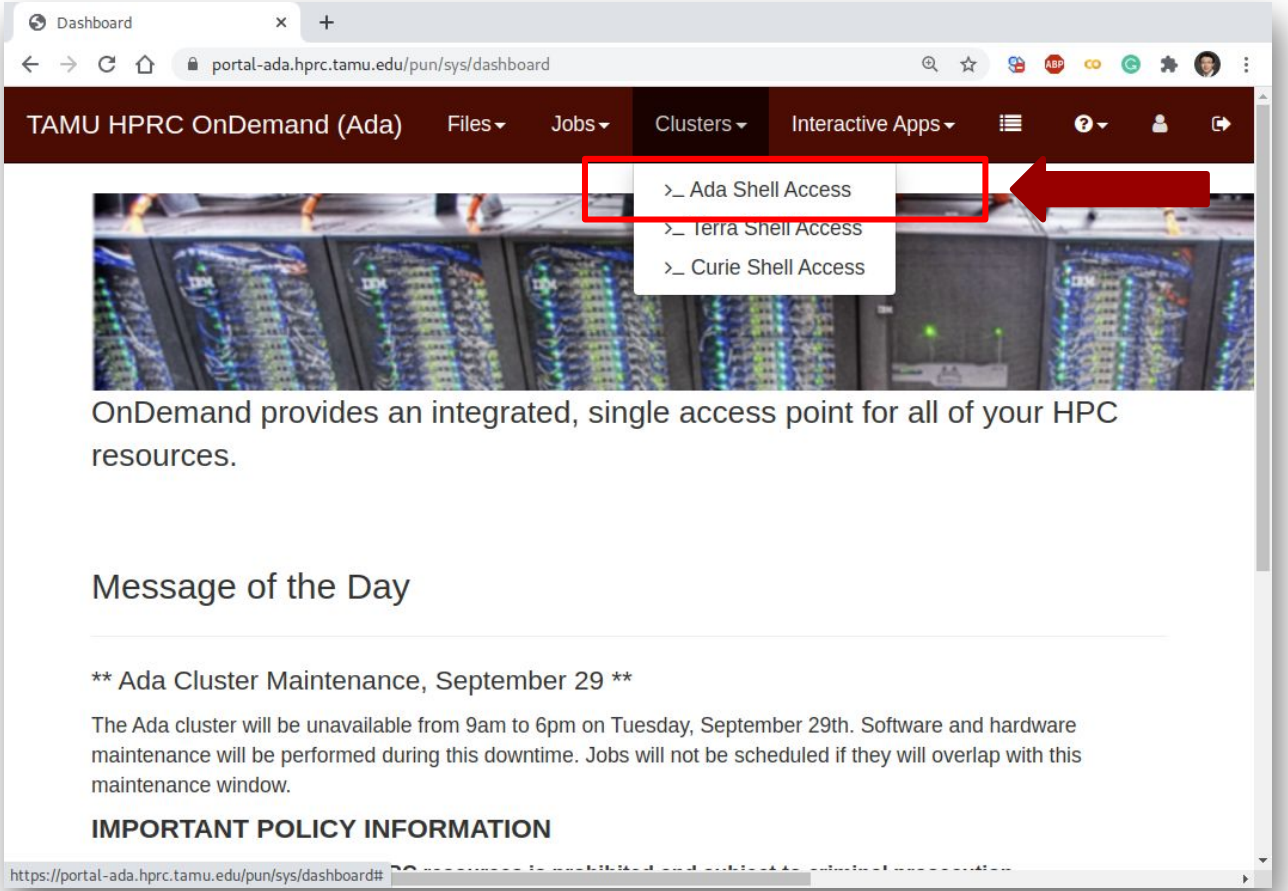
L1 - Resources

- Texas A&M High Performance Research Computing (HPRC)
- Ada Quick Start Guide
- HPRC Portal
- HPRC YouTube Channel
- Jupyter Project

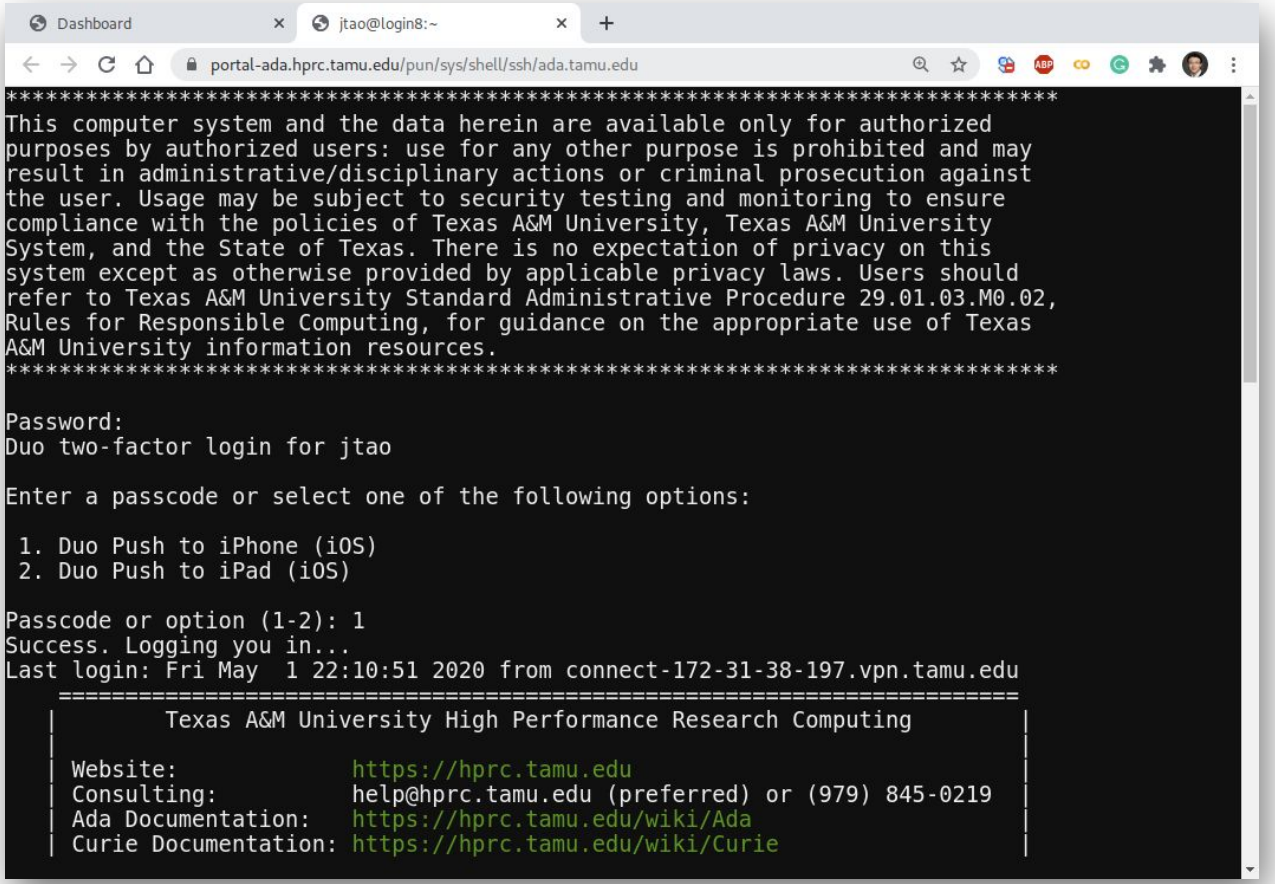
L1 - Login HPRC Portal



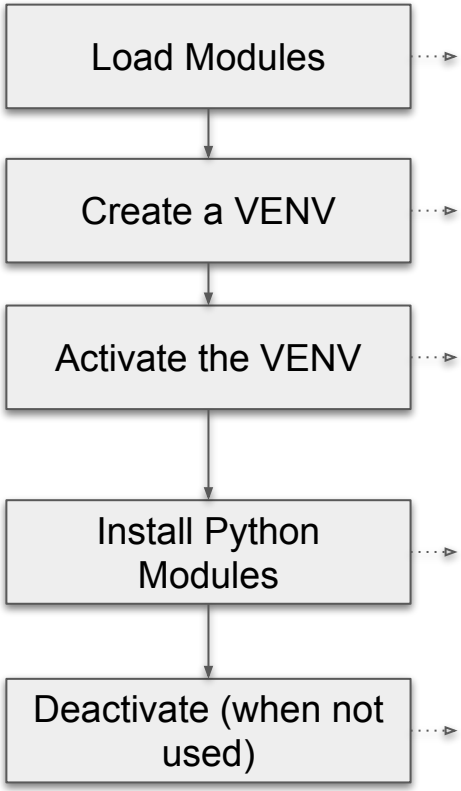
L1 - Shell Access - I



L1 - Shell Access - II



L1 - Python Virtual Environment (VENV)

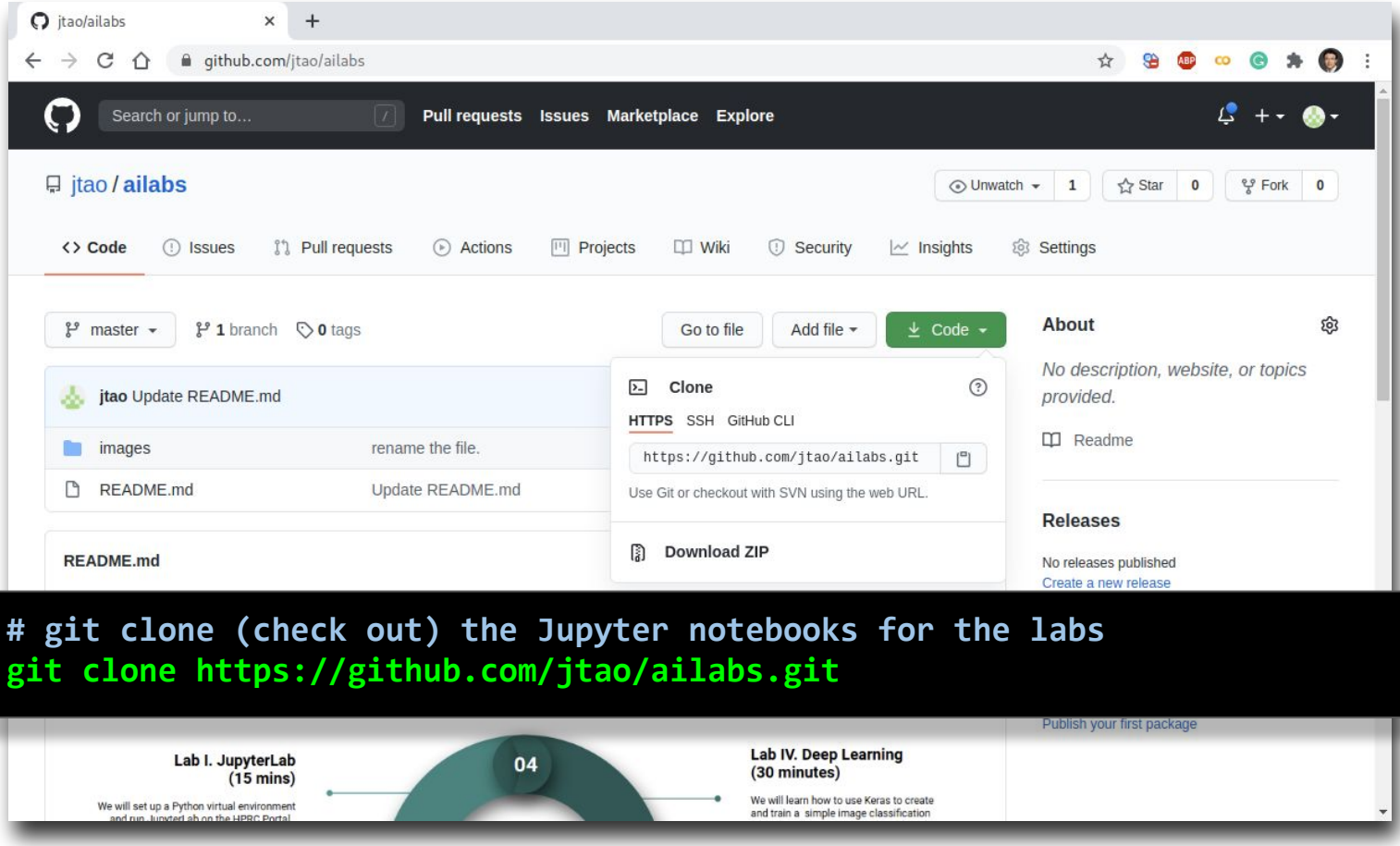



```
# clean up and load Anaconda
cd $SCRATCH
module purge
module load Anaconda/3-5.0.0.1

# create a Python virtual environment
conda create -n mylab

# activate the virtual environment
source activate mylab

# install required package to be used in the portal
conda install jupyterlab=1.2.2
conda install pandas matplotlib
conda install scikit-learn
conda install tensorflow

# deactivate the virtual environment
# source deactivate
```

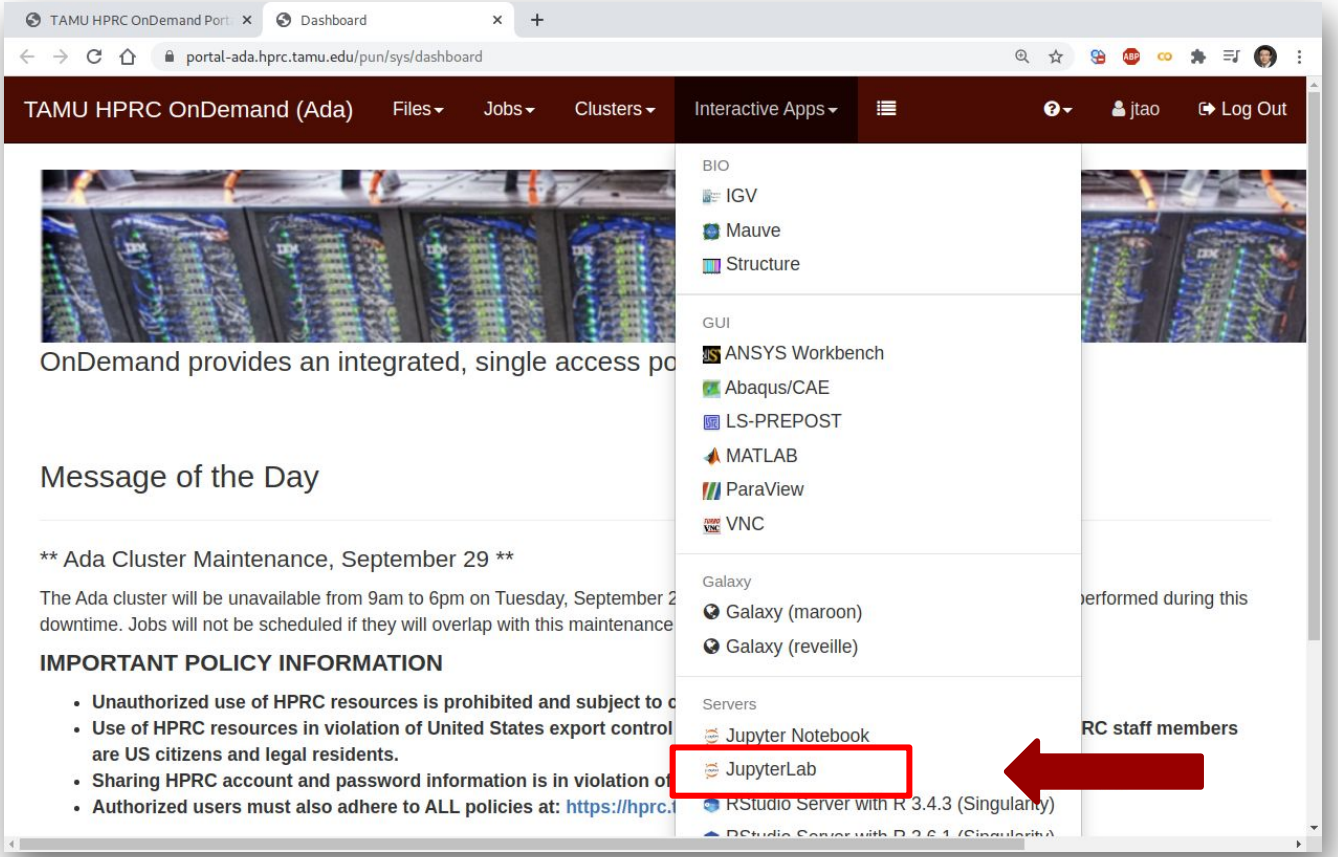
L1 - Common Anaconda Commands

```
# Conda virtual environment
conda info                          # show Conda installation
conda create -n VENV                # create a virtual environment
conda create -n VENV python=3.4     # create a venv with a py version
conda env list                      # list installed venv

# Conda package management
conda list                          # list all installed packages
conda search  PACKAGENAME           # search a Conda package
conda install PACKAGENAME           # install a Conda package
conda update  PACKAGENAME           # update a Conda package
conda remove  PACKAGENAME           # remove a Conda package

# install required package to be used in the portal
conda install jupyterlab=1.2.2
conda install pandas matplotlib
conda install scikit-learn
conda install tensorflow
```

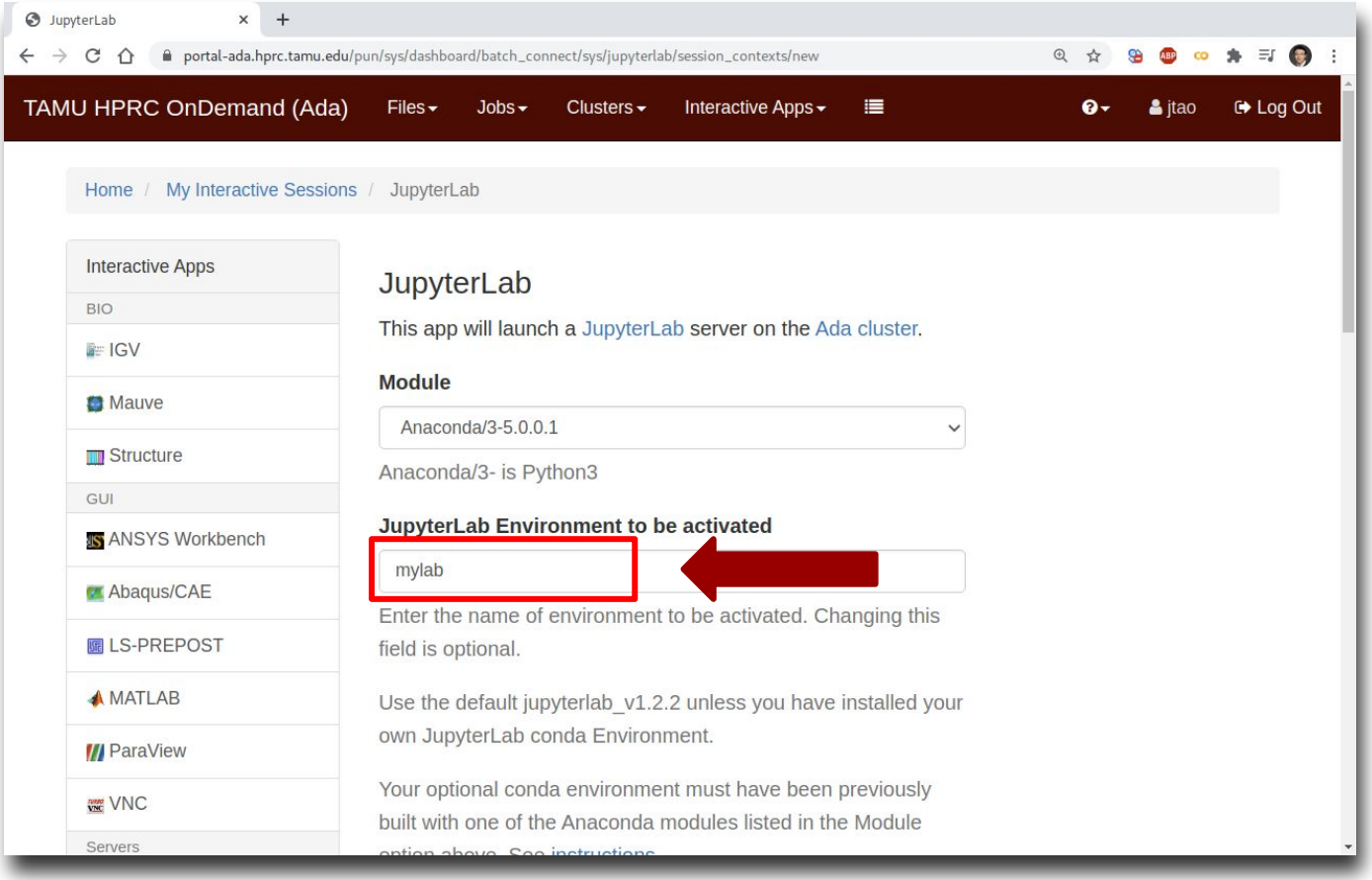
L1 - Check out Exercises



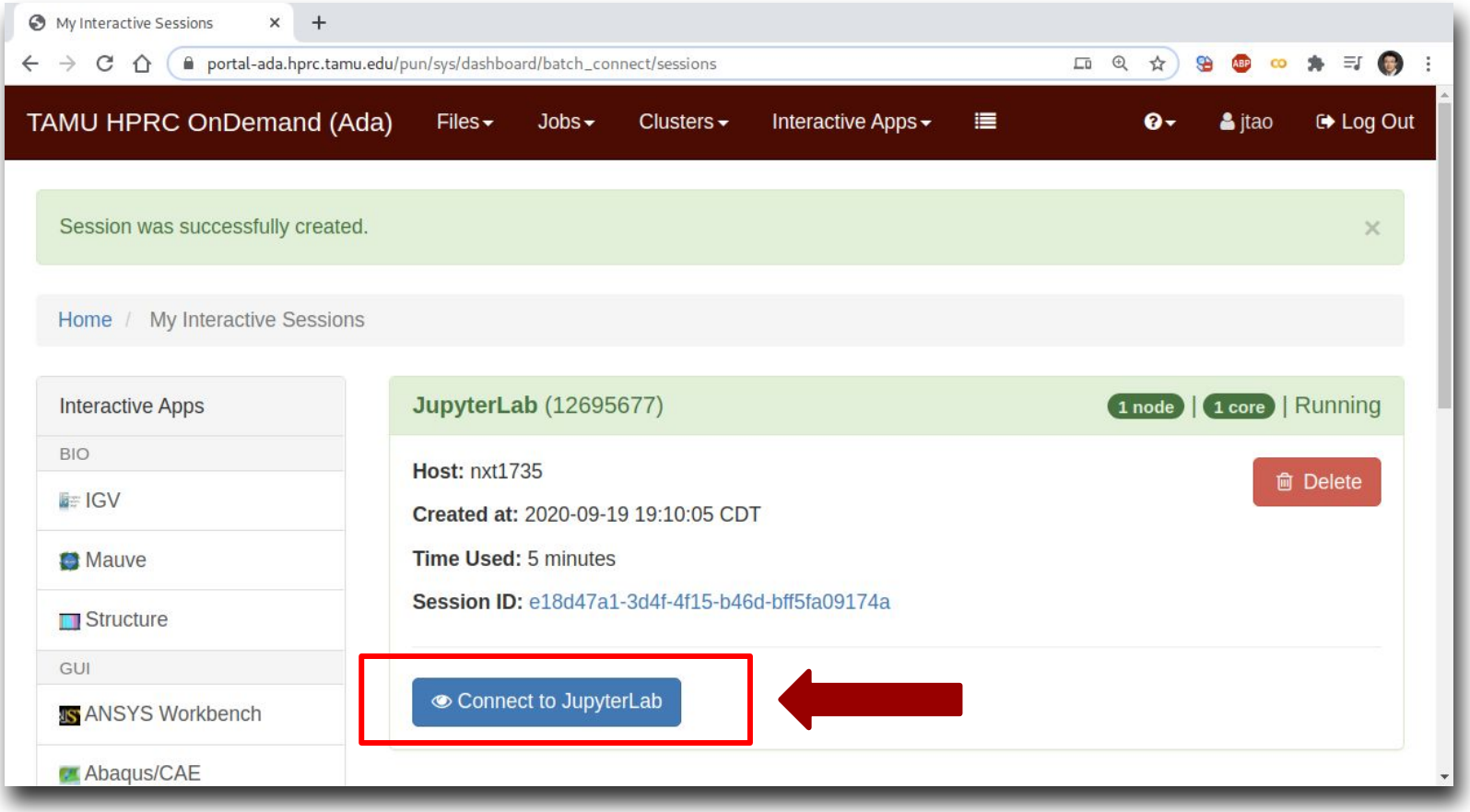
L1 - Go to JupyterLab Page



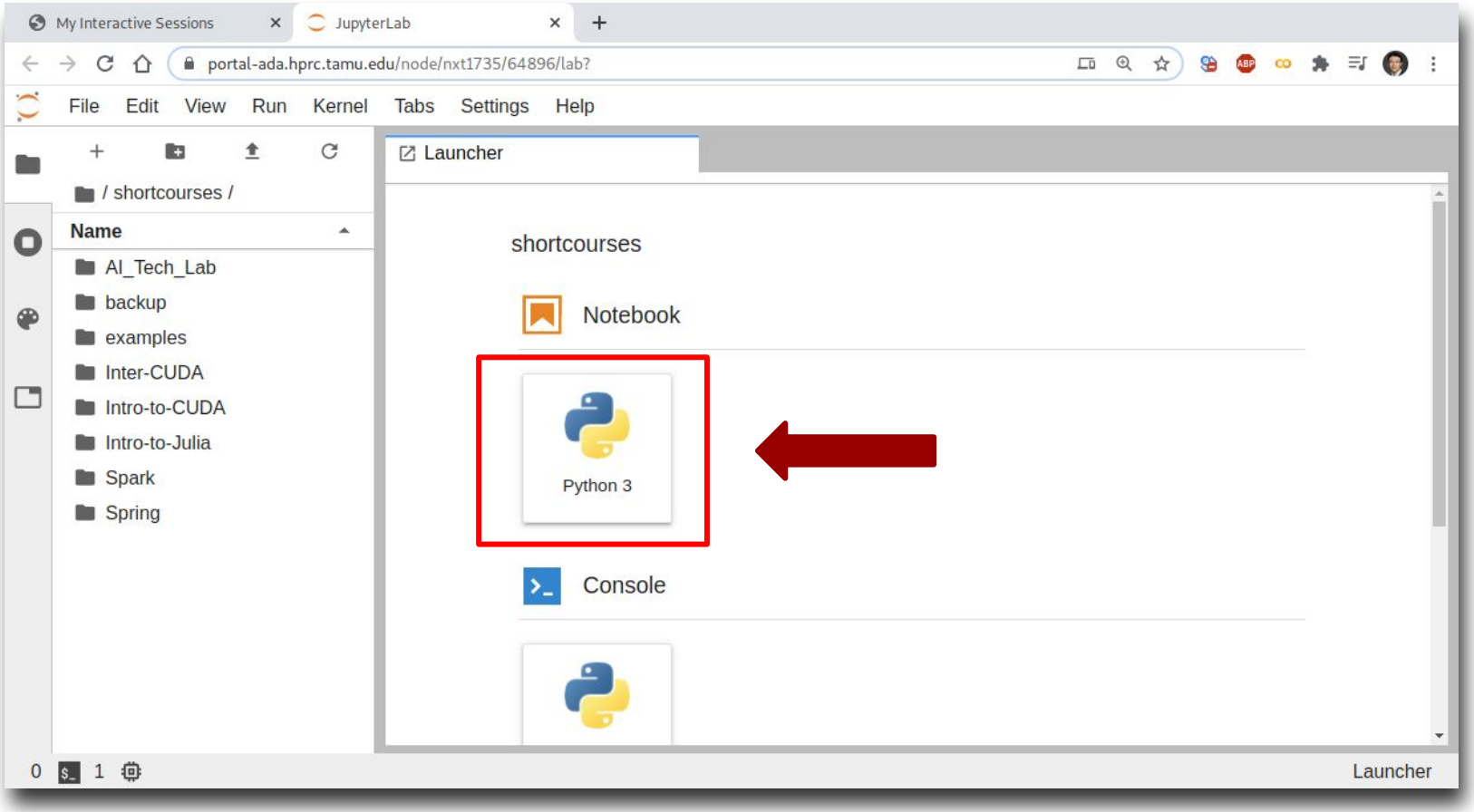
L1 - Set Virtual Environment



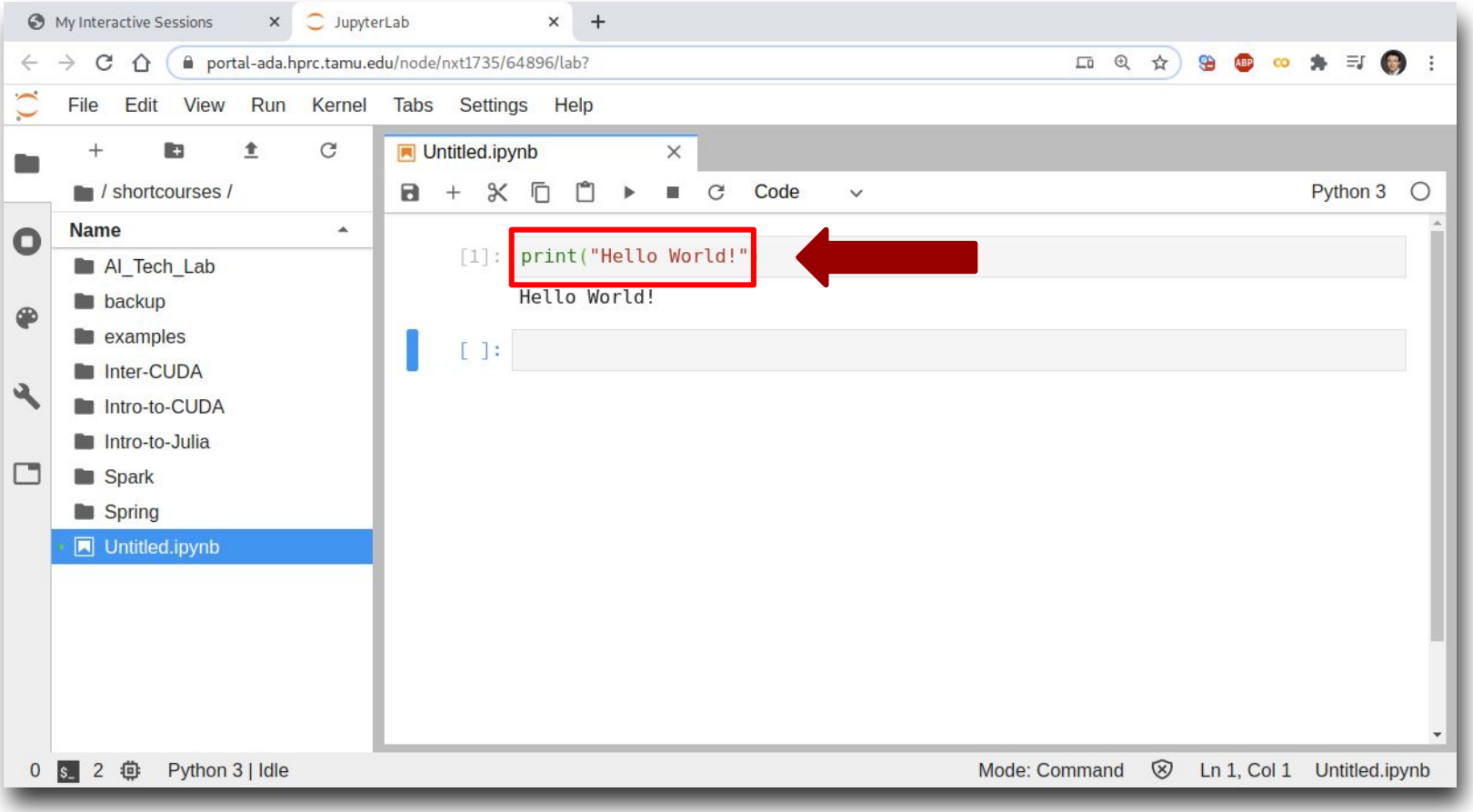
L1 - Connect to JupyterLab



L1 - Create a Jupyter Notebook



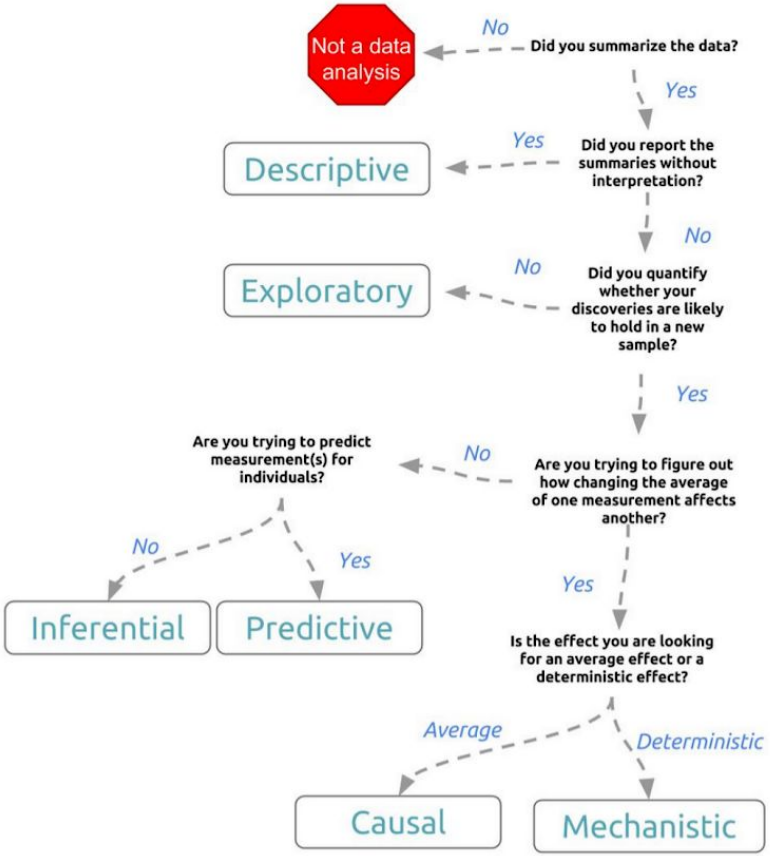
L1 - Test JupyterLab



Lab II. Data Exploration

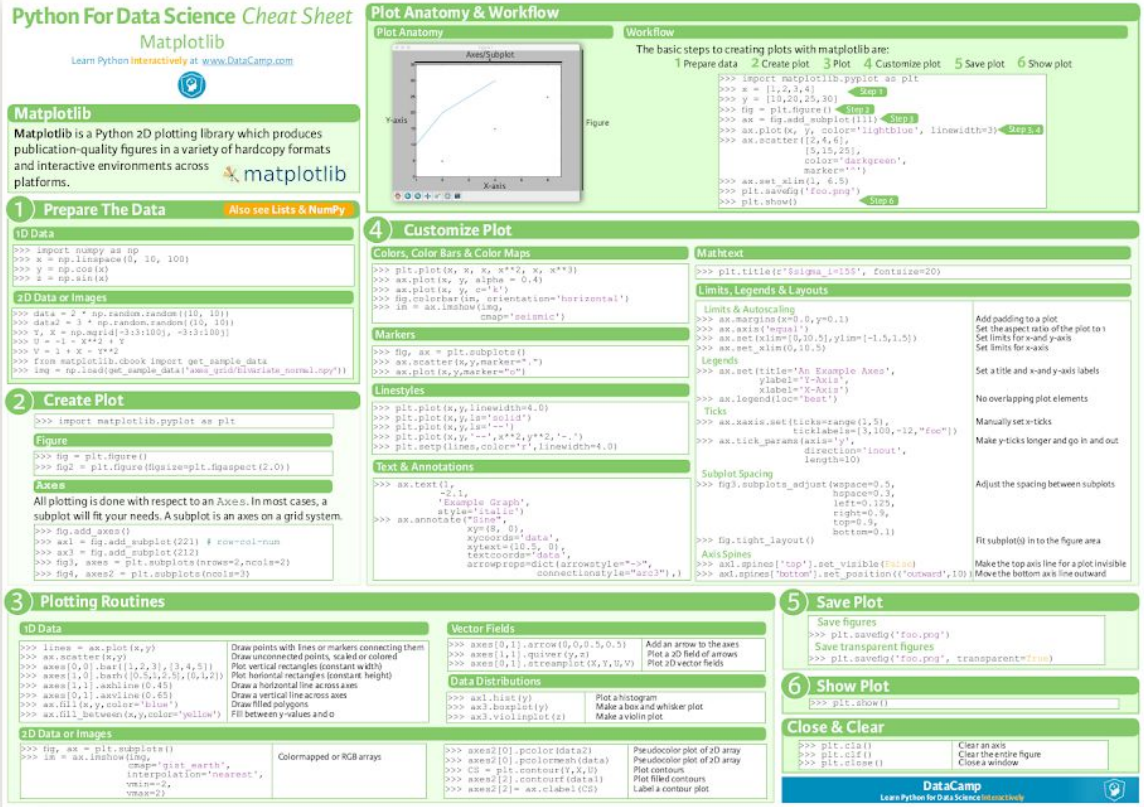
Types of Data Science Problems

- **Descriptive** (summaries, e.g., census)
- **Exploratory** (search for unknowns, e.g., SETI@home, Einstein@home)
- **Inferential** (find correlations, e.g., many social studies)
- **Predictive** (make predictions, e.g., Face ID, Echo, Siri)
- **Causal** (explore causation, e.g., smoking versus lung cancer)
- **Mechanistic** (determine governing principles, e.g., experimental science)



Credit: Jeff Leek - The Elements of Data Analytic Style

Matplotlib Cheat Sheet

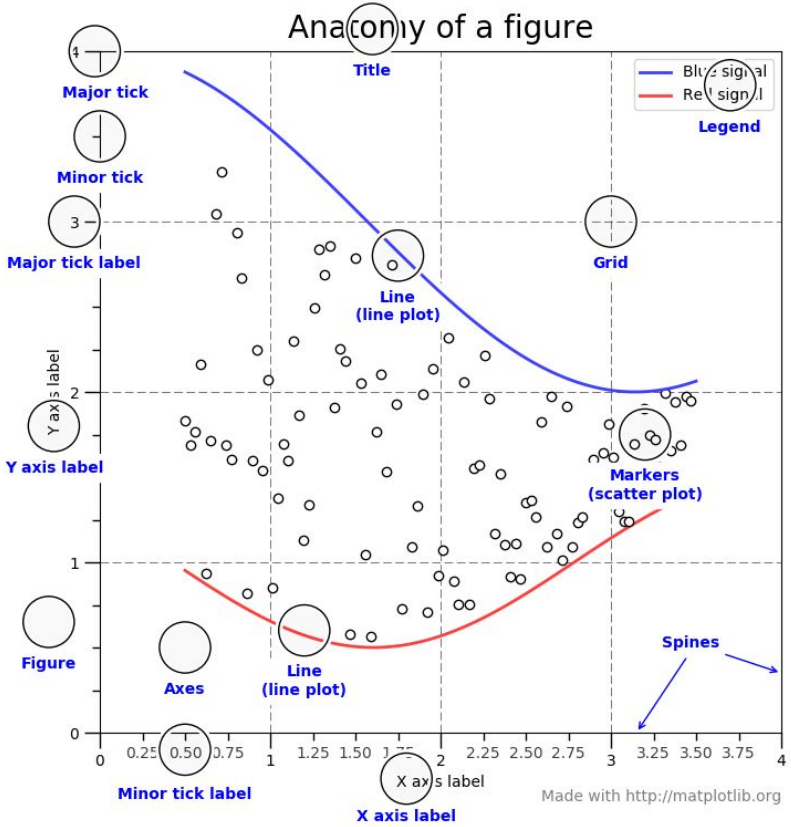


https://s3.amazonaws.com/assets.datacamp.com/blog_assets/Python_Matplotlib_Cheat_Sheet.pdf

Key Plotting Concepts in Matplotlib

- **Matplotlib: Figure**
  Figure is the object that keeps the whole image output.
  Adjustable parameters include:
  1. Image size (set_size_inches())
  2. Whether to use tight_layout (set_tight_layout())

- **Matplotlib: Axes**
  Axes object represents the pair of axis that contain a single plot (x-axis and y-axis). The Axes object also has more adjustable parameters:
  1. The plot frame (set_frame_on() or set_frame_off())
  2. X-axis and Y-axis limits (set_xlim() and set_ylim())
  3. X-axis and Y-axis Labels (set_xlabel() and set_ylabel())
  4. The plot title (set_title())



(Credit: matplotlib.org)

Data Structures

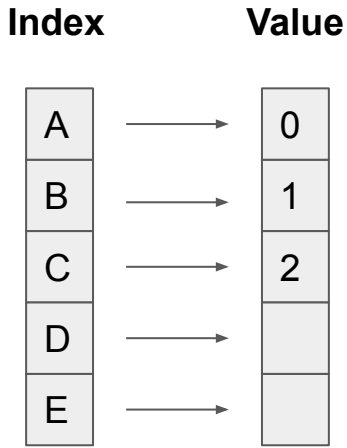
Pandas has two data structures that are descriptive and optimized for data with different dimensions.

- **Series:** 1D labeled homogeneously-typed array
- **DataFrame:** General 2D labeled, size-mutable tabular structure with potentially heterogeneously-typed columns

Series in pandas

"Series is a one-dimensional labeled array capable of holding any data type (integers, strings, floating point numbers, Python objects, etc.). The axis labels are collectively referred to as the index." - pandas site

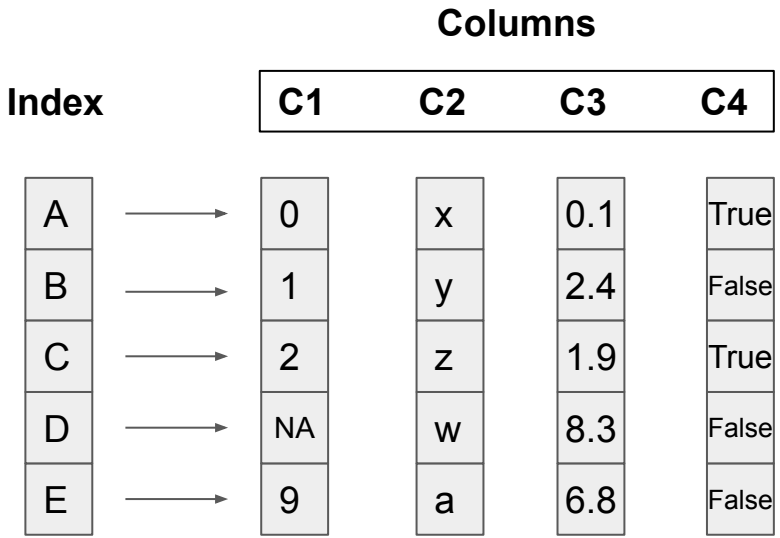
```
In [3]: s = pd.Series(np.random.randn(5),
                      index=['a', 'b', 'c', 'd', 'e'])
In [5]: s.index
In [6]: pd.Series(np.random.randn(5))
In [7]: d = {'b': 1, 'a': 0, 'c': 2}
In [8]: pd.Series(d)
In [12]: pd.Series(5., index=['a', 'b', 'c', 'd', 'e'])
```



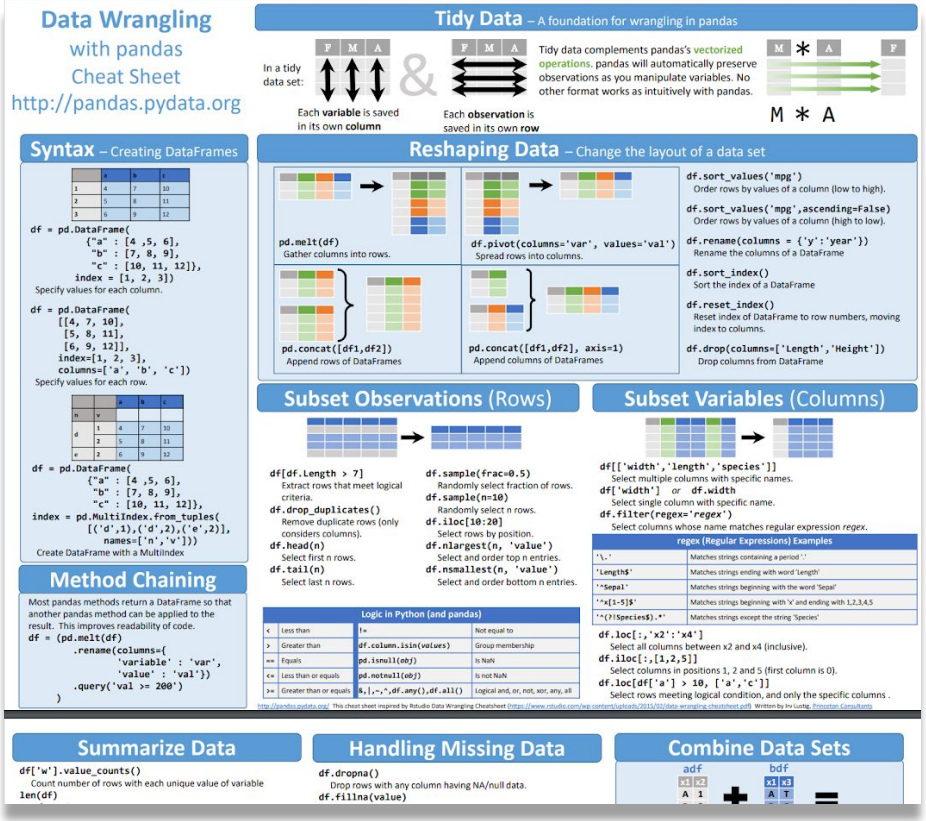
DataFrame in pandas

"Two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). Arithmetic operations align on both row and column labels. Can be thought of as a dict-like container for Series objects. The primary pandas data structure." - [pandas site]

```
In [2]: d = {'col1': [1, 2], 'col2': [3, 4]}
In [3]: df = pd.DataFrame(data=d)
In [5]: df.index
In [6]: df =  pd.DataFrame(
            np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]]),
            columns=['a', 'b', 'c'])
```

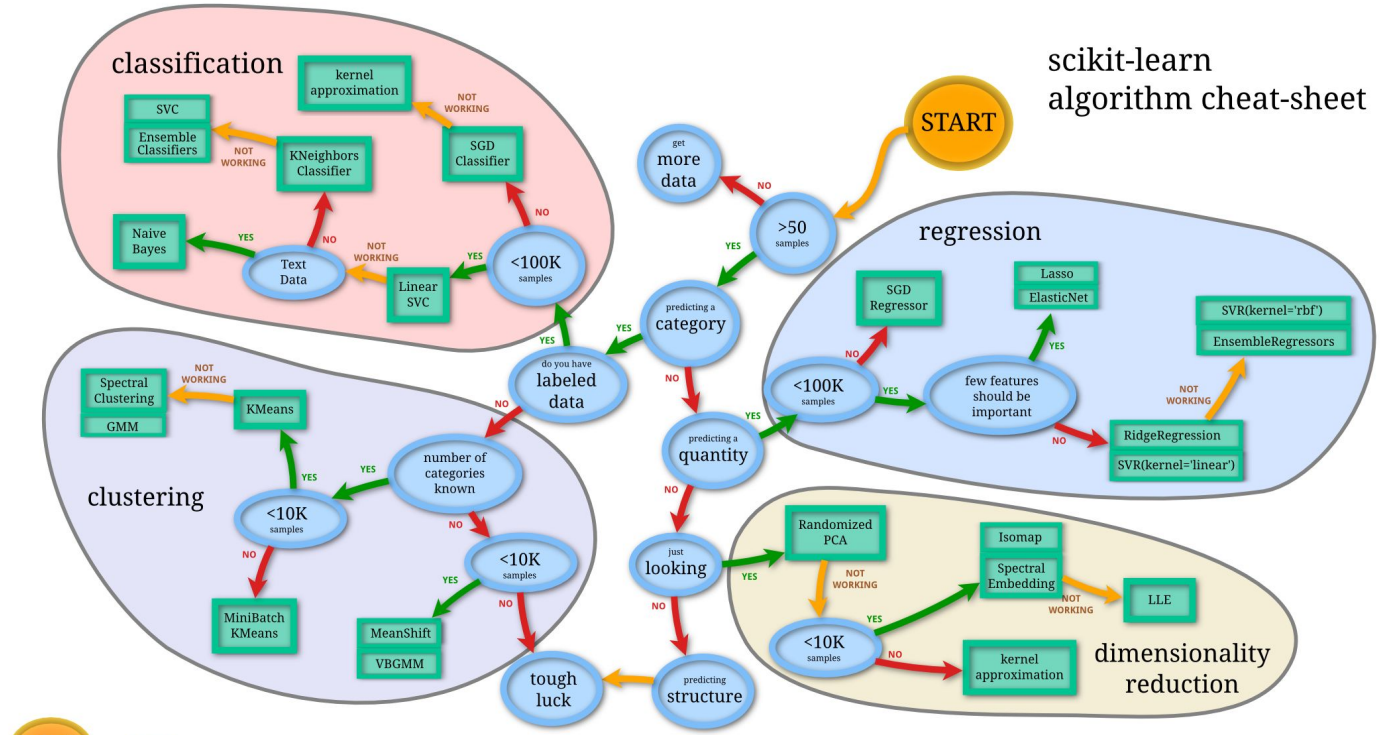


Pandas Cheat Sheet



https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf

Lab III. Machine Learning

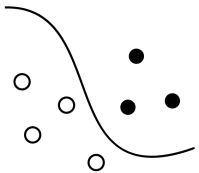


Main Features of scikit-learn

Classification

Identifying category of an object

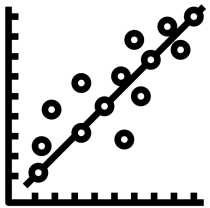
Applications: Spam detection, image recognition.
Algorithms: SVM, nearest neighbors, random forest, and more...



Regression

Predicting a attribute for an object

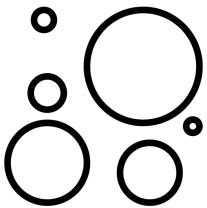
Applications: Drug response, Stock prices.
Algorithms: SVR, nearest neighbors, random forest, and more...



Clustering

Grouping similar objects into sets

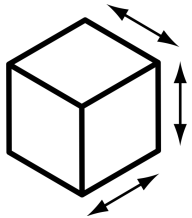
Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, and more...



Dimension Reduction

Reducing the number of dimensions

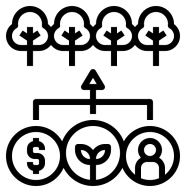
Applications: Visualization, Increased efficiency **Algorithms:** k-Means, feature selection, non-negative matrix factorization, and more...



Model Selection

Selecting models with parameter search

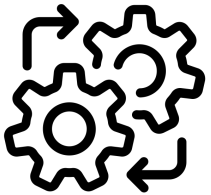
Applications: Improved accuracy via parameter tuning **Algorithms:** grid search, cross validation, metrics, and more...



Preprocessing

Preprocessing data to prepare for modeling

Applications: Transforming input data such as text for use with machine learning algorithms. **Algorithms:** preprocessing, feature extraction, and more...



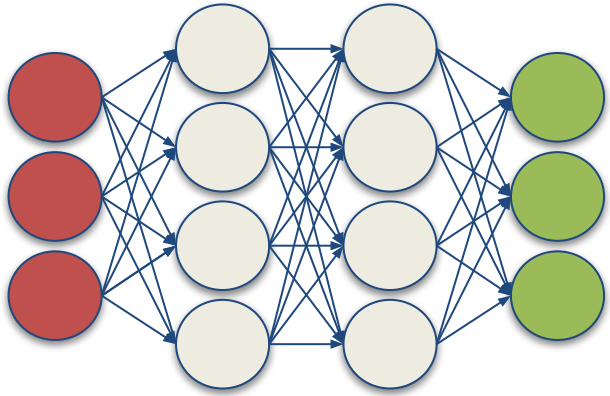
Jian Tao @ Texas A&M Engineering Experiment Station

Lab IV. Deep Learning

Deep Learning
by Ian Goodfellow, Yoshua Bengio, and Aaron Courville
http://www.deeplearningbook.org/

Animation of Neutron Networks
by Grant Sanderson
https://www.3blue1brown.com/

Visualization of CNN
by Adam Harley
https://www.cs.ryerson.ca/~aharley/vis/conv/

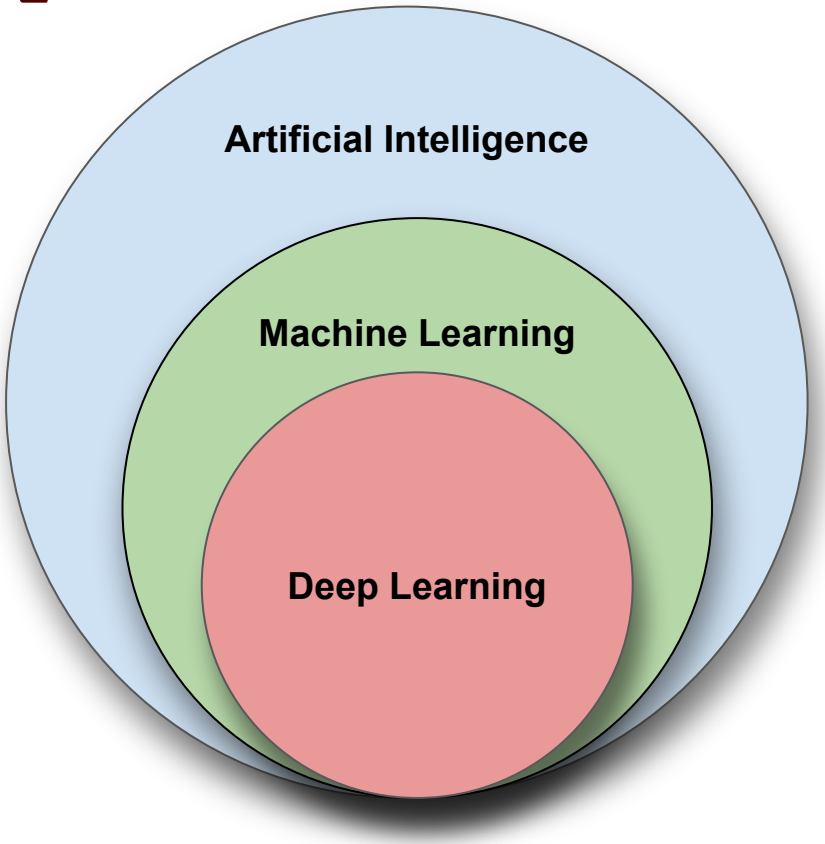


TensorFlow 2.0

K Keras

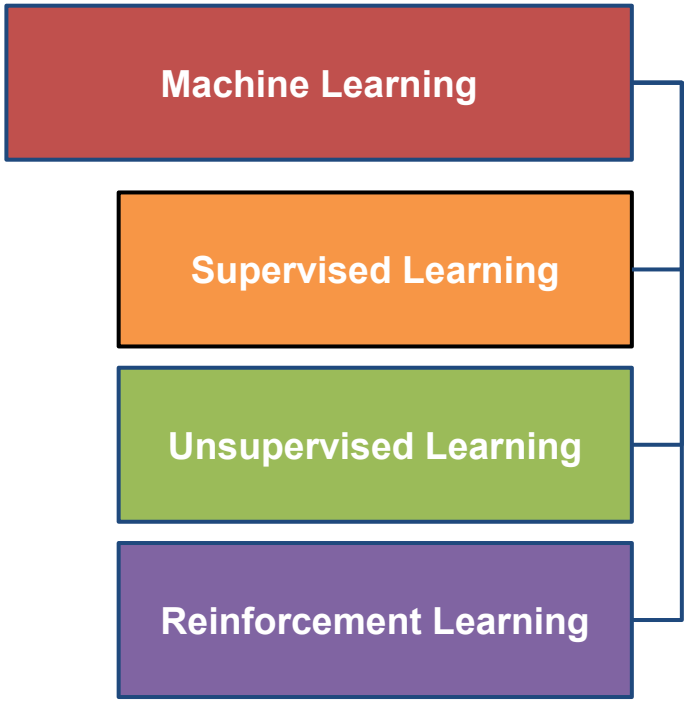
Relationship of AI, ML, and DL

- **Artificial Intelligence (AI)** is anything about man-made intelligence exhibited by machines.
- **Machine Learning (ML)** is an approach to achieve **AI**.
- **Deep Learning (DL)** is one technique to implement **ML**.

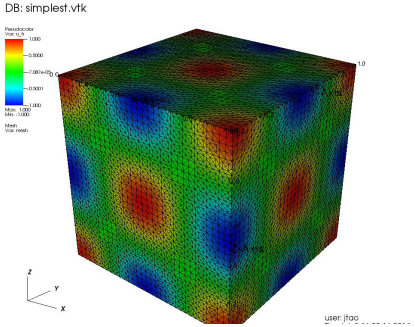


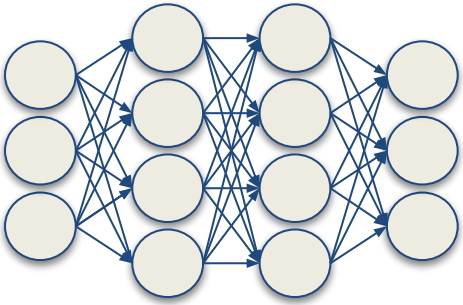
Types of ML Algorithms

- **Supervised Learning**
  - trained with labeled data; including regression and classification problems
- **Unsupervised Learning**
  - trained with unlabeled data; clustering and association rule learning problems.
- **Reinforcement Learning**
  - no training data; stochastic Markov decision process; robotics and self-driving cars.

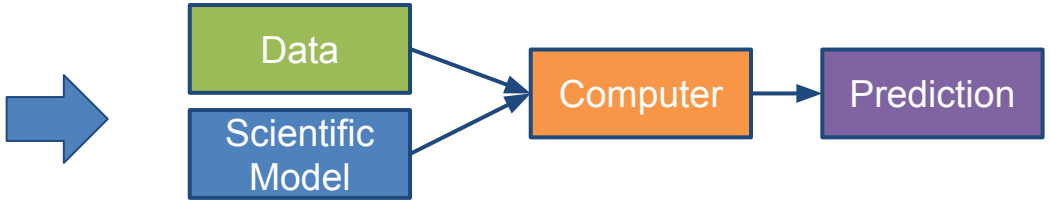


Machine Learning

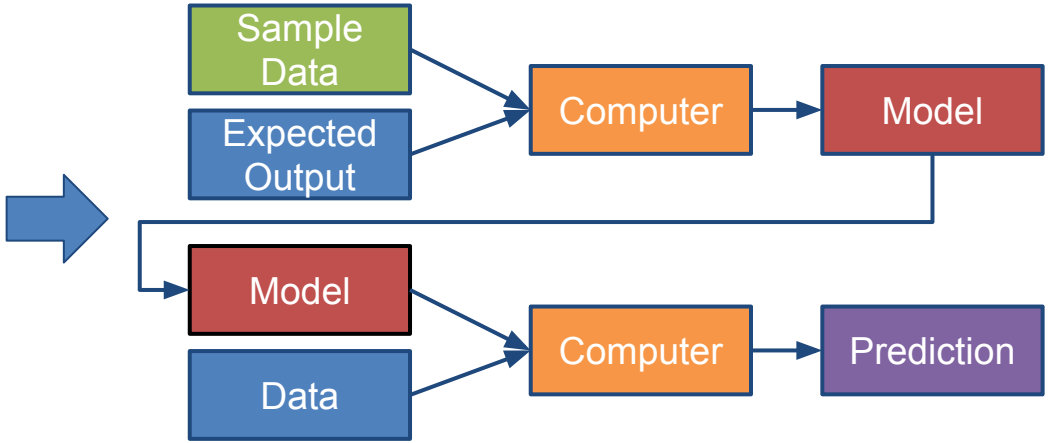




Traditional Modeling



Machine Learning (Supervised Learning)



Inputs and Outputs

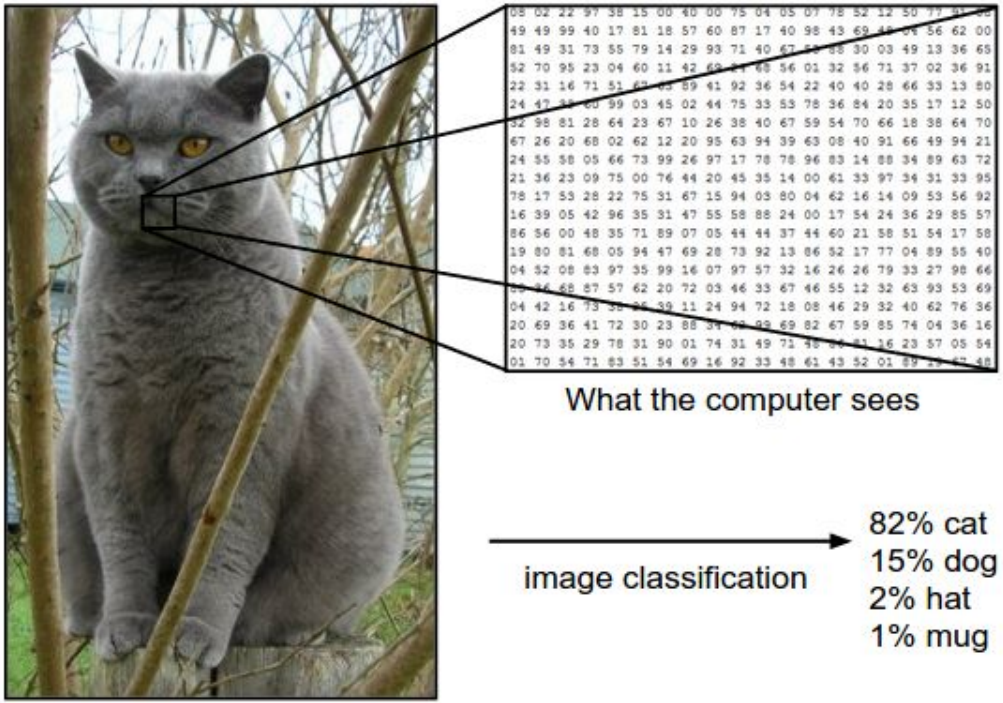
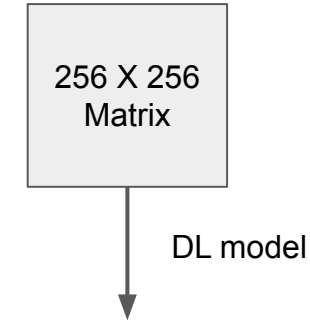
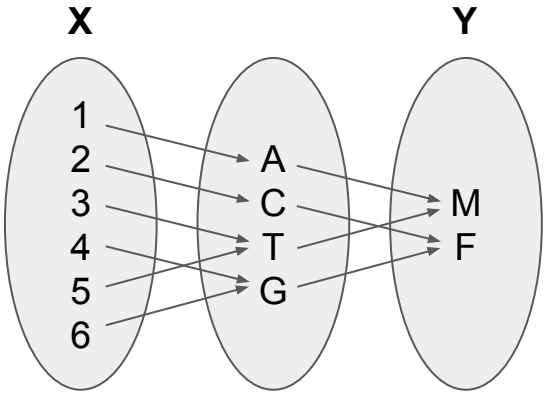


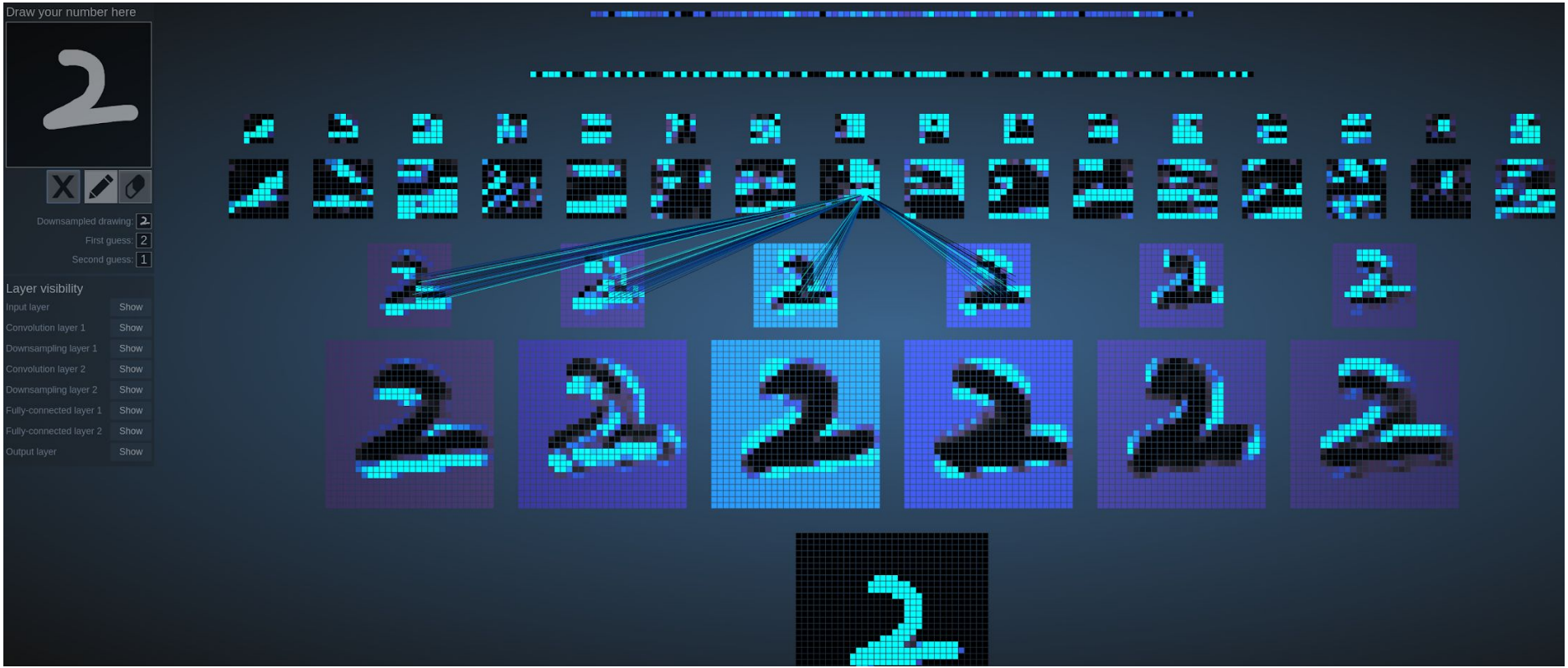
Image from the Stanford CS231 Course





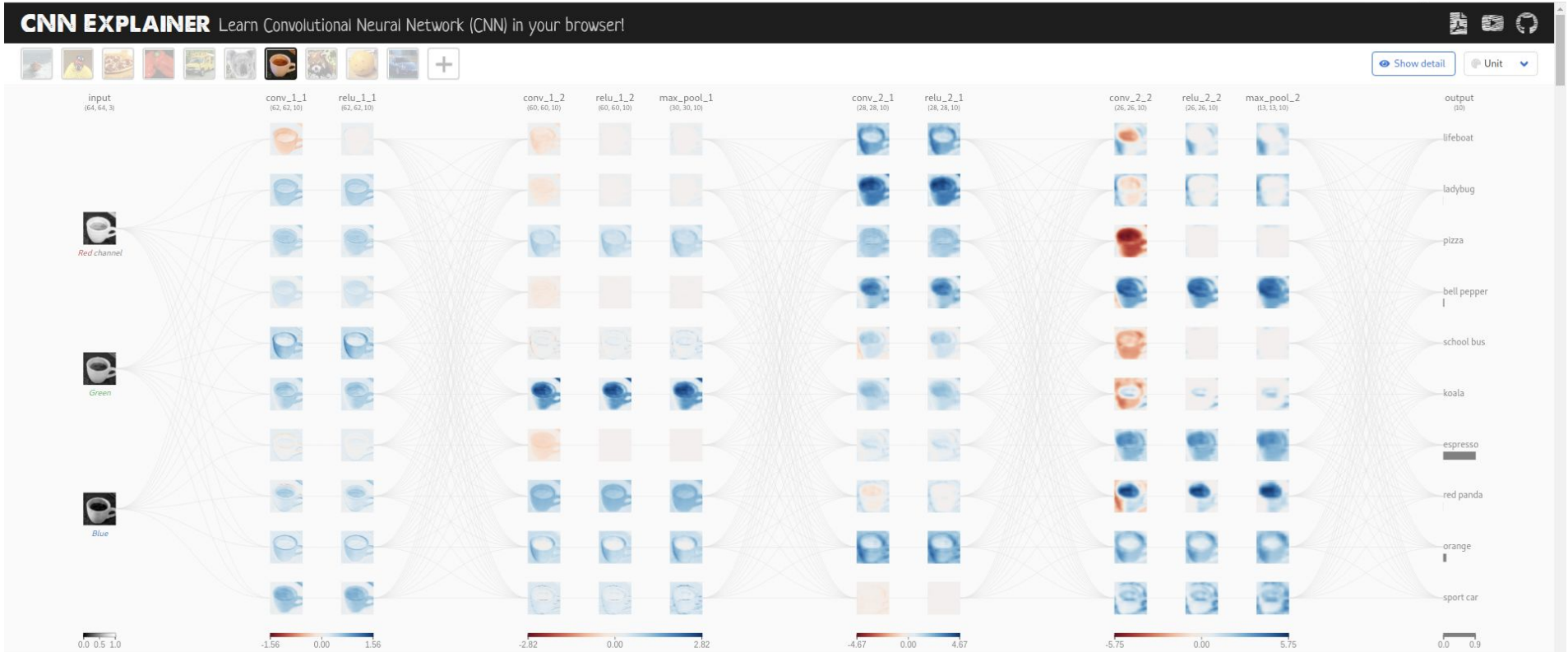
With deep learning, we are searching for a **surjective** (or **onto**) function **f** from a set **X** to a set **Y**.

MNIST - CNN Visualization

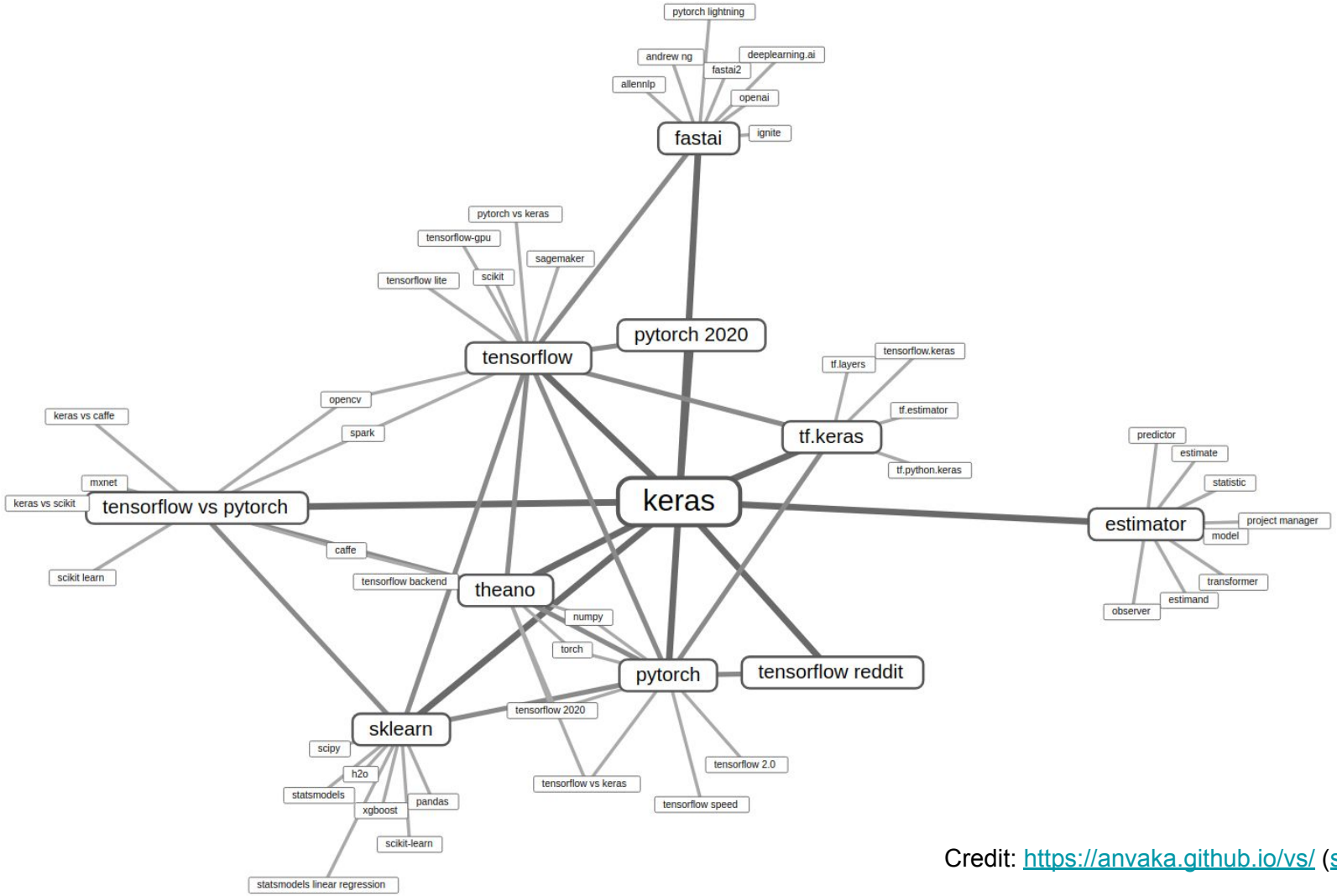


(Image Credit: http://scs.ryerson.ca/~aharley/vis/)

CNN Explainer

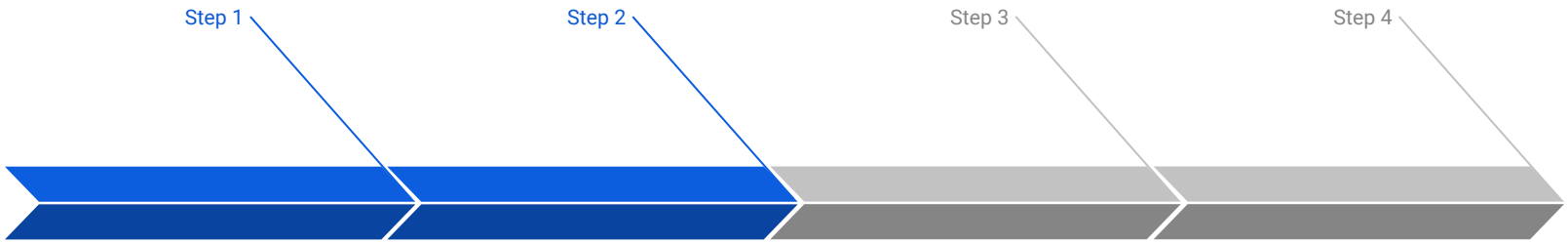


(Image Credit: https://poloclub.github.io/cnn-explainer/)



Credit: https://anvaka.github.io/vs/ (source)

Machine Learning Workflow with Keras



Prepare Train Data

The preprocessed data set needs to be shuffled and splitted into training and testing data.

Define Model

A model could be defined with Keras Sequential model for a linear stack of layers or Keras functional API for complex network.

Training Configuration

The configuration of the training process requires the specification of an optimizer, a loss function, and a list of metrics.

Train Model

The training begins by calling the fit function. The number of epochs and batch size need to be set. The measurement metrics need to be evaluated.