UC Berkeley Extension

Machine Learning With TensorFlow

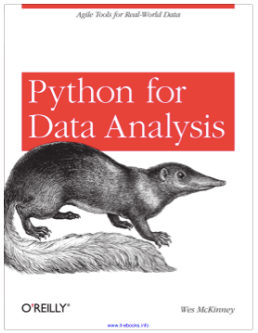
X433.7-001 (2 semester units in COMPSCI)

Instructor Alexander I. Iliev, Ph.D.

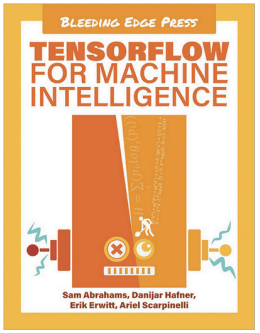
Course Content Outline

- **Optional Reading for the Course**

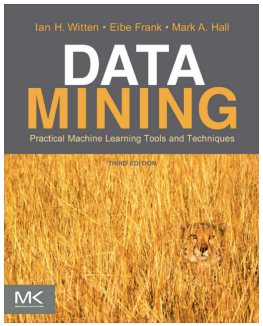
Title: Python for Data Analysis
ISBN-10: 1449319793 and ISBN-13: 978-1449319793
Author(s): Wes McKinney
Publisher: O'Reilly Media
Edition Number: 1 edition
Publication Date: November 1, 2012



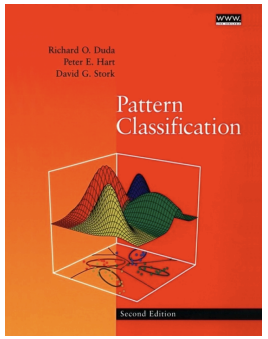
Title: Tensorflow for Machine Intelligence
ISBN:9781939902351, ISBN10:1939902355
Author(s): Sam Abrahams, Danijar Hafner
Erik Erwitt, Ariel Scarpinelli
Publisher: Bleeding Edge Press
Edition: 1 edition
Publication Date: July 2016



Title: Data Mining
ISBN: 978-0-12-374856-0
Author(s): Ian H. Witten, Eibe Frank, Mark A. Hall
Publisher: Elsevier
Edition Number: 3rd edition
Publication Date: 2011
Free Online copy



Title: Pattern Classification
ISBN: 111858600X, 9781118586006
Author(s): Richard O. Duda
Peter E. Hart, David G. Stork
Publisher: John Wiley & Sons, 2012
Edition: 2nd edition
Publication Date: 2012



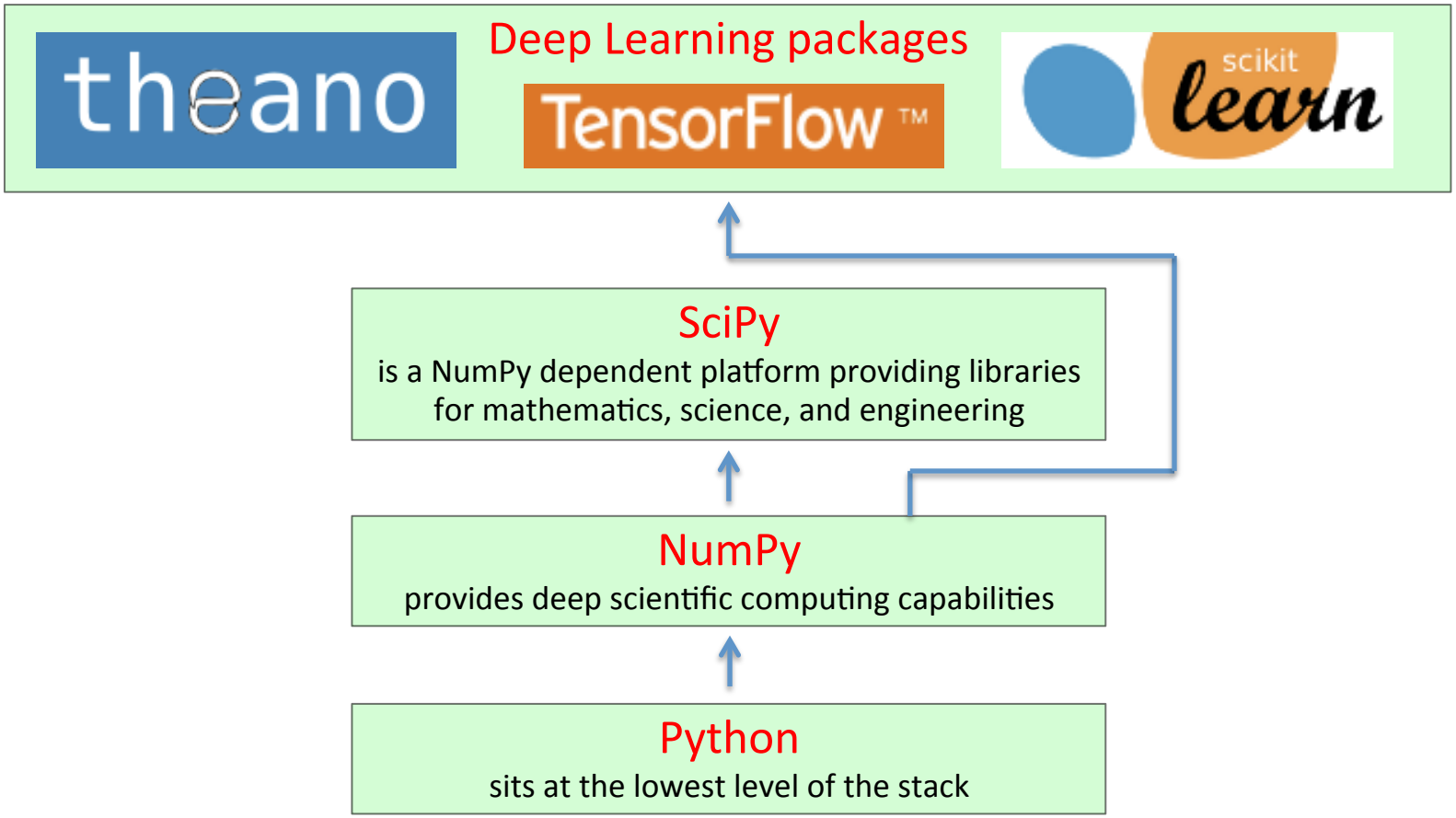
Data Mining

Class 1 …

Python, TensorFlow, installation and general concepts …

Machine Learning With TensorFlow

- Basic stack using NumPy, SciPy and Deep Learning in Python:

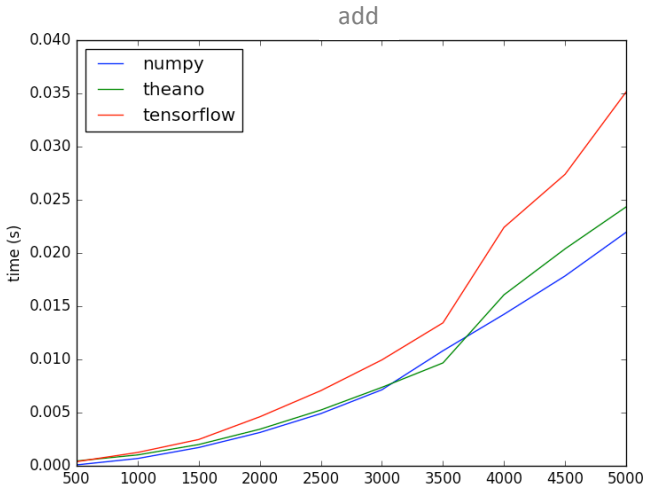


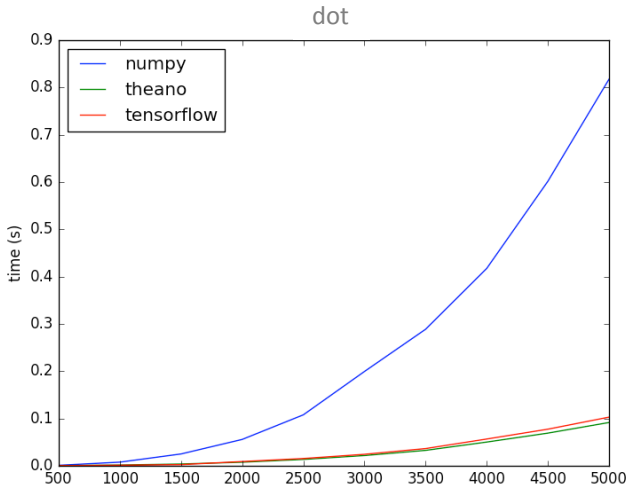
Machine Learning With TensorFlow

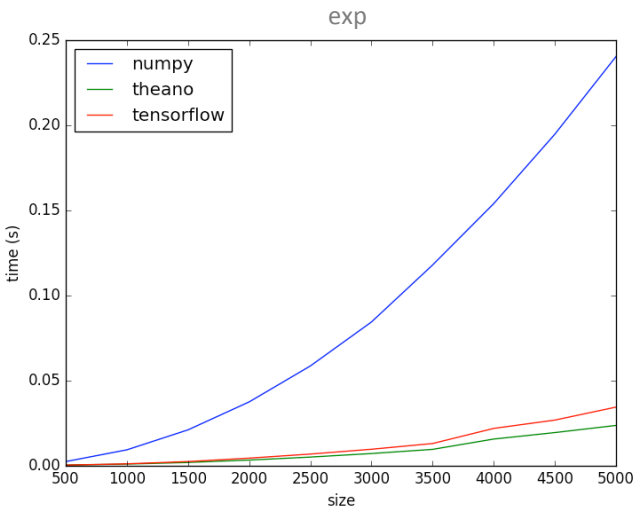
- Python + NumPy + SciPy + Theano + Scikit-Learn + Tensorflow
  - … is like Matlab + toolboxes + DL processing

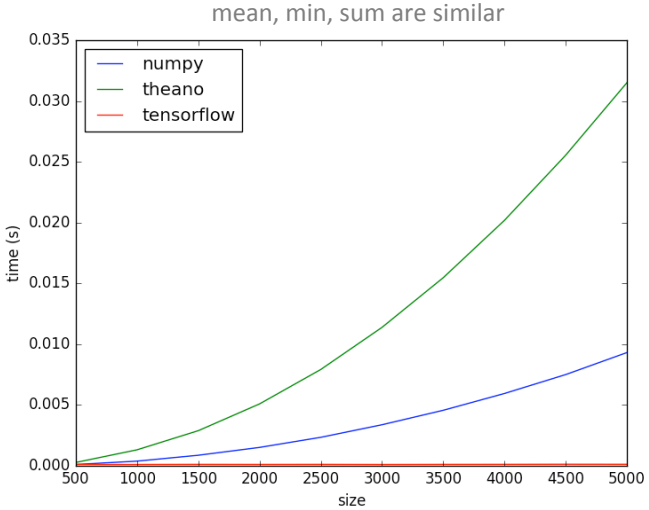
Machine Learning With TensorFlow

- Testing performance for NumPy + Theano + Tensorflow









Machine Learning

- Working with Theano, Scikit-Learn and TensorFlow

  - Theano is a Python library that lets you to define, optimize, and evaluate mathematical expressions, especially ones with multi-dimensional arrays (numpy.ndarray)
  - Speed is rivaling C implementations for problems involving large amounts of data
  - It can surpass C on a CPU by many orders of magnitude by taking advantage of GPUs
  - Theano combines aspects of a computer algebra system (CAS) with optimized compiler
  - It can also generate customized C code for many mathematical operations
  - Theano can minimize the amount of compilation/analysis overhead

Source: http://deeplearning.net/software/theano/introduction.html

Machine Learning

- Working with Theano, Scikit-Learn and TensorFlow

  - Theano's compiler applies many optimizations of varying complexity that include, but are not limited to:
    - use of GPU for computations
    - constant folding
    - merging of similar subgraphs, to avoid redundant calculation
    - arithmetic simplification (e.g. x*y/x -> y, --x -> x)
    - inserting efficient BLAS operations (e.g. GEMM) in a variety of contexts
    - using memory aliasing to avoid calculation
    - using inplace operations wherever it does not interfere with aliasing
    - loop fusion for elementwise sub-expressions
    - improvements to numerical stability (e.g. $\log(1+\exp(x))$ and $\log(\sum_i \exp(x[i]))$)
    - for a complete list, see http://deeplearning.net/software/theano/optimizations

Source: http://deeplearning.net/software/theano/introduction.html

Machine Learning

- Working with Theano, Scikit-Learn and TensorFlow

  - Theano is a Python library that allows you to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently. It features:

  - **tight integration with NumPy** – Use *numpy.ndarray* in Theano-compiled functions
  - **transparent use of a GPU** – Perform data-intensive calculations up to 140x faster than with CPU.(float32 only)
  - **efficient symbolic differentiation**
  - **speed and stability optimizations**
  - **dynamic C code generation** – Evaluate expressions faster
  - **extensive unit-testing and self-verification** – Detect and diagnose many types of errors

Machine Learning

- Working with Theano, Scikit-Learn and TensorFlow

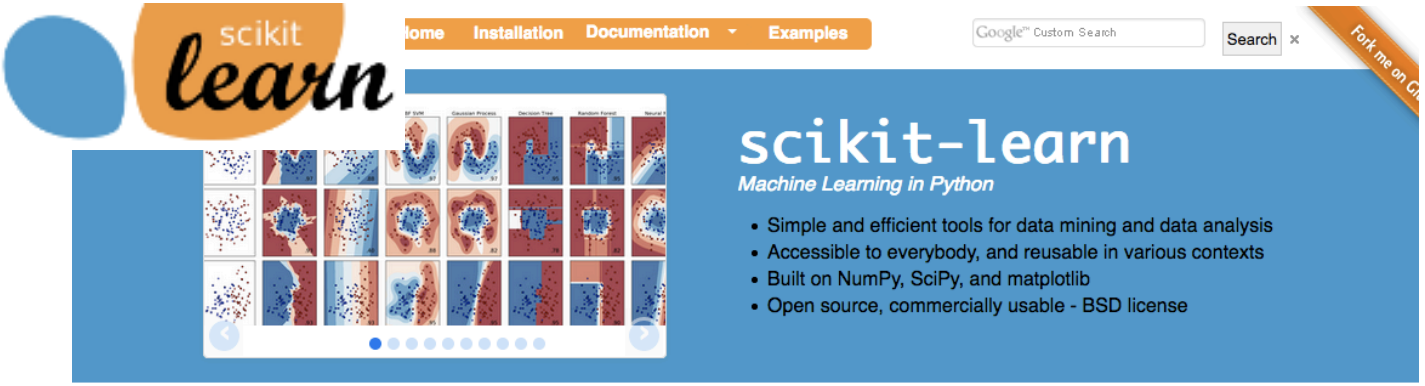
  - Theano and TensorFlow are competing heavily for which one is the best platform
  - Theano outperforms TensorFlow on a single GPU, but TensorFlow outperforms Theano for parallel execution on multiple GPUs
  - Theano has better documentation in general than TensorFlow, although that changes quickly
  - Theano has a native Windows support
  - Both use Numpy arrays
  - TensorFlow is more elegant conceptually cleaner than Theano

Machine Learning

- Working with Theano, Scikit-Learn and TensorFlow
  - **Classification**: Identifying to which category an object belongs to
  - **Regression**: Predicting a continuous-valued attribute associated with an object
  - **Clustering**: Automatic grouping of similar objects into sets
  - **Dimensionality reduction**: Reducing the number of random variables to consider
  - **Model selection**: Comparing, validating and choosing parameters and models
  - **Preprocessing**: Feature extraction and normalization

Machine Learning

- Working with Theano, Scikit-Learn and TensorFlow



Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ... — Examples

Clustering

Automatic grouping of similar objects into sets.

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

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics. — Examples

Preprocessing

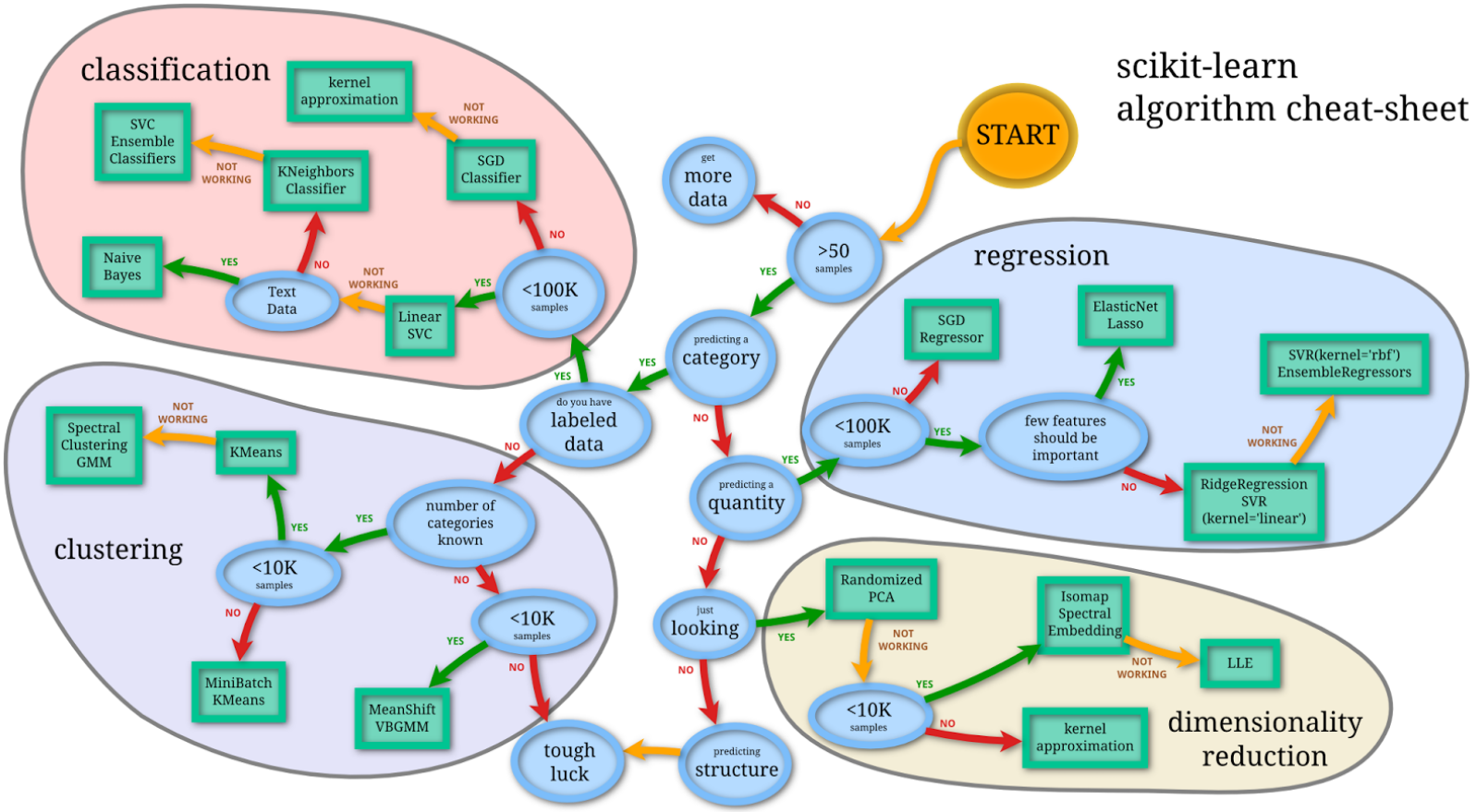
Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction. — Examples

Source: http://scikit-learn.org/stable/

Machine Learning

- Working with Theano, Scikit-Learn and TensorFlow



Machine Learning

- Working with Theano, Scikit-Learn and TensorFlow

  - TensorFlow™ is an open source software library for numerical computation using data flow graphs
  - TensorFlow was developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research
  - Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them
  - The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API

Source: https://www.tensorflow.org/

Machine Learning

- Working with Theano, Scikit-Learn and TensorFlow

  - TensorFlow was originally created by Google as an internal machine learning tool
  - An implementation of it was open sourced under the Apache 2.0 License in November 2015
  - TensorFlow is an interface for numerical computation as described in the TensorFlow white paper, and Google still maintains its own internal implementation of it
  - Google is constantly pushing internal improvements to the public repository
  - The open source release contains the same capabilities as Google's internal version

Machine Learning

- Working with Theano, Scikit-Learn and TensorFlow

  - TensorFlow's API is most similar to Theano's
  - TensorFlow does contain a package, "learn" (AKA "Scikit Flow"), that emulates the one-line modeling functionality of Scikit-Learn
  - TensorFlow provides an extensive suite of functions and classes that allow users to define models from scratch mathematically
  - This allows users with the appropriate technical background to create customized, flexible models quickly and intuitively
  - TensorFlow is the best library to use for both Research and Production because it scales across multiple GPUs better than Theano does

Machine Learning With TensorFlow

- Distributions and Dependencies:

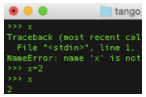
  - Distributions: Python 2.6+, 3.3+
  - Dependencies: Numpy 1.6.1+, Scipy 0.9+

  - Distributions: Python 2.6+, 3.3+
  - Dependencies: Numpy 1.7.1+, Scipy 0.11+

  - Distributions: Python 2.7, 3.4, 3.5
  - Dependencies: Numpy, bazel, six, wheel

Machine Learning With TensorFlow

- Choosing your environment:
  - Using the shell (primarily for Linux and OSX users):



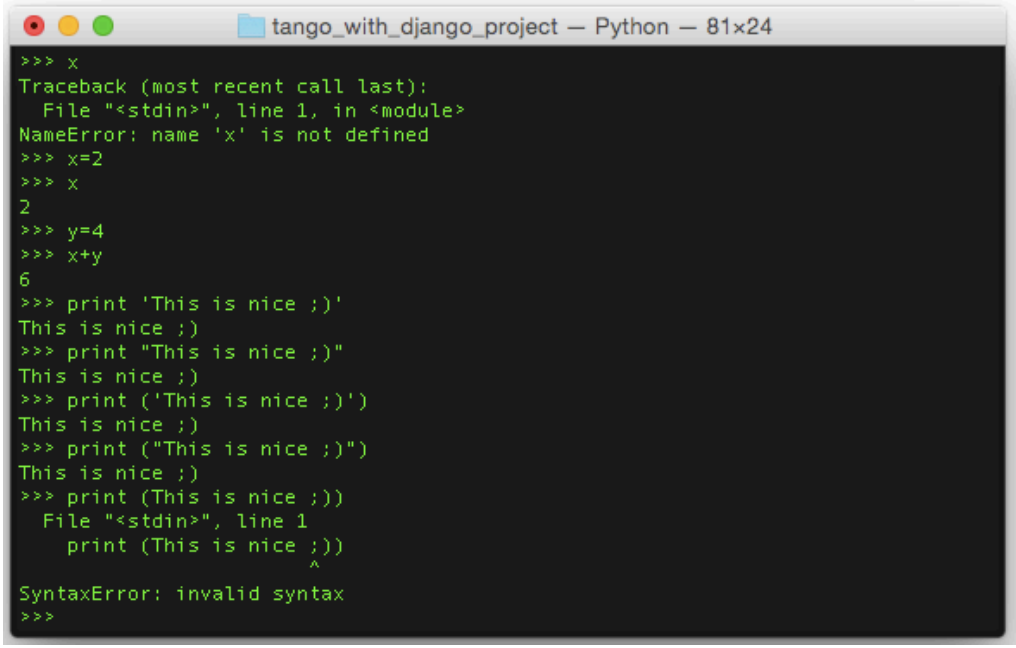
    - Obtaining Python with core packages:
      https://www.python.org/downloads/
    - Obtaining the NumPy & SciPy libraries:
      http://www.scipy.org/scipylib/download.html

  - GUI environments using an interactive editor and debuggers for Python:

    - Pyzo – a free open-source environment based on Python:
      http://www.pyzo.org/downloads.html
    - Anaconda – Anaconda is the leading open data science platform powered by Python:
      https://www.continuum.io/
    - Canopy – Python based environment providing core scientific analytic and scientific Python packages, for scientific development and visualization:
      https://store.enthought.com/

Machine Learning With TensorFlow

- Working with the shell
  - The shell:
    - present on OSX, Unix and Linux systems
    - provides a quick and easy to use plain text environment for swift execution of simple commands
    - it is flexible and lightweight
    - it has a Unix/Linux look and feel and is not used only for Python and it's packages and modules



Machine Learning With TensorFlow

- Working with the shell
  - Creating the Virtualenv environment:
    - to keep our dependencies nice and clean, we're going to be using virtualenv to create a virtual Python environment
    - first, we need to make sure that Virtualenv is installed along with pip, Python's package manager
    - run the following commands (on Mac OS X):

```
$ sudo easy_install pip
$ sudo pip install --upgrade virtualenv
```

    - create a directory to contain this environment:

```
$ sudo mkdir ~/env
```

Machine Learning With TensorFlow

- Working with the shell
  - Creating the Virtualenv environment:
    - now we'll create the environment using either the:
      - virtualenv command in Python 2 or
      - venv module built in Python 3, both located in ~/env/tensorflow

```
# Python 2.7
$ virtualenv --system-site-packages ~/env/tensorflow
# Python 3
$ sudo python3 -m venv --system-site-packages ~/env/tensorflow
```

    - Once it has been created, we can activate the environment using the source command:

```
$ source ~/env/tensorflow/bin/activate
# Notice that your prompt now has a '(tensorflow)' indicator
(tensorflow)$
```

Machine Learning With TensorFlow

- Working with the shell
  - Creating the Virtualenv environment:
    - make sure that the environment is active when we install anything with pip, since that is how Virtualenv keeps track of various dependencies
    - when done, shut it off by using the deactivate command:

```
(tensorflow)$ deactivate
```

    - create a shortcut for activating it by creating alias to your ~/.bashrc file:

```
$ sudo printf '\nalias tensorflow="source ~/env/tensorflow/bin/
activate"' >> ~/.bashrc
```

    - restart your bash:

```
:~ alex$ . ~/.bashrc
```

    - now test it:

```
$ tensorflow
# The prompt should change, as before
(tensorflow)$
```

Machine Learning With TensorFlow

- Working with the shell
  - Installing TensorFlow:
    - make sure that your Virtualenv environment from the previous section is active and run the following command corresponding to your operating system (Mac OS X) and version of Python:

```
# Mac OS X, Python 2.7:
(tensorflow)$ pip install --upgrade https://storage.googlea-
pis.com/tensorflow/mac/tensorflow-0.9.0-py2-none-any.whl

# Mac OS X, Python 3.4+
(tensorflow)$ pip3 install --upgrade https://storage.googlea-
pis.com/tensorflow/mac/tensorflow-0.9.0-py3-none-any.whl
```

Machine Learning With TensorFlow

- Working with the shell
  - Troubleshooting:
    - firstly, you might want to upgrade your pip:

      ```
      sudo pip install --upgrade pip
      ```

    - you might get the following message, despite that you "Successfully installed six-1.11.0 tensorflow-1.6.0" in Virtualdev:

      Illegal instruction: 4 when importing tensorflow in python 3.6

    - there is a bug in TF1.6, so change the TensorFlow version by typing:

      >>> `sudo pip install -Iv tensorflow==1.5`

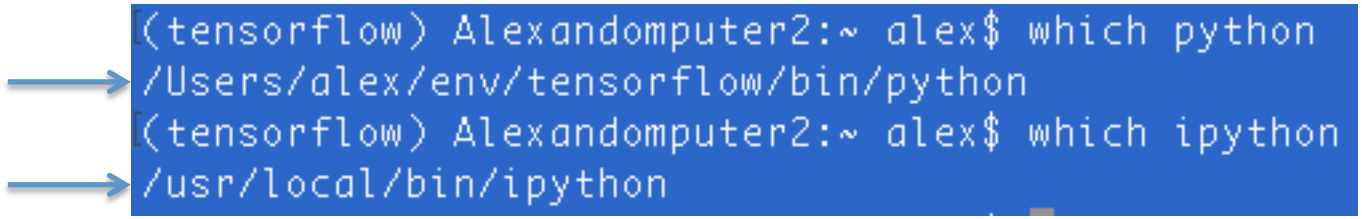
      >>> `sudo pip install -Iv numpy==1.13` → *… may not be necessary*

    - you can also try to uninstall TensorFlow first, but it may not be necessary:

      >>> `sudo pip uninstall tensorflow` → *… you may use pip3 too*

Machine Learning With TensorFlow

- Working with the shell
  - Running TensorFlow on iPython:
    - in case you'd like to tun TensorFlow in iPython you may have to check where TensorFlow is running from:



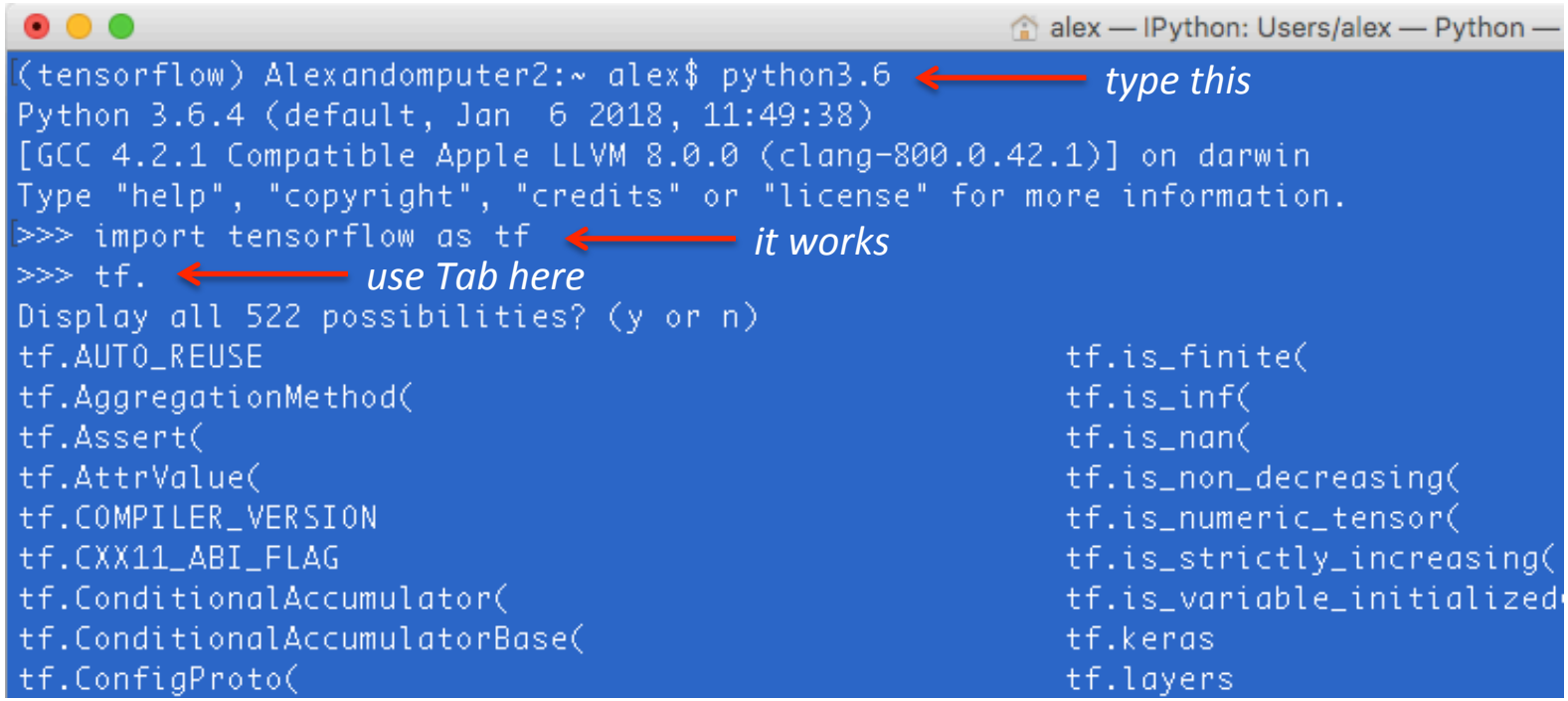
    - Since they run from different locations, iPython may not have the path to run TensorFlow so it is best to reinstall iPython like this in virtualenv:

```
(tensorflow) Alexandomputer2:~ alex$ sudo pip3 install --ignore-installed ipython
```

    - You can also try to change the path too

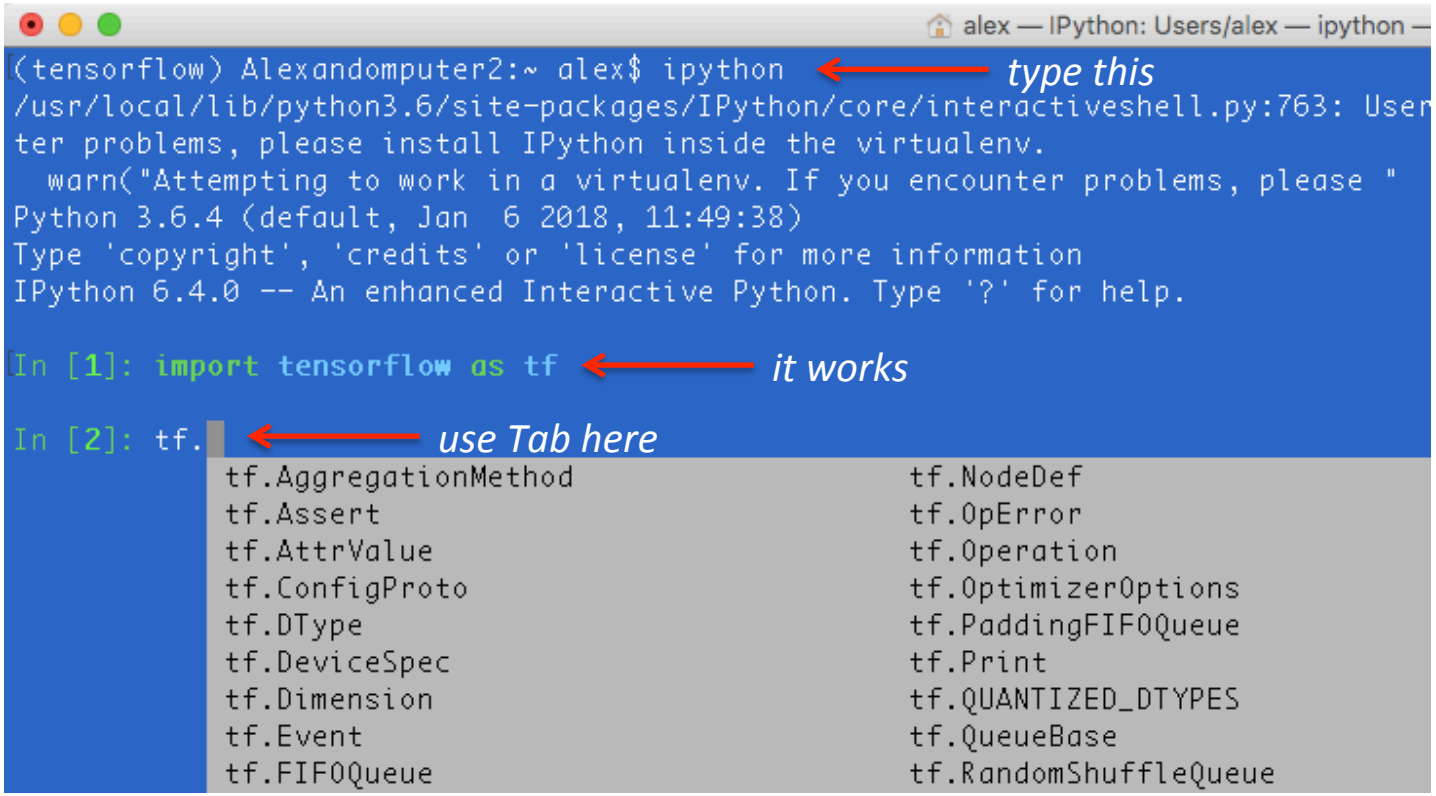
Machine Learning With TensorFlow

- Working with the shell
  - Try TensorFlow on virtualenv and Python3.6:



Machine Learning With TensorFlow

- Working with the shell
  - Try TensorFlow on virtualenv and iPython:



Machine Learning With TensorFlow

- Working with the shell
  - To see the available packages on virtualenv and iPython:

```
In [2]: pip list

The following command must be run outside of the IPython shell:

    $ pip list
```

this will not work, so we use 'help' ...

  - To check all available modules we type either:

```
In [3]: help('modules')

Please wait a moment while I gather a list of all available modules...
```

we see ... `numpy` ... `matplotlib` ... `tensorflow` ... `tensorboard`

  - Or:

```
In [4]: import
IPython                         alembic
abc                             antigravity
absl                            appnope
aifc                            argparse
```

this is a tab completion

Machine Learning With TensorFlow

- Working with the shell
  - Installing Jupyter notebook:
    - First, make sure you have installed iPython (on both Python 2 and 3):

```
# Python 2.7
$ sudo python2 -m pip install ipykernel
$ sudo python2 -m ipykernel install
# Python 3
$ sudo python3 -m pip install jupyterhub notebook ipykernel
$ sudo python3 -m ipykernel install
```

    - Then install all dependencies:

```
$ sudo apt-get install build-essential
```

    - use pip to install the Jupyter Notebook (or pip3 for Python 3):

```
# For Python 2.7
$ sudo pip install jupyter
# For Python 3
$ sudo pip3 install jupyter
```

jupyter

Machine Learning With TensorFlow

- Working with the shell
  - Installing Matplotlib and final check:
    - You should be able to see Jupyter in your environment:

```
In [3]: help('modules')

Please wait a moment while I gather a list of all available modules...
```

... jupyter

    - Then install Matplotlib:

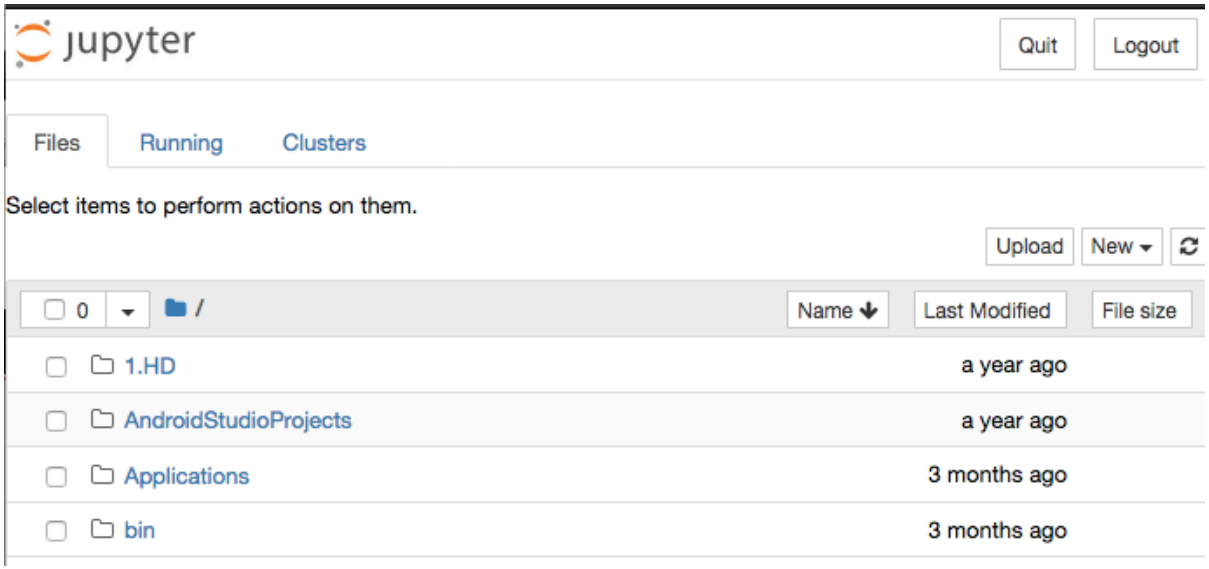
```
# Python 2.7
$ sudo apt-get build-dep python-matplotlib python-tk
# Python 3
$ sudo apt-get build-dep python3-matplotlib python3-tk
```

    - Now check if it works:

```
(tensorflow)$ jupyter notebook
```

Machine Learning With TensorFlow

- Working with the shell
  - Testing your Jupyter environment:
    - You should get a page like this in your browser:



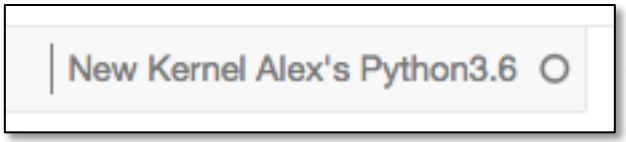
    - With URL: http://localhost:8888/tree

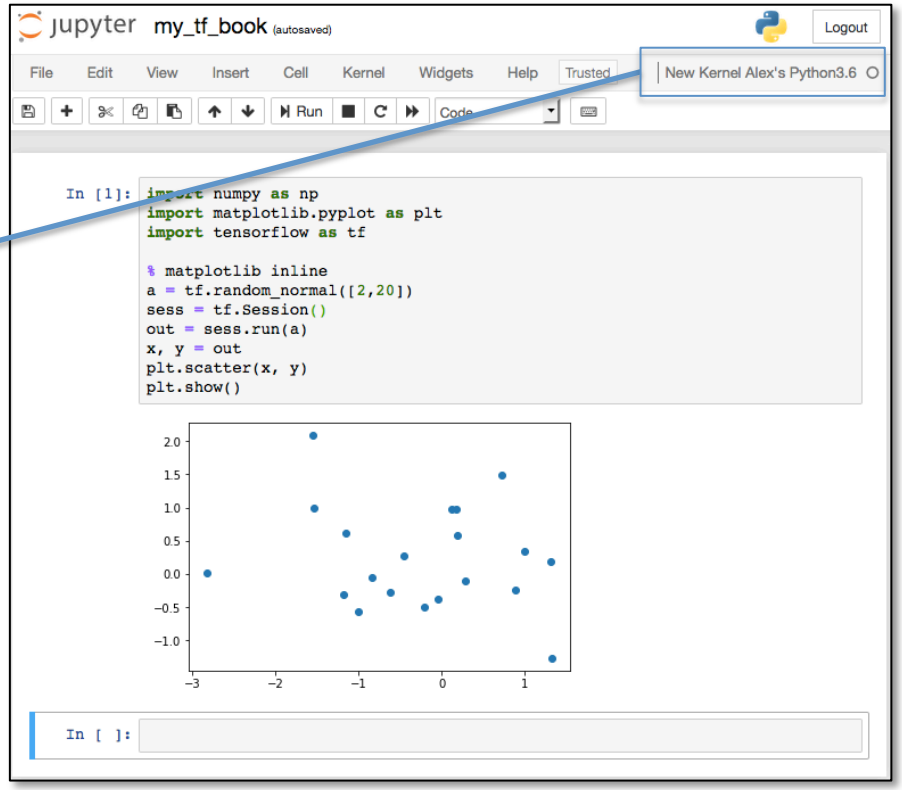
Machine Learning With TensorFlow

- Working with the shell
  - Testing your Jupyter environment:
    - You will be prompted to login:


```
alex$ jupyter notebook --NotebookApp.token=qwerty
```

    - Then run any file you might have:



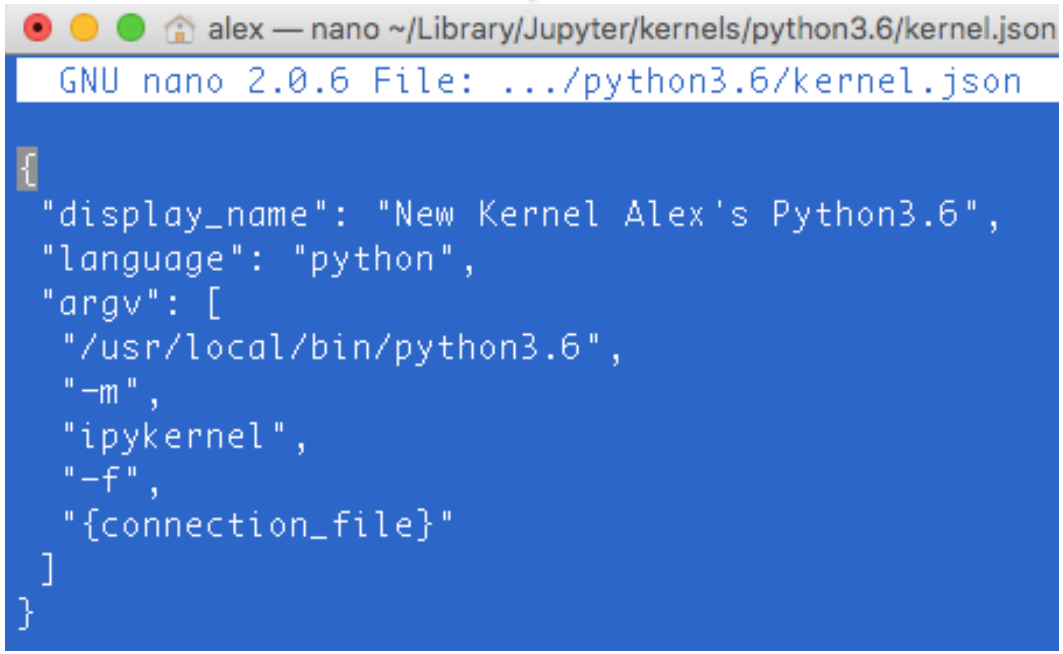


… see next slide

Machine Learning With TensorFlow

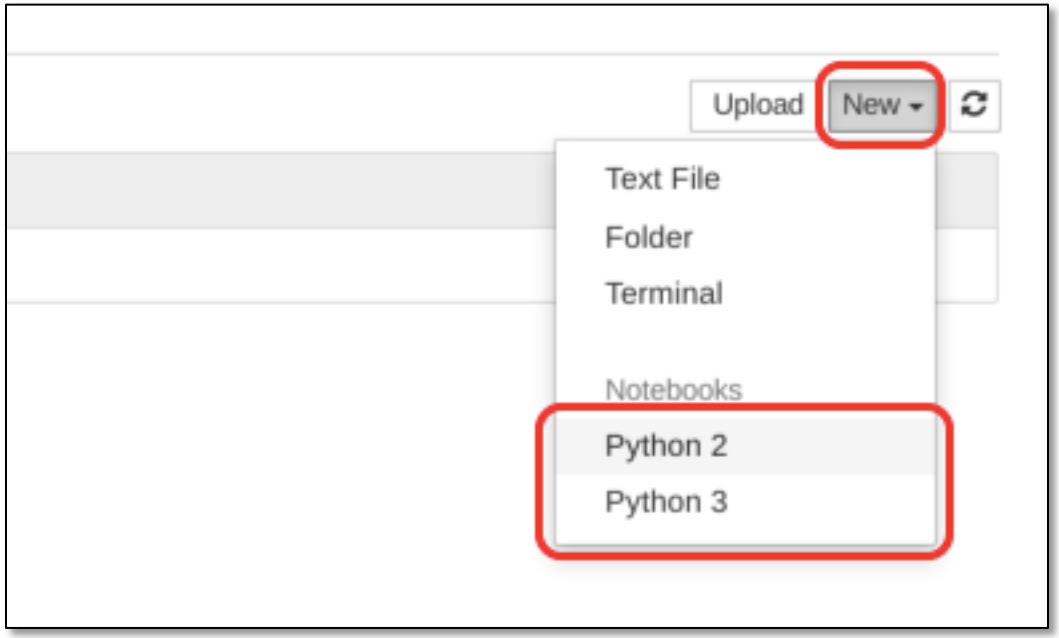
- Working with the shell
  - Testing your Jupyter environment:
    - Here is how to customize your preferred kernel, its source and name:

```
(tensorflow) Alexandomputer2:~ alex$ nano ~/Library/Jupyter/kernels/python3.6/kernel.json
```



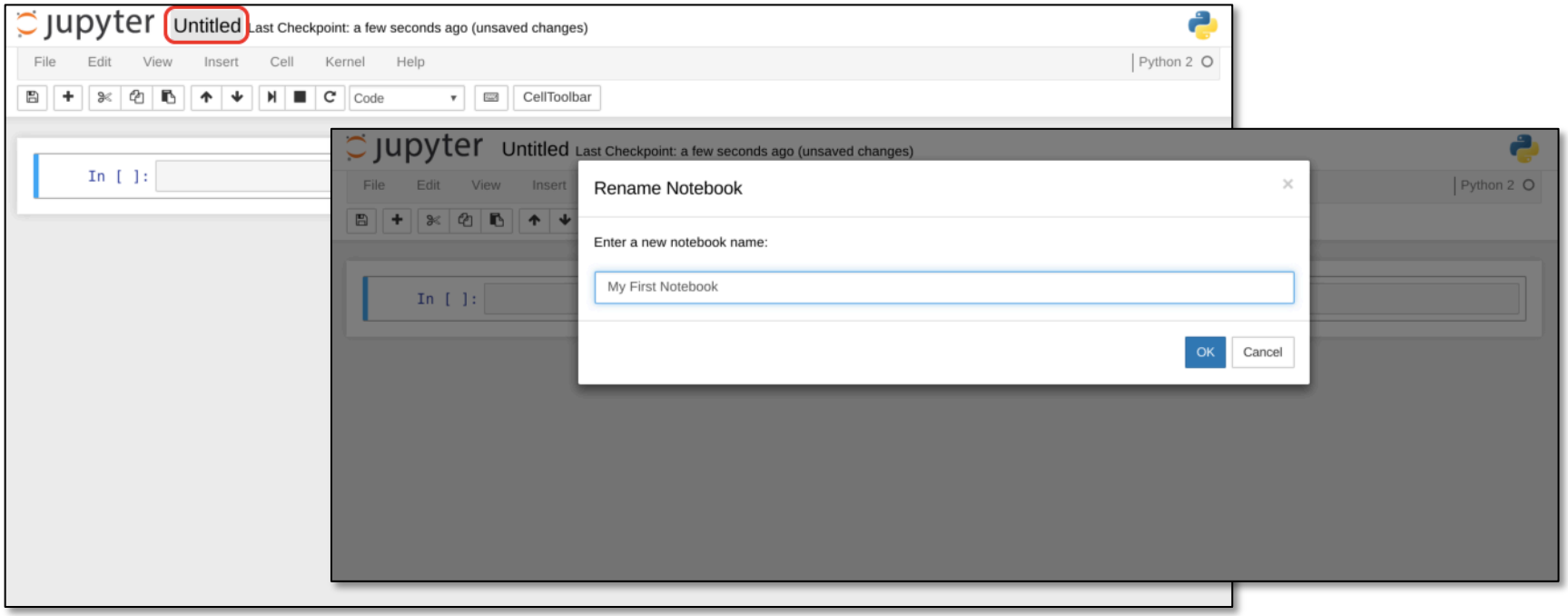
Machine Learning With TensorFlow

- Working with the shell
  - Testing your Jupyter environment:
    - You can start a new file using a different Python version (2 or 3):



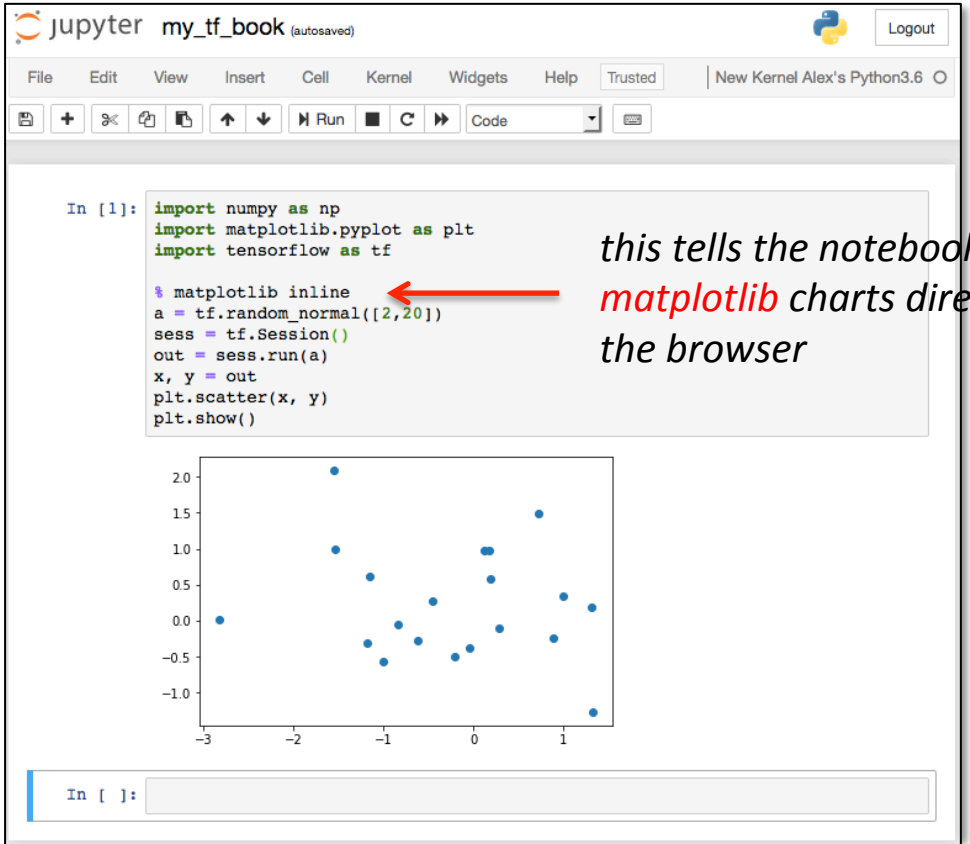
Machine Learning With TensorFlow

- Working with the shell
  - Testing your Jupyter environment:
    - Then you can begin working with it after giving it a name:



Machine Learning With TensorFlow

- Working with the shell
  - Testing your Jupyter environment:
    - You can type the code directly in the field or cut and paste it:



this tells the notebook to display matplotlib charts directly inside the browser

Machine Learning With TensorFlow

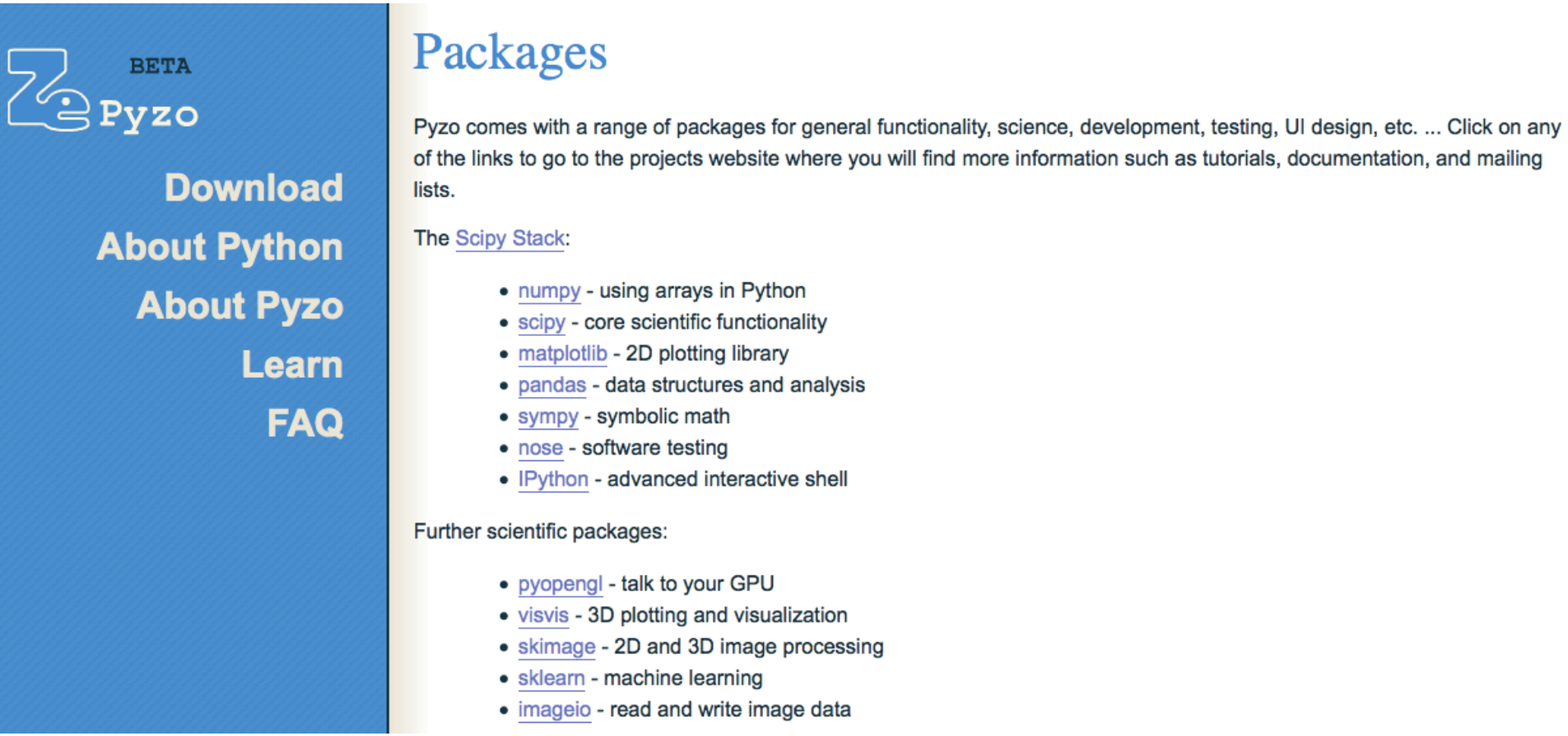
- Working with Pyzo

PYZO: PYTHON TO THE PEOPLE

  - Pyzo … what?
    - ***Pyzo*** is a free Open-source BSD-licensed
    - It is an easy to use GUI computing environment based on Python 3
    - It is built to make Python's potential for scientific and engineering computing to be easily accessible
    - Pizo is combined with IEP to make a powerful easy to use comprehensive environment for scientific computing and visualization
    - Pyzo is a comprehensive development environment pre-built with:
      - many scientific packages including NumPy, SciPy and Matplotlib
      - uses IEP as the front-end IDE
      - IPython and a notebook
    - Complete packages for Pyzo are given here: http://www.pyzo.org/packages.html

Machine Learning With TensorFlow

- Working with Pyzo



source: http://www.pyzo.org/packages.html#packages

Machine Learning With TensorFlow

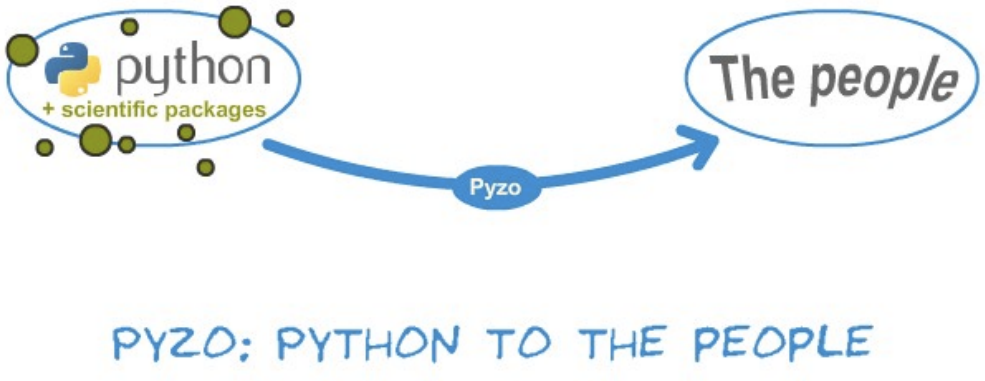
- Working with Pyzo
  - Useful magic commands:

Linux-like:

- pwd
- ls
- mkdir
- rmdir
- whos
- dir()
- cd
- history

Python IEP/Pyzo specific:

- edit *file*
- del(parameter)
- reset
- reset -f
- run *file*
- time
- colors (nocolor,linux,lightbg)
- dhist



Tip: Using a magic command followed by '?' will give you help for it. Example: 'cd?'

magic comprehensive list: http://ipython.org/ipython-doc/3/interactive/magics.html

Machine Learning With TensorFlow

- Working with Pyzo
  - Troubleshooting (*recall*)
    - you might get the following message, while trying to import tensorflow in Pyzo:

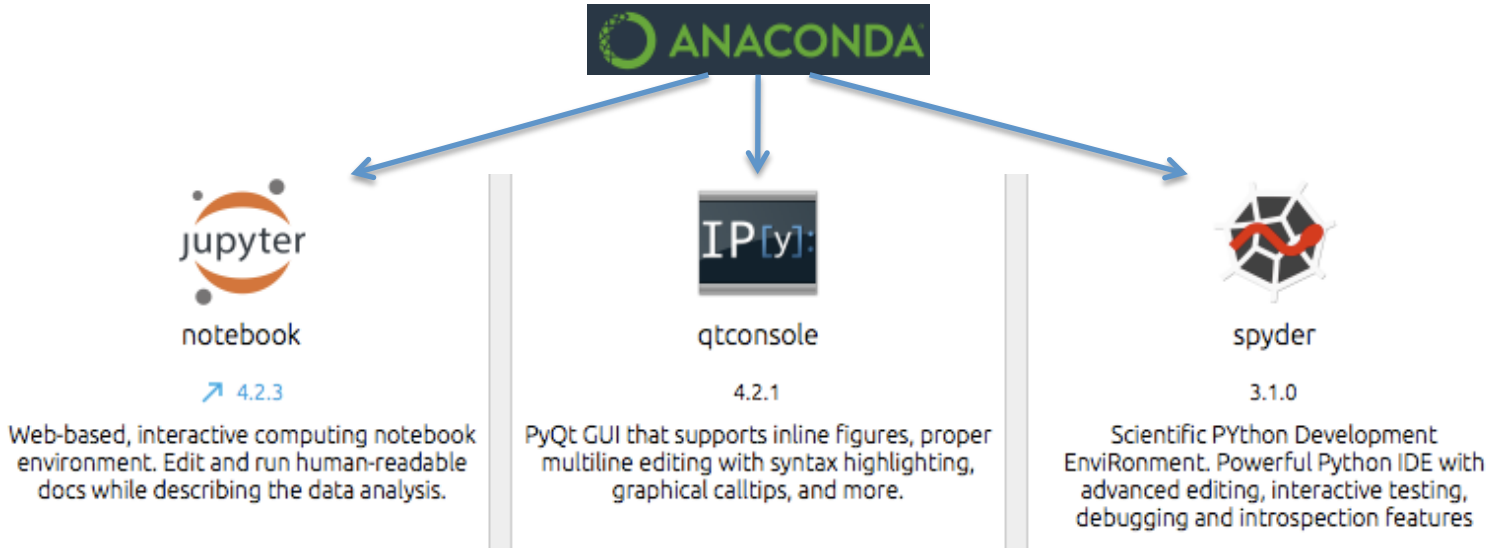
      Illegal instruction: 4 when importing tensorflow in python 3.6

    - Just as before, there is a bug in TensorFlow 1.6, so change the TensorFlow version by typing:

      >>> `sudo pip install -Iv tensorflow==1.5`

Machine Learning With TensorFlow

- Choosing your environment:
  - GUI environments using an interactive editor and debuggers for Python:
    - Anaconda – Anaconda is the leading open data science platform powered by Python: https://www.continuum.io/

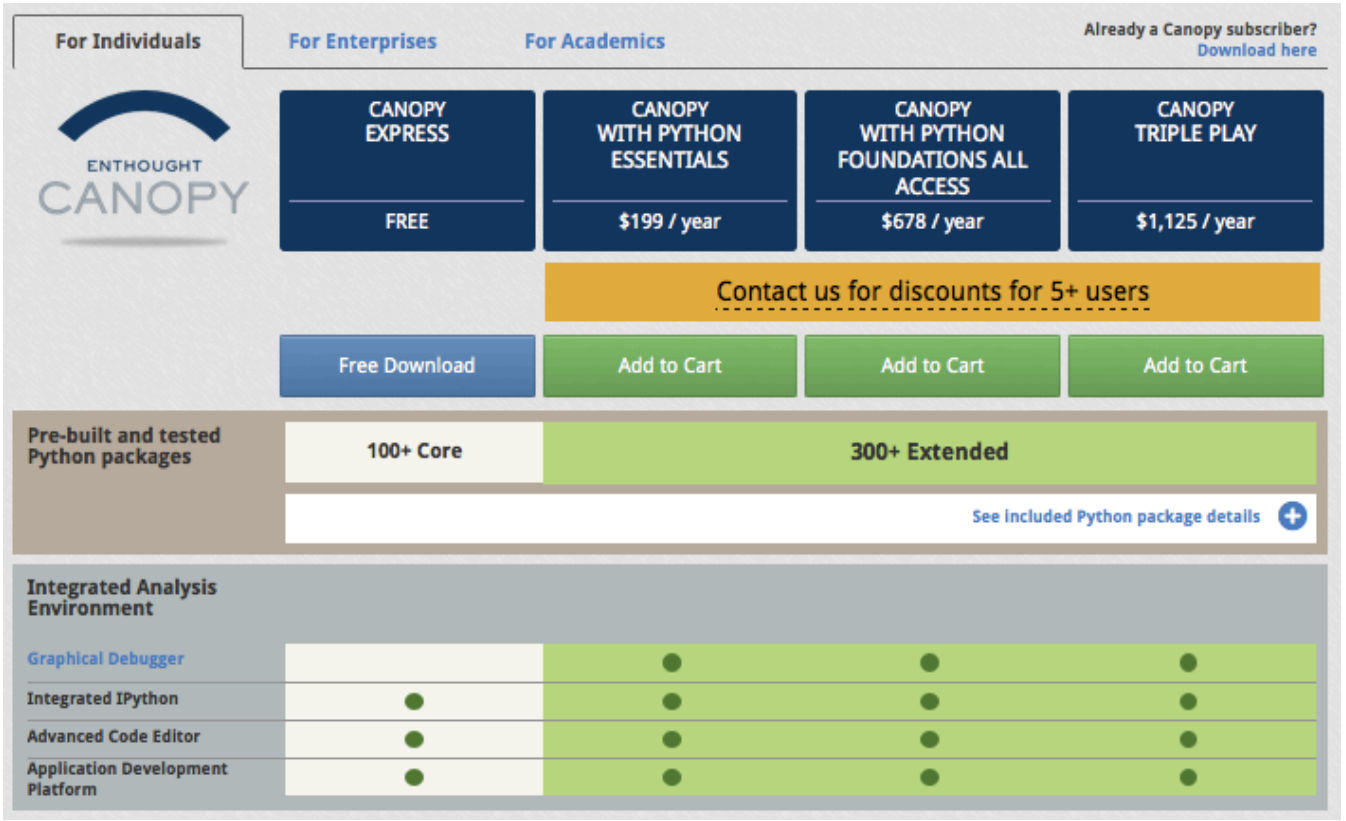


Machine Learning With TensorFlow

- Working with alternative environments:
  - Canopy – scientific and analytic Python package distribution. has a free version, but limited

It offers:

- object-oriented prog.
- MATLAB-like syntax
- many packages
- graphical debugger is NOT free



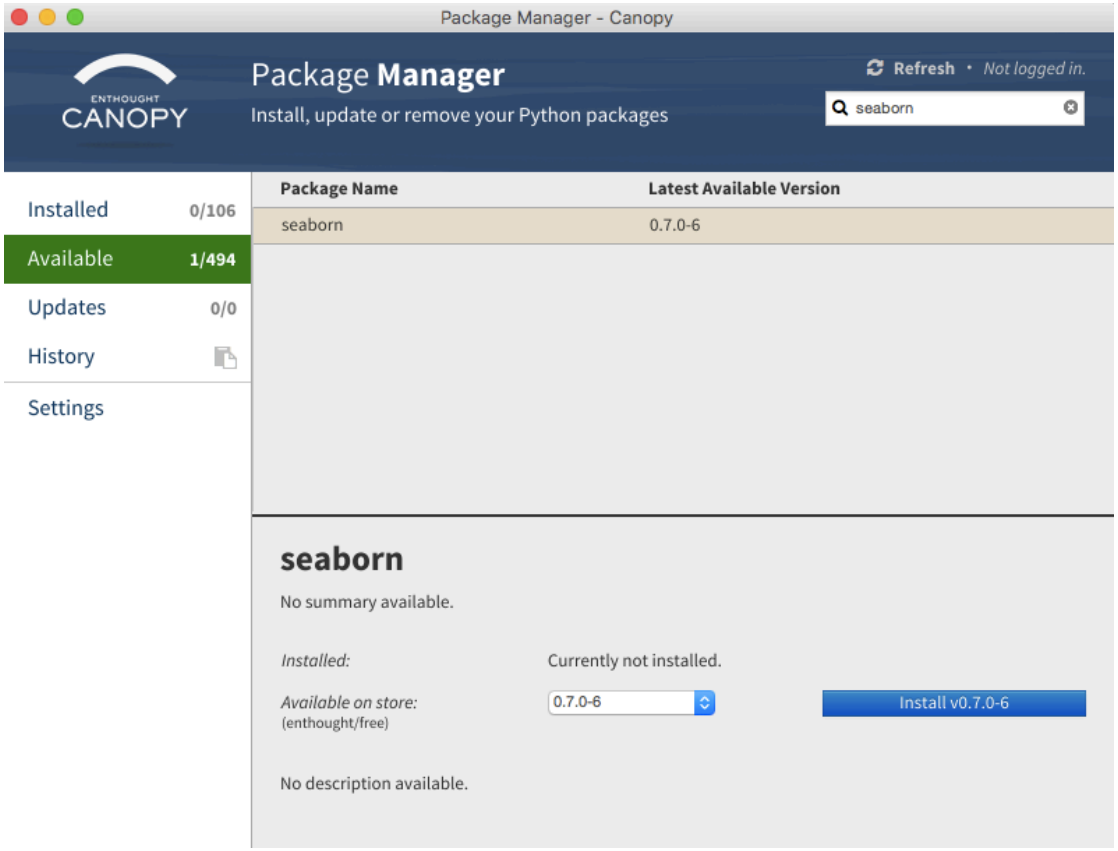
source: https://store.enthought.com/

Machine Learning With TensorFlow

- Working with alternative environments:
  - Canopy – scientific and analytic Python package distribution. has a free version, but limited

Package Manager:

- Packages are installed through Package Manager



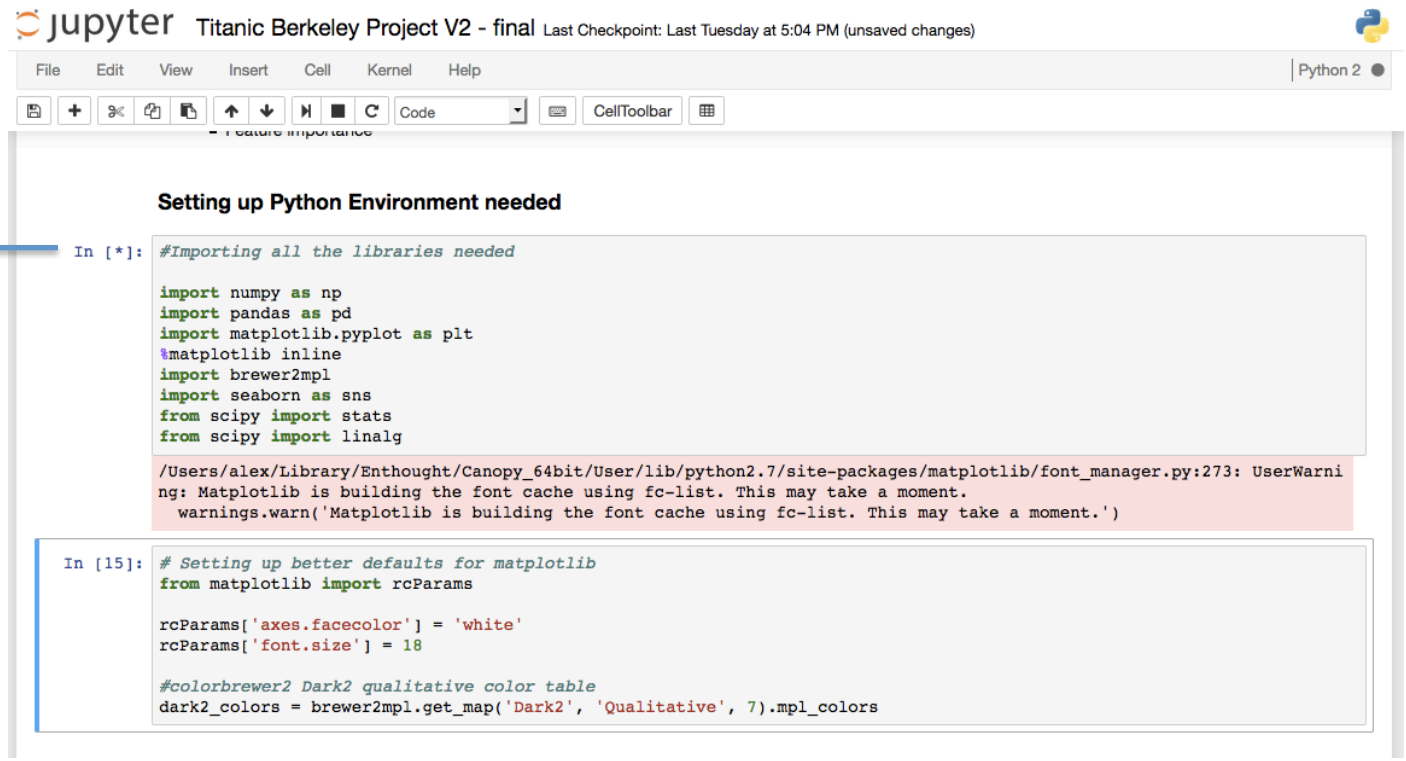
source: https://store.enthought.com/

Machine Learning With TensorFlow

- Working with alternative environments:
  - Canopy – scientific and analytic Python package distribution. has a free version, but limited

Using the browser

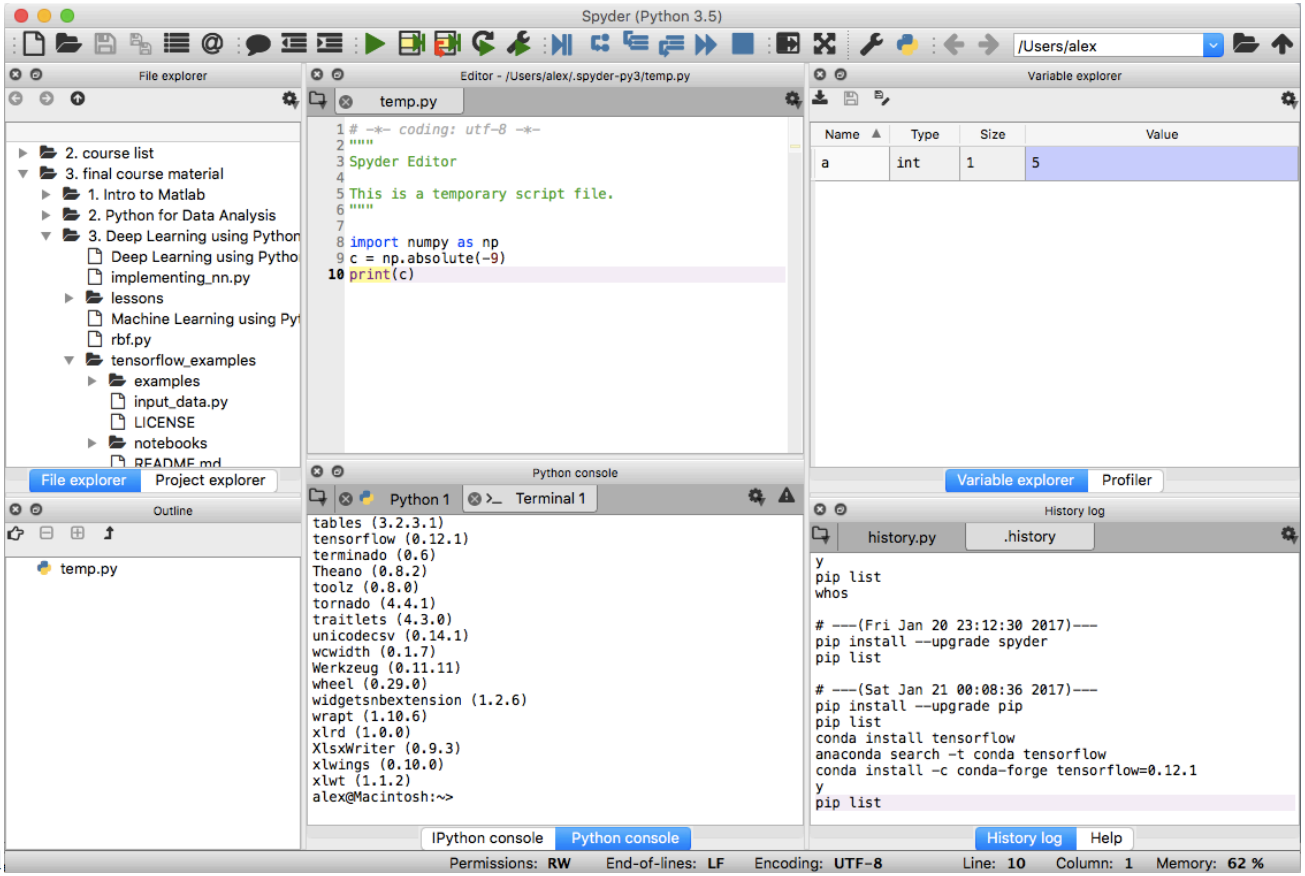
- [*] the line is in a process of executing
- Execute your code sequentially using Shift+Enter



source: https://store.enthought.com/

Machine Learning With TensorFlow

- Choosing your environment:
  - GUI environments using an interactive editor and debuggers for Python:
    - Spyder – a GUI environment for interactive Python use with shells (terminal- and Qt-based):



Machine Learning With TensorFlow

- Working with Canopy
  - Useful magic commands:

Linux-like:

- pwd
- ls
- mkdir
- rmdir
- whos
- who()
- dir()
- cd
- history

iPython specific:

- edit *file*
- del(parameter)
- reset
- reset -f
- run *file*
- time
- dhist

How to create an alias:

- [35] %alias clc clear

Tips: 1. Using a magic command followed by '?' will give you help for it. Example: 'cd?'

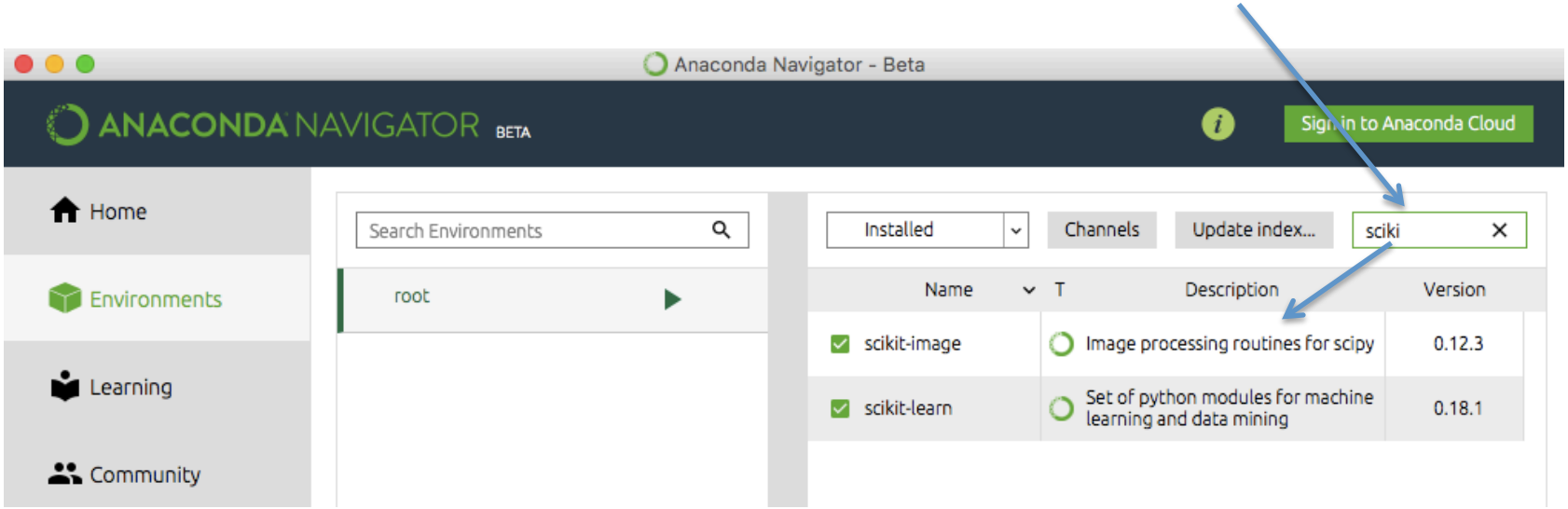
2. Esc will get rid of the help menu

magic comprehensive list: http://ipython.org/ipython-doc/3/interactive/magics.html

Machine Learning

- Working with Anaconda Navigator:

  - Scikit-learn comes installed with Anaconda distribution by default:

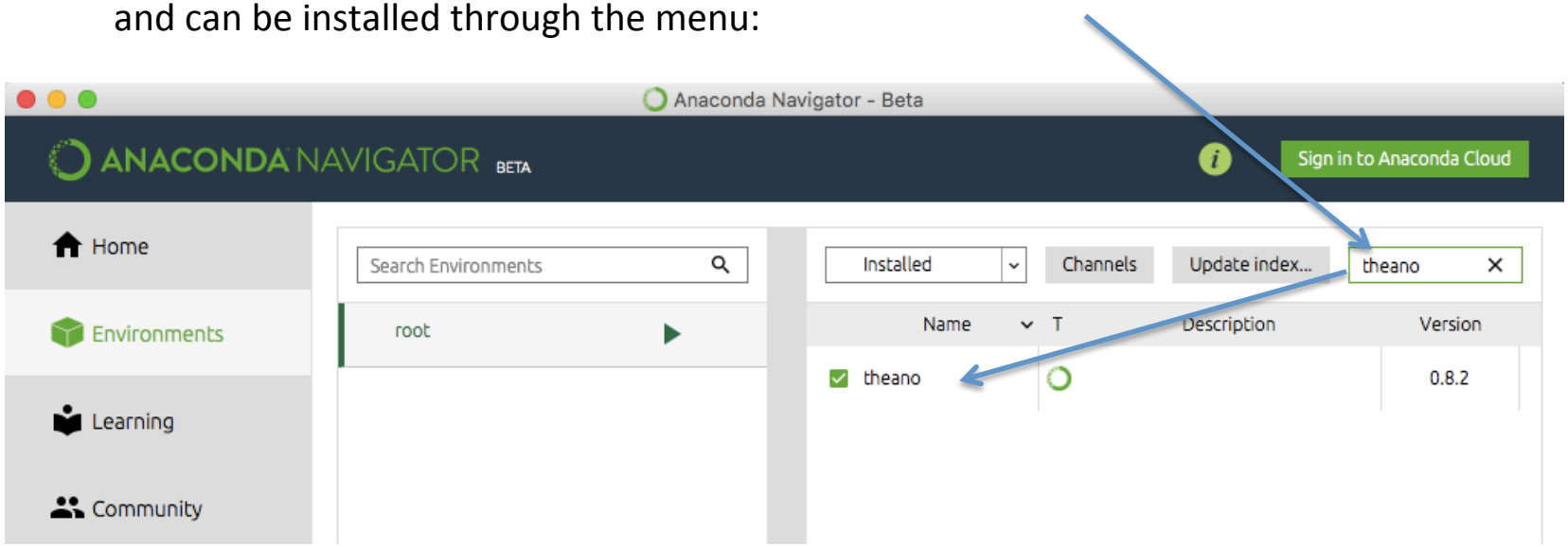


Machine Learning

- Working with Anaconda Navigator:

theano

  - Theano is not installed in Anaconda by default, but is part of the standard repository and can be installed through the menu:

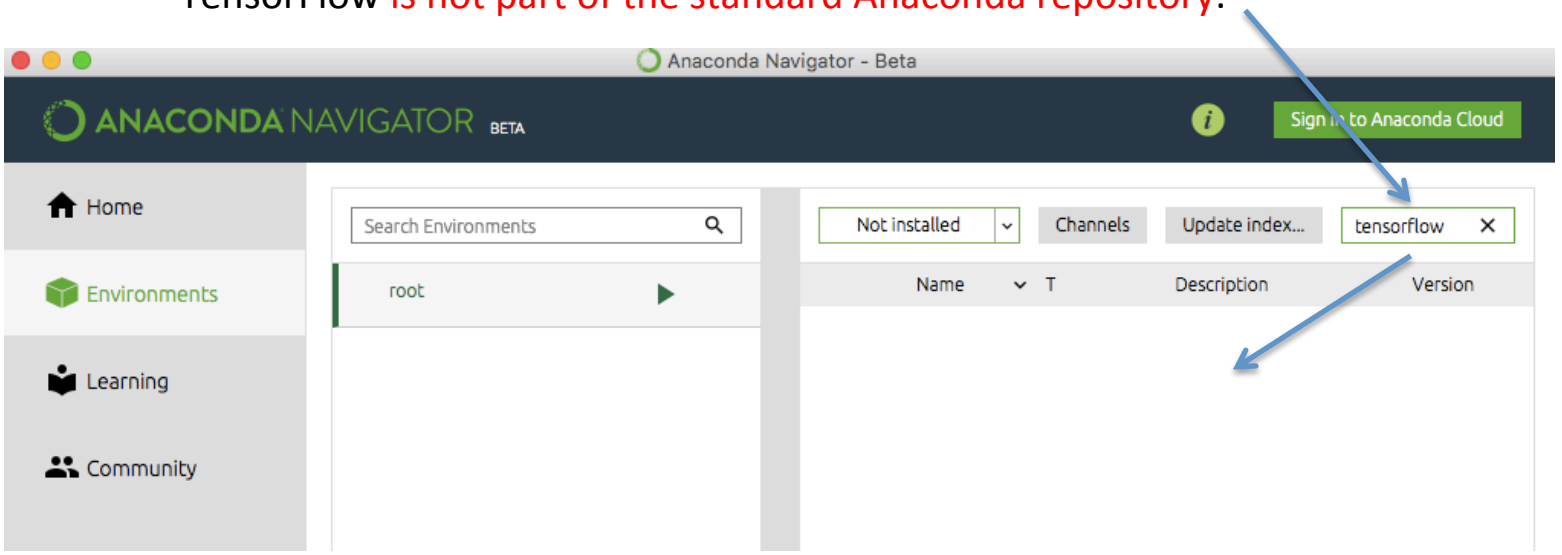


Machine Learning With TensorFlow

- TensorFlow in Anaconda:

TensorFlow™

  - TensorFlow is not part of the standard Anaconda repository:



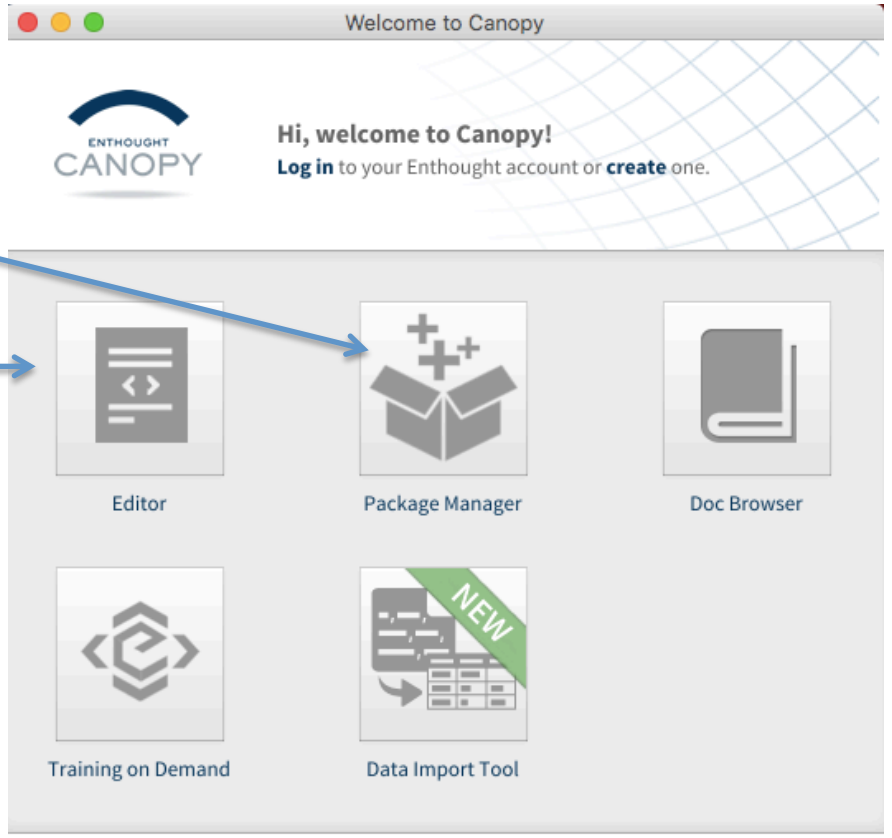
  - So you can install it via Spyder's terminal by typing:
    >>> conda install -c conda-forge tensorflow=0.12.1

Machine Learning With TensorFlow

- TensorFlow in Canopy:

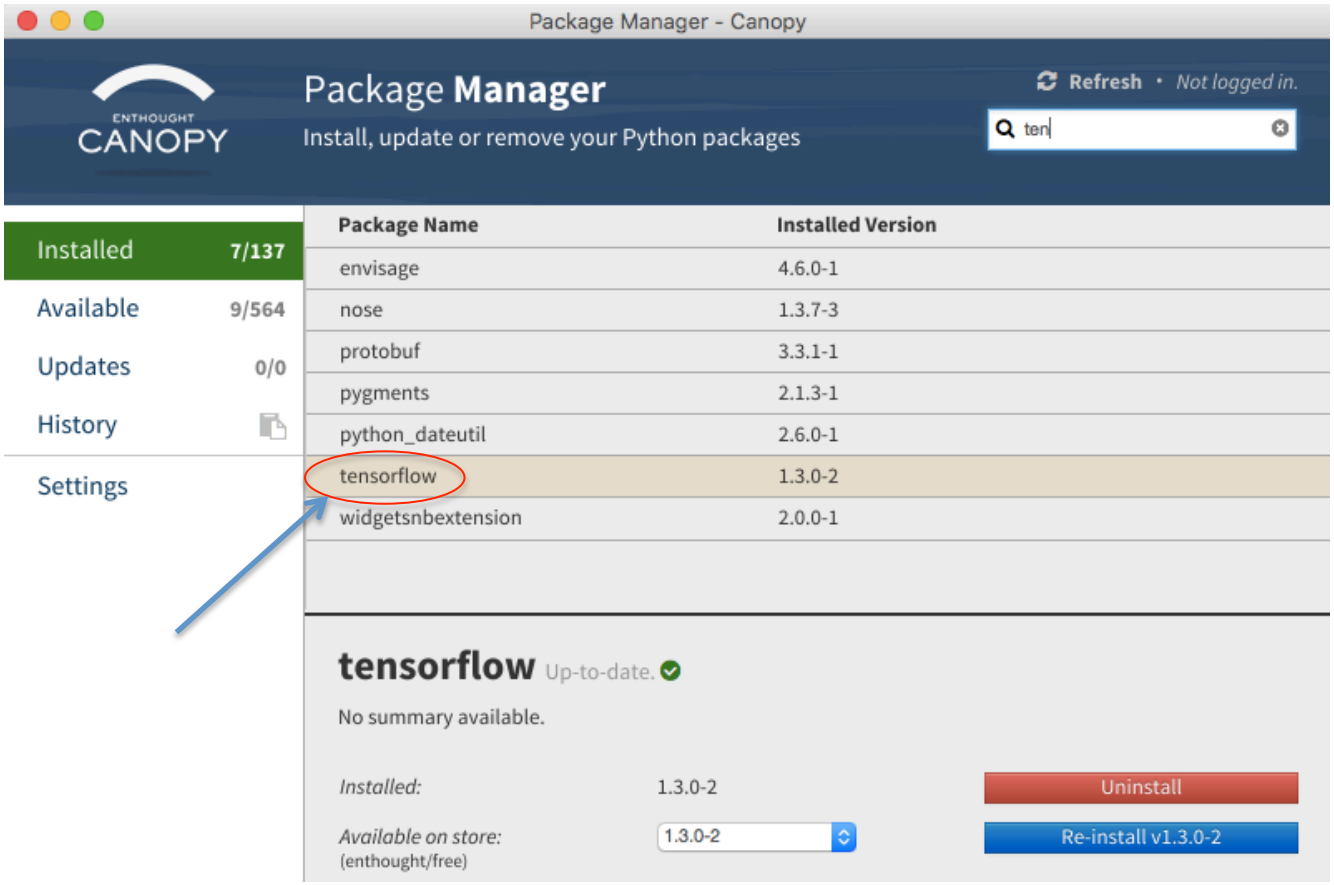
TensorFlow™

1. Install packages from Canopy repository
2. Projects are created here



Machine Learning With TensorFlow

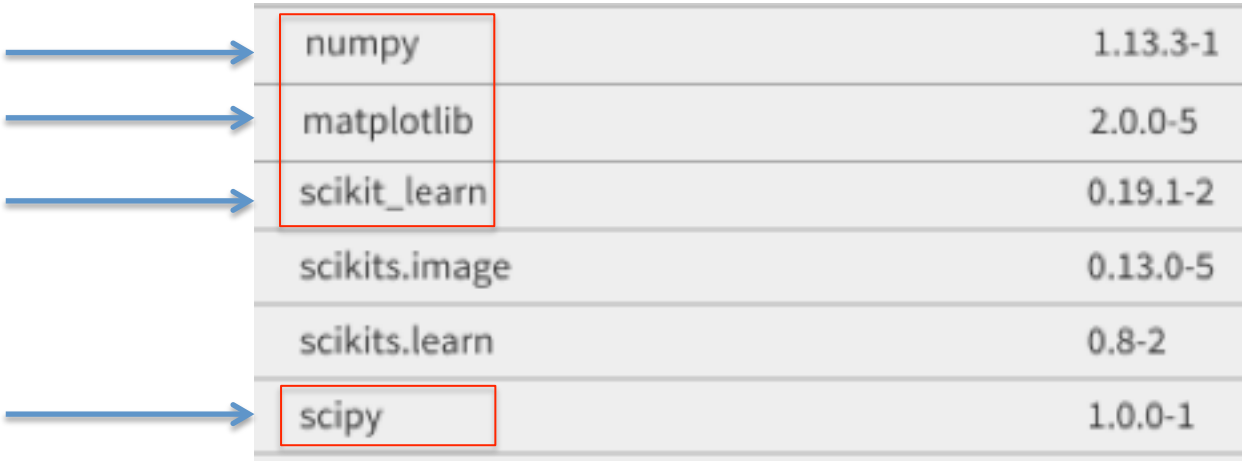
- TensorFlow in Canopy:



Machine Learning With TensorFlow

- TensorFlow in Canopy:

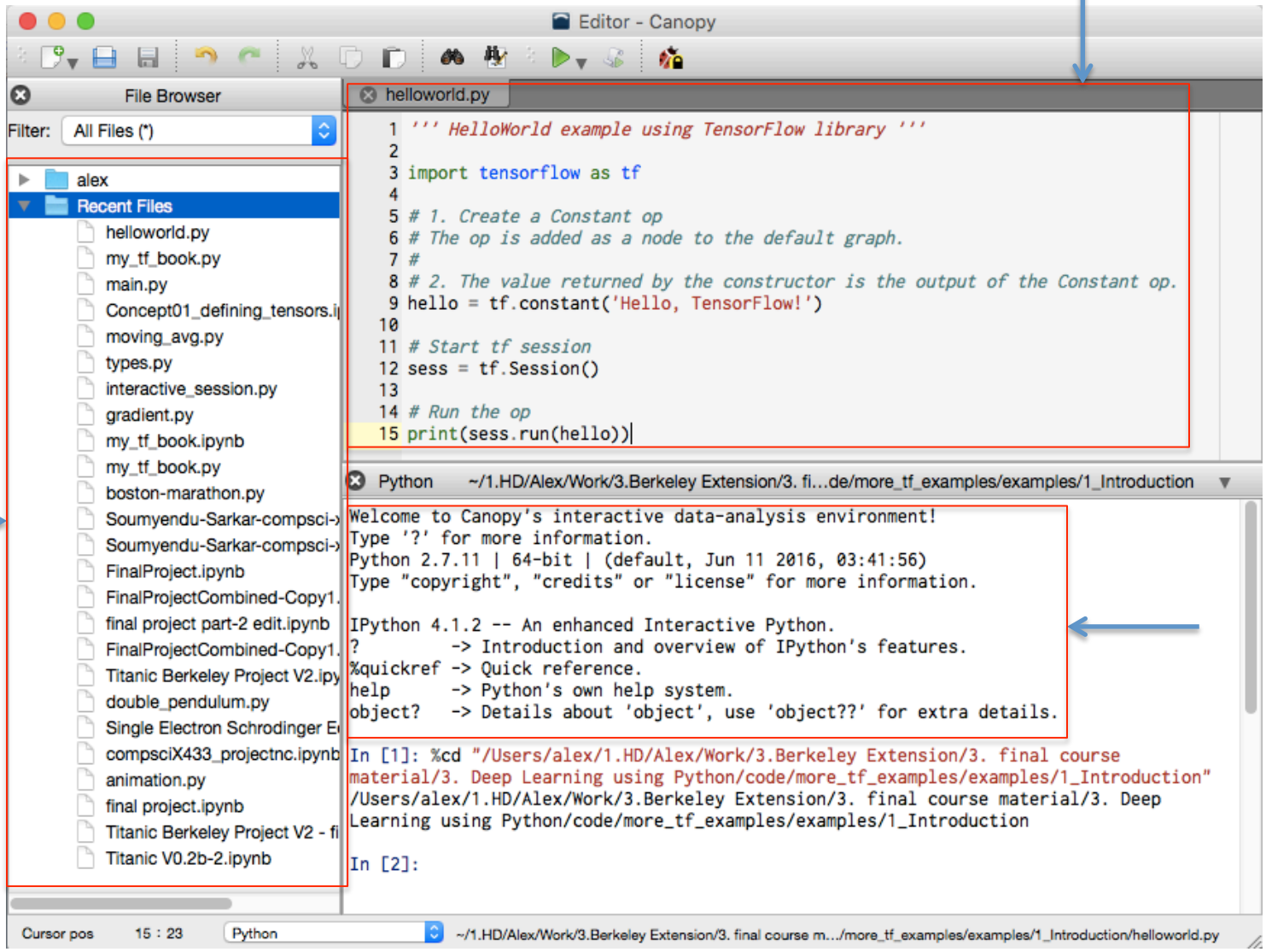
TensorFlow™



numpy	1.13.3-1
matplotlib	2.0.0-5
scikit_learn	0.19.1-2
scikits.image	0.13.0-5
scikits.learn	0.8-2
scipy	1.0.0-1

Machine Learning With TensorFlow

- TensorFlow in Canopy:

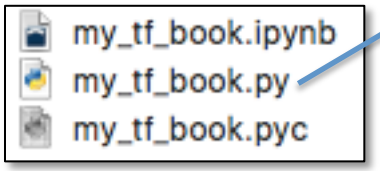


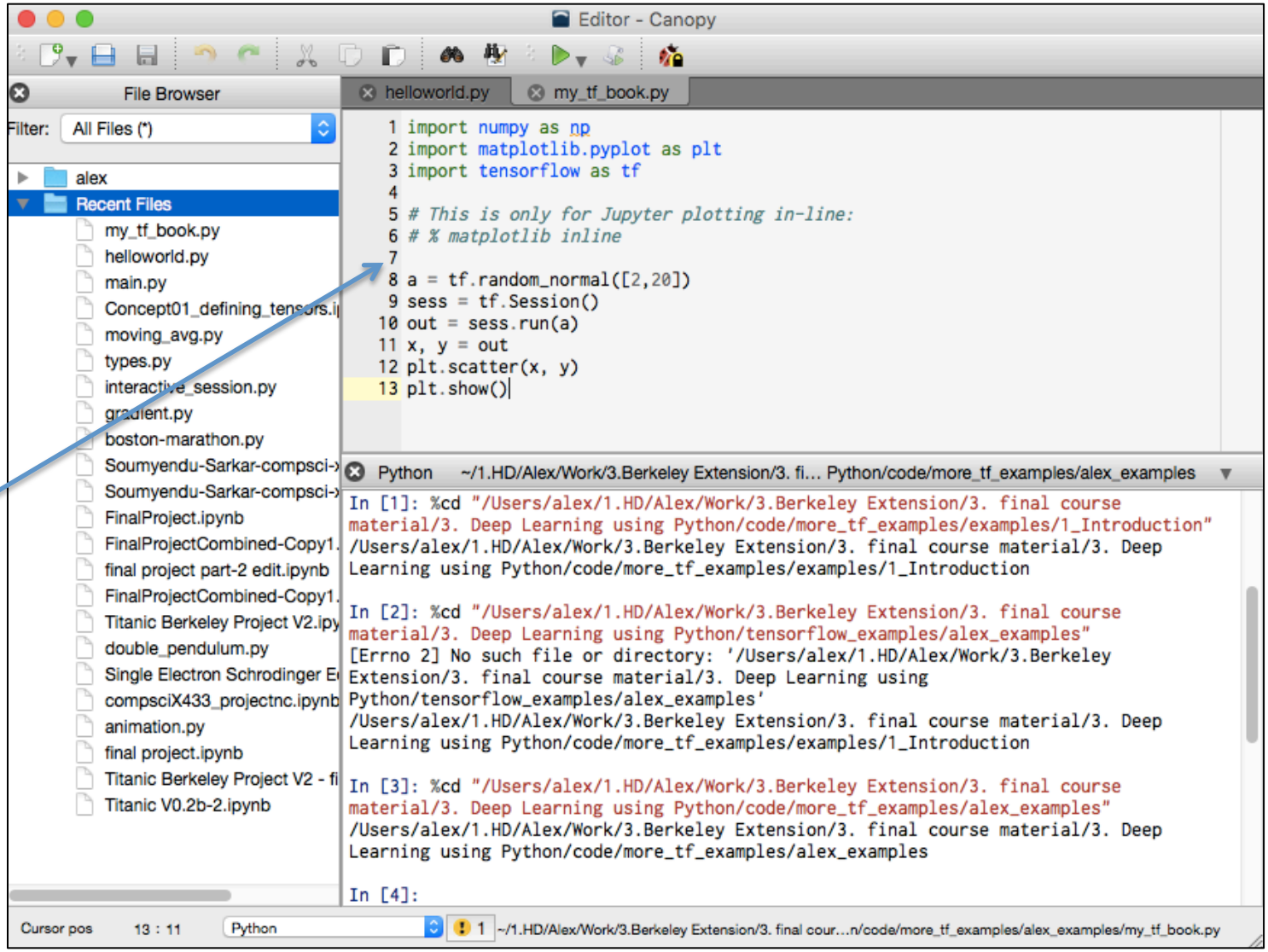
Machine Learning With TensorFlow

- TensorFlow in Canopy:

3 ways to run code:

1. as script file



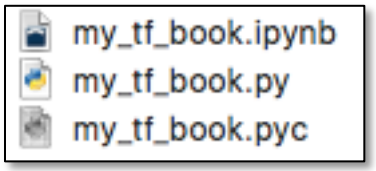


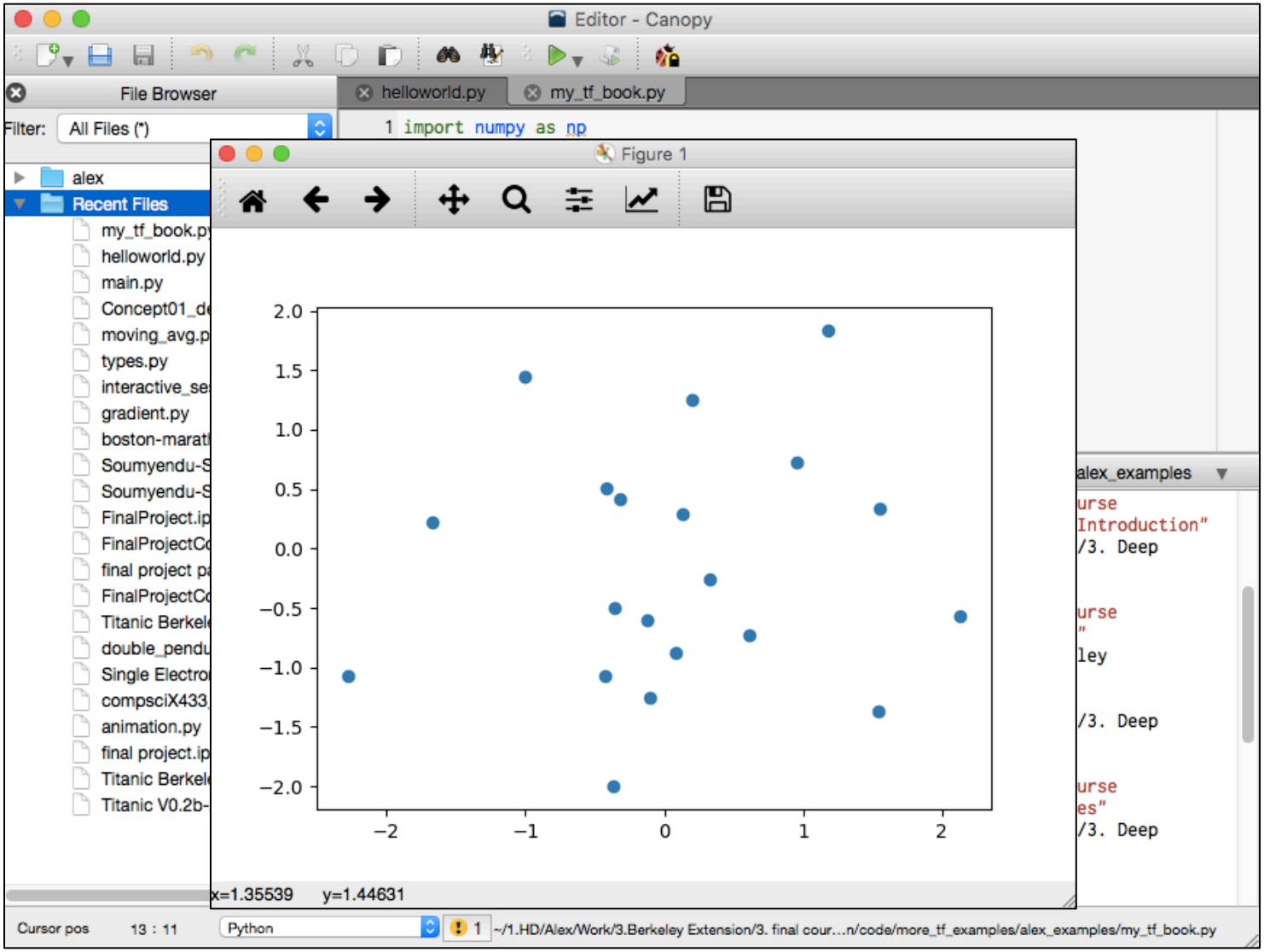
Machine Learning With TensorFlow

- TensorFlow in Canopy:

3 ways to run code:

1. as script file



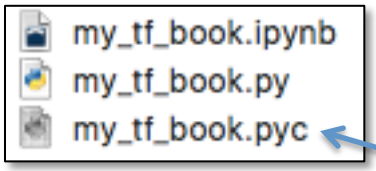


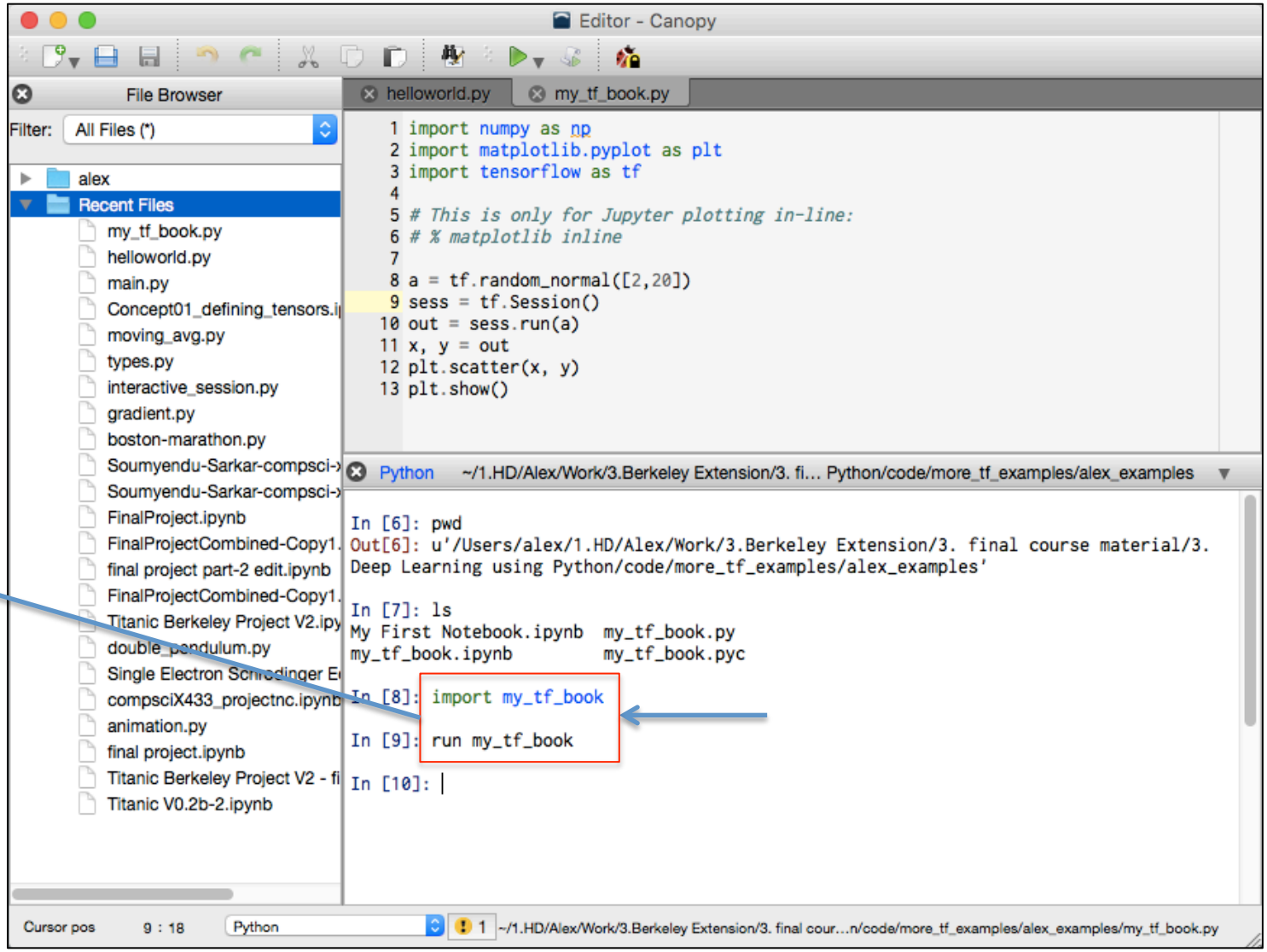
Machine Learning With TensorFlow

- TensorFlow in Canopy:

3 ways to run code:

2. as imported file
(creates compiled version .pyc)



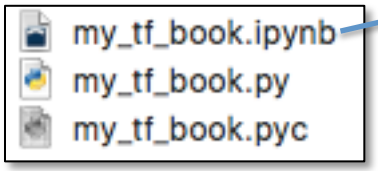


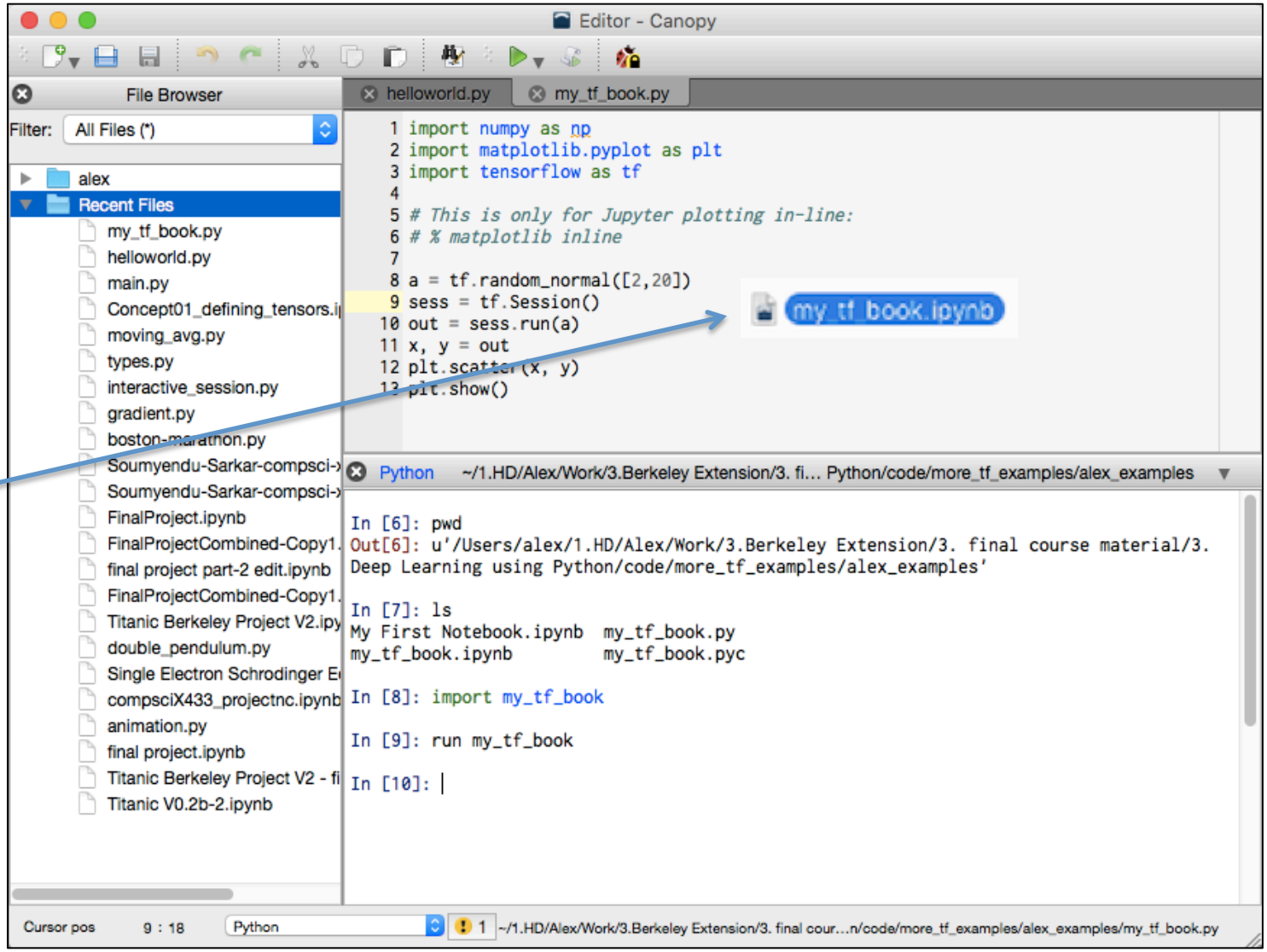
Machine Learning With TensorFlow

- TensorFlow in Canopy:

3 ways to run code:

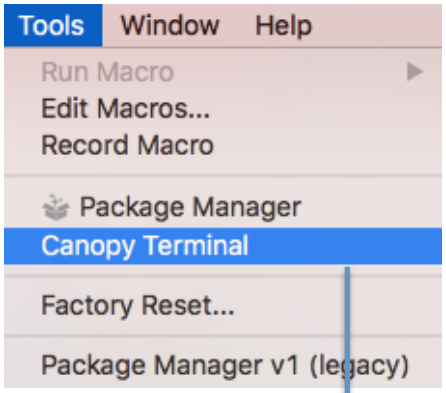
3. in notebook
(it is a .ipynb file)





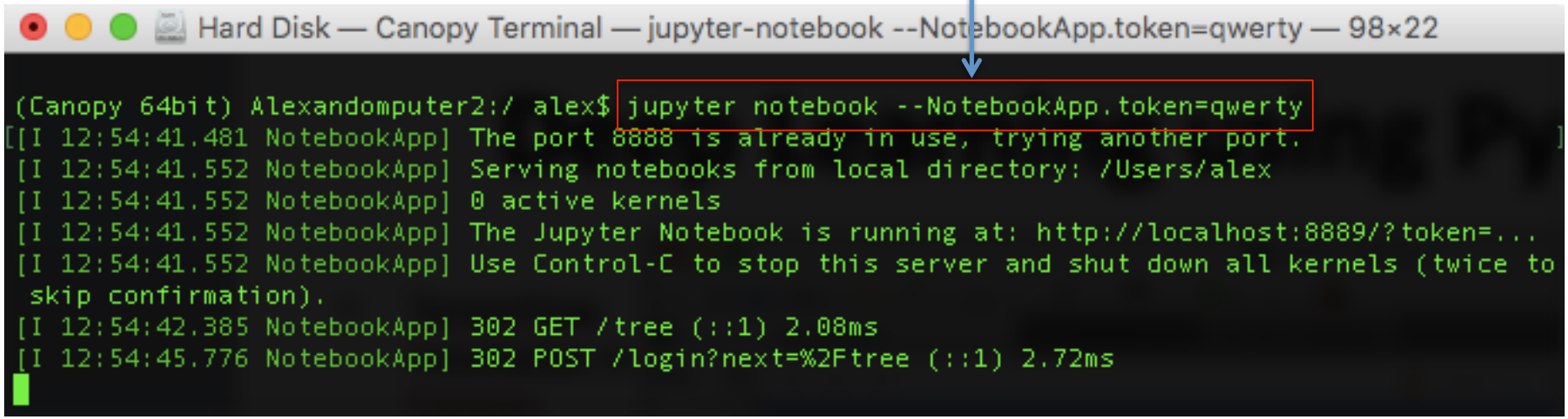
Machine Learning With TensorFlow

- TensorFlow in Canopy:



3 ways to run code:

3. in notebook
(it is a .ipynb file)

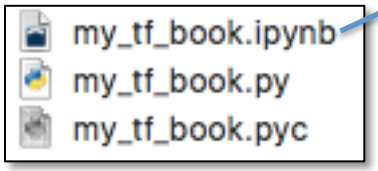


Machine Learning With TensorFlow

- TensorFlow in Canopy:

3 ways to run code:

3. in notebook
(it is a .ipynb file)



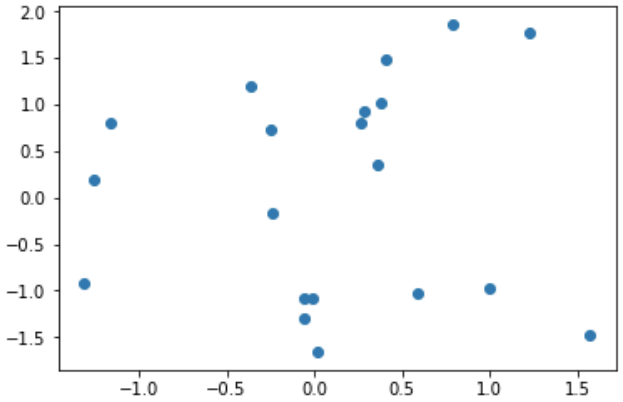
jupyter my_tf_book

Logout

File Edit View Insert Cell Kernel Help Python 2

Code CellToolbar

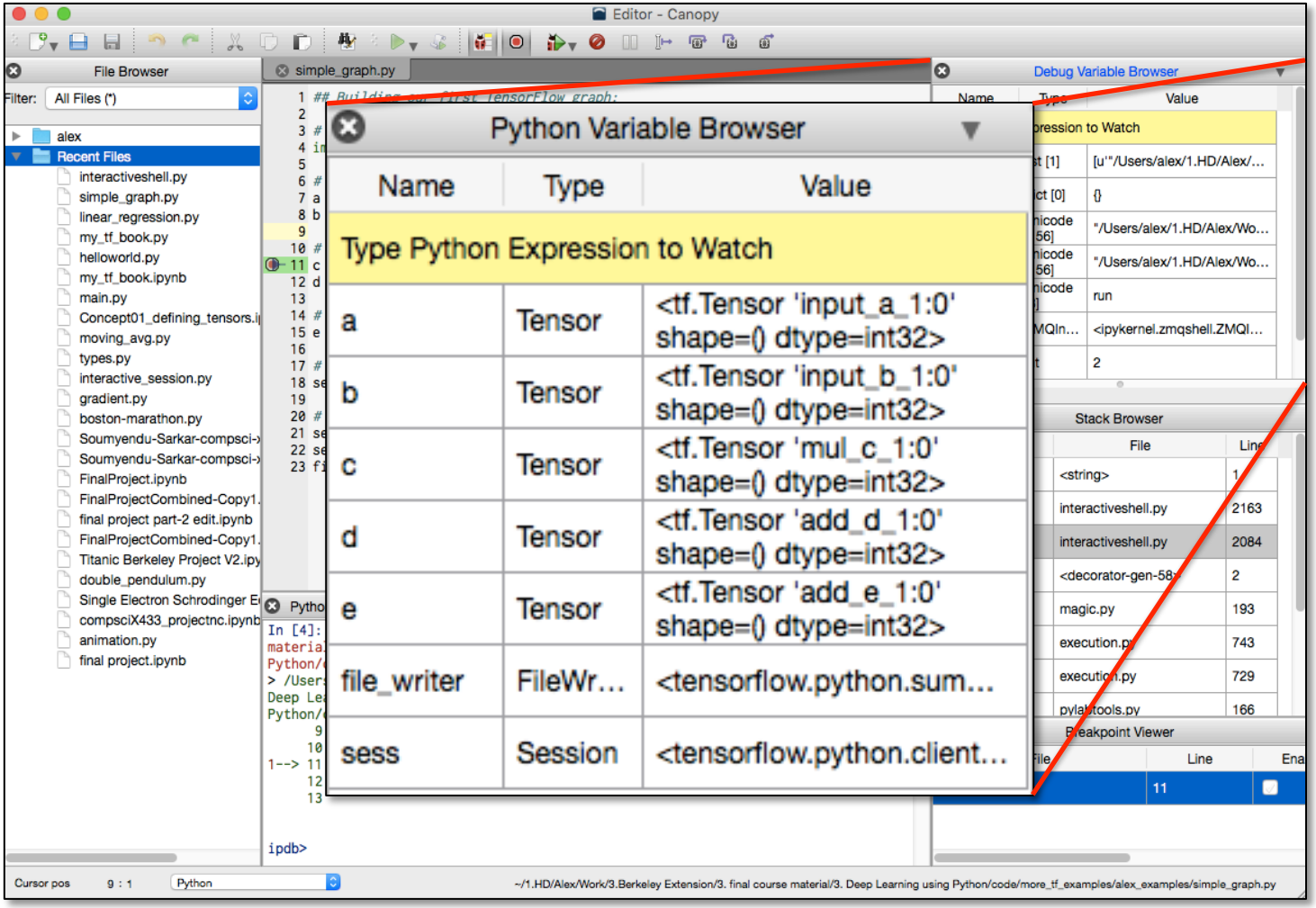
```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        import tensorflow as tf

        % matplotlib inline
        a = tf.random_normal([2,20])
        sess = tf.Session()
        out = sess.run(a)
        x, y = out
        plt.scatter(x, y)
        plt.show()
```




```
In [ ]:
```

Canopy debugger

- Running in debug mode:



Machine Learning With TensorFlow

- **tf.Session**:

    - Type: type
    - A class for running TensorFlow operations.
    - A `Session` object encapsulates the environment in which `Operation` objects are executed, and `Tensor` objects are evaluated.

For example:

```
# Build a graph.
a = tf.constant(5.0)
b = tf.constant(6.0)
c = a * b
sess = tf.Session()    # Launch the graph in a session.
print(sess.run(c))     # Evaluate the tensor `c`.
```

Machine Learning With TensorFlow

- Artificial Intelligence concept
  - It is a concept conceived in the mid 50s with the intention to construct complex machines (computers) that possessed the same characteristics of human intelligence
  - The main idea is to create technologies able to perform specific tasks better than humans can
  - Some of the first intelligent robots emerged in movies such as Star Wars – C-3PO
  - Artificial intelligence (AI) is already part of our everyday lives
  - Some examples of AI are: image classification, face recognition, voice recognition, speaker recognition, emotion recognition, etc.

Machine Learning With TensorFlow

- Data Mining and Machine Learning concepts
  - In data mining, the data is stored electronically and the search is automated (or semi-automated)
  - It has been estimated that the amount of data stored in the world's databases doubles every 20 months …
  - … Thus the opportunities for data mining increase
  - Data mining is about solving problems by analyzing data already present in databases
  - Data mining finds patterns that can be analyzed to identify distinguishing characteristics

Machine Learning With TensorFlow

- Data Mining and Machine Learning concepts
  - Machine learning came directly from the early AI scientists. The algorithmic approaches over the years included: *decision tree learning*, *logic programming*, *clustering*, *reinforcement learning*, *Bayesian networks*, etc.
  - To "learn" by definition for us humans means to:
    - Obtain knowledge of something and being aware of something
    - Be informed, receive instructions, commit to memory
  - An operational definition for "learn" can be formulated like:
    - Things learn when they change their behavior in a way that makes them perform better in the future. The definition of "better" is given by us and means "performance" rather than "knowledge"

Machine Learning With TensorFlow

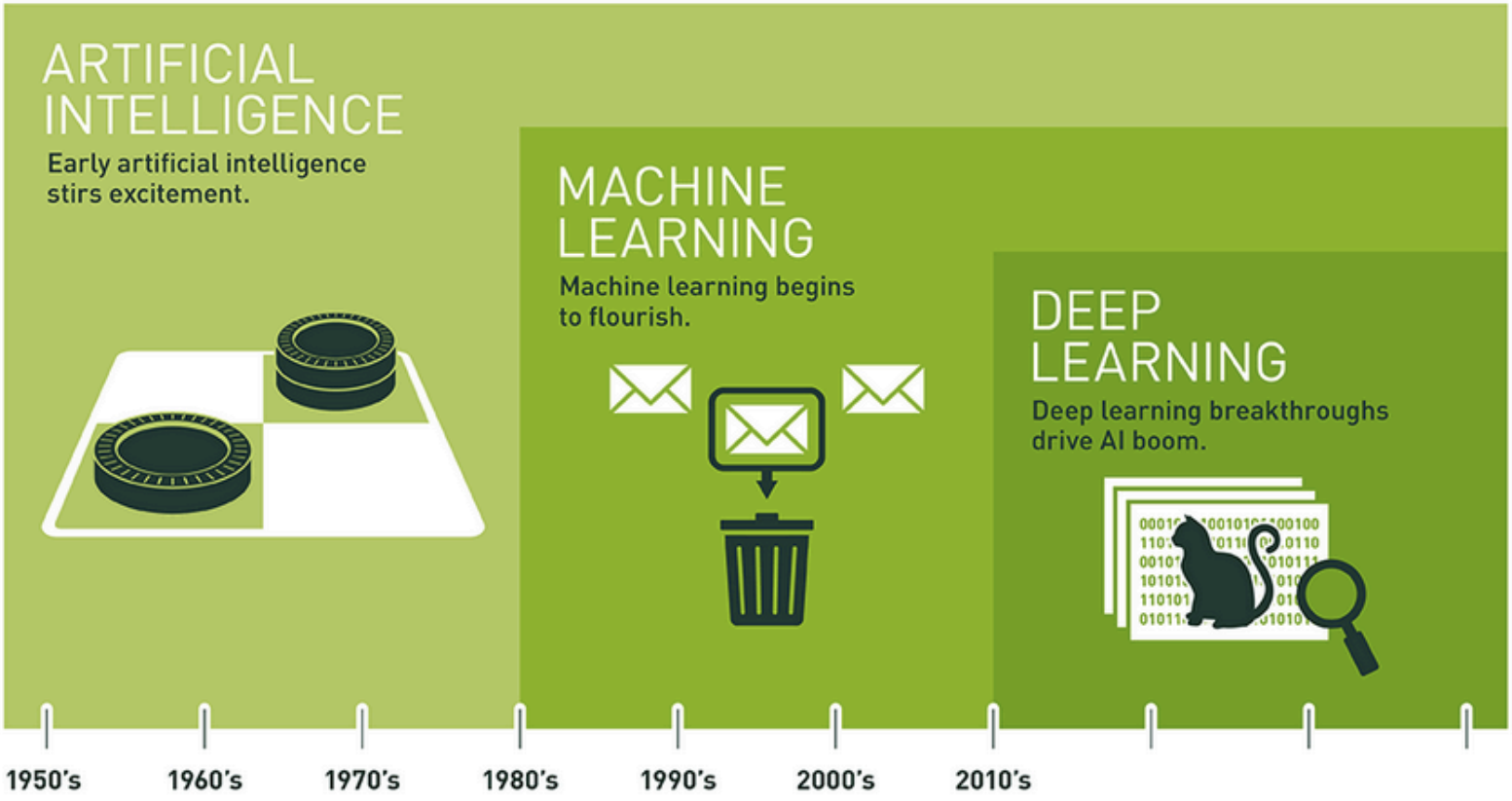
- Data Mining and Machine Learning concepts
  - Machine learning does not entail the conceptual limitations mentioned herein, and disregards any particular philosophical stance about what learning actually is
  - Machine learning is therefore a practical process connected to Data mining while using algorithms to parse data, learn from it, and then make a prediction
  - Performance:
    - the machine is "trained" using large amounts of data and algorithms that give it the ability to learn;
    - it is then "tested" on a new (usually unknown) set of data

Machine Learning With TensorFlow

- Deep Learning concept
  - Deep Learning is an area in Machine Learning that has been introduced with the objective of moving Machine Learning closer to its goal: Artificial Intelligence
  - Deep Learning has enabled many practical applications of Machine Learning and by extension the overall field of AI
  - Deep Learning breaks down tasks in ways that makes all kinds of machine assists seem possible
  - Areas of implementation are infinite, but some of them are: preventive healthcare, security systems, robotics, cars without drivers, better media recommendations, etc.

Machine Learning With TensorFlow

- Deep Learning concept



source: nvidia.com

Machine Learning With TensorFlow

HW assignment 1:

Chosen, install and configure your Python-Tensorflow environment