Purpose of this course

The primary objective of this course is to

- teach you how to write python code that makes use of a **modern machine learning software library**,
  - scikit-learn and Keras (a deep learning API) on TensorFlow
- to rapidly prototype and test machine learning techniques on data.

Means we will focus on how to use existing implementations of learning algorithms, and best practices for training and evaluating them on data.

Machine Learning is a Large Hairy Beast

Modern machine learning is really the nexus of several traditional ECE disciplines, including

- Mathematical statistics (Random variables, stochastic processes, detection & estimation)
- Optimization both numerical and analytical
- Computer Architecture (acceleration via GPU computing)
- Signal Processing & Information Theory (c.f. multimedia encoding, time series)

The ECE department has advanced courses, both undergraduate and graduate, in each of these topics.

What this course is NOT

We will spend at most a <u>small</u> amount of time on

- the detailed mathematics behind how these algorithms work and are trained,
  - (other than to explain at a high level what the algorithms do)
- how their implementations in the libraries make use of computer hardware architecture enabled accelerations.
- how the libraries we are using may be improved
- a historically accurate review of the development of ML
- existing theory (economic, physics, etc) for explaining phenomena in the datasets

BUT you would benefit from learning all of these.

Other courses in the ECE department emphasize those aspects.

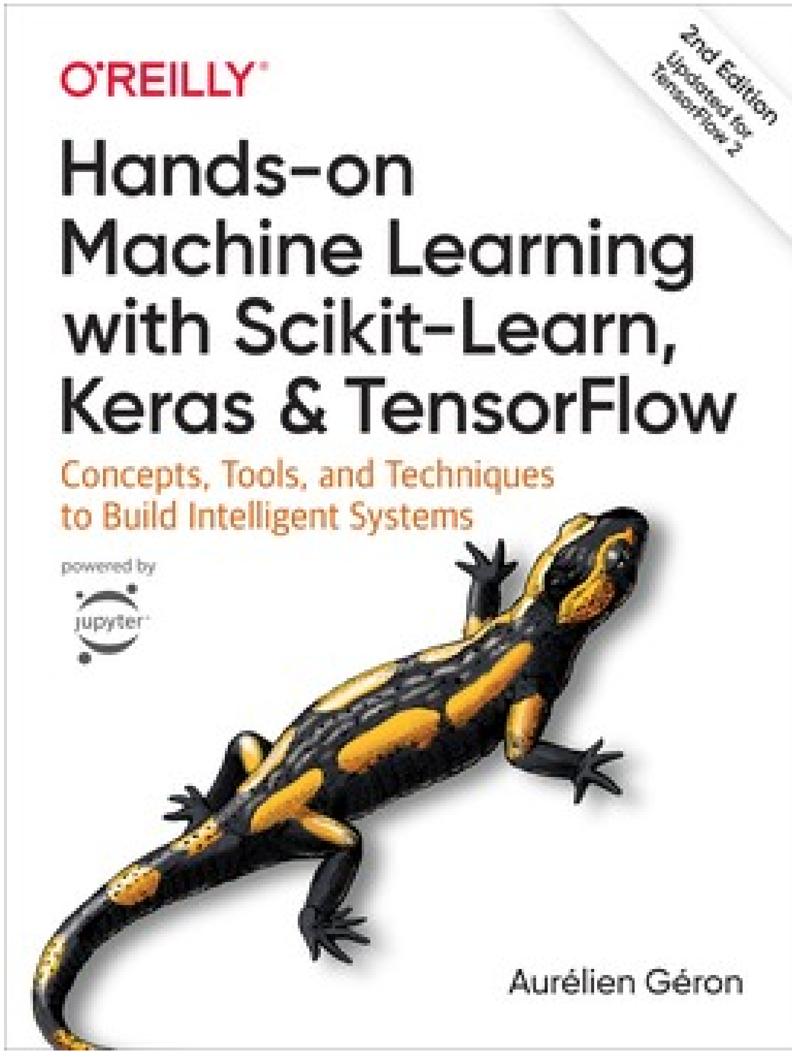
Please Learn More After This Course

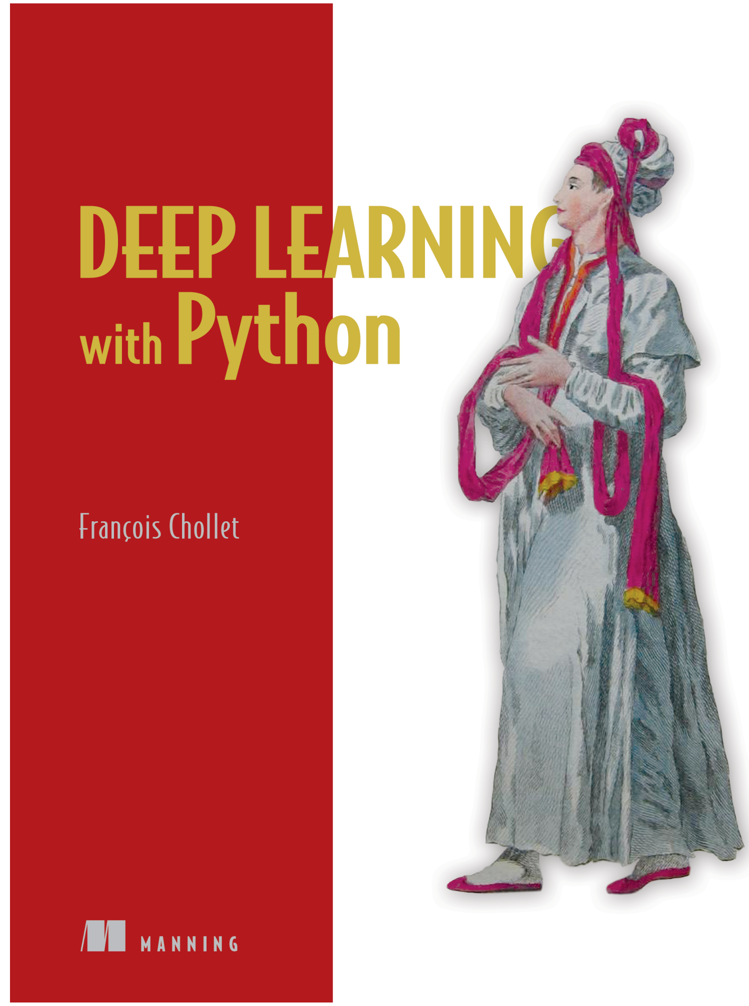
While you can drive the library without understanding in detail how and why it works and how its computations are accelerated (fun, but work), and you can learn how the library works without getting used to driving it (harder), the best job candidates will be those that understand both. Take the other courses, too.

- Mathematical statistics (Random variables, stochastic processes, detection & estimation)
- Optimization both numerical and analytical
- Computer Architecture (acceleration via GPU computing)
- Signal Processing & Information Theory (c.f. multimedia encoding, time series)

The ECE department has advanced courses, both undergraduate and graduate, in each of these topics.

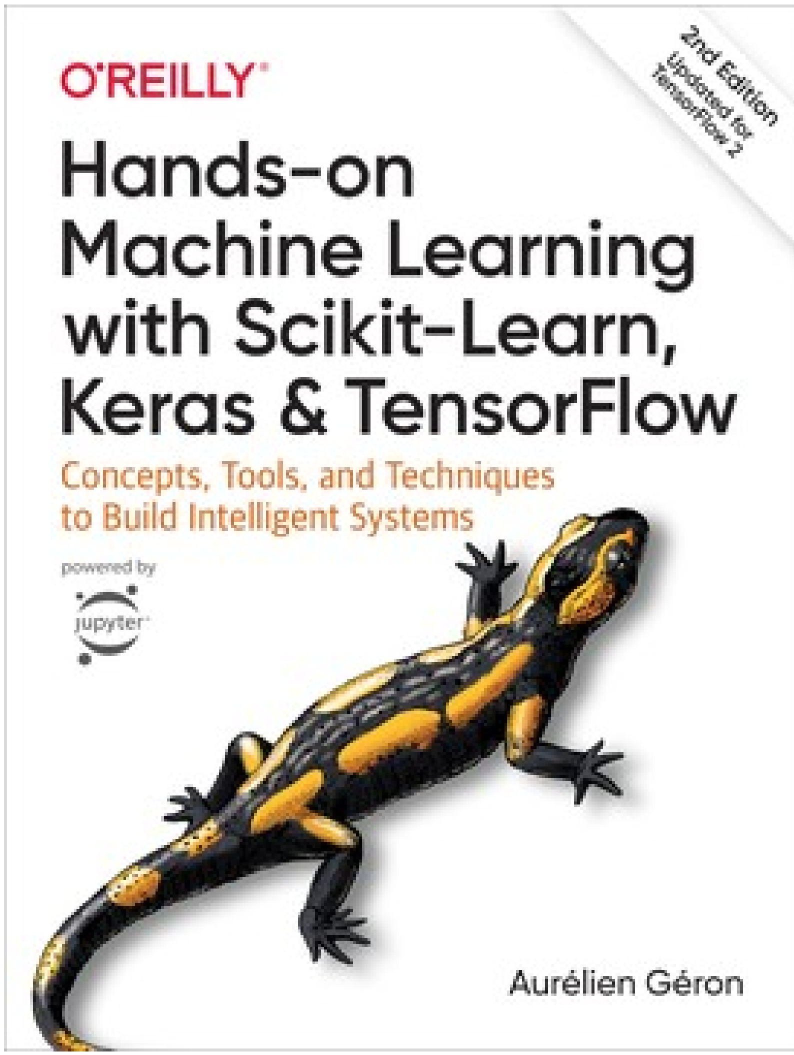
Course Textbooks





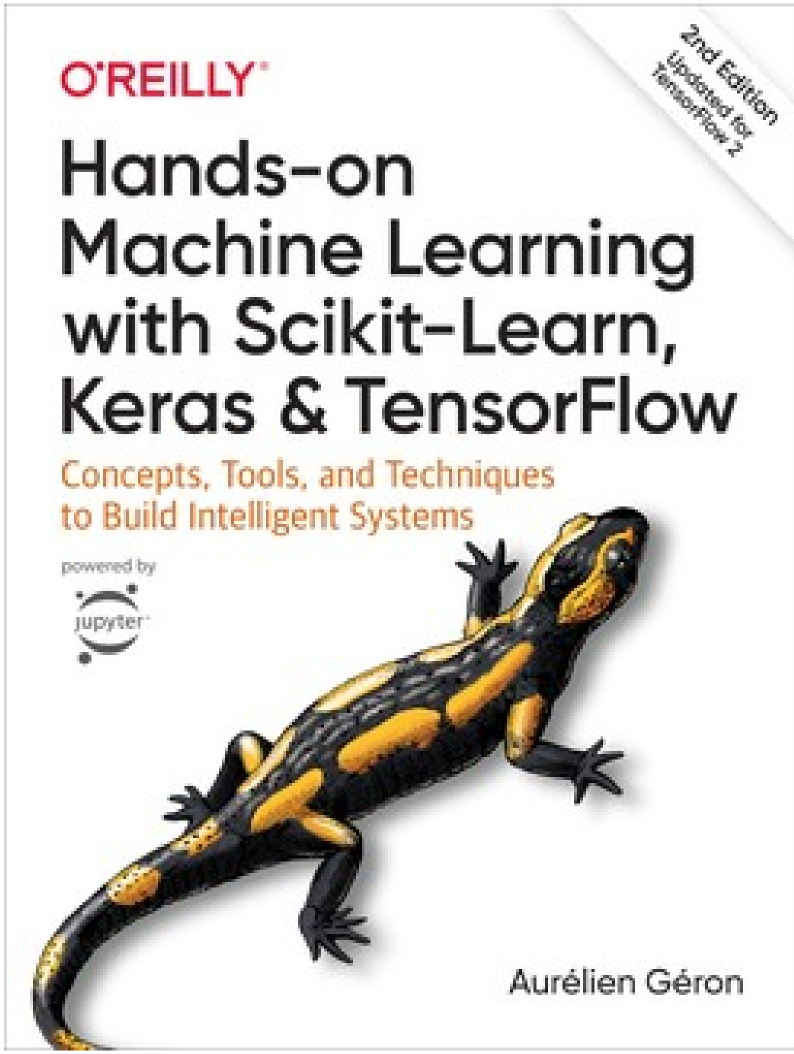
Obtain at a minimum a copy of HOML. We will be following HOML closely.

Why These Textbooks? – HOML



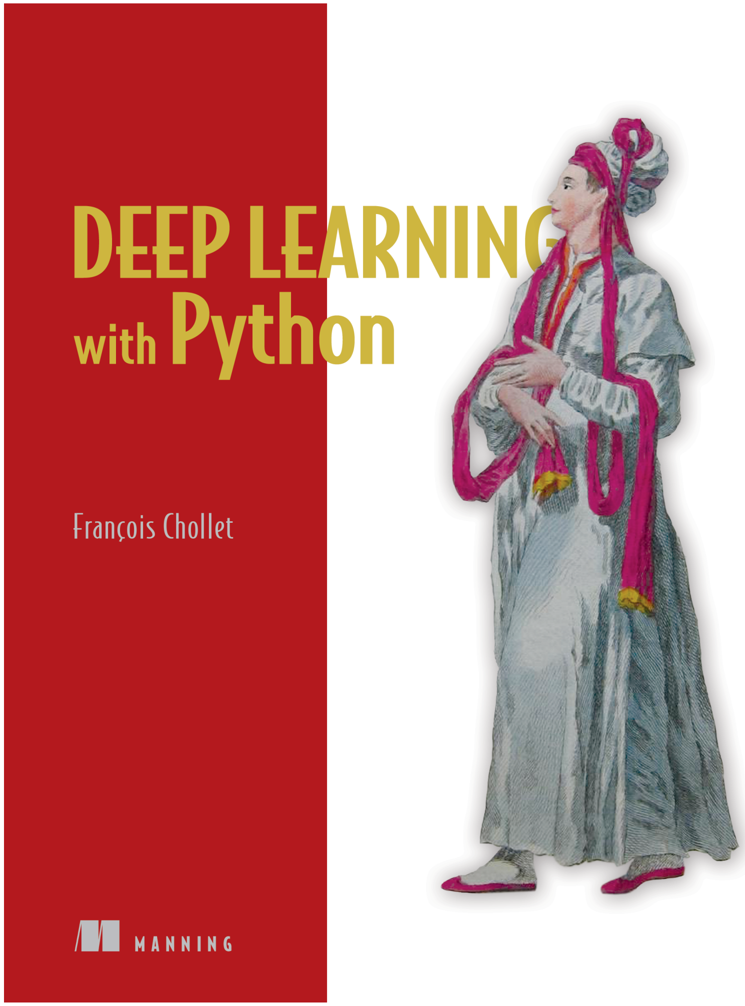
- Strengths:
  - super recent (October '19)
  - covers a unique sampling of topics
  - everything applied/coded immediately
- Weaknesses:
  - historically inaccurate
  - Over-emphasizes recent winners in ML power tug of war
  - Some important algo.s missing (graphical models/BP/variational inference)
  - at points improper attribution of ideas
  - Wiener, LMS, adaptive filtering missing (terrible!)
  - bad footnotes (e.g. signals & Shannon)
  - given wireless startup history author should know better

Why These Textbooks? – HOML



- Summary: use to learn how to rapidly prototype in scikit-learn and TensorFlow, but do not use to understand the historical development of ML, deep math details, or who the important figures are/were
- weekly reading on syllabus (ch 1 & 2 this week, ch. 3-4 next)

Why These Textbooks? – Chollet



- Chollet is the author of Keras API
- Keras API enables one to describe, train, test deep NN w/ very short code
- API can be run on several NN backends (theano, mxn, tensorflow)
- TensorFlow has extended it to include additional features
- No required reading from this text yet, supplemental later in course.

Course Syllabus

Go over course syllabus emphasizing

- office hours, TA, skype, emails
- Grading policy (5 biweekly coding projects, drop one)
- special topics course, first time, developed live, EVERYTHING subject to change
- required reading
- assignment deadlines

Outline of Rest of Lecture

- Setting up our development environment
- ML taxonomy (ch. 1)
  - dataset of instances: features, labels
  - classification vs. regression
  - supervised vs. unsupervised
  - supervised (model based) learning process
    - data preparation & segmentation (train, validate, test)
    - fit (optimize)
    - test
  - overfitting vs. underfitting
- some first machine learning algorithms
  - linear regression
  - SVMs & decision trees coming up

Setting up our Development Environment

- ssh into xunil-05.coe.drexel.edu with your drexel username and password
- python3 is located at /opt/rh/rh-python36/root/usr/bin/python3.6
- we will set up a venv (virtual environment) in which to install our packages so we can control package versions (and what is installed) independently across virtual environments
- type the following
  - **/opt/rh/rh-python36/root/usr/bin/python3.6 -m venv mlCourse**
- you can now load the virtual environment and install python packages as if you were root – type the following to activate the environment
  - **source mlCourse/bin/activate**
- later when we are done you will simply type **deactivate** to close the virtual environment

Setting up our Development Environment – cont.d

- check version of pip, (you will find its horribly old)
  - **python3 -m pip –version**
- upgrade pip to latest version
  - **python3 -m pip install –upgrade pip**
- install the packages we want to use for now
  python3 -m pip install -U jupyter matplotlib numpy pandas scipy scikit-learn
- test our installation
  - **python3 -c ”import jupyter, matplotlib, numpy, pandas, scipy, sklearn”**
- should execute and return nothing, indicating success

ML Taxonomy

Supervised learning (example spam detection)

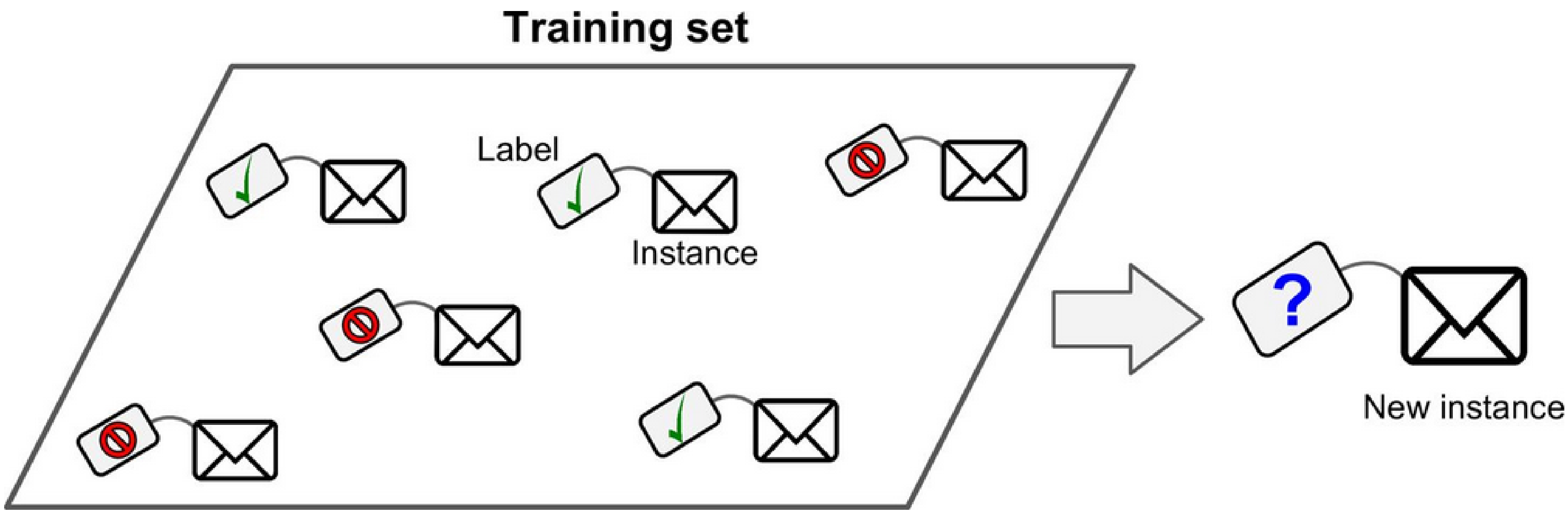


Figure 1-5. A labeled training set for spam classification (an example of supervised learning)

example of a *classification* task (spam or ham)