Machine Learning Software and Tools (DRAFT)

Mark A. Austin

austin@umd.edu
ENCE 688P, Spring Semester 2022

University of Maryland

March 22, 2026

Overview

Appendix A

Python Software Setup

TensorFlow, Keras, Jupyter Notebook, Anaconda

Working with TensorFlow and Keras

TensorFlow 2 (Open-Source Library for Machine Learning)

- TensorFlow provides high- and low-level APIs for development of deep learning models.

Keras (Neural Network Library)

- Keras is an extension of TensorFlow. It contains features to simplify the task of coding machine learning tasks.
- When you develop a model in Keras you are still developing a model in TensorFlow. Keras just makes things easier to code.

Jupyter Notebook (Web-based Application)

- Web-based authoring of documents that combine live code with narrative text, equations and visualization.

Installation (On a Mac)

Step 1. Make sure that you have Python 3.X ...

```
prompt >> python --version
Python 3.7.9
```

Step 2. Install TensorFlow from terminal window

```
prompt >> pip3 install tensorflow
   Collecting tensorflow
   Downloading tensorflow-2.3.1-cp37m-macox_10_9_x86_62.whl (165 MB) ..
```

Step 3. Install Keras from terminal window

```
prompt >> pip3 install Keras
```

Installation (On a Mac)

Step 4. Install and run Jupyter Notebook

```
prompt >> pip3 install jupyter
```

Step 5. Run TensorFlow ...

To run from the terminal window:

```
prompt >> python3 TensorFlowApplicationFile.py
```

To run Jupyter Notebook:

```
prompt >> jupyter notebook
```

Jupyter Notebook and TensorFlow

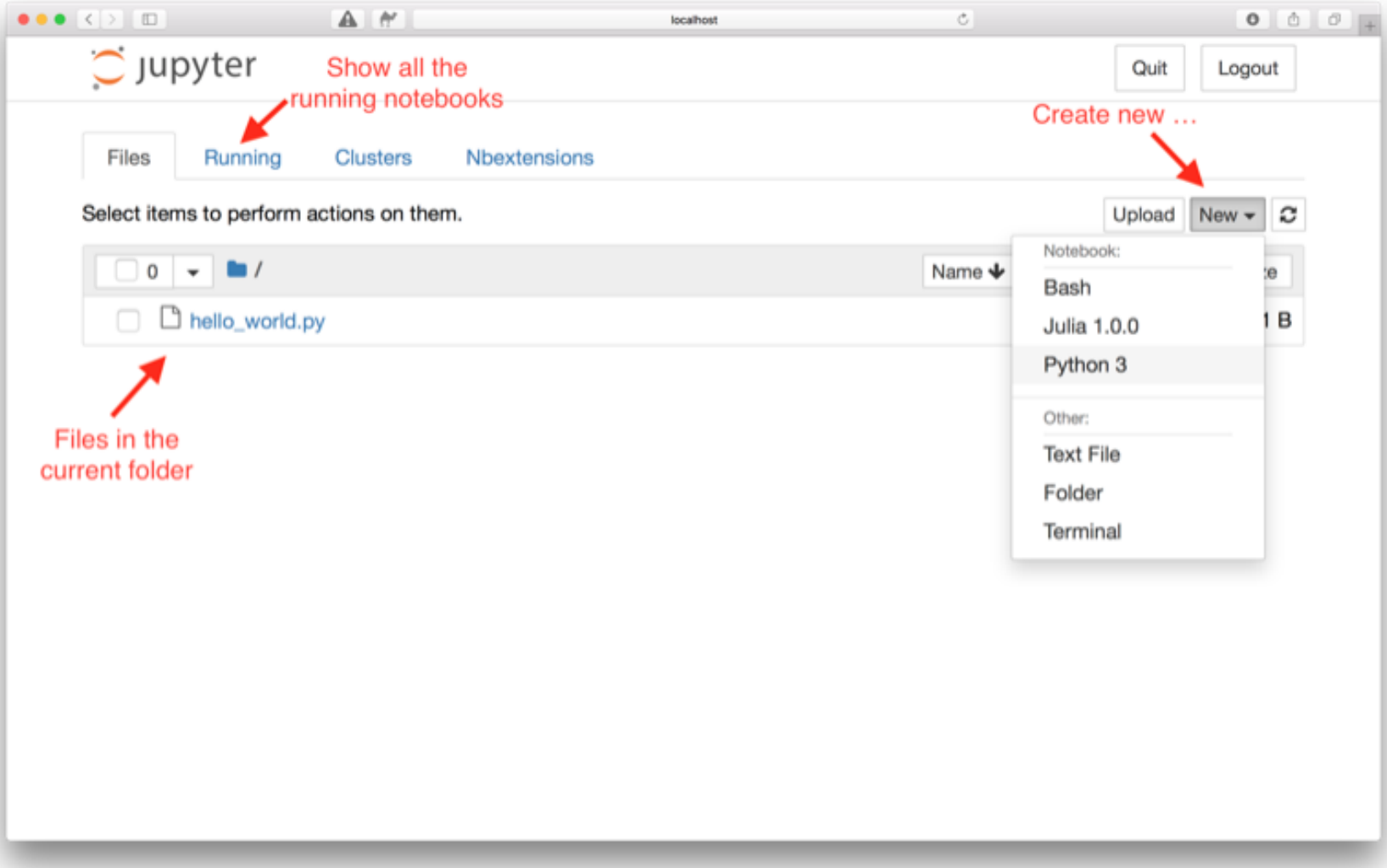
Jupyter Notebook Use Cases:

- Data cleaning and transformation.
- Numerical simulation.
- Statistical modeling.
- Data visualization.
- Machine learning.

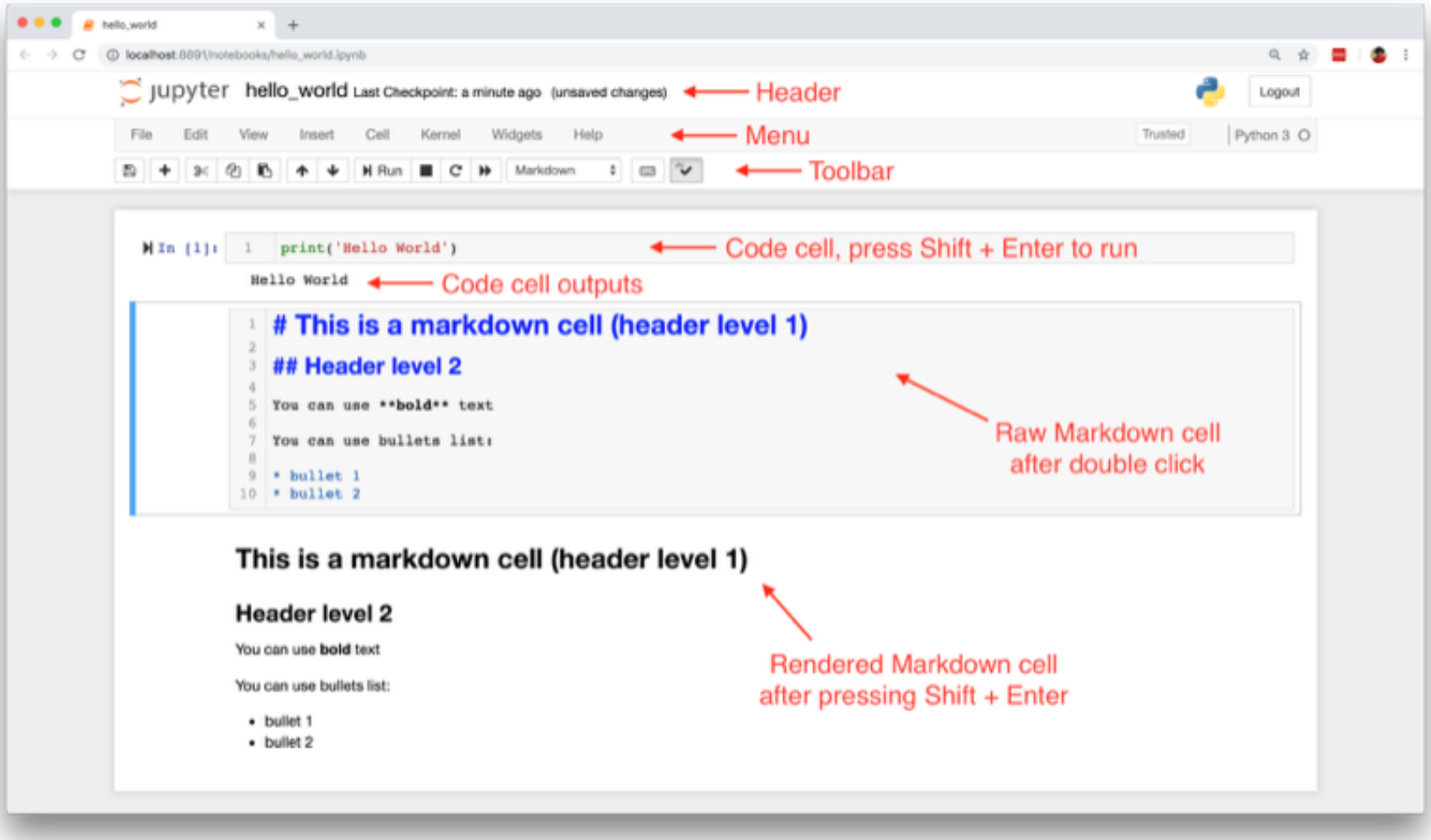
Jupyter Notebook File Format:

- File format is JSON-based with extension .ipynb (named after projects predecessor IPython).
- Supports documents containing text, source code, rich media data and metadata.

Jupyter Notebook User Interface



Jupyter Notebook User Interface



Jupyter Notebook Cells and Code Execution

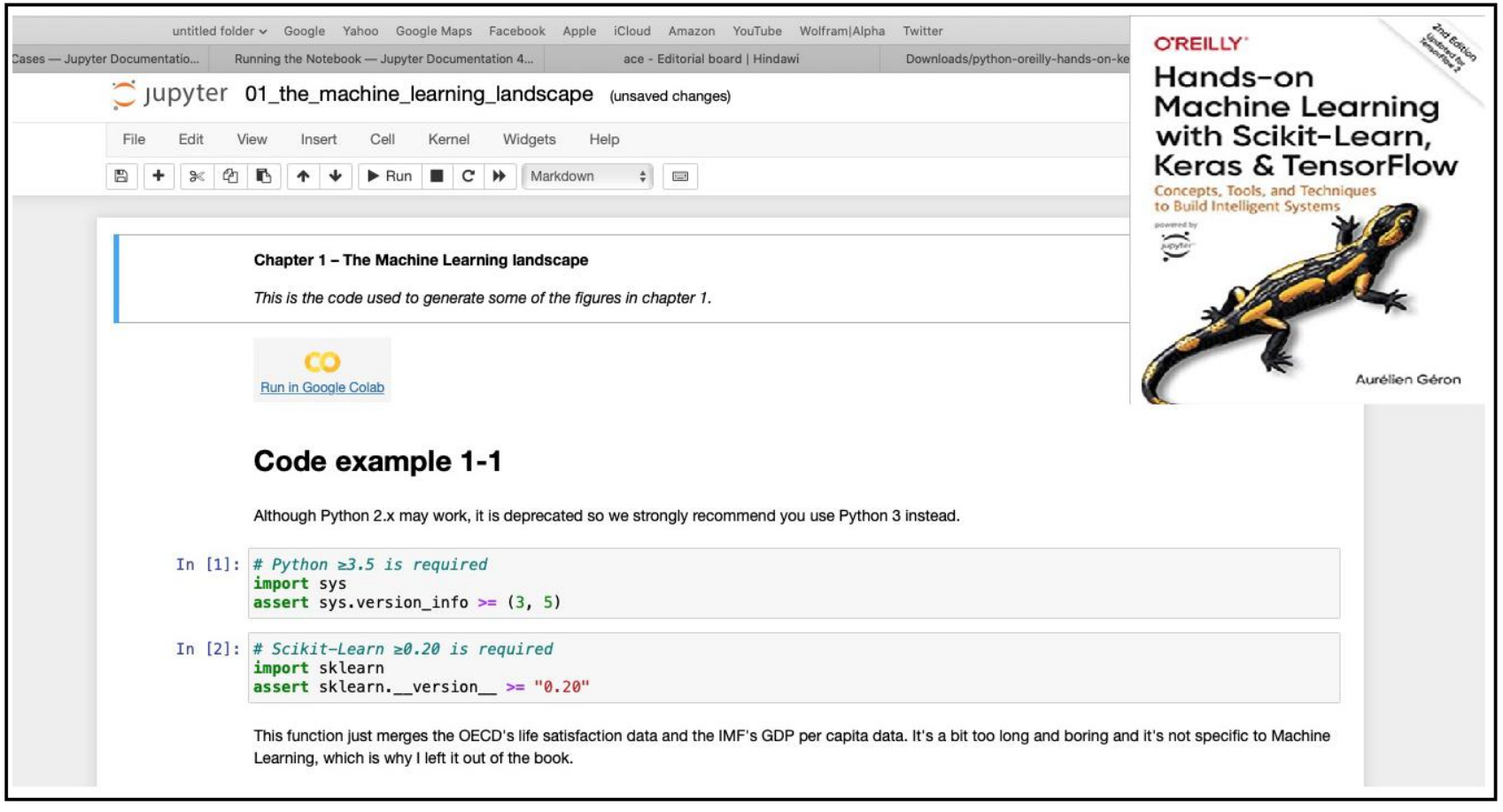
Jupyter Notebook Cells:

- **Code Cells:** Allows for development and editing of new code, with syntax highlighting and tab completion.
- **Markdown Cells:** Document the computational process with the Markdown language (a simple way to perform text markup). Can also include mathematics with LaTeX notion.
- **Raw Cells:** Provide a place in which you can write output directly.

Code Execution:

- When a code cell is executed, the code is sent to the kernel associated with the code.
- Results are returned to the computation and then displayed.

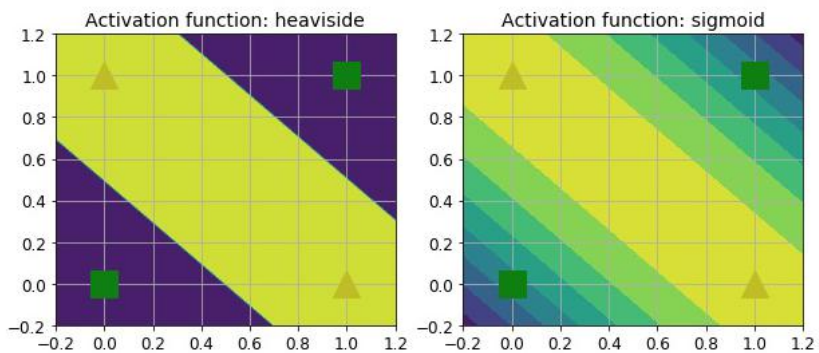
Jupyter Notebook and TensorFlow



Jupyter Notebook and TensorFlow

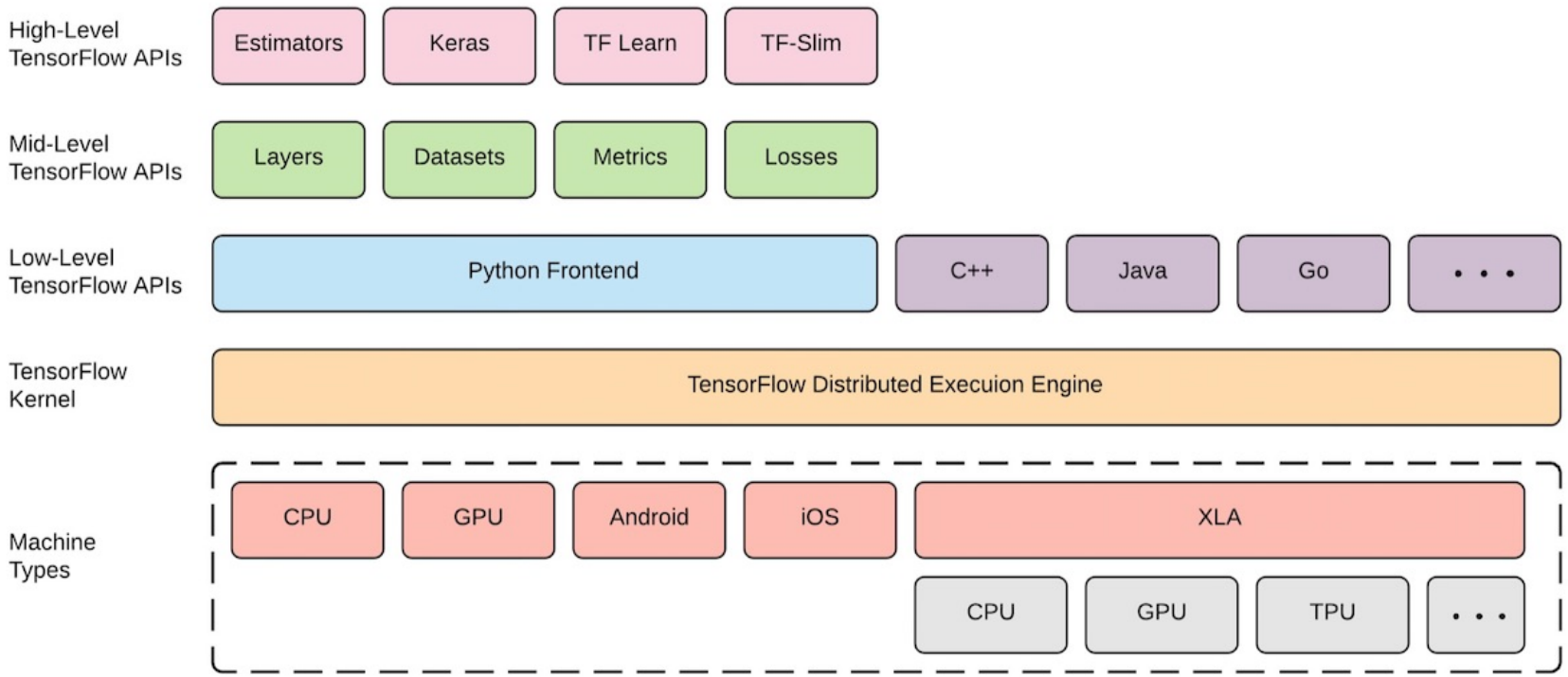
In [7]:
```
def heaviside(z):
    return (z >= 0).astype(z.dtype)

def mlp_xor(x1, x2, activation=heaviside):
    return activation(-activation(x1 + x2 - 1.5) + activation(x1 + x2 - 0.5) - 0.5)
```

In [8]:
```
x1s = np.linspace(-0.2, 1.2, 100)
x2s = np.linspace(-0.2, 1.2, 100)
x1, x2 = np.meshgrid(x1s, x2s)

z1 = mlp_xor(x1, x2, activation=heaviside)
z2 = mlp_xor(x1, x2, activation=sigmoid)

plt.figure(figsize=(10,4))

plt.subplot(121)
plt.contourf(x1, x2, z1)
plt.plot([0, 1], [0, 1], "gs", markersize=20)
plt.plot([0, 1], [1, 0], "y^", markersize=20)
plt.title("Activation function: heaviside", fontsize=14)
plt.grid(True)

plt.subplot(122)
plt.contourf(x1, x2, z2)
plt.plot([0, 1], [0, 1], "gs", markersize=20)
plt.plot([0, 1], [1, 0], "y^", markersize=20)
plt.title("Activation function: sigmoid", fontsize=14)
plt.grid(True)
```



TensorFlow Software Architecture

APIs to Keras, Java, C++, etc ....



TensorFlow Software Architecture

Swift Interface to TensorFlow

Swift for TensorFlow is a next generation system for deep learning and differentiable computing.

By integrating directly with a general purpose programming language, Swift for TensorFlow enables more powerful algorithms to be expressed like never before.

See tutorials

Tutorials show you how to use Swift

How it works

First-class autodiff

Differentiable programming gets first-class support in a general-purpose programming language. Take derivatives of functions, and make custom data structures differentiable in an instant.

Next-generation APIs

New APIs, informed by the best practices of today and the research directions of tomorrow, are easier to use and more powerful.

Builds on TensorFlow

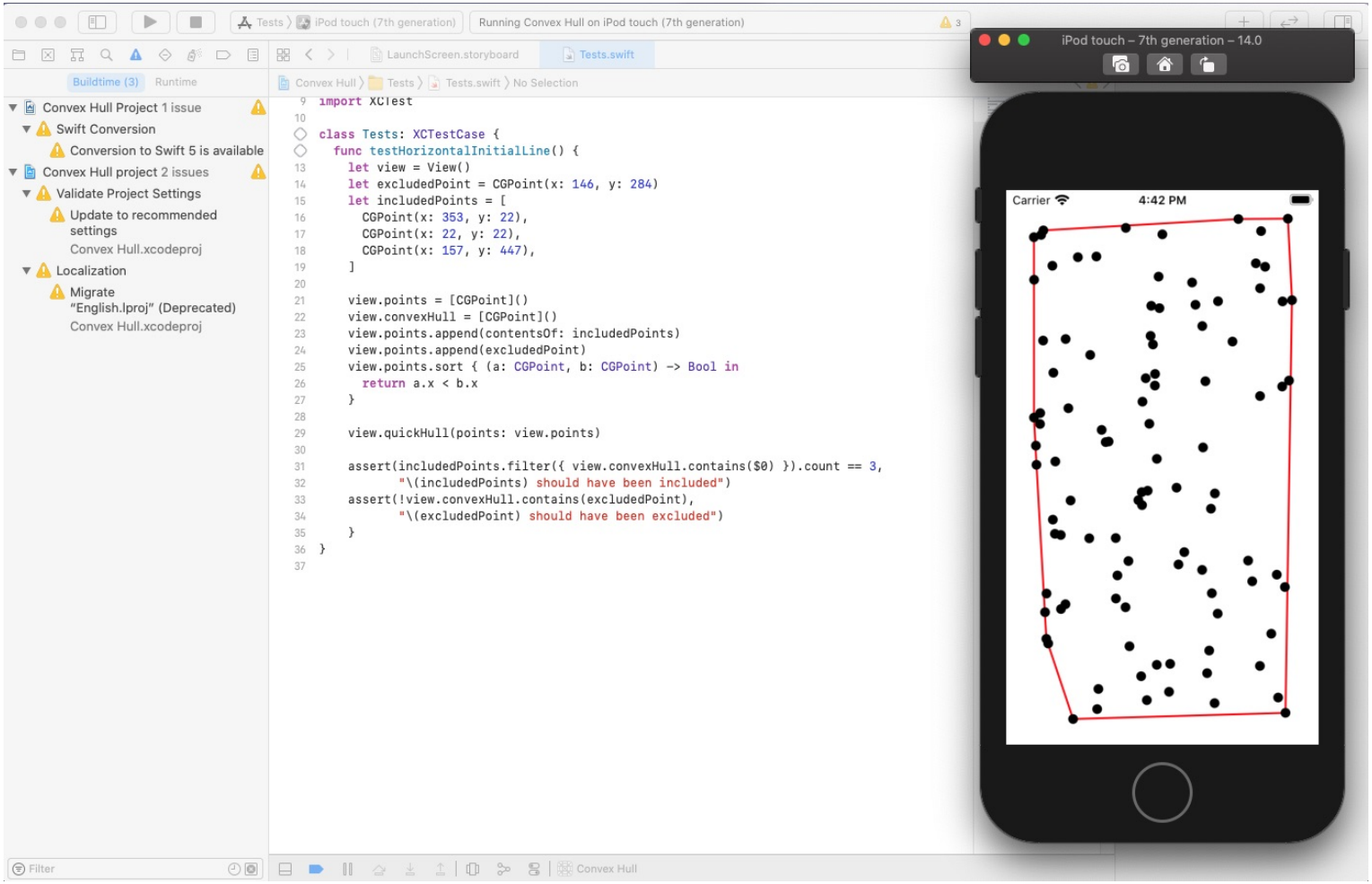
APIs give you transparent access to all low-level TensorFlow operators.

High-quality tooling

Building upon Jupyter and LLDB, Swift in Colab improves your productivity with helpful tooling such as context-aware autocomplete.

TensorFlow Software: Learning Swift

Xcode: Data Structures and Algorithms, iPhone Simulator ...



Working with Anaconda

Anaconda Individual Edition

A collection of open-source packages to perform data science in Python/R. Anaconda's packages include: SciPy, Jupyter, NumPy, TensorFlow, Pandas, matplotlib, and many many more, all under one roof.

Data Science Capability:

- Neural networks.
- Machine learning.
- Data visualization.
- Predictive analytics.

Download: https://www.anaconda.com

Appendix B

Java Software Setup

Java Software and Tools Setup (On a Mac)

Install Java and Compilation Software Tools

- Download Java: https://www.oracle.com/java/technologies/downloads/
- Install Homebrew and Apache Ant on your computer.

Data Mining and Machine Learning

- Install Weka and DeepLearning4J.

Next Steps

- Install LaTeX (useful for writing papers/documents).
- Install Xfig (useful for creating diagrams).
- Install Xcode (useful for development of iPhone Apps).
- Install an Integrated Development Environment (IDE).

Homebrew (On a Mac)

Homebrew

- Automates installation of software on your Mac.
- Homebrew is extremely useful!

Homebrew Download and Installation

- Go to the homebrew web site: https://brew.sh
- Open a terminal window. Type the command: `prompt >> printenv` to check that you are running the bash shell.
- Cut and paste the installation command into your terminal window.
- Look in the folder `/usr/local` to check that homebrew has been successfully installed and a Cellar has been created.
- Check that brew is accessible: `prompt >> which brew`.

Apache Ant (On a Mac)

Apache Ant (Another Neat Tool)

- Apache Ant is a software tool for automating software build processes.
- Implemented in Java and requires the Java Platform.
- Uses XML to describe the code build processes and their dependencies.

Ant Download and Installation

- From the terminal window type:

```
prompt >> brew install ant
```

- Check that ant is in your classpath, i.e.,

```
prompt >> which ant
```

- That's all!.

Appendix C

Working with Weka

Introduction

WEKA

WEKA (Waikato Environment for Knowledge Acquisition) is a workbench for data mining and machine learning.

Software Download and Installation

- WEKA is written in Java, so it will run on both PCs and Macs.
- Download from: https://www.cs.waikato.ac.nz/weka/

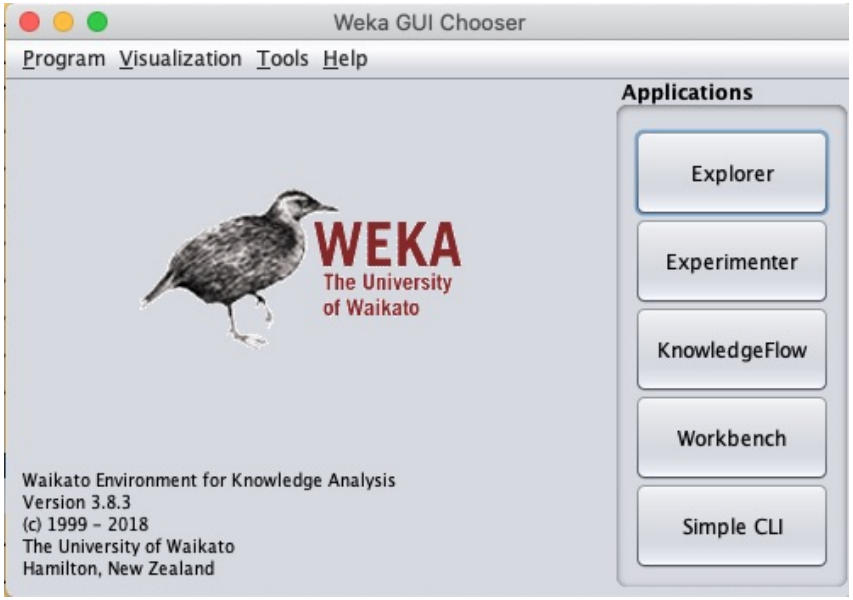
Online Resources

- See class web page for evolving list of links to WEKA resources ...
- Videos learning machine learning with WEKA are available on YouTube.

Getting Started

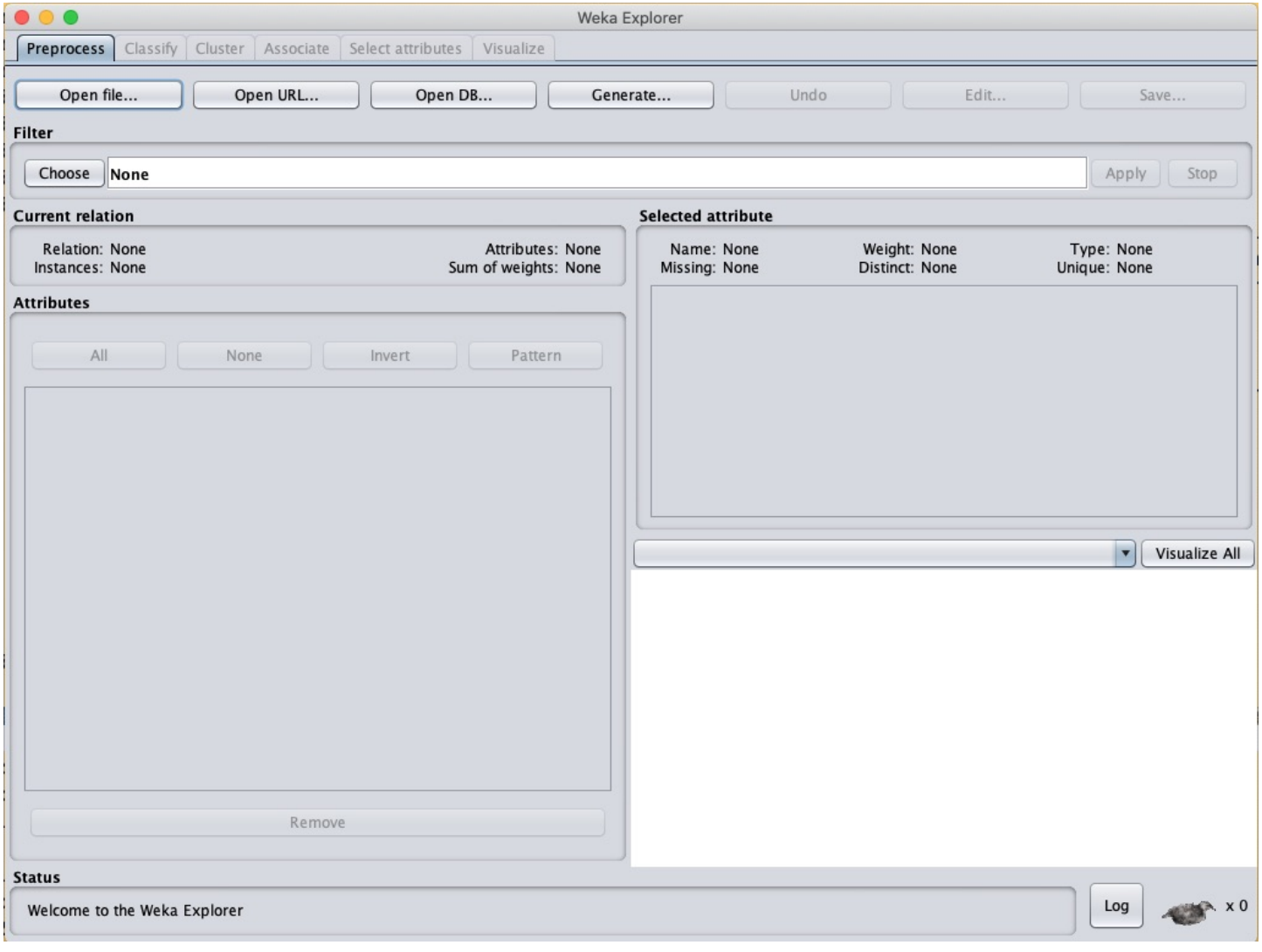
From the Terminal Window

```
prompt >> java -jar weka.jar
```

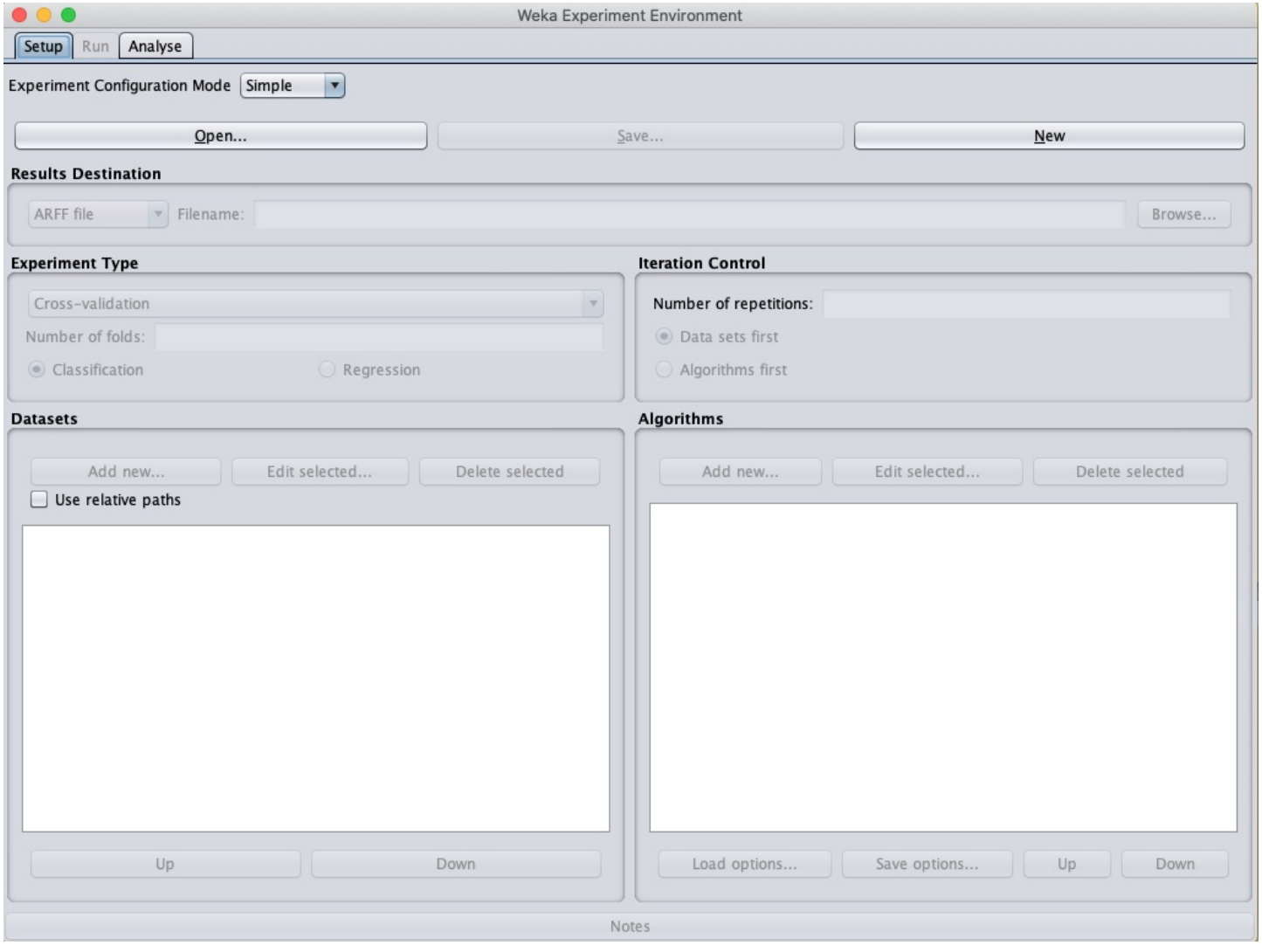


You can also write and run custom applications through the WEKA API.

Weka GUI Explorer



Weka GUI Experimenter



Appendix D

Working with Apache DataVec

Working with Apache DataVec

Apache DataVec

- Apache DataVec is an open source Java library for machine learning ETL.
- ETL operations transform raw data into usable vector formats that can be fed to machine learning algorithms.
- Apache DataVec has builtin transformation tools to convert and normalize data.

Data Schema

- A data schema is a high-level blueprint for how a data source (or database) is organized.
- Can think of the schama as being a logical model for how a data model (or database) will be configured.

Working with Apache DataVec

Data Transformation Operations:

- Read data in a variety or formats (e.g., csv, text, image).
- Remove unnecessary columns; rename columns.
- Filter data to keep only examples having specific values (e.g., "NZ" or "USA").
- Conditionally replace invalid values with new values computed by an external function.
- Convert categorical data into integers and one-hot encodiings.
- Parsing a data string and extracting lower-level detail (day, hr, min).

Source Code: See: java-code-ml-dl4j2021/src/datavec/

Working with Apache DataVec

Example 1: Consider the abbreviated data file:

Working with Apache DataVec

Data Schema:

Working with Apache DataVec

Setup Data Transformation Process:

Working with Apache DataVec

Execute Data Transformation Process:

Working with Apache DataVec

Transformed Data Format:

Appendix E

Working with DeepLearning4J

Working with DeepLearning4J (DL4J)

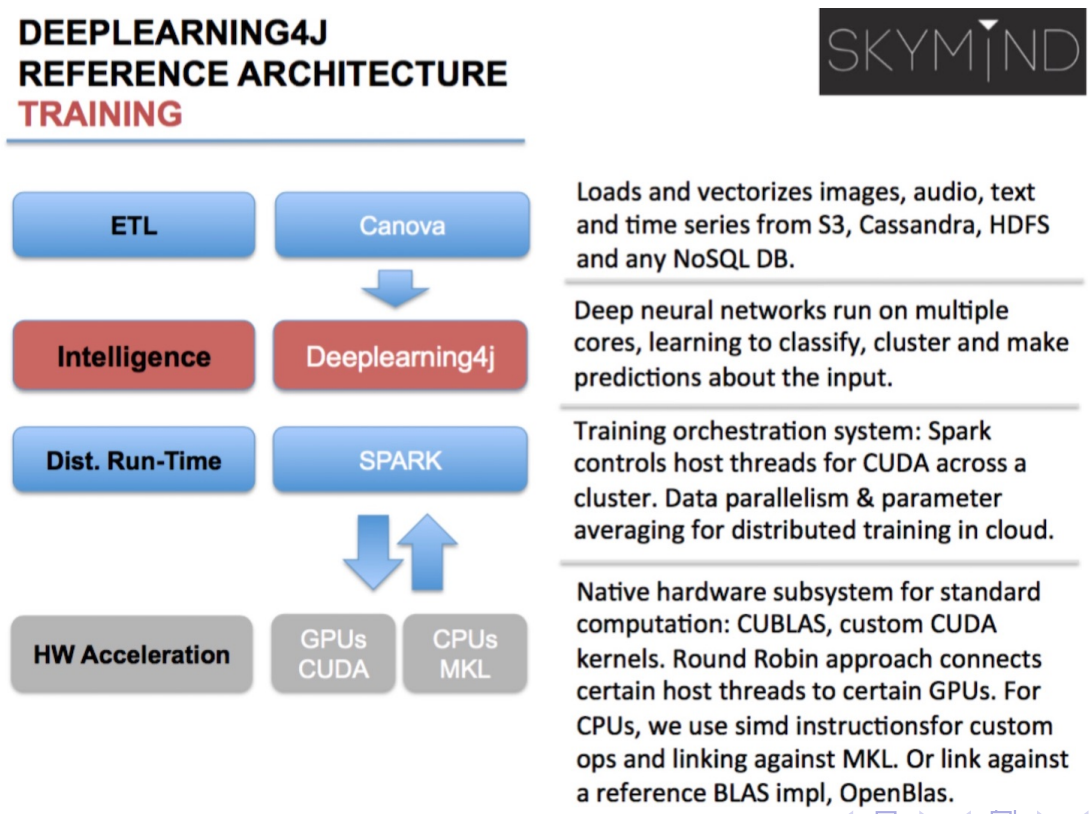
DeepLearning4J (Open-Source Java Library for Deep Learning)

- DeepLearning4J is the only deep learning programming library written in Java for the Java Virtual Machine (JVM).
- It provides wide support for development of deep learning algorthms including: autoencoders, recurrent networks, word2vec, etc.
- Can compose deep neural networks from shallow neural networks, each of which forms a so-called layer.
- Can combine variational autoencoders, sequence-to-sequence autoencoders, recurrent networks, convolution networks.
- Includes a n-dimensional array class called ND4J that is roughly equivalent to NumPy in Python.
- Keras will serve as the Python API.

Working with DeepLearning4J (DL4J)

DeepLearning4J (Open-Source Java Library for Deep Learning)

- Provides support for scalable deep learning with clusters of processors, Spark and GPUs.



Appendix F

Working with Apache Spark

Introduction to Apache Spark

What is Apache Spark?

Installation (On a Mac)

Simplest Example