From Notebook to Production Code

Christopher Messier

General Assembly

Washington, DC

June 1, 2018

Overview

1. Introduction
2. Code Conversion
3. Build File
4. Command Line Interfaces
5. Bonus Features

About Me (Chris Messier)

- Data Scientist at MainStreet Bank
- BAH Data Science Fundamentals IA
- Background in academic Economics
- Research Interests
  - NLP
  - Computer Vision
  - Probabilistic Programming
  - Graphical Models

What Does a Data Scientist *Produce*?

- Analytics
    - Explaining the data and giving recommendations

What Does a Data Scientist *Produce*?

- Analytics
  - Explaining the data and giving recommendations
- Research
  - Develop a deep understanding of a topic given the data

What Does a Data Scientist *Produce*?

- Analytics
  - Explaining the data and giving recommendations
- Research
  - Develop a deep understanding of a topic given the data
- Solutions
  - Developing software that enables data-driven solutions

What Is Production Code?

Production Code is...

- Code used in a software product

Why Worry About Production Code?

Production

What will be covered

- Using the commandline
  - bash profiles/scripting
- Converting Code
  - Notebook → Python files
- Best Practices
  - Project structure
  - Code style

What won't be covered

There's limited time available so we'll be skipping over...

- Optimization
- Unit Testing
- Packaging
- So much more...

Why the preference for Mac?

- ~~Aesthetics?~~
- ~~Hardware?~~
- UNIX operating system

UNIX

- A (large) family of multi-task operating systems
- Started development in 1969 at Bell Labs
- The foundation for all Linux flavors
- macOS is the most installed UNIX base

UNIX *cont.*

Linux is the choice of most production workstations/servers.

What does this mean?

- wide user/support base
- similar file systems for development/deployment
- similar backends/core libraries
- minimal changes between dev/deploy
- common interfaces (*bash*)

bash

A UNIX shell command language

- Default login shell for macOS and most Linux
- Defines the interactions with the command line
- Enables you to...
  - Navigate the file system
  - Launch programs

bash profile

A way to customize your bash environment

- customize aesthetics
- create shortcuts

Your bash profile can be found at:

- `~/.bash_profile`

Getting Started

We'll be working with the project here:

`www.github.com/messiest/pytorch-image-classifiers`

- What it is:
  - An implementation of the LeNet on *MNIST* data
  - An implementation of the GoogLeNet on *CIFAR10* data

Accessing the Example Code

The code is already included as a submodule to this repo. To populate the `pytorch-image-classifiers/` directory use the following commands:

- `git init submodule`
- `git submodule update`

What's wrong with a Notebook?

An important tool, but have disadvantages

- Non-executable
- No returns
- JavaScript Bloat (*ewwww*)
- "*slow*"

Converting Your Code

Exporting your code to a Python (`.py`) file

- File → Download As → Python (.py)
- Move file from Downloads to the Project directory
- Clean out notebook ”artifacts”

Build File

Install all packages required for a project

- `requirements.txt` - list of all packages
- Could be either a Python (`.py`) or Bash (`.sh`) script
- Ensures that there is a common environment for all users
- Good time to use a *virtual environment*

Command Line Interface

Interacting with your code through the command line

- Makes your code more flexible
- Gives greater control over code behavior
- Provides a common interface for

argparse

Part of the Python Standard Library (included w/ Python)

- Allows you to define custom arguments
- Provides help read-outs by default
- Easy to use, and descriptive

Progress Bar

Provides a simple way to track your program's progress

- Estimate how long your program will run
- Track values of important variables
- Provides a professional look/feel

tqdm

Provides a wrapper to output progress to command line

- Wraps an iterable, and automatically writes output
- Provides a way to customize the progress bar

Thank You