Westtown ML Codebook

Ben Drucker

Contents:

I. Large-scale overview

II. Introduction and goals of machine learning

III. Common ML models

IV. Lab

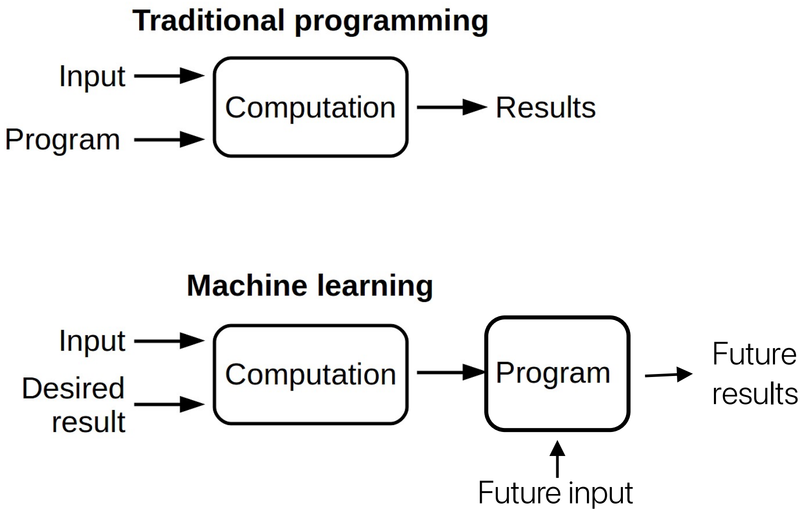
V. Extra resources

Large-scale overview

- The ML Codebook consists of the following components: this user guide, a student's GitHub repository, and an instructor's GitHub repository.
- This user guide is intended as an instructor's guide to teaching introductory machine learning.
- The student repository contains a Jupyter Notebook lab on handwriting recognition and associated datasets and other miscellaneous files.
- The instructor repository contains the same files, but the Jupyter Notebook is completely filled in (it is essentially an answer key).
- TODO: platform – cocalc or something else?

Introduction and Goals of ML

- Traditional computing vs. machine learning
  - View diagram to the right.
  - Traditionally, we are given an input and an accompanying program to solve a problem. A computer then generates desired results.
  - In ML we are also given an input, but instead of having an accompanying program, we are given human-generated desired results. These two entities allow the *computer* to *generate the program* (and future results).
  - To sum up, traditionally humans write the programs and the computer generate the results. In ML, humans generate initial results and the computer generates the program to compute future results.



Source Link

Introduction and Goals of ML

- Mechanics of ML
    - In ML, there are two common types of tasks: classification and regression
        - In classification, we are given information about an entity and desire to classify the entity into several discrete categories.
        - In regression, we are also given information about an entity but desire a continuous, numerical output.
    - ML can be broken into supervised and unsupervised learning.
        - Supervised learning occurs when we have a dataset containing many datapoints along with with associated human-labeled classifications. The computer uses this information to generate a mapping from data to classifications. This is the type of ML that is discussed in this codebook.
        - Unsupervised learning occurs when our dataset does not contain *labeled* examples. Here, we can only look at the data to generate classifications.

Introduction and Goals of ML

- ML Example Task
  - Suppose we work at a marble factory.
  - Our job is to **classify** marbles as “good” or “bad” quality.
  - Suppose we have many attributes about each marble, but it is difficult for humans to generate a rule/program that determines whether a marble is good or bad based on these attributes. An example dataset is below.

Features

Labels/Targets

	Diameter (cm)	Color	Material	Mass (g)	…	Has Swirl Pattern	**Class (Good/Bad)**
Marble #1	5	Red	Glass	40	…	True	**Bad**
Marble #2	3	Blue	Plastic	45	…	False	**Good**
Marble #3	2.5	Red	Glass	32	…	False	**Good**
…	…	…	…	…	…	…	**…**

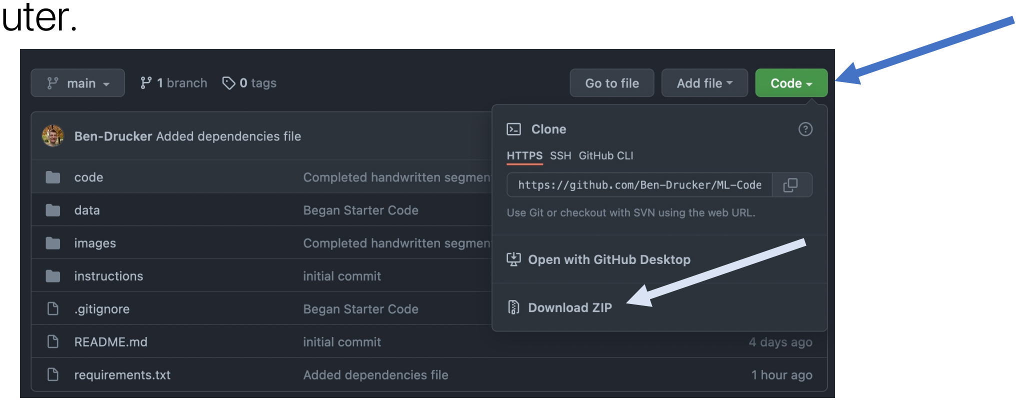
Examples

Common ML Models

- K-Nearest-Neighbors
- Linear Regression

Lab: setup

- For ease of use, I have found it would be best to use Colab as the lab platform. Colab uses a Jupyter Notebook (.ipynb — interactive python notebook) format.
- Setup instructions:
  - First, we need to download the git repository.
  - Go to the repository link (TODO)
  - Click the green "code ▼" button and then select "Download ZIP." Save the zip folder on your computer.



Ben-Drucker Added dependencies file

code	Completed handwritten segmen	
data	Began Starter Code	
images	Completed handwritten segmen	
instructions	initial commit	
.gitignore	Began Starter Code	
README.md	initial commit	4 days ago
requirements.txt	Added dependencies file	1 hour ago

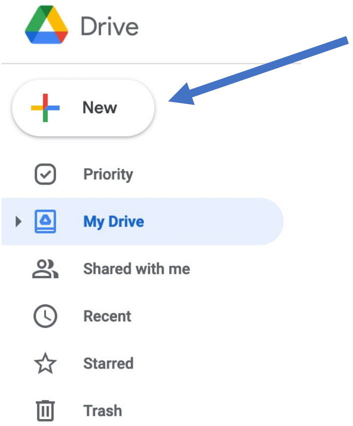
https://github.com/Ben-Drucker/ML-Code

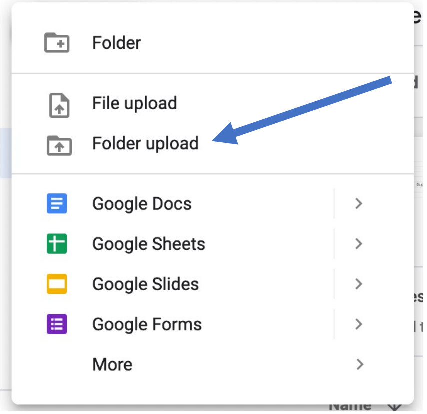
Use Git or checkout with SVN using the web URL.

  - On your computer, unzip the downloaded zip folder.
  - Upload to Google Drive (instructions on next slide)

Lab: setup

- Setup Instructions Continued
  - Go to your Google Drive. And select the "+ New" button at the upper right corner. Then choose "Folder upload." Upload the un-zipped folder you downloaded from the previous slide.

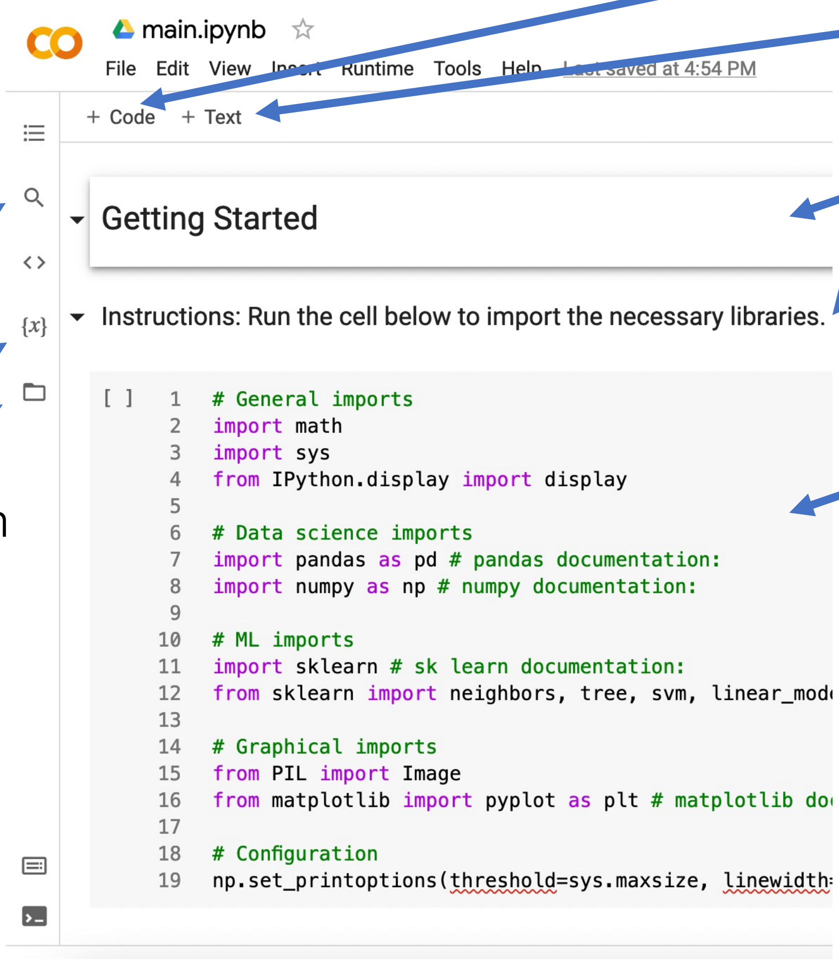




  - Open this folder in Google Drive. Open the "code" folder within. Double click the "main.ipynb" file. This will open Colab, where you can begin editing.

Lab: setup

- Tour of Colab


main.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:54 PM
+ Code + Text

Getting Started

Instructions: Run the cell below to import the necessary libraries.

```
[ ]  # General imports
     import math
     import sys
     from IPython.display import display

     # Data science imports
     import pandas as pd # pandas documentation:
     import numpy as np # numpy documentation:

     # ML imports
     import sklearn # sk learn documentation:
     from sklearn import neighbors, tree, svm, linear_mod

     # Graphical imports
     from PIL import Image
     from matplotlib import pyplot as plt # matplotlib do

     # Configuration
     np.set_printoptions(threshold=sys.maxsize, linewidth
```

Add Code Cell

Add Text Cell

Text Cells

Find and Replace

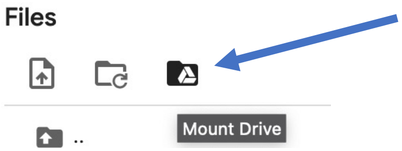
View Variables

File System

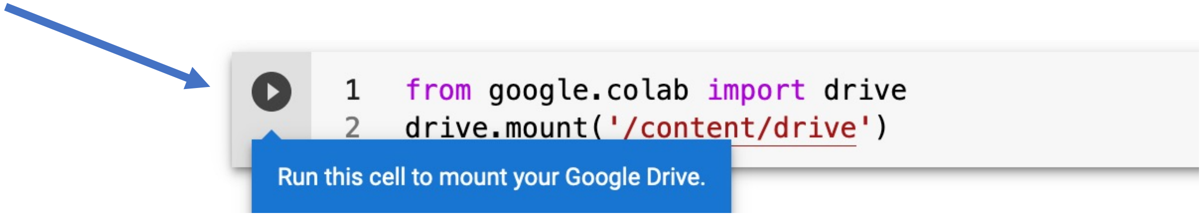
Code Cell

Lab: setup

- Colab uses google drive as a file management system. To view an access repository files, complete the following steps:
  - Click the file system icon on the left panel (🗀)
  - Click on the “Mount Google Drive” icon



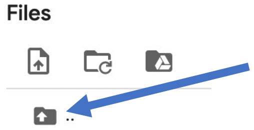
  - Click the “▶” button associated with the added cell with message “Run this cell to mount your Google Drive.” Enable permissions.



```
from google.colab import drive
drive.mount('/content/drive')
```

Run this cell to mount your Google Drive.

  - All Google Drive files (including the upload folder) are available by selecting the “up one level” option from the file system menu



and then navigating to content > drive > MyDrive.

Lab: Frameworks

- The lab portion of this codebook employs the following common ML/data science python libraries:
  - Sklearn — Robust machine learning library that implements many ML models
  - Pandas — Data manipulation library
  - Numpy — Math and data structure library
  - Scipy — Math library