Exploration: Introduction to Jupyter Notebooks (w1)

Introduction

We can build machine learning (ML) solutions in any programming language from scratch, using data, statistics and linear algebra. But over the past decade, ML practitioners have converged on a handful of fundamental tools: the **Python** (https://www.python.org) programming language, Python libraries (**NumPy** (https://numpy.org), **pandas** (https://pandas.pydata.org), **Matplotlib** (https://matplotlib.org), and others), and an interactive, mixed-media, experiment-driven programming environment known as **Jupyter Notebooks** (https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html).

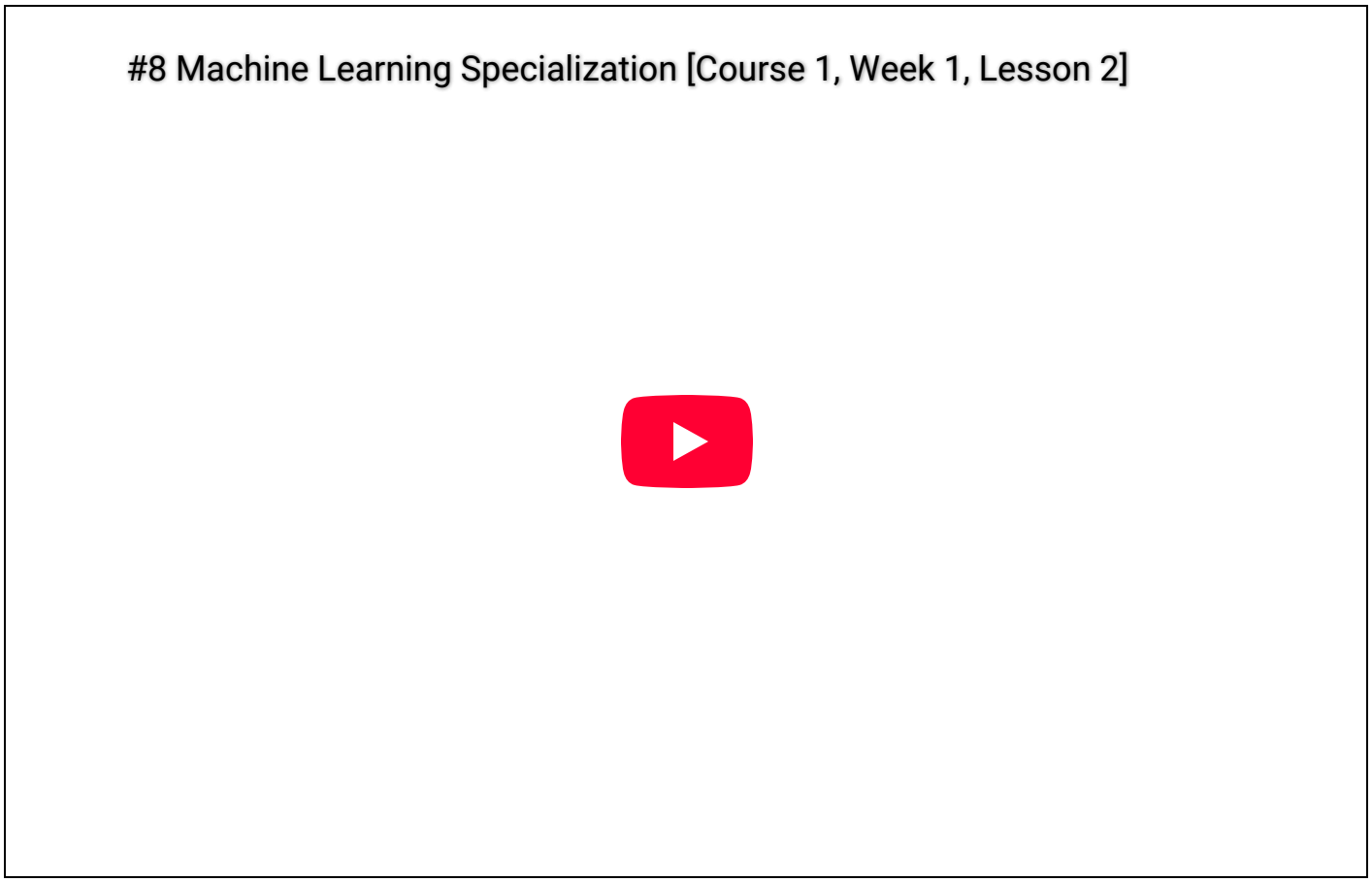
In this exploration, we will take a look at the basics of Jupyter Notebooks, and see how we can write prose and code to create, train and validate our ML models.

Jupyter Notebooks

Project Jupyter (https://jupyter.org) is an open source platform for running web application servers that deliver interactive programming environments, known as "notebooks," for data scientists and ML practitioners. A Jupyter Notebook enables us to load data, write prose, and write code for exploring and analyzing data, training ML models, and running experiments for validating and tuning our ML models. Jupyter Notebooks are awesome!

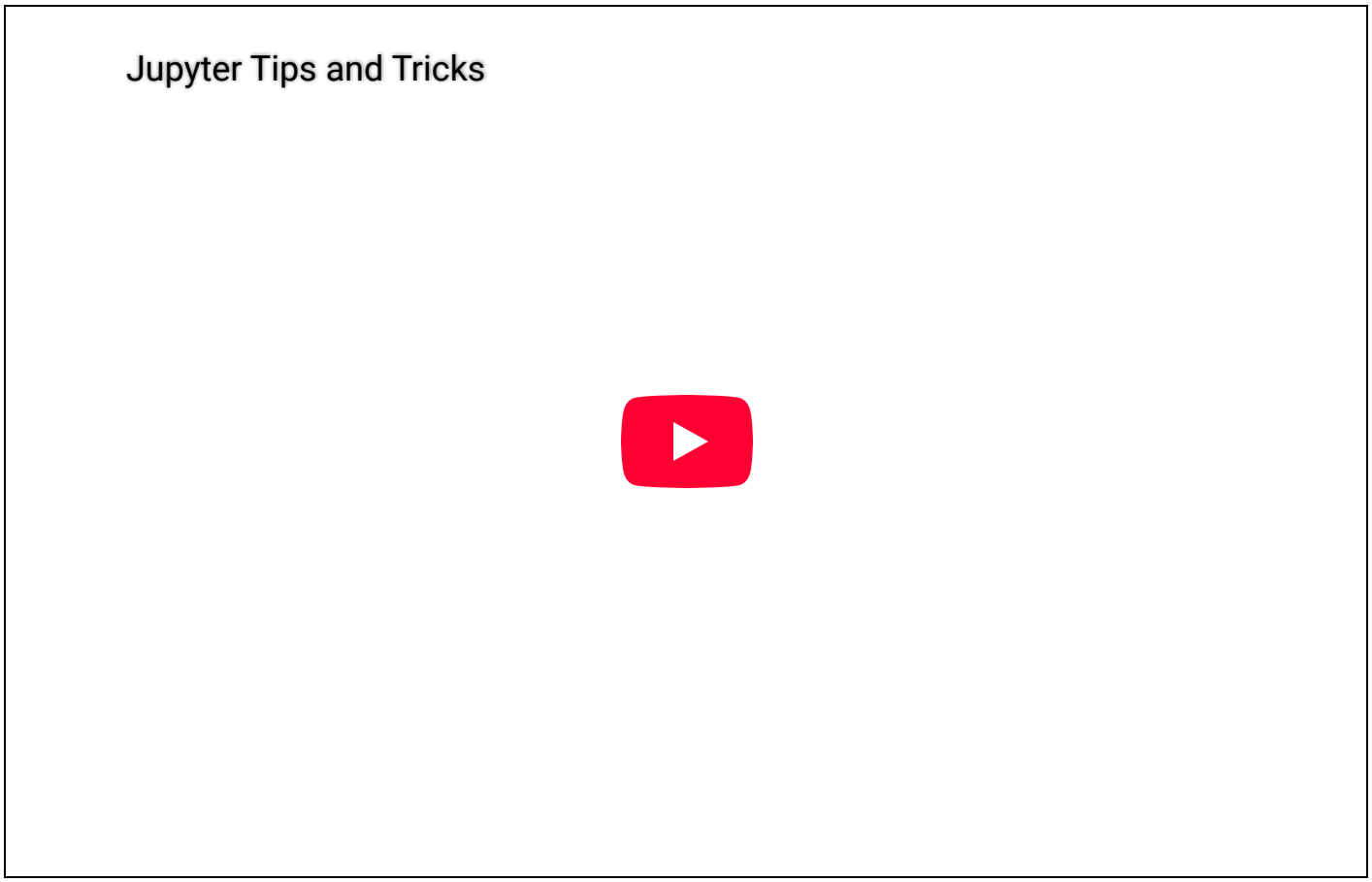
While you can run a Jupyter server yourself, many vendors are now hosting Jupyter Notebooks that are ready for practitioners to use. Some of these "brands" of notebooks include **Google Colaboratory** (https://colab.research.google.com), **Deepnote** (https://deepnote.com), and **Kaggle** (https://www.kaggle.com). While each vendor may provide different handy features on top of the Jupyter Notebook, the core features are all the same.

Let's take an initial look at a simple Jupyter Notebook.



A notebook consists of two general kinds of content known as *cells*: **Markdown (https://www.markdownguide.org/)** cells for formatted text, and code cells for runnable code. Both cells are editable, and the code cells are executable.

Let's take a look at running some Python code within a notebook, along with some tips and tricks for working with code in Jupyter Notebooks.

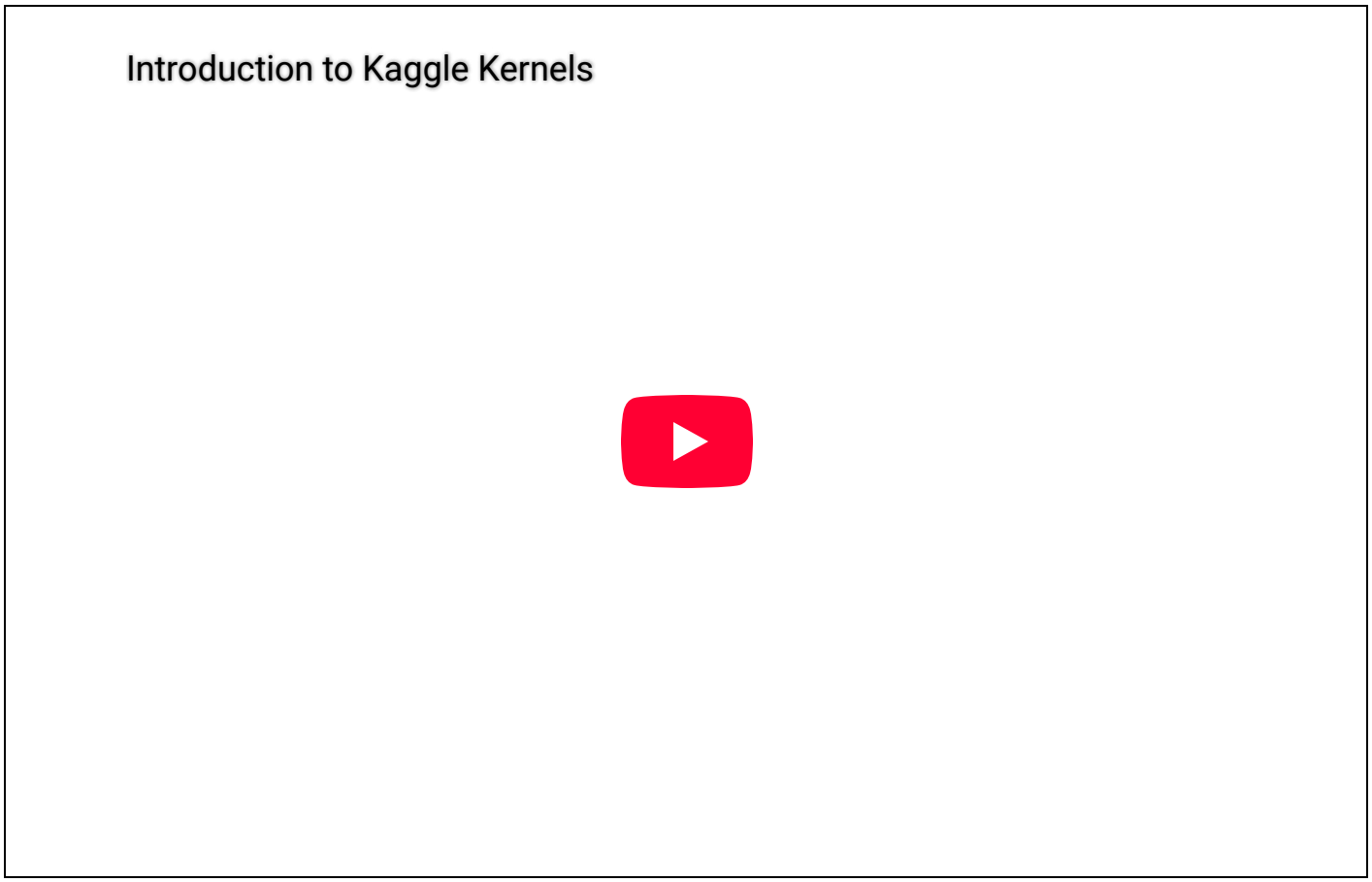


This presentation gives us a glimpse of some of the interesting, handy features of Jupyter Notebooks. In this course, our goal is to master the fundamentals well, so do not feel like you have to learn every single aspect of the Jupyter platform. We'll have many opportunities to practice working with notebooks in this course.

One of the most popular providers of Jupyter Notebooks is **Kaggle (https://www.kaggle.com/)**, the worldwide "home" of machine learning that each of us will become more familiar with in this course. Kaggle, however, calls these notebooks *kernels*, which is the more formal name for the Python runtime that is responsible for hosting your programming environment and running your code. You can think of a *kernel* as being all of the computing machinery behind the scenes of a notebook, that is responsible for running your code and displaying its results on the screen. But when you hear the term "Kaggle Kernel," just interpret that to mean "a Jupyter Notebook hosted by Kaggle."

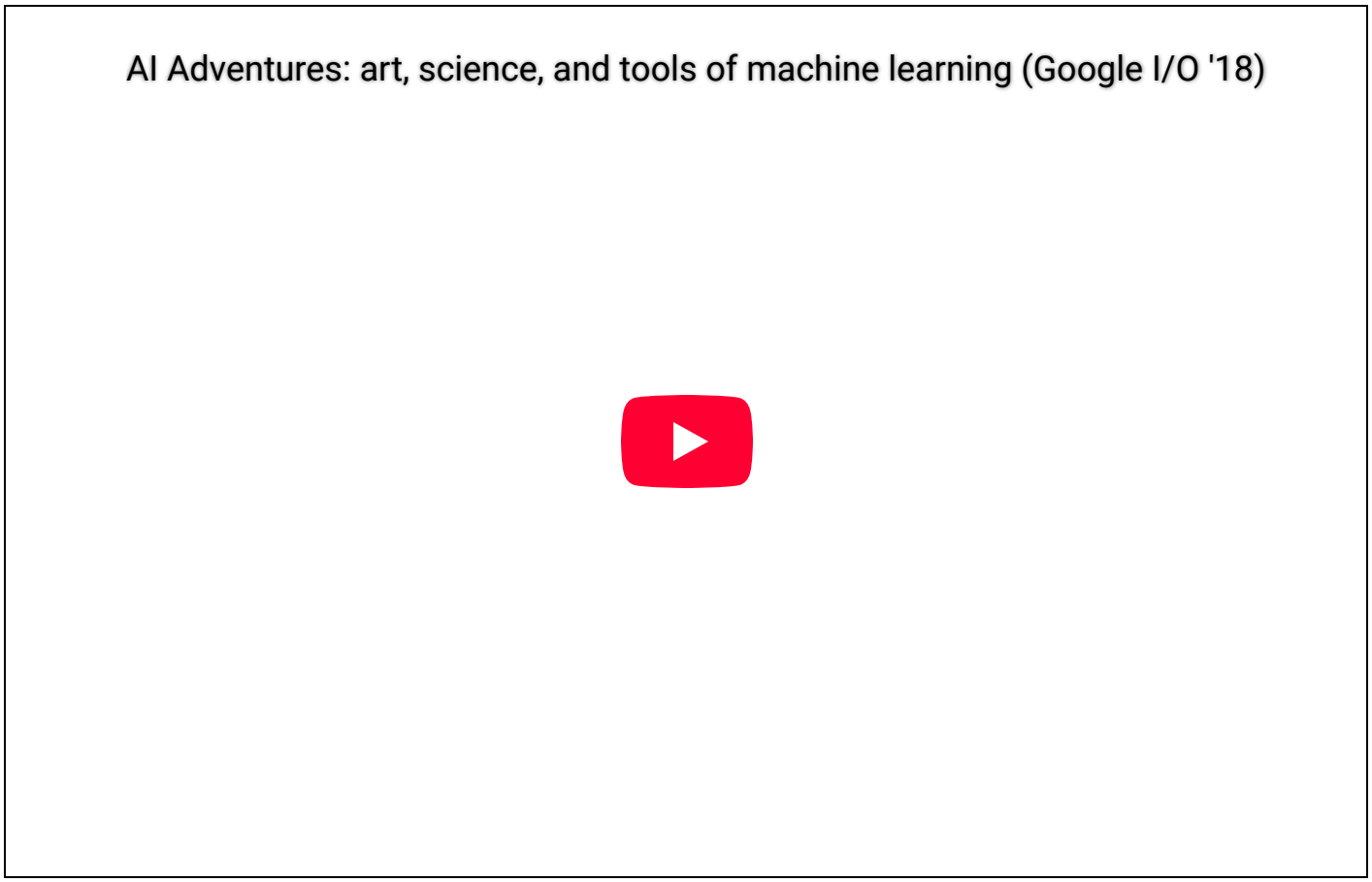
Note that Jupyter Notebooks and the vendor-specific flavors of notebooks have rapidly evolving features. You may find that the user interface of your particular notebook environment is different than what you see here, or in a book or online tutorial. Despite the visual differences, keep in mind that the concepts and features of notebooks remain the same, despite a button or menu being in a different place or appearing differently on the screen.

Let's take a deeper look at Kaggle Kernels, and see their features in action with the Fashion MNIST dataset.



With Jupyter Notebooks, and especially Kaggle Kernels, we can immediately load a dataset, import Python libraries, and begin our ML work. This is the power of Jupyter Notebooks - they make it easy for practitioners to carry out the ML process. Notice how the presenter also used the following popular Python libraries: **pandas (https://pandas.pydata.org)** and **Matplotlib (https://matplotlib.org)**. We'll learn more about these important libraries soon.

Let's take one more deeper look at Kaggle Kernels in action, and see how we can use Kaggle Kernels to collaborate with other people. (Just watch 25:30 to 29:56.)



We have seen how ML involves a process, and as practitioners we train, validate and experiment with ML models. Jupyter Notebooks provides us with a powerful, easy-to-use environment for conducting our ML process. But most importantly, it enables us to save, share and collaborate with others on our ML experiments - all without having to create such environments from scratch on our own.

Key Points

- The ML process requires data, mathematics and a programming environment, which we can build from scratch, but this can be time-consuming
- A Jupyter Notebook is an online, interactive programming environment tailored for machine learning practitioners
- A notebook consists of formatted text (in Markdown) and runnable code (usually in Python)
- A hosted notebook, such as Kaggle Kernels, makes it efficient to load data, explore and analyze data, train a model, validate a model, run ML experiments, and share our experiment results with other people

Check Your Understanding

Before you continue, please respond to the following questions and prompts to verify your understanding. Click any ▶ disclosure arrow to view a possible answer. ***[Accessible version for***

screen-reading software] (https://canvas.oregonstate.edu/courses/2025514/pages/exploration-questions-and-answers)

1. ▶ True or False: We must use the Python programming language to create ML models
2. ▶ What do Google Colaboratory, Deepnote and Kaggle all have in common?
3. ▶ What are the two primary kinds of content cells in a Jupyter Notebook?
4. Define the following terms.
    1. ▶ Markdown
    2. ▶ Kernel

Additional Resources

Consider these curated resources an *essential*, if not mandatory, starting point for a deeper investigation into the topics within this Exploration.

- Tutorial: **All About Markdown (https://www.kaggle.com/code/bakosy/all-about-markdown)** forked from hyu_sunwoong
- Reading: **The Markdown Guide (https://www.markdownguide.org)** by Matt Cone
- Reading: **Jupyter Project Documentation (https://docs.jupyter.org/en/latest/)**
- Reading: **Python 3.x Documentation (https://docs.python.org/3/)** by Python Software Foundation
- Tutorial: **Intro to Machine Learning (https://www.kaggle.com/learn/intro-to-machine-learning)** by Kaggle
- Library: **pandas (https://pandas.pydata.org)**
- Library: **Matplotlib (https://matplotlib.org)**
- Library: **NumPy (https://numpy.org/)**