Creating the Workspace - Jupyter Notebooks

First Things First: Why Python?

Of all the languages out there, why is Python the most popular choice in the machine learning and data science world? We like things that are simple and intuitive. Python is just that, simple and intuitive. It's readable, it's low in complexity, and it's easy to learn.
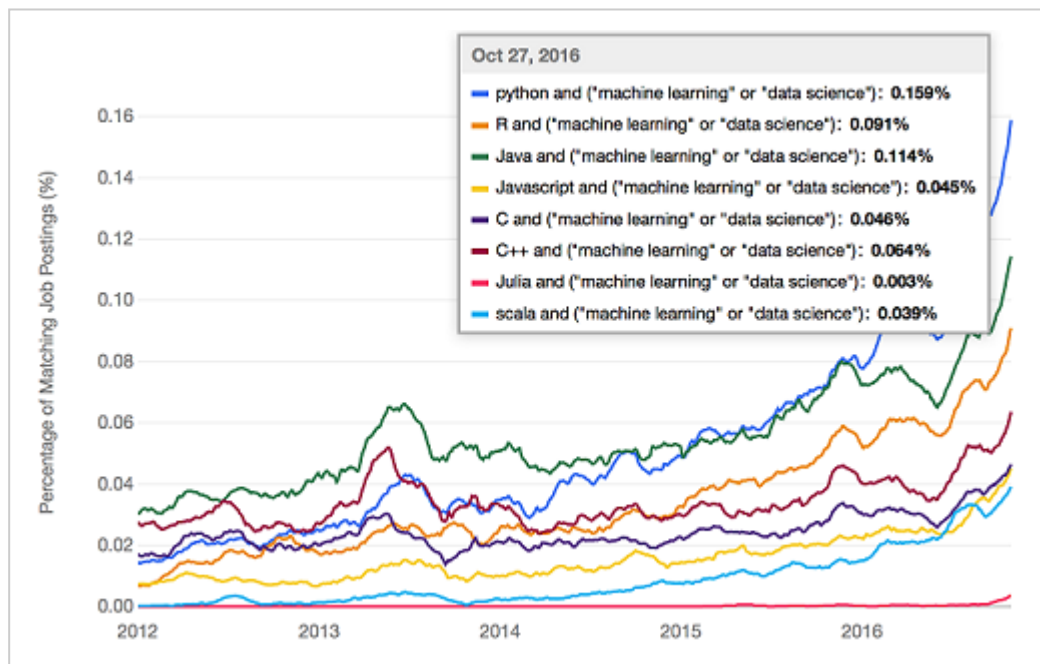
As a data scientist or machine learning engineer you can use a quick implementation in Python to validate your ideas about some complex mathy concepts in a fast and hassle-free manner. It's easily understandable for you and others.

As a Data Scientist, your life revolves around data. Outside the playground, you stumble upon reality. Data in real life is oftentimes raw, unstructured, incomplete, and large. Python comes with the promise of knowing how to handle these issues. But how does it do that? What is so special about Python?

All hail the mighty **packages!** What's so special about Python are the great open source code repositories that are continuously being updated. These open source contributions give Python its superpowers and an edge over

other languages. The best thing about using these packages is that they have a **minimal learning curve**. Once you have a basic understanding of Python, you can very easily import, use, and benefit from all of the packages out there without having to understand everything is going on under the hood. Last but not the least, these packages are completely free to use as well!

Since we have our data scientist hat on, let's talk about data. If you still have doubts, I'll let this survey from IBM convince you why we should learn data science in Python and not in R or any other language.
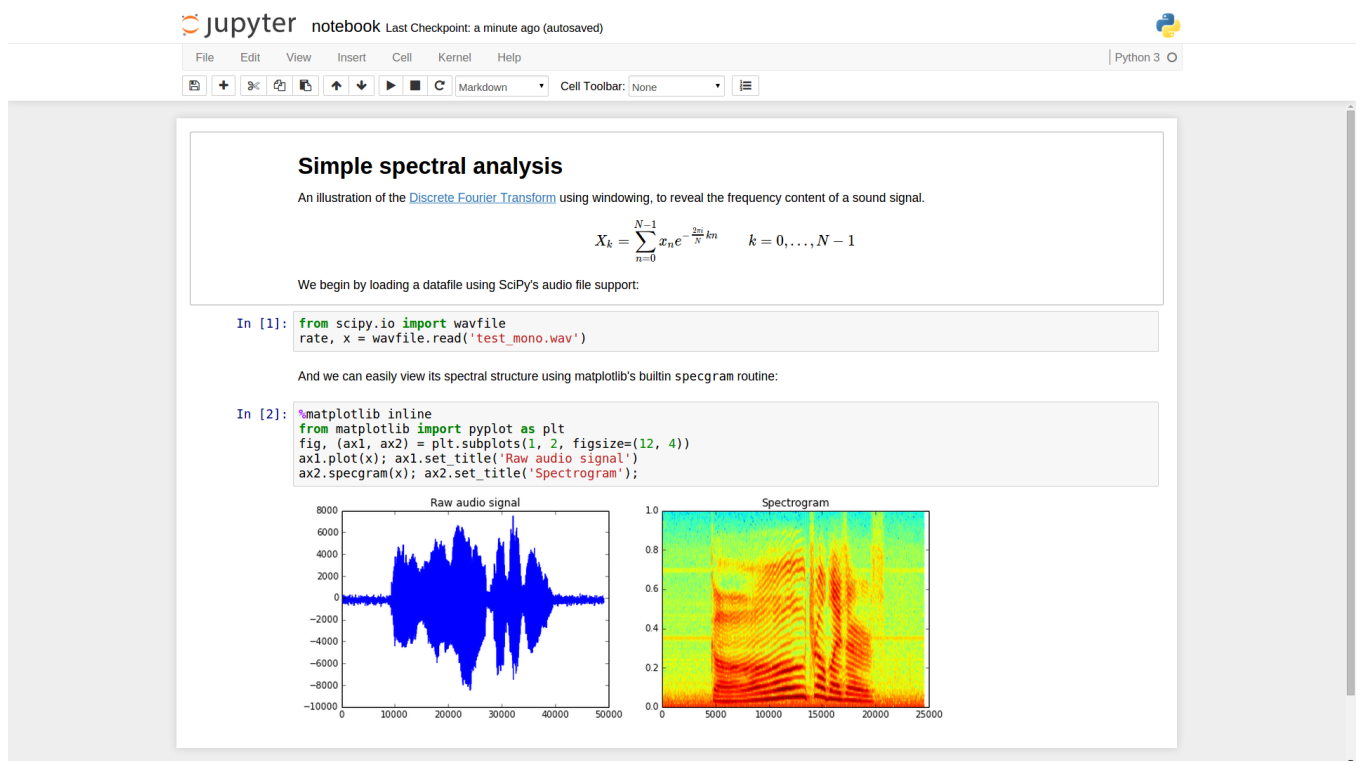

The Most Popular Language For Machine Learning Is ...

Jupyter Notebook: What It Is and Why Data Scientists Love It

This story begins with IPython. IPython is an interactive command-line terminal for Python. Command-line terminals are not everyone's cup of tea, so in 2011 IPython introduced a new tool named the Notebook — a modern and powerful web interface to Python. In 2015 the Notebook project was re-branded as the Jupyter Notebook.

The Jupyter Notebook is an incredibly powerful and sleek tool for **developing and presenting** data science projects. It can integrate code and its output into a single document, combining visualizations, narrative text, mathematical equations, and other rich media. It's simply awesome.

Simple spectral analysis

An illustration of the Discrete Fourier Transform using windowing, to reveal the frequency content of a sound signal.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \qquad k = 0, \ldots, N-1$$

We begin by loading a datafile using SciPy's audio file support:

```
In [1]: from scipy.io import wavfile
        rate, x = wavfile.read('test_mono.wav')
```

And we can easily view its spectral structure using matplotlib's builtin `specgram` routine:

```
In [2]: %matplotlib inline
        from matplotlib import pyplot as plt
        fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
        ax1.plot(x); ax1.set_title('Raw audio signal')
        ax2.specgram(x); ax2.set_title('Spectrogram');
```

As if all these features weren't sufficient enough, Jupyter Notebook can handle many other languages, like R, as well. Its **intuitive workflows, ease of use, and zero-cost** have made it *THE* **tool at the heart of any data science project**.

Essentially, Jupyter is a great interface to the Python language and a must-have for all Data Science projects.

How to Install Jupyter Notebook

1. Running Jupyter Notebooks With the Anaconda Python Distribution

The easiest and most straight forward way to get started with Jupyter Notebooks is by installing Anaconda. Anaconda is the most widely used Python distribution for data science. It comes pre-loaded with the most popular libraries and tools (e.g., NumPy, Pandas, and Matplotlib). What this means is that immediately get to real work, skipping the pain of managing tons of installations, dependencies, and OS-specific installation issues.

2. Getting Anaconda

Follow these simple steps:

1. Download the latest version of Anaconda for Python 3 (ignore Python 2.7).

2. Install Anaconda by following the instructions on the download page and/or in the executable.

Note: There are other ways of running the Jupyter Notebook as well, e.g., via pip and Docker, but we are going to keep it sweet and simple. We will stick with the most hassle-free approach.

Creating Your First Jupyter Notebook

- Background information? ✔
- Installations? ✔

Let's get started with a real Jupyter Notebook.

> 📌 **Note**: You do **not** need to go through the installation process right now. At the end of this lesson there is an in-built Jupyter Notebook for you to play with, **without having to leave this page!**

Run the following command from your Anaconda terminal:

```
jupyter notebook
```

A Jupyter server is now running in your terminal, listening to port 8888, and it will automatically launch the application in your default web browser at *http://localhost:8888* if it doesn't happen automatically, you can use the url to launch it yourself. You should see your workspace directory, like in the screenshot below:
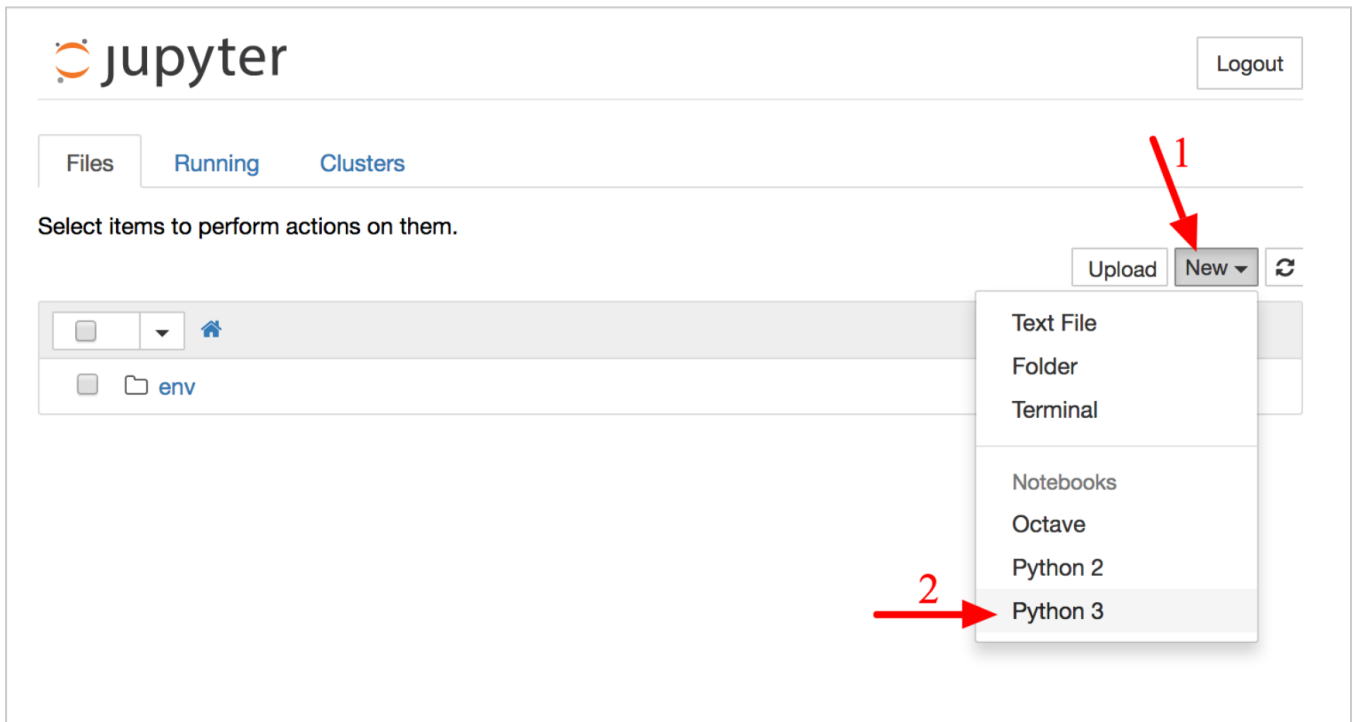

You can create a new Python notebook by clicking on the New [1] button (screenshot below) and selecting the appropriate Python version (Python3,

This does three things for you:

1. It creates a new Notebook file called Untitled.ipynb in your workspace.

2. It starts a Jupyter Python kernel to run this notebook.

3. It opens the newly created notebook in a new tab.

As a "Hello-World-step" you can rename your notebook to "Hello Data Science" by clicking Untitled [1] and typing the new name, as shown in the snapshot below.

A notebook contains a list of cells. Each cell can contain executable code or formatted text (Markdown). Right now, the notebook contains only one empty code cell. Try typing print("Hello Data Science!") in the cell[2], then click on the run button [3] (or press Shift-Enter). Hitting the run button sends the current cell to this notebook's Python kernel, which runs it and returns the output. The result is displayed below the code cell:

```
In [ ]:  ▶|
```
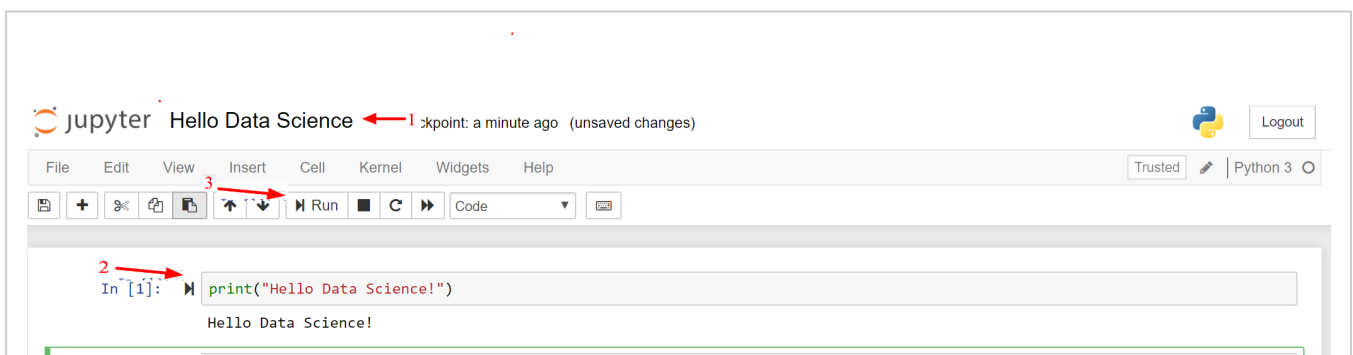
Jupyter Notebook

You can try this here:

> 📝 **How to Use a Jupyter NoteBook?**
>
> - Click on **"Click to Launch"** 🚀 button to work and see the code running live in the notebook.
>
> - You can click ⧉ to open the **Jupyter Notebook in a new tab**.
>
> - Go to files and click *Download as* and then choose the format of the file to **download** ⎙. You can choose Notebook(.ipynb) to download the file and work locally or on your personal Jupyter Notebook.
>
> - ⚠ The notebook **session expires after 30 minutes of inactivity**. It will reset if there is no interaction with the notebook for 30 consecutive minutes.

Your app can be found at: https://5lrqw92v88k2y-live-app.educative.run/notebooks/helloworld.ipynb ⧉

Click to launch app!

Congratulations! Now you have your first Jupyter Notebook up and running. You will need it for the IMDB and end-to-end ML project that are discussed later in this course.

Bonus Tip

When **working with other people on a data science project**, it is a good practice to:

1. Have explicit rules for naming all the documents.
2. Use a version control system like Git to share and save your progress.