towards
data science

OPINION

5 Reasons why you should Switch from Jupyter Notebook to Scripts

Using Scripts Helps me Realize the Drawbacks of Jupyter Notebook

Khuyen Tran Follow
Aug 24 · 7 min read ★

Florian Salihovic
2 days ago

You could easily simply import the code from Python files into the notebook. A better way would even to create a library and import the code as package. Notebooks are not inherently messy and code can be organized similarly to files in the file syste...

Read More

62 1

Liam Pearson
1 day ago

What I got from this article was: use functions. And you can use functions in jupyter notebooks. I'm sticking with notebooks.

36 1

Kevin Kirchhoff
1 day ago

You could just use an IDE or editor and turn on autoreload so you don't need to rerun the entire notebook.

15 1

Jupyter
s a medium to
ode in Jupyter

ce is that
e 'Shift +
for us to

ore data

- **Unorganized**: As my code gets bigger, it becomes increasingly difficult for me to keep track of what I write. No matter how many markdowns I use to separate the notebook into different sections, the disconnected cells make it difficult for me to concentrate on what the code does.

- **Difficult to experiment:** You may want to test with different methods of processing your data, choose different parameters for your machine learning algorithm to see if the accuracy increases. But every time you experiment with new methods, you need to rerun the entire notebook. This is time-consuming, especially when the processing procedure or the training takes a long time to run.

- **Not ideal for reproducibility:** If you want to use new data with a slightly different structure, it would be difficult to identify the source of error in your notebook.

- **Difficult to debug:** When you get an error in your code, it is difficult to know whether the reason for the error is the **code** or the change in **data**. If the error is in the code, which part of the code is causing the problem?

- **Not ideal for production:** Jupyter Notebook does not play very well with other tools. It is not easy to run the code from Jupyter Notebook while using other tools.

I knew there must be a better way to handle my code so I decided to give scripts a try. These are the benefits I found when using scripts:

Organized

The cells in Jupyter Notebook make it difficult to organize the code into different parts. With a script, we could create several small functions with each function specifies what the code does like this

Better yet, if these functions could be categorized in the same category such as functions to process the data, we could put them in the same class!

Whenever we want to process our data, we know the functions in the class `Preprocess` can be used for this purpose.

Encourage Experiment

When we want to experiment with a different approach to preprocess data, we could just add or remove a function by commenting out like this without being afraid to break the code! Even if we happen to break the code, we know exactly where to fix it.

We could also experiment with different parameters by changing the input of the functions. For example, if we want to see how different methods of resampling my Pandas series affect my results, we could just switch from `method_of_resample='sum’` to `method_of_resample= 'average'`. How neat!

You can still use functions in a notebook, but when your number of functions gets really big, you might want to split the functions in different notebooks. Importing functions across different notebook is not easy.

Ideal for Reproducibility

With classes and functions, we could make the code general enough so that it will be able to work with other data.

For example, if we want to drop different columns in my new data, we just need to change `columns_to_drop` to a list of columns, we want to drop and the code will run smoothly!

I can also create a pipeline that specifies steps to process and train the data! Once I have a pipeline, all I need to do is to use

```
pipline.fit_transform(data)
```

to apply the same processing to both the train and test data.

Easy to Debug

With functions, it is easier to test whether that function produces the output we expect. We can quickly spot out where in the code we should change to produce the output we want

If all of the tests pass but there is still an error in running our code, we know the data is where we should look next.

For example, after passing the test above, I still have a TypeError when running the script, which gives me the idea that my data has null values. I just need to take care of that to run the code smoothly.

Ideal for Production

We can use different functions in multiple scripts on top of something else like this

or to add a config file to control the values of the variables. This prevents us from wasting time tracking down a specific variable in the code just to change its value.

We could also easily add tools to track the experiment such as MLFlow or tools to handle configuration such as Hydra.cc!

I didn't like the Idea of Using Jupyter Notebook until I Pushed myself out of my Comfort Zone

I used to use Jupyter Notebook all the time. When some data scientists advise me to switch from Jupyter Notebook to script to prevent some problems listed above, I didn't understand and felt resistant to do so. I didn't like the uncertainty of not being able to see the outcome when I run the cell.

But the disadvantage of Jupyter Notebook grew as I started my first real data science project in my new company so I decided to push myself out of my comfort zone and experiment with scripts.

In the beginning, I felt uncomfortable but started to notice the benefits of using scripts. I started to feel more organized when my code is organized into different functions, classes, and into multiple scripts with each script serving different purposes such as preprocessing, training, and testing.

So are you Suggesting me to Stop Using Jupyter Notebook?

Don’t get me wrong. I still use Jupyter Notebook if my code is small and if I don’t plan to put my code into production. I use Jupyter Notebook when I want to explore and visualize the data. I also use it to explain how to use some python libraries. For example, I write use mostly Jupyter Notebooks in this repository as the medium to explain the code mentioned in all of my articles.

If you don’t feel comfortable with coding everything in scripts, you could use both scripts and Jupyter Notebook for different purposes. For example, you could create classes and functions in scripts then import them in the notebook so that the notebook is less messy.

Another alternative is to turn the notebook into the script after writing the notebook. I personally don't prefer this approach because it often takes me longer to organize the code in my notebook such as put them into functions and classes and write test functions.

I find writing a small function then writing a small test function is faster and safer. If I happen to want to speeds up my code with the new Python library, I could use the test function I already wrote to make sure it still works as I expected.

With that being said, I believe there are more ways to solve the disadvantage of Jupyter Notebook than what I mentioned here such as how Netflix uses put the notebook into production and schedule the notebook to run at a certain time.

Conclusion

Everybody has their own way to make their workflow more efficient and to me, it is to leverage the utility of scripts. If you have just switched from Jupyter Notebook to script, it might not be intuitive to write code in scripts, but trust me, you will get used to using scripts eventually.

Once that happens, you will start to realize many benefits of the scripts over the messy Jupyter Notebook and want to write most of your code in scripts.

If you don’t feel comfortable with the big change, start small.

Big changes start with small steps

I like to write about basic data science concepts and play with different algorithms and data science tools. You could connect with me on LinkedIn and Twitter.

Star this repo if you want to check out the codes for all of the articles I have written. Follow me on Medium to stay informed with my latest data science articles like these

How to Learn Data Science when Life does not Give You a Break

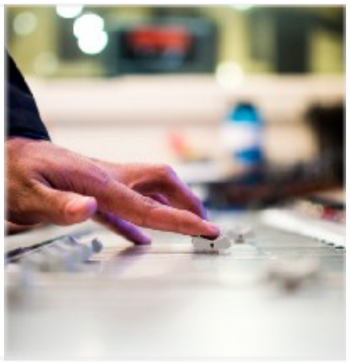
I Struggled to Dedicate Time for Data Science. But Finding new Strategies Enables me to Boost my Learning Rate and...

towardsdatascience.com

How to Efficiently Fine-Tune your Machine Learning Models

Find it Time-consuming to Find the Best Parameters for your ML Models? Use these Three Tricks

towardsdatascience.com



Introduction to IBM Federated Learning: A Collaborative Approach to Train ML Models on Private Data

How to Keep the Data Secured while Training the Data from Different Sources?

towardsdatascience.com

How to Create Reusable Command-Line

Can you Wrap your Multiple Useful Command Lines into one File for Quick Execution?

towardsdatascience.com