Like
Tweet
Share
0
Pin it

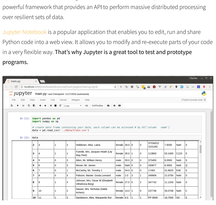
JANUARY 19, 2020

How to install PySpark and Jupyter Notebook in 3 Minutes

Spark is a fast and powerful framework.

Charles Bochet
Data Scientist

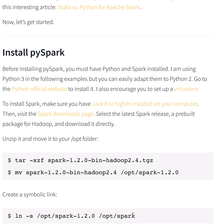
5 min read



Spark with Jupyter

Apache Spark is a must for Big data's lovers. In a few words, Spark is a fast and powerful framework that provides an API to perform massive distributed processing over resilient sets of data.

Jupyter Notebook is a popular application that enables you to edit, run and share Python code into a web view. It allows you to modify and re-execute parts of your code in a very flexible way. That's why Jupyter is a great tool to test and prototype programs.



Jupyter Notebook running Python code

I wrote this article for Linux users but I am sure Mac OS users can benefit from it too.

Why use PySpark in a Jupyter Notebook?

While using Spark, most data engineers recommends to develop either in Scala (which is the "native" Spark language) or in Python through complete PySpark API.

Python for Spark is obviously slower than Scala. However like many developers, I love Python because it's flexible, robust, easy to learn, and benefits from all my favorites libraries. In my opinion, Python is the perfect language for prototyping in Big Data/Machine Learning fields.

If you prefer to develop in Scala, you will find many alternatives on the following github repository: alexarchambault/jupyter-scala

To learn more about Python vs. Scala pro and cons for Spark context, please refer to this interesting article: Scala vs. Python for Apache Spark.

Now, let's get started.

Install pySpark

Before installing pySpark, you must have Python and Spark installed. I am using Python 3 in the following examples but you can easily adapt them to Python 2. Go to the Python official website to install it. I also encourage you to set up a virtualenv.

To install Spark, make sure you have Java 8 or higher installed on your computer. Then, visit the Spark downloads page. Select the latest Spark release, a prebuilt package for Hadoop, and download it directly.

Unzip it and move it to your /opt folder:

```
$ tar -xzf spark-1.2.0-bin-hadoop2.4.tgz
$ mv spark-1.2.0-bin-hadoop2.4 /opt/spark-1.2.0
```

Create a symbolic link:

```
$ ln -s /opt/spark-1.2.0 /opt/spark
```

This way, you will be able to download and use multiple Spark versions.

Finally, tell your bash (or zsh, etc.) where to find Spark. To do so, configure your $PATH variables by adding the following lines in your ~/.bashrc (or ~/.zshrc) file:

```
export SPARK_HOME=/opt/spark
export PATH=$SPARK_HOME/bin:$PATH
```

Install Jupyter Notebook

Install Jupyter notebook:

```
$ pip install jupyter
```

You can run a regular jupyter notebook by typing:

```
$ jupyter notebook
```

Your first Python program on Spark

Let's check if PySpark is properly installed without using Jupyter Notebook first.

You may need to restart your terminal to be able to run PySpark. Run:

```
$ pysparkWelcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.1.0
      /_/
Using Python version 3.5.2 (default, Jul  2 2016
17:53:06)
SparkSession available as 'spark'.
>>>
```

It seems to be a good start! Run the following program:
(I bet you understand what it does!)

```
import random
num_samples = 100000000

def inside(p):
  x, y = random.random(), random.random()
  return x*x + y*y < 1

count = sc.parallelize(range(0,
num_samples)).filter(inside).count()

pi = 4 * count / num_samples
print(pi)

sc.stop()
```

The output will probably be around 3.14.

PySpark in Jupyter

There are two ways to get PySpark available in a Jupyter Notebook:

- Configure PySpark driver to use Jupyter Notebook: running `pyspark` will automatically open a Jupyter Notebook
- Load a regular Jupyter Notebook and load PySpark using findSpark package

First option is quicker but specific to Jupyter Notebook, second option is a broader approach to get PySpark available in your favorite IDE.

Method 1 — Configure PySpark driver

Update PySpark driver environment variables: add these lines to your ~/.bashrc (or ~/.zshrc) file.

```
export PYSPARK_DRIVER_PYTHON=jupyter
export PYSPARK_DRIVER_PYTHON_OPTS='notebook'
```

Restart your terminal and launch PySpark again:

```
$ pyspark
```

Now, this command should start a Jupyter Notebook in your web browser. Create a new notebook by clicking on 'New' > 'Notebooks Python [default]'.

Copy and paste our Pi calculation script and run it by pressing Shift + Enter.

Jupyter Notebook: Pi Calculation script

Done!
You are now able to run PySpark in a Jupyter Notebook :)

Method 2 — FindSpark package

There is another and more generalized way to use PySpark in a Jupyter Notebook: use findSpark package to make a Spark Context available in your code.

findSpark package is not specific to Jupyter Notebook, you can use this trick in your favorite IDE too.

To install findspark:

```
$ pip install findspark
```

Launch a regular Jupyter Notebook:

```
$ jupyter notebook
```

Create a new Python [default] notebook and write the following script:

```
import findspark
findspark.init()

import pyspark
import random

sc = pyspark.SparkContext(appName="Pi")
num_samples = 100000000

def inside(p):
  x, y = random.random(), random.random()
  return x*x + y*y < 1

count = sc.parallelize(range(0,
num_samples)).filter(inside).count()

pi = 4 * count / num_samples
print(pi)

sc.stop()
```

The output should be:

Jupyter Notebook: Pi calculation

I hope this 3-minutes guide will help you easily getting started with Python and Spark. Here are a few resources if you want to go the extra mile:

- https://www.dezyre.com/article/scala-vs-python-for-apache-spark/213
- http://quabr.com/overview/comparing-interactive-solutions-for-running-apache-spark-zeppelin-spark-notebook-and-jupyter-scala
- http://spark.apache.org/docs/latest/api/python/index.html
- https://github.com/jadianes/spark-py-notebooks

And if you want to tackle some bigger challenges, don't miss out the more evolved JupyterLab environnement or the PyCharm integration of jupyter notebooks.

Thanks to Pierre-Henri Cumenge, Antoine Toubhans, Adil Baaj, Vincent Quagliaro, and Adrien Lina.

how to build a successful ai poc
thief
bridge

How To Build A Successful AI PoC
Turn Your Artificial Intelligence Ideas Into Working Software

Arnault Data Scientist — 9 min read

How to Perform Fraud Detection with Personalized Page Rank
This article shows how to perform fraud detection with Graph Analysis.

Antoine Data Scientist — 7 min read

Image Registration: From SIFT to Deep Learning
How the field has evolved from OpenCV to Neural Networks.

9 min read

SICARA

Contact
Sicara SAS
48 Bld des Batignolles
75017 Paris
contact@sicara.ai
UK : +44 20 3514 8322
US : +1 7070 406 3207

Legal infos
Terms
Privacy

Contact us

Your email
Your Phone Number
Your message
send

Sicara © 2016-2020 all rights reserved