

ILLINOIS DATA SCIENCE INITIATIVE

TECHNICAL REPORTS

---

## **Parsing Tweets using PySpark**

---

*Author:*  
Sameet Sapra & Joshua Chang

February 23, 2017

# Parsing Tweets using PySpark

**SAMEET SAPRA<sup>1</sup> AND JOSHUA CHANG<sup>2</sup>**

<sup>1</sup>National Center For Supercomputing Applications (NCSA)

<sup>2</sup>Laboratory for Computation, Data, and Machine Learning

<sup>3</sup>Illinois Data Science Initiative

Compiled February 23, 2017

---

**An introduction to using Apache Spark with Python by parsing tweets.**

<https://github.com/lcdm-uiuc>

---

## 1. SPARK INTRODUCTION: WHY PYSARK?

PySpark is an API that interfaces with RDD's in Python. It is built on top of Spark's Java API and exposes the Spark programming model to Python. PySpark makes use of a library called Py4J, which enables Python programs to dynamically access Java objects in a Java Virtual Machine. This allows data to be processed in Python and cached in the JVM.

## 2. PREREQUISITES TO PYSARK

Assume that the environment is a cluster with Hadoop installed. Verify that Java and Scala are installed by checking the versions. If they are installed, skip to the Spark configuration steps, otherwise follow the installation instructions given below.

Install Java on the system with:

```
> sudo yum install java-1.7.0-openjdk-devel
```

Install Spark on the system with:

```
> wget http://downloads.lightbend.com/scala/2.11.8/scala-2.11.8.rpm
> sudo yum install scala-2.11.8.rpm
```

Let's configure Spark to only show errors on startup, rather than all info. This will make the Spark shell less cluttered when it is opened on startup.

```
> cd /usr/hdp/2.3.6.0-3796/spark/conf
> vi log4j.properties
```

Find a line that describe the rootCategory of log:

```
log4j.rootCategory=INFO, console
```

Replace INFO with ERROR:

```
log4j.rootCategory=ERROR, console
```

## 3. SETTING UP PYSARK

Now let's ensure that python 2.7 is installed and configured.

```
> yum install -y centos-release-SCL
> yum install -y python27
> yum -y install python-pip
```

Finally, let's set up the environment variables for Spark and Python.

```
> vi ~/.bashrc
> export SPARK_HOME=/usr/hdp/2.3.6.0-3796/spark
> export PYTHONPATH=$SPARK_HOME/python
> export SPARK_HIVE=true
```

Then, reload the environment:

```
source ~/.bashrc
```

## 4. WRITING OUR FIRST LINES OF PYSARK

Once you've written the your PySpark, you can compile it with:

```
spark-submit --master yarn-cluster --num-executors 10 MY_PYTHON.
```