

## TASK

Load the complete Shakespeare writings, strip the header and search for the #24 most used word in his writings. Provide your code in one pdf, txt or by one link!

## THE CODE & THE ANSWER

Is public on kaggle: <https://www.kaggle.com/kewagbln/shakespeare-word-count-with-spark-python>

1 : the : 27572	13 : for : 8215
2 : and : 26752	14 : with : 7973
3 : i : 20191	15 : it : 7224
4 : to : 19338	16 : be : 6979
5 : of : 18135	17 : me : 6962
6 : a : 14520	18 : your : 6875
7 : you : 12991	19 : his : 6825
8 : my : 12468	20 : this : 6299
9 : that : 10964	21 : but : 6272
10 : in : 10914	22 : he : 6102
11 : is : 9503	23 : as : 5934
12 : not : 8453	24 : have : 5845

Future improvement: reading the text from line 245, not from 1.

## HISTORICAL SETUP DOCUMENTATION

### SETUP SPARK

#### TRY: INSTALL ON WINDOWS

<https://blog.sicara.com/get-started-pyspark-jupyter-guide-tutorial-ae2fe84f594f>

<https://medium.com/@GalarnykMichael/install-spark-on-windows-pyspark-4498a5d8d66c>

Doesn't work:

```
In [11]: import findspark
findspark.init()

import pyspark
import random

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

num_samples = 100

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()
```

```

Py4JavaError                                Traceback (most recent call last)
<ipython-input-11-dda98519d6e9> in <module>
    13     return x*x + y*y < 1
    14
--> 15 count = sc.parallelize(range(0, num_samples)).filter(inside).count()
    16
    17 pi = 4 * count / num_samples

~\Spark\spark-2.4.0-bin-hadoop2.7\python\pyspark\rdd.py in count(self)
    1053         3
    1054         """
-> 1055         return self.mapPartitions(lambda i: [sum(1 for _ in i)]).sum()
    1056
    1057     def stats(self):

```

<https://stackoverflow.com/questions/47761758/pyspark-python-issue-py4javaerror-an-error-occurred-while-calling-o48-showstr>

The error is

Caused by: java.lang.OutOfMemoryError: Java heap space

You need more memory to perform the operations and avoid the OOM error.

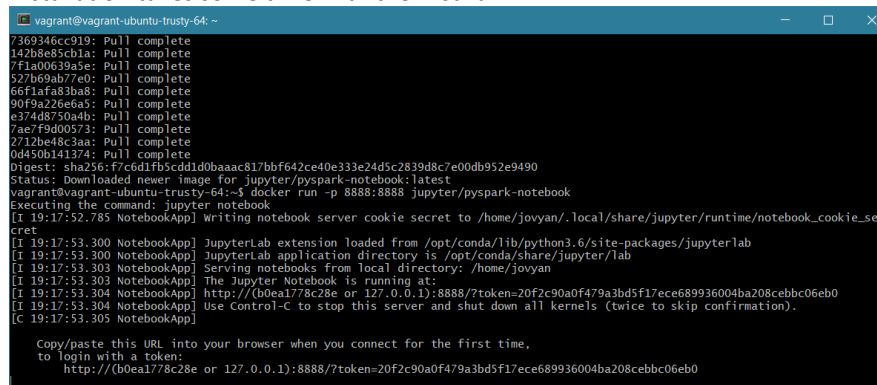
## TRY: DOCKER – PYSPARK

1. Setup a Vagrant VM: `vagrant init`
2. Setup a docker in the VM: `Refresh the vagrantfile / vagrant up / vagrant ssh`
3. Get an image: `docker pull jupyter/pyspark-notebook`

<https://hub.docker.com/r/jupyter/pyspark-notebook>

More information about the images: <https://jupyter-docker-stacks.readthedocs.io/en/latest/using/selecting.html>

4. Run docker: `docker run -p 4040:4040 jupyter/pyspark-notebook`  
Installation takes some time with the first run



```

vagrant@vagrant-ubuntu-trusty-64: ~
7369346cc919: Pull complete
142b8e85c31a: Pull complete
7f1a00639a5e: Pull complete
527b69ab77e0: Pull complete
66f1afa83ba8: Pull complete
90f9a226e6a5: Pull complete
e374d8750a4b: Pull complete
7ae7f9d00573: Pull complete
2712be48c3aa: Pull complete
0a450b141374: Pull complete
Digest: sha256:f7c6d1fb5cdd1d0baaac817bbf642ce40e333e24d5c2839d8c7e00db952e9490
Status: Downloaded newer image for jupyter/pyspark-notebook:latest
vagrant@vagrant-ubuntu-trusty-64:~$ docker run -p 8888:8888 jupyter/pyspark-notebook
Executing the command: jupyter notebook
[I 19:17:52.785 NotebookApp] Writing notebook server cookie secret to /home/jovyan/.local/share/jupyter/runtime/notebook_cookie_se
cret
[I 19:17:53.300 NotebookApp] JupyterLab extension loaded from /opt/conda/lib/python3.6/site-packages/jupyterlab
[I 19:17:53.300 NotebookApp] JupyterLab application directory is /opt/conda/share/jupyter/lab
[I 19:17:53.303 NotebookApp] Serving notebooks from local directory: /home/jovyan
[I 19:17:53.303 NotebookApp] The Jupyter Notebook is running at:
[I 19:17:53.304 NotebookApp] http://(b0ea1778c28e or 127.0.0.1):8888/?token=20f2c90a0f479a3bd5f17ece689936004ba208cebbc06eb0
[I 19:17:53.304 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 19:17:53.305 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://(b0ea1778c28e or 127.0.0.1):8888/?token=20f2c90a0f479a3bd5f17ece689936004ba208cebbc06eb0

```

5. Open notebook in browser:



---

TRY: DOCKER – GETTYIMAGES/SPARK

1. Setup a Vagrant VM: `vagrant init`
2. Setup a docker in the VM: Refresh the `vagrantfile`

If you just need to access a vagrant machine from only the host machine, setting up a "private network" is all you need. Either uncomment the appropriate line in a default `vagrantfile`, or add this snippet. If you want your VM to appear at `172.30.1.5` it would be the following:

```
config.vm.network "private_network", ip: "172.30.1.5"
```

192.168.3.10

<https://stackoverflow.com/questions/14870900/how-to-find-the-vagrant-ip>

`/vagrant up`

```
$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Checking if box 'ubuntu/trusty64' is up to date...
==> default: Resuming suspended VM...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
default: SSH address: 127.0.0.1:2222
default: SSH username: vagrant
default: SSH auth method: private key
==> default: Machine booted and ready!
==> default: Machine already provisioned. Run 'vagrant provision' or use the '--provision'
==> default: flag to force provisioning. Provisioners marked to run always will still run.
```

`/ vagrant ssh`

```
welcome to Ubuntu 14.04.5 LTS (GNU/Linux 3.13.0-164-generic x86_64)

* Documentation:  https://help.ubuntu.com/

System information as of Mon Feb  4 09:42:40 UTC 2019

System load:  0.02           Processes:            77
Usage of /:   6.7% of 39.34GB Users logged in:          0
Memory usage: 36%           IP address for eth0: 10.0.2.15
Swap usage:   0%            IP address for docker0: 172.17.0.1

Graph this data and manage this system at:
https://landscape.canonical.com/

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

New release '16.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Feb  4 09:38:19 2019 from 10.0.2.2
vagrant@vagrant-ubuntu-trusty-64:~$ docker --version
Docker version 18.06.1-ce, build e68fc7a
```

3. Get an image: `docker pull gettyimages/spark`

<https://hub.docker.com/r/gettyimages/spark/>

4. Run an example: `docker run --rm -it -p 4040:4040 gettyimages/spark bin/run-example SparkPi 10`

```
vagrant@vagrant-ubuntu-trusty-64:~$ docker run --rm -it -p 4040:4040 gettyimages/spark bin/run-example SparkPi 10
2019-02-04 05:29:57 WARN NativeCodeLoader:60 - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
2019-02-04 05:30:00 INFO SparkContext:54 - Running Spark version 2.4.0
2019-02-04 05:30:00 INFO SparkContext:54 - Submitted application: Spark Pi
2019-02-04 05:30:00 INFO SecurityManager:54 - Changing view acls to: root
2019-02-04 05:30:00 INFO SecurityManager:54 - Changing modify acls to: root
```

## Run spark

```
vagrant@vagrant-ubuntu-trusty-64:~$ docker run --rm -it -p 4040:4040 gettyimages/spark
2019-02-04 10:26:54 INFO Master:2566 - Started daemon with process name: 1@4d3607c8a93e
2019-02-04 10:26:54 INFO SignalUtils:54 - Registered signal handler for TERM
2019-02-04 10:26:54 INFO SignalUtils:54 - Registered signal handler for HUP
2019-02-04 10:26:54 INFO SignalUtils:54 - Registered signal handler for INT
2019-02-04 10:26:55 WARN NativeCodeLoader:60 - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
2019-02-04 10:26:55 INFO SecurityManager:54 - Changing view acls to: root
2019-02-04 10:26:55 INFO SecurityManager:54 - Changing modify acls to: root
```

## Run code – doesn't work:

The screenshot shows a Jupyter Notebook with a file explorer on the left containing files like 'Architectures', 'Data Preparation', 'NoSQL', 'Spark', and 'out'. The main area displays a code cell with the following Python code:

```
from pyspark import SparkContext, SparkConf

if __name__ == '__main__':

    conf = SparkConf().setAppName("create").setMaster("spark://172.17.0.2:7077")
    sc = SparkContext(conf=conf)

    inputStrings = ["Stefan 52", "Patrick 41", "Felix 43"]
    regularRDDs = sc.parallelize(inputStrings)

    pairRDDs = regularRDDs.map(lambda s: (s.split(" ")[0], s.split(" ")[1]))
    pairRDDs.coalesce(1).saveAsTextFile("out/RegularRDD2pairRDD")
```

The output of the code cell shows the following logs:

```
Run: spark_example x
C:\Users\kwagn\AppData\Local\Programs\Python\Python37-32\python.exe C:/Users/kwagn/Github/CS4BD-Ed1/Spark/spark_example.py
2019-02-04 12:49:30 WARN NativeCodeLoader:62 - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
2019-02-04 12:49:53 WARN StandaloneAppClient$ClientEndpoint:87 - Failed to connect to master 172.17.0.2:7077
org.apache.spark.SparkException: Exception thrown in awaitResult:
    at org.apache.spark.util.ThreadUtils$.awaitResult(ThreadUtils.scala:226)
    at org.apache.spark.rpc.RpcTimeout.awaitResult(RpcTimeout.scala:75)
    at org.apache.spark.rpc.RpcEnv$.setupEndpointRefFromURT(RpcEnv.scala:101)
```

<https://jaceklaskowski.gitbooks.io/mastering-apache-spark/spark-tips-and-tricks-running-spark-windows.html>

## TRY: KAGGLE

Works in <https://www.kaggle.com/kernels/notebooks/new?forkParentScriptVersionId=5495607>

```
[8]: from pyspark import SparkContext
from pyspark.sql import SparkSession
from pyspark.sql.functions import *
```

## But package missed in my notebook:

```
[10]: from pyspark import SparkContext
from pyspark.sql import SparkSession
from pyspark.sql.functions import *
```

Are we missing a package you need? You can add it on the Settings tab, and/or send us a pull request.

```
ModuleNotFoundError                               Traceback (most recent call last)
<ipython-input-10-84bac4128168> in <module>()
----> 1 from pyspark import SparkContext
      2 from pyspark.sql import SparkSession
      3 from pyspark.sql.functions import *

ModuleNotFoundError: No module named 'pyspark'
```

## Add custom package:

**Add a custom package**

Kaggle Kernels come preloaded with the most popular Python and R packages. If there's a package you'd like to use which isn't preinstalled, you can add it here.

In progress...

Custom packages

No custom packages installed

## Failed:

**Add a custom package**

Kaggle Kernels come preloaded with the most popular Python and R packages. If there's a package you'd like to use which isn't preinstalled, you can add it here.

Failed.

Custom packages

No custom packages installed

## Next try – pyspark installation:

```
[*]: !pip install pyspark

Collecting pyspark
  Downloading https://files.pythonhosted.org/packages/88/01/a37e827c2d80c6a754e40e99b9826d978b55254cc6c6672b5b88f2e18a7f/pyspark-2.4.0.tar.gz (213.4MB)
    100% |#####| 213.4MB 93kB/s eta 0:00:01 53.7MB/s eta 0:00:02
Collecting py4j==0.10.7 (from pyspark)
  Downloading https://files.pythonhosted.org/packages/e3/53/c737818eb9a7dc32a7cd4f1396e787bd94200c3997c72c1dbe028587bd76/py4j-0.10.7-py2.py3-none-any.whl (197kB)
    100% |#####| 204kB 30.8MB/s ta 0:00:01
Building wheels for collected packages: pyspark
Running setup.py bdist_wheel for pyspark ... /

[ ]: # This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load in

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
```

**draft** based on V1

1 committed version

V1 19m +25 -0

**Draft Environment**

[+ Add Data](#)

input (read-only)

shakespeare-online

t8.shakespeare.txt

**Settings**

Sharing Private, 0 collaborators

Language Python

Docker Latest available

GPU BETA GPU off

Internet **Internet connected**

Packages No custom packages