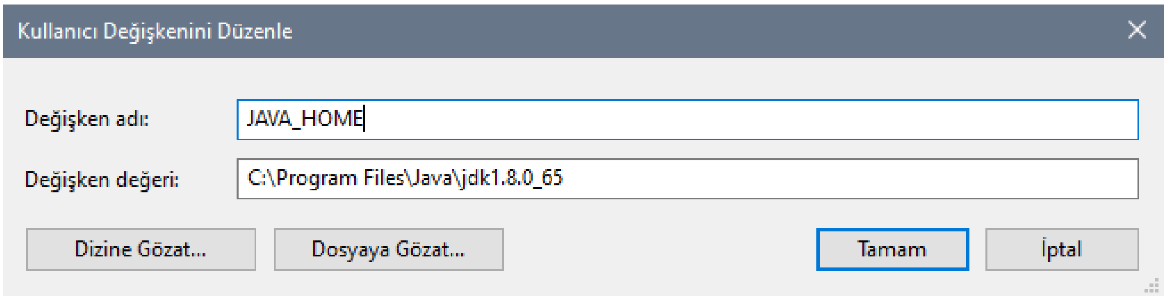
ADNAN MENDERES UNIVERSITY
CSE424 BIG DATA ANALYSIS

LAB 01
Apache Spark- Jupyter Notebook Installation with Python & Word Count Example

For this lab, you should submit the answer of **Homework** section.

1- Download **JDK** (Java Development Kit) from https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html. Choose correct installation file for your computer/OS requirements. After installation, add following new user variable and value to environment variables:

JAVA_HOME   C:\Program Files\Java\jdk1.8.0_65

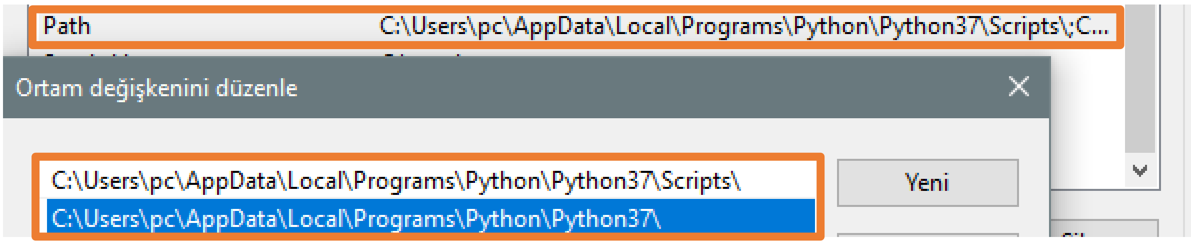


Kullanıcı Değişkenini Düzenle

Değişken adı: JAVA_HOME

Değişken değeri: C:\Program Files\Java\jdk1.8.0_65

2- Download latest version of **Python** from https://www.python.org/downloads/. Choose correct installation file for your computer/OS requirements. After installation, add python directory and Scripts directory into “path” user variable.



Path   C:\Users\pc\AppData\Local\Programs\Python\Python37\Scripts\;C...

Ortam değişkenini düzenle

C:\Users\pc\AppData\Local\Programs\Python\Python37\Scripts\
C:\Users\pc\AppData\Local\Programs\Python\Python37\

3- Download latest pre built version of **Spark** from https://spark.apache.org/downloads.html. Choose correct installation file for your computer/OS requirements. Create a folder in C drive, named as “spark”. Extract the downloaded installation file into “C:\spark”. Add following new user variable and value to environment variables:

SPARK_HOME   C:\spark

4- Download **winutils.exe** from https://github.com/steveloughran/winutils/tree/master/hadoop-2.7.1/bin , move it into a C:\winutils\bin folder that you've created. Add following new user variable and value to environment variables:

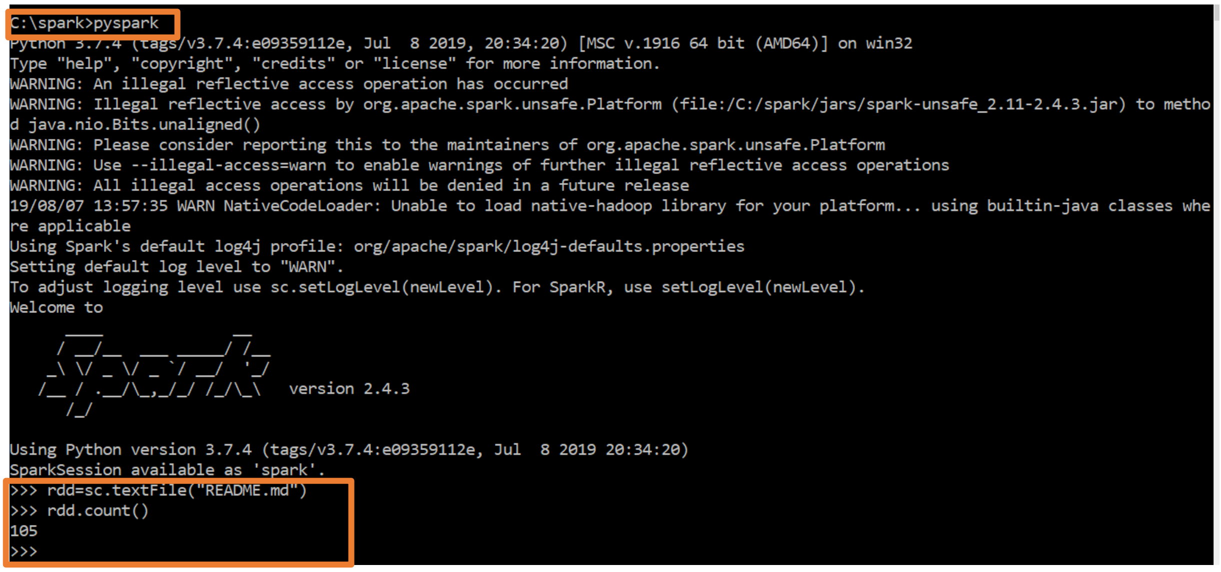
HADOOP_HOME C:\winutils

5- Right-click your Windows menu, select Control Panel, System and Security, and then System. Click on "Advanced System Settings" and then the "Environment Variables" button. Add the following paths to your PATH user variable:

%SPARK_HOME%\bin

C:\Users\PC\AppData\Local\Programs\Python\Python37\Scripts

6- To install Spark, open cmd, change directory to C:\spark\bin, write *pip install pyspark* command. After installation change directory to C:\spark, write **pyspark,** you will see the following screen:



```
C:\spark>pyspark
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/C:/spark/jars/spark-unsafe_2.11-2.4.3.jar) to metho
d java.nio.Bits.unaligned()
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
19/08/07 13:57:35 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes whe
re applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.4.3
      /_/

Using Python version 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019 20:34:20)
SparkSession available as 'spark'.
>>> rdd=sc.textFile("README.md")
>>> rdd.count()
105
>>>
```

We can test pyspark. At this point you should have a >>> prompt.

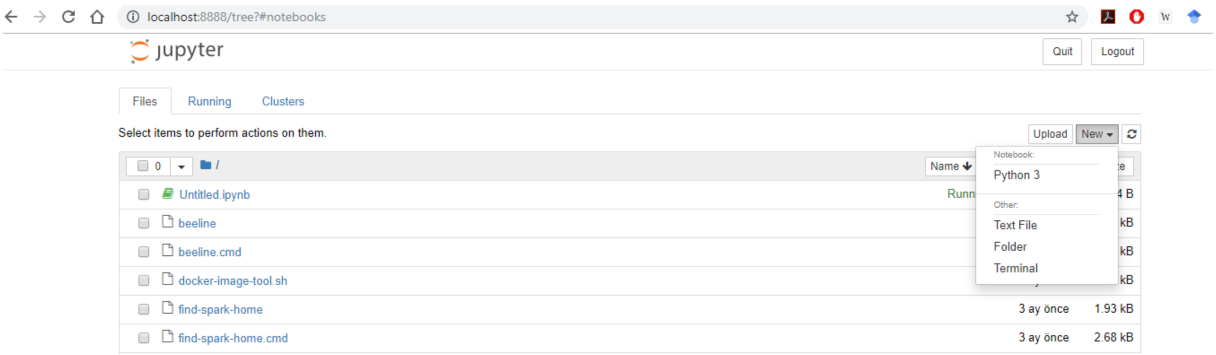
Enter **rdd = sc.textFile("README.md")** (or whatever text file you've found)

Enter **rdd.count()**

You should get a count of the number of lines in that file! Congratulations, you just ran your first Spark program!

Enter **quit()** to exit the spark shell, and close the console window.

7- **To install Jupyter Notebook**, change directory to C:\spark\bin in cmd, write *pip install Jupyter* command. After installation, write **jupyter notebook** on cmd**.** Jupyter Notebook UI will open on web browser.

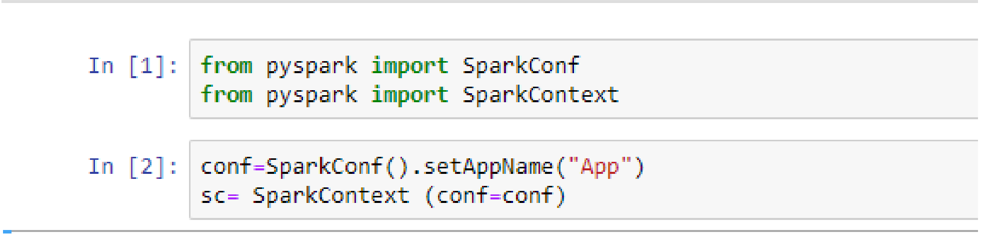


Select items to perform actions on them.

Untitled.ipynb Runn… 4 B
beeline kB
beeline.cmd kB
docker-image-tool.sh kB
find-spark-home 3 ay önce 1.93 kB
find-spark-home.cmd 3 ay önce 2.68 kB

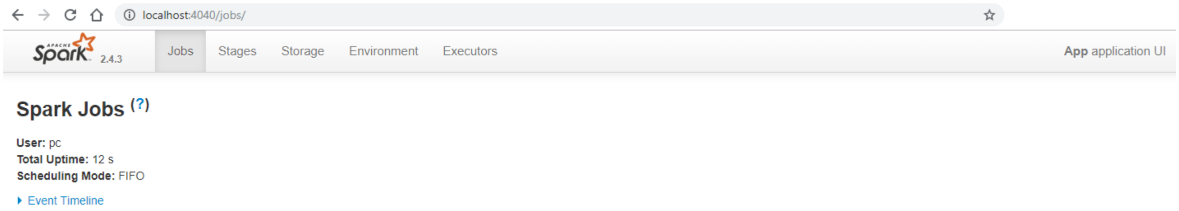
8- **Launching Apache Spark**

Open cmd, start jupyter notebook with “jupyter notebook” command. Create new “python3 notebook”. Write following codes and execute.



```
In [1]: from pyspark import SparkConf
        from pyspark import SparkContext

In [2]: conf=SparkConf().setAppName("App")
        sc= SparkContext (conf=conf)
```

To check whether it works or not, go to localhost’s 4040 port. You will see the spark UI.



Spark Jobs (?)

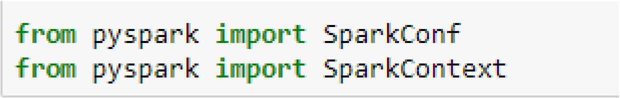
User: pc
Total Uptime: 12 s
Scheduling Mode: FIFO

Event Timeline

Word Count Example

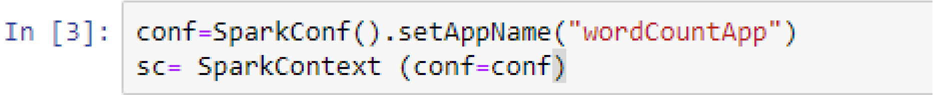
In this section, we will learn how to count the occurrences of unique words in a text, using basic map reduce.

Open cmd, change directory to C:\spark and write” jupyter notebook”. Create new python3 notebook. Import some necessary Spark classes into your program. Add the following lines:



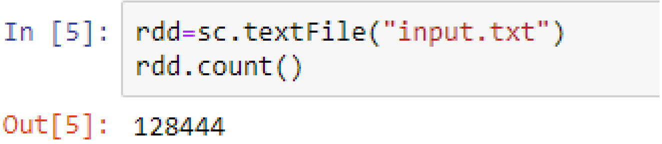
```
from pyspark import SparkConf
from pyspark import SparkContext
```

The first thing a Spark program must do is to create a SparkContext object, which tells Spark how to access a cluster. To create a SparkContext, there is need to build a SparkConf object that contains information about your application.



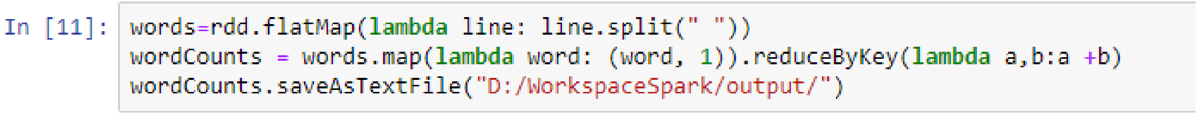
```
In [3]: conf=SparkConf().setAppName("wordCountApp")
        sc= SparkContext (conf=conf)
```

Spark revolves around the concept of a **resilient distributed dataset** (RDD), which is a fault-tolerant collection of elements that can be operated on in parallel. There are two ways to create RDDs: parallelizing an existing collection in your driver program, or referencing a dataset in an external storage system. In this example we will use the latter, external datasets. Following codes read input text file using SparkContext variable and find number of lines in the file.



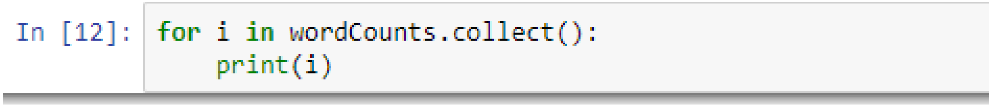
```
In [5]: rdd=sc.textFile("input.txt")
        rdd.count()

Out[5]: 128444
```

Next, in the first line, flatmap of words is created The words are type of PythonRDD. Each line is splitted into words using " " as a separator. In the second line, each word is mapped to key: value pair of *word*, *1*. 1 is the number of occurrences. The result is reduced by key, which is the word, and the values are added. In the third line, result is saved to a text file.



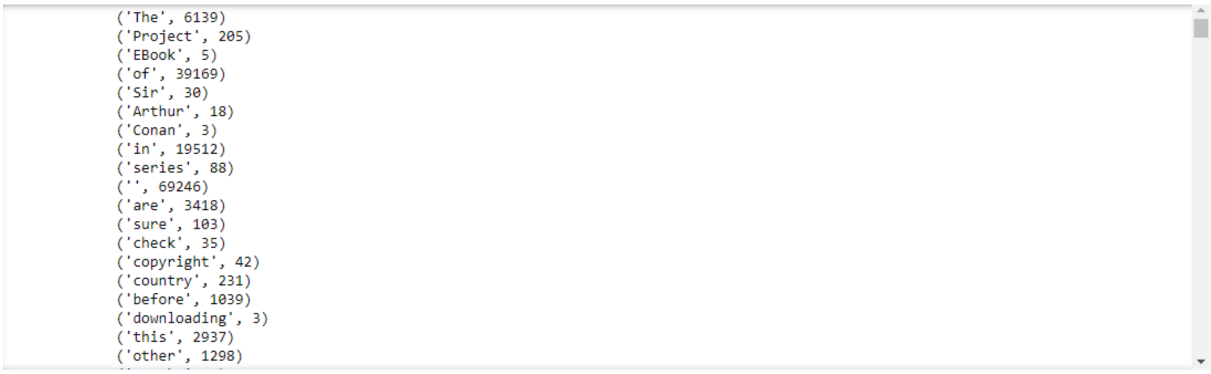
```
In [11]: words=rdd.flatMap(lambda line: line.split(" "))
         wordCounts = words.map(lambda word: (word, 1)).reduceByKey(lambda a,b:a +b)
         wordCounts.saveAsTextFile("D:/WorkspaceSpark/output/")
```

To print number of occurrences as an output use following lines:



```
In [12]: for i in wordCounts.collect():
             print(i)
```

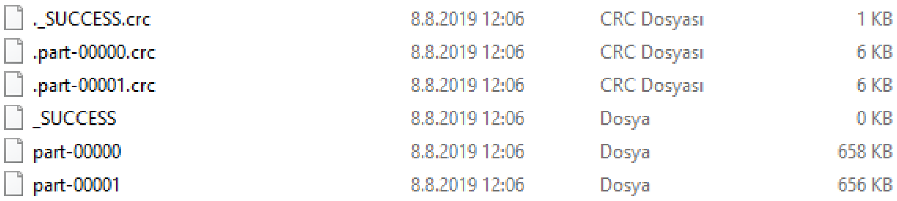
The output:



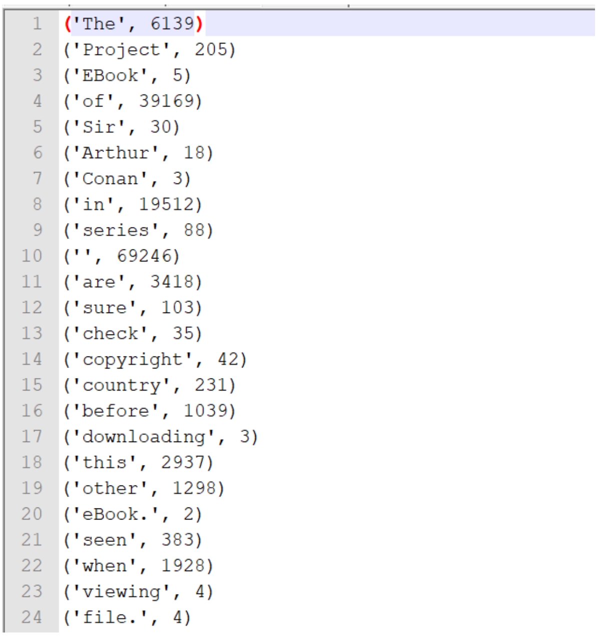
```
('The', 6139)
('Project', 205)
('EBook', 5)
('of', 39169)
('Sir', 30)
('Arthur', 18)
('Conan', 3)
('in', 19512)
('series', 88)
('', 69246)
('are', 3418)
('sure', 103)
('check', 35)
('copyright', 42)
('country', 231)
('before', 1039)
('downloading', 3)
('this', 2937)
('other', 1298)
```

Result

input.txt file is taken as an input and number of occurrences of unique words in the text is shown both as an output on ide and as a file. Content of output files are shown below:



._SUCCESS.crc	8.8.2019 12:06	CRC Dosyası	1 KB
.part-00000.crc	8.8.2019 12:06	CRC Dosyası	6 KB
.part-00001.crc	8.8.2019 12:06	CRC Dosyası	6 KB
_SUCCESS	8.8.2019 12:06	Dosya	0 KB
part-00000	8.8.2019 12:06	Dosya	658 KB
part-00001	8.8.2019 12:06	Dosya	656 KB



```
('The', 6139)
('Project', 205)
('EBook', 5)
('of', 39169)
('Sir', 30)
('Arthur', 18)
('Conan', 3)
('in', 19512)
('series', 88)
('', 69246)
('are', 3418)
('sure', 103)
('check', 35)
('copyright', 42)
('country', 231)
('before', 1039)
('downloading', 3)
('this', 2937)
('other', 1298)
('eBook.', 2)
('seen', 383)
('when', 1928)
('viewing', 4)
('file.', 4)
```

https://spark.apache.org/docs/latest/rdd-programming-guide.html

https://pythonexamples.org/pyspark-word-count-example/

HOMEWORK

124 MB text file and 1 GB text file is given as attachment in the google classroom. By appending 1 GB text file multiple times (you may use "copy" command in cmd, following link may be useful), create 12 GB text file. Now you have 2 text files, size of these are respectively 124MB and 12GB. https://www.wikihow.com/Merge-Text-(.Txt)-Files-in-Command-Prompt

You may use single or 2 different .ipynb file for processing the 2 input file. In both case, create different output folders.

Submit one file in zipped format and named file as **studentNo_NameSurname_Assignment1**. The zipped file should contain:

- your executed code for 124MB and 12GB (**studentNo_studentName**.ipynb)
- your executed code in html format for 124MB and 12GB (**studentNo_studentName**.html)

  In Jupyter notebook; File→ Download As→ .html
- your output directory for 124MB and 12GB (**studentNo_output_fileSize** folder)

1- Using **Word Count Example Code** which is mentioned above, implement following instructions for both file (124 MB and 12 GB):

- Create **SparkConf object** that contains information about your application and set App Name as **studentNo_studentName.**
- After creating **SparkContext**, print your **computer name (host name)** and **ip address** by writing python code.
- Normalize the text:
    - Make all text lowercase
    - Remove punctuations and digits
    - Remove empty lines
- Print number of lines in the file.
- Count (calculate) the occurrences of unique words in a text (key-value).
- Sort the result descending by value. Ex:

```
('the', 19)
('of', 18)
('was', 13)
('it', 12)
('a', 9)
('were', 6)
('and', 4)
('for', 3)
('to', 3)
('age', 2)
```

- Save first 10 key-value pair of sorted result as a text file (You may use saveAsTextFile or any print function for printing results into text file), named it's directory as **studentNo_output_fileSize,** and print these first 10 key-value pair in the Jupyter notebook. Do NOT print all result.
- Calculate and print execution time of program.

Necessary comands

```
!pip install psutil
!python -m pip install --upgrade pip
!pip install findspark
```