(https://cognitiveclass.ai)

This notebook is designed to run in a IBM Watson Studio default runtime (NOT the Watson Studio Apache Spark Runtime as the default runtime with 1 vCPU is free of charge). Therefore, we install Apache Spark in local mode for test purposes only. Please don't use it in production.

In case you are facing issues, please read the following two documents first:

https://github.com/IBM/skillsnetwork/wiki/Environment-Setup (https://github.com/IBM/skillsnetwork/wiki/Environment-Setup)

https://github.com/IBM/skillsnetwork/wiki/FAQ (https://github.com/IBM/skillsnetwork/wiki/FAQ)

Then, please feel free to ask:

https://coursera.org/learn/machine-learning-big-data-apache-spark/discussions/all (https://coursera.org/learn/machine-learning-big-data-apache-spark/discussions/all?cm_mmc=Email_Newsletter-_-Developer_Ed%2BTech-_-WW_WW-_-SkillsNetwork-Courses-IBMDeveloperSkillsNetwork-ML0201EN-SkillsNetwork-20647446&cm_mmca1=000026UJ&cm_mmca2=10006555&cm_mmca3=M12345678&cvosrc=email.Newslette

Please make sure to follow the guidelines before asking a question:

https://github.com/IBM/skillsnetwork/wiki/FAQ#im-feeling-lost-and-confused-please-help-me (https://github.com/IBM/skillsnetwork/wiki/FAQ#im-feeling-lost-and-confused-please-help-me)

If running outside Watson Studio, this should work as well. In case you are running in an Apache Spark context outside Watson Studio, please remove the Apache Spark setup in the first notebook cells.

In [1]:

```
from IPython.display import Markdown, display
def printmd(string):
    display(Markdown('# <span style="color:red">'+string+'</span>'))


if ('sc' in locals() or 'sc' in globals()):
    printmd('<<<<<!!!!! It seems that you are running in a IBM Watson Studio Apache Spa
rk Notebook. Please run it in an IBM Watson Studio Default Runtime (without Apache Spar
k) !!!!!>>>>>')
```

In [2]:

```
!pip install pyspark==2.4.5
```

```
Collecting pyspark==2.4.5
  Downloading pyspark-2.4.5.tar.gz (217.8 MB)
     |████████████████████████████████| 217.8 MB 12 kB/s s eta 0:00:01
Collecting py4j==0.10.7
  Downloading py4j-0.10.7-py2.py3-none-any.whl (197 kB)
     |████████████████████████████████| 197 kB 53.3 MB/s eta 0:00:01
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-2.4.5-py2.py3-none-any.whl s
ize=218257927 sha256=d11c1bbec9376d2fe1a80265d91b3cac04fc1219e2b6a2d7ada92
7156cd5992f
  Stored in directory: /tmp/wsuser/.cache/pip/wheels/01/c0/03/1c241c9c482b
647d4d99412a98a5c7f87472728ad41ae55e1e
Successfully built pyspark
Installing collected packages: py4j, pyspark
Successfully installed py4j-0.10.7 pyspark-2.4.5
```

In [3]:

```
try:
    from pyspark import SparkContext, SparkConf
    from pyspark.sql import SparkSession
except ImportError as e:
    printmd('<<<<<!!!!! Please restart your kernel after installing Apache Spark !!!!!>
>>>>')
```

In [4]:

```
sc = SparkContext.getOrCreate(SparkConf().setMaster("local[*]"))

spark = SparkSession \
    .builder \
    .getOrCreate()
```

Welcome to exercise three of “Apache Spark for Scalable Machine Learning on BigData”. In this exercise you’ll create a DataFrame, register a temporary query table and issue SQL commands against it.

Let’s create a little data frame:

In [5]:

```
from pyspark.sql import Row

df = spark.createDataFrame([Row(id=1, value='value1'),Row(id=2, value='value2')])

# let's have a look what's inside
df.show()

# let's print the schema
df.printSchema()
```

```
+---+------+
| id| value|
+---+------+
|  1|value1|
|  2|value2|
+---+------+

root
 |-- id: long (nullable = true)
 |-- value: string (nullable = true)
```

Now we register this DataFrame as query table and issue an SQL statement against it. Please note that the result of the SQL execution returns a new DataFrame we can work with.

In [6]:

```
# register dataframe as query table
df.createOrReplaceTempView('df_view')

# execute SQL query
df_result = spark.sql('select value from df_view where id=2')

# examine contents of result
df_result.show()

# get result as string
df_result.first().value
```

```
+------+
| value|
+------+
|value2|
+------+
```

Out[6]:

```
'value2'
```

Although we’ll learn more about DataFrames next week, please try to find a way to count the rows in this DataFrame by looking at the API documentation. No worries, we’ll cover DataFrames in more detail next week.

https://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.DataFrame (https://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.DataFrame?cm_mmc=Email_Newsletter-_-Developer_Ed%2BTech-_-WW_WW-_-SkillsNetwork-Courses-IBMDeveloperSkillsNetwork-ML0201EN-SkillsNetwork-20647446&cm_mmca1=000026UJ&cm_mmca2=10006555&cm_mmca3=M12345678&cvosrc=email.Newslette

In [7]:

```
df.count()
```

Out[7]:

```
2
```

Thank you for completing this lab!

This notebook was created by Romeo Kienzler (https://linkedin.com/in/romeo-kienzler-089b4557) I hope you found this lab interesting and educational. Feel free to contact me if you have any questions!

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2020-09-29	2.0	Srishti	Migrated Lab to Markdown and added to course repo in GitLab

In [ ]:

```
```