3_data_inputs_and_outputs

November 22, 2019

1 Reading and Writing Data with Spark

This notebook contains the code from the previous screencast. The only difference is that instead of reading in a dataset from a remote cluster, the data set is read in from a local file. You can see the file by clicking on the "jupyter" icon and opening the folder titled "data".

Run the code cell to see how everything works.

First let's import SparkConf and SparkSession

```
In [2]: import pyspark
        from pyspark import SparkConf
        from pyspark.sql import SparkSession
```

Since we're using Spark locally we already have both a sparkcontext and a sparksession running. We can update some of the parameters, such our application's name. Let's just call it "Our first Python Spark SQL example"

```
In [3]: spark = SparkSession \
            .builder \
            .appName("Our first Python Spark SQL example") \
            .getOrCreate()
```

Let's check if the change went through

```
In [4]: spark.sparkContext.getConf().getAll()
```

```
Out[4]: [('spark.app.name', 'Our first Python Spark SQL example'),
         ('spark.driver.port', '46857'),
         ('spark.driver.host', 'af7408195b7a'),
         ('spark.rdd.compress', 'True'),
         ('spark.serializer.objectStreamReset', '100'),
         ('spark.app.id', 'local-1574441416613'),
         ('spark.master', 'local[*]'),
         ('spark.executor.id', 'driver'),
         ('spark.submit.deployMode', 'client'),
         ('spark.ui.showConsoleProgress', 'true')]
```

```
In [6]: spark
```

```
Out[6]: <pyspark.sql.session.SparkSession at 0x7f7b6ab65ac8>
```

As you can see the app name is exactly how we set it

Let's create our first dataframe from a fairly small sample data set. Througout the course we'll work with a log file data set that describes user interactions with a music streaming service. The records describe events such as logging in to the site, visiting a page, listening to the next song, seeing an ad.

```
In [7]: path = "data/sparkify_log_small.json"
        user_log = spark.read.json(path)

In [8]: user_log.printSchema()

root
 |-- artist: string (nullable = true)
 |-- auth: string (nullable = true)
 |-- firstName: string (nullable = true)
 |-- gender: string (nullable = true)
 |-- itemInSession: long (nullable = true)
 |-- lastName: string (nullable = true)
 |-- length: double (nullable = true)
 |-- level: string (nullable = true)
 |-- location: string (nullable = true)
 |-- method: string (nullable = true)
 |-- page: string (nullable = true)
 |-- registration: long (nullable = true)
 |-- sessionId: long (nullable = true)
 |-- song: string (nullable = true)
 |-- status: long (nullable = true)
 |-- ts: long (nullable = true)
 |-- userAgent: string (nullable = true)
 |-- userId: string (nullable = true)



In [11]: user_log.describe()

Out[11]: DataFrame[summary: string, artist: string, auth: string, firstName: string, gender: str

In [12]: user_log.show(n=1)

+-------------+---------+---------+------+-------------+--------+---------+-----+--------------
|       artist|     auth|firstName|gender|itemInSession|lastName|   length|level|            loc
+-------------+---------+---------+------+-------------+--------+---------+-----+--------------
|Showaddywaddy|Logged In|  Kenneth|     M|          112|Matthews|232.93342| paid|Charlotte-Conco
+-------------+---------+---------+------+-------------+--------+---------+-----+--------------
only showing top 1 row
```

```
In [13]: user_log.take(5)
```

```
Out[13]: [Row(artist='Showaddywaddy', auth='Logged In', firstName='Kenneth', gender='M', itemInS
          Row(artist='Lily Allen', auth='Logged In', firstName='Elizabeth', gender='F', itemInSe
          Row(artist='Cobra Starship Featuring Leighton Meester', auth='Logged In', firstName='V
          Row(artist='Alex Smoke', auth='Logged In', firstName='Sophee', gender='F', itemInSessi
          Row(artist=None, auth='Logged In', firstName='Jordyn', gender='F', itemInSession=0, la
```

```
In [14]: out_path = "data/sparkify_log_small.csv"
```

```
In [15]: user_log.write.save(out_path, format="csv", header=True)
```

```
In [16]: user_log_2 = spark.read.csv(out_path, header=True)
```

```
In [17]: user_log_2.printSchema()
```

```
root
 |-- artist: string (nullable = true)
 |-- auth: string (nullable = true)
 |-- firstName: string (nullable = true)
 |-- gender: string (nullable = true)
 |-- itemInSession: string (nullable = true)
 |-- lastName: string (nullable = true)
 |-- length: string (nullable = true)
 |-- level: string (nullable = true)
 |-- location: string (nullable = true)
 |-- method: string (nullable = true)
 |-- page: string (nullable = true)
 |-- registration: string (nullable = true)
 |-- sessionId: string (nullable = true)
 |-- song: string (nullable = true)
 |-- status: string (nullable = true)
 |-- ts: string (nullable = true)
 |-- userAgent: string (nullable = true)
 |-- userId: string (nullable = true)
```

```
In [18]: user_log_2.take(2)
```

```
Out[18]: [Row(artist='Showaddywaddy', auth='Logged In', firstName='Kenneth', gender='M', itemInS
          Row(artist='Lily Allen', auth='Logged In', firstName='Elizabeth', gender='F', itemInSe
```

```
In [20]: user_log_2.select("userID").show()
```

```
+------+
|userID|
+------+
|  1046|
```

```
|  1000|
|  2219|
|  2373|
|  1747|
|  1747|
|  1162|
|  1061|
|   748|
|   597|
|  1806|
|   748|
|  1176|
|  2164|
|  2146|
|  2219|
|  1176|
|  2904|
|   597|
|   226|
+------+
only showing top 20 rows
```

```
In [21]: user_log_2.take(1)
```

```
Out[21]: [Row(artist='Showaddywaddy', auth='Logged In', firstName='Kenneth', gender='M', itemInS
```

```
In [ ]:
```