

Spark DataFrames: Takeaways

by Dataquest Labs, Inc. - All rights reserved © 2020

Syntax

- Instantiating the SQLContext class:

```
from pyspark.sql import SQLContext
sqlCtx = SQLContext(sc)
```

- Reading in JSON data:

```
df = sqlCtx.read.json("census_2010.json")
```

- Using the show() method to print the first five rows:

```
df.show(5)
```

- Using the head method and a for loop to return the first five rows of the DataFrame:

```
first_five = df.head(5)
for r in first_five:
    print(r.age)
```

- Using the show method to display columns:

```
df.select('age', 'males', 'females')
```

- Converting a Spark DataFrame to a pandas DataFrame:

```
pandas_df = df.toPandas()
```

Concepts

- The Spark DataFrame:
 - Allows you to create and work with dataframe objects.
 - Combines the scale and speed of Spark with the familiar query, filter, and analysis functions of pandas.
 - Allows you to modify and reuse existing pandas code to much larger data sets.
 - Has better support for various data formats.
 - Is immutable.
- The Spark SQL class gives Spark more information about that data structure you're using and the computation you want to perform.
- When you read data into the SQLContext object, Spark does the following:
 - Instantiates a Spark DataFrame object.
 - Infers the schema from the data and associates it with the DataFrame.
 - Reads in the data and distributes it across clusters (if multiple clusters are available).
 - Returns the DataFrame object.
- The Spark dataframe has some of the same method implementations of the pandas DataFrame, such as `agg()`, `join()`, `sort()`, and `where()`.
- To handle the weaknesses of the Spark library, we can convert a Spark DataFrame to a pandas DataFrame.

Resources

- [Spark programming guide](#)
- [Pandas and Spark DataFrames](#)



Takeaways by Dataquest Labs, Inc. - All rights reserved © 2020