How to Add a Count Column to a PySpark DataFrame

Authored by

stats writer

February 3, 2026

Adding a count column to a PySpark DataFrame can be done by using the built-in function "count()" and the "withColumn()" method. First, the DataFrame needs to be grouped by the desired column using the "groupBy()" function. Then, the "count()" function can be applied to the grouped DataFrame to calculate the number of rows in each group. Finally, the "withColumn()" method can be used to create a new column, giving it a name and assigning the count values to it. This will result in a new DataFrame with an additional count column, showing the number of rows in each group.

Add a Count Column to PySpark DataFrame

You can use the following syntax to add a 'count' column to a PySpark DataFrame:

```
import pyspark.sql.functions as F
from pyspark.sql import Window

#define column to count for
w = Window.partitionBy('team')

#add count column to DataFrame
df_new = df.select('team', 'points',
F.count('team').over(w).alias('n'))
```

This particular example adds a new column named n that shows the count of values in the team column.

The following example shows how to use this syntax in practice.

Example: Add Count Column to PySpark DataFrame

Suppose we have the following PySpark DataFrame that contains information about points scored by various basketball players:

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()

#define data
data = ,
,
,
,
,
,
,
,
]

#define column names
columns =

#create dataframe using data and column names
df = spark.createDataFrame(data, columns)
```

```
#view dataframe
df.show()

+----+------+
|team|points|
+----+------+
| A| 11|
| A| 8|
| A| 10|
| B| 6|
| B| 6|
| C| 5|
| C| 15|
| C| 31|
| D| 24|
+----+------+
```

Suppose we would like to add a new column that displays the count of each unique value in the team column.

We can use the following syntax to do so:

```
import pyspark.sql.functions as F
```

```
from pyspark.sql import Window

#define column to count for
w = Window.partitionBy('team')

#add count column to DataFrame
df_new = df.select('team', 'points',
F.count('team').over(w).alias('n'))

#view new DataFrame
df_new.show()

+----+------+---+
|team|points| n|
+----+------+---+
| A| 11| 3|
| A| 8| 3|
| A| 10| 3|
| B| 6| 2|
| B| 6| 2|
| C| 5| 3|
| C| 15| 3|
| C| 31| 3|
| D| 24| 1|
+----+------+---+
```

Here is how to interpret the output:

The value 'A' occurs 3 times in the team column. Thus, a value of 3 is assigned to each row in the n column where the team is A.The value 'B' occurs 2 times in the team column. Thus, a value of 2 is assigned to each row in the n column where the team is B.

And so on.

The following tutorials explain how to perform other common tasks in PySpark: