How can I perform a count distinct operation in PySpark on a DataFrame?

Authored by

stats writer

June 24, 2024

The count distinct operation in PySpark on a DataFrame is a function used to determine the number of unique values in a specific column or set of columns. It can be performed by using the "distinct" function followed by the "count" function on the desired DataFrame. This will return the count of distinct values in the specified column(s) and can be useful for data analysis and cleaning.

In PySpark, you can use `distinct().count()` of DataFrame or `countDistinct()` SQL function to get the count distinct.

`distinct()` eliminates duplicate records(matching all columns of a Row) from DataFrame, count() returns the count of records on DataFrame. By chaining these you can get the count distinct of PySpark DataFrame.

`countDistinct()` is a SQL function that could be used to get the count distinct of the selected multiple columns.

Let's see these two ways with examples.

Before we start, first let's create a DataFrame with some duplicate rows and duplicate values in a column.

```
# Create SparkSession and Prepare Data
from pyspark.sql import SparkSession
spark = SparkSession.builder
.appName('SparkByExamples.com')
.getOrCreate()
```

data =
columns =
df = spark.createDataFrame(data=data,schema=columns)
df.show()

Yields below output

```
# Output
+-------+---------+------+
| Name| Dept|Salary|
+-------+---------+------+
| James| Sales| 3000|
|Michael| Sales| 4600|
| Robert| Sales| 4100|
| Maria| Finance| 3000|
```

```
| James| Sales| 3000|
| Scott| Finance| 3300|
| Jen| Finance| 3900|
| Jeff|Marketing| 3000|
| Kumar|Marketing| 2000|
| Saif| Sales| 4100|
+-------+---------+------+
```

1. Using DataFrame distinct() and count()

On the above DataFrame, we have a total of 10 rows and one row with all values duplicated, performing distinct counts (`distinct().count()`) on this DataFrame should get us 9.

```
# Applying distinct() and count()
df1 = df.distinct()
print(df1.count())
df1.show()
```

This yields output **"Distinct Count: 9"**

```
# Output
9
+-------+---------+------+
| Name| Dept|Salary|
+-------+---------+------+
|Michael| Sales| 4600|
| James| Sales| 3000|
| Robert| Sales| 4100|
| Scott| Finance| 3300|
| Maria| Finance| 3000|
| Jen| Finance| 3900|
| Kumar|Marketing| 2000|
| Jeff|Marketing| 3000|
| Saif| Sales| 4100|
+-------+---------+------+
```

Below is an example to get Distinct values from a single column and then apply count() to get the count of distinct values.

```
# Apply distinct() and count() on a single column
df2 = df.select("Name").distinct()
print(df2.count())
df2.show()


# Output
9
+-------+
| Name|
+-------+
| James|
|Michael|
| Robert|
| Scott|
| Maria|
| Jen|
| Kumar|
| Saif|
| Jeff|
+-------+
```

If you want to count distinct values based on multiple columns, you can pass multiple column names to the `select` method. Example.

```
# Applying distinct(), count() on multiple columns
df3 = df.select("Name", "Dept").distinct().count()
print(df3)


# Output
9
```

2. Using countDistinct() SQL Function

DataFrame `distinct()` returns a new DataFrame after eliminating duplicate rows (distinct on all columns). if you want to get count distinct on selected multiple columns, use the PySpark SQL function `countDistinct()`. This function returns the number of distinct elements in a group.

In order to use this function, you need to import it first.

```
from pyspark.sql.functions import countDistinct
df2=df.select(countDistinct("Dept", "Salary"))
df2.show()
```

Yields below output

```
# Output
+---------------------------------+
|count(DISTINCT Dept, Salary)|
+---------------------------------+
|8 |
+---------------------------------+
```

Note that `countDistinct()` function returns a value in a `Column` type hence, you need to collect it to get the value from the DataFrame. This function can be used to get the distinct count of any number of selected or all columns.

```
# Applying collect() after countDistinct()
print("Distinct Count of Department & Salary: "+ str(df2.collect()))
```

This outputs **"Distinct Count of Department & Salary: 8"**

3. Using SQL to get Count Distinct

```
df.createOrReplaceTempView("EMP")
spark.sql("select distinct(count(*)) from EMP").show()
```

Displays this on console
+--------+
|count(1)|
+--------+
| 10|
+--------+

4. Source Code of PySpark Count Distinct Example

```
from pyspark.sql import SparkSession
```

```
spark = SparkSession.builder
.appName('SparkByExamples.com')
.getOrCreate()
```

```
data =
columns =
df = spark.createDataFrame(data=data,schema=columns)
df.distinct().show()
print("Distinct Count: " + str(df.distinct().count()))

# Using countDistrinct()
from pyspark.sql.functions import countDistinct
df2=df.select(countDistinct("Dept","Salary"))
df2.show()

print("Distinct Count of Department & Salary: "+ str(df2.collect()))
```

5. Conclusion

In this article, you have learned how to get a count distinct from all columns or selected multiple columns on PySpark DataFrame.

Happy Learning !!

Related Articles