Master PySpark: From Zero to Big Data Hero!!

Aggregate function in Dataframe – Part 1

Let's create a sample DataFrame using PySpark that includes various numerical values. This dataset will be useful for demonstrating the aggregate functions.

```
# Create sample data
data = [
    Row(id=1, value=10),
    Row(id=2, value=20),
    Row(id=3, value=30),
    Row(id=4, value=None),
    Row(id=5, value=40),
    Row(id=6, value=20)
]
# Create DataFrame
df = spark.createDataFrame(data)
# Show the DataFrame
df.show()
```

Sample Output

```
+---+-----+
| id|value|
+---+-----+
|  1|   10|
|  2|   20|
|  3|   30|
|  4| null|
|  5|   40|
|  6|   20|
+---+-----+
```

Aggregate Functions in PySpark

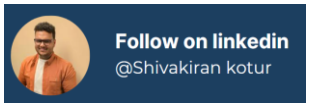
1. **Summation (sum)**: Sums up the values in a specified column.

```
from pyspark.sql import functions as F

# Summation
total_sum = df.select(F.sum("value")).show()
```

▸ (2) Spark Jobs

```
+----------+
|sum(value)|
+----------+
|       120|
+----------+
```



Follow me on LinkedIn – Shivakiran kotur

2. average of the values in a specified column.

```
# Average
average_value = df.select(F.avg("value")).show()
```

▸ (2) Spark Jobs

```
+----------+
|avg(value)|
+----------+
|      24.0|
+----------+
```

3. **Count (count)**: Counts the number of non-null values in a specified column.

```
# Count
non_null_count = df.select(F.count("value")).show()
```

▸ (2) Spark Jobs

```
+------------+
|count(value)|
+------------+
|           5|
+------------+
```

4. **Maximum (max) and Minimum (min)**: Finds the maximum and minimum values in a specified column.

```
# Maximum and Minimum
max_min_values = df.select(F.max("value"), F.min("value")).show()
```

▸ (2) Spark Jobs

```
+----------+----------+
|max(value)|min(value)|
+----------+----------+
|        40|        10|
+----------+----------+
```

5. **Distinct Values Count (countDistinct)**: Counts the number of distinct values in a specified column.

```
# Distinct Values Count
distinct_count = df.select(F.countDistinct("value")).show()
```

▸ (3) Spark Jobs

```
+---------------------+
|count(DISTINCT value)|
+---------------------+
|                    4|
+---------------------+
```

Notes

- **Handling Nulls**: The count function will count only non-null values, while sum, avg, max, and min will ignore null values in their calculations.
- **Performance**: Aggregate functions can be resource-intensive, especially on large datasets. Using the appropriate partitioning can improve performance.
- **Use Cases**:
  - **Summation**: Useful for calculating total sales, total revenue, etc.
  - **Average**: Helpful for finding average metrics like average sales per day.
  - **Count**: Useful for counting occurrences, such as the number of transactions.
  - **Max/Min**: Helps to determine the highest and lowest values, such as maximum sales on a specific day.
  - **Distinct Count**: Useful for finding unique items, like unique customers or products.

This should give you a solid understanding of aggregate functions in PySpark! If you have any specific questions or need further assistance, feel free to ask!