How to get a value from the Row object in Spark Dataframe?

for

```
averageCount = (wordCountsDF
                .groupBy().mean()).head()
```

I get

> Row(avg(count)=1.6666666666666667)

but when I try:

```
averageCount = (wordCountsDF
                .groupBy().mean()).head().getFloat(0)
```

I get the following error:

AttributeError: getFloat --- AttributeError Traceback (most recent call last) in () 1 # TODO: Replace with appropriate code ----> 2 averageCount = (wordCountsDF 3 .groupBy().mean()).head().getFloat(0) 4 5 print averageCount

/databricks/spark/python/pyspark/sql/types.py in **getattr**(self, item) 1270 raise AttributeError(item) 1271 except ValueError: -> 1272 raise AttributeError(item) 1273 1274 def **setattr**(self, key, value):

AttributeError: getFloat

What am I doing wrong?

apache-spark   pyspark   spark-dataframe

asked Jun 23 at 18:55

saptak
111   4

2 Answers

I figured it out. This will return me the value:

```
averageCount = (wordCountsDF
              .groupBy().mean()).head()[0]
```

answered Jun 23 at 19:02

saptak
111   4

Dataframe rows are inherited from namedtuples (from the collections library), so while you can index them like a traditional tuple the way you did above, you probably want to access it by the name of its fields. That is, after all, the point of named tuples, and it is also more robust to future changes. Like this:

```
averageCount = wordCountsDF.groupBy().mean().head()['avg(jobs)']
```

answered Jun 23 at 19:48

Jeff L.
1,018   1   15