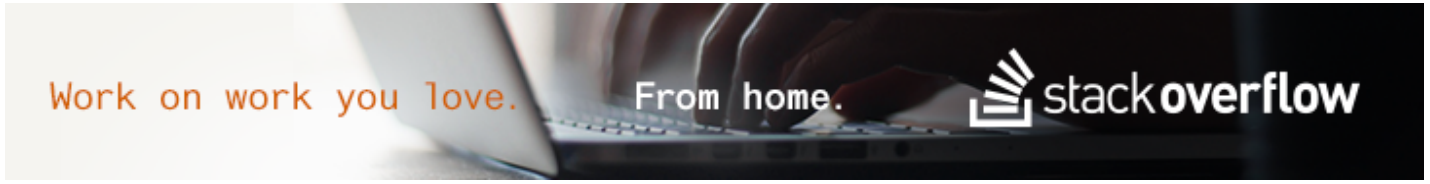
Spark Convert Data Frame Column to Vectors.dense



I am very new to Spark and I am trying to apply StandardScaler() to a column in a DataFrame.

```
+---------------+
|      DF_column|
+---------------+
| 0.114285714286|
| 0.115702479339|
| 0.267893660532|
|0.0730337078652|
| 0.124309392265|
| 0.365714285714|
| 0.111747851003|
| 0.279538904899|
| 0.134670487106|
| 0.523287671233|
| 0.404011461318|
|          0.375|
```

```
| 0.125517241379|
|0.0143266475645|
| 0.313684210526|
| 0.381088825215|
| 0.411428571429|
| 0.327683615819|
| 0.153409090909|
| 0.344827586207|
+---------------+
```

The problem is that applying it like this, gives me an error, "requirement failed: Input column DF_column must be a vector column". I tried using UDF but still doesn't work.

```
scaler = StandardScaler(inputCol='DF_column',
    outputCol="scaledFeatures",withStd=True, withMean=False)
```

I did the example of the LIBSVM but that is easy a the TXT file is loading features as Vectors.

python apache-spark pyspark spark-dataframe apache-spark-mllib

edited Mar 9 at 7:16

zero323
66.4k 16 78 138

asked Mar 9 at 3:40

harvybcn
28 5

1 Answer

If you have a column of scalars then `StandardScaler` is a serious overkill. You can scale directly:

```
from pyspark.sql.functions import col, stddev_samp

df.withColumn("scaled",
  col("DF_column") / df.agg(stddev_samp("DF_column")).first()[0])
```

but if you really want to use scaler than assemble a vector first:

```
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.feature import StandardScaler

assembler = VectorAssembler(
```

```
  inputCols=["DF_column"], outputCol="features"
)

assembled = assembler.transform(df)

scaler = StandardScaler(
  inputCol="features", outputCol="scaledFeatures",
  withStd=True, withMean=False
).fit(assembled)

scaler.transform(assembled)
```

answered Mar 9 at 6:55

zero323
66.4k 16 78 138