How do I use multiple conditions with pyspark.sql.funtions.when()?

I have a dataframe with a few columns. Now I want to derive a new column from 2 other columns:

```
from pyspark.sql import functions as F
new_df = df.withColumn("new_col", F.when(df["col-1"] > 0.0 & df["col-2"] > 0.0,
1).otherwise(0))
```

With this I only get an exception:

```
py4j.Py4JException: Method and([class java.lang.Double]) does not exist
```

It works with just one condition like this:

```
new_df = df.withColumn("new_col", F.when(df["col-1"] > 0.0, 1).otherwise(0))
```

Does anyone know to use multiple conditions?

I'm using Spark 1.4.

python apache-spark

asked Oct 15 '15 at 14:56

jho
58 6

in Python, shouldn't you write `df["col-1"] > 0.0 and df["col-2"]>0.0` ? – Ashalynd Oct 15 '15 at 15:01

1 Actually no. This would lead to the following error `ValueError: Cannot convert column into bool: please use '&' for 'and', '|' for 'or', '~' for 'not' when building DataFrame boolean expressions.` – jho Oct 15 '15 at 15:02

1 ah I see, then you have to use brackets I guess: (df["col-1"] > 0.0) & (df["col-2"] > 0.0), to fix the priority – Ashalynd Oct 15 '15 at 15:03

That's weird. I'm pretty sure I tested this, but now it works. Thanks! :) – jho Oct 15 '15 at 15:06

1 @Ashalynd Please post it as an answer. – zero323 Oct 15 '15 at 15:19

1 Answer

Use brackets to enforce the desired operator precedence:

```
F.when( (df["col-1"]>0.0) & (df["col-2"]>0.0), 1).otherwise(0)
```

edited Jul 19 at 16:35

Pyrce
3,544 15 31

answered Oct 15 '15 at 19:37

Ashalynd
8,524 2 15 21