PYSPARK中的groupby, agg，alias, orderby多个columns的操作

原创 anxingirl 于 2022-03-10 14:49:58 发布 2639 收藏 版权
分类专栏： PYSPARK 文章标签： python spark

```python
#Pyspark imports

import pyspark
from pyspark.sql import SQLContext
from pyspark.sql.functions import hour, when, col, date_format, to_timestamp
from pyspark.sql.functions import *


# Define Spark Context

sc = pyspark.SparkContext(appName="Homework")
sqlContext = SQLContext(sc)


# Function to load data

def load_data():
    df = sqlContext.read.option("header",True).csv("yellow_tripdata_2019-01_short.csv")
    return df

df = load_data()
```

https://spark.apache.org/docs/3.2.1/api/python/reference/api/pyspark.sql.DataFrame.orderBy.html?highlight=orderby#pyspark.sql.DataFrame.orderBy

在pyspark中，可以和pandas一样进行groupby操作，count 也是一样可以做的，例如我们可以使用下面的简单操作来去得到对column1进行group后，计算每个group的计数，并且展示出来。

```
df.groupy("column1").count().show()
```

现在我们开始在这个语句上面增加条件，加上各种变化，满足现实中各种奇怪的需求：

1.根据多个columns来进行group?

没问题！ 直接groupby 多个列就可以了！

```
df.groupy(["column1","column2"]).count().show()
```

2.我需要得到的一个计数之外，还有其他的agg操作，比如avg?

办得到！使用agg:

#1. 方法1 使用字典

#2. 方法2 不适用字典，可以加上alias，就是给咱们新生成的column增加别名，推荐这个方案，不然你不好确认你增加了个什么玩意，后续怎么调用

```
#1. 方法1 使用字典
df.groupby(['Column1','Column2']).agg(count("*").alias("count"), avg("Column3").alias("Column4")).show()

#2. 方法2 不适用字典，可以加上alias，就是给咱们新生成的column增加别名，推荐这个方案，不然你不好确认你增加了个什么玩意，后续怎么调用

df.groupby(['Column1','Column2']).agg(count("*").alias("count"), avg("Column3").alias("Column4")).show()
```

3.我需要得排序啊，而且要对多个字段排序，一会要倒排一会要正排！

阔以！使用orderby:

```
df.groupby(['column1','columns2']).agg(count("*").alias("count"), avg("column3").alias("column4")).orderBy(['count','column4'],ascending=[0,1])
```

记得将每次处理后的数据，都保存哦！！么么哒！