新的列到Spark DataFrame(使用PySpark)？

018-02-10 20:58 □dataframe, pyspark, Python, spark □去评论

)，并想添加一个新的列。

我已经尝试了以下方法，但没有任何成功的：

```
type(randomed_hours) # => list

# Create in Python and transform to RDD

new_col = pd.DataFrame(randomed_hours, columns=['new_col'])

spark_new_col = sqlContext.createDataFrame(new_col)

my_df_spark.withColumn("hours", spark_new_col["new_col"])
```

还有一个错误使用这个：

```
my_df_spark.withColumn("hours",  sc.parallelize(randomed_hours))
```

那么如何使用PySpark将新的列(基于Python向量)添加到现有的DataFrame？

最佳解决方法

您不能将任意列添加到Spark中的`DataFrame`。新列只能使用literal创建(其他literal类型在How to add a constant column in a Spark DataFrame?中描述)

```
from pyspark.sql.functions import lit

df = sqlContext.createDataFrame(
```

```
    [(1, "a", 23.0), (3, "B", -23.0)], ("x1", "x2", "x3"))

df_with_x4 = df.withColumn("x4", lit(0))
df_with_x4.show()

## +---+---+-----+---+
## | x1| x2|   x3| x4|
## +---+---+-----+---+
## |  1|  a| 23.0|  0|
## |  3|  B|-23.0|  0|
## +---+---+-----+---+
```

转换现有的列：

```
from pyspark.sql.functions import exp

df_with_x5 = df_with_x4.withColumn("x5", exp("x3"))
df_with_x5.show()

## +---+---+-----+---+--------------------+
## | x1| x2|   x3| x4|                  x5|
## +---+---+-----+---+--------------------+
## |  1|  a| 23.0|  0| 9.744803446248903E9|
## |  3|  B|-23.0|  0|1.026187963170189...|
## +---+---+-----+---+--------------------+
```

包括使用join：

```
from pyspark.sql.functions import exp

lookup = sqlContext.createDataFrame([(1, "foo"), (2, "bar")], ("k", "v"))
df_with_x6 = (df_with_x5
    .join(lookup, col("x1") == col("k"), "leftouter")
    .drop("k")
    .withColumnRenamed("v", "x6"))

## +---+---+-----+---+--------------------+----+
## | x1| x2|   x3| x4|                  x5|  x6|
## +---+---+-----+---+--------------------+----+
## |  1|  a| 23.0|  0| 9.744803446248903E9| foo|
## |  3|  B|-23.0|  0|1.026187963170189...|null|
## +---+---+-----+---+--------------------+----+
```

或使用function /udf生成：

```
from pyspark.sql.functions import rand

df_with_x7 = df_with_x6.withColumn("x7", rand())
df_with_x7.show()

## +---+---+-----+---+-------------------+----+------------------+
## | x1| x2|   x3| x4|                 x5|  x6|                x7|
## +---+---+-----+---+-------------------+----+------------------+
## |  1|  a| 23.0|  0| 9.744803446248903E9| foo|0.41930610446846617|
## |  3|  B|-23.0|  0|1.026187963170189...|null|0.37801881545497873|
## +---+---+-----+---+-------------------+----+------------------+
```

映射到Catalyst表达式的性能优先、内置函数(`pyspark.sql.functions`)通常优于Python用户定义的函数。

如果你想添加一个任意RDD的内容作为一个列，你可以

- 添加row numbers to existing data frame
- 在RDD上调用`zipWithIndex`并将其转换为数据帧
- 加入这两个使用索引作为连接键

次佳解决方法

使用UDF添加列：

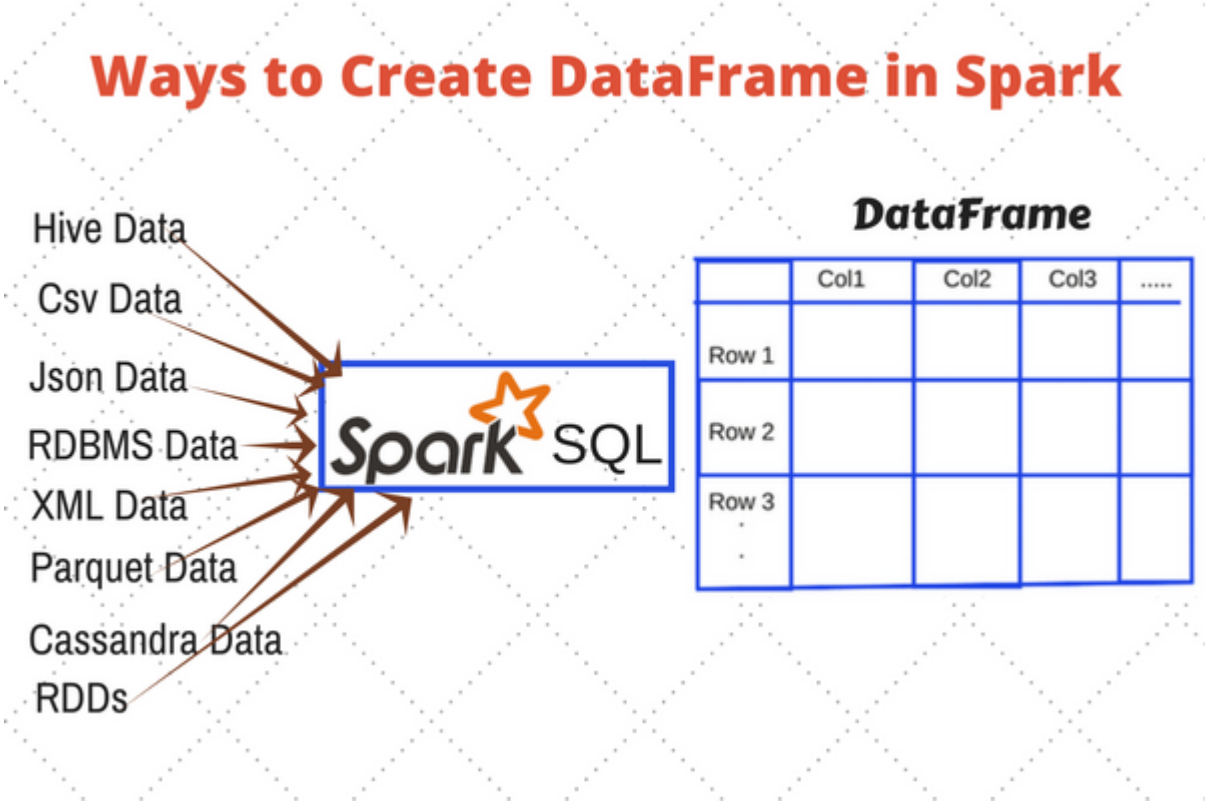
```
df = sqlContext.createDataFrame(
    [(1, "a", 23.0), (3, "B", -23.0)], ("x1", "x2", "x3"))

from pyspark.sql.functions import udf
from pyspark.sql.types import *

def valueToCategory(value):
   if   value == 1: return 'cat1'
   elif value == 2: return 'cat2'
   ...
   else: return 'n/a'

# NOTE: it seems that calls to udf() must be after SparkContext() is called
udfValueToCategory = udf(valueToCategory, StringType())
df_with_cat = df.withColumn("category", udfValueToCategory("x1"))
df_with_cat.show()

## +---+---+-----+---------+
```

```
## | x1| x2|   x3| category|
## +---+---+-----+---------+
## |  1|  a| 23.0|     cat1|
## |  3|  B|-23.0|      n/a|
## +---+---+-----+---------+
```

第三种解决方法

对于Spark 2.0

```
# assumes schema has 'age' column
df.select('*', (df.age + 10).alias('agePlusTen'))
```



参考资料

- How do I add a new column to a Spark DataFrame (using PySpark)?