UDF with DataFrame

1. Change firstname, lastname columns into

Mr.first\_two\_letters\_of\_firstname<space>lastname

for example - michael, phelps becomes Mr.mi phelps

Data is already loaded in sportData Data frame in previous assignment.

* Create UDF to concate name first name and last name according to the given condition.

def udfFullName=

org.apache.spark.sql.functions.udf((firstName:String,lastName:String) =>

{val initial=firstName.substring(0,2);

"Mr."+initial+" "+lastName

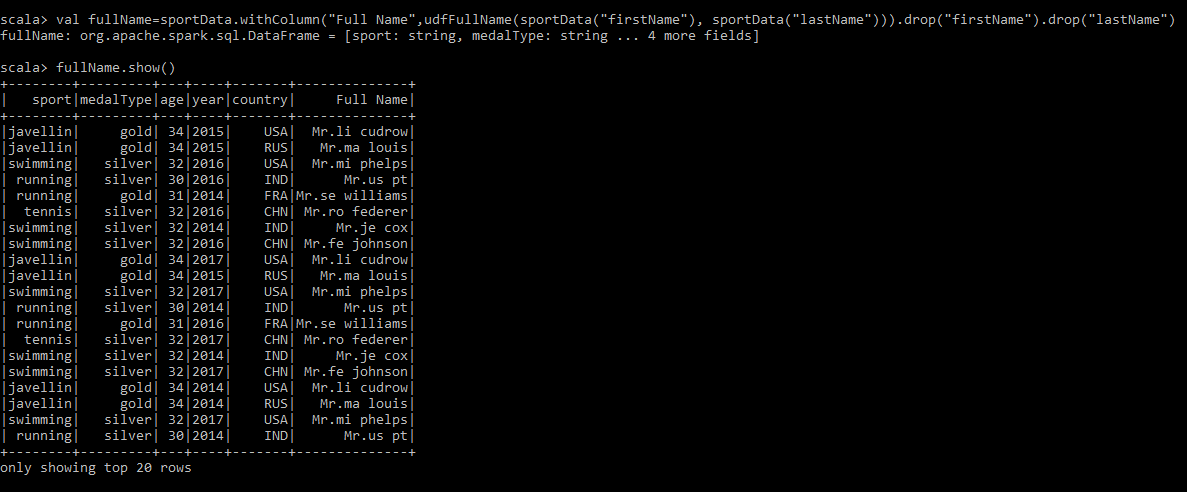
})

* Drop first name lastname and add new column with new UDF

val fullName=sportData.withColumn("Full Name",udfFullName(sportData("firstName"), sportData("lastName"))).drop("firstName").drop("lastName")

* Show data of final UDF

fullName.show()



1. Add a new column called ranking using udfs on dataframe, where :

gold medalist, with age >= 32 are ranked as pro

gold medalists, with age <= 31 are ranked amateur

silver medalist, with age >= 32 are ranked as expert

silver medalists, with age <= 31 are ranked rookie

* Write UDF with conditions

def udfRanking=org.apache.spark.sql.functions.udf((age:Int,medalType:String) => {

if(medalType.equals("gold") && age>=32) "pro"

else if(medalType.equals("gold") && age<=31) "amateur"

else if(medalType.equals("silver") && age>=32) "expert"

else if(medalType.equals("silver") && age<=31) "rookie" else "NA"})

* Call UDF by passing age and medal type.

val rankData=sportData.withColumn("ranking",udfRanking(sportData("age"),sportData("medalType")))

* Display result

rankData.show()

