Using udfs on dataframe

1. Change firstname, lastname columns into

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

for example - michael, phelps becomes Mr.mi phelps

Create a case class Sports

**case class Sports(firstname:String,lastname:String,sports:String,medal\_type:String,age:Int,year:Int,country:String)**

Create RDD from the file

**val sportsRDD = sc.textFile("/home/acadgild/inputdir/Sports\_data.txt")**

**val firstline = sportsRDD.first**

Filter out the first line from RDD as it contains headers

**val dataRDD = sportsRDD.filter(line=>(line!=firstline)).map(line=>line.split(","))**

map the RDD with case class Sports

**val datamapRDD = dataRDD.map(s=>Sports(s(0),s(1),s(2),s(3),s(4).toInt,s(5).toInt,s(6)))**

Convert to dataframe

**val sportsDF = datamapRDD.toDF**

Register Dataframe as Temp Table

**sportsDF.registerTempTable("sports")**

Create UDF for Concatenation**:**

**def concatFirstLast(firstname:String,lastname:String):String=**

**{**

**"Mr. "+firstname.substring(0,2)+" "+lastname**

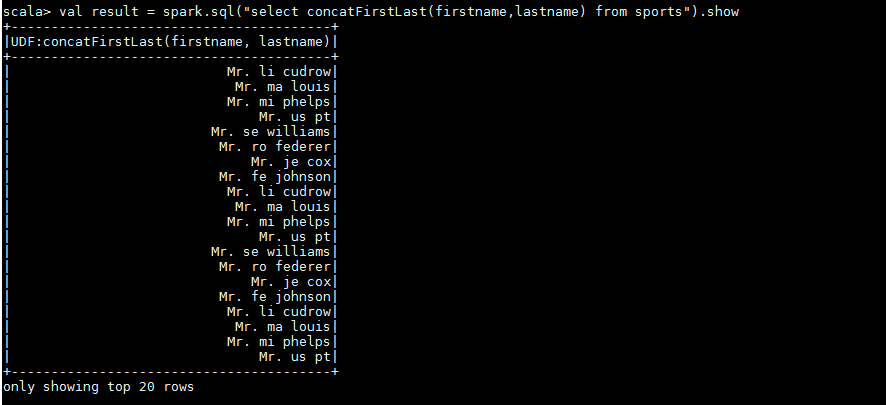
**}**

Register the UDF:

**spark.sqlContext.udf.register("concatFirstLast",concatFirstLast \_)**

Run the query:

**val result = spark.sql("select concatFirstLast(firstname,lastname) from sports").show**



**2. 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**

Create UDF for Ranking:

**def ranking(medaltype:String,age:Int):String=**

**{**

**if(medaltype=="gold" && age>=32)**

**{**

**"Pro"**

**}**

**else if(medaltype=="gold" && age<=31)**

**{**

**"amateur"**

**}**

**else if(medaltype=="silver" && age>=32)**

**{**

**"expert"**

**}**

**else if(medaltype=="silver" && age<=31)**

**{**

**"rookie"**

**}**

**else**

**{**

**"default"**

**}**

**}**

Register the UDF:

**spark.sqlContext.udf.register("ranking", ranking \_)**

Run the query:

**val result = spark.sql("select sports.\*,ranking(medal\_type,age) rank from sports").show**

