Associated Data Files

Link

https://drive.google.com/drive/folders/0B\_P3pWagdIrrTEVTVE1VWXI0YVk

5. Problem Statement

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

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

*val dataset = sc.textFile("file:///home/cloudera/sports\_data.txt")*

*val header = dataset.first()*

*val rowsRDD = dataset.filter(line => line != header)*

*val sportsRDD = rowsRDD.map{row => row.split(",")}.map{cols => sports(cols(0), cols(1), cols(2), cols(3), cols(4), cols(5), cols(6) )}*

*val sportsDF = sportsRDD.toDF()*

*sportsDF.registerTempTable("sports")*

*def subs = (x:String, y:String) => { val z = "Mr."+x.substring(0,2)+" "+y; z}*

*sqlContext.udf.register("parsed",subs)*

*val data = sqlContext.sql( "SELECT parsed(firstname,lastname) FROM sports")*

*data.collect.foreach(println)*



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

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

*val dataset = sc.textFile("file:///home/cloudera/sports\_data.txt")*

*val header = dataset.first()*

*val rowsRDD = dataset.filter(line => line != header)*

*val sportsRDD = rowsRDD.map{row => row.split(",")}.map{cols => sports(cols(0), cols(1), cols(2), cols(3), cols(4), cols(5), cols(6) )}*

*val sportsDF = sportsRDD.toDF()*

*sportsDF.registerTempTable("sports")*

*def rank = (medal:String, age:Int) => {*

*var subrank = "";*

*if(medal == "gold" && age >= 32){subrank = "pro";  }*

*else if(medal == "gold" && age <= 31){ subrank = "amateur";  }*

*else if(medal == "silver" && age >= 32){ subrank = "expert";  }*

*else if(medal == "silver" && age <= 31){ subrank = "rookie";  }*

*subrank*

*}*

*sqlContext.udf.register("ranks",rank)*

*val dataUDFMedal = sqlContext.sql( "SELECT ranks(medal\_type,age) as rank,medal\_type,age FROM sports")*

*dataUDFMedal.collect.foreach(println)*

