

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

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

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

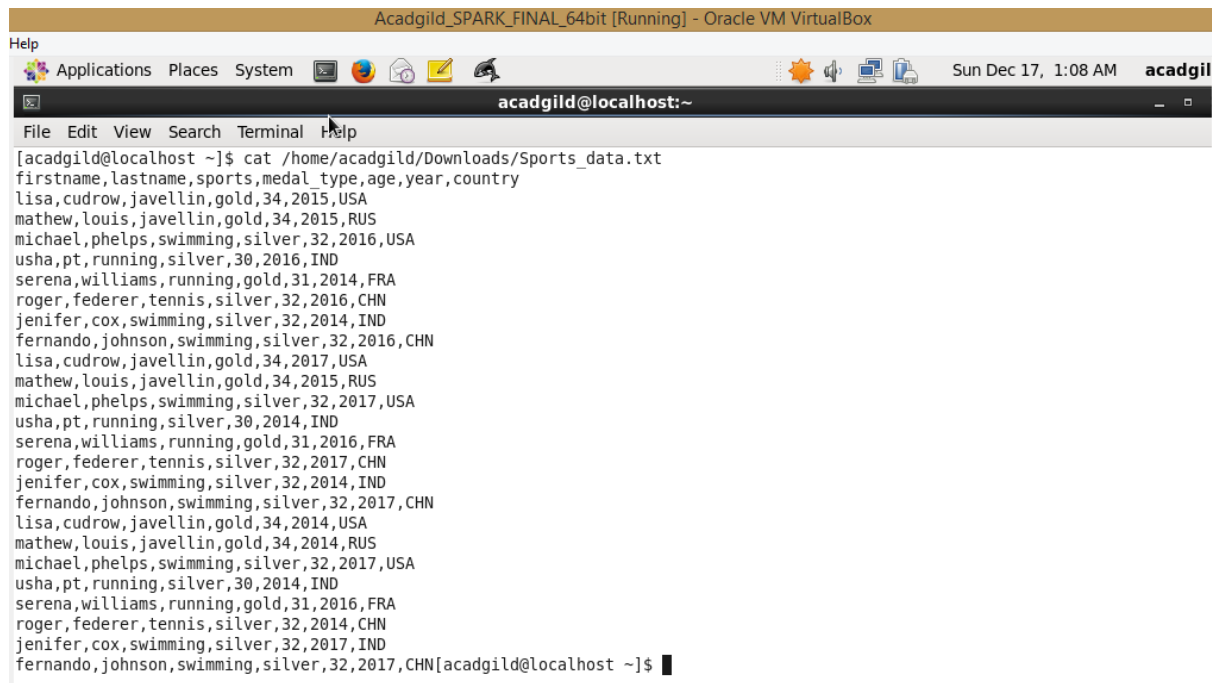
gold medalists, with age <= 31 are ranked amateur

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

silver medalists, with age <= 31 are ranked rookie

DataSet

The following dataset is a Sports Dataset with columns: **firstname, lastname, sports, medal_type, age, year, country**



```
Acadgild_SPARK_FINAL_64bit [Running] - Oracle VM VirtualBox
Help
Applications Places System [Icons] [Network] [Sound] [USB] [Printer] [CD/DVD] [Power] Sun Dec 17, 1:08 AM acadgil
acadgild@localhost:~
File Edit View Search Terminal Help
[acadgild@localhost ~]$ cat /home/acadgild/Downloads/Sports_data.txt
firstname,lastname,sports,medal_type,age,year,country
lisa,cudrow,javellin,gold,34,2015,USA
mathew,louis,javellin,gold,34,2015,RUS
michael,phelps,swimming,silver,32,2016,USA
usha,pt,running,silver,30,2016,IND
serena,williams,running,gold,31,2014,FRA
roger,federer,tennis,silver,32,2016,CHN
jenifer,cox,swimming,silver,32,2014,IND
fernando,johnson,swimming,silver,32,2016,CHN
lisa,cudrow,javellin,gold,34,2017,USA
mathew,louis,javellin,gold,34,2015,RUS
michael,phelps,swimming,silver,32,2017,USA
usha,pt,running,silver,30,2014,IND
serena,williams,running,gold,31,2016,FRA
roger,federer,tennis,silver,32,2017,CHN
jenifer,cox,swimming,silver,32,2014,IND
fernando,johnson,swimming,silver,32,2017,CHN
lisa,cudrow,javellin,gold,34,2014,USA
mathew,louis,javellin,gold,34,2014,RUS
michael,phelps,swimming,silver,32,2017,USA
usha,pt,running,silver,30,2014,IND
serena,williams,running,gold,31,2016,FRA
roger,federer,tennis,silver,32,2014,CHN
jenifer,cox,swimming,silver,32,2017,IND
fernando,johnson,swimming,silver,32,2017,CHN[acadgild@localhost ~]$
```

Importing Spark SQL Packages

```
scala> import org.apache.spark.sql._
import org.apache.spark.sql._

scala>

scala> import sqlContext.implicits._
import sqlContext.implicits._

scala> import org.apache.spark.sql.functions
import org.apache.spark.sql.functions
```

Converting text file into RDD with the help of SPARK CONTEXT object

```
scala> val sportsRDD = sc.textFile("/home/acadgild/Downloads/Sports_data.txt")
sportsRDD: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[163] at textFile at <console>:43
scala>
```

Put the first line of file into RDD which is header and remove the header from RDD for data manipulation

```
scala> val header = sportsRDD.first()
header: String = firstname,lastname,sports,medal_type,age,year,country

scala> val sportsdstaRDD = sportsRDD.filter(record => (record != header))
sportsdstaRDD: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[164] at filter at <console>:47

scala>
```

Printing Rdd which returns Array[String]

```
scala> sportsdstaRDD.foreach(println)
lisa,cudrow,javellin,gold,34,2015,USA
mathew,louis,javellin,gold,34,2015,RUS
michael,phelps,swimming,silver,32,2016,USA
usha,pt,running,silver,30,2016,IND
serena,williams,running,gold,31,2014,FRA
roger,federer,tennis,silver,32,2016,CHN
jenifer,cox,swimming,silver,32,2014,IND
fernando,johnson,swimming,silver,32,2016,CHN
lisa,cudrow,javellin,gold,34,2017,USA
mathew,louis,javellin,gold,34,2015,RUS
michael,phelps,swimming,silver,32,2017,USA
usha,pt,running,silver,30,2014,IND
serena,williams,running,gold,31,2016,FRA
roger,federer,tennis,silver,32,2017,CHN
jenifer,cox,swimming,silver,32,2014,IND
fernando,johnson,swimming,silver,32,2017,CHN
lisa,cudrow,javellin,gold,34,2014,USA
mathew,louis,javellin,gold,34,2014,RUS
michael,phelps,swimming,silver,32,2017,USA
usha,pt,running,silver,30,2014,IND
serena,williams,running,gold,31,2016,FRA
roger,federer,tennis,silver,32,2014,CHN
jenifer,cox,swimming,silver,32,2017,IND
fernando,johnson,swimming,silver,32,2017,CHN
```

Class Defination for sports data class and defining its respective schema

```
scala> case class
|   sportsdata(firstname:String,
|   lastname:String,
|   sports:String,
|   medal_type:String,
|   age:Integer,
|   year:Integer,
|   country:String)
defined class sportsdata

scala>
```

Mapping the record into word delimited by comma , defining the columns and converting to dataframe

```
scala> val rec = sportsdataRDD.map(x=> x.split(",")).map(x=>sportsdata(x(0),x(1),x(2),x(3),x(4).toInt,x(5).toInt,x(6))).toDF
rec: org.apache.spark.sql.DataFrame = [firstname: string, lastname: string, sports: string, medal_type: string, age: int, year: int, country: string]
scala>
```

Register temporary table sports for querying data

```
scala> rec.registerTempTable("sports")
scala>
```

Querying the entire table via sqlContext object and displaying the same

```
scala> sqlContext.sql("SELECT * FROM sports").show()
+-----+-----+-----+-----+-----+-----+-----+
|firstname|lastname| sports|medal_type|age|year|country|
+-----+-----+-----+-----+-----+-----+-----+
| lisa| cudrow|javellin| gold| 34|2015| USA|
| mathew| louis|javellin| gold| 34|2015| RUS|
| michael| phelps|swimming| silver| 32|2016| USA|
| usha| pt| running| silver| 30|2016| IND|
| serena|williams| running| gold| 31|2014| FRA|
| roger| federer| tennis| silver| 32|2016| CHN|
| jenifer| cox|swimming| silver| 32|2014| IND|
| fernando| johnson|swimming| silver| 32|2016| CHN|
| lisa| cudrow|javellin| gold| 34|2017| USA|
| mathew| louis|javellin| gold| 34|2015| RUS|
| michael| phelps|swimming| silver| 32|2017| USA|
| usha| pt| running| silver| 30|2014| IND|
| serena|williams| running| gold| 31|2016| FRA|
| roger| federer| tennis| silver| 32|2017| CHN|
| jenifer| cox|swimming| silver| 32|2014| IND|
| fernando| johnson|swimming| silver| 32|2017| CHN|
| lisa| cudrow|javellin| gold| 34|2014| USA|
| mathew| louis|javellin| gold| 34|2014| RUS|
| michael| phelps|swimming| silver| 32|2017| USA|
| usha| pt| running| silver| 30|2014| IND|
+-----+-----+-----+-----+-----+-----+-----+
only showing top 20 rows
```

- ✓ Using udfs on dataframe Change firstname, lastname columns into Mr.first_two_letters_of_firstname<space>lastname
for example - michael, phelps becomes Mr.mi phelps

Code:

Defining a UDFS for changing first name, last name

```
scala> def combinefirstlast=org.apache.spark.sql.functions.udf((firstname:String,lastname:String)=>{"Mr."+firstname+" "+lastname})
combinefirstlast: org.apache.spark.sql.UserDefinedFunction
```

```
scala> val newname =rec.withColumn("New Name",combinefirstlast(substring(rec("firstname"),1,2),rec("lastname")))
newname: org.apache.spark.sql.DataFrame = [firstname: string, lastname: string, sports: string, medal_type: string, age: int, year: int, country: string, New Name: string]
scala>
```

Output:

```
scala> newname.select("New Name").show()
+-----+
|      New Name|
+-----+
| Mr.li cudrow|
| Mr.ma louis|
| Mr.mi phelps|
| Mr.us pt|
| Mr.se williams|
| Mr.ro federer|
| Mr.je cox|
| Mr.fe johnson|
| Mr.li cudrow|
| Mr.ma louis|
| Mr.mi phelps|
| Mr.us pt|
| Mr.se williams|
| Mr.ro federer|
| Mr.je cox|
| Mr.fe johnson|
| Mr.li cudrow|
| Mr.ma louis|
| Mr.mi phelps|
| Mr.us pt|
+-----+
only showing top 20 rows

scala> █
```

- ✓ 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

Code:

Defining a UDFS for adding column ranking

```
scala> def findRankUDF=org.apache.spark.sql.functions.udf((age:Integer,medal_type:String)=>{
  | if(age >= 32 && medal_type == "gold") "pro"
  | else if(age <= 31 && medal_type == "gold") "amateur"
  | else if(age >= 32 && medal_type == "silver") "expert"
  | else if(age <= 31 && medal_type == "silver") "rookie"
  | else "No Ranking defined"})
findRankUDF: org.apache.spark.sql.UserDefinedFunction

scala>

scala> val rankUDF = rec.withColumn("ranking",findRankUDF(rec("age"),rec("medal_type")))
rankUDF: org.apache.spark.sql.DataFrame = [firstname: string, lastname: string, sports: string, medal_type: string, age: int,
year: int, country: string, ranking: string]

scala>
```

Output:

```
scala> rankUDF.show()
+-----+-----+-----+-----+-----+-----+-----+-----+
|firstname|lastname| sports|medal_type|age|year|country|ranking|
+-----+-----+-----+-----+-----+-----+-----+-----+
|    lisa|   cudrow|javellin|    gold| 34|2015|   USA|    pro|
|   mathew|   louis|javellin|    gold| 34|2015|   RUS|    pro|
| michael|  phelps|swimming|   silver| 32|2016|   USA| expert|
|    usha|    pt|running|   silver| 30|2016|   IND| rookie|
|   serena|williams|running|    gold| 31|2014|   FRA| amateur|
|   roger| federer|tennis|   silver| 32|2016|   CHN| expert|
| jenifer|    cox|swimming|   silver| 32|2014|   IND| expert|
|fernando| johnson|swimming|   silver| 32|2016|   CHN| expert|
|    lisa|   cudrow|javellin|    gold| 34|2017|   USA|    pro|
|   mathew|   louis|javellin|    gold| 34|2015|   RUS|    pro|
| michael|  phelps|swimming|   silver| 32|2017|   USA| expert|
|    usha|    pt|running|   silver| 30|2014|   IND| rookie|
|   serena|williams|running|    gold| 31|2016|   FRA| amateur|
|   roger| federer|tennis|   silver| 32|2017|   CHN| expert|
| jenifer|    cox|swimming|   silver| 32|2014|   IND| expert|
|fernando| johnson|swimming|   silver| 32|2017|   CHN| expert|
|    lisa|   cudrow|javellin|    gold| 34|2014|   USA|    pro|
|   mathew|   louis|javellin|    gold| 34|2014|   RUS|    pro|
| michael|  phelps|swimming|   silver| 32|2017|   USA| expert|
|    usha|    pt|running|   silver| 30|2014|   IND| rookie|
+-----+-----+-----+-----+-----+-----+-----+-----+
only showing top 20 rows

scala> █
```