

Session 19 Assignment 2

Problem 2 – To add a new column to dataframe using UDF

 bigdata@localhost:~/spark-2.3.0-bin-hadoop2.7/bin

```
scala> :q
[bigdata@localhost bin]$ ./spark-shell -i /home/bigdata/deepak/docs/Acadgild/Session19Assignment2/Session19Assignment2_2.scala

ranking: (medaltype: String, age: Int)String
output: org.apache.spark.sql.expressions.UserDefinedFunction = UserDefinedFunction(<function2>,<function2>,<function2>)
newssportsdataDF: org.apache.spark.sql.DataFrame = [firstname: string, lastname: string ... 6 more fields]
result: org.apache.spark.sql.DataFrame = [medal_type: string, age: string ... 1 more field]
+-----+-----+
|medal_type|age|ranking|
+-----+-----+
|      gold| 34|    pro|
|      gold| 34|    pro|
|    silver| 32| expert|
|    silver| 30| rookie|
|      gold| 31|amateur|
|    silver| 32| expert|
|    silver| 32| expert|
|    silver| 32| expert|
|      gold| 34|    pro|
|      gold| 34|    pro|
|    silver| 32| expert|
|    silver| 30| rookie|
|      gold| 31|amateur|
|    silver| 32| expert|
|    silver| 32| expert|
|    silver| 32| expert|
|      gold| 34|    pro|
|      gold| 34|    pro|
|    silver| 32| expert|
|    silver| 30| rookie|
+-----+-----+
only showing top 20 rows
```

Code Snapshot

```

Session19Assignment2_2.scala × Session19Assignment2_1.scala × Session14Assignment1.scala ×
/* Session19 Assignment 2 - UDFs on Dataframe */

//Problem 2 - Adding a new column using UDF
import spark.implicits._

//Read data from input file
val sportsdata=spark.read.csv("/home/bigdata/deepak/docs/Acadgild/Session19Assignment2/Sports_data.txt").toDF(
("firstname","lastname","sports","medal_type","age","year","country")

//Get the header
val header = sportsdata.first()

//Skip the header
val sportsdataDF = sportsdata.filter(line => line != header)

//UDF to decide the value of the new column - ranking
def ranking(medaltype: String, age: Int) = {
  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 ""
}

//Register the UDF
val output = udf(ranking(_:String, _:Int))

//Add the new column ranking using UDF
val newsportsdataDF=sportsdataDF.withColumn("ranking",output(sportsdataDF("medal_type"),sportsdataDF("age")))

//Select the new column in the output and display the output
val result=newsportsdataDF.select($"medal_type",$"age",$"ranking")
result.show

```