databricks

samp1 (Scala)

Import Notebook

```
val df = sqlContext.read.format("com.databricks.spark.csv").option("header","true").option("inferSchema","true").load("dbfs:/FileStore/shared_uploads/sinha.ashish.4.u@gmail.com/bankmarketingdata-3.csv")
df.show()
```

```
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+-----+--------+--------+-----+--------+--------+---+
|age|        job| marital|education|default|balance|housing|loan|contact|day|month|duration|campaign|pdays|previous|poutcome|  y|
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+-----+--------+--------+-----+--------+--------+---+
| 58|  management| married| tertiary|     no|   2143|    yes|  no|unknown|  5|  may|     261|       1|   -1|       0| unknown| no|
| 44|  technician|  single|secondary|     no|     29|    yes|  no|unknown|  5|  may|     151|       1|   -1|       0| unknown| no|
| 33|entrepreneur| married|secondary|     no|      2|    yes| yes|unknown|  5|  may|      76|       1|   -1|       0| unknown| no|
| 47| blue-collar| married|  unknown|     no|   1506|    yes|  no|unknown|  5|  may|      92|       1|   -1|       0| unknown| no|
| 33|     unknown|  single|  unknown|     no|      1|     no|  no|unknown|  5|  may|     198|       1|   -1|       0| unknown| no|
| 35|  management| married| tertiary|     no|    231|    yes|  no|unknown|  5|  may|     139|       1|   -1|       0| unknown| no|
| 28|  management|  single| tertiary|     no|    447|    yes| yes|unknown|  5|  may|     217|       1|   -1|       0| unknown| no|
| 42|entrepreneur|divorced| tertiary|    yes|      2|    yes|  no|unknown|  5|  may|     380|       1|   -1|       0| unknown| no|
| 58|     retired| married|  primary|     no|    121|    yes|  no|unknown|  5|  may|      50|       1|   -1|       0| unknown| no|
| 43|  technician|  single|secondary|     no|    593|    yes|  no|unknown|  5|  may|      55|       1|   -1|       0| unknown| no|
| 41|      admin.|divorced|secondary|     no|    270|    yes|  no|unknown|  5|  may|     222|       1|   -1|       0| unknown| no|
| 29|      admin.|  single|secondary|     no|    390|    yes|  no|unknown|  5|  may|     137|       1|   -1|       0| unknown| no|
| 53|  technician| married|secondary|     no|      6|    yes|  no|unknown|  5|  may|     517|       1|   -1|       0| unknown| no|
| 58|  technician| married|  unknown|     no|     71|    yes|  no|unknown|  5|  may|      71|       1|   -1|       0| unknown| no|
| 57|    services| married|secondary|     no|    162|    yes|  no|unknown|  5|  may|     174|       1|   -1|       0| unknown| no|
| 51|     retired| married|  primary|     no|    229|    yes|  no|unknown|  5|  may|     353|       1|   -1|       0| unknown| no|
| 45|      admin.|  single|  unknown|     no|     13|    yes|  no|unknown|  5|  may|      98|       1|   -1|       0| unknown| no|
| 57| blue-collar| married|  primary|     no|     52|    yes|  no|unknown|  5|  may|      38|       1|   -1|       0| unknown| no|
```

```
val bankDF = df.toDF()

bankDF: org.apache.spark.sql.DataFrame = [age: int, job: string ... 15 more fields]
```

```
bankDF.createOrReplaceTempView("bank")
```

```
val success = sqlContext.sql("select (a.subscribed/b.total)*100 as success_percent from (select count(*) as subscribed from bank where y='yes') a,(select count(*) as total from bank) b").show()

+------------------+
|   success_percent|
+------------------+
|11.698480458295547|
+------------------+

success: Unit = ()
```

```
val failure = sqlContext.sql("select (a.not_subscribed/b.total)*100 as failure_percent from (select count(*) as not_subscribed from bank where y='no') a,(select count(*) as total from bank) b").show()
```

```
+-----------------+
|  failure_percent|
+-----------------+
|88.30151954170445|
+-----------------+

failure: Unit = ()
```

```
bankDF.select(max($"age")).show()
```

```
+--------+
|max(age)|
+--------+
|      95|
+--------+
```

```
bankDF.select(min($"age")).show()
```

```
+--------+
|min(age)|
+--------+
|      18|
+--------+
```

```
bankDF.select(avg($"age")).show()
```

```
+-----------------+
|         avg(age)|
+-----------------+
|40.93621021432837|
+-----------------+
```

```
bankDF.select(avg($"balance")).show()
```

```
+------------------+
|      avg(balance)|
+------------------+
|1362.2720576850766|
+------------------+
```

```
val median = sqlContext.sql("SELECT percentile_approx(balance, 0.5) FROM bank").show()
```

```
+-----------------------------------------------------+
|percentile_approx(balance, CAST(0.5 AS DOUBLE), 10000)|
+-----------------------------------------------------+
|                                                  448|
+-----------------------------------------------------+

median: Unit = ()
```

```
val age = sqlContext.sql("select age, count(*) as number from bank where y='yes' group by age order by number desc ").show()
```

```
+---+------+
|age|number|
+---+------+
| 32|   221|
| 30|   217|
| 33|   210|
| 35|   209|
| 31|   206|
| 34|   198|
| 36|   195|
| 29|   171|
| 37|   170|
| 28|   162|
| 38|   144|
| 39|   143|
| 27|   141|
| 26|   134|
| 41|   120|
| 46|   118|
| 40|   116|
| 47|   113|
```

```
val marital = sqlContext.sql("select marital, count(*) as number from bank where y='yes' group by marital order by number desc ").show()
```

```
+--------+------+
| marital|number|
+--------+------+
| married|  2755|
|  single|  1912|
|divorced|   622|
+--------+------+

marital: Unit = ()
```

```
val age_marital = sqlContext.sql("select age, marital, count(*) as number from bank where y='yes' group by age,marital order by number desc ").show()
```

```
+---+-------+------+
|age|marital|number|
+---+-------+------+
| 30| single|   151|
| 28| single|   138|
| 29| single|   133|
| 32| single|   124|
| 26| single|   121|
| 34|married|   118|
| 31| single|   111|
| 27| single|   110|
| 35|married|   101|
| 36|married|   100|
| 25| single|    99|
| 37|married|    98|
| 33|married|    97|
| 33| single|    97|
| 32|married|    87|
| 39|married|    87|
| 38|married|    86|
| 35| single|    84|
```

```
import scala.reflect.runtime.universe
import org.apache.spark.SparkConf
import org.apache.spark.SparkContext
import org.apache.spark.sql.DataFrame
import org.apache.spark.sql.SQLContext
import org.apache.spark.sql.functions.mean
import org.apache.spark.ml.feature.StringIndexer

import scala.reflect.runtime.universe
```

```
val ageRDD = sqlContext.udf.register("ageRDD",(age:Int) => {
if (age < 20)
"Teen"
else if (age > 20 && age <= 32)
"Young"
else if (age > 33 && age <= 55)
"Middle Aged"
else
"Old"
})
```

```
ageRDD: org.apache.spark.sql.expressions.UserDefinedFunction = SparkUserDefinedFunction($Lambda$7158/708754380@c388944,StringType,List(Some(class[value[0]: int])),Some(class[value[0]: string]),Some(ageRDD),true,true)
```

```
val banknewDF = bankDF.withColumn("age",ageRDD(bankDF("age")))
banknewDF.createOrReplaceTempView("bank_new")
```

```
banknewDF: org.apache.spark.sql.DataFrame = [age: string, job: string ... 15 more fields]
```

```
val age_target = sqlContext.sql("select age, count(*) as number from bank_new where y='yes' group by age order by number desc ").show()
```

```
+-----------+------+
|        age|number|
+-----------+------+
|Middle Aged|  2601|
|      Young|  1539|
|        Old|  1131|
|       Teen|    18|
+-----------+------+

age_target: Unit = ()
```

```
val ageInd = new StringIndexer().setInputCol("age").setOutputCol("ageIndex")
```

```
ageInd: org.apache.spark.ml.feature.StringIndexer = strIdx_cd4353fd34ae
```

```
var strIndModel = ageInd.fit(banknewDF)
```

```
strIndModel: org.apache.spark.ml.feature.StringIndexerModel = StringIndexerModel: uid=strIdx_cd4353fd34ae, handleInvalid=error
```

```
val ageInd = new StringIndexer().setInputCol("age").setOutputCol("ageIndex")
```

```
ageInd: org.apache.spark.ml.feature.StringIndexer = strIdx_cd4353fd34ae
```

```
var strIndModel = ageInd.fit(banknewDF)
```

```
strIndModel: org.apache.spark.ml.feature.StringIndexerModel = StringIndexerModel: uid=strIdx_cd4353fd34ae, handleInvalid=error
```

```
strIndModel.transform(banknewDF).select("age","ageIndex").show(5)
```

```
+-----------+--------+
|        age|ageIndex|
+-----------+--------+
|        Old|     2.0|
|Middle Aged|     0.0|
|        Old|     2.0|
|Middle Aged|     0.0|
|        Old|     2.0|
+-----------+--------+
only showing top 5 rows
```