ip-10-0-1-10 login: emcsmarty\_gmail

Password:

Last login: Mon Jun 8 07:58:59 from 117.217.32.254

[emcsmarty\_gmail@ip-10-0-1-10 ~]$ spark2-shell

Setting default log level to "WARN".

To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).

20/06/08 08:35:16 WARN util.Utils: Service 'SparkUI' could not bind on port 42001. Attempting port 42002.

20/06/08 08:35:16 WARN util.Utils: Service 'SparkUI' could not bind on port 42002. Attempting port 42003.

20/06/08 08:35:16 WARN util.Utils: Service 'SparkUI' could not bind on port 42003. Attempting port 42004.

20/06/08 08:35:16 WARN util.Utils: Service 'SparkUI' could not bind on port 42004. Attempting port 42005.

20/06/08 08:35:16 WARN util.Utils: Service 'SparkUI' could not bind on port 42005. Attempting port 42006.

Spark context Web UI available at [http://ip-10-0-1-10.ec2.internal:42006](http://ip-10-0-1-10.ec2.internal:42006/)

Spark context available as 'sc' (master = yarn, app id = application\_1590984886986\_1499).

Spark session available as 'spark'.

Welcome to

\_\_\_\_ \_\_

/ \_\_/\_\_ \_\_\_ \_\_\_\_\_/ /\_\_

\_\ \/ \_ \/ \_ `/ \_\_/ '\_/

/\_\_\_/ .\_\_/\\_,\_/\_/ /\_/\\_\ version 2.4.0.cloudera2

/\_/

Using Scala version 2.11.12 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0\_144)

Type in expressions to have them evaluated.

Type :help for more information.

scala> import scala.reflect.runtime.universe

import scala.reflect.runtime.universe

scala>

scala> import org.apache.spark.SparkConf

import org.apache.spark.SparkConf

scala>

scala> import org.apache.spark.SparkContext

import org.apache.spark.SparkContext

scala>

scala> import org.apache.spark.ml.Pipeline

import org.apache.spark.ml.Pipeline

scala>

scala> import org.apache.spark.ml.classification.LogisticRegression

import org.apache.spark.ml.classification.LogisticRegression

scala>

scala> import org.apache.spark.ml.feature.Bucketizer

import org.apache.spark.ml.feature.Bucketizer

scala>

scala> import org.apache.spark.ml.feature.Normalizer

import org.apache.spark.ml.feature.Normalizer

scala>

scala> import org.apache.spark.ml.feature.StringIndexer

import org.apache.spark.ml.feature.StringIndexer

scala>

scala> import org.apache.spark.ml.feature.VectorAssembler

import org.apache.spark.ml.feature.VectorAssembler

scala>

scala> import org.apache.spark.mllib.evaluation.BinaryClassificationMetrics

import org.apache.spark.mllib.evaluation.BinaryClassificationMetrics

scala>

scala> import org.apache.spark.sql.DataFrame

import org.apache.spark.sql.DataFrame

scala>

scala> import org.apache.spark.sql.SQLContext

import org.apache.spark.sql.SQLContext

scala>

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

import org.apache.spark.sql.functions.mean

ultiline","true").json("/user/emcsmarty\_gmail/bank\_edited.json")

<console>:1: error: ';' expected but 'val' found.

val bank\_people\_data = spark.read.option("multiline","true").json("/user/emcsmarty\_gmail/bank\_edited.json")val bank\_people\_data = spark.read.option("multilin

e","true").json("/user/emcsmarty\_gmail/bank\_edited.json")

^

scala> val bank\_people\_data = spark.read.option("multiline","true").json("/user/emcsmarty\_gmail/bank\_edited.json")

bank\_people\_data: org.apache.spark.sql.DataFrame = [age: bigint, balance: bigint ... 15 more fields]

scala> bank\_people\_data.show()

+---+-------+--------+-------+---+-------+--------+---------+-------+------------+----+--------+-----+-----+--------+--------+---+

|age|balance|campaign|contact|day|default|duration|education|housing| job|loan| marital|month|pdays|poutcome|previous| y|

+---+-------+--------+-------+---+-------+--------+---------+-------+------------+----+--------+-----+-----+--------+--------+---+

| 58| 2143| 1|unknown| 5| no| 261| tertiary| yes| management| no| married| may| -1| unknown| 0| no|

| 44| 29| 1|unknown| 5| no| 151|secondary| yes| technician| no| single| may| -1| unknown| 0| no|

| 33| 2| 1|unknown| 5| no| 76|secondary| yes|entrepreneur| yes| married| may| -1| unknown| 0| no|

| 47| 1506| 1|unknown| 5| no| 92| unknown| yes| blue-collar| no| married| may| -1| unknown| 0| no|

| 33| 1| 1|unknown| 5| no| 198| unknown| no| unknown| no| single| may| -1| unknown| 0| no|

| 35| 231| 1|unknown| 5| no| 139| tertiary| yes| management| no| married| may| -1| unknown| 0| no|

| 28| 447| 1|unknown| 5| no| 217| tertiary| yes| management| yes| single| may| -1| unknown| 0| no|

| 42| 2| 1|unknown| 5| yes| 380| tertiary| yes|entrepreneur| no|divorced| may| -1| unknown| 0| no|

| 58| 121| 1|unknown| 5| no| 50| primary| yes| retired| no| married| may| -1| unknown| 0| no|

| 43| 593| 1|unknown| 5| no| 55|secondary| yes| technician| no| single| may| -1| unknown| 0| no|

| 41| 270| 1|unknown| 5| no| 222|secondary| yes| admin.| no|divorced| may| -1| unknown| 0| no|

| 29| 390| 1|unknown| 5| no| 137|secondary| yes| admin.| no| single| may| -1| unknown| 0| no|

| 53| 6| 1|unknown| 5| no| 517|secondary| yes| technician| no| married| may| -1| unknown| 0| no|

| 58| 71| 1|unknown| 5| no| 71| unknown| yes| technician| no| married| may| -1| unknown| 0| no|

| 57| 162| 1|unknown| 5| no| 174|secondary| yes| services| no| married| may| -1| unknown| 0| no|

| 51| 229| 1|unknown| 5| no| 353| primary| yes| retired| no| married| may| -1| unknown| 0| no|

| 45| 13| 1|unknown| 5| no| 98| unknown| yes| admin.| no| single| may| -1| unknown| 0| no|

| 57| 52| 1|unknown| 5| no| 38| primary| yes| blue-collar| no| married| may| -1| unknown| 0| no|

| 60| 60| 1|unknown| 5| no| 219| primary| yes| retired| no| married| may| -1| unknown| 0| no|

| 33| 0| 1|unknown| 5| no| 54|secondary| yes| services| no| married| may| -1| unknown| 0| no|

+---+-------+--------+-------+---+-------+--------+---------+-------+------------+----+--------+-----+-----+--------+--------+---+

only showing top 20 rows

scala> bank\_people\_data.registerTempTable("datanewtable"

|

|

You typed two blank lines. Starting a new command.

scala> bank\_people\_data.registerTempTable("datanewtable")

warning: there was one deprecation warning; re-run with -deprecation for details

scala> bank\_people\_data.select(max($"age")).show()

+--------+

|max(age)|

+--------+

| 95|

+--------+

scala> bank\_people\_data.select(min($"age")).show()

+--------+

|min(age)|

+--------+

| 18|

+--------+

scala> bank\_people\_data.select(avg($"age")).show()

+-----------------+

| avg(age)|

+-----------------+

|40.93621021432837|

+-----------------+

scala> bank\_people\_data.select(avg($"balance")).show()

+------------------+

| avg(balance)|

+------------------+

|1362.2720576850766|

+------------------+

scala> val median = spark.sql("SELECT percentile\_approx(balance, 0.5) FROM datanewtable").show()

+------------------------------------------------------+

|percentile\_approx(balance, CAST(0.5 AS DOUBLE), 10000)|

+------------------------------------------------------+

| 448|

+------------------------------------------------------+

median: Unit = ()

scala> val agedata = spark.sql("select age, count(\*) as number from datanewtable where y='yes' group by age order by number desc")

agedata: org.apache.spark.sql.DataFrame = [age: bigint, number: bigint]

scala>

scala> agedata.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|

| 25| 113|

| 47| 113|

| 42| 111|

+---+------+

only showing top 20 rows

scala> val maritaldata = spark.sql("select marital, count(\*) as number from datanewtable where y='yes' group by marital order by number desc")

maritaldata: org.apache.spark.sql.DataFrame = [marital: string, number: bigint]

scala>

scala> maritaldata.show()

+--------+------+

| marital|number|

+--------+------+

| married| 2755|

| single| 1912|

|divorced| 622|

+--------+------+

scala> val ageandmaritaldata = spark.sql("select age, marital, count(\*) as number from datanewtable where y='yes' group by age,marital order by number desc")

ageandmaritaldata: org.apache.spark.sql.DataFrame = [age: bigint, marital: string ... 1 more field]

scala>

scala> ageandmaritaldata.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|

| 39|married| 87|

| 32|married| 87|

| 38|married| 86|

| 35| single| 84|

| 47|married| 83|

| 31|married| 80|

+---+-------+------+

only showing top 20 rows

scala> val agedata = spark.udf.register("agedata",(age:Int) => {

|

| if (age < 20)

|

| "Teen"

|

| else if (age > 20 && age <= 32)

|

| "Young"

|

| else if (age > 33 && age <= 55)

|

| "Middle Aged"

|

| else

|

| "old"

|

| })

agedata: org.apache.spark.sql.expressions.UserDefinedFunction = UserDefinedFunction(<function1>,StringType,Some(List(IntegerType)))

scala> val banknewDF = bank\_people\_data.withColumn("age",agedata(bank\_people\_data("age")))

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

scala>

scala> banknewDF.show()

+-----------+-------+--------+-------+---+-------+--------+---------+-------+------------+----+--------+-----+-----+--------+--------+---+

| age|balance|campaign|contact|day|default|duration|education|housing| job|loan| marital|month|pdays|poutcome|previous| y|

+-----------+-------+--------+-------+---+-------+--------+---------+-------+------------+----+--------+-----+-----+--------+--------+---+

| old| 2143| 1|unknown| 5| no| 261| tertiary| yes| management| no| married| may| -1| unknown| 0| no|

|Middle Aged| 29| 1|unknown| 5| no| 151|secondary| yes| technician| no| single| may| -1| unknown| 0| no|

| old| 2| 1|unknown| 5| no| 76|secondary| yes|entrepreneur| yes| married| may| -1| unknown| 0| no|

|Middle Aged| 1506| 1|unknown| 5| no| 92| unknown| yes| blue-collar| no| married| may| -1| unknown| 0| no|

| old| 1| 1|unknown| 5| no| 198| unknown| no| unknown| no| single| may| -1| unknown| 0| no|

|Middle Aged| 231| 1|unknown| 5| no| 139| tertiary| yes| management| no| married| may| -1| unknown| 0| no|

| Young| 447| 1|unknown| 5| no| 217| tertiary| yes| management| yes| single| may| -1| unknown| 0| no|

|Middle Aged| 2| 1|unknown| 5| yes| 380| tertiary| yes|entrepreneur| no|divorced| may| -1| unknown| 0| no|

| old| 121| 1|unknown| 5| no| 50| primary| yes| retired| no| married| may| -1| unknown| 0| no|

|Middle Aged| 593| 1|unknown| 5| no| 55|secondary| yes| technician| no| single| may| -1| unknown| 0| no|

|Middle Aged| 270| 1|unknown| 5| no| 222|secondary| yes| admin.| no|divorced| may| -1| unknown| 0| no|

| Young| 390| 1|unknown| 5| no| 137|secondary| yes| admin.| no| single| may| -1| unknown| 0| no|

|Middle Aged| 6| 1|unknown| 5| no| 517|secondary| yes| technician| no| married| may| -1| unknown| 0| no|

| old| 71| 1|unknown| 5| no| 71| unknown| yes| technician| no| married| may| -1| unknown| 0| no|

| old| 162| 1|unknown| 5| no| 174|secondary| yes| services| no| married| may| -1| unknown| 0| no|

|Middle Aged| 229| 1|unknown| 5| no| 353| primary| yes| retired| no| married| may| -1| unknown| 0| no|

|Middle Aged| 13| 1|unknown| 5| no| 98| unknown| yes| admin.| no| single| may| -1| unknown| 0| no|

| old| 52| 1|unknown| 5| no| 38| primary| yes| blue-collar| no| married| may| -1| unknown| 0| no|

| old| 60| 1|unknown| 5| no| 219| primary| yes| retired| no| married| may| -1| unknown| 0| no|

| old| 0| 1|unknown| 5| no| 54|secondary| yes| services| no| married| may| -1| unknown| 0| no|

+-----------+-------+--------+-------+---+-------+--------+---------+-------+------------+----+--------+-----+-----+--------+--------+---+

only showing top 20 rows

scala> banknewDF.registerTempTable("banknewtable"

|

|

You typed two blank lines. Starting a new command.

scala> banknewDF.registerTempTable("banknewtable")

warning: there was one deprecation warning; re-run with -deprecation for details

scala> val targetage = spark.sql("select age, count(\*) as number from banknewtable where y='yes' group by age order by number desc")

targetage: org.apache.spark.sql.DataFrame = [age: string, number: bigint]

scala>

scala> targetage.show()

+-----------+------+

| age|number|

+-----------+------+

|Middle Aged| 2601|

| Young| 1539|

| old| 1131|

| Teen| 18|

+-----------+------+

scala> val agedata2 = new StringIndexer().setInputCol("age").setOutputCol("ageindex")

agedata2: org.apache.spark.ml.feature.StringIndexer = strIdx\_45d59241f13a

scala> var strindModel = agedata2.fit(banknewDF)

strindModel: org.apache.spark.ml.feature.StringIndexerModel = strIdx\_45d59241f13a

scala> 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