How to convert a dataframe to dataset in Apache Spark in Scala?

Asked 3 years, 9 months ago Active 2 years, 2 months ago Viewed 48k times

19

4

I need to convert my dataframe to a dataset and I used the following code:

```
val final_df = Dataframe.withColumn(
  "features",
  toVec4(
    // casting into Timestamp to parse the string, and then into Int
    $"time_stamp_0".cast(TimestampType).cast(IntegerType),
    $"count",
    $"sender_ip_1",
    $"receiver_ip_2"
  )
).withColumn("label", (Dataframe("count"))).select("features", "label")

final_df.show()

val trainingTest = final_df.randomSplit(Array(0.3, 0.7))
val TrainingDF = trainingTest(0)
val TestingDF=trainingTest(1)
TrainingDF.show()
TestingDF.show()

///lets create our liner regression
val lir= new LinearRegression()
.setRegParam(0.3)
.setElasticNetParam(0.8)
.setMaxIter(100)
.setTol(1E-6)

case class df_ds(features:Vector, label:Integer)
org.apache.spark.sql.catalyst.encoders.OuterScopes.addOuterScope(this)

val Training_ds = TrainingDF.as[df_ds]
```

My problem is that, **I got the following error:**

```
Error:(96, 36) Unable to find encoder for type stored in a Dataset.  Primitive types
(Int, String, etc) and Product types (case classes) are supported by importing
spark.implicits._  Support for serializing other types will be added in future
releases.
    val Training_ds = TrainingDF.as[df_ds]
```

It seems that the number of values in dataframe is different with the number of value in my class. However I am using `case class df_ds(features:Vector, label:Integer)` on my TrainingDF dataframe since, It has a vector of features and an integer label. Here is TrainingDF dataframe:

```
+--------------------+-----+
|            features|label|
+--------------------+-----+
|[1.497325796E9,19...|   19|
|[1.497325796E9,19...|   19|
|[1.497325796E9,19...|   19|
|[1.497325796E9,19...|   19|
|[1.497325796E9,19...|   19|
```

```
|[1.497325796E9,19...|   19|
|[1.497325796E9,19...|   19|
|[1.497325796E9,19...|   19|
|[1.497325796E9,19...|   19|
|[1.497325796E9,19...|   19|
|[1.497325796E9,19...|   19|
|[1.497325796E9,19...|   19|
|[1.497325796E9,19...|   19|
|[1.497325796E9,19...|   19|
|[1.497325796E9,19...|   19|
|[1.497325796E9,19...|   19|
|[1.497325796E9,19...|   19|
|[1.497325796E9,19...|   19|
|[1.497325796E9,19...|   19|
|[1.497325796E9,10...|   10|
+--------------------+-----+
```

Also here is my original **final_df** dataframe:

```
+------------+-----------+------------+-----+
|time_stamp_0|sender_ip_1|receiver_ip_2|count|
+------------+-----------+------------+-----+
|    05:49:56|   10.0.0.2|    10.0.0.3|   19|
|    05:49:56|   10.0.0.2|    10.0.0.3|   19|
|    05:49:56|   10.0.0.2|    10.0.0.3|   19|
|    05:49:56|   10.0.0.2|    10.0.0.3|   19|
|    05:49:56|   10.0.0.2|    10.0.0.3|   19|
|    05:49:56|   10.0.0.2|    10.0.0.3|   19|
|    05:49:56|   10.0.0.2|    10.0.0.3|   19|
|    05:49:56|   10.0.0.2|    10.0.0.3|   19|
|    05:49:56|   10.0.0.2|    10.0.0.3|   19|
|    05:49:56|   10.0.0.2|    10.0.0.3|   19|
|    05:49:56|   10.0.0.2|    10.0.0.3|   19|
|    05:49:56|   10.0.0.2|    10.0.0.3|   19|
|    05:49:56|   10.0.0.2|    10.0.0.3|   19|
|    05:49:56|   10.0.0.2|    10.0.0.3|   19|
|    05:49:56|   10.0.0.2|    10.0.0.3|   19|
|    05:49:56|   10.0.0.2|    10.0.0.3|   19|
|    05:49:56|   10.0.0.2|    10.0.0.3|   19|
|    05:49:56|   10.0.0.2|    10.0.0.3|   19|
|    05:49:56|   10.0.0.2|    10.0.0.3|   19|
|    05:49:56|   10.0.0.3|    10.0.0.2|   10|
+------------+-----------+------------+-----+
```

However I got the mentioned error! Can anybody help me? Thanks in advance.

scala apache-spark apache-spark-sql apache-spark-encoders

Share Improve this question Follow

edited Jan 4 '19 at 13:16

zero323
277k 77 841 866

asked Jun 13 '17 at 8:51

user8131063

2 Answers

Active Oldest Votes

The error message you are reading is a pretty good pointer.

30

When you convert a `DataFrame` to a `Dataset` you have to have a proper `Encoder` for whatever is stored in the `DataFrame` rows.

Encoders for primitive-like types (`Int` s, `String` s, and so on) and `case classes` are provided by just importing the implicits for your `SparkSession` like follows:

```
case class MyData(intField: Int, boolField: Boolean) // e.g.

val spark: SparkSession = ???
val df: DataFrame = ???

import spark.implicits._

val ds: Dataset[MyData] = df.as[MyData]
```

If that doesn't work either is because the type you are trying to *cast* the `DataFrame` to isn't supported. In that case, you would have to write your own `Encoder` : you may find more information about it here and see an example (the `Encoder` for `java.time.LocalDateTime`) here.

Share Improve this answer Follow

edited Feb 27 '18 at 13:26

answered Jun 13 '17 at 8:58

stefanobaghino

8,377 3 27 50

5

Spark 1.6.0

```
case class MyCase(id: Int, name: String)

val encoder = org.apache.spark.sql.catalyst.encoders.ExpressionEncoder[MyCase]

val dataframe = …

val dataset = dataframe.as(encoder)
```

Spark 2.0 or above

```
case class MyCase(id: Int, name: String)

val encoder = org.apache.spark.sql.Encoders.product[MyCase]

val dataframe = …

val dataset = dataframe.as(encoder)
```

Share Improve this answer Follow

edited Sep 10 '18 at 21:48

mana

5,431 5 42 66

answered Sep 10 '18 at 21:13

Shang Gao

51 1 2