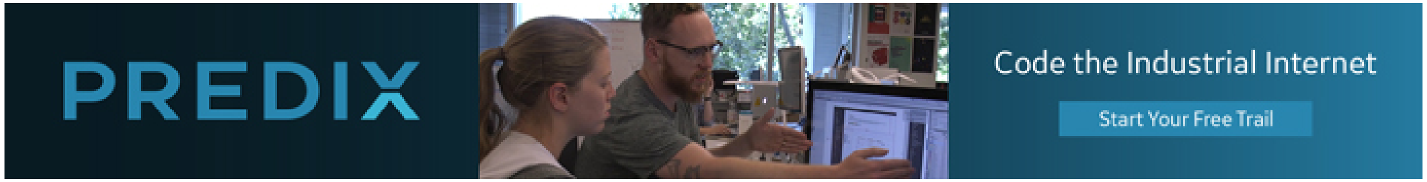
PySpark: Many features to Labeled Point RDD



New to Spark, and all examples I have read deal with small sets of data such as:

```
RDD = sc.parallelize([
LabeledPoint(1, [1.0, 2.0, 3.0]),
LabeledPoint(2, [3.0, 4.0, 5.0]),
```

However, I have a large dataset with 50+ features.

Example of a row

```
u'2596,51,3,258,0,510,221,232,148,6279,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

I want to quick create a Labeledpoint RDD in PySpark. I attempt to index the last position as my first data point in the Labeledpoint RDD, and then index the first n-1 positions as a dense vector. However I get the following error. Any guidance is appreciated! Note: if I change [] to () when creating the labeled point, I get the error "Invalid Syntax".

```
    df = myDataRDD.map(lambda line: line.split(','))
data = [
     LabeledPoint(df[54], df[0:53])
]
TypeError: 'PipelinedRDD' object does not support indexing
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-67-fa1b56e8441e> in <module>()
      2 df = myDataRDD.map(lambda line: line.split(','))
      3 data = [
----> 4      LabeledPoint(df[54], df[0:53])
      5 ]

TypeError: 'PipelinedRDD' object does not support indexing
```

apache-spark pyspark rdd apache-spark-mllib

edited Apr 25 at 10:56
zero323
65.9k 16 77 136

asked Sep 21 '15 at 3:05
adlopez15
13 3

For clarification, on you mentioning about the last position as your first data point, do you mean this as the label and the rest of the elements as the features for the LabaledPoint class? – Anchit Choudhry Oct 3 '15 at 20:16

2 Answers

As the error you get states you can not access an RDD by indices. You need a second `map` statement to transform your sequences into `LabeledPoint` s

```
rows =
[u'2596,51,3,258,0,510,221,232,148,6279,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

u'2596,51,3,258,0,510,221,232,148,6279,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0


rows_rdd = sc.parallelize(rows) # create RDD with given rows
```

```
labeled_points_rdd = rows_rdd\
                     .map(lambda row: row.split(','))\                  # split rows into
sequences
                     .map(lambda seq: LabeledPoint(seq[-1],seq[:-2]))   # create Labeled
Points from these sequences with last Item as label

print labeled_points_rdd.take(2)
# prints [LabeledPoint(5.0, [2596.0,51.0,3.0,258.0,0.0,510.0,221.0,...]),
#          LabeledPoint(5.0,[2596.0,51.0,3.0,258.0,0.0,510.0,221.0,...])
```

Note that the negative indices in python let you access sequences backwards.

With `.take(n)` you then an get the first `n` elements from your RDD.

Hope this helps.

edited Sep 23 '15 at 13:53 answered Sep 23 '15 at 13:34

ValD
66 5

You cannot use indexing, instead you have to use the methods available in the Spark API.
So:

```
data = [ LabeledPoint(myDataRDD.take(RDD.count()), #Last element
                      myDataRDD.top(RDD.count()-1)) #All but last ]
```

(Untested, nevertheless, this is the general idea)

answered Sep 21 '15 at 3:27

Dair
8,800 3 28 60

Thanks for the help, I believe this only works row-wise correct? How would I do this column-wise? –

adlopez15 Sep 21 '15 at 4:59