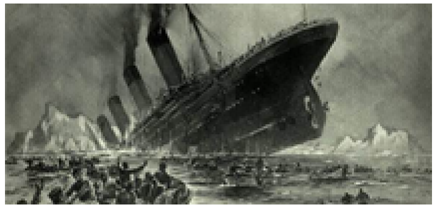

Knowledge • 4,499 teams

Titanic: Machine Learning from Disaster

Fri 28 Sep 2012 Sat 31 Dec 2016 (4 months to go)

My Titanic Survior in Spark/PySpark Diary

Start Watching

1 vote

Greetings, I admit that I was inspired by the R Diary I saw in the forums and by some of the amazing tutorials that people had put together for this project. That said, I did not see a "Getting Started In Spark" shortcut on the main page and I wish that there was one.

So, since there isn't and a scan of the forums does not turn up a Titanic Survivor solution in Spark, I am going to give it a shot, BUT ABSOLUTELY WELCOME ANY HELP AS I AM A TOTAL BEGINNER (which is why I'm on this project vs. the ones that pay $50k :)

To get started: I worked through a few of the posted tutorials and have a decent grasp of the models required for this project, therefore, I *think* have enough information to be dangerous, so here goes...

#1 | Posted 12 months ago

Chris C

0
votes

SETUP:

At this time, I am running this project on a Hadoop cluster, which uses HUE, and Spark (and a few other tools as well). If you do not already have an Hadoop cluster somewhere that you can practice on, you can download the Hortonworks VM. I plan on doing this tonight from another machine and will post a comment on any issues I ran into loading it and getting Spark to run.

Getting the data onto the HDFS was relatively simple as I just used HUE (the Hadoop User Experience), clicked on "File Browser" and then uploaded from my local machine into a newly created directory called "Data."

The next step was to fire up a terminal and run the command "PySpark". At this time, I am at the PySpark shell. A scan of StackOverflow indicates that I can load my data into an RDD via this command:

```
trainRDD = sc.textFile('hdfs://<hadoop location>/usr/home/<username>/Data/train.csv')`
```

For me, I found the location of HDFS via HUE in my browser window and just copied everything after the http:// and then appended my full HDFS directory.

This command seems to have worked as I only received "INFO" messages back, but now what? I asked around and it doesn't seem as if you can view the contents or health of your RDD (Resilent Distributed Dataset) in Spark.

But, it seems to be there, so pressing on.

I then loaded a couple of Machine Learning (ML) packages.

```
from pyspark.mllib.tree import RandomForest
from pyspark.mllib.util import MLUtils
```

So far, so good...

But, back to StackOverflow, I found **this** and decided to follow this example. Upon entering the line:

```
data = MLUtils.loadLibSVMFile(sc, 'hdfs://<yourServer>/usr/home/<username>/Data/train.csv')
```

It errors. I receive the following response:

```
Protocol message tag had invalid wire type
```

At this point, I'm stuck for the day. I'll do some more reading tonight and see what I can find.

#2 | Posted 12 months ago

Chris C

0
votes

Ok, problem solved. I had the wrong . For reasons that are beyond me at this point, the hostname that I used to load the file (which worked by the way), didn't work to do anything else after that.

I had to change the to our cluster's NameNode. This is what our system engineers told me to do and after that, everything worked fine.

So, if step one is getting the data loaded, let's do this again:

```
from pyspark import SparkContext
trainRDD = sc.textFile('<CORRECThostname>/usr/home/<username>/Data/train.csv').map(lambda
line: line.split(",")).map(lambda record: (record[0], record[1], record[2], record[3],
record[4], record[5], record[6], record[7], record[8], record[9], record[10], record[11]))
```

```
numRecords = data.count()
print "Total Records: %d" % numRecords
```

/** I went up to record[11] because there are 12 columns in the train data **/ And my output:

```
Total Records: 892
```

Boom! Data loaded

#3 | Posted 12 months ago

Chris C

0
votes

Hi, I'm trying to do the same for demonstration purposes, in scala. Here are a few comments about your posts:

First, I'm not sure what your "data" variable is in "data.count()". I assume you meant "trainRDD". However count() fails in my tests because of the newline at the end of the file creating a record with 0 values in it. Il solved it by adding a filter() before the last map() checking the number of records (or after textFile() to check only if the line is empty).

Some more problems:

- you'd have to remove the header line. Dropping the first line should be OK (or use a filter after parsing, see below). The expected number of records in the train.csv file is 891 (since they're numbered, you can check it by opening the file).
- splitting using the comma as a separator would be a good idea if there weren't present in passengers' names ("Braund, Mr. Owen Harris"). You end up with wrong records. Not sure if there's an elegant way to split properly unless specifying a regular expression or using a robust csv parser

Edit: In Java, you can split using this regex: line.split(",(?=([^\"]\"[^\"]\")[^\"]$)", -1). Found **here**. Quoting :

> split on the comma only if that comma has zero, or an even number of quotes ahead of it

- The first column will probably be a problem since in the LibSVM data files the first record contains the value to predict.

I'll post updates as I advance through my experiments and propose solutions if you're interested.

Hope it helps,

Julien

#4 | Posted 12 months ago | Edited 12 months ago

Julien Nauroy

0 votes

Hi Julien, thanks for the feedback and you are right on all counts. While I realize that this competition asked for R, Excel, or Python...it seems those have been completely covered and this would be a good time to add something new to the competition.

I also have come to the realization that to do this in Spark, I am going to have to learn Scala. And while I am suffering from "language fatigue" I'm going to learn the syntax and built-in functions so that I can get around in Spark. It didn't take me long to realize that most books, online tutorials, and examples regarding Spark were all in Scala and only occasionally in PySpark (and even then most people seemed to have chosen IPython).

Please post anything as you get it going, I'd love to see how you and others are doing this with Spark.

Chris.

#5 | Posted 12 months ago

Chris C

1 vote

Hi Chris,

I came to the same conclusion as you: in order to get into Spark, you have to learn Scala on the way. Truth is the Spark code you can find in Scala is always clean and concise.

I've come up with a first version of a Random Forest model using Scala and Spark. You can find the code attached to this post, as is. It's not very accurate (I scored around 0.74 which is more or less what the basic prediction "men die, women survive" yields). However there's a lot of room for improvement : first, it takes only part of the training dataset instead of the whole (since the MLLib demo takes 30% for evaluating the error in the prediction) and then some of the features are missing since the "points" have to be Doubles only. Plus, I'm clueless as to the values I should put into the "trainClassifier" method so I came up with a "higher should be better" approach which surely led to overfitting (my error was about 18% only on the training dataset).

That should serve you as a good kickstart for your experiments. If you find optimizations, please do share :)

Julien

1 Attachment —

SparkRandomForest.scala (3.14 KB)

#6 | Posted 12 months ago | Edited 12 months ago

Julien Nauroy

0
votes

Hi again,

I've realized from reading tutorials such as **Trevor Stehens'** that what's missing most for getting good results is data curation and defining new features. Thus I've separated the input data manipulation part from the machine learning algorithm and added "TODOs" where you can manipulate your data. I've also created a TitanicEntry class in order to organize the code a bit.

From now on doing simple analyses of the data becomes quite simple (though not as much as with R). As an example, counting the number of passenger grouped by (sex, survived) criterion gives:

```
trainCSV.map(record => (record.features("sex")+"/"+record.features("survived"), 1))
.reduceByKey(+).collect().foreach(println)
```

Julien

#7 | Posted 12 months ago

Julien Nauroy

0
votes

1 Attachment —

SparkRandomForestv2.scala (4.13 KB)

#8 | Posted 12 months ago

Julien Nauroy

1
vote

sorry to revive a dead thread. I had a quick go at doing this in Spark. It's not complete by any means (it generates predictions, but formatting it so it outputs the correct way for upload onto the server should be quite straightforward from here...)

It doesnt use loads of features, but it does use a decision tree in spark to do some classification. Havent tried to submit it yet so I dont know what the accuracy is like.

Basic flow is csv->Pandas Data Frame -> Cleaned Data Frame -> Spark Data Frame -> Spark Labeled Points ->Trained Algorithm

1 Attachment —

train.py (1.25 KB)

#9 | Posted 6 months ago

Henry

0
votes

Hi Henry,

Great to see an example in python. I find your code very clean and concise, perfect for a kickstart in using Spark with Python. Great job!

I didn't know about Pandas, I think I'll give it a try.

#10 | Posted 6 months ago

Julien Nauroy

1
vote

Thanks :)

It will certainly produce binary output, dont know how well it scored. I tried a Logistic Regression and an SVM but both of those gave horrendous (all 0) results. Dont know why. At least the decision tree did the job.

Pandas is a data frame object for Python. Possibly inherited from R (?) but its a nice way of loading all the data.

as a point, the pandas data frame is unnecessary. It is perfectly possible to do it without the pandas step - just by using tokens from a line.split, where line in file... The basic step would be to modify the map lambda function, to something like line:line.split(',')[...target...],[line.split(',')[variables...]] and then add a parallelise(....) around it. Havent tried to do it directly, but generally, thats how it should be doable.

I just like Pandas because it gives a nice data frame architecture which can be quite SQL like, and intuitive to manipulate. Plus for work recently I've been using a lot of R, so it comes more naturally in many ways now.

#11 | Posted 6 months ago | Edited 6 months ago

Henry

Reply You must be logged in to reply to this topic. **Log in »**

Start Watching « Back to forum

Our Team Careers Terms Privacy Contact/Support