Data Analysis – A Scala Project

ABSTRACT

This project is about exploring the 20GB of Tweets data available on Hadoop cluster.I tried to figure out what were the majority of tweets about, which led to some interesting revelations about huge surge of data in Oct Month of 2014. I also tried to put to practice all the methods I learnt in class.

Yogita Singh

SEIS 736-02

Outline

1. Introduction

2. Approach

3. Data source

4. Data description and schema

5. Data pre-processing (parsing, filtering, etc.)

6. Any bad data issues: Geolocation null for many.; Timestamp in string format. We tried sql function “extract” from datetime but couldn’t work as it was string.

7. Spark algorithm

8. Description of any other ecosystem or additional tools

9. Output description

10. How did I verify that my output was correct?

11. Performance/scale characteristics

12. What would I have done differently if I did this again?

13. Conclusions

Introduction

Twitter is an online news and social networking site, whereby users interact with each other via text messages, famously referred as “tweets”. In this project, I have used the “tweetfile” data in the directory named “tweets” stored on hdfs cluster. The datafile contains data of tweets for 3 years from 2013 to 2015 and the attributes were, ScreenName, Tweet, Language, Timestamp and GeoLocation.

Approach

I have tried to use all the methods that were taught in class to put everything to practice. Thus, I have used RDD programming model, dataframe model and by converting the dataframe to a relational table by registering it as temp table. I followed the following approach for this project:

- RDD: I used the RDD model for exploring the data. With RDD model I tried to figure out some information from all the data attributes. While doing so, I figured out that 99.98% of total tweets were done in Oct month of 2014 alone. Thus, I tried to probe further the reasons for that using dataframe programming model.
- Dataframe: As using RDD model I knew I needed to understand the data and if possible reasons for it. I tried certain questions focusing on it and was successful in getting the reasons for it.
- Dataframe to relational table: I used this method to find out any 3 questions from the data only to incorporate all the methods taught in class.

Data Source

I used the “allTweets” data stored in the directory named “tweets2” stored on hdfs cluster. The size of the data set was about 20 GB, 19.857 GB to be precise.

Data description and schema

It was twitter data, available on the Hadoop cluster. The data spanned over a period of 3 years from 2013 to 2015 most probably related to career.

Data had 5 key attributes namely, ScreenName, Tweet, Language, Timestamp and GeoLocation, with data type String. The attribute “Timestamp” had nested attributes, namely Day, Month, Date, Time, Time-zone & Year.

Data pre-processing

I worked upon the data by 3 ways as follows:

1) **By creating RDD:**

For this I created the RDD and then split each line at tab. After that I filtered it so that only those records with fields equal to 5 could pass, so that any record that seems incomplete is filtered out.

2) By creating Dataframe:

For this I created a dataframe by defining the case class for the five data attributes. After that I saved that dataframe named "df", on Hadoop cluster by writing that in parquet as "allTweets.parquet". Below is the screen capture of the dataframe "df" that I created first.

```
scala> val df = sqlContext.read.parquet("allTweets.parquet")
df: org.apache.spark.sql.DataFrame = [ScreenName: string, Tweet: string, Language: string, Timestamp: string, GeoLocation: string]

scala> df.show
+---------------+--------------------+--------+--------------------+-----------+
|     ScreenName|               Tweet|Language|           Timestamp|GeoLocation|
+---------------+--------------------+--------+--------------------+-----------+
|CareerProGlobal|MyPeopleBiz Blog:...|      en|Thu Sep 26 10:13:...|       null|
|     OllyGuseva|RT @trailof32: #9...|      ru|Thu Sep 26 12:12:...|       null|
|      trailof32|#99cents This Car...|      en|Thu Sep 26 15:20:...|       null|
|  paulregabooks|#99cents This Car...|      en|Thu Sep 26 15:20:...|       null|
|  paulregabooks|#99cents This Car...|      en|Thu Sep 26 19:20:...|       null|
|      trailof32|#99cents This Car...|      en|Thu Sep 26 19:20:...|       null|
|     Damexified|RT @TheITTimes: U...|      en|Thu Sep 26 22:02:...|       null|
|  coolniggal234|@StephMcMahon  so...|      en|Fri Sep 27 02:18:...|       null|
| AngieoftheMage|@Taegangers hi! :...|     fil|Fri Sep 27 06:37:...|       null|
|      pearcey33|@GLEMOODY there i...|      en|Fri Sep 27 12:08:...|       null|
|  paulageraghty|RT @YourRTEGuide:...|      en|Fri Sep 27 14:52:...|       null|
|Gottalove_Reezy|Aye anybody with ...|      en|Fri Sep 27 16:19:...|       null|
|       joycecom|Harness the power...|      en|Sat Sep 28 02:22:...|       null|
|       CIAngels|@TheHazelFaith th...|      en|Sat Sep 28 10:18:...|       null|
|      FWTruther|I'm anonymous bec...|      en|Sat Sep 28 22:38:...|       null|
|     ZaidJilani|One problem with ...|      en|Sun Sep 29 11:03:...|       null|
|      maggiebob|@OOK_Librarian I'...|      en|Sun Sep 29 15:52:...|       null|
| barleycoveband|RT @CariCole: Wan...|      en|Sun Sep 29 20:56:...|       null|
|kpowersofficial|HAPPY BIRTHDAY Gl...|      en|Mon Sep 30 03:22:...|       null|
|         APSOZA|How power poses c...|      en|Mon Sep 30 06:02:...|       null|
+---------------+--------------------+--------+--------------------+-----------+
only showing top 20 rows
```

As the column "Timestamp" further had 6 more nested attributes i.e. Day, Month, Date, Time, Time-zone & Year, I created one more dataframe from this named "Timestamp_df", to run queries related to Timestamp. To do this I mapped the RDD to extract the timestamp attribute and then created dataframe out of it by defining the case class for 6 sub attributes, timestamp had.

However, later I realized I wanted to do queries where I needed Timestamp's sub attributes along with the parent table to query them together. Thus I splitted the column Timestamp from the dataframe and the resultant dataframe was used to query. Below is the screen capture of the dataframe that I created next:

```
scala> myData.show
+---------------+--------------------+--------+--------------------+-----------+---+-----+----+--------+--------+----+
|     ScreenName|               Tweet|Language|           Timestamp|GeoLocation|Day|Month|Date|    Time|Timezone|Year|
+---------------+--------------------+--------+--------------------+-----------+---+-----+----+--------+--------+----+
|CareerProGlobal|MyPeopleBiz Blog:...|      en|Thu Sep 26 10:13:...|       null|Thu|  Sep|  26|10:13:00|     CDT|2013|
|     OllyGuseva|RT @trailof32: #9...|      ru|Thu Sep 26 12:12:...|       null|Thu|  Sep|  26|12:12:32|     CDT|2013|
|      trailof32|#99cents This Car...|      en|Thu Sep 26 15:20:...|       null|Thu|  Sep|  26|15:20:17|     CDT|2013|
|  paulregabooks|#99cents This Car...|      en|Thu Sep 26 15:20:...|       null|Thu|  Sep|  26|15:20:17|     CDT|2013|
|  paulregabooks|#99cents This Car...|      en|Thu Sep 26 19:20:...|       null|Thu|  Sep|  26|19:20:12|     CDT|2013|
|      trailof32|#99cents This Car...|      en|Thu Sep 26 19:20:...|       null|Thu|  Sep|  26|19:20:12|     CDT|2013|
|     Damexified|RT @TheITTimes: U...|      en|Thu Sep 26 22:02:...|       null|Thu|  Sep|  26|22:02:51|     CDT|2013|
|  coolnigga1234|@StephMcMahon  so...|      en|Fri Sep 27 02:18:...|       null|Fri|  Sep|  27|02:18:19|     CDT|2013|
| AngieoftheMage|@Taegangers hi! :...|     fil|Fri Sep 27 06:37:...|       null|Fri|  Sep|  27|06:37:33|     CDT|2013|
|      pearcey33|@GLEMOODY there i...|      en|Fri Sep 27 12:08:...|       null|Fri|  Sep|  27|12:08:19|     CDT|2013|
|  paulageraghty|RT @YourRTEGuide:...|      en|Fri Sep 27 14:52:...|       null|Fri|  Sep|  27|14:52:43|     CDT|2013|
|Gottalove_Reezy|Aye anybody with ...|      en|Fri Sep 27 16:19:...|       null|Fri|  Sep|  27|16:19:13|     CDT|2013|
|       joycecom|Harness the power...|      en|Sat Sep 28 02:22:...|       null|Sat|  Sep|  28|02:22:08|     CDT|2013|
|      CIAngels|@TheHazelFaith th...|      en|Sat Sep 28 10:18:...|       null|Sat|  Sep|  28|10:18:54|     CDT|2013|
|      FWTruther|I'm anonymous bec...|      en|Sat Sep 28 22:38:...|       null|Sat|  Sep|  28|22:38:56|     CDT|2013|
|     ZaidJilani|One problem with ...|      en|Sun Sep 29 11:03:...|       null|Sun|  Sep|  29|11:03:45|     CDT|2013|
|      maggiebob|@OOK_Librarian I'...|      en|Sun Sep 29 15:52:...|       null|Sun|  Sep|  29|15:52:05|     CDT|2013|
| barleycoveband|RT @CariCole: Wan...|      en|Sun Sep 29 20:56:...|       null|Sun|  Sep|  29|20:56:55|     CDT|2013|
|kpowersofficial|HAPPY BIRTHDAY Gl...|      en|Mon Sep 30 03:22:...|       null|Mon|  Sep|  30|03:22:37|     CDT|2013|
|         APSOZA|How power poses c...|      en|Mon Sep 30 06:02:...|       null|Mon|  Sep|  30|06:02:44|     CDT|2013|
+---------------+--------------------+--------+--------------------+-----------+---+-----+----+--------+--------+----+
only showing top 20 rows
```

3. By registering the dataframe in parquet as temp table

For this I used the dataframe I created in the previous section and registered it as temp table to use it further.

Bad data issues

- One of the attributes namely Geolocation was null for almost 100% of the data.
- The Timestamp attribute in the data had 6 sub attribute, namely Day, Month, Date, Time, Time-zone and Year respectively. Though it can't be called as bad data however, I couldn't segregate these sub attributes using an inbuilt function such as "extract" from timestamp, as it requires timestamp to be in a certain format to extract date or months etc. So when I tried to do it, it gave me the entire timestamp as a string in the first column and null for all the subsequent columns. So I had to divide it manually.

Spark algorithm

I used scala's RDD programming model first to explore the data and then used dataframe model for testing purpose and to verify the output is correct.

I also did a few of the questions by registering the dataframe as temp table so as to incorporate all the methods I learnt in this course.

Other ecosystem or additional tools

I used microsoft's Power BI to visualize the results and created some plots which are pasted below for your perusal. I extracted my result in excel at year, month and day level and used that as a source for Power BI desktop tool.

Output description

I first tried to explore the data by finding out the number of tweets per month throughout the time period of 3 years, for which the data was. When I did that, I got very interesting result which showed extremely skewed result, where Oct Month of 2014 got most number of tweets. This gave me a clear direction to move forward.

Then I moved forward and tried finding out the days in Oct month of 2014, for which the tweets were maximum. This resulted in Saturdays to be the days for which the tweets were maximum. Now I wanted to know if there were any specific dates for the surge and to my surprise it was one single date i.e Oct 11, 2014, which was responsible for such a huge number of tweets. This made me curious to find out the reason for such a surge in tweets on this particular day.

Thus I tried to find out the top 5 users who retweeted something.I got the following names with number of tweets:

[T_HADITIA,2397948]

[OllyGuseva,90]

[meOllyGuseva,86]

[AvailabilityUK,51]

[ITgrads,26]

This clearly shows one person with screen name "T_HADITIA" has retweeted the most. After this I figured out top 100 users as per the total number of tweets they did and I got the following names with the number of their tweets:

[asimohonda,2397948]

[T_HADITIA,2397948]

[galternativa,2397948]

[twittfrog,2397948]

[Geraldscamp,1198975]

[SocMediaNation,1198974]

[EchoingSoundz,1198974]

[Neha_Thakral,1198974]

[peter_handy,1198974]

[AAF_MAFA,1198974]

[OkanaganNow,1198974]

[adsociale,1198974]

[bloggerumer,1198974]

[KevinMorris101,1198974]

[SocialSnippet,1198974]

[OperationArmy,1198974]

[RLause,1198974]

[OperationRT,1198974]

[dainapeter,1198974]

[SEOvibes,1198974]

[CooeeMedia,1198974]

[TygrScott,1198974]

[rubensjose,1198974]

[IMJulieWatson,1198974]

[CapitalVA,1198974]

[Joe_Kolb,1198974]

[whitelabellocal,1198974]

[vandana_india,1198974]

[Nlt_21,1198974]

[GeorgeZisPaun,1198974]

[redzoneemb,1198974]

[Twibble_Test,1198974]

[LennJohnston,1198974]

[JoeTheSharer,1198974]

[Bajuluoflife,1198974]

[heyhleb,1198974]

[Guide2trick,1198974]

[rescarzaga,1198974]

[fentinak,1198974]

[raghav4web,1198974]

[CursoHootUNO,1198974]

[Maculous009,1198974]

[designsnake,1198974]

[AshkaITsolution,1198974]

[erin_mcneal,1198974]

[BitGadget,1198974]

[Work4Coffee,1198974]

[TheMehulPatel,1198974]

[ktsmithmelb,1198974]

[DwiCahya75,1198974]

[jisuotctmob,1198974]

[geeksntwits,1198974]

[4everabundance,1198974]

[World_Of_Hir,1198974]

[ReelMediaCoach,1198974]

[spotback,1198974]

[paulzuke,1198974]

[LGFundingGroup,1198974]

[HeathAJordan,1198974]

[igcstudios,1198974]

[alfrevela,1198974]

[FastWhisper,1198974]

[ZenFlint,1198974]

[Digitalcanyon,1198974]

[TalkRadiance,1198974]

[EpiphanyDM,1198974]

[IloveGAAMSocial,1198974]

[Ktownclassified,1198974]

[Games_Vale,1198974]

[Solo_Conectate,1198974]

[LeighScane,1198974]

[sarahalexandra3,1198974]

[FCC_ID,1198974]

[EfrainSalinasMX,1198974]

[SSMAAds,1198974]

[dimensionfour,1198974]

[bloggers,1198974]

[CreativeOctpus,1198974]

[DanDraperKC,1198974]

[GizmoFarm,1198974]

[Desarrollapp,1198974]

[paid4everything,1198974]

[Jkenton09,1198974]

[WebCastTVMedia,1198974]

[InteractiveIT,1198974]

[punithkumarn,1198974]

[shelbylaneMD,1198974]

[LarryAn1980,1198974]

[MySurryHills,1198974]

[RapidChangeNow,1198974]

[CarmenCiar,1198974]

[tamarsw,1198974]

[AnanseOnline,1198974]

[droidportfolio,1198974]

[mobilegurl,1198974]

[pattyblest,1198974]

[Timvmarketing,404]

[EngineersDay,136]

[Best_Schools,115]

[DSDBiz,92]

This clearly showed, same number of tweets count for a lot of screen names. Thus, I probed further. I tried to figure out the tweets done on Oct 11,2014 and took top 50 to get a glimpse of the tweets. It resulted in number of tweets with the same content as below:

"No Risk, No Reward: The Power of Risk-Taking for Your Career"

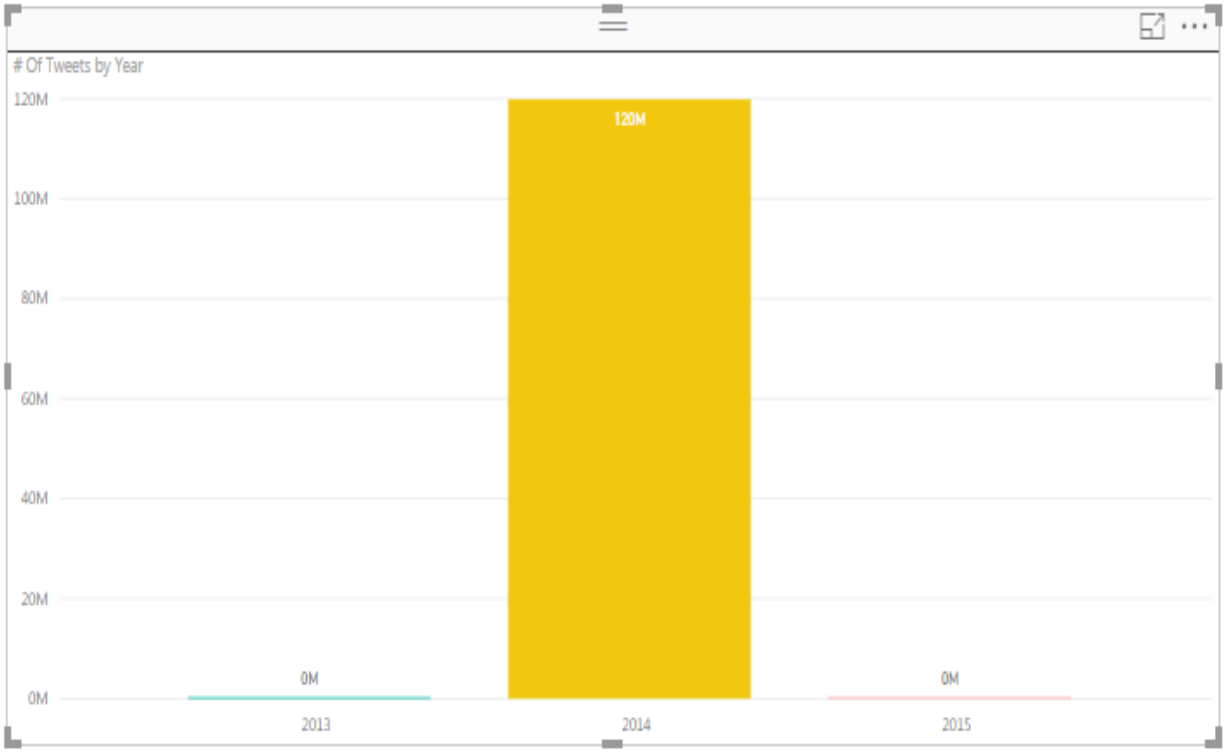
When I further probed by finding out the tweets for 3 randomly selected users from the list of users above, I found out that they tweeted the same tweet. I also noted that whereas T_HADITIA retweeted this ("No Risk, No Reward: The Power of Risk-Taking for Your Career"), however other 2 users that I randomly selected(galternativa and tamarsw), though not retweeted but shared the same post huge number of times.

When I googled this post ("No Risk, No Reward: The Power of Risk-Taking for Your Career"), I figured out that Mashable, a media company was about to launch its new UK website and a tool to predict which stories will go viral in the following week of Oct 11,2014. Moreover, this website published one article "No Risk, No Reward: The Power of Risk Taking for Your Career" on Oct 11, 2014. Thus, could be a marketing gimmick, which led to the surge of tweets and could have been all paid.

To make sure that I was on the right path, I also figured out the number of tweets which contain the string "No Risk, No Reward" and it resulted in 119897408 whereas total number of tweets on Oct 11,2014 were 119897438, which was quite close.

Following are some of the graphs I plotted using power BI:

Number of tweets by Year



No of tweets by Month, Year and Day

Top 100 most used words in tweets for all three years.

Verification of the output

I explored the data by RDD programming model first, tried to understand the story the data was telling and then did the questions (relevant to the story) by dataframe model. For verification of the questions in dataframe model, I used following strategies:

1) I did few of the questions(of the story) by RDD model
2) For some I used random filtering. For question 1st of dataframe model, to be sure that sep 2013 had only 187 posts. I filtered the tweets for Sep 2013 and checked the validity.
3) Some verified by spot checking through dataframe model itself. For example in question 4 of dataframe model, the number of retweets by the person named T_HADITIA, were coming out to be 2397948 and in next question the number of total tweets by this person were same. To check how was this possible, I spot checked in question 7 by filtering his tweets and it showed that he only retweeted the same post 2397948 times.

Performance/scale characteristics

The data source is around 20 GB in size. I computed the time by running randomly 3 queries of my code and I got time ranging from 92 , 102 and 118 seconds. I used the following code (Pasted for one of the queries):

```
val startTimeMillis = System.currentTimeMillis()val allTweetsRdd =
sc.textFile("/SEIS736/tweets2/allTweets").map(_.split("\t"))
//define the schema for DataFrame
case class allTweets(ScreenName: String, Tweet: String, Language: String, Timestamp: String,
GeoLocation: String)

val tweets = allTweetsRdd.filter(_.length == 5).map(x=> allTweets(x(0),x(1),x(2),x(3),x(4)))

val tweetsDF = tweets.toDF

tweetsDF.write.parquet("allTweets.parquet")

val df = sqlContext.read.parquet("allTweets.parquet")

val myData = df.withColumn("Day",split(col("Timestamp"),"
").getItem(0)).withColumn("Month",split(col("Timestamp"),"
").getItem(1)).withColumn("Date",split(col("Timestamp"),"
").getItem(2)).withColumn("Time",split(col("Timestamp"),"
").getItem(3)).withColumn("Timezone",split(col("Timestamp"),"
").getItem(4)).withColumn("Year",split(col("Timestamp")," ").getItem(5))

myData.groupBy("Month","Year").count.sort(desc("count")).collect.foreach(println)

val endTimeMillis = System.currentTimeMillis()
val durationSeconds = (endTimeMillis - startTimeMillis) / 1000

durationSeconds: Long = 92
```

What would I have done differently if I did this again?

First of all, I think I should have started working on this earlier than I did,so that I could incorporate a few more techniques/questions in this. So, I would avoid this the next time. Also, If I will do this again I would like to add the data output in terms of percentage also (like tweets per month, per person etc. as majority of tweets were done in Oct 2014 and done by a few people). Moreover, I would like to perform stopword elimination in the query for doing a word count on tweets attribute. I realized that I could do this(stopword elimination) with the data, only later in the course and then couldn’t find much time to work upon it.

Conclusions

Knowing scala helped a lot in simplifying such a challenging task of processing high volume of data and figuring out the story behind it. It was quite interesting to see how almost all of the 20 GB of data boiled down to just one day of tweet surge on Oct 11, 2014. I feel scala made it quite easy and interesting to do this project.