You need to create a Spark Application to perform sentiment analysis on tweets, and populate the sentiment polarity in an output file. The sentiment polarity contains either “Positive”, “Negative”, or “Neutral”.

**Populate all fields in the output file based on the following format:**

created\_at, text, screen\_name, source, location, followers\_count, friends\_count, retweet\_count, language, sentiment polarity



**Dataset**: blockchain.csv (tweets are in English)

**Fields in the dataset**: created\_at, text, screen\_name, source, location, followers\_count, friends\_count, retweet\_count, language

**Spark Application Template:** spark\_SA\_assessment.py

Step 1: Launch Firefox in VM, go to the training google drive to download the TWO files of *dataset* and s*park\_SA\_assessment.py*

Step 2: Move the dataset and s*park\_SA\_assessment.py* into **ipython** directory.

Step 3: Move the dataset in **ipython** directory to hdfs.

Step 4: Launch the spark service

Step 5: Use s*park\_SA\_assessment.py* as a start for your Spark application. Define Spark Context in one thread (**.setMaster**), and named the application (**.setAppName**)

Step 6: Ok, prepare the pipeline for sentiment analysis (Recommendation ONLY).

a. Identify the delimiter. [Hint: use split()]

b. Identify the length of fields [Hint: filter transformation; use len() and == sign] c. Remove all empty lines. [Hint: filter() transformation; len() index 0 > 1]

d. All tweets are in English. Do not need to perform translation.

e. Remove all the single (‘) and double(“)quotes

f. Use TextBlob package to identify the sentiment polarity (Hint: sentiment.polarity)

Step 7: You may use **.zip** to combine two RDDs. Remove all single and double quotes.

Step 8: Save the output by using **.saveAsTextFile()**

Step 9: Launch a new terminal, and run s*park\_SA\_assessment.py*

**Now, little bit of pre-processing is needed on the generated output file (as in step 8), prior ingesting into GIANT**

Step 10: Move the generated output file (*part-00000*) from hdfs (*/user/root/*) to local drive (*/root/ipython/*).

Step 11: In Jupyter, launch a Notebook

a) Run below code to label header for each field in *part-00000 (in local drive)*

**from pandas import read\_csv**

**label = read\_csv("part-00000")**

***#Label the column names according to the required output fields, below is the sample***

***label.columns = ["created\_at", "text", "screen\_name", "source", "location", "followers\_count", "friends\_count", "retweet\_count", "favourite\_count", "language", "sentiment\_polarity"]***

**label.to\_csv("part-00000headers1.csv",index = False)**

Step 12: Ingest the file with header into GIANT