Assignment 20.2

Step1: Download the demonetization\_tweets.csv from the Assignment Link given and copy to /home/acadgild/assignment\_20.2

Step2: Download AFINN.txt from the URL below

<https://raw.githubusercontent.com/wendykan/twitter-sentiment-analysis/master/AFINN-111.txt>

Step3: Process the tweets from demonetization-tweets.csv and store to temporary table tweets

* Create RDD from /home/acadgild/assignment\_20.2/demonetization-tweets.csv and put to tweets\_rdd\_with\_header
* Filter the header from tweets\_rdd\_with\_header and assign the RDD to tweets\_rdd
* Filter the records based on field separator comma (,) and filter all the records who has numbers of fields is greater than 2 and remove all quote characters, and map only first two fields and put the result to tweets\_filtered\_rdd
* Create a dataframe with two fields id, words from tweets\_filtered\_rdd to
* Create a temporary tables tweets
* Explode words and create a table tweet\_words

Code is as below:

val tweets\_rdd\_with\_header = sc.textFile("/home/acadgild/assignment\_20.2/demonetization-tweets.csv")

val header = tweets\_rdd\_with\_header.first()

val tweets\_rdd = tweets\_rdd\_with\_header.filter(row => row != header)

val tweets\_filtered\_rdd = tweets\_rdd.map(x => x.split(",")).filter(x=>x.length>=2).map(x => (x(0).replaceAll("\"",""),x(1).replaceAll("\"","").toLowerCase)).map(x => (x.\_1, x.\_2.split(" ")))

val tweets\_df = tweets\_filtered\_rdd.toDF("id","words")

tweets\_df.registerTempTable("tweets")

sqlContext.sql("select id as id,explode(words) as word from tweets").registerTempTable("tweet\_word")

Screenshot is as below:



Step4: Process AFINN.txt and store to temporary table afinn

* Create RDD from ("/home/acadgild/assignment\_20.2/AFINN.txt and put to afinn\_rdd
* Split words of afinn\_rdd based on tab and create a Dataframe afinn\_df with fields word, rating
* Create temporary table afinn

val afinn\_rdd = sc.textFile("/home/acadgild/assignment\_20.2/AFINN.txt")

val afinn\_df = afinn\_rdd.map(x => x.split("\t")).map(x => (x(0),x(1))).toDF("word","rating")

afinn\_df.registerTempTable("afinn")

sqlContext.sql("select t.id,AVG(a.rating) as rating from tweet\_word t join afinn a on t.word=a.word group by t.id order by rating desc").show(100)

Step5: Find the the rating for tweet

Join the tables tweet\_word and afinn on word and using avg function calculate average of rating group by tweet id, descending order by rating. Showing first 100 records of results

Code is as below:

sqlContext.sql("select t.id,AVG(a.rating) as rating from tweet\_word t join afinn a on t.word=a.word group by t.id order by rating desc").show(100)



