

## Bigdata Assignment 7.2

### Sentiment analysis on demonetization

Let us find out the views of different people on the demonetization by analysing the tweets from twitter. Here is the dataset where twitter tweets are gathered in CSV format. You can download the dataset from the below link

<https://drive.google.com/open?id=0ByJLBTmJojzNkRsZWJiY1VGc28>

Solution -

```
//Loaded the csv file
```

```
val tweets =
```

```
sc.textFile("file:///home/acadgild/RITESH/7.2/demonetization-  
tweets.csv").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(" "))).toDF("id","words")
```

```
//stored the header
```

```
val header = tweets.first
```

```
//Removed the header
```

```
val tweets_noheader = tweets.filter(row=> row!=header)
```

```
//Convert dataframe into table
```

```
tweets_noheader.registerTempTable("tweets")
```

```
//Create an array of words
```

```
val explode = spark.sql("select id as id,explode(words) as word from  
tweets").registerTempTable("tweet_word")
```

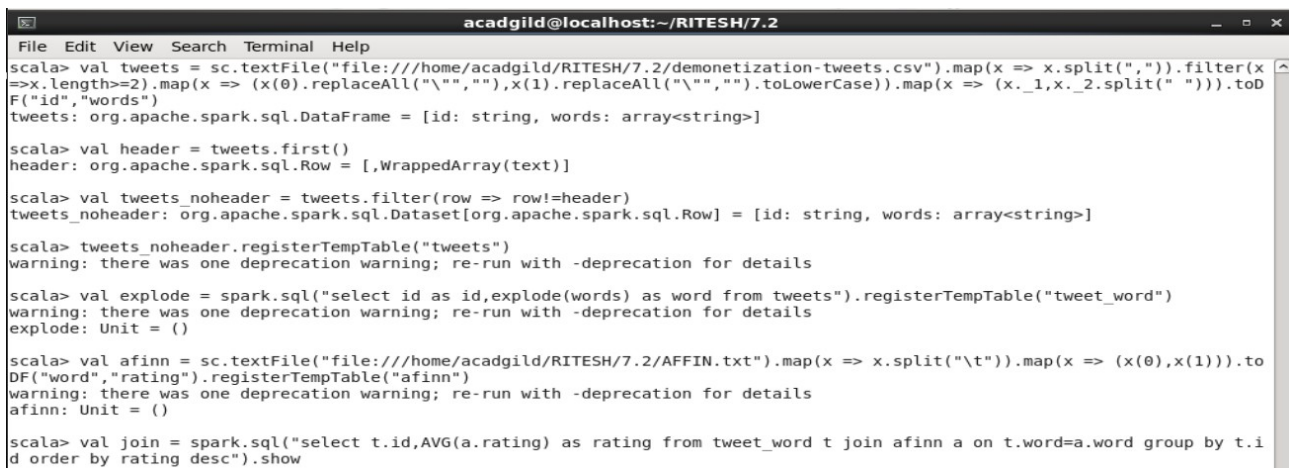
**//Load affin.txt file**

**val affin =**

**sc.textFile("file:///home/acadgild/RITESH/7.2/AFINN.txt").map(x=>x.  
split("\t")).map(x=>(x(0),x(1))).toDF("word","rating").registerTempT  
able("afinn")**

**//Perform a join operation to calculate avg rating and then sorted the  
ratings in descending order.**

**val join = spark.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**



```
acadgild@localhost:~/RITESH/7.2
File Edit View Search Terminal Help
scala> val tweets = sc.textFile("file:///home/acadgild/RITESH/7.2/demonetization-tweets.csv").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(" "))).toD
F("id","words")
tweets: org.apache.spark.sql.DataFrame = [id: string, words: array<string>]

scala> val header = tweets.first()
header: org.apache.spark.sql.Row = [,WrappedArray(text)]

scala> val tweets_noheader = tweets.filter(row => row!=header)
tweets_noheader: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [id: string, words: array<string>]

scala> tweets_noheader.registerTempTable("tweets")
warning: there was one deprecation warning; re-run with -deprecation for details

scala> val explode = spark.sql("select id as id,explode(words) as word from tweets").registerTempTable("tweet_word")
warning: there was one deprecation warning; re-run with -deprecation for details
explode: Unit = ()

scala> val afinn = sc.textFile("file:///home/acadgild/RITESH/7.2/AFFIN.txt").map(x => x.split("\t")).map(x => (x(0),x(1))).to
DF("word","rating").registerTempTable("afinn")
warning: there was one deprecation warning; re-run with -deprecation for details
afinn: Unit = ()

scala> val join = spark.sql("select t.id,AVG(a.rating) as rating from tweet_word t join afinn a on t.word=a.word group by t.i
d order by rating desc").show
```

Output -

```
+-----+
|  id|rating|
+-----+
|4185|  4.0|
|6610|  4.0|
|6546|  4.0|
|7281|  4.0|
|7994|  4.0|
|3822|  4.0|
|5733|  4.0|
|7025|  4.0|
| 308|  3.5|
|1500|  3.0|
|2654|  3.0|
|4144|  3.0|
|4484|  3.0|
|4862|  3.0|
|6491|  3.0|
|2696|  3.0|
|5829|  3.0|
|1497|  3.0|
|5473|  3.0|
|3494|  3.0|
+-----+
only showing top 20 rows

join: Unit = ()

scala> █
```

For each tweet , we got the rating.