**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. We will find the positive and negative sentiment on Demonetization based on the Tweets.  
  
**MetaData**

Id  
Text (Tweets)  
favorited  
favoriteCount  
replyToSN  
created  
truncated  
replyToSID  
id  
replyToUID  
statusSource  
screenname  
retweetCount  
isRetweet  
retweeted

1) Load the Demonetization tweets file.

grunt> load\_tweets\_fl = LOAD '/home/cloudera/khasimbabu/PIG\_Excercise/CaseStudy1/demonitization\_tweets.csv.txt' USING PigStorage(',');  
  
2) Extract ID,TweetText column values from the file.

grunt> extract\_details = FOREACH load\_tweets\_fl generate $0 as Id, $1 as TweetText;

grunt> describe extract\_details;  
extract\_details: {Id: bytearray,TweetText: bytearray}  
  
3) Split the TweetText into individual words. tweet\_text into words to calculate the sentiment of the whole tweet.

grunt> tokenize\_text = FOREACH extract\_details GENERATE Id,TweetText, FLATTEN(TOKENIZE(TweetText)) as word;

grunt> describe tokenize\_text;  
tokenize\_text: {Id: bytearray,TweetText: bytearray,word: chararray}  
  
4) Load Dictionary dataset for positive and negative words.

The AFINN is a dictionary which consists of 2500 words which are rated from +5 to -5 depending on their meaning.  
grunt> load\_dictionary = LOAD '/home/cloudera/khasimbabu/PIG\_Excercise/CaseStudy1/AFINN.txt' USING PigStorage('\t') as (word:chararray, rating:int);  
  
5) Join the Tweets dataset and Dictionary dataset for differentiate possitive and negative words.

grunt> join\_tweets\_dictionary = JOIN tokenize\_text by word LEFT OUTER, load\_dictionary by word USING 'replicated';

-> replicated is used for verifying all the tweet words with the dictionary words.

grunt> describe join\_tweets\_dictionary;  
join\_tweets\_dictionary: {tokenize\_text::Id: bytearray,tokenize\_text::TweetText: bytearray,tokenize\_text::word: chararray,load\_dictionary::word: chararray,load\_dictionary::rating: int}  
  
6) Get the rating for each word.

grunt> rating = FOREACH join\_tweets\_dictionary GENERATE tokenize\_text::Id as id,tokenize\_text::TweetText as text,load\_dictionary::rating as rate;  
  
7) Group the ID,Text for rating & Get the Average rating for each tweet.

grunt> grp\_word = GROUP rating by (id,text);

grunt> average\_rate = FOREACH grp\_word GENERATE group, AVG(rating.rate) as tweet\_rating;  
  
8) Classify Positive and Negative tweets.

grunt> positive\_tweet = FILTER average\_rate by tweet\_rating>=0;

grunt> negative\_tweet = FILTER average\_rate by tweet\_rating<0;