SENTIMENT ANALYSIS ON

DEMONETIZATION

USING PIG

ABSTRACT:

This project addresses the problem of sentiment analysis in twitter;

that is classifying tweets according to the sentiment expressed in them: positive, negative or neutral.

Twitter is an online micro-blogging and social-networking platform which allows users to write short status updates of maximum length 140 characters. It is a rapidly expanding service with over 200 million registered users

- out of which 100 million are active users and half of them log on twitter on a daily basis -generating nearly250 million tweets per day. Due to this large amount of usage we hope to achieve a reflection of public sentiment by analysing the sentiments expressed in the tweets.

Analysing the public sentiment is important for many applications such as

firms trying to find out the response of their products in the market, predicting political elections and predicting socioeconomic phenomena like stock exchange.

The aim of this project is to develop a functional classifier for accurate and automatic sentiment classification of an unknown tweet stream.

Sentiment Analysis On Demonetization Using Twitter Data:

**Summary:**

I am doing Sentiment Analysis on Demonetization across India by using data from twitter. For fetching the twitter data we can either use Apache Flume or download the available dataset from the internet.

After Fetching the data from twitter, it would be loaded directly to HDFS( Hadoop Distributed File System). This way we can reduce the extra overhead of transferring the data from local system to HDFS.

Next step deals with using the dictionary file to score the sentiment of each tweet by the number of positive words compared to number of negative words and then assigned a positive,negative or neutral sentiment value to each tweet.

At last I have generated the positive tweets and negative tweets and stored or loaded the results back to HDFS( Hadoop Distributed File System)

Data Sets and Tools Used:

The  dataset where twitter tweets are gathered in CSV format. The dataset has been downloaded and provided along.

The tool used here is PIG and HIVE. I am not using MapReduce Algorithm, since I want to do analysis on complete data and I don’t want to use aggregated measures.

I used Pig tool because Apache Pig is a dataflow language that is built on top of Hadoop to make it easier to process, clean and analyze "big data" without having to write vanilla map-reduce jobs in Hadoop.

Steps Performed and commands used:

Now we will load the data into [pig](https://bigishere.wordpress.com/2016/07/14/pig-programming-create-your-first-apache-pig-script/) using PigStorage as follows:

|  |  |
| --- | --- |
| 1 | load\_tweets = LOAD ‘/demonetizationtweets.csv’ USING PigStorage(‘,’); |

Now from this columns, we will extract the **id**and the **tweet\_text**as follows

|  |  |
| --- | --- |
| 1 | extract\_details = FOREACH load\_tweets GENERATE $0 as id,$1 as text; |

Now we will divide the tweet\_text into words to calculate the sentiment of the whole tweet.

|  |  |
| --- | --- |
|  | tokens = foreach extract\_details generate id,text, FLATTEN(TOKENIZE(text)) Asword; |

You can use the **describe tokens** command to check the schema of that relation and is as follows:

***tokens: {id: bytearray,text: bytearray,word: chararray}***

We will load the dictionary into pig by using the below statement:

|  |  |
| --- | --- |
|  | dictionary = load ‘/AFINN.txt’ using PigStorage(‘\t’)AS(word:chararray,rating:int); |

Now, let’s perform a map side join by joining the **tokens**statement and the dictionary contents using this relation:

|  |  |
| --- | --- |
|  | word\_rating = join tokens by word left outer, dictionary by word using‘replicated’; |

Now we will extract the **id,tweet text**and **word rating(**from the dictionary**)**by using the below relation.

|  |  |
| --- | --- |
|  | rating = foreach word\_rating generate tokens::id as id,tokens::text as text,dictionary::rating as rate; |

Now, we will group the **rating of all the words in a tweet**by using the below relation:

|  |  |
| --- | --- |
|  | word\_group = group rating by (id,text); |

Now, let’s perform the **Average** operation on the **rating of the words per each tweet**.

|  |  |
| --- | --- |
|  | avg\_rate = foreach word\_group generate group, AVG(rating.rate) astweet\_rating; |

Now we have calculated the Average rating of the tweet using the rating of each word.

From the above relation, we will get all the tweets i.e., both positive and negative.

Here, we can classify the positive tweets by taking the rating of the tweet which can be from **0-5.**We can classify the negative tweets by taking the rating of the tweet from **-5 to -1.**

Now we will filter the positive tweets using the below statement:

|  |  |
| --- | --- |
|  | positive\_tweets = filter avg\_rate by tweet\_rating>=0; |

Like this we will also filter the negative tweets as follows:

|  |  |
| --- | --- |
|  | negative\_tweets = filter avg\_rate by tweet\_rating<0; |

Sample tweets with positive rating:

((“7989”,“RT @rssurjewala: Critical question: Was PayTM informed about #Demonetization edict by PM? It’s clearly fishy and requires full disclosure &amp;�”),1.0)

Sample tweets with negative rating:

((“7997”,“RT @sukanyaiyer2: #DeMonetization AAP protests by marching Against Govts move over DeMonetization &amp; he is also detained as he Tried 2 March�”),–2.0)

**REFERENCES:**

[www.acadgild.com](http://www.acadgild.com/)