**Sentiment Analysis for the data collected from the application:**

**Step 1:** First collect the data from the application. The database used in this application generates the file that contains all the data gathered from the application.

**Step 2:** Copy this file to the hadoop environment and perform the following steps using Pig.

**Step 3:Load the dataset in pig.**

data=load 'fem.csv' using PigStorage(',') as (name:chararray,age1:int,occu:chararray.state:chararray,opinion:chararray);

**Step 4: Extract occupation, state and opinion details from the dataset:**

extract\_details = FOREACH data GENERATE $2 as occupation,$3 as state,$4 as text;

**Step 5: Now for the text we need to divide it into words to calculate sentiment analysis:**

tokens = foreach extract\_details generate occupation,state,text, FLATTEN(TOKENIZE(text)) As word;

**Step 6: Now load the dictionary file in pig:**

dictionary = load 'AFINN.txt' using PigStorage('\t') AS(word:chararray,rating:int);

**Step 7: 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';

**Step 8: Now we will extract the occupation, state, text and word rating(from the dictionary) by using the below relation.**

rating = foreach word\_rating generate tokens::occupation as occupation,tokens::state as state,tokens::text as text, dictionary::rating as rate;

**Step 9: Now, we will group the rating by occupation, state and opinion in the dataset by using the below relation:**

word\_group = group rating by (occupation,state,text);

**Step 10: Now, let’s perform the Average operation on the rating of the words for each opinion.**

avg\_rate = foreach word\_group generate group, AVG(rating.rate) as opinion\_rating;

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

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

**Step 11:Now we will filter the positive opinions using the below statement:**

positive\_opinion = filter avg\_rate by opinion\_rating>=0;

For negative opinions:

negative\_opinions= filter avg\_rate by opinion\_rating<0;

**STEP 12: To know the count first we need to group the opinions individually.**

grp= group positive\_opinions all;

pos\_count= foreach grp generate COUNT(positive\_opinions);

same with negative opinions

neg= group negative\_opinions all;

neg\_count= foreach neg generate COUNT(negative\_opinions);